

Contents

Azure Stack Hub Operator Documentation

Overview

[What is Azure Stack Hub?](#)

[Comparing the Azure ecosystem](#)

[Release notes](#)

[Security updates](#)

[Release notes](#)

[Known issues](#)

Quickstarts

[Use the administrator portal](#)

Tutorials

[Offer a service to users](#)

[Test a service offering](#)

Concepts

[Capacity Planning](#)

[Overview](#)

[Compute](#)

[Storage](#)

[Azure Stack Hub Capacity Planner](#)

[Datacenter integration](#)

[Walkthrough](#)

[General considerations](#)

[Connection models](#)

[Connected deployment](#)

[Disconnected deployment](#)

[Datacenter network integration](#)

[Network integration](#)

[Border connectivity](#)

[Switch configuration](#)

- [DNS integration](#)
 - [Configure DNS forwarder](#)
 - [Configure time server](#)
 - [Firewall integration](#)
 - [Publish endpoints](#)
 - [Extension host](#)
- [Deployment worksheet](#)
- [Identity integration](#)
 - [Choose an identity provider](#)
 - [Create registration role](#)
 - [Validate Azure identity](#)
 - [Validate ADFS integration](#)
 - [Validate graph integration](#)
- [Azure Stack Hub deployment PKI requirements](#)
 - [Generate PKI certificates](#)
 - [Prepare Azure Stack Hub PKI certificates](#)
 - [Validate PKI certificates](#)
 - [Remediate PKI certificates](#)
- [Deployment resources](#)
 - [About deployment networking](#)
 - [Validate Azure registration](#)
 - [Start-AzsReadiness cmdlet reference](#)
 - [View the readiness report](#)
 - [External monitoring](#)
 - [External auditing](#)
 - [Syslog forwarding](#)
 - [Device auditing](#)
- [How-to guides](#)
 - [Manage Azure Stack Hub](#)
 - [Administration basics](#)
 - [Clear user data from portal](#)
 - [Configure telemetry collection](#)

[Register Azure Stack Hub](#)

[Region management](#)

[Connect to Azure using ExpressRoute](#)

[Enable CLI for users](#)

[Set up management environment](#)

[Install Azure Stack Hub PowerShell](#)

[Download tools](#)

[Connect with PowerShell](#)

[Access privileged endpoint](#)

[Manage updates](#)

[Manage updates overview](#)

[Updates and servicing policy](#)

[Checklist for applying updates](#)

[Prepare the update package](#)

[Apply updates](#)

[Apply updates](#)

[Apply OEM updates](#)

[Monitor update status](#)

[Use PowerShell](#)

[Use privileged endpoint](#)

[Troubleshoot updates](#)

[Monitor health and alerts](#)

[Monitor hardware components](#)

[Manage network resources](#)

[Change user subscription owner](#)

[Start and stop Azure Stack Hub](#)

[Manage capacity](#)

[Manage storage resources](#)

[Manage storage capacity](#)

[Manage storage infrastructure](#)

[Manage physical memory capacity](#)

[Add scale unit nodes](#)

- [Add public IP addresses](#)
- [Replace hardware](#)
 - [Node actions](#)
 - [Replace node](#)
 - [Replace disk](#)
 - [Replace component](#)
- [Security, compliance & identity](#)
 - [Security overview](#)
 - [Configure security](#)
 - [Identity](#)
 - [Identity overview](#)
 - [Identity architecture](#)
 - [Manage RBAC](#)
 - [Add users for Azure AD](#)
 - [Add users for AD FS](#)
 - [Use an app identity to access resources](#)
 - [Enable multi-tenancy](#)
 - [Data at rest encryption](#)
 - [Rotate secrets](#)
 - [Update antivirus](#)
 - [Log collection and data handling](#)
- [Populate the marketplace](#)
 - [Marketplace overview](#)
 - [Marketplace changes](#)
 - [Azure Marketplace items for Azure Stack Hub](#)
 - [Download marketplace items](#)
 - [Add a custom virtual machine image](#)
 - [Create and publish a marketplace item](#)
 - [Supported guest operating systems for Azure Stack Hub](#)
 - [Offer a virtual machine scale set](#)
 - [Windows Server in Marketplace FAQ](#)
 - [Offer Linux virtual machines](#)

- [Add a Red Hat virtual machine](#)
- [Offer the AKS engine](#)
- [Offer Kubernetes marketplace item](#)
- [Create a site-to-site VPN connection](#)
- [Offer Commvault marketplace item](#)
- [Offer services](#)
 - [Service, plan, offer, subscription overview](#)
 - [Create a plan](#)
 - [Create an offer](#)
 - [Create an add-on plan](#)
 - [Subscribe to an offer](#)
 - [Delete offers](#)
 - [Delegate offers in Azure Stack Hub](#)
 - [Quota types](#)
- [Offer network solutions](#)
- [Manage value-add resource providers](#)
 - [App Service on Azure Stack Hub](#)
 - [Overview](#)
 - [App Service capacity planning](#)
 - [Before you get started](#)
 - [Deploy App Service](#)
 - [Deploy App Service for high availability](#)
 - [Deploy App Service offline](#)
 - [Update App Service](#)
 - [Update App Service Offline](#)
 - [Add App Service infrastructure roles](#)
 - [Configure deployment sources](#)
 - [Rotate App Service secrets and certificates](#)
 - [Back up App Service](#)
 - [Recover App Service](#)
 - [App Service billing FAQ](#)
 - [App Service on Azure Stack Hub release notes](#)

[App Service on Azure Stack Hub update 1 release notes](#)

[App Service on Azure Stack Hub update 2 release notes](#)

[App Service on Azure Stack Hub update 3 release notes](#)

[App Service on Azure Stack Hub update 4 release notes](#)

[App Service on Azure Stack Hub update 5 release notes](#)

[App Service on Azure Stack Hub update 6 release notes](#)

[App Service on Azure Stack Hub update 7 release notes](#)

[App Service on Azure Stack Hub update 8 release notes](#)

MySQL on Azure Stack Hub

[Overview](#)

[Deploy the MySQL Server resource provider](#)

[Add MySQL hosting servers](#)

[Create MySQL Databases](#)

[Deploy highly available MySQL databases](#)

[Update the MySQL Server resource provider](#)

[Maintain MySQL Server resource provider](#)

[Remove the MySQL Server resource provider](#)

[MySQL resource provider release notes](#)

[MySQL resource provider 1.1.47.0 release notes](#)

[MySQL resource provider 1.1.33.0 release notes](#)

[MySQL resource provider 1.1.30.0 release notes](#)

SQL on Azure Stack Hub

[Overview](#)

[Deploy the SQL Server resource provider](#)

[Add SQL hosting servers](#)

[Create SQL databases](#)

[Deploy highly available SQL databases](#)

[Update the SQL Server resource provider](#)

[Maintain the SQL Server resource provider](#)

[Remove the SQL Server resource provider](#)

[SQL resource provider release notes](#)

[SQL resource provider 1.1.47.0 release notes](#)

[SQL resource provider 1.1.33.0 release notes](#)

[SQL resource provider 1.1.30.0 release notes](#)

Usage and billing

[Manage usage and billing as a CSP](#)

[Add tenants for usage and billing](#)

[Manage tenant registration](#)

Usage and billing reference

[Usage data reporting](#)

[Usage reporting infrastructure](#)

[Provider usage API](#)

[Tenant usage API](#)

[Usage FAQ](#)

[Troubleshooting usage issues](#)

[Usage connectivity errors](#)

[Usage registration error codes](#)

Back up Azure Stack Hub

[Enable Backup for Azure Stack Hub from the administration console](#)

[Enable Backup for Azure Stack Hub with PowerShell](#)

[Back up Azure Stack Hub](#)

[Recover data with the backup service](#)

[Back up files and applications on Azure Stack Hub](#)

[Replicate Azure Stack Hub VMs to Azure](#)

[Recover from catastrophic data loss](#)

[Backup best practices](#)

[Backup reference](#)

Get support

[Help & Support portal](#)

[Overview](#)

[Diagnostic log collection](#)

[Overview](#)

[Automatic collection](#)

[Configure automatic collection](#)

- [Best practices](#)
- [On-demand collection](#)
- [Validation for Azure Stack Hub](#)
- [Troubleshooting](#)
- [Reference](#)
 - [Azure Stack Hub PowerShell](#)
 - [Azure Stack Hub User documentation](#)
 - [ASDK documentation](#)
- [Resources](#)
 - [Training and certification](#)
 - [Azure Stack Hub roadmap](#)
 - [Azure Stack Hub MSDN forum](#)
 - [Azure Stack Hub Documentation feedback](#)
 - [Pricing information](#)
 - [Stack Overflow](#)

Azure Stack Hub overview

7 minutes to read • [Edit Online](#)

Azure Stack Hub is an extension of Azure that provides a way to run apps in an on-premises environment and deliver Azure services in your datacenter. With a consistent cloud platform, organizations can confidently make technology decisions based on business requirements, rather than business decisions based on technology limitations.

Why use Azure Stack Hub?

Azure provides a rich platform for developers to build modern apps. However, some cloud-based apps face obstacles like latency, intermittent connectivity, and regulations. Azure and Azure Stack Hub unlock new hybrid cloud use cases for both customer-facing and internal line-of-business apps:

- **Edge and disconnected solutions.** Address latency and connectivity requirements by processing data locally in Azure Stack Hub and then aggregating it in Azure for further analytics, with common app logic across both. You can even deploy Azure Stack Hub disconnected from the internet without connectivity to Azure. Think of factory floors, cruise ships, and mine shafts as examples.
- **Cloud apps that meet varied regulations.** Develop and deploy apps in Azure with full flexibility to deploy on-premises with Azure Stack Hub to meet regulatory or policy requirements. No code changes are needed. App examples include global audit, financial reporting, foreign exchange trading, online gaming, and expense reporting.
- **Cloud app model on-premises.** Use Azure services, containers, serverless, and microservice architectures to update and extend existing apps or build new ones. Use consistent DevOps processes across Azure in the cloud and Azure Stack Hub on-premises to speed up app modernization for core mission-critical apps.

Azure Stack Hub architecture

Azure Stack Hub integrated systems are comprised in racks of 4-16 servers built by trusted hardware partners and delivered straight to your datacenter. After delivery, a solution provider will work with you to deploy the integrated system and ensure the Azure Stack Hub solution meets your business requirements. You can prepare your datacenter by ensuring all required power and cooling, border connectivity, and other required datacenter integration requirements are in place.

For more information about the Azure Stack Hub datacenter integration experience, see [Azure Stack Hub datacenter integration](#).

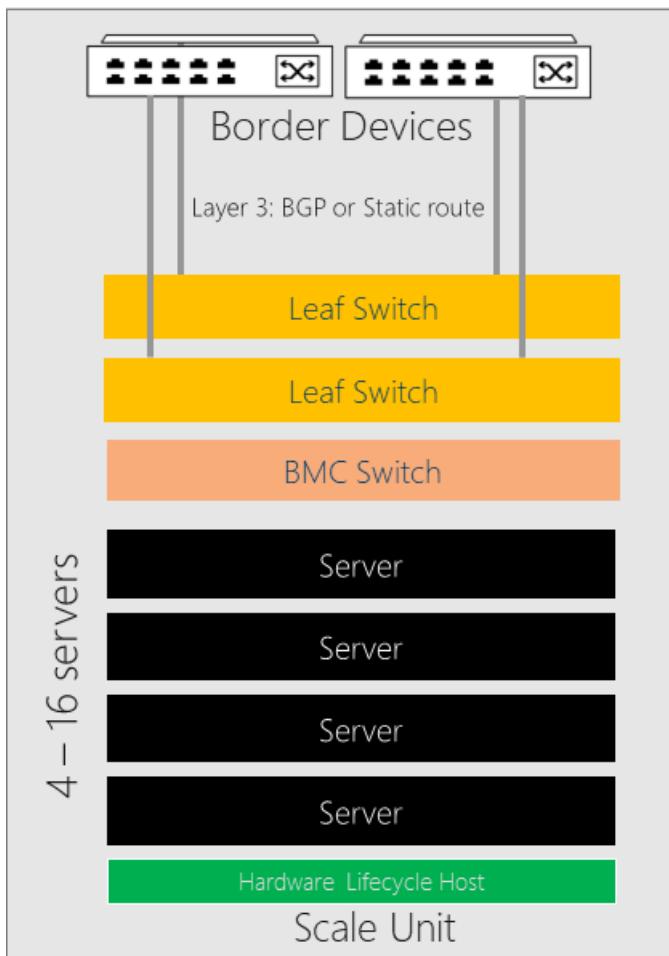
Azure Stack Hub is built on industry standard hardware and is managed using the same tools you already use for managing Azure subscriptions. As a result, you can apply consistent DevOps processes whether you're connected to Azure or not.

The Azure Stack Hub architecture lets you provide Azure services at the edge for remote locations or intermittent connectivity, disconnected from the internet. You can create hybrid solutions that process data locally in Azure Stack Hub and then aggregate it in Azure for additional processing and analytics. Finally, because Azure Stack Hub is installed on-premises, you can meet specific regulatory or policy requirements with the flexibility of deploying cloud apps on-premises without changing any code.

Deployment options

Azure Stack Hub integrated systems are offered through a partnership of Microsoft and hardware partners, creating a solution that offers cloud-paced innovation and computing management simplicity. Because Azure Stack Hub is offered as an integrated hardware and software system, you have the flexibility and control you need, along with the ability to innovate from the cloud.

An Azure Stack Hub integrated system can range in size from 4-16 servers, called a *scale unit*. Integrated systems are jointly supported by the hardware partner and Microsoft. The following diagram shows an example of a scale unit.



Connection models

You can choose to deploy Azure Stack Hub either **connected** to the internet (and to Azure) or **disconnected** from it.

For more information, see the considerations for [connected](#) and [disconnected](#) deployment models.

Identity provider

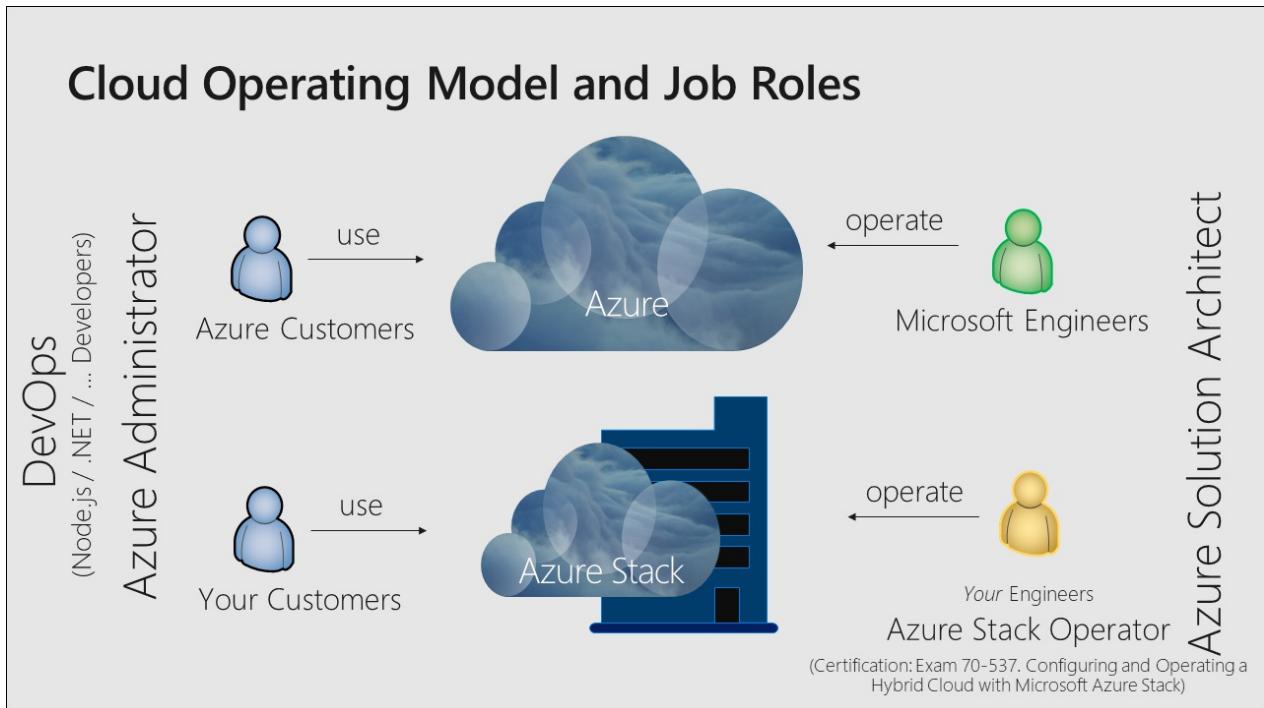
Azure Stack Hub uses either Azure Active Directory (Azure AD) or Active Directory Federation Services (AD FS). Azure AD is Microsoft's cloud-based, multi-tenant identity provider. Most hybrid scenarios with internet-connected deployments use Azure AD as the identity store.

For disconnected deployments of Azure Stack Hub, you need to use AD FS. Azure Stack Hub resource providers and other apps work similarly with AD FS or Azure AD. Azure Stack Hub includes its own Active Directory instance and an Active Directory Graph API.

How is Azure Stack Hub managed?

Azure Stack Hub uses the same operations model as Azure. An Azure Stack Hub operator can deliver a variety of

services and apps to tenant users, similar to how Microsoft delivers Azure services to tenant users.



You can manage Azure Stack Hub with the administrator portal, user portal, or [PowerShell](#). The Azure Stack Hub portals are each backed by separate instances of Azure Resource Manager. An **Azure Stack Hub Operator** uses the administrator portal to manage Azure Stack Hub, and to do things like create tenant offerings and maintain the health and monitor status of the integrated system. The user portal provides a self-service experience for consumption of cloud resources like virtual machines (VMs), storage accounts, and web apps.

For more information about managing Azure Stack Hub using the administrator portal, see the use the [Azure Stack Hub administration portal quickstart](#).

As an Azure Stack Hub operator, you can deliver [VMs](#), [web apps](#), highly available [SQL Server](#), and [MySQL Server](#) databases. You can also use [Azure Stack Hub quickstart Azure Resource Manager templates](#) to deploy SharePoint, Exchange, and more.

An operator can manage Azure Stack Hub with the [administrator portal](#) or [PowerShell](#). You can configure Azure Stack Hub to [deliver services](#) to tenants using plans, quotas, offers, and subscriptions. Tenant users can subscribe to multiple offers. Offers can have one or more plans, and plans can have one or more services. Operators also manage capacity and respond to alerts.

Users consume services that the operator offers. Users can provision, monitor, and manage services that they've subscribed to, like web apps, storage, and VMs. Users can manage Azure Stack Hub with the user portal or PowerShell.

To learn more about managing Azure Stack Hub, including what accounts to use where, typical operator responsibilities, what to tell your users, and how to get help, review [Azure Stack Hub administration basics](#).

Resource providers

Resource providers are web services that form the foundation for all Azure Stack Hub IaaS and PaaS services. Azure Resource Manager relies on different resource providers to provide access to services. Each resource provider helps you configure and control its respective resources. Service admins can also add new custom resource providers.

Foundational resource providers

There are three foundational IaaS resource providers:

- **Compute:** The Compute Resource Provider lets Azure Stack Hub tenants to create their own VMs. The Compute Resource Provider includes the ability to create VMs as well as VM extensions. The VM extension service helps provide IaaS capabilities for Windows and Linux VMs. As an example, you can use the Compute Resource Provider to provision a Linux VM and run Bash scripts during deployment to configure the VM.
- **Network Resource Provider:** The Network Resource Provider delivers a series of Software Defined Networking (SDN) and Network Function Virtualization (NFV) features for the private cloud. You can use the Network Resource Provider to create resources like software load balancers, public IPs, network security groups, and virtual networks.
- **Storage Resource Provider:** The Storage Resource Provider delivers four Azure-consistent storage services: [blob](#), [queue](#), [table](#), and [Key Vault](#) account management providing management and auditing of secrets, such as passwords and certificates. The storage resource provider also offers a storage cloud administration service to facilitate service provider administration of Azure-consistent storage services. Azure Storage provides the flexibility to store and retrieve large amounts of unstructured data, like documents and media files with Azure Blobs, and structured NoSQL based data with Azure Tables.

Optional resource providers

There are three optional PaaS resource providers that you can deploy and use with Azure Stack Hub:

- **App Service:** [Azure App Service on Azure Stack Hub](#) is a PaaS offering of Microsoft Azure available to Azure Stack Hub. The service enables your internal or external customers to create web, API, and Azure Functions apps for any platform or device.
- **SQL Server:** Use the [SQL Server resource provider](#) to offer SQL databases as a service of Azure Stack Hub. After you install the resource provider and connect it to one or more SQL Server instances, you and your users can create databases for cloud-native apps, websites that use SQL, and other workloads that use SQL.
- **MySQL Server:** Use the [MySQL Server resource provider](#) to expose MySQL databases as an Azure Stack Hub service. The MySQL resource provider runs as a service on a Windows Server 2019 Server Core VM.

Next steps

[Compare the Azure Stack Hub portfolio](#)

[Administration basics](#)

[Quickstart: use the Azure Stack Hub administration portal](#)

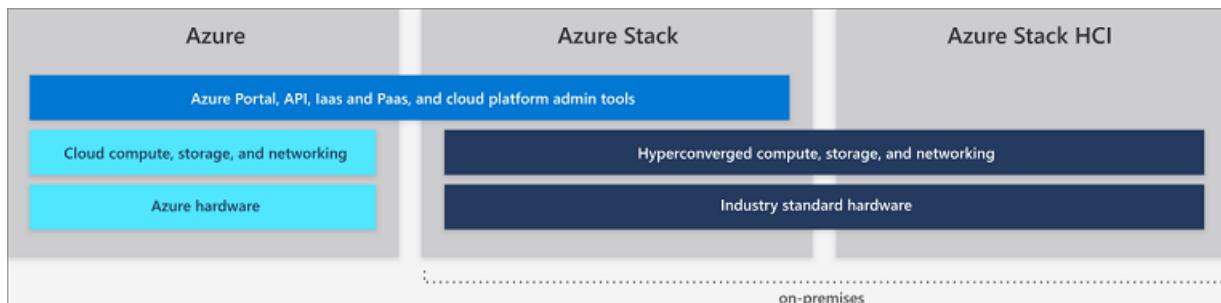
[Understand usage and billing.](#)

Differences between global Azure, Azure Stack Hub, and Azure Stack HCI

3 minutes to read • [Edit Online](#)

Microsoft provides Azure and the Azure Stack Hub family of services in one Azure ecosystem. Use the same application model, self-service portals, and APIs with Azure Resource Manager to deliver cloud-based capabilities whether your business uses global Azure or on-premises resources.

This article describes the differences between global Azure, Azure Stack Hub, and Azure Stack HCI capabilities. It provides common scenario recommendations to help you make the best choice for delivering Microsoft cloud-based services for your organization.



Global Azure

Microsoft Azure is an ever-expanding set of cloud services to help your organization meet your business challenges. It's the freedom to build, manage, and deploy apps on a massive, global network using your favorite tools and frameworks.

Global Azure offers more than 100 services available in 54 regions around the globe. For the most current list of global Azure services, see the [Products available by region](#). The services available in Azure are listed by category and also by whether they're generally available or available through preview.

For more information about global Azure services, see [Get started with Azure](#).

Azure Stack Hub

Azure Stack Hub is an extension of Azure that brings the agility and innovation of cloud computing to your on-premises environment. Deployed on-premises, Azure Stack Hub can be used to provide Azure consistent services either connected to the internet (and Azure) or in disconnected environments with no internet connectivity. Azure Stack Hub uses the same underlying technologies as global Azure, which includes the core components of Infrastructure-as-a-Service (IaaS), Software-as-a-Service (SaaS), and optional Platform-as-a-Service (PaaS) capabilities. These capabilities include:

- Azure VMs for Windows and Linux
- Azure Web Apps and Functions
- Azure Key Vault
- Azure Resource Manager
- Azure Marketplace
- Containers
- Admin tools (Plans, offers, RBAC, and so on)

The PaaS capabilities of Azure Stack Hub are optional because Azure Stack Hub isn't operated by Microsoft—it's operated by our customers. This means you can offer whatever PaaS service you want to end users if you're prepared to abstract the underlying infrastructure and processes away from the end user. However, Azure Stack Hub does include several optional PaaS service providers including App Service, SQL databases, and MySQL databases. These are delivered as resource providers so they're multi-tenant ready, updated over time with standard Azure Stack Hub updates, visible in the Azure Stack Hub portal, and well integrated with Azure Stack Hub.

In addition to the resource providers described above, there are additional PaaS services available and tested as [Azure Resource Manager template-based solutions](#) that run in IaaS. As an Azure Stack Hub operator, you can offer them as PaaS services to your users including:

- Service Fabric
- Kubernetes Container Service
- Ethereum Blockchain
- Cloud Foundry

Example use cases for Azure Stack Hub:

- Financial modeling
- Clinical and claims data
- IoT device analytics
- Retail assortment optimization
- Supply-chain optimization
- Industrial IoT
- Predictive maintenance
- Smart city
- Citizen engagement

Learn more about Azure Stack Hub at [What is Azure Stack Hub](#).

Azure Stack HCI

[Azure Stack HCI](#) solutions let you run virtual machines on-premises and easily connect to Azure with a hyperconverged infrastructure (HCI) solution. Build and run cloud apps using consistent Azure services on-premises to meet regulatory or technical requirements. In addition to running virtualized apps on-premises, Azure Stack HCI lets you replace and consolidate aging server infrastructure and connect to Azure for cloud services using the Windows Admin Center.

Azure Stack HCI provides validated HCI solutions powered by Hyper-V and Storage Spaces Direct with Windows Server 2019 Software-Defined Datacenter (SDDC). The Windows Admin Center is used for management and integrated access to Azure services such as:

- Azure Backup
- Azure Site Recovery
- Azure Monitor and Update

For an updated list of Azure services that you can connect Azure Stack HCI to, see [Connecting Windows Server to Azure hybrid services](#).

Example use cases for Azure Stack HCI

- Remote or branch office systems
- Datacenter consolidation
- Virtual desktop infrastructure

- Business-critical infrastructure
- Lower-cost storage
- High availability and disaster recovery in the cloud
- Enterprise apps like SQL Server

Visit the [Azure Stack HCI website](#) to view 70+ Azure Stack HCI solutions currently available from Microsoft partners.

Next steps

[Azure Stack Hub administration basics](#)

[Quickstart: use the Azure Stack Hub administration portal](#)

Azure Stack Hub security updates

2 minutes to read • [Edit Online](#)

This article lists all the security updates in the last four updates of Azure Stack Hub. This information is provided for reference purposes only.

1910 update

This update of Azure Stack Hub does not include security updates to the underlying operating system that hosts Azure Stack Hub. This update contains a security update for a component of the Azure Stack Hub infrastructure:

- [CVE-2019-1234](#)

1908 update

- [CVE-2019-1214](#)
- [CVE-2019-1215](#)
- [CVE-2019-1219](#)
- [CVE-2019-1232](#)
- [CVE-2019-1235](#)
- [CVE-2019-1240](#)
- [CVE-2019-1241](#)
- [CVE-2019-1242](#)
- [CVE-2019-1243](#)
- [CVE-2019-1244](#)
- [CVE-2019-1245](#)
- [CVE-2019-1246](#)
- [CVE-2019-1247](#)
- [CVE-2019-1248](#)
- [CVE-2019-1249](#)
- [CVE-2019-1250](#)
- [CVE-2019-1251](#)
- [CVE-2019-1252](#)
- [CVE-2019-1253](#)
- [CVE-2019-1254](#)
- [CVE-2019-1256](#)
- [CVE-2019-1267](#)
- [CVE-2019-1268](#)
- [CVE-2019-1269](#)
- [CVE-2019-1270](#)
- [CVE-2019-1271](#)
- [CVE-2019-1272](#)
- [CVE-2019-1273](#)
- [CVE-2019-1274](#)
- [CVE-2019-1277](#)
- [CVE-2019-1278](#)

- [CVE-2019-1280](#)
- [CVE-2019-1282](#)
- [CVE-2019-1285](#)
- [CVE-2019-1286](#)
- [CVE-2019-1287](#)
- [CVE-2019-1289](#)
- [CVE-2019-1290](#)
- [CVE-2019-1291](#)
- [CVE-2019-1292](#)
- [CVE-2019-1293](#)
- [CVE-2019-1294](#)
- [CVE-2019-1303](#)
- [CVE-2019-0714](#)
- [CVE-2019-0715](#)
- [CVE-2019-0716](#)
- [CVE-2019-0717](#)
- [CVE-2019-0718](#)
- [CVE-2019-0720](#)
- [CVE-2019-0723](#)
- [CVE-2019-0965](#)
- [CVE-2019-1057](#)
- [CVE-2019-1078](#)
- [CVE-2019-1143](#)
- [CVE-2019-1144](#)
- [CVE-2019-1145](#)
- [CVE-2019-1146](#)
- [CVE-2019-1147](#)
- [CVE-2019-1148](#)
- [CVE-2019-1149](#)
- [CVE-2019-1150](#)
- [CVE-2019-1151](#)
- [CVE-2019-1152](#)
- [CVE-2019-1153](#)
- [CVE-2019-1155](#)
- [CVE-2019-1156](#)
- [CVE-2019-1157](#)
- [CVE-2019-1158](#)
- [CVE-2019-1159](#)
- [CVE-2019-1162](#)
- [CVE-2019-1163](#)
- [CVE-2019-1164](#)
- [CVE-2019-1168](#)
- [CVE-2019-1170](#)
- [CVE-2019-1171](#)
- [CVE-2019-1172](#)
- [CVE-2019-1173](#)

- [CVE-2019-1174](#)
- [CVE-2019-1175](#)
- [CVE-2019-1176](#)
- [CVE-2019-1177](#)
- [CVE-2019-1178](#)
- [CVE-2019-1179](#)
- [CVE-2019-1180](#)
- [CVE-2019-1181](#)
- [CVE-2019-1182](#)
- [CVE-2019-1183](#)
- [CVE-2019-1184](#)
- [CVE-2019-1186](#)
- [CVE-2019-1187](#)
- [CVE-2019-1188](#)
- [CVE-2019-1190](#)
- [CVE-2019-1198](#)
- [CVE-2019-1206](#)
- [CVE-2019-1212](#)
- [CVE-2019-1222](#)
- [CVE-2019-1223](#)
- [CVE-2019-1224](#)
- [CVE-2019-1225](#)
- [CVE-2019-1226](#)
- [CVE-2019-1227](#)
- [CVE-2019-9506](#)
- [CVE-2019-9511](#)
- [CVE-2019-9512](#)
- [CVE-2019-9513](#)
- [CVE-2019-9514](#)
- [CVE-2019-9518](#)
- [CVE-2019-1125](#)
- [CVE-2019-0785](#)
- [CVE-2019-0811](#)
- [CVE-2019-0865](#)
- [CVE-2019-0880](#)
- [CVE-2019-0887](#)
- [CVE-2019-0966](#)
- [CVE-2019-0975](#)
- [CVE-2019-1006](#)
- [CVE-2019-1037](#)
- [CVE-2019-1067](#)
- [CVE-2019-1071](#)
- [CVE-2019-1073](#)
- [CVE-2019-1074](#)
- [CVE-2019-1085](#)
- [CVE-2019-1086](#)

- [CVE-2019-1087](#)
- [CVE-2019-1088](#)
- [CVE-2019-1089](#)
- [CVE-2019-1090](#)
- [CVE-2019-1091](#)
- [CVE-2019-1093](#)
- [CVE-2019-1094](#)
- [CVE-2019-1095](#)
- [CVE-2019-1096](#)
- [CVE-2019-1097](#)
- [CVE-2019-1102](#)
- [CVE-2019-1108](#)
- [CVE-2019-1117](#)
- [CVE-2019-1118](#)
- [CVE-2019-1119](#)
- [CVE-2019-1120](#)
- [CVE-2019-1121](#)
- [CVE-2019-1122](#)
- [CVE-2019-1123](#)
- [CVE-2019-1124](#)
- [CVE-2019-1126](#)
- [CVE-2019-1127](#)
- [CVE-2019-1128](#)
- [CVE-2019-1129](#)
- [CVE-2019-1130](#)
- [ADV190016](#)
- [CVE-2019-0620](#)
- [CVE-2019-0710](#)
- [CVE-2019-0711](#)
- [CVE-2019-0713](#)
- [CVE-2019-0722](#)
- [CVE-2019-0888](#)
- [CVE-2019-0904](#)
- [CVE-2019-0905](#)
- [CVE-2019-0906](#)
- [CVE-2019-0907](#)
- [CVE-2019-0908](#)
- [CVE-2019-0909](#)
- [CVE-2019-0941](#)
- [CVE-2019-0943](#)
- [CVE-2019-0948](#)
- [CVE-2019-0959](#)
- [CVE-2019-0972](#)
- [CVE-2019-0973](#)
- [CVE-2019-0974](#)
- [CVE-2019-0983](#)

- [CVE-2019-0984](#)
- [CVE-2019-0986](#)
- [CVE-2019-0998](#)
- [CVE-2019-1007](#)
- [CVE-2019-1010](#)
- [CVE-2019-1012](#)
- [CVE-2019-1014](#)
- [CVE-2019-1017](#)
- [CVE-2019-1018](#)
- [CVE-2019-1019](#)
- [CVE-2019-1021](#)
- [CVE-2019-1022](#)
- [CVE-2019-1025](#)
- [CVE-2019-1026](#)
- [CVE-2019-1027](#)
- [CVE-2019-1028](#)
- [CVE-2019-1039](#)
- [CVE-2019-1040](#)
- [CVE-2019-1041](#)
- [CVE-2019-1043](#)
- [CVE-2019-1044](#)
- [CVE-2019-1046](#)
- [CVE-2019-1050](#)
- [CVE-2019-1053](#)
- [CVE-2019-1064](#)
- [CVE-2019-1065](#)
- [CVE-2019-1069](#)

Because of the cumulative nature of Windows updates, this update also contains the security payloads for the previous months. This list of CVEs reflects the security payload of June, July, August and September 2019. For more information about these vulnerabilities, click on the preceding links, or see Microsoft Knowledge Base articles [4516077](#).

1907 update

This update of Azure Stack Hub does not include security updates to the underlying operating system that hosts Azure Stack Hub.

1906 update

This update of Azure Stack Hub does not include security updates to the underlying operating system that hosts Azure Stack Hub.

Next steps

- [Review update activity checklist](#)
- [Review list of known issues](#)

Azure Stack Hub release notes

28 minutes to read • [Edit Online](#)

This article describes the contents of Azure Stack Hub update packages. The update includes improvements and fixes for the latest release of Azure Stack Hub.

To access release notes for a different version, use the version selector dropdown above the table of contents on the left.

IMPORTANT

This update package is only for Azure Stack Hub integrated systems. Don't apply this update package to the Azure Stack Development Kit (ASDK).

IMPORTANT

If your Azure Stack Hub instance is behind by more than two updates, it's considered out of compliance. You must [update to at least the minimum supported version to receive support](#).

Update planning

Before applying the update, make sure to review the following information:

- [Known issues](#)
- [Security updates](#)
- [Checklist of activities before and after applying the update](#)

For help with troubleshooting updates and the update process, see [Troubleshoot patch and update issues for Azure Stack Hub](#).

1910 build reference

The Azure Stack Hub 1910 update build number is **1.1910.0.58**.

Update type

Starting with 1908, the underlying operating system on which Azure Stack Hub runs was updated to Windows Server 2019. This update enables core fundamental enhancements and the ability to bring additional capabilities to Azure Stack Hub.

The Azure Stack Hub 1910 update build type is **Express**.

The 1910 update package is larger in size compared to previous updates, which results in longer download times. The update will remain in the **Preparing** state for a long time and operators can expect this process to take longer than previous updates. The expected time for the 1910 update to complete is approximately 10 hours, regardless of the number of physical nodes in your Azure Stack Hub environment. Exact update runtimes typically depend on the capacity used on your system by tenant workloads, your system network connectivity (if connected to the internet), and your system hardware specifications. Runtimes lasting longer than the expected value aren't uncommon and don't require action by Azure Stack Hub operators unless the update fails. This runtime approximation is specific to the 1910 update and shouldn't be compared to other Azure Stack Hub updates.

For more information about update build types, see [Manage updates in Azure Stack Hub](#).

What's new

- The administrator portal now shows the privileged endpoint IP addresses in the region properties menu for easier discovery. In addition, it shows the current configured time server and DNS forwarders. For more information, see [Use the privileged endpoint in Azure Stack Hub](#).
- The Azure Stack Hub health and monitoring system can now raise alerts for various hardware components if an error happens. These alerts require additional configuration. For more information, see [Monitor Azure Stack Hub hardware components](#).
- **Cloud-init support for Azure Stack Hub:** Cloud-init is a widely used approach to customize a Linux VM as it boots for the first time. You can use cloud-init to install packages and write files, or to configure users and security. Because cloud-init is called during the initial boot process, there are no additional steps or required agents to apply your configuration. The Ubuntu images on the marketplace have been updated to support cloud-init for provisioning.
- Azure Stack Hub now supports all Windows Azure Linux Agent versions as Azure.
- A new version of Azure Stack Hub admin PowerShell modules is available.
- Added the **Set-AzSDefenderManualUpdate** cmdlet in the privileged endpoint (PEP) to configure the manual update for Windows Defender definitions in the Azure Stack Hub infrastructure. For more information, see [Update Windows Defender Antivirus on Azure Stack Hub](#).
- Added the **Get-AzSDefenderManualUpdate** cmdlet in the privileged endpoint (PEP) to retrieve the configuration of the manual update for Windows Defender definitions in the Azure Stack Hub infrastructure. For more information, see [Update Windows Defender Antivirus on Azure Stack Hub](#).
- Added the **Set-AzSDnsForwarder** cmdlet in the privileged endpoint (PEP) to change the forwarder settings of the DNS servers in Azure Stack Hub. For more information about DNS configuration, see [Azure Stack Hub datacenter DNS integration](#).
- Added the **Get-AzSDnsForwarder** cmdlet in the privileged endpoint (PEP) to retrieve the forwarder settings of the DNS servers in Azure Stack Hub. For more information about DNS configuration, see [Azure Stack Hub datacenter DNS integration](#).
- Added support for management of **Kubernetes clusters** using the [AKS engine](#). Starting with this update, customers can deploy production Kubernetes clusters. The AKS engine enables users to:
 - Manage the life cycle of their Kubernetes clusters. They can create, update, and scale clusters.
 - Maintain their clusters using managed images produced by the AKS and the Azure Stack Hub teams.
 - Take advantage of an Azure Resource Manager-integrated Kubernetes cloud provider that builds clusters using native Azure resources.
 - Deploy and manage their clusters in connected or disconnected Azure Stack Hub stamps.
 - Use Azure hybrid features:
 - Integration with Azure Arc.
 - Integration with Azure Monitor for Containers.
 - Use Windows Containers with AKS engine.
 - Receive CSS and engineering support for their deployments.

Improvements

- Azure Stack Hub has improved its ability to auto-remediate some patch and update issues that previously caused update failures or prevented operators from being able to initiate an Azure Stack Hub update. As a result, there are fewer tests included in the **Test-AzureStack -UpdateReadiness** group. For more information, see [Validate Azure Stack Hub system state](#). The following three tests remain in the

UpdateReadiness group:

- **AzSInfraFileValidation**
- **AzSActionPlanStatus**
- **AzsStampBMCSummary**
- Added an auditing rule to report when an external device (for example, a USB key) is mounted to a node of the Azure Stack Hub infrastructure. The audit log is emitted via syslog and will be displayed as **Microsoft-Windows-Security-Auditing: 6416|Plug and Play Events**. For more information about how to configure the syslog client, see [Syslog forwarding](#).
- Azure Stack Hub is moving to 4096-bit RSA keys for the internal certificates. Running internal secret rotation will replace old 2048-bit certificates with 4096-bit long certificates. For more information about secret rotation in Azure Stack Hub, see [Rotate secrets in Azure Stack Hub](#).
- Upgrades to the complexity of cryptographic algorithms and key strength for several internal components to comply with the Committee on National Security Systems - Policy 15 (CNSSP-15), which provides best practices for the use of public standards for secure information sharing. Among the improvements, there's AES256 for Kerberos authentication and SHA384 for VPN encryption. For more information about CNSSP-15, see the [Committee on National Security Systems, Policies page](#).
- Because of the above upgrade, Azure Stack Hub now has new default values for IPsec/IKEv2 configurations. The new default values used on the Azure Stack Hub side are as follows:

IKE Phase 1 (Main Mode) parameters

PROPERTY	VALUE
IKE Version	IKEv2
Diffie-Hellman Group	ECP384
Authentication method	Pre-shared key
Encryption & Hashing Algorithms	AES256, SHA384
SA Lifetime (Time)	28,800 seconds

IKE Phase 2 (Quick Mode) parameters

PROPERTY	VALUE
IKE Version	IKEv2
Encryption & Hashing Algorithms (Encryption)	GCMAES256
Encryption & Hashing Algorithms (Authentication)	GCMAES256
SA Lifetime (Time)	27,000 seconds
SA Lifetime (Kilobytes)	33,553,408
Perfect Forward Secrecy (PFS)	ECP384
Dead Peer Detection	Supported

These changes are reflected in the [default IPsec/IKE proposal](#) documentation as well.

- The infrastructure backup service improves logic that calculates desired free space for backups instead of relying on a fixed threshold. The service will use the size of a backup, retention policy, reserve, and current utilization of external storage location to determine if a warning needs to be raised to the operator.

Changes

- When downloading marketplace items from Azure to Azure Stack Hub, there's a new user interface that enables you to specify a version of the item when multiple versions exist. The new UI is available in both connected and disconnected scenarios. For more information, see [Download marketplace items from Azure to Azure Stack Hub](#).
- Starting with the 1910 release, the Azure Stack Hub system **requires** an additional /20 private internal IP space. This network is private to the Azure Stack Hub system and can be reused on multiple Azure Stack Hub systems within your datacenter. While the network is private to Azure Stack Hub, it must not overlap with a network in your datacenter. The /20 private IP space is divided into multiple networks that enable running the Azure Stack Hub infrastructure on containers (as previously mentioned in the [1905 release notes](#)). The goal of running the Azure Stack Hub infrastructure in containers is to optimize utilization and enhance performance. In addition, the /20 private IP space is also used to enable ongoing efforts that will reduce required routable IP space before deployment.
 - Please note that the /20 input serves as a prerequisite to the next Azure Stack Hub update after 1910. When the next Azure Stack Hub update after 1910 releases and you attempt to install it, the update will fail if you haven't completed the /20 input as described in the remediation steps as follows. An alert will be present in the administrator portal until the above remediation steps have been completed. See the [Datacenter network integration](#) article to understand how this new private space will be consumed.
 - Remediation steps: To remediate, follow the instructions to [open a PEP Session](#). Prepare a [private internal IP range](#) of size /20, and run the following cmdlet (only available starting with 1910) in the PEP session using the following example: `Set-AzsPrivateNetwork -UserSubnet 100.87.0.0/20`. If the operation is performed successfully, you'll receive the message **Azs Internal Network range added to the config**. If successfully completed, the alert will close in the administrator portal. The Azure Stack Hub system can now update to the next version.
- The infrastructure backup service deletes partially uploaded backup data if the external storage location runs out of capacity during the upload procedure.
- The infrastructure backup service adds identity service to the backup payload for AAD deployments.
- The Azure Stack Hub PowerShell Module has been updated to version 1.8.0 for the 1910 release. Changes include:
 - **New DRP Admin module:** The Deployment Resource Provider (DRP) enables orchestrated deployments of resource providers to Azure Stack Hub. These commands interact with the Azure Resource Manager layer to interact with DRP.
 - **BRP:**
 - Support single role restore for Azures stack infrastructure backup.
 - Add parameter `RoleName` to cmdlet `Restore-AzsBackup`.
 - **FRP:** Breaking changes for **Drive** and **Volume** resources with API version `2019-05-01`. The features are supported by Azure Stack Hub 1910 and later:
 - The value of `ID`, `Name`, `HealthStatus`, and `OperationalStatus` have been changed.
 - Supported new properties `FirmwareVersion`, `IsIndicationEnabled`, `Manufacturer`, and `StoragePool` for **Drive** resources.
 - The properties `CanPool` and `CannotPoolReason` of **Drive** resources have been deprecated; use

`OperationalStatus` instead.

Fixes

- Fixed an issue that prevented enforcing TLS 1.2 policy on environments deployed before the Azure Stack Hub 1904 release.
- Fixed an issue where an Ubuntu 18.04 VM created with SSH authorization enabled doesn't allow you to use the SSH keys to sign in.
- Removed **Reset Password** from the Virtual Machine Scale Set UI.
- Fixed an issue where deleting the load balancer from the portal didn't result in the deletion of the object in the infrastructure layer.
- Fixed an issue that showed an inaccurate percentage of the Gateway Pool utilization alert on the administrator portal.

Security updates

For information about security updates in this update of Azure Stack Hub, see [Azure Stack Hub security updates](#).

Update planning

Before applying the update, make sure to review the following information:

- [Known issues](#)
- [Security updates](#)
- [Checklist of activities before and after applying the update](#)

Download the update

You can download the Azure Stack Hub 1910 update package from [the Azure Stack Hub download page](#).

Hotfixes

Azure Stack Hub releases hotfixes on a regular basis. Be sure to install the latest Azure Stack Hub hotfix for 1908 before updating Azure Stack Hub to 1910.

Azure Stack Hub hotfixes are only applicable to Azure Stack Hub integrated systems; don't attempt to install hotfixes on the ASDK.

Prerequisites: Before applying the 1910 update

The 1910 release of Azure Stack Hub must be applied on the 1908 release with the following hotfixes:

- [Azure Stack Hub hotfix 1.1908.19.62](#)

After successfully applying the 1910 update

After the installation of this update, install any applicable hotfixes. For more information, see our [servicing policy](#).

- [Azure Stack Hub hotfix 1.1910.24.108](#)

1908 build reference

The Azure Stack Hub 1908 update build number is **1.1908.4.33**.

Update type

For 1908, the underlying operating system on which Azure Stack Hub runs has been updated to Windows Server 2019. This update enables core fundamental enhancements and the ability to bring additional capabilities to Azure Stack Hub.

The Azure Stack Hub 1908 update build type is **Full**. As a result, the 1908 update has a longer runtime than express updates like 1906 and 1907. Exact runtimes for full updates typically depend on the number of nodes that your Azure Stack Hub instance contains, the capacity used on your system by tenant workloads, your system's network connectivity (if connected to the internet), and your system hardware configuration. The 1908 update has had the following expected runtimes in our internal testing: 4 nodes - 42 hours, 8 nodes - 50 hours, 12 nodes - 60 hours, 16 nodes - 70 hours. Update runtimes lasting longer than these expected values aren't uncommon and don't require action by Azure Stack Hub operators unless the update fails.

For more information about update build types, see [Manage updates in Azure Stack Hub](#).

- Exact update runtimes typically depend on the capacity used on your system by tenant workloads, your system network connectivity (if connected to the internet), and your system hardware configuration.
- Runtimes lasting longer than expected aren't uncommon and don't require action by Azure Stack Hub operators unless the update fails.
- This runtime approximation is specific to the 1908 update and shouldn't be compared to other Azure Stack Hub updates.

What's new

- For 1908, note that the underlying operating system on which Azure Stack Hub runs has been updated to Windows Server 2019. This update enables core fundamental enhancements and the ability to bring additional capabilities to Azure Stack Hub.
- All components of Azure Stack Hub infrastructure now operate in FIPS 140-2 mode.
- Azure Stack Hub operators can now remove portal user data. For more information, see [Clear portal user data from Azure Stack Hub](#).

Improvements

- Improvements to data at rest encryption of Azure Stack Hub to persist secrets into the hardware Trusted Platform Module (TPM) of the physical nodes.

Changes

- Hardware providers will be releasing OEM extension package 2.1 or later at the same time as Azure Stack Hub version 1908. The OEM extension package 2.1 or later is a prerequisite for Azure Stack Hub version 1908. For more information about how to download OEM extension package 2.1 or later, contact your system's hardware provider, and see the [OEM updates](#) article.

Fixes

- Fixed an issue with compatibility with future Azure Stack Hub OEM updates and an issue with VM deployment using customer user images. This issue was found in 1907 and fixed in hotfix [KB4517473](#)
- Fixed an issue with OEM Firmware update and corrected misdiagnosis in Test-AzureStack for Fabric Ring Health. This issue was found in 1907 and fixed in hotfix [KB4515310](#)
- Fixed an issue with OEM Firmware update process. This issue was found in 1907 and fixed in hotfix [KB4515650](#)

Security updates

For information about security updates in this update of Azure Stack Hub, see [Azure Stack Hub security updates](#).

Download the update

You can download the Azure Stack Hub 1908 update package from the [Azure Stack Hub download page](#).

Hotfixes

Azure Stack Hub releases hotfixes on a regular basis. Be sure to install the latest Azure Stack Hub hotfix for 1907

before updating Azure Stack Hub to 1908.

Azure Stack Hub hotfixes are only applicable to Azure Stack Hub integrated systems; don't attempt to install hotfixes on the ASDK.

Prerequisites: Before applying the 1908 update

The 1908 release of Azure Stack Hub must be applied on the 1907 release with the following hotfixes:

- [Azure Stack Hub hotfix 1.1907.26.70](#)

The Azure Stack Hub 1908 Update requires **Azure Stack Hub OEM version 2.1 or later** from your system's hardware provider. OEM updates include driver and firmware updates to your Azure Stack Hub system hardware. For more information about applying OEM updates, see [Apply Azure Stack Hub original equipment manufacturer updates](#)

After successfully applying the 1908 update

After the installation of this update, install any applicable hotfixes. For more information, see our [servicing policy](#).

- [Azure Stack Hub hotfix 1.1908.19.62](#)

1907 build reference

The Azure Stack Hub 1907 update build number is **1.1907.0.20**.

Update type

The Azure Stack Hub 1907 update build type is **Express**. For more information about update build types, see the [Manage updates in Azure Stack Hub](#) article. Based on internal testing, the expected time it takes for the 1907 update to complete is approximately 13 hours.

- Exact update runtimes typically depend on the capacity used on your system by tenant workloads, your system network connectivity (if connected to the internet), and your system hardware configuration.
- Runtimes lasting longer than expected aren't uncommon and don't require action by Azure Stack Hub operators unless the update fails.
- This runtime approximation is specific to the 1907 update and shouldn't be compared to other Azure Stack Hub updates.

What's in this update

What's new

- General availability release of the Azure Stack Hub diagnostic log collection service to facilitate and improve diagnostic log collection. The Azure Stack Hub diagnostic log collection service provides a simplified way to collect and share diagnostic logs with Microsoft Customer Support Services (CSS). This diagnostic log collection service provides a new user experience in the Azure Stack Hub administrator portal that enables operators to set up the automatic upload of diagnostic logs to a storage blob when certain critical alerts are raised. The service can also be used to perform the same operation on demand. For more information, see the [Diagnostic log collection](#) article.
- General availability release of the Azure Stack Hub network infrastructure validation as a part of the Azure Stack Hub validation tool **Test-AzureStack**. Azure Stack Hub network infrastructure will be a part of **Test-AzureStack**, to identify if a failure occurs on the network infrastructure of Azure Stack Hub. The test checks connectivity of the network infrastructure by bypassing the Azure Stack Hub software-defined network. It demonstrates connectivity from a public VIP to the configured DNS forwarders, NTP servers, and identity endpoints. It also checks for connectivity to Azure when using Azure AD as the identity provider, or the federated server when using ADFS. For more information, see the [Azure Stack Hub validation tool](#) article.

- Added an internal secret rotation procedure to rotate internal SQL TLS certificates as required during a system update.

Improvements

- The Azure Stack Hub update blade now displays a **Last Step Completed** time for active updates. This addition can be seen by going to the update blade and clicking on a running update. **Last Step Completed** is then available in the **Update run details** section.
- Improvements to **Start-AzureStack** and **Stop-AzureStack** operator actions. The time to start Azure Stack Hub has been reduced by an average of 50%. The time to shut down Azure Stack Hub has been reduced by an average of 30%. The average startup and shutdown times remain the same as the number of nodes increases in a scale-unit.
- Improved error handling for the disconnected Marketplace tool. If a download fails or partially succeeds when using **Export-AzSOFFlineMarketplaceItem**, a detailed error message displays with more details about the error and any possible mitigation steps.
- Improved the performance of managed disk creation from a large page blob/snapshot. Previously, it triggered a timeout when creating a large disk.
- Improved virtual disk health check before shutting down a node to avoid unexpected virtual disk detaching.
- Improved storage of internal logs for administrator operations. This addition results in improved performance and reliability during administrator operations by minimizing the memory and storage consumption of internal log processes. You might also notice improved page load times of the update blade in the administrator portal. As part of this improvement, update logs older than six months will no longer be available in the system. If you require logs for these updates, be sure to [Download the summary](#) for all update runs older than six months before performing the 1907 update.

Changes

- Azure Stack Hub version 1907 contains a warning alert that instructs operators to be sure to update their system's OEM package to version 2.1 or later before updating to version 1908. For more information about how to apply Azure Stack Hub OEM updates, see [Apply an Azure Stack Hub original equipment manufacturer update](#).
- Added a new outbound rule (HTTPS) to enable communication for Azure Stack Hub diagnostic log collection service. For more information, see [Azure Stack Hub datacenter integration - Publish endpoints](#).
- The infrastructure backup service now deletes partially uploaded backups if the external storage location runs out of capacity.
- Infrastructure backups no longer include a backup of domain services data. This change only applies to systems using Azure Active Directory as the identity provider.
- We now validate that an image being ingested into the **Compute -> VM images** blade is of type page blob.
- The privileged endpoint command **Set-BmcCredential** now updates the credential in the Baseboard Management Controller.

Fixes

- Fixed an issue in which the publisher, offer, and SKU were treated as case sensitive in a Resource Manager template: the image wasn't fetched for deployment unless the image parameters were the same case as that of the publisher, offer, and SKU.
- Fixed an issue with backups failing with a **PartialSucceeded** error message, due to timeouts during

backup of storage service metadata.

- Fixed an issue in which deleting user subscriptions resulted in orphaned resources.
- Fixed an issue in which the description field wasn't saved when creating an offer.
- Fixed an issue in which a user with **Read only** permissions was able to create, edit, and delete resources. Now the user is only able to create resources when the **Contributor** permission is assigned.
- Fixed an issue in which the update fails due to a DLL file locked by the WMI provider host.
- Fixed an issue in the update service that prevented available updates from displaying in the update tile or resource provider. This issue was found in 1906 and fixed in hotfix [KB4511282](#).
- Fixed an issue that could cause updates to fail due to the management plane becoming unhealthy due to a bad configuration. This issue was found in 1906 and fixed in hotfix [KB4512794](#).
- Fixed an issue that prevented users from completing deployment of third-party images from the marketplace. This issue was found in 1906 and fixed in hotfix [KB4511259](#).
- Fixed an issue that could cause VM creation from managed images to fail due to our user image manager service crashing. This issue was found in 1906 and fixed in hotfix [KB4512794](#)
- Fixed an issue in which VM CRUD operations could fail due to the app gateway cache not being refreshed as expected. This issue was found in 1906 and fixed in hotfix [KB4513119](#)
- Fixed an issue in the health resource provider which impacted the availability of the region and alert blades in the administrator portal. This issue was found in 1906 and fixed in hotfix [KB4512794](#).

Security updates

For information about security updates in this update of Azure Stack Hub, see [Azure Stack Hub security updates](#).

Update planning

Before applying the update, make sure to review the following information:

- [Known issues](#)
- [Security updates](#)
- [Checklist of activities before and after applying the update](#)

Download the update

You can download the Azure Stack Hub 1907 update package from [the Azure Stack Hub download page](#).

Hotfixes

Azure Stack Hub releases hotfixes on a regular basis. Be sure to install the latest Azure Stack Hub hotfix for 1906 before updating Azure Stack Hub to 1907.

Azure Stack Hub hotfixes are only applicable to Azure Stack Hub integrated systems; don't attempt to install hotfixes on the ASDK.

Before applying the 1907 update

The 1907 release of Azure Stack Hub must be applied on the 1906 release with the following hotfixes:

- [Azure Stack Hub hotfix 1.1906.15.60](#)

After successfully applying the 1907 update

After the installation of this update, install any applicable hotfixes. For more information, see our [servicing policy](#).

- [Azure Stack Hub hotfix 1.1907.26.70](#)

1906 build reference

The Azure Stack Hub 1906 update build number is **1.1906.0.30**.

Update type

The Azure Stack Hub 1906 update build type is **Express**. For more information about update build types, see the [Manage updates in Azure Stack Hub](#) article. The expected time it takes for the 1906 update to complete is approximately 10 hours, regardless of the number of physical nodes in your Azure Stack Hub environment. Exact update runtimes will typically depend on the capacity used on your system by tenant workloads, your system network connectivity (if connected to the internet), and your system hardware specifications. Runtimes lasting longer than the expected value aren't uncommon and don't require action by Azure Stack Hub operators unless the update fails. This runtime approximation is specific to the 1906 update and shouldn't be compared to other Azure Stack Hub updates.

What's in this update

- Added a **Set-TLSPolicy** cmdlet in the privileged endpoint (PEP) to force TLS 1.2 on all the endpoints. For more information, see [Azure Stack Hub security controls](#).
- Added a **Get-TLSPolicy** cmdlet in the privileged endpoint (PEP) to retrieve the applied TLS policy. For more information, see [Azure Stack Hub security controls](#).
- Added an internal secret rotation procedure to rotate internal TLS certificates as required during a system update.
- Added a safeguard to prevent expiration of internal secrets by forcing internal secrets rotation in case a critical alert on expiring secrets is ignored. This safeguard shouldn't be relied on as a regular operating procedure. Secrets rotation should be planned during a maintenance window. For more information, see [Azure Stack Hub secret rotation](#).
- Visual Studio Code is now supported with Azure Stack Hub deployment using AD FS.

Improvements

- The **Get-GraphApplication** cmdlet in the privileged endpoint now displays the thumbprint of the currently used certificate. This update improves the certificate management for service principals when Azure Stack Hub is deployed with AD FS.
- New health monitoring rules have been added to validate the availability of AD Graph and AD FS, including the ability to raise alerts.
- Improvements to the reliability of the backup resource provider when the infrastructure backup service moves to another instance.
- Performance optimization of external secret rotation procedure to provide a uniform execution time to facilitate scheduling of maintenance window.
- The **Test-AzureStack** cmdlet now reports on internal secrets that are about to expire (critical alerts).
- A new parameter is available for the **Register-CustomAdfs** cmdlet in the privileged endpoint that enables skipping the certificate revocation list checking when configuring the federation trust for AD FS.
- The 1906 release introduces greater visibility into update progress, so you can be assured that updates aren't pausing. This update results in an increase in the total number of update steps shown to operators in the **Update** blade. You might also notice more update steps happening in parallel than in previous

updates.

Networking updates

- Updated lease time set in DHCP responder to be consistent with Azure.
- Improved retry rates to the resource provider in the scenario of failed deployment of resources.
- Removed the **Standard** SKU option from both the load balancer and public IP, as that is currently not supported.

Changes

- Creating a storage account experience is now consistent with Azure.
- Changed alert triggers for expiration of internal secrets:
 - Warning alerts are now raised 90 days before the expiration of secrets.
 - Critical alerts are now raised 30 days before the expiration of secrets.
- Updated strings in infrastructure backup resource provider for consistent terminology.

Fixes

- Fixed an issue where resizing a managed disk VM failed with an **Internal Operation Error**.
- Fixed an issue where a failed user image creation puts the service that manages images in a bad state; this blocks deletion of the failed image and creation of new images. This issue is also fixed in the 1905 hotfix.
- Active alerts on expiring internal secrets are now automatically closed after successful execution of internal secret rotation.
- Fixed an issue in which the update duration in the update history tab would trim the first digit if the update was running for more than 99 hours.
- The **Update** blade includes a **Resume** option for failed updates.
- In the administrator and user portals, fixed the issue in marketplace in which the Docker extension was incorrectly returned from search but no further action could be taken, as it isn't available in Azure Stack Hub.
- Fixed an issue in template deployment UI that doesn't populate parameters if the template name begins with '_' underscore.
- Fixed an issue where the virtual machine scale set creation experience provides CentOS-based 7.2 as an option for deployment. CentOS 7.2 isn't available on Azure Stack Hub. We now provide Centos 7.5 as our option for deployment
- You can now remove a scale set from the **Virtual machine scale sets** blade.

Security updates

For information about security updates in this update of Azure Stack Hub, see [Azure Stack Hub security updates](#).

Update planning

Before applying the update, make sure to review the following information:

- [Known issues](#)
- [Security updates](#)
- [Checklist of activities before and after applying the update](#)

Download the update

You can download the Azure Stack Hub 1906 update package from [the Azure Stack Hub download page](#).

Hotfixes

Azure Stack Hub releases hotfixes on a regular basis. Be sure to install the latest Azure Stack Hub hotfix for 1905 before updating Azure Stack Hub to 1906. After updating, install any [available hotfixes for 1906](#).

Azure Stack Hub hotfixes are only applicable to Azure Stack Hub integrated systems; don't attempt to install hotfixes on the ASDK.

Before applying the 1906 update

The 1906 release of Azure Stack Hub must be applied on the 1905 release with the following hotfixes:

- [Azure Stack Hub hotfix 1.1905.3.48](#)

After successfully applying the 1906 update

After the installation of this update, install any applicable hotfixes. For more information, see our [servicing policy](#).

- [Azure Stack Hub hotfix 1.1906.15.60](#)

Automatic update notifications

Systems that can access the internet from the infrastructure network will see the **Update available** message in the operator portal. Systems without internet access can download and import the .zip file with the corresponding .xml.

TIP

Subscribe to the following *RSS* or *Atom* feeds to keep up with Azure Stack Hub hotfixes:

- [RSS](#)
- [Atom](#)

Archive

To access archived release notes for an older version, use the version selector dropdown above the table of contents on the left and select the version you want to see.

Next steps

- For an overview of the update management in Azure Stack Hub, see [Manage updates in Azure Stack Hub overview](#).
- For more information about how to apply updates with Azure Stack Hub, see [Apply updates in Azure Stack Hub](#).
- To review the servicing policy for Azure Stack Hub and what you must do to keep your system in a supported state, see [Azure Stack Hub servicing policy](#).
- To use the privileged endpoint (PEP) to monitor and resume updates, see [Monitor updates in Azure Stack Hub using the privileged endpoint](#).

1905 archived release notes

1904 archived release notes

[1903 archived release notes](#)

[1902 archived release notes](#)

[1901 archived release notes](#)

[1811 archived release notes](#)

[1809 archived release notes](#)

[1808 archived release notes](#)

[1807 archived release notes](#)

[1805 archived release notes](#)

[1804 archived release notes](#)

[1803 archived release notes](#)

[1802 archived release notes](#)

You can access [older versions of Azure Stack Hub release notes on the TechNet Gallery](#). These archived documents are provided for reference purposes only and don't imply support for these versions. For information about Azure Stack Hub support, see [Azure Stack Hub servicing policy](#). For further assistance, contact Microsoft Customer Support Services.

Azure Stack Hub known issues

28 minutes to read • [Edit Online](#)

This article lists known issues in releases of Azure Stack Hub. The list is updated as new issues are identified.

To access known issues for a different version, use the version selector dropdown above the table of contents on the left.

IMPORTANT

Review this section before applying the update.

IMPORTANT

If your Azure Stack Hub instance is behind by more than two updates, it's considered out of compliance. You must [update to at least the minimum supported version to receive support](#).

Update

For known Azure Stack Hub update issues please see [Troubleshooting Updates in Azure Stack Hub](#).

Portal

Administrative subscriptions

- Applicable: This issue applies to all supported releases.
- Cause: The two administrative subscriptions that were introduced with version 1804 should not be used. The subscription types are **Metering** subscription, and **Consumption** subscription.
- Remediation: If you have resources running on these two subscriptions, recreate them in user subscriptions.
- Occurrence: Common

Subscriptions Lock blade

- Applicable: This issue applies to all supported releases.
- Cause: In the administrator portal, the **Lock** blade for user subscriptions has two buttons that say **Subscription**.
- Occurrence: Common

Subscription permissions

- Applicable: This issue applies to all supported releases.
- Cause: You cannot view permissions to your subscription using the Azure Stack Hub portals.
- Remediation: Use [PowerShell to verify permissions](#).
- Occurrence: Common

Storage account settings

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, the storage account **Configuration** blade shows an option to change **security transfer type**. The feature is currently not supported in Azure Stack Hub.
- Occurrence: Common

Upload blob with OAuth error

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, when you try to upload a blob using the **OAuth(preview)** option, the task fails with an error message.
- Remediation: Upload the blob using the SAS option.
- Occurrence: Common

Upload blob option unsupported

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, when you try to upload a blob in the upload blade, there is an option to select **AAD** or **Key Authentication**, however **AAD** is not supported in Azure Stack Hub.
- Occurrence: Common

Load balancer backend pool

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, when adding a **Load balancer** backend pool, the operation results in an error message of **Failed to save load balancer backend pool**; however, the operation did actually succeed.
- Occurrence: Common

Alert for network interface disconnected

- Applicable: This issue applies to the 1908 and 1910 releases.
- Cause: When a cable is disconnected from a network adapter, an alert does not show in the administrator portal. This issue is caused because this fault is disabled by default in Windows Server 2019.
- Occurrence: Common

Incorrect tooltip when creating VM

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, when you select a managed disk, with disk type Premium SSD, the drop-down list shows **OS Disk**. The tooltip next to that option says **Certain OS Disk sizes may be available for free with Azure Free Account**; however, this is not valid for Azure Stack Hub. In addition, the list includes **Free account eligible** which is also not valid for Azure Stack Hub.
- Occurrence: Common

VPN troubleshoot and metrics

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, the **VPN Troubleshoot** feature and **Metrics** in a VPN gateway resource appears, however this is not supported in Azure Stack Hub.
- Occurrence: Common

Adding extension to VM Scale Set

- Applicable: This issue applies to releases 1907 and later.
- Cause: In the user portal, once a virtual machine scale set is created, the UI does not permit the user to add an extension.
- Occurrence: Common

Delete a storage container

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, when a user attempts to delete a storage container, the operation fails when the user does not toggle **Override Azure Policy and RBAC Role settings**.
- Remediation: Ensure that the box is checked for **Override Azure Policy and RBAC Role settings**.
- Occurrence: Common

Refresh button on virtual machines fails

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, when you navigate to **Virtual Machines** and try to refresh using the button at the top, the states fail to update accurately.
- Remediation: The status is automatically updated every 5 minutes regardless of whether the refresh button has been clicked or not. Wait 5 minutes and check the status.
- Occurrence: Common

Virtual Network Gateway

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, when you create a route table, **Virtual Network gateway** appears as one of the next hop type options; however, this is not supported in Azure Stack Hub.
- Occurrence: Common

Storage account options

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, the name of storage accounts is shown as **Storage account - blob, file, table, queue**; however, **file** is not supported in Azure Stack Hub.
- Occurrence: Common

Storage account configuration

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, when you create a storage account and view its **Configuration**, you cannot save configuration changes, as it results in an AJAX error.
- Occurrence: Common

Capacity monitoring in SQL resource provider keeps loading

- Applicable: This issue applies to the Azure Stack Hub 1910 update or later, with SQL resource provider version 1.1.33.0 or earlier installed.
- Cause: The current version of the SQL resource provider is not compatible with some of the latest portal changes in the 1910 update.
- Remediation: Follow the resource provider update process to apply the SQL resource provider hotfix 1.1.47.0 after Azure Stack Hub is upgraded to the 1910 update ([SQL RP version 1.1.47.0](#)). For the MySQL resource provider, it is also recommended that you apply the MySQL resource provider hotfix 1.1.47.0 after Azure Stack Hub is upgraded to 1910 update ([MySQL RP version 1.1.47.0](#)).
- Occurrence: Common

Access Control (IAM)

- Applicable: This issue applies to all supported releases.
- Cause: The IAM extension is out of date. The Ibiza portal that shipped with Azure Stack Hub introduces a new behavior that causes the RBAC extension to fail if the user is opening the **Access Control (IAM)** blade for a subscription that is not selected in the global subscription selector (**Directory + Subscription** in the user portal). The blade displays **Loading** in a loop, and the user cannot add new roles to the subscription. The **Add** blade also displays **Loading** in a loop.
- Remediation: Ensure that the subscription is checked in the **Directory + Subscription** menu. The menu can be accessed from the top of the portal, near the **Notifications** button, or via the shortcut on the **All resources** blade that displays **Don't see a subscription? Open Directory + Subscription settings**. The subscription must be selected in this menu.

SQL resource provider

- Applicable: This issue applies to stamps that are running 1908 or earlier.

- Cause: When deploying the SQL resource provider (RP) version 1.1.47.0, the portal shows no assets other than those associated with the SQL RP.
- Remediation: Delete the RP, upgrade the stamp, and re-deploy the SQL RP.

Networking

Load balancer

- Applicable: This issue applies to all supported releases.
- Cause: When adding availability set VMs to the backend pool of a load balancer, an error message is displayed on the portal stating **Failed to save load balancer backend pool**. This is a cosmetic issue on the portal; the functionality is still in place and VMs are successfully added to the backend pool internally.
- Occurrence: Common

Network Security Groups

- Applicable: This issue applies to all supported releases.
- Cause: An explicit **DenyAllOutbound** rule cannot be created in an NSG as this will prevent all internal communication to infrastructure needed for the VM deployment to complete.
- Occurrence: Common

Service endpoints

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, the **Virtual Network** blade shows an option to use **Service Endpoints**. This feature is currently not supported in Azure Stack Hub.
- Occurrence: Common

Network interface

Adding/removing network interface

- Applicable: This issue applies to all supported releases.
- Cause: A new network interface cannot be added to a VM that is in a **running** state.
- Remediation: Stop the virtual machine before adding or removing a network interface.
- Occurrence: Common

Primary network interface

- Applicable: This issue applies to all supported releases.
- Cause: The primary NIC of a VM cannot be changed. Deleting or detaching the primary NIC results in issues when starting up the VM.
- Occurrence: Common

Virtual Network Gateway

Alerts

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, the **Virtual Network Gateway** blade shows an option to use **Alerts**. This feature is currently not supported in Azure Stack Hub.
- Occurrence: Common

Active-Active

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, while creating, and in the resource menu of **Virtual Network Gateway**, you will see an option to enable **Active-Active** configuration. This feature is currently not supported in Azure Stack Hub.
- Occurrence: Common

VPN troubleshooter

- Applicable: This issue applies to all supported releases.

- Cause: In the user portal, the **Connections** blade displays a feature called **VPN Troubleshooter**. This feature is currently not supported in Azure Stack Hub.
- Occurrence: Common

Documentation

- Applicable: This issue applies to all supported releases.
- Cause: The documentation links in the overview page of Virtual Network gateway link to Azure-specific documentation instead of Azure Stack Hub. Use the following links for the Azure Stack Hub documentation:
 - [Gateway SKUs](#)
 - [Highly Available Connections](#)
 - [Configure BGP on Azure Stack Hub](#)
 - [ExpressRoute circuits](#)
 - [Specify custom IPsec/IKE policies](#)

Compute

VM boot diagnostics

- Applicable: This issue applies to all supported releases.
- Cause: When creating a new Windows virtual machine (VM), the following error might be displayed: **Failed to start virtual machine 'vm-name'. Error: Failed to update serial output settings for VM 'vm-name'**. The error occurs if you enable boot diagnostics on a VM, but delete your boot diagnostics storage account.
- Remediation: Recreate the storage account with the same name you used previously.
- Occurrence: Common

Consumed compute quota

- Applicable: This issue applies to all supported releases.
- Cause: When creating a new virtual machine, you may receive an error such as **This subscription is at capacity for Total Regional vCPUs on this location. This subscription is using all 50 Total Regional vCPUs available.** This indicates that the quota for total cores available to you has been reached.
- Remediation: Ask your operator for an add-on plan with additional quota. Editing the current plan's quota will not work or reflect increased quota.
- Occurrence: Rare

Privileged Endpoint

- Applicable: This issue applies to 1910 and earlier releases.
- Cause: Unable to connect to the Privileged Endpoint (ERC VMs) from a computer running a non-English version of Windows.
- Remediation: This is a known issue that has been fixed in releases later than 1910. As a workaround you can run the **New-PSSession** and **Enter-PSSession** Powershell cmdlets using the **en-US** culture; for example, set the culture using this script: <https://resources.oreilly.com/examples/9780596528492/blob/master/Use-Culture.ps1>.
- Occurrence: Rare

Virtual machine scale set

Create failures during patch and update on 4-node Azure Stack Hub environments

- Applicable: This issue applies to all supported releases.
- Cause: Creating VMs in an availability set of 3 fault domains and creating a virtual machine scale set instance fails with a **FabricVmPlacementErrorUnsupportedFaultDomainSize** error during the update process on a 4-node Azure Stack Hub environment.

- Remediation: You can create single VMs in an availability set with 2 fault domains successfully. However, scale set instance creation is still not available during the update process on a 4-node Azure Stack Hub deployment.

1908 update process

- Applicable: This issue applies to all supported releases.
- Cause: When attempting to install the Azure Stack Hub update, the status for the update might fail and change state to **PreparationFailed**. This is caused by the update resource provider (URP) being unable to properly transfer the files from the storage container to an internal infrastructure share for processing.
- Remediation: Starting with version 1901 (1.1901.0.95), you can work around this issue by clicking **Update now** again (not **Resume**). The URP then cleans up the files from the previous attempt, and restarts the download. If the problem persists, we recommend manually uploading the update package by following the [Install updates section](#).
- Occurrence: Common

Portal

Administrative subscriptions

- Applicable: This issue applies to all supported releases.
- Cause: The two administrative subscriptions that were introduced with version 1804 should not be used. The subscription types are **Metering** subscription, and **Consumption** subscription.
- Remediation: If you have resources running on these two subscriptions, recreate them in user subscriptions.
- Occurrence: Common

Subscriptions Properties blade

- Applicable: This issue applies to all supported releases.
- Cause: In the administrator portal, the **Properties** blade for subscriptions does not load correctly
- Remediation: You can view these subscription properties in the **Essentials** pane of the **Subscriptions Overview** blade.
- Occurrence: Common

Subscriptions Lock blade

- Applicable: This issue applies to all supported releases.
- Cause: In the administrator portal, the **Lock** blade for user subscriptions has two buttons labeled **subscription**.
- Occurrence: Common

Subscription permissions

- Applicable: This issue applies to all supported releases.
- Cause: You cannot view permissions to your subscription using the Azure Stack Hub portals.
- Remediation: Use [PowerShell to verify permissions](#).
- Occurrence: Common

Storage account settings

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, the storage account **Configuration** blade shows an option to change **security transfer type**. The feature is currently not supported in Azure Stack Hub.
- Occurrence: Common

Upload blob

- Applicable: This issue applies to all supported releases.

- Cause: In the user portal, when you try to upload a blob using the **OAuth(preview)** option, the task fails with an error message.
- Remediation: Upload the blob using the SAS option.
- Occurrence: Common

Alert for network interface disconnected

- Applicable: This issue applies to the 1908 release.
- Cause: When a cable is disconnected from a network adapter, an alert does not show in the administrator portal. This issue is caused because this fault is disabled by default in Windows Server 2019.
- Occurrence: Common

Networking

Load Balancer

- Applicable: This issue applies to all supported releases.
- Cause: When adding Availability Set VMs to the backend pool of a Load Balancer, an error message is being displayed on the portal stating **Failed to save load balancer backend pool**. This is a cosmetic issue on the portal, the functionality is still in place and VMs are successfully added to the backend pool internally.
- Occurrence: Common

Network Security Groups

- Applicable: This issue applies to all supported releases.
- Cause: An explicit **DenyAllOutbound** rule cannot be created in an NSG as this will prevent all internal communication to infrastructure needed for the VM deployment to complete.
- Occurrence: Common

Service endpoints

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, the **Virtual Network** blade shows an option to use **Service Endpoints**. This feature is currently not supported in Azure Stack Hub.
- Occurrence: Common

Network interface

Adding/Removing Network Interface

- Applicable: This issue applies to all supported releases.
- Cause: A new network interface cannot be added to a VM that is in a **running** state.
- Remediation: Stop the virtual machine before adding/removing a network interface.
- Occurrence: Common

Primary Network Interface

- Applicable: This issue applies to all supported releases.
- Cause: A new network interface cannot be added to a VM that is in a **running** state.
- Remediation: Stop the virtual machine before adding/removing a network interface.
- Occurrence: Common

Virtual Network Gateway

Alerts

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, the **Virtual Network Gateway** blade shows an option to use **Alerts**. This feature is currently not supported in Azure Stack Hub.
- Occurrence: Common

Active-Active

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, while creating, and in the resource menu of **Virtual Network Gateway**, you will see an option to enable **Active-Active** configuration. This feature is currently not supported in Azure Stack Hub.
- Occurrence: Common

VPN troubleshooter

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, the **Connections** blade shows a feature called **VPN Troubleshooter**. This feature is currently not supported in Azure Stack Hub.
- Occurrence: Common

Documentation

- Applicable: This issue applies to all supported releases.
- Cause: The documentation links in the overview page of Virtual Network gateway link to Azure-specific documentation instead of Azure Stack Hub. Use the following links for the Azure Stack Hub documentation:
 - [Gateway SKUs](#)
 - [Highly Available Connections](#)
 - [Configure BGP on Azure Stack Hub](#)
 - [ExpressRoute circuits](#)
 - [Specify custom IPsec/IKE policies](#)

Compute

VM boot diagnostics

- Applicable: This issue applies to all supported releases.
- Cause: When creating a new Windows virtual machine (VM), the following error may be displayed: **Failed to start virtual machine 'vm-name'. Error: Failed to update serial output settings for VM 'vm-name'**.
The error occurs if you enable boot diagnostics on a VM, but delete your boot diagnostics storage account.
- Remediation: Recreate the storage account with the same name you used previously.
- Occurrence: Common

Virtual machine scale set

Create failures during patch and update on 4-node Azure Stack Hub environments

- Applicable: This issue applies to all supported releases.
- Cause: Creating VMs in an availability set of 3 fault domains and creating a virtual machine scale set instance fails with a **FabricVmPlacementErrorUnsupportedFaultDomainSize** error during the update process on a 4-node Azure Stack Hub environment.
- Remediation: You can create single VMs in an availability set with 2 fault domains successfully. However, scale set instance creation is still not available during the update process on a 4-node Azure Stack Hub.

Ubuntu SSH access

- Applicable: This issue applies to all supported releases.
- Cause: An Ubuntu 18.04 VM created with SSH authorization enabled does not allow you to use the SSH keys to sign in.
- Remediation: Use VM access for the Linux extension to implement SSH keys after provisioning, or use password-based authentication.
- Occurrence: Common

Virtual machine scale set reset password does not work

- Applicable: This issue applies to all supported releases.
- Cause: A new reset password blade appears in the scale set UI, but Azure Stack Hub does not support resetting password on a scale set yet.
- Remediation: None.
- Occurrence: Common

Rainy cloud on scale set diagnostics

- Applicable: This issue applies to all supported releases.
- Cause: The virtual machine scale set overview page shows an empty chart. Clicking on the empty chart opens a "rainy cloud" blade. This is the chart for scale set diagnostic information, such as CPU percentage, and is not a feature supported in the current Azure Stack Hub build.
- Remediation: None.
- Occurrence: Common

Virtual machine diagnostic settings blade

- Applicable: This issue applies to all supported releases.
- Cause: The virtual machine diagnostic settings blade has a **Sink** tab, which asks for an **Application Insight Account**. This is the result of a new blade and is not yet supported in Azure Stack Hub.
- Remediation: None.
- Occurrence: Common

1907 update process

- Applicable: This issue applies to all supported releases.
- Cause: When attempting to install the 1907 Azure Stack Hub update, the status for the update might fail and change state to **PreparationFailed**. This is caused by the update resource provider (URP) being unable to properly transfer the files from the storage container to an internal infrastructure share for processing.
- Remediation: Starting with version 1901 (1.1901.0.95), you can work around this issue by clicking **Update now** again (not **Resume**). The URP then cleans up the files from the previous attempt, and restarts the download. If the problem persists, we recommend manually uploading the update package by following the [Import and install updates section](#).
- Occurrence: Common

Portal

Administrative subscriptions

- Applicable: This issue applies to all supported releases.
- Cause: The two administrative subscriptions that were introduced with version 1804 should not be used. The subscription types are **Metering** subscription, and **Consumption** subscription.
- Remediation: If you have resources running on these two subscriptions, recreate them in user subscriptions.
- Occurrence: Common

Subscriptions Properties blade

- Applicable: This issue applies to all supported releases.
- Cause: In the administrator portal, the **Properties** blade for subscriptions does not load correctly
- Remediation: You can view these subscription properties in the **Essentials** pane of the **Subscriptions Overview** blade.
- Occurrence: Common

Subscription permissions

- Applicable: This issue applies to all supported releases.

- Cause: You cannot view permissions to your subscription using the Azure Stack Hub portals.
- Remediation: Use [PowerShell to verify permissions](#).
- Occurrence: Common

Storage account settings

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, the storage account **Configuration** blade shows an option to change **security transfer type**. The feature is currently not supported in Azure Stack Hub.
- Occurrence: Common

Upload blob

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, when you try to upload a blob using the **OAuth(preview)** option, the task fails with an error message.
- Remediation: Upload the blob using the SAS option.
- Occurrence: Common

Networking

Load Balancer

- Applicable: This issue applies to all supported releases.
- Cause: When adding Availability Set VMs to the backend pool of a Load Balancer, an error message is being displayed on the portal stating **Failed to save load balancer backend pool**. This is a cosmetic issue on the portal, the functionality is still in place and VMs are successfully added to the backend pool internally.
- Occurrence: Common

Network Security Groups

- Applicable: This issue applies to all supported releases.
- Cause: An explicit **DenyAllOutbound** rule cannot be created in an NSG as this will prevent all internal communication to infrastructure needed for the VM deployment to complete.
- Occurrence: Common

Service endpoints

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, the **Virtual Network** blade shows an option to use **Service Endpoints**. This feature is currently not supported in Azure Stack Hub.
- Occurrence: Common

Network interface

Adding/Removing Network Interface

- Applicable: This issue applies to all supported releases.
- Cause: A new network interface cannot be added to a VM that is in a **running** state.
- Remediation: Stop the virtual machine before adding/removing a network interface.
- Occurrence: Common

Primary Network Interface

- Applicable: This issue applies to all supported releases.
- Cause: A new network interface cannot be added to a VM that is in a **running** state.
- Remediation: Stop the virtual machine before adding/removing a network interface.
- Occurrence: Common

Virtual Network Gateway

Alerts

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, the **Virtual Network Gateway** blade shows an option to use **Alerts**. This feature is currently not supported in Azure Stack Hub.
- Occurrence: Common

Active-Active

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, while creating, and in the resource menu of **Virtual Network Gateway**, you will see an option to enable **Active-Active** configuration. This feature is currently not supported in Azure Stack Hub.
- Occurrence: Common

VPN troubleshooter

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, the **Connections** blade shows a feature called **VPN Troubleshooter**. This feature is currently not supported in Azure Stack Hub.
- Occurrence: Common

Network Connection Type

- Applicable: This issue applies to any 1906 or 1907 environment.
- Cause: In the user portal, the **AddConnection** blade shows an option to use **VNet-to-VNet**. This feature is currently not supported in Azure Stack Hub.
- Occurrence: Common

Documentation

- Applicable: This issue applies to all supported releases.
- Cause: The documentation links in the overview page of Virtual Network gateway link to Azure-specific documentation instead of Azure Stack Hub. Use the following links for the Azure Stack Hub documentation:
 - [Gateway SKUs](#)
 - [Highly Available Connections](#)
 - [Configure BGP on Azure Stack Hub](#)
 - [ExpressRoute circuits](#)
 - [Specify custom IPsec/IKE policies](#)

Compute

VM boot diagnostics

- Applicable: This issue applies to all supported releases.
- Cause: When creating a new Windows virtual machine (VM), the following error may be displayed: **Failed to start virtual machine 'vm-name'. Error: Failed to update serial output settings for VM 'vm-name'**. The error occurs if you enable boot diagnostics on a VM, but delete your boot diagnostics storage account.
- Remediation: Recreate the storage account with the same name you used previously.
- Occurrence: Common

Virtual machine scale set

Create failures during patch and update on 4-node Azure Stack Hub environments

- Applicable: This issue applies to all supported releases.
- Cause: Creating VMs in an availability set of 3 fault domains and creating a virtual machine scale set instance fails with a **FabricVmPlacementErrorUnsupportedFaultDomainSize** error during the update process on a 4-node Azure Stack Hub environment.

- Remediation: You can create single VMs in an availability set with 2 fault domains successfully. However, scale set instance creation is still not available during the update process on a 4-node Azure Stack Hub.

Ubuntu SSH access

- Applicable: This issue applies to all supported releases.
- Cause: An Ubuntu 18.04 VM created with SSH authorization enabled does not allow you to use the SSH keys to sign in.
- Remediation: Use VM access for the Linux extension to implement SSH keys after provisioning, or use password-based authentication.
- Occurrence: Common

Virtual machine scale set reset password does not work

- Applicable: This issue applies to the 1906 and 1907 releases.
- Cause: A new reset password blade appears in the scale set UI, but Azure Stack Hub does not support resetting password on a scale set yet.
- Remediation: None.
- Occurrence: Common

Rainy cloud on scale set diagnostics

- Applicable: This issue applies to the 1906 and 1907 releases.
- Cause: The virtual machine scale set overview page shows an empty chart. Clicking on the empty chart opens a "rainy cloud" blade. This is the chart for scale set diagnostic information, such as CPU percentage, and is not a feature supported in the current Azure Stack Hub build.
- Remediation: None.
- Occurrence: Common

Virtual machine diagnostic settings blade

- Applicable: This issue applies to the 1906 and 1907 releases.
- Cause: The virtual machine diagnostic settings blade has a **Sink** tab, which asks for an **Application Insight Account**. This is the result of a new blade and is not yet supported in Azure Stack Hub.
- Remediation: None.
- Occurrence: Common

1906 update process

- Applicable: This issue applies to all supported releases.
- Cause: When attempting to install the 1906 Azure Stack Hub update, the status for the update might fail and change state to **PreparationFailed**. This is caused by the update resource provider (URP) being unable to properly transfer the files from the storage container to an internal infrastructure share for processing.
- Remediation: Starting with version 1901 (1.1901.0.95), you can work around this issue by clicking **Update now** again (not **Resume**). The URP then cleans up the files from the previous attempt, and restarts the download. If the problem persists, we recommend manually uploading the update package by following the [Import and install updates section](#).
- Occurrence: Common

Portal

Administrative subscriptions

- Applicable: This issue applies to all supported releases.
- Cause: The two administrative subscriptions that were introduced with version 1804 should not be used. The subscription types are **Metering** subscription, and **Consumption** subscription.

- Remediation: If you have resources running on these two subscriptions, recreate them in user subscriptions.
- Occurrence: Common

Subscription resources

- Applicable: This issue applies to all supported releases.
- Cause: Deleting user subscriptions results in orphaned resources.
- Remediation: First delete user resources or the entire resource group, and then delete the user subscriptions.
- Occurrence: Common

Subscription permissions

- Applicable: This issue applies to all supported releases.
- Cause: You cannot view permissions to your subscription using the Azure Stack Hub portals.
- Remediation: Use [PowerShell to verify permissions](#).
- Occurrence: Common

Subscriptions Properties blade

- Applicable: This issue applies to all supported releases.
- Cause: In the administrator portal, the **Properties** blade for Subscriptions does not load correctly
- Remediation: You can view these subscriptions properties in the Essentials pane of the Subscriptions Overview blade
- Occurrence: Common

Storage account settings

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, the storage account **Configuration** blade shows an option to change **security transfer type**. The feature is currently not supported in Azure Stack Hub.
- Occurrence: Common

Upload blob

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, when you try to upload a blob using the **OAuth(preview)** option, the task fails with an error message.
- Remediation: Upload the blob using the SAS option.
- Occurrence: Common

Update

- Applicable: This issue applies to the 1906 release.
- Cause: In the operator portal, update status for the hotfix shows an incorrect state for the update. Initial state indicates that the update failed to install, even though it is still in progress.
- Remediation: Refresh the portal and the state will update to "in progress."
- Occurrence: Intermittent

Networking

Service endpoints

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, the **Virtual Network** blade shows an option to use **Service Endpoints**. This feature is currently not supported in Azure Stack Hub.
- Occurrence: Common

Network interface

- Applicable: This issue applies to all supported releases.
- Cause: A new network interface cannot be added to a VM that is in a **running** state.
- Remediation: Stop the virtual machine before adding/removing a network interface.
- Occurrence: Common

Virtual Network Gateway

Alerts

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, the **Virtual Network Gateway** blade shows an option to use **Alerts**. This feature is currently not supported in Azure Stack Hub.
- Occurrence: Common

Active-Active

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, while creating, and in the resource menu of **Virtual Network Gateway**, you will see an option to enable **Active-Active** configuration. This feature is currently not supported in Azure Stack Hub.
- Occurrence: Common

VPN troubleshooter

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, the **Connections** blade shows a feature called **VPN Troubleshooter**. This feature is currently not supported in Azure Stack Hub.
- Occurrence: Common

Documentation

- Applicable: This issue applies to all supported releases.
- Cause: The documentation links in the overview page of Virtual Network gateway link to Azure-specific documentation instead of Azure Stack Hub. Please use the following links for the Azure Stack Hub documentation:
 - [Gateway SKUs](#)
 - [Highly Available Connections](#)
 - [Configure BGP on Azure Stack Hub](#)
 - [ExpressRoute circuits](#)
 - [Specify custom IPsec/IKE policies](#)

Load balancer

Add backend pool

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, if you attempt to add a **Backend Pool** to a **Load Balancer**, the operation fails with the error message **failed to update Load Balancer....**
- Remediation: Use PowerShell, CLI or a Resource Manager template to associate the backend pool with a load balancer resource.
- Occurrence: Common

Create inbound NAT

- Applicable: This issue applies to all supported releases.
- Cause: In the user portal, if you attempt to create an **Inbound NAT Rule** for a **Load Balancer**, the operation fails with the error message **Failed to update Load Balancer....**
- Remediation: Use PowerShell, CLI or a Resource Manager template to associate the backend pool with a load balancer resource.
- Occurrence: Common

Compute

VM boot diagnostics

- Applicable: This issue applies to all supported releases.
- Cause: When creating a new Windows virtual machine (VM), the following error may be displayed: **Failed to start virtual machine 'vm-name'. Error: Failed to update serial output settings for VM 'vm-name'**. The error occurs if you enable boot diagnostics on a VM, but delete your boot diagnostics storage account.
- Remediation: Recreate the storage account with the same name you used previously.
- Occurrence: Common

Virtual machine scale set

Create failures during patch and update on 4-node Azure Stack Hub environments

- Applicable: This issue applies to all supported releases.
- Cause: Creating VMs in an availability set of 3 fault domains and creating a virtual machine scale set instance fails with a **FabricVmPlacementErrorUnsupportedFaultDomainSize** error during the update process on a 4-node Azure Stack Hub environment.
- Remediation: You can create single VMs in an availability set with 2 fault domains successfully. However, scale set instance creation is still not available during the update process on a 4-node Azure Stack Hub.

Ubuntu SSH access

- Applicable: This issue applies to all supported releases.
- Cause: An Ubuntu 18.04 VM created with SSH authorization enabled does not allow you to use the SSH keys to sign in.
- Remediation: Use VM access for the Linux extension to implement SSH keys after provisioning, or use password-based authentication.
- Occurrence: Common

Virtual machine scale set reset password does not work

- Applicable: This issue applies to the 1906 release.
- Cause: A new reset password blade appears in the scale set UI, but Azure Stack Hub does not support resetting password on a scale set yet.
- Remediation: None.
- Occurrence: Common

Rainy cloud on scale set diagnostics

- Applicable: This issue applies to the 1906 release.
- Cause: The virtual machine scale set overview page shows an empty chart. Clicking on the empty chart opens a "rainy cloud" blade. This is the chart for scale set diagnostic information, such as CPU percentage, and is not a feature supported in the current Azure Stack Hub build.
- Remediation: None.
- Occurrence: Common

Issues creating resources

- Applicable: This issue applies to the 1906 release.
- Cause: There is a known issue in 1906 with custom roles and permission allocation for resource creation. You might face issues creating resources even if you have the correct permissions.
- Remediation: Please update to build 1907 to mitigate this issue.
- Occurrence: Common

Virtual machine diagnostic settings blade

- Applicable: This issue applies to the 1906 release.

- Cause: The virtual machine diagnostic settings blade has a **Sink** tab, which asks for an **Application Insight Account**. This is the result of a new blade and is not yet supported in Azure Stack Hub.
- Remediation: None.
- Occurrence: Common

Archive

To access archived known issues for an older version, use the version selector dropdown above the table of contents on the left, and select the version you want to see.

Next steps

- [Review update activity checklist](#)
- [Review list of security updates](#)

[1905 archived known issues](#)

[1904 archived known issues](#)

[1903 archived known issues](#)

[1902 archived known issues](#)

[1901 archived known issues](#)

[1811 archived known issues](#)

[1809 archived known issues](#)

[1808 archived known issues](#)

[1807 archived known issues](#)

[1805 archived known issues](#)

[1804 archived known issues](#)

[1803 archived known issues](#)

[1802 archived known issues](#)

You can access [older versions of Azure Stack Hub known issues on the TechNet Gallery](#). These archived documents are provided for reference purposes only and do not imply support for these versions. For information about Azure Stack Hub support, see [Azure Stack Hub servicing policy](#). For further assistance, contact Microsoft Customer Support Services.

Use the administrator portal in Azure Stack Hub

4 minutes to read • [Edit Online](#)

There are two portals in Azure Stack Hub: the administrator portal and the user portal. As an Azure Stack Hub operator, you use the administrator portal for day-to-day management and operations of Azure Stack Hub.

Access the administrator portal

To access the administrator portal, browse to the portal URL and sign in by using your Azure Stack Hub operator credentials. For an integrated system, the portal URL varies based on the region name and external fully qualified domain name (FQDN) of your Azure Stack Hub deployment. The administrator portal URL is always the same for Azure Stack Development Kit (ASDK) deployments.

ENVIRONMENT	ADMINISTRATOR PORTAL URL
ASDK	https://adminportal.local.azurestack.external
Integrated systems	<a href="https://adminportal.<region>.<FQDN>">https://adminportal.<region>.<FQDN>

TIP

For an ASDK environment, you need to first make sure that you can [connect to the development kit host](#) through Remote Desktop Connection or through a virtual private network (VPN).

The screenshot shows the Azure Stack Hub Administration portal. The left sidebar contains navigation links like 'Create a resource', 'All services', 'Dashboard', 'All resources', 'Resource groups', 'Virtual machines', 'Recent', 'Plans', 'Offers', 'Marketplace management', and 'Monitor'. The main dashboard has sections for 'Region management' (showing 1 globe icon), 'Resource providers' (listing Capacity, Compute, Infrastructure backup, Key Vault, Network, Storage all in healthy status), and 'Alerts' (0 Critical, 0 Warning). On the right, there are 'Quickstart tutorials' with links to 'Create a virtual machine', 'Offering services', 'Populate the Azure Stack marketplace', 'Manage infrastructure', and 'Use the Azure Stack portal'.

The default time zone for all Azure Stack Hub deployments is set to Coordinated Universal Time (UTC).

In the administrator portal, you can do things like:

- [Register Azure Stack Hub with Azure](#)
- [Populate the marketplace](#)

- Create plans, offers, and subscriptions for users
- Monitor health and alerts
- Manage Azure Stack Hub updates

The **Quickstart tutorial** tile provides links to online documentation for the most common tasks.

Although an operator can create resources such as virtual machines (VMs), virtual networks, and storage accounts in the administrator portal, you should [sign in to the user portal](#) to create and test resources.

NOTE

The **Create a virtual machine** link in the quickstart tutorial tile has you create a VM in the administrator portal, but this is only intended to validate that Azure Stack Hub has been deployed successfully.

Understand subscription behavior

There are three subscriptions created by default in the administrator portal: consumption, default provider, and metering. As an operator, you'll mostly use the *Default Provider Subscription*. You can't add any other subscriptions and use them in the administrator portal.

Other subscriptions are created by users in the user portal based on the plans and offers you create for them. However, the user portal doesn't provide access to any of the administrative or operational capabilities of the administrator portal.

The administrator and user portals are backed by separate instances of Azure Resource Manager. Because of this Azure Resource Manager separation, subscriptions don't cross portals. For example, if you, as an Azure Stack Hub operator, sign in to the user portal, you can't access the *Default Provider Subscription*. Although you don't have access to any administrative functions, you can create subscriptions for yourself from available public offers. As long as you're signed in to the user portal, you're considered a tenant user.

NOTE

In an ASDK environment, if a user belongs to the same tenant directory as the Azure Stack Hub operator, they're not blocked from signing in to the administrator portal. However, they can't access any of the administrative functions or add subscriptions to access offers that are available to them in the user portal.

Administrator portal tips

Customize the dashboard

The dashboard contains a set of default tiles. You can select **Edit dashboard** to modify the default dashboard, or select **New dashboard** to add a custom dashboard. You can also add tiles to a dashboard. For example, select **+ Create a resource**, right-click **Offers + Plans**, and then select **Pin to dashboard**.

Sometimes, you might see a blank dashboard in the portal. To recover the dashboard, click **Edit Dashboard**, and then right-click and select **Reset to default state**.

Quick access to online documentation

To access the Azure Stack Hub operator documentation, use the help and support icon (question mark) in the upper-right corner of the administrator portal. Move your cursor to the icon, and then select **Help + support**.

Quick access to help and support

If you click the help icon (question mark) in the upper-right corner of the administrator portal, click **Help + support**, and then click **New support request** under **Support**, one of the following results happens:

- If you're using an integrated system, this action opens a site where you can directly open a support ticket with Microsoft Customer Support Services (CSS). Refer to [Where to get support](#) to understand when you should go through Microsoft support or through your original equipment manufacturer (OEM) hardware vendor support.
- If you're using the ASDK, this action opens the [Azure Stack Hub forums site](#) directly. These forums are regularly monitored. Because the ASDK is an evaluation environment, there's no official support offered through Microsoft CSS.

Quick access to the Azure roadmap

If you select **Help and support** (the question mark) in the upper right corner of the administrator portal, and then select **Azure roadmap**, a new browser tab opens and takes you to the Azure roadmap. By typing **Azure Stack Hub** in the **Products** search box, you can see all Azure Stack Hub roadmap updates.

Next steps

Register Azure Stack Hub with Azure and populate [Azure Stack Hub Marketplace](#) with items to offer your users.

Create a service offering for users in Azure Stack Hub

6 minutes to read • [Edit Online](#)

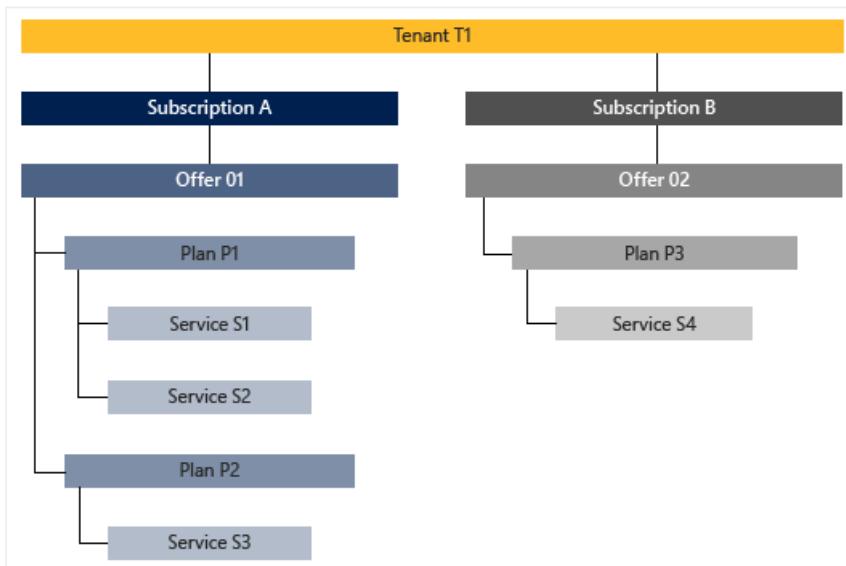
This tutorial shows an operator how to create an offer. An offer makes services available to users on a subscription basis. Once subscribed to an offer, a user is entitled to create and deploy resources within the services specified by the offer.

In this tutorial, you learn how to:

- Create an offer.
- Create a plan.
- Assign services and quotas to a plan.
- Assign a plan to an offer.

Overview

An offer consists of one or more plans. A plan entitles access to one or more services, by specifying each service's corresponding resource provider and a quota. Plans can be added to an offer as the base plan, or they can extend the offer as an add-on plan. To learn more, see the [Service, plan, offer, subscription overview](#).



Resource providers

A resource provider supports creation, deployment, and management of its resources as services. A common example is the Microsoft.Compute resource provider, which offers the ability to create and deploy virtual machines (VMs). See [Azure Resource Manager](#) for an overview of the Azure resource management model.

In Azure Stack Hub, there are two general categories of resource providers: ones that deploy resources as foundational services, and ones that deploy as value-add services.

Foundational services

NOTE

In this tutorial, you learn how to create an offer based on foundational services.

Foundational services are supported by the following resource providers, which are available natively with every

installation of Azure Stack Hub:

RESOURCE PROVIDER	EXAMPLE RESOURCES
Microsoft.Compute	VMs, disks, virtual machine scale sets
Microsoft.KeyVault	Key Vaults, secrets
Microsoft.Network	Virtual networks, public IP addresses, load balancers
Microsoft.Storage	Storage accounts, blobs, queues, tables

Value-add services

NOTE

In order to offer a value-add service, the corresponding resource provider must first be installed in Azure Stack Hub Marketplace. Once installed, its resources are offered to users in the same way as foundational services. Please see the **How-to guides** section of the TOC for the current set of resource providers that support value-add service offerings.

Value-add services are supported by resource providers that are installed after Azure Stack Hub has been deployed. Examples include:

RESOURCE PROVIDER	EXAMPLE RESOURCES
Microsoft.Web	App Service function apps, web apps, API apps
Microsoft.MySqlAdapter	MySQL hosting server, MySQL database
Microsoft.SqlAdapter	SQL Server hosting server, SQL Server database

Create an offer

During the offer creation process, you create both an offer and a plan. The plan is used as the offer's base plan. During plan creation, you specify the services made available in the plan and their respective quotas.

1. Sign in to the administrator portal with a cloud admin account.

- For an integrated system, the URL varies based on your operator's region and external domain name. The URL uses the format `https://adminportal.<region>.<FQDN>`.
- If you're using the Azure Stack Development Kit, the URL is `https://adminportal.local.azurestack.external`.

Then select + **Create a resource** > **Offers + Plans** > **Offer**.

The screenshot shows the Microsoft Azure Stack - Administration interface. On the left, there's a sidebar with various navigation options like Dashboard, All resources, Resource groups, etc. The main area is titled 'New' and features a search bar at the top. Below it, there's a section for 'Azure Marketplace' with a 'Featured' tab selected. A red box highlights the 'Offers + Plans' option under the 'Get started' section. Other options include Compute, Data + Storage, Networking, Custom, Security + Identity, and Capacity.

2. In **Create a new offer** under the **Basics** tab, enter a **Display name**, **Resource name**, and select an existing or create a new **Resource group**. The Display name is the offer's friendly name. Only the cloud operator can see the Resource name, which is the name that admins use to work with the offer as an Azure Resource Manager resource.

The screenshot shows the 'Create a new offer' form. At the top, there are tabs for 'Basics' (which is selected and highlighted with a red box), 'Base plans', 'Add-on plans', and 'Review + create'. The 'Basics' tab has several input fields:

- * Display name: A field with placeholder text 'Enter the display name that users see'.
- * Resource name: A field with placeholder text 'Enter the unique identifier of the offer'.
- Description: A large text area for entering a description.
- * Resource group: A dropdown menu with options 'Select existing...' and 'Create new'.

At the bottom, there's a question 'Make this offer public?' with 'Yes' and 'No' buttons, and navigation buttons for 'Review + create', 'Previous', and 'Next : Base plans >'.

3. Select the **Base plans** tab, then select **Create new plan** to create a new plan. The plan will also be added to the offer as a base plan.

Dashboard > New > Create a new offer

Create a new offer

Create a new offer for your users

Basics Base plans Add-on plans Review + create

Select any plans that should be made available immediately to a user subscribing to this offer

Create new plan

0 items

Search to filter items...

<input type="checkbox"/> DISPLAY NAME	DESCRIPTION
No plans found	

Please select at least one base plan for the offer

Review + create **Previous** **Next : Add-on plans >**

The screenshot shows the 'Create a new offer' interface. The 'Base plans' tab is highlighted with a red box. Below it is a large search bar with placeholder text 'Search to filter items...'. Underneath is a table with columns for 'DISPLAY NAME' and 'DESCRIPTION', which currently shows 'No plans found'. A red message at the bottom left says 'Please select at least one base plan for the offer'. At the bottom are navigation buttons: 'Review + create' (highlighted), 'Previous', and 'Next : Add-on plans >'.

4. In **New plan** under the **Basics** tab, enter a **Display name** and **Resource name**. The Display name is the plan's friendly name that users see. Only the cloud operator can see the Resource name, which is the name that cloud operators use to work with the plan as an Azure Resource Manager resource. **Resource group** will be set to the one specified for the Offer.

Dashboard > New > Create a new offer > New plan

New plan

Create a plan to offer to your users.

Basics Services Quotas Review + create

* Display name
Enter the display name that users see

* Resource name
Enter the unique identifier of the plan

Description

* Resource group
rgOffers
Create new

Review + create **Previous** **Next : Services >**

The screenshot shows the 'New plan' configuration page. The 'Basics' tab is highlighted with a red box. Two input fields are also highlighted with red boxes: 'Display name' (containing 'Enter the display name that users see') and 'Resource name' (containing 'Enter the unique identifier of the plan'). Below these is a large text area for 'Description' and a dropdown menu for 'Resource group' containing 'rgOffers' and a 'Create new' option. Navigation buttons are at the bottom.

5. Select the **Services** tab, and you see a list of services available from the installed resource providers. Select **Microsoft.Compute**, **Microsoft.Network**, and **Microsoft.Storage**.

Dashboard > New > Create a new offer > New plan

New plan

Create a plan to offer to your users.

Basics Services Quotas Review + create

Select one or more services to be offered as part of this plan

5 items

Search to filter items...

SERVICE

Microsoft.Compute

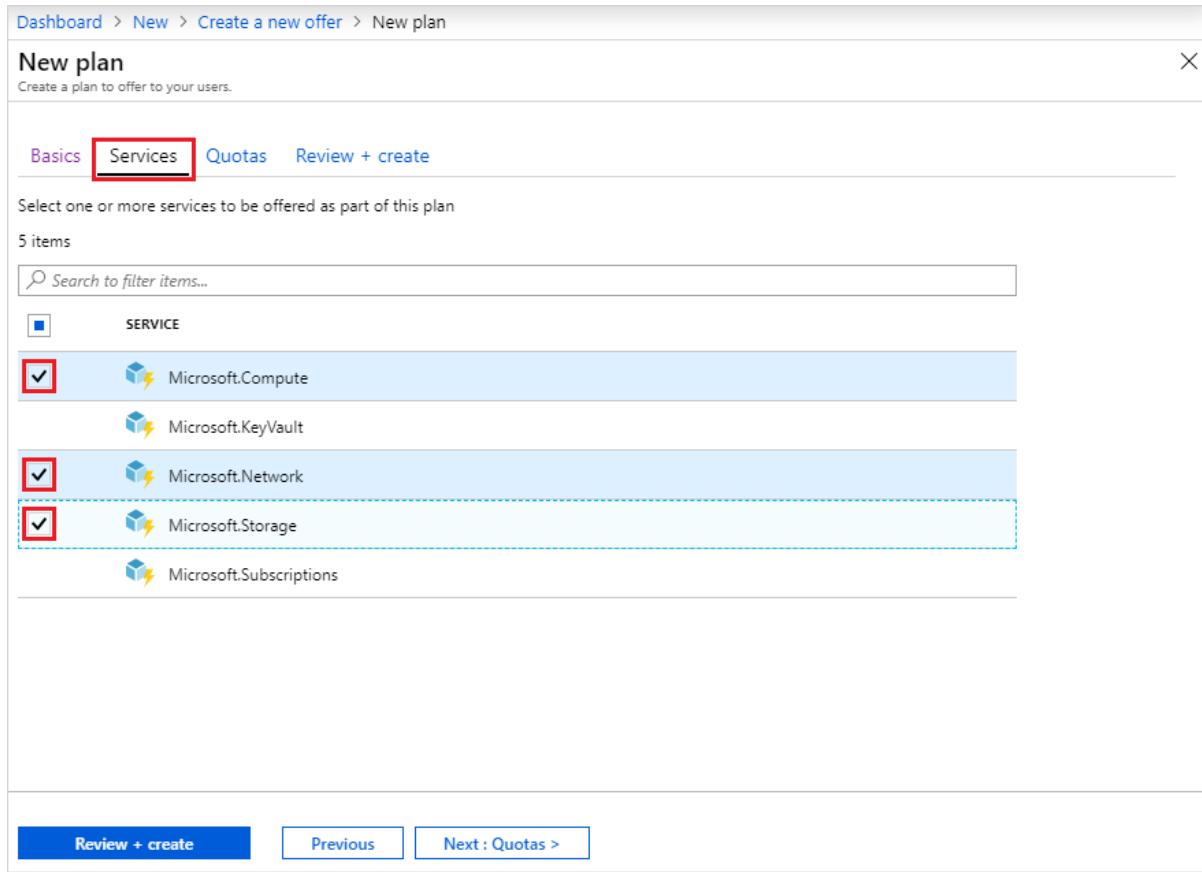
Microsoft.KeyVault

Microsoft.Network

Microsoft.Storage

Microsoft.Subscriptions

Review + create Previous Next : Quotas >



6. Select the **Quotas** tab, and you see the list of services you enabled for this plan. Select **Create New** to specify a custom quota for **Microsoft.Compute**. Quota **Name** is required; you can accept or change each quota value. Select **OK** when finished, then repeat these steps for the remaining services.

Dashboard > New > Create a new offer > New plan

New plan

Create a plan to offer to your users.

Basics Services Quotas Review + create

Select a quota for each of the selected services

3 items

Microsoft.Compute

Microsoft.Network

Microsoft.Storage

Create Compute quota

* Name

Number of virtual machines

Number of virtual machine cores

Number of availability sets

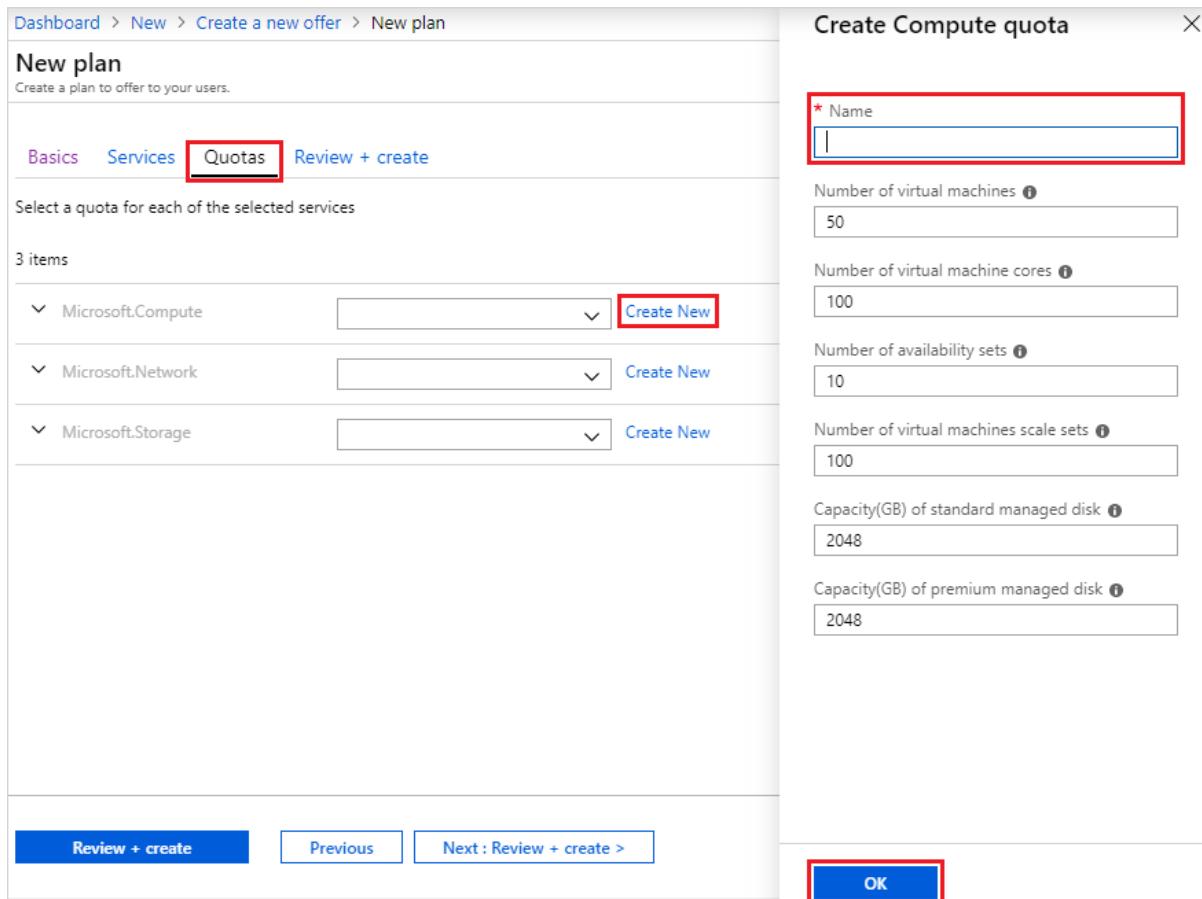
Number of virtual machines scale sets

Capacity(GB) of standard managed disk

Capacity(GB) of premium managed disk

OK

Review + create Previous Next : Review + create >



7. Select the **Review + create** tab. You should see a green "Validation passed" banner at the top, indicating the new base plan is ready to be created. Select **Create**. You should also see a notification indicating that the plan has been created.

Dashboard > New > Create a new offer > New plan

New plan

Create a plan to offer to your users.

✓ Validation passed

Basics Services Quotas **Review + create**

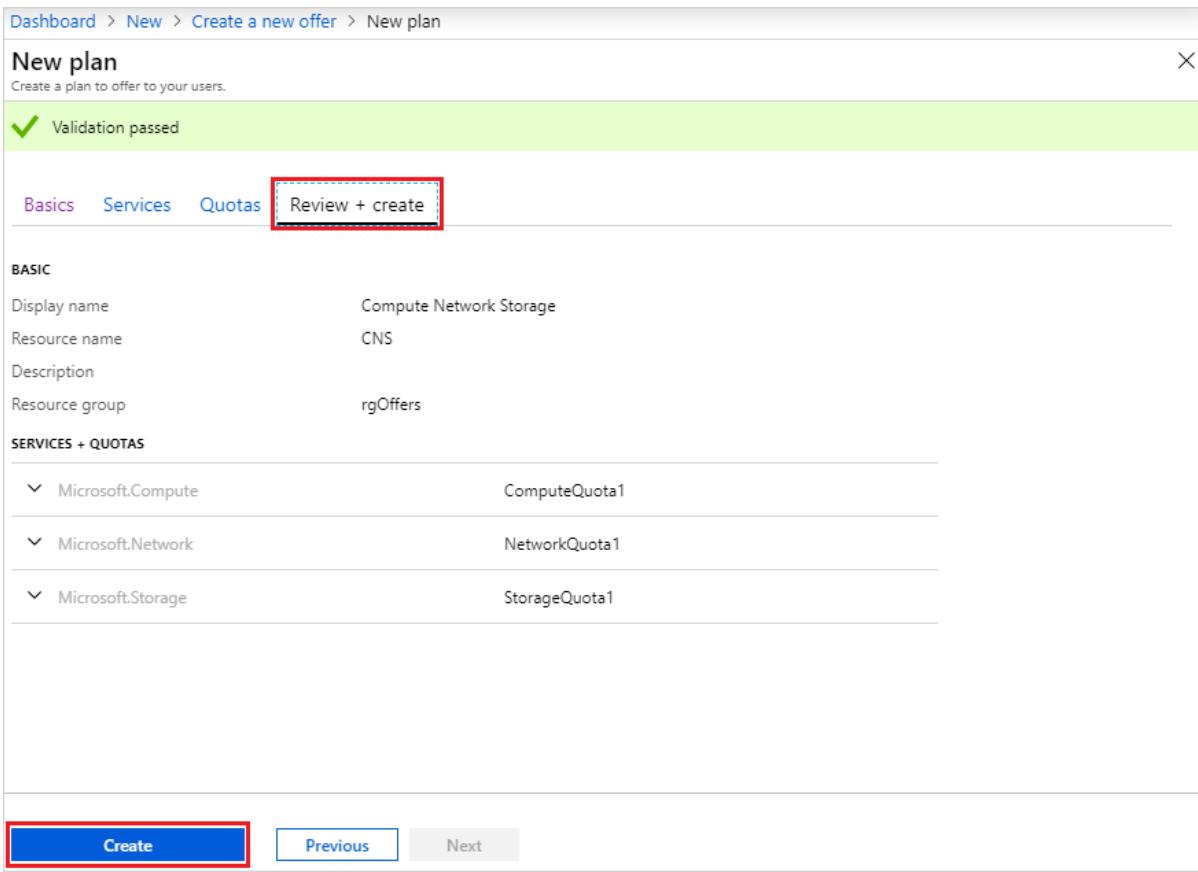
BASIC

Display name	Compute Network Storage
Resource name	CNS
Description	
Resource group	rgOffers

SERVICES + QUOTAS

Microsoft.Compute	ComputeQuota1
Microsoft.Network	NetworkQuota1
Microsoft.Storage	StorageQuota1

Create Previous Next



8. After returning to the **Base plans** tab of the **Create a new offer** page, you notice that the plan has been created. Be sure the new plan is selected for inclusion in the offer as the base plan, then select **Review + create**.

Dashboard > New > Create a new offer

Create a new offer

Create a new offer for your users

Basics **Base plans** Add-on plans Review + create

Select any plans that should be made available immediately to a user subscribing to this offer

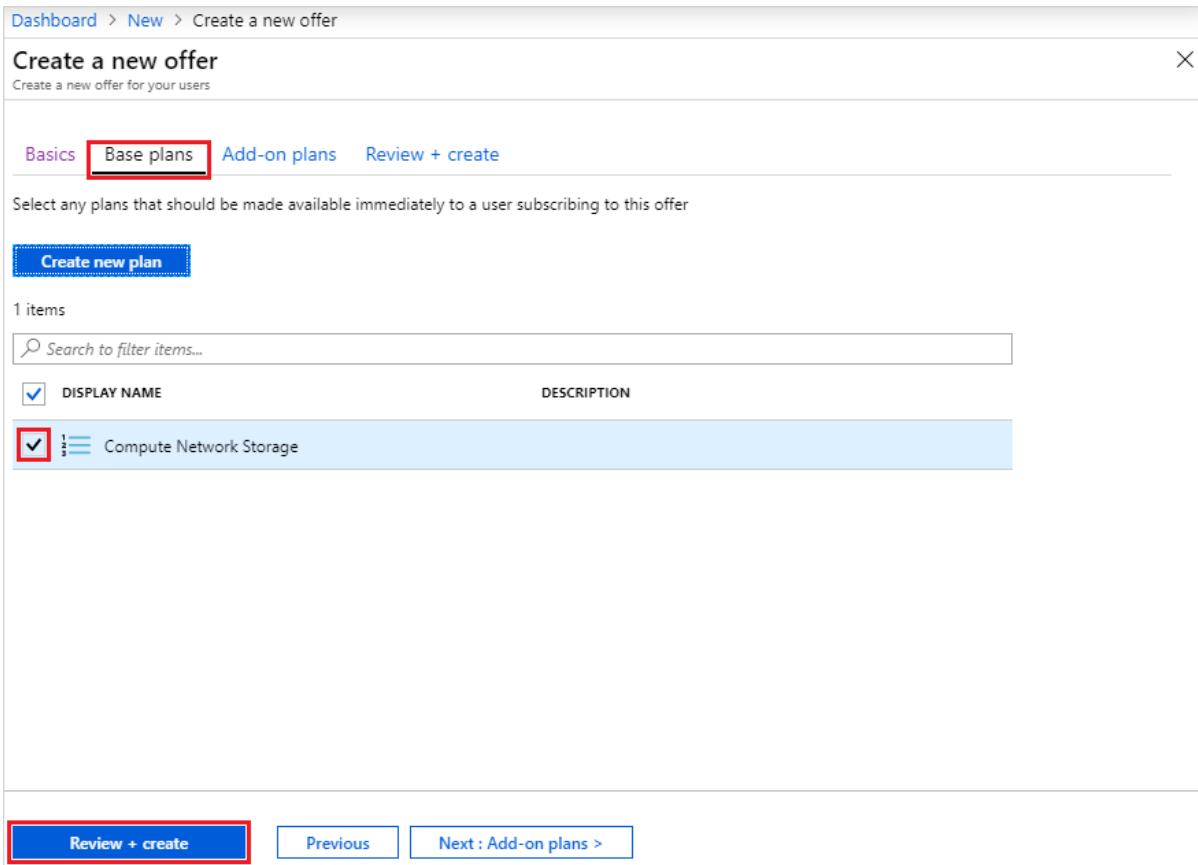
Create new plan

1 items

Search to filter items...

<input checked="" type="checkbox"/>	DISPLAY NAME	DESCRIPTION
<input checked="" type="checkbox"/>	Compute Network Storage	

Review + create Previous Next : Add-on plans >



9. On the **Review + create** tab, you should see a green "Validation passed" banner at the top. Review the "Basic" and "Base Plans" info, and select **Create** when ready.

The screenshot shows the 'Create a new offer' wizard. At the top, it says 'Validation passed'. Below that, there are tabs: 'Basics', 'Base plans', 'Add-on plans', and 'Review + create'. The 'Review + create' tab is selected. Under 'BASIC' settings, 'Display name' is 'Pay as you go', 'Resource name' is 'PAYG', 'Description' is 'rgOffers', and 'Resource group' is 'rgOffers'. Under 'BASE PLANS', 'Compute Network Storage' includes 'Microsoft.Storage' (StorageQuota1), 'Microsoft.Network' (NetworkQuota1), and 'Microsoft.Compute' (ComputeQuota1). Under 'ADD-ON PLANS', there is no content. At the bottom, there are buttons for 'Create' (highlighted with a red box), 'Previous', and 'Next'.

10. The "Your deployment is underway" page shows initially, followed by "Your deployment is complete" once the offer is deployed. Select on the name of the offer under the **Resource** column.

The screenshot shows the 'Microsoft Admin Offer - Overview' page. It displays a success message: 'Deployment succeeded' at 5:12 PM, stating that the deployment to resource group 'rgOffers' was successful. Below this, there's a summary: Deployment name: Microsoft.AdminOffer, Subscription: Default Provider Subscription, Resource group: rgOffers. A 'Go to resource' button is available. On the left, a sidebar lists 'Overview', 'Inputs', 'Outputs', and 'Template'. A table titled 'DEPLOYMENT DETAILS' shows deployment details: Start time: 10/11/2019, 5:12:14 PM; Duration: 26 seconds; Correlation ID: 262b415a-6d66-4cf2-9e46-a5ca24877dbe. A table of resources is shown below:

RESOURCE	TYPE	STATUS	OPERATIO...
PAYG	Microsoft.S...	Created	Operation de...

11. Notice the banner, showing your offer is still private, which prevents users from subscribing to it. Change it to public by selecting **Change State**, and then chose **Public**.

The screenshot shows the Microsoft Admin Offer - Overview page for a PAYG offer. The 'Change state' button is highlighted with a red box. A dropdown menu shows 'This offer' and 'Public'. The main pane displays offer details: Resource group rgOffers, Status Decommissioned, Location local, Subscription Default Provider Subscription, Subscription ID 0101536f-3f00-4321-a850-18a7b8c76ed3, Display name Pay as you go, State Private, Subscriptions 0 subscriptions, Base plans 1 base plans, and Add-on plans 0 add-on plans. Below this is a chart titled 'Subscriptions created over the last week'.

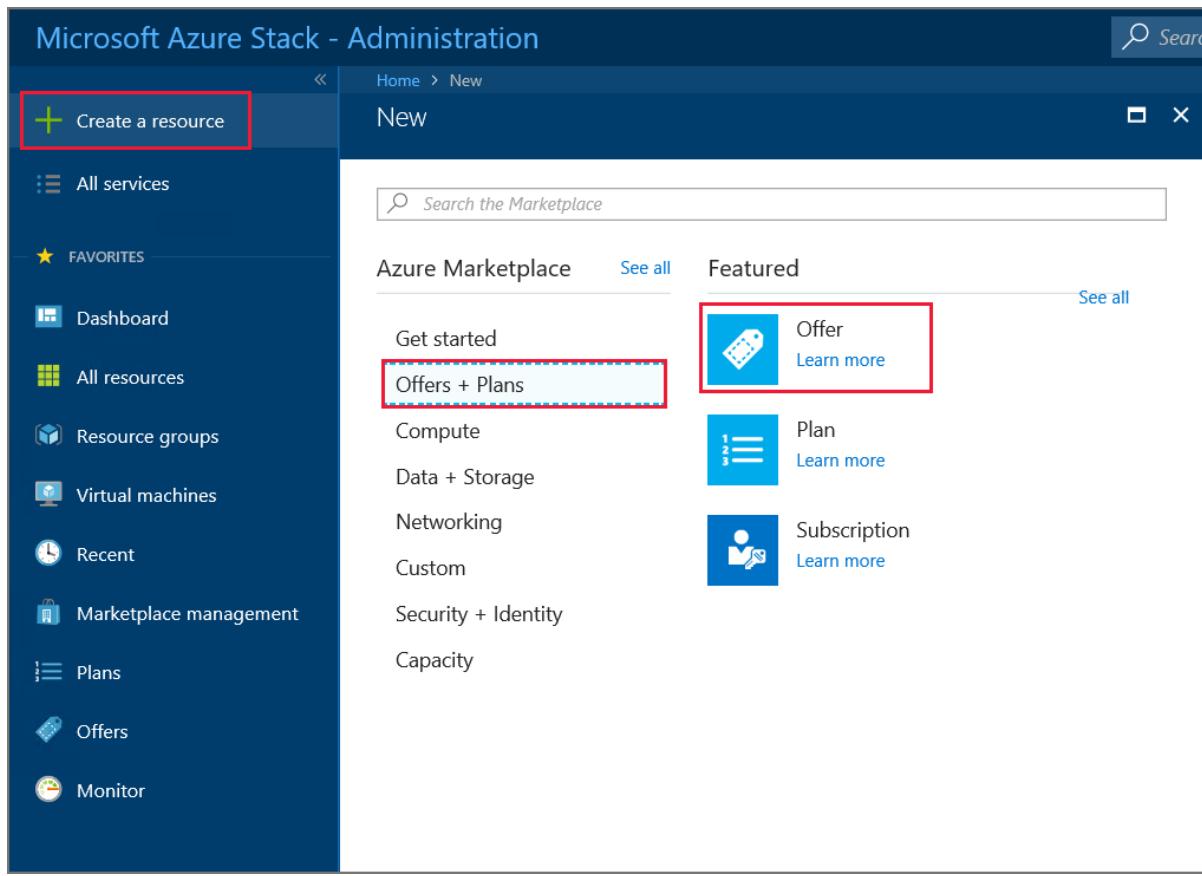
Create an offer (1901 and earlier)

During the offer creation process, you create both an offer and a plan. The plan is used as the offer's base plan. During plan creation, you specify the services made available in the plan and their respective quotas.

1. Sign in to the administrator portal with a cloud admin account.

- For an integrated system, the URL varies based on your operator's region and external domain name, using the format `https://adminportal.<region>.<FQDN>`.
- If you're using the Azure Stack Development Kit, the URL is <https://adminportal.local.azurestack.external>.

Then select + **Create a resource > Offers + Plans > Offer**.



2. In **New offer**, enter a **Display name** and **Resource name**, and then select a new or existing **Resource group**. The Display name is the offer's friendly name. Only the cloud operator can see the Resource name, which is the name that admins use to work with the offer as an Azure Resource Manager resource.

Home > New > New offer

New offer

Create a new offer for your users

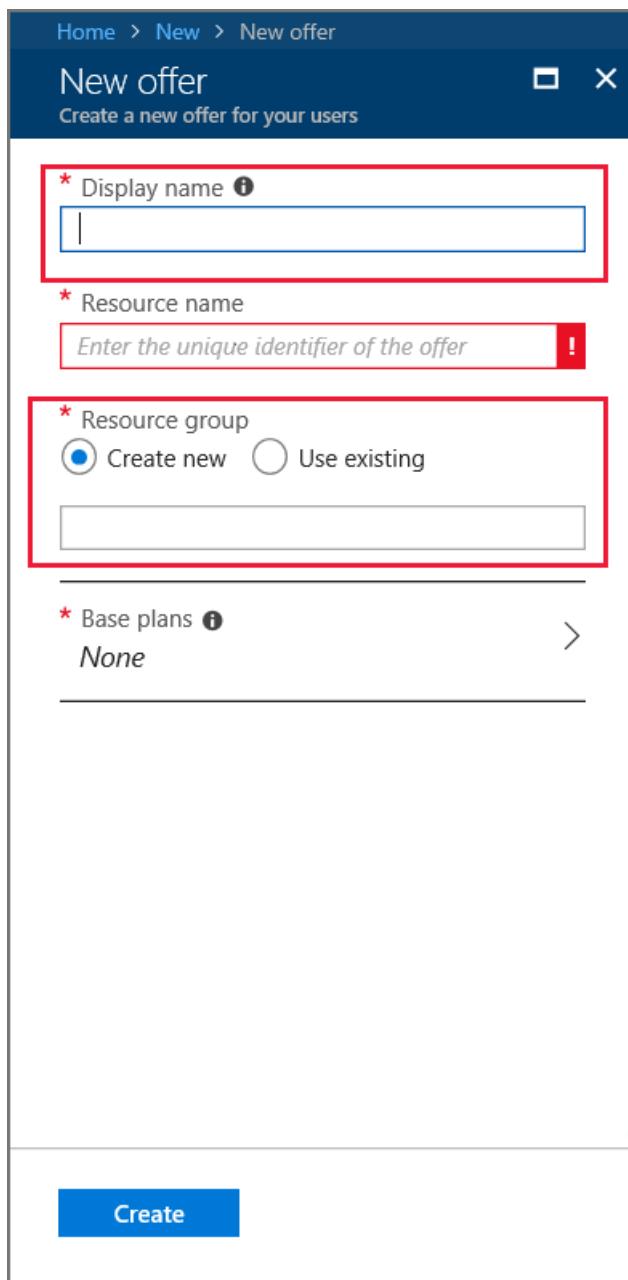
* Display name ⓘ
|

* Resource name
Enter the unique identifier of the offer !

* Resource group
 Create new Use existing
[empty input field]

* Base plans ⓘ >
None

Create



3. Select **Base plans**, and in the **Plan** section, select **Add** to add a new plan to the offer.

The screenshot shows two side-by-side configuration panels. On the left, the 'New offer' panel has a 'Display name' field set to 'UserOffer01' and a 'Resource name' field set to 'useroffer01'. A red box highlights the 'Base plans' section, which currently shows 'None'. On the right, the 'Plan' panel shows a list titled 'DISPLAY NAME' with a single entry 'No plans found'. A red box highlights the '+ Add' button at the top of this list.

4. In the **New plan** section, fill in **Display name** and **Resource name**. The Display name is the plan's friendly name that users see. Only the cloud operator can see the Resource name, which is the name that cloud operators use to work with the plan as an Azure Resource Manager resource.

The 'New plan' configuration interface is shown. It includes fields for 'Display name' (with placeholder 'Enter the display name that users see'), 'Resource name' (with placeholder 'Enter the unique identifier of the plan'), and 'Resource group' (radio buttons for 'Create new' or 'Use existing'). Below these are sections for 'Services' (set to 'Not selected') and 'Quotas' (set to 'None selected'). At the bottom is a large blue 'OK' button.

5. Select **Services**. From the list of Services, pick **Microsoft.Compute**, **Microsoft.Network**, and **Microsoft.Storage**. Choose **Select** to add these services to the plan.

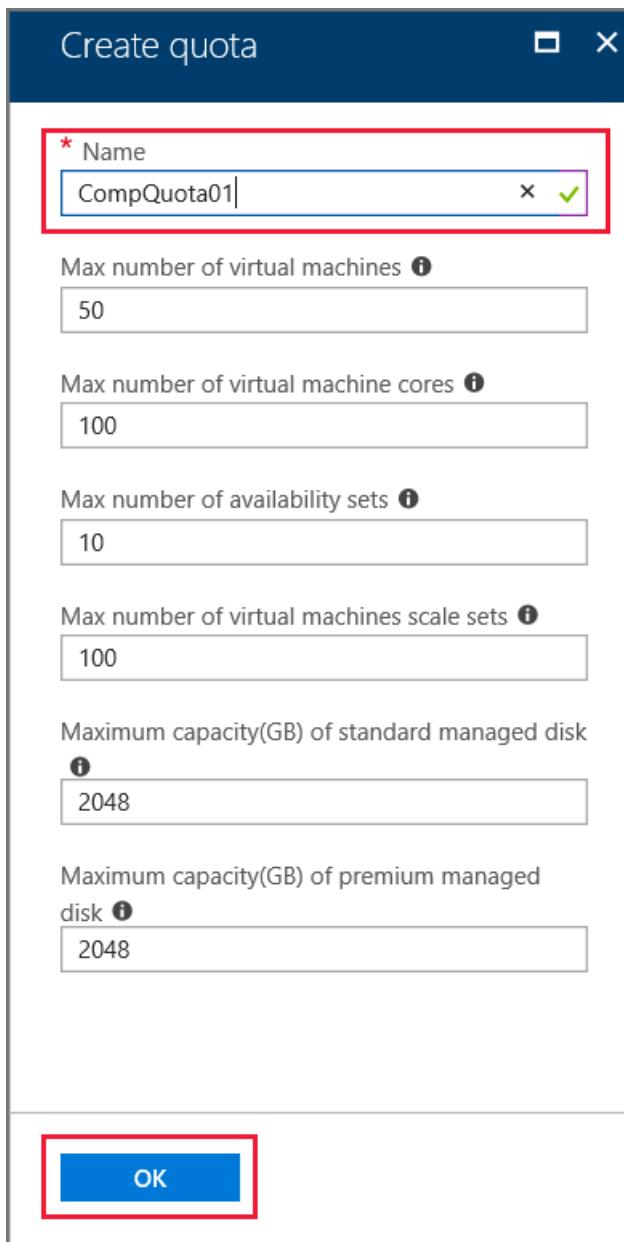
The screenshot shows two overlapping dialog boxes. The left dialog is titled 'New plan' and contains fields for 'Display name' (UserPlan01), 'Resource name' (userplan01), and 'Resource group' (radio buttons for 'Create new' or 'Use existing'). A section for 'Services' is highlighted with a red box, showing 'Not selected'. Another section for 'Quotas' is also present. The right dialog is titled 'Services' and lists five services: Microsoft.Compute local, Microsoft.KeyVault local, Microsoft.Network local, Microsoft.Storage local, and Microsoft.Subscriptions local. The first three services have their checkboxes checked and are highlighted with a red box. A 'Select' button is at the bottom of this dialog.

6. Select **Quotas**, and then select the first service that you want to create a quota for. For an IaaS quota, use the following example as a guide for configuring quotas for the Compute, Network, and Storage services.

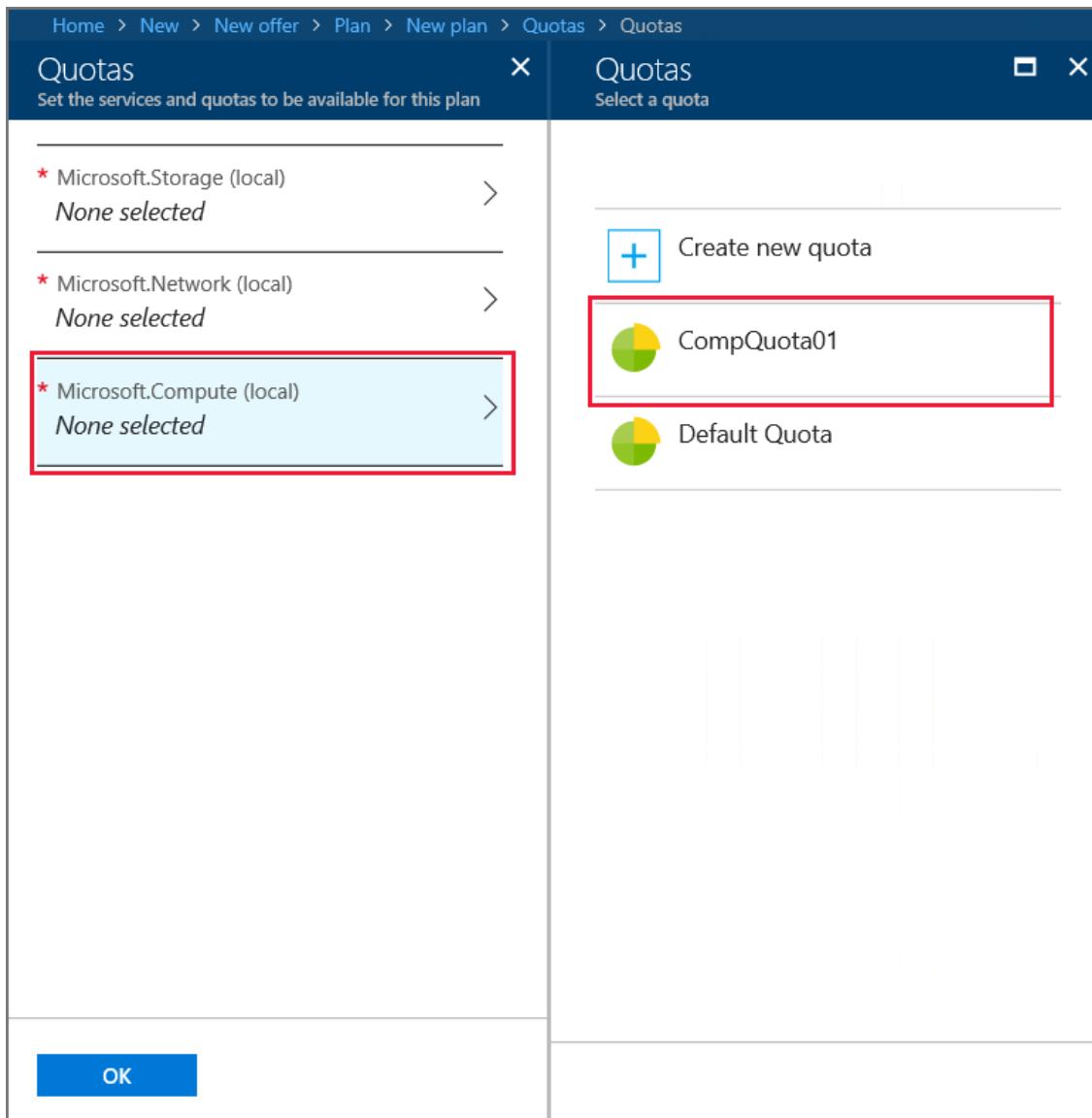
- First, create a quota for the Compute service. In the namespace list, select **Microsoft.Compute** and then select **Create new quota**.

The screenshot shows three dialog boxes. The left one is 'New plan' with fields for 'Display name' (UserPlan01), 'Resource name' (userplan01), and 'Resource group' (radio buttons for 'Create new' or 'Use existing'). A 'Services' section is highlighted with a red box, showing 'Microsoft.Storage and 2 other'. The middle dialog is 'Quotas' with sections for 'Microsoft.Storage (local)', 'Microsoft.Network (local)', and 'Microsoft.Compute (local)'. The 'Microsoft.Compute (local)' section is highlighted with a red box. The right dialog is 'Quotas' with a 'Create new quota' button and a 'Default Quota' entry, which is also highlighted with a red box. Both the 'Quotas' dialogs have an 'OK' button at the bottom.

- In **Create quota**, enter a name for the quota. You can change or accept any of the quota values that are shown. In this example, we accept the default settings and select **OK**.



- Pick **Microsoft.Compute** in the namespace list, and then select the quota that you created. This step links the quota to the Compute service.



Repeat these steps for the Network and Storage services. When you're finished, select **OK** in **Quotas** to save all the quotas.

7. In **New plan**, select **OK**.
8. Under **Plan**, select the new plan and then **Select**.
9. In **New offer**, select **Create**. You'll see a notification when the offer is created.
10. On the dashboard menu, select **Offers** and then pick the offer you created.
11. Select **Change State**, and then chose **Public**.

The screenshot shows the Azure portal interface for managing offers. On the left, there's a navigation bar with 'Offers' selected. The main area is titled 'useroffer01' and shows various settings like 'Overview', 'Activity log', and 'Access control (IAM)'. A sidebar on the right lists sections such as 'SETTINGS', 'USERS', and 'PLANS'. The 'Change state' button in the top right is highlighted with a red box. A tooltip for 'Public' is shown above it. The offer details include resource group 'RG01', status 'Decommissioned', and subscription information.

Next steps

In this tutorial you learned how to:

- Create an offer.
- Create a plan.
- Assign services and quotas to a plan.
- Assign a plan to an offer.

Advance to the next tutorial to learn how to:

[Test the services offered in this tutorial](#)

Tutorial: Test a service offering

4 minutes to read • [Edit Online](#)

In the previous tutorial, you created an offer for users. This tutorial shows you how to test that offer, by using it to create a subscription. You then create and deploy resources to the foundational services entitled by the subscription.

In this tutorial, you learn how to:

- Create a subscription
- Create and deploy resources

Prerequisites

Before starting this tutorial, you must complete the following prerequisites:

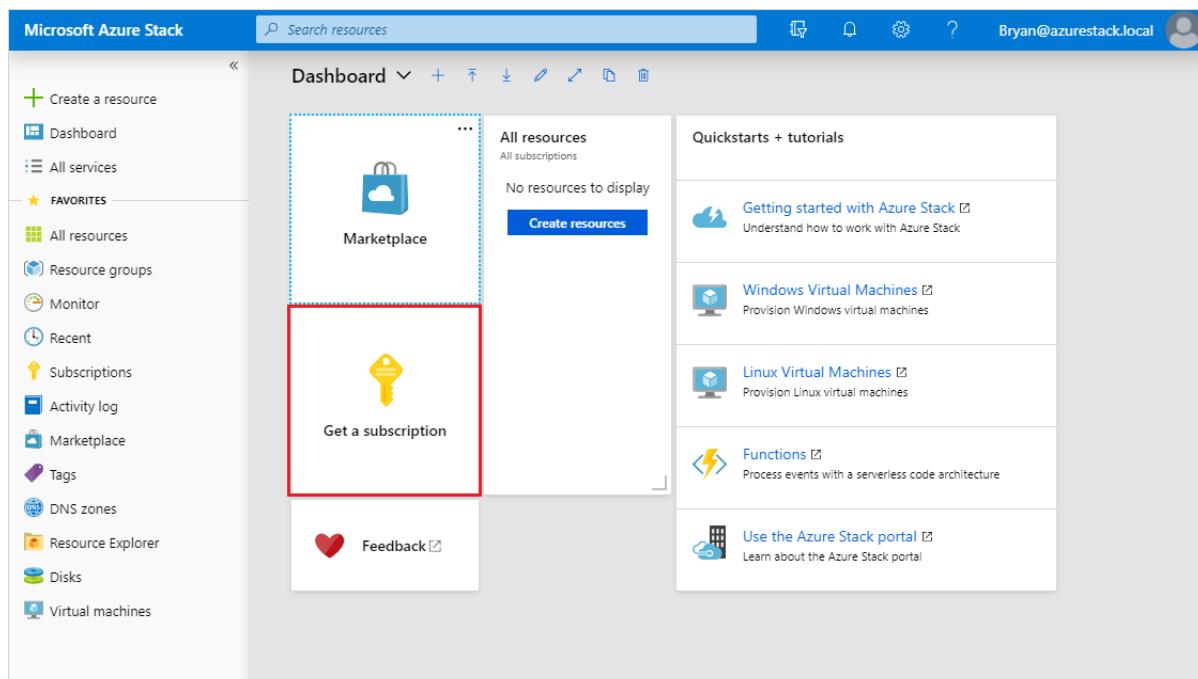
- Complete the [Offer a service to users](#) tutorial. In it, you learn how to create the offer used by this tutorial.
- The offer you subscribe to in this tutorial enables deployment of a virtual machine (VM) resource. If you'd like to test VM deployment, you must first make a VM image available in Azure Stack Hub Marketplace, by downloading it from Azure Marketplace. See [Download marketplace items from Azure to Azure Stack Hub](#) for instructions.

Subscribe to the offer

1. Sign in to the user portal with a user account

- For an integrated system, the URL varies based on your operator's region and external domain name, using the format <https://portal.<region>.<FQDN>>.
- If you're using the Azure Stack Development Kit, the portal address is <https://portal.local.azurestack.external>.

2. Select the **Get a Subscription** tile.



3. In **Get a Subscription**, enter a name for your new subscription in the **Display Name** field. Select **Offer**, and then choose the offer you created in the previous tutorial, from the **Choose an offer** list. Select **Create**.

The screenshot shows two adjacent windows. The left window is titled 'Get a subscription' and contains fields for 'Display name' (with placeholder 'Type a friendly name for the subscription') and 'Offer' (with a link 'Select an offer'). The right window is titled 'Choose an offer' and lists 'Pay as you go'. A note at the bottom left of the 'Get a subscription' window states: 'Note: After your subscription is created, you must refresh the portal to start accessing the new services in your subscription.' A blue 'Create' button is highlighted with a red box at the bottom left of the 'Get a subscription' window.

4. To view the subscription, select **All services**, and then under the **GENERAL** category select **Subscriptions**. Select your new subscription to view the offer it's associated with, and its properties.

NOTE

After you subscribe to an offer, you might have to refresh the portal to see which services are part of the new subscription.

Deploy a storage account resource

From the user portal, you provision a storage account using the subscription you created in the previous section.

1. Sign in to the user portal with a user account.
2. Select **+Create a resource > Data + Storage > Storage account - blob, file, table, queue**.
3. In **Create storage account**, provide the following information:
 - Enter a **Name**
 - Select your new **Subscription**
 - Select a **Resource group** (or create a one.)
 - Select **Create** to create the storage account.
4. Once deployment starts, you return to the dashboard. To see the new storage account, select **All resources**.

Search for the storage account and select its name from the search results. From here, you can manage the storage account and its contents.

Deploy a virtual machine resource

From the user portal, you provision a virtual machine using the subscription you created in the previous section.

1. Sign in to the user portal with a user account.
2. Select **+Create a resource > Compute > <image-name>**, where "image-name" is the name of the virtual machine you downloaded in prerequisites.
3. In **Create virtual machine / Basics**, provide the following information:
 - Enter a **Name** for the VM.
 - Enter a **User name** for the administrator account.
 - For Linux VMs, select "Password" for **Authentication type**.
 - Enter a **Password** and the same for **Confirm password**, for the administrator account.
 - Select your new **Subscription**.
 - Select a **Resource group** (or create a one).
 - Select **OK** to validate this information and continue.
4. In **Choose a size**, filter the list if necessary, select a VM SKU, and select **Select**.
5. In **Settings**, specify the port(s) to be opened under **Select public inbound ports**, and select **OK**.

NOTE

Selecting "RDP(3389)", for example, allows you to connect to the VM remotely when it's running.

6. In **Summary**, review your choices, then select **OK** to create the virtual machine.
7. Once deployment starts, you return to the dashboard. To see the new virtual machine, select **All resources**. Search for the virtual machine and select its name from the search results. From here, you can access and manage the virtual machine.

NOTE

Full deployment and starting of the VM can take several minutes. Once the VM is ready for use, the [status](#) will change to "Running".

Deploy a virtual machine resource (1901 and earlier)

From the user portal, you provision a virtual machine using the new subscription.

1. Sign in to the user portal with a user account.
2. On the dashboard, select **+Create a resource > Compute > Windows Server 2016 Datacenter Eval**, and then select **Create**.
3. In **Basics**, provide the following information:
 - Enter a **Name**
 - Enter a **User name**
 - Enter a **Password**
 - Choose your new **Subscription**

- Create a **Resource group** (or select an existing one.)
 - Select **OK** to save this information.
4. In **Choose a size**, select **A1 Standard**, and then **Select**.
5. In **Settings**, select **Virtual network**.
6. In **Choose virtual network**, select **Create new**.
7. In **Create virtual network**, accept all the defaults, and select **OK**.
8. Select **OK** in **Settings** to save the network configuration.
9. In **Summary**, select **OK** to create the virtual machine.
10. To see the new virtual machine, select **All resources**. Search for the virtual machine and select its name from the search results.

Next steps

In this tutorial you learned how to:

- Create a subscription
- Create and deploy resources

Next, learn about deploying resource providers for value-add services. They allow you to offer even more services to users in your plans:

- [Offer SQL on Azure Stack Hub](#)
- [Offer MySQL on Azure Stack Hub](#)
- [Offer App Service on Azure Stack Hub](#)

Capacity planning for Azure Stack Hub overview

2 minutes to read • [Edit Online](#)

When you're evaluating an Azure Stack Hub solution, consider the hardware configuration choices that have a direct impact on the overall capacity of the Azure Stack Hub cloud.

You will need to make choices regarding the CPU, memory density, storage configuration, and overall solution scale or number of servers. However, determining usable capacity will be different than a traditional virtualization solution because some capacity is already in use. Azure Stack Hub is built to host the infrastructure or management components within the solution itself. Also, some of the solution's capacity is reserved to support resiliency. Resiliency is defined as the updating of the solution's software in a way to minimize disruption of tenant workloads.

IMPORTANT

This capacity planning information and the [Azure Stack Hub Capacity Planner](#) are a starting point for Azure Stack Hub planning and configuration decisions. This information isn't intended to serve as a substitute for your own investigation and analysis. Microsoft makes no representations or warranties, express or implied, with respect to the information provided here.

Hyperconvergence and the scale unit

An Azure Stack Hub solution is built as a hyperconverged cluster of compute and storage. The convergence allows for the sharing of the hardware capacity in the cluster, referred to as a *scale unit*. In Azure Stack Hub, a scale unit provides the availability and scalability of resources. A scale unit consists of a set of Azure Stack Hub servers, referred to as *hosts*. The infrastructure software is hosted within a set of virtual machines (VMs), and shares the same physical servers as the tenant VMs. All Azure Stack Hub VMs are then managed by the scale unit's Windows Server clustering technologies and individual Hyper-V instances.

The scale unit simplifies the acquisition and management of Azure Stack Hub. The scale unit also allows for the movement and scalability of all services (tenant and infrastructure) across Azure Stack Hub.

The following topics provide more details about each component:

- [Azure Stack Hub compute](#)
- [Azure Stack Hub storage](#)
- [Azure Stack Hub Capacity Planner](#)

Azure Stack Hub compute capacity

9 minutes to read • [Edit Online](#)

The [virtual machine \(VM\) sizes](#) supported on Azure Stack Hub are a subset of those supported on Azure. Azure imposes resource limits along many vectors to avoid overconsumption of resources (server local and service-level). Without imposing some limits on tenant consumption, the tenant experiences will suffer when other tenants overconsume resources. For networking egress from the VM, there are bandwidth caps in place on Azure Stack Hub that match Azure limitations. For storage resources on Azure Stack Hub, storage IOPS limits avoid basic overconsumption of resources by tenants for storage access.

IMPORTANT

The [Azure Stack Hub Capacity Planner](#) does not consider or guarantee IOPS performance.

VM placement

The Azure Stack Hub placement engine places tenant VMs across the available hosts.

Azure Stack Hub uses two considerations when placing VMs. One, is there enough memory on the host for that VM type? And two, are the VMs a part of an [availability set](#) or are they [virtual machine scale sets](#)?

To achieve high availability of a multi-VM production system in Azure Stack Hub, virtual machines (VMs) are placed in an availability set that spreads them across multiple fault domains. A fault domain in an availability set is defined as a single node in the scale unit. Azure Stack Hub supports having an availability set with a maximum of three fault domains to be consistent with Azure. VMs placed in an availability set will be physically isolated from each other by spreading them as evenly as possible over multiple fault domains (Azure Stack Hub hosts). If there's a hardware failure, VMs from the failed fault domain will be restarted in other fault domains. If possible, they'll be kept in separate fault domains from the other VMs in the same availability set. When the host comes back online, VMs will be rebalanced to maintain high availability.

Virtual machine scale sets use availability sets on the back end and make sure each virtual machine scale set instance is placed in a different fault domain. This means they use separate Azure Stack Hub infrastructure nodes. For example, in a four-node Azure Stack Hub system, there may be a situation where a virtual machine scale set of three instances will fail at creation due to the lack of the four-node capacity to place three virtual machine scale set instances on three separate Azure Stack Hub nodes. In addition, Azure Stack Hub nodes can be filled up at varying levels before trying placement.

Azure Stack Hub doesn't overcommit memory. However, an overcommit of the number of physical cores is allowed.

Since placement algorithms don't look at the existing virtual to physical core overprovisioning ratio as a factor, each host could have a different ratio. As Microsoft, we don't provide guidance on the physical-to-virtual core ratio because of the variation in workloads and service level requirements.

Consideration for total number of VMs

There's a new consideration for accurately planning Azure Stack Hub capacity. With the 1901 update (and every update going forward), there's now a limit on the total number of VMs that can be created. This limit is intended to be temporary to avoid solution instability. The source of the stability issue at higher numbers of VMs is being addressed but a specific timeline for remediation hasn't been determined. There's now a per-server limit of 60 VMs with a total solution limit of 700. For example, an eight-server Azure Stack Hub VM limit would be 480 (8 * 60).

60). For a 12 to 16 server Azure Stack Hub solution, the limit would be 700. This limit has been created keeping all the compute capacity considerations in mind, such as the resiliency reserve and the CPU virtual-to-physical ratio that an operator would like to maintain on the stamp. For more information, see the new release of the capacity planner.

If the VM scale limit is reached, the following error codes are returned as a result: `VMsPerScaleUnitLimitExceeded`, `VMsPerScaleUnitNodeLimitExceeded`.

Considerations for deallocation

When a VM is in the *deallocated* state, memory resources aren't being used. This allows others VMs to be placed in the system.

If the deallocated VM is then started again, the memory usage or allocation is treated like a new VM placed into the system and available memory is consumed.

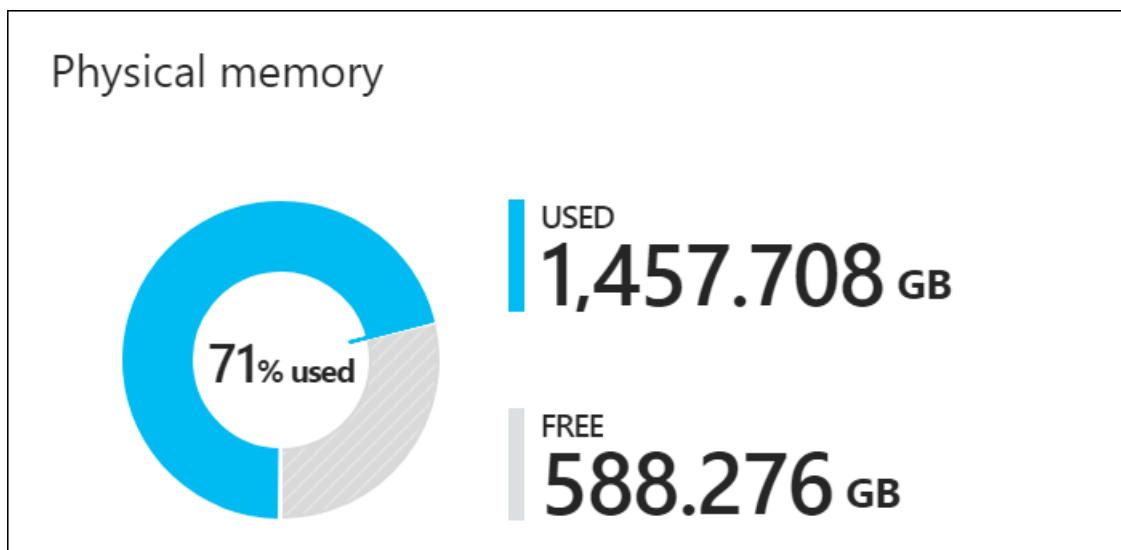
If there's no available memory, then the VM won't start.

Azure Stack Hub memory

Azure Stack Hub is designed to keep VMs running that have been successfully provisioned. For example, if a host is offline because of a hardware failure, Azure Stack Hub will attempt to restart that VM on another host. A second example during patching and updating of the Azure Stack Hub software. If there's a need to reboot a physical host, an attempt is made to move the VMs executing on that host to another available host in the solution.

This VM management or movement can only be achieved if there's reserved memory capacity to allow for the restart or migration to occur. A portion of the total host memory is reserved and unavailable for tenant VM placement.

You can review a pie chart in the administrator portal that shows the free and used memory in Azure Stack Hub. The following diagram shows the physical memory capacity on an Azure Stack Hub scale unit in the Azure Stack Hub:



Used memory is made up of several components. The following components consume the memory in the used section of the pie chart:

- **Host OS usage or reserve:** The memory used by the operating system (OS) on the host, virtual memory page tables, processes that are running on the host OS, and the Spaces Direct memory cache. Since this value is dependent on the memory used by the different Hyper-V processes running on the host, it can fluctuate.
- **Infrastructure services:** The infrastructure VMs that make up Azure Stack Hub. As of the 1904 release version of Azure Stack Hub, this entails ~31 VMs that take up 242 GB + (4 GB x # of nodes) of memory. The memory

utilization of the infrastructure services component may change as we work on making our infrastructure services more scalable and resilient.

- **Resiliency reserve:** Azure Stack Hub reserves a portion of the memory to allow for tenant availability during a single host failure as well as during patch and update to allow for successful live migration of VMs.
- **Tenant VMs:** The tenant VMs created by Azure Stack Hub users. In addition to running VMs, memory is consumed by any VMs that have landed on the fabric. This means that VMs in "Creating" or "Failed" state, or VMs shut down from within the guest, will consume memory. However, VMs that have been deallocated using the stop deallocated option from portal/powershell/cli won't consume memory from Azure Stack Hub.
- **Value-add resource providers (RPs):** VMs deployed for the value-add RPs like SQL, MySQL, App Service, and so on.

The best way to understand memory consumption on the portal is to use the [Azure Stack Hub Capacity Planner](#) to see the impact of various workloads. The following calculation is the same one used by the planner.

This calculation results in the total available memory that can be used for tenant VM placement. This memory capacity is for the entirety of the Azure Stack Hub scale unit.

Available memory for VM placement = total host memory - resiliency reserve - memory used by running tenant VMs - Azure Stack Hub Infrastructure Overhead¹

Resiliency reserve = $H + R * ((N-1) * H) + V * (N-2)$

Where:

- H = Size of single server memory
- N = Size of Scale Unit (number of servers)
- R = The operating system reserve for OS overhead, which is .15 in this formula²
- V = Largest VM in the scale unit

¹ Azure Stack Hub Infrastructure overhead = 242 GB + (4 GB x # of nodes). Approximately 31 VMs are used to host Azure Stack Hub's infrastructure and, in total, consume about 242 GB + (4 GB x # of nodes) of memory and 146 virtual cores. The rationale for this number of VMs is to satisfy the needed service separation to meet security, scalability, servicing, and patching requirements. This internal service structure allows for the future introduction of new infrastructure services as they're developed.

² Operating system reserve for overhead = 15% (.15) of node memory. The operating system reserve value is an estimate and will vary based on the physical memory capacity of the server and general operating system overhead.

The value V, largest VM in the scale unit, is dynamically based on the largest tenant VM memory size. For example, the largest VM value could be 7 GB or 112 GB or any other supported VM memory size in the Azure Stack Hub solution. Changing the largest VM on the Azure Stack Hub fabric will result in an increase in the resiliency reserve and also to the increase in the memory of the VM itself.

Frequently Asked Questions

Q: My tenant deployed a new VM, how long will it take for the capability chart on the administrator portal to show remaining capacity?

A: The capacity blade refreshes every 15 minutes, so take that into consideration.

Q: The number of deployed VMs on my Azure Stack Hub hasn't changed, but my capacity is fluctuating. Why?

A: The available memory for VM placement has multiple dependencies, one of which is the host OS reserve. This value is dependent on the memory used by the different Hyper-V processes running on the host, which isn't a constant value.

Q: What state do tenant VMs have to be in to consume memory?

A: In addition to running VMs, memory is consumed by any VMs that have landed on the fabric. This means that VMs that are in a "Creating" or "Failed" state will consume memory. VMs shut down from within the guest as opposed to stop deallocated from portal/powershell/cli will also consume memory.

Q: I have a four-host Azure Stack Hub. My tenant has 3 VMs that consume 56 GB of RAM (D5_v2) each. One of the VMs is resized to 112 GB RAM (D14_v2), and available memory reporting on dashboard resulted in a spike of 168 GB usage on the capacity blade. Subsequent resizing of the other two D5_v2 VMs to D14_v2 resulted in only 56 GB of RAM increase each. Why is this so?

A: The available memory is a function of the resiliency reserve maintained by Azure Stack Hub. The Resiliency reserve is a function of the largest VM size on the Azure Stack Hub stamp. At first, the largest VM on the stamp was 56 GB memory. When the VM was resized, the largest VM on the stamp became 112 GB memory which not only increased the memory used by that tenant VM but also increased the resiliency reserve. This change resulted in an increase of 56 GB (56 GB to 112 GB tenant VM memory increase) + 112 GB resiliency reserve memory increase. When subsequent VMs were resized, the largest VM size remained as the 112 GB VM and therefore there was no resultant resiliency reserve increase. The increase in memory consumption was only the tenant VM memory increase (56 GB).

NOTE

The capacity planning requirements for networking are minimal as only the size of the public VIP is configurable. For information about how to add more public IP addresses to Azure Stack Hub, see [Add public IP addresses](#).

Next steps

Learn about [Azure Stack Hub storage](#)

Azure Stack Hub storage capacity planning

4 minutes to read • [Edit Online](#)

The following sections provide Azure Stack Hub storage capacity planning information to assist in planning for the solution's storage needs.

Uses and organization of storage capacity

The hyperconverged configuration of Azure Stack Hub allows for the sharing of physical storage devices. There are three main divisions of the available storage that can be shared: the infrastructure, the temporary storage of the tenant virtual machines (VMs), and the storage backing the blobs, tables, and queues of the Azure Consistent Storage (ACS) services.

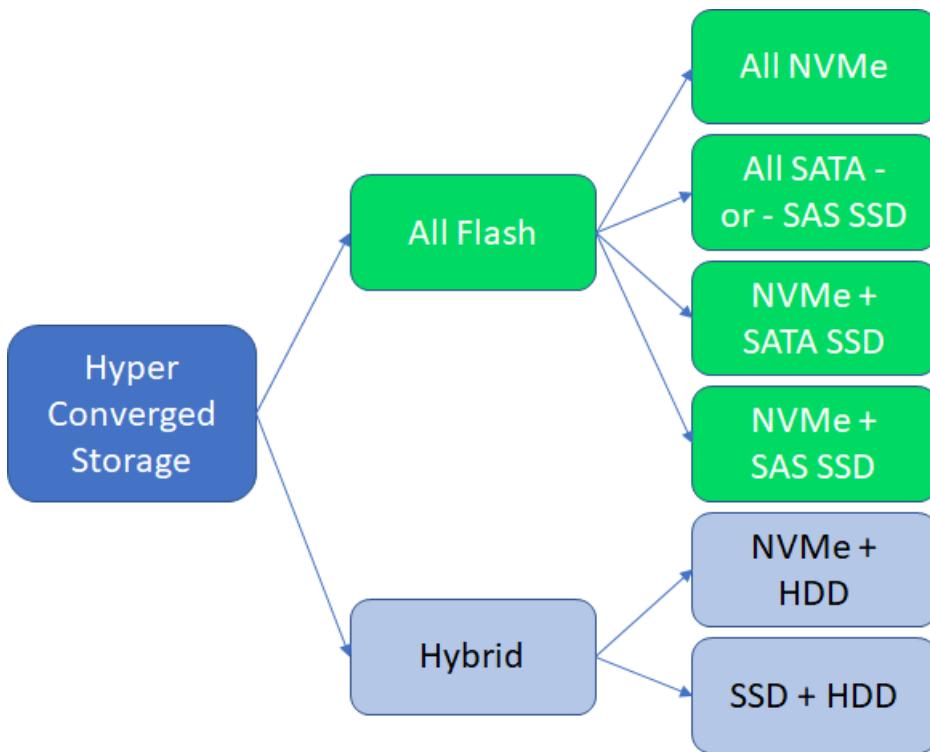
Storage Spaces Direct cache and capacity tiers

There's storage capacity used for the operating system, local logging, dumps, and other temporary infrastructure storage needs. This local storage capacity is separate (devices and capacity) from the storage devices brought under management of the Storage Spaces Direct configuration. The rest of the storage devices are placed in a single pool of storage capacity, regardless of the number of servers in the scale unit.

These devices are of two types: cache and capacity. Storage Spaces Direct consumes cache devices for write-back and read caching. The capacities of these cache devices, while used, aren't committed to the formatted and "visible" capacity of the formatted virtual disks. By contrast, Storage Spaces Direct does use capacity devices for this purpose, providing the "home location" of the managed data.

The Azure Stack Hub infrastructure directly allocates and manages all storage capacity. The operator doesn't need to make choices about configuration, allocation, capacity expansion. Azure Stack Hub automates these design decisions to align with the solution requirements, during either the initial installation and deployment or capacity expansion. Azure Stack Hub takes into consideration resiliency, reserved capacity for rebuilds, and other details, as part of the design.

Operators can choose between either an *all flash* or a *hybrid* storage configuration:



In the all flash configuration, the configuration can be either a two-tier or a single-tier configuration. If the configuration is single-tier, all capacity devices are of the same type (for example, NVMe or SATA SSD or SAS SSD), and cache devices aren't used. In a two-tier all flash configuration, the typical configuration is NVMe as the cache devices, and then either SATA or SAS SSDs as the capacity devices.

In the hybrid two-tier configuration, the cache is a choice among NVMe, SATA, or SAS SSD, and the capacity is HDD.

A brief summary of the Storage Spaces Direct and Azure Stack Hub storage configuration is as follows:

- One Storage Spaces Direct pool per scale unit (all storage devices are configured within a single pool).
- Virtual disks are created as a three-copy mirror for best performance and resiliency.
- Each virtual disk is formatted as an ReFS file system.
- Virtual disk capacity is calculated and allocated in a way as to leave one capacity device's amount of data capacity unallocated in the pool. This is the equivalent of one capacity drive per server.
- Each ReFS file system has BitLocker enabled for data-at-rest encryption.

The virtual disks created automatically and their capacities are as follows:

NAME	CAPACITY CALCULATION	DESCRIPTION
Local/boot device	Minimum of 340 GB ¹	Individual server storage for operating system images and "local" infrastructure VMs.
Infrastructure	3.5 TB	All Azure Stack Hub infrastructure usage.
VmTemp	See below ²	Tenant VMs have a temporary disk attached and that data is stored in these virtual disks.
ACS	See below ³	Azure Consistent Storage capacity for servicing blobs, tables, and queues.

¹ Minimum storage capacity required of the Azure Stack Hub solution partner.

² The virtual disk size used for tenant VM temporary disks is calculated as a ratio of the physical memory of the server. The temporary disk is a ratio of the physical memory assigned to the VM. The allocation done for "temp disk" storage in Azure Stack Hub captures most use cases but might not be able to satisfy all temp disk storage needs. The ratio is a trade-off between making temporary storage available and not consuming a majority of the solution's storage capacity for temp disk capacity only. One temporary storage disk is created per server in the scale unit. The capacity of the temporary storage doesn't grow beyond 10 percent of the overall available storage capacity in the storage pool of the scale unit. The calculation is something like the following example:

```
DesiredTempStoragePerServer = PhysicalMemory * 0.65 * 8
TempStoragePerSolution = DesiredTempStoragePerServer * NumberOfServers
PercentOfTotalCapacity = TempStoragePerSolution / TotalAvailableCapacity
If (PercentOfTotalCapacity <= 0.1)
    TempVirtualDiskSize = DesiredTempStoragePerServer
Else
    TempVirtualDiskSize = (TotalAvailableCapacity * 0.1) / NumberOfServers
```

³ The virtual disks created for use by ACS are a simple division of the remaining capacity. As noted, all virtual disks are a three-way mirror and one capacity drive's worth of capacity for each server is unallocated. The various virtual disks previously enumerated are allocated first and the remaining capacity is then used for the ACS virtual disks.

Next steps

Learn about the [Azure Stack Hub Capacity Planner](#).

Azure Stack Hub Capacity Planner

5 minutes to read • [Edit Online](#)

The Azure Stack Hub Capacity Planner is a spreadsheet that shows how different allocations of computing resources would fit across a selection of hardware offerings.

Worksheet descriptions

The following table describes each worksheet in the Azure Stack Hub Capacity Planner, which can be downloaded from <https://aka.ms/azstackcapacityplanner>.

WORKSHEET NAME	DESCRIPTION
Version-Disclaimer	Purpose of the calculator, version number, and release date.
Instructions	Step-by-step instructions to model capacity planning for a collection of virtual machines (VMs).
DefinedSolutionSKUs	Table with up to five hardware definitions. The entries are examples. Change the details to match system configurations under consideration.
DefineByVMFootprint	Find the appropriate hardware SKU by comparing configurations with different sizes and quantities of VMs.
DefineByWorkloadFootprint	Find the appropriate hardware SKU by creating a collection of Azure Stack Hub workloads.

DefinedSolutionSKUs instructions

This worksheet has up to five hardware definition examples. Change details to match the system configurations under consideration.

Hardware selections provided by authorized hardware partners

Azure Stack Hub is delivered as an integrated system with software installed by solution partners. Solution partners provide their own authoritative versions of Azure Stack Hub capacity planning tools. Use those tools for final discussions of solution capacity.

Multiple ways to model computing resources

Resource modeling within the Azure Stack Hub Capacity Planner depends upon the various sizes of Azure Stack Hub VMs. VMs range in size from the smallest, Basic_0, up to the largest, Standard_Fsv2. You can model computing resource allocations in two different ways:

- Select a specific hardware offering and see which combinations of different resources fit.
- Create a specific combination of VM allocations and let Azure Resource Calculator show which available hardware SKUs can support this VM configuration.

This tool provides two methods for allocating VM resources: either as one single collection of VM resource allocations, or as a collection of up to six differing workload configurations. Each workload configuration can

contain a different allocation of available VM resources. The next sections have step-by-step instructions to create and use each of these allocation models. Only values contained in non-background shaded cells or within SKU pull-down lists on this worksheet should be modified. Changes made within shaded cells might break resource calculations.

DefineByVMFootprint instructions

To create a model by using a single collection of various sizes and quantities of VMs, select the **DefineByVMFootprint** tab and follow these steps:

1. In the upper right corner of this worksheet, use the provided pull-down list box controls to select an initial number of servers (between 4 and 16) that you want installed in each hardware system (SKU). This number of servers can be modified at any time during the modeling process to see how this affects overall available resources for your resource allocation model.
2. If you want to model various VM resource allocations against one specific hardware configuration, find the blue pull-down list box directly below the **Current SKU** label in the upper right corner of the page. Pull down this list box and select your desired hardware SKU.
3. You're now ready to begin adding variously sized VMs to your model. To include a particular VM type, enter a quantity value into the blue outlined box to the left of that VM entry.

NOTE

Total VM Storage refers to the total capacity of the data disk of the VM (the number of supported disks multiplied by the maximum capacity of a single disk [1 TB]). Based on the configuration indicators, we've populated the Available Storage Configurations table so you can choose your desired level of storage resource for each Azure Stack Hub VM. However, it's important to note that you can add or change the Available Storage Configurations table as necessary.

Each VM starts with an initially assigned local temp storage. To reflect the thin provisioning of temp storage, you can change the local-temp number to anything in the drop-down menu, including the maximum allowable temp storage amount.

4. As you add VMs, you'll see the charts that show available SKU resources changing. These charts allow you to see the effects of adding various sizes and quantities of VMs during the modeling process. Another way to view the effect of changes is to watch the **Consumed** and **Still Available** numbers, listed directly below the list of available VMs. These numbers reflect estimated values based on the currently selected hardware SKU.
5. When you've created your set of VMs, you can find the suggested hardware SKU by selecting **Suggested SKU**. This button is located in the upper right corner of the page, directly below the **Current SKU** label. Using this button, you can then modify your VM configurations and see which hardware supports each configuration.

DefineByWorkloadFootprint instructions

To create a model by using a collection of Azure Stack Hub workloads, select the **DefineByWorkloadFootprint** tab and follow this sequence of steps. You create Azure Stack Hub workloads by using available VM resources.

TIP

To change the provided storage size for an Azure Stack Hub VM, see the note from step 3 in the preceding section.

1. In the upper right corner of this worksheet, use the provided pull-down list box controls to select an initial

number of servers (between 4 and 16) that you want installed in each hardware system (SKU).

2. If you want to model various VM resource allocations against one specific hardware configuration, find the blue pull-down list box directly below the **Current SKU** label in the upper right corner of the page. Pull down this list box and select your desired hardware SKU.
3. Select the appropriate storage size for each of your desired Azure Stack Hub VMs on the **DefineByVMFootprint** page. This process is described in step three of the previous section. The storage size per VM is defined in the DefineByVMFootprint sheet.
4. Starting on the upper left of the **DefineByWorkloadFootprint** page, create configurations for up to six different workload types. Enter the quantity of each VM type contained within that workload. You do this by placing numeric values into the column directly below that workload's name. You can modify workload names to reflect the type of workloads that will be supported by this particular configuration.
5. You can include a particular quantity of each workload type by entering a value at the bottom of that column, directly below the **Quantity** label.
6. When you've created workload types and quantities, select **Suggested SKU** in the upper right corner of the page, directly below the **Current SKU** label. The smallest SKU with enough resources to support this overall configuration of workloads will display.
7. You can accomplish further modeling by modifying the number of servers selected for a hardware SKU or by changing the VM allocations or quantities within your workload configurations. The associated graphs display immediate feedback, showing how your changes affect the overall resource consumption.
8. When you're satisfied with your changes, select **Suggested SKU** again to display the SKU suggested for your new configuration. You can also select the drop-down menu to select your desired SKU.

Next steps

Learn about [datacenter integration considerations for Azure Stack Hub](#).

Azure Stack Hub datacenter integration walkthrough

6 minutes to read • [Edit Online](#)

This article describes the end-to-end process for Azure Stack Hub datacenter integration, from purchasing to post-deployment support. The integration is a collaborative project between the customer, a solution provider, and Microsoft. Click the following tabs to see the specific steps for each member of the project, and see the next sections for a summary of different phases for the project timeline.

- [Customer](#)
- [Partner](#)
- [Microsoft](#)

1. Describe use cases and requirements
2. Determine the billing model
3. Review and approve contracts
4. Complete the [Deployment Worksheet](#)
5. Make sure deployment prerequisites are met
6. Prepare the datacenter
7. Provide subscription info during deployment
8. Resolve any questions about the provided data

Planning

Microsoft or an Azure Stack Hub solution partner will help evaluate your goals. They'll help you decide questions like:

- Is Azure Stack Hub the right solution for your organization?
- What type of billing and licensing model will work for your organization?
- What size solution will you need?
- What are the power and cooling requirements?

Use the [Azure Stack Hub Capacity Planner](#) to investigate and analyze the best hardware capacity and configuration for your needs.

Ordering

Your organization commits to purchasing Azure Stack Hub, signs contracts and purchase orders, and provides the integration requirements data to the solution provider.

Pre-deployment

You decide how to integrate Azure Stack Hub into your datacenter. Microsoft collaborated with solution providers to publish a [deployment worksheet](#) to help you gather the necessary information. The [general datacenter integration considerations](#) article provides information that helps you complete the template, known as the Deployment Worksheet.

IMPORTANT

All prerequisites are investigated before ordering the solution to help prevent deployment delays. Verifying prerequisites can take time and require coordination and data gathering from different departments within your organization.

You'll choose the following items:

- **Azure Stack Hub connection model and identity provider.** You can choose to deploy Azure Stack Hub either [connected to the internet \(and to Azure\) or disconnected](#). To get the most benefit from Azure Stack Hub, including hybrid scenarios, you'd want to deploy connected to Azure. Choosing Active Directory Federation Services (AD FS) or Azure Active Directory (Azure AD) is a one-time decision that you must make at deployment time. **You can't change your identity provider later without redeploying the entire system.**
- **Licensing model.** The licensing model options for you to choose from depend on the kind of deployment you'll have. Your identity provider choice has no bearing on tenant virtual machines or the identity system and accounts they use.
 - Customers that are in a [disconnected deployment](#) have only one option: capacity-based billing.
 - Customers that are in a [connected deployment](#) can choose between capacity-based billing and pay-as-you-use. Capacity-based billing requires an Enterprise Agreement (EA) Azure Subscription for registration. This is necessary for registration, which provides for the availability of items in Azure Marketplace through an Azure Subscription.
- **Network integration.** [Network integration](#) is crucial for deployment, operation, and management of Azure Stack Hub systems. There are several considerations that go into ensuring the Azure Stack Hub solution is resilient and has a highly available physical infrastructure to support its operations.
- **Firewall integration.** It's recommended that you [use a firewall](#) to help secure Azure Stack Hub. Firewalls can help prevent DDOS attacks, intrusion detection, and content inspection. However, it should be noted that it can become a throughput bottleneck for Azure storage services.
- **Certificate requirements.** It's critical that all [required certificates](#) are available *before* an onsite engineer arrives at your datacenter for deployment.

Once all the pre-requisite information is gathered through the deployment worksheet, the solution provider will kick off the factory process based on the data collected to ensure a successful integration of Azure Stack Hub into your datacenter.

Hardware delivery

Your solution provider will work with you on scheduling when the solution will arrive to your facility. Once received and put in place, you'll need to schedule time with the solution provider to have an engineer come onsite to perform the Azure Stack Hub deployment.

It's **crucial** that all prerequisite data is locked and available *before the onsite engineer arrives to deploy the solution.*

- All certificates must be purchased and ready.
- Region name must be decided on.
- All network integration parameters are finalized and match with what you have shared with your solution provider.

TIP

If any of this information has changed, make sure to communicate the change with the solution provider before you schedule the actual deployment.

Onsite deployment

To deploy Azure Stack Hub, an onsite engineer from your hardware solution provider will need to be present to kick off the deployment. To ensure a successful deployment, ensure that all information provided through the deployment worksheet hasn't changed.

The following checks are what you should expect from the onsite engineer during the deployment experience:

- Check all the cabling and border connectivity to ensure the solution is properly put together and meets your requirements.
- Configure the solution HLH (Hardware Lifecycle Host), if present.
- Check to make sure all BMC, BIOS, and network settings are correct.
- Make sure firmware for all components is at the latest approved version by the solution.
- Start the deployment.

NOTE

A deployment procedure by the onsite engineer might take about one business week to complete.

Post-deployment

Several steps must be performed by the partner before the solution is handed off to the customer in the post-integration phase. In this phase, validation is important to ensure the system is deployed and performing correctly.

Actions that should be taken by the OEM Partner are:

- [Run test-azurestack](#).
- [Registration with Azure](#).
- [Marketplace Syndication](#).
- Backup Switch Configuration and HLH Configuration files.
- Remove DVM.
- Prepare a customer summary for deployment.
- [Check updates to make sure the solution software is updated to the latest version](#).

There are several steps that are required or optional depending on the installation type.

- If deployment was completed using [AD FS](#), then the Azure Stack Hub stamp will need to be integrated with customer's own AD FS.

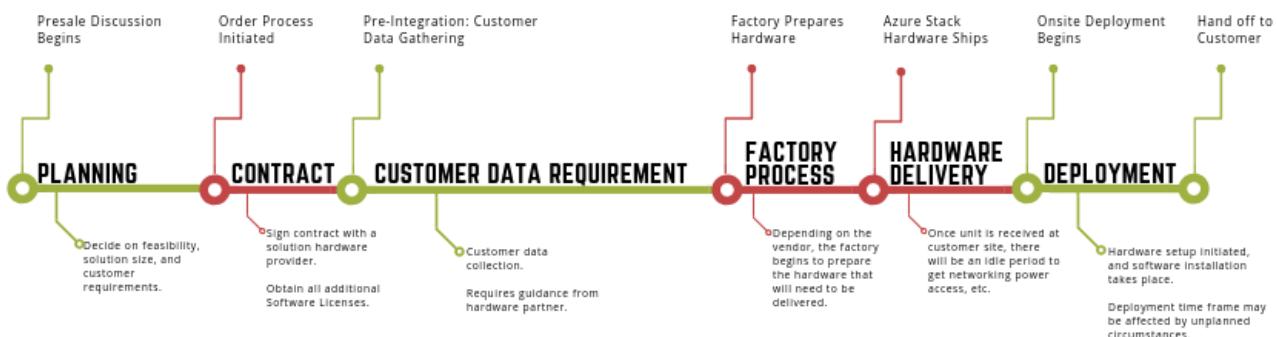
NOTE

This step is the responsibility of the customer, although the partner may optionally choose to offer services to do this.

- Integration with an existing monitoring system from the respective partner.

- [System Center Operations Manager Integration](#) also supports fleet management capabilities.
- [Nagios Integration](#).

Schedule



Support

Azure Stack Hub enables an Azure-consistent, integrated support experience that covers the full system lifecycle. To fully support Azure Stack Hub integrated systems, customers need two support contracts: one with Microsoft (or their Cloud Solution Provider) for Azure services support and one with the hardware provider for system support. The integrated support experience provides coordinated escalation and resolution so that customers get a consistent support experience no matter whom they call first. For customers who already have Premier, Azure - Standard / ProDirect or Partner support with Microsoft, Azure Stack Hub software support is included.

The integrated support experience makes use of a Case Exchange mechanism for bi-directional transfer of support cases and case updates between Microsoft and the hardware partner. Microsoft Azure Stack Hub will follow the [Modern Lifecycle policy](#).

Next steps

Learn more about [general datacenter integration considerations](#).

Datacenter integration planning considerations for Azure Stack Hub integrated systems

13 minutes to read • [Edit Online](#)

If you're interested in an Azure Stack Hub integrated system, you should understand the major planning considerations around deployment and how the system fits into your datacenter. This article provides a high-level overview of these considerations to help you make important infrastructure decisions for your Azure Stack Hub integrated systems. An understanding of these considerations helps when working with your OEM hardware vendor while they deploy Azure Stack Hub to your datacenter.

NOTE

Azure Stack Hub integrated systems can only be purchased from authorized hardware vendors.

To deploy Azure Stack Hub, you need to provide planning information to your solution provider before deployment starts to help the process go quickly and smoothly. The information required ranges across networking, security, and identity information with many important decisions that may require knowledge from many different areas and decision makers. You'll need people from multiple teams in your organization to ensure that you have all required information ready before deployment. It can help to talk to your hardware vendor while collecting this information because they might have helpful advice.

While researching and collecting the required information, you might need to make some pre-deployment configuration changes to your network environment. These changes could include reserving IP address spaces for the Azure Stack Hub solution as well as configuring your routers, switches, and firewalls to prepare for the connectivity to the new Azure Stack Hub solution switches. Make sure to have the subject area expert lined up to help you with your planning.

Capacity planning considerations

When you evaluate an Azure Stack Hub solution for acquisition, you make hardware configuration choices which have a direct impact on the overall capacity of the Azure Stack Hub solution. These include the classic choices of CPU, memory density, storage configuration, and overall solution scale (for example, number of servers). Unlike a traditional virtualization solution, the simple arithmetic of these components to determine usable capacity doesn't apply. The first reason is that Azure Stack Hub is architected to host the infrastructure or management components within the solution itself. The second reason is that some of the solution's capacity is reserved in support of resiliency by updating the solution's software in a way that minimizes disruption of tenant workloads.

The [Azure Stack Hub capacity planner spreadsheet](#) helps you make informed decisions for planning capacity in two ways. The first is by selecting a hardware offering and attempting to fit a combination of resources. The second is by defining the workload that Azure Stack Hub is intended to run to view the available hardware SKUs that can support it. Finally, the spreadsheet is intended as a guide to help in making decisions related to Azure Stack Hub planning and configuration.

The spreadsheet isn't intended to serve as a substitute for your own investigation and analysis. Microsoft makes no representations or warranties, express or implied, with respect to the information provided within the spreadsheet.

Management considerations

Azure Stack Hub is a sealed system, where the infrastructure is locked down both from a permissions and network perspective. Network access control lists (ACLs) are applied to block all unauthorized incoming traffic and all unnecessary communications between infrastructure components. This system makes it difficult for unauthorized users to access the system.

For daily management and operations, there's no unrestricted admin access to the infrastructure. Azure Stack Hub operators must manage the system through the administrator portal or through Azure Resource Manager (via PowerShell or the REST API). There's no access to the system by other management tools like Hyper-V Manager or Failover Cluster Manager. To help protect the system, third-party software (for example, agents) can't be installed inside the components of the Azure Stack Hub infrastructure. Interoperability with external management and security software occurs via PowerShell or the REST API.

Contact Microsoft Support when you need a higher level of access for troubleshooting issues that aren't resolved through alert mediation steps. Through support, there's a method to provide temporary full admin access to the system for more advanced operations.

Identity considerations

Choose identity provider

You'll need to consider which identity provider you want to use for Azure Stack Hub deployment, either Azure AD or AD FS. You can't switch identity providers after deployment without full system redeployment. If you don't own the Azure AD account and are using an account provided to you by your Cloud Solution Provider, and if you decide to switch provider and use a different Azure AD account, you'll have to contact your solution provider to redeploy the solution for you at your cost.

Your identity provider choice has no bearing on tenant virtual machines (VMs), the identity system, accounts they use, or whether they can join an Active Directory domain, and so on. These things are separate.

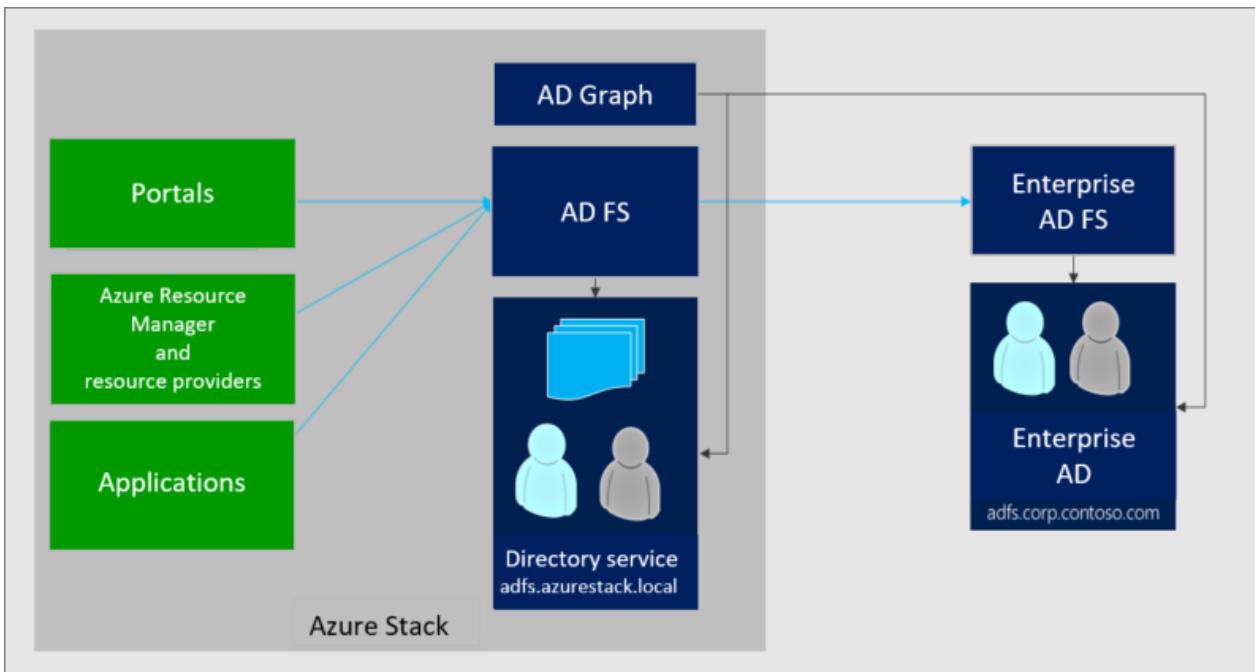
You can learn more about choosing an identity provider in the [Azure Stack Hub integrated systems connection models article](#).

AD FS and Graph integration

If you choose to deploy Azure Stack Hub using AD FS as the identity provider, you must integrate the AD FS instance on Azure Stack Hub with an existing AD FS instance through a federation trust. This integration allows identities in an existing Active Directory forest to authenticate with resources in Azure Stack Hub.

You can also integrate the Graph service in Azure Stack Hub with the existing Active Directory. This integration lets you manage Role-Based Access Control (RBAC) in Azure Stack Hub. When access to a resource is delegated, the Graph component looks up the user account in the existing Active Directory forest using the LDAP protocol.

The following diagram shows integrated AD FS and Graph traffic flow.



Licensing model

You must decide which licensing model you want to use. The available options depend on if you deploy Azure Stack Hub connected to the internet:

- For a [connected deployment](#), you can choose either pay-as-you-use or capacity-based licensing. Pay-as-you-use requires a connection to Azure to report usage, which is then billed through Azure commerce.
- Only capacity-based licensing is supported if you [deploy disconnected](#) from the internet.

For more information about the licensing models, see [Microsoft Azure Stack Hub packaging and pricing](#).

Naming decisions

You'll need to think about how you want to plan your Azure Stack Hub namespace, especially the region name and external domain name. The external fully qualified domain name (FQDN) of your Azure Stack Hub deployment for public-facing endpoints is the combination of these two names: <region>.<fqdn>. For example, *east.cloud.fabrikam.com*. In this example, the Azure Stack Hub portals would be available at the following URLs:

- <https://portal.east.cloud.fabrikam.com>
- <https://adminportal.east.cloud.fabrikam.com>

IMPORTANT

The region name you choose for your Azure Stack Hub deployment must be unique and will appear in the portal addresses.

The following table summarizes these domain naming decisions.

NAME	DESCRIPTION
------	-------------

NAME	DESCRIPTION
Region name	<p>The name of your first Azure Stack Hub region. This name is used as part of the FQDN for the public virtual IP addresses (VIPs) that Azure Stack Hub manages. Typically, the region name would be a physical location identifier such as a datacenter location.</p> <p>The region name must consist of only letters and numbers between 0-9. No special characters (like - , # , and so on) are allowed.</p>
External domain name	<p>The name of the Domain Name System (DNS) zone for endpoints with external-facing VIPs. Used in the FQDN for these public VIPs.</p>
Private (internal) domain name	<p>The name of the domain (and internal DNS zone) created on Azure Stack Hub for infrastructure management.</p>

Certificate requirements

For deployment, you'll need to provide Secure Sockets Layer (SSL) certificates for public-facing endpoints. At a high level, certificates have the following requirements:

- You can use a single wildcard certificate or you can use a set of dedicated certificates, and then use wildcards only for endpoints like storage and Key Vault.
- Certificates can be issued by a public trusted certificate authority (CA) or a customer-managed CA.

For more information about what PKI certificates are required to deploy Azure Stack Hub, and how to obtain them, see, [Azure Stack Hub Public Key Infrastructure certificate requirements](#).

IMPORTANT

The provided PKI certificate information should be used as general guidance. Before you acquire any PKI certificates for Azure Stack Hub, work with your OEM hardware partner. They'll provide more detailed certificate guidance and requirements.

Time synchronization

You must choose a specific time server which is used to synchronize Azure Stack Hub. Time synchronization is critical to Azure Stack Hub and its infrastructure roles because it's used to generate Kerberos tickets. Kerberos tickets are used to authenticate internal services with each other.

You must specify an IP for the time synchronization server. Although most of the components in the infrastructure can resolve a URL, some only support IP addresses. If you're using the disconnected deployment option, you must specify a time server on your corporate network that you're sure you can reach from the infrastructure network in Azure Stack Hub.

Connect Azure Stack Hub to Azure

For hybrid cloud scenarios, you'll need to plan how you want to connect Azure Stack Hub to Azure. There are two supported methods to connect virtual networks in Azure Stack Hub to virtual networks in Azure:

- **Site-to-site:** A virtual private network (VPN) connection over IPsec (IKE v1 and IKE v2). This type of

connection requires a VPN device or Routing and Remote Access Service (RRAS). For more information about VPN gateways in Azure, see [About VPN Gateway](#). The communication over this tunnel is encrypted and secure. However, bandwidth is limited by the maximum throughput of the tunnel (100-200 Mbps).

- **Outbound NAT:** By default, all VMs in Azure Stack Hub will have connectivity to external networks via outbound NAT. Each virtual network that's created in Azure Stack Hub gets a public IP address assigned to it. Whether the VM is directly assigned a public IP address or is behind a load balancer with a public IP address, it will have outbound access via outbound NAT using the VIP of the virtual network. This method only works for communication that's initiated by the VM and destined for external networks (either internet or intranet). It can't be used to communicate with the VM from outside.

Hybrid connectivity options

For hybrid connectivity, it's important to consider what kind of deployment you want to offer and where it will be deployed. You'll need to consider whether you need to isolate network traffic per tenant, and whether you'll have an intranet or internet deployment.

- **Single-tenant Azure Stack Hub:** An Azure Stack Hub deployment that looks, at least from a networking perspective, as if it's one tenant. There can be many tenant subscriptions, but like any intranet service, all traffic travels over the same networks. Network traffic from one subscription travels over the same network connection as another subscription and doesn't need to be isolated via an encrypted tunnel.
- **Multi-tenant Azure Stack Hub:** An Azure Stack Hub deployment where each tenant subscription's traffic that's bound for networks that are external to Azure Stack Hub must be isolated from other tenants' network traffic.
- **Intranet deployment:** An Azure Stack Hub deployment that sits on a corporate intranet, typically on private IP address space and behind one or more firewalls. The public IP addresses aren't truly public because they can't be routed directly over the public internet.
- **Internet deployment:** An Azure Stack Hub deployment that's connected to the public internet and uses internet-routable public IP addresses for the public VIP range. The deployment can still sit behind a firewall, but the public VIP range is directly reachable from the public internet and Azure.

The following table summarizes the hybrid connectivity scenarios with the pros, cons, and use cases.

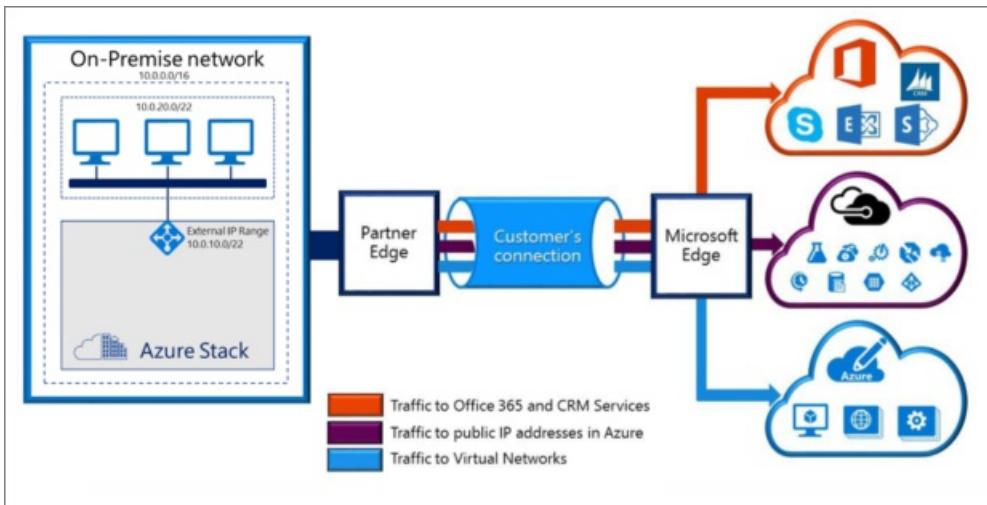
SCENARIO	CONNECTIVITY METHOD	PROS	CONS	GOOD FOR
Single tenant Azure Stack Hub, intranet deployment	Outbound NAT	Better bandwidth for faster transfers. Simple to implement; no gateways required.	Traffic not encrypted; no isolation or encryption outside the stack.	Enterprise deployments where all tenants are equally trusted. Enterprises that have an Azure ExpressRoute circuit to Azure.
Multi-tenant Azure Stack Hub, intranet deployment	Site-to-site VPN	Traffic from the tenant VNet to destination is secure.	Bandwidth is limited by site-to-site VPN tunnel. Requires a gateway in the virtual network and a VPN device on the destination network.	Enterprise deployments where some tenant traffic must be secured from other tenants.

SCENARIO	CONNECTIVITY METHOD	PROS	CONS	GOOD FOR
Single tenant Azure Stack Hub, internet deployment	Outbound NAT	Better bandwidth for faster transfers.	Traffic not encrypted; no isolation or encryption outside the stack.	Hosting scenarios where the tenant gets their own Azure Stack Hub deployment and a dedicated circuit to the Azure Stack Hub environment. For example, ExpressRoute and Multiprotocol Label Switching (MPLS).
Multi-tenant Azure Stack Hub, internet deployment	Site-to-site VPN	Traffic from the tenant VNet to destination is secure.	Bandwidth is limited by site-to-site VPN tunnel. Requires a gateway in the virtual network and a VPN device on the destination network.	Hosting scenarios where the provider wants to offer a multi-tenant cloud, where the tenants don't trust each other and traffic must be encrypted.

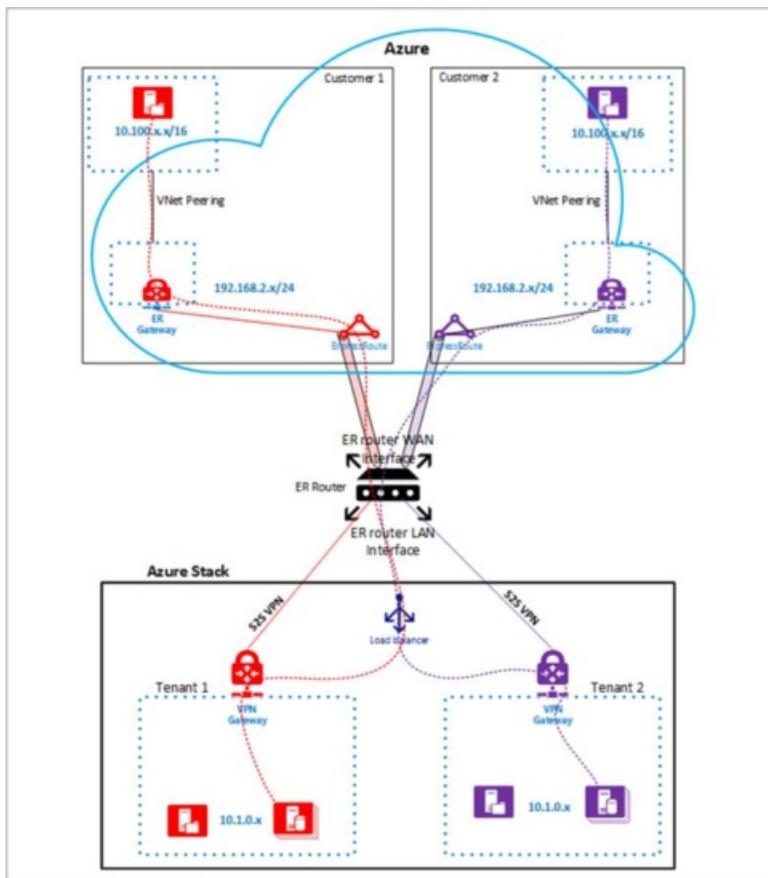
Using ExpressRoute

You can connect Azure Stack Hub to Azure via [ExpressRoute](#) for both single-tenant intranet and multi-tenant scenarios. You'll need a provisioned ExpressRoute circuit through [a connectivity provider](#).

The following diagram shows ExpressRoute for a single-tenant scenario (where "Customer's connection" is the ExpressRoute circuit).



The following diagram shows ExpressRoute for a multi-tenant scenario.



External monitoring

To get a single view of all alerts from your Azure Stack Hub deployment and devices, and to integrate alerts into existing IT Service Management workflows for ticketing, you can [integrate Azure Stack Hub with external datacenter monitoring solutions](#).

Included with the Azure Stack Hub solution, the hardware lifecycle host is a computer outside Azure Stack Hub that runs OEM vendor-provided management tools for hardware. You can use these tools or other solutions that directly integrate with existing monitoring solutions in your datacenter.

The following table summarizes the list of currently available options.

AREA	EXTERNAL MONITORING SOLUTION
Azure Stack Hub software	Azure Stack Hub Management Pack for Operations Manager Nagios plug-in REST-based API calls
Physical servers (BMCS via IPMI)	OEM hardware - Operations Manager vendor management pack OEM hardware vendor-provided solution Hardware vendor Nagios plug-ins. OEM partner-supported monitoring solution (included)
Network devices (SNMP)	Operations Manager network device discovery OEM hardware vendor-provided solution Nagios switch plug-in
Tenant subscription health monitoring	System Center Management Pack for Windows Azure

Note the following requirements:

- The solution you use must be agentless. You can't install third-party agents inside Azure Stack Hub components.
- If you want to use System Center Operations Manager, Operations Manager 2012 R2 or Operations Manager 2016 is required.

Backup and disaster recovery

Planning for backup and disaster recovery involves planning for both the underlying Azure Stack Hub infrastructure that hosts IaaS VMs and PaaS services, and for tenant apps and data. Plan for these things separately.

Protect infrastructure components

You can [back up Azure Stack Hub](#) infrastructure components to an SMB share that you specify:

- You'll need an external SMB file share on an existing Windows-based file server or a third-party device.
- Use this same share for the backup of network switches and the hardware lifecycle host. Your OEM hardware vendor will help provide guidance for backup and restore of these components because these are external to Azure Stack Hub. You're responsible for running the backup workflows based on the OEM vendor's recommendation.

If catastrophic data loss occurs, you can use the infrastructure backup to reseed deployment data such as:

- Deployment inputs and identifiers
- Service accounts
- CA root certificate
- Federated resources (in disconnected deployments)
- Plans, offers, subscriptions, and quotas
- RBAC policy and role assignments
- Key Vault secrets

Protect tenant apps on IaaS VMs

Azure Stack Hub doesn't back up tenant apps and data. You must plan for backup and disaster recovery protection to a target external to Azure Stack Hub. Tenant protection is a tenant-driven activity. For IaaS VMs, tenants can use in-guest technologies to protect file folders, app data, and system state. However, as an enterprise or service provider, you may want to offer a backup and recovery solution in the same datacenter or externally in a cloud.

To back up Linux or Windows IaaS VMs, you must use backup products with access to the guest operating system to protect file, folder, operating system state, and app data. You can use Azure Backup, System Center Datacenter Protection Manager, or supported third-party products.

To replicate data to a secondary location and orchestrate application failover if a disaster occurs, you can use Azure Site Recovery or supported third-party products. Also, apps that support native replication, like Microsoft SQL Server, can replicate data to another location where the app is running.

Learn more

- For information about use cases, purchasing, partners, and OEM hardware vendors, see the [Azure Stack Hub](#) product page.
- For information about the roadmap and geo-availability for Azure Stack Hub integrated systems, see the white paper: [Azure Stack Hub: An extension of Azure](#).

Next steps

Azure Stack Hub deployment connection models

Azure Stack Hub integrated systems connection models

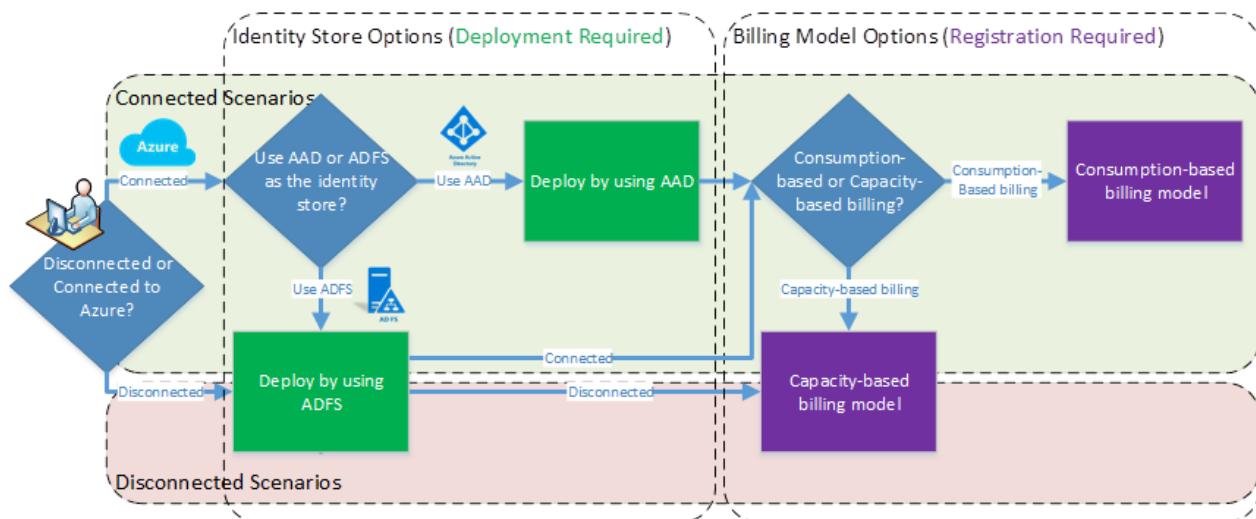
2 minutes to read • [Edit Online](#)

If you're interested in purchasing an Azure Stack Hub integrated system, you need to understand [several datacenter integration considerations](#) for Azure Stack Hub deployment to determine how the system will fit into your datacenter. In addition, you need to decide how you'll integrate Azure Stack Hub into your hybrid cloud environment. This article provides an overview of these major decisions including Azure connection models, identity store options, and billing model options.

If you decide to purchase an integrated system, your original equipment manufacturer (OEM) hardware vendor will help guide you through the planning process in more detail. The OEM hardware vendor also performs the actual deployment.

Choose an Azure Stack Hub deployment connection model

You can choose to deploy Azure Stack Hub either connected to the internet (and to Azure) or disconnected. Deploy connected to Azure to get the most benefit from Azure Stack Hub, including hybrid scenarios between Azure Stack Hub and Azure. This choice defines which options are available for your identity store (Azure Active Directory or Active Directory Federation Services) and billing model (pay as you use-based billing or capacity-based billing) as summarized in the following diagram and table:



IMPORTANT

This is a key decision point! Choosing Active Directory Federation Services (AD FS) or Azure Active Directory (Azure AD) is a one-time decision that you must make at deployment time. You can't change this later without re-deploying the entire system.

OPTIONS	CONNECTED TO AZURE	DISCONNECTED FROM AZURE
Azure AD	✓	
AD FS	✓	✓

OPTIONS	CONNECTED TO AZURE	DISCONNECTED FROM AZURE
Consumption-based billing	✓	
Capacity-based billing	✓	✓
Licensing	Enterprise Agreement or Cloud Solution Provider	Enterprise Agreement
Patch and update	Update package can be downloaded directly from the Internet to Azure Stack Hub	Required Also requires removable media and a separate connected device
Registration	Automated	Required Also requires removable media and a separate connected device

After you've decided on the Azure connection model to be used for your Azure Stack Hub deployment, additional connection-dependent decisions must be made for the identity store and billing method.

Next steps

[Azure connected Azure Stack Hub deployment decisions](#)

[Azure disconnected Azure Stack Hub deployment decisions](#)

Azure-connected deployment planning decisions for Azure Stack Hub integrated systems

4 minutes to read • [Edit Online](#)

After you've decided [how you'll integrate Azure Stack Hub into your hybrid cloud environment](#), you can finalize your Azure Stack Hub deployment decisions.

Deploying Azure Stack Hub connected to Azure means that you can have either Azure Active Directory (Azure AD) or Active Directory Federation Services (AD FS) for your identity store. You can also choose from either billing model: pay-as-you-use or capacity-based. A connected deployment is the default option because it allows customers to get the most value out of Azure Stack Hub, particularly for hybrid cloud scenarios that involve both Azure and Azure Stack Hub.

Choose an identity store

With a connected deployment, you can choose between Azure AD or AD FS for your identity store. A disconnected deployment, with no internet connectivity, can only use AD FS.

Your identity store choice has no bearing on tenant virtual machines (VMs). Tenant VMs may choose which identity store they want to connect to depending on how they'll be configured: Azure AD, Windows Server Active Directory domain-joined, workgroup, and so on. This is unrelated to the Azure Stack Hub identity provider decision.

For example, if you deploy IaaS tenant VMs on top of Azure Stack Hub, and want them to join a Corporate Active Directory Domain and use accounts from there, you still can. You aren't required to use the Azure AD identity store you select here for those accounts.

Azure AD identity store

Using Azure AD for your identity store requires two Azure AD accounts: a global admin account and a billing account. These accounts can be the same accounts, or different accounts. While using the same user account might be simpler and useful if you have a limited number of Azure accounts, your business needs might suggest using two accounts:

1. **Global admin account** (only required for connected deployments). This is an Azure account that's used to create apps and service principals for Azure Stack Hub infrastructure services in Azure AD. This account must have directory admin permissions to the directory that your Azure Stack Hub system will be deployed under. It will become the "cloud operator" Global Admin for the Azure AD user and is used for the following tasks:
 - To provision and delegate apps and service principals for all Azure Stack Hub services that need to interact with Azure AD and Graph API.
 - As the Service Administrator account. This account is the owner of the default provider subscription (which you can later change). You can log into the Azure Stack Hub administrator portal with this account, and can use it to create offers and plans, set quotas, and perform other administrative functions in Azure Stack Hub.
2. **Billing account** (required for both connected and disconnected deployments). This Azure account is used to establish the billing relationship between your Azure Stack Hub integrated system and the Azure commerce backend. This is the account that's billed for Azure Stack Hub fees. This account will also be used for offering items in the marketplace and other hybrid scenarios.

AD FS identity store

Choose this option if you want to use your own identity store, such as your corporate Active Directory, for your Service Administrator accounts.

Choose a billing model

You can choose either **Pay-as-you-use** or the **Capacity** billing model. Pay-as-you-use billing model deployments must be able to report usage through a connection to Azure at least once every 30 days. Therefore, the pay-as-you-use billing model is only available for connected deployments.

Pay-as-you-use

With the pay-as-you-use billing model, usage is charged to an Azure subscription. You only pay when you use the Azure Stack Hub services. If this is the model you decide on, you'll need an Azure subscription and the account ID associated with that subscription (for example, serviceadmin@contoso.onmicrosoft.com). EA, CSP, and CSL subscriptions are supported. Usage reporting is configured during [Azure Stack Hub registration](#).

NOTE

In most cases, Enterprise customers will use EA subscriptions, and service providers will use CSP or CSL subscriptions.

If you're going to use a CSP subscription, review the table below to identify which CSP subscription to use, as the correct approach depends on the exact CSP scenario:

SCENARIO	DOMAIN AND SUBSCRIPTION OPTIONS
You're a Direct CSP Partner or an Indirect CSP Provider , and you'll operate the Azure Stack Hub	Use a CSL (Common Service Layer) subscription. or Create an Azure AD tenant with a descriptive name in Partner Center. For example, <your organization>CSPAdmin with an Azure CSP subscription associated with it.
You're an Indirect CSP Reseller , and you'll operate the Azure Stack Hub	Ask your indirect CSP Provider to create an Azure AD tenant for your organization with an Azure CSP subscription associated with it using Partner Center.

Capacity-based billing

If you decide to use the capacity billing model, you must purchase an Azure Stack Hub Capacity Plan SKU based on the capacity of your system. You need to know the number of physical cores in your Azure Stack Hub to purchase the correct quantity.

Capacity billing requires an Enterprise Agreement (EA) Azure subscription for registration. The reason is that registration sets up the availability of items in the Marketplace, which requires an Azure subscription. The subscription isn't used for Azure Stack Hub usage.

Learn more

- For information about use cases, purchasing, partners, and OEM hardware vendors, see the [Azure Stack Hub](#) product page.
- For information about the roadmap and geo-availability for Azure Stack Hub integrated systems, see the white paper: [Azure Stack Hub: An extension of Azure](#).
- To learn more about Microsoft Azure Stack Hub packaging and pricing, [download the .pdf](#).

Next steps

Azure disconnected deployment planning decisions for Azure Stack Hub integrated systems

3 minutes to read • [Edit Online](#)

After you've decided [how you'll integrate Azure Stack Hub into your hybrid cloud environment](#), you can finish your Azure Stack Hub deployment decisions.

You can deploy and use Azure Stack Hub without a connection to the internet. However, with a disconnected deployment, you're limited to an Active Directory Federation Services (AD FS) identity store and the capacity-based billing model. Because multitenancy requires the use of Azure Active Directory (Azure AD), multitenancy isn't supported for disconnected deployments.

Choose this option if:

- You have security or other restrictions that require you to deploy Azure Stack Hub in an environment that isn't connected to the internet.
- You want to block data (including usage data) from being sent to Azure.
- You want to use Azure Stack Hub purely as a private cloud solution that's deployed to your corporate intranet, and aren't interested in hybrid scenarios.

TIP

Sometimes, this kind of environment is also referred to as a *submarine scenario*.

A disconnected deployment doesn't restrict you from later connecting your Azure Stack Hub instance to Azure for hybrid tenant VM scenarios. It means that you don't have connectivity to Azure during deployment or you don't want to use Azure AD as your identity store.

Features that are impaired or unavailable in disconnected deployments

Azure Stack Hub was designed to work best when connected to Azure, so it's important to note that there are some features and functionality that are either impaired or completely unavailable in the disconnected mode.

FEATURE	IMPACT IN DISCONNECTED MODE
VM deployment with DSC extension to configure VM post deployment	Impaired - DSC extension looks to the internet for the latest WMF.
VM deployment with Docker Extension to run Docker commands	Impaired - Docker will check the internet for the latest version and this check will fail.
Documentation links in the Azure Stack Hub Portal	Unavailable - Links like Give Feedback, Help, and Quickstart that use an internet URL won't work.
Alert remediation/mitigation that references an online remediation guide	Unavailable - Any alert remediation links that use an internet URL won't work.

FEATURE	IMPACT IN DISCONNECTED MODE
Marketplace - The ability to select and add Gallery packages directly from Azure Marketplace	Impaired - When you deploy Azure Stack Hub in a disconnected mode, you can't download marketplace items by using the Azure Stack Hub portal. However, you can use the marketplace syndication tool to download the marketplace items to a machine that has internet connectivity and then transfer them to your Azure Stack Hub environment.
Using Azure AD federation accounts to manage an Azure Stack Hub deployment	Unavailable - This feature requires connectivity to Azure. AD FS with a local Active Directory instance must be used instead.
App Services	Impaired - WebApps may require internet access for updated content.
Command Line Interface (CLI)	Impaired - CLI has reduced functionality for authentication and provisioning of service principals.
Visual Studio - Cloud discovery	Impaired - Cloud Discovery will either discover different clouds or won't work at all.
Visual Studio - AD FS	Impaired - Only Visual Studio Enterprise and Visual Studio Code support AD FS authentication.
Telemetry	Unavailable - Telemetry data for Azure Stack Hub and any third-party gallery packages that depend on telemetry data.
Certificates	Unavailable - internet connectivity is required for Certificate Revocation List (CRL) and Online Certificate Status Protocol (OSCP) services in the context of HTTPS.
Key Vault	Impaired - A common use case for Key Vault is to have an app read secrets at runtime. For this use case, the app needs a service principal in the directory. In Azure AD, regular users (non-admins) are by default allowed to add service principals. In Azure AD (using AD FS), they're not. This impairment places a hurdle in the end-to-end experience because one must always go through a directory admin to add any app.

Learn more

- For information about use cases, purchasing, partners, and OEM hardware vendors, see the [Azure Stack Hub](#) product page.
- For information about the roadmap and geo-availability for Azure Stack Hub integrated systems, see the white paper: [Azure Stack Hub: An extension of Azure](#).
- To learn more about Microsoft Azure Stack Hub packaging and pricing, [download the .pdf](#).

Next steps

[Datacenter network integration](#)

Network integration planning for Azure Stack

7 minutes to read • [Edit Online](#)

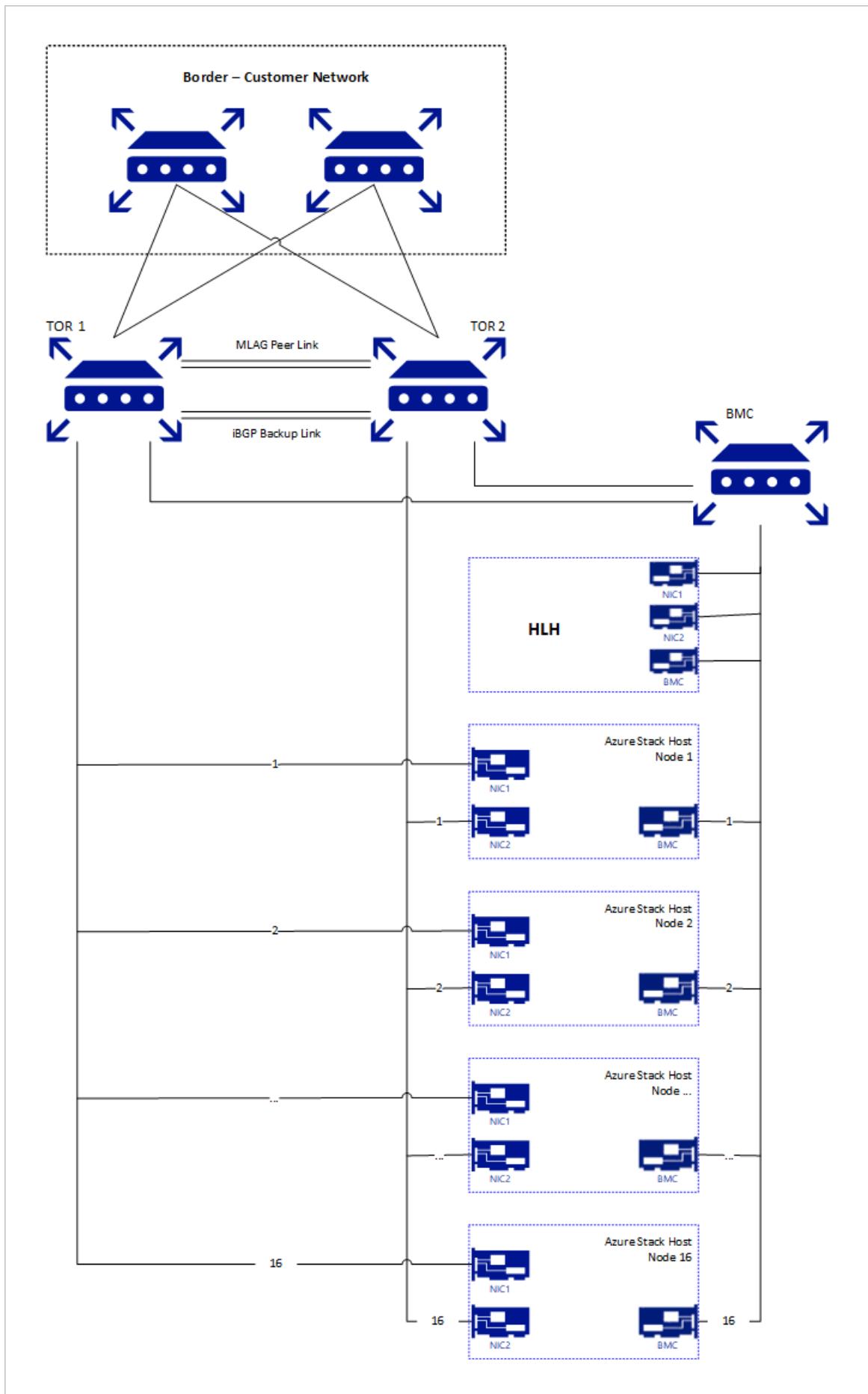
This article provides Azure Stack network infrastructure information to help you decide how to best integrate Azure Stack into your existing networking environment.

NOTE

To resolve external DNS names from Azure Stack (for example, www.bing.com), you need to provide DNS servers to forward DNS requests. For more information about Azure Stack DNS requirements, see [Azure Stack datacenter integration - DNS](#).

Physical network design

The Azure Stack solution requires a resilient and highly available physical infrastructure to support its operation and services. To integrate Azure Stack to the network it requires uplinks from the Top-of-Rack switches (ToR) to the nearest switch or router, which on this documentation is referred as Border. The ToRs can be uplinked to a single or a pair of Borders. The ToR is pre-configured by our automation tool, it expects a minimum of one connection between ToR and Border when using BGP Routing and a minimum of two connections (one per ToR) between ToR and Border when using Static Routing, with a maximum of four connections on either routing options. These connections are limited to SFP+ or SFP28 media and one GB, 10 GB, or 25-GB speeds. Check with your original equipment manufacturer (OEM) hardware vendor for availability. The following diagram presents the recommended design:



Logical Networks

Logical networks represent an abstraction of the underlying physical network infrastructure. They're used to organize and simplify network assignments for hosts, virtual machines (VMs), and services. As part of logical

network creation, network sites are created to define the virtual local area networks (VLANs), IP subnets, and IP subnet/VLAN pairs that are associated with the logical network in each physical location.

The following table shows the logical networks and associated IPv4 subnet ranges that you must plan for:

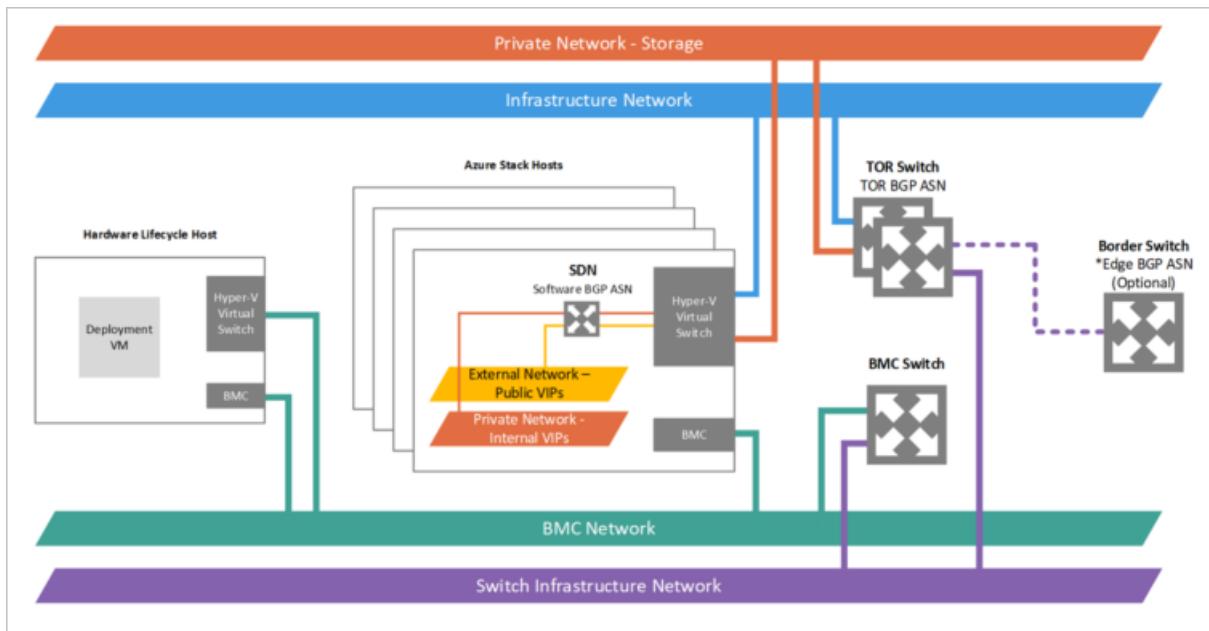
LOGICAL NETWORK	DESCRIPTION	SIZE
Public VIP	Azure Stack uses a total of 31 addresses from this network. Eight public IP addresses are used for a small set of Azure Stack services and the rest are used by tenant VMs. If you plan to use App Service and the SQL resource providers, 7 more addresses are used. The remaining 15 IPs are reserved for future Azure services.	/26 (62 hosts) - /22 (1022 hosts) Recommended = /24 (254 hosts)
Switch infrastructure	Point-to-point IP addresses for routing purposes, dedicated switch management interfaces, and loopback addresses assigned to the switch.	/26
Infrastructure	Used for Azure Stack internal components to communicate.	/24
Private	Used for the storage network, private VIPs, Infrastructure containers and other internal functions. Starting in 1910, the size for this subnet is changing to /20, for more details reference the Private network section in this article.	/20
BMC	Used to communicate with the BMCs on the physical hosts.	/26

NOTE

When the system is updated to 1910 version, an alert on the portal will remind the operator to run the new PEP cmdlet **Set-AzsPrivateNetwork** to add a new /20 Private IP space. Please see the [1910 release notes](#) for instructions on running the cmdlet. For more information and guidance on selecting the /20 private IP space, please see the [Private network](#) section in this article.

Network infrastructure

The network infrastructure for Azure Stack consists of several logical networks that are configured on the switches. The following diagram shows these logical networks and how they integrate with the top-of-rack (TOR), baseboard management controller (BMC), and border (customer network) switches.



BMC network

This network is dedicated to connecting all the baseboard management controllers (also known as BMC or service processors) to the management network. Examples include: iDRAC, iLO, iBMC, and so on. Only one BMC account is used to communicate with any BMC node. If present, the Hardware Lifecycle Host (HLH) is located on this network and may provide OEM-specific software for hardware maintenance or monitoring.

The HLH also hosts the Deployment VM (DVM). The DVM is used during Azure Stack deployment and is removed when deployment completes. The DVM requires internet access in connected deployment scenarios to test, validate, and access multiple components. These components can be inside and outside of your corporate network (for example: NTP, DNS, and Azure). For more information about connectivity requirements, see the [NAT section in Azure Stack firewall integration](#).

Private network

This /20 (4096 IPs) network is private to the Azure Stack region (doesn't route beyond the border switch devices of the Azure Stack system) and is divided into multiple subnets, here are some examples:

- **Storage network**: A /25 (128 IPs) network used to support the use of Spaces Direct and Server Message Block (SMB) storage traffic and VM live migration.
- **Internal virtual IP network**: A /25 network dedicated to internal-only VIPs for the software load balancer.
- **Container network**: A /23 (512 IPs) network dedicated to internal-only traffic between containers running infrastructure services.

Starting in 1910, the size for the Private Network will change to a /20 (4096 IPs) of private IP space. This network will be private to the Azure Stack system (doesn't route beyond the border switch devices of the Azure Stack system) and can be reused on multiple Azure Stack systems within your datacenter. While the network is private to Azure Stack, it must not overlap with other networks in the datacenter. For guidance on Private IP space, we recommend following the [RFC 1918](#).

This /20 Private IP space will be divided into multiple networks that will enable running the Azure Stack system internal infrastructure on containers in future releases. For more details, please refer to the [1910 release notes](#). In addition, this new Private IP space enables ongoing efforts to reduce the required routable IP space prior to deployment.

For systems deployed before 1910, this /20 subnet will be an additional network to be entered into systems after updating to 1910. The additional network will need to be provided to the system through the **Set-AzsPrivateNetwork** PEP cmdlet. For guidance on this cmdlet, please see the [1910 release notes](#).

Azure Stack infrastructure network

This /24 network is dedicated to internal Azure Stack components so that they can communicate and exchange data among themselves. This subnet can be routable externally of the Azure Stack solution to your datacenter, we do not recommend using Public or Internet routable IP addresses on this subnet. This network is advertised to the Border but most of its IPs are protected by Access Control Lists (ACLs). The IPs allowed for access are within a small range equivalent in size to a /27 network and host services like the [privileged end point \(PEP\)](#) and [Azure Stack Backup](#).

Public VIP network

The Public VIP Network is assigned to the network controller in Azure Stack. It's not a logical network on the switch. The SLB uses the pool of addresses and assigns /32 networks for tenant workloads. On the switch routing table, these /32 IPs are advertised as an available route via BGP. This network contains the external-accessible or public IP addresses. The Azure Stack infrastructure reserves the first 31 addresses from this Public VIP Network while the remainder is used by tenant VMs. The network size on this subnet can range from a minimum of /26 (64 hosts) to a maximum of /22 (1022 hosts). We recommend that you plan for a /24 network.

Switch infrastructure network

This /26 network is the subnet that contains the routable point-to-point IP /30 (two host IPs) subnets and the loopbacks, which are dedicated /32 subnets for in-band switch management and BGP router ID. This range of IP addresses must be routable outside the Azure Stack solution to your datacenter. They may be private or public IPs.

Switch management network

This /29 (six host IPs) network is dedicated to connecting the management ports of the switches. It allows out-of-band access for deployment, management, and troubleshooting. It's calculated from the switch infrastructure network mentioned above.

Permitted networks

Starting on 1910, the Deployment Worksheet will have this new field allowing the operator to change some access control list (ACL)s to allow access to network device management interfaces and the hardware lifecycle host (HLH) from a trusted datacenter network range. With the access control list change, the operator can allow their management jumpbox VMs within a specific network range to access the switch management interface, the HLH OS and the HLH BMC. The operator can provide one or multiple subnets to this list, if left blank it will default to deny access. This new functionality replaces the need for post-deployment manual intervention as it used to be described on the [Modify specific settings on your Azure Stack switch configuration](#).

Next steps

Learn about network planning: [Border connectivity](#).

Border connectivity

3 minutes to read • [Edit Online](#)

Network integration planning is an important prerequisite for successful Azure Stack Hub integrated systems deployment, operation, and management. Border connectivity planning begins by choosing if you want use dynamic routing with border gateway protocol (BGP). This requires assigning a 16-bit BGP autonomous system number (public or private) or using static routing, where a static default route is assigned to the border devices.

IMPORTANT

The top of rack (TOR) switches require Layer 3 uplinks with Point-to-Point IPs (/30 networks) configured on the physical interfaces. Layer 2 uplinks with TOR switches supporting Azure Stack Hub operations isn't supported.

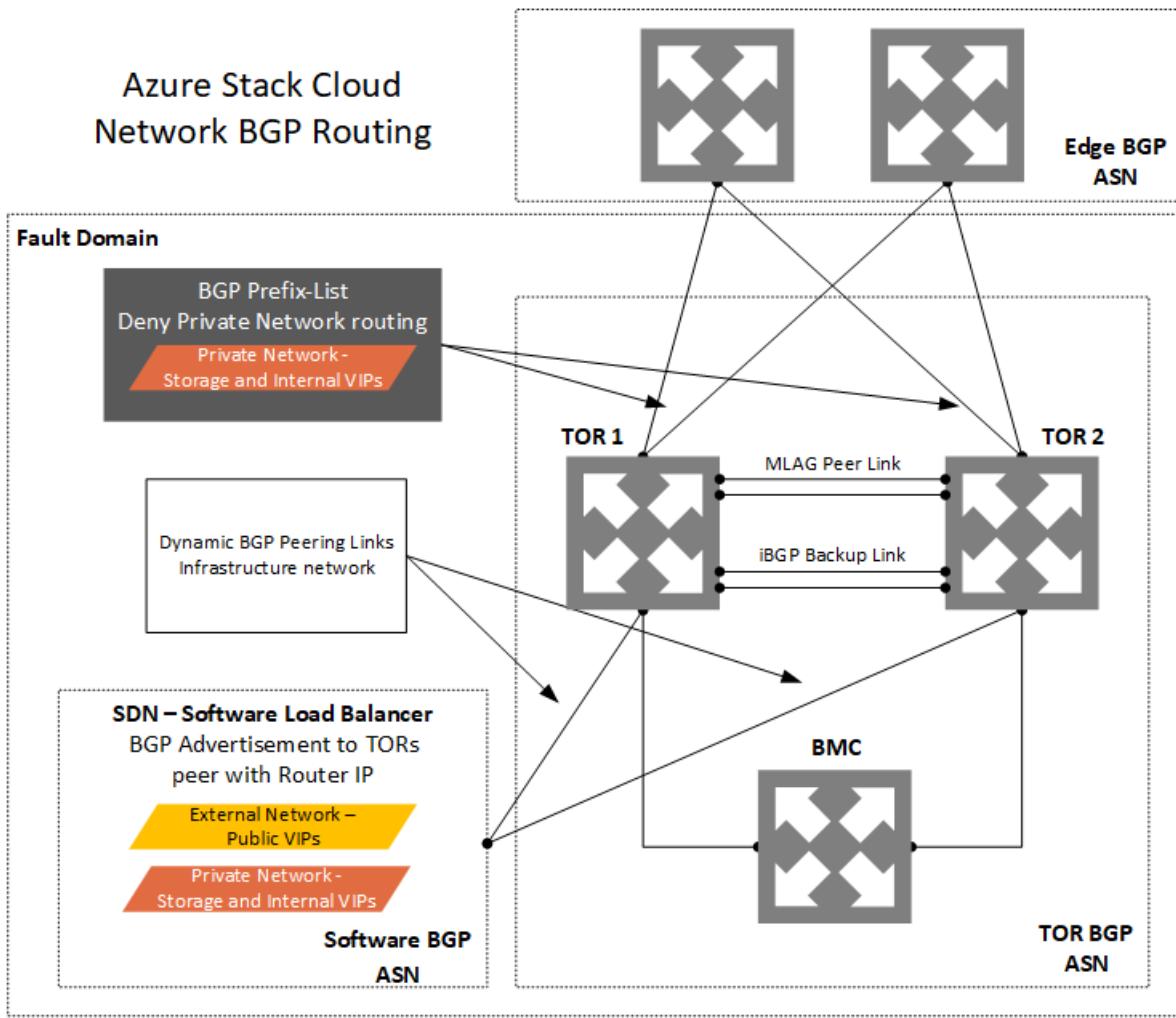
BGP routing

Using a dynamic routing protocol like BGP guarantees that your system is always aware of network changes and facilitates administration. For enhanced security, a password may be set on the BGP peering between the TOR and the Border.

As shown in the following diagram, advertising of the private IP space on the TOR switch is blocked using a prefix-list. The prefix list denies the advertisement of the Private Network and it's applied as a route-map on the connection between the TOR and the border.

The Software Load Balancer (SLB) running inside the Azure Stack Hub solution peers to the TOR devices so it can dynamically advertise the VIP addresses.

To ensure that user traffic immediately and transparently recovers from failure, the VPC or MLAG configured between the TOR devices allows the use of multi-chassis link aggregation to the hosts and HSRP or VRRP that provides network redundancy for the IP networks.



Static routing

Static routing requires additional configuration to the border devices. It requires more manual intervention and management as well as thorough analysis before any change. Issues caused by a configuration error may take more time to rollback depending on the changes made. This routing method isn't recommended, but it's supported.

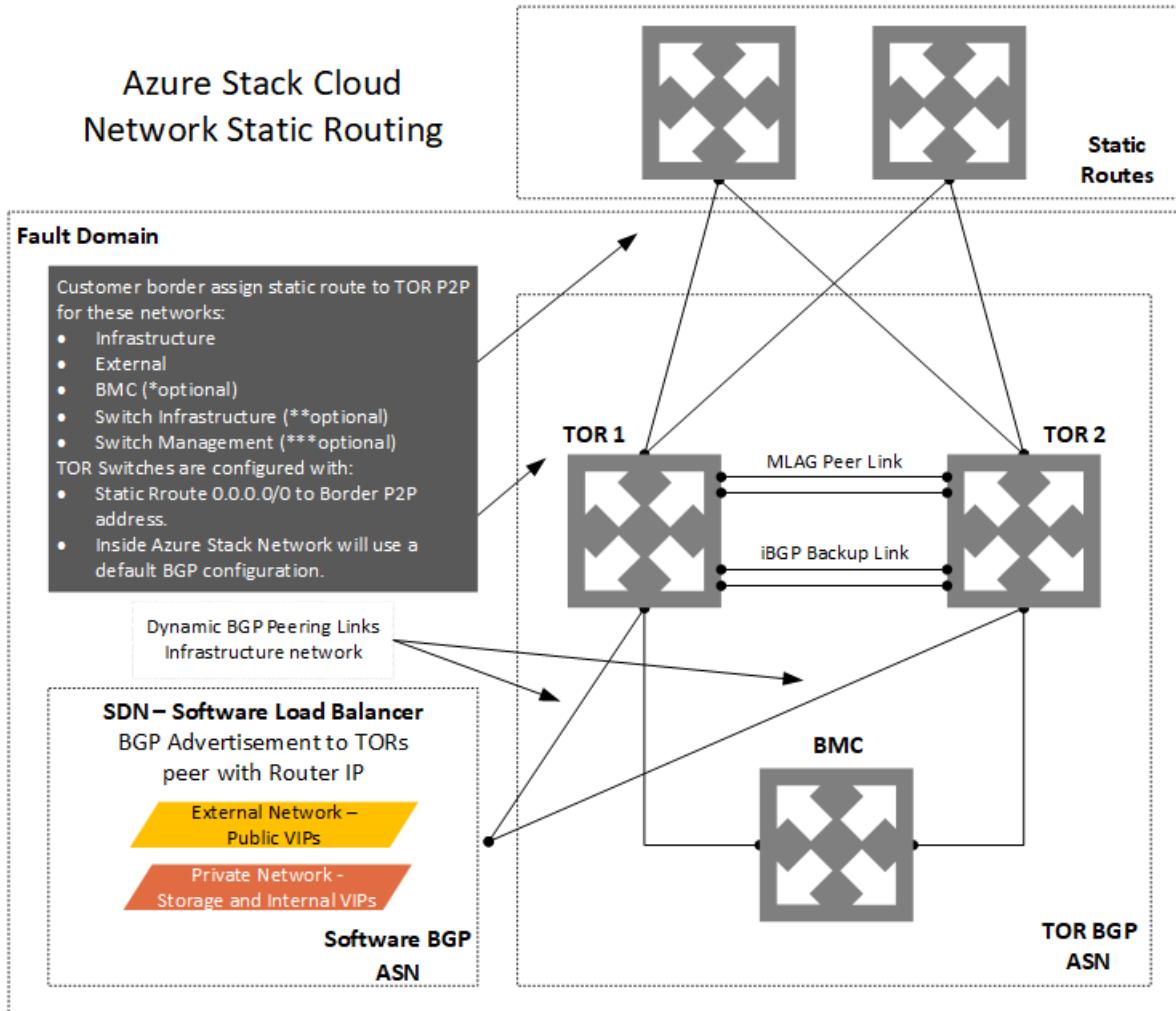
To integrate Azure Stack Hub into your networking environment using static routing, all four physical links between the border and the TOR device must be connected. High availability can't be guaranteed because of how static routing works.

The border device must be configured with static routes pointing to each one of the four P2P IP's set between the TOR and the Border for traffic destined to any network inside Azure Stack Hub, but only the *External* or Public VIP network is required for operation. Static routes to the *BMC* and the *External* networks are required for initial deployment. Operators can choose to leave static routes in the border to access management resources that reside on the *BMC* and the *Infrastructure* network. Adding static routes to *switch infrastructure* and *switch management* networks is optional.

The TOR devices are configured with a static default route sending all traffic to the border devices. The one traffic exception to the default rule is for the private space, which is blocked using an Access Control List applied on the TOR to border connection.

Static routing applies only to the uplinks between the TOR and border switches. BGP dynamic routing is used inside the rack because it's an essential tool for the SLB and other components and can't be disabled or removed.

Azure Stack Cloud Network Static Routing



* The BMC network is optional after deployment.

** The Switch Infrastructure network is optional, as the whole network can be included in the Switch Management network.

*** The Switch Management network is required and can be added separately from the Switch Infrastructure network.

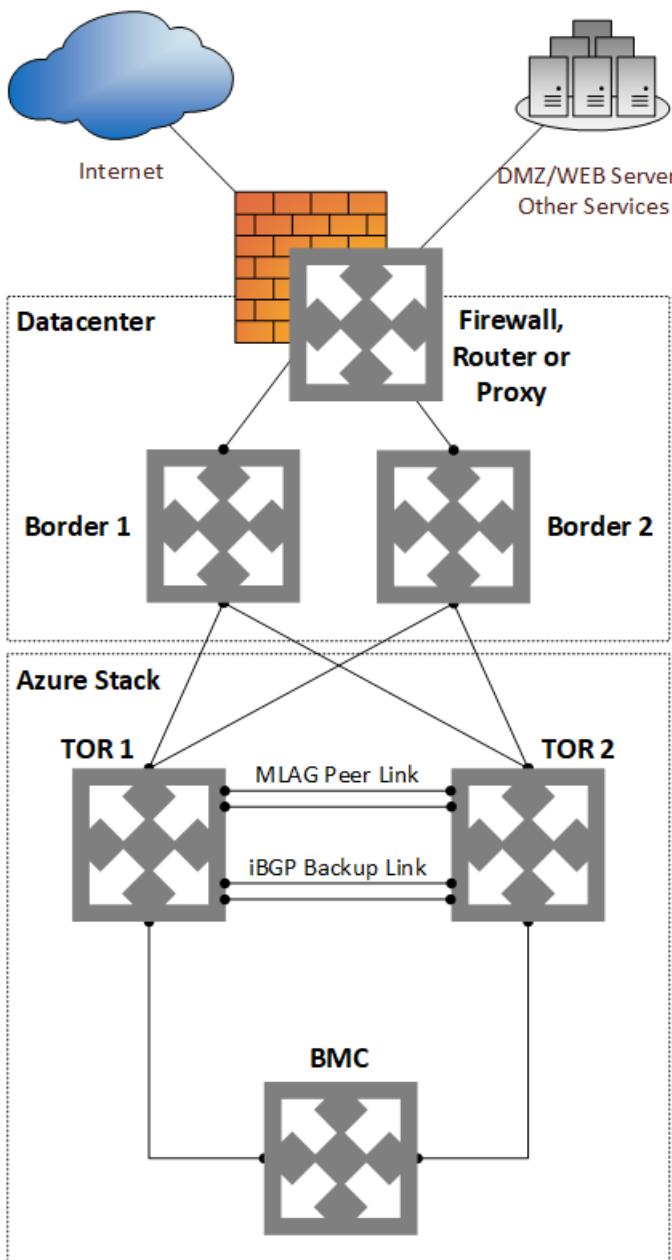
Transparent proxy

If your datacenter requires all traffic to use a proxy, you must configure a *transparent proxy* to process all traffic from the rack to handle it according to policy, separating traffic between the zones on your network.

IMPORTANT

The Azure Stack Hub solution doesn't support normal web proxies.

A transparent proxy (also known as an intercepting, inline, or forced proxy) intercepts normal communication at the network layer without requiring any special client configuration. Clients don't need to be aware of the existence of the proxy.



SSL traffic interception is [not supported](#) and can lead to service failures when accessing endpoints. The maximum supported timeout to communicate with endpoints required for identity is 60s with 3 retry attempts.

Next steps

[DNS integration](#)

Modify specific settings on your Azure Stack Hub switch configuration

3 minutes to read • [Edit Online](#)

You can modify a few environmental settings for your Azure Stack Hub switch configuration. You can identify which of the settings you can change in the template created by your original equipment manufacturer (OEM). This article explains each of those customizable settings, and how the changes can affect your Azure Stack Hub. These settings include password update, syslog server, SNMP monitoring, authentication, and the access control list.

During deployment of the Azure Stack Hub solution, the original equipment manufacturer (OEM) creates and applies the switch configuration for both TORs and BMC. The OEM uses the Azure Stack Hub automation tool to validate that the required configurations are properly set on these devices. The configuration is based the information in your Azure Stack Hub [Deployment Worksheet](#). After the OEM creates the configuration, **do not** alter the configuration without consent from either the OEM or the Microsoft Azure Stack Hub engineering team. A change to the network device configuration can significantly impact the operation or troubleshooting of network issues in your Azure Stack Hub instance.

However, there are some values that can be added, removed, or changed on the configuration of the network switches.

WARNING

Do not alter the configuration without consent from either the OEM or the Microsoft Azure Stack Hub engineering team. A change to the network device configuration can significantly impact the operation or troubleshooting of network issues in your Azure Stack Hub instance.

For more information about these functions on your network device, how to make these changes, please contact your OEM hardware provider or Microsoft support. Your OEM has the configuration file created by the automation tool based on your Azure Stack Hub deployment worksheet.

Password update

The operator may update the password for any user on the network switches at any time. There isn't a requirement to change any information on the Azure Stack Hub system, or to use the steps for [Rotate secrets in Azure Stack Hub](#).

Syslog server

Operators can redirect the switch logs to a syslog server on their datacenter. Use this configuration to ensure that the logs from a particular point in time can be used for troubleshooting. By default, the logs are stored on the switches; their capacity for storing logs is limited. Check the [Access control list updates](#) section for an overview of how to configure the permissions for switch management access.

SNMP monitoring

The operator can configure simple network management protocol (SNMP) v2 or v3 to monitor the network devices and send traps to a network monitoring application on the datacenter. For security reasons, use SNMPv3 since it is more secure than v2. Consult your OEM hardware provider for the MIBs and configuration required. Check the [Access control list updates](#) section for an overview of how to configure the permissions for switch management access.

Authentication

The operator can configure either RADIUS or TACACS to manage authentication on the network devices. Consult your OEM hardware provider for supported methods and configuration required. Check the [Access control list updates](#) section for an overview of how to configure the permissions for Switch Management access.

Access control list updates

NOTE

Starting in 1910, the deployment worksheet will have a new field for **Permitted Networks** which replaces the manual steps required to allow access to network device management interfaces and the hardware lifecycle host (HLH) from a trusted datacenter network range. For more information about this new feature, please check the [Network integration planning for Azure Stack Hub](#).

The operator can change some access control list (ACL)s to allow access to network device management interfaces and the hardware lifecycle host (HLH) from a trusted datacenter network range. With the access control list, The operator can allow their management jumpbox VMs within a specific network range to access the switch management interface, the HLH OS and the HLH BMC.

Next steps

[Azure Stack Hub datacenter integration - DNS](#)

Azure Stack Hub datacenter DNS integration

7 minutes to read • [Edit Online](#)

To be able to access Azure Stack Hub endpoints such as **portal**, **adminportal**, **management**, and **adminmanagement** from outside Azure Stack Hub, you need to integrate the Azure Stack Hub DNS services with the DNS servers that host the DNS zones you want to use in Azure Stack Hub.

Azure Stack Hub DNS namespace

You're required to provide some important information related to DNS when you deploy Azure Stack Hub.

FIELD	DESCRIPTION	EXAMPLE
Region	The geographic location of your Azure Stack Hub deployment.	east
External Domain Name	The name of the zone you want to use for your Azure Stack Hub deployment.	cloud.fabrikam.com
Internal Domain Name	The name of the internal zone that's used for infrastructure services in Azure Stack Hub. It's Directory Service-integrated and private (not reachable from outside the Azure Stack Hub deployment).	azurestack.local
DNS Forwarders	DNS servers that are used to forward DNS queries, DNS zones, and records that are hosted outside Azure Stack Hub, either on the corporate intranet or public internet. You can edit the DNS Forwarder value with the Set-AzSDnsForwarder cmdlet after deployment.	
Naming Prefix (Optional)	The naming prefix you want your Azure Stack Hub infrastructure role instance machine names to have. If not provided, the default is azs.	azs

The fully qualified domain name (FQDN) of your Azure Stack Hub deployment and endpoints is the combination of the Region parameter and the External Domain Name parameter. Using the values from the examples in the previous table, the FQDN for this Azure Stack Hub deployment would be the following name:

east.cloud.fabrikam.com

As such, examples of some of the endpoints for this deployment would look like the following URLs:

<https://portal.east.cloud.fabrikam.com>

<https://adminportal.east.cloud.fabrikam.com>

To use this example DNS namespace for an Azure Stack Hub deployment, the following conditions are required:

- The zone fabrikam.com is registered either with a domain registrar, an internal corporate DNS server, or both,

depending on your name resolution requirements.

- The child domain `cloud.fabrikam.com` exists under the zone `fabrikam.com`.
- The DNS servers that host the zones `fabrikam.com` and `cloud.fabrikam.com` can be reached from the Azure Stack Hub deployment.

To be able to resolve DNS names for Azure Stack Hub endpoints and instances from outside Azure Stack Hub, you need to integrate the DNS servers that host the external DNS zone for Azure Stack Hub with the DNS servers that host the parent zone you want to use.

DNS name labels

Azure Stack Hub supports adding a DNS name label to a public IP address to allow name resolution for public IP addresses. DNS labels are a convenient way for users to reach apps and services hosted in Azure Stack Hub by name. The DNS name label uses a slightly different namespace than the infrastructure endpoints. Following the previous example namespace, the namespace for DNS name labels appears as follows:

`*.east.cloudapp.cloud.fabrikam.com`

Therefore, if a tenant indicates a value **Myapp** in the DNS name label field of a public IP address resource, it creates an A record for **myapp** in the zone **east.cloudapp.cloud.fabrikam.com** on the Azure Stack Hub external DNS server. The resulting fully qualified domain name appears as follows:

`myapp.east.cloudapp.cloud.fabrikam.com`

If you want to leverage this functionality and use this namespace, you must integrate the DNS servers that host the external DNS zone for Azure Stack Hub with the DNS servers that host the parent zone you want to use as well. This is a different namespace than the namespace for the Azure Stack Hub service endpoints, so you must create an additional delegation or conditional forwarding rule.

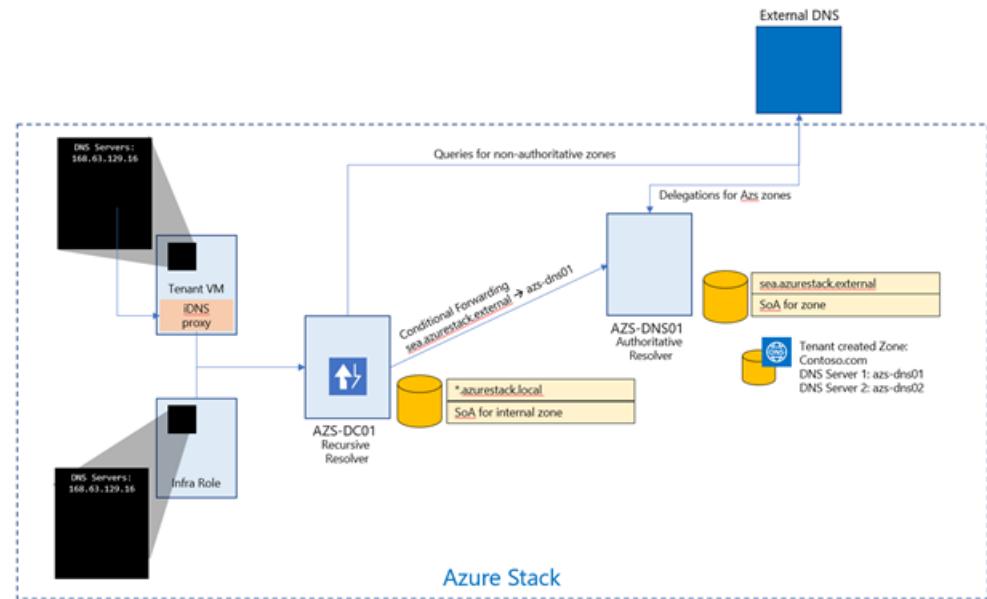
For more information about how the DNS Name label works, see [Using DNS in Azure Stack Hub](#).

Resolution and delegation

There are two types of DNS servers:

- An authoritative DNS server hosts DNS zones. It answers DNS queries for records in those zones only.
- A recursive DNS server doesn't host DNS zones. It answers all DNS queries by calling authoritative DNS servers to gather the data it needs.

Azure Stack Hub includes both authoritative and recursive DNS servers. The recursive servers are used to resolve names of everything except for the internal private zone and the external public DNS zone for that Azure Stack Hub deployment.



Resolving external DNS names from Azure Stack Hub

To resolve DNS names for endpoints outside Azure Stack Hub (for example: www.bing.com), you need to provide DNS servers that Azure Stack Hub can use to forward DNS requests for which Azure Stack Hub isn't authoritative. For deployment, DNS servers that Azure Stack Hub forwards requests to are required in the Deployment Worksheet (in the DNS Forwarder field). Provide at least two servers in this field for fault tolerance. Without these values, Azure Stack Hub deployment fails. You can edit the DNS Forwarder values with the [Set-AzSDnsForwarder cmdlet](#) after deployment.

Configure conditional DNS forwarding

IMPORTANT

This only applies to an AD FS deployment.

To enable name resolution with your existing DNS infrastructure, configure conditional forwarding.

To add a conditional forwarder, you must use the privileged endpoint.

For this procedure, use a computer in your datacenter network that can communicate with the privileged endpoint in Azure Stack Hub.

1. Open an elevated Windows PowerShell session (run as administrator), and connect to the IP address of the privileged endpoint. Use the credentials for CloudAdmin authentication.

```
$cred=Get-Credential
Enter-PSSession -ComputerName <IP Address of ERCS> -ConfigurationName PrivilegedEndpoint -Credential $cred
```

2. After you connect to the privileged endpoint, run the following PowerShell command. Substitute the sample values provided with your domain name and IP addresses of the DNS servers you want to use.

```
Register-CustomDnsServer -CustomDomainName "contoso.com" -CustomDnsIPAddresses "192.168.1.1", "192.168.1.2"
```

Resolving Azure Stack Hub DNS names from outside Azure Stack Hub

The authoritative servers are the ones that hold the external DNS zone information, and any user-created zones. Integrate with these servers to enable zone delegation or conditional forwarding to resolve Azure Stack Hub DNS names from outside Azure Stack Hub.

Get DNS Server external endpoint information

To integrate your Azure Stack Hub deployment with your DNS infrastructure, you need the following information:

- DNS server FQDNs
- DNS server IP addresses

The FQDNs for the Azure Stack Hub DNS servers have the following format:

```
<NAMINGPREFIX>-ns01.<REGION>.<EXTERNALDOMAINNAME>
```

```
<NAMINGPREFIX>-ns02.<REGION>.<EXTERNALDOMAINNAME>
```

Using the sample values, the FQDNs for the DNS servers are:

```
azs-ns01.east.cloud.fabrikam.com
```

```
azs-ns02.east.cloud.fabrikam.com
```

This information is also created at the end of all Azure Stack Hub deployments in a file named `AzureStackStampInformation.json`. This file is located in the `C:\CloudDeployment\logs` folder of the Deployment virtual machine. If you're not sure what values were used for your Azure Stack Hub deployment, you can get the values from here.

If the Deployment virtual machine is no longer available or is inaccessible, you can obtain the values by connecting to the privileged endpoint and running the `Get-AzureStackStampInformation` PowerShell cmdlet. For more information, see [privileged endpoint](#).

Setting up conditional forwarding to Azure Stack Hub

The simplest and most secure way to integrate Azure Stack Hub with your DNS infrastructure is to do conditional forwarding of the zone from the server that hosts the parent zone. This approach is recommended if you have direct control over the DNS servers that host the parent zone for your Azure Stack Hub external DNS namespace.

If you're not familiar with how to do conditional forwarding with DNS, see the following TechNet article: [Assign a Conditional Forwarder for a Domain Name](#), or the documentation specific to your DNS solution.

In scenarios where you specified your external Azure Stack Hub DNS Zone to look like a child domain of your corporate domain name, conditional forwarding can't be used. DNS delegation must be configured.

Example:

- Corporate DNS Domain Name: `contoso.com`
- Azure Stack Hub External DNS Domain Name: `azurestack.contoso.com`

Editing DNS Forwarder IPs

DNS Forwarder IPs are set during deployment of Azure Stack Hub. However, if the Forwarder IPs need to be updated for any reason, you can edit the values by connecting to the privileged endpoint and running the `Get-AzSDnsForwarder` and `Set-AzSDnsForwarder [-IPAddress] <IPAddress[]>` PowerShell cmdlets. For more information, see [privileged endpoint](#).

Delegating the external DNS zone to Azure Stack Hub

For DNS names to be resolvable from outside an Azure Stack Hub deployment, you need to set up DNS delegation.

Each registrar has their own DNS management tools to change the name server records for a domain. In the registrar's DNS management page, edit the NS records and replace the NS records for the zone with the ones in Azure Stack Hub.

Most DNS registrars require you to provide a minimum of two DNS servers to complete the delegation.

Next steps

[Firewall integration](#)

Update the DNS forwarder in Azure Stack Hub

2 minutes to read • [Edit Online](#)

At least one reachable DNS forwarder is necessary for the Azure Stack Hub infrastructure to resolve external names. A DNS forwarder must be provided for the deployment of Azure Stack Hub. That input is used for the Azure Stack Hub internal DNS servers as forwarder, and enables external name resolution for services like authentication, marketplace management, or usage.

DNS is a critical datacenter infrastructure service that can change, and if it does, Azure Stack Hub must be updated.

This article describes using the privileged endpoint (PEP) to update the DNS forwarder in Azure Stack Hub. It is recommended that you use two reliable DNS forwarder IP addresses.

1. Connect to the [privileged endpoint](#). Note that it is not necessary to unlock the privileged endpoint by opening a support ticket.
2. Run the following command to review the current configured DNS forwarder. As an alternative, you can also use the admin portal region properties:

```
Get-AzsDnsForwarder
```

3. Run the following command to update Azure Stack Hub to use the new DNS forwarder:

```
Set-AzsDnsForwarder -IPAddress "IPAddress 1","IPAddress 2"
```

4. Review the output of the command for any errors.

Next steps

[Firewall integration](#)

Configure the time server for Azure Stack Hub

2 minutes to read • [Edit Online](#)

You can use the privileged endpoint (PEP) to update the time server in Azure Stack Hub. Use a host name that resolves to two or more NTP server IP addresses.

Azure Stack Hub uses the Network Time Protocol (NTP) to connect to time servers on the Internet. NTP servers provide accurate system time. Time is used across Azure Stack Hub's physical network switches, hardware lifecycle host, infrastructure service, and virtual machines. If the clock isn't synchronized, Azure Stack Hub may experience severe issues with the network and authentication. Log files, documents, and other files may be created with incorrect timestamps.

Providing one time server (NTP) is required for Azure Stack Hub to synchronize time. When you deploy Azure Stack Hub, you provide the address of an NTP server. Time is a critical datacenter infrastructure service. If the service changes, you will need to update the time.

NOTE

Azure Stack Hub supports synchronizing time with only one time server (NTP). You cannot provide multiple NTPs for Azure Stack Hub to synchronize time with.

Configure time

1. Connect to the PEP.

NOTE

It isn't necessary to unlock the privileged endpoint by opening a support ticket.

2. Run the following command to review the current configured NTP server:

```
Get-AzsTimeSource
```

3. Run the following command to update Azure Stack Hub to use the new NTP Server and to immediately synchronize the time.

NOTE

This procedure doesn't update the time server on the physical switches. If your time server is not a Windows-based NTP server, you need to add the flag `0x8`.

```
Set-AzsTimeSource -TimeServer NEWTIMESERVERIP -resync
```

For servers other than Windows-based time servers:

```
Set-AzsTimeSource -TimeServer "NEWTIMESERVERIP,0x8" -resync
```

4. Review the output of the command for any errors.

Next steps

[View the readiness report](#)

[General Azure Stack Hub integration considerations](#)

Azure Stack Hub firewall integration

3 minutes to read • [Edit Online](#)

It's recommended that you use a firewall device to help secure Azure Stack Hub. Firewalls can help defend against things like distributed denial-of-service (DDOS) attacks, intrusion detection, and content inspection. However, they can also become a throughput bottleneck for Azure storage services like blobs, tables, and queues.

If a disconnected deployment mode is used, you must publish the AD FS endpoint. For more information, see the [datacenter integration identity article](#).

The Azure Resource Manager (administrator), administrator portal, and Key Vault (administrator) endpoints don't necessarily require external publishing. For example, as a service provider, you could limit the attack surface by only administering Azure Stack Hub from inside your network, and not from the internet.

For enterprise organizations, the external network can be the existing corporate network. In this scenario, you must publish endpoints to operate Azure Stack Hub from the corporate network.

Network Address Translation

Network Address Translation (NAT) is the recommended method to allow the deployment virtual machine (DVM) to access external resources and the internet during deployment as well as the Emergency Recovery Console (ERCS) VMs or privileged endpoint (PEP) during registration and troubleshooting.

NAT can also be an alternative to Public IP addresses on the external network or public VIPs. However, it's not recommended to do so because it limits the tenant user experience and increases complexity. One option would be a one to one NAT that still requires one public IP per user IP on the pool. Another option is a many to one NAT that requires a NAT rule per user VIP for all ports a user might use.

Some of the downsides of using NAT for Public VIP are:

- NAT adds overhead when managing firewall rules because users control their own endpoints and their own publishing rules in the software-defined networking (SDN) stack. Users must contact the Azure Stack Hub operator to get their VIPs published, and to update the port list.
- While NAT usage limits the user experience, it gives full control to the operator over publishing requests.
- For hybrid cloud scenarios with Azure, consider that Azure doesn't support setting up a VPN tunnel to an endpoint using NAT.

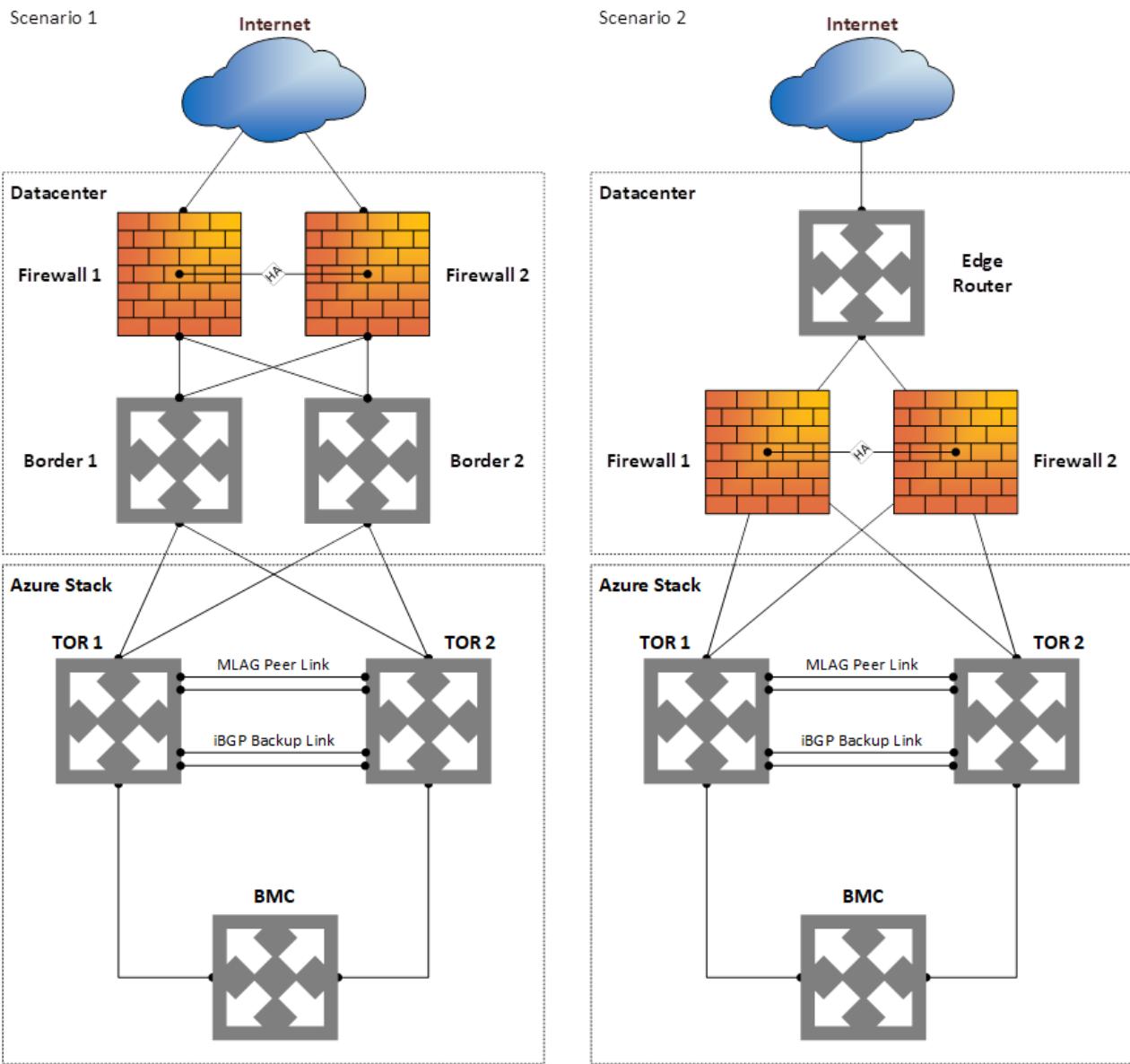
SSL interception

It's currently recommended to disable any SSL interception (for example decryption offloading) on all Azure Stack Hub traffic. If it's supported in future updates, guidance will be provided about how to enable SSL interception for Azure Stack Hub.

Edge firewall scenario

In an edge deployment, Azure Stack Hub is deployed directly behind the edge router or the firewall. In these scenarios, it's supported for the firewall to be above the border (Scenario 1) where it supports both active-active and active-passive firewall configurations or acting as the border device (Scenario 2) where it only supports active-active firewall configuration relying on equal-cost multi-path (ECMP) with either BGP or static routing for failover.

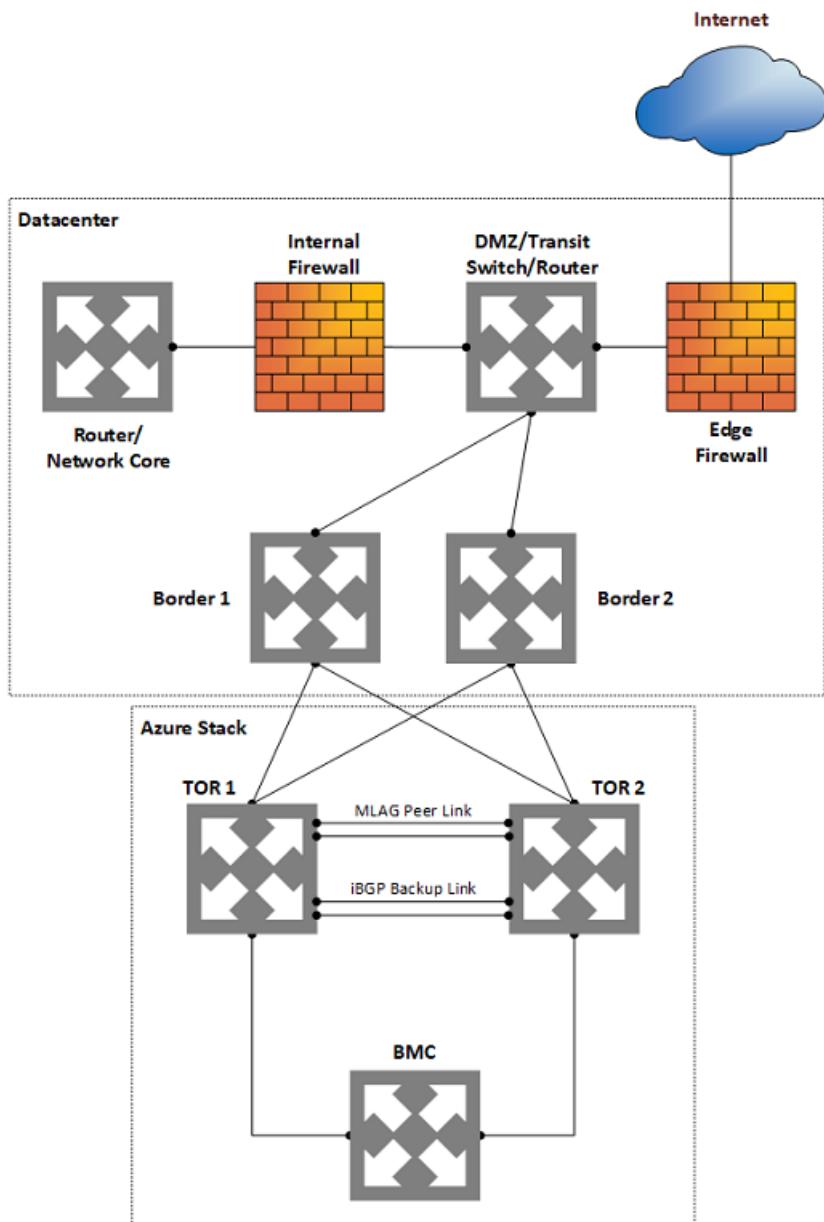
Public routable IP addresses are specified for the public VIP pool from the external network at deployment time. In an edge scenario, it's not recommended to use public routable IPs on any other network for security purposes. This scenario enables a user to experience the full self-controlled cloud experience as in a public cloud like Azure.



Enterprise intranet or perimeter network firewall scenario

In an enterprise intranet or perimeter deployment, Azure Stack Hub is deployed on a multi-zoned firewall or in between the edge firewall and the internal, corporate network firewall. Its traffic is then distributed between the secure, perimeter network (or DMZ), and unsecure zones as described below:

- **Secure zone:** This is the internal network that uses internal or corporate routable IP addresses. The secure network can be divided, have internet outbound access through NAT on the Firewall, and is usually accessible from anywhere inside your datacenter via the internal network. All Azure Stack Hub networks should reside in the secure zone except for the external network's public VIP pool.
- **Perimeter zone.** The perimeter network is where external or internet-facing apps like Web servers are typically deployed. It's usually monitored by a firewall to avoid attacks like DDoS and intrusion (hacking) while still allowing specified inbound traffic from the internet. Only the external network public VIP pool of Azure Stack Hub should reside in the DMZ zone.
- **Unsecure zone.** This is the external network, the internet. It **is not** recommended to deploy Azure Stack Hub in the unsecure zone.



Learn more

Learn more about [ports and protocols used by Azure Stack Hub endpoints](#).

Next steps

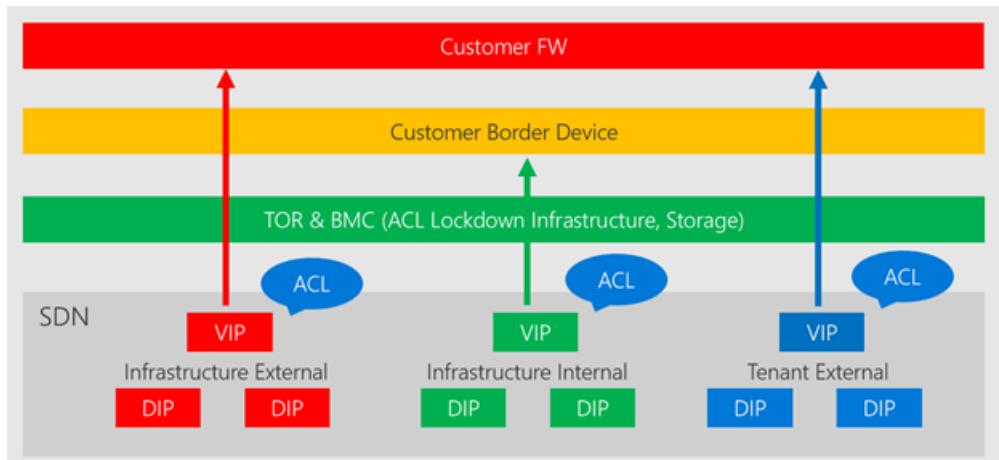
[Azure Stack Hub PKI requirements](#)

Publish Azure Stack Hub services in your datacenter

4 minutes to read • [Edit Online](#)

Azure Stack Hub sets up virtual IP addresses (VIPs) for its infrastructure roles. These VIPs are allocated from the public IP address pool. Each VIP is secured with an access control list (ACL) in the software-defined network layer. ACLs are also used across the physical switches (TORs and BMC) to further harden the solution. A DNS entry is created for each endpoint in the external DNS zone that's specified at deployment time. For example, the user portal is assigned the DNS host entry of portal.<region>.<fqdn>.

The following architectural diagram shows the different network layers and ACLs:



Ports and URLs

To make Azure Stack Hub services (like the portals, Azure Resource Manager, DNS, and so on) available to external networks, you must allow inbound traffic to these endpoints for specific URLs, ports, and protocols.

In a deployment where a transparent proxy uplinks to a traditional proxy server or a firewall is protecting the solution, you must allow specific ports and URLs for both **inbound** and **outbound** communication. These include ports and URLs for identity, the marketplace, patch and update, registration, and usage data.

SSL traffic interception is [not supported](#) and can lead to service failures when accessing endpoints.

Ports and protocols (inbound)

A set of infrastructure VIPs is required for publishing Azure Stack Hub endpoints to external networks. The *Endpoint (VIP)* table shows each endpoint, the required port, and protocol. Refer to the specific resource provider deployment documentation for endpoints that require additional resource providers, like the SQL resource provider.

Internal infrastructure VIPs aren't listed because they're not required for publishing Azure Stack Hub. User VIPs are dynamic and defined by the users themselves, with no control by the Azure Stack Hub operator.

NOTE

IKEv2 VPN is a standards-based IPsec VPN solution that uses UDP port 500 and 4500 and TCP port 50. Firewalls don't always open these ports, so an IKEv2 VPN might not be able to traverse proxies and firewalls.

With the addition of the [Extension Host](#), ports in the range of 12495-30015 aren't required.

ENDPOINT (VIP)	DNS HOST A RECORD	PROTOCOL	PORTS
AD FS	Adfs.<region>.<fqdn>	HTTPS	443
Portal (administrator)	Adminportal.<region>.<fqdn>	HTTPS	443
Adminhosting	*.adminhosting.<region>.<fqdn>	HTTPS	443
Azure Resource Manager (administrator)	Adminmanagement.<region>.<fqdn>	HTTPS	443
Portal (user)	Portal.<region>.<fqdn>	HTTPS	443
Azure Resource Manager (user)	Management.<region>.<fqdn>	HTTPS	443
Graph	Graph.<region>.<fqdn>	HTTPS	443
Certificate revocation list	Crl.<region>.<fqdn>	HTTP	80
DNS	*.<region>.<fqdn>	TCP & UDP	53
Hosting	*.hosting.<region>.<fqdn>	HTTPS	443
Key Vault (user)	*.vault.<region>.<fqdn>	HTTPS	443
Key Vault (administrator)	*.adminvault.<region>.<fqdn>	HTTPS	443
Storage Queue	*.queue.<region>.<fqdn>	HTTP HTTPS	80 443
Storage Table	*.table.<region>.<fqdn>	HTTP HTTPS	80 443
Storage Blob	*.blob.<region>.<fqdn>	HTTP HTTPS	80 443
SQL Resource Provider	sqladapter:dbadapter.<region>.<fqdn>	HTTPS	44300-44304
MySQL Resource Provider	mysqladapter:dbadapter.<region>.<fqdn>	HTTPS	44300-44304
App Service	*.appservice.<region>.<fqdn>	TCP	80 (HTTP) 443 (HTTPS) 8172 (MSDeploy)
	*.scm.appservice.<region>.<fqdn>	TCP	443 (HTTPS)

ENDPOINT (VIP)	DNS HOST A RECORD	PROTOCOL	PORTS
	api.appservice.<region>. <fqdn>	TCP	443 (HTTPS) 44300 (Azure Resource Manager)
	ftp.appservice.<region>. <fqdn>	TCP, UDP	21, 1021, 10001-10100 (FTP) 990 (FTPS)
VPN Gateways			See the VPN gateway FAQ.

Ports and URLs (outbound)

Azure Stack Hub supports only transparent proxy servers. In a deployment with a transparent proxy uplink to a traditional proxy server, you must allow the ports and URLs in the following table for outbound communication.

SSL traffic interception is [not supported](#) and can lead to service failures when accessing endpoints. The maximum supported timeout to communicate with endpoints required for identity is 60s.

NOTE

Azure Stack Hub doesn't support using ExpressRoute to reach the Azure services listed in the following table because ExpressRoute may not be able to route traffic to all of the endpoints.

PURPOSE	DESTINATION URL	PROTOCOL	PORTS	SOURCE NETWORK

PURPOSE	DESTINATION URL	PROTOCOL	PORTS	SOURCE NETWORK
Identity	Azure login.windows.net login.microsoftonline.com graph.windows.net https://secure.aadcdn.microsoftonline-p.com www.office.com ManagementServiceUri = https://management.core.windows.net ARMUri = https://management.azure.com https://*.msftauth.net https://*.msauth.net https://*.msocdn.com Azure Government https://login.microsoftonline.us/ https://graph.windows.net/ Azure China 21Vianet https://login.chinacloudapi.cn/ https://graph.chinacloudapi.cn/ Azure Germany https://login.microsoftonline.de/ https://graph.cloudapi.de/	HTTP HTTPS	80 443	Public VIP - /27 Public infrastructure Network
Marketplace syndication	Azure https://management.azure.com https://*.blob.core.windows.net https://*.azurededge.net Azure Government https://management.usgovcloudapi.net/ https://*.blob.core.usgovcloudapi.net/ Azure China 21Vianet https://management.chinacloudapi.cn/ http://*.blob.core.chinacloudapi.cn	HTTPS	443	Public VIP - /27
Patch & Update	https://*.azurededge.net https://aka.ms/azuresackautomaticupdate	HTTPS	443	Public VIP - /27

PURPOSE	DESTINATION URL	PROTOCOL	PORTS	SOURCE NETWORK
Registration	Azure https://management.azure.com Azure Government https://management.usgovcloudapi.net/ Azure China 21Vianet https://management.chinacloudapi.cn	HTTPS	443	Public VIP - /27
Usage	Azure https://*.trafficmanager.net Azure Government https://*.usgovtrafficmanager.net Azure China 21Vianet https://*.trafficmanager.cn	HTTPS	443	Public VIP - /27
Windows Defender	*.wdcp.microsoft.com *.wdcpalt.microsoft.com *.wd.microsoft.com *.update.microsoft.com *.download.microsoft.com https://www.microsoft.com/pkiops/crl https://www.microsoft.com/pkiops/certs https://crl.microsoft.com/pki/crl/products https://www.microsoft.com/pki/certs https://secure.aadcdn.microsoftonline-p.com	HTTPS	80 443	Public VIP - /27 Public infrastructure Network
NTP	(IP of NTP server provided for deployment)	UDP	123	Public VIP - /27
DNS	(IP of DNS server provided for deployment)	TCP UDP	53	Public VIP - /27
CRL	(URL under CRL Distribution Points on your certificate)	HTTP	80	Public VIP - /27
LDAP	Active Directory Forest provided for Graph integration	TCP UDP	389	Public VIP - /27

PURPOSE	DESTINATION URL	PROTOCOL	PORTS	SOURCE NETWORK
LDAP SSL	Active Directory Forest provided for Graph integration	TCP	636	Public VIP - /27
LDAP GC	Active Directory Forest provided for Graph integration	TCP	3268	Public VIP - /27
LDAP GC SSL	Active Directory Forest provided for Graph integration	TCP	3269	Public VIP - /27
AD FS	AD FS metadata endpoint provided for AD FS integration	TCP	443	Public VIP - /27
Diagnostic Log collection service	Azure Storage provided Blob SAS URL	HTTPS	443	Public VIP - /27

Outbound URLs are load balanced using Azure traffic manager to provide the best possible connectivity based on geographic location. With load balanced URLs, Microsoft can update and change backend endpoints without affecting customers. Microsoft doesn't share the list of IP addresses for the load balanced URLs. Use a device that supports filtering by URL rather than by IP.

Outbound DNS is required at all times; what varies is the source querying the external DNS and what type of identity integration was chosen. During deployment for a connected scenario, the DVM that sits on the BMC network needs outbound access. But after deployment, the DNS service moves to an internal component that will send queries through a Public VIP. At that time, the outbound DNS access through the BMC network can be removed, but the Public VIP access to that DNS server must remain or else authentication will fail.

Next steps

[Azure Stack Hub PKI requirements](#)

Prepare for extension host in Azure Stack Hub

5 minutes to read • [Edit Online](#)

The extension host secures Azure Stack Hub by reducing the number of required TCP/IP ports. This article looks at preparing Azure Stack Hub for the extension host that is automatically enabled through an Azure Stack Hub update package after the 1808 update. This article applies to Azure Stack Hub updates 1808, 1809, and 1811.

Certificate requirements

The extension host implements two new domain namespaces to guarantee unique host entries for each portal extension. The new domain namespaces require two additional wildcard certificates to ensure secure communication.

The table shows the new namespaces and the associated certificates:

DEPLOYMENT FOLDER	REQUIRED CERTIFICATE SUBJECT AND SUBJECT ALTERNATIVE NAMES (SAN)	SCOPE (PER REGION)	SUBDOMAIN NAMESPACE
Admin extension host	*.adminhosting.<region>. <fqdn> (Wildcard SSL Certificates)	Admin extension host	adminhosting.<region>. <fqdn>
Public extension host	*.hosting.<region>. <fqdn> (Wildcard SSL Certificates)	Public extension host	hosting.<region>. <fqdn>

For detailed certificate requirements, see [Azure Stack Hub public key infrastructure certificate requirements](#).

Create certificate signing request

The Azure Stack Hub Readiness Checker tool lets you create a certificate signing request for the two new and required SSL certificates. Follow the steps in the article [Azure Stack Hub certificates signing request generation](#).

NOTE

You may skip this step depending on how you requested your SSL certificates.

Validate new certificates

1. Open PowerShell with elevated permission on the hardware lifecycle host or the Azure Stack Hub management workstation.
2. Run the following cmdlet to install the Azure Stack Hub Readiness Checker tool:

```
Install-Module -Name Microsoft.AzureStack.ReadinessChecker
```

3. Run the following script to create the required folder structure:

```

New-Item C:\Certificates -ItemType Directory

$directories = 'ACSBlob','ACSQueue','ACSTable','Admin Portal','ARM Admin','ARM
Public','KeyVault','KeyVaultInternal','Public Portal', 'Admin extension host', 'Public extension host'

$destination = 'c:\certificates'

$directories | % { New-Item -Path (Join-Path $destination $PSITEM) -ItemType Directory -Force}

```

NOTE

If you deploy with Azure Active Directory Federated Services (AD FS) the following directories must be added to **\$directories** in the script: **ADFS** , **Graph** .

4. Place the existing certificates, which you're currently using in Azure Stack Hub, in appropriate directories. For example, put the **Admin ARM** certificate in the **Arm Admin** folder. And then put the newly created hosting certificates in the **Admin extension host** and **Public extension host** directories.
5. Run the following cmdlet to start the certificate check:

```

$pfxPassword = Read-Host -Prompt "Enter PFX Password" -AsSecureString

Start-AzsReadinessChecker -CertificatePath c:\certificates -pfxPassword $pfxPassword -RegionName east -
FQDN azurestack.contoso.com -IdentitySystem AAD

```

6. Check the output and if all certificates pass all tests.

Import extension host certificates

Use a computer that can connect to the Azure Stack Hub privileged endpoint for the next steps. Make sure you have access to the new certificate files from that computer.

1. Use a computer that can connect to the Azure Stack Hub privileged endpoint for the next steps. Make sure you access to the new certificate files from that computer.
2. Open PowerShell ISE to execute the next script blocks.
3. Import the certificate for the admin hosting endpoint.

```

$CertPassword = read-host -AsSecureString -prompt "Certificate Password"

$CloudAdminCred = Get-Credential -UserName <Privileged endpoint credentials> -Message "Enter the cloud
domain credentials to access the privileged endpoint."

[Byte[]]$AdminHostingCertContent = [Byte[]](Get-Content c:\certificate\myadminhostingcertificate.pfx -
Encoding Byte)

Invoke-Command -ComputerName <PrivilegedEndpoint computer name> ` 
-Credential $CloudAdminCred ` 
-ConfigurationName "PrivilegedEndpoint" ` 
-ArgumentList @($AdminHostingCertContent, $CertPassword) ` 
-ScriptBlock {
    param($AdminHostingCertContent, $CertPassword)
    Import-AdminHostingServiceCert $AdminHostingCertContent $certPassword
}

```

4. Import the certificate for the hosting endpoint.

```

$CertPassword = read-host -AsSecureString -prompt "Certificate Password"

$CloudAdminCred = Get-Credential -UserName <Privileged endpoint credentials> -Message "Enter the cloud
domain credentials to access the privileged endpoint."

[Byte[]]$HostingCertContent = [Byte[]](Get-Content c:\certificate\myhostingcertificate.pfx -Encoding
Byte)

Invoke-Command -ComputerName <PrivilegedEndpoint computer name> ` 
-Credential $CloudAdminCred ` 
-ConfigurationName "PrivilegedEndpoint" ` 
-ArgumentList @($HostingCertContent, $CertPassword) ` 
-ScriptBlock {
    param($HostingCertContent, $CertPassword)
    Import-UserHostingServiceCert $HostingCertContent $certPassword
}

```

Update DNS configuration

NOTE

This step isn't required if you used DNS Zone delegation for DNS Integration. If individual host A records have been configured to publish Azure Stack Hub endpoints, you need to create two additional host A records:

IP	HOSTNAME	TYPE
<IP>	*.Adminhosting.<Region>.<FQDN>	A
<IP>	*.Hosting.<Region>.<FQDN>	A

Allocated IPs can be retrieved using the privileged endpoint by running the cmdlet **Get-AzureStackStampInformation**.

Ports and protocols

The article [Azure Stack Hub datacenter integration - Publish endpoints](#) covers the ports and protocols that require inbound communication to publish Azure Stack Hub before the extension host rollout.

Publish new endpoints

There are two new endpoints required to be published through your firewall. The allocated IPs from the public VIP pool can be retrieved using the following code that must be run from your Azure Stack Hub [environment's privileged endpoint](#).

```

# Create a PEP Session
winrm s winrm/config/client '@{TrustedHosts= "<IpOfERCSMachine>"}'
$PEPCreds = Get-Credential
$PEPSession = New-PSSession -ComputerName <IpOfERCSMachine> -Credential $PEPCreds -ConfigurationName "PrivilegedEndpoint"

# Obtain DNS Servers and extension host information from Azure Stack Hub Stamp Information and find the IPs for the Host Extension Endpoints
$StampInformation = Invoke-Command $PEPSession {Get-AzureStackStampInformation} | Select-Object -Property ExternalDNSIPAddress01, ExternalDNSIPAddress02, @{n="TenantHosting";e= {($_.TenantExternalEndpoints.TenantHosting) -replace "https://*.", "testdnsentry"-replace "/"}}, @{n="AdminHosting";e= {($_.AdminExternalEndpoints.AdminHosting)-replace "https://*.", "testdnsentry"-replace "/"}}, @{n="TenantHostingDNS";e= {($_.TenantExternalEndpoints.TenantHosting) -replace "https://", ""-replace "/"}}, @{n="AdminHostingDNS";e= {($_.AdminExternalEndpoints.AdminHosting)-replace "https://", ""-replace "/"}}
If (Resolve-DnsName -Server $StampInformation.ExternalDNSIPAddress01 -Name $StampInformation.TenantHosting -ErrorAction SilentlyContinue) {
    Write-Host "Can access AZS DNS" -ForegroundColor Green
    $AdminIP = (Resolve-DnsName -Server $StampInformation.ExternalDNSIPAddress02 -Name $StampInformation.AdminHosting).IPAddress
    Write-Host "The IP for the Admin Extension Host is: $($StampInformation.AdminHostingDNS) - is: $($AdminIP)" -ForegroundColor Yellow
    Write-Host "The Record to be added in the DNS zone: Type A, Name: $($StampInformation.AdminHostingDNS), Value: $($AdminIP)" -ForegroundColor Green
    $TenantIP = (Resolve-DnsName -Server $StampInformation.ExternalDNSIPAddress01 -Name $StampInformation.TenantHosting).IPAddress
    Write-Host "The IP address for the Tenant Extension Host is $($StampInformation.TenantHostingDNS) - is: $($TenantIP)" -ForegroundColor Yellow
    Write-Host "The Record to be added in the DNS zone: Type A, Name: $($StampInformation.TenantHostingDNS), Value: $($TenantIP)" -ForegroundColor Green
}
Else {
    Write-Host "Cannot access AZS DNS" -ForegroundColor Yellow
    $AdminIP = (Resolve-DnsName -Name $StampInformation.AdminHosting).IPAddress
    Write-Host "The IP for the Admin Extension Host is: $($StampInformation.AdminHostingDNS) - is: $($AdminIP)" -ForegroundColor Yellow
    Write-Host "The Record to be added in the DNS zone: Type A, Name: $($StampInformation.AdminHostingDNS), Value: $($AdminIP)" -ForegroundColor Green
    $TenantIP = (Resolve-DnsName -Name $StampInformation.TenantHosting).IPAddress
    Write-Host "The IP address for the Tenant Extension Host is $($StampInformation.TenantHostingDNS) - is: $($TenantIP)" -ForegroundColor Yellow
    Write-Host "The Record to be added in the DNS zone: Type A, Name: $($StampInformation.TenantHostingDNS), Value: $($TenantIP)" -ForegroundColor Green
}
Remove-PSSession -Session $PEPSession

```

Sample Output

```

Can access AZS DNS
The IP for the Admin Extension Host is: *.adminhosting.\<region>.\<fqdn> - is: xxx.xxx.xxx.xxx
The Record to be added in the DNS zone: Type A, Name: *.adminhosting.\<region>.\<fqdn>, Value: xxx.xxx.xxx.xxx
The IP address for the Tenant Extension Host is *.hosting.\<region>.\<fqdn> - is: xxx.xxx.xxx.xxx
The Record to be added in the DNS zone: Type A, Name: *.hosting.\<region>.\<fqdn>, Value: xxx.xxx.xxx.xxx

```

NOTE

Make this change before enabling the extension host. This allows the Azure Stack Hub portals to be continuously accessible.

ENDPOINT (VIP)	PROTOCOL	PORTS
Admin Hosting	HTTPS	443

ENDPOINT (VIP)	PROTOCOL	PORTS
Hosting	HTTPS	443

Update existing publishing Rules (Post enablement of extension host)

NOTE

The 1808 Azure Stack Hub Update Package does **not** enable extension host yet. It lets you prepare for extension host by importing the required certificates. Don't close any ports before extension host is automatically enabled through an Azure Stack Hub update package after the 1808 update.

The following existing endpoint ports must be closed in your existing firewall rules.

NOTE

It's recommended to close those ports after successful validation.

ENDPOINT (VIP)	PROTOCOL	PORTS
Portal (administrator)	HTTPS	12495 12499 12646 12647 12648 12649 12650 13001 13003 13010 13011 13012 13020 13021 13026 30015
Portal (user)	HTTPS	12495 12649 13001 13010 13011 13012 13020 13021 30015 13003
Azure Resource Manager (administrator)	HTTPS	30024
Azure Resource Manager (user)	HTTPS	30024

Next steps

- Learn about [Firewall integration](#).

- Learn about [Azure Stack Hub certificates signing request generation](#).

Deployment worksheet for Azure Stack Hub integrated systems

3 minutes to read • [Edit Online](#)

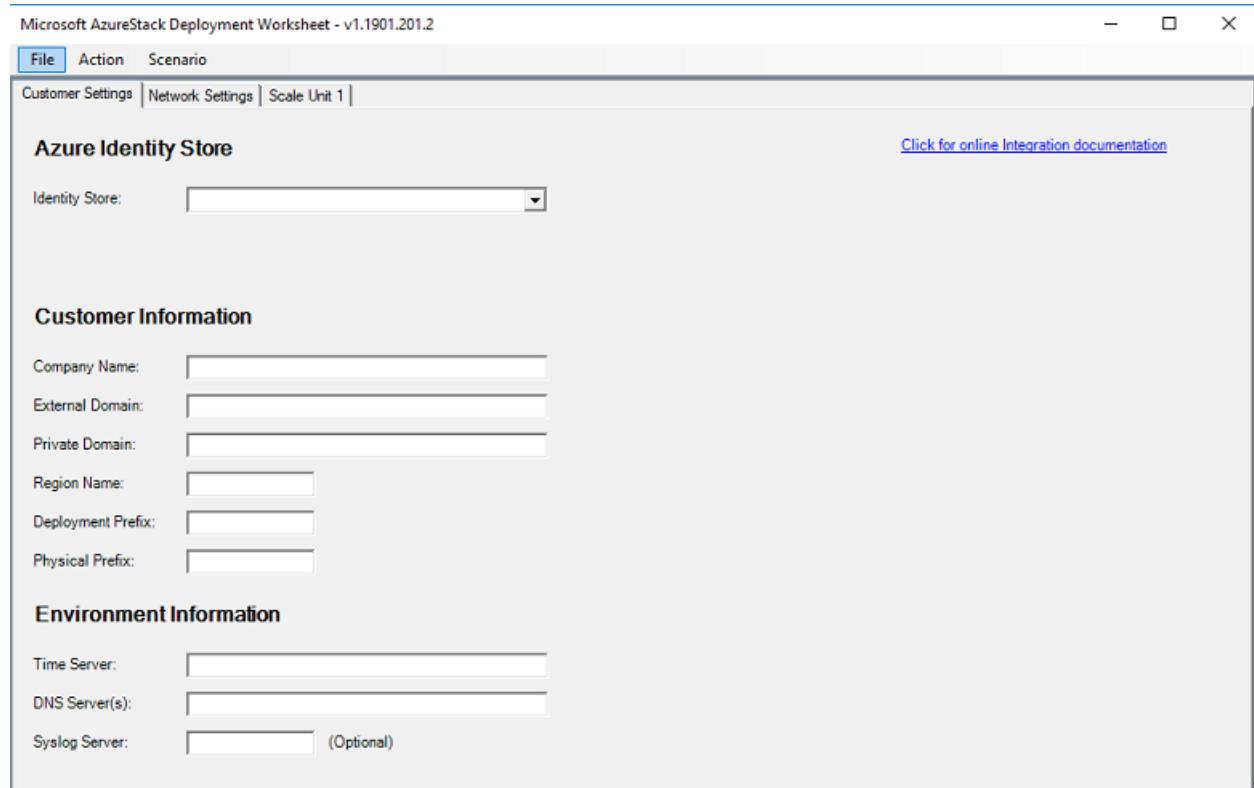
The Azure Stack Hub deployment worksheet is a Windows Forms application that aggregates all necessary deployment information and decisions in one place. You can complete the deployment worksheet during the planning process, and review it before the deployment starts.

The information required by the worksheet covers networking, security, and identity information. It requires important decisions that may need knowledge in many different areas; therefore, you might want to consult with teams possessing expertise in these areas in order to complete the worksheet.

While filling out the worksheet, you might need to make some pre-deployment configuration changes to your network environment. This can include reserving IP address spaces for the Azure Stack Hub solution, and configuring routers, switches, and firewalls, to prepare for connectivity to the new Azure Stack Hub solution.

NOTE

For more information about how to complete the deployment worksheet tool, see [this article in the Azure Stack Hub documentation](#).



Installing the Windows PowerShell module

For each release of the deployment worksheet, you must perform a one-time installation of a Powershell module for each machine on which you want to use the deployment worksheet.

NOTE

The computer must be connected to the internet for this method to work.

1. Open an elevated PowerShell prompt.
2. In the PowerShell window, install the module from the [PowerShell gallery](#):

```
Install-Module -Name Azs.Deployment.Worksheet -Repository PSGallery
```

If you receive a message about installing from an untrusted repository, press **Y** to continue installation.

Use the deployment worksheet tool

To launch and use the deployment worksheet on a computer on which you have installed the deployment worksheet PowerShell module, perform the following steps:

1. Start Windows PowerShell (do not use the PowerShell ISE, as unexpected results can occur). It is not necessary to run PowerShell as an Administrator.
2. Import the **AzS.Deployment.Worksheet** PowerShell module:

```
Import-Module AzS.Deployment.Worksheet
```

3. Once the module is imported, launch the deployment worksheet:

```
Start-DeploymentWorksheet
```

The deployment worksheet consists of separate tabs for collecting environment settings, such as **Customer Settings**, **Network Settings**, and **Scale Unit #**. You must supply all values (except for any marked **Optional**) on all tabs before any configuration data files can be generated. After all required values have been entered into the tool, you can use the **Action** menu to **Import**, **Export**, and **Generate**. The JSON files required for deployment are as follows:

Import: Enables you to import an Azure Stack Hub configuration data file (ConfigurationData.json) that was generated by this tool, or those created by any previous release of the deployment worksheet. Performing an import resets the forms and deletes any previously entered setting or generated data.

Export: Validates the data currently entered into the forms, generates the IP subnets and assignments, and then saves the content as JSON-formatted configuration files. You can then use these files to generate the network configuration and install Azure Stack Hub.

Generate: Validates the currently entered data and generates the network map without exporting the deployment JSON files. Two new tabs are created if **Generate** is successful: **Subnet Summary** and **IP Assignments**. You can analyze the data on these tabs to ensure the network assignments are as expected.

Clear All: Clears all data currently entered in the forms and returns them to default values.

Save or Open your work in-progress: You can save and open partially entered data as you are working on it, using the **File->Save** and **File->Open** menus. This differs from the **Import** and **Export** functions, as these require all data to be entered and validated. Open/save does not validate and does not require all fields to be entered to save your work in progress.

Logging and Warning messages: While the form is being used, you might see non-critical warning messages

displayed in the PowerShell window. Critical errors are displayed as a pop-up message. Optional detailed logging, including a log written to disk, can be enabled to assist in troubleshooting problems.

To start the tool with verbose logging:

```
Start-DeploymentWorksheet -EnableLogging
```

You can find the saved log in the current user's **Temp** directory; for example:

C:\Users\me\AppData\Local\Temp\Microsoft_AzureStack\DeploymentWorksheet_Log.txt.

Next steps

- [Azure Stack Hub deployment connection models](#)

Integrate AD FS identity with your Azure Stack Hub datacenter

11 minutes to read • [Edit Online](#)

You can deploy Azure Stack Hub using Azure Active Directory (Azure AD) or Active Directory Federation Services (AD FS) as the identity provider. You must make the choice before you deploy Azure Stack Hub. In a connected scenario, you can choose Azure AD or AD FS. For a disconnected scenario, only AD FS is supported. This article shows how to integrate Azure Stack Hub AD FS with your datacenter AD FS.

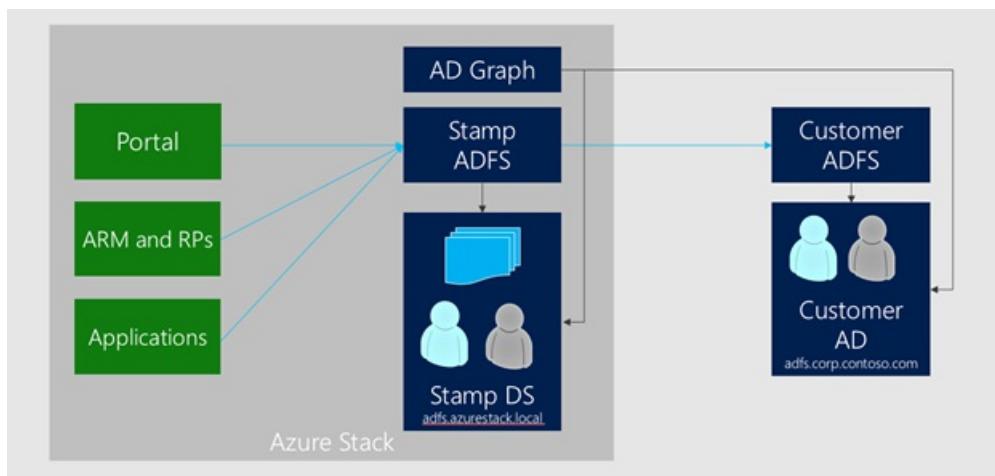
IMPORTANT

You can't switch the identity provider without redeploying the entire Azure Stack Hub solution.

Active Directory Federation Services and Graph

Deploying with AD FS allows identities in an existing Active Directory forest to authenticate with resources in Azure Stack Hub. This existing Active Directory forest requires a deployment of AD FS to allow the creation of an AD FS federation trust.

Authentication is one part of identity. To manage role-based access control (RBAC) in Azure Stack Hub, the Graph component must be configured. When access to a resource is delegated, the Graph component looks up the user account in the existing Active Directory forest using the LDAP protocol.



The existing AD FS is the account security token service (STS) that sends claims to the Azure Stack Hub AD FS (the resource STS). In Azure Stack Hub, automation creates the claims provider trust with the metadata endpoint for the existing AD FS.

At the existing AD FS, a relying party trust must be configured. This step isn't done by the automation, and must be configured by the operator. The Azure Stack Hub VIP endpoint for AD FS can be created by using the pattern `https://adfs.<Region>.<ExternalFQDN>/`.

The relying party trust configuration also requires you to configure the claim transformation rules that are provided by Microsoft.

For the Graph configuration, a service account must be provided that has read permission in the existing Active Directory. This account is required as input for the automation to enable RBAC scenarios.

For the last step, a new owner is configured for the default provider subscription. This account has full access to all resources when signed in to the Azure Stack Hub administrator portal.

Requirements:

COMPONENT	REQUIREMENT
Graph	Microsoft Active Directory 2012/2012 R2/2016
AD FS	Windows Server 2012/2012 R2/2016

Setting up Graph integration

Graph only supports integration with a single Active Directory forest. If multiple forests exist, only the forest specified in the configuration will be used to fetch users and groups.

The following information is required as inputs for the automation parameters:

PARAMETER	DEPLOYMENT WORKSHEET PARAMETER	DESCRIPTION	EXAMPLE
CustomADGlobalCatalog	AD FS Forest FQDN	FQDN of the target Active Directory forest that you want to integrate with	Contoso.com
CustomADAdminCredentials		A user with LDAP Read permission	YOURDOMAIN\graphservice

Configure Active Directory Sites

For Active Directory deployments having multiple sites, configure the closest Active Directory Site to your Azure Stack Hub deployment. The configuration avoids having the Azure Stack Hub Graph service resolve queries using a Global Catalog Server from a remote site.

Add the Azure Stack Hub [Public VIP network](#) subnet to the Active Directory Site closest to Azure Stack Hub. For example, let's say your Active Directory has two sites: Seattle and Redmond. If Azure Stack Hub is deployed at the Seattle site, you would add the Azure Stack Hub Public VIP network subnet to the Active Directory site for Seattle.

For more information on Active Directory Sites, see [Designing the site topology](#).

NOTE

If your Active Directory consist of a single site, you can skip this step. If you have a catch-all subnet configured, validate that the Azure Stack Hub Public VIP network subnet isn't part of it.

Create user account in the existing Active Directory (optional)

Optionally, you can create an account for the Graph service in the existing Active Directory. Do this step if you don't already have an account that you want to use.

1. In the existing Active Directory, create the following user account (recommendation):

- **Username:** graphservice
 - **Password:** Use a strong password and configure the password to never expire.
- No special permissions or membership is required.

Trigger automation to configure graph

For this procedure, use a computer in your datacenter network that can communicate with the privileged endpoint in Azure Stack Hub.

1. Open an elevated Windows PowerShell session (run as administrator), and connect to the IP address of the privileged endpoint. Use the credentials for **CloudAdmin** to authenticate.

```
$creds = Get-Credential  
Enter-PSSession -ComputerName <IP Address of ERCS> -ConfigurationName PrivilegedEndpoint -Credential  
$creds
```

2. Now that you're connected to the privileged endpoint, run the following command:

```
Register-DirectoryService -CustomADGlobalCatalog contoso.com
```

When prompted, specify the credential for the user account that you want to use for the Graph service (such as graphservice). The input for the Register-DirectoryService cmdlet must be the forest name / root domain in the forest rather than any other domain in the forest.

IMPORTANT

Wait for the credentials pop-up (Get-Credential isn't supported in the privileged endpoint) and enter the Graph Service Account credentials.

3. The **Register-DirectoryService** cmdlet has optional parameters that you can use in certain scenarios where the existing Active Directory validation fails. When this cmdlet is executed, it validates that the provided domain is the root domain, a global catalog server can be reached, and that the provided account is granted read access.

PARAMETER	DESCRIPTION
<code>-SkipRootDomainValidation</code>	Specifies that a child domain must be used instead of the recommended root domain.
<code>-Force</code>	Bypasses all validation checks.

Graph protocols and ports

Graph service in Azure Stack Hub uses the following protocols and ports to communicate with a writeable Global Catalog Server (GC) and Key Distribution Center (KDC) that can process login requests in the target Active Directory forest.

Graph service in Azure Stack Hub uses the following protocols and ports to communicate with the target Active Directory:

TYPE	PORT	PROTOCOL
LDAP	389	TCP & UDP
LDAP SSL	636	TCP
LDAP GC	3268	TCP
LDAP GC SSL	3269	TCP

Setting up AD FS integration by downloading federation metadata

The following information is required as input for the automation parameters:

PARAMETER	DEPLOYMENT WORKSHEET PARAMETER	DESCRIPTION	EXAMPLE
CustomAdfsName	AD FS Provider Name	Name of the claims provider. It appears that way on the AD FS landing page.	Contoso
CustomADFSFederationMetadataEndpointUri	AD FS Metadata URI	Federation metadata link.	https://ad01.contoso.com/federationmetadata/2007-06/federationmetadata.xml
SigningCertificateRevocationCheck	NA	Optional Parameter to skip CRL checking.	None

Trigger automation to configure claims provider trust in Azure Stack Hub

For this procedure, use a computer that can communicate with the privileged endpoint in Azure Stack Hub. It's expected that the certificate used by the account **STS AD FS** is trusted by Azure Stack Hub.

1. Open an elevated Windows PowerShell session and connect to the privileged endpoint.

```
$creds = Get-Credential  
Enter-PSSession -ComputerName <IP Address of ERCS> -ConfigurationName PrivilegedEndpoint -Credential  
$creds
```

2. Now that you're connected to the privileged endpoint, run the following command using the parameters appropriate for your environment:

```
Register-CustomAdfs -CustomAdfsName Contoso -CustomADFSFederationMetadataEndpointUri https://win-SQ00JN70SGL.contoso.com/federationmetadata/2007-06/federationmetadata.xml
```

3. Run the following command to update the owner of the default provider subscription using the parameters appropriate for your environment:

```
Set-ServiceAdminOwner -ServiceAdminOwnerUpn "administrator@contoso.com"
```

Setting up AD FS integration by providing federation metadata file

Beginning with version 1807, use this method if the either of the following conditions are true:

- The certificate chain is different for AD FS compared to all other endpoints in Azure Stack Hub.
- There's no network connectivity to the existing AD FS server from Azure Stack Hub's AD FS instance.

The following information is required as input for the automation parameters:

PARAMETER	DESCRIPTION	EXAMPLE
CustomAdfsName	Name of the claims provider. It appears that way on the AD FS landing page.	Contoso

PARAMETER	DESCRIPTION	EXAMPLE
CustomADSFederationMetadataFileContent	Metadata content.	\$using:federationMetadataFileContent

Create federation metadata file

For the following procedure, you must use a computer that has network connectivity to the existing AD FS deployment, which becomes the account STS. The necessary certificates must also be installed.

1. Open an elevated Windows PowerShell session, and run the following command using the parameters appropriate for your environment:

```
$url = "https://win-SQ00JN70SGL.contoso.com/FederationMetadata/2007-06/FederationMetadata.xml"
$webclient = New-Object System.Net.WebClient
$webclient.Encoding = [System.Text.Encoding]::UTF8
$metadataAsString = $webclient.DownloadString($url)
Set-Content -Path c:\metadata.xml -Encoding UTF8 -Value $metadataAsString
```

2. Copy the metadata file to a computer that can communicate with the privileged endpoint.

Trigger automation to configure claims provider trust in Azure Stack Hub

For this procedure, use a computer that can communicate with the privileged endpoint in Azure Stack Hub and has access to the metadata file you created in a previous step.

1. Open an elevated Windows PowerShell session and connect to the privileged endpoint.

```
$federationMetadataFileContent = get-content c:\metadata.xml
$creds=Get-Credential
Enter-PSSession -ComputerName <IP Address of ERCS> -ConfigurationName PrivilegedEndpoint -Credential
$creds
```

2. Now that you're connected to the privileged endpoint, run the following command using the parameters appropriate for your environment:

```
Register-CustomAdfs -CustomAdfsName Contoso -CustomADSFederationMetadataFileContent
$using:federationMetadataFileContent
```

3. Run the following command to update the owner of the default provider subscription. Use the parameters appropriate for your environment.

```
Set-ServiceAdminOwner -ServiceAdminOwnerUpn "administrator@contoso.com"
```

NOTE

When you rotate the certificate on the existing AD FS (account STS), you must set up the AD FS integration again. You must set up the integration even if the metadata endpoint is reachable or it was configured by providing the metadata file.

Configure relying party on existing AD FS deployment (account STS)

Microsoft provides a script that configures the relying party trust, including the claim transformation rules. Using the script is optional as you can run the commands manually.

You can download the helper script from [Azure Stack Hub Tools](#) on GitHub.

If you decide to manually run the commands, follow these steps:

1. Copy the following content into a .txt file (for example, saved as c:\ClaimRules.txt) on your datacenter's AD FS instance or farm member:

```
@RuleTemplate = "LdapClaims"
@RuleName = "Name claim"
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"]
=> issue(store = "Active Directory", types =
("http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name"), query = ";userPrincipalName;{0}", param
= c.Value);

@RuleTemplate = "LdapClaims"
@RuleName = "UPN claim"
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"]
=> issue(store = "Active Directory", types =
("http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"), query = ";userPrincipalName;{0}", param
= c.Value);

@RuleTemplate = "LdapClaims"
@RuleName = "ObjectID claim"
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid"]
=> issue(Type = "http://schemas.microsoft.com/identity/claims/objectidentifier", Issuer = c.Issuer,
OriginalIssuer = c.OriginalIssuer, Value = c.Value, ValueType = c.ValueType);

@RuleName = "Family Name and Given claim"
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"]
=> issue(store = "Active Directory", types =
("http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname",
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname"), query = ";sn,givenName;{0}", param
= c.Value);

@RuleTemplate = "PassThroughClaims"
@RuleName = "Pass through all Group SID claims"
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid", Issuer =~ "^(AD AUTHORITY|SELF AUTHORITY|LOCAL AUTHORITY)$"]
=> issue(claim = c);

@RuleTemplate = "PassThroughClaims"
@RuleName = "Pass through all windows account name claims"
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname"]
=> issue(claim = c);
```

2. Validate that Windows Forms-based authentication for extranet and intranet is enabled. You can check if its already enabled by running the following cmdlet:

```
Get-AdfsAuthenticationProvider | where-object { $_.name -eq "FormsAuthentication" } | select Name,
AllowedForPrimaryExtranet, AllowedForPrimaryIntranet
```

NOTE

The Windows Integrated Authentication (WIA) supported user agent strings may be outdated for your AD FS deployment and may require an update to support the latest clients. You can read more about updating the WIA supported user agent strings in the article [Configuring intranet forms-based authentication for devices that don't support WIA](#).

For steps to enable Form-based authentication policy, see [Configure Authentication Policies](#).

3. To add the relying party trust, run the following Windows PowerShell command on your AD FS instance or a farm member. Make sure to update the AD FS endpoint and point to the file created in Step 1.

For AD FS 2016

```
Add-ADFSRelyingPartyTrust -Name AzureStack -MetadataUrl  
"https://YourAzureStackADFSEndpoint/FederationMetadata/2007-06/FederationMetadata.xml" -  
IssuanceTransformRulesFile "C:\ClaimIssuanceRules.txt" -AutoUpdateEnabled:$true -  
MonitoringEnabled:$true -enabled:$true -AccessControlPolicyName "Permit everyone" -TokenLifeTime 1440
```

For AD FS 2012/2012 R2

```
Add-ADFSRelyingPartyTrust -Name AzureStack -MetadataUrl  
"https://YourAzureStackADFSEndpoint/FederationMetadata/2007-06/FederationMetadata.xml" -  
IssuanceTransformRulesFile "C:\ClaimIssuanceRules.txt" -AutoUpdateEnabled:$true -  
MonitoringEnabled:$true -enabled:$true -TokenLifeTime 1440
```

IMPORTANT

You must use the AD FS MMC snap-in to configure the Issuance Authorization Rules when using Windows Server 2012 or 2012 R2 AD FS.

4. When you use Internet Explorer or the Microsoft Edge browser to access Azure Stack Hub, you must ignore token bindings. Otherwise, the sign-in attempts fail. On your AD FS instance or a farm member, run the following command:

NOTE

This step isn't applicable when using Windows Server 2012 or 2012 R2 AD FS. In that case, it's safe to skip this command and continue with the integration.

```
Set-AdfsProperties -IgnoreTokenBinding $true
```

SPN creation

There are many scenarios that require the use of a service principal name (SPN) for authentication. The following are some examples:

- CLI usage with AD FS deployment of Azure Stack Hub.
- System Center Management Pack for Azure Stack Hub when deployed with AD FS.
- Resource providers in Azure Stack Hub when deployed with AD FS.
- Various apps.

- You require a non-interactive sign-in.

IMPORTANT

AD FS only supports interactive sign-in sessions. If you require a non-interactive sign-in for an automated scenario, you must use a SPN.

For more information on creating an SPN, see [Create service principal for AD FS](#).

Troubleshooting

Configuration Rollback

If an error occurs that leaves the environment in a state where you can no longer authenticate, a rollback option is available.

1. Open an elevated Windows PowerShell session and run the following commands:

```
$creds = Get-Credential  
Enter-PSSession -ComputerName <IP Address of ERCS> -ConfigurationName PrivilegedEndpoint -Credential  
$creds
```

2. Then run the following cmdlet:

```
Reset-DatacenterIntegrationConfiguration
```

After running the rollback action, all configuration changes are rolled back. Only authentication with the built-in **CloudAdmin** user is possible.

IMPORTANT

You must configure the original owner of the default provider subscription.

```
Set-ServiceAdminOwner -ServiceAdminOwnerUpn "azurestackadmin@[Internal Domain]"
```

Collecting additional logs

If any of the cmdlets fail, you can collect additional logs by using the `Get-AzureStackLogs` cmdlet.

1. Open an elevated Windows PowerShell session and run the following commands:

```
$creds = Get-Credential  
Enter-pssession -ComputerName <IP Address of ERCS> -ConfigurationName PrivilegedEndpoint -Credential  
$creds
```

2. Then run the following cmdlet:

```
Get-AzureStackLog -OutputPath \\myworkstation\AzureStackLogs -FilterByRole ECE
```

Next steps

[Integrate external monitoring solutions](#)

Create a custom role for Azure Stack Hub registration

2 minutes to read • [Edit Online](#)

WARNING

This isn't a security posture feature. Use it in scenarios where you want constraints to prevent accidental changes to the Azure Subscription. When a user is delegated rights to this custom role, the user has rights to edit permissions and elevate rights. Only assign users you trust to the custom role.

During Azure Stack Hub registration, you must sign in with an Azure Active Directory (Azure AD) account. The account requires the following Azure AD permissions and Azure Subscription permissions:

- **App registration permissions in your Azure AD tenant:** Admins have app registration permissions. The permission for users is a global setting for all users in the tenant. To view or change the setting, see [create an Azure AD app and service principal that can access resources](#).

The *user can register applications* setting must be set to **Yes** for you to enable a user account to register Azure Stack Hub. If the app registrations setting is set to **No**, you can't use a user account to register Azure Stack Hub—you have to use a global admin account.

- **A set of sufficient Azure Subscription permissions:** Users that belong to the Owner role have sufficient permissions. For other accounts, you can assign the permission set by assigning a custom role as outlined in the following sections.

Rather than using an account that has Owner permissions in the Azure subscription, you can create a custom role to assign permissions to a less-privileged user account. This account can then be used to register your Azure Stack Hub.

Create a custom role using PowerShell

To create a custom role, you must have the `Microsoft.Authorization/roleDefinitions/write` permission on all `AssignableScopes`, such as **Owner** or **User Access Administrator**. Use the following JSON template to simplify creation of the custom role. The template creates a custom role that allows the required read and write access for Azure Stack Hub registration.

1. Create a JSON file. For example, `C:\CustomRoles\registrationrole.json`.
2. Add the following JSON to the file. Replace `<SubscriptionID>` with your Azure subscription ID.

```
{  
  "Name": "Azure Stack Hub registration role",  
  "Id": null,  
  "IsCustom": true,  
  "Description": "Allows access to register Azure Stack Hub",  
  "Actions": [  
    "Microsoft.Resources/subscriptions/resourceGroups/write",  
    "Microsoft.Resources/subscriptions/resourceGroups/read",  
    "Microsoft.AzureStack/registrations/*",  
    "Microsoft.AzureStack/register/action",  
    "Microsoft.Authorization/roleAssignments/read",  
    "Microsoft.Authorization/roleAssignments/write",  
    "Microsoft.Authorization/roleAssignments/delete",  
    "Microsoft.Authorization/permissions/read"  
  ],  
  "NotActions": [  
  ],  
  "AssignableScopes": [  
    "/subscriptions/<SubscriptionID>"  
  ]  
}
```

3. In PowerShell, connect to Azure to use Azure Resource Manager. When prompted, authenticate using an account with sufficient permissions such as [Owner](#) or [User Access Administrator](#).

```
Connect-AzureRmAccount
```

4. To create the custom role, use **New-AzureRmRoleDefinition** specifying the JSON template file.

```
New-AzureRmRoleDefinition -InputFile "C:\CustomRoles\registrationrole.json"
```

Assign a user to registration role

After the registration custom role is created, assign the role to the user account that will be used for registering Azure Stack Hub.

1. Sign in with the account with sufficient permission on the Azure subscription to delegate rights—such as [Owner](#) or [User Access Administrator](#).
2. In **Subscriptions**, select **Access control (IAM)** > **Add role assignment**.
3. In **Role**, choose the custom role you created: *Azure Stack Hub registration role*.
4. Select the users you want to assign to the role.
5. Select **Save** to assign the selected users to the role.

The screenshot shows two windows side-by-side. On the left is the 'Subscriptions' page for 'Contoso'. It displays a list of subscriptions, with 'Visual Studio Enterprise' selected. A search bar and filter options are also present. On the right is the 'Add role assignment' dialog for the 'Azure Stack registration role'. It shows 'Erin Silva' selected under 'Assign access to'. The 'Selected members:' section lists 'Erin Silva' with an 'ES' icon and an email link. Buttons for 'Save' and 'Discard' are at the bottom.

For more information on using custom roles, see [manage access using RBAC and the Azure portal](#).

Next steps

[Register Azure Stack Hub with Azure](#)

Validate Azure identity

5 minutes to read • [Edit Online](#)

Use the Azure Stack Hub Readiness Checker tool (**AzsReadinessChecker**) to validate that your Azure Active Directory (Azure AD) is ready to use with Azure Stack Hub. Validate your Azure identity solution before you begin an Azure Stack Hub deployment.

The readiness checker validates:

- Azure AD as an identity provider for Azure Stack Hub.
- The Azure AD account that you plan to use can sign in as a global administrator of your Azure AD.

Validation ensures your environment is ready for Azure Stack Hub to store information about users, applications, groups, and service principals from Azure Stack Hub in your Azure AD.

Get the readiness checker tool

Download the latest version of the Azure Stack Hub Readiness Checker tool (AzsReadinessChecker) from the [PowerShell Gallery](#).

Prerequisites

The following prerequisites are required:

The computer on which the tool runs:

- Windows 10 or Windows Server 2016 with internet connectivity.
- PowerShell 5.1 or later. To check your version, run the following PowerShell command, and then review the **Major** version and **Minor** versions:

```
$PSVersionTable.PSVersion
```

- [PowerShell configured for Azure Stack Hub](#).
- The latest version of [Microsoft Azure Stack Hub Readiness Checker](#) tool.

Azure AD environment:

- Identify the Azure AD account to use for Azure Stack Hub and ensure it's an Azure AD global administrator.
- Identify your Azure AD tenant name. The tenant name must be the primary domain name for your Azure AD. For example, **contoso.onmicrosoft.com**.
- Identify the Azure environment you'll use. Supported values for the environment name parameter are **AzureCloud**, **AzureChinaCloud**, or **AzureUSGovernment**, depending on which Azure subscription you use.

Steps to validate Azure identity

1. On a computer that meets the prerequisites, open an elevated PowerShell command prompt, and then run the following command to install **AzsReadinessChecker**:

```
Install-Module Microsoft.AzureStack.ReadinessChecker -Force
```

2. From the PowerShell prompt, run the following command to set `$serviceAdminCredential` as the service administrator for your Azure AD tenant. Replace `serviceadmin@contoso.onmicrosoft.com` with your account and tenant name:

```
$serviceAdminCredential = Get-Credential serviceadmin@contoso.onmicrosoft.com -Message "Enter credentials for service administrator of Azure Active Directory tenant"
```

3. From the PowerShell prompt, run the following command to start validation of your Azure AD:

- Specify the environment name value for **AzureEnvironment**. Supported values for the environment name parameter are **AzureCloud**, **AzureChinaCloud**, or **AzureUSGovernment**, depending on which Azure subscription you use.
- Replace `contoso.onmicrosoft.com` with your Azure AD tenant name.

```
Invoke-AzsAzureIdentityValidation -AADServiceAdministrator $serviceAdminCredential -AzureEnvironment <environment name> -ADDirectoryTenantName contoso.onmicrosoft.com
```

4. After the tool runs, review the output. Confirm the status is **OK** for installation requirements. A successful validation appears like the following example:

```
Invoke-AzsAzureIdentityValidation v1.1809.1005.1 started.  
Starting Azure Identity Validation  
  
Checking Installation Requirements: OK  
  
Finished Azure Identity Validation  
  
Log location (contains PII):  
C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessChecker.log  
Report location (contains PII):  
C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessCheckerReport.json  
Invoke-AzsAzureIdentityValidation Completed
```

Report and log file

Each time validation runs, it logs results to **AzsReadinessChecker.log** and **AzsReadinessCheckerReport.json**. The location of these files displays with the validation results in PowerShell.

These files can help you share validation status before you deploy Azure Stack Hub or investigate validation problems. Both files persist the results of each subsequent validation check. The report provides your deployment team confirmation of the identity configuration. The log file can help your deployment or support team investigate validation issues.

By default, both files are written to

`C:\Users\<username>\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessCheckerReport.json`.

- Use the `-OutputPath <path>` parameter at the end of the run command line to specify a different report location.
- Use the `-CleanReport` parameter at the end of the run command to clear information about previous runs of the tool from **AzsReadinessCheckerReport.json**.

For more information, see [Azure Stack Hub validation report](#).

Validation failures

If a validation check fails, details about the failure display in the PowerShell window. The tool also logs information

to the AzsReadinessChecker.log file.

The following examples provide guidance on common validation failures.

Expired or temporary password

```
Invoke-AzsAzureIdentityValidation v1.1809.1005.1 started.
Starting Azure Identity Validation

Checking Installation Requirements: Fail
Error Details for Service Administrator Account admin@contoso.onmicrosoft.com
The password for account has expired or is a temporary password that needs to be reset before continuing. Run
Login-AzureRMAccount, login with credentials and follow the prompts to reset.
Additional help URL https://aka.ms/AzsRemediateAzureIdentity

Finished Azure Identity Validation

Log location (contains PII): C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessChecker.log
Report location (contains PII):
C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessCheckerReport.json
Invoke-AzsAzureIdentityValidation Completed
```

Cause - The account can't sign in because the password is either expired or temporary.

Resolution - In PowerShell, run the following command and then follow the prompts to reset the password:

```
Login-AzureRMAccount
```

Another way is to sign in to the [Azure portal](#) as the account owner and the user will be forced to change the password.

Unknown user type

```
Invoke-AzsAzureIdentityValidation v1.1809.1005.1 started.
Starting Azure Identity Validation

Checking Installation Requirements: Fail
Error Details for Service Administrator Account admin@contoso.onmicrosoft.com
Unknown user type detected. Check the account is valid for AzureChinaCloud
Additional help URL https://aka.ms/AzsRemediateAzureIdentity

Finished Azure Identity Validation

Log location (contains PII): C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessChecker.log
Report location (contains PII):
C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessCheckerReport.json
Invoke-AzsAzureIdentityValidation Completed
```

Cause - The account can't sign in to the specified Azure AD (**AADDirectoryTenantName**). In this example, **AzureChinaCloud** is specified as the **AzureEnvironment**.

Resolution - Confirm that the account is valid for the specified Azure environment. In PowerShell, run the following command to verify the account is valid for a specific environment:

```
Login-AzureRmAccount -EnvironmentName AzureChinaCloud
```

Account is not an administrator

```
Invoke-AzsAzureIdentityValidation v1.1809.1005.1 started.
Starting Azure Identity Validation

Checking Installation Requirements: Fail
Error Details for Service Administrator Account admin@contoso.onmicrosoft.com
The Service Admin account you entered 'admin@contoso.onmicrosoft.com' is not an administrator of the Azure
Active Directory tenant 'contoso.onmicrosoft.com'.
Additional help URL https://aka.ms/AzsRemediateAzureIdentity

Finished Azure Identity Validation

Log location (contains PII): C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessChecker.log
Report location (contains PII):
C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessCheckerReport.json
Invoke-AzsAzureIdentityValidation Completed
```

Cause - Although the account can successfully sign in, the account isn't an admin of the Azure AD (**AADDirectoryTenantName**).

Resolution - Sign in into the [Azure portal](#) as the account owner, go to **Azure Active Directory**, then **Users**, then **Select the User**. Then select **Directory Role** and ensure the user is a **Global administrator**. If the account is a **User**, go to **Azure Active Directory > Custom domain names** and confirm that the name you supplied for **AADDirectoryTenantName** is marked as the primary domain name for this directory. In this example, that's **contoso.onmicrosoft.com**.

Azure Stack Hub requires that the domain name is the primary domain name.

Next Steps

[Validate Azure registration](#)

[View the readiness report](#)

[General Azure Stack Hub integration considerations](#)

Validate AD FS integration for Azure Stack Hub

3 minutes to read • [Edit Online](#)

Use the Azure Stack Hub Readiness Checker tool (AzsReadinessChecker) to validate that your environment is ready for Active Directory Federation Services (AD FS) integration with Azure Stack Hub. Validate AD FS integration before you begin datacenter integration or before an Azure Stack Hub deployment.

The readiness checker validates:

- The *federation metadata* contains the valid XML elements for federation.
- The *AD FS SSL certificate* can be retrieved and a chain of trust can be built. On stamp, AD FS must trust the SSL certificate chain. The certificate must be signed by the same *certificate authority* used for the Azure Stack Hub deployment certificates or by a trusted root authority partner. For the full list of trusted root authority partners, see [TechNet](#).
- The *AD FS signing certificate* is trusted and not nearing expiration.

For more information about Azure Stack Hub datacenter integration, see [Azure Stack Hub datacenter integration - Identity](#).

Get the readiness checker tool

Download the latest version of the Azure Stack Hub Readiness Checker tool (AzsReadinessChecker) from the [PowerShell Gallery](#).

Prerequisites

The following prerequisites must be in place.

The computer where the tool runs:

- Windows 10 or Windows Server 2016 with domain connectivity.
- PowerShell 5.1 or later. To check your version, run the following PowerShell command and then review the *Major* version and *Minor* versions:

```
$PSVersionTable.PSVersion
```

- Latest version of the [Microsoft Azure Stack Hub Readiness Checker](#) tool.

Active Directory Federation Services environment:

You need at least one of the following forms of metadata:

- The URL for AD FS federation metadata. For example:
`https://adfs.contoso.com/FederationMetadata/2007-06/FederationMetadata.xml`.
- The federation metadata XML file. For example: FederationMetadata.xml.

Validate AD FS integration

1. On a computer that meets the prerequisites, open an administrative PowerShell prompt and then run the following command to install AzsReadinessChecker:

```
Install-Module Microsoft.AzureStack.ReadinessChecker -Force
```

2. From the PowerShell prompt, run the following command to start validation. Specify the value for **-CustomADFSFederationMetadataEndpointUri** as the URI for the federation metadata.

```
Invoke-AzsADFSValidation -CustomADFSFederationMetadataEndpointUri  
https://adfs.contoso.com/FederationMetadata/2007-06/FederationMetadata.xml
```

3. After the tool runs, review the output. Confirm that the status is OK for AD FS integration requirements. A successful validation is similar to the following example:

```
Invoke-AzsADFSValidation v1.1809.1001.1 started.

Testing ADFS Endpoint https://sts.contoso.com/FederationMetadata/2007-06/FederationMetadata.xml

Read Metadata:          OK
Test Metadata Elements: OK
Test SSL ADFS Certificate: OK
Test Certificate Chain: OK
Test Certificate Expiry: OK

Details:
[-] In standalone mode, some tests should not be considered fully indicative of connectivity or readiness the Azure Stack Hub Stamp requires prior to Datacenter Integration.
Additional help URL: https://aka.ms/AzsADFSIntegration

Log location (contains PII):
C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessChecker.log
Report location (contains PII):
C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessCheckerReport.json

Invoke-AzsADFSValidation Completed
```

In production environments, testing certificate chains of trust from an operator's workstation isn't fully indicative of the PKI trust posture in the Azure Stack Hub infrastructure. The Azure Stack Hub stamp's public VIP network needs the connectivity to the CRL for the PKI infrastructure.

Report and log file

Each time validation runs, it logs results to **AzsReadinessChecker.log** and **AzsReadinessCheckerReport.json**.

The location of these files appears with the validation results in PowerShell.

The validation files can help you share status before you deploy Azure Stack Hub or investigate validation problems. Both files persist the results of each subsequent validation check. The report gives your deployment team confirmation of the identity configuration. The log file can help your deployment or support team investigate validation issues.

By default, both files are written to `c:\Users\<username>\AppData\Local\Temp\AzsReadinessChecker\`.

Use:

- `-OutputPath` : The *path* parameter at the end of the run command to specify a different report location.
- `-CleanReport` : The parameter at the end of the run command to clear AzsReadinessCheckerReport.json of previous report information. For more information, see [Azure Stack Hub validation report](#).

Validation failures

If a validation check fails, details about the failure appear in the PowerShell window. The tool also logs information to *AzsReadinessChecker.log*.

The following examples provide guidance on common validation failures.

Command Not Found

```
Invoke-AzsADFSValidation : The term 'Invoke-AzsADFSValidation' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again.
```

Cause: PowerShell Autoload failed to load the Readiness Checker module correctly.

Resolution: Import the Readiness Checker module explicitly. Copy and paste the following code into PowerShell and update <version> with the number for the currently installed version.

```
Import-Module "c:\Program Files\WindowsPowerShell\Modules\Microsoft.AzureStack.ReadinessChecker\  
<version>\Microsoft.AzureStack.ReadinessChecker.psd1" -Force
```

Next steps

[View the readiness report](#)

[General Azure Stack Hub integration considerations](#)

Validate graph integration for Azure Stack Hub

3 minutes to read • [Edit Online](#)

Use the Azure Stack Hub Readiness Checker tool (AzsReadinessChecker) to validate that your environment is ready for graph integration with Azure Stack Hub. Validate graph integration before you begin datacenter integration or before an Azure Stack Hub deployment.

The readiness checker validates:

- The credentials to the service account created for graph integration have appropriate rights to query Active Directory.
- The *global catalog* can be resolved and is contactable.
- The KDC can be resolved and is contactable.
- Necessary network connectivity is in place.

For more information about Azure Stack Hub datacenter integration, see [Azure Stack Hub datacenter integration - Identity](#).

Get the readiness checker tool

Download the latest version of the Azure Stack Hub Readiness Checker tool (AzsReadinessChecker) from the [PowerShell Gallery](#).

Prerequisites

The following prerequisites must be in place.

The computer where the tool runs:

- Windows 10 or Windows Server 2016 with domain connectivity.
- PowerShell 5.1 or later. To check your version, run the following PowerShell command and then review the *Major* version and *Minor* versions:

```
$PSVersionTable.PSVersion
```

- Active Directory PowerShell module.
- Latest version of the [Microsoft Azure Stack Hub Readiness Checker](#) tool.

Active Directory environment:

- Identify the username and password for an account for the graph service in the existing Active Directory instance.
- Identify the Active Directory forest root FQDN.

Validate the graph service

1. On a computer that meets the prerequisites, open an administrative PowerShell prompt and then run the following command to install the AzsReadinessChecker:

```
Install-Module Microsoft.AzureStack.ReadinessChecker -Force
```

2. From the PowerShell prompt, run the following command to set the `$graphCredential` variable to the graph account. Replace `contoso\graphservice` with your account by using the `domain\username` format.

```
$graphCredential = Get-Credential contoso\graphservice -Message "Enter Credentials for the Graph Service Account"
```

3. From the PowerShell prompt, run the following command to start validation for the graph service. Specify the value for `-ForestFQDN` as the FQDN for the forest root.

```
Invoke-AzsGraphValidation -ForestFQDN contoso.com -Credential $graphCredential
```

4. After the tool runs, review the output. Confirm that the status is OK for graph integration requirements. A successful validation is similar to the following example:

```
Testing Graph Integration (v1.0)
  Test Forest Root:          OK
  Test Graph Credential:     OK
  Test Global Catalog:       OK
  Test KDC:                  OK
  Test LDAP Search:          OK
  Test Network Connectivity: OK
```

Details:

[–] In standalone mode, some tests should not be considered fully indicative of connectivity or readiness the Azure Stack Hub Stamp requires prior to Datacenter Integration.

Additional help URL: <https://aka.ms/AzsGraphIntegration>

AzsReadinessChecker Log location (contains PII):
C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessChecker.log

AzsReadinessChecker Report location (contains PII):
C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessCheckerReport.json

Invoke-AzsGraphValidation Completed

In production environments, testing network connectivity from an operator's workstation isn't fully indicative of the connectivity available to Azure Stack Hub. The Azure Stack Hub stamp's public VIP network needs the connectivity for LDAP traffic to perform identity integration.

Report and log file

Each time validation runs, it logs results to **AzsReadinessChecker.log** and **AzsReadinessCheckerReport.json**. The location of these files appears with the validation results in PowerShell.

The validation files can help you share status before you deploy Azure Stack Hub or investigate validation problems. Both files persist the results of each subsequent validation check. The report gives your deployment team confirmation of the identity configuration. The log file can help your deployment or support team investigate validation issues.

By default, both files are written to `C:\Users\<username>\AppData\Local\Temp\AzsReadinessChecker\`.

Use:

- `-outputPath`: The *path* parameter at the end of the run command to specify a different report location.
- `-CleanReport`: The parameter at the end of the run command to clear *AzsReadinessCheckerReport.json* of

previous report information. For more information, see [Azure Stack Hub validation report](#).

Validation failures

If a validation check fails, details about the failure appear in the PowerShell window. The tool also logs information to *AzsGraphIntegration.log*.

Next steps

[View the readiness report](#)

[General Azure Stack Hub integration considerations](#)

Azure Stack Hub public key infrastructure certificate requirements

7 minutes to read • [Edit Online](#)

Azure Stack Hub has a public infrastructure network using externally accessible public IP addresses assigned to a small set of Azure Stack Hub services and possibly tenant VMs. PKI certificates with the appropriate DNS names for these Azure Stack Hub public infrastructure endpoints are required during Azure Stack Hub deployment. This article provides information about:

- What certificates are required to deploy Azure Stack Hub.
- The process of obtaining certificates matching those specifications.
- How to prepare, validate, and use those certificates during deployment.

NOTE

Azure Stack Hub by default also uses certificates issued from an internal Active Directory-integrated certificate authority (CA) for authentication between the nodes. To validate the certificate, all Azure Stack Hub infrastructure machines trust the root certificate of the internal CA by means of adding that certificate to their local certificate store. There's no pinning or whitelisting of certificates in Azure Stack Hub. The SAN of each server certificate is validated against the FQDN of the target. The entire chain of trust is also validated, along with the certificate expiration date (standard TLS server authentication without certificate pinning).

Certificate requirements

The following list describes the certificate requirements that are needed to deploy Azure Stack Hub:

- Certificates must be issued from either an internal certificate authority or a public certificate authority. If a public certificate authority is used, it must be included in the base operating system image as part of the Microsoft Trusted Root Authority Program. For the full list, see [Microsoft Trusted Root Certificate Program: Participants](#).
- Your Azure Stack Hub infrastructure must have network access to the certificate authority's Certificate Revocation List (CRL) location published in the certificate. This CRL must be an http endpoint.
- When rotating certificates in pre-1903 builds, certificates must be either issued from the same internal certificate authority used to sign certificates provided at deployment or any public certificate authority from above. For 1903 and later, certificates can be issued by any enterprise or public certificate authority.
- The use of self-signed certificates aren't supported.
- For deployment and rotation, you can either use a single certificate covering all name spaces in the certificate's Subject Name and Subject Alternative Name (SAN) fields OR you can use individual certificates for each of the namespaces below that the Azure Stack Hub services you plan to utilize require. Both approaches require using wild cards for endpoints where they're required, such as **KeyVault** and **KeyVaultInternal**.
- The certificate's PFX Encryption should be 3DES.
- The certificate signature algorithm shouldn't be SHA1.
- The certificate format must be PFX, as both the public and private keys are required for Azure Stack Hub installation. The private key must have the local machine key attribute set.
- The PFX encryption must be 3DES (this encryption is default when exporting from a Windows 10 client or Windows Server 2016 certificate store).

- The certificate pfx files must have a value "Digital Signature" and "KeyEncipherment" in its "Key Usage" field.
- The certificate pfx files must have the values "Server Authentication (1.3.6.1.5.5.7.3.1)" and "Client Authentication (1.3.6.1.5.5.7.3.2)" in the "Enhanced Key Usage" field.
- The certificate's "Issued to:" field must not be the same as its "Issued by:" field.
- The passwords to all certificate pfx files must be the same at the time of deployment.
- Password to the certificate pfx has to be a complex password. Make note of this password because you'll use it as a deployment parameter. The password must meet the following password complexity requirements:
 - A minimum length of eight characters.
 - At least three of the following characters: uppercase letter, lowercase letter, numbers from 0-9, special characters, alphabetical character that's not uppercase or lowercase.
- Ensure that the subject names and subject alternative names in the subject alternative name extension (x509v3_config) match. The subject alternative name field lets you specify additional host names (websites, IP addresses, common names) to be protected by a single SSL certificate.

NOTE

Self-signed certificates aren't supported.

When deploying Azure Stack Hub in disconnected mode it is recommended to use certificates issued by an enterprise certificate authority. This is important because clients accessing Azure Stack Hub endpoints must be able to contact the certificate revocation list (CRL).

NOTE

The presence of Intermediary Certificate Authorities in a certificate's chain-of-trusts *is* supported.

Mandatory certificates

The table in this section describes the Azure Stack Hub public endpoint PKI certificates that are required for both Azure AD and AD FS Azure Stack Hub deployments. Certificate requirements are grouped by area, as well as the namespaces used and the certificates that are required for each namespace. The table also describes the folder in which your solution provider copies the different certificates per public endpoint.

Certificates with the appropriate DNS names for each Azure Stack Hub public infrastructure endpoint are required. Each endpoint's DNS name is expressed in the format: <prefix>. <region>. <fqdn>.

For your deployment, the [region] and [externalfqdn] values must match the region and external domain names that you chose for your Azure Stack Hub system. As an example, if the region name was *Redmond* and the external domain name was *contoso.com*, the DNS names would have the format <prefix>.redmond.contoso.com. The <prefix> values are predesignated by Microsoft to describe the endpoint secured by the certificate. In addition, the <prefix> values of the external infrastructure endpoints depend on the Azure Stack Hub service that uses the specific endpoint.

For the production environments, we recommend individual certificates are generated for each endpoint and copied into the corresponding directory. For development environments, certificates can be provided as a single wild card certificate covering all namespaces in the Subject and Subject Alternative Name (SAN) fields copied into all directories. A single certificate covering all endpoints and services is an insecure posture and hence development-only. Remember, both options require you to use wildcard certificates for endpoints like **acs** and Key Vault where they're required.

NOTE

During deployment, you must copy certificates to the deployment folder that matches the identity provider you're deploying against (Azure AD or AD FS). If you use a single certificate for all endpoints, you must copy that certificate file into each deployment folder as outlined in the following tables. The folder structure is pre-built in the deployment virtual machine and can be found at: C:\CloudDeployment\Setup\Certificates.

DEPLOYMENT FOLDER	REQUIRED CERTIFICATE SUBJECT AND SUBJECT ALTERNATIVE NAMES (SAN)	SCOPE (PER REGION)	SUBDOMAIN NAMESPACE
Public Portal	portal.<region>.<fqdn>	Portals	<region>.<fqdn>
Admin Portal	adminportal.<region>.<fqdn>	Portals	<region>.<fqdn>
Azure Resource Manager Public	management.<region>.<fqdn>	Azure Resource Manager	<region>.<fqdn>
Azure Resource Manager Admin	adminmanagement.<region>.<fqdn>	Azure Resource Manager	<region>.<fqdn>
ACSBlob	*.blob.<region>.<fqdn> (Wildcard SSL Certificate)	Blob Storage	blob.<region>.<fqdn>
ACSTable	*.table.<region>.<fqdn> (Wildcard SSL Certificate)	Table Storage	table.<region>.<fqdn>
ACSQueue	*.queue.<region>.<fqdn> (Wildcard SSL Certificate)	Queue Storage	queue.<region>.<fqdn>
KeyVault	*.vault.<region>.<fqdn> (Wildcard SSL Certificate)	Key Vault	vault.<region>.<fqdn>
KeyVaultInternal	*.adminvault.<region>.<fqdn> (Wildcard SSL Certificate)	Internal Keyvault	adminvault.<region>.<fqdn>
Admin Extension Host	*.adminhosting.<region>.<fqdn> (Wildcard SSL Certificates)	Admin Extension Host	adminhosting.<region>.<fqdn>
Public Extension Host	*.hosting.<region>.<fqdn> (Wildcard SSL Certificates)	Public Extension Host	hosting.<region>.<fqdn>

If you deploy Azure Stack Hub using the Azure AD deployment mode, you only need to request the certificates listed in previous table. But, if you deploy Azure Stack Hub using the AD FS deployment mode, you must also request the certificates described in the following table:

DEPLOYMENT FOLDER	REQUIRED CERTIFICATE SUBJECT AND SUBJECT ALTERNATIVE NAMES (SAN)	SCOPE (PER REGION)	SUBDOMAIN NAMESPACE
ADFS	adfs.<region>.<fqdn> (SSL Certificate)	ADFS	<region>.<fqdn>

DEPLOYMENT FOLDER	REQUIRED CERTIFICATE SUBJECT AND SUBJECT ALTERNATIVE NAMES (SAN)	SCOPE (PER REGION)	SUBDOMAIN NAMESPACE
Graph	graph.<region>.<fqdn> (SSL Certificate)	Graph	<region>.<fqdn>

IMPORTANT

All the certificates listed in this section must have the same password.

Optional PaaS certificates

If you're planning to deploy the additional Azure Stack Hub PaaS services (SQL, MySQL, and App Service) after Azure Stack Hub has been deployed and configured, you need to request additional certificates to cover the endpoints of the PaaS services.

IMPORTANT

The certificates that you use for App Service, SQL, and MySQL resource providers need to have the same root authority as those used for the global Azure Stack Hub endpoints.

The following table describes the endpoints and certificates required for the SQL and MySQL adapters and for App Service. You don't need to copy these certificates to the Azure Stack Hub deployment folder. Instead, you provide these certificates when you install the additional resource providers.

SCOPE (PER REGION)	CERTIFICATE	REQUIRED CERTIFICATE SUBJECT AND SUBJECT ALTERNATIVE NAMES (SANS)	SUBDOMAIN NAMESPACE
SQL, MySQL	SQL and MySQL	*.dbadapter.<region>.<fqdn> (Wildcard SSL Certificate)	dbadapter.<region>.<fqdn>
App Service	Web Traffic Default SSL Cert	*.appservice.<region>.<fqdn> *.scm.appservice.<region>.<fqdn> *.sso.appservice.<region>.<fqdn> (Multi Domain Wildcard SSL Certificate ¹)	appservice.<region>.<fqdn> scm.appservice.<region>.<fqdn>
App Service	API	api.appservice.<region>.<fqdn> (SSL Certificate ²)	appservice.<region>.<fqdn> scm.appservice.<region>.<fqdn>
App Service	FTP	ftp.appservice.<region>.<fqdn> (SSL Certificate ²)	appservice.<region>.<fqdn> scm.appservice.<region>.<fqdn>

SCOPE (PER REGION)	CERTIFICATE	REQUIRED CERTIFICATE SUBJECT AND SUBJECT ALTERNATIVE NAMES (SANS)	SUBDOMAIN NAMESPACE
App Service	SSO	sso.appservice.<region>. <fqdn> (SSL Certificate ²)	appservice.<region>. <fqdn> scm.appservice.<region>. <fqdn>

¹ Requires one certificate with multiple wildcard subject alternative names. Multiple wildcard SANs on a single certificate might not be supported by all public certificate authorities.

² A *.appservice.<region>. <fqdn> wild card certificate can't be used in place of these three certificates (api.appservice.<region>. <fqdn>, ftp.appservice.<region>. <fqdn>, and sso.appservice.<region>. <fqdn>). Appservice explicitly requires the use of separate certificates for these endpoints.

Learn more

Learn how to [generate PKI certificates for Azure Stack Hub deployment](#).

Next steps

[Integrate AD FS identity with your Azure Stack Hub datacenter](#).

Generate certificate signing requests for Azure Stack Hub

2 minutes to read • [Edit Online](#)

You can use the Azure Stack Hub Readiness Checker tool to create Certificate Signing Requests (CSRs) suitable for an Azure Stack Hub deployment. Certificates should be requested, generated, and validated with enough time to test before deployment. You can get the tool [from the PowerShell Gallery](#).

You can use the Azure Stack Hub Readiness Checker tool (AzsReadinessChecker) to request the following certificates:

- **Standard Certificate Requests** according to [Generate certificate signing request](#).
- **Platform-as-a-Service**: You can request platform-as-a-service (PaaS) names for certificates as specified in [Azure Stack Hub Public Key Infrastructure certificate requirements - Optional PaaS Certificates](#).

Prerequisites

Your system should meet the following prerequisites before generating any CSRs for PKI certificates for an Azure Stack Hub deployment:

- Microsoft Azure Stack Hub Readiness Checker
- Certificate attributes:
 - Region name
 - External fully qualified domain name (FQDN)
 - Subject
- Windows 10 or Windows Server 2016 or later

NOTE

When you receive your certificates back from your certificate authority, the steps in [Prepare Azure Stack Hub PKI certificates](#) will need to be completed on the same system!

Generate certificate signing requests

Use these steps to prepare and validate the Azure Stack Hub PKI certificates:

1. Install AzsReadinessChecker from a PowerShell prompt (5.1 or above), by running the following cmdlet:

```
Install-Module Microsoft.AzureStack.ReadinessChecker
```

2. Declare the **subject**. For example:

```
$subject = "C=US,ST=Washington,L=Redmond,O=Microsoft,OU=Azure Stack Hub"
```

NOTE

If a common name (CN) is supplied, it will be configured on every certificate request. If a CN is omitted, the first DNS name of the Azure Stack Hub service will be configured on the certificate request.

3. Declare an output directory that already exists. For example:

```
$outputDirectory = "$ENV:USERPROFILE\Documents\AzureStackCSR"
```

4. Declare identity system.

Azure Active Directory (Azure AD):

```
$IdentitySystem = "AAD"
```

Active Directory Federation Services (AD FS):

```
$IdentitySystem = "ADFS"
```

NOTE

The parameter is required only for CertificateType Deployment.

5. Declare **region name** and an **external FQDN** intended for the Azure Stack Hub deployment.

```
$regionName = 'east'  
$externalFQDN = 'azurestack.contoso.com'
```

NOTE

`<regionName>.<externalFQDN>` forms the basis on which all external DNS names in Azure Stack Hub are created.
In this example, the portal would be `portal.east.azurestack.contoso.com`.

6. To generate certificate signing requests for deployment:

```
New-AzsCertificateSigningRequest -certificateType Deployment -RegionName $regionName -FQDN  
$externalFQDN -subject $subject -OutputRequestPath $OutputDirectory -IdentitySystem $IdentitySystem
```

To generate certificate requests for other Azure Stack Hub services, change the value for `-CertificateType`.
For example:

```

# App Services
New-AzsCertificateSigningRequest -certificateType AppServices -RegionName $regionName -FQDN
$externalFQDN -subject $subject -OutputRequestPath $OutputDirectory

# DBAdapter
New-AzsCertificateSigningRequest -certificateType DBAdapter -RegionName $regionName -FQDN
$externalFQDN -subject $subject -OutputRequestPath $OutputDirectory

# EventHubs
New-AzsCertificateSigningRequest -certificateType EventHubs -RegionName $regionName -FQDN
$externalFQDN -subject $subject -OutputRequestPath $OutputDirectory

# IoTHub
New-AzsCertificateSigningRequest -certificateType IoTHub -RegionName $regionName -FQDN $externalFQDN -
subject $subject -OutputRequestPath $OutputDirectory

```

7. Alternatively, for Dev/Test environments, to generate a single certificate request with multiple Subject Alternative Names add **-RequestType SingleCSR** parameter and value (**not** recommended for production environments):

```

New-AzsCertificateSigningRequest -certificateType Deployment -RegionName $regionName -FQDN
$externalFQDN -RequestType SingleCSR -subject $subject -OutputRequestPath $OutputDirectory -
IdentitySystem $IdentitySystem

```

8. Review the output:

```

New-AzsCertificateSigningRequest v1.1912.1082.37 started.
Starting Certificate Request Process for Deployment
CSR generating for following SAN(s):
*.adminhosting.east.azurestack.contoso.com,*.adminvault.east.azurestack.contoso.com,*.blob.east.azurestack.contoso.com,*.hosting.east.azurestack.contoso.com,*.queue.east.azurestack.contoso.com,*.table.east.azurestack.contoso.com,*.vault.east.azurestack.contoso.com,adminmanagement.east.azurestack.contoso.com,adminportal.east.azurestack.contoso.com,management.east.azurestack.contoso.com,portal.east.azurestack.contoso.com
Present this CSR to your Certificate Authority for Certificate Generation:
C:\Users\checker\Documents\AzureStackCSR\wildcard_adminhosting_east_azurystack_contoso_com_CertRequest_20191219140359.req
Certreq.exe output: CertReq: Request Created

Log location (contains PII):
C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessChecker.log
New-AzsCertificateSigningRequest Completed

```

9. Submit the **.REQ** file generated to your CA (either internal or public). The output directory of **New-AzsCertificateSigningRequest** contains the CSR(s) necessary to submit to a Certificate Authority. The directory also contains, for your reference, a child directory containing the INF file(s) used during certificate request generation. Be sure that your CA generates certificates using your generated request that meet the [Azure Stack Hub PKI Requirements](#).

Next steps

[Prepare Azure Stack Hub PKI certificates](#)

Prepare Azure Stack Hub PKI certificates for deployment or rotation

2 minutes to read • [Edit Online](#)

The certificate files [obtained from your certificate authority \(CA\) of choice](#) must be imported and exported with properties matching Azure Stack Hub's certificate requirements.

Prepare certificates for deployment

Use the following steps to prepare and validate the Azure Stack Hub PKI certificates that will be used for deploying a new Azure Stack Hub environment or for rotating secrets in an existing Azure Stack Hub environment.

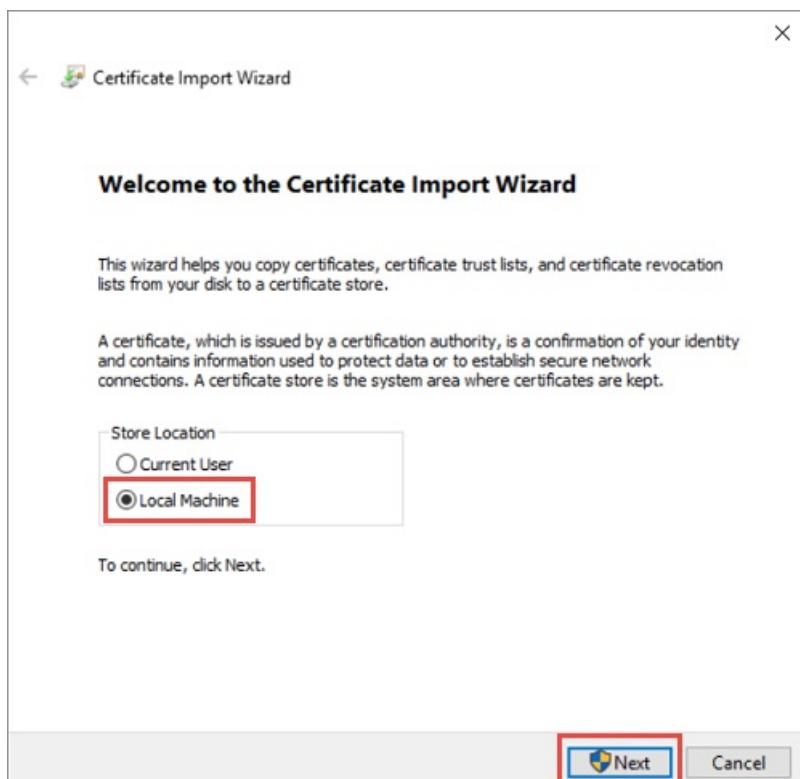
Import the certificate

1. Copy the original certificate versions [obtained from your CA of choice](#) into a directory on the deployment host.

WARNING

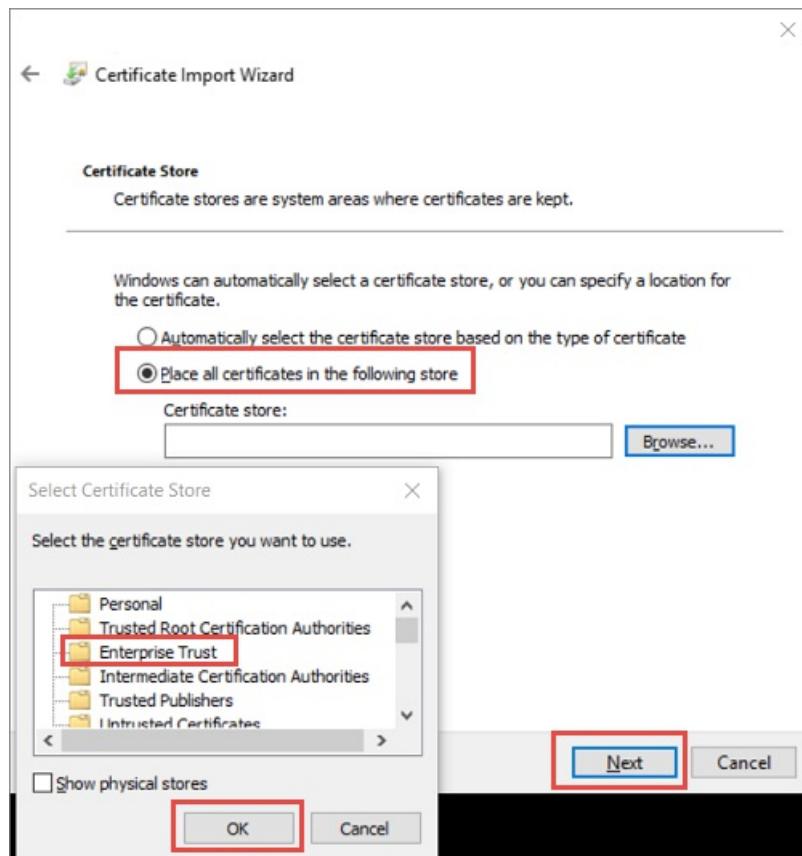
Don't copy files that have already been imported, exported, or altered in any way from the files provided directly by the CA.

2. Right-click on the certificate and select **Install Certificate** or **Install PFX**, depending on how the certificate was delivered from your CA.
3. In the **Certificate Import Wizard**, select **Local Machine** as the import location. Select **Next**. On the following screen, select next again.

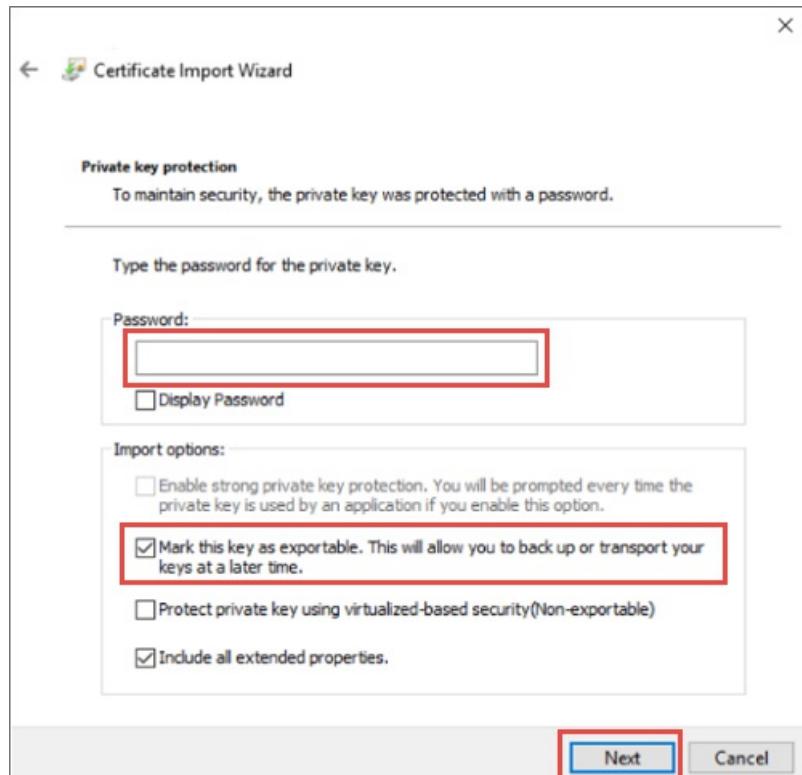


4. Choose **Place all certificate in the following store** and then select **Enterprise Trust** as the location.

Select **OK** to close the certificate store selection dialog box and then select **Next**.



a. If you're importing a PFX, you'll be presented with an additional dialog. On the **Private key protection** page, enter the password for your certificate files and then enable the **Mark this key as exportable. This allows you to back up or transport your keys at a later time** option. Select **Next**.



5. Select **Finish** to complete the import.

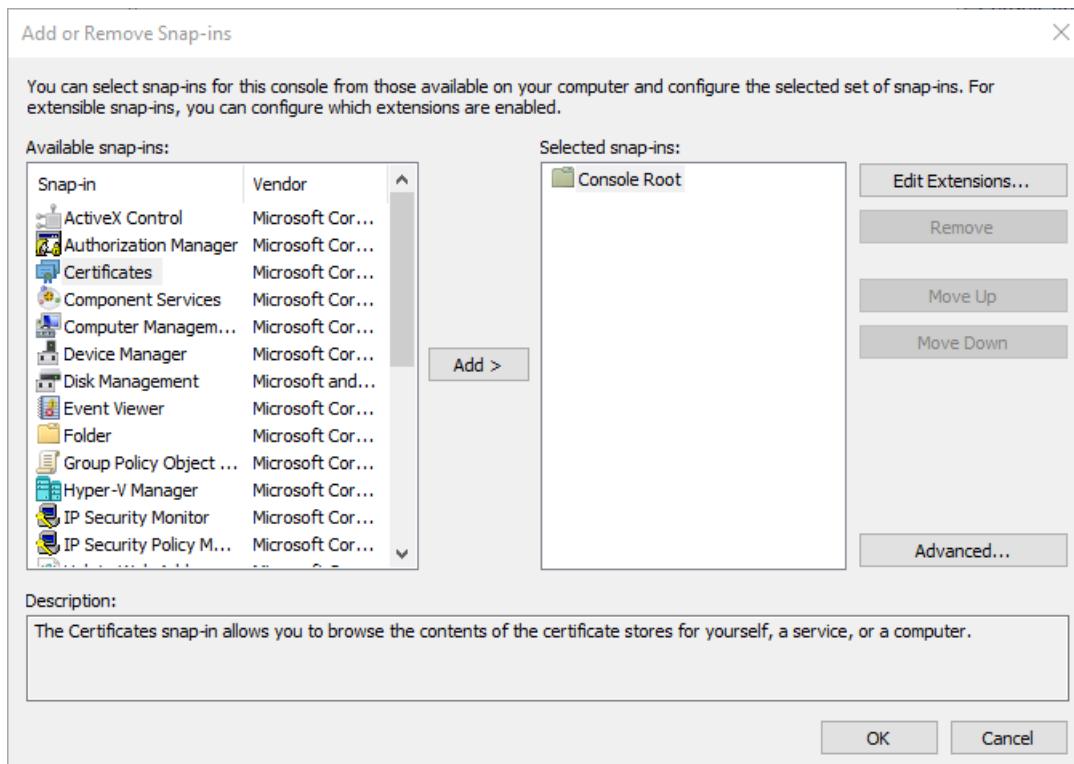
NOTE

After you import a certificate for Azure Stack Hub, the private key of the certificate is stored as a PKCS 12 file (PFX) on clustered storage.

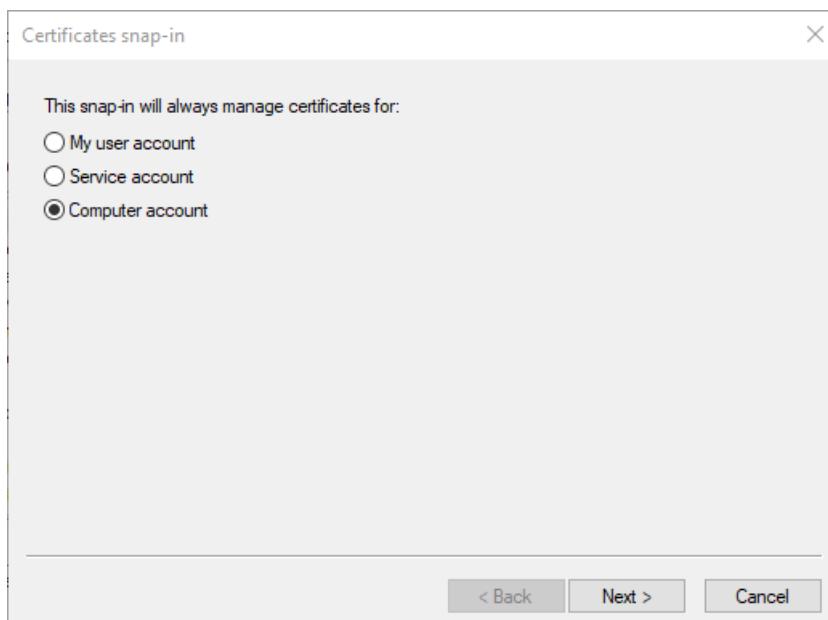
Export the certificate

Open Certificate Manager MMC console and connect to the Local Machine certificate store.

1. Open the Microsoft Management Console. To open the console in Windows 10, right click on the **Start Menu**, select **Run**, then type **mmc** and press enter.
2. Select **File > Add/Remove Snap-In**, then select **Certificates** and select **Add**.



3. Select **Computer account**, then select **Next**. Select **Local computer** and then **Finish**. Select **OK** to close the Add/Remove Snap-In page.



4. Browse to **Certificates > Enterprise Trust > Certificate location**. Verify that you see your certificate on

the right.

- From the Certificate Manager Console toolbar, select **Actions** > **All Tasks** > **Export**. Select **Next**.

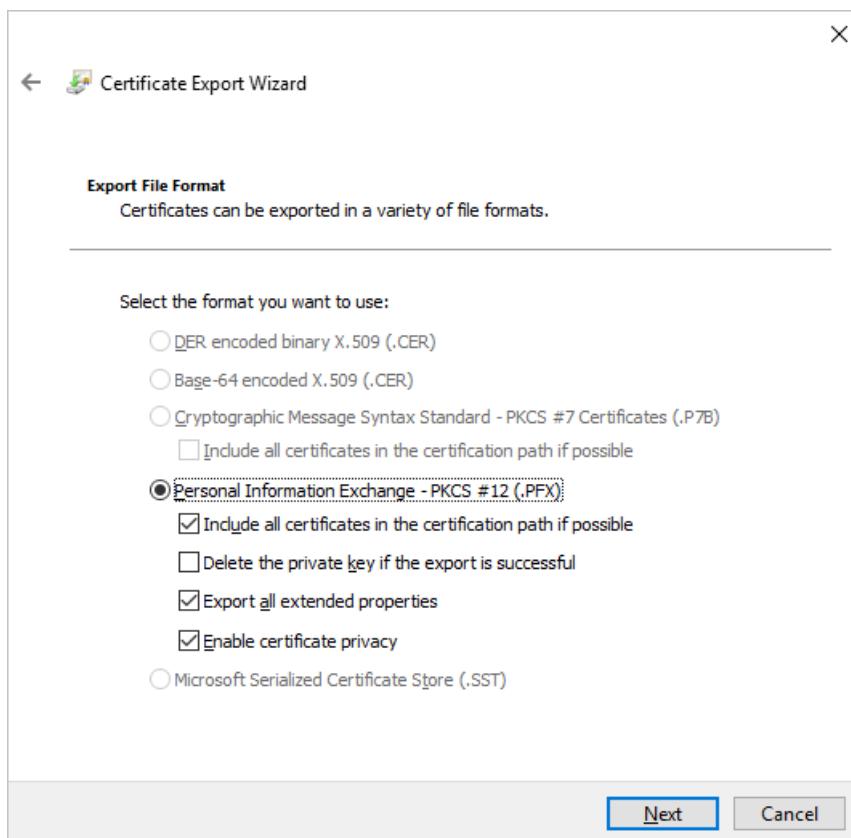
NOTE

Depending on how many Azure Stack Hub certificates you have, you may need to complete this process more than once.

- Select **Yes, Export the Private Key**, and then click **Next**.

- In the Export File Format section:

- Select **Include all certificates in the certificate if possible**.
- Select **Export all Extended Properties**.
- Select **Enable certificate privacy**.
- Click **Next**.



- Select **Password** and provide a password for the certificates. Create a password that meets the following password complexity requirements:

- A minimum length of eight characters.
- At least three of the following: uppercase letter, lowercase letter, numbers from 0-9, special characters, alphabetical character that's not uppercase or lowercase.

Make note of this password. You'll use it as a deployment parameter.

- Select **Next**.

- Choose a file name and location for the PFX file to export. Select **Next**.

- Select **Finish**.

Next steps

[Validate PKI certificates](#)

Validate Azure Stack Hub PKI certificates

5 minutes to read • [Edit Online](#)

The Azure Stack Hub Readiness Checker tool described in this article is available [from the PowerShell Gallery](#). Use the tool to validate that [generated public key infrastructure \(PKI\) certificates](#) are suitable for pre-deployment. Validate certificates by leaving enough time to test and reissue certificates if necessary.

The Readiness Checker tool performs the following certificate validations:

- **Parse PFX**

Checks for valid PFX file, correct password, and whether the public information is protected by the password.

- **Expiry Date**

Checks for minimum validity of seven days.

- **Signature algorithm**

Checks that the signature algorithm isn't SHA1.

- **Private Key**

Checks that the private key is present and is exported with the local machine attribute.

- **Cert chain**

Checks certificate chain is intact including a check for self-signed certificates.

- **DNS names**

Checks the SAN contains relevant DNS names for each endpoint or if a supporting wildcard is present.

- **Key usage**

Checks if the key usage contains a digital signature and key encipherment and checks if enhanced key usage contains server authentication and client authentication.

- **Key size**

Checks if the key size is 2048 or larger.

- **Chain order**

Checks the order of the other certificates validating that the order is correct.

- **Other certificates**

Ensure no other certificates have been packaged in PFX other than the relevant leaf certificate and its chain.

IMPORTANT

The PKI certificate is a PFX file and password should be treated as sensitive information.

Prerequisites

Your system should meet the following prerequisites before validating PKI certificates for an Azure Stack Hub deployment:

- Microsoft Azure Stack Hub Readiness Checker.
- SSL Certificate(s) exported following the [preparation instructions](#).
- DeploymentData.json.
- Windows 10 or Windows Server 2016.

Perform core services certificate validation

Use these steps to prepare and to validate the Azure Stack Hub PKI certificates for deployment and secret rotation:

1. Install **AzsReadinessChecker** from a PowerShell prompt (5.1 or above) by running the following cmdlet:

```
Install-Module Microsoft.AzureStack.ReadinessChecker -force
```

2. Create the certificate directory structure. In the example below, you can change `<C:\Certificates\Deployment>` to a new directory path of your choice.

```
New-Item C:\Certificates\Deployment -ItemType Directory

$directories = 'ACSBlob', 'ACSQueue', 'ACSTable', 'Admin Extension Host', 'Admin Portal', 'ARM Admin',
'ARM Public', 'KeyVault', 'KeyVaultInternal', 'Public Extension Host', 'Public Portal'

$destination = 'C:\Certificates\Deployment'

$directories | % { New-Item -Path (Join-Path $destination $PSITEM) -ItemType Directory -Force}
```

NOTE

AD FS and Graph are required if you're using AD FS as your identity system. For example:

```
$directories = 'ACSBlob', 'ACSQueue', 'ACSTable', 'ADFS', 'Admin Extension Host', 'Admin Portal',
'ARM Admin', 'ARM Public', 'Graph', 'KeyVault', 'KeyVaultInternal', 'Public Extension Host', 'Public
Portal'
```

- Place your certificate(s) in the appropriate directories created in the previous step. For example:

- `C:\Certificates\Deployment\ACSBlob\CustomerCertificate.pfx`
- `C:\Certificates\Deployment\Admin Portal\CustomerCertificate.pfx`
- `C:\Certificates\Deployment\ARM Admin\CustomerCertificate.pfx`

3. In the PowerShell window, change the values of `RegionName` and `FQDN` appropriate to the Azure Stack Hub environment and run the following cmdlet:

```
$pfxPassword = Read-Host -Prompt "Enter PFX Password" -AsSecureString
Invoke-AzsCertificateValidation -CertificateType Deployment -CertificatePath C:\Certificates\Deployment
-pfxPassword $pfxPassword -RegionName east -FQDN azurestack.contoso.com
```

4. Check the output and ensure that all certificates pass all tests. For example:

```

Invoke-AzsCertificateValidation v1.1912.1082.37 started.
Testing: KeyVaultInternal\adminvault.pfx
Thumbprint: B1CB76*****565B99
    Expiry Date: OK
    Signature Algorithm: OK
    DNS Names: OK
    Key Usage: OK
    Key Length: OK
    Parse PFX: OK
    Private Key: OK
    Cert Chain: OK
    Chain Order: OK
    Other Certificates: OK
Testing: ARM Public\management.pfx
Thumbprint: 44A35E*****36052A
    Expiry Date: OK
    Signature Algorithm: OK
    DNS Names: OK
    Key Usage: OK
    Key Length: OK
    Parse PFX: OK
    Private Key: OK
    Cert Chain: OK
    Chain Order: OK
    Other Certificates: OK
Testing: Admin Portal\adminportal.pfx
Thumbprint: 3F5E81*****9EBF9A
    Expiry Date: OK
    Signature Algorithm: OK
    DNS Names: OK
    Key Usage: OK
    Key Length: OK
    Parse PFX: OK
    Private Key: OK
    Cert Chain: OK
    Chain Order: OK
    Other Certificates: OK
Testing: Public Portal\portal.pfx

Log location (contains PII):
C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessChecker.log
Report location (contains PII):
C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessCheckerReport.json
Invoke-AzsCertificateValidation Completed

```

To validate certificates for other Azure Stack Hub services, change the value for `-CertificateType`. For example:

```

# App Services
Invoke-AzsCertificateValidation -CertificateType AppServices -CertificatePath
C:\Certificates\AppServices -pfxPassword $pfxPassword -RegionName east -FQDN azurestack.contoso.com

# DBAdapter
Invoke-AzsCertificateValidation -CertificateType DBAdapter -CertificatePath C:\Certificates\DBAdapter -
pfxPassword $pfxPassword -RegionName east -FQDN azurestack.contoso.com

# EventHubs
Invoke-AzsCertificateValidation -CertificateType EventHubs -CertificatePath C:\Certificates\EventHubs -
pfxPassword $pfxPassword -RegionName east -FQDN azurestack.contoso.com

# IoTHub
Invoke-AzsCertificateValidation -CertificateType IoTHub -CertificatePath C:\Certificates\IoTHub -
pfxPassword $pfxPassword -RegionName east -FQDN azurestack.contoso.com

```

Each folder should contain a single PFX file for the certificate type. If a certificate type has multi-certificate requirements, nested folders for each individual certificate are expected and name-sensitive. The following code shows an example folder/certificate structure for all certificate types, and the appropriate value for `-CertificateType` and `-CertificatePath`.

```
C:\>tree c:\SecretStore /A /F
Folder PATH listing
Volume serial number is 85AE-DF2E
C:\SECRETSTORE
\---AzureStack
    +---CertificateRequests
        \---Certificates
            +---AppServices      # Invoke-AzsCertificateValidation `
|           |     ---API      #   -CertificateType AppServices `-
|           |           api.pfx #       -CertificatePath C:\Certificates\AppServices
|
|           |
|           +---DefaultDomain
|           |           wappsvc.pfx
|
|           |
|           +---Identity
|           |           sso.pfx
|
|           \---Publishing
|                   ftp.pfx
|
+---DBAdapter      # Invoke-AzsCertificateValidation `
|       dbadapter.pfx #   -CertificateType DBAdapter `-
|                           #   -CertificatePath C:\Certificates\DBAdapter
|
+---Deployment      # Invoke-AzsCertificateValidation `
|   +---ACSBlob      #   -CertificateType Deployment `-
|   |       acsblob.pfx #       -CertificatePath C:\Certificates\Deployment
|
|   +---ACSQueue
|       |           acsqueue.pfx
.
./.. ./. ./. ./. ./. <- Deployment certificate tree trimmed.
|   \---Public Portal
|           portal.pfx
|
+---EventHubs      # Invoke-AzsCertificateValidation `
|       eventhubs.pfx #   -CertificateType EventHubs `-
|                           #   -CertificatePath C:\Certificates\EventHubs
|
\---IoTHub          # Invoke-AzsCertificateValidation `
|       iothub.pfx #   -CertificateType IoTHub `-
|                           #   -CertificatePath C:\Certificates\IoTHub
```

Known issues

Symptom: Tests are skipped

Cause: AzsReadinessChecker skips certain tests if a dependency isn't met:

- Other certificates are skipped if certificate chain fails.

```

Testing: ACSBlob\singlewildcard.pfx
  Read PFX: OK
  Signature Algorithm: OK
  Private Key: OK
  Cert Chain: OK
  DNS Names: Fail
  Key Usage: OK
  Key Size: OK
  Chain Order: OK
  Other Certificates: Skipped
Details:
The certificate records '*.east.azurestack.contoso.com' do not contain a record that is valid for
'*blob.east.azurestack.contoso.com'. Please refer to the documentation for how to create the required
certificate file.
The Other Certificates check was skipped because Cert Chain and/or DNS Names failed. Follow the
guidance to remediate those issues and recheck.

Log location (contains PII):
C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessChecker.log
Report location (contains PII):
C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessCheckerReport.json
Invoke-AzsCertificateValidation Completed

```

Resolution: Follow the tool's guidance in the details section under each set of tests for each certificate.

Certificates

DIRECTORY	CERTIFICATE
ACSBlob	wildcard_blob_<region>_<externalFQDN>
ACSQueue	wildcard_queue_<region>_<externalFQDN>
ACSTable	wildcard_table_<region>_<externalFQDN>
Admin Extension Host	wildcard_adminhosting_<region>_<externalFQDN>
Admin Portal	adminportal_<region>_<externalFQDN>
ARM Admin	adminmanagement_<region>_<externalFQDN>
ARM Public	management_<region>_<externalFQDN>
KeyVault	wildcard_vault_<region>_<externalFQDN>
KeyVaultInternal	wildcard_adminvault_<region>_<externalFQDN>
Public Extension Host	wildcard_hosting_<region>_<externalFQDN>
Public Portal	portal_<region>_<externalFQDN>

Using validated certificates

Once your certificates are validated by the AzsReadinessChecker, you're ready to use them in your Azure Stack Hub deployment or for Azure Stack Hub secret rotation.

- For deployment, securely transfer your certificates to your deployment engineer so that they can copy them onto the deployment host as specified in the [Azure Stack Hub PKI requirements documentation](#).
- For secret rotation, you can use the certificates to update old certificates for your Azure Stack Hub environment's public infrastructure endpoints by following the [Azure Stack Hub Secret Rotation documentation](#).
- For PaaS services, you can use the certificates to install SQL, MySQL, and App Services Resource Providers in Azure Stack Hub by following the [Overview of offering services in Azure Stack Hub documentation](#).

Next steps

[Datacenter identity integration](#)

Fix common issues with Azure Stack Hub PKI certificates

4 minutes to read • [Edit Online](#)

The information in this article helps you understand and resolve common issues with Azure Stack Hub PKI certificates. You can discover issues when you use the Azure Stack Hub Readiness Checker tool to [validate Azure Stack Hub PKI certificates](#). The tool checks if the certificates meet the PKI requirements of an Azure Stack Hub deployment and Azure Stack Hub secret rotation, and then logs the results to a [report.json file](#).

PKX Encryption

Issue - PKX encryption isn't TripleDES-SHA1.

Fix - Export PKX files with **TripleDES-SHA1** encryption. This is the default encryption for all Windows 10 clients when exporting from certificate snap-in or using [Export-PFXCertificate](#).

Read PKX

Warning - Password only protects the private information in the certificate.

Fix - Export PKX files with the optional setting for **Enable certificate privacy**.

Issue - PKX file invalid.

Fix - Re-export the certificate using the steps in [Prepare Azure Stack Hub PKI certificates for deployment](#).

Signature algorithm

Issue - Signature algorithm is SHA1.

Fix - Use the steps in Azure Stack Hub certificates signing request generation to regenerate the certificate signing request (CSR) with the signature algorithm of SHA256. Then resubmit the CSR to the certificate authority to reissue the certificate.

Private key

Issue - The private key is missing or doesn't contain the local machine attribute.

Fix - From the computer that generated the CSR, re-export the certificate using the steps in [Prepare Azure Stack Hub PKI certificates for deployment](#). These steps include exporting from the local machine certificate store.

Certificate chain

Issue - Certificate chain isn't complete.

Fix - Certificates should contain a complete certificate chain. Re-export the certificate using the steps in [Prepare Azure Stack Hub PKI certificates for deployment](#) and select the option **Include all certificates in the certification path if possible**.

DNS names

Issue - The **DNSNameList** on the certificate doesn't contain the Azure Stack Hub service endpoint name or a

valid wildcard match. Wildcard matches are only valid for the left-most namespace of the DNS name. For example, `*.region.domain.com` is only valid for `portal.region.domain.com`, not `*.table.region.domain.com`.

Fix - Use the steps in Azure Stack Hub certificates signing request generation to regenerate the CSR with the correct DNS names to support Azure Stack Hub endpoints. Resubmit the CSR to a certificate authority. Then follow the steps in [Prepare Azure Stack Hub PKI certificates for deployment](#) to export the certificate from the machine that generated the CSR.

Key usage

Issue - Key usage is missing digital signature or key encipherment, or enhanced key usage is missing server authentication or client authentication.

Fix - Use the steps in [Azure Stack Hub certificates signing request generation](#) to regenerate the CSR with the correct key usage attributes. Resubmit the CSR to the certificate authority and confirm that a certificate template isn't overwriting the key usage in the request.

Key size

Issue - Key size is smaller than 2048.

Fix - Use the steps in [Azure Stack Hub certificates signing request generation](#) to regenerate the CSR with the correct key length (2048), and then resubmit the CSR to the certificate authority.

Chain order

Issue - The order of the certificate chain is incorrect.

Fix - Re-export the certificate using the steps in [Prepare Azure Stack Hub PKI certificates for deployment](#) and select the option **Include all certificates in the certification path if possible**. Ensure that only the leaf certificate is selected for export.

Other certificates

Issue - The PFX package contains certificates that aren't the leaf certificate or part of the certificate chain.

Fix - Re-export the certificate using the steps in [Prepare Azure Stack Hub PKI certificates for deployment](#), and select the option **Include all certificates in the certification path if possible**. Ensure that only the leaf certificate is selected for export.

Fix common packaging issues

The **AzsReadinessChecker** tool contains a helper cmdlet called **Repair-AzsPfxCertificate**, which can import and then export a PFX file to fix common packaging issues, including:

- **PFX encryption** isn't TripleDES-SHA1.
- **Private key** is missing local machine attribute.
- **Certificate chain** is incomplete or wrong. The local machine must contain the certificate chain if the PFX package doesn't.
- **Other certificates**

Repair-AzsPfxCertificate can't help if you need to generate a new CSR and reissue a certificate.

Prerequisites

The following prerequisites must be in place on the computer on which the tool runs:

- Windows 10 or Windows Server 2016, with internet connectivity.
- PowerShell 5.1 or later. To check your version, run the following PowerShell cmdlet and then review the **Major** and **Minor** versions:

```
$PSVersionTable.PSVersion
```

- Configure [PowerShell for Azure Stack Hub](#).
- Download the latest version of the [Azure Stack Hub readiness checker](#) tool.

Import and export an existing PFX File

1. On a computer that meets the prerequisites, open an elevated PowerShell prompt, and then run the following command to install the Azure Stack Hub readiness checker:

```
Install-Module Microsoft.AzureStack.ReadinessChecker -Force
```

2. From the PowerShell prompt, run the following cmdlet to set the PFX password. Replace `PFXpassword` with the actual password:

```
$password = Read-Host -Prompt PFXpassword -AsSecureString
```

3. From the PowerShell prompt, run the following command to export a new PFX file:

- For `-PfxPath`, specify the path to the PFX file you're working with. In the following example, the path is `.\certificates\ssl.pfx`.
- For `-ExportPFXPath`, specify the location and name of the PFX file for export. In the following example, the path is `.\certificates\ssl_new.pfx`:

```
Repair-AzsPfxCertificate -PfxPassword $password -PfxPath .\certificates\ssl.pfx -ExportPFXPath  
.\\certificates\\ssl_new.pfx
```

4. After the tool completes, review the output for success:

```
Repair-AzsPfxCertificate v1.1809.1005.1 started.  
Starting Azure Stack Hub Certificate Import/Export  
Importing PFX .\\certificates\\ssl.pfx into Local Machine Store  
Exporting certificate to .\\certificates\\ssl_new.pfx  
Export complete. Removing certificate from the local machine store.  
Removal complete.  
Log location (contains PII):  
C:\\Users\\username\\AppData\\Local\\Temp\\AzsReadinessChecker\\AzsReadinessChecker.log  
Repair-AzsPfxCertificate Completed
```

Next steps

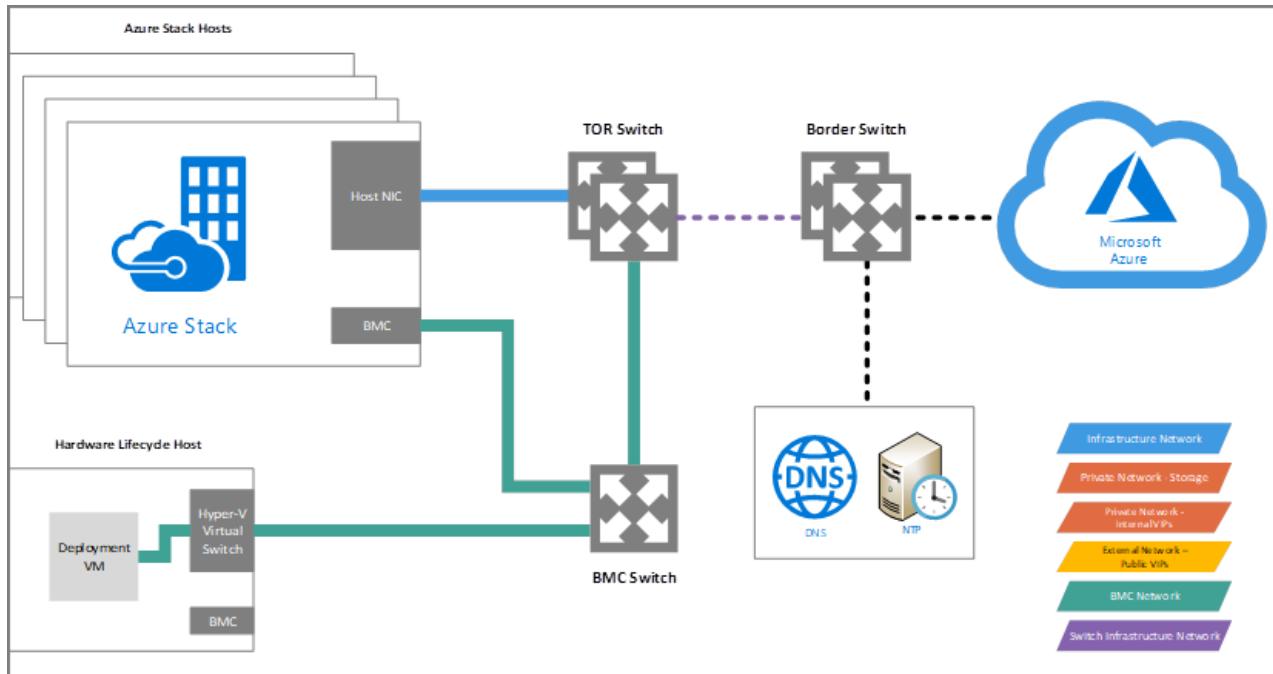
- [Learn more about Azure Stack Hub security](#)

Deployment network traffic

2 minutes to read • [Edit Online](#)

Understanding network traffic during Azure Stack Hub deployment will help make the deployment successful. This article walks you through the network traffic flow during the deployment process so you know what to expect.

This illustration shows all the components and connections involved in the deployment process:



NOTE

This article describes the requirements for a connected deployment. To learn about other deployment methods, see [Azure Stack Hub deployment connection models](#).

The Deployment VM

The Azure Stack Hub solution includes a group of servers that are used to host Azure Stack Hub components and an extra server called the Hardware Lifecycle Host (HLH). This server is used to deploy and manage the lifecycle of your solution and hosts the Deployment VM (DVM) during deployment.

Azure Stack Hub solution providers may provision additional management VMs. Confirm with the solution provider before making any changes to management VMs from a solution provider.

Deployment requirements

Before deployment starts, there are some minimum requirements that can be validated by your OEM to ensure deployment completes successfully:

- [Certificates](#).
- [Azure subscription](#). You may need to check your subscription.
- Internet access.
- DNS.
- NTP.

NOTE

This article focuses on the last three requirements. For more information on the first two, see the links above.

About deployment network traffic

The DVM is configured with an IP from the BMC network and requires network access to the internet. Although not all of the BMC network components require external routing or access to the internet, some OEM-specific components using IPs from this network might also require it.

During deployment, the DVM authenticates against Azure Active Directory (Azure AD) using an Azure account from your subscription. In order to do so, the DVM requires internet access to a list of [specific ports and URLs](#). The DVM will utilize a DNS server to forward DNS requests made by internal components to external URLs. The internal DNS forwards these requests to the DNS forwarder address that you provide to the OEM before deployment. The same is true for the NTP server: a reliable Time Server is required to maintain consistency and time synchronization for all Azure Stack Hub components.

The internet access required by the DVM during deployment is outbound only, no inbound calls are made during deployment. Keep in mind that it uses its IP as source and that Azure Stack Hub doesn't support proxy configurations. Therefore, if necessary, you need to provide a transparent proxy or NAT to access the internet. During deployment, some internal components will start accessing the internet through the external network using public VIPs. After deployment completes, all communication between Azure and Azure Stack Hub is made through the external network using public VIPs.

Network configurations on Azure Stack Hub switches contain access control lists (ACLs) that restrict traffic between certain network sources and destinations. The DVM is the only component with unrestricted access; even the HLH is restricted. You can ask your OEM about customization options to ease management and access from your networks. Because of these ACLs, it's important to avoid changing the DNS and NTP server addresses at deployment time. If you do so, you need to reconfigure all of the switches for the solution.

After deployment is completed, the provided DNS and NTP server addresses will continue to be used by the system's components through the SDN using the external network. For example, if you check DNS requests after deployment is completed, the source will change from the DVM IP to a public VIP.

Next steps

[Validate Azure registration](#)

Validate Azure registration

4 minutes to read • [Edit Online](#)

Use the Azure Stack Hub Readiness Checker tool (**AzsReadinessChecker**) to validate that your Azure subscription is ready to use with Azure Stack Hub before you begin an Azure Stack Hub deployment. The readiness checker validates that:

- The Azure subscription you use is a supported type. Subscriptions must be a Cloud Solution Provider (CSP) or Enterprise Agreement (EA).
- The account you use to register your subscription with Azure can sign in to Azure and is a subscription owner.

For more information about Azure Stack Hub registration, see [Register Azure Stack Hub with Azure](#).

Get the Readiness Checker tool

Download the latest version of **AzsReadinessChecker** from the [PowerShell Gallery](#).

Prerequisites

The following prerequisites are required:

The computer on which the tool runs

- Windows 10 or Windows Server 2016, with internet connectivity.
- PowerShell 5.1 or later. To check your version, run the following PowerShell cmdlet and then review the **Major** and **Minor** versions:

```
$PSVersionTable.PSVersion
```

- [PowerShell configured for Azure Stack Hub](#).
- The latest version of the [Microsoft Azure Stack Hub Readiness Checker](#) tool.

Azure Active Directory (AAD) environment

- Identify the username and password for an account that's an owner for the Azure subscription you'll use with Azure Stack Hub.
- Identify the subscription ID for the Azure subscription you'll use.
- Identify the **AzureEnvironment** you'll use. Supported values for the environment name parameter are **AzureCloud**, **AzureChinaCloud**, or **AzureUSGovernment**, depending on which Azure subscription you're using.

Steps to validate the Azure registration

1. On a computer that meets the prerequisites, open an elevated PowerShell prompt, and then run the following command to install **AzsReadinessChecker**:

```
Install-Module Microsoft.AzureStack.ReadinessChecker -Force
```

2. From the PowerShell prompt, run the following command to set `$registrationCredential` as the account that's the subscription owner. Replace `subscriptionowner@contoso.onmicrosoft.com` with your account and tenant name:

```
$registrationCredential = Get-Credential subscriptionowner@contoso.onmicrosoft.com -Message "Enter  
Credentials for Subscription Owner"
```

NOTE

As a CSP, when using a shared services or IUR subscription, you must provide the credentials of a user from that respective Azure AD. Usually this will be similar to `subscriptionowner@iurcontoso.onmicrosoft.com`. That user must have the appropriate credentials, as described in the previous step.

3. From the PowerShell prompt, run the following command to set `$subscriptionID` as the Azure subscription to use. Replace `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx` with your own subscription ID:

```
$subscriptionID = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
```

4. From the PowerShell prompt, run the following command to start validation of your subscription:

- Specify the value for `AzureEnvironment` as **AzureCloud**, **AzureGermanCloud**, or **AzureChinaCloud**.
- Provide your Azure AD administrator and your Azure AD tenant name.

```
Invoke-AzsRegistrationValidation -RegistrationAccount $registrationCredential -AzureEnvironment  
AzureCloud -RegistrationSubscriptionID $subscriptionID
```

5. After the tool runs, review the output. Confirm the status is correct for both sign-in and the registration requirements. Successful validation output appears similar to the following example:

```
Invoke-AzsRegistrationValidation v1.1809.1005.1 started.  
Checking Registration Requirements: OK  
Log location (contains PII):  
C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessChecker.log  
Report location (contains PII):  
C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessCheckerReport.json  
Invoke-AzsRegistrationValidation Completed
```

Report and log file

Each time validation runs, it logs results to **AzsReadinessChecker.log** and **AzsReadinessCheckerReport.json**. The location of these files displays along with the validation results in PowerShell.

These files can help you share validation status before you deploy Azure Stack Hub or investigate validation problems. Both files persist the results of each subsequent validation check. The report provides your deployment team confirmation of the identity configuration. The log file can help your deployment or support team investigate validation issues.

By default, both files are written to

```
C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessCheckerReport.json .
```

- Use the `-OutputPath <path>` parameter at the end of the run command line to specify a different report location.
- Use the `-CleanReport` parameter at the end of the run command to clear information about previous runs of the tool from **AzsReadinessCheckerReport.json**.

For more information, see [Azure Stack Hub validation report](#).

Validation failures

If a validation check fails, details about the failure display in the PowerShell window. The tool also logs information to the **AzsReadinessChecker.log** file.

The following examples provide more information about common validation failures.

User must be an owner of the subscription

```
Invoke-AzsRegistrationValidation v1.1809.1005.1 started.
Checking Registration Requirements: Fail
Error Details for registration account admin@contoso.onmicrosoft.com:
The user admin@contoso.onmicrosoft.com is role(s) Reader for subscription 3f961d1c-d1fb-40c3-99ba-44524b56df2d. User must be an owner of the subscription to be used for registration.
Additional help URL https://aka.ms/AzsRemediateRegistration

Log location (contains PII): C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessChecker.log
Report location (contains PII):
C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessCheckerReport.json
Invoke-AzsRegistrationValidation Completed
```

Cause - The account is not an administrator of the Azure subscription.

Resolution - Use an account that is an administrator of the Azure subscription that will be billed for usage from the Azure Stack Hub deployment.

Expired or temporary password

```
Invoke-AzsRegistrationValidation v1.1809.1005.1 started.
Checking Registration Requirements: Fail
Error Details for registration account admin@contoso.onmicrosoft.com:
Checking Registration failed with: Retrieving TenantId for subscription [subscription ID] using account admin@contoso.onmicrosoft.com failed with AADSTS50055: Force Change Password.
Trace ID: [Trace ID]
Correlation ID: [Correlation ID]
Timestamp: 2018-10-22 11:16:56Z: The remote server returned an error: (401) Unauthorized.

Log location (contains PII): C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessChecker.log
Report location (contains PII):
C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessCheckerReport.json
Invoke-AzsRegistrationValidation Completed
```

Cause - The account can't sign in because the password is either expired or temporary.

Resolution - In PowerShell, run the following command and follow the prompts to reset the password.

```
Login-AzureRMAccount
```

Another way is to sign in to the [Azure portal](#) as the account owner, and the user will be forced to change the password.

Unknown user type

```
Invoke-AzsRegistrationValidation v1.1809.1005.1 started.  
Checking Registration Requirements: Fail  
Error Details for registration account admin@contoso.onmicrosoft.com:  
Checking Registration failed with: Retrieving TenantId for subscription <subscription ID> using <account>  
failed with unknown_user_type: Unknown User Type  
  
Log location (contains PII): C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessChecker.log  
Report location (contains PII):  
C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessCheckerReport.json  
Invoke-AzsRegistrationValidation Completed
```

Cause - The account can't sign in to the specified Azure AD environment. In this example, **AzureChinaCloud** is specified as the **AzureEnvironment**.

Resolution - Confirm that the account is valid for the specified Azure environment. In PowerShell, run the following command to verify the account is valid for a specific environment:

```
Login-AzureRmAccount -EnvironmentName AzureChinaCloud
```

Next Steps

- [Validate Azure identity](#)
- [View the readiness report](#)
- [General Azure Stack Hub integration considerations](#)

Start-AzsReadinessChecker cmdlet reference

10 minutes to read • [Edit Online](#)

Module: **Microsoft.AzureStack.ReadinessChecker**

This module only contains a single cmdlet. The cmdlet performs one or more pre-deployment or pre-servicing functions for Azure Stack Hub.

Syntax

```
Start-AzsReadinessChecker
    [-CertificatePath <String>]
    -PfxPassword <SecureString>
    -RegionName <String>
    -FQDN <String>
    -IdentitySystem <String>
    [-CleanReport]
    [-OutputPath <String>]
    [-WhatIf]
    [-Confirm]
    [<CommonParameters>]
```

```
Start-AzsReadinessChecker
    [-CertificatePath <String>]
    -PfxPassword <SecureString>
    -DeploymentDataJSONPath <String>
    [-CleanReport]
    [-OutputPath <String>]
    [-WhatIf]
    [-Confirm]
    [<CommonParameters>]
```

```
Start-AzsReadinessChecker
    -PaaSCertificates <Hashtable>
    -DeploymentDataJSONPath <String>
    [-OutputPath <String>]
    [-WhatIf]
    [-Confirm]
    [<CommonParameters>]
```

```
Start-AzsReadinessChecker
    -PaaSCertificates <Hashtable>
    -RegionName <String>
    -FQDN <String>
    -IdentitySystem <String>
    [-OutputPath <String>]
    [-WhatIf]
    [-Confirm]
    [<CommonParameters>]
```

```
Start-AzsReadinessChecker
    -RegionName <String>
    -FQDN <String>
    -IdentitySystem <String>
    -Subject <OrderedDictionary>
    -RequestType <String>
    [-IncludePaaS]
    -OutputRequestPath <String>
    [-OutputPath <String>]
    [-WhatIf]
    [-Confirm]
    [<CommonParameters>]
```

```
Start-AzsReadinessChecker
    -PfxPassword <SecureString>
    -PfxPath <String>
    -ExportPFXPath <String>
    [-OutputPath <String>]
    [-WhatIf]
    [-Confirm]
    [<CommonParameters>]
```

```
Start-AzsReadinessChecker
    -AADServiceAdministrator <PSCredential>
    -ADDirectoryTenantName <String>
    -IdentitySystem <String>
    -AzureEnvironment <String>
    [-CleanReport]
    [-OutputPath <String>]
    [-WhatIf]
    [-Confirm]
    [<CommonParameters>]
```

```
Start-AzsReadinessChecker
    -AADServiceAdministrator <PSCredential>
    -DeploymentDataJSONPath <String>
    [-CleanReport]
    [-OutputPath <String>]
    [-WhatIf]
    [-Confirm]
    [<CommonParameters>]
```

```
Start-AzsReadinessChecker
    -RegistrationAccount <PSCredential>
    -RegistrationSubscriptionID <Guid>
    -AzureEnvironment <String>
    [-CleanReport]
    [-OutputPath <String>]
    [-WhatIf]
    [-Confirm]
    [<CommonParameters>]
```

```
Start-AzsReadinessChecker
    -RegistrationAccount <PSCredential>
    -RegistrationSubscriptionID <Guid>
    -DeploymentDataJSONPath <String>
    [-CleanReport]
    [-OutputPath <String>]
    [-WhatIf]
    [-Confirm]
    [<CommonParameters>]
```

```
Start-AzsReadinessChecker
    -ReportPath <String>
    [-ReportSections <String>]
    [-Summary]
    [-OutputPath <String>]
    [-WhatIf]
    [-Confirm]
    [<CommonParameters>]
```

Description

The **Start-AzsReadinessChecker** cmdlet validates certificates, Azure accounts, Azure subscriptions, and Azure Active Directories (Azure AD). Run validation before deploying Azure Stack Hub, or before Azure Stack Hub servicing actions such as secret rotation. The cmdlet can also be used to generate certificate signing requests for infrastructure certificates, and optionally, PaaS certificates. Finally, the cmdlet can repackage PFX certificates to remediate common packaging issues.

Examples

Example: generate certificate signing request

```
$regionName = 'east'
$externalFQDN = 'azurestack.contoso.com'
$subjectHash = [ordered]@{ "OU"="AzureStack"; "O"="Microsoft"; "L"="Redmond"; "ST"="Washington"; "C"="US" }
Start-AzsReadinessChecker -regionName $regionName -externalFQDN $externalFQDN -subject $subjectHash -
IdentitySystem ADFS -requestType MultipleCSR
```

In this example, `start-AzsReadinessChecker` generates multiple certificate signing requests (CSRs) for certificates suitable for an AD FS Azure Stack Hub deployment with a region name of **east** and an external FQDN of **azurestack.contoso.com**.

Example: validate certificates

```
$password = Read-Host -Prompt "Enter PFX Password" -AsSecureString
Start-AzsReadinessChecker -CertificatePath .\Certificates\ -PfxPassword $password -RegionName east -FQDN
azurestack.contoso.com -IdentitySystem AAD
```

In this example, the PFX password is required for security, and `Start-AzsReadinessChecker` checks the relative folder **Certificates** for certificates valid for an Azure AD deployment with a region name of **east** and an external FQDN of **azurestack.contoso.com**.

Example: validate certificates with deployment data (deployment and support)

```
$password = Read-Host -Prompt "Enter PFX Password" -AsSecureString
Start-AzsReadinessChecker -CertificatePath .\Certificates\ -PfxPassword $password -DeploymentDataJSONPath
.\deploymentdata.json
```

In this deployment and support example, the PFX password is required for security, and `Start-AzsReadinessChecker` checks the relative folder **Certificates** for certificates valid for a deployment where identity, region, and external FQDN are read from the deployment data JSON file generated for deployment.

Example: validate PaaS certificates

```
$PaaSCertificates = @{
    'PaaSDBCert' = @{$pfxPath' = '<Path to DBAdapter PFX>';'pfxPassword' = (ConvertTo-SecureString -String
    '<Password for PFX>' -AsPlainText -Force)}
    'PaaSDefaultCert' = @{$pfxPath' = '<Path to Default PFX>';'pfxPassword' = (ConvertTo-SecureString -String
    '<Password for PFX>' -AsPlainText -Force)}
    'PaaSAPICert' = @{$pfxPath' = '<Path to API PFX>';'pfxPassword' = (ConvertTo-SecureString -String
    '<Password for PFX>' -AsPlainText -Force)}
    'PaaSFTPCert' = @{$pfxPath' = '<Path to FTP PFX>';'pfxPassword' = (ConvertTo-SecureString -String
    '<Password for PFX>' -AsPlainText -Force)}
    'PaaSSSOCert' = @{$pfxPath' = '<Path to SSO PFX>';'pfxPassword' = (ConvertTo-SecureString -String
    '<Password for PFX>' -AsPlainText -Force)}
}
Start-AzsReadinessChecker -PaaSCertificates $PaaSCertificates -RegionName east -FQDN azurestack.contoso.com
```

In this example, a hashtable is constructed with paths and passwords to each PaaS certificate. Certificates can be omitted. `Start-AzsReadinessChecker` checks that each PFX path exists, and validates them using the region **east** and external FQDN **azurestack.contoso.com**.

Example: validate PaaS certificates with deployment data

```
$PaaSCertificates = @{
    'PaaSDBCert' = @{$pfxPath' = '<Path to DBAdapter PFX>';'pfxPassword' = (ConvertTo-SecureString -String
    '<Password for PFX>' -AsPlainText -Force)}
    'PaaSDefaultCert' = @{$pfxPath' = '<Path to Default PFX>';'pfxPassword' = (ConvertTo-SecureString -String
    '<Password for PFX>' -AsPlainText -Force)}
    'PaaSAPICert' = @{$pfxPath' = '<Path to API PFX>';'pfxPassword' = (ConvertTo-SecureString -String
    '<Password for PFX>' -AsPlainText -Force)}
    'PaaSFTPCert' = @{$pfxPath' = '<Path to FTP PFX>';'pfxPassword' = (ConvertTo-SecureString -String
    '<Password for PFX>' -AsPlainText -Force)}
    'PaaSSSOCert' = @{$pfxPath' = '<Path to SSO PFX>';'pfxPassword' = (ConvertTo-SecureString -String
    '<Password for PFX>' -AsPlainText -Force)}
}
Start-AzsReadinessChecker -PaaSCertificates $PaaSCertificates -DeploymentDataJSONPath .\deploymentdata.json
```

In this example, a hashtable is constructed with paths and passwords to each PaaS certificate. Certificates can be omitted. `Start-AzsReadinessChecker` checks that each PFX path exists, and validates them using the region and external FQDN read from the deployment data JSON file generated for deployment.

Example: validate Azure identity

```
$serviceAdminCredential = Get-Credential -Message "Enter Credentials for Service Administrator of Azure Active
Directory Tenant e.g. serviceadmin@contoso.onmicrosoft.com"
# Supported values for the <environment name> parameter are AzureCloud, AzureChinaCloud or AzureUSGovernment
depending which Azure subscription you are using.
Start-AzsReadinessChecker -AADServiceAdministrator $serviceAdminCredential -AzureEnvironment "<environment
name>" -AzureDirectoryTenantName azurestack.contoso.com
```

In this example, the service admin account credentials are required for security, and `Start-AzsReadinessChecker` checks that the Azure account and Azure AD are valid for an Azure AD deployment with a tenant directory name of

[azurestack.contoso.com](#).

Example: validate Azure identity with deployment data (deployment support)

```
$serviceAdminCredential = Get-Credential -Message "Enter Credentials for Service Administrator of Azure Active Directory Tenant e.g. serviceadmin@contoso.onmicrosoft.com"
Start-AzsReadinessChecker -AADServiceAdministrator $serviceAdminCredential -DeploymentDataJSONPath .\contoso-deploymentdata.json
```

In this example, the service admin account credentials are required for security, and `Start-AzsReadinessChecker` checks that the Azure account and Azure AD are valid for an Azure AD deployment, where **AzureCloud** and **TenantName** are read from the deployment data JSON file generated for the deployment.

Example: validate Azure registration

```
$registrationCredential = Get-Credential -Message "Enter Credentials for Subscription Owner e.g. subscriptionowner@contoso.onmicrosoft.com"
$subscriptionID = "<subscription ID>
# Supported values for the <environment name> parameter are AzureCloud, AzureChinaCloud or AzureUSGovernment depending which Azure subscription you are using.
Start-AzsReadinessChecker -RegistrationAccount $registrationCredential -RegistrationSubscriptionID $subscriptionID -AzureEnvironment "<environment name>"
```

In this example, the subscription owner credentials are required for security, and `Start-AzsReadinessChecker` then performs validation against the given account and subscription to ensure it can be used for Azure Stack Hub registration.

Example: validate Azure registration with deployment data (deployment team)

```
$registrationCredential = Get-Credential -Message "Enter Credentials for Subscription Owner e.g. subscriptionowner@contoso.onmicrosoft.com"
$subscriptionID = "<subscription ID>
Start-AzsReadinessChecker -RegistrationAccount $registrationCredential -RegistrationSubscriptionID $subscriptionID -DeploymentDataJSONPath .\contoso-deploymentdata.json
```

In this example, the subscription owner credentials are required for security, and `Start-AzsReadinessChecker` then performs validation against the given account and subscription to ensure it can be used for Azure Stack Hub registration, where additional details are read from the deployment data JSON file generated for the deployment.

Example: import/export PFX package

```
$password = Read-Host -Prompt "Enter PFX Password" -AsSecureString
Start-AzsReadinessChecker -PfxPassword $password -PfxPath .\certificates\ssl.pfx -ExportPFXPath .\certificates\ssl_new.pfx
```

In this example, the PFX password is required for security. The Ssl.pfx file is imported into the local machine certificate store, re-exported with the same password, and saved as Ssl_new.pfx. This procedure is used when certificate validation flags that a private key doesn't have the **Local Machine** attribute set, the certificate chain is broken, irrelevant certificates are present in the PFX, or the certificate chain is in the wrong order.

Example: view validation report (deployment and support)

```
Start-AzsReadinessChecker -ReportPath Contoso-AzsReadinessReport.json
```

In this example, the deployment or support team receives the readiness report from the customer (Contoso), and uses `Start-AzsReadinessChecker` to view the status of the validation executions Contoso performed.

Example: view validation report summary for certificate validation only (deployment and support)

```
Start-AzsReadinessChecker -ReportPath Contoso-AzsReadinessReport.json -ReportSections Certificate -Summary
```

In this example, the deployment or support team receives the readiness report from the customer (Contoso), and uses `Start-AzsReadinessChecker` to view a summarized status of the certificate validation executions Contoso performed.

Required parameters

-RegionName

Specifies the Azure Stack Hub deployment region name.

Type:	String
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-FQDN

Specifies the Azure Stack Hub deployment external FQDN, also aliased as **ExternalFQDN** and **ExternalDomainName**.

Type:	String
Position:	Named
Default value:	ExternalFQDN, ExternalDomainName
Accept pipeline input:	False
Accept wildcard characters:	False

-IdentitySystem

Specifies the Azure Stack Hub deployment identity system valid values, AAD or ADFS, for Azure Active Directory and Active Directory Federated Services, respectively.

Type:	String
Position:	Named
Default value:	None
Valid values:	'AAD','ADFS'

Accept pipeline input:	False
Accept wildcard characters:	False

-PfxPassword

Specifies the password associated with PFX certificate files.

Type:	SecureString
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-PaaSCertificates

Specifies the hashtable containing paths and passwords to PaaS certificates.

Type:	Hashtable
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-DeploymentDataJSONPath

Specifies the Azure Stack Hub deployment data JSON configuration file. This file is generated for deployment.

Type:	String
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-PfxPath

Specifies the path to a problematic certificate that requires an import/export routine to fix, as indicated by certificate validation in this tool.

Type:	String
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-ExportPFXPath

Specifies the destination path for the resultant PFX file from the import/export routine.

Type:	String
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-Subject

Specifies an ordered dictionary of the subject for the certificate request generation.

Type:	OrderedDictionary
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-RequestType

Specifies the SAN type of the certificate request. Valid values are **MultipleCSR**, **SingleCSR**.

- **MultipleCSR** generates multiple certificate requests, one for each service.
- **SingleCSR** generates one certificate request for all services.

Type:	String
Position:	Named

Default value:	None
Valid values:	'MultipleCSR','SingleCSR'
Accept pipeline input:	False
Accept wildcard characters:	False

-OutputRequestPath

Specifies the destination path for certificate request files. Directory must already exist.

Type:	String
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-AADServiceAdministrator

Specifies the Azure AD service admin to be used for Azure Stack Hub deployment.

Type:	PSCredential
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-AADDirectoryName

Specifies the Azure AD name to be used for Azure Stack Hub deployment.

Type:	String
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-AzureEnvironment

Specifies the instance of Azure Services containing the accounts, directories, and subscriptions to be used for Azure Stack Hub deployment and registration.

Type:	String
Position:	Named
Default value:	None
Valid values:	'AzureCloud','AzureChinaCloud','AzureUSGovernment'
Accept pipeline input:	False
Accept wildcard characters:	False

-RegistrationAccount

Specifies the registration account to be used for Azure Stack Hub registration.

Type:	String
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-RegistrationSubscriptionID

Specifies the registration subscription ID to be used for Azure Stack Hub registration.

Type:	Guid
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

-ReportPath

Specifies the path for readiness report, defaults to current directory and default report name.

Type:	String
-------	--------

Position:	Named
Default value:	All
Accept pipeline input:	False
Accept wildcard characters:	False

Optional parameters

-CertificatePath

Specifies the path under which only the certificate required certificate folders are present.

Required folders for Azure Stack Hub deployment with Azure AD identity system are:

- ACSBlob, ACSQueue, ACSTable, Admin Portal, ARM Admin, ARM Public, KeyVault, KeyVaultInternal, Public Portal

Required folders for Azure Stack Hub deployment with Active Directory Federation Services identity system are:

- ACSBlob, ACSQueue, ACSTable, ADFS, Admin Portal, ARM Admin, ARM Public, Graph, KeyVault, KeyVaultInternal, Public Portal

Type:	String
Position:	Named
Default value:	.\\Certificates
Accept pipeline input:	False

Accept wildcard characters:	False
-----------------------------	-------

-IncludePaaS

Specifies whether PaaS services/host names should be added to the certificate request(s).

Type:	SwitchParameter
Position:	Named
Default value:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-ReportSections

Specifies whether to only show report summary, omits detail.

Type:	String
Position:	Named
Default value:	All
Valid values:	'Certificate','AzureRegistration','AzureIdentity','Jobs','All'
Accept pipeline input:	False
Accept wildcard characters:	False

-Summary

Specifies whether to only show report summary, omits detail.

Type:	SwitchParameter
Position:	Named
Default value:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-CleanReport

Removes previous execution and validation history and writes validations to a new report.

Type:	SwitchParameter
Aliases:	cf
Position:	Named
Default value:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-OutputPath

Specifies a custom path to save readiness JSON report and verbose log file. If the path doesn't already exist, the command attempts to create the directory.

Type:	String
-------	--------

Position:	Named
Default value:	\$ENV:TEMP\AzsReadinessChecker
Accept pipeline input:	False
Accept wildcard characters:	False

-Confirm

Prompts for confirmation before running the cmdlet.

Type:	SwitchParameter
Aliases:	cf
Position:	Named
Default value:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-WhatIf

Shows what would happen if the cmdlet runs. The cmdlet isn't run.

Type:	SwitchParameter
Aliases:	wi
Position:	Named
Default value:	False
Accept pipeline input:	False
Accept wildcard characters:	False

Azure Stack Hub validation report

2 minutes to read • [Edit Online](#)

Use the [Azure Stack Hub Readiness Checker tool](#) to run validations that support deployment and servicing of an Azure Stack Hub environment. The tool writes results to a .json report file. The report displays detailed and summarized data about the state of prerequisites for deployment of Azure Stack Hub. The report also displays information about secrets rotation for existing Azure Stack Hub deployments.

Where to find the report

When the tool runs, it logs results to **AzsReadinessCheckerReport.json**. The tool also creates a log named **AzsReadinessChecker.log**. The location of these files displays along with the validation results in PowerShell:

```
Starting Azure Identity Validation
Checking Account(s) can logon: OK
Checking Installation Requirements: OK
Finished Azure Identity Validation
AzsReadinessChecker Log location:
    C:\Users\<username>\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessChecker.log
AzsReadinessChecker Report Location:
    C:\Users\<username>\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessCheckerReport.json
AzsReadinessChecker Completed
```

Both files persist the results of subsequent validation checks when run on the same computer. For example, the tool can be run to validate certificates, run again to validate Azure identity, and then a third time to validate registration. The results of all three validations are available in the resulting .json report.

By default, both files are written to

```
C:\Users\username\AppData\Local\Temp\AzsReadinessChecker\AzsReadinessCheckerReport.json
```

- Use the `-OutputPath <path>` parameter at the end of the command line to specify a different report location.
- Use the `-CleanReport` parameter at the end of the command line to clear information about previous runs of the tool from **AzsReadinessCheckerReport.json**.

View the report

To view the report in PowerShell, supply the path to the report as a value for `-ReportPath`. This command displays the contents of the report and identifies validations that don't yet have results.

For example, to view the report from a PowerShell prompt that's open to the location where the report is located, run the following command:

```
Read-AzsReadinessReport -ReportPath .\AzsReadinessReport.json
```

The output is similar to the following example:

```

Reading All Validation(s) from Report C:\Contoso-AzsReadinessCheckerReport.json

#####
# Certificate Validation Summary #####
#
Certificate Validation results not available.

#####
# Registration Validation Summary #####
#
Azure Registration Validation results not available.

#####
# Azure Identity Results #####
#
Test : ServiceAdministrator
Result : OK
AAD Service Admin : admin@contoso.onmicrosoft.com
Azure Environment : AzureCloud
Azure Active Directory Tenant : contoso.onmicrosoft.com
Error Details :

#####
# Azure Identity Validation Summary #####
#
Azure Identity Validation found no errors or warnings.

#####
# Azure Stack Hub Graph Validation Summary #####
#
Azure Stack Hub Graph Validation results not available.

#####
# Azure Stack Hub ADFS Validation Summary #####
#
Azure Stack Hub ADFS Validation results not available.

#####
# AzsReadiness Job Summary #####
#
Index : 0
Operations :
StartTime : 2018/10/22 14:24:16
EndTime : 2018/10/22 14:24:19
Duration : 3
PSBoundParameters :

```

View the report summary

To view a summary of the report, you can add the `-summary` parameter to the end of the PowerShell command. For example:

```
Read-AzsReadinessReport -ReportPath .\Contoso-AzsReadinessReport.json -summary
```

The summary shows validations that don't have results, and indicates pass or fail for validations that are complete. The output is similar to the following example:

```
Reading All Validation(s) from Report C:\Contoso-AzsReadinessCheckerReport.json

##### Certificate Validation Summary #####
Certificate Validation found no errors or warnings.

##### Registration Validation Summary #####
Registration Validation found no errors or warnings.

##### Azure Identity Validation Summary #####
Azure Identity Validation found no errors or warnings.

##### Azure Stack Hub Graph Validation Summary #####
Azure Stack Hub Graph Validation results not available.

##### Azure Stack Hub ADFS Validation Summary #####
Azure Stack Hub ADFS Validation results not available.
```

View a filtered report

To view a report that is filtered on a single type of validation, use the `-ReportSections` parameter with one of the following values:

- Certificate
- AzureRegistration
- AzureIdentity
- Graph
- ADFS
- Jobs
- All

For example, to view the report summary for certificates only, use the following PowerShell command line:

```
Read-AzsReadinessReport -ReportPath .\Contoso-AzsReadinessReport.json -ReportSections Certificate - Summary
```

Integrate external monitoring solution with Azure Stack Hub

6 minutes to read • [Edit Online](#)

For external monitoring of the Azure Stack Hub infrastructure, you need to monitor the Azure Stack Hub software, the physical computers, and the physical network switches. Each of these areas offers a method to retrieve health and alert information:

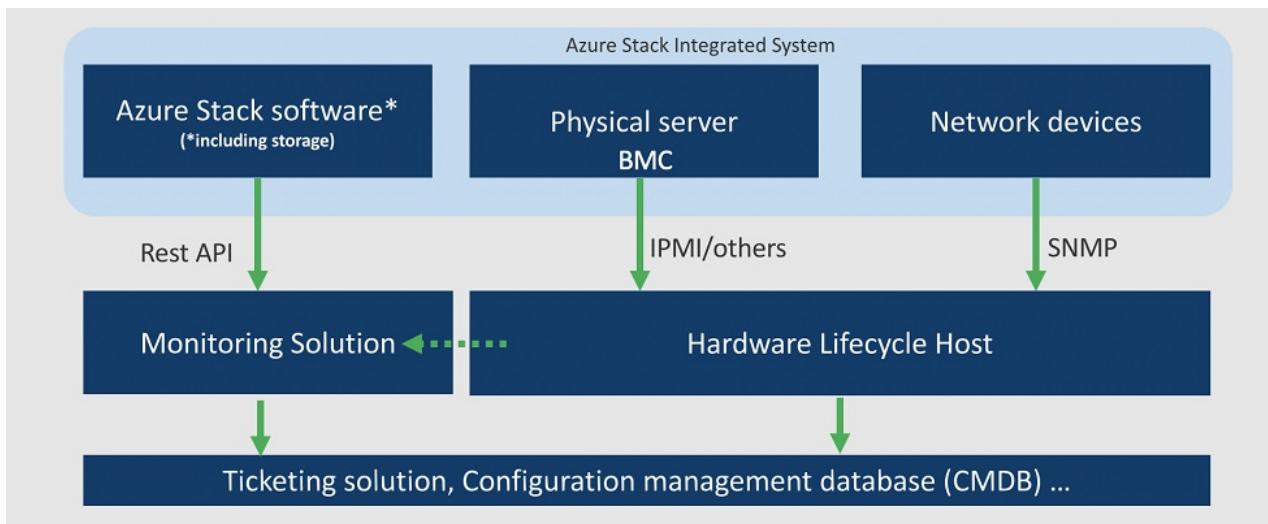
- Azure Stack Hub software offers a REST-based API to retrieve health and alerts. The use of software-defined technologies such as Storage Spaces Direct, storage health, and alerts are part of software monitoring.
- Physical computers can make health and alert information available via the baseboard management controllers (BMCs).
- Physical network devices can make health and alert information available via the SNMP protocol.

Each Azure Stack Hub solution ships with a hardware lifecycle host. This host runs the original equipment manufacturer (OEM) hardware vendor's monitoring software for the physical servers and network devices. Check with your OEM provider if their monitoring solutions can integrate with existing monitoring solutions in your datacenter.

IMPORTANT

The external monitoring solution you use must be agentless. You can't install third-party agents inside Azure Stack Hub components.

The following diagram shows traffic flow between an Azure Stack Hub integrated system, the hardware lifecycle host, an external monitoring solution, and an external ticketing/data collection system.



NOTE

External monitoring integration directly with physical servers isn't allowed and actively blocked by Access Control Lists (ACLs). External monitoring integration directly with physical network devices is supported. Check with your OEM provider on how to enable this feature.

Operations Manager and Nagios. It also includes how to work with alerts programmatically by using PowerShell or through REST API calls.

Integrate with Operations Manager

You can use Operations Manager for external monitoring of Azure Stack Hub. The System Center Management Pack for Microsoft Azure Stack Hub enables you to monitor multiple Azure Stack Hub deployments with a single Operations Manager instance. The management pack uses the health resource provider and update resource provider REST APIs to communicate with Azure Stack Hub. If you plan to bypass the OEM monitoring software that's running on the hardware lifecycle host, you can install vendor management packs to monitor physical servers. You can also use Operations Manager network device discovery to monitor network switches.

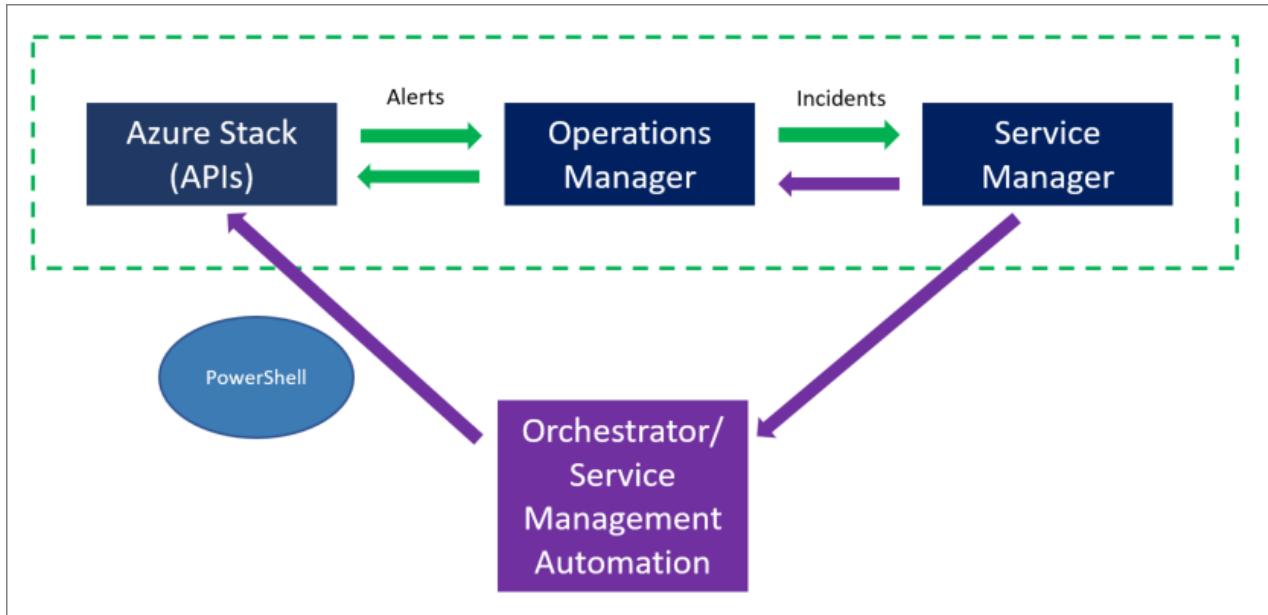
The management pack for Azure Stack Hub provides the following capabilities:

- You can manage multiple Azure Stack Hub deployments.
- There's support for Azure Active Directory (Azure AD) and Active Directory Federation Services (AD FS).
- You can retrieve and close alerts.
- There's a health and a capacity dashboard.
- Includes Auto Maintenance Mode detection for when patch and update (P&U) is in progress.
- Includes Force Update tasks for deployment and region.
- You can add custom information to a region.
- Supports notification and reporting.

To download the System Center Management Pack and the associated user guide, see [Download System Center Management Pack for Microsoft Azure Stack Hub](#). You can also download it directly from Operations Manager.

For a ticketing solution, you can integrate Operations Manager with System Center Service Manager. The integrated product connector enables bidirectional communication that allows you to close an alert in Azure Stack Hub and Operations Manager after you resolve a service request in Service Manager.

The following diagram shows integration of Azure Stack Hub with an existing System Center deployment. You can automate Service Manager further with System Center Orchestrator or Service Management Automation (SMA) to run operations in Azure Stack Hub.



Integrate with Nagios

You can set up and configure the Nagios Plugin for Microsoft Azure Stack Hub.

A Nagios monitoring plugin was developed together with partner Cloudbase Solutions, which is available under the permissive free software license - MIT (Massachusetts Institute of Technology).

The plugin is written in Python and leverages the health resource provider REST API. It offers basic functionality to retrieve and close alerts in Azure Stack Hub. Like the System Center management pack, it enables you to add multiple Azure Stack Hub deployments and to send notifications.

With Version 1.2 the Azure Stack Hub – Nagios plugin leverages the Microsoft ADAL library and supports authentication using Service Principal with a secret or certificate. Also, the configuration has been simplified using a single configuration file with new parameters. It now supports Azure Stack Hub deployments using Azure AD and AD FS as the identity system.

The plugin works with Nagios 4x and XI. To download the plugin, see [Monitoring Azure Stack Hub Alerts](#). The download site also includes installation and configuration details.

Requirements for Nagios

1. Minimum Nagios Version is 4.x
2. Microsoft Azure Active Directory Python library. This library can be installed using Python PIP.

```
sudo pip install adal pyyaml six
```

Install plugin

This section describes how to install the Azure Stack Hub plugin assuming a default installation of Nagios.

The plugin package contains the following files:

```
azurestack_plugin.py  
azurestack_handler.sh  
samples/etc/azurestack.cfg  
samples/etc/azurestack_commands.cfg  
samples/etc/azurestack_contacts.cfg  
samples/etc/azurestack_hosts.cfg  
samples/etc/azurestack_services.cfg
```

1. Copy the plugin `azurestack_plugin.py` into the following directory: `/usr/local/nagios/libexec`.

2. Copy the handler `azurestack_handler.sh` into the following directory:
`/usr/local/nagios/libexec/eventhandlers`.

3. Make sure the plugin file is set to be executable:

```
sudo cp azurestack_plugin.py <PLUGINS_DIR>  
sudo chmod +x <PLUGINS_DIR>/azurestack_plugin.py
```

Configure plugin

The following parameters are available to be configured in the `azurestack.cfg` file. Parameters in bold need to be configured independently from the authentication model you choose.

For more information on how to create an SPN, see [Use an app identity to access resources](#).

PARAMETER	DESCRIPTION	AUTHENTICATION
<code>**External_domain_fqdn **</code>	External Domain FQDN	

PARAMETER	DESCRIPTION	AUTHENTICATION
**region: **	Region Name	
**tenant_id: **	Tenant ID*	
client_id:	Client ID	SPN with secret
client_secret:	Client Password	SPN with secret
client_cert**:	Path to Certificate	SPN with certificate
client_cert_thumbprint**:	Certificate Thumbprint	SPN with certificate

*Tenant ID isn't required for Azure Stack Hub deployments with AD FS.

** Client secret and client cert are mutually exclusive.

The other configuration files contain optional configuration settings as they can be configured in Nagios as well.

NOTE

Check the location destination in azurestack_hosts.cfg and azurestack_services.cfg.

CONFIGURATION	DESCRIPTION
azurestack_commands.cfg	Handler configuration no changes requirement
azurestack_contacts.cfg	Notification Settings
azurestack_hosts.cfg	Azure Stack Hub Deployment Naming
azurestack_services.cfg	Configuration of the Service

Setup steps

1. Modify the configuration file.
2. Copy the modified configuration files into the following folder: `/usr/local/nagios/etc/objects`.

Update Nagios configuration

The Nagios configuration needs to be updated to ensure the Azure Stack Hub – Nagios Plugin is loaded.

1. Open the following file:

```
/usr/local/nagios/etc/nagios.cfg
```

2. Add the following entry:

```
# Load the Azure Stack Hub Plugin Configuration
cfg_file=/usr/local/Nagios/etc/objects/azurestack_contacts.cfg
cfg_file=/usr/local/Nagios/etc/objects/azurestack_commands.cfg
cfg_file=/usr/local/Nagios/etc/objects/azurestack_hosts.cfg
cfg_file=/usr/local/Nagios/etc/objects/azurestack_services.cfg
```

3. Reload Nagios.

```
sudo service nagios reload
```

Manually close active alerts

Active alerts can be closed within Nagios using the custom notification functionality. The custom notification must be:

```
/close-alert <ALERT_GUID>
```

An alert can also be closed using a terminal with the following command:

```
/usr/local/nagios/libexec/azurestack_plugin.py --config-file /usr/local/nagios/etc/objects/azurestack.cfg --action Close --alert-id <ALERT_GUID>
```

Troubleshooting

Troubleshooting the plugin is done by calling the plugin manually in a terminal. Use the following method:

```
/usr/local/nagios/libexec/azurestack_plugin.py --config-file /usr/local/nagios/etc/objects/azurestack.cfg --action Monitor
```

Use PowerShell to monitor health and alerts

If you're not using Operations Manager, Nagios, or a Nagios-based solution, you can use PowerShell to enable a broad range of monitoring solutions to integrate with Azure Stack Hub.

1. To use PowerShell, make sure that you have [PowerShell installed and configured](#) for an Azure Stack Hub operator environment. Install PowerShell on a local computer that can reach the Resource Manager (administrator) endpoint ([https://adminmanagement.\[region\].\[External_FQDN\]](https://adminmanagement.[region].[External_FQDN])).
2. Run the following commands to connect to the Azure Stack Hub environment as an Azure Stack Hub operator:

```
Add-AzureRMEEnvironment -Name "AzureStackAdmin" -ArmEndpoint https://adminmanagement.[Region].  
[External_FQDN] `  
-AzureKeyVaultDnsSuffix adminvault.[Region].[External_FQDN] `  
-AzureKeyVaultServiceEndpointResourceId https://adminvault.[Region].[External_FQDN]  
  
Connect-AzureRmAccount -EnvironmentName "AzureStackAdmin"
```

3. Use commands such as the following examples to work with alerts:

```
# Retrieve all alerts
$Alerts = Get-AzsAlert
$Alerts

# Filter for active alerts
$Active = $Alerts | Where-Object { $_.State -eq "active" }
$Active

# Close alert
Close-AzsAlert -AlertID "ID"

#Retrieve resource provider health
$RPHealth = Get-AzsRPHealth
$RPHealth

# Retrieve infrastructure role instance health
$FRPID = $RPHealth | Where-Object { $_.DisplayName -eq "Capacity" }
Get-AzsRegistrationHealth -ServiceRegistrationId $FRPID.RegistrationId
```

Learn more

For information about built-in health monitoring, see [Monitor health and alerts in Azure Stack Hub](#).

Next steps

[Security integration](#)

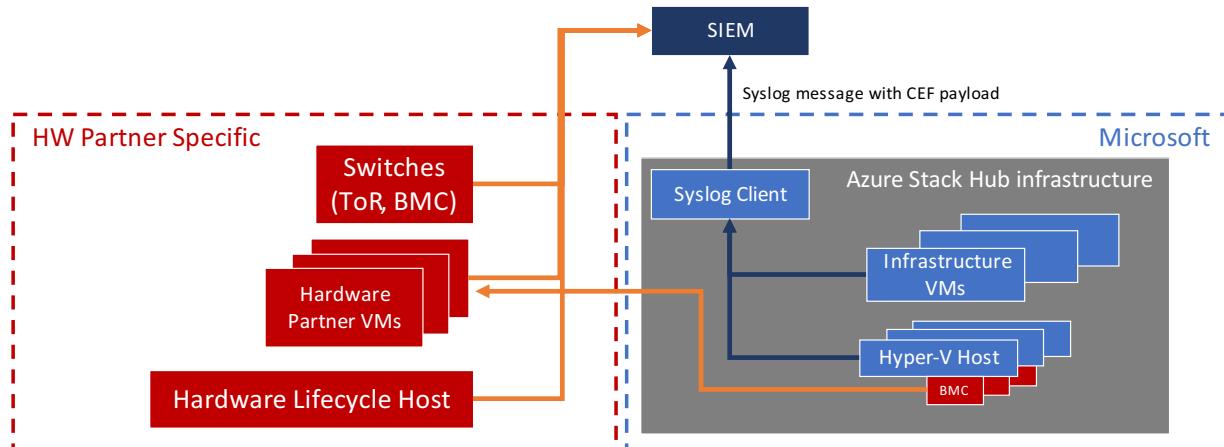
Integrate Azure Stack Hub with monitoring solutions using syslog forwarding

11 minutes to read • [Edit Online](#)

This article shows you how to use syslog to integrate Azure Stack Hub infrastructure with external security solution(s) already deployed in your datacenter. For example, a security information event management (SIEM) system. The syslog channel exposes audits, alerts, and security logs from all the components of the Azure Stack Hub infrastructure. Use syslog forwarding to integrate with security monitoring solutions and to retrieve all audits, alerts, and security logs to store them for retention.

Starting with the 1809 update, Azure Stack Hub has an integrated syslog client that, once configured, emits syslog messages with the payload in Common Event Format (CEF).

The following diagram describes the integration of Azure Stack Hub with an external SIEM. There are two integration patterns that need to be considered: the first one (the one in blue) is the Azure Stack Hub infrastructure that encompasses the infrastructure virtual machines and the Hyper-V nodes. All the audits, security logs, and alerts from those components are centrally collected and exposed via syslog with CEF payload. This integration pattern is described in this document page. The second integration pattern is the one depicted in orange and covers the baseboard management controllers (BMCs), the hardware lifecycle host (HLH), the virtual machines and virtual appliances that run the hardware partner monitoring and management software, and the top of rack (ToR) switches. Since these components are hardware-partner specific, contact your hardware partner for documentation on how to integrate them with an external SIEM.



Configuring syslog forwarding

The syslog client in Azure Stack Hub supports the following configurations:

- Syslog over TCP, with mutual authentication (client and server) and TLS 1.2 encryption:** In this configuration, both the syslog server and the syslog client can verify the identity of each other via certificates. The messages are sent over a TLS 1.2 encrypted channel.
- Syslog over TCP with server authentication and TLS 1.2 encryption:** In this configuration, the syslog client can verify the identity of the syslog server via a certificate. The messages are sent over a TLS 1.2 encrypted channel.
- Syslog over TCP, with no encryption:** In this configuration, the syslog client and syslog server identities aren't verified. The messages are sent in clear text over TCP.

4. **Syslog over UDP, with no encryption:** In this configuration, the syslog client and syslog server identities aren't verified. The messages are sent in clear text over UDP.

IMPORTANT

Microsoft strongly recommends to use TCP using authentication and encryption (configuration #1 or, at the very minimum, #2) for production environments to protect against man-in-the-middle attacks and eavesdropping of messages.

Cmdlets to configure syslog forwarding

Configuring syslog forwarding requires access to the privileged endpoint (PEP). Two PowerShell cmdlets have been added to the PEP to configure the syslog forwarding:

```
### cmdlet to pass the syslog server information to the client and to configure the transport protocol, the encryption and the authentication between the client and the server

Set-SyslogServer [-ServerName <String>] [-ServerPort <UInt16>] [-NoEncryption] [-SkipCertificateCheck] [-SkipCNCheck] [-UseUDP] [-Remove]

### cmdlet to configure the certificate for the syslog client to authenticate with the server

Set-SyslogClient [-pfxBinary <Byte[]>] [-CertPassword <SecureString>] [-RemoveCertificate] [-OutputSeverity]
```

Cmdlets parameters

Parameters for *Set-SyslogServer* cmdlet:

PARAMETER	DESCRIPTION	TYPE	REQUIRED
<i>ServerName</i>	FQDN or IP address of the syslog server.	String	yes
<i>ServerPort</i>	Port number the syslog server is listening on.	UInt16	yes
<i>NoEncryption</i>	Force the client to send syslog messages in clear text.	flag	no
<i>SkipCertificateCheck</i>	Skip validation of the certificate provided by the syslog server during initial TLS handshake.	flag	no
<i>SkipCNCheck</i>	Skip validation of the Common Name value of the certificate provided by the syslog server during initial TLS handshake.	flag	no
<i>UseUDP</i>	Use syslog with UDP as transport protocol.	flag	no
<i>Remove</i>	Remove configuration of the server from the client and stop syslog forwarding.	flag	no

Parameters for *Set-SyslogClient* cmdlet:

PARAMETER	DESCRIPTION	TYPE
<i>pfxBinary</i>	The contents of the pfx file, piped to a Byte[], containing the certificate to be used by the client as identity to authenticate against the syslog server.	Byte[]
<i>CertPassword</i>	Password to import the private key that's associated with the pfx file.	SecureString
<i>RemoveCertificate</i>	Remove certificate from the client.	flag
<i>OutputSeverity</i>	Level of output logging. Values are Default or Verbose . Default includes severity levels: warning, critical, or error. Verbose includes all severity levels: verbose, informational, warning, critical, or error.	String

Configuring syslog forwarding with TCP, mutual authentication, and TLS 1.2 encryption

In this configuration, the syslog client in Azure Stack Hub forwards the messages to the syslog server over TCP, with TLS 1.2 encryption. During the initial handshake, the client verifies that the server provides a valid, trusted certificate. The client also provides a certificate to the server as proof of its identity. This configuration is the most secure as it provides a full validation of the identity of both the client and the server and it sends messages over an encrypted channel.

IMPORTANT

Microsoft strongly recommends to use this configuration for production environments.

To configure syslog forwarding with TCP, mutual authentication, and TLS 1.2 encryption, run both these cmdlets on a PEP session:

```
# Configure the server
Set-SyslogServer -ServerName <FQDN or ip address of syslog server> -ServerPort <Port number on which the
syslog server is listening on>

# Provide certificate to the client to authenticate against the server
Set-SyslogClient -pfxBinary <Byte[] of pfx file> -CertPassword <SecureString, password for accessing the pfx
file>
```

The client certificate must have the same root as the one provided during the deployment of Azure Stack Hub. It also must contain a private key.

```

##Example on how to set your syslog client with the certificate for mutual authentication.
##This example script must be run from your hardware lifecycle host or privileged access workstation.

$ErcsNodeName = "<yourPEP>"
$password = ConvertTo-SecureString -String "<your cloudAdmin account password" -AsPlainText -Force

$cloudAdmin = "<your cloudAdmin account name>"
$CloudAdminCred = New-Object System.Management.Automation.PSCredential ($cloudAdmin, $password)

$certPassword = $password
$certContent = Get-Content -Path C:\cert\<yourClientCertificate>.pfx -Encoding Byte

$params = @{
    ComputerName = $ErcsNodeName
    Credential = $CloudAdminCred
    ConfigurationName = "PrivilegedEndpoint"
}

$session = New-PSSession @params

$params = @{
    Session = $session
    ArgumentList = @($certContent, $certPassword)
}
Write-Verbose "Invoking cmdlet to set syslog client certificate..." -Verbose
Invoke-Command @params -ScriptBlock {
    param($CertContent, $CertPassword)
    Set-SyslogClient -PfxBinary $CertContent -CertPassword $CertPassword }

```

Configuring syslog forwarding with TCP, Server authentication, and TLS 1.2 encryption

In this configuration, the syslog client in Azure Stack Hub forwards the messages to the syslog server over TCP, with TLS 1.2 encryption. During the initial handshake, the client also verifies that the server provides a valid, trusted certificate. This configuration prevents the client from sending messages to untrusted destinations. TCP using authentication and encryption is the default configuration and represents the minimum level of security that Microsoft recommends for a production environment.

```
Set-SyslogServer -ServerName <FQDN or ip address of syslog server> -ServerPort <Port number on which the syslog server is listening on>
```

In case you want to test the integration of your syslog server with the Azure Stack Hub client by using a self-signed or untrusted certificate, you can use these flags to skip the server validation done by the client during the initial handshake.

```

#Skip validation of the Common Name value in the server certificate. Use this flag if you provide an IP address for your syslog server
Set-SyslogServer -ServerName <FQDN or ip address of syslog server> -ServerPort <Port number on which the syslog server is listening on>
-SkipCNCheck

#Skip entirely the server certificate validation
Set-SyslogServer -ServerName <FQDN or ip address of syslog server> -ServerPort <Port number on which the syslog server is listening on>
-SkipCertificateCheck

```

IMPORTANT

Microsoft recommends against the use of -SkipCertificateCheck flag for production environments.

Configuring syslog forwarding with TCP and no encryption

In this configuration, the syslog client in Azure Stack Hub forwards the messages to the syslog server over TCP, with no encryption. The client doesn't verify the identity of the server nor does it provide its own identity to the server for verification.

```
Set-SyslogServer -ServerName <FQDN or ip address of syslog server> -ServerPort <Port number on which the  
syslog server is listening on> -NoEncryption
```

IMPORTANT

Microsoft recommends against using this configuration for production environments.

Configuring syslog forwarding with UDP and no encryption

In this configuration, the syslog client in Azure Stack Hub forwards the messages to the syslog server over UDP, with no encryption. The client doesn't verify the identity of the server nor does it provide its own identity to the server for verification.

```
Set-SyslogServer -ServerName <FQDN or ip address of syslog server> -ServerPort <Port number on which the  
syslog server is listening on> -UseUDP
```

While UDP with no encryption is the easiest to configure, it doesn't provide any protection against man-in-the-middle attacks and eavesdropping of messages.

IMPORTANT

Microsoft recommends against using this configuration for production environments.

Removing syslog forwarding configuration

To remove the syslog server configuration altogether and stop syslog forwarding:

Remove the syslog server configuration from the client

```
Set-SyslogServer -Remove
```

Remove the client certificate from the client

```
Set-SyslogClient -RemoveCertificate
```

Verifying the syslog setup

If you successfully connected the syslog client to your syslog server, you should soon start receiving events. If you don't see any event, verify the configuration of your syslog client by running the following cmdlets:

Verify the server configuration in the syslog client

```
Get-SyslogServer
```

Verify the certificate setup in the syslog client

```
Get-SyslogClient
```

Syslog message schema

The syslog forwarding of the Azure Stack Hub infrastructure sends messages formatted in Common Event Format (CEF). Each syslog message is structured based on this schema:

```
<Time> <Host> <CEF payload>
```

The CEF payload is based on the structure below, but the mapping for each field varies depending on the type of message (Windows Event, Alert created, Alert closed).

```
# Common Event Format schema
CEF: <Version>|<Device Vendor>|<Device Product>|<Device Version>|<Signature ID>|<Name>|<Severity>|<Extensions>
* Version: 0.0
* Device Vendor: Microsoft
* Device Product: Microsoft Azure Stack Hub
* Device Version: 1.0
```

CEF mapping for privileged endpoint events

```
Prefix fields
* Signature ID: Microsoft-AzureStack-PrivilegedEndpoint: <PEP Event ID>
* Name: <PEP Task Name>
* Severity: mapped from PEP Level (details see the PEP Severity table below)
```

Table of events for the privileged endpoint:

EVENT	PEP EVENT ID	PEP TASK NAME	SEVERITY
PrivilegedEndpointAccessed	1000	PrivilegedEndpointAccessedEvent	5
SupportSessionTokenRequested	1001	SupportSessionTokenRequestedEvent	5
SupportSessionDevelopmentTokenRequested	1002	SupportSessionDevelopmentTokenRequestedEvent	5
SupportSessionUnlocked	1003	SupportSessionUnlockedEvent	10
SupportSessionFailedToUnlock	1004	SupportSessionFailedToUnlockEvent	10
PrivilegedEndpointClosed	1005	PrivilegedEndpointClosedEvent	5
NewCloudAdminUser	1006	NewCloudAdminUserEvent	10
RemoveCloudAdminUser	1007	RemoveCloudAdminUserEvent	10

EVENT	PEP EVENT ID	PEP TASK NAME	SEVERITY
SetCloudAdminUserPassword	1008	SetCloudAdminUserPasswordEvent	5
GetCloudAdminPasswordRecoveryToken	1009	GetCloudAdminPasswordRecoveryTokenEvent	10
ResetCloudAdminPassword	1010	ResetCloudAdminPasswordEvent	10
PrivilegedEndpointSessionTimedOut	1017	PrivilegedEndpointSessionTimedOutEvent	5

PEP Severity table:

SEVERITY	LEVEL	NUMERICAL VALUE
0	Undefined	Value: 0. Indicates logs at all levels
10	Critical	Value: 1. Indicates logs for a critical alert
8	Error	Value: 2. Indicates logs for an error
5	Warning	Value: 3. Indicates logs for a warning
2	Information	Value: 4. Indicates logs for an informational message
0	Verbose	Value: 5. Indicates logs at all levels

CEF mapping for recovery endpoint events

Prefix fields

- * Signature ID: Microsoft-AzureStack-PrivilegedEndpoint: <REP Event ID>
- * Name: <REP Task Name>
- * Severity: mapped from REP Level (details see the REP Severity table below)

Table of events for the recovery endpoint:

EVENT	REP EVENT ID	REP TASK NAME	SEVERITY
RecoveryEndpointAccessed	1011	RecoveryEndpointAccessedEvent	5
RecoverySessionTokenRequested	1012	RecoverySessionTokenRequestedEvent	5
RecoverySessionDeveloperTokenRequested	1013	RecoverySessionDeveloperTokenRequestedEvent	5
RecoverySessionUnlocked	1014	RecoverySessionUnlockedEvent	10

EVENT	REP EVENT ID	REP TASK NAME	SEVERITY
RecoverySessionFailedToUnlock	1015	RecoverySessionFailedToUnlockEvent	10
RecoveryEndpointClosed	1016	RecoveryEndpointClosedEvent	5

REP Severity table:

SEVERITY	LEVEL	NUMERICAL VALUE
0	Undefined	Value: 0. Indicates logs at all levels
10	Critical	Value: 1. Indicates logs for a critical alert
8	Error	Value: 2. Indicates logs for an error
5	Warning	Value: 3. Indicates logs for a warning
2	Information	Value: 4. Indicates logs for an informational message
0	Verbose	Value: 5. Indicates logs at all levels

CEF mapping for Windows events

- * Signature ID: ProviderName:EventID
- * Name: TaskName
- * Severity: Level (for details, see the severity table below)
- * Extension: Custom Extension Name (for details, see the Custom Extension table below)

Severity table for Windows events:

CEF SEVERITY VALUE	WINDOWS EVENT LEVEL	NUMERICAL VALUE
0	Undefined	Value: 0. Indicates logs at all levels
10	Critical	Value: 1. Indicates logs for a critical alert
8	Error	Value: 2. Indicates logs for an error
5	Warning	Value: 3. Indicates logs for a warning
2	Information	Value: 4. Indicates logs for an informational message
0	Verbose	Value: 5. Indicates logs at all levels

Custom extension table for Windows events in Azure Stack Hub:

CUSTOM EXTENSION NAME	WINDOWS EVENT EXAMPLE
MasChannel	System

CUSTOM EXTENSION NAME	WINDOWS EVENT EXAMPLE
MasComputer	test.azurestack.contoso.com
MasCorrelationActivityID	C8F40D7C-3764-423B-A4FA-C994442238AF
MasCorrelationRelatedActivityID	C8F40D7C-3764-423B-A4FA-C994442238AF
MasEventData	svchost!!4132,G,0!!!!EseDiskFlushConsistency!!ESENT!!0x800000
MasEventDescription	The Group Policy settings for the user were processed successfully. There were no changes detected since the last successful processing of Group Policy.
MasEventID	1501
MasEventRecordID	26637
MasExecutionProcessID	29380
MasExecutionThreadID	25480
MasKeywords	0x8000000000000000
MasKeywordName	Audit Success
MasLevel	4
MasOpcode	1
MasOpcodeName	info
MasProviderEventSourceName	
MasProviderGuid	AEA1B4FA-97D1-45F2-A64C-4D69FFFD92C9
MasProviderName	Microsoft-Windows-GroupPolicy
MasSecurityUserId	<Windows SID>
MasTask	0
MasTaskCategory	Process Creation
MasUserData	KB4093112!!5112!!Installed!!0x0!!WindowsUpdateAgent Xpath: /Event/UserData/*
MasVersion	0

CEF mapping for alerts created

- * Signature ID: Microsoft Azure Stack Hub Alert Creation : FaultTypeId
- * Name: FaultTypeId : AlertId
- * Severity: Alert Severity (for details, see alerts severity table below)
- * Extension: Custom Extension Name (for details, see the Custom Extension table below)

Alerts severity table:

SEVERITY	LEVEL
0	Undefined
10	Critical
5	Warning

Custom Extension table for Alerts created in Azure Stack Hub:

CUSTOM EXTENSION NAME	EXAMPLE
MasEventDescription	DESCRIPTION: A user account <TestUser> was created for <TestDomain>. It's a potential security risk. -- REMEDIATION: Contact support. Customer Assistance is required to resolve this issue. Don't try to resolve this issue without their assistance. Before you open a support request, start the log file collection process using the guidance from https://aka.ms/azurestacklogfiles .

CEF mapping for alerts closed

- * Signature ID: Microsoft Azure Stack Hub Alert Creation : FaultTypeId
- * Name: FaultTypeId : AlertId
- * Severity: Information

The example below shows a syslog message with CEF payload:

```
2018:05:17:-23:59:28 -07:00 TestHost CEF:0.0|Microsoft|Microsoft Azure Stack Hub|1.0|3|TITLE: User Account Created -- DESCRIPTION: A user account \<TestUser\> was created for \<TestDomain\>. It's a potential security risk. -- REMEDIATION: Please contact Support. Customer Assistance is required to resolve this issue. Do not try to resolve this issue without their assistance. Before you open a support request, start the log file collection process using the guidance from https://aka.ms/azurestacklogfiles|10
```

Next steps

[Servicing policy](#)

Integrate physical device auditing with your Azure Stack Hub datacenter

2 minutes to read • [Edit Online](#)

All physical devices in Azure Stack Hub, like the baseboard management controllers (BMCs) and network switches, emit audit logs. You can integrate the audit logs into your overall auditing solution. Since the devices vary across the different Azure Stack Hub OEM hardware vendors, contact your vendor for the documentation on auditing integration. The sections below provide some general information for physical device auditing in Azure Stack Hub.

Physical device access auditing

All physical devices in Azure Stack Hub support the use of TACACS or RADIUS. Support includes access to the baseboard management controller (BMC) and network switches.

Azure Stack Hub solutions don't ship with either RADIUS or TACACS built-in. However, the solutions have been validated to support the use of existing RADIUS or TACACS solutions available in the market.

For RADIUS only, MSCHAPv2 was validated. This represents the most secure implementation using RADIUS. Consult with your OEM hardware vendor to enable TACAS or RADIUS in the devices included with your Azure Stack Hub solution.

Syslog forwarding for network devices

All physical networking devices in Azure Stack Hub support syslog messages. Azure Stack Hub solutions don't ship with a syslog server. However, the devices have been validated to support sending messages to existing syslog solutions available in the market.

The syslog destination address is an optional parameter collected for deployment, but it can also be added post deployment. Consult with your OEM hardware vendor to configure syslog forwarding on your networking devices.

Next steps

[Servicing policy](#)

Azure Stack Hub administration basics

6 minutes to read • [Edit Online](#)

If you're new to Azure Stack Hub administration, there are several things you need to know. This article provides an overview of your role as an Azure Stack Hub operator, and what you need to tell your users to help them become productive.

Understand the builds

Integrated systems

If you're using an Azure Stack Hub integrated system, update packages distribute updated versions of Azure Stack Hub. You can import these packages and apply them by using the **Updates** tile in the administrator portal.

Development kit

If you're using the Azure Stack Development Kit (ASDK), review [What is Azure Stack Hub?](#) to learn the purpose and limitations of the ASDK. You can use the ASDK as a *sandbox*, where you can evaluate Azure Stack Hub and develop and test your apps in a non-production environment. For deployment information, see [Azure Stack Development Kit deployment](#).

Like Azure, we innovate rapidly. We'll regularly release new builds. If you're running the ASDK and you want to move to the latest build, you must [redeploy Azure Stack Hub](#). You can't apply update packages. This process takes time, but the benefit is that you can try out the latest features. The ASDK documentation on our website reflects the latest release build.

Learn about available services

You'll need an awareness of which services you can make available to your users. Azure Stack Hub supports a subset of Azure services. The list of supported services will continue to evolve.

Foundational services

By default, Azure Stack Hub includes the following "foundational services" when you deploy Azure Stack Hub:

- Compute
- Storage
- Networking
- Key Vault

With these foundational services, you can offer Infrastructure-as-a-Service (IaaS) to your users with minimal configuration.

Additional services

Currently, we support the following additional Platform-as-a-Service (PaaS) services:

- App Service
- Azure Functions
- SQL and MySQL databases
- Kubernetes (in preview)

These services require additional configuration before you can make them available to your users. For more information, see the "Tutorials" and the "How-to guides\Offer services" sections of our Azure Stack Hub operator

documentation.

Service roadmap

Azure Stack Hub will continue to add support for Azure services. For the projected roadmap, see the [Azure Stack Hub: An extension of Azure whitepaper](#). You can also monitor the [Azure Stack Hub blog posts](#) for new announcements.

What account should I use?

There are a few account considerations to be aware of when managing Azure Stack Hub. Especially in deployments using Windows Server Active Directory Federation Services (AD FS) as the identity provider instead of Azure Active Directory (Azure AD). The following account considerations apply to both Azure Stack Hub integrated systems and ASDK deployments:

ACCOUNT	AZURE AD	AD FS
Local Administrator (\Administrator)	ASDK host administrator.	ASDK host administrator.
AzureStack\AzureStackAdmin	ASDK host administrator. Can be used to sign in to the Azure Stack Hub administrator portal. Access to view and administer Service Fabric rings.	ASDK host administrator. No access to the Azure Stack Hub administrator portal. Access to view and administer Service Fabric rings. No longer owner of the Default Provider Subscription (DPS).
AzureStack\CloudAdmin	Can access and run permitted commands within the privileged endpoint.	Can access and run permitted commands within the privileged endpoint. Can't sign in to the ASDK host. Owner of the Default Provider Subscription (DPS).
Azure AD Global Administrator	Used during installation. Owner of the Default Provider Subscription (DPS).	Not applicable.

What tools do I use to manage?

You can use the [administrator portal](#) or PowerShell to manage Azure Stack Hub. The easiest way to learn the basic concepts is through the portal. If you want to use PowerShell, there are preparation steps. Before you get started, you might want to get familiar with how PowerShell is used on Azure Stack Hub. For more information, see [Get started with PowerShell on Azure Stack Hub](#).

Azure Stack Hub uses Azure Resource Manager as its underlying deployment, management, and organization mechanism. If you're going to manage Azure Stack Hub and help support users, you can learn about Resource Manager. See the [Getting Started with Azure Resource Manager](#) whitepaper.

Your typical responsibilities

Your users want to use services. From their perspective, your main role is to make these services available to them. Decide which services to offer, and make those services available by creating plans, offers, and quotas. For more information, see [Overview of offering services in Azure Stack Hub](#).

You'll also need to add items to [Azure Stack Hub Marketplace](#). The easiest way is to [download marketplace items from Azure to Azure Stack Hub](#).

NOTE

If you want to test your plans, offers, and services, you can use the [user portal](#); not the administrator portal.

In addition to providing services, you must do the regular duties of an operator to keep Azure Stack Hub up and running. These duties include the following tasks:

- Add user accounts (for [Azure AD](#) deployment or for [AD FS](#) deployment).
- [Assign role-based access control \(RBAC\) roles](#) (This task isn't restricted to admins.)
- [Monitor infrastructure health](#).
- Manage [network](#) and [storage](#) resources.
- Replace bad hardware. For example, [replace a failed disk](#).

What to tell your users

You'll need to let your users know how to work with services in Azure Stack Hub, how to connect to the environment, and how to subscribe to offers. Besides any custom documentation that you may want to provide your users, you can direct users to [Azure Stack Hub User Documentation](#).

Understand how to work with services in Azure Stack Hub

There's information your users must understand before they use services and build apps in Azure Stack Hub. For example, there are specific PowerShell and API version requirements. Also, there are some feature deltas between a service in Azure and the equivalent service in Azure Stack Hub. Make sure that your users review the following articles:

- [Key considerations: Using services or building apps for Azure Stack Hub](#)
- [Considerations for Virtual Machines in Azure Stack Hub](#)
- [Storage: differences and considerations](#)

The information in these articles summarizes the differences between a service in Azure and Azure Stack Hub. It supplements the information that's available for an Azure service in the global Azure documentation.

Connect to Azure Stack Hub as a user

In an ASDK environment, if a user doesn't use Remote Desktop to connect to the ASDK host, they can configure a virtual private network (VPN) connection to connect to Azure Stack Hub. See [Connect to Azure Stack Hub](#).

Your users will want to know how to [access the user portal](#) or how to connect through PowerShell. In an integrated systems environment, the user portal address varies per deployment. You'll need to provide your users with the correct URL.

If using PowerShell, users may have to register resource providers before they can use services. A resource provider manages a service. For example, the networking resource provider manages resources like virtual networks, network interfaces, and load balancers. They must [install](#) PowerShell, [download](#) additional modules, and [configure](#) PowerShell (which includes resource provider registration).

Subscribe to an offer

Before a user can use services, they must [subscribe to an offer](#) that you've created as an operator.

Where to get support

NOTE

To find support information for earlier releases of Azure Stack Hub (pre-1905), see [Help and Support for earlier releases](#) [Azure Stack Hub \(pre-1905\)](#).

Integrated systems

For an integrated system, there's a coordinated escalation and resolution process between Microsoft and our original equipment manufacturer (OEM) hardware partners.

If there's a cloud services issue, support is offered through Microsoft Customer Support Services (CSS). To open a support request, select the Help and support icon (question mark) in the upper-right corner of the administrator portal, select **Help + support**, and then select **New support request** under the **Support** section.

If there's an issue with deployment, patch and update, hardware (including field replaceable units), or any hardware-branded software, like software running on the hardware lifecycle host, contact your OEM hardware vendor first.

For anything else, contact Microsoft CSS.

Azure Stack Development Kit (ASDK)

For the ASDK, you can ask support-related questions in the [Microsoft forums](#). To get to the forums, select the Help and support icon (question mark) in the upper-right corner of the administrator portal, then select **Help + support**, and then select **MSDN Forums** under the **Support** section. These forums are regularly monitored. Because the ASDK is an evaluation environment, there's no official support offered through Microsoft CSS.

Next steps

[Region management in Azure Stack Hub](#)

Clear portal user data from Azure Stack Hub

4 minutes to read • [Edit Online](#)

Azure Stack Hub operators can clear portal user data on demand, when Azure Stack Hub users request it. As an Azure Stack Hub user, the portal can be customized by pinning tiles and changing the dashboard layout. Users can also change the theme and adjust the default language to match personal preferences.

Portal user data includes favorites and recently accessed resources in the Azure Stack Hub user portal. This article describes how to clear the portal user data.

Removing portal user settings should only be done after the user subscription has been deleted.

NOTE

Some user data can still exist in the system section of event logs after following the guidance in this article. This data can remain for several days until the logs automatically roll over.

Requirements

- [Install PowerShell for Azure Stack Hub](#).
- [Download the latest Azure Stack Hub tools](#) from GitHub.
- The user account must still exist in the directory.
- Azure Stack Hub admin credentials to access the admin Resource Manager endpoint.

NOTE

If you attempt to delete portal user information from a user that was invited from a guest directory, (multi-tenancy), you must have read permission in that directory. For more information, see the [CSP scenario later in this article](#).

Clear portal user data using a user principal name

This scenario assumes that either the default provider subscription and the user are part of the same directory, or that you have read access to the directory in which the user resides.

Make sure to [download the latest version of the Azure Stack Hub tools](#) from GitHub before you proceed.

For this procedure, use a computer that can communicate with the admin Resource Manager endpoint of Azure Stack Hub.

1. Open an elevated Windows PowerShell session (run as administrator), navigate to the root folder in the **AzureStack-Tools-master** directory, and import the required PowerShell module:

```
Import-Module .\DatacenterIntegration\Portal\PortalUserDataUtilities.psm1
```

2. Run the following commands. Make sure to substitute the placeholders with values that match your environment:

```

## The following Azure Resource Manager endpoint is for the ASDK. If you are in a multinode environment,
contact your operator or service provider to get the endpoint.

$adminARMEndpoint = "https://adminmanagement.local.azurestack.external"

## Replace the following value with the Azure Stack Hub directory tenant ID.
$azureStackDirectoryTenantId = "f5025bf2-547f-4b49-9693-6420c1d5e4ca"

## Replace the following value with the user directory tenant ID.
$userDirectoryTenantId = "7ddf3648-9671-47fd-b63d-eecd82ed040e"

## Replace the following value with name of the user principal whose portal user data is to be cleared.
$userPrincipalName = "myaccount@contoso.onmicrosoft.com"

Clear-AzsUserDataWithUserPrincipalName -AzsAdminArmEndpoint $adminARMEndpoint `

-AzsAdminDirectoryTenantId $azureStackDirectoryTenantId `

-UserPrincipalName $userPrincipalName `

-DirectoryTenantId $userDirectoryTenantId

```

NOTE

`azureStackDirectoryTenantId` is optional. If you do not specify this value, the script searches for the user principal name in all tenant directories registered in Azure Stack Hub, and then clears the portal data for all matched users.

Clear portal user data in guest directory

In this scenario, the Azure Stack Hub operator has no access to the guest directory in which the user resides. This is a common scenario when you are a Cloud Solution Provider (CSP).

For an Azure Stack Hub operator to remove the portal user data, at a minimum the user object ID is required.

The user must query the object ID and provide it to the Azure Stack Hub operator. The operator does not have access to the directory in which the user resides.

User retrieves the user object ID

1. Open an elevated Windows PowerShell session (run as administrator), navigate to the root folder in the **AzureStack-Tools-master** directory, and then import the necessary PowerShell module.

```
Import-Module .\DatacenterIntegration\Portal\PortalUserDataUtilities.psm1
```

2. Run the following commands. Make sure to substitute the placeholders with values that match your environment.

```

## The following Azure Resource Manager endpoint is for the ASDK. If you are in a multinode environment,
contact your operator or service provider to get the endpoint.

$userARMEndpoint = "https://management.local.azurestack.external"

## Replace the following value with the directory tenant ID, which contains the user account.
$userDirectoryTenantId = "3160cbf5-c227-49dd-8654-86e924c0b72f"

## Replace the following value with the name of the user principal whose portal user data is to be
cleared.
$userPrincipleName = "myaccount@contoso.onmicrosoft.com"

Get-UserObjectId -DirectoryTenantId $userDirectoryTenantId `

-AzsArmEndpoint $userARMEndpoint `

-UserPricinpalName $userPrincipleName

```

NOTE

As a user, you must provide the user object ID, which is the output of the previous script, to the Azure Stack Hub operator.

Azure Stack Hub operator removes the portal user data

After receiving the user object ID as an Azure Stack Hub operator, run the following commands to remove the portal user data:

1. Open an elevated Windows PowerShell session (run as administrator), navigate to the root folder in the **AzureStack-Tools-master** directory, and then import the necessary PowerShell module.

```
Import-Module .\DatacenterIntegration\Portal\PortalUserDataUtilities.psm1
```

2. Run the following commands, making sure you adjust the parameter to match your environment:

```
## The following Azure Resource Manager endpoint is for the ASDK. If you are in a multinode environment,
## contact your operator or service provider to get the endpoint.
$AzsAdminARMEndpoint = "https://adminmanagement.local.azurestack.external"

## Replace the following value with the Azure Stack Hub directory tenant ID.
$AzsAdminDirectoryTenantId = "f5025bf2-547f-4b49-9693-6420c1d5e4ca"

## Replace the following value with the directory tenant ID of the user to clear.
$DirectoryTenantId = "3160cbf5-c227-49dd-8654-86e924c0b72f"

## Replace the following value with the name of the user principal whose portal user data is to be
## cleared.
$userObjectID = "s-1-*****"
Clear-AzsUserDataWithUserObject -AzsAdminArmEndpoint $AzsAdminARMEndpoint `

-AzsAdminDirectoryTenantId $AzsAdminDirectoryTenantId `

-DirectoryTenantID $DirectoryTenantId `

-UserObjectID $userObjectID `
```

Next steps

- [Register Azure Stack Hub with Azure](#) and populate the [Azure Stack Hub Marketplace](#) with items to offer your users.

Configure Azure Stack Hub telemetry

7 minutes to read • [Edit Online](#)

Azure Stack Hub telemetry automatically uploads system data to Microsoft via the Connected User Experience. Microsoft teams use the data that Azure Stack Hub telemetry gathers to improve customer experiences. This data is also used for security, health, quality, and performance analysis.

For an Azure Stack Hub operator, telemetry can provide valuable insights into enterprise deployments and gives you a voice that helps shape future versions of Azure Stack Hub.

NOTE

You can also configure Azure Stack Hub to forward usage information to Azure for billing. This is required for multi-node Azure Stack Hub customers who choose pay-as-you-use billing. Usage reporting is controlled independently from telemetry and isn't required for multi-node customers who choose the capacity model or for Azure Stack Development Kit users. For these scenarios, usage reporting can be turned off [using the registration script](#).

Azure Stack Hub telemetry is based on the Windows Server 2016 Connected User Experience and Telemetry component. This component uses the [Event Tracing for Windows \(ETW\)](#) TraceLogging technology to gather and store events and data. Azure Stack components use the same technology to publish events and data gathered by using public operating system event logging and tracing APIs. Examples of these Azure Stack Hub components include these providers: Network Resource, Storage Resource, Monitoring Resource, and Update Resource. The Connected User Experience and Telemetry component encrypts data using SSL and uses certificate pinning to transmit data over HTTPS to the Microsoft Data Management service.

IMPORTANT

To enable telemetry data flow, port 443 (HTTPS) must be open in your network. The Connected User Experience and Telemetry component connects to the Microsoft Data Management service at <https://v10.events.data.microsoft.com>. The Connected User Experience and Telemetry component also connects to <https://settings-win.data.microsoft.com> to download configuration information. Other diagnostic data services connect <https://watson.telemetry.microsoft.com> for error reporting.

Privacy considerations

The ETW service routes telemetry data back to protected cloud storage. The principle of least privilege guides access to telemetry data. Only Microsoft personnel with a valid business need are given access to the telemetry data. Microsoft doesn't share personal customer data with third parties, except at the customer's discretion or for the limited purposes described in the [Microsoft Privacy Statement](#). Business reports that are shared with OEMs and partners include aggregated, anonymized data. Data sharing decisions are made by an internal Microsoft team including privacy, legal, and data management stakeholders.

Microsoft believes in, and practices information minimization. We strive to gather only the information that's needed, and store it for only as long as necessary to provide a service or for analysis. Much of the information about how the Azure Stack Hub system and Azure services are functioning is deleted within six months. Summarized or aggregated data will be kept for a longer period.

We understand that the privacy and security of customer information is important. Microsoft takes a thoughtful and comprehensive approach to customer privacy and the protection of customer data in Azure Stack Hub. IT administrators have controls to customize features and privacy settings at any time. Our commitment to

transparency and trust is clear:

- We're open with customers about the types of data we gather.
- We put enterprise customers in control — they can customize their own privacy settings.
- We put customer privacy and security first.
- We're transparent about how telemetry data gets used.
- We use telemetry data to improve customer experiences.

Microsoft doesn't intend to gather sensitive data, like credit card numbers, usernames and passwords, email addresses, or similar sensitive information. If we determine that sensitive information has been inadvertently received, we delete it.

Examples of how Microsoft uses the telemetry data

Telemetry plays an important role in helping to quickly identify and fix critical reliability issues in customer deployments and configurations. Insights from telemetry data can help identify issues with services or hardware configurations. Microsoft's ability to get this data from customers and drive improvements to the ecosystem raises the bar for the quality of integrated Azure Stack Hub solutions.

Telemetry also helps Microsoft to better understand how customers deploy components, use features, and use services to achieve their business goals. These insights help prioritize engineering investments in areas that can directly impact customer experiences and workloads.

Some examples include customer use of containers, storage, and networking configurations that are associated with Azure Stack Hub roles. We also use the insights to drive improvements and intelligence into Azure Stack Hub management and monitoring solutions. These improvements make it easier for customers to diagnose issues and save money by making fewer support calls to Microsoft.

Manage telemetry collection

We don't recommend turning off telemetry in your organization. However, in some scenarios it may be necessary.

In these scenarios, you can configure the telemetry level sent to Microsoft by using registry settings before you deploy Azure Stack Hub, or by using the Telemetry Endpoints after you deploy Azure Stack Hub.

Telemetry levels and data collection

Before you change telemetry settings, you should understand the telemetry levels and what data is collected at each level.

The telemetry settings are grouped into four levels (0-3) that are cumulative and categorized as the follows:

0 (Security)

Security data only. Information that's required to keep the operating system secure. This includes data about the Connected User Experience and Telemetry component settings, and Windows Defender. No telemetry specific to Azure Stack Hub is emitted at this level.

1 (Basic)

Security data, and Basic Health and Quality data. Basic device information, including: quality-related data, app compatibility, app usage data, and data from the **Security** level. Setting your telemetry level to Basic enables Azure Stack Hub telemetry. The data gathered at this level includes:

- *Basic device information* that provides an understanding about the types and configurations of native and virtual Windows Server 2016 instances in the ecosystem. This includes:
 - Machine attributes, such as the OEM, and model.
 - Networking attributes, such as the number of network adapters and their speed.

- Processor and memory attributes, such as the number of cores, and amount of installed memory.
- Storage attributes, such as the number of drives, type of drive, and drive size.
- *Telemetry functionality*, including the percentage of uploaded events, dropped events, and the last data upload time.
- *Quality-related information* that helps Microsoft develop a basic understanding of how Azure Stack Hub is performing. For example, the count of critical alerts on a particular hardware configuration.
- *Compatibility data* that helps provide an understanding about which Resource Providers are installed on a system and a virtual machine (VM). This identifies potential compatibility problems.

2 (Enhanced)

Additional insights, including: how the operating system and Azure Stack Hub services are used, how these services perform, advanced reliability data, and data from the **Security** and **Basic** levels.

NOTE

This is the default telemetry setting.

3 (Full)

All data necessary to identify and help to fix problems, plus data from the **Security**, **Basic**, and **Enhanced** levels.

IMPORTANT

These telemetry levels only apply to Microsoft Azure Stack Hub components. Non-Microsoft software components and services that are running in the Hardware Lifecycle Host from Azure Stack Hub hardware partners may communicate with their cloud services outside of these telemetry levels. You should work with your Azure Stack Hub hardware solution provider to understand their telemetry policy, and how you can opt in or opt out.

Turning off Windows and Azure Stack Hub telemetry also disables SQL telemetry. For more information about the implications of the Windows Server telemetry settings, see the [Windows Telemetry Whitepaper](#).

ASDK: set the telemetry level in the Windows registry

You can use the Windows Registry Editor to manually set the telemetry level on the physical host computer before you deploy Azure Stack Hub. If a management policy already exists, such as Group Policy, it overrides this registry setting.

Before you deploy Azure Stack Hub on the development kit host, boot into CloudBuilder.vhd and run the following script in an elevated PowerShell window:

```
### Get current AllowTelemetry value on DVM Host
(Get-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\DataCollection" ` 
-Name AllowTelemetry).AllowTelemetry
### Set & Get updated AllowTelemetry value for ASDK-Host
Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\DataCollection" ` 
-Name "AllowTelemetry" -Value '0' # Set this value to 0,1,2,or3.
(Get-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\DataCollection" ` 
-Name AllowTelemetry).AllowTelemetry
```

ASDK and Multi-Node: enable or disable telemetry after deployment

To enable or disable telemetry after deployment, you need access to the privileged endpoint (PEP) which is exposed on the ERCS VMs.

- To Enable: `Set-Telemetry -Enable`
- To Disable: `Set-Telemetry -Disable`

PARAMETER details:

- `.PARAMETER Enable` - Turn on telemetry data upload
- `.PARAMETER Disable` - Turn off telemetry data upload

Script to enable telemetry:

```
$ip = "<IP ADDRESS OF THE PEP VM>" # You can also use the machine name instead of IP here.  
$pwd= ConvertTo-SecureString "<CLOUD ADMIN PASSWORD>" -AsPlainText -Force  
$cred = New-Object System.Management.Automation.PSCredential ("<DOMAIN NAME>\CloudAdmin", $pwd)  
$psSession = New-PSSession -ComputerName $ip -ConfigurationName PrivilegedEndpoint -Credential $cred  
Invoke-Command -Session $psSession {Set-Telemetry -Enable}  
if($psSession)  
{  
    Remove-PSSession $psSession  
}
```

Script to disable telemetry:

```
$ip = "<IP ADDRESS OF THE PEP VM>" # You can also use the machine name instead of IP here.  
$pwd= ConvertTo-SecureString "<CLOUD ADMIN PASSWORD>" -AsPlainText -Force  
$cred = New-Object System.Management.Automation.PSCredential ("<DOMAIN NAME>\CloudAdmin", $pwd)  
$psSession = New-PSSession -ComputerName $ip -ConfigurationName PrivilegedEndpoint -Credential $cred  
Invoke-Command -Session $psSession {Set-Telemetry -Disable}  
if($psSession)  
{  
    Remove-PSSession $psSession  
}
```

Next steps

[Register Azure Stack Hub with Azure](#)

Register Azure Stack Hub with Azure

19 minutes to read • [Edit Online](#)

Register Azure Stack Hub with Azure so you can download Azure Marketplace items from Azure and set up commerce data reporting back to Microsoft. After you register Azure Stack Hub, usage is reported to Azure commerce and you can see it under the Azure billing Subscription ID used for registration.

The information in this article describes registering Azure Stack Hub integrated systems with Azure. For information about registering the ASDK with Azure, see [Azure Stack Hub registration](#) in the ASDK documentation.

IMPORTANT

Registration is required to support full Azure Stack Hub functionality, including offering items in the marketplace. You'll be in violation of Azure Stack Hub licensing terms if you don't register when using the pay-as-you-use billing model. To learn more about Azure Stack Hub licensing models, see the [How to buy page](#).

Prerequisites

You need the following prerequisites in place before you register:

- Verify your credentials.
- Set the PowerShell language mode.
- Install PowerShell for Azure Stack Hub.
- Download the Azure Stack Hub tools.
- Determine your registration scenario.

Verify your credentials

Before registering Azure Stack Hub with Azure, you must have:

- The subscription ID for an Azure subscription. Only EA, CSP, or CSP shared services subscriptions are supported for registration. CSPs need to decide whether to [use a CSP or APSS subscription](#).

To get the ID, sign in to Azure, click **All services**. Then, under the **GENERAL** category, select **Subscriptions**, click the subscription you want to use, and under **Essentials** you can find the Subscription ID. As a best practice, use separate subscriptions for production and dev or test environments.

NOTE

Germany cloud subscriptions aren't currently supported.

- The username and password for an account that's an owner for the subscription.
- The user account needs to have access to the Azure subscription and have permissions to create identity apps and service principals in the directory associated with that subscription. We recommend that you register Azure Stack Hub with Azure using least-privilege administration. For more information on how to create a custom role definition that limits access to your subscription for registration, see [create a registration role for Azure Stack Hub](#).

- Registered the Azure Stack Hub resource provider (see the following Register Azure Stack Hub Resource Provider section for details).

After registration, Azure Active Directory (Azure AD) global administrator permission isn't required. However, some operations may require the global admin credential (for example, a resource provider installer script or a new feature requiring a permission to be granted). You can either temporarily reinstate the account's global admin permissions or use a separate global admin account that's an owner of the *default provider subscription*.

The user who registers Azure Stack Hub is the owner of the service principal in Azure AD. Only the user who registered Azure Stack Hub can modify the Azure Stack Hub registration. If a non-admin user that's not an owner of the registration service principal attempts to register or re-register Azure Stack Hub, they may come across a 403 response. A 403 response indicates the user has insufficient permissions to complete the operation.

If you don't have an Azure subscription that meets these requirements, you can [create a free Azure account here](#). Registering Azure Stack Hub incurs no cost on your Azure subscription.

NOTE

If you have more than one Azure Stack Hub, a best practice is to register each Azure Stack Hub to its own subscription. This makes it easier for you to track usage.

PowerShell language mode

To successfully register Azure Stack Hub, the PowerShell language mode must be set to **FullLanguageMode**. To verify that the current language mode is set to full, open an elevated PowerShell window and run the following PowerShell cmdlets:

```
$ExecutionContext.SessionState.LanguageMode
```

Ensure the output returns **FullLanguageMode**. If any other language mode is returned, registration needs to be run on another machine or the language mode needs to be set to **FullLanguageMode** before continuing.

Install PowerShell for Azure Stack Hub

Use the latest PowerShell for Azure Stack Hub to register with Azure.

If the latest version isn't already installed, see [install PowerShell for Azure Stack Hub](#).

Download the Azure Stack Hub tools

The Azure Stack Hub tools GitHub repository contains PowerShell modules that support Azure Stack Hub functionality, including registration functionality. During the registration process, you need to import and use the **RegisterWithAzure.psm1** PowerShell module (found in the Azure Stack Hub tools repository) to register your Azure Stack Hub instance with Azure.

To ensure you're using the latest version, delete any existing versions of the Azure Stack Hub tools and [download the latest version from GitHub](#) before registering with Azure.

Determine your registration scenario

Your Azure Stack Hub deployment may be *connected* or *disconnected*.

- **Connected**

Connected means you've deployed Azure Stack Hub so that it can connect to the internet and to Azure.

You either have Azure AD or Active Directory Federation Services (AD FS) for your identity store. With a connected deployment, you can choose from two billing models: pay-as-you-use or capacity-based.

- [Register a connected Azure Stack Hub with Azure using the pay-as-you-use billing model](#).

- Register a connected Azure Stack Hub with Azure using the **capacity** billing model.
- **Disconnected**
With the disconnected from Azure deployment option, you can deploy and use Azure Stack Hub without a connection to the internet. However, with a disconnected deployment, you're limited to an AD FS identity store and the capacity-based billing model.
 - Register a disconnected Azure Stack Hub using the **capacity** billing model .

Determine a unique registration name to use

When you register Azure Stack Hub with Azure, you must provide a unique registration name. An easy way to associate your Azure Stack Hub subscription with an Azure registration is to use your Azure Stack Hub **Cloud ID**.

NOTE

Azure Stack Hub registrations using the capacity-based billing model will need to change the unique name when re-registering after those yearly subscriptions expire unless you [delete the expired registration](#) and re-register with Azure.

To determine the Cloud ID for your Azure Stack Hub deployment, open PowerShell as an admin on a computer that can access the Privileged Endpoint, run the following commands, and then record the **CloudID** value:

```
Run: Enter-PSSession -ComputerName <privileged endpoint computer name> -ConfigurationName  
PrivilegedEndpoint  
Run: Get-AzureStackStampInformation
```

Register connected with pay-as-you-go billing

Use these steps to register Azure Stack Hub with Azure using the pay-as-you-use billing model.

NOTE

All these steps must be run from a computer that has access to the privileged endpoint (PEP). For details about the PEP, see [Using the privileged endpoint in Azure Stack Hub](#).

Connected environments can access the internet and Azure. For these environments, you need to register the Azure Stack Hub resource provider with Azure and then configure your billing model.

1. To register the Azure Stack Hub resource provider with Azure, start PowerShell ISE as an administrator and use the following PowerShell cmdlets with the **EnvironmentName** parameter set to the appropriate Azure subscription type (see parameters below).
2. Add the Azure account that you used to register Azure Stack Hub. To add the account, run the **Add-AzureRmAccount** cmdlet. You're prompted to enter your Azure account credentials and you may have to use two-factor authentication based on your account's configuration.

```
Add-AzureRmAccount -EnvironmentName "<environment name>"
```

PARAMETER	DESCRIPTION
-----------	-------------

PARAMETER	DESCRIPTION
EnvironmentName	The Azure cloud subscription environment name. Supported environment names are AzureCloud , AzureUSGovernment , or if using a China Azure Subscription, AzureChinaCloud .

NOTE

If your session expires, your password has changed, or you simply wish to switch accounts, run the following cmdlet before you sign in using Add-AzureRmAccount: `Remove-AzureRmAccount -Scope Process`

3. If you have multiple subscriptions, run the following command to select the one you want to use:

```
Get-AzureRmSubscription -SubscriptionID '<Your Azure Subscription GUID>' | Select-AzureRmSubscription
```

4. Run the following command to register the Azure Stack Hub resource provider in your Azure subscription:

```
Register-AzureRmResourceProvider -ProviderNamespace Microsoft.AzureStack
```

5. Start PowerShell ISE as an administrator and navigate to the **Registration** folder in the **AzureStack-Tools-master** directory created when you downloaded the Azure Stack Hub tools. Import the **RegisterWithAzure.psm1** module using PowerShell:

```
Import-Module .\RegisterWithAzure.psm1
```

6. Next, in the same PowerShell session, ensure you're signed in to the correct Azure PowerShell context. This context would be the Azure account that was used to register the Azure Stack Hub resource provider previously. Powershell to run:

```
Connect-AzureRmAccount -Environment "<environment name>"
```

PARAMETER	DESCRIPTION
EnvironmentName	The Azure cloud subscription environment name. Supported environment names are AzureCloud , AzureUSGovernment , or if using a China Azure Subscription, AzureChinaCloud .

7. In the same PowerShell session, run the **Set-AzsRegistration** cmdlet. PowerShell to run:

```
$CloudAdminCred = Get-Credential -UserName <Privileged endpoint credentials> -Message "Enter the cloud domain credentials to access the privileged endpoint."
$RegistrationName = "<unique-registration-name>"
Set-AzsRegistration ` 
    -PrivilegedEndpointCredential $CloudAdminCred ` 
    -PrivilegedEndpoint <PrivilegedEndPoint computer name> ` 
    -BillingModel PayAsYouUse ` 
    -RegistrationName $RegistrationName
```

For more information on the Set-AzsRegistration cmdlet, see [Registration reference](#).

The process takes between 10 and 15 minutes. When the command completes, you see the message **"Your environment is now registered and activated using the provided parameters."**

Register connected with capacity billing

Use these steps to register Azure Stack Hub with Azure using the pay-as-you-use billing model.

NOTE

All these steps must be run from a computer that has access to the privileged endpoint (PEP). For details about the PEP, see [Using the privileged endpoint in Azure Stack Hub](#).

Connected environments can access the internet and Azure. For these environments, you need to register the Azure Stack Hub resource provider with Azure and then configure your billing model.

1. To register the Azure Stack Hub resource provider with Azure, start PowerShell ISE as an administrator and use the following PowerShell cmdlets with the **EnvironmentName** parameter set to the appropriate Azure subscription type (see parameters below).
2. Add the Azure account that you used to register Azure Stack Hub. To add the account, run the **Add-AzureRmAccount** cmdlet. You're prompted to enter your Azure account credentials and you may have to use two-factor authentication based on your account's configuration.

```
Connect-AzureRmAccount -Environment "<environment name>"
```

PARAMETER	DESCRIPTION
EnvironmentName	The Azure cloud subscription environment name. Supported environment names are AzureCloud , AzureUSGovernment , or if using a China Azure Subscription, AzureChinaCloud .

3. If you have multiple subscriptions, run the following command to select the one you want to use:

```
Get-AzureRmSubscription -SubscriptionID '<Your Azure Subscription GUID>' | Select-AzureRmSubscription
```

4. Run the following command to register the Azure Stack Hub resource provider in your Azure subscription:

```
Register-AzureRmResourceProvider -ProviderNamespace Microsoft.AzureStack
```

5. Start PowerShell ISE as an administrator and navigate to the **Registration** folder in the **AzureStack-Tools-master** directory created when you downloaded the Azure Stack Hub tools. Import the **RegisterWithAzure.psm1** module using PowerShell:

```
$CloudAdminCred = Get-Credential -UserName <Privileged endpoint credentials> -Message "Enter the  
cloud domain credentials to access the privileged endpoint."  
$RegistrationName = "<unique-registration-name>"  
Set-AzsRegistration`  
    -PrivilegedEndpointCredential $CloudAdminCred`  
    -PrivilegedEndpoint <PrivilegedEndPoint computer name>`  
    -AgreementNumber <EA agreement number>`  
    -BillingModel Capacity`  
    -RegistrationName $RegistrationName
```

NOTE

You can disable usage reporting with the UsageReportingEnabled parameter for the **Set-AzsRegistration** cmdlet by setting the parameter to false.

For more information on the Set-AzsRegistration cmdlet, see [Registration reference](#).

Register disconnected with capacity billing

If you're registering Azure Stack Hub in a disconnected environment (with no internet connectivity), you need to get a registration token from the Azure Stack Hub environment. Then use that token on a computer that can connect to Azure and has PowerShell for Azure Stack Hub installed.

Get a registration token from the Azure Stack Hub environment

1. Start PowerShell ISE as an administrator and navigate to the **Registration** folder in the **AzureStack-Tools-master** directory created when you downloaded the Azure Stack Hub tools. Import the **RegisterWithAzure.psm1** module:

```
Import-Module .\RegisterWithAzure.psm1
```

2. To get the registration token, run the following PowerShell cmdlets:

```
$FilePathForRegistrationToken = "$env:SystemDrive\RegistrationToken.txt"  
$RegistrationToken = Get-AzsRegistrationToken -PrivilegedEndpointCredential $YourCloudAdminCredential  
-UsageReportingEnabled:$False -PrivilegedEndpoint $YourPrivilegedEndpoint -BillingModel Capacity -  
AgreementNumber '<EA agreement number>' -TokenOutputFilePath $FilePathForRegistrationToken
```

For more information on the Get-AzsRegistrationToken cmdlet, see [Registration reference](#).

TIP

The registration token is saved in the file specified for *\$FilePathForRegistrationToken*. You can change the filepath or filename at your discretion.

3. Save this registration token for use on the Azure-connected machine. You can copy the file or the text from *\$FilePathForRegistrationToken*.

Connect to Azure and register

On the computer that is connected to the internet, do the same steps to import the RegisterWithAzure.psm1 module and sign in to the correct Azure Powershell context. Then call Register-AzsEnvironment. Specify the registration token to register with Azure. If you're registering more than one instance of Azure Stack Hub using the same Azure Subscription ID, specify a unique registration name.

You need your registration token and a unique token name.

1. Start PowerShell ISE as an administrator and navigate to the **Registration** folder in the **AzureStack-Tools-master** directory created when you downloaded the Azure Stack Hub tools. Import the **RegisterWithAzure.psm1** module:

```
Import-Module .\RegisterWithAzure.psm1
```

2. Then run the following PowerShell cmdlets:

```
$RegistrationToken = "<Your Registration Token>"  
$RegistrationName = "<unique-registration-name>"  
Register-AzsEnvironment -RegistrationToken $RegistrationToken -RegistrationName $RegistrationName
```

Optionally, you can use the Get-Content cmdlet to point to a file that contains your registration token.

You need your registration token and a unique token name.

1. Start PowerShell ISE as an administrator and navigate to the **Registration** folder in the **AzureStack-Tools-master** directory created when you downloaded the Azure Stack Hub tools. Import the **RegisterWithAzure.psm1** module:

```
Import-Module .\RegisterWithAzure.psm1
```

2. Then run the following PowerShell cmdlets:

```
$RegistrationToken = Get-Content -Path '<Path>\<Registration Token File>'  
Register-AzsEnvironment -RegistrationToken $RegistrationToken -RegistrationName $RegistrationName
```

NOTE

Save the registration resource name and the registration token for future reference.

Retrieve an Activation Key from Azure Registration Resource

Next, you need to retrieve an activation key from the registration resource created in Azure during Register-AzsEnvironment.

To get the activation key, run the following PowerShell cmdlets:

```
$RegistrationResourceName = "<unique-registration-name>"  
$KeyOutputFilePath = "$env:SystemDrive\ActivationKey.txt"  
$ActivationKey = Get-AzsActivationKey -RegistrationName $RegistrationResourceName -KeyOutputFilePath  
$KeyOutputFilePath
```

TIP

The activation key is saved in the file specified for `$KeyOutputFilePath`. You can change the filepath or filename at your discretion.

Create an Activation Resource in Azure Stack Hub

Return to the Azure Stack Hub environment with the file or text from the activation key created from Get-

AzsActivationKey. Next create an activation resource in Azure Stack Hub using that activation key. To create an activation resource, run the following PowerShell cmdlets:

```
$ActivationKey = "<activation key>"  
New-AzsActivationResource -PrivilegedEndpointCredential $YourCloudAdminCredential -PrivilegedEndpoint  
$YourPrivilegedEndpoint -ActivationKey $ActivationKey
```

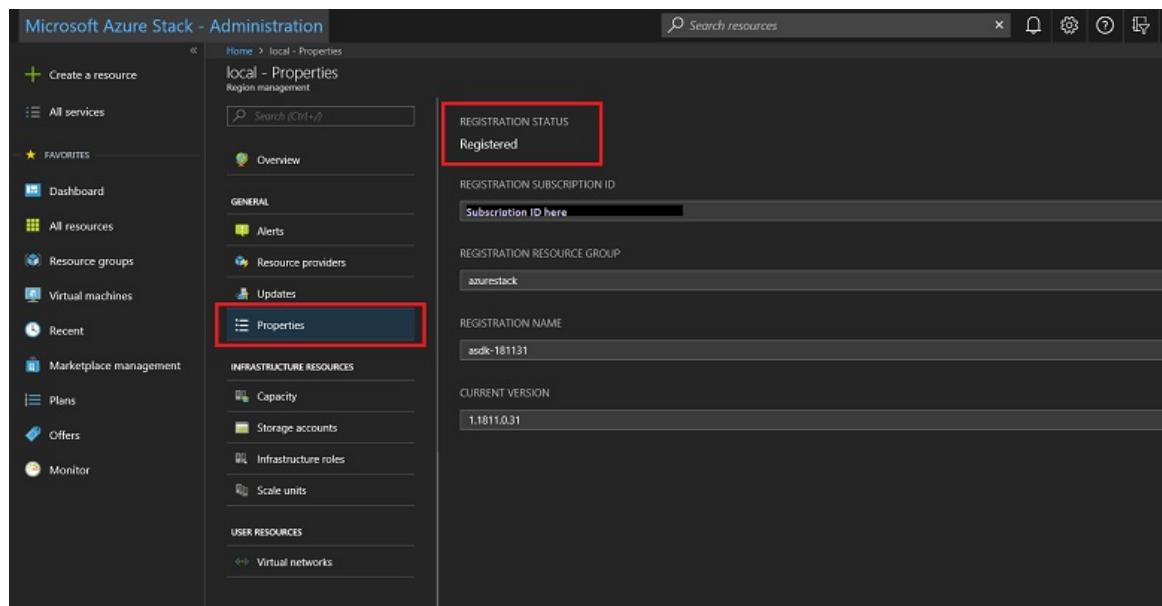
Optionally, you can use the Get-Content cmdlet to point to a file that contains your registration token:

```
$ActivationKey = Get-Content -Path '<Path>\<Activation Key File>'  
New-AzsActivationResource -PrivilegedEndpointCredential $YourCloudAdminCredential -PrivilegedEndpoint  
$YourPrivilegedEndpoint -ActivationKey $ActivationKey
```

Verify Azure Stack Hub registration

You can use the **Region management** tile to verify that the Azure Stack Hub registration was successful. This tile is available on the default dashboard in the administrator portal. The status can be registered, or not registered. If registered, it also shows the Azure subscription ID that you used to register your Azure Stack Hub along with the registration resource group and name.

1. Sign in to the [Azure Stack Hub administrator portal](#).
2. From the Dashboard, select **Region management**.
3. Select **Properties**. This blade shows the status and details of your environment. The status can be **Registered**, **Not registered**, or **Expired**.



If registered, the properties include:

- **Registration subscription ID:** The Azure subscription ID registered and associated to Azure Stack Hub.
 - **Registration resource group:** The Azure resource group in the associated subscription containing the Azure Stack Hub resources.
4. You can use the Azure portal to view Azure Stack Hub registration resources, and then verify that the registration succeeded. Sign in to the [Azure portal](#) using an account associated to the subscription you used to register Azure Stack Hub. Select **All resources**, enable the **Show hidden types** checkbox, and select the registration name.

5. If the registration didn't succeed, you must re-register by following the [steps here](#) to resolve the issue.

Alternatively, you can verify if your registration was successful by using the Marketplace management feature. If you see a list of marketplace items in the Marketplace management blade, your registration was successful. However, in disconnected environments, you can't see marketplace items in Marketplace management.

NOTE

After registration is complete, the active warning for not registering will no longer appear. In Azure Stack Hub releases before 1904, in disconnected scenarios, you see a message in Marketplace management asking you to register and activate your Azure Stack Hub, even if you have registered successfully. This message doesn't appear in release 1904 and later.

Renew or change registration

Renew or change registration in connected environments

You need to update or renew your registration in the following circumstances:

- After you renew your capacity-based yearly subscription.
- When you change your billing model.
- When you scale changes (add/remove nodes) for capacity-based billing.

Change the subscription you use

If you want to change the subscription you use, you must first run the **Remove-AzsRegistration** cmdlet, then ensure you're signed in to the correct Azure PowerShell context. Then run **Set-AzsRegistration** with any changed parameters including <billing model> :

```
Remove-AzsRegistration -PrivilegedEndpointCredential $YourCloudAdminCredential -PrivilegedEndpoint  
$YourPrivilegedEndpoint -RegistrationName $RegistrationName  
Set-AzureRmContext -SubscriptionId $NewSubscriptionId  
Set-AzsRegistration -PrivilegedEndpointCredential $YourCloudAdminCredential -PrivilegedEndpoint  
$YourPrivilegedEndpoint -BillingModel <billing model> -RegistrationName $RegistrationName
```

Change the billing model or how to offer features

If you want to change the billing model or how to offer features for your installation, you can call the registration function to set the new values. You don't need to first remove the current registration:

```
Set-AzsRegistration -PrivilegedEndpointCredential $YourCloudAdminCredential -PrivilegedEndpoint  
$YourPrivilegedEndpoint -BillingModel <billing model> -RegistrationName $RegistrationName
```

Renew or change registration in disconnected environments

You need to update or renew your registration in the following circumstances:

- After you renew your capacity-based yearly subscription.
- When you change your billing model.
- When you scale changes (add/remove nodes) for capacity-based billing.

Remove the activation resource from Azure Stack Hub

You first need to remove the activation resource from Azure Stack Hub, and then the registration resource in Azure.

To remove the activation resource in Azure Stack Hub, run the following PowerShell cmdlets in your Azure Stack Hub environment:

```
Remove-AzsActivationResource -PrivilegedEndpointCredential $YourCloudAdminCredential -PrivilegedEndpoint  
$YourPrivilegedEndpoint
```

Next, to remove the registration resource in Azure, ensure you're on an Azure-connected computer, sign in to the correct Azure PowerShell context, and run the appropriate PowerShell cmdlets as described below.

You can use the registration token used to create the resource:

```
$RegistrationToken = "<registration token>"  
Unregister-AzsEnvironment -RegistrationToken $RegistrationToken
```

Or you can use the registration name:

```
$RegistrationName = "AzureStack-<unique-registration-name>"  
Unregister-AzsEnvironment -RegistrationName $RegistrationName
```

Re-register using disconnected steps

You've now completely unregistered in a disconnected scenario and must repeat the steps for registering an Azure Stack Hub environment in a disconnected scenario.

Disable or enable usage reporting

For Azure Stack Hub environments that use a capacity billing model, turn off usage reporting with the **UsageReportingEnabled** parameter using either the **Set-AzsRegistration** or the **Get-AzsRegistrationToken** cmdlets. Azure Stack Hub reports usage metrics by default. Operators with capacity uses or supporting a disconnected environment need to turn off usage reporting.

With a connected Azure Stack Hub

```
$CloudAdminCred = Get-Credential -UserName <Privileged endpoint credentials> -Message "Enter the cloud  
domain credentials to access the privileged endpoint."  
$RegistrationName = "<unique-registration-name>"  
Set-AzsRegistration `  
    -PrivilegedEndpointCredential $CloudAdminCred `  
    -PrivilegedEndpoint <PrivilegedEndPoint computer name> `  
    -BillingModel Capacity  
    -RegistrationName $RegistrationName
```

With a disconnected Azure Stack Hub

1. To change the registration token, run the following PowerShell cmdlets:

```
$FilePathForRegistrationToken = $env:SystemDrive\RegistrationToken.txt  
$RegistrationToken = Get-AzsRegistrationToken -PrivilegedEndpointCredential -  
UsageReportingEnabled:$False  
$YourCloudAdminCredential -PrivilegedEndpoint $YourPrivilegedEndpoint -BillingModel Capacity -  
AgreementNumber '<EA agreement number>' -TokenOutputFilePath $FilePathForRegistrationToken
```

TIP

The registration token is saved in the file specified for `$FilePathForRegistrationToken`. You can change the filepath or filename at your discretion.

2. Save this registration token for use on the Azure connected machine. You can copy the file or the text from `$FilePathForRegistrationToken`.

Move a registration resource

Moving a registration resource between resource groups under the same subscription **is** supported for all environments. However, moving a registration resource between subscriptions is only supported for CSPs when both subscriptions resolve to the same Partner ID. For more information about moving resources to a new resource group, see [Move resources to new resource group or subscription](#).

IMPORTANT

To prevent accidental deletion of registration resources on the portal, the registration script automatically adds a lock to the resource. You must remove this lock before moving or deleting it. It's recommended that you add a lock to your registration resource to prevent accidental deletion.

Registration reference

Set-AzsRegistration

You can use **Set-AzsRegistration** to register Azure Stack Hub with Azure and enable or disable the offer of items in the marketplace and usage reporting.

To run the cmdlet, you need:

- A global Azure subscription of any type.
- To be signed in to Azure PowerShell with an account that's an owner or contributor to that subscription.

```
Set-AzsRegistration [-PrivilegedEndpointCredential] <PSCredential> [-PrivilegedEndpoint] <String> [[-AzureContext] <PSObject>] [[-ResourceGroupName] <String>] [[-ResourceGroupLocation] <String>] [[-BillingModel] <String>] [-MarketplaceSyndicationEnabled] [-UsageReportingEnabled] [[-AgreementNumber] <String>] [[-RegistrationName] <String>] [<CommonParameters>]
```

PARAMETER	TYPE	DESCRIPTION
PrivilegedEndpointCredential	PSCredential	The credentials used to access the privileged endpoint . The username is in the format AzureStackDomain\CloudAdmin .
PrivilegedEndpoint	String	A pre-configured remote PowerShell console that provides you with capabilities like log collection and other post deployment tasks. To learn more, refer to the using the privileged endpoint article.
AzureContext	PSObject	
ResourceGroupName	String	
ResourceGroupLocation	String	

PARAMETER	TYPE	DESCRIPTION
BillingModel	String	The billing model that your subscription uses. Allowed values for this parameter are: Capacity, PayAsYouUse, and Development.
MarketplaceSyndicationEnabled	True/False	Determines if the marketplace management feature is available in the portal. Set to true if registering with internet connectivity. Set to false if registering in disconnected environments. For disconnected registrations, the offline syndication tool can be used for downloading marketplace items.
UsageReportingEnabled	True/False	Azure Stack Hub reports usage metrics by default. Operators with capacity uses or supporting a disconnected environment need to turn off usage reporting. Allowed values for this parameter are: True, False.
AgreementNumber	String	The number of the EA agreement under which the Capacity SKU for this Azure Stack was ordered.
RegistrationName	String	Set a unique name for the registration if you're running the registration script on more than one instance of Azure Stack Hub using the same Azure Subscription ID. The parameter has a default value of AzureStackRegistration . However, if you use the same name on more than one instance of Azure Stack Hub, the script fails.

Get-AzsRegistrationToken

Get-AzsRegistrationToken generates a registration token from the input parameters.

```
Get-AzsRegistrationToken [-PrivilegedEndpointCredential] <PSCredential> [-PrivilegedEndpoint] <String>
[-BillingModel] <String> [[-TokenOutputFilePath] <String>] [-UsageReportingEnabled] [[-AgreementNumber]
<String>]
[<CommonParameters>]
```

PARAMETER	TYPE	DESCRIPTION
PrivilegedEndpointCredential	PSCredential	The credentials used to access the privileged endpoint . The username is in the format AzureStackDomain\CloudAdmin .

PARAMETER	TYPE	DESCRIPTION
PrivilegedEndpoint	String	A pre-configured remote PowerShell console that provides you with capabilities like log collection and other post deployment tasks. To learn more, refer to the using the privileged endpoint article.
AzureContext	PSObject	
ResourceGroupName	String	
ResourceGroupLocation	String	
BillingModel	String	The billing model that your subscription uses. Allowed values for this parameter are: Capacity, PayAsYouUse, and Development.
MarketplaceSyndicationEnabled	True/False	
UsageReportingEnabled	True/False	Azure Stack Hub reports usage metrics by default. Operators with capacity uses or supporting a disconnected environment need to turn off usage reporting. Allowed values for this parameter are: True, False.
AgreementNumber	String	

Registration failures

You might see one of the errors below while attempting to register your Azure Stack Hub:

- Could not retrieve mandatory hardware info for `$hostName`. Check physical host and connectivity, then try to re-run registration.
- Cannot connect to `$hostName` to get hardware info. Check physical host and connectivity, then try to re-run registration.

Cause: this is typically because we try to obtain hardware details such as UUID, Bios, and CPU from the hosts to attempt activation and weren't able to due to the inability to connect to the physical host.

- Cloud identifier [`GUID`] is already registered. Reusing cloud identifiers is not allowed.

Cause: this happens if your Azure Stack environment is already registered. If you want to re-register your environment with a different subscription or billing model, [see these instructions](#).

- When trying to access Marketplace management, an error occurs when trying to syndicate products.

Cause: this usually happens when Azure Stack Hub is unable to access the registration resource. One common reason for this is that when an Azure subscription's directory tenant changes, it resets the registration. You can't access the Azure Stack Hub Marketplace or report usage if you've changed the subscription's directory tenant. You need to re-register to fix this issue.

- Marketplace management still asks you to register and activate your Azure Stack Hub even when you've already registered your stamp using the disconnected process.

Cause: this is a known issue for disconnected environments. You can verify your registration status by [following these steps](#). In order to use Marketplace management, use [the offline tool](#).

Next steps

[Download marketplace items from Azure](#)

Region management in Azure Stack Hub

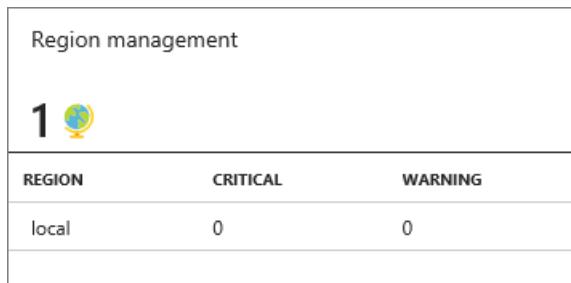
2 minutes to read • [Edit Online](#)

Azure Stack Hub uses the concept of *regions*, which are logical entities comprised of the hardware resources that make up the Azure Stack Hub infrastructure. In region management, you can find all resources that are required to successfully operate the Azure Stack Hub infrastructure.

One integrated system deployment (referred to as an *Azure Stack Hub cloud*) makes up a single region. Each Azure Stack Development Kit (ASDK) has one region, named **local**. If you deploy a second Azure Stack Hub integrated system, or you set up another instance of the ASDK on separate hardware, this Azure Stack Hub cloud is a different region.

Information available through the region management tile

Azure Stack Hub has a set of region management capabilities available in the **Region management** tile. This tile is available to an Azure Stack Hub operator on the default dashboard in the administrator portal. Through this tile, you can monitor and update your Azure Stack Hub region and its components, which are region-specific.



If you select a region in the **Region management** tile, you can access the following information:

NAME	HEALTH	ALERTS
Capacity	Healthy	0
Compute	Healthy	0
Infrastructure backup	Healthy	0
Key Vault	Healthy	0
Network	Healthy	0
Storage	Healthy	0

NAME	HEALTH	ALERTS
Authorization service (Administrator)	Healthy	0
Authorization service (User)	Healthy	0
Azure bridge	Healthy	0
Backup controller	Healthy	0
Compute controller	Healthy	0
Directory management	Healthy	0
Edge gateway	Healthy	0
Gallery service (Administrator)	Healthy	0

- The resource menu:** Access different infrastructure management areas, and view and manage user resources such as storage accounts and virtual networks.
- Alerts:** List system-wide alerts and provide details on each of those alerts.
- Updates:** View the current version of your Azure Stack Hub infrastructure, available updates, and the update history. You can also update your integrated system.
- Resource providers:** Manage the user functionality offered by the components required to run Azure Stack Hub. Each resource provider comes with an administrative experience. This experience can include alerts for

the specific provider, metrics, and other management capabilities specific to the resource provider.

5. **Infrastructure roles:** The components necessary to run Azure Stack Hub. Only the infrastructure roles that report alerts are listed. By selecting a role, you can view the alerts associated with the role and the role instances where this role is running.
6. **Properties:** The registration status and details of your environment in the region management blade. The status can be **Registered**, **Not registered**, or **Expired**. If registered, it also shows the Azure subscription ID that you used to register your Azure Stack Hub, along with the registration resource group and name.

Next steps

- [Monitor health and alerts in Azure Stack Hub](#)
- [Manage updates in Azure Stack Hub](#)

Connect Azure Stack Hub to Azure using Azure ExpressRoute

17 minutes to read • [Edit Online](#)

This article describes how to connect an Azure Stack Hub virtual network to an Azure virtual network using a [Microsoft Azure ExpressRoute](#) direct connection.

You can use this article as a tutorial and use the examples to set up the same test environment. Or, you can use the article as a walkthrough that guides you through setting up your own ExpressRoute environment.

Overview, assumptions, and prerequisites

Azure ExpressRoute lets you extend your on-premises networks into the Microsoft cloud over a private connection supplied by a connectivity provider. ExpressRoute is not a VPN connection over the public internet.

For more information about Azure ExpressRoute, see the [ExpressRoute overview](#).

Assumptions

This article assumes that:

- You have a working knowledge of Azure.
- You have a basic understanding of Azure Stack Hub.
- You have a basic understanding of networking.

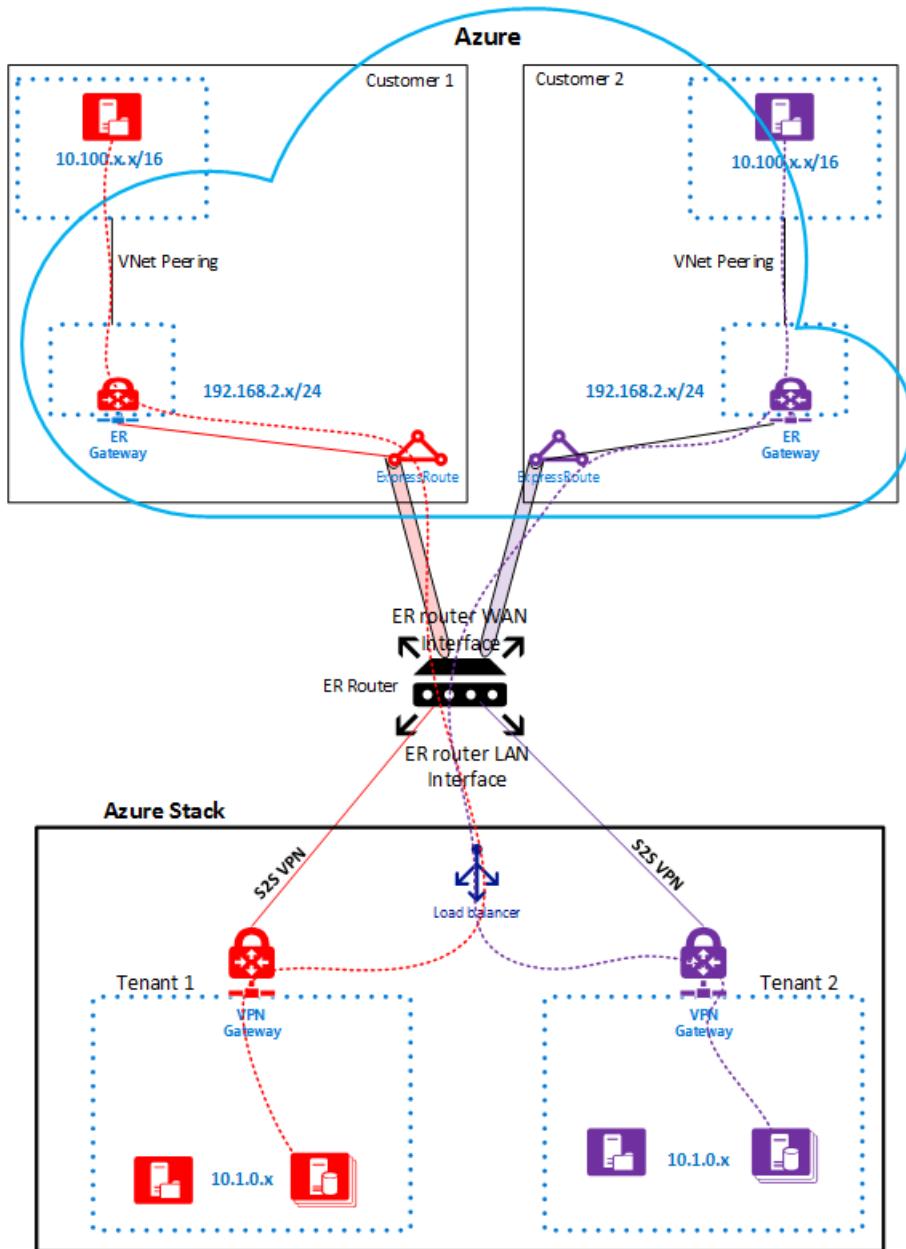
Prerequisites

To connect Azure Stack Hub and Azure using ExpressRoute, you must meet the following requirements:

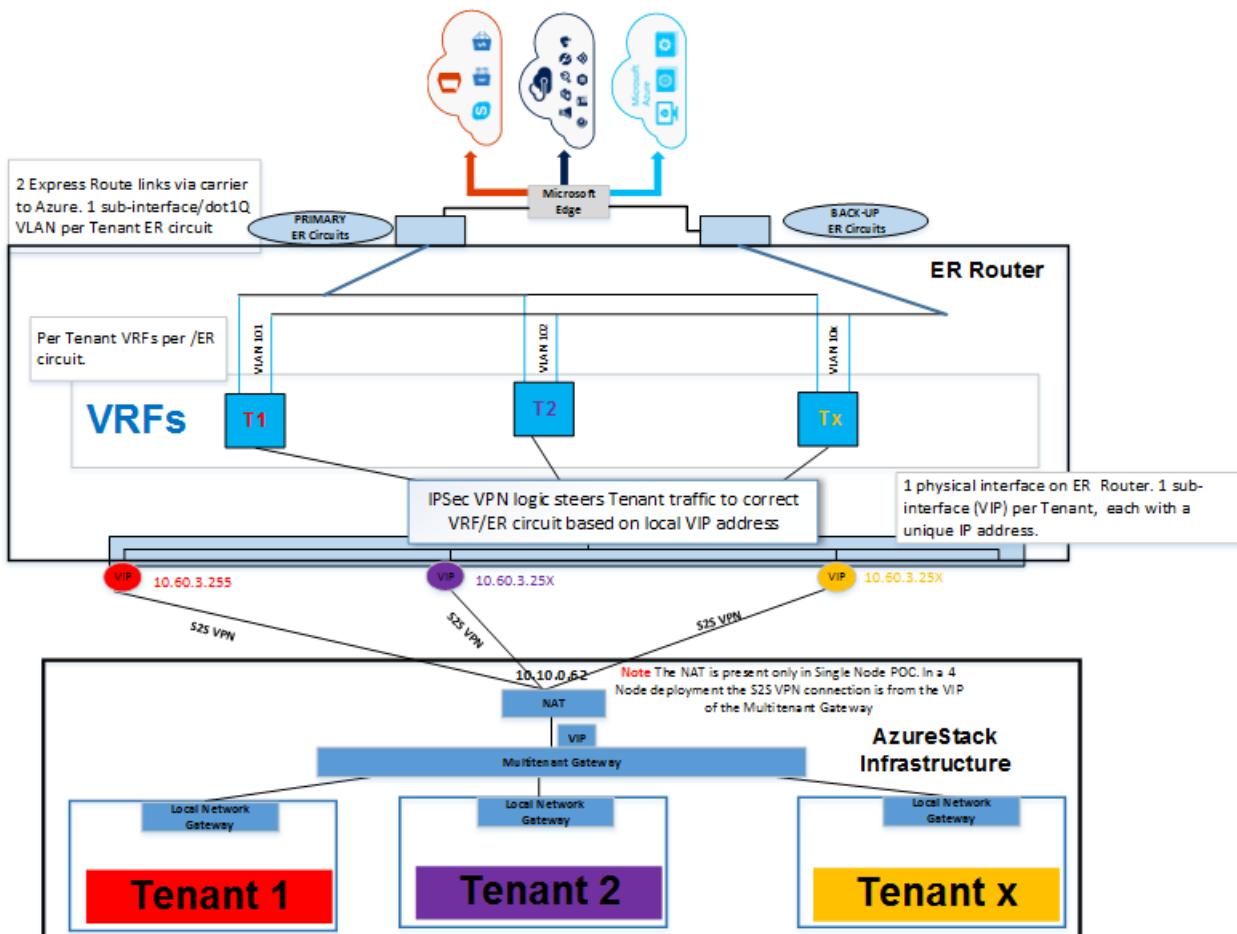
- A provisioned [ExpressRoute circuit](#) through a [connectivity provider](#).
- An Azure subscription to create an ExpressRoute circuit and VNets in Azure.
- A router that must:
 - Support site-to-site VPN connections between its LAN interface and Azure Stack Hub multi-tenant gateway.
 - Support creating multiple VRFs (Virtual Routing and Forwarding) if there is more than one tenant in your Azure Stack Hub deployment.
- A router that has:
 - A WAN port connected to the ExpressRoute circuit.
 - A LAN port connected to the Azure Stack Hub multi-tenant gateway.

ExpressRoute network architecture

The following figure shows the Azure Stack Hub and Azure environments after you finish setting up ExpressRoute using the examples in this article:



The following figure shows how multiple tenants connect from the Azure Stack Hub infrastructure through the ExpressRoute router to Azure:



The example in this article uses the same multi-tenant architecture shown in this diagram to connect Azure Stack Hub to Azure using ExpressRoute private peering. The connection is done using a site-to-site VPN connection from the virtual network gateway in Azure Stack Hub to an ExpressRoute router.

The steps in this article show you how to create an end-to-end connection between two VNets from two different tenants in Azure Stack Hub to corresponding VNets in Azure. Setting up two tenants is optional; you can also use these steps for a single tenant.

Configure Azure Stack Hub

To set up the Azure Stack Hub environment for the first tenant, use the following steps as a guide. If you're setting up more than one tenant, repeat these steps:

NOTE

These steps show how to create resources using the Azure Stack Hub portal, but you can also use PowerShell.



Before you begin

Before you start configuring Azure Stack Hub, you need:

- An Azure Stack Hub deployment.
- An offer in Azure Stack Hub that your users can subscribe to. For more information, see [Service, plan, offer, subscription overview](#).

Create network resources in Azure Stack Hub

Use the following procedures to create the required network resources in Azure Stack Hub for a tenant.

Create the virtual network and VM subnet

1. Sign in to the Azure Stack Hub user portal.
2. In the portal, select **+ Create a resource**.
3. Under **Azure Marketplace**, select **Networking**.
4. Under **Featured**, select **Virtual network**.
5. Under **Create virtual network**, enter the values shown in the following table into the appropriate fields:

FIELD	VALUE
Name	Tenant1VNet1
Address space	10.1.0.0/16
Subnet name	Tenant1-Sub1
Subnet address range	10.1.1.0/24

6. You should see the subscription you created earlier populated in the **Subscription** field. For the remaining fields:
 - Under **Resource group**, select **Create new** to create a new resource group or if you already have one, select **Use existing**.
 - Verify the default **Location**.
 - Click **Create**.
 - (Optional) Click **Pin to dashboard**.

Create the gateway subnet

1. Under **Virtual network**, select **Tenant1VNet1**.
2. Under **SETTINGS**, select **Subnets**.
3. Select **+ Gateway subnet** to add a gateway subnet to the virtual network.
4. The name of the subnet is set to **GatewaySubnet** by default. Gateway subnets are a special case and must use this name to function correctly.
5. Verify that the **Address range** is **10.1.0.0/24**.
6. Click **OK** to create the gateway subnet.

Create the virtual network gateway

1. In the Azure Stack Hub user portal, click **+ Create a resource**.
2. Under **Azure Marketplace**, select **Networking**.
3. Select **Virtual network gateway** from the list of network resources.
4. In the **Name** field, enter **GW1**.
5. Select **Virtual network**.
6. Select **Tenant1VNet1** from the drop-down list.
7. Select **Public IP address**, then **Choose public IP address**, and then click **Create new**.
8. In the **Name** field, type **GW1-PiP**, and then click **OK**.
9. The **VPN type** should have **Route-based** selected by default. Keep this setting.
10. Verify that **Subscription** and **Location** are correct. Click **Create**.

Create the local network gateway

The local network gateway resource identifies the remote gateway at the other end of the VPN connection. For this example, the remote end of the connection is the LAN sub-interface of the ExpressRoute router. For Tenant 1 in the previous diagram, the remote address is 10.60.3.255.

1. Sign in to the Azure Stack Hub user portal and select + **Create a resource**.
2. Under **Azure Marketplace**, select **Networking**.
3. Select **local network gateway** from the list of resources.
4. In the **Name** field, type **ER-Router-GW**.
5. For the **IP address** field, see the previous figure. The IP address of the ExpressRoute router LAN sub-interface for Tenant 1 is 10.60.3.255. For your own environment, enter the IP address of your router's corresponding interface.
6. In the **Address Space** field, enter the address space of the VNets that you want to connect to in Azure. The subnets for Tenant 1 are as follows:
 - 192.168.2.0/24 is the hub VNet in Azure.
 - 10.100.0.0/16 is the spoke VNet in Azure.

IMPORTANT

This example assumes that you are using static routes for the site-to-site VPN connection between the Azure Stack Hub gateway and the ExpressRoute router.

7. Verify that your **Subscription**, **Resource Group**, and **Location** are correct. Then select **Create**.

Create the connection

1. In the Azure Stack Hub user portal, select + **Create a resource**.
2. Under **Azure Marketplace**, select **Networking**.
3. Select **Connection** from the list of resources.
4. Under **Basics**, choose **Site-to-site (IPSec)** as the **Connection type**.
5. Select the **Subscription**, **Resource group**, and **Location**. Click **OK**.
6. Under **Settings**, select **Virtual network gateway**, and then select **GW1**.
7. Select **Local network gateway**, and then select **ER Router GW**.
8. In the **Connection name** field, enter **ConnectToAzure**.
9. In the **Shared key (PSK)** field, enter **abc123** and then select **OK**.
10. Under **Summary**, select **OK**.

Get the virtual network gateway public IP address

After you create the virtual network gateway, you can get the gateway's public IP address. Make a note of this address in case you need it later for your deployment. Depending on your deployment, this address is used as the **Internal IP address**.

1. In the Azure Stack Hub user portal, select **All resources**.
2. Under **All resources**, select the virtual network gateway, which is **GW1** in the example.
3. Under **Virtual network gateway**, select **Overview** from the list of resources. Alternatively, you can select **Properties**.
4. The IP address that you want to note is listed under **Public IP address**. For the example configuration, this address is 192.68.102.1.

Create a virtual machine (VM)

To test data traffic over the VPN connection, you need VMs to send and receive data in the Azure Stack Hub VNet.

Create a VM and deploy it to the VM subnet for your virtual network.

1. In the Azure Stack Hub user portal, select + **Create a resource**.
2. Under **Azure Marketplace**, select **Compute**.
3. In the list of VM images, select the **Windows Server 2016 Datacenter Eval** image.

NOTE

If the image used for this article is not available, ask your Azure Stack Hub operator to provide a different Windows Server image.

4. In **Create virtual machine**, select **Basics**, then type **VM01** as the **Name**.
5. Enter a valid user name and password. You'll use this account to sign in to the VM after it has been created.
6. Provide a **Subscription**, **Resource group**, and a **Location**. Select **OK**.
7. Under **Choose a size**, select a VM size for this instance, and then select **Select**.
8. Under **Settings**, confirm that:
 - The virtual network is **Tenant1VNet1**.
 - The subnet is set to **10.1.1.0/24**.Use the default settings and click **OK**.
9. Under **Summary**, review the VM configuration and then click **OK**.

To add more tenants, repeat the steps you followed in these sections:

- [Create the virtual network and VM subnet](#)
- [Create the gateway subnet](#)
- [Create the virtual network gateway](#)
- [Create the local network gateway](#)
- [Create the connection](#)
- [Create a virtual machine](#)

If you're using Tenant 2 as an example, remember to change the IP addresses to avoid overlaps.

Configure the NAT VM for gateway traversal

IMPORTANT

This section is for ASDK deployments only. The NAT is not needed for multi-node deployments.

The ASDK is self-contained and isolated from the network where the physical host is deployed. The VIP network that the gateways are connected to is not external; it is hidden behind a router performing Network Address Translation (NAT).

The router is the ASDK host running the Routing and Remote Access Services (RRAS) role. You must configure NAT on the ASDK host to enable the site-to-site VPN connection to connect on both ends.

Configure the NAT

1. Sign in to the Azure Stack Hub host computer with your admin account.
2. Run the script in an elevated PowerShell ISE. This script returns your **External BGP NAT address**.

```
Get-NetNatExternalAddress
```

3. To configure the NAT, copy and edit the following PowerShell script. Edit the script to replace the

External BGPNAT address and **Internal IP address** with the following example values:

- For *External BGPNAT address* use 10.10.0.62
- For *Internal IP address* use 192.168.102.1

Run the following script from an elevated PowerShell ISE:

```
$ExtBgpNat = 'External BGPNAT address'
$IntBgpNat = 'Internal IP address'

# Designate the external NAT address for the ports that use the IKE authentication.
Add-NetNatExternalAddress ` 
    -NatName BGPNAT ` 
    -IPAddress $Using:ExtBgpNat ` 
    -PortStart 499 ` 
    -PortEnd 501

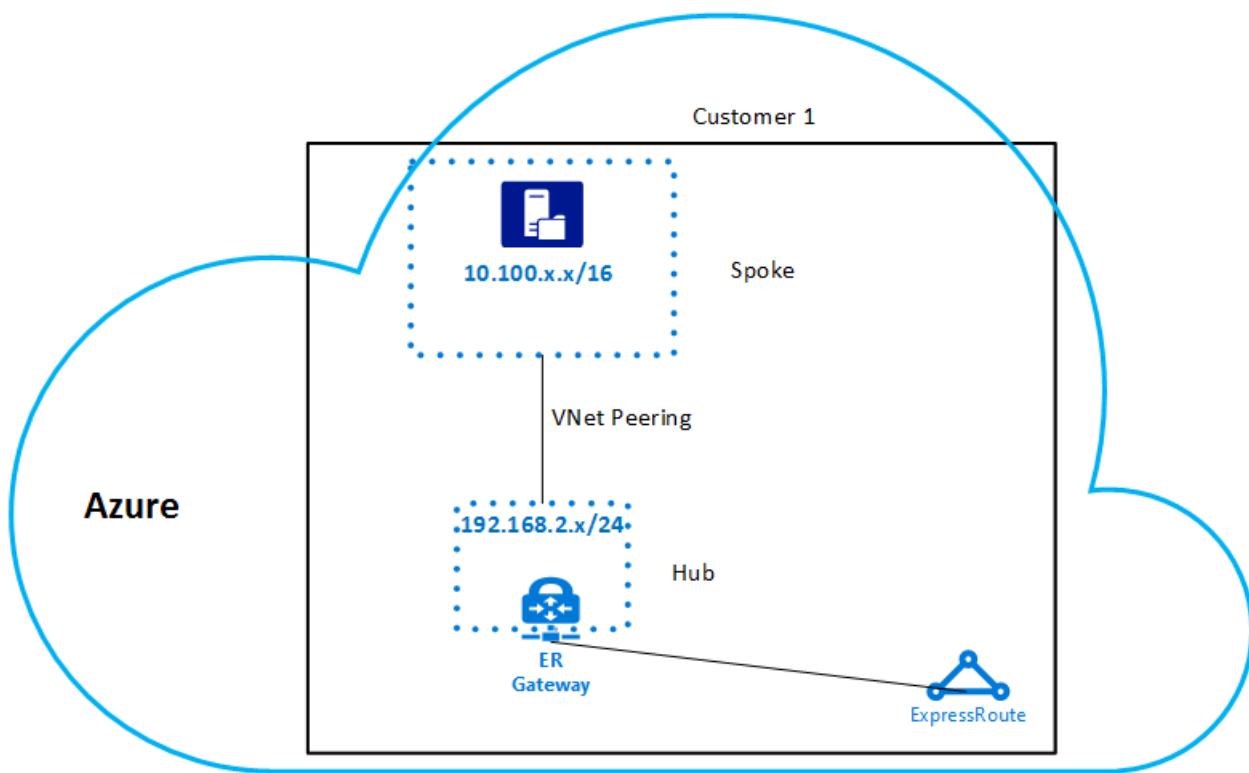
Add-NetNatExternalAddress ` 
    -NatName BGPNAT ` 
    -IPAddress $Using:ExtBgpNat ` 
    -PortStart 4499 ` 
    -PortEnd 4501

# Create a static NAT mapping to map the external address to the Gateway public IP address to map the
# ISAKMP port 500 for PHASE 1 of the IPSEC tunnel.
Add-NetNatStaticMapping ` 
    -NatName BGPNAT ` 
    -Protocol UDP ` 
    -ExternalIPAddress $Using:ExtBgpNat ` 
    -InternalIPAddress $Using:IntBgpNat ` 
    -ExternalPort 500 ` 
    -InternalPort 500

# Configure NAT traversal which uses port 4500 to establish the complete IPSEC tunnel over NAT
# devices.
Add-NetNatStaticMapping ` 
    -NatName BGPNAT ` 
    -Protocol UDP ` 
    -ExternalIPAddress $Using:ExtBgpNat ` 
    -InternalIPAddress $Using:IntBgpNat ` 
    -ExternalPort 4500 ` 
    -InternalPort 4500
```

Configure Azure

After you finish configuring Azure Stack Hub, you can deploy the Azure resources. The following figure shows an example of a tenant virtual network in Azure. You can use any name and addressing scheme for your VNet in Azure. However, the address range of the VNets in Azure and Azure Stack Hub must be unique and must not overlap:



The resources you deploy in Azure are similar to the resources you deployed in Azure Stack Hub. You deploy the following components:

- Virtual networks and subnets
- A gateway subnet
- A virtual network gateway
- A connection
- An ExpressRoute circuit

The example Azure network infrastructure is configured as follows:

- A standard hub (192.168.2.0/24) and spoke (10.100.0.0/16) VNet model. For more information about hub-spoke network topology, see [Implement a hub-spoke network topology in Azure](#).
- The workloads are deployed in the spoke VNet and the ExpressRoute circuit is connected to the hub VNet.
- The two VNets are connected using VNet peering.

Configure the Azure VNets

1. Sign in to the Azure portal with your Azure credentials.
2. Create the hub VNet using the 192.168.2.0/24 address range.
3. Create a subnet using the 192.168.2.0/25 address range, and add a gateway subnet using the 192.168.2.128/27 address range.
4. Create the spoke VNet and subnet using the 10.100.0.0/16 address range.

For more information about creating virtual networks in Azure, see [Create a virtual network](#).

Configure an ExpressRoute circuit

1. Review the ExpressRoute prerequisites in [ExpressRoute prerequisites & checklist](#).
2. Follow the steps in [Create and modify an ExpressRoute circuit](#) to create an ExpressRoute circuit using your Azure subscription.

NOTE

Give the service key for your circuit to your service so they can set up your ExpressRoute circuit at their end.

3. Follow the steps in [Create and modify peering for an ExpressRoute circuit](#) to configure private peering on the ExpressRoute circuit.

Create the virtual network gateway

Follow the steps in [Configure a virtual network gateway for ExpressRoute using PowerShell](#) to create a virtual network gateway for ExpressRoute in the hub VNet.

Create the connection

To link the ExpressRoute circuit to the hub VNet, follow the steps in [Connect a virtual network to an ExpressRoute circuit](#).

Peer the VNets

Peer the hub and spoke VNets using the steps in [Create a virtual network peering using the Azure portal](#). When configuring VNet peering, make sure you use the following options:

- From the hub to the spoke, **Allow gateway transit**.
- From the spoke to the hub, **Use remote gateway**.

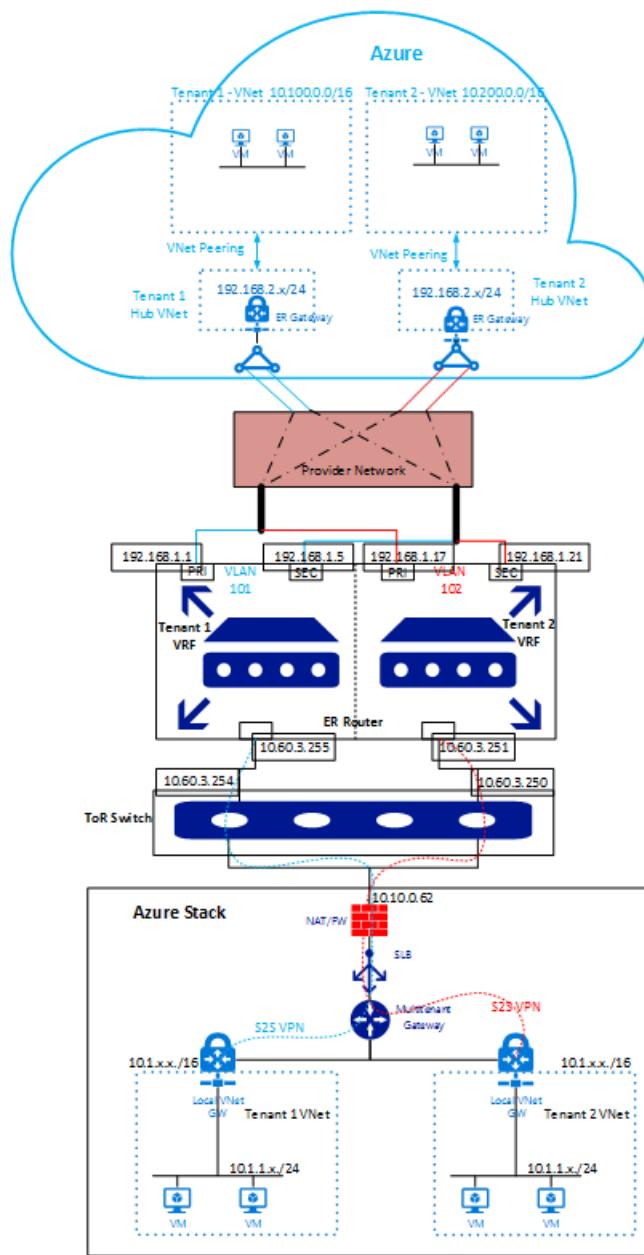
Create a virtual machine

Deploy your workload VMs into the spoke VNet.

Repeat these steps for any additional tenant VNets you want to connect in Azure through their respective ExpressRoute circuits.

Configure the router

You can use the following ExpressRoute router configuration diagram as a guide for configuring your ExpressRoute Router. This figure shows two tenants (Tenant 1 and Tenant 2) with their respective ExpressRoute circuits. Each tenant is linked to their own VRF (Virtual Routing and Forwarding) in the LAN and WAN side of the ExpressRoute router. This configuration ensures end-to-end isolation between the two tenants. Take note of the IP addresses used in the router interfaces as you follow the configuration example.



You can use any router that supports IKEv2 VPN and BGP to terminate the site-to-site VPN connection from Azure Stack Hub. The same router is used to connect to Azure using an ExpressRoute circuit.

The following Cisco ASR 1000 Series Aggregation Services Router configuration example supports the network infrastructure shown in the *ExpressRoute router configuration* diagram.

```

ip vrf Tenant 1
description Routing Domain for PRIVATE peering to Azure for Tenant 1
rd 1:1
!
ip vrf Tenant 2
description Routing Domain for PRIVATE peering to Azure for Tenant 2
rd 1:5
!
crypto ikev2 proposal V2-PROPOSAL2
description IKEv2 proposal for Tenant 1
encryption aes-cbc-256
integrity sha256
group 2
crypto ikev2 proposal V4-PROPOSAL2
description IKEv2 proposal for Tenant 2
encryption aes-cbc-256
integrity sha256
group 2
!
```

```

crypto ikev2 policy V2-POLICY2
description IKEv2 Policy for Tenant 1
match fvrf Tenant 1
  match address local 10.60.3.255
  proposal V2-PROPOSAL2
description IKEv2 Policy for Tenant 2
crypto ikev2 policy V4-POLICY2
  match fvrf Tenant 2
  match address local 10.60.3.251
  proposal V4-PROPOSAL2
!
crypto ikev2 profile V2-PROFILE
description IKEv2 profile for Tenant 1
match fvrf Tenant 1
  match address local 10.60.3.255
  match identity remote any
  authentication remote pre-share key abc123
  authentication local pre-share key abc123
  ivrf Tenant 1
!
crypto ikev2 profile V4-PROFILE
description IKEv2 profile for Tenant 2
match fvrf Tenant 2
  match address local 10.60.3.251
  match identity remote any
  authentication remote pre-share key abc123
  authentication local pre-share key abc123
  ivrf Tenant 2
!
crypto ipsec transform-set V2-TRANSFORM2 esp-gcm 256
  mode tunnel
crypto ipsec transform-set V4-TRANSFORM2 esp-gcm 256
  mode tunnel
!
crypto ipsec profile V2-PROFILE
  set transform-set V2-TRANSFORM2
  set ikev2-profile V2-PROFILE
!
crypto ipsec profile V4-PROFILE
  set transform-set V4-TRANSFORM2
  set ikev2-profile V4-PROFILE
!
interface Tunnel10
description S2S VPN Tunnel for Tenant 1
  ip vrf forwarding Tenant 1
  ip address 11.0.0.2 255.255.255.252
  ip tcp adjust-mss 1350
  tunnel source TenGigabitEthernet0/1/0.211
  tunnel mode ipsec ipv4
  tunnel destination 10.10.0.62
  tunnel vrf Tenant 1
  tunnel protection ipsec profile V2-PROFILE
!
interface Tunnel20
description S2S VPN Tunnel for Tenant 2
  ip vrf forwarding Tenant 2
  ip address 11.0.0.2 255.255.255.252
  ip tcp adjust-mss 1350
  tunnel source TenGigabitEthernet0/1/0.213
  tunnel mode ipsec ipv4
  tunnel destination 10.10.0.62
  tunnel vrf VNET3
  tunnel protection ipsec profile V4-PROFILE
!
interface GigabitEthernet0/0/1
description PRIMARY ExpressRoute Link to AZURE over Equinix
no ip address
negotiation auto
!
```

```

interface GigabitEthernet0/0/1.100
description Primary WAN interface of Tenant 1
description PRIMARY ER link supporting Tenant 1 to Azure
encapsulation dot1Q 101
ip vrf forwarding Tenant 1
ip address 192.168.1.1 255.255.255.252
!
interface GigabitEthernet0/0/1.102
description Primary WAN interface of Tenant 2
description PRIMARY ER link supporting Tenant 2 to Azure
encapsulation dot1Q 102
ip vrf forwarding Tenant 2
ip address 192.168.1.17 255.255.255.252
!
interface GigabitEthernet0/0/2
description BACKUP ExpressRoute Link to AZURE over Equinix
no ip address
negotiation auto
!
interface GigabitEthernet0/0/2.100
description Secondary WAN interface of Tenant 1
description BACKUP ER link supporting Tenant 1 to Azure
encapsulation dot1Q 101
ip vrf forwarding Tenant 1
ip address 192.168.1.5 255.255.255.252
!
interface GigabitEthernet0/0/2.102
description Secondary WAN interface of Tenant 2
description BACKUP ER link supporting Tenant 2 to Azure
encapsulation dot1Q 102
ip vrf forwarding Tenant 2
ip address 192.168.1.21 255.255.255.252
!
interface TenGigabitEthernet0/1/0
description Downlink to ---Port 1/47
no ip address
!
interface TenGigabitEthernet0/1/0.211
description LAN interface of Tenant 1
description Downlink to --- Port 1/47.211
encapsulation dot1Q 211
ip vrf forwarding Tenant 1
ip address 10.60.3.255 255.255.255.254
!
interface TenGigabitEthernet0/1/0.213
description LAN interface of Tenant 2
description Downlink to --- Port 1/47.213
encapsulation dot1Q 213
ip vrf forwarding Tenant 2
ip address 10.60.3.251 255.255.255.254
!
router bgp 65530
bgp router-id <removed>
bgp log-neighbor-changes
description BGP neighbor config and route advertisement for Tenant 1 VRF
address-family ipv4 vrf Tenant 1
network 10.1.0.0 mask 255.255.0.0
network 10.60.3.254 mask 255.255.255.254
network 192.168.1.0 mask 255.255.255.252
network 192.168.1.4 mask 255.255.255.252
neighbor 10.10.0.62 remote-as 65100
neighbor 10.10.0.62 description VPN-BGP-PEER-for-Tenant 1
neighbor 10.10.0.62 ebgp-multipath 5
neighbor 10.10.0.62 activate
neighbor 10.60.3.254 remote-as 4232570301
neighbor 10.60.3.254 description LAN peer for CPEC:INET:2112 VRF
neighbor 10.60.3.254 activate
neighbor 10.60.3.254 route-map BLOCK-ALL out
neighbor 192.168.1.2 remote-as 12076

```

```

neighbor 192.168.1.2 remote-as 12076
neighbor 192.168.1.2 description PRIMARY ER peer for Tenant 1 to Azure
neighbor 192.168.1.2 ebgp-multipath 5
neighbor 192.168.1.2 activate
neighbor 192.168.1.2 soft-reconfiguration inbound
neighbor 192.168.1.2 route-map Tenant 1-ONLY out
neighbor 192.168.1.6 remote-as 12076
neighbor 192.168.1.6 description BACKUP ER peer for Tenant 1 to Azure
neighbor 192.168.1.6 ebgp-multipath 5
neighbor 192.168.1.6 activate
neighbor 192.168.1.6 soft-reconfiguration inbound
neighbor 192.168.1.6 route-map Tenant 1-ONLY out
maximum-paths 8
exit-address-family
!
description BGP neighbor config and route advertisement for Tenant 2 VRF
address-family ipv4 vrf Tenant 2
  network 10.1.0.0 mask 255.255.0.0
  network 10.60.3.250 mask 255.255.255.254
  network 192.168.1.16 mask 255.255.255.252
  network 192.168.1.20 mask 255.255.255.252
  neighbor 10.10.0.62 remote-as 65300
  neighbor 10.10.0.62 description VPN-BGP-PEER-for-Tenant 2
  neighbor 10.10.0.62 ebgp-multipath 5
  neighbor 10.10.0.62 activate
  neighbor 10.60.3.250 remote-as 4232570301
  neighbor 10.60.3.250 description LAN peer for CPEC:INET:2112 VRF
  neighbor 10.60.3.250 activate
  neighbor 10.60.3.250 route-map BLOCK-ALL out
  neighbor 192.168.1.18 remote-as 12076
  neighbor 192.168.1.18 description PRIMARY ER peer for Tenant 2 to Azure
  neighbor 192.168.1.18 ebgp-multipath 5
  neighbor 192.168.1.18 activate
  neighbor 192.168.1.18 soft-reconfiguration inbound
  neighbor 192.168.1.18 route-map VNET-ONLY out
  neighbor 192.168.1.22 remote-as 12076
  neighbor 192.168.1.22 description BACKUP ER peer for Tenant 2 to Azure
  neighbor 192.168.1.22 ebgp-multipath 5
  neighbor 192.168.1.22 activate
  neighbor 192.168.1.22 soft-reconfiguration inbound
  neighbor 192.168.1.22 route-map VNET-ONLY out
  maximum-paths 8
exit-address-family
!
ip forward-protocol nd
!
ip as-path access-list 1 permit ^$  

ip route vrf Tenant 1 10.1.0.0 255.255.0.0 Tunnel10  

ip route vrf Tenant 2 10.1.0.0 255.255.0.0 Tunnel20
!
ip prefix-list BLOCK-ALL seq 5 deny 0.0.0.0/0 le 32
!
route-map BLOCK-ALL permit 10
  match ip address prefix-list BLOCK-ALL
!
route-map VNET-ONLY permit 10
  match as-path 1
!
```

Test the connection

Test your connection after you establish the site-to-site connection and the ExpressRoute circuit.

Perform the following ping tests:

- Sign in to one of the VMs in your Azure VNet and ping the VM you created in Azure Stack Hub.

- Sign in to one of the VMs you created in Azure Stack Hub and ping the VM you created in the Azure VNet.

NOTE

To make sure you are sending traffic over the site-to-site and ExpressRoute connections, you must ping the dedicated IP (DIP) address of the VM at both ends and not the VIP address of the VM.

Allow ICMP in through the firewall

By default, Windows Server 2016 does not allow incoming ICMP packets through the firewall. For every VM that you use for ping tests, you must allow incoming ICMP packets. To create a firewall rule for ICMP, run the following cmdlet in an elevated PowerShell window:

```
# Create ICMP firewall rule.  
New-NetFirewallRule  
    -DisplayName "Allow ICMPv4-In"  
    -Protocol ICMPv4
```

Ping the Azure Stack Hub VM

1. Sign in to the Azure Stack Hub user portal.
2. Find the VM that you created and select it.
3. Select **Connect**.
4. From an elevated Windows or PowerShell command prompt, enter **ipconfig /all**. Note the IPv4 address returned in the output.
5. Ping the IPv4 address from the VM in the Azure VNet.

In the example environment, the IPv4 address is from the 10.1.1.x/24 subnet. In your environment, the address might be different, but it should be in the subnet you created for the tenant VNet subnet.

View data transfer statistics

If you want to know how much traffic is passing through your connection, you can find this information on the Azure Stack Hub user portal. Viewing data transfer statistics is also a good way to find out whether or not your ping test data went through the VPN and ExpressRoute connections:

1. Sign in to the Azure Stack Hub user portal and select **All resources**.
2. Navigate to the resource group for your VPN Gateway and select the **Connection** object type.
3. Select the **ConnectToAzure** connection from the list.
4. Under **Connections > Overview**, you can see statistics for **Data in** and **Data out**. You should see some non-zero values.

The screenshot shows the Microsoft Azure Stack interface. On the left, the navigation pane includes 'Resource groups', 'Tenants', 'Tags', 'SETTINGS', 'Quickstart', 'Resource costs', 'Deployments', 'Properties', 'Locks', and 'Automation script'. The main area displays 'Resource groups' under 'Tenants'. A specific resource group 'TenantA-RG' is selected, with its 'Overview' tab highlighted. The 'Essentials' section shows the following details:

NAME	TYPE	LOCATION
ER-RouterConnection	Connection	local
SQLVM01	Virtual machine	local
sqlvm01844	Network interface	local
SQLVM01-ip	Public IP address	local
SQLVM01-msg	Network security gr...	local
TenantA-GW	Virtual network gate...	local
TenantA-GW-pIP	Public IP address	local
TenantA-LocalNWGW	Local network gate...	local
tenantargdisks099	Storage account	local
tenantargdisks506	Storage account	local
TenantAVNet1	Virtual network	local

The 'ER-RouterConnection' row is selected. The 'SETTINGS' section includes 'Activity log', 'Access control (IAM)', 'Tags', 'Shared key', 'Properties', 'Locks', and 'Automation script'. On the right, a detailed view of the 'ER-RouterConnection' is shown, with the 'Overview' tab selected. Key statistics are displayed:

- Data in: 108.75 kB
- Data out: 48.51 kB

Other details include the resource group name (changed to TenantA-RG), status (Connected), location (local), subscription name (changed to TenantA-Sub), and ID.

Next steps

Deploy apps to Azure and Azure Stack Hub

Enable Azure CLI for Azure Stack Hub users

2 minutes to read • [Edit Online](#)

You can provide the CA root certificate to users of Azure Stack Hub so that they can enable Azure CLI on their development machines. Your users need the certificate to manage resources through CLI.

- **The Azure Stack Hub CA root certificate** is required if users are using CLI from a workstation outside the Azure Stack Development Kit (ASDK).
- **The virtual machine (VM) aliases endpoint** provides an alias, like "UbuntuLTS" or "Win2012Datacenter," that references an image publisher, offer, SKU, and version as a single parameter when deploying VMs.

The following sections describe how to get these values.

Export the Azure Stack Hub CA root certificate

If you're using an integrated system, you don't need to export the CA root certificate. You need to export the CA root certificate on the ASDK.

To export the ASDK root certificate in PEM format, sign in and run the following script:

```
$label = "AzureStackSelfSignedRootCert"
Write-Host "Getting certificate from the current user trusted store with subject CN=$label"
$root = Get-ChildItem Cert:\CurrentUser\Root | Where-Object Subject -eq "CN=$label" | select -First 1
if (-not $root)
{
    Write-Error "Certificate with subject CN=$label not found"
    return
}

Write-Host "Exporting certificate"
Export-Certificate -Type CERT -FilePath root.cer -Cert $root

Write-Host "Converting certificate to PEM format"
certutil -encode root.cer root.pem
```

Set up the VM aliases endpoint

Azure Stack Hub operators should set up a publicly accessible endpoint that hosts a VM alias file. The VM alias file is a JSON file that provides a common name for an image. You use the name when you deploy a VM as an Azure CLI parameter.

Before you add an entry to an alias file, make sure that you [download images from the Azure Marketplace](#) or have [published your own custom image](#). If you publish a custom image, make note of the publisher, offer, SKU, and version info that you specified during publishing. If it's an image from the marketplace, you can view the info by using the `Get-AzureVMImage` cmdlet.

A [sample alias file](#) with many common image aliases is available. You can use that as a starting point. Host this file in a space where your CLI clients can reach it. One way is to host the file in a blob storage account and share the URL with your users:

1. Download the [sample file](#) from GitHub.
2. Create a storage account in Azure Stack Hub. When that's done, create a blob container. Set the access policy to "public."

3. Upload the JSON file to the new container. When that's done, you can view the URL of the blob. Select the blob name and then select the URL from the blob properties.

Next steps

- [Deploy templates with Azure CLI](#)
- [Connect with PowerShell](#)
- [Manage user permissions](#)

Install PowerShell for Azure Stack Hub

7 minutes to read • [Edit Online](#)

Azure PowerShell provides a set of cmdlets that use the Azure Resource Manager model for managing your Azure Stack Hub resources.

To work with your cloud, you need to install Azure Stack Hub compatible PowerShell modules. Azure Stack Hub uses the **AzureRM** module instead of the newer **AzureAZ** module used in global Azure. You also need to use *API profiles* to specify the compatible endpoints for the Azure Stack Hub resource providers.

API profiles provide a way to manage version differences between Azure and Azure Stack Hub. An API version profile is a set of Azure Resource Manager PowerShell modules with specific API versions. Each cloud platform has a set of supported API version profiles. For example, Azure Stack Hub supports a specific profile version such as **2019-03-01-hybrid**. When you install a profile, the Azure Resource Manager PowerShell modules that correspond to the specified profile are installed.

You can install Azure Stack Hub compatible PowerShell modules in internet-connected, partially connected, or disconnected scenarios. This article walks you through the detailed instructions for these scenarios.

1. Verify your prerequisites

Before you get started with Azure Stack Hub and PowerShell, you must have the following prerequisites:

- **PowerShell Version 5.0**

To check your version, run `$PSVersionTable.PSVersion` and compare the **Major** version. If you don't have PowerShell 5.0, follow the [Installing Windows PowerShell](#).

NOTE

PowerShell 5.0 requires a Windows machine.

- **Run PowerShell in an elevated command prompt.**

- **PowerShell Gallery access**

You need access to the [PowerShell Gallery](#). The gallery is the central repository for PowerShell content.

The **PowerShellGet** module contains cmdlets for discovering, installing, updating, and publishing PowerShell artifacts. Examples of these artifacts are modules, DSC resources, role capabilities, and scripts from the PowerShell Gallery and other private repositories. If you're using PowerShell in a disconnected scenario, you must retrieve resources from a machine with a connection to the internet and store them in a location accessible to your disconnected machine.

2. Validate the PowerShell Gallery accessibility

Validate if PS Gallery is registered as a repository.

NOTE

This step requires internet access.

Open an elevated PowerShell prompt, and run the following cmdlets:

```
Import-Module -Name PowerShellGet -ErrorAction Stop
Import-Module -Name PackageManagement -ErrorAction Stop
Get-PSRepository -Name "PSGallery"
```

If the repository isn't registered, open an elevated PowerShell session and run the following command:

```
Register-PSRepository -Default
Set-PSRepository -Name "PSGallery" -InstallationPolicy Trusted
```

3. Uninstall existing versions of the Azure Stack Hub PowerShell modules

Before installing the required version, make sure that you uninstall any previously installed Azure Stack Hub AzureRM PowerShell modules. Uninstall the modules by using one of the following two methods:

1. To uninstall the existing AzureRM PowerShell modules, close all the active PowerShell sessions, and run the following cmdlets:

```
Get-Module -Name Azs.* -ListAvailable | Uninstall-Module -Force -Verbose
Get-Module -Name Azure* -ListAvailable | Uninstall-Module -Force -Verbose
```

If you hit an error such as 'The module is already in use', close the PowerShell sessions that are using the modules and rerun the above script.

2. Delete all the folders that start with `Azure` or `Azs.` from the `C:\Program Files\WindowsPowerShell\Modules` and `C:\Users\{yourusername}\Documents\WindowsPowerShell\Modules` folders. Deleting these folders removes any existing PowerShell modules.

4. Connected: Install PowerShell for Azure Stack Hub with internet connectivity

The API version profile and Azure Stack Hub PowerShell modules you require will depend on the version of Azure Stack Hub you're running.

Install Azure Stack Hub PowerShell

Run the following PowerShell script to install these modules on your development workstation:

- For Azure Stack Hub 1910 or later:

```
# Install the AzureRM.BootStrapper module. Select Yes when prompted to install NuGet
Install-Module -Name AzureRM.BootStrapper

# Install and import the API Version Profile required by Azure Stack Hub into the current PowerShell
# session.
Use-AzureRmProfile -Profile 2019-03-01-hybrid -Force
Install-Module -Name AzureStack -RequiredVersion 1.8.0
```

- For Azure Stack Hub 1908 or after 1903:

```
# Install the AzureRM.BootStrapper module. Select Yes when prompted to install NuGet  
Install-Module -Name AzureRM.BootStrapper  
  
# Install and import the API Version Profile required by Azure Stack Hub into the current PowerShell  
session.  
Use-AzureRmProfile -Profile 2019-03-01-hybrid -Force  
Install-Module -Name AzureStack -RequiredVersion 1.7.2
```

- For Azure Stack Hub version 1903 or earlier, only install the two modules below:

```
# Install and import the API Version Profile required by Azure Stack Hub into the current PowerShell  
session.  
  
Install-Module -Name AzureRM -RequiredVersion 2.4.0  
Install-Module -Name AzureStack -RequiredVersion 1.7.1
```

NOTE

- Azure Stack Hub module version 1.8.0 is a breaking change release. Refer to the [release note](#) for details.
- The Azure Stack Hub module version 1.7.2 is a breaking change release. To migrate from Azure Stack Hub 1.6.0, please refer to the [migration guide](#).
- The AzureRM module version 2.4.0 comes with a breaking change for the cmdlet Remove-AzureRmStorageAccount. This cmdlet expects `-Force` parameter to be specified for removing the storage account without confirmation.
- You don't need to install **AzureRM.BootStrapper** to install the modules for Azure Stack Hub version 1901 or later.
- Don't install the 2018-03-01-hybrid profile in addition to using the above AzureRM modules on Azure Stack Hub version 1901 or later.

Confirm the installation of PowerShell

Confirm the installation by running the following command:

```
Get-Module -Name "Azure*" -ListAvailable  
Get-Module -Name "Azs*" -ListAvailable
```

If the installation is successful, the `AzureRM` and `AzureStack` modules are displayed in the output.

5. Disconnected: Install PowerShell without an internet connection

In a disconnected scenario, you first download the PowerShell modules to a machine that has internet connectivity. Then, you transfer them to the Azure Stack Development Kit (ASDK) for installation.

Sign in to a computer with internet connectivity and use the following scripts to download the Azure Resource Manager and Azure Stack Hub packages, depending on your version of Azure Stack Hub.

Installation has four steps:

1. Install Azure Stack Hub PowerShell to a connected machine.
2. Enable additional storage features.
3. Transport the PowerShell packages to your disconnected workstation.
4. Manually bootstrap the NuGet provider on your disconnected workstation.
5. Confirm the installation of PowerShell.

Install Azure Stack Hub PowerShell

- Azure Stack Hub 1910 or later.

```
Import-Module -Name PowerShellGet -ErrorAction Stop
Import-Module -Name PackageManagement -ErrorAction Stop

$Path = "<Path that is used to save the packages>"
Save-Package -ProviderName NuGet -Source https://www.powershellgallery.com/api/v2 -Name AzureRM -
Path $Path -Force -RequiredVersion 2.5.0
Save-Package -ProviderName NuGet -Source https://www.powershellgallery.com/api/v2 -Name AzureStack -
Path $Path -Force -RequiredVersion 1.8.0
```

- For Azure Stack Hub 1908 or after 1903:

```
Import-Module -Name PowerShellGet -ErrorAction Stop
Import-Module -Name PackageManagement -ErrorAction Stop

$Path = "<Path that is used to save the packages>"
Save-Package -ProviderName NuGet -Source https://www.powershellgallery.com/api/v2 -Name AzureRM -
Path $Path -Force -RequiredVersion 2.5.0
Save-Package -ProviderName NuGet -Source https://www.powershellgallery.com/api/v2 -Name AzureStack -
Path $Path -Force -RequiredVersion 1.7.2
```

- Azure Stack Hub 1903 or earlier.

```
Import-Module -Name PowerShellGet -ErrorAction Stop
Import-Module -Name PackageManagement -ErrorAction Stop

$Path = "<Path that is used to save the packages>"
Save-Package -ProviderName NuGet -Source https://www.powershellgallery.com/api/v2 -Name AzureRM -
Path $Path -Force -RequiredVersion 2.4.0
Save-Package -ProviderName NuGet -Source https://www.powershellgallery.com/api/v2 -Name AzureStack -
Path $Path -Force -RequiredVersion 1.7.1
```

NOTE

- Azure Stack Hub module version 1.8.0 is a breaking change release. Refer to the [release note](#) for details. The Azure Stack Hub module version 1.7.1 is a breaking change. To migrate from Azure Stack Hub 1.6.0 please refer to the [migration guide](#).

NOTE

On machines without an internet connection, we recommend executing the following cmdlet for disabling the telemetry data collection. You may experience a performance degradation of the cmdlets without disabling the telemetry data collection. This is applicable only for the machines without internet connections

```
Disable-AzureRmDataCollection
```

Add your packages to your workstation

1. Copy the downloaded packages to a USB device.
2. Sign in to the disconnected workstation and copy the packages from the USB device to a location on the workstation.
3. Manually bootstrap the NuGet provider on your disconnected workstation. For instructions, see [Manually bootstrapping the NuGet provider on a machine that isn't connected to the internet](#).

4. Register this location as the default repository and install the AzureRM and AzureStack modules from this repository:

```
# requires -Version 5
# requires -RunAsAdministrator
# requires -Module PowerShellGet
# requires -Module PackageManagement

$SourceLocation = "<Location on the development kit that contains the PowerShell packages>"
$RepoName = "MyNugetSource"

Register-PSRepository -Name $RepoName -SourceLocation $SourceLocation -InstallationPolicy Trusted

Install-Module -Name AzureRM -Repository $RepoName

Install-Module -Name AzureStack -Repository $RepoName
```

Confirm the installation of PowerShell

Confirm the installation by running the following command:

```
Get-Module -Name "Azure*" -ListAvailable
Get-Module -Name "Azs*" -ListAvailable
```

6. Configure PowerShell to use a proxy server

In scenarios that require a proxy server to access the internet, you first configure PowerShell to use an existing proxy server:

1. Open an elevated PowerShell prompt.
2. Run the following commands:

```
#To use Windows credentials for proxy authentication
[System.Net.WebRequest]::DefaultWebProxy.Credentials =
[System.Net.CredentialCache]::DefaultCredentials

#Alternatively, to prompt for separate credentials that can be used for #proxy authentication
[System.Net.WebRequest]::DefaultWebProxy.Credentials = Get-Credential
```

Next steps

- [Download Azure Stack Hub tools from GitHub](#)
- [Configure the Azure Stack Hub user's PowerShell environment](#)
- [Configure the Azure Stack Hub operator's PowerShell environment](#)
- [Manage API version profiles in Azure Stack Hub](#)

Download Azure Stack Hub tools from GitHub

2 minutes to read • [Edit Online](#)

AzureStack-Tools is a [GitHub repository](#) that hosts PowerShell modules for managing and deploying resources to Azure Stack Hub. If you're planning to establish VPN connectivity, you can download these PowerShell modules to the Azure Stack Development Kit (ASDK), or to a Windows-based external client. To get these tools, clone the GitHub repository or download the **AzureStack-Tools** folder by running the following script:

```
# Change directory to the root directory.  
cd \  
  
# Download the tools archive.  
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12  
invoke-webrequest `  
https://github.com/Azure/AzureStack-Tools/archive/master.zip `  
-OutFile master.zip  
  
# Expand the downloaded files.  
expand-archive master.zip `  
-DestinationPath . `  
-Force  
  
# Change to the tools directory.  
cd AzureStack-Tools-master
```

Functionality provided by the modules

The **AzureStack-Tools** repository has PowerShell modules that support the following functionalities for Azure Stack Hub:

FUNCTIONALITY	DESCRIPTION	WHO CAN USE THIS MODULE?
Cloud capabilities	Use this module to get the cloud capabilities of a cloud. For example, you can get cloud capabilities like API version and Azure Resource Manager resources. You can also get the VM extensions for Azure Stack Hub and Azure clouds.	Cloud operators and users
Resource Manager policy for Azure Stack Hub	Use this module to configure an Azure subscription or an Azure resource group with the same versioning and service availability as Azure Stack Hub.	Cloud operators and users
Register with Azure	Use this module to register your ASDK instance with Azure. After registering, you can download Azure Marketplace items use them in Azure Stack Hub.	Cloud operators

FUNCTIONALITY	DESCRIPTION	WHO CAN USE THIS MODULE?
Azure Stack Hub deployment	Use this module to prepare the Azure Stack Hub host computer to deploy and redeploy by using the Azure Stack Hub virtual hard disk (VHD) image.	Cloud operators
Connecting to Azure Stack Hub	Use this module to configure VPN connectivity to Azure Stack Hub.	Cloud operators and users
Template validator	Use this module to verify if an existing or a new template can be deployed to Azure Stack Hub.	Cloud operators and users

Next steps

- [Get started with PowerShell on Azure Stack Hub.](#)
- [Configure the Azure Stack Hub user's PowerShell environment](#)
- [Connect to Azure Stack Development Kit over a VPN.](#)

Connect to Azure Stack Hub with PowerShell

2 minutes to read • [Edit Online](#)

You can configure Azure Stack Hub to use PowerShell to manage resources like creating offers, plans, quotas, and alerts. This topic helps you configure the operator environment.

Prerequisites

Run the following prerequisites either from the [Azure Stack Development Kit \(ASDK\)](#) or from a Windows-based external client if you're [connected to the ASDK through VPN](#).

- Install [Azure Stack Hub-compatible Azure PowerShell modules](#).
- Download the [tools required to work with Azure Stack Hub](#).

Connect with Azure AD

To configure the Azure Stack Hub operator environment with PowerShell, run one of the scripts below. Replace the Azure Active Directory (Azure AD) tenantName and Azure Resource Manager endpoint values with your own environment configuration.

NOTE

If your session expires, your password has changed, or you simply wish to switch accounts, run the following cmdlet before you sign in using Add-AzureRmAccount: `Remove-AzureRmAccount -Scope Process`

```
# Register an Azure Resource Manager environment that targets your Azure Stack Hub instance. Get your
# Azure Resource Manager endpoint value from your service provider.
Add-AzureRmEnvironment -Name "AzureStackAdmin" -ArmEndpoint
"https://adminmanagement.local.azurestack.external" `

-AzureKeyVaultDnsSuffix adminvault.local.azurestack.external `

-AzureKeyVaultServiceEndpointResourceId https://adminvault.local.azurestack.external

# Set your tenant name.
$AuthEndpoint = (Get-AzureRmEnvironment -Name "AzureStackAdmin").ActiveDirectoryAuthority.TrimEnd('/')
$AADTenantName = "<myDirectoryTenantName>.onmicrosoft.com"
$TenantId = (Invoke-RestMethod "$($AuthEndpoint)/$($AADTenantName)/.well-known/openid-
configuration").issuer.TrimEnd('/').Split('/')[-1]

# After signing in to your environment, Azure Stack Hub cmdlets
# can be easily targeted at your Azure Stack Hub instance.
Add-AzureRmAccount -EnvironmentName "AzureStackAdmin" -TenantId $TenantId
```

Connect with AD FS

Connect to the Azure Stack Hub operator environment with PowerShell with Azure Active Directory Federated Services (Azure AD FS). For the ASDK, this Azure Resource Manager endpoint is set to `https://adminmanagement.local.azurestack.external`. To get the Azure Resource Manager endpoint for Azure Stack Hub integrated systems, contact your service provider.

```
# Register an Azure Resource Manager environment that targets your Azure Stack Hub instance. Get your Azure  
Resource Manager endpoint value from your service provider.  
Add-AzureRMEvironment -Name "AzureStackAdmin" -ArmEndpoint  
"https://adminmanagement.local.azurestack.external" `  
-AzureKeyVaultDnsSuffix adminvault.local.azurestack.external `  
-AzureKeyVaultServiceEndpointResourceId https://adminvault.local.azurestack.external  
  
# Sign in to your environment.  
Login-AzureRmAccount -EnvironmentName "AzureStackAdmin"
```

NOTE

AD FS only supports interactive authentication with user identities. If a credential object is required, you must use a service principal (SPN). For more information on setting up a service principal with Azure Stack Hub and AD FS as your identity management service, see [Manage an AD FS service principal](#).

Test the connectivity

Now that you've got everything set-up, use PowerShell to create resources within Azure Stack Hub. For example, you can create a resource group for an app and add a virtual machine. Use the following command to create a resource group named **MyResourceGroup**.

```
New-AzureRmResourceGroup -Name "MyResourceGroup" -Location "Local"
```

Next steps

- [Develop templates for Azure Stack Hub.](#)
- [Deploy templates with PowerShell.](#)
 - [Azure Stack Hub Module Reference](#).

Use the privileged endpoint in Azure Stack Hub

7 minutes to read • [Edit Online](#)

As an Azure Stack Hub operator, you should use the administrator portal, PowerShell, or Azure Resource Manager APIs for most day-to-day management tasks. However, for some less common operations, you need to use the *privileged endpoint* (PEP). The PEP is a pre-configured remote PowerShell console that provides you with just enough capabilities to help you do a required task. The endpoint uses [PowerShell JEA \(Just Enough Administration\)](#) to expose only a restricted set of cmdlets. To access the PEP and invoke the restricted set of cmdlets, a low-privileged account is used. No admin accounts are required. For additional security, scripting isn't allowed.

You can use the PEP to perform these tasks:

- Low-level tasks, such as [collecting diagnostic logs](#).
- Many post-deployment datacenter integration tasks for integrated systems, such as adding Domain Name System (DNS) forwarders after deployment, setting up Microsoft Graph integration, Active Directory Federation Services (AD FS) integration, certificate rotation, and so on.
- To work with support to obtain temporary, high-level access for in-depth troubleshooting of an integrated system.

The PEP logs every action (and its corresponding output) that you perform in the PowerShell session. This provides full transparency and complete auditing of operations. You can keep these log files for future audits.

NOTE

In the Azure Stack Development Kit (ASDK), you can run some of the commands available in the PEP directly from a PowerShell session on the development kit host. However, you may want to test some operations using the PEP, such as log collection, because this is the only method available to perform certain operations in an integrated systems environment.

Access the privileged endpoint

You access the PEP through a remote PowerShell session on the virtual machine (VM) that hosts the PEP. In the ASDK, this VM is named **AzS-ERCS01**. If you're using an integrated system, there are three instances of the PEP, each running inside a VM (*Prefix-ERCS01*, *Prefix-ERCS02*, or *Prefix-ERCS03*) on different hosts for resiliency.

Before you begin this procedure for an integrated system, make sure you can access the PEP either by IP address or through DNS. After the initial deployment of Azure Stack Hub, you can access the PEP only by IP address because DNS integration isn't set up yet. Your OEM hardware vendor will provide you with a JSON file named **AzureStackStampDeploymentInfo** that contains the PEP IP addresses.

You may also find the IP address in the Azure Stack Hub administrator portal. Open the portal, for example, <https://adminportal.local.azurestack.external>. Select **Region Management > Properties**.

You will need set your current culture setting to `en-US` when running the privileged endpoint, otherwise cmdlets such as `Test-AzureStack` or `Get-AzureStackLog` will not work as expected.

NOTE

For security reasons, we require that you connect to the PEP only from a hardened VM running on top of the hardware lifecycle host, or from a dedicated and secure computer, such as a [Privileged Access Workstation](#). The original configuration of the hardware lifecycle host must not be modified from its original configuration (including installing new software) or used to connect to the PEP.

1. Establish the trust.

- On an integrated system, run the following command from an elevated Windows PowerShell session to add the PEP as a trusted host on the hardened VM running on the hardware lifecycle host or the Privileged Access Workstation.

```
winrm s winrm/config/client '@{TrustedHosts=<IP Address of Privileged Endpoint>}'
```

- If you're running the ASDK, sign in to the development kit host.

2. On the hardened VM running on the hardware lifecycle host or the Privileged Access Workstation, open a Windows PowerShell session. Run the following commands to establish a remote session on the VM that hosts the PEP:

- On an integrated system:

```
$cred = Get-Credential  
  
$pep = New-PSSession -ComputerName <IP_address_of_ERCS> -ConfigurationName PrivilegedEndpoint -  
Credential $cred -SessionOption (New-PSSessionOption -Culture en-US -UICulture en-US)  
Enter-PSSession $pep
```

The `ComputerName` parameter can be either the IP address or the DNS name of one of the VMs that hosts the PEP.

NOTE

Azure Stack Hub doesn't make a remote call when validating the PEP credential. It relies on a locally-stored RSA public key to do that.

- If you're running the ASDK:

```
$cred = Get-Credential  
  
$pep = New-PSSession -ComputerName azs-ercs01 -ConfigurationName PrivilegedEndpoint -Credential  
$cred -SessionOption (New-PSSessionOption -Culture en-US -UICulture en-US)  
Enter-PSSession $pep
```

- When prompted, use the following credentials:

- **User name:** Specify the CloudAdmin account, in the format **<Azure Stack Hub domain>\clouadmin**. (For ASDK, the user name is **azurestack\clouadmin**.)
- **Password:** Enter the same password that was provided during installation for the AzureStackAdmin domain administrator account.

NOTE

If you're unable to connect to the ERCS endpoint, retry steps one and two with another ERCS VM IP address.

3. After you connect, the prompt will change to **[IP address or ERCS VM name]: PS>** or to **[azs-ercs01]: PS>**, depending on the environment. From here, run `Get-Command` to view the list of available cmdlets.

Many of these cmdlets are intended only for integrated system environments (such as the cmdlets related to datacenter integration). In the ASDK, the following cmdlets have been validated:

- Clear-Host
- Close-PrivilegedEndpoint
- Exit-PSSession
- Get-AzureStackLog
- Get-AzureStackStampInformation
- Get-Command
- Get-FormatData
- Get-Help
- Get-ThirdPartyNotices
- Measure-Object
- New-CloudAdminUser
- Out-Default
- Remove-CloudAdminUser
- Select-Object
- Set-CloudAdminUserPassword
- Test-AzureStack
- Stop-AzureStack
- Get-ClusterLog

Tips for using the privileged endpoint

As mentioned above, the PEP is a [PowerShell JEA](#) endpoint. While providing a strong security layer, a JEA endpoint reduces some of the basic PowerShell capabilities, such as scripting or tab completion. If you try any type of script operation, the operation fails with the error **ScriptsNotAllowed**. This failure is expected behavior.

For instance, to get the list of parameters for a given cmdlet, run the following command:

```
Get-Command <cmdlet_name> -Syntax
```

Alternatively, you can use the [Import-PSSession](#) cmdlet to import all the PEP cmdlets into the current session on your local machine. By doing so, all cmdlets and functions of the PEP are now available on your local machine, together with tab completion and, more in general, scripting.

To import the PEP session on your local machine, do the following steps:

1. Establish the trust.
 - On an integrated system, run the following command from an elevated Windows PowerShell session to add the PEP as a trusted host on the hardened VM running on the hardware lifecycle host or the Privileged Access Workstation.

```
winrm s winrm/config/client '@{TrustedHosts="
```

- If you're running the ASDK, sign in to the development kit host.
2. On the hardened VM running on the hardware lifecycle host or the Privileged Access Workstation, open a Windows PowerShell session. Run the following commands to establish a remote session on the virtual machine that hosts the PEP:
- On an integrated system:

```
$cred = Get-Credential  
  
$session = New-PSSession -ComputerName <IP_address_of_ERCS> `  
-ConfigurationName PrivilegedEndpoint -Credential $cred
```

The `ComputerName` parameter can be either the IP address or the DNS name of one of the VMs that hosts the PEP.

- If you're running the ASDK:

```
$cred = Get-Credential  
  
$session = New-PSSession -ComputerName azs-ercs01 `  
-ConfigurationName PrivilegedEndpoint -Credential $cred
```

When prompted, use the following credentials:

- **User name:** Specify the CloudAdmin account, in the format **<Azure Stack Hub domain>\clouddadmin**. (For ASDK, the user name is **azurestack\clouddadmin**.)
- **Password:** Enter the same password that was provided during installation for the AzureStackAdmin domain administrator account.

3. Import the PEP session into your local machine:

```
Import-PSSession $session
```

4. Now, you can use tab-completion and do scripting as usual on your local PowerShell session with all the functions and cmdlets of the PEP, without decreasing the security posture of Azure Stack Hub. Enjoy!

Close the privileged endpoint session

As mentioned earlier, the PEP logs every action (and its corresponding output) that you do in the PowerShell session. You must close the session by using the `Close-PrivilegedEndpoint` cmdlet. This cmdlet correctly closes the endpoint, and transfers the log files to an external file share for retention.

To close the endpoint session:

1. Create an external file share that's accessible by the PEP. In a development kit environment, you can just create a file share on the development kit host.
2. Run the following cmdlet:

```
Close-PrivilegedEndpoint -TranscriptsPathDestination "\\\fileshareIP\SharedFolder" -Credential Get-Credential
```

The cmdlet uses the parameters in the following table:

PARAMETER	DESCRIPTION	TYPE	REQUIRED
<i>TranscriptsPathDestination</i>	Path to the external file share defined as "fileshareIP\sharefoldername"	String	Yes
<i>Credential</i>	Credentials to access the file share	SecureString	Yes

After the transcript log files are successfully transferred to the file share, they're automatically deleted from the PEP.

NOTE

If you close the PEP session by using the cmdlets `Exit-PSSession` or `Exit`, or you just close the PowerShell console, the transcript logs don't transfer to a file share. They remain in the PEP. The next time you run `Close-PrivilegedEndpoint` and include a file share, the transcript logs from the previous session(s) will also transfer. Don't use `Exit-PSSession` or `Exit` to close the PEP session; use `Close-PrivilegedEndpoint` instead.

Next steps

[Azure Stack Hub diagnostic tools](#)

Manage updates in Azure Stack Hub

4 minutes to read • [Edit Online](#)

Full and express updates, hotfixes, as well as driver and firmware updates from the original equipment manufacturer (OEM) all help keep Azure Stack Hub up to date. This article explains the different types of updates, when to expect their release, and where to find more about the current release.

NOTE

You can't apply Azure Stack Hub update packages to the Azure Stack Development Kit (ASDK). The update packages are designed for integrated systems. For information, see [Redeploy the ASDK](#).

Update package types

There are three types of update packages for integrated systems:

- **Azure Stack Hub software updates.** Microsoft is responsible for the end-to-end servicing lifecycle for the Microsoft software update packages. These packages can include the latest Windows Server security updates, non-security updates, and Azure Stack Hub feature updates. You download these update packages directly from Microsoft.

Each update package has a corresponding type: **Full** or **Express**.

Full update packages update the physical host operating systems in the scale unit and require a larger maintenance window.

Express update packages are scoped and don't update the underlying physical host operating systems.

- **Azure Stack Hub hotfixes.** Microsoft provides hotfixes for Azure Stack Hub that address a specific issue that's often preventive or time-sensitive. Each hotfix is released with a corresponding Microsoft Knowledge Base article that details the issue, cause, and resolution. You download and install hotfixes just like the regular full update packages for Azure Stack Hub. Hotfixes are cumulative and can install in minutes.
- **OEM hardware-vendor-provided updates.** Azure Stack Hub hardware partners are responsible for the end-to-end servicing lifecycle (including guidance) for the hardware-related firmware and driver update packages. In addition, Azure Stack Hub hardware partners own and maintain guidance for all software and hardware on the hardware lifecycle host. The OEM hardware vendor hosts these update packages on their own download site.

When to update

The three types of updates are released with the following cadence:

- **Azure Stack Hub software updates.** Microsoft typically releases software update packages each month.
- **Azure Stack Hub hotfixes.** Hotfixes are time-sensitive releases that can be released at any time.
- **OEM hardware vendor-provided updates.** OEM hardware vendors release their updates on an as-needed basis.

To continue to receive support, you must keep your Azure Stack Hub environment on a supported Azure Stack

Hub software version. For more information, see [Azure Stack Hub Servicing Policy](#).

Where to get notice of an update

Notice of updates varies on a couple of factors, such as your connection to the internet and the type of update.

- **Microsoft software updates and hotfixes**

An update alert for Microsoft software updates and hotfixes will appear in the **Update** blade for Azure Stack Hub instances that are connected to the internet. If the **Update** blade isn't displayed, restart the infrastructure management controller VM.

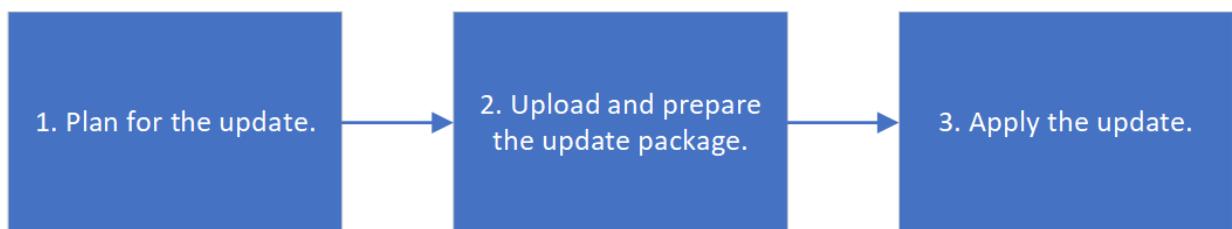
If your instance isn't connected and you would like to be notified about each hotfix release, subscribe to the [RSS](#) or [ATOM](#) feed.

- **OEM hardware vendor-provided updates**

OEM updates will depend on your manufacturer. You'll need to establish a communication channel with your OEM so that you can be aware of updates from your OEM that need to be applied. For more information about the OEMs and the OEM update process, see [Apply Azure Stack Hub original equipment manufacturer \(OEM\) updates](#).

Update processes

Once you know you have an update, apply it by using the following steps.



1. Plan for the update.

Prepare your Azure Stack Hub to make the update process go as smoothly as possible so that there's minimal impact on your users. Notify your users of any possible service outage and then follow the steps to prepare your instance for the update. Be sure to follow **ALL** steps in [Azure Stack Hub pre-update checklist](#) to ensure that you've completed the required presteps for applying an update. Also make sure to schedule an appropriate maintenance window for the update type being applied.

2. Upload and prepare the update package.

For internet-connected Azure Stack Hub environments, Azure Stack Hub software updates and hotfixes are automatically imported into the system and prepared for update.

For internet-disconnected Azure Stack Hub environments and environments with weak or intermittent internet connectivity, update packages are imported into Azure Stack Hub storage via the Azure Stack Hub administrator portal. For more steps to upload and prepare the update package, see [Upload and prepare an Azure Stack Hub update package](#).

All OEM update packages are manually imported into your environment, regardless of your Azure Stack Hub system's internet connectivity. For more steps to import and prepare the update package, see [Upload and prepare an Azure Stack Hub update package](#).

3. Apply the update.

Apply the update using the **Update** blade in Azure Stack Hub. During the update, monitor and troubleshoot the update progress. For more information, see [Apply an Azure Stack Hub update](#).

The update resource provider

Azure Stack Hub includes an update resource provider that handles the application of Microsoft software updates. This provider checks that updates are applied across all physical hosts, Service Fabric apps and runtimes, and all infrastructure virtual machines and their associated services.

As updates install, you can view high-level status as the update process targets the various subsystems in Azure Stack Hub (for example, physical hosts and infrastructure virtual machines).

Next steps

- To begin the update process, follow the steps in see [Azure Stack Hub update activity checklist](#).
- To learn what versions of Azure Stack Hub are in support, see [Azure Stack Hub Servicing Policy](#).
- To learn more about the current and recent updates, see the [Azure Stack Hub release notes](#).

Azure Stack Hub servicing policy

4 minutes to read • [Edit Online](#)

This article describes the servicing policy for Azure Stack Hub integrated systems and what you must do to keep your system in a supported state.

Download update packages for integrated systems

Microsoft releases both full monthly update packages as well as hotfix packages to address specific issues.

Monthly update packages are hosted in a secure Azure endpoint. You can download them manually using the [Azure Stack Hub Updates downloader tool](#). If your scale unit is connected, the update appears automatically in the administrator portal as **Update available**. Full, monthly update packages are well documented at each release. For more information about each release, you can click any release from the [Update package release cadence](#) section of this article.

Hotfix update packages are hosted in the same secure Azure endpoint. You can download them using the embedded links in each of the respective hotfix KB articles; for example, [Azure Stack Hub Hotfix 1.1809.12.114](#). Similar to the full, monthly update packages, Azure Stack Hub operators can download the .xml, .bin, and .exe files and import them using the procedure in [Apply updates in Azure Stack Hub](#). Azure Stack Hub operators with connected scale units will see the hotfixes automatically appear in the administrator portal with the message **Update available**.

If your scale unit isn't connected and you want to be notified about each hotfix release, subscribe to the [RSS](#) or [ATOM](#) feed noted in each release.

Update package types

There are two types of update packages for integrated systems:

- **Microsoft software updates.** Microsoft is responsible for the end-to-end servicing lifecycle for the Microsoft software update packages. These packages can include the latest Windows Server security updates, non-security updates, and Azure Stack Hub feature updates. You can download these update packages directly from Microsoft.
- **OEM hardware vendor-provided updates.** Azure Stack Hub hardware partners are responsible for the end-to-end servicing lifecycle (including guidance) for the hardware-related firmware and driver update packages. In addition, Azure Stack Hub hardware partners own and maintain guidance for all software and hardware on the hardware lifecycle host. The OEM hardware vendor hosts these update packages on their own download site.

Update package release cadence

Microsoft expects to release software update packages on a monthly cadence. However, it's possible to have multiple or no update releases in a month. OEM hardware vendors release their updates on an as-needed basis.

Find documentation on how to plan for and manage updates, and how to determine your current version in [Manage updates overview](#).

For information about a specific update, including how to download it, see the release notes for that update:

- [Azure Stack Hub 1910 update](#)
- [Azure Stack Hub 1908 update](#)
- [Azure Stack Hub 1907 update](#)
- [Azure Stack Hub 1906 update](#)

Hotfixes

Occasionally, Microsoft provides hotfixes for Azure Stack Hub that address a specific issue that's often preventative or time-sensitive. Each hotfix is released with a corresponding Microsoft Knowledge Base article that details the issue, cause, and resolution.

Hotfixes are downloaded and installed just like the regular full update packages for Azure Stack Hub. However, unlike a full update, hotfixes can install in minutes. We recommend Azure Stack Hub operators set maintenance windows when installing hotfixes. Hotfixes update the version of your Azure Stack Hub cloud so you can easily determine if the hotfix has been applied. A separate hotfix is provided for each version of Azure Stack Hub that's still in support. Each fix for a specific iteration is cumulative and includes the previous updates for that same version. You can read more about the applicability of a specific hotfix in the corresponding Knowledge Base article. See the release notes links in the previous section.

For information about currently available hotfixes, see the release notes for that update:

- [Azure Stack Hub 1910 hotfix](#)
- [Azure Stack Hub 1908 hotfix](#)
- [Azure Stack Hub 1907 hotfix](#)
- [Azure Stack Hub 1906 hotfix](#)

Keep your system under support

For your Azure Stack Hub instance to remain in a supported state, the instance must run the most recently released update version or run either of the two preceding update versions.

Hotfixes aren't considered major update versions. If your Azure Stack Hub instance is behind by *more than two updates*, it's considered out of compliance. You must update to at least the minimum supported version to receive support.

For example, if the most recently available update version is 1904, and the previous two update packages were versions 1903 and 1902, both 1902 and 1903 remain in support. However, 1901 is out of support. The policy holds true when there's no release for a month or two. For example, if the current release is 1807 and there was no 1806 release, the previous two update packages of 1805 and 1804 remain in support.

Microsoft software update packages are non-cumulative and require the previous update package or hotfix as a prerequisite. If you decide to defer one or more updates, consider the overall runtime if you want to get to the latest version.

Get support

Azure Stack Hub follows the same support process as Azure. Enterprise customers can follow the process described in [How to create an Azure support request](#). If you're a customer of a Cloud Solution Provider (CSP), contact your CSP for support. For more information, see the [Azure Support FAQs](#).

For help troubleshooting update issues, see [Best practices for troubleshooting Azure Stack Hub patch and update issues](#).

Next steps

- Manage updates in Azure Stack Hub
- Best practices for troubleshooting Azure Stack Hub patch and update issues

Azure Stack Hub update activity checklist

3 minutes to read • [Edit Online](#)

Review this checklist in order to prepare for an Azure Stack Hub update. This article contains a checklist of update-related activities for Azure Stack Hub operators.

Prepare for Azure Stack Hub update

ACTIVITY	DETAILS
Review known issues	List of known issues .
Review security updates	List of security updates .
Apply latest OEM package	Contact your OEM to ensure your system meets the minimum OEM package requirements for the Azure Stack Hub version your system is being updated to. Ensure your OEM package is compatible with the Azure Stack Hub version you are updating to. If your OEM package is not compatible with the Azure Stack Hub version you are updating to, you will need to perform an OEM package update before running an Azure Stack Hub update. For instructions, see "Apply Azure Stack Hub original equipment manufacturer (OEM) updates."
Optional: Configure automatic log collection	It is recommended that you configure automatic log collection on your Azure Stack Hub environment to streamline the process of collecting system logs in the event that you need to open a support ticket. To configure automatic log collection, see the instructions in Configure automatic Azure Stack Hub diagnostic log collection .
Apply latest hotfixes	Apply the latest hotfixes that apply to the currently installed release. For a list of the latest hotfixes, see the release notes Hotfixes section.
Run capacity planner tool	Make sure to use the latest version of the Azure Stack Hub Capacity Planner tool to perform your workload planning and sizing. The latest version contains bug fixes and provides new features that are released with each Azure Stack Hub update.
Run Test-AzureStack	Run <code>Test-AzureStack -Group UpdateReadiness</code> to identify operational issues. The cmdlet is accessible through the Privileged Endpoint Session (PEP). For more information, see Validate Azure Stack Hub system state .
Resolve issues	Resolve any operational issues identified by <code>Test-AzureStack</code> .

ACTIVITY	DETAILS
Update available	In connected scenarios only, Azure Stack Hub deployments periodically check a secured endpoint and automatically notify you if an update is available for your cloud. Disconnected customers can download and import new packages using the process described here .
Schedule a maintenance window and notify your users	You should notify users of any maintenance operations, and schedule normal maintenance windows during non-business hours if possible. Maintenance operations can affect existing tenant workloads and cause new tenants operations (for example, creating, reconfiguring, or deleting VMs) to fail - whether the operation is initiated from the portal or programmatically from the Azure Resource Manager API. Other operations such as backup may also be unavailable until the update is complete. For Azure Stack Hub express and full updates, you can check the release notes for a forecast of how long the update is expected to take for the version you are applying.

During Azure Stack Hub update

ACTIVITY	DETAILS
Manage the update	Manage updates in Azure Stack Hub using the operator portal .
Monitor the update	If the operator portal is unavailable, monitor updates in Azure Stack Hub using the privileged endpoint .
Resume updates	After remediating a failed update, resume updates in Azure Stack Hub using the privileged endpoint .

IMPORTANT

Do not run **Test-AzureStack** during an update, as this will cause the update to stall.

After Azure Stack Hub update

ACTIVITY	DETAILS
Apply latest hotfixes	Apply the latest hotfixes applicable to the updated version.
Retrieve encryption keys	Retrieve the data at rest encryption keys and securely store them outside of your Azure Stack Hub deployment. Follow the instructions on how to retrieve the keys .
Re-enable multi-tenancy	In case of a multi-tenanted Azure Stack Hub, make sure you configure all guest directory tenants after a successful update.

Next steps

- [Review list of known issues](#)
- [Review list of security updates](#)

Prepare an Azure Stack Hub update package

5 minutes to read • [Edit Online](#)

This article provides an overview of preparing Azure Stack Hub update packages so that they can be used to update your Azure Stack Hub environment. This process consists of:

- [Downloading the update package](#)
- [Importing the update package into your Azure Stack Hub environment via the Azure Stack Hub Administrator Portal](#)

On systems that can connect to the automatic update endpoints, Azure Stack Hub software updates and hotfixes are automatically downloaded and prepared. On systems without connectivity and for any update from the OEM, the update package must be prepared as explained in this topic.

The following table shows when update packages require manual preparation and when they are prepared automatically.

UPDATE TYPE	CONNECTIVITY	ACTION REQUIRED
Azure Stack Hub Software Updates	Connected	Update is automatically downloaded and prepared when the update is applied.
Azure Stack Hub Hotfixes	Connected	Update is automatically downloaded and prepared when the update is applied.
OEM Package Updates	Connected	The update package must be prepared. Follow the steps in this article.
Azure Stack Hub Software Updates	Disconnected or Weak Connection	The update package must be prepared. Follow the steps in this article.
Azure Stack Hub Hotfixes	Disconnected or Weak Connection	The update package must be prepared. Follow the steps in this article.
OEM Package Updates	Disconnected or Weak Connection	The update package must be prepared. Follow the steps in this article.

Download the update package

The update package for Azure Stack Hub updates and hotfixes is available through the update blade for connected systems. You will need to download the package and move the package to a location that is accessible to your Azure Stack Hub instance if you are updating an OEM package, or if you are supporting a disconnected system. You may also need to download and then upload the package to an accessible location if your are running a system with an intermittent connection.

Review the package contents. An update package typically consists of the following files:

- **A self-extracting <PackageName>.zip file.** This file contains the payload for the update.
- **A Metadata.xml file.** This file contains essential information about the update, for example, the publisher, name, prerequisite, size, and support path URL.

Automatic download and preparation for update packages

Azure Stack Hub software updates and hotfixes are prepared automatically for systems with connectivity to the

Azure Stack Hub automatic update endpoints: https://*.azureedge.net and

<https://aka.ms/azurestackautomaticupdate>. For more information about setting up connectivity to the **Azure Stack Hub automatic update endpoints**, see the **Patch and Update** endpoints outlined in [Azure Stack Hub Firewall Integration](#)

Where to download Azure Stack Hub update packages

Azure Stack Hub updates for [full and express updates](#) are hosted at a secure Azure endpoint. Azure Stack Hub operators with connected instances will see the [Azure Stack Hub updates automatically appear in the Administration portal](#). For internet disconnected systems or systems with weak internet connectivity, update packages can be downloaded using the [Azure Stack Hub Updates downloader tool](#). Azure Stack Hub software update packages may contain updates to Azure Stack Hub services as well as updates to the operating system of your Azure Stack Hub's scale units.

NOTE

The update package itself and its contents (such as binaries, PowerShell scripts, and so on) are signed with Microsoft-owned certificates. Tampering with the package will make the signature invalid.

Where to download Azure Stack Hub hotfix packages

Package for [Azure Stack Hub hotfixes](#) are hosted in the same secure Azure endpoint as for Azure Stack Hub updates. Azure Stack Hub operators with connected instances will see the [Azure Stack Hub updates automatically appear in the Administration portal](#). You can download them using the embedded links in each of the respective hotfix KB articles, such as [Azure Stack Hub hotfix 1.1906.11.52](#). You can find hotfixes in the release notes corresponding to your Azure Stack Hub version.

Where to download OEM update packages

Your OEM vendor will also release updates, such as driver and firmware updates. While these updates are delivered as separate [OEM package updates](#) by your hardware vendor, they are still imported, installed, and managed the same way as update packages from Microsoft. You can find a list of vendor contact links at [Apply Azure Stack Hub original equipment manufacturer \(OEM\) updates](#).

Import and install updates

The following procedure shows how to import and install update packages in the administration portal.

IMPORTANT

Notify users of any maintenance operations, and that you schedule normal maintenance windows during non-business hours as much as possible. Maintenance operations may affect both user workloads and portal operations.

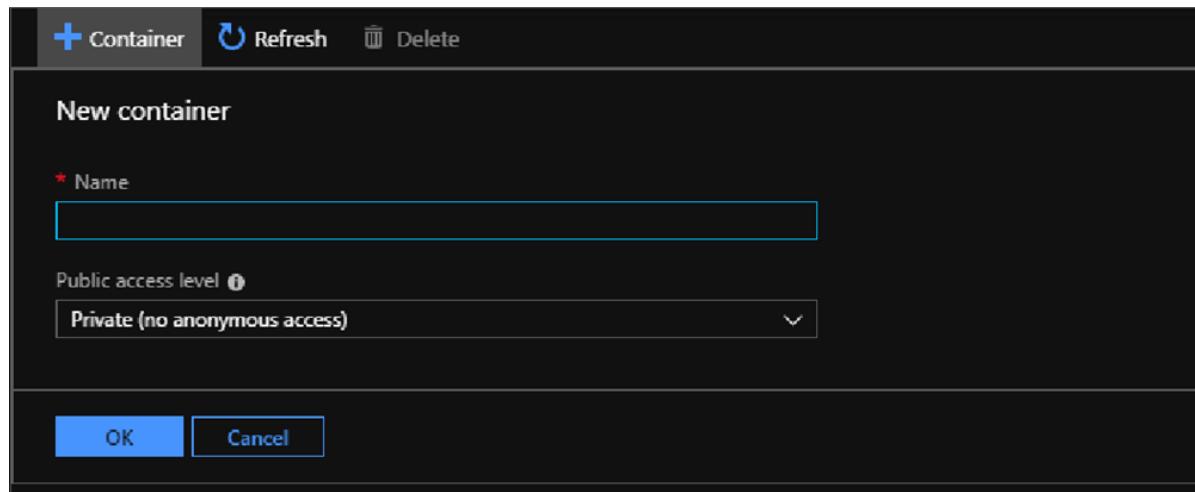
1. In the administration portal, select **All services**. Then, under the **DATA + STORAGE** category, select **Storage accounts**. (Or, in the filter box, start typing **storage accounts**, and select it.)

The screenshot shows the Microsoft Azure Stack - Administration interface. On the left, there's a navigation sidebar with options like Dashboard, All services, Favorites, Resource groups, Virtual machines, Load balancers, Storage accounts, Virtual networks, Offers, Marketplace management, Recent, and Monitor. The main area is titled 'All services' and shows a grid of service icons. A red box highlights the 'Storage accounts' icon in the 'STORAGE (1)' section. Other visible sections include ADMINISTRATION (9), ADMINISTRATIVE RESOURCES (9), GENERAL (9), COMPUTE (6), NETWORKING (10), and STORAGE (1).

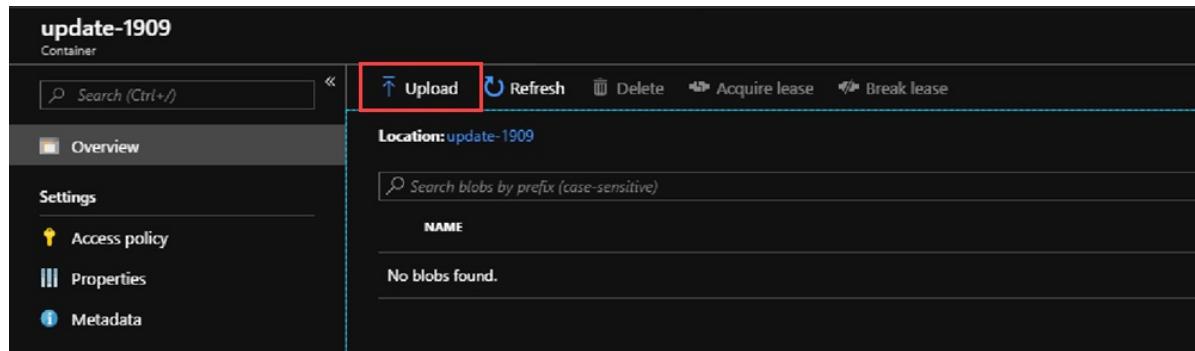
2. In the filter box, type **update**, and select the **updateadminaccount** storage account.
3. In the storage account details, under **Services**, select **Blobs**.

The screenshot shows the 'Storage accounts' details page for the 'updateadminaccount' storage account. The left sidebar lists services: Blob service, Table service, Queue service, Monitoring, Metrics, Monitoring (classic), Metrics (classic), Diagnostic logs (classic), and Usage (classic). The main pane shows the storage account details, including its location (local), replication (Locally-redundant storage (LRS)), and account kind (Storage (general purpose v1)). The 'Services' section is highlighted with a red box. It contains three items: 'Blobs' (REST-based object storage for unstructured data), 'Tables' (Tabular data storage), and 'Queues' (Effectively scale apps according to traffic). The 'Blobs' item is also highlighted with a red box.

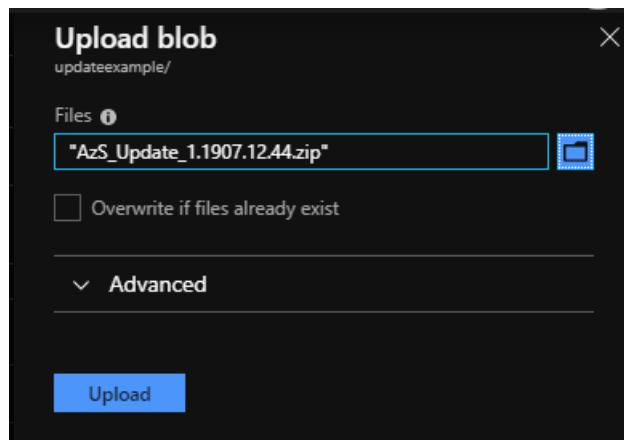
4. Under **Blob service**, select **+ Container** to create a container. Enter a name (for example *update-1811*) and then select **OK**.



5. After the container is created, click the container name, and then click **Upload** to upload the package files to the container.



6. Under **Upload blob**, click the folder icon, browse to the update package's .zip file and then click **Open** in the file explorer window.
7. Under **Upload blob**, click **Upload**.



8. Repeat steps 6 and 7 for the Metadata.xml file and any additional .zip files in the update package. Do not import the Supplemental Notice.txt file if included.
9. When done, you can review the notifications (bell icon in the top-right corner of the portal). The notifications should indicate that the upload has completed.
10. Navigate back to the Update blade on the dashboard. The blade should indicate that an update is available. This indicates that the update has been prepared successfully. Click the blade to review the newly added update package.
11. To install the update, select the package that's marked as **Ready** and either right-click the package and select **Update now**, or click the **Update now** action near the top.

12. When you click the installing update package, you can view the status in the **Update run details** area.

From here, you can also click **Download summary** to download the log files. Logs from update runs are available for 6 months after the attempt ended.

13. When the update completes, the Update blade shows the updated Azure Stack Hub version.

You can manually delete updates from the storage account after they have been installed on Azure Stack Hub.

Azure Stack Hub periodically checks for older update packages and removes them from storage. It may take Azure Stack Hub two weeks to remove the old packages.

Next steps

[Apply the update](#)

Install Azure Stack Hub Updates

2 minutes to read • [Edit Online](#)

You can install update packages using the **Update** blade in the Azure Stack Hub. This article walks you through the steps to update, monitor, and troubleshoot the update process. Use the Update blade to view update info, install updates, monitor update progress, review update history, and view the current Azure Stack Hub and OEM package version.

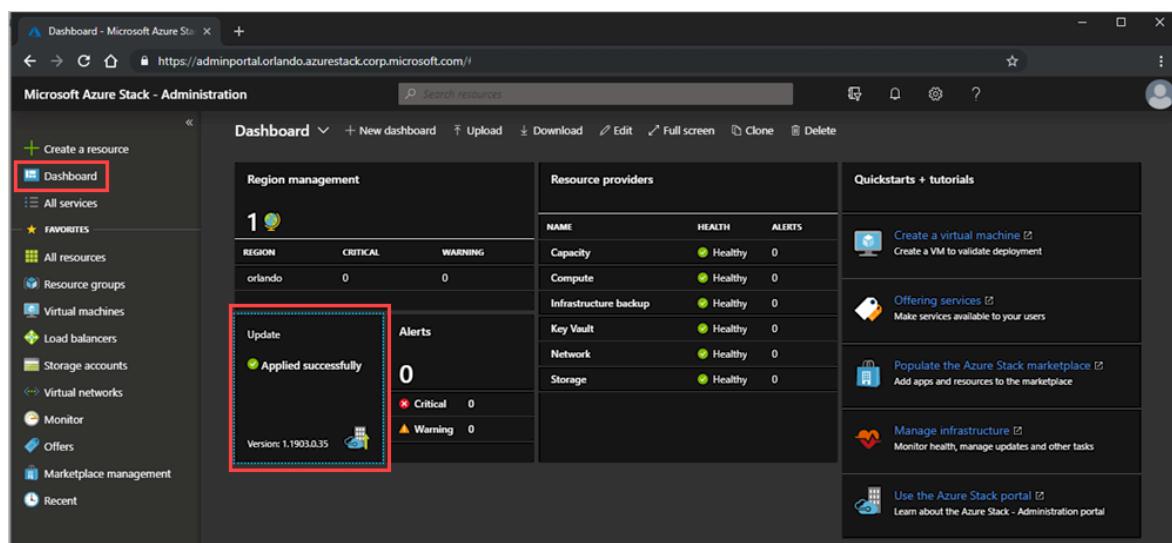
You can manage updates from the administrator portal and use the **Updates** section of the dashboard to:

- View important info, such as the current version.
- Install updates and monitor progress.
- Review update history for previously installed updates.
- View the cloud's current OEM package version.

Determine the current version

You can view the current version of Azure Stack Hub in the **Updates** blade. To open:

1. Open the Azure Stack Hub administrator portal.
2. Select **Dashboard**. In the **Updates** blade, the current version is listed.



For example, in this image the version is 1.1903.0.35.

Install updates and monitor progress

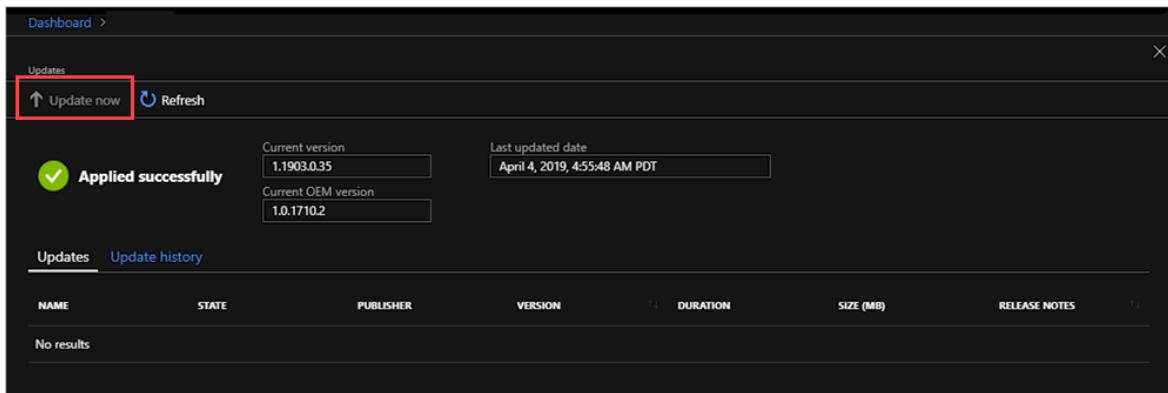
IMPORTANT

Before applying updates in Azure Stack Hub, ensure you have completed **ALL** steps in the [Pre-Update Checklist](#) and have scheduled an appropriate maintenance window for the update type that you are applying.

1. Open the Azure Stack Hub administrator portal.
2. Select **Dashboard**. Select **Update**.
3. Select the available update that you wish to install. If you don't have an update marked as **Available**, you

need to [Prepare the Update Package](#)

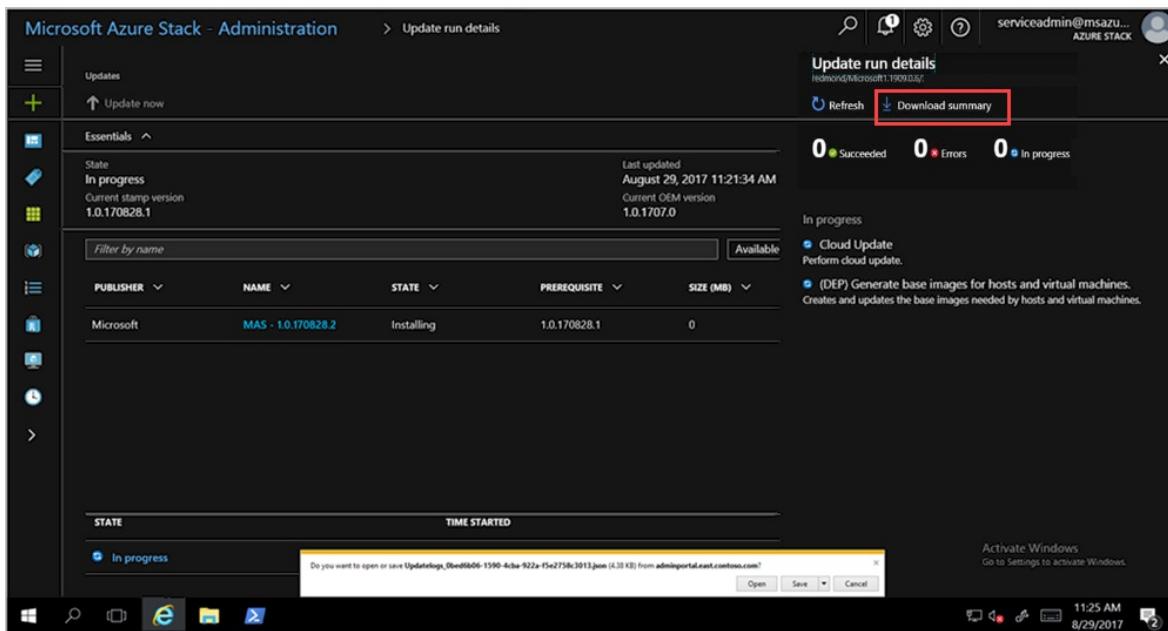
4. Select **Update now**.



5. You can view high-level status as the update process iterates through various subsystems in Azure Stack Hub. Example subsystems include physical hosts, Service Fabric, infrastructure virtual machines, and services that provide both the admin and user portals. Throughout the update process, the update resource provider reports additional details about the update, such as the number of steps that have succeeded, and the number in progress.

6. Select the **Download summary** from the Update run details blade to download full logs.

If you run into an issue while monitoring the update, you can use the [privileged endpoint](#) to monitor the progress of an Azure Stack Hub update run. You can also use the privileged endpoint to resume a failed update run from the last successful step if the Azure Stack Hub portal becomes unavailable. For instructions, see [Monitor updates in Azure Stack Hub using PowerShell](#).



7. Once completed, the update resource provider provides a **Succeeded** confirmation to show that the update process has finished and how long it took. From there, you can view info on all updates, available updates, or installed updates using the filter.

The screenshot shows the 'Updates' blade in the Azure Stack Hub administrator portal. At the top, there's a message 'Update available' with a current version of '1.1905.0.40'. Below this, there's a table for 'Infrastructure' updates, showing one entry: 'AzS Hotfix - 1.1905.3.48' in a 'Ready to install' state. A link to 'View' the release notes is provided. At the bottom, there's a section for 'Resource provider' updates, which currently shows 'No update'.

If the update fails, the **Update** blade reports **Needs attention**. Use the **Download full logs** option to get a high-level status of where the update failed. The Azure Stack Hub log collection helps with diagnostics and troubleshooting.

Review update history

1. Open the administrator portal.
2. Select **Dashboard**. Select **Update**.
3. Select **Update history**.

The screenshot shows the 'Update history' blade in the Azure Stack Hub administrator portal. It displays a summary at the top for an 'Install in progress' update, showing the current version '1.1811.0.108' and last updated date 'January 10, 2019, 6:15:39 PM PST'. Below this is a table of previous updates, all of which are listed as 'Installed'. The table includes columns for Name, State, Publisher, Version, Time Started, Time Completed, Duration, and Size (MB). The updates are: Microsoft1.1809.12.114, Microsoft1.1809.3.96, Microsoft1.1809.5.100, Microsoft1.1809.6.102, and Microsoft1.1811.0.108. The time completed for the last update is January 10, 2019, 6:15:39 PM PST, and its duration is 27:21 minutes.

Next steps

- [Manage updates in Azure Stack Hub overview](#)
- [Azure Stack Hub servicing policy](#)

Apply Azure Stack Hub original equipment manufacturer (OEM) updates

3 minutes to read • [Edit Online](#)

You can apply original equipment manufacturer (OEM) updates to your Azure Stack Hub hardware components to receive driver and firmware improvements as well as security patches while minimizing the impact on your users. In this article, you can learn about OEM updates, OEM contact information, and how to apply an OEM update.

Overview of OEM updates

In addition to Microsoft Azure Stack Hub updates, many OEMs also release regular updates for your Azure Stack Hub hardware, such as driver and firmware updates. These are referred to as **OEM Package Updates**. To understand whether your OEM releases OEM Package Updates, check your [OEM's Azure Stack Hub documentation](#).

These OEM package updates are uploaded into the **updateadminaccount** storage account and applied via the Azure Stack Hub Administrator portal. For more information, see [Applying OEM Updates](#).

Ask your original equipment manufacturer (OEM) about their specific notification process to ensure OEM package update notifications reach your organization.

Some hardware vendors may require a *hardware vendor VM* that handles the internal firmware update process. For more information, see [Configure hardware vendor VM](#)

OEM contact information

This section contains OEM contact information and links to OEM Azure Stack Hub reference material.

HARDWARE PARTNER	REGION	URL
Cisco	All	Cisco Integrated System for Microsoft Azure Stack Hub Operations Guide UCS C-Series Rack-Mount UCS-Managed Server Software
Dell EMC	All	Cloud for Microsoft Azure Stack Hub 14G (account and login required) Cloud for Microsoft Azure Stack Hub 13G (account and login required)
Fujitsu	JAPAN	Fujitsu managed service support desk (account and login required)
	EMEA & US	Fujitsu support IT products and systems
HPE	All	HPE ProLiant for Microsoft Azure Stack Hub

HARDWARE PARTNER	REGION	URL
Lenovo	All	ThinkAgile SXM Best Recipes
Wortmann		OEM/firmware package terra Azure Stack Hub documentation (including FRU)

Apply OEM updates

Apply the OEM packages with the following steps:

IMPORTANT

Before applying updates in Azure Stack Hub, ensure you have completed **ALL** steps in the [Pre-Update Checklist](#) and have scheduled an appropriate maintenance window for the update type that you are applying.

1. You will need to contact your OEM to:
 - Determine the current version of your OEM package.
 - Find the best method to download your OEM package.
2. Before applying an OEM package update, you should always apply the latest Azure Stack Hub hotfix available on your system's current Azure Stack Hub version. For more information about hotfixes see [Azure Stack Hub hotfixes](#).
3. Prepare your OEM package with the steps outlined in [Download update packages for integrated systems](#).
4. Apply the updates with the steps outlined in [Apply updates in Azure Stack Hub](#).

Configure hardware vendor VM

Some hardware vendors may require a VM to help with the OEM update process. Your hardware vendor will be responsible for creating these VMs and documenting if you require `ProxyVM` or `HardwareManager` for **-VMTYPE** when running the **Set-OEMExternalVM** cmdlet as well as which credential should be used for **-Credential**. Once the VMs are created, configure them with the **Set-OEMExternalVM** from the privileged endpoint.

For more information about the privileged endpoint on Azure Stack Hub, see [Using the privileged endpoint in Azure Stack Hub](#).

1. Access the privileged endpoint.

```
$cred = Get-Credential
$session = New-PSSession -ComputerName <IP Address of ERCS>
-ConfigurationName PrivilegedEndpoint -Credential $cred
```

2. Configure the hardware vendor VM using the **Set-OEMExternalVM** cmdlet. The cmdlet validates the IP address and credentials for **-VMTYPE** `ProxyVM`. For **-VMTYPE** `HardwareManager` the cmdlet won't validate the input. The **-Credential** parameter provided to **Set-OEMExternalVM** is one that will be clearly documented by the hardware vendor documentation. It is NOT the CloudAdmin credential used with the privileged endpoint, or any other existing Azure Stack Hub credential.

```
$VmCred = Get-Credential  
Invoke-Command -Session $session  
{  
Set-OEMExternalVM -VMTYPE <Either "ProxyVM" or "HardwareManager">  
-IPAddress <IP Address of hardware vendor VM> -Credential $using:VmCred  
}
```

Next steps

[Azure Stack Hub updates](#)

Monitor updates in Azure Stack Hub using Powershell

2 minutes to read • [Edit Online](#)

You can use the Azure Stack Hub administrative endpoints to monitor and manage your updates. They're accessible with PowerShell. For instructions on getting set up with PowerShell on Azure Stack Hub, see [Install PowerShell for Azure Stack Hub](#).

You can use the following PowerShell cmdlet to manage your updates:

CMDLET	DESCRIPTION
Get-AzsUpdate	Get the list of available updates.
Get-AzsUpdateLocation	Get the list of update locations.
Get-AzsUpdateRun	Get the list of update runs.
Install-AzsUpdate	Apply a specific update at an update location.
Resume-AzsUpdateRun	Resumes a previously started update run that failed.

Get a list of update runs

To get the list of update runs command:

```
Get-AzsUpdateRun -UpdateName Microsoft1.0.180302.1
```

Resume a failed update operation

If the update fails, you can resume the update run where it left off by running the following command:

```
Get-AzsUpdateRun -Name 5173e9f4-3040-494f-b7a7-738a6331d55c -UpdateName Microsoft1.0.180305.1 | Resume-AzsUpdateRun
```

Next steps

- [Managing updates in Azure Stack Hub](#)

Monitor updates in Azure Stack Hub using the privileged endpoint

4 minutes to read • [Edit Online](#)

You can use the [privileged endpoint](#) to monitor the progress of an Azure Stack Hub update run. You can also use the privileged endpoint to resume a failed update run from the last successful step should the Azure Stack Hub portal become unavailable. Using the Azure Stack Hub portal is the recommended method to manage updates in Azure Stack Hub.

The following new PowerShell cmdlets for update management are included in the 1710 update for Azure Stack Hub integrated systems.

CMDLET	DESCRIPTION
<code>Get-AzureStackUpdateStatus</code>	Returns the status of the currently running, completed, or failed update. Provides the high-level status of the update operation and an XML document that describes both the current step and the corresponding state.
<code>Resume-AzureStackUpdate</code>	Resumes a failed update at the point where it failed. In certain scenarios, you may have to complete mitigation steps before you resume the update.

Verify the cmdlets are available

Because the cmdlets are new in the 1710 update package for Azure Stack Hub, the 1710 update process needs to get to a certain point before the monitoring capability is available. Typically, the cmdlets are available if the status in the administrator portal indicates that the 1710 update is at the **Restart Storage Hosts** step. Specifically, the cmdlet update occurs during **Step: Running step 2.6 - Update PrivilegedEndpoint whitelist**.

You can also determine whether the cmdlets are available programmatically by querying the command list from the privileged endpoint. To do this query, run the following commands from the hardware lifecycle host or from a Privileged Access Workstation. Also, make sure the privileged endpoint is a trusted host. For more information, see step 1 of [Access the privileged endpoint](#).

1. Create a PowerShell session on any of the ERCS virtual machines (VMs) in your Azure Stack Hub environment (`Prefix-ERCS01`, `Prefix-ERCS02`, or `Prefix-ERCS03`). Replace `Prefix` with the VM prefix string that's specific to your environment.

```
$cred = Get-Credential  
  
$pepSession = New-PSSession -ComputerName <Prefix>-ercs01 -Credential $cred -ConfigurationName  
PrivilegedEndpoint
```

When prompted for credentials, use the `<Azure Stack Hub domain>\clouadmin` account, or an account that's a member of the CloudAdmins group. For the CloudAdmin account, enter the same password that was provided during installation for the AzureStackAdmin domain administrator account.

2. Get the full list of commands that are available in the privileged endpoint.

```
$commands = Invoke-Command -Session $pepSession -ScriptBlock { Get-Command }
```

3. Determine if the privileged endpoint was updated.

```
$updateManagementModuleName = "Microsoft.Azurestack.UpdateManagement"
if (($commands | ? Source -eq $updateManagementModuleName)) {
    Write-Host "Privileged endpoint was updated to support update monitoring tools."
} else {
    Write-Host "Privileged endpoint has not been updated yet. Please try again later."
}
```

4. List the commands specific to the Microsoft.AzureStack.UpdateManagement module.

```
$commands | ? Source -eq $updateManagementModuleName
```

For example:

```
$commands | ? Source -eq $updateManagementModuleName
```

CommandType	Name	Version	Source
PSComputerName			
Function	Get-AzureStackUpdateStatus	0.0	
Microsoft.Azurestack.UpdateManagement		Contoso-ercs01	
Function	Resume-AzureStackUpdate	0.0	
Microsoft.Azurestack.UpdateManagement		Contoso-ercs01	

Use the update management cmdlets

NOTE

Run the following commands from the hardware lifecycle host or from a Privileged Access Workstation. Also, make sure the privileged endpoint is a trusted host. For more information, see step 1 of [Access the privileged endpoint](#).

Connect to the privileged endpoint and assign session variable

Run the following commands to create a PowerShell session on any of the ERCS VMs in your Azure Stack Hub environment (*Prefix*-ERCS01, *Prefix*-ERCS02, or *Prefix*-ERCS03), and to assign a session variable.

```
$cred = Get-Credential

$pepSession = New-PSSession -ComputerName <Prefix>-ercs01 -Credential $cred -ConfigurationName
PrivilegedEndpoint
```

When prompted for credentials, use the *<Azure Stack Hub domain>\cloudadmin* account, or an account that's a member of the CloudAdmins group. For the CloudAdmin account, enter the same password that was provided during installation for the AzureStackAdmin domain administrator account.

Get high-level status of the current update run

To get a high-level status of the current update run, run the following commands:

```
$statusString = Invoke-Command -Session $pepSession -ScriptBlock { Get-AzureStackUpdateStatus -StatusOnly }

$statusString.Value
```

Possible values include:

- Running
- Completed
- Failed
- Canceled

You can run these commands repeatedly to see the most up-to-date status. You don't have to re-establish a connection to check again.

Get the full update run status with details

You can get the full update run summary as an XML string. You can write the string to a file for examination, or convert it to an XML document and use PowerShell to parse it. The following command parses the XML to get a hierarchical list of the currently running steps:

```
[xml]$updateStatus = Invoke-Command -Session $pepSession -ScriptBlock { Get-AzureStackUpdateStatus }

$updateStatus.SelectNodes("//Step[@Status='InProgress'])")
```

In the following example, the top-level step (Cloud Update) has a child plan to update and restart the storage hosts. It shows that the Restart Storage Hosts plan is updating the Blob Storage service on one of the hosts.

```
[xml]$updateStatus = Invoke-Command -Session $pepSession -ScriptBlock { Get-AzureStackUpdateStatus }

$updateStatus.SelectNodes("//Step[@Status='InProgress'])")

FullStepIndex : 2
Index         : 2
Name          : Cloud Update
Description   : Perform cloud update.
StartTimeUtc : 2017-10-13T12:50:39.9020351Z
Status        : InProgress
Task          : Task

FullStepIndex : 2.9
Index         : 9
Name          : Restart Storage Hosts
Description   : Restart Storage Hosts.
EceErrorAction: Stop
StartTimeUtc : 2017-10-13T15:44:06.7431447Z
Status        : InProgress
Task          : Task

FullStepIndex : 2.9.2
Index         : 2
Name          : PreUpdate ACS Blob Service
Description   : Check function level, update deployment artifacts, configure Blob service settings
StartTimeUtc : 2017-10-13T15:44:26.0708525Z
Status        : InProgress
Task          : Task
```

Resume a failed update operation

If the update fails, you can resume the update run where it left off.

```
Invoke-Command -Session $pepSession -ScriptBlock { Resume-AzureStackUpdate }
```

Troubleshoot

The privileged endpoint is available on all ERCS VMs in the Azure Stack Hub environment. Because the connection isn't made to a highly available endpoint, you may experience occasional interruptions, warning, or error messages. These messages may indicate that the session was disconnected or that there was an error communicating with the ECE Service. This behavior is expected. You can retry the operation in a few minutes or create a new privileged endpoint session on one of the other ERCS VMs.

Next steps

- [Managing updates in Azure Stack Hub](#)

Best practices for troubleshooting Azure Stack Hub patch and update issues

2 minutes to read • [Edit Online](#)

This article provides an overview of best practices for troubleshooting Azure Stack Hub patch and update issues as well as remediations to common patch and update issues.

The Azure Stack Hub patch and update process is designed to allow operators to apply update packages in a consistent, streamlined way. While uncommon, issues can occur during patch and update process. The following steps are recommended should you encounter an issue during the patch and update process:

0. **Prerequisites:** Be sure that you have followed the [Update Activity Checklist](#) and have [Configured Automatic Log Collection](#).
1. Follow the remediation steps in the failure alert created when your update failed.
2. Review the [Common Azure Stack Hub patch and update issues](#) and take the recommended actions if your issue is listed.
3. If you have been unable to resolve your issue with the above steps, create an [Azure Stack Hub support ticket](#). Be sure you have [logs collected](#) for the timespan that the issue occurred.

Common Azure Stack Hub patch and update issues

Applies to: Azure Stack Hub integrated systems

PreparationFailed

Applicable: This issue applies to all supported releases.

Cause: When attempting to install the Azure Stack Hub update, the status for the update might fail and change state to `PreparationFailed`. For internet-connected systems this is usually indicative of the update package being unable to download properly due to a weak internet connection.

Remediation: You can work around this issue by clicking **Install now** again. If the problem persists, we recommend manually uploading the update package by following the [Install updates](#) section.

Occurrence: Common

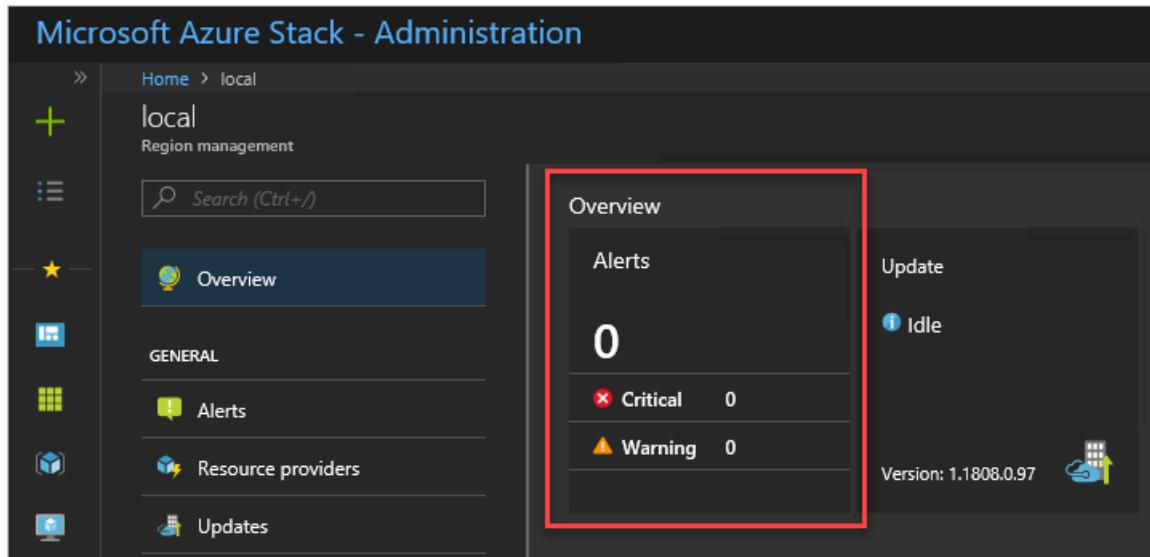
Next steps

- [Update Azure Stack Hub](#)
- [Microsoft Azure Stack Hub help and support](#)

Monitor health and alerts in Azure Stack Hub

4 minutes to read • [Edit Online](#)

Azure Stack Hub includes infrastructure monitoring capabilities that help you view health and alerts for an Azure Stack Hub region. The **Region management** tile lists all the deployed regions of Azure Stack Hub. It's pinned by default in the administrator portal for the Default Provider Subscription. The tile shows the number of active critical and warning alerts for each region. The tile is your entry point into the health and alert functionality of Azure Stack Hub.



The screenshot shows the Azure Stack Hub administration portal. On the left, there's a sidebar with icons for Home, Regions, Alerts, Resource providers, and Updates. The main area shows a 'local' region management tile. This tile has a search bar at the top. Below it is a large blue button labeled 'Overview'. To the right of this button is a box titled 'Overview' which contains 'Alerts' information. The 'Alerts' section shows a total of 0, with 0 Critical and 0 Warning alerts. A red box highlights this entire 'Overview' section. To the right of the overview box is an 'Update' section showing 'Idle' status and a version number 'Version: 1.1808.0.97'. At the bottom right of the tile is a small gear icon.

Understand health in Azure Stack Hub

The health resource provider manages health and alerts. Azure Stack Hub infrastructure components register with the health resource provider during Azure Stack Hub deployment and configuration. This registration enables the display of health and alerts for each component. Health in Azure Stack Hub is a simple concept. If alerts for a registered instance of a component exist, the health state of that component reflects the worst active alert severity: warning or critical.

Alert severity definition

Azure Stack Hub raises alerts with only two severities: **warning** and **critical**.

- **Warning**

An operator can address the warning alert in a scheduled manner. The alert typically doesn't impact user workloads.

- **Critical**

An operator should address the critical alert with urgency. These alerts indicate issues that currently impact or will soon impact Azure Stack Hub users.

View and manage component health state

You can view the health state of components in the administrator portal and through REST API and PowerShell.

To view the health state in the portal, click the region that you want to view in the **Region management** tile. You can view the health state of infrastructure roles and of resource providers.

Resource providers			Infrastructure roles			
NAME	HEALTH	ALERTS	NAME	HEALTH	ALERTS	
Compute	?	Unknown	---	Backup controller	✓ Healthy	0
Capacity	✓	Healthy	0	Compute controller	✓ Healthy	0
Key Vault	✓	Healthy	0	Directory management	✓ Healthy	0
Network	✓	Healthy	0	Edge gateway	✓ Healthy	0
Storage	✓	Healthy	0	Health controller	✓ Healthy	0

[See more](#)

You can click a resource provider or infrastructure role to view more detailed information.

WARNING

If you click an infrastructure role, and then click the role instance, there are options to **Start**, **Restart**, or **Shutdown**. Don't use these actions when you apply updates to an integrated system. Also, do **not** use these options in an Azure Stack Development Kit (ASDK) environment. These options are only designed for an integrated systems environment, where there's more than one role instance per infrastructure role. Restarting a role instance (especially AzS-Xrp01) in the ASDK causes system instability. For troubleshooting assistance, post your issue to the [Azure Stack Hub forum](#).

View alerts

The list of active alerts for each Azure Stack Hub region is available directly from the **Region management** blade. The first tile in the default configuration is the **Alerts** tile, which displays a summary of the critical and warning alerts for the region. You can pin the Alerts tile, like any other tile on this blade, to the dashboard for quick access.

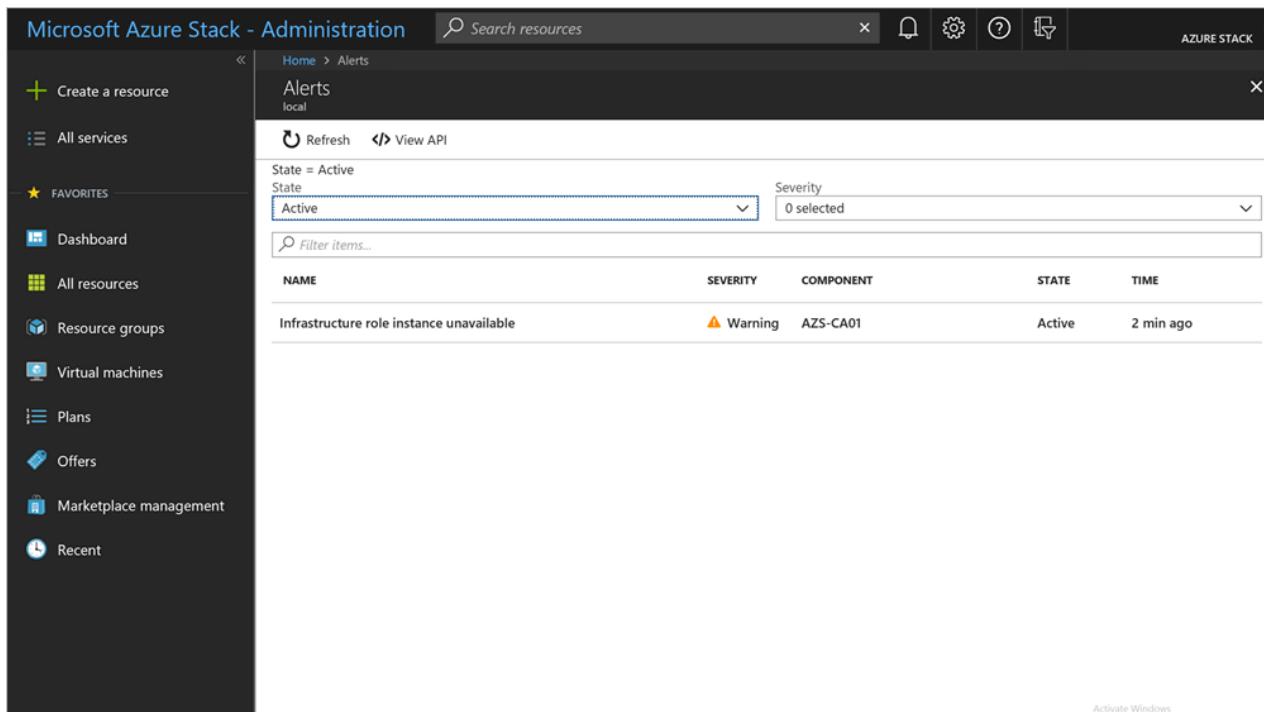
The screenshot shows the Microsoft Azure Stack - Administration blade. The top navigation bar says "local". On the left, there's a sidebar with icons for Home, Compute, Network, Storage, Key Vault, Capacity, and Resource providers. The main area is titled "Region management". It has a search bar and a "Overview" tile. Below the overview is a "GENERAL" section with links for "Alerts", "Resource providers", and "Updates". To the right is the "Alerts" tile, which is highlighted with a red box. The tile has a header "Alerts" with a globe icon, followed by a large "0", then two rows: "Critical 0" and "Warning 0".

To view a list of all active alerts for the region, select the top part of the **Alerts** tile. To view a filtered list of alerts (Critical or Warning), select either the **Critical** or **Warning** line item within the tile.

The **Alerts** blade supports the ability to filter both on status (Active or Closed) and severity (Critical or Warning). The default view displays all active alerts. All closed alerts are removed from the system after seven days.

NOTE

If an alert remains active but hasn't been updated in over a day, you can run [Test-AzureStack](#) and close the alert if no problems are reported.



The screenshot shows the Microsoft Azure Stack - Administration interface. On the left, there's a dark sidebar with various navigation options like 'Create a resource', 'All services', 'FAVORITES' (which includes 'Dashboard', 'All resources', 'Resource groups', 'Virtual machines', 'Plans', 'Offers', 'Marketplace management', and 'Recent'), and a 'Search resources' bar at the top. The main content area is titled 'Alerts local'. It has a 'Refresh' button and a 'View API' link. There are dropdown filters for 'State' set to 'Active' and 'Severity' set to '0 selected'. A search bar labeled 'Filter items...' is below the filters. A table lists one alert: 'Infrastructure role instance unavailable' with severity 'Warning', component 'AZS-CA01', state 'Active', and time '2 min ago'.

The **View API** action displays the REST API that was used to generate the list view. This action provides a quick way to become familiar with the REST API syntax that you can use to query alerts. You can use this API in automation or for integration with your existing datacenter monitoring, reporting, and ticketing solutions.

You can click a specific alert to view the alert details. The alert details show all fields that are associated with the alert and enable quick navigation to the affected component and source of the alert. For example, the following alert occurs if one of the infrastructure role instances goes offline or isn't accessible.

Home > Alerts > Infrastructure role instance unavailable

Infrastructure role instance unavailable

Alert details

X Close alert

NAME	Infrastructure role instance unavailable
SEVERITY	Warning
STATE	Active
CREATED TIME	12/13/2018 9:38:54 PM
UPDATED TIME	12/13/2018 9:40:56 PM
COMPONENT	AZS-CA01
DESCRIPTION	The infrastructure role instance AZS-CA01 is unavailable. This might impact performance and availability of Azure Stack services.
REMEDIATION	<ol style="list-style-type: none">1. Select the 'Repair' action to try to start the Infrastructure role instance, and then wait for the action to complete. Do not attempt to repair more than one alert at a time. Do not attempt the repair action if an update is in progress. Repair2. A few minutes after the Infrastructure role instance starts, the alert will automatically close. You can view the operational status of the role instance by navigating to the following AZS-CA01.3. If the alert remains active for more than a few minutes after the repair action completes, start the log file collection process using the guidance from https://aka.ms/azurestacklogfiles, and then contact support.

Repair alerts

You can select **Repair** in some alerts.

When selected, the **Repair** action performs steps specific to the alert to attempt to resolve the issue. Once selected, the status of the **Repair** action is available as a portal notification.

Home > Alerts > Infrastructure role instance unavailable

Infrastructure role instance unavailable

Alert details

X Close alert

NAME	Infrastructure role instance unavailable
SEVERITY	Warning
STATE	Active
CREATED TIME	12/13/2018 9:38:54 PM
UPDATED TIME	12/13/2018 9:40:56 PM
COMPONENT	AZS-CA01
DESCRIPTION	The infrastructure role instance AZS-CA01 is unavailable. This might impact performance and availability of Azure Stack services.
REMEDIATION	<ol style="list-style-type: none">Select the 'Repair' action to try to start the Infrastructure role instance, and then wait for the action to complete. Do not attempt to repair more than one alert at a time. Do not attempt the repair action if an update is in progress. RepairingA few minutes after the Infrastructure role instance starts, the alert will automatically close. You can view the operational status of the role instance by navigating to the following AZS-CA01.If the alert remains active for more than a few minutes after the repair action completes, start the log file collection process using the guidance from https://aka.ms/azurestacklogfiles, and then contact support.

*** Repair in progress 9:44 PM
Repair of alert "Infrastructure role instance unavailable" is in progress.

The **Repair** action will report successful completion or failure to complete the action in the same portal notification blade. If a Repair action fails for an alert, you may rerun the **Repair** action from the alert detail. If the Repair action successfully completes, **do not** rerun the **Repair** action.

Notifications X

Dismiss: Informational **Completed** All

✓ **Repair completed** 9:45 PM

Repair of alert "Infrastructure role instance unavailable" has completed successfully.

After the infrastructure role instance is back online, this alert automatically closes. Many, but not every alert automatically close when the underlying issue is resolved. Alerts that provide a Repair action button will close automatically if Azure Stack Hub resolves the issue. For all other alerts, select **Close Alert** after you do the remediation steps. If the issue persists, Azure Stack Hub generates a new alert. If you resolve the issue, the alert remains closed and requires no more steps.

Next steps

[Manage updates in Azure Stack Hub](#)

Monitor Azure Stack Hub hardware components

2 minutes to read • [Edit Online](#)

The Azure Stack Hub health and monitoring system monitors the status of the storage subsystem and raises alerts as needed. The health and monitoring system can also raise alerts for the following hardware components:

- System fans
- System temperature
- Power supply
- CPUs
- Memory
- Boot drives

NOTE

Prior to enabling this feature, you must validate with your hardware partner that they are ready. Your hardware partner will also provide the detailed steps for enabling this feature in the BMC.

SNMP listener scenario

An SNMP v3 listener is running on all three ERCS instances on TCP port 162. The baseboard management controller (BMC) must be configured to send SNMP traps to the Azure Stack Hub listener. You can get the three PEP IPs from the admin portal by opening the region properties view.

Sending traps to the listener requires authentication and must use the same credential as accessing base BMC itself.

When an SNMP trap is received on any of the three ERCS instances on TCP port 162, the OID is matched internally and an alert is raised. The Azure Stack Hub health and monitoring system only accepts OIDs defined by the hardware partner. If an OID is unknown to Azure Stack Hub, it will not match it to an alert.

Once a faulty component is replaced, an event is sent from the BMC to the SNMP listener that indicates the state change, and the alert will close automatically in Azure Stack Hub.

NOTE

Existing alerts will not close automatically when the entire node or motherboard is replaced. The same applies when the BMC loses its configuration; for example, due to a factory reset.

Next steps

[Firewall integration](#)

Manage network resources in Azure Stack Hub

3 minutes to read • [Edit Online](#)

MAC address pool

Azure Stack Hub uses a static MAC address pool to automatically generate and assign MAC address to virtual machines (VMs). This MAC address pool is automatically generated during deployment and uses the following range:

- StartMacAddress: 00-1D-D8-B7-00-00
- EndMacAddress: 00-1D-D8-F4-FF-FF

NOTE

This MAC address pool is the same across each Azure Stack Hub system and is not configurable.

Depending on how the virtual networks connect with existing corporate networks, you may expect duplicated MAC addresses of VMs.

More information can be found about MAC address pool utilization using the cmdlet [Get-AzsMacAddressPool](#) in the Azure Stack Hub administrator PowerShell module.

View public IP address consumption in Azure Stack Hub

As a cloud administrator, you can view:

- The number of public IP addresses that have been allocated to tenants.
- The number of public IP addresses that are still available for allocation.
- The percentage of public IP addresses that have been allocated in that location.

The **Public IP pools usage** tile shows the number of public IP addresses consumed across public IP address pools. For each IP address, the tile shows usage for tenant IaaS VM instances, fabric infrastructure services, and public IP address resources that were explicitly created by tenants.

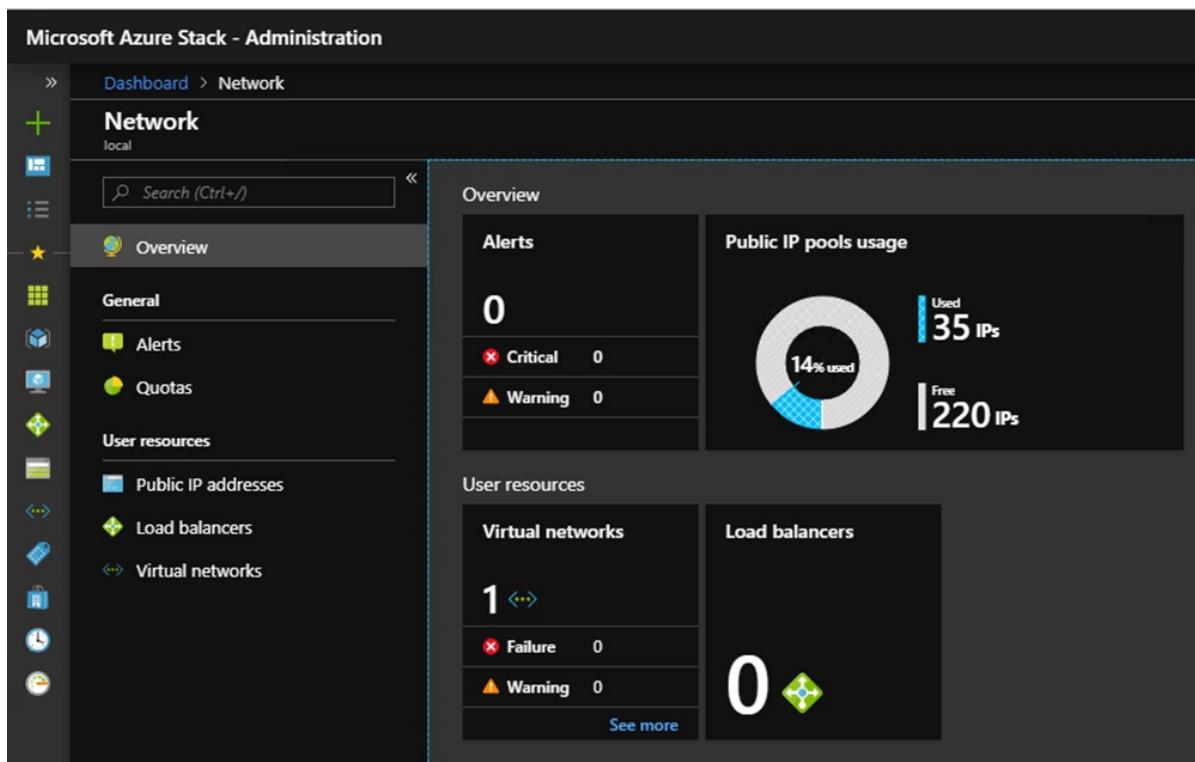
The purpose of the tile is to give Azure Stack Hub operators a sense of the number of public IP addresses used in this location. The number helps administrators determine whether they're running low on this resource.

The **Public IP addresses** menu item under **Tenant Resources** lists only those public IP addresses that have been *explicitly created by tenants*. You can find the menu item on the **Resource providers** -> **Network** pane. The number of **Used** public IP addresses on the **Public IP pools usage** tile is always different from (larger than) the number on the **Public IP Addresses** tile under **Tenant Resources**.

View the public IP address usage information

To view the total number of public IP addresses that have been consumed in the region:

1. In the Azure Stack Hub administrator portal, select **All services**. Then, under the **ADMINISTRATION** category, select **Network**.
2. The **Network** pane displays the **Public IP pools usage** tile in the **Overview** section.



The **Used** number represents the number of assigned public IP addresses from public IP address pools. The **Free** number represents the number of public IP addresses from public IP address pools that haven't been assigned and are still available. The **% Used** number represents the number of used or assigned addresses as a percentage of the total number of public IP addresses in public IP address pools in that location.

View the public IP addresses that were created by tenant subscriptions

Select **Public IP addresses** under **Tenant Resources**. Review the list of public IP addresses explicitly created by tenant subscriptions in a specific region.

ADDRESS	SUBSCRIPTION ID	ALLOCATION METHOD
192.168.102.32	[REDACTED]	Dynamic
192.168.102.34	[REDACTED]	Static
192.168.102.35	[REDACTED]	Dynamic

You might notice that some public IP addresses that have been dynamically allocated appear in the list. However, an address hasn't been associated with them yet. The address resource has been created in the Network Resource Provider, but not yet in the Network Controller.

The Network Controller doesn't assign an address to the resource until it binds to an interface, a network interface card (NIC), a load balancer, or a virtual network gateway. When the public IP address binds to an interface, the Network Controller allocates an IP address. The address appears in the **Address** field.

View the public IP address information summary table

In different cases, public IP addresses are assigned that determine whether the address appears in one list or another.

PUBLIC IP ADDRESS ASSIGNMENT CASE	APPEARS IN USAGE SUMMARY	APPEARS IN TENANT PUBLIC IP ADDRESSES LIST
Dynamic public IP address not yet assigned to an NIC or load balancer (temporary).	No	Yes
Dynamic public IP address assigned to an NIC or load balancer.	Yes	Yes
Static public IP address assigned to a tenant NIC or load balancer.	Yes	Yes
Static public IP address assigned to a fabric infrastructure service endpoint.	Yes	No
Public IP address implicitly created for IaaS VM instances and used for outbound NAT on the virtual network. These are created behind the scenes whenever a tenant creates a VM instance so that VMs can send information out to the Internet.	Yes	No

Next steps

[Manage Storage Accounts in Azure Stack Hub](#)

Change the billing owner for an Azure Stack Hub user subscription

2 minutes to read • [Edit Online](#)

Azure Stack Hub operators can use PowerShell to change the billing owner for a user subscription. One reason to change the owner, for example, is to replace a user that leaves your organization.

There are two types of *Owners* that are assigned to a subscription:

- **Billing owner:** By default, the billing owner is the user account that gets the subscription from an offer and then owns the billing relationship for that subscription. This account is also an administrator of the subscription. Only one user account can have this designation on a subscription. A billing owner is often an organization or team lead.

You can use the PowerShell cmdlet [Set-AzsUserSubscription](#) to change the billing owner.

- **Owners added through RBAC roles** - Additional users can be granted the **Owner** role using [role-based access control](#) (RBAC). Any number of additional user accounts can be added as owners to compliment the billing owner. Additional owners are also administrators of the subscription and have all privileges for the subscription, except permission to delete the billing owner.

You can use PowerShell to manage additional owners. For more information, see [this article](#).

Change the billing owner

Run the following script to change the billing owner of a user subscription. The computer that you use to run the script must connect to Azure Stack Hub and run the Azure Stack Hub PowerShell module 1.3.0 or later. For more information, see [Install Azure Stack Hub PowerShell](#).

NOTE

In a multi-tenant Azure Stack Hub, the new owner must be in the same directory as the existing owner. Before you can provide ownership of the subscription to a user that's in another directory, you must first [invite that user as a guest into your directory](#).

Replace the following values in the script before it runs:

- **\$ArmEndpoint:** The Resource Manager endpoint for your environment.
- **\$TenantId:** Your Tenant ID.
- **\$SubscriptionId:** Your subscription ID.
- **\$OwnerUpn:** An account, for example **user@example.com**, to add as the new billing owner.

```
# Set up Azure Stack Hub admin environment
Add-AzureRmEnvironment -ARMEndpoint $ArmEndpoint -Name AzureStack-admin
Add-AzureRmAccount -Environment AzureStack-admin -TenantId $TenantId

# Select admin subscription
$providerSubscriptionId = (Get-AzureRmSubscription -SubscriptionName "Default Provider Subscription").Id
Write-Output "Setting context to the Default Provider Subscription: $providerSubscriptionId"
Set-AzureRmContext -Subscription $providerSubscriptionId

# Change user subscription owner
$subscription = Get-AzsUserSubscription -SubscriptionId $SubscriptionId
$Subscription.Owner = $OwnerUpn
Set-AzsUserSubscription -InputObject $subscription
```

NOTE

If your session expires, your password has changed, or you simply wish to switch accounts, run the following cmdlet before you sign in using Add-AzureRmAccount: `Remove-AzureRmAccount -Scope Process`

Next steps

- [Manage Role-Based Access Control](#)

Start and stop Azure Stack Hub

2 minutes to read • [Edit Online](#)

Follow the procedures in this article to properly shut down and restart Azure Stack Hub services. *Stop* will physically shut down and power off the entire Azure Stack Hub environment. *Start* powers on all infrastructure roles and returns tenant resources to the power state they were in before shutdown.

Stop Azure Stack Hub

Stop or shut down Azure Stack Hub with the following steps:

1. Prepare all workloads running on your Azure Stack Hub environment's tenant resources for the upcoming shutdown.
2. Open a privileged endpoint session (PEP) from a machine with network access to the Azure Stack Hub ERCS VMs. For instructions, see [Using the privileged endpoint in Azure Stack Hub](#).
3. From the PEP, run:

```
Stop-AzureStack
```

4. Wait for all physical Azure Stack Hub nodes to power off.

NOTE

You can verify the power status of a physical node by following the instructions from the original equipment manufacturer (OEM) who supplied your Azure Stack Hub hardware.

Start Azure Stack Hub

Start Azure Stack Hub with the following steps. Follow these steps regardless of how Azure Stack Hub stopped.

1. Power on each of the physical nodes in your Azure Stack Hub environment. Verify the power on instructions for the physical nodes by following the instructions from the OEM who supplied the hardware for your Azure Stack Hub.
2. Wait until the Azure Stack Hub infrastructure services starts. Azure Stack Hub infrastructure services can require two hours to finish the start process. You can verify the start status of Azure Stack Hub with the [Get-ActionStatus cmdlet](#).
3. Ensure that all of your tenant resources have returned to the state they were in before shutdown. Workloads running on tenant resources may need to be reconfigured after startup by the workload manager.

Get the startup status for Azure Stack Hub

Get the startup for the Azure Stack Hub startup routine with the following steps:

1. Open a privileged endpoint session from a machine with network access to the Azure Stack Hub ERCS VMs.
2. From the PEP, run:

```
Get-ActionStatus Start-AzureStack
```

Troubleshoot startup and shutdown of Azure Stack Hub

Take the following steps if the infrastructure and tenant services don't successfully start two hours after you power on your Azure Stack Hub environment.

1. Open a privileged endpoint session from a machine with network access to the Azure Stack Hub ERCS VMs.
2. Run:

```
Test-AzureStack
```

3. Review the output and resolve any health errors. For more information, see [Run a validation test of Azure Stack Hub](#).

4. Run:

```
Start-AzureStack
```

5. If running **Start-AzureStack** results in a failure, contact Microsoft Support.

Next steps

Learn more about [Azure Stack Hub diagnostic tools](#)

Manage Azure Stack Hub storage accounts

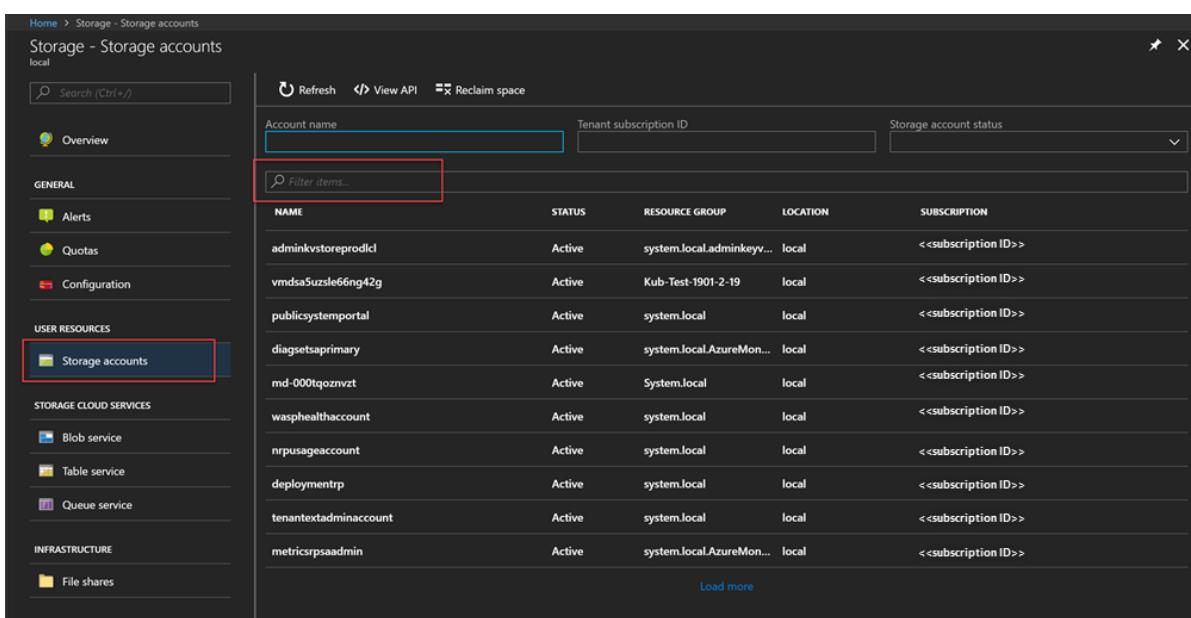
4 minutes to read • [Edit Online](#)

Learn how to manage Azure Stack Hub storage accounts. Find, recover, and reclaim storage capacity based on business needs.

Find a storage account

The list of storage accounts in the region can be viewed in Azure Stack Hub by following these steps:

1. Sign in to the [administrator portal](#).
2. Select **All services > Storage > Storage accounts**.



The screenshot shows the 'Storage - Storage accounts' page in the Azure Stack Hub administrator portal. On the left, there's a navigation sidebar with sections like Overview, General (Alerts, Quotas, Configuration), User Resources (Storage accounts, highlighted with a red box), Storage Cloud Services (Blob service, Table service, Queue service), and Infrastructure (File shares). The main pane displays a table of storage accounts with columns: NAME, STATUS, RESOURCE GROUP, LOCATION, and SUBSCRIPTION. A search bar at the top of the table is also highlighted with a red box. The table lists ten accounts, each with a status of Active and a location of local. At the bottom of the table, there's a 'Load more' link.

NAME	STATUS	RESOURCE GROUP	LOCATION	SUBSCRIPTION
adminkvstoreprod1	Active	system.local.Adminkeyv...	local	<<Subscription ID>>
vmnda5uzsle66ng42g	Active	Kub-Test-1901-2-19	local	<<Subscription ID>>
publicsys temporal	Active	system.local	local	<<Subscription ID>>
diagsetsprimary	Active	system.local.AzureMon...	local	<<Subscription ID>>
md-000tqoznvzt	Active	System.local	local	<<Subscription ID>>
washealthacount	Active	system.local	local	<<Subscription ID>>
nrpusageaccount	Active	system.local	local	<<Subscription ID>>
deploymenttrp	Active	system.local	local	<<Subscription ID>>
tenantextadminaccount	Active	system.local	local	<<Subscription ID>>
metricsrpsaadmin	Active	system.local.AzureMon...	local	<<Subscription ID>>

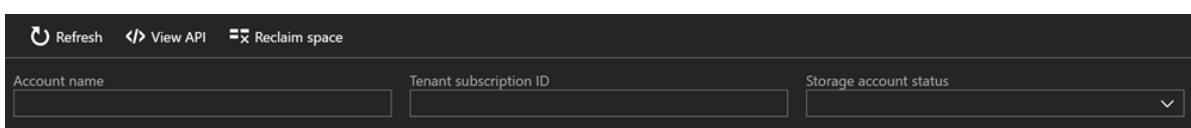
By default, the first 10 accounts are displayed. You can choose to fetch more by clicking the **Load more** link at the bottom of the list.

OR

If you're interested in a particular storage account - you can **filter and fetch the relevant accounts** only.

To filter for accounts:

1. Select **Filter** at the top of the pane.
2. On the Filter pane, it allows you to specify **account name**, **subscription ID**, or **status** to fine-tune the list of storage accounts to be displayed. Use them as appropriate.
3. As you type, the list will automatically apply the filter.



The screenshot shows the 'Storage - Storage accounts' page in the Azure Stack Hub administrator portal. At the top, there's a 'Filter' pane with three input fields: 'Account name', 'Tenant subscription ID', and 'Storage account status'. Below the filter pane, the main table of storage accounts is visible, showing the same 10 accounts as the previous screenshot.

4. To reset the filter: select **Filter**, clear out the selections and update.

The search text box (on the top of the storage accounts list pane) lets you highlight the selected text in the list of accounts. You can use this when the full name or ID isn't easily available.

You can use free text here to help find the account you're interested in.

NAME	STATUS	RESOURCE GROUP	LOCATION	SUBSCRIPTION
adminaccount@2016-09-08T17:26:43.2800...	Deleted	rg	local	7936c35c-c0d2-46b2-858c-e109
adminacc2	Active	rg	local	7936c35c-c0d2-46b2-858c-e109
adminacct3	Active	rg2	local	7936c35c-c0d2-46b2-858c-e109

Look at account details

Once you've located the accounts you're interested in viewing, you can select the particular account to view certain details. A new pane opens with the account details. These details include the kind of account, creation time, location, and so on.

STORAGE ACCOUNT NAME	adminacc2
RESOURCE GROUP NAME	rg
STATUS	Active
SUBSCRIPTION ID	{Account ID}
ACCOUNT ID	1,048,603
ACCOUNT TYPE	Standard_LRS
ACQUISITION OPERATION COUNT	0
CREATION TIME	Wed, 07 Sep 2016 23:23:44 GMT
CURRENT OPERATION	None
PRIMARY LOCATION	local
TENANT VIEW ID	/subscriptions/{Subscript ID} ...
STATUS OF PRIMARY	Available
RESOURCE ID	/subscriptions/{Subscript ID}
LOCATION	local

Recover a deleted account

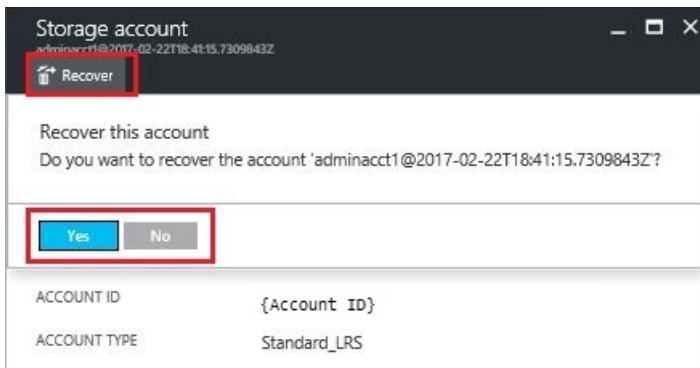
You may be in a situation where you need to recover a deleted account.

In Azure Stack Hub, there's a simple way to do that:

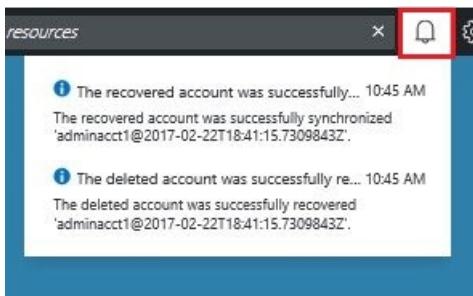
1. Browse to the storage accounts list. For more information, see [Find a storage account](#) at the top of this article.
2. Locate that particular account in the list. You may need to filter.
3. Check the *state* of the account. It should say **Deleted**.
4. Select the account, which opens the account details pane.

5. On top of this pane, locate the **Recover** button and select it.

6. Select **Yes** to confirm.



7. The recovery is now in process. Wait for an indication that it was successful. You can also select the "bell" icon at the top of the portal to view progress indications.



Once the recovered account is successfully synchronized, it can be used again.

Some Gotchas

- Your deleted account shows state as **out of retention**.

Out of retention means that the deleted account has exceeded the retention period and may not be recoverable.

- Your deleted account doesn't show in the accounts list.

Your account may not show in the account list when the deleted account has already been garbage collected. In this case, it can't be recovered. For more information, see [Reclaim capacity](#) in this article.

Set the retention period

The retention period setting allows a cloud operator to specify a time period in days (between 0 and 9999 days) during which any deleted account can potentially be recovered. The default retention period is set to 0 days.

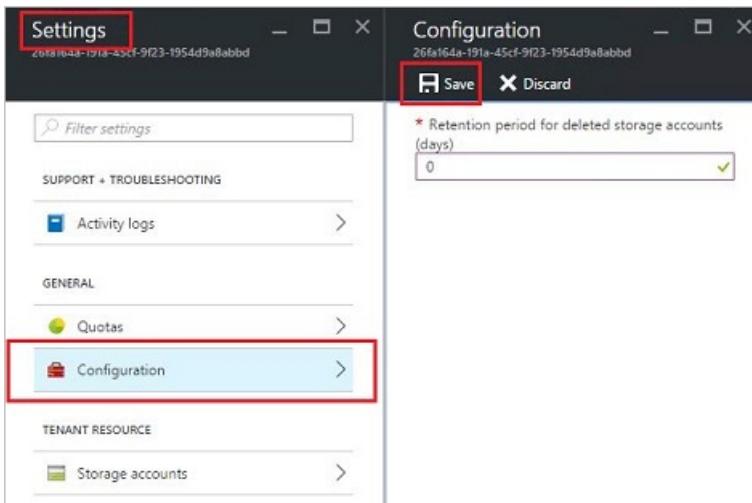
Setting the value to "0" means that any deleted account is immediately out of retention and marked for periodic garbage collection.

To change the retention period:

- Sign in to the [administrator portal](#).
- Select **All services > Region management** under **Administration**.
- Select **Resources providers > Storage > Settings**. Your path is Home > *region* - Resource providers > Storage.
- Select **Configuration** then edit the retention period value.

Set the number of days and then save it.

This value is immediately effective and is set for your entire region.



Reclaim capacity

One of the side effects of having a retention period is that a deleted account continues to consume capacity until it comes out of the retention period. As a cloud operator, you may need a way to reclaim the deleted account space even though the retention period hasn't yet expired.

You can reclaim capacity using either the portal or PowerShell.

To reclaim capacity using the portal:

1. Navigate to the storage accounts pane. See Find a storage account.
2. Select **Reclaim space** at the top of the pane.
3. Read the message and then select **OK**.

4. Wait for success notification. See the bell icon on the portal.



5. Refresh the Storage accounts page. The deleted accounts are no longer shown in the list because they've been purged.

You can also use PowerShell to explicitly override the retention period and immediately reclaim capacity.

To reclaim capacity using PowerShell:

1. Confirm that you have Azure PowerShell installed and configured. If not, use the following instructions:
 - To install the latest Azure PowerShell version and associate it with your Azure subscription, see [How to install and configure Azure PowerShell](#). For more information about Azure Resource Manager cmdlets, see [Using Azure PowerShell with Azure Resource Manager](#).
2. Run the following cmdlets:

NOTE

If you run these cmdlets, you permanently delete the account and its contents. It's not recoverable. Use this with care.

```
$farm_name = (Get-AzsStorageFarm)[0].name
Start-AzsReclaimStorageCapacity -FarmName $farm_name
```

For more information, see [Azure Stack Hub PowerShell documentation](#).

Next steps

- For information on managing permissions, see [Set access permissions using role-based access control](#).
- For information on managing storage capacity for Azure Stack Hub, see [Manage storage capacity for Azure Stack Hub](#).

Manage storage capacity for Azure Stack Hub

9 minutes to read • [Edit Online](#)

This article helps Azure Stack Hub cloud operators monitor and manage the storage capacity of their Azure Stack Hub deployment. The Azure Stack Hub storage infrastructure allocates a subset of the total storage capacity of the Azure Stack Hub deployment as *storage services*. Storage services store a tenant's data in shares on volumes that correspond to the nodes of the deployment.

As a cloud operator, you have a limited amount of storage to work with. The amount of storage is defined by the solution you implement. The solution is provided by your OEM vendor when you use a multinode solution, or it's provided by the hardware on which you install the Azure Stack Development Kit (ASDK).

Because Azure Stack Hub doesn't support expansion of storage capacity, it's important to [monitor](#) the available storage to ensure that efficient operations are maintained.

When the remaining free capacity of a share becomes limited, plan to [manage the available space](#) to prevent the shares from running out of capacity.

Your options for managing capacity include:

- Reclaiming capacity.
- Migrating a container.

When a share is 100 percent utilized, the storage service no longer functions for that share. To get assistance in restoring operations for the share, contact Microsoft support.

Understand volumes and shares, containers, and disks

Volumes and shares

The *storage service* partitions the available storage into separate, equal volumes that are allocated to hold tenant data. The number of volumes is equal to the number of nodes in the Azure Stack Hub deployment:

- In a four-node deployment, there are four volumes. Each volume has a single share. In a multinode deployment, the number of shares isn't reduced if a node is removed or if it malfunctions.
- If you use the ASDK, there's a single volume with a single share.

Because the storage service shares are for the exclusive use of storage services, you must not directly modify, add, or remove any files on the shares. Only storage services should work on the files stored in these volumes.

Shares on volumes hold tenant data. Tenant data includes page blobs, block blobs, append blobs, tables, queues, databases, and related metadata stores. Because the storage objects (blobs, and so on) are individually contained within a single share, the maximum size of each object can't exceed the size of a share. The maximum size of a new object depends on the capacity that remains in a share as unused space when the new object is created.

When a share is low on free space and actions to [reclaim](#) space aren't successful or they're unavailable, Azure Stack Hub cloud operators can migrate the blob containers from one share to another.

For information about how tenant users work with blob storage in Azure Stack Hub, see [Azure Stack Hub Storage services](#).

Containers

Tenant users create containers that are then used to store blob data. Although users decide in which container to place blobs, the storage service uses an algorithm to determine on which volume to put the container. The

algorithm typically chooses the volume with the most available space.

After a blob is placed in a container, the blob can grow to use more space. As you add new blobs and existing blobs grow, the available space in the volume that holds the container shrinks.

Containers aren't limited to a single share. When the combined blob data in a container grows to use 80 percent or more of the available space, the container enters *overflow* mode. When the container is in overflow mode, any new blobs that are created in the container are allocated to a different volume that has sufficient space. Over time, a container in overflow mode can have blobs that are distributed across multiple volumes.

When 80 percent (and then 90 percent) of the available space in a volume is used, the system raises alerts in the Azure Stack Hub administrator portal. Cloud operators should review available storage capacity and plan to rebalance the content. The storage service stops working when a disk is 100 percent used and no additional alerts are raised.

Disks

Virtual machine (VM) disks are added to containers by tenants, and they include an operating system disk. VMs can also have one or more data disks. Both types of disks are stored as page blobs. The guidance to tenants is to place each disk into a separate container to improve the performance of the VM.

- Each container that holds a disk, or page blob, from a VM is considered an attached container to the VM that owns the disk.
- A container that doesn't hold any disks from a VM is considered a free container.

The options to free up space on an attached container are limited. To learn more, see [Move VM disks](#).

TIP

Cloud operators don't directly manage disks, which are attached to VMs that tenants might add to a container. However, when you plan to manage space on storage shares, it can be useful to understand how disks relate to containers and shares.

Monitor shares

Use Azure PowerShell or the administrator portal to monitor shares so that you can understand when free space is limited. When you use the portal, you receive alerts about shares that are low on space.

Use PowerShell

As a cloud operator, you can monitor the storage capacity of a share by using the PowerShell `Get-AzsStorageShare` cmdlet. The cmdlet returns the total, allocated, and free space, in bytes, on each of the shares.

ShareName	HealthStatus	TotalCapacity	FreeCapacity	UsedCapacity
SU1FileServer.dell.selfhost.local SU1_Tenant_1	Healthy	12781621346304	12636103327744	145518018560
SU1FileServer.dell.selfhost.local SU1_Tenant_2	Healthy	12781621346304	12661266264064	120355082240
SU1FileServer.dell.selfhost.local SU1_Tenant_3	Healthy	12781621346304	12702893031424	78728314880
SU1FileServer.dell.selfhost.local SU1_Tenant_4	Healthy	12781621346304	11564357955584	1217263390720
SU1FileServer.dell.selfhost.local SU1_Tenant_5	Healthy	12781621346304	12705221623808	76399722496
SU1FileServer.dell.selfhost.local SU1_Tenant_6	Healthy	12781621346304	12702923898880	78697447424
SU1FileServer.dell.selfhost.local SU1_Tenant_7	Healthy	12781621346304	11565417799680	1216203546624

- **Total capacity:** The total space, in bytes, that's available on the share. This space is used for data and metadata that's maintained by the storage services.
- **Used capacity:** The amount of data, in bytes, that's used by all the extents from the files that store the tenant data and associated metadata.

Use the administrator portal

As a cloud operator, you can use the administrator portal to view the storage capacity of all shares.

1. Sign in to the administrator portal.
2. Select **All services > Storage > File shares** to open the file share list, where you can view the usage information.

The screenshot shows the Microsoft Azure Stack - Administration interface. In the left sidebar, under the INFRASTRUCTURE section, the 'File shares' link is highlighted with a red box. The main content area displays a table of file shares with columns for UNC PATH, STATUS, USED, AVAILABLE, and TOTAL. All listed shares are healthy and have 15.52 TB available space.

UNC PATH	STATUS	USED	AVAILABLE	TOTAL
\\$UFileServer.azurestack.local\\$UI_ObjStore_1	Healthy	533.55 GB	14.99 TB	15.52 TB
\\$UFileServer.azurestack.local\\$UI_ObjStore_2	Healthy	404.02 GB	15.12 TB	15.52 TB
\\$UFileServer.azurestack.local\\$UI_ObjStore_3	Healthy	379.25 GB	15.15 TB	15.52 TB
\\$UFileServer.azurestack.local\\$UI_ObjStore_4	Healthy	332.51 GB	15.19 TB	15.52 TB

- **Total:** The total space, in bytes, that's available on the share. This space is used for data and metadata that's maintained by the storage services.
- **Used:** The amount of data, in bytes, that's used by all the extents from the files that store the tenant data and associated metadata.

Storage space alerts

When you use the administrator portal, you receive alerts about shares that are low on space.

IMPORTANT

As a cloud operator, you should prevent shares from reaching full usage. When a share is 100 percent utilized, the storage service no longer functions for that share. To recover free space and restore operations on a share that's 100 percent utilized, you must contact Microsoft support.

- **Warning:** When a file share is over 80 percent utilized, you receive a *Warning* alert in the administrator portal:

The screenshot shows the Microsoft Azure Stack - Administration interface with the 'Alerts' page selected. The alert list shows one entry: 'A file share is over 80% utilized' with a severity of 'Warning' for the 'Storage' component. The alert was active 9 minutes ago.

NAME	SEVERITY	COMPONENT	STATE	TIME
A file share is over 80% utilized	⚠ Warning	Storage	Active	9 min ago

- **Critical:** When a file share is over 90 percent utilized, you receive a *Critical* alert in the administrator portal:

- **View details:** In the administrator portal, you can open an alert's details to view your mitigation options:

Alert details

NAME	A file share is over 90% utilized
SEVERITY	Critical
STATE	Active
CREATED TIME	10/19/2017, 3:35:42 PM
UPDATED TIME	10/19/2017, 4:47:46 PM
COMPONENT	Storage
DESCRIPTION	The file share SU1_ObjStore_4 on volume ObjStore_4 is over 90% utilized. If it reaches 100%, affected tenants will not be able to use blobs, tables, or queues.
REMEDIATION	<ol style="list-style-type: none"> 1. Navigate to Region management -> Resource providers -> Storage . 2. On the Storage blade, click the Storage accounts tile. 3. On the Storage accounts page, click Reclaim space to reclaim deleted account space. For more information, see https://aka.ms/reclaimcapacity . 4. If the issue persists, migrate storage to another file share. See https://aka.ms/migratecontainer . 5. If this didn't solve the problem, please contact Support. Before you do, start the log file collection process using the guidance from https://aka.ms/azurystacklogfiles .

Manage available space

When it's necessary to free up space on a share, use the least invasive methods first. For example, try to reclaim space before you choose to migrate a container.

Reclaim capacity

This option applies to both multinode deployments and the Azure Stack Development Kit.

You can reclaim the capacity that's used by tenant accounts that have been deleted. This capacity is automatically reclaimed when the data **retention period** is reached, or you can act to reclaim it immediately.

For more information, see the "Reclaim capacity" section of [Manage Azure Stack Hub storage accounts](#).

Migrate a container between volumes

This option applies only to Azure Stack Hub integrated systems.

Because of tenant usage patterns, some tenant shares use more space than others. This can result in some shares running low on space before other shares that are relatively unused.

You can free up space on an overused share by manually migrating some blob containers to a different share. You can migrate several smaller containers to a single share that has capacity to hold them all. Use migration to move **free** containers. Free containers are containers that don't contain a disk for a VM.

Migration consolidates all of a container's blobs on the new share.

- If a container has entered overflow mode and has placed blobs on additional volumes, the new share must have sufficient capacity to hold all of the blobs for the container you migrate. This includes the blobs that are located on additional shares.
- The PowerShell cmdlet `Get-AzsStorageContainer` identifies only the space in use on the initial volume for a container. The cmdlet doesn't identify space that's used by blobs that are put on additional volumes. Therefore, the full size of a container might not be evident. It's possible that consolidation of a container on a new share can send that new share into an overflow condition, where it places data onto additional shares. As a result, you might need to rebalance the shares.
- If you lack permissions to certain resource groups and can't use PowerShell to query the additional volumes for overflow data, work with the owner of those resource groups and containers to understand the total amount of data to migrate before you migrate it.

IMPORTANT

The migration of blobs for a container is an offline operation that requires the use of PowerShell. Until the migration is complete, all blobs for the container that you're migrating remain offline and can't be used. You should also avoid upgrading Azure Stack Hub until all ongoing migration is complete.

Migrate containers by using PowerShell

1. Confirm that you have [Azure PowerShell installed and configured](#). For more information, see [Manage Azure resources by using Azure PowerShell](#).
2. Examine the container to understand what data is on the share that you plan to migrate. To identify the best candidate containers for migration in a volume, use the `Get-AzsStorageContainer` cmdlet:

```
$farm_name = (Get-AzsStorageFarm)[0].name  
$shares = Get-AzsStorageShare -FarmName $farm_name  
$containers = Get-AzsStorageContainer -ShareName $shares[0].ShareName -FarmName $farm_name
```

Then examine `$containers`:

```
$containers
```

```

ContainerId      : 1
ContainerName    : azurestackhealthsecurityevents
ContainerState   : Active
ShareName        : \\SU1FileServer\SU1_Tenant_1
StorageAccountId : 0aba1fb8715d4b63a730f223adaf9aed
StorageAccountName : frphealthaccount
UsedBytesInPrimaryVolume : 1053159424

ContainerId      : 16
ContainerName    : azurestackhealthcentralmaeventtable
ContainerState   : Active
ShareName        : \\SU1FileServer\SU1_Tenant_1
StorageAccountId : 3f519f98c13647a08e564525fc7611f6
StorageAccountName : hrphealthaccount
UsedBytesInPrimaryVolume : 440938496

ContainerId      : 2
ContainerName    : azurestackhealthsystemevents
ContainerState   : Active
ShareName        : \\SU1FileServer\SU1_Tenant_1
StorageAccountId : 0aba1fb8715d4b63a730f223adaf9aed
StorageAccountName : frphealthaccount
UsedBytesInPrimaryVolume : 128196608

ContainerId      : 151
ContainerName    : 201702102100
ContainerState   : Active
ShareName        : \\SU1FileServer\SU1_Tenant_1
StorageAccountId : a8197ac0d70449bd8155ac1aed26d1d3
StorageAccountName : srphealthaccount
UsedBytesInPrimaryVolume : 12050432

```

3. Identify the best destination shares to hold the container you're migrating:

```
$destinationshare = ($shares | Sort-Object FreeCapacity -Descending)[0]
```

Then examine \$destinationshares:

```
$destinationshares
```

```

FreeCapacity      : 19423594041344
HealthStatus      : Unknown
ShareName         : \\su1fileserver\su1_tenant_2
TotalCapacity     : 19541899870208
UncPath           : \\su1fileserver\su1_tenant_2
UsedCapacity      : 118305828864
ResourceGroupName : system.local
FarmName          : caccfa54-37d9-40cb-af83-990ab9202cc4
Id                : /subscriptions/5bb84bc4-75f1-4f92-b603-8a0fb427bfa9/resourceGroups/sys
Location          : Local
Name              : caccfa54-37d9-40cb-af83-990ab9202cc4\\su1fileserver\su1_tenant_2
Tags              : {}
Type              : Microsoft.Storage.Admin/farms/shares

```

4. Start the migration for a container. Migration is asynchronous. If you start the migration of additional containers before the first migration is complete, use the job ID to track the status of each.

```
$job_id = Start-AzsStorageContainerMigration -StorageAccountName $containers[0].Accountname -
ContainerName $containers[0].Containername -ShareName $containers[0].Sharename -DestinationShareUncPath
$destinationshares[0].UncPath -FarmName $farm_name
```

Then examine \$jobId. In the following example, replace *d62f8f7a-8b46-4f59-a8aa-5db96db4ebb0* with the job ID you want to examine:

```
$jobId
d62f8f7a-8b46-4f59-a8aa-5db96db4ebb0
```

5. Use the job ID to check on the status of the migration job. When the container migration is complete, **MigrationStatus** is set to *Complete*.

```
Get-AzsStorageContainerMigrationStatus -JobId $job_id -FarmName $farm_name
```

```
ContainerName      : alerttemplates
DestinationShareName : \\su1fileserver\su1_tenant_2
FailureReason      :
JobId              : d62f8f7a-8b46-4f59-a8aa-5db96db4eb0
MigrationStatus    : Active
SourceShareName    : \\SU1FileServer\SU1_Tenant_1
StorageAccountName : frphealthaccount
SubEntitiesCompleted : 1
SubEntitiesFailed   : 0
RequestId          : 82b9ec24-c108-4809-b130-5234332a63ef
StatusCode         : OK
```

```
ContainerName      : alerttemplates
DestinationShareName : \\su1fileserver\su1_tenant_2
FailureReason      :
JobId              : d62f8f7a-8b46-4f59-a8aa-5db96db4eb0
MigrationStatus    : Complete
SourceShareName    : \\SU1FileServer\SU1_Tenant_1
StorageAccountName : frphealthaccount
SubEntitiesCompleted : 1
SubEntitiesFailed   : 0
RequestId          : 285dfd7e-4972-43d5-8492-df9c7e7ca5df
StatusCode         : OK
```

6. You can cancel an in-progress migration job. Canceled migration jobs are processed asynchronously. You can track cancellations by using \$jobid:

```
Stop-AzsStorageContainerMigration -JobId $job_id -FarmName $farm_name
```

RequestId	Status
ec0b1e3e-468c-4229-89c9-33f27b1b1e21	Accepted
51b7ad4c-4b88-486a-b98a-29859dd8c99f	OK

```
ContainerName      : 201702271200
DestinationShareName : \\su1fileserver\su1_tenant_3
FailureReason      :
JobId              : 7a23b432-79b6-438e-a8b7-0f242450dff
MigrationStatus    : Rollback
SourceShareName    : \\SU1FileServer\SU1_Tenant_1
StorageAccountName : srphealthaccount
SubEntitiesCompleted : 400
SubEntitiesFailed   : 0
RequestId          : 3a4f58e5-17b1-42af-811f-cc40bbcdff2
StatusCode         : OK
```

7. You can run the command from step 6 again, until the migration status is *Canceled*:

```
ContainerName      : 201702271200
DestinationShareName : \\su1fileserver\su1_tenant_3
FailureReason      :
JobId              : 7a23b432-79b6-438e-a8b7-0f242450dff
MigrationStatus    : Canceled
SourceShareName    : \\SU1FileServer\SU1_Tenant_1
StorageAccountName : srphealthaccount
SubEntitiesCompleted : 400
SubEntitiesFailed   : 0
RequestId          : fa06bc05-c372-4301-9fa8-4ba28a4d2fb6
StatusCode         : OK
```

Move VM disks

This option applies only to Azure Stack Hub integrated systems.

The most extreme method for managing space involves moving VM disks. Because moving an attached container (one that contains a VM disk) is complex, contact Microsoft support to accomplish this action.

Next steps

To learn more about offering VMs to users, see [Manage storage capacity for Azure Stack Hub](#).

Manage storage infrastructure for Azure Stack Hub

11 minutes to read • [Edit Online](#)

This article describes the health and operational status of Azure Stack Hub storage infrastructure resources. These resources include storage drives and volumes. The information in this topic helps you troubleshoot various issues, like when a drive can't be added to a pool.

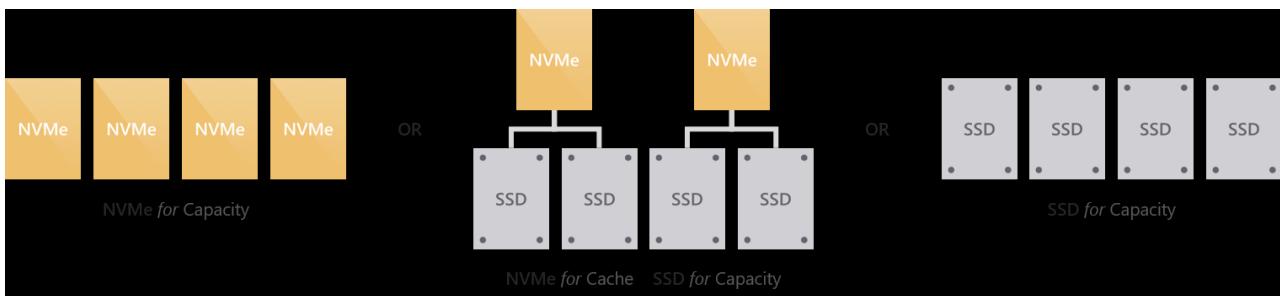
Understand drives and volumes

Drives

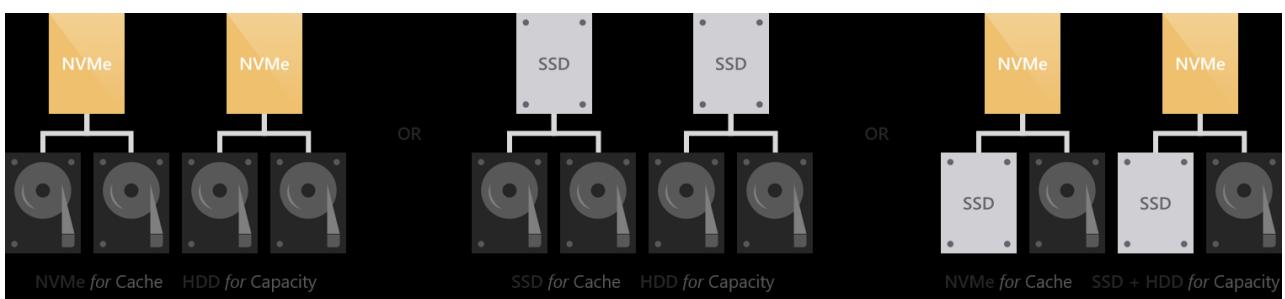
Powered by Windows Server software, Azure Stack Hub defines storage capabilities with a combination of Storage Spaces Direct (S2D) and Windows Server Failover Clustering. This combination provides a performant, scalable, and resilient storage service.

Azure Stack Hub integrated system partners offer many solution variations, including a wide range of storage flexibility. You currently can select a combination of three drive types: NVMe (non-volatile memory express), SATA/SAS SSD (solid-state drive), HDD (hard disk drive).

Storage Spaces Direct features a cache to maximize storage performance. In an Azure Stack Hub appliance with single or multiple types of drives, Storage Spaces Direct automatically use all drives of the "fastest" (NVMe > SSD > HDD) type for caching. The remaining drives are used for capacity. The drives could be grouped into either an "all-flash" or "hybrid" deployment:

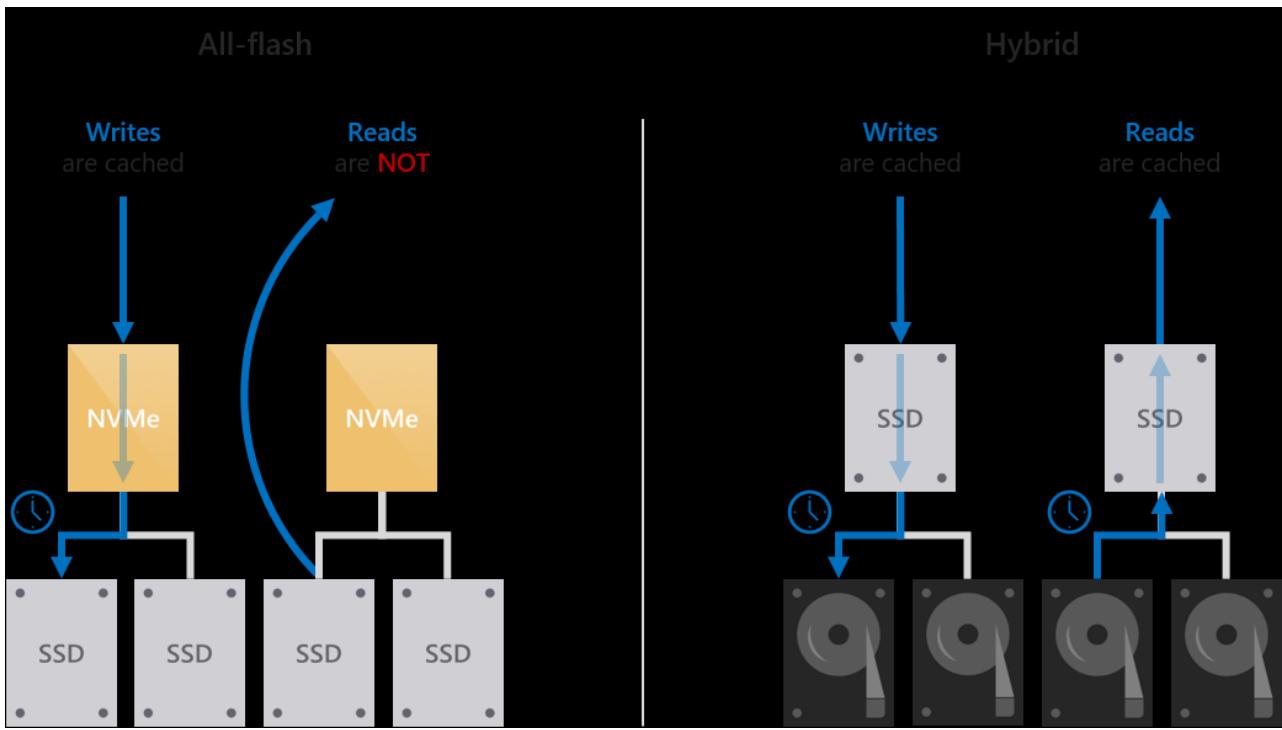


All-flash deployments aim to maximize storage performance and don't include rotational HDDs.



Hybrid deployments aim to balance performance and capacity or to maximize capacity and do include rotational HDDs.

The behavior of the cache is determined automatically based on the type(s) of drives that are being cached for. When caching for SSDs (such as NVMe caching for SSDs), only writes are cached. This reduces wear on the capacity drives, reducing the cumulative traffic to the capacity drives and extending their lifetime. In the meantime, reads aren't cached. They aren't cached because reads don't significantly affect the lifespan of flash and because SSDs universally offer low read latency. When caching for HDDs (such as SSDs caching for HDDs), both reads and writes are cached, to provide flash-like latency (often /~10x better) for both.



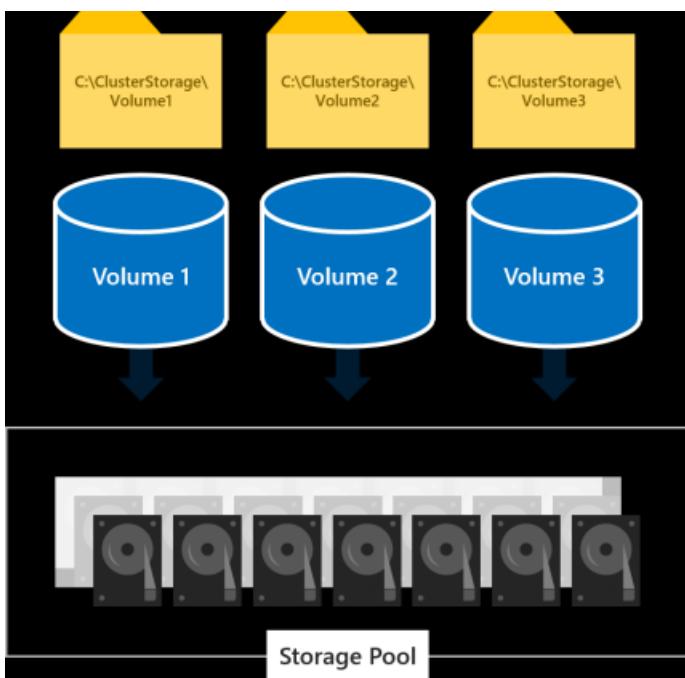
For the available configuration of storage, you can check Azure Stack Hub OEM partner (<https://azure.microsoft.com/overview/azure-stack/partners/>) for detailed specification.

NOTE

Azure Stack Hub appliance can be delivered in a hybrid deployment, with both HDD and SSD (or NVMe) drives. But the drives of faster type would be used as cache drives, and all remaining drives would be used as capacity drives as a pool. The tenant data (blobs, tables, queues, and disks) would be placed on capacity drives. Provisioning premium disks or selecting a premium storage account type doesn't guarantee the objects will be allocated on SSD or NVMe drives.

Volumes

The *storage service* partitions the available storage into separate volumes that are allocated to hold system and tenant data. Volumes combine the drives in the storage pool to provide the fault tolerance, scalability, and performance benefits of Storage Spaces Direct.



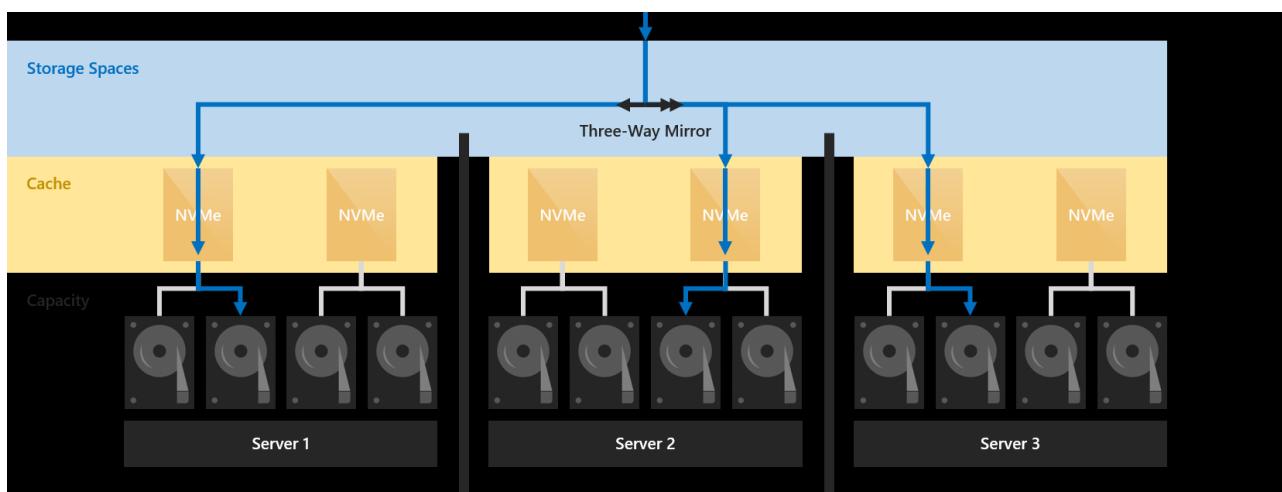
There are three types of volumes created on Azure Stack Hub storage pool:

- Infrastructure: host files used by Azure Stack Hub infrastructure VMs and core services.
- VM Temp: host the temporary disks attached to tenant VMs and that data is stored in these disks.
- Object Store: host tenant data servicing blobs, tables, queues, and VM disks.

In a multi-node deployment, you would see three infrastructure volumes, while the number of VM Temp volumes and Object Store volumes is equal to the number of the nodes in the Azure Stack Hub deployment:

- On a four-node deployment, there are four equal VM Temp volumes and four equal Object Store volumes.
- If you add a new node to the cluster, there would be a new volume for both types created.
- The number of volumes remains the same even if a node malfunctioning or is removed.
- If you use the Azure Stack Development Kit, there's a single volume with multiple shares.

Volumes in Storage Spaces Direct provide resiliency to protect against hardware problems, such as drive or server failures. They also enable continuous availability throughout server maintenance, like software updates. Azure Stack Hub deployment uses three-way mirroring to ensure data resilience. Three copies of tenant data are written to different servers, where they land in cache:



Mirroring provides fault tolerance by keeping multiple copies of all data. How that data is striped and placed is non-trivial, but it's true to say that any data stored using mirroring is written in its entirety multiple times. Each copy is written to different physical hardware (different drives in different servers) that are assumed to fail independently. Three-way mirroring can safely tolerate at least two hardware problems (drive or server) at a time. For example, if you're rebooting one server when suddenly another drive or server fails, all data remains safe and continuously accessible.

Volume states

To find out what state volumes are in, use the following PowerShell commands:

```
$scaleunit_name = (Get-AzsScaleUnit)[0].name
$subsystem_name = (Get-AzsStorageSubSystem -ScaleUnit $scaleunit_name)[0].name
Get-AzsVolume -ScaleUnit $scaleunit_name -StorageSubSystem $subsystem_name | Select-Object VolumeLabel,
HealthStatus, OperationalStatus, RepairStatus, Description, Action, TotalCapacityGB, RemainingCapacityGB
```

Here's an example of output showing a detached volume and a degraded/incomplete volume:

VOLUME LABEL	HEALTH STATUS	OPERATIONAL STATUS
ObjStore_1	Unknown	Detached
ObjStore_2	Warning	{Degraded, Incomplete}

The following sections list the health and operational states:

Volume health state: Healthy

OPERATIONAL STATE	DESCRIPTION
OK	The volume is healthy.
Suboptimal	<p>Data isn't written evenly across drives.</p> <p>Action: Contact Support to optimize drive usage in the storage pool. Before you do, start the log file collection process using the guidance from https://aka.ms/azurestacklogfiles. You may have to restore from backup after the failed connection is restored.</p>

Volume health state: Warning

When the volume is in a Warning health state, it means that one or more copies of your data are unavailable but Azure Stack Hub can still read at least one copy of your data.

OPERATIONAL STATE	DESCRIPTION
In service	<p>Azure Stack Hub is repairing the volume, like after adding or removing a drive. When the repair is complete, the volume should return to the OK health state.</p> <p>Action: Wait for Azure Stack Hub to finish repairing the volume and check the status afterward.</p>
Incomplete	<p>The resilience of the volume is reduced because one or more drives failed or are missing. However, the missing drives contain up-to-date copies of your data.</p> <p>Action: Reconnect any missing drives, replace any failed drives, and bring online any servers that are offline.</p>
Degraded	<p>The resilience of the volume is reduced because of one or more failed or missing drives as well as outdated copies of data on the drives.</p> <p>Action: Reconnect any missing drives, replace any failed drives, and bring online any servers that are offline.</p>

Volume health state: Unhealthy

When a volume is in an Unhealthy health state, some or all of the data on the volume is currently inaccessible.

OPERATIONAL STATE	DESCRIPTION

OPERATIONAL STATE	DESCRIPTION
No redundancy	<p>The volume has lost data because too many drives failed.</p> <p>Action: Contact Support. Before you do, start the log file collection process using the guidance from https://aka.ms/azurestacklogfiles.</p>

Volume health state: Unknown

The volume can also be in the Unknown health state if the virtual disk has become detached.

OPERATIONAL STATE	DESCRIPTION
Detached	<p>A storage device failure occurred which may cause the volume to be inaccessible. Some data may be lost.</p> <p>Action:</p> <ol style="list-style-type: none"> 1. Check the physical and network connectivity of all storage devices. 2. If all devices are connected correctly, contact Support. <p>Before you do, start the log file collection process using the guidance from https://aka.ms/azurestacklogfiles. You may have to restore from backup after the failed connection is restored.</p>

Drive states

Use the following PowerShell commands to monitor the state of drives:

```
$scaleunit_name = (Get-AzsScaleUnit)[0].name
$subsystem_name = (Get-AzsStorageSubSystem -ScaleUnit $scaleunit_name)[0].name
Get-AzsDrive -ScaleUnit $scaleunit_name -StorageSubSystem $subsystem_name | Select-Object StorageNode,
PhysicalLocation, HealthStatus, OperationalStatus, Description, Action, Usage, CanPool, CannotPoolReason,
SerialNumber, Model, MediaType, CapacityGB
```

The following sections describe the health states a drive can be in:

Drive health state: Healthy

OPERATIONAL STATE	DESCRIPTION
OK	The volume is healthy.
In service	The drive is doing some internal housekeeping operations. When the action is complete, the drive should return to the OK health state.

Drive health state: Healthy

A drive in the Warning state can read and write data successfully but has an issue.

OPERATIONAL STATE	DESCRIPTION

Operational State	Description
Lost communication	<p>Connectivity has been lost to the drive.</p> <p>Action: Bring all servers back online. If that doesn't fix it, reconnect the drive. If this state persists, replace the drive to ensure full resiliency.</p>
Predictive failure	<p>A failure of the drive is predicted to occur soon.</p> <p>Action: Replace the drive as soon as possible to ensure full resiliency.</p>
IO error	<p>There was a temporary error accessing the drive.</p> <p>Action: If this state persists, replace the drive to ensure full resiliency.</p>
Transient error	<p>There was a temporary error with the drive. This error usually means the drive was unresponsive, but it could also mean that the Storage Spaces Direct protective partition was inappropriately removed from the drive.</p> <p>Action: If this state persists, replace the drive to ensure full resiliency.</p>
Abnormal latency	<p>The drive is sometimes unresponsive and is showing signs of failure.</p> <p>Action: If this state persists, replace the drive to ensure full resiliency.</p>
Removing from pool	<p>Azure Stack Hub is in the process of removing the drive from its storage pool.</p> <p>Action: Wait for Azure Stack Hub to finish removing the drive, and check the status afterward. If the status remains, contact Support. Before you do, start the log file collection process using the guidance from https://aka.ms/azurestacklogfiles.</p>
Starting maintenance mode	<p>Azure Stack Hub is in the process of putting the drive in maintenance mode. This state is temporary—the drive should soon be in the In maintenance mode state.</p> <p>Action: Wait for Azure Stack Hub to finish the process and check the status afterward.</p>
In maintenance mode	<p>The drive is in maintenance mode, halting reads and writes from the drive. This state usually means Azure Stack Hub administration tasks such as PNU or FRU are operating the drive. But the admin could also place the drive in maintenance mode.</p> <p>Action: Wait for Hub Azure Stack Hub to finish the administration task, and check the status afterward. If the status remains, contact Support. Before you do, start the log file collection process using the guidance from https://aka.ms/azurestacklogfiles.</p>

OPERATIONAL STATE	DESCRIPTION
Stopping maintenance mode	Azure Stack Hub is in the process of bringing the drive back online. This state is temporary - the drive should soon be in another state, ideally Healthy. Action: Wait for Azure Stack Hub to finish the process and check the status afterward.

Drive health state: Unhealthy

A drive in the Unhealthy state can't currently be written to or accessed.

OPERATIONAL STATE	DESCRIPTION
Split	The drive has become separated from the pool. Action: Replace the drive with a new disk. If you must use this disk, remove the disk from the system, make sure there's no useful data on the disk, erase the disk, and then reseat the disk.
Not usable	The physical disk is quarantined because it's not supported by your solution vendor. Only disks that are approved for the solution and have the correct disk firmware are supported. Action: Replace the drive with a disk that has an approved manufacturer and model number for the solution.
Stale metadata	The replacement disk was previously used and may contain data from an unknown storage system. The disk is quarantined. Action: Replace the drive with a new disk. If you must use this disk, remove the disk from the system, make sure there's no useful data on the disk, erase the disk, and then reseat the disk.
Unrecognized metadata	Unrecognized metadata found on the drive, which usually means that the drive has metadata from a different pool on it. Action: Replace the drive with a new disk. If you must use this disk, remove the disk from the system, make sure there's no useful data on the disk, erase the disk, and then reseat the disk.
Failed media	The drive failed and won't be used by Storage Spaces anymore. Action: Replace the drive as soon as possible to ensure full resiliency.
Device hardware failure	There was a hardware failure on this drive. Action: Replace the drive as soon as possible to ensure full resiliency.

OPERATIONAL STATE	DESCRIPTION
Updating firmware	Azure Stack Hub is updating the firmware on the drive. This state is temporary and usually lasts less than a minute and during which time other drives in the pool handle all reads and writes. Action: Wait for Azure Stack Hub to finish the updating and check the status afterward.
Starting	The drive is getting ready for operation. This state should be temporary—once complete, the drive should transition to a different operational state. Action: Wait for Azure Stack Hub to finish the operation and check the status afterward.

Reasons a drive can't be pooled

Some drives just aren't ready to be in Azure Stack Hub storage pool. You can find out why a drive isn't eligible for pooling by looking at the `CannotPoolReason` property of a drive. The following table gives a little more detail on each of the reasons.

REASON	DESCRIPTION
Hardware not compliant	The drive isn't in the list of approved storage models specified by using the Health Service. Action: Replace the drive with a new disk.
Firmware not compliant	The firmware on the physical drive isn't in the list of approved firmware revisions by using the Health Service. Action: Replace the drive with a new disk.
In use by cluster	The drive is currently used by a Failover Cluster. Action: Replace the drive with a new disk.
Removable media	The drive is classified as a removable drive. Action: Replace the drive with a new disk.
Not healthy	The drive isn't in a healthy state and might need to be replaced. Action: Replace the drive with a new disk.
Insufficient capacity	There are partitions taking up the free space on the drive. Action: Replace the drive with a new disk. If you must use this disk, remove the disk from the system, make sure there's no useful data on the disk, erase the disk, and then reseat the disk.

Reason	Description
Verification in progress	<p>The Health Service is checking to see if the drive or firmware on the drive is approved for use.</p> <p>Action: Wait for Azure Stack Hub to finish the process, and check the status afterward.</p>
Verification failed	<p>The Health Service couldn't check to see if the drive or firmware on the drive is approved for use.</p> <p>Action: Contact Support. Before you do, start the log file collection process using the guidance from https://aka.ms/azurestacklogfiles.</p>
Offline	<p>The drive is offline.</p> <p>Action: Contact Support. Before you do, start the log file collection process using the guidance from https://aka.ms/azurestacklogfiles.</p>

Next step

[Manage storage capacity](#)

Manage physical memory capacity in Azure Stack Hub

2 minutes to read • [Edit Online](#)

To increase the total available memory capacity in Azure Stack Hub, you can add more memory. In Azure Stack Hub, your physical server is also referred to as a *scale unit node*. All scale unit nodes that are members of a single scale unit must have the same amount of memory.

NOTE

Before you continue, consult your hardware manufacturer's documentation to see if your manufacturer supports a physical memory upgrade. Your OEM hardware vendor support contract may require that the vendor perform the physical server rack placement and the device firmware update.

The following flow diagram shows the general process to add memory to each scale unit node.



Add memory to an existing node

The following steps provide a high-level overview of the process to add memory.

WARNING

Don't follow these steps without referring to your OEM-provided documentation.

WARNING

The entire scale unit must be shut down as a rolling memory upgrade isn't supported.

1. Stop Azure Stack Hub using the steps documented in the [Start and stop Azure Stack Hub](#) article.
2. Upgrade the memory on each physical computer using your hardware manufacturer's documentation.
3. Start Azure Stack Hub using the steps in the [Start and stop Azure Stack Hub](#) article.

Next steps

- To learn how to manage storage accounts in Azure Stack Hub, see [Manage storage accounts in Azure Stack Hub](#).
- To learn how to monitor and manage the storage capacity of your Azure Stack Hub deployment, see [Manage storage capacity for Azure Stack Hub](#).

Add additional scale unit nodes in Azure Stack Hub

4 minutes to read • [Edit Online](#)

Azure Stack Hub operators can increase the overall capacity of an existing scale unit by adding an additional physical computer. The physical computer is also referred to as a scale unit node. Each new scale unit node you add must be homogeneous in CPU type, memory, and disk number and size to the nodes that are already present in the scale unit.

To add a scale unit node, you act in Azure Stack Hub and run tooling from your hardware equipment manufacturer (OEM). The OEM tooling runs on the hardware lifecycle host (HLH) to make sure the new physical computer matches the same firmware level as existing nodes.

The following flow diagram shows the general process to add a scale unit node:



Whether your OEM hardware vendor enacts the physical server rack placement and updates the firmware varies based on your support contract.

The operation to add a new node can take several hours or days to complete. There is no impact to any running workloads on the system while an additional scale unit node is added.

NOTE

Don't attempt any of the following operations while an add scale unit node operation is already in progress:

- Update Azure Stack Hub
- Rotate certificates
- Stop Azure Stack Hub
- Repair scale unit node

Add scale unit nodes

The following steps are a high-level overview of how to add a node. Don't follow these steps without first referring to your OEM-provided capacity expansion documentation.

1. Place the new physical server in the rack and cable it appropriately.
2. Enable physical switch ports and adjust access control lists (ACLs) if applicable.
3. Configure the correct IP address in the baseboard management controller (BMC) and apply all BIOS settings per your OEM-provided documentation.
4. Apply the current firmware baseline to all components by using the tools that are provided by the hardware manufacturer that run on the HLH.
5. Run the add node operation in the Azure Stack Hub administrator portal.
6. Validate that the add node operation succeeds. To do so, check the [Status of the Scale Unit](#).

Add the node

You can use the administrator portal or PowerShell to add new nodes. The add node operation first adds the new

scale unit node as available compute capacity and then automatically extends the storage capacity. The capacity expands automatically because Azure Stack Hub is a hyperconverged system where *compute* and *storage* scale together.

Use the administrator portal

1. Sign in to the Azure Stack Hub administrator portal as an Azure Stack Hub operator.
2. Navigate to **+ Create a resource > Capacity > Scale Unit Node**.

The screenshot shows the Microsoft Azure Stack - Administration portal. On the left, there's a navigation sidebar with options like Dashboard, All resources, Resource groups, Virtual machines, Plans, Offers, Marketplace management, Recent, and More services. A red box highlights the 'New' button at the top of this sidebar. The main area is titled 'New' and features a search bar 'Search the Marketplace'. Below it, there are two tabs: 'Azure Marketplace' (selected) and 'Featured'. Under 'Azure Marketplace', there are several categories: Get started, Offers + Plans, Compute, Data + Storage, Networking, Custom, Security + Identity, and Capacity. A red box highlights the 'Scale Unit Node' option under the Compute category. Another red box highlights the 'Capacity' tab at the bottom of the list.

3. On the **Add node** pane, select the *Region*, and then select the *Scale unit* that you want to add the node to. Also specify the *BMC IP ADDRESS* for the scale unit node you're adding. You can only add one node at a time.

The screenshot shows the 'Add node' pane within the Microsoft Azure Stack - Administration portal. The left sidebar is identical to the previous screenshot. The main area is titled 'Add node'. It contains an information icon with the text: 'Adds new node(s) to a scale unit and configures the required settings. [Learn more](#)'. Below this, there are three input fields: a dropdown for 'Region', a dropdown for 'Scale unit', and a text input field for 'BMC IP ADDRESS' with the placeholder 'BMC IP addresses can't be empty'. A red box highlights the 'Region' dropdown.

Use PowerShell

Use the **New-AzsScaleUnitNodeObject** cmdlet to add a node.

Before using either of the following sample PowerShell scripts, replace the values *node names* and *IP addresses* with values from your Azure Stack Hub environment.

NOTE

When naming a node you must keep the name to less than 15 characters in length. You also can't use a name that contains a space or contains any of the following characters: \, /, :, *, ?, ", <, >, |, \, ~, !, @, #, \$, %, ^, &, (,), {, }, _.

Add a node:

```
## Add a single Node
$NewNode=New-AzsScaleUnitNodeObject -computername "<name_of_new_node>" -BMCIPv4Address "
<BMCIP_address_of_new_node>

Add-AzsScaleUnitNode -NodeList $NewNode -ScaleUnit "<name_of_scale_unit_cluster>"
```

Monitor add node operations

Use the administrator portal or PowerShell to get the status of the add node operation. Add node operations can take several hours to days to complete.

Use the administrator portal

To monitor the addition of a new node, review the scale unit or scale unit node objects in the administrator portal. To do so, go to **Region management** > **Scale units**. Next, select the scale unit or scale unit node you want to review.

Use PowerShell

The status for scale unit and scale unit nodes can be retrieved using PowerShell as follows:

```
#Retrieve Status for the Scale Unit
Get-AzsScaleUnit|select name,state

#Retrieve Status for each Scale Unit Node
Get-AzsScaleUnitNode |Select Name, ScaleUnitNodeStatus
```

Status for the add node operation

For a scale unit:

STATUS	DESCRIPTION
Running	All nodes are actively participating in the scale unit.
Stopped	The scale unit node is either down or unreachable.
Expanding	One or more scale unit nodes are currently being added as compute capacity.
Configuring Storage	The compute capacity has been expanded and the storage configuration is running.

STATUS	DESCRIPTION
Requires Remediation	An error has been detected that requires one or more scale unit nodes to be repaired.

For a scale unit node:

STATUS	DESCRIPTION
Running	The node is actively participating in the scale unit.
Stopped	The node is unavailable.
Adding	The node is actively being added to the scale unit.
Repairing	The node is actively being repaired.
Maintenance	The node is paused, and no active user workload is running.
Requires Remediation	An error has been detected that requires the node to be repaired.

Troubleshooting

The following are common issues seen when adding a node.

Scenario 1: The add scale unit node operation fails but one or more nodes are listed with a status of Stopped.

- Remediation: Use the repair operation to repair one or more nodes. Only a single repair operation can run at one time.

Scenario 2: One or more scale unit nodes have been added but the storage expansion failed. In this scenario, the scale unit node object reports a status of Running but the Configuring Storage task isn't started.

- Remediation: Use the privileged endpoint to review the storage health by running the following PowerShell cmdlet:

```
Get-VirtualDisk -CimSession s-cluster | Get-StorageJob
```

Scenario 3: You received an alert that indicates the storage scale-out job failed.

- Remediation: In this case, the storage configuration task has failed. This problem requires you to contact support.

Next steps

[Add public IP addresses](#)

Add public IP addresses

2 minutes to read • [Edit Online](#)

In this article, we refer to external addresses as public IP addresses. In the context of Azure Stack Hub, a public IP address is an IP address that's accessible from outside of Azure Stack Hub. Whether that external network is public internet routable or is on an intranet and uses private address space doesn't matter for the purposes of this article—the steps are the same.

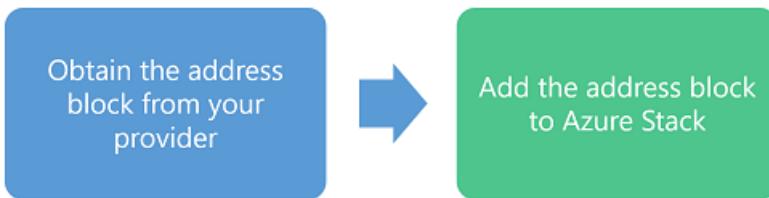
IMPORTANT

The steps in this article apply only to systems that were deployed using the partner toolkit version 1809 or later. Systems that were deployed before version 1809 require the top-of-rack (TOR) switch access control lists (ACLs) to be updated to PERMIT the new public VIP pool range. If you are running older switch configurations, work with your OEM to either add the appropriate PERMIT ACLs for the new public IP pool or reconfigure your switch using the latest partner toolkit to prevent the new public IP addresses from being blocked.

Add a public IP address pool

You can add public IP addresses to your Azure Stack Hub system at any time after the initial deployment of the Azure Stack Hub system. Check out how to [View public IP address consumption](#) to see what the current usage and public IP address availability is on your Azure Stack Hub.

At a high level, the process of adding a new public IP address block to Azure Stack Hub looks like this:



Obtain the address block from your provider

The first thing you'll need to do is to obtain the address block you want to add to Azure Stack Hub. Depending on where you obtain your address block from, consider what the lead time is and manage this against the rate at which you're consuming public IP addresses in Azure Stack Hub.

IMPORTANT

Azure Stack Hub will accept any address block that you provide if it's a valid address block and doesn't overlap with an existing address range in Azure Stack Hub. Please make sure you obtain a valid address block that's routable and non-overlapping with the external network to which Azure Stack Hub is connected. After you add the range to Azure Stack Hub, you can't remove it.

Add the IP address range to Azure Stack Hub

1. In a browser, go to your administrator portal dashboard. For this example, we'll use <https://adminportal.local.azurestack.external>.
2. Sign in to the Azure Stack Hub administrator portal as a cloud operator.

3. On the default dashboard, find the Region management list and select the region you want to manage. For this example, we use local.
4. Find the Resource providers tile and click on the network resource provider.
5. Click on the Public IP pools usage tile.
6. Click on the Add IP pool button.
7. Provide a name for the IP pool. The name you choose helps you easily identify the IP pool. It's a good practice to make the name the same as the address range, but that isn't required.
8. Enter the address block you want to add in CIDR notation. For example: 192.168.203.0/24
9. When you provide a valid CIDR range in the Address range (CIDR block) field the Start IP address, End IP address and Available IP addresses fields will automatically populate. They're read-only and automatically generated so you can't change these fields without modifying the value in the Address range field.
10. After you review the info on the blade and confirm that everything looks correct, select **Ok** to commit the change and add the address range to Azure Stack Hub.

Next steps

[Review scale unit node actions.](#)

Scale unit node actions in Azure Stack Hub

5 minutes to read • [Edit Online](#)

This article describes how to view the status of a scale unit. You can view the unit's nodes. You can run node actions like power on, power off, shut down, drain, resume, and repair. Typically, you use these node actions during field replacement of parts, or to help recover a node.

IMPORTANT

All node actions described in this article should target one node at a time.

View the node status

In the administrator portal, you can view the status of a scale unit and its associated nodes.

To view the status of a scale unit:

1. On the **Region management** tile, select the region.
2. On the left, under **Infrastructure resources**, select **Scale units**.
3. In the results, select the scale unit.
4. On the left, under **General**, select **Nodes**.

View the following information:

- The list of individual nodes.
- Operational Status (see list below).
- Power Status (running or stopped).
- Server model.
- IP address of the baseboard management controller (BMC).
- Total number of cores.
- Total amount of memory.

The screenshot shows the 'Nodes' page for a scale unit named 'redmond'. The top navigation bar includes 'Home', 'redmond - Scale units', 's-cluster - Nodes', and a user icon. Below the navigation is a search bar and a toolbar with buttons for 'Add node', 'Refresh', 'Start', 'Drain', 'Repair', 'Shutdown', and 'Stop'. A sidebar on the left has 'Overview' and 'Nodes' sections, with 'Nodes' currently selected. The main content area displays a table titled '4 items' with columns: NAME, OPERATIONAL STATE, POWER STATUS, MODEL, BMC, CORES, and MEMORY. The table lists four nodes, all of which are 'Running' and have a 'QuantaGrid D51PH-1ULH' model and '100.71.13.7' BMC IP address. Each row has a 'More' button (...).

NAME	OPERATIONAL STATE	POWER STATUS	MODEL	BMC	CORES	MEMORY
ASRR1546R06U05	Running	Running	QuantaGrid D51PH-1ULH	100.71.13.7	72	383.9 GB
ASRR1546R06U06	Running	Running	QuantaGrid D51PH-1ULH	100.71.13.8	72	383.9 GB
ASRR1546R06U07	Running	Running	QuantaGrid D51PH-1ULH	100.71.13.9	72	383.9 GB
ASRR1546R06U08	Running	Running	QuantaGrid D51PH-1ULH	100.71.13.10	72	383.9 GB

Node operational states

STATUS	DESCRIPTION
Running	The node is actively participating in the scale unit.
Stopped	The node is unavailable.

STATUS	DESCRIPTION
Adding	The node is actively being added to the scale unit.
Repairing	The node is actively being repaired.
Maintenance	The node is paused, and no active user workload is running.
Requires Remediation	An error has been detected that requires the node to be repaired.

Scale unit node actions

When you view information about a scale unit node, you can also perform node actions like:

- Start and stop (depending on current power status).
- Disable and resume (depending on operations status).
- Repair.
- Shutdown.

The operational state of the node determines which options are available.

You need to install Azure Stack Hub PowerShell modules. These cmdlets are in the **Azs.Fabric.Admin** module. To install or verify your installation of PowerShell for Azure Stack Hub, see [Install PowerShell for Azure Stack Hub](#).

Stop

The **Stop** action turns off the node. It's the same as pressing the power button. It doesn't send a shutdown signal to the operating system. For planned stop operations, always try the shutdown operation first.

This action is typically used when a node is in a hung state and no longer responds to requests.

To run the stop action, open an elevated PowerShell prompt, and run the following cmdlet:

```
Stop-AzsScaleUnitNode -Location <RegionName> -Name <NodeName>
```

In the unlikely case that the stop action doesn't work, retry the operation and if it fails a second time use the BMC web interface instead.

For more information, see [Stop-AzsScaleUnitNode](#).

Start

The **start** action turns on the node. It's the same as if you press the power button.

To run the start action, open an elevated PowerShell prompt, and run the following cmdlet:

```
Start-AzsScaleUnitNode -Location <RegionName> -Name <NodeName>
```

In the unlikely case that the start action doesn't work, retry the operation. If it fails a second time, use the BMC web interface instead.

For more information, see [Start-AzsScaleUnitNode](#).

Drain

The **drain** action moves all active workloads to the remaining nodes in that particular scale unit.

This action is typically used during field replacement of parts, like the replacement of an entire node.

IMPORTANT

Make sure you use a drain operation on a node during a planned maintenance window, where users have been notified. Under some conditions, active workloads can experience interruptions.

To run the drain action, open an elevated PowerShell prompt, and run the following cmdlet:

```
Disable-AzsScaleUnitNode -Location <RegionName> -Name <NodeName>
```

For more information, see [Disable-AzsScaleUnitNode](#).

Resume

The **resume** action resumes a disabled node and marks it active for workload placement. Earlier workloads that were running on the node don't fail back. (If you use a drain operation on a node be sure to power off. When you power the node back on it's not marked as active for workload placement. When ready, you must use the resume action to mark the node as active.)

To run the resume action, open an elevated PowerShell prompt, and run the following cmdlet:

```
Enable-AzsScaleUnitNode -Location <RegionName> -Name <NodeName>
```

For more information, see [Enable-AzsScaleUnitNode](#).

Repair

Caution

Firmware leveling is critical for the success of the operation described in this article. Missing this step can lead to system instability, a decrease in performance, security threads, or failure when Azure Stack Hub automation deploys the operating system. Always consult your hardware partner's documentation when replacing hardware to ensure the applied firmware matches the OEM Version displayed in the [Azure Stack Hub administrator portal](#).

For more information and links to partner documentation, see [Replace a hardware component](#).

HARDWARE PARTNER	REGION	URL
Cisco	All	Cisco Integrated System for Microsoft Azure Stack Hub Operations Guide Release Notes for Cisco Integrated System for Microsoft Azure Stack Hub
Dell EMC	All	Cloud for Microsoft Azure Stack Hub 14G (account and login required) Cloud for Microsoft Azure Stack Hub 13G (account and login required)

HARDWARE PARTNER	REGION	URL
Fujitsu	JAPAN	Fujitsu managed service support desk (account and login required)
	EMEA	Fujitsu support IT products and systems
		Fujitsu MySupport (account and login required)
HPE	All	HPE ProLiant for Microsoft Azure Stack Hub
Lenovo	All	ThinkAgile SXM Best Recipes

The **repair** action repairs a node. Use it only for either of the following scenarios:

- Full node replacement (with or without new data disks).
- After hardware component failure and replacement (if advised in the field replaceable unit [FRU] documentation).

IMPORTANT

See your OEM hardware vendor's FRU documentation for exact steps when you need to replace a node or individual hardware components. The FRU documentation will specify whether you need to run the repair action after replacing a hardware component.

When you run the repair action, you need to specify the BMC IP address.

To run the repair action, open an elevated PowerShell prompt, and run the following cmdlet:

```
Repair-AzsScaleUnitNode -Location <RegionName> -Name <NodeName> -BMCIPv4Address <BMCIPv4Address>
```

Shutdown

The **shutdown** action first moves all active workloads to the remaining nodes in the same scale unit. Then the action gracefully shuts down the scale unit node.

After you start a node that was shut down, you need to run the **resume** action. Earlier workloads that were running on the node don't fail back.

If the shutdown operation fails, attempt the **drain** operation followed by the shutdown operation.

To run the shutdown action, open an elevated PowerShell prompt, and run the following cmdlet:

```
Stop-AzsScaleUnitNode -Location <RegionName> -Name <NodeName> -Shutdown
```

Next steps

[Learn about the Azure Stack Hub Fabric operator module.](#)

Replace a scale unit node on an Azure Stack Hub integrated system

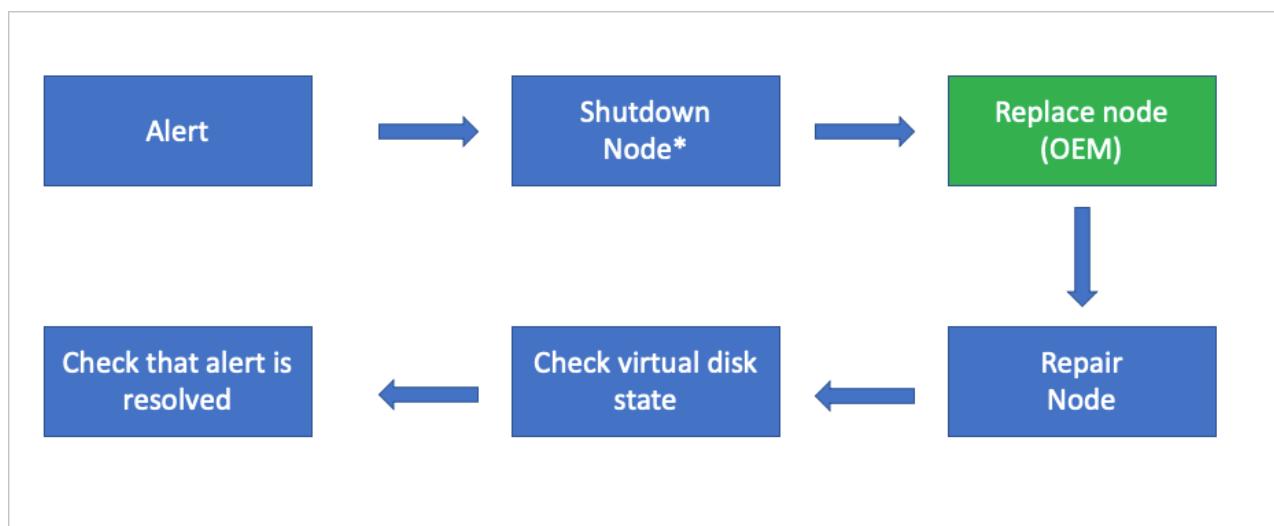
2 minutes to read • [Edit Online](#)

This article describes the general process to replace a physical computer (also referred to as a scale unit node) on an Azure Stack Hub integrated system. Actual scale unit node replacement steps will vary based on your original equipment manufacturer (OEM) hardware vendor. See your vendor's field replaceable unit (FRU) documentation for detailed steps that are specific to your system.

Caution

Firmware leveling is critical for the success of the operation described in this article. Missing this step can lead to system instability, performance decrease, security threads, or prevent Azure Stack Hub automation from deploying the operating system. Always consult your hardware partner's documentation when replacing hardware to ensure the applied firmware matches the OEM Version displayed in the [Azure Stack Hub administrator portal](#). For more information and links to partner documentation, see [Replace a hardware component](#).

The following flow diagram shows the general FRU process to replace an entire scale unit node.



*This action may not be required based on the physical condition of the hardware.

NOTE

If the shutdown operation does fail, it's recommended to use the drain operation followed by the stop operation. For more information, see [Scale unit node actions in Azure Stack Hub](#).

Review alert information

If a scale unit node is down, you'll receive the following critical alerts:

- Node not connected to network controller
- Node inaccessible for virtual machine placement
- Scale unit node is offline

Alerts	
orlando	
Filter Refresh </> View API	
Filtered by State = Active	
<input type="text"/> Filter items...	
NAME	SEVERITY
Node not connected to network controller	Critical
Node inaccessible for virtual machine placement	Critical
Scale unit node is offline	Critical

If you open the **Scale unit node is offline** alert, the alert description contains the scale unit node that's inaccessible. You may also receive additional alerts in the OEM-specific monitoring solution that's running on the hardware lifecycle host.

Scale unit node is offline	
Alert details	
Close alert	
NAME	Scale unit node is offline
SEVERITY	Critical
STATE	Active
CREATED TIME	10/6/2017 6:24:15 PM
UPDATED TIME	10/6/2017 6:28:17 PM
COMPONENT	NODE06
DESCRIPTION	<p>The node NODE06 in the scale unit is inaccessible. There is less capacity available for tenant workloads. A process has been started to move tenant workloads from this node to other nodes. If there is no available capacity, some workloads may not restart.</p>
REMEDIATION	<p>1. Click the node name link in the Description field and try to cycle the node using the Power off/Power on actions on the node blade. (A physical node restart might take up to 10 minutes.)</p> <p>2. If this didn't solve the problem, please contact Support. Before you do, start the log file collection process using the guidance from https://aka.ms/azurestacklogfiles. If hardware replacement is required, there are important pre- and post-replacement steps. See https://aka.ms/azurestackreplacenode.</p>

Scale unit node replacement process

The following steps are provided as a high-level overview of the scale unit node replacement process. See your OEM hardware vendor's FRU documentation for detailed steps that are specific to your system. Don't follow these steps without referring to your OEM-provided documentation.

1. Use the **Shutdown** action to gracefully shut down the scale unit node. This action may not be required based on the physical condition of the hardware.
2. In the unlikely case the shutdown action fails, use the **Drain** action to put the scale unit node into maintenance mode. This action may not be required based on the physical condition of the hardware.

NOTE

In any case, only one node can be disabled and powered off at the same time without breaking the S2D (Storage Spaces Direct).

3. After the scale unit node is in maintenance mode, use the [Stop](#) action. This action may not be required based on the physical condition of the hardware.

NOTE

In the unlikely case that the Power off action doesn't work, use the baseboard management controller (BMC) web interface instead.

4. Replace the physical computer. Typically, this replacement is done by your OEM hardware vendor.
5. Use the [Repair](#) action to add the new physical computer to the scale unit.
6. Use the privileged endpoint to [check the status of virtual disk repair](#). With new data drives, a full storage repair job can take multiple hours depending on system load and consumed space.
7. After the repair action has finished, validate that all active alerts have been automatically closed.

Next steps

- For information about replacing a physical disk while the system is powered on, see [Replace a disk](#).
- For information about replacing a hardware component that requires the system to be powered off, see [Replace a hardware component](#).

Replace a physical disk in Azure Stack Hub

3 minutes to read • [Edit Online](#)

This article describes the general process to replace a physical disk in Azure Stack Hub. If a physical disk fails, you should replace it as soon as possible.

NOTE

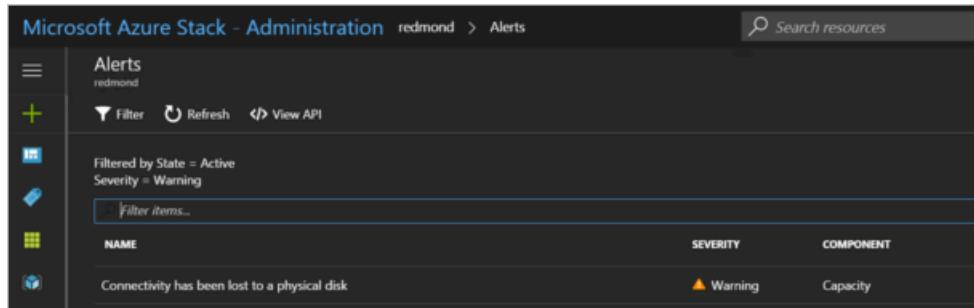
Replacing a physical data drive does **not** require the scale unit node to be put into maintenance mode (drain) upfront. Also after the physical drive has been replaced the scale unit node doesn't need to be repaired using the Azure Stack Hub administrator portal. The following article has more information when a repair is required [Replace a hardware component on an Azure Stack Hub scale unit node](#).

You can use this procedure for deployments that have hot-swappable disks.

Actual disk replacement steps will vary based on your original equipment manufacturer (OEM) hardware vendor. See your vendor's field replaceable unit (FRU) documentation for detailed steps that are specific to your system.

Review disk alert information

When a disk fails, you receive an alert that tells you that connectivity has been lost to a physical disk.



The screenshot shows the Azure Stack Administration interface under the 'Alerts' section for the 'redmond' scale unit node. A single alert is listed:

NAME	SEVERITY	COMPONENT
Connectivity has been lost to a physical disk	Warning	Capacity

If you open the alert, the alert description contains the scale unit node and the exact physical slot location for the disk that you must replace. Azure Stack Hub further helps you to identify the failed disk by using LED indicator capabilities.

Replace the physical disk

Follow your OEM hardware vendor's FRU instructions for actual disk replacement.

NOTE

Replace disks for one scale unit node at a time. Wait for the virtual disk repair jobs to complete before moving on to the next scale unit node.

To prevent the use of an unsupported disk in an integrated system, the system blocks disks that aren't supported by your vendor. If you try to use an unsupported disk, a new alert tells you a disk has been quarantined because of an unsupported model or firmware.

After you replace the disk, Azure Stack Hub automatically discovers the new disk and starts the virtual disk repair process.

Check the status of virtual disk repair using Azure Stack Hub PowerShell

After you replace the disk, you can monitor the virtual disk health status and repair job progress by using Azure Stack Hub PowerShell.

1. Check that you have Azure Stack Hub PowerShell installed. For more information, see [Install PowerShell for Azure Stack Hub](#).
2. Connect to Azure Stack Hub with PowerShell as an operator. For more information, see [Connect to Azure Stack Hub with PowerShell as an operator](#).
3. Run the following cmdlets to verify the virtual disk health and repair status:

```
$scaleunit=Get-AzsScaleUnit  
$StorageSubSystem=Get-AzsStorageSubSystem -ScaleUnit $scaleunit.Name  
Get-AzVolume -StorageSubSystem $StorageSubSystem.Name -ScaleUnit $scaleunit.name | Select-Object  
VolumeLabel, OperationalStatus, RepairStatus
```

```
PS C:\WINDOWS\system32> Get-AzVolume -StorageSubSystem $StorageSubSystem.Name -ScaleUnit $scaleunit.name |select VolumeLabel, OperationalStatus, RepairStatus  
VolumeLabel OperationalStatus RepairStatus  
-----  
VmTemp_1 OK  
ObjStore_5 OK  
ObjStore_7 OK  
VmTemp_4 OK  
Infrastructure_3 OK  
ObjStore_6 OK  
ObjStore_2 OK  
VmTemp_2 OK  
VmTemp_6 OK  
ObjStore_3 OK  
VmTemp_7 OK  
ObjStore_1 OK  
Infrastructure_2 OK  
Infrastructure_1 OK  
ObjStore_9 OK  
VmTemp_3 OK  
VmTemp_8 OK  
ObjStore_5 OK  
VmTemp_5 OK  
ObjStore_4 OK  
VmTemp_8 OK
```

4. Validate Azure Stack Hub system state. For instructions, see [Validate Azure Stack Hub system state](#).
5. Optionally, you can run the following command to verify the status of the replaced physical disk.

```
$scaleunit=Get-AzsScaleUnit  
$StorageSubSystem=Get-AzsStorageSubSystem -ScaleUnit $scaleunit.Name  
  
Get-AzDrive -StorageSubSystem $StorageSubSystem.Name -ScaleUnit $scaleunit.name | Sort-Object  
StorageNode, MediaType, PhysicalLocation | Format-Table StorageNode, Healthstatus, PhysicalLocation,  
Model, MediaType, CapacityGB, CanPool, CannotPoolReason
```

```
PS C:\WINDOWS\system32> Get-AzDrive -StorageSubSystem $StorageSubSystem.Name -ScaleUnit $scaleunit.name |ft StorageNode, Healthstatus, PhysicalLocation, Model, MediaType, CapacityGB, CanPool, CannotPoolReason  
StorageNode Healthstatus PhysicalLocation Model MediaType CapacityGB CanPool CannotPoolReason  
-----  
redmond/RedAz2-Node0 Healthy Slot 1 INTEL SSDSC2BA80 SSD 745 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 2 TOSHIBA MG04AC4M HDD 3726 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 2 INTEL SSDSC2BA80 SSD 745 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 1 INTEL SSDSC2BA80 SSD 745 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 17 TOSHIBA MG04AC4A HDD 3726 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 0 INTEL SSDSC2BA80 SSD 745 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 0 INTEL SSDSC2BA80 SSD 745 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 10 TOSHIBA MG04AC4A HDD 3726 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 3 INTEL SSDSC2BA80 SSD 745 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 14 TOSHIBA MG04AC4A HDD 3726 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 10 TOSHIBA MG04AC4A HDD 3726 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 9 TOSHIBA MG04AC4A HDD 3726 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 5 TOSHIBA MG04AC4A HDD 3726 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 1 TOSHIBA MG04AC4A HDD 3726 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 16 TOSHIBA MG04AC4A HDD 3726 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 2 INTEL SSDSC2BA80 SSD 745 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 11 TOSHIBA MG04AC4A HDD 3726 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 5 TOSHIBA MG04AC4A HDD 3726 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 4 TOSHIBA MG04AC4A HDD 3726 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 11 TOSHIBA MG04AC4A HDD 3726 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 9 TOSHIBA MG04AC4A HDD 3726 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 5 TOSHIBA MG04AC4A HDD 3726 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 1 TOSHIBA MG04AC4A HDD 3726 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 16 TOSHIBA MG04AC4A HDD 3726 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 3 INTEL SSDSC2BA80 SSD 745 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 14 TOSHIBA MG04AC4A HDD 3726 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 5 TOSHIBA MG04AC4A HDD 3726 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 3 INTEL SSDSC2BA80 SSD 745 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 2 INTEL SSDSC2BA80 SSD 745 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 15 TOSHIBA MG04AC4A HDD 3726 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 2 INTEL SSDSC2BA80 SSD 745 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 1 TOSHIBA MG04AC4A HDD 3726 False In a Pool  
redmond/RedAz2-Node0 Healthy Slot 16 TOSHIBA MG04AC4A HDD 3726 False In a Pool
```

Check the status of virtual disk repair using the privileged endpoint

After you replace the disk, you can monitor the virtual disk health status and repair job progress by using the privileged endpoint. Follow these steps from any computer that has network connectivity to the privileged endpoint.

1. Open a Windows PowerShell session and connect to the privileged endpoint.

```
$cred = Get-Credential  
Enter-PSSession -ComputerName <IP_address_of_ERCS>`  
-ConfigurationName PrivilegedEndpoint -Credential $cred
```

2. Run the following command to view virtual disk health:

```
Get-VirtualDisk -CimSession s-cluster
```

FriendlyName	ResiliencySettingName	OperationalStatus	HealthStatus	IsManualAttach	Size	PSComputerName
VmTemp_3	Mirror	OK	Healthy	True	1.09 TB	s-cluster
Infrastructure_3	Mirror	OK	Healthy	True	672 GB	s-cluster
ObjStore_4	Mirror	OK	Healthy	True	8.97 TB	s-cluster
VmTemp_4	Mirror	OK	Healthy	True	1.09 TB	s-cluster
VmTemp_2	Mirror	OK	Healthy	True	1.09 TB	s-cluster
ObjStore_3	Mirror	OK	Healthy	True	8.97 TB	s-cluster
Infrastructure_1	Mirror	OK	Healthy	True	1.25 TB	s-cluster
ObjStore_1	Mirror	OK	Healthy	True	8.97 TB	s-cluster
VmTemp_1	Mirror	OK	Healthy	True	1.09 TB	s-cluster
ObjStore_2	Mirror	OK	Healthy	True	8.97 TB	s-cluster
Infrastructure_2	Mirror	OK	Healthy	True	1.6 TB	s-cluster

3. Run the following command to view current storage job status:

```
Get-VirtualDisk -CimSession s-cluster | Get-StorageJob
```

Name	IsBackgroundTask	Elapsed Time	Job State	Percent Complete	Bytes Processed	Bytes Total	PSComputerName
Repair	False	00:00:00	Completed	100			s-cluster
Repair	False	00:00:00	Completed	100			s-cluster
Repair	False	00:00:00	Completed	100			s-cluster
Repair	False	00:00:00	Completed	100			s-cluster
Repair	False	00:00:00	Completed	100			s-cluster
Repair	False	00:00:00	Completed	100			s-cluster
Repair	False	00:00:00	Completed	100			s-cluster
Repair	False	00:00:00	Completed	100			s-cluster
Repair	False	00:00:00	Completed	100			s-cluster
Repair	False	00:00:00	Completed	100			s-cluster
Repair	False	00:00:00	Completed	100			s-cluster
Repair	False	00:00:00	Completed	100			s-cluster
Repair	False	00:00:00	Completed	100			s-cluster
Repair	False	00:00:00	Completed	100			s-cluster

4. Validate the Azure Stack Hub system state. For instructions, see [Validate Azure Stack Hub system state](#).

Troubleshoot virtual disk repair using the privileged endpoint

If the virtual disk repair job appears stuck, run the following command to restart the job:

```
Get-VirtualDisk -CimSession s-cluster | Repair-VirtualDisk
```

Replace a hardware component on an Azure Stack Hub scale unit node

3 minutes to read • [Edit Online](#)

This article describes the general process to replace hardware components that are non hot-swappable. Actual replacement steps vary based on your original equipment manufacturer (OEM) hardware vendor. See your vendor's field replaceable unit (FRU) documentation for detailed steps that are specific to your Azure Stack Hub integrated system.

Caution

Firmware leveling is critical for the success of the operation described in this article. Missing this step can lead to system instability, performance decrease, security threads, or prevent Azure Stack Hub automation from deploying the operating system. Always consult your hardware partner's documentation when replacing hardware to ensure the applied firmware matches the OEM Version displayed in the [Azure Stack Hub administrator portal](#).

HARDWARE PARTNER	REGION	URL
Cisco	All	Cisco Integrated System for Microsoft Azure Stack Hub Operations Guide
		Release Notes for Cisco Integrated System for Microsoft Azure Stack Hub
Dell EMC	All	Cloud for Microsoft Azure Stack Hub 14G (account and sign-in required)
		Cloud for Microsoft Azure Stack Hub 13G (account and sign-in required)
Fujitsu	JAPAN	Fujitsu managed service support desk (account and sign-in required)
	EMEA	Fujitsu support IT products and systems
HPE	EU	Fujitsu MySupport (account and sign-in required)
	All	HPE ProLiant for Microsoft Azure Stack Hub
Lenovo	All	ThinkAgile SXM Best Recipes
Wortmann		OEM/firmware package terra Azure Stack Hub documentation (including FRU)

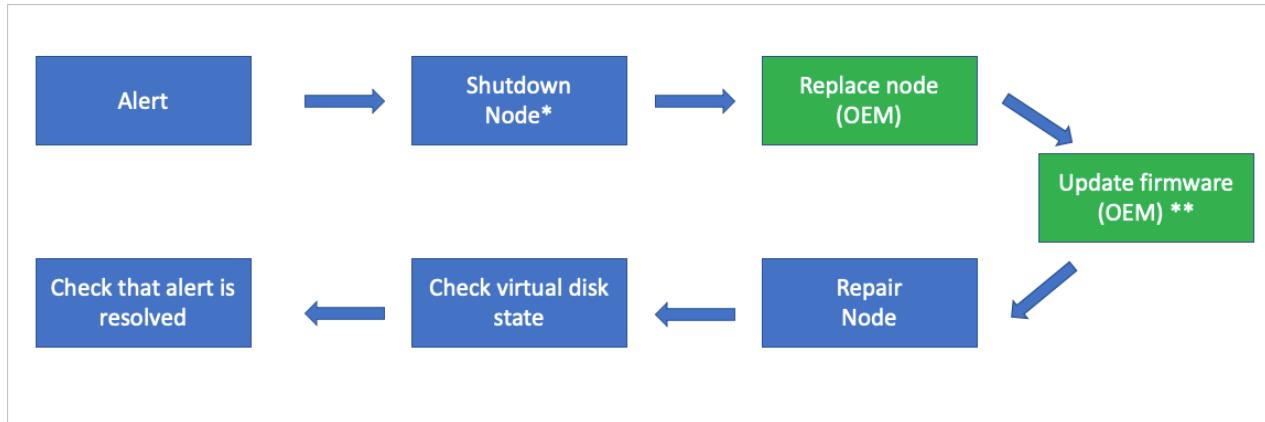
Non hot-swappable components include the following items:

- CPU*
- Memory*

- Motherboard/baseboard management controller (BMC)/video card
- Disk controller/host bus adapter (HBA)/backplane
- Network adapter (NIC)
- Operating system disk*
- Data drives (drives that don't support hot swap, for example PCI-e add-in cards)*

*These components may support hot swap, but can vary based on vendor implementation. See your OEM vendor's FRU documentation for detailed steps.

The following flow diagram shows the general FRU process to replace a non hot-swappable hardware component.



- This action may not be required based on the physical condition of the hardware.

** Whether your OEM hardware vendor does the component replacement and updates the firmware could vary based on your support contract.

Review alert information

The Azure Stack Hub health and monitoring system tracks the health of network adapters and data drives controlled by Storage Spaces Direct. It doesn't track other hardware components. For all other hardware components, alerts are raised in the vendor-specific hardware monitoring solution that runs on the hardware lifecycle host.

Component replacement process

The following steps provide a high-level overview of the component replacement process. Don't follow these steps without referring to your OEM-provided FRU documentation.

1. Use the Shutdown action to gracefully shut down the scale unit node. This action may not be required based on the physical condition of the hardware.
2. In an unlikely case the shutdown action does fail, use the [Drain](#) action to put the scale unit node into maintenance mode. This action may not be required based on the physical condition of the hardware.

NOTE

In any case, only one node can be disabled and powered off at the same time without breaking the S2D (Storage Spaces Direct).

3. After the scale unit node is in maintenance mode, use the [Power off](#) action. This action may not be required based on the physical condition of the hardware.

NOTE

In the unlikely case that the power off action doesn't work, use the baseboard management controller (BMC) web interface instead.

4. Replace the damaged hardware component. Whether your OEM hardware vendor does the component replacement could vary based on your support contract.
5. Update the firmware. Follow your vendor-specific firmware update process using the hardware lifecycle host to make sure the replaced hardware component has the approved firmware level applied. Whether your OEM hardware vendor does this step could vary based on your support contract.
6. Use the [Repair](#) action to bring the scale unit node back into the scale unit.
7. Use the privileged endpoint to [check the status of virtual disk repair](#). With new data drives, a full storage repair job can take multiple hours depending on system load and consumed space.
8. After the repair action has finished, validate that all active alerts have been automatically closed.

Next steps

- For information about replacing a hot-swappable physical disk, see [Replace a disk](#).
- For information about replacing a physical node, see [Replace a scale unit node](#).

Azure Stack Hub infrastructure security controls

5 minutes to read • [Edit Online](#)

Security considerations and compliance regulations are among the main drivers for using hybrid clouds. Azure Stack Hub is designed for these scenarios. This article explains the security controls in place for Azure Stack Hub.

Two security posture layers coexist in Azure Stack Hub. The first layer is the Azure Stack Hub infrastructure, which includes the hardware components up to the Azure Resource Manager. The first layer includes the administrator and the user portals. The second layer consists of the workloads created, deployed, and managed by tenants. The second layer includes items like virtual machines and App Services web sites.

Security approach

The security posture for Azure Stack Hub is designed to defend against modern threats and was built to meet the requirements from the major compliance standards. As a result, the security posture of the Azure Stack Hub infrastructure is built on two pillars:

- **Assume Breach**

Starting from the assumption that the system has already been breached, focus on *detecting and limiting the impact of breaches* versus only trying to prevent attacks.

- **Hardened by Default**

Since the infrastructure runs on well-defined hardware and software, Azure Stack Hub *enables, configures, and validates all the security features* by default.

Because Azure Stack Hub is delivered as an integrated system, the security posture of the Azure Stack Hub infrastructure is defined by Microsoft. Just like in Azure, tenants are responsible for defining the security posture of their tenant workloads. This document provides foundational knowledge on the security posture of the Azure Stack Hub infrastructure.

Data at rest encryption

All Azure Stack Hub infrastructure and tenant data are encrypted at rest using BitLocker. This encryption protects against physical loss or theft of Azure Stack Hub storage components. For more information, see [data at rest encryption in Azure Stack Hub](#).

Data in transit encryption

The Azure Stack Hub infrastructure components communicate using channels encrypted with TLS 1.2. Encryption certificates are self-managed by the infrastructure.

All external infrastructure endpoints, like the REST endpoints or the Azure Stack Hub portal, support TLS 1.2 for secure communications. Encryption certificates, either from a third party or your enterprise Certificate Authority, must be provided for those endpoints.

While self-signed certificates can be used for these external endpoints, Microsoft strongly advises against using them. For more information on how to enforce TLS 1.2 on the external endpoints of Azure Stack Hub, see [Configure Azure Stack Hub security controls](#).

Secret management

Azure Stack Hub infrastructure uses a multitude of secrets, like passwords, to function. Most of them are

automatically rotated frequently because they're group Managed Service Accounts (gMSA), which rotate every 24 hours.

The remaining secrets that aren't gMSA can be rotated manually with a script in the privileged endpoint.

Azure Stack Hub infrastructure uses 4096-bit RSA keys for all its internal certificates. Same key-length certificates can also be used for the external endpoints. For more information on secrets and certificate rotation, please refer to [Rotate secrets in Azure Stack Hub](#).

Windows Defender Application Control

Azure Stack Hub makes use of the latest Windows Server security features. One of them is Windows Defender Application Control (WDAC, formerly known as Code Integrity), which provides executables whitelisting and ensures that only authorized code runs within the Azure Stack Hub infrastructure.

Authorized code is signed by either Microsoft or the OEM partner. The signed authorized code is included in the list of allowed software specified in a policy defined by Microsoft. In other words, only software that has been approved to run in the Azure Stack Hub infrastructure can be executed. Any attempt to execute unauthorized code is blocked and an alert is generated. Azure Stack Hub enforces both User Mode Code Integrity (UMCI) and Hypervisor Code Integrity (HVCI).

The WDAC policy also prevents third-party agents or software from running in the Azure Stack Hub infrastructure. For more information on WDAC, please refer to [Windows Defender Application Control and virtualization-based protection of code integrity](#).

Credential Guard

Another Windows Server security feature in Azure Stack Hub is Windows Defender Credential Guard, which is used to protect Azure Stack Hub infrastructure credentials from Pass-the-Hash and Pass-the-Ticket attacks.

Antimalware

Every component in Azure Stack Hub (both Hyper-V hosts and virtual machines) is protected with Windows Defender Antivirus.

In connected scenarios, antivirus definition and engine updates are applied multiple times a day. In disconnected scenarios, antimalware updates are applied as part of monthly Azure Stack Hub updates. In case a more frequent update to the Windows Defender's definitions is required in disconnected scenarios, Azure Stack Hub also supports importing Windows Defender updates. For more information, see [update Windows Defender Antivirus on Azure Stack Hub](#).

Secure Boot

Azure Stack Hub enforces Secure Boot on all the Hyper-V hosts and infrastructure virtual machines.

Constrained administration model

Administration in Azure Stack Hub is controlled through three entry points, each with a specific purpose:

- The [administrator portal](#) provides a point-and-click experience for daily management operations.
- Azure Resource Manager exposes all the management operations of the administrator portal via a REST API, used by PowerShell and Azure CLI.
- For specific low-level operations (for example, datacenter integration or support scenarios), Azure Stack Hub exposes a PowerShell endpoint called [privileged endpoint](#). This endpoint exposes only a whitelisted set of cmdlets and it's heavily audited.

Network controls

Azure Stack Hub infrastructure comes with multiple layers of network Access Control List (ACL). The ACLs prevent unauthorized access to the infrastructure components and limit infrastructure communications to only the paths that are required for its functioning.

Network ACLs are enforced in three layers:

- Layer 1: Top of Rack switches
- Layer 2: Software Defined Network
- Layer 3: Host and VM operating system firewalls

Regulatory compliance

Azure Stack Hub has gone through a formal capability assessment by a third party-independent auditing firm. As a result, documentation on how the Azure Stack Hub infrastructure meets the applicable controls from several major compliance standards is available. The documentation isn't a certification of Azure Stack Hub because the standards include several personnel-related and process-related controls. Rather, customers can use this documentation to jump-start their certification process.

The assessments include the following standards:

- [PCI-DSS](#) addresses the payment card industry.
- [CSA Cloud Control Matrix](#) is a comprehensive mapping across multiple standards, including FedRAMP Moderate, ISO27001, HIPAA, HITRUST, ITAR, NIST SP800-53, and others.
- [FedRAMP High](#) for government customers.

The compliance documentation can be found on the [Microsoft Service Trust Portal](#). The compliance guides are a protected resource and require you to sign in with your Azure cloud service credentials.

Next steps

- [Configure Azure Stack Hub security controls](#)
- [Learn how to rotate your secrets in Azure Stack Hub](#)
- [PCI-DSS and the CSA-CCM documents for Azure Stack Hub](#)
- [DoD and NIST documents for Azure Stack Hub](#)

Configure Azure Stack Hub security controls

3 minutes to read • [Edit Online](#)

This article explains the security controls that can be changed in Azure Stack Hub and highlights the tradeoffs where applicable.

Azure Stack Hub architecture is built on two security principle pillars: assume breach and hardened by default. For more information Azure Stack Hub security, see [Azure Stack Hub infrastructure security posture](#). While the default security posture of Azure Stack Hub is production-ready, there are some deployment scenarios that require additional hardening.

TLS version policy

The Transport Layer Security (TLS) protocol is a widely adopted cryptographic protocol to establish an encrypted communication over the network. TLS has evolved over time and multiple versions have been released. Azure Stack Hub infrastructure exclusively uses TLS 1.2 for all its communications. For external interfaces, Azure Stack Hub currently defaults to use TLS 1.2. However, for backwards compatibility, it also supports negotiating down to TLS 1.1 and 1.0. When a TLS client requests to communicate over TLS 1.1 or TLS 1.0, Azure Stack Hub honors the request by negotiating to a lower TLS version. If the client requests TLS 1.2, Azure Stack Hub will establish a TLS connection using TLS 1.2.

Since TLS 1.0 and 1.1 are incrementally being deprecated or banned by organizations and compliance standards, beginning with the 1906 update, you can now configure the TLS policy in Azure Stack Hub. You can enforce a TLS 1.2 only policy where any attempt of establishing a TLS session with a version lower than 1.2 is not permitted and rejected.

IMPORTANT

Microsoft recommends using TLS 1.2 only policy for Azure Stack Hub production environments.

Get TLS policy

Use the [privileged endpoint \(PEP\)](#) to view the TLS policy for all Azure Stack Hub endpoints:

```
Get-TLSPolicy
```

Example output:

```
TLS_1.2
```

Set TLS policy

Use the [privileged endpoint \(PEP\)](#) to set the TLS policy for all Azure Stack Hub endpoints:

```
Set-TLSPolicy -Version <String>
```

Parameters for *Set-TLSPolicy* cmdlet:

PARAMETER	DESCRIPTION	TYPE	REQUIRED
<i>Version</i>	Allowed version(s) of TLS in Azure Stack Hub	String	yes

Use one of the following values to configure the permitted TLS versions for all Azure Stack Hub endpoints:

VERSION VALUE	DESCRIPTION
<i>TLS_All</i>	Azure Stack Hub TLS endpoints support TLS 1.2, but down negotiation to TLS 1.1 and TLS 1.0 is allowed.
<i>TLS_1.2</i>	Azure Stack Hub TLS endpoints support TLS 1.2 only.

Updating the TLS policy takes a few minutes to complete.

Enforce TLS 1.2 configuration example

This example sets your TLS policy to enforce TLS 1.2 only.

```
Set-TLSPolicy -Version TLS_1.2
```

Example output:

```
VERBOSE: Successfully setting enforce TLS 1.2 to True
VERBOSE: Invoking action plan to update GPOs
VERBOSE: Create Client for execution of action plan
VERBOSE: Start action plan
<....>
VERBOSE: Verifying TLS policy
VERBOSE: Get GPO TLS protocols registry 'enabled' values
VERBOSE: GPO TLS applied with the following preferences:
VERBOSE:     TLS protocol SSL 2.0 enabled value: 0
VERBOSE:     TLS protocol SSL 3.0 enabled value: 0
VERBOSE:     TLS protocol TLS 1.0 enabled value: 0
VERBOSE:     TLS protocol TLS 1.1 enabled value: 0
VERBOSE:     TLS protocol TLS 1.2 enabled value: 1
VERBOSE: TLS 1.2 is enforced
```

Allow all versions of TLS (1.2, 1.1 and 1.0) configuration example

This example sets your TLS policy to allow all versions of TLS (1.2, 1.1 and 1.0).

```
Set-TLSPolicy -Version TLS_All
```

Example output:

```
VERBOSE: Successfully setting enforce TLS 1.2 to False
VERBOSE: Invoking action plan to update GPOs
VERBOSE: Create Client for execution of action plan
VERBOSE: Start action plan
<...>
VERBOSE: Verifying TLS policy
VERBOSE: Get GPO TLS protocols registry 'enabled' values
VERBOSE: GPO TLS applied with the following preferences:
VERBOSE:     TLS protocol SSL 2.0 enabled value: 0
VERBOSE:     TLS protocol SSL 3.0 enabled value: 0
VERBOSE:     TLS protocol TLS 1.0 enabled value: 1
VERBOSE:     TLS protocol TLS 1.1 enabled value: 1
VERBOSE:     TLS protocol TLS 1.2 enabled value: 1
VERBOSE: TLS 1.2 is not enforced
```

Next steps

- [Learn about Azure Stack Hub infrastructure security posture](#)
- [Learn how to rotate your secrets in Azure Stack Hub](#)
- [Update Windows Defender Antivirus on Azure Stack Hub](#)

Overview of identity providers for Azure Stack Hub

10 minutes to read • [Edit Online](#)

Azure Stack Hub requires Azure Active Directory (Azure AD) or Active Directory Federation Services (AD FS), backed by Active Directory as an identity provider. The choice of a provider is a one-time decision that you make when you first deploy Azure Stack Hub. The concepts and authorization details in this article can help you choose between identity providers.

Your choice of either Azure AD or AD FS is determined by the mode in which you deploy Azure Stack Hub:

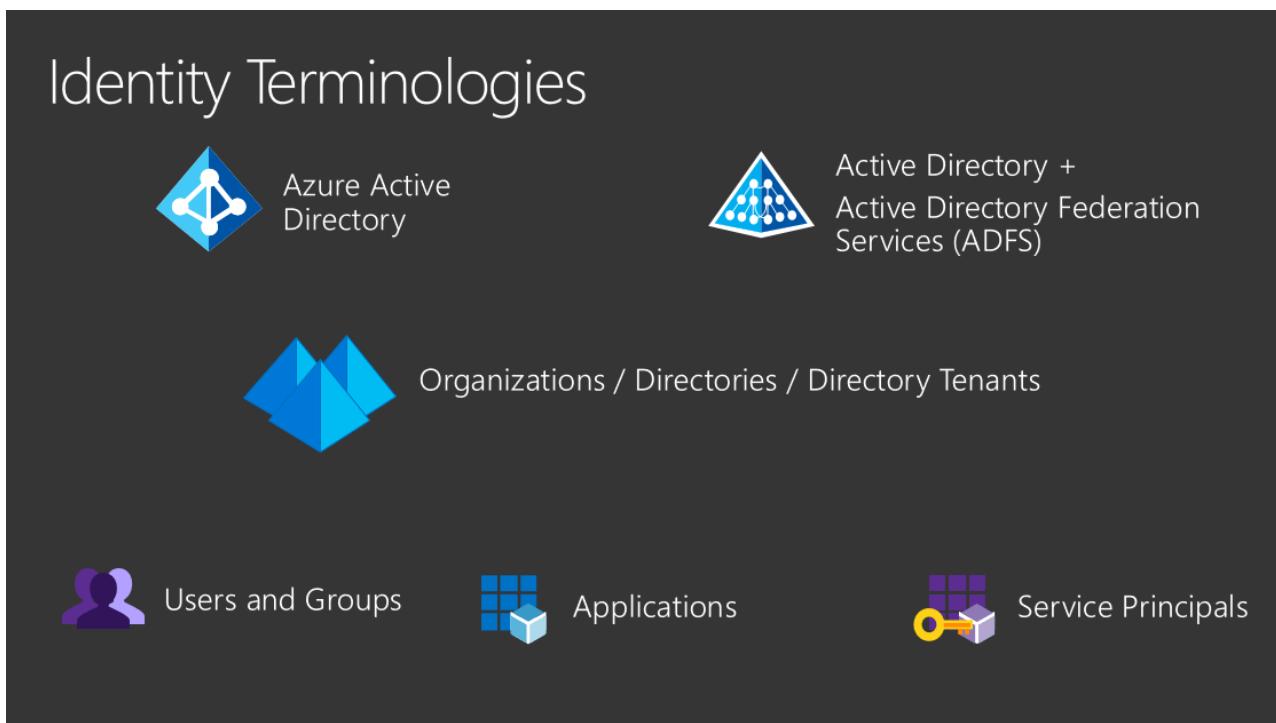
- When you deploy it in a connected mode, you can use either Azure AD or AD FS.
- When you deploy it in a disconnected mode, without a connection to the internet, only AD FS is supported.

For more information about your options, which depend on your Azure Stack Hub environment, see the following articles:

- Azure Stack Hub deployment kit: [Identity considerations](#).
- Azure Stack Hub integrated systems: [Deployment planning decisions for Azure Stack Hub integrated systems](#).

Common concepts for identity providers

The next sections discuss common concepts about identity providers and their use in Azure Stack Hub.



Directory tenants and organizations

A directory is a container that holds information about *users*, *applications*, *groups*, and *service principals*.

A directory tenant is an *organization*, such as Microsoft or your own company.

- Azure AD supports multiple tenants, and it can support multiple organizations, each in its own directory. If you use Azure AD and have multiple tenants, you can grant apps and users from one tenant access to other tenants of that same directory.
- AD FS supports only a single tenant and, therefore, only a single organization.

Users and groups

User accounts (identities) are standard accounts that authenticate individuals by using a user ID and password. Groups can include users or other groups.

How you create and manage users and groups depends on the identity solution you use.

In Azure Stack Hub, user accounts:

- Are created in the *username@domain* format. Although AD FS maps user accounts to an Active Directory instance, AD FS doesn't support the use of the \<domain>\<alias> format.
- Can be set up to use multi-factor authentication.
- Are restricted to the directory where they first register, which is their organization's directory.
- Can be imported from your on-premises directories. For more information, see [Integrate your on-premises directories with Azure Active Directory](#).

When you sign in to your organization's user portal, you use the <https://portal.local.azurestack.external> URL. When signing into the Azure Stack Hub portal from domains other than the one used to register Azure Stack Hub, the domain name used to register Azure Stack Hub must be appended to the portal url. For example, if Azure Stack Hub has been registered with fabrikam.onmicrosoft.com and the user account logging in is admin@contoso.com, the URL to use to log into the user portal would be: <https://portal.local.azurestack.external/fabrikam.onmicrosoft.com>.

Guest users

Guest users are user accounts from other directory tenants that have been granted access to resources in your directory. To support guest users, you use Azure AD and enable support for multi-tenancy. When support is enabled, you can invite guest users to access resources in your directory tenant, which in turn enables their collaboration with outside organizations.

To invite guest users, cloud operators and users can use [Azure AD B2B collaboration](#). Invited users get access to documents, resources, and apps from your directory, and you maintain control over your own resources and data.

As a guest user, you can sign in to another organization's directory tenant. To do so, you append that organization's directory name to the portal URL. For example, if you belong to the Contoso organization and want to sign in to the Fabrikam directory, you use <https://portal.local.azurestack.external/fabrikam.onmicrosoft.com>.

Apps

You can register apps to Azure AD or AD FS, and then offer the apps to users in your organization.

Apps include:

- **Web apps:** Examples include the Azure portal and Azure Resource Manager. They support Web API calls.
- **Native client:** Examples include Azure PowerShell, Visual Studio, and Azure CLI.

Apps can support two types of tenancy:

- **Single-tenant:** Supports users and services only from the same directory where the app is registered.

NOTE

Because AD FS supports only a single directory, apps you create in an AD FS topology are, by design, single-tenant apps.

- **Multi-tenant:** Supports use by users and services from both the directory where the app is registered and additional tenant directories. With multi-tenant apps, users of another tenant directory (another Azure AD tenant) can sign in to your app.

For more information about multi-tenancy, see [Enable multi-tenancy](#).

For more information about developing a multi-tenant app, see [Multi-tenant apps](#).

When you register an app, you create two objects:

- **Application object:** The global representation of the app across all tenants. This relationship is one-to-one with the software app and exists only in the directory where the app is first registered.
- **Service principal object:** A credential that's created for an app in the directory where the app is first registered. A service principal is also created in the directory of each additional tenant where that app is used. This relationship can be one-to-many with the software app.

To learn more about app and service principal objects, see [Application and service principal objects in Azure Active Directory](#).

Service principals

A service principal is a set of *credentials* for an app or service that grant access to resources in Azure Stack Hub. The use of a service principal separates the app permissions from the permissions of the user of the app.

A service principal is created in each tenant where the app is used. The service principal establishes an identity for sign-in and access to resources (such as users) that are secured by that tenant.

- A single-tenant app has only one service principal, which is in the directory where it's first created. This service principal is created and consents to being used during registration of the app.
- A multi-tenant web app or API has a service principal that's created in each tenant where a user from that tenant consents to the use of the app.

Credentials for service principals can be either a key that's generated through the Azure portal or a certificate. The use of a certificate is suited for automation because certificates are considered more secure than keys.

NOTE

When you use AD FS with Azure Stack Hub, only the administrator can create service principals. With AD FS, service principals require certificates and are created through the privileged endpoint (PEP). For more information, see [Use an app identity to access resources](#).

To learn about service principals for Azure Stack Hub, see [Create service principals](#).

Services

Services in Azure Stack Hub that interact with the identity provider are registered as apps with the identity provider. Like apps, registration enables a service to authenticate with the identity system.

All Azure services use [OpenID Connect](#) protocols and [JSON Web Tokens](#) to establish their identity. Because Azure AD and AD FS use protocols consistently, you can use [Azure Active Directory Authentication Library](#) (ADAL) to authenticate on-premises or to Azure (in a connected scenario). With ADAL, you can also use tools such as Azure PowerShell and Azure CLI for cross-cloud and on-premises resource management.

Identities and your identity system

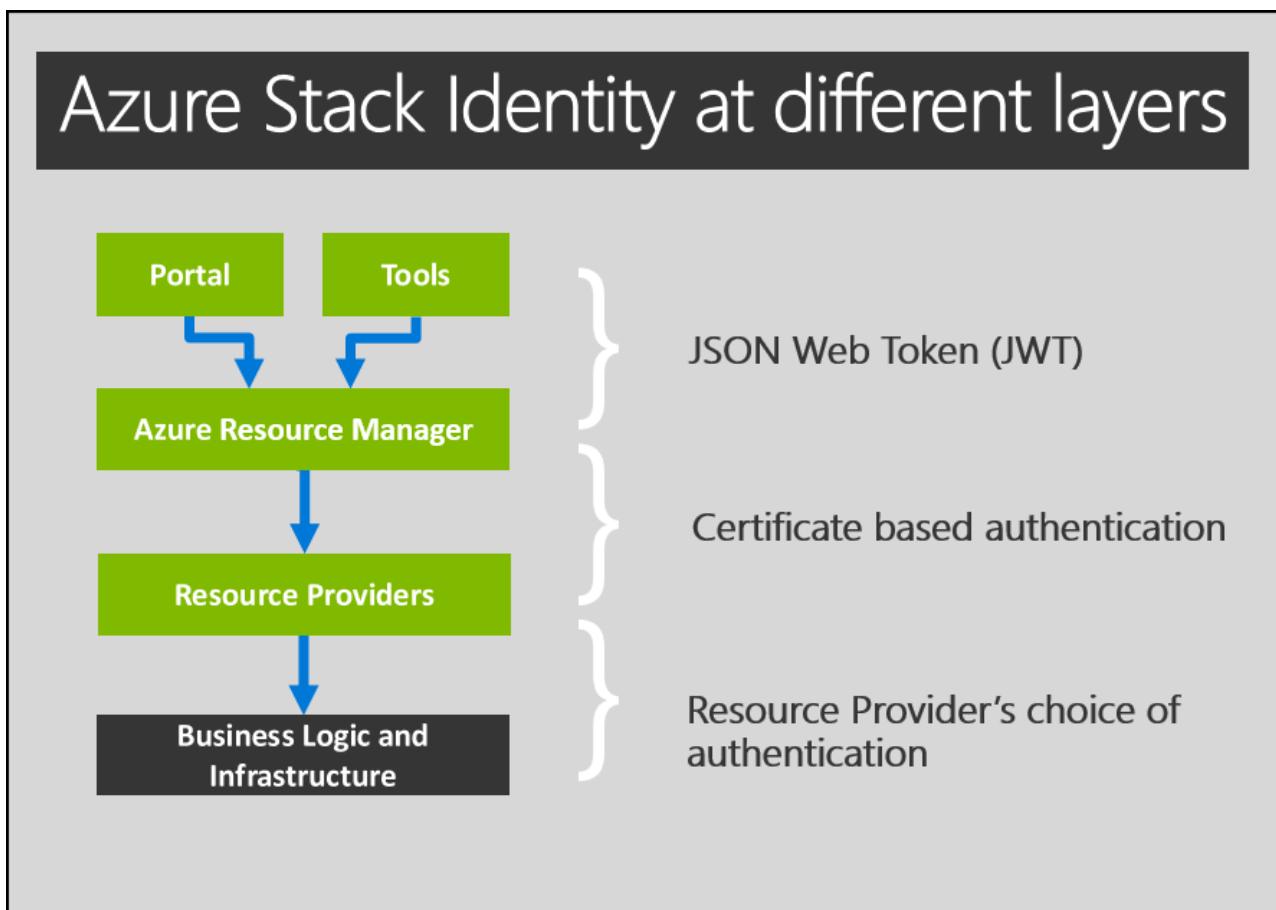
Identities for Azure Stack Hub include user accounts, groups, and service principals.

When you install Azure Stack Hub, several built-in apps and services automatically register with your identity provider in the directory tenant. Some services that register are used for administration. Other services are available for users. The default registrations give core services identities that can interact both with each other and with identities that you add later.

If you set up Azure AD with multi-tenancy, some apps propagate to the new directories.

Authentication and authorization

Authentication by apps and users

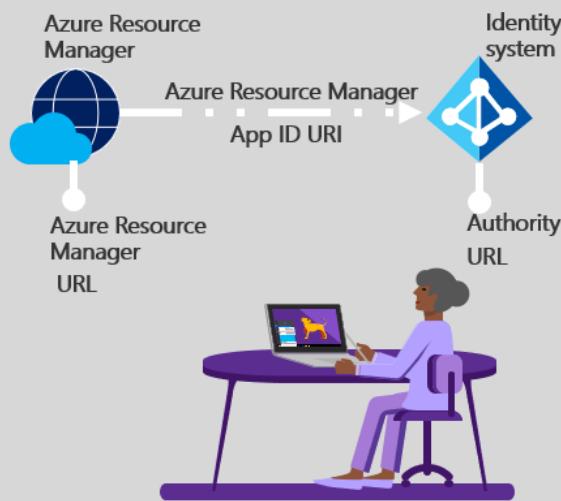


For apps and users, the architecture of Azure Stack Hub is described by four layers. Interactions between each of these layers can use different types of authentication.

LAYER	AUTHENTICATION BETWEEN LAYERS
Tools and clients, such as the administrator portal	To access or modify a resource in Azure Stack Hub, tools and clients use a JSON Web Token to place a call to Azure Resource Manager. Azure Resource Manager validates the JSON Web Token and peeks at the <i>claims</i> in the issued token to estimate the level of authorization that user or service principal has in Azure Stack Hub.
Azure Resource Manager and its core services	Azure Resource Manager communicates with resource providers to transfer communication from users. Transfers use <i>direct imperative</i> calls or <i>declarative</i> calls via Azure Resource Manager templates .
Resource providers	Calls passed to resource providers are secured with certificate-based authentication. Azure Resource Manager and the resource provider then stay in communication through an API. For every call that's received from Azure Resource Manager, the resource provider validates the call with that certificate.
Infrastructure and business logic	Resource providers communicate with business logic and infrastructure by using an authentication mode of their choice. The default resource providers that ship with Azure Stack Hub use Windows Authentication to secure this communication.

Information needed for Authentication

- Identity System's URL (Authority)
Specific to the installation of the cloud
- Azure Resource Manager App Identifier URL
Specific to the installation of the cloud
- Credentials
Common across clouds for hybrid
- Azure Resource Manager URL
Specific to the installation of the cloud



Authenticate to Azure Resource Manager

To authenticate with the identity provider and receive a JSON Web Token, you must have the following information:

1. **URL for the identity system (Authority)**: The URL at which your identity provider can be reached. For example, <https://login.windows.net>.
2. **App ID URI for Azure Resource Manager**: The unique identifier for Azure Resource Manager that's registered with your identity provider. It's also unique to each Azure Stack Hub installation.
3. **Credentials**: The credential you use to authenticate with the identity provider.
4. **URL for Azure Resource Manager**: The URL is the location of the Azure Resource Manager service. For example, <https://management.azure.com> or <https://management.local.azurestack.external>.

When a principal (a client, apps, or user) makes an authentication request to access a resource, the request must include:

- The principal's credentials.
- The app ID URI of the resource that the principal wants to access.

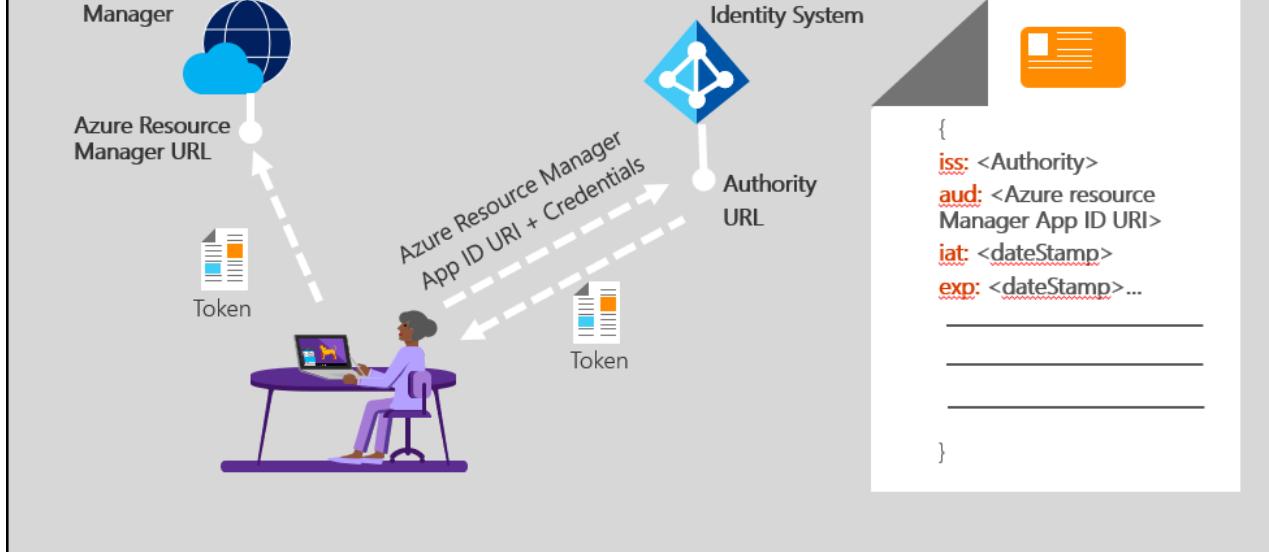
The credentials are validated by the identity provider. The identity provider also validates that the app ID URI is for a registered app, and that the principal has the correct privileges to obtain a token for that resource. If the request is valid, a JSON Web Token is granted.

The token must then pass in the header of a request to Azure Resource Manager. Azure Resource Manager does the following, in no specific order:

- Validates the *issuer* (iss) claim to confirm that the token is from the correct identity provider.
- Validates the *audience* (aud) claim to confirm that the token was issued to Azure Resource Manager.
- Validates that the JSON Web Token is signed with a certificate that's configured through OpenID and known to Azure Resource Manager.
- Review the *issued at* (iat) and *expiration* (exp) claims to confirm that the token is active and can be accepted.

When all validations are complete, Azure Resource Manager uses the *object id* (oid) and the *groups* claims to make a list of resources that the principal can access.

Token Exchange Protocol



NOTE

After deployment, Azure Active Directory global administrator permission isn't required. However, some operations may require the global admin credentials (for example, a resource provider installer script or a new feature requiring a permission to be granted). You can either temporarily re-instate the account's global admin permissions or use a separate global admin account that's an owner of the *default provider subscription*.

Use Role-Based Access Control

Role-Based Access Control (RBAC) in Azure Stack Hub is consistent with the implementation in Microsoft Azure. You can manage access to resources by assigning the appropriate RBAC role to users, groups, and apps. For information about how to use RBAC with Azure Stack Hub, see the following articles:

- [Get started with Role-Based Access Control in the Azure portal](#).
- [Use Role-Based Access Control to manage access to your Azure subscription resources](#).
- [Create custom roles for Azure Role-Based Access Control](#).
- [Manage Role-Based Access Control in Azure Stack Hub](#).

Authenticate with Azure PowerShell

Details about using Azure PowerShell to authenticate with Azure Stack Hub can be found at [Configure the Azure Stack Hub user's PowerShell environment](#).

Authenticate with Azure CLI

For information about using Azure PowerShell to authenticate with Azure Stack Hub, see [Install and configure Azure CLI for use with Azure Stack Hub](#).

Next steps

- [Identity architecture](#)
- [Datacenter integration - identity](#)

Identity architecture for Azure Stack Hub

3 minutes to read • [Edit Online](#)

When choosing an identity provider to use with Azure Stack Hub, you should understand the important differences between the options of Azure Active Directory (Azure AD) and Active Directory Federation Services (AD FS).

Capabilities and limitations

The identity provider that you choose can limit your options, including support for multi-tenancy.

CAPABILITY OR SCENARIO	AZURE AD	AD FS
Connected to the internet	Yes	Optional
Support for multi-tenancy	Yes	No
Offer items in the Marketplace	Yes	Yes (requires use of the offline Marketplace Syndication tool)
Support for Active Directory Authentication Library (ADAL)	Yes	Yes
Support for tools such as Azure CLI, Visual Studio, and PowerShell	Yes	Yes
Create service principals through the Azure portal	Yes	No
Create service principals with certificates	Yes	Yes
Create service principals with secrets (keys)	Yes	Yes
Applications can use the Graph service	Yes	No
Applications can use identity provider for sign-in	Yes	Yes (requires apps to federate with on-premises AD FS instances)

Topologies

The following sections discuss the different identity topologies that you can use.

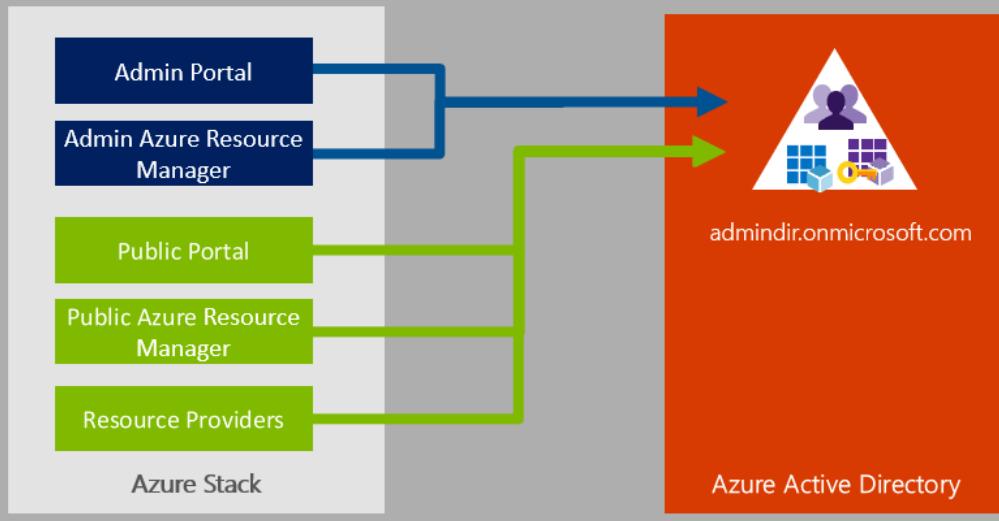
Azure AD: single-tenant topology

By default, when you install Azure Stack Hub and use Azure AD, Azure Stack Hub uses a single-tenant topology.

A single-tenant topology is useful when:

- All users are part of the same tenant.
- A service provider hosts an Azure Stack Hub instance for an organization.

Azure Stack with Azure AD – Single Tenant



Use cases: Enterprises, dedicated hosting

This topology features the following characteristics:

- Azure Stack Hub registers all apps and services to the same Azure AD tenant directory.
- Azure Stack Hub authenticates only the users and apps from that directory, including tokens.
- Identities for administrators (cloud operators) and tenant users are in the same directory tenant.
- To enable a user from another directory to access this Azure Stack Hub environment, you must [invite the user as a guest](#) to the tenant directory.

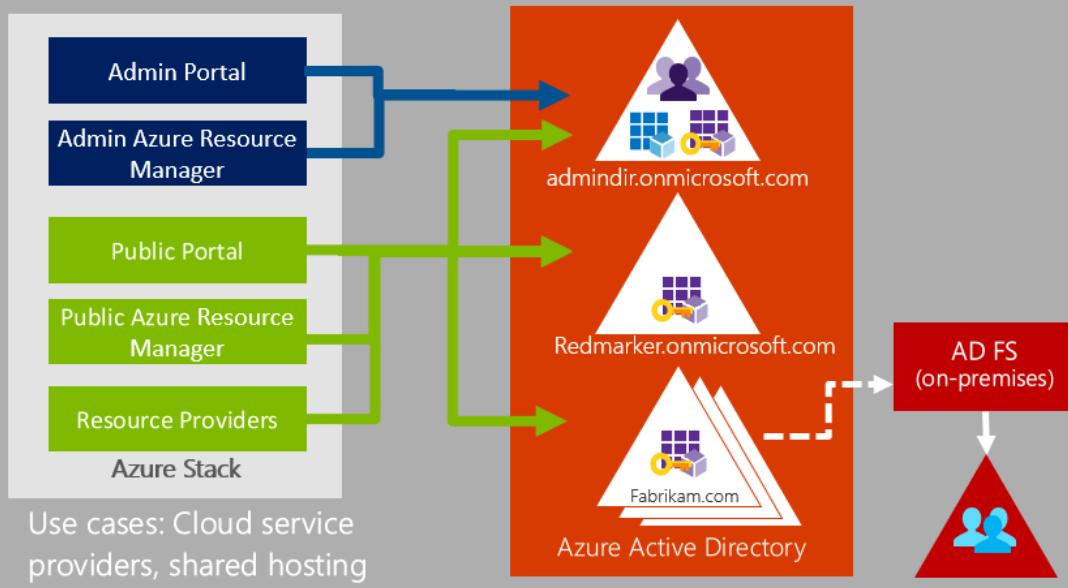
Azure AD: multi-tenant topology

Cloud operators can configure Azure Stack Hub to allow access to apps by tenants from one or more organizations. Users access apps through the Azure Stack Hub user portal. In this configuration, the administrator portal (used by the cloud operator) is limited to users from a single directory.

A multi-tenant topology is useful when:

- A service provider wants to allow users from multiple organizations to access Azure Stack Hub.

Azure Stack with Azure AD – Multi Tenant



This topology features the following characteristics:

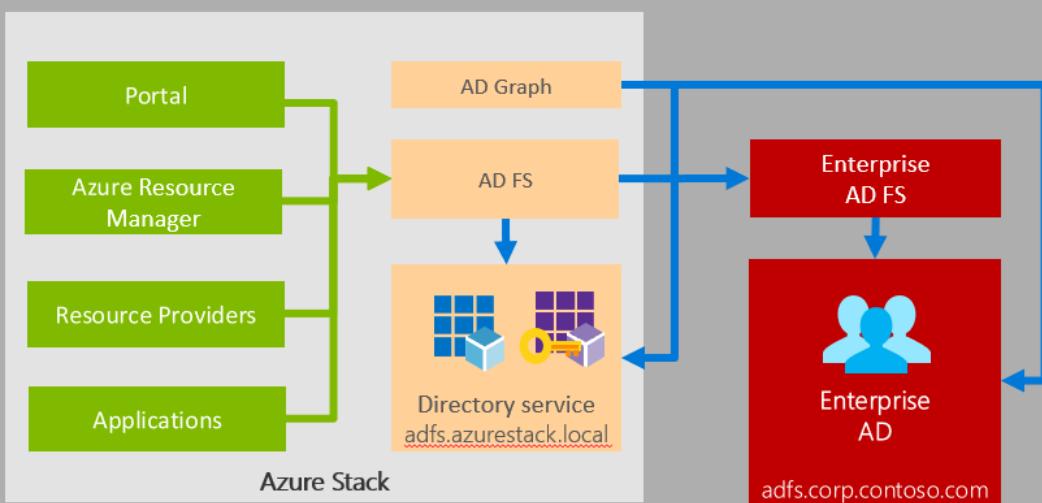
- Access to resources should be on a per-organization basis.
- Users from one organization should be unable to grant access to resources to users who are outside their organization.
- Identities for administrators (cloud operators) can be in a separate directory tenant from the identities for users. This separation provides account isolation at the identity provider level.

AD FS

The AD FS topology is required when either of the following conditions is true:

- Azure Stack Hub doesn't connect to the internet.
- Azure Stack Hub can connect to the internet, but you choose to use AD FS for your identity provider.

Azure Stack with AD FS



This topology features the following characteristics:

- To support the use of this topology in production, you must integrate the built-in Azure Stack Hub AD FS instance with an existing AD FS instance that's backed by Active Directory, through a federation trust.
- You can integrate the Graph service in Azure Stack Hub with your existing Active Directory instance. You can also use the OData-based Graph API service that supports APIs that are consistent with the Azure AD Graph API.

To interact with your Active Directory instance, the Graph API requires user credentials from your Active Directory instance that have read-only permissions.

- The built-in AD FS instance is based on Windows Server 2016.
- Your AD FS and Active Directory instances must be based on Windows Server 2012 or later.

Between your Active Directory instance and the built-in AD FS instance, interactions aren't restricted to OpenID Connect, and they can use any mutually supported protocol.

- User accounts are created and managed in your on-premises Active Directory instance.
- Service principals and registrations for apps are managed in the built-in Active Directory instance.

Next steps

- [Identity overview](#)
- [Datacenter integration - identity](#)

Set access permissions using role-based access control

2 minutes to read • [Edit Online](#)

A user in Azure Stack Hub can be a reader, owner, or contributor for each instance of a subscription, resource group, or service. For example, User A might have reader permissions to Subscription One, but have owner permissions to Virtual Machine Seven.

- Reader: User can view everything, but can't make any changes.
- Contributor: User can manage everything except access to resources.
- Owner: User can manage everything, including access to resources.
- Custom: User has limited, specific access to resources.

For more information about creating a custom role, see [Custom roles for Azure resources](#).

Set access permissions for a user

1. Sign in with an account that has owner permissions to the resource you want to manage.
2. In the blade for the resource, click the **Access** icon .
3. In the **Users** blade, click **Roles**.
4. In the **Roles** blade, click **Add** to add permissions for the user.

Set access permissions for a universal group

NOTE

Applicable only to Active Directory Federated Services (AD FS).

1. Sign in with an account that has owner permissions to the resource you want to manage.
2. In the blade for the resource, click the **Access** icon .
3. In the **Users** blade, click **Roles**.
4. In the **Roles** blade, click **Add** to add permissions for the Universal Group Active Directory Group.

Next steps

[Add an Azure Stack Hub tenant](#)

Add a new Azure Stack Hub user account in Azure Active Directory (Azure AD)

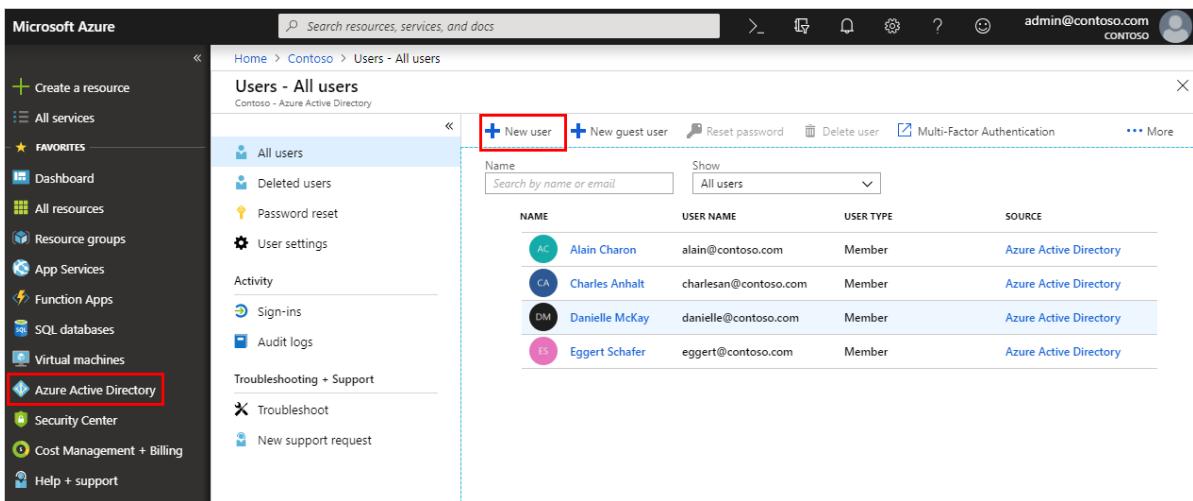
2 minutes to read • [Edit Online](#)

Before you can test offers and plans, and create resources, you'll need a user account. You create a user account in your Azure AD tenant, by using the Azure portal or PowerShell.

Create user account using the Azure portal

You must have an Azure subscription to use the Azure portal.

1. Sign in to [Azure](#).
2. In the left navigation bar, select **Active Directory** and switch to the directory that you want to use for Azure Stack Hub (or create a new one).
3. Select **Azure Active Directory > Users > New user**.



The screenshot shows the Azure portal's 'Users - All users' page for the 'Contoso - Azure Active Directory'. The left sidebar has 'Azure Active Directory' selected. The main area shows a list of users with columns for NAME, USER NAME, USER TYPE, and SOURCE. At the top, there are buttons for '+ New user' (highlighted with a red box), '+ New guest user', 'Reset password', 'Delete user', and 'Multi-Factor Authentication'. A search bar at the top allows filtering by name or email.

NAME	USER NAME	USER TYPE	SOURCE
AC	Alain Charon	Member	Azure Active Directory
CA	Charles Anhalt	Member	Azure Active Directory
DM	Danielle McKay	Member	Azure Active Directory
ES	Eggert Schafer	Member	Azure Active Directory

4. On the **User** page, fill out the required info.

Home > Contoso > Users - All users > User

User

Contoso

* Name ⓘ

Mary Parker ✓

* User name ⓘ

mary@contoso.com ✓

Profile ⓘ >

Not configured

Properties ⓘ >

Default

Groups ⓘ >

0 groups selected

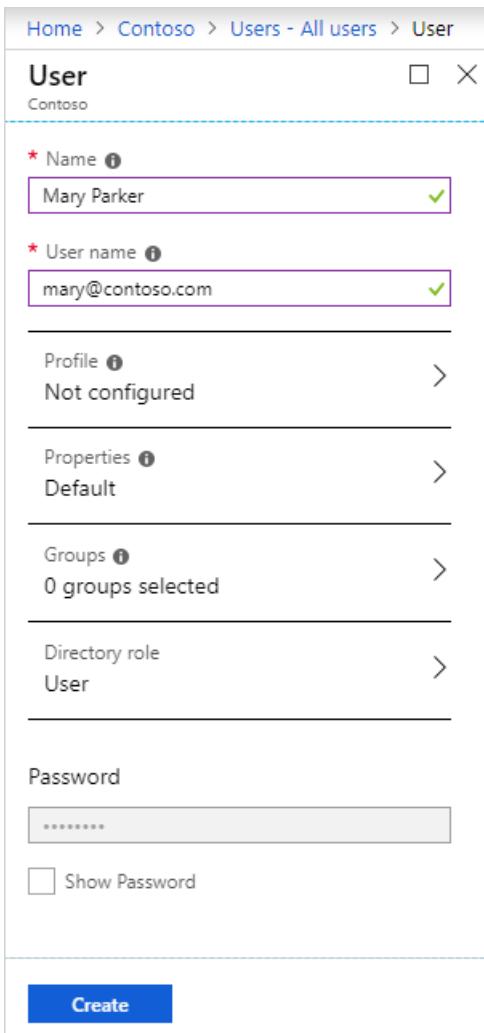
Directory role >

User

Password

Show Password

Create



- **Name (required)**: The first and last name of the new user. For example, Mary Parker.
- **User name (required)**: The user name of the new user. For example, mary@contoso.com. The domain part of the user name must use either the initial default domain name, <yourdomainname>.onmicrosoft.com, or a custom domain name, such as contoso.com. For more info on how to create a custom domain name, see [How to add a custom domain name to Azure AD](#).
- **Profile**: Optionally, you can add more info about the user. You can also add user info at a later time. For more info on adding user info, see [How to add or change user profile information](#).
- **Directory role**: choose **User**.

5. Check **Show Password** and copy the autogenerated password provided in the **Password** box. You'll need this password for the initial sign-in process.

6. Select **Create**.

The user is created and added to your Azure AD tenant.

7. Sign in to the Azure portal with the new account. Change the password when prompted.

8. Sign in to <https://portal.local.azurestack.external> with the new account to see the user portal.

Create a user account using PowerShell

If you don't have an Azure subscription, you can't use the Azure portal to add a tenant user account. In this case, you can use the Azure AD Module for Windows PowerShell instead.

NOTE

If you're using Microsoft Account to deploy the ASDK, you can't use Azure AD PowerShell to create a tenant account.

1. Install the **64-bit** version of the [Microsoft Online Services Sign-in Assistant for IT Professionals RTW](#).
2. Install the Microsoft Azure AD Module for Windows PowerShell with these steps:
 - Open an elevated Windows PowerShell command prompt (run Windows PowerShell as admin).
 - Run the **Install-Module MSOnline** command.
 - If you're prompted to install the NuGet provider, select **Y** and **Enter**.
 - If you're prompted to install the module from PSGallery, select **Y** and **Enter**.
3. Run the following cmdlets:

```
# Provide the Azure AD credential you use to deploy the ASDK.

$msolcred = get-credential

# Add a user account "Tenant Admin <username>@<yourdomainname>" with the initial password "<password>".

connect-msolservice -credential $msolcred
$user = new-msoluser -DisplayName "Tenant Admin" -UserPrincipalName <username>@<yourdomainname>
>Password <password>
Add-MsolRoleMember -RoleName "Company Administrator" -RoleMemberType User -RoleMemberObjectId
$user.ObjectId
```

4. Sign in to Azure with the new account. Change the password when prompted.
5. Sign in to <https://portal.local.azurestack.external> with the new account to see the user portal.

Next steps

[Add Azure Stack Hub users in AD FS](#)

Add a new Azure Stack Hub user account in Active Directory Federation Services (AD FS)

2 minutes to read • [Edit Online](#)

You can use the **Active Directory Users and Computers** snap-in to add additional users to an Azure Stack Hub environment, using AD FS as its identity provider.

Add Windows Server Active Directory users

1. Sign in to a computer with an account that provides access to the Windows Administrative Tools and open a new Microsoft Management Console (MMC).
2. Select **File > Add or remove snap-in.**

TIP

Replace *directory-domain* with the domain that matches your directory.

3. Select **Active Directory Users and Computers** > *directory-domain* > **Users**.
4. Select **Action > New > User**.
5. In New Object - User, provide user details. Select **Next**.
6. Provide and confirm a password.
7. Select **Next** to complete the values. Select **Finish** to create the user.

Next steps

[Create an app identity to access Azure Stack Hub resources](#)

Use an app identity to access Azure Stack Hub resources

14 minutes to read • [Edit Online](#)

An application that needs to deploy or configure resources through Azure Resource Manager must be represented by a service principal. Just as a user is represented by a user principal, a service principal is a type of security principal that represents an app. The service principal provides an identity for your app, allowing you to delegate only the necessary permissions to that service principal.

As an example, you may have a configuration management app that uses Azure Resource Manager to inventory Azure resources. In this scenario, you can create a service principal, grant the reader role to that service principal, and limit the configuration management app to read-only access.

Overview

Like a user principal, a service principal must present credentials during authentication. This authentication consists of two elements:

- An **Application ID**, sometimes referred to as a Client ID. This is a GUID that uniquely identifies the app's registration in your Active Directory tenant.
- A **secret** associated with the application ID. You can either generate a client secret string (similar to a password), or specify an X509 certificate (which uses its public key).

Running an app under the identity of a service principal is preferable to running it under a user principal because:

- A service principal can use an X509 certificate for **stronger credentials**.
- You can assign **more restrictive permissions** to a service principal. Typically, these permissions are restricted to only what the app needs to do, known as the *principle of least privilege*.
- Service principal **credentials and permissions don't change as frequently** as user credentials. For example, when the user's responsibilities change, password requirements dictate a change, or a user leaves the company.

You start by creating a new app registration in your directory, which creates an associated [service principal object](#) to represent the app's identity within the directory. This document describes the process of creating and managing a service principal, depending on the directory you chose for your Azure Stack Hub instance:

- Azure Active Directory (Azure AD). Azure AD is a multi-tenant, cloud-based directory, and identity management service. You can use Azure AD with a connected Azure Stack Hub instance.
- Active Directory Federation Services (AD FS). AD FS provides simplified, secured identity federation, and web single sign-on (SSO) capabilities. You can use AD FS with both connected and disconnected Azure Stack Hub instances.

First you learn how to manage a service principal, then how to assign the service principal to a role, limiting its resource access.

Manage an Azure AD service principal

If you deployed Azure Stack Hub with Azure AD as your identity management service, you can create service principals just like you do for Azure. This section shows you how to perform the steps through the Azure portal. Check that you have the [required Azure AD permissions](#) before beginning.

Create a service principal that uses a client secret credential

In this section, you register your app using the Azure portal, which creates the service principal object in your Azure AD tenant. In this example, the service principal is created with a client secret credential, but the portal also supports X509 certificate-based credentials.

1. Sign in to the [Azure portal](#) using your Azure account.
2. Select **Azure Active Directory > App registrations > New registration**.
3. Provide a **name** for the app.
4. Select the appropriate **Supported account types**.
5. Under **Redirect URI**, select **Web** as the app type, and (optionally) specify a redirect URI if your app requires it.
6. After setting the values, select **Register**. The app registration is created and the **Overview** page displays.
7. Copy the **Application ID** for use in your app code. This value is also referred to as the Client ID.
8. To generate a client secret, select the **Certificates & secrets** page. Select **New client secret**.
9. Provide a **description** for the secret, and an **expires** duration.
10. When done, select **Add**.
11. The value of the secret displays. Copy and save this value in another location, because you can't retrieve it later. You provide the secret with the Application ID in your client app during service principal sign-in.

Client secrets		
A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.		
+ New client secret		
DESCRIPTION	EXPIRES	VALUE
Clientsecret	12/31/2299	yoursecretmzb?5?]G1g3s?sPCDF2xu 

Manage an AD FS service principal

If you deployed Azure Stack Hub with AD FS as your identity management service, you must use PowerShell to manage the service principal. Examples are provided below for managing service principal credentials, demonstrating both an X509 certificate and a client secret.

The scripts must be run in an elevated ("Run as administrator") PowerShell console, which opens another session to a VM that hosts a privileged endpoint for your Azure Stack Hub instance. Once the privileged endpoint session has been established, additional cmdlets will execute and manage the service principal. For more information about the privileged endpoint, see [Using the privileged endpoint in Azure Stack Hub](#).

Create a service principal that uses a certificate credential

When creating a certificate for a service principal credential, the following requirements must be met:

- For production, the certificate must be issued from either an internal Certificate Authority or a Public Certificate Authority. If you use a public certificate authority, you must include the authority in the base operating system image as part of the Microsoft Trusted Root Authority Program. You can find the full list at [Microsoft Trusted Root Certificate Program: Participants](#). An example of creating a "self-signed" test certificate will also be shown later during [Update a service principal's certificate credential](#).
- The cryptographic provider must be specified as a Microsoft legacy Cryptographic Service Provider (CSP) key provider.
- The certificate format must be in PFX file, as both the public and private keys are required. Windows servers use .pfx files that contain the public key file (SSL certificate file) and the associated private key file.

- Your Azure Stack Hub infrastructure must have network access to the certificate authority's Certificate Revocation List (CRL) location published in the certificate. This CRL must be an HTTP endpoint.

Once you have a certificate, use the PowerShell script below to register your app and create a service principal. You also use the service principal to sign in to Azure. Substitute your own values for the following placeholders:

PLACEHOLDER	DESCRIPTION	EXAMPLE
<PepVM>	The name of the privileged endpoint VM on your Azure Stack Hub instance.	"AzS-ERCS01"
<YourCertificateLocation>	The location of your X509 certificate in the local certificate store.	"Cert:\CurrentUser\My\AB5A8A3533C C7AA2025BF05120117E06DE407B34"
<YourAppName>	A descriptive name for the new app registration.	"My management tool"

1. Open an elevated Windows PowerShell session, and run the following script:

```

# Sign in to PowerShell interactively, using credentials that have access to the VM running the
# Privileged Endpoint (typically <domain>\cloudadmin)
$creds = Get-Credential

# Create a PSSession to the Privileged Endpoint VM
$session = New-PSSession -ComputerName "<PepVm>" -ConfigurationName PrivilegedEndpoint -Credential
$creds

# Use the Get-Item cmdlet to retrieve your certificate.
# If you don't want to use a managed certificate, you can produce a self signed cert for testing
purposes:
# $cert = New-SelfSignedCertificate -CertStoreLocation "cert:\CurrentUser\My" -Subject "CN=
<YourAppName>" -KeySpec KeyExchange
$crt = Get-Item "<YourCertificateLocation>"

# Use the privileged endpoint to create the new app registration (and service principal object)
$spObject = Invoke-Command -Session $session -ScriptBlock {New-GraphApplication -Name "<YourAppName>" -ClientCertificates $using:crt}
$AzureStackInfo = Invoke-Command -Session $session -ScriptBlock {Get-AzureStackStampInformation}
$session | Remove-PSSession

# Using the stamp info for your Azure Stack Hub instance, populate the following variables:
# - Azurerm endpoint used for Azure Resource Manager operations
# - Audience for acquiring an OAuth token used to access Graph API
# - GUID of the directory tenant
$ArmEndpoint = $AzureStackInfo.TenantExternalEndpoints.TenantResourceManager
$GraphAudience = "https://graph." + $AzureStackInfo.ExternalDomainFQDN + "/"
$TenantID = $AzureStackInfo.AADTenantID

# Register and set an Azurerm environment that targets your Azure Stack Hub instance
Add-AzurermEnvironment -Name "AzureStackUser" -ArmEndpoint $ArmEndpoint

# Sign in using the new service principal identity
$spSignin = Connect-AzurermAccount -Environment "AzureStackUser" `

-ServicePrincipal `

-CertificateThumbprint $spObject.Thumbprint `

-ApplicationId $spObject.ClientId `

-TenantId $TenantID

# Output the service principal details
$spObject

```

2. After the script finishes, it displays the app registration info, including the service principal's credentials. As

demonstrated, the `ClientID` and `Thumbprint` are used to sign in under the service principal's identity. Upon successful sign-in, the service principal identity will be used for subsequent authorization and access to resources managed by Azure Resource Manager.

```
ApplicationIdentifier : S-1-5-21-1512385356-3796245103-1243299919-1356
ClientId           : 3c87e710-9f91-420b-b009-31fa9e430145
Thumbprint         : 30202C11BE6864437B64CE36C8D988442082A0F1
ApplicationName    : Azurestack-MyApp-c30febe7-1311-4fd8-9077-3d869db28342
ClientSecret       :
PSComputerName    : azs-ercs01
RunspaceId        : a78c76bb-8cae-4db4-a45a-c1420613e01b
```

Keep your PowerShell console session open, as you use it with the `ApplicationIdentifier` value in the next section.

Update a service principal's certificate credential

Now that you created a service principal, this section will show you how to:

1. Create a new self-signed X509 certificate for testing.
2. Update the service principal's credentials, by updating its `Thumbprint` property to match the new certificate.

Update the certificate credential using PowerShell, substituting your own values for the following placeholders:

PLACEHOLDER	DESCRIPTION	EXAMPLE
<PepVM>	The name of the privileged endpoint VM on your Azure Stack Hub instance.	"AzS-ERCS01"
<YourAppName>	A descriptive name for the new app registration.	"My management tool"
<YourCertificateLocation>	The location of your X509 certificate in the local certificate store.	"Cert:\CurrentUser\My\AB5A8A3533C C7AA2025BF05120117E06DE407B34"
<AppIdentifier>	The identifier assigned to the application registration.	"S-1-5-21-1512385356-3796245103-1243299919-1356"

1. Using your elevated Windows PowerShell session, run the following cmdlets:

```
# Create a PSSession to the PrivilegedEndpoint VM
$Session = New-PSSession -ComputerName "<PepVM>" -ConfigurationName PrivilegedEndpoint -Credential $creds

# Create a self-signed certificate for testing purposes.
$NewCert = New-SelfSignedCertificate -CertStoreLocation "cert:\CurrentUser\My" -Subject "CN=<YourAppName>" -KeySpec KeyExchange
# In production, use Get-Item and a managed certificate instead.
# $Cert = Get-Item "<YourCertificateLocation>

# Use the privileged endpoint to update the certificate thumbprint, used by the service principal associated with <AppIdentifier>
$SpObject = Invoke-Command -Session $Session -ScriptBlock {Set-GraphApplication -ApplicationIdentifier "<AppIdentifier>" -ClientCertificates $using:NewCert}
$Session | Remove-PSSession

# Output the updated service principal details
$SpObject
```

2. After the script finishes, it displays the updated app registration info, including the thumbprint value for the

new self-signed certificate.

```
ApplicationIdentifier : S-1-5-21-1512385356-3796245103-1243299919-1356
ClientId             :
Thumbprint          : AF22EE716909041055A01FE6C6F5C5CDE78948E9
ApplicationName     : Azurestack-MyApp-c30febe7-1311-4fd8-9077-3d869db28342
ClientSecret        :
PSCoputerName      : azs-ercs01
RunspaceId          : a580f894-8f9b-40ee-aa10-77d4d142b4e5
```

Create a service principal that uses client secret credentials

IMPORTANT

Using a client secret is less secure than using an X509 certificate credential. Not only is the authentication mechanism less secure, but it also typically requires embedding the secret in the client app source code. As such, for production apps, you're strongly encouraged to use a certificate credential.

Now you create another app registration, but this time specify a client secret credential. Unlike a certificate credential, the directory has the ability to generate a client secret credential. Instead of specifying the client secret, you use the `-GenerateClientSecret` switch to request that it be generated. Substitute your own values for the following placeholders:

PLACEHOLDER	DESCRIPTION	EXAMPLE
<PepVM>	The name of the privileged endpoint VM on your Azure Stack Hub instance.	"AzS-ERCS01"
<YourAppName>	A descriptive name for the new app registration.	"My management tool"

1. Open an elevated Windows PowerShell session, and run the following cmdlets:

```

# Sign in to PowerShell interactively, using credentials that have access to the VM running the
# Privileged Endpoint (typically <domain>\cloudadmin)
$creds = Get-Credential

# Create a PSSession to the Privileged Endpoint VM
$Session = New-PSSession -ComputerName "<PepVM>" -ConfigurationName PrivilegedEndpoint -Credential
$creds

# Use the privileged endpoint to create the new app registration (and service principal object)
$SpObject = Invoke-Command -Session $Session -ScriptBlock {New-GraphApplication -Name "<YourAppName>" -
GenerateClientSecret}
$AzureStackInfo = Invoke-Command -Session $Session -ScriptBlock {Get-AzureStackStampInformation}
$Session | Remove-PSSession

# Using the stamp info for your Azure Stack Hub instance, populate the following variables:
# - AzureRM endpoint used for Azure Resource Manager operations
# - Audience for acquiring an OAuth token used to access Graph API
# - GUID of the directory tenant
$ArmEndpoint = $AzureStackInfo.TenantExternalEndpoints.TenantResourceManager
$GraphAudience = "https://graph." + $AzureStackInfo.ExternalDomainFQDN + "/"
$TenantID = $AzureStackInfo.AADTenantID

# Register and set an AzureRM environment that targets your Azure Stack Hub instance
Add-AzureRMServer -Name "AzureStackUser" -ArmEndpoint $ArmEndpoint

# Sign in using the new service principal identity
$securePassword = $SpObject.ClientSecret | ConvertTo-SecureString -AsPlainText -Force
$credential = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList
$SpObject.ClientId, $securePassword
$SpSignin = Connect-AzureRmAccount -Environment "AzureStackUser" -ServicePrincipal -Credential
$credential -TenantId $TenantID

# Output the service principal details
$SpObject

```

- After the script finishes, it displays the app registration info, including the service principal's credentials. As demonstrated, the `ClientId` and generated `ClientSecret` are used to sign in under the service principal's identity. Upon successful sign-in, the service principal identity will be used for subsequent authorization and access to resources managed by Azure Resource Manager.

```

ApplicationIdentifier : S-1-5-21-1634563105-1224503876-2692824315-2623
ClientId           : 8e0ffd12-26c8-4178-a74b-f26bd28db601
Thumbprint         :
ApplicationName   : Azurestack-YourApp-6967581b-497e-4f5a-87b5-0c8d01a9f146
ClientSecret       : 6RUWLRoBw3EebBLgaWGiowCkoko5_j_ujIPjA8dS
PSCoputerName     : azs-ercs01
RunspaceId        : 286daaa1-c9a6-4176-a1a8-03f543f90998

```

Keep your PowerShell console session open, as you use it with the `ApplicationIdentifier` value in the next section.

Update a service principal's client secret

Update the client secret credential using PowerShell, using the `ResetClientSecret` parameter, which immediately changes the client secret. Substitute your own values for the following placeholders:

PLACEHOLDER	DESCRIPTION	EXAMPLE
<PepVM>	The name of the privileged endpoint VM on your Azure Stack Hub instance.	"AzS-ERCS01"

PLACEHOLDER	DESCRIPTION	EXAMPLE
<AppIdentifier>	The identifier assigned to the application registration.	"S-1-5-21-1634563105-1224503876-2692824315-2623"

- Using your elevated Windows PowerShell session, run the following cmdlets:

```
# Create a PSSession to the PrivilegedEndpoint VM
$Session = New-PSSession -ComputerName "<PepVM>" -ConfigurationName PrivilegedEndpoint -Credential $creds

# Use the privileged endpoint to update the client secret, used by the service principal associated with <AppIdentifier>
$SpObject = Invoke-Command -Session $Session -ScriptBlock {Set-GraphApplication -ApplicationIdentifier "<AppIdentifier>" -ResetClientSecret}
$Session | Remove-PSSession

# Output the updated service principal details
$SpObject
```

- After the script finishes, it displays the updated app registration info, including the newly generated client secret.

```
ApplicationIdentifier : S-1-5-21-1634563105-1224503876-2692824315-2623
ClientId           : 8e0ffd12-26c8-4178-a74b-f26bd28db601
Thumbprint         :
ApplicationName   : Azurestack-YourApp-6967581b-497e-4f5a-87b5-0c8d01a9f146
ClientSecret      : MKUNzeL6PwmlhWdHB59c25WDD1J1A6IWzwgv_Kn
PSComputerName    : azs-ercs01
RunspaceId        : 6ed9f903-f1be-44e3-9fef-e7e0e3f48564
```

Remove a service principal

Now you'll see how to remove/delete an app registration from your directory, and its associated service principal object, using PowerShell.

Substitute your own values for the following placeholders:

PLACEHOLDER	DESCRIPTION	EXAMPLE
<PepVM>	The name of the privileged endpoint VM on your Azure Stack Hub instance.	"AzS-ERCS01"
<AppIdentifier>	The identifier assigned to the application registration.	"S-1-5-21-1634563105-1224503876-2692824315-2623"

```

# Sign in to PowerShell interactively, using credentials that have access to the VM running the Privileged Endpoint (typically <domain>\clouddadmin)
$creds = Get-Credential

# Create a PSSession to the PrivilegedEndpoint VM
$session = New-PSSession -ComputerName "<PepVM>" -ConfigurationName PrivilegedEndpoint -Credential $creds

# OPTIONAL: Use the privileged endpoint to get a list of applications registered in AD FS
$appList = Invoke-Command -Session $session -ScriptBlock {Get-GraphApplication}

# Use the privileged endpoint to remove the application and associated service principal object for <AppIdentifier>
Invoke-Command -Session $session -ScriptBlock {Remove-GraphApplication -ApplicationIdentifier "<AppIdentifier>"}

```

There will be no output returned from calling the Remove-GraphApplication cmdlet on the privileged endpoint, but you'll see verbatim confirmation output to the console during execution of the cmdlet:

```

VERBOSE: Deleting graph application with identifier S-1-5-21-1634563105-1224503876-2692824315-2623.
VERBOSE: Remove-GraphApplication : BEGIN on AZS-ADFS01 on ADFSGraphEndpoint
VERBOSE: Application with identifier S-1-5-21-1634563105-1224503876-2692824315-2623 was deleted.
VERBOSE: Remove-GraphApplication : END on AZS-ADFS01 under ADFSGraphEndpoint configuration

```

Assign a role

Access to Azure resources by users and apps is authorized through Role-Based Access Control (RBAC). To allow an app to access resources in your subscription using its service principal, you must *assign* the service principal to a *role* for a specific *resource*. First decide which role represents the right *permissions* for the app. To learn about the available roles, see [Built-in roles for Azure resources](#).

The type of resource you choose also establishes the *access scope* for the service principal. You can set the access scope at the subscription, resource group, or resource level. Permissions are inherited to lower levels of scope. For example, adding an app to the "Reader" role for a resource group, means it can read the resource group and any resources it contains.

1. Sign in to the appropriate portal, based on the directory you specified during Azure Stack Hub installation (the Azure portal for Azure AD, or the Azure Stack Hub user portal for AD FS, for example). In this example, we show a user signed in to the Azure Stack Hub user portal.

NOTE

To add role assignments for a given resource, your user account must belong to a role that declares the `Microsoft.Authorization/roleAssignments/write` permission. For example, either the [Owner](#) or [User Access Administrator](#) built-in roles.

2. Navigate to the resource you wish to allow the service principal to access. In this example, assign the service principal to a role at the subscription scope, by selecting **Subscriptions**, then a specific subscription. You could instead select a resource group, or a specific resource like a virtual machine.

3. Select the **Access Control (IAM)** page, which is universal across all resources that support RBAC.
4. Select **+ Add**
5. Under **Role**, pick the role you wish to assign to the app.
6. Under **Select**, search for your app using a full or partial Application Name. During registration, the Application Name is generated as *Azurestack-<YourAppName>-<ClientId>*. For example, if you used an application name of *App2*, and ClientId *2bbe67d8-3fdb-4b62-87cf-cc41dd4344ff* was assigned during creation, the full name would be *Azurestack-App2-2bbe67d8-3fdb-4b62-87cf-cc41dd4344ff*. You can search for either the exact string, or a portion, like *Azurestack* or *Azurestack-App2*.
7. Once you find the app, select it and it will show under **Selected members**.
8. Select **Save** to finish assigning the role.

9. When finished, the app will show in the list of principals assigned for the current scope, for the given role.

The screenshot shows the 'Access control (IAM)' section of the Azure Stack portal. The left sidebar includes options like Overview, Settings, Add-on plans, Resource groups, Resources, My permissions, Resource providers, Deployments, Properties, and Resource locks. The 'Access control (IAM)' option is selected. The main area displays a table of role assignments:

NAME	TYPE	ROLE	SCOPE
AzureStack-DiskRP-Client AzureStack-DiskRP-Resource	App	Contributor	This resource
Subscription admins	Group	Owner	Subscription (Inherited)
Azurestack-App2-2bbe67d8-3fdb-4b6...	App	Reader	This resource

The last row, which contains the service principal for the application, is highlighted with a red box.

Now that you've created a service principal and assigned a role, you can begin using this service principal within your app to access Azure Stack Hub resources.

Next steps

[Add users for AD FS](#)

[Manage user permissions](#)

[Azure Active Directory Documentation](#)

[Active Directory Federation Services](#)

Configure multi-tenancy in Azure Stack Hub

4 minutes to read • [Edit Online](#)

You can configure Azure Stack Hub to support users from multiple Azure Active Directory (Azure AD) tenants, allowing them to use services in Azure Stack Hub. For example, consider the following scenario:

- You're the service administrator of contoso.onmicrosoft.com, where Azure Stack Hub is installed.
- Mary is the directory administrator of fabrikam.onmicrosoft.com, where guest users are located.
- Mary's company receives IaaS and PaaS services from your company and needs to allow users from the guest directory (fabrikam.onmicrosoft.com) to sign in and use Azure Stack Hub resources in contoso.onmicrosoft.com.

This guide provides the steps required, in the context of this scenario, to configure multi-tenancy in Azure Stack Hub. In this scenario, you and Mary must complete steps to enable users from Fabrikam to sign in and consume services from the Azure Stack Hub deployment in Contoso.

Enable multi-tenancy

There are a few prerequisites to account for before you configure multi-tenancy in Azure Stack Hub:

- You and Mary must coordinate administrative steps across both the directory Azure Stack Hub is installed in (Contoso), and the guest directory (Fabrikam).
- Make sure you've [installed](#) and [configured](#) PowerShell for Azure Stack Hub.
- [Download the Azure Stack Hub Tools](#), and import the Connect and Identity modules:

```
Import-Module .\Connect\AzureStack.Connect.psm1
Import-Module .\Identity\AzureStack.Identity.psm1
```

Configure Azure Stack Hub directory

In this section, you configure Azure Stack Hub to allow sign-ins from Fabrikam Azure AD directory tenants.

Onboard the guest directory tenant (Fabrikam) to Azure Stack Hub by configuring Azure Resource Manager to accept users and service principals from the guest directory tenant.

The service admin of contoso.onmicrosoft.com runs the following commands:

```

## The following Azure Resource Manager endpoint is for the ASDK. If you're in a multinode environment,
contact your operator or service provider to get the endpoint.
$adminARMEndpoint = "https://adminmanagement.local.azurestack.external"

## Replace the value below with the Azure Stack Hub directory
$azureStackDirectoryTenant = "contoso.onmicrosoft.com"

## Replace the value below with the guest tenant directory.
$guestDirectoryTenantToBeOnboarded = "fabrikam.onmicrosoft.com"

## Replace the value below with the name of the resource group in which the directory tenant registration
resource should be created (resource group must already exist).
$ResourceGroupName = "system.local"

## Replace the value below with the region location of the resource group.
$location = "local"

# Subscription Name
$SubscriptionName = "Default Provider Subscription"

Register-AzSGuestDirectoryTenant -AdminResourceManagerEndpoint $adminARMEndpoint ` 
    -DirectoryTenantName $azureStackDirectoryTenant ` 
    -GuestDirectoryTenantName $guestDirectoryTenantToBeOnboarded ` 
    -Location $location ` 
    -ResourceGroupName $ResourceGroupName ` 
    -SubscriptionName $SubscriptionName

```

Configure guest directory

Once the Azure Stack Hub operator has enabled the Fabrikam directory to be used with Azure Stack Hub, Mary must register Azure Stack Hub with Fabrikam's directory tenant.

Registering Azure Stack Hub with the guest directory

Mary (directory admin of Fabrikam) runs the following commands in the guest directory fabrikam.onmicrosoft.com:

```

## The following Azure Resource Manager endpoint is for the ASDK. If you're in a multinode environment,
contact your operator or service provider to get the endpoint.
$tenantARMEndpoint = "https://management.local.azurestack.external"

## Replace the value below with the guest tenant directory.
$guestDirectoryTenantName = "fabrikam.onmicrosoft.com"

Register-AzSWithMyDirectoryTenant ` 
    -TenantResourceManagerEndpoint $tenantARMEndpoint ` 
    -DirectoryTenantName $guestDirectoryTenantName ` 
    -Verbose

```

IMPORTANT

If your Azure Stack Hub admin installs new services or updates in the future, you may need to run this script again.

Run this script again at any time to check the status of the Azure Stack Hub apps in your directory.

If you've noticed issues with creating VMs in Managed Disks (introduced in the 1808 update), a new **Disk Resource Provider** was added requiring this script to be run again.

Direct users to sign in

Now that you and Mary have completed the steps to onboard Mary's directory, Mary can direct Fabrikam users to sign in. Fabrikam users (users with the fabrikam.onmicrosoft.com suffix) sign in by visiting <https://portal.local.azurestack.external>.

Mary will direct any [foreign principals](#) in the Fabrikam directory (users in the Fabrikam directory without the suffix of fabrikam.onmicrosoft.com) to sign in using <https://portal.local.azurestack.external/fabrikam.onmicrosoft.com>. If they don't use this URL, they're sent to their default directory (Fabrikam) and receive an error that says their admin hasn't consented.

Disable multi-tenancy

If you no longer want multiple tenants in Azure Stack Hub, you can disable multi-tenancy by doing the following steps in order:

- As the admin of the guest directory (Mary in this scenario), run *Unregister-AzsWithMyDirectoryTenant*. The cmdlet uninstalls all the Azure Stack Hub apps from the new directory.

```
## The following Azure Resource Manager endpoint is for the ASDK. If you're in a multinode environment, contact your operator or service provider to get the endpoint.  
$tenantARMEEndpoint = "https://management.local.azurestack.external"  
  
## Replace the value below with the guest tenant directory.  
$guestDirectoryTenantName = "fabrikam.onmicrosoft.com"  
  
Unregister-AzsWithMyDirectoryTenant `  
-TenantResourceManagerEndpoint $tenantARMEEndpoint `  
-DirectoryTenantName $guestDirectoryTenantName `  
-Verbose
```

- As the service admin of Azure Stack Hub (you in this scenario), run *Unregister-AzSGuestDirectoryTenant*.

```
## The following Azure Resource Manager endpoint is for the ASDK. If you're in a multinode environment, contact your operator or service provider to get the endpoint.  
$adminARMEEndpoint = "https://adminmanagement.local.azurestack.external"  
  
## Replace the value below with the Azure Stack Hub directory  
$azureStackDirectoryTenant = "contoso.onmicrosoft.com"  
  
## Replace the value below with the guest tenant directory.  
$guestDirectoryTenantToBeDecommissioned = "fabrikam.onmicrosoft.com"  
  
## Replace the value below with the name of the resource group in which the directory tenant registration resource should be created (resource group must already exist).  
$ResourceGroupName = "system.local"  
  
Unregister-AzSGuestDirectoryTenant -AdminResourceManagerEndpoint $adminARMEEndpoint `  
-DirectoryTenantName $azureStackDirectoryTenant `  
-GuestDirectoryTenantName $guestDirectoryTenantToBeDecommissioned `  
-ResourceGroupName $ResourceGroupName
```

WARNING

The disable multi-tenancy steps must be performed in order. Step #1 fails if step #2 is completed first.

Next steps

- [Manage delegated providers](#)
- [Azure Stack Hub key concepts](#)
- [Manage usage and billing for Azure Stack Hub as a Cloud Solution Provider](#)
- [Add tenant for usage and billing to Azure Stack Hub](#)

Data at rest encryption in Azure Stack Hub

2 minutes to read • [Edit Online](#)

Azure Stack Hub protects user and infrastructure data at the storage subsystem level using encryption at rest. Azure Stack Hub's storage subsystem is encrypted using BitLocker with 128-bit AES encryption. BitLocker keys are persisted in an internal secret store.

Data at rest encryption is a common requirement for many of the major compliance standards (for example, PCI-DSS, FedRAMP, HIPAA). Azure Stack Hub enables you to meet those requirements with no extra work or configurations required. For more information on how Azure Stack Hub helps you meet compliance standards, see the [Microsoft Service Trust Portal](#).

NOTE

Data at rest encryption protects your data against being accessed by someone who physically stole one or more hard drives. Data at rest encryption doesn't protect against data being intercepted over the network (data in transit), data currently being used (data in memory), or, more in general, data being exfiltrated while the system is up and running.

Retrieving BitLocker recovery keys

Azure Stack Hub BitLocker keys for data at rest are internally managed. You aren't required to provide them for regular operations or during system startup. However, support scenarios may require BitLocker recovery keys to bring the system online.

WARNING

Retrieve your BitLocker recovery keys and store them in a secure location outside of Azure Stack Hub. Not having the recovery keys during certain support scenarios may result in data loss and require a system restore from a backup image.

Retrieving the BitLocker recovery keys requires access to the [privileged endpoint \(PEP\)](#). From a PEP session, run the Get-AzsRecoveryKeys cmdlet.

```
##This cmdlet retrieves the recovery keys for all the volumes that are encrypted with BitLocker.  
Get-AzsRecoveryKeys
```

Optional parameters for `Get-AzsRecoveryKeys` cmdlet:

PARAMETER	DESCRIPTION	TYPE	REQUIRED
<code>raw</code>	Returns raw data of mapping between recovery key, computer name, and password id(s) of each encrypted volume.	Switch	No (designed for support scenarios)

Troubleshoot issues

In extreme circumstances, a BitLocker unlock request could fail resulting in a specific volume to not boot. Depending on the availability of some of the components of the architecture, this failure could result in downtime

and potential data loss if you don't have your BitLocker recovery keys.

WARNING

Retrieve your BitLocker recovery keys and store them in a secure location outside of Azure Stack Hub. Not having the recovery keys during certain support scenarios may result in data loss and require a system restore from a backup image.

If you suspect your system is experiencing issues with BitLocker, such as Azure Stack Hub failing to start, contact support. Support requires your BitLocker recovery keys. The majority of the BitLocker related issues can be resolved with a FRU operation for that specific VM/host/volume. For the other cases, a manual unlocking procedure using BitLocker recovery keys can be done. If BitLocker recovery keys aren't available, the only option is to restore from a backup image. Depending on when the last backup was done, you may experience data loss.

Next steps

- [Learn more about Azure Stack Hub security.](#)
- For more information on how BitLocker protects CSVs, see [protecting cluster shared volumes and storage area networks with BitLocker](#).

Rotate secrets in Azure Stack Hub

13 minutes to read • [Edit Online](#)

These instructions apply only to Azure Stack Hub Integrated Systems version 1803 and Later. Don't attempt secret rotation on pre-1802 Azure Stack Hub Versions

Secrets help you maintain secure communication between the Azure Stack Hub infrastructure resources and services.

Rotate secrets overview

1. Prepare the certificates which will be used for secret rotation.
2. Review the Azure Stack Hub [public key infrastructure certificate requirements](#).
3. [Use the privileged endpoint](#) and run **Test-azurestack** to confirm that everything is fine.
4. Review the [pre-steps for secret rotation](#).
5. [Validate Azure Stack Hub PKI certificates](#). Make sure there are no special characters in the password, like `*` or `)`.
6. Make sure the PFX encryption is **TripleDES-SHA1**. If you run into an issue, see [Fix common issues with Azure Stack Hub PKI certificates](#).
7. Prepare the folder structure. You can find an example in the [Rotating external secrets](#) section.
8. [Start the secret rotation](#).

Rotate secrets

Azure Stack Hub uses various secrets to maintain secure communication between the Azure Stack Hub infrastructure resources and services.

- **Internal secrets**

All the certificates, passwords, secure strings, and keys used by the Azure Stack Hub infrastructure without intervention of the Azure Stack Hub Operator.

- **External secrets**

Infrastructure service certificates for external-facing services that are provided by the Azure Stack Hub Operator. External secrets include the certificates for the following services:

- Administrator portal
- Public portal
- Administrator Azure Resource Manager
- Global Azure Resource Manager
- Administrator Key Vault
- Key Vault
- Admin Extension Host
- ACS (including blob, table, and queue storage)
- ADFS*
- Graph*

* Only applicable if the environment's identity provider is Active Directory Federated Services (AD FS).

NOTE

All other secure keys and strings, including BMC and switch passwords as well as user and administrator account passwords are still manually updated by the administrator.

IMPORTANT

Starting with Azure Stack Hub's 1811 release, secret rotation has been separated for internal and external certificates.

To maintain the integrity of the Azure Stack Hub infrastructure, operators need the ability to periodically rotate their infrastructure's secrets at frequencies that are consistent with their organization's security requirements.

Rotating Secrets with external certificates from a new Certificate Authority

Azure Stack Hub supports secret rotation with external certificates from a new Certificate Authority (CA) in the following contexts:

INSTALLED CERTIFICATE CA	CA TO ROTATE TO	SUPPORTED	AZURE STACK HUB VERSIONS SUPPORTED
From Self-Signed	To Enterprise	Supported	1903 & Later
From Self-Signed	To Self-Signed	Not Supported	
From Self-Signed	To Public*	Supported	1803 & Later
From Enterprise	To Enterprise	Supported. From 1803-1903: supported so long as customers use the SAME enterprise CA as used at deployment	1803 & Later
From Enterprise	To Self-Signed	Not Supported	
From Enterprise	To Public*	Supported	1803 & Later
From Public*	To Enterprise	Supported	1903 & Later
From Public*	To Self-Signed	Not Supported	
From Public*	To Public*	Supported	1803 & Later

*Indicates that the Public Certificate Authorities are those that are part of the Windows Trusted Root Program.

You can find the full list in the article [Microsoft Trusted Root Certificate Program: Participants \(as of June 27, 2017\)](#).

Fixing alerts

When secrets are within 30 days of expiration, the following alerts are generated in the administrator portal:

- Pending service account password expiration
- Pending internal certificate expiration
- Pending external certificate expiration

Running secret rotation using the instructions below will fix these alerts.

NOTE

Azure Stack Hub environments on pre-1811 versions may see alerts for pending internal certificate or secret expirations. These alerts are inaccurate and should be ignored without running internal secret rotation. Inaccurate internal secret expiration alerts are a known issue that's resolved in 1811. Internal secrets won't expire unless the environment has been active for two years.

Pre-steps for secret rotation

IMPORTANT

If secret rotation has already been performed on your Azure Stack Hub environment then you must update the system to version 1811 or later before you execute secret rotation again. Secret Rotation must be executed via the [Privileged Endpoint](#) and requires Azure Stack Hub Operator credentials. If your environment Azure Stack Hub Operator(s) don't know whether secret rotation has been run on your environment, update to 1811 before executing secret rotation again.

1. It's highly recommended you update your Azure Stack Hub instance to version 1811.

NOTE

For pre-1811 versions, you don't need to rotate secrets to add extension host certificates. You should follow the instructions in the article [Prepare for extension host for Azure Stack Hub](#) to add extension host certificates.

2. Operators may notice alerts open and automatically close during rotation of Azure Stack Hub secrets. This behavior is expected and the alerts can be ignored. Operators can verify the validity of these alerts by running [Test-AzureStack](#). For operators using System Center Operations Manager to monitor Azure Stack Hub systems, placing a system in maintenance mode will prevent these alerts from reaching their ITSM systems but will continue to alert if the Azure Stack Hub system becomes unreachable.
3. Notify your users of any maintenance operations. Schedule normal maintenance windows, as much as possible, during non-business hours. Maintenance operations may affect both user workloads and portal operations.

NOTE

The next steps only apply when rotating Azure Stack Hub external secrets.

4. Run [Test-AzureStack](#) and confirm all test outputs are healthy before rotating secrets.
5. Prepare a new set of replacement external certificates. The new set matches the certificate specifications outlined in the [Azure Stack Hub PKI certificate requirements](#). You can generate a certificate signing request (CSR) for purchasing or creating new certificates using the steps outlined in [Generate PKI Certificates](#) and prepare them for use in your Azure Stack Hub environment using the steps in [Prepare Azure Stack Hub PKI Certificates](#). Be sure to validate the certificates you prepare with the steps outlined in [Validate PKI Certificates](#).
6. Store a backup to the certificates used for rotation in a secure backup location. If your rotation runs and then fails, replace the certificates in the file share with the backup copies before you rerun the rotation. Keep backup copies in the secure backup location.
7. Create a fileshare you can access from the ERCS VMs. The file share must be readable and writable for the [CloudAdmin](#) identity.

8. Open a PowerShell ISE console from a computer where you have access to the fileshare. Navigate to your fileshare.
9. Run **CertDirectoryMaker.ps1** to create the required directories for your external certificates.

IMPORTANT

The CertDirectoryMaker script will create a folder structure that will adhere to:

.\Certificates\AAD or .\Certificates\ADFS depending on your Identity Provider used for Azure Stack Hub.

It's of utmost importance that your folder structure ends with **AAD** or **ADFS** folders and all subdirectories are within this structure; otherwise, **Start-SecretRotation** will come up with:

```
Cannot bind argument to parameter 'Path' because it is null.  
+ CategoryInfo          : InvalidData: (:) [Test-Certificate], ParameterBindingValidationException  
+ FullyQualifiedErrorId : ParameterArgumentValidationErrorNullNotAllowed,Test-Certificate  
+ PSComputerName        : xxx.xxx.xxx.xxx
```

The error message indicates that there's a problem accessing your fileshare but in reality it's the folder structure that's being enforced here. More information can be found in the Microsoft AzureStack Readiness Checker - [PublicCertHelper module](#).

It's also important that your fileshare folder structure begins with **Certificates** folder, otherwise it will also fail on validation. Fileshare mount should look like \\<IPAddress>\<ShareName>\ and it should contain folder **Certificates\AAD** or **Certificates\ADFS** inside.

For example:

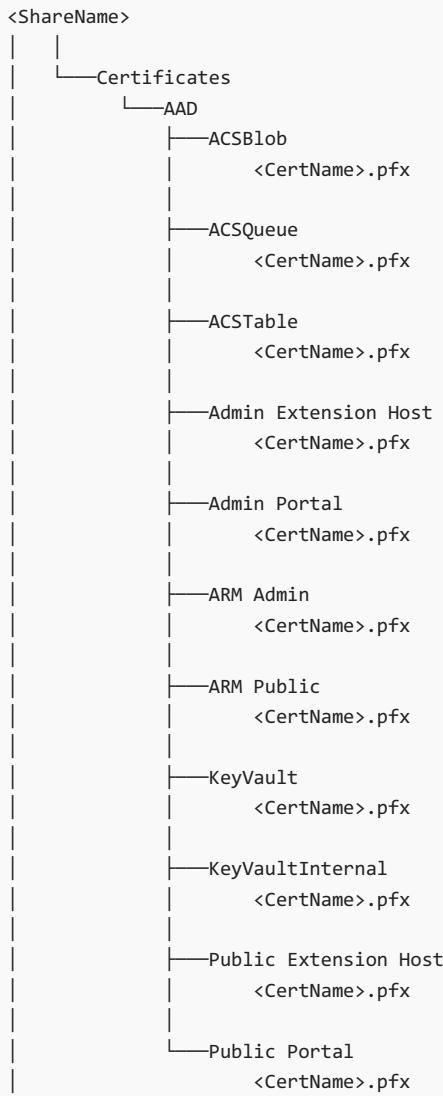
- Fileshare = \\<IPAddress>\<ShareName>\
- CertFolder = **Certificates\AAD**
- FullPath = \\<IPAddress>\<ShareName>\Certificates\AAD

Rotating external secrets

To rotate external secrets:

1. Within the newly created **\Certificates\<IdentityProvider>** directory created in the pre-steps, place the new set of replacement external certificates in the directory structure according to the format outlined in the **Mandatory certificates** section of the [Azure Stack Hub PKI certificate requirements](#).

Example of folder structure for the Azure AD Identity Provider:



2. Create a PowerShell Session with the [Privileged endpoint](#) using the **CloudAdmin** account and store the sessions as a variable. You'll use this variable as the parameter in the next step.

IMPORTANT

Don't enter the session. Store the session as a variable.

3. Run **Invoke-Command**. Pass your privileged endpoint PowerShell session variable as the **Session** parameter.

4. Run **Start-SecretRotation** with the following parameters:

- **PfxFilePath**

Specify the network path to your Certificates directory created earlier.

- **PathAccessCredential**

A PSCredential object for credentials to the share.

- **CertificatePassword**

A secure string of the password used for all of the pfx certificate files created.

5. Wait while your secrets rotate. External secret rotation takes approximately one hour.

When secret rotation successfully completes, your console will display **Overall action status: Success**.

NOTE

If secret rotation fails, follow the instructions in the error message and re-run **Start-SecretRotation** with the **-ReRun** parameter.

```
Start-SecretRotation -ReRun
```

Contact support if you experience repeated secret rotation failures.

6. After successful completion of secret rotation, remove your certificates from the share created in the pre-step and store them in their secure backup location.

Use PowerShell to rotate secrets

The following PowerShell example demonstrates the cmdlets and parameters to run in order to rotate your secrets.

```
# Create a PEP Session
winrm s winrm/config/client '@{TrustedHosts= "<IpOfERCSMachine>"}'
$PEPCreds = Get-Credential
$PEPSession = New-PSSession -ComputerName <IpOfERCSMachine> -Credential $PEPCreds -ConfigurationName
"PrivilegedEndpoint"

# Run Secret Rotation
$CertPassword = ConvertTo-SecureString "<CertPasswordHere>" -AsPlainText -Force
$CertShareCreds = Get-Credential
$CertSharePath = "<NetworkPathOfCertShare>"
Invoke-Command -Session $PEPSession -ScriptBlock {
    Start-SecretRotation -PfxFilePath $using:CertSharePath -PathAccessCredential $using:CertShareCreds -
    CertificatePassword $using:CertPassword
}
Remove-PSSession -Session $PEPSession
```

Rotating only internal secrets

NOTE

Internal secret rotation should only be done if you suspect an internal secret has been compromised by a malicious entity, or if you've received an alert (on build 1811 or later) indicating internal certificates are nearing expiration. Azure Stack Hub environments on pre-1811 versions may see alerts for pending internal certificate or secret expirations. These alerts are inaccurate and should be ignored without running internal secret rotation. Inaccurate internal secret expiration alerts are a known issue that's resolved in 1811. Internal secrets won't expire unless the environment has been active for two years.

1. Create a PowerShell session with the [Privileged endpoint](#).
2. In the Privileged Endpoint session, run **Start-SecretRotation -Internal**.

NOTE

Azure Stack Hub environments on pre-1811 versions won't require the **-Internal** flag. **Start-SecretRotation** will rotate only internal secrets.

3. Wait while your secrets rotate.

When secret rotation successfully completes, your console will display **Overall action status: Success**.

NOTE

If secret rotation fails, follow the instructions in the error message and rerun **Start-SecretRotation** with the **-Internal** and **-ReRun** parameters.

```
Start-SecretRotation -Internal -ReRun
```

Contact support if you experience repeated secret rotation failures.

Start-SecretRotation reference

Rotates the secrets of an Azure Stack Hub System. Only executed against the Azure Stack Hub privileged endpoint.

Syntax

For external secret rotation

```
Start-SecretRotation [-PfxFilePath <string>] [-PathAccessCredential <PSCredential>] [-CertificatePassword <SecureString>]
```

For internal secret rotation

```
Start-SecretRotation [-Internal]
```

For external secret rotation rerun

```
Start-SecretRotation [-ReRun]
```

For internal secret rotation rerun

```
Start-SecretRotation [-ReRun] [-Internal]
```

Description

The **Start-SecretRotation** cmdlet rotates the infrastructure secrets of an Azure Stack Hub system. By default, it rotates only the certificates of all external network infrastructure endpoints. If used with the **-Internal** flag, internal infrastructure secrets will be rotated. When rotating external network infrastructure endpoints, **Start-SecretRotation** should be run with an **Invoke-Command** script block with the Azure Stack Hub environment's privileged endpoint session passed in as the **Session** parameter.

Parameters

PARAMETER	TYPE	REQUIRED	POSITION	DEFAULT	DESCRIPTION
-----------	------	----------	----------	---------	-------------

Parameter	Type	Required	Position	Default	Description
PfxFilePath	String	False	Named	None	The fileshare path to the \Certificates directory containing all external network endpoint certificates. Only required when rotating external secrets. End directory must be \Certificates .
CertificatePassword	SecureString	False	Named	None	The password for all certificates provided in the -PfxFilePath. Required value if PfxFilePath is provided when external secrets are rotated.
Internal	String	False	Named	None	Internal flag must be used anytime an Azure Stack Hub operator wishes to rotate internal infrastructure secrets.
PathAccessCredential	PSCredential	False	Named	None	The PowerShell credential for the fileshare of the \Certificates directory containing all external network endpoint certificates. Only required when rotating external secrets.
ReRun	SwitchParameter	False	Named	None	ReRun must be used anytime secret rotation is reattempted after a failed attempt.

Examples

Rotate only internal infrastructure secrets

This command must be run via your Azure Stack Hub [environment's privileged endpoint](#).

```
PS C:\> Start-SecretRotation -Internal
```

This command rotates all of the infrastructure secrets exposed to the Azure Stack Hub internal network.

Rotate only external infrastructure secrets

```
# Create a PEP Session
winrm s winrm/config/client '@{TrustedHosts= "<IpOfERCSMachine>"}'
$PEPCreds = Get-Credential
$PEPSession = New-PSSession -ComputerName <IpOfERCSMachine> -Credential $PEPCreds -ConfigurationName
"PrivilegedEndpoint"

# Create Credentials for the fileshare
$CertPassword = ConvertTo-SecureString "<CertPasswordHere>" -AsPlainText -Force
$CertShareCreds = Get-Credential
$CertSharePath = "<NetworkPathOfCertShare>"
# Run Secret Rotation
Invoke-Command -Session $PEPSession -ScriptBlock {
    Start-SecretRotation -PfxFilePath $using:CertSharePath -PathAccessCredential $using:CertShareCreds -
    CertificatePassword $using:CertPassword
}
Remove-PSSession -Session $PEPSession
```

This command rotates the TLS certificates used for Azure Stack Hub's external network infrastructure endpoints.

Rotate internal and external infrastructure secrets (pre-1811 only)

IMPORTANT

This command only applies to Azure Stack Hub **pre-1811** as the rotation has been split for internal and external certificates.

From 1811+ you can't rotate both internal and external certificates anymore!

```
# Create a PEP Session
winrm s winrm/config/client '@{TrustedHosts= "<IpOfERCSMachine>"}'
$PEPCreds = Get-Credential
$PEPSession = New-PSSession -ComputerName <IpOfERCSMachine> -Credential $PEPCreds -ConfigurationName
"PrivilegedEndpoint"

# Create Credentials for the fileshare
$CertPassword = ConvertTo-SecureString "<CertPasswordHere>" -AsPlainText -Force
$CertShareCreds = Get-Credential
$CertSharePath = "<NetworkPathOfCertShare>"
# Run Secret Rotation
Invoke-Command -Session $PEPSession -ScriptBlock {
    Start-SecretRotation -PfxFilePath $using:CertSharePath -PathAccessCredential $using:CertShareCreds -
    CertificatePassword $using:CertPassword
}
Remove-PSSession -Session $PEPSession
```

This command rotates all of the infrastructure secrets exposed to the Azure Stack Hub internal network as well as the TLS certificates used for Azure Stack Hub's external network infrastructure endpoints. Start-SecretRotation rotates all stack-generated secrets, and because there are provided certificates, external endpoint certificates will also be rotated.

Update the baseboard management controller (BMC) credential

The baseboard management controller (BMC) monitors the physical state of your servers. Refer to your original equipment manufacturer (OEM) hardware vendor for instructions to update the user account name and

password of the BMC.

NOTE

Your OEM may provide additional management apps. Updating the user name or password for other management apps has no affect on the BMC user name or password.

1. **Versions earlier than 1910:** Update the BMC on the Azure Stack Hub physical servers by following your OEM instructions. The user name and password for each BMC in your environment must be the same. The BMC user names can't exceed 16 characters.

Version 1910 and later: It's no longer required that you first update the BMC credentials on the Azure Stack Hub physical servers by following your OEM instructions. The user name and password for each BMC in your environment must be the same. The BMC user names can't exceed 16 characters.

PARAMETER	DESCRIPTION	STATE
BypassBMCUpdate	When you use the parameter, credentials in the BMC aren't update. Only the Azure Stack Hub internal datastore is updated.	Optional

2. Open a privileged endpoint in Azure Stack Hub sessions. For instructions, see [Using the privileged endpoint in Azure Stack Hub](#).
3. After your PowerShell prompt has changed to **[IP address or ERCS VM name]: PS>** or to **[azs-ercs01]: PS>**, depending on the environment, run `Set-BmcCredential` by running `Invoke-Command`. Pass your privileged endpoint session variable as a parameter. For example:

```
# Interactive Version
$PEPIP = "<Privileged Endpoint IP or Name>" # You can also use the machine name instead of IP here.
$PEPCreds = Get-Credential "<Domain>\CloudAdmin" -Message "PEP Credentials"
$NewBmcPwd = Read-Host -Prompt "Enter New BMC password" -AsSecureString
$NewBmcUser = Read-Host -Prompt "Enter New BMC user name"

$PEPSession = New-PSSession -ComputerName $PEPIP -Credential $PEPCreds -ConfigurationName
"PrivilegedEndpoint"

Invoke-Command -Session $PEPSession -ScriptBlock {
    # Parameter BmcPassword is mandatory, while the BmcUser parameter is optional.
    Set-BmcCredential -BmcPassword $using:NewBmcPwd -BmcUser $using:NewBmcUser
}
Remove-PSSession -Session $PEPSession
```

You can also use the static PowerShell version with the Passwords as code lines:

```
# Static Version
$PEPIp = "<Privileged Endpoint IP or Name>" # You can also use the machine name instead of IP here.
$PEPUser = "<Privileged Endpoint user for example Domain\CloudAdmin>"
$PEPPwd = ConvertTo-SecureString "<Privileged Endpoint Password>" -AsPlainText -Force
$PEPCreds = New-Object System.Management.Automation.PSCredential ($PEPUser, $PEPPwd)
$NewBmcPwd = ConvertTo-SecureString "<New BMC Password>" -AsPlainText -Force
$NewBmcUser = "<New BMC User name>

$PEPSession = New-PSSession -ComputerName $PEPIp -Credential $PEPCreds -ConfigurationName
"PrivilegedEndpoint"

Invoke-Command -Session $PEPSession -ScriptBlock {
    # Parameter BmcPassword is mandatory, while the BmcUser parameter is optional.
    Set-BmcCredential -BmcPassword $using:NewBmcPwd -BmcUser $using:NewBmcUser
}
Remove-PSSession -Session $PEPSession
```

Next steps

[Learn more about Azure Stack Hub security](#)

Update Windows Defender Antivirus on Azure Stack Hub

3 minutes to read • [Edit Online](#)

Windows Defender Antivirus is an antimalware solution that provides security and virus protection. Every Azure Stack Hub infrastructure component (Hyper-V hosts and virtual machines) is protected with Windows Defender Antivirus. For up-to-date protection, you need periodic updates to Windows Defender Antivirus definitions, engine, and platform. How updates are applied depends on your configuration.

Connected scenario

The Azure Stack Hub [update resource provider](#) downloads antimalware definitions and engine updates multiple times per day. Each Azure Stack Hub infrastructure component gets the update from the update resource provider and applies the update automatically.

For those Azure Stack Hub deployments that are connected to the public Internet, apply the [monthly Azure Stack Hub update](#). The monthly Azure Stack Hub update includes Windows Defender Antivirus platform updates for the month.

Disconnected scenario

For those Azure Stack Hub deployments that are not connected to the public Internet (e.g. air-gapped datacenters), starting with the 1910 release, customers have the ability to apply the antimalware definitions and engine updates as they are published.

To apply the updates to your Azure Stack Hub solution, you first have to download them from the Microsoft site (links below) and subsequently, import them into a storage blob container under your *updateadminaccount*. A scheduled task scans the blob container every 30 minutes and, if new Defender definitions and engine updates are found, it applies them to the Azure Stack Hub infrastructure.

For those disconnected deployments that are not yet on 1910 or later, or that don't have the ability to download Defender definitions and engine updates on a daily basis, the monthly Azure Stack Hub update includes Windows Defender Antivirus definitions, engine, and platform updates for the month.

Set up Windows Defender for manual updates

With the 1910 release, two new cmdlets were added to the privileged endpoint to configure Windows Defender manual update in Azure Stack Hub.

```
### cmdlet to configure the storage blob container for the Defender updates
Set-AzsDefenderManualUpdate [-Container <string>] [-Remove]
### cmdlet to retrieve the configuration of the Defender manual update settings
Get-AzsDefenderManualUpdate
```

The following procedure shows how to setup Windows Defender manual update.

1. Connect to the privileged endpoint and run the following cmdlet to specify the name of the storage blob container where the Defender updates will be uploaded.

NOTE

The manual update process described below only works in disconnected environments where access to "go.microsoft.com" is not allowed. Trying to run the cmdlet Set-AzsDefenderManualUpdate in connected environments will result in an error.

```
### Configure the storage blob container for the Defender updates  
Set-AzsDefenderManualUpdate -Container <yourContainerName>
```

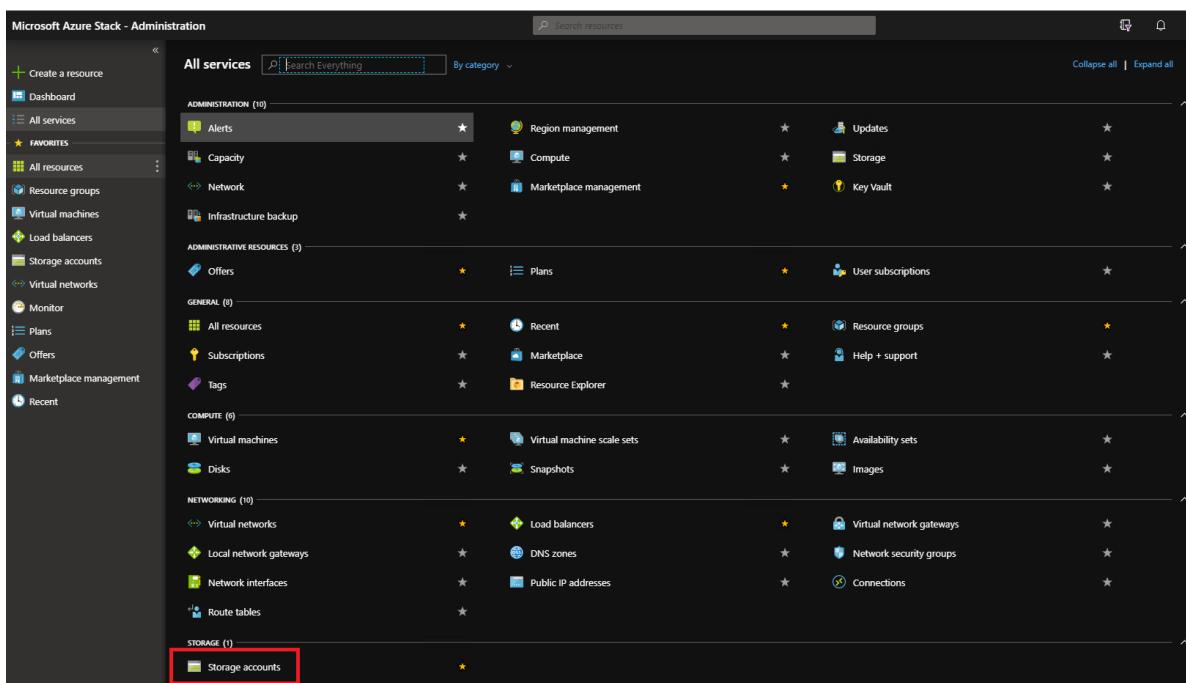
2. Download the two Windows Defender update packages and save them on a location that is reachable from your Azure Stack Hub administration portal.

- mpam-fe.exe from <https://go.microsoft.com/fwlink/?LinkId=121721&arch=x64>
- nis_full.exe from <https://go.microsoft.com/fwlink/?LinkId=197094>

NOTE

You'll have to download these two files **every time** you want to update the Defender signatures.

3. In the administration portal, select **All services**. Then, under the **DATA + STORAGE** category, select **Storage accounts**. (Or, in the filter box, start typing **storage accounts**, and select it.)

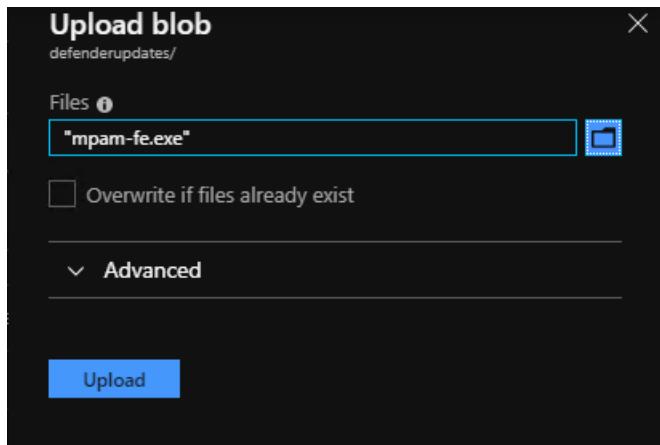


4. In the filter box, type **update**, and select the **updateadminaccount** storage account.
5. In the storage account details, under **Services**, select **Blobs**.

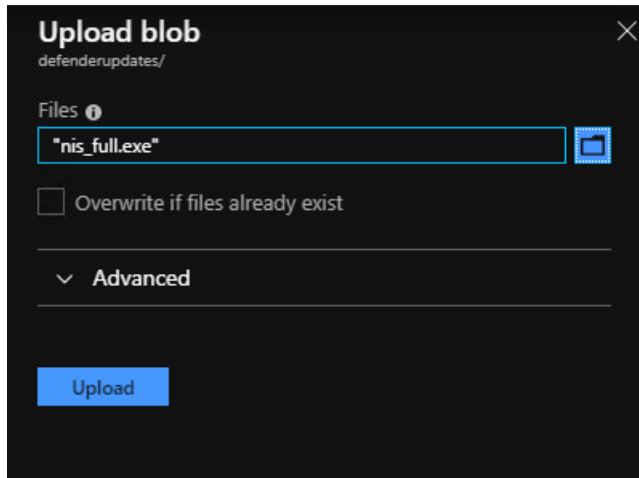
- Under **Blob service**, select **+ Container** to create a container. Enter the name that was specified with the Set-AzsDefenderManualUpdate (in this example *defenderupdates*) and then select **OK**.

- After the container is created, click the container name, and then click **Upload** to upload the package files to the container.

- Under **Upload blob**, click the folder icon, browse to the Windows Defender update *mpam-fe.exe* files and then click **Open** in the file explorer window.
- Under **Upload blob**, click **Upload**.



10. Repeat steps 8 and 9 for the *nis_full.exe* file.



A scheduled task scans the blob container every 30 minutes and applies any new Windows Defender package.

Next steps

[Learn more about Azure Stack Hub security](#)

Azure Stack Hub log and customer data handling

2 minutes to read • [Edit Online](#)

To the extent Microsoft is a processor or subprocessor of personal data in connection with Azure Stack Hub, Microsoft makes to all customers, effective May 25, 2018, the following commitments:

- The "Processing of Personal Data; GDPR" provision in the "Data Protection Terms" section of the [Online Services Terms](#).
- The European Union General Data Protection Regulation Terms in Attachment 4 of the [Online Services Terms](#).

As Azure Stack Hub resides in customer datacenters, Microsoft is the Data Controller solely of the data that is shared with Microsoft through [Diagnostics](#), [Telemetry](#), and [Billing](#).

Data access controls

Microsoft employees, who are assigned to investigate a specific support case, will be granted read-only access to the encrypted data. Microsoft employees also have access to tools used to delete the data if needed. All access to the customer data is audited and logged.

Data access controls:

- Data is only kept for a maximum of 90 days after case close.
- The customer always has the choice to have the data removed at any time in that 90-day period.
- Microsoft employees are given access to the data on a case-by-case basis and only as needed to help resolve the support issue.
- In the event where Microsoft must share customer data with OEM partners, customer consent is mandatory.

What Data Subject Requests (DSR) controls do customers have?

Microsoft supports on-demand data deletion per customer request. Customers can request that one of our support engineers delete all their logs for a given case at any time, before the data is permanently erased.

Does Microsoft notify customers when the data is deleted?

For the automated data deletion action (90 days after case close), we don't proactively contact customers and notify them about the deletion.

For the on-demand data deletion action, Microsoft support engineers have access to the tool that lets them delete data on demand. They can provide confirmation on the phone with the customer when it's done.

Diagnostic data

As part of the support process, Azure Stack Hub Operators can [share diagnostic logs](#) with Azure Stack Hub support and engineering teams to help with troubleshooting.

Microsoft provides a tool and script for customers to collect and upload requested diagnostic log files. Once collected, the log files are transferred over an HTTPS protected encrypted connection to Microsoft. Because HTTPS provides the encryption over the wire, there's no password needed for the encryption in transit. After they're received, logs are encrypted and stored until they're automatically deleted 90 days after the support case is closed.

Telemetry data

[Azure Stack Hub telemetry](#) automatically uploads system data to Microsoft via the Connected User Experience. Azure Stack Hub Operators have controls to customize telemetry features and privacy settings at any time.

Microsoft doesn't intend to gather sensitive data, such as credit card numbers, usernames and passwords, email addresses, and so on. If we determine that sensitive information has been inadvertently received, we delete it.

Billing data

[Azure Stack Hub Billing](#) leverages global Azure's Billing and Usage pipeline and is therefore in alignment with Microsoft compliance guidelines.

Azure Stack Hub Operators can configure Azure Stack Hub to forward usage information to Azure for billing. This configuration is required for Azure Stack Hub integrated systems customers who choose the pay-as-you-use billing model. Usage reporting is controlled independently from telemetry and isn't required for integrated systems customers who choose the capacity model or for Azure Stack Development Kit users. For these scenarios, usage reporting can be turned off using [the registration script](#).

Next steps

[Learn more about Azure Stack Hub security](#)

Azure Stack Hub Marketplace overview

2 minutes to read • [Edit Online](#)

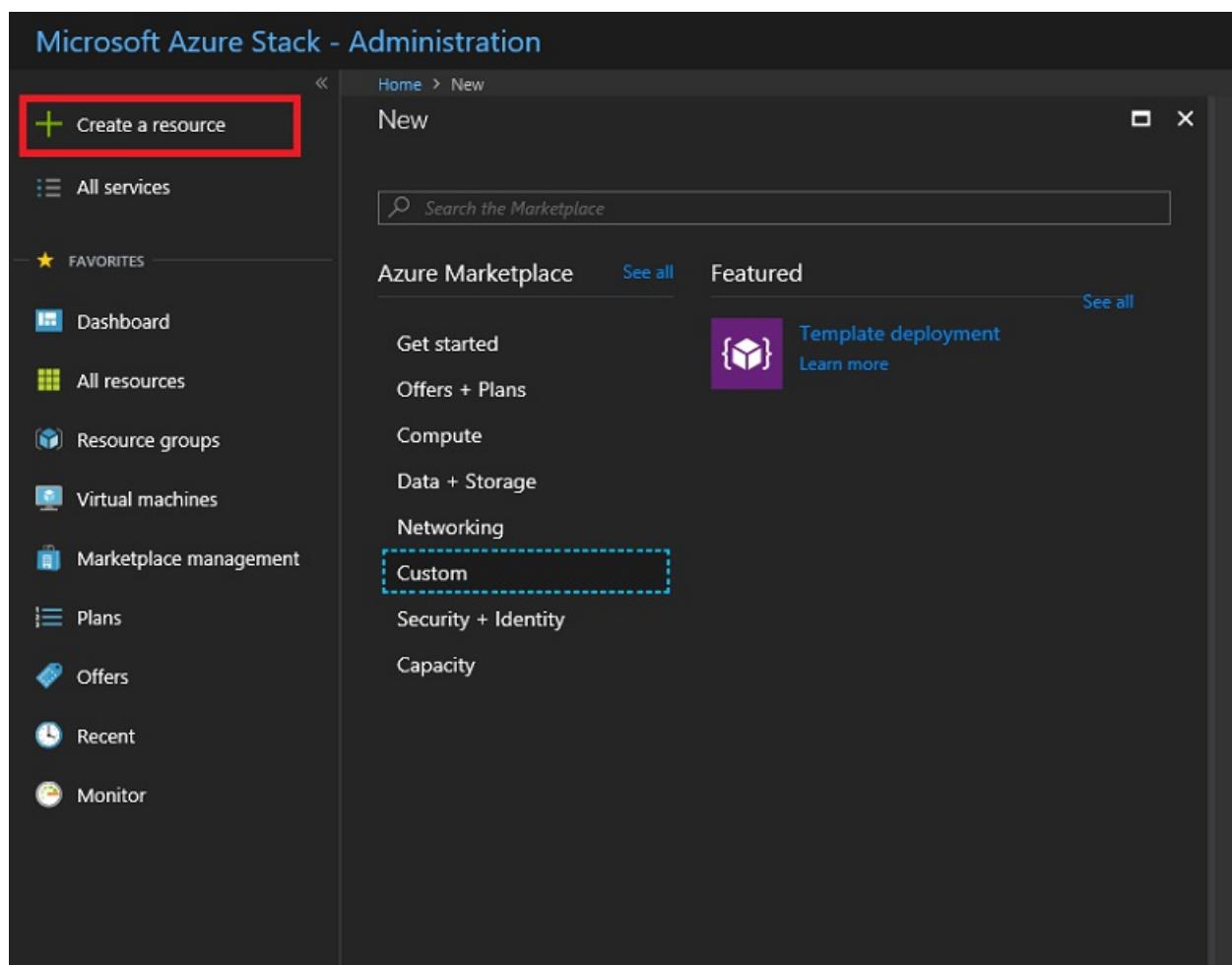
Azure Stack Hub Marketplace is a collection of services, apps, and resources customized for Azure Stack Hub. Resources include networks, virtual machines (VMs), storage, and more. Use Azure Stack Hub Marketplace to create new resources and deploy new apps or browse and choose the items you want to use. To use a marketplace item, users must subscribe to an offer that grants them access to the item.

As an Azure Stack Hub operator, you decide which items to add (publish) to Azure Stack Hub Marketplace. You can publish items such as databases, app services, and more. Publishing makes items visible to all your users. You can publish custom items that you create, or you can publish items from a growing [list of Azure Marketplace items](#). When you publish an item to Azure Stack Hub Marketplace, users can see it within five minutes.

Caution

All gallery item artifacts, including images and JSON files, are accessible without authentication after making them available in the Azure Stack Hub Marketplace. For more considerations when publishing custom marketplace items, see [Create and publish a Marketplace item](#).

To open the Marketplace, in the administrator portal select **+ Create a resource**.



Marketplace items

An Azure Stack Hub Marketplace item is a service, app, or resource that your users can download and use. All Azure Stack Hub Marketplace items are visible to all your users, including administrative items such as plans and offers. These administrative items don't require a subscription to view, but are non-functional to users.

Every Marketplace item has:

- An Azure Resource Manager template for resource provisioning.
- Metadata, such as strings, icons, and other marketing collateral.
- Formatting information to display the item in the portal.

Every item published to the Azure Stack Hub Marketplace uses the Azure Gallery Package (.azpkg) format. Add deployment or runtime resources (code, .zip files with software, or VM images) to Azure Stack Hub separately, not as part of the Marketplace item.

With version 1803 and later, Azure Stack Hub converts images to sparse files when they download from Azure or when you upload custom images. This process adds time when adding an image, but saves space and speeds up the deployment of those images. Conversion only applies to new images. Existing images aren't changed.

Next steps

- [Download existing marketplace items from Azure and publish to Azure Stack Hub](#)
- [Create and publish a custom Azure Stack Hub Marketplace item](#)

Azure Stack Hub Marketplace changes

2 minutes to read • [Edit Online](#)

This article describes recent additions, updates, changes, and removals of [Azure Stack Hub Marketplace items](#) over approximately the last 3 months. The information in this section is updated frequently, so check back often for changes.

The [Azure Stack Hub Marketplace items](#) article always contains the most current list of available Azure Stack Hub Marketplace items.

New Marketplace items

- 06/27/2019: SIOS Datakeeper Cluster Edition
- 06/27/2019: Veeam Backup & Replication
- 06/27/2019: Windows Server 2019 Datacenter Server Core With Containers Pay-as-you-use
- 06/27/2019: Windows Server 2019 Datacenter Server Core With Containers BYOL
- 06/27/2019: Windows Server 2019 Datacenter Pay-as-you-use
- 06/27/2019: Windows Server 2019 Datacenter BYOL
- 06/27/2019: Windows Server 2019 Datacenter Server Core Pay-as-you-use
- 06/27/2019: Windows Server 2019 Datacenter Server Core BYOL
- 06/27/2019: Windows Server 2019 Datacenter With Containers Pay-as-you-use
- 06/27/2019: Windows Server 2019 Datacenter With Containers BYOL
- 08/05/2019: Bitnami Drupal
- 08/05/2019: Bitnami etcd
- 08/05/2019: Bitnami Grafana
- 08/05/2019: Bitnami Neo4j
- 08/05/2019: Bitnami Parse Server
- 08/05/2019: Bitnami WordPress Multisite
- 08/05/2019: Bitnami ZooKeeper
- 08/05/2019: Bitnami TensorFlow Serving
- 08/05/2019: Bitnami NATS
- 08/05/2019: Bitnami Review Board
- 08/05/2019: Bitnami Composr
- 08/09/2019: Oracle Linux
- 08/19/2019: Iguazio Data Science Platform
- 10/16/2019: SIOS DataKeeper Cluster Edition
- 12/26/2019: CloudGuard IaaS High Availability
- 12/26/2019: Check Point CloudGuard IaaS Security Management
- 12/26/2019: Check Point CloudGuard IaaS Single Gateway
- 01/21/2020: Teradici Cloud Access Software

Deprecated Marketplace items

- Bitnami Codiad
- Bitnami X2Engine Sales CRM
- Bitnami SugarCRM

- Bitnami Node.js High-Availability Cluster
- A10 vThunder: L4-L7 Application Delivery Controller, Global Server Load Balancing (GSLB), SSL Insight
- Check Point vSEC Security Management

Updated Marketplace items

- 07/11/2019: Thales CipherTrust Cloud Key Manager - version 1.0.7
- 07/16/2019: Windows Server 2016 Datacenter Server Core BYOL - version 2016.127.20190522
- 07/16/2019: Windows Server 2016 Datacenter Server Core Pay-as-you-use - version 2016.127.20190522
- 07/16/2019: Windows Server 2016 Datacenter With Containers BYOL - version 2016.127.20190522
- 07/16/2019: Windows Server 2016 Datacenter With Containers Pay-as-you-use - version 2016.127.20190522
- 07/16/2019: Windows Server 2016 Datacenter BYOL - version 2016.127.20190522
- 07/16/2019: Windows Server 2016 Datacenter Pay-as-you-use - version 2016.127.20190522
- 07/16/2019: Windows Server 2008 R2 BYOL - version 2.127.20190522
- 07/16/2019: Windows Server 2008 R2 Pay-as-you-use - version 2.127.20190522
- 07/16/2019: Windows Server 2012 R2 Datacenter BYOL - version 4.127.20190522
- 07/16/2019: Windows Server 2012 R2 Datacenter Pay-as-you-use - version 4.127.20190522
- 07/30/2019: SQL Server 2016 SP1 (Express, Developer, Standard and Enterprise editions) - version 13.2.20190410
- 07/30/2019: SQL Server 2016 SP2 (Express, Developer, Standard and Enterprise editions) - version 13.2.20190410
- 07/30/2019: SQL Server 2017 (Express, Developer, Standard and Enterprise editions) - version 14.0.20190410
- 07/30/2019: SQLIaaS Extension - version 1.3.20180
- 08/08/2019: CloudLink SecureVM Extension for Linux - version 6.8
- 08/08/2019: CloudLink SecureVM Extension for Windows - version 6.8
- 08/30/2019: Azure Stack Hub Kubernetes Cluster - version 0.5.1
- 09/02/2019: AKS base image - version 2019.08.09 and version 2019.08.21
- 09/12/2019: Service Fabric - version 1.0.3
- 09/24/2019: F5-Networks Big-IP Virtual Edition - version 14.1.200000
- 09/25/2019: Debian 8 - version 8.0.20190806
- 10/1/2019: Microsoft JsonADDomain Extension - version 1.3.4

Next steps

For more information about the Azure Stack Hub Marketplace, see the following articles:

- [Azure Marketplace overview](#)
- [Azure Marketplace items available for Azure Stack Hub](#)
- [Create and publish an Azure Stack Hub Marketplace item](#)

Azure Marketplace items available for Azure Stack Hub

20 minutes to read • [Edit Online](#)

Virtual Machine extensions

Whenever there are updates to virtual machine (VM) extensions you use, you should download them. Extensions shipped in the product don't update in the normal patch and update process, so check for updates frequently. Other extensions are only available through Marketplace Management.

	ITEM NAME	DESCRIPTION	PUBLISHER	OS TYPE
	SQL IaaS Extension (SqlIaaSExtension)	Download this extension to deploy any SQL Server on Windows Marketplace item - this extension is required.	Microsoft	Windows
	Custom Script Extension	Download this update to the inbox version of the Custom Script Extension for Windows.	Microsoft	Windows
	PowerShell DSC Extension	Download this update to the inbox version of the PowerShell DSC Extension. Updated to support TLS v1.2.	Microsoft	Windows
	Microsoft Antimalware Extension	The Microsoft Antimalware for Azure is a single-agent solution for apps and tenant environments, designed to run in the background without human intervention. Download this update to the inbox version of the Antimalware Extension.	Microsoft	Windows

	ITEM NAME	DESCRIPTION	PUBLISHER	OS TYPE
	Microsoft Azure Diagnostic Extension	<p>Microsoft Azure Diagnostics is the capability within Azure that enables the collection of diagnostic data on a deployed app.</p> <p>Download this update to the inbox version of the Diagnostic Extension for Windows.</p>	Microsoft	Windows
	Azure Monitor, Update and Configuration Management Extension	<p>The Azure Monitor, Update and Configuration Management Extension is used with Log Analytics, Azure Security Center, and Azure Sentinel to provide VM monitoring capability.</p> <p>Download this update to the inbox version of the Monitoring Agent Extension for Windows.</p>	Microsoft	Windows
	- Custom Script Extension (version 1, deprecated) - Custom Script Extension (version 2)	<p>Download this update to the inbox version of the Custom Script Extension for Linux.</p> <p>There are multiple versions of this extension and you should download both 1.5.2.1 and 2.0.x.</p>	Microsoft	Linux
	VM Access for Linux	<p>Download this update to the inbox version of the VMAccess for Linux Extension. This update is important if you plan to use Debian Linux VMs.</p>	Microsoft	Linux
	Acronis Backup Extension for Linux	<p>The Acronis Backup Extension for Microsoft Azure is part of the Acronis Backup family of data protection products.</p>	Acronis International GmbH.	Linux

	ITEM NAME	DESCRIPTION	PUBLISHER	OS TYPE
	Acronis Backup Extension for Windows	The Acronis Backup Extension for Microsoft Azure is part of the Acronis Backup family of data protection products.	Acronis International GmbH.	Windows
	CloudLink SecureVM Extension for Linux	Control, monitor, and encrypt VMs with ease and confidence.	Dell EMC	Linux
	CloudLink SecureVM Extension for Windows	Control, monitor, and encrypt VMs with ease and confidence.	Dell EMC	Windows
	Kaspersky Hybrid Cloud Security Agent for Windows	With Kaspersky Hybrid Cloud Security, you can provision cybersecurity capabilities inside your cloud workloads via Azure Extensions.	Kaspersky Lab	Windows
	Kaspersky Hybrid Cloud Security Agent for Linux	With Kaspersky Hybrid Cloud Security, you can provision cybersecurity capabilities right inside your cloud workloads via Azure Extensions.	Kaspersky Lab	Linux

Microsoft VM images and solution templates

Microsoft Azure Stack Hub supports the following Azure Marketplace VMs and solution templates. Download any dependencies separately, as noted. Apps such as SQL Server and Machine Learning Server require proper licensing, except where marked as Free or Trial.

	ITEM NAME	DESCRIPTION	PUBLISHER
	Windows Server	Enterprise-class solutions that are simple to deploy, cost-effective, app-focused, and user-centric. These images are updated regularly with the latest patches. Important information: Images downloaded before January 18, 2018 must be deleted and replaced with the latest versions.	Microsoft

	ITEM NAME	DESCRIPTION	PUBLISHER
	SharePoint Server 2013 Trial	Microsoft SharePoint Server 2013 Trial on Windows Server 2012 Datacenter and Visual Studio 2019 community edition.	Microsoft
	SharePoint Server 2016 Trial	Microsoft SharePoint Server 2016 Trial on Windows Server 2016 Datacenter.	Microsoft
	SQL Server 2014 SP3 on Windows Server 2012 R2	SQL Server 2014 Service Pack 2. Required download: SQL IaaS Extension.	Microsoft
	Free License: SQL Server 2016 SP1 Developer on Windows Server 2016	Free developer version of SQL Server 2016 SP1 for transactional, data warehousing, business intelligence, and analytics workloads. Required download: SQL IaaS Extension.	Microsoft
	Free License: SQL Server 2016 SP1 Express on Windows Server 2016	Free express version of SQL Server 2016 SP1. Required download: SQL IaaS Extension.	Microsoft
	SQL Server 2016 SP1 Standard on Windows Server 2016	Database platform for intelligent, mission-critical apps. Required download: SQL IaaS Extension.	Microsoft
	SQL Server 2016 SP1 Enterprise on Windows Server 2016	Database platform for intelligent, mission-critical apps. Required download: SQL IaaS Extension.	Microsoft
	Free License: SQL Server 2016 SP2 Developer on Windows Server 2016	Free developer version of SQL Server 2016 SP2 for transactional, data warehousing, business intelligence, and analytics workloads. Required download: SQL IaaS Extension.	Microsoft
	Free License: SQL Server 2016 SP2 Express on Windows Server 2016	Free express version of SQL Server 2016 SP2. Required download: SQL IaaS Extension.	Microsoft
	SQL Server 2016 SP2 Standard on Windows Server 2016	Database platform for intelligent, mission-critical apps. Required download: SQL IaaS Extension.	Microsoft

	ITEM NAME	DESCRIPTION	PUBLISHER
	SQL Server 2016 SP2 Enterprise on Windows Server 2016	Database platform for intelligent, mission-critical apps. Required download: SQL IaaS Extension.	Microsoft
	Free License: SQL Server 2017 Developer on Windows Server 2016	Free developer version of SQL Server 2017 for transactional, data warehousing, business intelligence, and analytics workloads. Required download: SQL IaaS Extension.	Microsoft
	Free License: SQL Server 2017 Express on Windows Server 2016	Free express version of SQL Server 2017. Required download: SQL IaaS Extension.	Microsoft
	SQL Server 2017 Standard on Windows Server 2016	Database platform for intelligent, mission-critical apps. Required download: SQL IaaS Extension.	Microsoft
	SQL Server 2017 Enterprise on Windows Server 2016	Database platform for intelligent, mission-critical apps. Required download: SQL IaaS Extension.	Microsoft
	SQL Server 2017 Enterprise on Ubuntu Server 16.04 LTS	Database platform for intelligent, mission-critical apps.	Microsoft + Canonical
	SQL Server 2017 Standard on SUSE Linux Enterprise Server (SLES) 12 SP2	Database platform for intelligent, mission-critical apps.	Microsoft + SUSE
	Free License: SQL Server 2017 Developer on SUSE Linux Enterprise Server (SLES) 12 SP2	Free developer version of SQL Server 2017 for transactional, data warehousing, business intelligence, and analytics workloads.	Microsoft + SUSE
	Free License: SQL Server 2017 Express on SUSE Linux Enterprise Server (SLES) 12 SP2	Free express version of SQL Server 2017.	Microsoft + SUSE
	SQL Server 2017 Enterprise on SUSE Linux Enterprise Server (SLES) 12 SP2	Database platform for intelligent, mission-critical apps.	Microsoft + SUSE
	SQL Server 2017 Web on SUSE Linux Enterprise Server (SLES) 12 SP2	Database platform for intelligent, mission-critical apps.	Microsoft + SUSE

	ITEM NAME	DESCRIPTION	PUBLISHER
	Microsoft Machine Learning Server 9.3.0 on Windows Server 2016	Microsoft Machine Learning Server 9.3.0 on Windows Server 2016.	Microsoft
	Microsoft Machine Learning Server 9.3.0 on Ubuntu 16.04	Microsoft Machine Learning Server 9.3.0 on Ubuntu 16.04.	Microsoft + Canonical
	Microsoft Machine Learning Server 9.3.0 on CentOS Linux 7.2	Microsoft Machine Learning Server 9.3.0 on CentOS Linux 7.2.	Microsoft + Rogue Wave

Linux distributions

	ITEM NAME	DESCRIPTION	PUBLISHER
	Clear Linux OS	A reference Linux distribution optimized for Intel Architecture.	Clear Linux Project
	CoreOS Linux (Stable)	CoreOS is a modern and minimal Linux distribution, providing an easy way to run containers, manage clusters and seamlessly update your servers - all components that enable warehouse-scale compute.	CoreOS
	Ubuntu Server	Ubuntu Server is the world's most popular Linux for cloud environments.	Canonical
	Debian 8 "Jessie"	Debian GNU/Linux is one of the most popular Linux distributions.	credativ
	Oracle Linux	The Oracle Linux operating system is engineered for open cloud infrastructure. It delivers leading performance, scalability, and reliability for enterprise SaaS and PaaS workloads, as well as traditional enterprise apps.	Oracle
	CentOS-based 6.8	This distribution of Linux is based on CentOS and is provided by Rogue Wave Software.	Rogue Wave Software (formerly OpenLogic)
	CentOS-based 6.10	This distribution of Linux is based on CentOS and is provided by Rogue Wave Software.	Rogue Wave Software (formerly OpenLogic)

	ITEM NAME	DESCRIPTION	PUBLISHER
--	-----------	-------------	-----------

	CentOS-based 7.3	This distribution of Linux is based on CentOS and is provided by Rogue Wave Software.	Rogue Wave Software (formerly OpenLogic)
	CentOS-based 7.5	This distribution of Linux is based on CentOS and is provided by Rogue Wave Software.	Rogue Wave Software (formerly OpenLogic)
	CentOS-based 7.5-LVM	This distribution of Linux is based on CentOS and is provided by Rogue Wave Software.	Rogue Wave Software (formerly OpenLogic)
	SLES 11 SP4 (BYOS)	SUSE Linux Enterprise Server 11 SP4.	SUSE
	SLES 12 SP4 (BYOS)	SUSE Linux Enterprise Server 12 SP4.	SUSE
	SLES 15 (BYOS)	SUSE Linux Enterprise Server 15.	SUSE

Third-Party BYOL, free, trial images, and solution templates

	ITEM NAME	DESCRIPTION	PUBLISHER
	A10 vThunder ADC	The A10 Networks vThunder ADC (Application Delivery Controller) for Microsoft Azure is purpose-built for high performance, flexibility, and easy-to-deploy app delivery and server load balancing and optimized to run natively within the Azure cloud.	A10 Networks
	Arista vEOS Router	The Arista vEOS Router is a feature-rich, multi-cloud, and multi-hypervisor virtual router that empowers enterprises and cloud providers to build consistent, highly secure, and scalable hybrid networks.	Arista Networks

	ITEM NAME	DESCRIPTION	PUBLISHER
	Barracuda Application Security Control Center	Centrally manage multiple Barracuda Web Application Firewalls (WAF).	Barracuda Networks, Inc.
	Barracuda Email Security Gateway	Email security gateway to protect against inbound email-borne threats.	Barracuda Networks, Inc.
	Barracuda Web Application Firewall (WAF)	Security and DDoS Protection Against Automated & Targeted Attacks.	Barracuda Networks, Inc.
	Barracuda CloudGen Firewall Control Center	Centrally manage hundreds of Barracuda CloudGen Firewalls regardless of their location and form factor.	Barracuda Networks, Inc.
	Barracuda CloudGen Firewall for Azure	Provides firewall protection where the app and data reside, rather than solely where the connection terminates.	Barracuda Networks, Inc.
	AbanteCart	Open-source ecommerce shopping cart.	Bitnami
	ActiveMQ	Open-source message broker in Java.	Bitnami
	Akeneo	Powerful PIM designed to simplify management processes.	Bitnami
	Alfresco Community	ECM system for easy document management.	Bitnami
	Apache Solr	Reliable open-source enterprise search platform.	Bitnami
	Canvas LMS	Open-source learning management system.	Bitnami
	Cassandra	Scalable open-source database with high availability.	Bitnami

	ITEM NAME	DESCRIPTION	PUBLISHER
	Cassandra Cluster	Apache Cassandra is an open-source distributed database management system designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure. This solution template also requires Debian 8 and Custom Script for Linux 2.0 Extension.	Bitnami
	CiviCRM	Simple web-based relationship management system.	Bitnami
	CMS Made Simple	Fast and easy way to create and manage a website.	Bitnami
	Composr	Composr is a CMS with advanced content, social, interactive, and dynamic features. Fully flexible, theme-able, and extendable: suitable for building almost any kind of website.	Bitnami
	Concrete5	Easily deploy web apps, websites, stores, and forums.	Bitnami
	Coppermine	Multi-purpose, fully featured web gallery.	Bitnami
	CouchDB	Easy-to-use open-source database system.	Bitnami
	Diaspora	Popular personal web server.	Bitnami
	Discourse	High-resolution open-source discussion platform.	Bitnami
	Django	High-level Python Web framework.	Bitnami
	Dolibarr	Free open-source software package.	Bitnami
	DokuWiki	Versatile open-source wiki software.	Bitnami

	ITEM NAME	DESCRIPTION	PUBLISHER
	DreamFactory	Open-source REST API with services such as SQL, NoSQL, and BLOB.	Bitnami
	Drupal	Drupal is one of the most versatile open-source content management systems in the world. It's pre-configured with the Ctools and Views modules, Drush and Let's Encrypt autoconfiguration support.	Bitnami
	Elasticsearch	Flexible and powerful open-source analytics engine.	Bitnami
	ELK	Big data suite consisting of Elasticsearch, Kibana, and Logstash.	Bitnami
	ERPNext	Open-source Enterprise Resource Planning (ERP) platform.	Bitnami
	EspoCRM	Simple CRM system that helps manage customer relationships.	Bitnami
	etcd	etcd is a distributed key-value store designed to securely store data across a cluster. etcd is widely used in production for its reliability, fault-tolerance, and ease of use.	Bitnami
	eXo Platform	Open-source, social software designed for enterprises.	Bitnami
	Fat Free CRM	Open-source Ruby on Rails-based CRM.	Bitnami
	GitLab Community Edition	Fast, secure Git management software.	Bitnami
	Ghost	A platform dedicated to publishing.	Bitnami

	ITEM NAME	DESCRIPTION	PUBLISHER
	Grafana	Grafana is an open-source analytics and monitoring dashboard for more than 40 data sources, including Graphite, Elasticsearch, Prometheus, MariaDB/MySQL, PostgreSQL, InfluxDB, OpenTSDB, and many more.	Bitnami
	Hadoop	Framework for reliable, scalable, and distributed computing.	Bitnami
	HHVM	Fully integrated and ready-to-run development environment.	Bitnami
	Horde Groupware Webmail	Free, enterprise ready, browser-based communication suite.	Bitnami
	Jenkins	Integration server supporting SCM tools: CVS, Subversion, and Git.	Bitnami
	Jenkins Cluster	Jenkins CI is an open-source continuous integration server. This solution template also requires Debian 8 and Custom Script for Linux 2.0 Extension.	Bitnami
	JFrog Artifactory	Binary Repository software from the leading publisher.	Bitnami
	Joomla!	User-friendly CMS for easy website builds.	Bitnami
	JRuby	High-performance Java implementation of Ruby.	Bitnami
	Kafka	Powerful distributed publish-subscribe messaging system.	Bitnami

	ITEM NAME	DESCRIPTION	PUBLISHER
	Kafka Cluster	Apache Kafka is publish-subscribe messaging rethought as a distributed commit log. This solution improves the reliability of a Kafka cluster by provisioning multiple Kafka brokers and Zookeeper instances. This solution template also requires Debian 8 and Custom Script for Linux 2.0 Extension.	Bitnami
	LAMP	Fully integrated and ready to run development environment.	Bitnami
	LAPP	Complete PHP, PostgreSQL, and Apache development environment.	Bitnami
	Let's Chat	Open-source persistent messaging app.	Bitnami
	LimeSurvey	Question-and-answer poll management system.	Bitnami
	Live Helper Chat	Open-source live chat support.	Bitnami
	Mahara	Popular open-source ePortfolio and social networking web app.	Bitnami
	Magento	Popular eCommerce software and platform.	Bitnami
	Mantis	Advanced bug-tracking system.	Bitnami

	ITEM NAME	DESCRIPTION	PUBLISHER
	MariaDB with Replication	MariaDB is an open-source, community-developed SQL database server that is widely in use around the world because of its enterprise features, flexibility, and collaboration with leading tech firms. This solution uses multiple VMs to replicate the databases from the master node to a configurable number of replicas. This solution template also requires Debian 8 and Custom Script for Linux 2.0 Extension.	Bitnami
	Mattermost Team Edition	Open-source workplace messaging solution.	Bitnami
	Mautic	Open-source, enterprise marketing automation platform.	Bitnami
	MEAN	Popular development environment for mongoDB and Node.js.	Bitnami
	MediaWiki	Powerful, scalable wiki implementation.	Bitnami
	Memcached	High-performance, distributed memory object caching system.	Bitnami
	Memcached Multiple Instances	Memcached is a high-performance, distributed memory object caching system. This solution provisions multiple Memcached nodes to create a high performance, failure-resistant distributed cache for your app. This solution template also requires Debian 8 and Custom Script for Linux 2.0 Extension.	Bitnami
	MODX	Intuitive Web CMS.	Bitnami
	MongoDB	High-performance open-source NoSQL database written in C++.	Bitnami

	ITEM NAME	DESCRIPTION	PUBLISHER
	MongoDB with Replication	High-performance open-source NoSQL database written in C++. This solution template requires your Azure Stack Hub to be at version 1807 or later, and also requires Debian 8 and Custom Script for Linux 2.0 Extension.	Bitnami
	Moodle	Effective CMS designed for online learning communities.	Bitnami
	MyBB	Free and open-source Forum Software.	Bitnami
	MySQL	The most popular database system.	Bitnami
	MySQL with Replication	MySQL is a fast, reliable, scalable, and easy to use open-source relational database system. MySQL Server is intended for mission-critical, heavy-load production systems as well as for embedding into mass-deployed software. This solution uses multiple VMs to replicate the databases from the master node to a configurable number of replicas. This solution template also requires Debian 8 and Custom Script for Linux 2.0 Extension.	Bitnami
	NATS	NATS is an open-source, lightweight, and high-performance messaging system. It's ideal for distributed systems, and supports modern cloud architectures and publish-subscribe, request-reply, and queuing models.	Bitnami
	Neos	Versatile open-source Content Management System.	Bitnami

	ITEM NAME	DESCRIPTION	PUBLISHER
	Neo4j	Neo4j is a high-performance graph store with all the features expected in a mature and robust database, such as a friendly query language and ACID transactions.	Bitnami
	Nginx	A complete PHP, MySQL, and Nginx development environment.	Bitnami
	Noalys	Powerful double-entry accounting system.	Bitnami
	node.js	Open-source environment written in Javascript for easy building.	Bitnami
	Odoo	ERP and CRM system that effectively connects business processes.	Bitnami
	Open Atrium	Flexible, multi-faceted Intranet platform.	Bitnami
	OpenCart	Free e-commerce platform for online merchants.	Bitnami
	Open edX	eLearning software from the leading publisher.	Bitnami
	OpenFire	Open-source real-time collaboration server with XMPP.	Bitnami
	OpenProject	Popular open-source project management software.	Bitnami
	OrangeHRM	HR management system with a wealth of modules.	Bitnami
	OroCRM	Flexible open-source CRM app.	Bitnami
	Osclass	Create and manage free classified ads without technical know-how.	Bitnami
	ownCloud	Popular open-source file sync and share solution.	Bitnami

	ITEM NAME	DESCRIPTION	PUBLISHER
	OXID eShop	Trusted open-source e-commerce system.	Bitnami
	Parse Server	Parse is a platform that enables users to add a scalable and powerful backend to launch a full-featured app for iOS, Android, JavaScript, Windows, Unity, and more.	Bitnami
	phpBB	Customizable bulletin board solution.	Bitnami
	phpList	One-way email announcement delivery system.	Bitnami
	Pimcore	Powerful engagement management platform (CEM/CXM).	Bitnami
	Piwik	Real-time web analytics software program.	Bitnami
	Plone	Free open-source virtual appliance.	Bitnami
	Pootle	Easy-to-use web portal for translation projects.	Bitnami
	PostgreSQL	Highly advanced open-source database.	Bitnami
	PostgreSQL	PostgreSQL (Postgres) is an open-source object-relational database system. ACID-compliant, it supports foreign keys, joins, views, triggers, and stored procedures. It's known for reliability and data integrity. This solution uses multiple VMs to replicate the database from the master node to a configurable number of replicas. This solution template also requires Debian 8 and Custom Script for Linux 2.0 Extension.	Bitnami
	PrestaShop	Open-source e-commerce website builder.	Bitnami

	ITEM NAME	DESCRIPTION	PUBLISHER
	Process Maker Community Edition	Business Process Management and workflow automation platform.	Bitnami
	Process Maker Enterprise Edition	Open-source workflow and Business Process Management software.	Bitnami
	ProcessWire	Popular PHP5 open-source CMS.	Bitnami
	Publify	Ruby on Rails-based blogging platform.	Bitnami
	RabbitMQ	Efficient messaging broker offering a common platform.	Bitnami
	RabbitMQ Cluster	RabbitMQ is a messaging broker that gives your apps a common platform to send and receive messages, and your messages a safe place to live until received. This solution uses multiple VMs to provision multiple nodes in a RabbitMQ Cluster to form a single logical broker. This solution template also requires Debian 8 and Custom Script for Linux 2.0 Extension.	Bitnami
	Re:dash	Open-source Data Visualization and Collaboration Platform.	Bitnami
	Redis	Powerful open-source key-value store.	Bitnami
	Redis High Availability	Powerful open-source key-value store. This solution template requires your Azure Stack Hub to be at version 1807 or later, and also requires Debian 8 and Custom Script for Linux 2.0 Extension.	Bitnami
	Redmine	Powerful project management web app.	Bitnami
	Redmine+Agile	Project management app preconfigured with Agile plugin.	Bitnami

	ITEM NAME	DESCRIPTION	PUBLISHER
	ReportServer Community	Open-source business intelligence platform.	Bitnami
	ReportServer Enterprise	Enterprise business intelligence platform.	Bitnami
	ResourceSpace	Digital asset management system for improved collaboration.	Bitnami
	Review Board	Review Board is a web-based code review app that offers developers different kinds of tools to easily conduct simplified code reviews. It scales well from small projects to large companies.	Bitnami
	Roundcube	Browser-based IMAP client with functionality such as MIME support.	Bitnami
	Ruby	Easy-to-use development environment for Ruby on Rails.	Bitnami
	SEO Panel	Open-source SEO management app for tracking multiple websites.	Bitnami
	Shopware	Open-source eCommerce Platform.	Bitnami
	Simple Machines Forum	Simple forum software to build your own online community.	Bitnami
	Spree	Easy-to-use ecommerce platform.	Bitnami
	Subversion	Open-source version control system.	Bitnami
	SuiteCRM	Popular Enterprise-grade CRM app.	Bitnami

	ITEM NAME	DESCRIPTION	PUBLISHER
	TensorFlow Serving	TensorFlow Serving is a system for serving machine learning models. This stack comes with Inception version 3 with trained data for image recognition, but it can be extended to serve other models.	Bitnami
	TestLink	Test management software facilitating quality assurance.	Bitnami
	Tiki Wiki CMS Groupware	Fully featured wiki platform.	Bitnami
	Tiny Tiny RSS	Flexible open-source web-based news feed and aggregator.	Bitnami
	Tomcat	Popular platform implementing specifications from Java Community.	Bitnami
	Trac	Enhanced wiki and issue tracking system.	Bitnami
	Typo3	Fully flexible CMS.	Bitnami
	Weblate	Web-based translation management system.	Bitnami
	WebMail Pro PHP	Webmail system with enterprise features.	Bitnami
	WildFly	App server that includes Apache, WildFly, MySQL, and Java.	Bitnami
	WordPress	The most popular and ready-to-go CMS.	Bitnami
	WordPress Multisite	WordPress is the world's most popular blogging and content management platform. With WordPress Multisite, conserve resources by managing multiple blogs and websites from the same server and interface.	Bitnami

	ITEM NAME	DESCRIPTION	PUBLISHER
	Xoops	CMS and Web Portal Program that creates dynamic websites.	Bitnami
	Zurmo	Open-source CRM system: Mobile, Social, and Gamified.	Bitnami
	ZooKeeper	ZooKeeper provides a reliable, centralized register of configuration data and services for distributed apps.	Bitnami
	CloudGuard IaaS High Availability	This solution deploys a 2 member Check Point CloudGuard IaaS cluster. Each member has 2 network interfaces.	Check Point
	Check Point CloudGuard IaaS Security Management	This solution deploys a single Check Point Security Management Server with a single network interface.	Check Point
	Check Point CloudGuard IaaS Single Gateway	This solution deploys a single Check Point CloudGuard IaaS security gateway with 2 network interfaces. After deployment, you should set up User Defined Routes (UDRs) to route traffic through the gateway.	Check Point
	Chef Automate	Build, deploy, and manage with Chef Automate, the Continuous Automation Platform. Download both Chef marketplace items.	Chef Software, Inc
	Commvault	A comprehensive solution for backup and recovery, app and VM migration to Azure Stack Hub, and disaster recovery for Azure Stack Hub environments in a single solution.	Commvault
	CloudLink SecureVM	Control, monitor, and encrypt VMs with ease and confidence. Download all CloudLink SecureVM items.	Dell EMC

	ITEM NAME	DESCRIPTION	PUBLISHER
	EventTracker SIEM	EventTracker SIEM is a comprehensive security platform that delivers advanced security tools with audit-ready compliance capabilities.	EventTracker
	Exivity - Hybrid Cloud Billing Solution	A billing tool that can satisfy the requirements of virtually any IT service delivery model, whether deployed within on-premises, public cloud, or hybrid environments.	Exivity
	f5 Big-IP Virtual Edition	Advanced Load Balancing, GSLB, Network Firewall, DNS, WAF, and App Access.	F5 Networks
	FortiGate Next-Generation Firewall	Firewall technology that delivers complete content and network protection with a comprehensive suite of powerful security features. App control, antivirus, IPS, web filtering, and VPN along with advanced features such as vulnerability management work in concert to identify and mitigate the latest complex security threats.	Fortinet
	Hortonworks Data Platform (HDP) Sandbox	Powered by HDP 2.5 100% open-source platform for Hadoop, Spark, Storm, HBase, Kafka, Hive, Ambari.	Hortonworks
	Kaspersky Hybrid Cloud Security	The Kaspersky Hybrid Cloud Security enables a seamlessly orchestrated and adaptive cybersecurity ecosystem.	Kaspersky Lab
	KEMP LoadMaster Load Balancer ADC Content Switch	Layer 4-7 Application Delivery Controller (ADC) Load Balancer, Content Switch, and Traffic Manager.	KEMP Technologies Inc.

	ITEM NAME	DESCRIPTION	PUBLISHER
	Kubernetes	<p>This solution deploys a Kubernetes cluster running as a standalone cluster with templates generated using AKS-Engine.</p> <p>This solution template also requires Ubuntu Server 16.04 LTS and Custom Script for Linux 2.0.</p>	Microsoft
	Service Fabric Cluster	<p>This solution deploys Service Fabric running as a standalone cluster on a Virtual Machine Scale Set.</p> <p>This solution template requires you to also download the Windows Server 2016 Datacenter</p>	Microsoft
	mPLAT Suite - Multi-Cloud Conductor	A Single Pane of Glass to monitor, configure, provision, automate, and govern any workload or cloud.	NRI
	NooBaa Hybrid AWS S3 compatible - Community Edition	S3-compatible storage service that spans public and on-premises capacity resources.	NooBaa
	NetFoundry Gateway for Multipoint, Zero Trust Azure Stack Hub Connections	Software-only, multi-point connectivity between Azure Stack Hub and anywhere, over any network connection, with industry leading Zero Trust security, 5x the throughput of VPN, and unlimited concurrent users.	NetFoundry
	Palo Alto VM-Series Next Generation Firewall	<p>The VM-Series next-generation firewall allows customers to securely migrate their apps and data to Azure Stack Hub, protecting them from known and unknown threats with app whitelisting and threat prevention policies. This image requires a template to deploy; see this article for important information.</p>	Palo Alto Networks, Inc.

	ITEM NAME	DESCRIPTION	PUBLISHER
	PT Application Firewall	PT Application Firewall detects known & unknown vulnerabilities and prevents attacks on web apps. Download both PT Marketplace items.	Positive Technologies
	Puppet Enterprise	Puppet Enterprise lets you automate the entire lifecycle of your Azure Stack Hub infrastructure. Download both Puppet Marketplace items.	Puppet
	Quest Rapid Recovery Core	Rapid Recovery advanced data protection unifies backup, replication, and recovery in one easy-to-use software solution.	Quest Software
	SIOS DataKeeper Cluster Edition	SIOS DataKeeper provides high availability (HA) and disaster recovery (DR) in Azure Stack Hub. Simply add SIOS DataKeeper software as an ingredient to your Windows Server Failover Clustering (WSFC) environment in an Azure Stack Hub deployment to eliminate the need for shared storage.	SIOS Technology Corp.
	SUSE Manager 3.1 Proxy (BYOS)	Best-in-class open-source infrastructure management.	SUSE
	Teradici Cloud Access Software	Powered by PCoIP® technology, Cloud Access Software delivers remote desktops and workstations from Azure Stack to any device, anywhere. Consolidate data storage, enhance collaboration, secure data, streamline desktop management, and more.	Teradici

Item Name	Description	Publisher	
	<p>CipherTrust Cloud Key Manager</p>	<p>Leveraging Microsoft Azure and other cloud provider Bring Your Own Key (BYOK) APIs, the CipherTrust Cloud Key Manager reduces key management complexity and operational costs by giving you multicloud lifecycle control of encryption keys with centralized management and visibility.</p>	Thales eSecurity
	<p>Veeam Backup & Replication</p>	<p>Veeam® Backup & Replication™ helps businesses achieve comprehensive data protection for all workloads — virtual, physical, and cloud-based. With a single console, you can achieve fast, flexible, and reliable backup, recovery, and replication of all apps and data.</p>	Veeam Software
	<p>ZeroDown Software Business Continuity as a Service</p>	<p>ZeroDown® Software technology provides businesses with continuous access to their company data via their Business Continuity as a Service (BCaaS)™ architecture, protecting apps, and transactions, if network interruptions occur that would normally cripple the enterprise.</p>	ZeroDown Software

Download Marketplace items to Azure Stack Hub

7 minutes to read • [Edit Online](#)

As a cloud operator, you can download items to Azure Stack Hub from the Marketplace and make them available to all users using the Azure Stack Hub environment. The items you can choose are from a curated list of Azure Marketplace items that are pre-tested and supported to work with Azure Stack Hub. Additional items are frequently added to this list, so continue to check back for new content.

There are two scenarios for downloading Marketplace products:

- **Connected scenario:** Requires your Azure Stack Hub environment to be connected to the internet. You use the Azure Stack Hub administrator portal to locate and download items.
- **Disconnected or partially connected scenario:** Requires you to access the internet using the Marketplace syndication tool to download Marketplace items. Then, you transfer your downloads to your disconnected Azure Stack Hub installation. This scenario uses PowerShell.

See [Azure Marketplace items for Azure Stack Hub](#) for a complete list of the marketplace items you can download. See the [Azure Stack Hub Marketplace changes](#) article for a list of recent additions, deletions, and updates to Azure Stack Hub Marketplace.

NOTE

The catalog will be different based on the cloud your Azure Stack Hub system is connected to. The cloud environment is determined by the Azure subscription you use for registering your Azure Stack Hub.

Connected scenario

If Azure Stack Hub connects to the internet, you can use the administrator portal to download marketplace items.

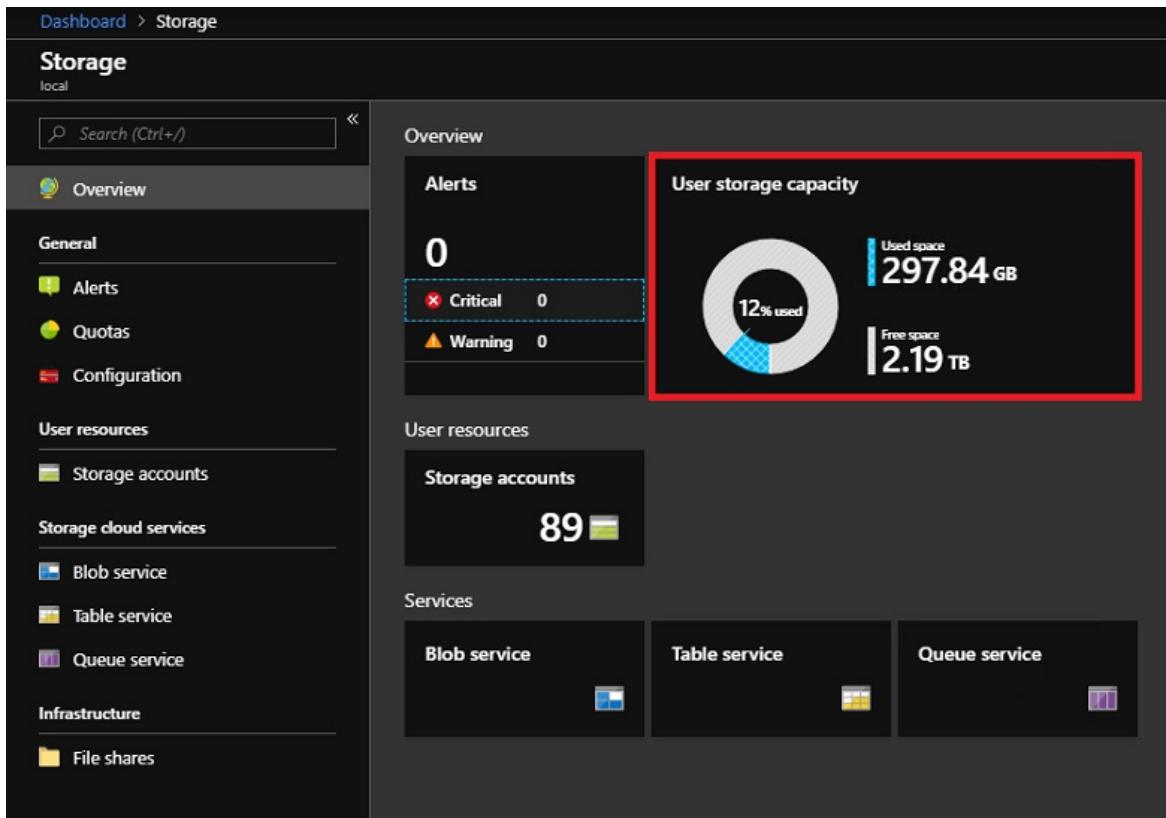
Prerequisites

Your Azure Stack Hub deployment must have internet connectivity and be registered with Azure.

Use the portal to download marketplace items

1. Sign into the Azure Stack Hub administrator portal.
2. Review the available storage space before downloading marketplace items. Later, when you select items for download, you can compare the download size to your available storage capacity. If capacity is limited, consider options for [managing available space](#).

To review available space: in **Region management**, select the region you want to explore and then go to **Resource Providers > Storage**:



3. Open Azure Stack Hub Marketplace and connect to Azure. To do so, select the **Marketplace management** service, select **Marketplace items**, and then select **Add from Azure**:

The screenshot shows the 'Marketplace management - Marketplace items' page. It has a search bar, a 'Marketplace items' button highlighted with a red box, and a 'Resource providers' button. There's also a 'Refresh' button and a 'Filter by name' input field. The results table shows two entries for 'Custom Script Extension'.

4. Each line item also shows the currently available version. If more than one version of a Marketplace item is available, the **Version** column shows **Multiple**. You can click on each item to view its description and additional information, including its download size:

The screenshot shows the 'Add from Azure' marketplace items list. It includes a 'Refresh' button, a 'Filter by name' input, and dropdowns for 'All publishers selected' and 'All types selected'. The table lists various marketplace items with columns for Name, Publisher, Type, and Version. Several items have 'Multiple' listed in the Version column, such as 'Azure Kubernetes Service' and 'Alfresco Community'.

NAME	PUBLISHER	TYPE	VERSION
AbaneCart Certified by Bitnami	Bitnami	Virtual Machine	1.2.1905211605
Acronis Backup	Acronis, Inc.	Virtual Machine	1.0.51
Acronis Backup for Linux (preview)	Acronis, Inc.	Virtual Machine	1.0
ActiveMQ Certified by Bitnami	Bitnami	Virtual Machine	5.15.1905152219
Akeneo Certified by Bitnami	Bitnami	Virtual Machine	3.1.1905272207
AKS Base Ubuntu 16.04-LTS Image Distro, August 2019	Azure Kubernetes Service	Virtual Machine	Multiple
AKS Base Ubuntu 16.04-LTS Image Distro, July 2019	Azure Kubernetes Service	Virtual Machine	Multiple
Alfresco Community	Bitnami	Virtual Machine	201704.0.0
Apache Solr Certified by Bitnami	Bitnami	Virtual Machine	8.1.1905170606
Arista vEOS Router 4.21.0F (BYOL)	Arista Networks	Virtual Machine	4.21.0
Azure Monitor Dependency Agent	Microsoft	Virtual Machine	9.7.4
Azure Monitor Dependency Agent for Linux VMs	Microsoft	Virtual Machine	9.7.4
Azure Monitor, Update and Configuration Management	Microsoft	Virtual Machine	1.0.11081.4
Azure Monitor, Update and Configuration Management for Linux VMs	Microsoft	Virtual Machine	1.8.11
Azure Performance Diagnostics	Microsoft Corp.	Virtual Machine	1.0.13
Azure Update and Configuration Management for Linux	Microsoft	Virtual Machine	1.8
Barracuda App Security Control Center - BYOL	Barracuda Networks, Inc.	Virtual Machine	2.1.100803

5. If the version of an item is shown as **Multiple**, you can select that item and then choose a specific version from the resulting version selector dropdown:

The screenshot shows the Microsoft Azure Stack - Administration portal. In the top navigation bar, it says "Microsoft Azure Stack - Administration". Below the navigation bar, there's a sidebar with various options like "Create a resource", "Dashboard", "All services", and "FAVORITES". Under "FAVORITES", there are links to "All resources", "Resource groups", "Virtual machines", "Load balancers", "Storage accounts", "Virtual networks", "Monitor", "Offers", and "Marketplace management". The main content area shows a marketplace item titled "AKS Base Ubuntu 16.04-LTS Image Distro, August 2019" from "Azure Kubernetes Service". It includes sections for "Publisher" (Azure Kubernetes Service), "Version" (with a dropdown menu highlighted by a red box containing "2019.08.21", "2019.08.21", and "2019.08.09"), "Type", "Download size" (300.0GB), "Terms of use", and "Privacy policy".

6. Select the item you want, and then select **Download**. Download times vary and depends on the network connectivity. After the download completes, you can deploy the new marketplace item as either an Azure Stack Hub operator or a user.
7. To deploy the downloaded item, select + **Create a resource**, and then search among the categories for the new marketplace item. Next, select the item to begin the deployment process. The process varies for different marketplace items.

Disconnected or a partially connected scenario

If Azure Stack Hub has limited or no internet connectivity, you can use PowerShell and the *marketplace syndication tool* to download the marketplace items to a machine with internet connectivity. You then transfer the items to your Azure Stack Hub environment. In a disconnected environment, you can't download marketplace items by using the Azure Stack Hub portal.

The marketplace syndication tool can also be used in a connected scenario.

There are two parts to this scenario:

- **Part 1:** Download from Marketplace items. On the computer with internet access, you configure PowerShell, download the syndication tool, and then download items from Azure Marketplace.
- **Part 2:** Upload and publish to Azure Stack Hub Marketplace. You move the files you downloaded to your Azure Stack Hub environment and then publish them to Azure Stack Hub Marketplace.

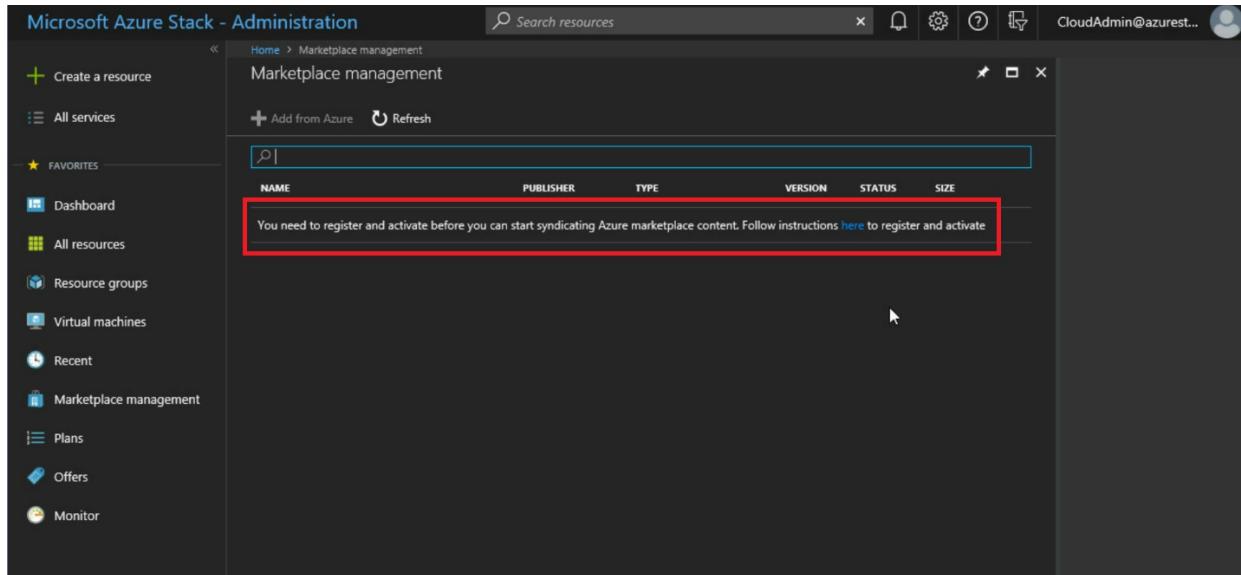
Prerequisites

- A connected environment (does not have to be Azure Stack Hub). You need connectivity to get the list of products from Azure with their details, and to download everything locally. Once this is done, the rest of the procedure does not require internet connectivity. It creates a catalog of items you've previously downloaded for you to use in your disconnected environment.
- Removable media to connect to your disconnected environment and transfer all the necessary artifacts.
- A disconnected Azure Stack Hub environment with the following prerequisites:
 - Your Azure Stack Hub deployment must be registered with Azure.
 - The computer that has internet connectivity must have **Azure Stack Hub PowerShell Module version 1.2.11** or later. If not already present, [install Azure Stack Hub-specific PowerShell modules](#).

- To enable import of a downloaded marketplace item, the [PowerShell environment for the Azure Stack Hub operator](#) must be configured.
- Download the Azs.Syndication.Admin Module from the PowerShell Gallery using the command below

```
Install-Module -Name Azs.Syndication.Admin
```

Once you have registered your Azure Stack, you can disregard the following message that appears on the Marketplace management blade, as this is not relevant for the disconnected use case:



Use the marketplace syndication tool to download marketplace items

IMPORTANT

Be sure to download the marketplace syndication tool each time you download marketplace items in a disconnected scenario. Frequent changes are made to this tool and the most current version should be used for each download.

1. On a computer with an Internet connection, open a PowerShell console as an administrator.
2. Sign in to the appropriate Azure cloud and AzureAD Directory Tenant using the Azure account that you've used to register Azure Stack Hub. To add the account, in PowerShell run **Add-AzureRmAccount**.

```
Login-AzureRmAccount -Environment AzureCloud -Tenant '<mydirectory>.onmicrosoft.com'
```

You are prompted to enter your Azure account credentials and you might have to use two-factor authentication, depending on your account configuration.

NOTE

If your session expires, your password has changed, or you simply wish to switch accounts, run the following cmdlet before you sign in using **Add-AzureRmAccount: Remove-AzureRmAccount-Scope Process**.

3. If you have multiple subscriptions, run the following command to select the one you've used for registration:

```
Get-AzureRmSubscription -SubscriptionID 'Your Azure Subscription GUID' | Select-AzureRmSubscription
```

4. If you haven't done it in the pre-requisites step already, download the latest version of the marketplace syndication tool:

```
Install-Module -Name Azs.Syndication.Admin
```

5. To select the Marketplace items such as VM images, extensions, or solution templates to download, run the following command.

```
$products = Select-AzsMarketplaceItem
```

This displays a table that lists all the Azure Stack registrations available in the selected subscription. Choose the registration that matches the Azure Stack environment you're downloading the marketplace items for, and select **OK**.

The screenshot shows a user interface for selecting a registration resource. At the top, there's a blue header bar with the text 'Please select registration resource'. Below it is a search bar labeled 'Filter'. Underneath is a button labeled 'Add criteria ▾'. The main area is a table with three columns: 'ResourceGroupName', 'Name', and 'ResourceId'. There are six rows in the table, each representing a different Azure Stack registration. The last row, 'azurestack' (highlighted with a light blue background), is currently selected.

ResourceGroupName	Name	ResourceId
azurestack	AzureStack-6748505a-e4d9-49fe-87c9-863311d0a148	/subscriptions/454b
azurestack	AzureStack-703031f7-0552-428c-baad-ccb0929b98b4	/subscriptions/454b
azurestack	AzureStack-7aa5713a-53a8-4072-b2f9-d51bb23a5cae	/subscriptions/454b
azurestack	AzureStack-7abb111f-447a-4620-9287-204f05889e67	/subscriptions/454b
azurestack	AzureStack-7c51f40e-9021-44c6-ba29-0599544de6c1	/subscriptions/454b
azurestack	AzureStack-842ad585-1792-4d96-b22b-9f2fe57e26c2	/subscriptions/454b

You should now see a second table listing all the marketplace items available for download. Select the item that you want to download and make a note of the **Version**. You can hold the **Ctrl** key to select multiple images.

Please select product(s) to export

Filter

+ Add criteria ▾

Name	Publisher	Type	Version	ResourceId
CloudLink SecureVM 6.6 BYOL	Dell EMC	virtualMachine	6.6.1	/subscriptions/454b15
CloudLink SecureVM 6.6 Solution BYOL	Dell EMC	virtualMachine	6.6.1	/subscriptions/454b15
CloudLink SecureVM 6.7 BYOL	Dell EMC	virtualMachine	6.7.1	/subscriptions/454b15
CloudLink SecureVM 6.7 Solution BYOL	Dell EMC	virtualMachine	1.0.0	/subscriptions/454b15
CloudLink SecureVM 6.8 BYOL	Dell EMC	virtualMachine	6.8.0	/subscriptions/454b15
CloudLink SecureVM 6.8 Solution BYOL	Dell EMC	virtualMachine	1.0.0	/subscriptions/454b15
CloudLink SecureVM 6.9 BYOL	Dell EMC	virtualMachine	6.9.0	/subscriptions/454b15
CloudLink SecureVM 6.9 Solution BYOL	Dell EMC	virtualMachine	1.0.0	/subscriptions/454b15
CloudLink SecureVM Agent	Dell EMC	virtualMachineExtension	6.8	/subscriptions/454b15
CloudLink SecureVM Agent	Dell EMC	virtualMachineExtension	6.8	/subscriptions/454b15
CloudLink SecureVM Agent for Linux	Dell EMC	virtualMachineExtension	6.0	/subscriptions/454b15
CloudLink SecureVM Agent for Windows	Dell EMC	virtualMachineExtension	6.5	/subscriptions/454b15
CMS Made Simple Certified by Bitnami	Bitnami	virtualMachine	2.2.1905210741	/subscriptions/454b15
Commvault Trial	Commvault	virtualMachine	11.13.1	/subscriptions/454b15
Composr Certified by Bitnami	Bitnami	virtualMachine	10.0.1905152221	/subscriptions/454b15
concrete5 Certified by Bitnami	Bitnami	virtualMachine	8.5.1905211806	/subscriptions/454b15
Coppermine Certified by Bitnami	Bitnami	virtualMachine	1.6.1905210742	/subscriptions/454b15
CoreOS Linux (Stable)	CoreOS	virtualMachine	1465.8.0	/subscriptions/454b15
CouchDB	Bitnami	virtualMachine	2.0.2	/subscriptions/454b15
CouchDB Certified by Bitnami	Bitnami	virtualMachine	2.3.1905152218	/subscriptions/454b15
Custom Script Extension	Microsoft Corp.	virtualMachineExtension	1.9.3	/subscriptions/454b15
Custom Script for Linux	Microsoft Corp.	virtualMachineExtension	1.5.2.2	/subscriptions/454b15
Custom Script for Linux 2.0	Microsoft Corp.	virtualMachineExtension	2.0.6	/subscriptions/454b15
Data Box Edge/Data Box Gateway	Microsoft Corp.	resourceProvider	1.0.5	/subscriptions/454b15
Data Box Gateway Virtual Device	Microsoft	virtualMachine	1.0.2001	/subscriptions/454b15
Debian 8 "Jessie"	credativ	virtualMachine	8.0.20190806	/subscriptions/454b15
Debian 9 "Stretch"	credativ	virtualMachine	9.0.201805160	/subscriptions/454b15
Debian 9 "Stretch"	credativ	virtualMachine	9.0.201807160	/subscriptions/454b15
Dell EMC Data Domain Virtual Edition 3.1 - 6.1.0.X	Dell EMC	virtualMachine	6.1.0110	/subscriptions/454b15
Discourse Certified by Bitnami	Bitnami	virtualMachine	2.2.1905152216	/subscriptions/454b15
Django Certified by Bitnami	Bitnami	virtualMachine	2.2.1905160606	/subscriptions/454b15
DokuWiki	Bitnami	virtualMachine	201702192.0.0	/subscriptions/454b15
Dolibarr Certified by Bitnami	Bitnami	virtualMachine	9.0.1905160607	/subscriptions/454b15

You can also filter the list of images by using the **Add criteria** option.

Please select product(s) to export

Filter

+ Add criteria ▾ **Clear All**

and Name **contains** AKS

Name	Publisher	Type	Version	ResourceId
AKS Base Ubuntu 16.04-LTS Image Distro, August 2019	Azure Kubernetes Service	virtualMachine	2019.08.09	/subscriptions/454b
AKS Base Ubuntu 16.04-LTS Image Distro, August 2019	Azure Kubernetes Service	virtualMachine	2019.08.21	/subscriptions/454b
AKS Base Ubuntu 16.04-LTS Image Distro, July 2019	Azure Kubernetes Service	virtualMachine	2019.07.30	/subscriptions/454b
AKS Base Ubuntu 16.04-LTS Image Distro, October 2019	Azure Kubernetes Service	virtualMachine	2019.10.24	/subscriptions/454b
AKS Base Ubuntu 16.04-LTS Image Distro, September 2019	Azure Kubernetes Service	virtualMachine	2019.09.19	/subscriptions/454b

Once you've made your selections, select OK.

- The IDs for the marketplace items you've selected for download are saved in the `$products` variable. Use the command below to begin downloading the selected items. Replace the destination folder path with a location to store the files you download from Azure Marketplace:

```
$products | Export-AzsMarketplaceItem -RepositoryDir "Destination folder path in quotes"
```

- The time that the download takes depends on the size of the item. After the download completes, the item is available in the folder that you specified in the script. The download includes a VHD file (for virtual machines), or a .zip file (for virtual machine extensions and resource providers). It might also include a gallery package in the .azpkg format, which is a .zip file.
- If the download fails, you can try again by re-running the following PowerShell cmdlet:

```
$products | Export-AzsMarketplaceItem -RepositoryDir "Destination folder path in quotes"
```

9. You should also export the **Azs.Syndication.Admin** module locally so that you can copy it over to the machine from which you are importing marketplace items to Azure Stack Hub.

NOTE

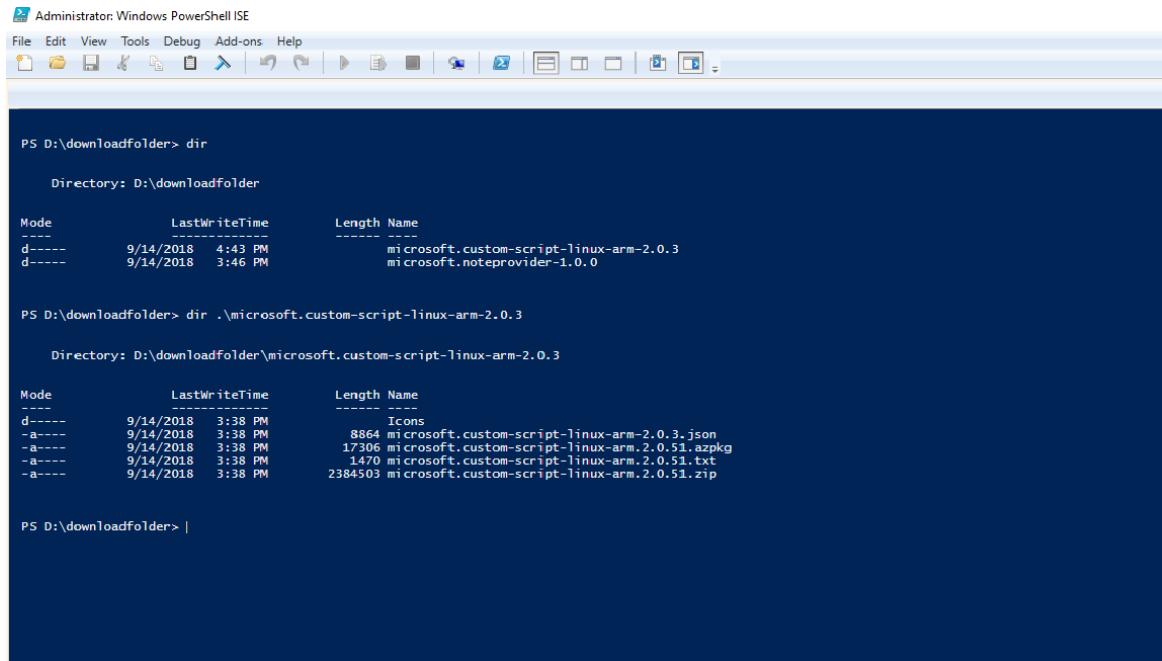
The destination folder for exporting this module should be different from the location to which you have exported the marketplace items.

```
Save-Package -ProviderName NuGet -Source https://www.powershellgallery.com/api/v2 -Name Azs.Syndication.Admin -Path "Destination folder path in quotes" -Force
```

Import the download and publish to Azure Stack Hub Marketplace using PowerShell

1. You must move the files that you have [previously downloaded](#) locally to a machine that has connectivity to your Azure Stack Hub environment. The marketplace syndication tool must also be available to your Azure Stack Hub environment because you need to use the tool to perform the import operation.

The following image shows a folder structure example. **D:\downloadfolder** contains all the downloaded marketplace items. Each subfolder is a marketplace item (for example, **microsoft.custom-script-linux-arm-2.0.3**), named by the product ID. Inside each subfolder is the marketplace item's downloaded content.



Administrator: Windows PowerShell ISE

```
PS D:\downloadfolder> dir

Directory: D:\downloadfolder

Mode LastWriteTime Length Name
---- -- -- -- --
d--- 9/14/2018 4:43 PM ----- microsoft.custom-script-linux-arm-2.0.3
d--- 9/14/2018 3:46 PM ----- microsoft.noteprovider-1.0.0

PS D:\downloadfolder> dir .\microsoft.custom-script-linux-arm-2.0.3

Directory: D:\downloadfolder\microsoft.custom-script-linux-arm-2.0.3

Mode LastWriteTime Length Name
---- -- -- -- --
d--- 9/14/2018 3:38 PM ----- Icons
a--- 9/14/2018 3:38 PM 8864 microsoft.custom-script-linux-arm-2.0.3.json
a--- 9/14/2018 3:38 PM 17306 microsoft.custom-script-linux-arm-2.0.51.azurepowershell.ps1
a--- 9/14/2018 3:38 PM 1470 microsoft.custom-script-linux-arm-2.0.51.txt
a--- 9/14/2018 3:38 PM 2384503 microsoft.custom-script-linux-arm-2.0.51.zip

PS D:\downloadfolder> |
```

2. Follow the instructions in [this article](#) to configure the Azure Stack Hub Operator PowerShell session.
3. Login to your Azure Stack Hub with an identity that has owner access to the "Default Provider Subscription".
4. Import the syndication module and then launch the marketplace syndication tool by running the following script:

```
Import-AzsMarketplaceItem -RepositoryDir "Source folder path in quotes"
```

5. After the script successfully completes, the marketplace items should be available in Azure Stack Hub

Marketplace.

Add a custom VM image to Azure Stack Hub

8 minutes to read • [Edit Online](#)

In Azure Stack Hub, you can add your custom virtual machine (VM) image to the marketplace and make it available to your users. You can add VM images to the Azure Stack Hub Marketplace through the administrator portal or Windows PowerShell. Use either an image from the global Azure Marketplace as a base for your custom image, or your create your own using Hyper-V.

Step 1: Create the custom VM image

Windows

Create a custom generalized VHD.

If the VHD is from outside Azure, follow the steps in [Upload a generalized VHD and use it to create new VMs in Azure](#) to correctly **Sysprep** your VHD and make it generalized.

If the VHD is from Azure, prior to generalizing the VM, make sure of the following:

- When you provision the VM on Azure, use PowerShell and provision it without the `-ProvisionVMAgent` flag.
- Remove all VM extensions using the **Remove-AzureRmVMExtension** cmdlet from the VM before generalizing the VM in Azure. You can find which VM extensions are installed by going to `Windows (C:) > WindowsAzure > Logs > Plugins`.

```
Remove-AzureRmVMExtension -ResourceGroupName winvrmrg1 -VMName windowsvm -Name "CustomScriptExtension"
```

Follow the instructions in [this article](#) to correctly generalize and download the VHD before porting it to Azure Stack Hub.

Linux

If the VHD is from outside Azure, follow the appropriate instructions to generalize the VHD:

- [CentOS-based Distributions](#)
- [Debian Linux](#)
- [Red Hat Enterprise Linux](#)
- [SLES or openSUSE](#)
- [Ubuntu Server](#)

If the VHD is from Azure, follow these instructions to generalize and download the VHD:

1. Stop the **waagent** service:

```
sudo waagent -force -deprovision  
export HISTSIZE=0  
logout
```

Keep in mind the Azure Linux Agent versions that work with Azure Stack Hub [as documented here](#). Make sure that the sysprepped image has an Azure Linux agent version that is compatible with Azure Stack Hub.

2. Stop deallocate the VM.
3. Download the VHD.

- a. To download the VHD file, you need to generate a shared access signature (SAS) URL. When the URL is generated, an expiration time is assigned to the URL.
- b. On the menu of the blade for the VM, select **Disks**.
- c. Select the operating system disk for the VM, and then select **Disk Export**.
- d. Set the expiration time of the URL to 36000.
- e. Select **Generate URL**.
- f. Generate the URL.
- g. Under the URL that was generated, select **Download the VHD file**.
- h. You might need to select **Save** in the browser to start the download. The default name for the VHD file is *abcd*.

Considerations

Before you upload the image, it's important to consider the following:

- Azure Stack Hub only supports generation one (1) VM in the fixed disk VHD format. The fixed-format structures the logical disk linearly within the file, so that disk offset *X* is stored at blob offset *X*. A small footer at the end of the blob describes the properties of the VHD. To confirm if your disk is fixed, use the **Get-VHD** PowerShell cmdlet.
- Azure Stack Hub does not support dynamic disk VHDs.

Step 2: Upload the VM Image to a storage account

1. [Install PowerShell for Azure Stack Hub](#).
2. Sign in to Azure Stack Hub as an operator. For instructions, see [Sign in to Azure Stack Hub as an operator](#).
3. Images must be able to be referenced by a blob storage URI. Prepare a Windows or Linux operating system image in VHD format (not VHDX), and then upload the image to a storage account in Azure Stack Hub.
 - If the VHD is in Azure, you can use a tool such as [Azcopy](#) to directly transfer the VHD between an Azure and your Azure Stack Hub storage account if you are running on a connected Azure Stack Hub.
 - On a disconnected Azure Stack Hub, if your VHD is in Azure, you will need to download the VHD to a machine that has connectivity to both Azure and Azure Stack Hub. Then you copy the VHD to this machine from Azure before you transfer the VHD to Azure Stack Hub using any of the common [storage data transfer tools](#) that can be used across Azure and Azure Stack Hub.

One such tool used in this example is the Add-AzureRmVhd command to upload a VHD to a storage account in the Azure Stack Hub Administrator portal.

```
Add-AzureRmVhd -Destination "https://bash.blob.redmond.azurestack.com/sample/vhdtestingmgd.vhd"
-LocalFilePath "C:\vhd\vhdtestingmgd.vhd"
```

4. Make a note of the blob storage URI where you upload the image. The blob storage URI has the following format: <*storageAccount*>/<*blobContainer*>/<*targetVHDName*>.vhd.
5. To make the blob anonymously accessible, go to the storage account blob container where the VM image VHD was uploaded. Select **Blob**, and then select **Access policy**. Optionally, you can generate a shared access signature for the container, and include it as part of the blob URI. This step makes sure the blob is

available to be used. If the blob isn't anonymously accessible, the VM image will be created in a failed state.

The screenshot shows the Azure Storage account overview for a storage account named 'bash'. The account is in a 'Primary: Available' status, located in 'redmond', and uses 'Locally-redundant storage (LRS)'. It is part of the 'Default Provider Subscription' with ID 'e75ca876-0638-49a1-891f-71b67fdcade'. The 'Tags' section has a note to 'Click here to add tags'. Below this, there are sections for 'Services' (Blobs, Tables, Queues), 'Tools and SDKs' (PowerShell, Azure CLI, .NET, Java, Python, Node.js), and 'Monitoring' (with a dropdown for time intervals: 1 hour, 6 hours, 12 hours, 1 day, 7 days). Metrics are also listed under Monitoring.

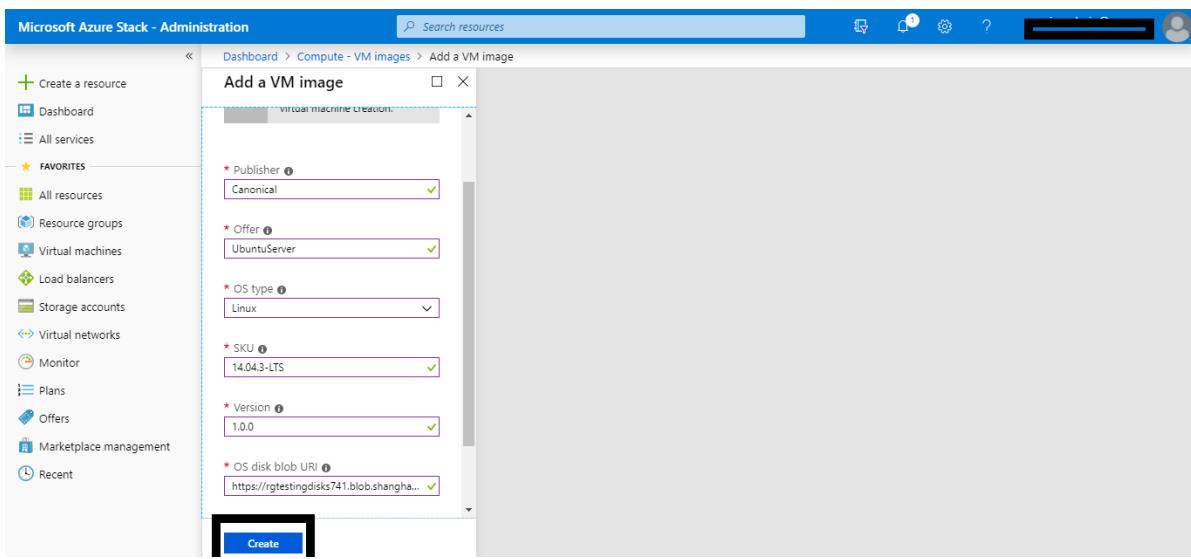
The screenshot shows the 'Blobs' blade for the 'bash' storage account. The left sidebar lists 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Settings' (Access keys, CORS, Configuration, Shared access signature, Properties, Locks), 'Blob service' (selected), and 'Metrics'. The main area shows a list of blobs with a checkbox next to each name. The 'sample' blob is selected, indicated by a checked checkbox. Other blobs listed are 'sample2' and 'sample3'. At the top of the blade are buttons for '+ Container', 'Refresh', 'Delete', and 'Change access level'.

Step 3, Option 1: Add the VM Image as an Azure Stack Hub operator using the portal

1. Sign in to Azure Stack Hub as operator. In the menu, select **All services > Compute** under **VM Images > Add.**

OS TYPE	STATUS	PUBLISHER	OFFER	SKU	VERSION	...
Linux	Succeeded	Bitnami	CassandraCluster	CassandraCluster	1.0.0	...
Linux	Succeeded	Bitnami	ElasticsearchClust...	ElasticsearchClust...	1.0.0	...
Linux	Succeeded	Bitnami	JenkinsCICluster	JenkinsCICluster	1.0.0	...
Linux	Succeeded	Bitnami	KafkaCluster	KafkaCluster	1.0.0	...
Linux	Succeeded	Bitnami	MariaDBwithRepli...	MariaDBwithRepli...	1.0.0	...
Linux	Succeeded	Bitnami	MemcachedMulti...	MemcachedMulti...	1.0.0	...
Linux	Succeeded	Bitnami	MoodleMultiTierS...	MoodleMultiTierS...	1.0.0	...
Linux	Succeeded	Bitnami	MySQLwithReplica...	MySQLwithReplica...	1.0.0	...
Linux	Succeeded	Bitnami	NodeJSCluster	NodeJSCluster	1.0.0	...
Linux	Succeeded	Bitnami	PostgreSQLwithRe...	PostgreSQLwithRe...	1.0.0	...
Linux	Succeeded	Bitnami	RabbitMQCluster	RabbitMQCluster	1.0.0	...
Linux	Succeeded	Bitnami	SolutionTemplate...	SolutionTemplate...	1.0.0	...

2. Under **Create image**, enter the Publisher, Offer, SKU, version and OS disk blob URI. Then, select **Create** to begin creating the VM image.



When the image is successfully created, the VM image status changes to **Succeeded**.

3. When you add an image, it is only available for Azure Resource Manager-based templates and PowerShell deployments. To make an image available to your users as a marketplace item, publish the marketplace item using the steps in the article [Create and publish a Marketplace item](#). Make sure you note the **Publisher**, **Offer**, **SKU**, and **Version** values. You will need them when you edit the Resource Manager template and Manifest.json in your custom .azpkg.

Step 3, Option 2: Add a VM image as an Azure Stack Hub operator using PowerShell

1. [Install PowerShell for Azure Stack Hub](#).
2. Sign in to Azure Stack Hub as an operator. For instructions, see [Sign in to Azure Stack Hub as an operator](#).
3. Open PowerShell with an elevated prompt, and run:

```
Add-AzsPlatformimage -publisher "<publisher>" ` 
    -offer "<Offer>" ` 
    -sku "<SKU>" ` 
    -version "<#,.#.#>" ` 
    -OSType "<OS type>" ` 
    -OSUri "<OS URI>"
```

The **Add-AzsPlatformimage** cmdlet specifies values used by the Azure Resource Manager templates to reference the VM image. The values include:

- **publisher**

For example: Canonical

The **publisher** name segment of the VM image that users use when they deploy the image. Don't include a space or other special characters in this field.

- **offer**

For example: UbuntuServer

The **offer** name segment of the VM image that users use when they deploy the VM image. Don't include a space or other special characters in this field.

- **sku**

For example: 14.04.3-LTS

The **SKU** name segment of the VM Image that users use when they deploy the VM image. Don't include a space or other special characters in this field.

- **version**

For example: `1.0.0`

The version of the VM Image that users use when they deploy the VM image. This version is in the format #.#.#. Don't include a space or other special characters in this field.

- **osType**

For example: `Linux`

The **osType** of the image must be either **Windows** or **Linux**.

- **OSUri**

For example: `https://storageaccount.blob.core.windows.net/vhds/Ubuntu1404.vhd`

You can specify a blob storage URI for an `osDisk`.

For more information, see the PowerShell reference for the [Add-AzsPlatformImage](#) cmdlet.

- When you add an image, it is only available for Azure Resource Manager-based templates and PowerShell deployments. To make an image available to your users as a marketplace item, publish the marketplace item using the steps in the article [Create and publish a Marketplace item](#). Make sure you note the **Publisher**, **Offer**, **SKU**, and **Version** values. You will need them when you edit the resource manager template and Manifest.json in your custom .azpkg.

Remove the VM image as an Azure Stack Hub operator using the portal

- Open the Azure Stack Hub [administrator portal](#).
- If the VM image has an associated Marketplace item, select **Marketplace management**, and then select the VM marketplace item you want to delete.
- If the VM image does not have an associated Marketplace item, navigate to **All services > Compute > VM Images**, and then select the ellipsis (...) next to the VM image.
- Select **Delete**.

Remove a VM image as an Azure Stack Hub operator using PowerShell

When you no longer need the VM image that you uploaded, you can delete it from the Marketplace by using the following cmdlet:

- [Install PowerShell for Azure Stack Hub](#).
- Sign in to Azure Stack Hub as an operator.
- Open PowerShell with an elevated prompt, and run:

```
Remove-AzsPlatformImage `  
-publisher "<Publisher>" `  
-offer "<Offer>" `  
-sku "<SKU>" `  
-version "<Version>" `
```

The **Remove-AzsPlatformImage** cmdlet specifies values used by the Azure Resource Manager templates to reference the VM image. The values include:

- **publisher**

For example: `Canonical`

The **publisher** name segment of the VM image that users use when they deploy the image. Don't

include a space or other special characters in this field.

- **offer**

For example: `UbuntuServer`

The **offer** name segment of the VM image that users use when they deploy the VM image. Don't include a space or other special characters in this field.

- **sku**

For example: `14.04.3-LTS`

The **SKU** name segment of the VM Image that users use when they deploy the VM image. Don't include a space or other special characters in this field.

- **version**

For example: `1.0.0`

The version of the VM Image that users use when they deploy the VM image. This version is in the format `#.#.#`. Don't include a space or other special characters in this field.

For more info about the **Remove-AzsPlatformImage** cmdlet, see the Microsoft PowerShell [Azure Stack Hub Operator module documentation](#).

Next steps

- [Create and publish a custom Azure Stack Hub Marketplace item](#)
- [Provision a virtual machine](#)

Create and publish a custom Azure Stack Hub Marketplace item

7 minutes to read • [Edit Online](#)

Every item published to the Azure Stack Hub Marketplace uses the Azure Gallery Package (.azpkg) format. The *Azure Gallery Packager* tool enables you to create a custom Azure Gallery package that you can upload to the Azure Stack Hub Marketplace, which can then be downloaded by users. The deployment process uses an Azure Resource Manager template.

Marketplace items

The examples in this article show how to create a single VM Marketplace offer, of type Windows or Linux.

Create a Marketplace item

IMPORTANT

Before creating the VM marketplace item, upload the custom VM image to the Azure Stack Hub portal, following the instructions in [Add a VM image to Azure Stack Hub](#). Then, follow the instructions in this article to package the image (create an .azpkg) and upload it to the Azure Stack Hub Marketplace.

To create a custom marketplace item, do the following:

1. Download the [Azure Gallery Packager tool](#) and the sample Azure Stack Hub gallery package. This download includes custom VM templates. Extract the .zip file, and under the folder **Custom VMs**, you can use either the Linux or the Windows templates that are available. You can decide to re-use the pre-made templates and modify the respective parameters with the product details of the item that you will show on your Azure Stack Hub portal. Or, you can simply re-use the .azpkg file available and skip the following steps to customize your own gallery package.
2. Create an Azure Resource Manager template or use our sample templates for Windows/Linux. These sample templates are provided in the packager tool .zip file you downloaded in step 1. You can either use the template and change the text fields, or you can download a pre-configured template from GitHub. For more information about Azure Resource Manager templates, see [Azure Resource Manager templates](#).
3. The Gallery package should contain the following structure:

Name	Type
_rels	File folder
DeploymentTemplates	File folder
icons	File folder
strings	File folder
Manifest	JSON File
UIDefinition	JSON File

The deployment templates file structure appears as follows:

_rels	File folder
createuidefinition	JSON File
DefaultTemplate	JSON File

4. Replace the following highlighted values (those with numbers) in the Manifest.json template with the value that you provided when [uploading your custom image](#).

NOTE

Never hard code any secrets such as product keys, password, or any customer identifiable information in the Azure Resource Manager template. Template JSON files are accessible without the need for authentication once published in the gallery. Store all secrets in [Key Vault](#) and call them from within the template.

The following template is a sample of the Manifest.json file:

```
{
    "$schema": "https://gallery.azure.com/schemas/2015-10-01/manifest.json#",
    "name": "Test", (1)
    "publisher": "<Publisher name>", (2)
    "version": "<Version number>", (3)
    "displayName": "ms-resource:displayName", (4)
    "publisherDisplayName": "ms-resource:publisherDisplayName", (5)
    "publisherLegalName": "ms-resource:publisherDisplayName", (6)
    "summary": "ms-resource:summary",
    "longSummary": "ms-resource:longSummary",
    "description": "ms-resource:description",
    "longDescription": "ms-resource:description",
    "uiDefinition": {
        "path": "UIDefinition.json" (7)
    },
    "links": [
        { "displayName": "ms-resource:documentationLink", "uri": "http://go.microsoft.com/fwlink/?LinkId=532898" }
    ],
    "artifacts": [
        {
            "name": "<Template name>",
            "type": "Template",
            "path": "DeploymentTemplates\\<Template name>.json", (8)
            "isDefault": true
        }
    ],
    "categories": [ (9)
        "Custom",
        "<Template name>"
    ],
    "images": [
        {
            "context": "ibiza",
            "items": [
                {
                    "id": "small",
                    "path": "icons\\Small.png", (10)
                    "type": "icon"
                },
                {
                    "id": "medium",
                    "path": "icons\\Medium.png",
                    "type": "icon"
                },
                {
                    "id": "large",
                    "path": "icons\\Large.png",
                    "type": "icon"
                },
                {
                    "id": "wide",
                    "path": "icons\\Wide.png",
                    "type": "icon"
                }
            ]
        }
    ]
}
```

The following list explains the preceding numbered values in the example template:

- (1) – The name of the offer.
- (2) – The name of the publisher, without a space.
- (3) – The version of your template, without a space.
- (4) – The name that customers see.
- (5) – The publisher name that customers see.
- (6) – The publisher legal name.

- (7) – The path to where your **UIDefinition.json** file is stored.
 - (8) – The path and the name of your JSON main template file.
 - (9) – The names of the categories in which this template is displayed.
 - (10) – The path and name for each icon.
5. For all fields referring to **ms-resource**, you must change the appropriate values inside the **strings/resources.json** file:
- ```
{
 "displayName": "<OfferName.PublisherName.Version>",
 "publisherDisplayName": "<Publisher name>",
 "summary": "Create a simple VM",
 "longSummary": "Create a simple VM and use it",
 "description": "<p>This is just a sample of the type of description you could create for your gallery item!</p><p>This is a second paragraph.</p>",
 "documentationLink": "Documentation"
}
```
- 
6. To ensure that the resource can be deployed successfully, test the template with the [Azure Stack Hub APIs](#).
7. If your template relies on a virtual machine (VM) image, follow the instructions to [add a VM image to Azure Stack Hub](#).
8. Save your Azure Resource Manager template in the **/Contoso.TodoList/DeploymentTemplates/** folder.
9. Choose the icons and text for your Marketplace item. Add icons to the **Icons** folder, and add text to the **resources** file in the **Strings** folder. Use the **small**, **medium**, **large**, and **wide** naming convention for icons. See the [Marketplace item UI reference](#) for a detailed description of these sizes.

#### NOTE

All four icon sizes (small, medium, large, wide) are required for building the Marketplace item correctly.

10. For any further edits to Manifest.json, see [Reference: Marketplace item manifest.json](#).
11. When you finish modifying your files, convert it to an .azpkg file. You perform the conversion using the **AzureGallery.exe** tool and the sample gallery package you downloaded previously. Run the following command:

```
.\AzureGallery.exe package -m c:\<path>\<gallery package name>\manifest.json -o c:\Temp
```

**NOTE**

The output path can be any path you choose, and does not have to be under the C: drive. However, the full path to both the manifest.json file, and the output package, must exist. For example, if the output path is `C:\<path>\galleryPackageName.azpkg`, the folder `c:\<path>` must exist.

## Publish a Marketplace item

1. Use PowerShell or Azure Storage Explorer to upload your Marketplace item (.azpkg) to Azure Blob storage. You can upload to local Azure Stack Hub storage or upload to Azure Storage, which is a temporary location for the package. Make sure that the blob is publicly accessible.
2. To import the gallery package into Azure Stack Hub, the first step is to remotely connect (RDP) to the client VM, in order to copy the file you just created to your Azure Stack Hub.
3. Add a context:

```
$ArmEndpoint = "https://adminmanagement.local.azurestack.external"
Add-AzureRMEnvironment -Name "AzureStackAdmin" -ArmEndpoint $ArmEndpoint
Add-AzureRmAccount -EnvironmentName "AzureStackAdmin"
```

4. Run the following script to import the resource into your gallery:

```
Add-AzsGalleryItem -GalleryItemUri `
https://sample.blob.core.windows.net/<temporary blob name>/<offerName.publisherName.version>.azpkg -
Verbose
```

5. Verify that you have a valid Storage account that is available to store your item. You can get the `GalleryItemURI` value from the Azure Stack Hub administrator portal. Select **Storage account -> Blob Properties -> URL**, with the extension .azpkg. The storage account is only for temporary use, in order to publish to the marketplace.

After completing your gallery package and uploading it using **Add-AzsGalleryItem**, your custom VM should now appear on the Marketplace as well as in the **Create a resource** view. Note that the custom gallery package is not visible in **Marketplace Management**.

- Once your item has been successfully published to the marketplace, you can delete the content from the storage account.

**Caution**

All default gallery artifacts and your custom gallery artifacts are now accessible without authentication under the following URLs:

```
https://adminportal.[Region].[external FQDN]:30015/artifact/20161101/[Template Name]/DeploymentTemplates/Template.json
```

```
https://portal.[Region].[external FQDN]:30015/artifact/20161101/[Template Name]/DeploymentTemplates/Template.json
```

- You can remove a Marketplace item by using the **Remove-AzureRMGalleryItem** cmdlet. For example:

```
Remove-AzsGalleryItem -Name <Gallery package name> -Verbose
```

**NOTE**

The Marketplace UI may show an error after you remove an item. To fix the error, click **Settings** in the portal. Then, select **Discard modifications** under **Portal customization**.

## Reference: Marketplace item manifest.json

### Identity information

| NAME      | REQUIRED | TYPE   | CONSTRAINTS  | DESCRIPTION |
|-----------|----------|--------|--------------|-------------|
| Name      | X        | String | [A-Za-z0-9]+ |             |
| Publisher | X        | String | [A-Za-z0-9]+ |             |
| Version   | X        | String | SemVer v2    |             |

### Metadata

| NAME                 | REQUIRED | TYPE   | CONSTRAINTS                     | DESCRIPTION                                                                                   |
|----------------------|----------|--------|---------------------------------|-----------------------------------------------------------------------------------------------|
| DisplayName          | X        | String | Recommendation of 80 characters | The portal might not display your item name correctly if it's longer than 80 characters.      |
| PublisherDisplayName | X        | String | Recommendation of 30 characters | The portal might not display your publisher name correctly if it's longer than 30 characters. |
| PublisherLegalName   | X        | String | Maximum of 256 characters       |                                                                                               |
| Summary              | X        | String | 60 to 100 characters            |                                                                                               |
| LongSummary          | X        | String | 140 to 256 characters           | Not yet applicable in Azure Stack Hub.                                                        |
| Description          | X        | HTML   | 500 to 5,000 characters         |                                                                                               |

## Images

The Marketplace uses the following icons:

| NAME       | WIDTH  | HEIGHT | NOTES           |
|------------|--------|--------|-----------------|
| Wide       | 255 px | 115 px | Always required |
| Large      | 115 px | 115 px | Always required |
| Medium     | 90 px  | 90 px  | Always required |
| Small      | 40 px  | 40 px  | Always required |
| Screenshot | 533 px | 324 px | Always required |

## Categories

Each Marketplace item should be tagged with a category that identifies where the item appears on the portal UI. You can choose one of the existing categories in Azure Stack Hub (**Compute**, **Data + Storage**, and so on) or choose a new one.

## Links

Each Marketplace item can include various links to additional content. The links are specified as a list of names and URIs:

| NAME        | REQUIRED | TYPE   | CONSTRAINTS               | DESCRIPTION |
|-------------|----------|--------|---------------------------|-------------|
| DisplayName | X        | String | Maximum of 64 characters. |             |

| NAME | REQUIRED | TYPE | CONSTRAINTS | DESCRIPTION |
|------|----------|------|-------------|-------------|
| Uri  | X        | URI  |             |             |

## Additional properties

In addition to the preceding metadata, Marketplace authors can provide custom key/value pair data in the following form:

| NAME        | REQUIRED | TYPE   | CONSTRAINTS               | DESCRIPTION |
|-------------|----------|--------|---------------------------|-------------|
| DisplayName | X        | String | Maximum of 25 characters. |             |
| Value       | X        | String | Maximum of 30 characters. |             |

## HTML sanitization

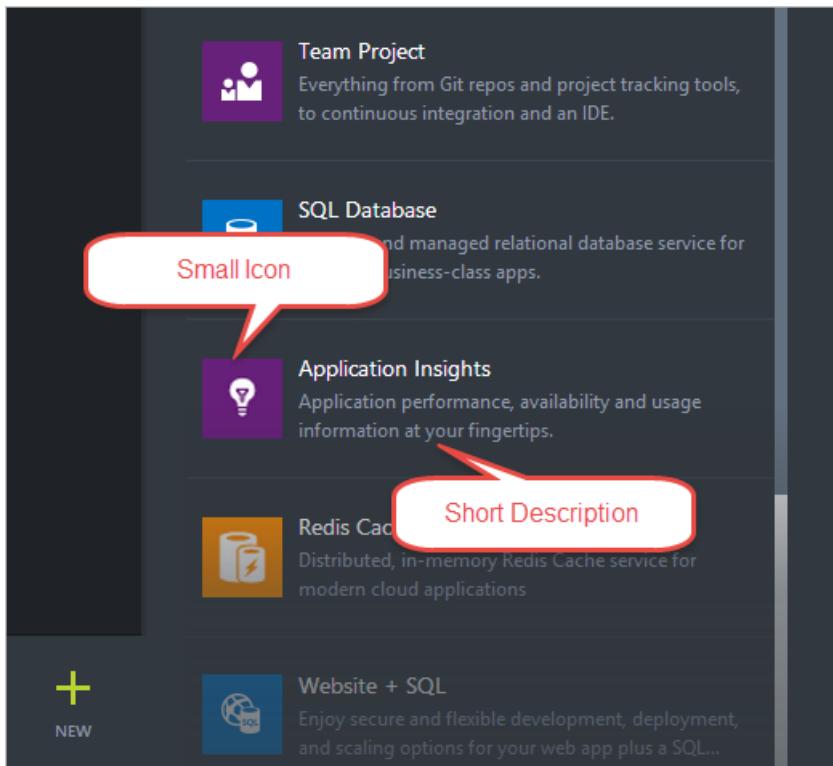
For any field that allows HTML, the following elements and attributes are allowed:

```
h1, h2, h3, h4, h5, p, ol, ul, li, a[target|href], br, strong, em, b, i
```

## Reference: Marketplace item UI

Icons and text for Marketplace items as seen in the Azure Stack Hub portal are as follows.

### Create blade



### Marketplace item details blade

**Redis Cache (Preview)**

**Microsoft**

**Medium Icon**

**Publisher Display Name**

**Display Name**

Microsoft Azure Redis Cache (preview) is based on the popular open source Redis Cache. It gives you the power of the Redis engine, managed by Microsoft. Users get the best of both worlds – the Redis ecosystem, and reliable hosting and monitoring from Microsoft.

Microsoft Azure Redis Cache (Preview) is available in two tiers:

- Basic – A single Cache node.
- Standard – A replicated Cache (Two nodes, Master and a Slave)

Microsoft Azure Redis Cache helps your application stay responsive even as user load increases. It does so, by leveraging the inherent capabilities of the Redis engine. This separate distributed cache scales independently for more efficient use of compute resources in your environment.

**Screenshot**

The screenshot shows the Azure portal interface for the Redis Cache service 'dfcachecow1'. On the left, there's a summary section with a tree view of the service structure. On the right, there are two main sections: 'Monitoring' and 'Metrics'. The 'Monitoring' section displays line charts for Cache Hits, Cache Misses, and Get Commands over time (TUE 29 MAY 03 SAT 03 MON 03). The 'Metrics' section provides detailed statistics for various Redis commands: Cache Hits (132.28 k), Cache Misses (28.35 k), Get Commands (160.63 k), Set Commands (0.04 k), Evicted Keys (0 k), and Expired Keys (0 k). Below the monitoring section is a table of metric names with their average, minimum, and maximum values.

| Metric Name  | Avg    | Min | Max      |
|--------------|--------|-----|----------|
| Cache Hits   | 0.01 k | 0 k | 132.28 k |
| Cache Misses | 0 k    | 0 k | 28.35 k  |
| Get Commands | 0.01 k | 0 k | 160.63 k |
| Set Commands | 0.04 k | 0 k | 493.56 k |
| Evicted Keys | 0 k    | 0 k | 0 k      |
| Expired Keys | 0 k    | 0 k | 0 k      |

**PUBLISHER**

**Microsoft**

**Publisher Display Name**

**USEFUL LINKS**

[Service Overview](#)

[Documentation](#)

[Pricing details](#)

**Links**

**Create**

## Next steps

- [Azure Stack Hub Marketplace overview](#)
- [Download Marketplace items](#)
- [Format and structure of Azure Resource Manager templates](#)

# Guest operating systems supported on Azure Stack Hub

2 minutes to read • [Edit Online](#)

## Windows

Azure Stack Hub supports the Windows guest operating systems listed in the following table:

| OPERATING SYSTEM             | DESCRIPTION                 | AVAILABLE IN AZURE STACK HUB MARKETPLACE                |
|------------------------------|-----------------------------|---------------------------------------------------------|
| Windows Server, version 1709 | 64-bit                      | Core with containers                                    |
| Windows Server 2019          | 64-bit                      | Datacenter, Datacenter core, Datacenter with containers |
| Windows Server 2016          | 64-bit                      | Datacenter, Datacenter core, Datacenter with containers |
| Windows Server 2012 R2       | 64-bit                      | Datacenter                                              |
| Windows Server 2012          | 64-bit                      | Datacenter                                              |
| Windows Server 2008 R2 SP1   | 64-bit                      | Datacenter                                              |
| Windows Server 2008 SP2      | 64-bit                      | Bring your own image                                    |
| Windows 10 (see note 1)      | 64-bit, Pro, and Enterprise | Bring your own image                                    |

### NOTE

To deploy Windows 10 client operating systems on Azure Stack Hub, you must have [Windows per-user licensing](#) or purchase through a Qualified Multitenant Hoster ([QMTH](#)).

Marketplace images are available for pay-as-you-use or BYOL (EA/SPLA) licensing. Use of both on a single Azure Stack Hub instance isn't supported. During deployment, Azure Stack Hub injects a suitable version of the guest agent into the image.

Datacenter editions are available in Azure Stack Hub Marketplace for downloading; customers can bring their own server images including other editions. Windows client images aren't available in Azure Stack Hub Marketplace.

## Linux

Linux distributions listed as available in Azure Stack Hub Marketplace include the necessary Windows Azure Linux Agent (WALA). If you bring your own image to Azure Stack, follow the guidelines in [Add Linux images to Azure Stack](#).

**NOTE**

Custom images should be built with the latest public WALA version (on the 1903 Azure Stack Hub build and above, or with the 1901/1902 hotfix), or with version 2.2.20. Versions before 2.2.20 and between 2.2.21 and 2.2.34 (inclusive) may not function properly on Azure Stack Hub. On Azure Stack Hub 1910 and above, all Azure WALA agent versions work with Azure Stack Hub.

[cloud-init](#) is supported on Azure Stack Hub 1910 and above.

| DISTRIBUTION                             | DESCRIPTION | PUBLISHER      | AZURE STACK HUB MARKETPLACE |
|------------------------------------------|-------------|----------------|-----------------------------|
| CentOS-based 6.9                         | 64-bit      | Rogue Wave     | Yes                         |
| CentOS-based 7.5                         | 64-bit      | Rogue Wave     | Yes                         |
| CentOS-based 7.3                         | 64-bit      | Rogue Wave     | Yes                         |
| ClearLinux                               | 64-bit      | ClearLinux.org | Yes                         |
| CoreOS Linux (Stable)                    | 64-bit      | CoreOS         | Yes                         |
| Debian 8 "Jessie"                        | 64-bit      | credativ       | Yes                         |
| Debian 9 "Stretch"                       | 64-bit      | credativ       | Yes                         |
| Oracle Linux                             | 64-bit      | Oracle         | Yes                         |
| Red Hat Enterprise Linux 7.1 (and later) | 64-bit      | Red Hat        | Bring your own image        |
| SLES 11SP4                               | 64-bit      | SUSE           | Yes                         |
| SLES 12SP3                               | 64-bit      | SUSE           | Yes                         |
| Ubuntu 14.04-LTS                         | 64-bit      | Canonical      | Yes                         |
| Ubuntu 16.04-LTS                         | 64-bit      | Canonical      | Yes                         |
| Ubuntu 18.04-LTS                         | 64-bit      | Canonical      | Yes                         |

For Red Hat Enterprise Linux support information, see [Red Hat and Azure Stack Hub: Frequently Asked Questions](#).

## Next steps

For more information about Azure Stack Hub Marketplace, see the following articles:

- [Download marketplace items](#)
- [Create and publish a marketplace item](#)

# Make virtual machine scale sets available in Azure Stack Hub

3 minutes to read • [Edit Online](#)

Virtual machine scale sets are an Azure Stack Hub compute resource. You can use scale sets to deploy and manage a set of identical virtual machines (VMs). With all VMs configured in the same way, scale sets do not require pre-provisioning of VMs. It is easier to build large-scale services that target big compute, big data, and containerized workloads.

This article guides you through the process of making scale sets available in the Azure Stack Hub Marketplace. After you complete this procedure, your users can add virtual machine scale sets to their subscriptions.

Virtual machine scale sets on Azure Stack Hub are similar to virtual machine scale sets on Azure. For more information, see the following videos:

- [Mark Russinovich talks Azure scale sets](#)
- [Virtual machine scale sets with Guy Bowerman](#)

On Azure Stack Hub, virtual machine scale sets do not support autoscale. You can add more instances to a scale set using Resource Manager templates, CLI, or PowerShell.

## Prerequisites

- **Azure Stack Hub Marketplace:** Register Azure Stack Hub with global Azure to enable the availability of items in the Azure Stack Hub Marketplace. Follow the instructions in [Register Azure Stack Hub with Azure](#).
- **Operating system image:** Before a virtual machine scale set can be created, you must download the VM images for use in the scale set from the [Azure Stack Hub Marketplace](#). The images must already be present before a user can create a new scale set.

## Use the Azure Stack Hub portal

### IMPORTANT

The information in this section applies when you use Azure Stack Hub version 1808 or later.

1. Sign in to the Azure Stack Hub portal. Then, go to **All services**, then **Virtual machine scale sets**, and then under **COMPUTE**, select **Virtual machine scale sets**.

Microsoft Azure Stack

All services

GENERAL (9)

- Dashboard
- All resources
- Recent
- Marketplace
- Portal settings

COMPUTE (5)

- Virtual machines
- Availability sets
- Snapshots

DATA + STORAGE (1)

- Storage accounts

Create a resource

All services

FAVORITES

- Dashboard
- All resources
- Resource groups
- Recent
- Monitor
- Virtual machines

Service Admin CI AZURE STACK

The screenshot shows the Microsoft Azure Stack interface. On the left, there's a sidebar with various navigation links like 'Create a resource', 'All services' (which is selected and highlighted with a red box), and 'FAVORITES'. The main content area is titled 'All services' and contains several sections: 'GENERAL' (9 items), 'COMPUTE' (5 items), and 'DATA + STORAGE' (1 item). Under 'COMPUTE', the 'Virtual machine scale sets' link is also highlighted with a red box. At the top right, there are icons for search, notifications, settings, help, and a user profile labeled 'Service Admin CI AZURE STACK'.

2. Select **Create Virtual machine scale sets**.

Microsoft Azure Stack

Virtual machine scale sets

Subscriptions: cool1808sub

No Virtual machine scale sets to display

Create a virtual machine scale set to deploy and manage a load balanced set of identical Windows or Linux virtual machines. Use autoscale to automatically scale virtual machine resources in and out. [Learn more](#)

Create Virtual machine scale sets

Home > Virtual machine scale sets

Add Edit columns Refresh

NAME STATUS INSTAN... RESOURC... LOCATION SUBSCRIP...

Dashboard All resources Resource groups Recent Monitor Virtual machines

Service Admin CI AZURE STACK

The screenshot shows the 'Virtual machine scale sets' list page. The left sidebar has the same structure as the previous screenshot. The main area shows a message 'No Virtual machine scale sets to display' with a sub-instruction about creating one. A prominent red box highlights the 'Create Virtual machine scale sets' button at the bottom of the page. The top navigation bar includes 'Home > Virtual machine scale sets', 'Add', 'Edit columns', 'Refresh', and filtering options for 'Subscriptions', 'Resource groups', and 'Grouping'.

3. Fill in the empty fields, choose from the dropdowns for **Operating system disk image**, **Subscription**, and **Instance size**. Select **Yes** for **Use managed disks**. Then, click **Create**.

- To see your new virtual machine scale set, go to **All resources**, search for the virtual machine scale set name, and then select its name in the search.

## Update images in a virtual machine scale set

After you create a virtual machine scale set, users can update images in the scale set without the scale set having to be recreated. The process to update an image depends on the following scenarios:

- Virtual machine scale set deployment template specifies **latest** for **version**:

When the `version` is set to **latest** in the `imageReference` section of the template for a scale set, scale-up operations on the scale set use the newest available version of the image for the scale set instances. After a scale-up is complete, you can delete older virtual machine scale sets instances. The values for `publisher`, `offer`, and `sku` remain unchanged.

The following JSON example specifies `latest`:

```
"imageReference": {
 "publisher": "[parameters('osImagePublisher')]",
 "offer": "[parameters('osImageOffer')]",
 "sku": "[parameters('osImageSku')]",
 "version": "latest"
}
```

2. Virtual machine scale set deployment template **does not specify latest** for **version** and specifies a version number instead:

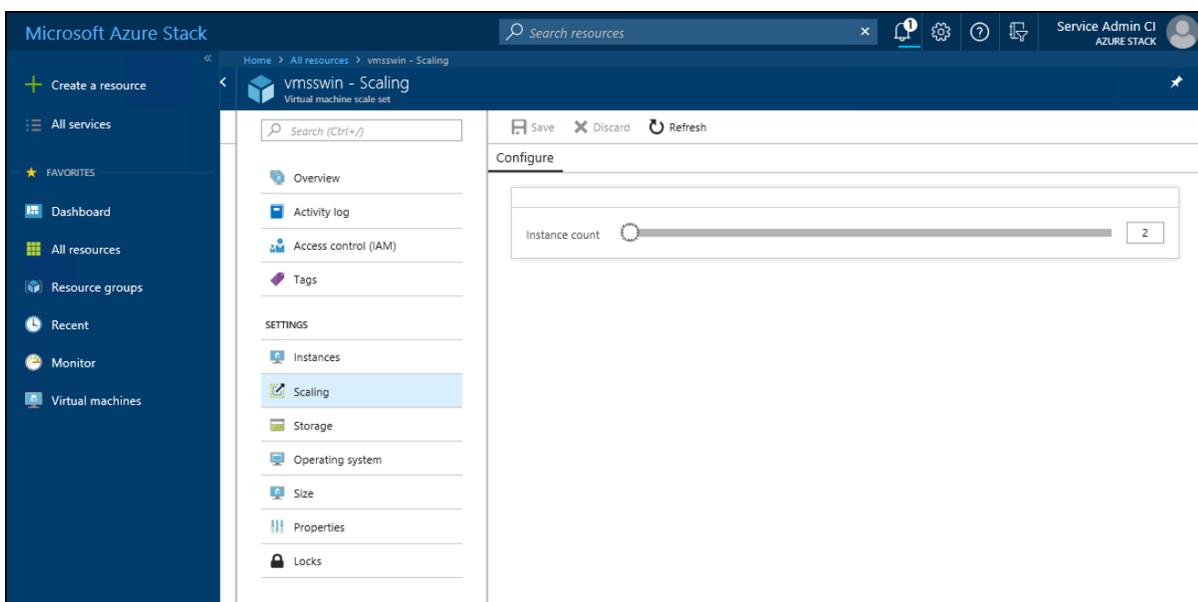
If the Azure Stack operator downloads an image with a newer version (and deletes the older version), the scale set cannot scale up. This is by design, as the image version specified in the scale set template must be available.

For more information, see [operating system disks and images](#).

## Scale a virtual machine scale set

You can scale the size of a virtual machine scale set to make it larger or smaller.

1. In the portal, select your scale set and then select **Scaling**.
2. Use the slide bar to set the new level of scaling for this virtual machine scale set, and then click **Save**.



## Next steps

- [Download marketplace items from Azure to Azure Stack Hub](#)

# Windows Server in Azure Stack Hub Marketplace FAQ

5 minutes to read • [Edit Online](#)

This article answers some frequently asked questions about Windows Server images in the [Azure Stack Hub Marketplace](#).

## Marketplace items

### How do I update to a newer Windows image?

First, determine if any Azure Resource Manager templates refer to specific versions. If so, update those templates, or keep older image versions. It's best to use **version: latest**.

Next, if any virtual machine scale sets refer to a specific version, you should think about whether these will be scaled later, and decide whether to keep older versions. If neither of these conditions apply, delete older images in Azure Stack Hub Marketplace before downloading newer ones. Use marketplace management to delete them if that's how the original was downloaded. Then download the newer version.

### What are the licensing options for Windows Server Marketplace images on Azure Stack Hub?

Microsoft offers two versions of Windows Server images through Azure Stack Hub Marketplace. Only one version of this image can be used in an Azure Stack Hub environment.

- **Pay as you go (PAYG):** These images run the full price Windows meters. Who should use this option: Enterprise Agreement (EA) customers who use the *Consumption billing model*; CSPs who don't want to use SPLA licensing.
- **Bring Your Own License (BYOL):** These images run basic meters. Who should use this option: EA customers with a Windows Server license; CSPs who use SPLA licensing.

Azure Hybrid Use Benefit (AHUB) isn't supported on Azure Stack Hub. Customers who license through the "Capacity" model must use the BYOL image. If you're testing with the Azure Stack Development Kit (ASDK), you can use either of these options.

### What if I downloaded the wrong version to offer my tenants/users?

Delete the incorrect version first through marketplace management. Wait for it to complete (look at the notifications for completion, not the **Marketplace Management** blade). Then download the correct version.

If you download both versions of the image, only the latest version is visible to end customers in Azure Stack Hub Marketplace.

### What if my user incorrectly checked the "I have a license" box in previous Windows builds, and they don't have a license?

You can change the license model attribute to switch from BYOL to the PAYG model by running the following script:

```
$vm= Get-Azurermvm -ResourceGroup "<your RG>" -Name "<your VM>"
$vm.LicenseType = "None"
Update-AzureRmVM -ResourceGroupName "<your RG>" -VM $vm
```

You can check the license type of your VM by running the following commands. If the license model says **Windows\_Server**, you'll be charged for the BYOL price. Otherwise, you'll be charged for the Windows meter per

the PAYG model:

```
$vm | ft Name, VmId,LicenseType,ProvisioningState
```

**What if I have an older image and my user forgot to check the "I have a license" box, or we use our own images and we do have Enterprise Agreement entitlement?**

You can change the license model attribute to the BYOL model by running the following commands:

```
$vm= Get-AzurermVM -ResourceGroup "<your RG>" -Name "<your VM>"
$vm.LicenseType = "Windows_Server"
Update-AzureRmVM -ResourceGroupName "<your RG>" -VM $vm
```

**What about other VMs that use Windows Server, such as SQL or Machine Learning Server?**

These images do apply the **licenseType** parameter, so they're PAYG. You can set this parameter (see the previous FAQ answer). This only applies to the Windows Server software, not to layered products such as SQL, which require you to bring your own license. PAYG licensing doesn't apply to layered software products.

You can only change the **licenseType** property for SQL Server images from Azure Stack Hub Marketplace if the version is XX.X.20190410 or higher. If you're running an older version of the SQL Server images from Azure Stack Hub Marketplace, you can't change the **licenseType** attribute and you must redeploy using the latest SQL Server images from Azure Stack Hub Marketplace.

**I have an Enterprise Agreement (EA) and will be using my EA Windows Server license; how do I make sure images are billed correctly?**

You can add **licenseType: Windows\_Server** in an Azure Resource Manager template. This setting must be added to each virtual machine (VM) resource block.

## Activation

To activate a Windows Server VM on Azure Stack Hub, the following conditions must be true:

- The OEM has set the appropriate BIOS marker on every host system in Azure Stack Hub.
- Windows Server 2012 R2 and Windows Server 2016 must use [Automatic VM Activation](#). Key Management Service (KMS) and other activation services aren't supported on Azure Stack Hub.

**How can I verify that my VM is activated?**

Run the following command from an elevated command prompt:

```
s1mgr /dlv
```

If it's correctly activated, you'll see this clearly indicated and the host name displayed in the `s1mgr` output. Don't depend on watermarks on the display as they might not be up to date, or are showing from a different VM behind yours.

**My VM isn't set up to use AVMA, how can I fix it?**

Run the following command from an elevated command prompt:

```
s1mgr /ipk <AVMA key>
```

See the [Automatic VM Activation](#) article for the keys to use for your image.

**I create my own Windows Server images, how can I make sure they use AVMA?**

It's recommended that you execute the `s1mgr /ipk` command line with the appropriate key before you run the `sysprep` command. Or, include the AVMA key in any Unattend.exe setup file.

**I am trying to use my Windows Server 2016 image created on Azure and it's not activating or using KMS activation.**

Run the `s1mgr /ipk` command. Azure images may not correctly fall back to AVMA, but if they can reach the Azure KMS system, they will activate. It's recommended that you ensure these VMs are set to use AVMA.

**I have performed all of these steps but my VMs are still not activating.**

Contact your hardware supplier to verify that the correct BIOS markers were installed.

**What about earlier versions of Windows Server?**

[Automatic VM Activation](#) isn't supported in earlier versions of Windows Server. You must activate the VMs manually.

## Next steps

For more information, see the following articles:

- [The Azure Stack Hub Marketplace overview](#)
- [Download marketplace items from Azure to Azure Stack Hub](#)

# Add Linux images to the Azure Stack Hub Marketplace

2 minutes to read • [Edit Online](#)

You can deploy Linux virtual machines (VMs) on Azure Stack Hub by adding a Linux-based image into Azure Stack Hub Marketplace. The easiest way to add a Linux image to Azure Stack Hub is through Marketplace Management. These images have been prepared and tested for compatibility with Azure Stack Hub.

## Marketplace Management

To download Linux images from Azure Marketplace, see [Download marketplace items from Azure to Azure Stack Hub](#). Select the Linux images that you want to offer users on your Azure Stack Hub.

There are frequent updates to these images, so check Marketplace Management often to keep up-to-date.

## Prepare your own image

Wherever possible, download the images available through Marketplace Management. These images have been prepared and tested for Azure Stack Hub.

### Azure Linux Agent

The Azure Linux Agent (typically called **WALinuxAgent** or **walinuagent**) is required, and not all versions of the agent work on Azure Stack Hub. Versions between 2.2.21 and 2.2.34 (inclusive) are not supported on Azure Stack Hub. To use the latest agent versions above 2.2.35, apply the 1901 hotfix/1902 hotfix, or update your Azure Stack Hub to the 1903 release (or above). Note that [cloud-init](#) is supported on Azure Stack Hub releases beyond 1910.

| AZURE STACK HUB BUILD  | AZURE LINUX AGENT BUILD       |
|------------------------|-------------------------------|
| 1.1901.0.99 or earlier | 2.2.20                        |
| 1.1902.0.69            | 2.2.20                        |
| 1.1901.3.105           | 2.2.35 or newer               |
| 1.1902.2.73            | 2.2.35 or newer               |
| 1.1903.0.35            | 2.2.35 or newer               |
| Builds after 1903      | 2.2.35 or newer               |
| Not supported          | 2.2.21-2.2.34                 |
| Builds after 1910      | All Azure WALA agent versions |

You can prepare your own Linux image using the following instructions:

- [CentOS-based Distributions](#)
- [Debian Linux](#)
- [Red Hat Enterprise Linux](#)

- [SLES & openSUSE](#)
- [Ubuntu Server](#)

## Cloud-init

[Cloud-init](#) is supported on Azure Stack Hub releases beyond 1910. To use cloud-init to customize your Linux VM, you can use the following PowerShell instructions.

### Step 1: Create a cloud-init.txt file with your cloud-config

Create a file named cloud-init.txt and paste the following cloud configuration:

```
#cloud-config
package_upgrade: true
packages:
 - nginx
 - nodejs
 - npm
write_files:
 - owner: www-data:www-data
 path: /etc/nginx/sites-available/default
 content: |
 server {
 listen 80;
 location / {
 proxy_pass http://localhost:3000;
 proxy_http_version 1.1;
 proxy_set_header Upgrade $http_upgrade;
 proxy_set_header Connection keep-alive;
 proxy_set_header Host $host;
 proxy_cache_bypass $http_upgrade;
 }
 }
 - owner: azureuser:azureuser
 path: /home/azureuser/myapp/index.js
 content: |
 var express = require('express')
 var app = express()
 var os = require('os');
 app.get('/', function (req, res) {
 res.send('Hello World from host ' + os.hostname() + '!')
 })
 app.listen(3000, function () {
 console.log('Hello world app listening on port 3000!')
 })
runcmd:
 - service nginx restart
 - cd "/home/azureuser/myapp"
 - npm init
 - npm install express -y
 - nodejs index.js
```

### Step 2: Reference the cloud-init.txt during the Linux VM deployment

Upload the file to an Azure storage account, Azure Stack Hub storage account, or GitHub repository reachable by your Azure Stack Hub Linux VM. Currently, using cloud-init for VM deployment is only supported on REST, Powershell, and CLI, and doesn't have an associated portal UI on Azure Stack Hub.

You can follow [these](#) instructions to create the Linux VM using powershell, but make sure to reference the cloud-init.txt as a part of the `-CustomData` flag:

```
$VirtualMachine =Set-AzureRmVMOperatingSystem -VM $VirtualMachine `
-Linux `
-ComputerName "MainComputer" `
-Credential $cred -CustomData "#include https://cloudinitstrg.blob.core.windows.net/strg/cloud-init.txt"
```

## Add your image to Marketplace

Follow [Add the image to the Marketplace](#). Make sure that the `OSType` parameter is set to `Linux`.

After you've added the image to the Marketplace, a Marketplace item is created and users can deploy a Linux VM.

## Next steps

- [Download marketplace items from Azure to Azure Stack Hub](#)
- [Azure Stack Hub Marketplace overview](#)

# Prepare a Red Hat-based virtual machine for Azure Stack Hub

18 minutes to read • [Edit Online](#)

In this article, you'll learn how to prepare a Red Hat Enterprise Linux (RHEL) virtual machine (VM) for use in Azure Stack Hub. The versions of RHEL that are covered in this article are 7.1+. The hypervisors for preparation that are covered in this article are Hyper-V, kernel-based virtual machine (KVM), and VMware.

For Red Hat Enterprise Linux support information, see [Red Hat and Azure Stack Hub: Frequently Asked Questions](#).

## Prepare a Red Hat-based VM from Hyper-V Manager

This section assumes that you already have an ISO file from the Red Hat website and have installed the RHEL image to a virtual hard disk (VHD). For more information about how to use Hyper-V Manager to install an operating system image, see [Install the Hyper-V Role and Configure a VM](#).

### RHEL installation notes

- Azure Stack Hub doesn't support the VHDX format. Azure supports only fixed VHD. You can use Hyper-V Manager to convert the disk to VHD format, or you can use the convert-vhd cmdlet. If you use VirtualBox, select **Fixed size** as opposed to the default dynamically allocated option when you create the disk.
- Azure Stack Hub supports only generation 1 VMs. You can convert a generation 1 VM from VHDX to the VHD file format and from dynamically expanding to a fixed-size disk. You can't change a VM's generation. For more information, see [Should I create a generation 1 or 2 VM in Hyper-V?](#).
- The maximum size that's allowed for the VHD is 1,023 GB.
- When you install the Linux operating system, we recommend that you use standard partitions rather than Logical Volume Manager (LVM), which is often the default for many installations. This practice avoids LVM name conflicts with cloned VMs, particularly if you ever need to attach an operating system disk to another identical VM for troubleshooting.
- Kernel support for mounting Universal Disk Format (UDF) file systems is required. At first boot, the UDF-formatted media that's attached to the guest passes the provisioning configuration to the Linux VM. The Azure Linux Agent must mount the UDF file system to read its configuration and provision the VM.
- Don't configure a swap partition on the operating system disk. The Linux Agent can be configured to create a swap file on the temporary resource disk. More information about can be found in the following steps.
- All VHDs on Azure must have a virtual size aligned to 1 MB. When converting from a raw disk to VHD, you must ensure that the raw disk size is a multiple of 1 MB before conversion. More details can be found in the steps below.
- Azure Stack Hub supports cloud-init. [Cloud-init](#) is a widely used approach to customize a Linux VM as it boots for the first time. You can use cloud-init to install packages and write files, or to configure users and security. Because cloud-init is called during the initial boot process, there are no additional steps or required agents to apply your configuration. For instructions on adding cloud-init to your image, see [Prepare an existing Linux Azure VM image for use with cloud-init](#).

### Prepare an RHEL 7 VM from Hyper-V Manager

1. In Hyper-V Manager, select the VM.
2. Select **Connect** to open a console window for the VM.
3. Create or edit the `/etc/sysconfig/network` file, and add the following text:

```
NETWORKING=yes
HOSTNAME=localhost.localdomain
```

4. Create or edit the `/etc/sysconfig/network-scripts/ifcfg-eth0` file, and add the following text as needed:

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
TYPE=Ethernet
USERCTL=no
PEERDNS=yes
IPV6INIT=no
NM_CONTROLLED=no
```

5. Ensure that the network service starts at boot time by running the following command:

```
sudo systemctl enable network
```

6. Register your Red Hat subscription to enable the installation of packages from the RHEL repository by running the following command:

```
sudo subscription-manager register --auto-attach --username=XXX --password=XXX
```

7. Modify the kernel boot line in your grub configuration to include additional kernel parameters for Azure. To make this modification, open `/etc/default/grub` in a text editor, and modify the `GRUB_CMDLINE_LINUX` parameter. For example:

```
GRUB_CMDLINE_LINUX="rootdelay=300 console=ttyS0 earlyprintk=ttyS0 net.ifnames=0"
```

This modification ensures all console messages are sent to the first serial port, which can assist Azure support with debugging issues. This configuration also turns off the new RHEL 7 naming conventions for NICs.

Graphical and quiet boot aren't useful in a cloud environment where we want all the logs to be sent to the serial port. You can leave the `crashkernel` option configured if desired. This parameter reduces the amount of available memory in the VM by 128 MB or more, which might be problematic on smaller VM sizes. We recommend that you remove the following parameters:

```
rhgb quiet crashkernel=auto
```

8. After you're done editing `/etc/default/grub`, run the following command to rebuild the grub configuration:

```
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

9. [Optional after 1910 release] Stop and Uninstall cloud-init:

```
systemctl stop cloud-init
yum remove cloud-init
```

10. Ensure that the SSH server is installed and configured to start at boot time, which is usually the default.

Modify `/etc/ssh/sshd_config` to include the following line:

```
ClientAliveInterval 180
```

11. When creating a custom vhd for Azure Stack Hub, keep in mind that WALinuxAgent version between 2.2.20 and 2.2.35 (both exclusive) don't work on Azure Stack Hub environments before the 1910 release. You can use versions 2.2.20/2.2.35 versions to prepare your image. To use versions above 2.2.35 to prepare your custom image, update your Azure Stack Hub to 1903 release and above or apply the 1901/1902 hotfix.

[Before 1910 release] Follow these instructions to download a compatible WALinuxAgent:

- Download setuptools.

```
wget https://pypi.python.org/packages/source/s/setuptools/setuptools-7.0.tar.gz --no-check-certificate
tar xzf setuptools-7.0.tar.gz
cd setuptools-7.0
```

- Download and unzip the 2.2.20 version of the agent from our GitHub.

```
wget https://github.com/Azure/WALinuxAgent/archive/v2.2.20.zip
unzip v2.2.20.zip
cd WALinuxAgent-2.2.20
```

- Install setup.py.

```
sudo python setup.py install
```

- Restart waagent.

```
sudo systemctl restart waagent
```

- Test if the agent version matches the one you downloaded. For this example, it should be 2.2.20.

```
waagent -version
```

[After 1910 release] Follow these instructions to download a compatible WALinuxAgent:

- The WALinuxAgent package, `WALinuxAgent-<version>`, has been pushed to the Red Hat extras repository. Enable the extras repository by running the following command:

```
subscription-manager repos --enable=rhel-7-server-extras-rpms
```

- Install the Azure Linux Agent by running the following command:

```
sudo yum install WALinuxAgent
sudo systemctl enable waagent.service
```

12. Don't create swap space on the operating system disk.

The Azure Linux Agent can automatically configure swap space by using the local resource disk that's attached to the VM after the VM is provisioned on Azure. The local resource disk is a temporary disk, and it might be emptied when the VM is deprovisioned. After you install the Azure Linux Agent in the previous

step, modify the following parameters in `/etc/waagent.conf` appropriately:

```
ResourceDisk.Format=y
ResourceDisk.Filesystem=ext4
ResourceDisk.MountPoint=/mnt/resource
ResourceDisk.EnableSwap=y
ResourceDisk.SwapSizeMB=2048 #NOTE: set this to whatever you need it to be.
```

13. If you want to unregister the subscription, run the following command:

```
sudo subscription-manager unregister
```

14. If you're using a system that was deployed using an Enterprise Certificate Authority, the RHEL VM won't trust the Azure Stack Hub root certificate. You need to place that into the trusted root store. For more information, see [Adding trusted root certificates to the server](#).
15. Run the following commands to deprovision the VM and prepare it for provisioning on Azure:

```
sudo waagent -force -deprovision
export HISTSIZE=0
logout
```

16. Select **Action** > **Shut Down** in Hyper-V Manager.
17. Convert the VHD to a fixed size VHD using either the Hyper-V Manager "Edit disk" feature, or the Convert-VHD PowerShell command. Your Linux VHD is now ready to be uploaded to Azure.

## Prepare a Red Hat-based virtual machine from KVM

1. Download the KVM image of RHEL 7 from the Red Hat website. This procedure uses RHEL 7 as the example.
2. Set a root password.

Generate an encrypted password, and copy the output of the command:

```
openssl passwd -1 changeme
```

Set a root password with guestfish:

```
guestfish --rw -a <image-name>
> <fs> run
> <fs> list-filesystems
> <fs> mount /dev/sda1 /
> <fs> vi /etc/shadow
> <fs> exit
```

Change the second field of root user from "!!" to the encrypted password.

3. Create a VM in KVM from the qcow2 image. Set the disk type to **qcow2**, and set the virtual network interface device model to **virtio**. Then, start the VM, and sign in as root.
4. Create or edit the `/etc/sysconfig/network` file, and add the following text:

```
NETWORKING=yes
HOSTNAME=localhost.localdomain
```

5. Create or edit the `/etc/sysconfig/network-scripts/ifcfg-eth0` file, and add the following text:

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
TYPE=Ethernet
USERCTL=no
PEERDNS=yes
IPV6INIT=no
NM_CONTROLLED=no
```

6. Ensure that the network service starts at boot time by running the following command:

```
sudo systemctl enable network
```

7. Register your Red Hat subscription to enable installation of packages from the RHEL repository by running the following command:

```
subscription-manager register --auto-attach --username=XXX --password=XXX
```

8. Modify the kernel boot line in your grub configuration to include additional kernel parameters for Azure. To do this configuration, open `/etc/default/grub` in a text editor, and modify the `GRUB_CMDLINE_LINUX` parameter. For example:

```
GRUB_CMDLINE_LINUX="rootdelay=300 console=ttyS0 earlyprintk=ttyS0 net.ifnames=0"
```

This command also ensures that all console messages are sent to the first serial port, which can assist Azure support with debugging issues. The command also turns off the new RHEL 7 naming conventions for NICs.

Graphical and quiet boot aren't useful in a cloud environment where all the logs are sent to the serial port. You can leave the `crashkernel` option configured if desired. This parameter reduces the amount of available memory in the VM by 128 MB or more, which might be problematic on smaller VM sizes. We recommend you remove the following parameters:

```
rhgb quiet crashkernel=auto
```

9. After you're done editing `/etc/default/grub`, run the following command to rebuild the grub configuration:

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

10. Add Hyper-V modules into initramfs.

Edit `/etc/dracut.conf` and add content:

```
add_drivers+="hv_vmbus hv_netvsc hv_storvsc"
```

Rebuild initramfs:

```
dracut -f -v
```

11. [Optional after 1910 release] Stop and Uninstall cloud-init:

```
systemctl stop cloud-init
yum remove cloud-init
```

12. Ensure that the SSH server is installed and configured to start at boot time:

```
systemctl enable sshd
```

Modify /etc/ssh/sshd\_config to include the following lines:

```
PasswordAuthentication yes
ClientAliveInterval 180
```

13. When creating a custom vhd for Azure Stack Hub, keep in mind that WALinuxAgent version between 2.2.20 and 2.2.35 (both exclusive) don't work on Azure Stack Hub environments before the 1910 release. You can use versions 2.2.20/2.2.35 versions to prepare your image. To use versions above 2.2.35 to prepare your custom image, update your Azure Stack Hub to 1903 release and above or apply the 1901/1902 hotfix.

[Before 1910 release] Follow these instructions to download a compatible WALinuxAgent:

- Download setuptools.

```
wget https://pypi.python.org/packages/source/s/setuptools/setuptools-7.0.tar.gz --no-check-certificate
tar xzf setuptools-7.0.tar.gz
cd setuptools-7.0
```

- Download and unzip the 2.2.20 version of the agent from our GitHub.

```
wget https://github.com/Azure/WALinuxAgent/archive/v2.2.20.zip
unzip v2.2.20.zip
cd WALinuxAgent-2.2.20
```

- Install setup.py.

```
sudo python setup.py install
```

- Restart waagent.

```
sudo systemctl restart waagent
```

- Test if the agent version matches the one you downloaded. For this example, it should be 2.2.20.

```
waagent -version
```

[After 1910 release] Follow these instructions to download a compatible WALinuxAgent:

- The WALinuxAgent package, `WALinuxAgent-<version>`, has been pushed to the Red Hat extras

repository. Enable the extras repository by running the following command:

```
subscription-manager repos --enable=rhel-7-server-extras-rpms
```

1. Install the Azure Linux Agent by running the following command:

```
```bash
sudo yum install WALinuxAgent
sudo systemctl enable waagent.service
````
```

14. Don't create swap space on the operating system disk.

The Azure Linux Agent can automatically configure swap space by using the local resource disk that's attached to the VM after the VM is provisioned on Azure. The local resource disk is a temporary disk, and it might be emptied when the VM is deprovisioned. After you install the Azure Linux Agent in the previous step, modify the following parameters in `/etc/waagent.conf` appropriately:

```
ResourceDisk.Format=y
ResourceDisk.Filesystem=ext4
ResourceDisk.MountPoint=/mnt/resource
ResourceDisk.EnableSwap=y
ResourceDisk.SwapSizeMB=2048 #NOTE: set this to whatever you need it to be.
```

15. Unregister the subscription (if necessary) by running the following command:

```
subscription-manager unregister
```

16. If you're using a system that was deployed using an Enterprise Certificate Authority, the RHEL VM won't trust the Azure Stack Hub root certificate. You need to place that into the trusted root store. For more information, see [Adding trusted root certificates to the server](#).

17. Run the following commands to deprovision the VM and prepare it for provisioning on Azure:

```
sudo waagent -force -deprovision
export HISTSIZE=0
logout
```

18. Shut down the VM in KVM.

19. Convert the qcow2 image to the VHD format.

#### NOTE

There's a known bug in qemu-img versions  $\geq 2.2.1$  that results in an improperly formatted VHD. The issue has been fixed in QEMU 2.6. It's recommended to use either qemu-img 2.2.0 or lower, or update to 2.6 or higher. Reference: <https://bugs.launchpad.net/qemu/+bug/1490611>.

First convert the image to raw format:

```
qemu-img convert -f qcow2 -O raw rhel-7.4.qcow2 rhel-7.4.raw
```

Make sure that the size of the raw image is aligned with 1 MB. Otherwise, round up the size to align with 1

MB:

```
MB=$((1024*1024))
size=$(qemu-img info -f raw --output json "rhel-7.4.raw" | \
gawk 'match($0, /"virtual-size": ([0-9]+),/, val) {print val[1]}')
rounded_size=$((($size/$MB + 1)*$MB))
qemu-img resize rhel-7.4.raw $rounded_size
```

Convert the raw disk to a fixed-sized VHD:

```
qemu-img convert -f raw -o subformat=fixed -O vpc rhel-7.4.raw rhel-7.4.vhd
```

Or, with qemu version **2.6+**, include the `force_size` option:

```
qemu-img convert -f raw -o subformat=fixed,force_size -O vpc rhel-7.4.raw rhel-7.4.vhd
```

## Prepare a Red Hat-based VM from VMware

This section assumes that you've already installed an RHEL VM in VMware. For details about how to install an operating system in VMware, see [VMware Guest Operating System Installation Guide](#).

- When you install the Linux operating system, we recommend that you use standard partitions rather than LVM, which is often the default for many installations. This method avoids LVM name conflicts with cloned VMs, particularly if an operating system disk ever needs to be attached to another VM for troubleshooting. LVM or RAID can be used on data disks if preferred.
- Don't configure a swap partition on the operating system disk. You can configure the Linux agent to create a swap file on the temporary resource disk. You can find more information about this configuration in the steps that follow.
- When you create the virtual hard disk, select **Store virtual disk as a single file**.

### Prepare an RHEL 7 VM from VMware

- Create or edit the `/etc/sysconfig/network` file, and add the following text:

```
NETWORKING=yes
HOSTNAME=localhost.localdomain
```

- Create or edit the `/etc/sysconfig/network-scripts/ifcfg-eth0` file, and add the following text:

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
TYPE=Ethernet
USERCTL=no
PEERDNS=yes
IPV6INIT=no
NM_CONTROLLED=no
```

- Ensure that the network service will start at boot time by running the following command:

```
sudo chkconfig network on
```

- Register your Red Hat subscription to enable the installation of packages from the RHEL repository by

running the following command:

```
sudo subscription-manager register --auto-attach --username=XXX --password=XXX
```

5. Modify the kernel boot line in your grub configuration to include additional kernel parameters for Azure. To make this modification, open `/etc/default/grub` in a text editor, and modify the `GRUB_CMDLINE_LINUX` parameter. For example:

```
GRUB_CMDLINE_LINUX="rootdelay=300 console=ttyS0 earlyprintk=ttyS0 net.ifnames=0"
```

This configuration also ensures that all console messages are sent to the first serial port, which can assist Azure support with debugging issues. It also turns off the new RHEL 7 naming conventions for NICs. We recommend that you remove the following parameters:

```
rhgb quiet crashkernel=auto
```

Graphical and quiet boot aren't useful in a cloud environment where we want all the logs to be sent to the serial port. You can leave the `crashkernel` option configured if desired. This parameter reduces the amount of available memory in the VM by 128 MB or more, which might be problematic on smaller VM sizes.

6. After you're done editing `/etc/default/grub`, run the following command to rebuild the grub configuration:

```
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

7. Add Hyper-V modules to initramfs.

Edit `/etc/dracut.conf`, add content:

```
add_drivers+="hv_vmbus hv_netvsc hv_storvsc"
```

Rebuild initramfs:

```
dracut -f -v
```

8. [Optional after 1910 release] Stop and uninstall cloud-init:

```
systemctl stop cloud-init
yum remove cloud-init
```

9. Ensure that the SSH server is installed and configured to start at boot time. This setting is usually the default. Modify `/etc/ssh/sshd_config` to include the following line:

```
ClientAliveInterval 180
```

10. When creating a custom vhd for Azure Stack Hub, keep in mind that WALinuxAgent version between 2.2.20 and 2.2.35 (both exclusive) don't work on Azure Stack Hub environments before the 1910 release. You can use versions 2.2.20/2.2.35 versions to prepare your image. To use versions above 2.2.35 to prepare your custom image, update your Azure Stack Hub to 1903 release and above or apply the 1901/1902 hotfix.

[Before 1910 release] Follow these instructions to download a compatible WALinuxAgent:

a. Download setuptools.

```
wget https://pypi.python.org/packages/source/s/setuptools/setuptools-7.0.tar.gz --no-check-certificate
tar xzf setuptools-7.0.tar.gz
cd setuptools-7.0
```

b. Download and unzip the 2.2.20 version of the agent from our GitHub.

```
wget https://github.com/Azure/WALinuxAgent/archive/v2.2.20.zip
unzip v2.2.20.zip
cd WALinuxAgent-2.2.20
```

c. Install setup.py.

```
sudo python setup.py install
```

d. Restart waagent.

```
sudo systemctl restart waagent
```

e. Test if the agent version matches the one you downloaded. For this example, it should be 2.2.20.

```
waagent -version
```

[After 1910 release] Follow these instructions to download a compatible WALinuxAgent:

a. The WALinuxAgent package, `WALinuxAgent-<version>`, has been pushed to the Red Hat extras repository. Enable the extras repository by running the following command:

```
subscription-manager repos --enable=rhel-7-server-extras-rpms
```

1. Install the Azure Linux Agent by running the following command:

```
```bash  
sudo yum install WALinuxAgent  
sudo systemctl enable waagent.service  
```
```

11. Don't create swap space on the operating system disk.

The Azure Linux Agent can automatically configure swap space by using the local resource disk that's attached to the VM after the VM is provisioned on Azure. Note that the local resource disk is a temporary disk, and it might be emptied when the VM is deprovisioned. After you install the Azure Linux Agent in the previous step, modify the following parameters in `/etc/waagent.conf` appropriately:

```
ResourceDisk.Format=
ResourceDisk.Filesystem=ext4
ResourceDisk.MountPoint=/mnt/resource
ResourceDisk.EnableSwap=
ResourceDisk.SwapSizeMB=2048 NOTE: set this to whatever you need it to be.
```

12. If you want to unregister the subscription, run the following command:

```
sudo subscription-manager unregister
```

13. If you're using a system that was deployed using an Enterprise Certificate Authority, the RHEL VM won't trust the Azure Stack Hub root certificate. You need to place that into the trusted root store. For more information, see [Adding trusted root certificates to the server](#).

14. Run the following commands to deprovision the VM and prepare it for provisioning on Azure:

```
sudo waagent -force -deprovision
export HISTSIZE=0
logout
```

15. Shut down the VM, and convert the VMDK file to the VHD format.

**NOTE**

There's a known bug in qemu-img versions >=2.2.1 that results in an improperly formatted VHD. The issue has been fixed in QEMU 2.6. It's recommended to use either qemu-img 2.2.0 or lower, or update to 2.6 or higher. Reference: <https://bugs.launchpad.net/qemu/+bug/1490611>.

First convert the image to raw format:

```
qemu-img convert -f qcow2 -O raw rhel-7.4.qcow2 rhel-7.4.raw
```

Make sure that the size of the raw image is aligned with 1 MB. Otherwise, round up the size to align with 1 MB:

```
MB=$((1024*1024))
size=$(qemu-img info -f raw --output json "rhel-7.4.raw" | \
gawk 'match($0, /"virtual-size": ([0-9]+),/, val) {print val[1]}')
rounded_size=$((($size/$MB + 1)*$MB))
qemu-img resize rhel-7.4.raw $rounded_size
```

Convert the raw disk to a fixed-sized VHD:

```
qemu-img convert -f raw -o subformat=fixed -O vpc rhel-7.4.raw rhel-7.4.vhd
```

Or, with qemu version **2.6+**, include the `force_size` option:

```
qemu-img convert -f raw -o subformat=fixed,force_size -O vpc rhel-7.4.raw rhel-7.4.vhd
```

## Prepare a Red Hat-based VM from an ISO by using a kickstart file automatically

1. Create a kickstart file that includes the following content, and save the file. Stopping and uninstalling cloud-init is optional (cloud-init is supported on Azure Stack Hub post 1910 release). Install the agent from the redhat repo only after the 1910 release. Prior to 1910, use the Azure repo as done in the previous section. For details about kickstart installation, see the [Kickstart Installation Guide](#).

```
Kickstart for provisioning a RHEL 7 Azure VM
```

```
System authorization information
auth --enablesshadow --passalgo=sha512
```

```
Use graphical install
text
```

```
Do not run the Setup Agent on first boot
firstboot --disable
```

```
Keyboard layouts
keyboard --vckeymap=us --xlayouts='us'
```

```
System language
lang en_US.UTF-8
```

```
Network information
network --bootproto=dhcp
```

```
Root password
rootpw --plaintext "to_be_disabled"
```

```
System services
services --enabled="sshd,waagent,NetworkManager"
```

```
System timezone
timezone Etc/UTC --isUtc --ntpservers
0.rhel.pool.ntp.org,1.rhel.pool.ntp.org,2.rhel.pool.ntp.org,3.rhel.pool.ntp.org
```

```
Partition clearing information
clearpart --all --initlabel
```

```
Clear the MBR
zerombr
```

```
Disk partitioning information
part /boot --fstype="xfs" --size=500
part / --fstype="xfs" --size=1 --grow --asprimary
```

```
System bootloader configuration
bootloader --location=mbr
```

```
Firewall configuration
firewall --disabled
```

```
Enable SELinux
selinux --enforcing
```

```
Don't configure X
skipx
```

```
Power down the machine after install
poweroff
```

```
%packages
@base
@console-internet
chrony
sudo
parted
-dracut-config-rescue
```

```
%end
```

```
%post --log=/var/log/anaconda/post-install.log
```

```
#!/bin/bash
```

```

Register Red Hat Subscription
subscription-manager register --username=XXX --password=XXX --auto-attach --force

Install latest repo update
yum update -y

Stop and Uninstall cloud-init
systemctl stop cloud-init
yum remove cloud-init

Enable extras repo
subscription-manager repos --enable=rhel-7-server-extras-rpms

Install WALinuxAgent
yum install -y WALinuxAgent

Unregister Red Hat subscription
subscription-manager unregister

Enable waagent at boot-up
systemctl enable waagent

Disable the root account
usermod root -p '!!!'

Configure swap in WALinuxAgent
sed -i 's/^(\ResourceDisk\EnableSwap\)=\([Nn]\$/\1=y/g' /etc/waagent.conf
sed -i 's/^(\ResourceDisk\SwapSizeMB\)=\([0-9]*\$/\1=2048/g' /etc/waagent.conf

Set the cmdline
sed -i 's/^(\GRUB_CMDLINE_LINUX\)=.*$/\1="console=tty1 console=ttyS0 earlyprintk=ttyS0
rootdelay=300"/g' /etc/default/grub

Enable SSH keepalive
sed -i 's/^#\!(ClientAliveInterval\).*/\1 180/g' /etc/ssh/sshd_config

Build the grub cfg
grub2-mkconfig -o /boot/grub2/grub.cfg

Configure network
cat << EOF > /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
TYPE=Ethernet
USERCTL=no
PEERDNS=yes
IPV6INIT=no
NM_CONTROLLED=no
EOF

Deprovision and prepare for Azure
waagent -force -deprovision

%end

```

2. Place the kickstart file where the installation system can access it.
3. In Hyper-V Manager, create a new VM. On the **Connect Virtual Hard Disk** page, select **Attach a virtual hard disk later**, and complete the New Virtual Machine Wizard.
4. Open the VM settings:
  - a. Attach a new virtual hard disk to the VM. Make sure to select **VHD Format** and **Fixed Size**.
  - b. Attach the installation ISO to the DVD drive.

- c. Set the BIOS to boot from CD.
5. Start the VM. When the installation guide appears, press **Tab** to configure the boot options.
6. Enter `inst.ks=<the location of the kickstart file>` at the end of the boot options, and press **Enter**.
7. Wait for the installation to finish. When it's finished, the VM is shut down automatically. Your Linux VHD is now ready to be uploaded to Azure.

## Known issues

### The Hyper-V driver couldn't be included in the initial RAM disk when using a non-Hyper-V hypervisor

In some cases, Linux installers might not include the drivers for Hyper-V in the initial RAM disk (initrd or initramfs) unless Linux detects that it's running in a Hyper-V environment.

When you're using a different virtualization system (like Oracle VM VirtualBox, Xen Project, and so on) to prepare your Linux image, you might need to rebuild initrd to ensure that at least the hv\_vmbus and hv\_storvsc kernel modules are available on the initial RAM disk. This is a known issue at least on systems that are based on the upstream Red Hat distribution.

To resolve this issue, add Hyper-V modules to initramfs and rebuild it:

Edit `/etc/dracut.conf`, and add the following content:

```
add_drivers+="hv_vmbus hv_netvsc hv_storvsc"
```

Rebuild initramfs:

```
dracut -f -v
```

For more information, see [rebuilding initramfs](#).

## Next steps

You're now ready to use your Red Hat Enterprise Linux virtual hard disk to create new VMs in Azure Stack Hub. If this is the first time that you're uploading the VHD file to Azure Stack Hub, see [Create and publish a Marketplace item](#).

For more information about the hypervisors that are certified to run Red Hat Enterprise Linux, see [the Red Hat website](#).

# Add the Azure Kubernetes Services (AKS) engine prerequisites to the Azure Stack Hub Marketplace

2 minutes to read • [Edit Online](#)

You can enable your users to set up the Azure Kubernetes Services (AKS) Engine by adding the items described in this article to your Azure Stack Hub. Your users can then deploy a Kubernetes cluster in a single, coordinated operation. This article walks you through the steps you need to make the AKS engine available to your users in both connected and disconnected environments. The AKS engine depends on a service principle identity, and in the marketplace, a Custom Script extension and the AKS Base Image. The AKS engine requires that you are running [Azure Stack Hub 1910](#) or greater.

## Check your user's service offering

Your users will need a plan, offer, and subscription to Azure Stack Hub with enough space. Users will often want to deploy clusters of up to six virtual machines, made of three masters and three worker nodes. You will want to make sure they have a large enough quota.

If you need more information about planning and setting up a service offering, see [Overview of offering services in Azure Stack Hub](#)

## Create a service principal and credentials

The Kubernetes cluster will need service principal (SPN) and role-based permissions in Azure Stack Hub.

### Create an SPN in Azure AD

If you use Azure Active Directory (Azure AD) for your identity management service, you will need to create a service principal for users deploying a Kubernetes cluster. Create a service principal using a client secret. For instructions, see [Create a service principal that uses a client secret credential](#).

### Create an SPN in AD FS

If you use Active Directory Federated Services (AD FS) for your identity management service, you will need to create a service principal for users deploying a Kubernetes cluster. Create a service principal using a client secret. For instructions, see [Create a service principal using a client secret](#).

## Add the AKS Base Image

You can add the AKS Base Image to the marketplace by getting the item from Azure. However, if your Azure Stack Hub is disconnected, use these instructions [Download marketplace items from Azure](#) to add the item. Add the item specified in step 5.

Add the following item to the marketplace:

1. Sign in to the [Administration portal](#).
2. Select **All services**, and then under the **ADMINISTRATION** category, select **Marketplace management**.
3. Select **+ Add from Azure**.
4. Enter **AKS Base**.
5. Select the image version that matches the version of the AKS engine. You can find listing of AKS Base Image to AKS engine version at [Supported Kubernetes Versions](#).

In the list, select:

- **Publisher:** Azure Kubernetes Service
- **Offer:** aks
- **Version:** AKS Base Image 16.04-LTS Image Distro, October 2019 (2019.10.24 or version that maps to AKS engine)

6. Select **Download**.

## Add a Custom Script extension

You can add the custom script to the marketplace by getting the item from Azure. However, if your Azure Stack Hub is disconnected, use the instructions [Download marketplace items from Azure](#) to add the item. Add the item specified in step 5.

1. Open the [Administration portal](#).
2. Select **ALL services** and then under the **ADMINISTRATION** category, select **Marketplace Management**.

3. Select **+ Add from Azure**.

4. Enter `Custom Script for Linux`.

5. Select the script with the following profile:

- **Offer:** Custom Script for Linux 2.0
- **Version:** 2.0.6 (or latest version)
- **Publisher:** Microsoft Corp

**NOTE**

More than one version of the Custom Script for Linux may be listed. You will need to add the last version of the item.

6. Select **Download**.

## Next steps

[What is the AKS engine on Azure Stack Hub?](#)

[Overview of offering services in Azure Stack Hub](#)

# Add Kubernetes to Azure Stack Hub Marketplace

3 minutes to read • [Edit Online](#)

## NOTE

Only use the Kubernetes Azure Stack Hub Marketplace item to deploy clusters as a proof-of-concept. For supported Kubernetes clusters on Azure Stack Hub, use [the AKS engine](#).

You can offer Kubernetes as a marketplace item to your users. Your users can then deploy Kubernetes in a single, coordinated operation.

This article looks at using an Azure Resource Manager template to deploy and provision the resources for a standalone Kubernetes cluster. Before you start, check your Azure Stack Hub and global Azure tenant settings. Collect the required information about your Azure Stack Hub. Add necessary resources to your tenant and to Azure Stack Hub Marketplace. The cluster depends on an Ubuntu server, custom script, and the Kubernetes Cluster marketplace item to be in Azure Stack Hub Marketplace.

## Create a plan, an offer, and a subscription

Create a plan, an offer, and a subscription for the Kubernetes marketplace item. You can also use an existing plan and offer.

1. Sign in to the [administrator portal](#).
2. Create a plan as the base plan. For instructions, see [Create a plan in Azure Stack Hub](#).
3. Create an offer. For instructions, see [Create an offer in Azure Stack Hub](#).
4. Select **Offers**, and find the offer you created.
5. Select **Overview** in the Offer blade.
6. Select **Change state**. Select **Public**.
7. Select **+ Create a resource > Offers and Plans > Subscription** to create a subscription.
  - a. Enter a **Display Name**.
  - b. Enter a **User**. Use the Azure AD account associated with your tenant.
  - c. **Provider Description**
  - d. Set the **Directory tenant** to the Azure AD tenant for your Azure Stack Hub.
  - e. Select **Offer**. Select the name of the offer that you created. Make note of the Subscription ID.

## Create a service principal and credentials in AD FS

If you use Active Directory Federated Services (AD FS) for your identity management service, you need to create a service principal for users deploying a Kubernetes cluster. Create service principal using a client secret. For instructions, see [Create a service principal using a client secret](#).

## Add an Ubuntu server image

Add the following Ubuntu Server image to Azure Stack Hub Marketplace:

1. Sign in to the [administrator portal](#).
2. Select **All services**, and then under the **ADMINISTRATION** category, select **Marketplace management**.
3. Select **+ Add from Azure**.
4. Enter `Ubuntu Server`.
5. Select the newest version of the server. Check the full version and ensure that you have the newest version:
  - **Publisher:** Canonical
  - **Offer:** UbuntuServer
  - **Version:** 16.04.201806120 (or latest version)
  - **SKU:** 16.04-LTS
6. Select **Download**.

## Add a custom script for Linux

Add the Kubernetes from Azure Stack Hub Marketplace:

1. Open the [administrator portal](#).
2. Select **ALL services** and then under the **ADMINISTRATION** category, select **Marketplace Management**.
3. Select **+ Add from Azure**.
4. Enter `Custom Script for Linux`.
5. Select the script with the following profile:
  - **Offer:** Custom Script for Linux 2.0
  - **Version:** 2.0.6 (or latest version)
  - **Publisher:** Microsoft Corp

### NOTE

More than one version of Custom Script for Linux may be listed. You need to add the last version of the item.

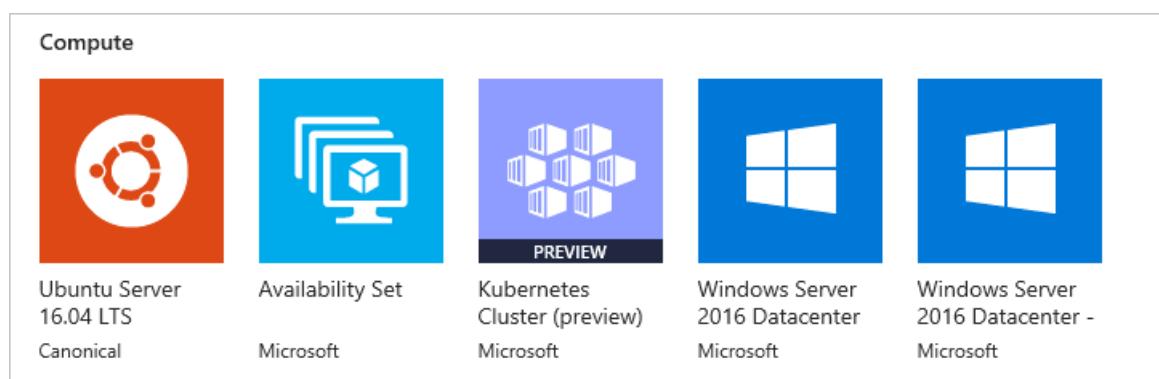
6. Select **Download**.

## Add Kubernetes to the marketplace

1. Open the [administrator portal](#).
2. Select **All services** and then under the **ADMINISTRATION** category, select **Marketplace Management**.
3. Select **+ Add from Azure**.
4. Enter `Kubernetes`.
5. Select `Kubernetes Cluster`.
6. Select **Download**.

#### NOTE

It may take five minutes for the marketplace item to appear in Azure Stack Hub Marketplace.



## Update or remove the Kubernetes

When updating the Kubernetes item, you remove the previous item in Azure Stack Hub Marketplace. Follow the instruction below to add the Kubernetes update to Azure Stack Hub Marketplace.

To remove the Kubernetes item:

1. Connect to Azure Stack Hub with PowerShell as an operator. For instruction, see [Connect to Azure Stack Hub with PowerShell as an operator](#).
2. Find the current Kubernetes Cluster item in the gallery.

```
Get-AzsGalleryItem | Select Name
```

3. Note name of the current item, such as `Microsoft.AzureStackKubernetesCluster.0.3.0`.
4. Use the following PowerShell cmdlet to remove the item:

```
$Itemname="Microsoft.AzureStackKubernetesCluster.0.3.0"
Remove-AzsGalleryItem -Name $Itemname
```

## Next steps

[Deploy a Kubernetes to Azure Stack Hub](#)

[Overview of offering services in Azure Stack Hub](#)

# Create a site-to-site VPN connection between two virtual networks in different ASDK environments

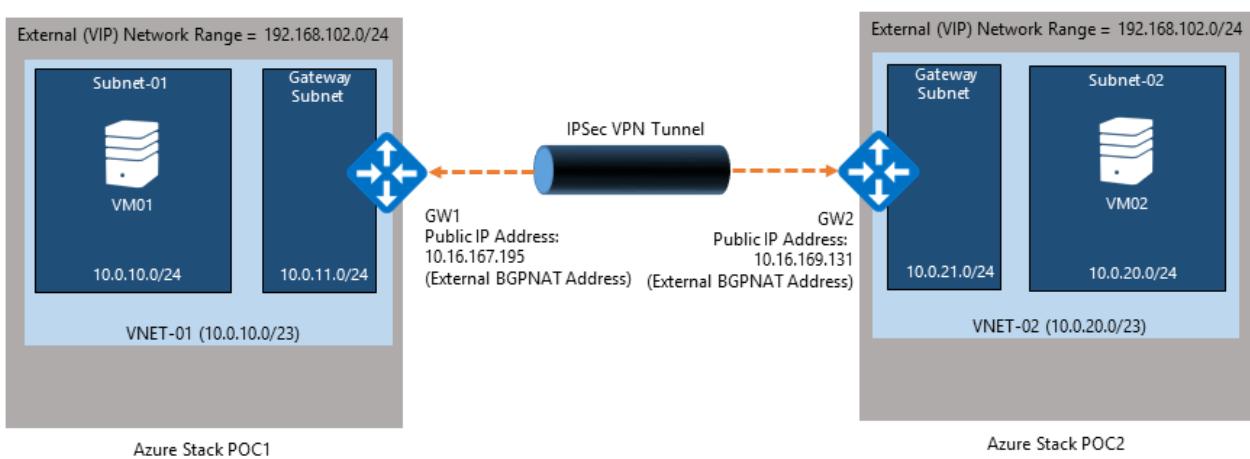
14 minutes to read • [Edit Online](#)

## Overview

This article describes how to create a site-to-site VPN connection between two virtual networks in two separate Azure Stack Development Kit (ASDK) environments. When you configure the connections, you learn how VPN gateways in Azure Stack Hub work.

### Connection

The following figure shows what the connection configuration should look like when you're done.



### Before you begin

To complete the connection configuration, ensure that you have the following items before you begin:

- Two servers and other prerequisites that meet the ASDK hardware requirements, as described in [Quickstart: Evaluate the Azure Stack Development Kit](#).
- The [ASDK](#) deployment package.

## Deploy the Azure Stack Development Kit environments

To complete the connection configuration, you must deploy two ASDK environments.

#### NOTE

For each ASDK that you deploy, follow the [deployment instructions](#). In this article, the ASDK environments are called **POC1** and **POC2**.

## Prepare an offer on POC1 and POC2

On both POC1 and POC2, prepare an offer so that a user can subscribe to the offer and deploy the virtual machines (VMs). For information on how to create an offer, see [Make VMs available to your Azure Stack Hub users](#).

## Review and complete the network configuration table

The following table summarizes the network configuration for both ASDK environments. Use the procedure that appears after the table to add the External BGPNAT address that's specific for your network.

### Network configuration table

|                               | POC1         | POC2         |
|-------------------------------|--------------|--------------|
| Virtual network name          | VNET-01      | VNET-02      |
| Virtual network address space | 10.0.10.0/23 | 10.0.20.0/23 |
| Subnet name                   | Subnet-01    | Subnet-02    |
| Subnet address range          | 10.0.10.0/24 | 10.0.20.0/24 |
| Gateway subnet                | 10.0.11.0/24 | 10.0.21.0/24 |
| External BGPNAT address       |              |              |

#### NOTE

The external BGPNAT IP addresses in the example environment are 10.16.167.195 for POC1, and 10.16.169.131 for POC2. Use the following procedure to determine the external BGPNAT IP addresses for your ASDK hosts, and then add them to the previous network configuration table.

### Get the IP address of the external adapter of the NAT VM

1. Sign in to the Azure Stack Hub physical machine for POC1.
2. Open PowerShell as an Administrator and run the following cmdlet:

```
Get-NetNatExternalAddress
```

3. Add the IP address to the network configuration table that appears in the previous section.
4. Repeat this procedure on POC2.

## Create the network resources in POC1

Now you can create the POC1 network resources that you need to set up your gateways. The following instructions describe how to create the resources by using the Azure Stack Hub user portal. You can also use PowerShell code to create the resources.



### Sign in as a tenant

A service administrator can sign in as a tenant to test the plans, offers, and subscriptions that their tenants might use. If you don't already have one, [create a tenant account](#) before you sign in.

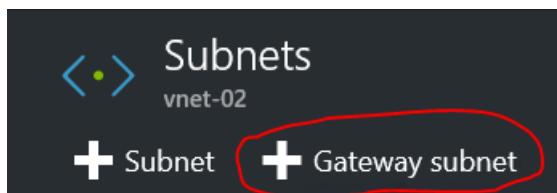
### Create the virtual network and VM subnet

1. Use a tenant account to sign in to the user portal.
2. In the user portal, select **+** **Create a resource**.
3. Go to **Marketplace**, and then select **Networking**.

4. Select **Virtual network**.
5. For **Name**, **Address space**, **Subnet name**, and **Subnet address range**, use the values that appear earlier in the network configuration table.
6. In **Subscription**, the subscription that you created earlier appears.
7. For **Resource Group**, you can either create a resource group or if you already have one, select **Use existing**.
8. Verify the default location.
9. Select **Pin to dashboard**.
10. Select **Create**.

#### Create the gateway subnet

1. On the dashboard, open the VNET-01 virtual network resource that you created previously.
2. On the **Settings** blade, select **Subnets**.
3. To add a gateway subnet to the virtual network, select **Gateway Subnet**.



4. By default, the subnet name is set to **GatewaySubnet**. Gateway subnets are special. To function properly, they must use the **GatewaySubnet** name.
  5. In **Address range**, verify that the address is **10.0.11.0/24**.
  6. Select **OK** to create the gateway subnet.
- Create the virtual network gateway**
1. In the Azure portal, select **+ Create a resource**.
  2. Go to **Marketplace**, and then select **Networking**.
  3. From the list of network resources, select **Virtual network gateway**.
  4. In **Name**, enter **GW1**.
  5. Select the **Virtual network** item to choose a virtual network. Select **VNET-01** from the list.
  6. Select the **Public IP address** menu item. When the **Choose public IP address** window opens, select **Create new**.
  7. In **Name**, enter **GW1-PiP**, and then select **OK**.
  8. By default, for **VPN type**, **Route-based** is selected. Keep the **Route-based** VPN type.
  9. Verify that **Subscription** and **Location** are correct. You can pin the resource to the dashboard. Select **Create**.

#### Create the local network gateway

The implementation of a *local network gateway* in this Azure Stack Hub evaluation deployment is a bit different than in an actual Azure deployment.

In an Azure deployment, a local network gateway represents an on-premises (at the tenant) physical device that you use to connect to a virtual network gateway in Azure. In this Azure Stack Hub evaluation deployment, both ends of the connection are virtual network gateways.

A way to think about this more generically is that the local network gateway resource always indicates the remote gateway at the other end of the connection. Because of the way the ASDK is designed, you must provide the IP address of the external network adapter on the network address translation (NAT) VM of the other ASDK as the public IP address of the local network gateway. You then create NAT mappings on the NAT VM to make sure that both ends are connected properly.

## Create the local network gateway resource

1. Sign in to the Azure Stack Hub physical machine for POC1.
2. In the user portal, select + **Create a resource**.
3. Go to **Marketplace**, and then select **Networking**.
4. From the list of resources, select **local network gateway**.
5. In **Name**, enter **POC2-GW**.
6. In **IP address**, enter the External BGP NAT address for POC2. This address appears earlier in the network configuration table.
7. In **Address Space**, for the address space of the POC2 VNET that you create later, enter **10.0.20.0/23**.
8. Verify that your **Subscription**, **Resource Group**, and **location** values are correct, and then select **Create**.

## Create the connection

1. In the user portal, select + **Create a resource**.
2. Go to **Marketplace**, and then select **Networking**.
3. From the list of resources, select **Connection**.
4. On the **Basics** settings blade, for the **Connection type**, select **Site-to-site (IPSec)**.
5. Select the **Subscription**, **Resource Group**, and **Location**, and then select **OK**.
6. On the **Settings** blade, select **Virtual network gateway**, and then select **GW1**.
7. Select **Local network gateway**, and then select **POC2-GW**.
8. In **Connection Name**, enter **POC1-POC2**.
9. In **Shared key (PSK)**, enter **12345**, and then select **OK**.
10. On the **Summary** blade, select **OK**.

## Create a virtual machine

To validate the data that travels through the VPN connection, you need the VMs to send and receive data in each ASDK. Create a VM in POC1 now, and then in your virtual network, put it on your VM subnet:

1. In the Azure portal, select + **Create a resource**.
2. Go to **Marketplace**, and then select **Compute**.
3. In the list of VM images, select the **Windows Server 2016 Datacenter Eval** image.
4. On the **Basics** blade, in **Name**, enter **VM01**.
5. Enter a valid username and password. You use this account to sign in to the VM after it's created.
6. Provide a **Subscription**, **Resource Group**, and **Location**, and then select **OK**.
7. On the **Size** blade, for this instance, select a VM size, and then select **Select**.
8. On the **Settings** blade, accept the defaults. Ensure that the **VNET-01** virtual network is selected. Verify that the subnet is set to **10.0.10.0/24**. Then select **OK**.
9. On the **Summary** blade, review the settings, and then select **OK**.

## Create the network resources in POC2

The next step is to create the network resources for POC2. The following instructions show how to create the resources by using the user portal.

### Sign in as a tenant again

A service administrator can sign in as a tenant to test the plans, offers, and subscriptions that their tenants might use. If you don't already have one, [create a tenant account](#) before you sign in.

### Create virtual network and VM subnet

1. Sign in by using a tenant account.
2. In the user portal, select + **Create a resource**.

3. Go to **Marketplace**, and then select **Networking**.
4. Select **Virtual network**.
5. Use the information appearing earlier in the network configuration table to identify the values for the POC2 **Name**, **Address space**, **Subnet name**, and **Subnet address range**.
6. In **Subscription**, the subscription that you created earlier appears.
7. For **Resource Group**, create a new resource group or, if you already have one, select **Use existing**.
8. Verify the default **Location**.
9. Select **Pin to dashboard**.
10. Select **Create**.

#### **Create gateway subnet**

1. Open the Virtual network resource you created (**VNET-02**) from the dashboard.
2. On the **Settings** blade, select **Subnets**.
3. Select **Gateway subnet** to add a gateway subnet to the virtual network.
4. The name of the subnet is set to **GatewaySubnet** by default. Gateway subnets are special and must have this specific name to function properly.
5. In the **Address range** field, verify the address is **10.0.21.0/24**.
6. Select **OK** to create the gateway subnet.

#### **Create virtual network gateway**

1. In the Azure portal, select **+ Create a resource**.
2. Go to **Marketplace**, and then select **Networking**.
3. From the list of network resources, select **Virtual network gateway**.
4. In **Name**, enter **GW2**.
5. To choose a virtual network, select **Virtual network**. Then select **VNET-02** from the list.
6. Select **Public IP address**. When the **Choose public IP address** blade opens, select **Create new**.
7. In **Name**, enter **GW2-PiP**, and then select **OK**.
8. By default, **Route-based** is selected for **VPN type**. Keep the **Route-based** VPN type.
9. Verify that **Subscription** and **Location** are correct. You can pin the resource to the dashboard. Select **Create**.

#### **Create local network gateway resource**

1. In the POC2 user portal, select **+ Create a resource**.
2. Go to **Marketplace**, and then select **Networking**.
3. From the list of resources, select **Local network gateway**.
4. In **Name**, enter **POC1-GW**.
5. In **IP address**, enter the External BGPNAT address for POC1 that's listed previously in the network configuration table.
6. In **Address Space**, from POC1, enter the **10.0.10.0/23** address space of **VNET-01**.
7. Verify that your **Subscription**, **Resource Group**, and **Location** are correct, and then select **Create**.

## **Create connection**

1. In the user portal, select **+ Create a resource**.
2. Go to **Marketplace**, and then select **Networking**.
3. From the list of resources, select **Connection**.
4. On the **Basic** settings blade, for the **Connection type**, choose **Site-to-site (IPSec)**.
5. Select the **Subscription**, **Resource Group**, and **Location**, and then select **OK**.
6. On the **Settings** blade, select **Virtual network gateway**, and then select **GW2**.
7. Select **Local network gateway**, and then select **POC1-GW**.

8. In **Connection name**, enter **POC2-POC1**.
9. In **Shared key (PSK)**, enter **12345**. If you choose a different value, remember that it must match the value for the shared key that you created on POC1. Select **OK**.
10. Review the **Summary** blade, and then select **OK**.

## Create a virtual machine

Now create a VM in POC2, and put it on your VM subnet in your virtual network:

1. In the Azure portal, select **+ Create a resource**.
2. Go to **Marketplace**, and then select **Compute**.
3. In the list of VM images, select the **Windows Server 2016 Datacenter Eval** image.
4. On the **Basics** blade, for **Name**, enter **VM02**.
5. Enter a valid username and password. You use this account to sign in to the VM after it's created.
6. Provide a **Subscription**, **Resource Group**, and **Location**, and then select **OK**.
7. On the **Size** blade, select a VM size for this instance, and then select **Select**.
8. On the **Settings** blade, you can accept the defaults. Ensure that the **VNET-02** virtual network is selected, and verify that the subnet is set to **10.0.20.0/24**. Select **OK**.
9. Review the settings on the **Summary** blade, and then select **OK**.

## Configure the NAT VM on each ASDK for gateway traversal

Because the ASDK is self-contained and isolated from the network on which the physical host is deployed, the *external* VIP network that the gateways are connected to isn't actually external. Instead, the VIP network is hidden behind a router that performs network address translation.

The router is a Windows Server VM, called **AzS-bgpnat01**, that runs the Routing and Remote Access Services (RRAS) role in the ASDK infrastructure. You must configure NAT on the AzS-bgpnat01 VM to allow the site-to-site VPN connection to connect on both ends.

To configure the VPN connection, you must create a static NAT map route that maps the external interface on the BGPNAT VM to the VIP of the edge gateway pool. A static NAT map route is required for each port in a VPN connection.

### NOTE

This configuration is required for ASDK environments only.

## Configure the NAT

### IMPORTANT

You must complete this procedure for both ASDK environments.

1. Determine the **Internal IP address** to use in the following PowerShell script. Open the virtual network gateway (GW1 and GW2). On the **Overview** blade, save the value for the **Public IP address** for later use.

The screenshot shows the Microsoft Azure Stack portal interface. The left sidebar has icons for Home, Compute, Storage, Network, and Monitoring. The main area is titled 'Microsoft Azure Stack' and shows a 'Virtual network gateway' named 'GW1'. The 'Overview' tab is selected. On the right, under the 'Essentials' section, there is a table with the following data:

|                                                                  |                                                              |
|------------------------------------------------------------------|--------------------------------------------------------------|
| Resource group (change)<br><a href="#">UserRG</a>                | Gateway type<br>VPN                                          |
| Location<br>local                                                | VPN type<br>Route-based                                      |
| Subscription name (change)<br><a href="#">TenantSubscription</a> | Virtual network<br><a href="#">VNET-01</a>                   |
| Subscription ID<br>bffe624c-2e78-44d0-9e4b-386623b59474          | Public IP address<br><a href="#">192.168.102.1 (GW1-Pip)</a> |

2. Sign in to the Azure Stack Hub physical machine for POC1.
3. Copy and edit the following PowerShell script. To configure the NAT on each ASDK, run the script in an elevated Windows PowerShell ISE. In the script, add values to the `<External BGPNAT address>` and `<Internal IP address>` placeholders:

```
Designate the external NAT address for the ports that use the IKE authentication.
Invoke-Command `

-ComputerName AzS-bgpnat01 `

{Add-NetNatExternalAddress `

-NatName BGPNAT `

-IPAddress <External BGPNAT address> `

-PortStart 499 `

-PortEnd 501}

Invoke-Command `

-ComputerName AzS-bgpnat01 `

{Add-NetNatExternalAddress `

-NatName BGPNAT `

-IPAddress <External BGPNAT address> `

-PortStart 4499 `

-PortEnd 4501}

create a static NAT mapping to map the external address to the Gateway
Public IP Address to map the ISAKMP port 500 for PHASE 1 of the IPSEC tunnel
Invoke-Command `

-ComputerName AzS-bgpnat01 `

{Add-NetNatStaticMapping `

-NatName BGPNAT `

-Protocol UDP `

-ExternalIPAddress <External BGPNAT address> `

-InternalIPAddress <Internal IP address> `

-ExternalPort 500 `

-InternalPort 500}

Finally, configure NAT traversal which uses port 4500 to
successfully establish the complete IPSEC tunnel over NAT devices
Invoke-Command `

-ComputerName AzS-bgpnat01 `

{Add-NetNatStaticMapping `

-NatName BGPNAT `

-Protocol UDP `

-ExternalIPAddress <External BGPNAT address> `

-InternalIPAddress <Internal IP address> `

-ExternalPort 4500 `

-InternalPort 4500}
```

4. Repeat this procedure on POC2.

## Test the connection

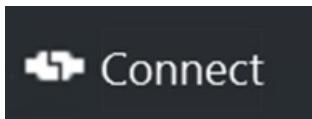
Now that the site-to-site connection is established, you should validate that you can get traffic flowing through it.

To validate, sign in to one of the VMs that you created in either ASDK environment. Then, ping the VM that you created in the other environment.

To ensure that you send the traffic through the site-to-site connection, ensure that you ping the Direct IP (DIP) address of the VM on the remote subnet, not the VIP. To do so, find the DIP address on the other end of the connection. Save the address for later use.

### Sign in to the tenant VM in POC1

1. Sign in to the Azure Stack Hub physical machine for POC1, and then use a tenant account to sign in to the user portal.
2. In the left navigation bar, select **Compute**.
3. In the list of VMs, find **VM01** that you created previously, and then select it.
4. On the blade for the virtual machine, click **Connect**, and then open the VM01.rdp file.



5. Sign in with the account that you configured when you created the VM.
6. Open an elevated **Windows PowerShell** window.
7. Enter **ipconfig /all**.
8. In the output, find the **IPv4 Address**, and then save the address for later use. This is the address that you'll ping from POC2. In the example environment, the address is **10.0.10.4**, but in your environment it might be different. It should fall within the **10.0.10.0/24** subnet that you created previously.
9. To create a firewall rule that allows the VM to respond to pings, run the following PowerShell command:

```
New-NetFirewallRule `
-DisplayName "Allow ICMPv4-In" `
-Protocol ICMPv4
```

### Sign in to the tenant VM in POC2

1. Sign in to the Azure Stack Hub physical machine for POC2, and then use a tenant account to sign in to the user portal.
2. In the left navigation bar, click **Compute**.
3. From the list of VMs, find **VM02** that you created previously, and then select it.
4. On the blade for the VM, click **Connect**.
5. Sign in with the account that you configured when you created the VM.
6. Open an elevated **Windows PowerShell** window.
7. Enter **ipconfig /all**.
8. An IPv4 address is displayed that falls within **10.0.20.0/24**. In the example environment, the address is **10.0.20.4**, but your address might be different.
9. To create a firewall rule that allows the VM to respond to pings, run the following PowerShell command:

```
New-NetFirewallRule `
-DisplayName "Allow ICMPv4-In" `
-Protocol ICMPv4
```

10. From the VM on POC2, ping the VM on POC1 through the tunnel. To do this, you ping the DIP that you recorded from VM01. In the example environment, this is **10.0.10.4**, but be sure to ping the address you noted in your lab. You should see a result that looks like the following example:

```
c:\Users\localadmin>ping 10.0.10.4

Pinging 10.0.10.4 with 32 bytes of data:
Reply from 10.0.10.4 bytes=32 time=6ms TTL=124
Reply from 10.0.10.4 bytes=32 time=6ms TTL=124
Reply from 10.0.10.4 bytes=32 time=4ms TTL=124
Reply from 10.0.10.4 bytes=32 time=4ms TTL=124

Ping statistics for 10.0.10.4
 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
 Approximate round trip times in milli-seconds:
 Minimum = 4ms, Maximum = 6ms, Average = 5ms

c:\Users\localadmin>_
```

11. A reply from the remote VM indicates a successful test. You can close the VM window. To test your connection, you can try other kinds of data transfers, such as a file copy.

#### **Viewing data transfer statistics through the gateway connection**

If you want to know how much data passes through your site-to-site connection, this information is available on the **Connection** blade. This test is also another way to verify that the ping you just sent actually went through the VPN connection.

1. While you're signed in to the tenant VM in POC2, use your tenant account to sign in to the user portal.
2. Go to **All resources**, and then select the **POC2-POC1** connection. **Connections** appears.
3. In the **Connection** window, the statistics for **Data in** and **Data out** appear. In the following screenshot, the large numbers are attributed to additional file transfer. You should see some nonzero values there.

 POC2-POC1  
Connection

 Settings  Delete

Essentials ^

|                                                                                                                                                  |                                |
|--------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| Resource group                                                                                                                                   | Data in                        |
| <a href="#">testgroup</a>                                                                                                                        | <b>59.74 MB</b>                |
| Status                                                                                                                                           | Data out                       |
| <b>Not connected</b>                                                                                                                             | <b>3.92 GB</b>                 |
| Location                                                                                                                                         | Virtual network                |
| local                                                                                                                                            | <a href="#">VNET-02</a>        |
| Subscription name                                                                                                                                | Virtual network gateway        |
| <a href="#">offer-01</a>                                                                                                                         | <b>GW2 (192.168.112.1)</b>     |
| Subscription ID                                                                                                                                  | Local network gateway          |
| <a href="#">/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/testgroup/providers/Microsoft.Network/networkGateways/POC1-GW</a> | <b>POC1-GW (10.16.167.195)</b> |

[All settings →](#)

# Add Commvault to the Azure Stack Hub Marketplace

2 minutes to read • [Edit Online](#)

This article walks through offering Commvault Live Sync to update a recovery VM located on a separate Azure Stack Hub scale unit. You can download and offer Commvault as a backup and replication solution for your users.

## Notes for Commvault

- Your user needs to install the backup and replication software on a VM in their the Source Azure Stack Hub subscription. Azure Site Recovery and Azure Backup can offer an off-Stack location to store your backups and recovery images. They both require the creation of a Recovery Services Vault in Azure prior to downloading the software images to be installed on your Azure Stack Hub from the following locations: [Azure Backup Server](#) and [Azure Site Recovery](#).
- You may need licenses for third Party Software (if chosen).
- Your users may need assistant in connecting their source and target through a Virtual Network Gateway (VPN) or Public IP on the backup and replication host.
- Target Azure Cloud subscription or subscription on a Recovery Target Azure Stack Hub.
- Target resource group and Blob Storage Account on a Recovery Target Azure Stack Hub.
- Some solutions require that you create virtual machines in the target subscription that need to run 24x7x365 in order to receive changes from the source server. In the [Back up your VM on Azure Stack Hub with Commvault](#), Commvault Live Sync creates the target recovery VMs during initial configuration and keeps them idle (not running, not billable) until changes need to be applied during a replication cycle.

## Get Commvault for your Marketplace

1. Open the Azure Stack Hub Administrative portal.
2. Select **Marketplace management > Add from Azure**.

| NAME            | PUBLISHER |
|-----------------|-----------|
| Commvault Trial | Commvault |

3. Enter `commvault`.
4. Select **Commvault Trial**. And then select **Download**.

## Next steps

[Back up your VM on Azure Stack Hub with Commvault](#)

[Overview of offering services in Azure Stack Hub](#)

# Azure Stack Hub services, plans, offers, subscriptions overview

5 minutes to read • [Edit Online](#)

Microsoft Azure Stack Hub is a hybrid cloud platform that lets you deliver services from your datacenter. Services include virtual machines (VMs), SQL Server databases, SharePoint, Exchange, and even [Azure Marketplace items](#). As a service provider, you can offer services to your tenants. Within a business or government agency, you can offer on-premises services to your employees.

## Overview

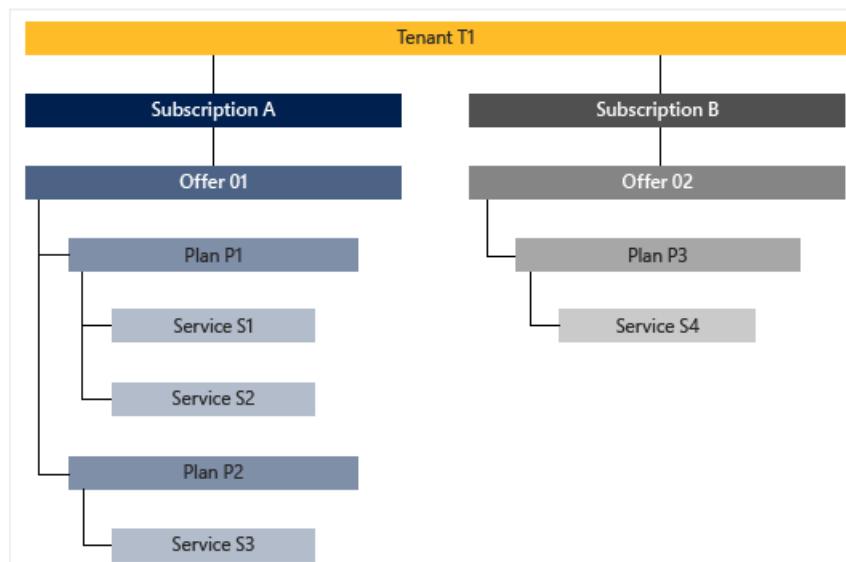
As an Azure Stack Hub operator, you configure and deliver services by using offers, plans, and subscriptions. Offers contain one or more plans, and each plan includes one or more services, each configured with quotas. By creating plans and combining them into different offers, users can subscribe to your offers and deploy resources. This structure lets you manage:

- Which services and resources your users can access.
- The amount of resources that users can consume.
- Which regions have access to the resources.

To deliver a service, follow these high-level steps:

1. Plan your service offering, using:
  - Foundational services, like compute, storage, networking, or Key Vault.
  - Value-add services, like App Service, SQL Server, or MySQL Server.
2. Create a plan that consists of one or more services. When creating a plan, select or create quotas that define the resource limits of each service in the plan.
3. Create an offer that has one or more plans. The offer can include base plans and optional add-on plans.

After you've created the offer, your users can subscribe to it to access the services and deploy resources. Users can subscribe to as many offers as they want. The following figure shows a simple example of a user who has subscribed to two offers. Each offer has a plan or two, and each plan gives them access to specific services.



# Services

You can offer [Infrastructure as a Service](#) (IaaS) services that enable your users to build an on-demand computing infrastructure, provisioned and managed from the Azure Stack Hub user portal.

You can also deploy [Platform as a Service](#) (PaaS) services for Azure Stack Hub from Microsoft and other third-party providers. The PaaS services that you can deliver include, but aren't limited to:

- [App Service](#)
- [SQL Server](#)
- [MySQL Server](#)

You can also combine services to integrate and build complex solutions for different users.

## Quotas

To help manage your cloud capacity, you can use pre-configured *quotas*, or create a new quota for each service in a plan. Quotas define the upper resource limits that a user subscription can provision or consume. For example, a quota might allow a user to create up to five VMs.

### IMPORTANT

It can take up to two hours for new quotas to be available in the user portal or before a changed quota is enforced.

You can set up quotas by region. For example, a plan that provides compute services for Region A could have a quota of two VMs.

### NOTE

In the Azure Stack Development Kit (ASDK), only one region (named *local*) is available.

Learn more about [quota types in Azure Stack Hub](#).

# Plans

Plans are groupings of one or more services. As an Azure Stack Hub operator, you [create plans](#) to offer to your users. In turn, your users subscribe to your offers to use the plans and services they include. When creating plans, make sure to set your quotas, define your base plans, and consider including optional add-on plans.

## Base plan

When creating an offer, the service administrator can include a base plan. These base plans are included by default when a user subscribes to that offer. When a user subscribes, they have access to all the resource providers specified in those base plans (with the corresponding quotas).

## Add-on plans

Add-on plans are optional plans you add to an offer. Add-on plans aren't included by default in the subscription. Add-on plans are additional plans (with quotas) available in an offer that a subscriber can add to their subscriptions. For example, you can offer a base plan with limited resources for a trial, and an add-on plan with more substantial resources to customers who decide to adopt the service.

# Offers

Offers are groups of one or more plans that you create so that users can subscribe to them. For example: Offer Alpha can contain Plan A, which provides a set of compute services, and Plan B, which provides a set of storage and network services.

When you [create an offer](#), you must include at least one base plan, but you can also create add-on plans that users can add to their subscription.

When you're planning your offers, keep the following points in mind:

**Trial offers:** You use trial offers to attract new users, who can then upgrade to additional services. To create a trial offer, create a small [base plan](#) with an optional larger add-on plan. Alternatively, you can create a trial offer consisting of a small base plan, and a separate offer with a larger "pay as you go" plan.

**Capacity planning:** You might be concerned about users who grab large amounts of resources and clog the system for all users. To help performance, you can [configure your plans with quotas](#) to cap usage.

**Delegated providers:** You can grant others the ability to create offers in your environment. For example, if you're a service provider, you can [delegate](#) this ability to your resellers. Or, if you're an organization, you can delegate to other divisions/subsidiaries.

## Subscriptions

Subscriptions let users access your offers. If you're an Azure Stack Hub operator for a service provider, your users (tenants) buy your services by subscribing to your offers. If you're an Azure Stack Hub operator at an organization, your users (employees) can subscribe to the services you offer without paying.

Users create new subscriptions and get access to existing subscriptions by signing in to Azure Stack Hub. Each subscription represents an association with a single offer. The offer (and its plans and quotas) assigned to one subscription can't be shared with other subscriptions. Each resource that a user creates is associated with one subscription.

### Default provider subscription

The default provider subscription is automatically created when you deploy the ASDK. This subscription can be used to manage Azure Stack Hub, deploy additional resource providers, and create plans and offers for users. For security and licensing reasons, it shouldn't be used to run customer workloads and apps. The quota of the default provider subscription can't be changed.

## Next steps

To learn more about creating plans, offers, and subscriptions, start with [Create a plan](#).

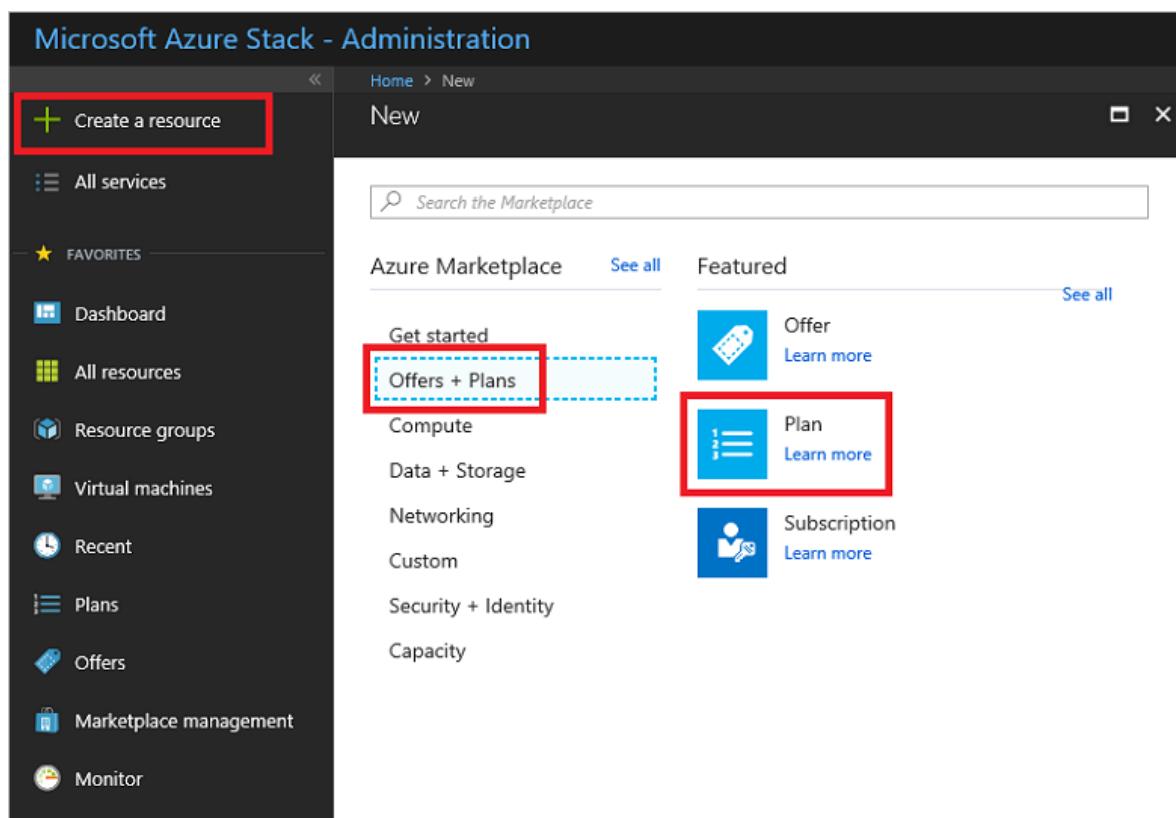
# Create a plan in Azure Stack Hub

3 minutes to read • [Edit Online](#)

Azure Stack Hub plans are groupings of one or more services and their quotas. As a provider, you can create plans to offer to your users. In turn your users subscribe to your offers to use the plans, services, and quotas they include. This example shows you how to create a plan that includes the compute, network, and storage resource providers. This plan gives subscribers the ability to provision virtual machines.

## Create a plan (1902 and later)

1. Sign in to the [Azure Stack Hub administrator portal](#).
2. To create a plan and offer that users can subscribe to, select + **Create a resource**, then **Offers + Plans**, then **Plan**.



3. A tabbed user interface appears that enables you to specify the plan name, add services, and define quotas for each of the selected services. Most importantly, you can review the details of the offer you create before you decide to create it.

Under the **Basics** tab of the **New plan** window, enter a **Display name** and a **Resource name**. The display name is the plan's friendly name that operators can see. In the administrator portal, plan details are only visible to operators.

[Create a resource](#)[Dashboard](#)[All services](#)**FAVORITES**[All resources](#)[Resource groups](#)[Virtual machines](#)[Load balancers](#)[Storage accounts](#)[Virtual networks](#)[Monitor](#)[Offers](#)[Marketplace management](#)[Recent](#)**New plan**

Create a plan to offer to your users.

[Basics](#)[Services](#)[Quotas](#)[Review + create](#)**\* Display name**

Enter the display name that users see

**\* Resource name**

Enter the unique identifier of the plan

Description

**\* Resource group**

Select existing...

[Create new](#)[Review + create](#)[Previous](#)[Next : Services >](#)

4. Create a new **Resource Group**, or select an existing one, as a container for the plan.

Microsoft Azure Stack - Administration

Dashboard > New > New plan

New plan

Create a plan to offer to your users.

Basics Services Quotas Review + create

\* Display name  
Enter the display name that users see

\* Resource name  
Enter the unique identifier of the plan

Description

\* Resource group  
Select existing... Create new

Review + create Previous Next : Services >

The screenshot shows the 'New plan' creation interface in the Microsoft Azure Stack Administration portal. On the left, a sidebar lists various administrative tasks like 'Create a resource', 'Dashboard', and 'All services'. The main area is titled 'New plan' with a sub-instruction 'Create a plan to offer to your users.' Below this are four tabs: 'Basics' (dashed border), 'Services' (solid blue border, indicating selection), 'Quotas', and 'Review + create'. The 'Services' tab contains fields for 'Display name' (placeholder 'Enter the display name that users see') and 'Resource name' (placeholder 'Enter the unique identifier of the plan'). There's also a 'Description' text area and a 'Resource group' section with 'Select existing...' and 'Create new' buttons ('Create new' is highlighted with a red box). At the bottom, there are 'Review + create', 'Previous', and 'Next : Services >' buttons, with 'Next : Services >' also highlighted with a red box.

5. Select the **Services** tab, or click the **Next : Services >** button, and then select the checkbox for **Microsoft.Compute**, **Microsoft.Network**, and **Microsoft.Storage**.

Microsoft Azure Stack - Administration

Home > New > New plan

New plan  
Create a plan to offer to your users.

Basics Services Quotas Review + create

Select one or more services to be offered as part of this plan

5 items

Search to filter items...

SERVICE

|                                     |                         |
|-------------------------------------|-------------------------|
| <input type="checkbox"/>            | Microsoft.Subscriptions |
| <input checked="" type="checkbox"/> | Microsoft.Storage       |
| <input checked="" type="checkbox"/> | Microsoft.Network       |
| <input checked="" type="checkbox"/> | Microsoft.Compute       |
| <input type="checkbox"/>            | Microsoft.KeyVault      |

Review + create Previous Next : Quotas >

The screenshot shows the 'New plan' configuration page. The 'Services' tab is active, indicated by a red box. A list of services is shown, with 'Microsoft.Storage' selected (indicated by a checkmark) and highlighted by a red box. At the bottom right, the 'Next : Quotas >' button is also highlighted with a red box.

6. Select the **Quotas** tab, or click the **Next : Quotas >** button. Next to **Microsoft.Storage**, choose either the default quota from the dropdown box, or select **Create New** to create a customized quota.

Microsoft Azure Stack - Administration

Home > New > New plan

New plan  
Create a plan to offer to your users.

Basics Services Quotas **Review + create**

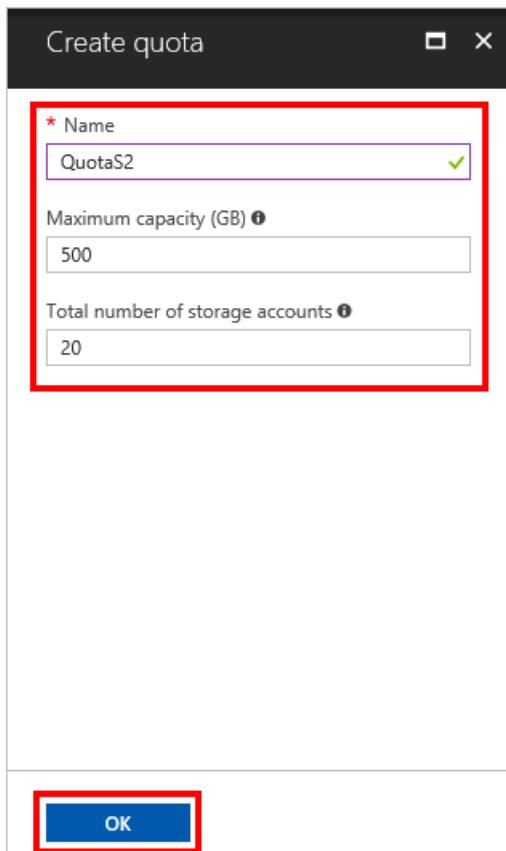
Select a quota for each of the selected services

3 items

|                   |                                           |
|-------------------|-------------------------------------------|
| Microsoft.Storage | <input type="button" value="Create New"/> |
| Microsoft.Network | <input type="button" value="Create New"/> |
| Microsoft.Compute | <input type="button" value="Create New"/> |

Review + create Previous Next : Review + create >

7. If you're creating a new quota, enter a **Name** for the quota, and then specify the quota values. Select **OK** to create the quota.



8. Repeat steps 6 and 7 to create and assign quotas for **Microsoft.Network** and **Microsoft.Compute**.  
When all three services have quotas assigned, they'll look like the next example.

Home > New > New plan

## New plan

Create a plan to offer to your users.

Basics • Services Quotas • Review + create

Select a quota for each of the selected services

3 items

|                   |         |                            |
|-------------------|---------|----------------------------|
| Microsoft.Storage | QuotaS2 | <a href="#">Create New</a> |
| Microsoft.Network | QuotaN2 | <a href="#">Create New</a> |
| Microsoft.Compute | QuotaC2 | <a href="#">Create New</a> |

[Review + create](#) [Previous](#) [Next : Review + create >](#)

The screenshot shows the 'Quotas' step of a 'New plan' wizard. It lists three selected items: Microsoft.Storage (QuotaS2), Microsoft.Network (QuotaN2), and Microsoft.Compute (QuotaC2). Each item has a dropdown menu and a 'Create New' link. Navigation buttons at the bottom include 'Review + create' (highlighted with a red box), 'Previous', and 'Next : Review + create >'.

9. Select **Review + create** to review the plan. Review all values and quotas to ensure they're correct. The interface enables you to expand the quotas in the chosen plans one at a time to view the details of each quota in a plan. You can also go back to make any necessary edits.

Home > New > New plan

## New plan

Create a plan to offer to your users.

Validation passed

Basics Services Quotas Review + create

**BASIC**

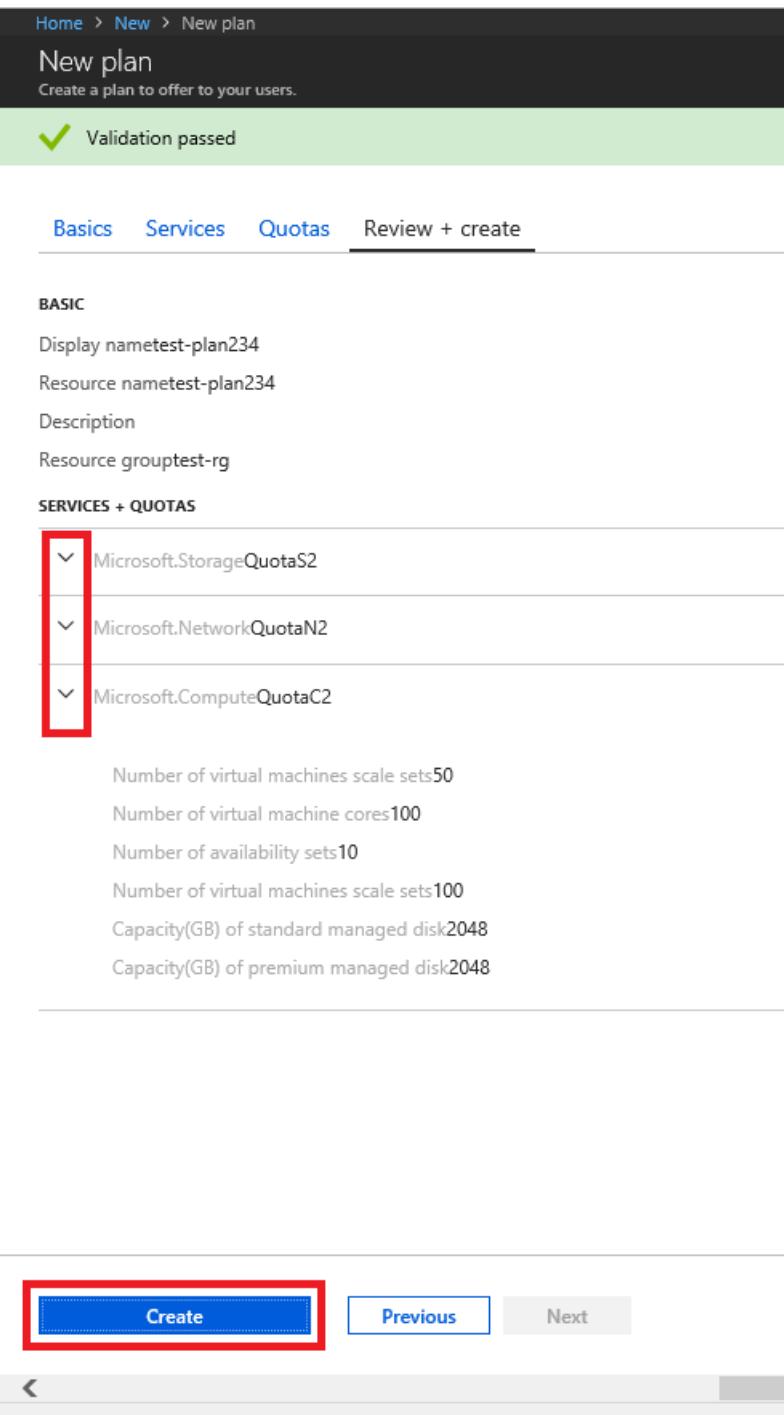
Display name test-plan234  
Resource name test-plan234  
Description  
Resource group test-rg

**SERVICES + QUOTAS**

Microsoft.StorageQuotaS2  
Microsoft.NetworkQuotaN2  
Microsoft.ComputeQuotaC2

Number of virtual machines scale sets 50  
Number of virtual machine cores 100  
Number of availability sets 10  
Number of virtual machines scale sets 100  
Capacity(GB) of standard managed disk 2048  
Capacity(GB) of premium managed disk 2048

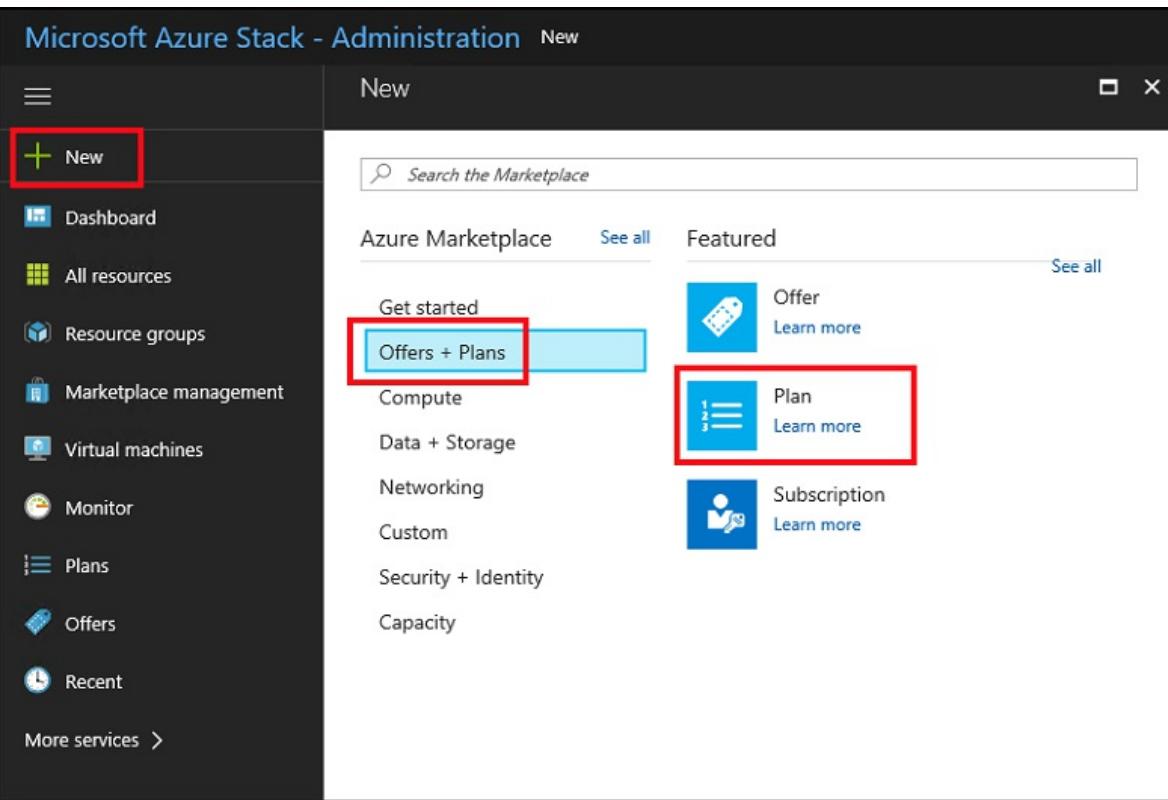
**Create** Previous Next



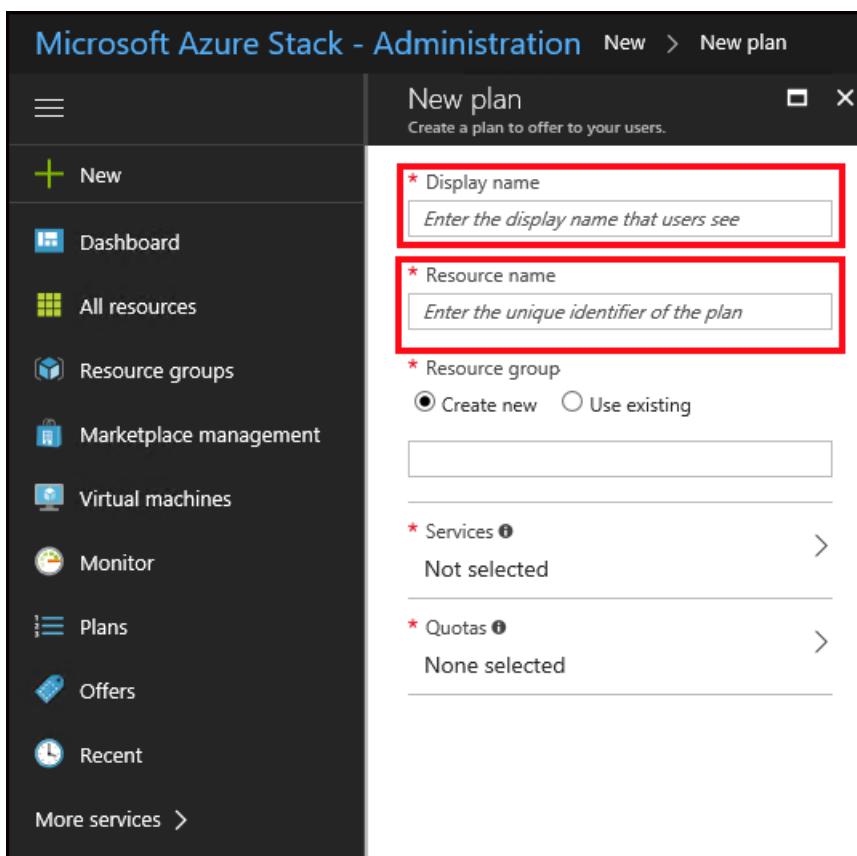
10. When you're ready, select **Create** to create the plan.
11. To see the new plan, on the left-hand side click **All services**, select **Plans**, and then search for the plan and select its name. If your list of resources is long, use **Search** to locate your plan by name.

## Create a plan (1901 and earlier)

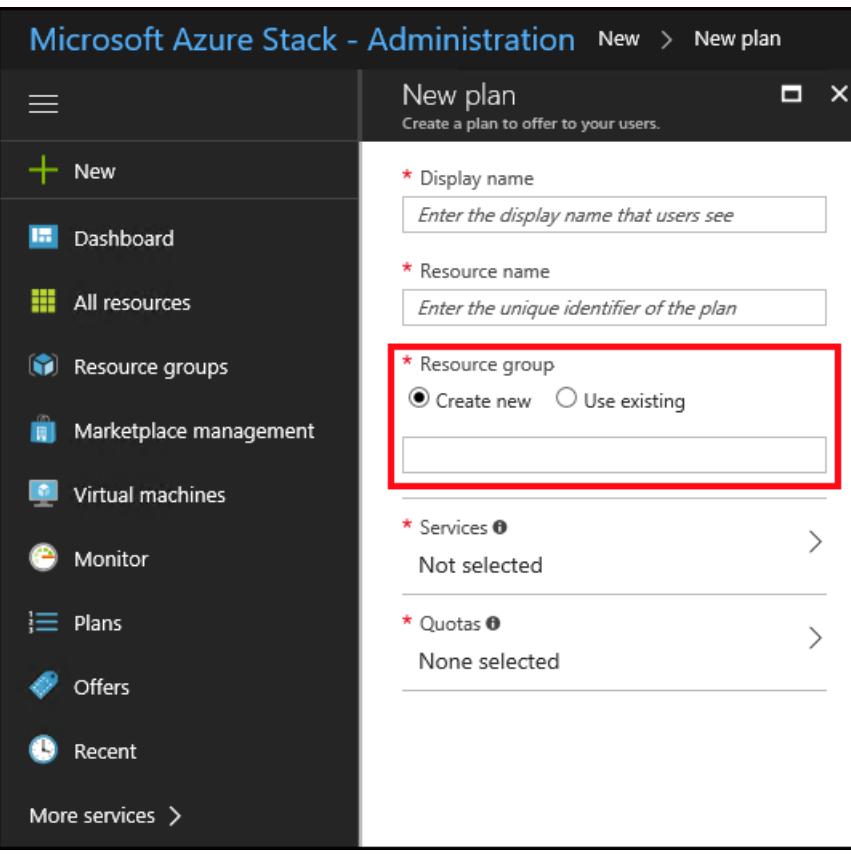
1. Sign in to the [Azure Stack Hub administrator portal](#).
2. To create a plan and offer that users can subscribe to, select + **New**, then **Offers + Plans**, then **Plan**.



3. Under **New plan**, enter a **Display name** and a **Resource name**. The display name is the plan's friendly name that users can see. Only the admin can see the resource name, which admins use to work with the plan as an Azure Resource Manager resource.



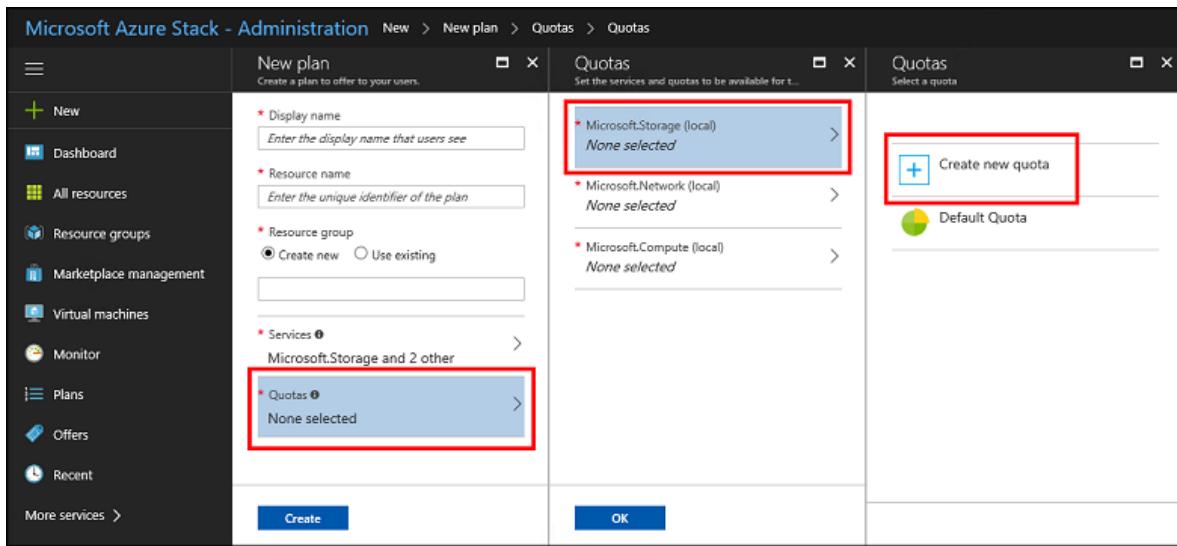
4. Create a new **Resource Group**, or select an existing one, as a container for the plan.



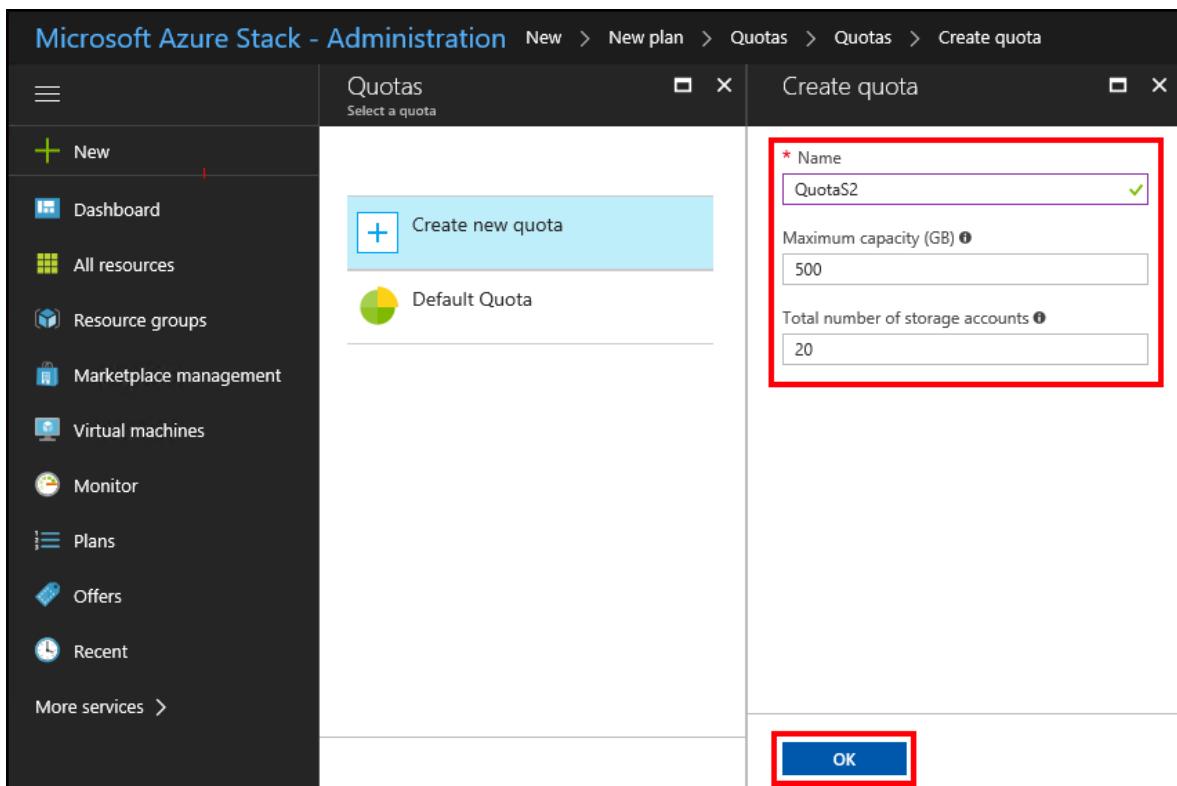
5. Select **Services** and then select the checkbox for **Microsoft.Compute**, **Microsoft.Network**, and **Microsoft.Storage**. Next, choose **Select** to save the configuration. Checkboxes appear when the mouse hovers over each option.

This screenshot continues from the previous one. The 'Services' section in the 'New plan' dialog is now expanded, showing a list of services: Microsoft.Compute (local), Microsoft.KeyVault (local), Microsoft.Network (local), Microsoft.Storage (local), and Microsoft.Subscriptions (local). Each service has a checkbox next to it; Microsoft.Compute, Microsoft.Network, and Microsoft.Storage have their checkboxes checked. The 'Select' button at the bottom right of the 'Services' list is highlighted with a red box. The rest of the 'New plan' dialog remains the same as the first screenshot.

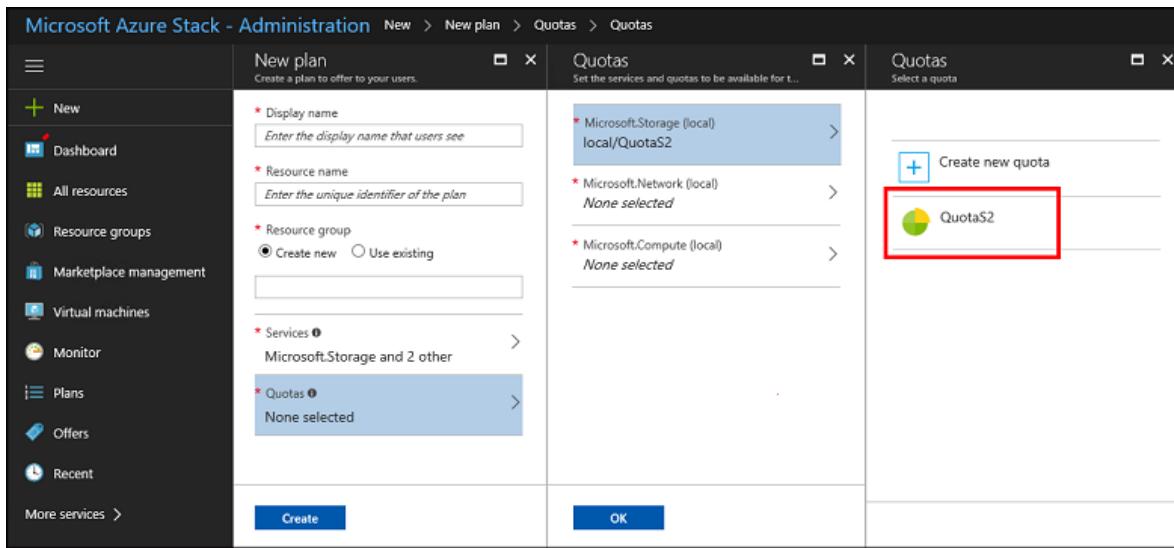
6. Select **Quotas**, **Microsoft.Storage (local)**, and then choose either the default quota or select **Create new quota** to create a customized quota.



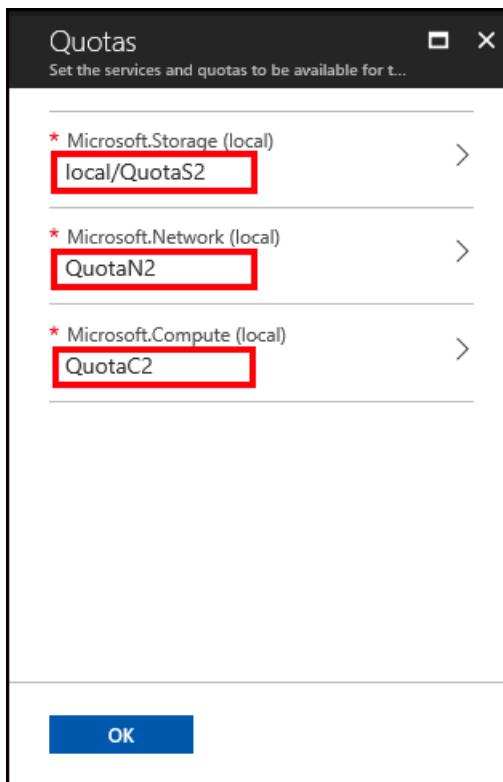
7. If you're creating a new quota, enter a **Name** for the quota > specify the quota values > select **OK**. The **Create quota** dialog closes.



You then select the new quota you created. Selecting the quota assigns it and closes the selection dialog.



8. Repeat steps 6 and 7 to create and assign quotas for **Microsoft.Network (local)** and **Microsoft.Compute (local)**. When all three services have quotas assigned, they'll look like the next example.



9. Under **Quotas**, choose **OK**, and then under **New plan**, choose **Create** to create the plan.

The screenshot shows the Microsoft Azure Stack - Administration interface. On the left, a sidebar lists various management options like Dashboard, All resources, Resource groups, etc. The main area is split into two tabs: 'New plan' and 'Quotas'. The 'New plan' tab contains fields for 'Display name' (Plan1), 'Resource name' (plan1), 'Resource group' (planRG), and lists 'Services' (Microsoft.Storage and 2 other) and 'Quotas' (local/QuotaS2 and 2 other). The 'Quotas' tab lists services with their local quotas: Microsoft.Storage (local) QuotaS2, Microsoft.Network (local) QuotaN2, and Microsoft.Compute (local) QuotaC2. Both the 'Create' button in the 'New plan' tab and the 'OK' button in the 'Quotas' tab are highlighted with red boxes.

- To see your new plan, select **All resources**, then search for the plan and select its name. If your list of resources is long, use **Search** to locate your plan by name.

The screenshot shows the 'All resources' page in Microsoft Azure Stack - Administration. The left sidebar shows 'All resources' selected. The main pane displays a list of resources under 'plan1'. One resource, named 'plan', is highlighted with a red box. The right pane provides detailed information about this resource, including its 'Overview', 'Activity log', and 'Access control (IAM)' sections. It also shows 'SETTINGS' like 'Plan settings', 'Services and quotas', and 'Parent offers'. A chart titled 'New subscriptions over time' tracks the number of subscriptions from April 13 to April 17, showing a steady increase from 0 to 100.

## Next steps

- [Create an offer](#)

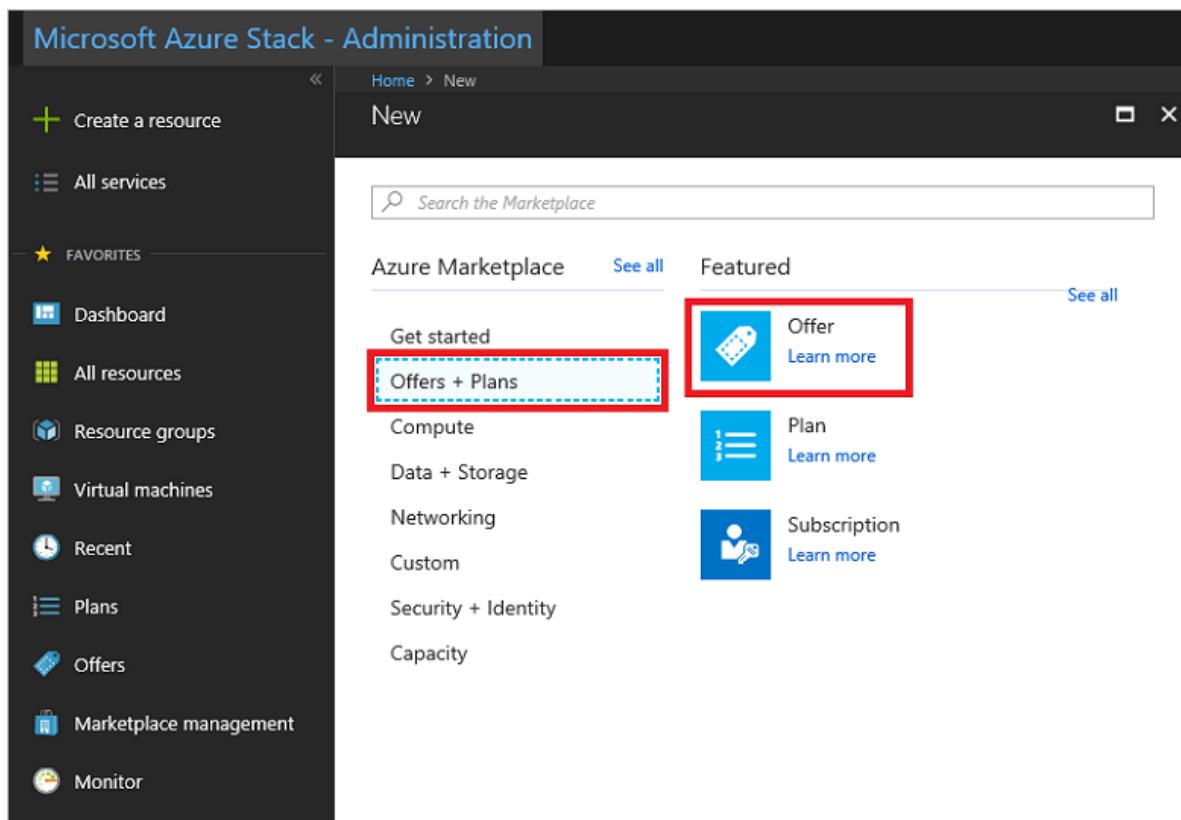
# Create an offer in Azure Stack Hub

4 minutes to read • [Edit Online](#)

Offers are groups of one or more plans that providers present to users, which those users can buy or subscribe to. This article describes how to create an offer that includes the [plan that you created](#). This offer gives subscribers the ability to set up virtual machines (VMs).

## Create an offer (1902 and later)

1. Sign in to the [Azure Stack Hub administrator portal](#) and select + **Create a resource**, then **Offers + Plans**, and then **Offer**.



2. A tabbed user interface appears that enables you to define the offer name. You can also add existing or create new base plans and add-on plans. Most importantly, you can review the details of the offer you create before you decide to create it.

In the **Basics** tab, enter a **Display Name** and a **Resource Name**, and then under **Resource Group**, select **Create new** or **Use existing**. The display name is the friendly name for the offer. This friendly name is the only information about the offer that users see when they subscribe to an offer in the user portal. Use an intuitive name that helps users understand what comes with the offer. Only the admin can see the resource name. It's the name that admins use to work with the offer as an Azure Resource Manager resource. In this tab, you can also choose to make this offer public or keep it private. The default setting is private. You can change the public or private state of the offer at any time.

## Create a new offer

Create a new offer for your users

Basics

Base plans

Add-on plans

Review + create

\* Display name ⓘ

Enter the display name that users see

\* Resource name

Enter the unique identifier of the offer

Description

\* Resource group

Select existing...

Create new

Make this offer public?

Yes

No

Review + create

Previous

Next : Base plans >



3. Select the **Base plans** tab or click the **Next : Base plans >** button. Select the plan(s) you want to include in the offer.

## Create a new offer

Create a new offer for your users

Basics Base plans Add-on plans Review + create

Select any plans that should be made available immediately to a user subscribing to this offer

[Create new plan](#)

1 items

Search to filter items...

| <input checked="" type="checkbox"/> | DISPLAY NAME                                                                            | DESCRIPTION |
|-------------------------------------|-----------------------------------------------------------------------------------------|-------------|
| <input checked="" type="checkbox"/> |  plan9 |             |

[Review + create](#)

[Previous](#)

[Next : Add-on plans >](#)



- At this point you can create an add-on plan to modify the base plan, but this is optional. You have the opportunity to create an add-on plan in the next article, [Azure Stack Hub add-on plans](#).
- Select the **Review + create** tab. Review the offer summary to ensure that all values are correct. The interface enables you to expand the quotas in the chosen plans one at a time to view the details of each quota in a plan. You can also go back to make any necessary edits.
- Select **Create** to create the offer.

Dashboard > New > Create a new offer

## Create a new offer

Create a new offer for your users

Validation passed

Basics Base plans Add-on plans **Review + create**

**BASIC**

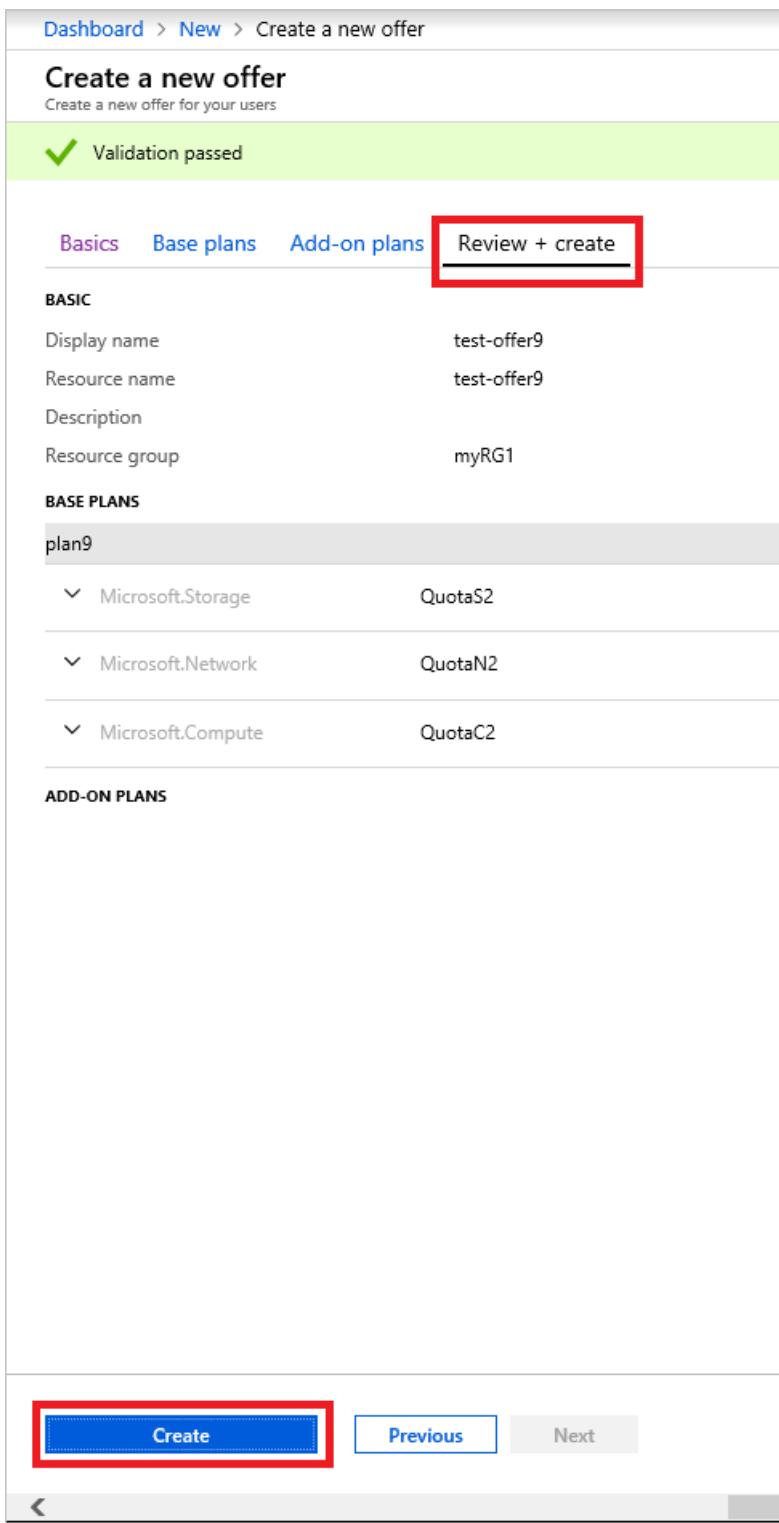
|                |             |
|----------------|-------------|
| Display name   | test-offer9 |
| Resource name  | test-offer9 |
| Description    |             |
| Resource group | myRG1       |

**BASE PLANS**

|                     |         |
|---------------------|---------|
| plan9               |         |
| ▼ Microsoft.Storage | QuotaS2 |
| ▼ Microsoft.Network | QuotaN2 |
| ▼ Microsoft.Compute | QuotaC2 |

**ADD-ON PLANS**

**Create** **Previous** **Next**



### Change the state of an offer

After creating the offer, you can change its state. Offers must be made **Public** for users to get the full view when they subscribe. Offers can be:

- **Public:** Visible to users.
- **Private:** Only visible to cloud administrators. This setting is useful while drafting the plan or offer, or if the cloud administrator wants to [create each subscription for users](#).
- **Decommissioned:** Closed to new subscribers. The cloud administrator can decommission offers to prevent future subscriptions, but leave current subscribers unaffected.

**TIP**

Changes to the offer aren't immediately visible to the user. To see the changes, users might have to sign out and sign in again to the user portal to see the new offer.

There are two ways to change the state of an offer:

1. In **All resources**, select the name of the offer. On the **Overview** screen for the offer, select **Change state**. Choose the state you want to use (for example, **Public**).

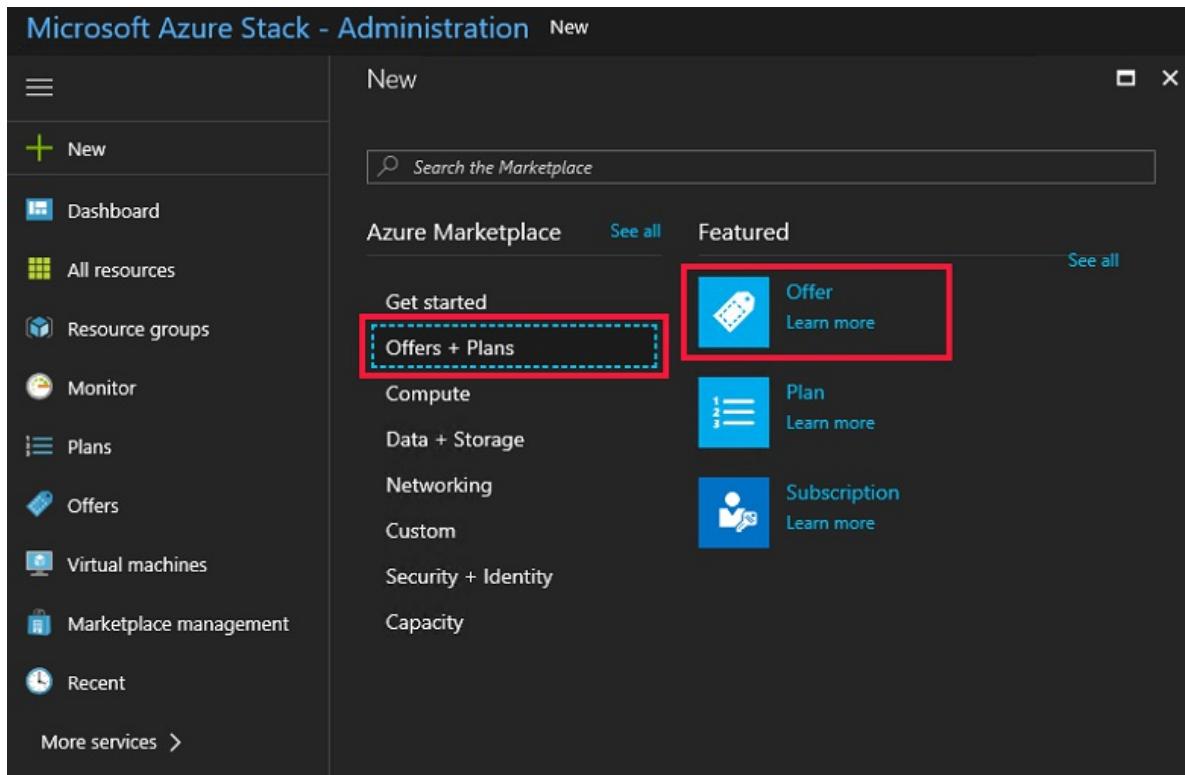
The screenshot shows the Azure portal interface for managing an offer named 'test-offer9'. On the left, there's a sidebar with navigation links: Overview, Activity log, Access control (IAM), Settings (Delegated providers, Offer settings, Properties, Locks), Users (Subscriptions), and Plans (Base plans, Add-on plans). The main area is titled 'test-offer9' and shows basic details: Resource group 'myRG1', Status 'Public', Location 'local', Subscription 'Default Provider Subscription', and a summary of 0 subscriptions, 1 base plan, and 0 add-on plans. Below this, there's a chart titled 'Subscriptions created over the last week' with a scale from 0 to 100. At the top of the main area, there are buttons for 'Clone', 'Change state' (which is highlighted with a red box), and 'Delete'. A dropdown menu for 'Change state' is open, showing options: Public (selected), Private, and Decommissioned.

2. Select **Offer settings**. Choose the state you want to use (for example, **Public**), then select **Save**.

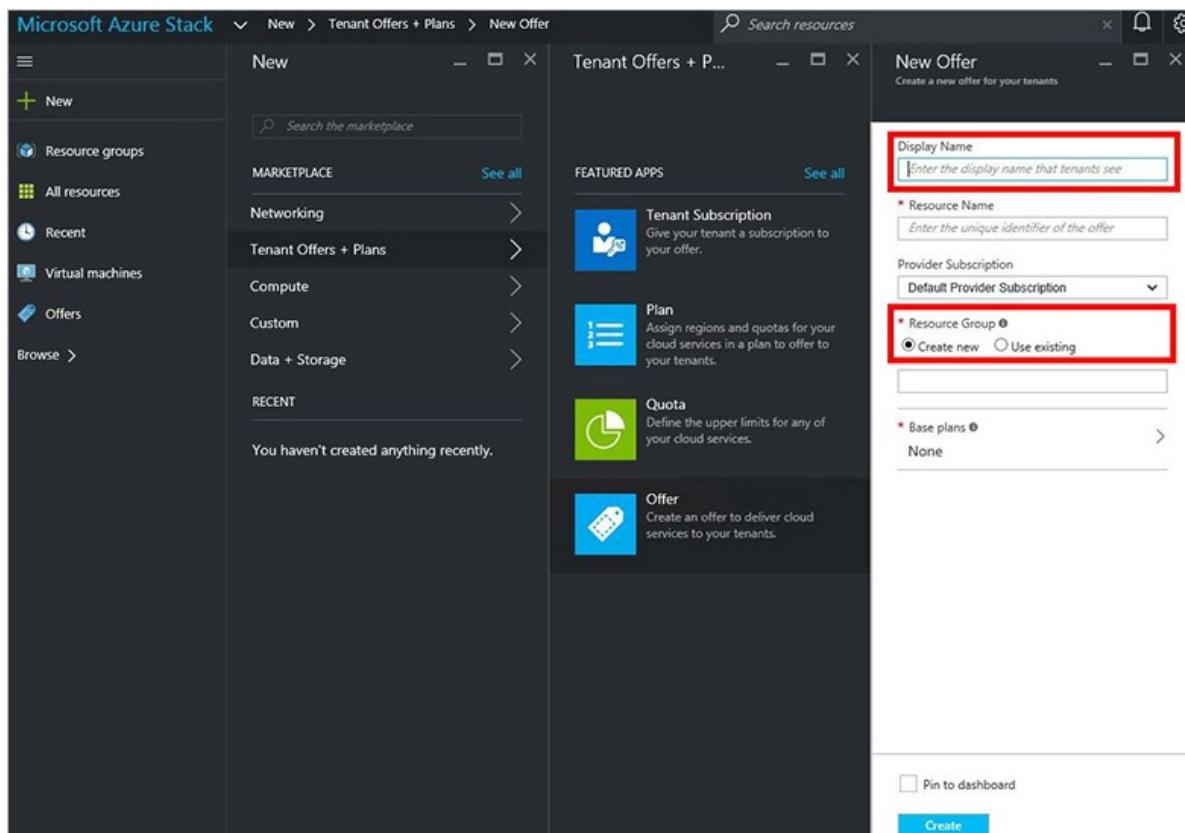
The screenshot shows the 'Offer settings' page for 'test-offer9'. The left sidebar lists navigation options: Overview, Activity log, Access control (IAM), Settings (selected), Delegated providers, Offer settings (highlighted with a red box), Properties, Locks, Users, Subscriptions, Plans, Base plans, and Add-on plans. The main area has a 'Save' button highlighted with a red box at the top right. Below it, the 'Display name' field contains 'test-offer9'. The 'Description' field is empty. A 'State' section shows three buttons: 'Public' (blue), 'Private' (white), and 'Decommissioned' (blue). The 'Decommissioned' button is highlighted with a red box.

## Create an offer (1901 and earlier)

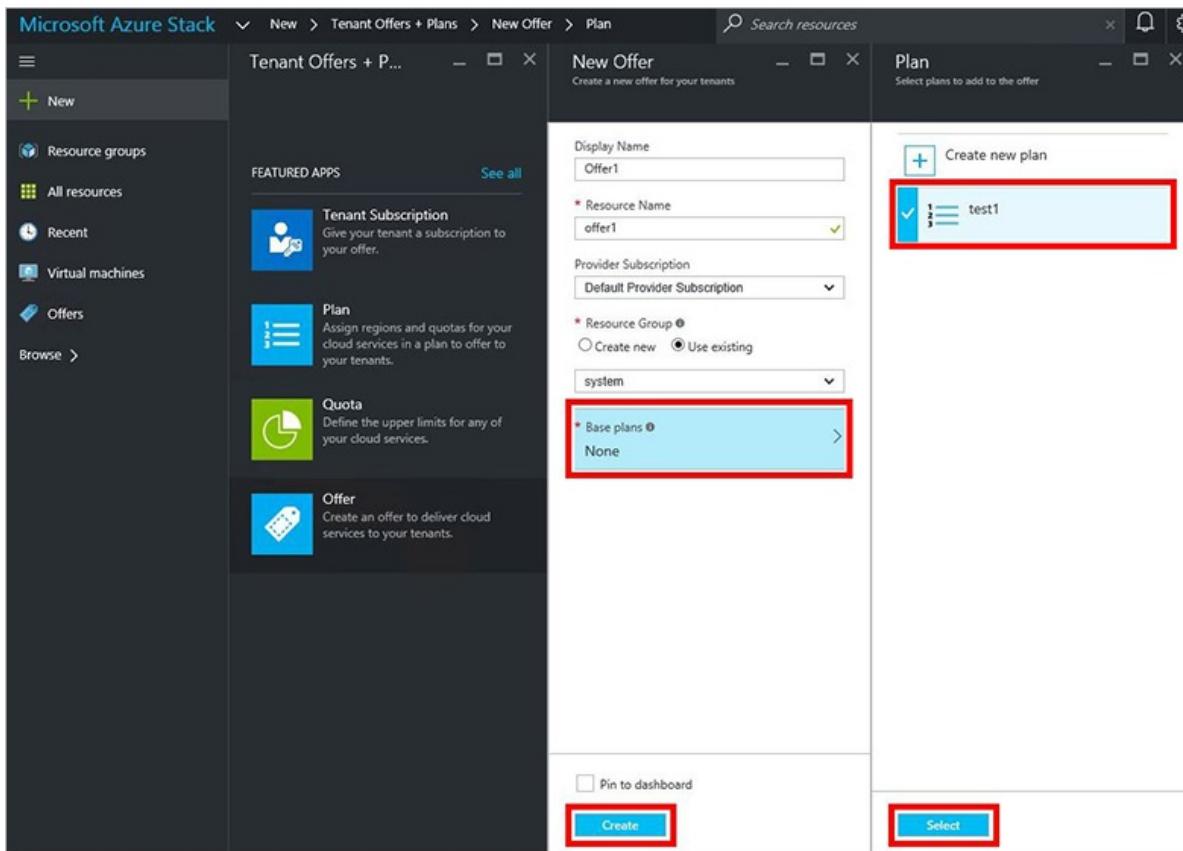
1. Sign in to the [Azure Stack Hub administrator portal](#) and select + **Create a resource**, then **Tenant Offers** + **Plans**, and then **Offer**.



- Under **New Offer**, enter a **Display Name** and a **Resource Name**, and then under **Resource Group**, select **Create new** or **Use existing**. The display name is the friendly name for the offer. This friendly name is the only information about the offer that users see when they subscribe to an offer. Use an intuitive name that helps users understand what comes with the offer. Only the admin can see the resource name. It's the name that admins use to work with the offer as an Azure Resource Manager resource.



- Select **Base plans** to open the **Plan**. Select the plans you want to include in the offer, and then choose **Select**. To create the offer, select **Create**.



4. After creating the offer, you can change its state. Offers must be made **Public** for users to get the full view when they subscribe. Offers can be:

- **Public:** Visible to users.
- **Private:** Only visible to cloud administrators. This setting is useful while drafting the plan or offer, or if the cloud administrator wants to [create each subscription for users](#).
- **Decommissioned:** Closed to new subscribers. The cloud administrator can decommission offers to prevent future subscriptions, but leave current subscribers unaffected.

#### TIP

Changes to the offer aren't immediately visible to the user. To see the changes, users might have to sign out and sign in again to the user portal to see the new offer.

On the overview screen for the offer, select **Accessibility state**. Choose the state you want to use (for example, **Public**), and then select **Save**.

The screenshot shows the 'test\_offer' resource group in the Azure Stack Hub portal. On the left, there's a navigation sidebar with options like Overview, Activity log, Access control (IAM), SETTINGS (Delegated providers, Offer settings, Properties, Locks), USERS (Subscriptions), and PLANS (Base plans, Add-on plans). The main area shows the 'Overview' tab for the 'test\_offer' resource group. It includes details such as Resource group (test\_rg), Status (--), Location (local), Subscription (Default Provider Subscription), and a Subscription ID. A warning message at the top says 'This offer is private and users cannot see it'. To the right, there's a section for 'Subscriptions created over the last week' with a bar chart showing values from 0 to 100. The 'Accessibility state' field is highlighted with a red box.

As an alternative, select **Change state** and then choose a state.

This screenshot shows the same 'test\_offer' resource group overview as the previous one, but with a different accessibility setting. The 'Change state' button was used to set the accessibility state to 'Public'. The 'Accessibility state' field is highlighted with a red box. The rest of the interface is identical to the first screenshot.

#### NOTE

You can also use PowerShell to create default offers, plans, and quotas. For more information, see [Azure Stack Hub PowerShell Module 1.4.0](#).

## Next steps

- To learn how to modify an offer and provide your users with an add-on plan, continue with [Create an add-on plan](#) (optional)
- Otherwise, jump to [Subscribe to an offer](#)

# Create add-on plans in Azure Stack Hub

2 minutes to read • [Edit Online](#)

As an Azure Stack Hub operator, you create add-on plans to modify a [base plan](#) when you want to offer additional services or extend *computer*, *storage*, or *network* quotas beyond the base plan initial offer. Add-on plans modify the base plan and are optional extensions that users can choose to enable in their subscription.

There are times when combining everything in a single plan is optimal. Other times you might want to have a base plan and then offer the additional services by using add-on plans. For instance, you could decide to offer IaaS services as part of a base plan with all PaaS services treated as add-on plans.

Another reason to use add-on plans is to help monitor resource usage. To do so, you could start with a base plan that includes relatively small quotas (depending on the services required). Then, as users reach capacity, they would be alerted that they've consumed the allocation of resources based on their assigned plan. From there, the users can select an add-on plan that provides the additional resources.

## NOTE

When you don't want to use an add-on plan to extend a quota, you can also choose to [edit the original configuration of the quota](#).

Add-on plans are [created the same way](#) as a base plan. The difference between the two is determined when the plan is added to an offer. It's designated as either a base plan or add-on plan. When you add an add-on plan to an existing offer, the additional resources can take up to an hour to appear in the subscription.

## Create an add-on plan (1902 and later)

1. Sign in to the Azure Stack Hub administrator portal as a cloud administrator.
2. Follow the same steps used to [create a new base plan](#) to create a new plan offering services that weren't previously offered.
3. In the administrator portal, select **Offers** and then select the offer to be updated with an add-on plan.

The screenshot shows the Azure Stack Hub administrator portal interface. On the left, a sidebar lists various management options like 'Create a resource', 'Dashboard', 'All services', 'Offers', 'Resource groups', 'Virtual machines', 'Load balancers', 'Storage accounts', 'Virtual networks', 'Monitor', 'Marketplace management', and 'Recent'. The 'Offers' link is highlighted with a red box. The main content area is titled 'Offers' and shows a table with one item: 'test-offer9'. The table has columns for NAME, STATE, BASE PLANS, ADD-ON PLANS, SUBSCRIPTIONS, and RESOURCE GROUP. The 'test-offer9' row is also highlighted with a red box. At the top of the main area, there are buttons for '+ Add', 'Edit columns', and 'Refresh', along with a 'Search resources' bar and filter options for 'Subscriptions', 'Filter by name...', 'All subscriptions', 'All resource groups', and 'All tags'.

4. At the bottom of the offer properties, select **Add-on plans**. Select **Add**.

The screenshot shows the Azure portal interface for managing add-on plans. On the left, there's a sidebar with 'Offers' selected. In the main area, the title is 'test-offer9 - Add-on plans'. A red box highlights the 'Add-on plans' link under the 'Plan' section of the navigation menu.

5. Select the plan to add. In this example, the plan is called **20-storageaccounts**. After selecting the plan, click **Select** to add the plan to the offer. You should receive a notification that the plan was added to the offer successfully.

The screenshot shows the 'Plan' page for the 'test-offer9' offer. It lists available add-on plans. The row for 'plan9' is selected, indicated by a checked checkbox and highlighted with a red box. At the bottom of the table, a large blue 'Select' button is also highlighted with a red box.

| DISPLAY NAME | SERVICES |
|--------------|----------|
| plan9        | 3        |

6. Review the list of add-on plans included with the offer to verify that the new add-on plan is listed.

The screenshot shows the Microsoft Azure Stack - Administration portal. In the left sidebar, under the 'Offers' section, the 'Offers' link is highlighted with a red box. In the main content area, the 'Add-on plans' link is also highlighted with a red box. The central pane displays a table titled 'test-offer9 - Add-on plans' with one item listed:

| NAME  | SERVICES | RESOURCE GROUP | ALLOWED ACQUISITIONS |
|-------|----------|----------------|----------------------|
| plan9 | 3        | myRG1          | 1                    |

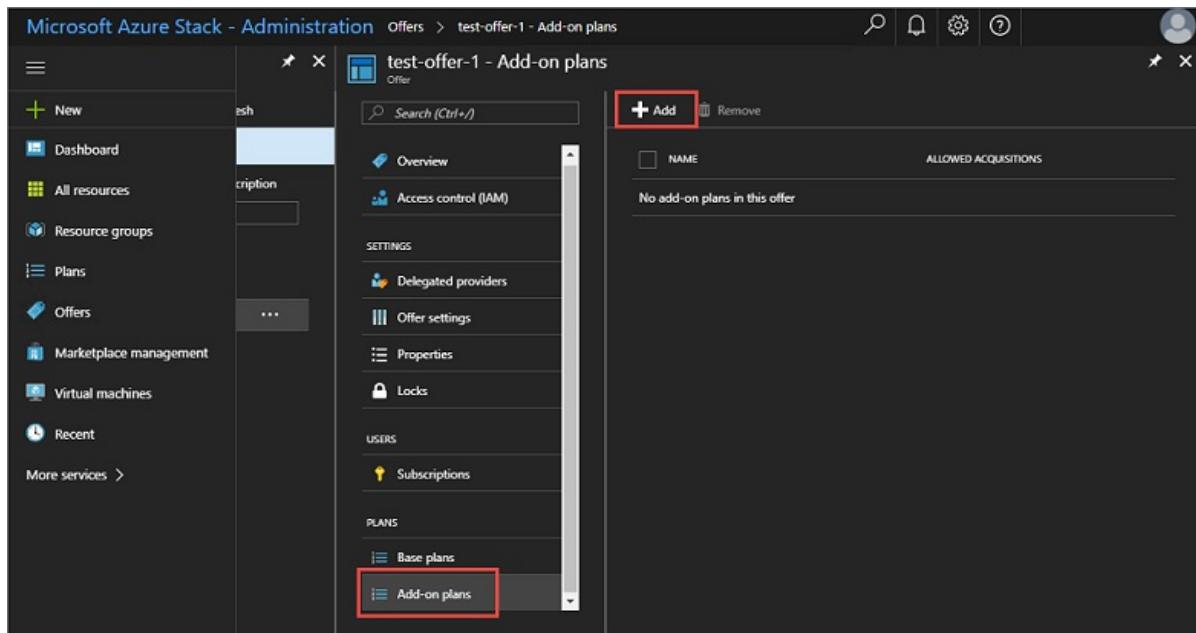
## Create an add-on plan (1901 and earlier)

1. Sign in to the Azure Stack Hub administrator portal as a cloud administrator.
2. Follow the same steps used to [create a new base plan](#) to create a new plan offering services that weren't previously offered. In this example, Key Vault (**Microsoft.KeyVault**) services will be included in the new plan.
3. In the administrator portal, select **Offers** and then select the offer to be updated with an add-on plan.

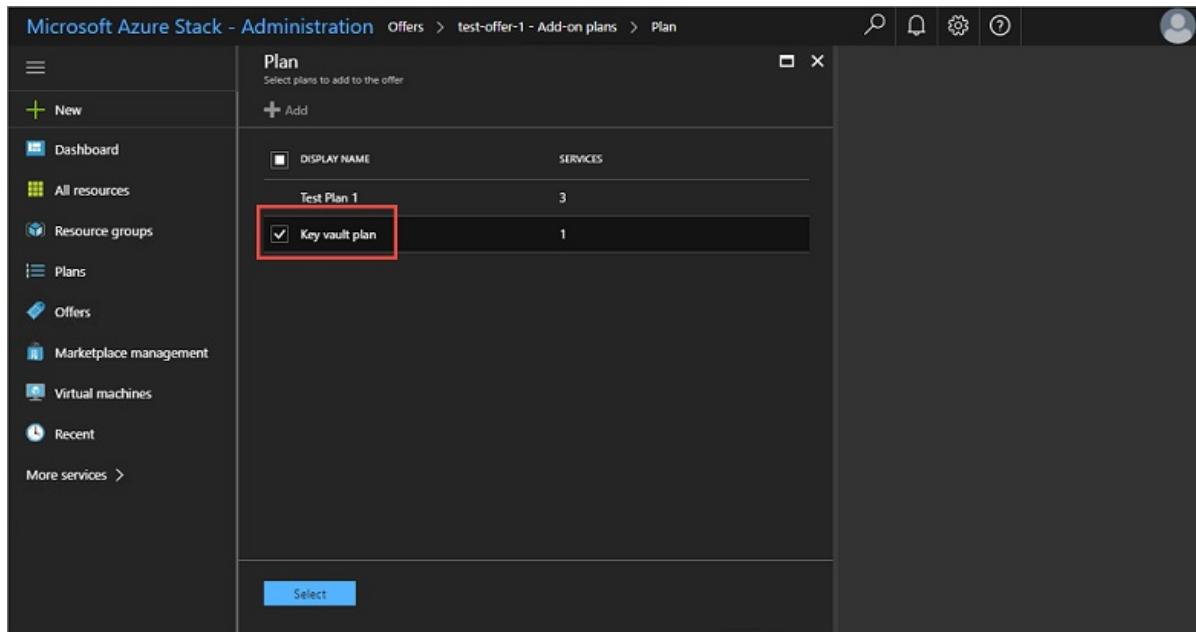
The screenshot shows the Microsoft Azure Stack - Administration portal. In the left sidebar, the 'Offers' link is highlighted with a red box. The main content area displays a table titled 'Offers' with one item listed:

| NAME         | TYPE  | RESOURCE GROUP | LOCATION | SUBSCRIPTION                   |
|--------------|-------|----------------|----------|--------------------------------|
| test-offer-1 | Offer | test-offer-1   | local    | Default Provider Subscr... *** |

4. Scroll to the bottom of the offer properties and select **Add-on plans**. Select **Add**.



5. Select the plan to add. In this example, the plan is called **Key vault plan**. After selecting the plan, click **Select** to add the plan to the offer. You should receive a notification that the plan was added to the offer successfully.



6. Review the list of add-on plans included with the offer to verify that the new add-on plan is listed.

Microsoft Azure Stack - Administration Offers > test-offer-1 - Add-on plans

New

Dashboard

All resources

Resource groups

Plans

Offers

Marketplace management

Virtual machines

Recent

More services >

test-offer-1 - Add-on plans

Offer

Search (Ctrl+ /)

Add Remove

| NAME           | ALLOWED ACQUISITIONS |
|----------------|----------------------|
| Key vault plan | 1                    |

Overview

Access control (IAM)

Delegated providers

Offer settings

Properties

Locks

Subscriptions

Base plans

Add-on plans

The screenshot shows the Microsoft Azure Stack Administration interface. On the left, there's a navigation sidebar with various options like Dashboard, All resources, Resource groups, etc. The main area is titled 'test-offer-1 - Add-on plans' under the 'Offer' category. It contains a table with one row, 'Key vault plan', which is highlighted with a red box. The table has columns for 'NAME' and 'ALLOWED ACQUISITIONS'. There are also tabs for Overview, Access control (IAM), Delegated providers, Offer settings, Properties, Locks, and Subscriptions.

## Next steps

- [Create an offer](#)

# Create subscriptions to offers in Azure Stack Hub

3 minutes to read • [Edit Online](#)

After you [create an offer](#), users need a subscription to that offer before they can use it. There are two ways that users can subscribe to an offer:

- As a cloud operator, you can create a subscription for a user from within the administrator portal. Subscriptions you create can be for both public and private offers.
- As a tenant user, you can subscribe to a public offer when you use the user portal.

## Create a subscription as a cloud operator

Cloud operators use the administrator portal to create a subscription to an offer for a user. Subscriptions can be created for members of your own directory tenant. When [multi-tenancy](#) is enabled, you can also create subscriptions for users in additional directory tenants.

If you don't want your tenants to create their own subscriptions, make your offers private, and then create subscriptions for your tenants. This approach is common when integrating Azure Stack Hub with external billing or service catalog systems.

After you create a subscription for a user, they can sign in to the user portal and see that they're subscribed to the offer.

### To create a subscription for a user

1. In the administrator portal, go to **User subscriptions**.
2. Select **Add**. Under **New user subscription**, enter the following information:
  - **Display name** - A friendly name for identifying the subscription that appears as the *User subscription name*.
  - **User** - Specify a user from an available directory tenant for this subscription. The user name appears as *Owner*. The format of the user name depends on your identity solution. For example:
    - **Azure AD:** <user1>@<contoso.onmicrosoft.com>
    - **AD FS:** <user1>@<azurestack.local>
  - **Directory tenant** - Select the directory tenant where the user account belongs. If you haven't enabled multi-tenancy, only your local directory tenant is available.
3. Select **Offer**. Under **Offers**, choose an **Offer** for this subscription. Because you're creating the subscription for a user, select **Private** as the accessibility state.
4. Select **Create** to create the subscription. The new subscription appears under **User subscription**. When the user signs in to the user portal, they can see the subscription details.

### To make an add-on plan available

A cloud operator can add a plan to a previously created subscription at any time:

1. In the administrator portal, select **All Services** and then under the **ADMINISTRATIVE RESOURCES** category, select **User subscriptions**. Select the subscription you want to change.
2. Select **Add-ons** and then select **+Add**.

3. Under **Add plan**, select the plan you want as an add-on.

## Create a subscription as a user

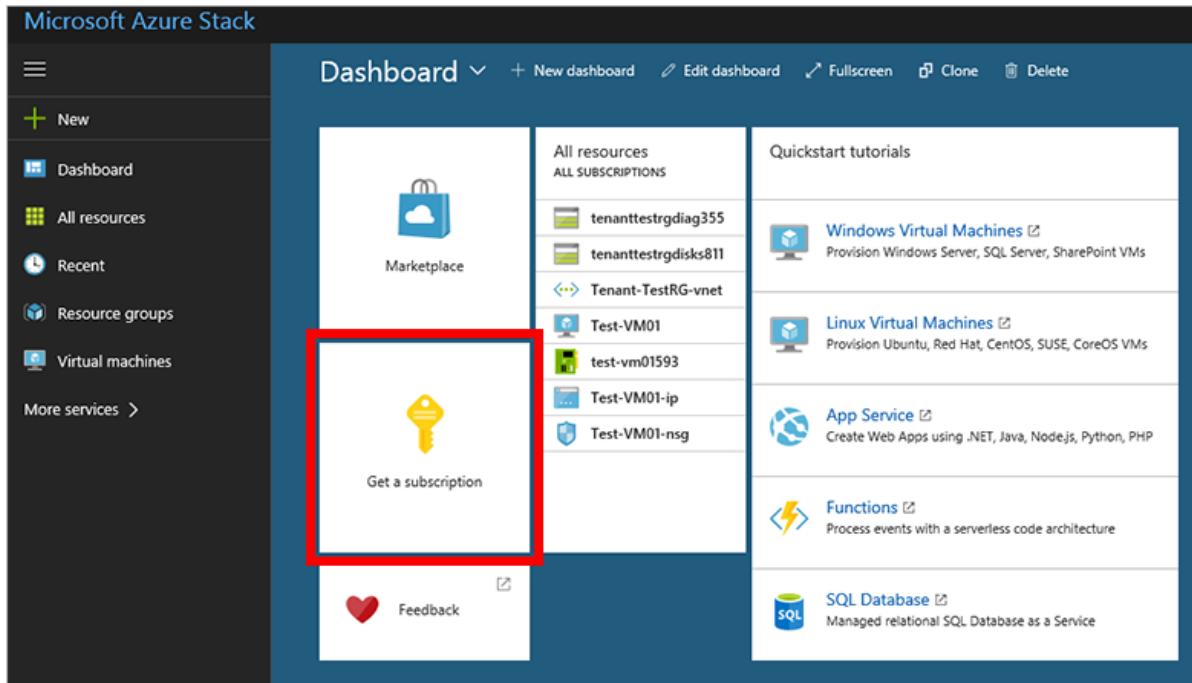
As a user, you can sign in to the user portal to locate and subscribe to public offers and add-on plans for your directory tenant (organization).

### NOTE

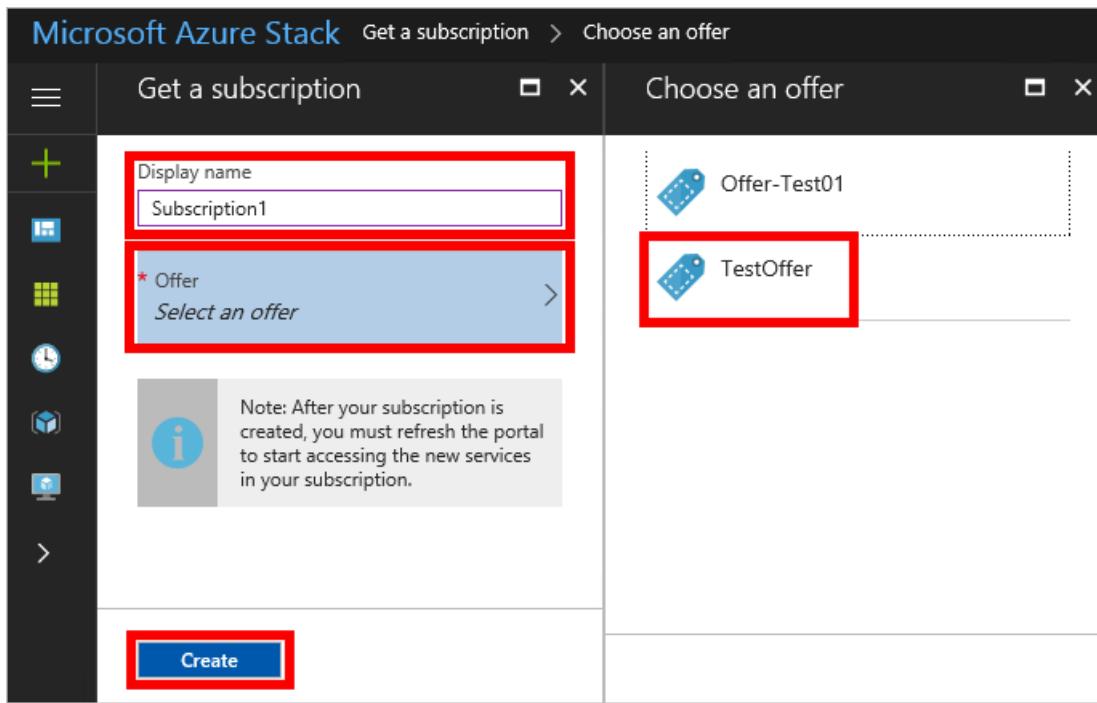
If your Azure Stack Hub environment supports [multi-tenancy](#), you can also subscribe to offers from a remote directory tenant.

### To subscribe to an offer

1. Sign in to the Azure Stack Hub user portal and select **Get a Subscription**.



2. Under **Get a subscription**, enter the friendly name of the subscription in **Display Name**. Select **Offer** and under **Choose an offer**, pick an offer. Select **Create** to create the subscription.



3. After you subscribe to an offer, refresh the portal to see which services are part of the new subscription.
4. To see the subscription you created, select **All services** and then under the **GENERAL** category select **Subscriptions**. Select the subscription to see the subscription details.

#### To enable an add-on plan in your subscription

If the offer you subscribe to has an add-on plan, you can add that plan to your subscription at any time.

1. In the user portal, select **All services**. Next, under the **GENERAL** category, select **Subscriptions**, and then select the subscription that you want change. If there are add-on plans available, **+ Add plan** is active and shows a tile for **Add-on plans**.

If **+ Add plan** isn't active, then there are no add-on plans for the offer associated with that subscription.

2. Select **+ Add plan** or the **Add-on plans** tile. Under **Add-on plans**, select the plan you want to add.

## Next steps

Learn more about how a user can now deploy resources into their subscription:

- [Several user quickstarts](#) show how to provision Windows and Linux virtual machines using PowerShell, Azure CLI, and the user portal.
- [A tutorial that uses an Azure Resource Manager template](#) shows how to deploy an Ubuntu 16.04 virtual machine running Minikube to manage Kubernetes cluster.

# Delete quotas, plans, offers, and subscriptions

2 minutes to read • [Edit Online](#)

This article describes how to delete quotas, plans, offers, and subscriptions that you no longer need. As a general principle, you can delete only what is not in use. For example, deleting an offer is only possible if there are no subscriptions that belong to that offer.

Subscriptions are the exception to this general principle: you can delete subscriptions that contain resources; and the resources will be deleted along with the subscription.

Therefore, if you want to delete a quota, you must work back through any plans and offers that use that quota: starting with the offers, ensure they have no subscriptions, delete each offer, then delete the plans that use the quota, and so on.

## Delete a subscription

To delete a subscription, select **All services**, then **User subscriptions**, to display a list of all subscriptions on the system. If you are working on an offer, you can also select **Subscriptions** from there.

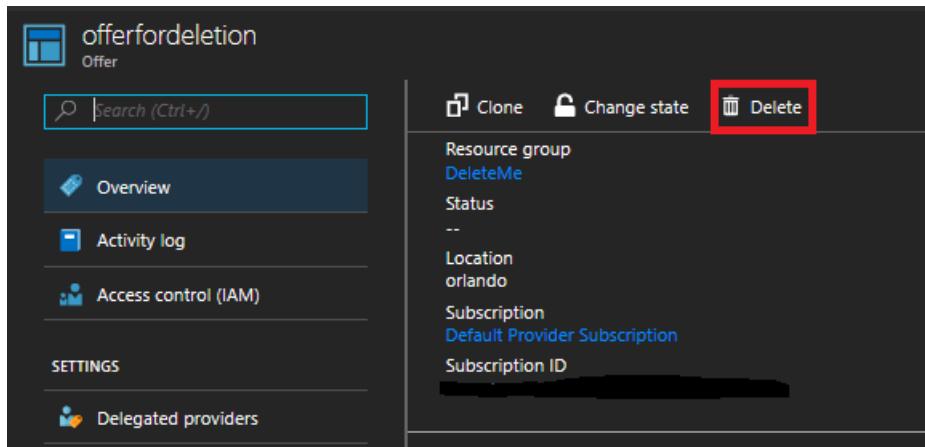
You can delete subscriptions from this list, or you can use PowerShell to write a script that deletes all subscriptions for you, using the commands documented in the [Subscriptions - Delete reference](#).

**Caution**

Deleting a subscription also deletes any data and resources it contains.

## Delete an offer

To delete an offer, in the administrator portal, go to **All services**, then **Offers**. Select the offer you want to delete, then select **Delete**.



You can only delete an offer when there are no subscriptions using it. If subscriptions exist based on the offer, the **Delete** option is not available. In this case, see the [Delete a subscription](#) section.

## Delete a plan

To delete a plan, in the administrator portal go to **All services**, then **Plans**. Select the plan you want to delete, then select **Delete**.

The screenshot shows the Azure portal interface for a resource plan named 'trial-plan'. On the left, there's a sidebar with various navigation links: Overview, Activity log, Access control (IAM), Plan settings, Services and quotas, Parent offers (which is highlighted with a red box), Properties, and Locks. The main content area displays the plan's details: Resource group (TrialPlanRG), Status (--), Location (orlando), Subscription (Default Provider Subscription), and Subscription ID (redacted). Below this, there's a chart titled 'New subscriptions over time' showing a downward trend from 100 to 60. At the top right of the main content area, there are 'Clone' and 'Delete' buttons, with the 'Delete' button also highlighted with a red box.

You can only delete a plan when there are no offers or subscriptions using it. If there are offers that use the plan, delete the plan, allow it to fail, and you will receive an error message. You can select **Parent offers** to display a list of offers that use the plan. For more information about deleting offers, see [Delete an offer](#).

Plans might have been added directly to a subscription as add-on plans, even if they are not part of the offer. In this case, they must be removed from the subscriptions that use them before the plan can be deleted.

Also, a plan cannot be removed from a subscription if it is the only source of a given resource for that subscription. For example, if Plan A has been added to Subscription 1, and it is the only plan providing a network quota to the subscription, it cannot be removed from the subscription. Therefore, it cannot be deleted.

## Edit and delete a quota

You can view and edit existing quotas using the administrator portal: select **Region Management**, then select the relevant resource provider, and click on **Quotas**. You can also delete quotas for certain resource providers.

The screenshot shows the Azure portal interface for 'Compute - Quotas' under the 'Compute' resource provider. The sidebar includes links for Overview, Alerts, and Quotas (which is highlighted with a red box). The main content area shows a table of 11 items with columns for Name, Plans, Virtual Machines, Cores, Availability Sets, Scale Sets, Standard Memory, Premium Memory, and three ellipsis buttons. One specific quota row, 'HighCoreCou...', is selected and highlighted with a dashed blue border. The 'Delete' button for this row is visible at the bottom right of the table.

Alternatively, you can delete some quotas using these REST APIs:

- [Compute](#)
- [Network](#)
- [Storage](#)

### NOTE

You cannot delete a quota if there are any current plans that use it. You must first delete the plan that references the quota.

## Next steps

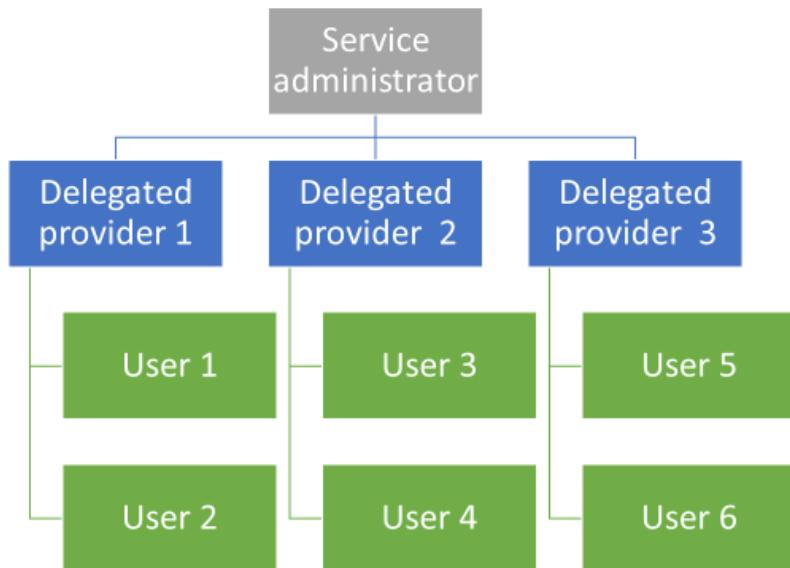
- [Create subscriptions](#)
- [Provision a virtual machine](#)

# Delegate offers in Azure Stack Hub

6 minutes to read • [Edit Online](#)

As an Azure Stack Hub operator, you might want to put other people in charge of signing up users and creating subscriptions. For example, if you're a service provider, you might want resellers to sign up customers and manage them on your behalf. Or, if you're part of a central IT group in an enterprise, you might want to delegate user sign-up to other IT staff.

Delegation makes it easier to reach and manage more users than you can by yourself, as shown in the following figure:



With delegation, the delegated provider manages an offer (called a *delegated offer*), and end customers obtain subscriptions under that offer without involvement from the system admin.

## Delegation roles

The following roles are part of delegation:

- The *Azure Stack Hub operator* manages the Azure Stack Hub infrastructure and creates an offer template. The operator delegates others to provide offers to their tenant.
- The delegated Azure Stack Hub operators are users with *Owner* or *Contributor* rights in the subscriptions called *delegated providers*. They can belong to other organizations, such as other Azure Active Directory (Azure AD) tenants.
- *Users* sign up for the offers and use them for managing their workloads, creating VMs, storing data, and so on.

## Delegation steps

There are two basic steps to setting up delegation:

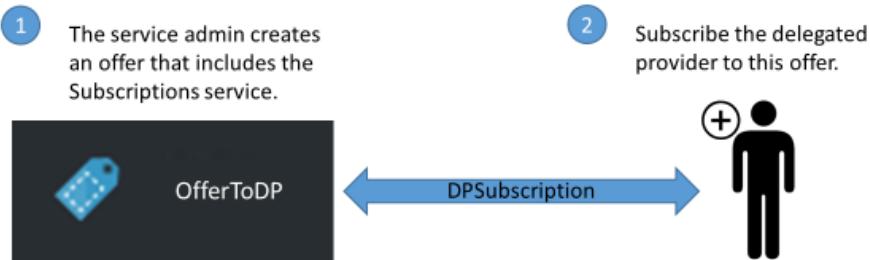
1. **Create a delegated provider subscription:** Subscribe a user to an offer containing only the subscriptions

service. Users who subscribe to this offer can then extend the delegated offers to other users by signing them up for those offers.

2. **Delegate an offer to the delegated provider:** This offer enables the delegated provider to create subscriptions or to extend the offer to their users. The delegated provider can now take the offer and offer it to other users.

The following figure shows the steps for setting up delegation:

### 1. Create the delegated provider



### 2. Create offers and enable Delegate Provider to sign up users



### Delegated provider requirements

To act as a delegated provider, a user establishes a relationship with the main provider by creating a subscription. This subscription identifies the delegated provider as having the right to present the delegated offers on behalf of the main provider.

After this relationship is established, the Azure Stack Hub operator can delegate an offer to the delegated provider. The delegated provider can take the offer, rename it (but not change its substance), and offer it to its customers.

## Delegation walkthrough

The following sections provide a walkthrough for setting up a delegated provider, delegating an offer, and verifying that users can sign up for the delegated offer.

### Set up roles

To use this walkthrough, you need two Azure AD accounts in addition to your Azure Stack Hub operator account. If you don't have these two accounts, you must create them. The accounts can belong to any Azure AD user and are referred to as the delegated provider and the user.

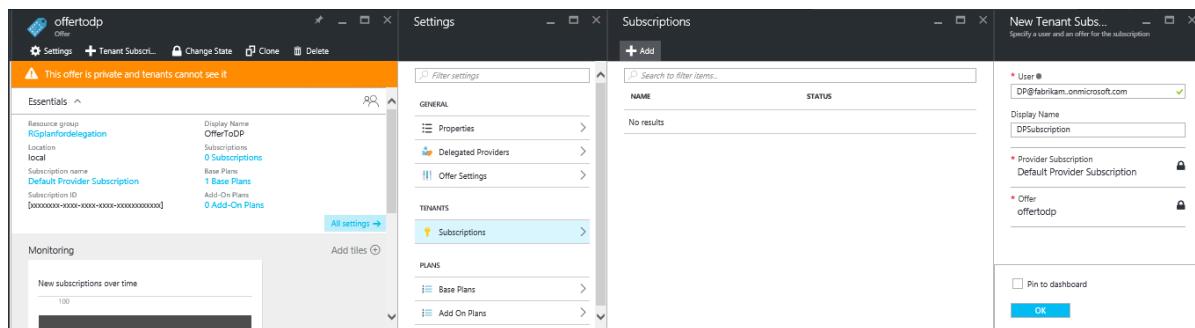
| ROLE               | ORGANIZATIONAL RIGHTS |
|--------------------|-----------------------|
| Delegated provider | User                  |
| User               | User                  |

## NOTE

In the case of a CSP-reseller, creating this delegated provider requires these users be in the tenant directory (the user Azure AD). The Azure Stack Hub operator must [first onboard](#) that tenant Azure AD, and then set up usage and billing by following [these steps](#).

## Identify the delegated provider

1. Sign in to the administrator portal as an Azure Stack Hub operator.
2. To create an offer that enables a user to become a delegated provider:
  - a. [Create a plan](#). This plan should include only the subscription service. This article uses a plan named **PlanForDelegation** as an example.
  - b. [Create an offer](#) based on this plan. This article uses an offer named **OfferToDP** as an example.
  - c. Add the delegated provider as a subscriber to this offer by selecting **Subscriptions**, then **Add**, then **New Tenant Subscription**.



## NOTE

As with all Azure Stack Hub offers, you have the option of making the offer public and letting users sign up for it, or keeping it private and letting the Azure Stack Hub operator manage the sign-up. Delegated providers are usually a small group. You want to control who is admitted to it, so keeping this offer private makes sense in most cases.

## Azure Stack Hub operator creates the delegated offer

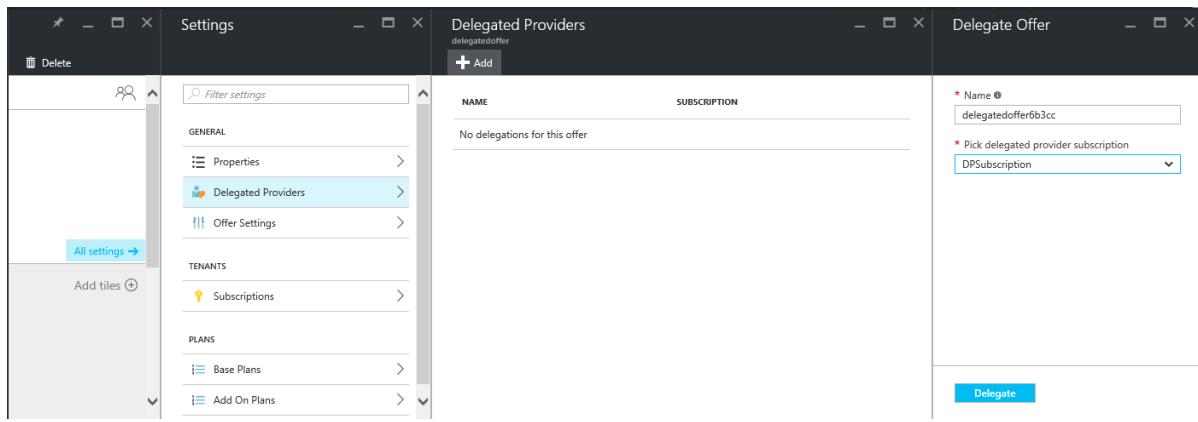
The next step is to create the plan and offer that you're going to delegate, and that your users will use. It's a good idea to define this offer as you want users to see it because the delegated provider can't change the plans and quotas it includes.

1. As an Azure Stack Hub operator, [create a plan](#) and [an offer](#) based on the plan. This article uses an offer named **DelegatedOffer** as an example.

## NOTE

This offer doesn't have to be public, but you can make it public. However, in most cases, you only want delegated providers to have access to the offer. After you delegate a private offer as described in the following steps, the delegated provider has access to it.

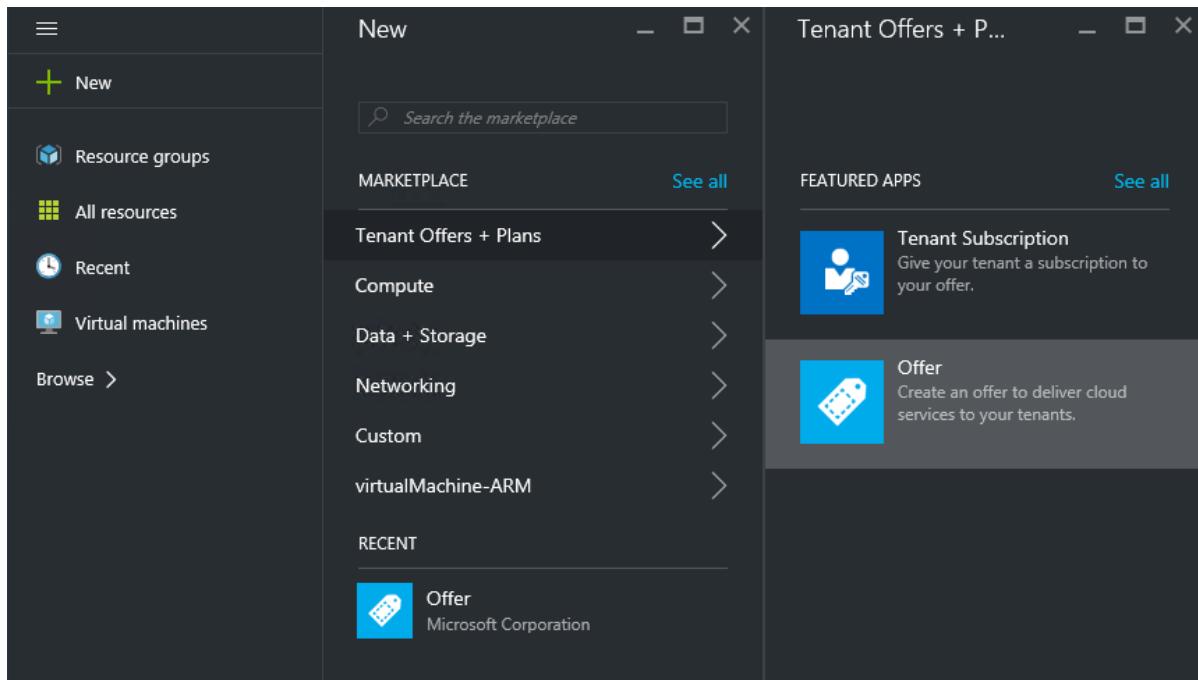
2. Delegate the offer. Go to **DelegatedOffer**. Under **Settings**, select **Delegated Providers**, then select **Add**.
3. Select the subscription for the delegated provider from the drop-down list, and then select **Delegate**.



## Delegated provider customizes the offer

Sign in to the user portal as the delegated provider and then create a new offer by using the delegated offer as a template.

1. Select **+ Create a resource**, then **Tenant Offers + Plans**, then select **Offer**.



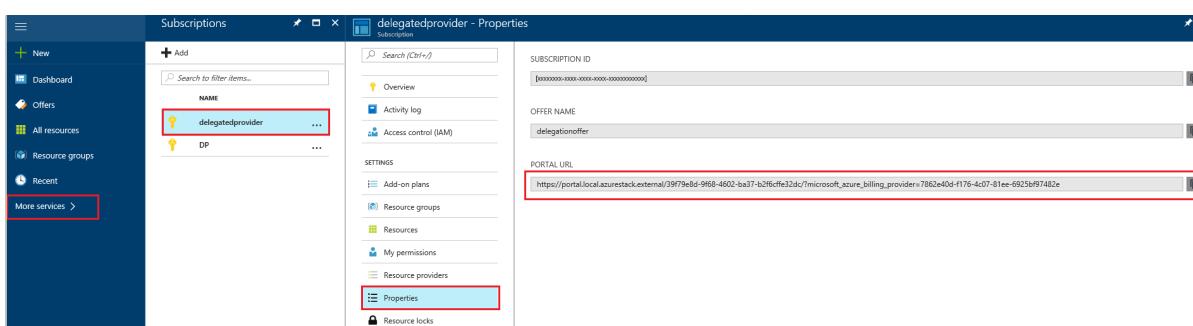
2. Assign a name to the offer. This example uses **ResellerOffer**. Select the delegated offer on which to base it, and then select **Create**.

The screenshot shows two adjacent browser windows. The left window is titled 'New Offer' and has a sub-header 'Create a new offer for your tenants'. It contains fields for 'Display Name' (ResellerOffer), 'Resource Name' (reselleroffer), 'Provider Subscription' (DPSSubscription), 'Resource Group' (RGreselleroffer), and 'Delegated offer' (None). The right window is titled 'Delegated Offer' and has a sub-header 'Select a delegated offer as a template'. It shows a list with one item: 'delegatedoffer6b3cc'.

#### IMPORTANT

It's important to understand that delegated providers can only choose offers that are delegated to them. They can't make changes to those offers. Only an Azure Stack Hub operator can change these offers. For example, only an operator can change their plans and quotas. A delegated provider doesn't construct an offer from base plans and add-on plans.

3. The delegated provider can make these offers public through their own portal URL. To make the offer public, select **Browse**, and then **Offers**. Select the offer, and then select **Change State**.
4. The public delegated offers are now visible only through the delegated portal. To find and change this URL:
  - a. Select **Browse**, then **All services**, and then under the **GENERAL** category, select **Subscriptions**. Select the delegated provider subscription (for example, **DPSSubscription**), then **Properties**.
  - b. Copy the portal URL to a separate location, such as Notepad.



You've finished creating a delegated offer as a delegated provider. Sign out as the delegated provider and close the browser window.

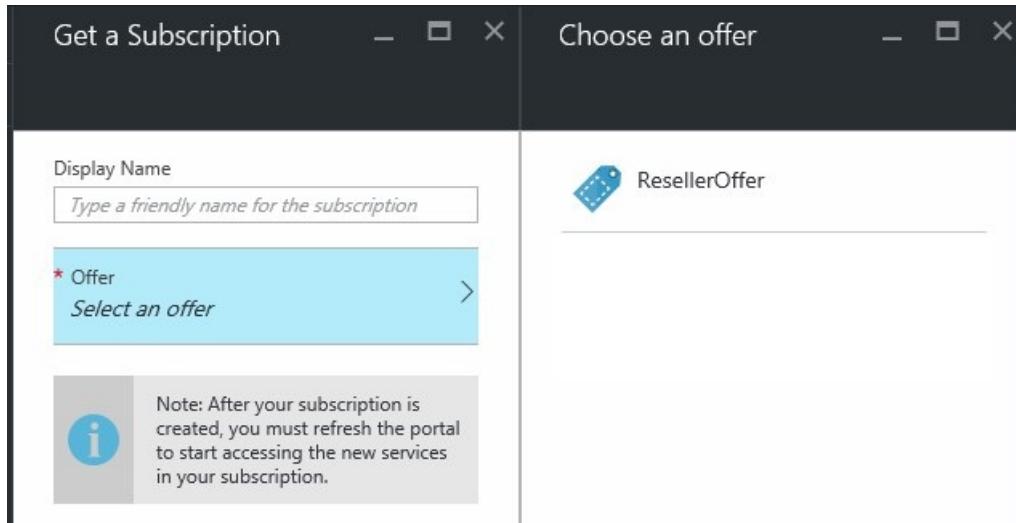
#### Sign up for the offer

1. In a new browser window, go to the delegated portal URL that you saved in the previous step. Sign in to the portal as a user.

#### NOTE

The delegated offers aren't visible unless you use the delegated portal.

2. In the dashboard, select **Get a subscription**. You'll see that only the delegated offers that were created by the delegated provider are presented to the user.



The process of delegating an offer is finished. Now a user can sign up for this offer by getting a subscription to it.

## Move subscriptions between delegated providers

If needed, a subscription can be moved between new or existing delegated provider subscriptions that belong to the same directory tenant. You can move them using the PowerShell cmdlet [Move-AzsSubscription](#).

Moving subscriptions is useful when:

- You onboard a new team member that will take on the delegated provider role and you want to assign to this team member user subscriptions that were previously created in the default provider subscription.
- You have multiple delegated providers subscriptions in the same directory tenant (Azure AD) and need to move user subscriptions between them. This scenario could occur when a team member moves between teams and their subscription must be allocated to the new team.

## Next steps

- [Provision a VM](#)

# Quota types in Azure Stack Hub

4 minutes to read • [Edit Online](#)

Quotas define the limits of resources that a user subscription can provision or consume. For example, a quota might allow a user to create up to five virtual machines (VMs). Each resource can have its own types of quotas.

## IMPORTANT

It can take up to two hours for new quotas to be available in the user portal or before a changed quota is enforced.

## Compute quota types

| Type                                              | Default Value | Description                                                                                                                                                                                                  |
|---------------------------------------------------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Maximum number of VMs                             | 50            | The maximum number of VMs that a subscription can create in this location.                                                                                                                                   |
| Maximum number of VM cores                        | 100           | The maximum number of cores that a subscription can create in this location (for example, an A3 VM has four cores).                                                                                          |
| Maximum number of availability sets               | 10            | The maximum number of availability sets that can be created in this location.                                                                                                                                |
| Maximum number of virtual machine scale sets      | 100           | The maximum number of scale sets that can be created in this location.                                                                                                                                       |
| Maximum capacity (in GB) of standard managed disk | 2048          | The maximum capacity of standard managed disks that can be created in this location. This value is a total of the allocation size of all standard managed disks and the used size of all standard snapshots. |
| Maximum capacity (in GB) of premium managed disk  | 2048          | The maximum capacity of premium managed disks that can be created in this location. This value is a total of the allocation size of all premium managed disks and the used size of all premium snapshots.    |

## NOTE

The maximum capacity of unmanaged disks (page blobs) is separate from the managed disk quota. You can set this value in **Maximum capacity (GB)** in **Storage quotas**.

## Storage quota types

| ITEM                             | DEFAULT VALUE | DESCRIPTION                                                                                                                                                                                                   |
|----------------------------------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Maximum capacity (GB)            | 2048          | Total storage capacity that can be consumed by a subscription in this location. This value is a total of the used size of all blobs (including unmanaged disks) and all associated snapshots, tables, queues. |
| Total number of storage accounts | 20            | The maximum number of storage accounts that a subscription can create in this location.                                                                                                                       |

#### NOTE

When **Maximum capacity (GB)** is exceeded in one subscription, you can't create new storage resource in this subscription. But you can continually using the unmanaged disks created in this subscription in VMs, which may cause total used capacity way beyond the quota limit.

The maximum capacity of managed disks is separate from the total storage quota. You can set this value in [Compute quotas](#).

## Network quota types

| ITEM                             | DEFAULT VALUE | DESCRIPTION                                                                                                                                                     |
|----------------------------------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Maximum virtual networks         | 50            | The maximum number of virtual networks that a subscription can create in this location.                                                                         |
| Maximum virtual network gateways | 1             | The maximum number of virtual network gateways (VPN gateways) that a subscription can create in this location.                                                  |
| Maximum network connections      | 2             | The maximum number of network connections (point-to-point or site-to-site) that a subscription can create across all virtual network gateways in this location. |
| Maximum public IPs               | 50            | The maximum number of public IP addresses that a subscription can create in this location.                                                                      |
| Maximum NICs                     | 100           | The maximum number of network interfaces that a subscription can create in this location.                                                                       |
| Maximum load balancers           | 50            | The maximum number of load balancers that a subscription can create in this location.                                                                           |
| Maximum network security groups  | 50            | The maximum number of network security groups that a subscription can create in this location.                                                                  |

# View an existing quota

There are two different ways to view an existing quota:

## Plans

1. In the left navigation pane of the administrator portal, select **Plans**.
2. Select the plan you want to view details for by clicking on its name.
3. In the blade that opens, select **Services and quotas**.
4. Select the quota you want to see by clicking it in the **Name** column.

The screenshot shows the 'Services and quotas' blade for the 'plan1' plan. The 'Services and quotas' link in the left sidebar is highlighted with a red box. In the main table, a quota named 'Small\_Storage\_quota' is selected, indicated by a red box around its name column.

| SERVICE           | NAME                | LOCATION |
|-------------------|---------------------|----------|
| Microsoft.Storage | Small_Storage_quota | local    |
| Microsoft.Network | Small_Network_quota | local    |
| Microsoft.Compute | Small_Compute_quota | local    |

## Resource providers

1. On the default dashboard of the administrator portal, find the **Resource providers** tile.
2. Select the service with the quota that you want to view, like **Compute**, **Network**, or **Storage**.
3. Select **Quotas**, and then select the quota you want to view.

# Edit a quota

There are two different ways to edit a quota:

## Edit a plan

1. In the left navigation pane of the administrator portal, select **Plans**.
2. Select the plan for which you want to edit a quota by clicking on its name.
3. In the blade that opens, select **Services and quotas**.
4. Select the quota you want to edit by clicking it in the **Name** column.

The screenshot shows the 'Services and quotas' blade for the 'plan1' plan. The 'Services and quotas' link in the left sidebar is highlighted with a red box. In the main table, a quota named 'Small\_Storage\_quota' is selected, indicated by a red box around its name column.

| SERVICE           | NAME                | LOCATION |
|-------------------|---------------------|----------|
| Microsoft.Storage | Small_Storage_quota | local    |
| Microsoft.Network | Small_Network_quota | local    |
| Microsoft.Compute | Small_Compute_quota | local    |

5. In the blade that opens, select **Edit in Compute**, **Edit in Network**, or **Edit in Storage**.

The screenshot shows a quota configuration page. At the top, there's a breadcrumb navigation: Home > Plans > plan1 - Services and quotas > Small\_Storage\_quota. Below the breadcrumb is a title 'Small\_Storage\_quota'. A red box highlights the 'Edit in Storage' button, which has a pencil icon. The page displays two quota details: Capacity (GB) is set to 500, and the Number of storage accounts is set to 10.

| Capacity (GB)              | 500 |
|----------------------------|-----|
| Number of storage accounts | 10  |

Alternatively, you can follow this procedure to edit a quota:

1. On the default dashboard of the administrator portal, find the **Resource providers** tile.
2. Select the service with the quota that you want to modify, like **Compute**, **Network**, or **Storage**.
3. Next, select **Quotas**, and then select the quota you want to change.
4. On the **Set Storage quotas**, **Set Compute quotas**, or **Set Network quotas** pane (depending on the type of quota you've chosen to edit), edit the values, and then select **Save**.

#### Edit original configuration

You can choose to edit the original configuration of a quota instead of [using an add-on plan](#). When you edit a quota, the new configuration automatically applies globally to all plans that use that quota and all existing subscriptions that use those plans. The editing of a quota is different than when you use an add-on plan to provide a modified quota, which a user chooses to subscribe to.

The new values for the quota apply globally to all plans that use the modified quota and to all existing subscriptions that use those plans.

## Next steps

- [Learn more about services, plans, offers, and quotas.](#)
- [Create quotas while creating a plan.](#)

# Offer a network solution in Azure Stack Hub with Fortinet FortiGate

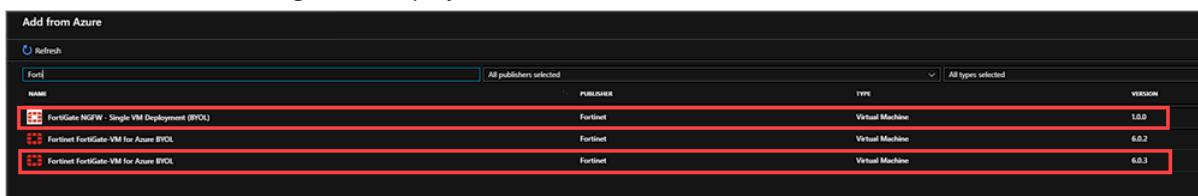
2 minutes to read • [Edit Online](#)

You can add FortiGate Next-Generation Firewall to your Azure Stack Hub Marketplace. FortiGate enables your users to create network solutions such as virtual private network (VPN) to Azure Stack Hub and VNET peering. A network virtual appliance (NVA) controls the flow of network traffic from a perimeter network to other networks or subnets.

For more information about FortiGate in the Azure Marketplace, see [Fortinet FortiGate Next-Generation Firewall Single VM Solution](#).

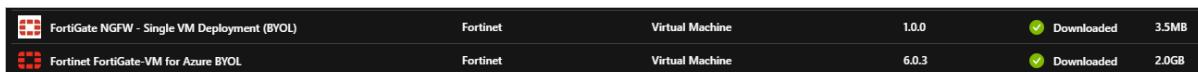
## Download the Required Azure Stack Hub Marketplace items

1. Open the Azure Stack Hub administrator portal.
2. Select **Marketplace management** and select **Add from Azure**.
3. Type **Forti** in the search box, and double-click > select **Download** to get the latest available versions of the following items:
  - Fortinet FortiGate-VM For Azure BYOL
  - FortiGate NGFW - Single VM Deployment (BYOL)



| NAME                                         | PUBLISHER | TYPE            | VERSION |
|----------------------------------------------|-----------|-----------------|---------|
| FortiGate NGFW - Single VM Deployment (BYOL) | Fortinet  | Virtual Machine | 1.0.0   |
| Fortinet FortiGate-VM for Azure BYOL         | Fortinet  | Virtual Machine | 6.0.2   |
| Fortinet FortiGate-VM for Azure BYOL         | Fortinet  | Virtual Machine | 6.0.3   |

4. Wait until your Marketplace items have a status of **Downloaded**. The items may take several minutes to download.



|                                              |          |                 |       |            |       |
|----------------------------------------------|----------|-----------------|-------|------------|-------|
| FortiGate NGFW - Single VM Deployment (BYOL) | Fortinet | Virtual Machine | 1.0.0 | Downloaded | 3.5MB |
| Fortinet FortiGate-VM for Azure BYOL         | Fortinet | Virtual Machine | 6.0.3 | Downloaded | 2.0GB |

## Next steps

[Setup VPN for Azure Stack Hub using FortiGate NVA](#)

[How to connect two VNETs through peering](#)

[How to establish a VNET to VNET connection with Fortinet FortiGate NVA](#)

# Azure App Service and Azure Functions on Azure Stack Hub overview

3 minutes to read • [Edit Online](#)

Azure App Service on Azure Stack Hub is a platform-as-a-service (PaaS) offering from Microsoft Azure available on Azure Stack Hub. The service enables your internal or external customers to create web, API, and Azure Functions apps for any platform or device. They can integrate your apps with on-premises apps and automate their business processes. Azure Stack Hub cloud operators can run customer apps on fully managed virtual machines (VMs) with their choice of shared VM resources or dedicated VMs.

Azure App Service enables you to automate business processes and host cloud APIs. As a single integrated service, Azure App Service lets you combine various components (like websites, REST APIs, and business processes) into a single solution.

## Why offer Azure App Service on Azure Stack Hub?

Here are some key features and capabilities of Azure App Service:

- **Multiple languages and frameworks:** Azure App Service has first-class support for ASP.NET, Node.js, Java, PHP, and Python. You can also run Windows PowerShell and other scripts or executables on App Service VMs.
- **DevOps optimization:** Set up continuous integration and deployment with GitHub, local Git, or BitBucket. Promote updates through test and staging environments, and manage your apps in App Service by using Azure PowerShell or the cross-platform command-line interface (CLI).
- **Visual Studio integration:** Dedicated tools in Visual Studio streamline the work of creating and deploying apps.

## App types in App Service

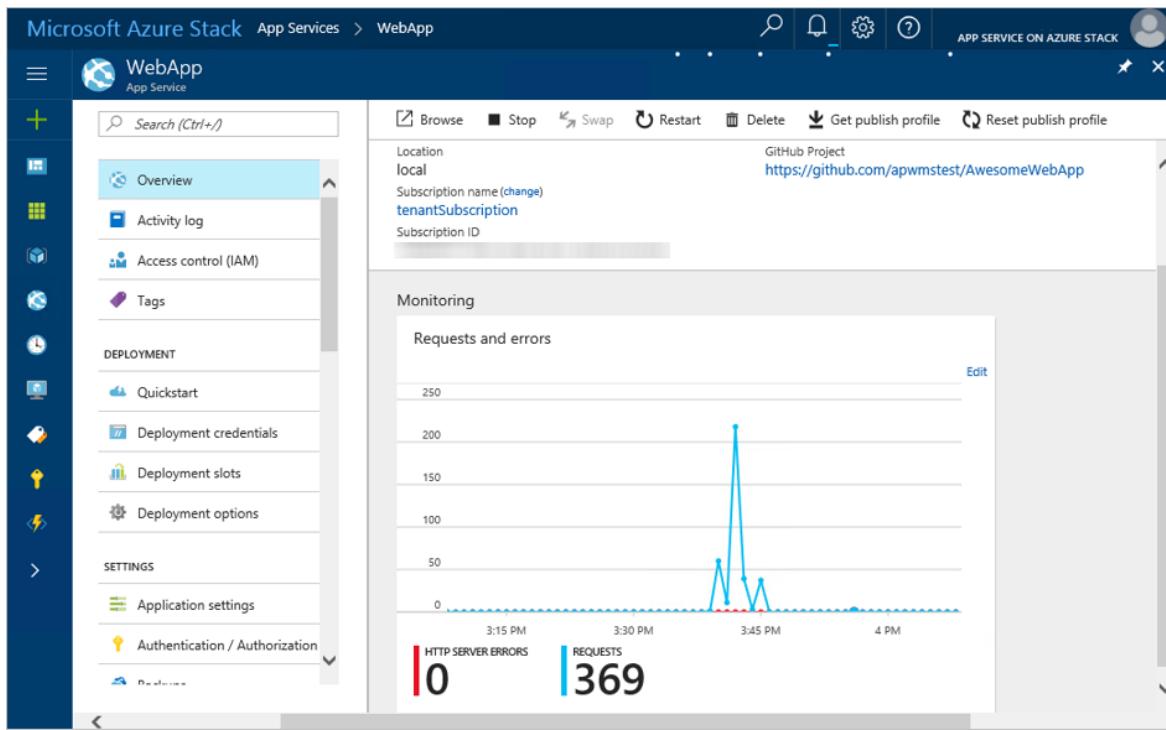
App Service offers several app types, each of which is intended to host a specific workload:

- [Web Apps](#) for hosting websites and web apps.
- [API Apps](#) for hosting REST APIs.
- Azure Functions for hosting event driven, serverless workloads.

The word *app* refers to the hosting resources dedicated to running a workload. Taking *web app* as an example, you're probably accustomed to thinking of a web app as both the compute resources and app code that together deliver functionality to a browser. In Azure App Service, a web app is the compute resource that Azure Stack Hub provides for hosting your app code.

Your app can be composed of multiple App Service apps of different kinds. For example, if your app is composed of a web front end and a REST API back end, you can:

- Deploy both (front end and API) to a single web app.
- Deploy your front-end code to a web app and your back-end code to an API app.



## What is an App Service plan?

The App Service resource provider uses the same code that Azure App Service uses, and thus shares some common concepts. In App Service, the pricing container for apps is called the *App Service plan*. It represents the set of dedicated VMs used to hold your apps. Within a given subscription, you can have multiple App Service plans.

In Azure, there are shared and dedicated workers. A shared worker supports high-density and multi-tenant app hosting, and there's only one set of shared workers. Dedicated servers are used by only one tenant and come in three sizes: small, medium, and large. The needs of on-premises customers can't always be described by using those terms. In App Service on Azure Stack Hub, resource provider admins define the worker tiers they want to make available. Based on your unique hosting needs, you can define multiple sets of shared workers or different sets of dedicated workers. By using those worker-tier definitions, they can then define their own pricing SKUs.

## Portal features

Azure App Service on Azure Stack Hub uses the same user interface that Azure App Service uses. The same is true with the back end. However, some features are disabled in Azure Stack Hub. The Azure-specific expectations or services that those features require aren't currently available in Azure Stack Hub.

## Next steps

- [Prerequisites for deploying App Service on Azure Stack Hub](#)
- [Install the Azure App Service resource provider](#)

You can also try out other [platform as a service \(PaaS\) services](#), such as the [SQL Server resource provider](#) and the [MySQL resource provider](#).

# Capacity planning for App Service server roles in Azure Stack Hub

5 minutes to read • [Edit Online](#)

To set up a production-ready deployment of Azure App Service on Azure Stack Hub, you must plan for the capacity you expect the system to support.

This article provides guidance for the minimum number of compute instances and compute SKUs you should use for any production deployment.

You can plan your App Service capacity strategy using these guidelines.

| APP SERVICE SERVER ROLE | MINIMUM RECOMMENDED NUMBER OF INSTANCES | RECOMMENDED COMPUTE SKU |
|-------------------------|-----------------------------------------|-------------------------|
| Controller              | 2                                       | A1                      |
| Front End               | 2                                       | A1                      |
| Management              | 2                                       | A3                      |
| Publisher               | 2                                       | A1                      |
| Web Workers - shared    | 2                                       | A1                      |
| Web Workers - dedicated | 2 per tier                              | A1                      |

## Controller role

**Recommended minimum:** Two instances of A1 Standard

The Azure App Service controller typically experiences low consumption of CPU, memory, and network resources. However, for high availability, you must have two controllers. Two controllers are also the maximum number of controllers permitted. You can create the second web sites controller direct from the installer during deployment.

## Front-end role

**Recommended minimum:** Two instances of A1 Standard

The front-end routes requests to web workers depending on web worker availability. For high availability, you should have more than one front end, and you can have more than two. For capacity planning purposes, consider that each core can handle approximately 100 requests per second.

## Management role

**Recommended minimum:** Two instances of A3 Standard

The Azure App classic deployment model role is responsible for the App Service Azure Resource Manager and API endpoints, portal extensions (admin, tenant, Functions portal), and the data service. The management server role typically requires only about 4-GB RAM in a production environment. However, it may experience high CPU levels when many management tasks (such as web site creation) are performed. For high availability, you should

have more than one server assigned to this role, and at least two cores per server.

## Publisher role

**Recommended minimum:** Two instances of A1 Standard

If many users are publishing simultaneously, the publisher role may experience heavy CPU usage. For high availability, make sure more than one publisher role is available. The publisher only handles FTP/FTPS traffic.

## Web worker role

**Recommended minimum:** Two instances of A1 Standard

For high availability, you should have at least four web worker roles: two for shared web site mode and two for each dedicated worker tier you plan to offer. The shared and dedicated compute modes provide different levels of service to tenants. You might need more web workers if many of your customers are:

- Using dedicated compute mode worker tiers (which are resource-intensive).
- Running in shared compute mode.

After a user has created an App Service plan for a dedicated compute mode SKU, the number of web worker(s) specified in that App Service plan is no longer available to users.

To provide Azure Functions to users in the consumption plan model, you must deploy shared web workers.

When deciding on the number of shared web worker roles to use, review these considerations:

- **Memory:** Memory is the most critical resource for a web worker role. Insufficient memory impacts web site performance when virtual memory is swapped from disk. Each server requires about 1.2 GB of RAM for the operating system. RAM above this threshold can be used to run web sites.
- **Percentage of active web sites:** Typically, about 5 percent of apps in an Azure App Service on Azure Stack Hub deployment are active. However, the percentage of apps that are active at any given moment can be higher or lower. With an active app rate of 5 percent, the maximum number of apps to place in an Azure App Service on Azure Stack Hub deployment should be less than 20 times the number of active web sites ( $5 \times 20 = 100$ ).
- **Average memory footprint:** The average memory footprint for apps observed in production environments is about 70 MB. Using this footprint, the memory allocated across all web worker role computers or VMs is calculated as follows:

`Number of provisioned applications * 70 MB * 5% - (number of web worker roles * 1044 MB)`

For example, if there are 5,000 apps on an environment running 10 web worker roles, each web worker role VM should have 7060-MB RAM:

`5,000 * 70 * 0.05 - (10 * 1044) = 7060 (= about 7 GB)`

For info on adding more worker instances, see [Adding more worker roles](#).

### Additional considerations for dedicated workers during upgrade and maintenance

During upgrade and maintenance of workers, Azure App Service on Azure Stack Hub will perform maintenance on 20% of each worker tier at any one time. Therefore, cloud admins must always maintain a 20% pool of unallocated workers per worker tier to ensure their tenants don't experience any loss of service during upgrade and maintenance. For example, if you have 10 workers in a worker tier you should ensure that 2 are unallocated to allow upgrade and maintenance. If the full 10 workers become allocated, you should scale the worker tier up to maintain a pool of unallocated workers.

During upgrade and maintenance, Azure App Service will move workloads to unallocated workers to ensure the

workloads will continue to operate. However, if there are no unallocated workers available during upgrade then there's potential for tenant workload downtime. With regards to shared workers, customers don't need to provision additional workers as the service will allocate tenant apps within available workers automatically. For high availability, there's a minimum requirement of two workers in this tier.

Cloud admins can monitor their worker tier allocation in the App Service admin area in the Azure Stack Hub administrator portal. Navigate to App Service and then select Worker Tiers in the left-hand pane. The Worker Tiers table shows worker tier name, size, image used, number of available workers (unallocated), total number of workers in each tier and the overall state of the worker tier.

The screenshot shows the 'App Service - Worker Tiers' page in the Azure Stack Hub administrator portal. On the left, a navigation sidebar lists various options like Overview, Properties, System configuration, Secrets, Source control configuration, Roles, IP SSL, IP Block List, SKUs, Pricing Tiers, Subscription Quotas, and Custom Software. The 'Worker Tiers' option is highlighted with a yellow box. The main content area displays two tables: 'Shared Worker Tiers' and 'Dedicated Worker Tiers'. The 'Shared Worker Tiers' table has one row for 'Shared' with a Standard\_A1 size, 2016-Datacenter image, 10 available, 10 total workers, and a 'Ready' state. The 'Dedicated Worker Tiers' table has three rows: 'Small' (Standard\_A1, 2016-Datacenter, 9 available, 10 total, Ready), 'Medium' (Standard\_A2, 2016-Datacenter, 5 available, 5 total, Ready), and 'Large' (Standard\_A3, 2016-Datacenter, 5 available, 5 total, Ready). A yellow box highlights the 'Available' and 'Total' columns in both tables.

| NAME   | SIZE        | IMAGE           | AVAILABLE | TOTAL | STATE |
|--------|-------------|-----------------|-----------|-------|-------|
| Shared | Standard_A1 | 2016-Datacenter | 10        | 10    | Ready |

| NAME   | SIZE        | IMAGE           | AVAILABLE | TOTAL | STATE |
|--------|-------------|-----------------|-----------|-------|-------|
| Small  | Standard_A1 | 2016-Datacenter | 9         | 10    | Ready |
| Medium | Standard_A2 | 2016-Datacenter | 5         | 5     | Ready |
| Large  | Standard_A3 | 2016-Datacenter | 5         | 5     | Ready |

## File server role

For the file server role, you can use a standalone file server for development and testing. For example, when deploying Azure App Service on the Azure Stack Development Kit (ASDK) you can use this [template](#). For production purposes, you should use a pre-configured Windows file server, or a pre-configured non-Windows file server.

In production environments, the file server role experiences intensive disk I/O. Because it houses all of the content and app files for user web sites, you should preconfigure one of the following resources for this role:

- Windows file server
- Windows file server cluster
- Non-Windows file server
- Non-Windows file server cluster
- NAS (Network Attached Storage) device

For more information, see [Provision a file server](#).

## Next steps

[Prerequisites for deploying App Service on Azure Stack Hub](#)

# Prerequisites for deploying App Service on Azure Stack Hub

13 minutes to read • [Edit Online](#)

Before you deploy Azure App Service on Azure Stack Hub, you must complete the prerequisite steps in this article.

## IMPORTANT

Apply the 1910 update to your Azure Stack Hub integrated system or deploy the latest Azure Stack Development Kit (ASDK) before you deploy Azure App Service 1.8.

## Download the installer and helper scripts

1. Download the [App Service on Azure Stack Hub deployment helper scripts](#).
2. Download the [App Service on Azure Stack Hub installer](#).
3. Extract the files from the helper scripts .zip file. The following files and folders are extracted:
  - Common.ps1
  - Create-AADIdentityApp.ps1
  - Create-ADFSIdentityApp.ps1
  - Create-AppServiceCerts.ps1
  - Get-AzureStackRootCert.ps1
  - Remove-AppService.ps1
  - Modules folder
    - GraphAPI.psm1

## Download items from the Azure Marketplace

Azure App Service on Azure Stack Hub requires items to be [downloaded from the Azure Marketplace](#), making them available in the Azure Stack Hub Marketplace. These items must be downloaded before you start the deployment or upgrade of Azure App Service on Azure Stack Hub:

1. The latest version of Windows Server 2016 Datacenter virtual machine image.
2. Custom Script Extension v1.9.1 or greater. This is a virtual machine extension.

## Get certificates

### Azure Resource Manager root certificate for Azure Stack Hub

Open an elevated PowerShell session on a computer that can reach the privileged endpoint on the Azure Stack Hub Integrated System or ASDK Host.

Run the `Get-AzureStackRootCert.ps1` script from the folder where you extracted the helper scripts. The script creates a root certificate in the same folder as the script that App Service needs for creating certificates.

When you run the following PowerShell command, you have to provide the privileged endpoint and the credentials for the `AzureStack\CloudAdmin`.

```
Get-AzureStackRootCert.ps1
```

#### Get-AzureStackRootCert.ps1 script parameters

| PARAMETER            | REQUIRED OR OPTIONAL | DEFAULT VALUE         | DESCRIPTION                                                |
|----------------------|----------------------|-----------------------|------------------------------------------------------------|
| PrivilegedEndpoint   | Required             | AzS-ERCS01            | Privileged endpoint                                        |
| CloudAdminCredential | Required             | AzureStack\CloudAdmin | Domain account credential for Azure Stack Hub cloud admins |

#### Certificates required for ASDK deployment of Azure App Service

The *Create-AppServiceCerts.ps1* script works with the Azure Stack Hub certificate authority to create the four certificates that App Service needs.

| FILE NAME                                    | USE                                          |
|----------------------------------------------|----------------------------------------------|
| _appservice.local.azurestack.external.pfx    | App Service default SSL certificate          |
| api.appservice.local.azurestack.external.pfx | App Service API SSL certificate              |
| ftp.appservice.local.azurestack.external.pfx | App Service publisher SSL certificate        |
| sso.appservice.local.azurestack.external.pfx | App Service identity application certificate |

To create the certificates, follow these steps:

1. Sign in to the ASDK host using the AzureStack\AzureStackAdmin account.
2. Open an elevated PowerShell session.
3. Run the *Create-AppServiceCerts.ps1* script from the folder where you extracted the helper scripts. This script creates four certificates in the same folder as the script that App Service needs for creating certificates.
4. Enter a password to secure the .pfx files, and make a note of it. You have to enter it in the App Service on Azure Stack Hub installer.

#### Create-AppServiceCerts.ps1 script parameters

| PARAMETER   | REQUIRED OR OPTIONAL | DEFAULT VALUE             | DESCRIPTION                                             |
|-------------|----------------------|---------------------------|---------------------------------------------------------|
| pfxPassword | Required             | Null                      | Password that helps protect the certificate private key |
| DomainName  | Required             | local.azurestack.external | Azure Stack Hub region and domain suffix                |

#### Certificates required for Azure Stack Hub production deployment of Azure App Service

To run the resource provider in production, you must provide the following certificates:

- Default domain certificate
- API certificate
- Publishing certificate
- Identity certificate

#### **Default domain certificate**

The default domain certificate is placed on the front-end role. User apps for wildcard or default domain request to Azure App Service use this certificate. The certificate is also used for source control operations (Kudu).

The certificate must be in .pfx format and should be a three-subject wildcard certificate. This requirement allows one certificate to cover both the default domain and the SCM endpoint for source control operations.

| FORMAT                                             | EXAMPLE                                      |
|----------------------------------------------------|----------------------------------------------|
| *.appservice.<region>.<DomainName>.<extension>     | *.appservice.redmond.azurestack.external     |
| *.scm.appservice.<region>.<DomainName>.<extension> | *.scm.appservice.redmond.azurestack.external |
| *.sso.appservice.<region>.<DomainName>.<extension> | *.sso.appservice.redmond.azurestack.external |

#### **API certificate**

The API certificate is placed on the Management role. The resource provider uses it to help secure API calls. The certificate for publishing must contain a subject that matches the API DNS entry.

| FORMAT                                           | EXAMPLE                                    |
|--------------------------------------------------|--------------------------------------------|
| api.appservice.<region>.<DomainName>.<extension> | api.appservice.redmond.azurestack.external |

#### **Publishing certificate**

The certificate for the Publisher role secures the FTPS traffic for app owners when they upload content. The certificate for publishing must contain a subject that matches the FTPS DNS entry.

| FORMAT                                           | EXAMPLE                                    |
|--------------------------------------------------|--------------------------------------------|
| ftp.appservice.<region>.<DomainName>.<extension> | ftp.appservice.redmond.azurestack.external |

#### **Identity certificate**

The certificate for the identity app enables:

- Integration between the Azure Active Directory (Azure AD) or Active Directory Federation Services (AD FS) directory, Azure Stack Hub, and App Service to support integration with the compute resource provider.
- Single sign-on scenarios for advanced developer tools within Azure App Service on Azure Stack Hub.

The certificate for identity must contain a subject that matches the following format.

| FORMAT                                           | EXAMPLE                                    |
|--------------------------------------------------|--------------------------------------------|
| sso.appservice.<region>.<DomainName>.<extension> | sso.appservice.redmond.azurestack.external |

#### **Validate certificates**

Before deploying the App Service resource provider, you should [validate the certificates to be used](#) by using the Azure Stack Hub Readiness Checker tool available from the [PowerShell Gallery](#). The Azure Stack Hub Readiness Checker Tool validates that the generated PKI certificates are suitable for App Service deployment.

As a best practice, when working with any of the necessary [Azure Stack Hub PKI certificates](#), you should plan enough time to test and reissue certificates if necessary.

## Virtual network

#### **NOTE**

The precreation of a custom virtual network is optional as the Azure App Service on Azure Stack Hub can create the required virtual network but will then need to communicate with SQL and File Server via public IP addresses.

Azure App Service on Azure Stack Hub lets you deploy the resource provider to an existing virtual network or lets you create a virtual network as part of the deployment. Using an existing virtual network enables the use of internal IPs to connect to the file server and SQL Server required by Azure App Service on Azure Stack Hub. The virtual network must be configured with the following address range and subnets before installing Azure App Service on Azure Stack Hub:

Virtual network - /16

Subnets

- ControllersSubnet /24
- ManagementServersSubnet /24
- FrontEndsSubnet /24
- PublishersSubnet /24
- WorkersSubnet /21

## Licensing concerns for required file server and SQL Server

Azure App Service on Azure Stack Hub requires a file server and SQL Server to operate. You're free to use pre-existing resources located outside of your Azure Stack Hub deployment or deploy resources within their Azure Stack Hub Default Provider Subscription.

If you choose to deploy the resources within your Azure Stack Hub Default Provider Subscription, the licenses for those resources (Windows Server Licenses and SQL Server Licenses) are included in the cost of Azure App Service on Azure Stack Hub subject to the following constraints:

- the infrastructure is deployed into the **Default Provider Subscription**;
- the infrastructure is exclusively used by the Azure App Service on Azure Stack Hub resource provider. No other workloads, administrative (other resource providers, for example: SQL-RP) or tenant (for example: tenant apps, which require a database), are permitted to make use of this infrastructure.

## Prepare the file server

Azure App Service requires the use of a file server. For production deployments, the file server must be configured to be highly available and capable of handling failures.

### **Quickstart template for file server for deployments of Azure App Service on ASDK.**

For ASDK deployments only, you can use the [example Azure Resource Manager deployment template](#) to deploy a configured single-node file server. The single-node file server will be in a workgroup.

### **Quickstart template for Highly Available file server and SQL Server**

A [reference architecture quickstart template](#) is now available which will deploy a file server and SQL Server. This template supports Active Directory infrastructure in a virtual network configured to support a highly available deployment of Azure App Service on Azure Stack Hub.

### **Steps to deploy a custom file server**

## IMPORTANT

If you choose to deploy App Service in an existing virtual network, the file server should be deployed into a separate Subnet from App Service.

## NOTE

If you have chosen to deploy a file server using either of the Quickstart templates mentioned above, you can skip this section as the file servers are configured as part of the template deployment.

### Provision groups and accounts in Active Directory

1. Create the following Active Directory global security groups:

- FileShareOwners
- FileShareUsers

2. Create the following Active Directory accounts as service accounts:

- FileShareOwner
- FileShareUser

As a security best practice, the users for these accounts (and for all web roles) should be unique and have strong usernames and passwords. Set the passwords with the following conditions:

- Enable **Password never expires**.
- Enable **User cannot change password**.
- Disable **User must change password at next logon**.

3. Add the accounts to the group memberships as follows:

- Add **FileShareOwner** to the **FileShareOwners** group.
- Add **FileShareUser** to the **FileShareUsers** group.

### Provision groups and accounts in a workgroup

## NOTE

When you're configuring a file server, run all the following commands from an **Administrator Command Prompt**.

***Don't use PowerShell.***

When you use the Azure Resource Manager template, the users are already created.

1. Run the following commands to create the FileShareOwner and FileShareUser accounts. Replace <password> with your own values.

```
net user FileShareOwner <password> /add /expires:never /passwordchg:no
net user FileShareUser <password> /add /expires:never /passwordchg:no
```

2. Set the passwords for the accounts to never expire by running the following WMIC commands:

```
WMIC USERACCOUNT WHERE "Name='FileShareOwner'" SET PasswordExpires=FALSE
WMIC USERACCOUNT WHERE "Name='FileShareUser'" SET PasswordExpires=FALSE
```

3. Create the local groups FileShareUsers and FileShareOwners, and add the accounts in the first step to them:

```
net localgroup FileShareUsers /add
net localgroup FileShareUsers FileShareUser /add
net localgroup FileShareOwners /add
net localgroup FileShareOwners FileShareOwner /add
```

### Provision the content share

The content share contains tenant website content. The procedure to provision the content share on a single file server is the same for both Active Directory and workgroup environments. But it's different for a failover cluster in Active Directory.

#### Provision the content share on a single file server (Active Directory or workgroup)

On a single file server, run the following commands at an elevated command prompt. Replace the value for `C:\WebSites` with the corresponding paths in your environment.

```
set WEBSITES_SHARE=WebSites
set WEBSITES_FOLDER=C:\WebSites
md %WEBSITES_FOLDER%
net share %WEBSITES_SHARE% /delete
net share %WEBSITES_SHARE%=%WEBSITES_FOLDER% /grant:Everyone,full
```

### Configure access control to the shares

Run the following commands at an elevated command prompt on the file server or on the failover cluster node, which is the current cluster resource owner. Replace values in *italicics* with values that are specific to your environment.

#### Active Directory

```
set DOMAIN=<DOMAIN>
set WEBSITES_FOLDER=C:\WebSites
icacls %WEBSITES_FOLDER% /reset
icacls %WEBSITES_FOLDER% /grant Administrators:(OI)(CI)(F)
icacls %WEBSITES_FOLDER% /grant %DOMAIN%\FileShareOwners:(OI)(CI)(M)
icacls %WEBSITES_FOLDER% /inheritance:r
icacls %WEBSITES_FOLDER% /grant %DOMAIN%\FileShareUsers:(CI)(S,X,RA)
icacls %WEBSITES_FOLDER% /grant *S-1-1-0:(OI)(CI)(IO)(RA,REA,RD)
```

#### Workgroup

```
set WEBSITES_FOLDER=C:\WebSites
icacls %WEBSITES_FOLDER% /reset
icacls %WEBSITES_FOLDER% /grant Administrators:(OI)(CI)(F)
icacls %WEBSITES_FOLDER% /grant FileShareOwners:(OI)(CI)(M)
icacls %WEBSITES_FOLDER% /inheritance:r
icacls %WEBSITES_FOLDER% /grant FileShareUsers:(CI)(S,X,RA)
icacls %WEBSITES_FOLDER% /grant *S-1-1-0:(OI)(CI)(IO)(RA,REA,RD)
```

## Prepare the SQL Server instance

#### NOTE

If you've chosen to deploy the Quickstart template for Highly Available File Server and SQL Server, you can skip this section as the template deploys and configures SQL Server in a HA configuration.

For the Azure App Service on Azure Stack Hub hosting and metering databases, you must prepare a SQL Server instance to hold the App Service databases.

For ASDK deployments, you can use SQL Server Express 2014 SP2 or later. SQL Server must be configured to support **Mixed Mode** authentication because App Service on Azure Stack Hub **DOES NOT** support Windows Authentication.

For production and high-availability purposes, you should use a full version of SQL Server 2014 SP2 or later, enable mixed-mode authentication, and deploy in a [highly available configuration](#).

The SQL Server instance for Azure App Service on Azure Stack Hub must be accessible from all App Service roles. You can deploy SQL Server within the Default Provider Subscription in Azure Stack Hub. Or you can make use of the existing infrastructure within your organization (as long as there's connectivity to Azure Stack Hub). If you're using an Azure Marketplace image, remember to configure the firewall accordingly.

#### NOTE

A number of SQL IaaS virtual machine images are available through the Marketplace Management feature. Make sure you always download the latest version of the SQL IaaS Extension before you deploy a VM using a Marketplace item.

The SQL images are the same as the SQL VMs that are available in Azure. For SQL VMs created from these images, the IaaS extension and corresponding portal enhancements provide features such as automatic patching and backup capabilities.

For any of the SQL Server roles, you can use a default instance or a named instance. If you use a named instance, be sure to manually start the SQL Server Browser service and open port 1434.

The App Service installer will check to ensure the SQL Server has database containment enabled. To enable database containment on the SQL Server that will host the App Service databases, run these SQL commands:

```
sp_configure 'contained database authentication', 1;
GO
RECONFIGURE;
GO
```

#### IMPORTANT

If you choose to deploy App Service in an existing virtual network the SQL Server should be deployed into a separate Subnet from App Service and the File Server.

## Create an Azure Active Directory app

Configure an Azure AD service principal, to support the following operations:

- Virtual machine scale set integration on worker tiers.
- SSO for the Azure Functions portal and advanced developer tools.

These steps apply to Azure AD-secured Azure Stack Hub environments only.

Admins must configure SSO to:

- Enable the advanced developer tools within App Service (Kudu).
- Enable the use of the Azure Functions portal experience.

Follow these steps to create the service principal in your Azure AD tenant:

1. Open a PowerShell instance as azurestack\AzureStackAdmin.
2. Go to the location of the scripts that you downloaded and extracted in the [prerequisite step](#).
3. [Install PowerShell for Azure Stack Hub](#).

4. Run the **Create-AADIdentityApp.ps1** script. When you're prompted, enter the Azure AD tenant ID that you're using for your Azure Stack Hub deployment. For example, enter **myazurestack.onmicrosoft.com**.
5. In the **Credential** window, enter your Azure AD service admin account and password. Select **OK**.
6. Enter the certificate file path and certificate password for the [certificate created earlier](#). The certificate created for this step by default is **sso.appservice.local.azurestack.external.pfx**.
7. Make note of the application ID that's returned in the PowerShell output. You use the ID in the following steps to provide consent for the application's permissions, and during installation.
8. Open a new browser window, and sign in to the [Azure portal](#) as the Azure Active Directory service admin.
9. Open the Azure Active Directory service.
10. Select **App Registrations** in the left pane.
11. Search for the application ID you noted in step 7.
12. Select the App Service application registration from the list.
13. Select **API permissions** in the left pane.
14. Select **Grant admin consent for <tenant>**, where <tenant> is the name of your Azure AD tenant. Confirm the consent grant by selecting **Yes**.

Create-AADIdentityApp.ps1

| PARAMETER                 | REQUIRED OR OPTIONAL | DEFAULT VALUE | DESCRIPTION                                                                                        |
|---------------------------|----------------------|---------------|----------------------------------------------------------------------------------------------------|
| DirectoryTenantName       | Required             | Null          | Azure AD tenant ID. Provide the GUID or string. An example is myazureaaddirectory.onmicrosoft.com. |
| AdminArmEndpoint          | Required             | Null          | Admin Azure Resource Manager endpoint. An example is adminmanagement.local.azurestack.external.    |
| TenantARMEndpoint         | Required             | Null          | Tenant Azure Resource Manager endpoint. An example is management.local.azurestack.external.        |
| AzureStackAdminCredential | Required             | Null          | Azure AD service admin credential.                                                                 |
| CertificateFilePath       | Required             | Null          | <b>Full path</b> to the identity application certificate file generated earlier.                   |
| CertificatePassword       | Required             | Null          | Password that helps protect the certificate private key.                                           |

| Parameter   | Required or Optional | Default Value | Description                                                                                                                                                                                                  |
|-------------|----------------------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Environment | Optional             | AzureCloud    | The name of the supported Cloud Environment in which the target Azure Active Directory Graph Service is available. Allowed values: 'AzureCloud', 'AzureChinaCloud', 'AzureUSGovernment', 'AzureGermanCloud'. |

## Create an Active Directory Federation Services app

For Azure Stack Hub environments secured by AD FS, you must configure an AD FS service principal to support the following operations:

- Virtual machine scale set integration on worker tiers.
- SSO for the Azure Functions portal and advanced developer tools.

Admins must configure SSO to:

- Configure a service principal for virtual machine scale set integration on worker tiers.
- Enable the advanced developer tools within App Service (Kudu).
- Enable the use of the Azure Functions portal experience.

Follow these steps:

1. Open a PowerShell instance as `azurestack\AzureStackAdmin`.
2. Go to the location of the scripts that you downloaded and extracted in the [prerequisite step](#).
3. [Install PowerShell for Azure Stack Hub](#).
4. Run the **Create-ADFSIdentityApp.ps1** script.
5. In the **Credential** window, enter your AD FS cloud admin account and password. Select **OK**.
6. Provide the certificate file path and certificate password for the [certificate created earlier](#). The certificate created for this step by default is **sso.appservice.local.azurestack.external.pfx**.

### Create-ADFSIdentityApp.ps1

| Parameter            | Required or Optional | Default Value | Description                                                                                                    |
|----------------------|----------------------|---------------|----------------------------------------------------------------------------------------------------------------|
| AdminArmEndpoint     | Required             | Null          | Admin Azure Resource Manager endpoint. An example is <code>adminmanagement.local.azurestack.external</code> .  |
| PrivilegedEndpoint   | Required             | Null          | Privileged endpoint. An example is <code>AzS-ERCS01</code> .                                                   |
| CloudAdminCredential | Required             | Null          | Domain account credential for Azure Stack Hub cloud admins. An example is <code>Azurestack\CloudAdmin</code> . |

| PARAMETER           | REQUIRED OR OPTIONAL | DEFAULT VALUE | DESCRIPTION                                                          |
|---------------------|----------------------|---------------|----------------------------------------------------------------------|
| CertificateFilePath | Required             | Null          | <b>Full path</b> to the identity application's certificate PFX file. |
| CertificatePassword | Required             | Null          | Password that helps protect the certificate private key.             |

## Next steps

[Install the App Service resource provider](#)

# Deploy App Service in Azure Stack Hub

9 minutes to read • [Edit Online](#)

This article describes how to deploy App Service in Azure Stack Hub.

## IMPORTANT

Apply the 1910 update to your Azure Stack Hub integrated system or deploy the latest Azure Stack Development Kit (ASDK) before you deploy Azure App Service 1.8.

You can give your users the ability to create web and API applications. To let users create these apps, you need to:

- Add the [App Service resource provider](#) to your Azure Stack Hub deployment using the steps described in this article.
- After you install the App Service resource provider, you can include it in your offers and plans. Users can then subscribe to get the service and start creating apps.

## IMPORTANT

Before you run the resource provider installer, make sure that you've followed the guidance in [Before you get started](#) and have read the [release notes](#) which accompany the 1.8 release. Reading this content helps you learn about new functionality, fixes, and any known issues which could affect your deployment.

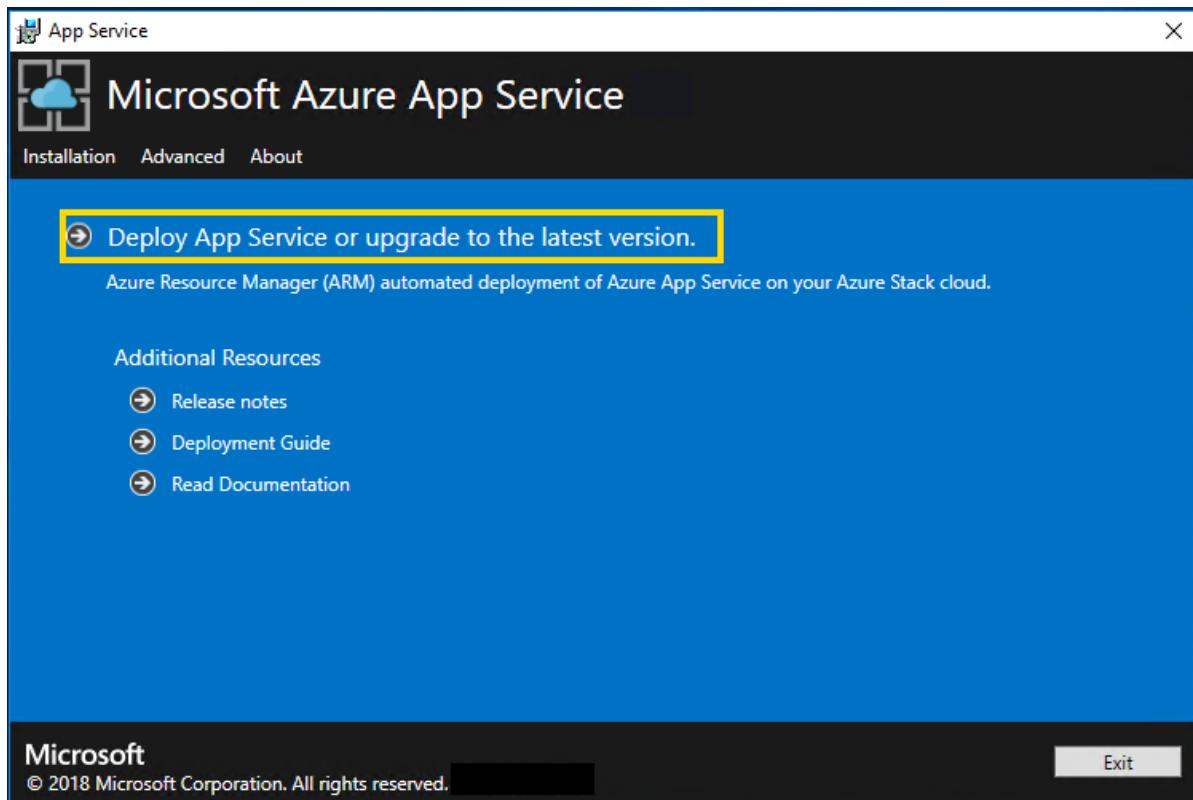
## Run the App Service resource provider installer

Installing the App Service resource provider takes at least an hour. The length of time needed depends on how many role instances you deploy. During the deployment, the installer runs the following tasks:

- Create a blob container in the specified Azure Stack Hub storage account.
- Create a DNS zone and entries for App Service.
- Register the App Service resource provider.
- Register the App Service gallery items.

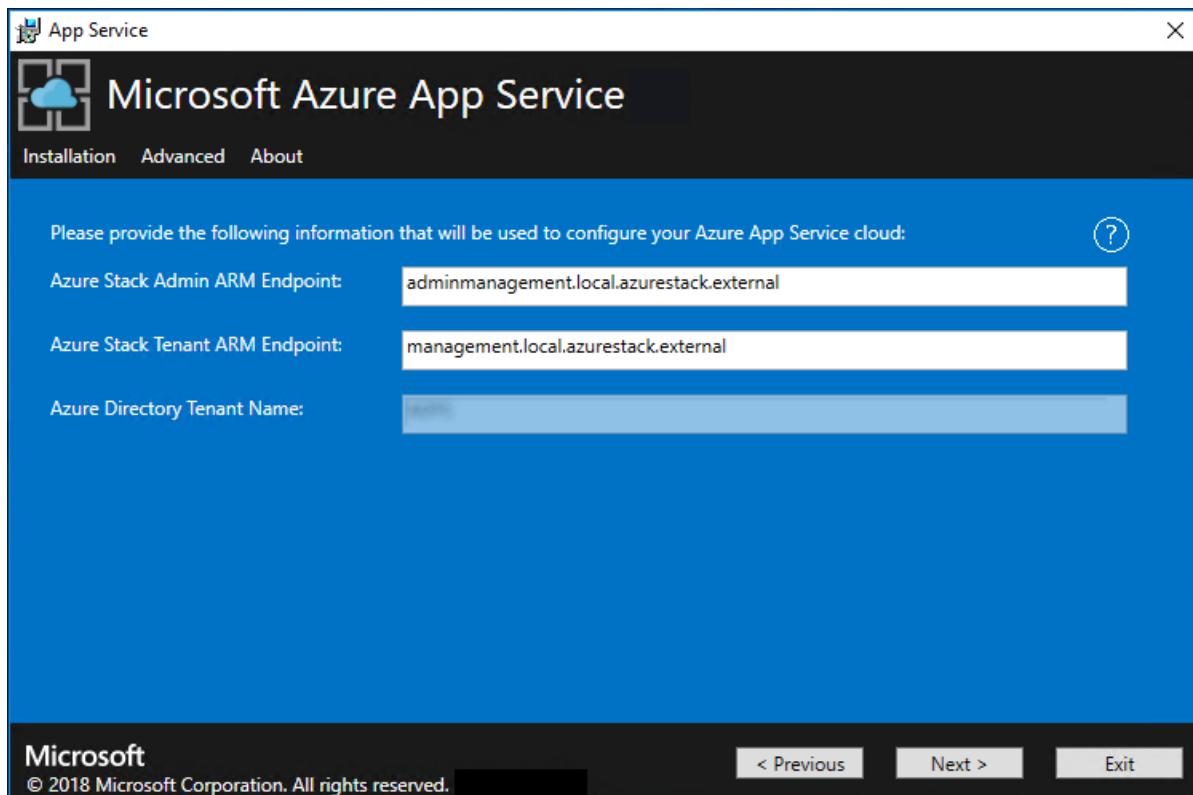
To deploy App Service resource provider, follow these steps:

1. Run appservice.exe as an admin from a computer that can access the Azure Stack Hub Admin Azure Resource Management Endpoint.
2. Select **Deploy App Service or upgrade to the latest version.**



3. Review and accept the Microsoft Software License Terms and then select **Next**.
4. Review and accept the third-party license terms and then select **Next**.
5. Make sure that the App Service cloud configuration information is correct. If you used the default settings during ASDK deployment, you can accept the default values. But, if you customized the options when you deployed the ASDK, or are deploying on an Azure Stack Hub integrated system, you must edit the values in this window to reflect the differences.

For example, if you use the domain suffix mycloud.com, your Azure Stack Hub Tenant Azure Resource Manager endpoint must change to management.<region>.mycloud.com. Review these settings, and then select **Next** to save the settings.



6. On the next App Service Installer page you will connect to your Azure Stack Hub:

- Select the connection method you wish to use - **Credential** or **Service Principal**

- **Credential**

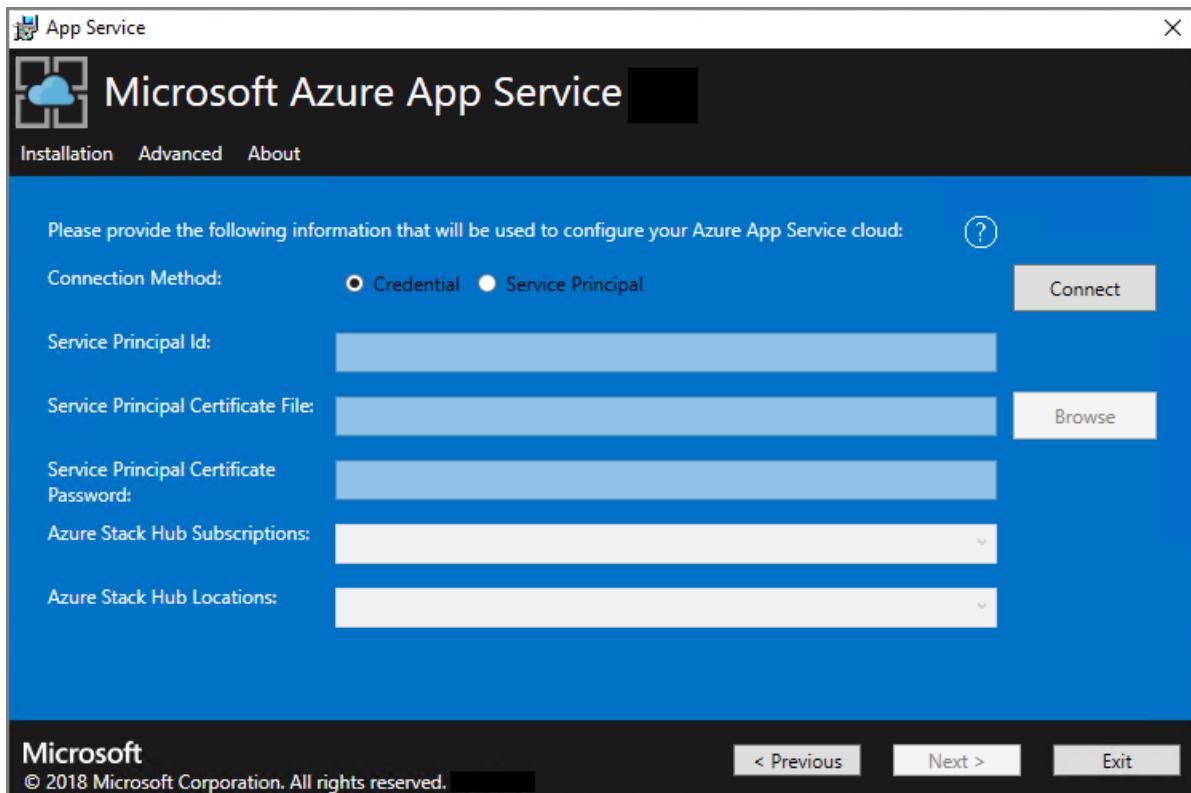
- If you're using Azure Active Directory (Azure AD), enter the Azure AD admin account and password that you provided when you deployed Azure Stack Hub. Select **Connect**.
- If you're using Active Directory Federation Services (AD FS), provide your admin account. For example, clouadmin@azurestack.local. Enter your password, and then select **Connect**.

- **Service Principal**

- The service principal which you use **must** have **Owner** rights on the **Default Provider Subscription**
- Provide the **Service Principal ID**, **Certificate File** and **Password** and select **Connect**.

b. In **Azure Stack Hub Subscriptions**, select the **Default Provider Subscription**. Azure App Service on Azure Stack Hub **must** be deployed in the **Default Provider Subscription**.

c. In the **Azure Stack Hub Locations**, select the location that corresponds to the region you're deploying to. For example, select **local** if you're deploying to the ASDK.

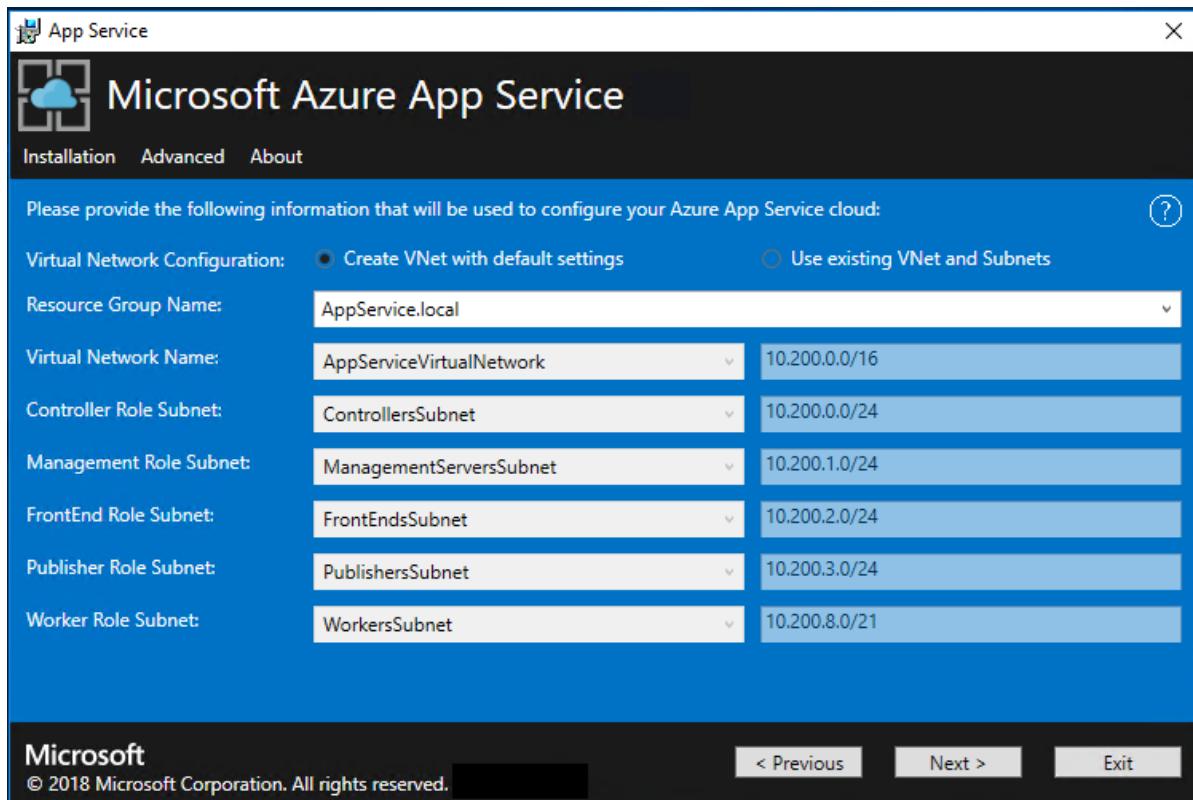


7. Now you can deploy into an existing virtual network that you configured [using these steps](#), or let the App Service installer create a new virtual network and subnets. To create a VNet, follow these steps:

a. Select **Create VNet with default settings**, accept the defaults, and then select **Next**.

b. Alternatively, select **Use existing VNet and Subnets**. Complete the following actions:

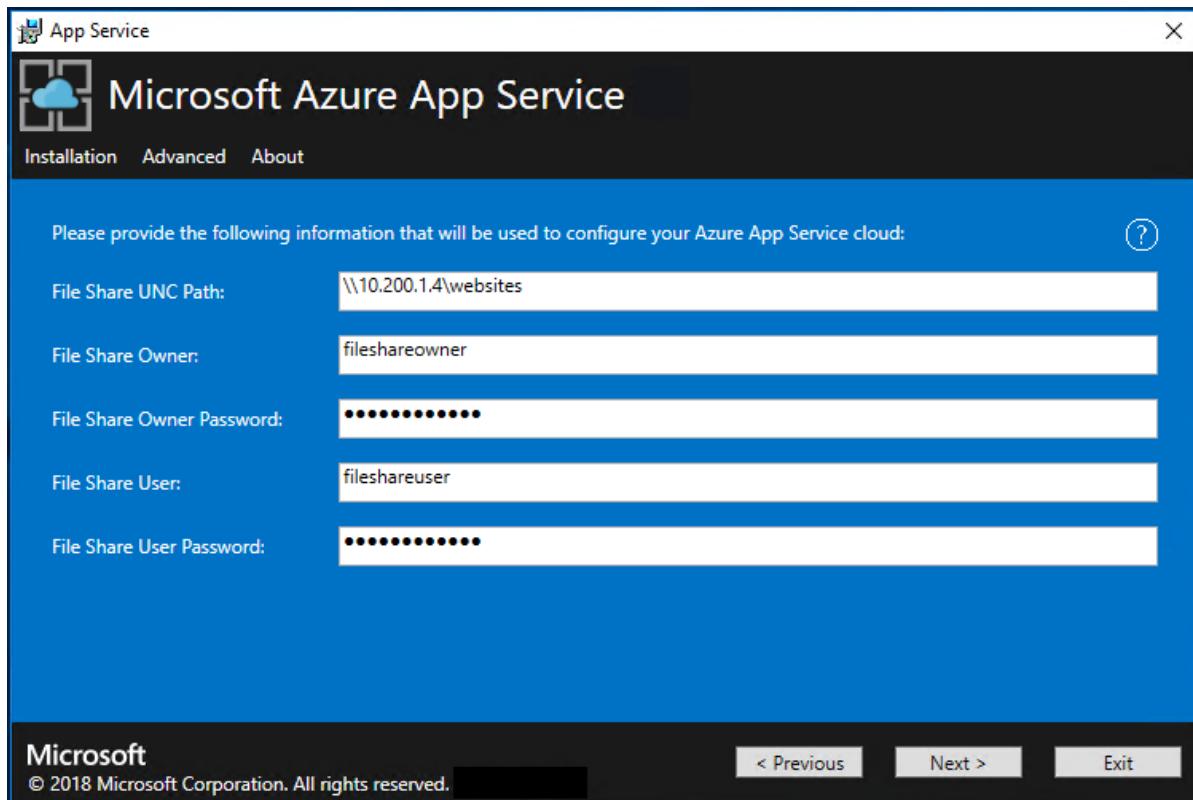
- Select the **Resource Group** that contains your virtual network.
- Choose the **Virtual Network** name that you want to deploy to.
- Select the correct **Subnet** values for each of the required role subnets.
- Select **Next**.



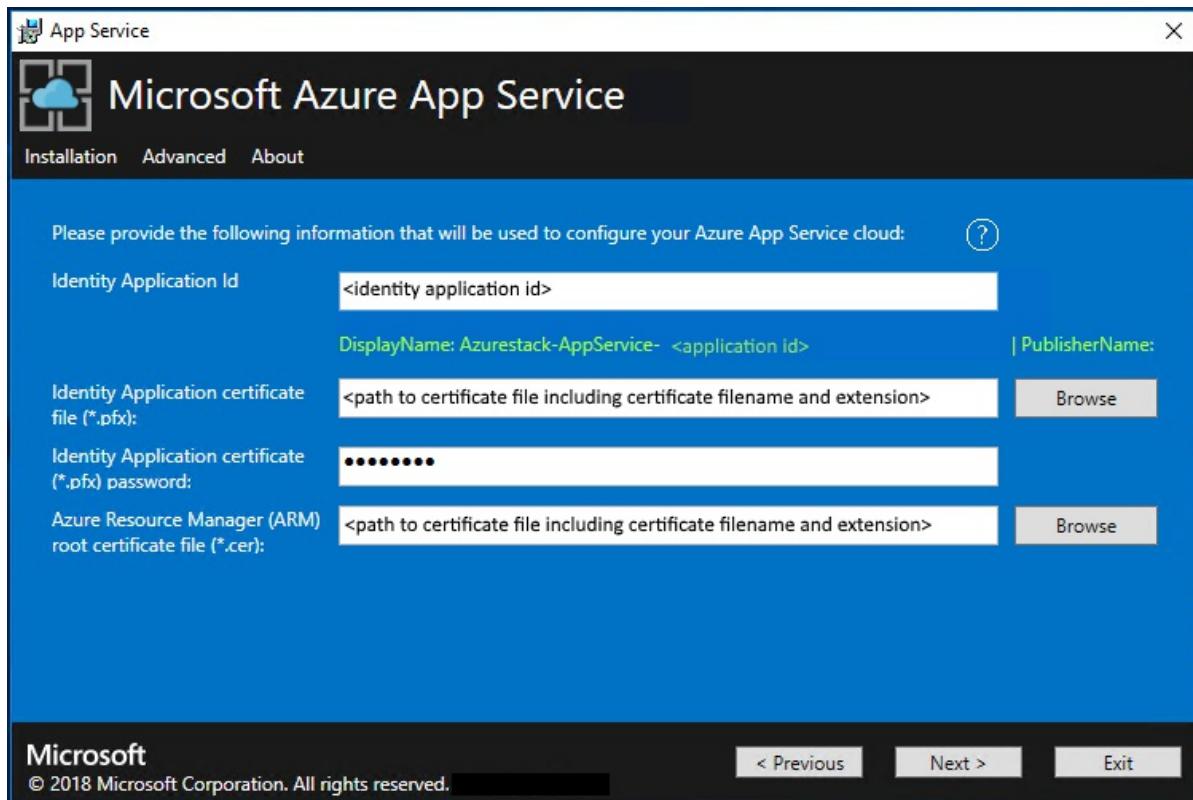
8. Enter the info for your file share and then select **Next**. The address of the file share must use the Fully Qualified Domain Name (FQDN) or the IP address of your File Server. For example, \\appservicefileserv.local.cloudapp.azurestack.external\websites, or \\10.0.0.1\websites. If you're using a file server, which is domain joined, you must provide the full username including domain. For example, myfileserverdomain\FileShareOwner.

**NOTE**

The installer tries to test connectivity to the file share before proceeding. But, if you're deploying to an existing virtual network, this connectivity test might fail. You're given a warning and a prompt to continue. If the file share info is correct, continue the deployment.



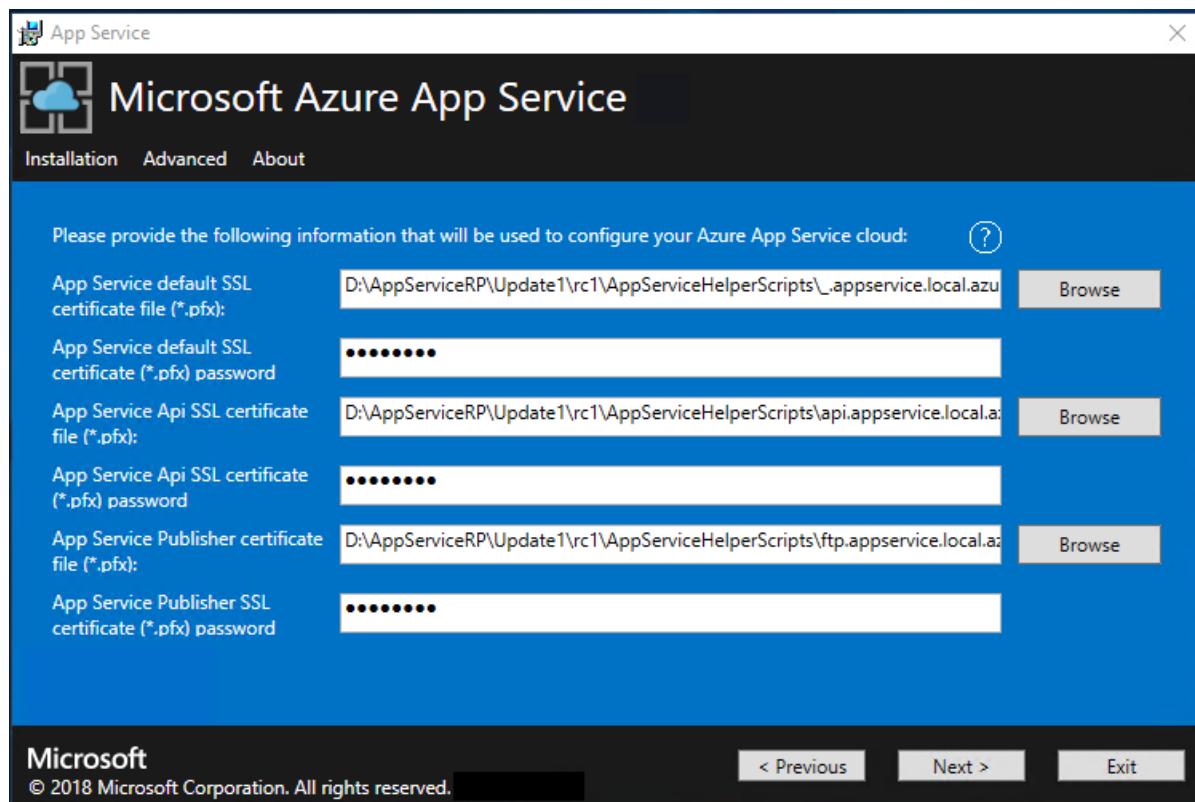
9. On the next App Service Installer page, follow these steps:
  - a. In the **Identity Application ID** box, enter the GUID for the app you're using for identity (from Azure AD).
  - b. In the **Identity Application certificate file** box, enter (or browse to) the location of the certificate file.
  - c. In the **Identity Application certificate password** box, enter the password for the certificate. This password is the one that you made note of when you used the script to create the certificates.
  - d. In the **Azure Resource Manager root certificate file** box, enter (or browse to) the location of the certificate file.
  - e. Select **Next**.



10. For each of the three certificate file boxes, select **Browse** and navigate to the appropriate certificate file. You must provide the password for each certificate. These certificates are the ones that you created in the [Create required certificates step](#). Select **Next** after entering all the information.

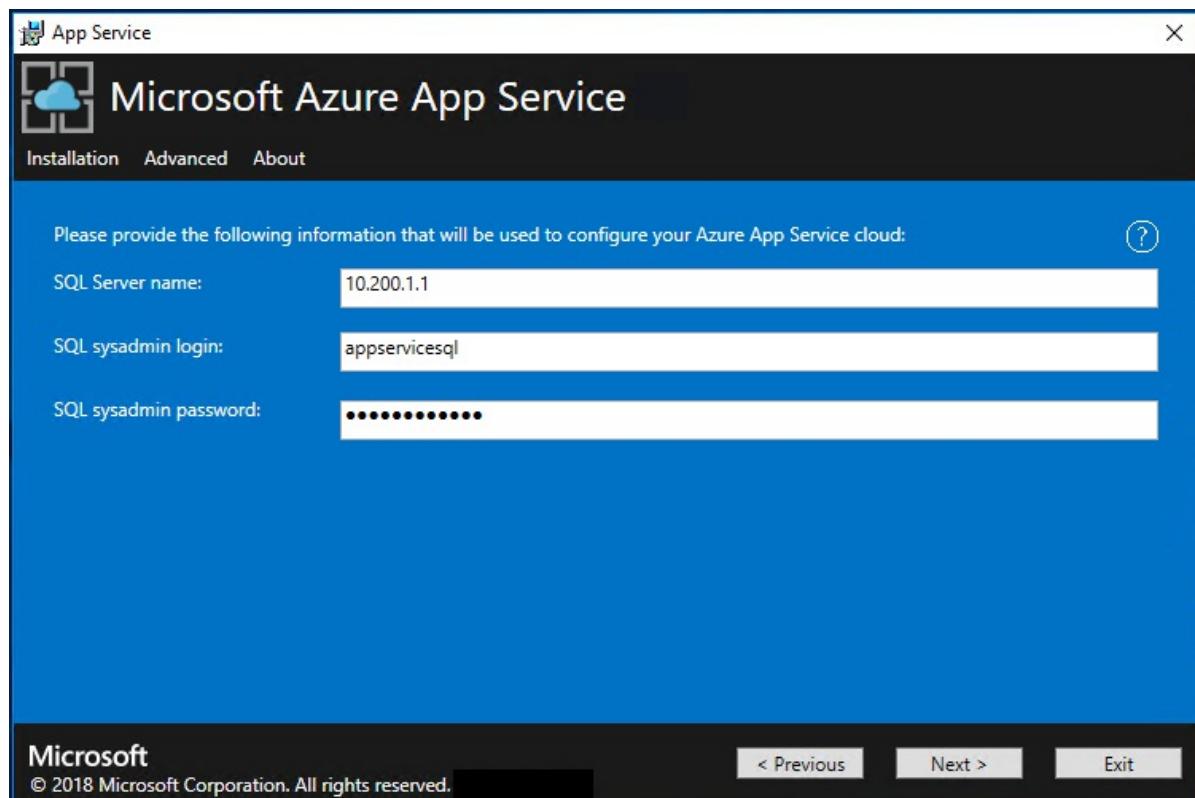
| BOX                                               | CERTIFICATE FILE NAME EXAMPLE                |
|---------------------------------------------------|----------------------------------------------|
| <b>App Service default SSL certificate file</b>   | _appservice.local.AzureStack.external.pfx    |
| <b>App Service API SSL certificate file</b>       | api.appservice.local.AzureStack.external.pfx |
| <b>App Service Publisher SSL certificate file</b> | ftp.appservice.local.AzureStack.external.pfx |

If you used a different domain suffix when you created the certificates, your certificate file names don't use *local.AzureStack.external*. Instead, use your custom domain info.



11. Enter the SQL Server details for the server instance used to host the App Service resource provider database and then select **Next**. The installer validates the SQL connection properties.

The App Service installer tries to test connectivity to the SQL Server before proceeding. If you're deploying to an existing virtual network, this connectivity test might fail. You're given a warning and a prompt to continue. If the SQL Server info is correct, continue the deployment.

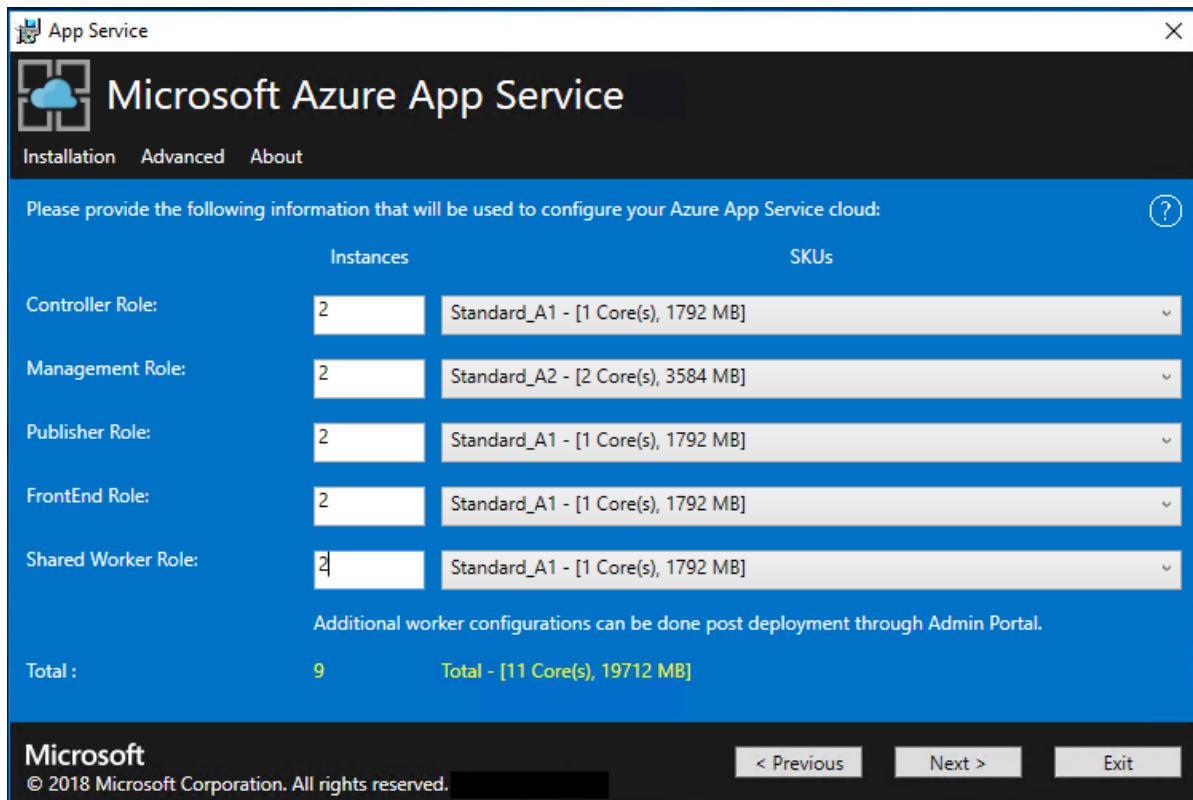


12. Review the role instance and SKU options. The defaults populate with the minimum number of instances and the minimum SKU for each role in an ASDK deployment. A summary of vCPU and memory requirements is provided to help plan your deployment. After you make your selections, select **Next**.

**NOTE**

For production deployments, following the guidance in [Capacity planning for Azure App Service server roles in Azure Stack Hub](#).

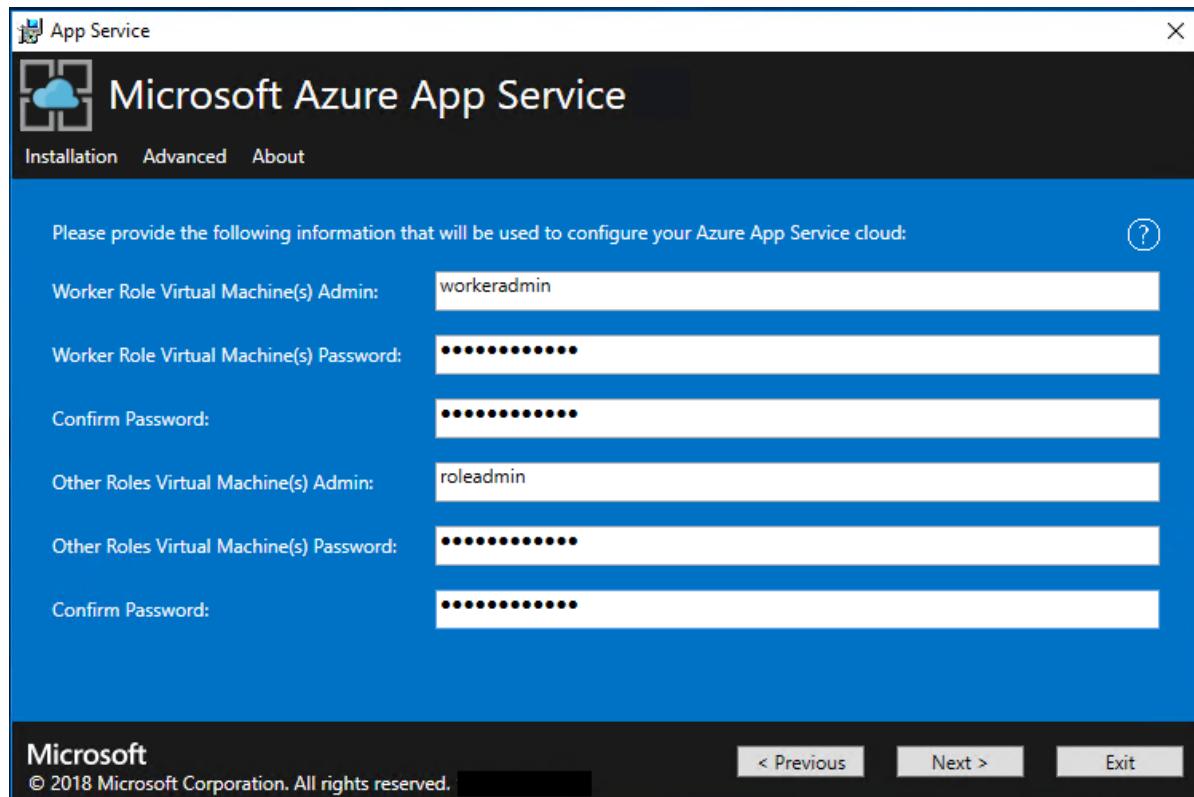
| ROLE          | MINIMUM INSTANCES | MINIMUM SKU                      | NOTES                                                                                                                                                                                                       |
|---------------|-------------------|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Controller    | 1                 | Standard_A2 - (2 vCPU, 3584 MB)  | Manages and maintains the health of the App Service cloud.                                                                                                                                                  |
| Management    | 1                 | Standard_A2 - (2 vCPUs, 3584 MB) | Manages the App Service Azure Resource Manager and API endpoints, portal extensions (admin, tenant, Functions portal), and the data service. To support failover, increased the recommended instances to 2. |
| Publisher     | 1                 | Standard_A1 - (1 vCPU, 1792 MB)  | Publishes content via FTP and web deployment.                                                                                                                                                               |
| FrontEnd      | 1                 | Standard_A1 - (1 vCPU, 1792 MB)  | Routes requests to App Service apps.                                                                                                                                                                        |
| Shared Worker | 1                 | Standard_A1 - (1 vCPU, 1792 MB)  | Hosts web or API apps and Azure Functions apps. You might want to add more instances. As an operator, you can define your offering and choose any SKU tier. The tiers must have a minimum of one vCPU.      |



#### NOTE

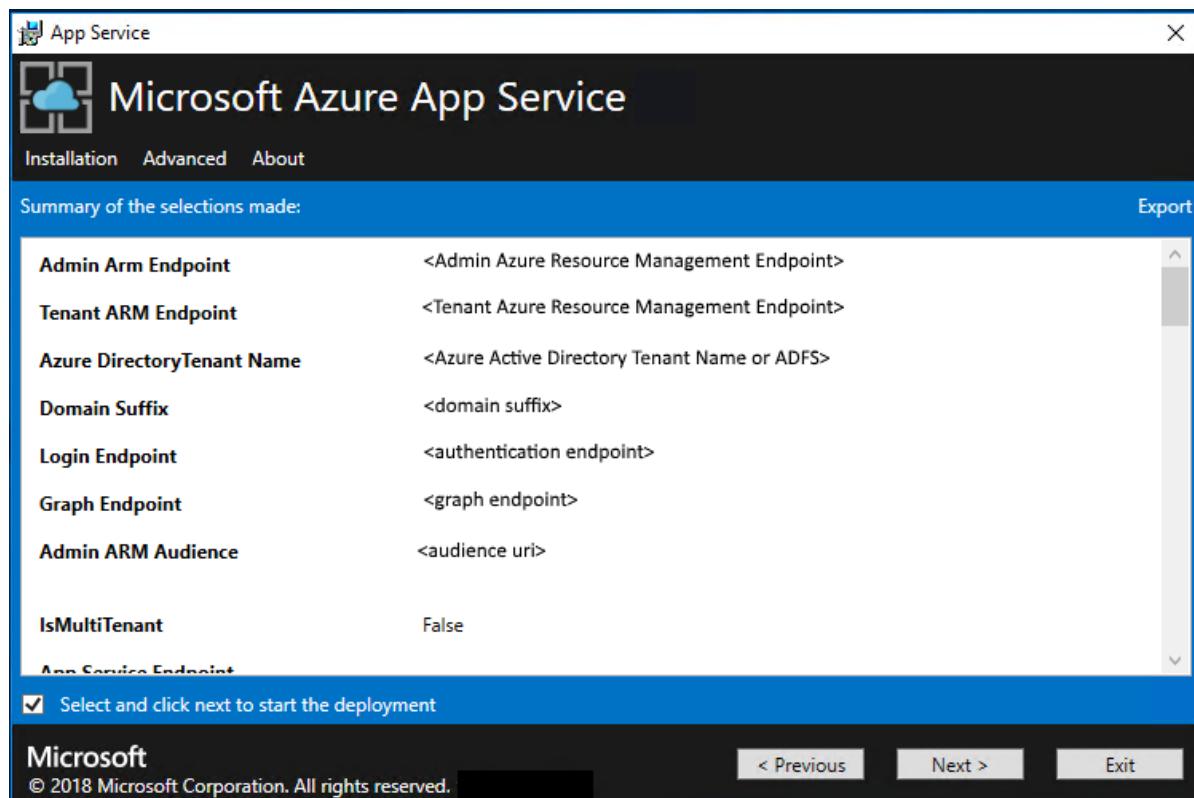
**Windows Server 2016 Core isn't a supported platform image for use with Azure App Service on Azure Stack Hub. Don't use evaluation images for production deployments.**

13. In the **Select Platform Image** box, choose your deployment Windows Server 2016 virtual machine (VM) image from the images available in the compute resource provider for the App Service cloud. Select **Next**.
14. On the next App Service Installer page, follow these steps:
  - a. Enter the Worker Role VM admin user name and password.
  - b. Enter the Other Roles VM admin user name and password.
  - c. Select **Next**.



15. On the App Service Installer summary page, follow these steps:

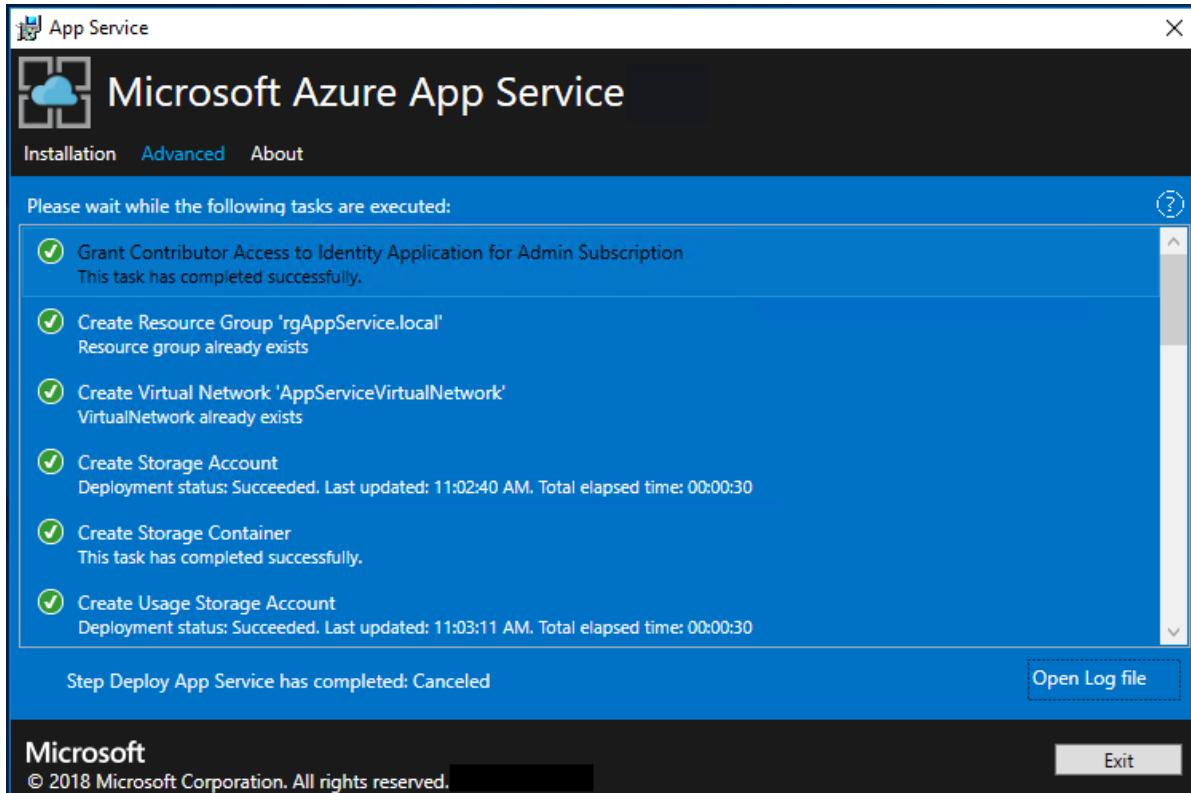
- Verify the selections you made. To make changes, use the **Previous** buttons to visit previous pages.
- If the configurations are correct, select the check box.
- To start the deployment, select **Next**.



16. On the next App Service Installer page, follow these steps:

- Track the installation progress. App Service on Azure Stack Hub can take up to 240 minutes to deploy based on the default selections and age of the base Windows 2016 Datacenter image.

b. After the installer successfully finishes, select **Exit**.



## Post-deployment Steps

### IMPORTANT

If you've provided the App Service RP with a SQL Always On Instance you **must** [add the appservice\\_hosting and appservice\\_metering databases to an availability group](#) and synchronize the databases to prevent any loss of service in the event of a database failover.

If you're deploying to an existing virtual network and using an internal IP address to connect to your file server, you must add an outbound security rule. This rule enables SMB traffic between the worker subnet and the file server. In the administrator portal, go to the WorkersNsg Network Security Group and add an outbound security rule with the following properties:

- Source: Any
- Source port range: \*
- Destination: IP addresses
- Destination IP address range: Range of IPs for your file server
- Destination port range: 445
- Protocol: TCP
- Action: Allow
- Priority: 700
- Name: Outbound\_Allow\_SMB445

## Validate the App Service on Azure Stack Hub installation

1. In the Azure Stack Hub administrator portal, go to **Administration - App Service**.
2. In the overview, under status, check to see that the **Status** displays **All roles are ready**.

The screenshot shows the Microsoft Azure Stack - Administration portal. On the left, there's a navigation sidebar with icons for Home, Create, App Service, Functions, Logic Apps, Data Factory, Container Registry, and more. The 'App Service' icon is selected. The main area has a title 'App Service' and a 'local' indicator. A search bar says 'Search (Ctrl+Shift+F)'. Below it are several tabs: Overview (selected), Properties, System configuration, Source control configuration, Roles, IP SSL, IP Block List, Worker Tiers, SKUs, Pricing Tiers, Subscription Quotas, and Custom Software. To the right, under 'Essentials', it shows 'Resource group AppService.local', 'Status All roles are ready' (which is highlighted with a red box), 'Location local', and 'Subscription ID <subscription id>'. At the bottom is a chart titled 'System' with a Y-axis from 0 to 100 and an X-axis showing time. The chart shows a blue line with some minor fluctuations.

## Test drive App Service on Azure Stack Hub

After you deploy and register the App Service resource provider, test it to make sure that users can deploy web and API apps.

### NOTE

You need to create an offer that has the Microsoft.Web namespace in the plan. You also need a tenant subscription that subscribes to the offer. For more info, see [Create offer](#) and [Create plan](#).

You *must* have a tenant subscription to create apps that use App Service on Azure Stack Hub. The only tasks that a service admin can complete in the administrator portal are related to the resource provider administration of App Service. This includes adding capacity, configuring deployment sources, and adding Worker tiers and SKUs.

To create web, API, and Azure Functions apps, you must use the user portal and have a tenant subscription.

To create a test web app, follow these steps:

1. In the Azure Stack Hub user portal, select + **Create a resource** > **Web + Mobile** > **Web App**.
2. Under **Web App**, enter a name in **Web app**.
3. Under **Resource Group**, select **New**. Enter a name for the **Resource Group**.
4. Select **App Service plan/Location** > **Create New**.
5. Under **App Service plan**, enter a name for the **App Service plan**.
6. Select **Pricing tier** > **Free-Shared** or **Shared-Shared** > **Select** > **OK** > **Create**.
7. A tile for the new web app appears on the dashboard. Select the tile.

8. On **Web App**, select **Browse** to view the default website for this app.

## Deploy a WordPress, DNN, or Django website (optional)

1. In the Azure Stack Hub user portal, select **+**, go to the Azure Marketplace, deploy a Django website, and then wait for the deployment to finish. The Django web platform uses a file system-based database. It doesn't require any additional resource providers, such as SQL or MySQL.
2. If you also deployed a MySQL resource provider, you can deploy a WordPress website from the Marketplace. When you're prompted for database parameters, enter the user name as *User1@Server1*, with the user name and server name of your choice.
3. If you also deployed a SQL Server resource provider, you can deploy a DNN website from the Marketplace. When you're prompted for database parameters, choose a database in the computer running SQL Server that's connected to your resource provider.

## Next steps

Prepare for additional admin operations for App Service on Azure Stack Hub:

- [Capacity Planning](#)
- [Configure Deployment Sources](#)

# Deploy App Service in a highly available configuration

8 minutes to read • [Edit Online](#)

This article shows you how to use Azure Stack Hub Marketplace items to deploy App Service for Azure Stack Hub in a highly available configuration. In addition to available marketplace items, this solution also uses the [appservice-fileshare-sqlserver-ha](#) Azure Stack Hub Quickstart template. This template automates the creation of a highly available infrastructure for hosting the App Service resource provider. App Service is then installed on this highly available VM infrastructure.

## Deploy the highly available App Service infrastructure VMs

The [appservice-fileshare-sqlserver-ha](#) Azure Stack Hub Quickstart template simplifies the deployment of App Service in a highly available configuration. It should be deployed in the Default Provider Subscription.

When used to create a custom resource in Azure Stack Hub, the template creates:

- A virtual network and required subnets.
- Network security groups for file server, SQL Server, and Active Directory Domain Services (AD DS) subnets.
- Storage accounts for VM disks and cluster cloud witness.
- One internal load balancer for SQL VMs with private IP bound to the SQL AlwaysOn listener.
- Three availability sets for the AD DS, file server cluster, and the SQL cluster.
- Two node SQL cluster.
- Two node file server cluster.
- Two domain controllers.

### Required Azure Stack Hub Marketplace items

Before using this template, ensure that the following [Azure Stack Hub Marketplace items](#) are available in your Azure Stack Hub instance:

- Windows Server 2016 Datacenter Core Image (for AD DS and file server VMs)
- SQL Server 2016 SP2 on Windows Server 2016 (Enterprise)
- Latest SQL IaaS Extension
- Latest PowerShell Desired State Configuration Extension

#### TIP

Review the [template readme file](#) on GitHub for additional details on template requirements and default values.

## Deploy the App Service infrastructure

Use the steps in this section to create a custom deployment using the [appservice-fileshare-sqlserver-ha](#) Azure Stack Hub Quickstart template.

### 1. Sign in to the administrator portal:

- For an integrated system deployment, the portal address varies based on your solution's region and external domain name. The address is in this format: `https://adminportal.<region>.<FQDN>`.
- For the Azure Stack Development Kit (ASDK), the portal address is

<https://adminportal.local.azurestack.external>.

2. Select **+ Create a resource** > **Custom**, and then **Template deployment**.

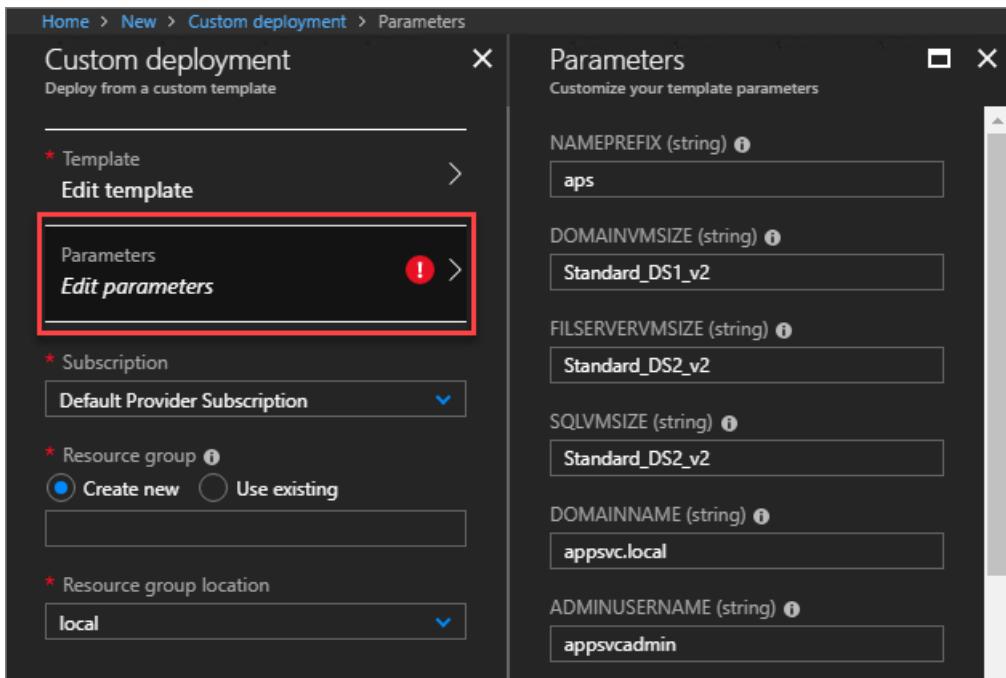
The screenshot shows the Microsoft Azure Stack Administration interface. On the left, there's a sidebar with various navigation options like 'Dashboard', 'All resources', 'Resource groups', etc. In the center, under 'New', there's a 'Search resources' bar and a 'Featured' section. A red box highlights the 'Custom' category, which is selected. To the right, a larger window titled 'Custom deployment' is open, showing fields for 'Template' (with 'Edit template'), 'Parameters' (with 'Edit parameters'), 'Subscription' (set to 'Default Provider Subscription'), 'Resource group' (radio buttons for 'Create new' or 'Use existing'), and 'Resource group location' (set to 'local'). A blue 'Create' button is at the bottom.

3. On the **Custom deployment** blade, select **Edit template** > **Quickstart template** and then use the drop-down list of available custom templates to select the **appservice-fileshare-sqlserver-ha** template. Click **OK**, and then **Save**.

The screenshot shows the 'Edit template' blade. At the top, there are buttons for 'Quickstart template', 'Load file', and 'Download'. Below that, a section titled 'Load a quickstart template' has a dropdown menu. A red box highlights this dropdown, and the option 'appservice-fileshare-sqlserver-ha' is selected. Below the dropdown, there's a brief description: 'Creates VNET, 2 node SQL cluster, 2 node FileServer cluster and 2 domain controllers'. It also shows the author 'appleby64' and the last update date '2018-09-21'. At the bottom, there are 'OK' and 'Cancel' buttons, with 'OK' being highlighted by a red box.

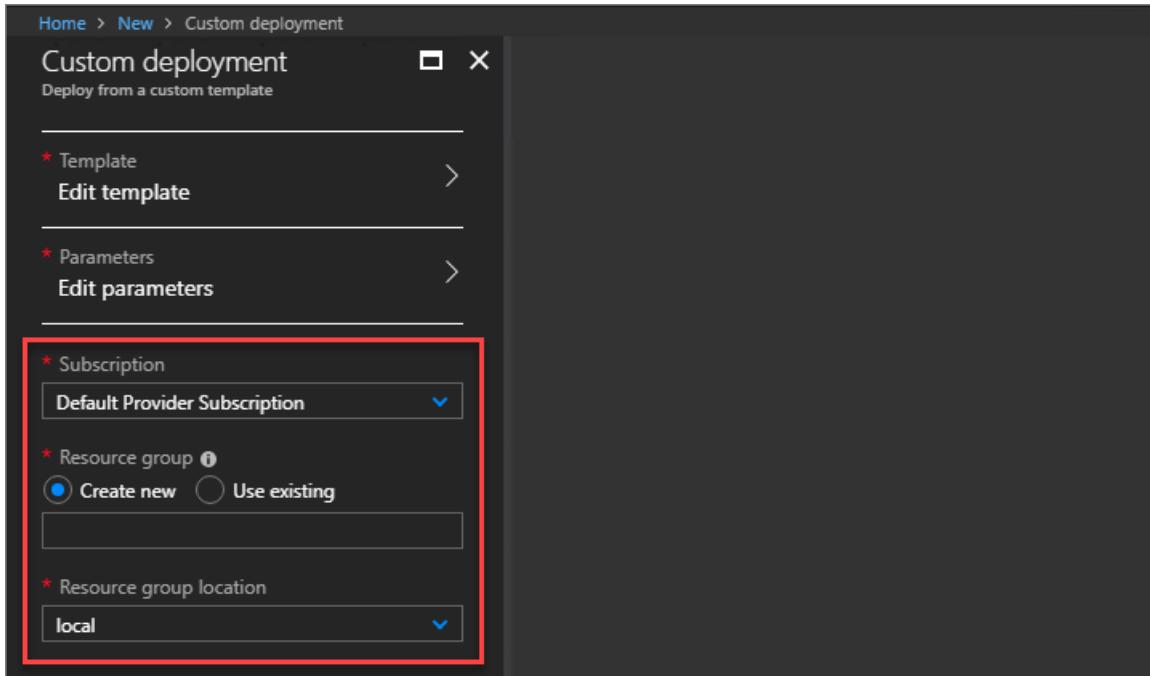
4. On the **Custom deployment** blade, select **Edit parameters** and scroll down to review the default template values. Modify these values as necessary to provide all required parameter info and then click **OK**.

At a minimum, provide complex passwords for the **ADMINPASSWORD**, **FILESHAREOWNERPASSWORD**, **FILESHAREUSERPASSWORD**, **SQLSERVERSERVICEACCOUNTPASSWORD**, and **SQLLOGINPASSWORD** parameters.



- On the **Custom deployment** blade, ensure **Default Provider Subscription** is selected as the subscription to use and then create a new resource group, or select an existing resource group, for the custom deployment.

Next, select the resource group location (**local** for ASDK installations) and then click **Create**. The custom deployment settings are validated before template deployment starts.



- In the administrator portal, select **Resource groups** and then the name of the resource group you created for the custom deployment (**app-service-ha** in this example). View the status of the deployment to ensure all deployments have completed successfully.

#### NOTE

The template deployment takes about an hour to complete.

| Subscription (change)         | Subscription ID                      | Deployments  |
|-------------------------------|--------------------------------------|--------------|
| Default Provider Subscription | bca36b9c-6fee-4c94-9cf5-f65bbcba5dc4 | 21 Succeeded |

Tags (change)  
Click here to add tags

Filter by name... All types

81 items Show hidden types ⓘ

| NAME ↑↓                              | TYPE ↑↓         |
|--------------------------------------|-----------------|
| appservice.local.azurestack.external | DNS zone        |
| appserviceusage                      | Storage account |
| appsvcadminhydration                 | Storage account |

## Record template outputs

After the template deployment completes successfully, record the template deployment outputs. You need this info when running the App Service installer.

Ensure you record each of these output values:

- FileSharePath
- FileShareOwner
- FileShareUser
- SQLserver
- SQLuser

Follow these steps to discover the template output values:

1. Sign in to the administrator portal:
  - For an integrated system deployment, the portal address varies based on your solution's region and external domain name. The address is in this format: <https://adminportal.<region>.<FQDN>>.
  - For the Azure Stack Development Kit (ASDK), the portal address is <https://adminportal.local.azurestack.external>.
2. In the administrator portal, select **Resource groups** and then the name of the resource group you created for the custom deployment (**app-service-ha** in this example).
3. Click **Deployments** and select **Microsoft.Template**.

| DEPLOYMENT NAME                   | STATUS    | LAST MODIFIED          | DURATION                    |
|-----------------------------------|-----------|------------------------|-----------------------------|
| AppService.DeployCloud            | Succeeded | 3/12/2019, 5:48:52 PM  | 7 minutes 21 seconds        |
| SmallWorkerTierScaleSetDeploy     | Succeeded | 3/12/2019, 5:43:12 PM  | 43 seconds                  |
| MediumWorkerTierScaleSetDeploy    | Succeeded | 3/12/2019, 5:43:12 PM  | 46 seconds                  |
| SharedWorkerTierScaleSetDeploy    | Succeeded | 3/12/2019, 5:43:11 PM  | 47 seconds                  |
| LargeWorkerTierScaleSetDeploy     | Succeeded | 3/12/2019, 5:43:06 PM  | 40 seconds                  |
| AppService.DeployTenantHydrati... | Succeeded | 3/12/2019, 3:47:12 PM  | 39 seconds                  |
| AppService.DeployAdminHydrati...  | Succeeded | 3/12/2019, 3:46:09 PM  | 37 seconds                  |
| AppService.DeployUsageStorage     | Succeeded | 3/12/2019, 3:45:17 PM  | 46 seconds                  |
| AppService.DeployStorage          | Succeeded | 3/12/2019, 3:44:11 PM  | 41 seconds                  |
| <b>Microsoft.Template</b>         | Succeeded | 3/12/2019, 1:08:08 PM  | 1 hour 5 minutes 29 seconds |
| DeploySQL                         | Succeeded | 3/12/2019, 1:07:49 PM  | 36 minutes 11 seconds       |
| deployS2DCluster                  | Succeeded | 3/12/2019, 12:49:28 PM | 17 minutes 49 seconds       |

- After selecting the **Microsoft.Template** deployment, select **Outputs** and record the template parameter output. This info is required when deploying App Service.

|                |                               |
|----------------|-------------------------------|
| FILESHAREPATH  | \\\fs01.appsvc.local\WebSites |
| FILESHAREOWNER | appsvc.local\FileShareOwner   |
| FILESHAREUSER  | appsvc.local\FileShareUser    |
| SQLSERVER      | 10.0.1.100                    |
| SQLUSER        | sa                            |

## Deploy App Service in a highly available configuration

Follow the steps in this section to deploy App Service for Azure Stack Hub in a highly available configuration based on the [appservice-fileshare-sqlserver-ha](#) Azure Stack Hub Quickstart template.

After you install the App Service resource provider, you can include it in your offers and plans. Users can then subscribe to get the service and start creating apps.

### IMPORTANT

Before you run the resource provider installer, make sure that you've read the release notes, which accompany each App Service release, to learn about new functionality, fixes, and any known issues which could affect your deployment.

### Prerequisites

Before you can run the App Service installer, several steps are required as described in the [Before you get started with App Service on Azure Stack Hub](#) article:

**TIP**

Not all steps described in the [Before you get started with App Service article](#) are required because the template deployment configures the infrastructure VMs for you.

- [Download the App Service installer and helper scripts.](#)
- [Download items from the Azure Stack Hub Marketplace.](#)
- [Generate required certificates.](#)
- Create the ID Application based on the identify provider you've chosen for Azure Stack Hub. An ID Application can be made for either [Azure AD](#) or [Active Directory Federation Services](#) and record the application ID.
- Ensure that you've added the Windows Server 2016 Datacenter image to the Azure Stack Hub Marketplace. This image is required for App Service installation.

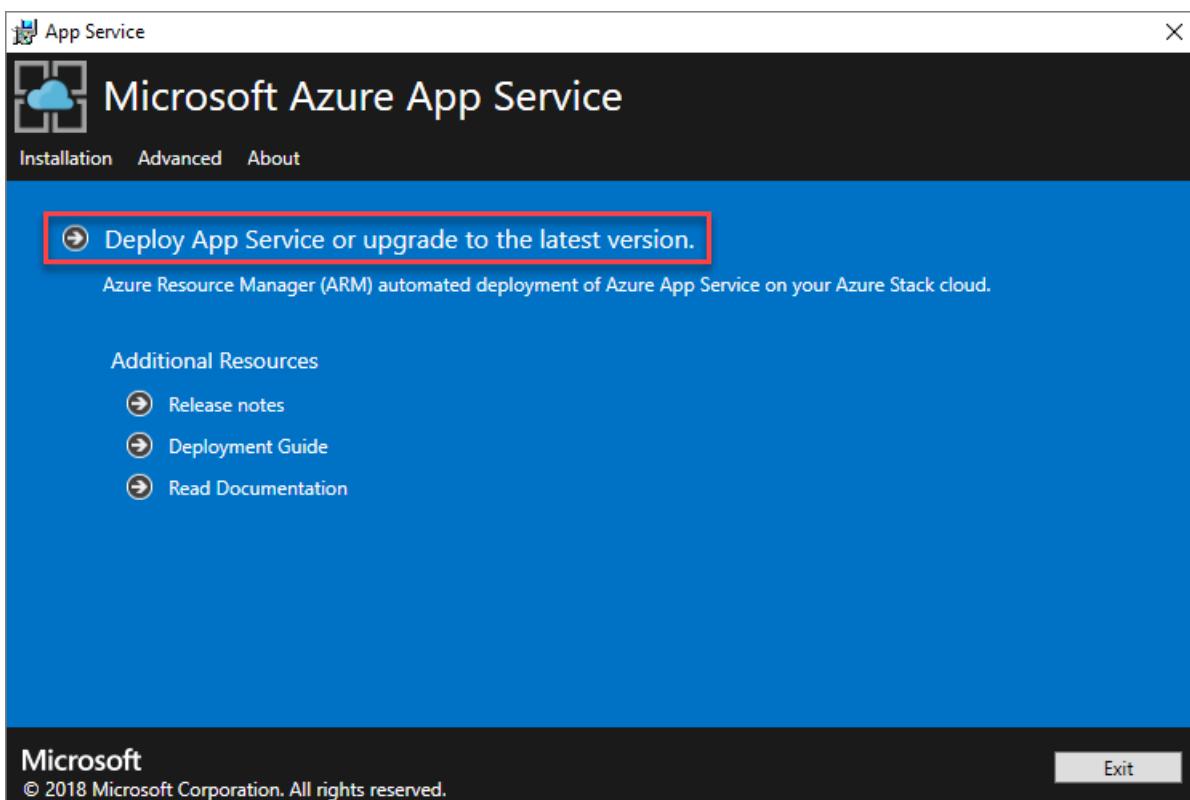
### Steps for App Service deployment

Installing the App Service resource provider takes at least an hour. The length of time needed depends on how many role instances you deploy. During the deployment, the installer runs the following tasks:

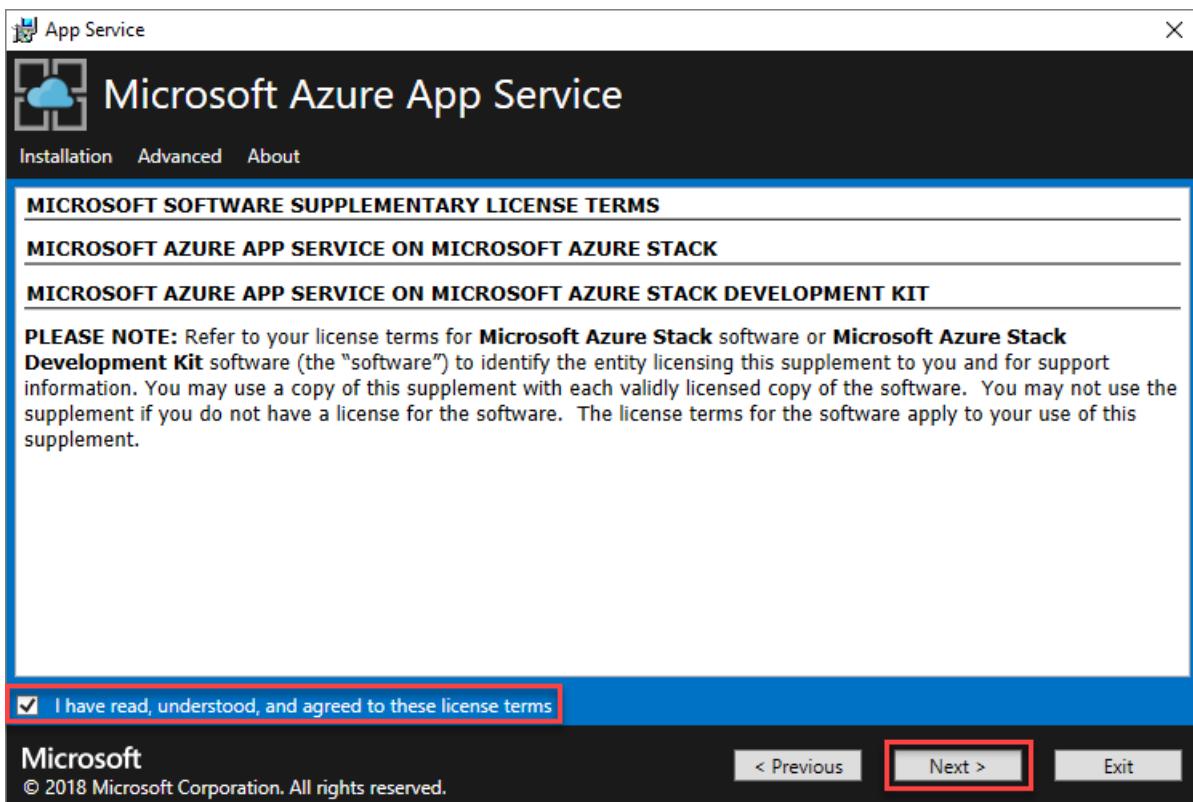
- Create a blob container in the specified Azure Stack Hub storage account.
- Create a DNS zone and entries for App Service.
- Register the App Service resource provider.
- Register the App Service gallery items.

To deploy the App Service resource provider, follow these steps:

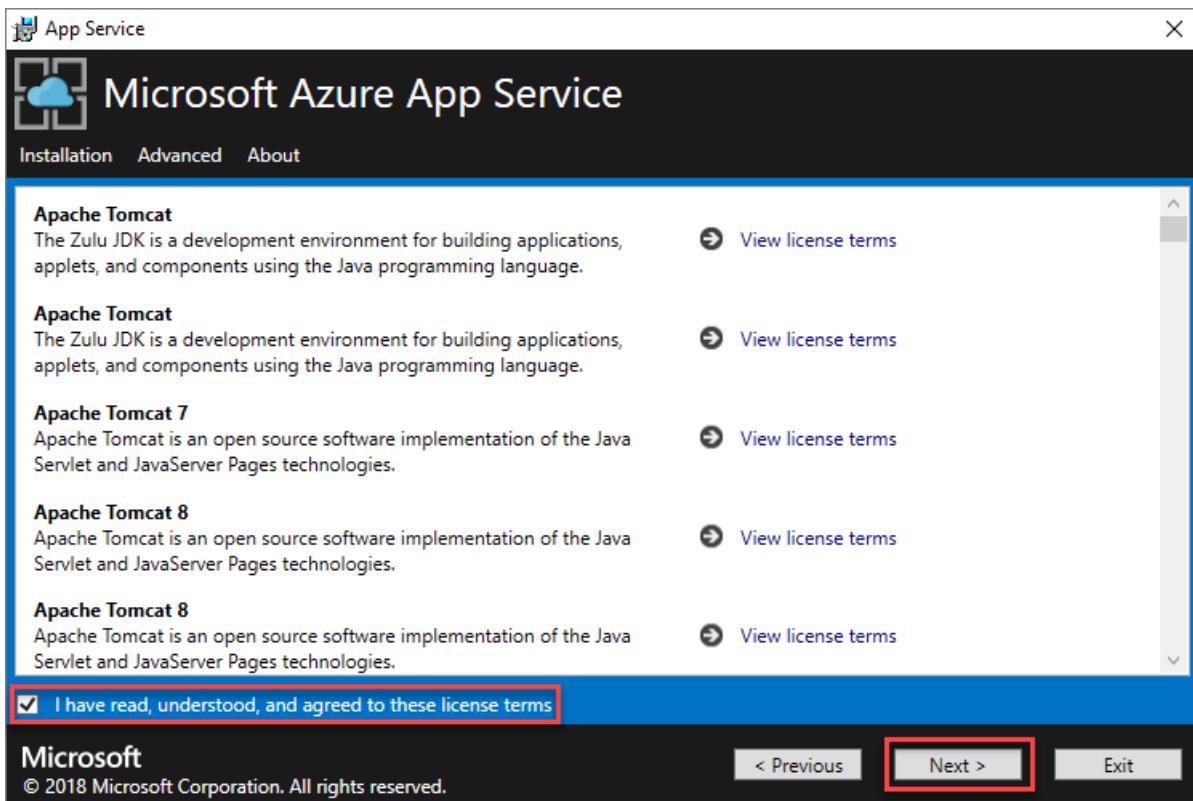
1. Run the previously downloaded App Service installer (**appservice.exe**) as an admin from a computer that can access the Azure Stack Hub Admin Azure Resource Management Endpoint.
2. Select **Deploy App Service or upgrade to the latest version.**



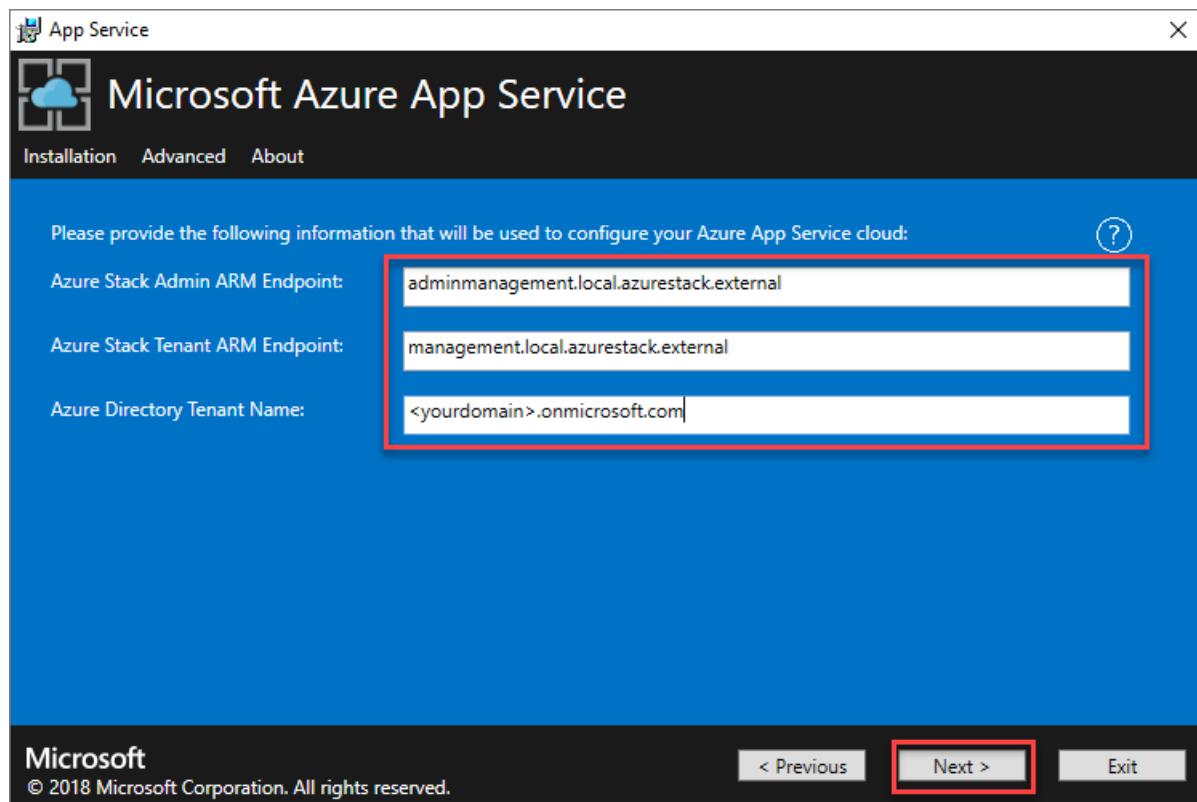
3. Accept Microsoft licensing terms and click **Next**.



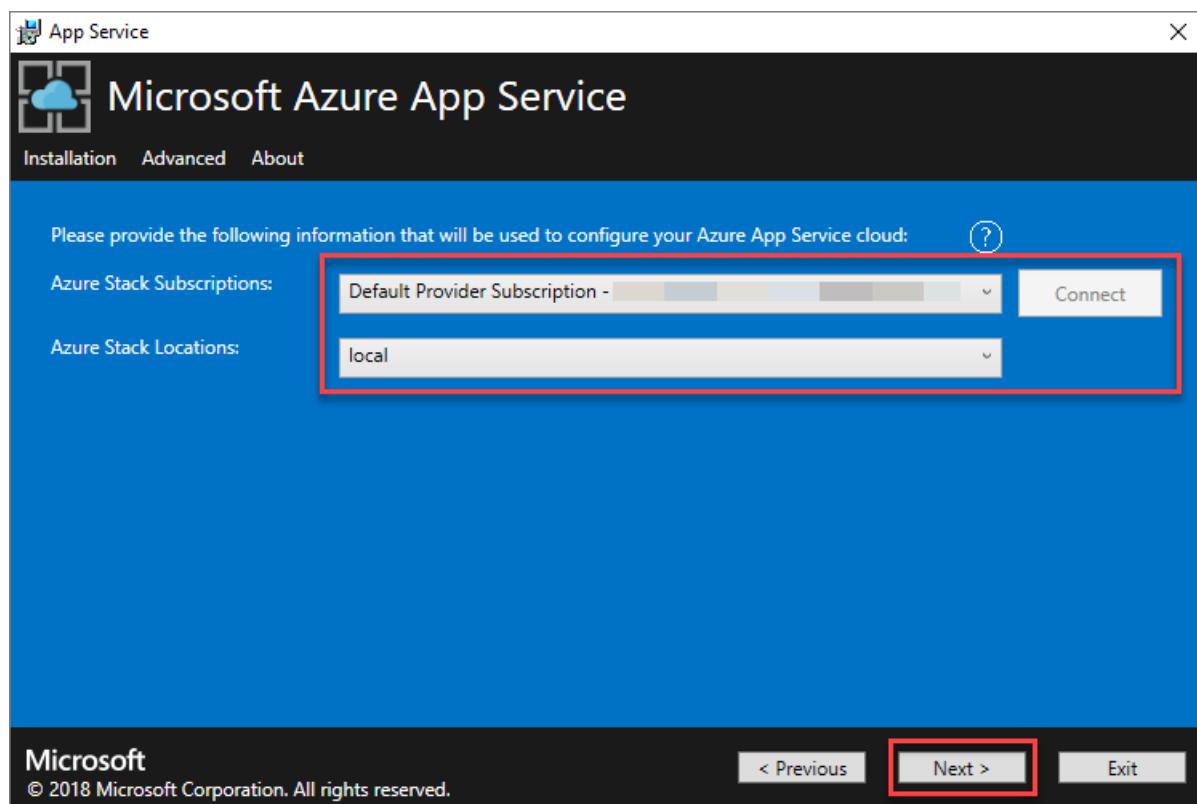
- Accept non-Microsoft licensing terms and click **Next**.



- Provide the App Service cloud endpoint configuration for your Azure Stack Hub environment.

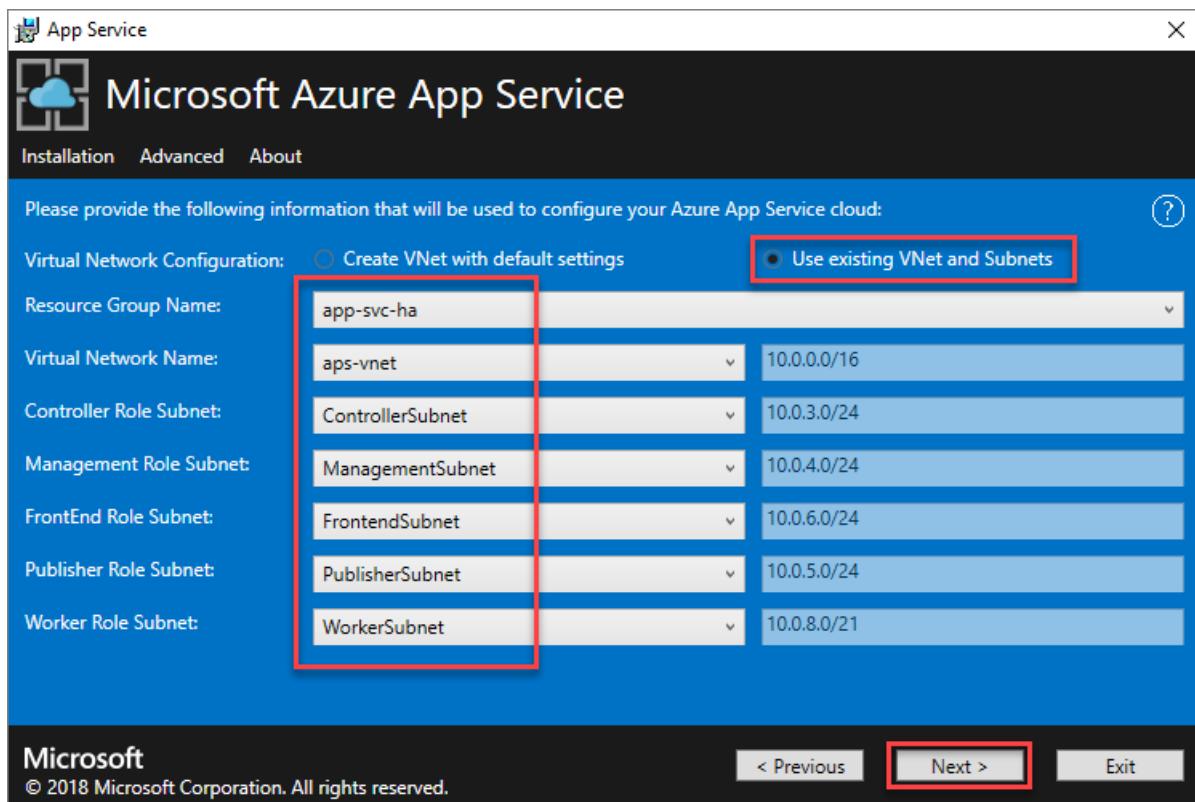


6. Connect to the Azure Stack Hub subscription to be used for the installation and choose the location.

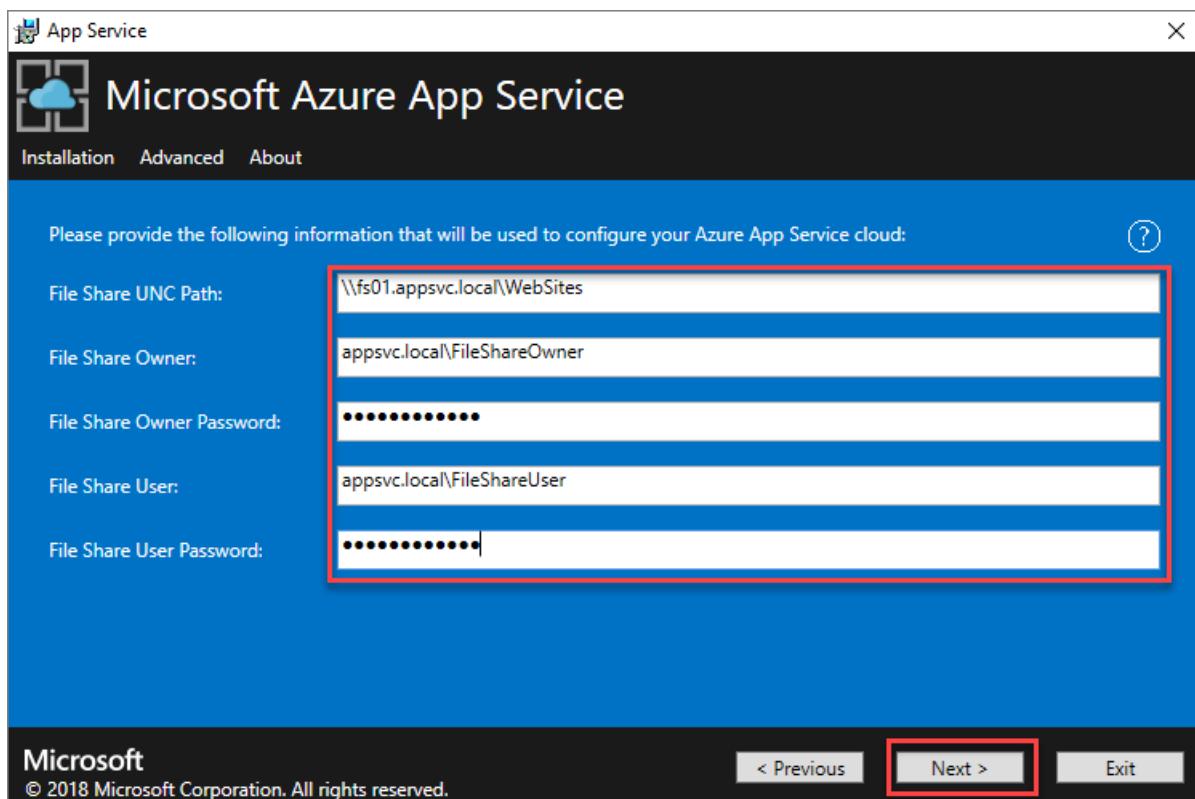


7. Select **Use existing VNet and Subnets** and the **Resource Group Name** for the resource group used to deploy the highly available template.

Next, select the virtual network created as part of the template deployment and then select the appropriate role subnets from the drop-down list options.

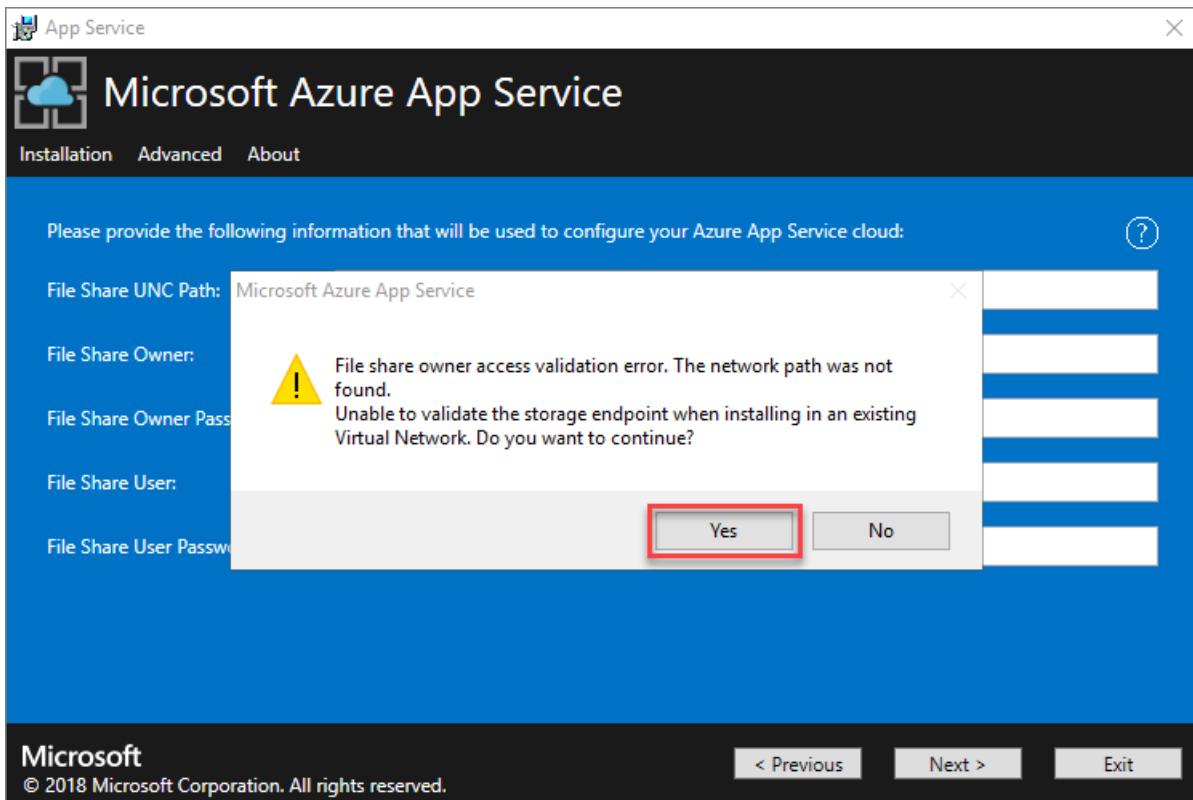


8. Provide the previously recorded template outputs info for the file share path and file share owner parameters. When finished, click **Next**.



9. Because the machine used to install App Service isn't located on the same VNet as the file server used to host the App Service file share, you can't resolve the name. **This error is expected behavior.**

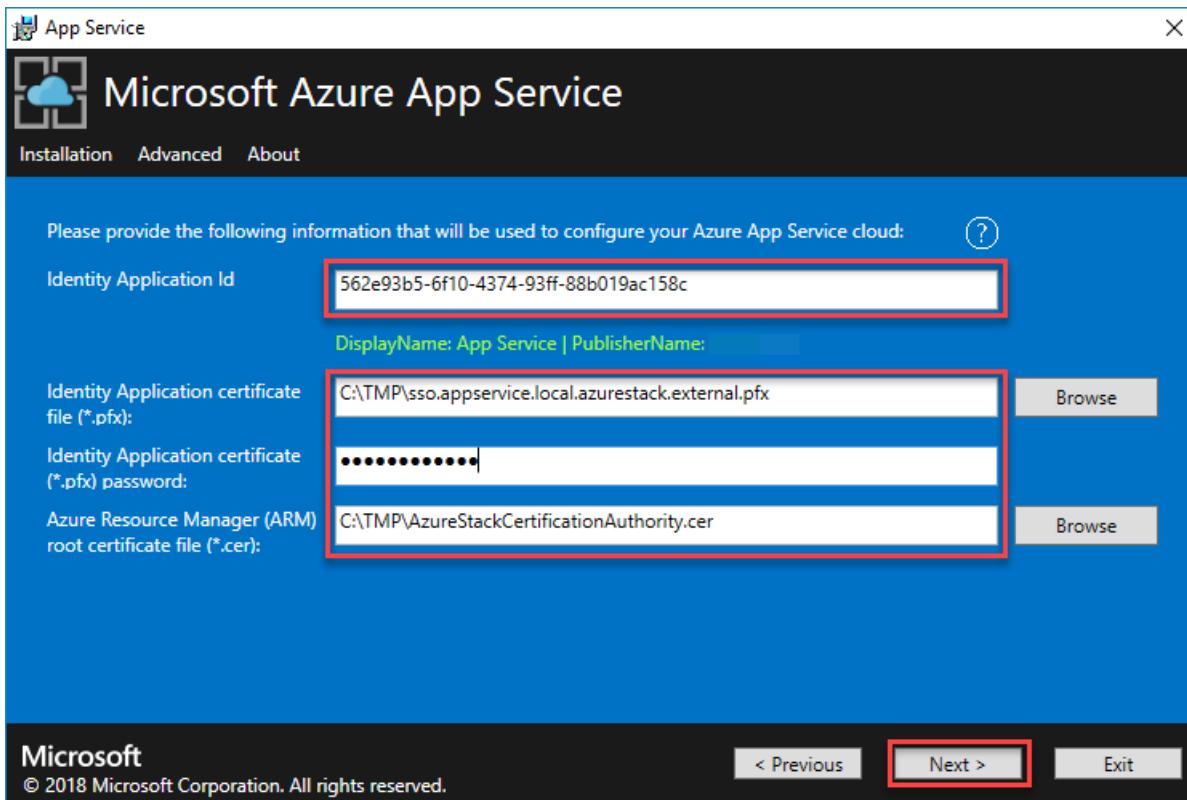
Verify that the info entered for the file share UNC path and accounts info is correct. Then press **Yes** on the alert dialog to continue App Service installation.



If you chose to deploy into an existing virtual network and an internal IP address to connect to your file server, you must add an outbound security rule. This rule enables SMB traffic between the worker subnet and the file server. Go to the WorkersNsg in the administrator portal and add an outbound security rule with the following properties:

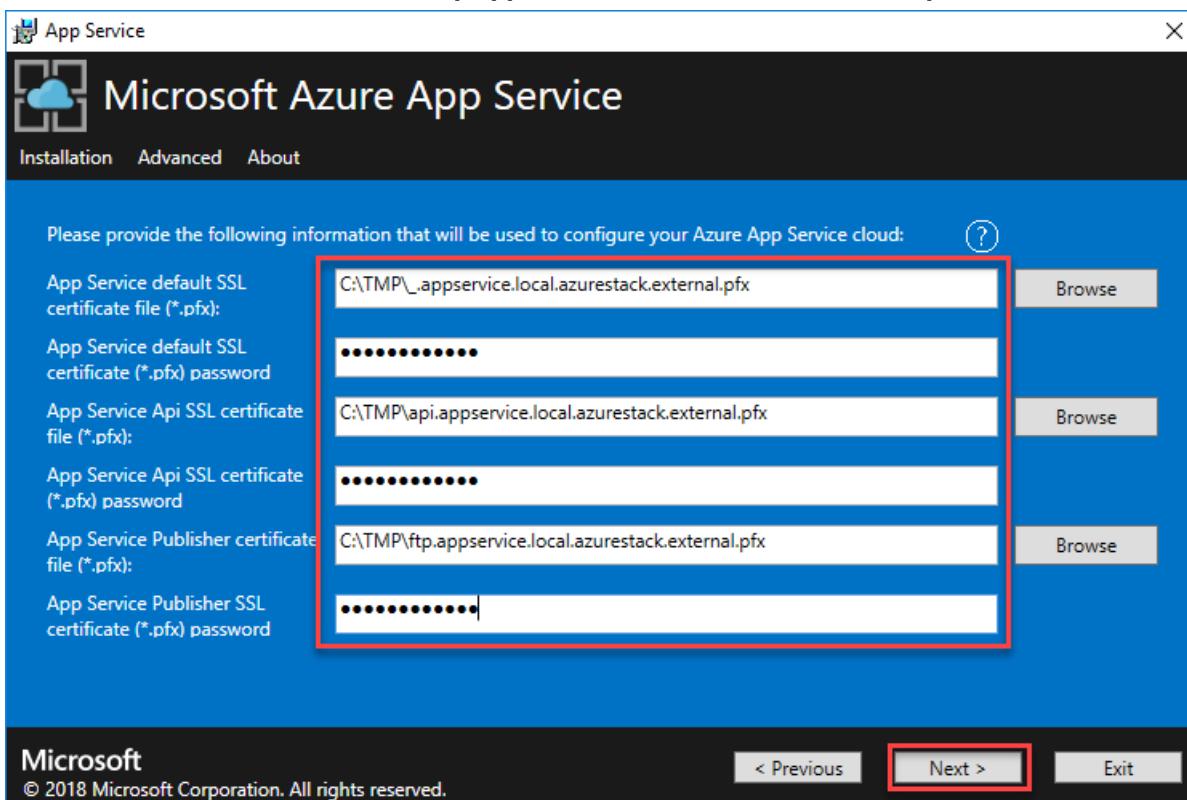
- Source: Any
- Source port range: \*
- Destination: IP Addresses
- Destination IP address range: Range of IPs for your file server
- Destination port range: 445
- Protocol: TCP
- Action: Allow
- Priority: 700
- Name: Outbound\_Allow\_SMB445

10. Provide the Identity Application ID and the path and passwords to the identity certificates and click **Next**:
  - Identity application certificate (in the format of **sso.appservice.local.azurestack.external.pfx**)
  - Azure Resource Manager root certificate (**AzureStackCertificationAuthority.cer**)

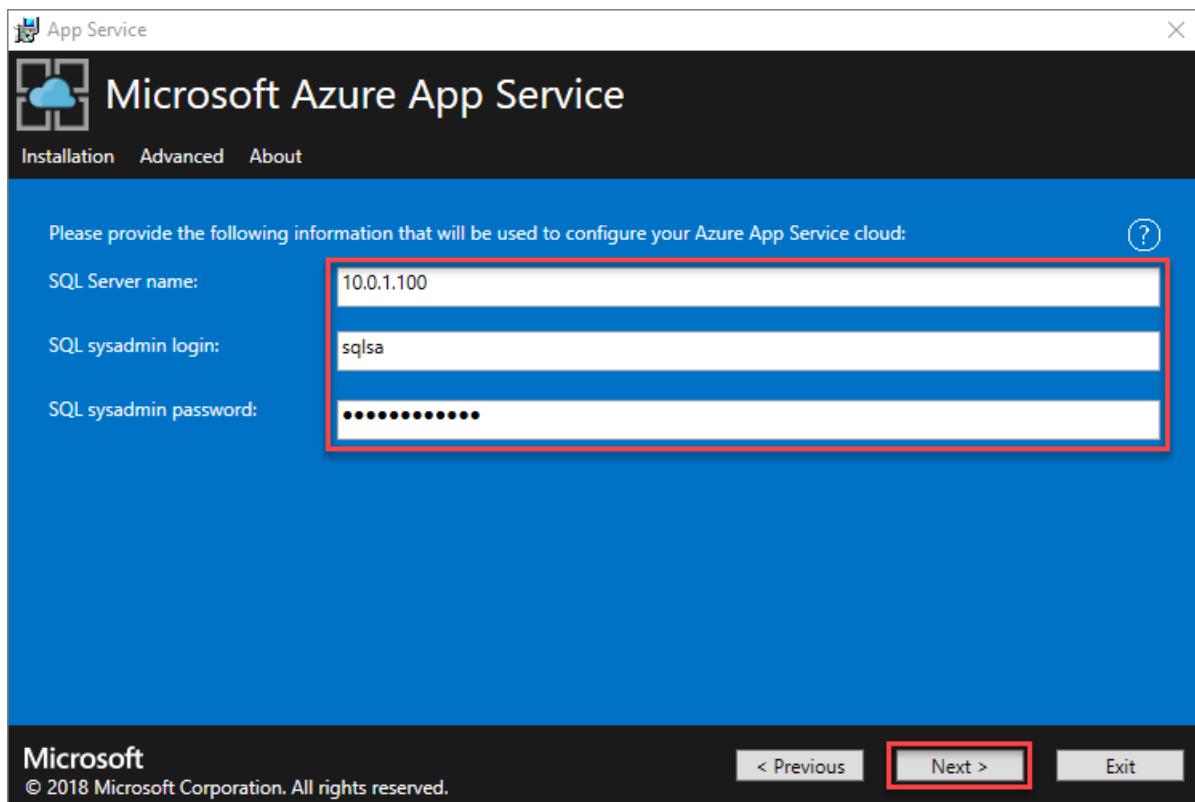


11. Next, provide the remaining required information for the following certificates and click **Next**:

- Default Azure Stack Hub SSL certificate (in the format of **\_appservice.local.azurestack.external.pfx**)
- API SSL certificate (in the format of **api.appservice.local.azurestack.external.pfx**)
- Publisher certificate (in the form of **ftp.appservice.local.azurestack.external.pfx**)

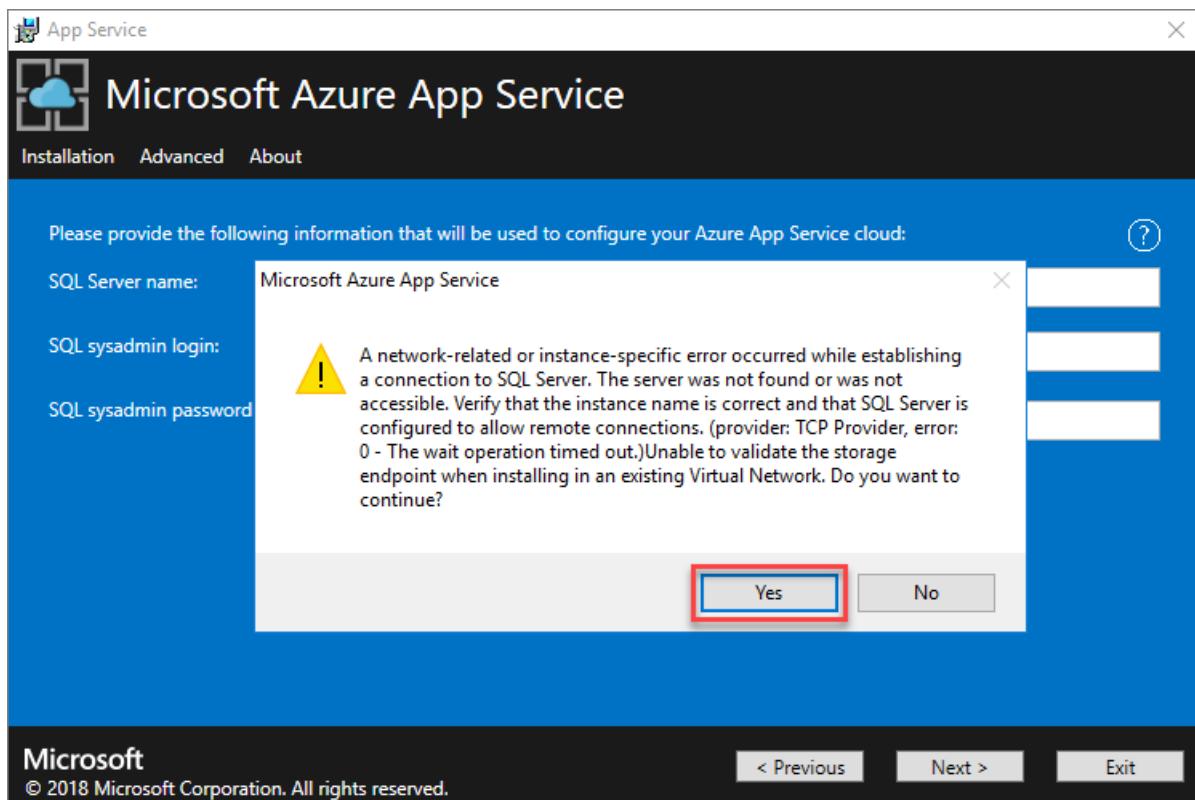


12. Provide the SQL Server connection info using the SQL Server connection info from the high availability template deployment outputs:



13. Because the machine used to install App Service isn't located on the same VNet as the SQL server used to host the App Service databases, you can't resolve the name. **This is expected behavior.**

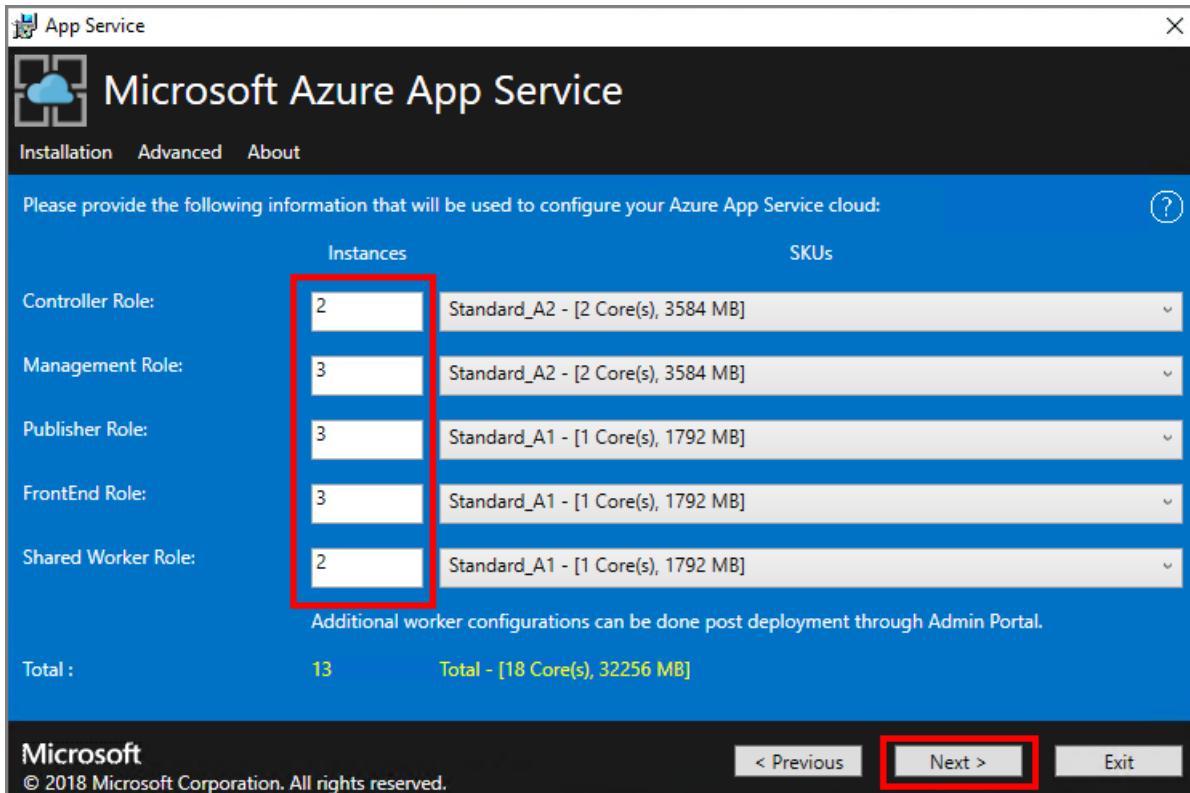
Verify that the info entered for the SQL Server name and accounts info is correct and press **Yes** to continue App Service installation. Click **Next**.



14. Accept the default role configuration values or change to the recommended values and click **Next**.

We recommend that the default values for the App Service infrastructure role instances be changed as follows for highly available configurations:

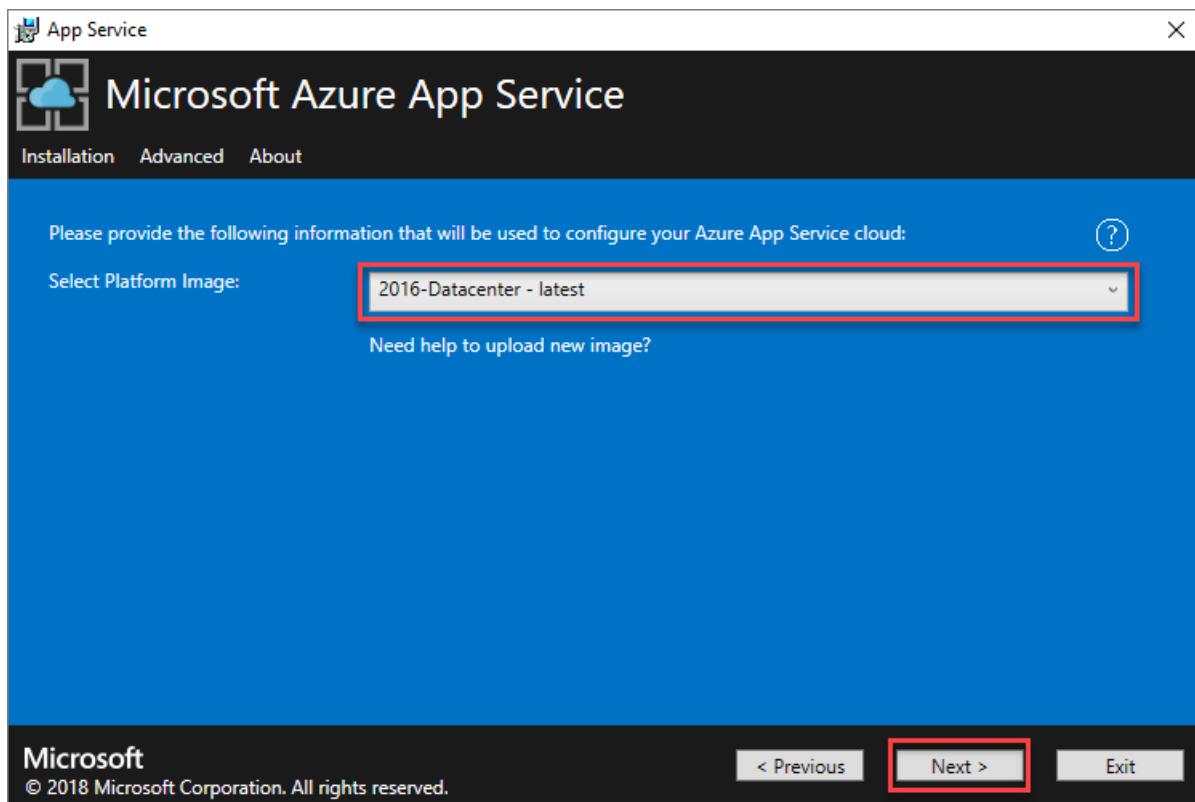
| ROLE               | DEFAULT | HIGHLY AVAILABLE RECOMMENDATION |
|--------------------|---------|---------------------------------|
| Controller Role    | 2       | 2                               |
| Management Role    | 1       | 3                               |
| Publisher Role     | 1       | 3                               |
| FrontEnd Role      | 1       | 3                               |
| Shared Worker Role | 1       | 2                               |



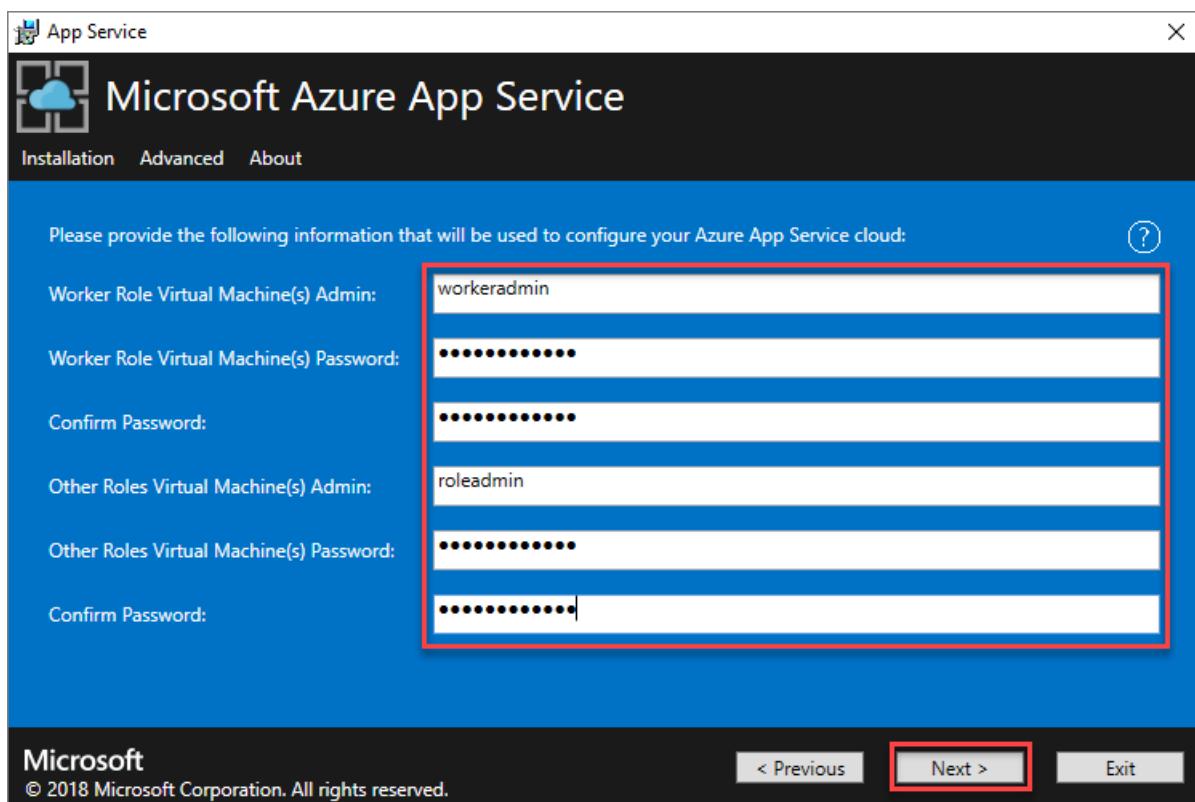
#### NOTE

Changing from the default values to those recommended in this tutorial increases the hardware requirements for installing App Service. A total of 18 cores and 32,256 MB of RAM is needed to support the recommended 13 VMs instead of the default 9 cores and 16,128 MB of RAM for 6 VMs.

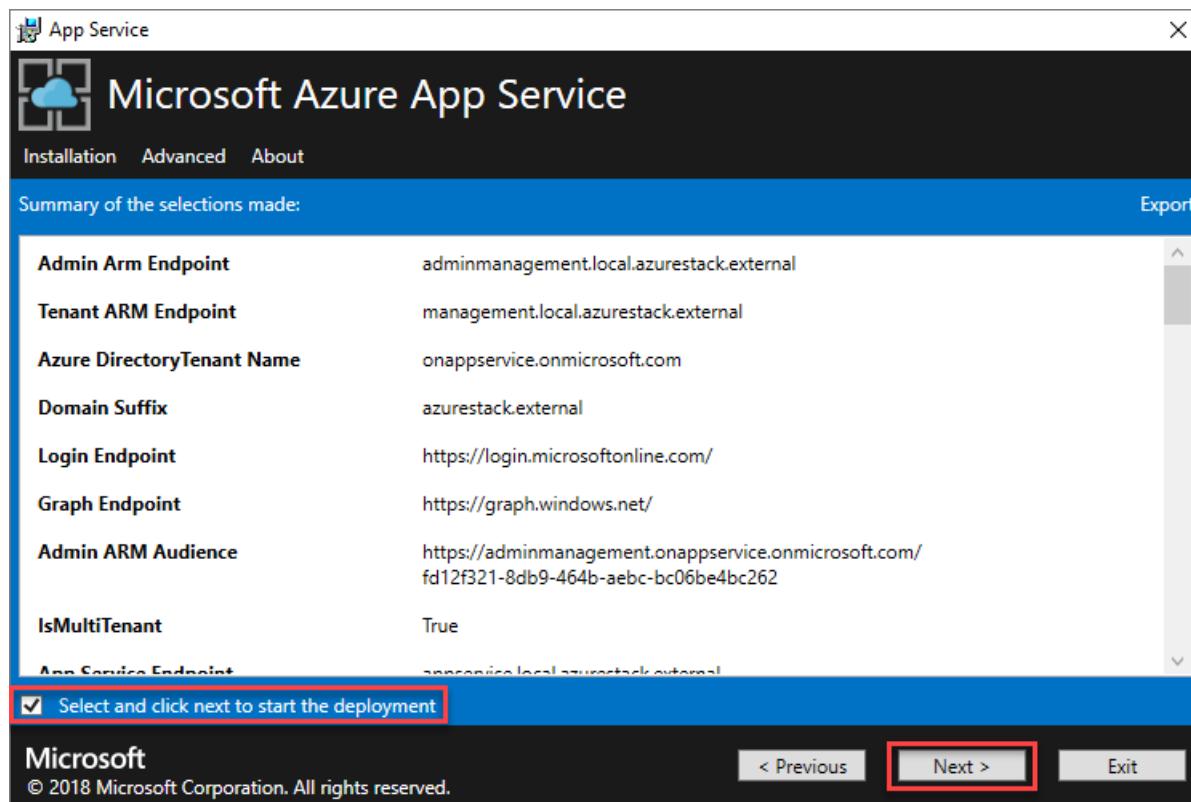
15. Select the platform image to use for installing the App Service infrastructure VMs and click **Next**:



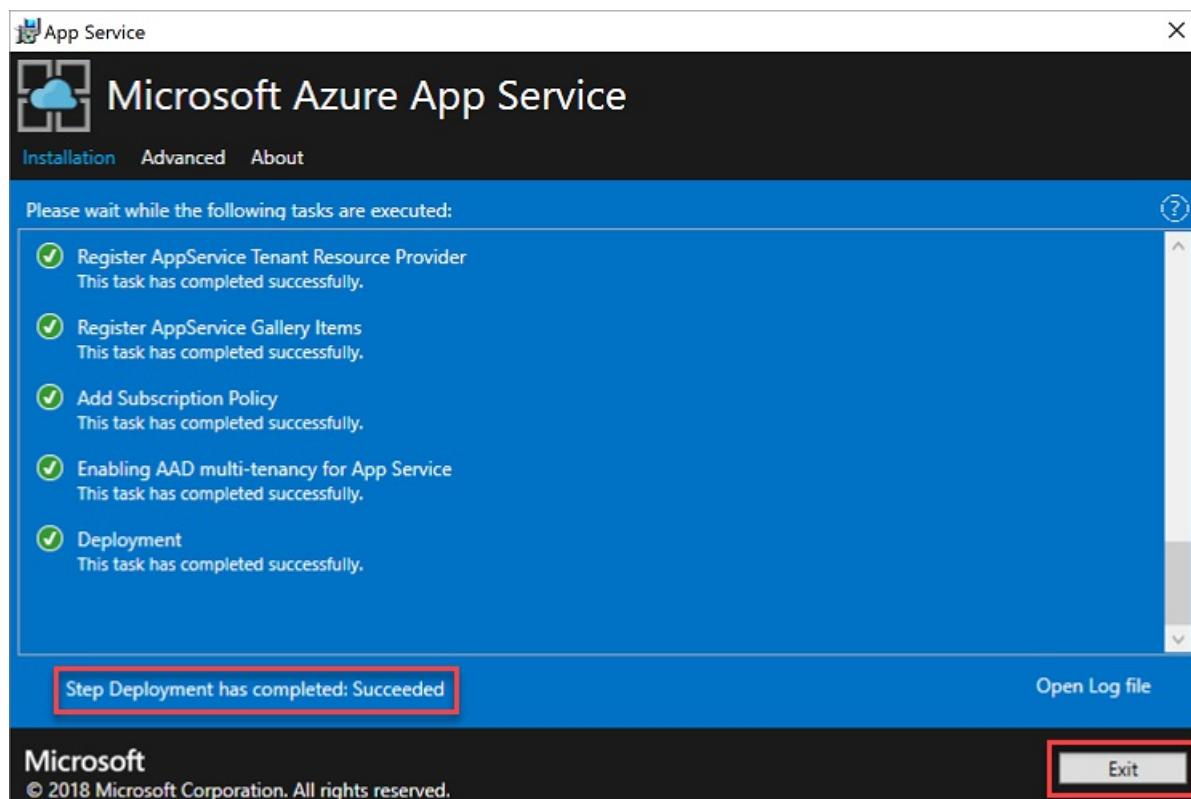
16. Provide App Service infrastructure role credential info to be used and click **Next**:



17. Review the info to be used to deploy App Service and click **Next** to begin deployment.



18. Review the App Service deployment progress. This deployment can take over an hour depending on your specific deployment configuration and hardware. After the installer successfully finishes, select **Exit**.



## Next steps

Add the `appservice_hosting` and `appservice_metering` databases to an availability group if you've provided the App Service resource provider with a SQL Always On Instance. Synchronize the databases to prevent any loss of service in the event of a database failover. You can also run a [script](#) to import the AppServices logins from the original primary server to a failover server.

[Scale out App Service](#). You might need to add additional App Service infrastructure role workers to meet expected

app demand in your environment. By default, App Service on Azure Stack Hub supports free and shared worker tiers. To add other worker tiers, you need to add more worker roles.

[Configure deployment sources](#). Additional configuration is required to support on-demand deployment from multiple source control providers like GitHub, BitBucket, OneDrive, and DropBox.

[Back up App Service](#). After successfully deploying and configuring App Service, you should ensure all components necessary for disaster recovery are backed up. Backing up your essential components helps prevent data loss and unnecessary service downtime during recovery operations.

# Deploy Azure App Service in an offline environment in Azure Stack Hub

11 minutes to read • [Edit Online](#)

## IMPORTANT

Apply the 1910 update to your Azure Stack Hub integrated system or deploy the latest Azure Stack Development Kit (ASDK) before you deploy Azure App Service 1.8.

By following the instructions in this article, you can deploy the [Azure App Service resource provider](#) to an Azure Stack Hub environment that is:

- Not connected to the internet.
- Secured by Active Directory Federation Services (AD FS).

## IMPORTANT

Before you run the resource provider installer, make sure you've completed the steps in [Prerequisites for deploying Azure App Service on Azure Stack Hub](#). You should also read the [release notes](#) that accompany the 1.8 release to learn about new functionality, fixes, and any known issues that could affect your deployment.

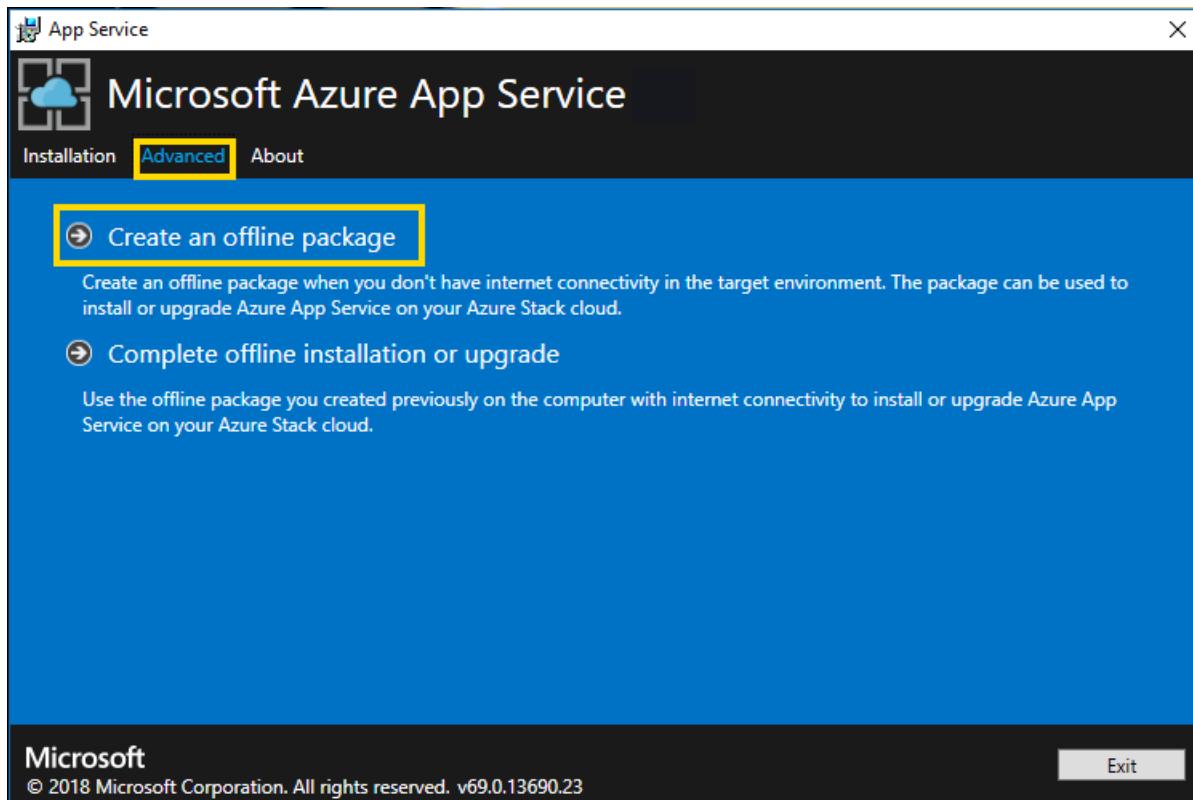
To add the Azure App Service resource provider to your offline Azure Stack Hub deployment, you must complete these top-level tasks:

1. Complete the [prerequisite steps](#) (like purchasing certificates, which can take a few days to receive).
2. [Download and extract the installation and helper files](#) to a machine that's connected to the internet.
3. Create an offline installation package.
4. Run the appservice.exe installer file.

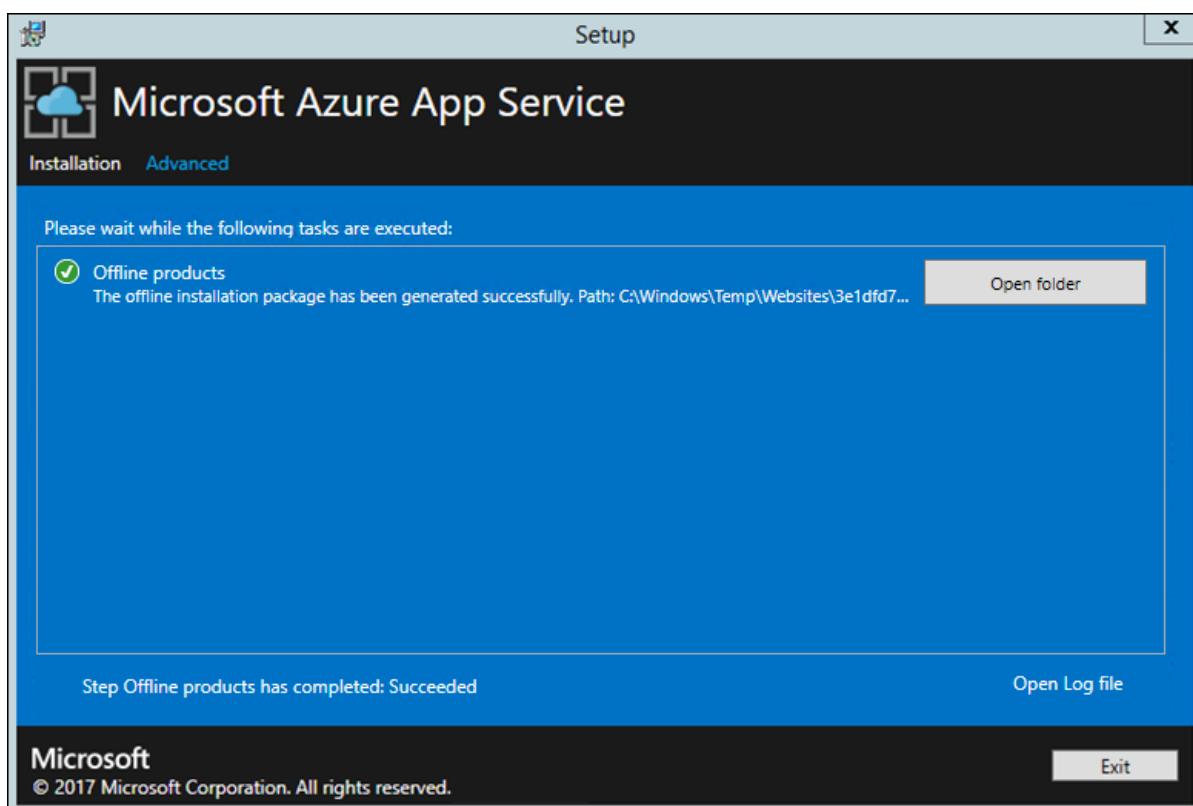
## Create an offline installation package

To deploy Azure App Service in an offline environment, first create an offline installation package on a machine that's connected to the internet.

1. Run the AppService.exe installer on a machine that's connected to the internet.
2. Select **Advanced** > **Create offline installation package**. This step will take several minutes to complete.



3. The Azure App Service installer creates an offline installation package and displays the path to it. You can select **Open folder** to open the folder in File Explorer.

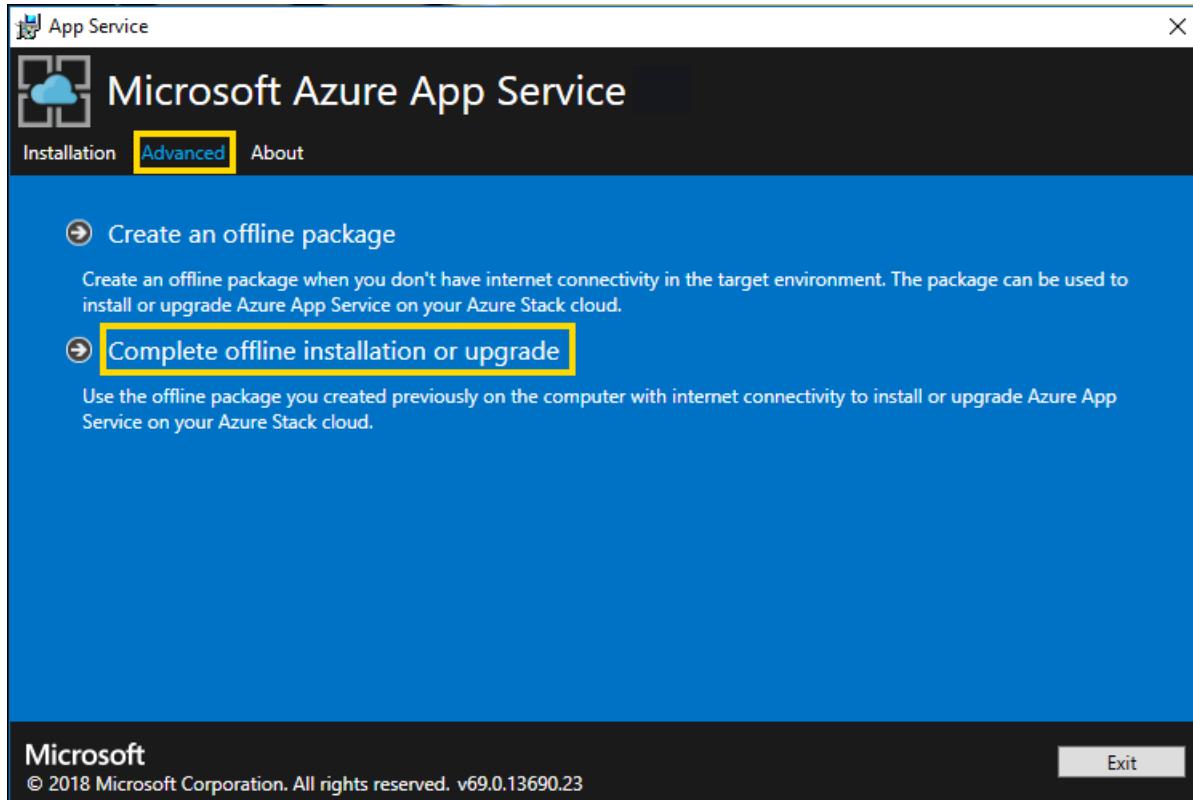


4. Copy the installer (AppService.exe) and the offline installation package to your Azure Stack Hub host machine.

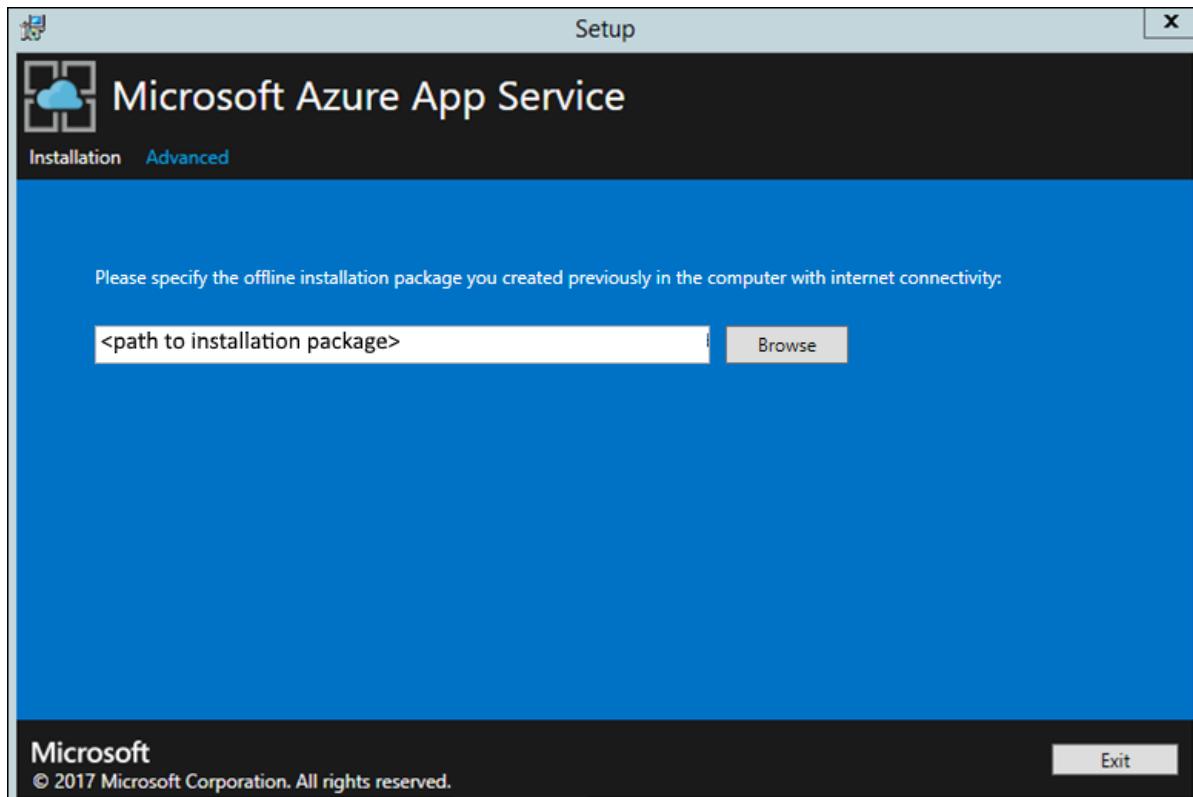
## Complete the offline installation of Azure App Service on Azure Stack Hub

1. Run appservice.exe as an admin from a computer that can reach the Azure Stack Hub Admin Azure Resource Management endpoint.

2. Select **Advanced > Complete offline installation.**

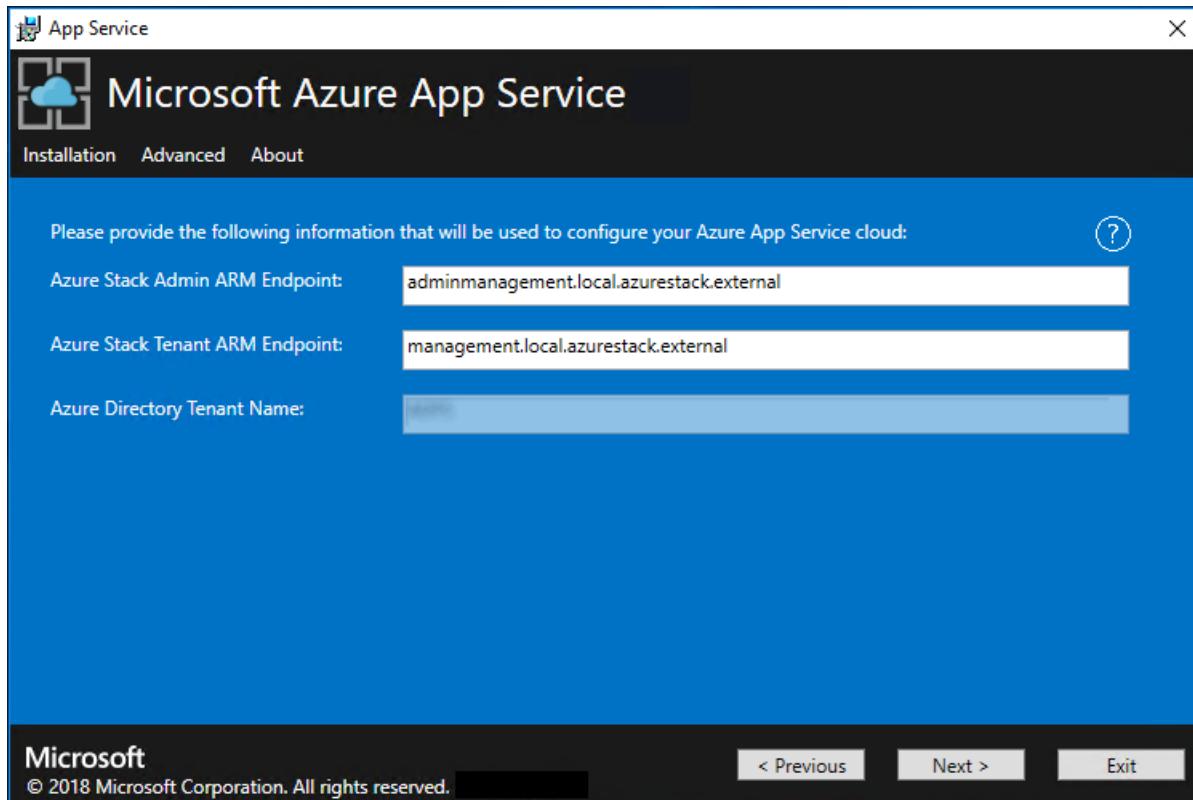


3. Browse to the location of the offline installation package you previously created, and then select **Next**.

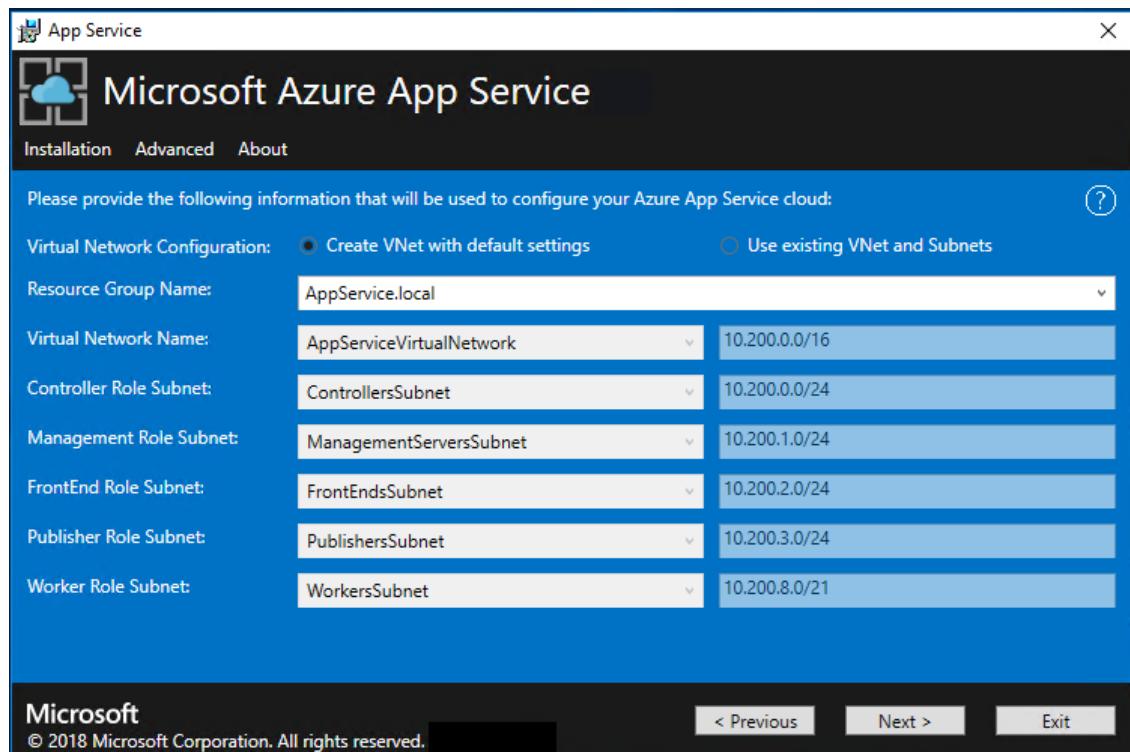


4. Review and accept the Microsoft Software License Terms, and then select **Next**.
5. Review and accept the third-party license terms, and then select **Next**.
6. Make sure the Azure App Service cloud configuration info is correct. If you used the default settings during ASDK deployment, you can accept the default values here. However, if you customized the options when you deployed Azure Stack Hub or are deploying on an integrated system, you must edit the values in this window to reflect those changes. For example, if you use the domain suffix mycloud.com, your Azure Stack

Hub Tenant Azure Resource Manager endpoint must change to `management.<region>.mycloud.com`. After you confirm your info, select **Next**.



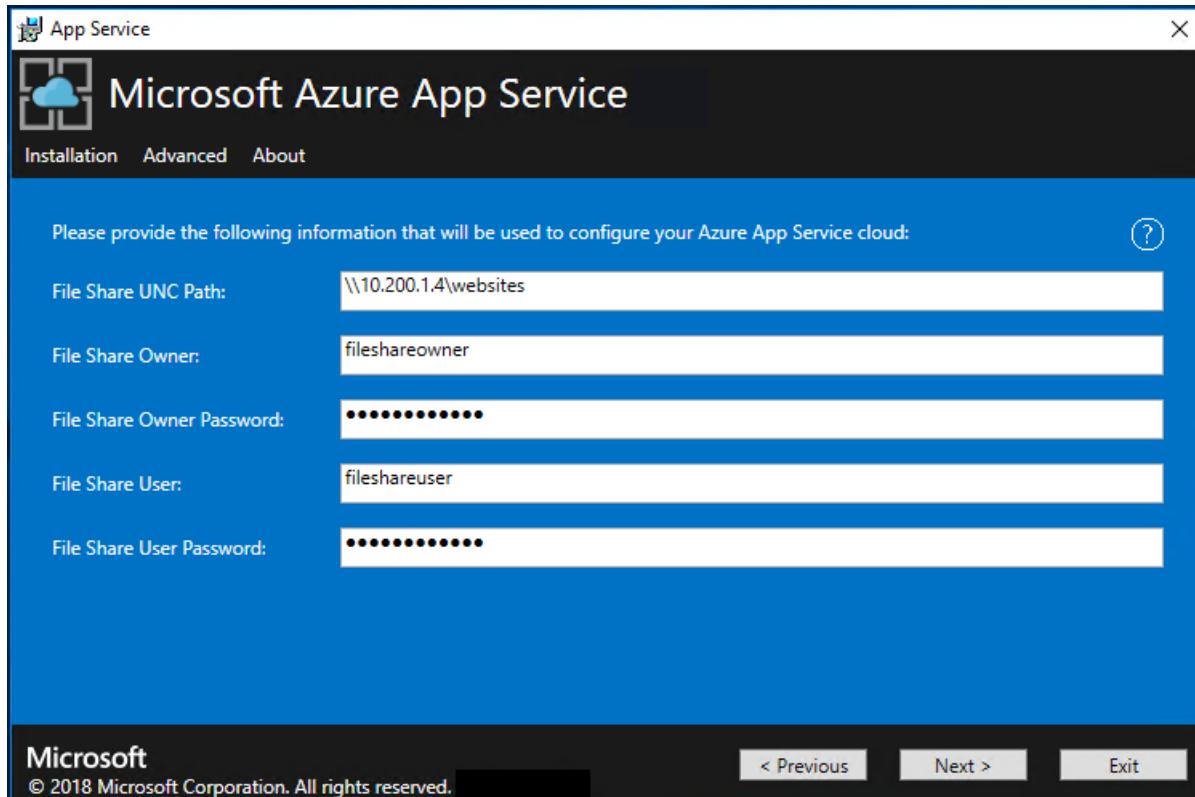
7. On the next App Service Installer page you will connect to your Azure Stack Hub:
  - a. Select the connection method you wish to use - **Credential** or **Service Principal**
    - **Credential**
      - If you're using Azure Active Directory (Azure AD), enter the Azure AD admin account and password that you provided when you deployed Azure Stack Hub. Select **Connect**.
      - If you're using Active Directory Federation Services (AD FS), provide your admin account. For example, clouddadmin@azurestack.local. Enter your password, and then select **Connect**.
    - **Service Principal**
      - The service principal which you use **must** have **Owner** rights on the **Default Provider Subscription**
      - Provide the **Service Principal ID**, **Certificate File** and **Password** and select **Connect**.
  - b. In **Azure Stack Hub Subscriptions**, select the **Default Provider Subscription**. Azure App Service on Azure Stack Hub **must** be deployed in the **Default Provider Subscription**.
  - c. In the **Azure Stack Hub Locations**, select the location that corresponds to the region you're deploying to. For example, select **local** if you're deploying to the ASDK.
8. You can allow the Azure App Service installer to create a virtual network and associated subnets. Or, you can deploy into an existing virtual network, as configured through [these steps](#).
  - To use the Azure App Service installer method, select **Create VNet with default settings**, accept the defaults, and then select **Next**.
  - To deploy into an existing network, select **Use existing VNet and Subnets**, and then:
    - a. Select the **Resource Group** option that contains your virtual network.
    - b. Choose the **Virtual Network** name you want to deploy into.
    - c. Select the correct **Subnet** values for each of the required role subnets.
    - d. Select **Next**.



9. Enter the info for your file share and then select **Next**. The address of the file share must use the Fully Qualified Domain Name (FQDN) or IP address of your file server. For example:  
\appservicefileserver.local.cloudapp.azurestack.external\websites, or \\10.0.0.1\websites. If you're using a file server that's domain-joined, you must provide the full user name, including the domain. For example:  
<myfileserverdomain>\<FileShareOwner> .

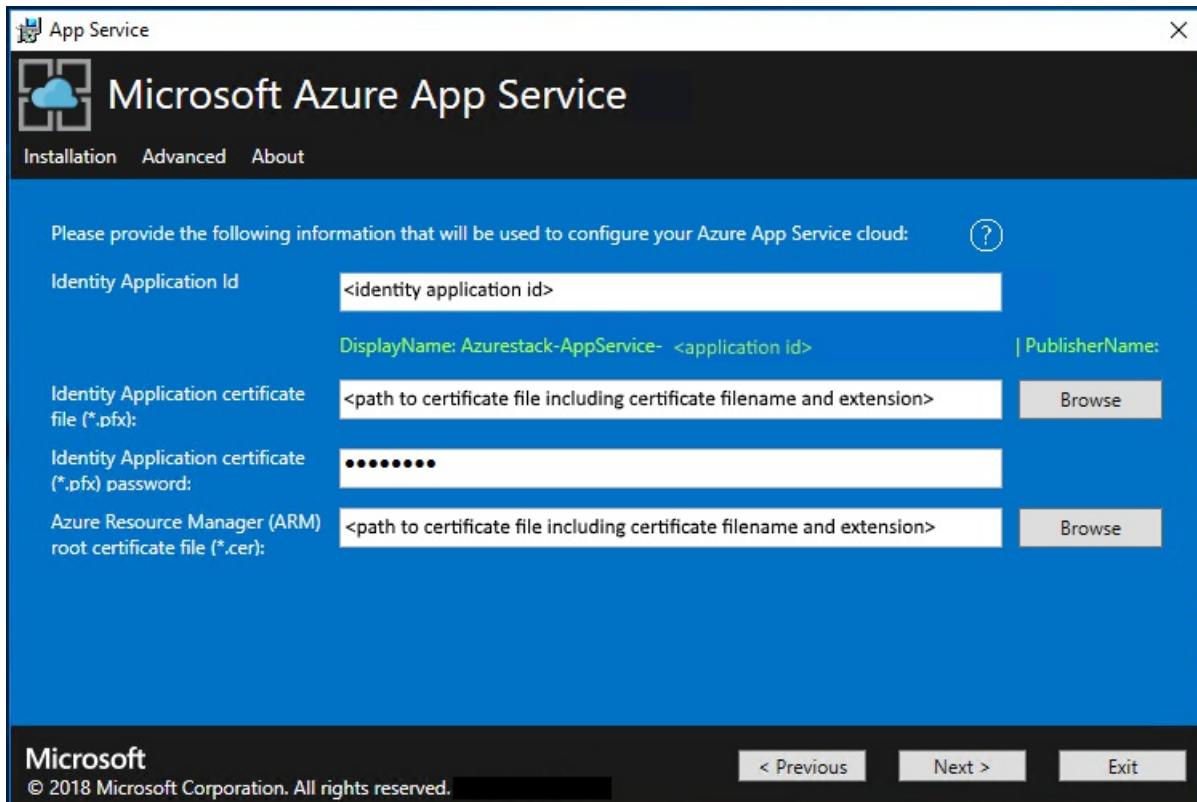
#### NOTE

The installer tries to test connectivity to the file share before proceeding. However, if you choose to deploy into an existing virtual network, the installer might be unable to connect to the file share and displays a warning asking whether you want to continue. Verify the file share info and continue if it's correct.



10. On the next page:

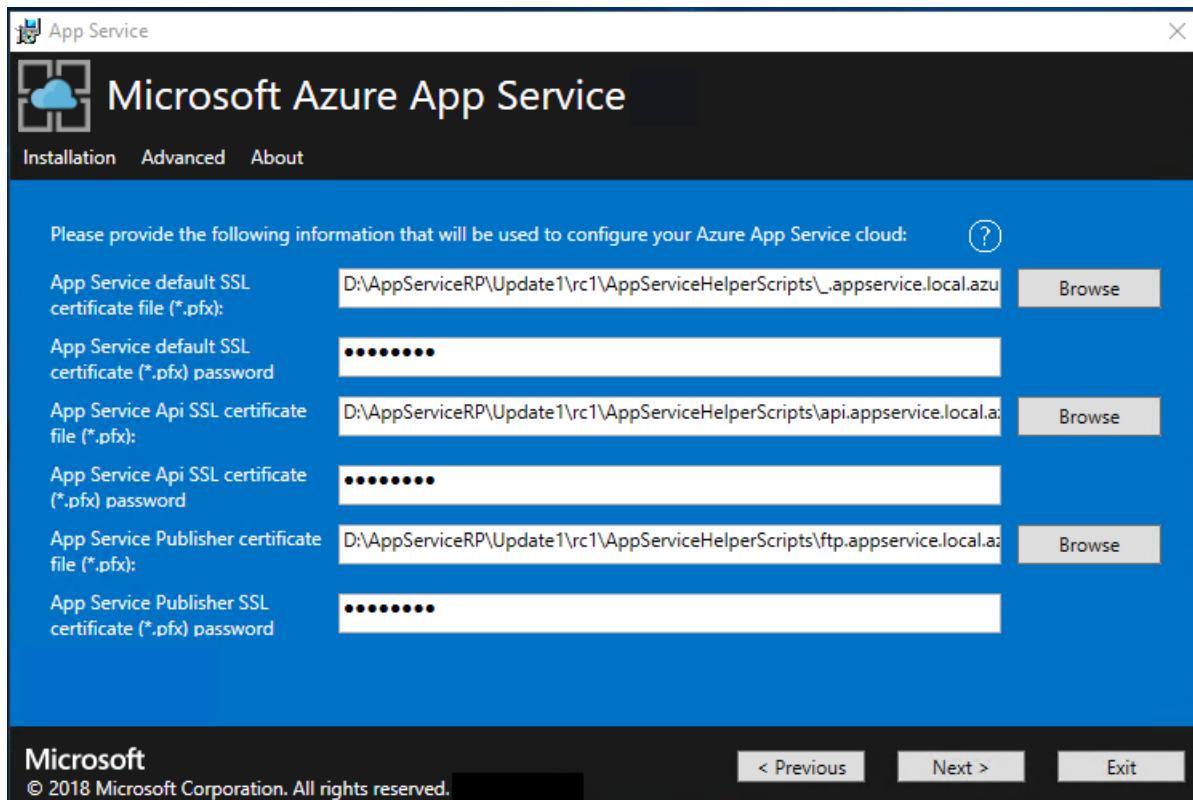
- a. In the **Identity Application ID** box, enter the GUID for the app you're using for identity (from Azure AD).
- b. In the **Identity Application certificate file** box, enter (or browse to) the location of the certificate file.
- c. In the **Identity Application certificate password** box, enter the password for the certificate. This password is the one that you made note of when you used the script to create the certificates.
- d. In the **Azure Resource Manager root certificate file** box, enter (or browse to) the location of the certificate file.
- e. Select **Next**.



11. For each of the three certificate file boxes, select **Browse**, and then go to the appropriate certificate file. You must provide the password for each certificate. These certificates are the ones that you created in the [Create required certificates step](#). Select **Next** after entering all the info.

| BOX                                               | CERTIFICATE FILE NAME EXAMPLE                |
|---------------------------------------------------|----------------------------------------------|
| <b>App Service default SSL certificate file</b>   | _appservice.local.AzureStack.external.pfx    |
| <b>App Service API SSL certificate file</b>       | api.appservice.local.AzureStack.external.pfx |
| <b>App Service Publisher SSL certificate file</b> | ftp.appservice.local.AzureStack.external.pfx |

If you used a different domain suffix when you created the certificates, your certificate file names don't use *local.AzureStack.external*. Instead, use your custom domain info.



12. Enter the SQL Server details for the server instance used to host the Azure App Service resource provider databases, and then select **Next**. The installer validates the SQL connection properties. You **must** enter either the internal IP or the FQDN for the SQL Server name.

#### NOTE

The installer tries to test connectivity to the computer running SQL Server before proceeding. However, if you chose to deploy into an existing virtual network, the installer might not be able to connect to the computer running SQL Server and displays a warning asking whether you want to continue. Verify the SQL Server info and continue if it's correct.

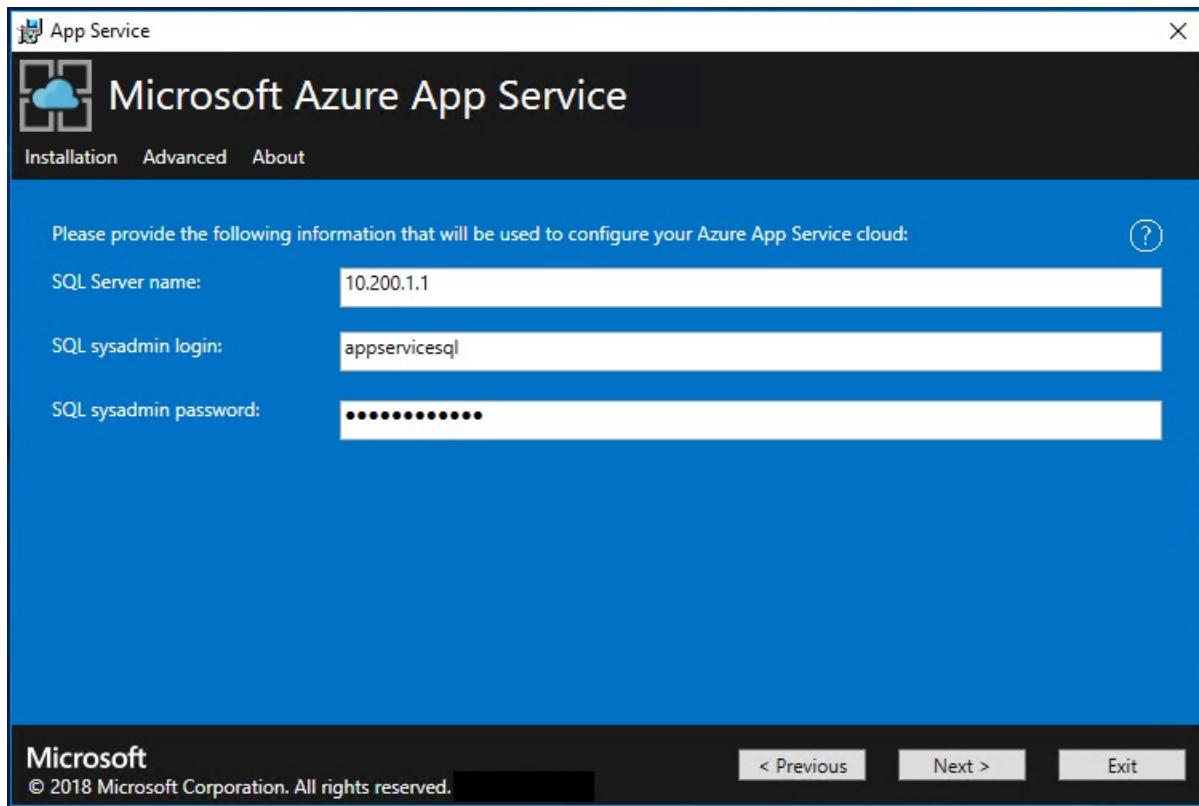
From Azure App Service on Azure Stack Hub 1.3 onward, the installer checks that the computer running SQL Server has database containment enabled at the SQL Server level. If it doesn't, you're prompted with the following exception:

```
Enable contained database authentication for SQL server by running below command on SQL server
(Ctrl+C to copy)

sp_configure 'contained database authentication', 1;
GO
RECONFIGURE;
GO

```

For more details, see the [release notes for Azure App Service on Azure Stack Hub 1.3](#).



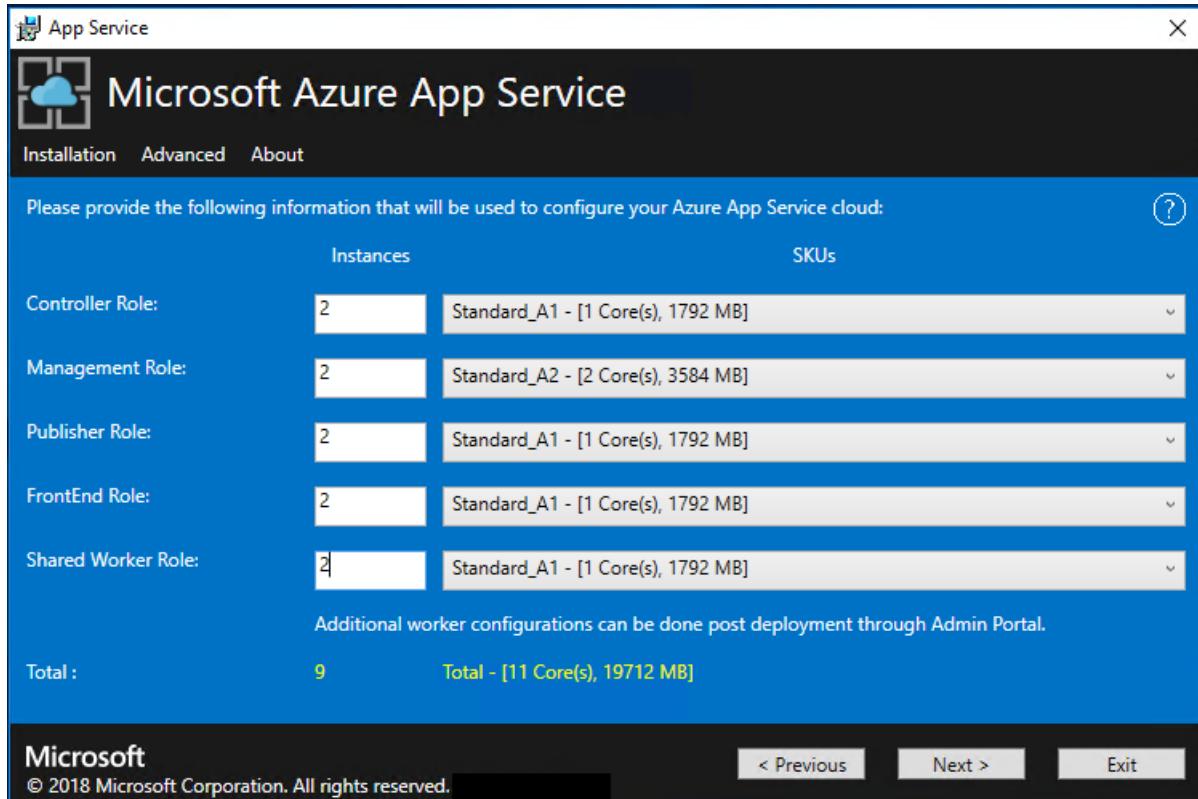
13. Review the role instance and SKU options. The defaults populate with the minimum number of instances and the minimum SKU for each role in an ASDK deployment. A summary of vCPU and memory requirements is provided to help plan your deployment. After you make your selections, select **Next**.

**NOTE**

For production deployments, follow the guidance in [Capacity planning for Azure App Service server roles in Azure Stack Hub](#).

| ROLE       | MINIMUM INSTANCES | MINIMUM SKU                      | NOTES                                                                                                                                                                                                            |
|------------|-------------------|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Controller | 1                 | Standard_A2 - (2 vCPU, 3584 MB)  | Manages and maintains the health of the Azure App Service cloud.                                                                                                                                                 |
| Management | 1                 | Standard_A2 - (2 vCPUs, 3584 MB) | Manages the Azure App Service Azure Resource Manager and API endpoints, portal extensions (admin, tenant, Functions portal), and the data service. To support failover, increase the recommended instances to 2. |
| Publisher  | 1                 | Standard_A1 - (1 vCPU, 1792 MB)  | Publishes content via FTP and web deployment.                                                                                                                                                                    |
| FrontEnd   | 1                 | Standard_A1 - (1 vCPU, 1792 MB)  | Routes requests to Azure App Service apps.                                                                                                                                                                       |

| ROLE          | MINIMUM INSTANCES | MINIMUM SKU                     | NOTES                                                                                                                                                                                                  |
|---------------|-------------------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Shared Worker | 1                 | Standard_A1 - (1 vCPU, 1792 MB) | Hosts web or API apps and Azure Functions apps. You might want to add more instances. As an operator, you can define your offering and choose any SKU tier. The tiers must have a minimum of one vCPU. |



14. In the **Select Platform Image** box, choose your deployment Windows Server 2016 virtual machine (VM) image from the images available on the compute resource provider for the Azure App Service cloud. Select **Next**.

#### NOTE

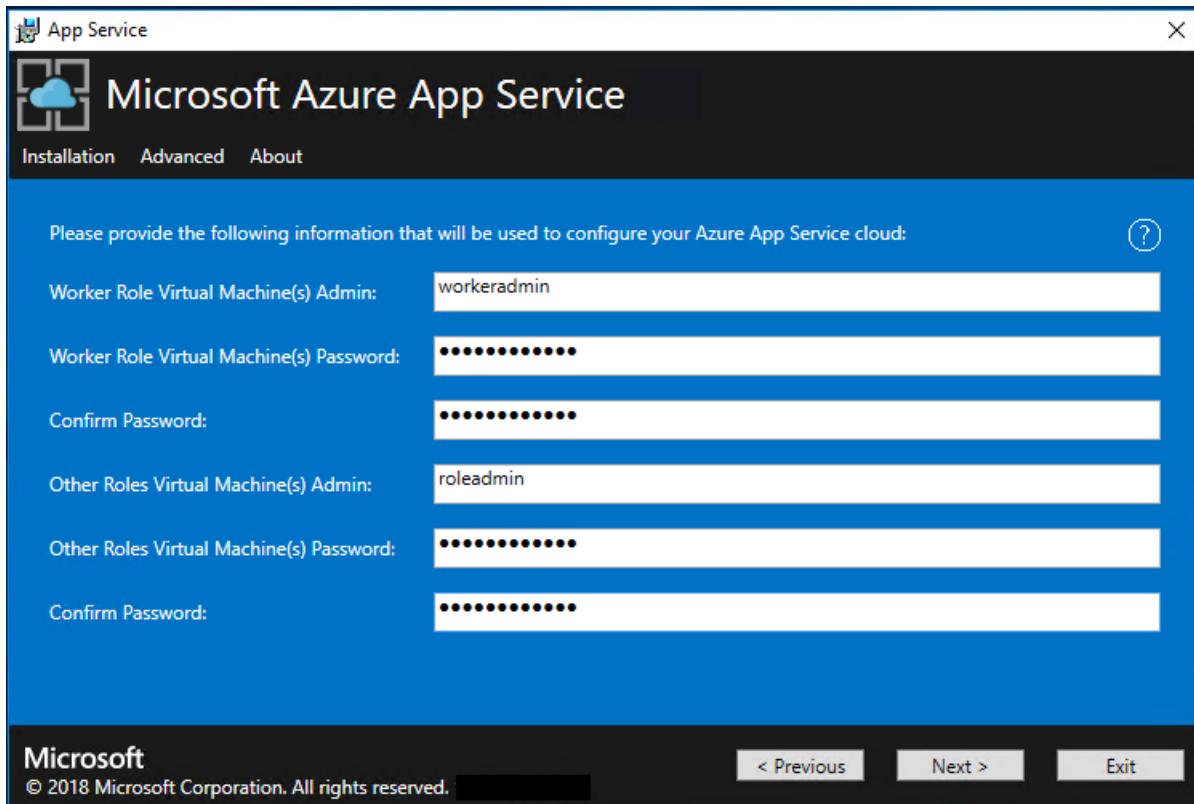
Windows Server 2016 Core is *not* a supported platform image for use with Azure App Service on Azure Stack Hub. Don't use evaluation images for production deployments. Azure App Service on Azure Stack Hub requires that Microsoft .NET 3.5.1 SP1 be activated on the image used for deployment. Marketplace-syndicated Windows Server 2016 images don't have this feature enabled. Therefore, you must create and use a Windows Server 2016 image with this feature pre-enabled.

See [Add a custom VM image to Azure Stack Hub](#) for details on creating a custom image and adding to Marketplace. Be sure to specify the following when adding the image to Marketplace:

- Publisher = MicrosoftWindowsServer
- Offer = WindowsServer
- SKU = 2016-Datacenter
- Version = Specify the "latest" version

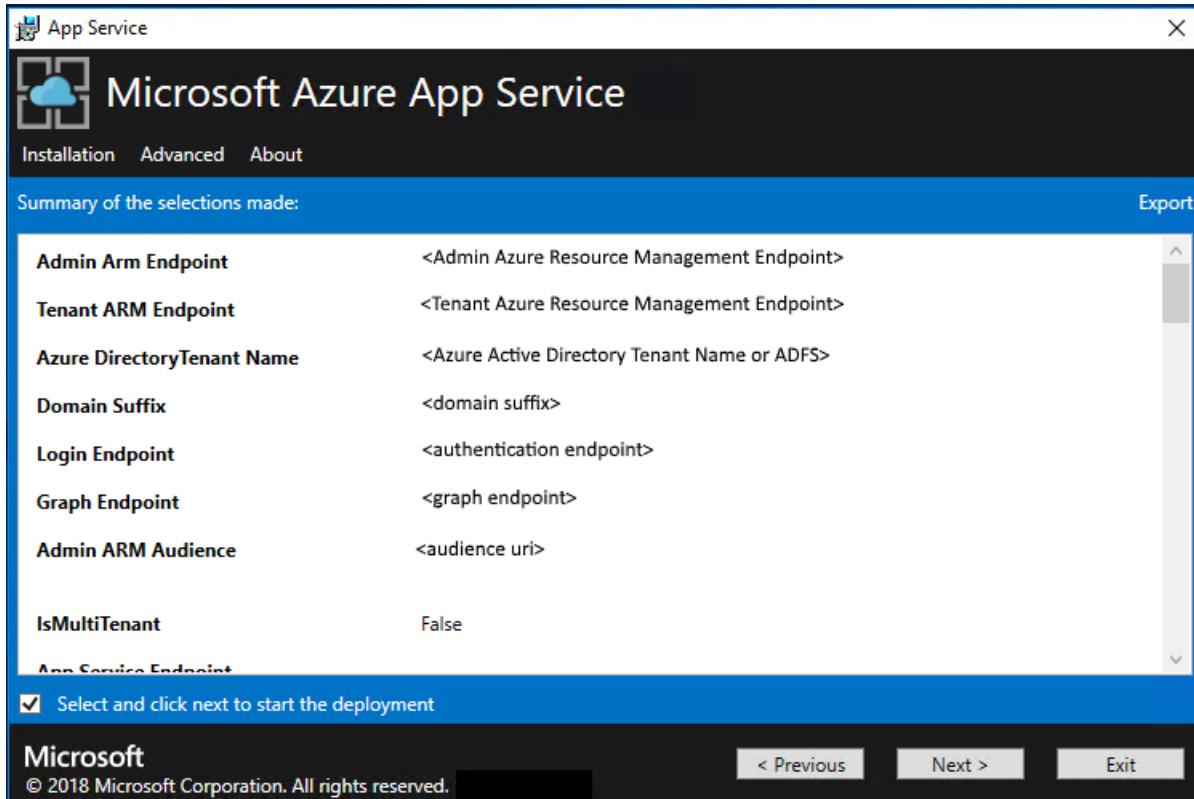
15. On the next page:

- Enter the Worker Role VM admin user name and password.
- Enter the Other Roles VM admin user name and password.
- Select **Next**.



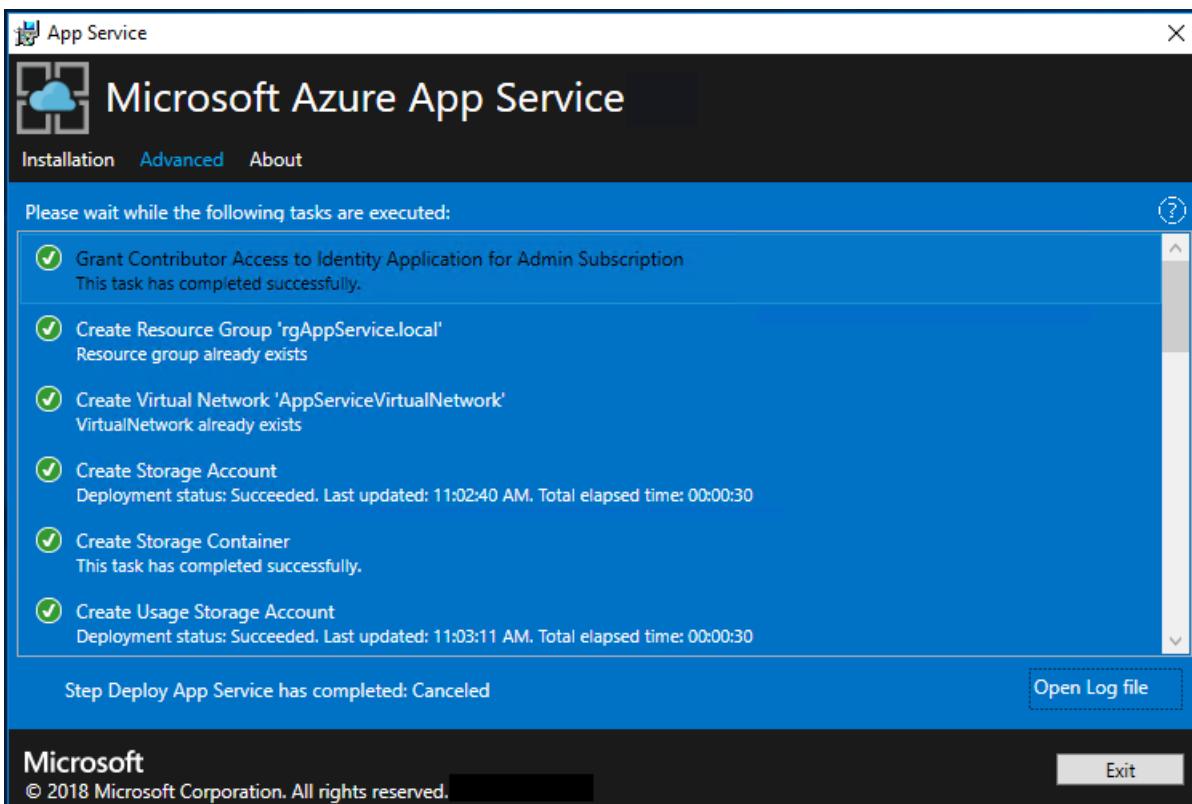
16. On the summary page:

- Verify the selections you made. To make changes, use the **Previous** buttons to visit previous pages.
- If the configurations are correct, select the check box.
- To start the deployment, select **Next**.



17. On the next page:

- a. Track the installation progress. App Service on Azure Stack Hub can take up to 240 minutes to deploy based on the default selections and age of the base Windows 2016 Datacenter image.
- b. After the installer finishes running, select **Exit**.



## Post-deployment steps

### IMPORTANT

If you've provided the Azure App Service RP with a SQL Always On Instance, you *must add the appservice\_hosting and appservice\_metering databases to an availability group*. You must also synchronize the databases to prevent any loss of service in the event of a database failover.

If you chose to deploy into an existing virtual network and an internal IP address to connect to your file server, you must add an outbound security rule, enabling SMB traffic between the worker subnet and the file server. In the administrator portal, go to the WorkersNsg Network Security Group and add an outbound security rule with the following properties:

- Source: Any
- Source port range: \*
- Destination: IP addresses
- Destination IP address range: Range of IPs for your file server
- Destination port range: 445
- Protocol: TCP
- Action: Allow
- Priority: 700
- Name: Outbound\_Allow\_SMB445

## Validate the Azure App Service on Azure Stack Hub installation

1. In the Azure Stack Hub administrator portal, go to **Administration - App Service**.

2. In the overview, under status, check to see that the **Status** displays **All roles are ready**.

The screenshot shows the Microsoft Azure Stack - Administration interface for an App Service named 'local'. The left sidebar has a search bar and links to Overview, Properties, System configuration, Source control configuration, Roles, IP SSL, IP Block List, Worker Tiers, SKUs, Pricing Tiers, Subscription Quotas, and Custom Software. The main area is titled 'Essentials' and shows the following details:

- Resource group: AppService.local
- Status: All roles are ready (highlighted with a red box)
- Location: local
- Subscription ID: <subscription id>

Below the essentials section is a 'System' performance chart with a blue line graph showing CPU usage over time, ranging from 0 to 100. The chart shows a baseline around 40% with some minor fluctuations.

## Test drive Azure App Service on Azure Stack Hub

After you deploy and register the Azure App Service resource provider, test it to make sure that users can deploy web and API apps.

### NOTE

You must create an offer that has the Microsoft.Web namespace within the plan. Then, you need to have a tenant subscription that subscribes to this offer. For more info, see [Create offer](#) and [Create plan](#).

You *must* have a tenant subscription to create apps that use Azure App Service on Azure Stack Hub. The only capabilities that a service admin can complete within the administrator portal are related to the resource provider administration of Azure App Service. These capabilities include adding capacity, configuring deployment sources, and adding Worker tiers and SKUs.

As of the third technical preview, to create web, API, and Azure Functions apps, you must use the user portal and have a tenant subscription.

1. In the Azure Stack Hub user portal, select + **Create a resource** > **Web + Mobile** > **Web App**.
2. On the **Web App** blade, type a name in the **Web app** box.
3. Under **Resource Group**, select **New**. Type a name in the **Resource Group** box.
4. Select **App Service plan/Location** > **Create New**.
5. On the **App Service plan** blade, type a name in the **App Service plan** box.
6. Select **Pricing tier** > **Free-Shared or Shared-Shared** > **Select** > **OK** > **Create**.
7. In less than a minute, a tile for the new web app appears on the dashboard. Select the tile.

8. On the **Web App** blade, select **Browse** to view the default website for this app.

## Deploy a WordPress, DNN, or Django website (optional)

1. In the Azure Stack Hub user portal, select +, go to Azure Marketplace, deploy a Django website, and wait for successful completion. The Django web platform uses a file system-based database. It doesn't require any additional resource providers, such as SQL or MySQL.
2. If you also deployed a MySQL resource provider, you can deploy a WordPress website from Azure Marketplace. When you're prompted for database parameters, enter the user name as *User1@Server1*, with the user name and server name of your choice.
3. If you also deployed a SQL Server resource provider, you can deploy a DNN website from Azure Marketplace. When you're prompted for database parameters, choose a database on the computer running SQL Server that's connected to your resource provider.

## Next steps

Prepare for additional admin operations for Azure App Service on Azure Stack Hub:

- [Capacity planning](#)
- [Configure deployment sources](#)

# Update Azure App Service on Azure Stack Hub

2 minutes to read • [Edit Online](#)

## IMPORTANT

Apply the 1910 update to your Azure Stack Hub integrated system or deploy the latest Azure Stack Development Kit (ASDK) before deploying Azure App Service 1.8.

In this article, we show you how to upgrade the [Azure App Service resource provider](#) deployed in an internet-connected Azure Stack Hub environment.

## IMPORTANT

Prior to running the upgrade, make sure that you've already completed the [deployment of the Azure App Service on Azure Stack Hub](#). You should also read the [release notes](#) which accompany the 1.8 release so you can learn about new functionality, fixes, and any known issues that could affect your deployment.

## Run the Azure App Service resource provider installer

During this process, the upgrade will:

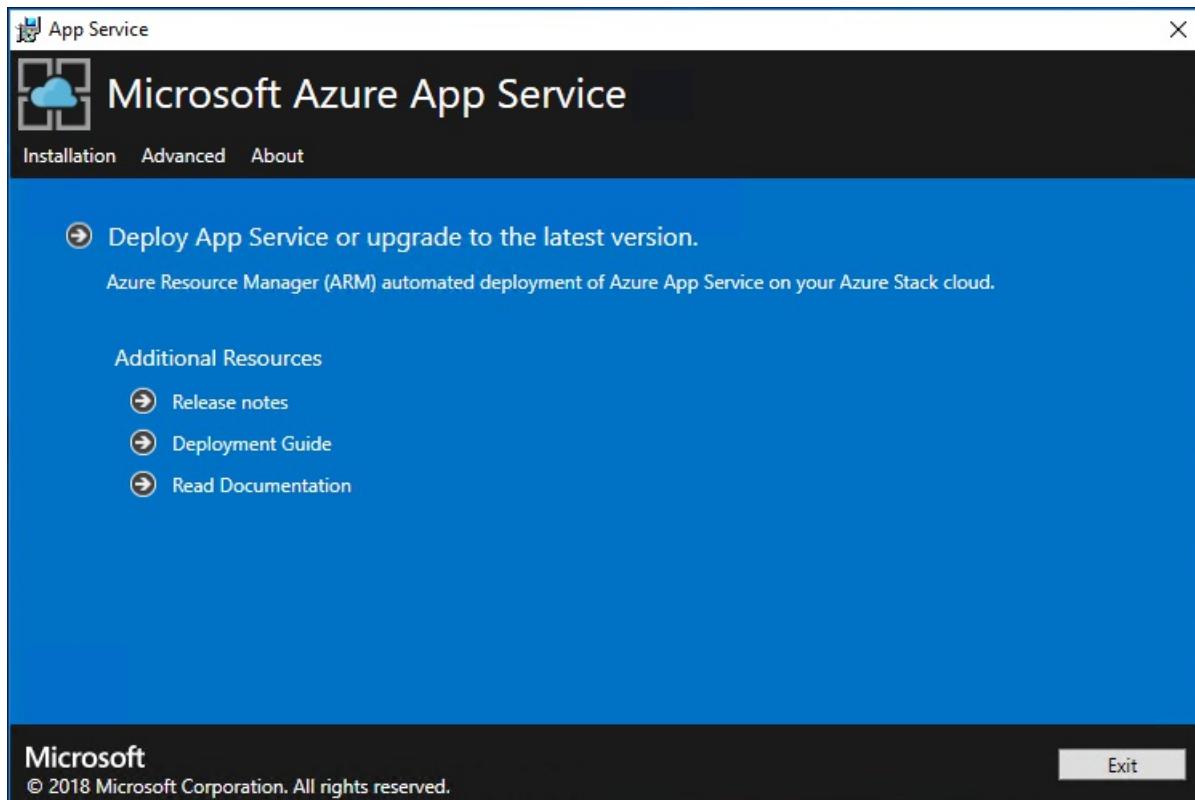
- Detect prior deployment of Azure App Service.
- Prepare all update packages and new versions of all OSS Libraries to be deployed.
- Upload to storage.
- Upgrade all Azure App Service roles (Controllers, Management, Front-End, Publisher, and Worker roles).
- Update Azure App Service scale set definitions.
- Update Azure App Service resource provider manifest.

## IMPORTANT

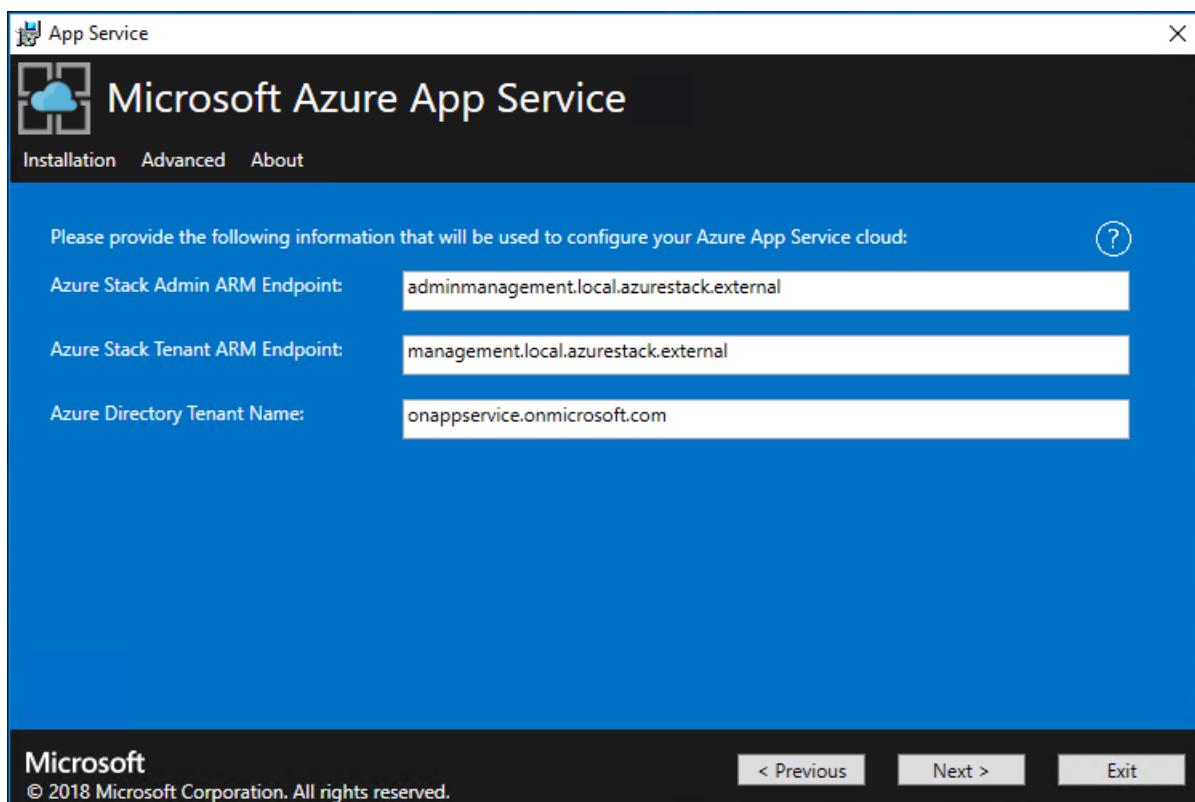
The Azure App Service installer must be run on a machine which can reach the Azure Stack Hub admin Azure Resource Manager endpoint.

To upgrade your deployment of Azure App Service on Azure Stack Hub, follow these steps:

1. Download the [Azure App Service Installer](#).
2. Run appservice.exe as an admin.



3. Click **Deploy Azure App Service or upgrade to the latest version**.
4. Review and accept the Microsoft Software License Terms and then click **Next**.
5. Review and accept the third-party license terms and then click **Next**.
6. Make sure that the Azure Stack Hub Azure Resource Manager endpoint and Active Directory Tenant info is correct. If you used the default settings during ASDK deployment, you can accept the default values here. However, if you customized the options when you deployed Azure Stack Hub, you must edit the values in this window. For example, if you use the domain suffix *mycloud.com*, your Azure Stack Hub Azure Resource Manager endpoint must change to *management.region.mycloud.com*. After you confirm your info, click **Next**.



7. On the next page:

- Select the connection method you wish to use - **Credential** or **Service Principal**

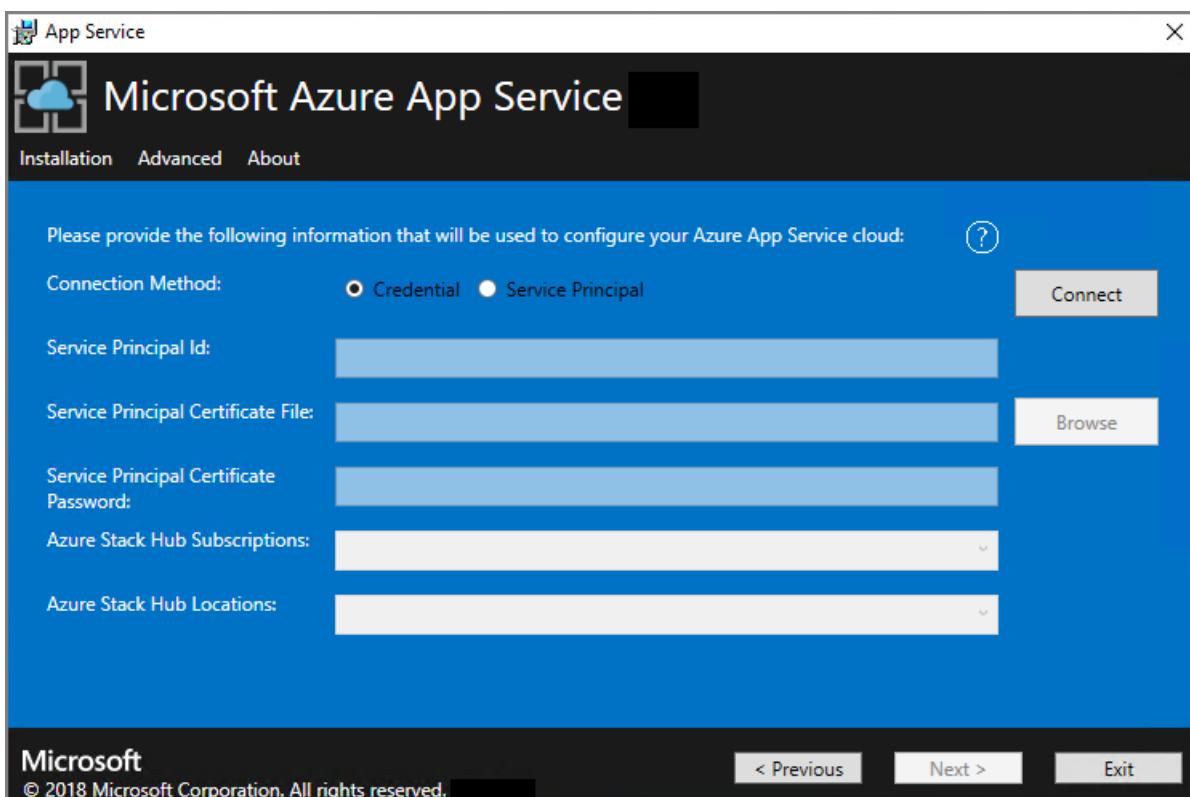
- **Credential**

- If you're using Azure Active Directory (Azure AD), enter the Azure AD admin account and password that you provided when you deployed Azure Stack Hub. Select **Connect**.
- If you're using Active Directory Federation Services (AD FS), provide your admin account. For example, clouddadmin@azurestack.local. Enter your password, and then select **Connect**.

- **Service Principal**

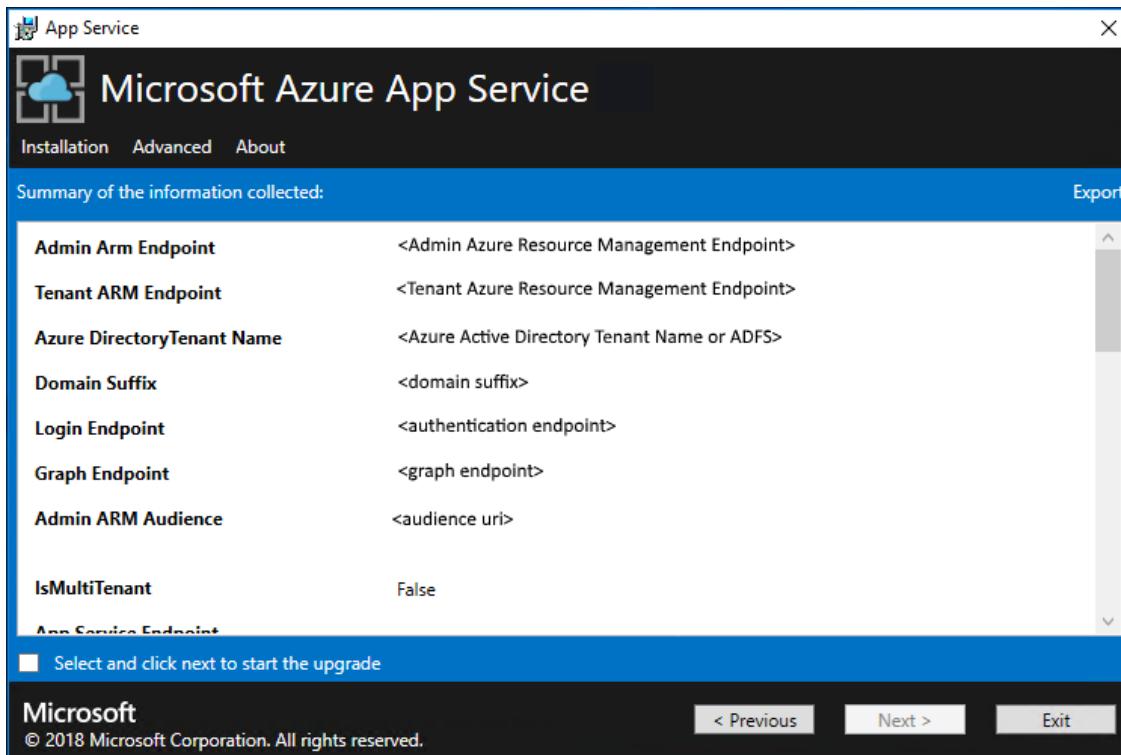
- The service principal which you use **must** have **Owner** rights on the **Default Provider Subscription**
- Provide the **Service Principal ID**, **Certificate File** and **Password** and select **Connect**.

- In **Azure Stack Hub Subscriptions**, select the **Default Provider Subscription**. Azure App Service on Azure Stack Hub **must** be deployed in the **Default Provider Subscription**.
- In the **Azure Stack Hub Locations**, select the location that corresponds to the region you're deploying to. For example, select **local** if you're deploying to the ASDK.
- If an existing Azure App Service deployment is detected, then the resource group and storage account are populated and unavailable.



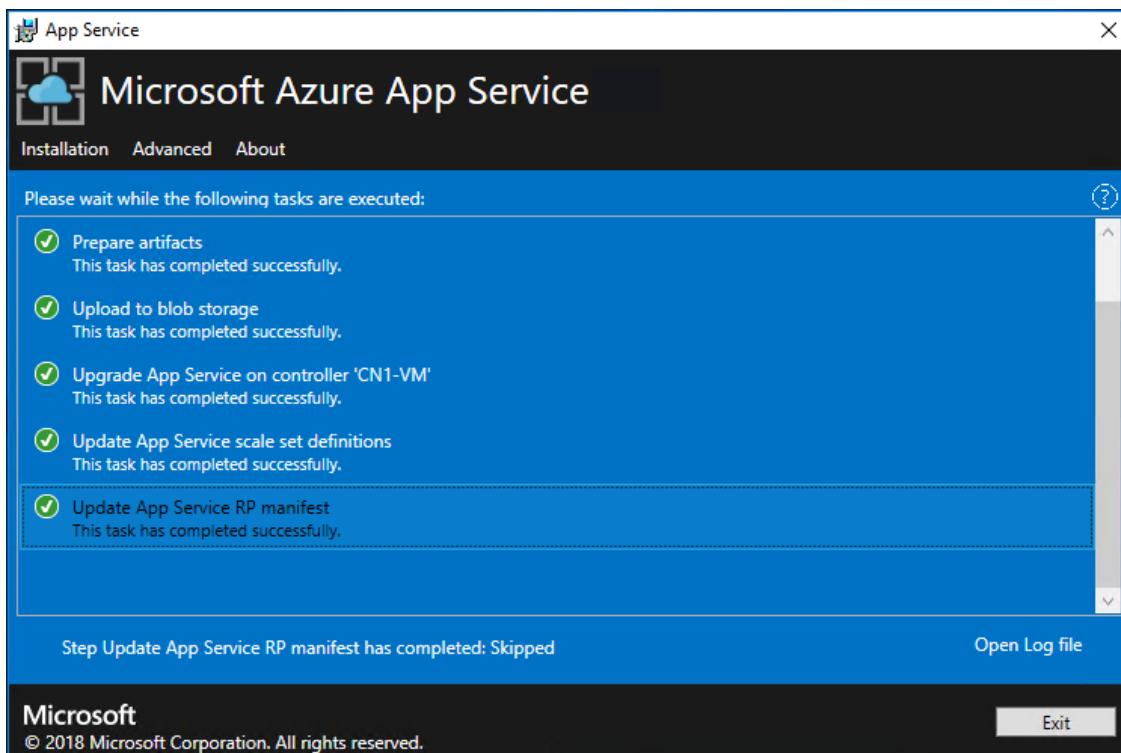
8. On the summary page:

- Verify the selections you made. To make changes, use the **Previous** buttons to visit previous pages.
- If the configurations are correct, select the check box.
- To start the upgrade, click **Next**.



9. Upgrade progress page:

- Track the upgrade progress. The duration of the upgrade of Azure App Service on Azure Stack Hub varies depending on the number of role instances deployed.
- After the upgrade successfully completes, click **Exit**.



## Next steps

Prepare for additional admin operations for Azure App Service on Azure Stack Hub:

- Plan for additional capacity
- Add additional capacity

# Offline update of Azure App Service on Azure Stack Hub

3 minutes to read • [Edit Online](#)

## IMPORTANT

Apply the 1910 update or later to your Azure Stack Hub integrated system or deploy the latest Azure Stack Development Kit before deploying Azure App Service 1.8.

By following the instructions in this article, you can upgrade the [Azure App Service resource provider](#) deployed in an Azure Stack Hub environment that is:

- not connected to the Internet
- secured by Active Directory Federation Services (AD FS).

## IMPORTANT

Prior to running the upgrade, make sure that you have already completed the [deployment of the Azure App Service on Azure Stack Hub Resource Provider](#) and that you have read the [release notes](#), which accompany the 1.8 release, to learn about new functionality, fixes, and any known issues that could affect your deployment.

## Run the App Service resource provider installer

To upgrade the App Service resource provider in an Azure Stack Hub environment, you must complete these tasks:

1. Download the [Azure App Service Installer](#).
2. Create an offline upgrade package.
3. Run the App Service installer (appservice.exe) and complete the upgrade.

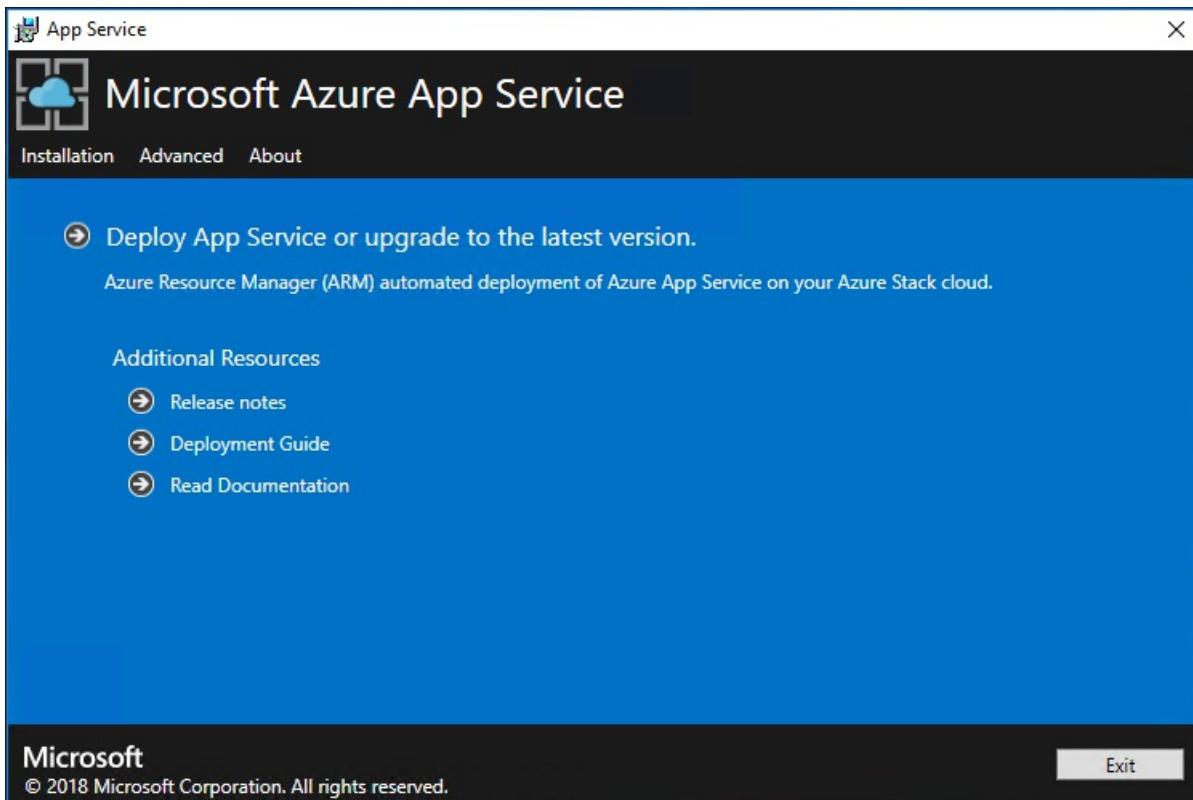
During this process, the upgrade will:

- Detect prior deployment of App Service
- Upload to Storage
- Upgrade all App Service roles (Controllers, Management, Front-End, Publisher, and Worker roles)
- Update App Service scale set definitions
- Update App Service Resource Provider Manifest

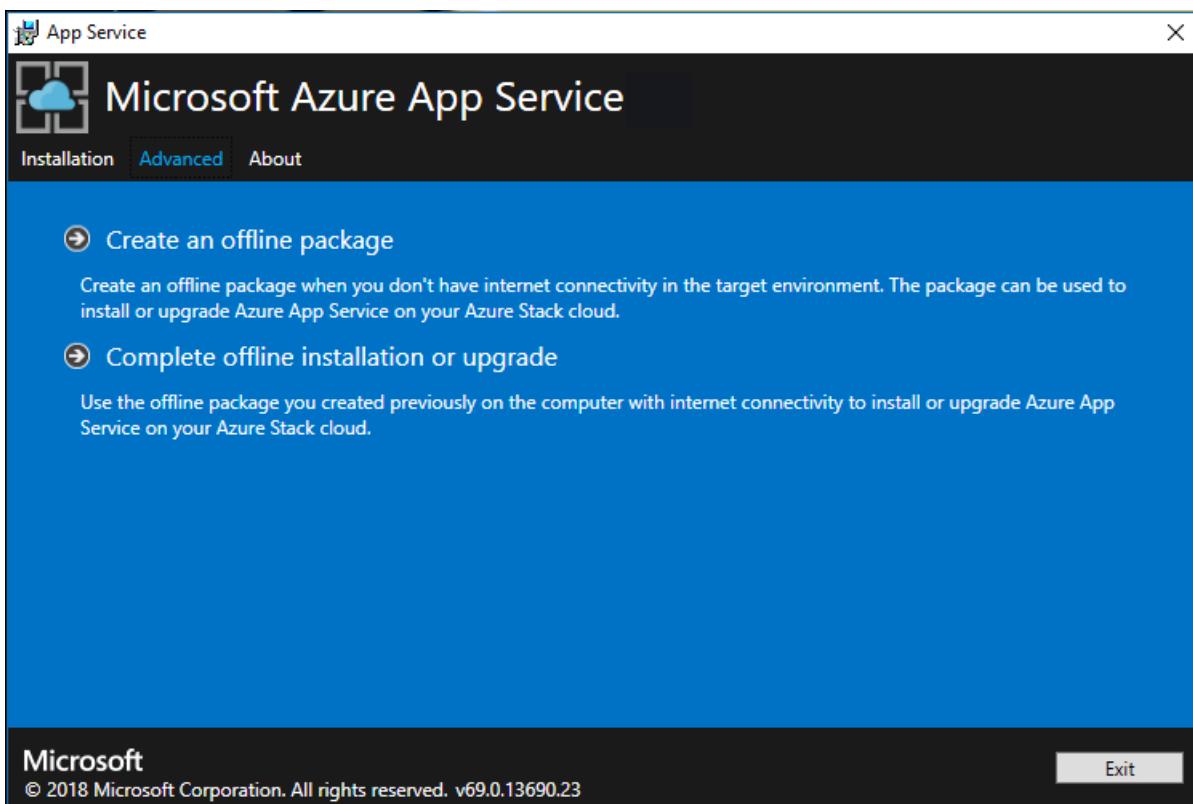
## Create an offline upgrade package

To upgrade App Service in a disconnected environment, you must first create an offline upgrade package on a machine that's connected to the Internet.

1. Run appservice.exe as an administrator



2. Click **Advanced** > **Create offline package**



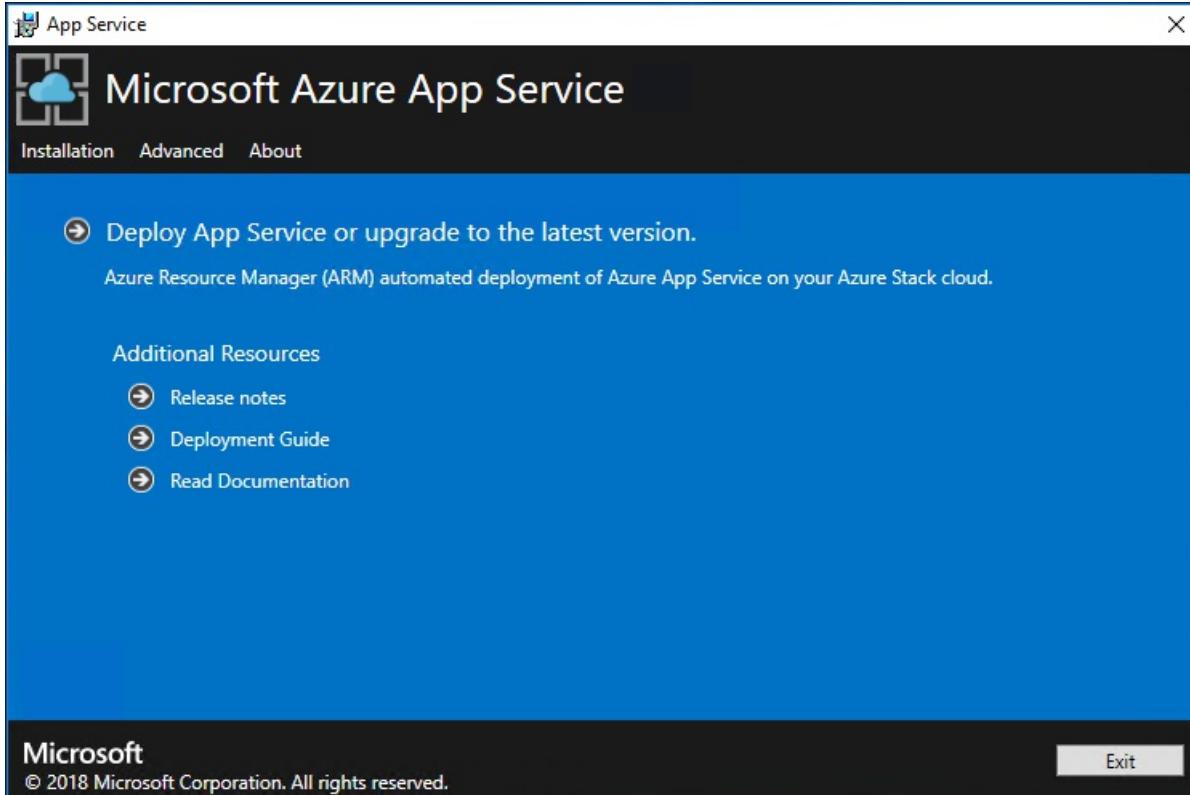
3. The Azure App Service installer creates an offline upgrade package and displays the path to it. You can click **Open folder** to open the folder in your file explorer.
4. Copy the installer (AppService.exe) and the offline upgrade package to your Azure Stack Hub host machine.

Complete the upgrade of App Service on Azure Stack Hub

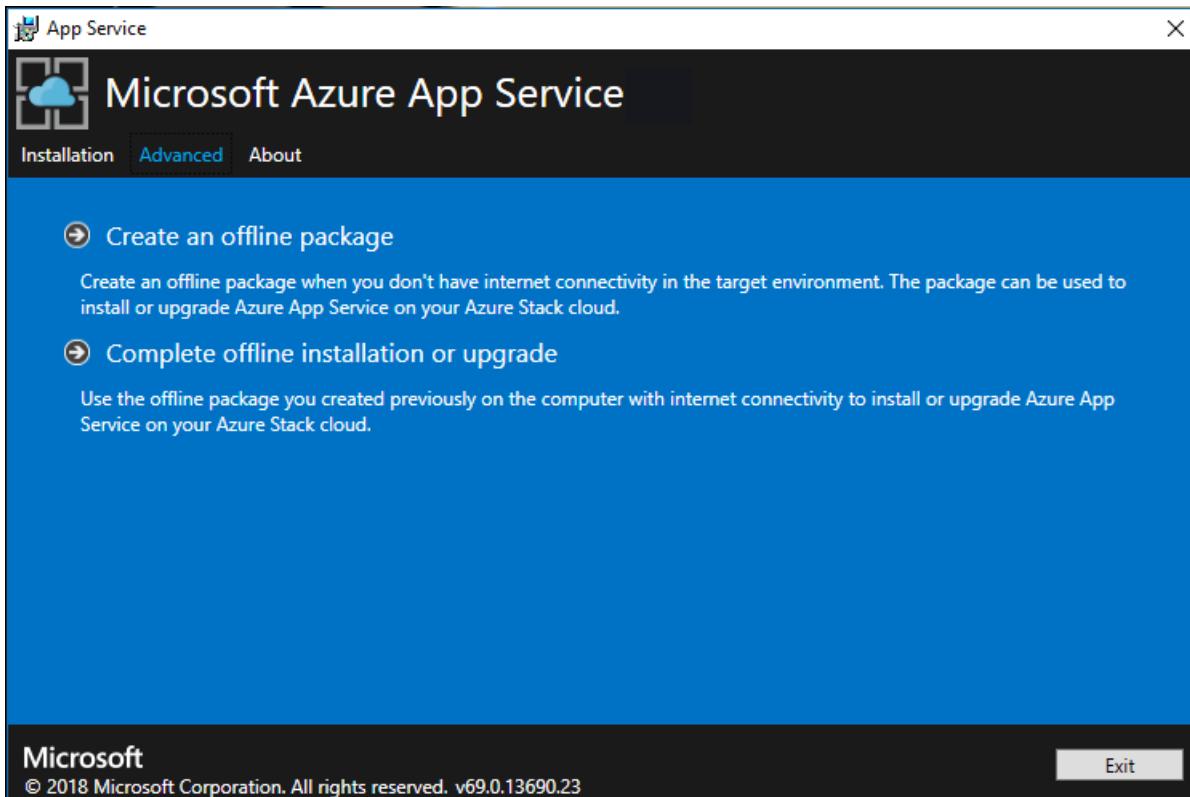
## IMPORTANT

The Azure App Service installer must be run on a machine which can reach the Azure Stack Hub Administrator Azure Resource Manager Endpoint.

1. Run appservice.exe as an administrator.

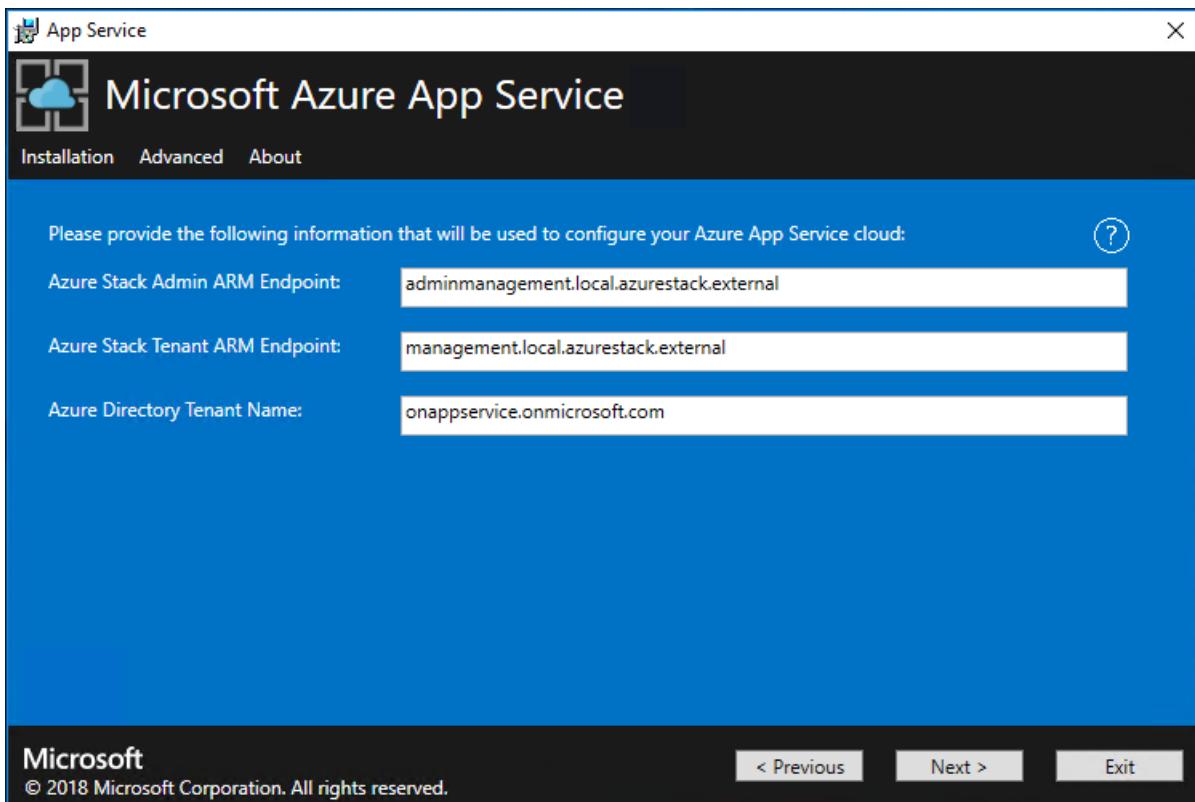


2. Click **Advanced > Complete offline installation or upgrade**.

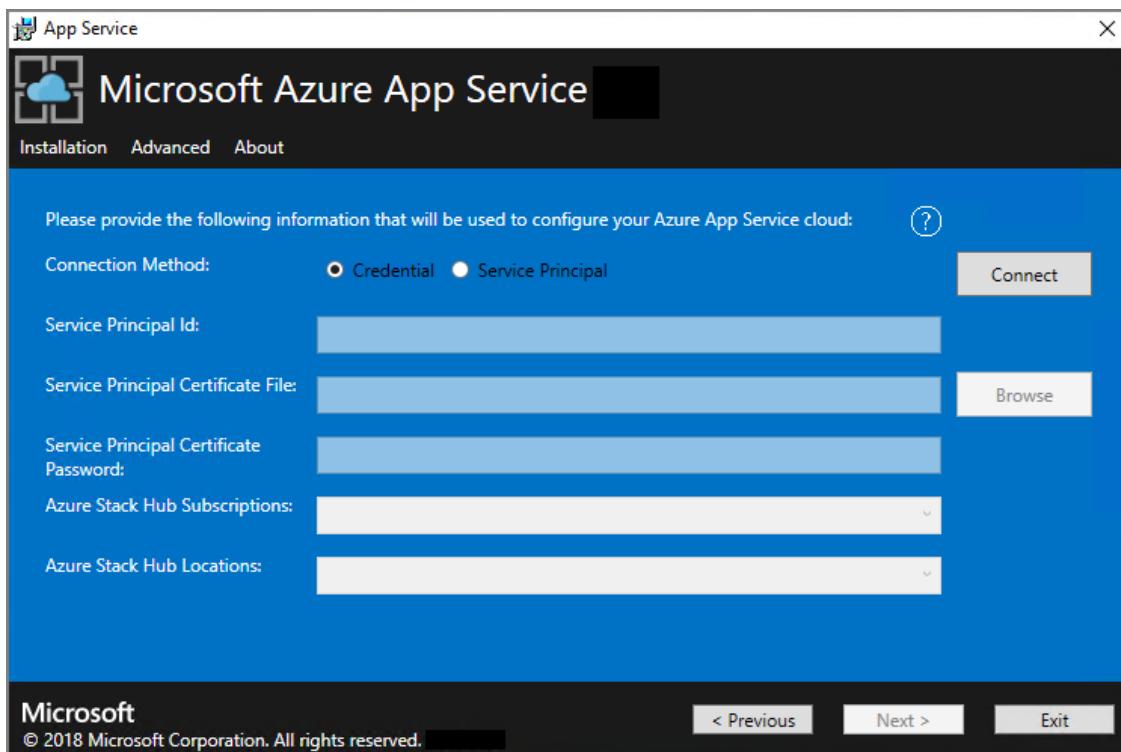


3. Browse to the location of the offline upgrade package you previously created and then click **Next**.

4. Review and accept the Microsoft Software License Terms and then click **Next**.
5. Review and accept the third-party license terms and then click **Next**.
6. Make sure that the Azure Stack Hub Azure Resource Manager endpoint and Active Directory Tenant information is correct. If you used the default settings during Azure Stack Development Kit deployment, you can accept the default values here. However, if you customized the options when you deployed Azure Stack Hub, you must edit the values in this window. For example, if you use the domain suffix *mycloud.com*, your Azure Stack Hub Azure Resource Manager endpoint must change to *management.region.mycloud.com*. After you confirm your information, click **Next**.

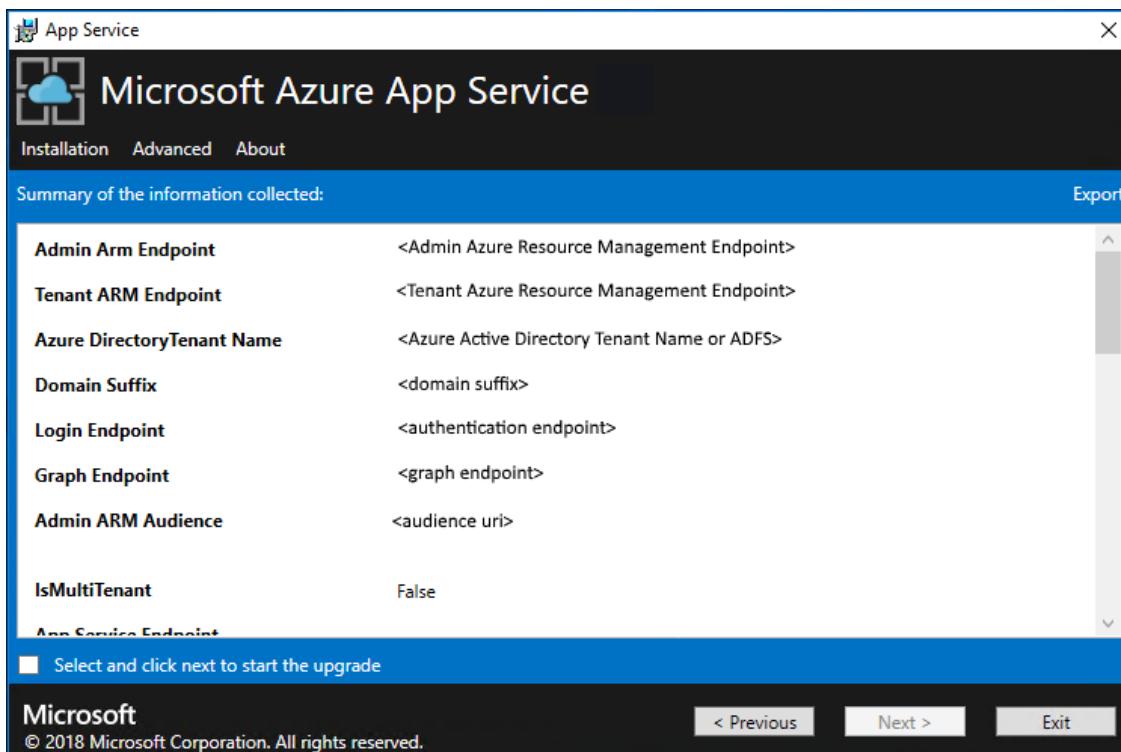


7. On the next page:
  - a. Select the connection method you wish to use - **Credential** or **Service Principal**
    - **Credential**
      - If you're using Azure Active Directory (Azure AD), enter the Azure AD admin account and password that you provided when you deployed Azure Stack Hub. Select **Connect**.
      - If you're using Active Directory Federation Services (AD FS), provide your admin account. For example, *cloudadmin@azurestack.local*. Enter your password, and then select **Connect**.
    - **Service Principal**
      - The service principal which you use **must** have **Owner** rights on the **Default Provider Subscription**
      - Provide the **Service Principal ID**, **Certificate File** and **Password** and select **Connect**.
  - b. In **Azure Stack Hub Subscriptions**, select the **Default Provider Subscription**. Azure App Service on Azure Stack Hub **must** be deployed in the **Default Provider Subscription**.
  - c. In the **Azure Stack Hub Locations**, select the location that corresponds to the region you're deploying to. For example, select **local** if you're deploying to the ASDK.
  - d. If an existing App Service deployment is detected, then the resource group and storage account will be populated and greyed out.



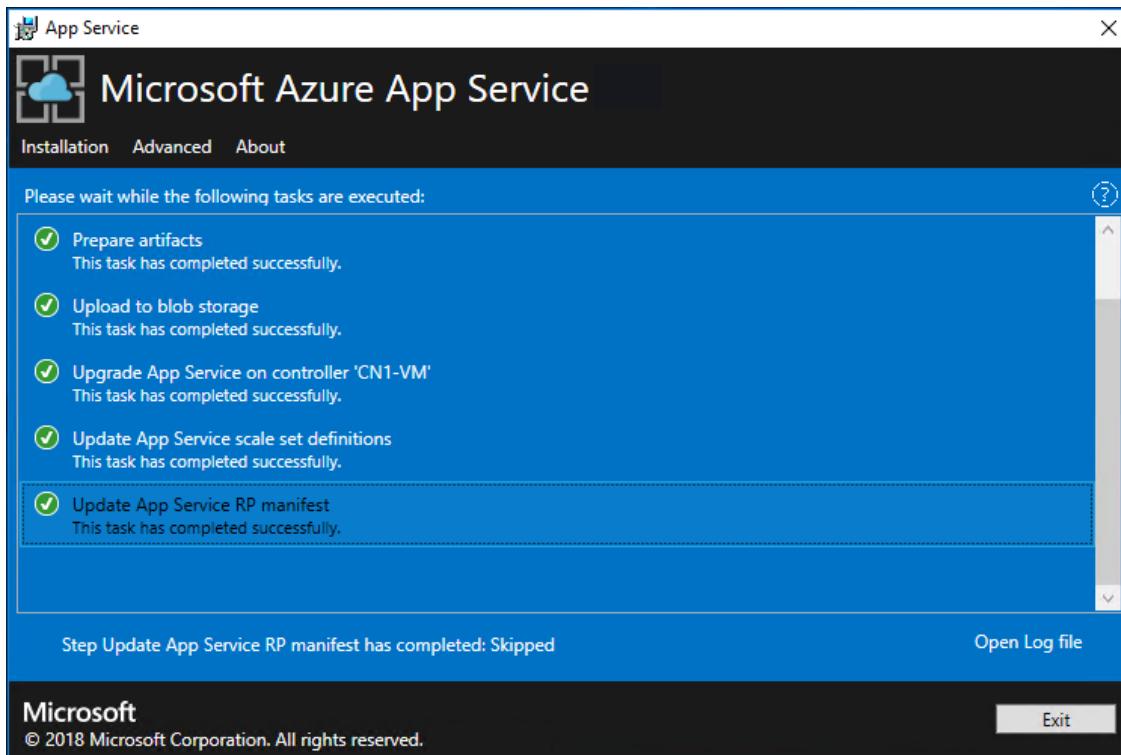
8. On the summary page:

- Verify the selections you made. To make changes, use the **Previous** buttons to visit previous pages.
- If the configurations are correct, select the check box.
- To start the upgrade, click **Next**.



9. Upgrade progress page:

- Track the upgrade progress. The duration of the upgrade of App Service on Azure Stack Hub varies dependent on number of role instances deployed.
- After the upgrade successfully completes, click **Exit**.



## Next steps

Prepare for additional administrator operations for Azure App Service on Azure Stack Hub

- [Plan for additional capacity](#)
- [Add additional capacity](#)

# Add workers and infrastructure in Azure App Service on Azure Stack Hub

2 minutes to read • [Edit Online](#)

This document provides instructions on how to scale infrastructure and worker roles in Azure App Service on Azure Stack Hub. We'll cover all the steps necessary for creating additional worker roles to support apps of any size.

## NOTE

If your Azure Stack Hub Environment doesn't have more than 96-GB RAM, you may have difficulties adding additional capacity.

Azure App Service on Azure Stack Hub supports free and shared worker tiers by default. To add other worker tiers, you need to add more worker roles.

If you're not sure what was deployed with the default Azure App Service on Azure Stack Hub installation, you can review additional info in the [App Service on Azure Stack Hub overview](#).

Azure App Service on Azure Stack Hub deploys all roles using Virtual Machine Scale Sets and as such takes advantage of the scaling capabilities of this workload. Therefore, all scaling of the worker tiers is done via the App Service Admin.

## Add additional workers with PowerShell

1. [Set up the Azure Stack Hub admin environment in PowerShell](#)
2. Use this example to scale out the scale set:

```

Scale out the AppService Role instances
Set context to AzureStack admin.
Login-AzureRmAccount -EnvironmentName AzureStackAdmin

Name of the Resource group where AppService is deployed.
$AppServiceResourceGroupName = "AppService.local"

Name of the ScaleSet : e.g. FrontEndsScaleSet, ManagementServersScaleSet, PublishersScaleSet , LargeWorkerTierScaleSet, MediumWorkerTierScaleSet, SmallWorkerTierScaleSet, SharedWorkerTierScaleSet
$ScaleSetName = "SharedWorkerTierScaleSet"

TotalCapacity is sum of the instances needed at the end of operation.
e.g. if your VMSS has 1 instance(s) currently and you need 1 more the TotalCapacity should be set to 2
$TotalCapacity = 2

Get current scale set
$vmss = Get-AzureRmVmss -ResourceGroupName $AppServiceResourceGroupName -VMScaleSetName $ScaleSetName

Set and update the capacity
$vmss.sku.capacity = $TotalCapacity
Update-AzureRmVmss -ResourceGroupName $AppServiceResourceGroupName -Name $ScaleSetName -VirtualMachineScaleSet $vmss

```

#### NOTE

This step can take a number of hours to complete depending on the type of role and the number of instances.

3. Monitor the status of the new role instances in the App Service administration. To check the status of an individual role instance, click the role type in the list.

## Add additional workers using the administrator portal

1. Sign in to the Azure Stack Hub administrator portal as the service admin.
2. Browse to **App Services**.

The screenshot shows the Microsoft Azure Stack - Administration portal interface. On the left, there's a navigation sidebar with various icons and links: Overview, Properties, System configuration, Source control configuration, Credentials, Roles, IP SSL, IP Block List, Worker Tiers, SKUs, Pricing Tiers, and Subscription Quotas. The main content area is titled 'App Service local'. It displays the following details:

- Resource group:** APPSERVICE-LOCAL
- Status:** All roles are ready
- Location:** local
- Subscription ID:** [redacted]
- DNS suffix:** appservice.local.azurestack.external
- Roles:** 10

Below this, there's a chart titled 'System' with two data series: one blue line representing CPU usage and another red line representing memory usage. The CPU usage line is relatively flat around 40%, while the memory usage line fluctuates between 0% and 10%.

3. Click **Roles**. Here you see the breakdown of all App Service roles deployed.

4. Right click on the row of the type you want to scale and then click **ScaleSet**.

The screenshot shows the 'App Service - Roles' blade. On the left, there's a navigation menu with items like Overview, Properties, System configuration, Source control configuration, Roles (which is selected and highlighted in blue), IP SSL, IP Block List, Worker Tiers, SKUs, Pricing Tiers, and Subscription Quotas. The main area displays a table with columns: ROLE TYPE, WORKER TIER, and INSTANCES. The table contains several rows: Controller (1 instance, General tier), Management Server (2 instances, General tier), Front End (2 instances, General tier), Publisher (2 instances, General tier), Web Worker (0 instances, Small tier), Web Worker (2 instances, Shared tier), Web Worker (0 instances, Medium tier), and Web Worker (0 instances, Large tier). A red box highlights the 'Web Worker' row under 'ROLE TYPE'. Another red box highlights the 'ScaleSet' button in the toolbar at the top right of the table area.

5. Click **Scaling**, select the number of instances you want to scale to, and then click **Save**.

The screenshot shows the 'MediumWorkerTierScaleSet - Scaling' blade. On the left, there's a navigation menu with items like Overview, Activity log, Access control (IAM), Tags, Instances, and Scaling (which is selected and highlighted in blue). The main area has tabs for Save and Discard. It shows settings for SKU (Standard A2) and Number of instances (set to 0). A warning message says: 'You must have at least two running instances in the scale set to qualify for the 99.95% Azure SLA guarantee.' A red box highlights the 'Scaling' tab in the navigation menu, and another red box highlights the 'Number of instances' input field set to 0.

6. Azure App Service on Azure Stack Hub will now add the additional VMs, configure them, install all the required software, and mark them as ready when this process is complete. This process can take approximately 80 minutes.

7. You can monitor the progress of the readiness of the new roles by viewing the workers in the **Roles** blade.

## Result

After they're fully deployed and ready, the workers become available for users to deploy their workload onto them. The following screenshot shows an example of the multiple pricing tiers available by default. If there are no available workers for a particular worker tier, the option to choose the corresponding pricing tier is unavailable.

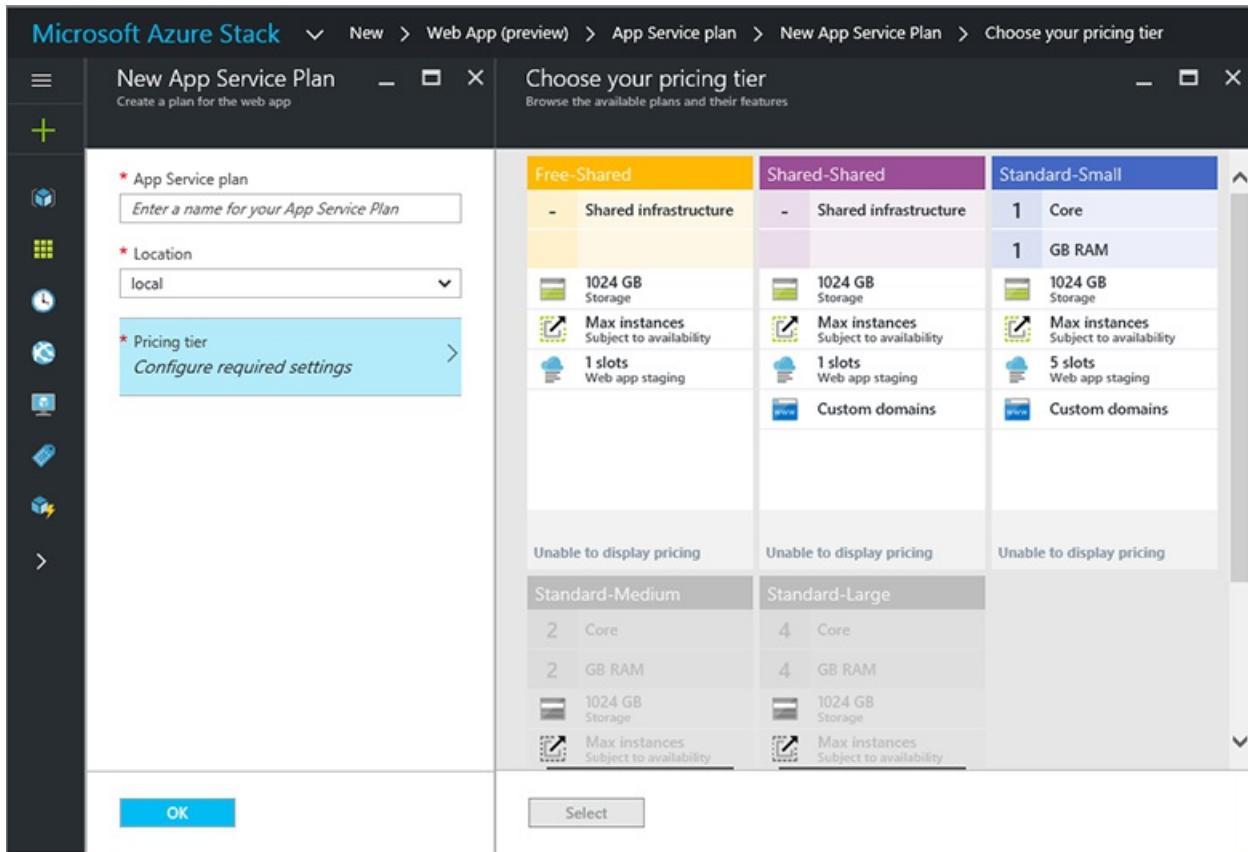
Microsoft Azure Stack    New > Web App (preview) > App Service plan > New App Service Plan > Choose your pricing tier

New App Service Plan    Create a plan for the web app

Choose your pricing tier  
Browse the available plans and their features

| Free-Shared                              | Shared-Shared                            | Standard-Small                           |
|------------------------------------------|------------------------------------------|------------------------------------------|
| - Shared infrastructure                  | - Shared infrastructure                  | 1 Core                                   |
| 1024 GB Storage                          | 1024 GB Storage                          | 1 GB RAM                                 |
| Max instances<br>Subject to availability | Max instances<br>Subject to availability | 1024 GB Storage                          |
| 1 slots<br>Web app staging               | 1 slots<br>Web app staging               | Max instances<br>Subject to availability |
|                                          | Custom domains                           | 5 slots<br>Web app staging               |
|                                          |                                          | Custom domains                           |
| Unable to display pricing                |                                          |                                          |
| Standard-Medium                          | Standard-Large                           | Standard-XLarge                          |
| 2 Core                                   | 4 Core                                   | 8 Core                                   |
| 2 GB RAM                                 | 4 GB RAM                                 | 8 GB RAM                                 |
| 1024 GB Storage                          | 1024 GB Storage                          | 1024 GB Storage                          |
| Max instances<br>Subject to availability | Max instances<br>Subject to availability | Max instances<br>Subject to availability |

OK    Select



#### NOTE

To scale out Management, Front End, or Publisher roles, follow the same steps selecting the appropriate role type. Controllers aren't deployed as Scale Sets and therefore two should be deployed at installation time for all production deployments.

#### Next steps

[Configure deployment sources](#)

# Configure deployment sources for App Services on Azure Stack Hub

4 minutes to read • [Edit Online](#)

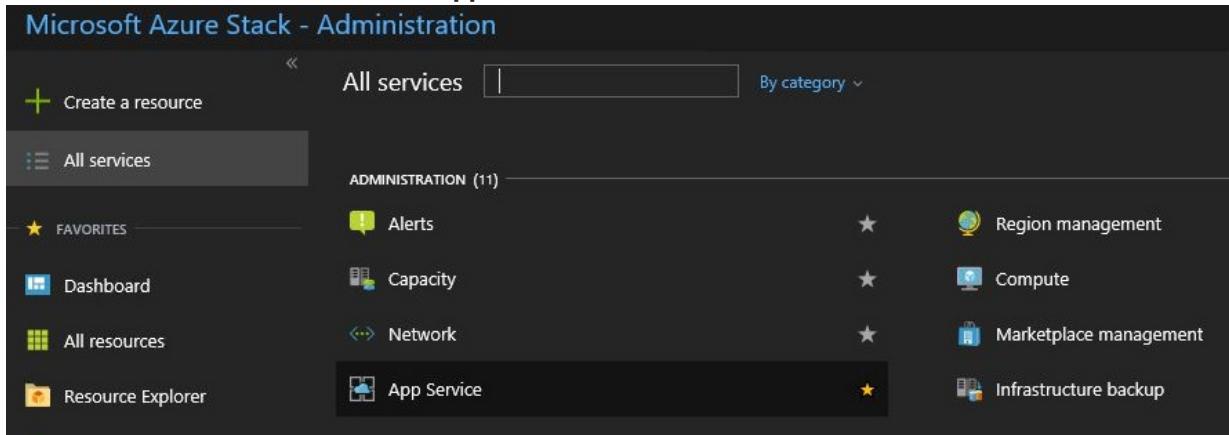
App Service on Azure Stack Hub supports on-demand deployment from multiple source control providers. This feature lets app developers deploy direct from their source control repositories. If users want to configure App Service to connect to their repositories, a cloud operator must first configure the integration between App Service on Azure Stack Hub and the source control provider.

In addition to local Git, the following source control providers are supported:

- GitHub
- BitBucket
- OneDrive
- DropBox

## View deployment sources in App Service administration

1. Sign in to the Azure Stack Hub admin portal (<https://adminportal.local.azurestack.external>) as the service admin.
2. Browse to **All Services** and select the **App Service**.



3. Click **Source control configuration**. You can see the list of all configured deployment sources.

## Configure GitHub

You must have a GitHub account to complete this task. You might want to use an account for your organization rather than a personal account.

1. Sign in to GitHub, browse to <https://www.github.com/settings/developers>, and then click **Register a new application**.

2. Enter an **Application name**. For example, **App Service on Azure Stack Hub**.
3. Enter the **Homepage URL**. The Homepage URL must be the Azure Stack Hub portal address. For example, <https://portal.local.azurestack.external>.
4. Enter an **Application Description**.
5. Enter the **Authorization callback URL**. In a default Azure Stack Hub deployment, the URL is in the form [https://portal.local.azurestack.external TokenNameAuthorise](https://portal.local.azurestack.external	TokenNameAuthorise). If you're running under a different domain, switch your domain name for local.azurestack.external.
6. Click **Register application**. A page is displayed listing the **Client ID** and **Client Secret** for the app.

The screenshot shows the GitHub application settings for "App Service on Azure Stack". The left sidebar lists personal settings like Profile, Account, Emails, Notifications, Billing, SSH and GPG keys, Security, Blocked users, Repositories, Organizations, Saved replies, Authorized applications, and Installed integrations. The main area displays the application details: "antaresonprem" owns the app, it has 0 users, and shows Client ID (6e...) and Client Secret (3c...). Buttons for "Revoke all user tokens" and "Reset client secret" are present. An "Upload new logo" button is available, with a placeholder "Drag & drop". A note says "You can also drag and drop a picture from your computer."

7. In a new browser tab or window, sign in to the Azure Stack Hub admin portal (<https://adminportal.local.azurestack.external>) as the service admin.
8. Browse to **Resource Providers**, and select the **App Service Resource Provider Admin**.
9. Click **Source control configuration**.
10. Copy and paste the **Client ID** and **Client Secret** into the corresponding input boxes for GitHub.
11. Click **Save**.

## Configure BitBucket

You must have a BitBucket account to complete this task. You might want to use an account for your organization rather than a personal account.

1. Sign in to BitBucket and browse to **Integrations** under your account.

The screenshot shows the BitBucket dashboard. The top navigation bar includes links for Bitbucket, Teams, Projects, Repositories, and Snippets. On the right, there's a search bar and user profile icons. The main content area is titled "Dashboard" and includes tabs for Overview, Repositories, Pull requests, Issues, and Snippets. On the far right, a vertical menu is open, showing options: "App Service On-Premises" (with a URL), "Manage Atlassian account", "View profile", "Bitbucket settings", "Integrations" (which is highlighted with a yellow background), and "Log out".

2. Click **OAuth** under Access Management and **Add consumer**.

## Add OAuth consumer

### Details

|                                                                                                                                 |                                |                                |
|---------------------------------------------------------------------------------------------------------------------------------|--------------------------------|--------------------------------|
| Name *                                                                                                                          | <input type="text"/>           |                                |
| Description                                                                                                                     | <input type="text"/>           |                                |
| Callback URL                                                                                                                    | <input type="text"/>           |                                |
| URL users will be redirected to after access authorization. Required for OAuth 2.                                               |                                |                                |
| URL                                                                                                                             | <input type="text"/>           |                                |
| Optional URL where users can learn more about your application.                                                                 |                                |                                |
| <input checked="" type="checkbox"/> This is a private consumer                                                                  |                                |                                |
| Installable applications that ship their OAuth consumer credentials as part of the application should not be marked as private. |                                |                                |
| <b>Permissions</b>                                                                                                              |                                |                                |
| Account                                                                                                                         | <input type="checkbox"/> Email | <input type="checkbox"/> Write |
|                                                                                                                                 | <input type="checkbox"/> Read  | <input type="checkbox"/> Read  |
|                                                                                                                                 | <input type="checkbox"/> Write | <input type="checkbox"/> Write |

3. Enter a **Name** for the consumer. For example, **App Service on Azure Stack Hub**.
4. Enter a **Description** for the app.
5. Enter the **Callback URL**. In a default Azure Stack Hub deployment, the callback URL is in the form <https://portal.local.azurestack.external/TokensAuthorize>. If you're running under a different domain, substitute your domain name for azurestack.local. For BitBucket integration to succeed, the URL must follow the capitalization listed here.
6. Enter the **URL**. This URL should be the Azure Stack Hub portal URL. For example, <https://portal.local.azurestack.external>.
7. Select the **Permissions** required:
  - **Repositories**: *Read*
  - **Webhooks**: *Read and write*
8. Click **Save**. You now see this new app, along with the **Key** and **Secret**, under **OAuth consumers**.

### OAuth consumers

Generate your own OAuth consumer key and secret to [build your own custom integration with Bitbucket](#).

[Add consumer](#)

| Name             | Description                                                                                              | ... |
|------------------|----------------------------------------------------------------------------------------------------------|-----|
| ▼ App Service... | App Service on Azure Stack<br>URL https://portal.azurestack.local<br>Key [REDACTED]<br>Secret [REDACTED] | ... |

9. In a new browser tab or window, sign in to the Azure Stack Hub admin portal (<https://adminportal.local.azurestack.external>) as the service admin.

10. Browse to **Resource Providers** and select the **App Service Resource Provider Admin**.
11. Click **Source control configuration**.
12. Copy and paste the **Key** into the **Client ID** input box and **Secret** into the **Client Secret** input box for BitBucket.
13. Click **Save**.

## Configure OneDrive

You must have a Microsoft Account linked to a OneDrive account to complete this task. You might want to use an account for your organization rather than a personal account.

### NOTE

OneDrive for Business accounts are currently not supported.

1. Browse to <https://apps.dev.microsoft.com/?referrer=https%3A%2F%2Fdev.onedrive.com%2Fapp-registration.htm> and sign in using your Microsoft Account.
2. Under **My applications**, click **Add an app**.

The screenshot shows the Microsoft Application Registration Portal. At the top, there's a header with the Microsoft logo, the text 'Application Registration Portal', and a user icon labeled 'OnPrem'. Below the header, the main title is 'My applications' with a 'Learn More' link. To the right of the title is a blue button labeled 'Add an app'. A table below the title lists two columns: 'Name' and 'App ID / Client Id'. There are no entries in the table.

3. Enter a **Name** for the new app registration: enter **App Service on Azure Stack Hub**, and then click **Create Application**.
4. The next screen lists the properties of your new app. Save the **Application ID** to some temporary location.

# App Service on Azure Stack Registration

## Properties Learn More

Name



Application Id



## Application Secrets Learn More

[Generate New Password](#) [Generate New Key Pair](#)

## Platforms

[Add Platform](#)

## Microsoft Graph Permissions

The settings you set here may vary depending on whether you get a token from our V1 or V2 endpoint. [Learn More](#)

Delegated Permissions [Add](#) [Learn More](#)

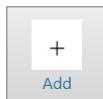
 

Application Permissions [Add](#) [Learn More](#)

## Profile Learn More

Logo

The logo must be a transparent 48 x 48 or 50 x 50 pixel image in a GIF, PNG or JPEG file that is 15 KB or smaller.



5. Under **Application Secrets**, click **Generate New Password**. Make a note of the **New password generated**. This password is your application secret and it's not retrievable after you click **OK**.

6. Under **Platforms**, click **Add Platform**, and then select **Web**.

7. Enter the **Redirect URI**. In a default Azure Stack Hub deployment, the Redirect URI is in the form <https://portal.local.azurestack.external/TokenAuthorize>. If you're running under a different domain, switch your domain name for azurestack.local.

## Platforms

[Add Platform](#)

|     |                        |
|-----|------------------------|
| Web | <a href="#">Delete</a> |
|-----|------------------------|

Allow Implicit Flow

Redirect URIs [Add Url](#)

Click here for help integrating your application with Microsoft.

8. Add the **Microsoft Graph Permissions - Delegated Permissions**.

- **Files.ReadWrite.AppFolder**
- **User. Read**

## Microsoft Graph Permissions

The settings you set here may vary depending on whether you get a token from our V1 or V2 endpoint. [Learn More](#)

Delegated Permissions [Add](#) [Learn More](#)

Files.ReadWrite.AppFolder [X](#)

User.Read [X](#)

Application Permissions [Add](#) [Learn More](#)

9. Click **Save**.
10. In a new browser tab or window, sign in to the Azure Stack Hub admin portal (<https://adminportal.local.azurestack.external>) as the service admin.
11. Browse to **Resource Providers** and select the **App Service Resource Provider Admin**.
12. Click **Source control configuration**.
13. Copy and paste the **Application ID** into the **Client ID** input box and **Password** into the **Client Secret** input box for OneDrive.
14. Click **Save**.

## Configure DropBox

### NOTE

You must have a DropBox account to complete this task. You might want to use an account for your organization rather than a personal account.

1. Browse to <https://www.dropbox.com/developers/apps> and sign in using your DropBox account credentials.
2. Click **Create app**.



3. Select **DropBox API**.
4. Set the access level to **App Folder**.
5. Enter a **Name** for your app.



## Create a new app on the Dropbox Platform

API v2

My apps

API Explorer

Documentation

HTTP

.NET

Java

JavaScript

Python

Swift

Objective-C

Community SDKs

References

Authentication types

Branding guide

Data ingress guide

Developer guide

OAuth guide

v2 migration guide

Webhooks

Chooser

Saver

API v1

Blog

Support

### 1. Choose an API

Dropbox API

For apps that need to access files in Dropbox. [Learn more](#)



Dropbox Business API

For apps that need access to Dropbox Business team info. [Learn more](#)



### 2. Choose the type of access you need

[Learn more about access types](#)

App folder – Access to a single folder created specifically for your app.

Full Dropbox – Access to all files and folders in a user's Dropbox.

### 3. Name your app

App Service on Azure Stack

x

**Create app**

6. Click **Create App**. You're presented with a page listing the settings for the app, including **App key** and **App secret**.
7. Make sure that the **App folder name** is set to **App Service on Azure Stack Hub**.
8. Set the **OAuth 2 Redirect URI** and then click **Add**. In a default Azure Stack Hub deployment, the Redirect URI is in the form [https://portal.local.azurestack.external TokenNameAuthorise](https://portal.local.azurestack.external	TokenNameAuthorise). If you're running under a different domain, switch your domain for azurestack.local.

The screenshot shows the Azure Stack Hub Admin portal interface. At the top, there are three tabs: Settings, Branding, and Analytics. The Analytics tab is currently selected. Below the tabs, there are several sections:

- Status**: Shows the status as **Development**. There are buttons for **Apply for production** and **Unlink all users**.
- Development users**: Shows 1 / 500 users. There is a button for **Unlink all users**.
- Permission type**: Set to **App folder**.
- App folder name**: Set to **App Service On Azure Stack**, with a **Change** button.
- App key**: A redacted value.
- App secret**: A link to **Show**.
- OAuth 2** section:
  - Redirect URIs**: A list containing <https://portal.azurestack.local/tokenauthorize>. There is a remove button (**X**) and an **Add** button.
  - Allow implicit grant**: A dropdown set to **Allow**.
  - Generated access token**: A **Generate** button.
- Chooser/Saver domains**: A list containing **example.com**, with an **Add** button. A note below says: "If using the [Chooser](#) or the [Saver](#) on a website, the domain of that site."
- Webhooks** section:
  - Webhook URIs**: A list containing <https://>, with an **Add** button.
- Delete app**: Buttons for **Delete app** and **Cancel**.

9. In a new browser tab or window, sign in to the Azure Stack Hub admin portal (<https://adminportal.local.azurestack.external>) as the service admin.
10. Browse to **Resource Providers** and select the **App Service Resource Provider Admin**.
11. Click **Source control configuration**.
12. Copy and paste the **Application Key** into the **Client ID** input box and **App secret** into the **Client Secret** input box for DropBox.
13. Click **Save**.

## Next steps

Users can now use the deployment sources for things like [continuous deployment](#), [local Git deployment](#), and [cloud folder synchronization](#).

# Rotate App Service on Azure Stack Hub secrets and certificates

2 minutes to read • [Edit Online](#)

These instructions only apply to Azure App Service on Azure Stack Hub. Rotation of Azure App Service on Azure Stack Hub secrets is not included in the centralized secret rotation procedure for Azure Stack Hub. Operators can monitor the validity of secrets within the system, the date on which they were last updated and the time remaining until the secrets expire.

## IMPORTANT

Operators will not receive alerts for secret expiration on the Azure Stack Hub dashboard as Azure App Service on Azure Stack Hub is not integrated with the Azure Stack Hub alerting service. Operators must regularly monitor their secrets using the Azure App Service on Azure Stack Hub administration experience in the Azure Stack Hub Administrators portal.

This document contains the procedure for rotating the following secrets:

- Encryption Keys used within Azure App Service on Azure Stack Hub;
- Database connection credentials used by Azure App Service on Azure Stack Hub to interact with the hosting and metering databases;
- Certificates used by Azure App Service on Azure Stack Hub to secure endpoints;
- System credentials for Azure App Service on Azure Stack Hub infrastructure roles.

## Rotate encryption keys

To rotate the encryption keys used within Azure App Service on Azure Stack Hub, complete the following steps:

1. Go to the App Service Administration experience in the Azure Stack Hub Administrators Portal.
2. Navigate to the **Secrets** menu option
3. Click the **Rotate** button in the Encryption Keys section
4. Click **OK** to start the rotation procedure.
5. The encryption keys are rotated and all role instances are updated. Operators can monitor the Status of the procedure using the **Status** button.

## Rotate connection strings

To update the credentials for the database connection string for the App Service hosting and metering databases, complete the following steps:

1. Go to the App Service Administration experience in the Azure Stack Hub Administrators Portal.
2. Navigate to the **Secrets** menu option
3. Click the **Rotate** button in the Connection Strings section
4. Provide the **SQL SA Username** and **Password** and click **OK** to start the rotation procedure.
5. The credentials will be rotated throughout the Azure App Service role instances. Operators can monitor the

Status of the procedure using the **Status** button.

## Rotate certificates

To rotate the certificates used within Azure App Service on Azure Stack Hub, complete the following steps:

1. Go to the App Service Administration experience in the Azure Stack Hub Administrators Portal.
2. Navigate to the **Secrets** menu option
3. Click the **Rotate** button in the Certificates section
4. Provide the **certificate file** and associated **password** for the certificates you wish to rotate and click **OK**.
5. The certificates will be rotated as required throughout the Azure App Service on Azure Stack Hub role instances. Operators can monitor the status of the procedure using the **Status** button.

## Rotate system credentials

To rotate the System Credentials used within Azure App Service on Azure Stack Hub, complete the following steps:

1. Go to the App Service Administration experience in the Azure Stack Hub Administrators Portal.
2. Navigate to the **Secrets** menu option
3. Click the **Rotate** button in the System Credentials section
4. Select the **Scope** of the System Credential you are rotating. Operators can choose to rotate the System Credentials for All roles or individual roles.
5. Specify the **Local Admin User Name**, new **Password** and confirm the **Password** and click **OK**
6. The credential(s) will be rotated as required throughout the corresponding Azure App Service on Azure Stack Hub role instance. Operators can monitor the status of the procedure using the **Status** button.

# Back up App Service on Azure Stack Hub

2 minutes to read • [Edit Online](#)

This document provides instructions on how to back up App Service on Azure Stack Hub.

## IMPORTANT

App Service on Azure Stack Hub isn't backed up as part of [Azure Stack Hub infrastructure backup](#). As an Azure Stack Hub Operator, you must take steps to ensure App Service can be successfully recovered if necessary.

Azure App Service on Azure Stack Hub has four main components to consider when planning for disaster recovery:

1. The resource provider infrastructure; server roles, worker tiers, and so on.
2. The App Service secrets.
3. The App Service SQL Server hosting and metering databases.
4. The App Service user workload content stored in the App Service file share.

## Back up App Service secrets

When recovering App Service from backup, you need to provide the App Service keys used by the initial deployment. This information should be saved as soon as App Service is successfully deployed and stored in a safe location. The resource provider infrastructure configuration is recreated from backup during recovery using App Service recovery PowerShell cmdlets.

Use the administration portal to back up app service secrets by following these steps:

1. Sign in to the Azure Stack Hub administrator portal as the service admin.
2. Browse to **App Service -> Secrets**.
3. Select **Download Secrets**.

The screenshot shows the 'App Service - Secrets' blade in the Azure portal. On the left, there's a navigation menu with items like Overview, Properties, System configuration, Secrets (which is selected and highlighted with a red box), Source control configuration, Roles, IP SSL, IP Block List, Worker Tiers, SKUs, Pricing Tiers, Subscription Quotas, and Custom Software. On the right, there are four sections: 'Encryption Keys', 'Connection Strings', 'Certificates', and 'System Credentials'. Each section has a 'Rotate' button and a 'Status' button. At the top right, there's a large blue button labeled 'Download Secrets' with a downward arrow icon, which is also highlighted with a red box.

4. When secrets are ready for downloading, click **Save** and store the App Service secrets (**SystemSecrets.JSON**) file in a safe location.

The screenshot shows the same 'App Service - Secrets' blade as before, but now it displays a message: 'Secrets ready for downloading.' Below this message, the 'Download Secrets' button is still present, but the 'Save' button in the row above it is now highlighted with a red box. The rest of the interface remains the same, with the 'Encryption Keys', 'Connection Strings', 'Certificates', and 'System Credentials' sections visible.

#### NOTE

Repeat these steps every time the App Service secrets are rotated.

## Back up the App Service databases

To restore App Service, you need the **Appservice\_hosting** and **Appservice\_metering** database backups. We recommend using SQL Server maintenance plans or Azure Backup Server to ensure these databases are backed up and saved securely on a regular basis. However, any method of ensuring regular SQL backups are created can be used.

To manually back up these databases while logged into the SQL Server, use the following PowerShell commands:

```
$s = "<SQL Server computer name>"
$u = "<SQL Server login>"
$p = read-host "Provide the SQL admin password"
sqlcmd -S $s -U $u -P $p -Q "BACKUP DATABASE appservice_hosting TO DISK = '<path>\hosting.bak'"
sqlcmd -S $s -U $u -P $p -Q "BACKUP DATABASE appservice_metering TO DISK = '<path>\metering.bak'"
```

### NOTE

If you need to back up SQL AlwaysOn databases, follow [these instructions](#).

After all databases have been successfully backed up, copy the .bak files to a safe location along with the App Service secrets info.

## Back up the App Service file share

App Service stores tenant app info in the file share. This file share must be backed up on a regular basis along with the App Service databases so that as little data as possible is lost if a restore is required.

To back up the App Service file share content, use Azure Backup Server or another method to regularly copy the file share content to the location you've saved all previous recovery info.

For example, you can use these steps to use Robocopy from a Windows PowerShell (not PowerShell ISE) console session:

```
$source = "<file share location>"
$destination = "<remote backup storage share location>"
net use $destination /user:<account to use to connect to the remote share in the format of domain\username> *
robocopy $source $destination
net use $destination /delete
```

## Next steps

[Restore App Service on Azure Stack Hub](#)

# App Service recovery on Azure Stack Hub

4 minutes to read • [Edit Online](#)

This topic provides instructions on what actions to take for App Service disaster recovery.

The following actions must be taken to recover App Service on Azure Stack Hub from backup:

1. Restore the App Service databases.
2. Restore the file server share content.
3. Restore App Service roles and services.

If Azure Stack Hub storage was used for Function Apps storage, then you must also take steps to restore Function Apps.

## Restore the App Service databases

The App Service SQL Server databases should be restored on a production ready SQL Server instance.

After [preparing the SQL Server instance](#) to host the App Service databases, use these steps to restore databases from backup:

1. Sign in to the SQL Server that will host the recovered App Service databases with admin permissions.
2. Use the following commands to restore the App Service databases from a command prompt running with admin permissions:

```
sqlcmd -U <SQL admin login> -P <SQL admin password> -Q "RESTORE DATABASE appservice_hosting FROM DISK='<full path to backup>' WITH REPLACE"
sqlcmd -U <SQL admin login> -P <SQL admin password> -Q "RESTORE DATABASE appservice_metering FROM DISK='<full path to backup>' WITH REPLACE"
```

3. Verify that both App Service databases have been successfully restored and exit SQL Server Management Studio.

### NOTE

To recover from a failover cluster instance failure, see [Recover from Failover Cluster Instance Failure](#).

## Restore the App Service file share content

After [preparing the file server](#) to host the App Service file share, you need to restore the tenant file share content from backup. You can use whatever method you have available to copy the files into the newly created App Service file share location. Running this example on the file server will use PowerShell and robocopy to connect to a remote share and copy the files to the share:

```
$source = "<remote backup storage share location>"
$destination = "<local file share location>
net use $source /user:<account to use to connect to the remote share in the format of domain\username> *
robocopy /E $source $destination
net use $source /delete"
```

In addition to copying the file share contents, you must also reset permissions on the file share itself. To reset

permissions, open an admin command prompt on the file server computer and run the **ReACL.cmd** file. The **ReACL.cmd** file is located in the App Service installation files in the **BCDR** directory.

## Restore App Service roles and services

After the App Service databases and file share content are restored, you next need to use PowerShell to restore the App Service roles and services. These steps will restore App Service secrets and service configurations.

1. Log into the App Service controller **CN0-VM** VM as **roleadmin** using the password you provided during App Service installation.

### TIP

You need to modify the VM's network security group to allow RDP connections.

2. Copy the **SystemSecrets.JSON** file locally to the controller VM. You need to provide the path to this file as the `$pathToExportedSecretFile` parameter in the next step.
3. Use the following commands in an elevated PowerShell console window to restore App Service roles and services:

```
Stop App Service services on the primary controller VM
net stop WebFarmService
net stop ResourceMetering
net stop HostingVssService # This service was deprecated in the App Service 1.5 release and is not required after the App Service 1.4 release.

Restore App Service secrets. Provide the path to the App Service secrets file copied from backup. For example, C:\temp\SystemSecrets.json.
Press ENTER when prompted to reconfigure App Service from backup

If necessary, use -OverrideDatabaseServer <restored server> with Restore-AppServiceStamp when the restored database server has a different address than backed-up deployment.
If necessary, use -OverrideContentShare <restored file share path> with Restore-AppServiceStamp when the restored file share has a different path from backed-up deployment.
Restore-AppServiceStamp -FilePath $pathToExportedSecretFile

Restore App Service roles
Restore-AppServiceRoles

Restart App Service services
net start WebFarmService
net start ResourceMetering
net start HostingVssService # This service was deprecated in the App Service 1.5 release and is not required after the App Service 1.4 release.

After App Service has successfully restarted, and at least one management server is in ready state, synchronize App Service objects to complete the restore
Enter Y when prompted to get all sites and again for all ServerFarm entities.
Get-AppServiceSite | Sync-AppServiceObject
Get-AppServiceServerFarm | Sync-AppServiceObject
```

### TIP

It's highly recommended to close this PowerShell session when the command completes.

## Restore Function Apps

App Service for Azure Stack Hub doesn't support restoring tenant user apps or data other than file share content.

All other data must be backed up and recovered outside of App Service backup and restore operations. If Azure Stack Hub storage was used for Function Apps storage, the following steps should be taken to recover lost data:

1. Create a new storage account to be used by the Function App. This storage can be Azure Stack Hub storage, Azure storage, or any compatible storage.
2. Retrieve the connection string for the storage.
3. Open the function portal and browse to the function app.
4. Browse to the **Platform features** tab and click **Application Settings**.
5. Change **AzureWebJobsDashboard** and **AzureWebJobsStorage** to the new connection string and click **Save**.
6. Switch to **Overview**.
7. Restart the app. It might take several tries to clear all errors.

## Next steps

[App Service on Azure Stack Hub overview](#)

# Azure App Service on Azure Stack Hub billing overview and FAQ

4 minutes to read • [Edit Online](#)

This article shows how cloud operators are billed for offering Azure App Service on Azure Stack Hub and how they can bill their tenants for their use of the service.

## Billing overview

Azure Stack Hub cloud operators choose to deploy the Azure App Service on Azure Stack Hub onto their Azure Stack Hub stamp to offer the tenant capabilities of Azure App Service and Azure Functions to their customers. The Azure App Service resource provider consists of multiple types of roles that can be divided between infrastructure and worker tiers.

Infrastructure roles aren't billed because they're required for the core operation of the service. Infrastructure roles can be scaled out as required to support the demands of the cloud operator's tenants. The infrastructure roles are as follows:

- Controllers
- Management roles
- Publishers
- Front ends

Worker tiers consist of two main types: shared and dedicated. Worker usage is billed to the cloud operator's default provider subscription according to the following criteria.

## Shared workers

Shared workers are multitenant and host free and shared App Service plans and consumption-based Azure functions for many tenants. Shared workers emit usage meters when marked as ready in the Azure App Service resource provider.

## Dedicated workers

Dedicated workers are tied to the App Service plans that tenants create. For example, in the S1 SKU, tenants can scale to 10 instances by default. When a tenant creates an S1 App Service plan, Azure App Service allocates one of the instances in the small worker tier scale set to that tenant's App Service plan. The assigned worker is then no longer available to be assigned to any other tenants. If the tenant chooses to scale the App Service plan to 10 instances, nine more workers are removed from the available pool and are assigned to the tenant's App Service plan.

Meters are emitted for dedicated workers when they're:

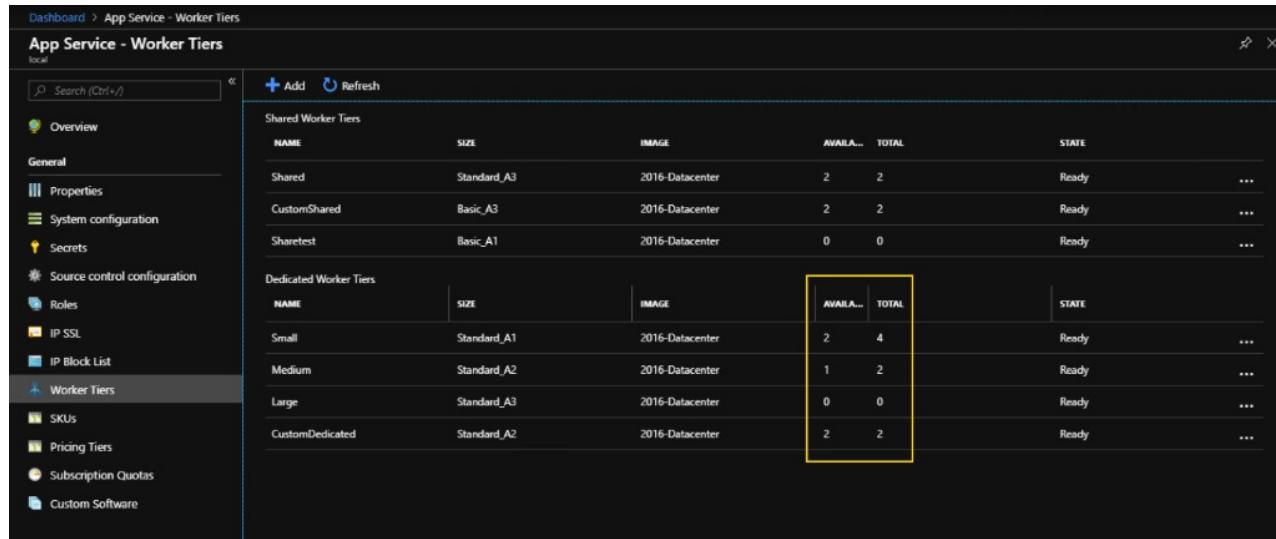
- Marked as ready in the Azure App Service resource provider.
- Assigned to an App Service plan.

This billing model enables cloud operators to provision a pool of dedicated workers ready for customers to use without paying for the workers until they're effectively reserved by their tenant's App Service plan.

For example, say you have 20 workers in the small worker tier. Then if you have five customers that create two S1 App Service plans each, and they each scale the App Service plan up to two instances, you have no workers

available. As a result, there's also no capacity for any of your customers or new customers to scale out or create new App Service plans.

Cloud operators can view the current number of available workers per worker tier by looking at the worker tiers in the Azure App Service configuration on Azure Stack Hub administration.



The screenshot shows the 'App Service - Worker Tiers' page in the Azure Stack Hub administration interface. On the left, a navigation sidebar lists various settings like Overview, Properties, System configuration, Secrets, Source control configuration, Roles, IP SSL, IP Block List, Worker Tiers (selected), SKUs, Pricing Tiers, Subscription Quotas, and Custom Software. The main area displays two tables: 'Shared Worker Tiers' and 'Dedicated Worker Tiers'. The 'Shared Worker Tiers' table has three rows: Shared (Standard\_A3, 2 available, 2 total, Ready), CustomShared (Basic\_A3, 2 available, 2 total, Ready), and Sharetest (Basic\_A1, 0 available, 0 total, Ready). The 'Dedicated Worker Tiers' table has four rows: Small (Standard\_A1, 2 available, 4 total, Ready), Medium (Standard\_A2, 1 available, 2 total, Ready), Large (Standard\_A3, 0 available, 0 total, Ready), and CustomDedicated (Standard\_A2, 2 available, 2 total, Ready). A yellow box highlights the 'Available' and 'Total' columns in the Dedicated Worker Tiers table.

| NAME         | SIZE        | IMAGE           | AVAILA... | TOTAL | STATE |
|--------------|-------------|-----------------|-----------|-------|-------|
| Shared       | Standard_A3 | 2016-Datacenter | 2         | 2     | Ready |
| CustomShared | Basic_A3    | 2016-Datacenter | 2         | 2     | Ready |
| Sharetest    | Basic_A1    | 2016-Datacenter | 0         | 0     | Ready |

| NAME            | SIZE        | IMAGE           | AVAILA... | TOTAL | STATE |
|-----------------|-------------|-----------------|-----------|-------|-------|
| Small           | Standard_A1 | 2016-Datacenter | 2         | 4     | Ready |
| Medium          | Standard_A2 | 2016-Datacenter | 1         | 2     | Ready |
| Large           | Standard_A3 | 2016-Datacenter | 0         | 0     | Ready |
| CustomDedicated | Standard_A2 | 2016-Datacenter | 2         | 2     | Ready |

## See customer usage by using the Azure Stack Hub usage service

Cloud operators can query the [Azure Stack Hub Tenant Resource Usage API](#) to retrieve usage information for their customers. You can find all of the individual meters that App Service emits to describe tenant usage in the [Usage FAQ](#). These meters then are used to calculate the usage per customer subscription to calculate charges.

## Frequently asked questions

### How do I license the SQL Server and file server infrastructure required in the prerequisites?

Licensing for SQL Server and file server infrastructure, required by the Azure App Service resource provider, is covered in the Azure App Service on Azure Stack Hub [Before you get started](#) article.

### The Usage FAQ lists the tenant meters but not the prices for those meters. Where can I find them?

As a cloud operator, you're free to apply your own pricing model to your customers. The usage service provides the usage metering. You can then use the meter quantity to charge your customers based on the pricing model you determine. The ability to set pricing enables operators to differentiate from other Azure Stack Hub operators.

### As a CSP, how can I offer free and shared SKUs for customers to try out the service?

As a cloud operator, you incur costs for offering free and shared SKUs because they're hosted in shared workers. To minimize that cost, you can choose to scale down the shared worker tier to a bare minimum.

For example, to offer free and shared App Service plan SKUs and to offer consumption-based functions, you need a minimum of one A1 instance available. Shared workers are multitenant, so they can host multiple customer apps, each individually isolated and protected by the App Service sandbox. By scaling the shared worker tier in this way, you can limit your outlay to the cost of one vCPU per month.

You can then choose to create a quota, for use in a plan, which only offers free and shared SKUs and limits the number of free and shared App Service plans your customer can create.

## Sample scripts to assist with billing

The Azure App Service team created sample PowerShell scripts to assist with querying the Azure Stack Hub usage service. Cloud operators can use these sample scripts to prepare their own billing for their tenants. The sample scripts are in the [Azure Stack Hub Tools repository](#) in GitHub. The App Service scripts are in the [AppService folder](#)

under Usage.

The sample scripts available are:

- [Get-AppServiceBillingRecords](#): This sample fetches Azure App Service on Azure Stack Hub billing records from the Azure Stack Hub Usage API.
- [Get-AppServiceSubscriptionUsage](#): This sample calculates Azure App Service on Azure Stack Hub usage amounts per subscription. This script calculates usage amounts based on data from the Usage API and the prices provided per meter by the cloud operator.
- [Suspend-UserSubscriptions](#): This sample suspends or enables subscriptions based on usage limits specified by the cloud operator.

## Next steps

- [Azure Stack Hub Tenant Resource Usage API](#)

# App Service on Azure Stack Hub Update 1 release notes

5 minutes to read • [Edit Online](#)

These release notes describe improvements, fixes, and known issues in Azure App Service on Azure Stack Hub Update 1. Known issues are divided into three sections: issues directly related to deployment, issues with the update process, and issues with the build (post-installation).

## IMPORTANT

Apply the 1802 update to your Azure Stack Hub integrated system or deploy the latest Azure Stack Development Kit (ASDK) before deploying Azure App Service.

## Build reference

The App Service on Azure Stack Hub Update 1 build number is **69.0.13698.9**.

## Prerequisites

### IMPORTANT

New deployments of Azure App Service on Azure Stack Hub now require a [three-subject wildcard certificate](#) due to improvements in the way in which SSO for Kudu is handled in Azure App Service. The new subject is **\*.sso.appservice.<region>.<domainname>.<extension>**

Refer to the [Prerequisites for deploying App Service on Azure Stack Hub](#) before beginning deployment.

## New features and fixes

Azure App Service on Azure Stack Hub Update 1 includes the following improvements and fixes:

- **High Availability of Azure App Service** - The Azure Stack Hub 1802 update enabled workloads to be deployed across fault domains, allowing App Service infrastructure to be fault tolerant as it's deployed across fault domains. By default, all new deployments of Azure App Service have this capability. However, for deployments completed prior to Azure Stack Hub 1802 update being applied, refer to the [App Service Fault Domain documentation](#).
- **Deploy in existing virtual network** - Customers can now deploy App Service on Azure Stack Hub within an existing virtual network. Deploying in an existing virtual network enables customers to connect to the SQL Server and file server, required for Azure App Service, over private ports. During deployment, customers can select to deploy in an existing virtual network, however they [must create subnets for use by App Service](#) prior to deployment.
- Updates to **App Service Tenant, Admin, Functions portals and Kudu tools**. Consistent with Azure Stack Hub portal SDK version.
- Updates **Azure Functions runtime to v1.0.11388**.
- **Updates to the following application frameworks and tools:**
  - Added **.NET Core 2.0** support.
  - Added **Node.JS** versions:

- 6.11.2
- 6.11.5
- 7.10.1
- 8.0.0
- 8.1.4
- 8.4.0
- 8.5.0
- 8.7.0
- 8.8.1
- 8.9.0
- Added **NPM** versions:
  - 3.10.10
  - 4.2.0
  - 5.0.0
  - 5.0.3
  - 5.3.0
  - 5.4.2
  - 5.5.1
- Added **PHP** updates:
  - 5.6.32
  - 7.0.26 (x86 and x64)
  - 7.1.12 (x86 and x64)
- Updated **Git for Windows** to v 2.14.1
- Updated **Mercurial** to v4.5.0
- Added support for **HTTPS Only** feature within Custom Domain feature in the App Service user portal.
- Added validation of storage connection in the custom storage picker for Azure Functions.

#### Fixes

- When creating an offline deployment package, customers will no longer receive an access denied error message when opening the folder from the App Service installer.
- Resolved issues when working in the custom domains feature in the App Service user portal.
- Prevent customers from using reserved admin names during setup.
- Enabled App Service deployment with **domain joined** file server.
- Improved retrieval of Azure Stack Hub root certificate in script and added ability to validate the root cert in the App Service installer.
- Fixed incorrect status being returned to Azure Resource Manager when a subscription is deleted that contained resources in the Microsoft.Web namespace.

#### Known issues with the deployment process

- Certificate validation errors.

Some customers have experienced issues when providing certificates to the App Service installer when deploying on an integrated system due to overly restrictive validation in the installer. The App Service

installer has been re-released and customers should [download the updated installer](#). If you continue to experience issues validating certificates with the updated installer, contact support.

- Problem retrieving Azure Stack Hub root certificate from integrated system.

An error in the Get-AzureStackRootCert.ps1 caused customers to fail to retrieve the Azure Stack Hub root certificate when executing the script on a machine that doesn't have the root certificate installed. The script has also now been re-released which resolves the issue. [Download the updated helper scripts here](#). If you continue to experience issues retrieving the root certificate with the updated script, contact support.

#### Known issues with the update process

- There are no known issues for the update of Azure App Service on Azure Stack Hub Update 1.

#### Known issues (post-installation)

- Slot swap doesn't function.

Site slot swap is broken in this release. To restore functionality, complete these steps:

1. Modify the ControllersNSG Network Security Group to **Allow** remote desktop connections to the App Service controller instances. Replace AppService.local with the name of the resource group you deployed App Service in.

```
Add-AzureRmAccount -EnvironmentName AzureStackAdmin

$nsg = Get-AzureRmNetworkSecurityGroup -Name "ControllersNsg" -ResourceGroupName "AppService.local"

$RuleConfig_Inbound_Rdp_3389 = $nsg | Get-AzureRmNetworkSecurityRuleConfig -Name "Inbound_Rdp_3389"

Set-AzureRmNetworkSecurityRuleConfig -NetworkSecurityGroup $nsg `
 -Name $RuleConfig_Inbound_Rdp_3389.Name `
 -Description "Inbound_Rdp_3389" `
 -Access Allow `
 -Protocol $RuleConfig_Inbound_Rdp_3389.Protocol `
 -Direction $RuleConfig_Inbound_Rdp_3389.Direction `
 -Priority $RuleConfig_Inbound_Rdp_3389.Priority `
 -SourceAddressPrefix $RuleConfig_Inbound_Rdp_3389.SourceAddressPrefix `
 -SourcePortRange $RuleConfig_Inbound_Rdp_3389.SourcePortRange `
 -DestinationAddressPrefix $RuleConfig_Inbound_Rdp_3389.DestinationAddressPrefix `
 -DestinationPortRange $RuleConfig_Inbound_Rdp_3389.DestinationPortRange

Commit the changes back to NSG
Set-AzureRmNetworkSecurityGroup -NetworkSecurityGroup $nsg
```

2. Browse to the **CN0-VM** under Virtual Machines in the Azure Stack Hub administrator portal and [click Connect](#) to open a remote desktop session with the controller instance. Use the credentials specified during the deployment of App Service.
3. Start **PowerShell as an Administrator** and execute the following script:

```

Import-Module appservice

$sm = new-object Microsoft.Web.Hosting.SiteManager

if($sm.HostingConfiguration.SlotsPollWorkerForChangeNotificationStatus=$true)
{
 $sm.HostingConfiguration.SlotsPollWorkerForChangeNotificationStatus=$false
 # 'Slot swap mode reverted'
}

Confirm new setting is false
$sm.HostingConfiguration.SlotsPollWorkerForChangeNotificationStatus

Commit Changes
$sm.CommitChanges()

Get-AppServiceServer -ServerType ManagementServer | ForEach-Object Repair-AppServiceServer

```

4. Close the remote desktop session.
5. Revert the ControllersNSG Network Security Group to **Deny** remote desktop connections to the App Service controller instances. Replace AppService.local with the name of the resource group you deployed App Service in.

```

Add-AzureRmAccount -EnvironmentName AzureStackAdmin

$nsg = Get-AzureRmNetworkSecurityGroup -Name "ControllersNsg" -ResourceGroupName "AppService.local"

$RuleConfig_Inbound_Rdp_3389 = $nsg | Get-AzureRmNetworkSecurityRuleConfig -Name "Inbound_Rdp_3389"

Set-AzureRmNetworkSecurityRuleConfig -NetworkSecurityGroup $nsg `
-Name $RuleConfig_Inbound_Rdp_3389.Name `
-Description "Inbound_Rdp_3389" `
-Access Deny `
-Protocol $RuleConfig_Inbound_Rdp_3389.Protocol `
-Direction $RuleConfig_Inbound_Rdp_3389.Direction `
-Priority $RuleConfig_Inbound_Rdp_3389.Priority `
-SourceAddressPrefix $RuleConfig_Inbound_Rdp_3389.SourceAddressPrefix `
-SourcePortRange $RuleConfig_Inbound_Rdp_3389.SourcePortRange `
-DestinationAddressPrefix $RuleConfig_Inbound_Rdp_3389.DestinationAddressPrefix `
-DestinationPortRange $RuleConfig_Inbound_Rdp_3389.DestinationPortRange

Commit the changes back to NSG
Set-AzureRmNetworkSecurityGroup -NetworkSecurityGroup $nsg

```

6. Workers are unable to reach file server when App Service is deployed in an existing virtual network and the file server is only available on the private network.

If you chose to deploy into an existing virtual network and an internal IP address to connect to your file server, you must add an outbound security rule which enables SMB traffic between the worker subnet and the file server. Go to the WorkersNsg in the administrator portal and add an outbound security rule with the following properties:

- Source: Any
- Source port range: \*
- Destination: IP addresses
- Destination IP address range: Range of IPs for your file server
- Destination port range: 445
- Protocol: TCP

- Action: Allow
- Priority: 700
- Name: Outbound\_Allow\_SMB445

#### **Known issues for cloud admins operating Azure App Service on Azure Stack Hub**

Refer to the documentation in the [Azure Stack Hub 1802 Release Notes](#)

## Next steps

- For an overview of Azure App Service, see [Azure App Service on Azure Stack Hub overview](#).
- For more information about how to prepare to deploy App Service on Azure Stack Hub, see [Prerequisites for deploying App Service on Azure Stack Hub](#).

# App Service on Azure Stack Hub Update 2 release notes

2 minutes to read • [Edit Online](#)

These release notes describe improvements, fixes, and known issues in Azure App Service on Azure Stack Hub Update 2. Known issues are divided into three sections: issues directly related to deployment, issues with the update process, and issues with the build (post-installation).

## IMPORTANT

Apply the 1804 update to your Azure Stack Hub integrated system or deploy the latest Azure Stack Development Kit (ASDK) before deploying Azure App Service 1.2.

## Build reference

The App Service on Azure Stack Hub Update 2 build number is **72.0.13698.10**.

## Prerequisites

### IMPORTANT

New deployments of Azure App Service on Azure Stack Hub now require a [three-subject wildcard certificate](#) due to improvements in the way in which SSO for Kudu is handled in Azure App Service. The new subject is: **\*.sso.appservice.<region>.<domainname>.<extension>**

Refer to the [Prerequisites for deploying App Service on Azure Stack Hub](#) before beginning deployment.

## New features and fixes

Azure App Service on Azure Stack Hub Update 2 includes the following improvements and fixes:

- Updates to **App Service Tenant, Admin, Functions portals and Kudu tools**. Consistent with Azure Stack Hub portal SDK version.
- Updates **Azure Functions runtime** to **v1.0.11612**.
- Updates to core service to improve reliability and error messaging enabling easier diagnosis of common issues.
- **Updates to the following application frameworks and tools:**
  - Added .NET Framework 4.7.1
  - Added **Node.JS** versions:
    - NodeJS 6.12.3
    - NodeJS 8.9.4
    - NodeJS 8.10.0
    - NodeJS 8.11.1
  - Added **NPM** versions:
    - 5.6.0
  - Updated .NET Core components to be consistent with Azure App Service in public cloud.
  - Updated Kudu

- Auto swap of deployment slots feature enabled - [Configuring Auto Swap](#).
- Testing in production feature enabled - [Introduction to Testing in Production](#).
- Azure Functions Proxies enabled - [Work with Azure Functions Proxies](#).
- App Service admin extension UX support added for:
  - Secret rotation
  - Certificate rotation
  - System credential rotation
  - Connection string rotation

#### **Known issues (post-installation)**

- Workers are unable to reach file server when App Service is deployed in an existing virtual network and the file server is only available on the private network.

If you chose to deploy into an existing virtual network and an internal IP address to connect to your file server, you must add an outbound security rule which enables SMB traffic between the worker subnet and the file server. Go to the WorkersNsg in the administrator portal and add an outbound security rule with the following properties:

- Source: Any
- Source port range: \*
- Destination: IP addresses
- Destination IP address range: Range of IPs for your file server
- Destination port range: 445
- Protocol: TCP
- Action: Allow
- Priority: 700
- Name: Outbound\_Allow\_SMB445

#### **Known issues for cloud admins operating Azure App Service on Azure Stack Hub**

Refer to the documentation in the [Azure Stack Hub 1804 Release Notes](#)

## Next steps

- For an overview of Azure App Service, see [Azure App Service on Azure Stack Hub overview](#).
- For more information about how to prepare to deploy App Service on Azure Stack Hub, see [Prerequisites for deploying App Service on Azure Stack Hub](#).

# App Service on Azure Stack Hub Update 3 release notes

4 minutes to read • [Edit Online](#)

These release notes describe improvements, fixes, and known issues in Azure App Service on Azure Stack Hub Update 3. Known issues are divided into three sections: issues directly related to deployment, issues with the update process, and issues with the build (post-installation).

## IMPORTANT

Apply the 1807 update to your Azure Stack Hub integrated system or deploy the latest Azure Stack Development Kit (ASDK) before deploying Azure App Service 1.3.

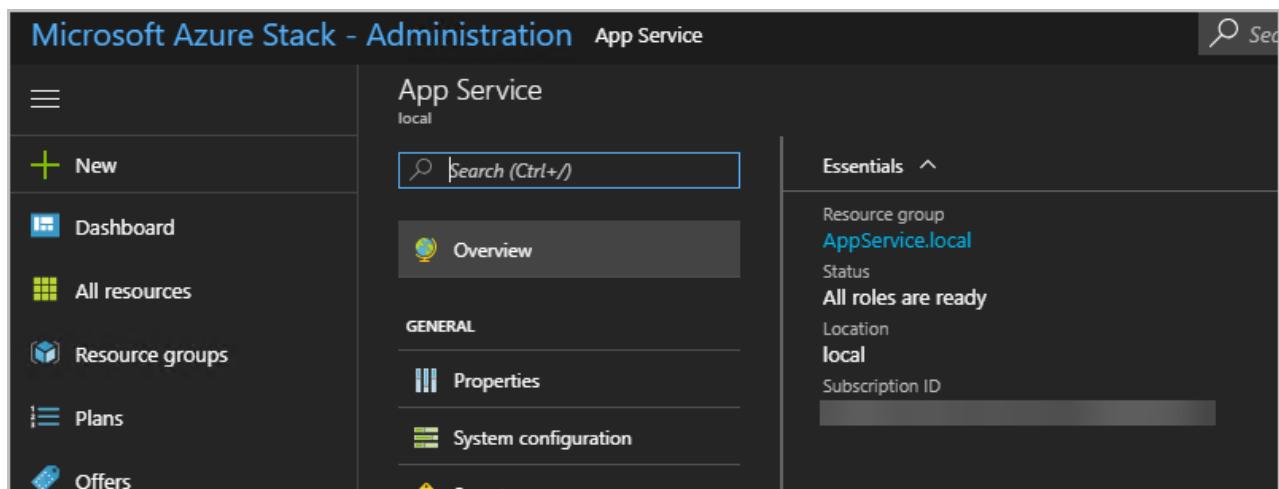
## Build reference

The App Service on Azure Stack Hub Update 3 build number is **74.0.13698.31**.

### Prerequisites

Refer to the [Prerequisites for deploying App Service on Azure Stack Hub](#) before beginning deployment.

Before you begin the upgrade of Azure App Service on Azure Stack Hub to 1.3, ensure all roles are ready in the Azure App Service administration in the Azure Stack Hub administrator portal.



### New features and fixes

Azure App Service on Azure Stack Hub Update 3 includes the following improvements and fixes:

- Support for use of SQL Server Always On for Azure App Service resource provider databases.
- Added new environment parameter to the Create-AADIdentityApp helper script to assist targeting different Azure AD regions.
- Updates to **App Service Tenant, Admin, Functions portals and Kudu tools**. Consistent with Azure Stack Hub portal SDK version.
- Updates **Azure Functions runtime** to **v1.0.11820**.
- Updates to core service to improve reliability and error messaging enabling easier diagnosis of common issues.

- **Updates to the following application frameworks and tools:**

- Added ASP.NET Core 2.1.2
- Added NodeJS 10.0.0
- Added Zulu OpenJDK 8.30.0.1
- Added Tomcat 8.5.31 and 9.0.8
- Added PHP Versions:
  - 5.6.36
  - 7.0.30
  - 7.1.17
  - 7.2.5
- Added Wincache 2.0.0.8
- Updated Git for Windows to v 2.17.1.2
- Updated Kudu to 74.10611.3437

- **Updates to underlying operating system of all roles:**

- [Servicing stack update for Windows Server 2016 for x64-based Systems \(KB4132216\)](#)
- [2018-07 Cumulative Update for Windows Server 2016 for x64-based Systems \(KB4338822\)](#)

#### **Post Update Steps (optional)**

For customers wishing to migrate to a contained database for existing Azure App Service on Azure Stack Hub deployments, execute these steps after the Azure App Service on Azure Stack Hub 1.3 update has completed:

##### **IMPORTANT**

This procedure takes approximately 5-10 minutes. This procedure involves killing the existing database login sessions. Plan for downtime to migrate and validate Azure App Service on Azure Stack Hub post migration

1. Add AppService databases (appservice\_hosting and appservice\_metering) to an Availability group.
2. Enable contained database.

```
sp_configure 'contained database authentication', 1;
GO
RECONFIGURE;
GO
```

3. Converting a database to partially contained. This step will incur downtime as all active sessions need to be killed.

```

***** [appservice_metering] Migration Start*****
USE [master];

-- kill all active sessions
DECLARE @kill varchar(8000) = '';
SELECT @kill = @kill + 'kill ' + CONVERT(varchar(5), session_id) + ';'
FROM sys.dm_exec_sessions
WHERE database_id = db_id('appservice_metering')

EXEC(@kill);

USE [master]
GO
ALTER DATABASE [appservice_metering] SET CONTAINMENT = PARTIAL
GO

*****[appservice_metering] Migration End*****

*****[appservice_hosting] Migration Start*****

-- kill all active sessions
USE [master];

DECLARE @kill varchar(8000) = '';
SELECT @kill = @kill + 'kill ' + CONVERT(varchar(5), session_id) + ';'
FROM sys.dm_exec_sessions
WHERE database_id = db_id('appservice_hosting')

EXEC(@kill);

-- Convert database to contained
USE [master]
GO
ALTER DATABASE [appservice_hosting] SET CONTAINMENT = PARTIAL
GO

*****[appservice_hosting] Migration End*****

...

```

4. Migrate logins to contained database users.

```

IF EXISTS(SELECT * FROM sys.databases WHERE Name=DB_NAME() AND containment = 1)
BEGIN
DECLARE @username sysname ;
DECLARE user_cursor CURSOR
FOR
 SELECT dp.name
 FROM sys.database_principals AS dp
 JOIN sys.server_principals AS sp
 ON dp.sid = sp.sid
 WHERE dp.authentication_type = 1 AND dp.name NOT IN
 ('dbo','sys','guest','INFORMATION_SCHEMA');
OPEN user_cursor
FETCH NEXT FROM user_cursor INTO @username
WHILE @@FETCH_STATUS = 0
BEGIN
 EXECUTE sp_migrate_user_to_contained
 @username = @username,
 @rename = N'copy_login_name',
 @disablelogin = N'do_not_disable_login';
 FETCH NEXT FROM user_cursor INTO @username
END
CLOSE user_cursor ;
DEALLOCATE user_cursor ;
END
GO

```

## Validate

1. Check if SQL Server has containment enabled.

```
sp_configure @configname='contained database authentication'
```

2. Check existing contained behavior.

```
SELECT containment FROM sys.databases WHERE NAME LIKE (SELECT DB_NAME())
```

## Known issues (post-installation)

- Workers are unable to reach file server when App Service is deployed in an existing virtual network and the file server is only available on the private network. This issue is called out in the Azure App Service on Azure Stack Hub deployment documentation.

If you chose to deploy into an existing virtual network and an internal IP address to connect to your file server, you must add an outbound security rule which enables SMB traffic between the worker subnet and the file server. Go to the WorkersNsg in the administrator portal and add an outbound security rule with the following properties:

- Source: Any
- Source port range: \*
- Destination: IP addresses
- Destination IP address range: Range of IPs for your file server
- Destination port range: 445
- Protocol: TCP
- Action: Allow
- Priority: 700
- Name: Outbound\_Allow\_SMB445

## Known issues for cloud admins operating Azure App Service on Azure Stack Hub

Refer to the documentation in the Azure Stack Hub 1807 release notes.

## Next steps

- For an overview of Azure App Service, see [Azure App Service on Azure Stack Hub overview](#).
- For more information about how to prepare to deploy App Service on Azure Stack Hub, see [Prerequisites for deploying App Service on Azure Stack Hub](#).

# App Service on Azure Stack Hub Update 4 release notes

4 minutes to read • [Edit Online](#)

These release notes describe improvements, fixes, and known issues in Azure App Service on Azure Stack Hub Update 4. Known issues are divided into three sections: issues directly related to deployment, issues with the update process, and issues with the build (post-installation).

## IMPORTANT

Apply the 1809 update to your Azure Stack Hub integrated system or deploy the latest Azure Stack Development Kit (ASDK) before deploying Azure App Service 1.4.

## Build reference

The App Service on Azure Stack Hub Update 4 build number is **78.0.13698.5**

### Prerequisites

Refer to the [Prerequisites for deploying App Service on Azure Stack Hub](#) before beginning deployment.

Before you begin the upgrade of Azure App Service on Azure Stack Hub to 1.4:

- Ensure all roles are ready in the Azure App Service administration in the Azure Stack Hub administrator portal.
- Back up the App Service and master databases:
  - AppService\_Hosting;
  - AppService\_Metering;
  - Master
- Back up the tenant app content file share.
- Syndicate the **Custom Script Extension** version **1.9** from Azure Marketplace.

### New features and fixes

Azure App Service on Azure Stack Hub Update 4 includes the following improvements and fixes:

- Resolution for [CVE 2018-8600](#) cross-site scripting (XSS) vulnerability.
- Added support for App Service 2018-02-01 API version.
- Updates to **App Service Tenant, Admin, Functions portals and Kudu tools**. Consistent with Azure Stack Hub portal SDK version.
- Updates **Azure Functions runtime** to **v1.0.11959**.
- Updates to core service to improve reliability and error messaging enabling easier diagnosis of common issues.
- **Updates to the following app frameworks and tools:**
  - Added NodeJS 10.6.0
  - Added NPM 6.1.0

- Added Zulu OpenJDK 8.31.0.2
- Added Tomcat 8.5.34 and 9.0.12
- Added PHP Versions:
  - 5.6.37
  - 7.0.31
  - 7.1.20
  - 7.2.8
- Update to Python versions:
  - 2.7.15
  - 3.6.6
- Updated Git for Windows to v 2.17.1.2
- Updated Kudu to 78.11022.3613

- **Updates to underlying operating system of all roles:**

- [2018-10 Cumulative Update for Windows Server 2016 for x64-based Systems \(KB4462928\)](#)
- Resolved template validation issue when deploying Wordpress, DNN, and Orchard CMS gallery items.
- Resolved configuration issue when Azure Stack Hub rotates the Azure Resource Manager client certificate.
- Restored functionality in the cross-origin resource sharing settings in the App Service user portal.
- Error message is now displayed in App Service administrator portal when the resource provider control plane can't connect to the configured SQL Server instance.
- Ensure endpoint is specified in custom storage connection string when specified in new Function app.

## Post-deployment steps

### IMPORTANT

If you've provided the App Service resource provider with a SQL Always On Instance you *must* [add the appservice\\_hosting and appservice\\_metering databases to an availability group](#) and synchronize the databases to prevent any loss of service in the event of a database failover.

## Post-update steps (optional)

For customers wishing to migrate to a contained database for existing Azure App Service on Azure Stack Hub deployments, execute these steps after the Azure App Service on Azure Stack Hub 1.4 update has completed:

### IMPORTANT

The migration procedure takes approximately 5-10 minutes. The procedure involves killing the existing database login sessions. Plan for downtime to migrate and validate Azure App Service on Azure Stack Hub post migration. If you completed these steps after updating to Azure App Service on Azure Stack Hub 1.3 then these steps aren't required.

1. Add [AppService databases \(appservice\\_hosting and appservice\\_metering\) to an Availability group](#).
2. Enable contained database.

```
sp_configure 'contained database authentication', 1;
GO
RECONFIGURE;
GO
```

3. Converting a database to partially contained, the conversion will incur downtime as all active sessions need to be killed.

```
***** [appservice_metering] Migration Start*****
USE [master];

-- kill all active sessions
DECLARE @kill varchar(8000) = '';
SELECT @kill = @kill + 'kill ' + CONVERT(varchar(5), session_id) + ';' +
FROM sys.dm_exec_sessions
WHERE database_id = db_id('appservice_metering')

EXEC(@kill);

USE [master]
GO
ALTER DATABASE [appservice_metering] SET CONTAINMENT = PARTIAL
GO

*****[appservice_metering] Migration End*****

*****[appservice_hosting] Migration Start*****

-- kill all active sessions
USE [master];

DECLARE @kill varchar(8000) = '';
SELECT @kill = @kill + 'kill ' + CONVERT(varchar(5), session_id) + ';' +
FROM sys.dm_exec_sessions
WHERE database_id = db_id('appservice_hosting')

EXEC(@kill);

-- Convert database to contained
USE [master]
GO
ALTER DATABASE [appservice_hosting] SET CONTAINMENT = PARTIAL
GO

*****[appservice_hosting] Migration End*****
```

4. Migrate logins to contained database users.

```

IF EXISTS(SELECT * FROM sys.databases WHERE Name=DB_NAME() AND containment = 1)
BEGIN
DECLARE @username sysname ;
DECLARE user_cursor CURSOR
FOR
 SELECT dp.name
 FROM sys.database_principals AS dp
 JOIN sys.server_principals AS sp
 ON dp.sid = sp.sid
 WHERE dp.authentication_type = 1 AND dp.name NOT IN
('dbo','sys','guest','INFORMATION_SCHEMA');
OPEN user_cursor
FETCH NEXT FROM user_cursor INTO @username
WHILE @@FETCH_STATUS = 0
BEGIN
 EXECUTE sp_migrate_user_to_contained
 @username = @username,
 @rename = N'copy_login_name',
 @disablelogin = N'do_not_disable_login';
 FETCH NEXT FROM user_cursor INTO @username
END
CLOSE user_cursor ;
DEALLOCATE user_cursor ;
END
GO

```

## Validate

1. Check if SQL Server has containment enabled.

```
sp_configure @configname='contained database authentication'
```

2. Check existing contained behavior.

```
SELECT containment FROM sys.databases WHERE NAME LIKE (SELECT DB_NAME())
```

## Known issues (post-installation)

- Workers are unable to reach file server when App Service is deployed in an existing virtual network and the file server is only available on the private network. This issue is called out in the Azure App Service on Azure Stack Hub deployment documentation.

If you chose to deploy into an existing virtual network and an internal IP address to connect to your file server, you must add an outbound security rule which enables SMB traffic between the worker subnet and the file server. Go to the WorkersNsg in the administrator portal and add an outbound security rule with the following properties:

- Source: Any
- Source port range: \*
- Destination: IP addresses
- Destination IP address range: Range of IPs for your file server
- Destination port range: 445
- Protocol: TCP
- Action: Allow
- Priority: 700
- Name: Outbound\_Allow\_SMB445

## Known issues for cloud admins operating Azure App Service on Azure Stack Hub

Refer to the documentation in the [Azure Stack Hub 1809 Release Notes](#)

## Next steps

- For an overview of Azure App Service, see [Azure App Service on Azure Stack Hub overview](#).
- For more info about how to prepare to deploy App Service on Azure Stack Hub, see [Prerequisites for deploying App Service on Azure Stack Hub](#).

# App Service on Azure Stack Hub Update 5 release notes

4 minutes to read • [Edit Online](#)

These release notes describe improvements, fixes, and known issues in Azure App Service on Azure Stack Hub Update 5. Known issues are divided into three sections: issues directly related to deployment, issues with the update process, and issues with the build (post-installation).

## IMPORTANT

Apply the 1901 update to your Azure Stack Hub integrated system or deploy the latest Azure Stack Development Kit (ASDK) before deploying Azure App Service 1.5.

## Build reference

The App Service on Azure Stack Hub Update 5 build number is **80.0.2.15**.

### Prerequisites

Refer to the [Prerequisites for deploying App Service on Azure Stack Hub](#) before beginning deployment.

Before you begin the upgrade of Azure App Service on Azure Stack Hub to 1.5:

- Ensure all roles are ready in the Azure App Service administration in the Azure Stack Hub administrator portal.
- Back up the App Service and master databases:
  - AppService\_Hosting;
  - AppService\_Metering;
  - Master
- Back up the tenant app content file share.
- Syndicate the **Custom Script Extension** version **1.9.1** from Azure Marketplace.

### New features and fixes

Azure App Service on Azure Stack Hub Update 5 includes the following improvements and fixes:

- Updates to **App Service Tenant, Admin, Functions portals and Kudu tools**. Consistent with Azure Stack Hub Portal SDK version.
- Updates **Azure Functions runtime** to **v1.0.12205**.
- Updates to **Kudu tools** to resolve issues with styling and functionality for customers operating **disconnected** Azure Stack Hub.
- Updates to core service to improve reliability and error messaging enabling easier diagnosis of common issues.
- **Updates to the following app frameworks and tools:**
  - Added ASP.NET Core 2.1.6 and 2.2.0
  - Added NodeJS 10.14.1

- Added NPM 6.4.1
- Updated Kudu to 79.20129.3767

- **Updates to underlying operating system of all roles:**

- [2019-02 Cumulative Update for Windows Server 2016 for x64-based Systems \(KB4487006\)](#)

## Post-deployment Steps

### IMPORTANT

If you've provided the App Service resource provider with a SQL Always On Instance you *must* [add the appservice\\_hosting and appservice\\_metering databases to an availability group](#) and synchronize the databases to prevent any loss of service in the event of a database failover.

## Post-update steps

For customers wishing to migrate to a contained database for existing Azure App Service on Azure Stack Hub deployments, execute these steps after the Azure App Service on Azure Stack Hub 1.5 update has completed:

### IMPORTANT

The migration procedure takes approximately 5-10 minutes. The procedure involves killing the existing database login sessions. Plan for downtime to migrate and validate Azure App Service on Azure Stack Hub post migration. If you completed these steps after updating to Azure App Service on Azure Stack Hub 1.3 then these steps aren't required.

1. Add [AppService databases \(appservice\\_hosting and appservice\\_metering\) to an Availability group.](#)
2. Enable contained database.

```
sp_configure 'contained database authentication', 1;
GO
RECONFIGURE;
GO
```

3. Converting a database to partially contained, the conversion will incur downtime as all active sessions need to be killed.

```

***** [appservice_metering] Migration Start*****
USE [master];

-- kill all active sessions
DECLARE @kill varchar(8000) = '';
SELECT @kill = @kill + 'kill ' + CONVERT(varchar(5), session_id) + ';'
FROM sys.dm_exec_sessions
WHERE database_id = db_id('appservice_metering')

EXEC(@kill);

USE [master]
GO
ALTER DATABASE [appservice_metering] SET CONTAINMENT = PARTIAL
GO

*****[appservice_metering] Migration End*****

*****[appservice_hosting] Migration Start*****

-- kill all active sessions
USE [master];

DECLARE @kill varchar(8000) = '';
SELECT @kill = @kill + 'kill ' + CONVERT(varchar(5), session_id) + ';'
FROM sys.dm_exec_sessions
WHERE database_id = db_id('appservice_hosting')

EXEC(@kill);

-- Convert database to contained
USE [master]
GO
ALTER DATABASE [appservice_hosting] SET CONTAINMENT = PARTIAL
GO

*****[appservice_hosting] Migration End*****
```

4. Migrate logins to contained database users.

```

IF EXISTS(SELECT * FROM sys.databases WHERE Name=DB_NAME() AND containment = 1)
BEGIN
DECLARE @username sysname ;
DECLARE user_cursor CURSOR
FOR
 SELECT dp.name
 FROM sys.database_principals AS dp
 JOIN sys.server_principals AS sp
 ON dp.sid = sp.sid
 WHERE dp.authentication_type = 1 AND dp.name NOT IN
('dbo','sys','guest','INFORMATION_SCHEMA');
OPEN user_cursor
FETCH NEXT FROM user_cursor INTO @username
WHILE @@FETCH_STATUS = 0
BEGIN
 EXECUTE sp_migrate_user_to_contained
 @username = @username,
 @rename = N'copy_login_name',
 @disablelogin = N'do_not_disable_login';
 FETCH NEXT FROM user_cursor INTO @username
END
CLOSE user_cursor ;
DEALLOCATE user_cursor ;
END
GO

```

## Validate

1. Check if SQL Server has containment enabled.

```
sp_configure @configname='contained database authentication'
```

2. Check existing contained behavior.

```
SELECT containment FROM sys.databases WHERE NAME LIKE (SELECT DB_NAME())
```

## Known issues (post-installation)

- Workers are unable to reach file server when App Service is deployed in an existing virtual network and the file server is only available on the private network. This issue is called out in the Azure App Service on Azure Stack Hub deployment documentation.

If you chose to deploy into an existing virtual network and an internal IP address to connect to your file server, you must add an outbound security rule which enables SMB traffic between the worker subnet and the file server. Go to the WorkersNsg in the administrator portal and add an outbound security rule with the following properties:

- Source: Any
- Source port range: \*
- Destination: IP addresses
- Destination IP address range: Range of IPs for your file server
- Destination port range: 445
- Protocol: TCP
- Action: Allow
- Priority: 700
- Name: Outbound\_Allow\_SMB445

## Known issues for cloud admins operating Azure App Service on Azure Stack Hub

Refer to the documentation in the [Azure Stack Hub 1809 release notes](#).

## Next steps

- For an overview of Azure App Service, see [Azure App Service on Azure Stack Hub overview](#).
- For more info on how to prepare to deploy App Service on Azure Stack Hub, see [Prerequisites for deploying App Service on Azure Stack Hub](#).

# App Service on Azure Stack Hub update 6 release notes

3 minutes to read • [Edit Online](#)

These release notes describe the improvements and fixes in Azure App Service on Azure Stack Hub Update 6 and any known issues. Known issues are divided into issues directly related to the deployment, update process, and issues with the build (post-installation).

## IMPORTANT

Apply the 1904 update to your Azure Stack Hub integrated system or deploy the latest Azure Stack Development kit before deploying Azure App Service 1.6.

## Build reference

The App Service on Azure Stack Hub Update 6 build number is **82.0.1.50**

### Prerequisites

Refer to the [Before You Get Started documentation](#) before beginning deployment.

Before you begin the upgrade of Azure App Service on Azure Stack Hub to 1.6:

- Ensure all roles are Ready in the Azure App Service Administration in the Azure Stack Hub Admin Portal
- Back up the App Service and Master Databases:
  - AppService\_Hosting;
  - AppService\_Metering;
  - Master
- Back up the Tenant App content file share
- Syndicate the **Custom Script Extension** version **1.9.1** from the Marketplace

### New features and fixes

Azure App Service on Azure Stack Hub Update 6 includes the following improvements and fixes:

- Updates to **App Service Tenant, Admin, Functions portals and Kudu tools**. Consistent with Azure Stack Hub Portal SDK version.
- Updates **Azure Functions runtime** to **v1.0.12299**.
- Updates to core service to improve reliability and error messaging enabling easier diagnosis of common issues.
- **Updates to the following application frameworks and tools:**
  - ASP.NET Core 2.2.4
  - NodeJS 10.15.2
  - Zulu OpenJDK 8.36.0.1
  - Tomcat 7.0.81
  - Tomcat 8.5.37

- Tomcat 9.0.14
- PHP 5.6.39
- PHP 7.0.33
- PHP 7.1.25
- PHP 7.2.13
- Updated Kudu to 81.10329.3844

- **Updates to underlying operating system of all roles:**

- [2019-04 Cumulative Update for Windows Server 2016 for x64-based Systems \(KB4493473\)](#)

## Post-deployment Steps

### IMPORTANT

If you have provided the App Service resource provider with a SQL Always On Instance you MUST [add the appservice\\_hosting and appservice\\_metering databases to an availability group](#) and synchronize the databases to prevent any loss of service in the event of a database failover.

## Known issues (post-installation)

- Workers are unable to reach file server when App Service is deployed in an existing virtual network and the file server is only available on the private network, as called out in the Azure App Service on Azure Stack Hub deployment documentation.

If you chose to deploy into an existing virtual network and an internal IP address to connect to your file server, you must add an outbound security rule, enabling SMB traffic between the worker subnet and the file server. Go to the WorkersNsg in the Admin Portal and add an outbound security rule with the following properties:

- Source: Any
- Source port range: \*
- Destination: IP Addresses
- Destination IP address range: Range of IPs for your file server
- Destination port range: 445
- Protocol: TCP
- Action: Allow
- Priority: 700
- Name: Outbound\_Allow\_SMB445

## Known issues for Cloud Admins operating Azure App Service on Azure Stack Hub

Refer to the documentation in the [Azure Stack Hub 1908 Release Notes](#)

## Known issues for Tenants deploying applications on Azure App Service on Azure Stack Hub

- Deployment Center is greyed out

Tenants cannot yet make use of Deployment Center, which is a feature that was released in the public cloud in late 2018. Tenants can still use the standard deployment methods (FTP, Web Deploy, Git, etc.) via the portal, CLI, and PowerShell.

- Deployment options (Classic) UX and Deployment credentials portal options not available

In order to reach the deployment options and deployment credentials user experiences in the Azure Stack Hub deployment, tenants should access the portal using this URL format - [https://portal.<region>.<FQDN>/?websitesExtension\\_oladvsts=true](https://portal.<region>.<FQDN>/?websitesExtension_oladvsts=true) - which, for the ASDK would be [https://portal.local.azurestack.external/?websitesExtension\\_oladvsts=true](https://portal.local.azurestack.external/?websitesExtension_oladvsts=true), and then navigate to their applications normally.

- Azure Function Monitoring continually shows "Loading" in the portal

When you attempt to monitor individual Functions, in the user portal, you will see no invocation log, success count, or error count. To re-enable this functionality, go to your **Function App**, go to **Platform Features**, and go to **Application settings**. Add a new app setting - name **AzureWebJobsDashboard** and set the value to the same value as set in `AzureWebJobsStorage`. Then go to the Monitor view on your function and you will see the monitoring information.

## Next steps

- For an overview of Azure App Service, see [Azure App Service on Azure Stack Hub overview](#).
- For more information about how to prepare to deploy App Service on Azure Stack Hub, see [Before you get started with App Service on Azure Stack Hub](#).

# App Service on Azure Stack Hub update 7 release notes

3 minutes to read • [Edit Online](#)

These release notes describe the improvements and fixes in Azure App Service on Azure Stack Hub Update 7 and any known issues. Known issues are divided into issues directly related to the deployment, update process, and issues with the build (post-installation).

## IMPORTANT

Apply the 1907 update to your Azure Stack Hub integrated system or deploy the latest Azure Stack Development kit before deploying Azure App Service 1.7.

## Build reference

The App Service on Azure Stack Hub Update 7 build number is **84.0.2.10**

### Prerequisites

Refer to the [Before You Get Started documentation](#) before beginning deployment.

Before you begin the upgrade of Azure App Service on Azure Stack Hub to 1.7:

- Ensure all roles are Ready in the Azure App Service Administration in the Azure Stack Hub Admin Portal
- Back up the App Service and Master Databases:
  - AppService\_Hosting;
  - AppService\_Metering;
  - Master
- Back up the Tenant App content file share
- Syndicate the **Custom Script Extension** version **1.9.3** from the Marketplace

### New features and fixes

Azure App Service on Azure Stack Hub Update 7 includes the following improvements and fixes:

- Resolution for [CVE-2019-1372](#) Remote Code Execution Vulnerability
- Updates to **App Service Tenant, Admin, Functions portals and Kudu tools**. Consistent with Azure Stack Hub Portal SDK version.
- Updates **Azure Functions runtime** to **v1.0.12582**.
- Updates to core service to improve reliability and error messaging enabling easier diagnosis of common issues.
- **Updates to the following application frameworks and tools:**
  - ASP.NET Core 2.2.46
  - Zul OpenJDK 8.38.0.13
  - Tomcat 7.0.94
  - Tomcat 8.5.42

- Tomcat 9.0.21
- PHP 5.6.40
- PHP 7.3.6
- Updated Kudu to 82.10503.3890

- **Updates to underlying operating system of all roles:**

- [2019-08 Cumulative Update for Windows Server 2016 for x64-based Systems \(KB4512495\)](#)

- **Access Restrictions now enabled in User Portal:**

- As of this release Users can configure Access Restrictions for their Web/Api/Functions applications according to the documentation published - [Azure App Service Access Restrictions](#), **NOTE:** Azure App Service on Azure Stack Hub does not support Service Endpoints.

- **Deployment Options (Classic) Functionality Restored:**

- Users can once again use the Deployment Options (Classic) to configure deployment of their apps from GitHub, Bitbucket, Dropbox, OneDrive, Local and External Repositories, and to set the Deployment Credentials for their applications.

- **Azure Function Monitoring** configured correctly.

- **Windows Update Behavior:** Based on customer feedback we have changed the way in which Windows Update is configured on App Service roles from Update 7:

- Three modes:
  - **Disabled** - Windows Update service disabled, Windows will be updated with the KB that is shipped with Azure App Service on Azure Stack Hub releases;
  - **Automatic** - Windows Update service enabled and Windows Update will determine how and when to update;
  - **Managed** - Windows Update service is disabled, Azure App Service will perform a Windows Update cycle during OnStart of the individual role.

**New** Deployments - Windows Update service is disabled by default.

**Existing** Deployments - If you have modified the setting on the Controller, the value will now change from **False** to **Disabled** and a previous value of **true** will become **Automatic**

### Post-deployment steps

#### IMPORTANT

If you have provided the App Service resource provider with a SQL Always On Instance you MUST [add the appservice\\_hosting and appservice\\_metering databases to an availability group](#) and synchronize the databases to prevent any loss of service in the event of a database failover.

### Known issues (post-installation)

- Workers are unable to reach file server when App Service is deployed in an existing virtual network and the file server is only available on the private network, as called out in the Azure App Service on Azure Stack Hub deployment documentation.

If you chose to deploy into an existing virtual network and an internal IP address to connect to your file server, you must add an outbound security rule, enabling SMB traffic between the worker subnet and the file server. Go to the WorkersNsg in the Admin Portal and add an outbound security rule with the following properties:

- Source: Any
- Source port range: \*
- Destination: IP Addresses

- Destination IP address range: Range of IPs for your file server
- Destination port range: 445
- Protocol: TCP
- Action: Allow
- Priority: 700
- Name: Outbound\_Allow\_SMB445

#### **Known issues for Cloud Admins operating Azure App Service on Azure Stack Hub**

Refer to the documentation in the [Azure Stack Hub 1907 Release Notes](#)

## Next steps

- For an overview of Azure App Service, see [Azure App Service on Azure Stack Hub overview](#).
- For more information about how to prepare to deploy App Service on Azure Stack Hub, see [Before you get started with App Service on Azure Stack Hub](#).

# App Service on Azure Stack Hub update 8 release notes

5 minutes to read • [Edit Online](#)

These release notes describe the improvements and fixes in Azure App Service on Azure Stack Hub Update 8 and any known issues. Known issues are divided into issues directly related to the deployment, update process, and issues with the build (post-installation).

## IMPORTANT

Apply the 1910 update to your Azure Stack integrated system or deploy the latest Azure Stack development kit before deploying Azure App Service 1.8.

## Build reference

The App Service on Azure Stack Hub Update 8 build number is **86.0.2.13**

### Prerequisites

Refer to the [Before You Get Started documentation](#) before beginning deployment.

Before you begin the upgrade of Azure App Service on Azure Stack to 1.8:

- Ensure all roles are Ready in the Azure App Service Administration in the Azure Stack Admin Portal
- Back up the App Service and Master Databases:
  - AppService\_Hosting;
  - AppService\_Metering;
  - Master
- Back up the Tenant App content file share
- Syndicate the **Custom Script Extension** version **1.9.3** from the Marketplace

### New features and fixes

Azure App Service on Azure Stack Update 8 includes the following improvements and fixes:

- Updates to **App Service Tenant, Admin, Functions portals and Kudu tools**. Consistent with Azure Stack Portal SDK version.
- Updates **Azure Functions runtime** to **v1.0.12615**.
- Updates to core service to improve reliability and error messaging enabling easier diagnosis of common issues.
- **Updates to the following application frameworks and tools:**
  - ASP.NET Core 3.1.0
  - ASP.NET Core 3.0.1
  - ASP.NET Core 2.2.8
  - ASP.NET Core Module v2 13.1.19331.0
  - Azul OpenJDK 8.38.0.13

- Tomcat 7.0.94
- Tomcat 8.5.42
- Tomcat 9.0.21
- PHP 7.1.32
- PHP 7.2.22
- PHP 7.3.9
- Updated Kudu to 85.11024.4154
- MSDeploy 3.5.80916.15
- NodeJS 10.16.3
- NPM 6.9.0
- Git for Windows 2.19.1.0

- **Updates to underlying operating system of all roles:**

- [2019-12 Cumulative Update for Windows Server 2016 for x64-based Systems \(KB4530689\)](#)

- **Managed Disk support for new deployments**

All new deployments of Azure App Service on Azure Stack Hub will make use of managed disks for all Virtual Machines and Virtual Machine Scale Sets. All existing deployments will continue to use unmanaged disks.

- **TLS 1.2 Enforced by Front End load balancers**

As of this update **TLS 1.2** will be enforced for all applications.

#### **Known issues (upgrade)**

- Upgrade will fail if SQL Server Always On Cluster has failed over to secondary node

During upgrade, there is a call to check database existence using the master connection string that will fail because the login was on the previous master node.

Take one of the following actions and click retry within the installer.

- Copy the appservice\_hostingAdmin login from the now secondary sql node;

#### **OR**

- Fail over the SQL Cluster to the previous active node.

#### **Post-deployment steps**

##### **IMPORTANT**

If you have provided the App Service resource provider with a SQL Always On Instance you MUST [add the appservice\\_hosting and appservice\\_metering databases to an availability group](#) and synchronize the databases to prevent any loss of service in the event of a database failover.

#### **Known issues (post-installation)**

- Workers are unable to reach file server when App Service is deployed in an existing virtual network and the file server is only available on the private network, as called out in the Azure App Service on Azure Stack deployment documentation.

If you chose to deploy into an existing virtual network and an internal IP address to connect to your file server, you must add an outbound security rule, enabling SMB traffic between the worker subnet and the file server. Go to the WorkersNsg in the Admin Portal and add an outbound security rule with the following properties:

- Source: Any

- Source port range: \*
- Destination: IP Addresses
- Destination IP address range: Range of IPs for your file server
- Destination port range: 445
- Protocol: TCP
- Action: Allow
- Priority: 700
- Name: Outbound\_Allow\_SMB445
- New deployments of Azure App Service on Azure Stack Hub 1.8 require databases to be converted to contained databases

Due to a regression in this release, both App Service databases (appservice\_hosting and appservice\_metering) must be converted to contained databases when **newly** deployed. This **DOES NOT** impact **upgraded** deployments.

#### IMPORTANT

This procedure takes approximately 5-10 minutes. This procedure involves killing the existing database login sessions. Plan for downtime to migrate and validate Azure App Service on Azure Stack Hub post migration

1. Add [AppService databases \(appservice\\_hosting and appservice\\_metering\)](#) to an Availability group.
2. Enable contained database.

```
sp_configure 'contained database authentication', 1;
GO
RECONFIGURE;
GO
```

3. Converting a database to partially contained. This step will incur downtime as all active sessions need to be killed.

```

***** [appservice_metering] Migration Start*****
USE [master];

-- kill all active sessions
DECLARE @kill varchar(8000) = '';
SELECT @kill = @kill + 'kill ' + CONVERT(varchar(5), session_id) + ';'
FROM sys.dm_exec_sessions
WHERE database_id = db_id('appservice_metering')

EXEC(@kill);

USE [master]
GO
ALTER DATABASE [appservice_metering] SET CONTAINMENT = PARTIAL
GO

*****[appservice_metering] Migration End*****

*****[appservice_hosting] Migration Start*****

-- kill all active sessions
USE [master];

DECLARE @kill varchar(8000) = '';
SELECT @kill = @kill + 'kill ' + CONVERT(varchar(5), session_id) + ';'
FROM sys.dm_exec_sessions
WHERE database_id = db_id('appservice_hosting')

EXEC(@kill);

-- Convert database to contained
USE [master]
GO
ALTER DATABASE [appservice_hosting] SET CONTAINMENT = PARTIAL
GO

*****[appservice_hosting] Migration End*****

...

```

4. Migrate logins to contained database users.

```

IF EXISTS(SELECT * FROM sys.databases WHERE Name=DB_NAME() AND containment = 1)
BEGIN
DECLARE @username sysname ;
DECLARE user_cursor CURSOR
FOR
 SELECT dp.name
 FROM sys.database_principals AS dp
 JOIN sys.server_principals AS sp
 ON dp.sid = sp.sid
 WHERE dp.authentication_type = 1 AND dp.name NOT IN
 ('dbo','sys','guest','INFORMATION_SCHEMA');
OPEN user_cursor
FETCH NEXT FROM user_cursor INTO @username
WHILE @@FETCH_STATUS = 0
BEGIN
 EXECUTE sp_migrate_user_to_contained
 @username = @username,
 @rename = N'copy_login_name',
 @disablelogin = N'do_not_disable_login';
 FETCH NEXT FROM user_cursor INTO @username
END
CLOSE user_cursor ;
DEALLOCATE user_cursor ;
END
GO

```

## Validate

5. Check if SQL Server has containment enabled.

```
sp_configure @configname='contained database authentication'
```

6. Check existing contained behavior.

```
SELECT containment FROM sys.databases WHERE NAME LIKE (SELECT DB_NAME())
```

- Unable to scale out workers

New workers are unable to acquire the required database connection string. To remedy this situation, connect to one of your controller instances, for example CN0-VM and run the following PowerShell script:

```

[System.Reflection.Assembly]::LoadWithPartialName("Microsoft.Web.Hosting")
$siteManager=New-ObjectMicrosoft.Web.Hosting.SiteManager
$builder=New-ObjectSystem.Data.SqlClient.SqlConnectionStringBuilder-ArgumentList(Get-
AppServiceConnectionString-TypeHosting)
$conn=New-ObjectSystem.Data.SqlClient.SqlConnection-ArgumentList$builder.ToString()

$siteManager.RoleServers | Where-Object {$_.IsWorker} | ForEach-Object {
 $worker=$_
 $dbUserName="WebWorker_"+$worker.Name

 if(!$siteManager.ConnectionStrings[$dbUserName]){
 $dbUserPassword=[Microsoft.Web.Hosting.Common.Security.PasswordHelper]::GenerateDatabasePassword()
 $conn.Open()
 $command=$conn.CreateCommand()
 $command.CommandText="CREATEUSER[$dbUserName]WITHPASSWORD='$dbUserPassword' "
 $command.ExecuteNonQuery()
 $conn.Close()
 $conn.Open()

 $command=$conn.CreateCommand()
 $command.CommandText="ALTERROLE[WebWorkerRole]ADDMEMBER[$dbUserName]"
 $command.ExecuteNonQuery()
 $conn.Close()

 $builder.Password=$dbUserPassword
 $builder["UserID"]=$dbUserName
 $siteManager.ConnectionStrings.Add($dbUserName,$builder.ToString())
 }
}
$siteManager.CommitChanges()

```

## Known issues for Cloud Admins operating Azure App Service on Azure Stack

Refer to the documentation in the [Azure Stack 1907 Release Notes](#)

## Next steps

- For an overview of Azure App Service, see [Azure App Service on Azure Stack overview](#).
- For more information about how to prepare to deploy App Service on Azure Stack, see [Before you get started with App Service on Azure Stack](#).

# Use MySQL databases on Microsoft Azure Stack Hub

2 minutes to read • [Edit Online](#)

Use the MySQL resource provider to offer MySQL databases on [Azure Stack Hub](#). After you deploy the resource provider and connect it to one or more MySQL server instances, you can create:

- MySQL databases for cloud-native apps.
- MySQL databases for web applications.

There are several limitations to consider, before installing the MySQL resource provider:

- Users can only create and manage individual databases. Database Server instance is not accessible to end users. This may limit compatibility with on-premises database applications that need access to master, Temp DB, or to dynamically manage databases.
- Your Azure Stack Hub operator is responsible for deploying, updating, securing, configuring and maintaining the MySQL database servers and hosts. The RP service does not provide any host and database server instance management functionality.
- Databases from different users in different subscriptions may be located on the same database server instance. The RP does not provide any mechanism for isolating databases on different hosts or database server instances.
- The RP does not provide any reporting on tenant usage of databases.

## MySQL resource provider adapter architecture

The resource provider has the following components:

- **The MySQL resource provider adapter virtual machine (VM)**, which is a Windows Server VM that's running the provider services.
- **The resource provider**, which processes requests and accesses database resources.
- **Servers that host MySQL Server**, which provide capacity for databases that are called hosting servers. You can create MySQL instances yourself, or provide access to external MySQL instances. The [Azure Stack Hub Quickstart Gallery](#) has an example template that you can use to:
  - Create a MySQL server for you.
  - Download and deploy a MySQL Server from Azure Marketplace.

### NOTE

Hosting servers that are installed on Azure Stack Hub integrated systems must be created from a tenant subscription. They can't be created from the default provider subscription. They must be created from the user portal or from a PowerShell session with an appropriate sign-in. All hosting servers are billable VMs and must have licenses. The service administrator can be the owner of the tenant subscription.

## Required privileges

The system account must have the following privileges:

- **Database:** create, drop
- **Login:** create, set, drop, grant, revoke

## Next steps

[Deploy the MySQL resource provider](#)

# Deploy the MySQL resource provider on Azure Stack Hub

7 minutes to read • [Edit Online](#)

Use the MySQL Server resource provider to expose MySQL databases as an Azure Stack Hub service. The MySQL resource provider runs as a service on a Windows Server 2016 Server Core virtual machine (VM).

## IMPORTANT

Only the resource provider is supported to create items on servers that host SQL or MySQL. Items created on a host server that aren't created by the resource provider might result in a mismatched state.

## Prerequisites

There are several prerequisites that need to be in place before you can deploy the Azure Stack Hub MySQL resource provider. To meet these requirements, complete the steps in this article on a computer that can access the privileged endpoint VM.

- If you haven't already, [register Azure Stack Hub](#) with Azure so you can download Azure Marketplace items.
- Add the required Windows Server core VM to Azure Stack Hub Marketplace by downloading the **Windows Server 2016 Datacenter - Server Core** image.
- Download the MySQL resource provider binary and then run the self-extractor to extract the contents to a temporary directory.

## NOTE

To deploy the MySQL provider on a system that doesn't have internet access, copy the [mysql-connector-net-6.10.5.msi](#) file to a local path. Provide the path name using the **DependencyFilesLocalPath** parameter.

- The resource provider has a minimum corresponding Azure Stack Hub build.

| MINIMUM AZURE STACK HUB VERSION | MYSQL RP VERSION                          |
|---------------------------------|-------------------------------------------|
| Version 1910 (1.1910.0.58)      | <a href="#">MySQL RP version 1.1.47.0</a> |
| Version 1808 (1.1808.0.97)      | <a href="#">MySQL RP version 1.1.33.0</a> |
| Version 1808 (1.1808.0.97)      | <a href="#">MySQL RP version 1.1.30.0</a> |
| Version 1804 (1.0.180513.1)     | <a href="#">MySQL RP version 1.1.24.0</a> |
|                                 |                                           |

## IMPORTANT

Before deploying the MySQL resource provider version 1.1.47.0, you should have your Azure Stack Hub system upgraded to 1910 update or later versions. The MySQL resource provider version 1.1.47.0 on previous unsupported Azure Stack Hub versions doesn't work.

- Ensure datacenter integration prerequisites are met:

| PREREQUISITE                                       | REFERENCE                                                                                                                                           |
|----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| Conditional DNS forwarding is set correctly.       | <a href="#">Azure Stack Hub datacenter integration - DNS</a>                                                                                        |
| Inbound ports for resource providers are open.     | <a href="#">Azure Stack Hub datacenter integration - Publish endpoints</a>                                                                          |
| PKI certificate subject and SAN are set correctly. | <a href="#">Azure Stack Hub deployment mandatory PKI prerequisites</a><br><a href="#">Azure Stack Hub deployment PaaS certificate prerequisites</a> |
|                                                    |                                                                                                                                                     |

In a disconnected scenario, complete the following steps to download the required PowerShell modules and register the repository manually.

1. Sign in to a computer with internet connectivity and use the following scripts to download the PowerShell modules.

```
Import-Module -Name PowerShellGet -ErrorAction Stop
Import-Module -Name PackageManagement -ErrorAction Stop

path to save the packages, c:\temp\azs1.6.0 as an example here
$Path = "c:\temp\azs1.6.0"
Save-Package -ProviderName NuGet -Source https://www.powershellgallery.com/api/v2 -Name AzureRM -Path $Path -Force -RequiredVersion 2.3.0
Save-Package -ProviderName NuGet -Source https://www.powershellgallery.com/api/v2 -Name AzureStack -Path $Path -Force -RequiredVersion 1.6.0
```

2. Then you copy the downloaded packages to a USB device.

3. Sign in to the disconnected workstation and copy the packages from the USB device to a location on the workstation.

4. Register this location as a local repository.

```
requires -Version 5
requires -RunAsAdministrator
requires -Module PowerShellGet
requires -Module PackageManagement

$SourceLocation = "C:\temp\azs1.6.0"
$RepoName = "azs1.6.0"

Register-PSRepository -Name $RepoName -SourceLocation $SourceLocation -InstallationPolicy Trusted

New-Item -Path $env:ProgramFiles -name "SqlMySqlPsh" -ItemType "Directory"
```

## Certificates

*For integrated systems installations only.* You must provide the SQL PaaS PKI certificate described in the optional PaaS certificates section of [Azure Stack Hub deployment PKI requirements](#). Place the .pfx file in the location specified by the **DependencyFilesLocalPath** parameter. Don't provide a certificate for ASDK systems.

## Deploy the resource provider

After you've installed all the prerequisites, you can run the **DeployMySqlProvider.ps1** script from a computer that can access both the Azure Stack Hub Admin Azure Resource Management Endpoint and Privileged Endpoint to deploy the MySQL resource provider. The DeployMySqlProvider.ps1 script is extracted as part of the MySQL resource provider installation files that you downloaded for your version of Azure Stack Hub.

### IMPORTANT

Before deploying the resource provider, review the release notes to learn about new functionality, fixes, and any known issues that could affect your deployment.

To deploy the MySQL resource provider, open a new elevated PowerShell window (not PowerShell ISE) and change to the directory where you extracted the MySQL resource provider binary files. We recommend using a new PowerShell window to avoid potential problems caused by PowerShell modules that are already loaded.

Run the **DeployMySqlProvider.ps1** script, which completes the following tasks:

- Uploads the certificates and other artifacts to a storage account on Azure Stack Hub.
- Publishes gallery packages so that you can deploy MySQL databases using the gallery.
- Publishes a gallery package for deploying hosting servers.
- Deploys a VM using the Windows Server 2016 core image you downloaded, and then installs the MySQL resource provider.
- Registers a local DNS record that maps to your resource provider VM.
- Registers your resource provider with the local Azure Resource Manager for the operator account.

### NOTE

When the MySQL resource provider deployment starts, the **system.local.mysqladapter** resource group is created. It may take up to 75 minutes to finish the deployments required to this resource group. You should not place any other resources in the **system.local.mysqladapter** resource group.

### DeployMySqlProvider.ps1 parameters

You can specify these parameters from the command line. If you don't, or if any parameter validation fails, you're prompted to provide the required parameters.

| PARAMETER NAME              | DESCRIPTION                                                                                                                                                                                                                                    | COMMENT OR DEFAULT VALUE |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| <b>CloudAdminCredential</b> | The credential for the cloud administrator, necessary for accessing the privileged endpoint.                                                                                                                                                   | <i>Required</i>          |
| <b>AzCredential</b>         | The credentials for the Azure Stack Hub service admin account. Use the same credentials that you used for deploying Azure Stack Hub. The script will fail if the account you use with AzCredential requires multi-factor authentication (MFA). | <i>Required</i>          |

| Parameter Name                       | Description                                                                                                                                                                                                                                                          | Comment or Default Value                                                        |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| <b>VMLocalCredential</b>             | The credentials for the local administrator account of the MySQL resource provider VM.                                                                                                                                                                               | <i>Required</i>                                                                 |
| <b>PrivilegedEndpoint</b>            | The IP address or DNS name of the privileged endpoint.                                                                                                                                                                                                               | <i>Required</i>                                                                 |
| <b>AzureEnvironment</b>              | The Azure environment of the service admin account used for deploying Azure Stack Hub. Required only for Azure AD deployments. Supported environment names are <b>AzureCloud</b> , <b>AzureUSGovernment</b> , or if using a China Azure AD, <b>AzureChinaCloud</b> . | AzureCloud                                                                      |
| <b>DependencyFilesLocalPath</b>      | For integrated systems only, your certificate .pfx file must be placed in this directory. For disconnected environments, download <a href="#">mysql-connector-net-6.10.5.msi</a> to this directory. You can optionally copy one Windows Update MSU package here.     | <i>Optional (mandatory for integrated systems or disconnected environments)</i> |
| <b>DefaultSSLCertificatePassword</b> | The password for the .pfx certificate.                                                                                                                                                                                                                               | <i>Required</i>                                                                 |
| <b>MaxRetryCount</b>                 | The number of times you want to retry each operation if there's a failure.                                                                                                                                                                                           | 2                                                                               |
| <b>RetryDuration</b>                 | The timeout interval between retries, in seconds.                                                                                                                                                                                                                    | 120                                                                             |
| <b>Uninstall</b>                     | Removes the resource provider and all associated resources (see the following notes).                                                                                                                                                                                | No                                                                              |
| <b>DebugMode</b>                     | Prevents automatic cleanup on failure.                                                                                                                                                                                                                               | No                                                                              |
| <b>AcceptLicense</b>                 | Skips the prompt to accept the GPL license.<br><a href="https://www.gnu.org/licenses/old-licenses/gpl-2.0.html">https://www.gnu.org/licenses/old-licenses/gpl-2.0.html</a>                                                                                           |                                                                                 |

## Deploy the MySQL resource provider using a custom script

If you are deploying the MySQL resource provider version 1.1.33.0 or previous versions, you need to install specific versions of AzureRm.BootStrapper and Azure Stack Hub modules in PowerShell. If you are deploying the MySQL resource provider version 1.1.47.0, the deployment script will automatically download and install the necessary PowerShell modules for you to path C:\Program Files\SqlMySqlPsh.

```
Install the AzureRM.Bootstrapper module, set the profile and install the AzureStack module
Note that this might not be the most currently available version of Azure Stack Hub PowerShell
Install-Module -Name AzureRm.BootStrapper -Force
Use-AzureRmProfile -Profile 2018-03-01-hybrid -Force
Install-Module -Name AzureStack -RequiredVersion 1.6.0
```

**NOTE**

In disconnected scenario, you need to download the required PowerShell modules and register the repository manually as a prerequisite.

To eliminate any manual configuration when deploying the resource provider, you can customize the following script. Change the default account information and passwords as needed for your Azure Stack Hub deployment.

```
Use the NetBIOS name for the Azure Stack Hub domain. On the Azure Stack Hub SDK, the default is AzureStack
but could have been changed at install time.
$domain = "AzureStack"

For integrated systems, use the IP address of one of the ERCS VMs.
$privilegedEndpoint = "AzS-ERCS01"

Provide the Azure environment used for deploying Azure Stack Hub. Required only for Azure AD deployments.
Supported environment names are AzureCloud, AzureUSGovernment, or AzureChinaCloud.
$AzureEnvironment = "<EnvironmentName>"

Point to the directory where the resource provider installation files were extracted.
$tempDir = 'C:\TEMP\MYSQLRP'

The service admin account (can be Azure Active Directory or Active Directory Federation Services).
$serviceAdmin = "admin@mydomain.onmicrosoft.com"
$AdminPass = ConvertTo-SecureString "P@ssw0rd1" -AsPlainText -Force
$AdminCreds = New-Object System.Management.Automation.PSCredential ($serviceAdmin, $AdminPass)

Set the credentials for the new resource provider VM local admin account
$vmLocalAdminPass = ConvertTo-SecureString "P@ssw0rd1" -AsPlainText -Force
$vmLocalAdminCreds = New-Object System.Management.Automation.PSCredential ("mysqlrpadmin", $vmLocalAdminPass)

And the cloudadmin credential required for privileged endpoint access.
$CloudAdminPass = ConvertTo-SecureString "P@ssw0rd1" -AsPlainText -Force
$CloudAdminCreds = New-Object System.Management.Automation.PSCredential ("$domain\cloudadmin",
$CloudAdminPass)

Change the following as appropriate.
$PfxPass = ConvertTo-SecureString "P@ssw0rd1" -AsPlainText -Force

For version 1.1.47.0, the PowerShell modules used by the RP deployment are placed in C:\Program
Files\SqlMySqlPsh,
The deployment script adds this path to the system $env:PSModulePath to ensure correct modules are used.
$rpModulePath = Join-Path -Path $env:ProgramFiles -ChildPath 'SqlMySqlPsh'
$env:PSModulePath = $env:PSModulePath + ";" + $rpModulePath

Change to the directory folder where you extracted the installation files. Don't provide a certificate on
ASDK!
. $tempDir\DeployMySQLProvider.ps1 `
-AzCredential $AdminCreds `
-VMLocalCredential $vmLocalAdminCreds `
-CloudAdminCredential $cloudAdminCreds `
-PrivilegedEndpoint $privilegedEndpoint `
-AzureEnvironment $AzureEnvironment `
-DefaultSSLCertificatePassword $PfxPass `
-DependencyFilesLocalPath $tempDir\cert `
-AcceptLicense
```

When the resource provider installation script finishes, refresh your browser to make sure you can see the latest updates and close the current PowerShell session.

## Verify the deployment by using the Azure Stack Hub portal

1. Sign in to the administrator portal as the service admin.
2. Select **Resource Groups**.
3. Select the **system.<location>.mysqladapter** resource group.
4. On the summary page for Resource group Overview, there should be no failed deployments.
5. Finally, select **Virtual machines** in the administrator portal to verify that the MySQL resource provider VM was successfully created and is running.

## Next steps

[Add hosting servers](#)

# Add MySQL hosting servers in Azure Stack Hub

5 minutes to read • [Edit Online](#)

You can host a MySQL hosting server instance on a virtual machine (VM) in [Azure Stack Hub](#), or on a VM outside your Azure Stack Hub environment, as long as the MySQL resource provider can connect to the instance.

## NOTE

The MySQL resource provider should be created in the default provider subscription while MySQL hosting servers should be created in billable, user subscriptions. The resource provider server shouldn't be used to host user databases.

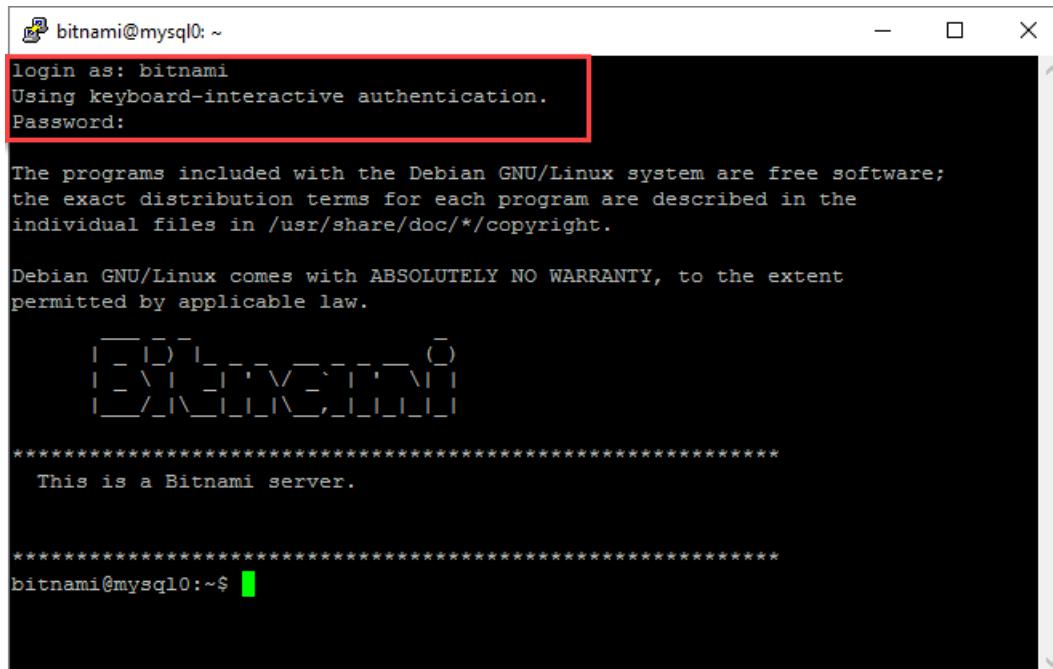
MySQL versions 5.6, 5.7 and 8.0 may be used for your hosting servers. The MySQL RP doesn't support caching\_sha2\_password authentication; that will be added in the next release. MySQL 8.0 servers must be configured to use mysql\_native\_password. MariaDB is also supported.

## Configure external access to the MySQL hosting server

Before the MySQL server can be added as an Azure Stack Hub MySQL Server host, external access must be enabled. Take BitNami MySQL which is available in Azure Stack Hub marketplace as an example, you can take the following steps to configure the external access.

1. Using an SSH client (this example uses [PuTTY](#)) log in to the MySQL server from a computer that can access the public IP.

Use the public IP and log in to the VM with the username of **bitnami** and the application password you created earlier without special characters.



A screenshot of a PuTTY terminal window. The title bar says "bitnami@mysql0: ~". The window shows a command-line interface. A red box highlights the first three lines of text: "login as: bitnami", "Using keyboard-interactive authentication.", and "Password:". Below this, there is a standard Debian Linux welcome message about free software and copyright, followed by a logo consisting of a grid of characters. At the bottom, it says "This is a Bitnami server." and ends with a prompt "bitnami@mysql0:~\$".

2. In the SSH client window, use the following command to ensure the bitnami service is active and running. Provide the bitnami password again when prompted:

```
sudo service bitnami status
```

```
bitnami@mysql0: ~
#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

[sudo] password for bitnami:
● bitnami.service - LSB: Bitnami Init Script
 Loaded: loaded (/etc/init.d/bitnami)
 Active: active (running) since Mon 2018-09-17 18:13:11 UTC; 48min ago
 Process: 38946 ExecStart=/etc/init.d/bitnami start (code=exited, status=0/SUCCESS)
 CGroup: /system.slice/bitnami.service
 ├─39004 /bin/sh ./bin/mysqld_safe --defaults-file=/opt/bitnami/mys...
 ├─39289 /opt/bitnami/mysql/bin/mysqld --defaults-file=/opt/bitnami...
 └─39373 /opt/bitnami/gonit/gonit-linux-x64.run

Sep 17 18:13:03 mysql0 systemd[1]: Starting LSB: Bitnami Init Script...
Sep 17 18:13:04 mysql0 bitnami[38946]: 2018-09-17T18:13:04.161Z - info: Sav...sk
Sep 17 18:13:04 mysql0 bitnami[38946]: 2018-09-17T18:13:04.464Z - info: Per...ql
Sep 17 18:13:10 mysql0 bitnami[38946]: com.bitnami.mysql started
Sep 17 18:13:11 mysql0 bitnami[38946]: undefined
Sep 17 18:13:12 mysql0 bitnami[38946]: 2018-09-17T18:13:11.928Z - info: Sta...ce
Sep 17 18:13:11 mysql0 systemd[1]: Started LSB: Bitnami Init Script.
Hint: Some lines were ellipsized, use -l to show in full.
bitnami@mysql0:~$
```

3. Create a remote access user account to be used by the Azure Stack Hub MySQL Hosting Server to connect to MySQL and then exit the SSH client.

Run the following commands to log in to MySQL as root, using the root password created earlier. Create a new admin user and replace <username> and <password> as required for your environment. In this example, the created user is named **sqlsa** and a strong password is used:

```
mysql -u root -p
create user <username>'%' identified by '<password>';
grant all privileges on *.* to <username>'%' with grant option;
flush privileges;
```

```
bitnami@mysql0: ~
bitnami@mysql0:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 5.7.22-log MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create user sqlsa@'%' identified by '████████';
Query OK, 0 rows affected (0.10 sec)

mysql> grant all privileges on *.* to sqlsa@'%' with grant option;
Query OK, 0 rows affected (0.10 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.05 sec)

mysql>
```

4. Record the new MySQL user information.

This username and password will be used while Azure Stack Hub operator create a MySQL hosting server using this MySQL server.

## Connect to a MySQL hosting server

Make sure you have the credentials for an account with system admin privileges.

**NOTE**

For MySQL 8.0 and above versions, the remote access isn't enabled by default. You need to create a new user account and grant the previledge of remote access to this user account before adding it as a hosting server.

To add a hosting server, follow these steps:

1. Sign in to the Azure Stack Hub administrator portal as a service admin.
2. Select **All services**.
3. Under the **ADMINISTRATIVE RESOURCES** category, select **MySQL Hosting Servers** > **+Add**. The **Add a MySQL Hosting Server** dialog will open, shown in the following screen capture.

## Add a MySQL Hosting Server

\* MySQL Hosting Server name

\* Username

\* Password

Size of Hosting Server in GB

Subscription

\* Resource group  Create new  Use existing

\* Location

\* SKUs  
 >

**Create**

4. Provide the connection details of your MySQL Server instance.

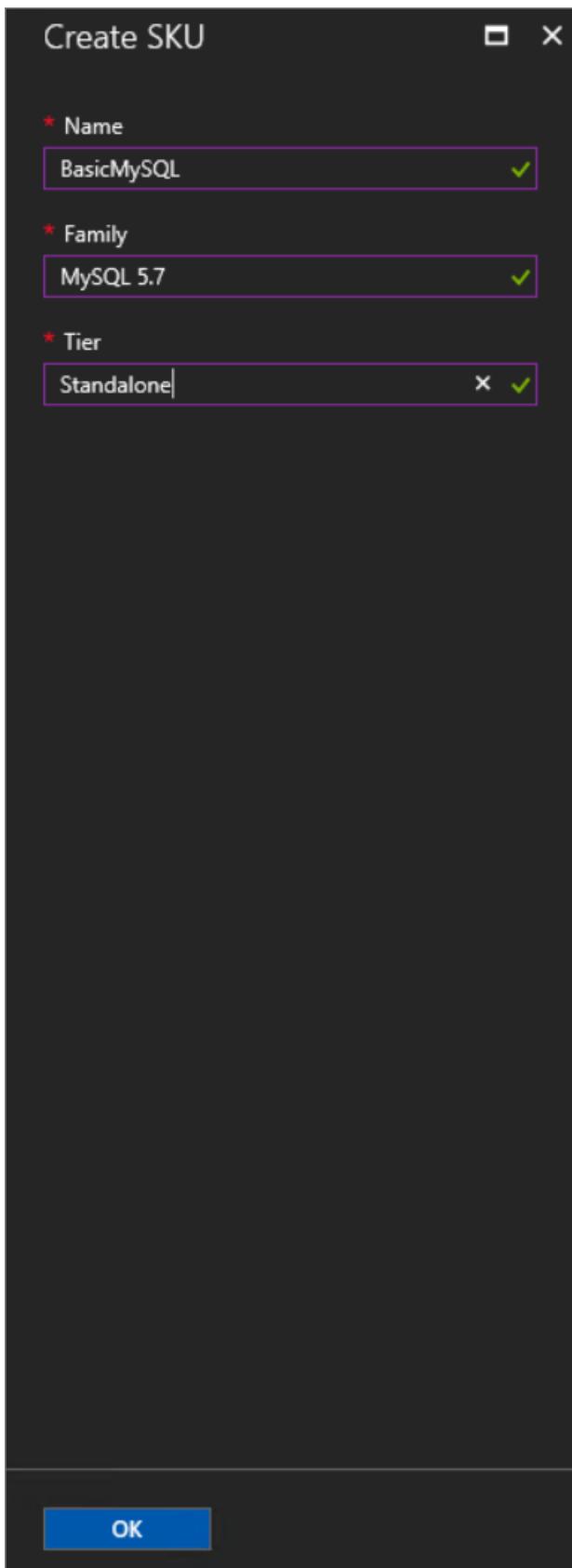
- For **MySQL Hosting Server Name**, provide the fully qualified domain name (FQDN) or a valid IPv4 address. Don't use the short VM name.
- The default admin **Username** for the Bitnami MySQL images available in Azure Stack Hub Marketplace is *root*.
- If you don't know the root **Password**, see the [Bitnami documentation](#) to learn how to get it.
- A default MySQL instance isn't provided, so you have to specify the **Size of Hosting Server in GB**. Enter a size that's close to the capacity of the database server.

- Keep the default setting for **Subscription**.
- For **Resource group**, create a new one, or use an existing group.

**NOTE**

If the MySQL instance can be accessed by the tenant and the admin Azure Resource Manager, you can put it under the control of the resource provider. But, the MySQL instance **must** be allocated exclusively to the resource provider.

5. Select **SKUs** to open the **Create SKU** dialog.



The SKU **Name** should reflect the properties of the SKU so users can deploy their databases to the appropriate SKU.

6. Select **OK** to create the SKU.

**NOTE**

SKUs can take up to an hour to be visible in the portal. You can't create a database until the SKU is deployed and running.

7. Under **Add a MySQL Hosting Server**, select **Create**.

As you add servers, assign them to a new or existing SKU to differentiate service offerings. For example, you can have a MySQL enterprise instance that provides increased database and automatic backups. You can reserve this high-performance server for different departments in your organization.

## Security considerations for MySQL

The following information applies to the RP and MySQL hosting servers:

- Ensure that all hosting servers are configured for communication using TLS 1.1. See [Configuring MySQL to Use Encrypted Connections](#).
- Employ [Transparent Data Encryption](#).
- The MySQL RP doesn't support caching\_sha2\_password authentication.

## Increase backend database capacity

You can increase backend database capacity by deploying more MySQL servers in the Azure Stack Hub portal. Add these servers to a new or existing SKU. If you add a server to an existing SKU, make sure the server characteristics are the same as the other servers in the SKU.

## SKU notes

Use a SKU name that describes the capabilities of the servers in the SKU, such as capacity and performance. The name serves as an aid to help users deploy their databases to the appropriate SKU. For example, you can use SKU names to differentiate service offerings by the following characteristics:

- high capacity
- high performance
- high availability

As a best practice, all the hosting servers in a SKU should have the same resource and performance characteristics.

SKUs can't be assigned to specific users or groups.

To edit a SKU, go to **All services > MySQL Adapter > SKUs**. Select the SKU to modify, make any necessary changes, and click **Save** to save changes.

To delete a SKU that's no longer needed, go to **All services > MySQL Adapter > SKUs**. Right-click the SKU name and select **Delete** to delete it.

**IMPORTANT**

It can take up to an hour for new SKUs to be available in the user portal.

# Make MySQL database servers available to your users

Create plans and offers to make MySQL database servers available to users. Add the Microsoft.MySqlAdapter service to the plan and create a new quota. MySQL doesn't allow limiting the size of databases.

## IMPORTANT

It can take up to two hours for new quotas to be available in the user portal or before a changed quota is enforced.

## Next steps

[Create a MySQL database](#)

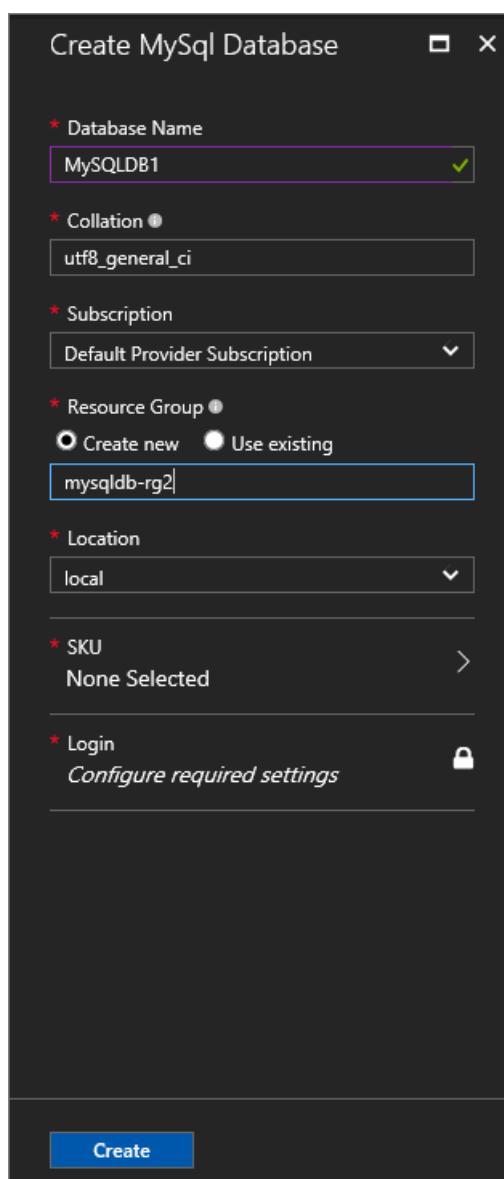
# Create MySQL databases in Azure Stack Hub

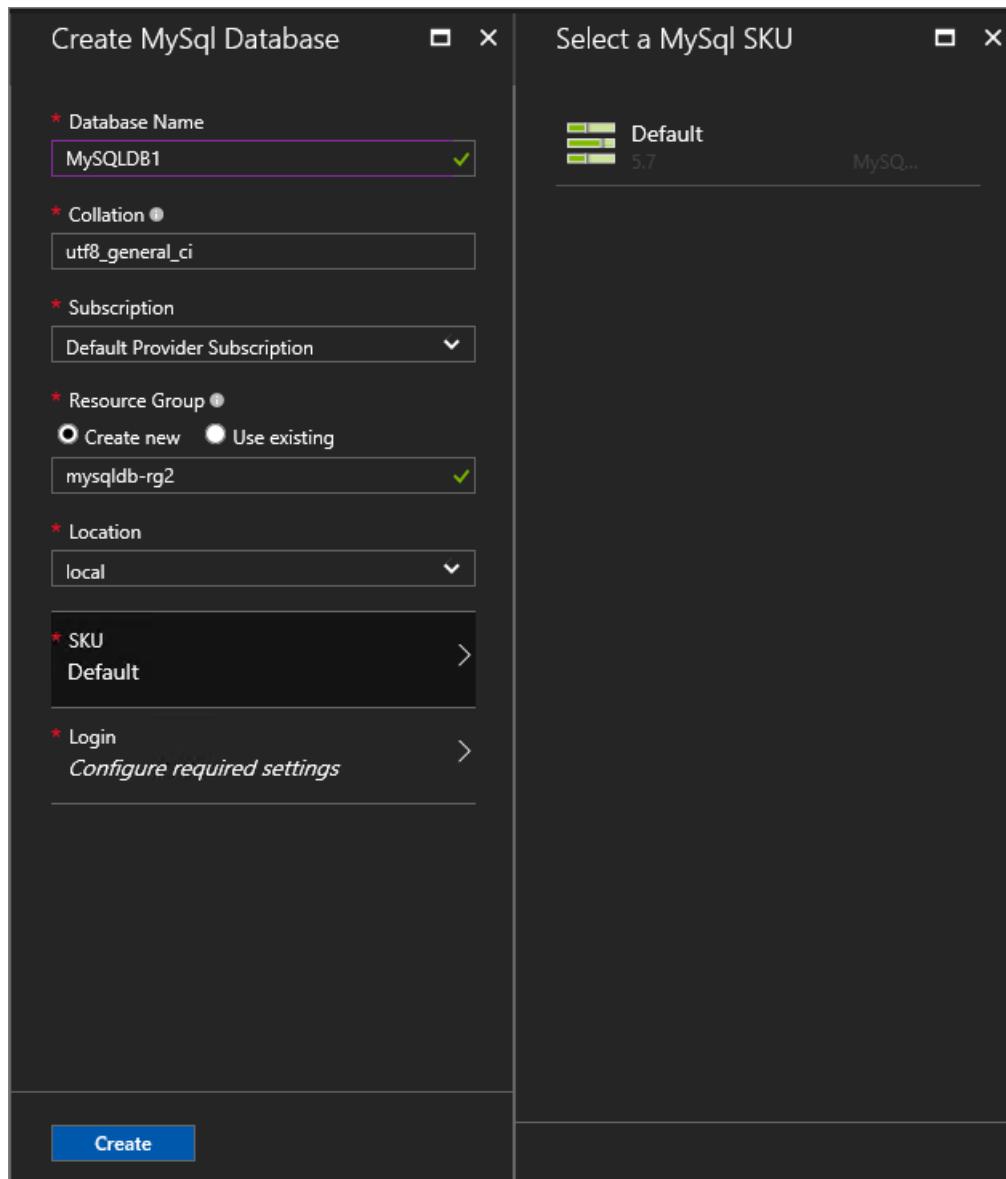
2 minutes to read • [Edit Online](#)

An Azure Stack Hub user that's subscribed to an offer that includes the MySQL database service can create and manage self-service MySQL databases in the user portal.

## Create a MySQL database

1. Sign in to the Azure Stack Hub user portal.
2. Select + **Create a resource** > **Data** > **Storage** > **MySQL Database** > **Add**.
3. Under **Create MySQL Database**, enter the Database Name, and configure the other settings as required for your environment.

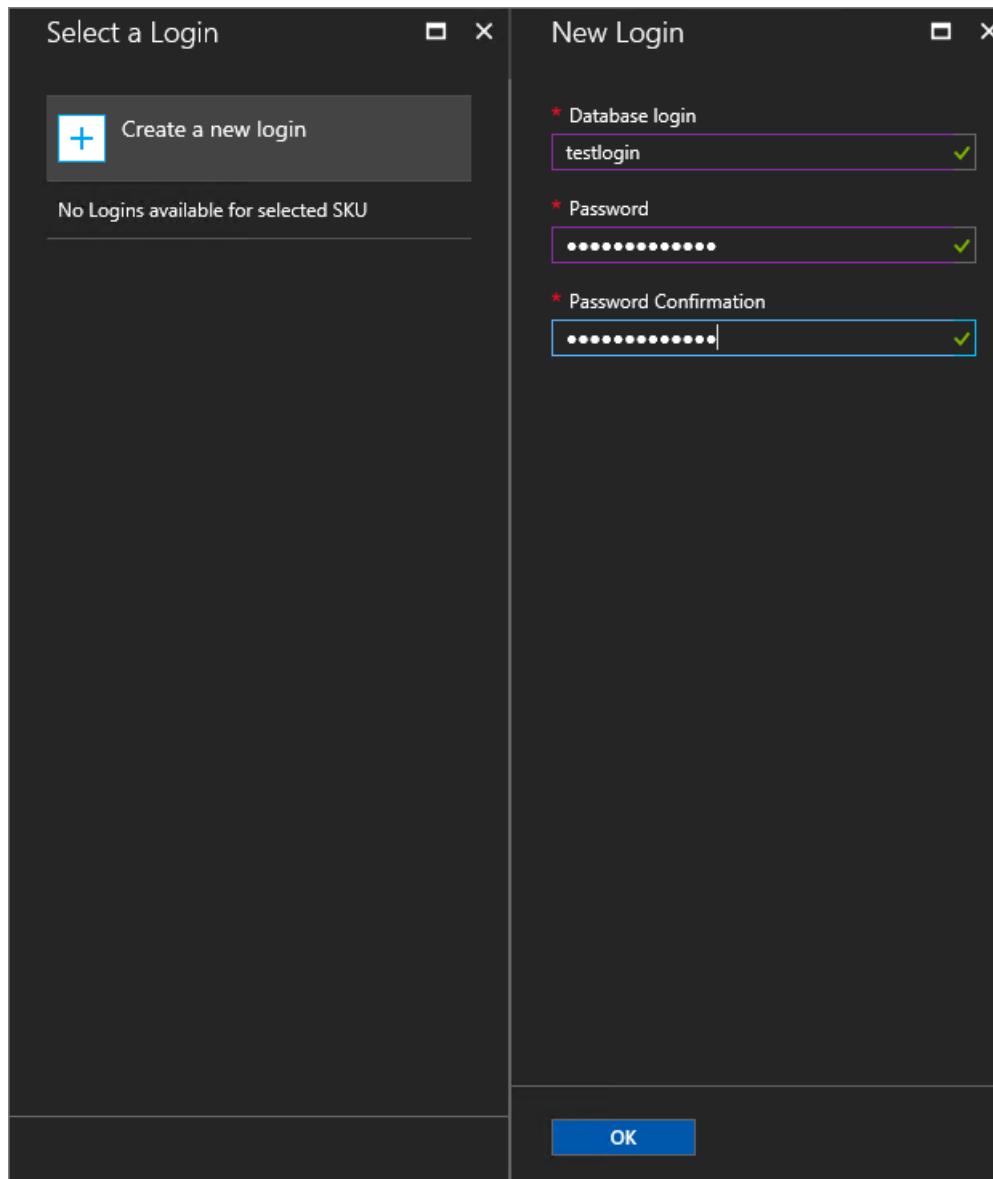




#### NOTE

As hosting servers are added to Azure Stack Hub, they're assigned a SKU. Databases are created in the pool of hosting servers in a SKU.

5. Under **Login**, select **Configure required settings**.
6. Under **Select a Login**, you can choose an existing login or select + **Create a new login** to set up a new login. Enter a **Database login** name and **Password**, and then select **OK**.



#### NOTE

The length of the Database login name can't exceed 32 characters in MySQL 5.7. In earlier editions, it can't exceed 16 characters.

7. Select **Create** to finish setting up the database.

After the database is deployed, take note of the **Connection String** under **Essentials**. You can use this string in any application that needs to access the MySQL database.

The screenshot shows the Azure portal interface for managing a MySQL database. On the left, there's a sidebar with 'MySQLDB1' and 'MySQL Database'. Below it are sections for 'Essentials' (Resource group: mysqladb-rg2, Location: local, Subscription name: Default Provider Subscription, Subscription ID: <Subscription ID>), 'Name' (MySQLDB1), 'Collation' (utf8\_general\_ci), and 'Connection String' (server=192.168.102.22;password=\*\*\*\*\*...). A red box highlights the 'Connection String' field. On the right, there's a 'Settings' pane with sections for 'SUPPORT + TROUBLESHOOTING' (Activity log, New support request), 'GENERAL' (Properties), 'RESOURCE MANAGEMENT' (Tags, Locks, Users, Automation script), and a search bar at the top.

## Update the administrative password

You can modify the password by changing it on the MySQL server instance.

1. Select **ADMINISTRATIVE RESOURCES** > **MySQL Hosting Servers**. Select the hosting server.
2. Under **Settings**, select **Password**.
3. Under **Password**, enter the new password and then select **Save**.

The screenshot shows the 'Password' settings dialog. It has a 'Save' and 'Discard' button at the top. Below is a form with fields for 'Username' (root) and 'Password' (\*\*\*\*\*). The left side of the dialog shows a 'Settings' pane with 'GENERAL' (Properties, Settings, Password selected), 'SUPPORT + TROUBLESHOOTING' (Activity log, New support request), and a 'Filter settings' input field.

## Next steps

Learn how to [offer highly available MySQL databases](#).

# Create highly available MySQL databases

8 minutes to read • [Edit Online](#)

As an Azure Stack Hub operator, you can configure server virtual machines (VMs) to host MySQL Server databases. After a MySQL cluster is successfully created and managed by Azure Stack Hub, users who have subscribed to MySQL services can easily create highly available MySQL databases.

This article shows how to use Azure Stack Marketplace items to create a [MySQL with replication cluster](#). This solution uses multiple VMs to replicate the databases from the master node to a configurable number of replicas. Once created, the cluster can then be added as an Azure Stack Hub MySQL Hosting Server, and then users can create highly available MySQL databases.

## IMPORTANT

The **MySQL with replication** Azure Stack Marketplace item might not be available for all Azure cloud subscription environments. Verify that the marketplace item is available in your subscription before attempting to follow the rest of this tutorial.

What you'll learn:

- Create a MySQL Server cluster from marketplace items.
- Create an Azure Stack Hub MySQL Hosting Server.
- Create a highly available MySQL database.

A three-VM MySQL Server cluster will be created and configured using available Azure Stack Marketplace items.

Before starting, ensure that the [MySQL Server resource provider](#) has been successfully installed and that the following items are available in Azure Stack Marketplace:

## IMPORTANT

All of the following are required to create the MySQL cluster.

- [MySQL with Replication](#): This is the Bitnami solution template that will be used for the MySQL cluster deployment.
- [Debian 8 "Jessie"](#): Debian 8 "Jessie" with backports kernel for Microsoft Azure provided by credativ. Debian GNU/Linux is one of the most popular Linux distributions.
- [Custom script for linux 2.0](#): Custom Script Extension is a tool to execute your VM customization tasks post VM provision. When this Extension is added to a VM, it can download scripts from Azure storage and run them on the VM. Custom Script Extension tasks can also be automated using the Azure PowerShell cmdlets and Azure Cross-Platform Command-Line Interface (xPlat CLI).
- [VM Access For Linux Extension 1.4.7](#): The VM Access extension enables you to reset the password, SSH key, or the SSH configurations so you can regain access to your VM. You can also add a new user with password or SSH key, or delete a user using this extension. This extension targets Linux VMs.

To learn more about adding items to Azure Stack Marketplace, see the [Azure Stack Marketplace overview](#).

You'll also need an SSH client like [PuTTY](#) to log in to the Linux VMs after they're deployed.

## Create a MySQL Server cluster

Use the steps in this section to deploy the MySQL Server cluster using the [MySQL with Replication](#) marketplace item. This template deploys three MySQL Server instances configured in a highly available MySQL cluster. By default, it creates the following resources:

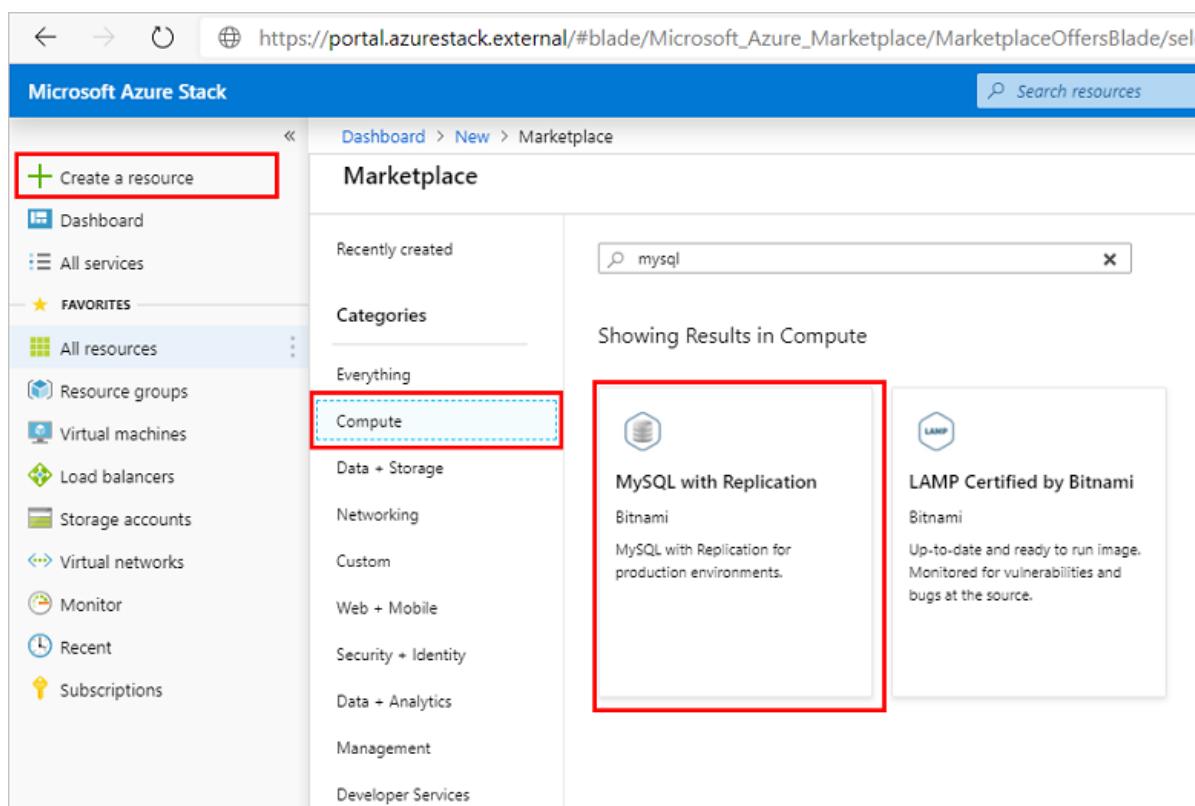
- A virtual network
- A network security group
- A storage account
- An availability set
- Three network interfaces (one for each of the default VMs)
- A public IP address (for the primary MySQL cluster VM)
- Three Linux VMs to host the MySQL cluster

1. Sign in to the user portal:

- For an integrated system deployment, the portal address will vary based on your solution's region and external domain name. It will be in the format of <https://portal.<region>.<FQDN>>.
- For the Azure Stack Development Kit (ASDK), the portal address is <https://portal.local.azurestack.external>.

2. If no subscriptions were assigned yet, select **Get a Subscription** from the Dashboard. In the blade, type a name for the subscription, and then select an offer. It is recommended that you keep the MySQL cluster deployment in its own subscription to prevent accidental removal.

3. Select **+ Create a resource > Compute**, and then **MySQL with Replication**.



4. Provide basic deployment information on the **Basics** page. Review the default values and change as needed and select **OK**.

At a minimum, provide the following info:

- Deployment name (default is mymysql).
- Application root password. Provide a 12 character alphanumeric password with **no special characters**.

- Application database name (default is bitnami).
- Number of MySQL database replica VMs to create (default is 2).
- Select the subscription to use.
- Select the resource group to use or create a new one.
- Select the location (default is local for ASDK).

Dashboard > New > Marketplace > MySQL with Replication > Create MySQL with Replication > Basics

### Create MySQL with Replication

**Basics**

1 Basics >  
Configure basic settings

2 Environment Configuration >  
Provide Environment Configuration

3 Summary >  
MySQL with Replication

4 Buy >

**Basics**

\* Deployment name

**Application**

\* Application password (user name is "root")  
 ✓

\* Confirm password  
 ✓

**Database**

\* Application database

Number of slave machines to create

Subscription

5. On the **Environment Configuration** page, provide the following information and then select **OK**:

- Password or SSH public key to use for secure shell (SSH) authentication. If using a password, it must contain letters, numbers, and **can** contain special characters.
- VM size (default is Standard D1 v2 VMs).
- Data disk size in GB

Dashboard > New > Marketplace > MySQL with Replication > Create MySQL with Replication > Environment information

### Create MySQL with Replication

**Environment information**

1 Basics Done ✓

2 Environment Configuration >  
Provide Environment Configuration

3 Summary >  
MySQL with Replication

4 Buy >

**Environment information**

\* Authentication type  
 Password  SSH public key

\* Password  ✓

\* Confirm password  
 ✓

\* VM size

\* Data disk size (GB)

6. Review the deployment **Summary**. Optionally, you can download the customized template and parameters and then select **OK**.

| Create MySQL with Replication |                                   | Summary                              |
|-------------------------------|-----------------------------------|--------------------------------------|
| <b>1</b>                      | Basics<br>Done                    | <span style="color: green;">✓</span> |
| <b>2</b>                      | Environment Configuration<br>Done | <span style="color: green;">✓</span> |
| <b>3</b>                      | Summary<br>MySQL with Replication | >                                    |
| <b>4</b>                      | Buy                               | >                                    |

**i** Validation passed

**Basics**

Subscription: MySqlTarget  
Resource group: mysql-ha  
Location: Inv5

Deployment name: mymysql  
Application password (user n...): \*\*\*\*  
Application database: bitnami  
Number of slave machines t...: 2

Environment information

Password: \*\*\*\*  
VM size: Standard D1 v2  
Data disk size (GB): 50

7. Select **Create** on the **Buy** page to start the deployment.

| Create MySQL with Replication |                                   | Create                               |
|-------------------------------|-----------------------------------|--------------------------------------|
| <b>1</b>                      | Basics<br>Done                    | <span style="color: green;">✓</span> |
| <b>2</b>                      | Environment Configuration<br>Done | <span style="color: green;">✓</span> |
| <b>3</b>                      | Summary<br>MySQL with Replication | <span style="color: green;">✓</span> |
| <b>4</b>                      | Buy                               | >                                    |

**MySQL with Replication**  
by Bitnami  
[Terms of use](#) | [privacy policy](#)

Deploying this template will result in various actions being performed, which may include the deployment of one or more Azure resources or Marketplace offerings and/or transmission of the information you provided as part of the deployment process to one or more parties, as specified in the template. You are responsible for reviewing the text of the template to determine which actions will be performed and which resources or offerings will be deployed, and for locating and reviewing the pricing and legal terms associated with those resources or offerings.

Current retail prices for Azure resources are set forth [here](#) and may not reflect discounts applicable to your Azure subscription.

Prices for Marketplace offerings are set forth [here](#), and the legal terms associated with any Marketplace offering may be found in the Azure portal; both are subject to change at any time prior to deployment.

Neither subscription credits nor monetary commitment funds may be used to purchase non-Microsoft offerings. These purchases are billed separately. If any Microsoft products are included in a Marketplace offering (e.g., Windows Server or SQL Server), such products are licensed by Microsoft and not by any third party.

**Template deployment is intended for advanced users only.** If you are uncertain which actions will be performed by this template, which resources or offerings will be deployed, or what prices or legal terms pertain to those resources or offerings, do not deploy this template.

**Terms of use**

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) provided above as well as the legal terms and privacy statement(s) associated with each Marketplace offering that will be deployed using this template, if any; (b) authorize Microsoft to charge or bill my current payment method for the fees associated with my use of the offering(s), including applicable taxes, with the same billing frequency as my Azure subscription, until I discontinue use of the offering(s); and (c) agree that Microsoft may share my contact information and transaction details with any third-party sellers of the offering(s). Microsoft assumes no responsibility for any actions performed by third-party templates and does not provide rights for third-party products or services. See the [Azure Marketplace Terms](#) for additional terms.

**Create**

#### NOTE

The deployment will take about an hour. Ensure that the deployment has finished and the MySQL cluster has been completely configured before continuing.

8. After all deployments have completed successfully, review the resource group items and select the **mysqlip** Public IP address item. Record the public IP address and full FQDN of the public IP for the cluster.

You'll need to provide this IP address to an Azure Stack Hub operator so they can create a MySQL hosting server leveraging this MySQL cluster.

#### Create a network security group rule

By default, no public access is configured for MySQL into the host VM. For the Azure Stack Hub MySQL resource provider to connect and manage the MySQL cluster, an inbound network security group (NSG) rule needs to be created.

1. In the administrator portal, go to the resource group created when deploying the MySQL cluster and select the network security group (**default-subnet-sg**):

The screenshot shows the Azure Stack Hub Resource Group Overview page for a resource group named 'MySqlTarget'. The left sidebar lists various navigation options: Overview, Activity log, Access control (IAM), Tags, Quickstart, Deployments, Properties, Locks, Monitoring, and Metrics. The main content area displays the resource group details, including the Subscription ID and Tags. Below this is a list of resources, with one item, 'MySqlTargetLnv5-SSG', highlighted by a red box. The table columns are NAME, TYPE, and LOCATION.

| NAME                       | TYPE                   | LOCATION |
|----------------------------|------------------------|----------|
| MySQLTargetLnv5            | Virtual machine        | Inv5     |
| MySQLTargetLnv5-NIC        | Network interface      | Inv5     |
| MySQLTargetLnv5-PublicIP   | Public IP address      | Inv5     |
| <b>MySQLTargetLnv5-SSG</b> | Network security group | Inv5     |
| MySQLTargetLnv5-VNET       | Virtual network        | Inv5     |
| mysqltargetlsvbstdiba3qvi  | Storage account        | Inv5     |

2. Select **Inbound security rules** and then select **Add**.

Enter **3306** in the **Destination port range** and optionally provide a description in the **Name** and **Description** fields.

The screenshot shows the Azure portal interface for managing network security groups. On the left, the 'Inbound security rules' section is selected. A red box highlights the 'Add' button. On the right, a detailed configuration dialog for an 'Inbound security rule' is displayed. The 'Basic' tab is selected. The 'Destination port ranges' field is set to '3306' and is also highlighted with a red box. The 'Name' field is set to 'MySQL\_3306'. The 'Add' button at the bottom of the dialog is also highlighted with a red box.

3. Select **Add** to close the inbound security rule dialog.

#### Configure external access to the MySQL cluster

Before the MySQL cluster can be added as an Azure Stack Hub MySQL Server host, external access must be enabled.

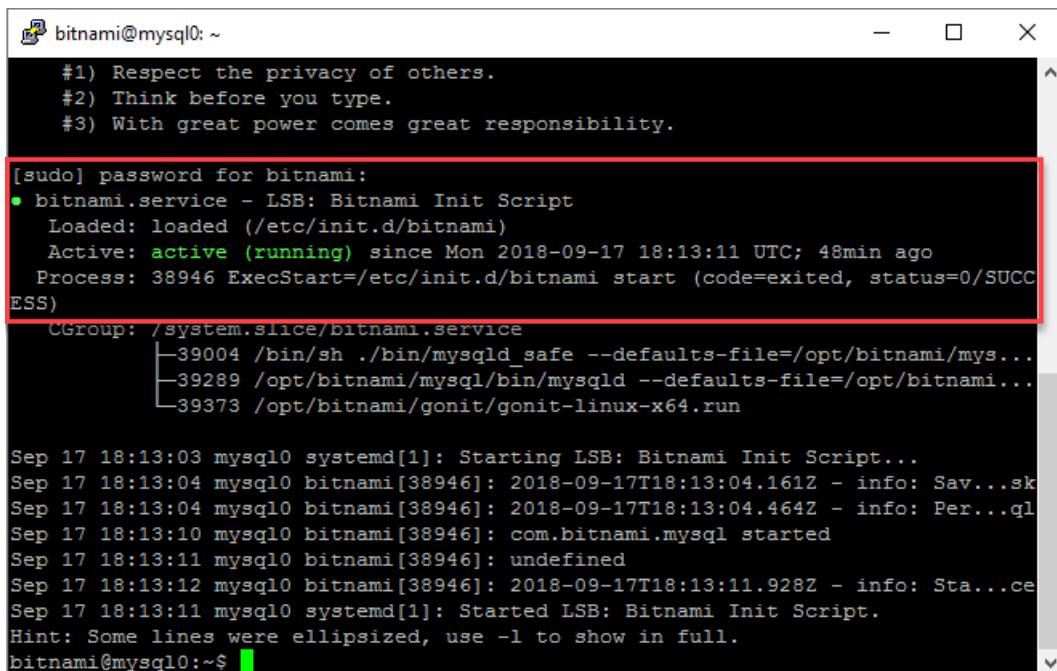
1. Using an SSH client (this example uses [PuTTY](#)) log in to the primary MySQL machine from a computer that can access the public IP. The primary MySQL VM name usually ends with **0** and has a public IP assigned to it.

Use the public IP and log in to the VM with the username of **bitnami** and the application password you created earlier without special characters.

The screenshot shows a PuTTY terminal window. The session title is 'bitnami@mysql0: ~'. The command 'login as: bitnami' is entered, followed by a password prompt. A red box highlights this password input area. Below the password prompt, the message 'Using keyboard-interactive authentication.' is visible. The terminal then displays the Bitnami MySQL welcome message, which includes the license terms and a copyright notice. The message concludes with 'This is a Bitnami server.' and ends with a prompt 'bitnami@mysql0:~\$'.

2. In the SSH client window, use the following command to ensure the bitnami service is active and running. Provide the bitnami password again when prompted:

```
sudo service bitnami status
```



```
bitnami@mysql0: ~
#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

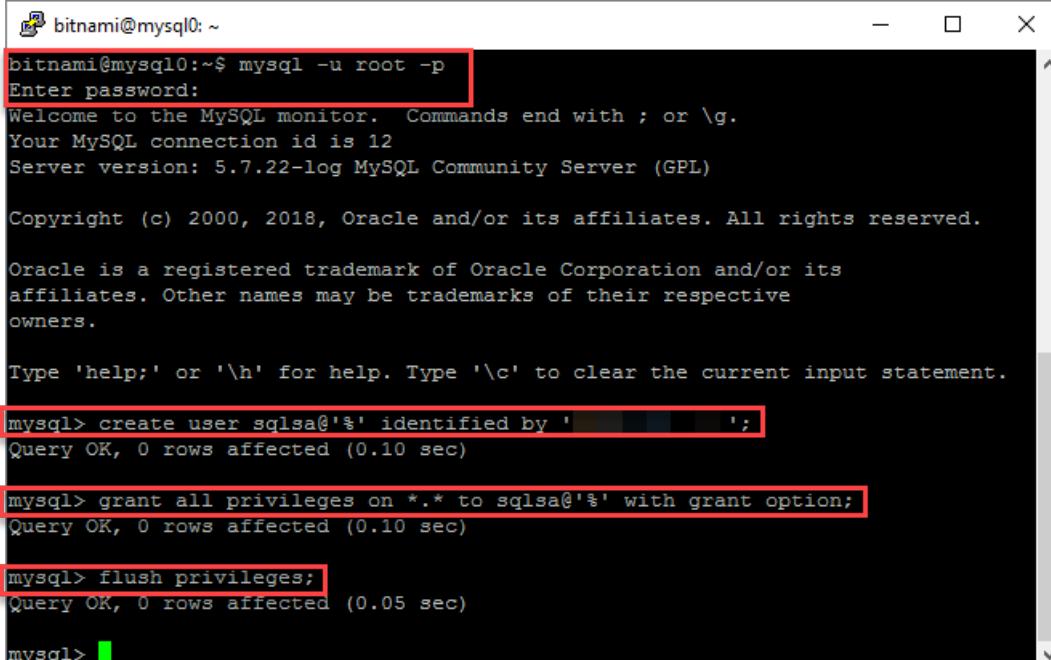
[sudo] password for bitnami:
● bitnami.service - LSB: Bitnami Init Script
 Loaded: loaded (/etc/init.d/bitnami)
 Active: active (running) since Mon 2018-09-17 18:13:11 UTC; 48min ago
 Process: 38946 ExecStart=/etc/init.d/bitnami start (code=exited, status=0/SUCCESS)
 CGroup: /system.slice/bitnami.service
 └─39004 /bin/sh ./bin/mysqld_safe --defaults-file=/opt/bitnami/mys...
 ├─39289 /opt/bitnami/mysql/bin/mysqld --defaults-file=/opt/bitnami...
 ├─39373 /opt/bitnami/gonit/gonit-linux-x64.run

Sep 17 18:13:03 mysql0 systemd[1]: Starting LSB: Bitnami Init Script...
Sep 17 18:13:04 mysql0 bitnami[38946]: 2018-09-17T18:13:04.161Z - info: Sav...sk
Sep 17 18:13:04 mysql0 bitnami[38946]: 2018-09-17T18:13:04.464Z - info: Per...ql
Sep 17 18:13:10 mysql0 bitnami[38946]: com.bitnami.mysql started
Sep 17 18:13:11 mysql0 bitnami[38946]: undefined
Sep 17 18:13:12 mysql0 bitnami[38946]: 2018-09-17T18:13:11.928Z - info: Sta...ce
Sep 17 18:13:11 mysql0 systemd[1]: Started LSB: Bitnami Init Script.
Hint: Some lines were ellipsized, use -l to show in full.
bitnami@mysql0:~$
```

3. Create a remote access user account to be used by the Azure Stack Hub MySQL Hosting Server to connect to MySQL and then exit the SSH client.

Run the following commands to log in to MySQL as root, using the root password created earlier. Create a new admin user and replace <username> and <password> as required for your environment. In this example, the created user is named **sqlsa** and a strong password is used:

```
mysql -u root -p
create user <username>'%' identified by '<password>';
grant all privileges on *.* to <username>'%' with grant option;
flush privileges;
```



```
bitnami@mysql0: ~
bitnami@mysql0:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 5.7.22-log MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create user sqlsa@'%' identified by '';
Query OK, 0 rows affected (0.10 sec)

mysql> grant all privileges on *.* to sqlsa@'%' with grant option;
Query OK, 0 rows affected (0.10 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.05 sec)

mysql>
```

4. Record the new MySQL user information.

You'll need to provide this username and password, along with the public IP address or full FQDN of the public IP for the cluster, to an Azure Stack Hub operator so they can create a MySQL hosting server using this MySQL cluster.

## Create an Azure Stack Hub MySQL Hosting Server

After the MySQL Server cluster is created and properly configured, an Azure Stack Hub operator must create an Azure Stack Hub MySQL Hosting Server to make the additional capacity available for users to create databases.

Be sure to use the public IP or full FQDN for the public IP of the MySQL cluster primary VM recorded previously when the MySQL cluster's resource group was created (**mysqlip**). In addition, the operator needs to know the MySQL Server authentication credentials you created to remotely access the MySQL cluster database.

### NOTE

This step must be run from the Azure Stack Hub administrator portal by an Azure Stack Hub operator.

Using the MySQL cluster's Public IP and MySQL authentication login information, an Azure Stack Hub operator can now [create a MySQL Hosting Server using the new MySQL cluster](#).

Also ensure that you've created plans and offers to make MySQL database creation available for users. An operator will need to add the **Microsoft.MySqlAdapter** service to a plan and create a new quota specifically for highly available databases. For more information about creating plans, see [Service, plan, offer, subscription overview](#).

### TIP

The **Microsoft.MySqlAdapter** service won't be available to add to plans until the [MySQL Server resource provider has been deployed](#).

## Create a highly available MySQL database

After the MySQL cluster is created and configured, and added as an Azure Stack Hub MySQL Hosting Server by an Azure Stack Hub operator, a tenant user with a subscription including MySQL Server database capabilities can create highly available MySQL databases by following the steps in this section.

### NOTE

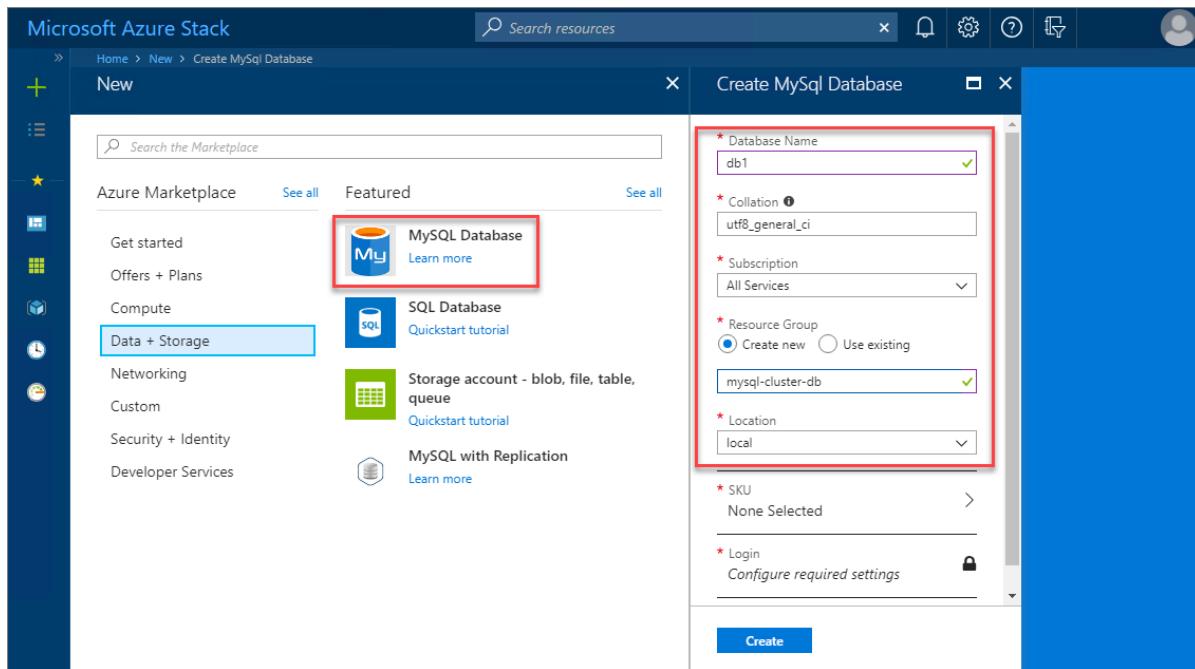
Run these steps from the Azure Stack Hub user portal as a tenant user with a subscription providing MySQL Server capabilities (Microsoft.MySQLAdapter service).

### 1. Sign in to the user portal:

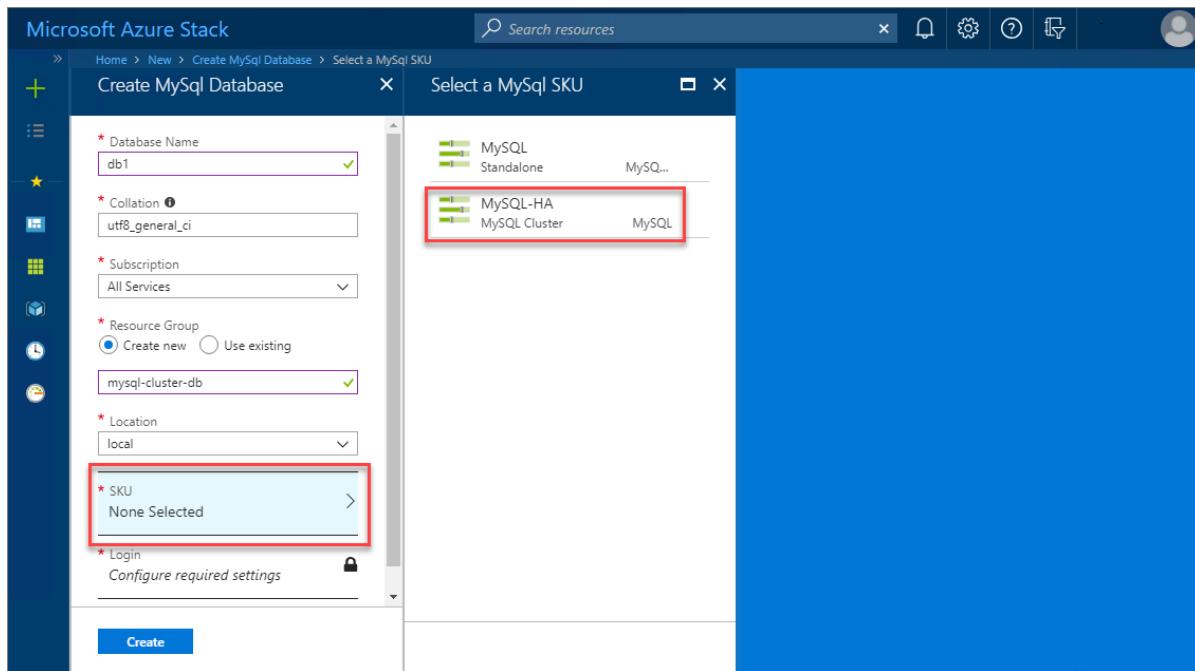
- For an integrated system deployment, the portal address will vary based on your solution's region and external domain name. It will be in the format of `https://portal.<region>.<FQDN>`.
- For the Azure Stack Development Kit (ASDK), the portal address is  
`https://portal.local.azurestack.external`.

### 2. Select + **Create a resource** > **Data + Storage**, and then **MySQL Database**.

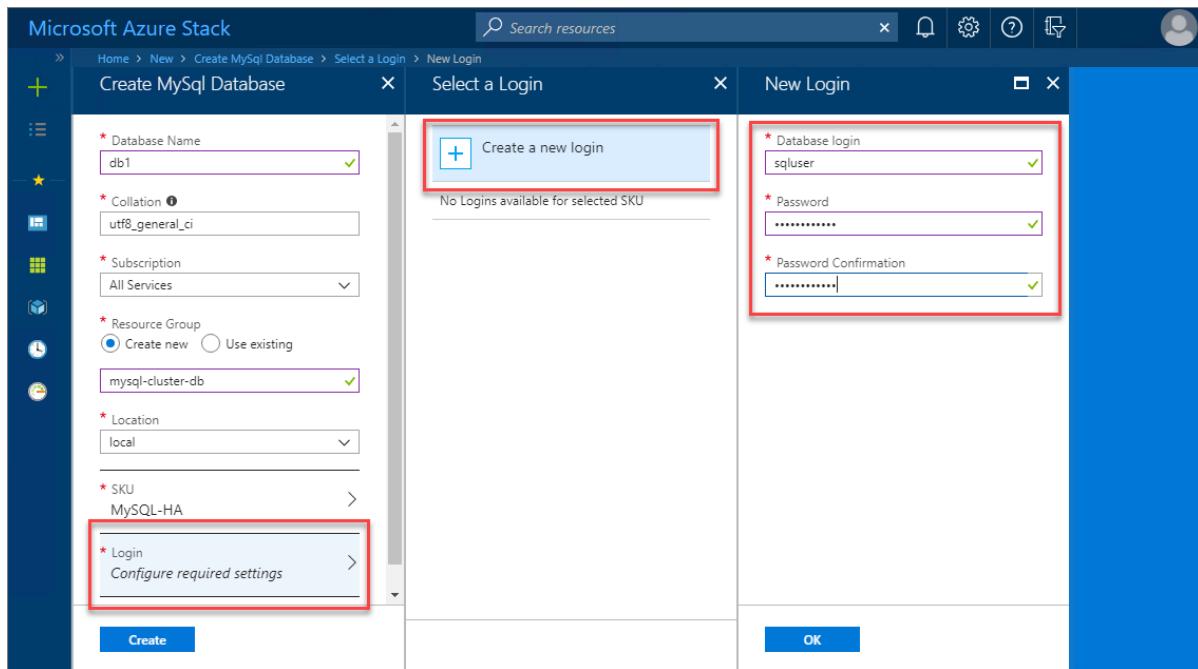
Provide the required database property information including name, collation, the subscription to use, and location to use for the deployment.



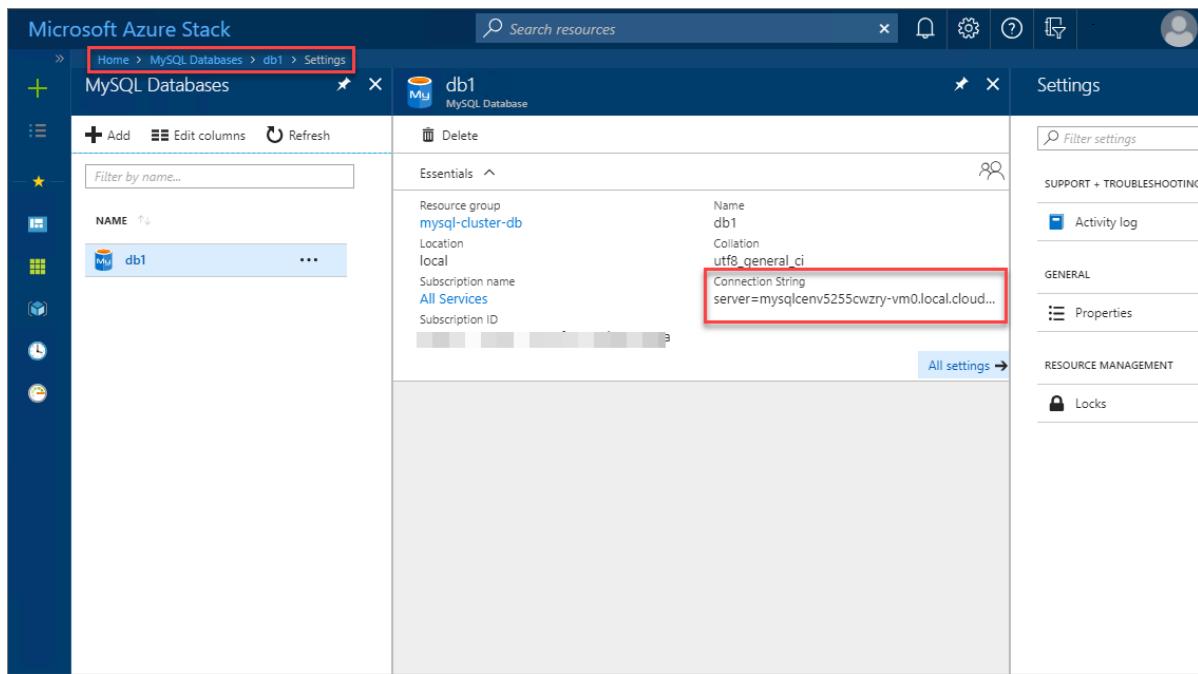
3. Select **SKU** and then choose the appropriate MySQL Hosting Server SKU to use. In this example, the Azure Stack Hub operator has created the **MySQL-HA** SKU to support high availability for MySQL cluster databases.



4. Select **Login > Create a new login** and then provide the MySQL authentication credentials to be used for the new database. When finished, select **OK** and then **Create** to begin the database deployment process.



- When the MySQL database deployment completes successfully, review the database properties to discover the connection string to use for connecting to the new highly available database.



## Next steps

[Update the MySQL resource provider](#)

# Update the MySQL resource provider in Azure Stack Hub

4 minutes to read • [Edit Online](#)

A new MySQL resource provider adapter might be released when Azure Stack Hub builds are updated. While the existing adapter continues to work, we recommend updating to the latest build as soon as possible.

Starting with the MySQL resource provider version 1.1.33.0 release, updates are cumulative and don't need to be installed in the order in which they were released as long as you're starting from version 1.1.24.0 or later. For example, if you're running version 1.1.24.0 of the MySQL resource provider, then you can upgrade to version 1.1.33.0 or later without needing to first install version 1.1.30.0. To review available resource provider versions, and the version of Azure Stack Hub they're supported on, refer to the versions list in [Deploy the resource provider prerequisites](#).

To update of the resource provider, you use the **UpdateMySQLProvider.ps1** script. The process is similar to the process used to install a resource provider, as described in the Deploy the resource provider section of this article. The script is included with the download of the resource provider.

## IMPORTANT

Before upgrading the resource provider, review the release notes to learn about new functionality, fixes, and any known issues that could affect your deployment.

## Update script processes

The **UpdateMySQLProvider.ps1** script creates a new virtual machine (VM) with the latest resource provider code and migrates the settings from the old VM to the new VM. The settings that migrate include database and hosting server information and the necessary DNS record.

## NOTE

We recommend that you download the latest Windows Server 2016 Core image from Marketplace Management. If you need to install an update, you can place a **single** MSU package in the local dependency path. The script will fail if there's more than one MSU file in this location.

The script requires use of the same arguments that are described for the `DeployMySqlProvider.ps1` script. Provide the certificate here as well.

## Update script parameters

Specify the following parameters from the command line when you run the **UpdateMySQLProvider.ps1** PowerShell script. If you don't, or if any parameter validation fails, you're prompted to provide the required parameters.

| PARAMETER NAME              | DESCRIPTION                                                                          | COMMENT OR DEFAULT VALUE |
|-----------------------------|--------------------------------------------------------------------------------------|--------------------------|
| <b>CloudAdminCredential</b> | The credential for the cloud admin, necessary for accessing the privileged endpoint. | <i>Required</i>          |

| PARAMETER NAME                       | DESCRIPTION                                                                                                                                                                                                                                                          | COMMENT OR DEFAULT VALUE                   |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| <b>AzCredential</b>                  | The credentials for the Azure Stack Hub service admin account. Use the same credentials as you used for deploying Azure Stack Hub.                                                                                                                                   | <i>Required</i>                            |
| <b>VMLocalCredential</b>             | The credentials for the local admin account of the SQL resource provider VM.                                                                                                                                                                                         | <i>Required</i>                            |
| <b>PrivilegedEndpoint</b>            | The IP address or DNS name of the privileged endpoint.                                                                                                                                                                                                               | <i>Required</i>                            |
| <b>AzureEnvironment</b>              | The Azure environment of the service admin account used for deploying Azure Stack Hub. Required only for Azure AD deployments. Supported environment names are <b>AzureCloud</b> , <b>AzureUSGovernment</b> , or if using a China Azure AD, <b>AzureChinaCloud</b> . | AzureCloud                                 |
| <b>DependencyFilesLocalPath</b>      | Your certificate .pfx file must be placed in this directory as well.                                                                                                                                                                                                 | <i>Optional (mandatory for multi-node)</i> |
| <b>DefaultSSLCertificatePassword</b> | The password for the .pfx certificate.                                                                                                                                                                                                                               | <i>Required</i>                            |
| <b>MaxRetryCount</b>                 | The number of times you want to retry each operation if there's a failure.                                                                                                                                                                                           | 2                                          |
| <b>RetryDuration</b>                 | The timeout interval between retries, in seconds.                                                                                                                                                                                                                    | 120                                        |
| <b>Uninstall</b>                     | Remove the resource provider and all associated resources (see the following notes).                                                                                                                                                                                 | No                                         |
| <b>DebugMode</b>                     | Prevents automatic cleanup on failure.                                                                                                                                                                                                                               | No                                         |
| <b>AcceptLicense</b>                 | Skips the prompt to accept the GPL license.<br>( <a href="https://www.gnu.org/licenses/old-licenses/gpl-2.0.html">https://www.gnu.org/licenses/old-licenses/gpl-2.0.html</a> )                                                                                       |                                            |

## Update script example

**NOTE**

The update process only applies to integrated systems.

If you are updating the MySQL resource provider version to 1.1.33.0 or previous versions, you need to install specific versions of AzureRm.BootStrapper and Azure Stack Hub modules in PowerShell. If you are updating the

MySQL resource provider to version 1.1.47.0, the deployment script will automatically download and install the necessary PowerShell modules for you to path C:\Program Files\SqlMySqlPsh.

```
Install the AzureRM.Bootstrapper module, set the profile and install the AzureStack module
Note that this might not be the most currently available version of Azure Stack Hub PowerShell.
Install-Module -Name AzureRm.BootStrapper -Force
Use-AzureRmProfile -Profile 2018-03-01-hybrid -Force
Install-Module -Name AzureStack -RequiredVersion 1.6.0
```

#### NOTE

In disconnected scenario, you need to download the required PowerShell modules and register the repository manually as a prerequisite. You can get more information in [Deploy MySQL resource provider](#)

The following example shows the *UpdateMySQLProvider.ps1* script that you can run from an elevated PowerShell console. Be sure to change the variable information and passwords as needed:

```

Use the NetBIOS name for the Azure Stack Hub domain. On the Azure Stack Hub SDK, the default is AzureStack
but could have been changed at install time.
$domain = "AzureStack"

For integrated systems, use the IP address of one of the ERCS VMs.
$privilegedEndpoint = "AzS-ERCS01"

Provide the Azure environment used for deploying Azure Stack Hub. Required only for Azure AD deployments.
Supported environment names are AzureCloud, AzureUSGovernment, or AzureChinaCloud.
$AzureEnvironment = "<EnvironmentName>"

Point to the directory where the resource provider installation files were extracted.
$tempDir = 'C:\TEMP\MYSQLRP'

The service admin account (can be Azure Active Directory or Active Directory Federation Services).
$serviceAdmin = "admin@mydomain.onmicrosoft.com"
$AdminPass = ConvertTo-SecureString "P@ssw0rd1" -AsPlainText -Force
$AdminCreds = New-Object System.Management.Automation.PSCredential ($serviceAdmin, $AdminPass)

Set credentials for the new resource provider VM.
$vmLocalAdminPass = ConvertTo-SecureString "P@ssw0rd1" -AsPlainText -Force
$vmLocalAdminCreds = New-Object System.Management.Automation.PSCredential ("mysqlrpadmin", $vmLocalAdminPass)

And the cloudadmin credential required for privileged endpoint access.
$CloudAdminPass = ConvertTo-SecureString "P@ssw0rd1" -AsPlainText -Force
$CloudAdminCreds = New-Object System.Management.Automation.PSCredential ("$domain\cloudadmin",
$CloudAdminPass)

Change the following as appropriate.
$PfxPass = ConvertTo-SecureString "P@ssw0rd1" -AsPlainText -Force

For version 1.1.47.0, the PowerShell modules used by the RP deployment are placed in C:\Program
Files\SqlMySqlPsh
The deployment script adds this path to the system $env:PSModulePath to ensure correct modules are used.
$rpModulePath = Join-Path -Path $env:ProgramFiles -ChildPath 'SqlMySqlPsh'
$env:PSModulePath = $env:PSModulePath + ";" + $rpModulePath

Change directory to the folder where you extracted the installation files.
Then adjust the endpoints.
.$tempDir\UpdateMySQLProvider.ps1 -AzCredential $AdminCreds `

-VMLocalCredential $vmLocalAdminCreds `

-CloudAdminCredential $cloudAdminCreds `

-PrivilegedEndpoint $privilegedEndpoint `

-AzureEnvironment $AzureEnvironment `

-DefaultSSLCertificatePassword $PfxPass `

-DependencyFilesLocalPath $tempDir\cert `

-AcceptLicense

```

When the resource provider update script finishes, close the current PowerShell session.

## Next steps

[Maintain MySQL resource provider](#)

# MySQL resource provider maintenance operations in Azure Stack Hub

5 minutes to read • [Edit Online](#)

The MySQL resource provider runs on a locked down virtual machine (VM). To enable maintenance operations, you need to update the VM's security. To do this using the principle of least privilege (POLP), you can use PowerShell Just Enough Administration (JEA) endpoint DBAdapterMaintenance. The resource provider installation package includes a script for this operation.

## Update the VM operating system

Because the resource provider runs on a *user* VM, you need to apply the required patches and updates when they're released. You can use the Windows update packages that are provided as part of the patch-and-update cycle to apply updates to the VM.

Update the provider VM using one of the following methods:

- Install the latest resource provider package using a currently patched Windows Server 2016 Core image.
- Install a Windows Update package during the installation or update of the resource provider.

## Update the VM Windows Defender definitions

To update the Defender definitions, follow these steps:

1. Download the Windows Defender definitions update from [Windows Defender Definition](#).

On the definitions page, scroll down to "Manually download and install the definitions". Download the "Windows Defender Antivirus for Windows 10 and Windows 8.1" 64-bit file.

Alternatively, use [this direct link](#) to download/run the fpam-fe.exe file.

2. Open a PowerShell session to the MySQL resource provider adapter VM's maintenance endpoint.
3. Copy the definitions update file to the resource provider adapter VM using the maintenance endpoint session.
4. On the maintenance PowerShell session, run the *Update-DBAdapterWindowsDefenderDefinitions* command.
5. After you install the definitions, we recommend that you delete the definitions update file by using the *Remove-ItemOnUserDrive*) command.

### **PowerShell script example for updating definitions.**

You can edit and run the following script to update the Defender definitions. Replace values in the script with values from your environment.

```

Set credentials for the local admin on the resource provider VM.
$vmLocalAdminPass = ConvertTo-SecureString "<local admin user password>" -AsPlainText -Force
$vmLocalAdminUser = "<local admin user name>"
$vmLocalAdminCreds = New-Object System.Management.Automation.PSCredential `
 ($vmLocalAdminUser, $vmLocalAdminPass)

Provide the public IP address for the adapter VM.
$databaseRPMachine = "<RP VM IP address>"
$localPathToDefenderUpdate = "C:\DefenderUpdates\mpam-fe.exe"

Download Windows Defender update definitions file from https://www.microsoft.com/en-us/wdsi/definitions.
Invoke-WebRequest -Uri 'https://go.microsoft.com/fwlink/?LinkID=121721&arch=x64' `
 -Outfile $localPathToDefenderUpdate

Create a session to the maintenance endpoint.
$session = New-PSSession -ComputerName $databaseRPMachine `
 -Credential $vmLocalAdminCreds -ConfigurationName DBAdapterMaintenance

Copy the defender update file to the adapter VM.
Copy-Item -ToSession $session -Path $localPathToDefenderUpdate `
 -Destination "User:\"

Install the update definitions.
Invoke-Command -Session $session -ScriptBlock `
 {Update-AzSDBAdapterWindowsDefenderDefinition -DefinitionsUpdatePackageFile "User:\mpam-fe.exe"}

Cleanup the definitions package file and session.
Invoke-Command -Session $session -ScriptBlock `
 {Remove-AzSItemOnUserDrive -ItemPath "User:\mpam-fe.exe"}
$session | Remove-PSSession

```

## Secrets rotation

*These instructions only apply to Azure Stack Hub Integrated Systems.*

When using the SQL and MySQL resource providers with Azure Stack Hub integrated systems, the Azure Stack Hub operator is responsible for rotating the following resource provider infrastructure secrets to ensure that they don't expire:

- External SSL Certificate provided during deployment.
- The resource provider VM local administrator account password provided during deployment.
- Resource provider diagnostic user (dbadapterdiag) password.

### PowerShell examples for rotating secrets

**Change all the secrets at the same time:**

```
.
.\SecretRotationMySQLProvider.ps1 `
 -Privilegedendpoint $Privilegedendpoint `
 -CloudAdminCredential $cloudCreds `
 -AzCredential $adminCreds `
 -DiagnosticsUserPassword $passwd `
 -DependencyFilesLocalPath $certPath `
 -DefaultSSLCertificatePassword $certPasswd `
 -VMLocalCredential $localCreds
```

**Change the diagnostic user password:**

```
.\\SecretRotationMySQLProvider.ps1 `
-Privilegedendpoint $Privilegedendpoint `
-CloudAdminCredential $cloudCreds `
-AzCredential $adminCreds `
-DiagnosticsUserPassword $passwd
```

## Change the VM local admin account password:

```
.\\SecretRotationMySQLProvider.ps1 `
-Privilegedendpoint $Privilegedendpoint `
-CloudAdminCredential $cloudCreds `
-AzCredential $adminCreds `
-VMLocalCredential $localCreds
```

## Change the SSL certificate password:

```
.\\SecretRotationMySQLProvider.ps1 `
-Privilegedendpoint $Privilegedendpoint `
-CloudAdminCredential $cloudCreds `
-AzCredential $adminCreds `
-DependencyFilesLocalPath $certPath `
-DefaultSSLCertificatePassword $certPasswd
```

## SecretRotationMySQLProvider.ps1 parameters

| PARAMETER                     | DESCRIPTION                                                   |
|-------------------------------|---------------------------------------------------------------|
| AzCredential                  | Azure Stack Hub service admin account credential.             |
| CloudAdminCredential          | Azure Stack Hub cloud admin domain account credential.        |
| PrivilegedEndpoint            | Privileged Endpoint to access Get-AzureStackStampInformation. |
| DiagnosticsUserPassword       | Diagnostics user account password.                            |
| VMLocalCredential             | The local admin account on the MySQLAdapter VM.               |
| DefaultSSLCertificatePassword | Default SSL Certificate (*.pfx) password.                     |
| DependencyFilesLocalPath      | Dependency files local path.                                  |

## Known issues

### Issue:

The logs for secrets rotation aren't automatically collected if the secret rotation script fails when it's run.

### Workaround:

Use the Get-AzsDBAdapterLogs cmdlet to collect all the resource provider logs, including AzureStack.DatabaseAdapter.SecretRotation.ps1\_\* .log, saved in C:\Logs.

# Collect diagnostic logs

To collect logs from the locked down VM, use the PowerShell Just Enough Administration (JEA) endpoint DBAdapterDiagnostics. This endpoint provides the following commands:

- **Get-AzsDBAdapterLog.** This command creates a zip package of the resource provider diagnostics logs and saves the file on the session's user drive. You can run this command without any parameters and the last four hours of logs are collected.
- **Remove-AzsDBAdapterLog.** This command removes existing log packages on the resource provider VM.

## Endpoint requirements and process

When a resource provider is installed or updated, the dbadapterdiag user account is created. You'll use this account to collect diagnostic logs.

### NOTE

The dbadapterdiag account password is the same as the password used for the local admin on the VM that's created during a provider deployment or update.

To use the *DBAdapterDiagnostics* commands, create a remote PowerShell session to the resource provider VM and run the **Get-AzsDBAdapterLog** command.

You set the time span for log collection by using the **FromDate** and **ToDate** parameters. If you don't specify one or both of these parameters, the following defaults are used:

- FromDate is four hours before the current time.
- ToDate is the current time.

## PowerShell script example for collecting logs:

The following script shows how to collect diagnostic logs from the resource provider VM.

```
Create a new diagnostics endpoint session.
$databaseRPMachineIP = '<RP VM IP address>'
$diagnosticsUserName = 'dbadapterdiag'
$diagnosticsUserPassword = '<Enter Diagnostic password>'
$diagCreds = New-Object System.Management.Automation.PSCredential `
 ($diagnosticsUserName, (ConvertTo-SecureString -String $diagnosticsUserPassword -AsPlainText -Force))
$session = New-PSSession -ComputerName $databaseRPMachineIP -Credential $diagCreds
 -ConfigurationName DBAdapterDiagnostics

Sample that captures logs from the previous hour.
$fromDate = (Get-Date).AddHours(-1)
$dateNow = Get-Date
$sb = {param($d1,$d2) Get-AzSDBAdapterLog -FromDate $d1 -ToDate $d2}
$logs = Invoke-Command -Session $session -ScriptBlock $sb -ArgumentList $fromDate,$dateNow

Copy the logs to the user drive.
$sourcePath = "User:\{0}" -f $logs
$destinationPackage = Join-Path -Path (Convert-Path '.') -ChildPath $logs
Copy-Item -FromSession $session -Path $sourcePath -Destination $destinationPackage

Cleanup the logs.
$cleanup = Invoke-Command -Session $session -ScriptBlock {Remove-AzsDBAdapterLog}
Close the session.
$session | Remove-PSSession
```

# Configure Azure Diagnostics extension for MySQL resource provider

The Azure Diagnostics extension is installed on the MySQL resource provider adapter VM by default. The following steps show how to customize the extension for gathering the MySQL resource provider operational event logs and IIS logs for troubleshooting and auditing purposes.

1. Sign in to the Azure Stack Hub administrator portal.
2. Select **Virtual machines** from the pane on the left, search for the MySQL resource provider adapter VM and select the VM.
3. In the **Diagnostics settings** of the VM, go to the **Logs** tab and choose **Custom** to customize event logs being collected.

The screenshot shows the 'Diagnostics settings' page for a virtual machine named 'MySQLVM11470'. The left sidebar lists various management options like Overview, Activity log, Access control (IAM), Tags, Networking, Disks, Size, Extensions, Availability set, Properties, Locks, Monitoring, and Support + troubleshooting. The 'Diagnostics settings' option under Monitoring is currently selected. The main content area is titled 'MySQLVM11470 - Diagnostics settings' and includes tabs for Overview, Performance counters, Logs (which is selected and highlighted with a red box), Crash dumps, Sinks, and Agent. Under the Logs tab, there's a section for 'Event logs' with a note about choosing 'Basic' or 'Custom' collection. A 'Custom' button is also highlighted with a red box. Below this, there's a list of event logs to collect, including Application, Security, and System logs, each with an ellipsis button. There's also a section for 'Directories' where users can choose IIS logs and specify a storage container name, which is marked with a red asterisk as required. An 'Add' button is located at the bottom right of the log configuration area.

4. Add **Microsoft-AzureStack-DatabaseAdapter/Operational!\*** to collect MySQL resource provider operational event logs.

Dashboard > Virtual machines > MySqlVM11470 - Diagnostics settings

## MySQLVM11470 - Diagnostics settings

Save Discard

Overview Performance counters Logs Crash dumps Sinks Agent

Event logs

Choose **Basic** to enable collection of event logs. Choose **Custom** if you want more control over which event logs are collected.

None Basic Custom

Configure the event logs and levels to collect:

Microsoft-AzureStack-DatabaseAdapter/Operational!

EVENT LOGS

Application!\* [Application[(Level = 1 or Level = 2 or Level = 3)]]

Security!\* [System[band(Keywords,4503599627370496)]]

System!\* [System[(Level = 1 or Level = 2 or Level = 3)]]

Directories

Choose the IIS logs to collect and the log directories to monitor.

IIS logs ?

\* Storage container name:

Failed request logs ?

\* Storage container name:

Boot diagnostics

5. To enable the collection of IIS logs, check **IIS logs** and **Failed request logs**.

Dashboard > Virtual machines > MySqlVM11470 - Diagnostics settings

## MySQLVM11470 - Diagnostics settings

Save Discard

Directories

Choose the IIS logs to collect and the log directories to monitor.

IIS logs ?

\* Storage container name:

Failed request logs ?

\* Storage container name:

Application logs

Collect the tracing output generated by your .NET application.

Disabled  Enabled

Event tracing for Windows (ETW) events

Collect ETW data generated from the event sources and manifests you specify.

Disabled  Enabled

6. Finally, select **Save** to save all the diagnostics settings.

Once the event logs and IIS logs collection are configured for MySQL resource provider, the logs can be found in a system storage account named **mysqladapterdiagaccount**.

To learn more about the Azure Diagnostics extension, see [What is Azure Diagnostics extension](#).

## Next steps

[Remove the MySQL resource provider](#)

# Remove the MySQL resource provider in Azure Stack Hub

2 minutes to read • [Edit Online](#)

Before you remove the MySQL resource provider, you must remove all the provider dependencies. You'll also need a copy of the deployment package that was used to install the resource provider.

## NOTE

You can find the download links for the resource provider installers in [Deploy the resource provider prerequisites](#).

Removing the MySQL resource provider will delete the associated plans and quotas managed by operator. But it doesn't delete tenant databases from hosting servers.

## To remove the MySQL resource provider

1. Verify that you've removed all the existing MySQL resource provider dependencies.

## NOTE

Uninstalling the MySQL resource provider will proceed even if dependent resources are currently using the resource provider.

2. Get a copy of the MySQL resource provider installation package and then run the self-extractor to extract the contents to a temporary directory.

3. Open a new elevated PowerShell console window and change to the directory where you extracted the MySQL resource provider installation files.

4. Run the DeployMySqlProvider.ps1 script using the following parameters:

- **Uninstall:** Removes the resource provider and all associated resources.
- **PrivilegedEndpoint:** The IP address or DNS name of the privileged endpoint.
- **AzureEnvironment:** The Azure environment used for deploying Azure Stack Hub. Required only for Azure AD deployments.
- **CloudAdminCredential:** The credential for the cloud administrator, necessary to access the privileged endpoint.
- **DirectoryTenantID**
- **AzCredential:** The credential for the Azure Stack Hub service admin account. Use the same credentials that you used for deploying Azure Stack Hub.

## Next steps

[Offer App Services as PaaS](#)

# MySQL resource provider 1.1.47.0 release notes

2 minutes to read • [Edit Online](#)

These release notes describe the improvements and known issues in MySQL resource provider version 1.1.47.0.

## Build reference

Download the MySQL resource provider binary and then run the self-extractor to extract the contents to a temporary directory. The resource provider has a minimum corresponding Azure Stack Hub build. The minimum Azure Stack Hub release version required to install this version of the MySQL resource provider is listed below:

| MINIMUM AZURE STACK HUB VERSION | MYSQL RESOURCE PROVIDER VERSION           |
|---------------------------------|-------------------------------------------|
| Version 1910 (1.1910.0.58)      | <a href="#">MySQL RP version 1.1.47.0</a> |

### IMPORTANT

Apply the minimum supported Azure Stack Hub update to your Azure Stack Hub integrated system or deploy the latest Azure Stack Development Kit (ASDK) before deploying the latest version of the MySQL resource provider.

## New features and fixes

This version of the Azure Stack Hub MySQL resource provider is a hotfix release to make the resource provider compatible with some of the latest portal changes in the 1910 update without any new feature.

It also supports the current latest Azure Stack Hub API version profile 2019-03-01-hybrid and Azure Stack Hub PowerShell module 1.8.0. So during deployment and update, no specific history versions of modules need to be installed.

It is recommended that you apply the MySQL resource provider hotfix 1.1.47.0 after Azure Stack Hub is upgraded to the 1910 release.

## Known issues

None.

## Next steps

[Learn more about the MySQL resource provider.](#)

[Prepare to deploy the MySQL resource provider.](#)

[Upgrade the MySQL resource provider from a previous version.](#)

# MySQL resource provider 1.1.33.0 release notes

2 minutes to read • [Edit Online](#)

These release notes describe the improvements and known issues in MySQL resource provider version 1.1.33.0.

## Build reference

Download the MySQL resource provider binary and then run the self-extractor to extract the contents to a temporary directory. The resource provider has a minimum corresponding Azure Stack Hub build. The minimum Azure Stack Hub release version required to install this version of the MySQL resource provider is listed below:

| MINIMUM AZURE STACK HUB VERSION | MYSQL RESOURCE PROVIDER VERSION           |
|---------------------------------|-------------------------------------------|
| Version 1808 (1.1808.0.97)      | <a href="#">MySQL RP version 1.1.33.0</a> |

### IMPORTANT

Apply the minimum supported Azure Stack Hub update to your Azure Stack Hub integrated system or deploy the latest Azure Stack Development Kit (ASDK) before deploying the latest version of the MySQL resource provider.

## New features and fixes

This version of the Azure Stack Hub MySQL resource provider includes the following improvements and fixes:

### Fixes

- MySQL resource provider portal extension might choose the wrong subscription.** The MySQL resource provider uses Azure Resource Manager calls to determine the first service admin subscription to use, which might not be the *Default Provider Subscription*. If that happens, the MySQL resource provider doesn't work normally.
- MySQL hosting server doesn't list hosted databases.** User-created databases might not be listed when viewing tenant resources for MySQL hosting servers.
- Previous MySQL resource provider (1.1.30.0) deployment could fail if TLS 1.2 isn't enabled.** Updated the MySQL resource provider 1.1.33.0 to enable TLS 1.2 when deploying the resource provider, updating the resource provider, or rotating secrets.
- MySQL resource provider secret rotation fails.** Fixed an issue resulting in the following error code when rotating secrets:  
`New-AzureRmResourceGroupDeployment - Error: Code=InvalidDeploymentParameterValue; Message=The value of deployment parameter 'StorageAccountBlobUri' is null.`

## Known issues

- MySQL SKUs can take up to an hour to be visible in the portal.** It can take up to an hour for newly created SKUs to be visible for use when creating new MySQL databases.

**Workaround:** None.

- **Reused MySQL logins.** Attempting to create a new MySQL login with the same username as an existing login under the same subscription will result in reusing the same login and the existing password.

**Workaround:** Use different usernames when creating new logins under the same subscription or create logins with the same username under different subscriptions.

- **Shared MySQL logins cause data inconsistency.** If a MySQL login is shared for multiple MySQL databases under the same subscription, changing the login password will cause data inconsistency.

**Workaround:** Always use different logins for different databases under the same subscription.

#### Known issues for Cloud Admins operating Azure Stack Hub

Refer to the documentation in the [Azure Stack Hub Release Notes](#).

## Next steps

[Learn more about the MySQL resource provider.](#)

[Prepare to deploy the MySQL resource provider.](#)

[Upgrade the MySQL resource provider from a previous version.](#)

# MySQL resource provider 1.1.30.0 release notes

2 minutes to read • [Edit Online](#)

These release notes describe the improvements and known issues in MySQL resource provider version 1.1.30.0.

## Build reference

Download the MySQL resource provider binary and then run the self-extractor to extract the contents to a temporary directory. The resource provider has a minimum corresponding Azure Stack Hub build. The minimum Azure Stack Hub release version required to install this version of the MySQL resource provider is listed below:

| MINIMUM AZURE STACK HUB VERSION           | MYSQL RESOURCE PROVIDER VERSION |
|-------------------------------------------|---------------------------------|
| Azure Stack Hub 1808 update (1.1808.0.97) | <a href="#">1.1.30.0</a>        |

### IMPORTANT

Apply the minimum supported Azure Stack Hub update to your Azure Stack Hub integrated system or deploy the latest Azure Stack Development Kit (ASDK) before deploying the latest version of the MySQL resource provider.

## New features and fixes

This version of the Azure Stack Hub MySQL resource provider includes the following improvements and fixes:

- **Telemetry enabled for MySQL resource provider deployments.** Telemetry collection has been enabled for MySQL resource provider deployments. Telemetry collected includes resource provider deployment, start and stop times, exit status, exit messages, and error details (if applicable).
- **TLS 1.2 encryption update.** Enabled TLS 1.2-only support for resource provider communication with internal Azure Stack Hub components.

### Fixes

- **MySQL resource provider Azure Stack Hub PowerShell compatibility.** The MySQL resource provider has been updated to work with the Azure Stack Hub 2018-03-01-hybrid PowerShell profile and to provide compatibility with AzureRM 1.3.0 and later.
- **MySQL login change password blade.** Fixed an issue where the password can't be changed on the change password blade. Removed links from password change notifications.

## Known issues

- **MySQL SKUs can take up to an hour to be visible in the portal.** It can take up to an hour for newly created SKUs to be visible for use when creating new MySQL databases.

**Workaround:** None.

- **Reused MySQL logins.** Attempting to create a new MySQL login with the same username as an existing login under the same subscription will result in reusing the same login and the existing password.

**Workaround:** Use different usernames when creating new logins under the same subscription or create logins with the same username under different subscriptions.

- **TLS 1.2 support requirement.** If you try to deploy or update the MySQL resource provider from a computer where TLS 1.2 isn't enabled, the operation might fail. Run the following PowerShell command on the computer being used to deploy or update the resource provider to verify that TLS 1.2 is returned as supported:

```
[System.Net.ServicePointManager]::SecurityProtocol
```

If **Tls12** isn't included in the output of the command, TLS 1.2 isn't enabled on the computer.

**Workaround:** Run the following PowerShell command to enable TLS 1.2 and then start the resource provider deployment or update script from the same PowerShell session:

```
[System.Net.ServicePointManager]::SecurityProtocol = [System.Net.SecurityProtocolType]::Tls12
```

#### Known issues for Cloud Admins operating Azure Stack Hub

Refer to the documentation in the [Azure Stack Hub Release Notes](#).

## Next steps

[Learn more about the MySQL resource provider.](#)

[Prepare to deploy the MySQL resource provider.](#)

[Upgrade the MySQL resource provider from a previous version.](#)

# Use SQL databases on Azure Stack Hub

2 minutes to read • [Edit Online](#)

Use the SQL resource provider to offer SQL databases on [Azure Stack Hub](#). After you install the resource provider and connect it to one or more SQL Server instances, you and your users can create:

- SQL databases for cloud-native apps.
- SQL databases for web applications.

Limitations to consider before installing the SQL resource provider:

- Users can only create and manage individual databases. Database Server instance isn't accessible to end users. This may limit compatibility with on-premises database apps that need access to master, Temp DB, or to dynamically manage databases.
- Your Azure Stack Hub operator is responsible for deploying, updating, securing, configuring and maintaining the SQL database servers and hosts. The RP service doesn't provide any host and database server instance management functionality.
- Databases from different users in different subscriptions may be located on the same database server instance. The RP doesn't provide any mechanism for isolating databases on different hosts or database server instances.
- The RP doesn't provide any reporting on tenant usage of databases.

For traditional SQL Server workload on premises, SQL Server virtual machine on Azure Stack Hub is recommended.

## SQL resource provider adapter architecture

The resource provider consists of the following components:

- **The SQL resource provider adapter virtual machine (VM)**, which is a Windows Server VM that runs the provider services.
- **The resource provider**, which processes requests and accesses database resources.
- **Servers that host SQL Server**, which provide capacity for databases called hosting servers.

You must create at least one instance of SQL Server or provide access to external SQL Server instances.

### NOTE

Hosting servers that are installed on Azure Stack Hub integrated systems must be created from a tenant subscription. They can't be created from the default provider subscription. They must be created from the user portal or by using PowerShell with the appropriate sign-in. All hosting servers are billable VMs and must have licenses. The service admin can be the owner of the tenant subscription.

## Next steps

[Deploy the SQL Server resource provider](#)

# Deploy the SQL Server resource provider on Azure Stack Hub

7 minutes to read • [Edit Online](#)

Use the Azure Stack Hub SQL Server resource provider to expose SQL databases as an Azure Stack Hub service. The SQL resource provider runs as a service on a Windows Server 2016 Server Core virtual machine (VM).

## IMPORTANT

Only the resource provider is supported to create items on servers that host SQL or MySQL. Items created on a host server that aren't created by the resource provider might result in a mismatched state.

## Prerequisites

There are several prerequisites that need to be in place before you can deploy the Azure Stack Hub SQL resource provider. To meet these requirements, complete the following steps on a computer that can access the privileged endpoint VM:

- If you haven't already, [register Azure Stack Hub](#) with Azure so you can download Azure Marketplace items.
- Add the required Windows Server core VM to Azure Stack Hub Marketplace by downloading the **Windows Server 2016 Datacenter - Server Core** image.
- Download the SQL resource provider binary and then run the self-extractor to extract the contents to a temporary directory. The resource provider has a minimum corresponding Azure Stack Hub build.

| MINIMUM AZURE STACK HUB VERSION | SQL RP VERSION                          |
|---------------------------------|-----------------------------------------|
| Version 1910 (1.1910.0.58)      | <a href="#">SQL RP version 1.1.47.0</a> |
| Version 1808 (1.1808.0.97)      | <a href="#">SQL RP version 1.1.33.0</a> |
| Version 1808 (1.1808.0.97)      | <a href="#">SQL RP version 1.1.30.0</a> |
| Version 1804 (1.0.180513.1)     | <a href="#">SQL RP version 1.1.24.0</a> |
|                                 |                                         |

## IMPORTANT

Before deploying the SQL resource provider version 1.1.47.0, you should have your Azure Stack Hub system upgraded to 1910 update or later versions. The SQL resource provider version 1.1.47.0 on previous unsupported Azure Stack Hub versions doesn't work.

- Ensure datacenter integration prerequisites are met:

| PREREQUISITE                                 | REFERENCE                                                    |
|----------------------------------------------|--------------------------------------------------------------|
| Conditional DNS forwarding is set correctly. | <a href="#">Azure Stack Hub datacenter integration - DNS</a> |

| PREREQUISITE                                       | REFERENCE                                                                                                                                           |
|----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| Inbound ports for resource providers are open.     | <a href="#">Azure Stack Hub datacenter integration - Ports and protocols inbound</a>                                                                |
| PKI certificate subject and SAN are set correctly. | <a href="#">Azure Stack Hub deployment mandatory PKI prerequisites</a><br><a href="#">Azure Stack Hub deployment PaaS certificate prerequisites</a> |
|                                                    |                                                                                                                                                     |

In a disconnected scenario, complete the following steps to download the required PowerShell modules and register the repository manually.

1. Sign in to a computer with internet connectivity and use the following scripts to download the PowerShell modules.

```
Import-Module -Name PowerShellGet -ErrorAction Stop
Import-Module -Name PackageManagement -ErrorAction Stop

path to save the packages, c:\temp\azs1.6.0 as an example here
$Path = "c:\temp\azs1.6.0"
Save-Package -ProviderName NuGet -Source https://www.powershellgallery.com/api/v2 -Name AzureRM -Path $Path -Force -RequiredVersion 2.3.0
Save-Package -ProviderName NuGet -Source https://www.powershellgallery.com/api/v2 -Name AzureStack -Path $Path -Force -RequiredVersion 1.6.0
```

2. Then you copy the downloaded packages to a USB device.
3. Sign in to the disconnected workstation and copy the packages from the USB device to a location on the workstation.
4. Register this location as a local repository.

```
requires -Version 5
requires -RunAsAdministrator
requires -Module PowerShellGet
requires -Module PackageManagement

$SourceLocation = "C:\temp\azs1.6.0"
$RepoName = "azs1.6.0"

Register-PSRepository -Name $RepoName -SourceLocation $SourceLocation -InstallationPolicy Trusted

New-Item -Path $env:ProgramFiles -name "SqlMySqlPsh" -ItemType "Directory"
```

## Certificates

*For integrated systems installations only.* You must provide the SQL PaaS PKI certificate described in the optional PaaS certificates section of [Azure Stack Hub deployment PKI requirements](#). Place the .pfx file in the location specified by the **DependencyFilesLocalPath** parameter. Don't provide a certificate for ASDK systems.

## Deploy the SQL resource provider

After you've installed all the prerequisites, run the **DeploySqlProvider.ps1** script from a computer that can access both the Azure Stack Hub Admin Azure Resource Management Endpoint and Privileged Endpoint to deploy the SQL resource provider. The DeploySqlProvider.ps1 script is extracted as part of the SQL resource provider binary that you downloaded for your version of Azure Stack Hub.

## IMPORTANT

Before deploying the resource provider, review the release notes to learn about new functionality, fixes, and any known issues that could affect your deployment.

To deploy the SQL resource provider, open a **new** elevated PowerShell window (not PowerShell ISE) and change to the directory where you extracted the SQL resource provider binary files. We recommend using a new PowerShell window to avoid potential problems caused by PowerShell modules that are already loaded.

Run the DeploySqlProvider.ps1 script, which completes the following tasks:

- Uploads the certificates and other artifacts to a storage account on Azure Stack Hub.
- Publishes gallery packages so you can deploy SQL databases using the gallery.
- Publishes a gallery package for deploying hosting servers.
- Deploys a VM using the Windows Server 2016 core image you downloaded, and then installs the SQL resource provider.
- Registers a local DNS record that maps to your resource provider VM.
- Registers your resource provider with the local Azure Resource Manager for the operator account.

## NOTE

When the SQL resource provider deployment starts, the **system.local.sqladapter** resource group is created. It may take up to 75 minutes to finish the required deployments to this resource group. You should not place any other resources in the **system.local.sqladapter** resource group.

## DeploySqlProvider.ps1 parameters

You can specify the following parameters from the command line. If you don't, or if any parameter validation fails, you're prompted to provide the required parameters.

| PARAMETER NAME              | DESCRIPTION                                                                                                                          | COMMENT OR DEFAULT VALUE |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| <b>CloudAdminCredential</b> | The credential for the cloud admin, necessary for accessing the privileged endpoint.                                                 | <i>Required</i>          |
| <b>AzCredential</b>         | The credentials for the Azure Stack Hub service admin account. Use the same credentials that you used for deploying Azure Stack Hub. | <i>Required</i>          |
| <b>VMLocalCredential</b>    | The credentials for the local admin account of the SQL resource provider VM.                                                         | <i>Required</i>          |
| <b>PrivilegedEndpoint</b>   | The IP address or DNS name of the privileged endpoint.                                                                               | <i>Required</i>          |

| Parameter Name                       | Description                                                                                                                                                                                                                                                                        | Comment or Default Value                           |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|
| <b>AzureEnvironment</b>              | The Azure environment of the service admin account used for deploying Azure Stack Hub. Required only for Azure AD deployments. Supported environment names are <b>AzureCloud</b> , <b>AzureUSGovernment</b> , or if using a China Azure Active Directory, <b>AzureChinaCloud</b> . | AzureCloud                                         |
| <b>DependencyFilesLocalPath</b>      | For integrated systems only, your certificate .pfx file must be placed in this directory. You can optionally copy one Windows Update MSU package here.                                                                                                                             | <i>Optional (mandatory for integrated systems)</i> |
| <b>DefaultSSLCertificatePassword</b> | The password for the .pfx certificate.                                                                                                                                                                                                                                             | <i>Required</i>                                    |
| <b>MaxRetryCount</b>                 | The number of times you want to retry each operation if there's a failure.                                                                                                                                                                                                         | 2                                                  |
| <b>RetryDuration</b>                 | The timeout interval between retries, in seconds.                                                                                                                                                                                                                                  | 120                                                |
| <b>Uninstall</b>                     | Removes the resource provider and all associated resources (see the following notes).                                                                                                                                                                                              | No                                                 |
| <b>DebugMode</b>                     | Prevents automatic cleanup on failure.                                                                                                                                                                                                                                             | No                                                 |

## Deploy the SQL resource provider using a custom script

If you're deploying the SQL resource provider version 1.1.33.0 or previous versions, you need to install specific versions of AzureRm.BootStrapper and Azure Stack Hub modules in PowerShell. If you're deploying the SQL resource provider version 1.1.47.0, the deployment script will automatically download and install the necessary PowerShell modules for you to path C:\Program Files\SqlMySqlPsh.

```
Install the AzureRM.Bootstrapper module, set the profile, and install the AzureStack module
Note that this might not be the most currently available version of Azure Stack Hub PowerShell
Install-Module -Name AzureRm.BootStrapper -RequiredVersion 0.5.0 -Force
Use-AzureRmProfile -Profile 2018-03-01-hybrid -Force
Install-Module -Name AzureStack -RequiredVersion 1.6.0
```

### NOTE

In disconnected scenario, you need to download the required PowerShell modules and register the repository manually as a prerequisite.

To eliminate any manual configuration when deploying the resource provider, you can customize the following script. Change the default account information and passwords as needed for your Azure Stack Hub deployment.

```

Use the NetBIOS name for the Azure Stack Hub domain. On the Azure Stack Hub SDK, the default is AzureStack
but could have been changed at install time.
$domain = "AzureStack"

For integrated systems, use the IP address of one of the ERCS VMs
$privilegedEndpoint = "AzS-ERCS01"

Provide the Azure environment used for deploying Azure Stack Hub. Required only for Azure AD deployments.
Supported values for the <environment name> parameter are AzureCloud, AzureChinaCloud, or AzureUSGovernment
depending which Azure subscription you're using.
$AzureEnvironment = "<EnvironmentName>"

Point to the directory where the resource provider installation files were extracted.
$tempDir = 'C:\TEMP\SQLRP'

The service admin account can be Azure Active Directory or Active Directory Federation Services.
$serviceAdmin = "admin@mydomain.onmicrosoft.com"
$AdminPass = ConvertTo-SecureString "P@ssw0rd1" -AsPlainText -Force
$AdminCreds = New-Object System.Management.Automation.PSCredential ($serviceAdmin, $AdminPass)

Set credentials for the new resource provider VM local admin account.
$vmLocalAdminPass = ConvertTo-SecureString "P@ssw0rd1" -AsPlainText -Force
$vmLocalAdminCreds = New-Object System.Management.Automation.PSCredential ("sqlrpadmin", $vmLocalAdminPass)

Add the cloudadmin credential that's required for privileged endpoint access.
$CloudAdminPass = ConvertTo-SecureString "P@ssw0rd1" -AsPlainText -Force
$CloudAdminCreds = New-Object System.Management.Automation.PSCredential ("$domain\cloudadmin",
$CloudAdminPass)

Change the following as appropriate.
$PfxPass = ConvertTo-SecureString "P@ssw0rd1" -AsPlainText -Force

For version 1.1.47.0, the PowerShell modules used by the RP deployment are placed in C:\Program
Files\SqlMySqlPsh
The deployment script adds this path to the system $env:PSModulePath to ensure correct modules are used.
$rpModulePath = Join-Path -Path $env:ProgramFiles -ChildPath 'SqlMySqlPsh'
$env:PSModulePath = $env:PSModulePath + ";" + $rpModulePath

Change to the directory folder where you extracted the installation files. Don't provide a certificate on
ASDK!
. $tempDir\DeploySQLProvider.ps1 `
 -AzCredential $AdminCreds `
 -VMLocalCredential $vmLocalAdminCreds `
 -CloudAdminCredential $cloudAdminCreds `
 -PrivilegedEndpoint $privilegedEndpoint `
 -AzureEnvironment $AzureEnvironment `
 -DefaultSSLCertificatePassword $PfxPass `
 -DependencyFilesLocalPath $tempDir\cert

```

When the resource provider installation script finishes, refresh your browser to make sure you can see the latest updates and close the current PowerShell session.

## Verify the deployment using the Azure Stack Hub portal

You can use the following steps verify that the SQL resource provider is successfully deployed.

1. Sign in to the administrator portal as the service admin.
2. Select **Resource Groups**.
3. Select the **system.<location>.sqladapter** resource group.
4. On the summary page for Resource group Overview, there should be no failed deployments.

The screenshot shows the Azure portal's 'Essentials' blade. At the top, there are navigation icons for 'Add', 'Columns', 'Delete resource group', 'Refresh', and 'Move'. Below these are sections for 'Subscription name (change)', 'Default Provider Subscription', and 'Subscription ID'. On the right side, there is a summary box with the heading 'Deployments' and the sub-heading '4 Succeeded', which is highlighted with a red box.

5. Finally, select **Virtual machines** in the administrator portal to verify that the SQL resource provider VM was successfully created and is running.

## Next steps

[Add hosting servers](#)

# Add hosting servers for the SQL resource provider

7 minutes to read • [Edit Online](#)

You can create SQL Server database hosting servers on a virtual machine (VM) in [Azure Stack Hub](#), or on a VM outside your Azure Stack Hub environment, as long as the SQL resource provider can connect to the instance.

## NOTE

The SQL resource provider should be created in the default provider subscription while SQL hosting servers should be created in a billable, user subscription. The resource provider server shouldn't be used to host user databases.

## Overview

Before you add a SQL hosting server, review the following mandatory and general requirements.

### Mandatory requirements

- Enable SQL authentication on the SQL Server instance. Because the SQL resource provider VM isn't domain-joined, it can only connect to a hosting server using SQL authentication.
- Configure the IP addresses for the SQL instances as Public when installed on Azure Stack Hub. The resource provider and users, such as web apps, communicate over the user network, so connectivity to the SQL instance on this network is required.

### General requirements

- Dedicate the SQL instance for use by the resource provider and user workloads. You can't use a SQL instance that's being used by any other consumer. This restriction also applies to App Services.
- Configure an account with the appropriate privilege levels for the resource provider (described below).
- You're responsible for managing the SQL instances and their hosts. For example, the resource provider doesn't apply updates, handle backups, or handle credential rotation.

### SQL Server VM images

SQL IaaS VM images are available through the Marketplace Management feature. These images are the same as the SQL VMs that are available in Azure.

Make sure you always download the latest version of the **SQL IaaS Extension** before you deploy a SQL VM using a Marketplace item. The IaaS extension and corresponding portal enhancements provide additional features such as automatic patching and backup. For more information about this extension, see [Automate management tasks on Azure VMs with the SQL Server Agent Extension](#).

## NOTE

The SQL IaaS Extension is *required* for all SQL on Windows images in the marketplace; the VM will fail to deploy if you didn't download the extension. It's not used with Linux-based SQL VM images.

There are other options for deploying SQL VMs, including templates in the [Azure Stack Hub Quickstart Gallery](#).

#### **NOTE**

Any hosting servers installed on a multi-node Azure Stack Hub must be created from a user subscription and not the Default Provider Subscription. They must be created from the user portal or from a PowerShell session with an appropriate login. All hosting servers are billable VMs and must have appropriate SQL licenses. The service admin *can* be the owner of that subscription.

## **Required Privileges**

You can create an admin user with lower privileges than a SQL sysadmin. The user only needs permissions for the following operations:

- Database: Create, Alter, With Containment (for Always On only), Drop, Backup
- Availability Group: Alter, Join, Add/Remove Database
- Login: Create, Select, Alter, Drop, Revoke
- Select Operations: [master].[sys].[availability\_group\_listeners] (AlwaysOn), sys.availability\_replicas (AlwaysOn), sys.databases, [master].[sys].[dm\_os\_sys\_memory], SERVERPROPERTY, [master].[sys].[availability\_groups] (AlwaysOn), sys.master\_files

## **Additional Security Information**

The following information provides additional security guidance:

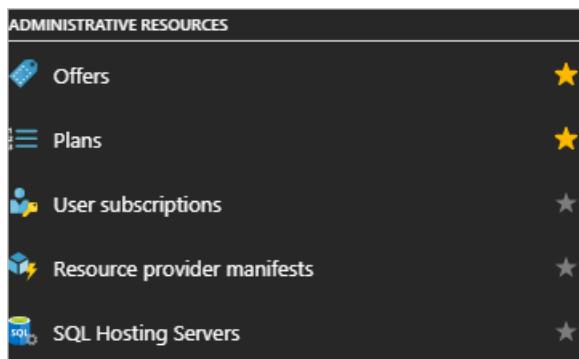
- All Azure Stack Hub storage is encrypted using BitLocker, so any SQL instance on Azure Stack Hub will use encrypted blob storage.
- The SQL Resource Provider fully supports TLS 1.2. Ensure that any SQL Server that's managed through the SQL RP is configured for TLS 1.2 *only* and the RP will default to that. All supported versions of SQL Server support TLS 1.2. For more information, see [TLS 1.2 support for Microsoft SQL Server](#).
- Use SQL Server Configuration Manager to set the **ForceEncryption** option to ensure all communications to the SQL server are always encrypted. For more information, see [To configure the server to force encrypted connections](#).
- Ensure any client app is also communicating over an encrypted connection.
- The RP is configured to trust the certificates used by the SQL Server instances.

## Provide capacity by connecting to a standalone hosting SQL server

You can use standalone (non-HA) SQL servers using any edition of SQL Server 2014 or SQL Server 2016. Make sure you have the credentials for an account with sysadmin privileges.

To add a standalone hosting server that's already set up, follow these steps:

1. Sign in to the Azure Stack Hub administrator portal as a service admin.
2. Select **All services > ADMINISTRATIVE RESOURCES > SQL Hosting Servers**.



Under **SQL Hosting Servers**, you can connect the SQL resource provider to instances of SQL Server that

will serve as the resource provider's backend.

The screenshot shows two adjacent blades in the Azure portal:

- SQL Adapter** (Resource provider):
  - Essentials**: Shows a single hosting server named "SQLAdapter".
  - Monitoring**: Displays capacity usage with a donut chart showing 0.258% ALLOCATED (0.26 GB) and 99.74 GB AVAILABLE.
  - Hosting Server Management**: Shows 1 Hosting Server.
- SQL Hosting Servers** (All Hosting Servers in this region):
  - Add** button.
  - Table with columns: NAME, DATABASE COUNT, CAPACITY (GB), SKU. One row is listed: sqldev.local.cloudapp...., 3 databases, 100 GB capacity, SQLSKU.

3. Click **Add** and then provide the connection details for your SQL Server instance on the **Add a SQL Hosting Server** blade.

Add a SQL Hosting Server

\* SQL Server Name ⓘ  
[SQL Server FQDN or IPv4[Port] or \InstanceName]

\* Username  
Hosting Server SQL login

\* Password  
Password of the SQL Hosting Server

Size of Hosting Server in GB

Always On Availability Group ⓘ

Subscription  
Default Provider Subscription

\* Resource group ⓘ  
 Create new    Use existing  
[Resource Group Name]

\* Location  
local

\* SKUs  
None >

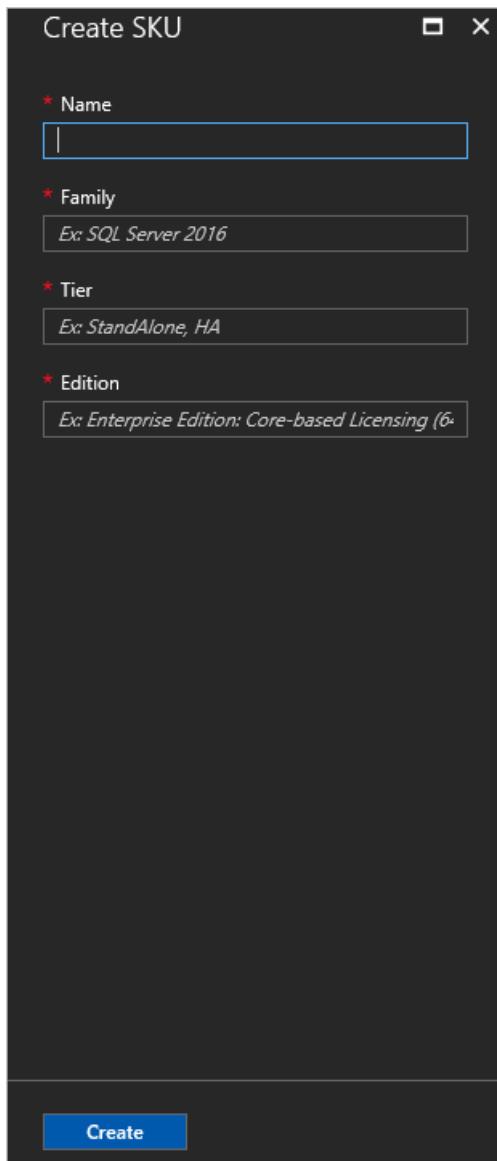
**Create**

Optionally, provide an instance name, and specify a port number if the instance isn't assigned to the default port of 1433.

**NOTE**

As long as the SQL instance can be accessed by the user and admin Azure Resource Manager, it can be placed under control of the resource provider. The SQL instance **must** be allocated exclusively to the resource provider.

4. As you add servers, you must assign them to an existing SKU or create a new SKU. Under **Add a Hosting Server**, select **SKUs**.
  - To use an existing SKU, choose an available SKU and then select **Create**.
  - To create a SKU, select + **Create new SKU**. In **Create SKU**, enter the required information, and then select **OK**.



## Provide high availability using SQL Always On Availability Groups

Configuring SQL Always On instances requires additional steps and requires three VMs (or physical machines). This article assumes that you already have a solid understanding of Always On availability groups. For more information, see the following articles:

- [Introducing SQL Server Always On availability groups on Azure virtual machines](#)
- [Always On Availability Groups \(SQL Server\)](#)

### NOTE

The SQL adapter resource provider *only* supports SQL 2016 SP1 Enterprise or later instances for Always On Availability Groups. This adapter configuration requires new SQL features such as automatic seeding.

### Automatic seeding

You must enable [Automatic Seeding](#) on each availability group for each instance of SQL Server.

To enable automatic seeding on all instances, edit and then run the following SQL command on the primary replica for each secondary instance:

```
ALTER AVAILABILITY GROUP [<availability_group_name>]
 MODIFY REPLICA ON '<secondary_node>'
 WITH (SEEDING_MODE = AUTOMATIC)
GO
```

The availability group must be enclosed in square brackets.

On the secondary nodes, run the following SQL command:

```
ALTER AVAILABILITY GROUP [<availability_group_name>] GRANT CREATE ANY DATABASE
GO
```

### Configure contained database authentication

Before adding a contained database to an availability group, ensure that the contained database authentication server option is set to 1 on every server instance that hosts an availability replica for the availability group. For more information, see [contained database authentication Server Configuration Option](#).

Use these commands to set the contained database authentication server option for each instance:

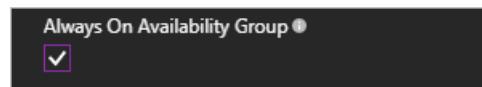
```
EXEC sp_configure 'contained database authentication', 1
GO
RECONFIGURE
GO
```

### To add SQL Always On hosting servers

1. Sign in to the Azure Stack Hub administrator portal as a service admin.
2. Select **Browse > ADMINISTRATIVE RESOURCES > SQL Hosting Servers > +Add**.

Under **SQL Hosting Servers**, you can connect the SQL Server Resource Provider to actual instances of SQL Server that serve as the resource provider's backend.

3. Fill out the form with the connection details for your SQL Server instance. Make sure that you use the FQDN address of the Always On Listener (and optional port number and instance name). Provide the information for the account you configured with sysadmin privileges.
4. Check the Always On Availability Group box to enable support for SQL Always On Availability Group instances.



5. Add the SQL Always On instance to a SKU.

#### IMPORTANT

You can't mix standalone servers with Always On instances in the same SKU. Attempting to mix types after adding the first hosting server results in an error.

## SKU notes

Use a SKU name that describes the capabilities of the servers in the SKU, such as capacity and performance. The name serves as an aid to help users deploy their databases to the appropriate SKU. For example, you can use SKU names to differentiate service offerings by the following characteristics:

- high capacity
- high performance
- high availability

As a best practice, all the hosting servers in a SKU should have the same resource and performance characteristics.

SKUs can't be assigned to specific users or groups.

SKUs can take up to an hour to be visible in the portal. Users can't create a database until the SKU is fully created.

To edit a SKU, go to **All services > SQL Adapter > SKUs**. Select the SKU to modify, make any necessary changes, and click **Save** to save changes.

To delete a SKU that's no longer needed, go to **All services > SQL Adapter > SKUs**. Right-click the SKU name and select **Delete** to delete it.

**IMPORTANT**

It can take up to an hour for new SKUs to be available in the user portal.

## Make SQL databases available to users

Create plans and offers to make SQL databases available for users. Add the **Microsoft.SqlAdapter** service to the plan and create a new quota.

**IMPORTANT**

It can take up to two hours for new quotas to be available in the user portal or before a changed quota is enforced.

## Next steps

[Add databases](#)

# Create SQL databases

2 minutes to read • [Edit Online](#)

You can create and manage self-service databases in the user portal. An Azure Stack Hub user needs a subscription with an offer that includes the SQL database service.

1. Sign in to the [Azure Stack Hub](#) user portal.
2. Select + **New > Data + Storage > SQL Server Database > Add.**
3. Under **Create Database**, enter the required information, such as **Database Name** and **Max Size in MB**.

## NOTE

The database size must be at least 64 MB, which can be increased after you deploy the database.

Configure the other settings as required for your environment.

4. Under **Create Database**, select **SKU**. Under **Select a SKU**, select the SKU for your database.

### Create Database

\* Database Name  
SQLTestDB

\* Collation ⓘ  
SQL\_Latin1\_General\_CI\_AS

\* Max Size in MB  
10000

\* Subscription  
DataSvcSub

\* Resource Group  
SQLTestRG  
[Create new](#)

\* Location  
shanghai

\* SKU  
None Selected >

\* Login  
*Configure required settings*

### Select a SKU

|            |            |         |
|------------|------------|---------|
| sql2016std | StandAlone | SQL2... |
|------------|------------|---------|

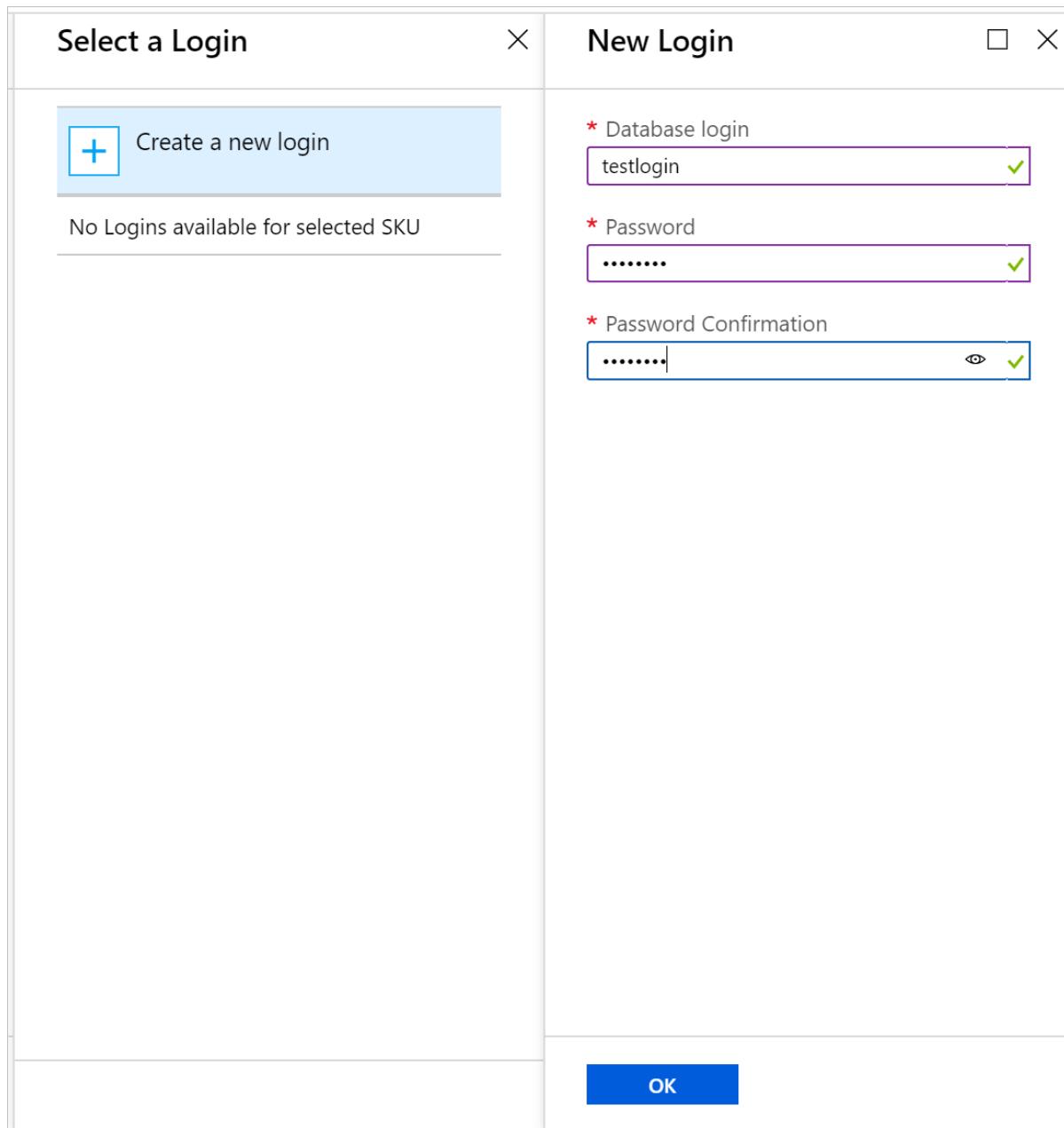
**NOTE**

As hosting servers are added to Azure Stack Hub, they're assigned a SKU. Databases are created in the pool of hosting servers in a SKU.

5. Select **Login**.
6. Under **Select a Login**, choose an existing login, or select + **Create a new login**.
7. Under **New Login**, enter a name for **Database login** and a **Password**.

**NOTE**

These settings are the SQL authentication credential that's created for your access to this database only. The login user name must be globally unique. You can reuse login settings for other databases that use the same SKU.



8. Select **OK** to finish deploying the database.

Under **Essentials**, which is shown after the database is deployed, take note of the **Connection string**. You can use this string in any app that needs to access the SQL Server database.

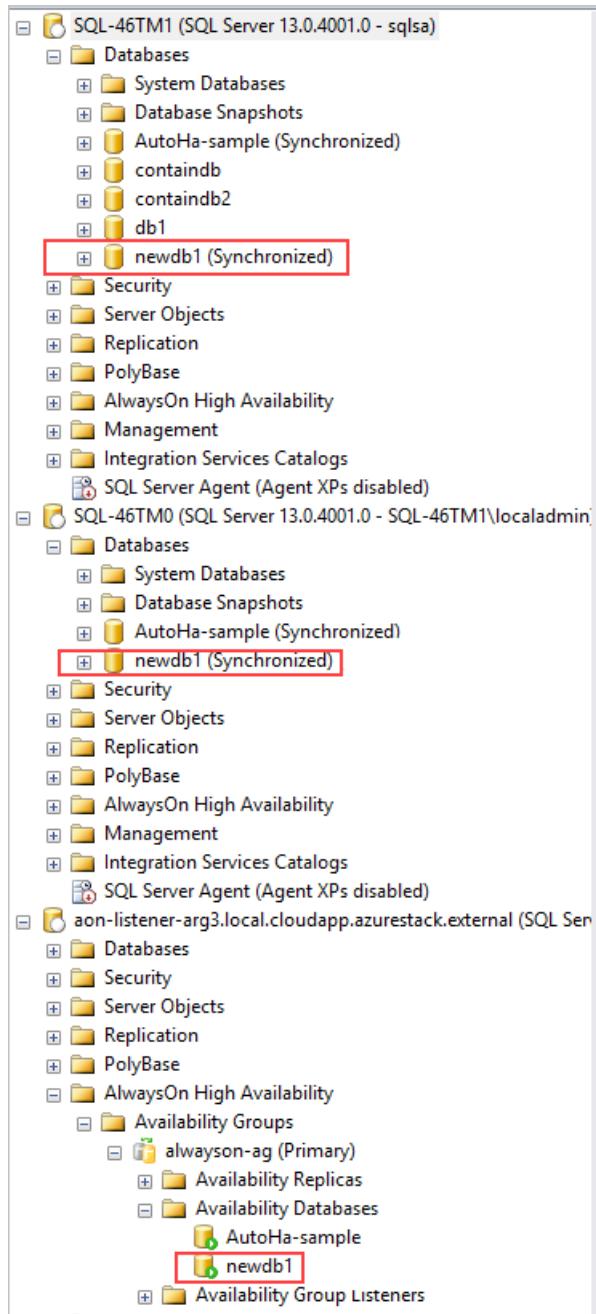
| Dashboard > SQL Databases > SQLTestDB          |                                                                      |
|------------------------------------------------|----------------------------------------------------------------------|
| <b>SQL Databases</b><br>Microsoft Selfhost     | <b>SQLTestDB</b><br>SQL Database                                     |
| <b>Add</b> <b>Edit columns</b> <b>Refresh</b>  |                                                                      |
| <input type="text" value="Filter by name..."/> | <b>Essentials</b>                                                    |
| <b>NAME</b>                                    | Resource group<br><b>SQLTestRG</b>                                   |
| <b>SQLTestDB</b>                               | Location<br>shanghai                                                 |
|                                                | Subscription name<br><b>DataSvcSub</b>                               |
|                                                | Subscription ID<br>09eb8b77-xxxx-xxxx-9ce4-b5132d1d8fd4              |
|                                                | Name<br><b>SQLTestDB</b>                                             |
|                                                | Connection String<br>Data Source=10.156.100.109,1433;Initial Cata... |
|                                                | SKU<br>sql2016std                                                    |
|                                                | <a href="#">All settings →</a>                                       |

## SQL Always On databases

By design, Always On databases are handled differently than in a standalone server environment. For more information, see [Introducing SQL Server Always On availability groups on Azure virtual machines](#).

## Verify SQL Always On databases

The following screen capture shows how you can use SQL Server Management Studio to look at database status in SQL Always On.



Always On databases should show as **Synchronized** and available on all the SQL instances and appear in **Availability Groups**. In the previous screenshot, the database example is newdb1 and its status is **newdb1 (Synchronized)**.

## Delete an Always On database

When you delete a SQL Always On database from the resource provider, SQL deletes the database from the **Primary** replica and from the availability group.

SQL then puts the database into the **Restoring** state on the other replicas and doesn't drop the database unless triggered. If the database isn't dropped, the secondary replicas go into a **Not Synchronizing** state.

## Next steps

Learn how to [offer highly available SQL databases](#)

# Create highly available SQL databases with Azure Stack Hub

7 minutes to read • [Edit Online](#)

As an Azure Stack Hub Operator, you can configure server VMs to host SQL Server databases. After a SQL hosting server is created and managed by Azure Stack Hub, users who have subscribed to SQL services can easily create SQL databases.

This article shows how to use an Azure Stack Hub quickstart template to create a [SQL Server AlwaysOn availability group](#), add it as an Azure Stack Hub SQL Hosting Server, and then create a highly available SQL database.

What you'll learn:

- Create a SQL Server AlwaysOn availability group from a template.
- Create an Azure Stack Hub SQL Hosting Server.
- Create a highly available SQL database.

A two VM SQL Server AlwaysOn availability group will be created and configured using available Azure Stack Marketplace items.

Before starting, ensure that the [SQL Server resource provider](#) has been successfully installed and the following items are available in Azure Stack Marketplace:

## IMPORTANT

All of the following are required for the Azure Stack Hub quickstart template to be used.

- [Windows Server 2016 Datacenter](#) marketplace image.
- SQL Server 2016 SP1 or SP2 (Enterprise, Standard, or Developer) on Windows Server 2016 server image. This article uses the [SQL Server 2016 SP2 Enterprise on Windows Server 2016](#) marketplace image.
- [SQL Server IaaS Extension](#) version 1.2.30 or higher. The SQL IaaS Extension installs necessary components that are required by the Marketplace SQL Server items for all Windows versions. It enables SQL-specific settings to be configured on SQL virtual machines (VMs). If the extension isn't installed in the local marketplace, provisioning of SQL will fail.
- [Custom script extension for Windows](#) version 1.9.1 or higher. Custom Script Extension is a tool that can be used to automatically launch post-deployment VM customization tasks.
- [PowerShell Desired State Configuration \(DSC\)](#) version 2.76.0.0 or higher. DSC is a management platform in Windows PowerShell that enables deploying and managing configuration data for software services. The platform also manages the environment in which these services run.

To learn more about adding items to Azure Stack Marketplace, see the [Azure Stack Hub Marketplace overview](#).

## Create a SQL Server AlwaysOn availability group

Use the steps in this section to deploy the SQL Server AlwaysOn availability group by using the [sql-2016-alwayson Azure Stack Hub quickstart template](#). This template deploys two SQL Server Enterprise, Standard or Developer instances in an Always On Availability Group. It creates the following resources:

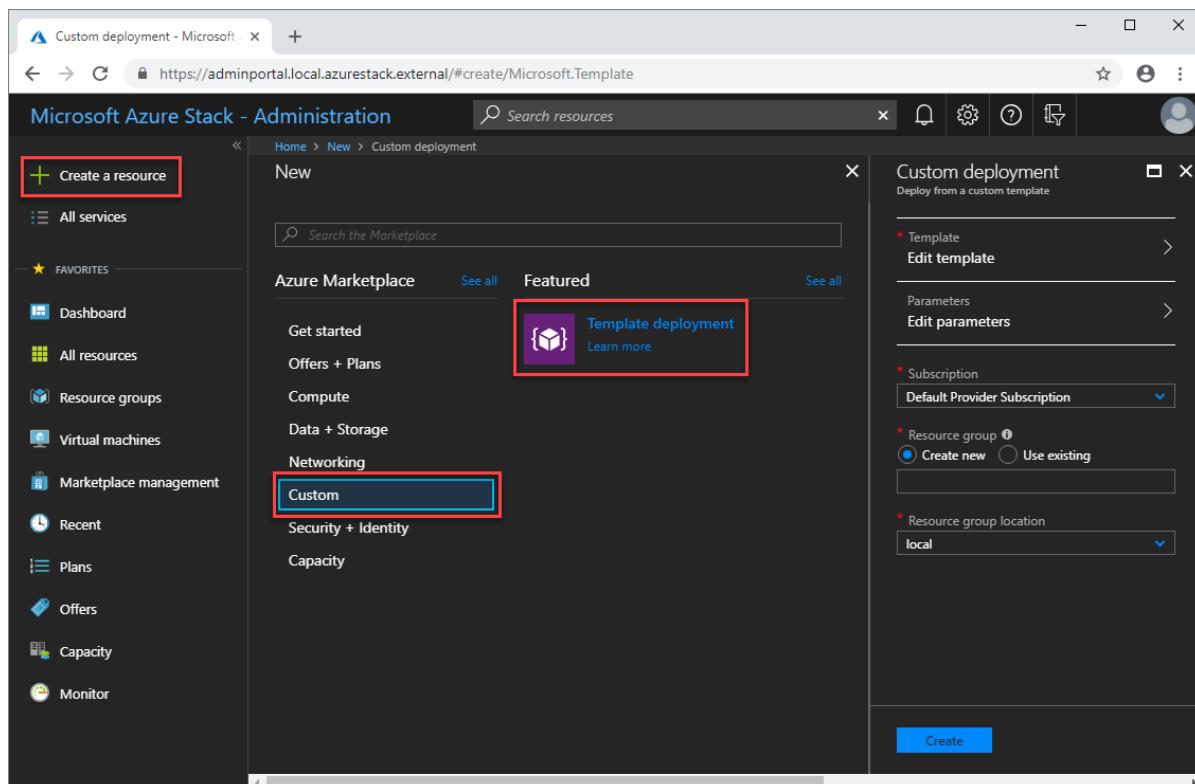
- A network security group.

- A virtual network.
- Four storage accounts (one for Active Directory (AD), one for SQL, one for file share witness, and one for VM diagnostics).
- Four public IP addresses (one for AD, two for each SQL VM, and one for public load balancer bound to SQL AlwaysOn listener).
- One external load balancer for SQL VMs with Public IP bound to the SQL AlwaysOn listener.
- One VM (Windows Server 2016) configured as Domain Controller for a new forest with a single domain.
- Two VMs (Windows Server 2016) configured with SQL Server 2016 SP1 or SP2 Enterprise, Standard, or Developer Edition and clustered. These must be marketplace images.
- One VM (Windows Server 2016) configured as the file share witness for the cluster.
- One availability set containing the SQL and file share witness VMs.

1. Sign in to the administrator portal:

- For an integrated system deployment, the portal address varies based on your solution's region and external domain name. The address is in this format: <https://adminportal.<region>.<FQDN>>.
- For the Azure Stack Development Kit (ASDK), the portal address is <https://adminportal.local.azurestack.external>.

2. Select + **Create a resource** > **Custom**, and then **Template deployment**.



3. On the **Custom deployment** blade, select **Edit template** > **Quickstart template** and then use the drop-down list of available custom templates to select the **sql-2016-alwayson** template. Select **OK**, then **Save**.

The screenshot shows the 'Edit template' blade in the Azure portal. At the top, there's a breadcrumb navigation: Home > New > Custom deployment > Edit template. Below it, the title 'Edit template' and a subtitle 'Edit your Azure Resource Manager template' are displayed. There are three buttons: 'Quickstart template' (with an upward arrow icon), 'Load file' (with a downward arrow icon), and 'Download' (with a download icon). A section titled 'Load a quickstart template' contains a dropdown menu with the option 'sql-2016-alwayson' selected. This option is highlighted with a red box. Below the dropdown, a note states: 'This template creates 4 AzureStack VMs with Active Directory and SQL Server Always On'. It also shows the author 'azurestack' and the last updated date '2018-08-27', with a 'Learn more' link. At the bottom are 'OK' and 'Cancel' buttons.

4. On the **Custom deployment** blade, select **Edit parameters** and review the default values. Modify the values as necessary to provide all required parameter information and then select **OK**.

At a minimum:

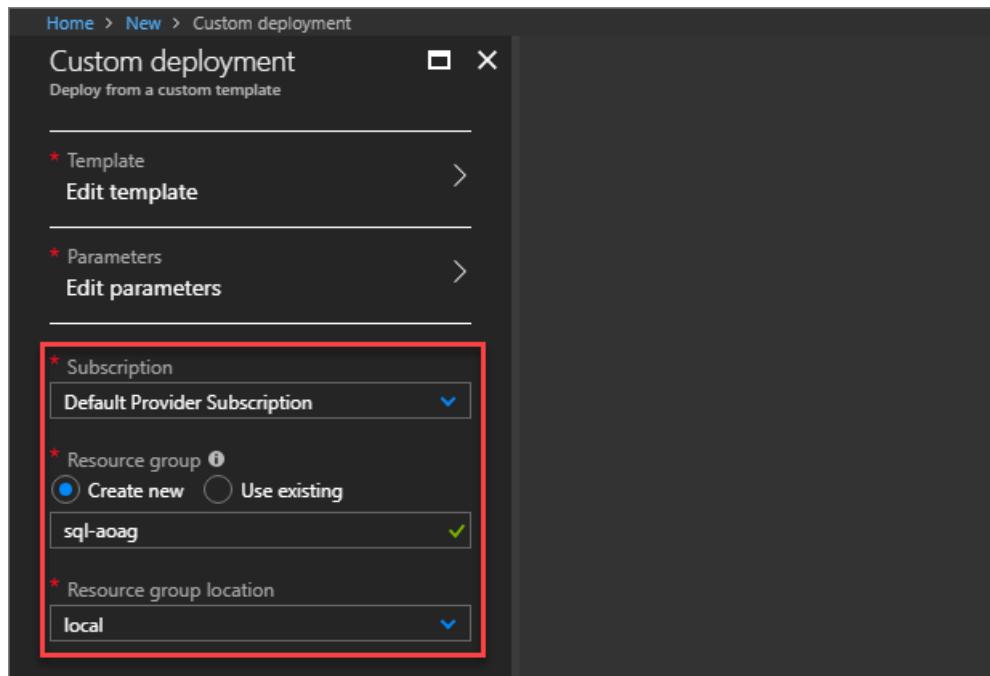
- Provide complex passwords for the ADMINPASSWORD, SQLSERVERSERVICEACCOUNTPASSWORD, and SQLAUTHPASSWORD parameters.
- Enter the DNS Suffix for reverse lookup in all lowercase letters for the DNSSUFFIX parameter (**azurestack.external** for ASDK installations).

The screenshot shows the 'Parameters' blade. On the left, under 'Custom deployment' settings, there's a 'Template' section with 'Edit template' and a 'Parameters' section with 'Edit parameters' (which is highlighted with a red box). On the right, the 'Parameters' section lists several parameters with their current values and validation status:

| Parameter                      | Value                                                                                               | Status |
|--------------------------------|-----------------------------------------------------------------------------------------------------|--------|
| ARTIFACTSLOCATION (string)     | <a href="https://raw.githubusercontent.com/Azure...">https://raw.githubusercontent.com/Azure...</a> | Valid  |
| ADMINUSERNAME (string)         | localadmin                                                                                          | Valid  |
| * ADMINPASSWORD (securestring) | .....                                                                                               | Valid  |
| ADVMSIZE (string)              | Standard_D2_v2                                                                                      | Valid  |
| WITNESSVMSIZE (string)         | Standard_D1_v2                                                                                      | Valid  |

5. On the **Custom deployment** blade, choose the subscription to use and create a new resource group or select an existing resource group for the custom deployment.

Next, select the resource group location (**local** for ASDK installations) and then click **Create**. The custom deployment settings will be validated and then the deployment will start.



6. In the administrator portal, select **Resource groups** and then the name of the resource group you created for the custom deployment (**resource-group** for this example). View the status of the deployment to ensure all deployments have completed successfully.

Next, review the resource group items and select the **SQLPIPsql<resource group name>** public IP address item. Record the public IP address and full FQDN of the load balancer public IP. You'll need to provide this to an Azure Stack Hub operator so they can create a SQL hosting server leveraging this SQL AlwaysOn availability group.

#### NOTE

The template deployment will take several hours to complete.

#### Enable automatic seeding

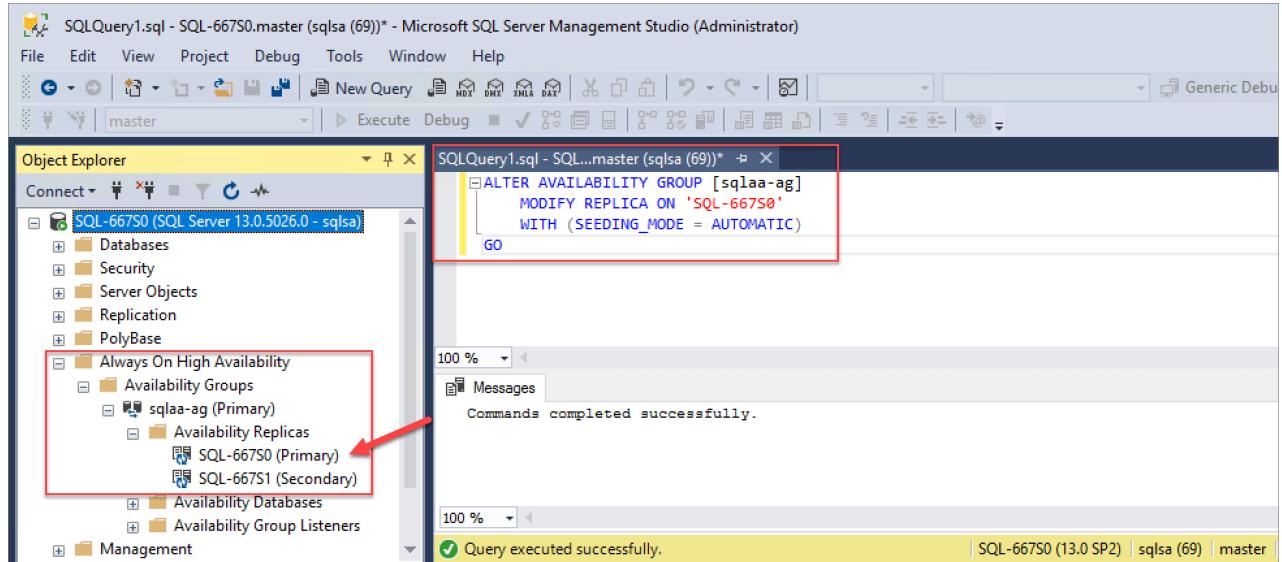
After the template has successfully deployed and configured the SQL AlwaysON availability group, you must enable **automatic seeding** on each instance of SQL Server in the availability group.

When you create an availability group with automatic seeding, SQL Server automatically creates the secondary replicas for every database in the group without any other manual intervention necessary. This measure ensures high availability of AlwaysOn databases.

Use these SQL commands to configure automatic seeding for the AlwaysOn availability group. Replace <InstanceName> with the primary instance SQL Server name and <availability\_group\_name> with the AlwaysOn availability group name as necessary.

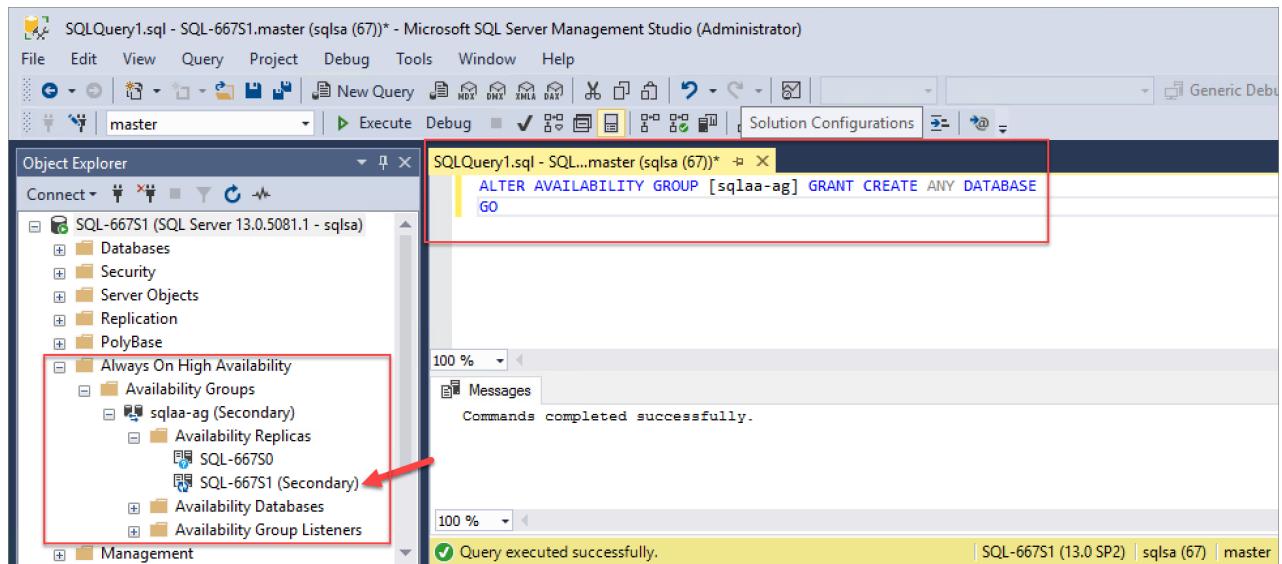
On the primary SQL instance:

```
ALTER AVAILABILITY GROUP [<availability_group_name>]
 MODIFY REPLICA ON '<InstanceName>'
 WITH (SEEDING_MODE = AUTOMATIC)
GO
```



On secondary SQL instances:

```
ALTER AVAILABILITY GROUP [<availability_group_name>] GRANT CREATE ANY DATABASE
GO
```



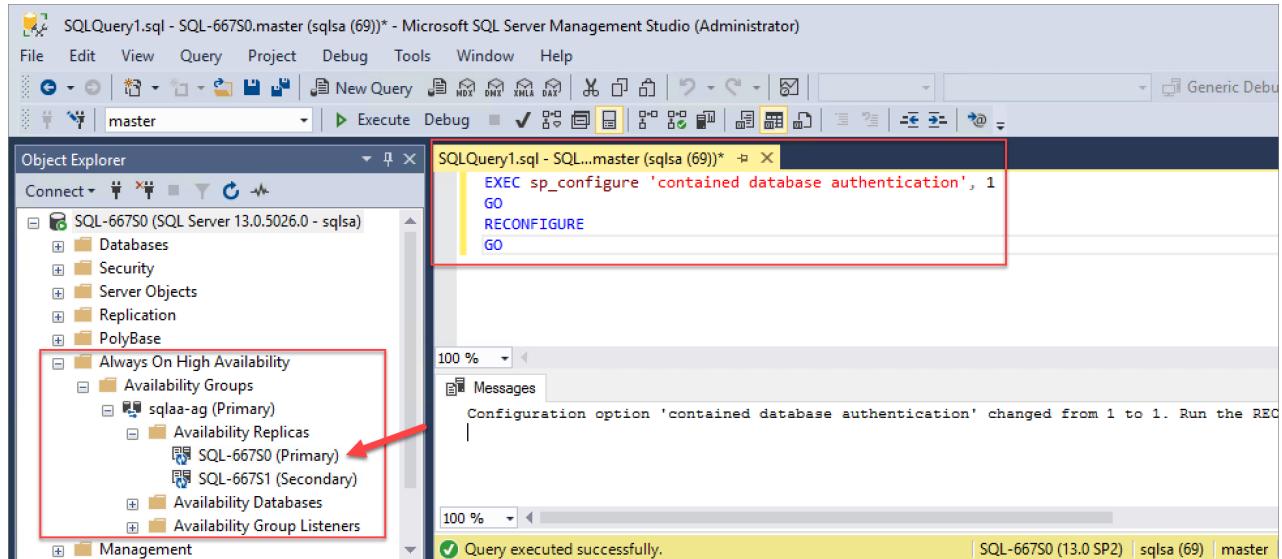
### Configure contained database authentication

Before adding a contained database to an availability group, ensure that the contained database authentication server option is set to 1 on every server instance that hosts an availability replica for the availability group. For

more information, see [contained database authentication](#).

Use these commands to set the contained database authentication server option for each SQL Server instance in the availability group:

```
EXEC sp_configure 'contained database authentication', 1
GO
RECONFIGURE
GO
```



## Create an Azure Stack Hub SQL Hosting Server

After the SQL Server AlwaysOn availability group has been created and properly configured, an Azure Stack Hub operator has to create an Azure Stack Hub SQL Hosting Server. The SQL Hosting Server makes the additional capacity available for users to create databases.

Be sure to use the public IP or full FQDN for the public IP of the SQL load balancer recorded previously when the SQL AlwaysOn availability group's resource group was created (**SQLIPs`<resource group name>`**). In addition, you need to know the SQL Server authentication credentials used to access the SQL instances in the AlwaysOn availability group.

### NOTE

This step must be run from the Azure Stack Hub administrator portal by an Azure Stack Hub operator.

With the SQL AlwaysOn availability group's load balancer listener public IP and SQL authentication login information, an Azure Stack Hub operator can [create a SQL Hosting Server using the SQL AlwaysOn availability group](#).

Also ensure that you have created plans and offers to make SQL AlwaysOn database creation available for users. The operator will need to add the **Microsoft.SqlAdapter** service to a plan and create a new quota specifically for highly available databases. For more information about creating plans, see [Service, plan, offer, subscription overview](#).

### TIP

The **Microsoft.SqlAdapter** service won't be available to add to plans until the [SQL Server resource provider](#) has been deployed.

# Create a highly available SQL database

After the SQL AlwaysOn availability group has been created, configured, and added as an Azure Stack Hub SQL Hosting Server by an Azure Stack Hub operator, a tenant user with a subscription including SQL Server database capabilities can create SQL databases supporting AlwaysOn functionality. They can create those databases by following the steps in this section.

## NOTE

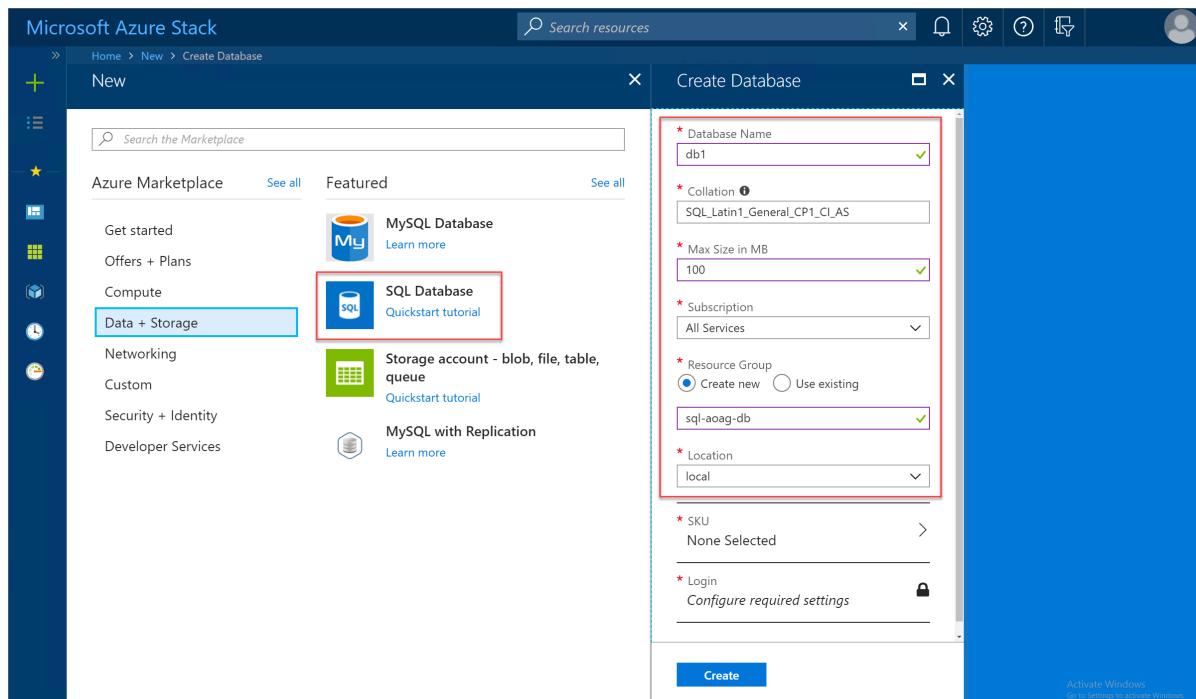
Run these steps from the Azure Stack Hub user portal as a tenant user with a subscription providing SQL Server capabilities (Microsoft.SQLAdapter service).

### 1. Sign in to the user portal:

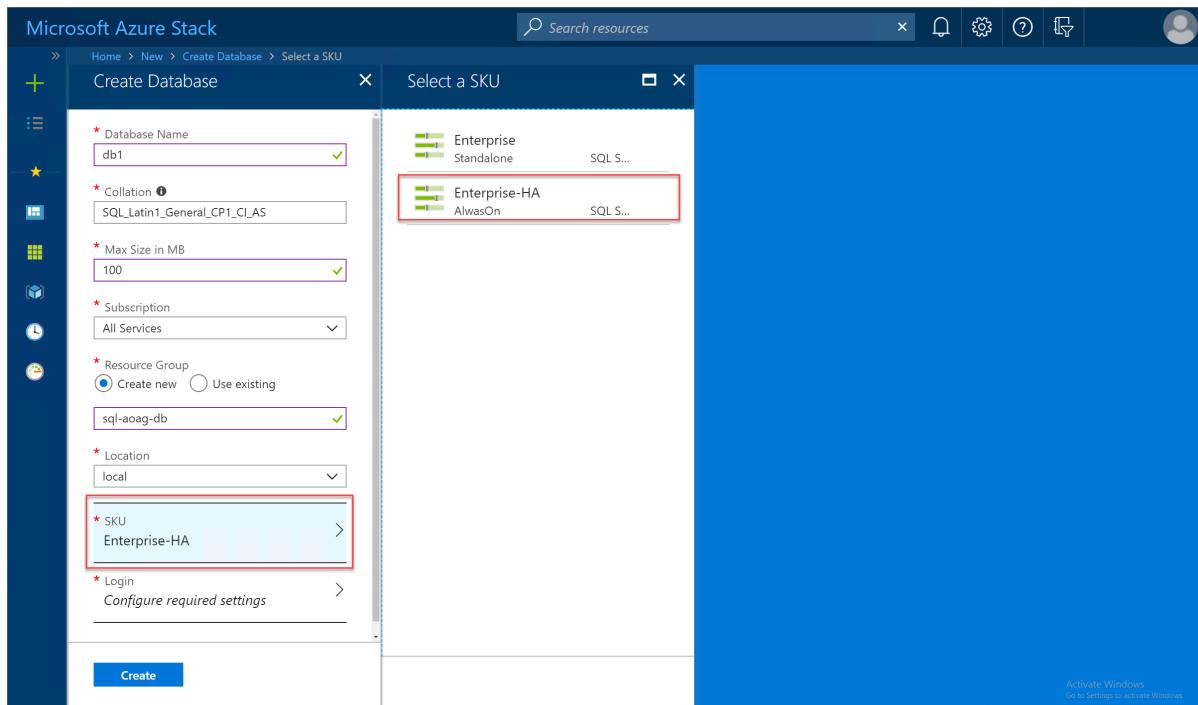
- For an integrated system deployment, the portal address will vary based on your solution's region and external domain name. It will be in the format of <https://portal.<region>.<FQDN>>.
- For the Azure Stack Development Kit (ASDK), the portal address is <https://portal.local.azurestack.external>.

### 2. Select + Create a resource > Data + Storage, and then SQL Database.

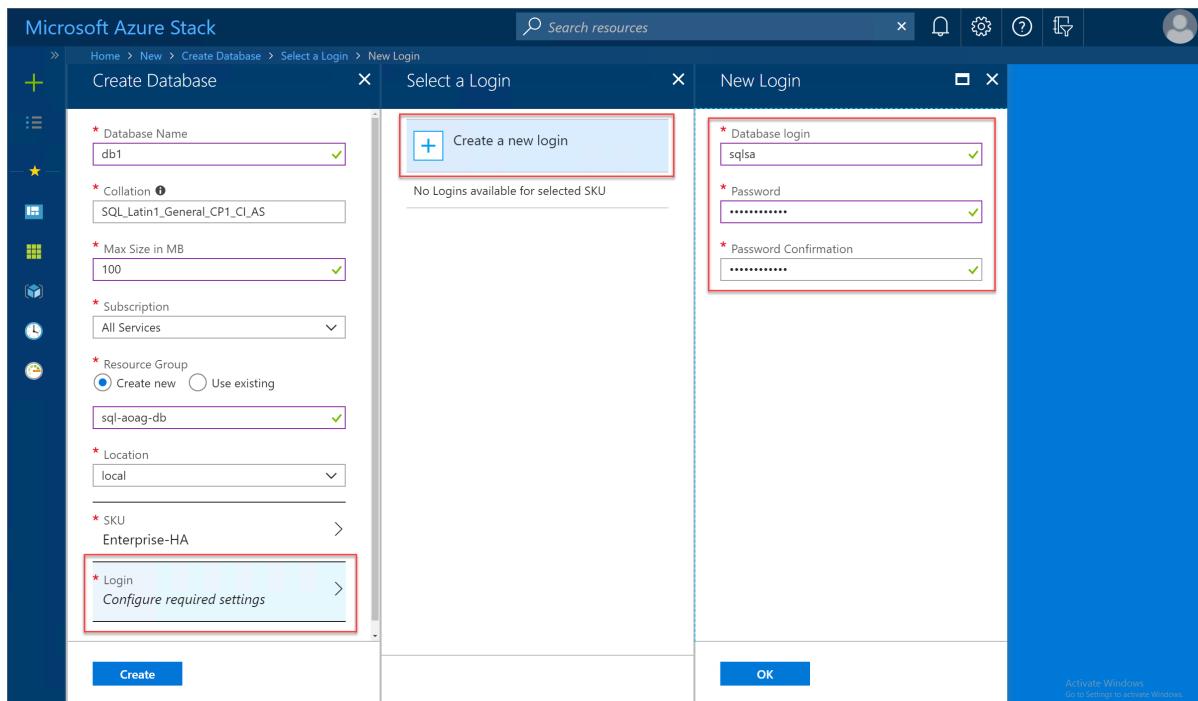
Provide the required database property information. This info includes name, collation, maximum size, and the subscription, resource group, and location to use for the deployment.



### 3. Select SKU and then choose the appropriate SQL Hosting Server SKU to use. In this example, the Azure Stack Hub operator has created the Enterprise-HA SKU to support high availability for SQL AlwaysOn availability groups.



4. Select **Login** > **Create a new login** and then provide the SQL authentication credentials to be used for the new database. When finished, select **OK** and then **Create** to begin the database deployment process.



5. When the SQL database deployment completes successfully, review the database properties to discover the connection string to use for connecting to the new highly available database.

The screenshot shows the Microsoft Azure Stack interface. The top navigation bar includes 'Search resources' and various icons for notifications, settings, and help. The main title is 'Microsoft Azure Stack'. Below it, the breadcrumb navigation shows 'Home > SQL Databases > db1 > Settings'. The left sidebar has a '+' icon, a star icon, and several other icons for different services. The main content area is titled 'SQL Databases' and shows a list with one item: 'db1' (SQL Database). A red box highlights the 'Settings' link next to the database name. The 'db1' card shows the following details:

| Essentials        |               |
|-------------------|---------------|
| Name              | db1           |
| Location          | local         |
| Subscription name | All Services  |
| Subscription ID   | [REDACTED]    |
| SKU               | Enterprise-HA |

A red box highlights the 'Connection String' section, which contains the value: 'Data Source=ao-listen-sql-alwayson.local.clo...'. There is also a 'All settings →' button at the bottom right of the card.

On the right side, there is a vertical sidebar with sections: 'SUPPORT + TROUBLESHOOTING' (Activity log), 'GENERAL' (Properties), and 'RESOURCE MANAGEMENT' (Locks). At the bottom right of the sidebar, it says 'Activate Windows Go to Settings to activate Windows.'

## Next steps

[Update the SQL resource provider](#)

# Update the SQL resource provider

4 minutes to read • [Edit Online](#)

A new SQL resource provider might be released when Azure Stack Hub is updated to a new build. Although the existing resource provider continues to work, we recommend updating to the latest build as soon as possible.

Starting with the SQL resource provider version 1.1.33.0 release, updates are cumulative and don't need to be installed in the order in which they were released as long as you're starting from version 1.1.24.0 or later. For example, if you're running version 1.1.24.0 of the SQL resource provider, then you can upgrade to version 1.1.33.0 or later without needing to first install version 1.1.30.0. To review available resource provider versions, and the version of Azure Stack Hub they're supported on, see the versions list in [Deploy the resource provider prerequisites](#).

To update the resource provider, use the *UpdateSQLProvider.ps1* script. Use your service account with local administrative rights and is an **owner** of the subscription. This script is included with the download of the new SQL resource provider. The update process is similar to the process used to [Deploy the resource provider](#). The update script uses the same arguments as the *DeploySqlProvider.ps1* script, and you'll need to provide certificate information.

## IMPORTANT

Before upgrading the resource provider, review the release notes to learn about new functionality, fixes, and any known issues that could affect your deployment.

## Update script processes

The *UpdateSQLProvider.ps1* script creates a new virtual machine (VM) with the latest resource provider code.

## NOTE

We recommend that you download the latest Windows Server 2016 Core image from Marketplace Management. If you need to install an update, you can place a **single** MSU package in the local dependency path. The script will fail if there's more than one MSU file in this location.

After the *UpdateSQLProvider.ps1* script creates a new VM, the script migrates the following settings from the old provider VM:

- database information
- hosting server information
- required DNS record

## Update script parameters

You can specify the following parameters from the command line when you run the **UpdateSQLProvider.ps1** PowerShell script. If you don't, or if any parameter validation fails, you're prompted to provide the required parameters.

| PARAMETER NAME                       | DESCRIPTION                                                                                                                                                                                                                                                                    | COMMENT OR DEFAULT VALUE                                      |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| <b>CloudAdminCredential</b>          | The credential for the cloud admin, necessary for accessing the privileged endpoint.                                                                                                                                                                                           | <i>Required</i>                                               |
| <b>AzCredential</b>                  | The credentials for the Azure Stack Hub service admin account. Use the same credentials that you used for deploying Azure Stack Hub.                                                                                                                                           | <i>Required</i>                                               |
| <b>VMLocalCredential</b>             | The credentials for the local admin account of the SQL resource provider VM.                                                                                                                                                                                                   | <i>Required</i>                                               |
| <b>PrivilegedEndpoint</b>            | The IP address or DNS name of the privileged endpoint.                                                                                                                                                                                                                         | <i>Required</i>                                               |
| <b>AzureEnvironment</b>              | The Azure environment of the service admin account which you used for deploying Azure Stack Hub. Required only for Azure AD deployments. Supported environment names are <b>AzureCloud</b> , <b>AzureUSGovernment</b> , or if using a China Azure AD, <b>AzureChinaCloud</b> . | AzureCloud                                                    |
| <b>DependencyFilesLocalPath</b>      | You must also put your certificate .pfx file in this directory.                                                                                                                                                                                                                | <i>Optional for single node, but mandatory for multi-node</i> |
| <b>DefaultSSLCertificatePassword</b> | The password for the .pfx certificate.                                                                                                                                                                                                                                         | <i>Required</i>                                               |
| <b>MaxRetryCount</b>                 | The number of times you want to retry each operation if there's a failure.                                                                                                                                                                                                     | 2                                                             |
| <b>RetryDuration</b>                 | The timeout interval between retries, in seconds.                                                                                                                                                                                                                              | 120                                                           |
| <b>Uninstall</b>                     | Removes the resource provider and all associated resources.                                                                                                                                                                                                                    | No                                                            |
| <b>DebugMode</b>                     | Prevents automatic cleanup on failure.                                                                                                                                                                                                                                         | No                                                            |

## Update script PowerShell example

**NOTE**

This update process only applies to Azure Stack Hub integrated systems.

If you're updating the SQL resource provider version to 1.1.33.0 or previous versions, you need to install specific versions of AzureRm.BootStrapper and Azure Stack Hub modules in PowerShell. If you're updating to the SQL resource provider version 1.1.47.0, the deployment script will automatically download and install the necessary PowerShell modules for you to path C:\Program Files\SqlMySqlPsh.

```
Install the AzureRM.Bootstrapper module, set the profile, and install the AzureStack module.
Note that this might not be the most currently available version of Azure Stack Hub PowerShell.
Install-Module -Name AzureRm.BootStrapper -Force
Use-AzureRmProfile -Profile 2018-03-01-hybrid -Force
Install-Module -Name AzureStack -RequiredVersion 1.6.0
```

#### NOTE

In disconnected scenario, you need to download the required PowerShell modules and register the repository manually as a prerequisite. You can get more information in [Deploy SQL resource provider](#)

The following is an example of using the *UpdateSQLProvider.ps1* script that you can run from an elevated PowerShell console. Be sure to change the variable information and passwords as needed:

```
Use the NetBIOS name for the Azure Stack Hub domain. On the Azure Stack Hub SDK, the default is AzureStack
but this might have been changed at installation.
$domain = "AzureStack"

For integrated systems, use the IP address of one of the ERCS VMs.
$privilegedEndpoint = "AzS-ERCS01"

Provide the Azure environment used for deploying Azure Stack Hub. Required only for Azure AD deployments.
Supported values for the <environment name> parameter are AzureCloud, AzureChinaCloud, or AzureUSGovernment
depending which Azure subscription you're using.
$AzureEnvironment = "<EnvironmentName>"

Point to the directory where the resource provider installation files were extracted.
$tempDir = 'C:\TEMP\SQLRP'

The service admin account (this can be Azure AD or AD FS).
$serviceAdmin = "admin@mydomain.onmicrosoft.com"
$AdminPass = ConvertTo-SecureString "P@ssw0rd1" -AsPlainText -Force
$AdminCreds = New-Object System.Management.Automation.PSCredential ($serviceAdmin, $AdminPass)

Set the credentials for the new resource provider VM.
$vmLocalAdminPass = ConvertTo-SecureString "P@ssw0rd1" -AsPlainText -Force
$vmLocalAdminCreds = New-Object System.Management.Automation.PSCredential ("sqlrpadmin", $vmLocalAdminPass)

Add the cloudadmin credential required for privileged endpoint access.
$CloudAdminPass = ConvertTo-SecureString "P@ssw0rd1" -AsPlainText -Force
$CloudAdminCreds = New-Object System.Management.Automation.PSCredential ("$domain\cloudadmin",
$CloudAdminPass)

Change the following as appropriate.
$PfxPass = ConvertTo-SecureString "P@ssw0rd1" -AsPlainText -Force

For version 1.1.47.0, the PowerShell modules used by the RP deployment are placed in C:\Program
Files\SqlMySqlPsh
The deployment script adds this path to the system $env:PSModulePath to ensure correct modules are used.
$rpModulePath = Join-Path -Path $env:ProgramFiles -ChildPath 'SqlMySqlPsh'
$env:PSModulePath = $env:PSModulePath + ";" + $rpModulePath

Change directory to the folder where you extracted the installation files.
Then adjust the endpoints.
. $tempDir\UpdateSQLProvider.ps1 -AzCredential $AdminCreds `
-VMLocalCredential $vmLocalAdminCreds `
-CloudAdminCredential $cloudAdminCreds `
-PrivilegedEndpoint $privilegedEndpoint `
-AzureEnvironment $AzureEnvironment `
-DefaultSSLCertificatePassword $PfxPass `
-DependencyFilesLocalPath $tempDir\cert
```

When the resource provider update script finishes, close the current PowerShell session.

## Next steps

[Maintain the SQL resource provider](#)

# SQL resource provider maintenance operations

6 minutes to read • [Edit Online](#)

The SQL resource provider runs on a locked down virtual machine (VM). To enable maintenance operations, you need to update the VM's security. To do this using the principle of Least Privilege, use [PowerShell Just Enough Administration \(JEA\)](#) endpoint `DBAdapterMaintenance`. The resource provider installation package includes a script for this action.

## Patching and updating

The SQL resource provider isn't serviced as part of Azure Stack Hub because it's an add-on component. Microsoft provides updates to the SQL resource provider as necessary. When an updated SQL adapter is released, a script is provided to apply the update. This script creates a new resource provider VM, migrating the state of the old provider VM to the new VM. For more information, see [Update the SQL resource provider](#).

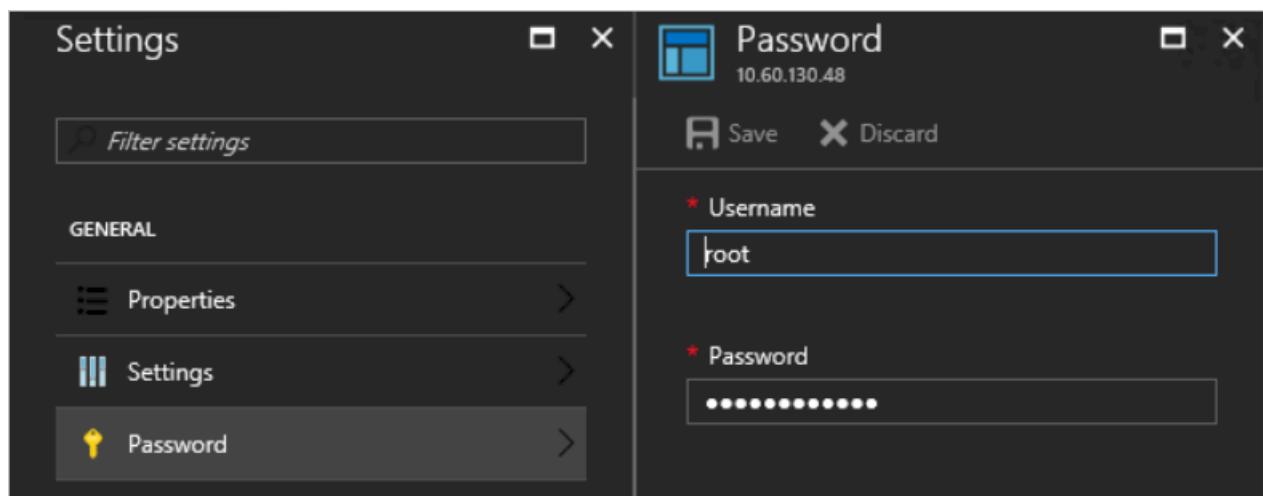
### Provider VM

Because the resource provider runs on a *user* VM, you need to apply the required patches and updates when they're released. Use the Windows update packages that are provided as part of the patch-and-update cycle to apply updates to the VM.

## Updating SQL credentials

You're responsible for creating and maintaining sysadmin accounts on your SQL servers. The resource provider needs an account with these privileges to manage databases for users, but it doesn't need access to the users' data. If you need to update the sysadmin passwords on your SQL servers, you can use the resource provider's administrator interface to change a stored password. These passwords are stored in a Key Vault on your Azure Stack Hub instance.

To modify the settings, select **Browse > ADMINISTRATIVE RESOURCES > SQL Hosting Servers > SQL Logins** and select a user name. The change must be made on the SQL instance first (and any replicas, if necessary.) Under **Settings**, select **Password**.



## Secrets rotation

*These instructions only apply to Azure Stack Hub Integrated Systems.*

When using the SQL and MySQL resource providers with Azure Stack Hub integrated systems, the Azure Stack

Hub operator is responsible for rotating the following resource provider infrastructure secrets to ensure that they don't expire:

- External SSL certificate [provided during deployment](#).
- The resource provider VM local admin account password provided during deployment.
- Resource provider diagnostic user (dbadAPTERdiag) password.

## PowerShell examples for rotating secrets

### Change all the secrets at the same time.

```
.\SecretRotationSQLProvider.ps1 `
 -Privilegedendpoint $Privilegedendpoint `
 -CloudAdminCredential $cloudCreds `
 -AzCredential $adminCreds `
 -DiagnosticsUserPassword $passwd `
 -DependencyFilesLocalPath $certPath `
 -DefaultSSLCertificatePassword $certPasswd `
 -VMLocalCredential $localCreds
```

### Change the diagnostic user password.

```
.\SecretRotationSQLProvider.ps1 `
 -Privilegedendpoint $Privilegedendpoint `
 -CloudAdminCredential $cloudCreds `
 -AzCredential $adminCreds `
 -DiagnosticsUserPassword $passwd
```

### Change the VM local admin account password.

```
.\SecretRotationSQLProvider.ps1 `
 -Privilegedendpoint $Privilegedendpoint `
 -CloudAdminCredential $cloudCreds `
 -AzCredential $adminCreds `
 -VMLocalCredential $localCreds
```

### Change the SSL certificate password.

```
.\SecretRotationSQLProvider.ps1 `
 -Privilegedendpoint $Privilegedendpoint `
 -CloudAdminCredential $cloudCreds `
 -AzCredential $adminCreds `
 -DependencyFilesLocalPath $certPath `
 -DefaultSSLCertificatePassword $certPasswd
```

## SecretRotationSQLProvider.ps1 parameters

| PARAMETER               | DESCRIPTION                                                   |
|-------------------------|---------------------------------------------------------------|
| AzCredential            | Azure Stack Hub service admin account credential.             |
| CloudAdminCredential    | Azure Stack Hub cloud admin domain account credential.        |
| PrivilegedEndpoint      | Privileged Endpoint to access Get-AzureStackStampInformation. |
| DiagnosticsUserPassword | Diagnostics user account password.                            |

| PARAMETER                     | DESCRIPTION                                 |
|-------------------------------|---------------------------------------------|
| VMLocalCredential             | Local admin account on the MySQLAdapter VM. |
| DefaultSSLCertificatePassword | Default SSL certificate (*.pfx) password.   |
| DependencyFilesLocalPath      | Dependency files local path.                |

## Known issues

### Issue:

Secrets rotation logs. The logs for secrets rotation aren't automatically collected if the secret rotation custom script fails when it's run.

### Workaround:

Use the Get-AzsDBAdapterLogs cmdlet to collect all resource provider logs, including AzureStack.DatabaseAdapter.SecretRotation.ps1\_\* .log, saved in C:\Logs.

## Update the VM operating system

Use one of the following methods to update the VM operating system.

- Install the latest resource provider package using a currently patched Windows Server 2016 Core image.
- Install a Windows Update package during the installation of, or update to, the resource provider.

## Update the VM Windows Defender definitions

To update the Windows Defender definitions:

1. Download the Windows Defender definitions update from [Security intelligence updates for Windows Defender](#).

On the definitions update page, scroll down to "Manually download the update". Download the "Windows Defender Antivirus for Windows 10 and Windows 8.1" 64-bit file.

You can also use [this direct link](#) to download/run the fpam-fe.exe file.

2. Create a PowerShell session to the SQL resource provider adapter VM's maintenance endpoint.
3. Copy the definitions update file to the VM using the maintenance endpoint session.
4. On the maintenance PowerShell session, run the *Update-DBAdapterWindowsDefenderDefinitions* command.
5. After you install the definitions, we recommend you delete the definitions update file by using the *Remove-ItemOnUserDrive* command.

### PowerShell script example for updating definitions

You can edit and run the following script to update the Defender definitions. Replace values in the script with values from your environment.

```

Set credentials for local admin on the resource provider VM.
$vmLocalAdminPass = ConvertTo-SecureString "<local admin user password>" -AsPlainText -Force
$vmLocalAdminUser = "<local admin user name>"
$vmLocalAdminCreds = New-Object System.Management.Automation.PSCredential `
 ($vmLocalAdminUser, $vmLocalAdminPass)

Provide the public IP address for the adapter VM.
$databaseRPMachine = "<RP VM IP address>"
$localPathToDefenderUpdate = "C:\DefenderUpdates\mpam-fe.exe"

Download the Windows Defender update definitions file from https://www.microsoft.com/wdsi/definitions.
Invoke-WebRequest -Uri 'https://go.microsoft.com/fwlink/?LinkID=121721&arch=x64' `
 -Outfile $localPathToDefenderUpdate

Create a session to the maintenance endpoint.
$session = New-PSSession -ComputerName $databaseRPMachine `
 -Credential $vmLocalAdminCreds -ConfigurationName DBAdapterMaintenance
Copy the defender update file to the adapter VM.
Copy-Item -ToSession $session -Path $localPathToDefenderUpdate `
 -Destination "User:\"
Install the update definitions.
Invoke-Command -Session $session -ScriptBlock `
 {Update-AzSDBAdapterWindowsDefenderDefinition -DefinitionsUpdatePackageFile "User:\mpam-fe.exe"}
Cleanup the definitions package file and session.
Invoke-Command -Session $session -ScriptBlock `
 {Remove-AzSItemOnUserDrive -ItemPath "User:\mpam-fe.exe"}
$session | Remove-PSSession

```

## Collect diagnostic logs

To collect logs from the locked down VM, use the PowerShell Just Enough Administration (JEA) endpoint *DBAdapterDiagnostics*. This endpoint provides the following commands:

- **Get-AzsDBAdapterLog.** This command creates a zip package of the resource provider diagnostics logs and saves the file on the session's user drive. You can run this command without any parameters and the last four hours of logs are collected.
- **Remove-AzsDBAdapterLog.** This command removes existing log packages on the resource provider VM.

### Endpoint requirements and process

When a resource provider is installed or updated, the **dbadapterdiag** user account is created. You'll use this account to collect diagnostic logs.

#### NOTE

The dbadapterdiag account password is the same as the password used for the local admin on the VM that's created during a provider deployment or update.

To use the *DBAdapterDiagnostics* commands, create a remote PowerShell session to the resource provider VM and run the **Get-AzsDBAdapterLog** command.

You set the time span for log collection by using the **FromDate** and **ToDate** parameters. If you don't specify one or both of these parameters, the following defaults are used:

- FromDate is four hours before the current time.
- ToDate is the current time.

### PowerShell script example for collecting logs

The following script shows how to collect diagnostic logs from the resource provider VM.

```

Create a new diagnostics endpoint session.
$databaseRPMachineIP = '<RP VM IP address>'
$diagnosticsUserName = 'dbadapterdiag'
$diagnosticsUserPassword = '<Enter Diagnostic password>'

$diagCreds = New-Object System.Management.Automation.PSCredential `
 ($diagnosticsUserName, (ConvertTo-SecureString -String $diagnosticsUserPassword -AsPlainText -Force))
$session = New-PSSession -ComputerName $databaseRPMachineIP -Credential $diagCreds `
 -ConfigurationName DBAdapterDiagnostics

Sample that captures logs from the previous hour.
$fromDate = (Get-Date).AddHours(-1)
$dateNow = Get-Date
$sb = {param($d1,$d2) Get-AzSDBAdapterLog -FromDate $d1 -ToDate $d2}
$logs = Invoke-Command -Session $session -ScriptBlock $sb -ArgumentList $fromDate,$dateNow

Copy the logs to the user drive.
$sourcePath = "User:\{0}" -f $logs
$destinationPackage = Join-Path -Path (Convert-Path '.') -ChildPath $logs
Copy-Item -FromSession $session -Path $sourcePath -Destination $destinationPackage

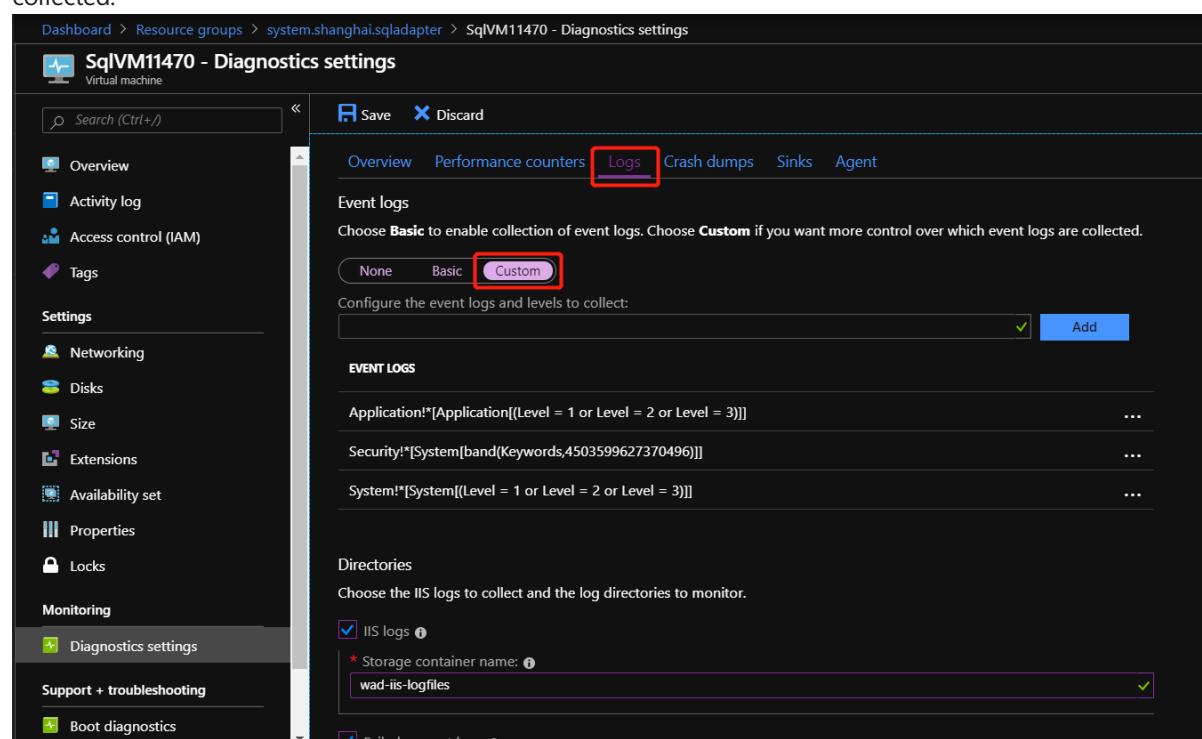
Clean up the logs.
$cleanup = Invoke-Command -Session $session -ScriptBlock {Remove-AzsDBAdapterLog}
Close the session.
$session | Remove-PSSession

```

## Configure Azure Diagnostics extension for SQL resource provider

Azure Diagnostics extension is installed on the SQL resource provider adapter VM by default. The following steps show how to customize the extension for gathering the SQL resource provider operational event logs and IIS logs for troubleshooting and auditing purpose.

1. Sign in to the Azure Stack Hub administrator portal.
2. Select **Virtual machines** from the pane on the left, search for the SQL resource provider adapter VM and select the VM.
3. In **Diagnostics settings** of the VM, go to the **Logs** tab and choose **Custom** to customize event logs being collected.



4. Add **Microsoft-AzureStack-DatabaseAdapter/Operational!\*** to collect SQL resource provider operational event logs.

The screenshot shows the 'Logs' tab of the 'Diagnostics settings' for a virtual machine named 'SqlVM11470'. In the 'Event logs' section, there is a search bar containing 'Microsoft-AzureStack-DatabaseAdapter/Operational!\*'. Below the search bar is a red box around the 'Add' button. The 'EVENT LOGS' section lists three log types: Application!, Security!, and System!. The 'Directories' section shows IIS logs are selected, with a storage container name 'wad-iis-logs' specified. Failed request logs are also selected.

5. To enable the collection of IIS logs, check **IIS logs** and **Failed request logs**.

The screenshot shows the 'Directories' section of the 'Diagnostics settings' for a virtual machine. Two checkboxes are checked: 'IIS logs' and 'Failed request logs'. Both checkboxes have a red box around them. The 'Storage container name' field for both is set to 'wad-iis-logs'. Other sections like 'Application logs' and 'Event tracing for Windows (ETW) events' are shown below.

6. Finally select **Save** to save all the Diagnostics settings.

Once the event logs and IIS logs collection are configured for SQL resource provider, the logs can be found in a system storage account named **sqladapterdiagaccount**.

To learn more about Azure Diagnostics extension, please see [What is Azure Diagnostics extension](#).

## Next steps

[Add SQL Server hosting servers](#)

# Remove the SQL resource provider

2 minutes to read • [Edit Online](#)

Before you remove the SQL resource provider, you must remove all the provider dependencies. You'll also need a copy of the deployment package that was used to install the resource provider.

## NOTE

You can find the download links for the resource provider installers in [Deploy the resource provider prerequisites](#).

Removing the SQL resource provider will delete the associated plans and quotas managed by operator. But it doesn't delete tenant databases from hosting servers.

## To remove the SQL resource provider

1. Verify that you've removed all the existing SQL resource provider dependencies.

## NOTE

Uninstalling the SQL resource provider will proceed even if dependent resources are currently using the resource provider.

2. Get a copy of the SQL resource provider installation package and then run the self-extractor to extract the contents to a temporary directory.
3. Open a new elevated PowerShell console window and change to the directory where you extracted the SQL resource provider installation files.
4. Run the DeploySqlProvider.ps1 script using the following parameters:
  - **Uninstall**: Removes the resource provider and all associated resources.
  - **PrivilegedEndpoint**: The IP address or DNS name of the privileged endpoint.
  - **AzureEnvironment**: The Azure environment used for deploying Azure Stack Hub. Required only for Azure AD deployments.
  - **CloudAdminCredential**: The credential for the cloud admin, necessary to access the privileged endpoint.
  - **AzCredential**: The credential for the Azure Stack Hub service admin account. Use the same credentials that you used for deploying Azure Stack Hub.

## Next steps

[Offer App Services as PaaS](#)

# SQL resource provider 1.1.47.0 release notes

2 minutes to read • [Edit Online](#)

These release notes describe the improvements and known issues in SQL resource provider version 1.1.47.0.

## Build reference

Download the SQL resource provider binary and then run the self-extractor to extract the contents to a temporary directory. The resource provider has a minimum corresponding Azure Stack Hub build. The minimum Azure Stack Hub release version required to install this version of the SQL resource provider is listed below:

| MINIMUM AZURE STACK HUB VERSION | SQL RESOURCE PROVIDER VERSION           |
|---------------------------------|-----------------------------------------|
| Version 1910 (1.1910.0.58)      | <a href="#">SQL RP version 1.1.47.0</a> |

### IMPORTANT

Apply the minimum supported Azure Stack Hub update to your Azure Stack Hub integrated system before deploying the latest version of the SQL resource provider.

## New features and fixes

This version of the Azure Stack Hub SQL resource provider is a hotfix release to make the resource provider compatible with some of the latest portal changes in the 1910 update without any new feature.

It also supports the current latest Azure Stack Hub API version profile 2019-03-01-hybrid and Azure Stack Hub PowerShell module 1.8.0. So during deployment and update, no specific history versions of modules need to be installed.

Please follow the resource provider update process to apply the SQL resource provider hotfix 1.1.47.0 after Azure Stack Hub is upgraded to the 1910 update. It will help address a known issue in the administrator portal where Capacity Monitoring in SQL resource provider keeps loading.

## Known issues

None.

## Next steps

[Learn more about the SQL resource provider.](#)

[Prepare to deploy the SQL resource provider.](#)

[Upgrade the SQL resource provider from a previous version.](#)

# SQL resource provider 1.1.33.0 release notes

2 minutes to read • [Edit Online](#)

These release notes describe the improvements and known issues in SQL resource provider version 1.1.33.0.

## Build reference

Download the SQL resource provider binary and then run the self-extractor to extract the contents to a temporary directory. The resource provider has a minimum corresponding Azure Stack Hub build. The minimum Azure Stack Hub release version required to install this version of the SQL resource provider is listed below:

| MINIMUM AZURE STACK HUB VERSION | SQL RESOURCE PROVIDER VERSION           |
|---------------------------------|-----------------------------------------|
| Version 1808 (1.1808.0.97)      | <a href="#">SQL RP version 1.1.33.0</a> |

### IMPORTANT

Apply the minimum supported Azure Stack Hub update to your Azure Stack Hub integrated system before deploying the latest version of the SQL resource provider.

## New features and fixes

This version of the Azure Stack Hub SQL resource provider includes the following improvements and fixes:

### Fixes

- **SQL resource provider portal extension might choose the wrong subscription.** The SQL resource provider uses Azure Resource Manager calls to determine the first service admin subscription to use, which might not be the *Default Provider Subscription*. If that happens, the SQL resource provider doesn't work normally.
- **SQL hosting server doesn't list hosted databases.** User-created databases might not be listed when viewing tenant resources for SQL hosting servers.
- **Previous SQL resource provider (1.1.30.0) deployment could fail if TLS 1.2 isn't enabled.** Updated the SQL resource provider 1.1.33.0 to enable TLS 1.2 when deploying the resource provider, updating the resource provider, or rotating secrets.
- **SQL resource provider secret rotation fails.** Fixed issue resulting in the following error code when rotating secrets:

```
New-AzureRmResourceGroupDeployment - Error: Code=InvalidDeploymentParameterValue; Message=The value of deployment parameter 'StorageAccountBlobUri' is null.
```

## Known issues

- **SQL SKUs can take up to an hour to be visible in the portal.** It can take up to an hour for newly created SKUs to be visible for use when creating new SQL databases.

**Workaround:** None.

- **Reused SQL logins.** Attempting to create a new SQL login with the same username as an existing login under the same subscription will result in reusing the same login and the existing password.  
**Workaround:** Use different usernames when creating new logins under the same subscription or create logins with the same username under different subscriptions.
- **Shared SQL logins cause data inconsistency.** If a SQL login is shared for multiple SQL databases under the same subscription, changing the login password will cause data inconsistency.  
**Workaround:** Always use different logins for different databases under the same subscription.
- **SQL resource provider fails to add SQL Server Always On listener.** When using the listener IP address of the SQL Server Always On Listener, the SQL resource provider VM can't resolve the listener's host name.  
**Workaround:** Ensure that DNS works correctly to resolve the listener IP to listener host name.

#### Known issues for Cloud Admins operating Azure Stack Hub

Refer to the documentation in the [Azure Stack Hub release notes](#).

## Next steps

[Learn more about the SQL resource provider.](#)

[Prepare to deploy the SQL resource provider.](#)

[Upgrade the SQL resource provider from a previous version.](#)

# SQL resource provider 1.1.30.0 release notes

2 minutes to read • [Edit Online](#)

These release notes describe the improvements and known issues in SQL resource provider version 1.1.30.0.

## Build reference

Download the SQL resource provider binary and then run the self-extractor to extract the contents to a temporary directory. The resource provider has a minimum corresponding Azure Stack Hub build. The minimum Azure Stack Hub release version required to install this version of the SQL resource provider is listed below:

| MINIMUM AZURE STACK HUB VERSION | SQL RESOURCE PROVIDER VERSION |
|---------------------------------|-------------------------------|
| Version 1808 (1.1808.0.97)      | 1.1.30.0                      |

### IMPORTANT

Apply the minimum supported Azure Stack Hub update to your Azure Stack Hub integrated system before deploying the latest version of the SQL resource provider.

## New features and fixes

This version of the Azure Stack Hub SQL resource provider includes the following improvements and fixes:

- **Telemetry enabled for SQL resource provider deployments.** Telemetry collection has been enabled for SQL resource provider deployments. Telemetry collected includes resource provider deployment, start and stop times, exit status, exit messages, and error details (if applicable).
- **TLS 1.2 encryption update.** Enabled TLS 1.2-only support for resource provider communication with internal Azure Stack Hub components.

### Fixes

- **SQL resource provider Azure Stack Hub PowerShell compatibility.** The SQL resource provider has been updated to work with the Azure Stack Hub 2018-03-01-hybrid PowerShell profile and to provide compatibility with AzureRM 1.3.0 and later.
- **SQL login change password blade.** Fixed an issue where the password can't be changed on the change password blade. Removed links from password change notifications.
- **SQL hosting server settings blade update.** Fixed an issue where the settings blade was incorrectly titled as "Password".

## Known issues

- **SQL SKUs can take up to an hour to be visible in the portal.** It can take up to an hour for newly created SKUs to be visible for use when creating new SQL databases.

**Workaround:** None.

- **Reused SQL logins.** Attempting to create a new SQL login with the same username as an existing login under the same subscription will result in reusing the same login and the existing password.  
**Workaround:** Use different usernames when creating new logins under the same subscription or create logins with the same username under different subscriptions.
- **Shared SQL logins cause data inconsistency.** If a SQL login is shared for multiple SQL databases under the same subscription, changing the login password will cause data inconsistency.  
**Workaround:** Always use different logins for different databases under the same subscription.
- **TLS 1.2 support requirement.** If you try to deploy or update the SQL resource provider from a computer where TLS 1.2 isn't enabled, the operation might fail. Run the following PowerShell command on the computer being used to deploy or update the resource provider to verify that TLS 1.2 is returned as supported:

```
[System.Net.ServicePointManager]::SecurityProtocol
```

If **Tls12** isn't included in the output of the command, TLS 1.2 isn't enabled on the computer.

**Workaround:** Run the following PowerShell command to enable TLS 1.2 and then start the resource provider deployment or update script from the same PowerShell session:

```
[System.Net.ServicePointManager]::SecurityProtocol = [System.Net.SecurityProtocolType]::Tls12
```

- **SQL resource provider fails to add SQL Server Always On listener.** When using the listener IP address of the SQL Server Always On Listener, the SQL resource provider VM can't resolve the listener's host name.

**Workaround:** Ensure that DNS works correctly to resolve the listener IP to listener host name.

## Known issues for Cloud Admins operating Azure Stack Hub

Refer to the documentation in the [Azure Stack Hub release notes](#).

## Next steps

[Learn more about the SQL resource provider.](#)

[Prepare to deploy the SQL resource provider.](#)

[Upgrade the SQL resource provider from a previous version.](#)

# Usage and billing in Azure Stack Hub

2 minutes to read • [Edit Online](#)

This article describes how Azure Stack Hub users are billed for resource usage, and how the billing information is accessed for analytics and chargeback.

Azure Stack Hub collects and groups usage data for resources that are used, then forwards this data to Azure Commerce. Azure Commerce bills you for Azure Stack Hub usage in the same way it bills you for Azure usage.

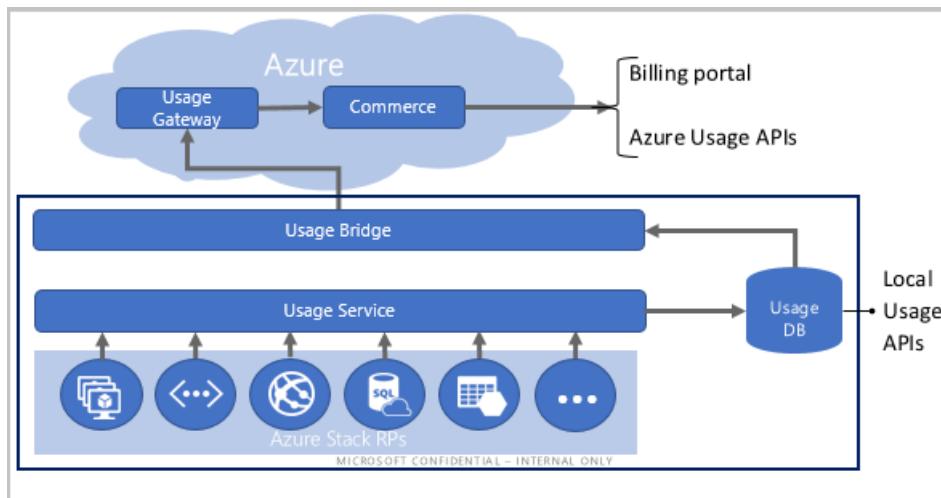
You can also get usage data and export it to your own billing or chargeback system by using a billing adapter, or export it to a business intelligence tool such as Microsoft Power BI.

## Usage pipeline

Each resource provider in Azure Stack Hub posts usage data per resource usage. The usage service periodically (hourly and daily) aggregates usage data and stores it in the usage database. Azure Stack Hub operators and users can access the stored usage data through the Azure Stack Hub resource usage APIs.

If you've [registered your Azure Stack Hub instance with Azure](#), Azure Stack Hub is configured to send the usage data to Azure Commerce. After the data is uploaded to Azure, you can access it through the billing portal or by using Azure resource usage APIs. For more information about what usage data is reported to Azure, see [Usage data reporting](#).

The following image shows the key components in the usage pipeline:



## What usage information can I find, and how?

Azure Stack Hub resource providers (such as Compute, Storage, and Network) generate usage data at hourly intervals for each subscription. The usage data contains information about the resource used, such as resource name, subscription used, and quantity used. To learn about the meters' ID resources, see the [Usage API FAQ](#).

After the usage data has been collected, it is [reported to Azure](#) to generate a bill, which can be viewed through the Azure billing portal.

**NOTE**

Usage data reporting is not required for the Azure Stack Development Kit (ASDK) and for Azure Stack Hub integrated system users who license under the capacity model. To learn more about licensing in Azure Stack Hub, see the [packaging and pricing data sheet](#).

The Azure billing portal shows usage data for the chargeable resources. In addition to the chargeable resources, Azure Stack Hub captures usage data for a broader set of resources, which you can access in your Azure Stack Hub environment through REST APIs or PowerShell cmdlets. Azure Stack Hub operators can get the usage data for all user subscriptions. Individual users can only get their own usage details.

## Usage reporting for multi-tenant Cloud Solution Providers

A multi-tenant Cloud Solution Provider (CSP) using Azure Stack Hub might want to report each customer usage separately, so that the provider can charge usage to different Azure subscriptions.

Each customer has their identity represented by a different Azure Active Directory (Azure AD) tenant. Azure Stack Hub supports assigning one CSP subscription to each Azure AD tenant. You can add tenants and their subscriptions to the base Azure Stack Hub registration. The base registration is done for all Azure Stack Hub instances. If a subscription is not registered for a tenant, the user can still use Azure Stack Hub, and their usage is sent to the subscription used for the base registration.

## Next steps

- [Register with Azure Stack Hub](#)
- [Report Azure Stack Hub usage data to Azure](#)
- [Provider Resource Usage API](#)
- [Tenant Resource Usage API](#)
- [Usage-related FAQ](#)

# Manage usage and billing for Azure Stack Hub as a Cloud Solution Provider

3 minutes to read • [Edit Online](#)

This article describes how to register Azure Stack Hub as a Cloud Solution Provider (CSP) and how to add customers.

As a CSP, you work with diverse customers using your Azure Stack Hub. Each customer has a CSP subscription in Azure. You must direct usage from your Azure Stack Hub to each user subscription.

The following figure shows the required steps to choose your shared services account, and to register the Azure account with the Azure Stack Hub account. Once registered, you can onboard your end customers:



## Create a CSP or APSS subscription

### CSP subscription types

Choose the type of shared services account that you use for Azure Stack Hub. The types of subscriptions that can be used for registration of a multi-tenant Azure Stack Hub are:

- Cloud Solution Provider
- Partner Shared Services subscription

### Azure Partner Shared Services

Azure Partner Shared Services (APSS) subscriptions are the preferred choice for registration when a direct CSP or a CSP distributor operates Azure Stack Hub.

APSS subscriptions are associated with a shared-services tenant. When you register Azure Stack Hub, you provide credentials for an account that's an owner of the subscription. The account you use to register Azure Stack Hub can be different from the admin account that you use for deployment. Furthermore, the two accounts do not need to belong to the same domain; you can deploy using the tenant that you already use. For example, you can use `ContosoCSP.onmicrosoft.com`, then register using a different tenant; for example, `IURContosoCSP.onmicrosoft.com`.

You must remember to sign in using `ContosoCSP.onmicrosoft.com` when you perform daily Azure Stack Hub administration. You sign in to Azure using `IURContosoCSP.onmicrosoft.com` when you need to perform registration operations.

For a description of APSS subscriptions and how to create them, see [Add Azure Partner Shared Services](#).

### CSP subscriptions

CSP subscriptions are the preferred choice for registration when a CSP reseller or an end customer operates Azure Stack Hub.

## Register Azure Stack Hub

Use the APSS subscription created using the information in the preceding section to register Azure Stack Hub with Azure. For more information, see [Register Azure Stack Hub with your Azure Subscription](#).

## Add end customer

To configure Azure Stack Hub so that a new tenant's resource usage is reported to their CSP subscription, see [Add tenant for usage and billing to Azure Stack Hub](#).

## Charge the right subscriptions

Azure Stack Hub uses a feature called *registration*. A registration is an object stored in Azure. The registration object documents which Azure subscription(s) to use to charge for a given Azure Stack Hub. This section addresses the importance of registration.

Using registration, Azure Stack Hub can:

- Forward [Azure Stack Hub usage data](#) to Azure Commerce and bill an Azure subscription.
- Report each customer's usage on a different subscription with a multi-tenant Azure Stack Hub deployment. Multi-tenancy enables Azure Stack Hub to support different organizations on the same Azure Stack Hub instance.

For each Azure Stack Hub, there is one default subscription and many tenant subscriptions. The default subscription is an Azure subscription that is charged if there's no tenant-specific subscription. It must be the first subscription to be registered. For multi-tenant usage reporting to work, the subscription must be a CSP or APSS subscription.

Then, the registration is updated with an Azure subscription for each tenant that uses Azure Stack Hub. Tenant subscriptions must be of the CSP type, and must roll up to the partner who owns the default subscription. You cannot register someone else's customers.

When Azure Stack Hub forwards usage info to global Azure, a service in Azure consults the registration and maps each tenant's usage to the appropriate tenant subscription. If a tenant has not been registered, that usage goes to the default subscription for the Azure Stack Hub instance from which it originated.

Because tenant subscriptions are CSP subscriptions, their bill is sent to the CSP partner, and usage info is not visible to the end customer.

## Next steps

- To learn more about the CSP program, see [Cloud Solution Provider program](#).
- To learn more about how to retrieve resource usage info from Azure Stack Hub, see [Usage and billing in Azure Stack Hub](#).

# Add tenant for usage and billing to Azure Stack Hub

4 minutes to read • [Edit Online](#)

This article shows you how to add a tenant to an Azure Stack Hub deployment managed by a Cloud Solution Provider (CSP). When the new tenant uses resources, Azure Stack Hub reports usage to their CSP subscription.

CSPs often offer services to multiple end customers (tenants) on their Azure Stack Hub deployment. Adding tenants to the Azure Stack Hub registration ensures that each tenant's usage is reported and billed to the corresponding CSP subscription. If you don't complete the steps in this article, tenant usage is charged to the subscription used in the initial registration of Azure Stack Hub. Before you can add an end customer to Azure Stack Hub for usage tracking and to manage their tenant, you must configure Azure Stack Hub as a CSP. For steps and resources, see [Manage usage and billing for Azure Stack Hub as a Cloud Solution Provider](#).

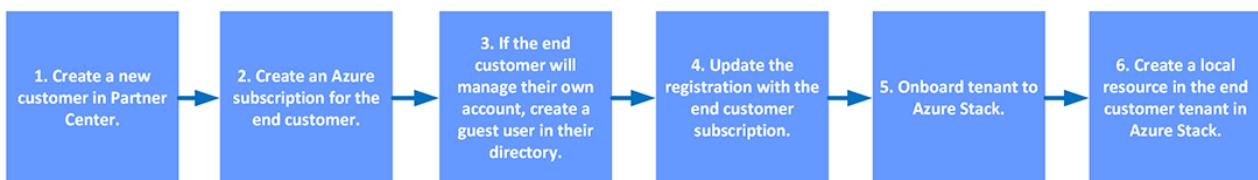
The following figure shows the steps that a CSP needs to follow to enable a new end customer to use Azure Stack Hub, and to set up usage tracking for the customer. By adding the end customer, you're also able to manage resources in Azure Stack Hub. You have two options for managing their resources:

- You can maintain the end customer and provide credentials for the local Azure Stack Hub subscription to the end customer.
- The end customer can work with their subscription locally and add the CSP as a guest with owner permissions.

## Add an end customer

Before you add an end customer, you must enable multi-tenant billing on your registration. In order to enable multi-tenant billing, send the registration subscription ID, resource group name, and registration name to [azstcsp@microsoft.com](mailto:azstcsp@microsoft.com). It usually takes 1-2 business days to enable multi-tenancy.

Perform the following steps to add an end customer, as pictured in the following figure:



### Create a new customer in Partner Center

In Partner Center, create a new Azure subscription for the customer. For instructions, see [Add a new customer](#).

### Create an Azure subscription for the end customer

After you've created a record of your customer in Partner Center, you can sell them subscriptions to products in the catalog. For instructions, see [Create, suspend, or cancel customer subscriptions](#).

### Create a guest user in the end customer directory

By default, you, as the CSP, do not have access to the end customer's Azure Stack Hub subscription. However, if your customer wants you to manage their resources, they can then add your account as owner/contributor to their Azure Stack Hub subscription. In order to do that, they must add your account as guest user to their Azure AD tenant. It's advised that you use a different account from your Azure CSP account to manage your customer's Azure Stack Hub subscription to ensure you don't lose access to your customer's Azure subscription.

### Update the registration with the end customer subscription

Update your registration with the new customer subscription. Azure reports the customer usage using the customer identity from Partner Center. This step ensures that each customer's usage is reported under that

customer's individual CSP subscription. This makes tracking usage and billing easier. To perform this step, you must first [register Azure Stack Hub](#).

1. Open Windows PowerShell with an elevated prompt, and run:

```
Add-AzureRmAccount
```

#### NOTE

If your session expires, your password has changed, or you simply wish to switch accounts, run the following cmdlet before you sign in using Add-AzureRmAccount: `Remove-AzureRmAccount -Scope Process`

2. Type your Azure credentials.

3. In the PowerShell session, run:

```
New-AzureRmResource -ResourceId
"subscriptions/{registrationSubscriptionId}/resourceGroups/{resourceGroup}/providers/Microsoft.AzureStack/registrations/{registrationName}/customerSubscriptions/{customerSubscriptionId}" -ApiVersion 2017-06-01
```

### New-AzureRmResource PowerShell parameters

The following section describes the parameters for the **New-AzureRmResource** cmdlet:

| PARAMETER                  | DESCRIPTION                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| registrationSubscriptionID | The Azure subscription that was used for the initial registration of the Azure Stack Hub.                                                                                                                                                                                                                                                                                                         |
| customerSubscriptionID     | The Azure subscription (not Azure Stack Hub) belonging to the customer to be registered. Must be created in the CSP offer. In practice, this means through Partner Center. If a customer has more than one Azure Active Directory tenant, this subscription must be created in the tenant that will be used to log into Azure Stack Hub. The customer subscription ID must use lowercase letters. |
| resourceGroup              | The resource group in Azure in which your registration is stored.                                                                                                                                                                                                                                                                                                                                 |
| registrationName           | The name of the registration of your Azure Stack Hub. It's an object stored in Azure.                                                                                                                                                                                                                                                                                                             |

#### NOTE

Tenants must be registered with each Azure Stack Hub they use. If you have two Azure Stack Hub deployments, and a tenant uses both of them, you must update the initial registrations of each deployment with the tenant subscription.

### Onboard tenant to Azure Stack Hub

Configure Azure Stack Hub to support users from multiple Azure AD tenants to use services in Azure Stack Hub. For instructions, see [Enable multi-tenancy in Azure Stack Hub](#).

### Create a local resource in the end customer tenant in Azure Stack Hub

Once you've added the new customer to Azure Stack Hub, or the end customer tenant has enabled your guest

account with owner privileges, verify that you can create a resource in their tenant. For example, they can [Create a Windows virtual machine with the Azure Stack Hub portal](#).

## Next steps

- To review error messages if they're triggered in your registration process, see [Tenant registration error messages](#).
- To learn more about how to retrieve resource usage information from Azure Stack Hub, see [Usage and billing in Azure Stack Hub](#).
- To review how an end customer may add you, the CSP, as the manager for their Azure Stack Hub tenant, see [Enable a Cloud Solution Provider to manage your Azure Stack Hub subscription](#).

# Register tenants for usage tracking in Azure Stack Hub

3 minutes to read • [Edit Online](#)

This article contains details about registration operations. You can use these operations to:

- Manage tenant registrations
- Manage tenant usage tracking

## Add tenant to registration

You can use this operation when you want to add a new tenant to your registration. Tenant usage is reported under an Azure subscription connected with the Azure Active Directory (Azure AD) tenant.

You can also use this operation to change the subscription associated with a tenant. Call PUT or the **New-AzureRMResource** PowerShell cmdlet to overwrite the previous mapping.

You can associate a single Azure subscription with a tenant. If you try to add a second subscription to an existing tenant, the first subscription is overwritten.

### Use API profiles

The following registration cmdlets require that you specify an API profile when running PowerShell. API profiles represent a set of Azure resource providers and their API versions. They help you use the right version of the API when interacting with multiple Azure clouds. For example, if you work with multiple clouds when working with global Azure and Azure Stack Hub, API profiles specify a name that matches their release date. You use the **2017-09-03** profile.

For more information about Azure Stack Hub and API profiles, see [Manage API version profiles in Azure Stack Hub](#).

### Parameters

| PARAMETER                  | DESCRIPTION                                                                                                                                                                                                                                                                                 |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| registrationSubscriptionID | The Azure subscription that was used for the initial registration.                                                                                                                                                                                                                          |
| customerSubscriptionID     | The Azure subscription (not Azure Stack Hub) belonging to the customer to be registered. Must be created in the Cloud Solution Provider (CSP) offer through the Partner Center. If a customer has more than one tenant, create a subscription for the tenant to sign in to Azure Stack Hub. |
| resourceGroup              | The resource group in Azure in which your registration is stored.                                                                                                                                                                                                                           |
| registrationName           | The name of the registration of your Azure Stack Hub. It's an object stored in Azure. The name is usually in the form <b>azurestack-CloudID</b> , where <b>CloudID</b> is the cloud ID of your Azure Stack Hub deployment.                                                                  |

### NOTE

Tenants must be registered with each Azure Stack Hub deployment that they use. If a tenant uses more than one Azure Stack Hub, update the initial registrations of each deployment with the tenant subscription.

### PowerShell

Use the **New-AzureRmResource** cmdlet to add a tenant. [Connect to Azure Stack Hub](#), and then from an elevated prompt use the following cmdlet:

```
New-AzureRmResource -ResourceId
"subscriptions/{registrationSubscriptionId}/resourceGroups/{resourceGroup}/providers/Microsoft.AzureStack/registrations/{registrationName}/customerSubscriptions/{customerSubscriptionId}" -ApiVersion 2017-06-01
```

### API call

**Operation:** PUT

**RequestURI:**

```
subscriptions/{registrationSubscriptionId}/resourceGroups/{resourceGroup}/providers/Microsoft.AzureStack/registrations/{registrationName}/customerSubscriptions/{customerSubscriptionId}
api-version=2017-06-01 HTTP/1.1
```

**Response:** 201 Created

**Response Body:** Empty

## List all registered tenants

Get a list of all tenants that have been added to a registration.

### NOTE

If no tenants have been registered, you won't receive a response.

### Parameters

| PARAMETER                  | DESCRIPTION                                                        |
|----------------------------|--------------------------------------------------------------------|
| registrationSubscriptionId | The Azure subscription that was used for the initial registration. |
| resourceGroup              | The resource group in Azure in which your registration is stored.  |

| PARAMETER        | DESCRIPTION                                                                                                                                                                                                                              |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| registrationName | The name of the registration of your Azure Stack Hub deployment. It's an object stored in Azure. The name is usually in the form of <b>azurystack-CloudID</b> , where <b>CloudID</b> is the cloud ID of your Azure Stack Hub deployment. |

## PowerShell

Use the **Get-AzureRmResource** cmdlet to list all registered tenants. [Connect to Azure Stack Hub](#), and then from an elevated prompt run the following cmdlet:

```
Get-AzureRmResource -ResourceId
"subscriptions/{registrationSubscriptionId}/resourceGroups/{resourceGroup}/providers/Microsoft.AzureStack/registrations/{registrationName}/customerSubscriptions" -
ApiVersion 2017-06-01
```

## API call

You can get a list of all tenant mappings using the GET operation.

### Operation: GET

#### RequestURI:

```
subscriptions/{registrationSubscriptionId}/resourceGroups/{resourceGroup}/providers/Microsoft.AzureStack/registrations/{registrationName}/customerSubscriptions?api-version=2017-06-01 HTTP/1.1
```

#### Response: 200

#### Response Body:

```
{
 "value": [
 {
 "id": "subscriptions/{subscriptionId}/resourceGroups/{resourceGroup}/providers/Microsoft.AzureStack/registrations/{registrationName}/customerSubscriptions/{cspSubscriptionId} 1",
 "name": "cspSubscriptionId 1",
 "type": "Microsoft.AzureStack/customerSubscriptions",
 "properties": { "tenantId": "tId1" }
 },
 {
 "id": "subscriptions/{subscriptionId}/resourceGroups/{resourceGroup}/providers/Microsoft.AzureStack/registrations/{registrationName}/customerSubscriptions/{cspSubscriptionId} 2",
 "name": "cspSubscriptionId2",
 "type": "Microsoft.AzureStack/customerSubscriptions",
 "properties": { "tenantId": "tId2" }
 }
],
 "nextLink": "{originalRequestUrl}?$skipToken={opaqueString}"
}
```

## Remove a tenant mapping

You can remove a tenant that has been added to a registration. If that tenant is still using resources on Azure Stack Hub, their usage is charged to the subscription used in the initial Azure Stack Hub registration.

### Parameters

| PARAMETER                  | DESCRIPTION                              |
|----------------------------|------------------------------------------|
| registrationSubscriptionId | Subscription ID for the registration.    |
| resourceGroup              | The resource group for the registration. |
| registrationName           | The name of the registration.            |
| customerSubscriptionId     | The customer subscription ID.            |

## PowerShell

Use the **Remove-AzureRmResource** cmdlet to remove a tenant. [Connect to Azure Stack Hub](#), and then from an elevated prompt run the following cmdlet:

```
Remove-AzureRmResource -ResourceId
"subscriptions/{registrationSubscriptionId}/resourceGroups/{resourceGroup}/providers/Microsoft.AzureStack/registrations/{registrationName}/customerSubscriptions/{customerSubscriptionId}" -ApiVersion 2017-06-01
```

## API call

You can remove tenant mappings using the DELETE operation.

### Operation: DELETE

#### RequestURI:

```
subscriptions/{registrationSubscriptionId}/resourceGroups/{resourceGroup}/providers/Microsoft.AzureStack/registrations/{registrationName}/customerSubscriptions/{customerSubscriptionId}
api-version=2017-06-01 HTTP/1.1
```

#### Response: 204 No Content

#### Response Body: Empty

## Next steps

- [How to retrieve resource usage information from Azure Stack Hub](#)

# Report Azure Stack Hub usage data to Azure

5 minutes to read • [Edit Online](#)

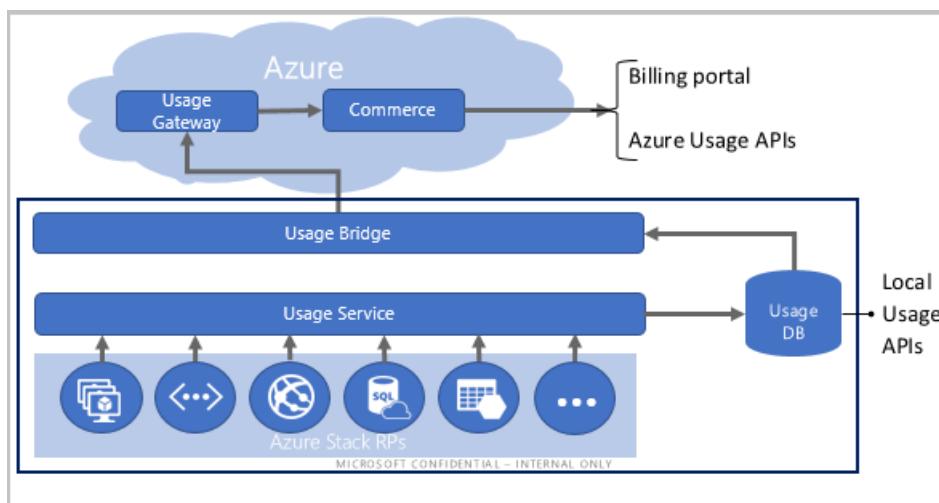
Usage data, also called consumption data, represents the amount of resources used.

Azure Stack Hub multi-node systems that use the consumption-based billing model should report usage data to Azure for billing purposes. Azure Stack Hub operators should configure their Azure Stack Hub instance to report usage data to Azure.

## IMPORTANT

All workloads [must be deployed under tenant subscriptions](#) to comply with the licensing terms of Azure Stack Hub.

Usage data reporting is required for the Azure Stack Hub multi-node users who license under the pay-as-you-use model. It's optional for customers who license under the capacity model (see the [How to buy](#) page). For Azure Stack Development Kit (ASDK) users, Azure Stack Hub operators can report usage data and test the feature. However, users won't be charged for any usage they incur.



Usage data is sent from Azure Stack Hub to Azure through the Azure Bridge. In Azure, the commerce system processes the usage data and generates the bill. After the bill is generated, the Azure subscription owner can view and download it from the [Azure Account Center](#). To learn about how Azure Stack Hub is licensed, see the [Azure Stack Hub packaging and pricing document](#).

## Set up usage data reporting

To set up usage data reporting, you must [register your Azure Stack Hub instance with Azure](#). As part of the registration process, the Azure Bridge component of Azure Stack Hub is configured. The Azure Bridge component is what connects Azure Stack Hub to Azure. The following usage data is sent from Azure Stack Hub to Azure:

- **Meter ID** - Unique ID for the resource that was consumed.
- **Quantity** - Amount of resource usage.
- **Location** - Location where the current Azure Stack Hub resource is deployed.
- **Resource URI** - Fully qualified URI of the resource for which usage is being reported.
- **Subscription ID** - Subscription ID of the Azure Stack Hub user, which is the local (Azure Stack Hub) subscription.

- **Time** - Start and end time of the usage data. There's some delay between the time when these resources are consumed in Azure Stack Hub and when the usage data is reported to commerce. Azure Stack Hub aggregates usage data for every 24 hours, and reporting usage data to the commerce pipeline in Azure takes another few hours. Therefore, usage that happens shortly before midnight can appear in Azure the following day.

## Generate usage data reporting

- To test usage data reporting, create a few resources in Azure Stack Hub. For example, you can create a [storage account](#), [Windows Server VM](#), and a Linux VM with Basic and Standard SKUs to see how core usage is reported. The usage data for different types of resources are reported under different meters.
- Leave your resources running for a few hours. Usage information is collected approximately once every hour. After collecting, this data is transmitted to Azure and processed into the Azure commerce system. This process can take up to a few hours.

## View usage - CSP subscriptions

If you registered your Azure Stack Hub using a CSP subscription, you can view your usage and charges in the same way you view Azure consumption. Azure Stack Hub usage is included in your invoice and in the reconciliation file, which is available through the [Partner Center](#). The reconciliation file is updated monthly. If you need to access recent Azure Stack Hub usage information, you can use the Partner Center APIs.

The screenshot shows the Microsoft Partner Center dashboard. On the left, there's a sidebar with links like Overview, Customers, Service requests, Service health, Product analytics, Azure spending, Activity log, Billing, Pricing and offers, Promotions, and Referrals. The main area has a "Welcome, Azure Stack!" message and a "Current tasks" section with a "New! Partner Center Analytics app for Power BI" card. Below that is a "Let us connect you with customers!" card. At the bottom of the main area, there's a "Billing" section showing an account balance of \$260,295.93, an August bill with an invoice and reconciliation link, and a last payment status. The top navigation bar includes links for Partner with us, Learn more, Find a Partner, Get support, and Dashboard.

## View usage - Enterprise Agreement subscriptions

If you registered your Azure Stack Hub using an Enterprise Agreement subscription, you can view your usage and charges in the [EA portal](#). Azure Stack Hub usage is included in the advanced downloads along with Azure usage under the reports section in this portal.

## View usage - other subscriptions

If you registered your Azure Stack Hub using any other subscription type (for example, a pay-as-you-go subscription), you can view usage and charges in the Azure Account Center. Sign in to the [Azure Account Center](#) as the Azure account administrator and select the Azure subscription that you used to register Azure Stack Hub. You can view the Azure Stack Hub usage data and the amount charged for each of the used resources, as shown in the following image:

The screenshot shows the Microsoft Azure Subscriptions dashboard. At the top, there's a navigation bar with links for HOME, PRICING, DOCUMENTATION, DOWNLOADS, COMMUNITY, SUPPORT, and ACCOUNT. The ACCOUNT section includes links for subscriptions, marketplace, profile, preview features, and a sign-out option. A blue button labeled 'Portal' with a right-pointing arrow is also present.

The main area is titled 'Summary for Pay-As-You-Go'. It has tabs for OVERVIEW and BILLING HISTORY. Below these are sections for 'USAGE YOU ARE RESPONSIBLE FOR' and 'CURRENT BALANCE'.

**Usage Details:**

- 0.00 GB STANDARD IO - TABLE (GB) - LOCALLY REDUNDANT
- 0.27 10,000s STANDARD IO - TABLE WRITE OPERATION UNITS (IN 10,000S) - DATA MANAGEMENT
- 27327.00 1 Core Hour VM - AZURE STACK
- 219.28 1 GB STORAGE - AZURE STACK
- 74186.00 1 Core Hour VM ADMIN - AZURE STACK
- 265.61 1 GB STORAGE ADMIN - AZURE STACK
- 428039.29 1 GB STORAGE - AZURE STACK
- 175.33 1 GB STORAGE - AZURE STACK
- 30330.12 1 GB STORAGE ADMIN - AZURE STACK

**Pricing Note:** A callout bubble highlights that pricing is zero dollars for the development kit. For multinode systems, actual pricing is displayed.

**Current Balance:** \$0.00

**Subscription Information:**

- DATE PURCHASED: 11/17/2016
- CURRENT BILLING PERIOD: 3/3/2017 - 4/2/2017

**Account Options:**

- Manage payment methods
- Download usage details
- Contact Microsoft Support
- Edit subscription details
- Change subscription address
- Partner information
- Switch to another offer
- Transfer subscription
- Cancel subscription

**Account Administrator:** [REDACTED]

**Subscription ID:** [REDACTED]

**Order ID:** [REDACTED]

**Offer:** Pay-As-You-Go

For the ASDK, Azure Stack Hub resources aren't charged, so the price shown is \$0.00.

## Which Azure Stack Hub deployments are charged?

Resource usage is free for the ASDK. Azure Stack Hub multi-node systems, workload VMs, storage services, and App Services are charged.

## Are users charged for the infrastructure VMs?

No. Usage data for some Azure Stack Hub resource provider VMs are reported to Azure, but there are no charges for these VMs, nor for the VMs created during deployment to enable the Azure Stack Hub infrastructure.

Users are only charged for VMs that run under tenant subscriptions. All workloads must be deployed under tenant subscriptions to comply with the licensing terms of Azure Stack Hub.

## I have a Windows Server license I want to use on Azure Stack Hub, how do I do it?

Using the existing licenses avoids generating usage meters. Existing Windows Server licenses can be used in Azure Stack Hub. This process is described in the "Using existing software with Azure Stack Hub" section of the

[Azure Stack Hub Licensing Guide](#). In order to use their existing licenses, customers must deploy their Windows Server VMs as described in [Hybrid benefit for Windows Server license](#).

## Which subscription is charged for the resources consumed?

The subscription that's provided when [registering Azure Stack Hub with Azure](#) is charged.

## What types of subscriptions are supported for usage data reporting?

For Azure Stack Hub multi-node, Enterprise Agreement (EA) and CSP subscriptions are supported. For the ASDK, Enterprise Agreement (EA), pay-as-you-go, CSP, and MSDN subscriptions support usage data reporting.

## Does usage data reporting work in sovereign clouds?

In the ASDK, usage data reporting requires subscriptions that are created in the global Azure system. Subscriptions created in one of the sovereign clouds (the Azure Government, Azure Germany, and Azure China 21Vianet clouds) can't be registered with Azure, so they don't support usage data reporting.

## Why doesn't the usage reported in Azure Stack Hub match the report generated from Azure Account Center?

There's always a delay between the usage data reported by the Azure Stack Hub usage APIs and the usage data reported in the Azure Account Center. This delay is the time required to upload usage data from Azure Stack Hub to Azure commerce. Because of this delay, usage that occurs shortly before midnight might appear in Azure the following day. If you use the [Azure Stack Hub usage APIs](#) and compare the results to the usage reported in the Azure billing portal, you can see a difference.

## Next steps

- [Provider usage API](#)
- [Tenant usage API](#)
- [Usage FAQ](#)
- [Manage usage and billing as a Cloud Solution Provider](#)

# Usage reporting infrastructure for Cloud Solution Providers

2 minutes to read • [Edit Online](#)

Azure Stack Hub includes the infrastructure needed to track usage as it occurs and forwards it to Azure. In Azure, Azure Commerce processes the [usage data and charges usage](#) to the appropriate Azure subscriptions. This process works in the same way as usage tracking in the global Azure cloud.

Some concepts are consistent between global Azure and Azure Stack Hub. Azure Stack Hub has local subscriptions, which fulfill a similar role to an Azure subscription. Local subscriptions are only valid locally. Local subscriptions are mapped to Azure subscriptions when usage is forwarded to Azure.

Azure Stack Hub has local usage meters. Local usage is mapped to the meters used in Azure commerce. However, the meter IDs are different. There are more meters available locally than the one Microsoft uses for billing.

There are some differences between how services are priced in Azure Stack Hub and Azure. For example, in Azure Stack Hub, the charge for VMs is only based on vcore/hours, with the same rate for all VM series, unlike Azure. The reason is that in global Azure the different prices reflect different hardware. In Azure Stack Hub, the customer provides the hardware, so there's no reason to charge different rates for different VM classes.

You can find out about the Azure Stack Hub meters used in Commerce and their prices in Partner Center. The process is the same as it is for Azure services:

1. In Partner Center, go to the **Dashboard** menu, then select **Sell**, then select **Pricing and offers**.
2. Under **Usage-based services**, select **Current**.
3. Open the **Azure in Global CSP price list** spreadsheet.
4. Filter on **Region = Azure Stack Hub**.

## Terms used for billing and usage

The following terms and concepts are used for usage and billing in Azure Stack Hub:

| TERM               | DEFINITION                                                                                                                                                                                                                                                                  |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Direct CSP partner | A direct CSP partner receives an invoice directly from Microsoft for Azure and Azure Stack Hub usage, and bills customers directly.                                                                                                                                         |
| Indirect CSP       | Indirect resellers work with an indirect provider (also known as a distributor). The resellers recruit end customers; the indirect provider holds the billing relationship with Microsoft, manages customer billing, and provides additional services like product support. |
| End customer       | End customers are the businesses and government agencies that own the apps and other workloads that run on Azure Stack Hub.                                                                                                                                                 |

## Next steps

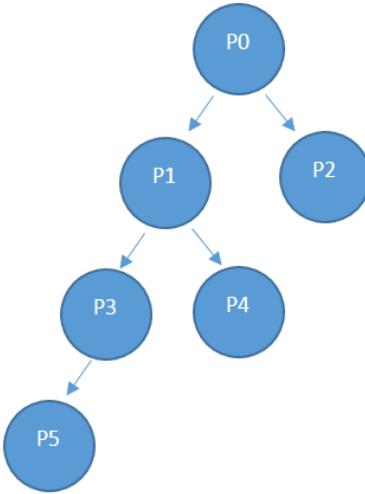
- To learn more about the CSP program, see [Cloud Solutions](#).

- To learn more about how to retrieve resource usage information from Azure Stack Hub, see [Usage and billing in Azure Stack Hub](#).

# Provider resource usage API

3 minutes to read • [Edit Online](#)

The term *provider* applies to the service administrator and to any delegated providers. Azure Stack Hub operators and delegated providers can use the provider usage API to view the usage of their direct tenants. For example, as shown in the following diagram, P0 can call the provider API to get direct usage information on P1 and P2, and P1 can call for usage information on P3 and P4.



## API call reference

### Request

The request gets consumption details for the requested subscriptions and for the requested time frame. There's no request body.

This usage API is a provider API, so the caller must be assigned an **Owner**, **Contributor**, or **Reader** role in the provider's subscription.

| METHOD | REQUEST URI                                                                                                                                                                                                                                                                                                                                                                                |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GET    | <code>https://{{armendpoint}}/subscriptions/{{subId}}/providers/Microsoft.Commerce.Admin/subscriptions/{{subId}}/consumption?reportedStartTime={{reportedStartTime}}&amp;reportedEndTime={{reportedEndTime}}&amp;aggregationGranularity={{aggregationGranularity}}&amp;subscriberId={{subscriberId}}&amp;api-version=2015-06-01-preview&amp;continuationToken={{continuationToken}}</code> |

### Arguments

| ARGUMENT                            | DESCRIPTION                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>armendpoint</code>            | Azure Resource Manager endpoint of your Azure Stack Hub environment. The Azure Stack Hub convention is that the name of the Azure Resource Manager endpoint is in the format <code>https://adminmanagement.{domain-name}</code> . For example, for the Azure Stack Development Kit (ASDK), if the domain name is <code>local.azurestack.external</code> , then the Resource Manager endpoint is <code>https://adminmanagement.local.azurestack.external</code> . |
| <code>subId</code>                  | Subscription ID of the user who makes the call.                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <code>reportedStartTime</code>      | Start time of the query. The value for <code>DateTime</code> should be in Coordinated Universal Time (UTC) and at the beginning of the hour; for example, 13:00. For daily aggregation, set this value to UTC midnight. The format is escaped ISO 8601; for example, <code>2015-06-16T18%3a53%3a11%2b00%3a00Z</code> , where the colon is escaped to <code>%3a</code> and the plus is escaped to <code>%2b</code> so that it's URI-friendly.                     |
| <code>reportedEndTime</code>        | End time of the query. The constraints that apply to <code>reportedStartTime</code> also apply to this argument. The value for <code>reportedEndTime</code> can't be either in the future, or the current date. If it is, the result is set to "processing not complete."                                                                                                                                                                                        |
| <code>aggregationGranularity</code> | Optional parameter that has two discrete potential values: <b>daily</b> and <b>hourly</b> . As the values suggest, one returns the data in daily granularity, and the other is an hourly resolution. The <b>daily</b> option is the default.                                                                                                                                                                                                                     |
| <code>subscriberId</code>           | Subscription ID. To get filtered data, the subscription ID of a direct tenant of the provider is required. If no subscription ID parameter is specified, the call returns usage data for all the provider's direct tenants.                                                                                                                                                                                                                                      |

| ARGUMENT                       | DESCRIPTION                                                                                                                                                                                                                                                                                          |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>api-version</code>       | Version of the protocol that's used to make this request. This value is set to <code>2015-06-01-preview</code> .                                                                                                                                                                                     |
| <code>continuationToken</code> | Token retrieved from the last call to the usage API provider. This token is needed when a response is greater than 1,000 lines. It acts as a bookmark for the progress. If the token isn't present, the data is retrieved from the beginning of the day or hour, based on the granularity passed in. |

## Response

```
GET
/subscriptions/sub1/providers/Microsoft.Commerce.Admin/subscriberUsageAggregates?reportedStartTime=reportedStartTime=2014-05-01T00%3a00%2b00%3a00&reportedEndTime=2015-06-01T00%3a00%2b00%3a00&aggregationGranularity=Daily&subscriberId=sub1.1&api-version=1.0
```

```
{
"value": [
{
"id":
"/subscriptions/sub1.1/providers/Microsoft.Commerce.Admin/UsageAggregate/sub1.1-
meterID1",
"name": "sub1.1-meterID1",
"type": "Microsoft.Commerce.Admin/UsageAggregate",

"properties": {
"subscriptionId":"sub1.1",
"usageStartTime": "2015-03-03T00:00:00+00:00",
"usageEndTime": "2015-03-04T00:00:00+00:00",
"instanceData": "{\"Microsoft.Resources\":{\"resourceUri\":\"resourceUri1\",\"location\":\"Alaska\",\"tags\":null,\"additionalInfo\":null}}",
"quantity":2.400000000,
"meterId":"meterID1"

},
},
...
}
```

## Response details

| ARGUMENT                    | DESCRIPTION                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>id</code>             | Unique ID of the usage aggregate.                                                                                                                                                                                                                                                                                                                                                            |
| <code>name</code>           | Name of the usage aggregate.                                                                                                                                                                                                                                                                                                                                                                 |
| <code>type</code>           | Resource definition.                                                                                                                                                                                                                                                                                                                                                                         |
| <code>subscriptionId</code> | Subscription identifier of the Azure Stack Hub user.                                                                                                                                                                                                                                                                                                                                         |
| <code>usageStartTime</code> | UTC start time of the usage bucket to which this usage aggregate belongs.                                                                                                                                                                                                                                                                                                                    |
| <code>usageEndTime</code>   | UTC end time of the usage bucket to which this usage aggregate belongs.                                                                                                                                                                                                                                                                                                                      |
| <code>instanceData</code>   | Key-value pairs of instance details (in a new format):<br>resourceUri : Fully qualified resource ID, which includes the resource groups and the instance name.<br>location : Region in which this service was run.<br>tags : Resource tags that are specified by the user.<br>additionalInfo : More details about the resource that was consumed; for example, the OS version or image type. |
| <code>quantity</code>       | Amount of resource consumption that occurred in this time frame.                                                                                                                                                                                                                                                                                                                             |
| <code>meterId</code>        | Unique ID for the resource that was consumed (also called <code>ResourceID</code> ).                                                                                                                                                                                                                                                                                                         |

## Retrieve usage information

### PowerShell

To generate the usage data, you should have resources that are running and actively using the system; for example, an active virtual machine (VM), or a storage account containing some data. If you're not sure whether you have any resources running in the Azure Stack Hub Marketplace, deploy a VM, and verify the VM monitoring blade to make sure it's running. Use the following PowerShell cmdlets to view the usage data:

1. [Install PowerShell for Azure Stack Hub](#).

2. Configure the Azure Stack Hub user or the [Azure Stack Hub operator](#) PowerShell environment.

3. To retrieve the usage data, call the [Get-AzsSubscriberUsage](#) PowerShell cmdlet:

```
Get-AzsSubscriberUsage -ReportedStartTime "2017-09-06T00:00:00Z" -ReportedEndTime "2017-09-07T00:00:00Z"
```

## REST API

You can collect usage information for deleted subscriptions by calling the **Microsoft.Commerce.Admin** service.

### Return all tenant usage for deleted for active users

| METHOD | REQUEST URI                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GET    | <a href="https://{{armendpoint}}/subscriptions/{{subId}}/providersMicrosoft.Commerce.Admin/subscriptions/{{subId}}/usage?reportedStartTime={{start-time}}&amp;reportedEndTime={{end-endtime}}&amp;aggregationGranularity={{aggregationGranularity}}&amp;version=2015-06-01-preview">https://{{armendpoint}}/subscriptions/{{subId}}/providersMicrosoft.Commerce.Admin/subscriptions/{{subId}}/usage?reportedStartTime={{start-time}}&amp;reportedEndTime={{end-endtime}}&amp;aggregationGranularity={{aggregationGranularity}}&amp;version=2015-06-01-preview</a> |

### Return usage for deleted or active tenant

| METHOD | REQUEST URI                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GET    | <a href="https://{{armendpoint}}/subscriptions/{{subId}}/providersMicrosoft.Commerce.Admin/subscriptions/{{subId}}/usage?reportedStartTime={{start-time}}&amp;reportedEndTime={{end-endtime}}&amp;aggregationGranularity={{aggregationGranularity}}&amp;subscriber-id={{subscriber-id}}&amp;api-version=2015-06-01-preview">https://{{armendpoint}}/subscriptions/{{subId}}/providersMicrosoft.Commerce.Admin/subscriptions/{{subId}}/usage?reportedStartTime={{start-time}}&amp;reportedEndTime={{end-endtime}}&amp;aggregationGranularity={{aggregationGranularity}}&amp;subscriber-id={{subscriber-id}}&amp;api-version=2015-06-01-preview</a> |

## Next steps

- [Tenant resource usage API reference](#)
- [Usage-related FAQ](#)

# Tenant resource usage API reference

2 minutes to read • [Edit Online](#)

A tenant can use the tenant APIs to view the tenant's own resource usage data. These APIs are consistent with the Azure usage APIs.

You can use the Windows PowerShell cmdlet [Get-UsageAggregates](#) to get usage data, just like in Azure.

## API call

### Request

The request gets consumption details for the requested subscriptions and for the requested time frame. There is no request body.

| METHOD | REQUEST URI                                                                                                                                                                                                                                                                                                   |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GET    | <code>https://{{armendpoint}}/subscriptions/{{subId}}/providers/Microsoft.Commerce/usageAggregates?reportedStartTime={{reportedStartTime}}&amp;reportedEndTime={{reportedEndTime}}&amp;aggregationGranularity={{granularity}}&amp;api-version=2015-06-01-preview&amp;continuationToken={{token-value}}</code> |

### Parameters

| PARAMETER         | DESCRIPTION                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Armendpoint       | Azure Resource Manager endpoint of your Azure Stack Hub environment. The Azure Stack Hub convention is that the name of Azure Resource Manager endpoint is in the format <code>https://management.{domain-name}</code> . For example, for the development kit, the domain name is local.azurestack.external, then the Resource Manager endpoint is <code>https://management.local.azurestack.external</code> . |
| subId             | Subscription ID of the user who is making the call. You can use this API only to query for a single subscription's usage. Providers can use the provider resource usage API to query usage for all tenants.                                                                                                                                                                                                    |
| reportedStartTime | Start time of the query. The value for <i>DateTime</i> should be in UTC and at the beginning of the hour; for example, 13:00. For daily aggregation, set this value to UTC midnight. The format is escaped ISO 8601; for example, <b>2015-06-16T18%3a53%3a11%2b00%3a00Z</b> , where colon is escaped to %3a and plus is escaped to %2b so that it's URI friendly.                                              |
| reportedEndTime   | End time of the query. The constraints that apply to <b>reportedStartTime</b> also apply to this parameter. The value for <b>reportedEndTime</b> can't be in the future.                                                                                                                                                                                                                                       |

| PARAMETER              | DESCRIPTION                                                                                                                                                                                                                                                                          |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| aggregationGranularity | Optional parameter that has two discrete potential values: <b>daily</b> and <b>hourly</b> . As the values suggest, one returns the data in daily granularity, and the other is an hourly resolution. The <b>daily</b> option is the default.                                         |
| api-version            | Version of the protocol that's used to make this request. You must use <b>2015-06-01-preview</b> .                                                                                                                                                                                   |
| continuationToken      | Token retrieved from the last call to the usage API provider. This token is needed when a response is greater than 1,000 lines. It acts as a bookmark for progress. If not present, the data is retrieved from the beginning of the day or hour, based on the granularity passed in. |

## Response

```
GET
/subscriptions/sub1/providers/Microsoft.Commerce/UsageAggregates?reportedStartTime=reportedStartTime=2014-05-01T00%3a00%3a00%2b00%3a00&reportedEndTime=2015-06-01T00%3a00%3a00%2b00%3a00&aggregationGranularity=Daily&api-version=1.0
```

```
{
"value": [
{
"id": "/subscriptions/sub1/providers/Microsoft.Commerce/UsageAggregate/sub1-meterID1",
"name": "sub1-meterID1",
"type": "Microsoft.Commerce/UsageAggregate",

"properties": {
"subscriptionId": "sub1",
"usageStartTime": "2015-03-03T00:00:00+00:00",
"usageEndTime": "2015-03-04T00:00:00+00:00",
"instanceData": "{\"Microsoft.Resources\": {\"resourceUri\":\"resourceUri1\", \"location\":\"Alaska\", \"tags\":null, \"additionalInfo\":null}}",
"quantity": 2.4000000000,
"meterId": "meterID1"
}
},
...
}
```

## Response details

| PARAMETER      | DESCRIPTION                                |
|----------------|--------------------------------------------|
| id             | Unique ID of the usage aggregate.          |
| name           | Name of the usage aggregate.               |
| type           | Resource definition.                       |
| subscriptionId | Subscription identifier of the Azure user. |

| PARAMETER      | DESCRIPTION                                                                                                                                                                                                                                                                                                                                                                                      |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| usageStartTime | UTC start time of the usage bucket to which this usage aggregate belongs.                                                                                                                                                                                                                                                                                                                        |
| usageEndTime   | UTC end time of the usage bucket to which this usage aggregate belongs.                                                                                                                                                                                                                                                                                                                          |
| instanceData   | Key-value pairs of instance details (in a new format):<br><i>resourceUri</i> : Fully qualified resource ID, including resource groups and instance name.<br><i>location</i> : Region in which this service was run.<br><i>tags</i> : Resource tags that the user specifies.<br><i>additionalInfo</i> : More details about the resource that was consumed. For example, OS version or image type. |
| quantity       | Amount of resource consumption that occurred in this time frame.                                                                                                                                                                                                                                                                                                                                 |
| meterId        | Unique ID for the resource that was consumed (also called <b>ResourceId</b> ).                                                                                                                                                                                                                                                                                                                   |

## Next steps

- [Provider resource usage API](#)
- [Usage-related FAQ](#)

# Frequently asked questions about Azure Stack Hub usage

10 minutes to read • [Edit Online](#)

This article answers some frequently asked questions about Azure Stack Hub usage and the Azure Stack Hub usage API.

## What meter IDs can I see?

Usage is reported for the following resource providers:

### Network

**Meter ID:** F271A8A388C44D93956A063E1D2FA80B

**Meter name:** Static IP Address Usage

**Unit:** IP addresses

**Notes:** Count of IP addresses used. If you call the usage API with a daily granularity, the meter returns IP address multiplied by the number of hours.

**Meter ID:** 9E2739BA86744796B465F64674B822BA

**Meter name:** Dynamic IP Address Usage

**Unit:** IP addresses

**Notes:** Count of IP addresses used. If you call the usage API with a daily granularity, the meter returns IP address multiplied by the number of hours.

### Storage

**Meter ID:** B4438D5D-453B-4EE1-B42A-DC72E377F1E4

**Meter name:** TableCapacity

**Unit:** GB\*hours

**Notes:** Total capacity consumed by tables.

**Meter ID:** B5C15376-6C94-4FDD-B655-1A69D138ACA3

**Meter name:** PageBlobCapacity

**Unit:** GB\*hours

**Notes:** Total capacity consumed by page blobs.

**Meter ID:** B03C6AE7-B080-4BFA-84A3-22C800F315C6

**Meter name:** QueueCapacity

**Unit:** GB\*hours

**Notes:** Total capacity consumed by queue.

**Meter ID:** 09F8879E-87E9-4305-A572-4B7BE209F857

**Meter name:** BlockBlobCapacity

**Unit:** GB\*hours

**Notes:** Total capacity consumed by block blobs.

**Meter ID:** B9FF3CD0-28AA-4762-84BB-FF8FBAEA6A90

**Meter name:** TableTransactions

**Unit:** Request count in 10,000s

**Notes:** Table service requests (in 10,000s).

**Meter ID:** 50A1AEAF-8ECA-48A0-8973-A5B3077FEE0D

**Meter name:** TableDataTransIn

**Unit:** Ingress data in GB

**Notes:** Table service data ingress in GB.

**Meter ID:** 1B8C1DEC-EE42-414B-AA36-6229CF199370

**Meter name:** TableDataTransOut

**Unit:** Egress in GB

**Notes:** Table service data egress in GB.

**Meter ID:** 43DAF82B-4618-444A-B994-40C23F7CD438

**Meter name:** BlobTransactions

**Unit:** Requests count in 10,000s

**Notes:** Blob service requests (in 10,000s).

**Meter ID:** 9764F92C-E44A-498E-8DC1-AAD66587A810

**Meter name:** BlobDataTransIn

**Unit:** Ingress data in GB

**Notes:** Blob service data ingress in GB.

**Meter ID:** 3023FEF4-ECA5-4D7B-87B3-CFBC061931E8

**Meter name:** BlobDataTransOut

**Unit:** Egress in GB

**Notes:** Blob service data egress in GB.

**Meter ID:** EB43DD12-1AA6-4C4B-872C-FAF15A6785EA

**Meter name:** QueueTransactions

**Unit:** Requests count in 10,000s

**Notes:** Queue service requests (in 10,000s).

**Meter ID:** E518E809-E369-4A45-9274-2017B29FFF25

**Meter name:** QueueDataTransIn

**Unit:** Ingress data in GB

**Notes:** Queue service data ingress in GB.

**Meter ID:** DD0A10BA-A5D6-4CB6-88C0-7D585CEF9FC2

**Meter name:** QueueDataTransOut

**Unit:** Egress in GB

**Notes:** Queue service data egress in GB.

## Compute

**Meter ID:** FAB6EB84-500B-4A09-A8CA-7358F8BBAEA5

**Meter name:** Base VM Size Hours

**Unit:** Virtual core hours

**Notes:** Number of virtual cores multiplied by the hours the VM ran.

**Meter ID:** 9CD92D4C-BAFD-4492-B278-BEDC2DE8232A

**Meter name:** Windows VM Size Hours

**Unit:** Virtual core hours

**Notes:** Number of virtual cores multiplied by hours the VM ran.

**Meter ID:** 6DAB500F-A4FD-49C4-956D-229BB9C8C793

**Meter name:** VM size hours

**Unit:** VM hours

**Notes:** Captures both base and Windows VM. Doesn't adjust for cores.

## Managed Disks

**Meter ID:** 380874f9-300c-48e0-95a0-d2d9a21ade8f **Meter name:** S4 **Unit:** Count of Disks\*month **Notes:** Standard Managed Disk - 32 GB

**Meter ID:** 1b77d90f-427b-4435-b4f1-d78adec53222 **Meter name:** S6 **Unit:** Count of Disks\*month **Notes:** Standard Managed Disk - 64 GB

**Meter ID:** d5f7731b-f639-404a-89d0-e46186e22c8d **Meter name:** S10 **Unit:** Count of Disks\*month **Notes:** Standard Managed Disk - 128 GB

**Meter ID:** ff85ef31-da5b-4eac-95dd-a69d6f97b18a **Meter name:** S15 **Unit:** Count of Disks\*month **Notes:** Standard Managed Disk - 256 GB

**Meter ID:** 88ea9228-457a-4091-adc9-ad5194f30b6e **Meter name:** S20 **Unit:** Count of Disks\*month **Notes:** Standard Managed Disk - 512 GB

**Meter ID:** 5b1db88a-8596-4002-8052-347947c26940 **Meter name:** S30 **Unit:** Count of Disks\*month **Notes:** Standard Managed Disk - 1024 GB

**Meter ID:** 7660b45b-b29d-49cb-b816-59f30fbab011 **Meter name:** P4 **Unit:** Count of Disks\*month **Notes:** Premium Managed Disk - 32 GB

**Meter ID:** 817007fd-a077-477f-bc01-b876f27205fd **Meter name:** P6 **Unit:** Count of Disks\*month **Notes:** Premium Managed Disk - 64 GB

**Meter ID:** e554b6bc-96cd-4938-a5b5-0da990278519 **Meter name:** P10 **Unit:** Count of Disks\*month **Notes:** Premium Managed Disk - 128 GB

**Meter ID:** cdc0f53a-62a9-4472-a06c-e99a23b02907 **Meter name:** P15 **Unit:** Count of Disks\*month **Notes:** Premium Managed Disk - 256 GB

**Meter ID:** b9cb2d1a-84c2-4275-aa8b-70d2145d59aa **Meter name:** P20 **Unit:** Count of Disks\*month **Notes:** Premium Managed Disk - 512 GB

**Meter ID:** 06bde724-9f94-43c0-84c3-d0fc54538369 **Meter name:** P30 **Unit:** Count of Disks\*month **Notes:** Premium Managed Disk - 1024 GB

**Meter ID:** 7ba084ec-ef9c-4d64-a179-7732c6cb5e28 **Meter name:** ActualStandardDiskSize **Unit:** GB\*month **Notes:** The actual size on disk of standard managed disk.

**Meter ID:** daef389a-06e5-4684-a7f7-8813d9f792d5

**Meter name:** ActualPremiumDiskSize **Unit:** GB\*month **Notes:** The actual size on disk of premium managed disk.

**Meter ID:** 108fa95b-be0d-4cd9-96e8-5b0d59505df1

**Meter name:** ActualStandardSnapshotSize **Unit:** GB\*month **Notes:** The actual size on disk of managed standard snapshot.

**Meter ID:** 578ae51d-4ef9-42f9-85ae-42b52d3d83ac **Meter name:** ActualPremiumSnapshotSize **Unit:** GB\*month **Notes:** The actual size on disk of managed premium snapshot.

**Meter ID:** 5d76e09f-4567-452a-94cc-7d1f097761f0 **Meter name:** S4 **Unit:** Count of Disks\*hours **Notes:** Standard Managed Disk - 32 GB (Deprecated)

**Meter ID:** dc9fc6a9-0782-432a-b8dc-978130457494 **Meter name:** S6 **Unit:** Count of Disks\*hours **Notes:** Standard Managed Disk - 64 GB (Deprecated)

**Meter ID:** e5572fce-9f58-49d7-840c-b168c0f01fff **Meter name:** S10 **Unit:** Count of Disks\*hours **Notes:** Standard Managed Disk - 128 GB (Deprecated)

**Meter ID:** 9a8caedd-1195-4cd5-80b4-a4c22f9302b8 **Meter name:** S15 **Unit:** Count of Disks\*hours **Notes:**

Standard Managed Disk - 256 GB (Deprecated)

**Meter ID:** 5938f8da-0ecd-4c48-8d5a-c7c6c23546be **Meter name:** S20 **Unit:** Count of Disks\*hours **Notes:** Standard Managed Disk - 512 GB (Deprecated)

**Meter ID:** 7705a158-bd8b-4b2b-b4c2-0782343b81e6 **Meter name:** S30 **Unit:** Count of Disks\*hours **Notes:** Standard Managed Disk - 1024 GB (Deprecated)

**Meter ID:** 5c105f5f-cbdf-435c-b49b-3c7174856dcc **Meter name:** P4 **Unit:** Count of Disks\*hours **Notes:** Premium Managed Disk - 32 GB (Deprecated)

**Meter ID:** 518b412b-1927-4f25-985f-4aea24e55c4f **Meter name:** P6 **Unit:** Count of Disks\*hours **Notes:** Premium Managed Disk - 64 GB (Deprecated)

**Meter ID:** 5cfb1fed-0902-49e3-8217-9add946fd624 **Meter name:** P10 **Unit:** Count of Disks\*hours **Notes:** Premium Managed Disk - 128 GB (Deprecated)

**Meter ID:** 8de91c94-f740-4d9a-b665-bd5974fa08d4 **Meter name:** P15

**Unit:** Count of Disks\*hours **Notes:** Premium Managed Disk - 256 GB (Deprecated)

**Meter ID:** c7e7839c-293b-4761-ae4c-848eda91130b **Meter name:** P20 **Unit:** Count of Disks\*hours **Notes:** Premium Managed Disk - 512 GB (Deprecated)

**Meter ID:** 9f502103-adf4-4488-b494-456c95d23a9f **Meter name:** P30 **Unit:** Count of Disks\*hours **Notes:** Premium Managed Disk - 1024 GB (Deprecated)

**Meter ID:** 8a409390-1913-40ae-917b-08d0f16f3c38 **Meter name:** ActualStandardDiskSize **Unit:** Byte\*hours **Notes:** The actual size on disk of standard managed disk (Deprecated).

**Meter ID:** 1273b16f-8458-4c34-8ce2-a515de551ef6

**Meter name:** ActualPremiumDiskSize **Unit:** Byte\*hours **Notes:** The actual size on disk of premium managed disk (Deprecated).

**Meter ID:** 89009682-df7f-44fe-aeb1-63fba3ddbf4c

**Meter name:** ActualStandardSnapshotSize **Unit:** Byte\*hours **Notes:** The actual size on disk of managed standard snapshot (Deprecated).

**Meter ID:** 95b0c03f-8a82-4524-8961-ccfbf575f536 **Meter name:** ActualPremiumSnapshotSize **Unit:** Byte\*hours **Notes:** The actual size on disk of managed premium snapshot (Deprecated).

**Meter ID:** 75d4b707-1027-4403-9986-6ec7c05579c8 **Meter name:** ActualStandardSnapshotSize **Unit:** GB\*month **Notes:** The actual size on disk of managed standard snapshot (Deprecated).

**Meter ID:** 5ca1ccb9-6f14-4e76-8be8-1ca91547965e **Meter name:** ActualPremiumSnapshotSize **Unit:** GB\*month **Notes:** The actual size on disk of managed premium snapshot (Deprecated).

## Sql RP

**Meter ID:** CBCFEF9A-B91F-4597-A4D3-01FE334BED82

**Meter name:** DatabaseSizeHourSqlMeter

**Unit:** MB\*hours

**Notes:** Total DB capacity at creation. If you call the usage API with a daily granularity, the meter returns MB multiplied by the number of hours.

## MySql RP

**Meter ID:** E6D8CFCD-7734-495E-B1CC-5AB0B9C24BD3

**Meter name:** DatabaseSizeHour MySqlMeter

**Unit:** MB\*hours

**Notes:** Total DB capacity at creation. If you call the usage API with a daily granularity, the meter returns MB multiplied by the number of hours.

## Key Vault

**Meter ID:** EBF13B9F-B3EA-46FE-BF54-396E93D48AB4

**Meter name:** Key Vault transactions

**Unit:** Request count in 10,000s

**Notes:** Number of REST API requests received by Key Vault data plane.

**Meter ID:** 2C354225-B2FE-42E5-AD89-14F0EA302C87

**Meter name:** Advanced keys transactions

**Unit:** 10K transactions

**Notes:** RSA 3K/4K, ECC key transactions (preview).

## App service

**Meter ID:** 190C935E-9ADA-48FF-9AB8-56EA1CF9ADAA

**Meter name:** App Service

**Unit:** Virtual core hours

**Notes:** Number of virtual cores used to run app service.

### NOTE

Microsoft uses this meter to charge the App Service on Azure Stack Hub. Cloud Solution Providers can use the other App Service meters (below) to calculate usage for their tenants.

**Meter ID:** 67CC4AFC-0691-48E1-A4B8-D744D1FEDBDE

**Meter name:** Functions Requests

**Unit:** 10 Requests

**Notes:** Total number of requested executions (per 10 executions). Executions are counted each time a function runs in response to an event, or is triggered by a binding.

**Meter ID:** D1D04836-075C-4F27-BF65-0A1130EC60ED

**Meter name:** Functions - Compute

**Unit:** GB-s

**Notes:** Resource consumption measured in gigabyte seconds (GB/s). **Observed resource consumption** is calculated by multiplying average memory size in GB by the time in milliseconds it takes to execute the function. Memory used by a function is measured by rounding up to the nearest 128 MB, up to the maximum memory size of 1,536 MB, with execution time calculated by rounding up to the nearest 1 ms. The minimum execution time and memory for a single function execution is 100 ms and 128 mb respectively.

**Meter ID:** 957E9F36-2C14-45A1-B6A1-1723EF71A01D

**Meter name:** Shared App Service Hours

**Unit:** 1 hour **Notes:** Per hour usage of shard App Service Plan. Plans are metered on a per App basis.

**Meter ID:** 539CDEC7-B4F5-49F6-AAC4-1F15CFF0EDA9

**Meter name:** Free App Service Hours

**Unit:** 1 hour **Notes:** Per hour usage of free App Service Plan. Plans are metered on a per App basis.

**Meter ID:** 88039D51-A206-3A89-E9DE-C5117E2D10A6

**Meter name:** Small Standard App Service Hours

**Unit:** 1 hour **Notes:** Calculated based on size and number of instances.

**Meter ID:** 83A2A13E-4788-78DD-5D55-2831B68ED825

**Meter name:** Medium Standard App Service Hours

**Unit:** 1 hour **Notes:** Calculated based on size and number of instances.

**Meter ID:** 1083B9DB-E9BB-24BE-A5E9-D6FDD0DDEFE6

**Meter name:** Large Standard App Service Hours

**Unit:** 1 hour **Notes:** Calculated based on size and number of instances.

### Custom Worker Tiers

**Meter ID:** Custom Worker Tiers **Meter name:** Custom Worker Tiers

**Unit:** Hours **Notes:** Deterministic meter ID is created based on SKU and custom worker tier name. This meter ID is unique for each custom worker tier.

**Meter ID:** 264ACB47-AD38-47F8-ADD3-47F01DC4F473

**Meter name:** SNI SSL

**Unit:** Per SNI SSL Binding

**Notes:** App Service supports two types of SSL connections: Server Name Indication (SNI) SSL Connections and IP Address SSL Connections. SNI-based SSL works on modern browsers while IP-based SSL works on all browsers.

**Meter ID:** 60B42D72-DC1C-472C-9895-6C516277EDB4

**Meter name:** IP SSL **Unit:** Per IP Based SSL Binding **Notes:** App Service supports two types of SSL connections: Server Name Indication (SNI) SSL Connections and IP Address SSL Connections. SNI-based SSL works on modern browsers while IP-based SSL works on all browsers.

**Meter ID:** 73215A6C-FA54-4284-B9C1-7E8EC871CC5B

**Meter name:** Web Process **Unit:**

**Notes:** Calculated per active site per hour.

**Meter ID:** 5887D39B-0253-4E12-83C7-03E1A93DFFD9

**Meter name:** External Egress Bandwidth

**Unit:** GB

**Notes:** Total incoming request response bytes + total outgoing request bytes + total incoming FTP request response bytes + total incoming web deploy request response bytes.

## How do the Azure Stack Hub usage APIs compare to the [Azure usage API](#) (currently in public preview)?

- The tenant usage API is consistent with the Azure API, with one exception: the *showDetails* flag currently isn't supported in Azure Stack Hub.
- The provider usage API applies only to Azure Stack Hub.
- Currently, the [RateCard API](#) that is available in Azure isn't available in Azure Stack Hub.

## What is the difference between usage time and reported time?

Usage data reports have two main time values:

- **Reported Time:** The time when the usage event entered the usage system.
- **Usage Time:** The time when the Azure Stack Hub resource was consumed.

You might see a discrepancy in values for usage time and reported time for a specific usage event. The delay can be as long as several hours in any environment.

Currently, you can query only by **Reported Time**.

## What do these usage API error codes mean?

| HTTP STATUS CODE | ERROR CODE          | DESCRIPTION                                        |
|------------------|---------------------|----------------------------------------------------|
| 400/Bad Request  | <i>NoApiVersion</i> | The <i>api-version</i> query parameter is missing. |

| HTTP STATUS CODE | ERROR CODE                            | DESCRIPTION                                                                                                                        |
|------------------|---------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| 400/Bad Request  | <i>InvalidProperty</i>                | A property is missing or has an invalid value. The message in the error code in the response body identifies the missing property. |
| 400/Bad Request  | <i>RequestEndTimeIsInFuture</i>       | The value for <i>ReportedEndTime</i> is in the future. Values in the future aren't allowed for this argument.                      |
| 400/Bad Request  | <i>SubscriberIdIsNotDirectTenant</i>  | A provider API call has used a subscription ID that isn't a valid tenant of the caller.                                            |
| 400/Bad Request  | <i>SubscriptionIdMissingInRequest</i> | The subscription ID of the caller is missing.                                                                                      |
| 400/Bad Request  | <i>InvalidAggregationGranularity</i>  | An invalid aggregation granularity was requested. Valid values are daily and hourly.                                               |
| 503              | <i>ServiceUnavailable</i>             | A retryable error occurred because the service is busy or the call is being throttled.                                             |

## What is the policy for charging for VMs?

Running and stopped VMs generate usage data. Consistent with Azure, deallocation is needed to stop the emission of usage data. In the case in which the portal is unavailable but the compute resource provider is still running, usage will be emitted.

## How do I extract usage data from the Azure Stack Hub usage APIs?

The easiest way to extract usage data from local usage APIs on an Azure Stack Hub is by using the [usage summary script on GitHub](#). The script requires the start and end dates as input parameters.

Alternatively, you can use the REST APIs, as explained in the [Provider resource usage API](#) and [Tenant resource usage API](#) articles.

## How can I associate usage extracted from Azure usage APIs to a specific Azure Stack Hub user subscription?

The usage records include a property bag called **additionalinfo**, which includes the Azure Stack Hub subscription ID. This ID is the user subscription emitting the corresponding usage record.

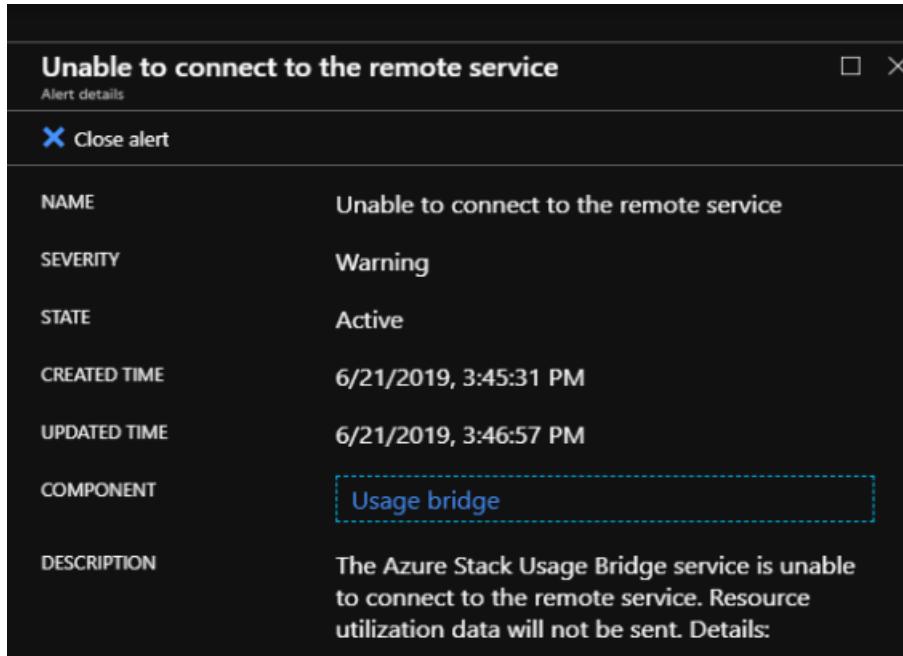
## Next steps

- [Customer billing and chargeback in Azure Stack Hub](#)
- [Provider Resource Usage API](#)
- [Tenant Resource Usage API](#)

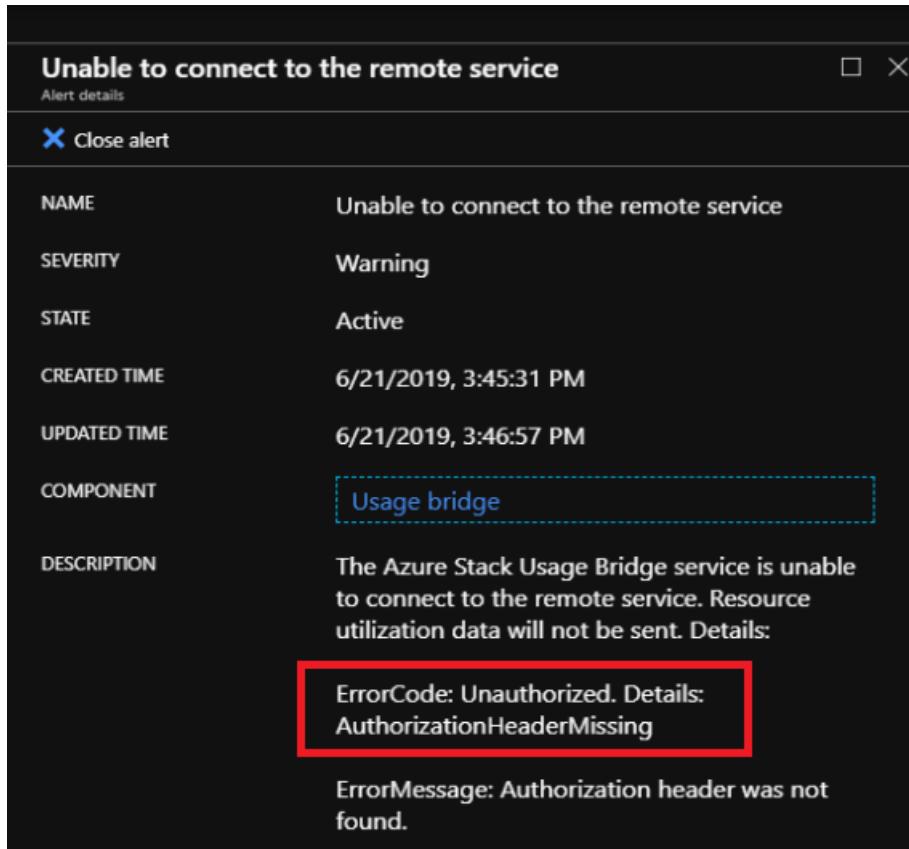
# Usage connectivity errors

2 minutes to read • [Edit Online](#)

Azure Stack Hub usage data is sent to Azure by the [Azure Bridge](#) component in Azure Stack Hub. If the bridge within Azure Stack Hub is unable to connect to the Azure usage service, the following error appears:



The window may provide more information about the error and resolution:



## Resolve connectivity issues

To mitigate the issue, try the following steps:

- Verify that network configuration allows the Azure Bridge to connect to the remote service.
- Go to the **Region Management > Properties** blade to find the Azure subscription ID used for the registration, resource group, and name of the registration resource. Verify that the registration resource exists under the correct Azure subscription ID in Azure portal. To do so, go to **All resources** created under the Azure subscription ID, and check the **Show hidden types** box. If you can't find the registration resource, follow the steps in [Renew or change registration](#) to re-register your Azure Stack Hub.

The screenshot shows the 'All resources' blade in the Azure portal. At the top, there are buttons for 'Add', 'Edit columns', 'Refresh', 'Export to CSV', 'Assign tags', 'Delete', 'Feedback', and a profile icon. Below these are filters for 'Subscription == all', 'Resource group == all', 'Type == all', and 'Location == all'. A search bar says 'Filter by name...'. Underneath, it says 'Showing 1 to 3 of 3 records.' and has a checked checkbox for 'Show hidden types'. The main table has columns for NAME, RESOURCE GROUP, and LOCATION. One record is listed: NAME is '<unique-registration-name>', RESOURCE GROUP is 'azorestack', and LOCATION is 'global'.

## Error codes

This section describes the usage error codes.

| ERROR CODE                 | ISSUE                                                                                                                                                  | REMEDIATION                                                                                                                                                                                                                                                                            |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NetworkError               | Azure Stack Hub bridge is unable to send request to usage service endpoint in Azure.                                                                   | Check if a proxy is blocking or intercepting access to the usage service endpoint.                                                                                                                                                                                                     |
| RequestTimedOut            | Request was sent from the Azure Bridge but the usage service in Azure failed to respond within the timeout period.                                     | Check if a proxy is blocking or intercepting access to the usage service endpoint.                                                                                                                                                                                                     |
| LoginError                 | Unable to authenticate with Microsoft Azure Active Directory.                                                                                          | Ensure the Azure AD login endpoint is accessible from all XRP VMs in Azure Stack Hub.                                                                                                                                                                                                  |
| CertificateValidationError | The Azure bridge is unable to send the request because it is unable to authenticate with the Azure service.                                            | Check if there is a proxy intercepting HTTPS traffic between the Azure Stack Hub XRP machine and the usage gateway endpoint.                                                                                                                                                           |
| Unauthorized               | The Azure bridge is unable to push data to the usage service in Azure, because the Azure service is unable to authenticate the Azure Stack Hub bridge. | Check if the registration resource has been modified, and if so, re-register Azure Stack Hub.<br>Sometimes, a time sync issue between Azure Stack Hub and Azure AD can cause this failure. In this case, ensure the times on the XRP VMs on Azure Stack Hub are in sync with Azure AD. |

Additionally, you may be required to provide the log files for the Azure Bridge, WAS, and WASPublic components

by following [these steps](#).

## Next steps

- Learn more about [reporting Azure Stack Hub usage data to Azure](#).
- To review error messages if they are triggered in your registration process, see [Tenant registration error messages](#).
- Learn more about the [Usage reporting infrastructure for Cloud Solution Providers](#).

# Usage and billing registration error codes

2 minutes to read • [Edit Online](#)

If you are a CSP, the following error messages can occur when [adding tenants](#) to a registration for reporting usage against the customer's Azure subscription ID.

## List of registration error codes

| ERROR                                | DETAILS                                                                                                                                                                                                                                                                                                                   | COMMENTS                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RegistrationNotFound</b>          | The provided registration was not found. Make sure the following information was provided correctly:<br>1. Subscription identifier (value provided: <b>subscription identifier</b> ),<br>2. Resource group (value provided: <b>resource group</b> ),<br>3. Registration name (value provided: <b>registration name</b> ). | This error usually occurs when the information pointing to the initial registration is not correct. If you need to verify the resource group and name of your registration, you can find it in the Azure portal, by listing all resources. If you find more than one registration resource, look at the <b>CloudDeploymentID</b> in the properties, and select the registration whose <b>CloudDeploymentID</b> matches that of your cloud. To find the <b>CloudDeploymentID</b> , you can use this PowerShell command on Azure Stack Hub:<br><pre>\$azureStackStampInfo = Invoke-Command -Session \$session -ScriptBlock { Get-AzureStackStampInformation }</pre> |
| <b>BadCustomerSubscriptionId</b>     | The provided <b>customer subscription identifier</b> and the <b>registration name</b> subscription identifier are not owned by the same Microsoft CSP. Check that the customer subscription identifier is correct. If the problem persists, contact support.                                                              | This error occurs when the customer subscription is a CSP subscription, but it rolls up to a CSP partner different from the one to which the subscription used in the initial registration rolls up. This check is made to prevent a situation that would result in billing a CSP partner who is not responsible for the Azure Stack Hub used.                                                                                                                                                                                                                                                                                                                    |
| <b>InvalidCustomerSubscriptionId</b> | The <b>customer subscription identifier</b> is not valid. Make sure a valid Azure subscription is provided.                                                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>CustomerSubscriptionNotFound</b>  | <b>Customer subscription identifier</b> was not found under <b>registration name</b> . Make sure a valid Azure subscription is being used and that the subscription ID was added to the registration using the PUT operation.                                                                                             | This error occurs when trying to verify that a tenant has been added to a subscription, and the customer subscription is not found to be associated with the registration. The customer has not been added to the registration, or the subscription ID has been written incorrectly.                                                                                                                                                                                                                                                                                                                                                                              |

| ERROR                                  | DETAILS                                                                                                                                                                                                                          | COMMENTS                                                                                                                                                                                                                                                                                                                                  |
|----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>UnauthorizedCspRegistration</b>     | The provided <b>registration name</b> is not approved to use multi-tenancy. Send an email to azstCSP@microsoft.com and include your registration name, resource group, and the subscription identifier used in the registration. | A registration must be approved for multi-tenancy by Microsoft before you can start adding tenants to it.                                                                                                                                                                                                                                 |
| <b>CustomerSubscriptionsNotAllowed</b> | Customer subscription operations are not supported for disconnected customers. In order to use this feature, re-register with Pay As You Use licensing.                                                                          | The registration to which you are trying to add tenants is a capacity registration; that is, when the registration was created, the parameter <code>BillingModel Capacity</code> was used. Only pay-as-you-use registrations are allowed to add tenants. You must re-register using the parameter <code>BillingModel PayAsYouUse</code> . |
| <b>InvalidCSPSubscription</b>          | The provided <b>customer subscription identifier</b> is not a valid CSP subscription. Make sure a valid Azure subscription is provided.                                                                                          | This is most likely due to the customer subscription being mistyped.                                                                                                                                                                                                                                                                      |
| <b>MetadataResolverBadGatewayError</b> | One of the upstream servers returned an unexpected error. Try again later. If the problem persists, contact support.                                                                                                             |                                                                                                                                                                                                                                                                                                                                           |

## Next steps

- Learn more about the [Usage reporting infrastructure for Cloud Solution Providers](#).
- To learn more about the CSP program, see [Cloud Solutions](#).
- To learn more about how to retrieve resource usage information from Azure Stack Hub, see [Usage and billing in Azure Stack Hub](#).

# Enable backup for Azure Stack Hub from the administrator portal

6 minutes to read • [Edit Online](#)

You can enable the Infrastructure Backup Service from the administrator portal so that Azure Stack Hub can generate infrastructure backups. The hardware partner can use these backups to restore your environment using cloud recovery in the event of a [catastrophic failure](#). The purpose of cloud recovery is to ensure that your operators and users can log back into the portal after recovery is complete. Users will have their subscriptions restored, including:

- Role-based access permissions and roles.
- Original plans and offers.
- Previously defined compute, storage, and network quotas.
- Key Vault secrets.

However, the Infrastructure Backup Service doesn't back up IaaS VMs, network configurations, and storage resources such as storage accounts, blobs, tables, and so on. Users logging in after cloud recovery won't see any of these previously existing resources. Platform as a Service (PaaS) resources and data are also not backed up by the service.

Admins and users are responsible for backing up and restoring IaaS and PaaS resources separately from the infrastructure backup processes. For info on backing up IaaS and PaaS resources, see the following links:

- [Protect VMs deployed on Azure Stack Hub](#)
- [Back up your app in Azure](#)
- [What is SQL Server on Azure VMs? \(Windows\)](#)

## Enable or reconfigure backup

1. Open the [Azure Stack Hub administrator portal](#).
2. Select **All services**, and then under the **ADMINISTRATION** category select **Infrastructure backup**. Choose **Configuration** in the **Infrastructure backup** blade.
3. Type the path to the **Backup storage location**. Use a Universal Naming Convention (UNC) string for the path to a file share hosted on a separate device. A UNC string specifies the location of resources such as shared files or devices. For the service, you can use an IP address. To ensure availability of the backup data after a disaster, the device should be in a separate location.

### NOTE

If your environment supports name resolution from the Azure Stack Hub infrastructure network to your enterprise environment, you can use a Fully Qualified Domain Name (FQDN) rather than the IP.

4. Type the **Username** using the domain and username with sufficient access to read and write files. For example, `Contoso\backupshareuser`.
5. Type the **Password** for the user.
6. Type the password again to **Confirm Password**.

7. The **frequency in hours** determines how often backups are created. The default value is 12. Scheduler supports a maximum of 12 and a minimum of 4.
8. The **retention period in days** determines how many days of backups are preserved on the external location. The default value is 7. Scheduler supports a maximum of 14 and a minimum of 2. Backups older than the retention period are automatically deleted from the external location.

#### NOTE

If you want to archive backups older than the retention period, make sure to back up the files before the scheduler deletes the backups. If you reduce the backup retention period (e.g. from 7 days to 5 days), the scheduler will delete all backups older than the new retention period. Make sure you're OK with the backups getting deleted before you update this value.

9. In Encryption Settings, provide a certificate in the Certificate .cer file box. Backup files are encrypted using this public key in the certificate. Provide a certificate that only contains the public key portion when you configure backup settings. Once you set this certificate for the first time or rotate the certificate in the future, you can only view the thumbprint of the certificate. You can't download or view the uploaded certificate file. To create the certificate file, run the following PowerShell command to create a self-signed certificate with the public and private keys and export a certificate with only the public key portion. You can save the certificate anywhere that can be accessed from admin portal.

```
$cert = New-SelfSignedCertificate `
 -DnsName "www.contoso.com" `
 -CertStoreLocation "cert:\LocalMachine\My"

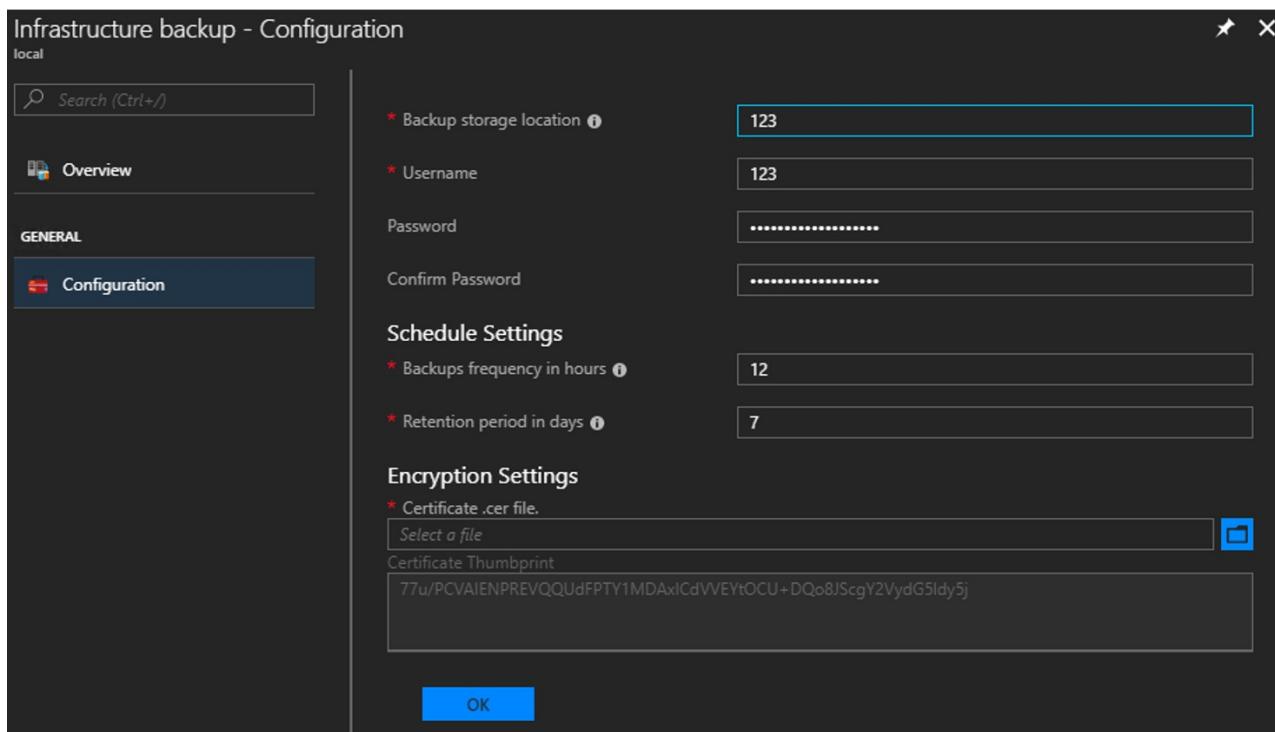
New-Item -Path "C:\" -Name "Certs" -ItemType "Directory"
Export-Certificate `
 -Cert $cert `
 -FilePath c:\certs\AzSIBCCert.cer
```

#### NOTE

**1901 and above:** Azure Stack Hub accepts a certificate to encrypt infrastructure backup data. Make sure to store the certificate with the public and private key in a secure location. For security reasons, it's not recommended that you use the certificate with the public and private keys to configure backup settings. For more info on how to manage the lifecycle of this certificate, see [Infrastructure Backup Service best practices](#).

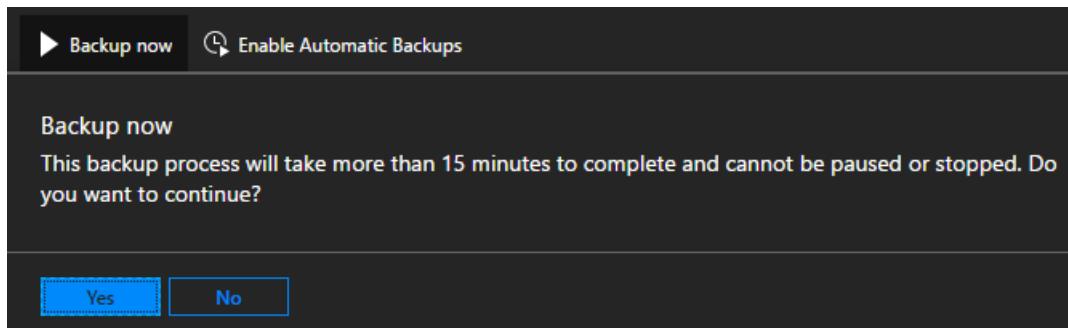
**1811 or earlier:** Azure Stack Hub accepts a symmetric key to encrypt infrastructure backup data. Use the [New-AzEncryptionKey64 cmdlet to create a key](#). After you upgrade from 1811 to 1901, backup settings will retain the encryption key. We recommend you update backup settings to use a certificate. Encryption key support is now deprecated. You have at least 3 releases to update settings to use a certificate.

10. Select **OK** to save your backup controller settings.



## Start backup

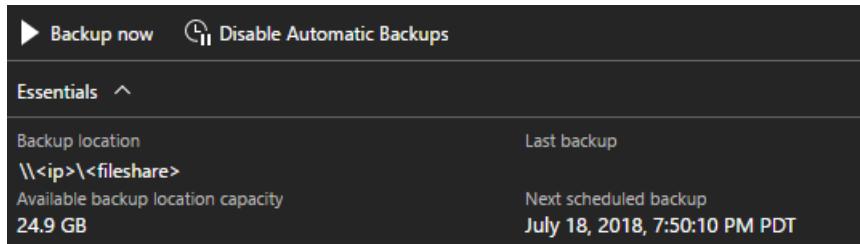
To start a backup, click on **Backup now** to start an on-demand backup. An on-demand backup won't modify the time for the next scheduled backup. After the task completes, you can confirm the settings in **Essentials**:



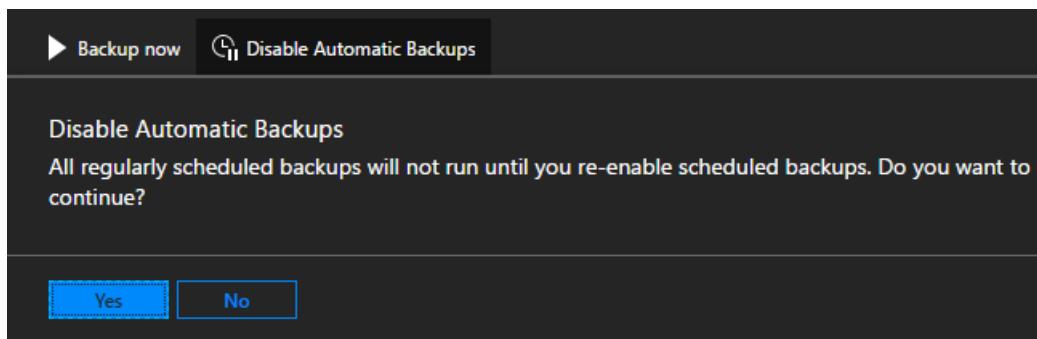
You can also run the PowerShell cmdlet **Start-AzsBackup** on your Azure Stack Hub admin computer. For more info, see [Back up Azure Stack Hub](#).

## Enable or disable automatic backups

Backups are automatically scheduled when you enable backup. You can check the next schedule backup time in **Essentials**.



If you need to disable future scheduled backups, click on **Disable Automatic Backups**. Disabling automatic backups keeps backup settings configured and retains the backup schedule. This action simply tells the scheduler to skip future backups.

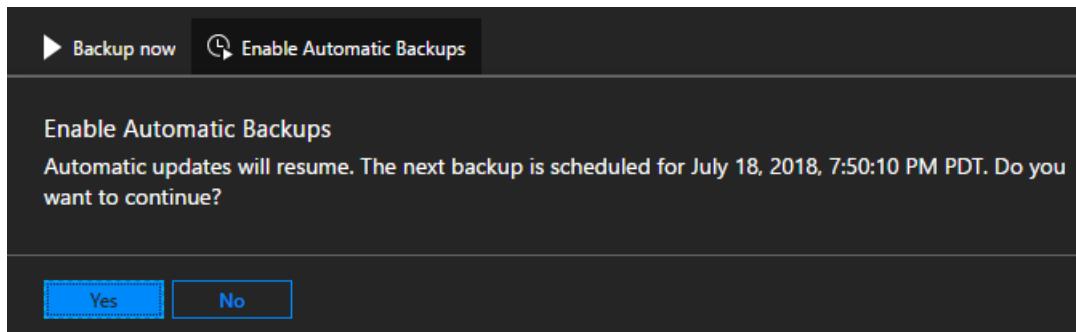


Confirm that future scheduled backups have been disabled in **Essentials**:

The screenshot shows the 'Essentials' section of the Azure Stack Hub interface:

- Backup location: '\\<ip>\<fileshare>'.
- Last backup: July 19, 2018, 7:12:56 AM PDT.
- Available backup location capacity: 2.14 TB.
- Next scheduled backup: (Disabled) July 19, 2018, 3:12:51 PM PDT.

Click on **Enable Automatic Backups** to inform the scheduler to start future backups at the scheduled time.



#### NOTE

If you configured infrastructure backup before updating to 1807, automatic backups will be disabled. This way the backups started by Azure Stack Hub don't conflict with backups started by an external task scheduling engine. Once you disable any external task scheduler, click on **Enable Automatic Backups**.

## Update backup settings

As of 1901, support for encryption key is deprecated. If you're configuring backup for the first time in 1901, you must use a certificate. Azure Stack Hub supports encryption key only if the key is configured before updating to 1901. Backward compatibility mode will continue for three releases. After that, encryption keys will no longer be supported.

#### Default mode

In encryption settings, if you're configuring infrastructure backup for the first time after installing or updating to 1901, you must configure backup with a certificate. Using an encryption key is no longer supported.

To update the certificate used to encrypt backup data, upload a new .CER file with the public key portion and select OK to save settings.

New backups will start to use the public key in the new certificate. There's no impact to all existing backups created with the previous certificate. Make sure to keep the older certificate around in a secure location in case you need it for cloud recovery.

## Encryption Settings

\* Certificate .cer file.

Select a file



Certificate Thumbprint

77u/PCVAIENPREVQQUdFPTY1MDAxICdVVEYtOCU+DQo8JScgY2VydG5Idy5j

OK

### Backwards compatibility mode

If you configured backup before updating to 1901, the settings are carried over with no change in behavior. In this case, the encryption key is supported for backwards compatibility. You can update the encryption key or switch to use a certificate. You have at least three releases to continue updating the encryption key. Use this time to transition to a certificate. To create a new encryption key, use [New-AzsEncryptionKeyBase64](#).

## Encryption Settings

Use encryption key  Use certificate

\* Encryption Key

\*\*\*\*\*

OK

### NOTE

Updating from encryption key to certificate is a one-way operation. After making this change, you can't switch back to encryption key. All existing backups will remain encrypted with the previous encryption key.

## Encryption Settings

Use encryption key  Use certificate

\* Certificate .cer file.

Select a file



Providing a certificate will replace the encryption key. New backups will use the certificate for encryption. All existing backups will continue to use the encryption key.

OK

## Next steps

Learn to run a backup. See [Back up Azure Stack Hub](#).

Learn to verify that your backup ran. See [Confirm backup completed in administrator portal](#).

# Enable Backup for Azure Stack Hub with PowerShell

4 minutes to read • [Edit Online](#)

Enable the Infrastructure Backup Service with Windows PowerShell to take periodic backups of:

- Internal identity service and root certificate.
- User plans, offers, subscriptions.
- Compute, storage, and network user quotas.
- User Key Vault secrets.
- User RBAC roles and policies.
- User storage accounts.

You can access the PowerShell cmdlets to enable backup, start backup, and get backup information via the operator management endpoint.

## Prepare PowerShell environment

For instructions on configuring the PowerShell environment, see [Install PowerShell for Azure Stack Hub](#). To sign in to Azure Stack Hub, see [Configure the operator environment and sign in to Azure Stack Hub](#).

## Provide the backup share, credentials, and encryption key to enable backup

In the same PowerShell session, edit the following PowerShell script by adding the variables for your environment. Run the updated script to provide the backup share, credentials, and encryption key to the Infrastructure Backup Service.

| VARIABLE                | DESCRIPTION                                                                                                                                                                                                                                                                                                                                       |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \$username              | Type the <b>Username</b> using the domain and username for the shared drive location with sufficient access to read and write files. For example, <code>Contoso\backupshareuser</code> .                                                                                                                                                          |
| \$password              | Type the <b>Password</b> for the user.                                                                                                                                                                                                                                                                                                            |
| \$sharepath             | Type the path to the <b>Backup storage location</b> . You must use a Universal Naming Convention (UNC) string for the path to a file share hosted on a separate device. A UNC string specifies the location of resources such as shared files or devices. To ensure availability of the backup data, the device should be in a separate location. |
| \$frequencyInHours      | The frequency in hours determines how often backups are created. The default value is 12. Scheduler supports a maximum of 12 and a minimum of 4.                                                                                                                                                                                                  |
| \$retentionPeriodInDays | The retention period in days determines how many days of backups are preserved on the external location. The default value is 7. Scheduler supports a maximum of 14 and a minimum of 2. Backups older than the retention period get automatically deleted from the external location.                                                             |

| VARIABLE             | DESCRIPTION                                                                                                                                                                                                                                 |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \$encryptioncertpath | Applies to 1901 and later. Parameter is available in Azure Stack Hub Module version 1.7 and later. The encryption certificate path specifies the file path to the .CER file with public key used for data encryption.                       |
| \$encryptionkey      | Applies to build 1811 or earlier. Parameter is available in Azure Stack Hub Module version 1.6 or earlier. The encryption key is used for data encryption. Use the <a href="#">New-AzsEncryptionKeyBase64</a> cmdlet to generate a new key. |

## Enable backup on 1901 and later using certificate

```
Example username:
$username = "domain\backupadmin"

Example share path:
$sharepath = "\serverIP\AzSBackupStore\contoso.com\seattle"

$password = Read-Host -Prompt ("Password for: " + $username) -AsSecureString

Create a self-signed certificate using New-SelfSignedCertificate, export the public key portion and save it locally.

$cert = New-SelfSignedCertificate `

-DnsName "www.contoso.com" `

-CertStoreLocation "cert:\LocalMachine\My"

New-Item -Path "C:\" -Name "Certs" -ItemType "Directory"

#make sure to export the PFX format of the certificate with the public and private keys and then delete the certificate from the local certificate store of the machine where you created the certificate

Export-Certificate `

-Cert $cert `

-FilePath c:\certs\AzSIBCCert.cer

Set the backup settings with the name, password, share, and CER certificate file.
Set-AzsBackupConfiguration -BackupShare $sharepath -Username $username -Password $password -EncryptionCertPath "c:\temp\cert.cer"
```

## Enable backup on 1811 or earlier using certificate

```
Example username:
$username = "domain\backupadmin"

Example share path:
$sharepath = "\serverIP\AzSBackupStore\contoso.com\seattle"

$password = Read-Host -Prompt ("Password for: " + $username) -AsSecureString

Create a self-signed certificate using New-SelfSignedCertificate, export the public key portion and save it locally.

$key = New-AzsEncryptionKeyBase64
$Securekey = ConvertTo-SecureString -String ($key) -AsPlainText -Force

Set the backup settings with the name, password, share, and CER certificate file.
Set-AzsBackupConfiguration -BackupShare $sharepath -Username $username -Password $password -EncryptionKey $Securekey
```

## Confirm backup settings

In the same PowerShell session, run the following commands:

```
Get-AzsBackupConfiguration | Select-Object -Property Path, UserName
```

The result should look like the following example output:

|          |   |                                              |
|----------|---|----------------------------------------------|
| Path     | : | \serverIP\AzsBackupStore\contoso.com\seattle |
| UserName | : | domain\backupadmin                           |

## Update backup settings

In the same PowerShell session, you can update the default values for retention period and frequency for backups.

```
#Set the backup frequency and retention period values.
$frequencyInHours = 10
$retentionPeriodInDays = 5

Set-AzsBackupConfiguration -BackupFrequencyInHours $frequencyInHours -BackupRetentionPeriodInDays
$retentionPeriodInDays

Get-AzsBackupConfiguration | Select-Object -Property Path, UserName, AvailableCapacity,
BackupFrequencyInHours, BackupRetentionPeriodInDays
```

The result should look like the following example output:

|                             |   |                                              |
|-----------------------------|---|----------------------------------------------|
| Path                        | : | \serverIP\AzsBackupStore\contoso.com\seattle |
| UserName                    | : | domain\backupadmin                           |
| AvailableCapacity           | : | 60 GB                                        |
| BackupFrequencyInHours      | : | 10                                           |
| BackupRetentionPeriodInDays | : | 5                                            |

### Azure Stack Hub PowerShell

The PowerShell cmdlet to configure infrastructure backup is Set-AzsBackupConfiguration. In previous releases, the cmdlet was Set-AzsBackupShare. This cmdlet requires providing a certificate. If infrastructure backup is configured with an encryption key, you can't update the encryption key or view the property. You need to use version 1.6 of the Admin PowerShell.

If infrastructure backup was configured before updating to 1901, you can use version 1.6 of the admin PowerShell to set and view the encryption key. Version 1.6 won't allow you to update from encryption key to a certificate file. Refer to [Install Azure Stack Hub PowerShell](#) for more info on installing the correct version of the module.

## Next steps

Learn to run a backup, see [Back up Azure Stack Hub](#).

Learn to verify that your backup ran, see [Confirm backup completed in administration portal](#).

# Back up Azure Stack Hub

2 minutes to read • [Edit Online](#)

This article shows you how to do an on-demand backup on Azure Stack Hub. For instructions on configuring the PowerShell environment, see [Install PowerShell for Azure Stack Hub](#). To sign in to Azure Stack Hub, see [Using the administrator portal in Azure Stack Hub](#).

## Start Azure Stack Hub backup

### Start a new backup without job progress tracking

Use Start-AzSBackup to start a new backup immediately with no job progress tracking.

```
Start-AzsBackup -Force
```

### Start Azure Stack Hub backup with job progress tracking

Use Start-AzSBackup to start a new backup with the **-AsJob** parameter and save it as a variable to track backup job progress.

#### NOTE

Your backup job appears as successfully completed in the portal about 10-15 minutes before the job finishes.

The actual status is better observed via the code below.

#### IMPORTANT

The initial 1 millisecond delay is introduced because the code is too quick to register the job correctly and it comes back with no **PSBeginTime** and in turn with no **State** of the job.

```

$BackupJob = Start-AzsBackup -Force -AsJob
While (!$BackupJob.PSBeginTime) {
 Start-Sleep -Milliseconds 1
}
Write-Host "Start time: $($BackupJob.PSBeginTime)"
While ($BackupJob.State -eq "Running") {
 Write-Host "Job is currently: $($BackupJob.State) - Duration: $((New-Timespan -Start
($BackupJob.PSBeginTime) -End (Get-Date)).ToString().Split(".").[0])"
 Start-Sleep -Seconds 30
}

If ($BackupJob.State -eq "Completed") {
 Get-AzsBackup | Where-Object {$_._BackupId -eq $BackupJob.Output.BackupId}
 $Duration = $BackupJob.Output.TimeTakenToCreate
 $Pattern = '^P?T?((?<Years>\d+)Y)?((?<Months>\d+)M)?((?<Weeks>\d+)W)?((?<Days>\d+)D)?(T((?
<Hours>\d+)H)?((?<Minutes>\d+)M)?((?<Seconds>\d*(\.)?\d*)S)?)'
 If ($Duration -match $Pattern) {
 If (!$Matches.ContainsKey("Hours")) {
 $Hours = ""
 }
 Else {
 $Hours = ($Matches.Hours).ToString + 'h '
 }
 $Minutes = ($Matches.Minutes)
 $Seconds = [math]::round(($Matches.Seconds))
 $Runtime = '{0}{1:00}m {2:00}s' -f $Hours, $Minutes, $Seconds
 }
 Write-Host "BackupJob: $($BackupJob.Output.BackupId) - Completed with Status:
 $($BackupJob.Output.Status) - It took: $($Runtime) to run" -ForegroundColor Green
}
ElseIf ($BackupJob.State -ne "Completed") {
 $BackupJob
 $BackupJob.Output
}

```

## Confirm backup has completed

### Confirm backup has completed using PowerShell

Use the following PowerShell commands to ensure the backup has completed successfully:

```
Get-AzsBackup
```

The result should look like the following output:

```

BackupDataVersion : 1.0.1
BackupId : <backup ID>
RoleStatus : {NRP, SRP, CRP, KeyVaultInternalControlPlane...}
Status : Succeeded
CreatedDateTime : 7/6/2018 6:46:24 AM
TimeTakenToCreate : PT20M32.364138S
DeploymentID : <deployment ID>
StampVersion : 1.1807.0.41
OemVersion :
Id : /subscriptions/<subscription
ID>/resourceGroups/System.local/providers/Microsoft.Backup.Admin/backupLocations/local/backups/<backup ID>
Name : local/<local name>
Type : Microsoft.Backup.Admin/backupLocations/backups
Location : local
Tags : {}

```

### Confirm backup has completed in the administrator portal

Use the Azure Stack Hub administrator portal to verify that backup has completed successfully by following these steps:

1. Open the [Azure Stack Hub administrator portal](#).
2. Select **All services**, and then under the **ADMINISTRATION** category select > **Infrastructure backup**. Choose **Configuration** in the **Infrastructure backup** blade.
3. Find the **Name** and **Date Completed** of the backup in **Available backups** list.
4. Verify the **State** is **Succeeded**.

## Next steps

Learn more about the workflow for [recovering from a data loss event](#).

# Recover data in Azure Stack Hub with the Infrastructure Backup Service

2 minutes to read • [Edit Online](#)

You can back up and restore configuration and service data using the Azure Stack Hub Infrastructure Backup Service. Each Azure Stack Hub installation contains an instance of the service. You can use backups created by the service for the redeployment of the Azure Stack Hub cloud to restore identity, security, and Azure Resource Manager data.

Enable backup when you're ready to put your cloud into production. Don't enable backup if you plan to perform testing and validation for a long period of time.

Before you enable your backup service, make sure you have the [requirements in place](#).

## NOTE

The Infrastructure Backup Service doesn't include user data and apps. For more info on how to protect IaaS VM-based apps, see [protect VMs deployed on Azure Stack Hub](#). For a comprehensive understanding of how to protect apps on Azure Stack Hub, see the [Azure Stack Hub considerations for business continuity and disaster recovery whitepaper](#).

## The Infrastructure Backup Service

The service contains the following features:

| FEATURE                                            | DESCRIPTION                                                                                                                                               |
|----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Backup Infrastructure Services                     | Coordinate backup across a subset of infrastructure services in Azure Stack Hub. If there's a disaster, the data can be restored as part of redeployment. |
| Compression and encryption of exported backup data | Backup data is compressed and encrypted by the system before it's exported to the external storage location provided by the admin.                        |
| Backup job monitoring                              | System notifies you when backup jobs fail and how to fix the problem.                                                                                     |
| Backup management experience                       | Backup RP supports enabling backup.                                                                                                                       |
| Cloud recovery                                     | If there's a catastrophic data loss, backups can be used to restore core Azure Stack Hub info as part of deployment.                                      |

## Verify requirements for the Infrastructure Backup Service

### • Storage location

You need a file share accessible from Azure Stack Hub that can contain seven backups. Each backup is about 10 GB. Your share should be able to store 140 GB of backups. For more info on selecting a storage location for the Infrastructure Backup Service, see [Backup Controller requirements](#).

### • Credentials

You need a domain user account and credentials. For example, you can use your Azure Stack Hub admin

credentials.

- **Encryption certificate**

Backup files are encrypted using the public key in the certificate. Make sure to store this certificate in a secure location.

## Next steps

Learn how to [Enable Backup for Azure Stack Hub from the administrator portal](#).

Learn how to [Enable Backup for Azure Stack Hub with PowerShell](#).

Learn how to [Back up Azure Stack Hub](#).

Learn how to [Recover from catastrophic data loss](#).

# Recover from catastrophic data loss

3 minutes to read • [Edit Online](#)

Azure Stack Hub runs Azure services in your datacenter and can run on environments as small as four nodes installed in a single rack. In contrast, Azure runs in more than 40 regions in multiple datacenters and multiple zones in each region. User resources can span multiple servers, racks, datacenters, and regions. With Azure Stack Hub, you currently only have the choice to deploy your entire cloud to a single rack. This limitation exposes your cloud to the risk of catastrophic events at your datacenter or failures due to major product bugs. When a disaster strikes, the Azure Stack Hub instance goes offline. All of the data is potentially unrecoverable.

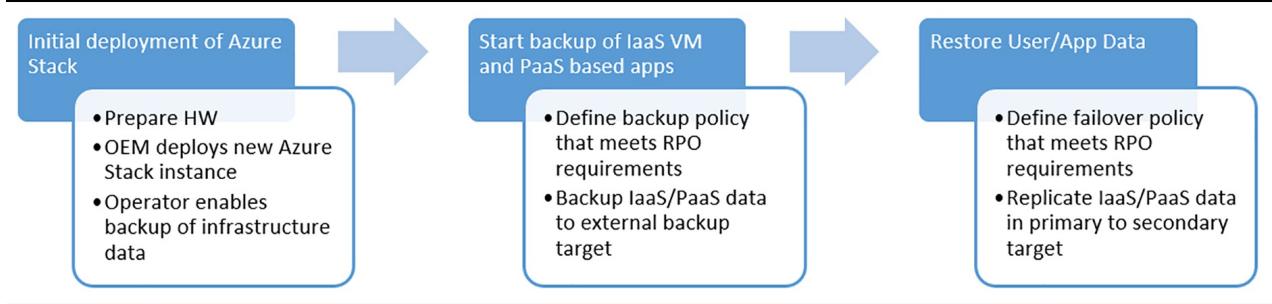
Depending on the root cause of the data loss, you may need to repair a single infrastructure service or restore the entire Azure Stack Hub instance. You may even need to restore to different hardware in the same location or in a different location.

This scenario addresses recovering your entire installation if there's a failure and the redeployment of the private cloud.

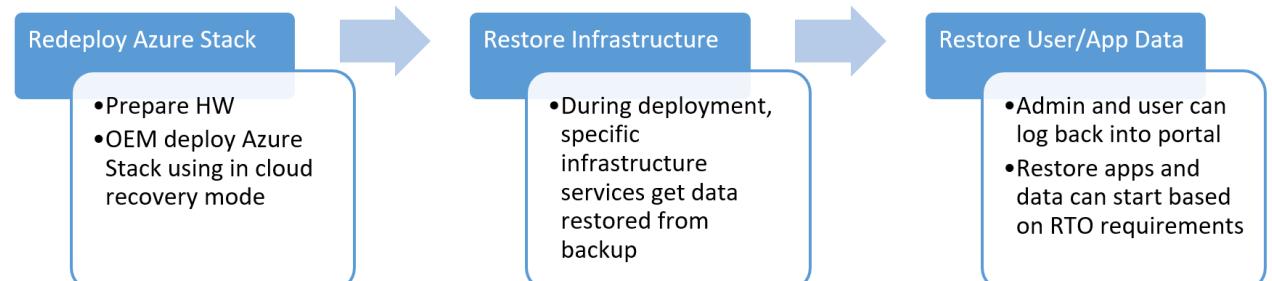
| SCENARIO                                                            | DATA LOSS                                 | CONSIDERATIONS                                                       |
|---------------------------------------------------------------------|-------------------------------------------|----------------------------------------------------------------------|
| Recover from catastrophic data loss due to disaster or product bug. | All infrastructure and user and app data. | User app and data are protected separately from infrastructure data. |

## Workflows

The journey of protecting Azure Start starts with backing up the infrastructure and app/tenant data separately. This document covers how to protect the infrastructure.



In worst case scenarios where all data is lost, recovering Azure Stack Hub is the process of restoring the infrastructure data unique to that deployment of Azure Stack Hub and all user data.



## Restore

If there's catastrophic data loss but the hardware is still usable, redeployment of Azure Stack Hub is required.

During redeployment, you can specify the storage location and credentials required to access backups. In this mode, there's no need to specify the services that need to be restored. Infrastructure Backup Controller injects control plane state as part of the deployment workflow.

If there's a disaster that renders the hardware unusable, redeployment is only possible on new hardware. Redeployment can take several weeks while replacement hardware is ordered and arrives in the datacenter. Restore of control plane data is possible at any time. However, restore isn't supported if the version of the redeployed instance is more than one version greater than the version used in the last backup.

| DEPLOYMENT MODE | STARTING POINT | END POINT                                                                                                                                                                                                  |
|-----------------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Clean install   | Baseline build | OEM deploys Azure Stack Hub and updates to the latest supported version.                                                                                                                                   |
| Recovery mode   | Baseline build | OEM deploys Azure Stack Hub in recovery mode and handles the version matching requirements based on the latest backup available. The OEM completes the deployment by updating to latest supported version. |

## Data in backups

Azure Stack Hub supports a type of deployment called cloud recovery mode. This mode is used only if you choose to recover Azure Stack Hub after a disaster or product bug rendered the solution unrecoverable. This deployment mode doesn't recover any of the user data stored in the solution. The scope of this deployment mode is limited to restoring the following data:

- Deployment inputs
- Internal identity service data (ADFS deployments).
- Federated identify configuration (ADFS deployments).
- Root certificates used by internal certificate authority.
- Azure Resource Manager configuration user data, such as subscriptions, plans, offers, storage quotas, network quotas, and compute resources.
- Key Vault secrets and vaults.
- RBAC policy assignments and role assignments.

None of the user Infrastructure as a Service (IaaS) or Platform as a Service (PaaS) resources are recovered during deployment. These losses include IaaS VMs, storage accounts, blobs, tables, network configuration, and so on. The purpose of cloud recovery is to ensure your operators and users can sign back in to the portal after deployment is complete. Users signing back in won't see any of their resources. Users have their subscriptions restored and along with that the original plans, offers, and policies defined by the admin. Users signing back in to the system operate under the same constraints imposed by the original solution before the disaster. After cloud recovery completes, the operator can manually restore value-add and third-party RPs and associated data.

## Next steps

Learn about the best practices for [using the Infrastructure Backup Service](#).

# Infrastructure Backup Service best practices

4 minutes to read • [Edit Online](#)

Follow these best practices when you deploy and manage Azure Stack Hub to help mitigate data loss if there's a catastrophic failure.

Review the best practices regularly to verify that your installation is still in compliance when changes are made to the operation flow. If you come across any issues while implementing these best practices, contact Microsoft Support for help.

## Configuration best practices

### Deployment

Enable Infrastructure Backup after deployment of each Azure Stack Hub Cloud. Using Azure Stack Hub PowerShell, you can schedule backups from any client/server with access to the operator management API endpoint.

### Networking

The Universal Naming Convention (UNC) string for the path must use a fully qualified domain name (FQDN). IP address can be used if name resolution isn't possible. A UNC string specifies the location of resources such as shared files or devices.

### Encryption

#### Version 1901 and newer

The encryption certificate is used to encrypt backup data that gets exported to external storage. The certificate can be a self-signed certificate since the certificate is only used to transport keys. Refer to [New-SelfSignedCertificate](#) for more info on how to create a certificate.

The key must be stored in a secure location (for example, global Azure Key Vault certificate). The CER format of the certificate is used to encrypt data. The PFX format must be used during cloud recovery deployment of Azure Stack Hub to decrypt backup data.

### Create a certificate



#### Method of Certificate Creation

Import

\* Certificate Name ⓘ

AzSIBCCert

\* Upload Certificate File

"AzSIBCCert\_Vault.pfx"



Password

.....

#### 1811 and older

The encryption key is used to encrypt backup data that gets exported to external storage. The key is generated as part of [enabling backup for Azure Stack Hub with PowerShell](#).

The key must be stored in a secure location (for example, global Azure Key Vault secret). This key must be used during redeployment of Azure Stack Hub.

Secret

Content type (optional)

base64

Show secret value

## Operational best practices

### Backups

- Backup jobs execute while the system is running so there's no downtime to the management experiences or user apps. Expect the backup jobs to take 20-40 minutes for a solution that's under reasonable load.
- Using OEM provided instructions, manually backed up network switches and the hardware lifecycle host (HLH) should be stored on the same backup share where the Infrastructure Backup Controller stores control plane backup data. Consider storing switch and HLH configurations in the region folder. If you have multiple Azure Stack Hub instances in the same region, consider using an identifier for each configuration that belongs to a scale unit.

### Folder Names

- Infrastructure creates MASBACKUP folder automatically. This is a Microsoft-managed share. You can create shares at the same level as MASBACKUP. It's not recommended to create folders or storage data inside of MASBACKUP that Azure Stack Hub doesn't create.
- User FQDN and region in your folder name to differentiate backup data from different clouds. The FQDN of your Azure Stack Hub deployment and endpoints is the combination of the Region parameter and the External Domain Name parameter. For more info, see [Azure Stack Hub datacenter integration - DNS](#).

For example, the backup share is AzS Backups hosted on fileserver01.contoso.com. In that file share there may be a folder per Azure Stack Hub deployment using the external domain name and a subfolder that uses the region name.

FQDN: contoso.com

Region: nyc

```
\fileserver01.contoso.com\AzSBackups
\fileserver01.contoso.com\AzSBackups\contoso.com
\fileserver01.contoso.com\AzSBackups\contoso.com\nyc
\fileserver01.contoso.com\AzSBackups\contoso.com\nyc\MASBackup
```

MASBackup folder is where Azure Stack Hub stores its backup data. Don't use this folder to store your own data. OEMs shouldn't use this folder to store any backup data either.

OEMs are encouraged to store backup data for their components under the region folder. Each network switch, hardware lifecycle host (HLH), and so on, may be stored in its own subfolder. For example:

```
\fileserver01.contoso.com\AzSBackups\contoso.com\nyc\HLH
\fileserver01.contoso.com\AzSBackups\contoso.com\nyc\Switches
\fileserver01.contoso.com\AzSBackups\contoso.com\nyc\DeploymentData
\fileserver01.contoso.com\AzSBackups\contoso.com\nyc\Registration
```

### Monitoring

The following alerts are supported by the system:

| ALERT                                                    | DESCRIPTION                                                                                    | REMEDIATION                                                                                                                               |
|----------------------------------------------------------|------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| Backup failed because the file share is out of capacity. | File share is out of capacity and backup controller can't export backup files to the location. | Add more storage capacity and try back up again. Delete existing backups (starting from oldest first) to free up space.                   |
| Backup failed due to connectivity problems.              | Network between Azure Stack Hub and the file share is experiencing issues.                     | Address the network issue and try backup again.                                                                                           |
| Backup failed due to a fault in the path.                | The file share path can't be resolved.                                                         | Map the share from a different computer to ensure the share is accessible. You may need to update the path if it's no longer valid.       |
| Backup failed due to authentication issue.               | There might be an issue with the credentials or a network issue that impacts authentication.   | Map the share from a different computer to ensure the share is accessible. You may need to update credentials if they're no longer valid. |
| Backup failed due to a general fault.                    | The failed request could be due to an intermittent issue. Try to back up again.                | Call support.                                                                                                                             |

## Next steps

Review the reference material for the [Infrastructure Backup Service](#).

Enable the [Infrastructure Backup Service](#).

# Infrastructure Backup Service reference

6 minutes to read • [Edit Online](#)

## Azure backup infrastructure

Azure Stack Hub consists of many services that comprise the portal (Azure Resource Manager) and the overall infrastructure management experience. The app-like management experience of Azure Stack Hub focuses on reducing the complexity exposed to the operator of the solution.

Infrastructure Backup Service is designed to internalize the complexity of backing up and restoring data for infrastructure services, ensuring operators can focus on managing the solution and maintaining an SLA to users.

Exporting the backup data to an external share is required to avoid storing backups on the same system. Requiring an external share gives the admin the flexibility to determine where to store the data based on existing company BC/DR policies.

### Infrastructure Backup Service components

Infrastructure Backup Service includes the following components:

- **Infrastructure Backup Controller**

The Infrastructure Backup Controller is instantiated with and resides in every Azure Stack Hub Cloud.

- **Backup Resource Provider**

The Backup Resource Provider (Backup RP) is composed of the user interface and APIs exposing basic backup functionality for Azure Stack Hub infrastructure.

#### Infrastructure Backup Controller

The Infrastructure Backup Controller is a Service Fabric service that gets instantiated for an Azure Stack Hub Cloud. Backup resources are created at a regional level and capture region-specific service data from AD, CA, Azure Resource Manager, CRP, SRP, NRP, Key Vault, RBAC.

#### Backup Resource Provider

The Backup Resource Provider presents a user interface in the Azure Stack Hub portal for basic configuration and listing of backup resources. Operators can do the following actions in the user interface:

- Enable backup for the first time by providing external storage location, credentials, and encryption key.
- View completed created backup resources and status resources under creation.
- Modify the storage location where Backup Controller places backup data.
- Modify the credentials that Backup Controller uses to access external storage location.
- Modify the encryption key that Backup Controller uses to encrypt backups.

## Backup Controller requirements

This section describes the important requirements for Infrastructure Backup Service. We recommend you review the info carefully before you enable backup for your Azure Stack Hub instance, and then refer back to it as necessary during deployment and subsequent operation.

The requirements include:

- **Software requirements** - describes supported storage locations and sizing guidance.
- **Network requirements** - describes network requirements for different storage locations.

### Software requirements

## Supported storage locations

| Storage location                                                                  | Details                                                                                                                                                          |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SMB file share hosted on a storage device within the trusted network environment. | SMB share in the same datacenter where Azure Stack Hub is deployed or in a different datacenter. Multiple Azure Stack Hub instances can use the same file share. |
| SMB file share on Azure.                                                          | Not currently supported.                                                                                                                                         |
| Blob storage on Azure.                                                            | Not currently supported.                                                                                                                                         |

## Supported SMB versions

| SMB | Version |
|-----|---------|
| SMB | 3.x     |

## SMB encryption

### 1907 and beyond

Infrastructure Backup Service supports transferring backup data to an external storage location with SMB encryption enabled on the server side. If the server doesn't support SMB Encryption or doesn't have the feature enabled, Infrastructure Backup Service will fall back to unencrypted data transfer. Backup data placed on the external storage location is always encrypted at rest and isn't dependent on SMB encryption.

## Storage location sizing

We recommend you back up at least two times a day and keep at most seven days of backups. This is the default behavior when you enable infrastructure backups on Azure Stack Hub.

### 1907 and beyond

#### *System connected to Azure AD identity provider*

| Environment scale | Projected size of backup | Total amount of space required |
|-------------------|--------------------------|--------------------------------|
| 4-16 nodes/ASDK   | 1 GB                     | 20 GB                          |

#### *System connected to corporate AD identity provider via ADFS*

| Environment scale | Projected size of backup | Total amount of space required |
|-------------------|--------------------------|--------------------------------|
| 4-16 nodes        | 20 GB                    | 280 GB                         |
| ASDK              | 10 GB                    | 140 GB                         |

## Pre-1907

| Environment scale | Projected size of backup | Total amount of space required |
|-------------------|--------------------------|--------------------------------|
| 4-16 nodes        | 20 GB                    | 280 GB                         |
| ASDK              | 10 GB                    | 140 GB                         |

## Network requirements

| Storage Location                                                                  | Details                                                                                                                                                                                     |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SMB file share hosted on a storage device within the trusted network environment. | Port 445 is required if the Azure Stack Hub instance resides in a firewalled environment. Infrastructure Backup Controller will initiate a connection to the SMB file server over port 445. |
| To use FQDN of file server, the name must be resolvable from the PEP.             |                                                                                                                                                                                             |

#### NOTE

No inbound ports need to be opened.

## Encryption Requirements

Starting in 1901, the Infrastructure Backup Service will use a certificate with a public key (.CER) to encrypt backup data and a certificate with the private key (.PFX) to decrypt backup data during cloud recovery.

- The certificate is used for transport of keys and isn't used to establish secure authenticated communication. For this reason, the certificate can be a self-signed certificate. Azure Stack Hub doesn't need to verify root or trust for this certificate so external internet access isn't required.

The self-signed certificate comes in two parts, one with the public key and one with the private key:

- Encrypt backup data: Certificate with the public key (exported to .CER file) is used to encrypt backup data.
- Decrypt backup data: Certificate with the private key (exported to .PFX file) is used to decrypt backup data.

The certificate with the public key (.CER) isn't managed by internal secret rotation. To rotate the certificate, you need to create a new self-signed certificate and update backup settings with the new file (.CER).

- All existing backups remain encrypted using the previous public key. New backups use the new public key.

The certificate used during cloud recovery with the private key (.PFX) is not persisted by Azure Stack Hub for security reasons. This file will need to be provided explicitly during cloud recovery.

**Backwards compatibility mode** Starting in 1901, encryption key support is deprecated and will be removed in a future release. If you updated from 1811 with backup already enabled using an encryption key, Azure Stack Hub will continue to use the encryption key. Backwards compatibility mode will be supported for at least three releases. After that time, a certificate will be required.

- Updating from encryption key to certificate is a one-way operation.
- All existing backups will remain encrypted using the encryption key. New backups will use the certificate.

## Infrastructure Backup Limits

Consider these limits as you plan, deploy, and operate your Microsoft Azure Stack Hub instances. The following table describes these limits.

### Infrastructure Backup limits

| Limit Identifier | Limit     | Comments                                                                                           |
|------------------|-----------|----------------------------------------------------------------------------------------------------|
| Backup type      | Full only | Infrastructure Backup Controller only supports full backups. Incremental backups aren't supported. |

| LIMIT IDENTIFIER                                                 | LIMIT                | COMMENTS                                                                                                                                 |
|------------------------------------------------------------------|----------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| Scheduled backups                                                | Scheduled and manual | Backup controller supports scheduled and on-demand backups.                                                                              |
| Maximum concurrent backup jobs                                   | 1                    | Only one active backup job is supported per instance of Backup Controller.                                                               |
| Network switch configuration                                     | Not in scope         | Admin must back up network switch configuration using OEM tools. Refer to documentation for Azure Stack Hub provided by each OEM vendor. |
| Hardware Lifecycle Host                                          | Not in scope         | Admin must back up Hardware Lifecycle Host using OEM tools. Refer to documentation for Azure Stack Hub provided by each OEM vendor.      |
| Maximum number of file shares                                    | 1                    | Only one file share can be used to store backup data.                                                                                    |
| Backup App Services, Function, SQL, mysql resource provider data | Not in scope         | Refer to guidance published for deploying and managing value-add RPs created by Microsoft.                                               |
| Backup third-party resource providers                            | Not in scope         | Refer to guidance published for deploying and managing value-add RPs created by third-party vendors.                                     |

## Next steps

- To learn more about the Infrastructure Backup Service, see [Backup and data recovery for Azure Stack Hub with the Infrastructure Backup Service](#).

# Microsoft Azure Stack Hub help and support

3 minutes to read • [Edit Online](#)

**Help + support** in the Azure Stack Hub portal has resources to help operators learn more about Azure Stack Hub, check their support options, and get expert help. Beginning with the 1907 release, operators can also use Help + support to collect diagnostic logs for troubleshooting.

## Help resources

Operators can also use **Help + support** to learn more about Azure Stack Hub, check their support options, and get expert help.

### Things to try first

At the top of **Help + support** are links to things you might try first, like read up about a new concept, understand how billing works, or see which support options are available.

#### Have you tried one of these?

|                                                                   |                                                                                     |                                                                |
|-------------------------------------------------------------------|-------------------------------------------------------------------------------------|----------------------------------------------------------------|
| <b>Documentation</b><br>Azure Stack tutorials and how-to articles | <b>Learn about billing</b><br>Tips for monitoring usage and understanding your bill | <b>Support options</b><br>Learn how to get Azure Stack support |
|-------------------------------------------------------------------|-------------------------------------------------------------------------------------|----------------------------------------------------------------|

- **Documentation.** [Azure Stack Hub Operator Documentation](#) includes concepts, how-to topics and tutorials that show how to offer Azure Stack Hub services such as virtual machines, SQL databases, web apps, and more.
- **Learn about billing.** Get tips on [usage and billing](#).
- **Support options.** Azure Stack Hub operators can choose from a range of [Azure support options](#) that can fit the needs of any enterprise.

### Get expert help

For an integrated system, there is a coordinated escalation and resolution process between Microsoft and our original equipment manufacturer (OEM) hardware partners.

If there is a cloud services issue, support is offered through Microsoft Customer Support Services (CSS). You can click **Help** (question mark) in the upper-right corner of the administrator portal and then click **Help + support** to open **Help + Support Overview** and submit a new support request. Creating a support request will preselect Azure Stack Hub service. We highly recommend that customers use this experience to submit tickets rather than using the public Azure portal.

If there is an issue with deployment, patch and update, hardware (including field replaceable units), and any hardware-branded software, such as software running on the hardware lifecycle host, contact your OEM hardware vendor first. For anything else, contact Microsoft CSS.

## Support



### Support requests

Quickly connect with our problem solving experts

[New support request](#)

For the ASDK, you can ask support-related questions in the [Azure Stack Hub MSDN Forum](#).

You can click **Help** (question mark) in the upper-right corner of the administrator portal and then click **Help + support** to open **Help + Support Overview**, which has a link to the forum. MSDN forums are regularly monitored.

Because the development kit is an evaluation environment, there is no official support offered through Microsoft CSS.

## Support



### ASDK support

Quickly connect with our problem solving experts

[MSDN Forums](#)

You can also reach out to the MSDN Forums to discuss an issue, or take online training and improve your own skills.

## Community



### MSDN Forums

Information and discussion by Microsoft and community

[MSDN Forums](#)

## Learning



### Online course

Learn about configuring and operating Azure Stack

[Go to course](#)

## Get up to speed with Azure Stack Hub

This set of tutorials is customized depending on whether you're running the ASDK or integrated systems so you can quickly get up to speed with your environment.

### Tutorials

[Getting started with Azure Stack...](#)

[Learn how to manage offerings...](#)

[Learn how to manage updates...](#)

[Using the admin portal](#)

[Offering IaaS and PaaS services](#)

[Servicing policy and release notes](#)

[Install PowerShell for Azure Stack](#)

[Plans, offers, quotas, and subscriptions](#)

[Manage updates in admin portal](#)

[Renew or change registration](#)

[Available Azure marketplace items](#)

[Apply updates in Azure Stack](#)

[Register your ASDK](#)

[Download marketplace items](#)

[Add custom VM images](#)

## Diagnostic log collection

Beginning with the 1907 release, there are two new ways to collect logs in **Help and support**:

- **Automatic collection:** If enabled, log collection is triggered by specific health alerts
- **Collect logs now:** You can choose a 1-4 hour sliding window from the last seven days

Dashboard > Overview - Log Collection

Overview - Log Collection

Search (Ctrl+)

Collect logs now Automatic collection settings

There are one or more log collection alerts. View alerts →

Automatic log collection (change)  
Disabled

Storage account (change)  
satesu100

Collection Time  
Last 7 days

Learn more  
Log collection  
Manage Automatic Log Collection Storage Account

1 items

| COLLECTION TIME | TYPE | STATUS | FROM DATE | TO DATE | LOG UPLOAD SIZE |
|-----------------|------|--------|-----------|---------|-----------------|
| Last 7 days     |      |        |           |         |                 |

Integrated systems can share the diagnostic logs with Microsoft Customer Support Service (CSS). Because Azure Stack Development Kit (ASDK) is an evaluation environment, it is not supported by CSS. For more information, see [Azure Stack Hub diagnostic log collection overview](#).

## Help and support for earlier releases Azure Stack Hub (pre-1905)

Previous Azure Stack Hub releases also have a link to **Help + support** that redirects to the [Azure Stack Hub Operator Documentation](#).

CloudAdmin@azurest...

Help

Help + support

Azure roadmap

Keyboard shortcuts

Show diagnostics

New support request

If there is a cloud services issue, support is offered through Microsoft Customer Support Services (CSS). You can click **Help** (question mark) in the upper-right corner of the administrator portal, click **Help and Support**, and then click **New support request** to directly submit a new support request with CSS.

For an integrated system, there is a coordinated escalation and resolution process between Microsoft and our OEM partners. If there is a cloud services issue, support is offered through Microsoft CSS.

If there is an issue with deployment, patch and update, hardware (including field replaceable units), and any hardware-branded software, such as software running on the hardware lifecycle host, contact your OEM hardware vendor first. For anything else, contact Microsoft CSS.

For the development kit, you can ask support-related questions in the [Azure Stack Hub MSDN Forum](#). You can click **Help** (question mark) in the upper-right corner of the administrator portal and then click **New support request** to get help from experts in the Azure Stack Hub community. Because the development kit is an evaluation environment, there is no official support offered through Microsoft CSS.

## Next steps

- Learn about the [Troubleshooting Azure Stack Hub](#)

# Overview of Azure Stack Hub diagnostic log collection

2 minutes to read • [Edit Online](#)

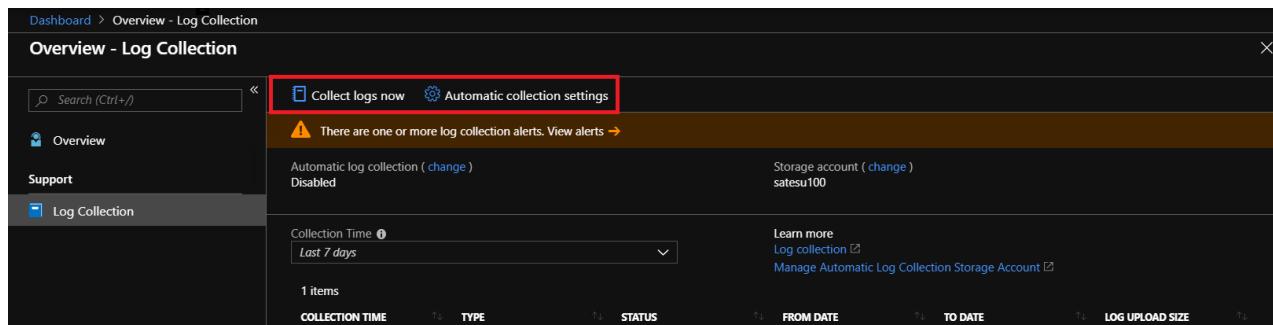
Azure Stack Hub is a large collection of components working together and interacting with each other. All these components generate their own unique logs. This can make diagnosing issues a challenging task, especially for errors coming from multiple, interacting Azure Stack Hub components. In order to address this challenge, we have designed a diagnostic log collection experience.

Prior to 1907, the diagnostic experience included using [Test-AzureStack](#) to validate system health and using the [privileged endpoint \(PEP\)](#) to collect logs for troubleshooting.

Beginning with the 1907 release, the **Help and Support** page adds a simpler experience using **Diagnostic log collection**. **Diagnostic log collection** is part of an ongoing investment to improve Azure Stack Hub operator's experience with the troubleshooting process. With these improvements, operators can quickly collect and share diagnostic logs with Microsoft Customer Support Services (CSS). The logs can be stored in a blob container in Azure, where access can be customized as needed.

**Diagnostic log collection** works in two different ways:

- **Automatic collection:** If enabled (recommended), log collection is automatically triggered by specific health alerts and stored in your Azure storage account
- **Collect logs now:** This is an on-demand option where you can choose to collect logs from a 1-4 hour sliding window from the last seven days



**Diagnostic log collection** has an easy user interface and doesn't require PowerShell. Logs get collected reliably even if infrastructure services are down. If your policy allows sharing diagnostic logs with CSS, **Diagnostic log collection** is the recommended collection method beginning with the 1907 release. You should only use [the PEP](#) to collect logs if **Diagnostic log collection** in Help and Support is unavailable.

## Automatic diagnostic log collection

When a [specific health alert](#) is active, automatic diagnostic log collection starts and proactively uploads diagnostic logs from Azure Stack Hub to a storage blob in Azure, significantly reducing the time required to share diagnostic logs with CSS. Diagnostic logs are only collected when an alert is raised.

For more information about automatic log collection, see [Configure automatic Azure Stack Hub diagnostic log collection](#).

## On-demand diagnostic log collection

With on-demand collection, diagnostic logs are uploaded from Azure Stack Hub to a storage blob in Azure when

an Azure Stack Hub operator manually triggers the collection. CSS will provide shared access signature (SAS) URL to a CSS-owned storage blob. An Azure Stack Hub operator can click **Collect logs now** and enter the SAS URL. Diagnostic logs will then get uploaded directly to the CSS blob without needing an intermediate share.

For more information about collecting logs on demand, see [Collect Azure Stack Hub diagnostic logs now](#).

## Bandwidth considerations

The average size of diagnostic log collection varies based on whether it runs on-demand or automatic. The average size for automatic log collection is around 2 GB, whereas on-demand log collection size depends on how many hours are being collected.

The following table lists considerations for environments with limited or metered connections to Azure.

| NETWORK CONNECTION                    | IMPACT                                                                             |
|---------------------------------------|------------------------------------------------------------------------------------|
| Low-bandwidth/high-latency connection | Log upload will take an extended amount of time to complete                        |
| Shared connection                     | The upload may also impact other applications/users sharing the network connection |
| Metered connection                    | There may be an additional charge from your ISP for the additional network usage   |

For more information, see [Best practices for automatic Azure Stack Hub log collection](#).

## See also

[Azure Stack Hub log and customer data handling](#)

[Using shared access signatures \(SAS\)](#)

[Best practices for automatic Azure Stack Hub log collection](#)

# Configure automatic Azure Stack Hub diagnostic log collection

4 minutes to read • [Edit Online](#)

We recommend configuring the automatic diagnostic log collection feature to streamline your log collection and customer support experience. If system health conditions need to be investigated, the logs can be uploaded automatically for analysis by Microsoft Customer Support Services (CSS).

## Create an Azure blob container SAS URL

Before you can configure automatic log collection, you'll need to get a shared access signature (SAS) for a blob container. A SAS lets you grant access to resources in your storage account without sharing your account keys. You can save Azure Stack Hub log files to a blob container in Azure, and then provide the SAS URL where CSS can collect the logs.

### Prerequisites

You can use a new or existing blob container in Azure. To create a blob container in Azure, you need at least the [storage blob contributor role](#) or the [specific permission](#). Global administrators also have the necessary permission.

For best practices about choosing parameters for the automatic log collection storage account, see [Best practices for automatic Azure Stack Hub log collection](#). For more information about types of storage accounts, see [Azure storage account overview](#)

### Create a blob storage account

1. Sign in to the [Azure portal](#).
2. Click **Storage accounts > Add**.
3. Create a blob container with these settings:
  - **Subscription:** Choose your Azure subscription
  - **Resource group:** Specify a resource group
  - **Storage account name:** Specify a unique storage account name
  - **Location:** Choose a datacenter in accordance with your company policy
  - **Performance:** Choose Standard
  - **Account kind** Choose StorageV2 (general purpose v2)
  - **Replication:** Choose Locally-redundant storage (LRS)
  - **Access tier:** Choose Cool

## Create storage account

Tables. The cost of your storage account depends on the usage and the options you choose below. [Learn more](#)

### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

|                    |                            |
|--------------------|----------------------------|
| * Subscription     | <input type="text"/>       |
| └ * Resource group | <input type="text"/>       |
|                    | <a href="#">Create new</a> |

### Instance details

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. [Choose classic deployment model](#)

|                                         |                                                                                                   |
|-----------------------------------------|---------------------------------------------------------------------------------------------------|
| * Storage account name <small>i</small> | <input type="text" value="diagnosticlogcollection"/> <span style="color: green;">✓</span>         |
| * Location                              | <input type="text" value="(US) West US"/> <span style="color: green;">✓</span>                    |
| Performance <small>i</small>            | <input checked="" type="radio"/> Standard <input type="radio"/> Premium                           |
| Account kind <small>i</small>           | <input type="text" value="StorageV2 (general purpose v2)"/> <span style="color: green;">✓</span>  |
| Replication <small>i</small>            | <input type="text" value="Locally-redundant storage (LRS)"/> <span style="color: green;">✓</span> |
| Access tier (default) <small>i</small>  | <input checked="" type="radio"/> Cool <input type="radio"/> Hot                                   |

[Review + create](#)

[< Previous](#)

[Next : Advanced >](#)

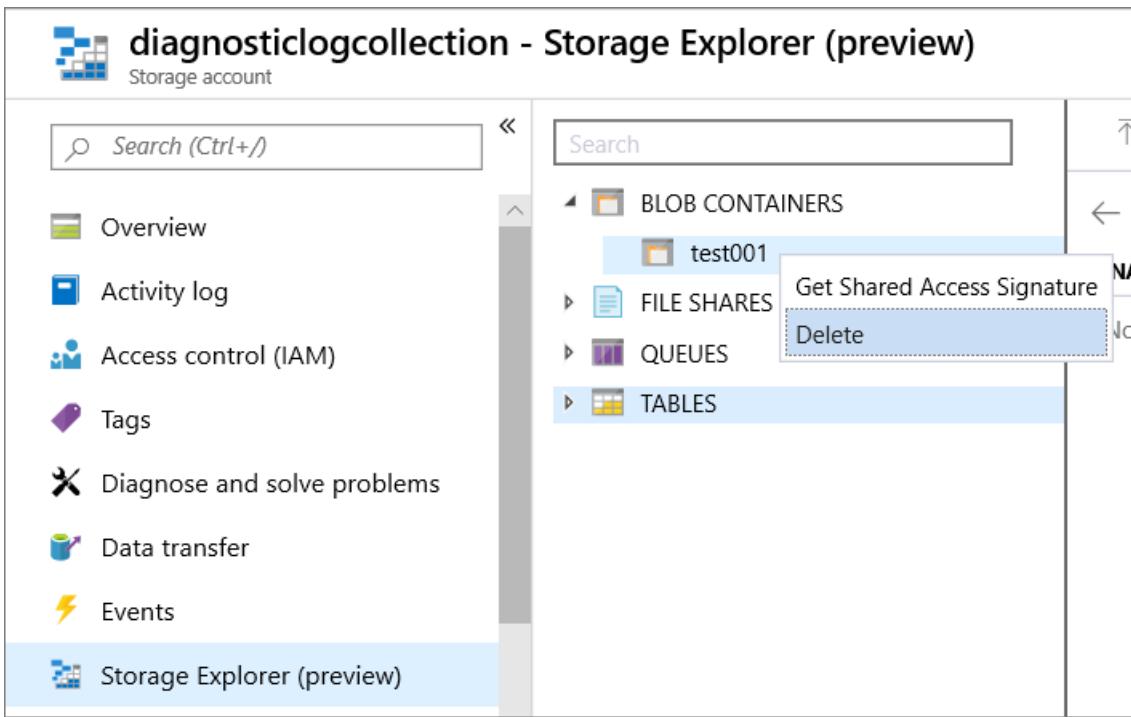
4. Click **Review + create** and then click **Create**.

### Create a blob container

1. After the deployment succeeds, click **Go to resource**. You can also pin the storage account to the Dashboard for easy access.
2. Click **Storage Explorer (preview)**, right-click **Blob containers**, and click **Create blob container**.
3. Enter a name for the new container and click **OK**.

## Create a SAS URL

1. Right-click the container, click **Get Shared Access Signature**.



2. Choose these properties:

- Start time: You can optionally move the start time back
- Expiry time: Two years
- Time zone: UTC
- Permissions: Read, Write, and List

### Shared Access Signature

Access policy: (none)

Start time: 7/24/2019 6:34 PM

Expiry time: 7/24/2021 6:34 PM

Time zone:  
 Local  
 UTC

Permissions:

Add  
 Create  
 Write  
 Delete  
 List

3. Click **Create**.

Copy the URL and enter it when you [configure automatic log collection](#). For more information about SAS URLs, see [Using shared access signatures \(SAS\)](#).

## Steps to configure automatic log collection

Follow these steps to add the SAS URL to the log collection UI:

1. Sign in to the Azure Stack Hub administrator portal.

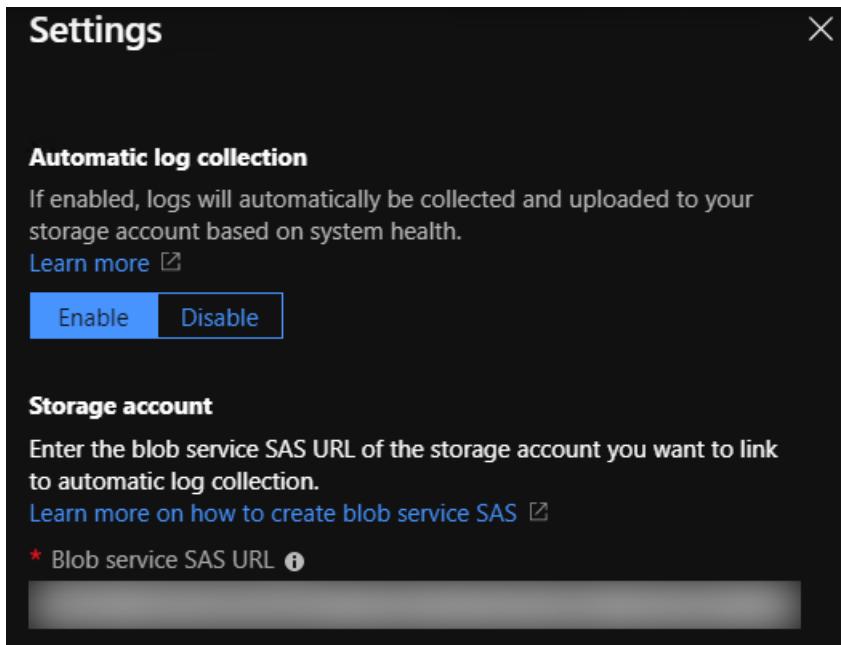
2. Open **Help and support Overview**.

3. Click **Automatic collection settings**.



4. Set Automatic log collection to **Enabled**.

5. Enter the shared access signature (SAS) URL of the storage account blob container.



#### NOTE

Automatic log collection can be disabled and re-enabled anytime. The SAS URL configuration won't change. If automatic log collection is re-enabled, the previously entered SAS URL will undergo the same validation checks, and an expired SAS URL will be rejected.

## View log collection

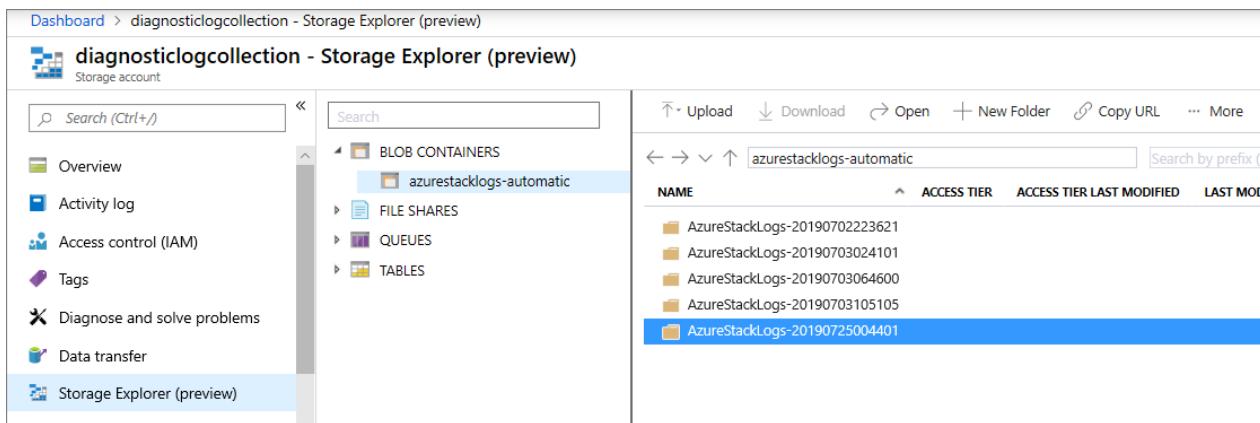
The history of logs collected from Azure Stack Hub appears on the **Log collection** page in Help and Support, with the following dates and times:

- **Collection time:** When the log collection operation began
- **From Date:** Start of the time period for which you want to collect
- **To Date:** End of the time period

| Automatic log collection ( <a href="#">change</a> )<br>Enabled                                                                                                    | Storage account ( <a href="#">change</a> )<br><a href="#">diagnosticlogcollection</a> |                                                |                          |                          |                 |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|------------------------------------------------|--------------------------|--------------------------|-----------------|
| Collection Time <a href="#">(1)</a><br><input type="button" value="Last day"/> <input type="button" value="Last week"/> <input type="button" value="Last month"/> | <a href="#">Learn more</a><br><a href="#">Log collection overview</a>                 |                                                |                          |                          |                 |
| 1 items                                                                                                                                                           |                                                                                       |                                                |                          |                          |                 |
| COLLECTION TIME                                                                                                                                                   | TYPE                                                                                  | STATUS                                         | FROM DATE                | TO DATE                  | LOG UPLOAD SIZE |
| 7/25/2019, 12:22:06 A...                                                                                                                                          | Automatic                                                                             | <span style="color: green;">✔ Succeeded</span> | 7/24/2019, 10:47:03 P... | 7/25/2019, 12:22:06 A... | 2.46 GB         |

If diagnostic log collection fails, verify the SAS URL is valid. If failure persists or you see multiple failures, call Microsoft CSS for help.

Operators can also check the storage account for automatically collected logs. For example, this screenshot shows log collections by using the Storage Explorer preview from the Azure portal:



The screenshot shows the Azure Storage Explorer (preview) interface. The left sidebar has links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Data transfer, and Storage Explorer (preview). The main area shows a blob container named 'azurestacklogs-automatic' under 'BLOB CONTAINERS'. Inside this container are four files: 'AzureStackLogs-20190702223621', 'AzureStackLogs-20190703024101', 'AzureStackLogs-20190703064600', and 'AzureStackLogs-20190703105105'. A fifth file, 'AzureStackLogs-20190725004401', is selected and highlighted in blue. The top navigation bar includes 'Upload', 'Download', 'Open', 'New Folder', 'Copy URL', and 'More'.

## Automatic diagnostic log collection alerts

If enabled, automatic diagnostic log collection occurs only when necessary. Only the alerts in the following table trigger collection.

For example, **Update failed** is an alert that triggers automatic diagnostic log collection. If automatic collection is enabled, diagnostic logs will be proactively captured during an update failure to help CSS troubleshoot the problem. The diagnostic logs are only collected when the alert for **Update failed** is raised.

| ALERT TITLE                                                         | FAULTIDTYPE                                                       |
|---------------------------------------------------------------------|-------------------------------------------------------------------|
| Unable to connect to the remote service                             | UsageBridge.NetworkError                                          |
| Update failed                                                       | Urp.UpdateFailure                                                 |
| Storage Resource Provider infrastructure/dependencies not available | StorageResourceProviderDependencyUnavailable                      |
| Node not connected to controller                                    | ServerHostNotConnectedToController                                |
| Route publication failure                                           | SlbMuxRoutePublicationFailure                                     |
| Storage Resource Provider internal data store unavailable           | StorageResourceProvider. DataStoreConnectionFail                  |
| Storage device failure                                              | Microsoft.Health.FaultType.VirtualDisks.Detached                  |
| Health controller cannot access storage account                     | Microsoft.Health.FaultType.StorageError                           |
| Connectivity to a physical disk has been lost                       | Microsoft.Health.FaultType.PhysicalDisk.LostCommunication         |
| The blob service isn't running on a node                            | StorageService.The.blob.service.is.not.running.on.a.node-Critical |
| Infrastructure role unhealthy                                       | Microsoft.Health.FaultType.GenericExceptionFault                  |
| Table service errors                                                | StorageService.Table.service.errors-Critical                      |

| ALERT TITLE                                                                                 | FAULTIDTYPE                                                       |
|---------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| A file share is over 80% utilized                                                           | Microsoft.Health.FaultType.FileShare.Capacity.Warning.Infra       |
| Scale unit node is offline                                                                  | FRP.Heartbeat.PhysicalNode                                        |
| Infrastructure role instance unavailable                                                    | FRP.Heartbeat.InfraVM                                             |
| Infrastructure role instance unavailable                                                    | FRP.Heartbeat.NonHaVm                                             |
| The infrastructure role, Directory Management, has reported time synchronization errors     | DirectoryServiceTimeSynchronizationError                          |
| Pending external certificate expiration                                                     | CertificateExpiration.ExternalCert.Warning                        |
| Pending external certificate expiration                                                     | CertificateExpiration.ExternalCert.Critical                       |
| Unable to provision virtual machines for specific class and size due to low memory capacity | AzureStack.ComputeController.VmCreationFailure.LowMemory          |
| Node inaccessible for virtual machine placement                                             | AzureStack.ComputeController.HostUnresponsive                     |
| Backup failed                                                                               | AzureStack.BackupController.BackupFailedGeneralFault              |
| The scheduled backup was skipped due to a conflict with failed operations                   | AzureStack.BackupController.BackupSkippedWithFailedOperationFault |

## See also

[Azure Stack Hub log and customer data handling](#)

[Using shared access signatures \(SAS\)](#)

[Best practices for automatic Azure Stack Hub log collection](#)

# Best practices for automatic Azure Stack Hub log collection

2 minutes to read • [Edit Online](#)

This topic covers best practices for managing automatic diagnostic log collection for Azure Stack Hub.

## Collecting logs from multiple Azure Stack Hub systems

Set up one blob container for every Azure Stack Hub scale unit you want to collect logs from. For more information about how to configure the blob container, see [Configure automatic Azure Stack Hub diagnostic log collection](#). As a best practice, only save diagnostic logs from the same Azure Stack Hub scale unit within a single blob container.

## Retention policy

Create an Azure Blob storage [lifecycle management rule](#) to manage the log retention policy. We suggest retaining diagnostic logs for 30 days. To create a lifecycle management rule in Azure storage, sign in to the Azure portal, click **Storage accounts**, click the blob container, and under **Blob service**, click **Lifecycle Management**.

The screenshot shows the Azure portal interface. On the left, there's a dark sidebar with various navigation options like 'Create a resource', 'Home', 'Dashboard', etc. The main area shows a 'Storage accounts' blade for an account named 'akazslogs'. Under 'Blob service', the 'Lifecycle Management' option is highlighted with a red box. The right side shows a 'Lifecycle Management' page with tabs for 'List view' and 'Code view'. It includes a note about GPv2 tiers and a table with columns for 'NAME' and 'STATUS'. The table currently shows 'No results'.

## SAS token expiration

Set the SAS URL expiry to two years. If you ever renew your storage account keys, make sure to regenerate the SAS URL. You should manage the SAS token according to best practices. For more information, see [Best practices when using SAS](#).

## Bandwidth consumption

The average size of diagnostic log collection varies based on whether log collection is on-demand or automatic.

For on-demand log collection, the size of the logs collection depends on how many hours are being collected. You can choose any 1-4 hour sliding window from the last seven days.

When automatic diagnostic log collection is enabled, the service monitors for critical alerts. After a critical alert gets raised and persists for around 30 minutes, the service collects and uploads appropriate logs. This log collection size is around 2 GB on average. In the case of a patch and update failure, automatic log collection will start only if a critical alert is raised and persists for around 30 minutes. We recommend that you follow [guidance on monitoring the patch and update](#). Alert monitoring, log collection, and upload are transparent to the user.

In a healthy system, logs will not be collected at all. In an unhealthy system, log collection may run two or three times in a day, but typically only once. At most, it could potentially run up to ten times in a day in a worst-case scenario.

The following table can help environments with limited or metered connections to Azure consider the impact of enabling automatic log collection.

| NETWORK CONNECTION                    | IMPACT                                                                             |
|---------------------------------------|------------------------------------------------------------------------------------|
| Low-bandwidth/high-latency connection | Log upload will take an extended amount of time to complete                        |
| Shared connection                     | The upload may also impact other applications/users sharing the network connection |
| Metered connection                    | There may be an additional charge from your ISP for the additional network usage   |

## Managing costs

Azure [blob storage charges](#) depend on how much data is saved each month and other factors such as data redundancy. If you don't have an existing storage account, you can sign in to the Azure portal, click **Storage accounts**, and follow the steps to [create an Azure blob container SAS URL](#).

As a best practice, create an Azure Blob storage [lifecycle management policy](#) to minimize ongoing storage costs. For more information about how to set up the storage account, see [Configure automatic Azure Stack Hub diagnostic log collection](#)

## See also

[Configure automatic Azure Stack Hub log collection](#)

# Collect Azure Stack Hub diagnostic logs on demand

9 minutes to read • [Edit Online](#)

As part of troubleshooting, Microsoft Customer Support Services (CSS) may need to analyze diagnostic logs. Beginning with the 1907 release, Azure Stack Hub operators can upload diagnostic logs to a blob container in Azure by using **Help and Support**. Using **Help and Support** is recommended over the previous method of using PowerShell because it's simpler. But if the portal is unavailable, operators can continue to collect logs using **Get-AzureStackLog** through the privileged endpoint (PEP) as in previous releases. This topic covers both ways of collecting diagnostic logs on demand.

## NOTE

As an alternative to collecting logs on demand, you can streamline the troubleshooting process by enabling [automatic diagnostic log collection](#). If system health conditions need to be investigated, the logs are uploaded automatically for analysis by CSS.

## Use Help and Support to collect diagnostic logs on demand

To troubleshoot a problem, CSS might request an Azure Stack Hub operator to collect diagnostic logs on demand for a specific time window from the previous week. In that case, CSS will provide the operator with a SAS URL for uploading the collection. Use the following steps to configure on-demand log collection using the SAS URL from CSS:

1. Open **Help and Support Overview** and click **Collect logs now**.
2. Choose a 1-4 hour sliding window from the last seven days.
3. Choose the local time zone.
4. Enter the SAS URL that CSS provided.

## Collect logs now

For on-demand log collection, enter the blob service SAS URL of the storage account where you want to upload these logs.

\* Blob service SAS URL [?](#)

✓

Logs time range

- Last hour
- Last 2 hours
- Last 3 hours
- Last 4 hours
- Custom

Start

|            |                 |            |
|------------|-----------------|------------|
| 2019-06-25 | [Calendar icon] | 7:46:30 PM |
|------------|-----------------|------------|

End

|                                                                            |                 |            |
|----------------------------------------------------------------------------|-----------------|------------|
| 2019-06-25                                                                 | [Calendar icon] | 8:46:30 PM |
| (UTC+00:00) --- Current Time Zone --- <span style="float: right;">▼</span> |                 |            |

### NOTE

If automatic diagnostic log collection is enabled, **Help and Support** shows when log collection is in progress. If you click **Collect logs now** to collect logs from a specific time while automatic log collection is in progress, on-demand collection begins after automatic log collection is complete.

## Use the privileged endpoint (PEP) to collect diagnostic logs

### Run `Get-AzureStackLog` on Azure Stack Hub integrated systems

To run `Get-AzureStackLog` on an integrated system, you need to have access to the Privileged End Point (PEP). Here's an example script you can run using the PEP to collect logs on an integrated system:

```
$ipAddress = "<IP ADDRESS OF THE PEP VM>" # You can also use the machine name instead of IP here.

$password = ConvertTo-SecureString "<CLOUD ADMIN PASSWORD>" -AsPlainText -Force
$cred = New-Object -TypeName System.Management.Automation.PSCredential ("<DOMAIN NAME>\CloudAdmin", $password)

$shareCred = Get-Credential

$session = New-PSSession -ComputerName $ipAddress -ConfigurationName PrivilegedEndpoint -Credential $cred

$fromDate = (Get-Date).AddHours(-8)
$toDate = (Get-Date).AddHours(-2) # Provide the time that includes the period for your issue

Invoke-Command -Session $session { Get-AzureStackLog -OutputSharePath "<EXTERNAL SHARE ADDRESS>" -
OutputShareCredential $using:shareCred -FilterByRole Storage -FromDate $using:fromDate -ToDate $using:toDate}

if ($session) {
 Remove-PSSession -Session $session
}
```

### Run `Get-AzureStackLog` on an Azure Stack Development Kit (ASDK) system

Use these steps to run `Get-AzureStackLog` on an ASDK host computer.

1. Sign in as **AzureStack\CloudAdmin** on the ASDK host computer.
2. Open a new PowerShell window as an administrator.
3. Run the **Get-AzureStackLog** PowerShell cmdlet.

#### Examples

- Collect all logs for all roles:

```
Get-AzureStackLog -OutputSharePath "<path>" -OutputShareCredential $cred
```

- Collect logs from VirtualMachines and BareMetal roles:

```
Get-AzureStackLog -OutputSharePath "<path>" -OutputShareCredential $cred -FilterByRole
VirtualMachines,BareMetal
```

- Collect logs from VirtualMachines and BareMetal roles, with date filtering for log files for the past 8 hours:

```
Get-AzureStackLog -OutputSharePath "<path>" -OutputShareCredential $cred -FilterByRole
VirtualMachines,BareMetal -FromDate (Get-Date).AddHours(-8)
```

- Collect logs from VirtualMachines and BareMetal roles, with date filtering for log files for the time period between 8 hours ago and 2 hours ago:

```
Get-AzureStackLog -OutputSharePath "<path>" -OutputShareCredential $cred -FilterByRole
VirtualMachines,BareMetal -FromDate (Get-Date).AddHours(-8) -ToDate (Get-Date).AddHours(-2)
```

- Collect logs from tenant deployments running self-managed Kubernetes clusters (AKS engine) on Azure Stack. Kubernetes logs should be stored in a tenant storage account in a format that will enable the collection time range to be applied to them as well.

```
Get-AzureStackLog -OutputPath <Path> -InputSasUri "<Blob Service Sas Uri>" -FromDate "<Beginning of the
time range>" -ToDate "<End of the time range>"
```

For example:

```
Get-AzureStackLog -OutputPath C:\KubernetesLogs -InputSasUri
"https://<storageAccountName>.blob.core.windows.net/<ContainerName><SAS token>" -FromDate (Get-
Date).AddHours(-8) -ToDate (Get-Date).AddHours(-2)
```

- Collect logs and store them in the specified Azure Storage blob container. The general syntax for this operation is as follows:

```
Get-AzureStackLog -OutputSasUri "<Blob service SAS Uri>"
```

For example:

```
Get-AzureStackLog -OutputSasUri "https://<storageAccountName>.blob.core.windows.net/<ContainerName><SAS
token>"
```

#### NOTE

This procedure is useful for uploading logs. Even if you don't have an SMB share accessible or internet access, you can create a blob storage account on your Azure Stack Hub to transfer the logs, and then use your client to retrieve those logs.

To generate the SAS token for the storage account, the following permissions are required:

- Access to the Blob Storage service.
- Access to the container resource type.

To generate a SAS Uri value to be used for the `-OutputSasUri` parameter, follow these steps:

1. Create a storage account, following the steps [in this article](#).
2. Open an instance of the Azure Storage Explorer.
3. Connect to the storage account created in step 1.
4. Navigate to **Blob Containers** in **Storage Services**.
5. Select **Create a new container**.
6. Right-click the new container, then click **Get Shared Access Signature**.
7. Select a valid **Start Time** and **End Time**, depending on your requirements.
8. For the required permissions, select **Read**, **Write**, and **List**.
9. Select **Create**.
10. You'll get a Shared Access Signature. Copy the URL portion and provide it to the `-OutputSasUri` parameter.

#### Parameter considerations for both ASDK and integrated systems

- The parameters **OutputSharePath** and **OutputShareCredential** are used to store logs in a user specified location.
- The **FromDate** and **ToDate** parameters can be used to collect logs for a particular time period. If these parameters aren't specified, logs are collected for the past four hours by default.
- Use the **FilterByNode** parameter to filter logs by computer name. For example:

```
Get-AzureStackLog -OutputSharePath "<path>" -OutputShareCredential $cred -FilterByNode azs-xrp01
```

- Use the **FilterByLogType** parameter to filter logs by type. You can choose to filter by File, Share, or WindowsEvent. For example:

```
Get-AzureStackLog -OutputSharePath "<path>" -OutputShareCredential $cred -FilterByLogType File
```

- You can use the **TimeOutInMinutes** parameter to set the timeout for log collection. It's set to 150 (2.5 hours) by default.
- Dump file log collection is disabled by default. To enable it, use the **IncludeDumpFile** switch parameter.
- Currently, you can use the **FilterByRole** parameter to filter log collection by the following roles:

| ACS     | CA           | HRP | OboService | VirtualMachines |
|---------|--------------|-----|------------|-----------------|
| ACSBlob | CacheService | IBC | OEM        | WAS             |

|                       |                              |                               |                         |           |
|-----------------------|------------------------------|-------------------------------|-------------------------|-----------|
|                       |                              |                               |                         |           |
| ACSDownloadService    | Compute                      | InfraServiceController        | OnboardRP               | WASPUBLIC |
| ACSFabric             | CPI                          | KeyVaultAdminResourceProvider | PXE                     |           |
| ACSFrontEnd           | CRP                          | KeyVaultControlPlane          | QueryServiceCoordinator |           |
| ACSMetrics            | DeploymentMachine            | KeyVaultDataPlane             | QueryServiceWorker      |           |
| ACSMigrationService   | DiskRP                       | KeyVaultInternalControlPlane  | SeedRing                |           |
| ACSMonitoringService  | Domain                       | KeyVaultInternalDataPlane     | SeedRingServices        |           |
| ACSSettingsService    | ECE                          | KeyVaultNamingService         | SLB                     |           |
| ACSTableMaster        | EventAdminRP                 | MDM                           | SQL                     |           |
| ACSTableServer        | EventRP                      | MetricsAdminRP                | SRP                     |           |
| ACSWac                | ExternalDNS                  | MetricsRP                     | Storage                 |           |
| ADFS                  | FabricRing                   | MetricsServer                 | StorageController       |           |
| ApplicationController | FabricRingServices           | MetricsStoreService           | URP                     |           |
| ASAppGateway          | FirstTierAggregation Service | MonAdminRP                    | SupportBridgeController |           |
| AzureBridge           | FRP                          | MonRP                         | SupportRing             |           |
| AzureMonitor          | Gateway                      | NC                            | SupportRingServices     |           |
| BareMetal             | HealthMonitoring             | NonPrivilegedAppGateway       | SupportBridgeRP         |           |
| BRP                   | HintingServiceV2             | NRP                           | UsageBridge             |           |
|                       |                              |                               |                         |           |

### Additional considerations on diagnostic logs

- The command takes some time to run based on which role(s) the logs are collecting. Contributing factors also include the time duration specified for log collection, and the numbers of nodes in the Azure Stack Hub environment.
- As log collection runs, check the new folder created in the **OutputSharePath** parameter specified in the command.

- Each role has its logs inside individual zip files. Depending on the size of the collected logs, a role may have its logs split into multiple zip files. For such a role, if you want to have all the log files unzipped into a single folder, use a tool that can unzip in bulk. Select all the zipped files for the role and select **extract here**. All the log files for that role will be unzipped into a single merged folder.
- A file called **Get-AzureStackLog\_Output.log** is also created in the folder that contains the zipped log files. This file is a log of the command output, which can be used for troubleshooting problems during log collection. Sometimes the log file includes `PS>TerminatingError` entries which can be safely ignored, unless expected log files are missing after log collection runs.
- To investigate a specific failure, logs may be needed from more than one component.
  - System and event logs for all infrastructure VMs are collected in the **VirtualMachines** role.
  - System and event logs for all hosts are collected in the **BareMetal** role.
  - Failover cluster and Hyper-V event logs are collected in the **Storage** role.
  - ACS logs are collected in the **Storage** and **ACS** roles.

#### **NOTE**

Size and age limits are enforced on the logs collected as it's essential to ensure efficient utilization of your storage space and to avoid getting flooded with logs. However, when diagnosing a problem, you sometimes need logs that don't exist anymore because of these limits. Thus, it's **highly recommended** that you offload your logs to an external storage space (a storage account in Azure, an additional on premises storage device, etc.) every 8 to 12 hours and keep them there for 1 - 3 months, depending on your requirements. You should also ensure this storage location is encrypted.

### **Invoke-AzureStackOnDemandLog**

You can use the **Invoke-AzureStackOnDemandLog** cmdlet to generate on-demand logs for certain roles (see the list at the end of this section). The logs generated by this cmdlet aren't present by default in the log bundle you receive when you execute the **Get-AzureStackLog** cmdlet. Also, it's recommended that you collect these logs only when requested by the Microsoft support team.

Currently, you can use the `-FilterByRole` parameter to filter log collection by the following roles:

- OEM
- NC
- SLB
- Gateway

#### **Example of collecting on-demand diagnostic logs**

```

$ipAddress = "<IP ADDRESS OF THE PEP VM>" # You can also use the machine name instead of IP here.

$password = ConvertTo-SecureString "<CLOUD ADMIN PASSWORD>" -AsPlainText -Force
$cred = New-Object -TypeName System.Management.Automation.PSCredential ("<DOMAIN NAME>\CloudAdmin", $password)

$shareCred = Get-Credential

$session = New-PSSession -ComputerName $ipAddress -ConfigurationName PrivilegedEndpoint -Credential $cred

$fromDate = (Get-Date).AddHours(-8)
$toDate = (Get-Date).AddHours(-2) # Provide the time that includes the period for your issue

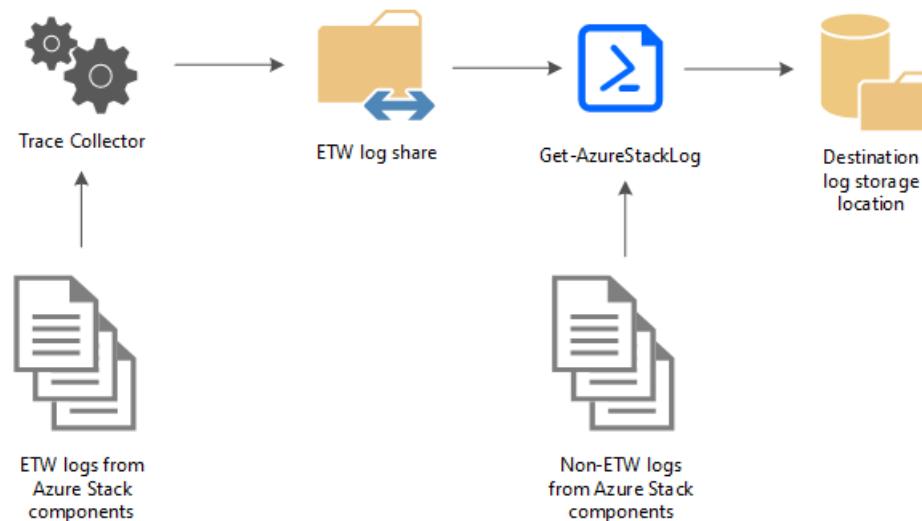
Invoke-Command -Session $session {
 Invoke-AzureStackOnDemandLog -Generate -FilterByRole "<on-demand role name>" # Provide the supported on-demand role name e.g. OEM, NC, SLB, Gateway
 Get-AzureStackLog -OutputSharePath "<external share address>" -OutputShareCredential $using:shareCred -FilterByRole Storage -FromDate $using:fromDate -ToDate $using:toDate
}

if ($session) {
 Remove-PSSession -Session $session
}

```

## How diagnostic log collection using the PEP works

Azure Stack Hub diagnostics tools help make log collection easy and efficient. The following diagram shows how the diagnostics tools work:



### Trace Collector

The Trace Collector is enabled by default and runs continuously in the background to collect all Event Tracing for Windows (ETW) logs from Azure Stack Hub component services. ETW logs are stored in a common local share with a five-day age limit. Once this limit is reached, the oldest files are deleted as new ones are created. The default maximum size allowed for each file is 200 MB. A size check happens every 2 minutes, and if the current file is  $\geq$  200 MB, it's saved and a new file generates. There's also an 8 GB limit on the total file size generated per event session.

### Get-AzureStackLog

The PowerShell cmdlet **Get-AzureStackLog** can be used to collect logs from all the components in an Azure Stack Hub environment. It saves them in zip files in a user-defined location. If the Azure Stack Hub technical support team needs your logs to help troubleshoot an issue, they may ask you to run **Get-AzureStackLog**.

#### Caution

These log files may contain personally identifiable information (PII). Take this into account before you publicly post any log files.

The following are some example log types that are collected:

- **Azure Stack Hub deployment logs**
- **Windows event logs**
- **Panther logs**
- **Cluster logs**
- **Storage diagnostic logs**
- **ETW logs**

These files are collected and saved in a share by Trace Collector. Get-AzureStackLog can then be used to collect them when necessary.

# Validate Azure Stack Hub system state

7 minutes to read • [Edit Online](#)

As an Azure Stack Hub operator, being able to determine the health and status of your system on demand is essential. The Azure Stack Hub validation tool (**Test-AzureStack**) is a PowerShell cmdlet that lets you run a series of tests on your system to identify failures if present. You'll typically be asked to run this tool through the [privileged end point \(PEP\)](#) when you contact Microsoft Customer Services Support (CSS) with an issue. With the system-wide health and status information at hand, CSS can collect and analyze detailed logs, focus on the area where the error occurred, and work with you to fix the issue.

## Running the validation tool and accessing results

As stated above, the validation tool is run via the PEP. Each test returns a **PASS/FAIL** status in the PowerShell window. Here's an outline of the end-to-end validation testing process:

1. Establish the trust. On an integrated system, run the following command from an elevated Windows PowerShell session to add the PEP as a trusted host on the hardened VM running on the hardware lifecycle host or the Privileged Access Workstation.

```
winrm s winrm/config/client '@{TrustedHosts=<IP Address of Privileged Endpoint>}'
```

If you're running the Azure Stack Development Kit (ASDK), sign in to the development kit host.

2. Access the PEP. Run the following commands to establish a PEP session:

```
Enter-PSSession -ComputerName "<ERCS VM-name/IP address>" -ConfigurationName PrivilegedEndpoint - Credential $localcred
```

### TIP

To access the PEP on an Azure Stack Development Kit (ASDK) host computer, use AzS-ERCS01 for - ComputerName.

3. Once you're in the PEP, run:

```
Test-AzureStack
```

For more information, see [Parameter considerations](#) and [Use case examples](#).

4. If any tests report **FAIL**, run `Get-AzureStackLog`. For instructions on an integrated system, see [To run Get-AzureStackLog on Azure Stack Hub integrated systems](#), or on the ASDK, see [Run Get-AzureStackLog on an ASDK system](#).

The cmdlet gathers logs generated by Test-AzureStack. We recommend you don't collect logs and contact CSS instead if tests report **WARN**.

5. If you're instructed to run the validation tool by the CSS, the CSS representative will request the logs you collected to continue troubleshooting your issue.

# Tests available

The validation tool lets you run a series of system-level tests and basic cloud scenarios that provide you with insight to the current state, allowing you to fix issues in your system.

## Cloud infrastructure tests

These low impact tests work on an infrastructure level and provide you with information on various system components and functions. Currently, tests are grouped into the following categories:

| TEST CATEGORY                                            | ARGUMENT FOR -INCLUDE AND -IGNORE |
|----------------------------------------------------------|-----------------------------------|
| Azure Stack Hub ACS Summary                              | AzsAcsSummary                     |
| Azure Stack Hub Active Directory Summary                 | AzsAdSummary                      |
| Azure Stack Hub Alert Summary                            | AzsAlertSummary                   |
| Azure Stack Hub Application Crash Summary                | AzsApplicationCrashSummary        |
| Azure Stack Hub Backup Share Accessibility Summary       | AzsBackupShareAccessibility       |
| Azure Stack Hub BMC Summary                              | AzsStampBMCSummary                |
| Azure Stack Hub Cloud Hosting Infrastructure Summary     | AzsHostingInfraSummary            |
| Azure Stack Hub Cloud Hosting Infrastructure Utilization | AzsHostingInfraUtilization        |
| Azure Stack Hub Control Plane Summary                    | AzsControlPlane                   |
| Azure Stack Hub Defender Summary                         | AzsDefenderSummary                |
| Azure Stack Hub Hosting Infrastructure Firmware Summary  | AzsHostingInfraFWSummary          |
| Azure Stack Hub Infrastructure Capacity                  | AzsInfraCapacity                  |
| Azure Stack Hub Infrastructure Performance               | AzsInfraPerformance               |
| Azure Stack Hub Infrastructure Role Summary              | AzsInfraRoleSummary               |
| Azure Stack Hub Network Infra                            | AzsNetworkInfra                   |
| Azure Stack Hub Portal and API Summary                   | AzsPortalAPISummary               |
| Azure Stack Hub Scale Unit VM Events                     | AzsScaleUnitEvents                |
| Azure Stack Hub Scale Unit VM Resources                  | AzsScaleUnitResources             |
| Azure Stack Hub Scenarios                                | AzsScenarios                      |
| Azure Stack Hub SDN Validation Summary                   | AzsSDNValidation                  |
| Azure Stack Hub Service Fabric Role Summary              | AzsSFRoleSummary                  |

| TEST CATEGORY                            | ARGUMENT FOR -INCLUDE AND -IGNORE |
|------------------------------------------|-----------------------------------|
| Azure Stack Hub Storage Data Plane       | AzsStorageDataPlane               |
| Azure Stack Hub Storage Services Summary | AzsStorageSvcsSummary             |
| Azure Stack Hub SQL Store Summary        | AzsStoreSummary                   |
| Azure Stack Hub Update Summary           | AzsInfraUpdateSummary             |
| Azure Stack Hub VM Placement Summary     | AzsVmPlacement                    |

## Cloud scenario tests

In addition to the infrastructure tests above, you can also run cloud scenario tests to check functionality across infrastructure components. Cloud admin credentials are required to run these tests because they involve resource deployment.

### NOTE

Currently you can't run cloud scenario tests using Active Directory Federated Services (AD FS) credentials.

The following cloud scenarios are tested by the validation tool:

- Resource group creation
- Plan creation
- Offer creation
- Storage account creation
- Virtual machine creation (VM)
- Blob storage operation
- Queue storage operation
- Table storage operation

## Parameter considerations

- The parameter **List** can be used to display all available test categories.
- The parameters **Include** and **Ignore** can be used to include or exclude test categories. For more information about these arguments, see the following section.

```
Test-AzureStack -Include AzsSFRoleSummary, AzsInfraCapacity
```

```
Test-AzureStack -Ignore AzsInfraPerformance
```

- A tenant VM is deployed as part of the cloud scenario tests. You can use **DoNotDeployTenantVm** to disable this VM deployment.
- You need to supply the **ServiceAdminCredential** parameter to run cloud scenario tests as described in the [Use case examples](#) section.
- **BackupSharePath** and **BackupShareCredential** are used when testing infrastructure backup settings as shown in the [Use case examples](#) section.

- **DetailedResults** can be used to get pass/fail/warning information for each test, as well as the overall run. When not specified, **Test-AzureStack** returns **\$true** if there are no failures, and **\$false** if there are failures.
- **TimeoutSeconds** can be used to set a specific time for each group to complete.
- The validation tool also supports common PowerShell parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see [About Common Parameters](#).

## Use case examples

### Run validation without cloud scenarios

Run the validation tool without the **ServiceAdminCredential** parameter to skip running cloud scenario tests:

```
New-PSSession -ComputerName "<ERCS VM-name/IP address>" -ConfigurationName PrivilegedEndpoint -Credential $localcred
Test-AzureStack
```

### Run validation with cloud scenarios

Supplying the validation tool with the **ServiceAdminCredentials** parameter runs the cloud scenario tests by default:

```
Enter-PSSession -ComputerName "<ERCS VM-name/IP address>" -ConfigurationName PrivilegedEndpoint -Credential $localcred
Test-AzureStack -ServiceAdminCredential "<Cloud administrator user name>"
```

If you wish to run ONLY cloud scenarios without running the rest of the tests, you can use the **Include** parameter to do so:

```
Enter-PSSession -ComputerName "<ERCS VM-name/IP address>" -ConfigurationName PrivilegedEndpoint -Credential $localcred
Test-AzureStack -ServiceAdminCredential "<Cloud administrator user name>" -Include AzsScenarios
```

The cloud admin user name must be typed in the UPN format: serviceadmin@contoso.onmicrosoft.com (Azure AD). When prompted, type the password to the cloud admin account.

### Groups

To improve the operator experience, a **Group** parameter has been enabled to run multiple test categories at the same time. Currently, there are three groups defined: **Default**, **UpdateReadiness**, and **SecretRotationReadiness**.

- **Default**: Considered to be a standard run of **Test-AzureStack**. This group is run by default if no other groups are selected.
- **UpdateReadiness**: A check to see if the Azure Stack Hub instance can be updated. When the **UpdateReadiness** group is run, warnings are displayed as errors in the console output, and they should be considered as blockers for the update. As of Azure Stack Hub Version 1910 the following categories are part of the **UpdateReadiness** group:
  - **AzsInfraFileValidation**
  - **AzsActionPlanStatus**
  - **AzsStampBMCSummary**
- **SecretRotationReadiness**: A check to see if the Azure Stack Hub instance is in a state in which secret

rotation can be run. When the **SecretRotationReadiness** group is run, warnings are displayed as errors in the console output and they should be considered as blockers for secret rotation. The following categories are part of the SecretRotationReadiness Group:

- **AzsAcsSummary**
- **AzsDefenderSummary**
- **AzsHostingInfraSummary**
- **AzsInfraCapacity**
- **AzsInfraRoleSummary**
- **AzsPortalAPISummary**
- **AzsSFRoleSummary**
- **AzsStorageSvcsSummary**
- **AzsStoreSummary**

#### Group parameter example

The following example runs **Test-AzureStack** to test system readiness before installing an update or hotfix using **Group**. Before you start the installation of an update or hotfix, run **Test-AzureStack** to check the status of your Azure Stack Hub:

```
Test-AzureStack -Group UpdateReadiness
```

If your Azure Stack Hub is running a version before 1811, use the following PowerShell commands to run **Test-AzureStack**:

```
New-PSSession -ComputerName "<ERCS VM-name/IP address>" -ConfigurationName PrivilegedEndpoint -Credential $localcred
Test-AzureStack -Include AzsControlPlane, AzsDefenderSummary, AzsHostingInfraSummary,
AzsHostingInfraUtilization, AzsInfraCapacity, AzsInfraRoleSummary, AzsPortalAPISummary, AzsSFRoleSummary,
AzsStampBMCSummary
```

#### Run validation tool to test infrastructure backup settings

Before configuring infrastructure backup, you can test the backup share path and credential using the **AzsBackupShareAccessibility** test:

```
Enter-PSSession -ComputerName "<ERCS VM-name/IP address>" -ConfigurationName PrivilegedEndpoint -Credential $localcred
Test-AzureStack -Include AzsBackupShareAccessibility -BackupSharePath "\\<fileserver>\<fileshare>" -
BackupShareCredential $using:backupcred
```

After configuring backup, you can run **AzsBackupShareAccessibility** to validate the share is accessible from the ERCS:

```
Enter-PSSession -ComputerName "<ERCS VM-name/IP address>" -ConfigurationName PrivilegedEndpoint -Credential $localcred
Test-AzureStack -Include AzsBackupShareAccessibility
```

To test new credentials with the configured backup share, run:

```
Enter-PSSession -ComputerName "<ERCS VM-name/IP address>" -ConfigurationName PrivilegedEndpoint -Credential $localcred
Test-AzureStack -Include AzsBackupShareAccessibility -BackupShareCredential "<PSCredential for backup
share>"
```

## Run validation tool to test network infrastructure

This test checks the connectivity of the network infrastructure bypassing the Azure Stack Hub software defined network (SDN). It demonstrates connectivity from a Public VIP to the configured DNS forwarders, NTP servers, and authentication endpoints. This includes connectivity to Azure when using Azure AD as identity provider or the federated server when using AD FS as identity provider.

Include the debug parameter to get a detailed output of the command:

```
Test-AzureStack -Include AzsNetworkInfra -Debug
```

## Next steps

To learn more about Azure Stack Hub diagnostics tools and issue logging, see [Azure Stack Hub diagnostics tools](#).

To learn more about troubleshooting, see [Microsoft Azure Stack Hub troubleshooting](#).

# Troubleshoot issues in Azure Stack Hub

3 minutes to read • [Edit Online](#)

This document provides troubleshooting information for Azure Stack Hub integrated environments. For help with the Azure Stack Development Kit, see [ASDK Troubleshooting](#) or get help from experts on the [Azure Stack Hub MSDN Forum](#).

## Frequently asked questions

These sections include links to docs that cover common questions sent to Microsoft Customer Support Services (CSS).

### Purchase considerations

- [How to buy](#)
- [Azure Stack Hub overview](#)

### Updates and diagnostics

- [How to use diagnostics tools in Azure Stack Hub](#)
- [How to validate Azure Stack Hub system state](#)
- [Update package release cadence](#)

### Supported operating systems and sizes for guest VMs

- [Guest operating systems supported on Azure Stack Hub](#)
- [VM sizes supported in Azure Stack Hub](#)

### Azure Marketplace

- [Azure Marketplace items available for Azure Stack Hub](#)

### Manage capacity

#### Memory

To increase the total available memory capacity for Azure Stack Hub, you can add additional memory. In Azure Stack Hub, your physical server is also referred to as a scale unit node. All scale unit nodes that are members of a single scale unit must have [the same amount of memory](#).

#### Retention period

The retention period setting lets a cloud operator to specify a time period in days (between 0 and 9999 days) during which any deleted account can potentially be recovered. The default retention period is set to **0** days. Setting the value to **0** means that any deleted account is immediately out of retention and marked for periodic garbage collection.

- [Set the retention period](#)

### Security, compliance, and identity

#### Manage RBAC

A user in Azure Stack Hub can be a reader, owner, or contributor for each instance of a subscription, resource group, or service.

- [Azure Stack Hub Manage RBAC](#)

If the built-in roles for Azure resources don't meet the specific needs of your organization, you can create your own custom roles. For this tutorial, you create a custom role named Reader Support Tickets using Azure PowerShell.

- [Tutorial: Create a custom role for Azure resources using Azure PowerShell](#)

## Manage usage and billing as a CSP

- [Manage usage and billing as a CSP](#)
- [Create a CSP or APSS subscription](#)

Choose the type of shared services account that you use for Azure Stack Hub. The types of subscriptions that can be used for registration of a multi-tenant Azure Stack Hub are:

- Cloud Solution Provider
- Partner Shared Services subscription

## Get scale unit metrics

You can use PowerShell to get stamp utilization information without help from CSS. To obtain stamp utilization:

1. Create a PEP session.
2. Run `test-azurestack`.
3. Exit PEP session.
4. Run `get-azurestacklog -filterbyrole seedring` using an invoke-command call.
5. Extract the seedring .zip. You can obtain the validation report from the ERCS folder where you ran `test-azurestack`.

For more information, see [Azure Stack Hub Diagnostics](#).

## Troubleshoot virtual machines (VMs)

### Default image and gallery item

A Windows Server image and gallery item must be added before deploying VMs in Azure Stack Hub.

### I've deleted some VMs, but still see the VHD files on disk

This behavior is by design:

- When you delete a VM, VHDs aren't deleted. Disks are separate resources in the resource group.
- When a storage account gets deleted, the deletion is visible immediately through Azure Resource Manager. But the disks it may contain are still kept in storage until garbage collection runs.

If you see "orphan" VHDs, it's important to know if they're part of the folder for a storage account that was deleted. If the storage account wasn't deleted, it's normal that they're still there.

You can read more about configuring the retention threshold and on-demand reclamation in [manage storage accounts](#).

## Troubleshoot storage

### Storage reclamation

It may take up to 14 hours for reclaimed capacity to show up in the portal. Space reclamation depends on different factors including usage percentage of internal container files in block blob store. Therefore, depending on how much data is deleted, there's no guarantee on the amount of space that could be reclaimed when garbage collector runs.

### Azure Storage Explorer not working with Azure Stack Hub

If you're using an integrated system in a disconnected scenario, it's recommended to use an Enterprise Certificate Authority (CA). Export the root certificate in a Base-64 format and then import it in Azure Storage Explorer. Make sure that you remove the trailing slash (/) from the Resource Manager endpoint. For more information, see [Prepare for connecting to Azure Stack Hub](#).

# Troubleshooting App Service

## Create-AADIdentityApp.ps1 script fails

If the Create-AADIdentityApp.ps1 script that's required for App Service fails, be sure to include the required `-AzureStackAdminCredential` parameter when running the script. For more information, see [Prerequisites for deploying App Service on Azure Stack Hub](#).

# Azure Stack Hub training and certification

2 minutes to read • [Edit Online](#)

*Applies to: Azure Stack Hub integrated systems*

Want to learn about Azure Stack Hub and demonstrate your Azure Stack Hub proficiency? Check out the following training and certification opportunities.

## Training

- Microsoft official courses on-demand:
  - [ODX20537: Configuring and Operating a Hybrid Cloud with Microsoft Azure Stack Hub \(180 Day\)](#)
  - [OD20537: Configuring and Operating a Hybrid Cloud with Microsoft Azure Stack Hub \(90 Day\)](#)
- Microsoft IT training course:
  - [Course 20537A: Configuring and Operating a Hybrid Cloud with Microsoft Azure Stack Hub](#)
- Open edx:
  - [edX: Configuring and Operating Microsoft Azure Stack Hub online course](#)
- Microsoft Learning Paths:
  - [Job roles and learning paths](#)

## Certification

*Configuring and Operating a Hybrid Cloud with Microsoft Azure Stack Hub certification, Exam 70-537*

## Next steps

[Azure Stack Hub documentation](#)