

Table of Contents

Azure CSP Documentation

Overview

- Azure CSP overview
- Available Azure services
- Partner Center overview
- Partner shared services
- Azure CSP management options

Manage Azure CSP customers

- Create a new Azure CSP customer
- Add an existing Azure customer
- Assign permissions to an Azure CSP subscription
- Move an Azure CSP subscription to another CSP partner

Customer support in Azure CSP

- Support options for Azure CSP partners
- Escalate Azure CSP issues to Microsoft

Azure CSP billing

- Work with the price list
- Understand invoices

Integration with Azure CSP

- Azure CSP APIs
- Integration scenarios
 - Manage billing
 - Manage customers
 - Manage orders
 - Manage accounts and profiles
 - Customer support
 - Utilities

Migration to Azure CSP

- From on-premises to Azure CSP

Architectural design and considerations guide

Migrate initial workload

Advanced networking scenarios

Migrate additional workloads

Azure Site Recovery capacity planning

Azure Site Recovery setup and configuration

From Enterprise Agreement to Azure CSP

Partner EA to CSP

Customer EA to CSP

Migration assessment

Upgrade to Azure Resource Manager

Move Azure resources

From pay-as-you-go to Azure CSP

Learn how Microsoft partners who participate in the Azure Cloud Solution Provider (Azure CSP) program can use Azure services to deliver managed services to customers. Discover the Azure services that are available in Azure CSP. Learn about partner account management, customer management, billing, and integration with Azure CSP.

- □

[Learn about Azure CSP](#)

- □

[Learn how to integrate with Azure CSP](#)

- □

[Get started with migrations to Azure CSP](#)

Learn how Microsoft partners who participate in the Azure Cloud Solution Provider (Azure CSP) program can use Azure services to deliver managed services to customers. Discover the Azure services that are available in Azure CSP. Learn about partner account management, customer management, billing, and integration with Azure CSP.

- □

[Learn about Azure CSP](#)

- □

[Learn how to integrate with Azure CSP](#)

- □

[Get started with migrations to Azure CSP](#)

Azure CSP overview

10/6/2017 • 8 min to read • [Edit Online](#)

Welcome to the Azure documentation for Azure Cloud Solution Provider (Azure CSP) partners. This page gives you more information about the basics of the Azure CSP program and some specifics about Azure.

What is Azure CSP?

Azure CSP is a program for Microsoft partners. It's also a license channel for various Microsoft cloud services. In addition, it's a new way to provide added value for customers while creating an end-to-end relationship with them and becoming a trusted advisor for various Microsoft cloud services, including Azure.

The Azure CSP product portfolio includes Office 365, Dynamics 365, Enterprise Mobility + Security, Azure, and other Microsoft online services for business.

Azure CSP enables partners to have end-to-end ownership of the customer lifecycle and relationship for Microsoft Azure. That means partners are empowered to manage sales, own the billing relationship, provide technical and billing support, and be the customer's single point of contact. In addition, Azure CSP provides a full set of tools, including a self-service portal and accompanying APIs to easily provision, manage, and provide billing for customers and their subscriptions.

The [Partner Center](#) portal acts as an entry point for all Azure CSP partners. It provides rich customer management capabilities, automated processing, and more. Azure CSP partners can use Partner Center capabilities by using a web-based UI or by using [PowerShell and various API calls](#).

TIP

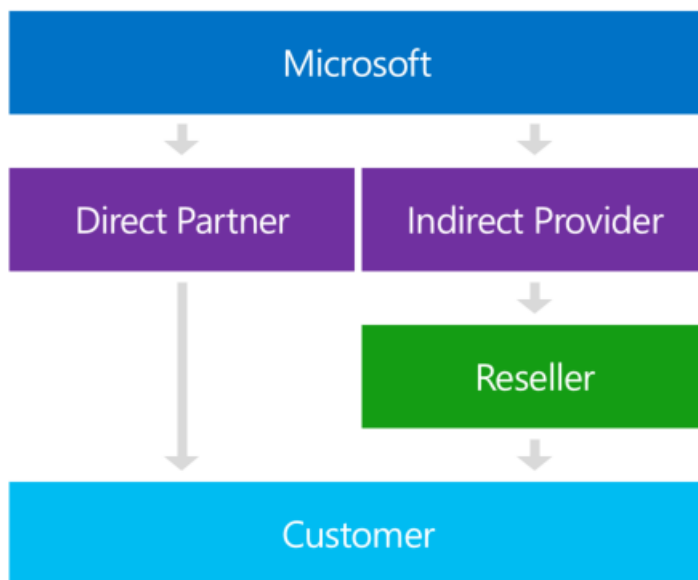
Check the [Azure in CSP 101](#) PowerPoint deck for a high-level program overview.

Azure CSP offers many benefits over traditional cloud reselling, including the following:

- *Bundle services more easily:* With Azure CSP, you can bundle third-party services (including your own unique solutions) with traditional Azure services to create a product that only you can provide.
- *Control pricing:* Because you own the relationship and Microsoft is not involved in any billing or invoicing, you can name your price based on the value that you believe you give customers.
- *Complete management:* You are completely in charge of managing your customer subscriptions. This means that you have the power to make any changes, additions, or removals yourself, without needing to go through Microsoft. (Don't worry, if you run into a problem you can't tackle on your own, you can still escalate issues to Microsoft.)
- *Tiers of investment:* If you like the idea of Azure CSP, but aren't quite ready to invest in full billing or support infrastructure, you can always become an indirect reseller. (See details in the following section.)

Azure CSP direct and Azure CSP indirect

There are two different business models in the Azure CSP program: *Azure CSP direct* (one tier) and *Azure CSP indirect* (two tiers).

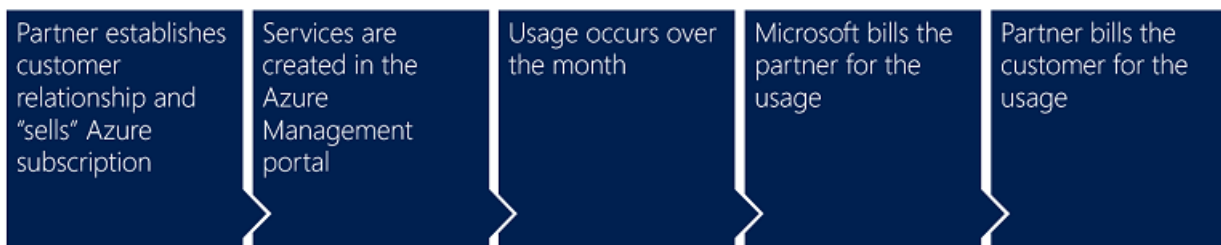


Watch this video to learn about the fundamental differences between the two models:

Azure CSP direct

Azure CSP direct (Tier 1) partners work with Microsoft directly. They take on the entire customer relationship, including support, billing, and invoicing. They become a customer's only point of contact for their Azure services. This provides continuity in the customer experience and helps build strong business relationships.

Azure CSP direct partners are responsible for customer support. Microsoft doesn't provide support for Azure CSP customers. This means Microsoft relies on Azure CSP partners to manage their Azure workloads and solve technical problems. To learn more about customer support in the Azure CSP program, see the [Azure CSP support overview](#) page.



Azure CSP direct partners are also responsible for customer billing. Microsoft provides partner-facing billing capabilities to Azure CSP direct partners through the Partner Center portal and APIs. Azure CSP direct partners are responsible for the customer-facing part of pricing, billing and invoicing. For more information about billing in the Azure CSP program, see the [Azure CSP billing overview](#) page.

The Azure CSP direct program requires that partners invest in the support and billing practices for the Microsoft cloud products that they want to deliver to their customers.

Azure CSP indirect

The Azure CSP indirect model (Tier 2) defines two types of partners: Azure CSP indirect providers (distributors) and Azure CSP indirect resellers. Azure CSP indirect providers work with Microsoft directly, but reach customers indirectly through their partner channel, Azure CSP resellers.

Azure CSP indirect reseller is a good choice for partners who don't want to manage as much infrastructure as an Azure CSP direct partner, so they team up with an indirect provider to handle their support, billing, and invoicing needs. They still build strong relationships with the customer and get many of the benefits of the Azure CSP program, but they offload support and billing to Azure CSP indirect providers.

A partner can be an Azure CSP indirect reseller for Azure (to reduce investments to Azure billing and support) and an Azure CSP direct partner at the same time (for Office 365, Dynamics 365, and other license-based products). Such mixes are not prohibited.

Enroll in the Azure CSP program

If you are a Microsoft partner and want to enroll in the Azure CSP program, visit the [Cloud Solution Provider program page](#). This page provides more information about the Azure CSP direct and indirect program requirements. You can enroll in any of those programs easily from this page.

The global Azure CSP program is divided into several multi-regional markets. If you want to transact in multiple markets, you must apply separately in every market. More details are available at [Enroll in the Cloud Solution Provider program](#).

Two national clouds--[Microsoft Azure Germany](#) and [Microsoft Azure Government](#)--have special application processes and require separate authorizations.

- To get access to Azure Germany, follow the instructions at [Enroll in the Cloud Solution Provider program for Microsoft Cloud Germany](#), and enroll in **Azure CSP for Microsoft Cloud Germany**.
- To get access to Azure Government, follow the instructions at [Enroll in the Cloud Solution Provider program for Microsoft Cloud for US Government](#), and enroll in **Azure CSP for Microsoft US Government Cloud**.

NOTE

[Azure China](#) is not available in the Azure CSP program.

Frequently asked questions

How is Azure CSP billed?

In the Azure CSP direct model, Microsoft bills the Azure CSP direct partner monthly based on the usage of all Azure CSP customers and all Azure CSP subscriptions. Partners receive an invoice for all charges, along with an accompanying reconciliation file that provides additional details. For more information about billing, see [Read your bill](#). In the Azure CSP indirect model, Microsoft bills Azure CSP providers in a similar way, and Azure CSP indirect providers set the billing rules for their Azure CSP indirect resellers.

How is Azure CSP priced?

Azure CSP pricing is specific to the services that are consumed. The partner sets the price for the customer. Detailed pricing for Azure services is available to Azure CSP direct partners via the [Partner Center pricing and offers page](#) or via the [RateCard API](#). Azure CSP indirect resellers should contact their Azure CSP indirect providers for Azure CSP pricing details.

I heard that not all Azure services are available in Azure CSP. Is that true?

Yes, it's true that not all Azure services are available in Azure CSP. Although some niche services are currently missing, all major Azure services are available, so it's definitely not a blocker for partners. Microsoft is working to create full-service parity between Azure CSP and other Azure partner models. For more information, see [Available Azure services in Azure CSP](#).

I want to be an Azure CSP partner. I want to focus on sales, and I don't want to worry about customer support because it's not my specialty. Is this possible?

Yes. In this case, you should consider enrolling in the Azure CSP indirect model as an Azure CSP indirect reseller. You focus on sales, and your Azure CSP indirect provider manages customer support. By the way, your Azure CSP indirect provider might also help you to increase the sales skills of your sellers and build a cloud sales practice if needed.

I want to enroll in the Azure CSP indirect program as an Azure CSP indirect reseller. I want to know who the Azure CSP indirect providers are in my region.

The list of all Azure CSP indirect providers in different countries is available at [Find a provider](#) . Note that not all Azure CSP indirect providers have Azure practice (support, billing, or sales enablement, for example). It's best to contact these providers directly and ask which capabilities they provide to accelerate your Azure business.

I prefer to manage Azure subscriptions of my customers through the [Azure portal](#). Why should I use [Partner Center](#)?

Partner Center is a portal where Azure CSP partners manage their customers and get information about different aspects of the Azure CSP program. Azure CSP direct partners also receive invoices from Microsoft through the Partner Center portal. Azure resource management inside a subscription (for example, creating a VM, configuring a virtual network, and so on) is still done through the [Azure portal](#). Learn more about Partner Center portal capabilities on the [Partner Center overview page](#).

I'm an Azure CSP partner and want to teach my support engineers how to work with Azure. Do you provide free Azure resources for testing purposes?

Yes. Make sure you are taking advantage of the [Partner Center sandbox capability](#) for non-production use. Every Azure CSP direct partner can create up to 375 free Azure subscriptions inside the sandbox. Azure CSP indirect resellers can ask their Azure CSP indirect providers to create sandbox subscriptions for them.

I'm an Azure CSP partner and I want to provide a managed cloud solution for my customer. But that customer already has an Azure subscription, purchased through an Enterprise Agreement. Is this a blocker?

The Azure CSP model supports multi-channel and multi-partner capabilities. One customer can have several Azure subscriptions, purchased by using different channels (for example EA, Open, or Azure CSP), or even from different Azure CSP partners. For more information, see [Work with other partners](#).

Is it possible to use CSP to deploy multi-tenant applications and solutions, that are spanned among different customers?

Yes, you can use [partner shared services](#) functionality for that.

I have a question about Azure CSP, and it is not covered in this article or in this FAQ section. Can you help me?

Sure! Just leave a comment below and we will answer your question.

Next steps

- [Review](#) which Azure services are available in Azure CSP.
- [Learn](#) how customer support works in the Azure CSP model.
- [Enroll](#) in the Azure CSP program and start creating business through Azure CSP.

Available Azure services in Azure CSP

2/27/2018 • 6 min to read • [Edit Online](#)

This article discusses the Azure services that are and are *not* available in the Azure Cloud Solution Provider (CSP) program. It also discusses service availability in the national clouds [Microsoft Azure Germany](#) and [Microsoft Azure Government](#).

NOTE

[Azure China](#) is not available in the Azure CSP program.

If you are a partner or customer who already uses Azure through [Pay-As-You-Go](#), [Enterprise Agreement](#), or other license channels, keep in mind that not all Azure services are currently available in the CSP program.

For example, because Azure CSP supports only the Azure Resource Manager model, non-Azure Resource Manager services are not available in the program. Several Azure Resource Manager services are also not available in the program. However, the situation is changing quickly.

Azure products, services, and features in Azure CSP

The following table displays a regularly updated list of the available first-party Azure products, services, and features in Azure CSP. For more information about services at the regional level, see [Products available by region](#).

AZURE PRODUCT, SERVICE, OR FEATURE	GLOBAL	US GOVERNMENT	GERMANY
Compute			
Virtual Machines	X	X	X
Cloud Services			
Virtual machine scale sets	X	X	X
Functions	X		
Azure Container Service	X		
Networking			
Azure DNS	X		
Content Delivery Network	X		
Traffic Manager	X		
ExpressRoute	X	X	X
Virtual Network	X	X	X

AZURE PRODUCT, SERVICE, OR FEATURE	GLOBAL	US GOVERNMENT	GERMANY
Load Balancer	X	X	X
VPN Gateway	X	X	X
Application Gateway	X	X	X
Network Watcher	X	X	X
Storage			
Storage	X	X	X
Backup	X	X	X
StorSimple	X		X
Site Recovery	X	X	X
Data Lake Store	X		
Managed Disks	X		X
Web + Mobile			
App Service	X	X	X
App Service on Linux	X		X
API Management	X	X	
Content Delivery Network	X		
Media Services	X	X	X
Notification Hubs	X	X	X
Azure Search	X		
Mobile Engagement			
Logic Apps feature of Azure App Service	X		
Containers			
App Service	X	X	X
App Service on Linux	X		X
Batch	X	X	X

AZURE PRODUCT, SERVICE, OR FEATURE	GLOBAL	US GOVERNMENT	GERMANY
Container Registry	X		
Container Instances	X		
Service Fabric	X	X	X
Container Service	X		
Databases			
SQL Database	X	X	X
Azure Cosmos DB	X	X	X
SQL Data Warehouse	X	X	X
Redis Cache	X	X	X
SQL Server Stretch Database	X	X	X
Azure Database for MySQL	X		
Azure Database for PostgreSQL	X		
Data + Analytics			
Databricks	X		
HDInsight	X	X	X
Stream Analytics	X		X
Data Factory	X		
Log Analytics	X	X	
Data Catalog	X		
Data Lake Store	X		
Data Lake Analytics	X		
Azure Analysis Services	X		
Power BI Embedded	X		
** AI + Cognitive Services **			

AZURE PRODUCT, SERVICE, OR FEATURE	GLOBAL	US GOVERNMENT	GERMANY
Machine Learning	X		X
Azure Bot Service	X		
Cognitive Services	X		
Azure Batch AI	X		
Internet of Things			
IoT Hub	X	X	X
Machine Learning	X		X
Stream Analytics	X		X
Event Hubs	X	X	X
Location Based Services	X		
Notification Hubs	X	X	X
Time Series Insights	X		
Enterprise Integration			
StorSimple	X	X	
API Management	X		
Event Grid	X		
Data Factory	X		
Service Bus	X	X	X
Data Catalog	X		
SQL Server Stretch Database	X		X
Logic Apps feature of Azure App Service	X		
Security + Identity			
Azure Active Directory	X	X	X
Multi-Factor Authentication	X		

AZURE PRODUCT, SERVICE, OR FEATURE	GLOBAL	US GOVERNMENT	GERMANY
Azure Active Directory Domain Services	X		
Azure Active Directory B2C			
Key Vault	X	X	X
Security Center	X	X	X
Developer Tools			
Visual Studio Team Services	X		
Application Insights	X		
DevTest Labs	X		
Visual Studio App Center	X		
Xamarin University	X		
Monitoring + Management			
Advisor	X		
Backup	X	X	X
Site Recovery	X	X	X
Scheduler	X	X	X
Automation	X	X	X
Log Analytics	X	X	
Azure Monitor	X		
Azure Managed Applications	X		
Azure Migrate	X		

Comments

- Some Azure network services are not region-specific. They can be deployed inside a global Azure subscription and used for national cloud purposes. These services include:
 - Azure Content Delivery Network
 - Azure DNS
 - Azure Traffic Manager

- [Non-Azure Resource Manager services](#) and the [Azure classic portal](#). Services that are not available in Azure CSP include:
 - *VMs, virtual networks, and storage accounts that were created by using the classic deployment model.*
 - [Azure Cloud Services](#): Consider switching to Azure Resource Manager-based VMs, App Service, and virtual machine scale sets.
 - *Mobile Services*: Use [Mobile Apps feature of Azure App Service](#) instead.
 - [Azure RemoteApp](#): This service is deprecated.
- [StorSimple Virtual Array Manager](#): This service is available in Azure CSP, but StorSimple Physical Array Manager is not. You can purchase and manage physical StorSimple devices only through an Enterprise Agreement.
- [Operations Management Suite \(OMS\) components](#): These components are available as separate Azure services. OMS E1 and E2 SKUs are not available in CSP.
- [Azure Active Directory Premium](#) and [Azure Multi-Factor Authentication](#): These services are available as license-based subscriptions in Partner Center, outside the regular Azure usage-based subscriptions.
- [Azure Active Directory Domain Services](#) - This service is now available in CSP. For more information see [Azure Active Directory \(AD\) Domain Services for Azure Cloud Solution Providers \(CSP\)](#).
- [Azure Active Directory B2C](#): This service is currently not available in Azure CSP. The estimated launch timeframe is middle of 2018.
- [Azure App Service](#): There is no support for integration with third-party Azure Marketplace applications (for example, ClearDB) because third-party Marketplace applications aren't enabled for Azure CSP subscriptions yet.
- [Key Vault](#): The process for setting up Key Vault for Azure CSP subscriptions is the same as that for other types of Azure subscriptions (Enterprise Agreement, Pay-As-You-Go). However, the vault access control list (ACL) policy must be set up manually. For more information, see [Setting up Key Vault for Azure CSP subscriptions](#).
- [Azure Time Series Insights](#): This service is available in Azure CSP, but please note that only users from the customer tenant can access data in their Time Series Insights environment. Partners can manage their customer's Time Series Insights environment by default, but if they need access to the data in it, they must be added to the customer tenant. For more information see [Azure Time Series Insights](#).

Visual Studio Marketplace

You can now purchase items from Visual Studio Marketplace with the exception of third party extensions for your customers. This includes:

- [Visual Studio Team Services](#)
- [Visual Studio cloud subscriptions](#)
- [Xamarin University training](#)

To help get you started, we have created videos and documentation about [how to set up, purchase, and manage Visual Studio](#) in CSP.

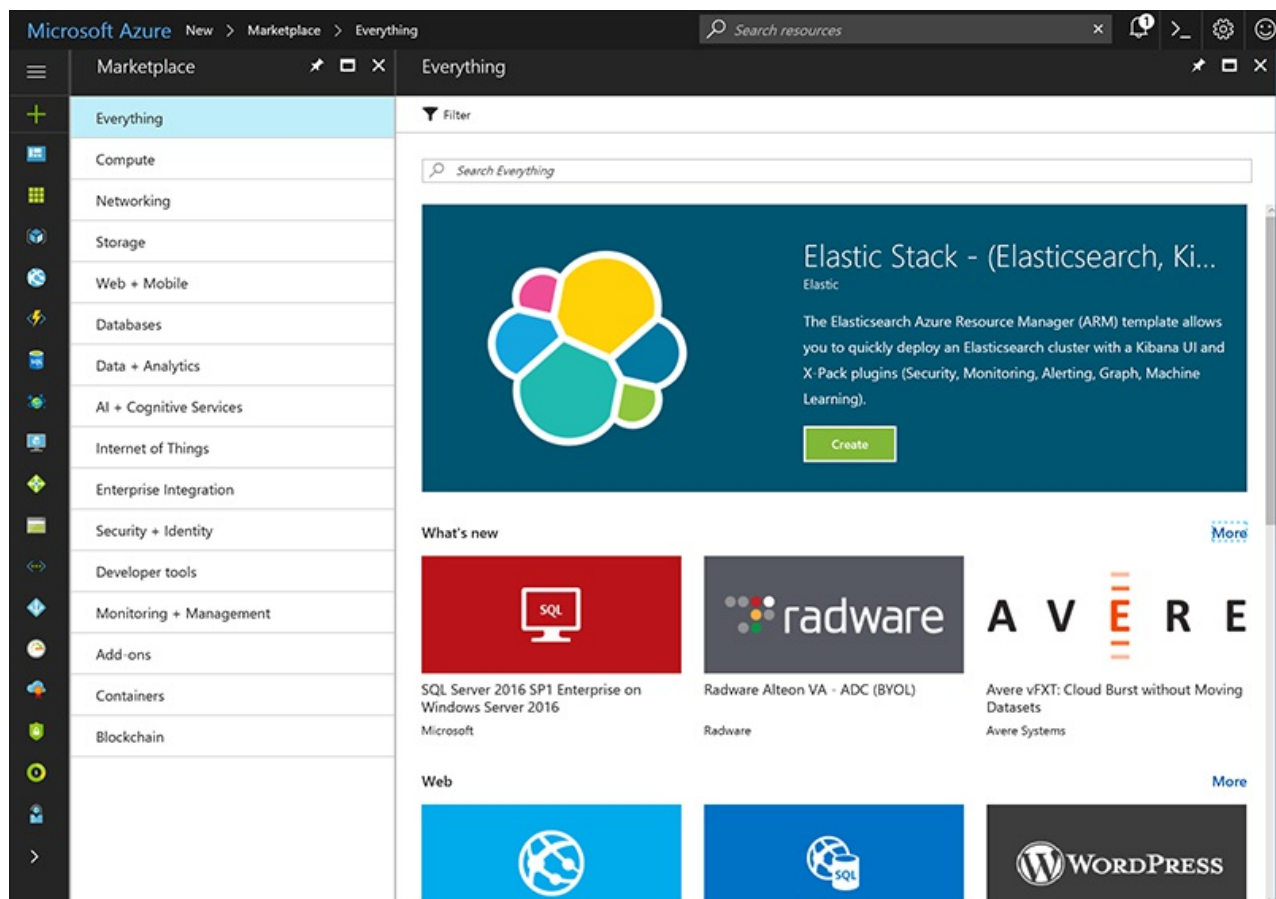
TIP

Check Azure Services in the Azure CSP Release Notes on the [Partner Center Pricing and Offers page](#) (requires authorization).

Azure Marketplace items in Azure CSP

Not all Azure Marketplace items are currently available in Azure CSP subscriptions.

- *Microsoft-based Azure services:* These services are available. Review the previous table and comments.
- *Bring your own license (BYOL) items:* These items are available. A full list of BYOL-enabled Azure Marketplace items is available on the [Azure Marketplace BYOL page](#).
- *Pay-As-You-Go third-party Azure Marketplace items:* These items are *not* available, with few exceptions like Red Hat Enterprise Linux.
- *Citrix XenApp Essentials:* Partners can purchase XenApp Essentials for customers in CSP. For more information, see Citrix Blog- [Distribution of XenApp Essentials now available through Microsoft Cloud Solution Provider Channel](#).



Though Pay-As-You-Go Azure Marketplace items are not available in Azure CSP, many of these services can still be obtained for a customer. One of the strengths of the Azure CSP model is that an Azure CSP partner can bundle Azure services and BYOL items with third-party software licenses, which are purchased through a separate partner channel (outside Azure CSP).

In this way, a customer can use a third-party product that's deployed in Azure and then pay for it in a single bill.

Next steps

- [Learn](#) about the available capabilities for Azure in Partner Center.
- [Create](#) your first customer in Azure CSP, and deploy Azure services.

Partner Center overview

10/17/2017 • 4 min to read • [Edit Online](#)

The [Partner Center](#) portal acts as an entry point for all Azure Cloud Solution Provider (CSP) partners. It provides rich customer management capabilities and automated processing. CSP partners can use Partner Center through a web-based UI, or by using [PowerShell](#) and various [API calls](#).

The following short video introduces you to basic concepts and functions of Partner Center:

IMPORTANT

Partner Center is a portal for CSP partners only. It is not available for CSP customers.

Partner Center capabilities for Azure

On the Partner Center portal, CSP direct partners, CSP indirect providers, and CSP indirect resellers can:

- Manage customers and create Azure subscriptions.
- Manage customers' Azure resources on their behalf.
- Manage a partner's [profile](#).
- Manage a partner's [user accounts](#), and assign permissions to internal staff.
- Create a [marketing profile](#), and respond to referrals.

In addition, CSP direct partners and CSP indirect providers can:

- View a customer's Azure usage.
- Receive invoices from Microsoft, view reconciliation files, and check invoice payment status.
- Get Azure CSP pricing.
- [Escalate issues](#) with Partner Center to Microsoft, and track ticket progress.

CSP indirect resellers should use solutions, provided by their CSP indirect providers, for support escalations, billing, and pricing.

Partner Center user roles

Partner Center access is role-based, so the user's view can be customized to only show features that the user needs. Note that a user's role can be changed when necessary through Partner Center web UI or APIs.

Each user has two permission levels:

- **Agent permissions** control what kind of customer data and account information the user can see and change. These settings affect all CSP customers managed in this Partner Center account.
- **Admin permissions** control the level of access the user has to Partner Center features. This setting has impact outside Partner Center. For example, a billing admin can access bills for all Microsoft services (even those unrelated to CSP), and a global admin can access user accounts and customer accounts beyond CSP as well.

Account settings

Partner profile

Partner billing profile

Organization profile

User management

Add user

Catalog

App Management

Integration sandbox

Web storefront

Test drive new features

[← Overview](#)

Add user

Name *

John Johnson

Email *

john

@contoso.onmicrosoft.com

Assists your customers as [Learn more](#)☒ Admin agent

Has access to all Partner Center features

☐ Sales agent

Adds customers and manages their subscriptions

☐ Helpdesk agent

Helps resolve customer issues with their services

☐ Not an agentManages your company as [Learn more](#)☐ Global admin

Has full access to all administrative and Partner Center features

☐ Billing admin

Can view invoices and billing information

☐ User management admin

Creates agents and resets their passwords

☒ Not an admin

Add

Cancel

You can find more details about available Partner Center user roles in the article [Create user accounts and set permissions](#).

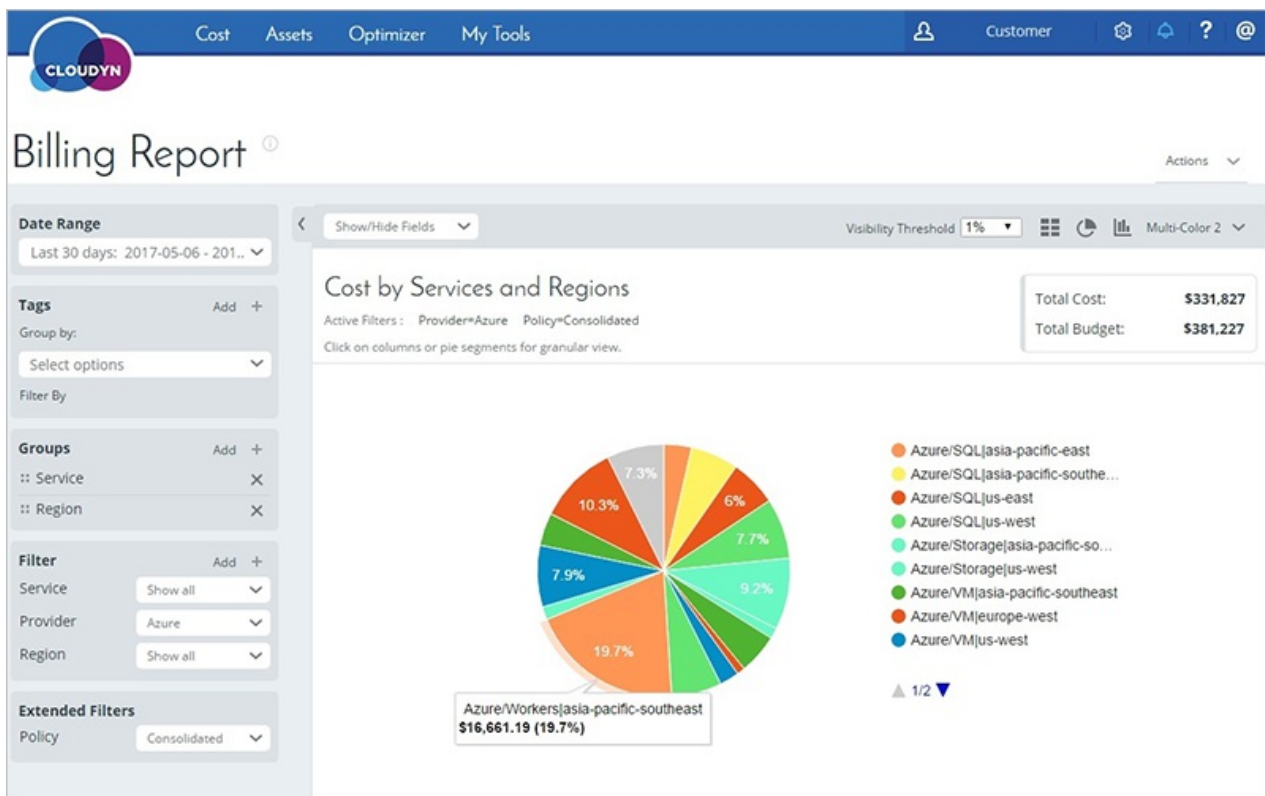
Manage multiple Partner Center accounts

A partner receives a Partner Center account for every CSP authorization in every region. For example, let's say that a partner is enrolled and authorized as a CSP direct partner in both the United States and the European Union. This partner has two different Partner Center accounts, isolated from each other.

Or, let's say that a partner wants to provide services based on global Azure datacenters, and also use [Azure Germany](#) (which required an additional CSP enrollment). This means that the customer also has two Partner Center accounts, global and national.

Some multinational partners, covering different regions, may have five or more Partner Center accounts, and they all are absolutely isolated from each other. There are different user accounts and credentials to sign in to Partner Center, different sets of customers, and separate invoices.

In this case, management of all CSP customers across different Partner Center accounts can be challenging. To solve this challenge and manage multiple Azure subscriptions of customers in different Partner Center accounts from a single panel, CSP direct partners and CSP indirect providers can use [Azure Cost Management by Cloudyn](#).



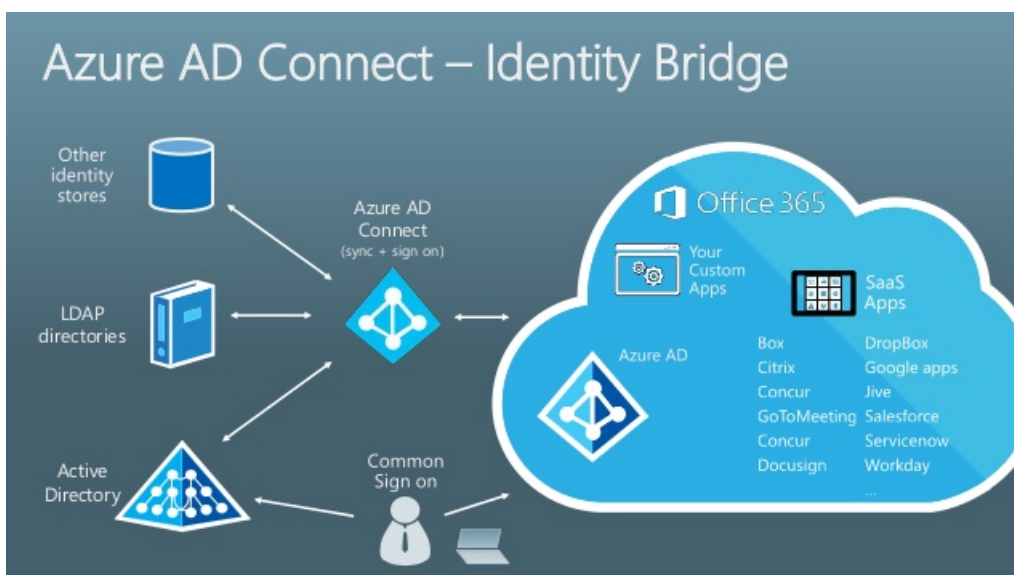
Also partners can use third-party cloud management panels that support Azure CSP to manage multiple Partner Center accounts from a single panel:

- [CloudCheckr](#)
- [CloudHealth](#)
- [RightScale](#)

Azure Active Directory and Partner Center

Partner Center uses Azure Active Directory (Azure AD) authentication to sign in. If you already use Azure AD in your organization (for example, for an internal Office 365 subscription), you can use the same user credentials and rights management capabilities in Partner Center.

Also, you can [connect](#) your on-premises Active Directory to Azure Active Directory by using Azure AD Connect. Then sign in to Partner Center by using corporate domain credentials and custom domain. With Azure AD, you can use [Active Directory Federation Services](#) and [multi-factor authentication capabilities](#) to strengthen the Partner Center authentication process.



Partner Center Integration Sandbox

Inside every Partner Center account for CSP direct partners or CSP indirect providers, you can create one [Partner Center Integration Sandbox](#) account.

The Partner Center Integration Sandbox account and regular Partner Center accounts are very similar, except that the sandbox is a "practice" space that does not touch any live customer data. It is meant for development and testing of new integration options and Partner Center functions, without tampering with your customers' data. Use of the sandbox account does not appear on your invoice.

A few other notes about the Partner Center Integration Sandbox account:

- The sandbox account and the primary Partner Center account act independently, and do not share admin accounts, user accounts, customers, orders, subscriptions, or other data.
- The sandbox account is [limited](#) to 75 customers, with 5 Azure subscriptions in each. Each subscription is limited to \$200 per month. It can't be extended.
- By policy, sandbox accounts are for internal testing purposes only.
- There is no way to migrate customers and Azure workloads from the sandbox to production. Don't use it for proof-of-concept or pilot environments that you want to continue using in production.

Although sandbox accounts have some significant limitations, they provide up to \$75,000 (75 x 5 x 200) of Azure, free of charge every month for internal testing purposes.

To create a sandbox account, from the Partner Center **Dashboard**, go to **Account Settings > Integration Sandbox**. If this option does not appear, it is because you lack admin permissions, or because you are already on an Integration Sandbox account. For more information about sandbox accounts, see the [Partner Center documentation](#).

Next steps

- [Create](#) your first customer in CSP, and deploy Azure services.
- [Learn](#) how customer support in the CSP model works.
- [Migrate](#) existing customers from on-premises and other channels to Azure CSP.

Azure partner shared services

10/3/2017 • 2 min to read • [Edit Online](#)

Azure Partner Shared Services is a new offer type for partners in the CSP program. It allows partners to purchase Azure subscriptions for their own use. This allows partners to use a uniform method for purchasing, tracking and managing Azure. Partners may now be able to consolidate their Azure licensing and reselling agreements with Microsoft. With Azure Partner Shared Services partners now have the same flexibility to use Azure subscriptions in CSP as they do in the Microsoft Enterprise Agreement and Web Direct programs. Partners in CSP will now be able to do things like:

- Host shared services and multi-tenant applications
- Deploy internal workloads and management solutions, that manage different customers with different Azure subscriptions
- Build development and test environments, that are not fitting into [Partner Center Integration Sandbox](#)

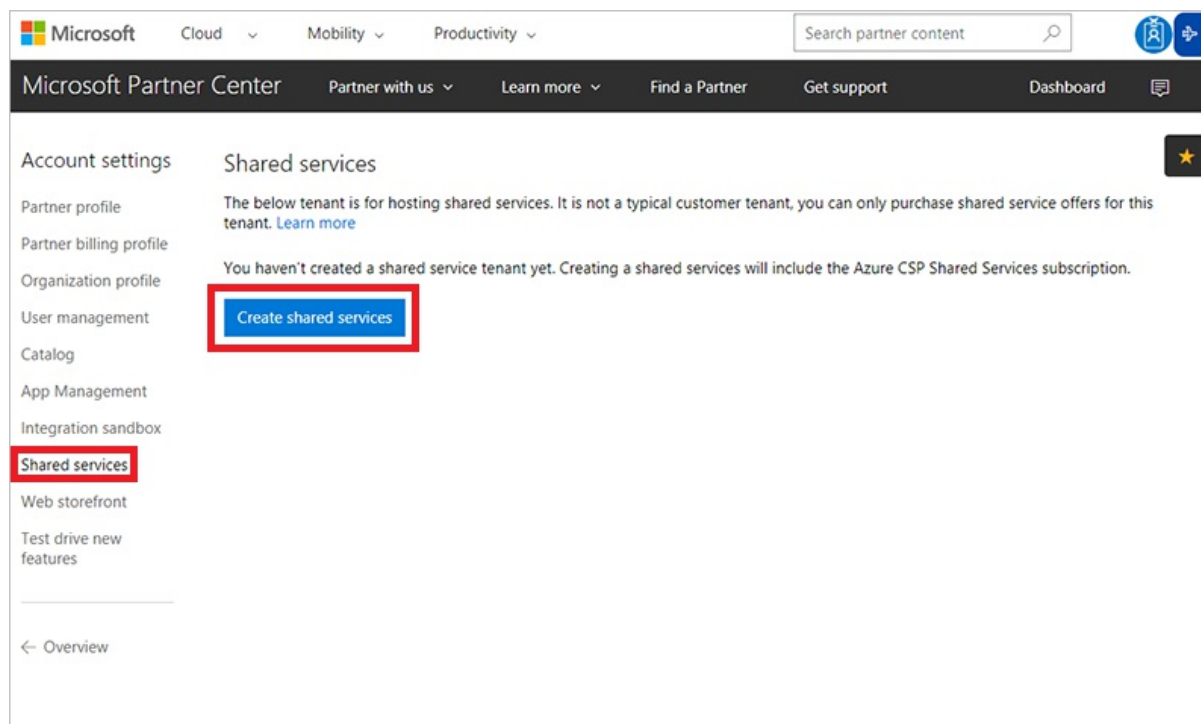
NOTE

Regular CSP discounts and incentives are not applied on Azure Partner Shared Services. To estimate the cost of solution, deployed inside shared services tenant, partners can use [Azure Pricing Calculator](#).

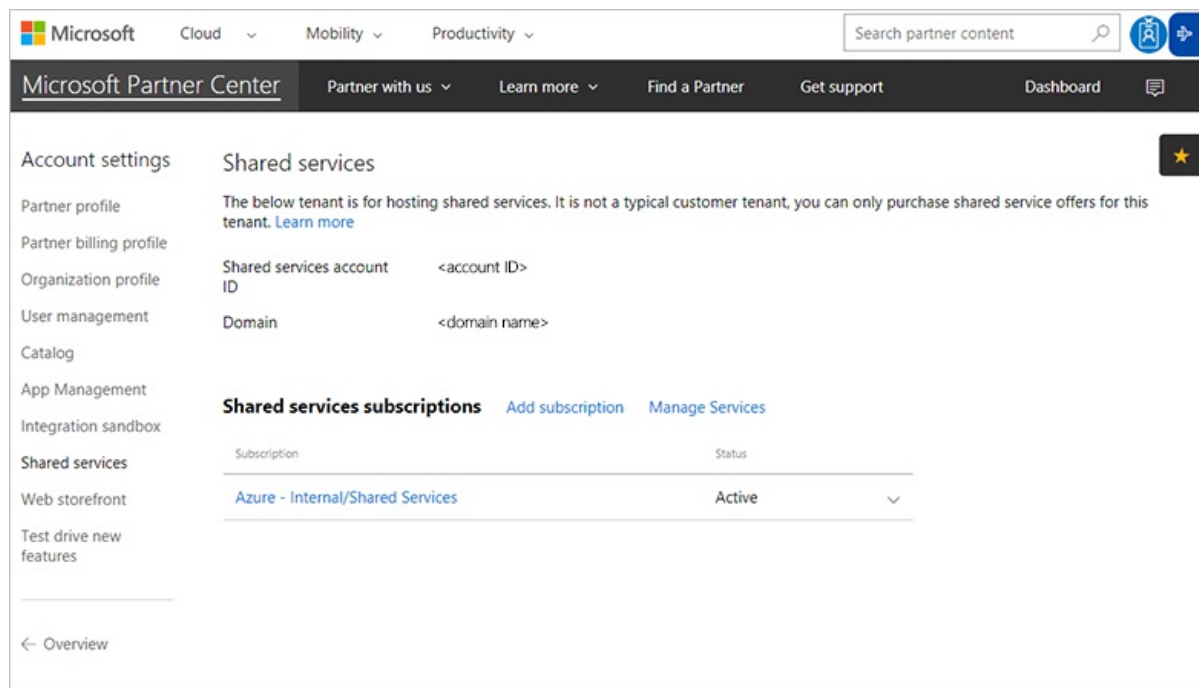
Create shared services tenant

To leverage this new capability a partner needs to create a shared services tenant that applies an Azure CSP Shared Services subscription.

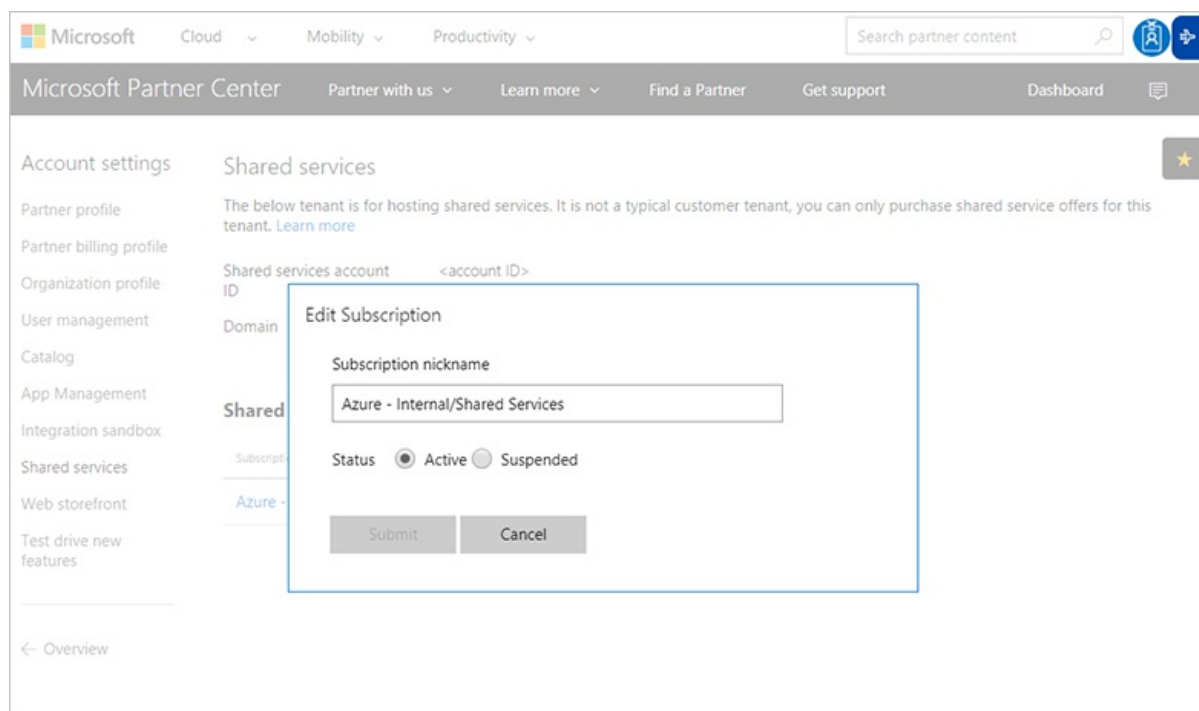
1. In Partner Center, go to the **Dashboard** menu > **Account settings** > **Shared services**. If a shared services tenant does not exist, the partner can click **Create shared services**.



2. This will create a shared services tenant and purchase the Azure CSP Shared Services subscription, to be used for shared resources and internal workload.



3. Partners can cancel the subscription after it is purchased like any other subscription, this is done in **Dashboard** menu > **Account settings** > **Shared services** and then selecting the **Azure - Internal/Shared Services subscription**, enabling the Partner to change the friendly name or cancel the subscription.



Shared services tenant specifics

- Azure – Internal/Shared Services offer does not share CSP pricing and are not eligible for discounts and incentives.
- A partner in CSP only needs one shared services tenant, multiple Azure – Internal/Shared Services offers can be purchased after the tenant is created.
- The Shared services tenant cannot be created without purchasing at least one subscription of Azure – Internal/Shared Services.
- A shared services tenant cannot be removed once created, however if a CSP doesn't want to use the tenant for shared services they can simply cancel the Azure – Internal/Shared Services subscription.
- Azure consumption through this offer would appear on the CSP invoice and the reconciliation file. It would be

included as part of Microsoft Azure line item in the invoice. While the granular consumption information will be accessible through the reconciliation file against the tenant that was created for this offer.

Next steps

- [Learn](#) about the available capabilities for Azure in Partner Center.
- [Migrate](#) existing customers from on-premises and other channels to Azure CSP.

Azure CSP management options

10/6/2017 • 2 min to read • [Edit Online](#)

The Azure Cloud Solution Provider (Azure CSP) model provides three ways to manage Azure CSP resources:

- Web UI: This is a manual approach where all steps are taken through a web browser.
- PowerShell: This is an automated approach that uses scripts.
- REST API or managed API: This is an integrated approach that requires developer skills.

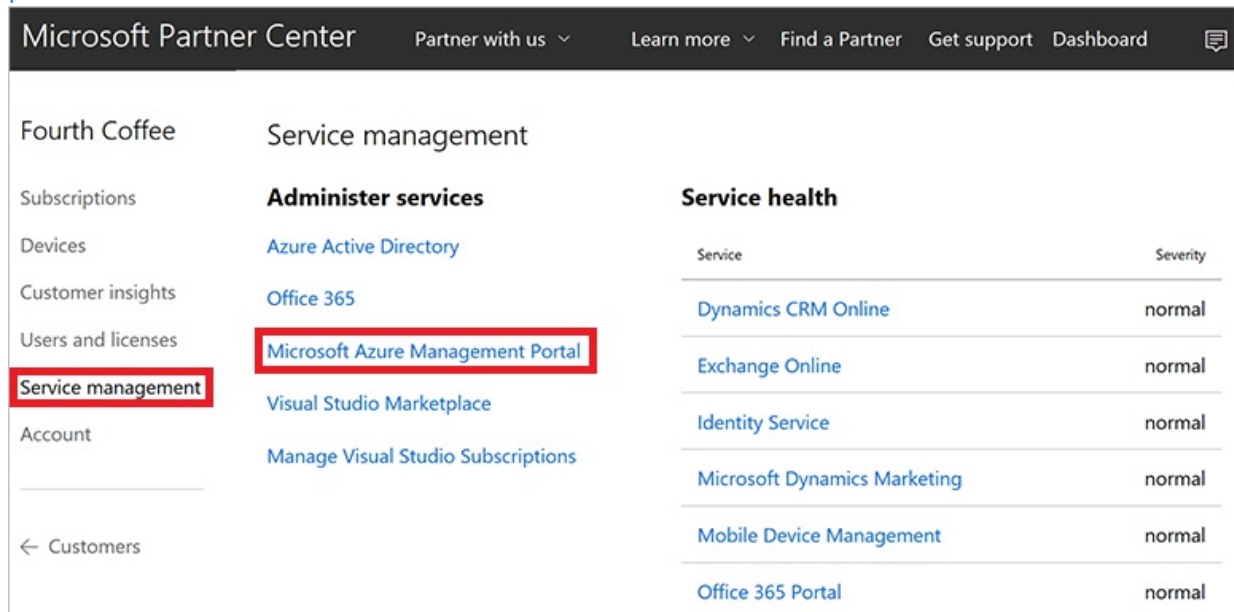
Web UI

The web UI approach is a good choice when you start your journey with Azure CSP and take all steps manually through a web browser. There are two portals that you use:

- [Partner Center portal](#): For more information about Partner Center capabilities for Azure management, read [Partner Center overview](#).
- [Azure portal](#): Use the Azure portal to manage Azure resources inside Azure CSP subscriptions.

To manage a customer's Azure CSP subscription in the Azure portal, you need to use the context of the customer's tenant. There are two ways to do that:

- Use the **Microsoft Azure Management Portal** link on the **Service Management** page in the [Partner Center portal](#).



- Use a link in the following format: <https://portal.azure.com/customertenantname.onmicrosoft.com>.

A partner user with an **AdminAgent** role in Partner Center can use both methods to access the Azure portal in the context of customer's tenant. There they can see and manage the Azure CSP subscriptions that were created inside the customer's tenant (tenant directory).

PowerShell

If you want to automate routine Azure CSP management operations, you can write PowerShell scripts by using the following two modules:

- [Partner Center PowerShell module](#)

- [Azure Resource Manager PowerShell module](#)

IMPORTANT

The Partner Center PowerShell module is an open source project that's not officially supported by Microsoft.

Partner Center PowerShell cmdlets provide customer management, subscription management, and billing capabilities that are similar to those in the web UI approach. Azure Resource Manager PowerShell cmdlets make it possible for you to manage Azure services inside Azure CSP subscriptions.

Make sure your app management is [already configured](#) to enable access to the Partner Center API.

```
# Install the Partner Center and Azure Resource Manager modules from the PowerShell Gallery
Install-Module PartnerCenterModule
Install-Module AzureRM

# Get a partner's user credentials.
$credentials = Get-Credential '<username@domain>'

# Authenticate in Partner Center and Azure Resource Manager.
Add-PCAuthentication -cspappID '<native app id GUID>' -cspDomain '<csp partner domain>' -credential $credentials
Login-AzureRmAccount -credential $credentials

#Find an Azure CSP customer and select their subscription.
$Customer = Get-PCCustomer -startswith '<company name>'
$Subscription = Get-PCSubscription -TenantId $Customer.id
Select-AzureRmSubscription -SubscriptionID $Subscription.id -TenantId $Customer.id
```

REST APIs and managed APIs

If you want to integrate Azure CSP capabilities with existing systems, you can use a [rich set of APIs in Partner Center and Azure](#):

- The Partner Center REST API and managed API scenarios overview is available on the [Partner Center scenarios page](#).
- Azure REST API documentation for different Azure components is available in the [Azure REST API reference](#).

In the following articles, we cover most common Azure CSP management scenarios by using all three approaches.

Next steps

- [Review](#) the list of management scenarios.
- [Review](#) the available APIs for Azure CSP integration.
- [Create](#) your first customer in Azure CSP and deploy Azure services.

Create an Azure CSP customer

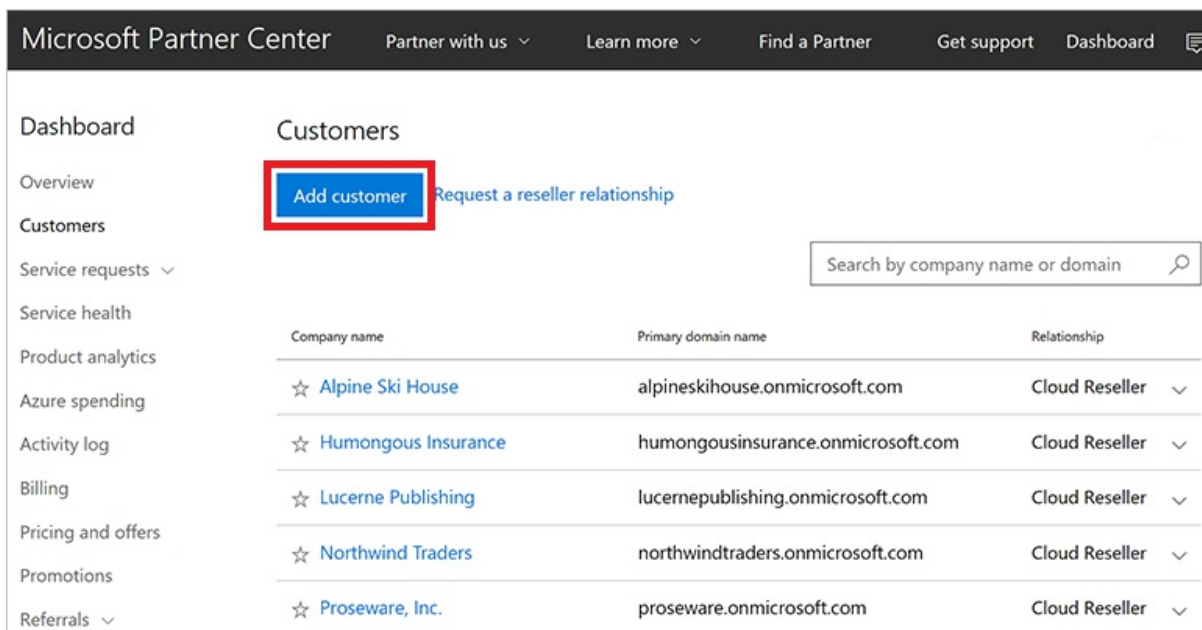
9/19/2017 • 5 min to read • [Edit Online](#)

Before you can access a customer's support or billing information or place orders for a customer, you must create the customer in the system. When you create a customer, you also create:

- An Azure Active Directory (Azure AD) tenant object for the customer.
- A relationship between the reseller and customer, which is used for delegated administrator privileges.
- A username and password for signing in as an administrator for the customer.

Web UI

1. On the Microsoft Partner Center dashboard, in the left pane, select **Customers**, and then select **Add customer**.



The screenshot shows the Microsoft Partner Center dashboard. The top navigation bar includes 'Microsoft Partner Center', 'Partner with us', 'Learn more', 'Find a Partner', 'Get support', and 'Dashboard'. The left sidebar lists various sections: Dashboard, Overview, Customers, Service requests, Service health, Product analytics, Azure spending, Activity log, Billing, Pricing and offers, Promotions, and Referrals. The 'Customers' section is selected, and the 'Add customer' button is highlighted with a red box. Below the button, there is a search bar labeled 'Search by company name or domain'. A table lists existing customers with columns for Company name, Primary domain name, and Relationship.

Company name	Primary domain name	Relationship
☆ Alpine Ski House	alpineskihouse.onmicrosoft.com	Cloud Reseller
☆ Humongous Insurance	humongousinsurance.onmicrosoft.com	Cloud Reseller
☆ Lucerne Publishing	lucernepublishing.onmicrosoft.com	Cloud Reseller
☆ Northwind Traders	northwindtraders.onmicrosoft.com	Cloud Reseller
☆ Proseware, Inc.	proseware.onmicrosoft.com	Cloud Reseller

2. On the **Account info** form, enter the customer's account information.

NOTE

Microsoft Partner Center

Partner with us

Learn more

Find a Partner

Get support

Dashboard

New customer

Account info

Subscriptions

Review

Confirmation

← Customers

New subscription

Top offers

Choose offers to place at the top of the Catalog so that sales agents can quickly find the ones that matter most to your company.

Catalog

EnterpriseSmall businessGovernment

☐ Azure Active Directory Basic

☐ Azure Active Directory Premium P1

☐ Azure Active Directory Premium P2

☐ Azure Information Protection Plan 1

☐ Azure Information Protection Premium P2

☐ Dyn365 Cloud Add-on From AX Enterprise to Retail

...

☐ Exchange Online Protection

☐ Intune

☐ Intune Device

☒ Microsoft AzureUsage-based

☐ Microsoft Azure Multi-Factor Authentication

☐ Microsoft Cloud App Security

...

☐ Windows 10 Enterprise E3

☐ Windows 10 Enterprise E5

☐ Yammer Enterprise

Next: Review

Cancel

Microsoft Azure

Azure Cloud Solution Provider offer for Partner and Resellers

Offer ID: MS-AZR-0145P

5. Select **Next: Review**.

6. Review the customer and offer details. To make changes, select the corresponding **Update** link. Otherwise, select **Submit**.

Microsoft Partner Center
Partner with us
Learn more
Find a Partner
Get support
Dashboard

New customer
Account info
Subscriptions
Review
Confirmation

Review
Select **Submit** to confirm the information below. Select **Cancel** to make changes.

Account info
Update

Company
Fourth Coffee
One Microsoft Way
Redmond WA 98052
fourthcoffeecsp.onmicrosoft.com

Primary contact info
Marshall Levi
marshall@fourthcoffee.com
1234567890

Subscriptions
Update

Enterprise:
Microsoft Azure
Quantity
Usage-based
Promotions Applied

Submit
Cancel

- In the confirmation window, note the **Admin user account** and **Password** information. The information will not be viewable after you leave this page, so copy and store it to send to the customer.

Microsoft Partner Center
Partner with us
Learn more
Find a Partner
Get support
Dashboard

New customer
Account info
Subscriptions
Review
Confirmation

Confirmation
We have received your order. It may take a few minutes to process. To see this order, view this customer's subscriptions. Be sure to copy the following customer and subscription information for your records.

Setup info

Microsoft ID
<tenant ID>
Admin user account
admin@fourthcoffeecsp.onmicrosoft.com
Password
<password>

Please copy and store this password. After you leave this page, you can't view this password again.

Account info

Company
Fourth Coffee
One Microsoft Way
Redmond WA 98052
fourthcoffeecsp.onmicrosoft.com

Primary contact info
Marshall Levi
marshall@fourthcoffee.com
1234567890

Subscriptions

Enterprise:
Microsoft Azure
Subscription ID
Usage-based
<subscription ID>

Done

IMPORTANT

By default, the customer administrator does *not* have access to this Azure subscription and cannot manage it by using the [Azure portal](#). The CSP model was designed for a variety of scenarios, and in some of them (for example, Managed Cloud Services) the customer doesn't need to access the Azure portal. The CSP partner manages the customer's resources instead. If you want to grant the customer access to the Azure CSP subscription, an [additional step](#) is required.

8. Select **Done**.
9. On the Partner Center dashboard, select **Customers**, select the customer that you just created, select **Service management**, and then select **Microsoft Azure Management Portal**.

The screenshot shows the Microsoft Partner Center interface. The top navigation bar includes 'Partner with us', 'Learn more', 'Find a Partner', 'Get support', and 'Dashboard'. The left sidebar lists various sections: 'Subscriptions', 'Devices', 'Customer insights', 'Users and licenses', 'Service management' (highlighted with a red box), and 'Account'. The main content area is titled 'Service management' and is divided into two columns. The left column, 'Administer services', lists 'Azure Active Directory', 'Office 365', 'Microsoft Azure Management Portal' (highlighted with a red box), 'Visual Studio Marketplace', and 'Manage Visual Studio Subscriptions'. The right column, 'Service health', displays a table of services and their health status.

Service	Severity
Dynamics CRM Online	normal
Exchange Online	normal
Identity Service	normal
Microsoft Dynamics Marketing	normal
Mobile Device Management	normal
Office 365 Portal	normal

The Azure portal opens.

10. To add the customer as an owner of the Azure subscription, see [Assign permissions to an Azure CSP subscription](#).

The screenshot shows the Microsoft Azure portal dashboard. The top navigation bar includes 'Dashboard', 'New dashboard', 'Edit dashboard', 'Share', 'Fullscreen', 'Clone', and 'Delete'. The left sidebar lists various resource types: 'New', 'Dashboard', 'All resources', 'Resource groups', 'App Services', 'Function Apps', 'SQL databases', 'Azure Cosmos DB', 'Virtual machines', 'Load balancers', 'Storage accounts', 'Virtual networks', and 'Azure Active Directory'. The main content area is titled 'Dashboard' and displays 'All resources ALL SUBSCRIPTIONS'. It shows a message 'No resources to display' with a 'Create resources' button. The right sidebar contains 'Quickstart tutorials' for 'Windows Virtual Machines', 'Linux Virtual Machines', 'App Service', 'Functions', and 'SQL Database'.

NOTE

One CSP customer account can have multiple Azure CSP subscriptions. By default, all subscriptions are named *Microsoft Azure*. To better distinguish them, rename them in Partner Center to something more descriptive (for example, *Production environment* or *HR System*).

PowerShell

Use the following command to create a new customer with PowerShell. To learn how to configure the PowerShell environment for Azure CSP, see [Azure CSP management options](#).

```
$newDefaultAddress = New-PCCustomerDefaultAddress -Country '<country code>' -Region '<region>' -City '<city>' -State '<state>' -
AddressLine1 '<address 1>' -PostalCode '<postal code>' -FirstName '<first name>' -LastName '<last name>' -PhoneNumber '<phone number>'

$newBillingProfile = New-PCCustomerBillingProfile -Email '<email>' -Culture '<ex: en.us>' -Language '<ex: en>' -CompanyName '<company name>' -
DefaultAddress $newDefaultAddress

$newCompanyProfile = New-PCCustomerCompanyProfile -Domain '<company name>.onmicrosoft.com'

$newCustomer = New-PCCustomer -BillingProfile $newBillingProfile -CompanyProfile $newCompanyProfile
```

C#

To add a customer, do the following:

1. Instantiate a new **Customer** object.
2. Fill in the **BillingProfile** and **CompanyProfile** information.
3. Add the object to your **IAggregatePartners.Customers** collection by calling **Create()** or **CreateAsync()**.

```
// IAggregatePartner partnerOperations;
var partnerOperations = this.Context.UserPartnerOperations;

var customerToCreate = new Customer()
{
    CompanyProfile = new CustomerCompanyProfile()
    {
        Domain = string.Format(CultureInfo.InvariantCulture,
            "SampleApplication {0}.{1}",
            new Random().Next(),
            this.Context.Configuration.Scenario.CustomerDomainSuffix)
    },
    BillingProfile = new CustomerBillingProfile()
    {
        Culture = "en-us",
        Email = "marshall@fourthcoffee.com",
        Language = "en",
        CompanyName = "Fourth Coffee" + new Random().Next(),
        DefaultAddress = new Address()
        {
            FirstName = "Marshall",
            LastName = "Levi",
            AddressLine1 = "One Microsoft Way",
            City = "Redmond",
            State = "WA",

            Country = "US",
            PostalCode = "98052",
            PhoneNumber = "1234567890"
        }
    }
};

var newCustomer = partnerOperations.Customers.Create(customerToCreate);
```

REST API

The following is a guide to formatting a request to add a customer by using the REST API.

Request

Request syntax

METHOD	REQUEST URI
POST	{baseURI}/v1/customers HTTP/1.1

Request headers

- This API will not produce different results if it is called multiple times.
- Request ID and correlation ID are required.

Request body

The request body has the following *required* properties:

NAME	TYPE	DESCRIPTION
BillingProfile	object	The customer's billing profile information.

NAME	TYPE	DESCRIPTION
CompanyProfile	object	The customer's company profile information.

Billing profile

The CustomerBillingProfile resources needed to create a new customer have the following minimum *required* fields:

NAME	TYPE	DESCRIPTION
email	string	The customer's email address.
culture	string	The customer's preferred culture for communication and currency, such as "en-US."
language	string	The default language. Two-character language codes (en, fr, and so on) are supported.
company_name	string	The registered company or organization name.
default_address	address	The registered address of the customer's company or organization.

Company profile

The CustomerCompanyProfile resource needed to create a new customer has the following minimum *required* field:

NAME	TYPE	DESCRIPTION
domain	string	The customer's domain name, such as <i>contoso.onmicrosoft.com</i> .

Request example

POST https://api.partnercenter.microsoft.com/v1/customers HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: 94e4e214-6b06-4fb7-96d1-94d559f9b47f
MS-CorrelationId: ab993325-1605-4cf4-bac4-fb584142a31b
X-Locale: en-US
Content-Type: application/json
Host: api.partnercenter.microsoft.com
Content-Length: 789
Expect: 100-continue
Connection: Keep-Alive

```
{
  "Id": null,
  "CommerceId": null,
  "CompanyProfile": {
    "TenantId": null,
    "Domain": "SampleApplication1047419322.onmicrosoft.com",
    "CompanyName": null,
    "Attributes": {
      "ObjectType": "CustomerCompanyProfile"
    }
  },
  "BillingProfile": {
    "Id": null,
    "FirstName": null,
    "LastName": null,
    "Email": "marshall@fourthcoffee.com",
    "Culture": "en-us",
    "Language": "en",
    "CompanyName": "Fourth Coffee",
    "DefaultAddress": {
      "Country": "US",
      "Region": null,
      "City": "Redmond",
      "State": "WA",
      "AddressLine1": "One Microsoft Way",
      "AddressLine2": null,
      "PostalCode": "98052",
      "FirstName": "Marshall",
      "LastName": "Levi",
      "PhoneNumber": "1234567890"
    },
    "Attributes": {
      "ObjectType": "CustomerBillingProfile"
    }
  },
  "RelationshipToPartner": "none",
  "AllowDelegatedAccess": null,
  "UserCredentials": null,
  "CustomDomains": null,
  "AssociatedPartnerId": null,
  "Attributes": {
    "ObjectType": "Customer"
  }
}
```

Response

If the request is successful, this API returns a customer resource for the new customer. Save the customer ID and Azure AD details for future use, such as account management.

Response example

HTTP/1.1 201 Created
Content-Length: 834
Content-Type: application/json; charset=utf-8
MS-CorrelationId: ab993325-1605-4cf4-bac4-fb584142a31b
MS-RequestId: 94e4e214-6b06-4fb7-96d1-94d559f9b47f
MS-CV: ObwhuhD2tUKJoM+Z.0
MS-ServerId: 202010223
Date: Tue, 14 Feb 2017 20:06:02 GMT

```
{
  "id": "dfd8cc0a-c592-468c-8461-869a38d24738",
  "commerceId": "0a4ce58a-6f96-4273-8035-d9c7d31b9ba4",
  "companyProfile": {
    "tenantId": "dfd8cc0a-c572-468c-8461-869a38d24734",
    "domain": "SampleApplication1047419322.onmicrosoft.com",
    "attributes": {
      "objectType": "CustomerCompanyProfile"
    }
  },
  "billingProfile": {
    "id": "d17c0275-da92-5c33-9032-782efd0b69b",
    "email": "marshall@fourthcoffee.com",
    "culture": "en-us",
    "language": "en",
    "companyName": "Fourth Coffee",
    "defaultAddress": {
      "country": "US",
      "city": "Redmond",
      "state": "WA",
      "addressLine1": "One Microsoft Way",
      "postalCode": "98052",
      "firstName": "Marshall",
      "lastName": "Levi",
      "phoneNumber": "1234567890"
    },
    "attributes": {
      "etag": "1234567890",
      "objectType": "CustomerBillingProfile"
    }
  },
  "relationshipToPartner": "none",
  "userCredentials": {
    "userName": "admin",
    "password": "P@ssword12345"
  },
  "attributes": {
    "objectType": "Customer"
  }
}
```

Next steps

- [Learn how to add an existing Azure customer to your Partner Center account](#)
- [Learn how to assign Azure CSP subscription permissions to the customer](#)

Add an existing customer to Azure Cloud Solution Provider

9/19/2017 • 2 min to read • [Edit Online](#)

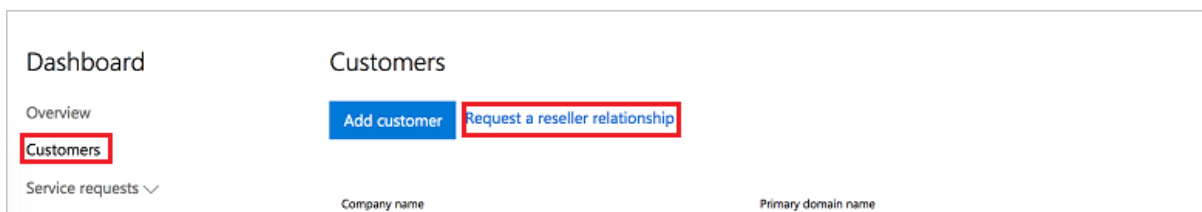
Learn how to add an existing Microsoft Partner Center customer account, and then associate it with an active Azure subscription. You can do this in Partner Center and the Azure portal, or use one of the Partner Center APIs. To add an existing Partner Center customer account, you must have an existing Partner Center account and an active Azure subscription.

In this article, we describe how to add an existing Partner Center account for Azure Cloud Solution Provider (Azure CSP) integration by using one of the following options:

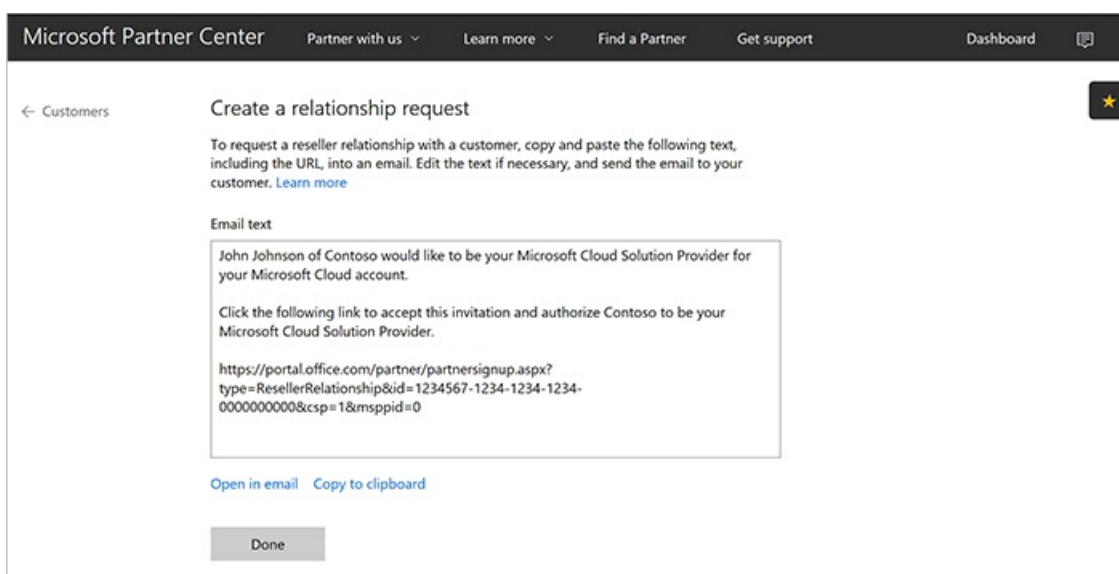
- Partner Center web UI
- PowerShell
- C#
- REST API

Partner Center web UI

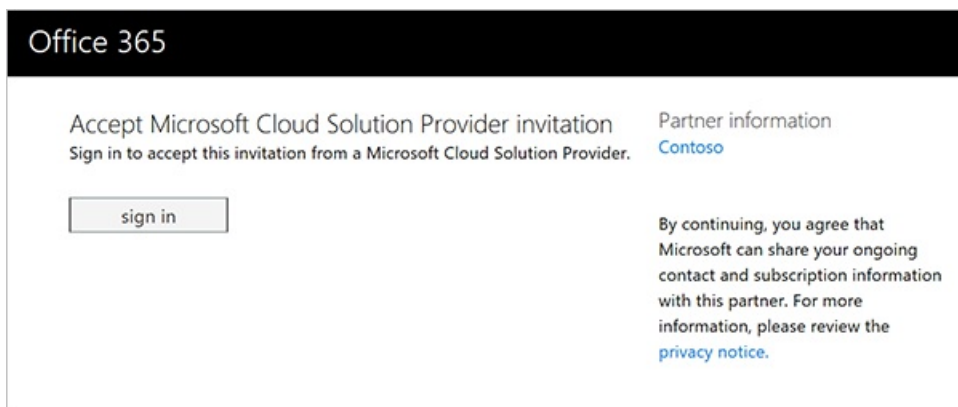
1. In Partner Center, on the **Dashboard** menu, select **Customers** > **Request a reseller relationship**.



2. On the **Create a relationship request** page, copy the email text that appears. You can edit or delete the first two paragraphs of the email text, but keep the link intact. The link in the email text applies specifically to your reseller account. The link doesn't include any customer information. You can reuse the same link each time you request a reseller relationship.



3. Send the relationship request email to your customer. To accept the relationship request, the customer's global admin selects the invitation link, and then signs in.



4. On the **Partner Relationships** page, the customer can review their partners for their Azure services. This includes which partners who have delegated admin rights for each service.

NOTE

In the multi-channel model, for each service, a customer can have only one partner who has delegated admin rights for Azure CSP transactions. The customer might also have granted rights to other partners who don't participate in the Azure CSP program.

5. The invited customer appears in the partner's customer list in Partner Center. The partner can then provision and manage the Azure CSP subscriptions for the customer.

Customers have the option to remove delegated admin rights from any or all partners (including those in the Azure CSP program). Customers can do this in their own admin portal. The customer account and commercial relationship with the partner still applies, unless it is renegotiated by the customer and the partner. The partner is still responsible for providing support to the customer. A partner also must still adhere to the terms of the Microsoft Cloud Reseller Agreement, even if a customer removes delegated admin rights from the partner. In this scenario, if the customer requires support, the partner must contact Microsoft Support to submit a service request on behalf of the customer.

To re-establish delegated admin rights that the customer has revoked or removed, repeat the process of requesting a relationship.

PowerShell

To request a relationship by using PowerShell, get the invitation URL as described in steps 1 and 2 in the [Partner Center web UI](#) section. Then, in PowerShell, run the following command:

```
$inviteURL = (New-PCRelationshipRequest).url
```

C#

1. Use your **IAggregatePartner.Customers** collection.
2. Call the **RelationshipRequest** property.
3. Call the **Get()** or the **GetAsync()** method.

```
// IAggregatePartner partnerOperations;  
  
var customerRelationshipRequest = partnerOperations.Customers.RelationshipRequest.Get();
```

REST API

Request

Request syntax

METHOD	REQUEST URI
GET	{baseUrl}/v1/customers/relationshiprequests HTTP/1.1

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers/relationshiprequests HTTP/1.1
```

```
Authorization: Bearer <token>
```

```
Accept: application/json
```

```
MS-RequestId: f281903b-7b57-4883-8bc1-5caca1b22a37
```

```
MS-CorrelationId: dd048ef2-d737-4b11-a885-e1eab8fc94ba
```

```
Connection: Keep-Alive
```

Response

If the request is successful, this method returns a **CustomerRelationshipRequest** object in the response.

Response example

```
HTTP/1.1 200 OK
```

```
Content-Length: 204
```

```
Content-Type: application/json; charset=utf-8
```

```
MS-CorrelationId: dd048ef2-d737-4b11-a885-e1eab8fc94ba
```

```
MS-RequestId: f281903b-7b57-4883-8bc1-5caca1b22a37
```

```
Date: Mon, 21 Mar 2016 05:39:51 GMT
```

```
{
  "url": "https://portal.office.com/partner/partnersignup.aspx?type=ResellerRelationship&id=3b33e682-00c3-41ee-9dd2-a548adf56438&csp=1&msppid=0",
  "attributes": {
    "objectType": "CustomerRelationshipRequest"
  }
}
```

Next steps

- Learn how to [assign Azure CSP subscription permissions to the customer](#).
- Learn how to [create a new Azure CSP customer in Partner Center](#).

Assign and manage permissions within an Azure subscription

9/19/2017 • 5 min to read • [Edit Online](#)

When customers are first created in Partner Center, they do not have access to the Azure subscription by default. The Azure Cloud Solution Provider (CSP) model was designed for various scenarios, and in some of them (for example, managed cloud services), customers don't need to access the Azure portal. The CSP partner manages customer resources instead.

If you want to grant a customer access to an Azure CSP subscription, an additional step is required. Over time, you might also want to add users with varying levels of permissions to an Azure subscription.

TIP

If you grant a customer access to the Azure portal, learn about [Azure Resource Manager Policies](#). You can use the policies to limit which Azure services and regions are available to CSP customers in the Azure portal.

This guide covers how to grant customer access from the Azure portal, PowerShell, and the REST API.

Web UI

1. Open the [Azure portal](#) in either of the following ways:

- Go to <https://portal.azure.<customer tenant name>.com/>.
- From the Partner Center dashboard, go to **Customers** > **<customer name>** > **Service Management** > **Microsoft Azure Management Portal**.

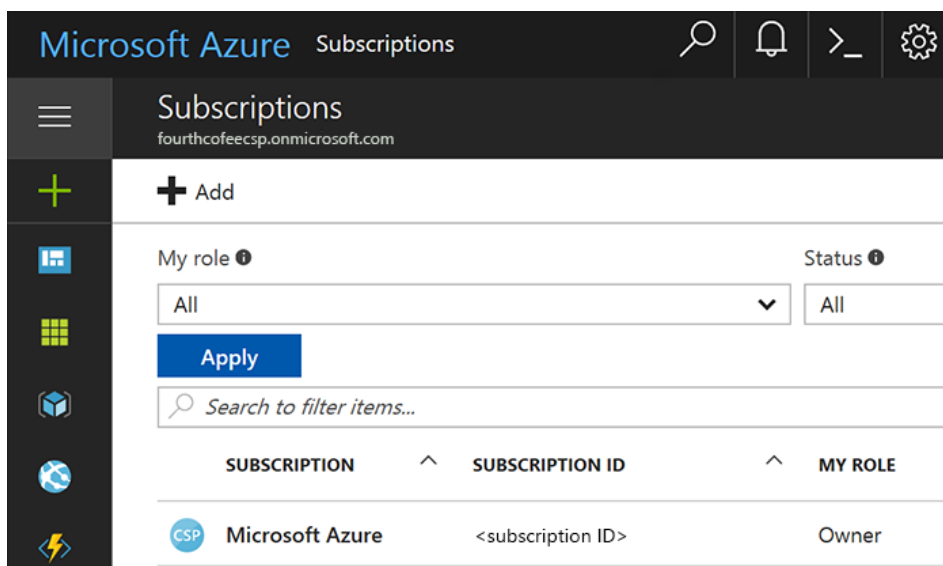
2. Sign in.

Add access permissions

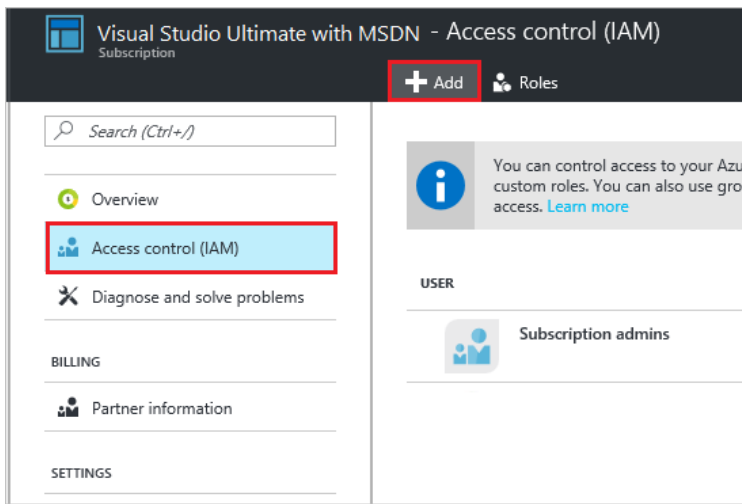
1. In the left pane, select **Subscriptions**, and then select the subscription you want the admin to access.

NOTE

To access the **Subscriptions** button, you might need to select **More services** at the bottom of the pane.



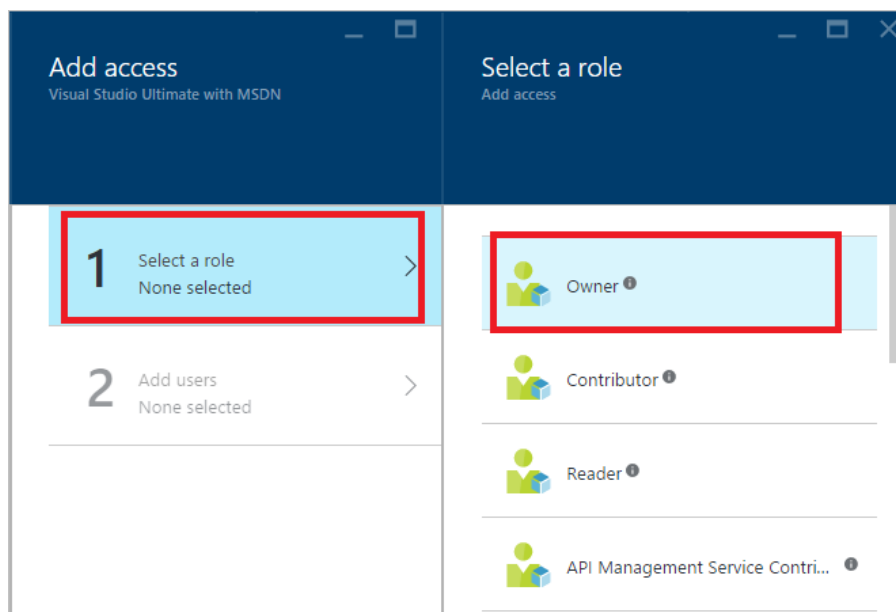
2. On the **Subscription** blade, select **Access control (IAM)**, and then select the **Add** button.



3. On the **Add access** blade, select **Select a role**, and then select the role that you want to assign.

- **Owner:** Can manage all aspects of a subscription, including giving access to other users via this process. If this is a newly created customer, you probably want to give the customer admin *Owner* permissions.
- **Contributor:** Can manage all aspects of a subscription *except* giving access to other users.
- **Reader:** Can view all aspects of a subscription, but cannot make any changes.

The roles listed here are the most basic that you can assign to users in the Azure portal. For a more detailed list, see [Built-in roles for Azure role-based access control](#).



4. In the **User** field, type the email address of the user you want to add. The user appears below the search box. Select the user, and then select **Submit**. The user now has access to the subscription.

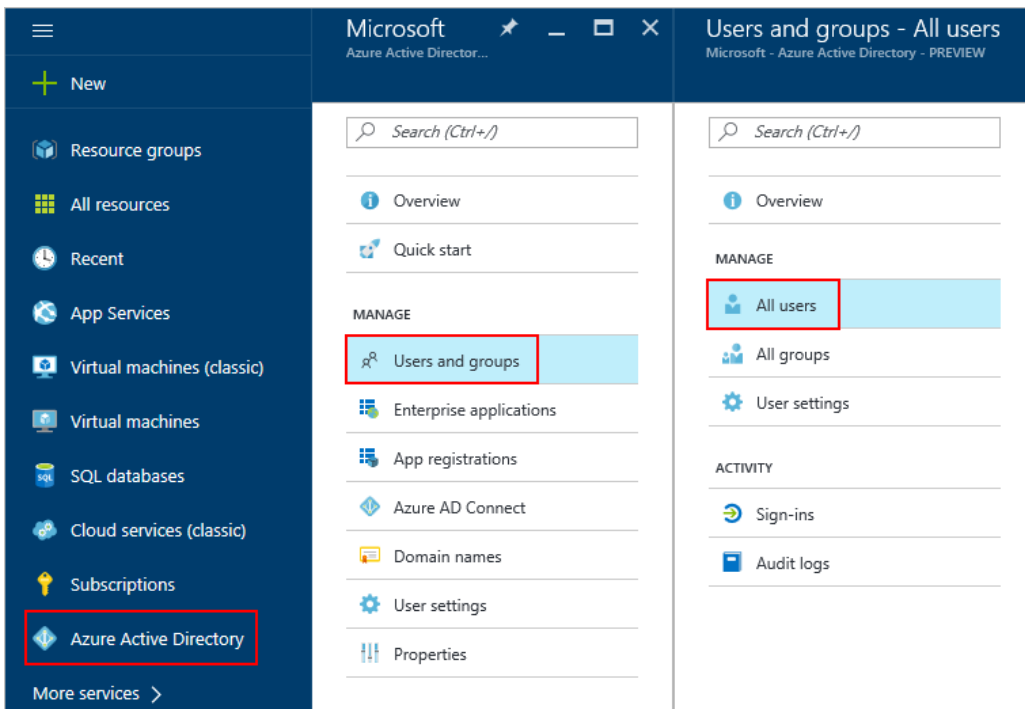
View access permissions

You can also check a user's access permissions through the Azure portal.

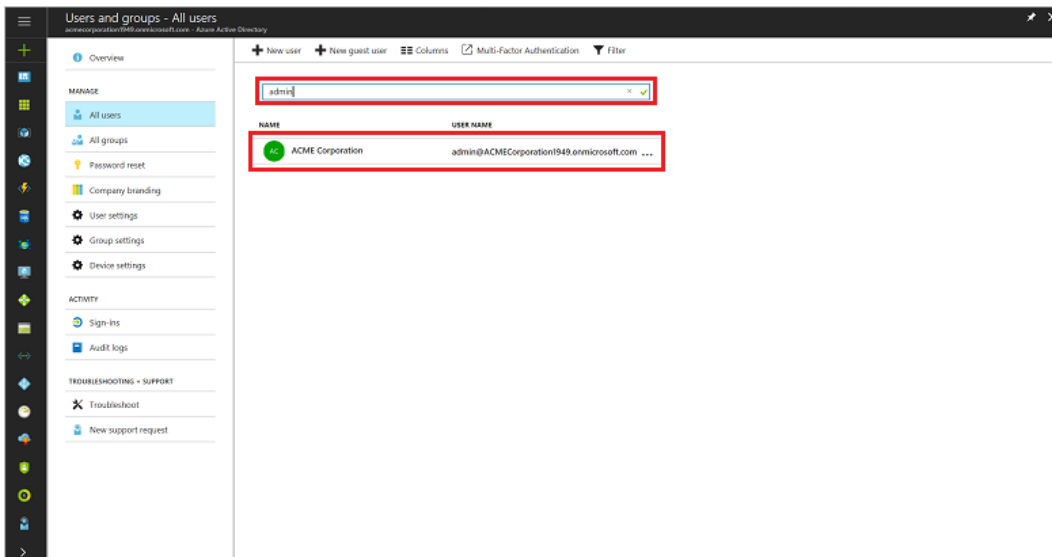
1. In the left pane, select **Azure Active Directory** > **Users and groups** > **All users**.

NOTE

To access the **Azure Active Directory** button, you might need to select **More services** at the bottom of the pane.

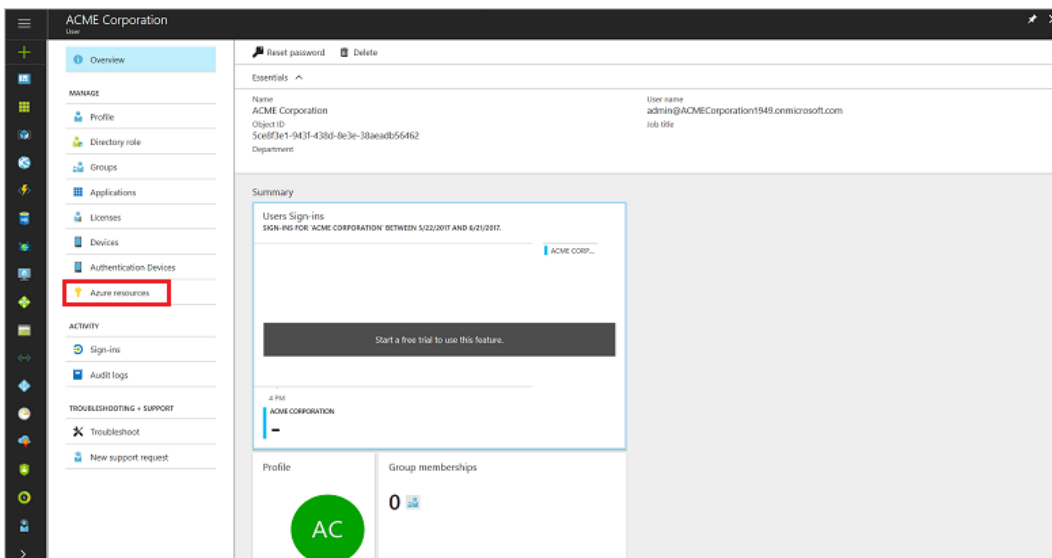


2. In the search box, search for the user by name or username and then, in the results list, select the user.



A new user blade opens.

3. In the left pane, select **Azure resources**.

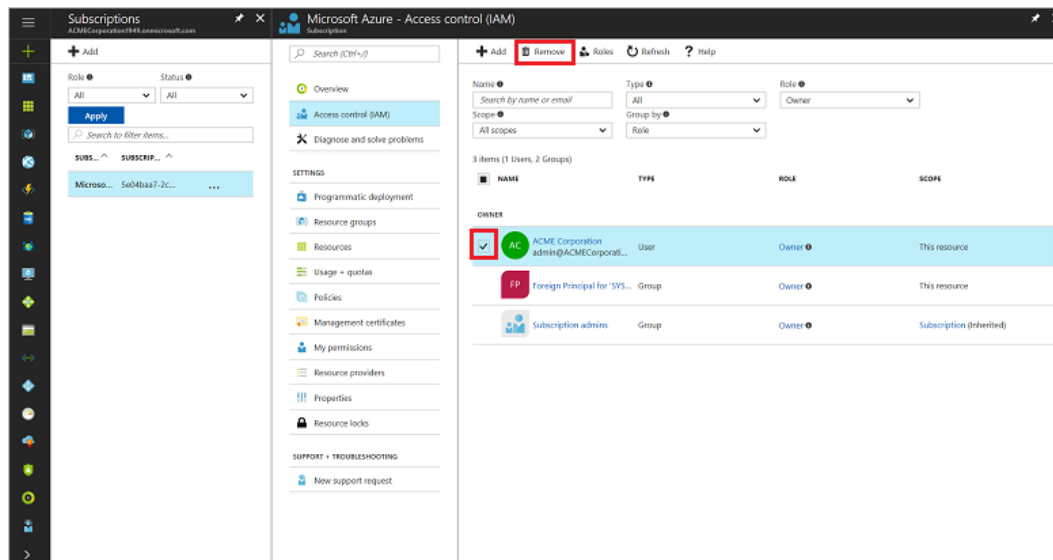


The user's access permissions are displayed.

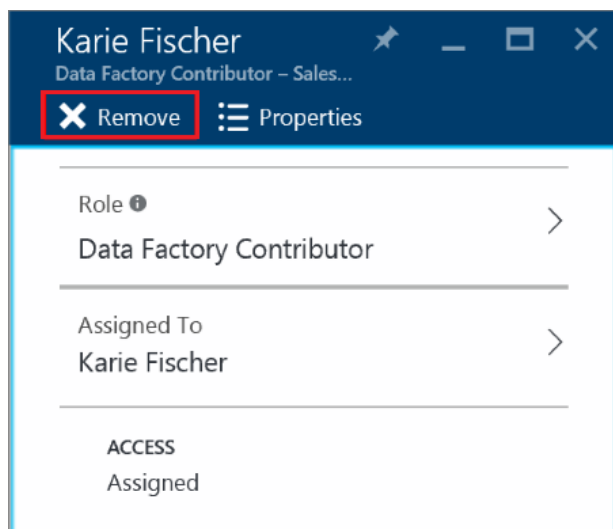
Delete access permissions

You can easily remove a user's subscription access in either of two ways:

- In the **Access control** window:
 1. To get to the **Access control (IAM)** window, follow steps 1 and 2 in the [Add access permissions](#) section.
 2. Select the check box next to the user's name, and then select **Remove**.



- On the user's page:
 1. To view the list of a user's access permissions, follow the steps in the [View access permissions](#) section.
 2. Select the subscription that you want to remove, and then select **Remove**.



PowerShell

Grant access permissions

To grant a user access to a subscription by using PowerShell, run the following commands:

```
# Get subscription ID (if necessary)
C:\PS>Get-AzureSubscription -SubscriptionName "MyProdSubscription"

# Assign role to subscription
New-AzureRmRoleAssignment -ObjectId <application id> -RoleDefinitionName <role name> -Scope <subscription id>
```

View access permissions

To view the access assignments for a specific subscription, run the following commands:

```
Get-AzureRmRoleAssignment -SubscriptionName "MyProdSubscription" | FL DisplayName, RoleDefinitionName, Scope
```

Remove access permissions

To remove a user's subscription access, run the following commands:

```
Remove-AzureRmRoleAssignment -ObjectId <object id> -RoleDefinitionName <role name> -Scope <scope such as subscription id>
```

REST API

Grant access permissions

To create a role assignment, you must have access to the `Microsoft.Authorization/roleAssignments/write` operation. Of the built-in roles, only *Owner* and *User Access Administrator* are granted access to this operation.

Request syntax

METHOD	REQUEST URI
PUT	https://management.azure.com/subscriptions/{subscription-id}/providers/Microsoft.Authorization/roleAssignments/{role-assignment-id}?api-version={api-version}

Request parameters

These are the required parameters for the request:

NAME	TYPE	DESCRIPTION
subscription-id	string	The ID number for the subscription.
role-assignment-id	string	The ID number for the new role assignment.
api-version	string	The current API version that you are using.

Request body

The request body should look like this:

```
{
  "properties": {
    "roleDefinitionId": "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/resourceGroups/Network/providers/Microsoft.Network/virtualNetworks/EASTUS-VNET-01/subnets/Devices-Engineering-ProjectRND/providers/Microsoft.Authorization/roleDefinitions/9980e02c-c2be-4d73-94e8-173b1dc7cf3c",
    "principalId": "5ac84765-1c8c-4994-94b2-629461bd191b"
  }
}
```

NAME	TYPE	DESCRIPTION
roleDefinitionId	String	The identifier of the role. The format is <code>{scope}/providers/Microsoft.Authorization/roleDefinition-id-guid</code> .
principalId	String	The objectId of the Azure Active Directory (Azure AD) principal (user, group, or service to which the role is assigned).

Response example

```
{
  "properties": {
    "roleDefinitionId": "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/providers/Microsoft.Authorization/roleDefinitions/9980e02c-c2be-4d73-94e8-173b1dc7cf3c",
    "principalId": "5ac84765-1c8c-4994-94b2-629461bd191b",
    "scope": "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/resourceGroups/Network/providers/Microsoft.Network/virtualNetworks/EASTUS-VNET-01/subnets/Devices-Engineering-ProjectRND",
    "createdOn": "2015-12-16T00:27:19.6447515Z",
    "updatedOn": "2015-12-16T00:27:19.6447515Z",
    "createdBy": null,
    "updatedBy": "877f0ab8-9c5f-420b-bf88-a1c6c7e2643e"
  },
  "id": "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/resourceGroups/Network/providers/Microsoft.Network/virtualNetworks/EASTUS-VNET-01/subnets/Devices-Engineering-ProjectRND/providers/Microsoft.Authorization/roleAssignments/2e9e86c8-0e91-4958-b21f-20f51f27bab2",
  "type": "Microsoft.Authorization/roleAssignments",
  "name": "2e9e86c8-0e91-4958-b21f-20f51f27bab2"
}
```

View access permissions

To list roles, you must have access to the `Microsoft.Authorization/roleDefinitions/read` operation at the scope. All the built-in roles are granted access to this operation.

Request syntax

METHOD	REQUEST URI
GET	https://management.azure.com/subscriptions/{subscription-id}/providers/Microsoft.Authorization/roleAssignments/{role-assignment-id}?api-version={api-version}&{\$filter={filter}}

Request parameters

These are the required parameters for the request:

NAME	TYPE	DESCRIPTION
subscription-id	string	The ID number for the subscription.
role-assignment-id	string	The ID number for the new role assignment.
api-version	string	The current API version that you are using.
filter	string	The specific conditions, such as viewing permissions for any child resources in that subscription with <code>atScopeAndBelow()</code> , or searching for a role for a specific user or group, such as <code>principalId%20eq%20'{objectId of user, group, or service principal}'</code> .

Response example

```
{
  "value": [
    {
      "properties": {
        "roleDefinitionId": "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/providers/Microsoft.Authorization/roleDefinitions/acdd72a7-3385-48ef-bd42-f606fba81ae7",
        "principalId": "2f9d4375-cbf1-48e8-83c9-2a0be4cb33fb",
        "scope": "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e",
        "createdOn": "2015-10-08T07:28:24.3905077Z",
        "updatedOn": "2015-10-08T07:28:24.3905077Z",
        "createdBy": "877f0ab8-9c5f-420b-bf88-a1c6c7e2643e",
        "updatedBy": "877f0ab8-9c5f-420b-bf88-a1c6c7e2643e"
      },
      "id": "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/providers/Microsoft.Authorization/roleAssignments/baa6e199-ad19-4667-b768-623fde31aedd",
      "type": "Microsoft.Authorization/roleAssignments",
      "name": "baa6e199-ad19-4667-b768-623fde31aedd"
    }
  ],
  "nextLink": null
}
```

Remove access permissions

To delete a role assignment, you must have access to the `Microsoft.Authorization/roleAssignments/delete` operation. Of the built-in roles, only *Owner* and *User Access Administrator* are granted access to this operation.

Request syntax

METHOD	REQUEST URI
DELETE	https://management.azure.com/subscriptions/{subscription-id}/providers/Microsoft.Authorization/roleAssignments/{role-assignment-id}?api-version={api-version}

Request parameters

These are the required parameters for the request:

NAME	TYPE	DESCRIPTION
subscription-id	string	The ID number for the subscription.
role-assignment-id	string	The ID number for the new role assignment.
api-version	string	The current API version that you are using.

Response example

```
{
  "properties": {
    "roleDefinitionId": "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/providers/Microsoft.Authorization/roleDefinitions/9980e02c-c2be-4d73-94e8-173b1dc7ef3c",
    "principalId": "5ac84765-1c8c-4994-94b2-629461bd191b",
    "scope": "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/resourceGroups/Network/providers/Microsoft.Network/virtualNetworks/EASTUS-VNET-01/subnets/Devices-Engineering-ProjectRND",
    "createdOn": "2015-12-17T23:21:40.8921564Z",
    "updatedOn": "2015-12-17T23:21:40.8921564Z",
    "createdBy": "877f0ab8-9c5f-420b-bf88-a1c6c7e2643e",
    "updatedBy": "877f0ab8-9c5f-420b-bf88-a1c6c7e2643e"
  },
  "id": "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/resourceGroups/Network/providers/Microsoft.Network/virtualNetworks/EASTUS-VNET-01/subnets/Devices-Engineering-ProjectRND/providers/Microsoft.Authorization/roleAssignments/5eec22ee-ea5c-431e-8f41-82c560706fd2",
  "type": "Microsoft.Authorization/roleAssignments",
  "name": "5eec22ee-ea5c-431e-8f41-82c560706fd2"
}
```

Next steps

- [Add an existing Azure customer to your Partner Center account](#)

Switch an Azure CSP subscription to a different CSP partner

9/19/2017 • 3 min to read • [Edit Online](#)

This article describes how a customer can change which partner in the Azure Cloud Solution Provider (CSP) program to use for Azure services.

NOTE

This process is complicated, and it requires both partner and customer actions. Currently there is no automatic process to switch an Azure subscription from one CSP partner to another.

This process allows only CSP-to-CSP subscription migrations between existing CSP partners.

Switch CSP partners for an Azure subscription

Three parties are involved in the process of switching CSP partners:

- CSP customer
- Existing CSP partner: The partner who currently owns the subscription
- New CSP partner: The partner who receives the subscription

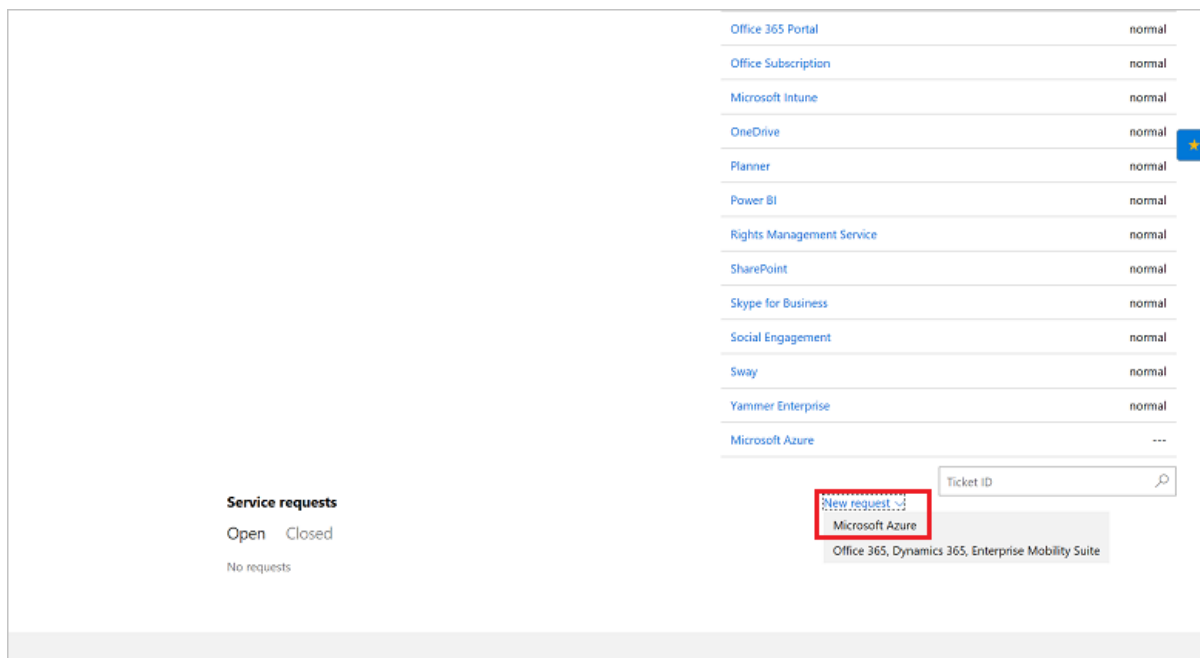
Depending on your role in the partnership, switch partners by doing the following:

1. CSP customer: Contact the existing CSP partner in writing, stating a desire to begin the transfer process.
2. Existing CSP partner: Request a subscription transfer by creating an Azure service ticket in Partner Center:
 - a. On the Partner Center dashboard menu, select **Customers**, select your customer in the list, and then select **classic deployment model**.

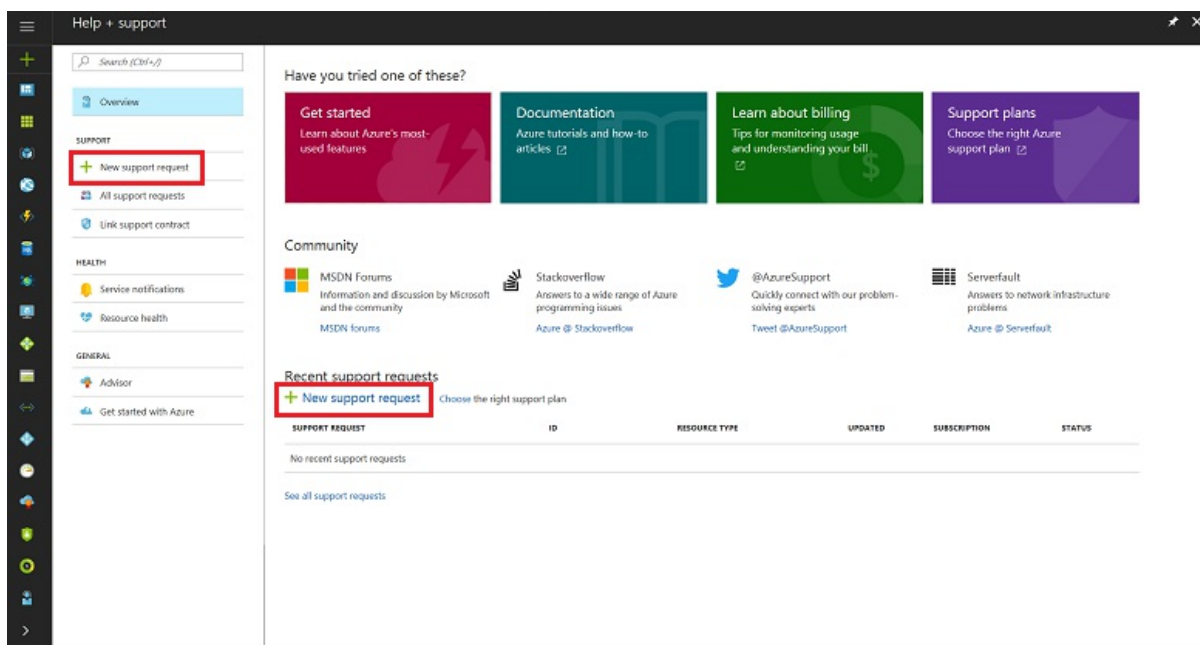
The screenshot shows the Microsoft Partner Center interface. The top navigation bar includes 'Microsoft Partner Center', 'Programs', 'How-to', 'Support', 'Find a Partner', 'Dashboard', and a notification icon. The left sidebar shows a navigation menu with 'Subscriptions', 'Customer insights', 'Users and licenses', 'Service management' (highlighted with a red box), and 'Account'. The main content area is titled 'Subscriptions' and shows 'License-based subscriptions' and 'Usage-based subscriptions'. A table lists the subscription details for 'Microsoft Azure'.

Subscription	Current estimate	% of total	Status
Microsoft Azure		0%	Active

- b. In the **Support tickets** section, in the **New ticket** drop-down list, select **Microsoft Azure**.



c. On the Azure portal **Help & support** page, select **New support request**.



d. In the **New support request** pane, select step **1 - Basics**, select **Subscription management** as the issue type, specify the subscription ID you want to transfer, and then select **Cloud Solution Provider** as the support plan.

e. Select step **2 - Problem**, select **C - Minimal impact** as the severity level, and then select **Other General Questions** as the problem type.

f. Download the [CSP Subscription Transfer form](#).

3. Existing CSP partner: Complete the CSP Subscription Transfer form, sign it, and then send it to the customer. To complete the form, you need the following information:
 - The existing CSP partner's contact information and Microsoft ID. This information is listed on the Partner Center menu under **Account settings > Organization profile**.
 - The customer's Microsoft ID. On the Partner Center menu, select **Customers**, and then expand the listing to view the customer's Microsoft ID.
 - The subscription ID that you want to transfer. In the expanded customer listing, select **View Subscriptions**, and then expand the selected subscription to view the subscription ID.
4. Customer and new CSP partner: Review the form, add information about the new CSP partner, and then sign the form. Confirm that the new customer has a contract agreement in place, and then send the form back to the existing CSP partner.

IMPORTANT

If the new CSP partner does not have a reseller relationship with the customer, establish one prior to transferring the subscription by following the instructions for [adding an existing customer](#).

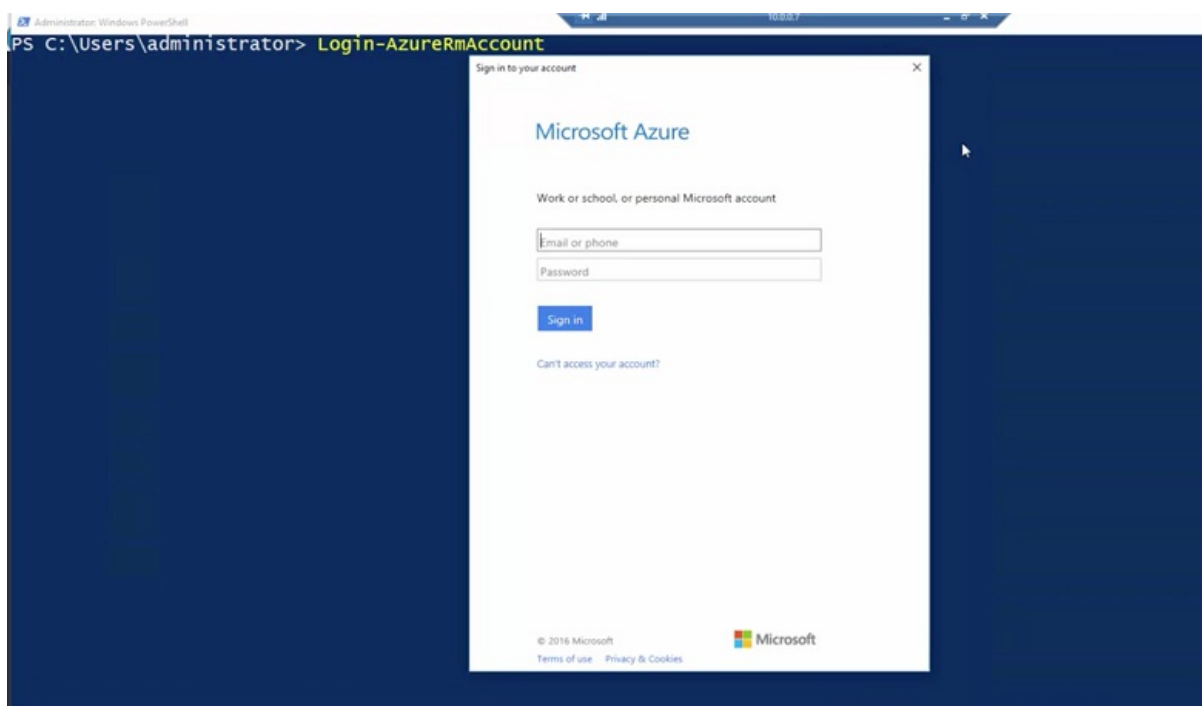
5. Existing CSP partner: Ensure that the form contains correct contact information for both partner admins and that all three signatures are in place. Attach the completed form to the service request by using the **File upload** option. A Microsoft support engineer will contact you within 8 business hours to validate receipt and completion.

The screenshot shows the Microsoft Support Center interface for creating a new support request. The 'Problem' tab is selected, and the 'File upload' section is highlighted with a red box. The 'Details' section contains a text area with the text 'Switching Azure subscription to new partner'. The 'When did the problem start?' section has a date picker set to 2/25. The 'Related help' tab on the right provides additional resources and recommended documents.

6. New CSP partner: Update the Azure subscription settings to remove the existing CSP partner from the account. To see which role assignments are set up, run two PowerShell cmdlets.

a. To add the new CSP partner as the reseller on the account, run:

```
PS C:\> Add-AzureRMAccount -tenant "CustomerDomainName"
```



To find the customer's domain name, in the Partner Center left pane, select **Customers**. In the customer list, select the customer. In the left pane of the customer's page, select **Account**, and then copy the domain name.

Microsoft Partner Center

ACME Corporation

Subscriptions

Customer insights

Users and licenses

Service management

Account

Microsoft ID: 81a570b1-db47-4da6-9abd-a31abacd60b8

Domain name: ACMECorporation1949.onmicrosoft.com

Company info [How can I change this?](#)

Bill-to info [Update](#)

Company

ACME Corporation
1 Tower Lane
Los Angeles CA 90001

Primary contact

mb Blanc@outlook.com

ACME Corporation
1 Tower Lane
Los Angeles CA 90001

Melvin Blanc
mb Blanc@outlook.com

Industries

Manufacturing & resources

Financial services

Retail

Health

Government

Education

For customers

Small & midsize businesses

Enterprise

Security at home

Microsoft Envision

Microsoft Ignite

Microsoft AppSource

For partners

Get listed

Microsoft partner resources

Find a solutions provider

Partner with Microsoft Azure

Microsoft Inspire

For developers

Microsoft Azure

Microsoft Visual Studio

Microsoft Developer Network

TechNet

Application Development

Values

Diversity and inclusion

Accessibility

Microsoft in education

Microsoft philanthropies

Corporate social responsibility

Privacy at Microsoft

Company

Careers

About Microsoft

Company news

Investors

Research

Site map

English (United States)

Contact us

Privacy & cookies

Terms of use

Trademarks

About our ads

© Microsoft 2017

b. To view roles on the account, including previous CSP partners, run:

```
PS C:\> Get-AzureRmRoleAssignment
```

```
Windows PowerShell
PS C:\Users\kirilk> Get-AzureRmRoleAssignment

RoleAssignmentId : /subscriptions/3e951c94-95bd-4d29-b5d2-b2e727268e3a/providers/Microsoft.Authorization/roleAssignme
Scope             : /subscriptions/3e951c94-95bd-4d29-b5d2-b2e727268e3a
DisplayName       : Foreign Principal for 'TEST_TEST_CSP2TDemoTenant' in Role 'TenantAdmins' (Kotyarenko LLC)
SignInName       :
RoleDefinitionName : Owner
RoleDefinitionId  : 8e3af657-a8ff-443c-a75c-2fe8c4bcb635
ObjectId         : 23c52aa4-d972-4230-ae2b-c78b3816d35b
ObjectType       : Group

PS C:\Users\kirilk>
```

c. To remove outdated access permissions to the subscription and resources in the Azure portal, on the Partner Center menu, select **Customers**.

d. Select the customer.

e. On the customer menu, select **Service management**.

Microsoft Partner Center Programs How-to Support Find a Partner Dashboard

ACME Corporation Subscriptions [Export subscriptions](#)

Subscriptions [Add subscription](#)

Customer insights

Users and licenses

Service management

Account

← Customers

License-based subscriptions

This customer doesn't have any license-based subscriptions.

Usage-based subscriptions [All resources \(Azure portal\)](#)

Usage data isn't provided for integration sandbox accounts.

Subscription	Current estimate	% of total	Status
Microsoft Azure		0%	Active

Industries For customers For partners For developers Values Company

Manufacturing & resources Small & midsize businesses Get listed Microsoft Azure Diversity and inclusion Careers

Financial services Enterprise Microsoft partner resources Microsoft Visual Studio Accessibility About Microsoft

Retail Security at home Find a solutions provider Microsoft Developer Network Microsoft in education Company news

Health Microsoft Envision Partner with Microsoft Azure TechNet Microsoft philanthropies Investors

Government Microsoft Ignite Microsoft Inspire Application Development Corporate social responsibility Research

f. To go to the Azure portal, select the **Microsoft Azure** link. To remove unwanted user access, follow the instructions in [Assign permissions to an Azure CSP subscription](#).

Next steps

- [Learn how to add an existing Azure customer to your Partner Center account](#)
- [Learn how to assign Azure CSP subscription permissions to a customer](#)

Azure CSP support overview

9/19/2017 • 7 min to read • [Edit Online](#)

In the Azure Cloud Solution Provider (Azure CSP) model, the CSP partner is the first point of contact for CSP customers. This article provides an overview of the customer support model in Azure CSP.

Customer support services

Azure CSP direct partners provide technical and account support services for their customers. This includes the following:

- Describe the capabilities of different Azure services.
- Provide answers to Azure pricing and usage questions.
- Provide billing and subscription support.
- Provide provisioning and deployment help.
- Resolve performance problems, service availability problems, incomplete software integration problems, or other deployment problems.

IMPORTANT

In the Azure CSP indirect model, these responsibilities are shared between the Azure CSP indirect provider and the Azure CSP indirect reseller. If you are an Azure CSP indirect reseller, ask your Azure CSP indirect provider for details.

Azure CSP partners must provide a clear description to customers of how they will receive support. If you do not provide round-the-clock phone support, you must provide a service level agreement to explain the options to your customers.

Microsoft does not provide any support to Azure CSP customers. These customers rely on their Azure CSP partners.

Customer self-service troubleshooting

Azure CSP customers can handle troubleshooting tasks on their own, without an Azure CSP partner. If the Azure CSP partner provided their customers with [access to the Azure portal](#), they can sign in to and view several helpful features. These include viewing information about service health, seeing in-product notifications and communications, and performing administrative tasks.

Azure CSP customers can get information about publicly known service interruptions in Azure by visiting the [Azure status](#) page. They can also subscribe to [service health notifications](#).

They cannot alter, add, or cancel subscriptions without help from their Azure CSP partner. Nor can customers escalate any service problems to Microsoft. Their Azure CSP partner [should do it](#) on their behalf.

For example, if a customer has access to the Azure portal, and tries to create a new support request via the portal, it doesn't work. Instead, the customer sees a message that this subscription is an Azure CSP subscription, and all support requests must be addressed to a managing Azure CSP partner.

Microsoft Azure

admin@lullabyllc.on... LULLABY LLC

New support request HELP + SUPPORT

Basics NEW SUPPORT REQUEST

1 Basics

2 Problem

3 Contact information

* Issue type
Technical

* Subscription
Microsoft Azure (841fb73a-3261-4cfc-9e...)

* Service
VPN Gateway

Information icon: Your cloud service provider manages your subscription. To open a support request through your provider, use the following contact information.

Contoso
+1(123)456-78-90
support@contoso.com
<https://support.contoso.com>

Alternatively, find solutions to common problems in the Troubleshoot section of your resource. [Learn more](#)

Next

TIP

Don't forget to update your partner's organizational profile on Partner Center. Specify the correct email and phone number of your helpdesk. If you have a web-based ticket managing system or IT Service Management solution, provide a link to the page where your Azure CSP customers can submit a support request.

Partner technical support

Because Azure CSP partners are responsible for technical support, here are some general activities you need to provide:

- Check Azure service health regularly, and notify customers who may be affected by service interruptions.
- Build a knowledge base, and provide resolution suggestions for common problems.
- Use monitoring solutions to see the full picture, and prevent problems.
- To do deep troubleshooting and apply fixes, use admin-on-behalf capabilities to access customer workloads in Azure.
- [Escalate](#) customer problems that you can't resolve to Microsoft.

Check Azure service health

If your customers are having a problem with Azure services, such as services that are down, or a degraded or buggy experience, check the health of that service. You can determine if the problems are on your end, or if there is a known problem with a product. In the latter case, Microsoft might already be working to solve it.

You can get information about known service interruptions in Azure by visiting [Azure status](#). You can also subscribe to subscription-specific [service health notifications](#), which are more granular.

Use integrated monitoring solutions

It is always better to know about a problem before it causes customer service interruption. If you know what's

going on with the Azure workloads of your customers on all levels, you can prevent problems, and resolve problems much faster.

Use an integrated monitoring solution that supports multi-customer view, and can provide deep analysis of Azure service reliability and performance.

Review the [Azure monitoring and diagnostics guide](#) to get started.

Microsoft provides the following solutions:

- [Azure Log Analytics](#)
- [Operations Management Suite](#). Operations Management Suite includes Azure Log Analytics as an engine, and provides additional monitoring capabilities like Service Map and compliance monitoring.

You can also use the following third-party integrated monitoring solutions that support Azure CSP:

- [Datadog](#). This is a monitoring solution with greater than 200 integrations with other services.
- [Alert Logic](#). This is a deep security monitoring and breach prevention solution.

All the preceding solutions are offered in a software as a service (SaaS) model. You don't need to install anything additional in your environment.

Administer on behalf of the customer

As an Azure CSP partner, you have delegated admin privileges that give you access to your customers' environments. This allows you to directly support, configure, and manage their subscriptions.

With Partner Center, you can view customer service health and service incidents, as well as perform administrative tasks on their behalf to correct missing or poorly configured settings.

To perform administrative tasks in Partner Center, from the **Dashboard**, go to **Customers** > select the specific customer > **Service management**. Then, under **Administer services**, select **Microsoft Azure Management Portal**.

Service	Severity
Dynamics CRM Online	normal
Exchange Online	normal
Identity Service	normal
Microsoft Dynamics Marketing	normal
Mobile Device Management	normal
Office 365 Portal	normal
Office Subscription	normal
Microsoft Intune	normal
OneDrive	normal
Planner	normal
Power BI	normal
Rights Management Service	normal
SharePoint	normal

Billing support

You own the billing relationship with your customers, and must fully support any billing-related questions or problems they have. For more details about billing in Azure CSP, see the [billing overview](#).

Azure CSP customers might ask you about Azure service pricing rules, how they can optimize their spending, and

what their current spending is between billing cycles.

To help your customers better understand their spending in Azure CSP, you can use the following third-party cloud management solutions that support Azure CSP:

- [Cloudyn](#)
- [CloudCheckr](#)
- [CloudHealth](#)
- [RightScale](#)

If you are a CSP direct partner, you are responsible for payment to Microsoft. You receive an invoice for consumed Azure CSP services by all your customers every billing cycle, and that invoice should be paid within 60 days. It is then your responsibility to ensure collection of on-time payment from your customers. Microsoft recommends establishing a clear collections policy with customers, including when a subscription may be affected by non-payment.

If necessary, you can [suspend a customer's subscription](#) due to non-payment.

If the customer wants to dispute a bill, has concerns about fraudulent subscription use, or has any other billing concerns that you cannot handle, you can escalate the problem to Microsoft.

Escalate problems to Microsoft

Although you are responsible for customer technical and billing support, your customer might have problems that only Microsoft can solve. These include the following:

- Undocumented problems with services that aren't operating according to service descriptions
- Unavailable services
- Bugs and other irregularities that affect service appearance or operation
- Large scale network disruptions
- Regional problems with multi-tenant impact
- Billing disputes

In these cases, you must escalate such problems to Microsoft. Create a service request in Partner Center, as outlined in the [escalation guide](#).

Even if you've escalated a customer's problem to Microsoft, it is still your responsibility to manage the process. Microsoft support engineers communicate with a customer only through partner.

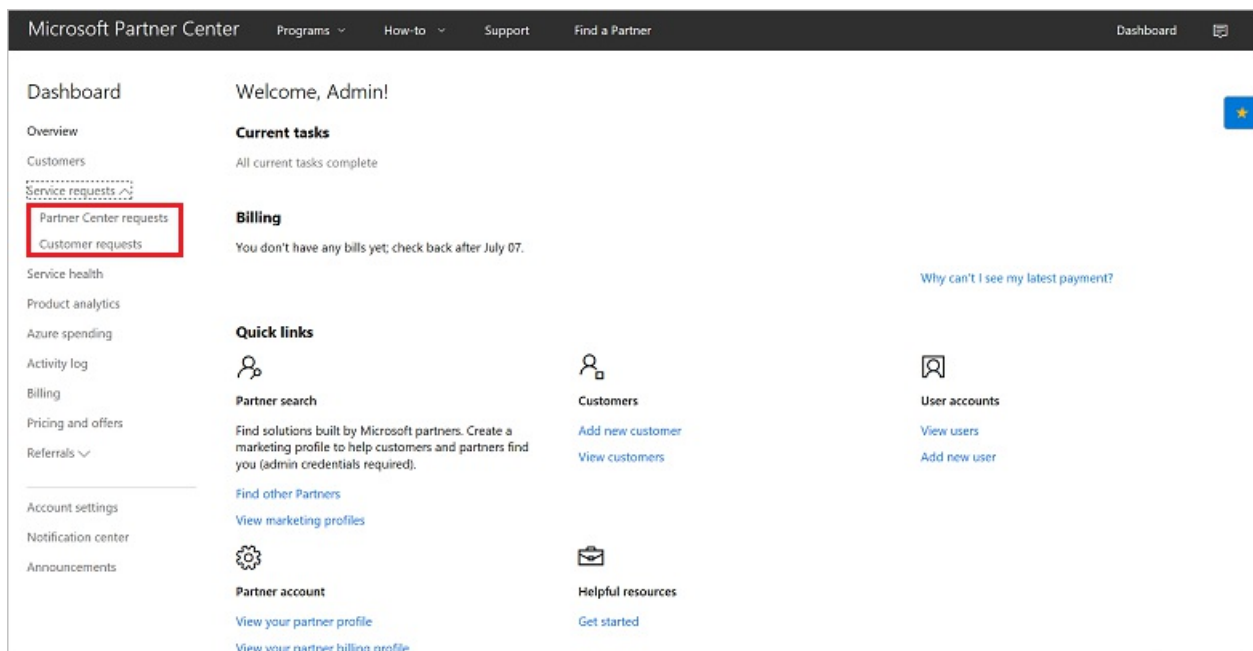
You can expect a response from Microsoft within 8 hours for a Severity C request, 4 hours for Severity B, and 2 hours for Severity A. Note that these times are reduced if you have purchased other [partner support plans](#). Phone and email support is similarly only available to those who have purchased additional support plans.

After you communicate with Microsoft, you must communicate with the customer to provide updates or confirmation that the problem is resolved.

Support tickets in Partner Center and the Azure portal

Be aware that a support request filed through Partner Center and one filed through the Azure portal are not the same.

From the Partner Center **Dashboard**, in the left pane, you can select **Service Requests**. You then see two options: **Partner Center requests** and **Customer requests**.



Partner Center requests are strictly for reporting problems with Partner Center itself. Customer requests are where you can see any support requests that your customers have filed with you.

If a customer request is for a problem you cannot solve, you must go to the Azure portal to submit a request to Microsoft, as detailed in the [escalation guide](#).

If you are filing a service request about Azure services with Microsoft, and you are not in the Azure portal, you are filing your service request in the wrong place.

Next steps

- Learn [how Azure CSP partners can escalate customer problems with Azure to Microsoft](#).
- Review [available support plans for Azure CSP partners](#).

Support options for Azure CSP partners

9/19/2017 • 3 min to read • [Edit Online](#)

If you are an Azure CSP direct partner, with your Microsoft Partner Center account, you receive a free Azure Cloud Solution Provider (Azure CSP) support plan. You are still [responsible for Azure CSP customer support](#). But, if you encounter a problem that you cannot fix on your own, you can use your support plan to escalate customer issues to Microsoft.

Some Azure CSP partners might want a deeper level of support for Azure services for their customers. For this reason, two higher levels of support are available: [Advanced Support for Partners \(ASfP\)](#) and [Premier Support for Partners \(PSfP\)](#). In this article, we describe the difference between these paid support plans and the free pricing tier.

IMPORTANT

This article covers only the support options that are available to Azure CSP direct partners. If you are an Azure CSP indirect reseller, contact your Azure CSP indirect provider to learn about the options that are available to you.

Azure CSP support plan comparison

The following table gives you an at-a-glance view of the primary differences between the free Azure CSP support plan and the paid support tiers (ASfP and PSfP).

SERVICE	INCLUDED	ASFP	PSFP
Services Account Management	No dedicated manager	Assigned from a regional pool; limited hours per month	Designated support manager
Azure Cloud Services technical break-fixes	Through Partner Center	Through Partner Center, email, or phone	Through Partner Center, email, or phone
Initial response times by severity	<ul style="list-style-type: none">Severity A: 2 hSeverity B: 4 hSeverity C: 8 h	<ul style="list-style-type: none">Severity A: 1 hSeverity B: 2 hSeverity C: 4 h	<ul style="list-style-type: none">Severity A: 1 hSeverity B: 2 hSeverity C: 4 h
Resolution priority	Low	Medium	High
24/7 critical situation support	No	No	Yes
Advisory "how to" services	Requires Microsoft Partner Network (MPN) Silver/Gold competency	Yes	Yes
Technical presales assistance	Requires MPN Silver/Gold competency	Yes	Yes
Azure CSP integration consulting	No	Packaged	Customizable

SERVICE	INCLUDED	ASFP	PSFP
Proactive support services	No	Packaged	Customizable
Minimum price (USD)	Free	\$15,000/year	\$28,000/year

Service details

Services Account Management

The Services Account Management team is a team of managers who act as service and support advisors. Team members deliver proactive guidance about potential blockers, and offer assistance in the most critical cases.

At the ASfP level, you have access to 5 hours per month with the Services Account Management team. Additional hours per month are available as an add-on purchase.

At the PSfP level, you have a designated Technical Account Manager (TAM) who provides personalized services in proactive planning and resolution handling.

Azure Cloud Services technical break-fixes

With both the ASfP and PSfP levels of support, you can contact Microsoft as needed by phone and email to resolve Azure Cloud Services break-fix support incidents. This gives you a more direct line to the support you need when issues arise.

Reduced response times

As described in the Azure CSP support plan comparison table, response times by severity are cut in half for Azure CSP partners who have ASfP and PSfP levels of support. For partners who have these support levels, support requests filed with Microsoft are given priority. Severity A issues receive a response in 1 hour or less. If necessary, Severity A issues can be escalated to the Services Account Management team. This does consume an ASfP partner's allotted plan hours.

24/7 critical situation support

PSfP partners receive support 24/7 if critical situations arise and support is urgently needed. Microsoft Support engineers work on problem resolution 24/7. Support engineers transfer a case from one time zone to another to ensure that an engineer is working on a problem continuously.

Additional information

- Learn more about [Advanced Support for Partners](#).
- Learn more about [Premier Support for Partners](#).

Next steps

- [Choose the support plan](#) that can help you ensure a great Azure CSP support experience for your customers.
- Learn how to [escalate issues to Microsoft through Partner Center](#), regardless of the support plan you have.

Escalate Azure Cloud Solution Provider issues to Microsoft

10/6/2017 • 2 min to read • [Edit Online](#)

If you are an Azure Cloud Solution Provider (Azure CSP) partner, you are [responsible for customer technical and billing support](#). But sometimes your customers will experience issues with Azure that only Microsoft can solve. These might include the following:

- Undocumented problems with Azure services that aren't operating according to service descriptions
- Unavailable services
- Bugs and other irregularities that affect service appearance or operation
- Large-scale network disruptions
- Regional issues that have multitenant impact
- Billing disputes

You need to escalate these kinds of issues with Azure services to Microsoft, and create a support request. First, though, ensure that the issue is not caused by an incorrect service configuration. Also, ensure that Microsoft supports the customer's architecture.

IMPORTANT

Don't escalate every customer issue to Microsoft. First, complete proper troubleshooting on your side.

Even if you've escalated a customer's issue with Azure to Microsoft, it's still your responsibility to manage the resolution process. Microsoft support engineers will communicate with a customer only through a partner.

IMPORTANT

In this article, we cover the support process for Azure CSP direct partners. If you're an Azure CSP indirect reseller, contact your Azure CSP indirect provider for details about how this process will work for you.

Create a service request

1. On the Partner Center **Dashboard** menu, select **Customers**. Select the customer, and then select **Service management**.

Microsoft Partner Center Programs How-to Support Find a Partner Dashboard

ACME Corporation Subscriptions Export subscriptions

Subscriptions Add subscription

Customer insights

Users and licenses

Service management

Account

← Customers

License-based subscriptions

This customer doesn't have any license-based subscriptions.

Usage-based subscriptions All resources (Azure portal)

Usage data isn't provided for integration sandbox accounts.

Subscription	Current estimate	% of total	Status
Microsoft Azure		0%	Active

Industries	For customers	For partners	For developers	Values	Company
Manufacturing & resources	Small & midsize businesses	Get listed	Microsoft Azure	Diversity and inclusion	Careers
Financial services	Enterprise	Microsoft partner resources	Microsoft Visual Studio	Accessibility	About Microsoft
Retail	Security at home	Find a solutions provider	Microsoft Developer Network	Microsoft in education	Company news
Health	Microsoft Envision	Partner with Microsoft Azure	Technet	Microsoft philanthropies	Investors
Government	Microsoft Ignite	Microsoft Inspire	Application Development	Corporate social responsibility	Research

2. In the **Support tickets** section, select the **New request** drop-down menu, and then select **Microsoft Azure**.

Office 365 Portal	normal
Office Subscription	normal
Microsoft Intune	normal
OneDrive	normal
Planner	normal
Power BI	normal
Rights Management Service	normal
SharePoint	normal
Skype for Business	normal
Social Engagement	normal
Sway	normal
Yammer Enterprise	normal
Microsoft Azure	---

Service requests

Open Closed

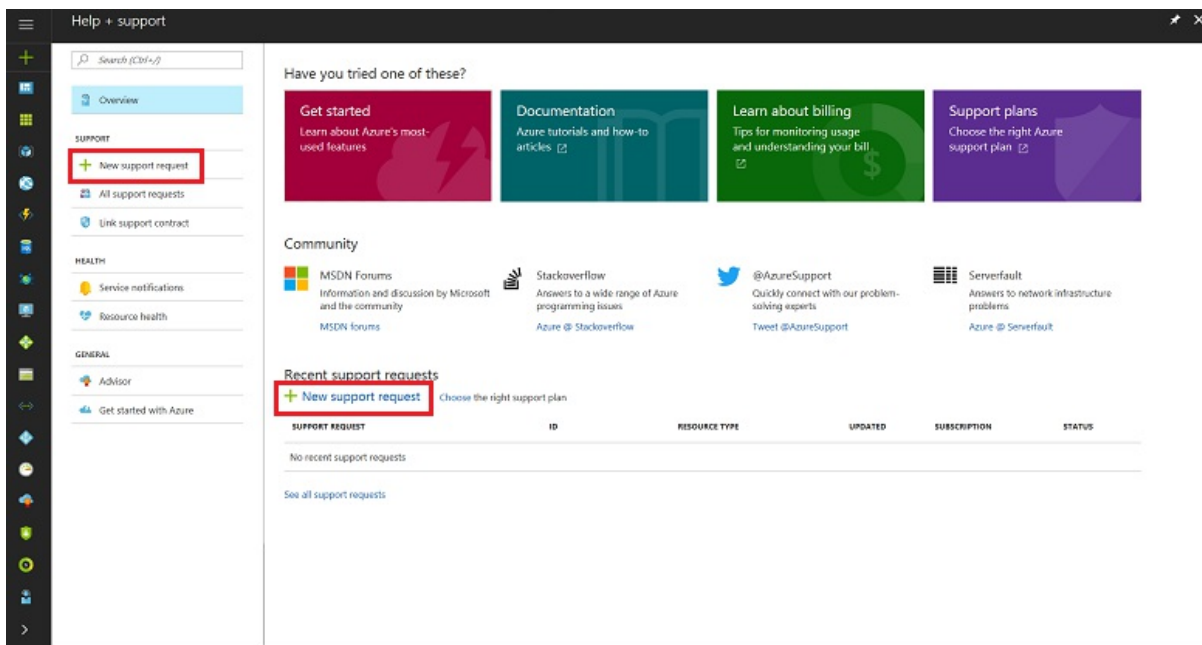
No requests

New request Ticket ID

Microsoft Azure

Office 365, Dynamics 365, Enterprise Mobility Suite

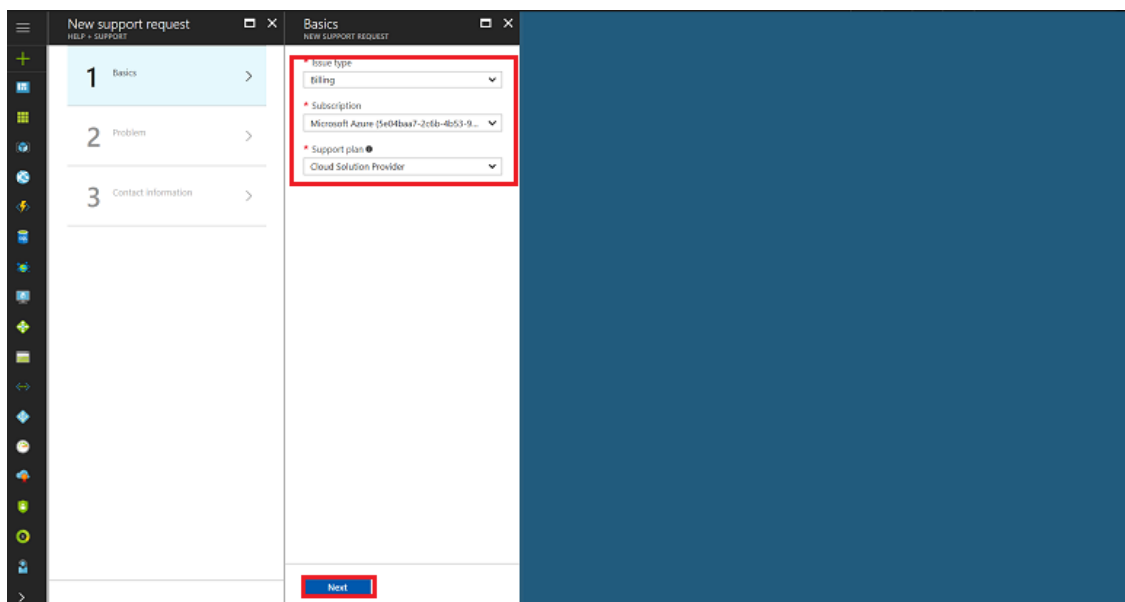
3. In the Azure portal, select **New support request**.



4. Enter the relevant information for the service request, and then select **Submit**. For more information about how to fill out the support request, see [Support request form](#).

Support request form

1. On the **Basics** blade:
 - a. Select an **Issue type** (**Technical**, **Billing**, **Quota**, or **Subscription management**).
 - b. Select the **Subscription** to which the issue applies.
 - c. Select the **Support plan**. The default support plan is **Cloud Solution Provider**. [Additional plans](#) might be listed, if you've purchased them.



2. On the **Problem** blade:
 - a. Select the problem's **Severity**. This ranges from **C - Minimal impact** (low priority) to **A - Critical impact** (immediate priority).
 - b. Select the **Problem type**. The options vary, depending on the issue type you selected in step 1. Select the option that most closely resembles the problem. When you select the problem type, you see a list of recommended steps and documents that might help you solve those types of problems. *Try these solutions before you submit a support request.*
 - c. In the **Details** section, describe the issue with as much information as possible. Include any error

messages, if applicable.

- d. Under **When did the problem start**, enter the date and local time at which the problem first occurred.
- e. If applicable, upload a file to the **File upload** section.

The screenshot shows the 'Problem' tab of the 'New support request' form in the Microsoft Azure portal. The form is divided into three main sections: 'Basics', 'Problem', and 'Contact information'. The 'Problem' section is currently active and highlighted in blue. It contains the following fields: 'Severity' (set to 'C - Minimal impact'), 'Problem type' (set to 'Change Payment Method'), and 'Details' (with the text 'I need to change my payment method.'). Below these fields, there is a section for 'When did the problem start?' with options to 'Choose a date' or 'Enter a local time'. There is also a 'File upload' section with a 'Select a file' button. At the bottom of the form, there is a 'Next' button. The right side of the screen shows a 'Related help' section with various links and solutions related to the problem description.

3. On the **Contact information** blade:

- a. For **Preferred contact method**, select email or phone as the preferred way to reach the point of contact.
- b. In **Contact information**, provide *your* contact information, *not* your customer's information.

4. Select **Create**. The support request appears in Partner Center for the support team to address.

The screenshot shows the 'Contact information' tab of the 'New support request' form in the Microsoft Azure portal. The form is divided into three main sections: 'Basics', 'Problem', and 'Contact information'. The 'Contact information' section is currently active and highlighted in blue. It contains the following fields: 'Preferred contact method' (set to 'Email'), 'Response' (set to 'Business Hours'), 'Language' (set to 'English'), 'First name' (set to 'John'), 'Last name' (set to 'Doe'), 'Email' (set to 'jg1990@outlook.com'), 'Who else should we email?' (set to ' '), 'Phone number' (set to ' '), and 'Country/region' (set to 'United States'). There is also a checkbox for 'Save contact changes for future support requests'. At the bottom of the form, there is a 'Create' button. The right side of the screen shows a 'Related help' section with various links and solutions related to the problem description.

Later, review your customer's service requests within the Microsoft Azure portal by selecting **Manage support requests**.

If you are experiencing issues with Partner Center portal itself, not related to Azure services, you should use [Partner Center problem reporting](#) procedure.

Next steps

- Review [support plans for Azure CSP partners](#).

- Learn about [Azure CSP partner support responsibilities](#).
- Review [support from Microsoft](#) section of Partner Center documentation.

Azure CSP billing overview

10/17/2017 • 6 min to read • [Edit Online](#)

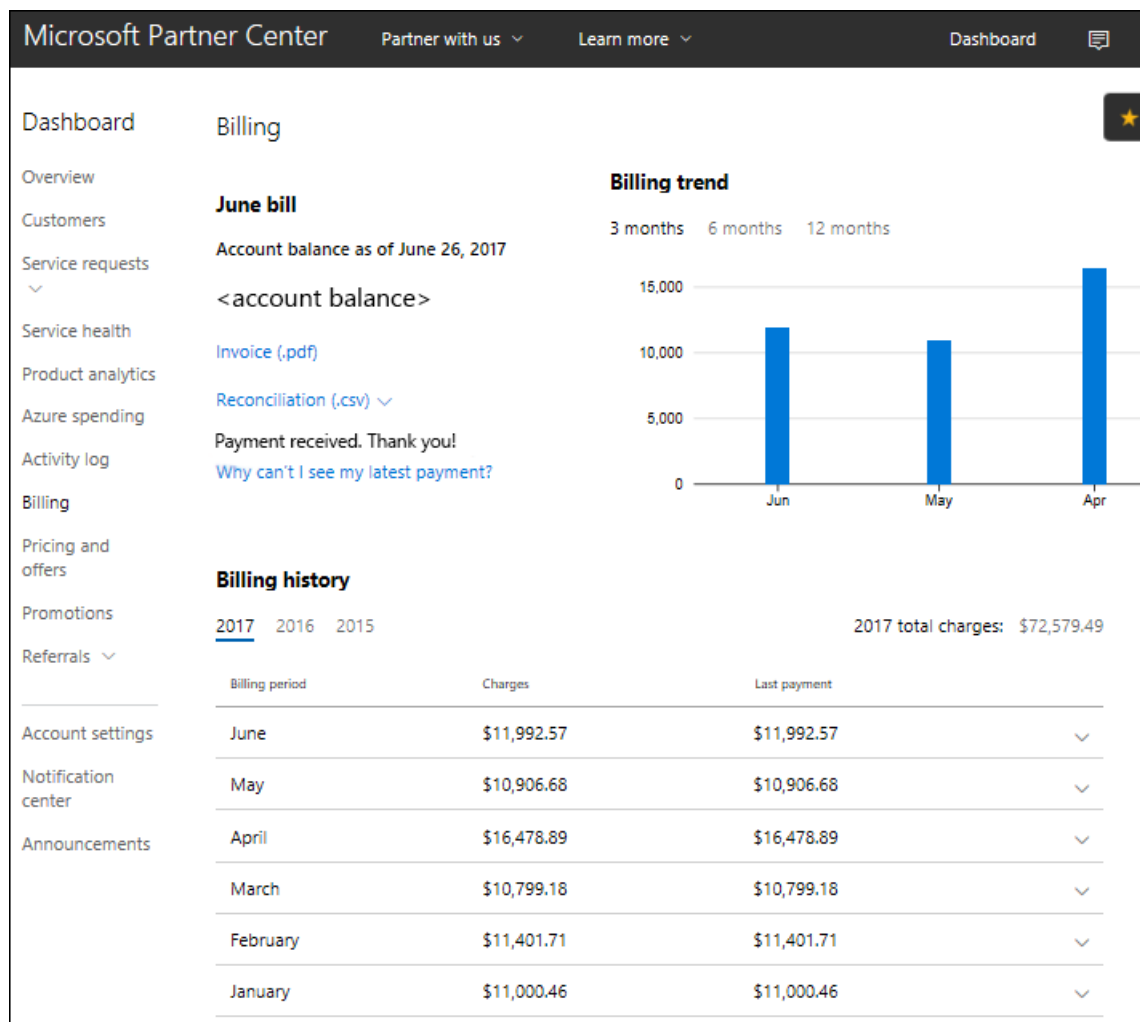
This guide covers the basics of how billing works in the Azure Cloud Solution Provider (Azure CSP) program, including the billing cycle, pricing, and usage.

IMPORTANT

This article covers the billing model for Azure CSP direct partners. If you are an Azure CSP indirect reseller, ask your Azure CSP indirect provider for details.

Billing cycle

When you enroll in the Azure CSP program, you select the first day of the billing cycle (usually partners select the first day of the month). This choice determines your monthly billing cycle.

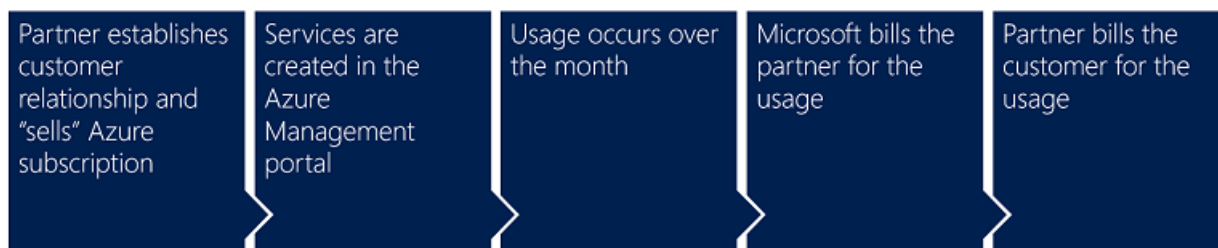


General billing rules:

- Pay to Microsoft within 60 days after the invoice is generated.
- The invoice is based on the total consumption of all your customers during the last billing cycle.
- Microsoft doesn't have any requirements or conditions about how you handle your own invoicing. You define the rules for your customers.

- To determine the Azure usage of a specific customer, use the customer ID/name and other relevant fields within the single reconciliation file. This file is generated every month on Partner Center.
- Add your own fees and charges for services that you provide. Microsoft is not aware of this information.
- Microsoft generates a single invoice and a single reconciliation file, containing information for all your Azure CSP customers. You are responsible for splitting this information and providing separate bills to your customers.

The billing, pricing, and usage structure of Azure CSP follows a simple sequence of events:



- **Step 1:** The Azure CSP partner (you) establishes a relationship with a customer ([new](#) or [existing](#) Microsoft customer). In securing this relationship, you agree to "sell" them an Azure subscription. You now own this relationship from end to end, including handling all billing and collections.
- **Step 2:** You go to the Azure portal and create the services that you are selling to the customer. Essentially, you are purchasing the service from Microsoft and then giving it to your customer.
- **Step 3:** Usage occurs throughout the month. Because Azure services are usage-based, it is impossible to know what the monthly bill will be up front. The bill changes from month to month based on how much of the Azure services your customer uses.
- **Step 4:** At the end of the billing cycle, Microsoft sends you an invoice for your customer's usage. You are responsible for paying your bill to Microsoft. That invoice is not necessarily restricted to Azure services. You may see other Microsoft products (for example, Office 365 or Enterprise Mobility + Security) if you've provided them through this Partner Center account to your customers.
- **Step 5:** After you receive the itemized bill from Microsoft, it is your responsibility to create invoices and collect payment from your customer.

Typically you will have more than one customer. Therefore, when you receive one large bill from Microsoft, you will break it up into many smaller bills in order to collect from each of your customers.

For more information on reading and understanding your bill, see [Azure CSP invoice](#).

Azure usage information

Because Azure subscriptions are usage-based, it is hard to predict the exact price of a given month. However, with previous usage data, you can get an estimate of how much Azure usage to expect from a particular customer. This information can also be useful for determining the source of charges if your bill from Microsoft is larger than expected.

NOTE

Microsoft doesn't provide any usage reports or consumption data to Azure CSP customers. Partner Center usage data is partner-facing.

You can retrieve this information either through the Partner Center portal, or by using Partner Center usage APIs. You can get Azure usage information for all your customers.

Microsoft Partner Center

Partner with usFind a PartnerGet support

Dashboard

Dashboard

OverviewCustomersService requestsService healthProduct analyticsAzure spendingActivity logBillingPricing and offersPromotionsReferralsAccount settingsNotification centerAnnouncements

Azure spending

Email notifications

Billing period

Customers with Microsoft Azure subscriptions

Monthly budget

Customer	Monthly budget	Current estimate	% used
SEATTLEIOT	\$100	\$123.39	123%
ANDREWS CONCERT TICKETS	\$1,000	\$244.11	24%
DAN INC	\$5,000	\$835.31	17%

Azure spending

Email notifications

Email notifications

Keep an eye on your azure bill. Set a monthly budget for each customer - if they use 80% or exceed their set spend budget, we'll send you an email warning. The email will be sent to john@contoso.com

[Change email address](#)

Get emails

☒ On

Billing period

June 26 - July 25

Days remaining

20

Customers with Microsoft Azure subscriptions

Report for 7/4/17 8:38 PM

This data is an estimate, and may be delayed by up to 48 hours.

Monthly budget [Apply](#) [Remove budget](#)

<input type="checkbox"/>	Customer	Monthly budget	Current estimate	% used	
<input type="checkbox"/>	SEATTLEIOT	\$100	\$123.39	123%	<div></div>
<input type="checkbox"/>	ANDREWS CONCERT TICKETS	\$1,000	\$244.11	24%	<div></div>
<input type="checkbox"/>	DAN INC	\$5,000	\$835.31	17%	<div></div>

You can also drill down to usage details inside a specific Azure CSP subscription of a particular customer:

Microsoft Partner Center

Partner with us

Learn more

Dashboard

Fabrikam

Subscriptions

Microsoft Azure

Devices

Customer insights

Users and licenses

Service management

Account

Microsoft Azure

Subscription ID: 4327C34B-75A7-489A-B823-822F9A1C7F12

Details

Subscription nickname

Microsoft Azure

Quantity

Usage-based

Status

☒ Active

☐ Suspended

Submit

Cancel

← Customers

Estimated usage

Report for 7/4/17 6:23 PM

Current estimate

\$26.40

Days remaining

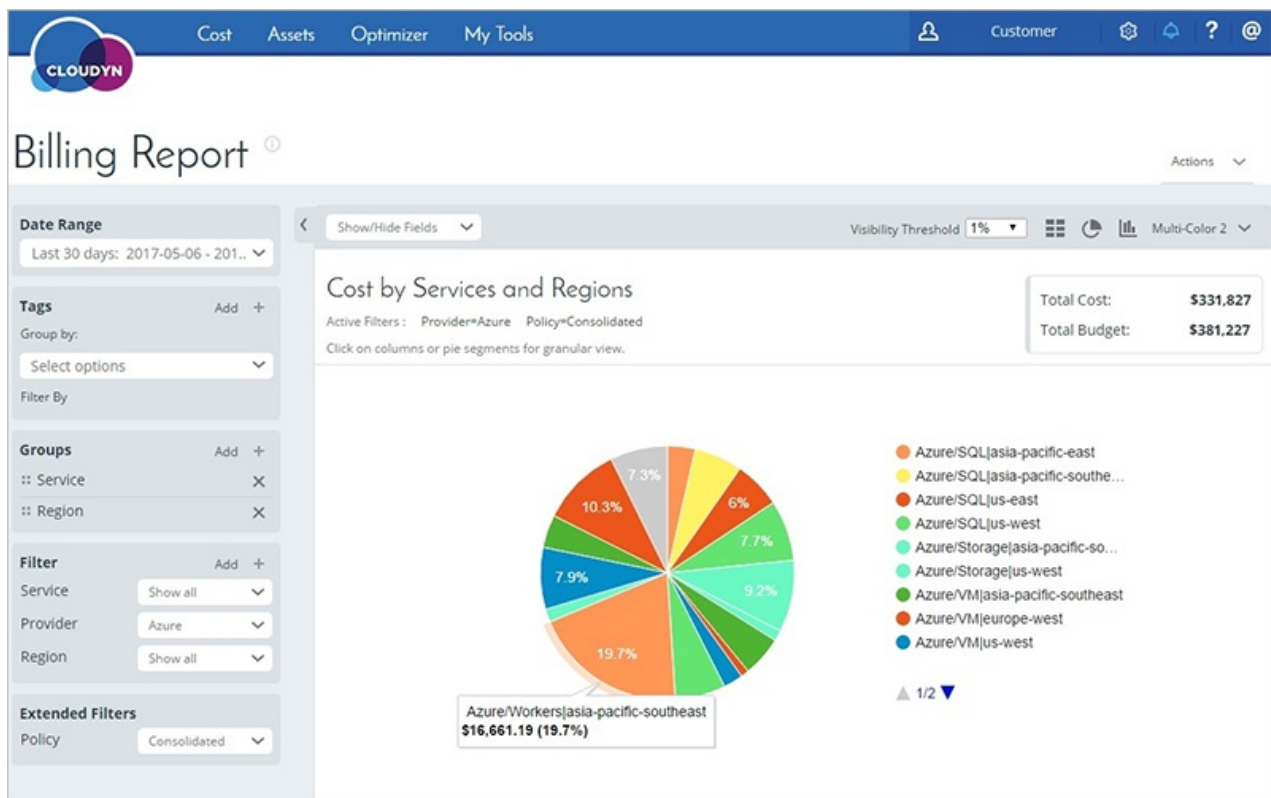
20

Itemized costs

Report for 7/4/17 6:23 PM

Item	Amount used	Estimated cost ↑
Azure App Service - Standard Small App Service Hours	203.00 Hours	\$17.08
Data Services - Azure Database for Postgres - Basic Compute Unit	10,150.00 Hours	\$4.07
Data Services - Azure Database for MySQL - Basic Compute Unit	10,150.00 Hours	\$4.07
Cognitive Services - Custom Speech Service - Offline Testing	0.00 1 Second	\$0.00
Service Bus - Hybrid Connections - Data Transfer	0.00 GB	\$0.00
Service Bus - Hybrid Connections - Data Transfer	0.00 GB	\$0.00
	0.00	\$0.00

If you'd like to provide Azure usage details and rich cost analysis capabilities to your customers, you can use [Azure Cost Management by Cloudyn](#), integrated with Partner Center.



Azure Cost Management by Cloudyn allows you to track cloud usage and expenditures for Azure and other cloud providers including AWS and Google for all your customers. Easy-to-understand dashboard reports help with cost allocation and showbacks/chargebacks as well. Cost Management helps optimize your cloud spending by identifying underutilized resources that you can then manage and adjust.

You can also use the following third-party cloud management solutions that support Azure CSP, to manage the billing process and provide usage insights to your customers:

- [CloudCheckr](#)
- [CloudHealth](#)
- [RightScale](#)

Set usage limits

Partners sometimes want protection from the possibility of customers' overspending. But the challenge is that Microsoft never knows what to do with customers' Azure resources when some spending threshold is reached, so there is no way to set hard limits for Azure usage in the Partner Center portal.

To completely stop Azure usage for the customer, the Azure subscription should be suspended. As a result of this suspension, all of a customer's VMs are shut down, all storage accounts become inaccessible, and all other Azure resources that the customer used stop functioning. Suspension leads to a complete interruption of workloads that were deployed in the customer's subscription. This level of response may be unnecessary. For example, maybe the customer simply had an expensive marketing campaign, supported by several Azure VMs, and usage was slightly higher than expected.

To avoid excessive Azure usage in Azure CSP, you can set **budgets** on the Partner level (for all Azure subscriptions of all customers), and also on the customer's Azure subscription level.

Microsoft Partner Center Partner with us Learn more Find a Partner Get support Dashboard

Dashboard Azure spending Email notifications

Overview Customers Service requests Service health Product analytics Azure spending Activity log Billing Pricing and offers Promotions Referrals Account settings Notification center Announcements

Email notifications
Keep an eye on your azure bill. Set a monthly budget for each customer - if they use 80% or exceed their set spend budget, we'll send you an email warning. The email will be sent to john@contoso.com
[Change email address](#)

Get emails
On

Billing period
June 26 - July 25

Days remaining
20

Customers with Microsoft Azure subscriptions Report for 7/4/17 8:38 PM
This data is an estimate, and may be delayed by up to 48 hours.

Monthly budget Apply Remove budget

Customer	Monthly budget	Current estimate	% used
SEATTLEIOT	\$100	\$123.39	123%
ANDREWS CONCERT TICKETS	\$1,000	\$244.11	24%
DAN INC	\$5,000	\$835.31	17%

Microsoft Partner Center Partner with us Learn more Find a Partner Get support

SeattleIoT Subscriptions

Subscriptions Devices Customer insights Users and licenses Service management Account

Add subscription

License-based subscriptions
This customer doesn't have any license-based subscriptions.

Usage-based subscriptions Report for 7/4/17 6:23 PM [All resources \(Azure portal\)](#)

Current estimate
\$123.39

62% of their \$200 monthly budget.

[Apply](#) [Cancel](#)

Billing period June 26 - July 25. There are 20 days left.

Subscription	Current estimate	% of total	Status
Microsoft Azure	\$123.39	100%	Active

The budget is a soft threshold. When you reach the threshold, you receive a notification. You can then do one of the following:

- **Leave it as it is.** Excessive usage was planned, and the customer will pay for it.
- **Notify the customer about the excessive Azure usage.** Get the customer's approval that he or she will pay for it.
- **Revoke the customer's access to the Azure portal until the issue is resolved.** All Azure resources will be functional, but customer won't be able to create new or modify existing resources.
- **Stop the customer's VMs.** Data is still accessible, but the customer won't be charged for additional compute hours.
- **Suspend the Azure CSP subscription.** This completely stops Azure usage meters for that subscription. All of the customer's VMs are shut down, all storage accounts are inaccessible, and all other Azure resources that the customer used stop functioning.

For more information, see [Set an Azure budget for your customers](#).

There is no single rule for every excessive usage situation. Budgeting capabilities allow you to set per-partner or per-customer soft thresholds, and react in a required way after receiving the notification.

Azure pricing information

In addition to being usage-based, Azure services are **dynamically priced**. This means that the prices for different Azure services may change from month to month, and sometimes even within a month.

All of this can make estimating the price of an Azure service very difficult, but there are tools in place to make it easier. For example, try the [Azure channel pricing calculator](#) (requires authorization with Azure CSP partner credentials).

The screenshot shows the 'Azure Channel Pricing Calculator' interface. At the top, there's a blue header with the title and a 'PREVIEW' label. Below the header, there's a green '+ Add items' button and a 'Clear all' link. The main area is divided into two columns. The left column has a 'Name: Your estimate' field, a section for 'Commonly purchased together' with icons for VPN Gateway, SQL Database, and App Service, and a detailed configuration for 'Web Servers'. The 'Web Servers' configuration includes dropdowns for 'REGION' (West US 2), 'TYPE' (Windows), and 'PRICING TIER' (Standard), along with an 'ADD MANAGED DISKS' checkbox. Below this is a 'NOTE' about saving up to 40 percent on Windows Server VMs. The 'INSTANCE SIZE' is set to 'A2m v2: 2 cores, 16 GB RAM, 20 GB disk'. At the bottom, there's a calculation: '2 Virtual Machines' multiplied by '744 Hours' equals '<price>/MO' for SKU 'F2D0E4D6-CC5F-44CD-BCE0-2198D7B00225'. The right column shows a summary of the estimate, including currency (US Dollar), a list of services (Web Servers, Support Options), a 'SUB-TOTAL', 'DISCOUNTS AND MARKUPS', a 'CSP' dropdown, a 'Partner Markup' of 20%, and an 'ESTIMATED MONTHLY COST' of '<total cost>'. A green 'Save' button with a right arrow is highlighted with a red rectangle. Below the 'Save' button is a 'Show my estimates' link and a disclaimer: 'Prices are estimates and are not intended as actual price quotes.'

You may also want to try this [Open source pricing calculator on GitHub](#). It can be used to estimate the cost of hosting a solution in an Azure CSP subscription. It takes the Azure Resource Manager template as an input, and gives the estimated costs as an output.

NOTE

Customers don't see any prices on the Azure portal. This is because Microsoft doesn't know the final customer price for Azure services and billing rules. These are set by the Azure CSP partner for their customers.

Choose a size
Browse the available sizes and their features

Supported disk type: SSD Minimum cores: 1 Minimum memory (GiB): 0

★ Recommended | [View all](#)

DS1_V2 Standard ★	DS2_V2 Standard ★	DS11_V2 Standard ★
1 Core	2 Cores	2 Cores
3.5 GB	7 GB	14 GB
2 Data disks	4 Data disks	4 Data disks
3200 Max IOPS	6400 Max IOPS	6400 Max IOPS
7 GB Local SSD	14 GB Local SSD	28 GB Local SSD
Load balancing	Load balancing	Load balancing
Premium disk support	Premium disk support	Premium disk support

For pricing information, please contact your reseller.

For pricing information, please contact your reseller.

For pricing information, please contact your reseller.

Next steps

- [Learn](#) how to work with the Azure CSP price list.
- [Learn](#) how to analyze a Microsoft CSP invoice.
- [Read](#) more to better understand how Azure CSP billing works.

Azure Cloud Solution Provider price list

9/22/2017 • 3 min to read • [Edit Online](#)

Azure prices are dynamic, so it's impossible to have a standard usage rate for Azure services. Microsoft offers tools to help you get the most up-to-date Azure prices. One of these tools is the Azure Cloud Solution Provider (Azure CSP) price list. You can download the Azure CSP price list from Microsoft Partner Center.

The Azure CSP price list is an Excel file that has the most current pricing information for all Azure services. It's a long list, with more than 11,000 rows. It lists all available Azure services, regions, and plans; all VM sizes; and so on. Prices for all available currencies are included.

	B	C	D	E	F	J	K	L
1	Meter Category	Meter Subcategory	Region	Name	Resource ID	AUD	CAD	CHF
7408	Virtual Machines	A0 VM	US Central	Compute Hours	5b78f2d3-3c78-4333-b6a8-7e2233433a0b	<price>	<price>	<price>
7409	Virtual Machines	A0 VM	US East	Compute Hours	8be390a3-b121-4937-bc0c-67776102ae5d	<price>	<price>	<price>
7410	Virtual Machines	A0 VM	US East 2	Compute Hours	e7d4f4ca-0abf-4f2f-9c32-0d608668db67	<price>	<price>	<price>
7411	Virtual Machines	A0 VM	US Gov AZ	Compute Hours	59751b5f-4a67-4493-8404-9bf15d8b1b52	<price>	<price>	<price>
7412	Virtual Machines	A0 VM	US Gov TX	Compute Hours	f2df7611-15cf-4ac8-887f-81ac23691eb7	<price>	<price>	<price>
7413	Virtual Machines	A0 VM	US North Central	Compute Hours	5b69bef7-ee70-4b02-89b8-3b3dd5ddcc00	<price>	<price>	<price>
7414	Virtual Machines	A0 VM	US South Central	Compute Hours	f0876270-a519-4c34-b8d6-12cf8cb86fea	<price>	<price>	<price>
7415	Virtual Machines	A0 VM	US West	Compute Hours	7625ba50-fbc6-4e6a-9e25-4c8354088790	<price>	<price>	<price>
7416	Virtual Machines	A0 VM	US West 2	Compute Hours	66fe442d-3ec3-4bf3-89d5-943124d4aeb3	<price>	<price>	<price>
7417	Virtual Machines	A0 VM	US West Central	Compute Hours	4bdf6b62-85a8-4241-9b92-ce226b2c5f23	<price>	<price>	<price>
7418	Virtual Machines	A0 VM	USGov	Compute Hours	23d23bb4-1cef-4bf8-b3ec-f0a15c831182	<price>	<price>	<price>
7419	Virtual Machines	A0 VM (Windows)	AP East	Compute Hours	795420fb-100a-4a39-bff2-9f45d4aee1b8	<price>	<price>	<price>
7420	Virtual Machines	A0 VM (Windows)	AP Southeast	Compute Hours	750d52f8-b415-48de-b41d-56199cc4ca91	<price>	<price>	<price>
7421	Virtual Machines	A0 VM (Windows)	AU East	Compute Hours	0e59ad56-03e5-4c3d-90d4-6670874d7e29	<price>	<price>	<price>
7422	Virtual Machines	A0 VM (Windows)	AU Southeast	Compute Hours	d67b3fc7-7f2f-4b81-a6b9-c04c830883a8	<price>	<price>	<price>
7423	Virtual Machines	A0 VM (Windows)	BR South	Compute Hours	5f08fa38-ace4-4e75-b13f-ff37a34e0541	<price>	<price>	<price>
7424	Virtual Machines	A0 VM (Windows)	CA Central	Compute Hours	7e419ac4-8d7b-468a-9f14-4da5289d7770	<price>	<price>	<price>
7425	Virtual Machines	A0 VM (Windows)	CA East	Compute Hours	5166f630-3883-470b-81ef-9d77b25b964f	<price>	<price>	<price>
7426	Virtual Machines	A0 VM (Windows)	EU North	Compute Hours	9743abf5-55ee-432f-8799-60d975d11121	<price>	<price>	<price>
7427	Virtual Machines	A0 VM (Windows)	EU West	Compute Hours	378ae26e-4328-424a-a19b-d7f9cf39610e	<price>	<price>	<price>
7428	Virtual Machines	A0 VM (Windows)	IN Central	Compute Hours	2493e198-3750-445c-8e83-d9de68e8029d	<price>	<price>	<price>
7429	Virtual Machines	A0 VM (Windows)	IN South	Compute Hours	3f3ca9cb-3ccd-4e0e-a619-c796c1b9edff	<price>	<price>	<price>
7430	Virtual Machines	A0 VM (Windows)	IN West	Compute Hours	9f8f5c87-885a-4f66-87ba-12c4cb761548	<price>	<price>	<price>
7431	Virtual Machines	A0 VM (Windows)	JA East	Compute Hours	2f8dec6b-293f-44b6-ab6c-c6fad81904a0	<price>	<price>	<price>

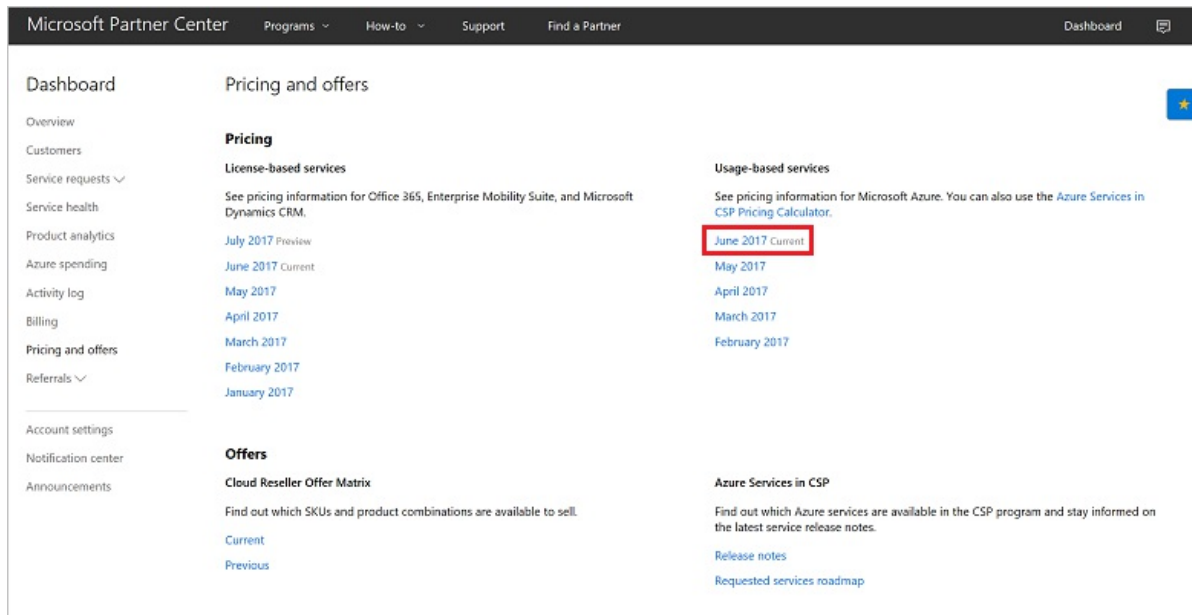
Download the Azure CSP price list

1. On the Partner Center **Dashboard** menu, select **Pricing and offers**.

The screenshot shows the Microsoft Partner Center Dashboard. The top navigation bar includes 'Microsoft Partner Center', 'Programs', 'How-to', 'Support', 'Find a Partner', and 'Dashboard'. The main content area is divided into several sections:

- Dashboard**: A sidebar menu on the left with items like Overview, Customers, Service requests, Service health, Product analytics, Azure spending, Activity log, Billing, **Pricing and offers** (highlighted with a red box), Referrals, Account settings, Notification center, and Announcements.
- Welcome, Admin!**: A greeting message.
- Current tasks**: A section showing 'All current tasks complete'.
- Billing**: A section with the message 'You don't have any bills yet; check back after July 07.' and a link 'Why can't I see my latest payment?'.
- Quick links**: A section with icons and links for 'Partner search', 'Partner account', and 'Helpful resources'.

2. On the **Pricing and offers** page, under **Usage-based services**, select the monthly price list marked **Current**.

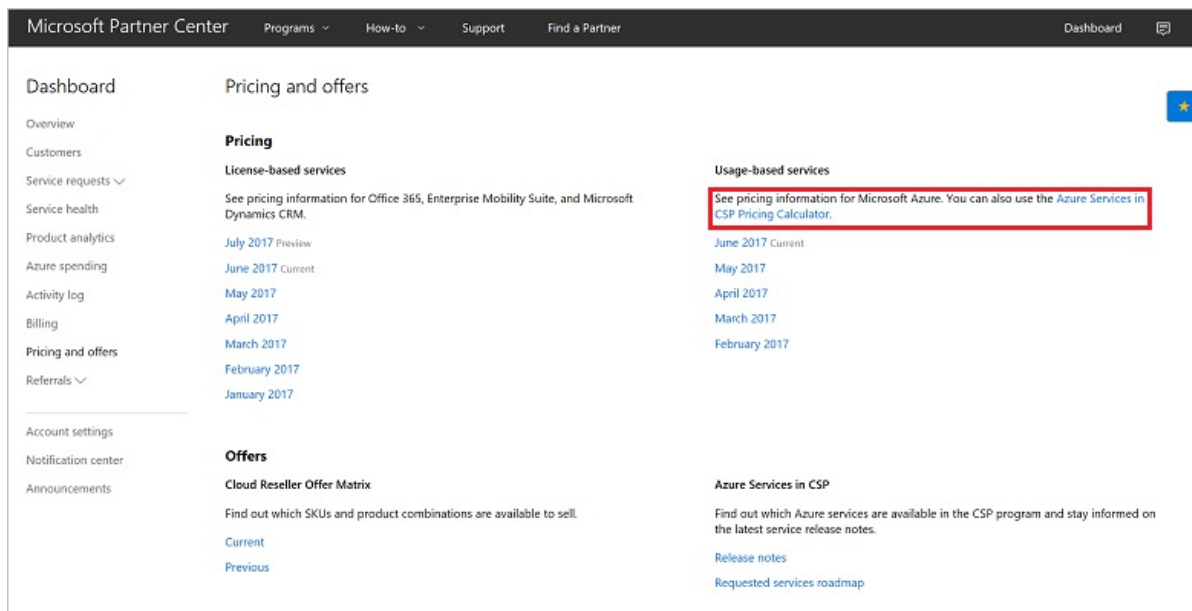


Selecting the **Current** monthly price list downloads a .zip file. The .zip file includes an Excel file that lists all current Azure prices. The first worksheet tab of the Excel file describes how to work with the price list.

CSP Pricing Calculator

Another tool that's available is an automated tool that can help you estimate the monthly cost of your Azure services, based on the most up-to-date price list.

1. On the **Pricing and offers** page, under **Usage-based services**, select the **Azure Services in CSP Pricing Calculator** link.



You also can access the pricing calculator by using a direct URL for the [Azure Channel Pricing Calculator](#) (requires sign-in by using Azure CSP partner credentials).

2. To use the pricing calculator, select the **+Add items** button.

Azure Channel Pricing Calculator PREVIEW

+ Add items
Clear all

Name: Your estimate

Commonly purchased together (Click to add to cart)

VPN Gateway
 SQL Database
 App Service

Web Servers

REGION: West US 2
TYPE: Windows
☐ ADD MANAGED DISKS

PRICING TIER: Standard

NOTE: Save up to 40 percent on Windows Server virtual machines using your existing licenses with Software Assurance. [Learn more](#)

INSTANCE SIZE: A2m v2: 2 cores, 16 GB RAM, 20 GB disk, <price>/hr

2
Virtual Machines

744
Hours

= <price>/MO

SKU: F2D0E4D6-CC5F-44CD-BCE0-2198D7B00225

Your estimate

US Dollar (\$)

Web Servers <cost>
Support Options \$0.00

SUB-TOTAL <cost>

DISCOUNTS AND MARKUPS -

CSP

Partner Markup

20 %

<cost>

ESTIMATED MONTHLY COST

<total cost>

Save

Show my estimates

Prices are estimates and are not intended as actual price quotes.

- On the **Products** menu, select the services that you want a price estimate for. Then, select the **X** in the upper-right corner.

Calculator My Estimates

- Add items
Clear all

Products

Featured
Compute
Networking
Storage
Web + Mobile
Databases
Data + Analytics
Internet of Things
Enterprise Integration
Security + Identity
Developer Tools
Monitoring + Management

App Service
Quickly create powerful cloud apps for web and mobile
 Azure Cosmos DB
Try Azure Cosmos DB for a globally distributed, multi-model database
 HDInsight
Provision cloud Hadoop, Spark, R Server, HBase, and Storm clusters
 SQL Database
Managed relational SQL Database as a service
 Storage
Durable, highly-available, and massively-scalable cloud storage
 Virtual Machines
Provision Windows and Linux virtual machines in seconds

1
Virtual Machines

744
Hours

= \$88.54/MO

SKU: 62C64B6C-4033-4E20-A833-9E81271AC12A

Prices are estimates and are not intended as actual price quotes.

- On the next page, you can select options to fine-tune the estimate for your Azure services usage. You can select quantity of resources, region, time of use, and any discounts or markups you might be eligible for. Under **Estimated Monthly Cost**, you can see your total estimated monthly cost.

To retain the estimate for later use, select **Save**.

Azure Channel Pricing Calculator

PREVIEW

+ Add items

Clear all

Name: Your estimate

Commonly purchased together (Click to add to cart)

VPN Gateway

SQL Database

App Service

Web Servers

REGION:

West US 2

TYPE:

Windows

ADD MANAGED DISKS

PRICING TIER:

Standard

NOTE: Save up to 40 percent on Windows Server virtual machines using your existing licenses with Software Assurance. [Learn more](#)

INSTANCE SIZE:

A2m v2: 2 cores, 16 GB RAM, 20 GB disk, <price>/hr

2

×

744

=

<price>/MO

Virtual Machines

Hours

SKU F2D0E4D6-CC5F-44CD-BCE0-2198D7B00225

Your estimate

US Dollar (\$)

Web Servers <cost>

Support Options \$0.00

SUB-TOTAL <cost>

DISCOUNTS AND MARKUPS -

CSP

Partner Markup 20 % <cost>

ESTIMATED MONTHLY COST <total cost>

Save

Show my estimates

Prices are estimates and are not intended as actual price quotes.

Get prices by using the Azure Rate Card

You also can use Azure Rate Card to get Azure service rates, without going through Partner Center portal. Azure pricing is quite dynamic and changes frequently. Azure CSP price list is published on Partner Center portal (go to the Dashboard, then select Sales), but this REST API provides the fastest way for Cloud Solution Provider partners to get current prices.

IMPORTANT

You can't use native [Azure Resource RateCard API](#) to get prices for Azure CSP Subscription. Azure Rate Card should be accessed through [Partner Center API](#) instead.

Request syntax

METHOD	REQUEST URI
GET	{baseUrl}/v1/ratecards/azure¤cy={currency}®ion={region}

URI parameters

NAME	TYPE	DESCRIPTION
------	------	-------------

NAME	TYPE	DESCRIPTION
currency	string	Optional three letter ISO code for the currency in which the resource rates will be provided (e.g. "EUR"). The default is the currency associated with the market in the partner profile.
region	string	Optional two-letter ISO country/region code that indicates the market where the offer is purchased (e.g. "FR"). The default is the country/region code set in the partner profile.

No request body is required. If this is successful, it returns an Azure Rate Card resource.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/ratecards/azure HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: 07ced227-3f32-4eeb-8062-f0bef849a9bc
MS-CorrelationId: a687bc47-8d08-4b78-aff6-5a59aa2055c2
X-Locale: en-US
Host: api.partnercenter.microsoft.com
Connection: Keep-Alive
```

Response example

HTTP/1.1 200 OK
Content-Length: 1545508
Content-Type: application/json; charset=utf-8
MS-CorrelationId: 57b25659-fc00-4215-87e7-2b09bac6845d
MS-RequestId: 870118d0-adbb-41a3-82d2-a3d45ade3c73
MS-CV: CYBB8PXMSEukJBIn.0
MS-ServerId: 201021413
Date: Wed, 01 Feb 2017 00:13:45 GMT

```
{
  "locale": "en-US",
  "currency": "USD",
  "isTaxIncluded": false,
  "meters": [{
    "id": "4b836326-7e19-46e6-8bce-1b19bb6cd91e",
    "name": "Unlimited Data - 1 Gbps",
    "rates": {
      "0": 7395.0
    },
    "tags": [],
    "category": "Networking",
    "subcategory": "ExpressRoute",
    "region": "Zone 2",
    "unit": "Connections",
    "includedQuantity": 0.0,
    "effectiveDate": "2015-09-01T00:00:00Z"
  }, {
    "id": "1e8f6d9f-8b40-4c97-80cc-cff87a290a93",
    "name": "Compute Hours",
    "rates": {
      "0": 3.9729
    },
    "tags": [],
    "category": "Cloud Services",
    "subcategory": "Standard_L16 Cloud Services",
    "region": "AU East",
    "unit": "1 Hour",
    "includedQuantity": 0.0,
    "effectiveDate": "2016-09-01T00:00:00Z"
  }, {
    "id": "7a2639ce-ae47-4413-9837-6b4f4b78be3d",
    "name": "Compute Hours",
    "rates": {
      "0": 0.1122
    },
    "tags": [],
    "category": "Virtual Machines",
    "subcategory": "Standard_D1_v2 VM (Windows)",
    "region": "BR South",
    "unit": "Hours",
    "includedQuantity": 0.0,
    "effectiveDate": "2017-01-01T00:00:00Z"
  }
],
  "offerTerms": [{
    "name": "Overage discount",
    "discount": 0.15,
    "excludedMeterIds": ["53cc0061-0fe2-4249-bf62-e1008c811f5c", "c82dbd27-c978-43a7-ad41-525a90d8962b"],
    "effectiveDate": "2014-01-01T00:00:00"
  }
],
  "attributes": {
    "objectType": "AzureRateCard"
  }
}
```


Next steps

- Learn [how Azure CSP billing works](#).
- Learn [how to analyze an Azure CSP invoice](#).
- Learn [how to access Azure Rate Card through Partner Center API](#).

How to read your monthly Azure CSP invoice

9/22/2017 • 8 min to read • [Edit Online](#)

Every month, you will receive an invoice from Microsoft detailing the total charges from all subscriptions of all Azure Cloud Solution Provider (Azure CSP) customers you manage. Because all of your customers' subscriptions are on one bill, sometimes you might need to determine where exactly a charge came from. There are a few ways to do this.

Partner Center provides two different reconciliation files: one for license-based services (such as Office 365, Enterprise Mobility + Security, or Dynamics 365), and another for usage-based services (such as Azure). These files are detailed, line-item breakdowns of all the usage of all customers during the last billing cycle.

TIP

You can see examples of these files: [Azure CSP invoice](#) and [Usage-based reconciliation file](#).

This guide covers how to:

- Read and interpret your monthly invoice.
- Compare that information to the reconciliation file to determine the source of a charge.
- Get more detailed usage data from a specific subscription.

Compare the invoice and the reconciliation file

View the invoice in the **Partner Center Dashboard**, by selecting **Billing** in the left pane.

Billing

June bill

Account balance as of June 07, 2017

\$ XXXX.XX

[Invoice \(.pdf\)](#)

[Reconciliation \(.csv\)](#) ✓

\$ XXXX.XX payment received. Thank you!

[Why can't I see my latest payment?](#)

The following table explains the different columns shown on your bill.

COLUMN	DESCRIPTION
US FEIN	Your federal tax ID number.
Customer number	Your customer number.
Bill to	Address that receives the invoice. To change this address, edit your Partner Center account profile .
Recurring charges	The flat charges for the purchased usage-based licenses, billed in advance of the service.
Usage charges	Azure usage, including new services or applications enabled and used during the billing month.
Credits & adjustments	Credits or adjustments made to the subscriptions (such as seat increases or decreases).
Other discounts	For example, a discount the customer receives from the normal price of a subscription.
Taxes	Tax for the current charges, as totaled in the details section beginning on page 2 of the invoice.
Total current charges	Amount due for the billing period, due by the payment due date.
Payment instructions	How to pay the invoice, based on region. Always include the invoice number when you are making your payment.
Invoice no.	Invoice number to be provided upon payment.
Billing period	Applicable dates for this payment. Azure CSP partners are billed monthly.
Invoice date	Date the invoice was received.
Payment due date	Date that payment must be received.
Customer PO	Purchase order number.
Customer service	Website address to access customer service.
Service recipient	The address where the service is used. (This is the legal company address associated with company vetting, and it cannot be changed.)

The reconciliation file

You can also view the reconciliation file in the **Partner Center Dashboard**, by selecting **Billing** in the left pane.

Billing

June bill

Account balance as of June 07, 2017

\$ XXXX.XX

[Invoice \(.pdf\)](#)

[Reconciliation \(.csv\) ▾](#)

\$ XXXX.XX payment received. Thank you!

[Why can't I see my latest payment?](#)

The following table explains the different columns in the reconciliation file.

COLUMN	DESCRIPTION	EXAMPLE VALUE
PartnerId	Partner ID, in GUID format.	DA41BC5F-C52D-4464-8A8D-8C8DCC43503B
PartnerName	Partner name.	Acme Incorporated
PartnerBillableAccountID	Partner account ID.	1010578050
CustomerName	Customer's organization name as reported in Partner Center. This is very important for reconciling the invoice with your system information.	Test Customer A
MPNID	MPN ID of the Azure CSP partner.	4390934
ResellerMPNID	MPN ID of the reseller of record for the subscription, if applicable.	4390934
InvoiceNumber	Invoice number where the specified transaction appears.	D020001IVK
ChargeStartDate	Start date of billing cycle, except when presenting dates of previously uncharged latent usage data (from previous bill cycle). The time is always 0:00.	2/1/2014 0:00
ChargeEndDate	End date of billing cycle, except when presenting dates of previously uncharged latent usage data (from previous bill cycle). The time is always 23:59.	2/28/2014 23:59

COLUMN	DESCRIPTION	EXAMPLE VALUE
SubscriptionID	Unique identifier for a subscription in the Microsoft billing platform. Might be useful to identify the subscription when you are contacting support, but not for reconciliation. This is not the same as the subscription ID on the Partner Admin Console.	usCBMgAAAAAAAAAIA
SubscriptionName	Nickname of the service offering.	Microsoft Azure
SubscriptionDescription	Line of business of the service offering.	Microsoft Azure
OrderID	Unique identifier for an order in the Microsoft billing platform. Might be useful to identify the subscription when you are contacting support, but not for reconciliation.	566890604832738111
ServiceName	Name of the Azure service in question.	VIRTUAL MACHINES
ServiceType	Specific type of Azure service.	<ul style="list-style-type: none"> • Service Bus: Individual or Pack • SQL Azure database: Business or Web Edition
ResourceGUID	Specific unique identifier for all the service data and pricing structure.	DA41BC5F-C52D-4464-8A8D-8C8DCC43503B
Resource Name	Name of the Azure resource.	<ul style="list-style-type: none"> • Data Transfer In(GB) • Data Transfer Out (GB)
SKU	Microsoft unique identifier for the offer.	7UD-00001
DetailLineItemId	ID and quantity for itemizing the different rates for a service or resource in a particular billing period. For Azure tiered rating, there might be one rate up to a certain quantity of billable units, and then a different rate after that.	1
ConsumedQuantity	Amount of service consumed (for example, hours or GB) during the reporting period. Also includes any unbilled usage from previous reporting periods.	11
IncludedQuantity	Units included as part of the offer. Not typically present in Azure CSP.	0
OverageQuantity	Units not included as part of the offer, that the partner must pay for. Equal to the ConsumedQuantity minus the IncludedQuantity.	11

COLUMN	DESCRIPTION	EXAMPLE VALUE
ListPrice	Offer price in effect at the subscription start date.	\$0.0808
PretaxCharges	ListPrice times OverageQuantity, rounded to the nearest cent.	\$0.85
TaxAmount	Tax amount charge, based on your market's tax rules and specific circumstances.	\$0.08
PostTaxTotal	Total after tax, when tax is applicable.	\$0.93
Currency	Currency type. Each billing entity has only one currency. Check that it matches your first invoice, and then after any major billing platform update.	EUR
PretaxEffectiveRate	Pretax price per unit. Equal to PretaxCharges / OverageQuantity, rounded to the nearest cent.	\$0.08
PostTaxEffectRate	Post tax price per unit. Equal to PostTaxTotal / OverageQuantity, or PretaxEffectiveRate + tax rate per unit amount, rounded to the nearest cent.	\$0.08
ChargeType	Type of charge or adjustment.	See "Mapping Charges between invoice and reconciliation file", below
CustomerBillableAccount	Unique account ID in the Microsoft billing platform.	1280018095
UsageDate	Date of service deployment.	2/1/2014 0:00
MeteredRegion	Location of a datacenter within the region for services where this is applicable and populated.	East Asia, South East Asia, North Europe, West Europe, North Central US, South Central US
MeteredService	Tracks individual Microsoft Azure services that might not be specifically identified in the Service Name column. For example, data transfers are reported as "Microsoft Azure - All Services" in the Service Name column. This MeteredService column indicates a specific service.	AccessControl, CDN, Compute, Database, ServiceBus, Storage
Project	Customer-defined name for their service instance.	ORDDC52E52FDEF405786F0642DD0108BE4

COLUMN	DESCRIPTION	EXAMPLE VALUE
ServiceInfo	Number of Service Bus connections that were provisioned and utilized on a specific day.	For an individually provisioned connection during a month of 30 days, Service Info 1 would read "1.000000 Connections / 30 days". For a 25 pack of Service Bus connections provisioned that had utilized 1 during that day, the daily usage statement for that day would indicate "25 Connections / 30 Days – Used: 1.000000".
CustomerID	Unique Microsoft ID, in GUID format, used to identify the customer.	ORDDC52E52FDEF405786F0642DD0108BE4
DomainName	Customer's domain name, used to help identify the customer.	example.onmicrosoft.com

Map charges between the invoice and the reconciliation file

The invoice provides a summary of charges, whereas the reconciliation file provides a detailed breakdown of line-item transactions. To cross-reference amounts between the two documents, map the invoice charges to the applicable charge breakdowns on the reconciliation file. You can use Excel to do this.

The following table shows invoice sections and their associated charge types on the reconciliation files.

INVOICE CHARGE DESCRIPTION	RECONCILIATION FILE CHARGE DESCRIPTION (CHARGETYPE COLUMN)	WHAT IS THIS CHARGE?	HOW DO I MAP THESE CHARGETYPES TO THE INVOICE?
Usage Charges	Access usage fee when canceled.	Access usage fee upon cancellation for unpaid usage during the current billing period.	Sum the PretaxCharges column.
	Access usage fee for current cycle.	Access usage fee for the current billing period.	
Credits & Adjustment	Offset a line item.	Partial or whole refund to a line item, including taxes.	Sum the PostTaxTotal column.
Other Discounts	Activation discount.	Discount applied when the subscription is activated.	Sum the PretaxCharges column.
	Cycle discount.	Discount applied on periodic charges.	
	Renew discount.	Discount applied when the subscription is renewed.	
	Cancel discount.	Charges applied when discounts are canceled.	

INVOICE CHARGE DESCRIPTION	RECONCILIATION FILE CHARGE DESCRIPTION (CHARGETYPE COLUMN)	WHAT IS THIS CHARGE?	HOW DO I MAP THESE CHARGETYPES TO THE INVOICE?
Taxes or VAT	Can be applied to multiple charge types. Exception: "Offset a line item" already includes taxes. See the preceding row, Credits & Adjustments .	Taxes or value-added taxes (VAT).	Sum the TaxAmount column.

Retrieve usage data for a specific subscription

If you want to see the usage data of one particular subscription, you can retrieve that information through the Partner Center portal web UI, or through Partner Center APIs. You can use PowerShell, C#, or REST API to retrieve the information from Partner Center.

IMPORTANT

You can't use native [Azure Resource Usage API](#) to get usage for Azure CSP Subscription. Resource Usage should be accessed through [Partner Center API](#) instead.

Web UI

1. From the **Dashboard**, go to **Customers** > select the specific customer > **Subscriptions** > select the specific subscription.

The screenshot shows the Microsoft Partner Center interface. The top navigation bar includes the Microsoft logo, categories like Cloud, Mobility, and Productivity, a search bar, and a user profile. The main header shows 'Microsoft Partner Center' and navigation links like Programs, How-to, Support, and Find a Partner. The left sidebar lists navigation options: Subscriptions, Customer insights, Users and licenses, Service management, and Account. The main content area is titled 'Subscriptions' and includes an 'Add subscription' button. It is divided into two sections: 'License-based subscriptions' and 'Usage-based subscriptions'. The 'License-based subscriptions' section shows a table with one entry: 'Azure Active Directory Basic' with a quantity of '1 Licenses' and a status of 'Auto renews on 1/6/18'. The 'Usage-based subscriptions' section shows a table with two entries: 'Microsoft Azure' and 'CSP Tool', both with a 'Current estimate' of '0%' and a status of 'Active'. The 'Microsoft Azure' entry is highlighted with a red box. The bottom of the page features a footer with links for Industries, For customers, For partners, For developers, Values, and Company.

2. The estimated usage of the subscription appears at the bottom of the screen under **Estimated Usage**.

Itemized costs <small>Report for 6/22/17 7:44 AM</small>	
Item	Amount used
Networking - Data Transfer Out (GB)	43,001.36 GB
Data Services - SQL Data Warehouse - DWU	892.00 Hours
Virtual Machines - A4 VM (Windows) - Compute Hours	1,157.32 Hours
Data Services - SQL Database - Standard S2 Database Days	111.17 Days
Virtual Machines - BASIC,A4 VM (Windows) - Compute Hours	355.97 Hours
Virtual Machines - A4 VM (Windows) - Compute Hours	259.62 Hours
Virtual Machines - BASIC,A3 VM - Compute Hours	906.17 Hours
Storage - Locally Redundant - Premium Storage - Page Blob/P10 (Units)	8.61 Units
Virtual Machines - A4 VM - Compute Hours	236.72 Hours
Virtual Machines - BASIC,A4 VM - Compute Hours	279.75 Hours
< First < Previous Next Last >	

PowerShell

To receive a specific subscription's usage data in PowerShell, enter the following commands:

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'

Get-PCSubscriptionMonthlyUsageRecords -tenantid $customer.id
```

C#

To get a subscription's resource usage information, use your **IAggregatePartner.Customers** collection and call the **ById()** method. Then call the **Subscriptions** property, as well as **UsageRecords**. Then call the **Resources** property, and finish by calling the **Get()** or **GetAsync()** method.

```
// IAggregatePartner partnerOperations;
// var selectedCustomerId as string;
// var selectedSubscriptionID as string;

var usageRecords =
partnerOperations.Customers.ById(selectedCustomerId).Subscriptions.ById(selectedSubscriptionId).UsageRecords.Resources.Get();
```

REST API

Request syntax

METHOD	REQUEST URI
GET	<i>{baseUrl}/v1/customers/{customer-tenant-id}/subscriptions/{id-for-subscription}/usagerecords/resources</i> HTTP/1.1

URI parameter

The following table lists the required query parameters to get the rated usage information.

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	A GUID that corresponds to the customer.

NAME	TYPE	DESCRIPTION
id-for-subscription	guid	A GUID that corresponds to the subscription.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers/{customer-tenant-id}/subscriptions/{id-for-subscription}/usagerecords/resources
HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: 65b26053-37d0-4303-9fd1-46ad8012bcb6
MS-CorrelationId: 47c36033-af5d-4457-80a4-512c1626fac4
```

If successful, this method returns a collection of **AzureResourceMonthlyUsageRecord** resources in the response body.

Response example

HTTP/1.1 200 OK
Content-Length: 12014
Content-Type: application/json
MS-CorrelationId: 648a26a4-a63e-459f-844b-4f29d7913353
MS-RequestId: be82a8ba-4a53-49f7-8313-b033c058687e
Date: Tue, 10 Nov 2015 19:09:59 GMT

```
{
  "totalCount":20,
  "items":[{
    "category":"Storage",
    "subcategory":"LOCALLY REDUNDANT",
    "quantityUsed":0.151287527825352,
    "unit":"GB",
    "id":"2a2419c0-cefe-46b2-8004-8eb002ad606c",
    "name":"Azure Resource 1",
    "totalCost":0.195779159290613,
    "currencyLocale":"en-US",
    "attributes":{
      "objectType":"AzureResourceMonthlyUsageRecord"
    }
  },
  {
    "category":"Remote App",
    "subcategory":"Remote App",
    "quantityUsed":0.932546524299563,
    "unit":"GB",
    "id":"7e4099c8-2b3d-41a6-a1bd-d5cf315989b2",
    "name":"Azure Resource 2",
    "totalCost":0.920983775016379,
    "currencyLocale":"en-US",
    "attributes":{
      "objectType":"AzureResourceMonthlyUsageRecord"
    }
  }
],
  "links":{
    "self":{
      "uri":"/v1/customers/<customer-tenant-id>/subscriptions/<id-for-subscription>/%20/usagerecords",
      "method":"GET",
      "headers":[]
    }
  },
  "attributes":{
    "objectType":"Collection"
  }
}
```

Next steps

- [Learn](#) how Azure CSP billing works.
- [Learn](#) how to work with the Azure CSP price list.
- [Learn](#) how to integrate with Azure CSP billing.

Azure CSP API integration options

9/22/2017 • 4 min to read • [Edit Online](#)

This article provides an overview of the available APIs that are used to integrate the Azure Cloud Solution Provider (Azure CSP) system with already-existing systems. The systems might manage customer accounts, place orders, manage subscriptions, or handle support requests.

Partner Center APIs

With the use of these APIs, you can perform just about any task in a script-based format without having to touch the Partner Center UI directly. They interact with the Partner Center interface, not Azure directly.

Partner Center C# SDK

The Partner Center managed API helps Azure CSP partners integrate their existing systems with Microsoft systems to perform functions in Partner Center. This essentially allows you to merge Partner Center features into your own applications by using C#.

C# is the most feature-rich of the available APIs, supporting just about any function of Partner Center through C#. It also includes token management, which eliminates the need to refresh an Azure AD token and authentication each hour. There is a simple interface library for network calls.

For more information, [download the C# SDK](#), or review the [API reference documentation](#).

REST API

The Partner Center REST API helps Azure CSP partners integrate their existing systems with Partner Center by using common REST API mechanics. By submitting REST requests to the proper endpoints, you can perform many of the Partner Center functions through this API.

The major advantage of this API is that it is highly compatible. If your internal system doesn't work with C#, REST can perform many of the same functions.

For more information, see the [REST API reference](#), and [download the Swagger document](#).

IMPORTANT

[CREST API](#) - predecessor of the Partner Center REST API - is not available to new CSP partners and will be deprecated on October 31st, 2017. We recommend you update your apps and services to support the Partner Center REST APIs as soon as possible.

PowerShell

Use Partner Center PowerShell Module to implement the available scenarios through PowerShell. Scenarios include management of customers, offers, subscriptions, and usage.

IMPORTANT

Partner Center PowerShell Module is an open source project, and it is not supported by Microsoft.

Use PowerShell to write scripts by using short user-friendly commands to perform many of the most common Partner Center scenarios. However, Partner Center PowerShell Module was created outside the Partner Center engineering team, and it does not have as many available functions as native APIs. It can still cover the most common scenarios, so it is still something that partners should know going in.

For more information, download the [Azure PowerShell Module](#), and learn how to [use PowerShell for Azure CSP purposes](#).

Azure APIs

The following APIs bypass Partner Center completely and interact directly with Azure.

Resource Manager API

Azure Resource Manager is the deployment model, replacing Azure Service Management (ASM). Although you can perform many functions of Resource Manager through the portal UI, these functions can also be performed automatically through the Resource Manager API. These functions include, but are not limited to, configuring Azure Virtual Machines and other Azure services, generating reports for debugging and other tasks, and assigning policies inside Azure subscriptions.

The infrastructure of an application may be made up of many components: virtual machines, storage accounts, web applications, or third-party services, just to name a few. These are separate components, but are frequently seen as parts of a single entity that should be deployed, managed, and monitored as a group.

With Resource Manager, you can work with the resources in your solution as a group. You can deploy, update, or delete all the resources for your solution in a single, coordinated operation. You use a template for deployment. The template can work for different environments such as testing, staging, and production. Resource Manager provides security, auditing, and tagging features to help you manage your resources after deployment.

For more information about Resource Manager, see [the Resource Manager documentation](#).

Resource Manager uses Azure SDKs in multiple languages and platforms to allow for control through whatever platform works best for your solution. Each SDK is generated from open-source RESTful API specifications. The tools used to create the SDKs are open-source and freely available. The available SDKs are as follows:

NOTE

If the SDK you're investigating does not have the functionality you want, you can call directly to the Azure REST API instead.

- [.NET](#)
- [Java](#)
- [Node.js](#)
- [PHP](#)
- [Python](#)
- [Ruby](#)

For more information about the Resource Manager functions and features, see [the Resource Manager website](#).

IMPORTANT

Native [Azure Billing APIs](#) like RateCard API and Resource Usage API can't be used because Microsoft.Commerce provider is not available for Azure CSP Subscriptions. Similar [Partner Center APIs](#) should be used instead.

Graph API

With the Microsoft Graph API, you can interact with the data of millions of users in the Microsoft cloud. Use it

to connect to a wealth of resources, relationships, and intelligence, all through a single endpoint.

Microsoft Graph is a network of resources and the relationships between those resources. By defining the relationships, you can connect resources and perform actions through the Graph API.

These relationships also bring valuable insights to your solution. For example, you can see the subscriptions that a user can access, or get a list of the people who are most relevant to a user.

To learn more about the Graph API, [see the developer documentation](#).

Next steps

- [Review](#) the list of available Azure CSP Integration scenarios.
- [Review](#) management options for Azure CSP.
- [Review](#) how billing works in the Azure CSP model.

Azure Cloud Solution Provider API scenario index

10/6/2017 • 1 min to read • [Edit Online](#)

This is a catalog of API integration scenarios for tasks related to Azure Cloud Solution Provider (Azure CSP) in Microsoft Partner Center. The linked articles might describe tasks and actions by using the Partner Center web UI, the C# SDK, PowerShell, or the REST API.

Manage billing

- [Work with the Azure CSP price list](#)
- [Work with Azure CSP invoices](#)
- [Get prices for an Azure service](#)
- [Get utilization records for a customer](#)
- [Get a customer invoice](#)
- [Get a collection of invoices](#)
- [Get invoice line items](#)
- [Get a service cost summary](#)
- [Get service cost line items](#)
- [Get customer usage data](#)
- [Get a customer usage summary](#)
- [Update a customer's billing info](#)

Manage customers

- [Create a new Azure CSP customer](#)
- [Add an existing Azure customer to Azure CSP](#)
- [Add an Azure CSP subscription](#)
- [Get customer subscriptions by ID](#)
- [Update a subscription "nickname"](#)
- [Assign permissions to an Azure CSP subscription](#)
- [Create an order for an Azure CSP indirect reseller](#)
- [Manage user accounts](#)
- [Get customer profiles](#)
- [Get customer orders](#)
- [Search for customers](#)
- [Find a user by ID](#)
- [Get a list of all a customer's users](#)
- [Get a list of all user roles for a customer](#)
- [Manage user permissions](#)
- [Reset a user password](#)
- [View deleted users](#)

Manage orders

- [Place an order](#)
- [Place an order for an indirect reseller](#)

- [Find an offer by ID](#)
- [Find an order by ID](#)
- [Get a subscription's provisioning status](#)
- [Get a list of subscriptions by order](#)
- [Suspend or reactivate a subscription](#)

Manage accounts and profiles

- [Create Partner Center user accounts](#)
- [Restore a suspended Partner Center account](#)
- [Update partner profiles](#)
- [Get a list of Azure CSP indirect resellers](#)
- [Get and verify a partner's Microsoft Partner Network \(MPN\) profile](#)
- [Get a customer's subscriptions by partner MPN ID](#)

Provide support

- [Get a list of managed services](#)
- [Get a list of service request support topics](#)
- [Create, update, and retrieve service requests](#)

Utilities

- [Delete a customer account in the integration sandbox](#)
- [Get a record of Partner Center activity by user](#)
- [Get address formatting rules by market](#)
- [Validate a customer's address](#)
- [Verify a customer's domain availability](#)

Next steps

- Learn about [additional integration scenarios using Partner Center SDK](#).
- Learn about [Partner Center capabilities for Azure CSP](#).
- Learn about [management options for Azure CSP](#).

Get pricing for Microsoft Azure

9/22/2017 • 2 min to read • [Edit Online](#)

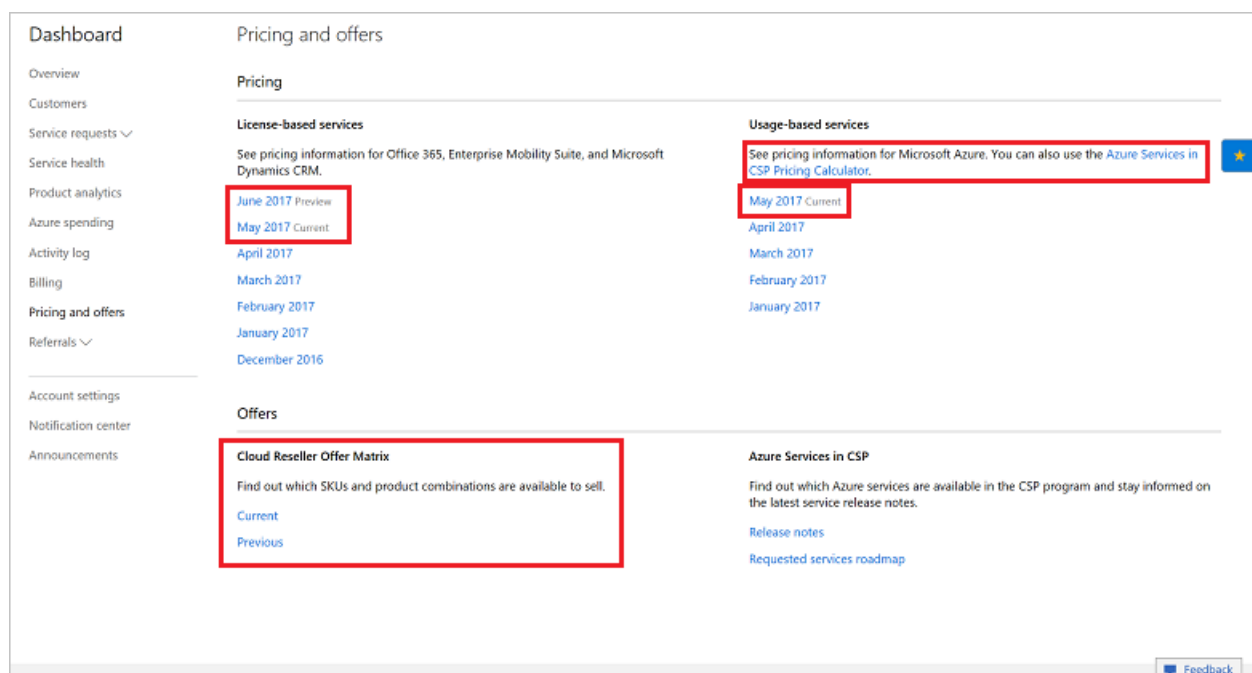
This guide covers how to get an Azure rate card with real-time pricing for an Azure offer.

Azure pricing is dynamic, and it changes frequently. Prices also differ by market and currency. The Azure rate card takes these facts into consideration and predicts the monthly bill for the partner and customers.

Web UI

In the Microsoft Partner Center **Dashboard**, in the left pane, select **Pricing and offers**.

On the **Pricing and offers** page, you can view the pricing for license-based and usage-based services, access the **Azure Services in CSP** pricing calculator, and view current offers.



For more information, see the [Azure Cloud Solution Provider \(CSP\) price list](#).

PowerShell

To retrieve an Azure rate card by using PowerShell, run the following commands. The command to run differs depending on whether you want a rate card for the default context or a specific currency and region.

Default context

```
Get-PCAzureRateCard'
```

Specific currency and region

```
Get-PCAzureRateCard -currency '<three digits currency>' -region '<two digits region code>'
```

C#

To obtain the Azure rate card, call the **IAzureRateCard.Get** method to return an **AzureRateCard** resource that contains the Azure prices.

```
IPartner partner = PartnerService.Instance.CreatePartnerOperations(credentials);  
var azureRateCard = partner.RateCards.Azure.Get();
```

REST API

IMPORTANT

[Azure Resource RateCard API](#) is not available for Azure CSP subscriptions. Azure Rate Card should be accessed through [Partner Center API](#) instead.

Request

Request syntax

METHOD	REQUEST URI
GET	<code>{baseUrl}/v1/ratecards/azure&currency={currency}&region={region}</code>

URI parameters

You can use these parameters to retrieve a specific currency or region. The parameters are *optional*. If you leave them blank, they default to the currency or country that's set in the partner profile.

NAME	TYPE	DESCRIPTION
currency	string	The three-letter ISO code for the currency in which the resource rates are provided.
region	string	The two-letter ISO country/region code that indicates the market where the offer is purchased.

If you include the optional **X-Locale** header in the request, it determines the language that's used for the response details.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/ratecards/azure HTTP/1.1  
Authorization: Bearer <token>  
Accept: application/json  
MS-RequestId: 07ced227-3f32-4eeb-8062-f0bef849a9bc  
MS-CorrelationId: a687bc47-8d08-4b78-aff6-5a59aa2055c2  
X-Locale: en-US  
Host: api.partnercenter.microsoft.com  
Connection: Keep-Alive
```

Response

If the request is successful, it returns an Azure rate card resource.

HTTP/1.1 200 OK
Content-Length: 1545508
Content-Type: application/json; charset=utf-8
MS-CorrelationId: 57b25659-fc00-4215-87e7-2b09bac6845d
MS-RequestId: 870118d0-adbb-41a3-82d2-a3d45ade3c73
MS-CV: CYBB8PXMsEukJBI.0
MS-ServerId: 201021413
Date: Wed, 01 Feb 2017 00:13:45 GMT

```
{
  "locale": "en-US",
  "currency": "USD",
  "isTaxIncluded": false,
  "meters": [{
    "id": "4b836326-7e19-46e6-8bce-1b19bb6cd91e",
    "name": "Unlimited Data - 1 Gbps",
    "rates": {
      "0": 7395.0
    },
    "tags": [],
    "category": "Networking",
    "subcategory": "ExpressRoute",
    "region": "Zone 2",
    "unit": "Connections",
    "includedQuantity": 0.0,
    "effectiveDate": "2015-09-01T00:00:00Z"
  }, {
    "id": "1e8f6d9f-8b40-4c97-80cc-cff87a290a93",
    "name": "Compute Hours",
    "rates": {
      "0": 3.9729
    },
    "tags": [],
    "category": "Cloud Services",
    "subcategory": "Standard_L16 Cloud Services",
    "region": "AU East",
    "unit": "1 Hour",
    "includedQuantity": 0.0,
    "effectiveDate": "2016-09-01T00:00:00Z"
  }, {
    "id": "7a2639ce-ae47-4413-9837-6b4f4b78be3d",
    "name": "Compute Hours",
    "rates": {
      "0": 0.1122
    },
    "tags": [],
    "category": "Virtual Machines",
    "subcategory": "Standard_D1_v2 VM (Windows)",
    "region": "BR South",
    "unit": "Hours",
    "includedQuantity": 0.0,
    "effectiveDate": "2017-01-01T00:00:00Z"
  }
],
  "offerTerms": [{
    "name": "Overage discount",
    "discount": 0.15,
    "excludedMeterIds": ["53cc0061-0fe2-4249-bf62-e1008c811f5c", "c82dbd27-c978-43a7-ad41-525a90d8962b"],
    "effectiveDate": "2014-01-01T00:00:00"
  }
],
  "attributes": {
    "objectType": "AzureRateCard"
  }
}
```

Next steps

- [Review](#) what are the available APIs for Azure CSP integration.
- [Review](#) the list of available Azure CSP Integration scenarios.

Get a customer's utilization records for Azure

9/19/2017 • 2 min to read • [Edit Online](#)

This guide covers how to retrieve the utilization records of a customer's Azure subscription for a specified period.

PowerShell

To retrieve the utilization records of a customer's Azure subscription for a specified period, run the following PowerShell command:

```
Select-PCustomer -tenantid '<tenant id GUID>'

Get-PCSubscription -all

Get-PCUsage -subscriptionid $subscription.id -start_time "01-12-1999 00:00:00" -end_time "31-12-1999 00:00:00" -granularity {daily | hourly} -
show_details <bool> -size <int>
```

C#

To obtain the Azure utilization records, you first need a customer ID and a subscription ID. You then call the **IAzureUtilizationCollection.Query** method to return a **ResourceCollection** that contains the utilization records. Because the resource collection is paged, you must then obtain an Azure utilization record enumerator to traverse the utilization pages.

```
IPartner partner = PartnerService.Instance.CreatePartnerOperations(credentials);

// Retrieve the utilization records for the last year in pages of 100 records.
var utilizationRecords = partner.Customers[customerId].Subscriptions[subscriptionId].Utilization.Azure.Query(
    DateTimeOffset.Now.AddYears(-1),
    DateTimeOffset.Now,
    size: 100);

// Create an Azure utilization enumerator, which helps you traverse the utilization pages.
var utilizationRecordEnumerator = partner.Enumerators.Utilization.Azure.Create(utilizationRecords);
int pageNumber = 1;

while (utilizationRecordEnumerator.Has Value)
{
    // Get the next page.
    utilizationRecordEnumerator.Next();

    //
    // Insert code here to work with this page.
    //
    pageNumber++;
}
```

REST API

Request

Request syntax

METHOD	REQUEST URI
GET	<code>{baseUrl}/v1/customers/{customer-tenant-id}/subscriptions/{subscription-id}/utilizations/azure?start_time={start-time}&end_time={end-time}&granularity={granularity}&show_details={True}</code>

URI parameters

Use the following parameters to get the utilization records.

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	The customer ID.
subscription-id	guid	The subscription ID.
start_time	DateTimeOffset	The start of the time range when the utilization was metered in the billing system.
end_time	DateTimeOffset	The end of the time range when the utilization was metered in the billing system.
granularity	string	<i>Optional.</i> Defines the granularity of usage aggregations. Options are <i>daily</i> and <i>hourly</i> . Defaults to <i>daily</i> .
show_details	bool	<i>Optional.</i> Specifies whether to get the instance-level usage details. Defaults to <i>true</i> .
size	int	<i>Optional.</i> Specifies the number of aggregations returned by a single API call. Defaults to <i>1000</i> , the maximum.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers/65726577-c208-40fd-9735-8c85ac9cac68/subscriptions/87F4B92F-A490-485E-AD34-5B70CBA4AF74/utilizations/azure?start_time=2015-12-02T10:54:34Z&end_time=2016-12-02T10:54:34Z&granularity=Daily&show_details=True&size=100 HTTP/1.1
Authorization: Bearer <authorization token>
Accept: application/json
MS-RequestId: e6a3b6b2-230a-4813-999d-57f883b60d38
MS-CorrelationId: a687bc47-8d08-4b78-aff6-5a59aa2055c2
X-Locale: en-US
Host: api.partnercenter.microsoft.com
```

Response

If the request is successful, this method returns a collection of Azure utilization record resources in the response body.

Response example

```
HTTP / 1.1 200 OK
Content - Length: 51145
Content - Type: application / json
```

Content-Type: application/json,

charset=utf-8

MS - CorrelationId: a687bc47 - 8d08 - 4b78 - aff6 - 5a59aa2055c2

MS - RequestId: e6a3b6b2 - 230a - 4813 - 999d - 57f883b60d38

MS - CV: gPPoyNX1X0asAAcw.0

MS - ServerId: 101112202

Date: Fri, 02 Dec 2016 18: 54: 38 GMT

```
{
  "totalCount": 2,
  "items": [
    {
      "usageStartTime": "2015-11-30T16:00:00-08:00",
      "usageEndTime": "2015-12-01T16:00:00-08:00",
      "resource": {
        "id": "505db374-df8a-44df-9d8c-13c14b61dee1",
        "name": "Standard Small App Service Hours",
        "category": "Azure App Service",
        "subcategory": ""
      },
      "quantity": 2.0,
      "unit": "Hours",
      "infoFields": {
        "meteredRegion": "North Central US",
        "meteredService": "Azure App Service",
        "meteredServiceType": "abtintest",
        "project": "JBlack2-NorthCentralUSwebspace"
      },
      "attributes": {
        "objectType": "AzureUtilizationRecord"
      }
    },
    {
      "usageStartTime": "2015-11-30T16:00:00-08:00",
      "usageEndTime": "2015-12-01T16:00:00-08:00",
      "resource": {
        "id": "c0f5cb45-6fb1-41c9-8545-72ad400d9da4",
        "name": "Free App Service",
        "category": "Azure App Service",
        "subcategory": ""
      },
      "quantity": 0.002688,
      "unit": "Apps",
      "infoFields": {
        "meteredRegion": "North Central US",
        "meteredService": "Azure App Service",
        "meteredServiceType": "abtintest",
        "project": "abtintest"
      },
      "attributes": {
        "objectType": "AzureUtilizationRecord"
      }
    }
  ],
  "links": {
    "self": {
      "uri": "customers/65726577-c208-40fd-9735-8c85ac9cac68/subscriptions/87f4b92f-A490-485E-AD34-5B70CBA4AF74/utilizations/azure?start_time=2015-12-02T00:00:00Z&end_time=2016-12-02T00:00:00Z&granularity=Daily&show_details=True&size=100",
      "method": "GET",
      "headers": []
    },
    "next": {
      "uri": "customers/65726577-c208-40fd-9735-8c85ac9cac68/subscriptions/87f4b92f-A490-485E-AD34-5B70CBA4AF74/utilizations/azure?seek_operation=Next",
      "method": "GET",
      "headers": [{
        "key": "MS-ContinuationToken",
        "value": "/1f5ccb06-8e36-4a74-a74c-fcaa9e87e26a/usage-records?entitlement_id=87f4b92f-a490-485e-ad34-5b70cba4af74&reported_start_time=2015-12-02+00%3a00%3a00Z&reported_end_time=2016-12-02+00%3a00%3a00Z&granularity=Daily&show_details=True&size=100"
      ]
    }
  ]
}
```

```
0Z+00%5a00%5a00Z&count=100&show_details=true&error_if_not_processed=true&granularity=Daily&continuation_token=eyJyIjo1MjAxN1E  
yMTYtNjM1ODU4MjA4MDAwMDAwMDAwMDAwIiwiaSI6IldNR2hrYWwhKU1VBesS9CZkFnSDNKSlhCZERZVEVmY3Z5NWQvOHFmSEthWHdiNi9W  
d2VvZWVscVltL0Rtbys3QUU0NGVUZmFvc2ZMTG5GM0hScW92WkhPT0iLCJzIjoiaFVrUGVaczd0ifQ%3d%3d"  
    }  
  ]  
}  
,  
"attributes": {  
  "objectType": "Collection"  
}  
}
```

Next steps

- [Review](#) what are the available APIs for Azure CSP integration.
- [Review](#) the list of available Azure CSP Integration scenarios.

Get customer invoices

9/19/2017 • 3 min to read • [Edit Online](#)

This article covers how to retrieve customer invoices one at a time, in a group, or as wish line items, by using the Microsoft Partner Center APIs.

PowerShell

Get invoice by ID

To retrieve a specific invoice by ID in PowerShell, enter the following command:

```
$invoice = Get-PCInvoice -invoiceid '<invoice id>'
```

Get a collection of all invoices

To retrieve a list of all your invoices in PowerShell, enter the following command:

```
Get-PCInvoice -all
```

Get invoice line items

To retrieve the line items of a customer invoice in PowerShell, run the following command:

```
$invoice = Get-PCInvoice -invoiceid '<invoice id>'

Get-PCInvoiceLineItems -invoiceid $invoice.id -billingprovider '<provider>' -invoicelineitemtype '<line item type>'
```

C#

Get invoice by ID

To get an invoice by ID, use your **IPartner.Invoices** collection and call the **ById()** method, and then call the **Get()** or **GetAsync()** method.

```
// IPartner scopedPartnerOperations;
// var selectedInvoiceId as string;
var invoice = scopedPartnerOperations.Invoices.ById(selectedInvoiceId).Get();
```

Get a collection of all invoices

To get a collection of invoices, use your **IPartner.Invoices** collection to call the **Query** method and build your collection.

```
// IPartner scopedPartnerOperations;

var pagedInvoiceCollection = scopedPartnerOperations.Invoices.Query(QueryFactory.Instance.BuildIndexedQuery(5, 0));
```

Get invoice line items

To get the reconciliation file information for a specific invoice, use your **IPartnerOperations.Invoices** collection and call the **ByID()** method, and then call the **Get()** or **GetAsync()** method on the resulting object.

```
// IPartner scopedPartnerOperations;  
// var selectedInvoiceId as string;  
  
var invoiceOperations = partnerOperations.InvoicesById(invoiceId);  
var invoice = invoiceOperations.Get();  
  
foreach (var invoiceDetail in invoice.InvoiceDetails)  
{...}
```

REST API

Get invoice by ID

Request syntax

METHOD	REQUEST URI
GET	<i>{baseUrl}/v1/invoices/{invoice-id}</i> HTTP/1.1

URI parameter

To get the invoice, use the following query parameter:

NAME	TYPE	DESCRIPTION
invoice-id	string	The value allows the reseller to filter the results for a specific invoice.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/invoices/<invoice-id> HTTP/1.1  
Authorization: Bearer <token>  
Accept: application/json  
MS-RequestId: 8ac25aa5-9537-4b6d-b782-aa0c8e979e99  
MS-CorrelationId: 57eb2ca7-755f-450f-9187-eae1e75a0114
```

If the request is successful, this method returns an **InvoiceSummary** object in the response.

Response example

```

HTTP/1.1 200 OK
Content-Length: 676
Content-Type: application/json; charset=utf-8
MS-CorrelationId: 57eb2ca7-755f-450f-9187-eae1e75a0114
MS-RequestId: 8ac25aa5-9537-4b6d-b782-aa0c8e979e99
Date: Thu, 24 Mar 2016 05:22:14 GMT

{
  "id": "<invoice-id>",
  "invoiceDate": "2015-08-11T00:00:00",
  "totalCharges": 3822.14,
  "paidAmount": 0.0,
  "currencyCode": "USD",
  "pdfDownloadLink": "<pdfDownloadLink>",
  "invoiceDetails": [{
    "invoiceLineItemType": "billing_line_items",
    "billingProvider": "office",
    "links": {
      "self": {
        "uri": "/invoices/<invoice-id>/lineitems/Office/BillingLineItems",
        "method": "GET",
        "headers": []
      }
    },
    "attributes": {
      "objectType": "InvoiceDetail"
    }
  }],
  "links": {
    "self": {
      "uri": "/invoices/<invoice-id>",
      "method": "GET",
      "headers": []
    }
  },
  "attributes": {
    "objectType": "Invoice"
  }
}

```

Get a collection of all invoices

Request syntax

METHOD	REQUEST URI
GET	{baseUrl}/v1/invoices HTTP/1.1

Request example

```

GET https://api.partnercenter.microsoft.com/v1/invoices HTTP/1.1
Authorization: Bearer
Accept: application/json
MS-RequestId: 8ba3af44-dfcd-4aaa-b04c-53363bee0287
MS-CorrelationId: 57eb2ca7-755f-450f-9187-eae1e75a0114

```

If the request is successful, this method returns a collection of **InvoiceDetail** objects in the response.

```

HTTP/1.1 200 OK
Content-Length: 5310
Content-Type: application/json; charset=utf-8
MS-CorrelationId: 57eb2ca7-755f-450f-9187-eae1e75a0114
MS-RequestId: 8ba3af44-dfcd-4aaa-b04c-53363bee0287

```

Date: Thu, 24 Mar 2016 05:23:53 GMT

```
{
  "totalCount":9,
  "items":[{
    "id":"D070002EGD",
    "invoiceDate":"2016-03-11T00:00:00",
    "totalCharges":17383.43,
    "paidAmount":0.0,
    "currencyCode":"USD",
    "invoiceDetails":[{
      "invoiceLineItemType":"billing_line_items",
      "billingProvider":"office",
      "links":{
        "self":{
          "uri":"/invoices/D070002EGD/lineitems/Office/BillingLineItems",
          "method":"GET",
          "headers":[]
        }
      },
      "attributes":{
        "objectType":"InvoiceDetail"
      }
    }],
    {
      "invoiceLineItemType":"billing_line_items",
      "billingProvider":"azure",
      "links":{
        "self":{
          "uri":"/invoices/D070002EGD/lineitems/Azure/BillingLineItems",
          "method":"GET",
          "headers":[]
        }
      },
      "attributes":{
        "objectType":"InvoiceDetail"
      }
    }],
    {
      "invoiceLineItemType":"usage_line_items",
      "billingProvider":"azure",
      "links":{
        "self":{
          "uri":"/invoices/D070002EGD/lineitems/Azure/UsageLineItems",
          "method":"GET",
          "headers":[]
        }
      },
      "attributes":{
        "objectType":"InvoiceDetail"
      }
    }],
    "links":{
      "self":{
        "uri":"/invoices/D070002EGD",
        "method":"GET",
        "headers":[]
      }
    },
    "attributes":{
      "objectType":"Invoice"
    }
  }],
  "links":{
    "self":{
      "uri":"/invoices",
      "method":"GET",
      "headers":[]
    }
  }
}
```

```

    }
  },
  "attributes": {
    "objectType": "Collection"
  }
}

```

Get invoice line items

Request syntax

Use the first syntax to return a full list of every line item for a specific invoice. For large invoices, use the second syntax with a specified size and 0-based offset to return a paged list of line items.

METHOD	REQUEST URI
GET	{baseUrl}/v1/invoices/{invoice-id}/lineitems/{billing-provider}/{invoice-line-item-type} HTTP/1.1
GET	{baseUrl}/v1/invoices/{invoice-id}/lineitems/{billing-provider}/{invoice-line-item-type}?size={size}&offset=0 HTTP/1.1

URI parameter

To get the invoice line items, use the following *required* parameters:

NAME	TYPE	DESCRIPTION
invoice-id	string	The value allows the reseller to filter the results for a specific invoice.
billing-provider	string	A billing-provider that allows the filtering of results for a specific service within the invoice.
invoice-line-item-type		A value that allows the filtering of results by a specific type of line item.
size	int	The number of results to be displayed at one time.

Request example

```

GET https://api.partnercenter.microsoft.com/v1/invoices/<invoice-id>/lineitems/<billing-provider>/<invoice-line-item-type>?size=
<size>&offset=0 HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: 646e3cfe-16a8-4b8f-8171-11a48a60ef28
MS-CorrelationId: 57eb2ca7-755f-450f-9187-eae1e75a0114

```

If the request is successful, this method returns an **InvoiceSummary** object in the response.

Response example

HTTP/1.1 200 OK
Content-Length: 40019
Content-Type: application/json; charset=utf-8
MS-CorrelationId: 57eb2ca7-755f-450f-9187-eae1e75a0114
MS-RequestId: 646e3cfe-16a8-4b8f-8171-11a48a60ef28
Date: Thu, 24 Mar 2016 05:26:38 GMT

```
{
  "totalCount":45,
  "items":[{
    "partnerId":<partnerId>,
    "customerId":<customerId>,
    "customerName":"COMPANYNAME",
    "mpnId":<mpnId>,
    "tier2MpnId":-1,
    "orderId":<orderId>,
    "subscriptionId":<subscriptionId>,
    "syndicationPartnerSubscriptionNumber":<subscriptionNumber>,
    "offerId":<offerId>,
    "durableOfferId":<durabeOfferId>,
    "offerName":"OFFICE 365 ENTERPRISE E3",
    "subscriptionStartDate":"2016-08-10T00:00:00",
    "subscriptionEndDate":"2016-08-10T00:00:00",
    "chargeStartDate":"2015-07-13T00:00:00",
    "chargeEndDate":"2015-08-09T00:00:00",
    "chargeType":"PURCHASE FEE",
    "unitPrice":0.0,
    "quantity":10,
    "amount":0.0,
    "totalOtherDiscount":0.0,
    "subtotal":0.0,"tax":0.0,
    "totalForCustomer":0.0,
    "currency":"USD",
    "invoiceLineItemType":"billing_line_items",
    "billingProvider":"office",
    "attributes":{
      "objectType":"LicenseBasedLineItem"
    }
  }],
  "links":{
    "self":{
      "uri":"/invoices/<invoice-id>/lineitems/<billing-provider>/<invoice-line-item-type>?size=<size>&offset=0",
      "method":"GET",
      "headers":[]
    }
  },
  "attributes":{
    "objectType":"Collection"
  }
}
```

Next steps

- [Review](#) what are the available APIs for Azure CSP integration.
- [Review](#) the list of available Azure CSP Integration scenarios.

Get a customer's service cost summary

9/19/2017 • 1 min to read • [Edit Online](#)

Learn how to get a summary of a customer's service costs for a specific billing period. You can use one of the following options:

- PowerShell
- C#
- REST API

PowerShell

```
Select-PCCustomer -tenantid '<tenant id GUID>'

Get-PCCustomerServiceCostSummary -BillingPeriod MostRecent
```

C#

1. To identify the customer, call the **IAggregatePartner.Customers.ById** method with the customer ID.
2. To get an interface for collection operations for customer service costs, use the **ServiceCosts** property.
3. To return a value for **IServiceCostsCollection**, call the **ByBillingPeriod** method with a member of the billing period enumerator.
4. To get the customer's service cost summary, use the **IServiceCostsCollection.Summary.Get** or the **GetAsync** method.

```
// IAggregatePartner partnerOperations;
// string selectedCustomerId;

var serviceCostsSummary =
    partnerOperations.Customers.ById(selectedCustomerId).ServiceCosts.ByBillingPeriod(ServiceCostsBillingPeriod.MostRecent).Summary.Get();
```

REST API

Request

Request syntax

METHOD	REQUEST URI
GET	{baseUrl}/v1/customers/{customer-id}/servicecosts/{billing-period} HTTP/1.1

URI parameters

The following table lists the parameters that are required for identifying a customer.

NAME	TYPE	DESCRIPTION
------	------	-------------

NAME	TYPE	DESCRIPTION
customer-id	guid	A GUID-format customer ID that identifies the customer.
billing-period	string	The billing period. The only supported value is MostRecent . The string is not case-sensitive.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers/65726577-c208-40fd-9735-8c85ac9cac68/servicecosts/mostrecent HTTP/1.1
Authorization: Bearer <authorization token>
Accept: application/json
MS-RequestId: e6a3b6b2-230a-4813-999d-57f883b60d38
MS-CorrelationId: a687bc47-8d08-4b78-aff6-5a59aa2055c2
X-Locale: en-US
Host: api.partnercenter.microsoft.com
```

Response

If the request is successful, the response body contains a **ServiceCostsSummary** resource that provides information about the customer's service costs.

Response example

```
HTTP/1.1 200 OK
Content-Length: 766
Content-Type: application/json; charset=utf-8
MS-CorrelationId: a687bc47-8d08-4b78-aff6-5a59aa2055c2
MS-RequestId: e6a3b6b2-230a-4813-999d-57f883b60d38
MS-CV: gPPoyNX1X0asAAcw.0
MS-ServerId: 101112202
Date: Fri, 02 Dec 2016 18: 54: 38 GMT

{
  "billingStartDate": "2015-12-12T00:00:00Z",
  "billingEndDate": "2016-01-11T00:00:00Z",
  "pretaxTotal": 17.22,
  "tax": 0.0,
  "afterTaxTotal": 17.22,
  "currencySymbol": "$",
  "customerId": "ae1d5b32-f9ff-4252-b2bf-40e21937a51a",
  "links": {
    "serviceCostLineItems": {
      "uri": "/customers/ae1d5b32-f9ff-4252-b2bf-40e21937a51a/servicecosts/MostRecent/lineitems",
      "method": "GET",
      "headers": []
    },
    "self": {
      "uri": "/customers/ae1d5b32-f9ff-4252-b2bf-40e21937a51a/servicecosts/MostRecent",
      "method": "GET",
      "headers": []
    }
  },
  "attributes": {
    "objectType": "ServiceCostsSummary"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Get a customer's service cost line items

9/19/2017 • 1 min to read • [Edit Online](#)

Learn how to get a list of a customer's service cost line items for a specific billing period. You can use one of the following options:

- C#
- REST API

C#

1. To identify the customer, call the **IAggregatePartner.Customers.ById** method.
2. To get an interface for collection operations for customer service costs, use the **ServiceCosts** property.
3. To return a value for **IServiceCostsCollection**, call the **ByBillingPeriod** method with a member of the billing period enumerator.
4. To get the customer's service cost line items, use the **IServiceCostsCollection.LineItems.Get** or the **GetAsync** method.

```
// IAggregatePartner partnerOperations;  
// string selectedCustomerId;  
  
var serviceCostsSummary =  
    partnerOperations.Customers.ById(selectedCustomerId).ServiceCosts.ByBillingPeriod(ServiceCostsBillingPeriod.MostRecent).LineItems.Get();
```

REST API

Request

Request syntax

METHOD	REQUEST URI
GET	{baseUrl}/v1/customers/{customer-id}/servicecosts/{billing-period}/lineitems HTTP/1.1

URI parameter

The following table lists the parameters that are required for identifying a customer.

NAME	TYPE	DESCRIPTION
customer-id	guid	A GUID-format customer ID that identifies the customer.
billing-period	string	The billing period. The only supported value is MostRecent . The string is not case-sensitive.

```
GET https://api.partnercenter.microsoft.com/v1/customers/65726577-c208-40fd-9735-8c85ac9cac68/servicecosts/mostrecent/lineitems HTTP/1.1
Authorization: Bearer <authorization token>
Accept: application/json
MS-RequestId: e6a3b6b2-230a-4813-999d-57f883b60d38
MS-CorrelationId: a687bc47-8d08-4b78-aff6-5a59aa2055c2
X-Locale: en-US
Host: api.partnercenter.microsoft.com
```

Response

If the request is successful, the response body contains a **ServiceCostLineItem** resource that provides information about the customer's service costs.

Response example

HTTP/1.1 200 OK
Content-Length: 2148
Content-Type: application/json; charset=utf-8
MS-CorrelationId: a687bc47-8d08-4b78-aff6-5a59aa2055c2
MS-RequestId: e6a3b6b2-230a-4813-999d-57f883b60d38
MS-CV: gPPoyNX1X0asAAcw.0
MS-ServerId: <server ID>
Date: Fri, 02 Dec 2016 18: 54: 38 GMT

```
{
  "attributes": {
    "objectType": "Collection"
  },
  "items": [
    {
      "afterTaxTotal": 0.0,
      "chargeType": "PURCHASE FEE",
      "currencyCode": "USD",
      "currencySymbol": "$",
      "customerId": "<customer ID>",
      "customerName": "<customer name>",
      "endDate": "2016-01-11T00:00:00",
      "offerId": "<offer ID>",
      "offerName": "Project for Office 365 (Government Pricing)",
      "orderId": "<order ID>",
      "pretaxTotal": 0.0,
      "quantity": 1.0,
      "resellerMPNId": "-1",
      "startDate": "2015-12-15T00:00:00",
      "subscriptionFriendlyName": "Project Pro for Office 365 (Government Pricing)",
      "subscriptionId": "<subscription ID>",
      "tax": 0.0,
      "unitPrice": 0.0
    }, {
      "afterTaxTotal": 17.219999999999999,
      "chargeType": "CYCLE FEE",
      "currencyCode": "USD",
      "currencySymbol": "$",
      "customerId": "<customer ID>",
      "customerName": "<customer name>",
      "endDate": "2016-02-11T00:00:00",
      "offerId": "<offer ID>",
      "offerName": "Project for Office 365 (Government Pricing)",
      "orderId": "<order ID>",
      "pretaxTotal": 17.219999999999999,
      "quantity": 1.0,
      "resellerMPNId": "-1",
      "startDate": "2016-01-12T00:00:00",
      "subscriptionFriendlyName": "Project Pro for Office 365 (Government Pricing)",
      "subscriptionId": "<subscription ID>",
      "tax": 0.0,
      "unitPrice": 17.219999999999999
    }
  ],
  "links": {
    "self": {
      "headers": [],
      "method": "GET",
      "uri": "/customers/<customer ID>/servicecosts/MostRecent/lineitems"
    }
  },
  "totalCount": 2
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Get subscription usage data

9/22/2017 • 1 min to read • [Edit Online](#)

Learn how to get the usage data for the subscriptions on a customer account. You can use one of the following options:

- Partner Center web UI
- PowerShell
- C#
- REST API

Partner Center web UI

1. On the Microsoft Partner Center **Dashboard**, go to **Customers**. Select the specific customer.
2. Select **Subscriptions**, and then select the specific subscription.

The screenshot shows the Microsoft Partner Center interface. On the left, a sidebar lists navigation options: Subscriptions, Customer insights, Users and licenses, Service management, and Account. The main content area is titled 'Subscriptions' and includes an 'Add subscription' button. Below this, there are two sections: 'License-based subscriptions' and 'Usage-based subscriptions'. The 'Usage-based subscriptions' section contains a table with the following data:

Subscription	Current estimate	% of total	Status
Microsoft Azure		0%	Active
CSP Tool		0%	Active

The 'Microsoft Azure' row is highlighted with a red box. At the bottom of the page, there is a footer with links for Industries, For customers, For partners, For developers, Values, and Company.

The estimated usage of the subscription is shown at the bottom of the page, under **Estimated Usage**.

PowerShell

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'
Get-PCSubscriptionMonthlyUsageRecords -tenantid $customer.id
```

C#

1. Use your **IAggregatePartner.Customers** collection, and then call the **ById()** method.
2. Call the **Subscriptions** property, then **UsageRecords**, and then the **Resources** property.
3. Call the **Get()** or the **GetAsync()** method.

```
// IAggregatePartner partnerOperations;  
// var selectedCustomerId as string;  
// var selectedSubscriptionID as string;  
  
var usageRecords =  
partnerOperations.Customers.ById(selectedCustomerId).Subscriptions.ById(selectedSubscriptionId).UsageRecords.Resources.Get();
```

REST API

Request

Request syntax

METHOD	REQUEST URI
GET	{baseUrl}/v1/customers/{customer-tenant-id}/subscriptions/{id-for-subscription}/usagerecords/resources HTTP/1.1

URI parameter

The following table lists the query parameters that are required for getting the rated usage information.

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	A GUID that corresponds to the customer.
id-for-subscription	guid	A GUID that corresponds to the subscription.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers/{customer-tenant-id}/subscriptions/{id-for-subscription}/usagerecords/resources  
HTTP/1.1  
Authorization: Bearer <token>  
Accept: application/json  
MS-RequestId: 65b26053-37d0-4303-9fd1-46ad8012bcb6  
MS-CorrelationId: 47c36033-af5d-4457-80a4-512c1626fac4
```

Response

If the request is successful, this method returns a collection of **AzureResourceMonthlyUsageRecord** resources in the response body.

Response example

HTTP/1.1 200 OK
Content-Length: 12014
Content-Type: application/json
MS-CorrelationId: 648a26a4-a63e-459f-844b-4f29d7913353
MS-RequestId: be82a8ba-4a53-49f7-8313-b033c058687e
Date: Tue, 10 Nov 2015 19:09:59 GMT

```
{
  "totalCount":20,
  "items":[{
    "category":"Storage",
    "subcategory":"LOCALLY REDUNDANT",
    "quantityUsed":0.151287527825352,
    "unit":"GB",
    "id":"2a2419c0-cefe-46b2-8004-8eb002ad606c",
    "name":"Azure Resource 1",
    "totalCost":0.195779159290613,
    "currencyLocale":"en-US",
    "attributes":{
      "objectType":"AzureResourceMonthlyUsageRecord"
    }
  },
  {
    "category":"Remote App",
    "subcategory":"Remote App",
    "quantityUsed":0.932546524299563,
    "unit":"GB",
    "id":"7e4099c8-2b3d-41a6-a1bd-d5cf315989b2",
    "name":"Azure Resource 2",
    "totalCost":0.920983775016379,
    "currencyLocale":"en-US",
    "attributes":{
      "objectType":"AzureResourceMonthlyUsageRecord"
    }
  }
],
  "links":{
    "self":{
      "uri":"/v1/customers/<customer-tenant-id>/subscriptions/<id-for-subscription>/%20/usagerecords",
      "method":"GET",
      "headers":[]
    }
  },
  "attributes":{
    "objectType":"Collection"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Get a usage summary for a customer's subscriptions

9/22/2017 • 1 min to read • [Edit Online](#)

Learn how to get a **CustomerUsageSummary** resource that represents the customer's usage of a specific Azure service or resource during the current billing period. You can use one of the following options:

- PowerShell
- C#
- REST API

PowerShell

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'

Get-PCCustomerUsageSummary -tenantid $customer.id
```

C#

1. Use your **IAggregatePartner.Customers** collection, and then call the **ById()** method.
2. Call the **UsageSummary** property.
3. Call the **Get()** or the **GetAsync()** method.

```
// IAggregatePartner partnerOperations;
// var selectedCustomerId as string;

var usageSummary = partnerOperations.Customers.ById(selectedCustomerId).UsageSummary.Get();
```

REST API

IMPORTANT

Azure Resource Usage API is not available for Azure CSP Subscriptions. Resource Usage should be accessed through [Partner Center API](#) instead.

Request

Request syntax

METHOD	REQUEST URI
GET	{baseUrl}/v1/customers/{customer-tenant-id}/usagesummary HTTP/1.1

URI parameter

The following query parameter is required for getting the customer's rated usage information.

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	A GUID that corresponds to the customer.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers/{customer-tenant-id}/usagesummary HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: e128c8e2-4c33-4940-a3e2-2e59b0abdc67
MS-CorrelationId: 47c36033-af5d-4457-80a4-512c1626fac4
```

Response

If the request is successful, this method returns a **CustomerUsageSummary** resource in the response body.

Response example

```
HTTP/1.1 200 OK
Content-Length: 1120
Content-Type: application/json
MS-CorrelationId: 47c36033-af5d-4457-80a4-512c1626fac4
MS-RequestId: e128c8e2-4c33-4940-a3e2-2e59b0abdc67
Date: Fri, 26 Feb 2016 20:31:45 GMT

{
  "budget": {
    "ammount": 300.000000,
    "attributes": {
      "objectType": "SpendingBudget"
    }
  },
  "id": "65726577-C208-40FD-9735-8C85AC9CAC68",
  "name": "600 test",
  "billingStartDate": "2016-02-06T00:00:00-08:00",
  "billingEndDate": "2016-03-05T00:00:00-08:00",
  "totalCost": 0.0,
  "currencyLocale": "en-US",
  "lastModifiedDate": "2016-02-26T09:42:54.5130558+00:00",
  "links": {
    "self": {
      "uri": "/customers/{customer-tenant-id}/usagesummary",
      "method": "GET",
      "headers": []
    }
  },
  "attributes": {
    "objectType": "CustomerUsageSummary"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Update a customer's billing profile

9/19/2017 • 2 min to read • [Edit Online](#)

Learn how to update a customer's billing information. You can use one of the following options:

- Partner Center web UI
- PowerShell
- C#
- REST API

Partner Center web UI

1. In Microsoft Partner Center, on the **Dashboard** menu, select **Customers**. Then, select a customer.
2. On the customer menu, select **Account**, and then select **Update**.

The screenshot shows the Microsoft Partner Center web interface. At the top, there's a navigation bar with the Microsoft logo and links for Cloud, Mobility, and Productivity. Below this is a search bar and a navigation menu with links like Programs, How-to, Support, and Find a Partner. The main content area is titled 'ACME Corporation' and 'Account'. On the left, there's a sidebar with links for Subscriptions, Customer insights, Users and licenses, Service management, and Account (which is highlighted with a red box). The main content area displays account details for ACME Corporation, including Microsoft ID, Domain name, Company info, and Bill-to info. The 'Update' button next to 'Bill-to info' is highlighted with a red box. At the bottom, there's a footer with various links organized into columns: Industries, For customers, For partners, For developers, Values, and Company.

3. After you've made your changes, select **Submit**.

Microsoft Partner Center

ACME Corporation

Subscriptions

Customer insights

Users and licenses

Service management

Account

← Customers

Bill-to info

Country/region Why can't I change this?

United States

Address line 1 *

1 Tower Lane

City *

Los Angeles

First name *

Melvin

Email address *

mblanc@outlook.com

Company name *

ACME Corporation

Address line 2

State/Province *

California

ZIP/Postal code *

90002

Last name *

Blanc

Phone number

Submit Cancel

Industries
Manufacturing & resources

For customers
Small & midsize businesses

For partners
Get listed

For developers
Microsoft Azure

Values
Diversity and inclusion

Company
Careers

PowerShell

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'

$customerBillingProfile = Get-PCCustomerBillingProfile -tenantid $customer.id

$customerBillingProfile.firstName = '<first name>'
$customerBillingProfile.lastName = '<last name>'
$customerBillingProfile.email = '<email>'
Set-PCCustomerBillingProfile -tenantid $customer.id -billingprofile $customerBillingProfile
```

C#

1. Retrieve the billing profile. Update the properties as needed.
2. Retrieve your **IPartner.Customers** collection, and then call the **ById()** method.
3. Call the **Profiles** property, followed by the **Billing** property.
4. Call the **Update()** or the **UpdateAsync()** method.

```
// IAggregatePartner partnerOperations;
// var selectedCustomerId as string;

var billingProfile = partnerOperations.Customers.ById(selectedCustomerId).Profiles.Billing.Get();

// Apply changes to profile;

billingProfile = partnerOperations.Customers.ById(selectedCustomerId).Profiles.Billing.Update(billingProfile);
```

REST API

Request

Request syntax

METHOD	REQUEST URI
--------	-------------

METHOD	REQUEST URI
PUT	{baseUrl}/v1/customers/{customer-tenant-id}/profiles/billing HTTP/1.1

URI parameter

Use the query parameter in the following table to update the billing profile.

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	The value is a GUID-format customer tenant ID. The reseller can use customer-tenant-id to filter the results for a specific customer.

If-Match: An HTML entity tag (ETag) is required for concurrency detection.

Request example

```
PUT https://api.partnercenter.microsoft.com/v1/customers/<customer-tenant-id>/profiles/billing HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: ff85f13a-eb65-4b8e-9b67-05beb0565410
MS-CorrelationId: ff1b757d-cfaf-463a-b48b-0f96d05e95d7
Content-Type: application/json
Content-Length: 639
Expect: 100-continue
```

```
{
  "Id": "a58ceba5-60ac-48e4-a0bc-2a855811807a",
  "FirstName": "FirstName",
  "LastName": "LastName",
  "Email": "email@sample.com",
  "Culture": "en-US",
  "Language": "en",
  "CompanyName": "CompanyName",
  "DefaultAddress": {
    "Country": "US",
    "Region": null,
    "City": "Redmond",
    "State": "WA",
    "AddressLine1": "One Microsoft Way",
    "AddressLine2": null,
    "PostalCode": "98052",
    "FirstName": "FirstName",
    "LastName": "LastName",
    "PhoneNumber": "4255555555"
  },
  "Links": {
    "Self": {
      "Uri": "/v1/customers/<customer-tenant-id>/profiles/billing",
      "Method": "GET",
      "Headers": []
    }
  },
  "Attributes": {
    "Etag": "<etag>",
    "ObjectType": "CustomerBillingProfile"
  }
}
```

Response

If the request is successful, this method returns updated properties for **Profile** resources in the response body. This call requires an ETag for concurrency detection.

Response example

```
HTTP/1.1 200 OK
Content-Length: 1210
Content-Type: application/json
MS-CorrelationId: ff1b757d-cfaf-463a-b48b-0f96d05e95d7
MS-RequestId: ff85f13a-eb65-4b8e-9b67-05beb0565410
Date: Mon, 23 Nov 2015 18:20:43 GMT

{
  "id": "a58ceba5-60ac-48e4-a0bc-2a855811807a",
  "firstName": "FirstName",
  "lastName": "LastName",
  "email": "email@sample.com",
  "culture": "en-US",
  "language": "en",
  "companyName": "companyName",
  "defaultAddress": {
    "country": "US",
    "city": "Redmond",
    "state": "WA",
    "addressLine1": "One Microsoft Way",
    "postalCode": "98052",
    "firstName": "FirstName",
    "lastName": "LastName",
    "phoneNumber": "4255555555"
  },
  "links": {
    "self": {
      "uri": "/v1/customers/<customer-tenant-id>/profiles/billing",
      "method": "GET",
      "headers": []
    }
  },
  "attributes": {
    "etag": "<etag>",
    "objectType": "CustomerBillingProfile"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Set an Azure spending budget for a customer

9/19/2017 • 1 min to read • [Edit Online](#)

You can set up a monthly spending budget to help a customer manage their Azure costs. A spending budget can help a customer ensure that their Azure bill isn't higher than they had anticipated.

This feature includes:

- Email notifications if a customer is nearing a spending limit.
- Review of a customer's estimated costs per month.
- Notifications of unusual usage trends that might suggest fraud.
- Cost management and identification of root issues.

Azure spending data is an estimate. Actual billing amounts might vary. Azure spending data does not include taxes, credits, adjustments, and so on. Spending data is refreshed once each day.

NOTE

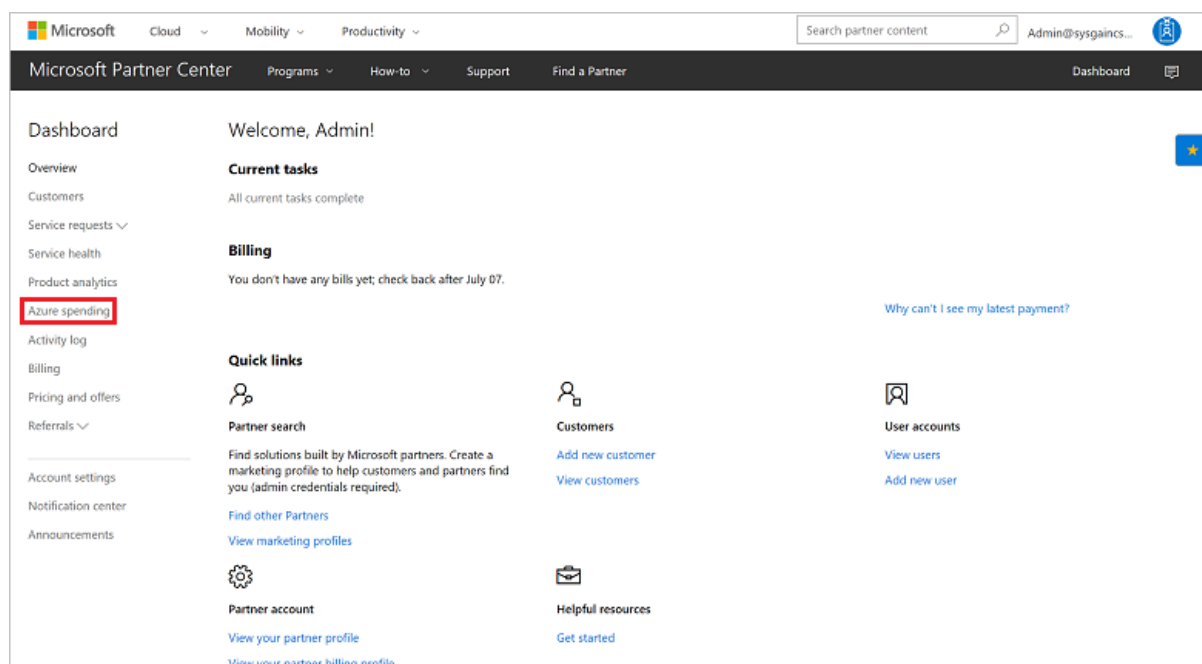
This feature is not available in sandbox accounts or in production test accounts.

WARNING

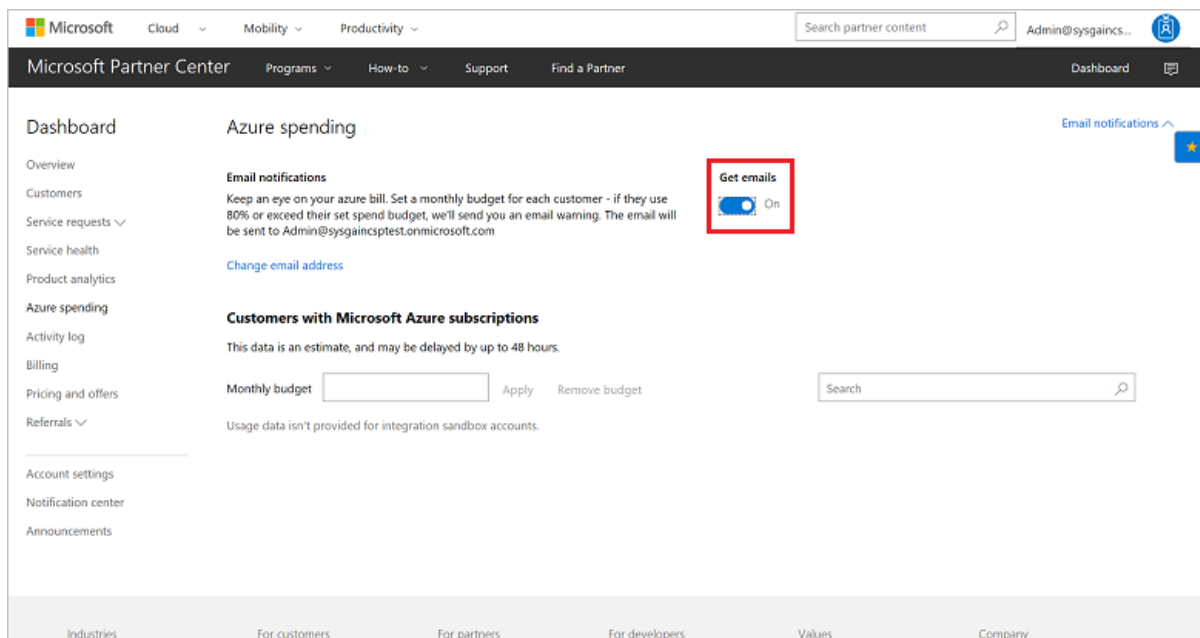
Creating a budget does not stop a customer from going over the set spending limit, or from being charged for additional services. A budget is only a preventative tool.

Turn on email notifications

1. On the Microsoft Partner Center **Dashboard** menu, select **Azure spending**.

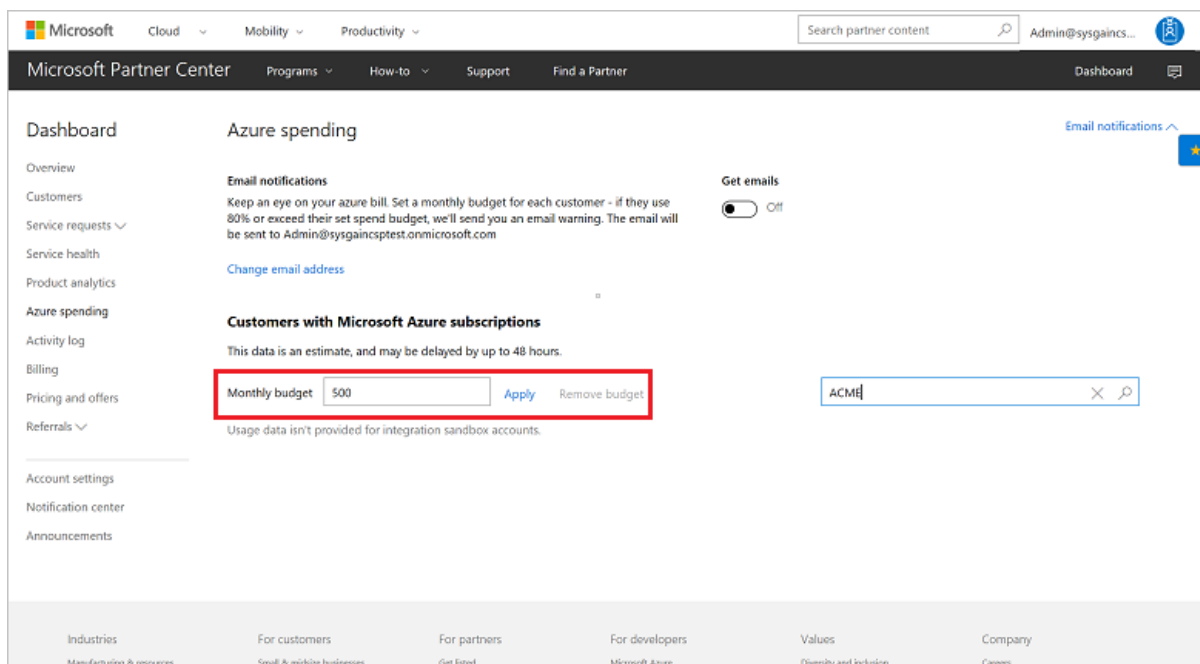


2. Under **Get emails**, select **On**. With this setting turned on, you are notified if the customer uses 80 percent or more of their budget. This can help you monitor the customer's Azure bill. Make sure that the default email address is one that you will monitor for notifications.



Set a budget

1. On the **Dashboard** menu, select **Azure spending**.
2. Select a customer.
3. In the **Monthly budget** box, enter a value, and then select **Apply**.



4. To check current spending, return to the **Azure spending** page.

You also can set a spending budget for a customer on the customer management page, under **Usage-based subscriptions**.

See itemized costs

1. On the **Dashboard** menu, select **Customers**, and then select a customer.
2. Select **Subscriptions**, and then select the subscription.

Microsoft Partner Center

Subscriptions

[Add subscription](#)

License-based subscriptions

Subscription	Quantity	Status
Azure Active Directory Basic	1 Licenses	Auto renews on 1/6/18

Usage-based subscriptions [All resources \(Azure portal\)](#)

Usage data isn't provided for integration sandbox accounts.

Subscription	Current estimate	% of total	Status
Microsoft Azure		0%	Active
CSP Tool		0%	Active

Industries: Manufacturing & resources

For customers: Small & midsize businesses

For partners: Get listed

For developers: Microsoft Azure

Values: Diversity and inclusion

Company: Careers

On a customer management page, under **Usage-based subscriptions**, select a subscription. You can view the current estimated usage for the subscription, and see a list of itemized costs by service.

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Get a partner's current account balance

9/19/2017 • 1 min to read • [Edit Online](#)

This guide covers how to retrieve the current account balance of a partner.

C# guide

To retrieve your reseller account balance, do the following:

1. Use your **IAggregatePartner.Invoices** collection.
2. Call the **Summary** property.
3. Call the **Get** function.
4. Call the **BalanceAmount** property.

```
// IAggregatePartner scopedPartnerOperations;  
  
var invoiceSummary = scopedPartnerOperations.Invoices.Summary.Get();  
  
Console.Out.WriteLine("Current Account Balance: {0:C}", invoiceSummary.BalanceAmount);
```

REST API guide

Request

Request syntax

METHOD	REQUEST URI
GET	{baseUrl}/v1/invoices/summary HTTP/1.1

Request example

```
GET https://api.partnercenter.microsoft.com/v1/invoices/summary HTTP/1.1  
Authorization: Bearer <token>  
Accept: application/json  
MS-RequestId: a45e6643-1caf-4429-8f90-07c03d85bc2b  
MS-CorrelationId: 57eb2ca7-755f-450f-9187-eae1e75a0114  
Connection: Keep-Alive
```

Response

If the request is successful, this method returns an **InvoiceSummary** object in the response.

Response example

HTTP/1.1 200 OK
Content-Length: 256
Content-Type: application/json; charset=utf-8
MS-CorrelationId: 57eb2ca7-755f-450f-9187-eae1e75a0114
MS-RequestId: a45e6643-1caf-4429-8f90-07c03d85bc2b
Date: Thu, 24 Mar 2016 05:21:01 GMT

```
{
  "balanceAmount":68059.03,
  "currencyCode":"USD",
  "accountingDate":"2016-03-11T00:00:00",
  "firstInvoiceCreationDate":"2015-07-11T00:00:00",
  "links":{
    "self":{
      "uri":"/invoices/summary",
      "method":"GET",
      "headers":[]
    }
  },
  "attributes":{
    "objectType":"InvoiceSummary"
  }
}
```

Next steps

- [Review](#) what are the available APIs for Azure CSP integration.
- [Review](#) the list of available Azure CSP Integration scenarios.

Create an Azure CSP customer

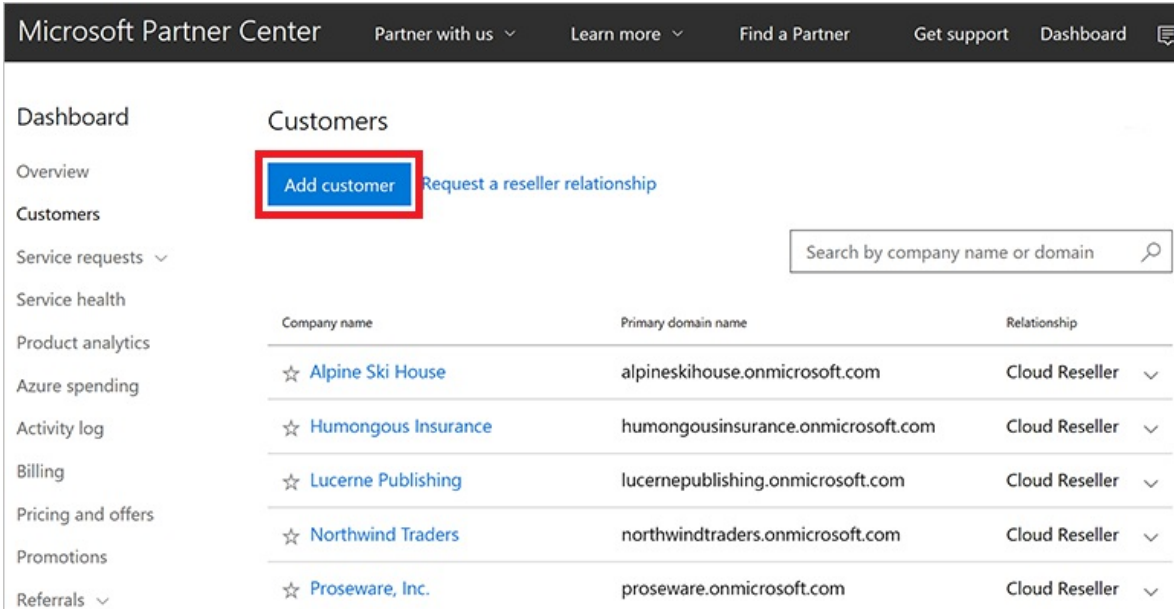
9/19/2017 • 5 min to read • [Edit Online](#)

Before you can access a customer's support or billing information or place orders for a customer, you must create the customer in the system. When you create a customer, you also create:

- An Azure Active Directory (Azure AD) tenant object for the customer.
- A relationship between the reseller and customer, which is used for delegated administrator privileges.
- A username and password for signing in as an administrator for the customer.

Web UI

1. On the Microsoft Partner Center dashboard, in the left pane, select **Customers**, and then select **Add customer**.



The screenshot shows the Microsoft Partner Center dashboard. The top navigation bar includes links for 'Partner with us', 'Learn more', 'Find a Partner', 'Get support', and 'Dashboard'. The left sidebar contains a list of navigation items: 'Dashboard', 'Overview', 'Customers', 'Service requests', 'Service health', 'Product analytics', 'Azure spending', 'Activity log', 'Billing', 'Pricing and offers', 'Promotions', and 'Referrals'. The main content area is titled 'Customers' and features a red-bordered button labeled 'Add customer' and a link 'Request a reseller relationship'. Below this is a search bar with the placeholder text 'Search by company name or domain'. A table lists existing customers with columns for 'Company name', 'Primary domain name', and 'Relationship'.

Company name	Primary domain name	Relationship
☆ Alpine Ski House	alpineskihouse.onmicrosoft.com	Cloud Reseller
☆ Humongous Insurance	humongousinsurance.onmicrosoft.com	Cloud Reseller
☆ Lucerne Publishing	lucernepublishing.onmicrosoft.com	Cloud Reseller
☆ Northwind Traders	northwindtraders.onmicrosoft.com	Cloud Reseller
☆ Proseware, Inc.	proseware.onmicrosoft.com	Cloud Reseller

2. On the **Account info** form, enter the customer's account information.

Microsoft Partner Center
Partner with us
Learn more
Find a Partner
Get support
Dashboard

New customer
Account info
Subscriptions
Review
Confirmation

Account info
Subscriptions
Review
Confirmation

← Customers

Account info

Company

Country/region [Why can't I change this?](#)
United States

Company name *
Fourth Coffee

Primary domain name * [What is this?](#)
fourthcoffeecsp.onmicrosoft.com
fourthcoffeecsp.onmicrosoft.com is available ✓

Address line 1 *
One Microsoft Way

Address line 2

City *
Redmond

State/Province *
Washington

ZIP/Postal cod
98052

Primary contact

First name *
Marshall

Last name *
Levi

Email address *
marshall@fourthcoffee.com

Phone number
1234567890

Next: Subscriptions Cancel

IMPORTANT

You can choose *Country/region* only among the list of countries that are available for the market where you've enrolled as a CSP Partner. If you are authorized in a single country or region (for example, US, Canada, or Brazil), you can't change the country or region during new customer creation process. For more information, see [CSP regional authorization](#).

- Select **Next: Subscriptions**.
- Select the check boxes next to **Microsoft Azure** and any other offers you want, and then select the number of licenses for each offer, where applicable.

NOTE

There are separate tabs for enterprise, small business, and government offers. Make sure you're using the **Enterprise** offers tab.

Microsoft Partner Center

Partner with us

Learn more

Find a Partner

Get support

Dashboard

New customer

Account info

Subscriptions

Review

Confirmation

← Customers

New subscription

Top offers

Choose offers to place at the top of the Catalog so that sales agents can quickly find the ones that matter most to your company.

Catalog

Enterprise Small business Government

☐ Azure Active Directory Basic

☐ Azure Active Directory Premium P1

☐ Azure Active Directory Premium P2

☐ Azure Information Protection Plan 1

☐ Azure Information Protection Premium P2

☐ Dyn365 Cloud Add-on From AX Enterprise to Retail

...

☐ Exchange Online Protection

☐ Intune

☐ Intune Device

☒ Microsoft Azure Usage-based

☐ Microsoft Azure Multi-Factor Authentication

☐ Microsoft Cloud App Security

...

☐ Windows 10 Enterprise E3

☐ Windows 10 Enterprise E5

☐ Yammer Enterprise

Next: Review

Cancel

Microsoft Azure

Azure Cloud Solution Provider offer for Partner and Resellers

Offer ID: MS-AZR-0145P

5. Select **Next: Review**.

6. Review the customer and offer details. To make changes, select the corresponding **Update** link. Otherwise, select **Submit**.

Microsoft Partner Center
Partner with us
Learn more
Find a Partner
Get support
Dashboard

New customer
Account info
Subscriptions
Review
Confirmation

Review
Select **Submit** to confirm the information below. Select **Cancel** to make changes.

Account info

Update

Company
Fourth Coffee
One Microsoft Way
Redmond WA 98052
fourthcoffeecsp.onmicrosoft.com

Primary contact info
Marshall Levi
marshall@fourthcoffee.com
1234567890

Subscriptions

Update

Enterprise:
Microsoft Azure
Quantity
Usage-based
Promotions Applied

Submit

Cancel

7. In the confirmation window, note the **Admin user account** and **Password** information. The information will not be viewable after you leave this page, so copy and store it to send to the customer.

Microsoft Partner Center
Partner with us
Learn more
Find a Partner
Get support
Dashboard

New customer
Account info
Subscriptions
Review
Confirmation

Confirmation
We have received your order. It may take a few minutes to process. To see this order, view this customer's subscriptions. Be sure to copy the following customer and subscription information for your records.

Setup info

Microsoft ID

<tenant ID>

Admin user account

admin@fourthcoffeecsp.onmicrosoft.com

Password

<password>

Please copy and store this password. After you leave this page, you can't view this password again.

Account info

Company
Fourth Coffee
One Microsoft Way
Redmond WA 98052
fourthcoffeecsp.onmicrosoft.com

Primary contact info
Marshall Levi
marshall@fourthcoffee.com
1234567890

Subscriptions

Enterprise:
Microsoft Azure
Subscription ID
Usage-based
<subscription ID>

Done

IMPORTANT

By default, the customer administrator does *not* have access to this Azure subscription and cannot manage it by using the [Azure portal](#). The CSP model was designed for a variety of scenarios, and in some of them (for example, Managed Cloud Services) the customer doesn't need to access the Azure portal. The CSP partner manages the customer's resources instead. If you want to grant the customer access to the Azure CSP subscription, an [additional step](#) is required.

8. Select **Done**.
9. On the Partner Center dashboard, select **Customers**, select the customer that you just created, select **Service management**, and then select **Microsoft Azure Management Portal**.

The screenshot shows the Microsoft Partner Center dashboard. The left sidebar contains a list of navigation items: Subscriptions, Devices, Customer insights, Users and licenses, **Service management** (highlighted with a red box), and Account. Below this is a 'Customers' link with a left arrow. The main content area is titled 'Service management' and is divided into two columns. The left column, 'Administer services', lists links for Azure Active Directory, Office 365, **Microsoft Azure Management Portal** (highlighted with a red box), Visual Studio Marketplace, and Manage Visual Studio Subscriptions. The right column, 'Service health', contains a table of services and their severity.

Service	Severity
Dynamics CRM Online	normal
Exchange Online	normal
Identity Service	normal
Microsoft Dynamics Marketing	normal
Mobile Device Management	normal
Office 365 Portal	normal

The Azure portal opens.

10. To add the customer as an owner of the Azure subscription, see [Assign permissions to an Azure CSP subscription](#).

The screenshot shows the Microsoft Azure portal dashboard. The left sidebar contains a list of navigation items: Dashboard, All resources, Resource groups, App Services, Function Apps, SQL databases, Azure Cosmos DB, Virtual machines, Load balancers, Storage accounts, Virtual networks, and Azure Active Directory. The main content area is titled 'Dashboard' and contains a 'All resources' section with a message 'No resources to display' and a 'Create resources' button. To the right is a 'Quickstart tutorials' section with links to Windows Virtual Machines, Linux Virtual Machines, App Service, Functions, and SQL Database.

NOTE

One CSP customer account can have multiple Azure CSP subscriptions. By default, all subscriptions are named *Microsoft Azure*. To better distinguish them, rename them in Partner Center to something more descriptive (for example, *Production environment* or *HR System*).

PowerShell

Use the following command to create a new customer with PowerShell. To learn how to configure the PowerShell environment for Azure CSP, see [Azure CSP management options](#).

```
$newDefaultAddress = New-PCCustomerDefaultAddress -Country '<country code>' -Region '<region>' -City '<city>' -State '<state>' -
AddressLine1 '<address 1>' -PostalCode <postal code> -FirstName '<first name>' -LastName '<last name>' -PhoneNumber <phone number>

$newBillingProfile = New-PCCustomerBillingProfile -Email '<email>' -Culture '<ex: en.us>' -Language '<ex: en>' -CompanyName '<company
name>' -DefaultAddress $newDefaultAddress

$newCompanyProfile = New-PCCustomerCompanyProfile -Domain '<company name>.onmicrosoft.com'

$newCustomer = New-PCCustomer -BillingProfile $newBillingProfile -CompanyProfile $newCompanyProfile
```

C#

To add a customer, do the following:

1. Instantiate a new **Customer** object.
2. Fill in the **BillingProfile** and **CompanyProfile** information.
3. Add the object to your **IAggregatePartners.Customers** collection by calling **Create()** or **CreateAsync()**.

```
// IAggregatePartner partnerOperations;
var partnerOperations = this.Context.UserPartnerOperations;

var customerToCreate = new Customer()
{
    CompanyProfile = new CustomerCompanyProfile()
    {
        Domain = string.Format(CultureInfo.InvariantCulture,
            "SampleApplication{0}.{1}",
            new Random().Next(),
            this.Context.Configuration.Scenario.CustomerDomainSuffix)
    },
    BillingProfile = new CustomerBillingProfile()
    {
        Culture = "en-us",
        Email = "marshall@fourthcoffee.com",
        Language = "en",
        CompanyName = "Fourth Coffee" + new Random().Next(),
        DefaultAddress = new Address()
        {
            FirstName = "Marshall",
            LastName = "Levi",
            AddressLine1 = "One Microsoft Way",
            City = "Redmond",
            State = "WA",

            Country = "US",
            PostalCode = "98052",
            PhoneNumber = "1234567890"
        }
    }
};

var newCustomer = partnerOperations.Customers.Create(customerToCreate);
```

REST API

The following is a guide to formatting a request to add a customer by using the REST API.

Request

Request syntax

METHOD	REQUEST URI
POST	{baseURI}/v1/customers HTTP/1.1

Request headers

- This API will not produce different results if it is called multiple times.
- Request ID and correlation ID are required.

Request body

The request body has the following *required* properties:

NAME	TYPE	DESCRIPTION
BillingProfile	object	The customer's billing profile information.

NAME	TYPE	DESCRIPTION
CompanyProfile	object	The customer's company profile information.

Billing profile

The CustomerBillingProfile resources needed to create a new customer have the following minimum *required* fields:

NAME	TYPE	DESCRIPTION
email	string	The customer's email address.
culture	string	The customer's preferred culture for communication and currency, such as "en-US."
language	string	The default language. Two-character language codes (en, fr, and so on) are supported.
company_name	string	The registered company or organization name.
default_address	address	The registered address of the customer's company or organization.

Company profile

The CustomerCompanyProfile resource needed to create a new customer has the following minimum *required* field:

NAME	TYPE	DESCRIPTION
domain	string	The customer's domain name, such as <i>contoso.onmicrosoft.com</i> .

Request example

POST https://api.partnercenter.microsoft.com/v1/customers HTTP/1.1

Authorization: Bearer <token>

Accept: application/json

MS-RequestId: 94e4e214-6b06-4fb7-96d1-94d559f9b47f

MS-CorrelationId: ab993325-1605-4cf4-bac4-fb584142a31b

X-Locale: en-US

Content-Type: application/json

Host: api.partnercenter.microsoft.com

Content-Length: 789

Expect: 100-continue

Connection: Keep-Alive

```
{
  "Id": null,
  "CommerceId": null,
  "CompanyProfile": {
    "TenantId": null,
    "Domain": "SampleApplication1047419322.onmicrosoft.com",
    "CompanyName": null,
    "Attributes": {
      "ObjectType": "CustomerCompanyProfile"
    }
  },
  "BillingProfile": {
    "Id": null,
    "FirstName": null,
    "LastName": null,
    "Email": "marshall@fourthcoffee.com",
    "Culture": "en-us",
    "Language": "en",
    "CompanyName": "Fourth Coffee",
    "DefaultAddress": {
      "Country": "US",
      "Region": null,
      "City": "Redmond",
      "State": "WA",
      "AddressLine1": "One Microsoft Way",
      "AddressLine2": null,
      "PostalCode": "98052",
      "FirstName": "Marshall",
      "LastName": "Levi",
      "PhoneNumber": "1234567890"
    },
    "Attributes": {
      "ObjectType": "CustomerBillingProfile"
    }
  },
  "RelationshipToPartner": "none",
  "AllowDelegatedAccess": null,
  "UserCredentials": null,
  "CustomDomains": null,
  "AssociatedPartnerId": null,
  "Attributes": {
    "ObjectType": "Customer"
  }
}
```

Response

If the request is successful, this API returns a customer resource for the new customer. Save the customer ID and Azure AD details for future use, such as account management.

Response example

HTTP/1.1 201 Created
Content-Length: 834
Content-Type: application/json; charset=utf-8
MS-CorrelationId: ab993325-1605-4cf4-bac4-fb584142a31b
MS-RequestId: 94e4e214-6b06-4fb7-96d1-94d559f9b47f
MS-CV: ObwhuhD2tUKJoM+Z.0
MS-ServerId: 202010223
Date: Tue, 14 Feb 2017 20:06:02 GMT

```
{
  "id": "dfd8cc0a-c592-468c-8461-869a38d24738",
  "commerceId": "0a4ce58a-6f96-4273-8035-d9c7d31b9ba4",
  "companyProfile": {
    "tenantId": "dfd8cc0a-c572-468c-8461-869a38d24734",
    "domain": "SampleApplication1047419322.onmicrosoft.com",
    "attributes": {
      "objectType": "CustomerCompanyProfile"
    }
  },
  "billingProfile": {
    "id": "d17c0275-da92-5c33-9032-782ef1d0b69b",
    "email": "marshall@fourthcoffee.com",
    "culture": "en-us",
    "language": "en",
    "companyName": "Fourth Coffee",
    "defaultAddress": {
      "country": "US",
      "city": "Redmond",
      "state": "WA",
      "addressLine1": "One Microsoft Way",
      "postalCode": "98052",
      "firstName": "Marshall",
      "lastName": "Levi",
      "phoneNumber": "1234567890"
    },
    "attributes": {
      "etag": "1234567890",
      "objectType": "CustomerBillingProfile"
    }
  },
  "relationshipToPartner": "none",
  "userCredentials": {
    "userName": "admin",
    "password": "P@ssword12345"
  },
  "attributes": {
    "objectType": "Customer"
  }
}
```

Next steps

- [Learn how to add an existing Azure customer to your Partner Center account](#)
- [Learn how to assign Azure CSP subscription permissions to the customer](#)

Add an existing customer to Azure Cloud Solution Provider

9/19/2017 • 2 min to read • [Edit Online](#)

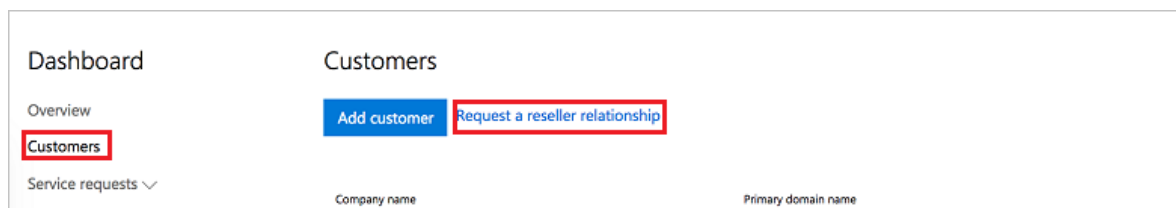
Learn how to add an existing Microsoft Partner Center customer account, and then associate it with an active Azure subscription. You can do this in Partner Center and the Azure portal, or use one of the Partner Center APIs. To add an existing Partner Center customer account, you must have an existing Partner Center account and an active Azure subscription.

In this article, we describe how to add an existing Partner Center account for Azure Cloud Solution Provider (Azure CSP) integration by using one of the following options:

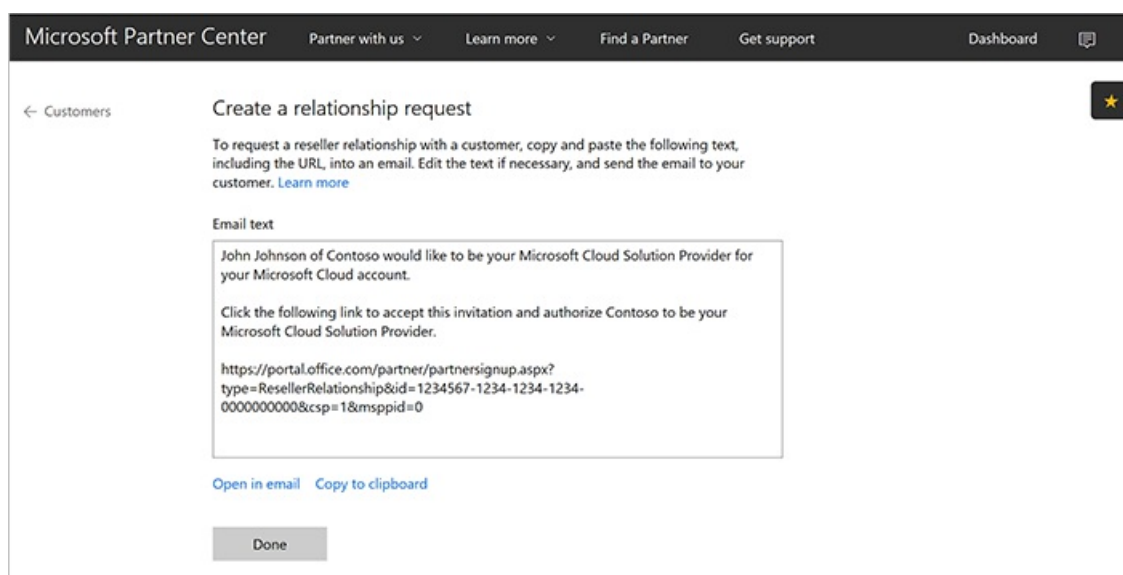
- Partner Center web UI
- PowerShell
- C#
- REST API

Partner Center web UI

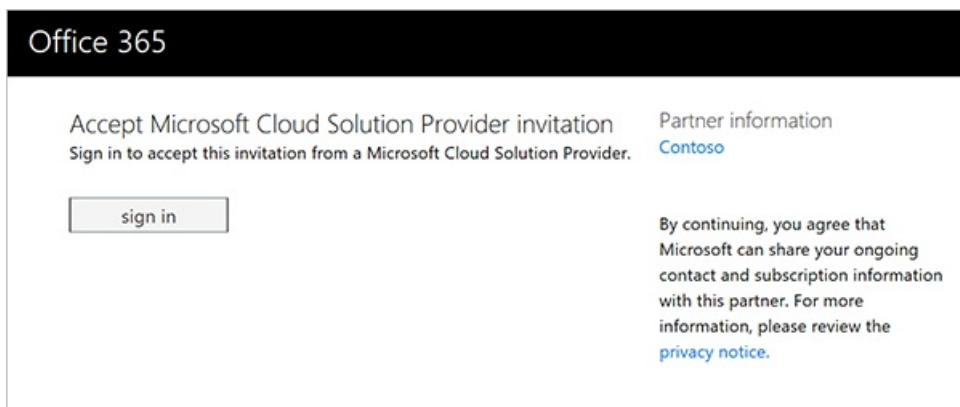
1. In Partner Center, on the **Dashboard** menu, select **Customers** > **Request a reseller relationship**.



2. On the **Create a relationship request** page, copy the email text that appears. You can edit or delete the first two paragraphs of the email text, but keep the link intact. The link in the email text applies specifically to your reseller account. The link doesn't include any customer information. You can reuse the same link each time you request a reseller relationship.



3. Send the relationship request email to your customer. To accept the relationship request, the customer's global admin selects the invitation link, and then signs in.



4. On the **Partner Relationships** page, the customer can review their partners for their Azure services. This includes which partners who have delegated admin rights for each service.

NOTE

In the multi-channel model, for each service, a customer can have only one partner who has delegated admin rights for Azure CSP transactions. The customer might also have granted rights to other partners who don't participate in the Azure CSP program.

5. The invited customer appears in the partner's customer list in Partner Center. The partner can then provision and manage the Azure CSP subscriptions for the customer.

Customers have the option to remove delegated admin rights from any or all partners (including those in the Azure CSP program). Customers can do this in their own admin portal. The customer account and commercial relationship with the partner still applies, unless it is renegotiated by the customer and the partner. The partner is still responsible for providing support to the customer. A partner also must still adhere to the terms of the Microsoft Cloud Reseller Agreement, even if a customer removes delegated admin rights from the partner. In this scenario, if the customer requires support, the partner must contact Microsoft Support to submit a service request on behalf of the customer.

To re-establish delegated admin rights that the customer has revoked or removed, repeat the process of requesting a relationship.

PowerShell

To request a relationship by using PowerShell, get the invitation URL as described in steps 1 and 2 in the [Partner Center web UI](#) section. Then, in PowerShell, run the following command:

```
$inviteURL = (New-PCRelationshipRequest).url
```

C#

1. Use your **IAggregatePartner.Customers** collection.
2. Call the **RelationshipRequest** property.
3. Call the **Get()** or the **GetAsync()** method.

```
// IAggregatePartner partnerOperations;  
  
var customerRelationshipRequest = partnerOperations.Customers.RelationshipRequest.Get();
```

REST API

Request

Request syntax

METHOD	REQUEST URI
GET	{baseUrl}/v1/customers/relationshiprequests HTTP/1.1

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers/relationshiprequests HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: f281903b-7b57-4883-8bc1-5caca1b22a37
MS-CorrelationId: dd048ef2-d737-4b11-a885-e1eab8fc94ba
Connection: Keep-Alive
```

Response

If the request is successful, this method returns a **CustomerRelationshipRequest** object in the response.

Response example

```
HTTP/1.1 200 OK
Content-Length: 204
Content-Type: application/json; charset=utf-8
MS-CorrelationId: dd048ef2-d737-4b11-a885-e1eab8fc94ba
MS-RequestId: f281903b-7b57-4883-8bc1-5caca1b22a37
Date: Mon, 21 Mar 2016 05:39:51 GMT

{
  "url": "https://portal.office.com/partner/partnersignup.aspx?type=ResellerRelationship&id=3b33e682-00c3-41ee-9dd2-a548adf56438&csp=1&msppid=0",
  "attributes": {
    "objectType": "CustomerRelationshipRequest"
  }
}
```

Next steps

- Learn how to [assign Azure CSP subscription permissions to the customer](#).
- Learn how to [create a new Azure CSP customer in Partner Center](#).

Add subscriptions to a customer account

9/19/2017 • 3 min to read • [Edit Online](#)

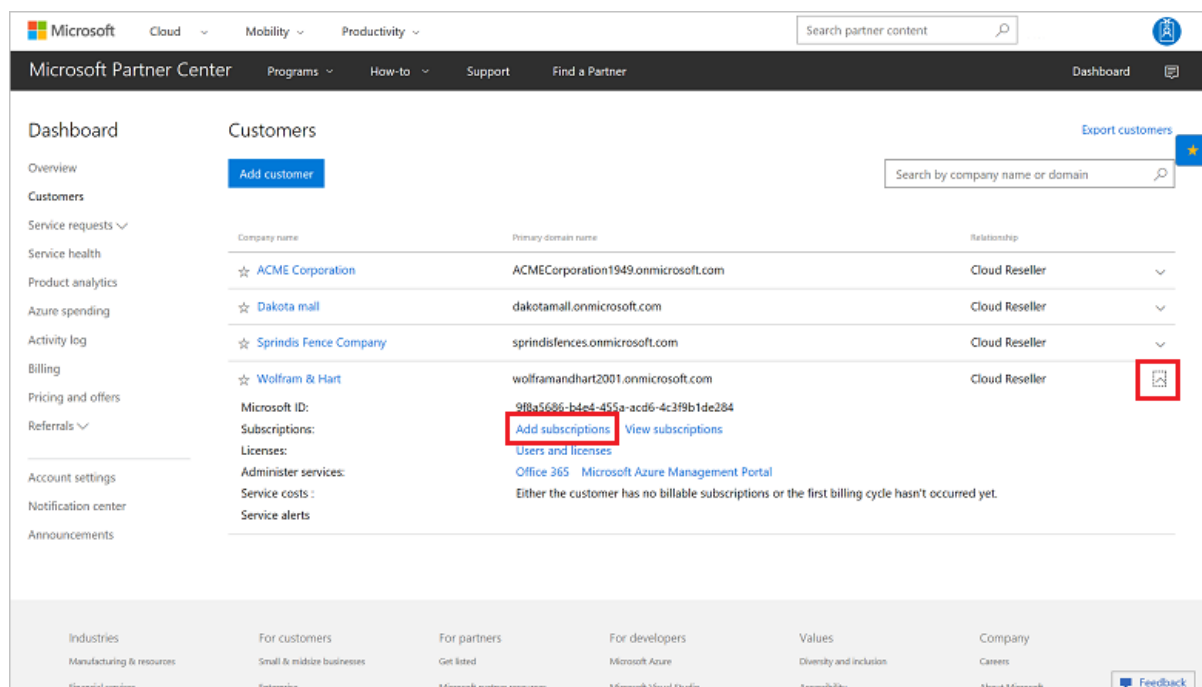
Learn how to add and edit the subscriptions and licenses that are available to a specific customer. You can use one of the following options:

- Partner Center web UI
- PowerShell
- C#
- REST API

Partner Center web UI

Order a new subscription

1. In Microsoft Partner Center, go to the **Customers** page.
2. In the far-right column, select the drop-down box. This shows expanded details for the customer. Select **Add subscriptions**.



The screenshot shows the Microsoft Partner Center interface. The top navigation bar includes the Microsoft logo, tabs for Cloud, Mobility, and Productivity, a search bar for partner content, and a user profile icon. The main header displays 'Microsoft Partner Center' with links to Programs, How-to, Support, and Find a Partner. The left sidebar contains a 'Dashboard' menu with options like Overview, Customers, Service requests, Service health, Product analytics, Azure spending, Activity log, Billing, Pricing and offers, Referrals, Account settings, Notification center, and Announcements. The main content area is titled 'Customers' and features an 'Add customer' button and a search bar. A table lists customers with columns for Company name, Primary domain name, and Relationship. The customer 'Wolfram & Hart' is highlighted, and its details are expanded. The 'Add subscriptions' link is highlighted with a red box. The 'Add subscriptions' link is located under the 'Subscriptions' section of the customer details.

Company name	Primary domain name	Relationship
☆ ACME Corporation	ACMECorporation1949.onmicrosoft.com	Cloud Reseller
☆ Dakota mail	dakotamail.onmicrosoft.com	Cloud Reseller
☆ Sprindis Fence Company	sprindisfences.onmicrosoft.com	Cloud Reseller
☆ Wolfram & Hart	wolframandhart2001.onmicrosoft.com	Cloud Reseller

Microsoft ID: 9f8a5686-b4e4-455a-acd6-4c3f9b1de284

Subscriptions: **Add subscriptions** View subscriptions

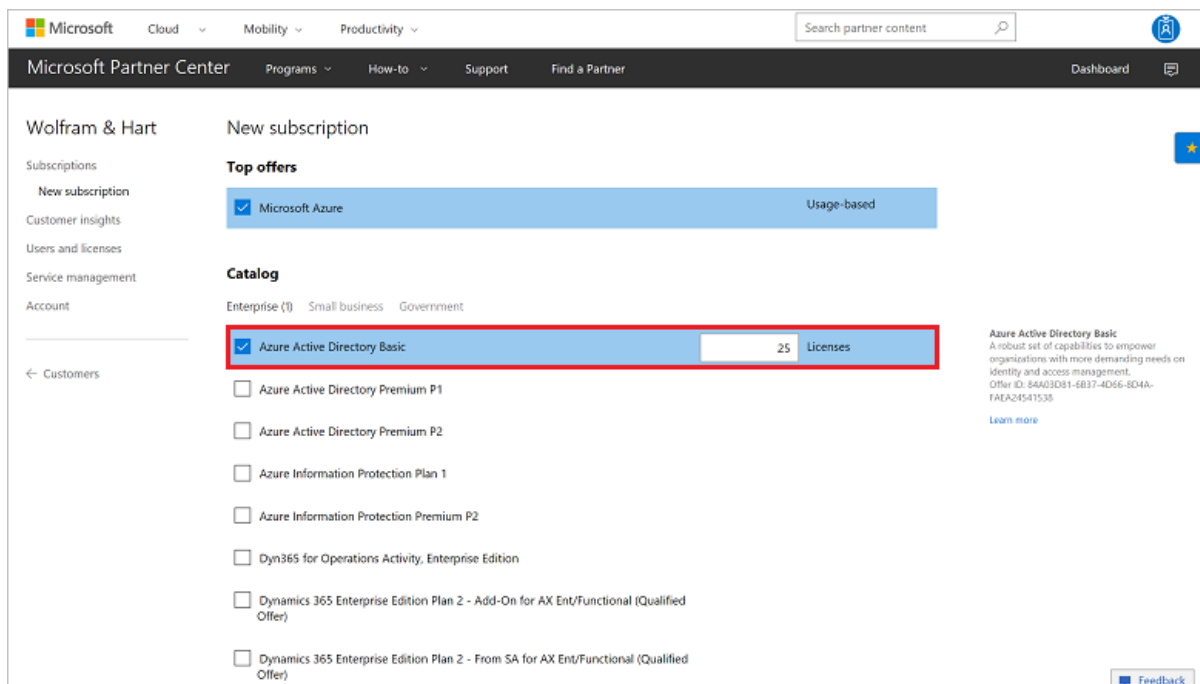
Licenses: Users and licenses

Administer services: Office 365 Microsoft Azure Management Portal

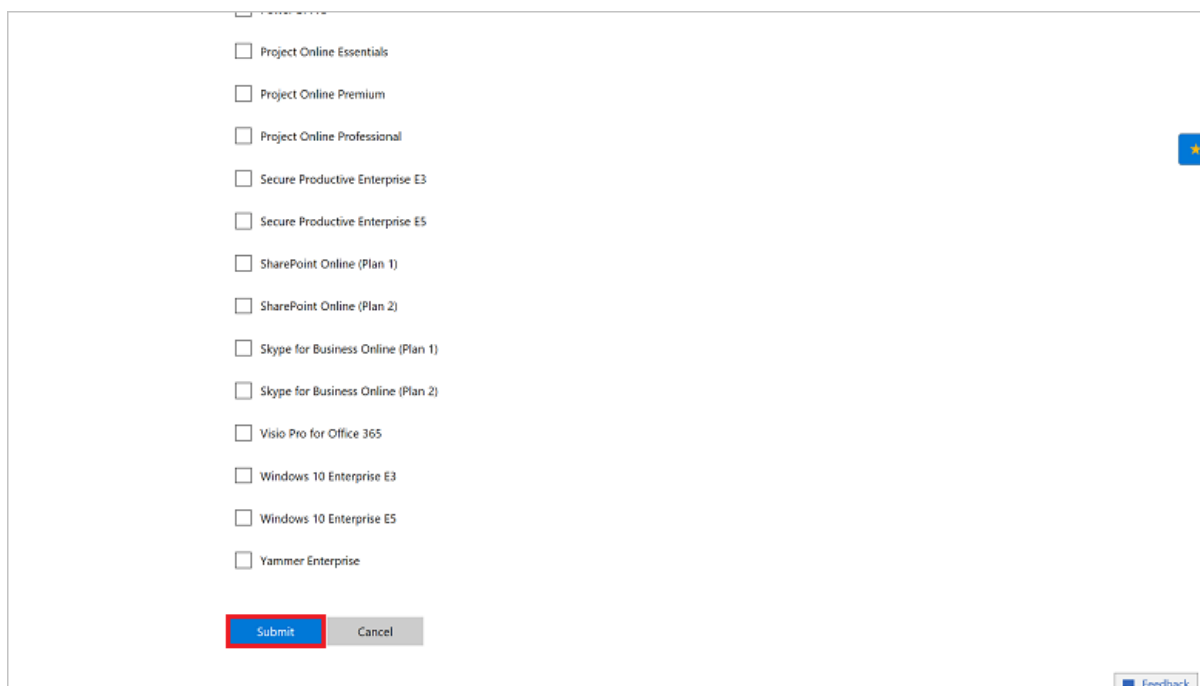
Service costs: Either the customer has no billable subscriptions or the first billing cycle hasn't occurred yet.

Service alerts

3. On the **New subscription** page, select the offer or offers, and the number of licenses.



4. Select **Submit**.



The new subscription now appears in the customer's **Subscriptions** list.

PowerShell

Get a customer

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'
```

Order a new subscription

```
# Get offer
$offer = Get-PCOffer -countryid '<country two-digit ID>' -offerid '<offer ID GUID>'

# Create the OrderLineItem
$lineItems = @()
$lineItems += [OrderLineItem]::new()
$lineItems[0].LineNumber = 0
$lineItems[0].FriendlyName = '<friendly name>'
$lineItems[0].OfferId = $offer.id
$lineItems[0].Quantity = <quantity>

# Send order
New-PCOrder -tenantid $customer.id -LineItems $lineItems
```

C#

1. Instantiate an **Order** object. To record the customer, set the **ReferenceCustomerId** property to the customer ID.
2. Create a list of **OrderLineItem** objects. Assign the list to the order's **LineItems** property. Each order line item contains the purchase information for one offer. You must have at least one order line item.
3. To get an interface for order operations:
 - a. To identify the customer, call the **IAggregatePartner.Customers.ById** method with the customer ID.
 - b. To retrieve the interface, use the **Orders** property.
4. To create the order, call the **Create** or the **CreateAsync** method.

```
IAggregatePartner partnerOperations;
string customerId;
string offerId;

var order = new Order()
{
    ReferenceCustomerId = customerId,
    LineItems = new List<OrderLineItem>()
    {
        new OrderLineItem()
        {
            OfferId = offerId,
            FriendlyName = "new offer purchase",
            Quantity = 5
        }
    }
};

var createdOrder = partnerOperations.Customers.ById(customerId).Orders.Create(order);
```

REST API

Request

Request syntax

METHOD	REQUEST URI
POST	{baseUrl}/v1/customers/{customer-id}/orders HTTP/1.1

URI parameters

Use the following path parameter to identify the customer.

NAME	TYPE	DESCRIPTION
customer-id	guid	A GUID-format customer ID that identifies the customer.

Request body

Order

The following table describes the **Order** properties in the request body.

NAME	TYPE	REQUIRED?	DESCRIPTION
id	string	N	An order identifier that is supplied when the order is successfully created.
referenceCustomerId	string	Y	The customer identifier.
billingCycle	string	N	The frequency with which the partner is billed for this order. The default is Monthly . The frequency is applied when the order is successfully created. Supported values are the member names found in BillingCycleType . Note that the annual billing feature is not yet generally available. Support for annual billing is coming soon.
lineItems	array of objects	Y	An array of OrderLineItem resources.
creationDate	string	N	The date the order was created, in date/time format. The value for creationDate is applied when the order is successfully created.
attributes	object	N	Contains <code>ObjectType</code> : <code>Order</code> .

OrderLineItem

The following table describes the **OrderLineItem** properties in the request body.

NAME	TYPE	REQUIRED?	DESCRIPTION
lineItemNumber	int	Y	Each line item in the collection has a unique line number, counting up from 0 to <i>count</i> -1.
offerId	string	Y	The offer identifier.

NAME	TYPE	REQUIRED?	DESCRIPTION
subscriptionId	string	N	The subscription identifier.
parentSubscriptionId	string	N	The ID of the parent subscription in an add-on offer. Applies to PATCH only.
friendlyName	string	N	The display name for the subscription, which is defined by the partner. You can use the display name to help disambiguate subscriptions.
quantity	int	Y	The number of licenses for a license-based subscription.
partnerIdOnRecord	string	N	When an indirect provider places an order on behalf of an indirect reseller, populate this field <i>only</i> with the Microsoft Partner Network (MPN) ID of the indirect reseller (never use the MPN ID of the indirect provider). This ensures proper accounting for incentives.
attributes	object	N	Contains <code>ObjectType</code> : <code>OrderLineItem</code> .

Request example

POST https://api.partnercenter.microsoft.com/v1/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/orders HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: 57870501-203b-468e-8a63-078a3826d8ec
MS-CorrelationId: 9c272436-538d-4dd4-a421-c811e004784c
X-Locale: en-US
Content-Type: application/json
Host: api.partnercenter.microsoft.com
Content-Length: 405
Expect: 100-continue
Connection: Keep-Alive

```
{
  "Id": null,
  "ReferenceCustomerId": "4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04",
  "BillingCycle": "unknown",
  "LineItems": [{
    "LineItemNumber": 0,
    "OfferId": "84A03D81-6B37-4D66-8D4A-FAEA24541538",
    "SubscriptionId": null,
    "ParentSubscriptionId": null,
    "FriendlyName": "new offer purchase",
    "Quantity": 5,
    "PartnerIdOnRecord": null,
    "Attributes": {
      "ObjectType": "OrderLineItem"
    }
  }
],
  "CreationDate": null,
  "Attributes": {
    "ObjectType": "Order"
  }
}
```

Response

If the request is successful, the response body contains the populated **Order** resource.

Response example

HTTP/1.1 201 Created
Content-Length: 801
Content-Type: application/json; charset=utf-8
MS-CorrelationId: 9c272436-538d-4dd4-a421-c811e004784c
MS-RequestId: 57870501-203b-468e-8a63-078a3826d8ec
MS-CV: VcfS+fqdQUW8Nap6.0
MS-ServerId: 030020525
Date: Thu, 30 Mar 2017 17:43:08 GMT

```
{
  "id": "074bd849-9106-405c-9923-fa061839d487",
  "referenceCustomerId": "4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04",
  "billingCycle": "monthly",
  "lineItems": [ {
    "lineItemNumber": 0,
    "offerId": "84A03D81-6B37-4D66-8D4A-FAEA24541538",
    "subscriptionId": "1DA3295E-C59A-476C-A7E7-D7E981FE79BE",
    "friendlyName": "new offer purchase",
    "quantity": 5,
    "links": {
      "subscription": {
        "uri": "/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/subscriptions/1DA3295E-C59A-476C-A7E7-D7E981FE79BE",
        "method": "GET",
        "headers": []
      }
    }
  }
],
  "creationDate": "2017-03-30T10:43:07.157-07:00",
  "links": {
    "self": {
      "uri": "/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/orders/074bd849-9106-405c-9923-fa061839d487",
      "method": "GET",
      "headers": []
    }
  },
  "attributes": {
    "etag": "eyJpZCI6IjA3NGJkODQ5LTkxMDYtNDA1Yy05OTIzLWZhMDYxODM5ZDQ4NyIsInZlcnNpb24iOiJF9",
    "objectType": "Order"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Get a subscription by ID

9/19/2017 • 1 min to read • [Edit Online](#)

This guide covers how to retrieve a specific subscription by providing the customer and subscription IDs.

PowerShell

To retrieve a subscription by ID, enter the following commands:

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'

Get-PCSubscription -tenantid $customer.id -subscriptionid '<subscription id GUID>'
```

C#

To get a subscription by ID, do the following:

1. Get an interface to the subscription operations by calling both of the following:
 - The **IAggregatePartner.Customers.ById** method with the customer ID to identify the customer.
 - The **Subscriptions.ById** method to identify the subscription.
2. Use the interface to retrieve the subscription details by calling **Get**.

```
// IAggregatePartner partnerOperations;
// string selectedCustomerId;
// string subscriptionID;

var subscriptionDetails = partnerOperations.Customers.ById(selectedCustomerId).Subscriptions.ById(subscriptionID).Get();
```

REST API

Request

Request syntax

METHOD	REQUEST URI
GET	<code>{baseUrl}/v1/customers/{customer-tenant-id}/subscriptions/{id-for-subscription}</code> HTTP/1.1

URI parameter

To get the subscription, use the following *required* query parameters:

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	A GUID that corresponds to the customer.
id-for-subscription	guid	A GUID that corresponds to the subscription.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/subscriptions/A356AC8C-E310-44F4-BF85-C7F29044AF99 HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: 8f489776-a3f3-47cb-91c3-538e1f70f560
MS-CorrelationId: e72e1dc3-4abd-4ce0-908b-d23fdaedcb28
X-Locale: en-US
Host: api.partnercenter.microsoft.com
```

Response

If the request is successful, this method returns a **Subscription** resource in the response body.

Response example (standard subscription)

```
HTTP/1.1 200 OK
Content-Length: 833
Content-Type: application/json; charset=utf-8
MS-CorrelationId: e72e1dc3-4abd-4ce0-908b-d23fdaedcb28
MS-RequestId: 8f489776-a3f3-47cb-91c3-538e1f70f560
MS-CV: 7v11Wa//5EuGEo+A.0
MS-ServerId: 202010406
Date: Fri, 27 Jan 2017 21:51:40 GMT

{
  "id": "A356AC8C-E310-44F4-BF85-C7F29044AF99",
  "offerId": "MS-AZR-0145P",
  "offerName": "Microsoft Azure",
  "friendlyName": "Microsoft Azure",
  "quantity": 1,
  "unitType": "Usage-based",
  "creationDate": "2016-05-10T07:30:05.427Z",
  "effectiveStartDate": "2016-05-10T00:00:00Z",
  "commitmentEndDate": "9999-12-10T00:00:00Z",
  "status": "active",
  "autoRenewEnabled": false,
  "billingType": "usage",
  "contractType": "subscription",
  "links": {
    "offer": {
      "uri": "/offers/MS-AZR-0145P?country=US",
      "method": "GET",
      "headers": []
    },
    "self": {
      "uri": "/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/subscriptions/A356AC8C-E310-44F4-BF85-C7F29044AF99",
      "method": "GET",
      "headers": []
    }
  },
  "orderId": "B23FDEDD-D6BD-415A-8B71-3624C81C9644",
  "attributes": {
    "etag": "eyJpZCI6ImFzNTZhYzhjLWUzMtAtNDRmNC1iZjg1LWw3ZjI5MDQ0YWY5OSIsInZlcnNpb24iOiJ9",
    "objectType": "Subscription"
  }
}
```

Response example (add-on response)

HTTP/1.1 200 OK

Content-Length: 1132

Content-Type: application/json; charset=utf-8

MS-CorrelationId: 6eacec93-852d-4167-9d96-c57809bea7ed

MS-RequestId: 22bfd0fb-d1e6-4a8f-aa1a-124b7c820d80

MS-CV: cmde2DtbuUWi8JLq.0

MS-ServerId: 201022015

Date: Fri, 27 Jan 2017 00:12:53 GMT

```
{
  "id": "968BA1CF-C146-4ADF-A300-308DCF718EEE",
  "offerId": "2828BE95-46BA-4F91-B2FD-0BEF192ECF60",
  "offerName": "Exchange Online Archiving for Exchange Online",
  "friendlyName": "Some friendly name",
  "quantity": 2,
  "unitType": "Licenses",
  "parentSubscriptionId": "1C2B75C1-74A5-472A-A729-7F8CEFC477F9",
  "creationDate": "2017-01-25T23:01:08.693Z",
  "effectiveStartDate": "2017-01-25T00:00:00Z",
  "commitmentEndDate": "2018-02-10T00:00:00Z",
  "status": "active",
  "autoRenewEnabled": true,
  "billingType": "license",
  "contractType": "subscription",
  "links": {
    "offer": {
      "uri": "/offers/2828BE95-46BA-4F91-B2FD-0BEF192ECF60?country=US",
      "method": "GET",
      "headers": []
    },
    "parentSubscription": {
      "uri": "/customers/4d3cf487-70f4-4e1e-9ff1-b2bfc8d9f04/subscriptions/1C2B75C1-74A5-472A-A729-7F8CEFC477F9",
      "method": "GET",
      "headers": []
    },
    "self": {
      "uri": "/customers/4d3cf487-70f4-4e1e-9ff1-b2bfc8d9f04/subscriptions/968BA1CF-C146-4ADF-A300-308DCF718EEE",
      "method": "GET",
      "headers": []
    }
  },
  "orderId": "CF3B0E37-BE0B-4CDD-B584-D1A97D98A922",
  "attributes": {
    "etag": "eyJpZCI6IjQOGJhMWNmLWMxNDYtNGFkZi1hMzAwLTMwOGRjZjcxOGVZSIInZlcnNpb24iOiJF9",
    "objectType": "Subscription"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- View a list of [Azure CSP integration scenarios](#).

Update a subscription nickname

9/19/2017 • 2 min to read • [Edit Online](#)

Learn how to update the nickname for a subscription. Subscription nicknames help create common terminology between the provider and the customer. They also help differentiate between subscriptions in a customer's account. To update a subscription nickname, you can use one of the following options:

- Partner Center web UI
- PowerShell
- C#
- REST API

Partner Center web UI

1. In Microsoft Partner Center, on the **Dashboard** menu, select **Customer**. Select the customer.
2. Select **Subscriptions**, and then select the subscription.
3. Under **Details**, in the **Subscription nickname** box, enter the new nickname for the subscription. Then, select **Submit**.

The screenshot shows the Microsoft Partner Center interface. At the top, there's a navigation bar with the Microsoft logo, 'Cloud', 'Mobility', and 'Productivity' dropdowns, a search bar, and a user profile icon. Below this is a dark header with 'Microsoft Partner Center' and links for 'Partner with us', 'Learn more', 'Find a Partner', and 'Dashboard'. The main content area shows the 'Fabrikam' customer's 'Microsoft Azure' subscription. The 'Details' section is active, showing the 'Subscription nickname' field with the value 'Microsoft Azure'. The 'Quantity' is 'Usage-based' and the 'Status' is 'Active'. At the bottom, there are 'Submit' and 'Cancel' buttons. The 'Submit' button is highlighted with a red box.

PowerShell

```
$subscription | Set-PCSubscription -tenantid $customer.id -friendlyName '<friendly name>'
```

C#

1. [Get the subscription](#).
2. Change the subscription's **FriendlyName** property.
3. Use your **IPartner.Customers** collection and call the **ById()** method.
4. Call the **Subscriptions** property, followed by the **ById()** method, and then the **Patch()** method.

```
// IAggregatePartner partnerOperations;  
// var SelectedcustomerId as string;  
  
ResourceCollection<Subscription> customerSubscriptions = partnerOperations.Customers.ById(selectedCustomerId).Subscriptions.Get();  
Subscription selectedSubscription = customerSubscriptions.Items.FirstOrDefault(sub => sub.Status == SubscriptionStatus.Active);  
  
// Apply changes to the subscription.  
  
var updatedSubscription =  
partnerOperations.Customers.ById(selectedCustomerId).Subscriptions.ById(selectedSubscription.Id).Patch(selectedSubscription);
```

REST API

Request

Request syntax

METHOD	REQUEST URI
PATCH	{baseUrl}/v1/customers/{customer-tenant-id}/subscriptions/{id-for-subscription} HTTP/1.1

URI parameters

The following query parameters are required for updating the subscription nickname.

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	A GUID that represents the customer.
id-for-subscription	guid	A GUID that represents the subscription.

Request body

A full **Subscription** resource is required in the request body. Ensure that the **FriendlyName** property has been updated.

Request example

```
PATCH https://api.partnercenter.microsoft.com/v1/customers/<customer-tenant-id>/subscriptions/<subscriptionID> HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: ca7c39f7-1a80-43bc-90d8-ee7d1cad3831
MS-CorrelationId: ec8f62e5-1d92-47e9-8d5d-1924af105f2c
Content-Type: application/json
Content-Length: 1029
Expect: 100-continue
Connection: Keep-Alive
```

```
{
  "Id": "<subscriptionID>",
  "FriendlyName": "nickname",
  "Quantity": 2,
  "UnitType": "none",
  "ParentSubscriptionId": null,
  "CreationDate": "2015-11-25T06:41:12Z",
  "EffectiveStartDate": "2015-11-24T08:00:00Z",
  "CommitmentEndDate": "2016-12-12T08:00:00Z",
  "Status": "active",
  "AutoRenewEnabled": false,
  "BillingType": "none",
  "PartnerId": null,
  "ContractType": "subscription",
  "OrderId": "6183db3d-6318-4e52-877e-25806e4971be",
  "Attributes": {
    "Etag": "<etag>",
    "ObjectType": "Subscription"
  }
}
```

Response

If the request was successful, this method returns updated **Subscription** resource properties in the response body.

Response example

```
PATCH http://partnerapi.store.microsoft.com/v1/customers/<customer-tenant-id>/subscriptions/<subscriptionID> HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-Contract-Version: v1
MS-RequestId: ca7c39f7-1a80-43bc-90d8-ee7d1cad3831
MS-CorrelationId: ec8f62e5-1d92-47e9-8d5d-1924af105f2c
Content-Type: application/json
Content-Length: 1029
Expect: 100-continue
Connection: Keep-Alive
```

```
{
  "Id": "<subscriptionID>",
  "FriendlyName": "nickname",
  "Quantity": 2,
  "UnitType": "none",
  "ParentSubscriptionId": null,
  "CreationDate": "2015-11-25T06:41:12Z",
  "EffectiveStartDate": "2015-11-24T08:00:00Z",
  "CommitmentEndDate": "2016-12-12T08:00:00Z",
  "Status": "active",
  "AutoRenewEnabled": false,
  "BillingType": "none",
  "PartnerId": null,
  "ContractType": "subscription",
  "Links": {
    "Offer": {
      "Uri": "/v1/offers/0CCA44D6-68E9-4762-94EE-31ECE98783B9",
      "Method": "GET",
      "Headers": []
    },
    "Entitlement": {
      "Uri": "/entitlements?key=<key>",
      "Method": "GET",
      "Headers": []
    },
    "Self": {
      "Uri": "/subscriptions?key=<key>",
      "Method": "GET",
      "Headers": []
    }
  },
  "OrderId": "6183db3d-6318-4e52-877e-25806e4971be",
  "Attributes": {
    "Etag": "<etag>",
    "ObjectType": "Subscription"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Assign and manage permissions within an Azure subscription

9/19/2017 • 5 min to read • [Edit Online](#)

When customers are first created in Partner Center, they do not have access to the Azure subscription by default. The Azure Cloud Solution Provider (CSP) model was designed for various scenarios, and in some of them (for example, managed cloud services), customers don't need to access the Azure portal. The CSP partner manages customer resources instead.

If you want to grant a customer access to an Azure CSP subscription, an additional step is required. Over time, you might also want to add users with varying levels of permissions to an Azure subscription.

TIP

If you grant a customer access to the Azure portal, learn about [Azure Resource Manager Policies](#). You can use the policies to limit which Azure services and regions are available to CSP customers in the Azure portal.

This guide covers how to grant customer access from the Azure portal, PowerShell, and the REST API.

Web UI

1. Open the [Azure portal](#) in either of the following ways:

- Go to <https://portal.azure.<customer tenant name>.com/>.
- From the Partner Center dashboard, go to **Customers** > **<customer name>** > **Service Management** > **Microsoft Azure Management Portal**.

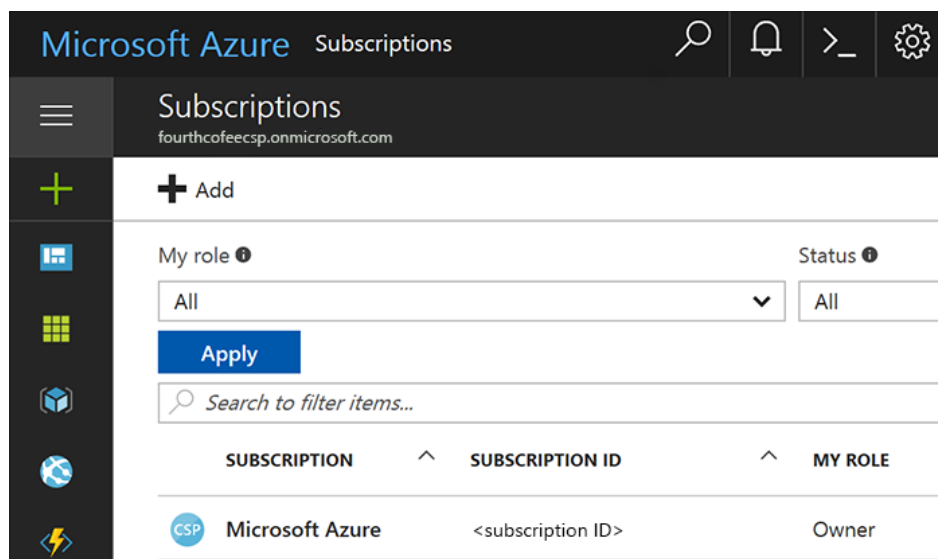
2. Sign in.

Add access permissions

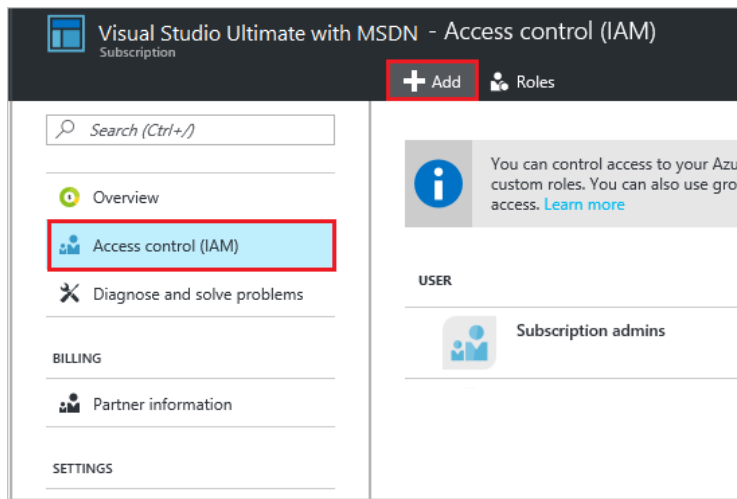
1. In the left pane, select **Subscriptions**, and then select the subscription you want the admin to access.

NOTE

To access the **Subscriptions** button, you might need to select **More services** at the bottom of the pane.



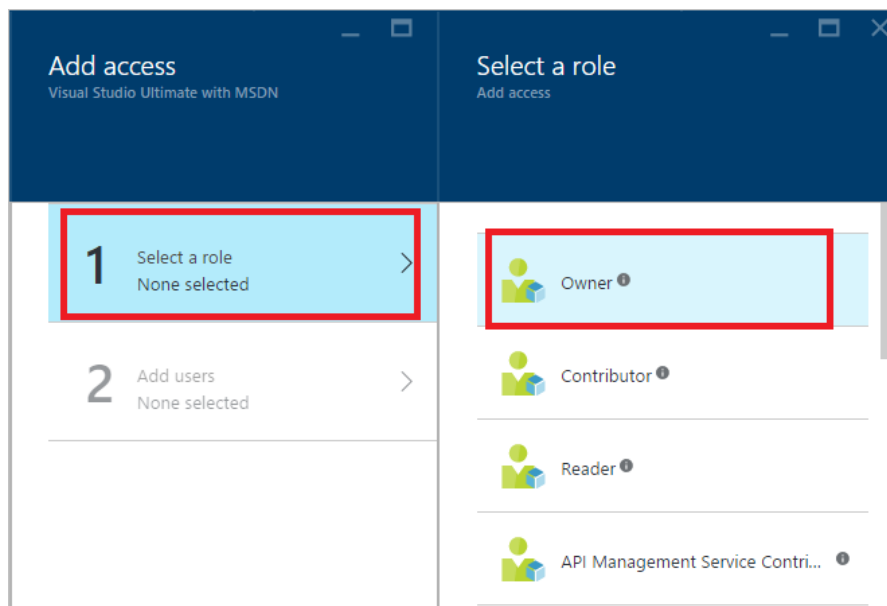
2. On the **Subscription** blade, select **Access control (IAM)**, and then select the **Add** button.



3. On the **Add access** blade, select **Select a role**, and then select the role that you want to assign.

- **Owner:** Can manage all aspects of a subscription, including giving access to other users via this process. If this is a newly created customer, you probably want to give the customer admin *Owner* permissions.
- **Contributor:** Can manage all aspects of a subscription *except* giving access to other users.
- **Reader:** Can view all aspects of a subscription, but cannot make any changes.

The roles listed here are the most basic that you can assign to users in the Azure portal. For a more detailed list, see [Built-in roles for Azure role-based access control](#).



4. In the **User** field, type the email address of the user you want to add. The user appears below the search box. Select the user, and then select **Submit**. The user now has access to the subscription.

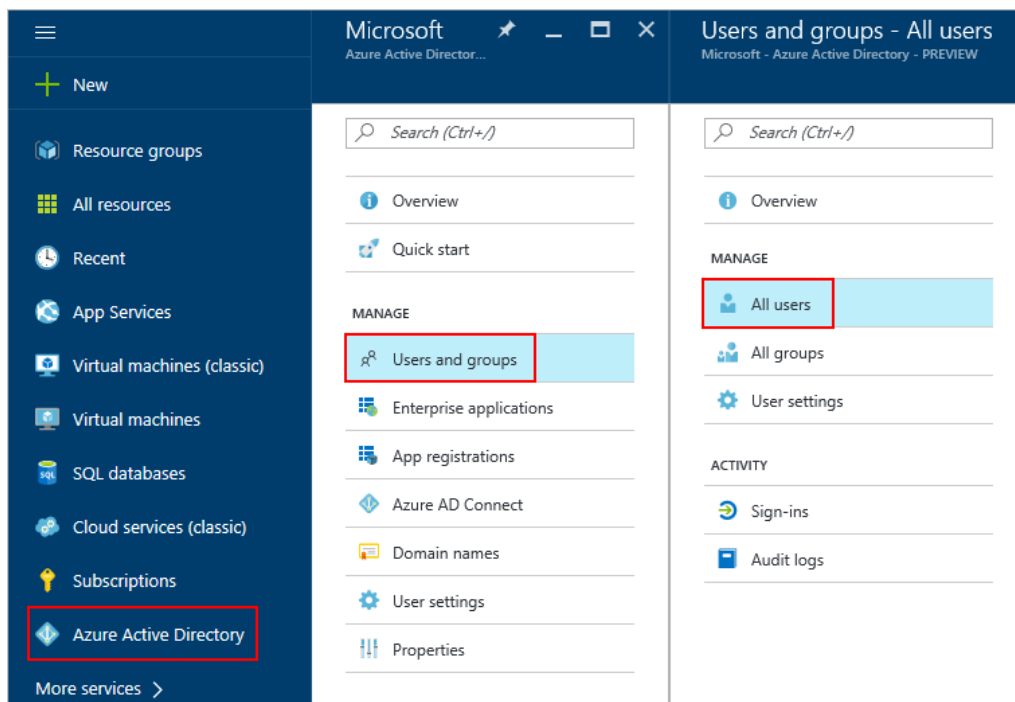
View access permissions

You can also check a user's access permissions through the Azure portal.

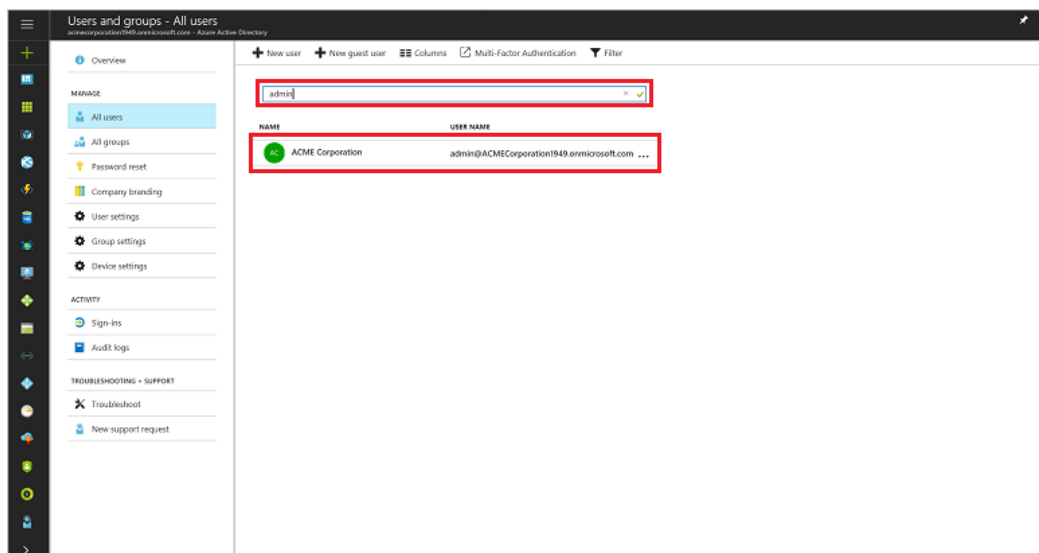
1. In the left pane, select **Azure Active Directory** > **Users and groups** > **All users**.

NOTE

To access the **Azure Active Directory** button, you might need to select **More services** at the bottom of the pane.

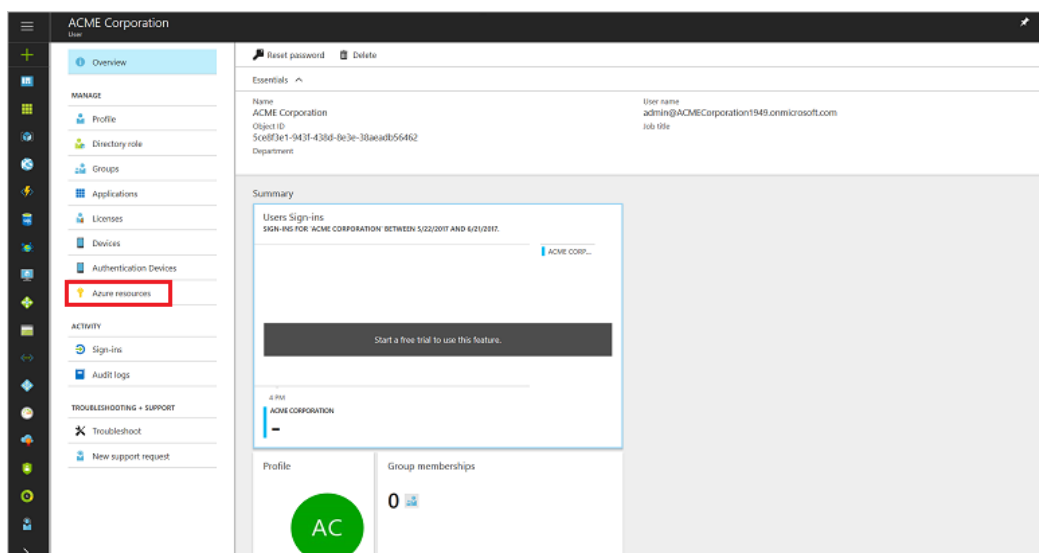


2. In the search box, search for the user by name or username and then, in the results list, select the user.



A new user blade opens.

3. In the left pane, select **Azure resources**.

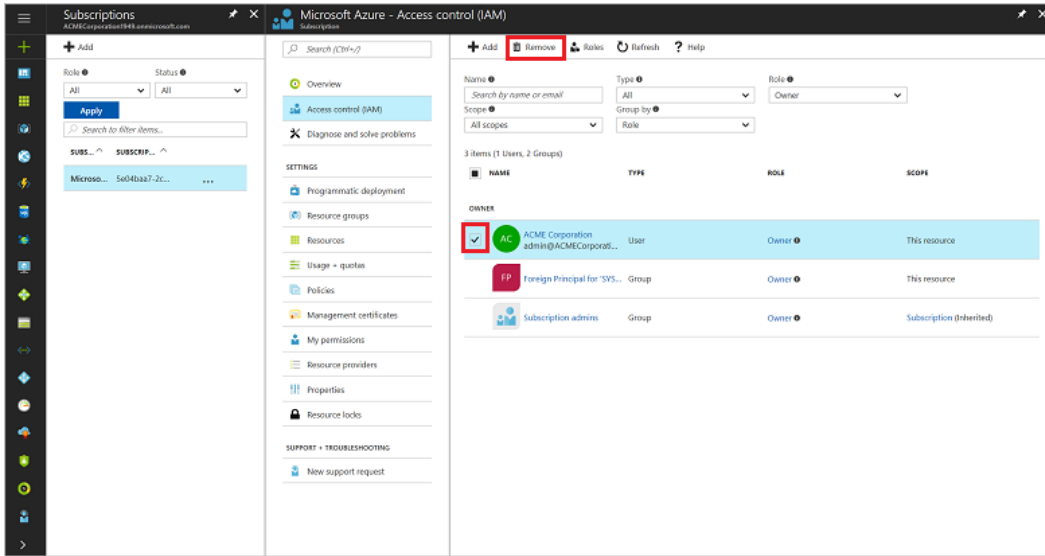


The user's access permissions are displayed.

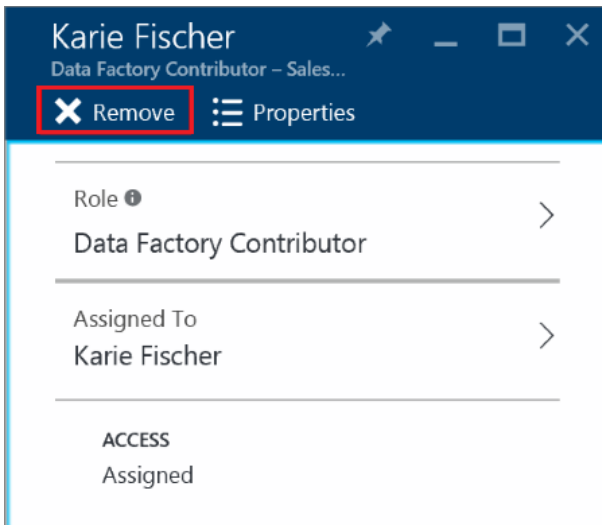
Delete access permissions

You can easily remove a user's subscription access in either of two ways:

- In the **Access control** window:
 1. To get to the **Access control (IAM)** window, follow steps 1 and 2 in the [Add access permissions](#) section.
 2. Select the check box next to the user's name, and then select **Remove**.



- On the user's page:
 1. To view the list of a user's access permissions, follow the steps in the [View access permissions](#) section.
 2. Select the subscription that you want to remove, and then select **Remove**.



PowerShell

Grant access permissions

To grant a user access to a subscription by using PowerShell, run the following commands:

```
# Get subscription ID (if necessary)
C:\PS>Get-AzureSubscription -SubscriptionName "MyProdSubscription"

# Assign role to subscription
New-AzureRmRoleAssignment -ObjectId <application id> -RoleDefinitionName <role name> -Scope <subscription id>
```

View access permissions

To view the access assignments for a specific subscription, run the following commands:

```
Get-AzureRmRoleAssignment -SubscriptionName "MyProdSubscription" | FL DisplayName, RoleDefinitionName, Scope
```

Remove access permissions

To remove a user's subscription access, run the following commands:

```
Remove-AzureRmRoleAssignment -ObjectId <object id> -RoleDefinitionName <role name> -Scope <scope such as subscription id>
```

REST API

Grant access permissions

To create a role assignment, you must have access to the `Microsoft.Authorization/roleAssignments/write` operation. Of the built-in roles, only *Owner* and *User Access Administrator* are granted access to this operation.

Request syntax

METHOD	REQUEST URI
PUT	<code>https://management.azure.com/subscriptions/{subscription-id}/providers/Microsoft.Authorization/roleAssignments/{role-assignment-id}?api-version={api-version}</code>

Request parameters

These are the required parameters for the request:

NAME	TYPE	DESCRIPTION
subscription-id	string	The ID number for the subscription.
role-assignment-id	string	The ID number for the new role assignment.
api-version	string	The current API version that you are using.

Request body

The request body should look like this:

```
{
  "properties": {
    "roleDefinitionId": "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/resourceGroups/Network/providers/Microsoft.Network/virtualNetworks/EASTUS-VNET-01/subnets/Devices-Engineering-ProjectRND/providers/Microsoft.Authorization/roleDefinitions/9980e02c-c2be-4d73-94e8-173b1dc7cf3c",
    "principalId": "5ac84765-1c8c-4994-94b2-629461bd191b"
  }
}
```

NAME	TYPE	DESCRIPTION
roleDefinitionId	String	The identifier of the role. The format is <code>{scope}/providers/Microsoft.Authorization/roleDefinition-id-guid</code> .
principalId	String	The objectId of the Azure Active Directory (Azure AD) principal (user, group, or service to which the role is assigned).

Response example

```
{
  "properties": {
    "roleDefinitionId": "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/providers/Microsoft.Authorization/roleDefinitions/9980e02c-c2be-4d73-94e8-173b1dc7cf3c",
    "principalId": "5ac84765-1c8c-4994-94b2-629461bd191b",
    "scope": "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/resourceGroups/Network/providers/Microsoft.Network/virtualNetworks/EASTUS-VNET-01/subnets/Devices-Engineering-ProjectRND",
    "createdOn": "2015-12-16T00:27:19.6447515Z",
    "updatedOn": "2015-12-16T00:27:19.6447515Z",
    "createdBy": null,
    "updatedBy": "877f0ab8-9c5f-420b-bf88-a1c6c7e2643e"
  },
  "id": "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/resourceGroups/Network/providers/Microsoft.Network/virtualNetworks/EASTUS-VNET-01/subnets/Devices-Engineering-ProjectRND/providers/Microsoft.Authorization/roleAssignments/2e9e86c8-0e91-4958-b21f-20f51f27bab2",
  "type": "Microsoft.Authorization/roleAssignments",
  "name": "2e9e86c8-0e91-4958-b21f-20f51f27bab2"
}
```

View access permissions

To list roles, you must have access to the `Microsoft.Authorization/roleDefinitions/read` operation at the scope. All the built-in roles are granted access to this operation.

Request syntax

METHOD	REQUEST URI
GET	https://management.azure.com/subscriptions/{subscription-id}/providers/Microsoft.Authorization/roleAssignments/{role-assignment-id}?api-version={api-version}&{\$filter={filter}}

Request parameters

These are the required parameters for the request:

NAME	TYPE	DESCRIPTION
subscription-id	string	The ID number for the subscription.
role-assignment-id	string	The ID number for the new role assignment.
api-version	string	The current API version that you are using.
filter	string	The specific conditions, such as viewing permissions for any child resources in that subscription with <code>atScopeAndBelow()</code> , or searching for a role for a specific user or group, such as <code>principalId%20eq%20'{objectId of user, group, or service principal}'</code> .

Response example

```
{
  "value": [
    {
      "properties": {
        "roleDefinitionId": "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/providers/Microsoft.Authorization/roleDefinitions/acdd72a7-3385-48ef-bd42-f606fba81ae7",
        "principalId": "2f9d4375-cbf1-48e8-83c9-2a0bc4cb33fb",
        "scope": "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e",
        "createdOn": "2015-10-08T07:28:24.3905077Z",
        "updatedOn": "2015-10-08T07:28:24.3905077Z",
        "createdBy": "877f0ab8-9c5f-420b-bf88-a1c6c7e2643e",
        "updatedBy": "877f0ab8-9c5f-420b-bf88-a1c6c7e2643e"
      },
      "id": "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/providers/Microsoft.Authorization/roleAssignments/baa6e199-ad19-4667-b768-623fde31aedd",
      "type": "Microsoft.Authorization/roleAssignments",
      "name": "baa6e199-ad19-4667-b768-623fde31aedd"
    }
  ],
  "nextLink": null
}
```

Remove access permissions

To delete a role assignment, you must have access to the `Microsoft.Authorization/roleAssignments/delete` operation. Of the built-in roles, only *Owner* and *User Access Administrator* are granted access to this operation.

Request syntax

METHOD	REQUEST URI
DELETE	<code>https://management.azure.com/subscriptions/{subscription-id}/providers/Microsoft.Authorization/roleAssignments/{role-assignment-id}?api-version={api-version}</code>

Request parameters

These are the required parameters for the request:

NAME	TYPE	DESCRIPTION
subscription-id	string	The ID number for the subscription.
role-assignment-id	string	The ID number for the new role assignment.
api-version	string	The current API version that you are using.

Response example

<pre>{ "properties": { "roleDefinitionId": "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/providers/Microsoft.Authorization/roleDefinitions/9980e02c-c2be-4d73-94e8-173b1dc7cf3c", "principalId": "5ac84765-1c8c-4994-94b2-629461bd191b", "scope": "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/resourceGroups/Network/providers/Microsoft.Network/virtualNetworks/EASTUS-VNET-01/subnets/Devices-Engineering-ProjectRND", "createdOn": "2015-12-17T23:21:40.8921564Z", "updatedOn": "2015-12-17T23:21:40.8921564Z", "createdBy": "877f0ab8-9c5f-420b-bf88-a1c6c7e2643e", "updatedBy": "877f0ab8-9c5f-420b-bf88-a1c6c7e2643e" }, "id": "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/resourceGroups/Network/providers/Microsoft.Network/virtualNetworks/EASTUS-VNET-01/subnets/Devices-Engineering-ProjectRND/providers/Microsoft.Authorization/roleAssignments/5eec22ee-ea5c-431e-8f41-82c560706fd2", "type": "Microsoft.Authorization/roleAssignments", "name": "5eec22ee-ea5c-431e-8f41-82c560706fd2" }</pre>

Next steps

- [Add an existing Azure customer to your Partner Center account](#)

Manage user accounts

9/19/2017 • 9 min to read • [Edit Online](#)

This guide covers the various actions you take to manage a customer's user accounts in Partner Center, including *adding, deleting, updating, and restoring* user accounts.

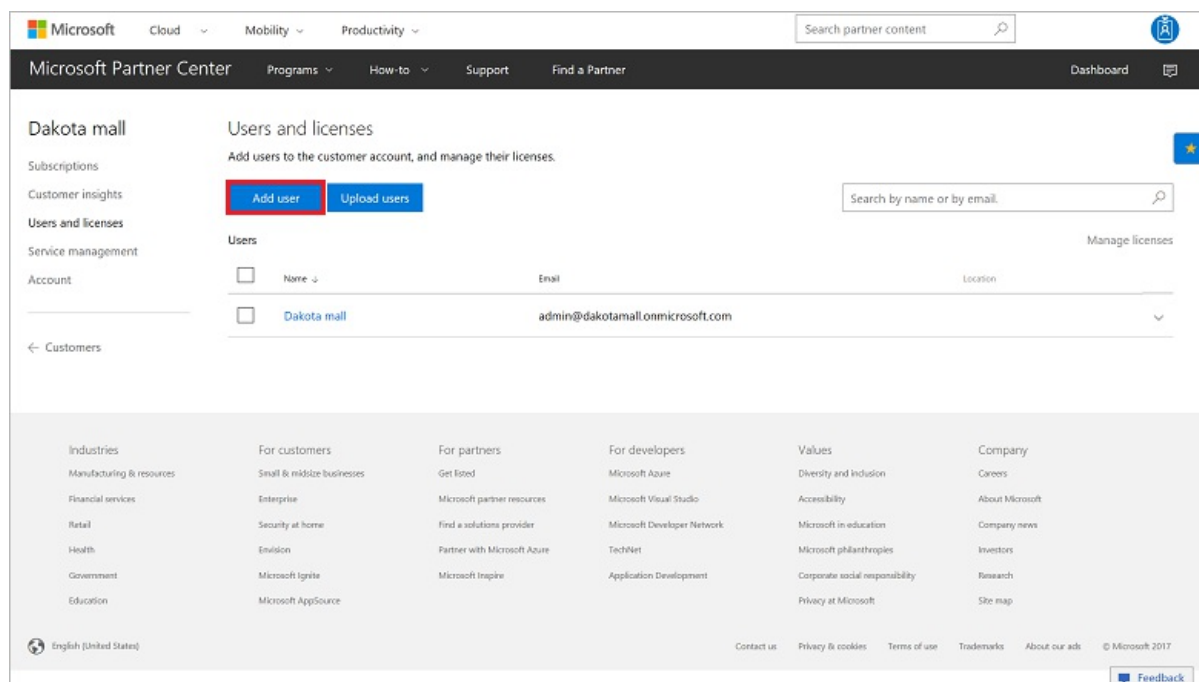
Web UI method

Create users by using the web UI method

With the Partner Center Web portal, you can upload users either manually (one at a time) or in bulk with a .csv file. Instructions for both methods are described here.

Create users one at a time by using the web UI method

1. From the Partner Center **Dashboard**, select **Customers**. Next, select the specific customer, and then select **Users and Licenses**.
2. Select the **Add User** button.



3. Enter the user's first and last name (optional), display name as it will be shown in Partner Center, on microsoft.com email address, and location.
4. Select the user permissions.
5. When you're finished, select **Save**.

Subscriptions
Customer insights
Users and licenses
Add user
Service management
Account

← Customers

Add new user accounts to your directory, set their permissions, and assign licenses.

User info

First name
John

Last name
Doe

Display name *
JohnDoe

Email *
jd1990 @dakotamall.onmicrosoft.com

Location * [What is this?](#)
United States

Permissions

☐ User (can use licenses and services)
☒ Global admin (has full permissions)
☐ Custom permissions

Assign licenses

Product	Available licenses
<input checked="" type="checkbox"/> Azure Active Directory Basic	0

Save Cancel

Feedback

6. Make sure to copy the temporary password information for the new user.

IMPORTANT

This temporary password information won't be available later.

Create users in bulk by using the web UI method

Before you can upload users in bulk, the following criteria must be met:

- You must have global administrator permissions for the account.
- Each user must have a unique email address in the customer's email domain(s).
- A maximum of 100 users can be uploaded at a time. Batches of more than 100 users must be split across multiple uploads.
- All users must be in the same location.
- Only the necessary data should be entered. *Extra data causes the upload to fail.*

1. Create a comma-separated value (.csv) data file with the data that's described in the following section. Then save the file.
2. From the **Dashboard** menu, select **Customers**. Then choose a customer from the list.
3. Go to **Users and licenses**, and then select **Upload users**.
4. Under **Upload user info**, select **Browse**, and then choose the .csv file from step 1.
5. Select **Validate**.

NOTE

Most account creation errors are caused by data file issues, including missing information, malformed or duplicate email addresses, or too many records in the file. Ensure that the data is accurate before uploading it.

6. After the data has been validated, select the geographic **Location** for the new users, and then select **Save**.
7. Download the temporary password information for the users.

IMPORTANT

This temporary password information won't be available later and must be downloaded immediately.

CSV data file parameters

The following data must be present in the data file.

COLUMN NAME	DESCRIPTION	LIMITATION
First name	User's first name (optional)	50-character limit
Last name	User's last name (optional)	50-character limit
Display name	Name that's displayed in Partner Center	50-character limit
Email	User's business email address at customer company	Each user must have a unique email address
Status update	An update that indicates whether the new user record was successfully created	<i>Must be left empty</i>

Update a user account by using the web UI method

1. On the **Dashboard** screen, go to **Customers**. Select the customer's name, and then select **Users and Licenses**. Then select the user's name.

The screenshot shows the Microsoft Partner Center interface. The top navigation bar includes the Microsoft logo, links for Cloud, Mobility, and Productivity, a search bar for partner content, and a user profile icon. Below this is a secondary navigation bar with links for Programs, How-to, Support, and Find a Partner. The main content area is titled 'Dakota mall' and 'Users and licenses'. On the left, a sidebar menu lists various options, with 'Users and licenses' highlighted in a red box. The main area shows a table of users with columns for Name, Email, and Location. Two users are listed: 'Dakota mall' (admin@dakotamall.onmicrosoft.com) and 'JohnDoe' (jd1990@dakotamall.onmicrosoft.com). The 'JohnDoe' user's name is highlighted in a red box. At the bottom of the page, there is a footer with links for Industries, For customers, For partners, For developers, Values, and Company.

2. Here the user's information can be updated, including name, display name, email, location, and permissions. After you make the changes, select **Submit**.

Users and licenses

Update user

Service management

Account

← Customers

John Doe

Display name * JohnDoe

Email * jd1991 @dakotamall.onmicrosoft.com

Location * What is this? United States

Permissions

☐ User (can use licenses and services)

☐ Global admin (has full permissions)

☒ Custom permissions

☐ Password admin (can reset user passwords)

☒ User management admin (can manage user accounts)

☐ Service admin (can manage services)

Assign licenses

Product	Available licenses
<input checked="" type="checkbox"/> Azure Active Directory Basic	0

Submit Cancel Reset password Delete user Feedback

Delete a user account by using the web UI method

1. From the **Dashboard** menu, select **Customers**. Choose the customer from the list.
2. In the customer menu, select **Users and licenses**. Choose the user from the list.

Microsoft Cloud Mobility Productivity Search partner content

Microsoft Partner Center Programs How-to Support Find a Partner Dashboard

Dakota mall

Subscriptions

Customer insights

Users and licenses

Service management

Account

← Customers

Users and licenses

Add users to the customer account, and manage their licenses.

Add user Upload users

Search by name or by email.

Users

Name	Email	Location
Dakota mall	admin@dakotamall.onmicrosoft.com	
JohnDoe	jd1990@dakotamall.onmicrosoft.com	United States

Manage licenses

Industries

Manufacturing & resources

Financial services

Retail

Health

Government

Education

For customers

Small & midsize businesses

Enterprise

Security at home

Microsoft Envision

Microsoft Ignite

Microsoft AppSource

For partners

Get listed

Microsoft partner resources

Find a solutions provider

Partner with Microsoft Azure

Microsoft Inspire

For developers

Microsoft Azure

Microsoft Visual Studio

Microsoft Developer Network

TechNet

Application Development

Values

Diversity and inclusion

Accessibility

Microsoft in education

Microsoft philanthropies

Corporate social responsibility

Privacy at Microsoft

Company

Careers

About Microsoft

Company news

Investors

Research

Site map

3. At the bottom of the screen, select **Delete user account**.

Dakota mall

Subscriptions
Customer insights
Users and licenses
Update user
Service management
Account

JohnDoe

User info

First name

John

Last name

Doe

Display name *

JohnDoe

Email *

jd1990 @dakotamall.onmicrosoft.com

Location * [What is this?](#)

United States

Permissions

☒ User (can use licenses and services)
☐ Global admin (has full permissions)
☐ Custom permissions

Assign licenses

Product	Available licenses
<input checked="" type="checkbox"/> Azure Active Directory Basic	0

Submit

Cancel

Reset password

Delete user

Feedback

Restore a deleted user by using the web UI method

If a user was deleted in the last 30 days, you can restore them by taking the following steps:

1. From the **Dashboard** menu, select **Customers**. Then choose the customer from the list.
2. Select **Users and licenses**.
3. Select the **Deleted users ()** tab. (It reads **(1)** or greater when the tab is active.)

Microsoft
Cloud
Mobility
Productivity

Search partner content

Microsoft Partner Center
Programs
How-to
Support
Find a Partner
Dashboard

Dakota mall

Subscriptions
Customer insights
Users and licenses
Service management
Account

Users and licenses
Add users to the customer account, and manage their licenses.

Add user
Upload users

Search by name or by email.

Users
Deleted Users(1)
Manage licenses

	Name	Email	Location
<input type="checkbox"/>	Dakota mall	admin@dakotamall.onmicrosoft.com	

Industries
Manufacturing & resources
Financial services
Retail
Health
Government
Education

For customers
Small & midsize businesses
Enterprise
Security at home
Microsoft Envision
Microsoft Ignite
Microsoft AppSource

For partners
Get listed
Microsoft partner resources
Find a solutions provider
Partner with Microsoft Azure
Microsoft Inspire

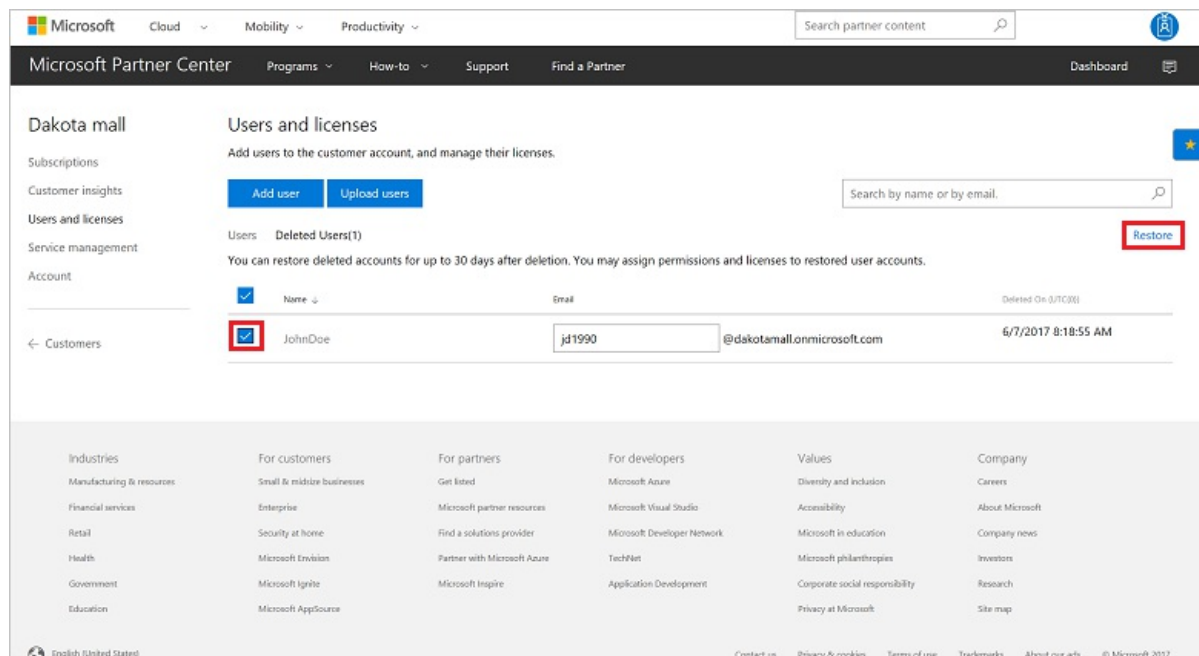
For developers
Microsoft Azure
Microsoft Visual Studio
Microsoft Developer Network
TechNet
Application Development

Values
Diversity and inclusion
Accessibility
Microsoft in education
Microsoft philanthropies
Corporate social responsibility
Privacy at Microsoft

Company
Careers
About Microsoft
Company news
Investors
Research
Site map

English (United States)
Contact us
Privacy & cookies
Terms of use
Trademarks
About our ads
© Microsoft 2017

4. Select one or more of the deleted user's check boxes, and then select **Restore**.



All selected user accounts reappear in the **Users and licenses** page.

PowerShell method

Create a user by using PowerShell

To create a new user account by using PowerShell, use the following commands:

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'

$password = '<password>'

$passwordSecure = $password | ConvertTo-SecureString -AsPlainText -Force

New-PCCustomerUser -tenantid $customer.id -usageLocation '<country code>' -userPrincipalName '<upn>' -firstName '<first name>' -lastName '<last name>' -displayName '<display name>' -forceChangePassword $true -password $passwordSecure
```

Update a user by using PowerShell

To update a customer user by using PowerShell, input the following commands:

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'

$user = Get-PCCustomerUser -tenantid $customer.id -userid '<user id>'

Set-PCCustomerUser -tenantid $customer.id -user $user -userPrincipalName '<new UPN>'
```

Delete a user by using PowerShell

To delete a user account by using PowerShell, input the following commands:

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'

$user = Get-PCCustomerUser -tenantid $customer.id -userid '<user id>'

Remove-PCCustomerUser -tenantid $customer.id -user $user
```

Restore a deleted user by using PowerShell

To restore a deleted user by using PowerShell, input the following commands:

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'

$user = Get-PCCustomerUser -tenantid $customer.id -deleted | ? id -EQ '<user id>'

Restore-PCCustomerUser -tenantid $customer.id -user $user
```

C# method

Create a user by using C#

To create a new user by using C#, take the following steps:

1. Create a new **CustomerUser** object with the relevant user information.
2. Use the **IAggregatePartner.Customers** collection and call the **ById()** method.
3. Call the **Users** property, followed by the **Create** method.

```
// string selectedCustomerId;
// IAggregatePartner partnerOperations;
// var SelectedCustomer;

var userToCreate = new CustomerUser()
{
    PasswordProfile = new PasswordProfile() { ForceChangePassword = true, Password = "Password!1" },
    DisplayName = "TestDisplayName",
    FirstName = "TestFirstName",
    LastName = "TestLastName",
    UsageLocation = "US",
    UserPrincipalName = Guid.NewGuid().ToString("N") + "@" + selectedCustomer.CompanyProfile.Domain.ToString()
};

User createdUser = partnerOperations.Customers.ById(selectedCustomerId).Users.Create(userToCreate);
```

Update a user by using C#

To update the details for a specified customer user by using C#, take the following steps:

1. Retrieve the specified customer ID and user to update.
2. Create an updated version of the user in a new **CustomerUser** object.
3. Use the **IAggregatePartner.Customers** collection and call the **ById()** method.
4. Call the **Users** property and the **ById()** method, followed by the **Patch()** method.

```
//string selectedCustomerId;
//customerUser specifiedUser;
//IAggregatePartner partnerOperations;

//updated information
var userToUpdate = new CustomerUser()
{
    PasswordProfile = new PasswordProfile() { ForceChangePassword = true, Password = "testPw@!122B" },
    DisplayName = "Roger Federer",
    FirstName = "Roger",
    LastName = "Federer",
    UsageLocation = "US",
    UserPrincipalName = Guid.NewGuid().ToString("N") + "@" + selectedCustomer.CompanyProfile.Domain.ToString()
};

// update customer user information
User updatedCustomerUserInfo = partnerOperations.Customers.ById(selectedCustomerId).Users.ById(specifiedUser.Id).Patch(userToUpdate);
```

Delete a user by using C#

To delete a user by using C#, take the following steps:

1. Use the **IAggregatePartner.Customers.ById** method with the customer ID to identify the customer.
2. Call the **Users.ById** method to identify the user.
3. Call the **Delete** method to delete the user and set the user state to inactive.

```
// string selectedCustomerId;
// IAggregatePartner partnerOperations;
// string customerUserIdToDelete;

partnerOperations.Customers.ById(selectedCustomerId).Users.ById(customerUserIdToDelete).Delete();
```

Restore a deleted user by using C#

To restore a deleted user by using C#, take the following steps:

1. Create a new instance of the **CustomerUser** class, and then set the **User.State** to **UserState.Active**.
Remaining fields in the user resource are automatically restored from the deleted, inactive user resource.
2. Use the **IAggregatePartner.Customers.ById** method with the customer ID to identify the customer, and the **Users.ById** method to identify the user.
3. Call the **Patch** method and pass the **CustomerUser** instance to send the request to restore the user.

```
// string selectedCustomerId;
// IAggregatePartner partnerOperations;
// string selectedCustomerUserId;

var updatedCustomerUser = new CustomerUser()
{
    State = UserState.Active
};

// Restore customer user information.
var restoredCustomerUserInfo =
partnerOperations.Customers.ById(selectedCustomerId).Users.ById(selectedCustomerUserId).Patch(updatedCustomerUser);
```

REST API method

Create a user by using the REST API method

Request Syntax

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	Allows the reseller to filter the results for a customer that belongs to the reseller.
user-id	guid	Belongs to a single user account. (<i>Optional</i>)

Request example

```
POST https://api.partnercenter.microsoft.com/v1/customers/<customer-tenant-id>/users HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: b1317092-f087-471e-a637-f66523b2b94c
MS-CorrelationId: 8a53b025-d5be-4d98-ab20-229d1813de76
{
  "usageLocation": "country/region code",
  "userPrincipalName": "userid@domain.onmicrosoft.com",
  "firstName": "First",
  "lastName": "Last",
  "displayName": "User name",
  "passwordProfile": {
    password: "abCD123*",
    forceChangePassword: true
  },
  "attributes": {
    "objectType": "CustomerUser"
  }
}
```

Response

If this method is successful, it returns a user account, including the GUID.

Response example

```
HTTP/1.1 200 OK
Content-Length: 31942
Content-Type: application/json
MS-CorrelationId: 8a53b025-d5be-4d98-ab20-229d1813de76
MS-RequestId: b1317092-f087-471e-a637-f66523b2b94c
Date: June 24 2016 22:00:25 PST
```

```
{
  "usageLocation": "country/region code",
  "id": "4b10bf41-ab11-40e3-8c53-cd67849b50de",
  "userPrincipalName": "userid@domain.onmicrosoft.com",
  "firstName": "First",
  "lastName": "Last",
  "displayName": "User name",
  "passwordProfile": {
    "forceChangePassword": true,
    "password": "abCD123*"
  },
  "lastDirectorySyncTime": null,
  "userDomainType": "none",
  "state": "active",
  "softDeletionTime": null,
  "attributes": {
    "objectType": "CustomerUser"
  }
}
```

Update a user by using the REST API method

Request syntax

METHOD	REQUEST URI
PATCH	<i>{baseUrl}</i> /v1/customers/{customer-tenant-id}/users HTTP/1.1

URI parameters

These are the *required* query parameters for identifying the correct customer:

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	Identifies the customer.
user-id	guid	Belongs to a single user account.

Request example

```
PATCH https://api.partnercenter.microsoft.com/v1/customers/<customer-tenant-id>/users/<user-id> HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: b1317092-f087-471e-a637-f66523b2b94c
MS-CorrelationId: 8a53b025-d5be-4d98-ab20-229d1813de76
{
  "usageLocation": "new country/region code",

  "attributes": {
    "objectType": "CustomerUser"
  }
}
```

Response

If this method is successful, it returns a user account with the updated information.

Response example

```
HTTP/1.1 200 OK
Content-Length: 31942
Content-Type: application/json
MS-CorrelationId: 8a53b025-d5be-4d98-ab20-229d1813de76
MS-RequestId: b1317092-f087-471e-a637-f66523b2b94c
Date: June 24 2016 22:00:25 PST
{
  "usageLocation": "new country/region code",
  "id": "4b10bf41-ab11-40e3-8c53-cd67849b50de",
  "userPrincipalName": "emailidchange@abcdefgh1234.ccsctp.net",
  "firstName": "FirstNameChange",
  "lastName": "LastNameChange",
  "displayName": "DisplayNameChange",
  "userDomainType": "none",
  "state": "active",
  "links": {
    "self": {
      "uri": "/customers/eebd1b55-5360-4438-a11d-5c06918c3014/users/4b10bf41-ab11-40e3-8c53-cd67849b50de",
      "method": "GET",
      "headers": [

    ]
    }
  },
  "attributes": {
    "objectType": "CustomerUser"
  }
}
```

Delete a user by using the REST API method

Request syntax

METHOD	REQUEST URI
DELETE	<i>{baseUrl}</i> /v1/customers/{customer-tenant-id}/users/{user-id} HTTP/1.1

URI parameters

Use the following parameters to identify the customer and user:

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	Allows the reseller to filter the results for a given customer.
user-id	guid	Belongs to a single user account.

Request example


```
DELETE https://api.partnercenter.microsoft.com/v1/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/users/a45f1416-3300-4f65-9e8d-f123b397a4ea
HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: f113b126-ec13-4baa-ab4d-67c245244971
MS-CorrelationId: 709c0b80-016c-4662-b29f-697fdff03e87a
X-Locale: en-US
Host: api.partnercenter.microsoft.com
Content-Length: 0
```

Response

If the method is successful, it returns **204 No Content**.

Response example

```
HTTP/1.1 204 No Content
Content-Length: 0
MS-CorrelationId: 709c0b80-016c-4662-b29f-697fdff03e87a
MS-RequestId: f113b126-ec13-4baa-ab4d-67c245244971
MS-CV: 90KUJA7HKEaG8wHu.0
MS-ServerId: 101112616
Date: Tue, 24 Jan 2017 23:27:18 GMT
```

Restore a deleted user by using the REST API method

Request syntax

METHOD	REQUEST URI
PATCH	<i>{baseUrl}</i> /v1/customers/{customer-tenant-id}/users/{user-id} HTTP/1.1

URI parameters

Following are the query parameters that are required to specify the customer ID and user ID:

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	Allows the reseller to filter the results to a given customer.
user-id	guid	Belongs to a single user account.

Request body

These are the properties in the request body.

NAME	TYPE	DESCRIPTION
State	string	The user state. To restore a deleted user, this must contain "active"
Attributes	object	Contains "ObjectType":"CustomerUser". (Optional)

Request example

```
PATCH https://api.partnercenter.microsoft.com/v1/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/users/a45f1416-3300-4f65-9e8d-f123b397a4ea
HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: 6e668bc0-5bd7-44d6-b6fa-529d41ce9659
MS-CorrelationId: 32be760f-8282-4e01-a37b-829c8a700e8a
X-Locale: en-US
Content-Type: application/json
Host: api.partnercenter.microsoft.com
Content-Length: 269
Expect: 100-continue

{
  "State": "active",
  "Attributes": {
    "ObjectType": "CustomerUser"
  }
}
```

Response

If this method is successful, it returns the restored user information in the response body.

Response example

```
HTTP/1.1 200 OK
Content-Length: 465
Content-Type: application/json; charset=utf-8
MS-CorrelationId: 32be760f-8282-4e01-a37b-829c8a700e8a
MS-RequestId: 6e668bc0-5bd7-44d6-b6fa-529d41ce9659
MS-CV: ZTeBriO7mEaiM13+.0
MS-ServerId: 101112616
Date: Fri, 20 Jan 2017 22:24:55 GMT

{
  "usageLocation": "US",
  "id": "a45f1416-3300-4f65-9e8d-f123b397a4ea",
  "userPrincipalName": "e83763f7f2204ac384cfd49f79f2749@dttdemocspcustomer005.onmicrosoft.com",
  "firstName": "Ferdinand",
  "lastName": "Filibuster",
  "displayName": "Ferdinand",
  "userDomainType": "none",
  "state": "active",
  "links": {
    "self": {
      "uri": "/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/users/a45f1416-3300-4f65-9e8d-f123b397a4ea",
      "method": "GET",
      "headers": []
    }
  },
  "attributes": {
    "objectType": "CustomerUser"
  }
}
```

Get company profile information

9/19/2017 • 4 min to read • [Edit Online](#)

This guide covers how to retrieve and update billing or business profile information for a customer on your Azure Cloud Solution Provider (Azure CSP) account.

Web UI

1. On the Partner Center **Dashboard** page, select **Customers** > *customer name* > **Account**.

- The customer's company information is displayed under **Company Info**.
- The customer's billing information is displayed under **Bill-to info**.

2. To update the billing information, select **Update**.

The screenshot shows the Microsoft Partner Center interface. The top navigation bar includes the Microsoft logo, tabs for Cloud, Mobility, and Productivity, a search bar, and a user profile icon. Below this is a secondary navigation bar with links to Programs, How-to, Support, and Find a Partner. The main content area is titled 'ACME Corporation' and 'Account'. On the left, a sidebar lists various management options, with 'Account' highlighted. The main content is divided into two sections: 'Company info' and 'Bill-to info'. The 'Company info' section displays the company name 'ACME Corporation', address '1 Tower Lane, Los Angeles CA 90001', and primary contact 'mblanc@outlook.com'. The 'Bill-to info' section displays the same information and includes an 'Update' button, which is highlighted with a red box. At the bottom, there is a footer with links to various resources like Industries, For customers, For partners, For developers, Values, and Company.

3. After you've made your changes, select **Submit**.

The screenshot shows the 'Bill-to info' update form in the Microsoft Partner Center. The top navigation bar is the same as the previous screenshot. The main content area is titled 'ACME Corporation' and 'Bill-to info'. On the left, a sidebar lists various management options, with 'Account' highlighted. The main content is a form with fields for 'Country/region', 'Address line 1', 'City', 'First name', 'Email address', 'Company name', 'Address line 2', 'State/Province', 'ZIP/Postal code', 'Last name', and 'Phone number'. The 'Submit' button is highlighted with a red box. At the bottom, there is a footer with links to various resources like Industries, For customers, For partners, For developers, Values, and Company.

PowerShell

Get a company profile

To retrieve a customer's company profile, enter the following commands:

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'
Get-PCCustomerCompanyProfile -tenantid $customer.id
```

Get a billing profile

To retrieve a customer's billing profile, enter the following commands:

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'
Get-PCCustomerBillingProfile -tenantid $customer.id
```

Update a billing profile

To update a customer's billing profile, enter the following commands:

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'

$customerBillingProfile = Get-PCCustomerBillingProfile -tenantid $customer.id

$customerBillingProfile.firstName = '<first name>'
$customerBillingProfile.lastName = '<last name>'
$customerBillingProfile.email = '<email>'
Set-PCCustomerBillingProfile -tenantid $customer.id -billingprofile $customerBillingProfile
```

C#

View a company profile

1. To get the company profile for a customer, call the **IAggregatePartner.Customers.ById** method with the customer ID to identify the customer.
2. To access the customer's Company property, get the customer's **ICustomerProfileCollection** interface from the **Profiles** property.
3. Get the **ICustomerReadOnlyProfile** interface from the **ICustomerProfileCollection.Company** property, and then call its **Get()** or **GetAsync()** method.

```
// IAggregatePartner partnerOperations;
// var selectedCustomerId as string;

var companyProfile = partnerOperations.Customers.ById(selectedCustomerId).Profiles.Company.Get();
```

View a billing profile

1. To get a customer's billing profile, use your **IPartner.Customers** collection, and then call the **ById()** method.
2. Call the **Profiles** property, followed by the **Billing** property.
3. Call the **Get()** or **GetAsync()** method.

```
// IAggregatePartner partnerOperations;  
// var selectedCustomerId as string;  
  
var billingProfile = partnerOperations.Customers.ById(selectedCustomerId).Profiles.Billing.Get();
```

Update a billing profile

To update a customer's billing profile, do the following:

1. Retrieve the billing profile, and then update the properties as necessary.
2. Retrieve your **IPartner.Customers** collection, and then call the **ById()** method.
3. Call the **Profiles** property, followed by the **Billing** property.
4. Call the **Update()** or **UpdateAsync()** method.

```
// IAggregatePartner partnerOperations;  
// var selectedCustomerId as string;  
  
var billingProfile = partnerOperations.Customers.ById(selectedCustomerId).Profiles.Billing.Get();  
  
// Apply changes to profile;  
  
billingProfile = partnerOperations.Customers.ById(selectedCustomerId).Profiles.Billing.Update(billingProfile);
```

REST API

View a company profile

Request syntax

METHOD	REQUEST URI
GET	<i>{baseUrl}/v1/customers/{customer-tenant-id}/profiles/company</i> HTTP/1.1

URI parameter

To get the company profile, use the following query parameter:

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	A GUID-formatted string that allows the reseller to filter the results for a specific customer that belongs to the reseller.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/profiles/company HTTP/1.1  
Authorization: Bearer <token>  
Accept: application/json  
MS-RequestId: 0b6f039c-e4b5-4b9e-bdac-b39077bb60da  
MS-CorrelationId: ffa9174c-dbc-b47de-b70d-10e640a7f1b4  
X-Locale: en-US  
Host: api.partnercenter.microsoft.com  
Connection: Keep-Alive
```

If the request is successful, this method returns information in the response body.

Response example

```
HTTP/1.1 200 OK
Content-Length: 488
Content-Type: application/json; charset=utf-8
MS-CorrelationId: ffa9174c-dbc4-47de-b70d-10e640a7f1b4
MS-RequestId: 0b6f039c-e4b5-4b9e-bdac-b39077bb60da
MS-CV: /e74N8OrkE29ycwZ.0
MS-ServerId: 101112202
Date: Wed, 04 Jan 2017 19:48:51 GMT
```

```
{
  "tenantId": "4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04",
  "domain": "dtdemocspcustomer005.onmicrosoft.com",
  "companyName": "DT Demo CSP Customer 005",
  "address": {
    "country": "US",
    "region": "WA",
    "city": "Redmond ",
    "addressLine1": "1 Microsoft Way",
    "postalCode": "98052",
    "phoneNumber": "4155551212"
  },
  "email": "daniel@hotmail.comtw",
  "links": {
    "self": {
      "uri": "/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/profiles/company",
      "method": "GET",
      "headers": []
    }
  },
  "attributes": {
    "objectType": "CustomerCompanyProfile"
  }
}
```

View a billing profile

Request syntax

METHOD	REQUEST URI
GET	<i>{baseUrl}</i> /v1/customers/{customer-tenant-id}/profiles/billing HTTP/1.1

URI parameter

This query parameter is *required* for getting the billing profile.

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	A GUID-formatted string that indicates a customer.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers/<customer-tenant-id>/profiles/billing HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: a5581a74-2778-4e34-9172-18baa4877081
MS-CorrelationId: 51d521b3-62db-4682-b75d-fb8ab09113b2
```

If the request is successful, this method returns a collection of **Profile** resources in the response body.

Response example

```
HTTP/1.1 200 OK
Content-Length: 1206
Content-Type: application/json
MS-CorrelationId: 51d521b3-62db-4682-b75d-fb8ab09113b2
MS-RequestId: a5581a74-2778-4e34-9172-18baa4877081
Date: Mon, 23 Nov 2015 18:13:43 GMT

{
  "id": "<billing-profile-id>",
  "firstName": "FirstName",
  "lastName": "LastName",
  "email": "email@sample.com",
  "culture": "en-US",
  "language": "en",
  "companyName": "CompanyName",
  "defaultAddress": {
    "country": "US",
    "city": "Redmond",
    "state": "WA",
    "addressLine1": "1 Microsoft Way",
    "postalCode": "98052",
    "firstName": "FirstName",
    "lastName": "LastName",
    "phoneNumber": "4255555555"
  },
  "links": {
    "self": {
      "uri": "/v1/customers/<customer-tenant-id>/profiles/billing",
      "method": "GET",
      "headers": []
    }
  },
  "attributes": {
    "etag": "<etag>",
    "objectType": "CustomerBillingProfile"
  }
}
```

Update a billing profile

Request syntax

METHOD	REQUEST URI
PUT	<i>{baseUrl}</i> /v1/customers/{customer-tenant-id}/profiles/billing HTTP/1.1

URI parameter

To update the billing profile, use the following query parameter:

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	The value is a GUID-formatted string that allows the reseller to filter the results for a specific customer that belongs to the reseller.

If-Match: An ETag is required for concurrency detection.

Request example

```
PUT https://api.partnercenter.microsoft.com/v1/customers/<customer-tenant-id>/profiles/billing HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: ff85f13a-eb65-4b8e-9b67-05beb0565410
MS-CorrelationId: ff1b757d-cfaf-463a-b48b-0f96d05e95d7
Content-Type: application/json
Content-Length: 639
Expect: 100-continue
```

```
{
  "Id": "a58ceba5-60ac-48e4-a0bc-2a855811807a",
  "FirstName": "FirstName",
  "LastName": "LastName",
  "Email": "email@sample.com",
  "Culture": "en-US",
  "Language": "en",
  "CompanyName": "CompanyName",
  "DefaultAddress": {
    "Country": "US",
    "Region": null,
    "City": "Redmond",
    "State": "WA",
    "AddressLine1": "One Microsoft Way",
    "AddressLine2": null,
    "PostalCode": "98052",
    "FirstName": "FirstName",
    "LastName": "LastName",
    "PhoneNumber": "4255555555"
  },
  "Links": {
    "Self": {
      "Uri": "/v1/customers/<customer-tenant-id>/profiles/billing",
      "Method": "GET",
      "Headers": []
    }
  },
  "Attributes": {
    "Etag": "<etag>",
    "ObjectType": "CustomerBillingProfile"
  }
}
```

If the request is successful, this method returns updated **Profile** resource properties in the response body. This call requires an ETag for concurrency detection.

Response example

HTTP/1.1 200 OK
Content-Length: 1210
Content-Type: application/json
MS-CorrelationId: ff1b757d-cfaf-463a-b48b-0f96d05e95d7
MS-RequestId: ff85f13a-eb65-4b8e-9b67-05beb0565410
Date: Mon, 23 Nov 2015 18:20:43 GMT

```
{
  "id": "a58ceba5-60ac-48e4-a0bc-2a855811807a",
  "firstName": "FirstName",
  "lastName": "LastName",
  "email": "email@sample.com",
  "culture": "en-US",
  "language": "en",
  "companyName": "companyName",
  "defaultAddress": {
    "country": "US",
    "city": "Redmond",
    "state": "WA",
    "addressLine1": "One Microsoft Way",
    "postalCode": "98052",
    "firstName": "FirstName",
    "lastName": "LastName",
    "phoneNumber": "4255555555"
  },
  "links": {
    "self": {
      "uri": "/v1/customers/<customer-tenant-id>/profiles/billing",
      "method": "GET",
      "headers": []
    }
  },
  "attributes": {
    "etag": "<etag>",
    "objectType": "CustomerBillingProfile"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- View a list of [Azure CSP integration scenarios](#).

Get a customer's orders

9/19/2017 • 1 min to read • [Edit Online](#)

Learn how to retrieve a list of all of a customer's orders. You can use one of the following options:

- PowerShell
- C#
- REST API

PowerShell

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'

Get-PCOrder -tenantid $customer.id
```

C#

1. Use your **IAggregatePartner.Customers** collection and call the **ById()** method.
2. Call the **Orders** property.
3. Call the **Get()** or the **GetAsync()** method.

```
// IAggregatePartner partnerOperations;
// var selectedCustomerId as String;

var orders = partnerOperations.Customers.ById(selectedCustomerId).Orders.Get();
```

REST API

Request

Request syntax

METHOD	REQUEST URI
GET	{baseUrl}/v1/customers/{customer-tenant-id}/orders HTTP/1.1

URI parameter

Use the following query parameter to get all orders.

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	A GUID-format customer tenant ID. The reseller can use the customer tenant ID to filter results for a specific customer that is associated with the reseller.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers/<customer-tenant-id>/orders HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: b1317092-f087-471e-a637-f66523b2b94c
MS-CorrelationId: 8a53b025-d5be-4d98-ab20-229d1813de76
```

Response

If the request is successful, this method returns a collection of **Order** resources in the response body.

Response example

HTTP/1.1 200 OK
Content-Length: 31942
Content-Type: application/json
MS-CorrelationId: 8a53b025-d5be-4d98-ab20-229d1813de76
MS-RequestId: b1317092-f087-471e-a637-f66523b2b94c
Date: Mon, 23 Nov 2015 22:00:25 GMT

```
{
  "totalCount": 24,
  "items": [{
    "id": "d6595733-265f-4918-a62e-026e64bc8384",
    "referenceCustomerId": "<customer-tenant-id>",
    "lineItems": [{
      "lineItemNumber": 0,
      "offerId": "031C9E47-4802-4248-838E-778FB1D2CC05",
      "friendlyName": "nickname",
      "quantity": 1,
      "links": {
        "subscription": {
          "uri": "/subscriptions?key=<key>",
          "method": "GET",
          "headers": []
        }
      }
    }
  ]},
  "status": "none",
  "creationDate": "2015-10-08T10: 42: 36.54-07: 00",
  "attributes": {
    "etag": "<etag>",
    "objectType": "Order"
  }
},
{
  "id": "b8fe2ddb-6e85-4f8b-a5ba-06c20ce6d6c5",
  "referenceCustomerId": "<customer-tenant-id>",
  "lineItems": [{
    "lineItemNumber": 0,
    "offerId": "4271AA57-4129-477E-BA88-0A276D4A1619",
    "quantity": 1,
    "links": {
      "subscription": {
        "uri": "/subscriptions?key=<key>",
        "method": "GET",
        "headers": []
      }
    }
  ]},
  "status": "none",
  "creationDate": "2015-10-09T21: 53: 18.83-07: 00",
  "attributes": {
    "etag": "<etag>",
    "objectType": "Order"
  }
}
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Search for customers

9/19/2017 • 6 min to read • [Edit Online](#)

This guide covers the automated API and web UI methods that are available for searching through your list of customers. The methods include isolating specific customers, applying search filters, and retrieving a bulk list of customers.

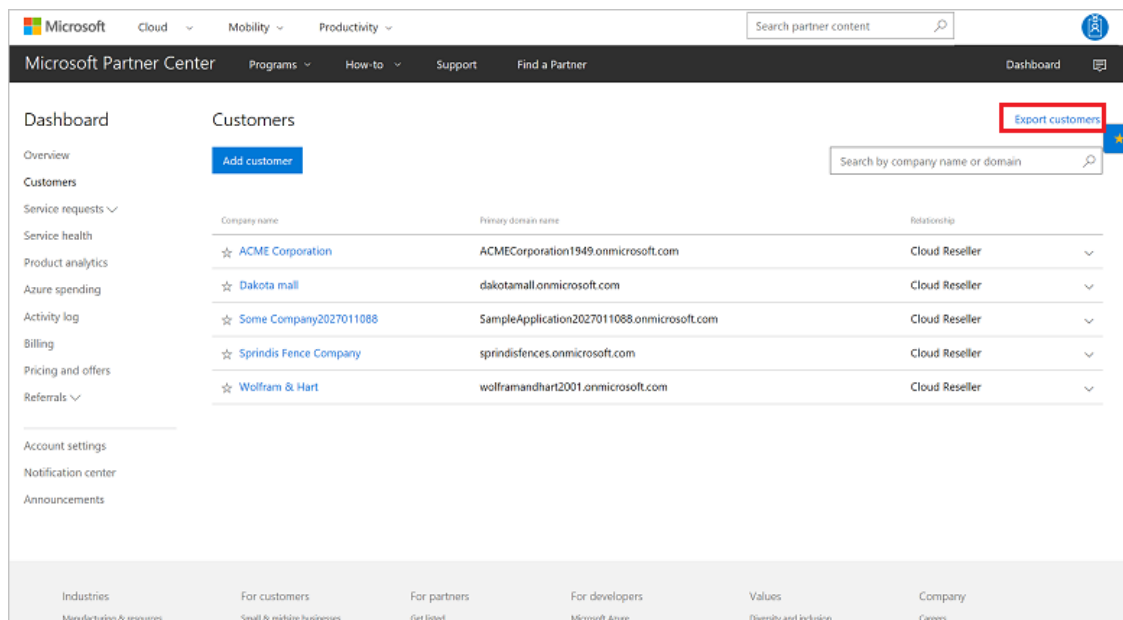
Web UI

Search for a customer by using the search filter

1. On the Microsoft Partner Center **Dashboard** menu, select **Customers**.
2. To search for a customer, enter the customer name or domain name in the search box.
3. To view the customer's Microsoft ID, associated subscriptions, and services quick links, select the down arrow at the right of the customer row.

Get a list of all customers

1. On the **Dashboard** menu, select **Customers**.
2. Do one of the following:
 - To filter for the subset of data that you want to export, use the search filter, and then select **Export customers**.
 - To export the entire customer list, leave the search field blank, and then select **Export customers**.



The screenshot shows the Microsoft Partner Center interface. The top navigation bar includes the Microsoft logo, tabs for Cloud, Mobility, and Productivity, and a search bar. Below this is the 'Microsoft Partner Center' header with links to Programs, How-to, Support, and Find a Partner. The main content area is divided into a left sidebar with navigation links (Dashboard, Overview, Customers, Service requests, Service health, Product analytics, Azure spending, Activity log, Billing, Pricing and offers, Referrals, Account settings, Notification center, Announcements) and a main panel titled 'Customers'. The 'Customers' panel has an 'Add customer' button and a search bar labeled 'Search by company name or domain'. Below the search bar is a table with three columns: 'Company name', 'Primary domain name', and 'Relationship'. The table contains five rows of customer data. In the top right corner of the 'Customers' panel, there is a red box highlighting the 'Export customers' button.

Company name	Primary domain name	Relationship
☆ ACME Corporation	ACMECorporation1949.onmicrosoft.com	Cloud Reseller
☆ Dakota mall	dakotamall.onmicrosoft.com	Cloud Reseller
☆ Some Company2027011088	SampleApplication2027011088.onmicrosoft.com	Cloud Reseller
☆ Sprindis Fence Company	sprindisfences.onmicrosoft.com	Cloud Reseller
☆ Wolfram & Hart	wolframandhart2001.onmicrosoft.com	Cloud Reseller

- **Indirect providers:** You can filter your customer list by indirect reseller. To do so, select **Filter by indirect reseller** from the list, and then select a reseller.

Partner Center converts the complete customer list into a .csv file and then uploads it to the default download folder on your computer. The data columns include the following:

- **Microsoft ID**
- **Company name**
- **Primary domain name**
- **Relationship:** The partner's business relationship with each listed customer

PowerShell

Get a customer by ID

To find a customer by ID, enter the following command:

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'
```

Get a list of all customers

To get a list of all customers, enter the following command:

```
Get-PCCustomer
```

C#

Get a customer by ID

To get a customer by ID, use your **IAggregatePartner.Customers** collection, call the **ById()** method, then call the **Get()** or **GetAsync()** method.

```
// IAggregatePartner partnerOperations;  
// string customerIdToRetrieve;  
  
Customer customerInfo = partnerOperations.Customers.ById(customerIdToRetrieve).Get();
```

Get customers by search filter

To get a collection of customers that match a filter, do the following:

1. To create the filter, instantiate a **SimpleFieldFilter** object.
2. To pass a string that contains the **CustomerSearchField** and indicate the type of filter operation, choose the **FiledFilterOperation** enumeration.
3. To instantiate an **iQuery** object to pass to the query, call the **BuildSimpleQuery** method.
Now it's time to execute the filter and get the result.
4. To get an interface to the partner's customer operations, use **IAggregatePartner.Customers**.
5. Call the **Query** or **QueryAsync** method.

```
IAggregatePartner partnerOperations;  
  
// Specify the partial string to filter by (to match Contoso).  
string searchPrefix = "cont"  
  
// Create a simple field filter.  
var fieldFilter = new SimpleFieldFilter(  
    CustomerSearchField.CompanyName.ToString(),  
    FieldFilterOperation.StartsWith,  
    searchPrefix);  
  
// Create an iQuery object to pass to the Query method.  
var myQuery = QueryFactory.Instance.BuildSimpleQuery(fieldFilter);  
  
// Get the collection of matching customers.  
var customers = partnerOperations.Customers.Query(myQuery);
```

Get a list of all customers

To get a list of all customers, do the following:

1. To create an **IPartner** object, use your **IAggregatePartner.Customers** collection.
2. Retrieve the customer list by using the **Query()** or **QueryAsync()** method. For instructions on creating a query,

see the **QueryFactory** class.

```
// IAggregatePartner partnerOperations;

var allCustomers = new List<Customer>();

// All the operations executed on this partner operation instance will share the same correlation Id but will differ in request Id
IPartner scopedPartnerOperations = partnerOperations.With(RequestContextFactory.Instance.Create(Guid.NewGuid()));

// Read customers into chunks of 40s
var customersBatch = scopedPartnerOperations.Customers.Query(QueryFactory.Instance.BuildIndexedQuery(40));
var customersEnumerator = scopedPartnerOperations.Enumerators.Customers.Create(customersBatch);
```

REST API

Get a customer by ID

Request syntax

METHOD	REQUEST URL
GET	<i>{baseUrl}/v1/customers/{customer-tenant-id}</i> HTTP/1.1

URL parameter

To get a specific customer, use the following query parameter:

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	The value is a GUID-formatted string that lets the reseller filter the results to find a specific customer that belongs to the reseller.

Request example

```
GET https://api.partnercenter.microsoft-ppe.com/v1/customers/<customer-tenant-id> HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-CorrelationId: a176c585-b5de-4d65-824c-67a6deb45cd9
MS-RequestId: 74ca1db9-df92-41c6-a362-a16433b0542b
```

If the request is successful, this method returns a customer resource in the response body.

Response example

```
HTTP/1.1 200 OK
Content-Length: 1530
Content-Type: application/json; charset=utf-8
MS-CorrelationId: a176c585-b5de-4d65-824c-67a6deb45cd9
MS-RequestId: 74ca1db9-df92-41c6-a362-a16433b0542b
{
  "id": "eebd1b55-5360-4438-a11d-5c06918c3014",
  "commerceId": "99e6a635-48e7-424d-9059-c9db944e3c54",
  "companyProfile": {
    "tenantId": "eebd1b55-5360-4438-a11d-5c06918c3014",
    "domain": "abcdefgh1234.ccsctp.net",
    "companyName": "1kl as kjk",
    "address": {
      "country": "US",
```

[illegible]


```
"objectType": "Customer"
}
}
```

Get customers by using the search filter

Request syntax

METHOD	REQUEST URI
GET	<i>{baseURI}</i> /v1/customers?size={size}&filter={filter} HTTP/1.1

URL parameters

NAME	TYPE	DESCRIPTION
size	int	<i>Optional.</i> The number of results to be displayed at one time.
filter	filter	The filter to apply to customers. This filter must be an encoded string.

Filter syntax

Compose the filter parameter as a series of comma-separated, key/value pairs. Each key and value must be individually quoted and separated by a colon. The entire filter must be encoded.

An unencoded example looks like this: `?filter{"Field":"CompanyName","Value":"cont","Operator":"starts_with"}`

The following table describes the key/value pairs:

KEY	VALUE
Field	The field to filter. You can find the valid values in CustomerSearchField .
Value	The value to filter by. Ignore the case of the value.
Operator	The operator to apply. You can find the valid operators in FieldFilterOperation .

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers?
size=0&filter=%7B%22Field%22%3A%22CompanyName%22%2C%22Value%22%3A%22Cont%22%2C%22Operator%22%3A%22starts_with%22%7D HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: 5ce66de5-eea9-486f-a11c-c852aa3d1502
MS-CorrelationId: a2a912ee-d595-47e2-97ae-1b0ae1efa13d
X-Locale: en-US
Host: api.partnercenter.microsoft.com
Connection: Keep-Alive
```

If the request is successful, this method returns a collection of matching customer resources in the response body.

Response example

HTTP/1.1 200 OK
Content-Length: 1839
Content-Type: application/json; charset=utf-8
MS-CorrelationId: a2a912ee-d595-47e2-97ae-1b0ae1efa13d
MS-RequestId: dfeda56c-1af5-43fc-a9c0-346b9e85dc96
MS-CV: n0lMNyJtaUC802pO.0
MS-ServerId: 202010223
Date: Fri, 24 Feb 2017 22:08:20 GMT

```
{
  "totalCount": 3,
  "items": [ {
    "id": "c5757d70-06f3-4f23-8367-5a9e55019f94",
    "companyProfile": {
      "tenantId": "c5757d70-06f3-4f23-8367-5a9e55019f94",
      "domain": "contoso190.onmicrosoft.com",
      "companyName": "Contoso190",
      "links": {
        "self": {
          "uri": "/customers/c5757d70-06f3-4f23-8367-5a9e55019f94/profiles/company",
          "method": "GET",
          "headers": []
        }
      },
      "attributes": {
        "objectType": "CustomerCompanyProfile"
      }
    },
    "relationshipToPartner": "reseller",
    "links": {
      "self": {
        "uri": "/customers/c5757d70-06f3-4f23-8367-5a9e55019f94",
        "method": "GET",
        "headers": []
      }
    },
    "attributes": {
      "objectType": "Customer"
    }
  }, {
    "id": "7b26b357-9ca3-48b8-a58e-4febe2662a5d",
    "companyProfile": {
      "tenantId": "7b26b357-9ca3-48b8-a58e-4febe2662a5d",
      "domain": "ContosoCorpCo.onmicrosoft.com",
      "companyName": "Contoso",
      "links": {
        "self": {
          "uri": "/customers/7b26b357-9ca3-48b8-a58e-4febe2662a5d/profiles/company",
          "method": "GET",
          "headers": []
        }
      },
      "attributes": {
        "objectType": "CustomerCompanyProfile"
      }
    },
    "relationshipToPartner": "reseller",
    "links": {
      "self": {
        "uri": "/customers/7b26b357-9ca3-48b8-a58e-4febe2662a5d",
        "method": "GET",
        "headers": []
      }
    },
    "attributes": {
      "objectType": "Customer"
    }
  }, {
    "id": "bfb6d6ef0-311f-47ec-bbd7-0fcb7846661b",
```

```

"companyProfile": {
  "tenantId": "bfb6ef0-311f-47ec-bbd7-0fcb7846661b",
  "domain": "contosocorpdemo.onmicrosoft.com",
  "companyName": "Contoso",
  "links": {
    "self": {
      "uri": "/customers/bfb6ef0-311f-47ec-bbd7-0fcb7846661b/profiles/company",
      "method": "GET",
      "headers": []
    }
  },
  "attributes": {
    "objectType": "CustomerCompanyProfile"
  }
},
"relationshipToPartner": "reseller",
"links": {
  "self": {
    "uri": "/customers/bfb6ef0-311f-47ec-bbd7-0fcb7846661b",
    "method": "GET",
    "headers": []
  }
},
"attributes": {
  "objectType": "Customer"
}
},
"links": {
  "self": {
    "uri": "/customers?
size=0&filter=%7B%22Field%22%3A%22Domain%22%2C%22Value%22%3A%22cont%22%2C%22Operator%22%3A%22starts_with%22%7D",
    "method": "GET",
    "headers": []
  }
},
"attributes": {
  "objectType": "Collection"
}
}

```

Get a list of all customers

Request syntax

METHOD	REQUEST URI
GET	{baseUrl}/v1/customers?size={size} HTTP/1.1

URI parameter

To get a list of customers, use the following query parameter:

NAME	TYPE	DESCRIPTION
size	int	The number of results to be displayed at one time.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers?size=40 HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: 3705fc6d-4127-4a87-bdba-9658f73fe019
MS-CorrelationId: b12260fb-82de-4701-a25f-dcd367690645
```

If the request is successful, this method returns a collection of customer resources in the response body.

Response example

```
HTTP/1.1 200 OK
Content-Length: 15650
Content-Type: application/json
MS-CorrelationId: b12260fb-82de-4701-a25f-dcd367690645
MS-RequestId: 3705fc6d-4127-4a87-bdba-9658f73fe019
Date: Fri, 20 Nov 2015 01:08:23 GMT
```

```
{
  "totalCount": 25,
  "items": [ {
    "id": "b44bb1fb-c595-45b0-9e09-d657365580bf",
    "companyProfile": {
      "tenantId": "<guid>",
      "domain": "domain",
      "companyName": "companyName",
      "attributes": {
        "objectType": "CustomerCompanyProfile"
      }
    },
    "relationshipToPartner": "reseller",
    "attributes": {
      "objectType": "Customer"
    }
  },
  {
    "id": "45c44870-ef77-4fdd-b6fe-3dacb075cff2",
    "companyProfile": {
      "tenantId": "<guid>",
      "domain": "domain",
      "companyName": "companyName",
      "attributes": {
        "objectType": "CustomerCompanyProfile"
      }
    },
    "relationshipToPartner": "reseller",
    "attributes": {
      "objectType": "Customer"
    }
  }
],
  "links": {
    "self": {
      "uri": "/v1/customers?size=40",
      "method": "GET",
      "headers": []
    }
  },
  "attributes": {
    "objectType": "Collection"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- View a list of [Azure CSP integration scenarios](#).

Get a user account by ID

9/19/2017 • 1 min to read • [Edit Online](#)

This guide covers how to retrieve a specific user account for a customer by providing an ID.

PowerShell

To retrieve a specific user account, enter the following commands:

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'

$user = Get-PCCustomerUser -tenantid $customer.id -userid '<user id>'
```

C#

To retrieve a user account for a customer, do the following:

1. To identify the customer, call the **IAggregatePartner.Customers.ById** method with the customer ID.
2. To retrieve the specific user, call the **Users.ById** method.
3. To retrieve the user account, call the **Users.Get** or **GetAsync** method.

```
// IAggregatePartner partnerOperations;
// string selectedCustomerId;
// string selectedCustomerUserId;

// Get customer user detail.
var customerUsers = partnerOperations.Customers.ById(selectedCustomerId).Users.ById(selectedCustomerUserId).Get();
```

REST API

Request

Request syntax

METHOD	REQUEST URI
GET	<i>{baseUrl}/v1/customers/{customer-tenant-id}/users/{user-id}</i> HTTP/1.1

URI parameter

To identify the correct customer and user, use the following URI parameters:

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	A GUID-formatted string that allows the reseller to filter the results for a specific customer that belongs to the reseller.

NAME	TYPE	DESCRIPTION
user-id	guid	A GUID-formatted string that belongs to a single user account.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/users/a9ef48bb-8758-4590-a312-d4a47bfaded4
HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: c1f673cb-655c-45a7-8a6b-257a0a006f4b
MS-CorrelationId: 24a631eb-a110-49dc-8325-99d4b196774c
X-Locale: en-US
Host: api.partnercenter.microsoft.com
```

Response

If the request is successful, this method returns the user account for the customer.

Response example

```
HTTP/1.1 200 OK
Content-Length: 432
Content-Type: application/json; charset=utf-8
MS-CorrelationId: 24a631eb-a110-49dc-8325-99d4b196774c
MS-RequestId: c1f673cb-655c-45a7-8a6b-257a0a006f4b
MS-CV: uWM1EGU7+0aI2MvV.0
MS-ServerId: 020021921
Date: Wed, 21 Dec 2016 22:59:10 GMT

{
  "usageLocation": "US",
  "id": "a9ef48bb-8758-4590-a312-d4a47bfaded4",
  "userPrincipalName": "Daniel@dtmocspcustomer005.onmicrosoft.com",
  "firstName": "Daniel",
  "lastName": "Tsai",
  "displayName": "Daniel Tsai",
  "userDomainType": "none",
  "state": "active",
  "links": {
    "self": {
      "uri": "/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/users/a9ef48bb-8758-4590-a312-d4a47bfaded4",
      "method": "GET",
      "headers": []
    }
  },
  "attributes": {
    "objectType": "CustomerUser"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- View a list of [Azure CSP integration scenarios](#).

Get a list of all users

9/19/2017 • 1 min to read • [Edit Online](#)

This guide covers how to retrieve a list of all user accounts for a specific customer. You can retrieve the list either in the Microsoft Partner Center web UI or through API calls.

Web UI

To retrieve a list of all of a customer's users in Partner Center, go to **Customers > Users and Licenses**.

C#

To retrieve the collection of all user accounts for a specified customer, do the following:

1. To identify the customer, call the **IAggregatePartner.Customers.ById** method with the specified customer ID.
2. To retrieve the collection, call the **Users.Get** or **GetAsync** method.

```
// IAggregatePartner partnerOperations;  
// string selectedCustomerId;  
  
// Get customer users collection.  
var customerUsers = partnerOperations.Customers.ById(selectedCustomerId).Users.Get();
```

REST API

Request syntax

METHOD	REQUEST URI
GET	{baseUrl}/v1/customers/{customer-tenant-id}/users HTTP/1.1

URI parameters

NAME	NAME	DESCRIPTION
customer-tenant-id	GUID	A GUID that represents the customer.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/users HTTP/1.1  
Authorization: Bearer <token>  
Accept: application/json  
MS-RequestId: 5d845377-5b7d-4cd4-98f6-19e5ae3faa81  
MS-CorrelationId: 5a3d64d4-4490-4932-bf5e-0dc9a58f27ca  
X-Locale: en-US  
Host: api.partnercenter.microsoft.com
```

Response example

If the request is successful, this method returns a collection of users.

HTTP/1.1 200 OK
Content-Length: 1030
Content-Type: application/json; charset=utf-8
MS-CorrelationId: 5a3d64d4-4490-4932-bf5e-0dc9a58f27ca
MS-RequestId: 5d845377-5b7d-4cd4-98f6-19e5ae3faa81
MS-CV: 6zmKqrSFB0+t7m3y.0
MS-ServerId: 101112616
Date: Wed, 21 Dec 2016 21:13:24 GMT

```
{
  "totalCount": 2,
  "items": [{
    "usageLocation": "US",
    "id": "a9ef48bb-8758-4590-a312-d4a47bfaded4",
    "userPrincipalName": "Daniel@dt democspcustomer005.onmicrosoft.com",
    "firstName": "Daniel",
    "lastName": "Tsai",
    "displayName": "Daniel Tsai",
    "userDomainType": "none",
    "state": "active",
    "links": {
      "self": {
        "uri": "/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/users/a9ef48bb-8758-4590-a312-d4a47bfaded4",
        "method": "GET",
        "headers": []
      }
    },
    "attributes": {
      "objectType": "CustomerUser"
    }
  }, {
    "id": "6e668259-1f09-479d-bcb8-d9b03e826b8d",
    "userPrincipalName": "admin@dt democspcustomer005.onmicrosoft.com",
    "firstName": "Daniel",
    "lastName": "Tsai",
    "displayName": "DT Demo CSP Customer 005",
    "userDomainType": "none",
    "state": "active",
    "links": {
      "self": {
        "uri": "/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/users/6e668259-1f09-479d-bcb8-d9b03e826b8d",
        "method": "GET",
        "headers": []
      }
    },
    "attributes": {
      "objectType": "CustomerUser"
    }
  }
],
  "links": {
    "self": {
      "uri": "/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/users",
      "method": "GET",
      "headers": []
    }
  },
  "attributes": {
    "objectType": "Collection"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- View a list of [Azure CSP integration scenarios](#).

Get a list of user roles for a customer

9/19/2017 • 1 min to read • [Edit Online](#)

This guide covers how to retrieve a list of roles that users of a particular customer hold. You can accomplish this task with the C# or REST APIs.

Retrieve roles by using C#

To retrieve all the directory roles for a specified customer, take the following steps:

1. Retrieve the specified customer ID. Then use the **IAggregatePartner.Customers** collection followed by the **ById()** method.
2. Call the **DirectoryRoles** property, followed by the **Get()** or **GetAsync()** method.

```
// string selectedCustomerId;  
// IAggregatePartner partnerOperations;  
  
var directoryRoles = partnerOperations.Customers.ById(selectedCustomerId).DirectoryRoles.Get();
```

To retrieve a list of customer users that have a given role, take the following steps:

1. Retrieve the specified customer ID and the directory role ID.
2. Use the **IAggregatePartner.Customers** collection to call the **ById()** method.
3. Call the **DirectoryRoles** property, followed by the **ById()** method.
4. Call the **UserMembers** property, followed by the **Get()** or **GetAsync()** method.

```
// string selectedCustomerId;  
// IAggregatePartner partnerOperations;  
// string selectedDirectoryRoleId;  
  
var userMembers = partnerOperations.Customers.ById(selectedCustomerId).DirectoryRoles.ById(selectedDirectoryRoleId).UserMembers.Get();
```

Retrieve roles by using the REST API

Request syntax

METHOD	REQUEST URI
GET	{baseUrl}/v1/customers/{customer-tenant-id}/users/{user-id}/directoryroles HTTP/1.1
GET	{baseUrl}/v1/customers/{customer-tenant-id}/directoryroles HTTP/1.1
GET	{baseUrl}/v1/customers/{customer-tenant-id}/directoryroles/{role-ID}/usermembers

Request parameters

NAME	TYPE	DESCRIPTION
customer-tenant-id	GUID	Allows the reseller to filter the results for a given customer that belongs to the reseller.
user-id	GUID	Belongs to a single user account.
role-id	GUID	Belongs to a type of role. You can get these IDs by querying all the directory roles for a customer, across all user accounts (as described earlier in the second scenario).

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers/<customer-tenant-id>/users/<user-id>/directoryroles HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: b1317092-f087-471e-a637-f66523b2b94c
MS-CorrelationId: 8a53b025-d5be-4d98-ab20-229d1813de76
```

Response example

If the request is successful, this method returns a list of the roles that are associated with the given user account.

```
HTTP/1.1 200 OK
Content-Length: 31942
Content-Type: application/json
MS-CorrelationId: 8a53b025-d5be-4d98-ab20-229d1813de76
MS-RequestId: b1317092-f087-471e-a637-f66523b2b94c
Date: June 24 2016 22:00:25 PST
```

```
{
  "totalCount": 2,
  "items": [
    {
      "name": "Helpdesk Administrator",
      "id": "729827e3-9c14-49f7-bb1b-9608f156bbb8",
      "attributes": { "objectType": "DirectoryRole" }
    },
    {
      "name": "User Account Administrator",
      "id": "fe930be7-5e62-47db-91af-98c3a49a38b1",
      "attributes": { "objectType": "DirectoryRole" }
    }
  ],
  "attributes": { "objectType": "Collection" }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Manage user permissions

9/19/2017 • 5 min to read • [Edit Online](#)

Learn about the different ways you can manage the roles and permissions of users, both for a customer and for you, the Azure CSP partner. You can use one of the following options:

- Partner Center web UI
- PowerShell
- C#
- REST API

Partner Center web UI

Retrieve user permissions

Before you can modify a user's permissions, you must first find out what permissions the user has.

1. In Microsoft Partner Center, on the **Dashboard** menu, go to **Customers**. Select the specific customer.
2. Select **Users and licenses**, and then select the specific user.

The screenshot shows the Microsoft Partner Center interface. The top navigation bar includes 'Microsoft', 'Cloud', 'Mobility', and 'Productivity'. The main header shows 'Microsoft Partner Center' with links to 'Programs', 'How-to', 'Support', and 'Find a Partner'. The left sidebar for 'Dakota mall' lists 'Subscriptions', 'Customer insights', 'Users and licenses' (highlighted with a red box), 'Service management', and 'Account'. The main content area is titled 'Users and licenses' and includes buttons for 'Add user' and 'Upload users'. A search bar is present with the text 'Search by name or by email'. Below this is a table of users:

<input type="checkbox"/>	Name	Email	Location
<input type="checkbox"/>	Dakota mall	admin@dakotamall.onmicrosoft.com	
<input type="checkbox"/>	JohnDoe	jd1990@dakotamall.onmicrosoft.com	United States

The 'JohnDoe' user is highlighted with a red box. At the bottom of the page, there is a footer with links organized into columns: Industries, For customers, For partners, For developers, Values, and Company.

3. The user's permissions are displayed under **Permissions**. For an explanation of what the different permissions mean, see [Partner Center overview](#).

Edit permissions

To change the permissions you see on the user's permissions page, select or clear the permissions. You can set a user role to **User** or **Global admin**. Or, you can select **Custom permissions** to open new options for **Password admin**, **User management admin**, or **Service admin**.

Dakota mail JohnDoe

Subscriptions
Customer insights
Users and licenses
Update user
Service management
Account

← Customers

User info

First name: John
Last name: Doe
Display name: JohnDoe
Email: jd1990@dakotamail.onmicrosoft.com
Location: United States

Permissions

- ☐ User (can use licenses and services)
- ☐ Global admin (has full permissions)
- ☒ Custom permissions
 - ☐ Password admin (can reset user passwords)
 - ☒ User management admin (can manage user accounts)
 - ☐ Service admin (can manage services)

Assign licenses

Product Available licenses

When you are finished making changes, select **Submit**.

PowerShell

Retrieve a user role

To get a list of a customer user's roles by using PowerShell, use the following commands:

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'

$user = Get-PCCustomerUser -tenantid $customer.id -userid '<user id>'

Get-PCCustomerUserRole -tenantid $customer.id -user $user
```

Set a user role

To add a user to a specific role by using PowerShell, use the following commands:

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'

$role = Get-PCCustomerRole -tenantid $customer.id | Where-Object name -Contains '<role name>'

$user = Get-PCCustomerUser -tenantid $customer.id -userid '<user id guid>'

$customerRoleMember = [DirectoryRoleMember]::new()

$customerRoleMember.id = $user.id

Add-PCCustomerRoleMember -tenantid $customer.id -roleid $role.id -customerrolemember $customerRoleMember
```

Remove a user role

To remove a user from a role by using PowerShell, use the following commands:

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'

$role = Get-PCDirectoryRoles -tenantid $customer.id | Where-Object name -Contains '<role name>'

$user = Get-PCCustomerUser -tenantid $customer.id -userid '<user id guid>'

Remove-PCCustomerRoleMember -tenantid $customer.id -roleid $role.id -userid $user.id
```

C#

Retrieve a user role

To retrieve all the directory roles for a specific customer:

1. Retrieve the specified customer ID.
2. Use your **IAggregatePartner.Customers** collection and call the **ById()** method.
3. Call the **DirectoryRoles** property, followed by the **Get()** or the **GetAsync()** method.

```
// string selectedCustomerId;
// IAggregatePartner partnerOperations;

var directoryRoles = partnerOperations.Customers.ById(selectedCustomerId).DirectoryRoles.Get();
```

To retrieve a list of customer users that are assigned a specific role:

1. Retrieve the specified customer ID and the directory role ID.
2. Use your **IAggregatePartner.Customers** collection and call the **ById()** method.
3. Call the **DirectoryRoles** property, then the **ById()** method, and then the **UserMembers** property.
4. Call the **Get()** or the **GetAsync()** method.

```
// string selectedCustomerId;
// IAggregatePartner partnerOperations;
// string selectedDirectoryRoleId;

var userMembers = partnerOperations.Customers.ById(selectedCustomerId).DirectoryRoles.ById(selectedDirectoryRoleId).UserMembers.Get();
```

Set a user role

To assign a directory role to a customer user:

1. Create a new **UserMember** with the relevant user details.
2. To identify the customer, call the **IAggregatePartner.Customers.ById** method with the specified customer ID.
3. To specify the role, use the **DirectoryRoles.ById** method with the directory role ID.
4. To add the new user member to the collection of user members assigned to that role, access the **UserMembers** collection, and then use the **Create** method.

```
// UserMember createdUser;
// IAggregatePartner partnerOperations;
// Customer selectedCustomer;
// IDirectoryRole selectedRole;

// Create the new user member.
UserMember userMemberToAdd = new UserMember()
{
    UserPrincipalName = createdUser.UserPrincipalName,
    DisplayName = createdUser.DisplayName,
    Id = createdUser.Id
};

// Add the new user member to the role.
var userMemberAdded =
partnerOperations.Customers.ById(selectedCustomer.Id).DirectoryRoles.ById(selectedRole.Id).UserMembers.Create(userMemberToAdd);
```

Remove a user role

To remove a user from a directory role:

1. To select the customer that has the user whose permissions you want to modify, call the **IAggregatePartner.Customers.ById** method.
2. To specify the role, use the **DirectoryRoles.ById** method with the directory role ID.
3. To identify the user to remove, access the **UserMembers.ById** method.
4. To remove the user from the role, call the **Delete** method.

```
// IAggregatePartner partnerOperations;
// string selectedCustomerId;
// string selectedRoleId;
// string selectedUserMemberId;

partnerOperations.Customers.ById(selectedCustomerId).DirectoryRoles.ById(selectedRoleId).UserMembers.ById(selectedUserMemberId).Delete();
```

REST API

Retrieve a user role

Request syntax

METHOD	REQUEST URI
GET	{baseUrl}/v1/customers/{customer-tenant-id}/users/{user-id}/directoryroles HTTP/1.1
GET	{baseUrl}/v1/customers/{customer-tenant-id}/directoryroles HTTP/1.1
GET	{baseUrl}/v1/customers/{customer-tenant-id}/directoryroles/{role-ID}/usermembers

URI parameters

Use the following query parameters to identify the correct customer.

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	A GUID-format customer tenant ID. The reseller can use the value for customer-tenant-id to filter the results for a specific customer that is associated with the reseller.
user-id	guid	(Optional) A GUID-format user ID that belongs to a single user account.
role-id	guid	A GUID-format role ID that belongs to a type of role. You can get these IDs by querying all the directory roles for a customer, across all user accounts.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers/<customer-tenant-id>/users/<user-id>/directoryroles HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: b1317092-f087-471e-a637-f66523b2b94c
MS-CorrelationId: 8a53b025-d5be-4d98-ab20-229d1813de76
```

If the request is successful, this method returns a list of the roles associated with the user account.

Response example

```
HTTP/1.1 200 OK
Content-Length: 31942
Content-Type: application/json
MS-CorrelationId: 8a53b025-d5be-4d98-ab20-229d1813de76
MS-RequestId: b1317092-f087-471e-a637-f66523b2b94c
Date: June 24 2016 22:00:25 PST
```

```
{
  "totalCount": 2,
  "items": [
    {
      "name": "Helpdesk Administrator",
      "id": "729827e3-9c14-49f7-bb1b-9608f156bbb8",
      "attributes": { "objectType": "DirectoryRole" }
    },
    {
      "name": "User Account Administrator",
      "id": "fe930be7-5e62-47db-91af-98c3a49a38b1",
      "attributes": { "objectType": "DirectoryRole" }
    }
  ],
  "attributes": { "objectType": "Collection" }
}
```

Set a user role

Request syntax

METHOD	REQUEST URI
POST	{baseUrl}/v1/customers/{customer-tenant-id}/directoryroles/{role-ID}/usermembers HTTP/1.1

URI parameters

Use the following URI parameters to identify the correct customer and role.

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	The value is a GUID-format customer tenant ID. The reseller can use the value for customer-tenant-id to filter the results for a specific customer that is associated with the reseller.
role-id	guid	The value is a GUID-format role ID that identifies the role to assign to the user.

Request body

The following table lists the properties that you must include in the request body.

NAME	TYPE	DESCRIPTION
id	string	The ID of the user you are adding to the role.
DisplayName	string	The display name of the user.
UserPrincipalName	string	The name of the user principal.
Attributes	object	Contains <code>ObjectType</code> : <code>UserMember</code>

Request example

```
POST https://api.partnercenter.microsoft.com/v1/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/directoryroles/f023fd81-a637-4b56-95fd-791ac0226033/usermembers HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: a56cb2e5-a156-4f68-9155-57ffe2b93d18
MS-CorrelationId: 90bda268-7929-4ad6-be01-89c5af5fc504
X-Locale: en-US
Content-Type: application/json
Host: api.partnercenter.microsoft.com
Content-Length: 180
Expect: 100-continue

{
  "Id": "a9ef48bb-8758-4590-a312-d4a47bfaded4",
  "DisplayName": "Donovan Luong",
  "UserPrincipalName": "Donovan@dt democspcustomer005.onmicrosoft.com",
  "Attributes": {
    "ObjectType": "UserMember"
  }
}
```

When the user is successfully assigned the role, this method returns the user account, along with the role ID.

Response example

HTTP/1.1 201 Created
Content-Length: 231
Content-Type: application/json; charset=utf-8
MS-CorrelationId: 90bda268-7929-4ad6-be01-89c5af5fc504
MS-RequestId: a56cb2e5-a156-4f68-9155-57ffe2b93d18
MS-CV: aia94+gnrEeQqkGr.0
MS-ServerId: 101112202
Date: Tue, 20 Dec 2016 23:36:55 GMT

```
{
  "displayName": "Donovan Luong",
  "userPrincipalName": "Donovan@dttdemocspcustomer005.onmicrosoft.com",
  "roleId": "f023fd81-a637-4b56-95fd-791ac0226033",
  "id": "a9ef48bb-8758-4590-a312-d4a47bfaded4",
  "attributes": {
    "objectType": "UserMember"
  }
}
```

Remove a user role

Request syntax

METHOD	REQUEST URI
DELETE	{baseUrl}/v1/customers/{customer-tenant-id}/directoryroles/{role-ID}/usermembers/{user-ID} HTTP/1.1

URI parameters

The following URI parameters are required for identifying the correct customer, role, and user.

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	A GUID-format customer tenant ID that identifies the customer.
role-id	guid	A GUID-format role ID that identifies the role.
user-id	guid	A GUID-format user ID that identifies a single user account.

Request example

```
DELETE https://api.partnercenter.microsoft.com/v1/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04%20/directoryroles/729827e3-9c14-49f7-bb1b-9608f156bbb8/usermembers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04%20 HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: 0a00ec08-6273-46bb-ab6f-14a13959b381
MS-CorrelationId: 87d18a45-81fc-40cf-921a-b91cb82d67fe
X-Locale: en-US
Host: api.partnercenter.microsoft.com
Content-Length: 0
Connection: Keep-Alive
```

If the user is successfully removed from the role, the response body is empty.

Response example

HTTP/1.1 204 No Content
Content-Length: 0
MS-CorrelationId: 90bda268-7929-4ad6-be01-89c5af5fc504
MS-RequestId: e784d7aa-8c8d-45ee-8f97-9e09823d7338
MS-CV: es01VX8do0u2aTXw.0
MS-ServerId: 101112616
Date: Tue, 20 Dec 2016 23:16:35 GMT

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Reset a user password

9/19/2017 • 2 min to read • [Edit Online](#)

Learn how to reset a user password for a customer, if you need to. You can use one of the following options:

- Partner Center web UI
- PowerShell
- C#
- REST API

NOTE

This task is not the same as resetting a user password in Microsoft Partner Center.

Partner Center web UI

1. In Partner Center, on the **Dashboard** menu, select **Customers**. Select the customer.
2. On the customer menu, select **Users and licenses**. Select the user.

The screenshot shows the Microsoft Partner Center web UI. The top navigation bar includes the Microsoft logo, dropdown menus for Cloud, Mobility, and Productivity, a search bar for partner content, and a user profile icon. The main header shows 'Microsoft Partner Center' with links to Programs, How-to, Support, and Find a Partner. The 'Dashboard' link is highlighted. The left sidebar shows the 'Dakota mall' customer profile with links to Subscriptions, Customer insights, Users and licenses (highlighted with a red box), Service management, and Account. The main content area is titled 'Users and licenses' and includes a search bar and buttons for 'Add user' and 'Upload users'. Below this is a table of users:

<input type="checkbox"/>	Name	Email	Location
<input type="checkbox"/>	Dakota mall	admin@dakotamall.onmicrosoft.com	
<input type="checkbox"/>	JohnDoe	jd1990@dakotamall.onmicrosoft.com	United States

The 'JohnDoe' user is highlighted with a red box. The bottom of the page features a grid of links for various Microsoft services and resources.

3. Select **Reset password**.

← Customers

Location ^{What is this?}
United States

Permissions

☐ User (can use licenses and services)
☐ Global admin (has full permissions)
☒ Custom permissions

☐ Password admin (can reset user passwords)
☒ User management admin (can manage user accounts)
☐ Service admin (can manage services)

Assign licenses

Product	Available licenses
<input checked="" type="checkbox"/> Azure Active Directory Basic	0

[Submit](#)
[Cancel](#)
[Reset password](#)
[Delete user](#)

4. You are redirected to the **Password has been reset** page (it might take a few minutes). Send the new, temporary password to the user.

Microsoft Cloud Mobility Productivity Search partner content

Microsoft Partner Center Programs How-to Support Find a Partner Dashboard

Dakota mail

Subscriptions
Customer insights
Users and licenses
Update user
Service management
Account

← Customers

Password has been reset

Username jd1990@dakotamail.onmicrosoft.com

Temporary password 5Cdxk5P_

Copy the email and temporary password to give to the user. After you leave this page, you can't view this temporary password again. Updates take a few minutes to complete.

[Done](#)

Industries	For customers	For partners	For developers	Values	Company
Manufacturing & resources	Small & midsize businesses	Get listed	Microsoft Azure	Diversity and inclusion	Careers
Financial services	Enterprise	Microsoft partner resources	Microsoft Visual Studio	Accessibility	About Microsoft
Retail	Security at home	Find a solutions provider	Microsoft Developer Network	Microsoft in education	Company news
Health	Microsoft Envision	Partner with Microsoft Azure	TechNet	Microsoft philanthropies	Investors
Government	Microsoft Ignite	Microsoft Inspire	Application Development	Corporate social responsibility	Research
Education	Microsoft AppSource			Privacy at Microsoft	Site map

PowerShell

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'

$user = Get-PCCustomerUser -tenantid $customer.id -userid '<user id>'

$password = '<password>'
$passwordSecure = $password | ConvertTo-SecureString -AsPlainText -Force

Set-PCCustomerUser -tenantid $customer.id -user $user -password $passwordSecure -forceChangePassword $true/$false
```

C#

1. Retrieve the specified customer ID and the targeted user.
2. Create a new **CustomerUser** object that contains the information for the existing customer, but with a new

PasswordProfile object.

3. Use your **IAggregatePartner.Customers** collection and call the **ById()** method.
4. Call the **Users** property, the **ById()** method, and then the **Patch()** method.

```
// string selectedCustomerId;
// IAggregatePartner partnerOperations;
// CustomerUser specifiedUser;

var selectedCustomer = partnerOperations.Customers.ById(selectedCustomerId).Get();
var userToUpdate = new CustomerUser()
{
    PasswordProfile = new PasswordProfile() { ForceChangePassword = true, Password = "newPassword" },
    DisplayName = "User name",
    FirstName = "First",
    LastName = "Last",
    UsageLocation = "US",
    UserPrincipalName = Guid.NewGuid().ToString("N") + "@" + selectedCustomer.CompanyProfile.Domain.ToString()
};

// Update customer user information.
User updatedCustomerUserInfo = partnerOperations.Customers.ById(selectedCustomerId).Users.ById(specifiedUser.Id).Patch(userToUpdate);
```

REST API

Request

Request syntax

METHOD	REQUEST URI
PATCH	{baseUrl}/v1/customers/{customer-tenant-id}/users HTTP/1.1

URI parameters

The following query parameters are required for identifying the correct customer.

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	A GUID-format customer tenant ID. The reseller can use the customer tenant ID to filter results for a specific customer that is associated with the reseller.
user-id	guid	A GUID-format user ID that belongs to a single user account.

Request example

```
PATCH https://api.partnercenter.microsoft.com/v1/customers/<customer-tenant-id>/users/<user-id> HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: b1317092-f087-471e-a637-f66523b2b94c
MS-CorrelationId: 8a53b025-d5be-4d98-ab20-229d1813de76
{
  "passwordProfile": {
    password: "Renew456*",
    forceChangePassword: true
  },

  "attributes": {
    "objectType": "CustomerUser"
  }
}
```

Response

If the request is successful, this method returns the user information and the updated password.

Response example

```
HTTP/1.1 200 OK
Content-Length: 31942
Content-Type: application/json
MS-CorrelationId: 8a53b025-d5be-4d98-ab20-229d1813de76
MS-RequestId: b1317092-f087-471e-a637-f66523b2b94c
Date: June 24 2016 22:00:25 PST
{
  "usageLocation": "AX",
  "id": "95794928-9abe-4548-8b43-50ffc20b9404",
  "userPrincipalName": "username@contoso.onmicrosoft.com",
  "firstName": "First",
  "lastName": "Last",
  "displayName": "User name",
  "passwordProfile": {
    "forceChangePassword": false,
    "password": "Renew456*"
  },
  "lastDirectorySyncTime": null,
  "userDomainType": "none",
  "state": "active",
  "softDeletionTime": null,
  "links": {
    "self": {
      "uri": "/customers/eebd1b55-5360-4438-a11d-5c06918c3014/users/95794928-9abe-4548-8b43-50ffc20b9404",
      "method": "GET",
      "headers": [

    ]
    }
  },
  "attributes": {
    "objectType": "CustomerUser"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

View a customer's deleted users

9/19/2017 • 2 min to read • [Edit Online](#)

When a user account is deleted, the user state is set to "inactive." The user account remains in an inactive state for 30 days. After 30 days, the account and its associated data is purged. Purged data cannot be recovered.

Within the 30-day window, deleted accounts can be restored. For more information about restoring a deleted user, see [Manage user accounts](#).

Inactive user accounts do not return as part of the user collection. You must query them separately. You can use one of the following options:

- PowerShell
- C#
- REST API

PowerShell

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'

$user = Get-PCCustomerUser -tenantid $customer.id -deleted | ? id -EQ '<user id>'
```

C#

To retrieve a list of deleted users, construct a query that filters for customer users whose status is set to inactive:

1. Create the filter by instantiating a **SimpleFieldFilter** object by using the parameters shown in the following code example.
2. Create the query by using the **BuildIndexedQuery** method. If you do not want paged results, you can use the **BuildSimpleQuery** method instead.
3. To identify the customer, use the **IAggregatePartner.Customers.ById** method with the customer ID.
4. To send the request, call the **Query** method.

```
// IAggregatePartner partnerOperations;
// int customerUserPageSize;

// Create a filter for users whose status is inactive (deleted).
var filter = new SimpleFieldFilter("UserState", FieldFilterOperation.Equals, "Inactive");

// Build a paged query.
var simpleQueryWithFilter = QueryFactory.Instance.BuildIndexedQuery(customerUserPageSize, 0, filter);

// Send the request.
var customerUsers = partnerOperations.Customers.ById(selectedCustomerId).Users.Query(simpleQueryWithFilter);
```

REST API

Request

Request syntax

METHOD	REQUEST URI
GET	{baseUrl}/v1/customers/{customer-id}/users?size={size}&filter={filter} HTTP/1.1

URI parameters

Use the following path and query parameters when you create the request.

NAME	TYPE	DESCRIPTION
customer-id	guid	A GUID-format customer ID that identifies the customer.
size	int	(Optional) The number of results to display at one time.
filter	filter	The query that filters the user search. To retrieve deleted users, you must include and encode the following string: {"Field": "UserState", "Value": "Inactive", "Operator": "equals"}.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/users?
size=500&filter=%7B%22Field%22%3A%22UserState%22%2C%22Value%22%3A%22Inactive%22%2C%22Operator%22%3A%22equals%22%7D
HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: c11feb95-55d2-45b6-9d1b-74b55d2221fb
MS-CorrelationId: 2b4ab588-f48c-4874-b479-a61895e107b2
X-Locale: en-US
Host: api.partnercenter.microsoft.com
```

Response

If the request is successful, this method returns a collection of **CustomerUser** resources in the response body.

Response example

HTTP/1.1 200 OK
Content-Length: 802
Content-Type: application/json; charset=utf-8
MS-CorrelationId: 690b34ca-07c8-4f8a-ab13-f22a50594a43
MS-RequestId: 1187f9ad-02b4-4d96-b668-7cf3d289467b
MS-CV: 3TLmR9gz6EaCVCjR.0
MS-ServerId: 101112616
Date: Fri, 20 Jan 2017 19:13:14 GMT

```
{
  "totalCount": 1,
  "items": [{
    "usageLocation": "US",
    "id": "a45f1416-3300-4f65-9e8d-f123b397a4ea",
    "userPrincipalName": "e83763f7f2204ac384cfd49f79f2749@dttdemospcustomer005.onmicrosoft.com",
    "firstName": "Ferdinand",
    "lastName": "Filibuster",
    "displayName": "Ferdinand",
    "userDomainType": "none",
    "state": "inactive",
    "softDeletionTime": "2017-01-20T00:33:34Z",
    "links": {
      "self": {
        "uri": "/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/users/a45f1416-3300-4f65-9e8d-f123b397a4ea",
        "method": "GET",
        "headers": []
      }
    },
    "attributes": {
      "objectType": "CustomerUser"
    }
  ]},
  "links": {
    "self": {
      "uri": "/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/users?size=500&filter=%7B%22Field%22%3A%22UserStatus%22%2C%22Value%22%3A%22Inactive%22%2C%22Operator%22%3A%22equals%22%7D",
      "method": "GET",
      "headers": []
    }
  },
  "attributes": {
    "objectType": "Collection"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Create an order

9/19/2017 • 3 min to read • [Edit Online](#)

This guide covers how to create an order for a customer by using the Microsoft Partner Center APIs.

PowerShell

To create an order in PowerShell, enter the following commands:

```
# Get a customer
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'
# Get offer
$offer = Get-PCOffer -countryid '<country two digits id>' -offerid '<offer id GUID>'
# Create the OrderLineItem
$lineItems = @()
$lineItems += [OrderLineItem]::new()
$lineItems[0].LineNumber = 0
$lineItems[0].FriendlyName = '<friendly name>'
$lineItems[0].OfferId = $offer.id
$lineItems[0].Quantity = <quantity>
# Send order
New-PCOrder -tenantid $customer.id -LineItems $lineItems
```

C#

To create an order for a customer in C#, do the following:

1. To record the customer, instantiate an **Order** object, and then set the **ReferenceCustomerId** property to the customer ID to record the customer.
2. Create a list of **OrderLineItem** objects, and then assign the list to the order's **LineItems** property. Each order line item contains the purchase information for one offer. You must have at least one order line item.
3. To identify the customer, obtain an interface to order operations by calling the **IAggregatePartner.Customers.ById** method with the customer ID, and then retrieve the interface from the **Orders** property.
4. To create the order, call the **Create** or **CreateAsync** method.

```

IAggregatePartner partnerOperations;
string customerId;
string offerId;

var order = new Order()
{
    ReferenceCustomerId = customerId,
    LineItems = new List<OrderLineItem>()
    {
        new OrderLineItem()
        {
            OfferId = offerId,
            FriendlyName = "new offer purchase",
            Quantity = 5
        }
    }
};

var createdOrder = partnerOperations.Customers.ById(customerId).Orders.Create(order);

```

REST API

Request syntax

METHOD	REQUEST URI
POST	{baseUrl}/v1/customers/{customer-id}/orders HTTP/1.1

URI parameters

NAME	TYPE	REQUIRED	DESCRIPTION
customer-id	string	A GUID-formatted string that identifies the customer.	

Request body

The request body contains the following **Order** properties:

NAME	TYPE	REQUIRED?	DESCRIPTION
id	string	N	The order identifier that is supplied when the order is created successfully.
referenceCustomerId	string	Y	The customer identifier.
billingCycle	string	N	The frequency with which the partner is billed for this order. The default is <i>monthly</i> , and it is applied when the order is created successfully. Supported values are the member names found in BillingCycleType . Note that annual billing is not yet supported.

NAME	TYPE	REQUIRED?	DESCRIPTION
lineItems	array of objects	Y	An array of OrderLineItem resources.
creationDate	string	N	The date when the order was created, in date-time format. It is applied when the order is created successfully.
attributes	object	N	Contains the Order object type.

The request body contains the following **OrderLineItem** object properties:

NAMNE	TYPE	REQUIRED?	DESCRIPTION
lineItemNumber	int	Y	Each line item in the collection gets a unique line number, counting up from 0 to count-1.
offerId	string	Y	The offer identifier.
subscriptionId	string	N	The subscription identifier.
parentSubscriptionId	string	N	The ID of the parent subscription in an add-on offer. Applies to PATCH only.
friendlyName	string	N	The friendly name of the subscription, as defined by the partner to help disambiguate.
quantity	int	Y*	The number of licenses for a license-based subscription. *Required only for license-based subscriptions.
partnerIdOnRecord	string	N*	When an indirect provider places an order on behalf of an indirect reseller, populate this field with the MPN ID of the <i>indirect reseller only</i> , never the ID of the indirect provider. This approach ensures proper accounting for incentives.
attributes	object	N	Contains the OrderLineItem object type.

Request example

```
POST https://api.partnercenter.microsoft.com/v1/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/orders HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: 57870501-203b-468e-8a63-078a3826d8ec
MS-CorrelationId: 9c272436-538d-4dd4-a421-c811e004784c
X-Locale: en-US
Content-Type: application/json
Host: api.partnercenter.microsoft.com
Content-Length: 405
Expect: 100-continue
Connection: Keep-Alive
```

```
{
  "Id": null,
  "ReferenceCustomerId": "4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04",
  "BillingCycle": "unknown",
  "LineItems": [{
    "LineItemNumber": 0,
    "OfferId": "84A03D81-6B37-4D66-8D4A-FAEA24541538",
    "SubscriptionId": null,
    "ParentSubscriptionId": null,
    "FriendlyName": "new offer purchase",
    "Quantity": 5,
    "PartnerIdOnRecord": null,
    "Attributes": {
      "ObjectType": "OrderLineItem"
    }
  }
],
  "CreationDate": null,
  "Attributes": {
    "ObjectType": "Order"
  }
}
```

If the request is successful, the response body contains the populated **Order** resource.

Response example

HTTP/1.1 201 Created
Content-Length: 801
Content-Type: application/json; charset=utf-8
MS-CorrelationId: 9c272436-538d-4dd4-a421-c811e004784c
MS-RequestId: 57870501-203b-468e-8a63-078a3826d8ec
MS-CV: VcfS+fqdQUW8Nap6.0
MS-ServerId: 030020525
Date: Thu, 30 Mar 2017 17:43:08 GMT

```
{
  "id": "074bd849-9106-405c-9923-fa061839d487",
  "referenceCustomerId": "4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04",
  "billingCycle": "monthly",
  "lineItems": [ {
    "lineItemNumber": 0,
    "offerId": "84A03D81-6B37-4D66-8D4A-FAEA24541538",
    "subscriptionId": "1DA3295E-C59A-476C-A7E7-D7E981FE79BE",
    "friendlyName": "new offer purchase",
    "quantity": 5,
    "links": {
      "subscription": {
        "uri": "/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/subscriptions/1DA3295E-C59A-476C-A7E7-D7E981FE79BE",
        "method": "GET",
        "headers": []
      }
    }
  }
],
  "creationDate": "2017-03-30T10:43:07.157-07:00",
  "links": {
    "self": {
      "uri": "/customers/4d3cf487-70f4-4e1e-9ff1-b2bfce8d9f04/orders/074bd849-9106-405c-9923-fa061839d487",
      "method": "GET",
      "headers": []
    }
  },
  "attributes": {
    "etag": "eyJpZCI6IjA3NGJkODQ5LTkxMDYtNDA1Yy05OTIzLWZhMDYxODM5ZDQ4NyIsInZlcnNpb24iOiJF9",
    "objectType": "Order"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- View a list of [Azure CSP integration scenarios](#).

Create an order for a customer of an indirect reseller

9/19/2017 • 3 min to read • [Edit Online](#)

Learn how to create an order for a customer of an indirect reseller. You can use one of the following options:

- C#
- REST API

C#

To create an order for a customer of an indirect reseller, do the following:

1. Get a collection of the indirect resellers that have a relationship with the signed-in partner.
2. Set a local variable to the item in the collection that matches the indirect reseller ID. This setting helps you access the reseller's **MpnId** property when you create the order.
3. To record the customer, instantiate an **Order** object, and then set the **ReferenceCustomerId** property to the customer identifier.
4. Create a list of **OrderLineItem** objects, and then assign the list to the order's **LineItems** property. Each order line item contains the purchase information for one offer. Be sure to populate the **PartnerIdOnRecord** property in each line item with the MPN ID of the indirect reseller. You must have at least one order line item.
5. To identify the customer, obtain an interface to order operations by calling the **IAggregatePartner.Customers.ById** method with the customer ID, and then retrieve the interface from the **Orders** property.
6. To create the order, call the **Create** or **CreateAsync** method.

```

// IAggregatePartner partnerOperations;
// string customerId;
// string offerId;
// string indirectResellerId;

// Get the indirect resellers with a relationship to the signed-in partner.
var indirectResellers = partnerOperations.Relationships.Get(PartnerRelationshipType.IsIndirectCloudSolutionProviderOf);

// Find the matching reseller in the collection.
var selectedIndirectReseller = (indirectResellers != null && indirectResellers.Items.Any()) ?
    indirectResellers.Items.FirstOrDefault(reseller => reseller.Id.Equals(indirectResellerId, StringComparison.OrdinalIgnoreCase)) :
    null;

// Prepare the order and populate the PartnerIdOnRecord with the reseller's Microsoft Partner Network ID.
var order = new Order()
{
    ReferenceCustomerId = customerId,
    LineItems = new List<OrderLineItem>()
    {
        new OrderLineItem()
        {
            OfferId = offerId,
            FriendlyName = "New offer purchase.",
            Quantity = 5,
            PartnerIdOnRecord = selectedIndirectReseller != null ? selectedIndirectReseller.MpnId : null
        }
    }
};

// Place the order.
var createdOrder = partnerOperations.Customers.ById(customerId).Orders.Create(order);

```

REST API

Request

METHOD	REQUEST URI
POST	{baseUrl}/v1/customers/{customer-id}/orders HTTP/1.1

URI parameters

To identify the customer, use the following *required* parameter:

NAME	TYPE	DESCRIPTION
customer-id	string	A GUID-formatted string that identifies the customer.

Request body

The request body contains the following **Order** properties:

NAME	TYPE	REQUIRED?	DESCRIPTION
id	string	N	The order identifier that is supplied when the order is created successfully.
referenceCustomerId	string	Y	The customer identifier.

NAME	TYPE	REQUIRED?	DESCRIPTION
billingCycle	string	N	The frequency with which the partner is billed for this order. The default setting is <i>monthly</i> , and it is applied when the order is created successfully. The supported values are the member names found in BillingCycleType . Note that annual billing is not yet supported, but support is coming soon.
lineItems	array of objects	Y	An array of OrderLineItem resources.
creationDate	string	N	The date the order was created, in date-time format. The setting is applied when the order is created successfully.
attributes	object	N	Contains the Order object type.

The request body contains the following **OrderLineItem** object properties:

NAME	TYPE	REQUIRED?	DESCRIPTION
lineItemNumber	int	Y	Each line item in the collection gets a unique line number, counting up from 0 to count-1.
offerId	string	Y	The offer identifier.
subscriptionId	string	N	The subscription identifier.
parentSubscriptionId	string	N	The ID of the parent subscription in an add-on offer. Applies to PATCH only.
friendlyName	string	N	The friendly name of the subscription defined by the partner to help disambiguate.
quantity	int	Y*	The number of licenses for a license-based subscription. *Required only for license-based subscriptions.

NAME	TYPE	REQUIRED?	DESCRIPTION
partnerIdOnRecord	string	N	When an indirect provider places an order on behalf of an indirect reseller, populate this field with the MPN ID of the <i>indirect reseller only</i> , never the ID of the indirect provider. This approach ensures proper accounting for incentives.
attributes	object	N	Contains the OrderLineItem object type.

Request example

```

POST https://api.partnercenter.microsoft.com/v1/customers/<customer ID>/orders HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: 02109f46-3ff2-4be4-9f37-b2eb6d58d542
MS-CorrelationId: 85195ae6-3de5-4978-abd4-7be2fbfe4c84
X-Locale: en-US
Content-Type: application/json
Host: api.partnercenter.microsoft.com
Content-Length: 410
Expect: 100-continue

```

```

{
  "Id": null,
  "ReferenceCustomerId": "<customer ID>",
  "BillingCycle": "unknown",
  "LineItems": [{
    "LineItemNumber": 0,
    "OfferId": "<offer ID>",
    "SubscriptionId": null,
    "ParentSubscriptionId": null,
    "FriendlyName": "New offer purchase.",
    "Quantity": 5,
    "PartnerIdOnRecord": "<partner ID>",
    "Attributes": {
      "ObjectType": "OrderLineItem"
    }
  }
],
  "CreationDate": null,
  "Attributes": {
    "ObjectType": "Order"
  }
}

```

Response

If the request is successful, the response body contains the populated **Order** resource.

HTTP/1.1 201 Created
Content-Length: 831
Content-Type: application/json; charset=utf-8
MS-CorrelationId: 85195ae6-3de5-4978-abd4-7be2fbfe4c84
MS-RequestId: 02109f46-3ff2-4be4-9f37-b2eb6d58d542
MS-CV: Nd3Oum/L5EywKQK.0
MS-ServerId: <server ID>
Date: Mon, 10 Apr 2017 23:02:24 GMT

```
{
  "id": "<order ID>",
  "referenceCustomerId": "<customer ID>",
  "billingCycle": "monthly",
  "lineItems": [ {
    "lineItemNumber": 0,
    "offerId": "<offer ID>",
    "subscriptionId": "<subscription ID>",
    "friendlyName": "New offer purchase.",
    "quantity": 5,
    "partnerIdOnRecord": "<partner ID>",
    "links": {
      "subscription": {
        "uri": "/customers/<customer ID>/subscriptions/<subscription ID>",
        "method": "GET",
        "headers": []
      }
    }
  }
],
  "creationDate": "2017-04-10T16:02:25.983-07:00",
  "links": {
    "self": {
      "uri": "/customers/<customer ID>/orders/<subscription ID>",
      "method": "GET",
      "headers": []
    }
  },
  "attributes": {
    "etag": "eyJpZCI6IjNIZGRjYWMyLTZyYjltNGM0MC1iMG12LWY0N2UxODMwMTQ5MjlsInZlcnNpb24iOjF9",
    "objectType": "Order"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- View a list of [Azure CSP integration scenarios](#).

Get an offer by ID

9/19/2017 • 1 min to read • [Edit Online](#)

This guide covers how to retrieve an **Offer** resource that matches the provided offer ID.

PowerShell

To retrieve a specific offer from Microsoft Partner Center by using PowerShell, enter the following command:

```
Get-PCOffer -countryid '<country two digits id>' -offerid '<offer id GUID>'
```

C#

To find a specific offer by ID, do the following:

1. Use your **IAggregatePartner.Offers** collection, establish the country with a call to **ByCountry()**, and then call the **ById()** method.
2. Call the **Get()** or **Get Async()** method.

```
// IAggregatePartner partnerOperations;  
  
//retrieve the offer ID  
var offers = state[FeatureSamples.Application.OffersKey] as List<Offer>;  
  
var offerDetails = partnerOperations.Offers.ByCountry("US").ById(offers[0].Id).Get();
```

REST API

Request

Request syntax

METHOD	REQUEST URI
GET	<i>{baseUrl}/v1/offers/{offer-id}?country={country-id}</i> HTTP/1.1

URI parameter

For the URI request, use the following *required* parameters:

NAME	TYPE	DESCRIPTION
offer-id	guid	A GUID that corresponds to the offer.
country-id	string	The country/region ID.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/offers/<offer-id>?country=<country-id> HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: ac943950-ba3d-47a0-bd2a-c5617a7fefe8
MS-CorrelationId: 7c1f6619-c176-4040-a88f-2c71f3ba4533
X-Locale: <locale-id>
Connection: Keep-Alive
```

Response

If the request is successful, this method returns an **Offer** resource in the response body.

Response example

```
HTTP/1.1 200 OK
Content-Length: 1918
Content-Type: application/json
MS-CorrelationId: 7c1f6619-c176-4040-a88f-2c71f3ba4533
MS-RequestId: ac943950-ba3d-47a0-bd2a-c5617a7fefe8
Date: Mon, 23 Nov 2015 23:13:01 GMT

{
  "id": "031C9E47-4802-4248-838E-778FB1D2CC05",
  "name": "Office 365 Business Premium",
  "description": "For businesses with 1 to 300 users that need the latest desktop version of Office,
    plus anywhere access to email, filesharing, and online conferencing.",
  "minimumQuantity": 1,
  "maximumQuantity": 300,
  "rank": 56,
  "uri": "/3c95518e-8c37-41e3-9627-0ca339200f53/Offers/031C9E47-4802-4248-838E-778FB1D2CC05",
  "locale": "en-us",
  "country": "US",
  "category": {
    "id": "SmallBusiness_Key",
    "name": "Small Business",
    "rank": 30,
    "locale": "en-us",
    "country": "US",
    "attributes": {
      "objectType": "OfferCategory"
    }
  },
  "prerequisiteOffers": [],
  "isAddOn": false,
  "isAvailableForPurchase": true,
  "billing": "license",
  "isAutoRenewable": true,
  "product": {
    "id": "f245ecc8-75af-4f8e-b61f-27d8114de5f3",
    "name": "Office 365 Business Premium",
    "unit": "Licenses"
  },
  "unitType": "Licenses",
  "links": {
    "learnMore": {
      "uri": "http://g.microsoftonline.com/0BXPS00en/909",
      "method": "GET",
      "headers": []
    }
  },
  "attributes": {
    "objectType": "Offer"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- View a list of [Azure CSP integration scenarios](#).

Get an order by ID

9/19/2017 • 1 min to read • [Edit Online](#)

This guide covers how to retrieve an **Order** resource that matches the provided customer and order ID.

PowerShell

To retrieve an order, enter the following commands:

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'
Get-PCOrder -tenantid $customer.id -orderid '<order id guid>'
```

C#

To get a customer's order by ID, do the following:

1. Use your **IAggregatePartner.Customers** collection, and call the **ById()** method.
2. Call the **Orders** property, and then call the **ById()** method again.
3. Call the **Get()** or **GetAsync()** method.

```
// IAggregatePartner partnerOperations;
// string selectedCustomerId;
// string selectedOrderId;

var order = partnerOperations.Customers.ById(selectedCustomerId).Orders.ById(selectedOrderId).Get();
```

REST API

Request

Request syntax

METHOD	REQUEST URI
GET	<i>{baseUrl}/v1/customers/{customer-tenant-id}/orders/{id-for-order}</i> HTTP/1.1

URI parameter

To get an order by ID, use the following *required* query parameters:

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	A GUID that corresponds to the customer.
id-for-order	guid	A GUID that corresponds to the order.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers/<customer-tenant-id>/orders/<id-for-order> HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: 0e5fc923-8e3c-4560-9100-ce7283c3e081
MS-CorrelationId: 8a53b025-d5be-4d98-ab20-229d1813de76
Connection: Keep-Alive
```

Response

If the request is successful, this method returns an **Order** resource in the response body.

Response example

```
HTTP/1.1 200 OK
Content-Length: 1252
Content-Type: application/json
MS-CorrelationId: 8a53b025-d5be-4d98-ab20-229d1813de76
MS-RequestId: 0e5fc923-8e3c-4560-9100-ce7283c3e081
Date: Mon, 23 Nov 2015 22:02:59 GMT

{
  "id": "d6595733-265f-4918-a62e-026e64bc8384",
  "referenceCustomerId": "<customer-tenant-id>",
  "lineItems": [ {
    "lineItemNumber": 0,
    "offerId": "031C9E47-4802-4248-838E-778FB1D2CC05",
    "friendlyName": "nickname",
    "quantity": 1,
    "links": {
      "subscription": {
        "uri": "/subscriptions?key=<key>",
        "method": "GET",
        "headers": []
      }
    }
  } ],
  "status": "none",
  "creationDate": "2015-10-08T10:42:36.54-07:00",
  "attributes": {
    "etag": "<etag>",
    "objectType": "Order"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- View a list of [Azure CSP integration scenarios](#).

Get subscription provisioning status

9/19/2017 • 1 min to read • [Edit Online](#)

This guide covers how to retrieve the subscription provisioning status for a customer subscription by using the REST API. The status is updated every 15 minutes.

Request

Request syntax

METHOD	REQUEST URI
GET	<code>{baseUrl}/v1/customers/{customer-id}/subscriptions/{subscription-id}/provisioningstatus</code> HTTP/1.1

URI parameters

To identify the customer and subscription, use the following *required* path parameters:

NAME	TYPE	SUBSCRIPTION
customer-id	guid	A GUID-formatted string that identifies the customer.
subscription-id	guid	A GUID-formatted string that identifies the subscription.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers/0c39d6d5-c70d-4c55-bc02-f620844f3fd1/subscriptions/34828C05-C16C-4D6F-9CFC-4D2650EF19A1/provisioningstatus HTTP/1.1
Accept: application/json, text/plain, */*
Authorization: Bearer <token>
MS-RequestId: d0e38dfd-a2c5-4a14-ac06-12d30f0ec54e
MS-CorrelationId: e937630b-8341-4d70-8f73-450d32ee0189
X-Locale: en-US
Host: api.partnercenter.microsoft.com
```

Response

If the request is successful, the response body contains a **SubscriptionProvisioningStatus** resource.

Response example

```
HTTP/1.1 200 OK
Content-Length: 177
Content-Type: application/json; charset=utf-8
MS-CorrelationId: e937630b-8341-4d70-8f73-450d32ee0189
MS-RequestId: d0e38dfd-a2c5-4a14-ac06-12d30f0ec54e
MS-CV: InsWEQre402koceL.0
MS-ServerId: 030020344
Date: Thu, 20 Apr 2017 19:23:39 GMT
```

```
{
  "skuId": "6FD2C87F-B296-42F0-B197-1E91E994B900",
  "status": "success",
  "quantity": 5,
  "endDate": "2018-05-10T00:00:00Z",
  "attributes": {
    "objectType": "SubscriptionProvisioningStatus"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- View a list of [Azure CSP integration scenarios](#).

Get a list of subscriptions by order for Azure CSP integration

9/19/2017 • 1 min to read • [Edit Online](#)

Learn how to retrieve a collection of Azure Cloud Solution Provider (Azure CSP) subscription resources that correspond to an order. To retrieve a resource, you can use PowerShell, C#, or REST API commands.

PowerShell guide

To retrieve all of the customer subscriptions from an order, enter the following command into PowerShell:

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'
Get-PCSubscription -tenantid $customer.id -orderid '<order id GUID>'
```

C# SDK guide

To get a list of subscriptions by order, use your **IAggregatePartner.Customers** collection and call the **ById()** method. Then call the **Subscriptions** property, followed by the **ByOrderId()** method. Finish by calling **Get()** or **GetAsync()**.

```
// IAggregatePartner partnerOperations;
// var selectedCustomerId as string;
// string orderID;

ResourceCollection<Subscription> customerSubscriptions =
partnerOperations.Customers.ById(selectedCustomerId).Subscriptions.ByOrder(orderID).Get();
```

REST API guide

Request

Request syntax

METHOD	REQUEST URI
GET	<i>{baseUrl}/v1/customers/{customer-tenant-id}/subscriptions?order_id={id-for-order}</i> HTTP/1.1

URI parameter

These are the *required* query parameters to get all the subscriptions.

NAME	TYPE	DESCRIPTION
<i>customer-tenant-id</i>	guid	A GUID that corresponds to the customer.
<i>id-for-order</i>	guid	A GUID that corresponds to the order.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers/{customer-tenant-id}/subscriptions?order_id={id-for-order} HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: 16fee928-dc2c-412f-adbb-871f68babf16
MS-CorrelationId: c49004b1-224f-4d86-a607-6c8bcc52cfdd
Connection: Keep-Alive
```

Response

If the request is successful, this method returns a collection of **Subscription** resources in the response body.

Response example

```
HTTP/1.1 200 OK
Content-Length: 73754
Content-Type: application/json
MS-CorrelationId: c49004b1-224f-4d86-a607-6c8bcc52cfdd
MS-RequestId: 16fee928-dc2c-412f-adbb-871f68babf16
Date: Wed, 25 Nov 2015 05:50:45 GMT

{
  "totalCount": 37,
  "items": [ {
    "id": "83ef9d05-4169-4ef9-9657-0e86b1eab1de",
    "friendlyName": "Myofferpurchase",
    "quantity": 1,
    "unitType": "none",
    "creationDate": "2015-11-25T06: 41: 12Z",
    "effectiveStartDate": "2015-11-24T08: 00: 00Z",
    "commitmentEndDate": "2016-12-12T08: 00: 00Z",
    "status": "active",
    "autoRenewEnabled": false,
    "billingType": "none",
    "contractType": "subscription",
    "links": {
      "offer": {
        "uri": "/v1/offers/0CCA44D6-68E9-4762-94EE-31ECE98783B9",
        "method": "GET",
        "headers": []
      },
      "entitlement": {
        "uri": "/entitlements?key=<key>",
        "method": "GET",
        "headers": []
      },
      "self": {
        "uri": "/subscriptions?key=<key>",
        "method": "GET",
        "headers": []
      }
    },
    "orderId": "{id-for-order}",
    "attributes": {
      "etag": "<etag>",
      "objectType": "Subscription"
    }
  } ],
  "attributes": {
    "objectType": "Collection"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Suspend or reactivate a customer subscription for Azure CSP integration

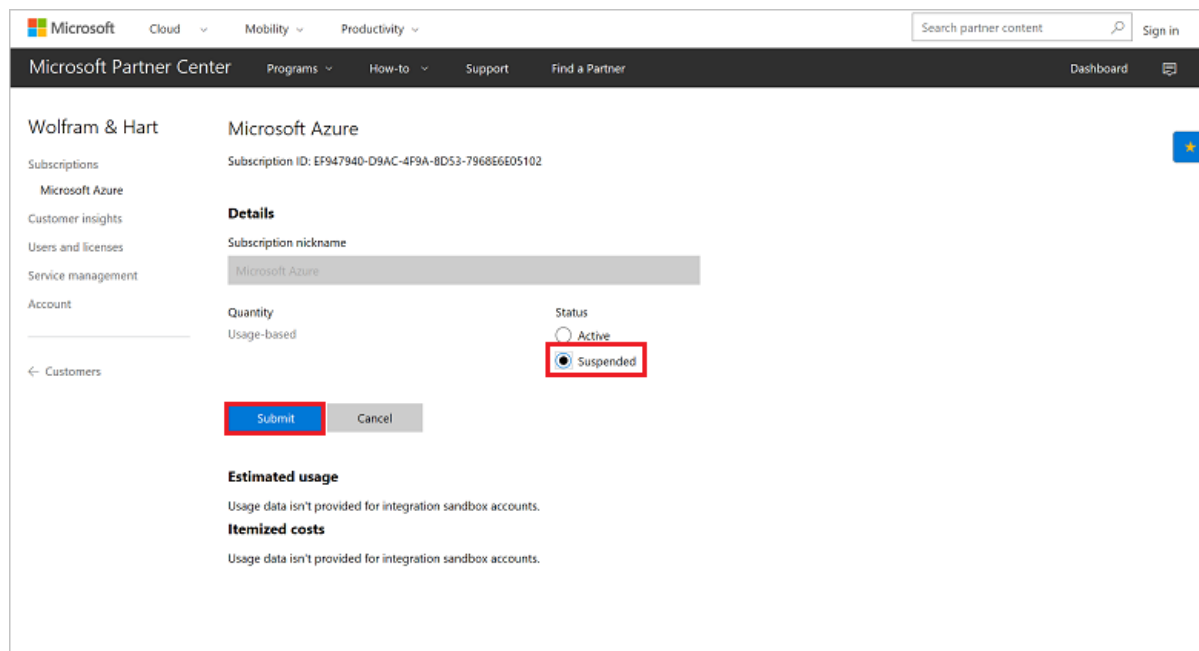
9/19/2017 • 4 min to read • [Edit Online](#)

You can suspend an Azure Cloud Solution Provider (Azure CSP) subscription due to overuse, possibility of fraud or misuse, or lack of payment. With Microsoft Partner Center, you can easily suspend and reactivate subscriptions. Use Web UI, PowerShell, C#, or REST API to suspend or reactivate a subscription.

Web UI

Suspend the subscription

1. From the Partner Center **Dashboard** page, go to **Customers** > select the specific customer > **Subscriptions** > select the specific subscription.
2. For **Status**, select **Suspended**, and then select **Submit**.



Reactivate the subscription

1. From the Partner Center **Dashboard** page, go to **Customers** > select the specific customer > **Subscriptions** > select the specific subscription.
2. For **Status**, select **Active**, and then select **Submit**.

ACME Corporation

Subscriptions

Microsoft Azure

Customer insights

Users and licenses

Service management

Account

← Customers

Microsoft Azure

Subscription ID: 5E04BAA7-2C68-4B53-94DD-601404FD3A56

Details

Subscription nickname

Microsoft Azure

Quantity

Usage-based

Status

☒ Active

☐ Suspended

Submit Cancel

Estimated usage

Usage data isn't provided for integration sandbox accounts.

Itemized costs

Usage data isn't provided for integration sandbox accounts.

PowerShell

Suspend the subscription

To suspend a subscription by using PowerShell, enter the following command:

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'
$subscription = Get-PCSubscription -tenantid $customer.id -subscriptionid '<subscription id GUID>'
$subscription | Set-PCSubscription -tenantid $customer.id -status suspended
```

Reactivate the subscription

To reactivate a suspended subscription in PowerShell, enter the following command:

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'
$subscription = Get-PCSubscription -tenantid $customer.id -subscriptionid '<subscription id GUID>'
$subscription | Set-PCSubscription -tenantid $customer.id -status active
```

C#

Suspend the subscription

1. [Get the subscription](#), and then change the subscription's **Status** property.
2. After the change is made, use your **IAggregatePartner.Customers** collection to call the **ById()** method.
3. Call the **Subscriptions** property, followed by the **ById()** method.
4. Call the **Patch()** method.

```
// IAggregatePartner partnerOperations;
// var selectedCustomerId as string;
// Subscription selectedSubscription;

updatedSubscription = partnerOperations.Customers.ById(selectedCustomerId).Subscriptions.ById(selectedSubscription.Id).Patch(
    new Subscription()
    {
        Status = SubscriptionStatus.Suspended
    });
```

Reactivate the subscription

1. [Get the subscription](#), and then change the subscription's **Status** property. For information on **Status** codes, see

[SubscriptionStatus enumeration](#).

2. After the change is made, use your **IPartner.Customers** collection to call the **ById()** method.
3. Call the **Subscriptions** property, followed by the **ById()** method.
4. Call the **Update()** method.

```
// IPartner partnerOperations;  
// var selectedCustomer as Customer;  
// var selectedSubscription as Subscription;  
  
updatedSubscription = partnerOperations.Customers.ById(selectedCustomerId).Subscriptions.ById(selectedSubscription.Id).Patch(  
    new Subscription()  
    {  
        Status = SubscriptionStatus.Active  
    });
```

REST API

Suspend the subscription

Request syntax

METHOD	REQUEST URI
PATCH	<i>{baseUrl}/v1/customers/{customer-tenant-id}/subscriptions/{id-for-subscription}</i> HTTP/1.1

URI parameter

These are the *required* query parameters that are used to suspend a subscription.

NAME	TYPE	DESCRIPTION
<i>customer-tenant-id</i>	guid	A GUID that corresponds to the customer.
<i>id-for-subscription</i>	guid	A GUID that corresponds to the subscription.

Request body

A full **Subscription** resource is required in the request body. Ensure that the **Status** property is updated.

Request example

```
PATCH https://api.partnercenter.microsoft.com/v1/customers/<customer-tenant-id>/subscriptions/<id-for-subscription> HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: ca7c39f7-1a80-43bc-90d8-ee7d1cad3831
MS-CorrelationId: ec8f62e5-1d92-47e9-8d5d-1924af105f2c
If-Match: <etag>
Content-Type: application/json
Content-Length: 1029
Expect: 100-continue
Connection: Keep-Alive
```

```
{
  "Id": "83ef9d05-4169-4ef9-9657-0e86b1eab1de",
  "FriendlyName": "nickname",
  "Quantity": 2,
  "UnitType": "none",
  "ParentSubscriptionId": null,
  "CreationDate": "2015-11-25T06:41:12Z",
  "EffectiveStartDate": "2015-11-24T08:00:00Z",
  "CommitmentEndDate": "2016-12-12T08:00:00Z",
  "Status": "suspended",
  "AutoRenewEnabled": false,
  "BillingType": "none",
  "PartnerId": null,
  "ContractType": "subscription",
  "OrderId": "6183db3d-6318-4e52-877e-25806e4971be",
  "Attributes": {
    "Etag": "<etag>",
    "ObjectType": "Subscription"
  }
}
```

If the response is successful, this method returns an updated **Subscription** resource in the response body.

Response example

PATCH https://api.partnercenter.microsoft.com/v1/customers/{customer-tenant-id}/subscriptions/{subscriptionID} HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-Contract-Version: v1
MS-RequestId: ca7c39f7-1a80-43bc-90d8-ee7d1cad3831
MS-CorrelationId: ec8f62e5-1d92-47e9-8d5d-1924af105f2c
Content-Type: application/json
Content-Length: 1029
Expect: 100-continue
Connection: Keep-Alive

```
{
  "Id": "83ef9d05-4169-4ef9-9657-0e86b1eab1de",
  "FriendlyName": "nickname",
  "Quantity": 2,
  "UnitType": "none",
  "ParentSubscriptionId": null,
  "CreationDate": "2015-11-25T06:41:12Z",
  "EffectiveStartDate": "2015-11-24T08:00:00Z",
  "CommitmentEndDate": "2016-12-12T08:00:00Z",
  "Status": "suspended",
  "AutoRenewEnabled": false,
  "BillingType": "none",
  "PartnerId": null,
  "ContractType": "subscription",
  "Links": {
    "Offer": {
      "Uri": "/v1/offers/0CCA44D6-68E9-4762-94EE-31ECE98783B9",
      "Method": "GET",
      "Headers": []
    },
    "Entitlement": {
      "Uri": "/entitlements?key=<key>",
      "Method": "GET",
      "Headers": []
    },
    "Self": {
      "Uri": "/subscriptions?key=<key>",
      "Method": "GET",
      "Headers": []
    }
  },
  "OrderId": "6183db3d-6318-4e52-877e-25806e4971be",
  "Attributes": {
    "Etag": "<etag>",
    "ObjectType": "Subscription"
  }
}
```

Reactivate the subscription

Request syntax

METHOD	REQUEST URI
PATCH	{baseUrl}/v1/customers/{customer-tenant-id}/subscriptions/{id-for-subscription} HTTP/1.1

URI parameter

These are the *required* query parameters that are used to reactivate a subscription.

NAME	TYPE	DESCRIPTION
<i>customer-tenant-id</i>	guid	A GUID that corresponds to the customer.
<i>id-for-subscription</i>	guid	A GUID that corresponds to the subscription.

Request body

A full **Subscription** resource is required in the request body. Ensure that the **Status** property is updated.

Request example

```
PATCH https://api.partnercenter.microsoft.com/v1/customers/<customer-tenant-id>/subscriptions/<id-for-subscription> HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: ca7c39f7-1a80-43bc-90d8-ee7d1cad3831
MS-CorrelationId: ec8f62e5-1d92-47e9-8d5d-1924af105f2c
Content-Type: application/json
Content-Length: 1029
Expect: 100-continue
Connection: Keep-Alive

{
  "Id": "83ef9d05-4169-4ef9-9657-0e86b1eab1de",
  "FriendlyName": "nickname",
  "Quantity": 2,
  "UnitType": "none",
  "ParentSubscriptionId": null,
  "CreationDate": "2015-11-25T06:41:12Z",
  "EffectiveStartDate": "2015-11-24T08:00:00Z",
  "CommitmentEndDate": "2016-12-12T08:00:00Z",
  "Status": "active",
  "AutoRenewEnabled": false,
  "BillingType": "none",
  "PartnerId": null,
  "ContractType": "subscription",
  "OrderId": "6183db3d-6318-4e52-877e-25806e4971be",
  "Attributes": {
    "Etag": "<etag>",
    "ObjectType": "Subscription"
  }
}
```

If the response is successful, this method returns updated **Subscription** resource properties in the response body.

Example response

PATCH https://api.partnercenter.microsoft.com/v1/customers/<customer-tenant-id>/subscriptions/<subscriptionID> HTTP/1.1

Authorization: Bearer <token>

Accept: application/json

MS-Contract-Version: v1

MS-RequestId: ca7c39f7-1a80-43bc-90d8-ee7d1cad3831

MS-CorrelationId: ec8f62e5-1d92-47e9-8d5d-1924af105f2c

Content-Type: application/json

Content-Length: 1029

Expect: 100-continue

Connection: Keep-Alive

```
{
  "Id": "83ef9d05-4169-4ef9-9657-0e86b1eab1de",
  "FriendlyName": "nickname",
  "Quantity": 2,
  "UnitType": "none",
  "ParentSubscriptionId": null,
  "CreationDate": "2015-11-25T06:41:12Z",
  "EffectiveStartDate": "2015-11-24T08:00:00Z",
  "CommitmentEndDate": "2016-12-12T08:00:00Z",
  "Status": "active",
  "AutoRenewEnabled": false,
  "BillingType": "none",
  "PartnerId": null,
  "ContractType": "subscription",
  "Links": {
    "Offer": {
      "Uri": "/v1/offers/0CCA44D6-68E9-4762-94EE-31ECE98783B9",
      "Method": "GET",
      "Headers": []
    },
    "Entitlement": {
      "Uri": "/entitlements?key=<key>",
      "Method": "GET",
      "Headers": []
    },
    "Self": {
      "Uri": "/subscriptions?key=<key>",
      "Method": "GET",
      "Headers": []
    }
  },
  "OrderId": "6183db3d-6318-4e52-877e-25806e4971be",
  "Attributes": {
    "Etag": "<etag>",
    "ObjectType": "Subscription"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

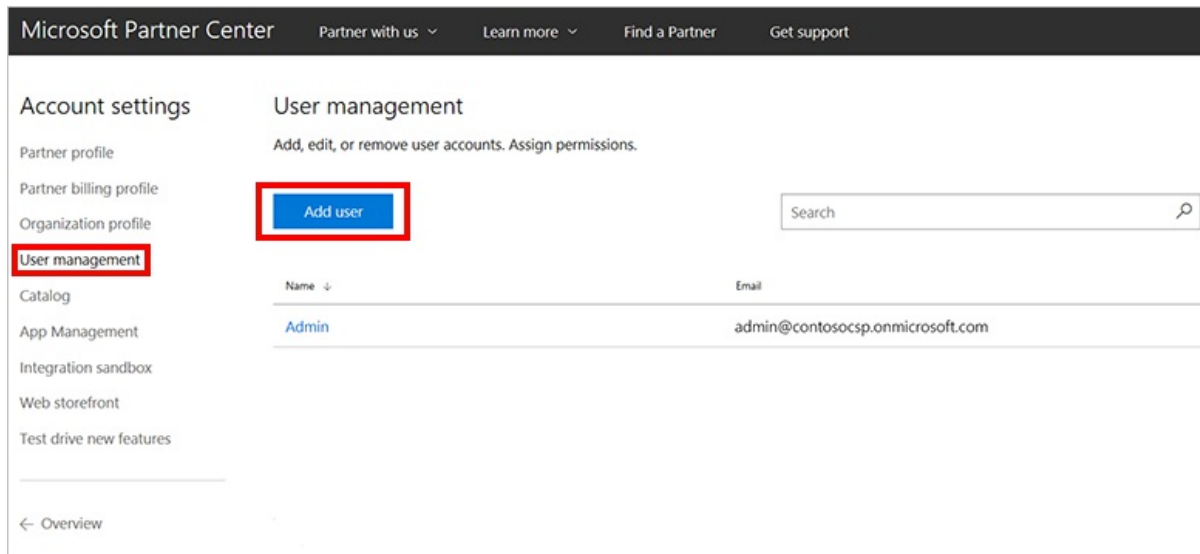
Create a user account in Partner Center

9/19/2017 • 1 min to read • [Edit Online](#)

Learn how to create user accounts for employees who need to access Microsoft Partner Center. The following tasks must be completed by an admin who has user management admin rights.

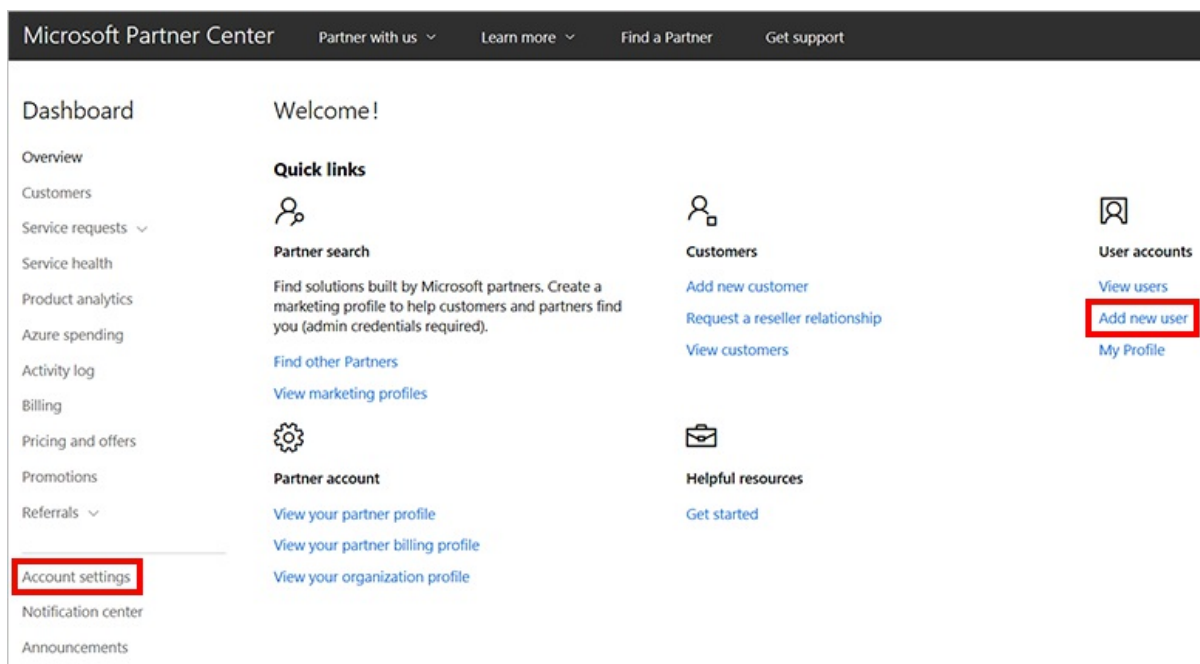
Add a new user

1. In Partner Center, on the **Dashboard** menu, select **Account settings** > **User management** > **Add User**.



TIP

For another option, in the **Quick links** section of the dashboard, under **User accounts**, select **Add new user**.



2. Enter the user's full name and a unique email address.
3. Select the type of agent and the admin type. Because Partner Center access is role-based, you can assign

permissions to set the user's view to show only the features the user needs to complete specific tasks. For more information about what each role can do, see [Partner Center user roles](#).

4. To create the user account, select **Add**. On the next page, confirm the user's details.

Microsoft Partner Center Partner with us Learn more Find a Partner Get support

Account settings Add user

Partner profile
Partner billing profile
Organization profile
User management
Add user
Catalog
App Management
Integration sandbox
Web storefront
Test drive new features

← Overview

Name *
Lucy Allen

Email *
lucy.allen @contosocsp.onmicrosoft.com

Assists your customers as [Learn more](#)

☐ Admin agent
Has access to all Partner Center features

☒ Sales agent
Adds customers and manages their subscriptions

☐ Helpdesk agent
Helps resolve customer issues with their services

☐ Not an agent

Manages your company as [Learn more](#)

☐ Global admin
Has full access to all administrative and Partner Center features

☒ Billing admin
Can view invoices and billing information

☐ User management admin
Creates agents and resets their passwords

☐ Not an admin

Add Cancel

IMPORTANT

Take note of the new user's sign-in information that is displayed on the add user page. Copy it and send it to the user. You won't have access to this information again later.

Microsoft Partner Center Partner with us Learn more Find a Partner Get support

Account settings Lucy Allen has been added

Partner profile
Partner billing profile
Organization profile
User management
Add user
Catalog
App Management
Integration sandbox
Web storefront
Test drive new features

← Overview

Permissions:

Assists your customers as Sales agent

Manages your company as Billing admin

Sign in info:

Username
lucy.allen@contosocsp.onmicrosoft.com

Temporary password
0SJ-14R9i

Please copy this information to give to Lucy Allen.
NOTE: After you leave this page, you can't view this temporary password again.
It takes a few minutes to complete these updates.

Done Add another user

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Restore a suspended Microsoft Partner Center account

9/19/2017 • 1 min to read • [Edit Online](#)

Learn how to restore a Microsoft Partner Center account that has been suspended.

NOTE

This task is different from [restoring a deleted customer account](#).

Why Partner Center accounts are suspended

Microsoft might suspend a Partner Center account for one of the following reasons:

- The current billing amount is more than 60 days past due.
- A significant amount of abuse is detected.
- Issues that affect indirect provider operations are negatively affecting your account (indirect resellers only).

What to do if your account is suspended

- If you know that your account billing is current, [submit a support request](#) to find out why your account was suspended.
- If the account is past due, follow the instructions on your bill to arrange a payment. Then, [submit a support request](#) to have the account reinstated.
- If you are not sure whether your account billing is current, to check your bill, on the **Dashboard** menu, select **Billing**. This page shows billing history and trends, the balance due for the current month, and the most recent payment. It also has links to invoices and reconciliation files.

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Update partner profiles in your CSP account

9/19/2017 • 5 min to read • [Edit Online](#)

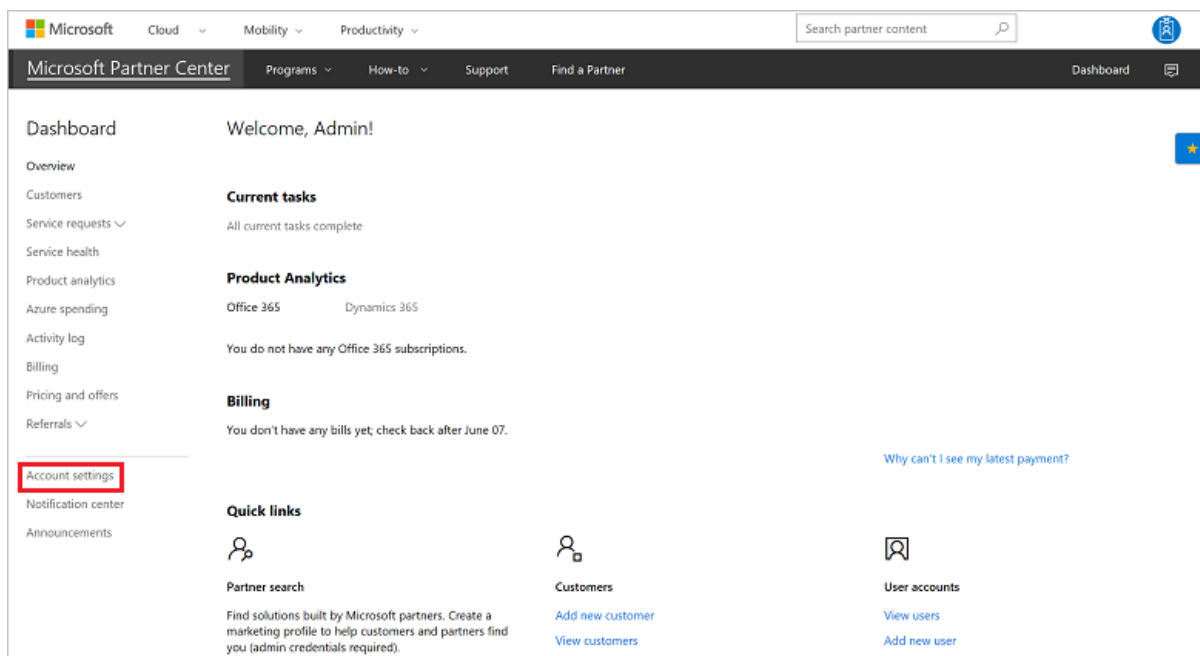
Your Azure Cloud Solution Provider (CSP) account in Microsoft Partner Center has numerous profiles associated with it, including a business profile, billing profile, and support profile. This guide covers how to change each of them by using the web UI, C#, PowerShell, or the REST API.

Business profile

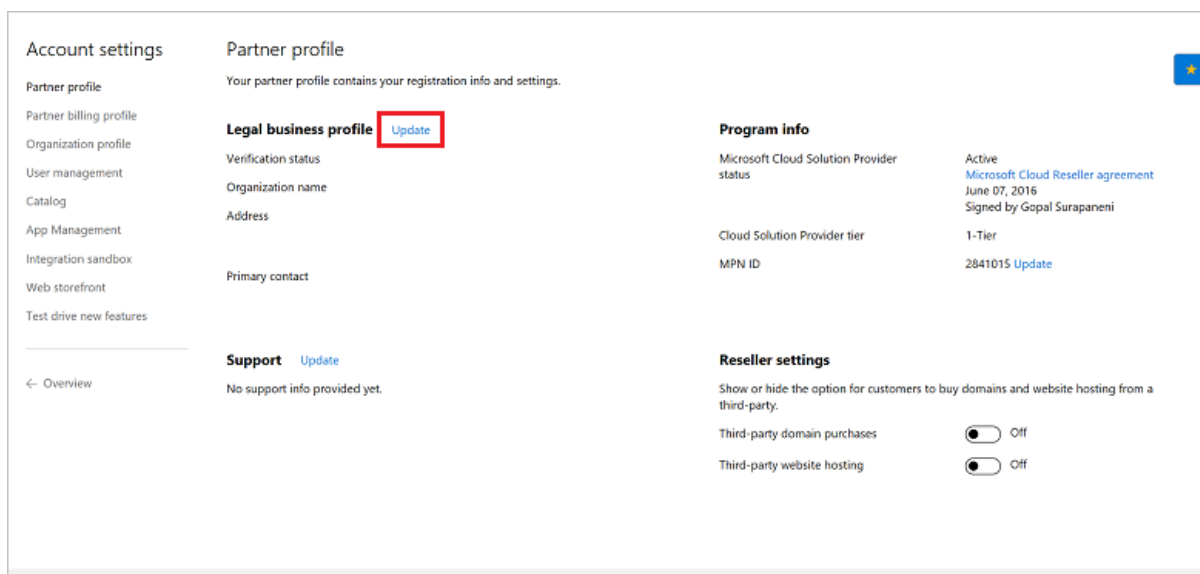
The partner business profile contains legal business-contact information about your business. For this reason, if you change this information in Partner Center, Microsoft might contact your business for verification. This guide covers how to update your partner business profile.

Web UI

1. In Partner Center, go to the **Dashboard** menu > **Account settings** > **Partner profile**.



2. Next to **Legal business profile**, select **Update**.



3. The legal business profile contains the street address of your business, as well as a primary contact's name, email address, and phone number. Ensure that the information is up to date, and then select **Submit**.

The screenshot shows the Microsoft Partner Center interface. The top navigation bar includes Microsoft, Cloud, Mobility, Productivity, and a search bar. The main header is 'Microsoft Partner Center' with links to Programs, How-to, Support, and Find a Partner. The left sidebar lists 'Account settings' and 'Partner profile'. The main content area is titled 'Legal business profile' and contains a form with the following fields: Country/region (United States), Organization name (SYSGAIN INC.), Address line 1 (123 Fake Street), Address line 2, City (Bellevue), State/Province (Washington), ZIP/Postal code (98007), Primary contact first name (Jim), Primary contact last name (Johnson), Primary contact email (jj@outlook.com), and Phone number (5555555555). A red box highlights the 'Submit' button.

PowerShell

To update your legal business profile by using PowerShell, run the following command:

```
Set-PCLegalBusinessProfile -addressLine1 '<address 1>' -AddressLine2 '<address 2>' -primarycontactfirstname '<first name>' -primarycontactlastname '<last name>' -primarycontactphonenumber '<phone number>' -primarycontactemail '<email>'
```

C#

To update the partner legal business profile, do the following:

1. Instantiate a **LegalBusinessProfile** object, populate it with the existing profile, and then update its properties. The code sample in this section illustrates how to change the address and primary contact phone numbers.
2. Get an interface to the partner profile operations collection from the **IAggregatePartner.Profiles** property.
3. Retrieve the value of the **LegalBusinessProfile** property to get an interface to legal business profile operations.
4. To update the profile, call the **Update** or **UpdateAsync** method with the changed object.

```
// IAggregatePartner partnerOperations;  
  
var legalBusinessProfile = partnerOperations.Profiles.LegalBusinessProfile.Get();  
  
// Change the address and primary contact phone number.  
legalBusinessProfile.Address.PhoneNumber = "4255550110";  
legalBusinessProfile.PrimaryContact.PhoneNumber = "4255550110";  
  
// Apply changes to the profile.  
var updatedLegalBusinessProfile = partnerOperations.Profiles.LegalBusinessProfile.Update(legalBusinessProfile);
```

REST API

Request syntax

METHOD	REQUEST URI
PUT	<i>{baseUrl}</i> /v1/profiles/legalbusiness HTTP/1.1

Request body

The legal business-profile resource.

Request example

PUT https://api.partnercenter.microsoft.com/v1/profiles/legalbusiness HTTP/1.1

Authorization: Bearer <token>

Accept: application/json

MS-RequestId: 4549ac0c-0f1d-4d8f-b02f-6d36fadcccee

MS-CorrelationId: aa2a0d8c-7a41-470f-97ae-b82e6c338ee4

X-Locale: en-US

Content-Type: application/json

Host: api.partnercenter.microsoft.com

Content-Length: 806

Expect: 100-continue

```
{
  "CompanyName": "Lucerne Publishing",
  "Address": {
    "Country": "US",
    "Region": null,
    "City": "Redmond",
    "State": "WA",
    "AddressLine1": "123 Main Street",
    "AddressLine2": "",
    "PostalCode": "98052",
    "FirstName": "Gena",
    "LastName": "Soto",
    "PhoneNumber": "4255550110"
  },
  "PrimaryContact": {
    "FirstName": "Gena",
    "LastName": "Soto",
    "Email": "gena@lucemepublishing.com",
    "PhoneNumber": "4255550110"
  },
  "CompanyApproverAddress": {
    "Country": "US",
    "Region": null,
    "City": "Redmond",
    "State": "WA",
    "AddressLine1": "123 Main Street",
    "AddressLine2": "",
    "PostalCode": "98052",
    "FirstName": null,
    "LastName": null,
    "PhoneNumber": null
  },
  "CompanyApproverEmail": "gena@lucemepublishing.com",
  "VettingStatus": "authorized",
  "VettingSubStatus": "none",
  "Links": {
    "Self": {
      "Uri": "/profiles/legalbusiness",
      "Method": "GET",
      "Headers": []
    }
  },
  "Attributes": {
    "ObjectType": "LegalBusinessProfile"
  }
}
```

Response example

If the request is successful, the response body contains the updated **LegalBusinessProfile**.

HTTP/1.1 200 OK
Content-Length: 1157
Content-Type: application/json; charset=utf-8
MS-CorrelationId: aa2a0d8c-7a41-470f-97ae-b82e6c338ee4
MS-RequestId: 4549ac0c-0f1d-4d8f-b02f-6d36fadcccee
MS-CV: KZLU42qJ4EObO75q.0
MS-ServerId: 030020643
Date: Tue, 21 Mar 2017 22:03:15 GMT

```
{
  "companyName": "Lucerne Publishing",
  "address": {
    "country": "US",
    "city": "Redmond",
    "state": "WA",
    "addressLine1": "123 Main Street",
    "addressLine2": "",
    "postalCode": "98052",
    "firstName": "Gena",
    "lastName": "Soto",
    "phoneNumber": "4255550110"
  },
  "primaryContact": {
    "firstName": "Gena",
    "lastName": "Soto",
    "email": "gena@lucernepublishing.com",
    "phoneNumber": "4255550110"
  },
  "companyApproverAddress": {
    "country": "US",
    "city": "Redmond",
    "state": "WA",
    "addressLine1": "123 Main Street",
    "addressLine2": "",
    "postalCode": "98052"
  },
  "companyApproverEmail": "gena@lucernepublishing.com",
  "vettingStatus": "authorized",
  "vettingSubStatus": "none",
  "profileType": "LegalBusinessProfile",
  "links": {
    "self": {
      "uri": "/profiles/legalbusiness",
      "method": "GET",
      "headers": []
    }
  },
  "attributes": {
    "objectType": "LegalBusinessProfile"
  }
}
```

Billing profile

The partner billing profile determines where Microsoft sends CSP invoices. It is important to keep the information up to date so that you receive all billing invoices.

This guide covers how to update your billing profile when necessary.

Web UI

1. In Partner Center, go to the **Dashboard** menu > **Account settings** > **Partner billing profile**.
2. Next to **Bill-to info**, select **Update**.

Microsoft Partner Center

Account settings

Partner profile

Partner billing profile

Organization profile

User management

Catalog

App Management

Integration sandbox

Web storefront

Test drive new features

← Overview

Partner billing profile

Microsoft sends your Cloud Solution Provider invoice to the bill-to address in your billing profile.

Bill-to info **Update**

Bill-to company name

Bill-to address

Bill-to contact

PO number

Financial data

Company Tax ID (not an individual Tax ID)

Billing currency

54-2013417

USD

Industries

Manufacturing & resources

Financial services

Retail

For customers

Small & midsize businesses

Enterprise

Security at home

For partners

Get listed

Microsoft partner resources

Find a solutions provider

For developers

Microsoft Azure

Microsoft Visual Studio

Microsoft Developer Network

Values

Diversity and inclusion

Accessibility

Microsoft in education

Company

Careers

About Microsoft

Company news

- The partner billing profile contains the street address where invoices are to be sent, plus the contact name and phone number. Ensure that this information is present, and then select **Submit**.

Microsoft Partner Center

Account settings

Partner profile

Partner billing profile

Organization profile

User management

Catalog

App Management

Integration sandbox

Web storefront

Test drive new features

← Overview

Bill-to info

Country/region

United States

Bill-to address line 1 *

123 Oak Blvd

City *

Bellevue

Bill-to contact first name *

John

Bill-to contact phone

555-555-5555

Bill-to company name

SYSGAIN INC.

Bill-to address line 2

State/Province *

Washington

ZIP/Postal code *

98007

Bill-to contact last name *

Smith

PO number

Submit Cancel

Industries

For customers

For partners

For developers

Values

Company

PowerShell

To update your Partner Center billing profile by using PowerShell, run the following command:

```
Set-PCBillingProfile -firstname '<first name>' -lastname '<last name>' -phonenumber '<phone number>' -AddressLine1 '<address 1>' -AddressLine2 '<address 2>'
```

C#

To update a partner billing profile, do the following:

- Retrieve the existing profile, and then update it as necessary.
- Use your **IAggregatePartner.Profiles** collection, and call the **BillingProfile** property.
- Call the **Update()** method.

```
// IAggregatePartner partnerOperations;

BillingProfile existingBillingProfile = partnerOperations.Profiles.BillingProfile.Get();

//updating the profile with a randomphone number
existingBillingProfile.PurchaseOrderNumber = new Random().Next(9000, 10000).ToString(CultureInfo.InvariantCulture);

BillingProfile updatedPartnerBillingProfile = partnerOperations.Profiles.BillingProfile.Update(existingBillingProfile);
```

REST API

Request syntax

METHOD	REQUEST URI
PUT	<i>{baseUrl}</i> /v1/profiles/billing HTTP/1.1

Request example

```
PUT https://api.partnercenter.microsoft.com/v1/profiles/billing HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: 9231559e-ed95-41e4-810b-e754ae32dc2f
MS-CorrelationId: cb9f3209-d020-4bf9-871c-e1f1c75348f8
Content-Length: 613
Expect: 100-continue
```

```
{
  "CompanyName":"TEST_TEST_BugBash1",
  "Address":{
    "Country":"US",
    "Region":null,
    "City":"Redmond",
    "State":"WA",
    "AddressLine1":"1 Microsoft Way",
    "AddressLine2":"","PostalCode":"98052",
    "FirstName":null,
    "LastName":null,
    "PhoneNumber":null
  },
  "PrimaryContact":{
    "FirstName":"Test",
    "LastName":"Customer",
    "Email":null,
    "PhoneNumber":""
  },
  "PurchaseOrderNumber":"9888",
  "TaxId":<TaxId>,
  "BillingCurrency":"USD",
  "Links":{
    "Self":{
      "Uri":"/profiles/billing",
      "Method":"GET","Headers":[]
    }
  },
  "Attributes":{
    "Etag":<etag>,
    "ObjectType":"BillingProfile"
  }
}
```

Response example

If the request is successful, this method returns a **BillingProfile** object in the response body.

```
HTTP/1.1 200 OK
Content-Length: 568
Content-Type: application/json; charset=utf-8
MS-CorrelationId: cb9f3209-d020-4bf9-871c-e1f1c75348f8
MS-RequestId: 9231559e-ed95-41e4-810b-e754ae32dc2f
Date: Mon, 21 Mar 2016 05:47:16 GMT
```

```
{
  "CompanyName": "TEST_TEST_BugBash1",
  "Address": {
    "Country": "US",
    "Region": null,
    "City": "Redmond",
    "State": "WA",
    "AddressLine1": "1 Microsoft Way",
    "AddressLine2": "", "PostalCode": "98052",
    "FirstName": null,
    "LastName": null,
    "PhoneNumber": null
  },
  "PrimaryContact": {
    "FirstName": "Test",
    "LastName": "Customer",
    "Email": null,
    "PhoneNumber": ""
  },
  "PurchaseOrderNumber": "9888",
  "TaxId": "<TaxId>",
  "BillingCurrency": "USD",
  "Links": {
    "Self": {
      "Uri": "/profiles/billing",
      "Method": "GET", "Headers": []
    }
  },
  "Attributes": {
    "Etag": "<etag>",
    "ObjectType": "BillingProfile"
  }
}
```

Support profile

If your customers call Microsoft looking for support, Microsoft will point them back to the website, email, and phone contact information that's listed in this support profile.

This guide covers how to update your support profile if necessary.

Web UI

1. In Partner Center, go to the **Dashboard** menu > **Account settings** > **Partner profile**.

Microsoft Partner Center Dashboard

Welcome, Admin!

Current tasks
All current tasks complete

Product Analytics
Office 365 Dynamics 365
You do not have any Office 365 subscriptions.

Billing
You don't have any bills yet; check back after June 07.
[Why can't I see my latest payment?](#)

Quick links

- Partner search**
Find solutions built by Microsoft partners. Create a marketing profile to help customers and partners find you (admin credentials required).
- Customers**
[Add new customer](#)
[View customers](#)
- User accounts**
[View users](#)
[Add new user](#)

2. In the **Support** section, select **Update**.

Account settings

Partner profile
Your partner profile contains your registration info and settings.

Legal business profile [Update](#)

Verification status
Organization name
Address
Primary contact

Program info

Microsoft Cloud Solution Provider status	Active Microsoft Cloud Reseller agreement June 07, 2016 Signed by Gopal Surapaneni
Cloud Solution Provider tier	1-Tier
MPN ID	2841015 Update

Reseller settings
Show or hide the option for customers to buy domains and website hosting from a third-party.

Third-party domain purchases	<input type="checkbox"/> Off
Third-party website hosting	<input type="checkbox"/> Off

3. The support profile contains a support website URL, email address, and phone number. Ensure that the information is up to date, and then select **Submit**.

PowerShell

To update your support profile by using PowerShell, run the following command:

```
Set-PCSupportProfile -website '<url>' -email '<email>' -phone '<phone number>'
```

C#

To update your support profile, do the following:

1. Get your support profile, and then update it as necessary.
2. Use your **IPartnerOperations.Profiles** collection. Call the **SupportProfile** property, followed by the **Update()** or **UpdateAsync()** method.

```
// IAggregatePartner partnerOperations;

// updated profile
SupportProfile newSupportProfile = new SupportProfile
{
    Email = supportProfile.Email,
    Website = supportProfile.Website,
    Telephone = new Random().Next(10000000, 99999999).ToString(CultureInfo.InvariantCulture)
};

SupportProfile updatedSupportProfile = partnerOperations.Profiles.SupportProfile.Update(newSupportProfile);
```

REST API

Request syntax

METHOD	REQUEST URI
PUT	{baseUrl}/v1/profiles/supportprofile HTTP/1.1

Request body

Enter the full support-profile resource in the request body.

Request example

```
PUT https://api.partnercenter.microsoft.com/v1/profiles/supportprofile HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: 603f3cd9-01b8-48f2-b65d-855a246f5bfd
MS-CorrelationId: 20604323-50bf-4738-9968-c5486ab32be0
Content-Type: application/json
Content-Length: 167
Expect: 100-continue
```

```
{
  "Email": "email@sample.com",
  "Telephone": "4255555555",
  "Website": "www.microsoft.com",
  "ProfileType": "support_profile",
  "Attributes": {
    "ObjectType": "PartnerSupportProfile"
  }
}
```

Response example

If the request is successful, this method returns updated **SupportProfile** object properties in the response body.

```
HTTP/1.1 200 OK
Content-Length: 502
Content-Type: application/json
MS-CorrelationId: 20604323-50bf-4738-9968-c5486ab32be0
MS-RequestId: 603f3cd9-01b8-48f2-b65d-855a246f5bfd
Date: Wed, 25 Nov 2015 07:16:18 GMT
```

```
{
  "email": "email@sample.com",
  "telephone": "4255555555",
  "website": "www.microsoft.com",
  "profileType": "support_profile",
  "links": {
    "self": {
      "uri": "/v1/profiles/support",
      "method": "GET",
      "headers": []
    }
  },
  "attributes": {
    "objectType": "PartnerSupportProfile"
  }
}
```

Next steps

- [Review](#) what are the available APIs for Azure CSP integration.
- [Review](#) the list of available Azure CSP Integration scenarios.

Get a list of indirect resellers

9/19/2017 • 1 min to read • [Edit Online](#)

Learn how to get a list of your indirect resellers. You can use one of the following options:

- PowerShell
- C#
- REST API

PowerShell

```
Get-PCIndirectResellers
```

C#

1. To establish an interface for relationship collection operations, use the **partnerOperations.Relationships** property.
2. Call the **Get** or the **Get_Async** method.
3. To identify the relationship type, pass a member of the **PartnerRelationshipType** enumeration. To retrieve indirect resellers, you must use **IsIndirectCloudSolutionProviderOf**.

```
// IAggregatePartner partnerOperations;  
  
var indirectResellers = partnerOperations.Relationships.Get(PartnerRelationshipType.IsIndirectCloudSolutionProviderOf);
```

REST API

Request

Request syntax

METHOD	REQUEST URI
GET	{baseUrl}/v1/relationships? relationship_type=IsIndirectCloudSolutionProviderOf HTTP/1.1

URI parameters

The following table lists the query parameters that are required for identifying the relationship type.

NAME	TYPE	DESCRIPTION
------	------	-------------

NAME	TYPE	DESCRIPTION
relationship_type	string	<ul style="list-style-type: none"> The string representation of a member name from the PartnerRelationshipType enumeration. If the partner is signed in as a provider and you want a list of the indirect resellers with whom the provider has established a relationship, use IsIndirectCloudSolutionProviderOf. If the partner is signed in as a reseller and you want a list of the indirect providers with whom the user has established a relationship, use IsIndirectResellerOf.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/relationships?relationship_type=IsIndirectCloudSolutionProviderOf HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: 144391a4-fb06-41ae-b684-3308ce4706bd
MS-CorrelationId: 72524ef8-81aa-4141-a049-45a4fece5d84
X-Locale: en-US
Host: api.partnercenter.microsoft.com
```

Response

If the request is successful, the response body contains a collection of **PartnerRelationship** resources that identify the resellers.

Response example

HTTP/1.1 200 OK
Content-Length: 298
Content-Type: application/json; charset=utf-8
MS-CorrelationId: 72524ef8-81aa-4141-a049-45a4fece5d84
MS-RequestId: 144391a4-fb06-41ae-b684-3308ce4706bd
MS-CV: b21LlImIM0yFMPQQ.0
MS-ServerId: 030020643
Date: Wed, 05 Apr 2017 21:08:44 GMT

```
{
  "totalCount": 2,
  "items": [ {
    "id": "484e548c-f5f3-4528-93a9-c16c6373cb59",
    "name": "First Up Consultants",
    "relationshipType": "is_indirect_cloud_solution_provider_of",
    "state": "Active",
    "mpnId": "4847383",
    "location": "US",
    "attributes": {
      "objectType": "PartnerRelationship"
    }
  }, {
    "id": "b01b1487-b36e-4e6d-9b5e-0b58974c4b28",
    "name": "ReleCloud",
    "relationshipType": "is_indirect_cloud_solution_provider_of",
    "state": "Active",
    "mpnId": "4847433",
    "location": "BR",
    "attributes": {
      "objectType": "PartnerRelationship"
    }
  }
],
  "attributes": {
    "objectType": "Collection"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Get and verify a partner's Microsoft Partner Network profile

9/19/2017 • 2 min to read • [Edit Online](#)

Learn how to get a partner's Microsoft Partner Network (MPN) profile, and verify an MPN ID. You can use one of the following options:

- PowerShell
- C#
- REST API

PowerShell

```
Get-PCMpnProfile -mpnid <MPNId>
```

C#

Verify

To verify a partner's MPN ID:

1. To get an interface for partner profile collection operations, use the **IAggregatePartner.Profiles** property.
2. To get an interface for MPN profile operations, use the **MpnProfile** property.
3. To retrieve the MPN profile, call the **Get** or the **GetAsync** method with the MPN ID.

If you omit the MPN ID from the **Get** or **GetAsync** call, the request attempts to retrieve the MPN profile of the partner who currently is signed in.

```
// IAggregatePartner partnerOperations;  
// string partnerMpnId;  
  
var partnerProfile = partnerOperations.Profiles.MpnProfile.Get(partnerMpnId);
```

Retrieve

To get a partner's MPN profile:

1. Use your **IAggregatePartner.Profiles** collection.
2. Call the **MpnProfile** property.
3. Call the **Get()** or the **GetAsync()** method.

```
// IAggregatePartner partnerOperations;  
  
mpnProfile = partnerOperations.Profiles.MpnProfile.Get();
```

REST API

Verify

Request syntax

METHOD	REQUEST URI
GET	{baseUrl}/v1/profiles/mpn?mpnId={mpn-id} HTTP/1.1

URI parameter

NAME	TYPE	DESCRIPTION
mpn-id	int	An MPN ID that identifies the partner. If you don't include this value, the profile of the partner who currently is signed in is retrieved.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/profiles/mpn?mpnId=9999999 HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: 560df6b9-6e53-4954-aed7-133477ac1194
MS-CorrelationId: e937630b-8341-4d70-8f73-450d32ee0189
X-Locale: en-US
MS-PartnerCenter-Client: Partner Center .NET SDK
Host: api.partnercenter.microsoft.com
Connection: Keep-Alive
```

Response example

If the request is successful, the response body contains the **MpnProfile** resource for the partner.

Success

```
HTTP/1.1 200 OK
Content-Length: 159
Content-Type: application/json; charset=utf-8
MS-CorrelationId: e937630b-8341-4d70-8f73-450d32ee0189
MS-RequestId: e39e0ddf-3fd0-4b7e-bb4e-8aebc242d3ee
MS-CV: s2GvkNngZsUSadxQX.0
MS-ServerId: 030011719
Date: Thu, 13 Apr 2017 18:13:40 GMT
```

```
{
  "mpnId": "4391507",
  "profileType": "MpnProfile",
  "links": {
    "self": {
      "uri": "/profiles/mpn",
      "method": "GET",
      "headers": []
    }
  },
  "attributes": {
    "objectType": "MpnProfile"
  }
}
```

Failure

HTTP/1.1 404 Not Found
Content-Length: 124
Content-Type: application/json; charset=utf-8
MS-CorrelationId: e937630b-8341-4d70-8f73-450d32ee0189
MS-RequestId: 560df6b9-6e53-4954-aed7-133477ac1194
MS-CV: sLRFZMWm+EKuL47u.0
MS-ServerId: 102030524
Date: Thu, 13 Apr 2017 18:26:51 GMT

```
{
  "code": 3000,
  "description": "Partner Organization with partner_id 9999999 could not be found",
  "data": [],
  "source": "PartnerFD"
}
```

Retrieve

Request syntax

METHOD	REQUEST URI
GET	{baseUrl}/v1/profiles/mpn HTTP/1.1

Request example

GET https://api.partnercenter.microsoft.com/v1/profiles/mpn HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: 76879323-92d1-437e-90dd-c84dbb9f7dec
MS-CorrelationId: cb9f3209-d020-4bf9-871c-e1f1c75348f8
Connection: Keep-Alive

Response example

If the request is successful, this method returns an **MpnProfile** object in the response body.

HTTP/1.1 200 OK
Content-Length: 177
Content-Type: application/json; charset=utf-8
MS-CorrelationId: cb9f3209-d020-4bf9-871c-e1f1c75348f8
MS-RequestId: 76879323-92d1-437e-90dd-c84dbb9f7dec
Date: Mon, 21 Mar 2016 05:51:29 GMT

```
{
  "mpnId": "<mpnID>",
  "profileType": "MpnProfile",
  "links": {
    "self": {
      "uri": "/profiles/mpn",
      "method": "GET",
      "headers": []
    }
  },
  "attributes": {
    "objectType": "MpnProfile"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Get a list of a customer's subscriptions by using a Microsoft Partner Network ID

9/19/2017 • 1 min to read • [Edit Online](#)

Learn how to use a partner's Microsoft Partner Network (MPN) ID to get a list of a customer's subscriptions that the partner gave to the customer. You can use the information for Azure Cloud Solution Provider (Azure CSP) integration. To get the list of subscriptions, you can use one of the following options:

- C#
- REST API

C#

1. To identify the customer, use the **IAggregatePartner.Customers.ById** method with the customer ID.
2. To get an interface for the collection operations of customer subscriptions, use the **Subscriptions** property.
3. To identify the partner and to get an interface for partner subscription operations, call the **ByPartner** method with the MPN ID.
4. To get the collection, call the **Get** or the **GetAsync** method.

```
// IAggregatePartner partnerOperations;  
// string customerId;  
// string partnerMpnId;  
  
var customerSubscriptionsByMpnId = partnerOperations.Customers.ById(customerId).Subscriptions.ByPartner(partnerMpnId).Get();
```

REST API

Request syntax

METHOD	REQUEST URI
GET	{baseUrl}/v1/customers/{customer-id}/subscriptions?mpn_id={mpn-id} HTTP/1.1

URI parameters

NAME	TYPE	DESCRIPTION
customer-id	string	A GUID-format string that identifies the customer.
mpn-id	int	An MPN ID that identifies the partner.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers/c501c3c4-d776-40ef-9ecf-9cefb59442c1/subscriptions?mpn_id=4847383 HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: d0e38dfd-a2c5-4a14-ac06-12d30f0ec54e
MS-CorrelationId: e937630b-8341-4d70-8f73-450d32ee0189
X-Locale: en-US
Host: api.partnercenter.microsoft.com
Connection: Keep-Alive
```

Response example

If the request is successful, the response body contains the collection of subscription resources.

```

HTTP/1.1 200 OK
Content-Length: 985
Content-Type: application/json; charset=utf-8
MS-CorrelationId: e937630b-8341-4d70-8f73-450d32ee0189
MS-RequestId: d0e38dfd-a2c5-4a14-ac06-12d30f0ec54e
MS-CV: LdFhumsx6Ea0KI5Z.0
MS-ServerId: 101112202
Date: Thu, 13 Apr 2017 20:58:08 GMT

{
  "totalCount": 1,
  "items": [{
    "id": "42226ED6-070A-4E0F-B80C-4CDFB3E97AA7",
    "offerId": "DB2E705F-B82A-4024-A3D5-D88E12F2DB35",
    "offerName": "Intune Device",
    "friendlyName": "new offer purchase",
    "quantity": 5,
    "unitType": "Licenses",
    "creationDate": "2017-04-10T23:02:26.02Z",
    "effectiveStartDate": "2017-04-10T00:00:00Z",
    "commitmentEndDate": "2018-05-07T00:00:00Z",
    "status": "active",
    "autoRenewEnabled": true,
    "isTrial": false,
    "billingType": "license",
    "billingCycle": "monthly",
    "partnerId": "4847383",
    "contractType": "subscription",
    "links": {
      "offer": {
        "uri": "/offers/DB2E705F-B82A-4024-A3D5-D88E12F2DB35?country=US",
        "method": "GET",
        "headers": []
      },
      "self": {
        "uri": "/customers/c501c3c4-d776-40ef-9ecf-9cefb59442c1/subscriptions/42226ED6-070A-4E0F-B80C-4CDFB3E97AA7",
        "method": "GET",
        "headers": []
      }
    },
    "orderId": "3EDDCAC6-63B2-4C40-B0B6-F47E18301492",
    "attributes": {
      "etag": "eyJpZCI6IjYmJlZWQ2LTA3MGEtNGUwZiIiODBiJTRjZGZiM2U5N2FhNyIsInZlcnNpb24iOiJF9",
      "objectType": "Subscription"
    }
  ]
},
{
  "attributes": {
    "objectType": "Collection"
  }
}
}

```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Get managed services for a customer by ID

9/19/2017 • 1 min to read • [Edit Online](#)

Learn how to retrieve a list of all of a customer's services for which you have delegated admin permissions. For information about providing support for customers and escalating service requests to Microsoft, see [Azure Cloud Solution Provider support](#). To get a list of managed services by using a customer ID, you can use one of the following options:

- PowerShell
- C#
- REST API

PowerShell

```
$customer = Get-PCCustomer -tenantid '<tenant id GUID>'
Get-PCManagedServices -tenantid $customer.id
```

C#

1. Use your **IAggregatePartner.Customers** collection and call the **ById()** method.
2. Call the **ManagedServices** property, followed by the **Get()** or the **GetAsync()** method.

```
// IAggregatePartner partnerOperations;
// var selectedCustomerId as Customer;

ResourceCollection<ManagedService> managedServices = partnerOperations.Customers.ById(selectedCustomerId).ManagedServices.Get();
```

REST API

Request

METHOD	REQUEST URI
GET	{baseUrl}/v1/customers/{customer-tenant-id}/managementservices HTTP/1.1

URI parameter

The following query parameter is required for getting the customer's managed services.

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	A GUID that corresponds to the customer.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/customers/<customer-tenant-id>/managementservices HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: 4ff57220-f17b-4d8f-8e09-78334c57ba00
MS-CorrelationId: 03d6064a-f048-4aee-8892-ed46dc5c8bee
```

Response

If the request is successful, this method returns a collection of **Managed Service** objects in the response body.

HTTP/1.1 200 OK
Content-Length: 10588
Content-Type: application/json
MS-CorrelationId: 03d6064a-f048-4ace-8892-ed46dc5c8bee
MS-RequestId: 4ff57220-f17b-4d8f-8e09-78334c57ba00
Date: Mon, 23 Nov 2015 18:02:12 GMT

```
{
  "totalCount": 2,
  "items": [ {
    "id": "Exchange",
    "name": "Exchange",
    "groupName": "Office",
    "links": {
      "adminService": {
        "uri": "https://portal.office.com/Partner/BeginClientSession.aspx?CTID=<ctid>&CSDEST=Exchange&InitialDomain=
<domain>&PrimaryDomain=<domain>",
        "method": "GET",
        "headers": []
      },
      "serviceHealth": {
        "uri": "https://portal.office.com/Partner/BeginClientSession.aspx?CTID=<ctid>&CSDEST=ServiceStatus",
        "method": "GET",
        "headers": []
      },
      "serviceTicket": {
        "uri": "https://portal.office.com/Partner/BeginClientSession.aspx?CTID=<ctid>&CSDEST=Support",
        "method": "GET",
        "headers": []
      }
    },
    "attributes": {
      "objectType": "ManagedService"
    }
  },
  {
    "id": "MicrosoftCommunicationsOnline",
    "name": "SkypeforBusiness",
    "groupName": "Office",
    "links": {
      "adminService": {
        "uri": "https://portal.office.com/Partner/BeginClientSession.aspx?CTID=<ctid>&CSDEST=MicrosoftCommunicationsOnline",
        "method": "GET",
        "headers": []
      },
      "serviceHealth": {
        "uri": "https://portal.office.com/Partner/BeginClientSession.aspx?CTID=<ctid>&CSDEST=ServiceStatus",
        "method": "GET",
        "headers": []
      },
      "serviceTicket": {
        "uri": "https://portal.office.com/Partner/BeginClientSession.aspx?CTID=<ctid>&CSDEST=Support",
        "method": "GET",
        "headers": []
      }
    },
    "attributes": {
      "objectType": "ManagedService"
    }
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).

- See the list of [Azure CSP integration scenarios](#).

Get service request support topics

9/19/2017 • 1 min to read • [Edit Online](#)

Learn how to retrieve a list of valid topics for service requests. You can use one of the following options:

- PowerShell
- C#
- REST API

PowerShell

```
Get-PCSRTopics
```

C#

1. Use your **IPartnerOperations** collection to retrieve the **ServiceRequests** property of the resulting object.
2. Use the **SupportTopics** property, and then call the **Get()** or the **GetAsync()** method.

```
// IPartner partnerOperations;  
  
ResourceCollection<SupportTopic> supportTopicsCollection = partnerOperations.ServiceRequests.SupportTopics.Get();
```

REST API

Request

Request syntax

METHOD	REQUEST URI
GET	{baseUrl}/v1/servicerequests/supporttopics HTTP/1.1

Request example

```
GET https://api.partnercenter.microsoft.com/v1/servicerequests/supporttopics HTTP/1.1  
Authorization: Bearer <token>  
Accept: application/json  
MS-RequestId: 4c1e8b6c-d136-4931-9df9-d85ec08ccffe  
MS-CorrelationId: f447b215-f9bc-48da-a05a-3b5322d86a9c  
X-Locale: en-US
```

Response

If the request is successful, this method returns a collection of the valid topics for a support request.

Response example

HTTP/1.1 200 OK
Content-Length: 4982
Content-Type: application/json
MS-CorrelationId: f447b215-f9bc-48da-a05a-3b5322d86a9c
MS-RequestId: 4c1e8b6c-d136-4931-9df9-d85ec08ccffe
Date: Fri, 29 Jan 2016 22:31:36 GMT

```
{
  "totalCount":14,
  "items":[{
    "name":"Partner Center Issues",
    "description":"Office 365 questions from the CSP (Cloud Solution Provider) partners using Partner Center",
    "id":32444667,
    "attributes":{
      "objectType":"SupportTopic"
    }
  },
  {
    "name":"Cannot manage my profile",
    "description":"Issues that are related to errors or problems with managing a profile",
    "id":32444671,
    "attributes":{
      "objectType":"SupportTopic"
    }
  }
],
  "attributes":{
    "objectType":"Collection"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Escalate Azure Cloud Solution Provider issues to Microsoft

10/6/2017 • 2 min to read • [Edit Online](#)

If you are an Azure Cloud Solution Provider (Azure CSP) partner, you are [responsible for customer technical and billing support](#). But sometimes your customers will experience issues with Azure that only Microsoft can solve. These might include the following:

- Undocumented problems with Azure services that aren't operating according to service descriptions
- Unavailable services
- Bugs and other irregularities that affect service appearance or operation
- Large-scale network disruptions
- Regional issues that have multitenant impact
- Billing disputes

You need to escalate these kinds of issues with Azure services to Microsoft, and create a support request. First, though, ensure that the issue is not caused by an incorrect service configuration. Also, ensure that Microsoft supports the customer's architecture.

IMPORTANT

Don't escalate every customer issue to Microsoft. First, complete proper troubleshooting on your side.

Even if you've escalated a customer's issue with Azure to Microsoft, it's still your responsibility to manage the resolution process. Microsoft support engineers will communicate with a customer only through a partner.

IMPORTANT

In this article, we cover the support process for Azure CSP direct partners. If you're an Azure CSP indirect reseller, contact your Azure CSP indirect provider for details about how this process will work for you.

Create a service request

1. On the Partner Center **Dashboard** menu, select **Customers**. Select the customer, and then select **Service management**.

Microsoft Partner Center Programs How-to Support Find a Partner Dashboard

ACME Corporation Subscriptions [Export subscriptions](#)

Subscriptions Add subscription

Customer insights

Users and licenses

Service management

Account

← Customers

License-based subscriptions

This customer doesn't have any license-based subscriptions.

Usage-based subscriptions [All resources \(Azure portal\)](#)

Usage data isn't provided for integration sandbox accounts.

Subscription	Current estimate	% of total	Status
Microsoft Azure		0%	Active

Industries For customers For partners For developers Values Company

Manufacturing & resources	Small & midsize businesses	Get listed	Microsoft Azure	Diversity and inclusion	Careers
Financial services	Enterprise	Microsoft partner resources	Microsoft Visual Studio	Accessibility	About Microsoft
Retail	Security at home	Find a solutions provider	Microsoft Developer Network	Microsoft in education	Company news
Health	Microsoft Envision	Partner with Microsoft Azure	TechNet	Microsoft philanthropies	Investors
Government	Microsoft Ignite	Microsoft Inspire	Application Development	Corporate social responsibility	Research

2. In the **Support tickets** section, select the **New request** drop-down menu, and then select **Microsoft Azure**.

Office 365 Portal normal

Office Subscription normal

Microsoft Intune normal

OneDrive normal

Planner normal

Power BI normal

Rights Management Service normal

SharePoint normal

Skype for Business normal

Social Engagement normal

Sway normal

Yammer Enterprise normal

Microsoft Azure ...

Service requests

Open Closed

No requests

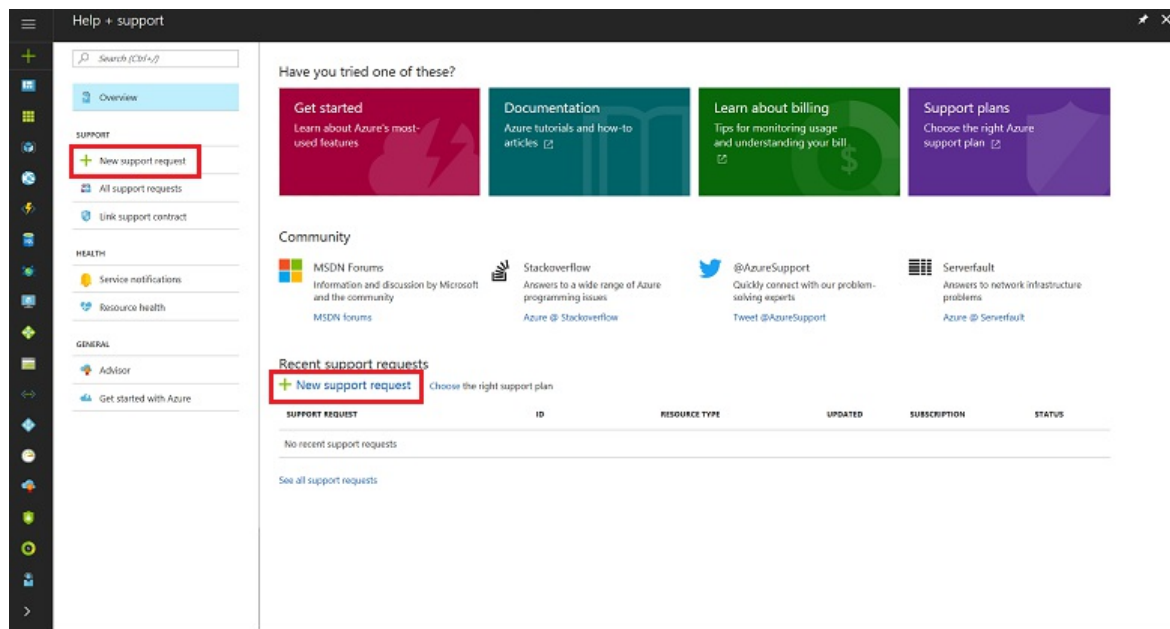
New request

Microsoft Azure

Office 365, Dynamics 365, Enterprise Mobility Suite

Ticket ID

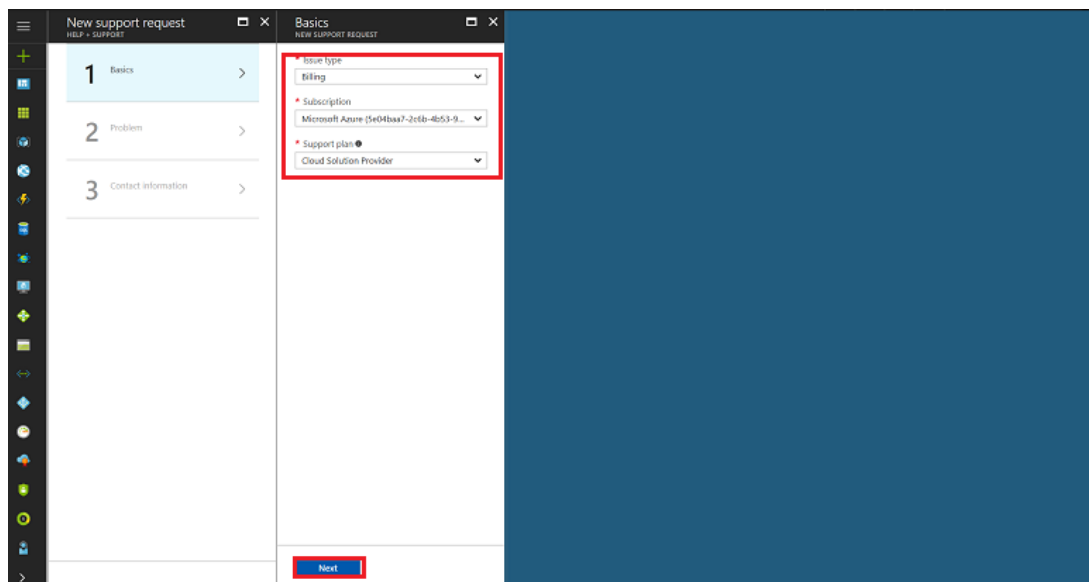
3. In the Azure portal, select **New support request**.



4. Enter the relevant information for the service request, and then select **Submit**. For more information about how to fill out the support request, see [Support request form](#).

Support request form

1. On the **Basics** blade:
 - a. Select an **Issue type** (**Technical**, **Billing**, **Quota**, or **Subscription management**).
 - b. Select the **Subscription** to which the issue applies.
 - c. Select the **Support plan**. The default support plan is **Cloud Solution Provider**. [Additional plans](#) might be listed, if you've purchased them.



2. On the **Problem** blade:
 - a. Select the problem's **Severity**. This ranges from **C - Minimal impact** (low priority) to **A - Critical impact** (immediate priority).
 - b. Select the **Problem type**. The options vary, depending on the issue type you selected in step 1. Select the option that most closely resembles the problem. When you select the problem type, you see a list of recommended steps and documents that might help you solve those types of problems. *Try these solutions before you submit a support request.*
 - c. In the **Details** section, describe the issue with as much information as possible. Include any error messages, if applicable.

- d. Under **When did the problem start**, enter the date and local time at which the problem first occurred.
- e. If applicable, upload a file to the **File upload** section.

Microsoft Azure Help + support > New support request > Problem > Related help

New support request HELP + SUPPORT

1 Basics ✓

2 Problem >

3 Contact information >

Problem new support request

Severity: Minimal impact

Problem type: Change Payment Method

Details: I need to change my payment method.

Found solutions based on your problem description. See solutions on the right.

When did the problem start?

Choose a date: [Date Picker] Enter a local time: [Time Input]

File upload: Select a file [File Upload Button]

Next

Related help

Want a solution right now?

Solutions based on your problem description

Reactivate disabled Azure subscription: The bill exceeds your credit card limit
Feb 3, 2017 - To resolve this issue, switch to a different credit card. Or if you're representing a business, you can switch to pay by invoice. - Continue reading on Microsoft Docs

How to change payment method?
You can add, change, or remove a credit card that's associated with your Azure account when you sign in to the Azure Account Center as the account administrator.

Add a credit card as a payment method
See full solution

I need to change the payment method
Recommended steps

1. Find out who's the Account Admin.
2. Sign in to the Account Center as the Account Admin.
3. Under Click a subscription to view details and usage, select the subscription for which

See full solution

3. On the **Contact information** blade:
 - a. For **Preferred contact method**, select email or phone as the preferred way to reach the point of contact.
 - b. In **Contact information**, provide *your* contact information, *not* your customer's information.
4. Select **Create**. The support request appears in Partner Center for the support team to address.

Microsoft Azure << New support request > Contact information > Related help

New support request HELP + SUPPORT

1 Basics ✓

2 Problem ✓

3 Contact information >

Contact information new support request

Contact options

Preferred contact method: Email

Response: Business Hours

Language: English

Contact information

First name: John ✓

Last name: Doe ✓

Email: jg1990@outlook.com ✓

Who else should we email? ✓

Phone number: ✓

Country/region: United States

Save contact changes for future support requests: ✓

By clicking create you accept the terms and conditions. View our privacy policy.

Create

Related help

Want a solution right now?

Solutions based on your problem description

Reactivate disabled Azure subscription: The bill exceeds your credit card limit
Feb 3, 2017 - To resolve this issue, switch to a different credit card. Or if you're representing a business, you can switch to pay by invoice. - Continue reading on Microsoft Docs

How to change payment method?
You can add, change, or remove a credit card that's associated with your Azure account when you sign in to the Azure Account Center as the account administrator.

Add a credit card as a payment method
See full solution

I need to change the payment method
Recommended steps

1. Find out who's the Account Admin.
2. Sign in to the Account Center as the Account Admin.
3. Under Click a subscription to view details and usage, select the subscription for which

See full solution

Later, review your customer's service requests within the Microsoft Azure portal by selecting **Manage support requests**.

If you are experiencing issues with Partner Center portal itself, not related to Azure services, you should use [Partner Center problem reporting](#) procedure.

Next steps

- Review [support plans for Azure CSP partners](#).

- Learn about [Azure CSP partner support responsibilities](#).
- Review [support from Microsoft](#) section of Partner Center documentation.

Delete a customer account from an integration sandbox

9/19/2017 • 1 min to read • [Edit Online](#)

Learn how to remove a customer account from the Testing in Production integration sandbox. You can use one of the following options:

- C#
- REST API

C#

1. Sign in to the sandbox by using your sandbox credentials.
2. Use your **IPartner.Customer** collection and call the **ById()** method.
3. Call the **Delete** method.

```
// IPartnerCredentials tipAccountCredentials;  
// string customerTenantId;  
  
IPartner tipAccountPartnerOperations = PartnerService.Instance.CreatePartnerOperations(tipAccountCredentials);  
  
tipAccountPartnerOperations.Customers.ById(customerTenantId).Delete();
```

REST API

Request

Request syntax

METHOD	REQUEST URI
DELETE	{baseUrl}/v1/customers/{customer-tenant-id} HTTP/1.1

URI parameter

The following query parameter is required for deleting an integration sandbox account.

NAME	TYPE	DESCRIPTION
customer-tenant-id	guid	A GUID-format customer tenant ID that refers to a specific customer.

Request example

```
DELETE https://api.partnercenter.microsoft.com/v1/customers/<customer-tenant-id> HTTP/1.1  
Accept: application/json  
MS-RequestId: 655890ba-4d2b-4d09-a95f-4ea1348686a5  
MS-CorrelationId: 1438ea3d-b515-45c7-9ec1-27ee0cc8e6bd  
Content-Length: 0
```

Response

If the request is successful, this method returns an empty response.

Response example

```
HTTP/1.1 204 No Content
Content-Length: 0
MS-CorrelationId: 1438ea3d-b515-45c7-9ec1-27ee0cc8e6bd
MS-RequestId: 655890ba-4d2b-4d09-a95f-4ea1348686a5
Date: Wed, 16 Mar 2016 00:43:02 GMT
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Get a record of Partner Center activity by user

9/19/2017 • 1 min to read • [Edit Online](#)

Learn how to get detailed information about changes that have been made to customer accounts and subscriptions over a period of time. You can use the information for auditing or for analytical purposes. To get a record of Microsoft Partner Center activity by user, you can use one of the following options:

- Included information
- C#
- REST API

Included information

The following actions are included in the log, along with the user account that made the changes, and the date.

ACTIONS IN PARTNER CENTER	ACTIONS VIA THE PARTNER CENTER SDK
<ul style="list-style-type: none">• Update subscriptions• Update orders• Update an Azure spending budget• Upgrade subscriptions	<ul style="list-style-type: none">• Create, update, or delete customer accounts• Update subscriptions• Update orders• Update an Azure spending budget• Upgrade subscriptions <p>Note that changes made by using the CREST API are not included in the log.</p>

C#

To retrieve a record of Partner Center activity by user, use **IPartner.AuditRecords** to get a collection that enumerates the actions.

```
//IPartner ScopedPartnerOperations;  
//var filter;  
//var startDate;  
  
//Setting up the filter, per the code sample.  
var filter = new SimpleFieldFilter(AuditRecordSearchField.CompanyName.ToString(), FieldFilterOperation.Substring, "00");  
  
var collection = scopedPartnerOperations.AuditRecords.Query(startDate.Date, query: QueryFactory.Instance.BuildSimpleQuery(filter: filter));
```

REST API

Request

Request syntax

METHOD	REQUEST URI
GET	{baseUrl}/v1/auditrecords?startDate=yyyy-mm-dd HTTP/1.1
GET	{baseUrl}/v1/auditrecords?startDate=yyyy-mm-dd&endDate=yyyy-mm-dd HTTP/1.1

METHOD	REQUEST URI
GET	{baseUrl}/v1/auditrecords?startDate=yyyy-mm-dd&endDate=yyyy-mm-dd&filter={{"Field":"CompanyName","Value":"Central","Operator":0}} HTTP/1.1

URI parameter

Use the following query parameter search *only* for actions that affect a single customer.

NAME	TYPE	DESCRIPTION
CompanyName	string	The exact name of the customer, as listed in Partner Center.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/auditrecords?startDate=2016-05-02
Authorization: Bearer <AUTH-TOKEN>
Accept: application/json
MS-Contract-Version: v1
MS-RequestId: be82a8ba-4a53-49f7-8313-b033c058687e
MS-CorrelationId: 648a26a4-a63e-459f-844b-4f29d7913353
Content-Length: 0
Content-Type: application/json
```

Response

If the request is successful, this method returns a set of activities that meet the filter requirements.

Response example

```

HTTP/1.1 200 OK
Content-Length: 1346
Content-Type: application/json; charset=utf-8
Server: Microsoft-HTTPAPI/2.0
MS-CorrelationId: 648a26a4-a63e-459f-844b-4f29d7913353
MS-RequestId: be82a8ba-4a53-49f7-8313-b033c058687e
Date: Tue, 10 May 2016 19:00:10 GMT

{
  "totalCount": 1,
  "items": [ {
    "customerId": "49bd2a7e-0776-487d-99f6-e7536fb82049",
    "customerName": "Customer",
    "userPrincipalName": "user@domain.onmicrosoft.com",
    "applicationId": "87fe183e-041c-4cf7-b221-c065d9582cde",
    "resourceType": "order",
    "resourceNewValue": "\ { \"Id\": null, \"ReferenceCustomerId\": \"49bd2a7e-0776-487d-99f6-e7536fb82049\", \"LineItems\": [ { \"LineNumber\": 0, \"OfferId\": \"5C9FD4CC-EDCE-44A8-8E91-07DF09744609\", \"SubscriptionId\": null, \"ParentSubscriptionId\": null, \"FriendlyName\": \"Office 365 Business\", \"Quantity\": 2, \"PartnerIdOnRecord\": null, \"Attributes\": { \"ObjectType\": \"OrderLineItem\" }, \"CreationDate\": null, \"Attributes\": { \"ObjectType\": \"Order\" } } ], \"operationType\": \"create_order\", \"operationDate\": \"2016-05-10T18:34:37.5980461Z\", \"operationStatus\": \"succeeded\", \"originalCorrelationId\": \"94588c3f-5e41-4dbd-986f-1d5146453a46\", \"sessionId\": \"9c75b432-a7b3-4772-bff1-c9d44b190b98\", \"attributes\": { \"objectType\": \"AuditRecord\" } } ],
    \"links\": {
      \"self\": {
        \"uri\": \"/auditrecords?startDate=2016-05-02&filter={\" Field \": \" CompanyName \", \" Value \": \" Central \", \" Operator \": \" equals \" }\",
        \"method\": \"GET\",
        \"headers\": [
        ]
      }
    },
    \"attributes\": {
      \"objectType\": \"Collection\"
    }
  }
}

```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Get address formatting rules by market

9/19/2017 • 1 min to read • [Edit Online](#)

Learn how to retrieve the expected address format based on the ISO code for the respective market. You can use one of the following options:

- PowerShell
- REST API

PowerShell

```
Get-PCAddressRulesByMarket -countryid '<country two-digit id>'
```

REST API

Request

Request syntax

METHOD	REQUEST URI
GET	{baseUrl}/v1/countryvalidationrules/{isocode-id} HTTP/1.1

URI parameter

NAME	TYPE	DESCRIPTION
isocode-id	string	The two-character ISO country code.

Request example

```
GET https://api.partnercenter.microsoft.com/v1/countryvalidationrules/{isocode-id} HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: 124b0e41-a093-4fec-b871-3eeb45fd734b
MS-CorrelationId: 5cfd634d-b936-47af-87f0-0f0217425dcc
```

Response

If a request is successful, this method returns a **CountryInformation** object in the response body.

Response example

HTTP/1.1 200 OK
Content-Length: 1856
Content-Type: application/json
MS-CorrelationId: 5cfd634d-b936-47af-87f0-0f0217425dcc
MS-RequestId: 124b0e41-a093-4fec-b871-3eeb45fd734b
Date: Wed, 25 Nov 2015 06:36:47 GMT

```
{
  "iso2Code": "US",
  "defaultCulture": "en-US",
  "isStateRequired": true,
  "states": {
    "value": ["AK", "AL", "AR", "AZ", "CA", "CO", "CT", "DC", "DE", "FL", "GA", "HI", "IA", "ID", "IL", "IN",
      "KS", "KY", "LA", "MA", "MD", "ME", "MI", "MN", "MO", "MS", "MT", "NC", "ND", "NE", "NH", "NJ", "NM", "NV",
      "NY", "OH", "OK", "OR", "PA", "RI", "SC", "SD", "TN", "TX", "UT", "VA", "VT", "WA", "WI", "WV", "WY"]
  },
  "supportedLanguages": {
    "value": ["en",
      "es"]
  },
  "supportedCultures": {
    "value": ["en-US",
      "es-US"]
  },
  "isPostalCodeRequired": true,
  "postalCodeRegex": "^\\d{
    5
  }(-\\d{
    4
  })?$",
  "isCityRequired": true,
  "isVatIdSupported": false,
  "taxIdFormat": "US#####",
  "taxIdSample": "US999965",
  "phoneNumberRegex": "^(1[\\-\\+\\.]?((\\d{3})\\d{3})[\\-\\+\\.]?(\\d{3})[\\-\\+\\.]?(\\d{4})$)",
  "isRegistrationNumberSupported": false,
  "isTaxIdSupported": true,
  "resellerAgreementRegion": "AOC",
  "geographicRegion": "NorthAndLatinAmerica",
  "countryCallingCodes": {
    "value": ["1"]
  },
  "attributes": {
    "objectType": "CountryInformation"
  }
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Validate a customer address

9/19/2017 • 2 min to read • [Edit Online](#)

Learn how to validate an address by using the address validation API. You can use one of the following options:

- PowerShell
- C#
- REST API

PowerShell

To validate an address by using PowerShell, you have two options. You can use this command:

```
``powershell
$address = New-PCAddress -AddressLine1 '<string>' -AddressLine2 '<string>' -City '<string>' -State '<string>' -PostalCode '<string>' -country
'two-digit country code' -region '<string>'
Test-PCAddress -Address <DefaultAddress>
``
```

Or, you can use this command:

```
``powershell
Test-PCAddress -AddressLine1 '<string>' -AddressLine2 '<string>' -City '<string>' -State '<string>' -PostalCode '<string>' -country 'two-digit
country code' -region '<string>'
``
```

C#

1. Instantiate a new **Address** object and populate it with the address you want to validate.
2. Retrieve an interface for **Validations** operations from the **IAggregatePartners.Validations** property.
3. Call the **IsAddressValid** method with the address object.

```
// Create an address to validate.
Address address = new Address()
{
    AddressLine1 = "One Microsoft Way",
    City = "Redmond",
    State = "WA",
    PostalCode = "98052",
    Country = "US"
};

// Validate the address.
bool result = partnerOperations.Validations.IsAddressValid(address);

// If the address is valid, the result should equal true.
Console.WriteLine("Result: " + result.ToString());

// The following is an example that causes address validation to fail.
try
{
    // Change to an invalid postal code for this address.
    address.PostalCode = "98007";

    // Validate the address.
    result = partnerOperations.Validations.IsAddressValid(address);

    Console.WriteLine("ERROR: The code should have thrown an exception - BadRequest(400).");
}
catch (PartnerException exception)
{
    if (exception.ErrorCategory == PartnerErrorCategory.BadInput)
    {
        Console.WriteLine(exception.ErrorCategory.ToString());
        Console.WriteLine("Exception:");
        Console.WriteLine("Message: {0}", exception.Message);
    }
    else
    {
        throw;
    }
}
}
```

REST API

Request

Request syntax

METHOD	REQUEST URI
POST	{baseUrl}/v1/validations/address HTTP/1.1

Request body

The following properties are present in the request body.

NAME	TYPE	DESCRIPTION
addressline1	string	The first line of the address.
addressline2	string	(Optional) The second line of the address.

NAME	TYPE	DESCRIPTION
city	string	The city.
state	string	The state.
postalcode	string	The postal code.
country	string	The two-character ISO alpha-2 country code.

Request example

```
POST https://api.partnercenter.microsoft.com/v1/validations/address HTTP/1.1
Content-Type: application/json
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: 0b30452a-8be2-4b8b-b25b-2d4850f4345f
MS-CorrelationId: 8a853a1a-b0e6-4cb0-ae87-d6dd32ac3a0c
X-Locale: en-US
Host: api.partnercenter.microsoft.com
Content-Length: 129
```

```
{
  AddressLine1: "One Microsoft Way",
  City: "Redmond",
  State: "WA",
  PostalCode: "98052",
  Country: "US"
}
```

Response

If the request is successful, the method returns a status code 200, as demonstrated in the success example.

If the request fails, the method returns a status code 400, as demonstrated in the failure example. The response body contains a JSON payload that has additional information about the error.

Response succeeded example

```
HTTP/1.1 200 OK
Content-Length: 0
MS-CorrelationId: 8a853a1a-b0e6-4cb0-ae87-d6dd32ac3a0c
MS-RequestId: 0b30452a-8be2-4b8b-b25b-2d4850f4345f
MS-CV: IqhjoWVYq0KI81dO.0
MS-ServerId: 030011719
Date: Mon, 13 Mar 2017 23:56:12 GMT
```

Response failed example

HTTP/1.1 400 Bad Request

Content-Length: 418

Content-Type: application/json; charset=utf-8

MS-CorrelationId: 8a853a1a-b0e6-4cb0-ae87-d6dd32ac3a0c

MS-RequestId: 0b30452a-8be2-4b8b-b25b-2d4850f4345f

MS-CV: pdlltMyvtkmGHDWt.0

MS-ServerId: 101112012

Date: Tue, 14 Mar 2017 01:57:55 GMT

```
{
  "code": 2007,
  "description": "{'code':'60071','reason':'ZipCityInvalid - Details: Field - 'City' is corrected from OldValue: 'Redmond' to NewValue: 'BELLEVUE','corrected_address':{'country':'US','region':'WA','city':'BELLEVUE','address_line1':'One Microsoft Way','postal_code':'98007'}},'object_type':'AddressValidation','resource_status':'Active'}",
  "data": [],
  "source": "PartnerFD"
}
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Verify domain availability

9/19/2017 • 1 min to read • [Edit Online](#)

Learn how to determine whether a domain is available for a customer to use. You can use one of the following options:

- PowerShell
- C#
- REST API

PowerShell

```
$domainname = '<name>'
$domain = $domainname+'.onmicrosoft.com'
Get-PCDomainAvailability -domain $domain
```

C#

1. To get an interface for domain operations, call **IAggregatePartner.Domains**.
2. Call the **ByDomain** method with the domain to check. This retrieves an interface for the operations that are available for a specific domain.
3. To see if the domain already exists, call the **Exists** method.

```
// IAggregatePartner partnerOperations;
// const string domain = "contoso.onmicrosoft.com";

bool result = partnerOperations.Domains.ByDomain(domain).Exists();
```

REST API

Request

Request syntax

METHOD	REQUEST URI
HEAD	{baseUrl}/v1/domains/{domain} HTTP/1.1

URI parameter

The following query parameter is required for verifying domain availability.

NAME	TYPE	DESCRIPTION
domain	string	A string that identifies the domain to check.

Request example

```
HEAD https://api.partnercenter.microsoft.com/v1/domains/contoso.onmicrosoft.com HTTP/1.1
Authorization: Bearer <token>
Accept: application/json
MS-RequestId: cf5b00d6-9240-431c-a973-cc06c904e5bf
MS-CorrelationId: ec57501a-a4c3-45ee-ab2b-da4250545fc9
X-Locale: en-US
Host: api.partnercenter.microsoft.com
Connection: Keep-Alive
```

Response

If the domain exists, it is not available for use. A response status code 200 OK is returned.

If the domain is not found, it is available for use. A response status code 404 Not Found is returned.

Response example when the domain is in use

```
HTTP/1.1 200 OK
Content-Length: 0
MS-CorrelationId: ec57501a-a4c3-45ee-ab2b-da4250545fc9
MS-RequestId: cf5b00d6-9240-431c-a973-cc06c904e5bf
MS-CV: 7UXAHds8J0mNUCSp.0
MS-ServerId: 201022015
Date: Tue, 31 Jan 2017 22:22:35 GMT
```

Response example when the domain is available

```
HTTP/1.1 404 Not Found
Content-Length: 0
MS-CorrelationId: 54770745-17f0-433c-bd7b-0265e5b38f98
MS-RequestId: 1169a4cd-3be7-4e29-9cb3-0f78ffa2e91e
MS-CV: RRmc+bEw9U2e97CC.0
MS-ServerId: 202010406
Date: Tue, 31 Jan 2017 22:36:01 GMT
```

Next steps

- Learn about [APIs for Azure CSP integration](#).
- See the list of [Azure CSP integration scenarios](#).

Migrate your customers to Azure CSP

11/2/2017 • 1 min to read • [Edit Online](#)

Are you an Azure Cloud Solution Provider (Azure CSP) partner, looking for new opportunities for your Azure CSP business? Start with migrating your customers to Azure CSP.

Migrate from on-premises workloads to Azure CSP

If your customers have on-premises workloads, running on VMs or bare metal servers, you can migrate these workloads to Azure CSP. For more information, see [Migrate on-premises workloads to Azure CSP](#).

Migrate from other channels to Azure CSP

You may also want to migrate existing Azure subscriptions from other channels to Azure CSP. This includes Enterprise Agreement and pay-as-you-go (sometimes called Azure Direct) subscriptions. For more information, see the following:

- [Azure subscription migration from Enterprise Agreement to Azure CSP](#)
- [Azure subscription migration from pay-as-you-go \(Azure Direct\) to Azure CSP](#)

For Open and BizSpark Azure subscriptions, follow the instructions for [pay-as-you-go \(Azure Direct\)](#). In these scenarios, unused credit cannot be transferred to CSP.

Next steps

- [Review](#) what Azure services are available in Azure CSP.
- [Learn](#) how customer support works in the Azure CSP model.
- [Enroll](#) in the Azure CSP program, and start making Azure business through Azure CSP.

Migrate on-premises workloads to Azure CSP

9/19/2017 • 2 min to read • [Edit Online](#)

Many customers want to migrate their existing workloads to the cloud. You, as an Azure Cloud Solution Provider (Azure CSP) partner, can help them migrate to Azure.

Be aware that some customers have never used cloud services before. All of their workloads are on-premises, or handled by physical servers they own. Transferring these workloads to Azure CSP can sometimes be challenging, but this guide will help you through the process.

Note that there is no easy "switch" for this process. Make sure that you and the customer are both aware that migration takes time. Do not leave your customer to do this alone. In the Azure CSP model, you are responsible for customer support.

Migration process overview

After your customer is ready to make the change to Azure, there are really only three major steps left.

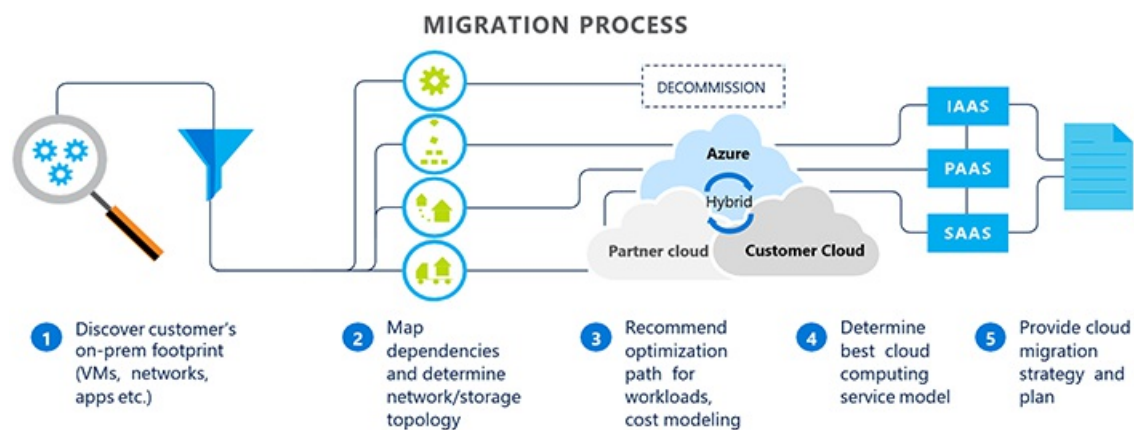
TIP

If your customer is not convinced yet, and doesn't understand the benefits of moving to Azure in CSP, see the [Why migrate from on-premises to Azure CSP deck](#).

Discovery: Work together with your customer to determine exactly what needs to be migrated. Because some of your customers might have complicated processes running on their on-premises servers, it's important to make sure you don't miss a vital component when carrying everything over in the migration.

Assessment: After you know what needs to be migrated, you must evaluate the processes and workloads, and determine how to migrate it to Azure CSP. This includes moving over a few small workloads to test how they work in Azure, and identifying any issues that might arise. In many cases, this is a straight-forward transfer, but sometimes the customer's existing system involves something that Azure CSP cannot do. It's good to know what these blockers are ahead of time, so you can account for them during the migration process.

Migrate: If any blockers appeared during the assessment step, you need to work together with the customer to restructure the architecture of their system. Your goal is to circumvent the blocking issues, producing the same results. After that is done, you can begin the process of transferring remaining workloads to Azure CSP.



Migration guides

For much greater detail about the process of migration, consult the following guides.

Note that whereas these guides are provided to help in the process, Microsoft offers no warranty, expressed or implied, in these guides.

Discovery

These resources are helpful when you're first engaging the customer and planning out the migration process:

- [Architectural design and considerations guide](#)
- [Crossing the chasm, part 1: Migrating the first workloads](#)

Assessment

After you have completed basic discovery, use these resources to help determine the best method for migrating, and to identify any issues that might arise:

- [Architectural design and considerations guide – advanced networking scenarios](#)
- [Crossing the chasm, part 2: Migrating additional workloads](#)

Migration

Use these guides when it is time to migrate the complete workload. They walk you through the use of the Azure Site Recovery tool.

- [Azure Site Recovery capacity planning guide](#)
- [Azure Site Recovery setup and configuration guide](#)

Additional Azure Site Recovery resources

These Azure websites might also be useful during the migration process:

- [Azure Site Recovery overview: What is site recovery?](#)
- [Azure Site Recovery in multi-tenant mode](#)
- [Migrate on-premises virtualized workloads to Azure using Azure Site Recovery](#)

These slide decks offer a simple look at the Azure CSP migration process and Azure Site Recovery:

- [Why migrate from on-premises to Azure CSP](#)
- [Azure Site Recovery training](#)

Next steps

- [Review](#) the Azure CSP model overview.
- [Learn more](#) about migration scenarios from other channels to Azure CSP.

Architectural design and considerations guide: Initial workload migration

9/19/2017 • 14 min to read • [Edit Online](#)

This article is primarily a mentor's guide to help technical-implementation consultants begin onboarding customers or organizations to Microsoft Azure Cloud Solution Provider (Azure CSP). The guide describes a typical process that a consultant would undergo to drive successful initial workload migration.

The article includes links to existing Microsoft technical step-by-step implementation resources and associated training materials. Such resources can help implementation consultants of varying Azure experience and expertise to navigate the process of migrating workloads to Azure. Use the article as a reference guide to Azure and the various options for migrating workloads from on-premises to Azure.

Create the virtual network and site-to-site VPN for the Azure virtual machines

One of the first steps an organization does in integrating an on-premises datacenter with Microsoft Azure is to stretch the on-premises datacenter network to Azure. You achieve this connectivity through a site-to-site VPN, which provides a security boundary so that the servers and workloads that are running in Azure are integrated with the stretched network of the on-premises datacenter. The network in Azure appears as part of the on-premises network, allowing servers or VMs in Azure to be joined to on-premises Active Directory and allowing servers and VMs in Azure to be patched, monitored, and managed as if they were within the on-premises datacenter.

A virtual network is required for the Azure virtual machines that are running in the cloud to cross a VPN gateway and connect back to the on-premises network. If a virtual machine in Azure is not part of a virtual network, or if the virtual network does not have a gateway, the Azure virtual machine still works but is unable to connect back to the on-premises network directly. Starting with step 1 of the Microsoft online article, [Create a virtual network with a site-to-site connection by using the Azure portal](#), the guide walks you through the creation of an Azure network and a gateway as well as the creation of the VPN tunnel.

IMPORTANT

- In step 1 of [Create a virtual network with a site-to-site connection by using the Azure portal](#), you define the virtual network's address space. Make sure that this process doesn't overlap your on-premises address space. For performance, choose an Azure region that reasonably corresponds to your on-premises datacenter.
- In step 3, you specify DNS servers. You can specify the IP addresses of machines across the VPN in the on-premises network, if that is the desired configuration. You can update the addresses later if a domain controller or other DNS server is placed in an Azure virtual network after it is created.
- In step 5, you create the VPN gateway. This process takes some time (about one hour), and it depends on what type of device is on the other end (on-premises) of the connection. Check this list for compatible and supported VPN devices. If you don't have any of those appliances available, you can do the same thing with remote routing and access services that are running on a Windows Server 2012 or later machine with a public IP address.
- In step 6, you configure the network on Azure so that the gateway knows what traffic to route. You'll need the public IP address of the VPN endpoint that you plan to use in the on-premises network, as well as the address spaces that will be routed to on-premises through the site-to-site VPN.

Migrate the first workload

Now that you've created an Azure virtual network and established a site-to-site link with the on-premises network, you can [migrate the first workload](#) to Azure CSP. For this example, a standalone Internet Information Services (IIS) web server will be migrated. After you've created the virtual machine in Azure, creating the web server is identical to creating an on-premises server.

IMPORTANT

- In the "Create the Windows virtual machine" section of the [Create your first Windows virtual machine in the Azure portal](#), use a size A1 virtual machine for this example.
- If the virtual machine is to be public facing, it needs a public IP and network security group to allow port 80/443 through on the public IP. Otherwise, you can omit the public IP. For information about network security groups and allowing access, see the "Open port 80" section in [Experiment with installing a role on your Windows VM](#). Follow these steps after you create the virtual machine in [Create your first Windows virtual machine in the Azure portal](#).

Follow the instructions in [Create a Windows virtual machine with the Azure portal](#) to create a Windows Server 2016 virtual machine. The article walks you through creating a virtual machine on Azure Resource Manager and networking the virtual machine.

- There are also some valuable training videos and guides on Microsoft Virtual Academy. For a training refresher and visual step-by-step walkthrough on the creation of virtual machines in Azure, see [Deploy Virtual Machines in the Cloud: Part 1](#) and [Deploy Virtual Machines in the Cloud: Part 2](#).
- If you want to create virtual machines through PowerShell, see [Create a Windows virtual machine using Resource Manager and PowerShell](#). You can refer to it to see the PowerShell syntax for the Resource Manager operations.

After you've created a virtual machine and it is online, it is accessible for remote access through the steps outlined in [Create a Windows virtual machine with the Azure portal](#). To allow connectivity for testing purposes, create a DNS entry for the virtual machine in your internal DNS server.

You can now install Windows Server roles and applications on the newly created virtual machine, such as installing IIS through Server Manager. For step-by-step documentation, see [Experiment with installing a role on your Windows VM](#). With IIS installed and running, you can push HTML code to the server either over the site-to-site VPN from the on-premises network using a simple `\servername\sharename` SMB connection or by copying it through the remote desktop session.

After you've installed the web application and code on the server, test the virtual machine to ensure functionality by doing the following:

- Go to the webpage from `localhost` (Azure virtual machine) to make sure that all the code was successfully copied and is functioning.
- Go to the webpage from across the site-to-site VPN from an on-premises machine by using the host name that you put into the DNS server previously to ensure that traffic is traversing the VPN correctly.
- *If the web server is public*, go to the webpage from the public IP address. You can collect the public IP address on the **Essentials** blade of your virtual machine when you select it. For more information, see [Experiment with installing a role on your Windows VM](#).

After the validation has succeeded, you can fail over the DNS name for the on-premises web server (www, web, and so on) to point to the Azure virtual machine. The migration is complete, and the Azure virtual machine, rather than the on-premises server, can now serve requests.

Confirm basic functionality in Microsoft Azure

Now that you have a virtual network and a site-to-site VPN, and the first workload is running in Azure, you can

explore other functions.

- **Access the virtual machine:** You can access the virtual machine directly through the Azure interface by using the Windows Remote Desktop method, as documented in [Create a Windows virtual machine with the Azure portal](#). Or you can use Remote Desktop to access it from a machine in your on-premises network by IP or host name if you have allowed for that in your virtual machine's firewall settings.
- **Patch the virtual machine:** Azure virtual machines need to be patched just as on-premises systems do. These virtual machines can work with your standard patching process (for example, Windows Server Update Service, System Center Configuration Manager, or standalone patching through Windows Update).
- **Monitor the virtual machine:** Azure virtual machines are monitored and managed just as the on-premises machines are. You can install agents (for example, [OMS](#) or [Datadog](#)) and report into the integrated monitoring solution.

Migrate the second workload (example: A simple, single-server web and SQL configuration)

Now that the first workload is working and you've covered some basic Azure functions, you can [migrate the second workload](#). This example is similar to the first workload, except that this server is an IIS server with an integrated SQL Server back end on the same virtual machine. Where building the virtual machine is concerned, follow the instructions for the first workload. This machine must be larger than that in the first workload because of SQL Server, so this machine should be an A3 Azure instance or later.

IMPORTANT

- In the [Create the Windows virtual machine](#) section of the previously mentioned guide, use a size A3 virtual machine for this example.
- Following the previously mentioned guide [Create a Windows virtual machine with the Azure portal](#), create a Windows Server 2016 virtual machine on the previously created virtual network.
- Connect to the virtual machine, install Microsoft SQL Server, and install the IIS role as you did in the first workload. For more information, see [Experiment with installing a role on your Windows VM](#).
- Back up the existing SQL database through SQL Server Management Studio. Copy the HTML code from the on-premises machine to the Azure virtual machine over the network. For more information, see [Back Up and Restore of SQL Server Databases](#).
- Restore the database in SQL Server running on Azure, and install the HTML code into IIS.

After you've mounted the database and installed the HTML code, test the virtual machine to ensure functionality by doing the following:

- Go to the webpage from localhost (Azure virtual machine) to make sure that all the code was successfully copied and is functioning.
- Go to the webpage from across the site-to-site VPN from an on-premises machine by using the host name that you put into DNS server previously to ensure that traffic is traversing the VPN correctly.
- *If web server is public*, go to the webpage from the public IP address. You can collect the public IP address on the **Essentials** blade of your virtual machine when you select it. For more information, see [Experiment with installing a role on your Windows VM](#).

Migrate the second workload (example: A two-server web and SQL Server configuration)

Another common scenario for a second workload that you might target for the migration to Azure is a two-server configuration that comprises a web server and a separate SQL server. In this example, an IIS server with a SQL back end will be migrated, but SQL Server will be on a separate virtual machine. Overall, this configuration is not much

more complicated than the first and second workloads we migrated. The only addition to the process is that you need to validate network connectivity between the two machines.

IMPORTANT

- In the [Create the Windows virtual machine](#) section of the previously mentioned guide, use size A2 virtual machines for this example.
- In the [Virtual machine creation guide](#), there is an option to build a virtual machine that already has SQL Server 2016 Enterprise installed. To save a step if your application already supports SQL Server 2016, choose this option instead of Windows Server 2016. However, for demonstrating Azure and for testing purposes, it is likely better to preserve parity with the on-premises application.
- This workload introduces a second way to migrate workloads to Azure. Both methods are viable, but the second method is more scalable to complex workloads and allows for failover without your needing to build a second set of servers.

Migration method 1

Following the previously mentioned guide, [Create a Windows virtual machine with the Azure portal](#), create two Windows Server 2016 virtual machines on the virtual network that you created previously.

On the virtual machine that you created for SQL, install the relevant SQL Server version to the app that's being migrated. Back up the database from the on-premises SQL Server and restore it onto this Azure SQL Server virtual machine.

On the virtual machine that you created for IIS, install IIS and copy the HTML code from the on-premises workload that's being migrated. For more information, see [Experiment with installing a role on your Windows VM](#).

With everything now copied to the Azure virtual machines, test the workload functionality by accessing the web server just as you did when you tested the first and second workload.

Migration method 2

With this migration method, you copy the virtual machines from on-premises to Azure by using Azure Site Recovery. You then fail over the virtual machines to Azure by using the test failover feature to start the virtual machines in Azure and validate their functionality. More setup is involved in this method, because you must prepare either a Hyper-V or VMware environment to copy the virtual machines to Azure.

[Site Recovery setup and configuration guides](#), along with guides for various virtualized environments, are part of the content framework that all program partners have access to. For more information about setting up Site Recovery in various environments, refer to those documents. Additionally, depending on the environment you are using, you can take advantage of the following articles:

- [Hyper-V without System Center Virtual Machine Manager (VMM)] (<https://docs.microsoft.com/en-us/azure/site-recovery/site-recovery-hyper-v-site-to-azure>)
- [Hyper-V with System Center VMM](#)
- [VMware VMs](#)

Follow the guide that suits your environment to replicate the virtual machines to Azure. The following example assumes that you are using a Hyper-V environment with VMM, so refer to that guide when reviewing the following steps:

1. Follow the [Replicate Hyper-V virtual machines in VMM clouds to Azure using Site Recovery in the Azure portal](#) guide, up to "Step 7: Test your deployment."

NOTE

- The test failover does not shut down on-premises virtual machines.
- If you're connecting virtual machines via the site-to-site network, we recommend that you shut down the on-premises virtual machines to avoid conflicts on the network. Do this at an acceptable time, such as during a maintenance window for testing of the application.
- If you're connecting the virtual machines to an isolated virtual network, you don't need to shut down the on-premises counterparts. However, it is likely that the app that you're testing needs access to authentication and other domain controller services. Thus, you'll need some extra configuration in Site Recovery to bring a domain controller into the test mix, and the failed-over domain controller will also need some light configuration to update the DNS entries of the newly failed-over workload.

2. Test the workload by accessing the web server and testing website functionality just as you did with the first and second workloads.
3. At the end of the test ([5c of Step 7](#)), the test virtual machines are cleaned up, and you can restart the on-premises virtual machines. Unlike the changes you make during a planned failover, the changes you make during your test are written back to on-premises.
4. When the test is successful, move forward with a planned failover to bring the workload live into Azure. The on-premises counterparts replicate with the active Azure instance to stay up to date.

Extend on-premises Active Directory to the Azure network

In a potential initial migration scenario where the servers are standalone systems, you might not need to add a Microsoft Active Directory domain controller in Azure if no Active Directory authentication is needed. However, as migrated workloads get more complex, you might need to join the servers to Active Directory, or you might need a domain controller in Azure.

To add a domain controller in Azure, simply create a Windows Server virtual machine. Be sure to match the appropriate version of Windows Server on-premises Active Directory domain controllers in the organization. Then promote the server to a domain controller. The process of creating a domain controller in Azure is as follows:

1. Follow the steps for the first or second workload earlier in this guide to build a virtual machine as a domain controller in Azure. Because this is a small test deployment, a size A2 virtual machine should be sufficient.
2. Follow the "Install AD DS on Azure VMs" and "Reconfigure DNS Server for the virtual network" sections in [Install a replica Active Directory domain controller in an Azure virtual network](#) to promote a virtual machine to a domain controller in Azure.

a. We recommend that you reserve a static IP address for the domain controller.

b. Use the following cmdlet. Update it with your virtual machine name (in place of *AzureDC1* in this example) and your IP address (in place of *10.0.0.4* in this example) in the virtual network:

- Test to see that the IP is available by using: `Test-AzureStaticVNetIP -VNetName "Vnet-Name" -IPAddress 10.0.0.4`

- Set the StaticIP and update the VM by using:

```
Get-AzureRMVM -ResourceGroupName "RG-Name" -Name "AzureDC1" | Set-AzureStaticVNetIP -IPAddress 10.0.0.4 | Update-AzureRMVM
```

Summary

Now you have one or two workloads and likely an Active Directory domain controller (potentially 3-5 total virtual machines) running in Microsoft Azure. Integrate these Azure virtual machines into their normal server and application administration and management process, which includes patching, monitoring, and ongoing maintenance.

With these initial workloads running in Azure, the next step is to quickly identify the next 12-15 servers that will be migrated to Azure, and continue migrating virtual machines to Azure via tools and automation.

Next steps

- Start [migrating the first workload](#) to Azure CSP.
- Review [Architectural design and considerations guide: Advanced networking scenarios](#).

Migrate initial workloads from on-premises to Azure CSP

9/19/2017 • 11 min to read • [Edit Online](#)

When a customer is considering migrating their on-premises systems to the cloud, it's important to start out by migrating just a few simple workloads. This tests the migration process, and helps increase customer confidence in the cloud services that you provide.

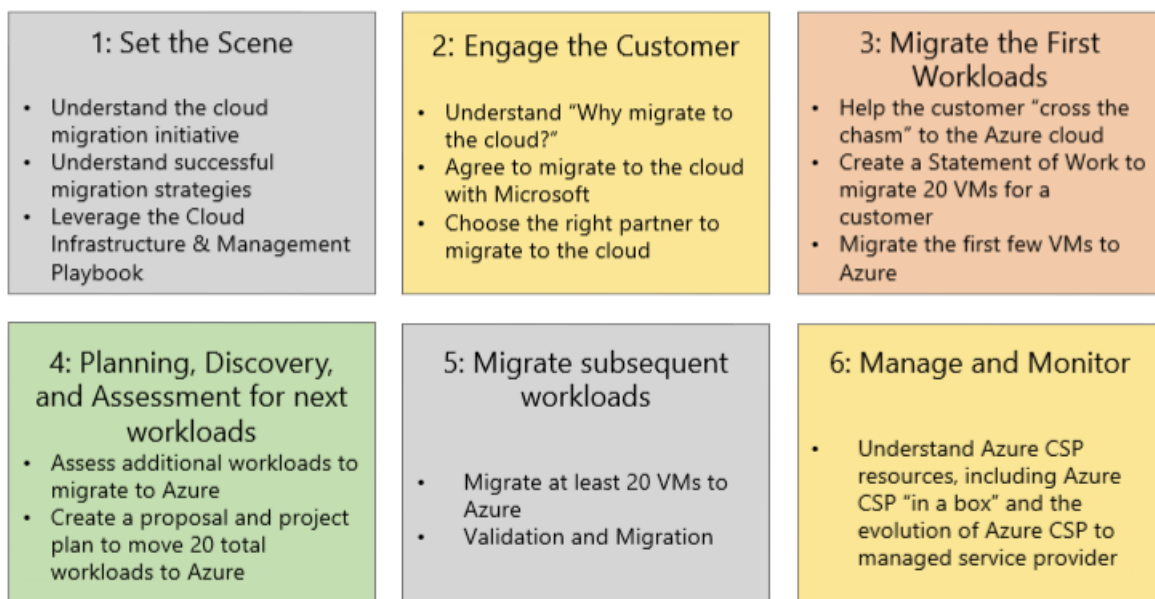
In this article, we describe how to migrate initial workloads for a customer. Initial workloads usually consist of between 3 and 5 virtual machines (VMs).

NOTE

This article covers just one of many possible migration methods for Azure. The method that we describe here is only a suggestion. If you think another method will work better for your purposes, or if you feel more comfortable with a different method, feel free to use that method.

Cloud migration timelines

The overall migration timeline is shown in the following figure:



Workload migration from on-premises datacenters to Azure typically occurs in two steps:

1. Migrate initial workloads
2. Migrate subsequent workloads

Initial workloads

For initial workloads, identify and migrate a handful of basic application servers to Azure. This is on an accelerated timeline, where a partner can help a customer migrate the initial 3 to 5 VMs to Azure within a week or two.

This initial experience helps the customer better understand Azure, and how administering and managing workloads works in Azure. It's also an opportunity for a customer to gain confidence in the reliability and dependability of Azure. The customer can extend the process of performing their usual, day-to-day tasks with

workloads that have been moved to Azure in the cloud.

Subsequent workloads

When a customer has a foothold of servers running in Azure and is comfortable with how Azure operates, the next step is to migrate another 15 to 20 additional VMs to Azure. These workloads typically consist of more complex applications. They might have more structure and downtime service-level agreements (SLAs) associated with them.

Even with subsequent workloads, partners should ensure that the 20 or so VMs that are migrated are expected to be the easiest applications or workloads as part of the initial extension of the enterprise datacenter to the cloud. For more information about migrating subsequent workloads, see [Migrating additional workloads from on-premises to Azure CSP](#).

Migration process (initial workloads)

The migration process for the initial workloads focuses on a few simple workloads (a maximum of 3-5 VMs) that are intentionally easy workloads to migrate.

Choose initial workloads

The selection of the initial workloads is critical to ensuring that the customer's first experience with migrations and Azure is not overly complicated. It's also important that learning and training exercises include giving the customer an opportunity to gain basic familiarity with Azure. Show the customer how to run VMs in Azure, and how to administer and manage Azure-based systems. One of the keys to a successful migration is giving the customer proof that they can manage Azure workloads with the same level of familiarity that they have been managing their on-premises systems.

Successful candidates for initial workload migration might include the following:

- **Standalone systems:** A server or application that is a standalone service. A standalone system is self-contained, with as few dependencies as possible. These qualities might make it the best candidate for an initial migration to Azure. This might be a standalone web server (like an internet web server) or a standalone website.
- **Single-node systems:** A server or application that is a single-node configuration, rather than a server that is part of a cluster, or a complex client/server application. This also makes a good candidate for an initial migration to Azure.

The initial workload might not seem strategic or complicated. But, for the customer who has had limited (or no) experience with Azure, the initial workload migration can help get the customer started with Azure. The goal is minimal effort, few challenges, and a high success factor.

For information about general application support in Azure (for example, the versions of Windows Server or Linux OS that are supported in Azure), see [Microsoft server software support for Azure virtual machines](#).

Confirm initial workload compatibility with Azure

After you've identified the initial candidate for a VM (workload) to migrate from an on-premises datacenter to Azure, the first step in the migration process is to run the [Microsoft Azure Virtual Machine Readiness Assessment tool](#).

The Azure Virtual Machine Readiness Assessment tool automatically inspects the on-premises environment, whether it is physical or virtual. It gives you a checklist and detailed report on steps you need to take to move the on-premises environment to the cloud. The Virtual Machine Readiness Assessment tool also validates the version of the operating system, running services, and the environment configuration, to make sure that the VM can run in Azure. An assessment report notes any areas that might require configuration or design changes.

If the assessment tool reports multiple items that require remediation, you might want to choose a different

(simpler) workload for the initial migration. Again, the purpose of the exercise of choosing the first workload to migrate is to get an initial VM running in Azure as quickly and as easily as possible.

Choose the best approach for migrating initial workloads

You have multiple options for getting the first workload to Azure:

- Reinstallation
- Export and import
- Azure Site Recovery

For a simple application like a web server with HTML pages, you can install a Windows Server VM in Azure. Then, install or copy the customer's HTML code onto that server. Many organizations have simple installation and reinstallation procedures documented for building a web server. This is why a web server application typically is recommended as the best first workload. We discuss this in the [Reinstallation approach](#) section.

For some migrations, it might be easier to export the entire server to a file, and then import the file as a server to Azure. This is noted in the [Export and import approach](#) section. A third approach is to use [Azure Site Recovery](#). Site Recovery replicates VMs between on-premises datacenters and Azure. This approach is a good method for later, automated VM migrations. But, for the first couple of workload migrations, we suggest that you choose an initial application and one of the two simpler migration approaches.

Reinstallation approach

A reinstallation approach might involve creating a new VM and installation, or restoring the application program and data onto a newly created VM. For some applications and for some organizational processes, this manual process might be familiar. The customer's IT team might prefer it. For example, if an application is a Microsoft Internet Information Services (IIS) web server, building a new Windows Server VM in Azure, enabling the IIS service on the server, and then installing the application code and data is something that many organizations are familiar with. They can successfully complete the process in an hour or two.

The advantage of this option is that the organization is following well-known and well-defined installation processes. Also, this installation method might lead an organization to install the latest Windows Server 2016 environment, potentially replacing an earlier Windows Server 2012 R2 or Windows Server 2012 configuration. The organization not only gets its IIS-based web application running in Azure, but also upgrades the underlying operating system so that it's running the latest Windows Server operating system.

The disadvantage of this method is that the organization must have all the application code, plug-ins, or other configuration settings to properly reinstall the application. Many organizations no longer have the installation discs or installation instructions to re-create installations from scratch. This might be a roadblock for using the reinstallation approach.

Export and import approach

Another migration option is to export an existing server configuration, and then import the entire configuration to Azure. This commonly is referred to as a P2V (physical-to-virtual) or V2V (virtual-to-virtual) migration. First, you use an export tool to export an existing (running) system. Then, you import the exported image to Azure.

The advantage of this type of migration is that it captures the entire system configuration, including the operating system, application, and data. This is a good solution for an organization that no longer has its installation disks, or that might not have the installation and configuration instructions to reinstall the application from scratch.

The disadvantage of an export and import approach is that the organization brings whatever was working (or not working) in their on-premises environment to Azure. If an application has an atypical configuration or operating issues, existing issues likely will be imported to Azure. So, potentially there might be problems with moving from on-premises to Azure.

However, for well-running, complex applications, the export and import approach allows a full migration of a

working application to Azure with little effort.

Azure Site Recovery approach

For organizations that believe the export and import approach of moving an entire application image to Azure might not be its best option, Azure Site Recovery simplifies the process. Site Recovery effectively takes a running physical server or VM (running in Microsoft Hyper-V or in VMware) and replicates the running system in Azure.

The advantage of using Site Recovery is that it provides scale via automation and by seeding the on-premises servers with full synchronization and continuous replication (in 30-second increments) until the organization is ready to start using the application in Azure. After seeding is implemented, an organization can choose when to perform the full migration. Site Recovery also has a test failover process that an organization can use to test its migration. Test failover helps to ensure that the end-state configuration works before using the application in a fully operational production state. With Site Recovery, you also can fail back a migration to the on-premises environment if there are problems in the initial move to the Azure cloud.

These are the disadvantages of using Site Recovery for migration:

- If a customer chooses this approach, they need to learn a new tool like Site Recovery. They will need to understand how to set it up and configure it. Microsoft offers step-by-step guidance on how to use Site Recovery. Despite a short learning curve, Site Recovery typically is the recommended method for helping an organization migrate their applications to Azure.
- Site Recovery has some specific requirements. To see the supported applications, configurations, and prerequisites for using Site Recovery, see the [Site Recovery support matrix](#).

As noted, using Site Recovery to migrate workloads to Azure is a focus for subsequent workload migrations. For the initial workloads, start with the simpler reinstallation or export and import methods.

Migrate the first workload to Azure

If a customer chooses the reinstallation approach, the process of migrating the first workloads might be as simple as building a VM in Azure and then installing the application code (and data) on the VM. For detailed procedures for building VMs in Azure, see [Architectural design and considerations guide: Initial workload migration](#). The guide includes detailed information about steps you can take if the on-premises network needs to be “stretched” to Azure to accommodate a hybrid integrated networking mode.

Azure workload administration and management knowledge sharing

After the first workload has been migrated to Azure, it's a good idea to make sure that the customer becomes familiar with administering and managing the VM that's now running in Azure. For the customer, having the knowledge and reassurance that they can perform usual, daily tasks in Azure that they might have been doing for years in their on-premises environment can help ensure acceptance of the move to Azure. This reinforces for the customer that running their application in Azure is “just the same” as running their applications on-premises.

These are some of the common management tasks to discuss with the customer:

- Remotely accessing the running VM (by using a Microsoft Remote Desktop client to connect to the server instance).
- Setting up the VM for patching and updates (by using Windows Software Update Services, System Center Configuration Manager, or another update tool).
- Monitoring the health and operations of the VM in Azure (by using System Center Operations Manager, or by using the Azure portal to get information about server operations).

Typically, this familiarization and knowledge-sharing process is done in a session after the first workload is migrated to Azure. The session might run from two to four hours. Because VMs that are running in Azure are similar to VMs that run on-premises, customers typically learn the process quickly.

For managed service providers that handle daily tasks, administration, and management, knowledge sharing might be just a one-hour overview with the customer. You can show them that the tasks that the partner will be doing for the customer are similar to what the managed service provider has done for the customer with their on-premises workloads. These activities can go a long way in easing any apprehension the customer might have. It also can build trust, so that the customer knows that the partner transferred their knowledge, expertise, and services to a similar platform with proven experience.

What's next?

After the first workload has been migrated to Azure, the next step is to immediately begin the process of migrating the second workload. The process of migrating the second workload is identical to the migration of the first workload. Again, select a workload that you anticipate will be easy to migrate. It might be a little more complicated, like potentially an IIS or web application, with an integrated SQL Server component.

Select a system that can be migrated with limited challenges. While the migration of the second workload is underway, it's important to simultaneously get ready to migrate the next 3 to 5 subsequent workloads. Avoiding a delay between the completion of the initial 3 to 5 VMs and the start of the next set of VMs helps ensure continuity and partner success.

Next steps

- Review the [advanced design considerations guide](#).
- Review [Migrate additional workloads from on-premises to Azure CSP](#).

Architectural design and considerations guide: Advanced networking scenarios

9/19/2017 • 7 min to read • [Edit Online](#)

At this point in the migration process, an organization has migrated the initial three to five simple workloads (virtual machines) to Azure. The organization is also familiar with Azure administration and management, and has run discovery and assessment tools to assess the next series of workloads that are going to be migrated to Azure.

With an additional 12-15 servers being targeted for migration to Azure, the organization might need to extend its initial network integration to Azure with a more advanced networking solution. In addition, more robust networking scenarios might be necessary to provide redundant connections to Azure to implement high availability, load balancing, integrated virtual networks, or Border Gateway Protocol (BGP) configurations.

Advanced networking solutions

With plans to migrate additional workloads to Azure, organizations typically want to improve redundancy and high availability. They want to exceed the redundancy and availability of the initial single virtual network that was established in the migration of the first few workloads to Azure.

Within Azure, you can connect virtual networks that span multiple Azure regions or have multiple connections from on-premises datacenters to an Azure virtual network. You can also advertise your Azure subnets to the on-premises network with your existing BGP. All these elements work together to increase the redundancy and availability of your Azure workloads.

Connect a virtual network to another virtual network

In the initial Azure site-to-site network connection, the network connection from the customer's primary datacenter to Azure was created, establishing an extension of the customer's infrastructure into Azure. Now, based on how geo-distributed your customer's infrastructure is, you need to account for either recreating or consolidating resources into corresponding Azure regions and virtual networks.

Following is an example of a geo-distributed on-premises infrastructure and the corresponding Azure virtual networks that must be created from a 1:1 standpoint (not from the standpoint of consolidating sites):

ON-PREMISES SITE	ON-PREMISES REGION	AZURE VIRTUAL NETWORK	AZURE REGION
DC1	Seattle, WA	SEA-VNET	West-US (WUS)
DC2	London, UK	LON-VNET	West-Europe (WEU)
DC3	Mumbai, India	MUM-VNET	West-India (WEI)

The next step is to create the remaining Azure virtual networks for DC2 and DC3, after which VNet-to-VNet peering must be established between the sites for internetwork communications as follows:

- SEA-VNET to LON-VNET
- SEA-VNET to MUM-VNET
- LON-VNET to MUM-VNET

If you want to have disaster recovery sites within Azure, then you need to ensure that the disaster recovery sites are configured the same way as your standard Azure sites with VNet-to-VNet peering established between them

for communications.

IMPORTANT

- If a virtual network and gateway has already been created or already exists, start at [Step 6](#) of the [Configure a VNet-to-VNet connection using the Azure portal](#) guide to create the secondary virtual network.
- A VNet-to-VNet connection does not have to span multiple Azure regions. However, if it does, it provides geo-redundancy.
- If only the first virtual network has a site-to-site VPN connection, the on-premises network can still access the second virtual network by way of the first virtual network's VPN gateway, though this is a single point of failure.
- Each virtual network can have only one virtual network gateway. Each gateway can have a maximum of 10 VPN tunnels, with the exception of high-performance gateways, which can have up to 30 tunnels. A VNet-to-VNet connection consumes a tunnel just as a site-to-site connection does. Keep this in mind when designing your network topology.

Following the [Configure a VNet-to-VNet VPN gateway connection using the Azure portal](#) guide, provision a second virtual network and virtual network gateway and connect it to the first virtual network. This guide assumes there is not an existing virtual network. This means that if you're using an existing network, such as the one from the start of this document, pick up at [Step 6](#).

After the networks are connected, you can migrate workloads to the second virtual network, which now offers high availability to the workloads in the first network. Workloads such as SQL Server AlwaysOn Availability groups can benefit from such a setup.

Configure BGP on an Azure VPN Gateway

Another common request that organizations make when they want to extend Azure is to tie the networking in Azure to the customer's existing BGP network configuration. BGP is a common routing protocol that advertises network segments and routes as part of a networking environment. With BGP configured and enabled, Azure routes and networks interact better with on-premises BGP advertisements and routing management.

Configuring BGP on your Azure VPN gateway enables Azure networks to advertise the routes into the on-premises network without having to add static routes. This is also a stepping stone to the next example: multiple and redundant gateways to Azure.

IMPORTANT

- The following guide uses Azure Resource Manager and Azure PowerShell. Instructions for adding the AzureRM module can be found at [Overview of Azure PowerShell](#).
- The [How to configure BGP on Azure VPN Gateways using Azure Resource Manager and PowerShell](#) guide specifies adding the Autonomous System Numbers (ASN) when the gateway is created. If you're working with an existing network, the ASN can be added to an existing gateway through [PowerShell](#). This cmdlet replaces item #2 of [Step 2 - Create the VPN Gateway for TestVNet1 with BGP parameters](#). See example 1 of the linked cmdlet for the exact syntax.
- Azure reserves certain ASNs that cannot be used by tenants in VPN gateways. Consult the [Azure BGP overview](#) to make sure the desired ASN is not reserved. The overview also contains a FAQ regarding the Azure BGP implementation.
- In the "Getting started with BGP on Azure VPN gateways" section of the [How to configure BGP on Azure VPN Gateways using PowerShell guide](#), follow the steps for configuring the virtual network for BGP. The virtual network that was created at the beginning of this document or the virtual network that you created earlier with the VNet-to-VNet connection can be reconfigured for BGP routing, and it does not require creating all new networks.
- After you've finished this configuration, check the advertised BGP routes in the on-premises router to be sure peering is successful and that the correct routes are advertised.

Configure a secondary gateway to Azure

Just as multiple virtual networks enable redundancy and even geo-redundancy on the Azure side of networking, configuring a secondary gateway in Azure allows for the interconnection to multiple on-premises networks. Multiple gateways increase the redundancy of the VPN connection as well as connecting the on-premises network

from multiple locations.

IMPORTANT

- Configuring Azure virtual network gateways to be in an active-active configuration requires the high-performance SKU.
- The [Configure active-active S2S VPN connections with Azure VPN Gateways using Azure Resource Manager and PowerShell](#) guide specifies adding the ASN when the gateway is created. If you're working with an existing network, the ASN can be added to an existing gateway through [PowerShell](#). This cmdlet replaces item #2 of [Step 2 - Create the VPN gateway for TestVNet1 with active-active mode](#). See example 1 of the linked cmdlet for the exact syntax.
- There are multiple ways to configure a secondary gateway to Azure. The overview in [Highly available cross-premises and VNet-to-VNet connectivity](#) covers the various methods and requirements. The example here uses the guide to set up an active-active configuration by using BGP and dual on-premises gateways for the highest reliability and redundancy.
- Following the [Configure active-active S2S VPN connections with Azure VPN Gateways guide](#), configure the virtual network gateway for high availability. As in the last section, the guide assumes that you are creating new networks. This is not necessary if they already exist. [Part 4](#) covers reconfiguring an existing gateway to be an active-active gateway by assigning the second IP and updating the SKU. From there, you can continue with [Part 1](#), Step 2, because the ASN still needs to be set along with the rest of the BGP configuration.
- After both gateways are configured on the Azure side and the on-premises side, this configuration will be both highly available and redundant. Traffic will be balanced between the four tunnels, though traffic state will be respected. There might be a performance increase, but the goal is availability more than performance.

Configure an Azure load balancer

In this scenario, it's likely that workloads need to be highly available. We recommend an Azure load balancer for distributing traffic among healthy instances. It can be configured to balance Internet traffic or internal traffic to the load-balanced set. In addition, specific monitors can be created for the load balancer to check the health of the members.

Following the [Create an internal load balancer using PowerShell](#) guide, deploy an internal load balancer and add network interfaces to workloads that are in need of high availability. Test connectivity from the on-premises network to the load balancer IP to ensure that local network machines can access the workload behind the Azure load balancer. Depending on how the workload handles load-balanced connections, you might need to adjust the distribution mode ("stickiness" or affinity) of the load balancer as detailed in [Configure the distribution mode for load balancer](#).

Summary

This advanced networking scenarios guide was intended to help an Azure architect with the more complex integration of on-premises networks with Azure networking, which is a common requirement when 10, 20, or more workloads are deployed into Azure. With a more robust network implementation, the organization can begin the task of migrating complex workloads such as SharePoint Server, Exchange Server, and SQL Server to Azure, by using tools such as Azure Site Recovery as the replication and migration solution.

Next steps

- Review [Azure Site Recovery capacity planning guide](#)
- Review [Azure Site Recovery setup and configuration guide](#)

Migrate additional workloads from on-premises to Azure CSP

9/19/2017 • 10 min to read • [Edit Online](#)

After you complete the [first migration](#) to test the migration process and increase customer confidence, you should then perform assessment and discovery for the next wave of migrations. These migrations usually involve multi-server dependent processes.

This article provides direction on migrating those workloads, usually composed of 12-15 virtual machines.

NOTE

This guide covers just one of many possible migration methods for Azure. The method provided here is just a suggestion. Feel free to use a different method if that works better for your purposes.

Migration planning summary, goals, and objectives

Migration from on-premises datacenters to Microsoft Azure minimizes an organization's need to continue to maintain and manage expensive datacenters. An application is still the same application, regardless of whether it is running inside an organization's datacenter, or hosted in another datacenter like Azure.

Azure services provide an organization with additional technology solutions that benefit the business. These solutions include:

- Datacenters geographically distributed around the globe to serve organizations anywhere in the world.
- Geo-redundancy of the Azure datacenter infrastructure, for high availability with business continuity and disaster recovery.
- Virtually limitless capacity to scale as organizations' needs and requirements grow.
- Resources needed are used on a pay "as needed" basis. Capacity can be decreased or eliminated without an organization incurring fixed capital costs for operations.
- Availability of integrated technology services enables organizations to become agile through digital transformation. These include backup, monitoring, management, data analytics, infrastructure as a service (IaaS), platform as a service (PaaS), mobile compute services, and Internet of Things.

Organizations of any size are no longer excluded from using enterprise-scale technology features and services. These organizations can extend their IT needs to meet core business objectives by enabling the services required by the organization.

Best practice: start with simpler workloads, progress to more complex

The applications and servers included in the next 15 (or greater) virtual machines being migrated to Azure may be more complex than those included in the first migration. Still, it is best not to pick the most complicated, most cumbersome, or most visible application the organization has for the next migration. Although it is great to tackle high visibility application migrations, Azure Cloud Solution Provider (Azure CSP) partners have been more successful with a series of migrations. Progressively build up to tackling the most strategic and complicated applications.

The migration process is an iterative learning process, both for the partner and the customer. For the partner, the learning process involves understanding the change management control processes the customer has in place. Then the partner can truly internalize the existing datacenter structure and operations, and establish a good

migration process between the customer's IT team and the partner's team.

For the customer, the learning process involves understanding how Azure works, and how they will administer and manage their applications when they are migrated to Azure. This includes establishing all networking, security, change controls, monitoring, and ongoing maintenance practices for their day-to-day operations.

This is primarily why partners should walk through the migration process with simple workloads to start the journey. It is during these cycles that all operational processes are updated, and IT personnel become familiarized with and trained in the hybrid cloud integration experience. From there, they can then take on more complex applications.

Select the subsequent workloads for migration

With this iterative migration process in mind, the next batch of 15 (or greater) virtual machines to be selected for migration would typically include:

- **Standalone front-end and back-end applications:** These might include Internet Information Services (IIS) web or SQL applications that have front-end and back-end server roles. These should be relatively independent applications, with limited integration with other servers and systems. This provides a good migration experience of moving servers to Azure.
- **Load-balanced applications:** Another good, relatively contained workload to migrate to Azure includes applications that have a load-balancing component to them, like web server farms. These workloads might involve migrating web servers and potentially implementing a virtual load balancer appliance on Azure.
- **Replicated applications:** These include applications that might have data replication between multiple servers, such as SQL Server running "AlwaysOn Availability Groups." In such cases, existing replication occurs between servers (on-premises or between sites), and an Azure site can now host the node of the replicated environment.

You might also migrate workloads such as contained SharePoint farms. These include load-balanced, front-end servers, and replicated back-end servers, with a combination of a handful of technical scenarios. Note that the organization may choose to migrate one or two smaller and simpler groups of servers to Azure, and then the last group of servers might be the more complicated SharePoint server farm.

In the cloud, there is no need to overbuy capacity. An organization can buy and add more capacity at any time. Thus, "rightsizing" servers and storage to what the organization needs is significantly more cost effective than overbuying capacity, and letting it sit idle and unused.

This is where a technical assessment of existing application workloads helps organizations determine exactly what they need. There are tools that run on existing systems, and over a two-week period collect and measure demand, performance, and capacity requirements. These tools can also provide growth projections, to help an organization select a cloud server and resource configuration that exactly meets their needs.

Application selection migration assessment

In the process of migrating applications from an on-premises environment to Azure, it is common to select which applications will migrate to the cloud and which applications may not be able to. It is not a foregone conclusion that everything will be migrated. You conduct an assessment process to better understand and validate the best workloads to migrate.

For example, consider an application that is no longer used or no longer useful for the organization, and is slated for retirement. Migrating this application to the cloud is a waste of time and money. Or, an organization might identify an application that needs to be upgraded. Rather than migrating an old or legacy application, and debugging it in the cloud, the organization might prefer to run an upgraded version of the application in the cloud. In other cases, an organization might decide to adopt a software as a service (SaaS) option, and therefore might not need to migrate the old application. So, the migration process is just part of a staged upgrade process in which the application might be upgraded while being migrated to the cloud.

You can also assess applications based on how easy or difficult it is to migrate them to Azure. There are a variety

of factors you might consider here. For example, limitations caused by application compatibility, or vendor support of an application running in the cloud. Also consider that some applications have specific hardware dependencies that cannot be replicated in a cloud environment, or other limitations like performance. When the assessment is completed and data has been gathered, partners can help customers categorize applications into the following groups:

- **Green:** Applications that can be readily moved to Azure without any engineering or modifications. Typically, these are Azure and Azure Site Recover validated Linux and Windows workloads. (Also see the "Technical Feasibility Guide" in section 4 of the program support portal.) These may include stateless applications and User Acceptance Test/Dev/Test labs.
- **Yellow:** Applications that require some level of reengineering or refactoring. They can be moved to Azure in a hybrid fashion, or physical roles can be moved to Azure PaaS solutions (as this requires an architecture overhaul).
- **Red:** Applications that cannot be moved to Azure due to legal, security, regulatory, business, or high cost of re-engineering reasons. These applications should be revisited later in the migration process.

Technical assessment tools

There are several tools available that gather runtime statistics of the application, and provide recommendations regarding a migration of virtual machines to Azure CSP:

- [Cloudamize](#)
- [RISC Networks](#)
- [ATADData](#)

The data gathered from such tools is usually consolidated into reports for assessment. The reports often include:

- Inventory. Identification of all the nodes (physical and virtual machines), and applications running on these nodes.
- Performance analysis. Collection of performance metrics, including recommendations for Azure virtual machine instance sizes and storage options.
- Infrastructure performance. Key metrics summarized by the tools include:
 - Peak CPU utilization
 - Allocated and peak RAM usage
 - Observed storage on-premises (capacity and current occupancy)
 - Disk IOPS and bandwidth
 - Throughput
 - Usage patterns (identifies how often compute and storage resources are on, idle, and unused)
- Azure configuration and cost projections. Projections on the recommended Azure virtual machine sizes (including CPU, memory, and disk storage) that correspond to on-premises servers. The projections also include anticipated monthly costs, according to the recommended workload configurations. Use this data to run projections and cost comparisons between existing operational cost outlays and the monthly cost to run the same workload in Azure.

Common assessment pitfalls

It is important to run an accurate cloud workload assessment and cost analysis. Organizations make mistakes that result in inaccurate total cost of ownership (TCO) and cost comparison analysis. Common pitfalls are:

- **Sizing and pricing overcapacity configurations:** Frequently, organizations simply take existing system configurations (for example, 16-core processors, 256 GB RAM, 10 TB storage) and price for the exact same configuration in Azure. But the on-premises server is only running at 5 percent or 10 percent utilization. This will result in pricing 10-20 times more expensive than needed to run the workload. Thus, the tools are helpful in accurately advising an organization that they only need, say, a 2-core machine with 26 GB RAM.
- **Storage misalignment:** Disk IOPS and bandwidth are too often overlooked when an organization is selecting

a cloud server configuration. Every type of disk has limitations for these measures, and underestimating your requirements can have a negative impact on performance. Under-sizing storage performance metrics can cause sluggishness of an application. The tools go beyond what you think you might need for storage performance requirements, and provide you a report of what it recommends the application needs in an Azure configuration.

- **Assessment over time:** The tools typically run for a period of two weeks, assessing peak loads and troughs over that time period. Simply using average performance numbers doesn't account for peak times, or doing an assessment at night that might not accurately reflect normal runtime conditions. Instead, the tools provide high and low ranges, as well as idle and busy time statistics.

Migrate subsequent servers to Azure

With the assessment complete, your next step is to begin the actual migration process.

Extend advanced networking integration components

Some organizations now may need to improve the simple site-to-site VPN network connection implemented for the first couple of workloads migrated to Azure. They may need to expand the networking to include Border Gateway Protocol routing, create connections between virtual networks, or configure additional gateways into Azure. For more guidance on advanced network connectivity, see the [Architectural design and considerations guide: advanced networking scenarios](#) article.

Use Azure Site Recovery for workload migration

Azure Site Recovery is a common tool to move complex workloads from an on-premises datacenter to Azure. Rather than rebuilding servers and installation applications from scratch or manually exporting and importing virtual machines, Azure Site Recovery replicates entire machine states from on-premises to Azure.

Azure Site Recovery encapsulates the underlying operating system, the applications, and the data as part of the replication process. The tool is built to support the migration of physical servers, virtual machines running on Microsoft HyperV, and VMware.

You can also configure Azure Site Recovery to bring together server dependencies. If a workload requires 8 servers to be migrated all at the same time, the tool can encapsulate, replicate, and migrate all 8 servers together.

Azure Site Recovery capacity planning, setup, and configuration

For details regarding capacity planning, setup, and configuration, see [Azure Site Recovery capacity planning](#) and [Azure Site Recovery setup guide](#).

These documents are step-by-step guides to set up and configure Azure Site Recovery services for migrating Hyper-V, VMware, and physical-based VMs and workloads to Azure. Specifically, these documents cover how to set up and configure the Configuration Server and Mobility Services components for on-premises environments. They also cover how to set up the Azure Site Recovery components in Azure to perform migrations.

Migration of specific workloads

Common workload migration scenarios are covered in the following documents:

- [Migrate SQL Server 2012 from on-premises to Azure CSP](#)
- [Migrate Dynamics AX 2012 R3 from on-premises to Azure CSP](#)
- [Migrate SharePoint Server 2013 from on-premises to Azure CSP](#)
- [Migrate SharePoint Server 2010 from on-premises to Azure CSP](#)
- [Migrate BizTalk Server 2013 R2 from on-premises to Azure CSP](#)
- [Migrate RedHat Enterprise Linux with WordPress site from on-premises to Azure CSP](#)
- [Migrate Oracle Enterprise Linux with Oracle Database from on-premises to Azure CSP](#)

These documents are detailed guides describing the steps for migrating common on-premises workloads, running

in different environments, to Azure CSP by using Azure Site Recovery.

Next steps

- Review [Azure Site Recovery capacity planning guide](#).
- Review [Azure Site Recovery setup and configuration guide](#).

Azure Site Recovery capacity-planning guide for migrations

9/19/2017 • 17 min to read • [Edit Online](#)

This guide covers how to complete Azure Site Recovery capacity planning for migrations to Azure Cloud Solution Provider (Azure CSP). Capacity planning is an important step that must be completed before migration can occur.

Prerequisites

Before you begin Azure Site Recovery capacity planning, ensure that the following tasks are complete:

- **Discovery** of source (on-premises) infrastructure details.
- **Planning, assessment, and cloud suitability** steps for all on-premises applications. Also, select the applications that are best suited for migration to Azure.
- **Azure Network Mapping** setup, to support the workload or application migration.

To get the *daily data change rate (%)* for Hyper-V VMs, your setup must meet the following requirements on the Hyper-V hosts:

- Operating system: Windows Server 2012/2012 R2 with Hyper-V enabled
- Memory: 20 MB in addition to total guest VM requirements
- CPU: 5 percent overhead (minimum)
- Disk space: 5-MB overhead (minimum)

Azure Site Recovery Capacity Planner tool

You should already have a source inventory report from the Discovery phase, which you generated by using either the Microsoft Assessment and Planning Toolkit or a third-party or custom solution. This report contains details of all the servers and VMs that are running workloads that you can now put into the Azure Site Recovery Capacity Planner tool.

NOTE

Ensure that, at minimum, you have supplied the following information for each server, VM, or application to be migrated. The information is required for accurate data computation:

- **Processor cores:** The total number of cores on the source server or VM.
- **Memory allocation (MB):** The RAM that's available to the source server or VM.
- **Number of NICs:** The number of active network adapters on a source server or VM.
- **Total storage (GB):** The total size of the storage. For example, if the source server has three disks with 500 GB each, the total storage is 1,500 GB.
- **Number of disks attached:** The total number of disks (not volumes) that are attached to a source server.
- **Disk capacity utilization (%):** The average disk capacity that's being utilized. If the source disk is 1 TB and 500 GB is being used, the utilization is 50%.
- **Daily change rate (%):** The daily data change rate (%) from the source server.

The Azure Site Recovery Capacity Planner is a macro-enabled Excel workbook. You can run it in either of two modes:

- **Detailed Planner:** Provides detailed information for each server in the source environment to get the final

assessment. We recommend this mode for medium to large enterprise migration projects.

- **Quick Planner:** Provides basic information about the source environment, based on input average. We recommend this mode for small enterprise migration projects.

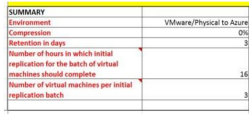
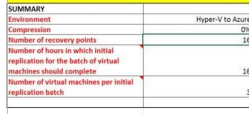
Detailed Planner

1. Download the [Azure Site Recovery Capacity Planner](#), and then open the **Site Recovery Capacity Planner.XLSM** worksheet.
2. To work with the file, select **Enable Editing**, and then select **Enable Content**. (You must have the current version of Office or Excel installed.)
3. On the **Getting Started** worksheet, in the **Select a planner type** list, select **Detailed Planner**.



4. You are directed to the **Workload Qualification** worksheet where, under **Summary**, you can select the appropriate environment. This selection changes the remaining properties slightly.

PROPERTY	DESCRIPTION	AZURE DATACENTER/PHYSICAL TO AZURE	HYPER-V TO AZURE
Compression	Generally refers to WAN compression.	30% is the default for Azure datacenter, and 0 is the default for physical servers. If you're using third-party WAN compression tools, specify the appropriate values.	Use 0 for no compression or use the respective values, based on the WAN compression solution that your enterprise uses.
Retention in days	The number of days for which you want to save the application recovery points in Azure after migration has begun.	3	N/A
Number of recovery points	The number of recovery points to be saved in Azure for your application when it is replicating.	N/A	16 (default, can be changed)
Number of hours in which initial replication for the batch of virtual machines should be completed		16 (default, can be changed)	16 (default, can be changed)
Number of virtual machines per initial replication batch		3 (default, can be changed)	3 (default, can be changed)

PROPERTY	DESCRIPTION	AZURE DATACENTER/PHYSICAL TO AZURE	HYPER-V TO AZURE
Example	A preview of the chart.		

5. Using the details from the source inventory (discovery) report, for each server, do the following:
 - a. Fill out the mandatory columns (in red), through **Disk capacity utilization (%)**.
 - b. Leave the **Daily data change rate (%)** column empty for now.
 - c. Either leave or delete the default values in the **Mapping Azure VM Size** column. These values will be computed later.
 - d. Fill out the additional columns (in gray), from **Operating System & Version** through **EFI Boot?**, and replace any default values if they are already present.

Server Name/Hostname	Processor Cores	Memory allocation (in MBs)	Number of NICs	Total Storage (in GBs)	Number of disks attached	Disk capacity utilization (%)	Daily data change rate (%)	Mapping Azure VM size (either fill or use the button above to compute)	Operating System & Version
BELSYSGLAB02	2	16384	1	500	1	50%			Microsoft Windows Server 2012 R2 Standard
BELSYSGLABVM01	2	4096	1	1000	2	35%			Microsoft Windows Server 2012 R2 Enterprise

6. Calculate the daily data change rate for servers or VMs, based on their type. For more information, see [Daily data change rate & calculation](#). Fill out the form for all corresponding servers.
7. After you've entered all server details, select **Computer IaaS VMs**. The macro runs and populates some columns:
 - **Mapping Azure VM:** Provides the corresponding virtual-machine SKU sizes in Azure.
 - **Is VM Qualified?:** Provides an update if the server or VM can be migrated to Azure.
 - To reveal the Azure Storage Account details, such as Standard or Premium, scroll to the right.
8. Select **Submit Data to Planner Tool**.
After you've submitted the data, a new worksheet called **Capacity Planner** is generated. It contains the assessment summary. To better understand the computed values, see [Understanding the Azure Site Recovery Capacity Planning Report](#).

Quick Planner

1. Download the [Azure Site Recovery Capacity Planner](#), and then open the **Site Recovery Capacity Planner.XLSM** worksheet.
2. To work with the file, select **Enable Editing**, and then select **Enable Content**. (You must have the current version of Office or Excel installed.)
3. On the **Getting Started** worksheet, in the **Select a planner type** list, select **Quick Planner**.



Azure Site Recovery
Planning Tool

Select a planner type

Quick Planner

4. On the **Capacity Planner** worksheet, you are *required* to provide *average values* about your source environment. The properties are as follows:

PROPERTY	DESCRIPTION
Select your scenario	Choose either Hyper-V to Azure or Azure datacenter/physical to Azure .
Total number of virtual machines	The number of servers (physical or VM) in the source environment.
Average number of VHDs per virtual machine	The average number of physical disks attached to each server (physical or VM).
Average size of VHD (in GBs)	The average size or total capacity of each physical disk.
Average utilization per disk (%)	Refer to Daily data change rate & calculation to collect this data and enter that data here.
Compression	Compression is offered with replication Azure datacenter VMs. The values here are typical but can be changed based on your Enterprise guidelines: <ul style="list-style-type: none">• Physical only: 0• Azure datacenter: 30%• Hyper-V: 0
Retention Inputs	Specifies how long replicas should be retained. For Azure datacenter or physical servers, the input value is expressed as days. For Hyper-V, the input value is expressed as hours.
Number of hours in which initial replication for the batch of virtual machines should complete and Number of virtual machines per initial replication batch	Enter data, expressed as hours, that you expect the Site Recovery infrastructure to replicate the entire initial data set to Azure.

INPUTS

Infra Inputs source	Manual
Select your scenario	VMware/Physical to Azure
Total number of virtual machines	2
Average number of VHDs per virtual machine	2
Average size of VHD (in GBs)	500
Average utilization per disk (%)	35%
Total data to be replicated (in GBs)	700
Churn Inputs	
Average daily data change rate (%)	5%
Amount of data changed per day (in GBs)	35
Compression	0%
Amount of data Xfered per day (in GBs)	35
Retention Inputs	
Retention in days	3
Initial Replication Inputs	
Number of hours in which initial replication for the batch of virtual machines should complete	16
Number of virtual machines per initial replication batch	3

Azure Site Recovery Capacity Planning report

Regardless of which planner type you use, the final assessment report consists of two sections:

- **Inputs:** Summarizes data of the source infrastructure, based on the details provided.

INPUTS

Infra Inputs source	Manual
Select your scenario	VMware/Physical to Azure
Total number of virtual machines	2
Average number of VHDs per virtual machine	2
Average size of VHD (in GBs)	500
Average utilization per disk (%)	35%
Total data to be replicated (in GBs)	700
Churn Inputs	
Average daily data change rate (%)	5%
Amount of data changed per day (in GBs)	35
Compression	0%
Amount of data Xfered per day (in GBs)	35
Retention Inputs	
Retention in days	3
Initial Replication Inputs	
Number of hours in which initial replication for the batch of virtual machines should complete	16
Number of virtual machines per initial replication batch	3

- **Output:** Displays the network bandwidth requirements, Azure storage account components, and Azure Site

Recovery infrastructure components that are *required* to support migrations to Azure.

OUTPUT

Network Bandwidth requirements	
Bandwidth required for delta replication (in Megabits/sec)	10 Mbps
Bandwidth required for initial replication (In Megabits/sec)	100 Mbps
Bandwidth refers to dedicated bandwidth for replication.	
Azure requirements	
Storage required (in GBs)	781.6666667
Total IOPS on standard storage accounts	342
Number of standard storage accounts required	1
Number of Blob disks required	4
Number of premium storage accounts required	0
Total IOPS on premium storage accounts	0
Other Infra requirements	
Number of Configuration Servers required	1
Number of additional Process Servers required	0
100% additional storage on the Source	NA

OUTPUT FIELD	DESCRIPTION
Bandwidth required for delta replication (in Megabits/sec)	Calculated based on average daily data change rate.
Bandwidth required for initial replication (in Megabits/sec)	Calculated based on initial the replication values provided.
Storage required (in GBs)	Total Azure storage required.
Total IOPS on standard storage accounts	Calculated based on 8,000 input/output operations per second (IOPS) unit size for the total standard storage accounts. This number is calculated from the total number of VMs that are mapped to standard Azure VMs, and the data change rate for those VMs.
Number of standard storage accounts	The total number of standard storage accounts needed to protect the VMs. A standard account holds up to 20,000 IOPS across all VMs, and a maximum of 500 IOPS per disk.
Number of Blob disks required	The number of disks that are created on Azure storage.
Number of premium storage accounts required	The total number of premium storage accounts needed to protect the VMs. A premium storage account can store up to 80,000 IOPS.
Total IOPS on premium storage	Calculated based on 256,000 IOPS unit size for the total premium storage accounts. This number is calculated from the total number of VMs that are mapped to premium Azure VMs (DS and GS series), and the data change rate on those VMs.

OUTPUT FIELD	DESCRIPTION
Number of configuration servers required	The number of configuration servers that are required for deployment.
Number of additional process servers required	Shows whether additional process servers are required, in addition to the process server that's running on the configuration server by default.
100% additional storage on the source	Shows whether additional storage is required in the source location.

Based on the report, you must meet the following requirements. You might need to set up new resources in Azure:

- The source + ISP network bandwidth must be capable of handling 100 Mbps or greater.
- Create Azure resource groups to bind migration-related Azure resources into manageable containers.
- Create one *standard* storage account in Azure.
- Set up one configuration server.
- *If necessary*, set up premium storage accounts.
- *If necessary*, set up additional process servers.

Network setup

Create Azure resource groups

1. Sign in to the Azure portal, select **New > Resource Groups**, and then select **Add**.
2. Type a meaningful name that's based on the convention determined in the planning phase.
3. Select the correct subscription and region details, and then select **Save**.
4. Repeat steps 1, 2, and 3 as needed if additional resource groups are required.

Create Azure storage accounts

1. Sign in to the Azure portal, and then select **New > Storage > Storage Account**.
2. Type an appropriate name for the storage account, and then set the following options:
 - **Deployment model:** *Resource Manager*
 - **Account kind:** *General Purpose*
 - **Performance:** *Standard or Premium*, based on capacity-planning results
 - **Replication:** *Geo-redundant storage (GRS)*
 - **Storage service encryption:** *Disabled*
 - **Resource group:** The resource group name that you created previously.
 - **Location:** The region where the resource group is located.

Create storage account

The cost of your storage account depends on the usage and the options [Learn more](#)

*

Name

asrstorageaccount

!

.core.windows.net

Deployment model

Resource manager

Classic

Account kind

General purpose

Performance

Standard

Premium

Replication

Locally-redundant storage (LRS)

*

Storage service encryption

Disabled

Enabled

*

Subscription

CSP_ACCELERATE_MIGRATION_OFFER_DE

*

Resource group

Create new

Use existing

SYSGLAB-PHYSICAL-RG

*

Location

West US

Set up configuration servers on-premises

Depending on the number of servers or VMs that you're migrating to Azure, use the following table to set up Windows servers to run configuration server components in the source environment. They must be Windows Server hosts for migrating to Azure.

CPU	MEMORY	CACHE DISK SIZE	DATA CHANGE RATE	NUMBER OF SERVERS TO MIGRATE TO AZURE
8 vCPUs (2 sockets * 4 cores @ 2.5 GHz)	16 GB	300 GB	500 GB or less	Fewer than 100
12 vCPUs (2 sockets * 6 cores @ 2.5 GHz)	18 GB	600 GB	500 GB to 1 TB	100 to 150
16 vCPUs (2 sockets * 8 Cores @ 2.5 GHz)	32 GB	1 TB	1 TB to 2 TB	150 to 200
Deploy another process server			>2 TB	Deploy additional process servers if you're migrating more than 200 machines, or if the daily data change rate exceeds 2 TB.

NOTE

- The server should have a static IP address.
- The server should not be a domain controller.
- The host name of the server should contain 15 or fewer characters.
- The operating system should be in English only.
- The configuration server needs Internet access. Outbound access is required as follows:
 - Temporary access on HTTP 80 during setup of the Site Recovery components (to download MySQL).
 - Ongoing outbound access to HTTPS 443 for replication management.
 - Ongoing outbound access on HTTPS 9443 for replication traffic (this port can be modified).
 - Access to the following URLs to connect to Azure:
 - *.hypervrecoverymanager.windowsazure.com
 - *.accesscontrol.windows.net
 - *.backup.windowsazure.com
 - *.blob.core.windows.net
 - *.store.core.windows.net
- If you have IP address-based firewall rules on the server, check to ensure that the rules allow communication to Azure. The rules must allow the Azure datacenter IP ranges and the HTTPS (443) protocol.

Set up source process servers

Depending upon the number of servers or VMs that you're migrating to Azure, set up Windows servers to run additional process servers in the source environment by using the following table:

CONFIGURATION SERVER	ADDITIONAL PROCESS SERVER	CACHE DISK SIZE	DATA CHANGE RATE	NUMBER OF SERVERS TO MIGRATE TO AZURE
8 vCPUs (2 sockets * 4 cores @ 2.5 GHz), 16 GB of memory	4 vCPUs (2 sockets * 2 cores @ 2.5 GHz), 8 GB of memory	300 GB	250 GB or less	Fewer than 85
8 vCPUs (2 sockets * 4 cores @ 2.5 GHz), 16 GB of memory	8 vCPUs (2 sockets * 4 cores @ 2.5 GHz), 12 GB of memory	600 GB	250 GB to 1 TB	85 to 150

CONFIGURATION SERVER	ADDITIONAL PROCESS SERVER	CACHE DISK SIZE	DATA CHANGE RATE	NUMBER OF SERVERS TO MIGRATE TO AZURE
12 vCPUs (2 sockets * 6 cores @ 2.5 GHz), 18 GB of memory	12 vCPUs (2 sockets * 6 cores @ 2.5 GHz), 24 GB of memory	1 TB	1 TB to 2 TB	150 to 225

NOTE

- The server should have a static IP address.
- The server should not be a domain controller.
- The host name of the server should contain 15 or fewer characters.
- The operating system should be in English only.
- The configuration server needs Internet access. Outbound access is required as follows:
 - Temporary access on HTTP 80 during setup of the Site Recovery components (to download MySQL).
 - Ongoing outbound access to HTTPS 443 for replication management.
 - Ongoing outbound access on HTTPS 9443 for replication traffic (this port can be modified).
 - Access to the following URLs to connect to Azure:
 - *.hypervrecoverymanager.windowsazure.com
 - *.accesscontrol.windows.net
 - *.backup.windowsazure.com
 - *.blob.core.windows.net
 - *.store.core.windows.net
- If you have IP address-based firewall rules on the server, check to ensure that the rules allow communication to Azure. They need to allow the Azure datacenter IP ranges and the HTTPS (443) protocol.

If it is required, set up additional process servers based on the capacity-planning recommendations. Also consider the following suggestions:

Depending on the number of servers or VMs that you need to migrate to Azure or for daily data change rates greater than 2 TB, you can either scale up your existing process server or scale out your process servers by adding more.

The way you scale your servers depends on your preference for a scale-up or scale-out model. You scale up by deploying fewer higher-end configuration and process servers, or scale out by deploying more servers with fewer resources. For example, if you need to protect or migrate 220 machines, you can do either of the following:

- Set up the configuration server with 12vCPUs and 18 GB of memory, set up an additional process server with 12vCPUs and 24 GB of memory, and configure protected machines to use the additional process server only.
- Configure two configuration servers (2 x 8vCPU, 16 GB of RAM), set up two additional process servers (1 x 8vCPU and 1 x 4vCPU to handle 135 + 85 [220] machines), and configure protected machines to use the additional process servers only.

Calculate the daily data change rate

Physical servers

Unfortunately, there is currently no guidance for calculating the daily data change rate for physical servers. However, there are many third- party solutions to complete this step for your source (physical) servers. Azure architects must work with system administrators and other stakeholders to estimate the rates from surveys and

discussions.

Hyper-V

NOTE

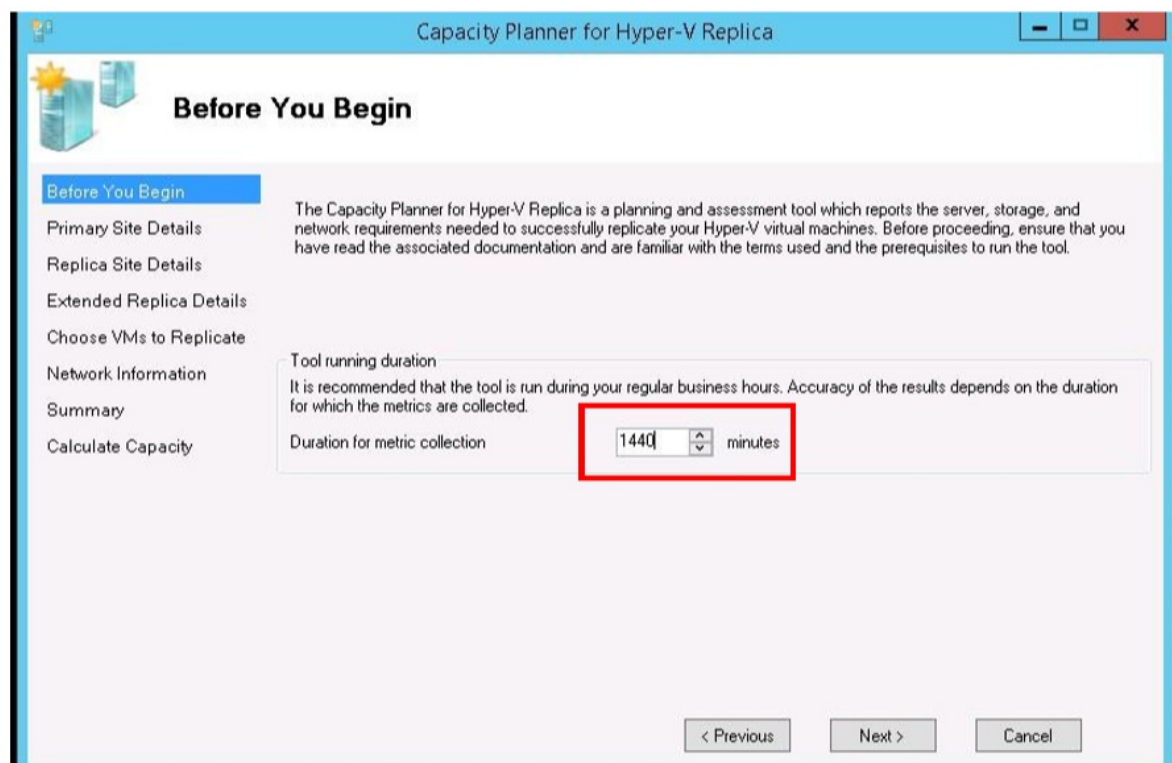
The daily data change rate cannot be calculated for VMs if they:

- Have already been enabled for replication.
- Are not *running*.
- Are running on a remote standalone server with their virtual disks residing in SMB shares. The error you encounter is related to constrained delegation. For more information, see [Using SMB shares with Hyper-V Replica](#).

1. Go to the [Hyper-V Capacity Planning tool](#), select **Download**, select **HVRCapacityPlanner.Exe**, and then save the file to a known location, such as [C:\Temp] for each Hyper-V host from which VMs will be migrated to Azure.
2. On the Hyper-V host, run **HVRCapacityPlanner.exe**, select **Yes**, accept the license terms, and then select **Next**.
3. Leave the **Duration for metric collection** default setting at **1440 minutes**, because you need to calculate the rate daily and over an entire day.

NOTE

We highly recommend that you run the tool during production hours, a practice that ensures that you're collecting the most representative data.



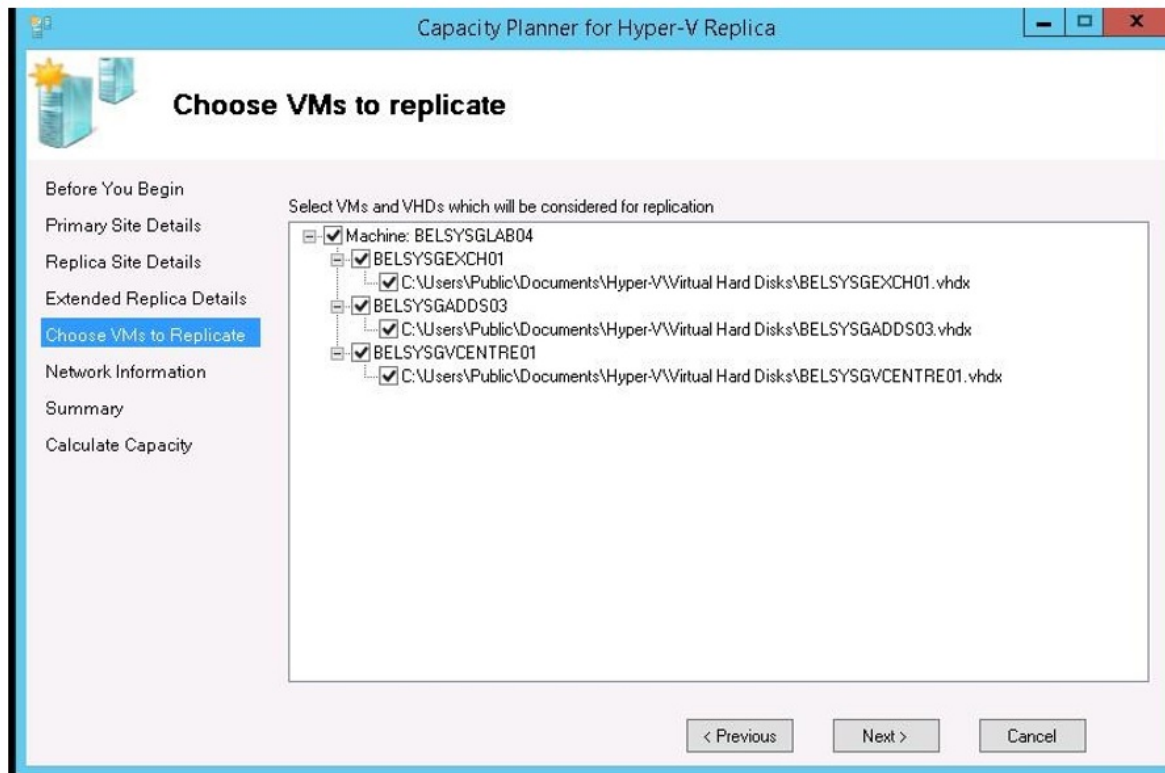
4. On the **Primary Site Details** page, enter the **FQDN** or **NetBIOS** names of the Hyper-V hosts that you want to collect data from, and then select **Add**.

5. On the **Replica Site Details** page, select the **Skip tests involving replica site** check box, because the migration is to Azure and not to another onsite replica.

6. Select **Next**, and then select **Next** again.
7. When you get a warning that only VMs that are *not* enabled for replication are displayed, select **OK**.
8. Choose the VMs to collect data for, and then select **Next**.

NOTE

- If you plan to exclude a virtual disk, clear the check box next to the VM name so that the capacity-planning guidance is reported correctly.
- If you plan to exclude a full VM, make sure the VM and all its virtual disks are unchecked.



9. On the **Network Information** page, select **Next**.
10. Review the **Summary** to ensure that the correct properties are selected, and then select **Next**.
11. The tool now captures the metrics in the background for the time selected (30 minutes). After the report is completed, select **View report**.
12. On the generated report, compute the daily data change rate as follows:
 - a. In the **Additional Storage during delta replication (GB)** column, note the value.

Virtual Machine	VHD Size (GB)	Additional Storage during delta replication (GB)
BELSYSGEXCH01	42.13	0.06
BELSYSGADDS03	9.04	0

- b. Open the **ASR Capacity Planning report**, select the same server, and then note the value in the **Total Storage (in GBs)** column.

Server Name/Hostname	Processor Cores	Memory allocation (in MBs)	Number of NICs	Total Storage (in GBs)	Number of disks attached	Disk capacity utilization (%)	Daily data change rate (%)	Mapping Azure VM size (either fill or use the button above to compute)
BELSYSGEXCH01	2	8096	1	42.1	1	20%		

- c. Calculate the daily data change rate as **[additional storage during delta replication / total storage * 100]**. For example: $[(0.06 / 42.1) * 100] = 0.142\%$
 - d. In the Azure Site Recovery Capacity Planning tool, in the **Daily data change rate (%)** column, enter the final value (for example, *0.142*), and then repeat these steps for any other VMs.

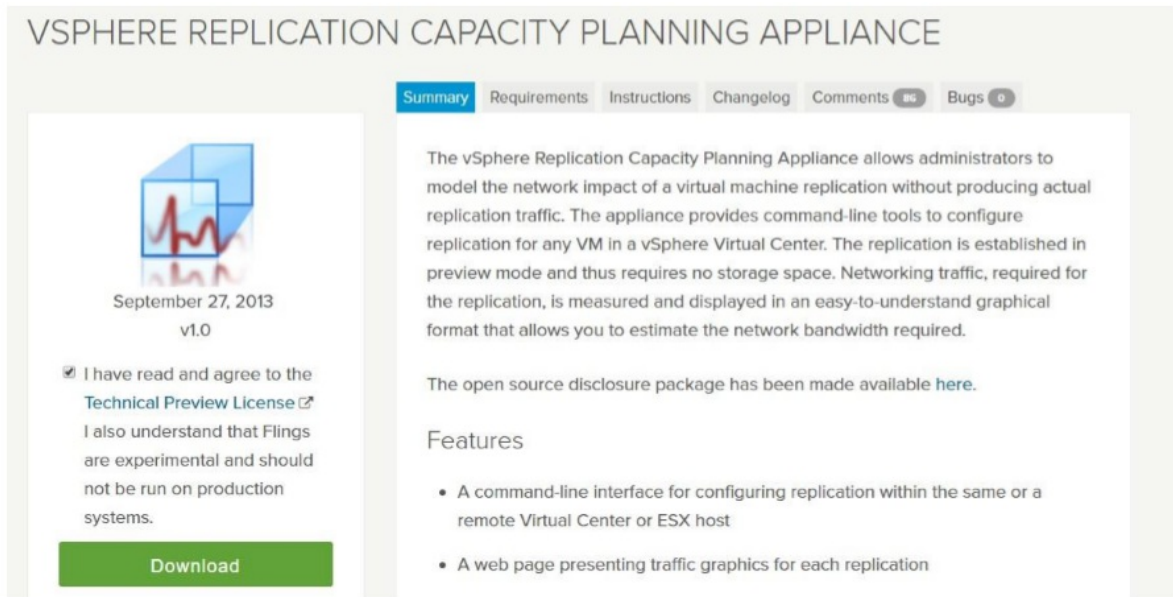
NOTE

Any values less than 1% are automatically converted to 0.

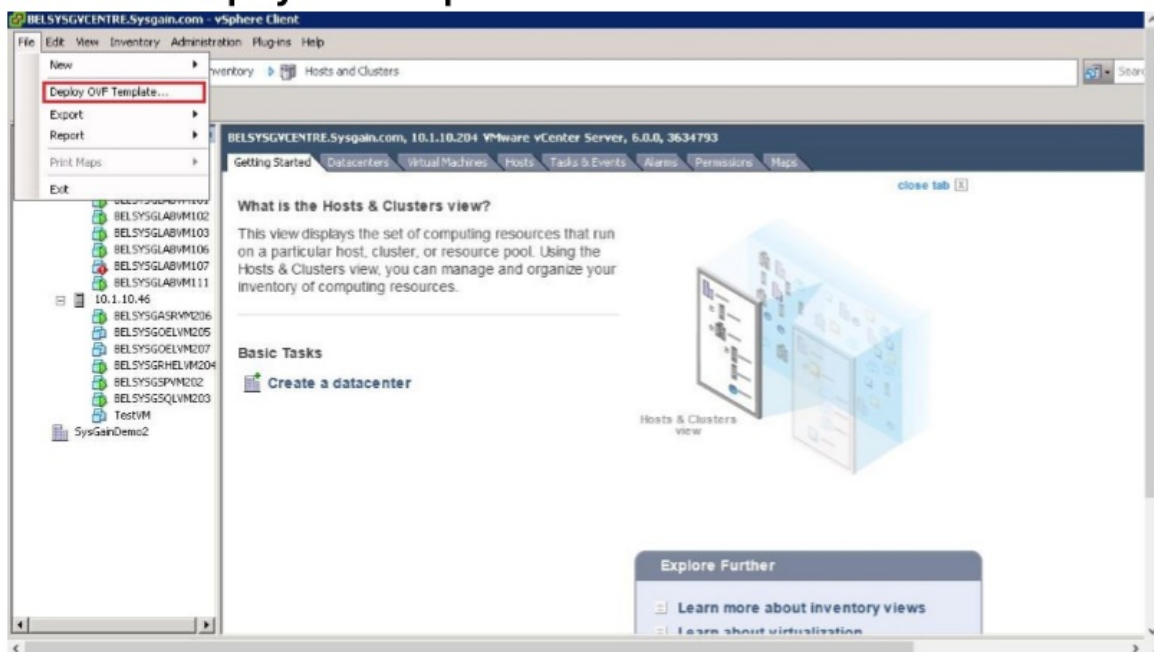
13. Repeat the previous steps for all Hyper-V VMs.

Azure datacenter

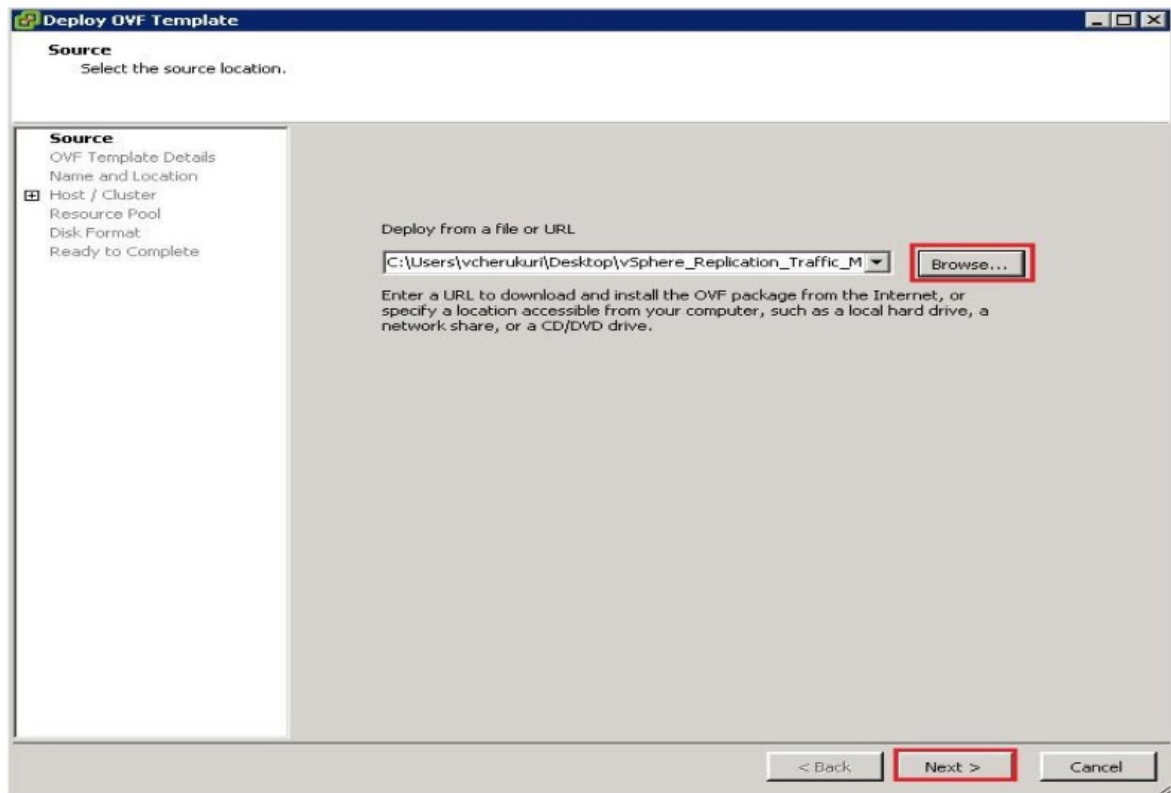
1. On a server that has the vSphere client installed, go to the [vSphere capacity planning appliance](#), download the tool to a known location, and then extract the files to a known folder.



2. Connect to your **VCenter** server by using the **vSphere Client**.
3. Select **File > Deploy OVF Template**.



4. Select **Browse**, select **vSphere_Replication_Traffic_Modeler_OVF10.ovf** at the download location, select **Next**, and then select **Next** again.



5. On the **Name and Location** page, enter a suitable name, select the datacenter and possibly the ESXi host where you want to run and monitor the VMs, and then select **Next**.

Deploy OVF Template

Name and Location
Specify a name and location for the deployed template

[Source](#)
[OVF Template Details](#)
Name and Location
[Disk Format](#)
Network Mapping
IP Address Allocation
Properties
Ready to Complete

Name:
TrafficModeller1
The name can contain up to 80 characters and it must be unique within the inventory folder.

Inventory Location:
SysGainDemo

< Back Next > Cancel

6. Choose a suitable datastore, and then select **Next**.

The screenshot shows the 'Deploy OVF Template' wizard window. The title bar is blue with standard window controls. The main content area is white. On the left, there is a vertical sidebar with a list of steps: 'Source', 'OVF Template Details', 'Name and Location', 'Disk Format' (which is highlighted in bold), 'Network Mapping', 'IP Address Allocation', 'Properties', and 'Ready to Complete'. The main area has a heading 'Disk Format' and a sub-heading 'In which format do you want to store the virtual disks?'. Below this, there are two input fields: 'Datastore:' with the value 'SG2' and 'Available space (GB):' with the value '33.3'. There are three radio button options: 'Thick Provision Lazy Zeroed' (which is selected), 'Thick Provision Eager Zeroed', and 'Thin Provision'. At the bottom right, there are three buttons: '< Back', 'Next >', and 'Cancel'.

Deploy OVF Template

Disk Format
In which format do you want to store the virtual disks?

Source
OVF Template Details
Name and Location
Disk Format
Network Mapping
IP Address Allocation
Properties
Ready to Complete

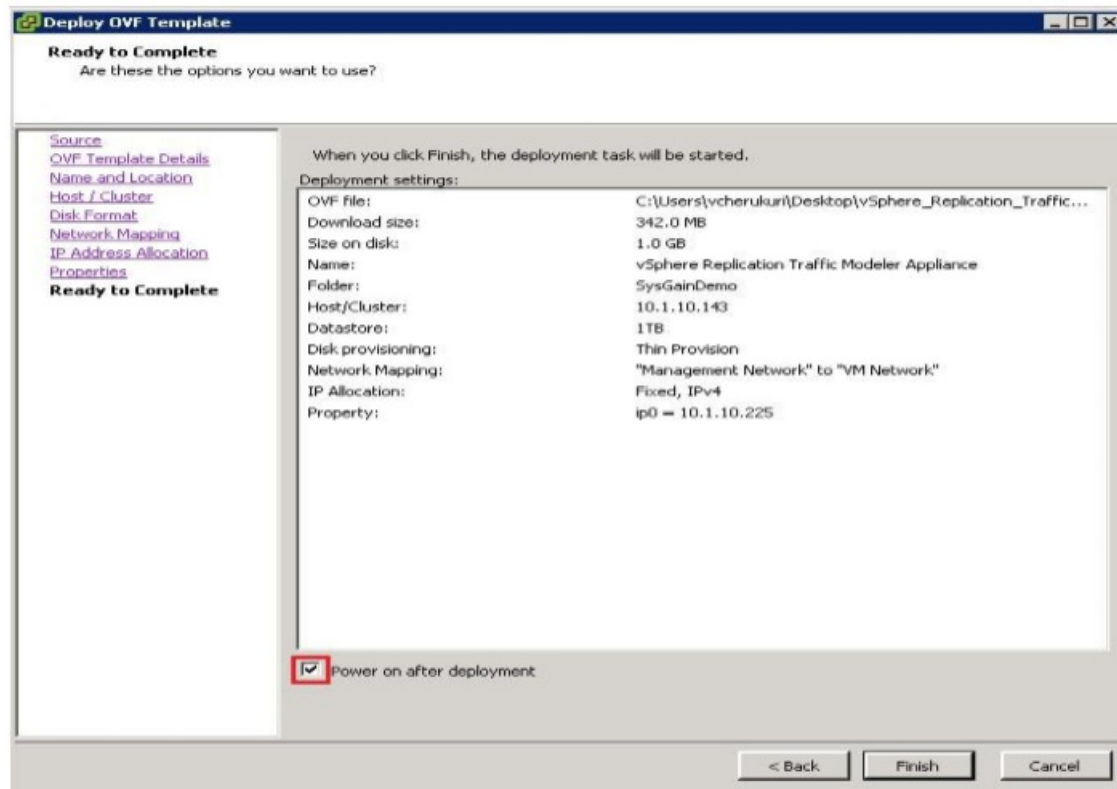
Datastore: SG2

Available space (GB): 33.3

☒ Thick Provision Lazy Zeroed
☐ Thick Provision Eager Zeroed
☐ Thin Provision

< Back Next > Cancel

7. On the **Network Mapping** page, select a suitable network profile, and then select **Next**.
8. Set **IP address allocation** to **StaticIP**, and then select **Next**.
9. On the **Ready to Complete** page, select the **Power on after deployment** check box, and then select **Finish**.



10. After the VM is deployed, connect to the appliance via SSH (using PuTTY or a similar tool), and sign in by using the default credentials:

- **Username:** *root*
- **Password:** *Azure datacenter*

11. After you are signed in, run the following commands:

```
cd /opt/Azure datacenter/hbrtraffic/bin
./configureReplication -vc <vcenter server address> --vcuser <vcenter username> --vcpass <vcenter password> --lwd
<IP_of_the_Appliance> --vmname <VM_name>
```

- <vcenter server address>: The IP address of the vCenter server
- <vcenter username>: The vCenter username
- <vcenter password>: The vCenter password
- <IP_of_the_Appliance>: The IP address of the vSphere replication traffic-modeler appliance
- <VM_name>: The name of the VM that you want to monitor or collect data from

NOTE

To enable monitoring of other VMs that are managed by this vCenter server, repeat the preceding command by changing <VM_name>.

For additional help with these commands, enter **./configureReplication -help**.

12. After you have enabled capacity planning for the VMs, you can view their statistics by going to **<https://<IP address of capacity planning appliance>:5480/vr-graphs/>**.

NOTE

We recommend that you monitor during production hours for at least 24 hours to capture quality metrics. Please allow at least 15 to 20 minutes for initial data captures to begin.

- The website displays a list of VMs. To open a report, select a VM name.



The report has two columns: **LWD (Light Weight Delta) Network Traffic** and **Delta Size**.

- **LWD In:** The network traffic that's generated by the VM replication data.
- **LWD Out:** The network traffic that's generated by the replication appliance, mostly acknowledgments and status information/updates, as well as checksums.
- **Delta Size:** The change rate of the VM size.

- To compute the daily date change rate for a server, do the following:

a. In the Azure datacenter server report, get the **Average delta size computer for 1**.

b. If the average delta size is expressed as GB, use the value as is for the next step. However, if it is expressed as MB, you must first convert it to GB by dividing by 1,000. For example, 10.78 MB = $[10.78/1000] = 0.0178$ GB.

c. Open the **ASR Capacity Planning report**, select the same server, and then note the value in the **Total Storage (in GBs)** column.

Server Name/Hostname	Processor Cores	Memory allocation (in MBs)	Number of NICs	Total Storage (in GBs)	Number of disks attached	Disk capacity utilization (%)	Daily data change rate (%)	Mapping Azure VM size (either fill or use the button above to compute)
BELSYSSEXCH01	2	8096	1	42.1	1	20%		

d. Calculate the daily data change rate as $[Additional\ Storage\ during\ delta\ Replication / Total\ Storage\ (in\ GBs) * 100]$.

e. In the Azure Site Recovery Capacity Planning tool, in the **Daily data change rate (%)** column, enter the preceding value, and repeat the steps for any other VMs.

NOTE

Any values less than 1% are automatically converted to 0.

Next steps

- Review [Azure Site Recovery setup and configuration guide](#).
- Start [migrating your first workloads](#).

Azure Site Recovery setup guide for migrations

9/19/2017 • 18 min to read • [Edit Online](#)

This guide covers how to migrate all relevant data from the on-premises environment to Azure Cloud Solution Provider (Azure CSP) by using Azure Site Recovery. Specifically, it covers the setup and configuration of the configuration server, process server, and mobility services components of the on-premises environment. It also covers setting up the Site Recovery components in Azure to perform migrations.

What is Azure Site Recovery?

Site Recovery is a cloud service that provides management and orchestration for disaster-recovery scenarios. When you've set up and configured Site Recovery properly, you can use it to migrate on-premises workloads (virtualized or bare metal environments) to Microsoft Azure.

To learn more about Site Recovery before you begin the process, see the "Additional Azure Site Recovery resources" section of [Migrate on-premises workloads to Azure CSP](#).

Prerequisites

Before you begin Site Recovery setup, ensure that *all* of the following tasks are complete:

- **Discovery** of source (on-premises) infrastructure details, including the migration of the [first](#) and [second](#) test workloads.
- **Planning, assessment, and cloud suitability** steps for all on-premises applications. For architectural design tips, see the [Architectural Design and Considerations Guide – Advanced Networking Scenarios](#) page.
- **Azure Network Mapping** setup, to support the migrated applications post-failover.
- **Site Recovery capacity planning**.

You must also provide a Windows Server 2012 R2 server that you can use as the configuration or process server. We recommend high availability, but it is not required. The server must meet the following requirements:

- The server is domain joined and has a static IP.
- The operating system is English only.
- The server does *not* have the following services or roles installed:
 - Active Directory Domain Services
 - Internet Information Services
 - Hyper-V
- The server has vSphere PowerCLI 6.0 installed, if the infrastructure contains VMware VMs that also need to be migrated to Azure.
- The server has internet access (inbound and outbound), with the following additional requirements:
 - Outbound internet access on HTTP Port 80 for Site Recovery setup. You can turn off the access after setup is complete.
 - Outbound internet access on HTTPS port 443 for replication management. You cannot change the port.
 - Inbound internet access on HTTPS port 9443 for replicating traffic to Azure. If your IT team wants to implement it, the port can be a custom port.
 - To communicate with Azure, the server must be able to access the following URLs:
 - *.hypervrecoverymanager.windowsazure.com
 - *.accesscontrol.windows.net

- o *.backup.windowsazure.com
- o *.blob.core.windows.net
- o *.store.core.windows.net
- o <https://cdn.mysql.com/archives/mysql-5.5/mysql-5.5.37win32.msi> (not required for a scale-out process server)
- o time.nist.gov
- o time.windows.com

VMware setup

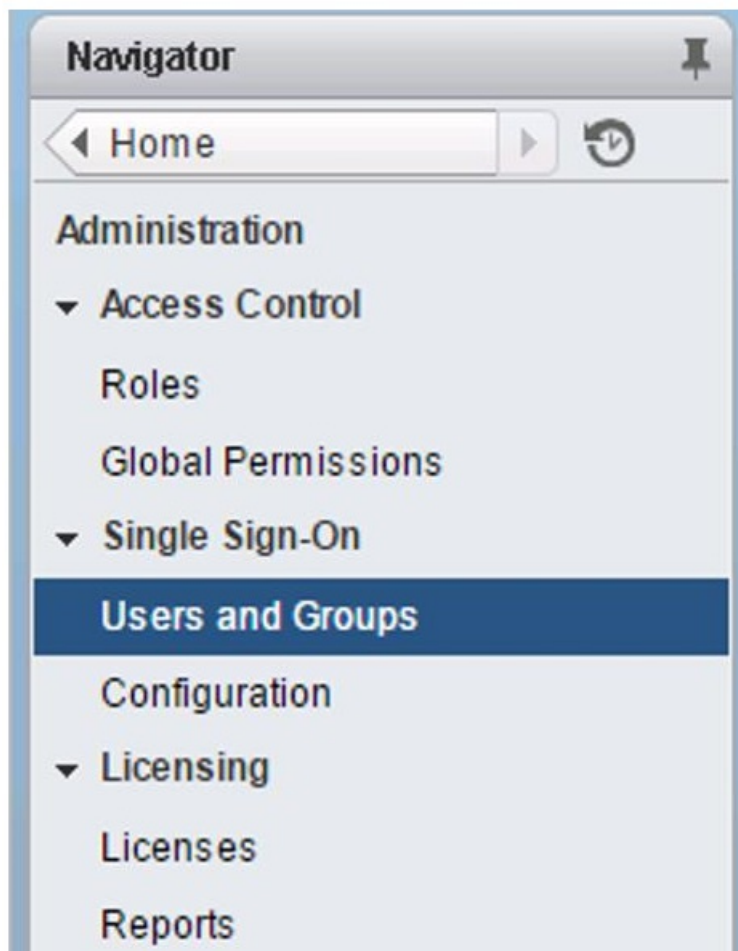
If your on-premises environment contains a VMware infrastructure and your migration plan includes VMs that are hosted on this infrastructure, the environment must meet certain requirements before you can migrate the VMs to Azure by using Site Recovery. *If you are not migrating any VMware VMs, skip this section.*

- The VMware hosts should be running a supported version of ESXi with the latest updates installed.
- At least one vCenter server *must* be deployed to manage the VMware hosts with the latest updates.

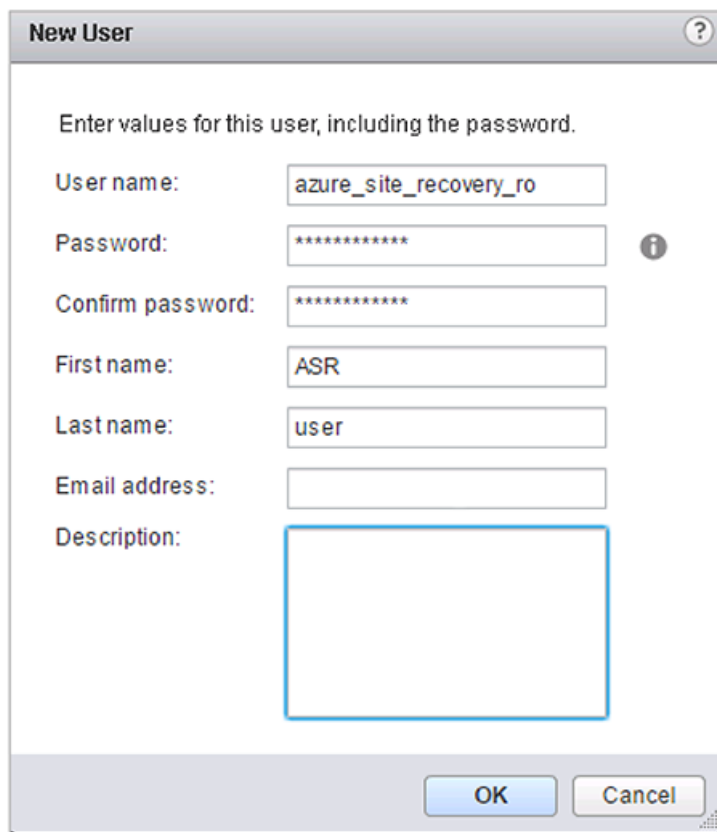
Prepare the VMware account for automatic VM discovery

The Site Recovery process server can automatically discover VMware VMs from a vCenter server that manages hosts. To perform automatic discovery, Site Recovery needs credentials that can access VMs on the vSphere servers. To create a dedicated account for automatic discovery of VMs at the vCenter level, do the following:

1. Sign in to your vCenter server.
2. Go to **Administration > Users and Groups**.



3. Create a user account according to your company policies.



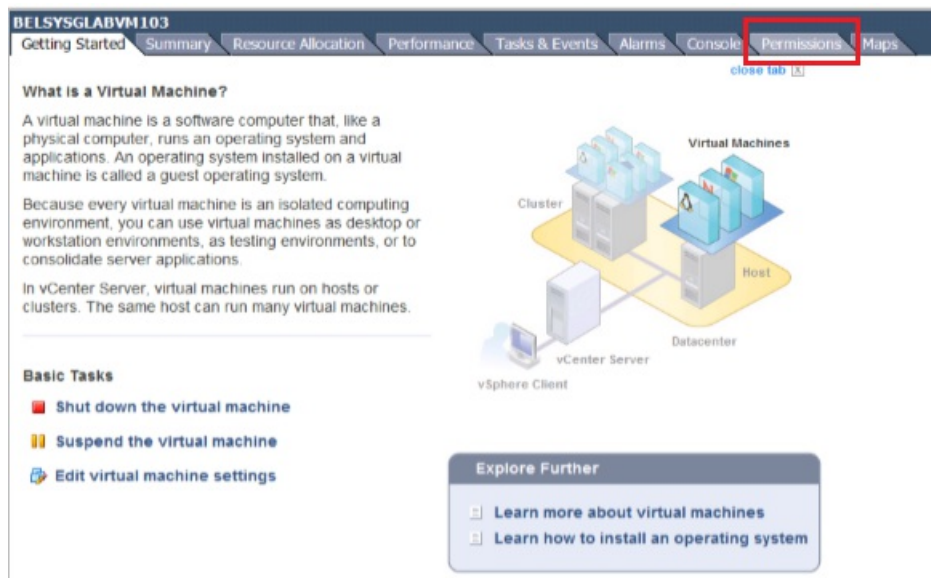
A dialog box titled "New User" with a question mark icon in the top right corner. It contains a message: "Enter values for this user, including the password." Below this are several input fields: "User name:" with the value "azure_site_recovery_ro", "Password:" with masked characters "*****", "Confirm password:" with masked characters "*****", "First name:" with the value "ASR", "Last name:" with the value "user", "Email address:" (empty), and "Description:" (a large empty text area). An information icon (i) is located to the right of the password fields. At the bottom are "OK" and "Cancel" buttons.

4. Using a vSphere client, connect to either the vCenter server or the VMware host, and then grant the new account access to the VMs.

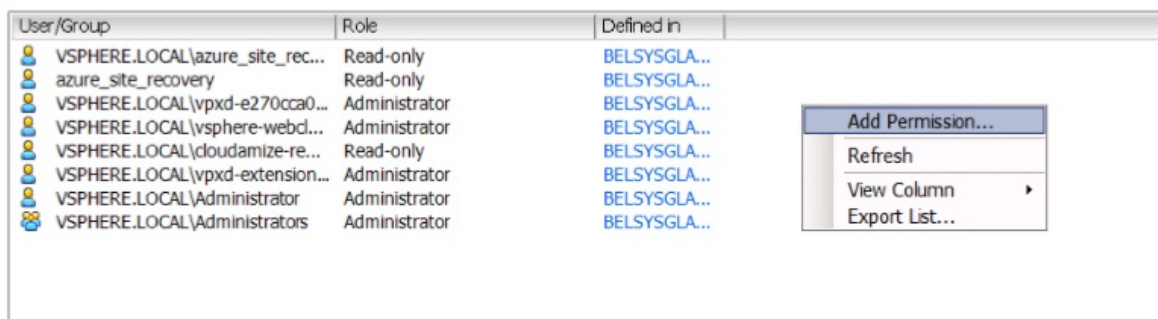


A screenshot of the VMware vSphere Client login dialog. The title bar says "VMware vSphere Client". The main area has the VMware logo and the text "VMware vSphere Client". Below this is a message: "All vSphere features introduced in vSphere 5.5 and beyond are available only through the vSphere Web Client. The traditional vSphere Client will continue to operate, supporting the same feature set as vSphere 5.0." There are two instructions: "To directly manage a single host, enter the IP address or host name." and "To manage multiple hosts, enter the IP address or name of a vCenter Server." Below these are input fields: "IP address / Name:" with the value "10.1.10.123", "User name:" with the value "azure_site_recovery_ro@vsphere.local", and "Password:" with masked characters "*****". There is a checkbox labeled "Use Windows session credentials" which is unchecked. At the bottom are "Login" and "Close" buttons.

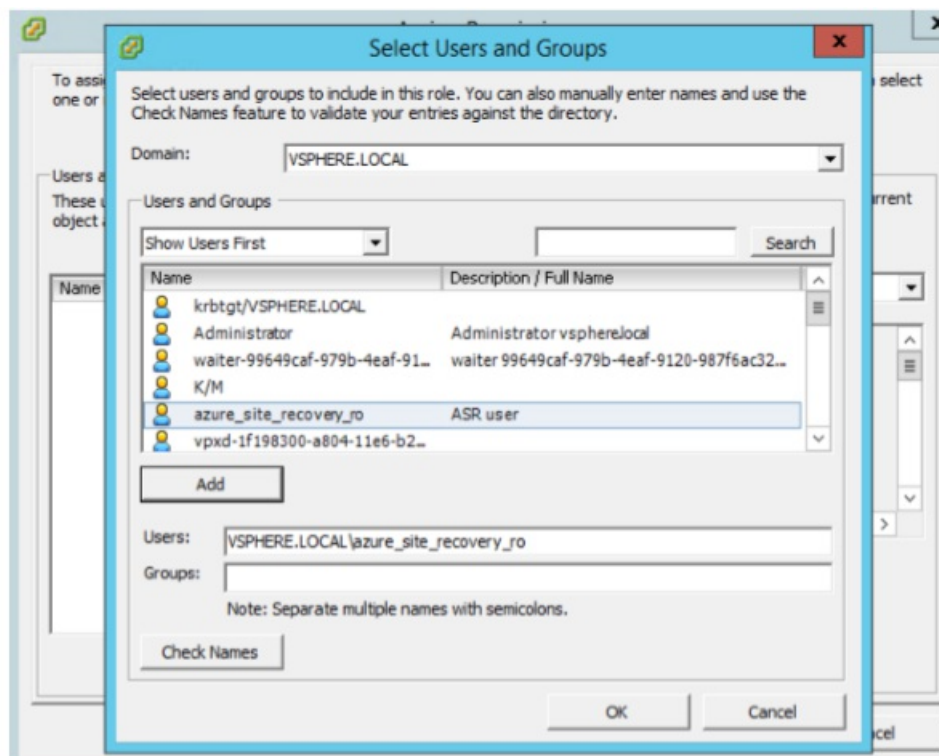
5. Select a VMware host, and then select the **Permissions** tab.



6. Right-click the name, and then select **Add Permission**.



7. Select **Add** and then, in the **Domain** box, select the appropriate domain in the drop-down list.



8. Select the user account that you created earlier, select **Add**, and then select **OK**.

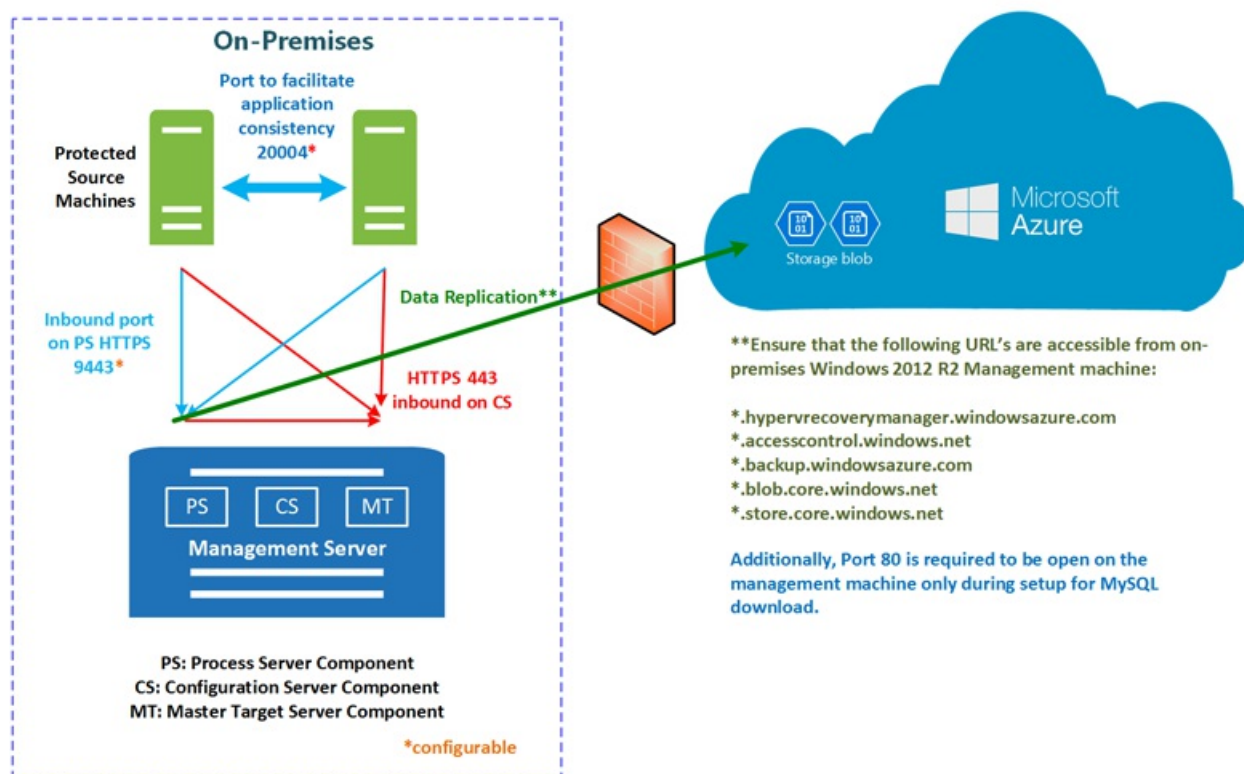
9. Confirm that the information is correct, and then select **OK**.

Azure Site Recovery architecture

Make sure that you have the following components to set up the on-premises Site Recovery infrastructure and its Azure counterparts.

- Configuration server and process server
- Mobility services agent
- Microsoft Azure

The following diagram shows a typical on-premises environment for preparing for Azure migration:



Configuration server

This component is a pre-provisioned, on-premises Windows Server that manages communication, data replication, and recovery processes. Run Unified Setup on this machine to install the configuration server, which includes the process server component.

The process server acts as a replication gateway. It receives replication data from the protected source machines. It optimizes this data with caching, compression, and encryption, and then sends it to Azure Storage. It also handles push installation of the mobility service to protected machines. The default process server is installed on the configuration server. To scale your migrations, you can deploy additional standalone process servers.

Mobility services

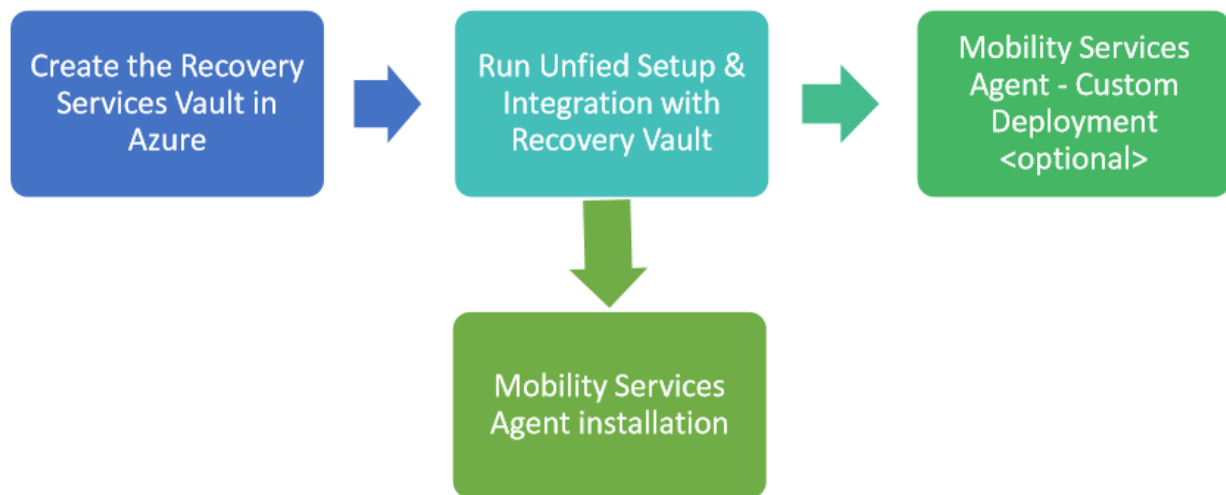
This component is deployed as an agent on every server that's replicated to Azure. It captures data, writes it to the server, and then forwards it to the process server. You can deploy the agents to the on-premises servers in a variety of ways.

Azure

In this section, you set up an Azure Recovery Services vault. Additionally, you set up Azure subscriptions, storage accounts, resource groups, and Azure virtual networks. These components should have been addressed during Site Recovery capacity planning.

Azure Site Recovery vault setup

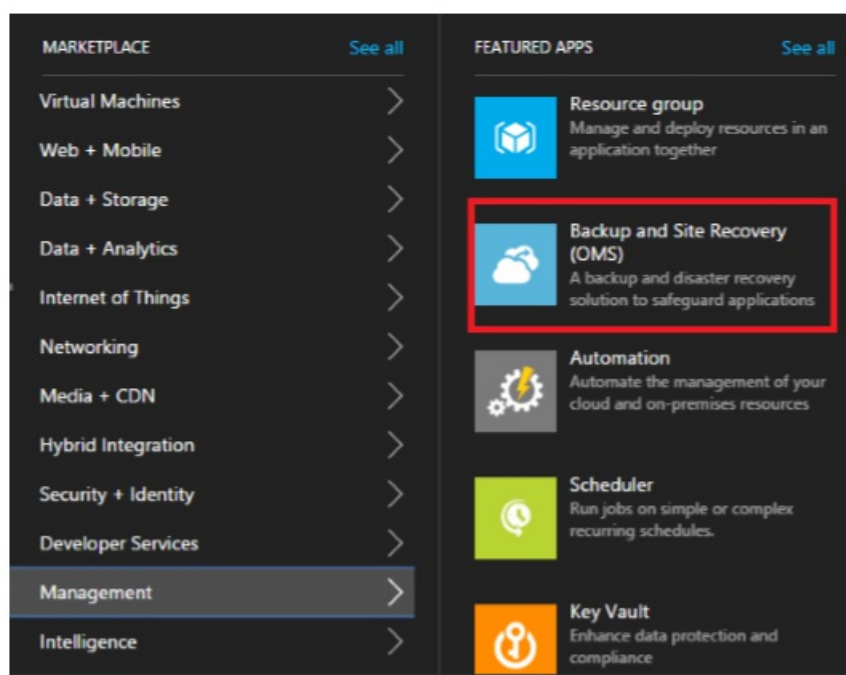
Site Recovery configuration contains two major steps, as illustrated in the following diagram:



Create a Recovery Services vault

First, create a Recovery Services vault in Azure that will host the Site Recovery components and the servers to be migrated to Azure.

1. From the configuration server, sign in to the Azure portal.
2. Select **New > Management > Backup and Site Recovery (OMS)**.



3. In the **Name** box, select a friendly name to identify the vault.
4. Select the correct Site Recovery subscription.
5. In the **Resource Group** box, select **Use existing**, and then select the resource group that you created for the migration efforts during [Site Recovery capacity planning on-site setup](#).
6. Select the same location as your other Site Recovery-related resources.
7. Select the **Pin to dashboard** check box, if you want, and then select **Create**.

The new vault appears on **Dashboard > All resources**, and on the main **Recovery Services Vaults** blade.

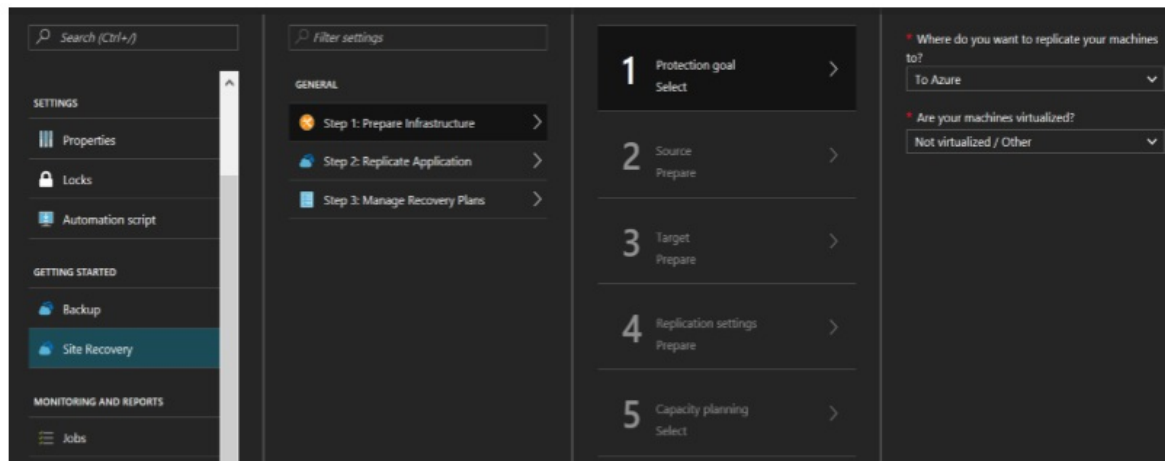
Run Unified Setup and integration with the Site Recovery vault

After you've set up the configuration server (on-premises) and created a Recovery Services vault, configure them

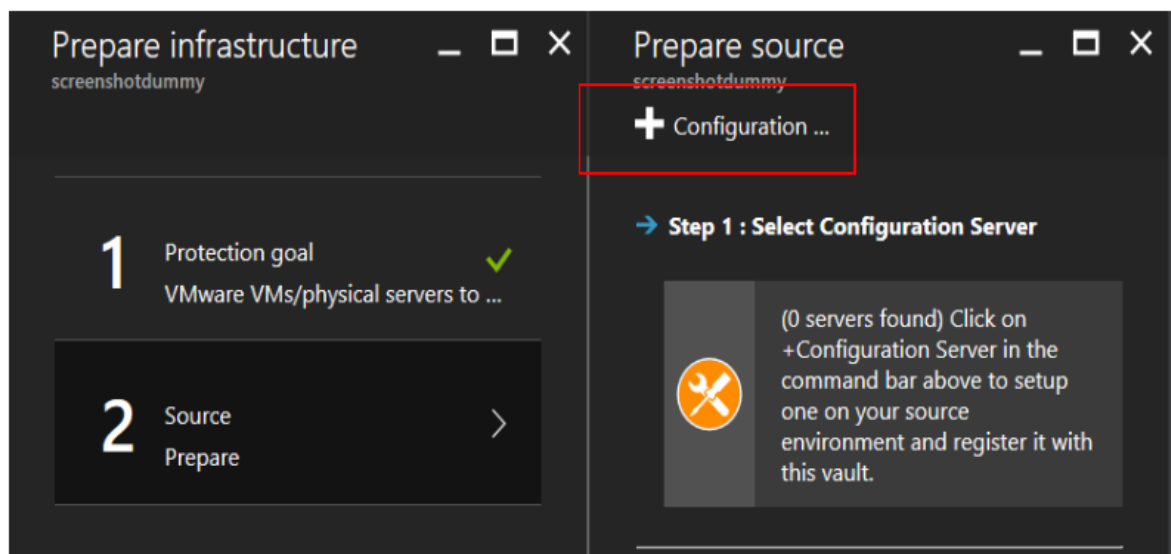
correctly to ensure that they can protect (and migrate) the workloads. This phase also includes the installation of the mobility services agent to the servers that must be migrated.

To choose your protection goals and set up your source environment, do the following:

1. Under **All Resources**, search for or select the Site Recovery vault that you just created, and then select the blade.
2. Select **Site Recovery** > **Step 1: Prepare Infrastructure** > **Protection goal**.
3. On the **Protection goal** blade, select **to Azure**, select **Not virtualized / Other**, and then select **OK**.



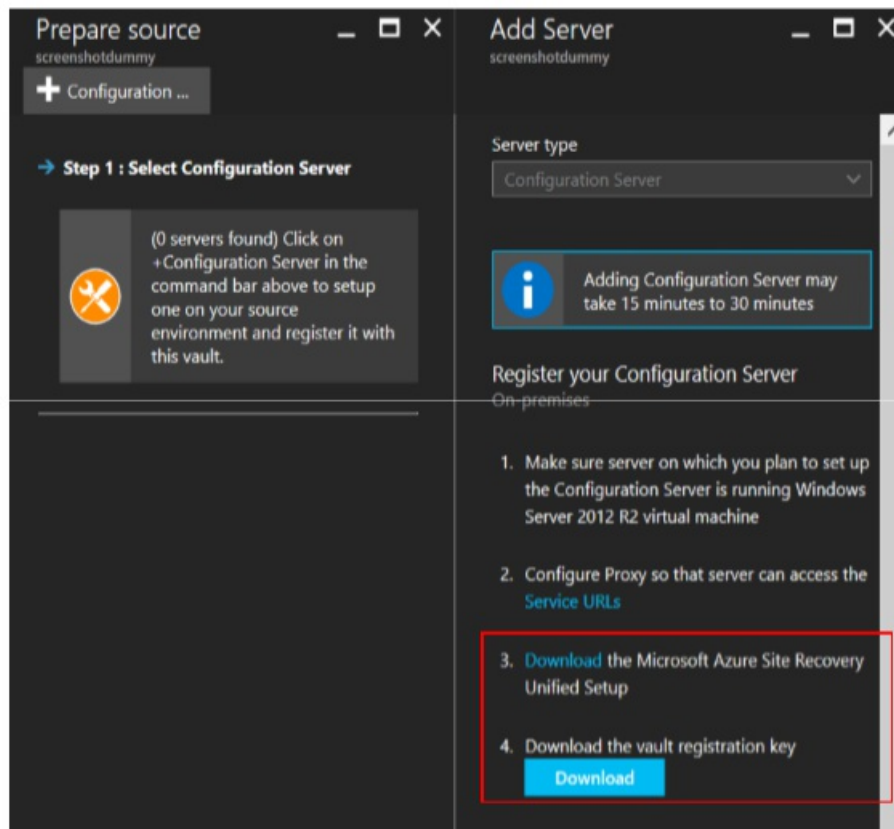
4. If you don't yet have a configuration server, select **Source**.



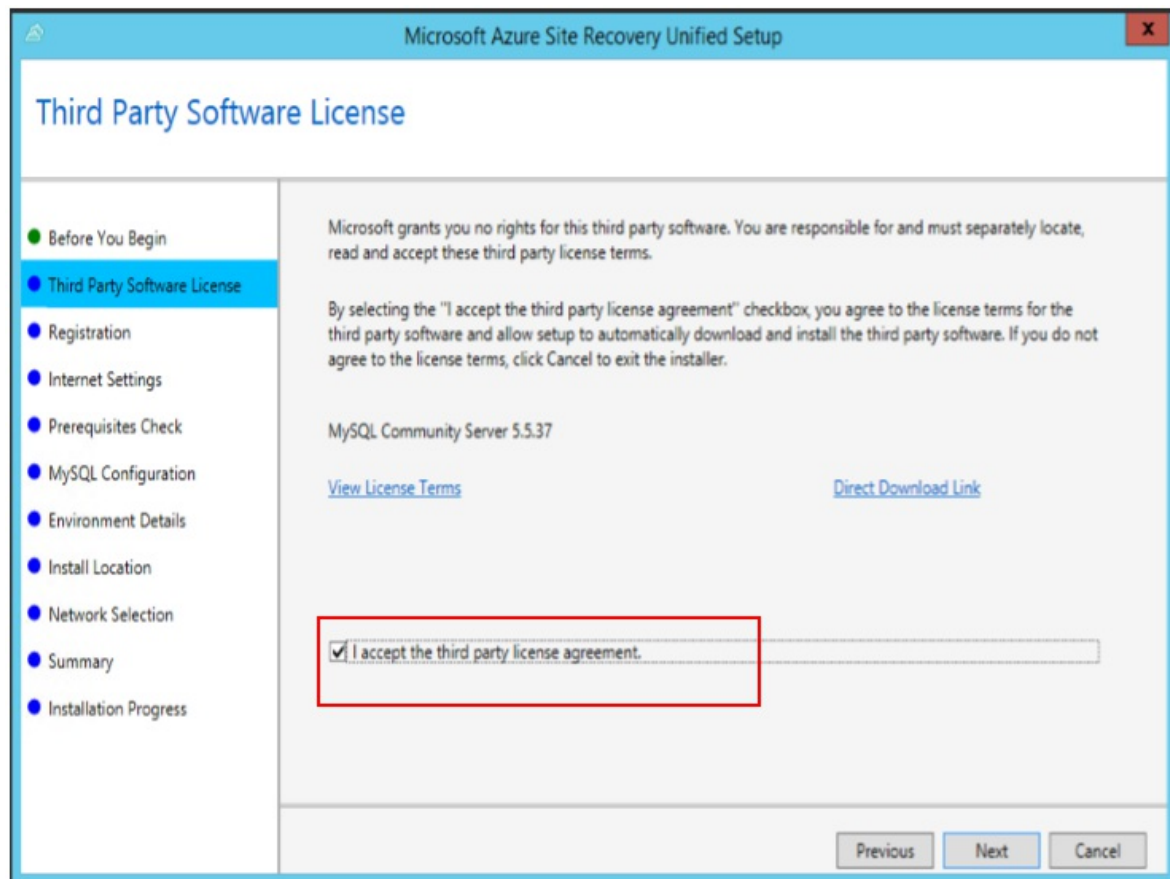
5. On the **Prepare source** blade, select **Configuration**.
6. On the **Add Server** blade, ensure that the **Server type** box contains **Configuration Server**.
7. Download the **Microsoft Azure Site Recovery Unified Setup** installation file and the **vault registration key** to a known location on the configuration server.

NOTE

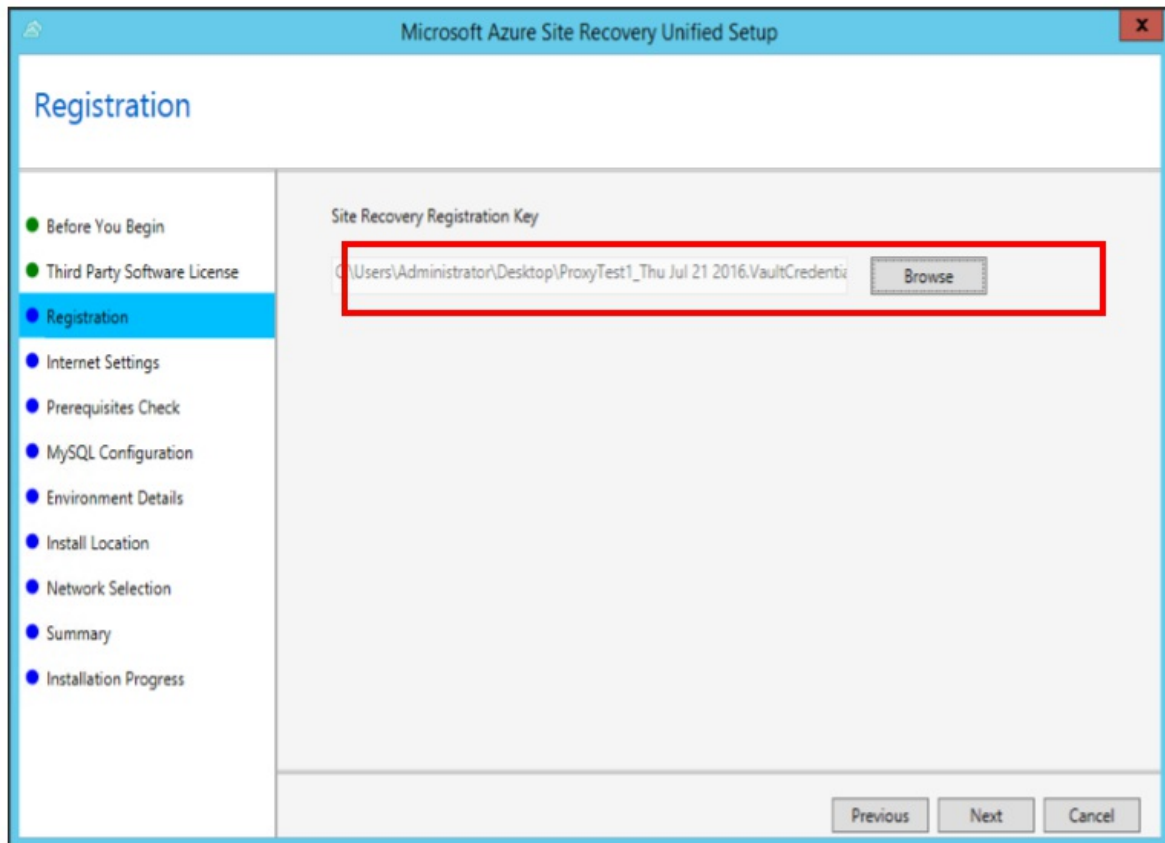
The vault registration key is valid for only five days after you generate it. You must either install the configuration server within five days or repeat steps 1-6 to download a new registration key file.



8. To run the Microsoft Azure Site Recovery Unified Setup installation file (MSI) as an administrator, right-click it, and then select **Run as administrator**.
9. On the **Before you Begin** tab, select **Install the configuration server and process server**, and then select **Next**.
10. On the **Third Party Software License** tab, select the **I accept the third party license agreement** check box to download and install MySQL, and then select **Next**.



11. On the **Registration** tab, select **Browse**, select the registration key that you downloaded from the vault in step 6, and then select **Next**.



12. On the **Internet Settings** tab, select the internet connection method, and then select **Next**.

NOTE

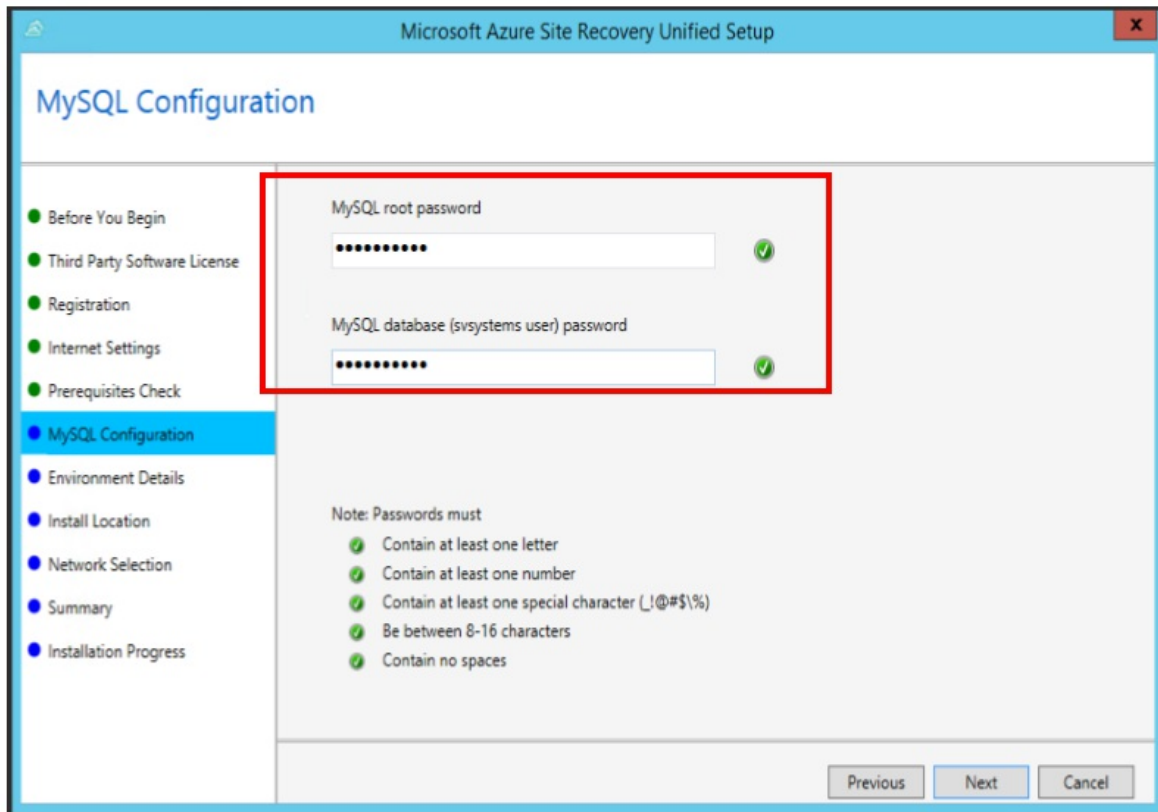
In the example scenario, the connection is direct, without a proxy server. If you are using a proxy server, provide the required additional details. If you need assistance, contact your IT or networking team.

13. On the **Prerequisites Check** tab, setup runs a series of checks to make sure that installation can run without issues. After you've completed the setup successfully, select **Next**.

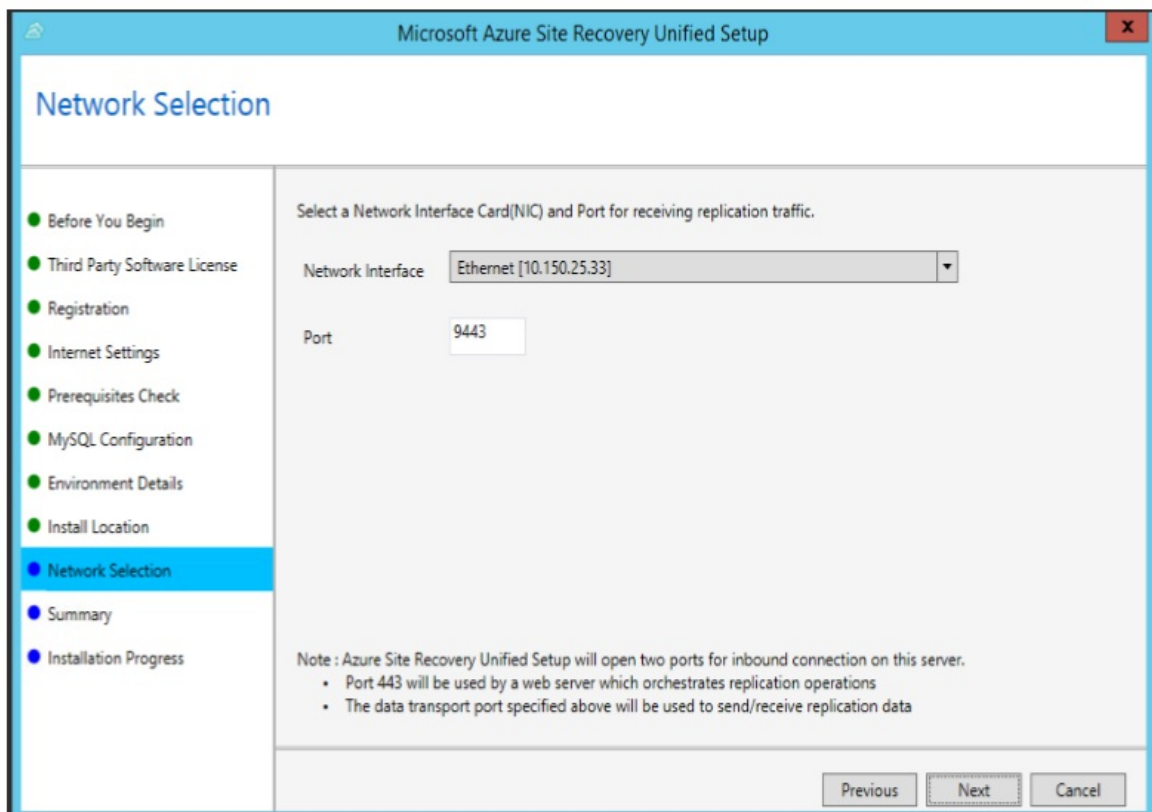
IMPORTANT

To avoid future issues, view and fix any warnings or errors. To view a message with more details about a warning or error, select the link under the message's status.

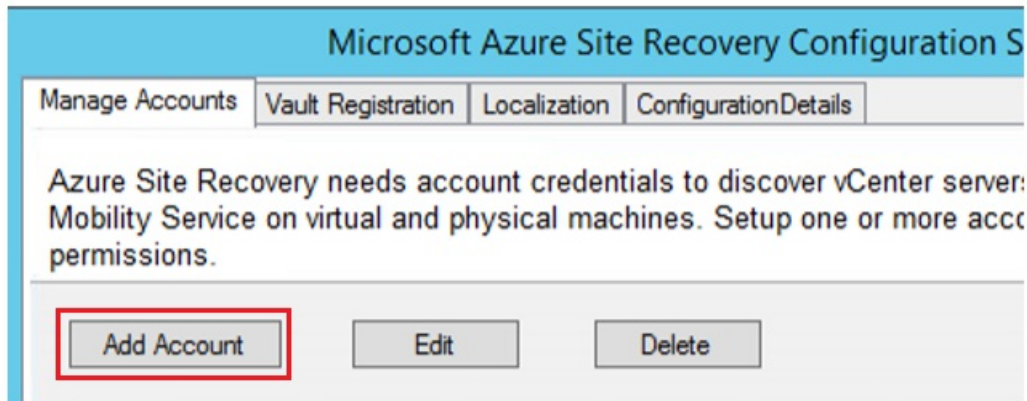
14. On the **MySQL Configuration** tab, provide credentials for the local instance of MySQL server that will be installed, and then select **Next**.



15. On the **Environment Details** tab, indicate whether you are migrating VMware VMs, and then select **Next**. Selecting **Yes** checks to ensure that vSphere PowerCLI 6.0 is installed.
16. On the **Install Location** tab, select where to install the binaries and store the cache. You can select any drive with at least 5 GB of storage, but we recommend that the cache drive have *at least 600 GB* of free space. Select **Next**.
17. On **Network Selection** tab, in the **Network Interface** box, select the correct network interface, and leave the port default setting at **9443**. You can change the default setting if you want, but it is not necessary or recommended. *Port 443 is prohibited*. Select **Next**.



18. On the **Summary** tab, review the information, and then select **Install**.
When the installation is complete, a passphrase is generated. Save it to a file at a secure location for later use. The cspconfigtool.exe file should open. If it doesn't, open it manually by using the desktop shortcut.
19. On the **Manage Accounts** tab, select **Add Account**, and then add the management accounts that the Site Recovery server will use to communicate with the migrating servers.

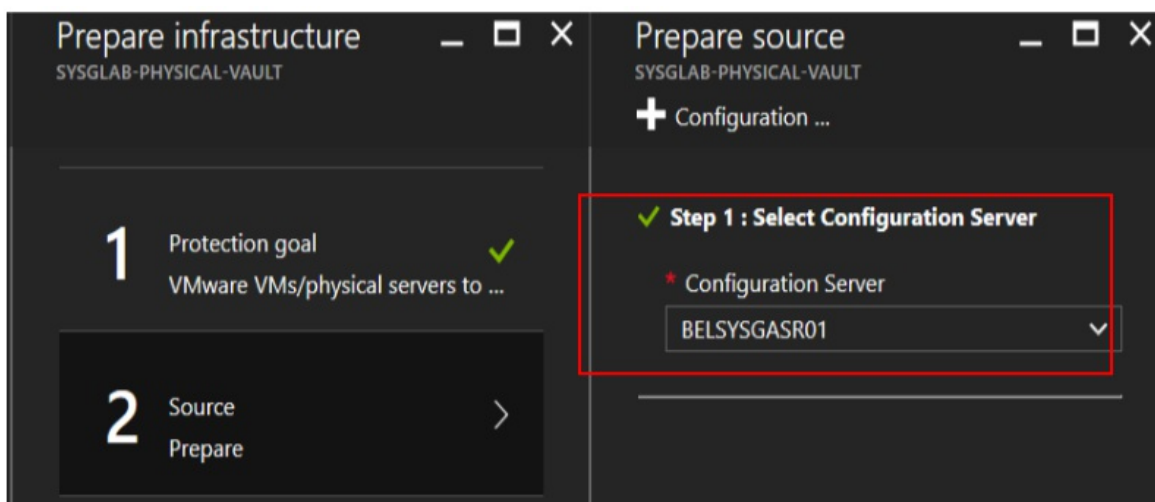


20. Add a domain account to the configuration server. This account should also be a Local Administrator on all migrating servers.

NOTE

This guide does not cover creating this domain account in Active Directory (or a similar identity management service). For assistance, contact your IT team.

21. Add the vCenter discovery account, a read-only account that you can use to automatically discover VMs that are managed by the vCenter server. *If you are not migrating VMware VMs, skip this step.*
22. Go to the Site Recovery vault, and then select **Your Recovery Vault > Site Recovery > Prepare infrastructure > Source**. The new configuration server is displayed. Select **OK**.



NOTE

If the server is not listed, go to **Your Recovery Vault > Site Recovery Infrastructure > Configuration Servers (For VMware & Physical Machines)**, select the ellipsis (...), and then select **Refresh Server**.

23. In the **Target** pane, make sure that the correct **Subscription** is selected, and then select **Resource Manager** for the deployment model. Wait for Azure to confirm steps 2 and 3, and then select **OK**.

Target

SYSGLAB-PHYSICAL-VAULT

+

Storage account

+

Network

✓

Step 1 : Select Azure subscription

* Subscription ⓘ

CSP_ACCELERATE_MIGRATION_OFFER... ▾

* Select the deployment model used after failover ⓘ

Resource Manager ▾

✓

Step 2 : Ensure that at least one compatible Azure storage account exist

Storage account(s) ⓘ

Found 4 compatible Azure storage accounts out of 4 available in the subscription

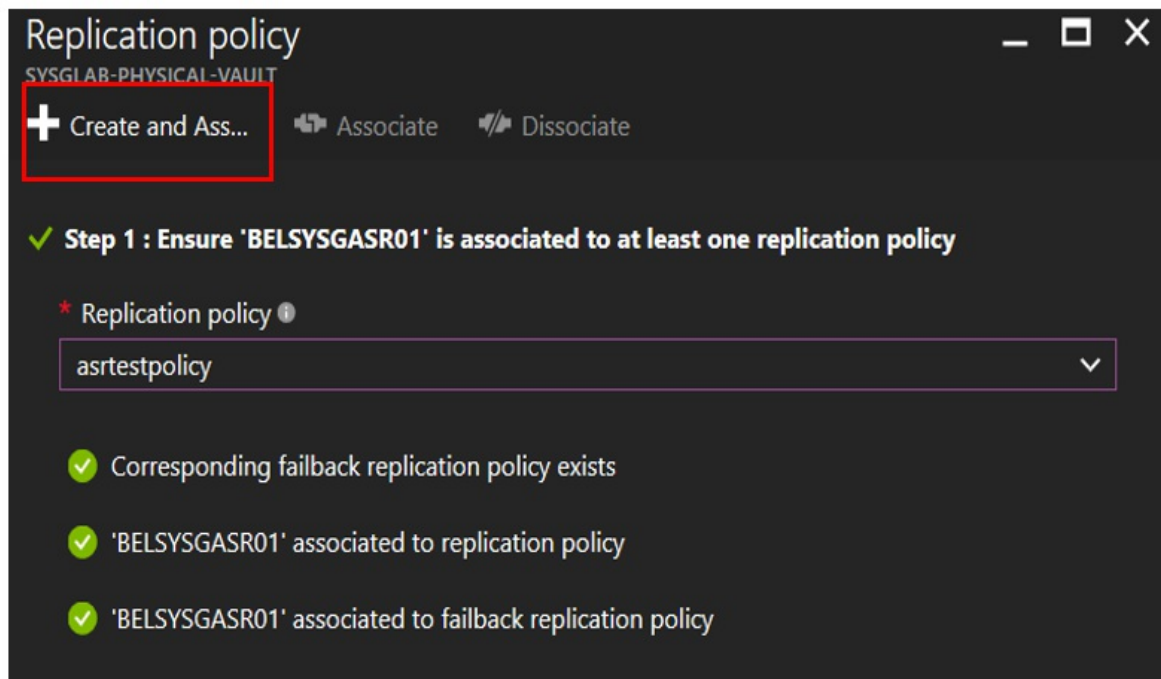
✓

Step 3 : Ensure that at least one compatible Azure virtual network exist

Network(s) ⓘ

Found 5 compatible Azure virtual networks out of 5 available in the subscription

24. In the **Replication policy** pane, select the **Create and Associate Policy** button, and then do the following:



a. Specify a suitable name.

b. Leave the defaults for **RPO threshold** (15), **Recovery point retention** (24), and **App-consistent snapshot frequency** (60). For more information about each feature, right-click the small **i** icon.

c. Every replication policy automatically creates a matching policy for failback. For example, if the policy is called **asrtestpolicy**, another policy will be created called **asrtestpolicy-failback**. This is default behavior, and the failback policy is of no use for migrations.

d. Select **OK** to create the policy. It is automatically associated with the configuration server.

25. For **Capacity Planning**, select **Yes, I have completed**, select **OK**, and then select **OK** again.

Add the vCenter server to the vault

NOTE

If you are not migrating any VMware virtual machines, skip this section.

1. In the Azure portal, go to **Your Recovery Vault > Site Recovery Infrastructure > Configuration Servers (For VMware & Physical Machines)**, select the ellipsis (...) next to the configuration server, and then select **Add vCenter**.
2. Provide the following required information:
 - **vCenter / vSphere Host Friendly Name:** Any friendly name, according to company policy.
 - **vCenter / vSphere IPAddress:** The on-premises IP address of the vCenter server.
 - **Port:** Use 443 (the default port) unless you're using another port.
 - **vCenter / vSphere host account:** Select the vCenter discovery account on the Site Recovery server.
3. To add the VCenter server to your vault, select **OK**.
4. The **Notifications** tab (the bell icon at the top right) shows the progress of the task. After the task is completed, select the configuration server to open a new blade, which you can expand to display your vCenter server and its status.

Deploying the mobility services agent

The mobility services agent gathers local server information and data, communicates with the configuration server, and assists in the overall data migration process. You can install the agent in a variety of ways:

Automated push from the Azure portal

After you have decided to replicate an application and servers, the workload-specific migration documentation provides instructions for how the mobility services agent is automatically pushed by using the Site Recovery vault in Azure.

Manual installation

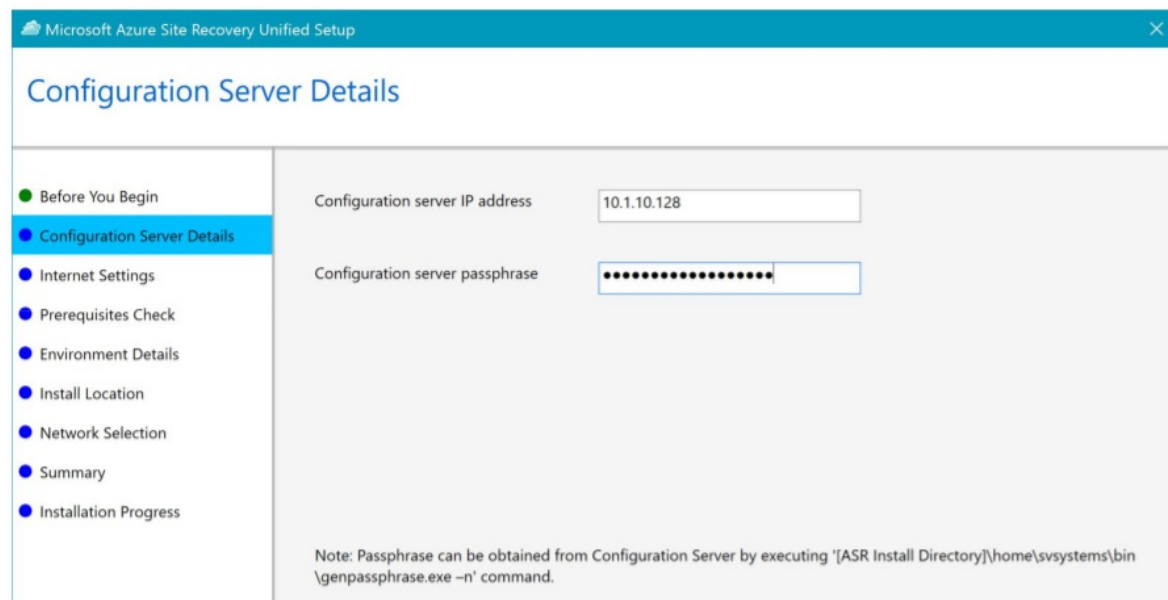
To manually install the mobility services agent on either a Windows or Linux server, do the following:

Windows

1. On the configuration server, go to `%Install Path%\Microsoft Azure Site Recovery\home\svsystems\pushinstallsvc\repository\`, and then download the installer file **Microsoft-ASRUA_9.0.0_Windowsrelease.exe** to the migrating servers.
2. On the server, run the installer as an administrator by right-clicking **File** and then selecting **Run as administrator**.
3. On the **Before You Begin** tab, select **Mobility service**, and then select **Next**.
4. On the **Configuration Server Details** tab, specify the IP address of the server and the passphrase that you generated during Unified Setup, and then select **Next**.

NOTE

If you do not have the passphrase, see [Retrieving the Unified Setup passphrase](#).



5. Leave the default **Install Location** settings, and then select **Next**.
6. Monitor the **Installation Progress** screen, and restart the machine if you are prompted to do so. After you've installed the service, it can take about 15 minutes for the status to be updated in the portal.
7. Repeat steps 1-6 for any additional servers that you want to migrate to Azure.

Linux

1. On the configuration server, go to `%Install Path%\Microsoft Azure Site Recovery\home\svsystems\pushinstallsvc\repository\`, and then download and execute

the relevant installer from the following table to the migrating server.

SOURCE OPERATING SYSTEM	MOBILITY SERVICES INSTALLATION FILE
CentOS 6.4, 6.5, 6.6 (64 bit only)	Microsoft-ASRUA_9..0.0_RHEL6-64release.tar.gz
SUSE Linux Enterprise Server 11 SP3 (64 bit only)	Microsoft-ASRUA_9..0.0_SLES11-SP3-64release.tar.gz
Oracle Enterprise Linux 6.4, 6.5 (64 bit only)	Microsoft-ASRUA_9..0.0_OL6-64release.tar.gz

2. In the table, copy the appropriate tar.gz string to the Linux machines that you want to protect.

3. Open a shell program, and then extract the zipped tar.gz string to a local path by running:

```
tar -xvzf Microsoft-ASR_UA_9.3.0.0_OL6-64_GA_17Aug2016_Release.tar.gz
```

4. Copy the passphrase from the configuration server located at

C:\ProgramData\Microsoft Azure Site Recover\private\connection.passphrase , and then save the passphrase by running

```
# echo [the file path the tar.gz string was extracted to] > ASRpassphrase.txt .
```

Running the command creates an ASRpassphrase.txt* file with the same path as the tar.gz string.

5. Install the mobility service by running the following command:

```
./install -t both -a host -R Agent -d <path to install> -i <IPAddress of ASR server> -p 443 -s y -c https -P ASRpassphrase.txt
```

6. Repeat steps 1-5 for any Linux servers to be migrated to Azure.

Enterprise-wide custom push

NOTE

The configuration server setup must be complete before you proceed with this section.

Your enterprise can also automatically push the mobility services agents to all servers that need to be protected by using custom change-management processes, such as Windows Server Update Services (WSUS), Microsoft System Center Configuration Manager, or Azure Automation Desired State Configuration.

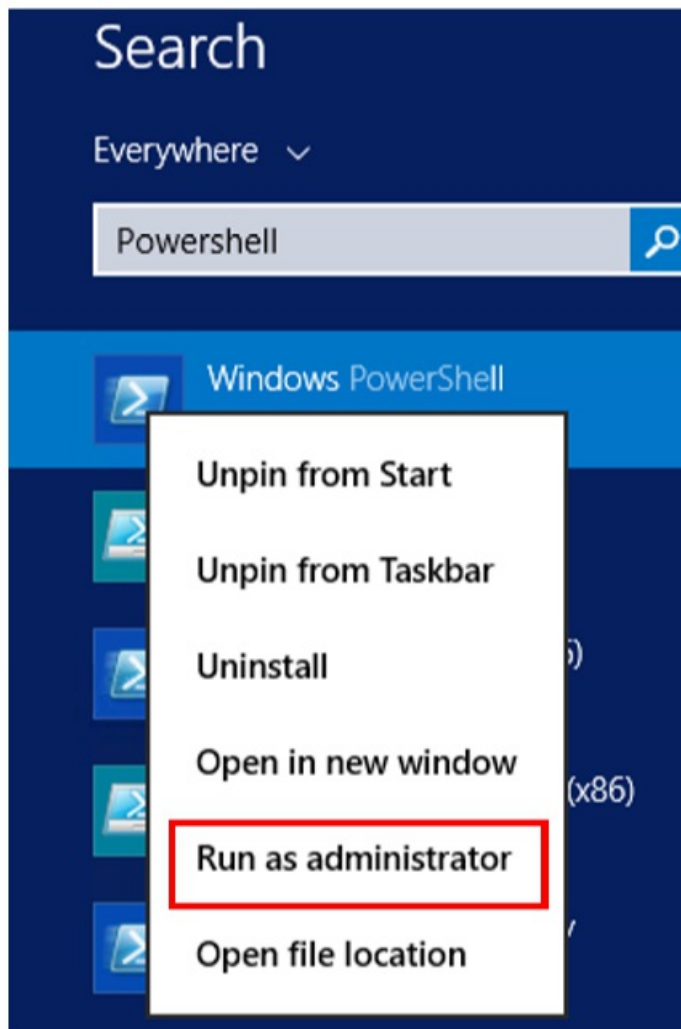
Because this guide does not explain the process involved here, follow your customized enterprise execution model.

Retrieve the Unified Setup passphrase

Unless you need to retrieve the configuration server passphrase in the [Windows installation](#) step, you can skip this section.

1. Sign in to the configuration server.

2. Open PowerShell as an administrator (select the **Windows** key+**Q**, type **PowerShell**, right-click the result, and then select **Run as administrator**).



3. In the command prompt:

- Type `CD "%installpath%\Microsoft Azure Site Recovery\agent\"` to change the directory to the folder where the configuration server is installed. (The default installation path is `C:\Program Files(x86)\Microsoft Azure Site Recovery\Agent`.)

```
Administrator: Windows PowerShell
PS C:\> cd "C:\Program Files (x86)\Microsoft Azure Site Recovery\agent"
PS C:\Program Files (x86)\Microsoft Azure Site Recovery\agent> .\genpassphrase.exe -v
DcXR7qtvNE1bAZp3PS C:\Program Files (x86)\Microsoft Azure Site Recovery\agent>
```

- Type `./genpassphrase.exe -v`.

Validate the infrastructure

To migrate on-premises workloads or servers, validate the Site Recovery on-premises infrastructure and ensure that it successfully communicates with Azure. To do so:

1. In the Azure portal, go to **Your Recover Vault > Site Recovery Infrastructure > Configuration Servers**, and then select your server to view various operational details of the configuration server.
2. Under **Associated servers**, in the **Status** column, ensure that there is a green check mark and the word **Connected** next to the process server, vCenter server (if applicable), and configuration server.

Additional steps

Site Recovery infrastructure should be ready to protect, replicate, and migrate all workloads to Azure. However,

some additional steps might be required, depending on the scale of the migration.

Complete the following steps *only if* you determined during capacity planning that you need more configuration or process servers during migration.

Additional servers

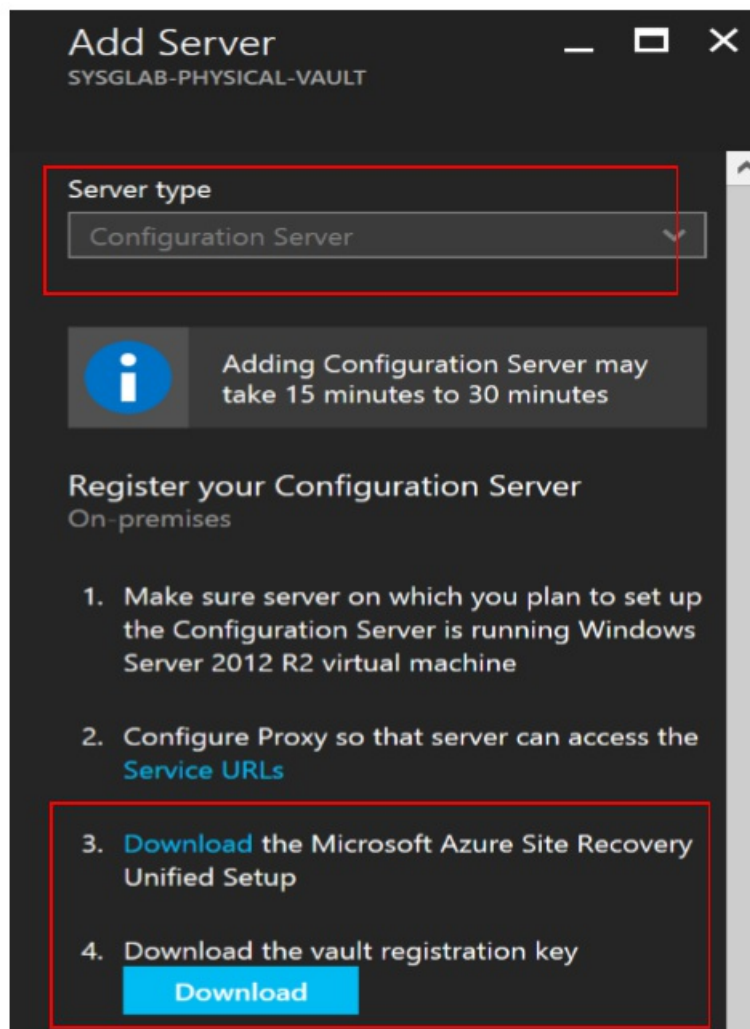
This step is required *only if* you are migrating a large number of applications, workloads, or servers to Azure from on-premises and you need to ensure that the configuration servers and process server can scale accordingly.

Additional configuration servers

1. In the Azure portal, under **All Resources**, search for or select **Your Recovery Vault**, and then select the blade.
2. Select **Your Recovery Vault** > **Site Recovery Infrastructure** > **Configuration Servers (For VMware & Physical Machines)**.
3. Select **Server**, and then do the following:
 - a. Ensure that **Server type** is **Configuration Server**.
 - b. Download the Site Recovery Unified Setup installation file to a known location (for example, C:\Temp).
 - c. Download the vault registration key to the same location that you downloaded the file from in the preceding step (for example, C:\Temp).

NOTE

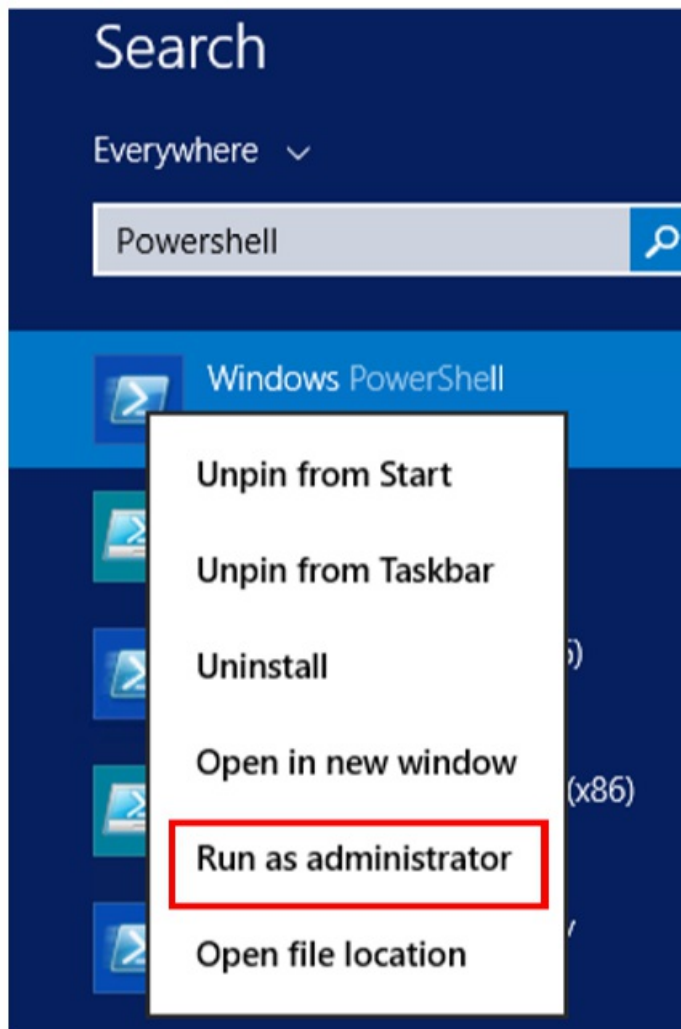
The key is valid for five days after you generate it.



4. To complete the new server setup, follow the setup instructions from the [Run Site Recovery Unified Setup](#) section.
5. After you've completed the preceding steps, you should now be able to see the new configuration server in the Azure portal. To do so, go to **Your Recovery Vault > Site Recovery Infrastructure > Configuration Servers (For VMware & Physical Machines)**.

Additional process server

1. In the Azure portal, go to **Your Recovery Vault > Site Recovery Infrastructure > Configuration Servers (For VMware & Physical Machines)**.
2. Select the configuration server (for example, ConfigurationServer#1) to which you want to add the new process server.
3. On the new configuration server blade, select **Process Server**, and then select **Deploy a scale out process server on-premises**.
4. Copy the Unified Setup file to the new process server to be configured. If you do not have the file, select **Download**, and then save the file to the new process server.
5. On the new process server, to run the Site Recovery Unified Setup installation file (MSI) as an administrator, right-click the file name, and then select **Run as administrator**.
6. On the **Before You Begin** tab, select **Add Additional Process Server to Scale Out Deployment**, and then select **Next**.
7. Under **Configuration Server Details**, specify the IP address of the configuration server (for example, ConfigurationServer#1) and the passphrase. To get the passphrase, do the following:
 - a. Sign in to the configuration server (for example, ConfigurationServer#1).
 - b. Open PowerShell as an administrator by selecting the **Windows** key+**Q**, typing **PowerShell**, right-clicking the result, and then selecting **Run as administrator**.



c. In the PowerShell command prompt:

- Type `cd "%installpath%\Microsoft Azure Site Recovery\agent"` to change the directory to the folder where the configuration server is installed. In this scenario, the default installation path is `C:\Program Files (x86)\Microsoft Azure Site Recovery\Agent`.
- Type `.\genpassphrase.exe -v`

```
Administrator: Windows PowerShell
PS C:\> cd "C:\Program Files (x86)\Microsoft Azure Site Recovery\agent"
PS C:\Program Files (x86)\Microsoft Azure Site Recovery\agent> .\genpassphrase.exe -v
DcXR7qtvNE1bAZp3PS C:\Program Files (x86)\Microsoft Azure Site Recovery\agent>
```

8. Select **Next**.

9. On the **Internet Settings** tab, select the internet connection method, and then select **Next**.

NOTE

In the example scenario, the connection is direct, without a proxy server. If you are using a proxy server, provide the additional required details. If you need assistance, contact your IT or networking team.

10. On the **Prerequisites Check** tab, setup runs a series of checks to make sure that installation can run without issues. After you've completed setup successfully, select **Next**.

IMPORTANT

To avoid future issues, view and fix any warnings or errors. To view a message with more details about a warning or error, select the link under the message's status.

11. On the **Environment Details** tab, indicate whether you are migrating VMware VMs, and then select **Next**. If you select **Yes**, this option checks to see whether vSphere PowerCLI 6.0 is installed.
12. On the **Install Location** tab, select where to install the binaries and store the cache. You can select any drive with at least 5 GB of storage, but we recommend that the cache drive have *at least 600 GB* of free space. Select **Next**.
13. On the **Network Selection** tab, select the correct network interface, and leave the default port at **9443**. *Port 443 is prohibited*. You can change the default if you want, but it is not necessary or recommended. Select **Next**.
14. On the **Summary** tab, review the information, and then select **Install**.
15. You can now use this new process server to either *switch* (from an existing server to the new process server) or *load balance* (across multiple process servers).
 - To switch, do the following:
 - a. Select any existing process server, and then select **Switch**.
 - b. Select the new process server you want to use, and then select the virtual machines that the new process server will handle.
 - d. Select the information icon to get information about the server. To help you make load decisions, the pane displays the average space that's needed to replicate each selected virtual machine to the new process server.
 - e. Select the check mark to start replicating to the new process server.
 - To load balance, do the following:
 - a. Select any existing process server, and then select **Load Balance**.
 - b. Select the new process server, and then select **OK**.
 - c. Select the servers that the new process server will manage, and then select **OK**.

Next steps

- Start [migrating your first workloads](#).
- Review [Azure Site Recovery capacity-planning guide](#).

Migrate an Azure subscription from Enterprise Agreement to Cloud Solution Provider

2/2/2018 • 1 min to read • [Edit Online](#)

Learn how to assess and then complete the migration of a customer's existing Azure subscription from Microsoft Enterprise Agreement or Microsoft Server and Cloud Enrollment (SCE) to the Azure Cloud Solution Provider (Azure CSP) program.

Process overview

Migrating an Azure subscription from Enterprise Agreement to CSP involves three major steps:

1. Ensure that source Azure EA subscription and target Azure CSP subscription are within the same tenant (directory). Refer to [partner EA to CSP](#) or [customer EA to CSP](#) migration procedure accordingly.
2. [Perform an assessment](#): Before you begin, assess all Azure resources in all subscriptions in the Enterprise Agreement account that you want to move to Azure CSP. This helps you determine what you can and can't move to Azure CSP, and what moving a service entails.

NOTE

Some Azure services are not available in Azure CSP. It's better to know as early as possible whether these services need to be restructured or cannot be migrated.

3. [Upgrade Azure classic deployment model to Azure Resource Manager](#): If a customer is still using resources that they created by using the Azure classic deployment model, the resources must be transferred to the Azure Resource Manager model before you move them to Azure CSP. This includes any virtual machines, virtual networks, and storage accounts that were created by using the Azure classic deployment model.

NOTE

Azure CSP supports only Azure Resource Manager services. Azure classic deployment services cannot be migrated to Azure CSP.

4. [Move resources from target subscription to Azure CSP](#): When all of the customer's services are using the Azure Resource Manager model, you can perform the final step of migrating them to an Azure CSP subscription.

Next steps

- Review the process of [Azure subscription migration from other channels](#), including from pay-as-you-go (PAYG) or the open model.
- Learn how to [migrate customers from on-premises to Azure CSP](#).
- If you need additional assistance with moving Azure subscriptions from EA to CSP - send an e-mail to azurecspmigrations@microsoft.com.

Migrate Azure subscription from partner EA to Azure CSP

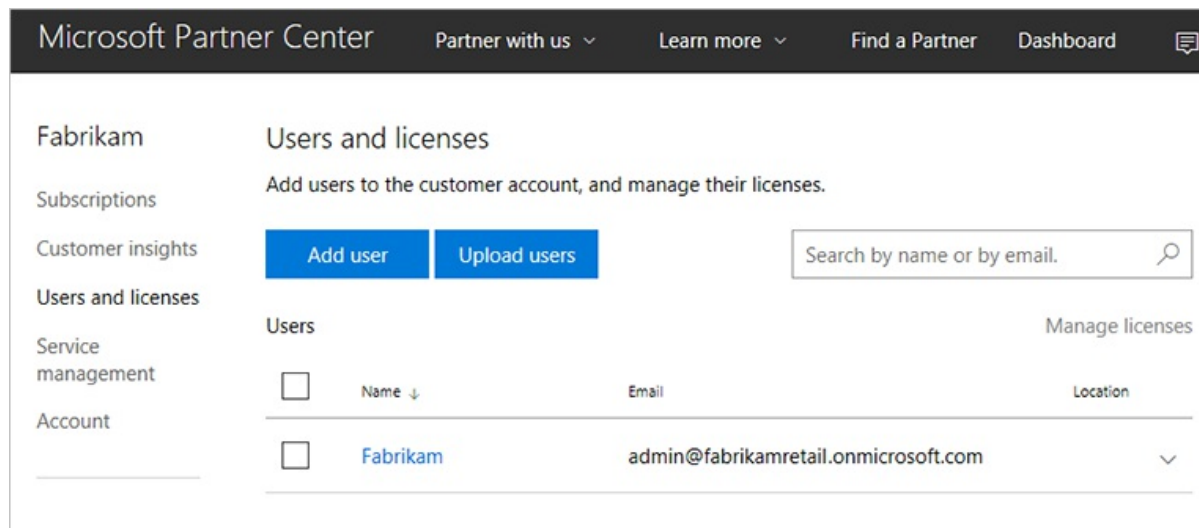
9/19/2017 • 2 min to read • [Edit Online](#)

If you are a Cloud Solution Provider and previously you've also signed an Enterprise Agreement to host workloads of your customers inside Azure subscriptions, you may would like to migrate Azure subscriptions from your EA to CSP.

[Azure Resource Move](#) requires source and target subscriptions to be inside the same Microsoft Online tenant (directory). If you use your own partner EA, most probably source Azure subscription exists inside your partner tenant. CSP requires to create Customer accounts, which also creates a tenant for customer. In general you will need to move source subscription from partner tenant to customer tenant in order to use Azure Resource Move.

Move Azure subscription from partner tenant to customer tenant

1. Sign in to [Partner Center portal](#) using CSP partner credentials.
2. Create a [new customer account](#) or [add an existing one](#). In the following example, we use the *fabrikamretail.onmicrosoft.com* directory, which belongs to a fictional customer, Fabrikam.
3. Save admin user credentials of the customer's tenant, that are shown in clear text during the customer creation. As alternative, you can create a new user account inside customer tenant and assign Global Administrator privileges. We will use *admin@fabrikamretail.onmicrosoft.com* user in this example.



4. Create new Azure CSP subscription inside this customer account (if you haven't did it on step 2) and [assign Owner role](#) to the customer's tenant admin user.
5. Sign in to [Azure EA portal](#) using Enrollment Administrator credentials of partner's EA. In this example we will use partner *Contoso* with *admin@contoso.onmicrosoft.com* as Enrollment Administrator.
6. Go to **Manage** section and open **Enrollment** tab. Change **Auth Level** to **Work or School Account Cross Tenant** in order to add another tenants to this Azure EA enrollment.

The screenshot displays the 'Enrollment Detail' page in the Azure portal. The left sidebar contains navigation links: 'Manage' (active), 'Reports', 'Notification', and 'Help'. The top navigation bar shows tabs: 'Enrollment' (active), 'Department', 'Account', and 'Subscription'. The main content area lists enrollment details:

Field	Value
Enrollment Number	<enrollment number>
Company Name	<partner name>
Country	United States
Auth Level	Work or School Account Cross Tenant
Start/End Date	
Billing Cycle	
Status	
Support Level	Pro-Direct
Support Coverage	<support coverage>
Azure Marketplace	Enabled
DA view charges	Enabled
AO view charges	Enabled
Cloud	Azure Commercial

A red box highlights the 'Work or School Account Cross Tenant' option in the 'Auth Level' dropdown menu. A blue button at the bottom left reads 'See related accounts'.

7. Switch to **Account** tab and click **Add account**. In the appeared menu select *Work of School Account* for *Authentication Type*, provide a friendly name in *Account Name* field and provide the e-mail of customer tenant admin.

Enrollment Department **Account** Subscription

Account

All Departments

Account Name Account Owner Auth Type Dev/

Add Account

Department *
Unassigned

Authentication Type *
☐ Microsoft Account ☒ Work or School Account

Account Name *
CSP

Email Address *
admin@fabrikamretail.onmicrosoft.com

Confirm Email Address *
admin@fabrikamretail.onmicrosoft.com

Cost Center

Add Cancel

8. If customer tenant admin user is mail-enabled (which is not required), you will receive a notification e-mail from Azure EA portal.

Invitation to Activate your Account on the Microsoft Azure service

waep@microsoft.com
Yesterday, 11:22 PM

👍 🔄 Reply all | ▼

📁 | Action Items

You have been invited to create a Microsoft Azure™ account associated with this email address: admin@fabrikamretail.onmicrosoft.com. This invitation comes from the administrator of the following Enterprise Agreement (EA) enrollment:

Company Name: <partner name>
 Enrollment Number: <enrollment ID>
 Subscription End Effective Date: <date>

If you are receiving this in error, no action is required. If you do desire to create a Microsoft Azure account related to this EA enrollment, please follow these simple steps:

Step 1: Log in to the [Microsoft Azure Enterprise Portal](#) using your Microsoft account or Work or School account: admin@fabrikamretail.onmicrosoft.com. If you have not already created an account, you will need to do so.

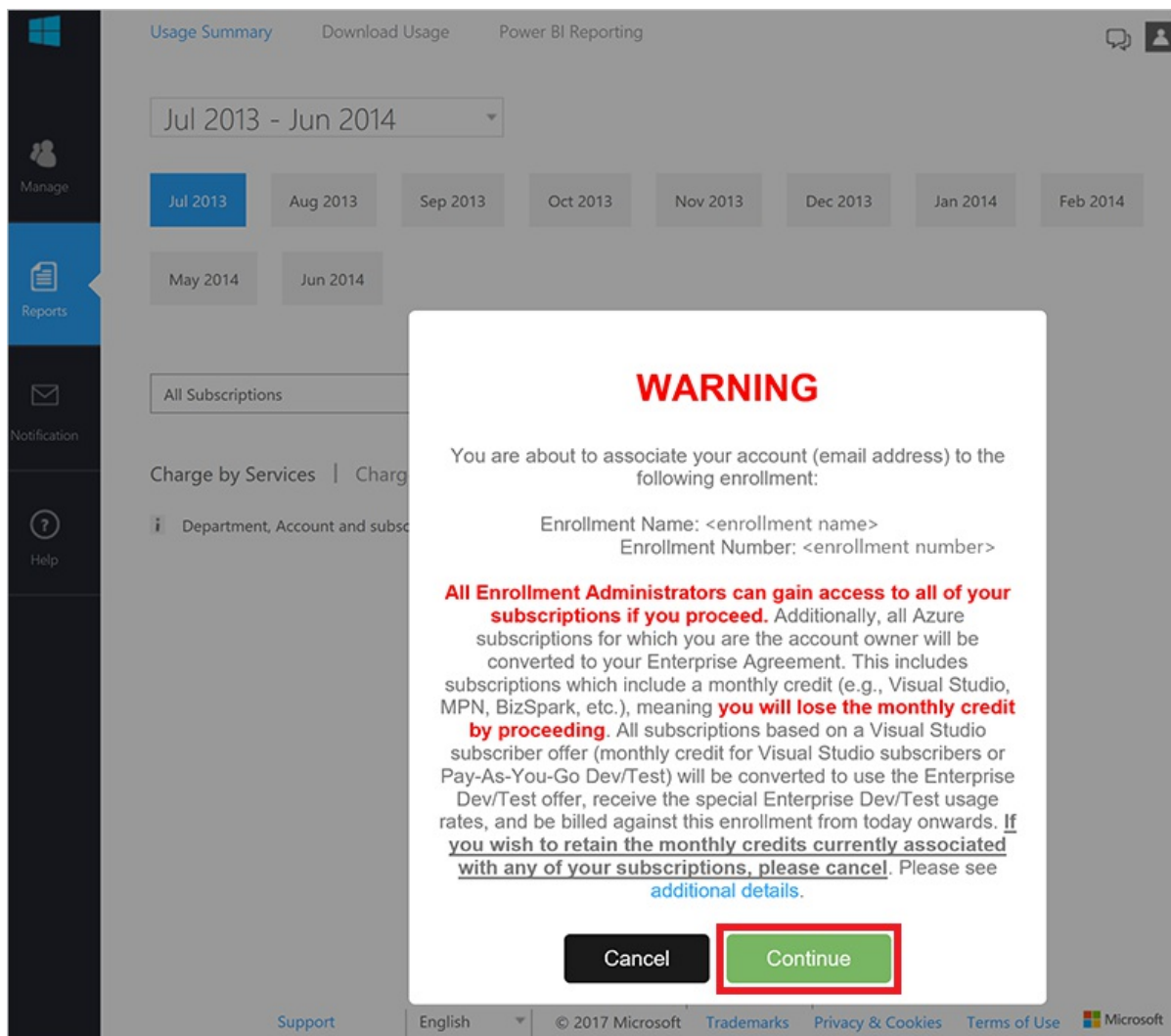
Step 2: Create an initial subscription by clicking on **Add Subscription** and stepping through the Account Sign-Up process on the Microsoft Azure Portal.

If you experience any problems accessing the Microsoft Azure Enterprise Portal website, please click [here](#).

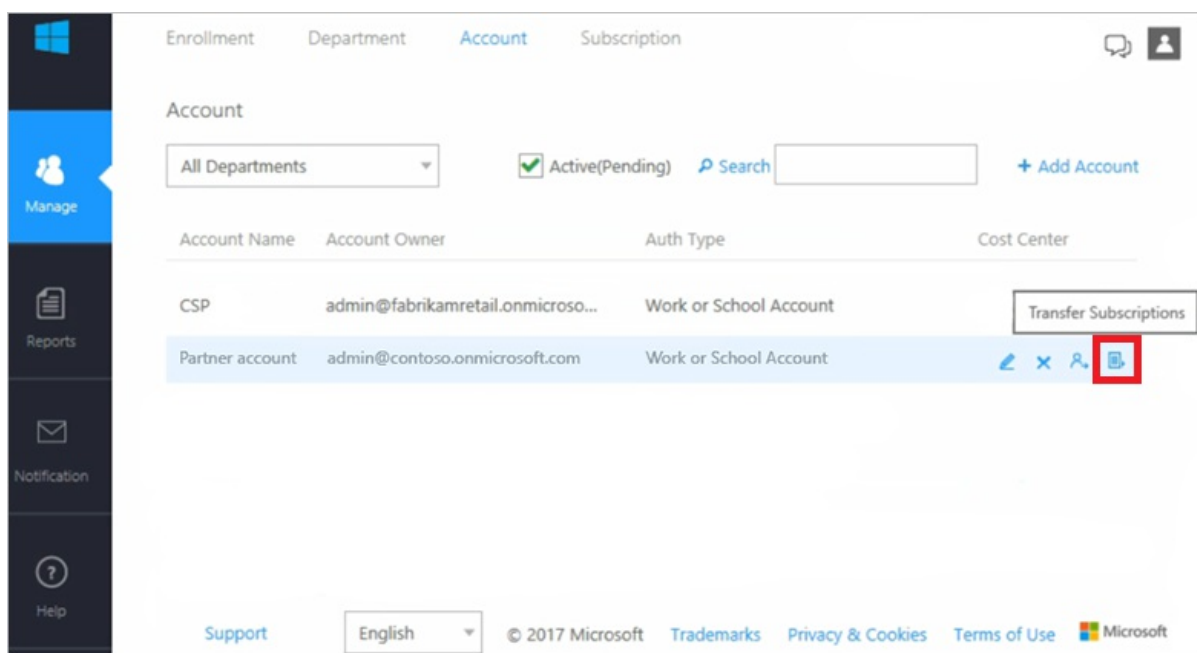
Microsoft respects your privacy. Please read our [Privacy Statement](#).

One Microsoft Way
 Redmond, WA, USA
 98052

9. Sign in to [Azure EA portal](#) using customer tenant admin credentials (admin@fabrikamretail.onmicrosoft.com in this case) and click **Continue** to associate customer's tenant with partner's Azure EA enrollment.



10. Sign in again to [Azure EA portal](#) using Enrollment Administrator credentials of partner's EA (*admin@fabrikamretail.onmicrosoft.com* in this case). Go to **Manage** section and open **Account** tab. Now you should see customer account in the account list as *Active*. Select partner account and click ****Transfer Subscription***.



11. Select the source subscription you would like to transfer from partner's tenant to customer's tenant.

Transfer Subscriptions

1 — 2 — 3

Source Account: Partner account

Select Subscriptions

Subscription Name	Subscription GUID
<input checked="" type="checkbox"/> Microsoft Azure Enterprise	<subscription ID>

Next Cancel

12. Select the customer account as target account.

Transfer Subscriptions

1 — 2 — 3

Select Target Account

Account Name	Account Owner
<input checked="" type="radio"/> CSP	admin@fabrikamretail.onmicrosoft.com

Note

If you are unable to select the account for transfer, please contact [Support](#).

Prev Next Cancel

13. Review the information on the page and click **Submit**.

Transfer Subscriptions



Confirm Information

Source Account Owner

Partner account
admin@contoso.onmicrosoft.com
Work or School Account

Selected Subscriptions

Subscription Name	Subscription GUID
Microsoft Azure Enterprise	<subscription ID>

Target Account Owner

CSP
admin@fabrikamretail.onmicrosoft.com
Work or School Account

Note

Source and Destination Account MUST be active.

Transfer Subscriptions to Microsoft Account is supported. Please note the Microsoft Account must have created an Azure subscription in order for it to be a valid target account. If the account is empty, please ask the Microsoft Account owner to first create an empty Azure subscription before attempting the transfer of subscriptions to the account. When you complete a transfer of subscriptions we will update the account administrator accordingly.

Please note: only subscription transfers between two organizational ID's within the same tenant will preserve existing service administrators and co-administrators. Any other subscription transfers will result in the loss of existing service administrators and co-administrators. Service administrators will be updated to the owner of destination account.

If you wish to change the service administrator, ensure the new owner of the subscriptions follows these steps:

1. Sign in to account.windowsazure.com
2. Click into one of the subscriptions that appear in the subscriptions list (following steps should be completed for all subscriptions)
3. Click on 'Edit Subscription Details' from the options on the right
4. Update the service administrator field accordingly

Digital Microsoft Partner ID's will not be preserved for any type of subscription transfer. If you're working with a partner, consider updating the partner ID on this subscription. You can do this in the [Azure Account Center](#)

For additional help, please contact [Support](#).

Prev

Submit

Cancel

14. The process will take few minutes. If target account is mail-enabled, you will receive a notification when subscription migration from partner's tenant to customer's tenant will be completed.

A subscription transfer has been completed

admin@fabrikamretail.onmicrosoft.com has successfully replaced
admin@contoso.onmicrosoft.com as account administrator of this subscription.

Subscription name: Microsoft Azure Enterprise

Subscription ID: <subscription ID>

Please review [this list](#) of recommended next steps.

Thanks,
The Microsoft Azure team

This message from Microsoft is an important part of a program, service, or product that you or your company purchased or participates in. Microsoft respects your privacy. Please read our [Privacy Statement](#).

15. Sign in to [Azure Management portal](#) using customer's admin user credentials and you should see both subscriptions - EA and CSP - in the Subscription list.

SUBSCRIPTION	SUBSCRIPTION ID	MY ROLE	CURRENT COST	STATUS
Microsoft Azure CSP	<subscriptionID>	Owner	Not available	Active
Microsoft Azure Enterprise	<subscriptionID>	Unknown	Unauthorized	Active

Now you can process to next steps and migrate Azure resources from Azure EA subscription to Azure CSP subscription withing the same customer tenant.

Next steps

- [Assess the migration](#) to understand the issues that you identify during the process.
- [Convert](#) resources from the Azure classic deployment model to Resource Manager.

- [Migrate](#) supported Resource Manager resources to Azure CSP.

Migrate an Azure subscription from a customer Enterprise Agreement to an Azure CSP

9/19/2017 • 1 min to read • [Edit Online](#)

Because [Azure Resource Move](#) doesn't support cross-tenant migrations, the source subscription and destination subscription must be in the same directory. You need to migrate Azure resources from a subscription inside a customer's Enterprise Agreement enrollment to a Cloud Solution Provider (CSP). To do so, create an Azure CSP subscription within an existing customer tenant.

Create an Azure CSP subscription in the existing directory

The source subscription should exist within the Azure Active Directory (Azure AD)-based directory. You can't use a subscription that's associated with a Microsoft account. If the source subscription is associated with a Microsoft account, see [Transfer ownership of an Azure subscription to another account](#). Then move the subscription to an Azure AD-based directory. In the following example, we use the fabrikamretail.onmicrosoft.com directory, which belongs to a fictional customer, Fabrikam.

1. To generate a link, see [Add an existing customer to an Azure Cloud Solution Provider](#). Send the link to the global administrator of the fabrikamretail.onmicrosoft.com directory.
2. After the global administrator approves the reseller relationship, Fabrikam appears in the list of Azure CSP customers. Create a new Azure CSP subscription within this Azure CSP customer in the Microsoft Partner Center.

Microsoft Partner Center

Partner with us ▾ Learn more ▾ Find a Partner > Dashboard

Fabrikam Subscriptions

Subscriptions **Add subscription**

Devices **License-based subscriptions**

Customer insights This customer doesn't have any license-based subscriptions.

Users and licenses **Usage-based subscriptions** [All resources \(Azure portal\)](#)

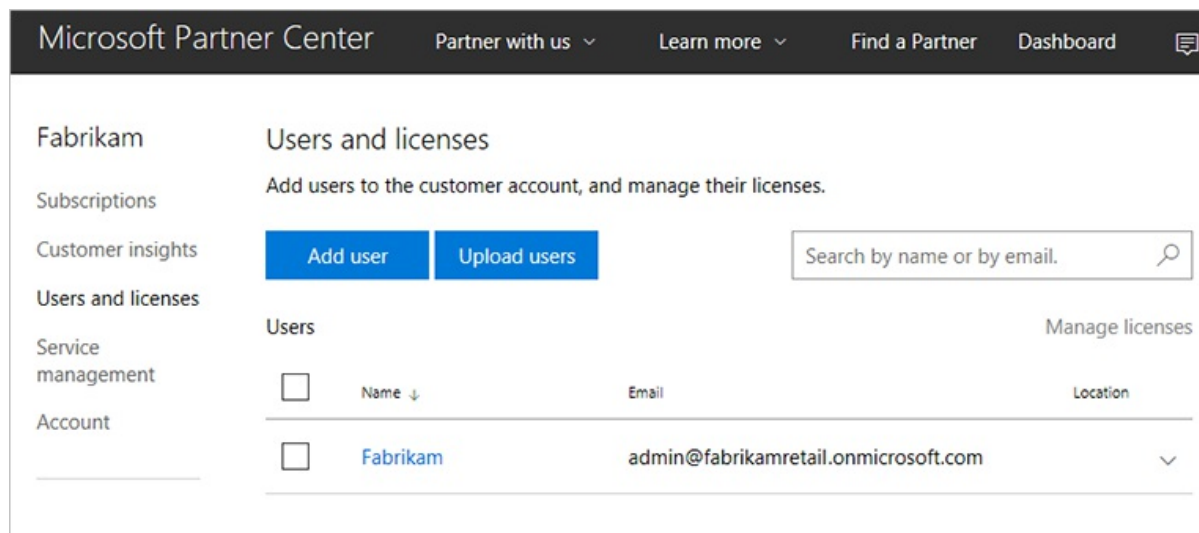
Service management Current estimate **\$0.00** Set a budget, and you'll be able to track the usage on the portal if your monthly spending will exceed it.

Account [Apply](#)

← Customers Billing period July 26 - August 25. There are 4 days left.

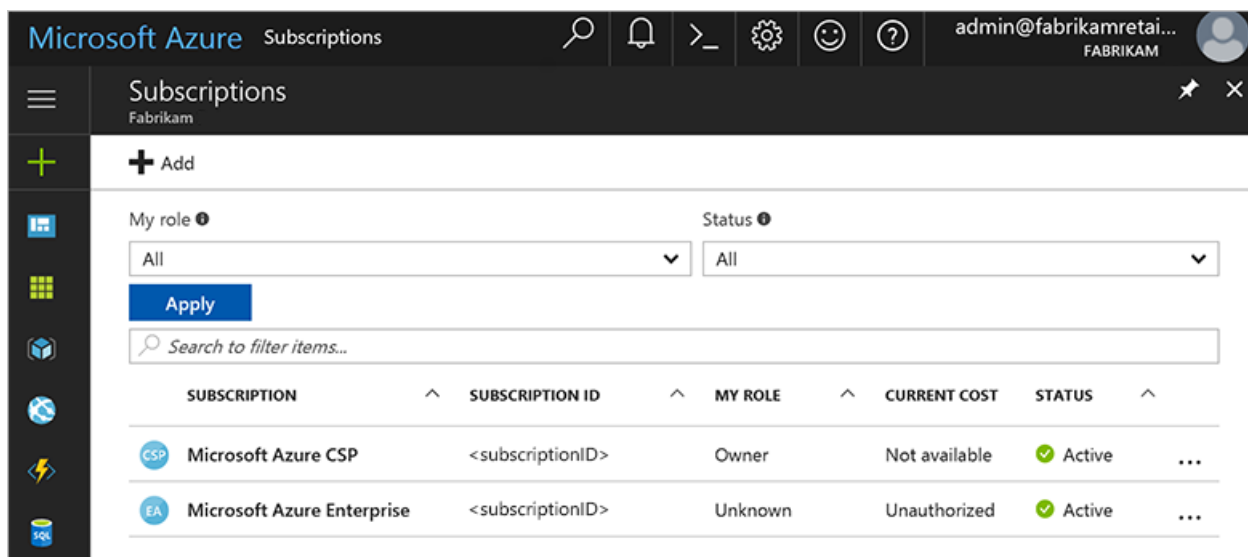
Subscription	Current estimate	% of total ↑	Status
Microsoft Azure CSP	\$0.00	0%	Active ▾

3. [Create a new user](#) in the Fabrikam customer account. In our example, we use admin@fabrikamretail.onmicrosoft.com.



4. Assign owner rights to the admin@fabrikamretail.onmicrosoft.com user account in the destination Azure CSP subscription. For instructions, see [Assign and manage permissions within an Azure subscription](#).
5. Ask the customer to assign owner rights to the admin@fabrikamretail.onmicrosoft.com user account in the source Azure Enterprise Agreement subscription. Use a similar procedure to Step 4, but on the customer side.

After you finish the preceding steps, sign in to the [Azure portal](#) by using the admin@fabrikamretail.onmicrosoft.com account. You should see both the source subscription and the destination subscription in the subscription list.



Now you can migrate Azure resources from an Azure Enterprise Agreement subscription to an Azure CSP subscription within the same customer tenant.

Next steps

- [Assess the migration](#) to understand the issues that you identify during the process.
- [Convert](#) resources from the Azure classic deployment model to the Azure Resource Manager deployment model.
- [Migrate](#) supported Resource Manager resources to an Azure CSP.

Assessment tool for migrating an existing Azure subscription to Azure CSP

2/2/2018 • 2 min to read • [Edit Online](#)

Before you can migrate an existing Azure subscription to Azure Cloud Solution Provider (Azure CSP), it's a good idea to complete an assessment to analyze existing resources. An assessment can help you identify resources that might present roadblocks in the migration process.

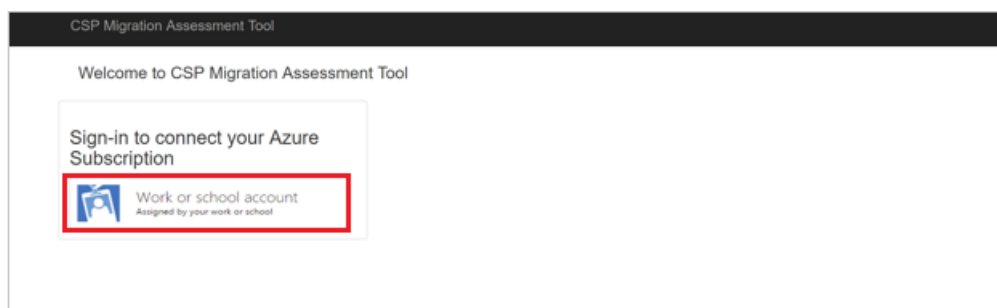
The Azure CSP Migration Assessment Tool analyzes all your existing resources by subscription. It provides a migration feasibility assessment for each resource, including cost comparisons and recommendations for migration.

Migration assessment

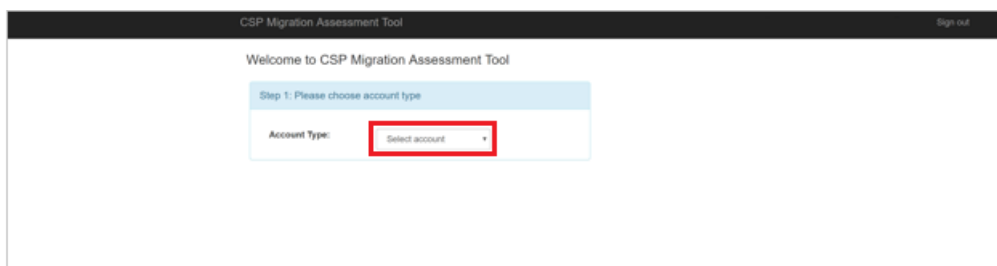
1. Go to the [Azure CSP Migration Assessment Tool](#).
2. Sign in to the assessment tool by using your Azure Active Directory (Azure AD) credentials.

NOTE

If you have trouble signing in to the tool, you might be using the Microsoft Account (LiveID) credentials that you use to sign in to your Azure subscription. To use the assessment tool, you must first switch from your Microsoft Account to an Azure AD based account. For detailed instructions about how to switch to Azure AD account, see [Transfer ownership of an Azure subscription to another account](#).



3. On the **Welcome** page, for **Account Type**, select either **EA** or **Other**. **Other** is for direct or open Azure accounts.



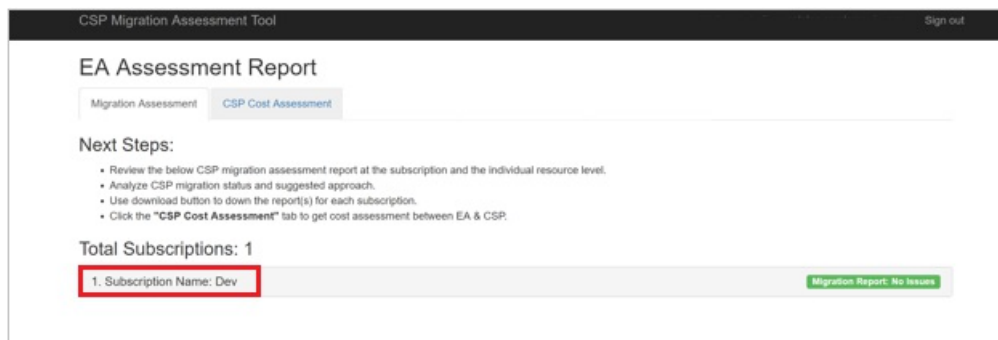
4. Based on your selection in step 3, complete the rest of the settings on the **Welcome** page:
 - a. For Enterprise Agreement accounts, enter the relevant information in the **EA Enrollment Number** box and the **Primary/Secondary Access Key** box.

b. For open or direct accounts, ensure that the URL for the account that is signed in is correct.

5. Select the **Run Assessment** button.

It might take a minute for the assessment tool to collect usage details.

6. On the **EA Assessment Report** page, the results show a list of all subscriptions that are associated with this account. Select the subscription name.



7. The assessment report displays the subscriptions. The **CSP Status** column shows one of the following for migration status:

- **NO ISSUES** (green) means the subscription is ready to be moved to Azure CSP.
- **MODIFY** (yellow) means that one or more resources in the subscription need to be modified before they can be moved to Azure CSP.
- **BLOCKED** (red) means that one or more resources in the subscription cannot be moved to Azure CSP at this time. These resources must be excluded if you want to migrate the subscription.

8. The **Suggested Approach** column suggests a migration approach for each resource. The suggested approach is either a one-step migration (Azure Resource Manager to Azure CSP) or a two-step migration (Azure classic deployment model to Resource Manager, and then Resource Manager to Azure CSP).

CSP Migration Assessment Tool

Sign out

Total Subscriptions: 1

1. Subscription Name: Dev Migration Assessment: No Issues

Subscription Id: 60e5990f-d7cd-4894-b095-bded8682484b

Download Assessment Report

Search:

Resource Group	Resource ID	Resource Name	Consumed Service	Meter Category	Meter Sub-Category	Meter Name	CSP Status	Suggested Approach
	0e9d0c9b-ab6d-4312-9c7e-3794e22af9c4		Storage	Storage	Geo Redundant	Standard IO - Page Blob/Disk (GB)	NO ISSUES	ASM to CSP (two steps migration)
	32c3ebec-1646-49e3-8127-2cafbcd3a04d8		Compute	Networking	Networking	Data Transfer In (GB)	NO ISSUES	ASM to CSP (two steps migration)
	9995d93a-7d35-4d3f-9c69-7a7fea447ef4		Compute	Networking	Networking	Data Transfer Out (GB)	NO ISSUES	ASM to CSP (two steps migration)
	ec2457ae-8775-48bf-8bdc-f1c94e96eb56		Compute	Virtual Machines	A0 VM (Windows)	Compute Hours	NO ISSUES	ASM to CSP (two steps migration)
	92567832-8613-47c4-9e8a-b2b9461a4875		Storage	Storage	Geo Redundant	Standard IO - Table (GB)	NO ISSUES	ASM to CSP (two steps migration)
	11278850-f161-4a6e-86ef-d650a29fb62f		Storage	Data Management	Geo Redundant	Standard IO- Page Blob Write Operation Units	NO ISSUES	ASM to CSP (two steps migration)

For more information about these migration processes, see [Upgrade from classic to Resource Manager to move an existing subscription to Azure CSP](#) and [Move an existing subscription from Resource Manager to Azure CSP](#).

Cost assessment

The next step is to get an estimate for what migrated subscriptions will cost after you switch to Azure CSP.

1. In the Azure CSP Migration Assessment Tool, select the **CSP Cost Assessment** tab.

CSP Migration Assessment Tool Sign out

EA Assessment Report

Migration Assessment
CSP Cost Assessment

Next Steps:

- Select the currency and enter CSP rebate if any
- Click the "Run CSP Cost Assessment" button to run the cost assessment
- Analyze and review the current cost, CSP costs and savings
- Click the "Migration Planning" tab to get resource dependencies and to plan for migration.

Select your Currency:

Enter your CSP discount [%]:

Run CSP Cost Assessment

- Select your currency. Then, enter your CSP discount as a percentage. Select **Run CSP Cost Assessment**.

CSP Migration Assessment Tool Sign out

EA Assessment Report

Migration Assessment
CSP Cost Assessment

Next Steps:

- Select the currency and enter CSP rebate if any
- Click the "Run CSP Cost Assessment" button to run the cost assessment
- Analyze and review the current cost, CSP costs and savings
- Click the "Migration Planning" tab to get resource dependencies and to plan for migration.

Select your Currency:

Enter your CSP discount [%]:

Run CSP Cost Assessment

- When the assessment is finished, *select each subscription* to see a detailed list of resource costs. Resource costs are based on *the past 30 days of usage*.

CSP Migration Assessment Tool Sign out

15

Run CSP Cost Assessment

1. Subscription Name: Dev

Subscription Id: 60e5990f-d7cd-4894-b095-bded8682484b

Download Assessment Report

Search:

Resource Group	Resource ID	ConsumedService	MeterName	ConsumedQuantity	EA Rate	EA Cost	CSP Rate	CSP Cost	Cost Diff
	0e9d3c9b-ab6d-4312-9c7e-3794e22af9c4	Storage	Standard IO - Page Blob/Disk (GB)	0.7829	0.095	0.0744	0.0808	0.0632	0.0112
	32c3ebec-1646-49e3-8127-2cafb3a04d8	Compute	Data Transfer In (GB)	0.018	0	0	0	0	0
	9995d93a-7d35-4d3f-9c69-7a7fea447ef4	Compute	Data Transfer Out (GB)	0.0408	0.087	0.0036	0.0739	0.003	0.0005
	ec2457ae-8775-48bf-8bdc-11c94e96eb56	Compute	Compute Hours	24	0.02	0.48	0.017	0.408	0.072
	923078e1-fd3f-4bd5-a798-44b533057e46	Storage	Standard IO-Block Blob Delete Operation Units	0.1439	0.0004	0.0001	0.0003	0	0
	ad22fac8-9da5-4577-8683-56ae94d39e42	Storage	Standard IO-Table Write Operation Units	0.0146	0.0004	0	0.0003	0	0
					50.85			43.21	7.62

Showing 1 to 10 of 375 entries

Previous 1 2 3 4 5 ... 38 Next

Total Services:375

Next steps

- If you are experiencing any issues with the assessment portal or require additional help to migrate existing Azure subscriptions to CSP - please, send an e-mail to azurecspmigrations@microsoft.com.
- [Convert](#) resources from the Azure classic deployment model to Resource Manager.
- [Migrate](#) supported Resource Manager resources to Azure CSP.

Upgrade from classic to Resource Manager to move an existing subscription to Azure CSP

9/19/2017 • 2 min to read • [Edit Online](#)

If you have resources that you created by using the Azure classic deployment model, but which you've assessed as eligible to move to Azure Cloud Solution Provider (Azure CSP), you need to transfer your resources to Azure Resource Manager before you can migrate them to Azure CSP. You can use the following process to migrate existing resources without any downtime.

Migrate Azure classic deployment model resources to Resource Manager within the same source subscription by using subscription administrator permissions. In the classic deployment model, the process requires service administrator or co-administrator permissions.

The migration process involves three major steps:

1. **Move VMs.** You can do this by using PowerShell or the Azure command-line interface (Azure CLI).
2. **Move dependencies that are in the same resource group.** You also do this by using PowerShell.
3. **Remove the "secrets" dependency.** After the VMs are in Resource Manager, you must remove a secrets dependency that is part of Windows Remote Management (WinRM).

The migration process creates three resource groups:

- **One for each cloud service.** Name the resource group *[cloud service]-migrated*. This resource group contains all the VMs in that cloud service.
- **One for the virtual network.** Name the resource group *[virtual network]-migrated*.
- **One for storage accounts.** Name the resource group *[storage account]-migrated*.

When you've finished this migration process, you can move resources between resource groups. You can configure names and permissions without any downtime in service.

Move VMs

For detailed information about moving your VMs by using PowerShell, see [Migrate resources from classic to Azure Resource Manager by using Azure PowerShell](#).

For detailed information about moving your VMs by using Azure CLI, see [Migrate resources from classic to Azure Resource Manager by using Azure CLI](#).

Move dependencies

For detailed information about moving dependencies within a resource group by using PowerShell or Azure CLI, see [Move resources to a new resource group or subscription](#).

Remove the secrets dependency

VMs that you originally created by using the classic deployment model have a specific secrets dependency that you must remove before you can move the VMs to an Azure CSP subscription. You can remove these dependencies by using one of the following options:

- PowerShell
- Azure CLI

- Azure Resource Explorer

PowerShell

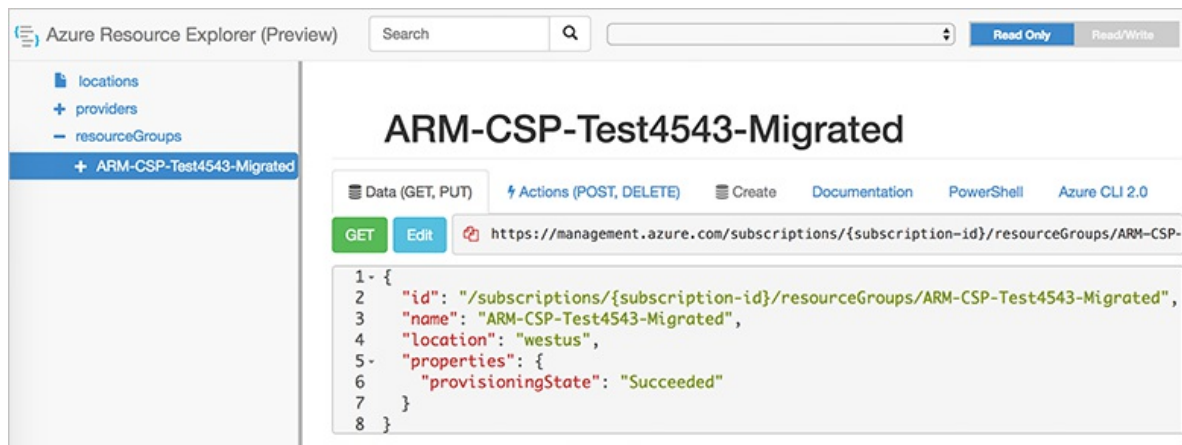
```
$vm=Get-AzureRmVM -ResourceGroupName $rgName -Name $vmName
$vm.OSProfile.Secrets = New-Object -TypeName
[System.Collections.Generic.List[Microsoft.Azure.Management.Compute.Models.VaultSecretGroup]]
Update-AzureRmVM -ResourceGroupName $rgName -VM $vm -Debug
```

Azure CLI

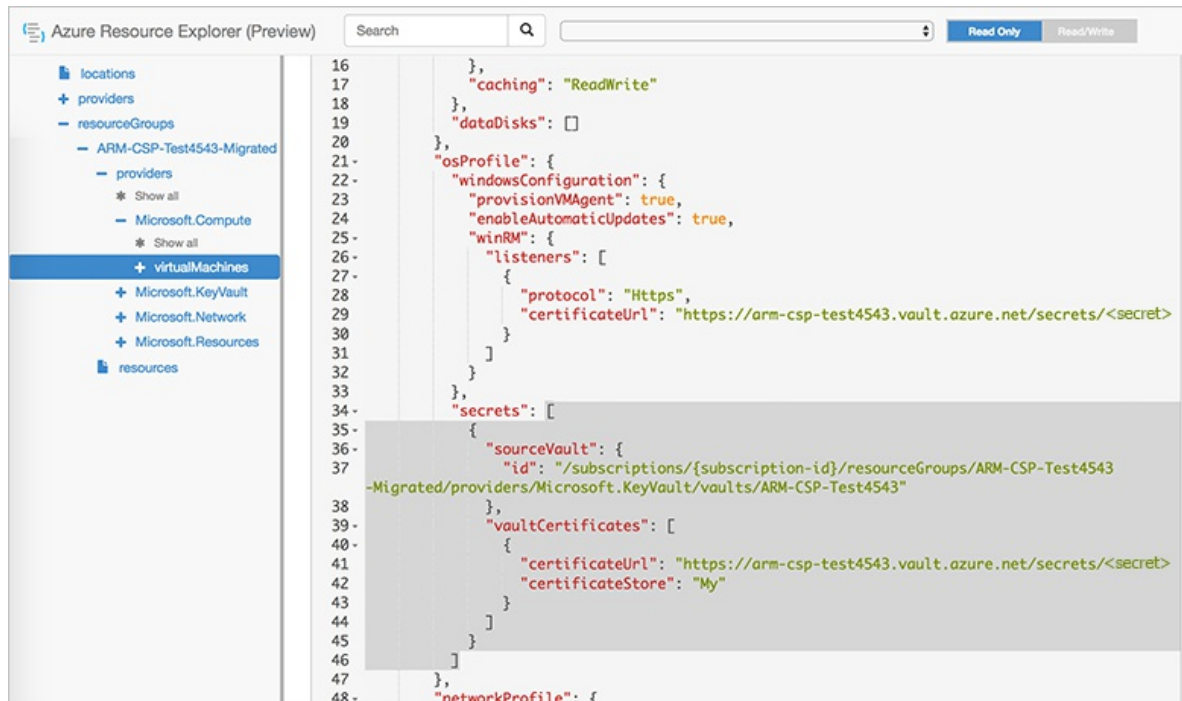
```
az vm update -g rgname -n vmname --set osProfile.Secrets=[]
```

Azure Resource Explorer

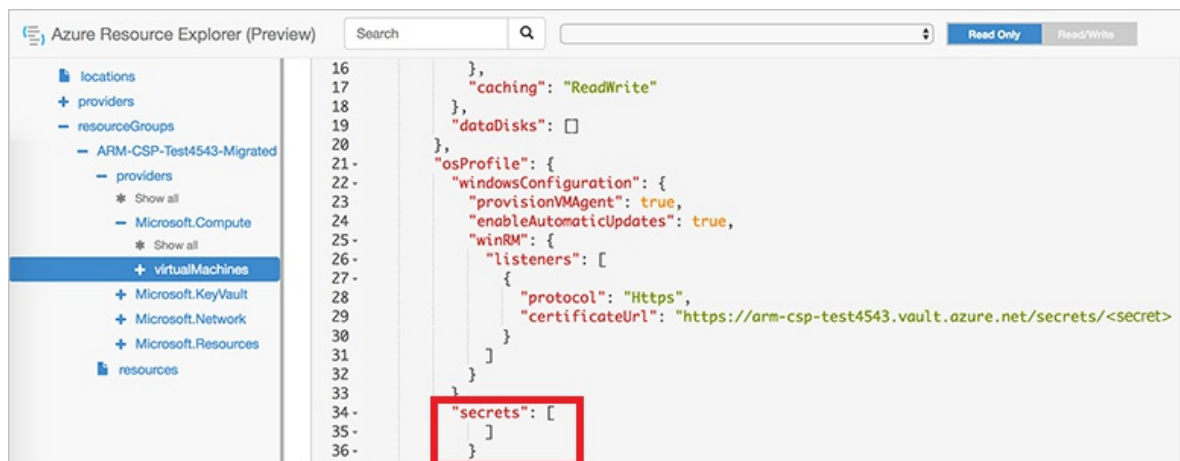
1. Go to resources.azure.com, and then select **subscriptions** > **resourceGroups**. Find the name of the resource group where the objects were migrated.



2. For each VM, select the **Microsoft.Compute** resource, remove the secrets values (highlighted in the following screenshot), and then select the **PUT** button.



3. When the secrets dependency has been removed, the page looks similar to this:



4. Repeat this process for all migrated VMs.

Next steps

- Do an [assessment](#) to identify the caveats for migrating your Azure subscription to Azure CSP.
- [Migrate](#) supported Resource Manager resources to Azure CSP.

Use Azure Resource Move to migrate an EA or PAYG subscription to Azure CSP

9/19/2017 • 3 min to read • [Edit Online](#)

After you've [upgraded all Azure classic deployment model resources](#) to the Azure Resource Manager model, you can migrate the resources from an Enterprise Agreement (EA) or pay-as-you-go (PAYG) subscription to an Azure Cloud Solution Provider (Azure CSP) subscription by using [Azure Resource Move](#).

NOTE

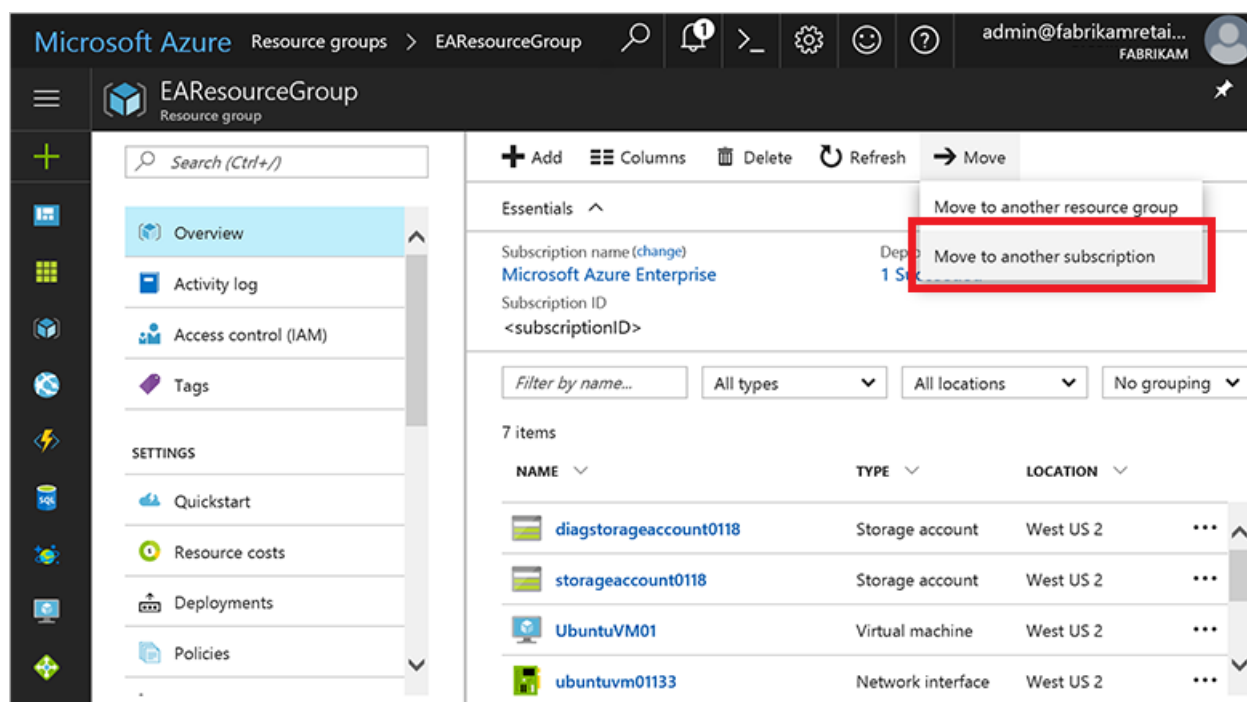
Resource Move is a detailed process. If you don't want to [perform this upgrade](#), you can leave the resources as is and migrate only those that were created in Resource Manager mode.

Although the following scenario shows the procedure for a single subscription, the procedure also applies to multiple subscriptions.

In the procedure, the term *source subscription* refers to the EA, Open, or PAYG subscription that is to be migrated. The term *destination subscription* refers to the Azure CSP subscription.

Move resources from the source subscription to the Azure CSP subscription

Choose a resource group in the source subscription that you want to migrate to the Azure CSP subscription. Select **Move**, and then select **Move to another subscription**.



Cross-subscription migrations require resources and all their dependencies to be moved at once. If you have VM resources in one resource group, and that VM's disks are stored in a storage account in another resource group, first move all dependent resources to a single resource group, and then migrate them to another subscription at the same time.

Select all resources that you want to migrate from the source subscription to the Azure CSP subscription, and

then select a resource group in the destination subscription. After you select **OK**, the migration possibility is validated. Select **OK** again to confirm the migration of selected resources from the source subscription to the Azure CSP subscription.

Move resources

EAResourceGroup


[Learn more](#)

Resources to move


☒

Select all


☒

 UbuntuVM01
Virtual machine


☒

 ubuntuvm01133
Network interface


☒

 UbuntuVM01-nsg
Network security group


☒

 UbuntuVM01-ip
Public IP address

☒

 vNet001
Virtual network

☒

 diagstorageaccount0118
Storage account

Move these resources to

* Subscription

Microsoft Azure CSP

* Resource group

CSPResourceGroup


Create a new group

☒

I understand that tools and scripts associated with moved resources will not work until I update them to use new resource IDs ⓘ

OK

Migration usually takes from several seconds to a few minutes. To ensure that all resources are migrated without errors, check the activity status.



Moving resources complete 14:50

Successfully moved 7 resources from resource group 'EAResourceGroup' in subscription 'Microsoft Azure Enterprise' to resource group 'CSPResourceGroup' in subscription 'Microsoft Azure CSP'

At this point, all resources in the selected resource group have been moved. In this example, the resource group includes the VMs and all storage accounts, public IPs, and NICs that are associated with the running VMs. After

the migration is complete, the source resource group is displayed as empty.

Resource Move is not supported by all Azure services

Some Azure services do not support the Resource Move feature. For a list of supported services and workaround suggestions for services that are not supported, see [Move resources to new resource group or subscription](#).

Known errors

During the migration, you might encounter one or more known errors, which are discussed in the following sections.

Dependency errors

The platform-native Resource Move feature requires that you place all objects and their dependencies in the same resource group.

If any resource with a dependency on other objects isn't in the same resource group as those objects (for example, a storage account that isn't in the same resource group as the VM), Resource Move displays an error when you start the migration process.

NOTE

Because the migration process takes place on the management plane, this error has no impact on the service itself. In this example, the migration will fail but the VM will continue to work.

You can move the object (a storage account, for example) within the resource group, and then you can restart the migration between subscriptions.

You can move resources between resource groups by using the native platform-migration feature. To access the feature, open the resource group, and then select the **Move** button. On the **Move resources** blade, select the resources that you want to move, select the correct resource group, and then move the resources. Again, because the migration process takes place on the management plane, no changes are made to the data itself.

Storage account-location errors

When you move a storage account and multiple other resources from a different region, you might see this error:

ERROR DETAILS

The move resources request contains resources that cannot be found. Please check details for these resource ids. (Code: MoveResourcesNotFound)

WAS THIS HELPFUL?  

You must migrate the storage account separately from the other resources. Clear the check box next to the storage account that's based in a different location, and then move the other resources. After the move is complete, you can move the storage account successfully. Alternatively, you can choose to move the storage account first.

Fabric errors

Another type of error might result from fabric changes that occur at the time of the migration:

ERROR DETAILS

The Microsoft.Compute resource provider cannot perform this operation currently due to system upgrade. Please retry after some time. (Code: OperationNotAllowed)

After you've completed the fabric changes, you can complete the migration successfully. In the tested case, the migration succeeded after about six hours. No downtime was involved.

Next steps

- [Assess the migration](#) to understand the issues that you identify during the process.
- Read an overview of [Azure CSP migration scenarios](#).

Migrate Azure subscriptions from pay-as-you-go to Azure CSP

9/19/2017 • 1 min to read • [Edit Online](#)

Here is how to assess and migrate existing Azure subscriptions from the pay-as-you-go program (sometimes called Azure Direct) to the Azure Cloud Solution Provider (Azure CSP) program.

Process overview

Overall, the migration process comprises the following steps:

1. Sign in to [Partner Center portal](#) using CSP partner credentials and create a [new customer account](#).
2. By default, pay-as-you-go Azure subscriptions are using Microsoft Account for sign in. You cannot use a subscription that is associated with a Microsoft Account in CSP. Follow the [Transfer ownership of an Azure subscription to another account](#) guide, and move it to an Azure AD-based directory, created during step 1.
3. [Perform an assessment](#): Assess Azure resources in the pay-as-you-go subscription that you want to move to Azure CSP. This helps you determine what you can and can't move to Azure CSP, and what moving a service entails.

NOTE

Some Azure services are not available in Azure CSP. It's better to know as early as possible whether these services need to be restructured or cannot be migrated.

4. [Upgrade Azure classic deployment model to Azure Resource Manager](#): If a customer is still using resources that they created by using the Azure classic deployment model, the resources must be transferred to the Azure Resource Manager model before you move them to Azure CSP. This includes any virtual machines, virtual networks, and storage accounts that were created by using the Azure classic deployment model.

NOTE

Azure CSP supports only Azure Resource Manager services. Azure classic deployment services cannot be migrated to Azure CSP.

5. [Move resources from target subscription to Azure CSP](#): When all of the customer's services are using the Azure Resource Manager model, you can perform the final step of migrating them to an Azure CSP subscription.

Next steps

- Review the [Azure subscription migration process from the Enterprise Agreement to Azure CSP](#).
- Learn [how to migrate customers from on-premises resources to Azure CSP](#).