# **Manage Azure identities and governance (15–20%)**

## **Manage Azure Active Directory (Azure AD) objects**

An object in Azure AD is a programmatic high-level data construct that represents such things as users, groups, and contacts. When you create a new user or contact in Azure AD, you're creating a new instance of that object

### **Create users and groups**

To add or delete users and groups you must be a User administrator or Global administrator.

Users:

You can create new users and using the Azure Active Directory portal. The following fields are required:

* **Name:** The first and last name of the new user. For example, Mary Parker.
* **User name:** The user's 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.
* Groups. Optionally, add the user to one or more existing groups.
* **Roles**: If you require Azure AD administrative permissions for the user, you can add them to an Azure AD role.
* **Job info:** Add more information about the user such as title, department etc

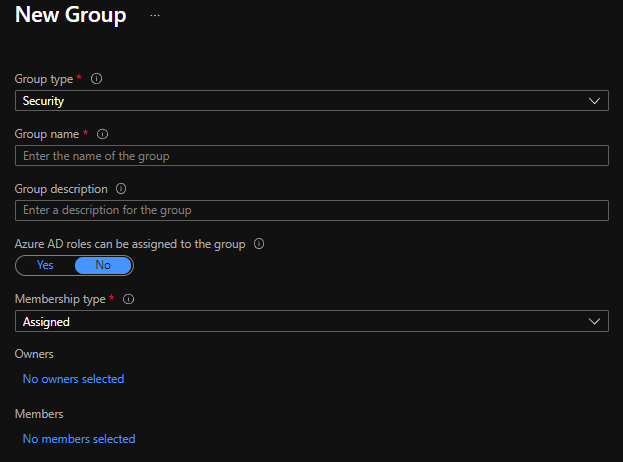


Once complete, copy the autogenerated password provided in the Password box to give this password to the user to sign in for the first time.

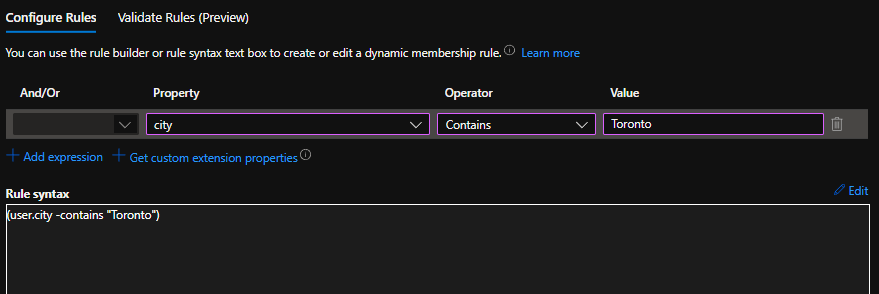
Groups:

There are several group types:

* **Security:** A security group can have users, devices, groups and service principals as its members and users and service principals as its owners. Used to manage member and computer access to shared resources for a group of users. For example, creating a security group for a specific security policy. By doing it this way, you can give a set of permissions to all the members at once, instead of having to add permissions to each member individually.
* **Microsoft 365:** Gives members access to a shared mailbox, calendar, files, SharePoint site, etc. A Microsoft 365 group can have only users as its members. Both users and service principals can be owners of a Microsoft 365 group



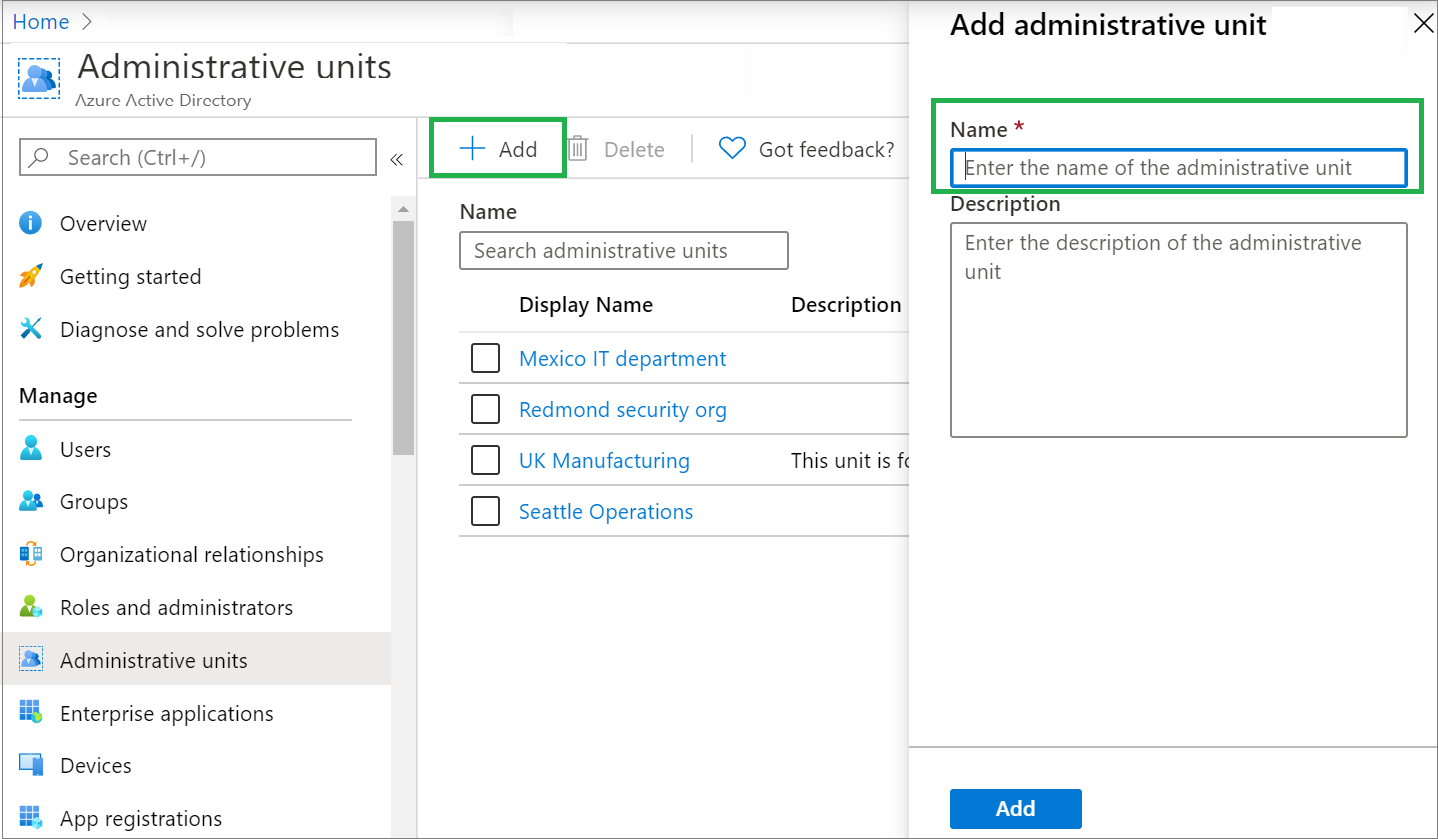
There are also several membership types:

* **Assigned:** Allows adding specific users to be members of this group and to have unique permissions.
* **Dynamic user:** Allows using dynamic membership rules to automatically add and remove members. If a member's attributes change, the system looks at your dynamic group rules for the directory to see if the member meets the rule requirements (is added) or no longer meets the rules requirements (is removed).
* **Dynamic Device:** Same as dynamic user but for device properties (I.e. deviceOsType Contains X) 

### **Create administrative units**

Administrative units allow subdividing an organization into any unit that you want, and then assign specific administrators that can manage only the members of that unit. For example, you could use administrative units to delegate permissions to administrators of each school at a large university, so they could control access, manage users, and set policies only in the School of Engineering.

Administrative units are created by using either the Azure portal, PowerShell or Microsoft Graph.

An Azure AD role can then be assigned with a scope that's limited to one or more administrative units and users or groups can be added to an administrative unit to restrict the scope of role permissions

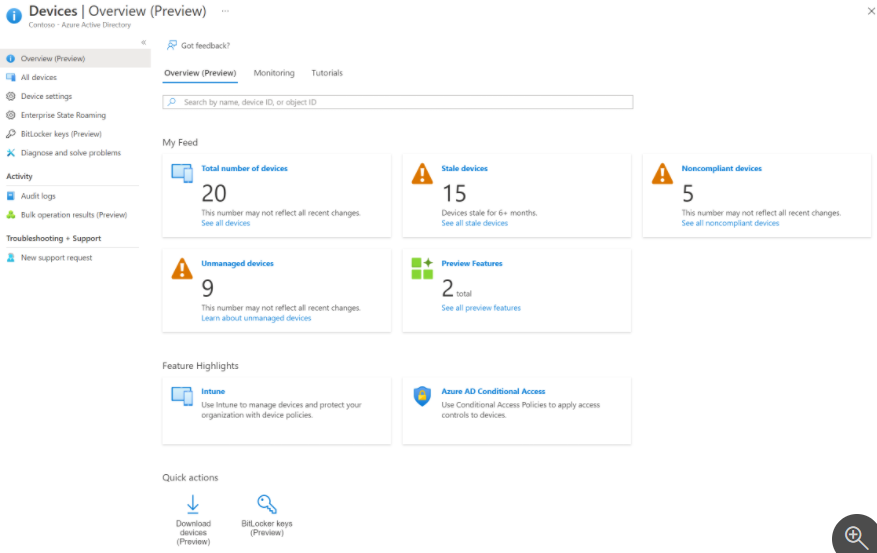
### **Manage user and group properties**

Add user profile information, including a profile picture, job-specific information, contact info, and some settings using Azure AD.

Using Azure AD, a group's settings can be edited, including updating its name, description, or membership type.

### **Manage device settings**

Azure Active Directory (Azure AD) provides a central place to manage device identities and monitor related event information by going to Azure Active Directory > Devices



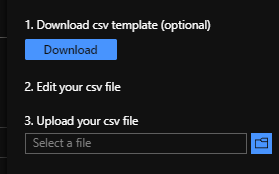
In the devices overview, the number of total devices, stale devices, noncompliant devices, and unmanaged devices can be managed. There are also links to Intune, Conditional Access, BitLocker keys, and basic monitoring.

### **Perform bulk user updates**

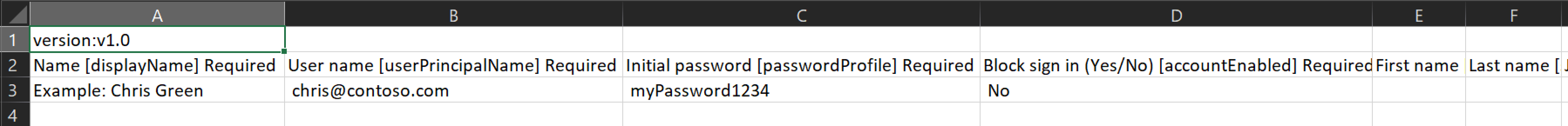
Azure AD supports bulk user create and delete operations and supports downloading lists of users as a csv.

To create users in bulk

1. As a “User administrator” in the organization. Navigate to Azure AD > Users > Bulk create.
2. On the Bulk create user page, select Download to receive a valid comma-separated values (CSV) file of user properties.



1. Open the CSV file and add a line for each user you want to create. The only required values are Name, User principal name, Initial password and Block sign in (Yes/No). Then save the file.



1. On the Bulk create user page, under Upload your CSV file, browse to the file. When you select the file and click Submit, validation of the CSV file starts.
2. After the file contents are validated, you’ll see File uploaded successfully. If there are errors, you must fix them before you can submit the job.
3. When your file passes validation, select Submit to start the Azure bulk operation that imports the new users.
4. When the import operation completes, you'll see a notification of the bulk operation job status.

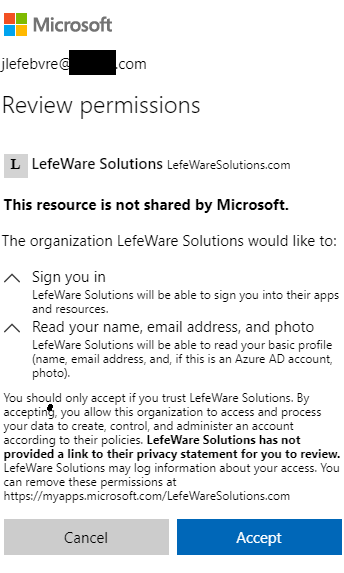
You can see the status of all of your pending bulk requests in the Bulk operation results page.

### **Manage guest accounts**

Azure AD business-to-business (B2B) collaboration is a feature within External Identities that allows inviting guest users to collaborate with the current organization. B2B collaboration, allows for securely sharing a company’s applications and services with guest users from any other organization even if they are not using Azure AD. It also allows the guest to use their own credentials.

To add guest users to an organization in the Azure portal:

1. Click + New User in Azure AD
2. Instead of creating a new user, select “Invite user”
3. Select groups & roles to assign to user just like any other user
4. User will be sent an invitation for which they will accept and provide a redemption page



User can go to <https://myaccount.microsoft.com/organizations> to see what apps and orgs they are part of

### **Configure Azure AD join**

Azure AD joined devices are signed in to using an organizational Azure AD account. Access to resources in the organization can be further limited based on that Azure AD account and Conditional Access policies applied to the device identity.

Azure AD Join provide a means to enforce organization-required configurations like requiring storage to be encrypted, password complexity, software installations, and software updates. Administrators can make organization applications available to Azure AD joined etc.

Azure AD join can be accomplished using

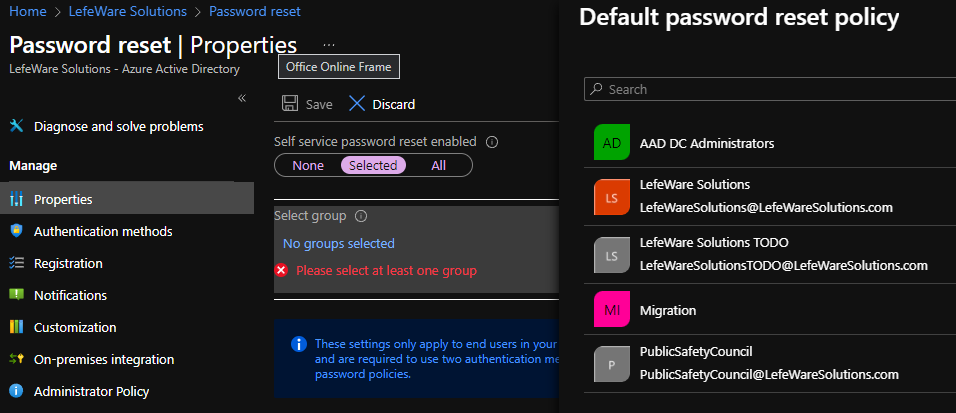
* Self-service options like the Out of Box Experience (OOBE),
* Bulk enrollment
* Windows Autopilot.

### **Configure self-service password reset (SSPR)**

Enables users to reset their passwords without contacting IT staff for help. If Azure AD locks a user's account or they forget their password, the user can unblock themselves and continue working, thus reducing the non-productive time and high support costs for most common password-related issues.

Azure AD lets you enable SSPR for None, Selected, or All users:

1. In Azure Active Directory, then select Password reset from the menu on the left side.
2. From the Properties page, under the option Self service password reset enabled, select Select group
3. Browse for and select your Azure AD group, like SSPR-Test-Group, then choose Select.



With SSPR enabled and set up, test the SSPR process as follows:

1. Ensure user has authentication methods contact information registered .
2. Open a new browser window in InPrivate or incognito mode, and browse to <https://aka.ms/sspr>.
3. Enter your non-administrator test users' account information, like phone number, the characters from the CAPTCHA, and then select Next.
4. Follow the verification steps to reset your password. When finished, you'll receive an email notification that your password was reset

## **Manage role-based access control (RBAC)**

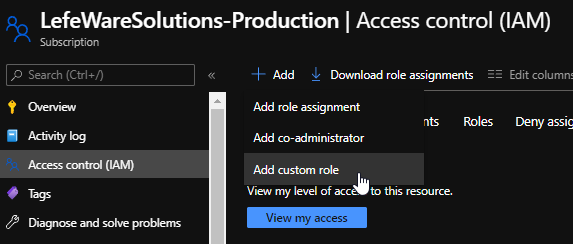
Azure role-based access control (Azure RBAC) helps you manage who has access to Azure resources, what they can do with those resources, and what areas they have access to.

### **Create a custom role**

If the Azure built-in roles don't meet the specific needs of your organization, you can create your own custom roles. Just like built-in roles, custom roles can be assigned to users, groups, and service principals at subscription and resource group scopes. Custom roles are stored in an Azure AD directory and can be shared across subscriptions.

Custom roles can be created using the Azure portal, Azure PowerShell, Azure CLI, or the REST API. To create a custom role from portal:

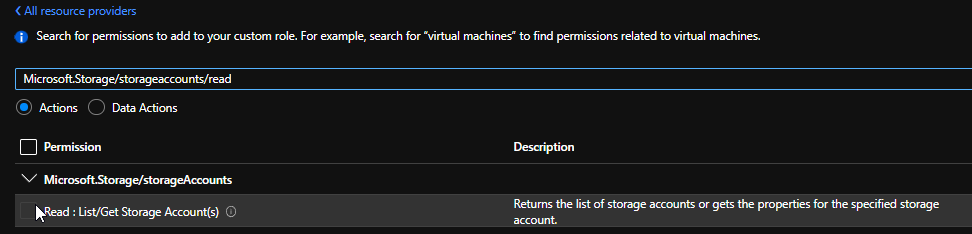
1. Open a subscription or resource group where the custom role should be assignable and then open “Access control (IAM)”
2. Select “Add Custom role”



1. Give it a Name, Description and for Baseline permissions select one of the following:
   1. Clone a role
   2. Start from scratch
   3. Start from JSON



1. On the Permissions tab, specify the permissions for the custom role. Depending on whether cloning a role or if starting with JSON, the Permissions tab might already list some permissions.



1. On the Assignable scopes tab, specify where the custom role is available for assignment, such as management group, subscriptions, or resource groups. This tab might already list the scope where the tab was opened the Access control (IAM) page.
2. On the JSON tab, the custom role is formatted in JSON. The JSON can be directly edited here (I.e. add a wildcard (\*) permission – Microsoft.Storage/storageAccounts/\*).

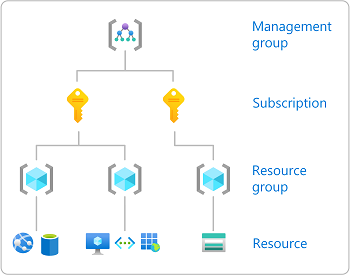


1. Review and create

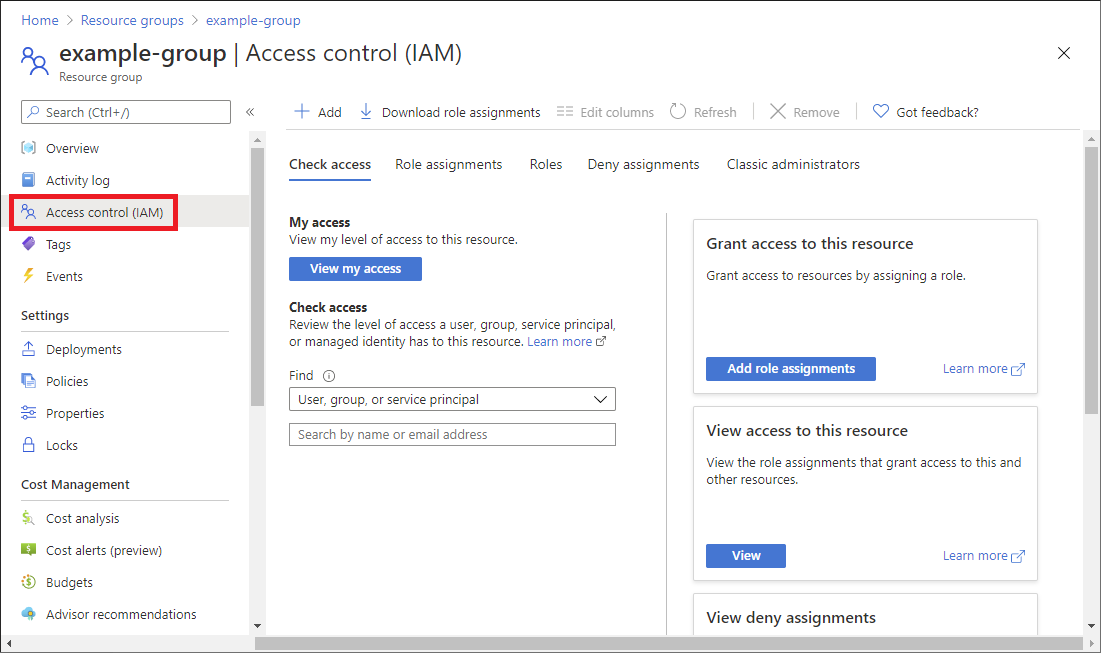
View the custom role in the Roles list

### **Provide access to Azure resources by assigning roles at different scopes**

When assigning roles, a scope must be specified. Scope is the set of resources the access applies to. In Azure, a scope can be applied at four levels from broad to narrow: management group, subscription, resource group, and resource.



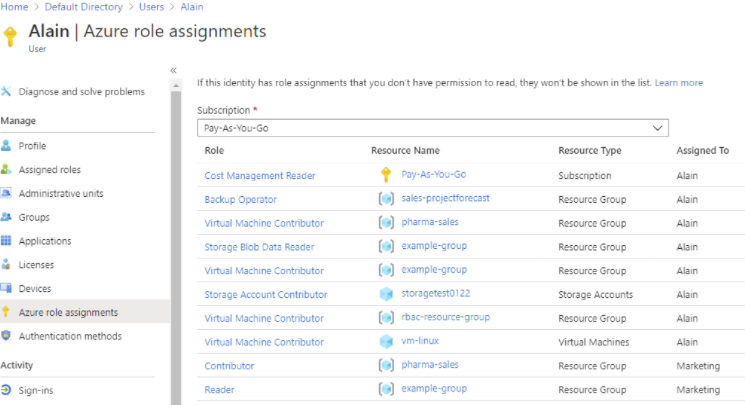
Access control (IAM) is the page that is typically used to assign roles to grant access to Azure resources. It's also known as identity and access management (IAM) and appears in several locations in the Azure portal. The following shows an example of the Access control (IAM) page for a resource group:



### **Interpret access assignments**

A quick way to see the roles assigned to a user or group in a subscription is to use the Azure role assignments pane.

1. In the portal select Azure AD and then select Users or Groups.
2. Click the user or group to list the role assignments for.
3. Click “Azure role assignments”.
4. A list of roles assigned to the selected user or group at various scopes such as management group, subscription, resource group, or resource will be listed.



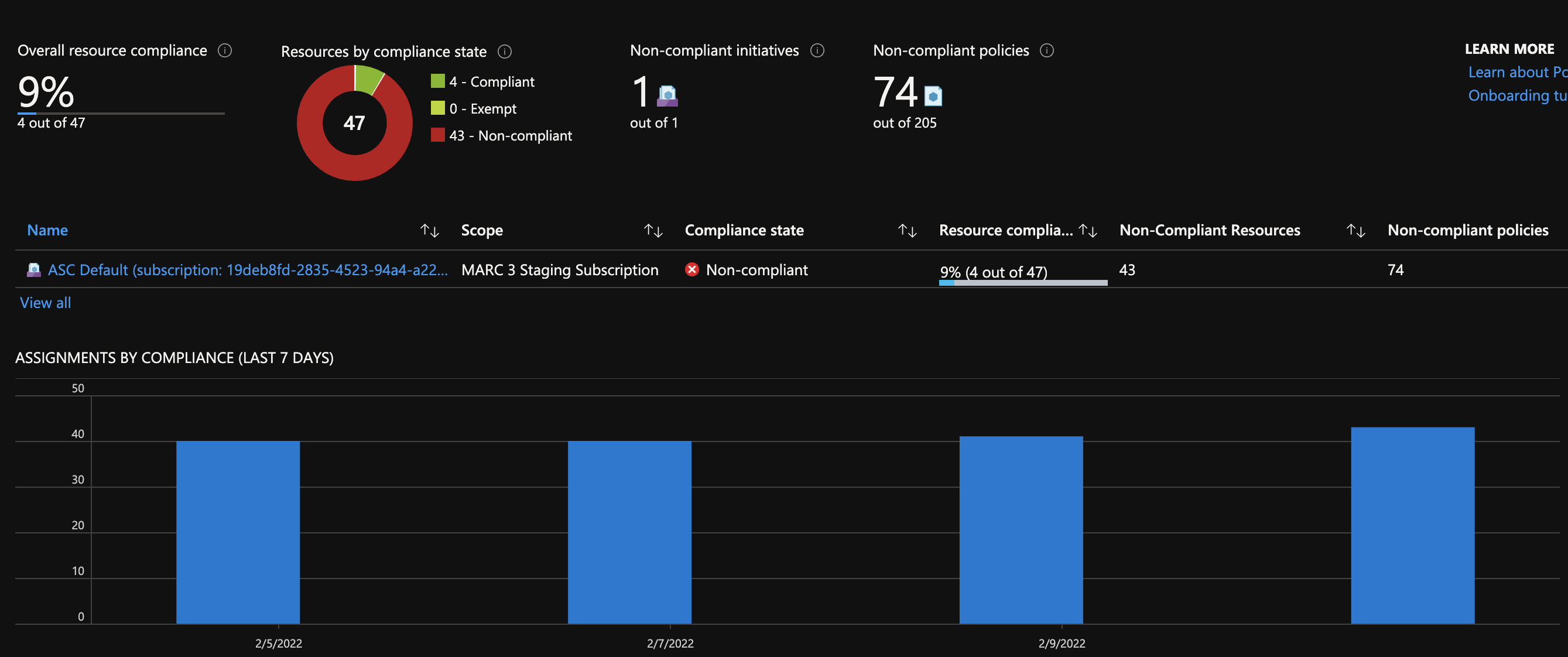
## **Manage subscriptions and governance**

Governance provides mechanisms and processes to maintain control over applications and resources in Azure. It involves planning initiatives and setting strategic priorities.

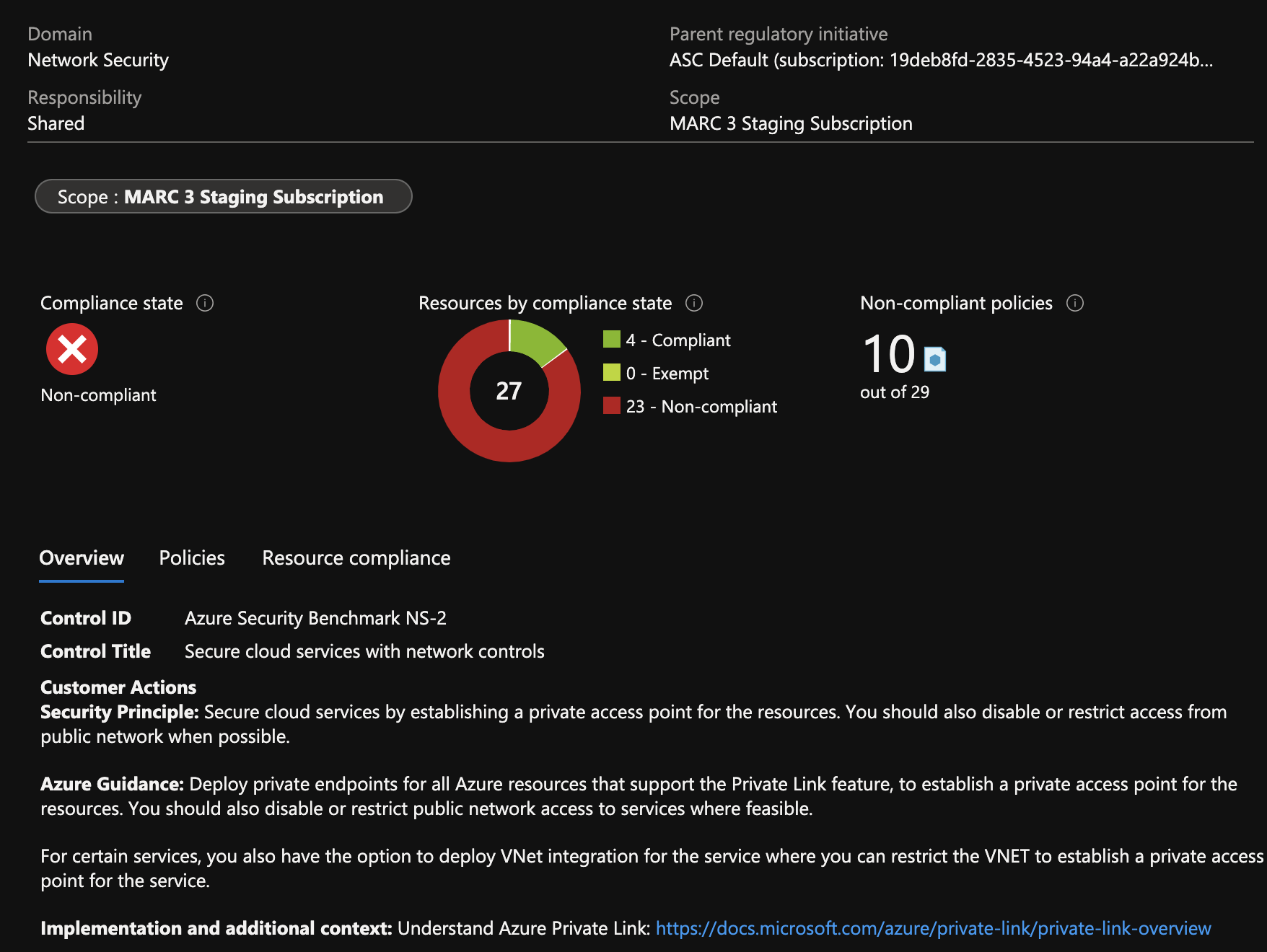
### **Configure Azure policies**

Azure Policy helps to enforce organizational standards and to assess compliance at-scale. The compliance dashboard in Azure provides an aggregated view to evaluate the overall state of the environment, with the ability to drill down to the per-resource, per-policy granularity. It also helps to bring resources to compliance through bulk remediation for existing resources and automatic remediation for new resources.

To simplify management, several business rules can be grouped together to form a policy initiative (sometimes called a policySet)



Azure Policy evaluates resources in Azure by comparing the properties of those resources to business rules. These business rules, described in JSON format, are known as policy definitions.

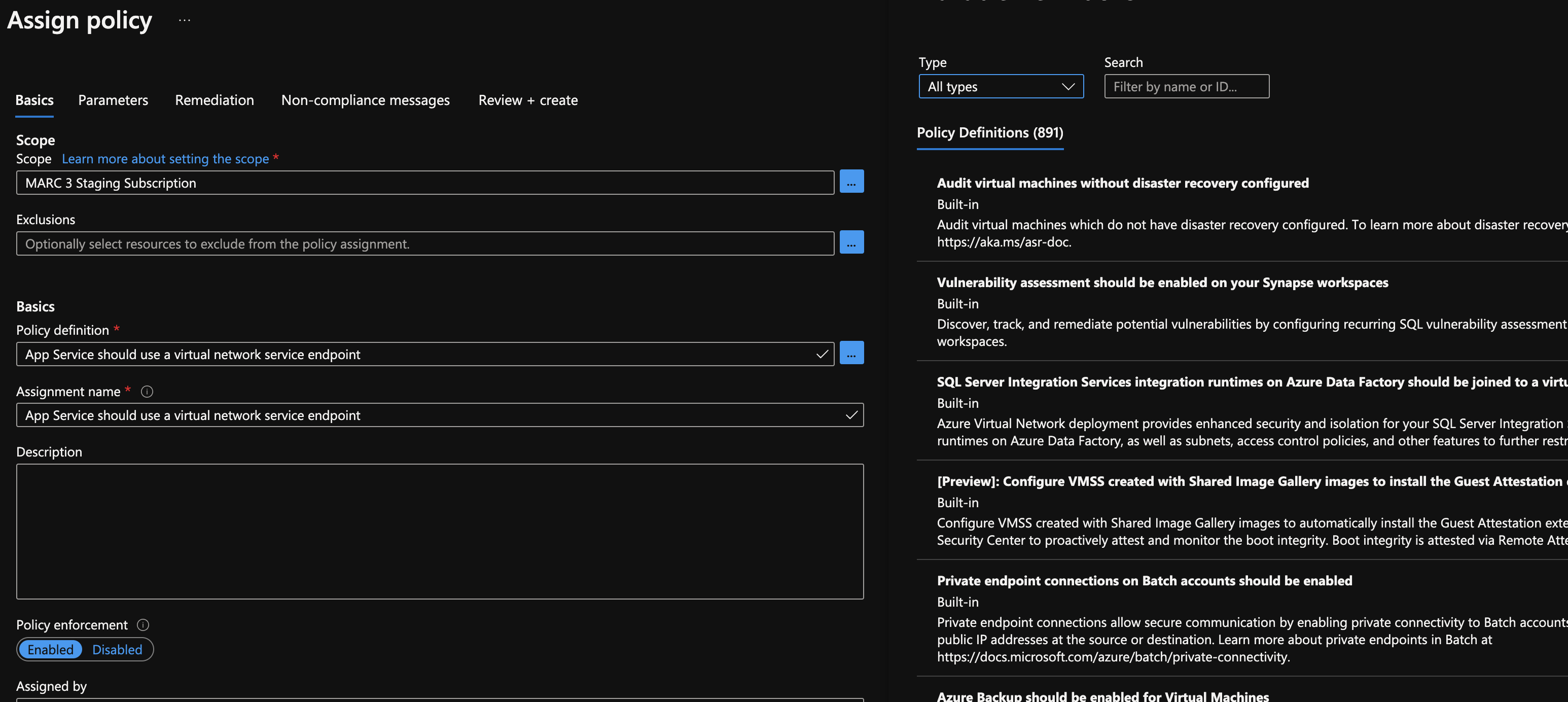


Creating and implementing a policy in Azure Policy begins with creating a policy definition. Every policy definition has conditions under which it's enforced and has a defined effect that takes place if the conditions are met. There are several built-in policies that are available by default. For example:

* Allowed Storage Account SKUs (Deny)
* Allowed Locations (Deny)
* Allowed Virtual Machine SKUs
* Add a tag to resources (Modify): Applies a required tag and its default value if it's not specified by the deploy request.
* Not allowed resource types (Deny)

To assign a custom policy follow these steps:

1. From the Policy page, select Assignments on the left side of the Azure Policy page.
2. Select Assign Policy from the top of the Policy - Assignments page.
3. On the Assign Policy page:
   1. Set the Scope (I.e., subscription or resourcegroup). A scope determines what resources or grouping of resources the policy assignment gets enforced on. Then use the Select button at the bottom of the Scope page.
   2. Select resource exclusions
   3. Select the Policy definition ellipsis to open the list of available definitions. Azure Policy comes with built-in policy definitions you can use. Enforce tag and its value
   4. The Assignment name is automatically populated with the policy name you selected, but you can change it. F
   5. Enable or disable this policy



1. Leave Create a Managed Identity unchecked. This box must be checked when the policy or initiative includes a policy with either the deployIfNotExists or modify effect.
2. Set the Non-compliance message
3. Review and Create

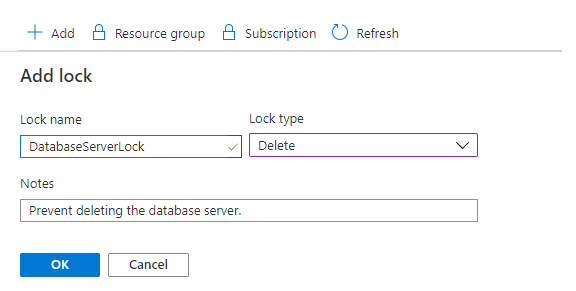
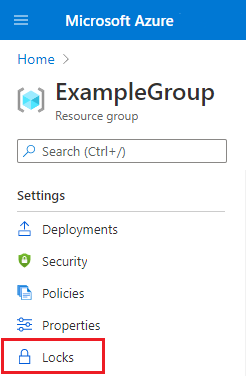
The assignment policy can then be viewed from Policy dashboard to begin identifying non-compliant resources to understand the compliance state of your environment.

### **Configure resource locks**

An Azure administrator can lock a subscription, resource group, or resource to prevent other users in an organization from accidentally deleting or modifying critical resources. The lock overrides any permissions the user might have. You can set the lock level to CanNotDelete or ReadOnly

To configure a lock:

1. In the Settings blade for the resource, resource group, or subscription that you wish to lock, select Locks.

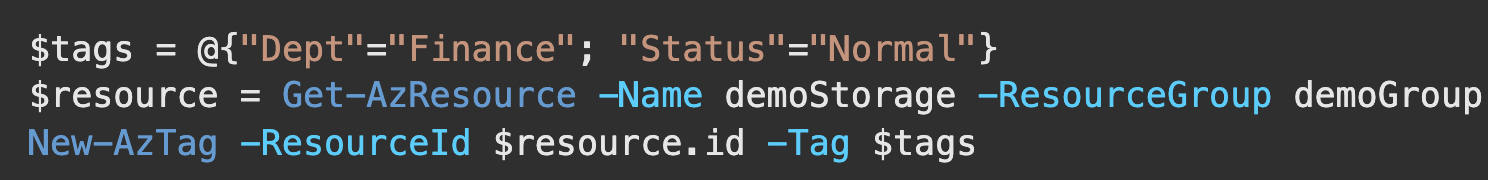


1. To add a lock, select Add. If you want to create a lock at a parent level (e.g. resource group where the resource resides), select the parent.
2. Give the lock a name and lock level. Optionally, you can add notes that describe the lock.

### **Apply and manage tags on resources**

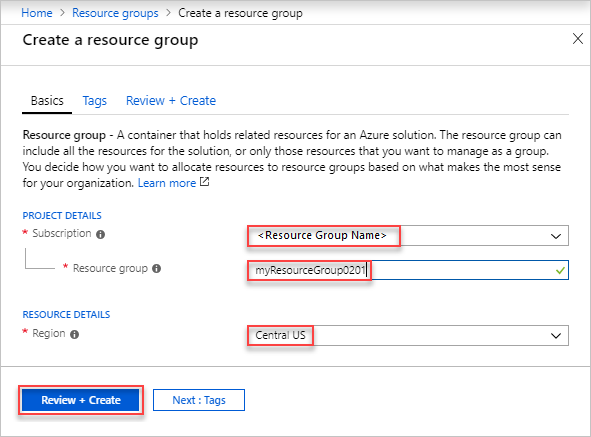
Tags can be applied to Azure resources, resource groups, and subscriptions to logically organize them into a taxonomy. Each tag consists of a name and a value pair. For example, applying the name Environment and the value Production to all the resources in production.

The Contributor role or Microsoft.Resources/tags permission grants the required access to apply tags to any entity. The following PowerShell script can be used to apply tags



### **Manage resource groups**

A resource group is a container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that should be managed as a group. Generally, add resources that share the same lifecycle to the same resource group so you can easily deploy, update, and delete them as a group.



### **Manage subscriptions**

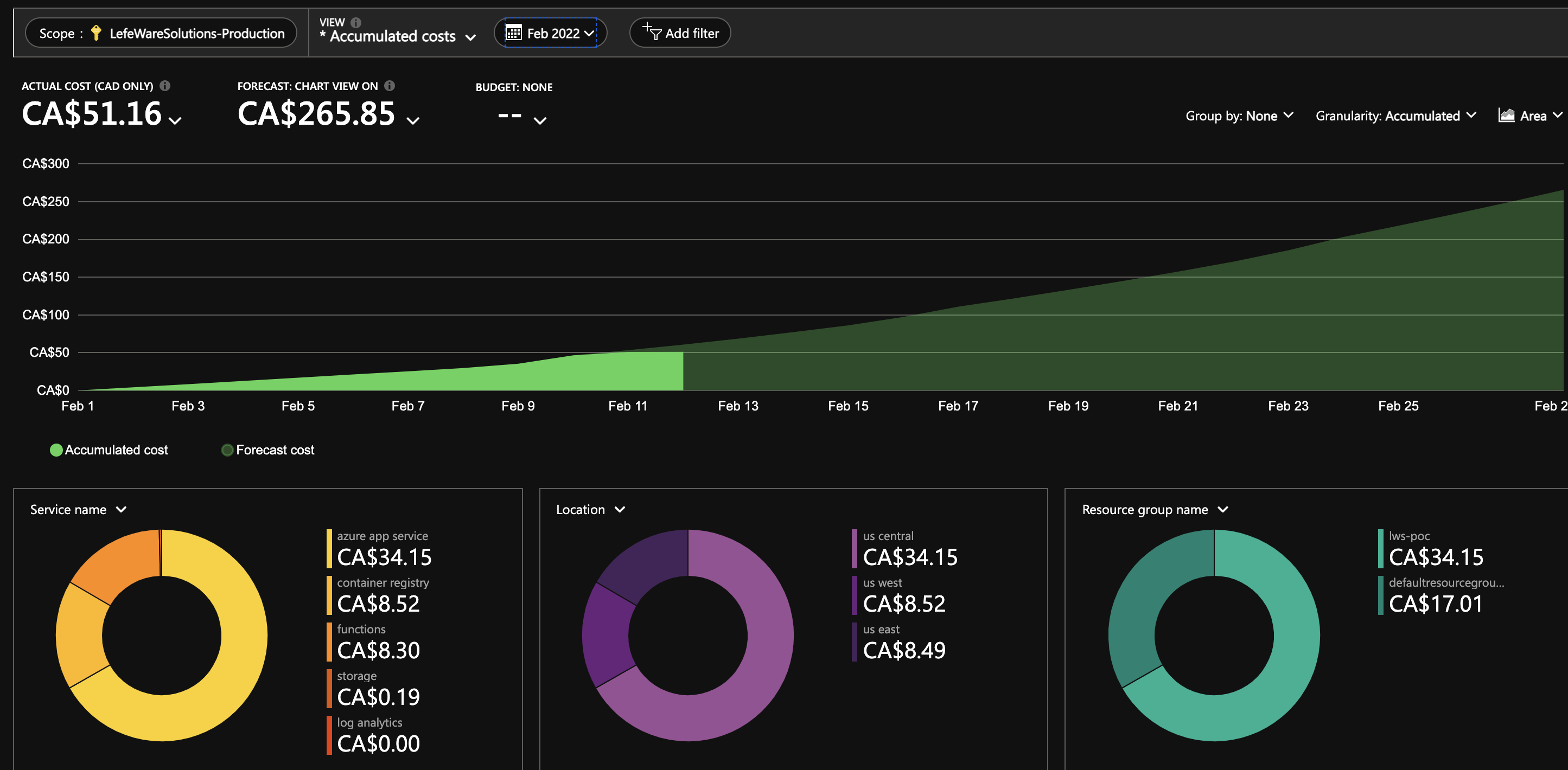
Organizations often use multiple Azure subscriptions to avoid per-subscription resource limits and to better manage and govern their Azure resources.

### **Manage costs**

Cost Management + Billing is a suite of tools provided by Microsoft that help you analyze, manage, and optimize the costs of workloads. The Cost Management + Billing features can be used to:

* Conduct billing administrative tasks such as paying bills
* Manage billing access to costs
* Download cost and usage data that was used to generate monthly invoices
* Proactively apply data analysis to costs
* Set spending thresholds
* Identify opportunities for workload changes that can optimize spending

Cost Management shows organizational cost and usage patterns with advanced analytics:



### **Configure management groups**

If an organization has multiple subscriptions, then a management group hierarchy can be used to help manage subscriptions and resources. Azure management groups help efficiently manage:

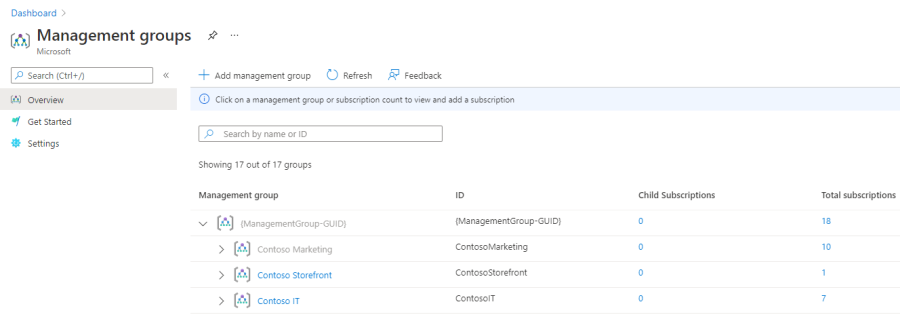
* Access
* Policies
* Compliance

If applying access or policy at one level in the hierarchy, it propagates down to the lower levels.



To create a management group in the portal:

1. Select Management groups in Azure.
2. Select + Add management group.
3. Select Create New and give the management an id and name.
4. Once created, subscriptions can be moved between groups and access and policies can be set at the management group level by selecting



# **Implement and manage storage (15–20%)**

## **Secure storage**

The Azure Storage platform is Microsoft's cloud storage solution for modern data storage scenarios. Core storage services offer a massively scalable object store for data objects (Blob Storage), disk storage for Azure virtual machines (VMs), a file system service for the cloud (File storage), a messaging store for reliable messaging (queue storage), and a NoSQL store (table storage)

### **Configure network access to storage accounts**

Azure storage accounts need to be globally unique because each account is bound to an Azure Resource Manager API endpoint:

http://<***storage account name***>.<***service type name***>.core.windows.net/<***service containe***r>

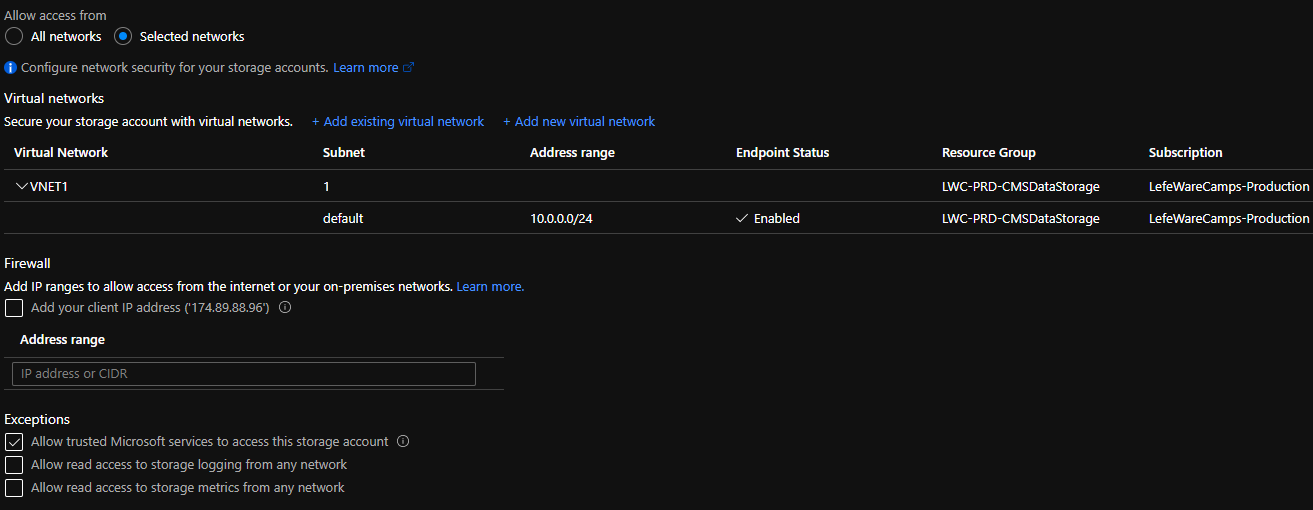
Access to any Azure storage account is prohibited by default. Access is given by role-based access control (RBAC) in Azure.

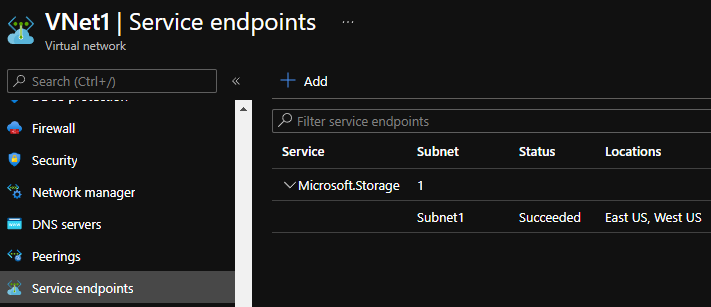
For additional control over storage accounts, we can also limit access to storage account to requests originating from specified IP addresses, IP ranges or from a list of subnets in an Azure VNet. An application that accesses a storage account when network rules are in effect still requires proper authorization for the request.

To set up an Azure Storage account inside a VNet:

1. Navigate to the storage account and select “Firewalls and Virtual Network Settings” blade.
2. To secure your storage account, you should first configure the rule to deny access to traffic from all networks (including internet traffic) on the public endpoint, by default. This can be done by selecting “Selected Networks” radio button
3. Click “Add existing virtual network” and chose the VNet and Subnet for which you wish to limit access to the storage account.
4. This will automatically create a “Service Endpoint” on the VNet
5. It is also possible to add Ip Address/Ranges from other clients outside the VNet.

\*In the firewalls and virtual network tab there is also an option called “Allow trusted Microsoft services to access this storage account” which is turned on by default. This allows connectivity to Azure Monitor Backups and Event Hubs.





Even if using firewall rules, storage accounts will still have a public endpoint that is accessible through the internet. You can also create Private Endpoints for your storage account, which assigns a private IP address from your VNet to the storage account, and secures all traffic between your VNet and the storage account over a private link.

### **Create and configure storage accounts**

An Azure storage account contains all Azure Storage data objects: blobs, files, queues, tables, and disks. The storage account provides a unique namespace for your Azure Storage data that is accessible from anywhere in the world over HTTP or HTTPS. A storage account name must be

* All lower case,
* 3-24 chars
* Contain only alphanumeric characters
* Be globally unique

Azure Storage offers several types of storage accounts. Each type supports different features and has its own pricing model. The types of storage accounts are:

* **General-purpose v2 accounts:** Basic storage account type for blobs, files, queues, and tables. Recommended for most scenarios using Azure Storage.
* **~~General-purpose v1 accounts:~~** Legacy account type for blobs, files, queues, and tables.
* **BlockBlobStorage accounts:** Storage accounts with premium performance characteristics for block blobs and append blobs. Recommended for scenarios with high transaction rates, or scenarios that use smaller objects or require consistently low storage latency.
* **FileStorage accounts:** Files-only storage accounts with premium performance characteristics. Recommended for enterprise or high performance scale applications.
* **~~BlobStorage accounts:~~** Legacy Blob-only storage accounts. Use general-purpose v2 accounts instead when possible.

For storage accounts, Microsoft charges for both amount of data stored in the account and outbound (egress) data transfer for every REST API transaction.

### **Generate shared access signature (SAS) tokens**

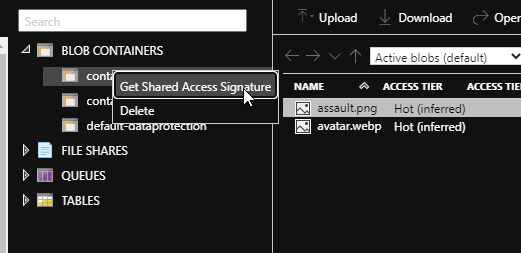
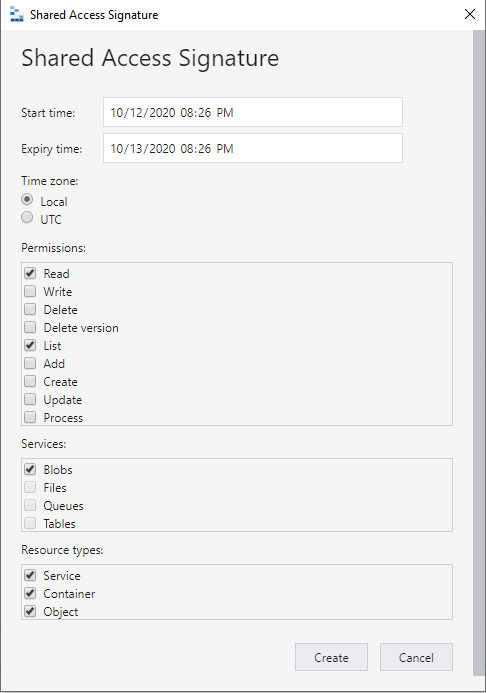
A Shared Access Signature (SAS) represents a way to grant limited, time bound access to individual storage account objects. Azure Storage supports three types of shared access signatures:

* **User Delegation SAS**: Secured with Azure AD credentials instead of the master key and also by the permissions specified for the SAS. A user delegation SAS applies to Blob storage only.
* **Account SAS:** An account SAS is secured with the storage account key. An account SAS delegates access to resources in one or more of the storage services
* **Service SAS:** Secured with the storage account key. A service SAS delegates access to a resource in only one of the Azure Storage services: Blob storage, Queue storage, Table storage, or Azure Files.

A SAS is essentially a URL that consists of the following components:

* Protocol: HTTP or HTTPS
* Address: Fully qualified address to the storage account
* Permissions: CRUD operations
* Time Interval: As long as the time interval has not expired, anyone on earth can use the SAS URI.
* Digital Signature

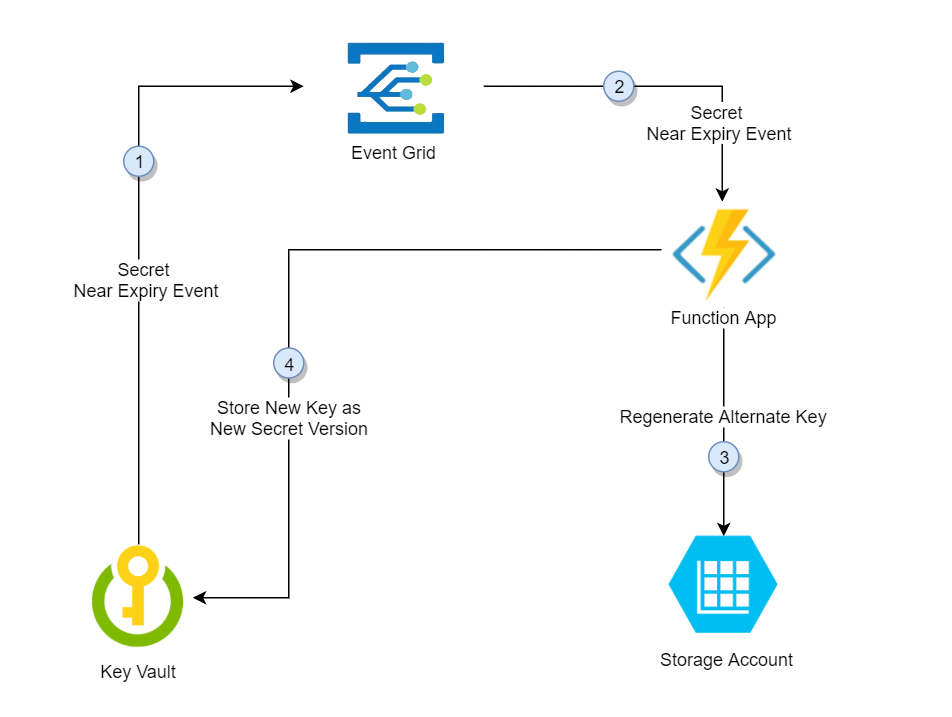
To Create a SAS:

1. In storage explorer, browse to the assets container and locate the container or file you with to create the SAS for
2. Right click an object and select “Get Shared Access Signature” from the menu:
3. Complete all SAS dialog boxes; add start & end times and permissions (CRUD):

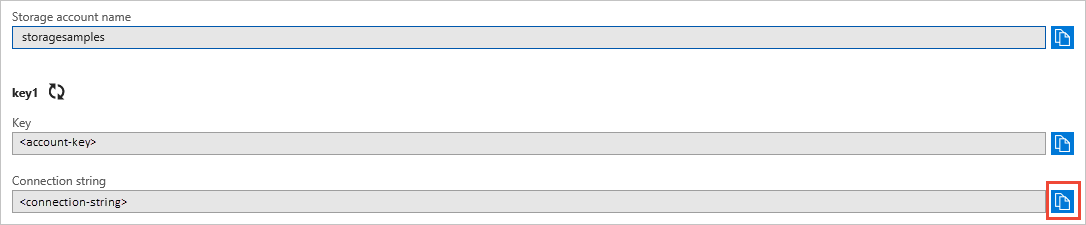
The resulting URL will look like the following where sig represents a digital signature:  
<https://lwcmedia.blob.core.windows.net/insights-logs-applicationgatewayaccesslog?sv=2019-10-10&st=2020-10-13T00%3A29%3A07Z&se=2020-10-13T00%3A30%3A00Z&sr=c&sp=rl&sig=QxC20mJ425bF2kV4UTz2fIu7PbMnjj1GAysUygtj3Jg%3D>

### **Manage access keys**

Azure generates two 512-bit storage account access keys to authorize access to data in a storage account via Shared Key authorization. It is recommended to use Azure Key Vault to manage and rotate access keys and also so that application can securely access keys in Key Vault, thus avoiding storing them within application code.



You can view and copy your account access keys with the Azure portal, PowerShell, or Azure CLI:



### **Configure Azure AD authentication for a storage account**

Each time data is accessed in a storage account, the client application makes a request over HTTP/HTTPS to Azure Storage that must be authorized to ensure appropriate permissions to access data in the account.

Azure Storage supports using Azure Active Directory (Azure AD) to authorize requests to blob data. With Azure AD, Azure RBAC can be used to grant permissions to a security principal, which may be a user, group, or application service principal. The security principal is authenticated by Azure AD to return an OAuth 2.0 token. The token can then be used to authorize a request against the Blob service. Authorizing requests against Azure Storage with Azure AD provides superior security and ease of use over Shared Key authorization

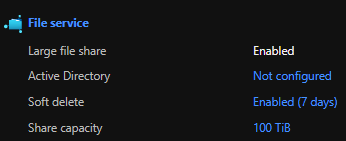
Azure RBAC provides a number of built-in roles for authorizing access to blob data using Azure AD and OAuth:

* Storage Blob Data Owner: Use to set ownership and manage POSIX access control for Azure Data Lake Storage Gen2.
* Storage Blob Data Contributor: Use to grant read/write/delete permissions to Blob storage resources.
* Storage Blob Data Reader: Use to grant read-only permissions to Blob storage resources.
* Storage Blob Delegator: Get a user delegation key to use to create a shared access signature that is signed with Azure AD credentials for a container or blob.

Built-in roles such as Owner, Contributor, and Storage Account Contributor permit a security principal to manage a storage account, but do not provide access to the blob or queue data within that account via Azure AD. However, if a role includes the Microsoft.Storage/storageAccounts/listKeys/action, then a user to whom that role is assigned can access data in the storage account via Shared Key authorization with the account access keys.

### **Configure access to Azure Files**

Azure Files supports identity-based authentication over Server Message Block (SMB) through two types of Domain Services: on-premises Active Directory Domain Services (AD DS) and Azure Active Directory Domain Services (Azure AD DS).



Neither identity-based authentication method is supported with Network File System (NFS) shares

## **Manage storage**

### **Import/Export from Azure job**

Azure Import service is used to securely import large amounts of data to Azure Blob storage and Azure Files by shipping disk drives to an Azure datacenter.

Azure Export service can also be used to transfer data from Azure Blob storage to disk drives and ship to your on-premises sites.

Supply your own disk drives and transfer data with the Azure Import/Export service. You can also use disk drives supplied by Microsoft. If you want to transfer data using disk drives supplied by Microsoft, you can use Azure Data Box Disk to import data into Azure. Microsoft ships up to 5 encrypted solid-state disk drives (SSDs) with a 40 TB total capacity.

The import/export processes can be created from within the Azure portal.

### **Install and use Azure Storage Explorer**

Microsoft Azure Storage Explorer is a standalone app that makes it easy to work with Azure Storage data on Windows, macOS, and Linux.

Storage Explorer provides several ways to connect to Azure resources:

* Sign in to Azure to access your subscriptions and their resources
* Attach to an individual Azure Storage resource – Through Account Name and Key or Shared Access Signature (SAS)

### **Copy data by using AZCopy**

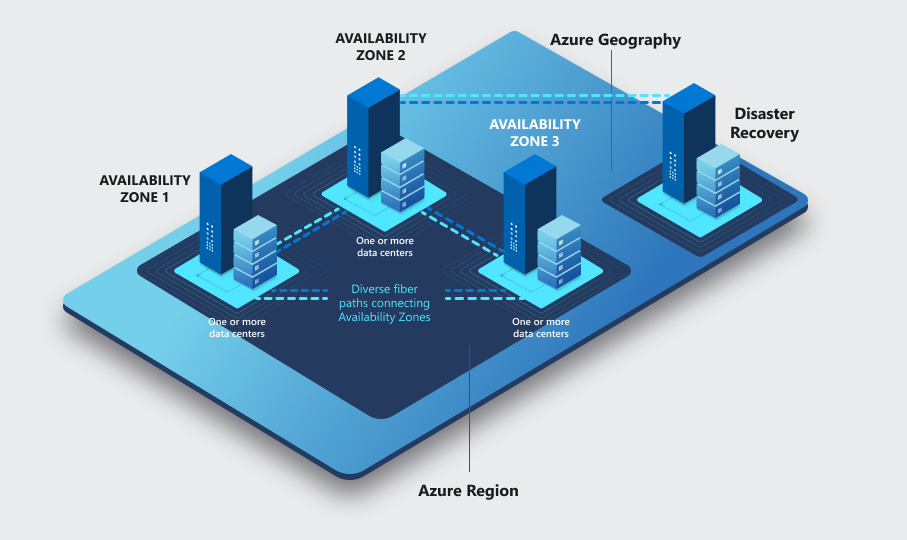
AzCopy is a command-line utility that you can use to copy blobs or files to or from a storage account. This article helps you download AzCopy, connect to your storage account, and then transfer files.

You can provide authorization credentials by using Azure Active Directory (AD), or by using a Shared Access Signature (SAS) token.

Example command:  
*azcopy copy 'C:\myDirectory\myTextFile.txt' 'https://mystorageaccount.blob.core.windows.net/mycontainer/myTextFile.txt'*

### **Implement Azure Storage replication**

Azure Storage stores multiple copies of data so that it is protected from planned and unplanned events, including transient hardware failures, network/power outages, and massive natural disasters. Redundancy ensures that storage accounts meet the Service-Level Agreement (SLA) for Azure Storage even in the face of failures.



Redundancy in the primary region:  
Data in an Azure Storage account is always replicated three times in the primary region using one of the following options:

* **Locally redundant storage (LRS)** copies your data synchronously three times within a single physical location in the primary region (not recommended for apps requiring HA) otherwise referred to as an Availability Set . LRS protects your data against server rack and drive failures. However, if a disaster such as fire or flooding occurs within the data center, all replicas of a storage account using LRS may be lost or unrecoverable
* **Zone-redundant storage (ZRS)** copies your data synchronously across three Azure availability zones in the primary region. For HA use ZRS in the primary region, and also replicate to a secondary region.

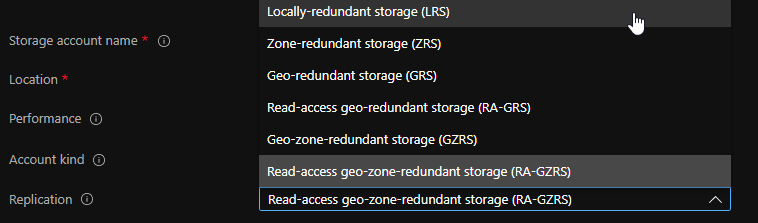
Redundancy in a secondary region:

For apps requiring HA, it is possible to copy the data in your storage account to a secondary region. If your storage account is copied to a secondary region, then your data is durable even in the case of a complete regional outage or a disaster in which the primary region isn't recoverable. When you create a storage account, you select the primary region for the account, the paired secondary region is determined based on the primary region, and can't be changed. Azure Storage offers two options for copying your data to a secondary region:

**Geo-redundant storage (GRS)** copies your data synchronously three times within a single physical location in the primary region using LRS. It then copies your data asynchronously to a single physical location in the secondary region.

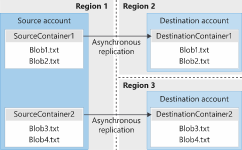
**Geo-zone-redundant storage (GZRS)** copies your data synchronously across three Azure availability zones in the primary region using ZRS. It then copies your data asynchronously to a single physical location in the secondary region.

With GRS or GZRS, the data in the secondary region isn't available for read or write access unless there is a failover to the secondary region. For read access to the secondary region, configure your storage account to use read-access geo-redundant storage (RA-GRS) or read-access geo-zone-redundant storage (RA-GZRS)



### **Configure blob object replication**

Object replication asynchronously copies block blobs between a source storage account and a destination account. The following diagram shows how object replication replicates block blobs from a source storage account in one region to destination accounts in two different regions.



Object replication requires that the following Azure Storage features are also enabled:

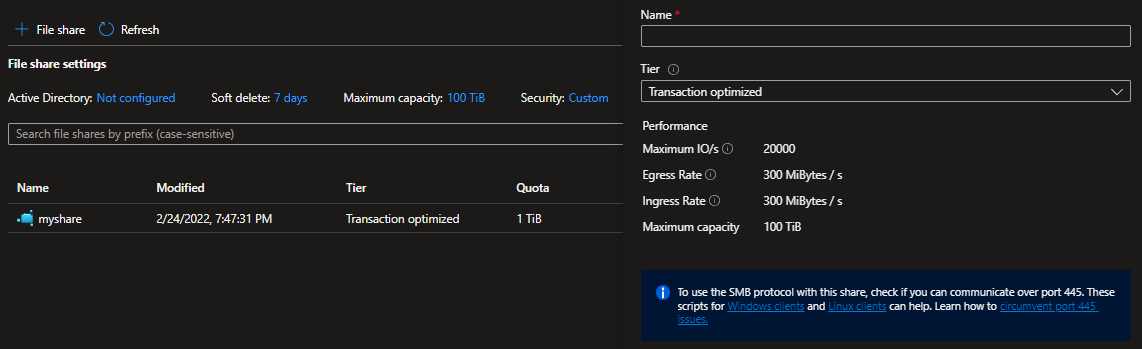
1. Change feed: Must be enabled on the source account.
2. Blob versioning: Must be enabled on both the source and destination accounts.

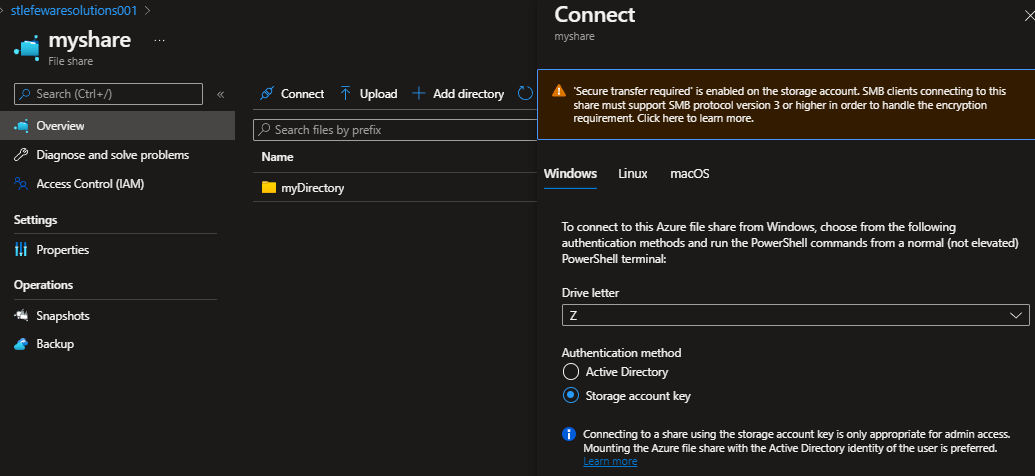
## **Configure Azure files and Azure Blob Storage**

### **Create an Azure file share**

Azure Files offers fully managed file shares in the cloud that are accessible via the industry standard Server Message Block (SMB) protocol or network file sharing (NFS) protocol. Azure Files file shares can be mounted concurrently by cloud or on-premises Windows, Linux or MacOS clients and docker containers

* Can be hosted on hard disk-based (HDD-based) hardware, or premium file shares, which are hosted on solid-state disk-based (SSD-based) hardware.
* Standard file shares offer locally-redundant (LRS), zone redundant (ZRS), geo-redundant (GRS), or geo-zone-redundant (GZRS) storage.
* Premium file shares are available with locally redundancy and zone redundancy in a subset of regions but not geo-redundancy.
* In local and zone redundant storage accounts, Azure file shares can span up to 100 TiB, however in geo- and geo-zone redundant storage accounts, Azure file shares can span only up to 5 TiB.





### **Create and configure Azure File Sync service**

Azure File Sync enables centralizing your organization's file shares in Azure Files. With cloud tiering enabled, frequently accessed files are cached on the local server and the least frequently accessed files are tiered to the cloud.

Azure file sync allows control of how much local disk space is used for caching. Tiered files can quickly be recalled on-demand, enabling cutting down on costs as only a fraction of data is stored on-premises.

### **Configure Azure Blob Storage**

Azure Binary Large OBject storage is optimized for storing massive amounts of unstructured data. Unstructured data is data that doesn't adhere to a particular data model or definition, such as text(logs) or binary data (images).

### **Configure storage tiers**

To manage costs of expanding storage needs, data can be organized based on how frequently it will be accessed and how long it will be retained. Azure storage offers different access tiers:

* If blob data is accessed frequently, set the blob storage to hot tier as this will give a discount on transaction costs.
* If data is accessed infrequently, set the blob storage to cold storage and Microsoft will give a discount on data storage.
* Data can also be set to “Archive”. This is for data that is only ever accessed on rare occasions. In order to access the data, you must “rehydrate” the archived blob and are charged to do so.

Storage accounts have a default access tier setting that indicates the online tier in which a new blob is created. The default access tier setting can be set to either Hot or Cool. Users can override the default setting for an individual blob when uploading the blob or changing its tier.

The default access tier for a new general-purpose v2 storage account is set to the Hot tier by default.

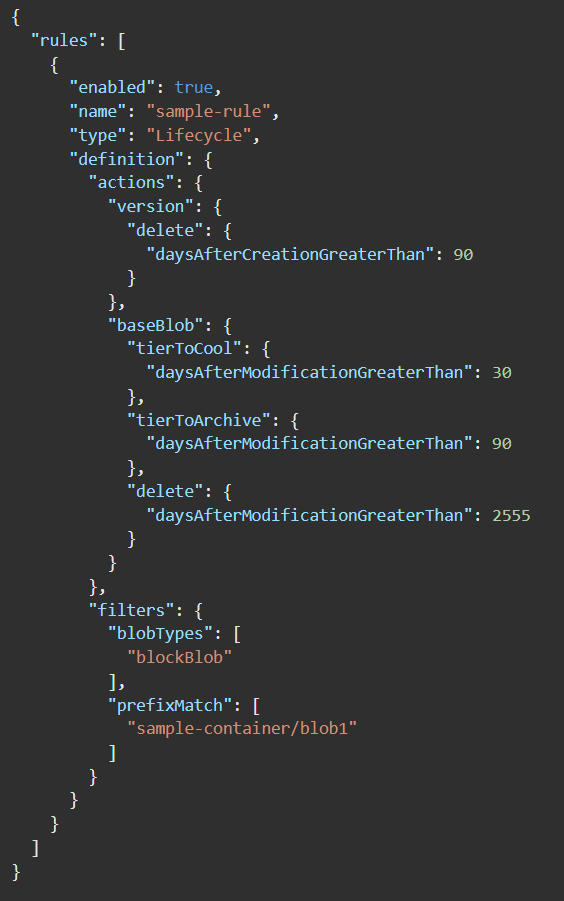
### **Configure blob lifecycle management**

Data sets have unique lifecycles in that some people access some data often, some data sets expire days or months after creation, while other data sets are actively read and modified throughout their lifetimes. Azure Storage lifecycle management offers a rule-based policy that can be used to transition blob data to the appropriate access tiers or to expire data at the end of the data lifecycle using a lifecycle management policy:

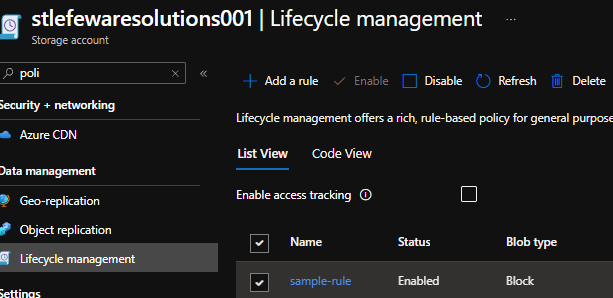
* Transition blobs from cool to hot immediately when they are accessed, to optimize performance.
* Transition current versions of a blob, previous versions of a blob, or blob snapshots to a cooler storage tier if these objects have not been accessed or modified for a period of time, to optimize cost.
* Delete current versions of a blob, previous versions of a blob, or blob snapshots at the end of their lifecycles.
* Define rules to be run once per day at the storage account level.
* Apply rules to containers or to a subset of blobs, using name prefixes or blob index tags as filters.

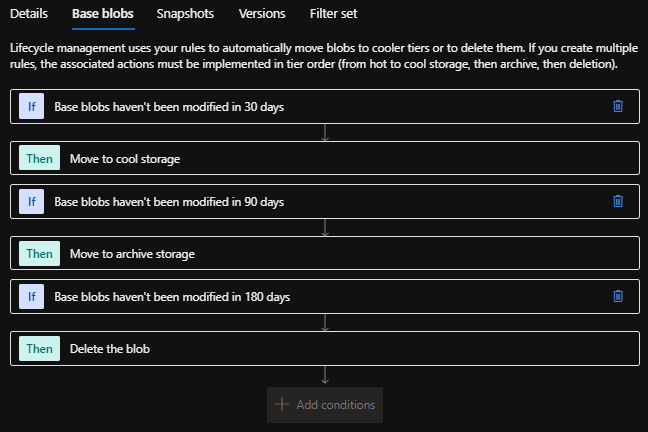
A lifecycle management policy is a collection of rules in a JSON document. The following sample JSON shows a complete rule definition made up actions and filters with the following policies:

* Tier blob to cool tier 30 days after last modification
* Tier blob to archive tier 90 days after last modification
* Delete blob 2,555 days (seven years) after last modification
* Delete previous versions 90 days after creation



A lifecycle management policy can be added, edited, or removed with the Azure portal, PowerShell, Azure CLI.





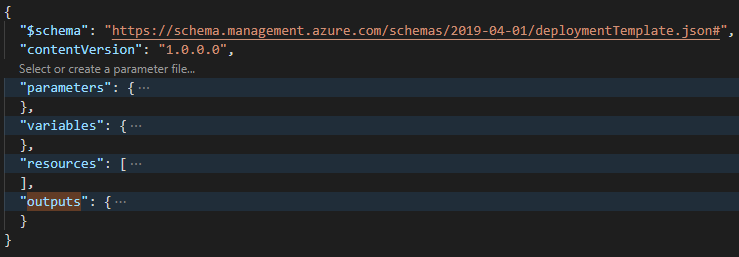
# **Deploy and manage Azure compute resources (20–25%)**

## **Automate deployment of virtual machines (VMs) by using Azure Resource Manager templates**

### **Modify an Azure Resource Manager template**

Use the following steps to create/modify an ARM Template

1. Create a file named azuredeploy.json.
2. Copy and paste the following JSON into the file:



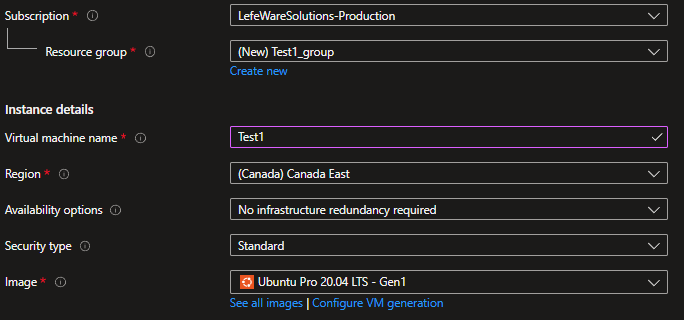
The JSON file has these elements:

* **$schema:** Specifies the location of the JSON schema file. The schema file describes the properties that are available within a template. For example, the schema defines resources as one of the valid properties for a template.
* **contentVersion:** Specifies the version of the template (such as 1.0.0.0). Use this value to document significant changes in your template. When deploying resources using the template, this value can be used to make sure that the right template is being used.
* **resources:** Contains the resources you want to deploy or update.
* **variables:** Define values that are reused in your templates. They can be constructed from parameter values.
* **outputs:** Return values from the deployed resources.
* **parameters:** Provide values during deployment that allow the same template to be used with different environments.
* **User-defined functions:** Create customized functions that simplify your template.

### **Configure a virtual hard disk (VHD) template**

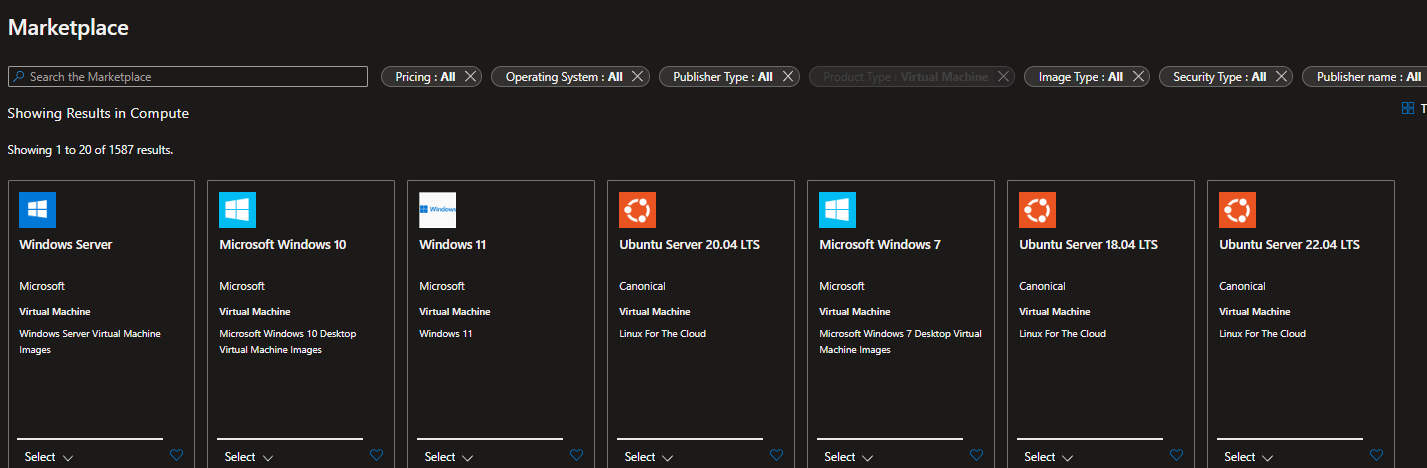
A virtual hard disk is similar to a physical hard disk with many advantages such as physical security, high availability, scalability etc. A VHD can hold anything that you can store on a physical hard disk such as hosting the operating system needed to run the VM, as well as any other software and user data.

When creating a virtual machine in Azure, you must specify a virtual machine image to use. This image is basically a template that contains the operating system and (optionally) other preconfigured software.



Azure enables deploying Windows and Linux virtual machines from standard virtual machine images available in Azure Marketplace:

* Various versions of Windows Server, optionally with SQL Server installed.
* Linux variants with software such as MySQL, MongoDB, Cassandra,etc



Azure uses this image to create a new virtual hard disk (VHD) from which it can start your virtual machine. You can then customize the virtual machine by configuring and installing additional applications, according to your requirements.

Azure also allows creating your own customized images that can be used to deploy virtual machine to match customized requirements in one of two ways:

* From scratch using Hyper-V, and then uploading to Azure for VMs to be used
* By extending an existing VM, such as from the Azure Marketplace. The image provides the operating system and base functionality. Then add your own software, operating system updates, and other packages as required.

#### ***Generalized Image:***

When creating a VM from a custom image, the operating system data is updated with several items, including username and credentials, logs etc. These items must be reset back to a default state before using the image to create other VMs. Otherwise, you might end up with multiple virtual machines that have the same identities. The process of resetting this data is called generalization, and the result is a generalized image.

The tools for preparing a virtual machine for generalization vary according to the operating system that's being installed and configured. For Windows, use the Microsoft System Preparation (Sysprep) tool. For Linux, use the Windows Azure Linux Agent (waagent) tool.

#### ***Specialized Image:***

A specialized virtual image is a copy of a live virtual machine after it has reached a specific state. For example, a specialized image might contain a copy of the configured operating system, software, user accounts, databases, connection information, and other data for your system.

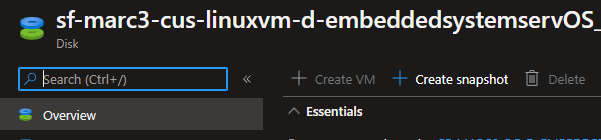
You can use a specialized virtual image as a backup of your system at a particular point in time. If you need to recover after a catastrophic failure, or you need to roll back the virtual machine, you can restore your virtual machine from this image.

#### ***VM Snapshot:***

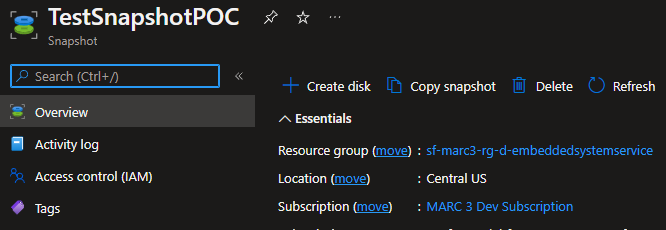
It is possible to create a VM snapshot from an existing VM that creates a new managed disk from the snapshot to use as a template. Creating a snapshot takes a few more steps, but it preserves the original VHD and provides you with a fallback.

To create a snapshot from an existing disk:

1. From the Azure portal, select the disk that you would like to use.
2. From the menu of the disk page at the top, select Create snapshot.



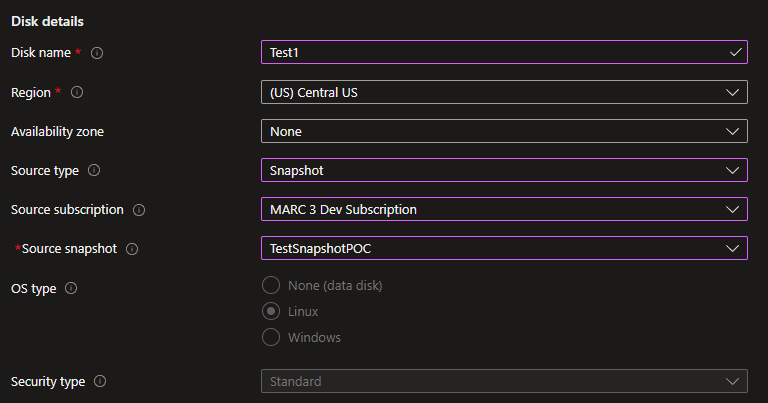
1. Choose a Resource group and name for the snapshot.
2. For Snapshot type, choose either Full or Incremental.
   1. Full: make a complete read-only copy of the selected disk.
   2. Incremental: save on storage costs by making a partial copy of the disk based on the difference between the last snapshot.
3. For Storage type, choose Standard HDD, Premium SSD, or Zone-redundant storage.
4. When you're done, select Create to create the snapshot.



### **Deploy from a template**

To Create a disk from a snapshot

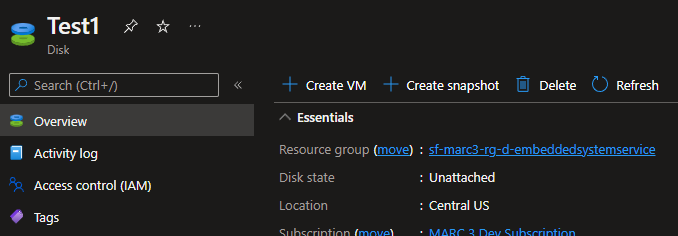
1. From the Azure Portal, Create a managed Disk.
2. Choose a Resource group and name for the disk.
3. In Source type, ensure Snapshot is selected.
4. In the Source snapshot drop-down, select the snapshot you want to use.
5. For Size, choose either Standard (HDD) or Premium (SSD) storage.



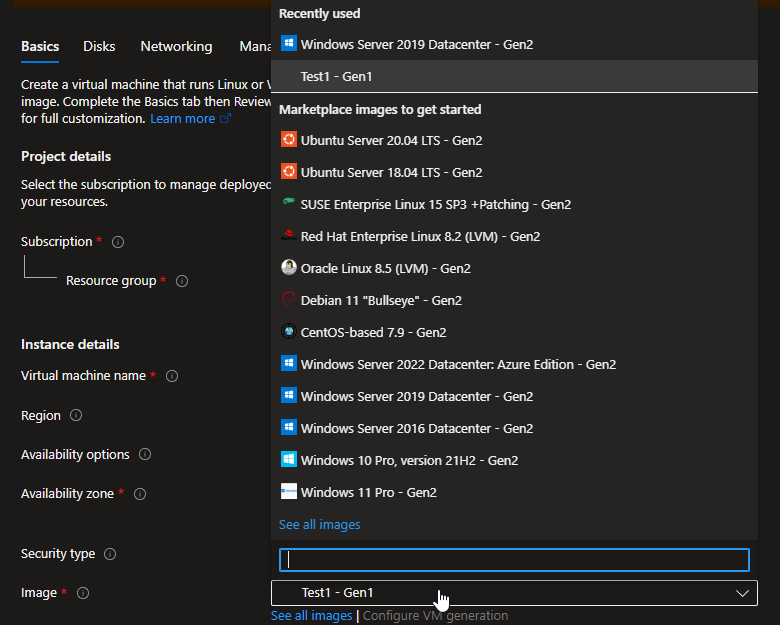
1. Make any other adjustments as needed and then select Create to create the disk.

Create a VM from a disk

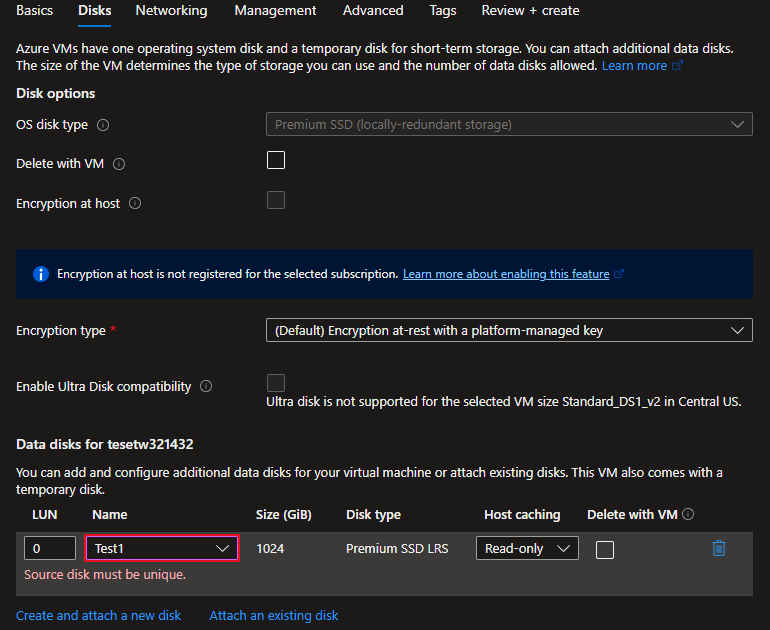
1. From the Azure portal, select Disks to display the list of available disks.
2. In the disk Overview page, ensure that DISK STATE is listed as Unattached.
   1. If it isn't, you might need to either detach the disk from the VM or delete the VM to free up the disk.
3. In the menu at the top of the page, select Create VM.



1. On the Basics page for the new VM, enter a Virtual machine name and either select an existing Resource group or create a new one.
2. Under “Image” it should already be populated with disk created in the previous step:



1. Select a VM size row and then choose Select.
2. On the Disks page, you may notice that the "OS Disk Type" cannot be changed. This preselected value is configured at the point of Snapshot or VHD creation and will carry over to the new VM.

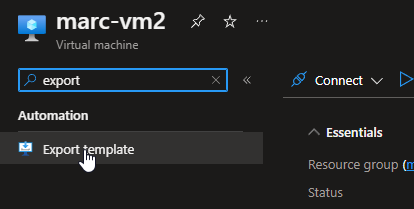


### **Save a deployment as an Azure Resource Manager template**

When creating a VM in Azure using the portal or PowerShell, a Resource Manager template is automatically created and can be used to quickly duplicate a deployment. The template contains information about all of the resources in a resource group. For a virtual machine, this means the template contains everything that is created in support of the VM in that resource group, including the networking resources.

To save an ARM template from portal:

1. Log in to the Azure portal and select the virtual machine (or resource group) from the list.
2. Select Export template.



1. Select Download from the menu at the top and save the .zip file to your local computer.
2. Open the .zip file and extract the files to a folder. The .zip file contains:

* parameters.json
* template.json

To save an ARM template from powershell:  
Export-AzResourceGroup -ResourceGroupName "myResourceGroup" -Path "C:\users\public\downloads"

### **Deploy virtual machine extensions**

Extensions are small applications that provide post-deployment configuration and automation on Azure VMs such as covering VM configuration, monitoring, security, and utility applications. Publishers take an application, wrap it into an extension, and simplify the installation.

For example, Azure Backup, amongst other offerings, provides support for backing up workloads such as SQL Server running in Azure VMs. Since the SQL application is running within an Azure VM, the backup service needs permission to access the application and fetch the necessary details. To do that, Azure Backup installs the AzureBackupWindowsWorkload extension on the VM, in which the SQL Server is running, during the registration process triggered by the user.

Deploying an extension via powershell:  
$myVM = Get-AzVM -ResourceGroupName <VMRG Name> -Name <VMName>

Register-AzRecoveryServicesBackupContainer -ResourceId $myVM.ID -BackupManagementType AzureWorkload -WorkloadType MSSQL -VaultId $targetVault.ID -Force

There are 3 different ways of deploying an extension to VMSS via ARM template:

* Keep extension as a separate resource outside of VMSS definition
* Add extension to extensionProfile of VMSS
* Add extension in a child resources section of VMSS



## **Configure VMs**

### **Configure Azure Disk Encryption**

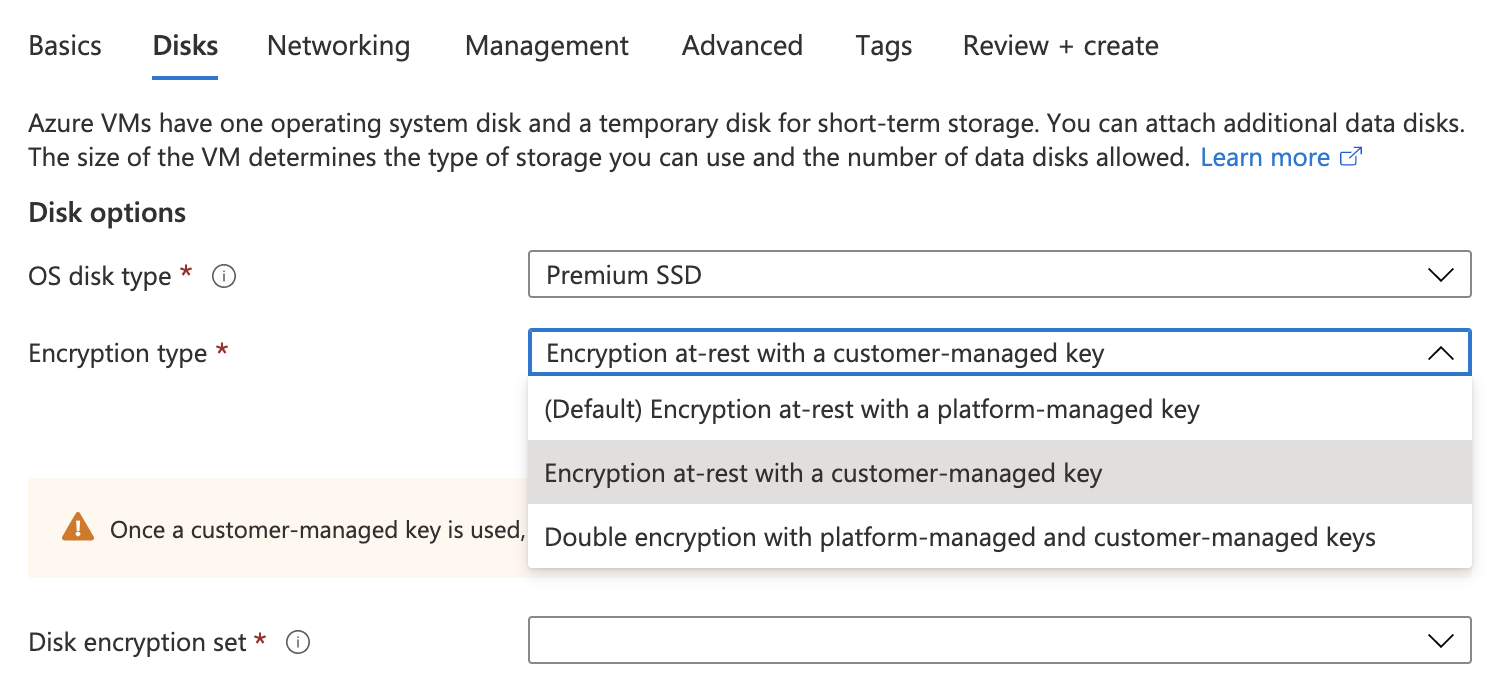
Managed disks offer two different kinds of encryption:

Server-Side Encryption (SSE):

Encryption-at-rest is designed to prevent the attacker from accessing the unencrypted data by ensuring the data is encrypted when on disk. If an attacker obtains a hard drive with encrypted data but not the encryption keys, the attacker must defeat the encryption to read the data.

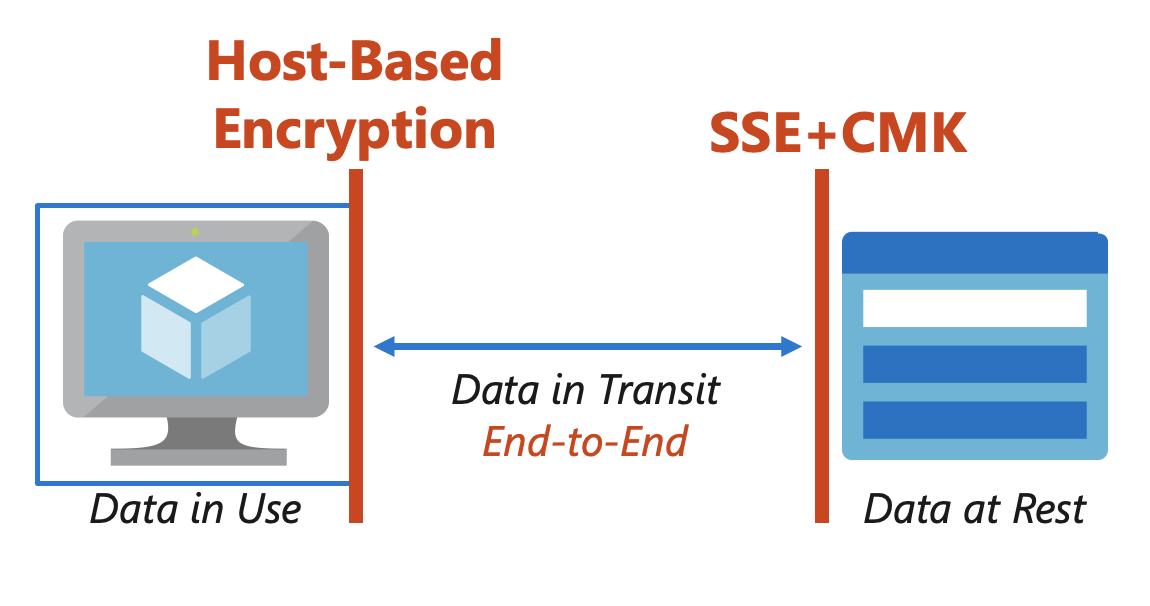
Azure Managed Disks also allow managing encryption at the level of each managed disk, with customer managed keys. Server-side encryption for managed disks with customer-managed keys integrates with Azure Key Vault. You can either import RSA keys in Key Vault or generate new RSA keys in Azure Key Vault.

Azure provides encryption-at-rest and is enabled by default for all managed disks, snapshots, and images using platform-managed encryption keys. Data in Azure managed disks is encrypted transparently using 256-bit AES encryption, one of the strongest block ciphers available, and is FIPS 140-2 compliant.



**Azure Disk Encryption (ADE):** ADE is enabled by the OS and data disks for your VMs. For Windows, the drives are encrypted using industry-standard BitLocker encryption technology. For Linux, the disks are encrypted using DM-Crypt technology. The ADE process is also integrated with Azure Key Vault to allow controlling and managing the disk encryption keys. Azure Disk Encryption can be enabled and managed through the Azure CLI and Azure PowerShell.

**Encryption at Host:** A host is the physical server that a VM runs on, and multiple VMs run on a single host. Encryption at host ensures that data stored on the VM host hosting your VM is encrypted at rest and flows encrypted to the Storage clusters.



### **Move VMs from one resource group to another**

**Using Portal:**

1. Choose the resource group containing the VM that you would like to move.
2. At the top of the page for the resource group, select Move and then select Move to another resource group. The Move resources page opens.
3. Select each of the resources to move. In most cases, you should move all of the related resources that are listed.
4. Select an existing Resource group, or enter a name to have a new resource group created.
5. When you are done, select that you understand that new resource IDs will be created and that the new IDs will need to be used with the VM after it is moved, and then select OK.

Using Powershell:

Move-AzResource -DestinationResourceGroupName $destinationRG -ResourceId $resourceIds

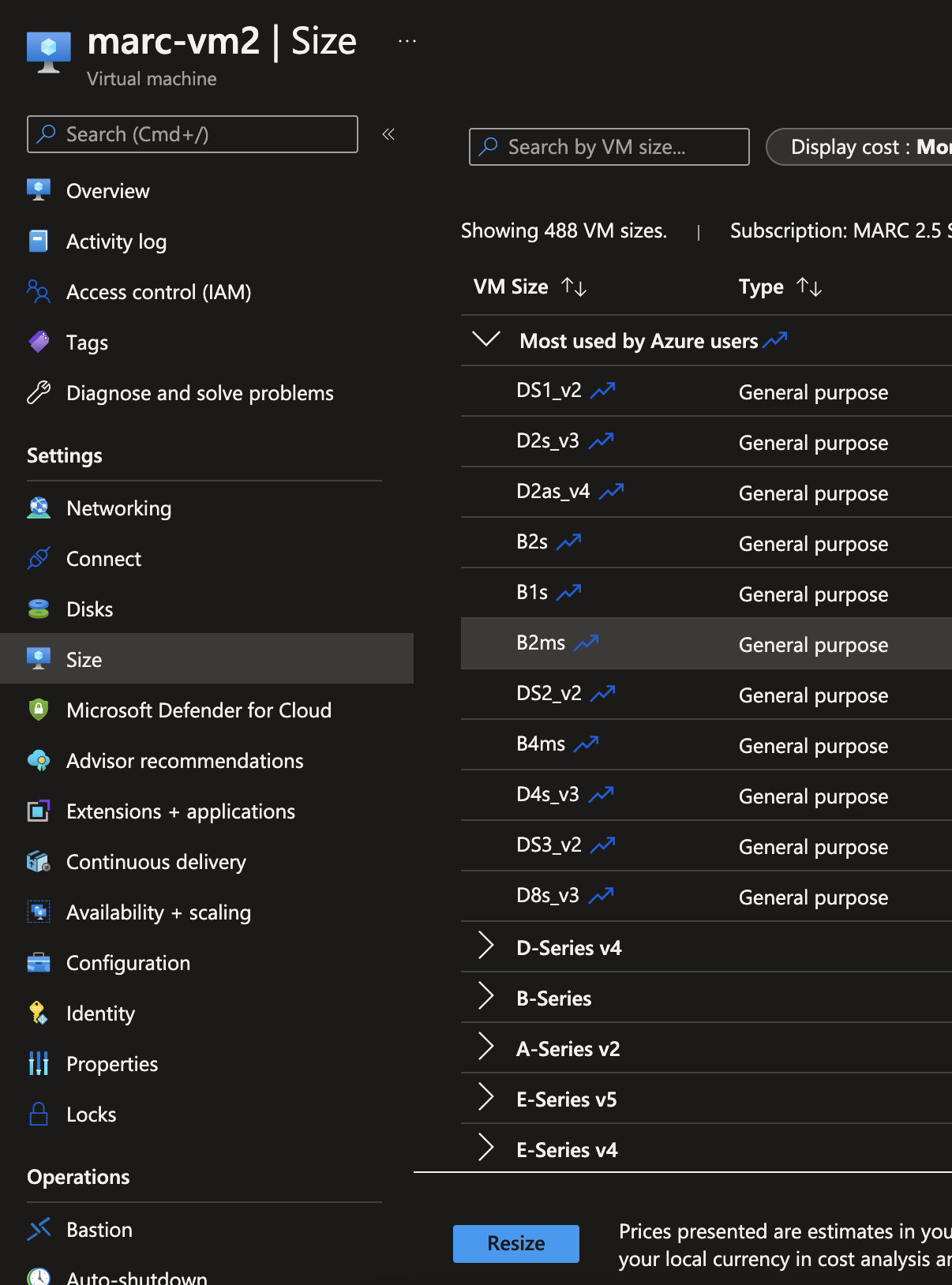
Where resourceIds is a list of comma separated resource Ids

### **Manage VM sizes & add data disks**

After you create a VM, you can scale the VM up or down by changing the VM size. In some cases, you must deallocate the VM first.

From Portal

1. Navigate to the Azure Portal virtual machine page.
2. In the left menu, select Size.
3. Pick a new size from the list of available sizes and then select Resize.



From Powershell:

$vm = Get-AzVM -ResourceGroupName $resourceGroup -VMName $vmName

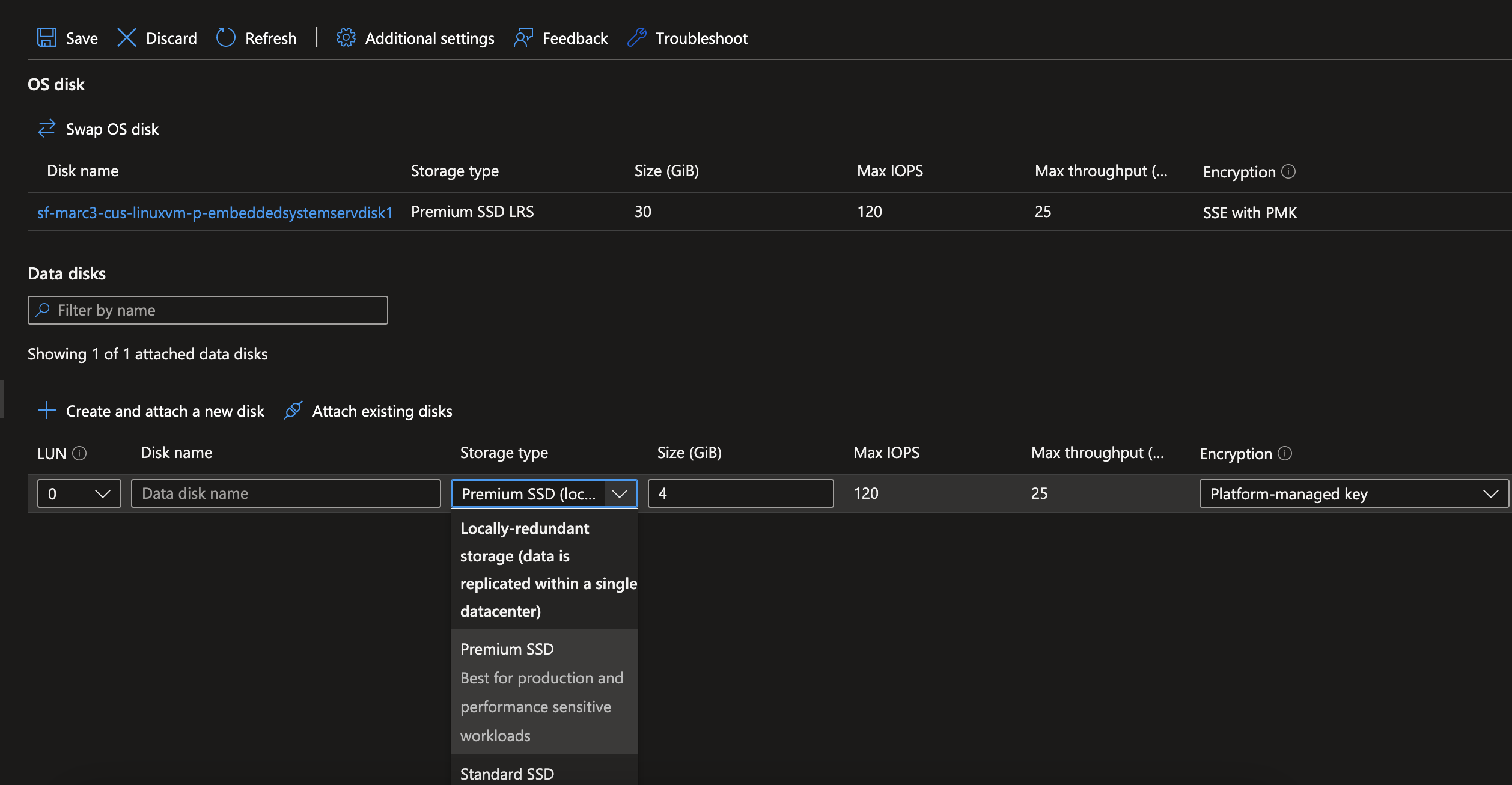
$vm.HardwareProfile.VmSize = "<newVMsize>"

Update-AzVM -VM $vm -ResourceGroupName $resourceGroup

The size of the VM determines how many data disks you can attach. To use premium SSDs, you'll need a premium storage-enabled VM type, like the DS-series or GS-series virtual machine.

From Portal:

1. On the Virtual machine pane, select Disks.
2. On the Disks pane, select Create and attach a new disk.
3. In the drop-downs for the new disk, make the selections you want, and name the disk.



1. Select Save to create and attach the new data disk to the VM.

PowerShell:

$rgName = 'myResourceGroup'

$vmName = 'myVM'

$location = 'East US'

$storageType = 'Premium\_LRS'

$dataDiskName = $vmName + '\_datadisk1'

$diskConfig = New-AzDiskConfig -SkuName $storageType -Location $location -CreateOption Empty -DiskSizeGB 128

$dataDisk1 = New-AzDisk -DiskName $dataDiskName -Disk $diskConfig -ResourceGroupName $rgName

$vm = Get-AzVM -Name $vmName -ResourceGroupName $rgName

$vm = Add-AzVMDataDisk -VM $vm -Name $dataDiskName -CreateOption Attach -ManagedDiskId $dataDisk1.Id -Lun 1

Update-AzVM -VM $vm -ResourceGroupName $rgName

### **Configure networking**

### **Redeploy VMs**

If issues arise with Remote Desktop (RDP) connection or application access to Windows-based Azure virtual machine (VM), redeploying the VM may help. When redeploying a VM, Azure will shut down the VM, move the VM to a new node within the Azure infrastructure, and then power it back on, retaining all configuration options and associated resources.

From Powershell:

Set-AzVM -Redeploy -ResourceGroupName "myResourceGroup" -Name "myVM"

From Portal:

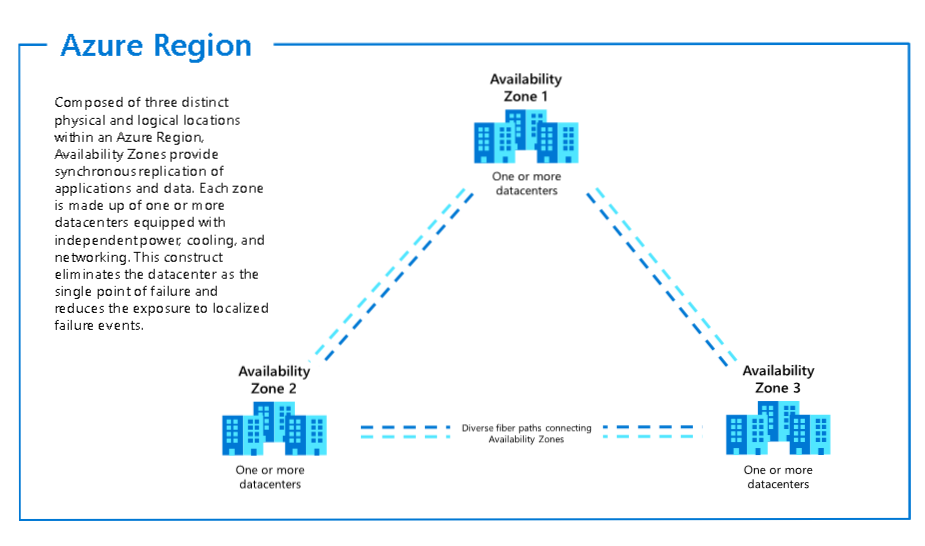
1. Select the VM to redeploy, then select the Redeploy button in the Settings blade.
2. Confirm
3. The Status of the VM changes to Updating as the VM prepares to redeploy
4. The Status then changes to Starting as the VM boots up on a new Azure host
5. After the VM finishes the boot process, the Status then returns to Running, indicating the VM has been successfully redeployed:

### **Configure high availability**

Workloads can be spread across different virtual machines to gain high throughput, performance, and to create redundancy in case a VM is impacted due to an update or other event.

**Availability Zones:** An Availability Zone is a physically separate zone, within an Azure region. There are three Availability Zones per Azure region. Each Availability Zone has a distinct power source, network, and cooling. By architecting solutions to use replicated VMs in zones, you can protect your apps and data from the loss of a datacenter. If one zone is compromised, then replicated apps and data are instantly available in another zone using zone redundant replication. With Availability Zones, Azure offers industry best 99.99% Virtual Machine (VM) uptime service-level agreement (SLA)

**Azure Region ====> Azure Availability Zone**



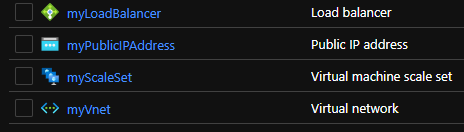
An Availability Zone in an Azure region is a combination of a fault domain and an update domain. For example, if you create three or more VMs across three zones in an Azure region, your VMs are effectively distributed across three fault domains and three update domains. The Azure platform recognizes this distribution across update domains to take care that VMs in different zones are not updated at the same time.

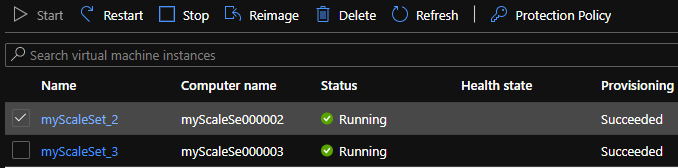
**Virtual Machines Scale Sets:** Azure virtual machine scale sets let you create and manage a group of load balanced VMs. The number of VM instances can automatically increase or decrease in response to demand or a defined schedule. Scale sets provide high availability to your applications, and allow you to centrally manage, configure, and update many VMs.

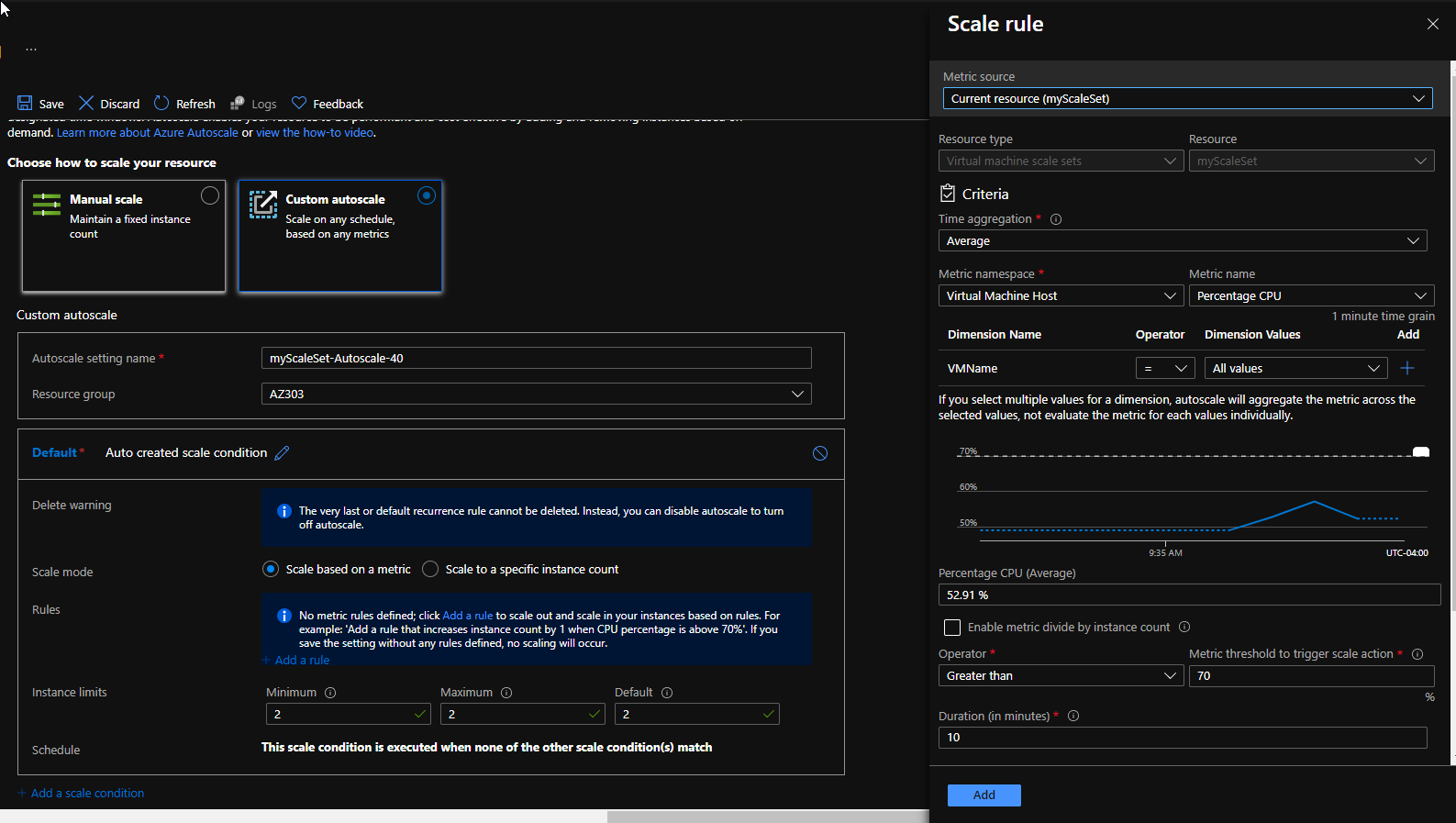
Scale sets provide the following key benefits:

* **Easy to create and manage multiple VMs:** All VM instances are created from the same base OS image and configuration. Allows easy management of hundreds of VMs without additional configuration tasks or network management.
* **Provides high availability and application resiliency:** VMs in a scale set can be deployed across multiple update domains and fault domains to maximize availability and resilience to outages due to data center outages, and planned or unplanned maintenance events. If a VM instance has a problem, customers continue to access the app through one of the other VM instances with minimal interruption. Virtual machines in a scale set can also be deployed into a single Availability zone, or regionally.
* **Allows your application to automatically scale as resource demand changes:** To match customer demand, scale sets can automatically increase the number of VM instances as application demand increases, then reduce the number of VM instances as demand decreases to reduce cost.
* **Works at large-scale:** Scale sets support up to 1,000 VM instances or 600VM if you use your own custom image.

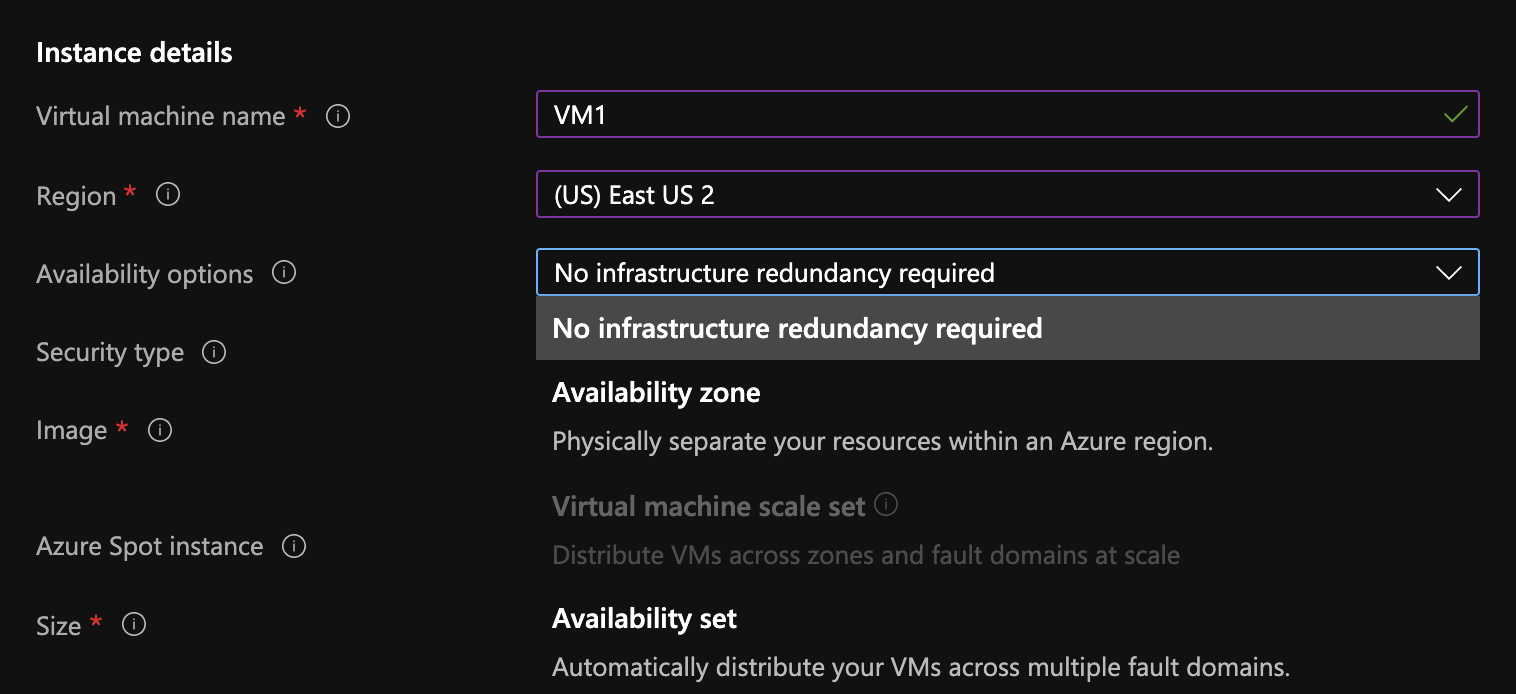
Shared VMSS Infrastructure:



Load Balanced VM Instances:  


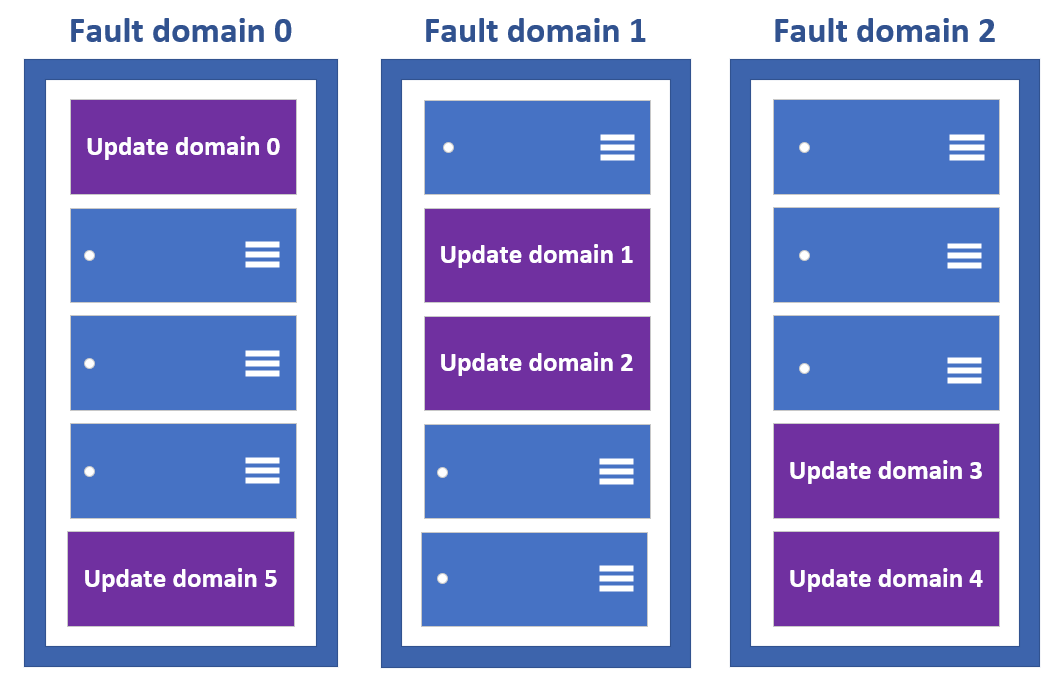
Autoscale Rules (CPU % > 70% for 10 minutes):  


**Availability sets:** An availability set is a logical grouping of VMs that allows Azure to understand how your application is built to provide for redundancy and availability. Azure recommends that two or more VMs are created within an availability set to provide for a highly available application and to meet the 99.95% Azure SLA.



Each VM in your availability set is assigned an update domain and a fault domain by the underlying Azure platform. Each availability set can be configured with up to three fault domains and twenty update domains.

**Update Domains:** Update domains indicate groups of VMs and underlying physical hardware that can be rebooted at the same time. When more than five VMs are configured within a single availability set, the sixth VM is placed into the same update domain as the first VM, the seventh in the same update domain as the second virtual machine, and so on. The order of update domains being rebooted may not proceed sequentially during planned maintenance, but only one update domain is rebooted at a time. A rebooted update domain is given 30 minutes to recover before maintenance is initiated on a different update domain.

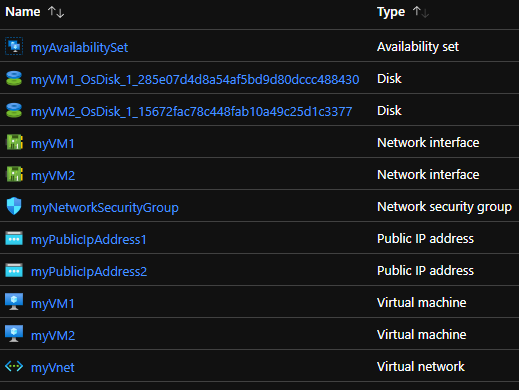


Fault Domains:

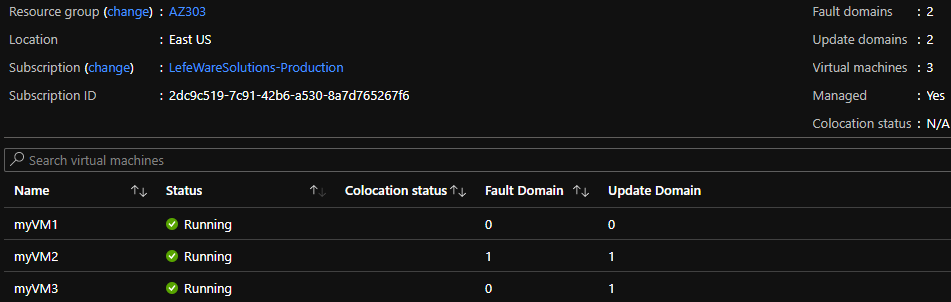
Fault domains define the group of VMs that share a common power source and network switch. By default, the virtual machines configured within your availability set are separated across up to three fault domains. While placing your virtual machines into an availability set does not protect your application from operating system or application-specific failures, it does limit the impact of physical hardware failures, network outages, or power interruptions.

VMs are also aligned with disk fault domains. This alignment ensures that all the managed disks attached to a VM are within the same fault domains.

Unlike VMSS, Availability Sets don’t share Infra, load balancer etc:



3 Virtual Machines, spread out over 2 fault domains and 2 update domains:

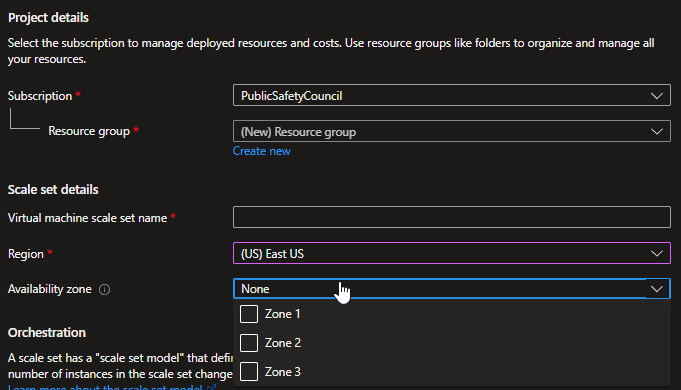


### **Deploy and configure virtual machine scale sets**

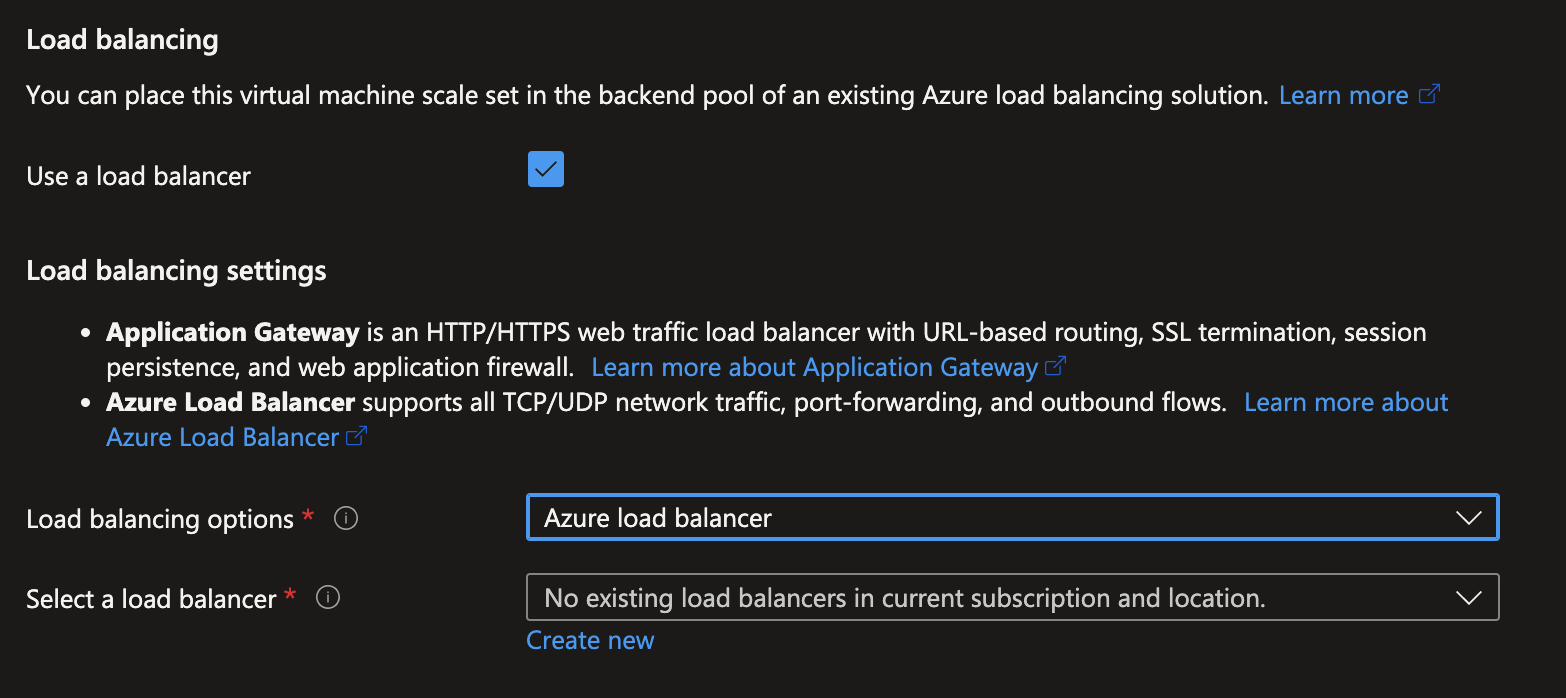
You can deploy a scale set with a Windows Server image or Linux image such as RHEL, CentOS, Ubuntu, or SLES.

From Portal:

1. Select Virtual machine scale sets, Select Create on the Virtual machine scale sets
2. In the Basics tab, under Project details:
   1. Select correct subscription is selected and select a resource group from resource group list.
   2. Provide a name for the scale set.
   3. Select a region that is close to your area. Only certain regions have availability zones



1. Under Orchestration
   1. Ensure the Uniform option is selected for Orchestration mode.
      1. In uniform orchestration mode, you define a virtual machine model and Azure will generate identical instances based on that model.
   2. Select a marketplace image for Image. I.e Ubuntu Server 18.04 LTS.
   3. Enter your desired username, and select which authentication type you prefer.
   4. A Password must be at least 12 characters long and meet three out of the four following complexity requirements: one lower case character, one upper case character, one number, and one special character. For more information, see username and password requirements.
   5. If you select a Linux OS disk image, you can instead choose SSH public key.
2. Leave the defaults for the Disks page.
3. On the Networking page, under Load balancing, select the Use a load balancer option to put the scale set instances behind a load balancer.
   1. In Load balancing options, select Azure load balancer.
   2. Create a new load balancer
   3. For Select a backend pool, select Create new, type myBackendPool, then select Create.



1. After it passes validation, select Create to deploy the scale set.

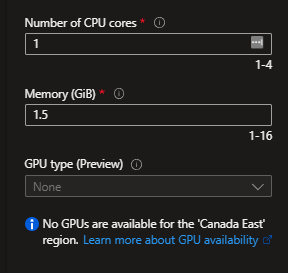
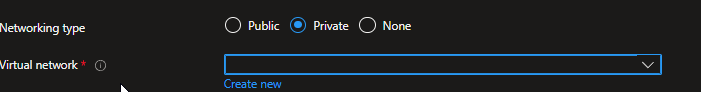
From Powershell:

PowerShell\VMs\CreateVMSS.ps1

## **Create and configure containers**

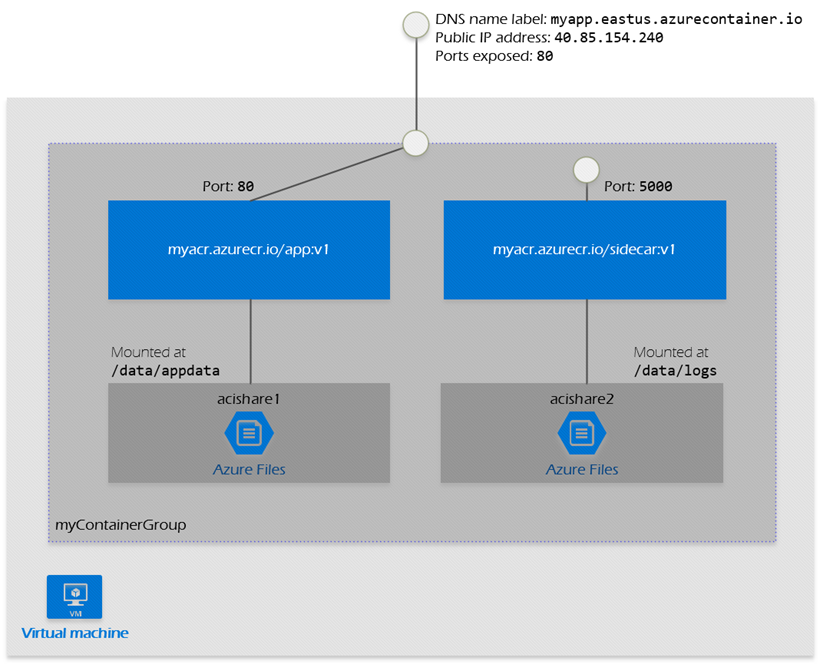
### **Configure sizing and scaling for Azure Container Instances**

Azure Container Instances offers the fastest and simplest way to run a container in Azure, without having to manage any VMs and without having to adopt a higher-level service.

* ACI can run simple applications, task automation, and build jobs.
* Can start containers in Azure in seconds
* Enables exposing container groups directly to the internet with an IP address and a FQDN.
  + Can specify a custom DNS name label so the application is reachable at customlabel.azureregion.azurecontainer.io.
* Allows exact specifications of CPU cores and memory
  + billed by the second
  + 
* Can schedule both Windows and Linux containers
* Deployment of container instances into an Azure virtual network subnet. network,
  + container instances can communicate securely with other resources in the VNET, including those that are on premises (through VPN gateway or ExpressRoute).
  + 

### **Configure container groups for Azure Container Instances**

ACI supports scheduling of multi-container groups that share a host machine, local network, storage, and lifecycle. This enables combing the main app container with other supporting role containers, such as logging sidecars. It's similar in concept to a pod in Kubernetes.



There are two common ways to deploy a multi-container group: use a Resource Manager template or a YAML file.

A Resource Manager template is recommended when you need to deploy additional Azure service resources (for example, an Azure Files share) when you deploy the container instances.

Due to the YAML format's more concise nature, a YAML file is recommended when your deployment includes only container instances.

### **Configure storage for Azure Kubernetes Service (AKS)**

### **Configure scaling for AKS**

### **Configure network connections for AKS**

### **Upgrade an AKS cluster**

## **Create and configure Azure App Service**

### **Create an App Service plan**

### ** configure scaling settings in an App Service plan**

### ** create an App Service**

### ** secure an App Service**

### ** configure custom domain names**

### ** configure backup for an App Service**

### ** configure networking settings**

### ** configure deployment settings**

# **Configure and manage virtual networking (25–30%)**

## **Implement and manage virtual networking**

 create and configure virtual networks, including peering

 configure private and public IP addresses

 configure user-defined network routes

 implement subnets

 configure endpoints on subnets

 configure private endpoints

 configure Azure DNS, including custom DNS settings and private or public DNS zones

## **Secure access to virtual networks**

 create security rules

 associate a network security group (NSG) to a subnet or network interface

 evaluate effective security rules

 implement Azure Firewall

 implement Azure Bastion

## **Configure load balancing**

 configure Azure Application Gateway

 configure an internal or public load balancer

 troubleshoot load balancing

## **Monitor and troubleshoot virtual networking**

 monitor on-premises connectivity

 configure and use Azure Monitor for Networks

 use Azure Network Watcher

 troubleshoot external networking

 troubleshoot virtual network connectivity

## **Integrate an on-premises network with an Azure virtual network**

 create and configure Azure VPN Gateway

 create and configure Azure ExpressRoute

 configure Azure Virtual WAN Monitor and back up

# **Azure resources (10–15%)**

## **Monitor resources by using Azure Monitor**

 configure and interpret metrics

 configure Azure Monitor logs

 query and analyze logs

 set up alerts and actions

 configure Application Insights

## **Implement backup and recovery**

 create a Recovery Services vault

 create a Backup vault

 create and configure backup policy

 perform backup and restore operations by using Azure Backup

 perform site-to-site recovery by using Azure Site Recovery

 configure and review backup reports