AZURE 303

1- Implement and Monitor Infrastructure

Azure Overview

Subscription and Services Layer

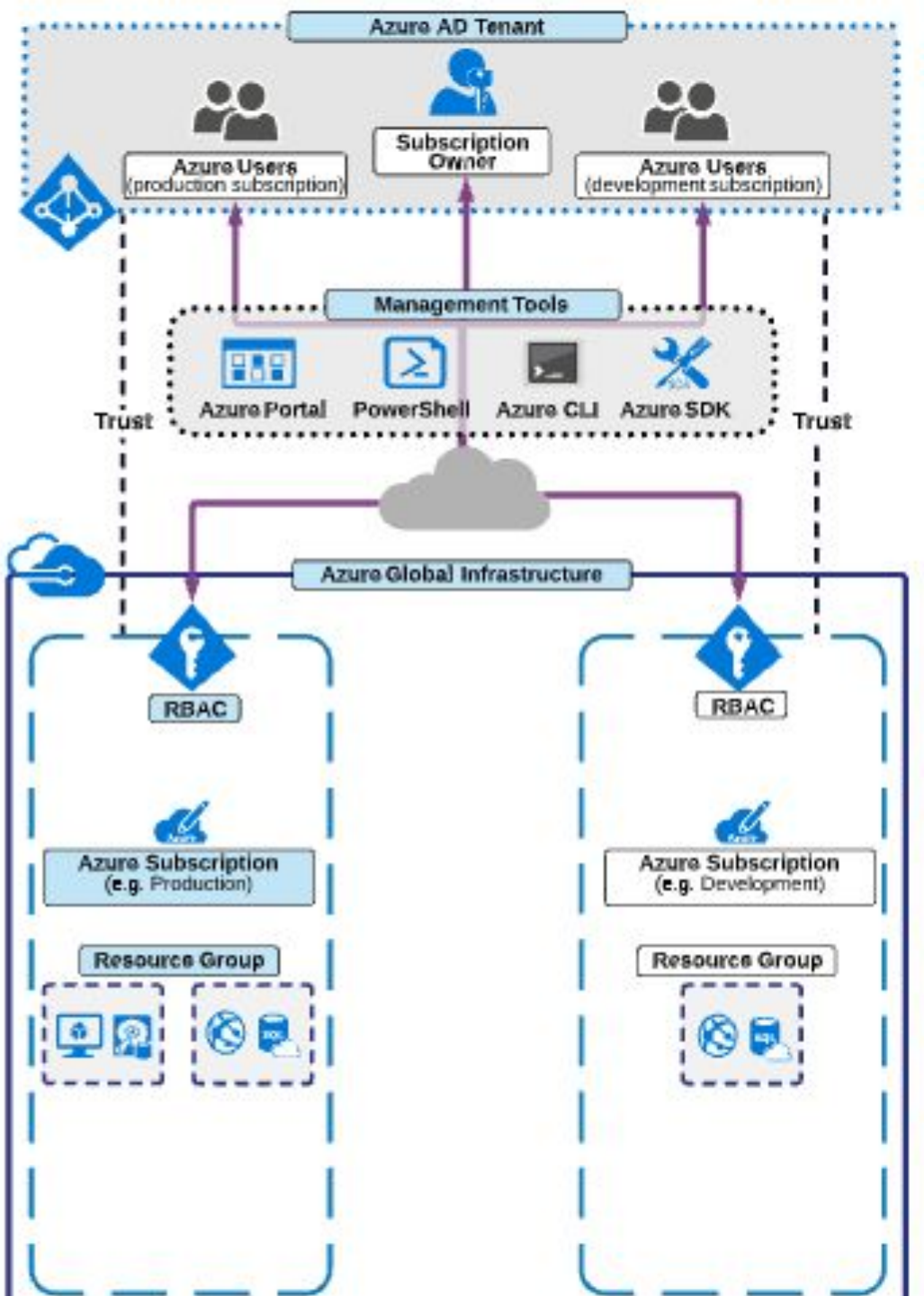
Physical and Networking Layer

Subscription and Services

Within Azure, the subscription is where all resources are ultimately contained. This is the uppermost billing and management layer.

This page provides a high-level view and recap of key components and tools that will be used throughout this course.

Appendix



Subscription

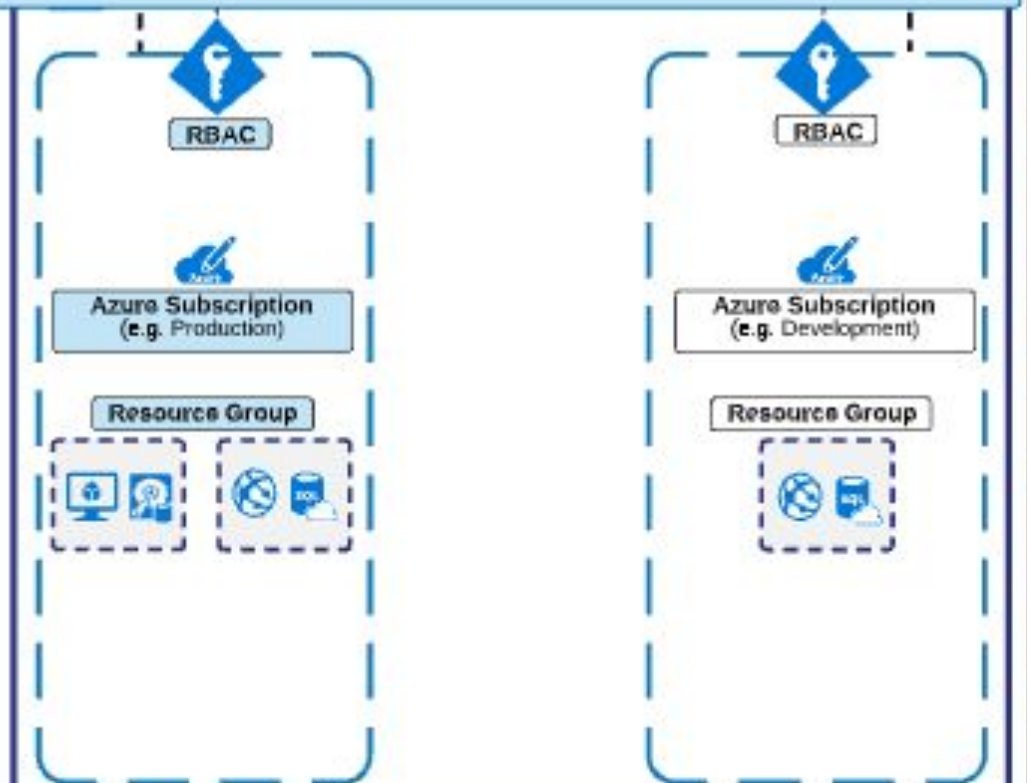
Azure Global Infrastructure

Close

Azure global infrastructure is the underlying physical infrastructure that powers Azure. This is managed by Microsoft and includes data centers and network connectivity across the world.

More information can be found on the key components that make up the Azure global infrastructure within the Physical page found here.

Appendix



Subscription

Management Tools

Close

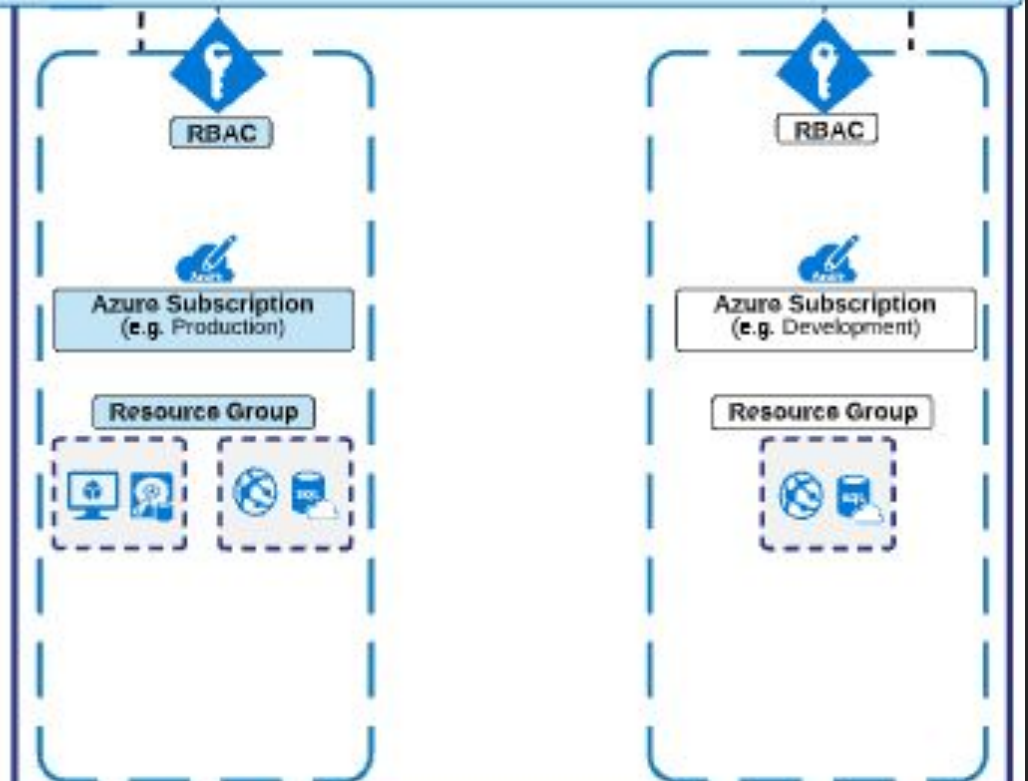
Microsoft provides a range of tools for creating, accessing, managing, configuring, and monitoring resources within Azure. These are available across multiple platforms, including Windows, Linux, and macOS.

Below is a summary of some of the important tools:

- Azure Portal: Comprehensive management of Azure services through a web browser (portal.azure.com)
- PowerShell: Various cmdlets available to create, test, deploy, and manage Azure solutions
- Azure CLI: Lightweight cross-platform command line tool with functionality similar to PowerShell
- Azure SDK: Tools and templates to help develop Azure solutions across various programming languages

All tools ultimately interact with Azure through the Azure Resource Manager API.

Appendix



Subscription

Azure AD Tenant

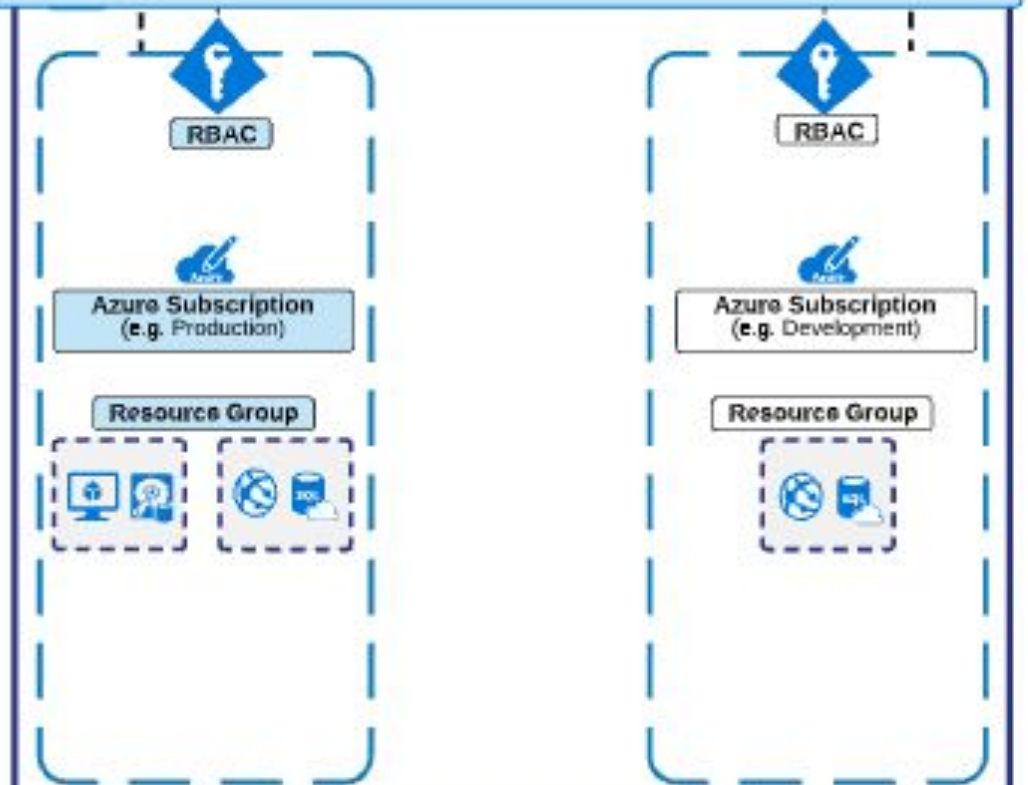
Close

The Azure AD tenant, based on Azure Active Directory, is where identity and access is managed for Azure as well as a range of other Microsoft Cloud services. An Azure AD tenant refers to a single directory (for example, linuxacademy.onmicrosoft.com).

Here are some important notes about the Azure AD tenant:

- An Azure AD tenant can be associated with multiple subscriptions — this is useful when your company wishes to use separate subscriptions for production and development, while maintaining a single identity.
- Azure subscriptions can only be associated with one AD tenant.

Appendix



Subscription

Azure Subscription

Close

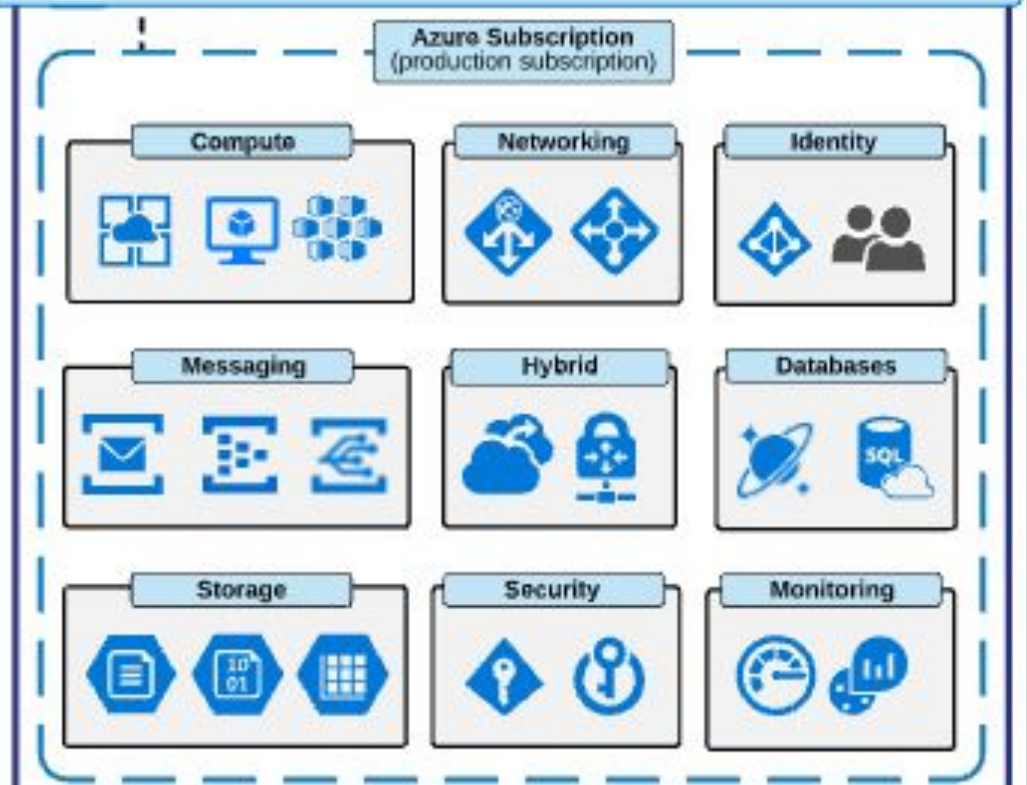
Azure subscriptions are the absolute root-level container of any resource within Azure. You must first have a subscription before any Azure services can be created.

Here are some important reasons to use an Azure subscription:

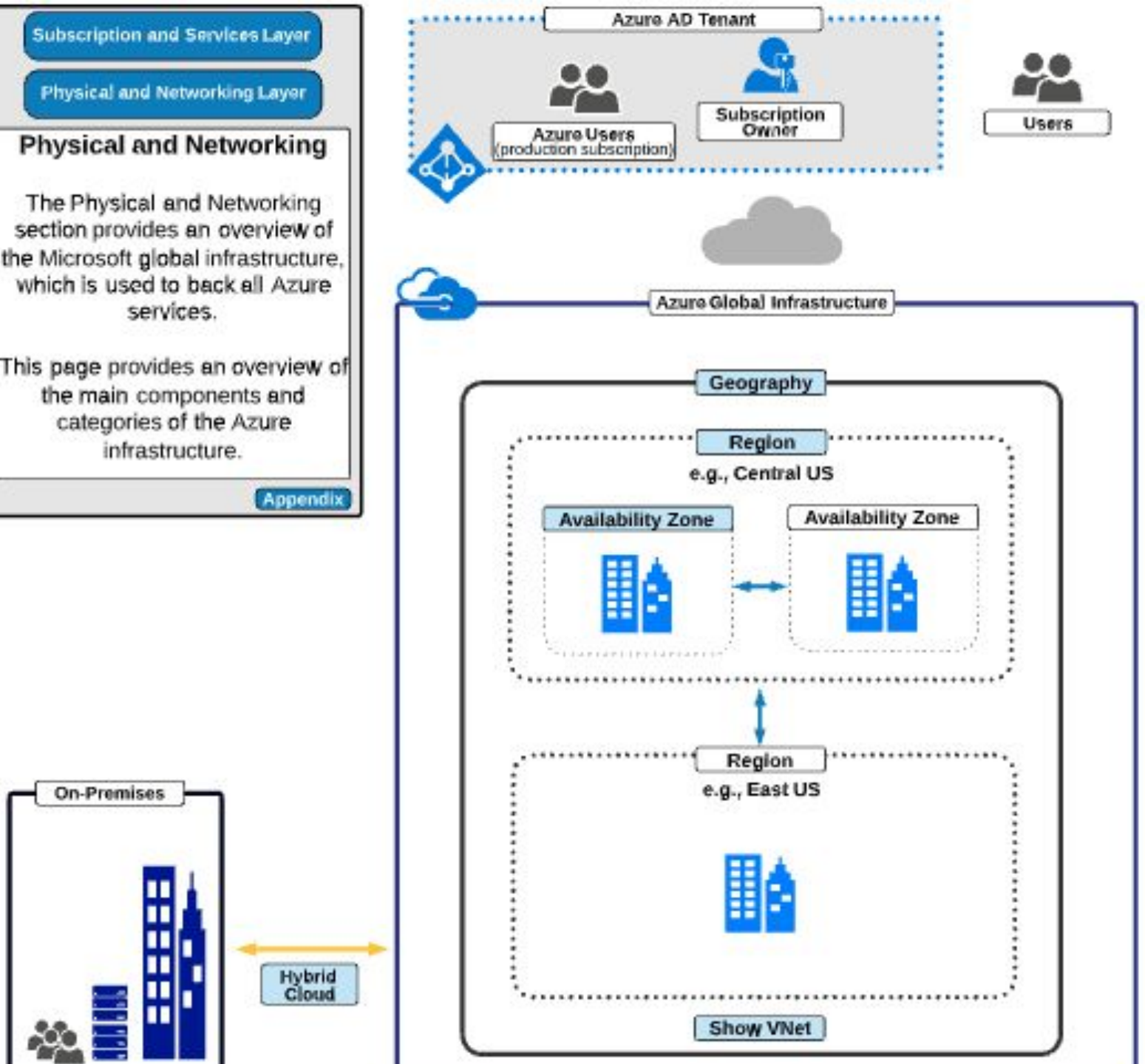
- As a way to organize your resources at the billing and management layer
  - For example, one subscription for production purposes and a separate one for development
  - Another example may be to have one subscription for each internationally separate business unit

Resources, costs, and metrics can be viewed for all resources across your subscription.

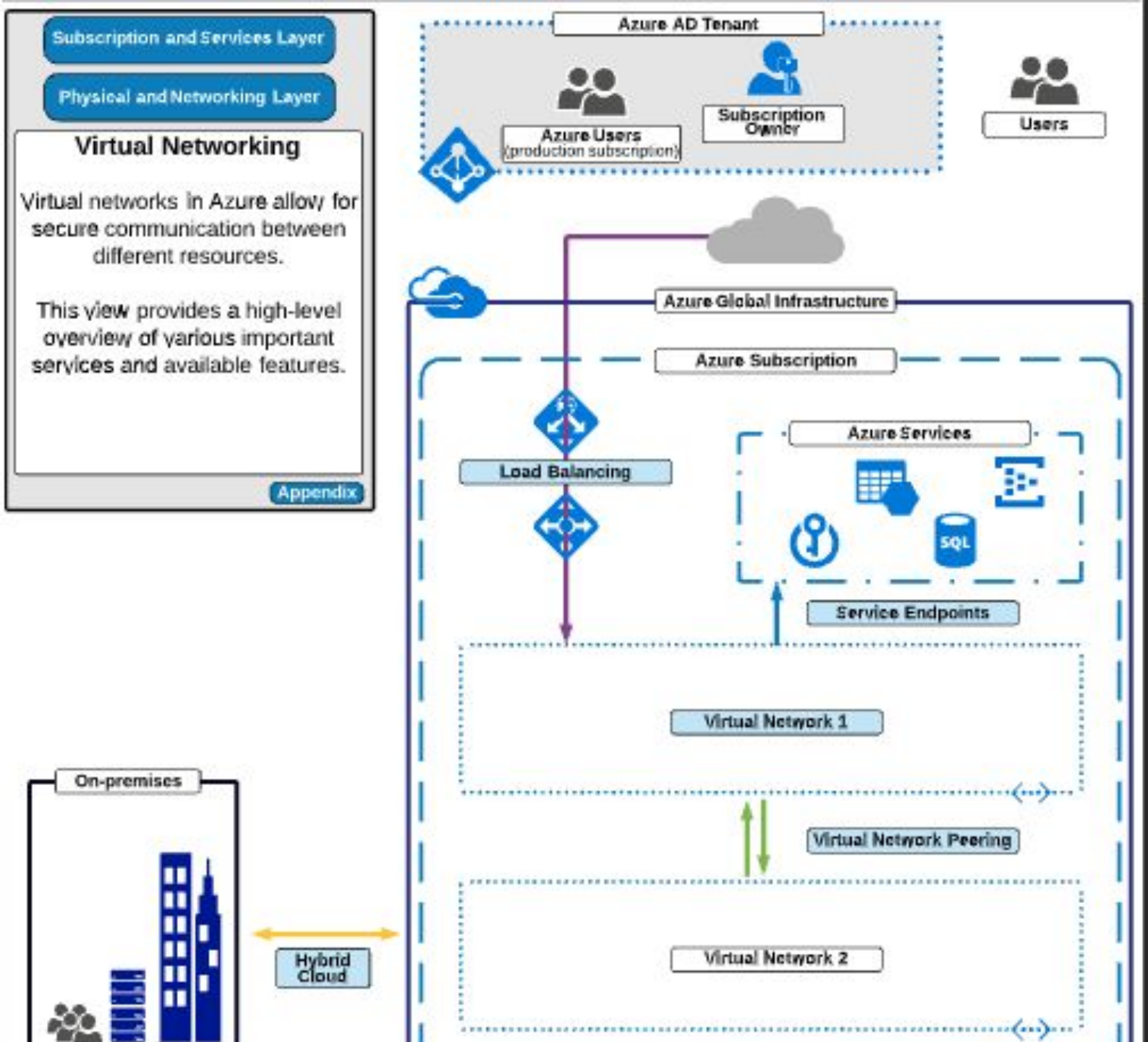
Go Back | Appendix



Subscription


Subscription and Services Layer
Physical and Networking Layer
Physical and Networking
The Physical and Networking section provides an overview of the Microsoft global infrastructure, which is used to back all Azure services.
This page provides an overview of the main components and categories of the Azure infrastructure.
Appendix
Azure AD Tenant
Azure Users (production subscription)
Subscription Owner
Users
Azure Global Infrastructure
Geography
Region
e.g., Central US
Availability Zone
Availability Zone
Region
e.g., East US
Show VNet
On-Premises
Hybrid Cloud

Physical


Subscription and Services Layer
Physical and Networking Layer
Virtual Networking
Virtual networks in Azure allow for secure communication between different resources.
This view provides a high-level overview of various important services and available features.
Appendix
Azure AD Tenant
Azure Users
(production subscription)
Subscription Owner
Users
Azure Global Infrastructure
Azure Subscription
Load Balancing
Azure Services
SQL
Service Endpoints
Virtual Network 1
Virtual Network Peering
Virtual Network 2
On-premises
Hybrid Cloud

Networking

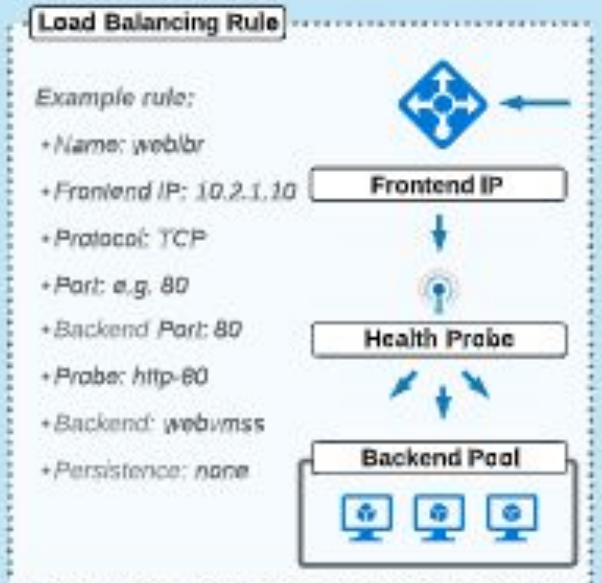
Load Balancer

Close

Load Balancers provide the ability to distribute traffic amongst multiple resources, helping to support high availability and elasticity. An Internal Load Balancer is only accessible to resources on or connected to an Azure Virtual Network (VNet). Public Load Balancers, on the other hand, are accessible over the Internet.

Key information about the Load Balancer:

- Azure Load Balancer supports the distribution of traffic at layer 4 (TCP, UDP).
- Session affinity / persistence can be configured to ensure a client requesting traffic through the Load Balancer gets a response from the same backend VM, based on Client IP, or Client IP + Protocol.
- Load Balancers can be either Standard or Baskic SKU, some of the key differences are as follows:
  - Backend pool size: Standard SKU supports up to 1000 instances, basic SKU supports up to 100
  - Backend pool endpoints: Standard SKU adds the ability to blend types (VMs, Availability Sets, VMSS)
  - Health probes: Standard SKU adds HTTPS
  - Availability Zones: Standard SKU supports zone-redundant and zonal frontends for inbound/outbound
  - Outbound connections: Standard SKU adds greater control over outbound connectivity



Configuration Items:

Frontend IP: The IP address of the Load Balancer itself

Health Probe: Used for determining whether the final destination instance is available to receive traffic

Backend Pool: Defines the ultimate destination for traffic

Load Balancing Rule: Ties all configuration items together, allowing the definition of an inbound port mapping

Inbound NAT Rules: Provides the ability to map inbound traffic directly to instances behind the Load Balancer.

Networking

Application Gateway - Overview

Close

Overview | Configuration

Azure Application Gateway is a load balancer which for web traffic. This allows distribution of traffic, in a similar fashion to the Azure Load Balancer, but with additional Layer 7 (http/https) functionality.

Load balancing at layer 7 means that the Application Gateway can provide additional functionality (compared to the Load Balancer) such as, URL-based routing, SSL termination, web application firewall functionality, session affinity, and more.

Key information about the Application Gateway:

- Provides layer 7 load balancing functionality (URL-based routing, SSL termination, etc.)
- A number of features are currently "In Preview" and do not currently appear on the exam
- The Application Gateway can either be Standard or V/AF tier:
  - V/AF (web application firewall) tier includes protection against common web application exploits.
- Using the current generation of Application Gateways requires manual sizing and high availability:
  - Increasing instance count to two or more helps achieve high availability of your Application Gateway,
  - Use a SKU size of medium or large, for production environments.
- Using version 2 of the Application Gateway provides a somewhat more automated experience:
  - High availability can be automated,
  - Availability Zones can be used,
  - SKU size is no longer required to be defined during provisioning.

Important limitations to be aware of:

- Only one public IP address is supported on an Application Gateway
- Public IP addresses cannot be static (unless using V2 SKU); for this reason a CNAME DNS entry is recommended to be pointed to the DNS address of the Application Gateway
- The same port (e.g. port 80) cannot be used for both public and private listeners
- Rules are processed in the order they are listed (ensure path based rules appear before basic rules)

Networking

Application Gateway - Configuration

Close

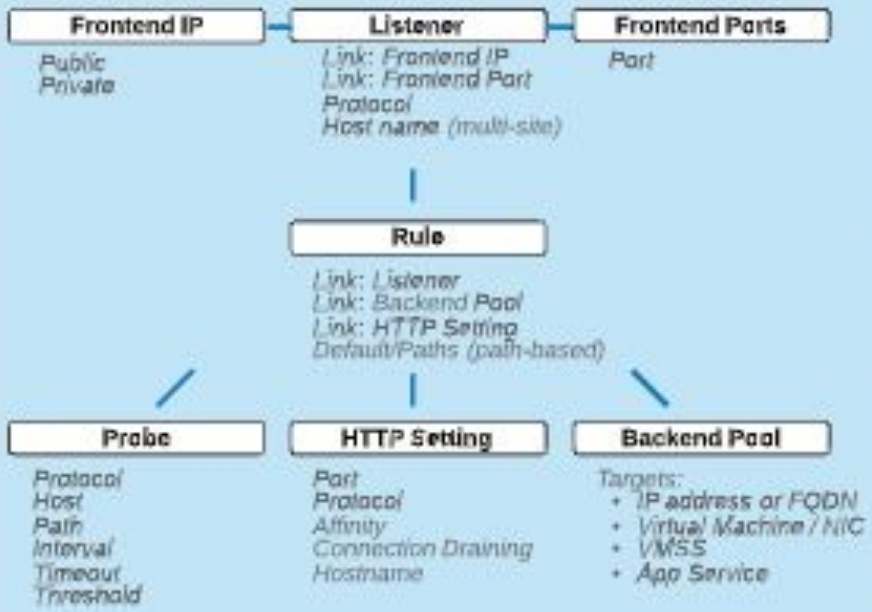
Overview | Configuration

Since the Application Gateway provides more functionality than the Load Balancer, the configuration is more detailed and versatile.

Below is an overview of the various resource properties, and how they tie together to create the overall functioning configuration of the Application Gateway.

Important configuration notes:

- The Application Gateway must exist in a dedicated subnet which cannot be shared by any other resources, other than additional Application Gateways.
- Rules are processed in the order they are configured; multi-site rules should appear first.



Configuration Items:

Frontend IP: The IP address of the Application Gateway: supports both public and private IP addressing

Frontend Port: The port that the Application Gateway listens on

Listener: Combination of a protocol, frontend IP, and frontend port.

- Multi-site listeners allow the specification of a host name.

Backend Pool: IP addresses/FQDN's (any that are reachable), or NICs belonging to resources that are hosting the web application

Custom Health Probe: Used to determine whether the backend member is healthy

HTTP setting: Configures how traffic is routed to backend members (including port, protocol, cookie-based-affinity, probe, timeout)

Rule: Ties all configuration items together, allowing

Networking

Subscription and Services Layer

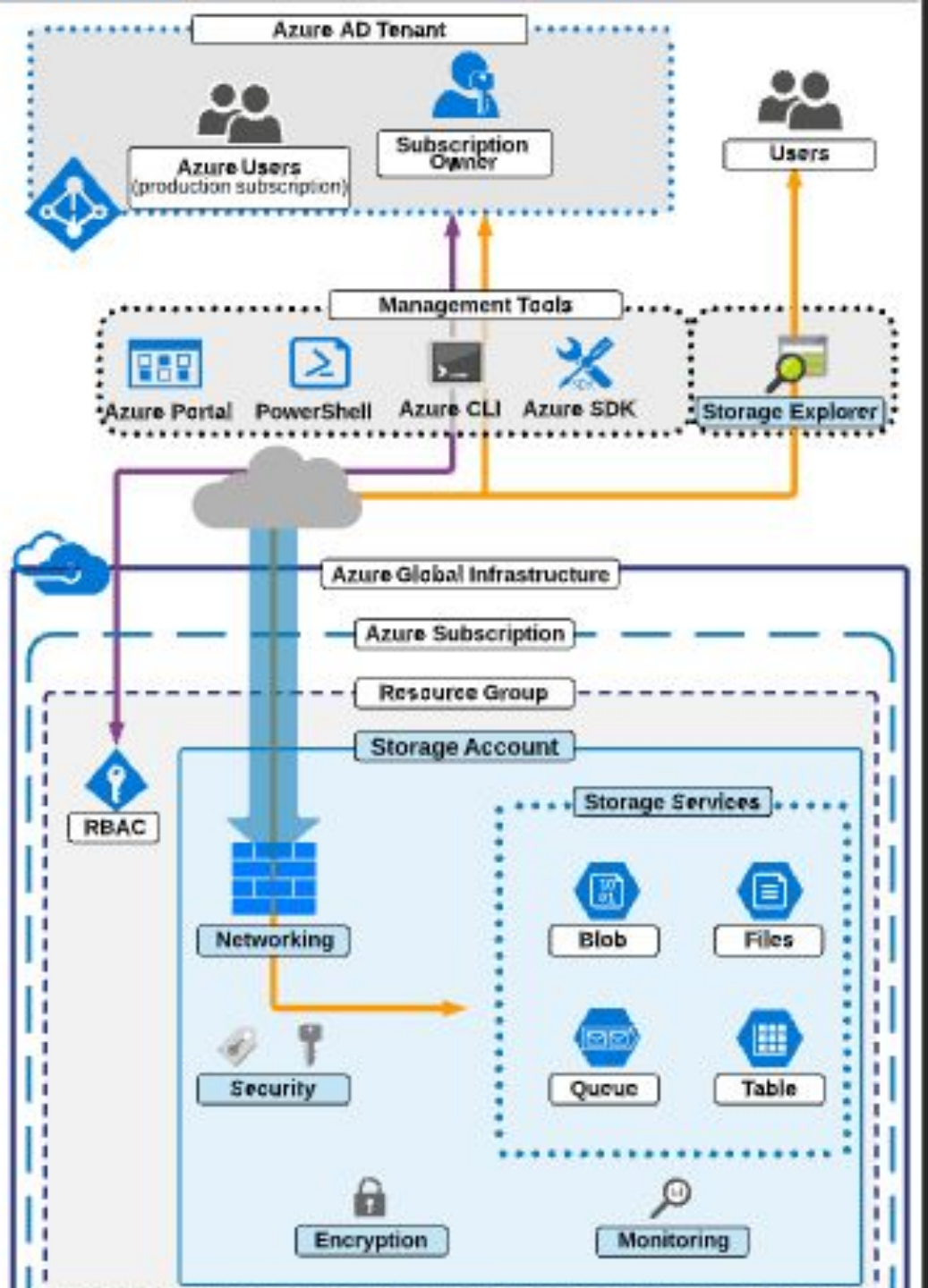
Physical and Networking Layer

Azure Storage

Azure Storage represents a range of services used for housing data. At the root of these services is the Azure Storage Account.

Storage Accounts can contain:
Blob: Object based storage
Files: Managed file sharing
Queues: Message queues
Table: NoSQL key-value data

Appendix



Storage

Subscription and Services Layer

Physical and Networking Layer

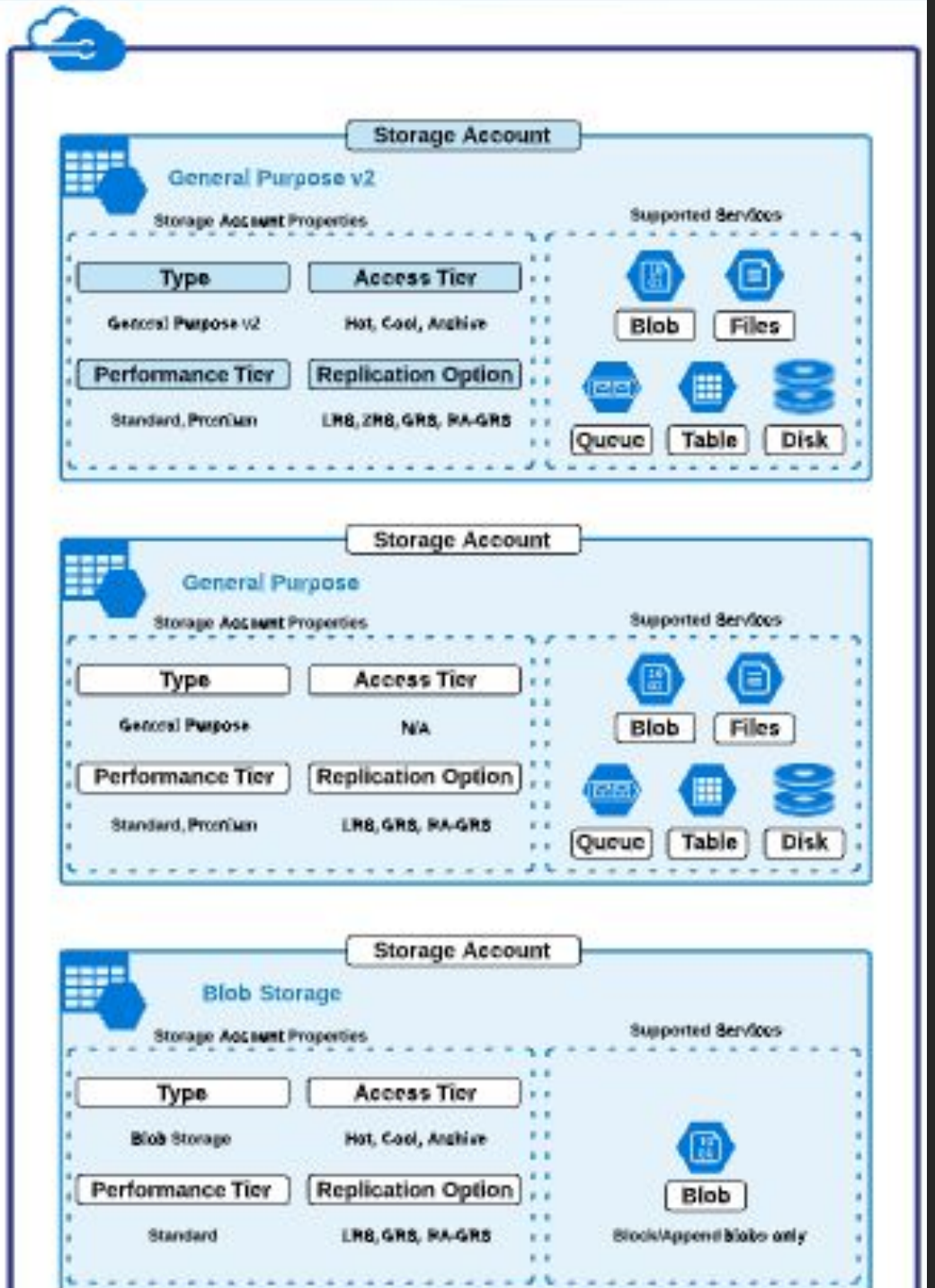
Storage Accounts
(Detailed View)

All Blobs, Files, Queues, Tables, and Disks are created within Storage Accounts.

This page helps to illustrate key configuration properties, and important differences with the three types of Storage Accounts.

Go Back

Appendix



Storage

Storage Accounts (SA)

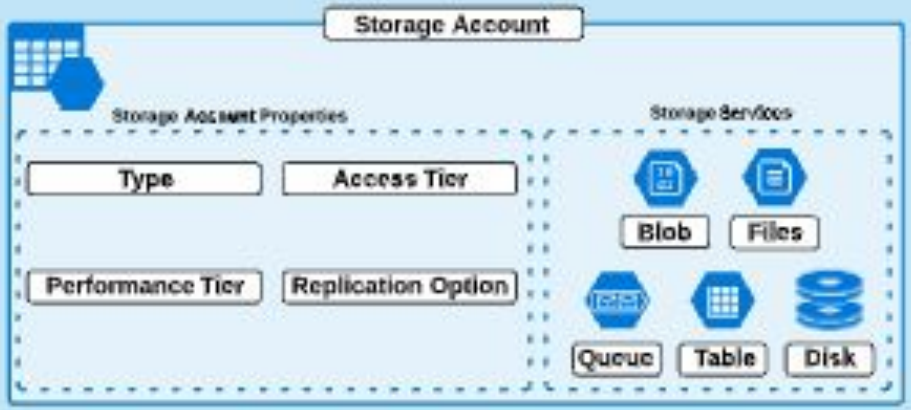
Close

Storage Accounts (SA) are the parent container of the primary Azure Storage objects: Blobs, Files, Queues, and Tables. Before you create any of these services, you must create an SA.

A number of important configuration properties are defined at the SA level, which directly influences the configuration of the contained storage services. Examples are replication, performance, and access tiers.

Key information:

- Storage Accounts must be uniquely named across Azure (for ALL of Azure, not just your subscription).
- Three types are available: general-purpose v2 (GPv2), general-purpose (GP), and blob storage (Blob).
- There are limitations and variations between all of the three SA types. Refer to the previous SA diagram
  - For example, Cool and Archive access tiers are only available with GPv2 and Blob SAs
- All SAs are encrypted at rest using Storage Service Encryption. This cannot be disabled.



Storage

Storage Account - Type

Close

There are three types of Storage Accounts (SA) you can create, and these directly influence the features, pricing, and properties available to you. The three SA types are:

1. Blob storage
2. General-purpose (GPv1)
3. General-purpose v2 (GPv2)

Blob storage

- Special type of SA which only supports the creation of Blob storage (and only block and append blobs)
- Originally (prior to GPv2) required in order to leverage the additional access tiers

General-purpose v1

- Standard type of Storage Account which supports all storage services, now superseded by GPv2
- Provides lowest transaction prices, but higher storage prices
- Supports Azure Service Manager (Classic)

General-purpose v2

- Supports all storage services, and provides access to the latest features
- Supports Access Tiers (Cool and Archive storage)
- Supports zone-redundant storage
- Recommended for most scenarios

Storage Account Type	Supported Services	Supported Performance Tiers	Supported Access Tiers	Replication Options
General-purpose v2	Blob, File, Queue, Table, and Disk	Standard, Premium	Hot, Cool, Archive	LRS, ZRS, GRS, RA-GRS
General-purpose v1	Blob, File, Queue, Table, and Disk	Standard, Premium	N/A	LRS, GRS, RA-GRS
Blob storage	Blob (block and append blobs only)	Standard	Hot, Cool, Archive	LRS, GRS, RA-GRS

Storage

Storage Account - Access Tier

Close

Microsoft provides three Access Tiers for Storage Accounts. This ultimately helps optimize the pricing model of your ***blob storage***, based on the frequency of reads and writes. The three options are:

- Hot storage: for storing *frequently accessed data*
- Cool storage: for storing *infrequently accessed data* which is *stored for at least 30 days*
- Archive storage: for storing *rarely accessed data* which is *stored for at least 180 days*

Use cases:

- Hot storage: the typical use case of data which is actively read from and written to frequently
- Cool storage: often used for short-term backup, or large volumes of infrequently used (e.g. media) content
- Archive storage: commonly used for long-term backups, or for data which must be retained for long-term in order to comply with law/regulation (e.g. financial records going back 7 years)

Key information:

- Configuring the Access Tier at the SA level will set the default; this can still be set at the object level.
- Archive tier can only be set at the object level, not as a default at the SA level.
- General Purpose v1 (GPv1) type Storage Accounts do not support Access Tiers.

Storage

Storage Account - Replication Options

Close

To ensure that data within Storage Accounts is always available, it is possible to configure Replication Options. This controls how and where your data is replicated. There are four available options:

1. Locally-redundant storage (LRS): Replicates data to another storage scale unit
2. Zone-redundant storage (ZRS): Replicates data across three storage clusters within a single region
3. Geo-redundant storage (GRS): Replicates data to a separate region
4. Read-access GRS (RA-GRS): GRS with the added ability of being able to read from the secondary copy

Important information about how data is synchronized:

- LRS and ZRS replication occurs synchronously
- GRS and RA-GRS replication occurs asynchronously

Scenario	LRS	ZRS	GRS	RA-GRS
Node failure within a datacentrer	Yes	Yes	Yes	Yes
Entire datacenter failure	No	Yes	Yes	Yes
Region-wide outage	No	No	Yes	Yes
Read access to your data in the event of region-wide outage	No	No	No	Yes

More information can be found in the following Microsoft article:
https://azure.microsoft.com/en-us/support/legal/sla/storage/v1_3/

Storage

Storage Account - Performance Tiers

Close

There are two Performance Tiers available to Storage Accounts (SA):

1. Standard: Storage Accounts which are backed by magnetic drives and provide low cost storage
2. Premium: Storage Accounts backed by solid state drives with high performance, low-latency access

Use cases:

- Premium Performance Storage Accounts can currently only be used with Azure Virtual Machine (VM) disks
  - They are suitable for database servers and other high I/O applications
- VMs which use Premium storage also qualify for a 99.9% uptime/connectivity SLA, even when running as a single instance. This is compared to 99.99% for two VMs running in an Availability Set.
- Standard Performance Storage Accounts are currently used for the majority of use cases, as they are supported by all Storage Account storage services.

Key information:

- Premium Performance Access Tier is only supported by General Purpose type SAs.
- Currently, creating an SA with the Premium Performance tier SA will result in the following limitations:
  - Only Virtual Machine disks can be stored in the SA
  - Only Page Blobs will be able to be created in the SA
  - Only locally redundant storage (LRS) will be able to configured for the SA

Storage

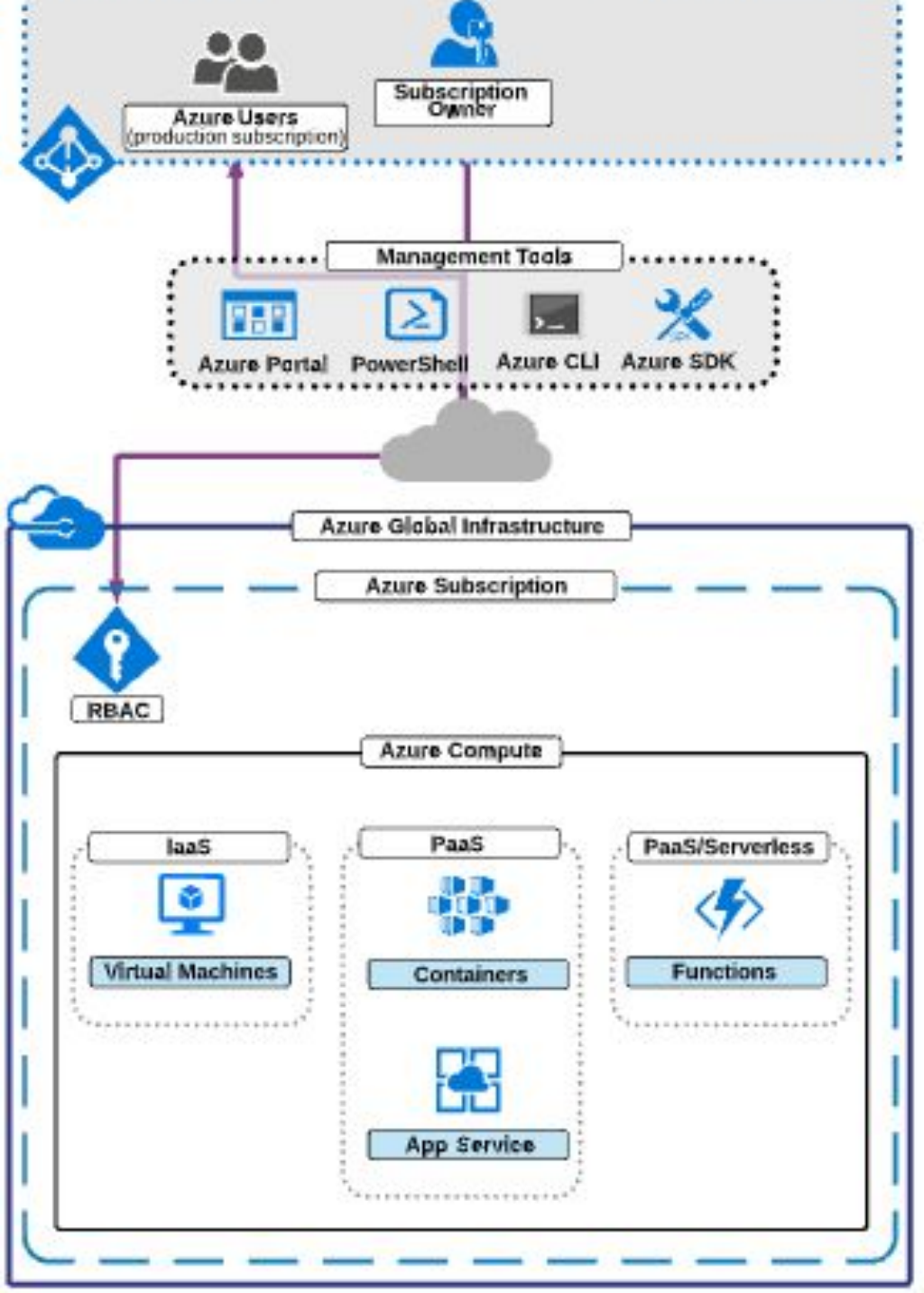
Physical and Networking Layer

Azure Compute Services

Microsoft provides a range of hosting models to represent the various services for hosting applications in Azure.

Of these services, Virtual Machines are the most common. However as more cloud-native development occurs, this is changing.

Appendix



Compute

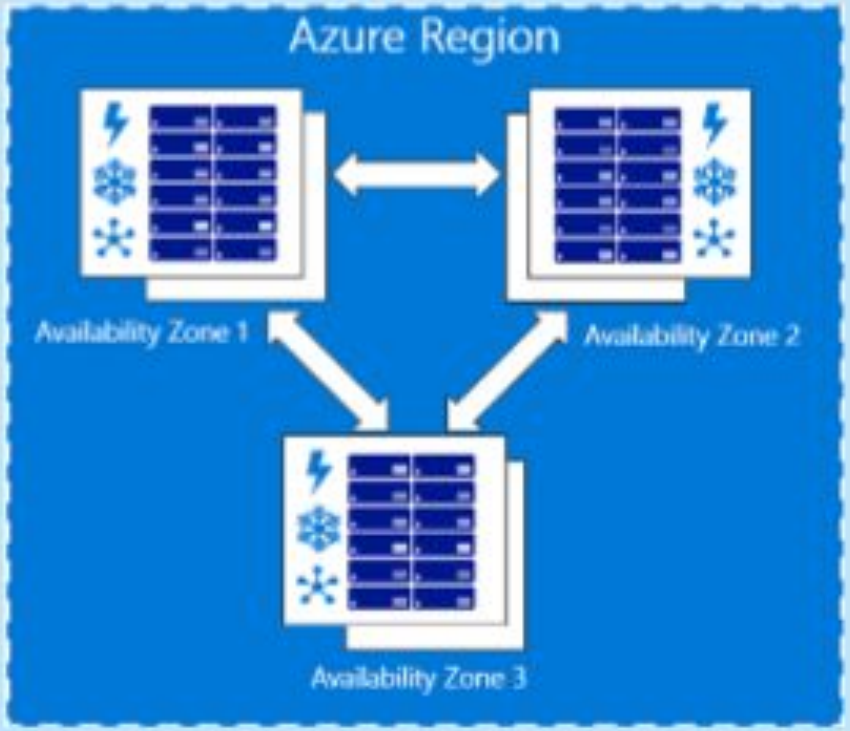
Availability Zones (AZ)

Close

Availability Zones (AZ) are a relatively new service from Microsoft, and are focused on improving high availability. AZs are not yet available for all regions or services.

An AZ works at the Microsoft datacenter layer by providing isolated locations within an Azure region that have redundant power, cooling, and networking. There are two ways to use an AZ:

1. **Zonal services** - where you pin a resource to a specific zone (e.g. VMs, managed disks, load balancers)
2. **Zone-redundant services** - where the platform replicates across zones (e.g. ZRS storage, SQL)



https://docs.microsoft.com/en-us/azure/availability-zones/az-overview

Compute

Virtual Machines (VM)

Close

Virtual Machines (VMs) can be thought of as virtual computers which operate on the Azure platform. Using an image, the VM can run an Operating System as if it were a real physical server.

VMs are considered Infrastructure as a Service (IaaS), which at a very high level means that you manage most of the environment, except for the underlying physical infrastructure.

Use cases:

- VMs can be used for operating servers in the cloud, similar to how you would currently do something, like running an on-premises file or a web server.

Key information about Virtual Machines:

- VMs have important characteristics that define how they behave, such as size, image, network, etc.
- Both Windows and Linux VMs are supported in Azure, but only the 64-bit versions.

Key information about Virtual Machine SLA:

- Microsoft will only provide an SLA for a Virtual Machine under the following conditions:
  - When running a single VM only, with premium storage being required for all OS disks and data disks
  - When running two or more instances of a VM within the same Availability Set
  - When running two or more instances of a VM across two or more Availability Zones in the same region

More information can be found at the Microsoft article, here:
https://azure.microsoft.com/en-us/support/legal/sla/virtual-machines/v1_8/

Compute

Virtual Machine (VM) Storage

Close

Overview | Encryption

Virtual Machines disks are housed in Azure storage. A VM disk is where the operating system (OS) is installed, where applications are installed, and where data resides.

There are three main disks used by VMs:
1. **Operating system disk**: based upon a VM image which includes the necessary operating system files
2. **Data disks**: can be used as desired and are typically created as a blank disk
3. **Temporary disk**: included with all VMs by default for temporary purposes only - data can be lost

Important information about VM disks:
- All VM disks are essentially virtual hard disks (VHDs) stored in an Azure storage account as page blobs
- VHD files can be used as a source to create disks or images (e.g. if you upload a VHD)
- VM disks can either be managed or unmanaged:
  - Unmanaged disks require manual configuration of the storage account (SA) the disk resides in.
  - Managed disks are still housed within an SA, however this is managed by Microsoft to provide greater simplicity, reliability, and control over performance.
  - Managed disks are independent resources, whereas unmanaged disks belong to the VM resource.
- A VM will l always include an operating system disk, and a temporary disk, at a minimum
- VM disks can be one of three performance tiers:
  - Standard HDD disks - cost effective, low IOPS, suitable for bulk storage, backups, etc.
  - Standard SSD disks - similar to Standard HDD disks, but greater consistency and reliability.
  - Premium SSD disks - high performance, low latency disks, accessible to VMs with an 's' in the name.
- For high-performance applications that require high IOPS, Premium SSD Disks should be used
- See more information here:
https://docs.microsoft.com/en-au/azure/virtual-machines/linux/about-disks-and-vhds

Important information about securing data:
- Azure Storage Service Encryption safeguards VM disks at the Azure infrastructure/data level.
- Azure Disk Encryption also protects at the VM disk resource level, protecting OS/boot and data disks.
- Refer to the Security page for more information.

Compute

Azure Disk Encryption

Close

Overview | Encryption

Azure Disk Encryption (ADE) encrypts data for either Windows or Linux VM disks. It helps protect against the risk of data theft from a disk that an unauthorized individual may gain access to.

Important information about Azure Disk Encryption:

- Windows VM disks are protected with BitLocker.
- Linux VM disks are protected with DM-Crypt.
- Azure Disk Encryption provides volume encryption for the OS and data disks.

Enabling Azure Disk Encryption:

- ADE can be enabled in a number of ways, including PowerShell, CLI, and templates.
- The following commands can be used:
  - PowerShell: `Set-AzureRmVMDiskEncryptionExtension`
  - CLI: `az vm encryption enable`
- Be aware that the latest release of Azure Disk Encryption relies on a Key Vault.
- The Key Vault used for ADE must have 'enable access to ADE for volume encryption' selected.

Compute

Virtual Machine (VM) Images

Close

VM Images help solve two main problems:
1. Performing an operating system (OS) installation within Azure is not possible.
2. Within the cloud, it's very beneficial to be able to pre-configure a VM and re-deploy it repeatedly.

For these reasons, we can either use readily available VM Images from the Azure Marketplace, create our own custom images from existing VMs, or upload a generalized VHD to blob storage.

Information about Marketplace Images:

- Can be deployed with ARM templates, PowerShell, or CLI, by specifying these image parameters:

Parameter	Description	Example
Publisher	The organization which built and supplied the image	Canonical, MicrosoftWindowsServer
Offer	Typically represents a category/grouping of related images from a Publisher	UbuntuServer, WindowsServer
SKU	An instance of an offer	16.04-LTS, 2016-Datacenter
Version	The version number of an image SKU	Determined by the Publisher. The term latest can be specified to use the latest version.

Information about Custom Images:

- To create an image that can be used for deploying VMs, the OS of the VM must first be generalized.
- Generalizing removes unnecessary account and application specific information from an OS image.
- When you create an image from a VM, it can no longer be powered on or used as a VM normally.

Operating System	How to Generalize
Windows	Generalize the VM (`%windir%\system32\sysprep\sysprep.exe`, OOBE, generalize); Deallocate the VM; Mark the VM as generalized; Create an image from the VM.
Linux	Generalize/Deprovision the VM (`sudo waagent -deprovision+user`); Deallocate the VM; Mark the VM as generalized; Create an image from the VM.

Compute

Virtual Machine (VM) Extensions

Close

Virtual Machine (VM) extensions are generally lightweight applications or services which assist with post-deployment configuration, or various automation tasks on VMs.

Extensions can be enabled through the Portal, CLI, PowerShell, and with ARM Templates.

Use cases:

- Using the Custom Script Extension to change the timezone and language settings for a newly deployed VM
- Using the Azure Diagnostics Extension to support additional monitoring of VMs

NAME	TYPE	VERSION	STATUS
MicrosoftInsights.VMDiagnosticsSettings	Microsoft.Azure.Diagnostics.IaaSDiagnostics	1.*	Provisioning succeeded

NAME	TYPE	VERSION	STATUS
LinuxDiagnostic	Microsoft.Azure.Diagnostics.LinuxDiagnostic	3.*	Provisioning succeeded

Compute

Network Interfaces

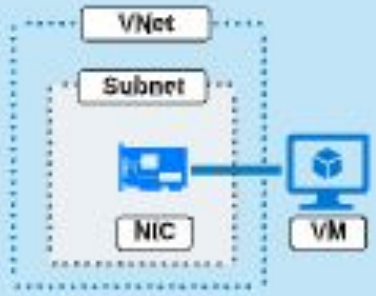
Close

Network Interfaces (also known as Network Interface Cards, or NICs) are independent resources which, when connected to a Virtual Machine (VM), enable network connectivity with other networked resources.

Use cases:
- VM network communication with the Internet, Azure, and even on-premises (via VPN or ExpressRoute)

Important information about VM NICs:
- NICs are independent resources which are created separately from a Virtual Machine
- When a VM is configured to forward packets (e.g. as a firewall), *IP Forwarding* must be enabled for the NIC
- It is recommended to not statically assign private IP addressing within the operating system (OS) of a VM
- If the VM is using a NIC with multiple IP addresses, then static configuration *will* be required within the OS
- DNS settings can be specified manually for an individual NIC, or be inherited from a VNet

As the NIC is ultimately a network resource, greater detail can be found on the Networking page here.



Compute

Virtual Machine (VM) Sizes

Close

The size of a virtual machine (VM) specifies important characteristics of the virtual hardware available to a VM. This is not just CPU and memory, but also network, storage, and other key properties.

It is important to be familiar with these characteristics and the various VM families, but the AZ-300 does not require you to memorize specific resource amounts or figures.

Important information about VM sizes:

- VM size defines the vCPU and memory for a VM.
- The maximum number of network interfaces, or disks, that can be attached is limited by the VM size.
- Maximum disk IOPS and network bandwidth is also limited by the VM size.

Below is a summary of current VM families:

See https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sizes for updated details.

Type	Sizes	Description
General purpose	B, Dsv3, Dv3, DSv2, Dv2, Av2, DC	Balanced CPU-to-memory ratio. Ideal for testing and development, small to medium databases, and low to medium traffic web servers.
Compute optimized	Fsv2, Fs, F	High CPU-to-memory ratio. Good for medium traffic web servers, network appliances, batch processes, and application servers.
Memory optimized	Esv3, Ev3, M, GS, G, DSv2, Dv2	High memory-to-CPU ratio. Great for relational database servers, medium to large caches, and in-memory analytics.
Storage optimized	Lsv2, Ls	High disk throughput and IO ideal for Big Data, SQL, NoSQL databases, data warehousing and large transactional databases.
GPU	NV, NVv2, NC, NCv2, NCv3, ND, Ndv2 (Preview)	Specialized virtual machines targeted for heavy graphic rendering and video editing, as well as model training and inferencing (ND) with deep learning. Available with single or multiple GPUs.
High performance	H	Our fastest and most powerful CPU virtual machines with optional high-throughput network interfaces (RDMA).

Compute

ARM Templates for VMs

Close

Azure Resource Manager (ARM) templates can be used to deploy *infrastructure as code*. This means using text files to represent infrastructure. For more information, refer to the Deployments page.

Components of the VM Resource

It is important that you are familiar with the ARM Template, specifically for the VM resource. Below is a summary:

- **hardwareProfile**: VM size
- **osProfile**: Values required by the OS (hostname, credentials)
- **storageProfile**: The OS image to be used
- **osDisk**: Properties of the OS disk (caching, creation method)
- **dataDisks**: Array of properties for 1 or more data disks
- **networkProfile**: Network interface
- **diagnosticsProfile**: Monitoring and diagnostics

Compute

Virtual Machine Scale Sets (VMSS)

Close

A Virtual Machine Scale Set (VMSS) is essentially a service which provides the ability to dynamically scale Virtual Machines (VM) in and out based on demand.

There are two main components to a VMSS:
- The definition of the VM; image, size, connectivity, etc
- The autoscaling configuration; defining when, why, and how a VMSS will scale

What is an Autoscale setting:
Enabling autoscaling for VMSS requires an Azure Monitor Autoscale setting. An Autoscale setting is comprised of the following:
- One or more profiles - profiles define default, maximum, and minimum capacity, as well as any rules
  - Regular/default profile - default profile which includes rules to scale based on metrics
  - Fixed date profile - profile which applies only on a certain single date/time (e.g. January 1, 2019)
  - Recurrence profile - profile which applies on recurring date/time ranges (e.g. Monday - Friday)
- One or more metric rules - used to define what a criteria and action
  - Condition - specific metric criteria to occur, e.g. average CPU reaches 70%
  - Response - specific action to apply if the criteria occurs, e.g. increase instances by 1
- Note that metric rules are optional for fixed/recurring profiles

Which profile will Autoscale run?
- Fixed date profiles will run first; if there are multiple fixed date profiles, Autoscale will select the first one
- If no fixed date profiles exist, Autoscale will run any recurrence profiles that exist
- If there are no fixed date or recurrence profiles, Autoscale will run the regular/default profile

Which rules will Autoscale run?
- Autoscale will select which rules to run, after it determines which profile should run
- Scale-out rules (direction = increase) will run first
- If there are multiple scale-out rules, Autoscale will use the largest capacity
- If no scale-out rules should trigger, Autoscale will evaluate scale-in rules (direction = decrease)
- If there are multiple scale-in rules, Autoscale will use the largest capacity

Compute

Availability Sets (AS)

Close

Availability Sets (AS) are a tool for managing the high availability of Virtual Machines (VMs). In order to use an AS, you create the AS and then place VMs with a duplicate purpose in the AS.

The AS helps you to achieve the 99.95% Azure SLA by managing how VMs within the AS respond to planned (i.e. updates) or unplanned (i.e. faults) outages. This is managed through **update domains (UD)** and **fault domains (FD)** which are explained further below.

Use cases:
- Two more more duplicate file servers within an AS, ensuring one file server is always available to users
- Two or more duplicate web servers within an AS, ensuring one web server is always available to users

Important information:
- A single VM within an AS will not achieve the 99.95% Azure SLA; there must be at least two VMs present.
- Five (5) to twenty (20) update domains can be configured; the default value is five.
- Two (2) to three (3) fault domains can be configured, and this varies by region.

Fault Domains (FD):
Fault domains represent a common source of failure. For example, when the underlying Microsoft Azure infrastructure which the VMs operate on shares the same network switch, or power source.

Update Domains (UD):
Update domains are used to logically group VMs (and underlying hardware) which can be rebooted by Microsoft for the purpose of platform updates. *Only one UD will be rebooted at a time.*

NAME	STATUS	FAULT DOMAIN	UPDATE DOMAIN
vm1	Running	0	0
vm2	Running	1	1
vm3	Running	0	2
vm4	Running	1	3
vm5	Running	0	4

Compute

Subscription and Services Layer

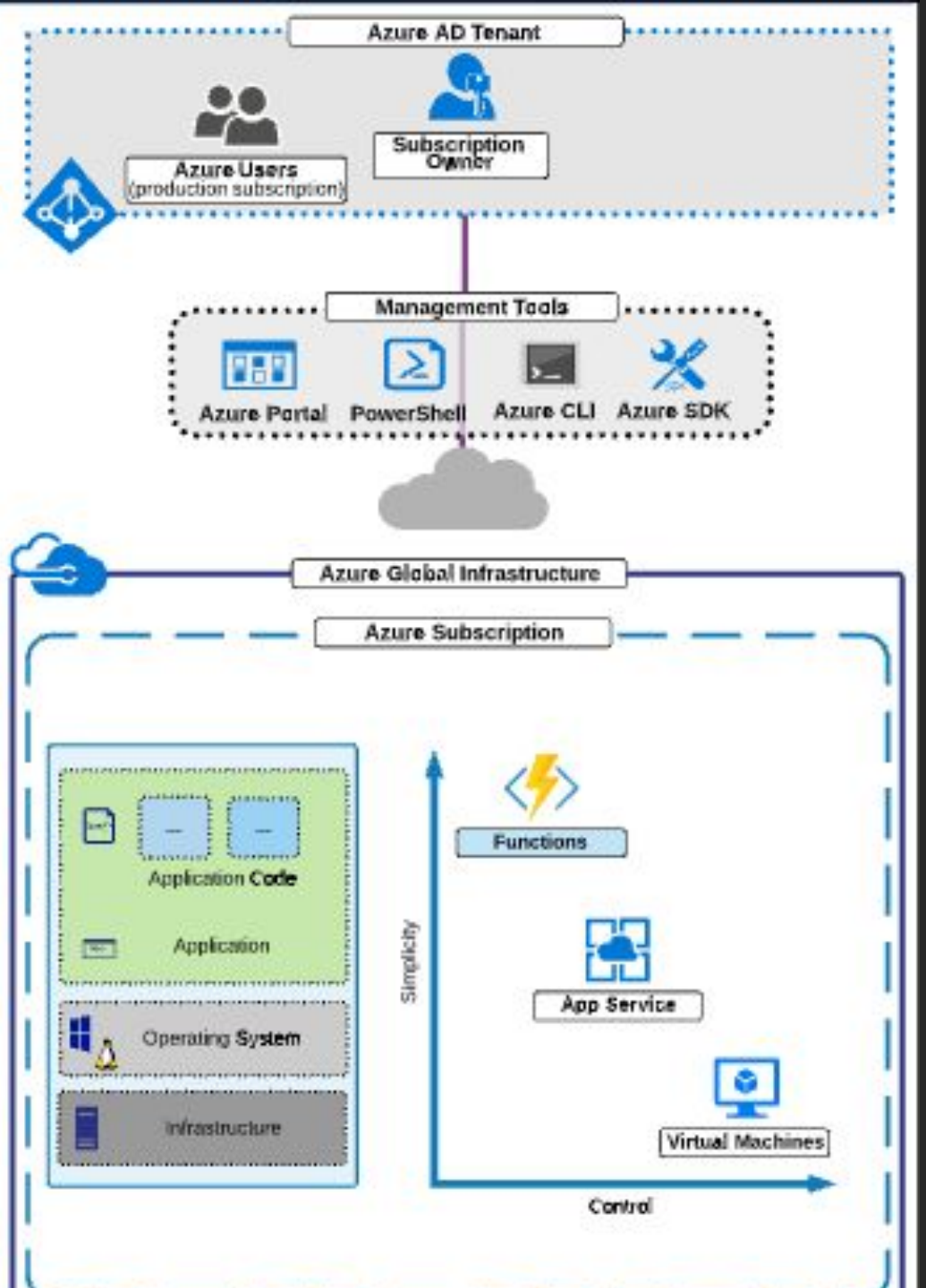
Physical and Networking Layer

Azure Functions

Azure Functions provides us the ability to run a small piece of code, which is focused on an individual task.

This diagram helps to illustrate the level of difficulty and control when we compare Functions to other servers.

Go Back | Appendix



Compute

Web Apps

Close

Web Apps provides a managed hosting solution for deploying scalable web applications atop App Service.

A summary of important Web App features is:

- Functionality will be determined by the underlying App Service / App Service Plan
- Supports Windows and Linux platforms:
  - Windows supports: .Net, Java, Node.js, PHP and Python
  - Linux supports: .Net Core, Node.js, PHP and Ruby
- Can also be configured as Web Apps for containers
- Supports various developer and deployment features such as:
  - Continuous and integrated deployments
  - Support for Git, OneDrive, BitBucket, Azure Repos, etc.
- Provides the capability of integration with on-premises environments using Hybrid Connections

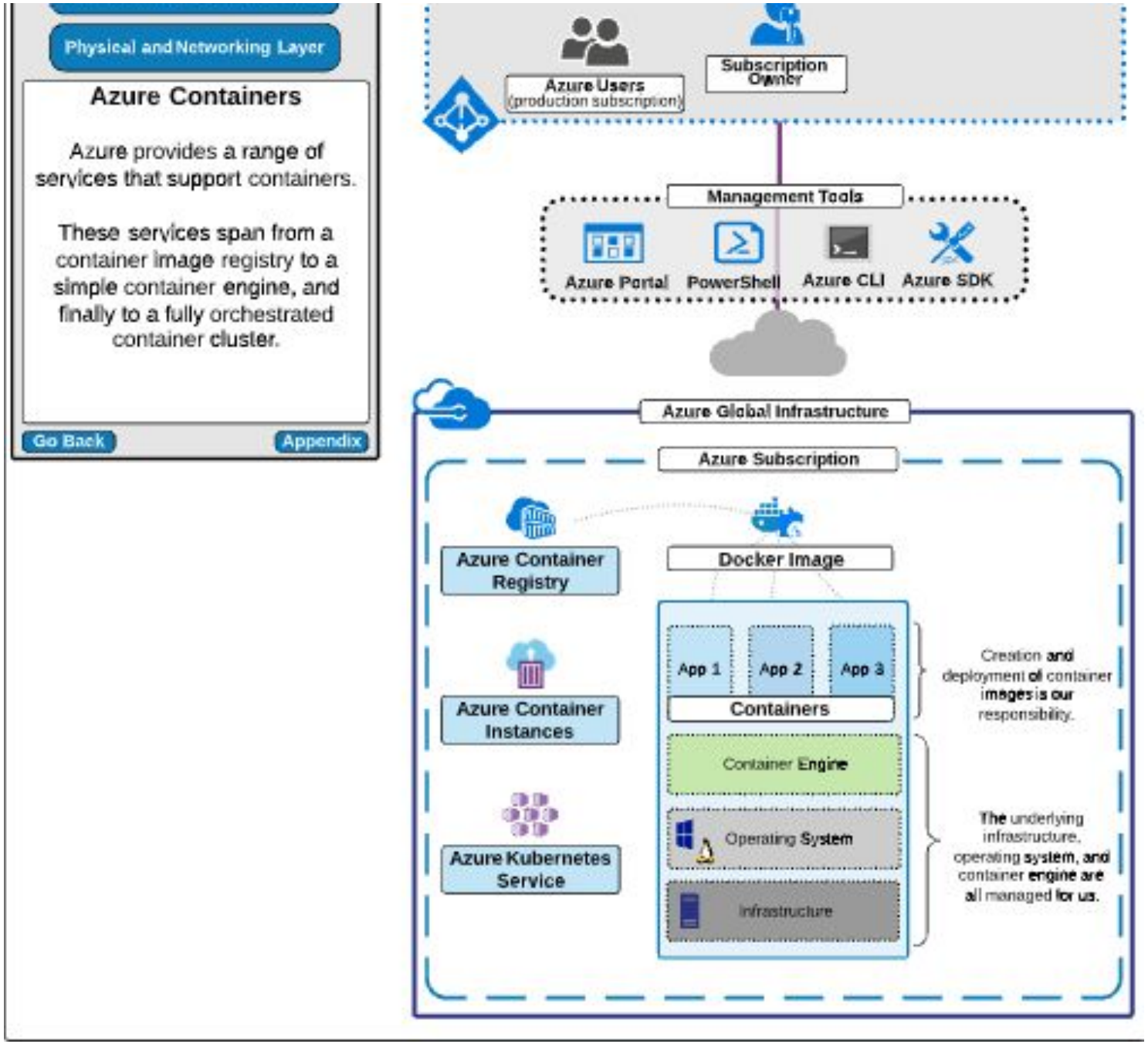
Background tasks using WebJobs

WebJobs are a feature of Web Apps which provide the ability to schedule either continuous or triggered tasks for running in the background of a Web App.

A summary of important information about WebJobs is:

- Continuous jobs:
  - Start immediately once created (and can be restarted if the job ends)
  - Run on all instances that the web app runs on, within the server farm (App Service Plan)
  - Store copies of runtime files in App_Data/jobs/continuous
- Triggered jobs:
  - Only start once triggered manually, or based on a schedule
  - Run only on a single instance within the App Service Plan as determined by Azure
  - Store copies of runtime files in App_Data/jobs/triggered
  - Support scheduling using CRON expressions (refer to this Microsoft article for more information)

Compute


Physical and Networking Layer
Azure Containers
Azure provides a range of services that support containers.
These services span from a container image registry to a simple container engine, and finally to a fully orchestrated container cluster.
Go Back
Appendix
Azure Users
(production subscription)
Subscription Owner
Management Tools
Azure Portal
PowerShell
Azure CLI
Azure SDK
Azure Global Infrastructure
Azure Subscription
Azure Container Registry
Docker Image
Azure Container Instances
App 1
App 2
App 3
Containers
Container Engine
Operating System
Infrastructure
Azure Kubernetes Service
Creation and deployment of container images is our responsibility.
The underlying infrastructure, operating system, and container engine are all managed for us.

Compute

Azure Container Instances

Close

Azure Container Instances (ACI) are one of the fastest options for deploying container applications within Azure, when compared to the other Azure container services.

ACI supports deployment of containers from both public repositories, like a Docker Hub or private repositories, when using Azure Container Registry.

Important information about Azure Container Instances:

- ACI is useful for isolated, simple applications. For multi-container apps, use Azure Kubernetes Service.
- Container groups (for Linux containers only) enable deployment of multiple containers to a single host.

Configuring Azure Container Instances:

- The image to create the container is required using a public or private repository.
- An ACI can be configured as either Linux or Windows, but note that Windows do not support Container Groups.
- ACI is publicly accessible using a unique DNS label *<dns-name-label>.<region>.azurecontainer.io*.
- It will soon be possible to deploy container instances to an Azure VNet. This feature is in preview.
- It is not possible to change the public IP address of container instances.

Compute

Azure Container Registry (ACR)

Close

Azure Container Registry (ACR) provides the ability to store images for your containerized applications. The underlying service is a docker registry based on Docker Registry 2.0, which is fully managed by Microsoft.

ACR also includes Azure Container Registry Build (ACR Build), a suite of tools which helps building container images and automating container development and maintenance processes.

Key features of Azure Container Registry:

- Private storage of docker-formatted images
- Supports tooling similar to Docker Registry 2.0
- Windows and Linux container images are supported within a single registry

Configuration of Azure Container Registry:

- Can be configured as either Basic, Standard or Premium pricing SKU
  - Differs in daily cost, included storage, total webhooks, and support for geo-replication (premium):
    - View current information at this Microsoft pricing article.
- Requires a registry name (accessible at *registryname.azurecr.io*)
- Supports authentication from Azure AD, managed identity, service principal, or an admin account

Compute

Azure Kubernetes Service (AKS)

Close

Azure Kubernetes Service (AKS) is a Microsoft hosted and managed Kubernetes service that allows for easy deployment of a Kubernetes cluster to Azure.

Kubernetes is an open-source solution which helps manage multi-container environments. This includes scalability, automated deployments, and resource management for containerized applications. Kubernetes helps provide a focus on application workloads by abstracting the underlying infrastructure.

Important information about AKS:

- Deploying an AKS cluster results in two key components:
  - The Kubernetes cluster master:
    - This is managed by Microsoft, and does not incur any cost.
    - It includes kube-apiserver, etcd, kube-scheduler, and kube-controller-manager.
  - The Kubernetes nodes:
    - They are managed by you and *do* incur costs.
    - Microsoft recommends a minimum of 3 nodes for resiliency.
    - Node count can be changed, but node size cannot.
- To interact with your AKS cluster use:
  - The Azure portal and CLI
  - The `kubectl` command-line client
  - The Kubernetes dashboard (by using `az aks browse`)

Configuring AKS:

- Creating an AKS cluster requires several details:
  - A name for your Kubernetes cluster
  - The version of Kubernetes you wish to deploy
  - A DNS name prefix for connecting to the Kubernetes API
  - VM size to be used for your cluster nodes (cannot be changed after creation)
  - The number of nodes for your cluster
  - An Azure AD Service Principal (used to create other Azure resources such as load balancers)
- AKS is deployed to two resource groups:
  - One which contains the Kubernetes service resource
  - Another (created by AKS) which contains all infrastructure associated with the cluster

Compute

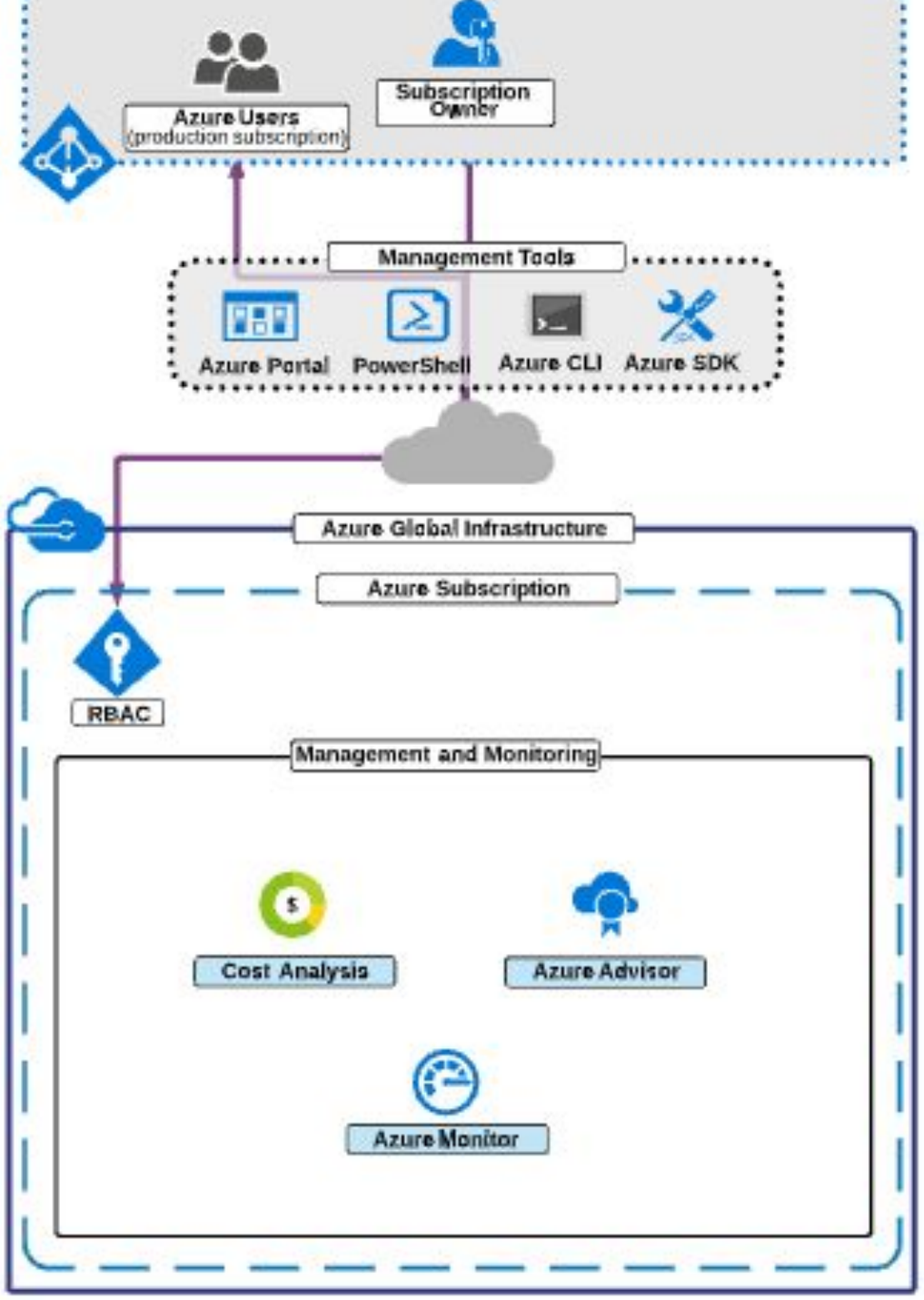
Physical and Networking Layer

Monitoring and Management

Microsoft provides a range of tools to manage and monitor your environment.

In this page you can find a link to the main services you must be familiar with for the AZ-300 exam.

Appendix



Monitoring

Subscription and Services Layer

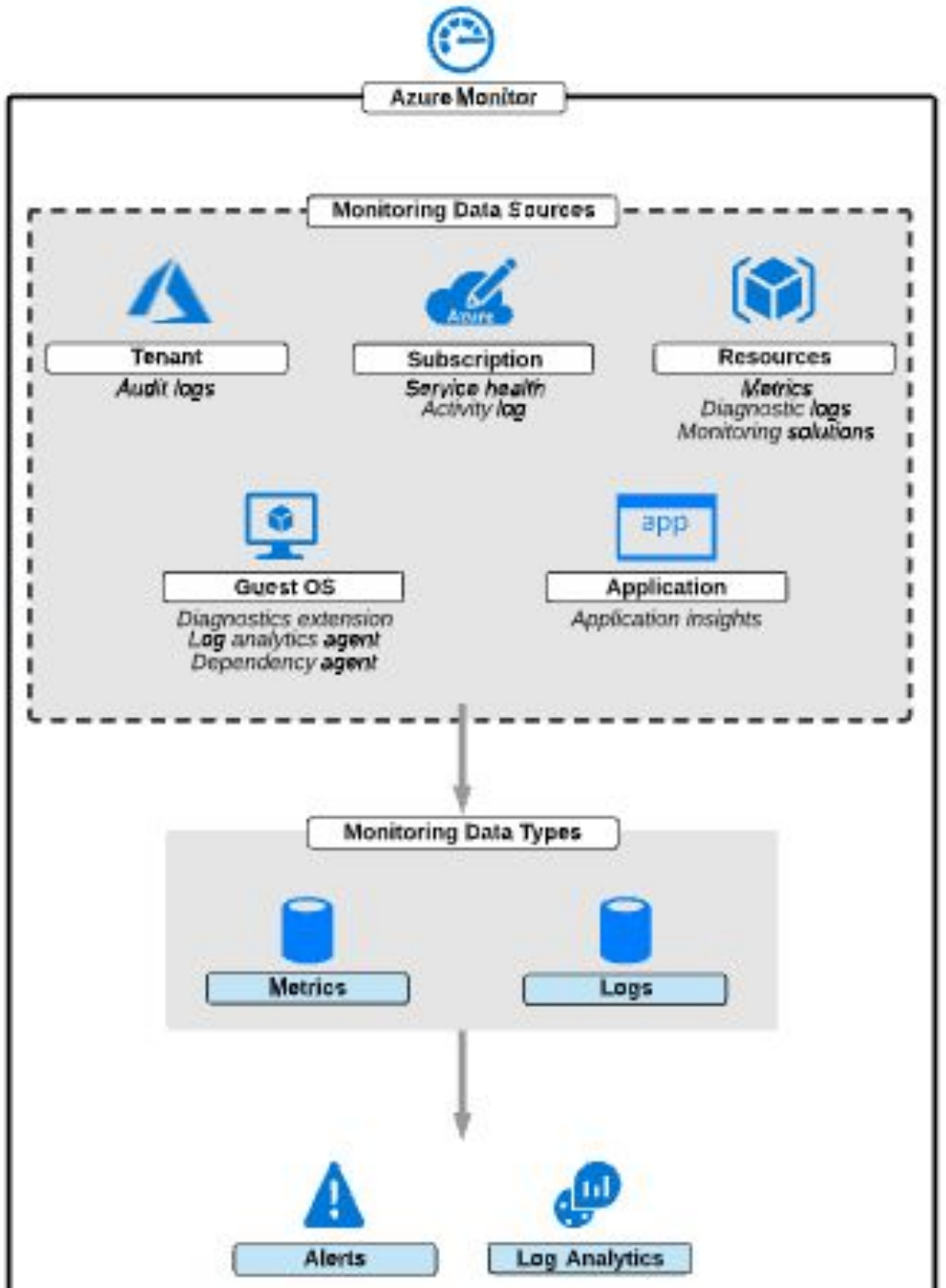
Physical and Networking Layer

Azure Monitor

Azure Monitor is a collection of services which help monitor and diagnose Azure resources.

This page provides an overview of Azure Monitor; where data is collected from, the two fundamental types of data, and finally some of the main functions within the Azure Monitor toolkit.

Go Back | Appendix



Monitoring

Metrics

Close

Metrics are lightweight numerical values that describe something about a resource over a given time period.

A number of metrics are provided by the Azure platform and are available by default, similar to log data. Additional metrics are also available from other sources, like application metrics via the Application Insights instrumentation, or OS-specific metrics via the diagnostics VM extension.

Examples of metrics:

- Average CPU utilization (as a percentage over time)
- Network throughput (kilobytes per second)

Important information about metrics:

- Metrics are collected at a frequency of one-minute by default and stored for 93 days
- There are many metrics available by default with various Azure resources
- Some additional metrics require further configuration (e.g. VM Diagnostics Extension) before they are available
- Metrics can be viewed through Azure Monitor, and also within the resource tab itself

More information can be found at this Microsoft article:
https://docs.microsoft.com/en-us/azure/azure-monitor/platform/data-collection#metrics

Monitoring

Logs - Overview

Close

Overview | Diagnostic Logs | Activity Logs

Logs are larger sets of textual data, containing detailed descriptions and information about a specific resource, application, or activity. Data is typically collected sporadically.

Examples of logs:

- Activity logs from Azure resources
- Detailed diagnostics collected from Windows and Linux VM guest operating systems and applications
- Resource-level diagnostic logs provided through the Azure platform
- Custom log data

https://docs.microsoft.com/en-us/azure/azure-monitor/platform/tutorial-dashboards

More information can be found at the following Microsoft article:
https://docs.microsoft.com/en-us/azure/azure-monitor/platform/data-collection#logs

Monitoring

Logs - Diagnostic Logs

Close

Overview | Diagnostic Logs | Activity Logs

Diagnostic logs is typically rich and frequent data that details the operation of a given service or application. Exactly what they look like differs on a case-by-case basis.

There are three types of diagnostic logs:

- Tenant logs: Logs from outside of an Azure subscription (e.g. Active Directory logs)
- Resource logs: Logs emitted by resources deployed within an Azure subscription (e.g. storage accounts)
- OS-level logs: Logs collected by an agent running inside a compute resource (e.g. virtual machine)

Important information about diagnostic logs:

- Azure Monitor is able to provide diagnostic logs for a number of resources, with minimal configuration.
  - Logging for these resources can be enabled within Azure Monitor, in *Diagnostic Settings*.
  - These logs are collected by the Azure platform itself.
  - More information on supported resources can be found here: https://docs.microsoft.com/en-us/azure/azure-monitor/platform/tutorial-dashboards
- OS-level logs require an additional agent and configuration, in order to capture the log data
  - For virtual machines, this agent is installed using a VM extension

What happens to the logs?

- Azure Monitor uses *diagnostic settings* to configure three possible destinations:
  - A storage account, for archival
  - Log Analytics, for search and analytics
  - Event Hubs, to be integrated with other solutions
- When using the OS-level Azure Diagnostics Extension, data is stored in a specified storage account

Monitoring

Logs - Activity Logs

Close

Overview | Diagnostic Logs | Activity Logs

Activity Logs, previously known as *Audit Logs* or *Operational Logs*, provide information about events that have occurred across a subscription. These logs are used to determine "who, what, and when" for operations performed on resources within a subscription.

Important information about Activity Logs:

- Subscription level log for write operations (PUT, POST, DELETE)
- Does not include read operations (GET)
- Does not include operations for resources that use the classic/"RDFE" model

There are a range of sources for Activity Logs:

- Administrative: Create, update, and/or delete actions performed through Azure Resource Manager
- Service Health: Service health incidents that have occurred in Azure
- Resource Health: Resource health events for resources (available, unavailable, degraded, unknown)
- Alert: A record of any Azure alerts activating
- Autoscale: Events relating to the operation of any autoscale settings you have defined
- Security: Alerts generated by Azure Security Center
- Policy: Reserved for future use by Microsoft

More information can be found here:
https://docs.microsoft.com/en-us/azure/azure-monitor/platform/activity-logs-overview.

Monitoring

Alerts - Overview

Close

Overview | Action Groups

Alerts are a feature of Azure Monitor which provide the ability to *perform some action(s)* in response to *something happening*.

In order to configure alerts, we must create an **alert rule** (including a name and description), which defines an **action group** to trigger, based on a **condition** being met for a given **resource**.

Important information about alerts:

- Alert conditions can only be based on two signal types: metrics, or activity logs.
- It's important to understand limits associated with the actions we can perform
  - Refer to the Action Groups tab above
- Triggered alerts are visible within *Azure Monitor* > *Alerts*, where they can be managed
  - Alert state management allows us to use three state alerts: new, acknowledged, and closed
  - Alert states are independent of the underlying monitoring condition
  - Historical information about the alert states, and monitor condition, are recorded

Monitoring

Alerts - Action Groups

Close

Overview | Action Groups

Action groups are used within alerts to define *what should happen* when an alert is triggered. This can include emailing users, calling an Azure Function, and much more.

Below is a list of the types of actions that can be performed, as well as the limits per action group.

Action groups (limits per action group displayed in parenthesis):

- Email (up to 1000): Email notifications to an email address
- SMS (up to 10): SMS notifications to a phone number
- Push (up to 10): Push notification to the Azure app
- Voice (up to 10): Voice calls to a phone number
- Azure Function: Call to a Function app
- LogicApp (up to 10): Call to a Logic app
- Webhook (up to 10): URI call to a webhook
- ITSM (up to 10): Call to supported ITSM via an ITSM Connection
- Automation Runbook (up to 10): Call to a built-in or user Runbook

Rate limiting for certain notification types:
Microsoft imposes certain rate limits to ensure that alerts are manageable. To achieve this, notifications may be suspended if too many are sent to a particular phone number, email address, or device.

The rate limit thresholds are:

- SMS: No more than 1 SMS every 5 minutes
- Voice: No more than 1 voice call every 5 minutes
- Email: No more than 100 emails in an hour

Refer to the following Microsoft article for more information:
https://docs.microsoft.com/en-us/azure/azure-monitor/platform/alerts-rate-limiting

Monitoring

Log Analytics - Overview

Close

Overview | Log Search

Log Analytics is a service within Azure Monitor which helps to centralize log management for a variety of Azure services, resources, and other log data.

Log Analytics provides the storage and search functionality of logs within Azure Monitor. To send data to Log Analytics, **Diagnostic Settings** must be configured for a given resource.

Important information about Log Analytics:

- Log Analytics provides uses a powerful query language, based on the Data Explorer query language.
- All log data collected by Azure Monitor is stored within a Log Analytics workspace:
  - A workspace must be created for log data to be stored within Log Analytics.
  - All Log Analytics queries are scoped to the respective workspace by default.
  - Workspace data sources can include Azure resources, storage accounts, activity logs, VMs, etc.
- Historically, Log Analytics was a standalone service, however it is now part of Azure Monitor

Important information about Log Analytics data sources:

- Data sources can include Azure resources, activity logs, storage accounts logs, and virtual machines
- Diagnostic Settings are used to configure resources to send data to a Log Analytics workspace

Monitoring

Log Analytics - Log Search

Close

Overview | Log Search

Log Analytics queries provide a way to perform powerful analysis of large volumes of log data. Microsoft manages the underlying analysis infrastructure so you do not have to (based on the Data Explorer service).

Below are some tips for log queries:

- Log Analytics data is organized in tables, which can be queried by piping the table through to an operator
- The below query shows a maximum of 50 results from the SecurityEvent table

```
SecurityEvent | limit 50
```

- The below query searches the Azure Activity Log for evidence of Storage Account keys being changed

```
AzureActivity | where OperationName == "Regenerate Storage Account Keys"
```

Queries can be saved as functions to simplify advanced queries

- For example, if the preceding query string is saved as "sakeyregen", then we could re-use it as follows:

```
sakeyregen | summarize count(Caller) by Caller
```

  - This would take the output of the earlier command, and pipe it to the second command.
  - The result of this query is to list and count all distinct users who have regenerated a storage account key.

More information:
The log query language is very flexible and powerful. Consequently, it can either be simple or complex. For this reason it is recommended you are familiar with the following article:

https://docs.microsoft.com/en-us/azure/azure-monitor/log-query/get-started-queries

Monitoring

Subscription and Services Layer

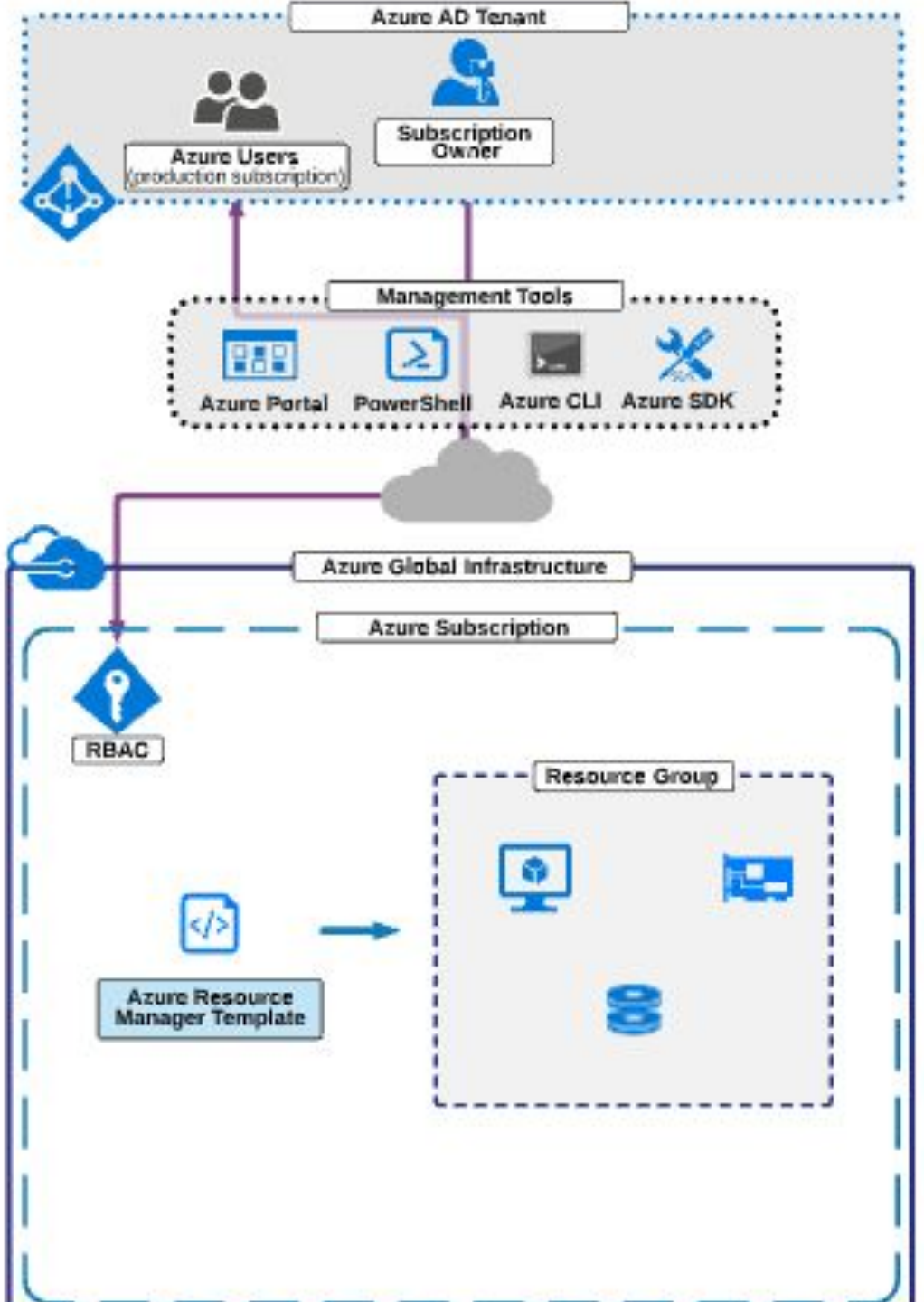
Physical and Networking Layer

Automated Deployments

At the core, of automated deployments within Azure, is the Azure Resource Manager (ARM) Template. This template allows infrastructure to be defined as code.

Using ARM Templates, we can automate the deployment of infrastructure with scripts or code.

Appendix



Deployment

Azure Resource Manager (ARM) Templates

Close

Overview | Structure | Example

Azure Resource Manager (ARM) Templates essentially allow us to use text to represent infrastructure. That is why this form of deployment is often called infrastructure-as-code.

General Information on ARM Templates

- ARM Templates use **declarative JSON syntax** to represent the infrastructure as code/text.
- We can automate the deployment of infrastructure using these templates with scripts or code.
- ARM Templates have a special structure (detailed more within the *Structure* tab above) allowing us to define multiple resources which are ***deployed to a Resource Group***.

Information about saving ARM Templates

- You can create and save your own ARM Templates to JSON files, and deploy them as illustrated below.
- Existing resources can be exported to ARM Templates via the portal:
  - Original template: Navigate to a Resource group, then **Deployments**, where you can view details/files for past deployments
  - Generated template: Navigate to a Resource, then **Automation script**, to view what was automatically generated

Information about deployment modes

- Incremental and Complete are the two modes that deal with *already existing* Resources in a Resource Group that you deploy an ARM Template to.
- Whichever mode is used, resources defined in the ARM Template, resources that already exist, will have their properties updated to match the ARM Template.
  - **Incremental mode:** Existing resources will be left unchanged if they are not specified in the template.
  - **Complete mode:** Resources in the Resource group but not the template *will be deleted*.

Scripted deployment (using Azure CLI and PowerShell):

```
az group deployment create \
  --name deploymentTest \
  --resource-group lab01rg \
  --template-file vmdeploy.json \
  --parameters vmName=testvm01

New-AzureRmResourceGroupDeployment -Name deploymentTest `
  -ResourceGroupName lab01rg `
  -TemplateFile c:\lab\vmdeploy.json `
```

Deployment

ARM Template Structure

Close

Overview | Structure | Example

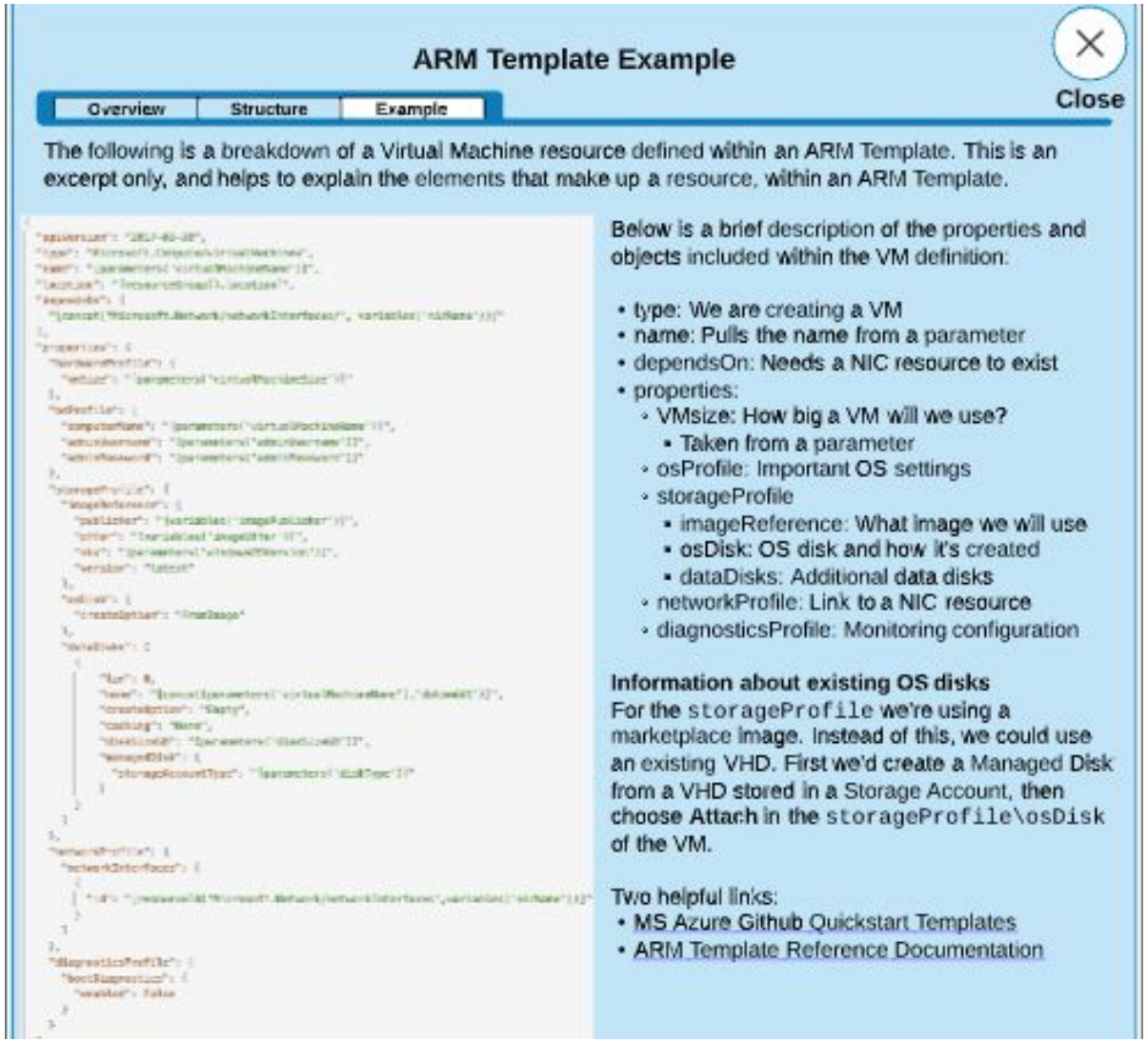
ARM Templates are very versatile, so template files can either be large and complex, small and simple, or anywhere in between. At the very core, all ARM Templates are made up of the following elements:

- **$schema and contentVersion:** Required details describing the version and format of the template:
  - These values rarely change and are primarily used by Microsoft to describe the template structure.
- **Parameters:** Custom values you can use at deployment time to change what is being deployed:
  - Parameters help simplify your template by allowing you to have a single template with values which may change from time to time, rather than creating multiple templates with fixed/unchangeable values.
    - For example, you might have VMNAME or OPERATINGSYSTEM as parameters.
- **Variables:** Values used within the template, largely to help simplify the template usability and readability:
  - Variables can help having to re-write (or even re-calculate) the same piece of information repeatedly.
    - For example, a variable might include a reference to a subnet, in which you will deploy multiple VMs.
- **Functions**: User-defined functions which you can create and call within a template
- **Resources**: The very specification of the resource(s) you wish to deploy with the template
- **Outputs:** Values you wish to be returned after deployment

Note that `$schema`, `contentVersion`, and `resources` are always required within an ARM Template. The other elements do not have to be used.

```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
  },
  "variables": {
  },
  "functions": [
  ],
  "resources": [
  ],
  "outputs": {
  }
}
```

Deployment


ARM Template Example
Close
Overview
Structure
Example
The following is a breakdown of a Virtual Machine resource defined within an ARM Template. This is an excerpt only, and helps to explain the elements that make up a resource, within an ARM Template.
Below is a brief description of the properties and objects included within the VM definition:
• type: We are creating a VM
• name: Pulls the name from a parameter
• dependsOn: Needs a NIC resource to exist
• properties:
◦ VMsize: How big a VM will we use?
▪ Taken from a parameter
◦ osProfile: Important OS settings
◦ storageProfile
▪ imageReference: What image we will use
▪ osDisk: OS disk and how it's created
▪ dataDisks: Additional data disks
◦ networkProfile: Link to a NIC resource
◦ diagnosticsProfile: Monitoring configuration
Information about existing OS disks
For the storageProfile we're using a marketplace image. Instead of this, we could use an existing VHD. First we'd create a Managed Disk from a VHD stored in a Storage Account, then choose Attach in the storageProfile\osDisk of the VM.
Two helpful links:
• MS Azure Github Quickstart Templates
• ARM Template Reference Documentation

Deployment

Subscription and Services Layer

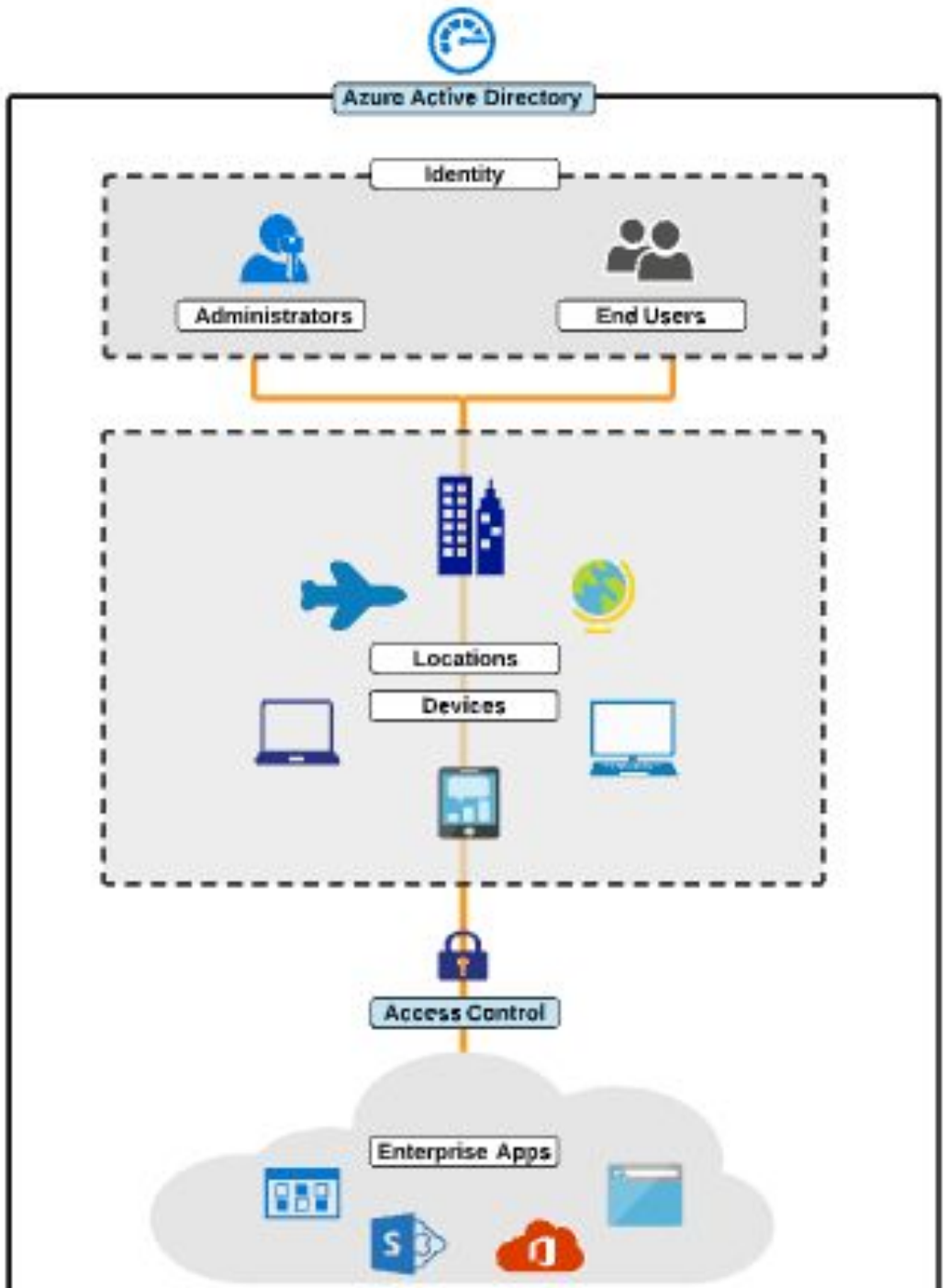
Physical and Networking Layer

Identity and Access Management (IAM)

At the core of IAM within Azure (and other Microsoft services) is Azure Active Directory (AD).

This page depicts the real world scenario that Azure AD helps to address: managing many users accessing many applications from many locations.

Appendix



IAM

Subscription and Services Layer

Physical and Networking Layer

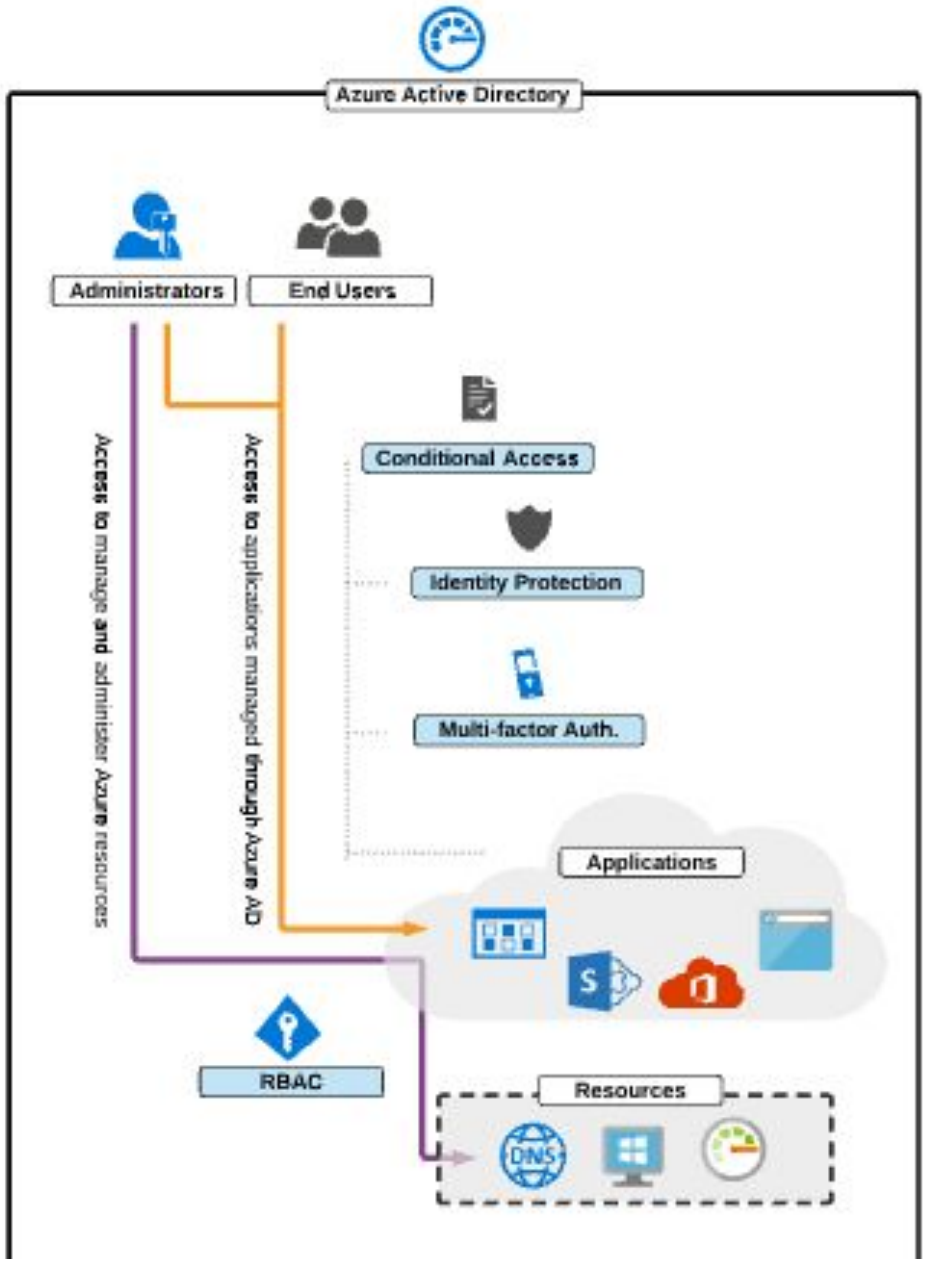
Access Control

Azure AD provides features which help secure access to resources.

This page illustrates the main services available to protect Azure AD managed resources.

Role-based access control (RBAC) manages direct access to Azure resources.

Go Back | Appendix



IAM

Role-based Access Control (RBAC)

Close

While Conditional Access and Identity Protection are used to control access to Azure AD managed resources, RBAC is used to provide granular access to Azure resources themselves.

These roles can be assigned at the subscription, resource group, or resource level.

Important information about RBAC:

- Azure includes a range of over 70 **built-in roles** for controlling access to Azure resources:
  - **Owner:** Includes full access to the assigned resource(s) including rights to grant access to others
  - Contributor: Provides full access to the assigned resource(s) **except** rights to change permissions
  - Reader: Provides full view access to the assigned resource(s), but no ability to make changes

  For more information, refer to the article on built-in roles for Azure resources.

- If the built-in roles are not sufficient, **custom roles** can be created.
- For roles to take affect, they must be assigned:
  - Roles are assigned to an Azure AD user, group, or service principal
  - They must be assigned to something: a subscription, resource group, or resource
- Permissions granted through roles can be inherited (e.g. subscription owner owns all resources).
- Roles are versatile, and can be used for a range of access, for example:
  - Granting systems administrators access to all resources, except for billing
  - Granting service desk staff the ability to restart virtual machines only
  - Granting the finance team with access to billing, and support, but nothing else
- RBAC applies to Azure Resources only, but Conditional Access and Identity Protection can still block access to the Microsoft Azure Managent app, which includes all Azure management endpoints.

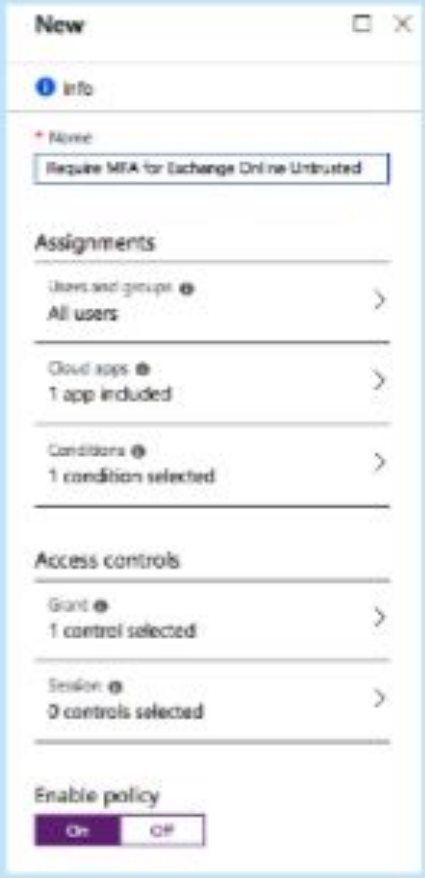
IAM

Conditional Access

Close

Conditional Access provides the ability to configure a range of different access or deny rules to Azure AD managed resources, depending on a range of conditions.

For example, you could configure a Conditional Access Policy (CAP) that requires multi-factor authentication for all users when they sign into Microsoft Office 365 Exchange Online, unless they're signing in from a company's head-office.



Creating a new Conditional Access Policy:

The image to the left shows a new CAP being created within the Azure Portal. Below is an overview of key parameters of a CAP.

Assignment:
Each CAP defines a range of criteria which determine who, what, and under which circumstances a CAP will apply:

- Users and groups: Which users or groups the CAP applies to
- Cloud apps: Enterprise Applications, e.g. the Azure Portal
- Conditions: Risk (Azure AD ID protection), devices, locations

Access controls:
The access controls define whether access to the cloud app will be granted or blocked, and additional conditions which apply:

- Grant: Block, or grant (can also require other conditions like MFA)
- Session: Ability to enable limited experience access to an app

Best practices and policy processing:

- Block will always take precedence when multiple policies apply
- All access requirements must be met when multiple policies apply
- It is recommended that care is taken when using all users/groups
- For more information refer to this Microsoft article

IAM

Azure AD Identity Protection - Overview

Close

Overview | Risk Events | Vulnerabilities

Azure Active Directory (AD) Identity Protection is a service provided by Microsoft. It is used to intelligently identify risks and vulnerabilities affecting your Azure AD identities. Identity Protection is powered by adaptive machine learning which detects anomalies and suspicious incidents relating to identity.

To enable Azure AD Identity Protection:

- You must have Azure AD Premium P2 licensing
- You must also create Azure AD Identity Protection, as a new resource

A summary of Identity Protection capabilities is as follows:

- Detect vulnerabilities and risky accounts based on various patterns and heuristics
- Policies which can be configured to:
  - Mitigate risky sign-ins by blocking sign-ins entirely or requiring multi-factor authentication
  - Block or secure risky user accounts
  - Require users to register for multi-factor authentication

Policies within Identity Protection:

It is possible to configure policies within Identity Protection itself, and/or Conditional Access Policies. The policies within Identity Protection which can be configured are as follows:

- Multi-factor authentication (MFA) registration policy:
  - Policy for enforcing user MFA registration,
  - Allows more granular control over registration compared to standard Azure MFA settings
- User risk policy:
  - Policies applying to accounts which Identity Protection has identified as being at risk
  - Allows or denies access based on risk level, and can enforce a password change
- Sign-in risk policy:
  - Policies get applied in real-time, during sign-in, and can therefore be used to prevent sign-in.
  - They allow or deny access based on risk level, and can enforce the use of MFA.
  - These apply to browser and modern-auth traffic, but not apps which use older security protocols.

IAM

Azure Multi-Factor Authentication (MFA)

Close

Microsoft provides a fully-managed multi-factor authentication (MFA) solution, which is a feature of Azure AD. This is often referred to as Azure MFA, or Cloud-based Azure MFA. This differs to MFA Server, which requires additional infrastructure to be deployed and managed.

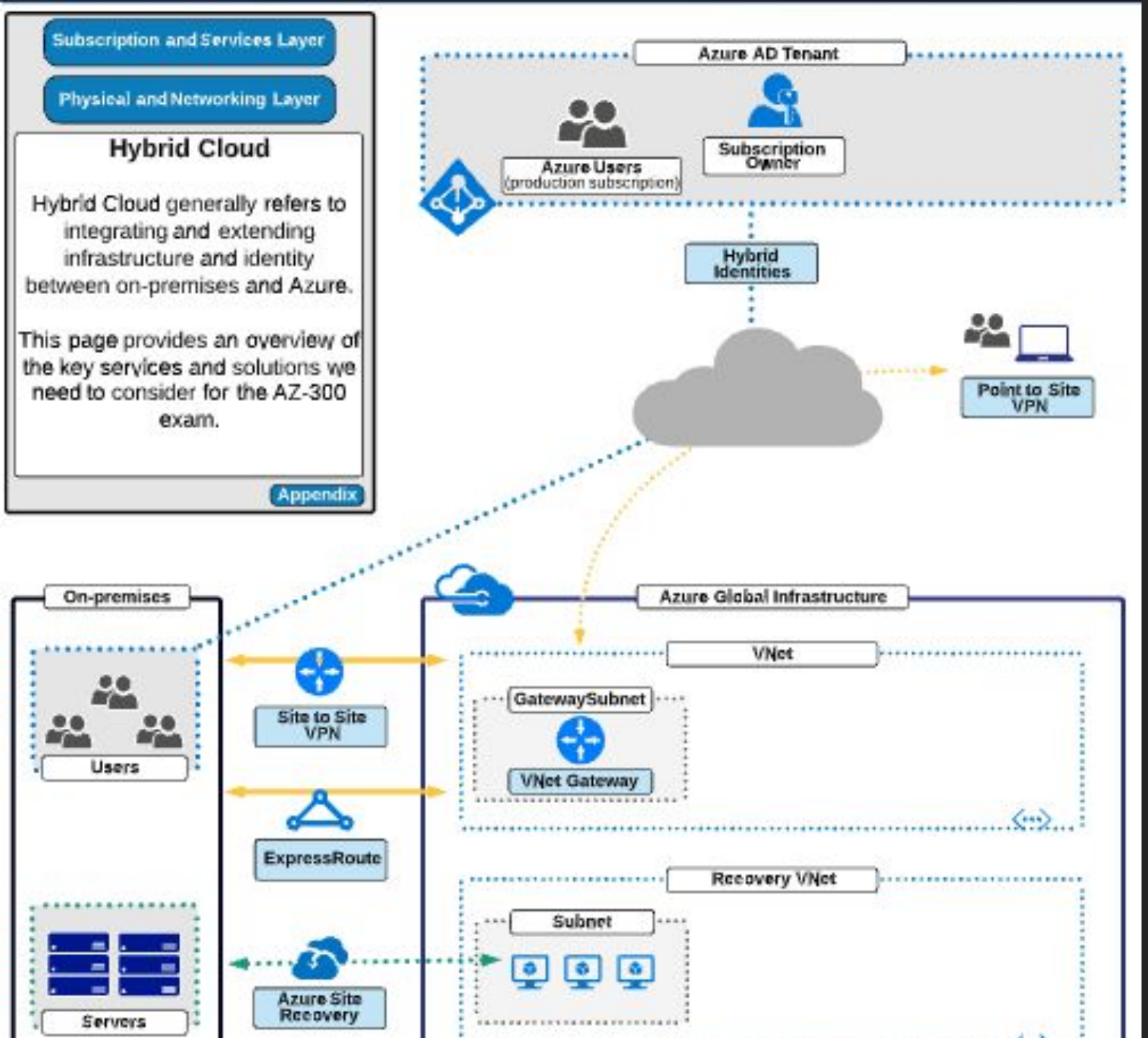
Refer to this Microsoft article for a summary of differences between the two versions of MFA.

The purpose of MFA:
We use MFA as additional security to help authenticate a user. A common example of MFA is logging in to a bank with your credentials (username + password) as well as a code from an SMS. The terminology "multi-factor" refers to the need for multiple things being required to login: what you know (credentials) and what you have (access to your mobile).

Important information about Azure MFA:

- Azure MFA can be enabled in a number of ways, including:
  - Individually (through Azure cloud based MFA settings)
  - Conditionally (through Identity Protection, or Conditional Access Policies)
- Ideally a user should enroll and setup MFA *before* they are forced to authenticate with it, otherwise they may experience difficulties authenticating with apps (https://aka.ms/mfasetup)
- A range of verification options are available to users, including:
  - Call to a phone
  - Text message to a phone
  - Notification through mobile app
  - Verification code from mobile app or hardware token
- MFA can be bypassed:
  - According to specific criteria, through Conditional Access Policies
  - Using one-time bypass (for MFA Server) for a set amount of time
  - When the authentication request comes from a trusted IP

IAM


Subscription and Services Layer
Physical and Networking Layer
Hybrid Cloud
Hybrid Cloud generally refers to integrating and extending infrastructure and identity between on-premises and Azure.
This page provides an overview of the key services and solutions we need to consider for the AZ-300 exam.
Appendix
Azure AD Tenant
Azure Users
(production subscription)
Subscription Owner
Hybrid Identities
Point to Site VPN
On-premises
Azure Global Infrastructure
Users
Site to Site VPN
VNet
GatewaySubnet
VNet Gateway
ExpressRoute
Recovery VNet
Subnet
Servers
Azure Site Recovery

Hybrid

VPN Gateway - Overview

Close

Overview | Site-to-site | Point-to-site

VPN Gateways provide private and encrypted connectivity between an Azure Virtual Network (VNet) and one or more remote resources. Technically this is a VNet Gateway configured with **gateway type = vpn**.

In this configuration, the VNet Gateway (often referred to as a VPN Gateway) provides support for:

- Site-to-site (S2S) VPN: IPsec/IKE encrypted VPN tunnel with remote sites over the internet
- Point-to-site (P2S) VPN: IKEv2 or SSTP VPN with remote clients over the internet
- VNet-to-VNet: IPsec/IKE encrypted VPN tunnel with another VNet

VPN Gateways have the following key configuration requirements:

- VPN Gateways support the following SKUs:
  - Basic - A legacy SKU with a number of limitations
  - VpnGw1, VpnGw2, VpnGw3 - Differ based on connection limits and bandwidth
  - Inclusions regularly change, so for up-to-date information, see this Microsoft article
- VPN type:
  - Route-based (dynamic), which typically allows all traffic, and relies on route-tables to determine what networks are available for connecting, through the VPN
  - Policy-based (static), which includes network routes/prefixes being hard-coded in to the VPN connection itself
- All VPN Gateways must also be configured with the following related items:
  - Deployed to a **gateway subnet**, which should:
    - Be named "GatewaySubnet"
    - Only contain VNet Gateways
    - Have an address space sufficient to hold all VNet Gateways you will create
  - Linked to a **dynamic public IP address**

Refer to the Microsoft VPN Gateway FAQ for more details about VPN Gateways.

Multi-Site VPN connections:
Creating multiple concurrent S2S connections from one VPN Gateway to many different remote sites over the internet is possible. This is referred to as multi-site, and **requires** a route-based VPN type.

A Note about VNet-to-VNet connections:
VNet-to-VNet is very similar to S2S connections, with the main difference being that the Local Network Gateway address space is automatically created and populated.

Hybrid

Site-to-Site (S2S) VPN

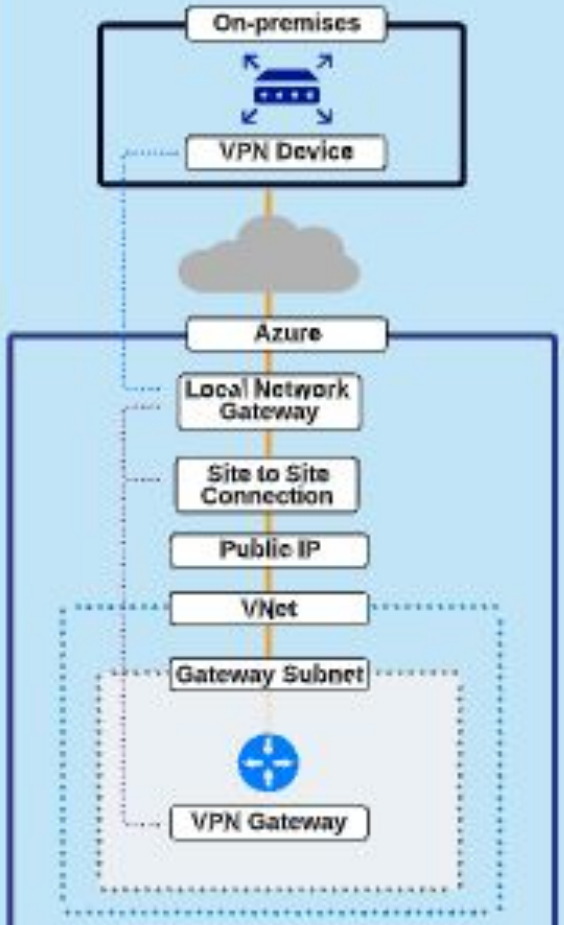
Close

Overview | Site-to-site | Point-to-site

Site-to-site (S2S) VPNs are configured to connect an Azure Virtual Network (VNet) with a remote site, over private IP addressing. This connection is encrypted with IPsec/IKE (IKEv1 or IKEv2) and relies on a VPN device located at the remote location.

Important information about S2S connections:

- They require a **VPN device** at the remote location with a **public IP, not** located behind NAT.
- Multiple S2S connections can be configured using one VPN Gateway with the RouteBased VPN type.



Configuration overview:

The following independent resources must be configured and associated with the VPN Gateway:

- Public IP: Address of VPN gateway (dynamic)
- Local Network Gateway (LNG): Defines the destination network
  - IP address of remote VPN device
  - Address space(s) available at remote site
  - BGP details if desired/required for remote site routing

Configuring a VPN Gateway

- Set up a VNet Gateway, including these main properties:
  - Configured with type=**vpn**
  - VPN-type: Either route or policy-based
  - SKU: Select any, according to needs
  - Must be associated to a public IP resource
  - Must be deployed to a gateway subnet

The Connection resource is used to establish the connection. This ties the VPN Gateway and the Local Network Gateway together:

- Connection: Establishes an encrypted tunnel
  - Connection type: S2S IPsec
  - Local Network Gateway: Associated to the LNG above
  - Shared key (PSK): Used for establishing the connection
- The connection resource includes downloadable VPN device

Hybrid

Point-to-Site (P2S) VPN

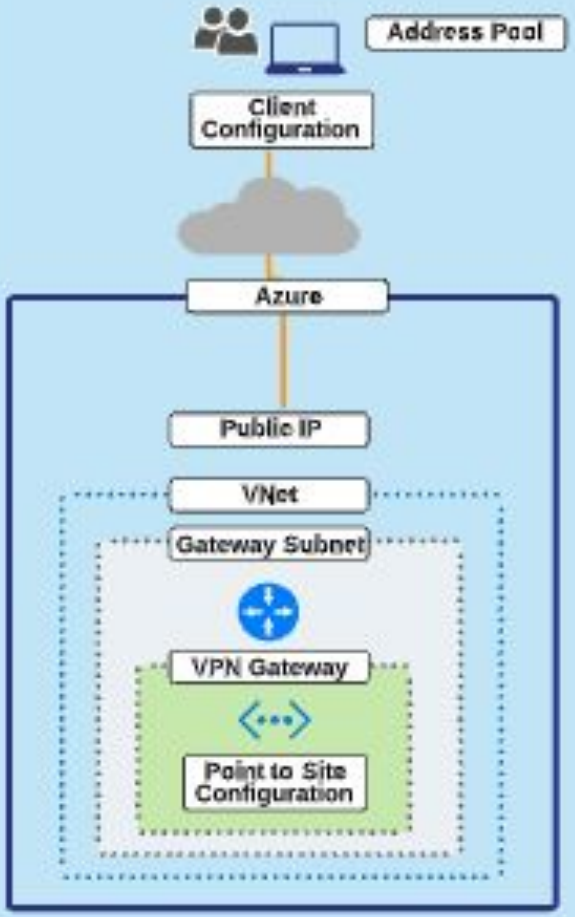
Close

Overview | Site-to-site | Point-to-site

Point-to-site (P2S) VPN allows a secure connection to be established between an individual client computer and an Azure Virtual Network (VNet). This supports Windows, Mac, and Linux.

Important information about P2S configuration:

- P2S supports a range of encryption, including IKE, IKEv2, OpenSSL, and SSTP.
- Client authentication can be configured through either certificates or RADIUS.
- The VPN-type of the VPN gateway must be route-based (dynamic).



Configuration overview:

The following independent resources must be configured and associated with the VPN Gateway:

- Public IP: Address of VPN gateway (dynamic)

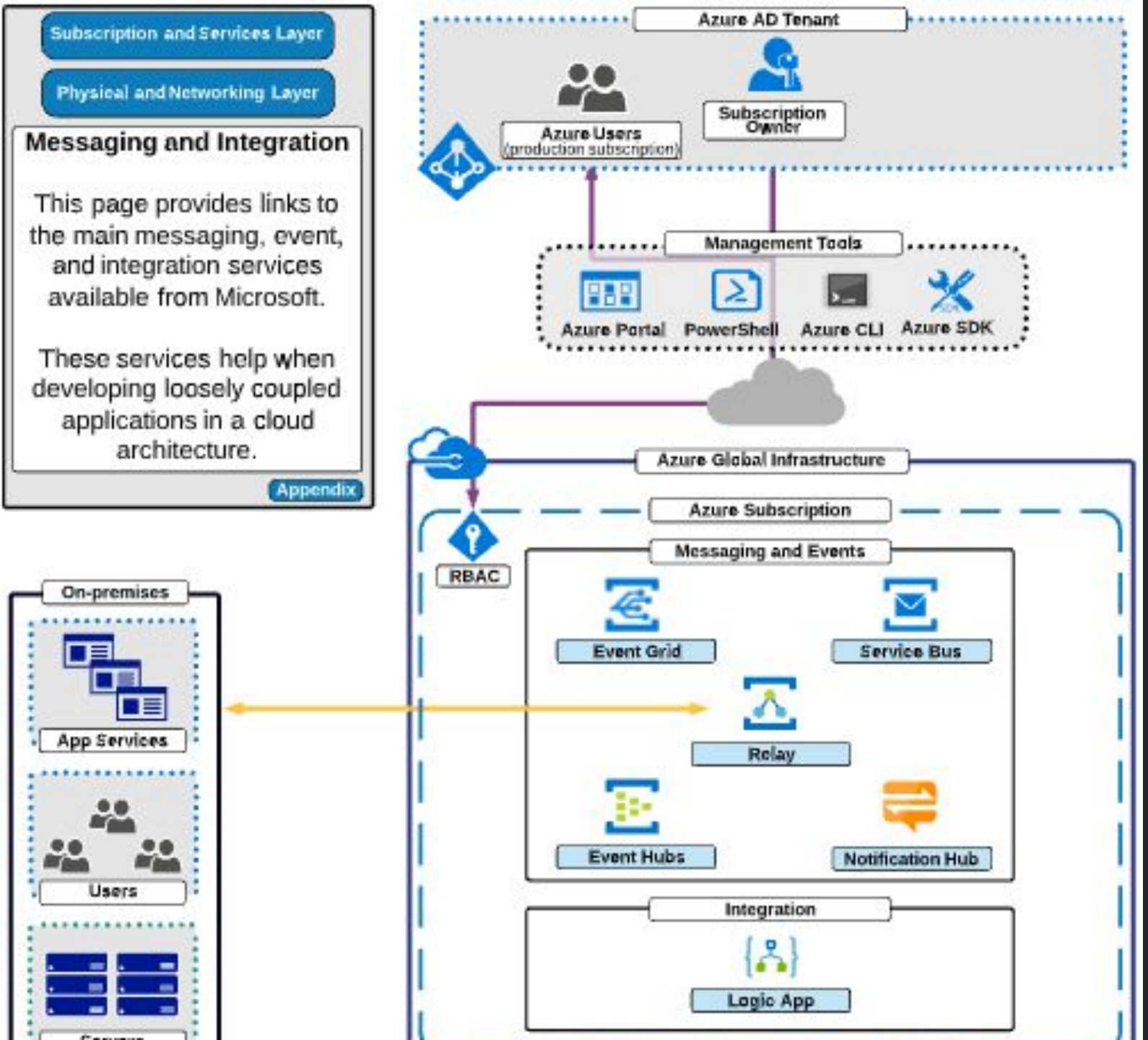
Configuring a VPN Gateway

- Set up a VNet Gateway, including these main properties:
  - Type: **vpn**
  - VPN type: **route-based** (dynamic)
  - SKU: Any will work, according to needs
  - Must be associated to a public IP resource
  - Must be deployed to a gateway subnet
  - P2S Configuration (allows client computer connectivity)
    - Address pool: IP addressing to assign to clients
    - Tunnel type: for example SSL, IKEv2
    - Auth. type: Azure certificate or RADIUS

Key information about client configuration as follows:

- Depends on the authentication and tunnel type
- VPN client files can be downloaded from the portal

Hybrid



Messaging & Integration

Subscription and Services Layer

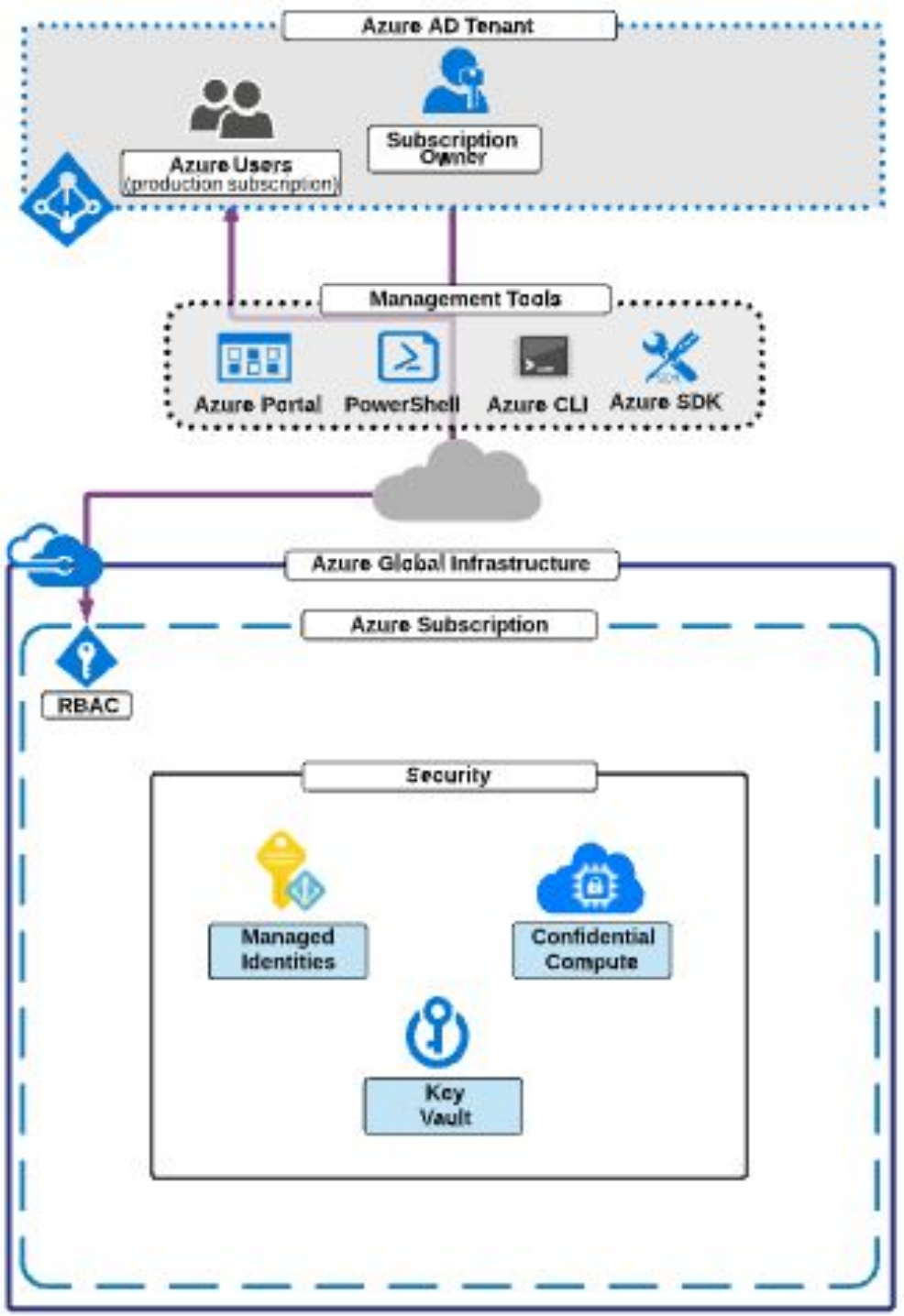
Physical and Networking Layer

Security

Azure provides a number of different services to protect your resources within Azure.

This page lists a number of the main services you should be familiar with for AZ-300.

Appendix



Security

Key Vault

Close

Key Vault is a service which allows for secure storage, and retrieval, of cryptographic keys and other secrets. It is programmatically accessible, and stores secrets in hardware security modules (HSM).

High level summary of Key Vault features is as follows:

- Allows you to store a range of items, including passwords, tokens, certificates, API keys, and more
- Can be used for key management, supporting the creation, control, and rotation of encryption keys:
  - Keys can be imported/generated in HSMs certified to FIPS 140-2 level 2 standards
- Enables certificate management, including enrollment of certificates from third-party Public CAs

Important Key Vault properties include:

- Name of the Key Vault, which creates a Vault URI: https://name.vault.azure.net
- Access Policies, which can be used to control access on the data-plane of the Key Vault
- Pricing tier including standard or premium (only premium supports HSM-protected keys)

An overview of Key Vault management through the REST API:

- Key Vault can be managed and operated through the Key Vault REST API.
- This includes operations such as management, creation, and deletion of a Key Vault or keys/secrets/certs.
- All requests to Azure Key Vault MUST be authenticated:
  - Key Vault supports Azure AD access tokens using OAuth2.
  - Managed Identities can be used to achieve authentication.
- Refer to this Microsoft Article for more information on the various REST API commands.
- An example REST API call, for creating/updating a Vault is as follows:

HTTP GET

```
https://management.azure.com/subscriptions/{subscriptionId}/resources?$filter=resourceType eq 'Microsoft.KeyVault/vaults'&api-version=2018-02-14
```

Security

Managed Identities

Close

Managed Identities provides a secure method for authenticating Azure resources against other Azure services, without needing to include credentials. Managed Identities is a feature of Azure AD which specifically provides an Azure resource with a managed identity within Azure AD.

This feature provides the ability to authenticate an Azure resource "behind-the-scenes." This does not provide any implicit permissions (authorization) though. That must still be configured separately.

Important facts about Managed Identities:

- Avoids the need for application credentials to be stored in code (e.g. Client ID and secrets)
- Is fully managed by Microsoft, so credentials no longer need to be rotated by developers
- Automates the creation/registration of an application within Azure AD, Service Principal, and Client ID
- Includes built-in functionality for Azure resources to securely obtain an authentication token
- Does not imply any authorization, since the identity must still be granted whatever permissions are desired

Important components of Managed Identities:

- Managed Identities include three main elements:
  - **Principal ID:** The object ID of the service principal object for the managed identity
  - **Client ID:** Unique identifier generated by Azure AD
  - **Azure Instance Metadata Service (IMDS):** REST endpoint accessible to VMs via 169.254.169.254
- Two types of managed identities are available:
  - System-assigned:
    - Fully managed by Azure and directly created for and associated with a resource
    - Exists with the resources, and is deleted if the resource is deleted
    - Associated with a single resource
  - User-assigned:
    - Created as an individual resource, managed by an administrator (e.g. you)
    - Must be managed (created, deleted, etc.) by an administrator
    - Can be associated with one or more Azure services

Security

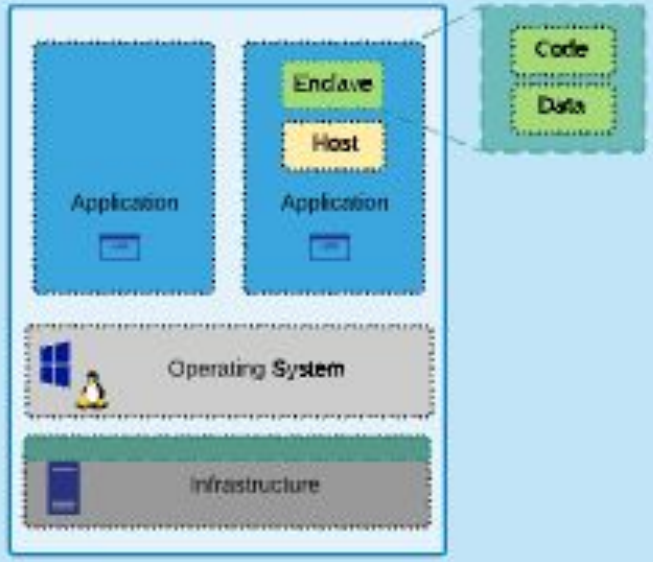
Confidential Compute

Close

Confidential Compute refers to a number of efforts in research, hardware, and software, which are aimed at protecting data whilst it is being processed in the cloud.

This represents a focus on encryption/security of data "in use." It differs from the common models of encryption at rest or encryption in transit, and is primarily comprised of the following:

- Security is achieved through the use of a protected **Trusted Execution Environment (TEE):**
  - TEEs are also referred to as *enclaves*.
  - Enclaves provide a highly secure, isolated area for code and data.
  - Applications can be developed to use these enclaves, using the **Open Enclave SDK**.
- Microsoft Azure supports the use of TEE via:
  - Compute-based, using Intel's Software Guard eXtensions (SGX) which is available in the DC series VMs
  - An expanding number of services which support the Open Enclave SDK



Information about the use of a TEE:

- It is not possible to view data/operations within a TEE from outside of the TEE
- Sometimes this is referred to as trusted/untrusted
- More information on Open Enclave SDK can be found at this Microsoft Github page

Security

Subscription and Services Layer

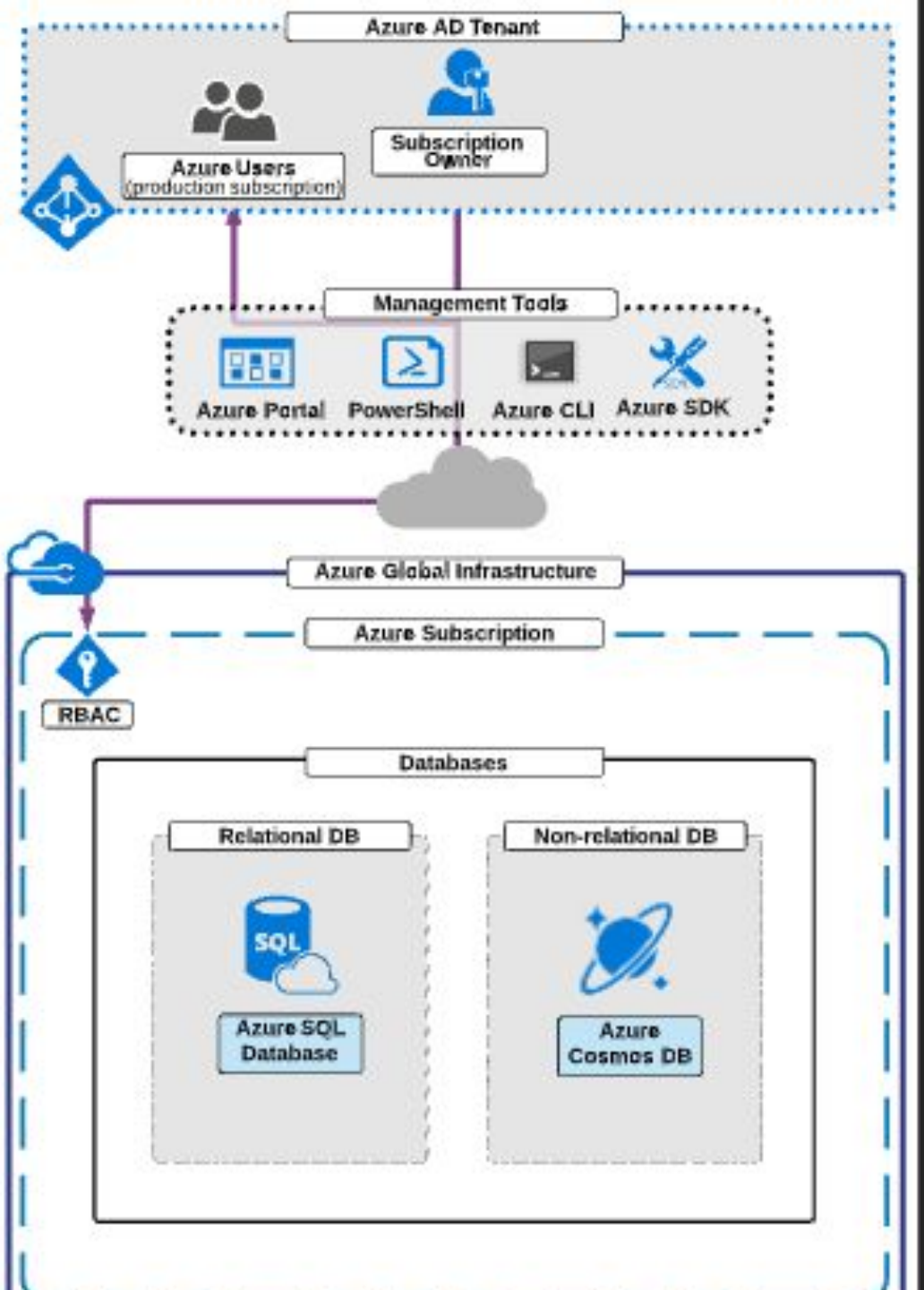
Physical and Networking Layer

Databases

Microsoft provide a range of data storage and processing services.

The two main data database facilities are Azure SQL and Azure Cosmos DB.

Appendix



Databases

Azure SQL Database

Close

Overview | Encryption

Azure SQL Database is a database-as-a-service solution fully managed by Microsoft. It provides features and functionality similar to traditional Microsoft SQL Server, for relational databases.

Important information about Azure SQL Database:

- Created with with the following settings:
  - SQL Logical Server, which configures: `name.database.windows.net` and a location
  - Pricing tier which sets resource and storage limits
- Supports a range of features like firewall, geo-replication, backups, etc.

Azure SQL Database pricing models:

- Database Transaction Units (DTU) - An abstracted representation of the underlying resources
- vCPU - A clearer view of actual underlying resources, with control over storage and CPU

Information about Azure SQL Database deployment options:

- Single database: Managed SQL Database server, recommended for cloud-born applications
- Elastic pools: A resource pool intended to be shared by multiple single databases
- Managed instance: Intended for databases migrated from on-premises with near 100% compatibility with on-premises SQL server

Elastic Pools:

Elastic Pools can improve and simplify the manageability of scale when multiple databases are involved. Pools are suitable when multiple databases have low average utilization and relatively infrequent utilization spikes.

Databases

Azure SQL Databases - Encryption

Close

Overview | Encryption

Microsoft provides two main methods for managing encryption on Azure SQL Database - Transparent Data Encryption (TDE) and Always Encrypted.

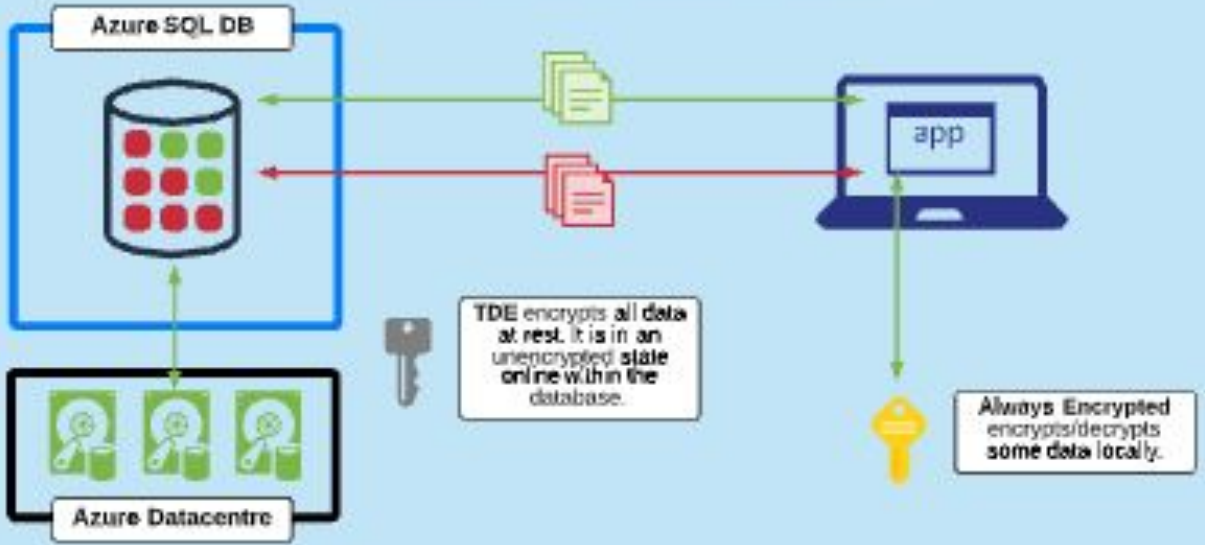
The following is a list of the key features and functions of both methods:

Transparent Data Encryption (TDE):

- The focus of TDE is protection of data at rest - e.g. datacentre of storage theft
- TDE performs decrypt/encrypt on the underlying server, transparent to the application
- Information within the database is accessible/visible to any who have access to the data (unlike below)

Always Encrypted:

- The focus of Always Encrypted is protection of confidential/sensitive information from DB administrators
- The flow of traffic for always encrypted is as follows:
  - Client computers have an Always Encrypted driver installed,
  - Confidential/sensitive data (e.g. credit card information) is encrypted client-side,
  - The encrypted data is transferred to the database in encrypted form,
  - The encrypted data is stored in the database in encrypted form.
- Client applications never reveal the encryption keys to Azure SQL



Databases

Cosmos DB

Close

Overview | Partitioning | Consistency

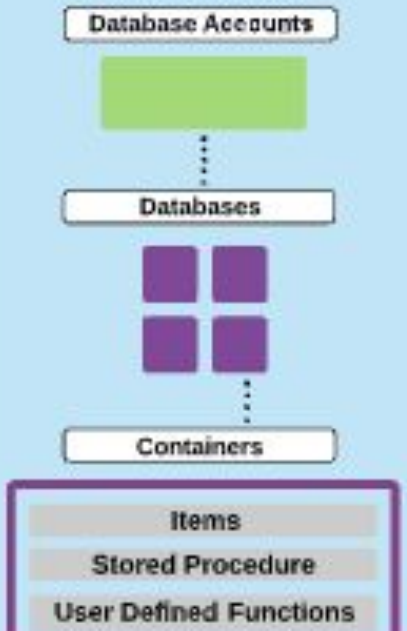
Cosmos DB is an advanced database service designed for global data distribution and accessibility. Microsoft manages the underlying infrastructure to provide scale, availability, performance, and replication for a range of non-relational data types.

Some of the main Cosmos DB features include:

- Multi-master geographic distribution and transparent replication of data across Azure regions
- Five well defined consistency levels for replication spanning strong <> eventual
- Elastic horizontal scalability and single-digit-millisecond accessibility
- Support for multiple data types and APIs, including SQL, MongoDB, Cassandra, and Gremlin
- Server-side programming including stored procedures, triggers, and user defined functions

Important information about Request Units which are used for managing throughput:

- 1 RU is calculated as the throughput required to read a 1 KB document
- Abstracts the capacity planning difficulties of understanding required CPU/memory/IOPS
- Can be provisioned at either the Database or Container layer



Cosmos DB is made up of the following components:

- Cosmos DB Account includes the following properties:
  - Name: establishes the URI: name.documents.azure.com,
  - API: for data type / API,
  - Location: initial location (can be expanded with replication).
- Database - namespace / management layer for containers
- Containers - unit of scalability:
  - Type of container depends on data type / API,
  - Can be Collections (document oriented APIs), Graphs (Gremlin API), Tables (table API), etc,
- Items - actual entities being stored
  - Based on type (data type / API)
  - Can be Documents (document APIs), Tables (Cassandra), Graph (Gremlin), etc.

Databases

Cosmos DB - Partitioning

Close

Overview | Partitioning | Consistency

Partitioning is used at the Container layer so that the Cosmos DB service can distribute data amongst infrastructure so as to maintain performance levels.

You must specify a partition key for the Cosmos DB service to use to evenly distribute data and access.

Important information about partitioning:

- Partitioning ***divides items in a container in to logical partitions***, based on the partition key
- Items within a container will be sent to logical partitions based on the hash of partition key of the item
- Partitioning is important and can impact:
  - Pricing - queries that access data in a single partition are cheaper than across multiple partitions
  - Transactions in stored procedures or triggers can only be performed against a single partition
- Partition management, scalability, and distribution

Considerations for selecting a partition key:

- Each partition has a limit of 10 GB of storage; partitioning should consider your ***storage*** requirements
- Throughput is limited at the partition layer; partitioning needs to consider read/write ***request***-distribution (e.g. partitioning based on timestamp would result in a hotspot for requests)
- Transactions can be scoped to a partition key; consider your top queries & common scope/filters

Databases

Cosmos DB - Consistency

Close

Overview | Partitioning | Consistency

A major consideration of any distributed database is balancing the levels of availability, performance, and consistency. Most distributed databases have two options: strong constency, and eventual consistency.

With Cosmos DB, there is much greater control over this balance, as depicted below.

Image taken from this Microsoft article on Cosmos DB Consistency Levels.

Important information on Cosmos DB consistency levels:

- The five consistency models from strongest to weakest are: strong, bounded staleness, session, consistent prefix, and eventual
- Consistency levels are region agnostic and are guaranteed for all read operations
- Default consistency levels are configured at the Cosmos DB Account level
- Consistency levels can be overriden through the SDK for a whole client or per-request

Information for each consistency level:

- **Strong:** reads are guaranteed to return the most recent committed version of an item.
- **Bounded staleness:** will be consistent to an agreed amount, this can be configured in two ways:
  - Reads might lag behind writes by at most ___ version of an item, or
  - Reads might lag behind writes by at most ___ amount of time.
- **Session:** reads are guarnteed to honor writes for a given client session.
- **Consistent prefix:** guarantees that reads never see out-of-order writes.
- **Eventual:** no guarantee about orders of reads; replicas will *eventually* converge.

Databases

Azure solutions architect expert:

Azure 303 Exam
+
Azure 304 Exam

Certification details

Take two exams

CERTIFICATION EXAM
Microsoft Azure Architect Technologies

AND

CERTIFICATION EXAM
Microsoft Azure Architect Design

Earn the certification

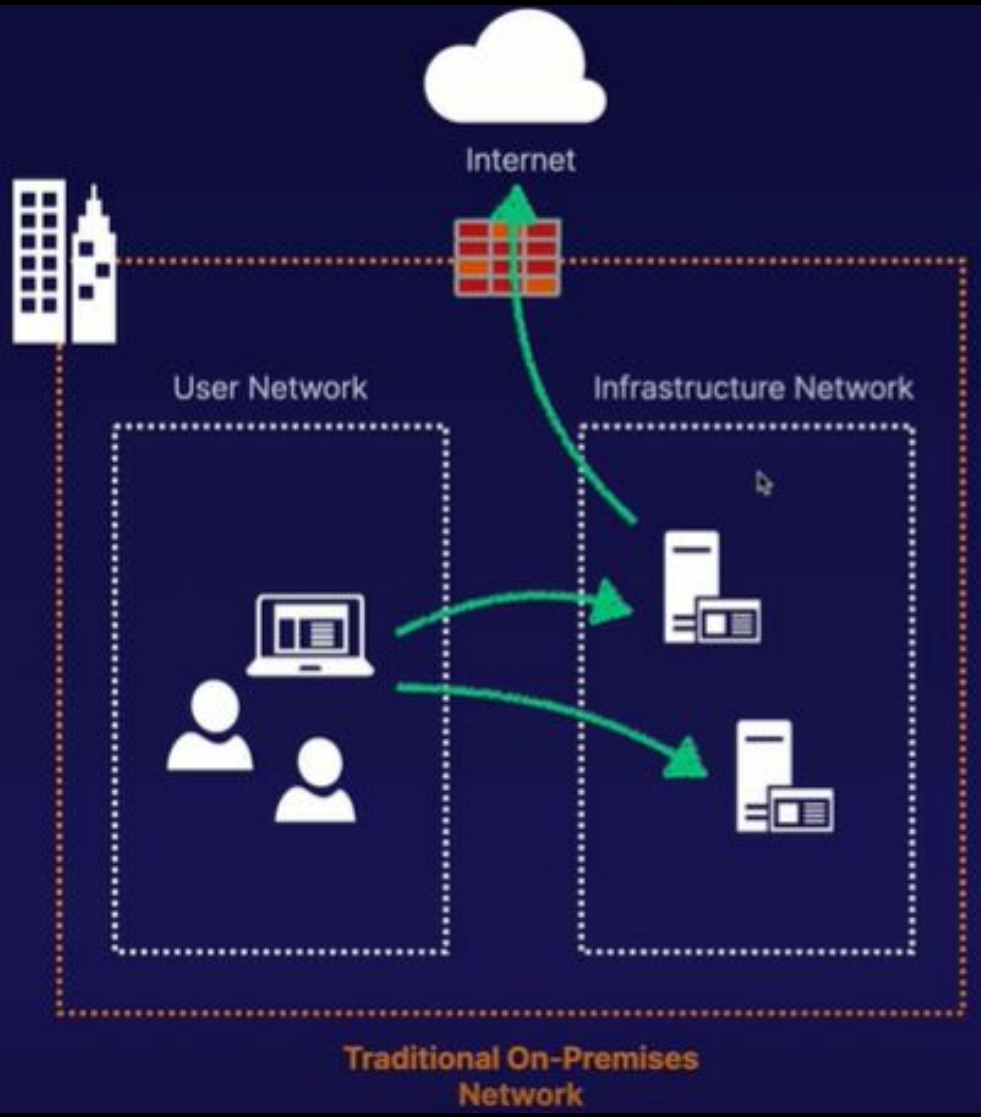
EXPERT CERTIFICATION
Microsoft Certified: Azure Solutions Architect Expert

Skills measured

- Implement and monitor an Azure infrastructure
- Implement management and security solutions
- Implement solutions for apps
- Implement and manage data platforms
- Design monitoring
- Design identity and security
- Design data storage
- Design business continuity
- Design infrastructure

Virtual Networking

The purpose of a Network



Communication:
Connectivity between resources.

Networks are crucial; they help users access resources that exist somewhere other than their personal devices.

For example:

- Users in an office accessing files on a server
- Remote staff accessing a web server
- Application servers accessing data on the Internet

The purpose of a Network

Virtual Network (VNet)

The VNet is an isolated container within Azure, which provides network connectivity for resources.

Subnet

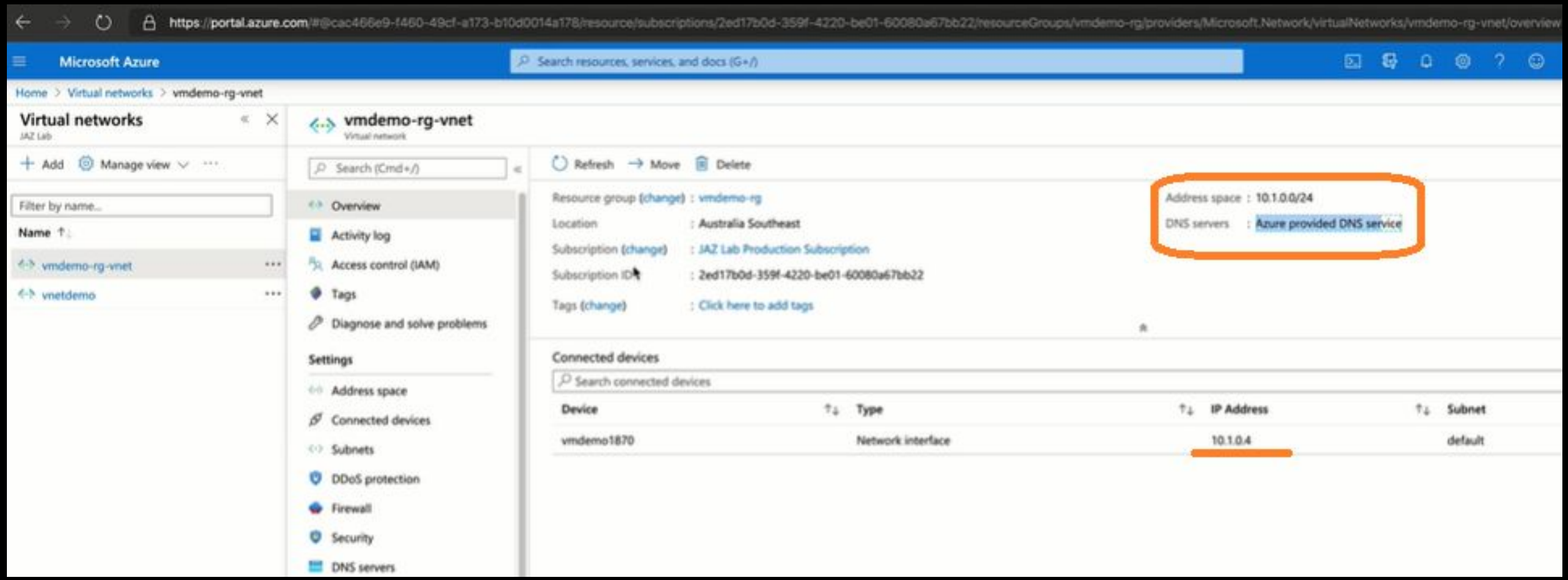
A virtual network has one or more subnets. Resources that require network connectivity must reside in a subnet.

10.0.0.0

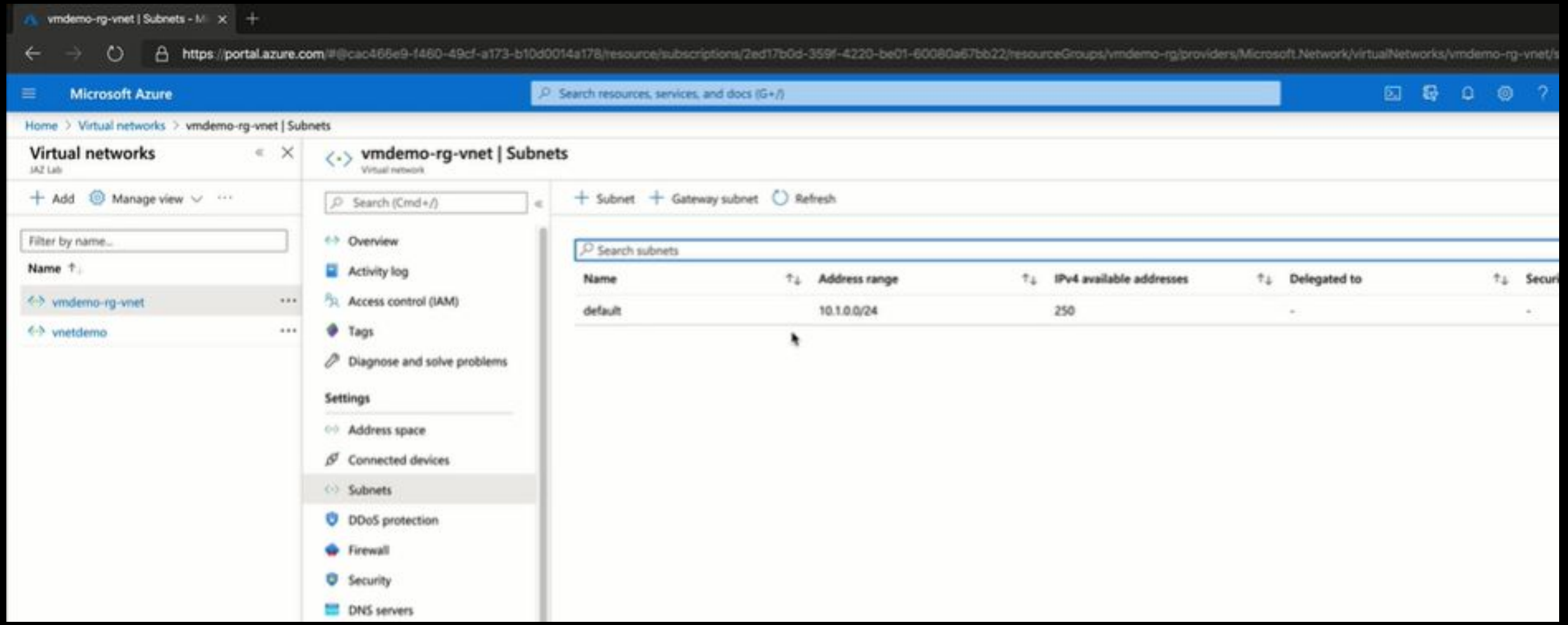
Address Space

Virtual networks require one or more address spaces to provide private IP addresses to resources.

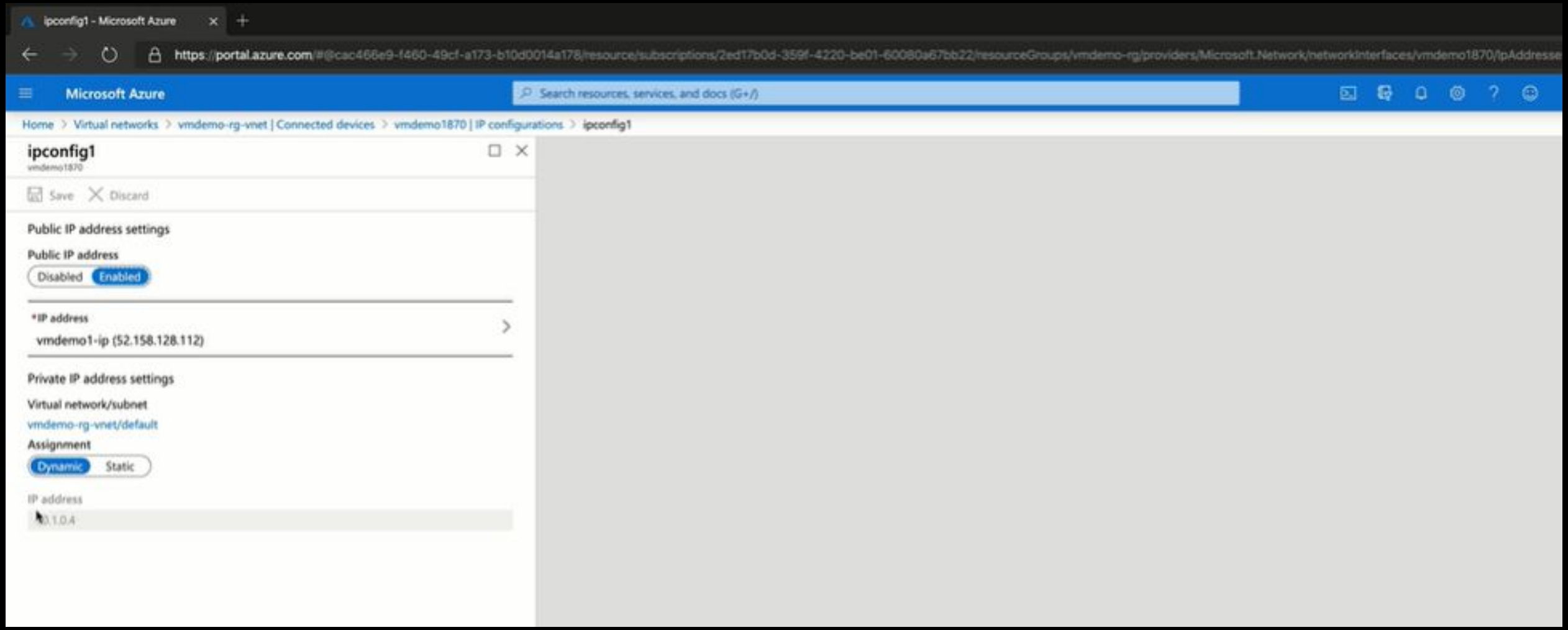
Virtual Networks in Azure



Virtual Networks in Azure



Virtual Networks in Azure | Subnets



Virtual Networks in Azure | ipconfig

Important Considerations

1 **Default Connectivity**

After creating a virtual network, some connectivity is enabled by default (e.g., Internet access, inter-subnet access).

2 **Address Range Restrictions**

Private IP address ranges (as per RFC 1918) are allowed. The smallest allowed VNet/subnet is /29, and the largest is /8.

3 **Reserved IP Addresses**

5 IP addresses are reserved in each subnet. This includes the first three and last one (e.g. x.x.x.0-3, x.x.x.255).

4 **DNS and DHCP**

Custom DNS can be configured for your VNet. DHCP is built-in, however, and custom DHCP cannot be deployed.

5 **Supported Protocols**

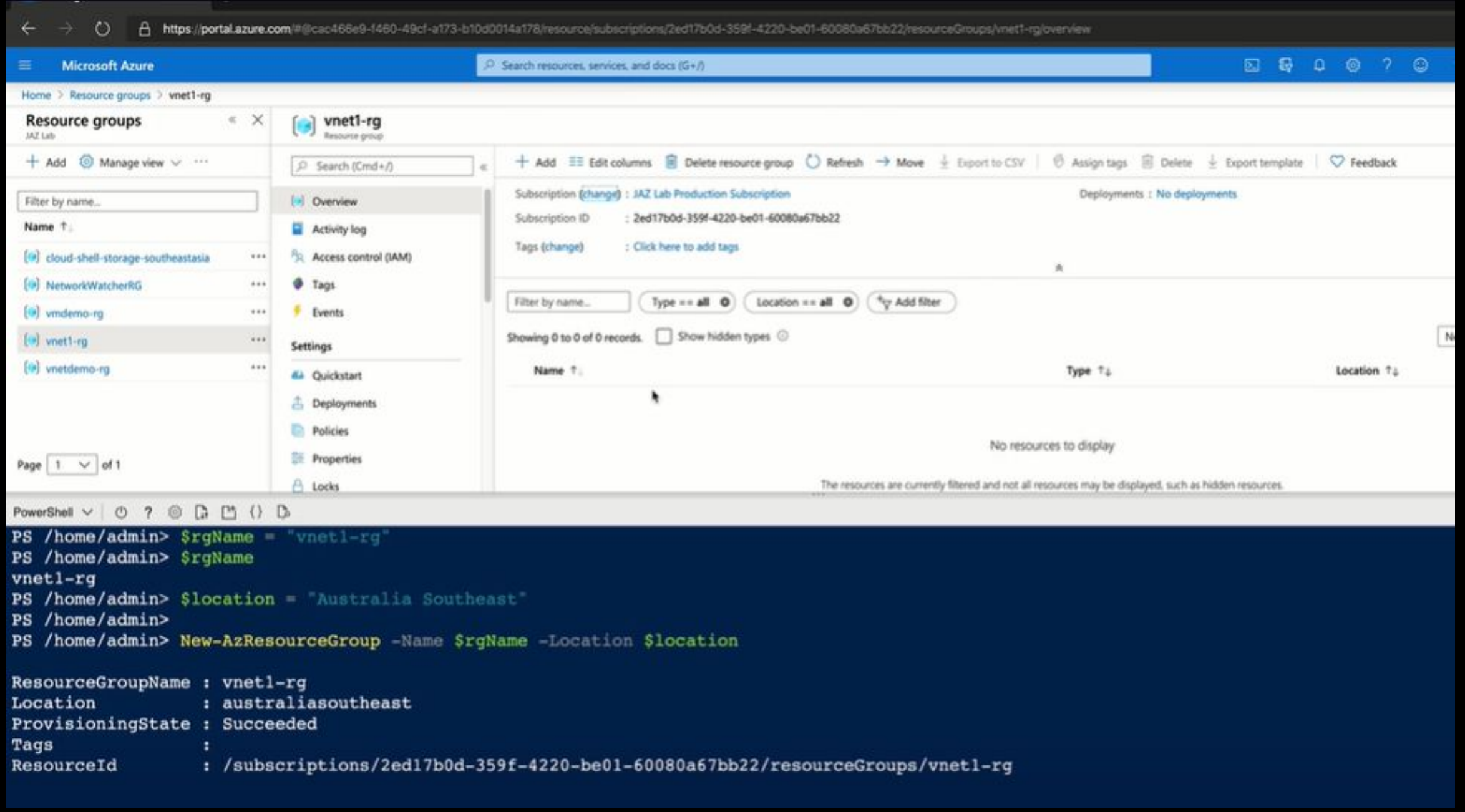
VNets support TCP, UDP, and ICMP TCP/IP protocols. Some common protocols such as multicast and GRE are blocked.

6 **Integrated Connectivity**

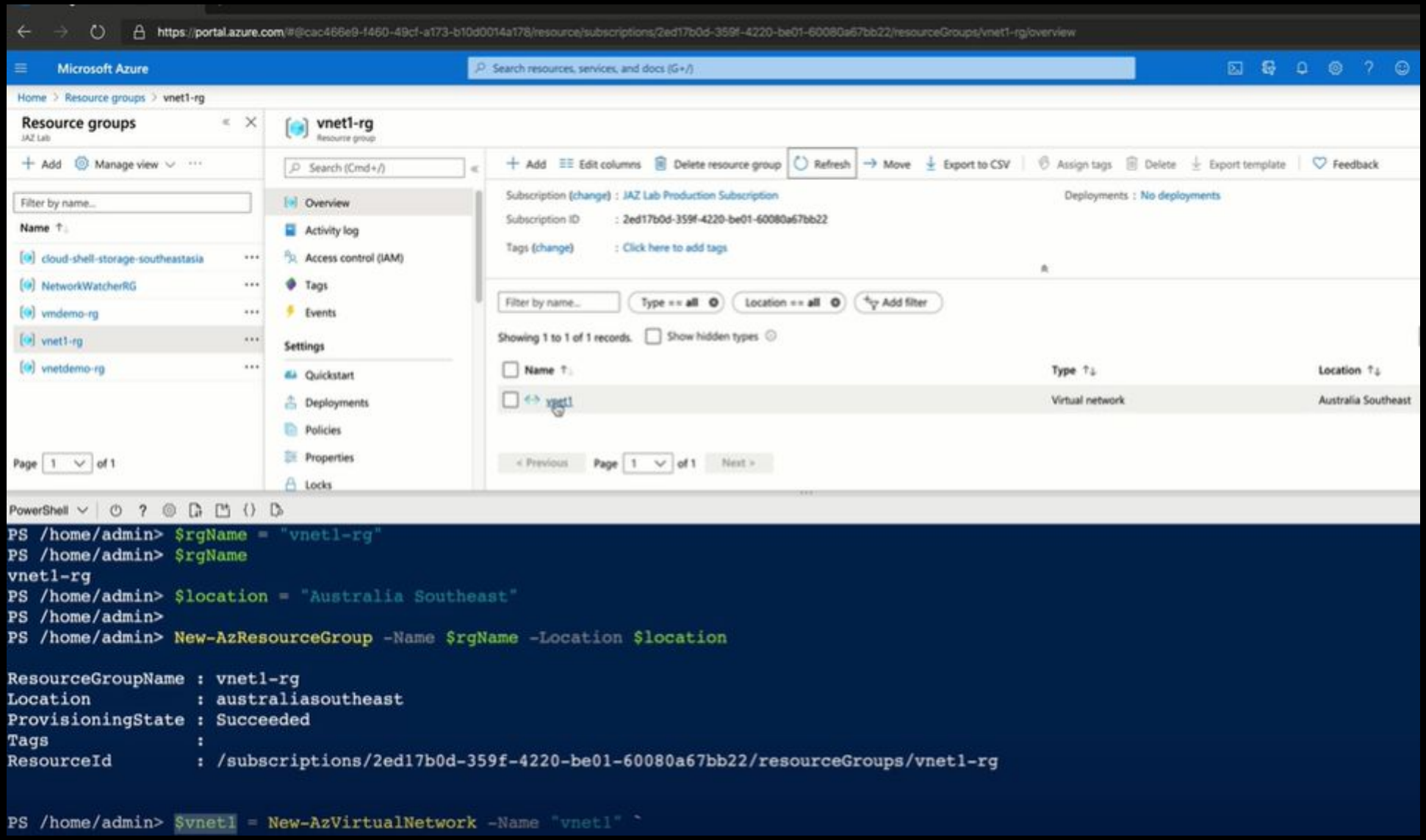
VNets are also built for various forms of integration, including ExpressRoute, Private Link, VPN, and more.

```powershell
$rgName = "vnet1-rg"
$location = "Australia Southeast"

# Create a resource group
New-AzResourceGroup -Name $rgName -Location $location

# Create the virtual network
$vnet1 = New-AzVirtualNetwork -Name "vnet1" `
    -ResourceGroupName $rgName `
    -Location $location `
    -AddressPrefix "10.1.0.0/16"

# Create a subnet, and add it to the new virtual network
Add-AzVirtualNetworkSubnetConfig -Name "subnet1" `
    -AddressPrefix "10.1.1.0/24" `
    -VirtualNetwork $vnet1

Set-AzVirtualNetwork -VirtualNetwork $vnet1
```

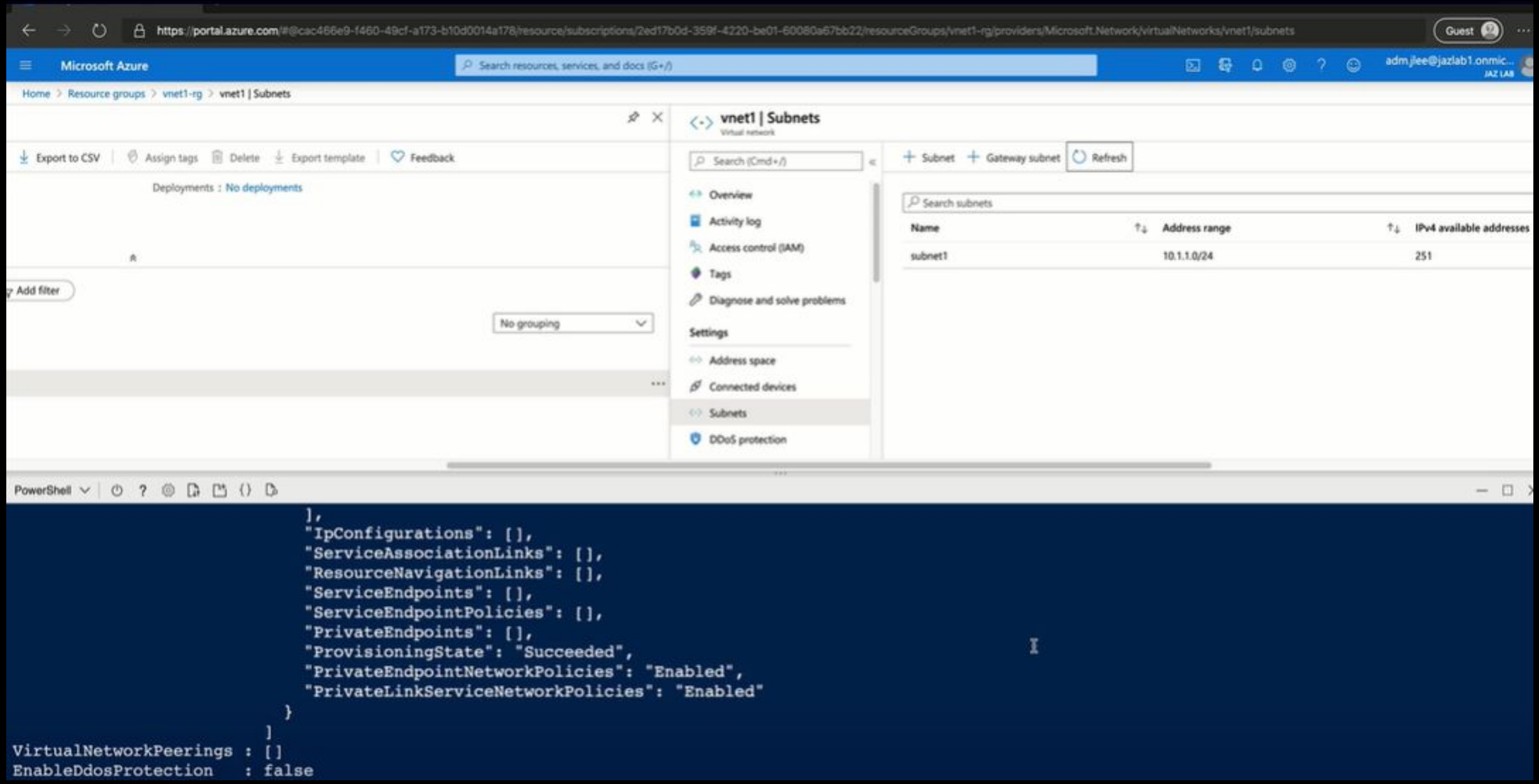
Configuring Virtual Networks



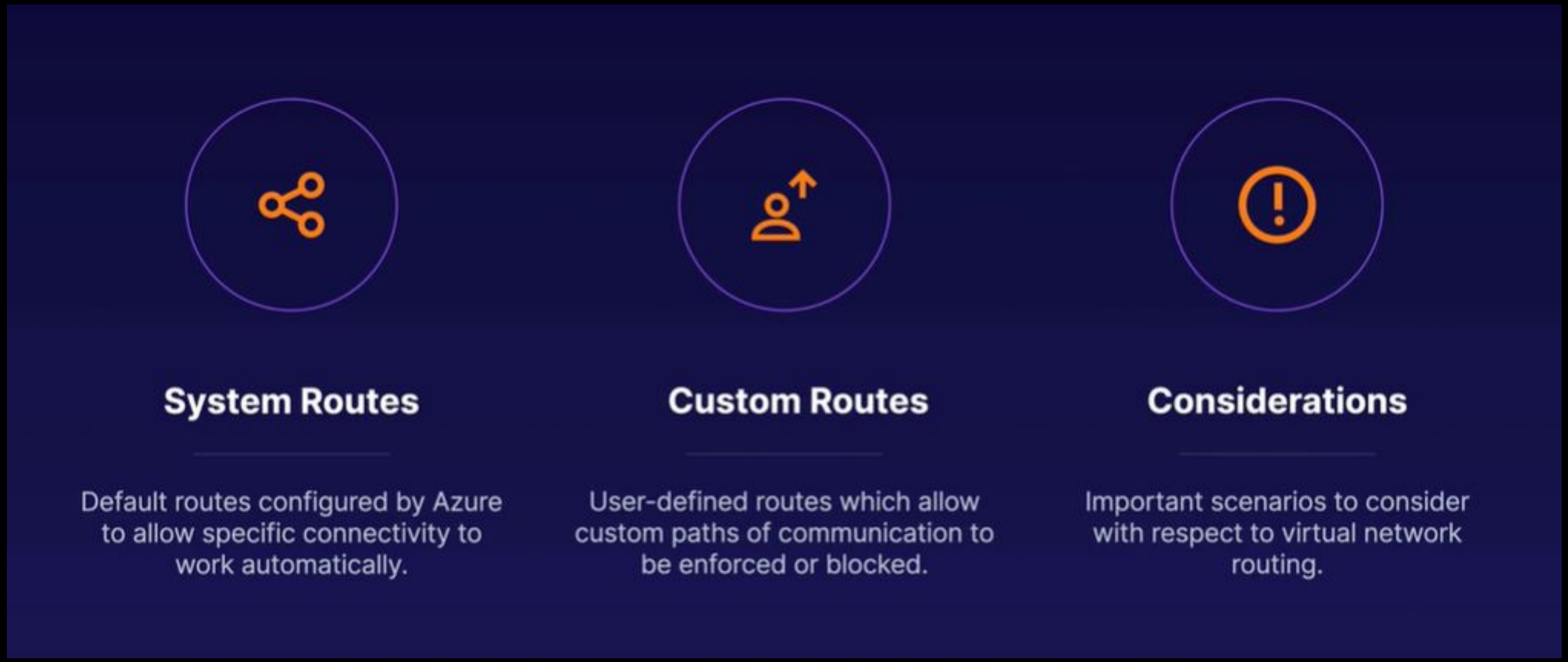
Configuring Virtual Networks



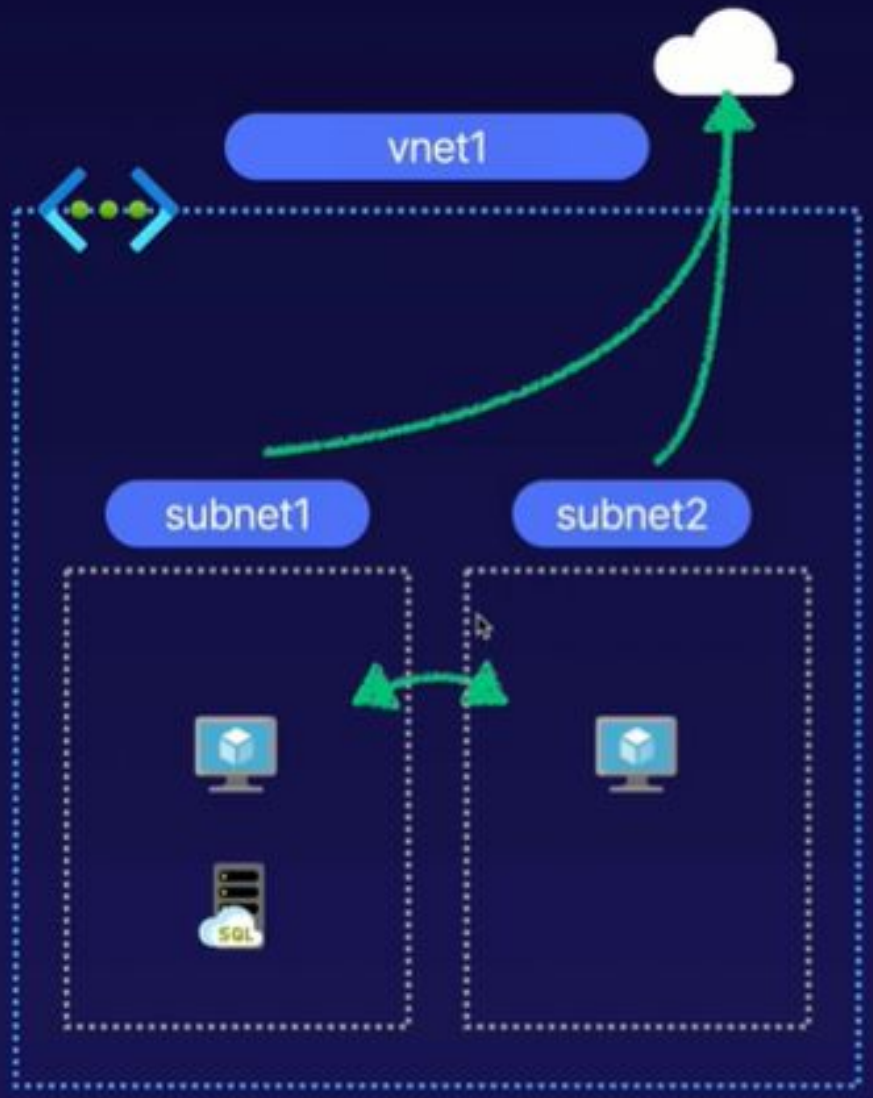
Configuring Virtual Networks



Configuring Virtual Networks



Virtual Networks Routing And Connectivity



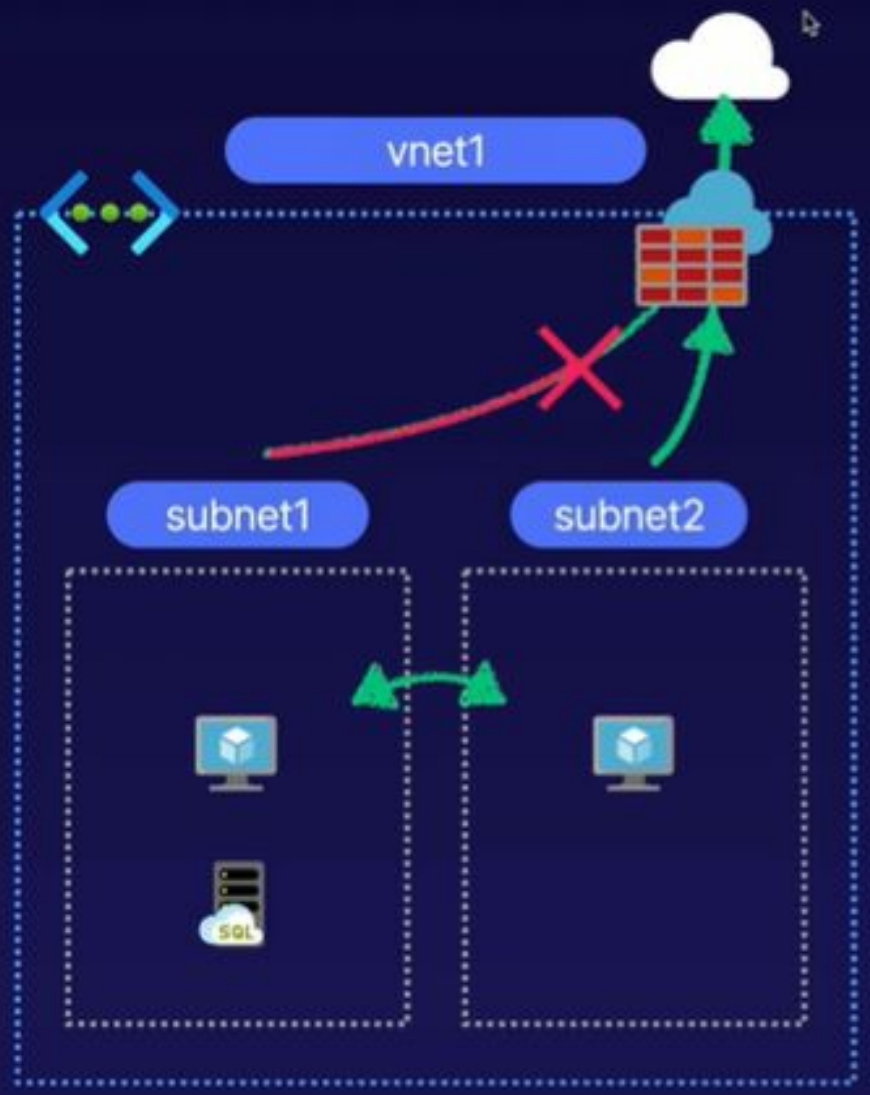
Default Connectivity

Virtual Networks come pre-configured with a range of "paths" already allowed.

The following routes are allowed:

- Outbound access to the Internet
- Subnet to subnet connectivity
- Other default routes to support advanced connectivity (e.g., peering)

Virtual Networks Routing And Connectivity



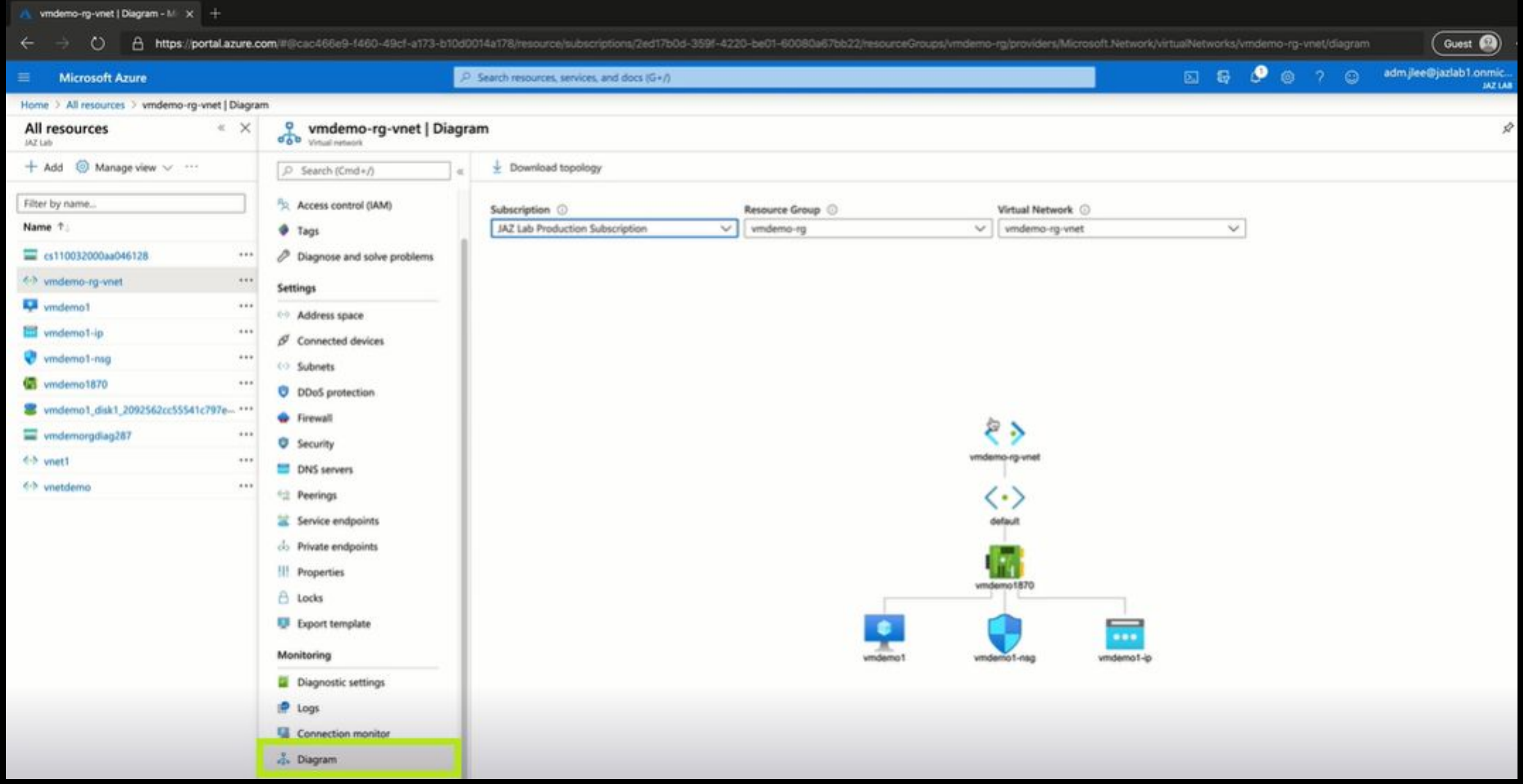
Custom Routes

Custom Routes allow us to change the default routing behavior.

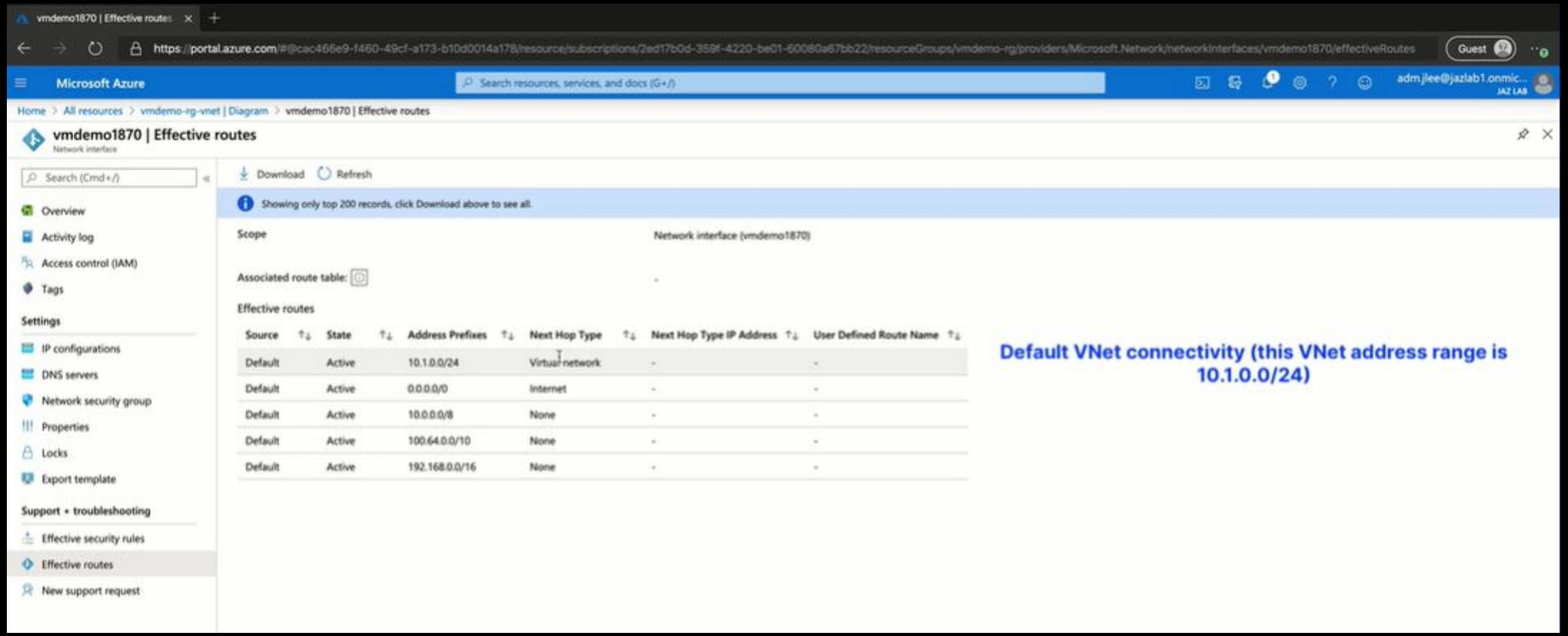
For example:

- Blocking all access to the Internet
- Forcing traffic via another address (e.g., all traffic via a network virtual appliance, or Azure Firewall)

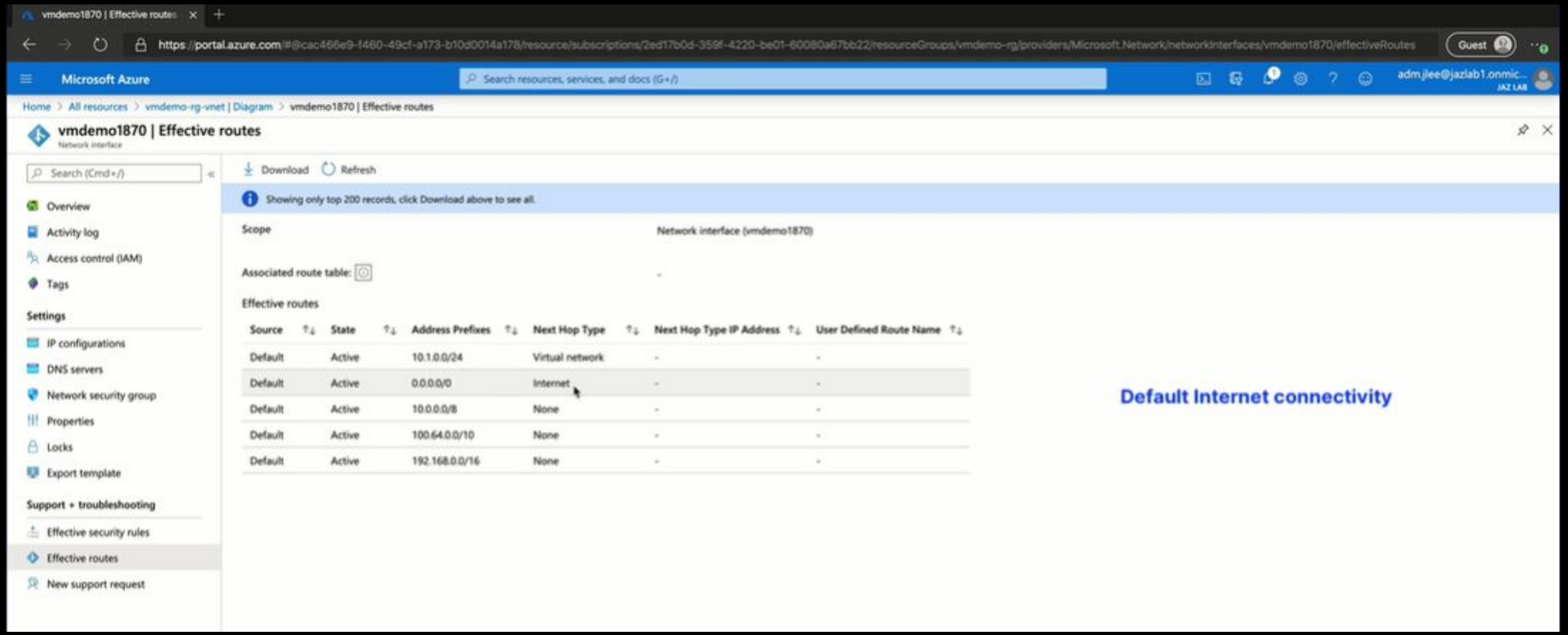
Virtual Networks Routing And Connectivity



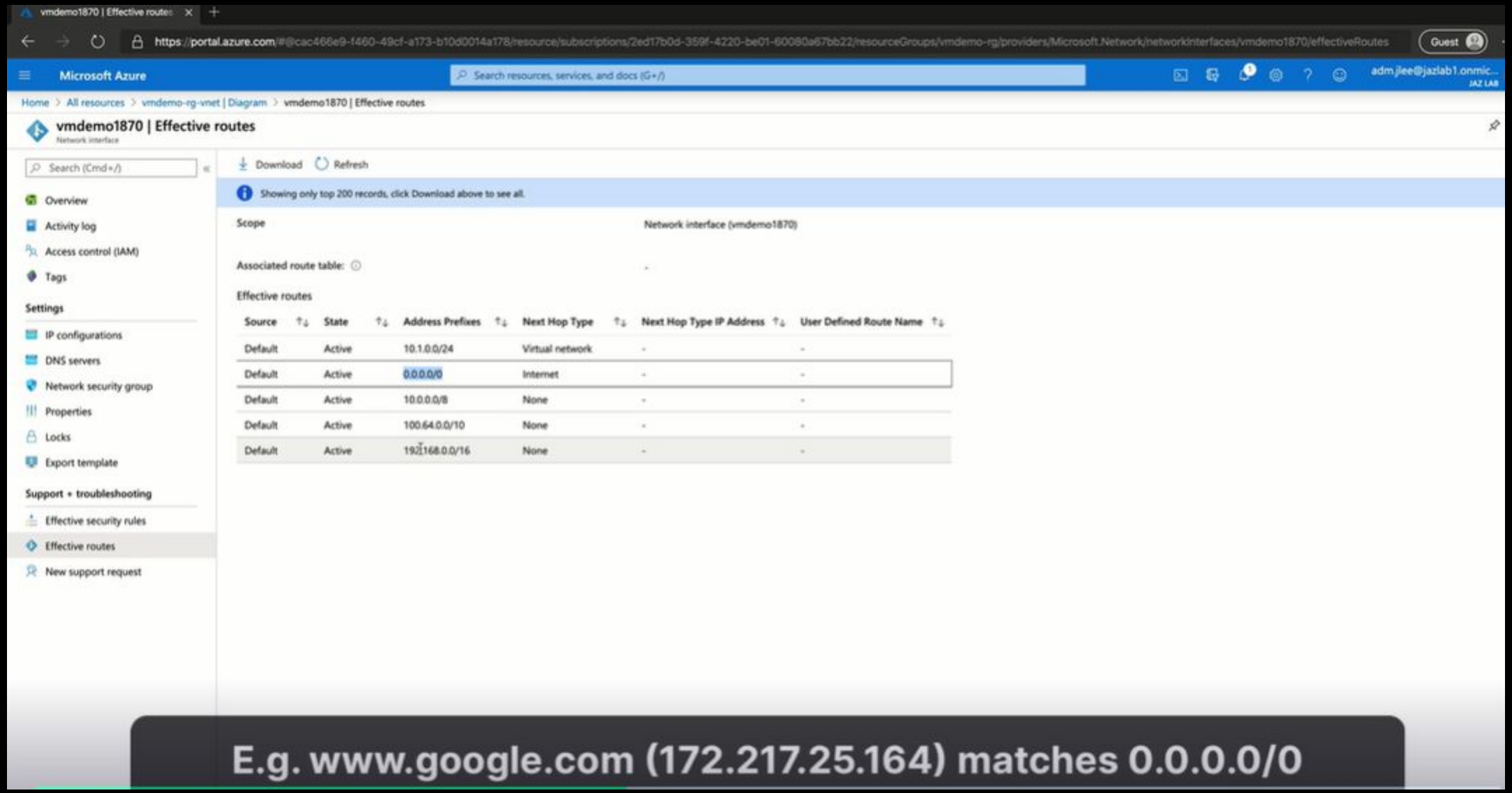
Virtual Networks Routing And Connectivity



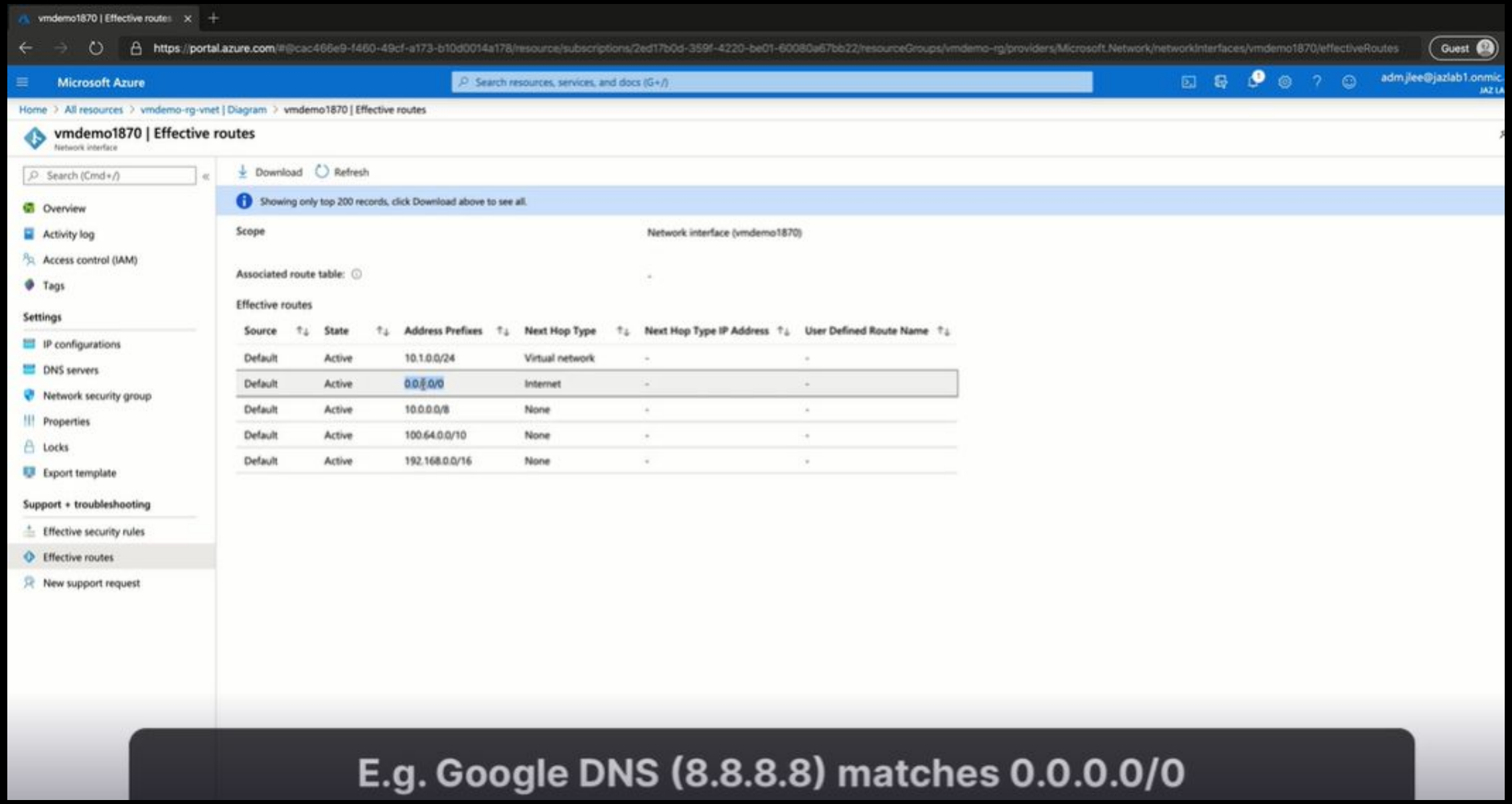
Virtual Networks Routing And Connectivity



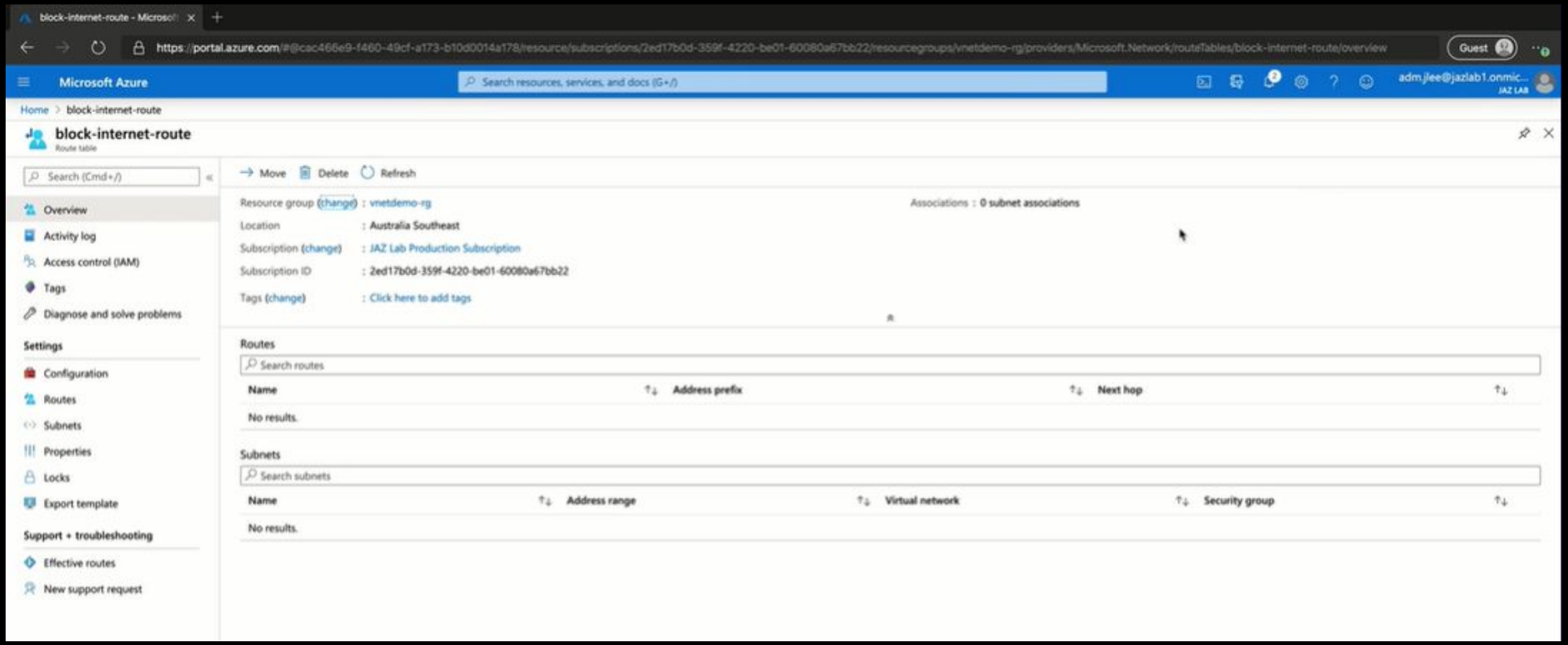
Virtual Networks Routing And Connectivity



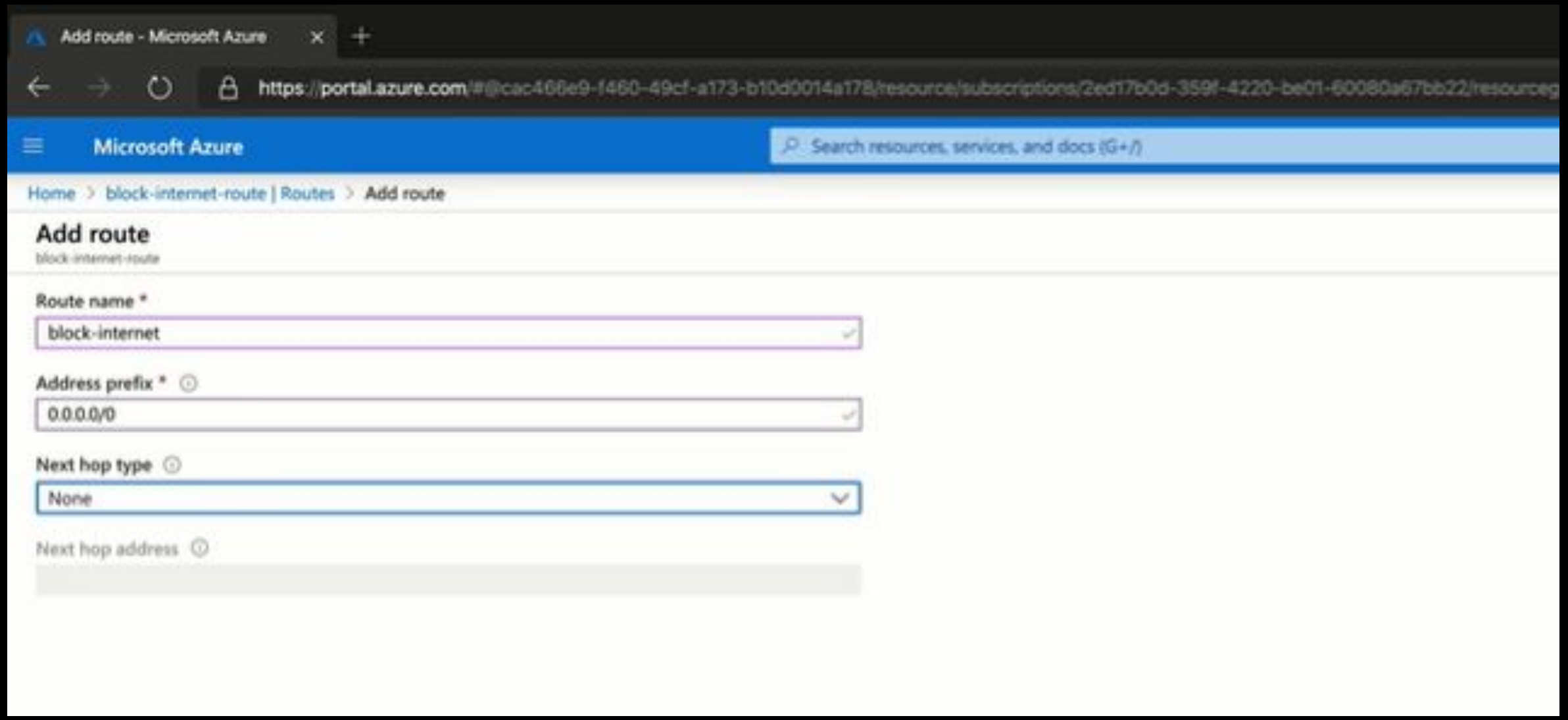
Effective routes



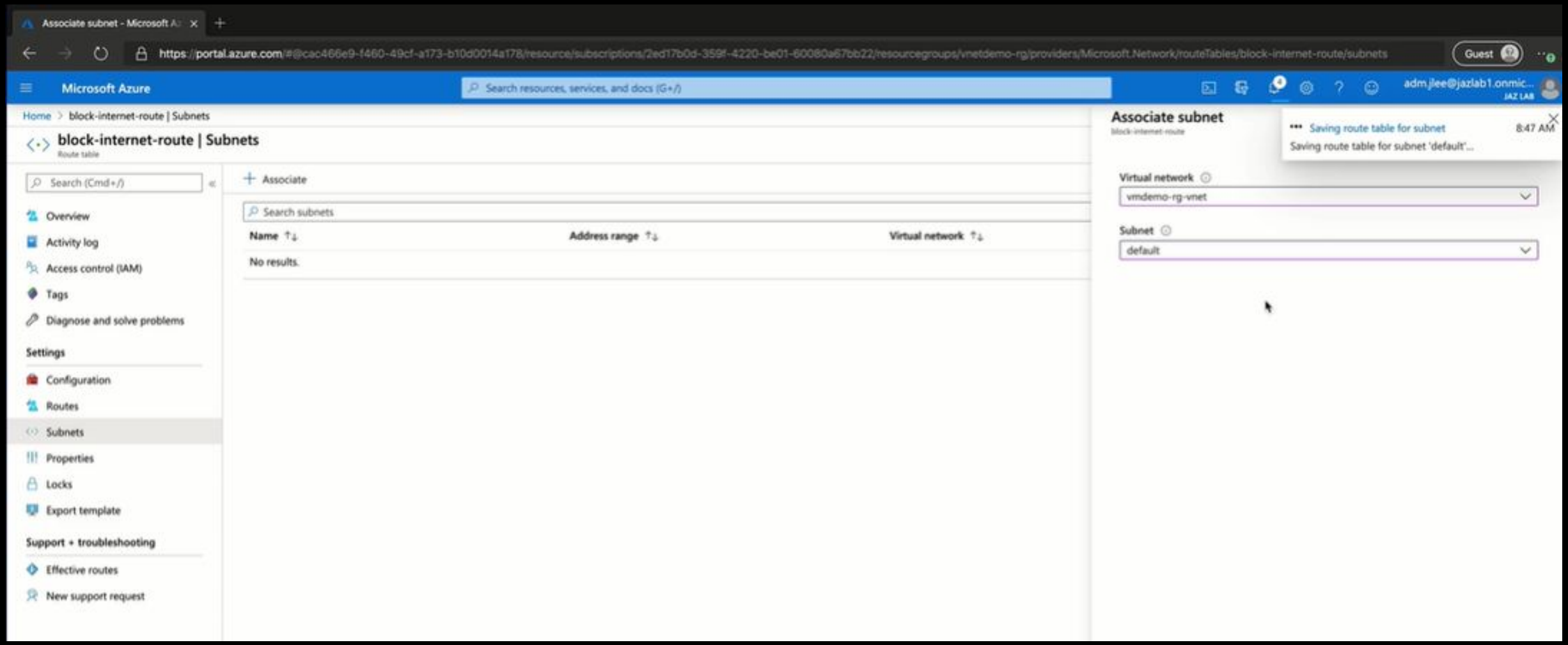
Effective routes



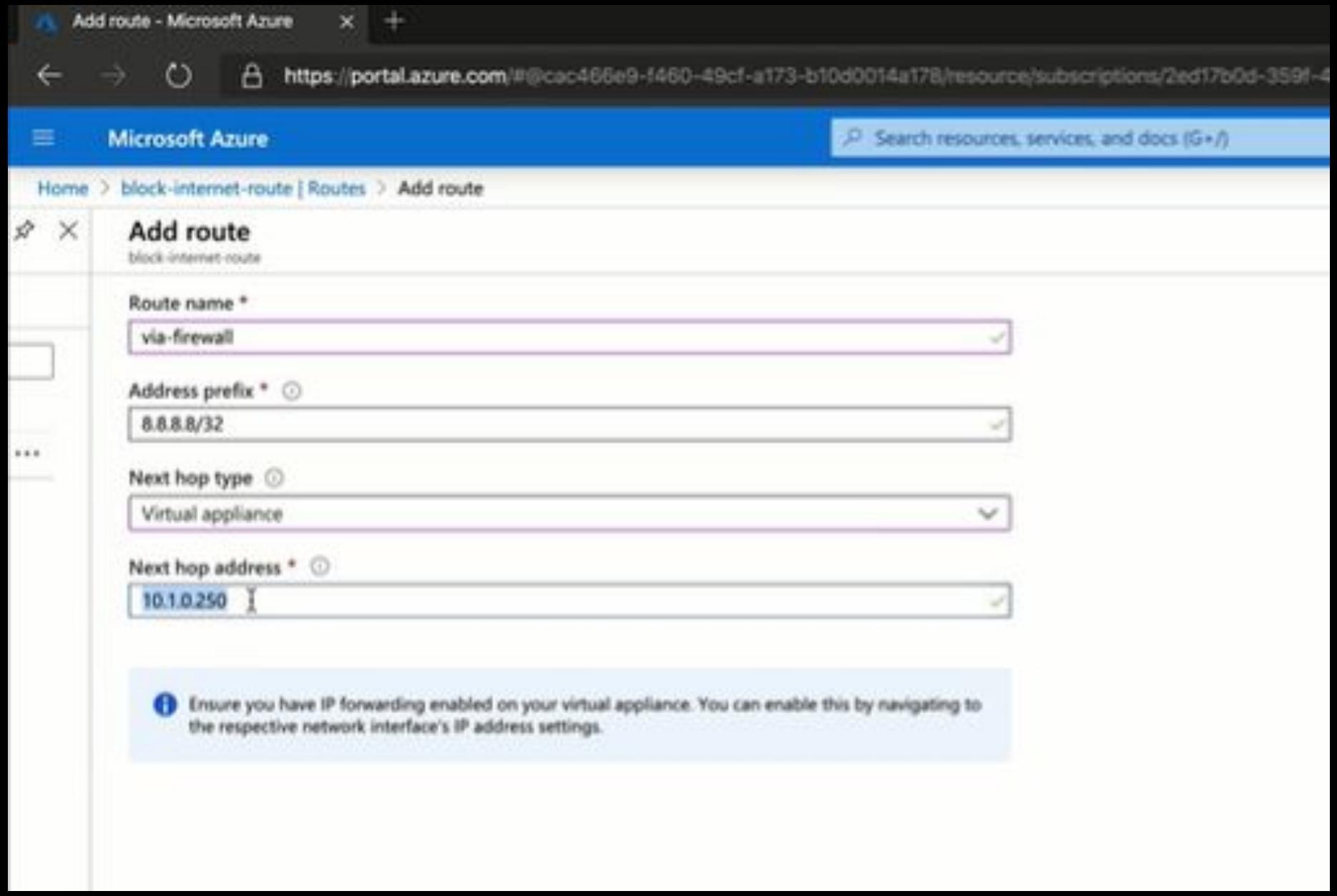
Create a custom route table



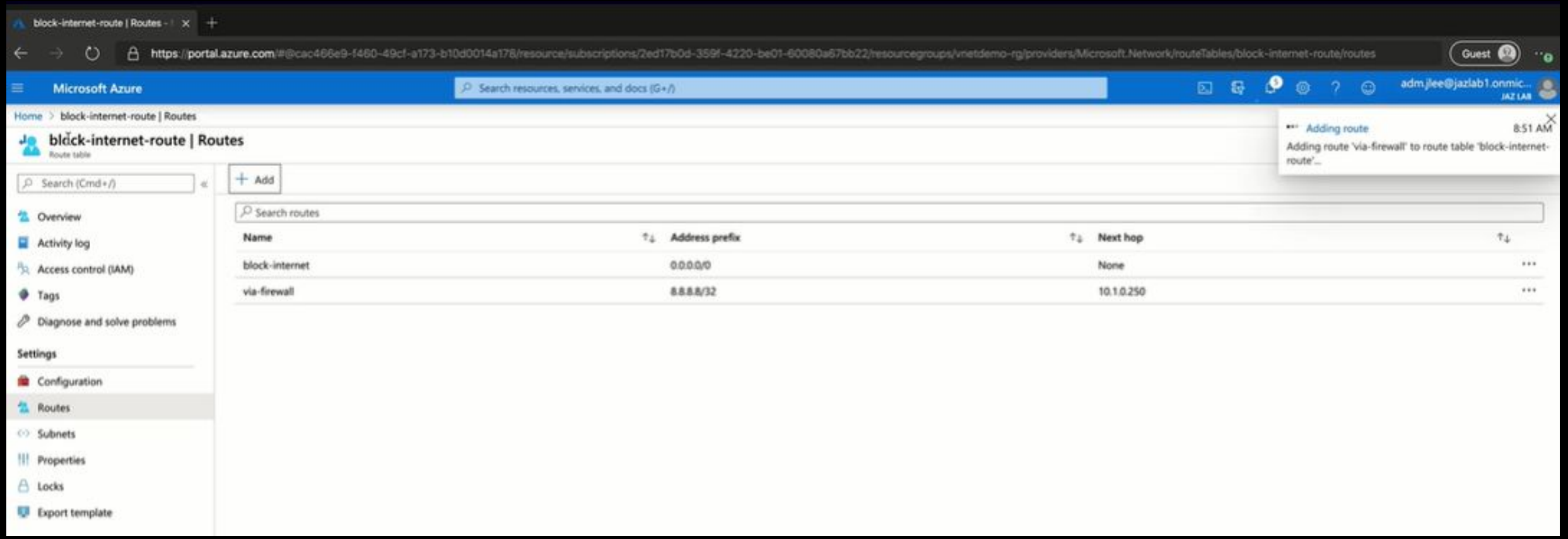
Create a new route



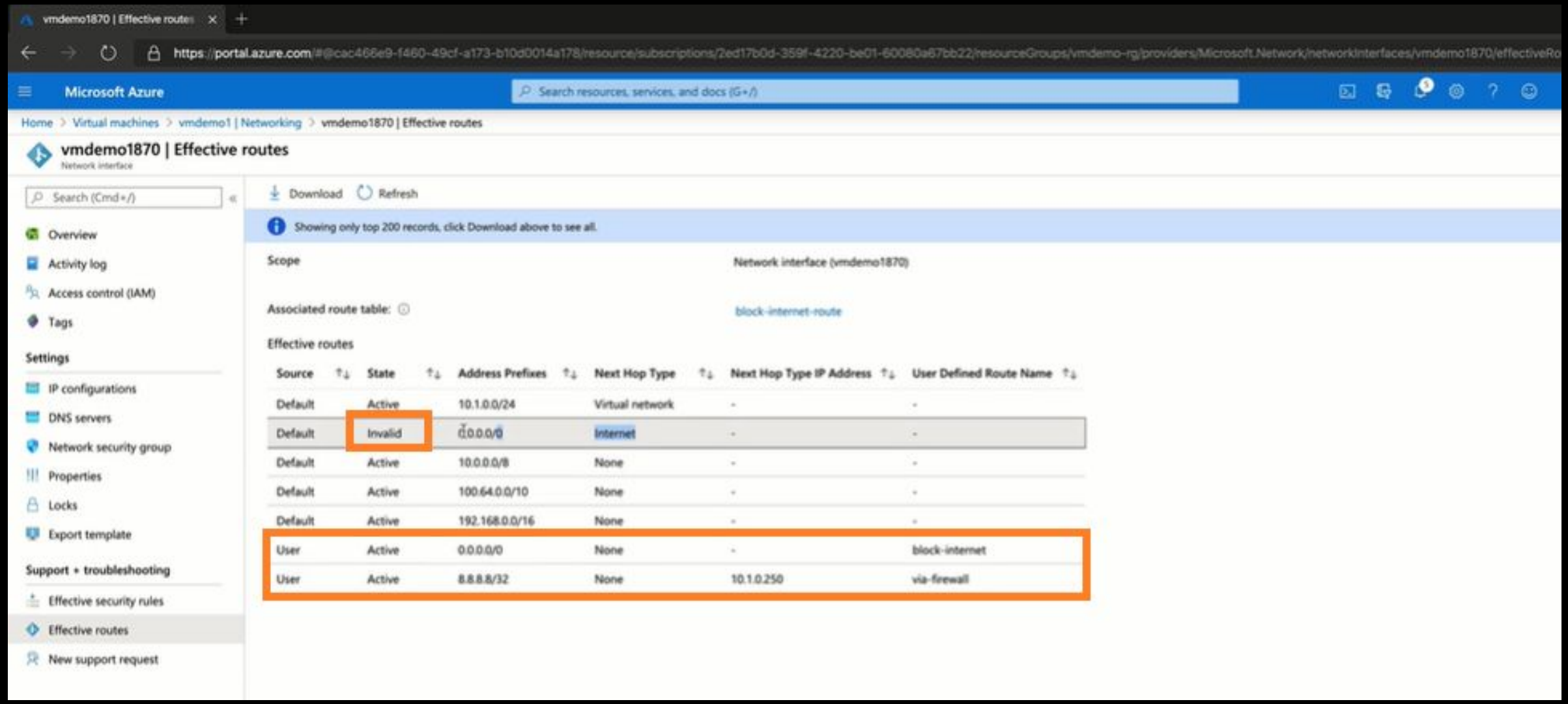
Associate subnet to the new route



Adding a new route to a virtual applicance

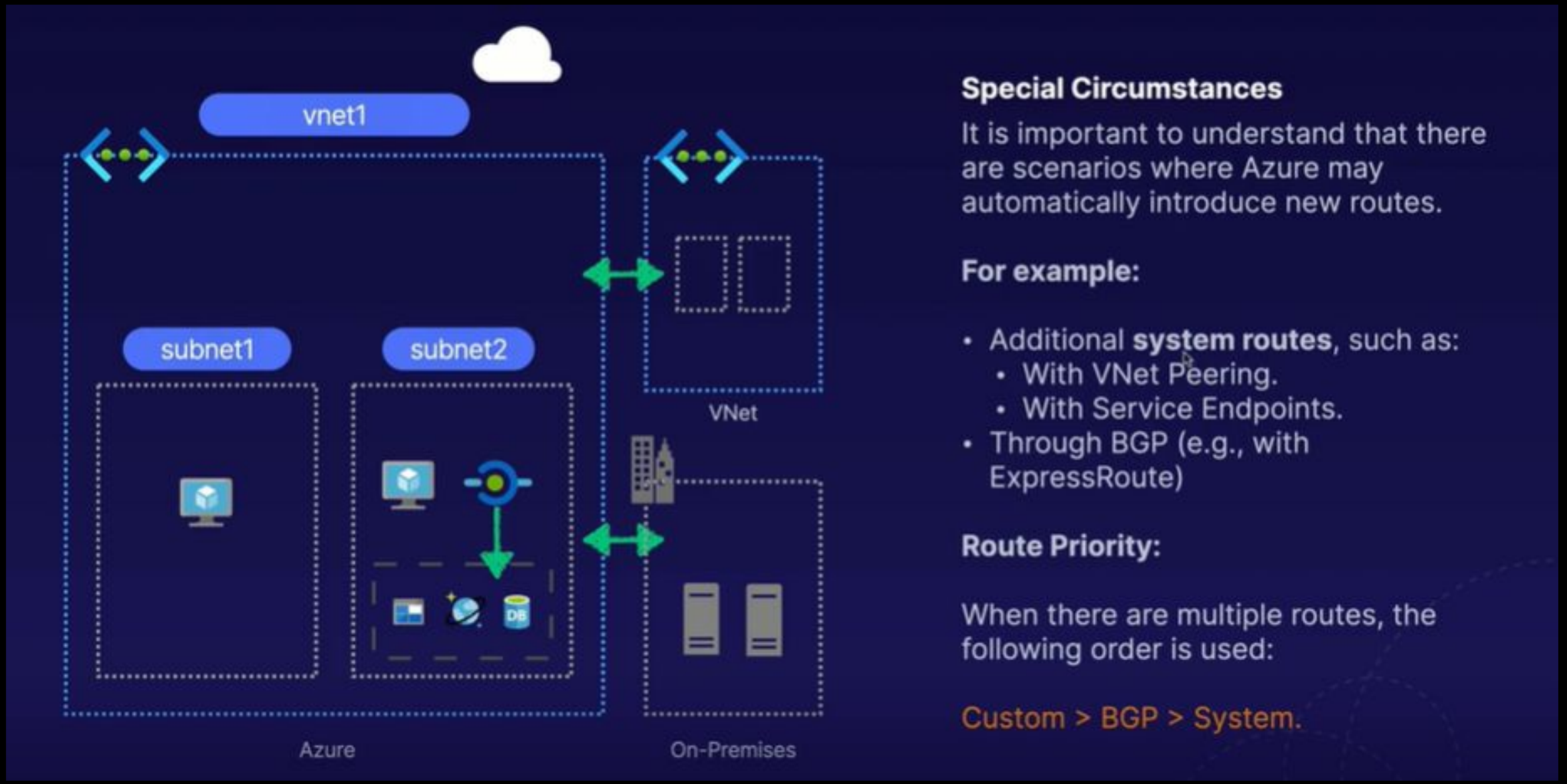


Adding a new route to a virtual appliance



Updated effective routes in my VM

E.g: a 10.0.0.0/32 will match more closely than 10.0.0.0/24



Important considerations

Virtual Machines

Purpose of Virtual Machines

1. **Windows and Linux Compute**

   Deploy virtualized Windows or Linux virtual machines.

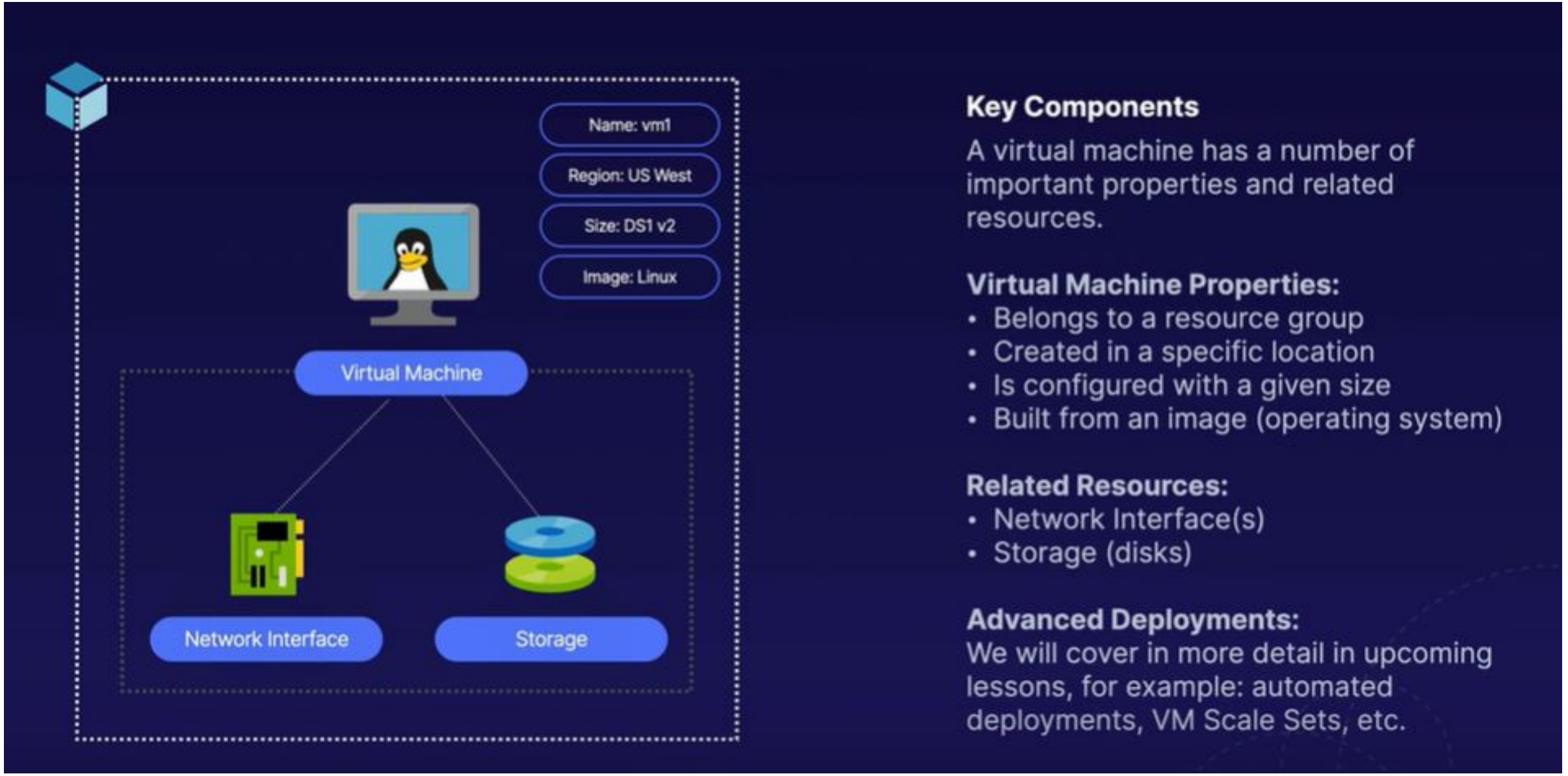
2. **Migrate Existing Workloads**

   Migrate existing compute workloads to Azure, where other services are not yet viable.

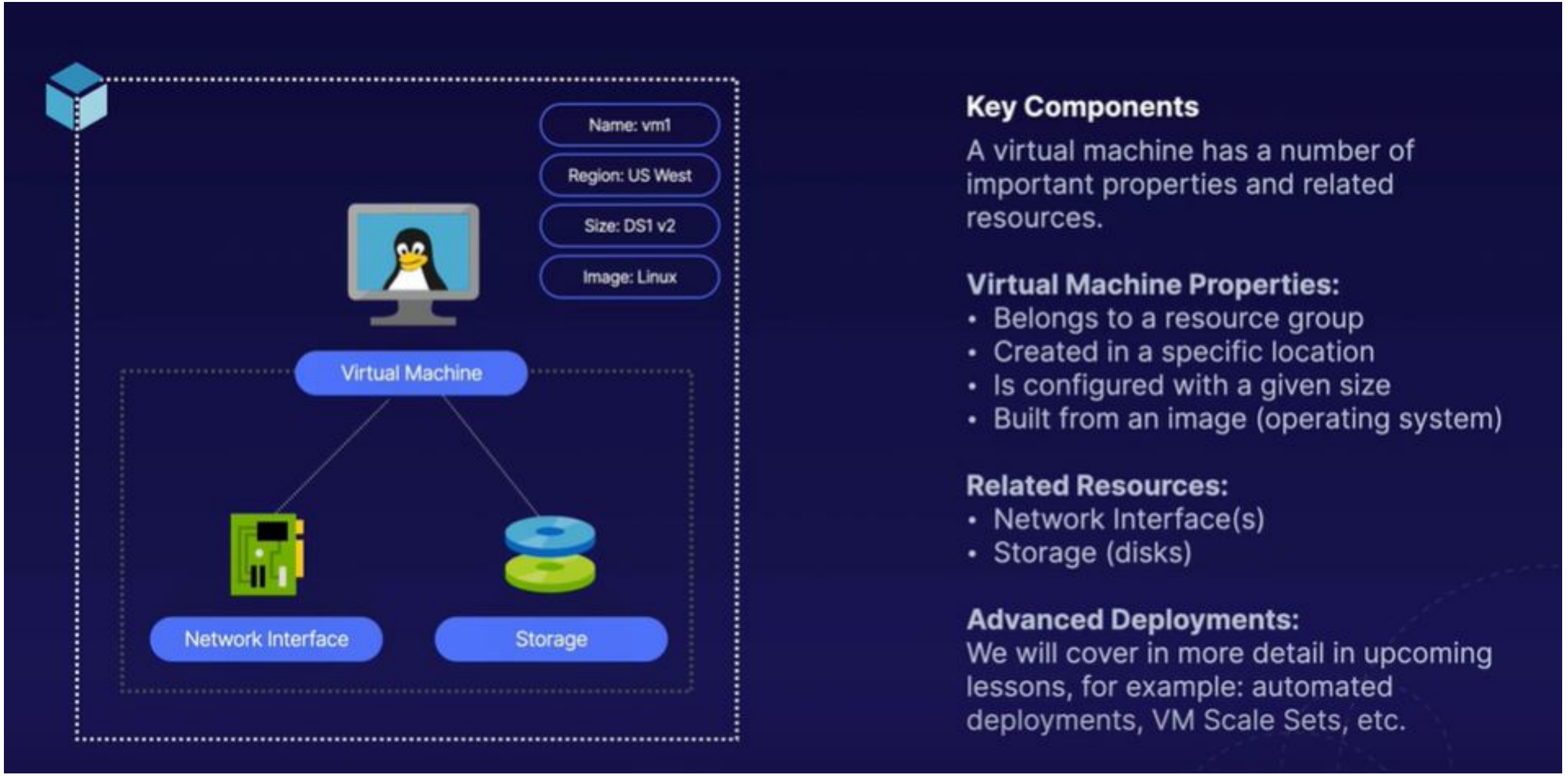
3. **Deploy Advanced Compute Solutions**

   Virtual Machines provide a foundation for advanced solutions such as high performance compute, decoupled, and scalable applications.

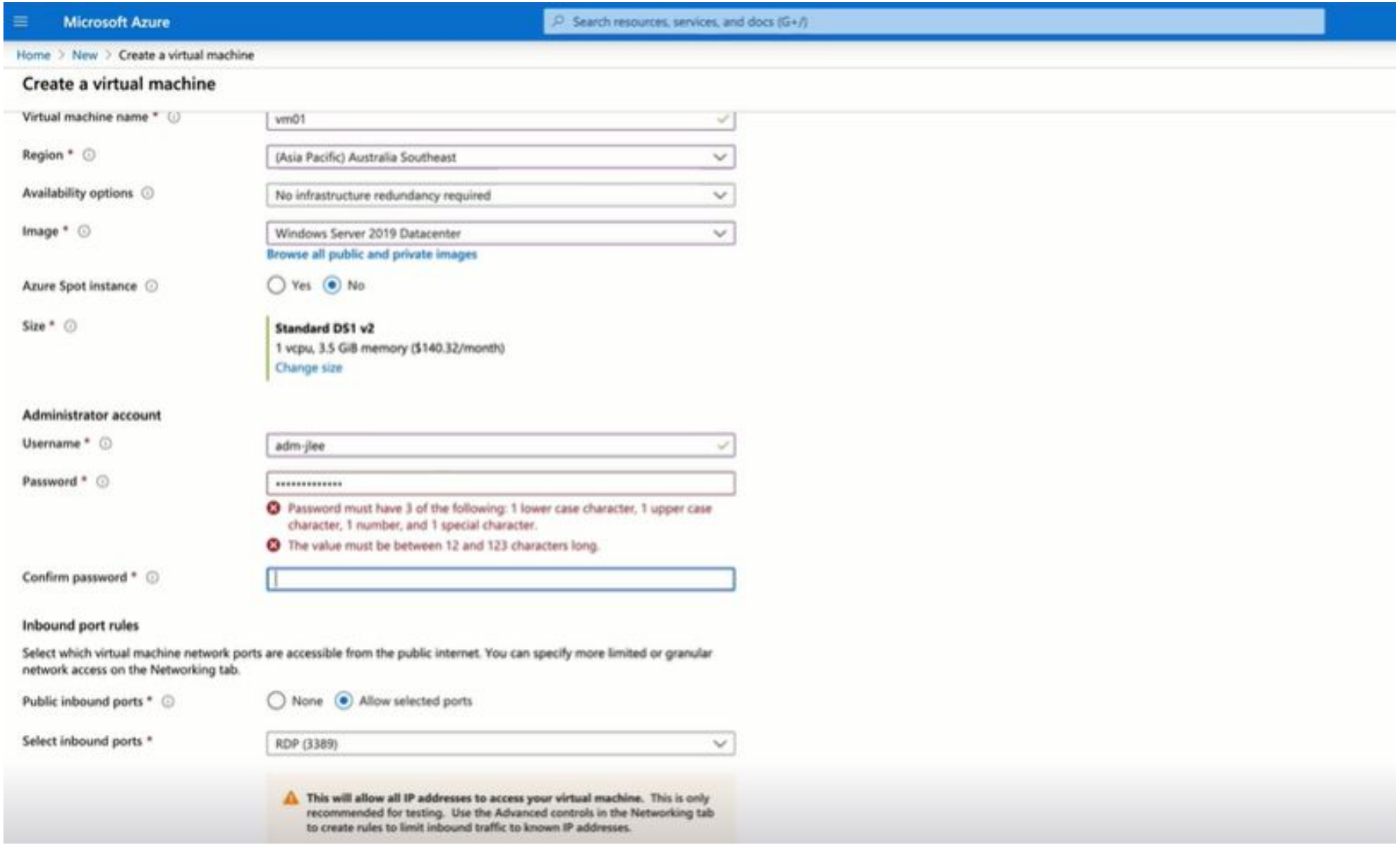
Purpose of Virtual Machines



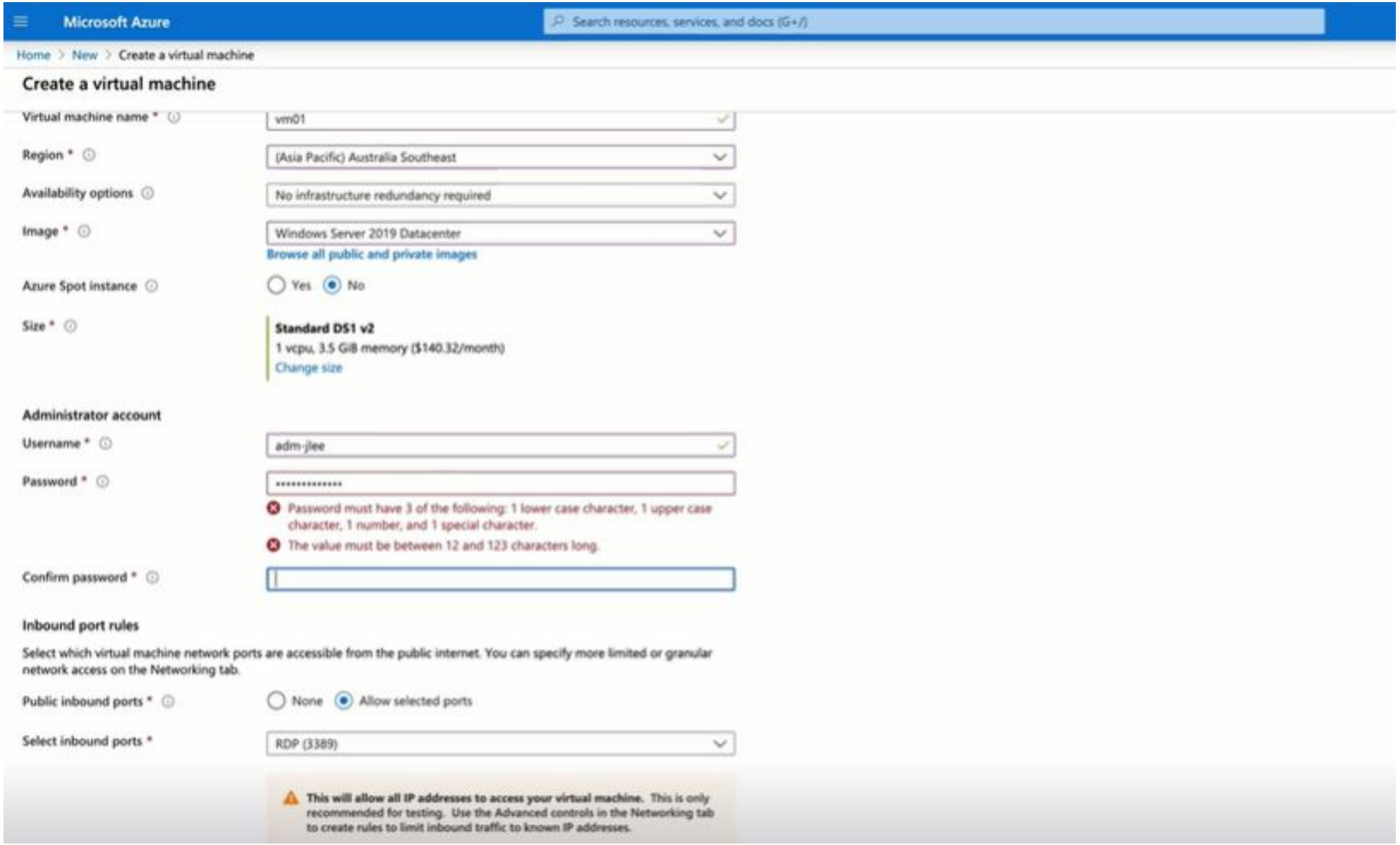
Creating a Virtual Machine



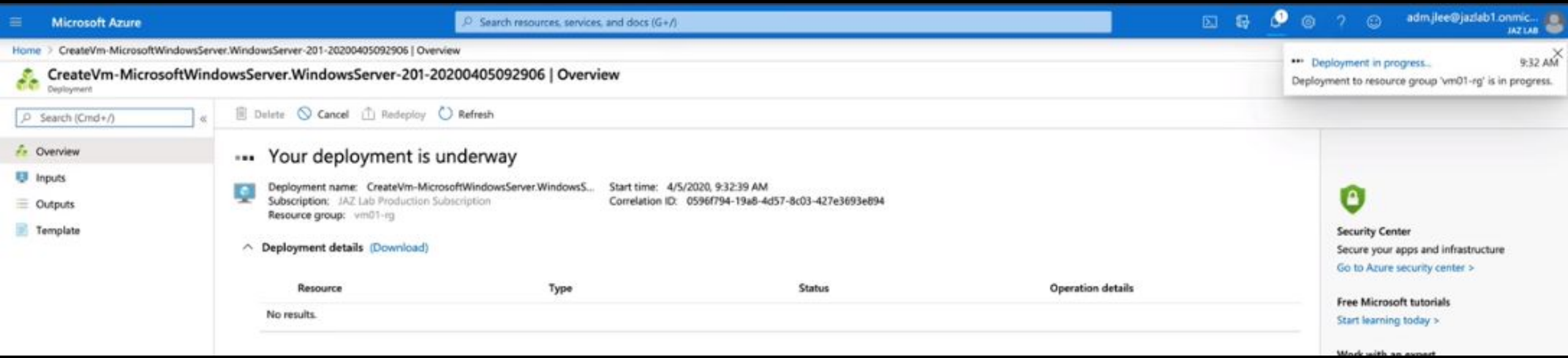
Creating a Virtual Machine



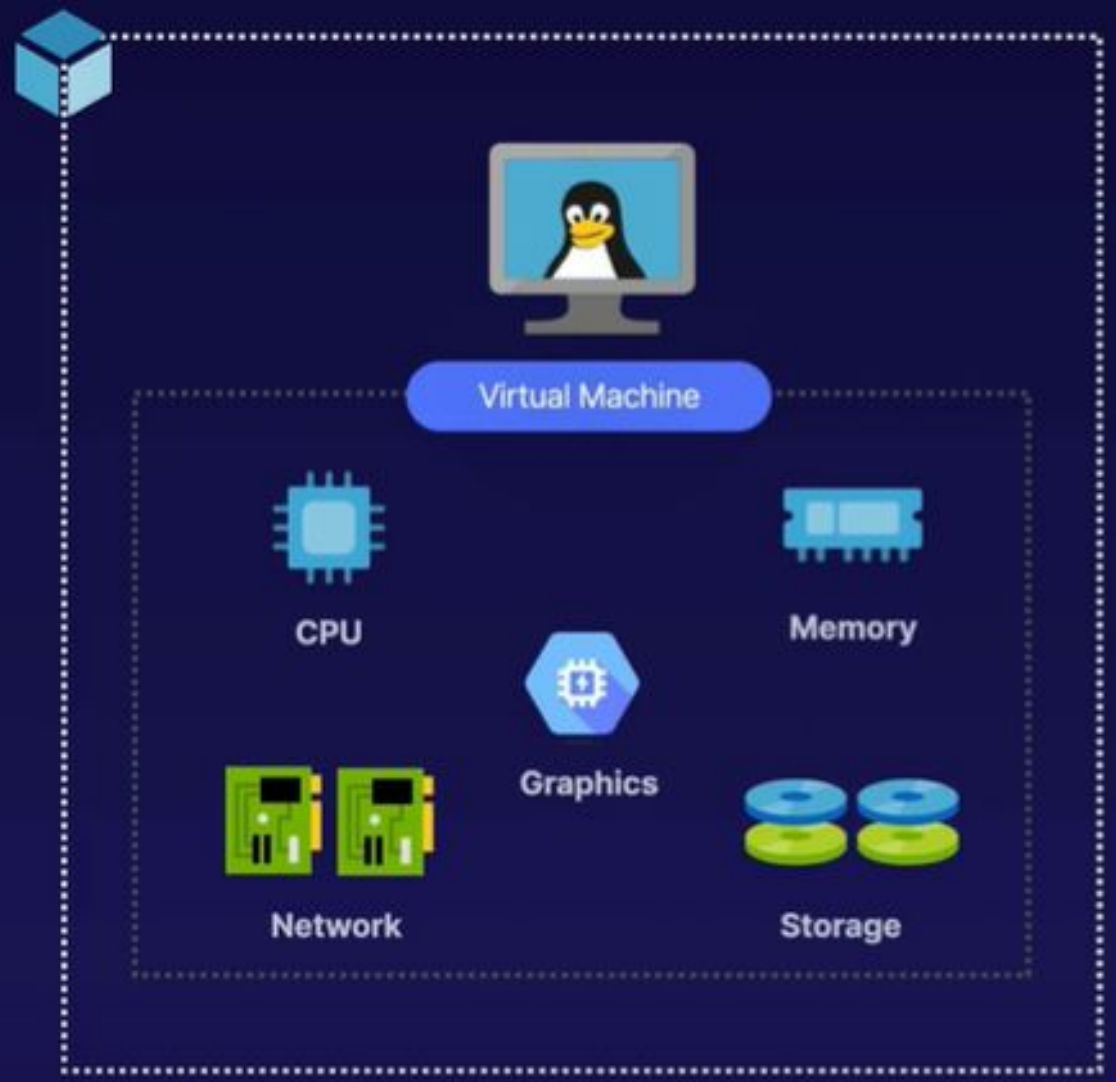
Creating a Virtual Machine



Creating a Virtual Machine



Creating a Virtual Machine



Virtual Machine Size

The size of a virtual machine influences several characteristics.

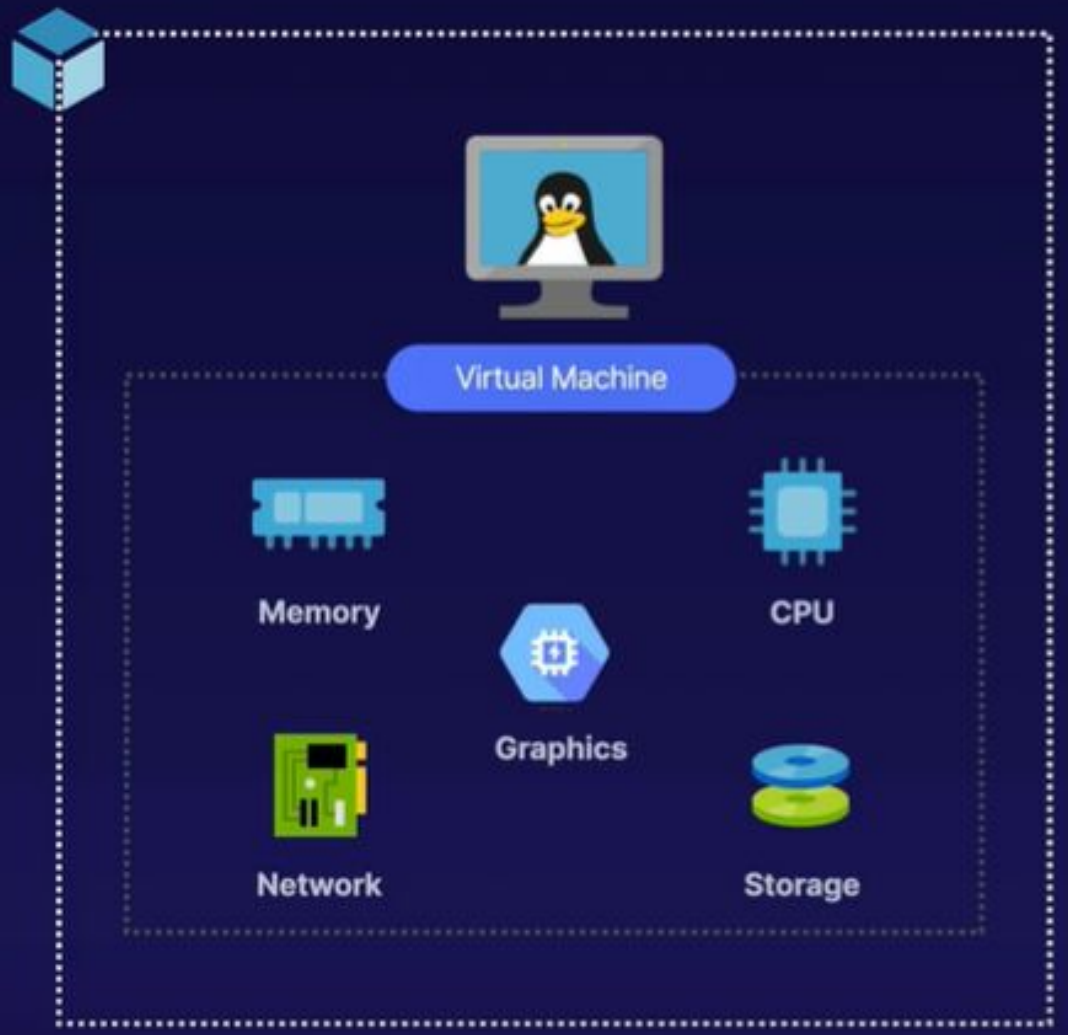
Important Considerations:

- CPU resource allocation
- Available total memory
- Graphics capabilities
- Network interface card (NIC) performance
- Storage (Azure Disks) performance
- Influences limits (maximum NICs, disks)

Configuring Virtual Machine Size:

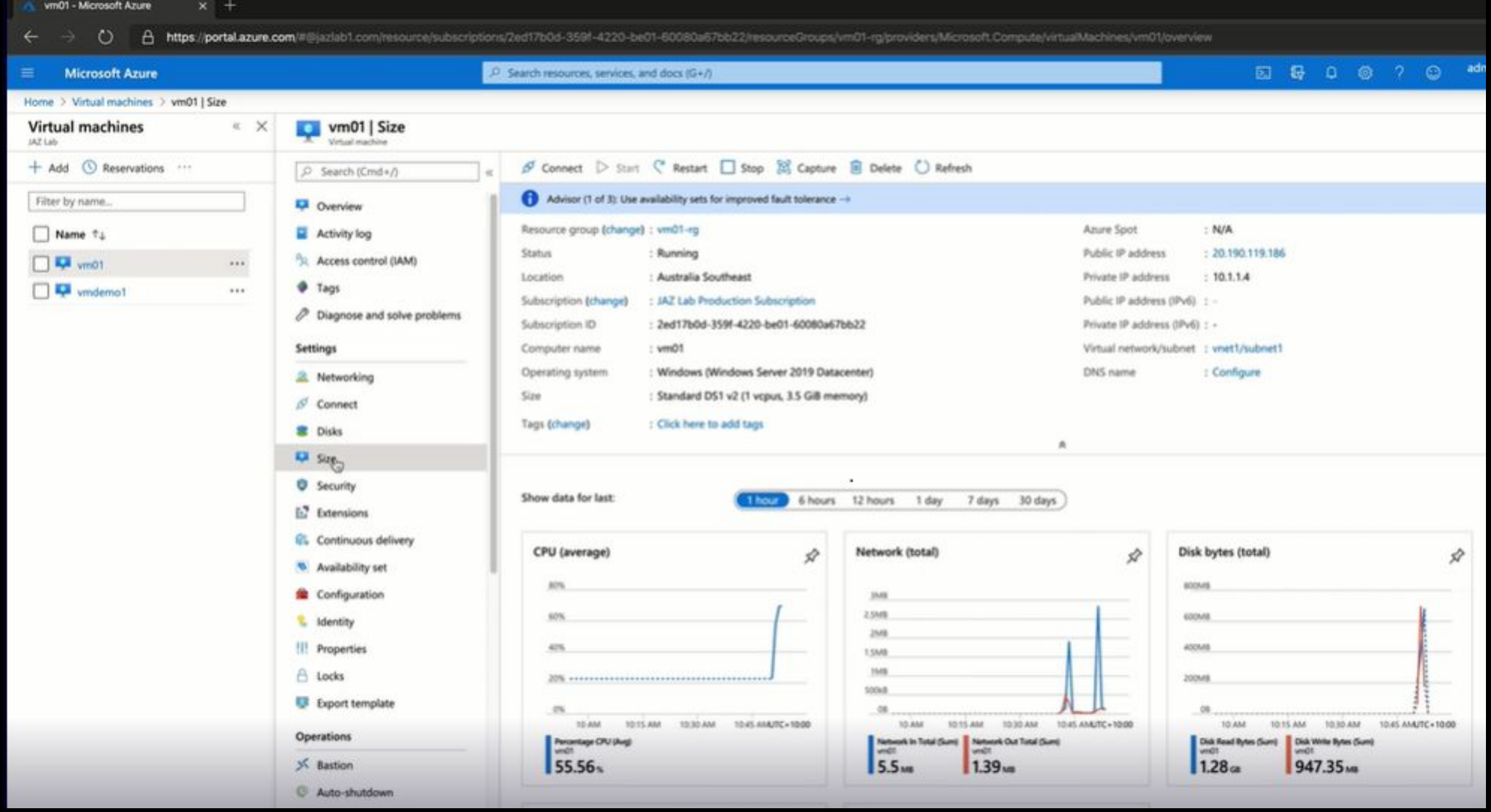
- Can be configured at time of creation
- Can be changed (requires a restart)
- Available sizes depends on:
  - Whether the virtual machine is running
  - The location of the virtual machine

Virtual Machine Size

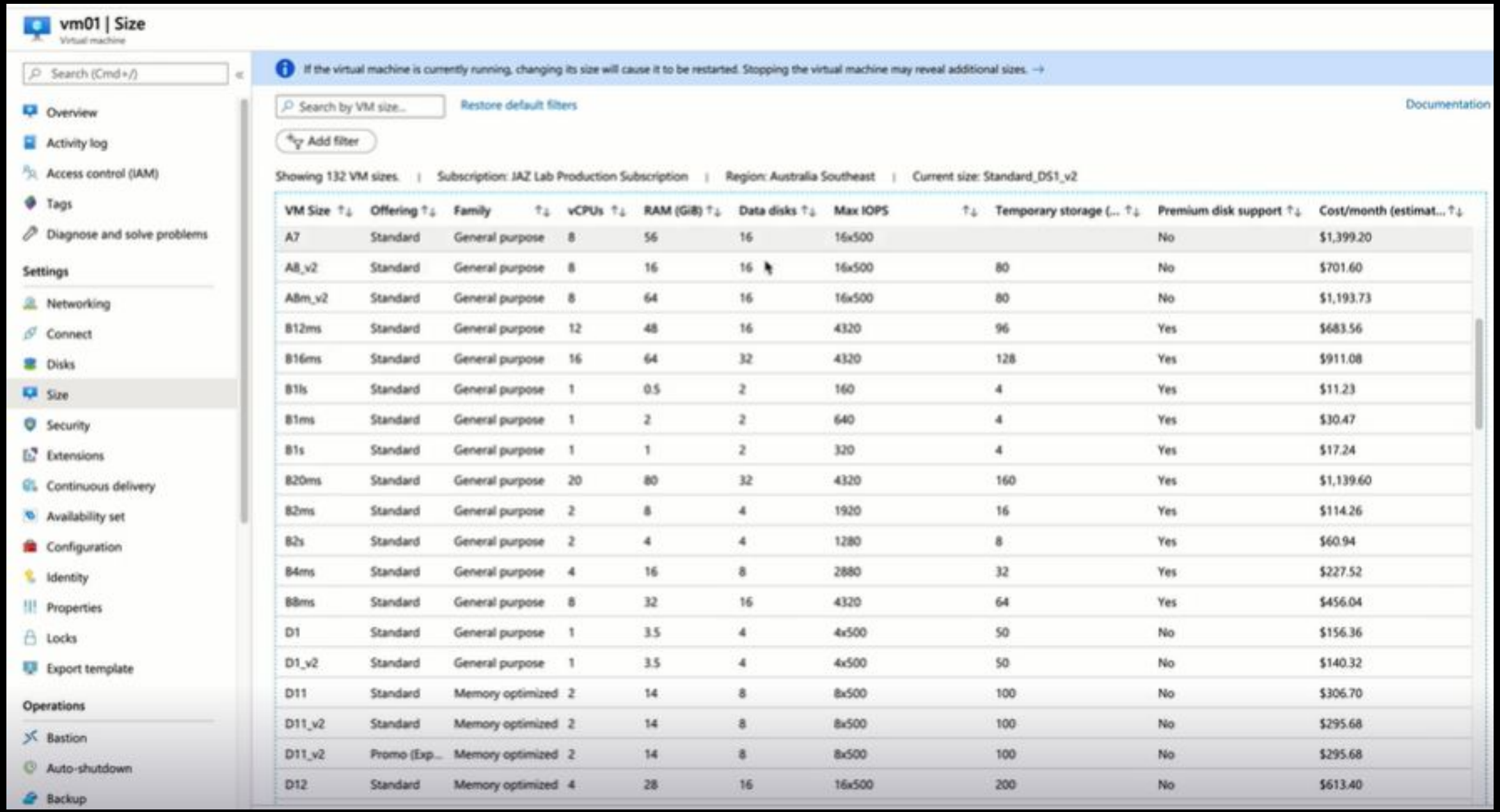


VM Size Type	Description
General Purpose	Balanced CPU-to-memory ratio. Ideal for testing and development, small to medium databases, and low to medium traffic web servers.
Compute Optimized	High CPU-to-memory ratio. Good for medium traffic web servers, network appliances, batch processes, and application servers.
Memory Optimized	High memory-to-CPU ratio. Great for relational database servers, medium to large caches, and in-memory analytics.
Storage Optimzed	High disk throughput and IO ideal for Big Data, SQL, NoSQL databases, data warehousing and large transactional databases.
GPU	Specialized virtual machines targeted for heavy graphic rendering and video editing, as well as model training and inferencing (ND) with deep learning. Available with single or multiple GPUs.
HPC	Our fastest and most powerful CPU virtual machines with optional high-throughput network interfaces (RDMA).

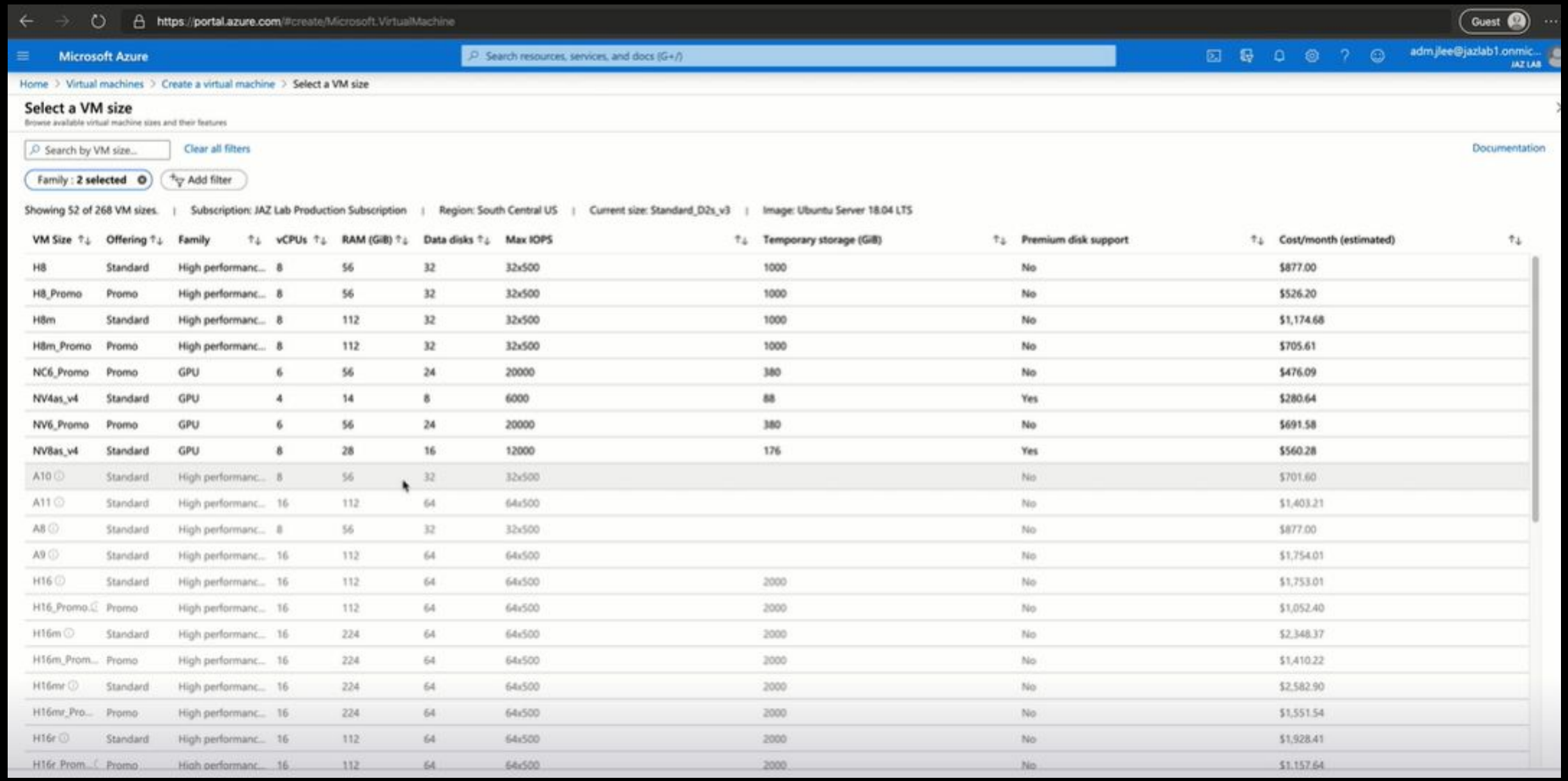
Creating a Virtual Machine



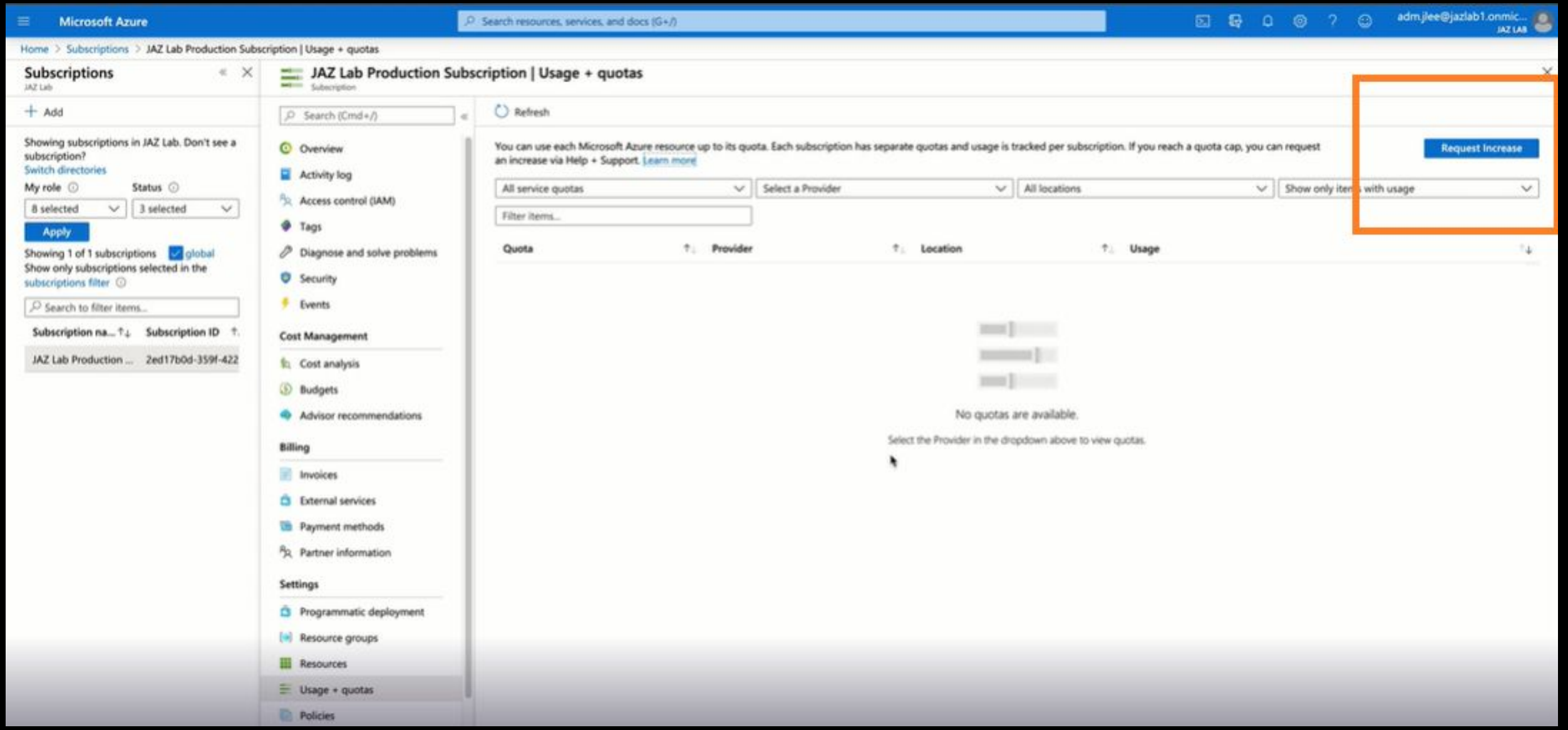
Virtual Machine size



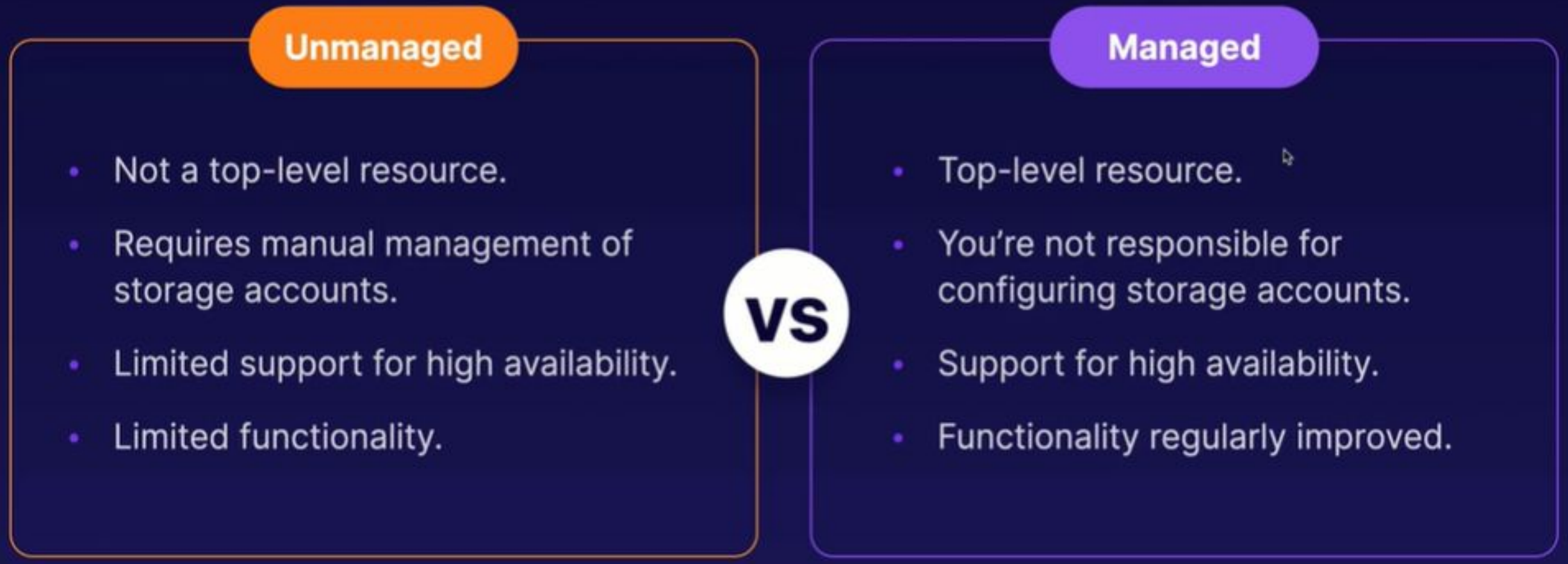
Virtual Machine size



Quotas for Virtual Machine Sizes

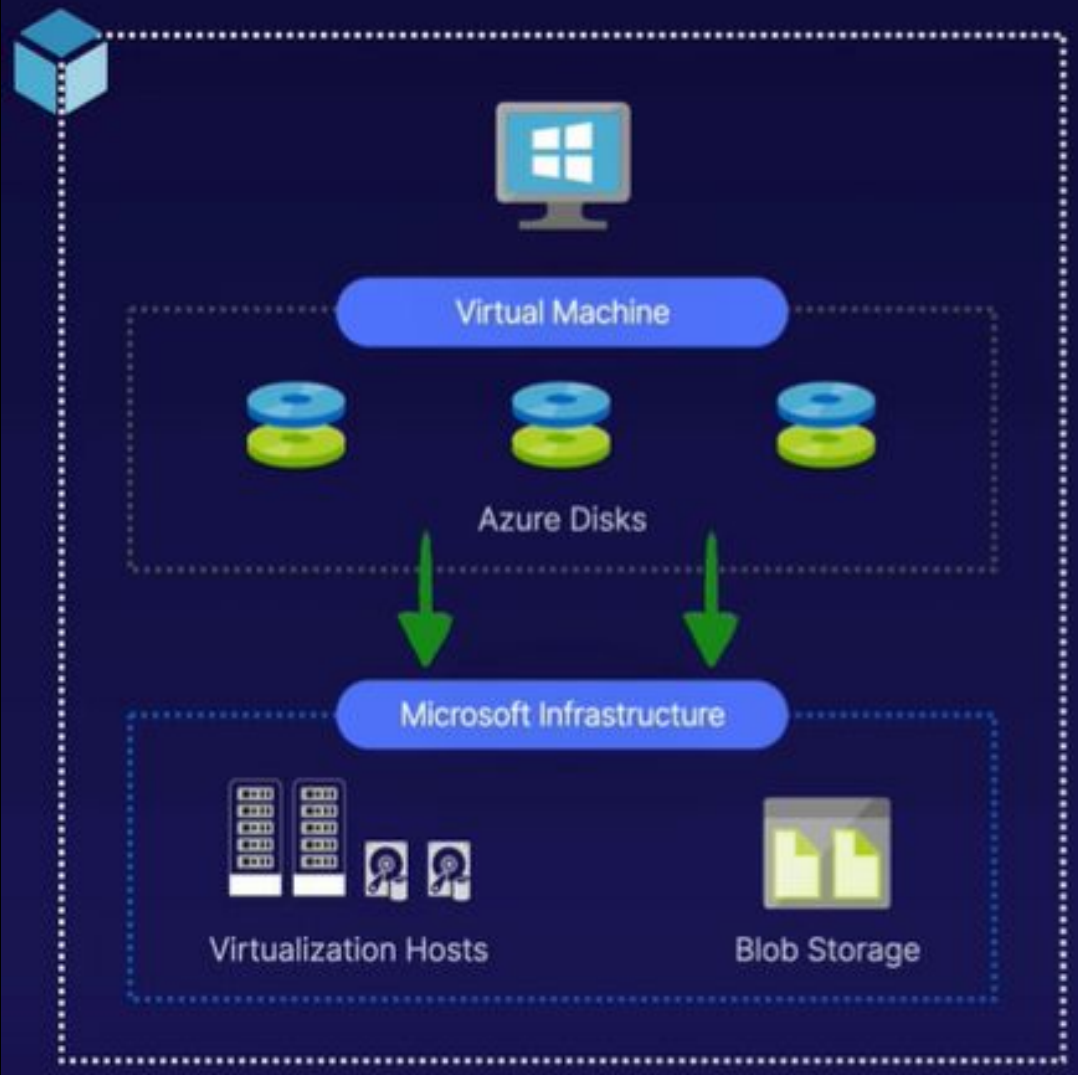


Quotas for Virtual Machine Sizes



Virtual Machine storage

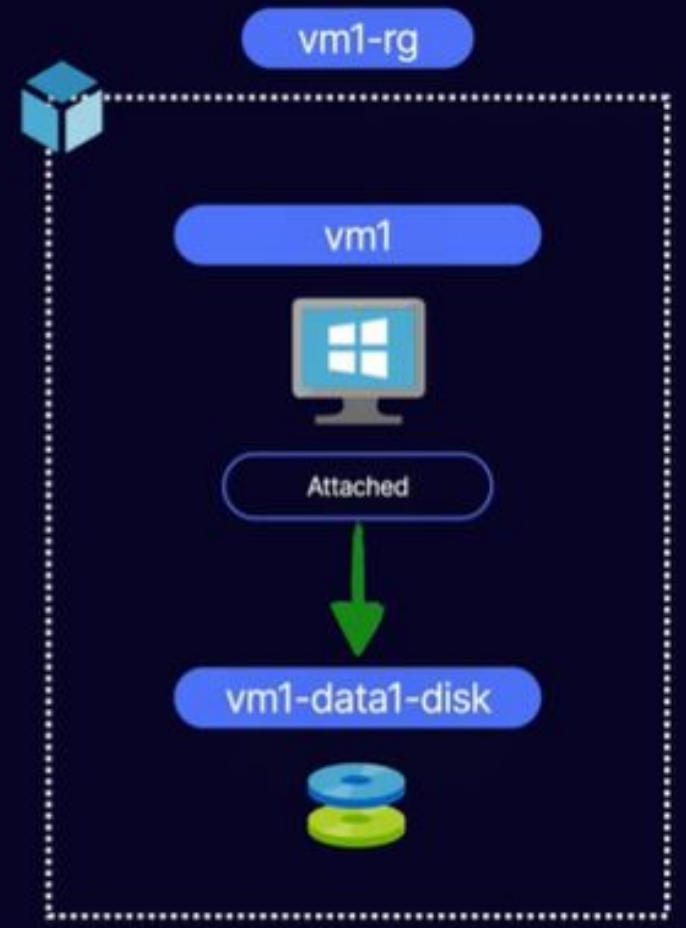
Unmanaged vs Managed Disks



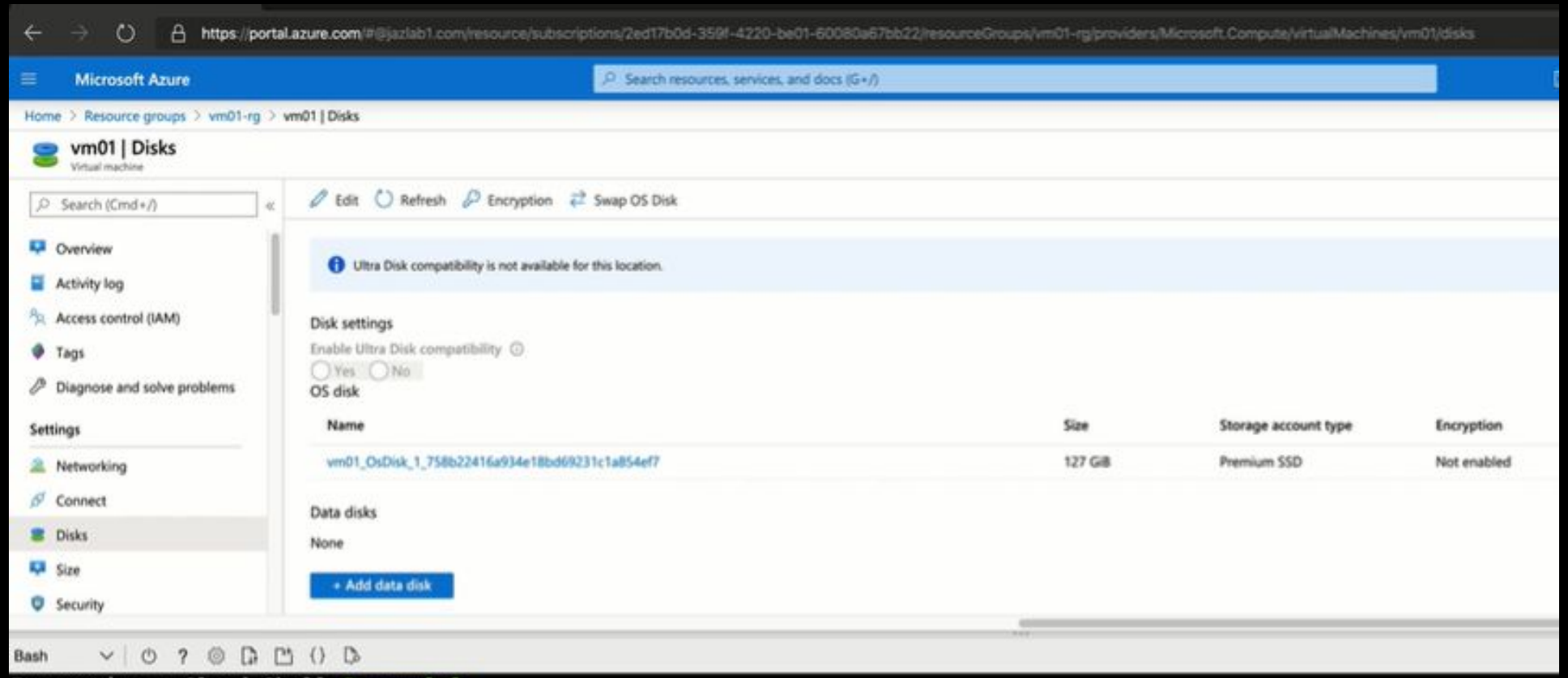
Disk Type	Description
OS Disk	Required by all virtual machines, the OS disk is built from an image containing a pre-installed operating system (OS). **OS disks are a default disk.**
Temporary Disk	A local disk (local to the underlying Microsoft host infrastructure which is hosting the virtual machine) that is NOT persistent. **Temporary disks are a default disk.**
Data Disk	A disk used to store information persistently, such as application data, etc.
Ephemeral OS Disk	A special type of OS Disk which contains an operating system installation, but where data cannot be changed.

Disk Types

```bash
# Add a data disk to an existing VM using Azure CLI

rgName="vm1-rg"
vmName="vm1"
diskName="vm1-data1-disk"

# Create a new disk
az disk create --name "vm1-data1-disk" \
    --resource-group $rgName \
    --location "Australia Southeast" \
    --size-gb 10

# Add disk to existing VM
az vm disk attach --vm-name $vmName \
    --name $diskName \
    --resource-group $rgName
```

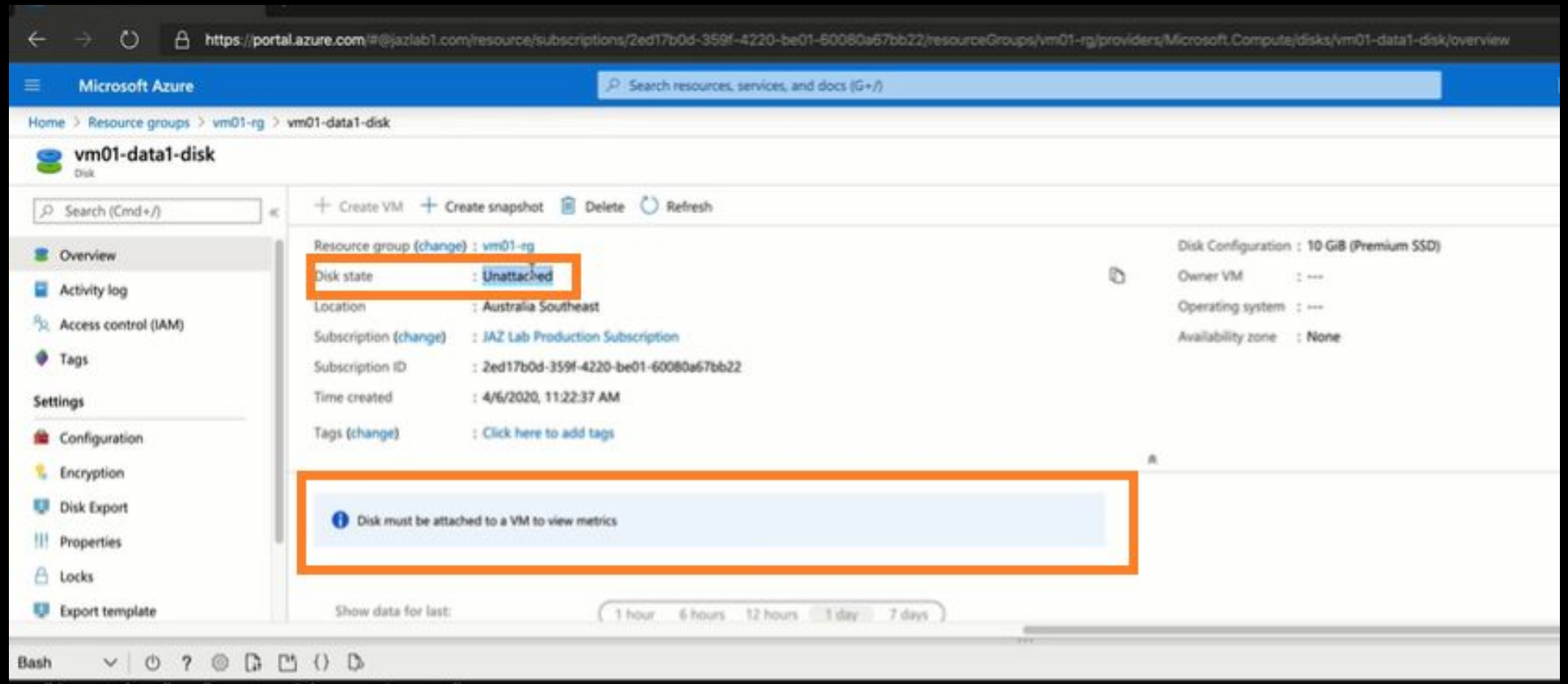


Adding a Data Disk to a VM



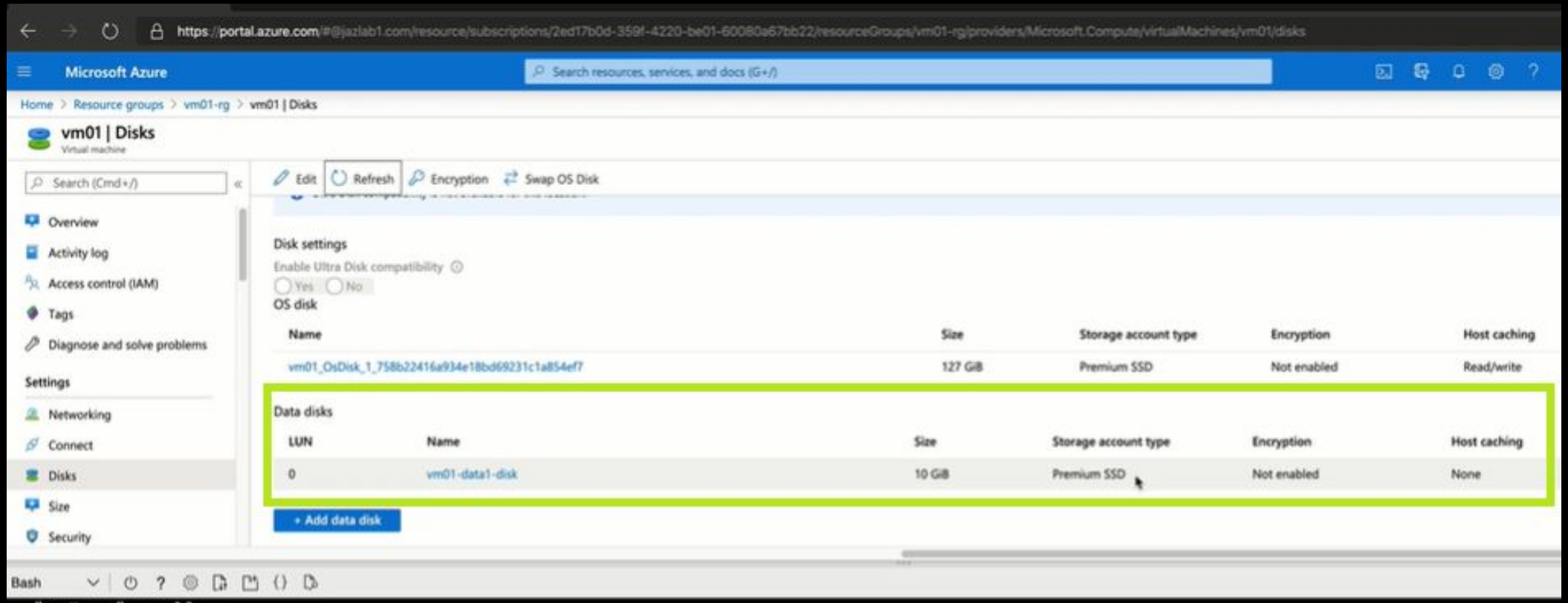

```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

admin@Azure:~$ rgName="vm01-rg"
admin@Azure:~$ vmName="vm01"
admin@Azure:~$ diskName="vm01-data1-disk"
admin@Azure:~$
admin@Azure:~$ az disk create --name $diskName \
> --resource-group $rgName \
> --location "Australia Southeast" \
> --size-gb 10
```

Adding a Data Disk to a VM



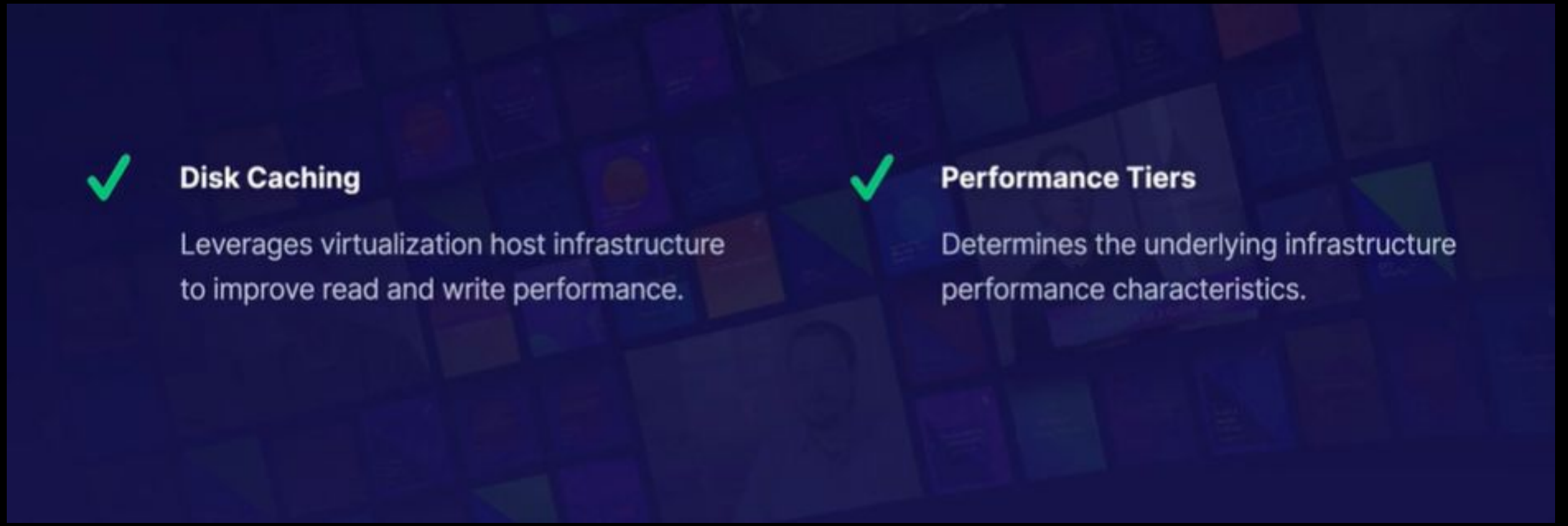

```
"location": "australiasoutheast",
"managedBy": null,
"name": "vm01-data1-disk",
"osType": null,
"provisioningState": "Succeeded",
"resourceGroup": "vm01-rg",
"sku": {
  "name": "Premium_LRS",
  "tier": "Premium"
},
"tags": {},
"timeCreated": "2020-04-06T01:22:37.625757+00:00",
"type": "Microsoft.Compute/disks",
"uniqueId": "27aa018c-739c-4704-8926-6af2f3f155c7",
```


Adding a Data Disk to a VM

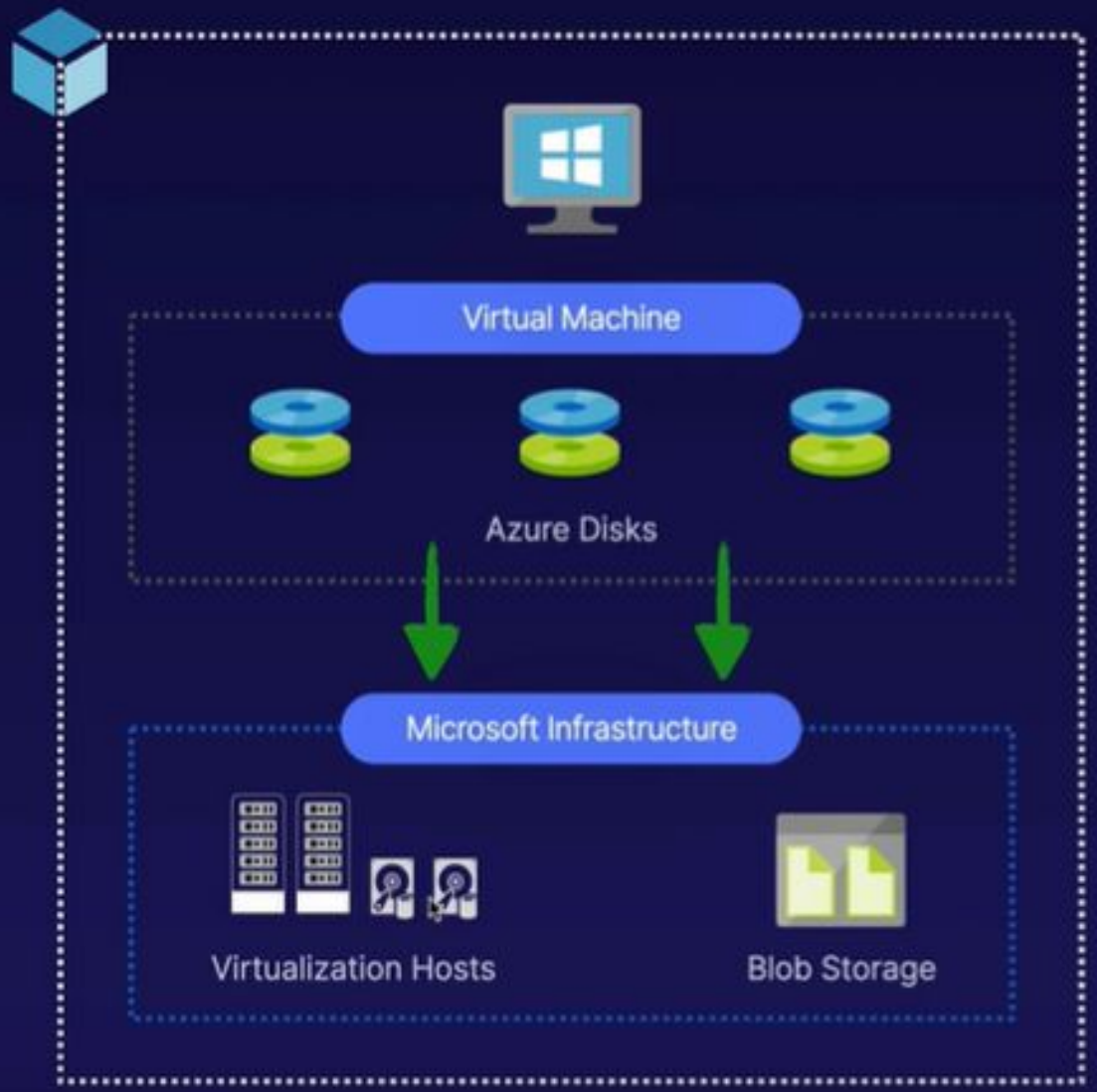



```
  "osType": null,
  "provisioningState": "Succeeded",
  "resourceGroup": "vm01-rg",
  "sku": {
    "name": "Premium_LRS",
    "tier": "Premium"
  },
  "tags": {},
  "timeCreated": "2020-04-06T01:22:37.625757+00:00",
  "type": "Microsoft.Compute/disks",
  "uniqueId": "27aa018c-739c-4704-8926-6af2f3f155c7",
  "zones": null
}
admin@Azure:~$ az vm disk attach --vm-name $vmName \
```

Adding a Data Disk to a VM



Azure Disk Performance



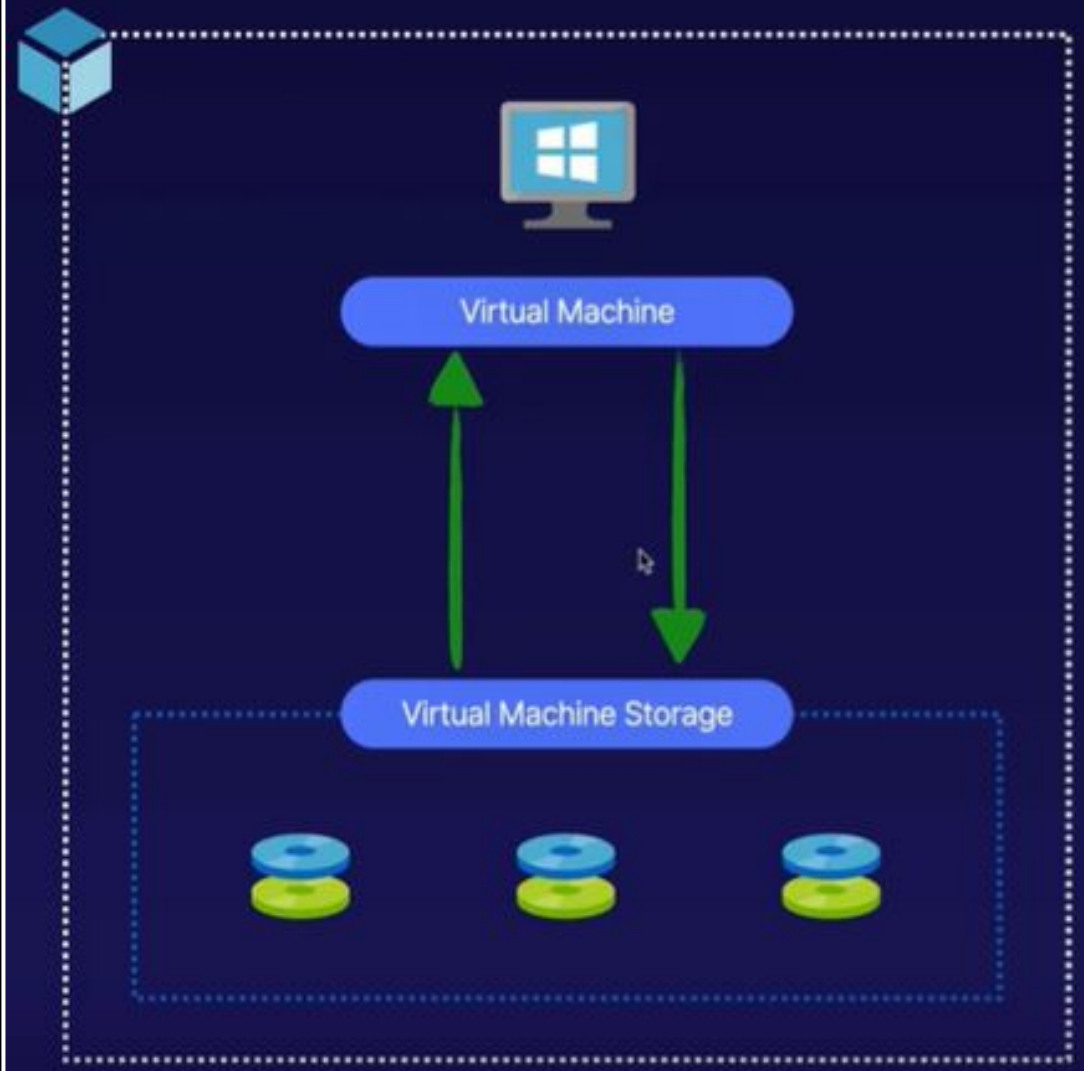
Disk Caching

Disk caching (also referred to as **host caching**) is a feature that leverages specialised high-performance (read and write) storage systems to ensure data can be accessed **very fast**.

BlobCache

Disk caching in Azure leverages a technology referred to as BlobCache, which uses a combination of host SSDs and memory to provide each to virtual machines.

Disk Caching



Cache Option	Description
Read/Write	Leverages a cache for both reads and writes. This MUST have proper support from applications on the VM to ensure data is correctly persisted from cache to storage. This type of cache is often referred to as "write-back caching". Note: this option is enabled by default for OS disks, and also supports Data disks.
Read-Only	Improves read performance (throughput and latency) by reading data from the cache, with all writes going direct to storage. This is often referred to as "read-through caching".
None	Both read and write operations do not use a cache and are instead directed to the storage backend.

Disk Caching

Azure Disk Performance Tiers

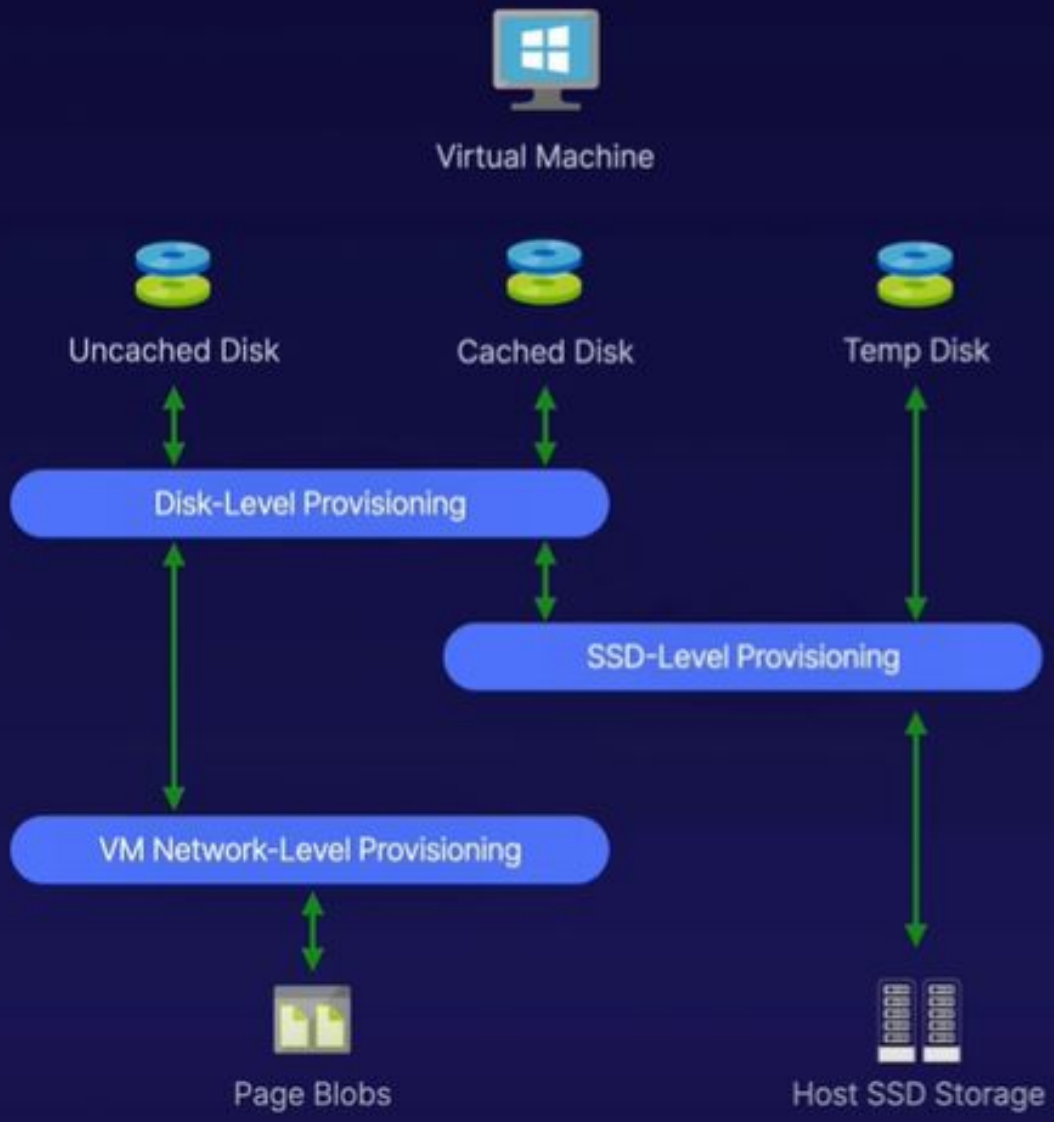
Microsoft provide several options to cater to the various performance requirements of different workloads.

Important Considerations:

- Different performance have different:
  - Maximum disk sizes.
  - Maximum throughput.
  - Maximum IOPS.
- Note, not all VM sizes support all performance tiers

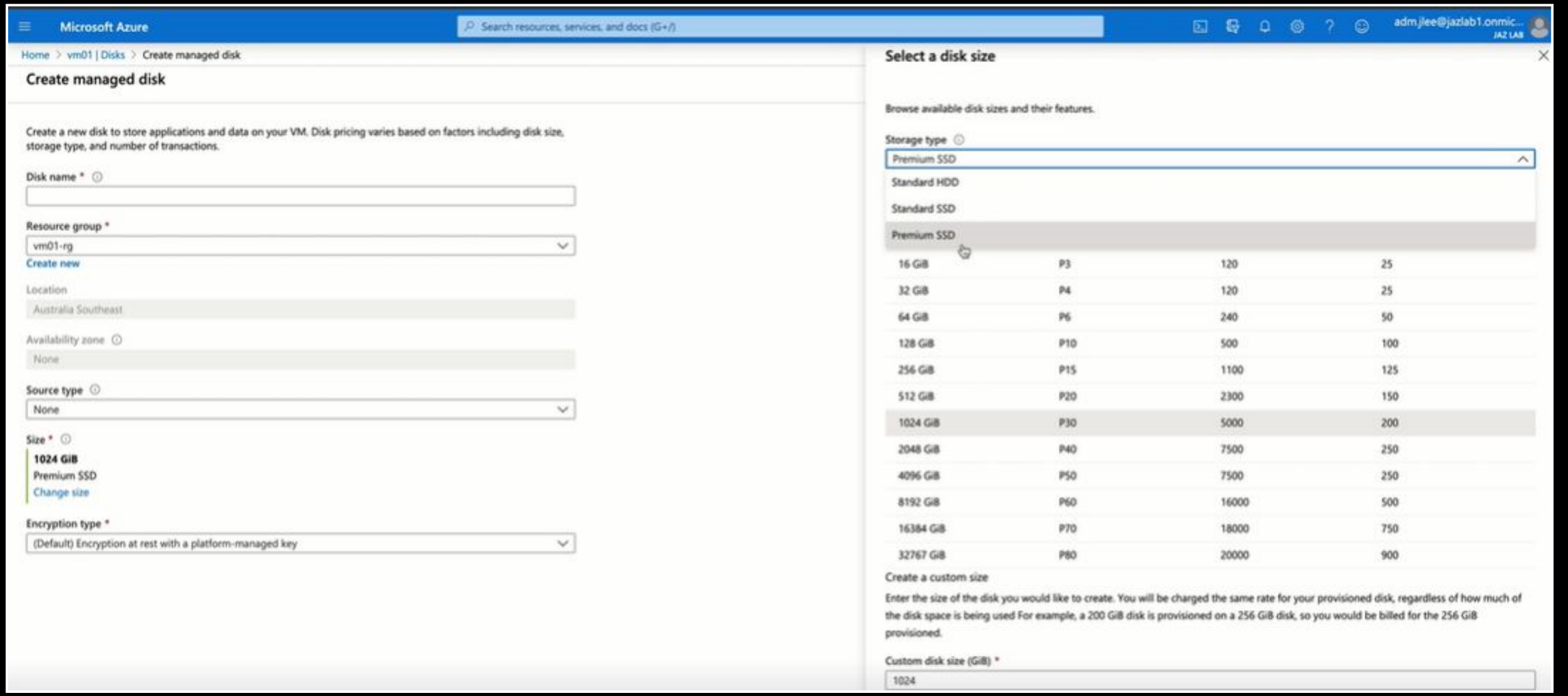
Tier	Description
Ultra	Designed for extremely input/output intensive workloads (for example, SAP HANA, top-tier databases, etc.).
Premium SSD	Intended for production workloads with high-performances requirements.
Standard SSD	Built for light-usage applications, or development and testing workloads.
Standard HDD	Only intended for backup or non-critical workloads.

Disk Caching

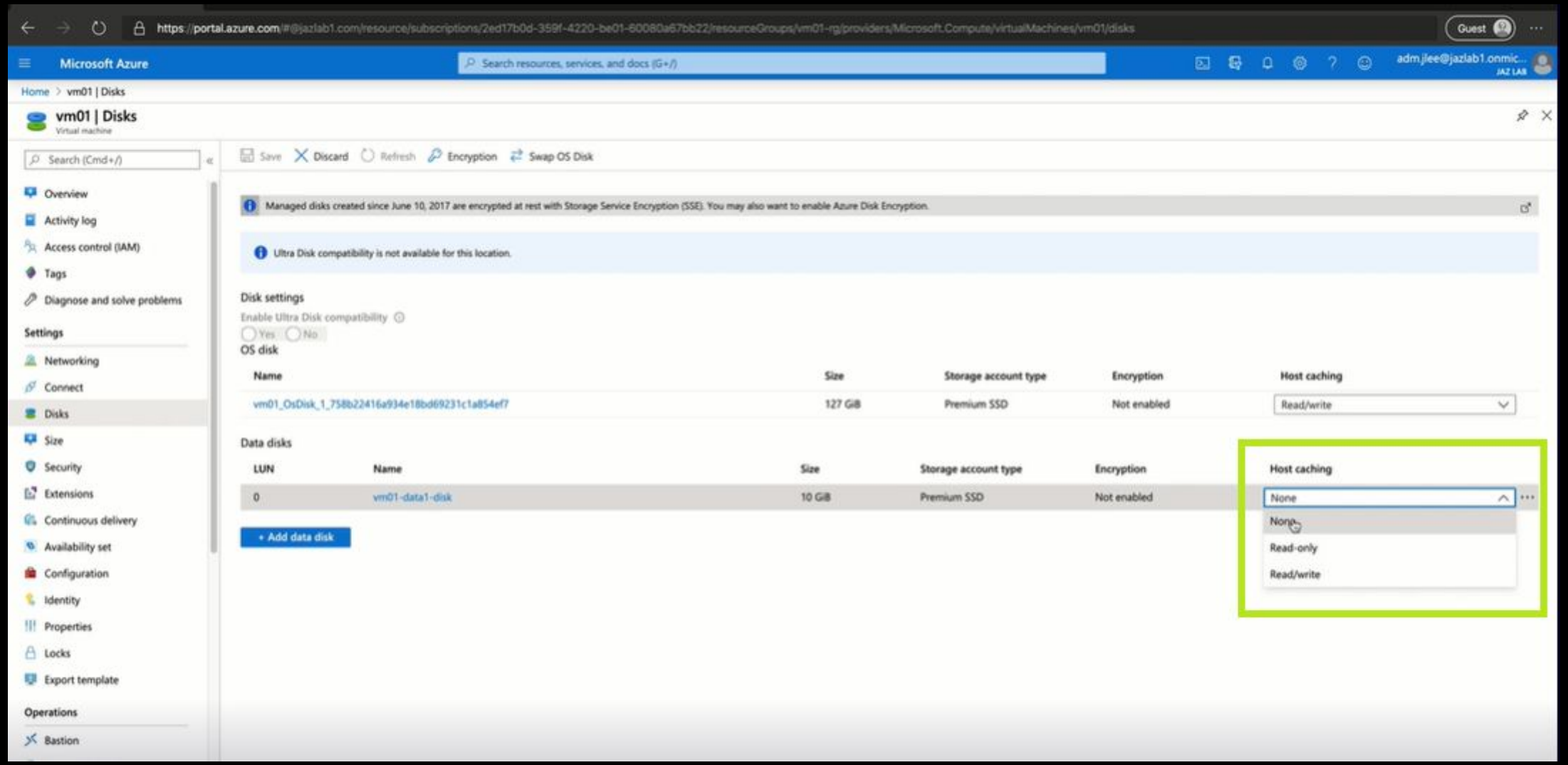


Tier	Description
Ultra	Designed for extremely input/output intensive workloads (for example, SAP HANA, top-tier databases, etc.).
Premium SSD	Intended for production workloads with high-performances requirements.
Standard SSD	Built for light-usage applications, or development and testing workloads.
Standard HDD	Only intended for backup or non-critical workloads.

Performance Tiers



Performance Tiers

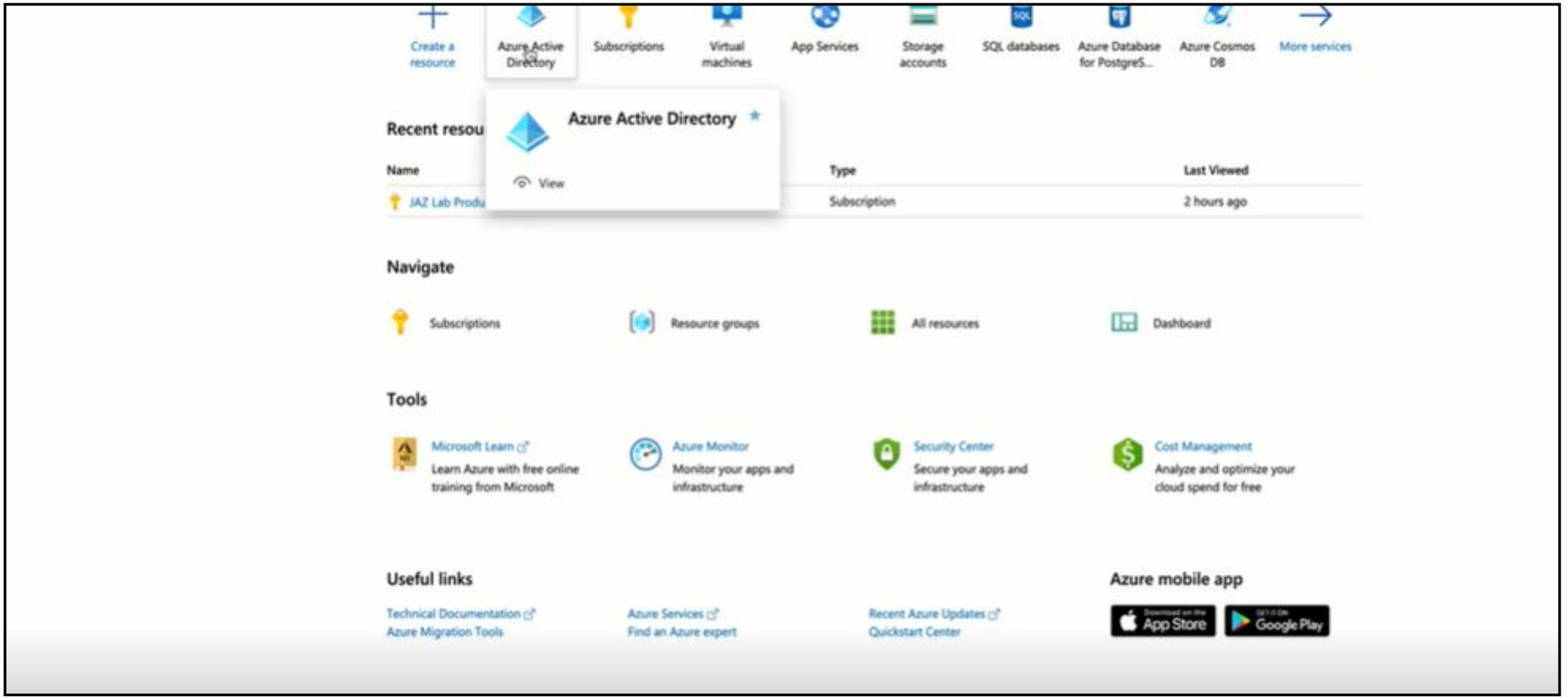


Disk Caching

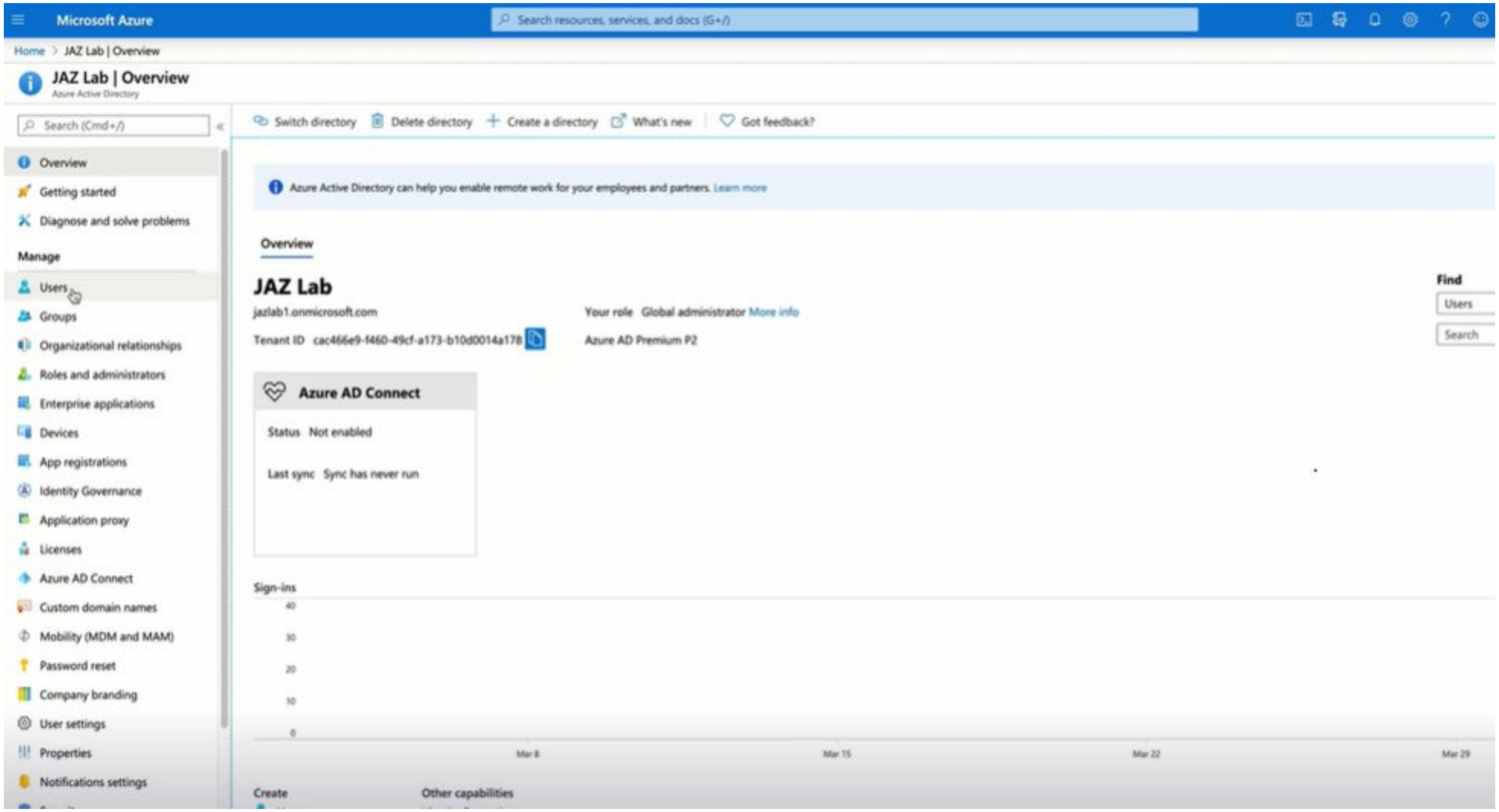
Azure Active Directory


Identity Management
Repository for storing identity information: users, security groups, related metadata (name, contact details, department, etc.)
Enterprise Access Management
Control access to enterprise resources: internal and external applications, single sign-on, device management, etc.
Identity and Access Security
Secure identity and access: multi-factor authentication, just-in-time access, identity protection, identity risk monitoring, etc.

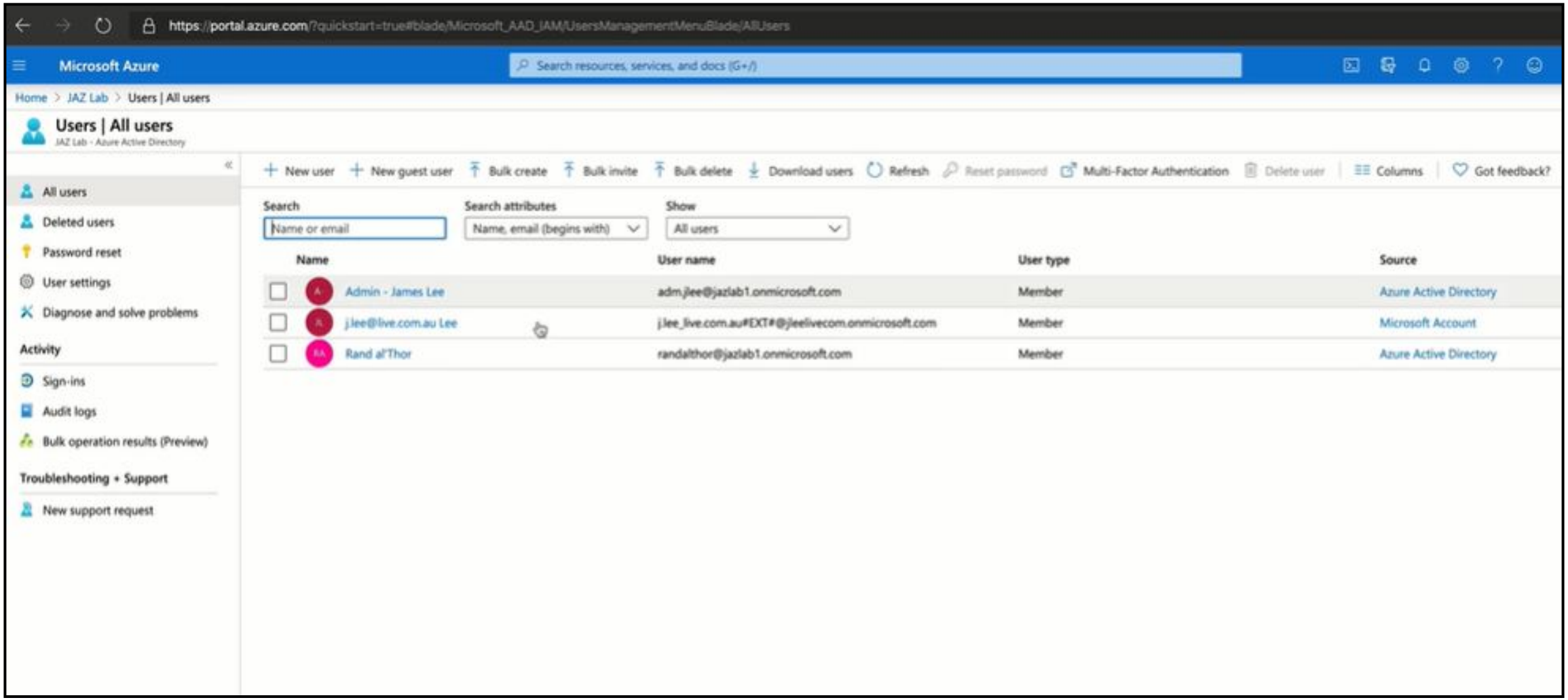
What is it?



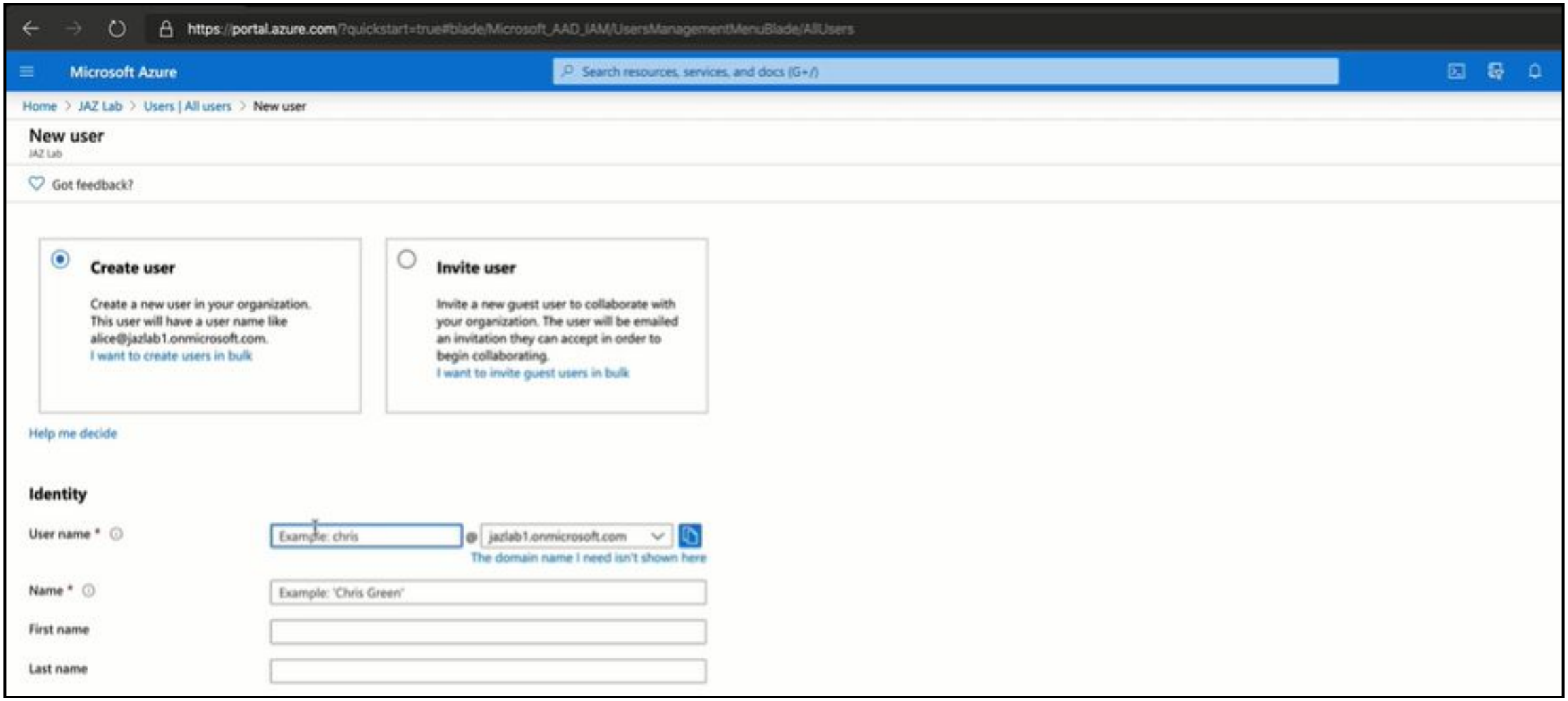
Azure Active Directory



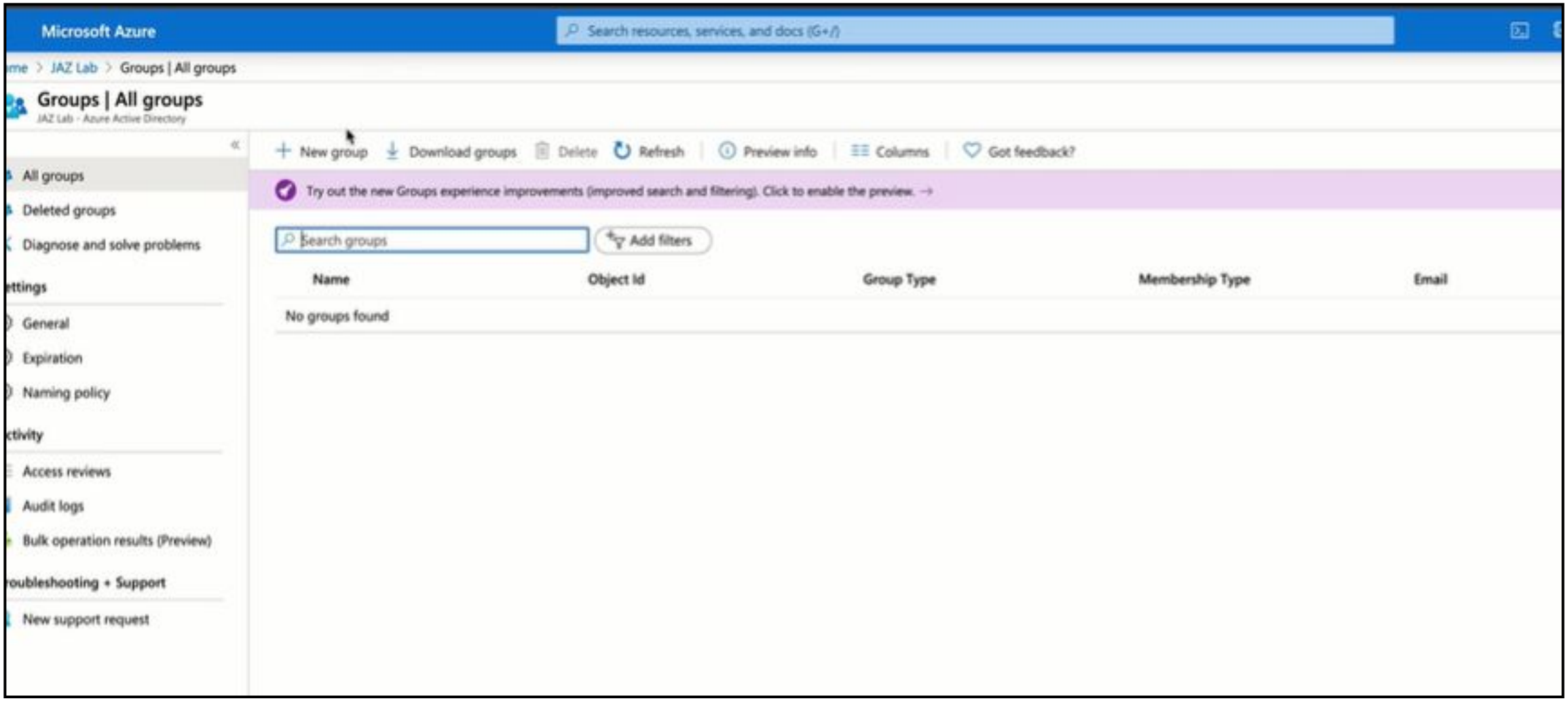
Azure Active Directory


https://portal.azure.com/?quickstart=true#blade/Microsoft_AAD_IAM/UsersManagementMenuBlade/AllUsers
Microsoft Azure
Search resources, services, and docs (G+/)
Home > JAZ Lab > Users | All users
Users | All users
JAZ Lab - Azure Active Directory
All users
Deleted users
Password reset
User settings
Diagnose and solve problems
Activity
Sign-ins
Audit logs
Bulk operation results (Preview)
Troubleshooting + Support
New support request
New user
New guest user
Bulk create
Bulk invite
Bulk delete
Download users
Refresh
Reset password
Multi-Factor Authentication
Delete user
Columns
Got feedback?
Search
Name or email
Search attributes
Name, email (begins with)
Show
All users
Name
User name
User type
Source
Admin - James Lee
adm.jlee@jazlab1.onmicrosoft.com
Member
Azure Active Directory
j.lee@live.com.au Lee
j.lee_live.com.au#EXT#@jleelivecom.onmicrosoft.com
Member
Microsoft Account
Rand al'Thor
randalthor@jazlab1.onmicrosoft.com
Member
Azure Active Directory

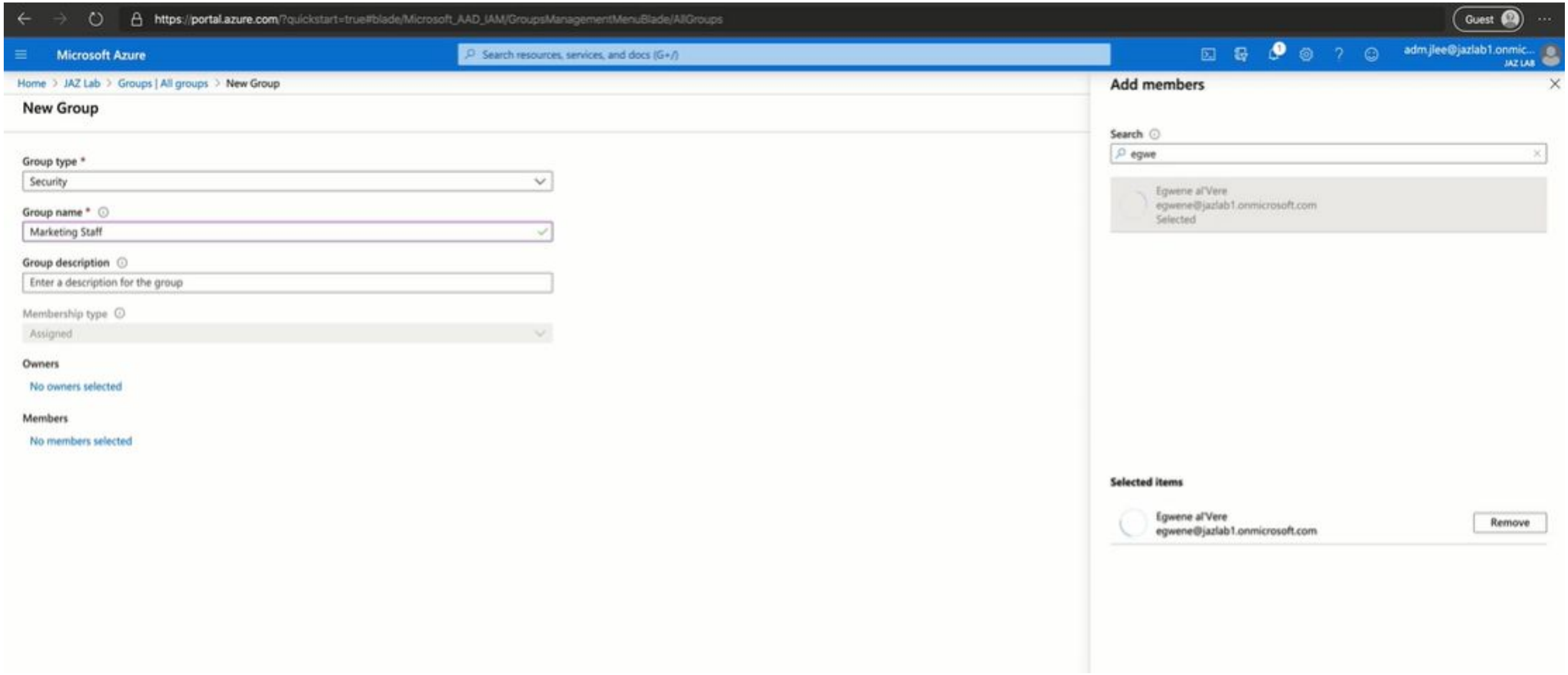
Users in AD



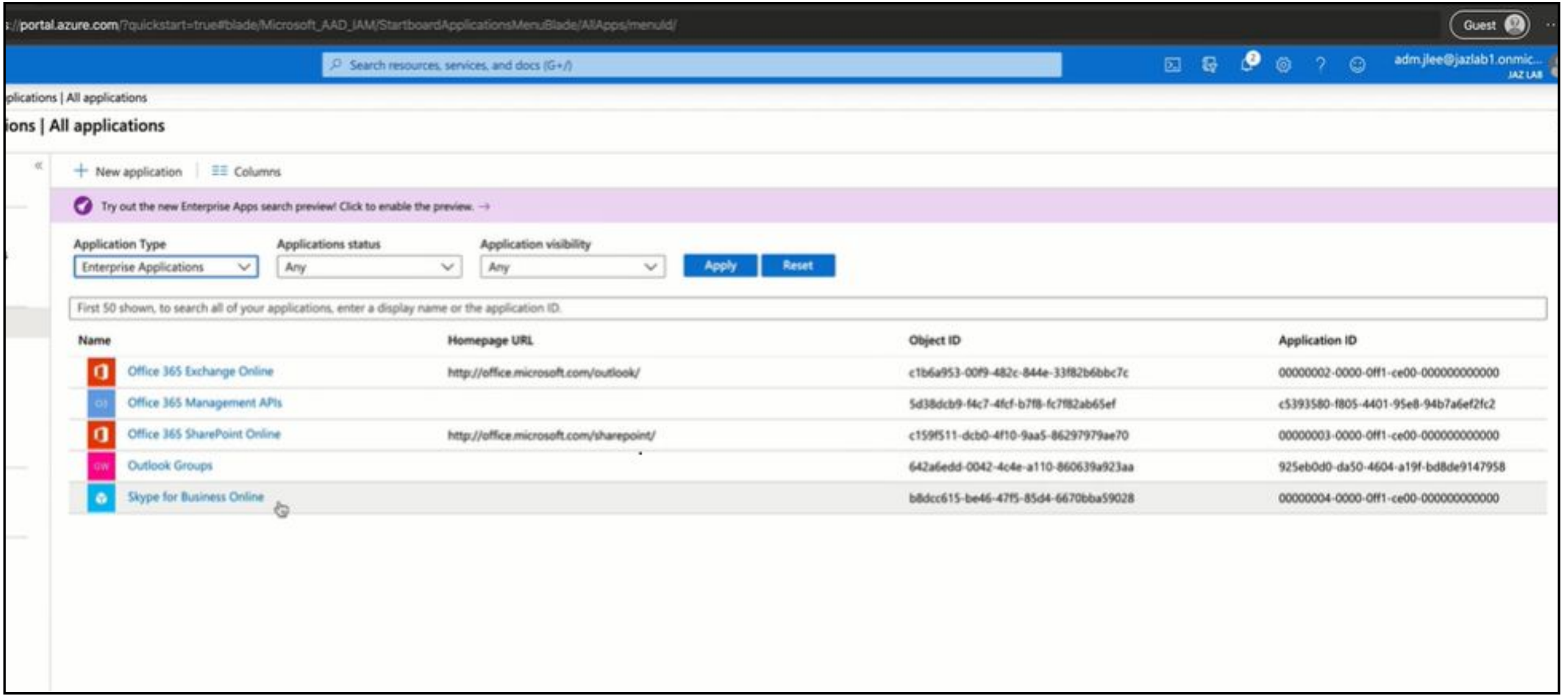
Create New AD user



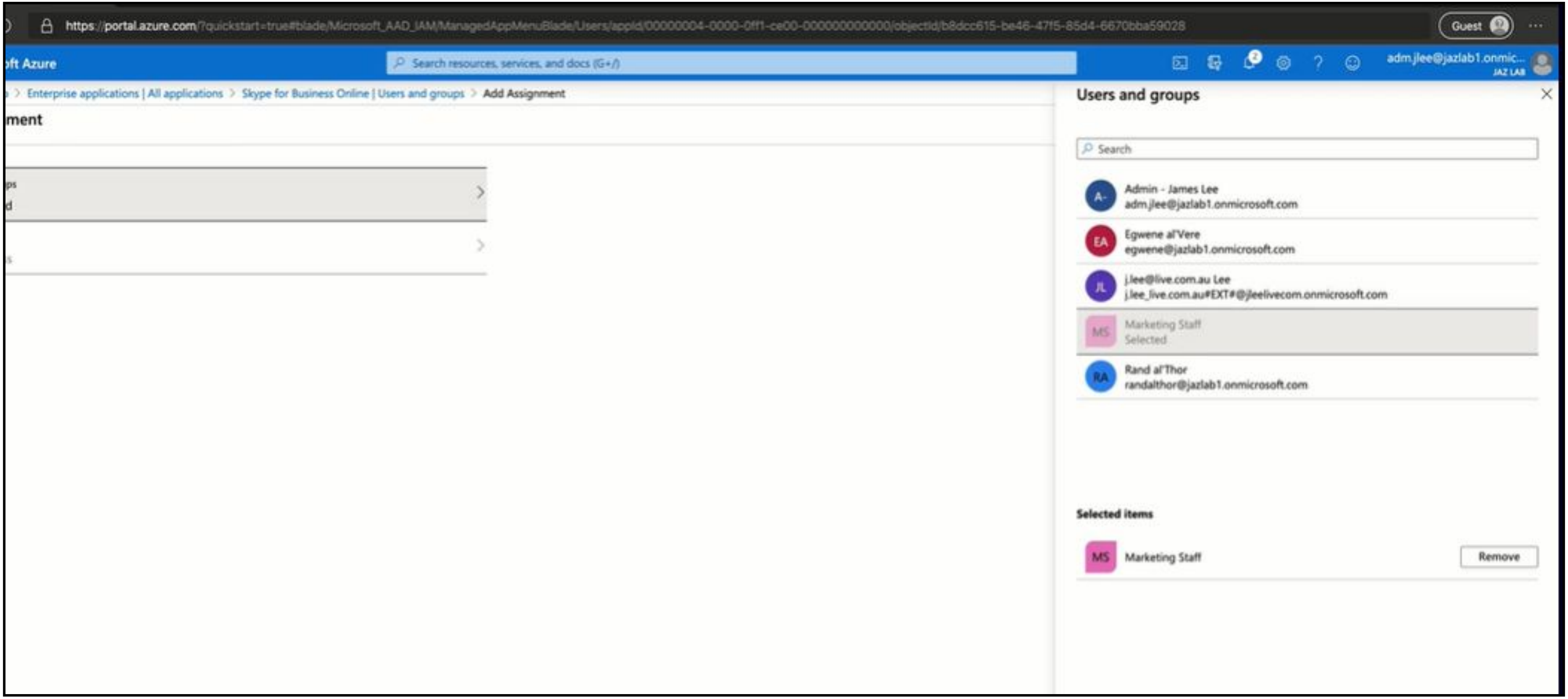
Create New AD group



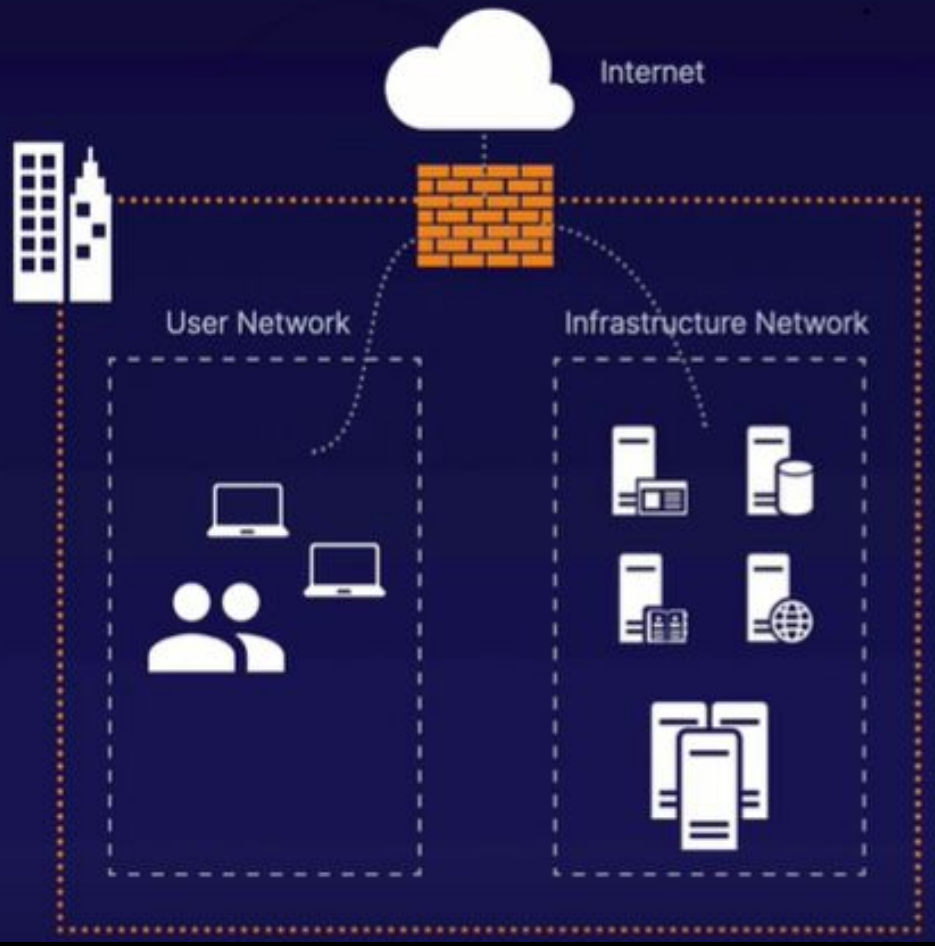
Create New AD user



Add users to enterprise applications



Add users to enterprise applications



Security is based on the location of infrastructure.

Traditional models for security are based on users and resources that are all in a central or "trusted" location:

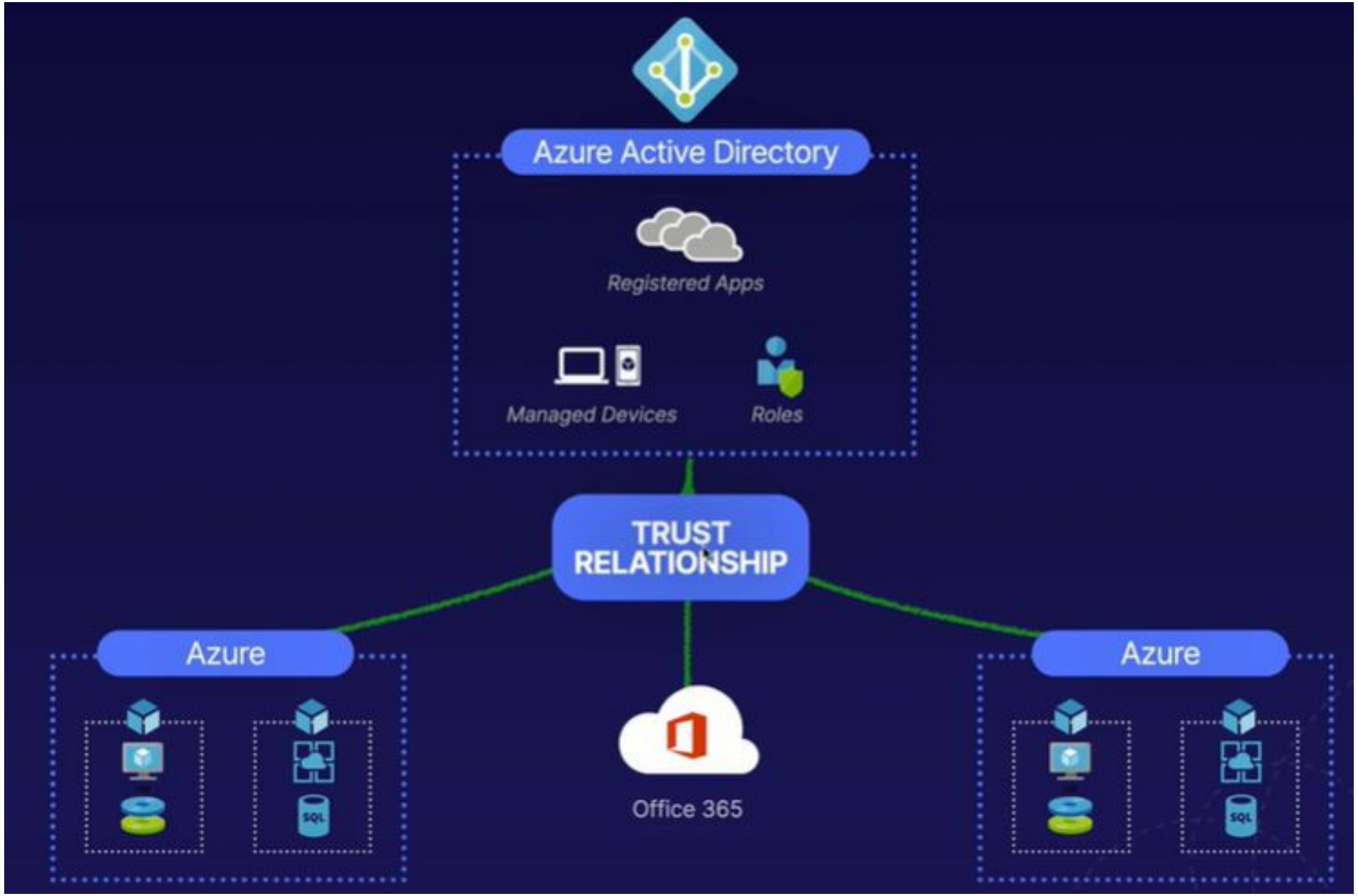
- Infrastructure is housed in a central, secure location
- Access to resources is typically private/internal
- Any external access is protected at the perimeter

Architecture: Traditional Network Security Perimeter

Architecture: Modern Identity-Centric Security

Key Features and Azure AD Services

Implementing Azure AD



Architecture

Tenant

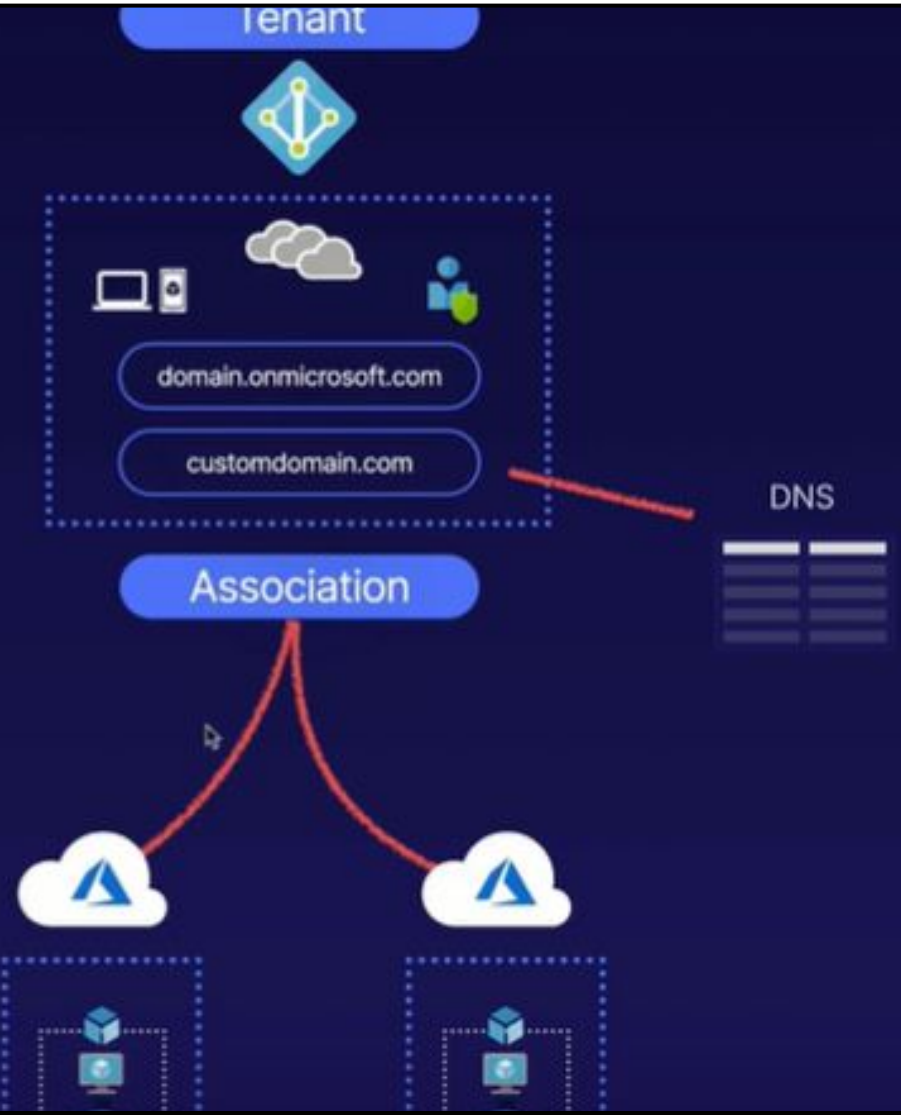
A Tenant is the actual representation of an organization within the Azure Active Directory.

Domain

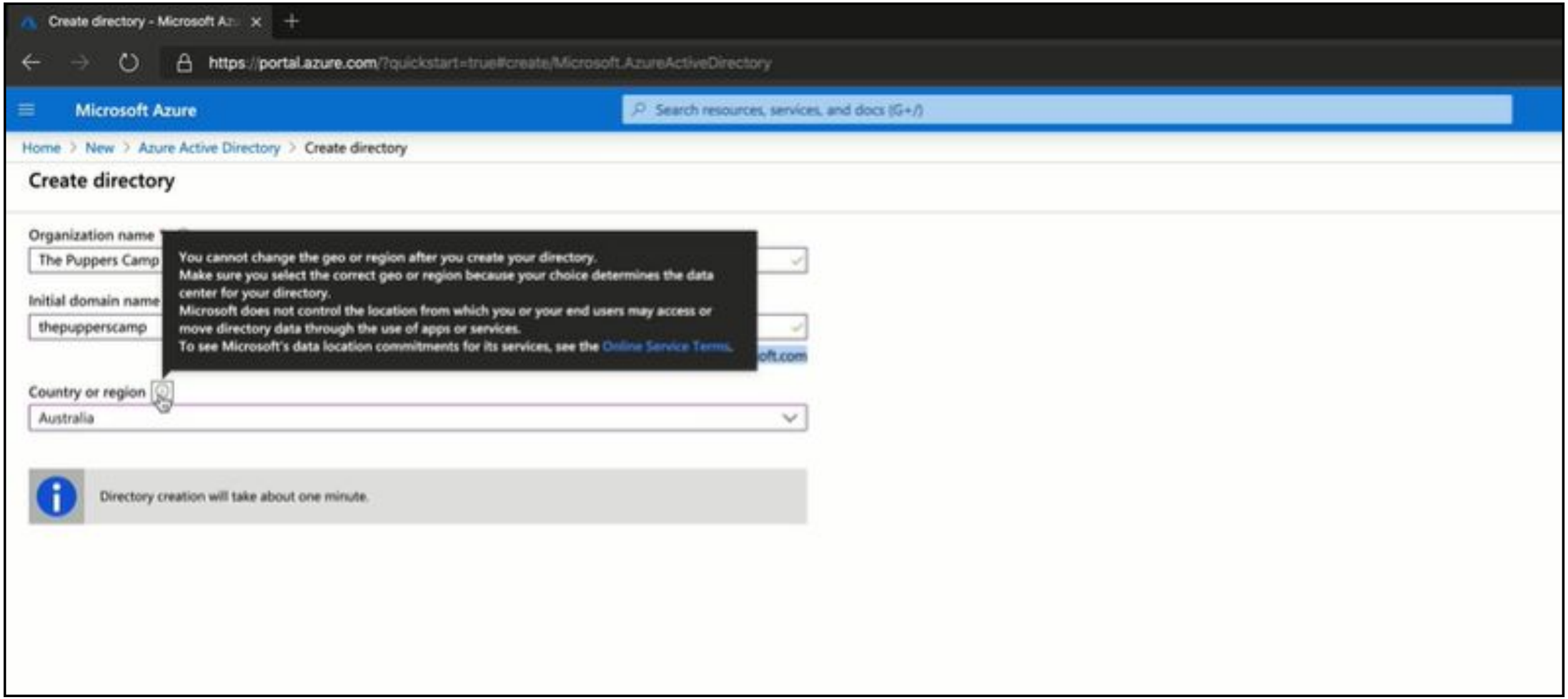
Each tenant will have **one** initial domain, and zero or more custom domains (added using **TXT** or **MX** DNS records).

Association

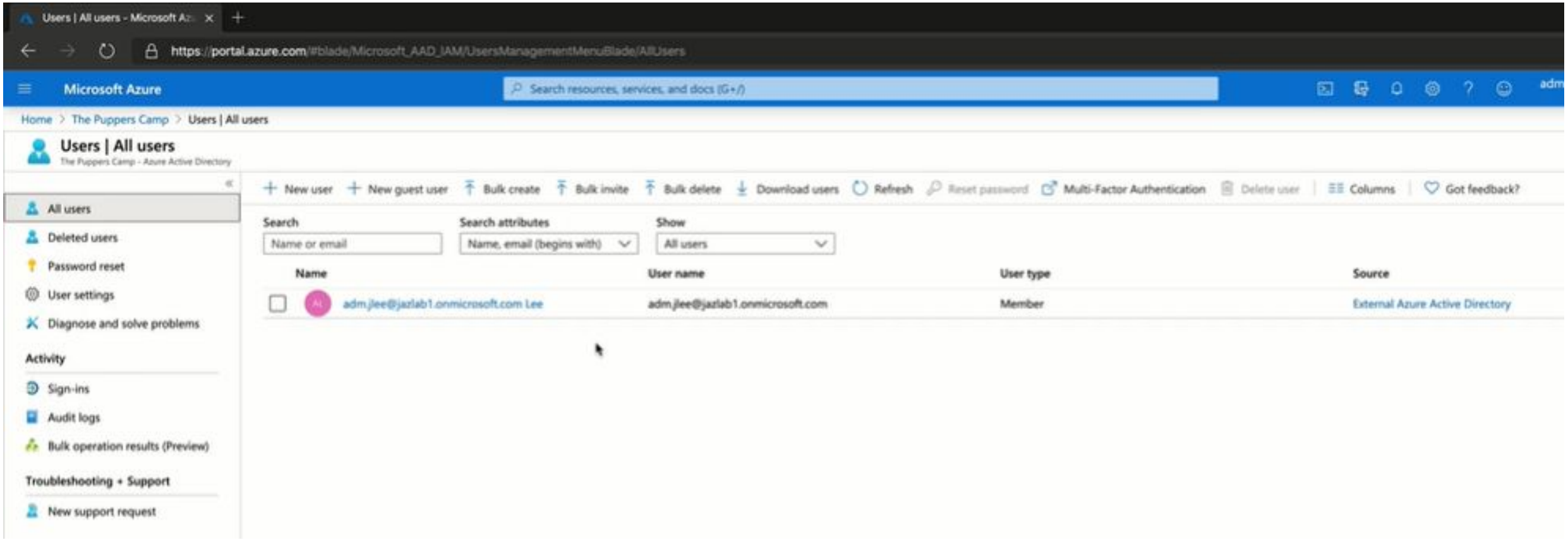
The **association/trust relationship** is a link between an Azure AD tenant and one (or more) Azure subscriptions.



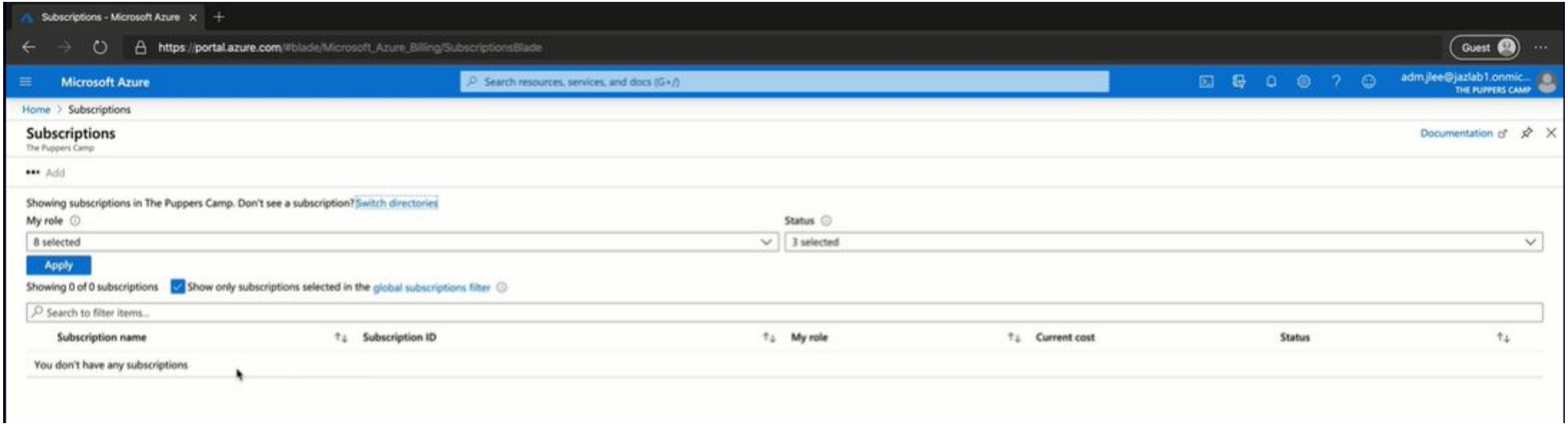
Important components



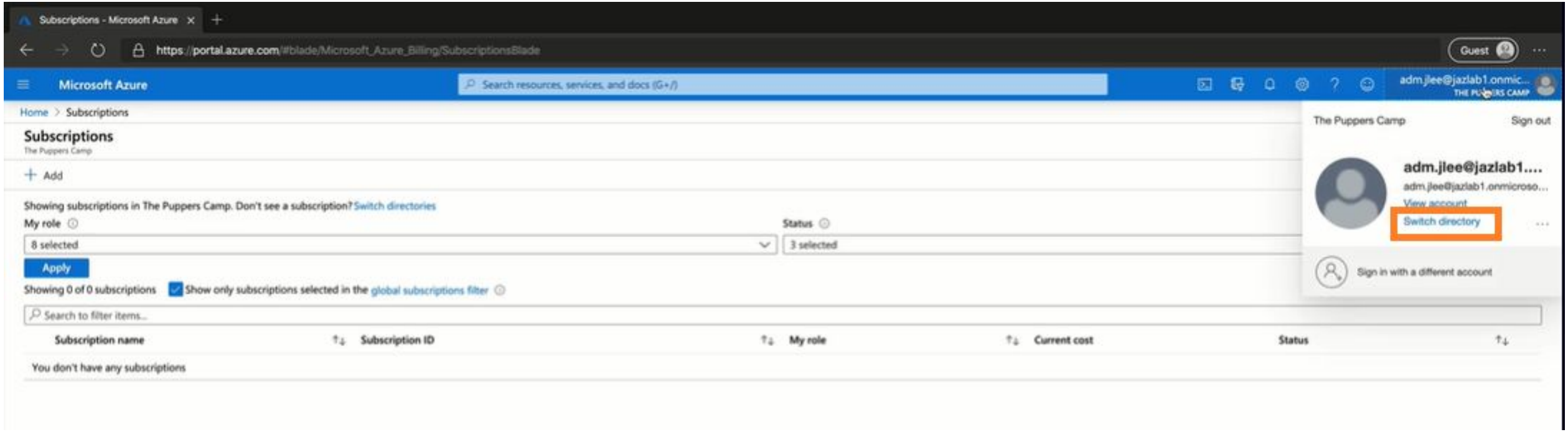
Create a new Directory



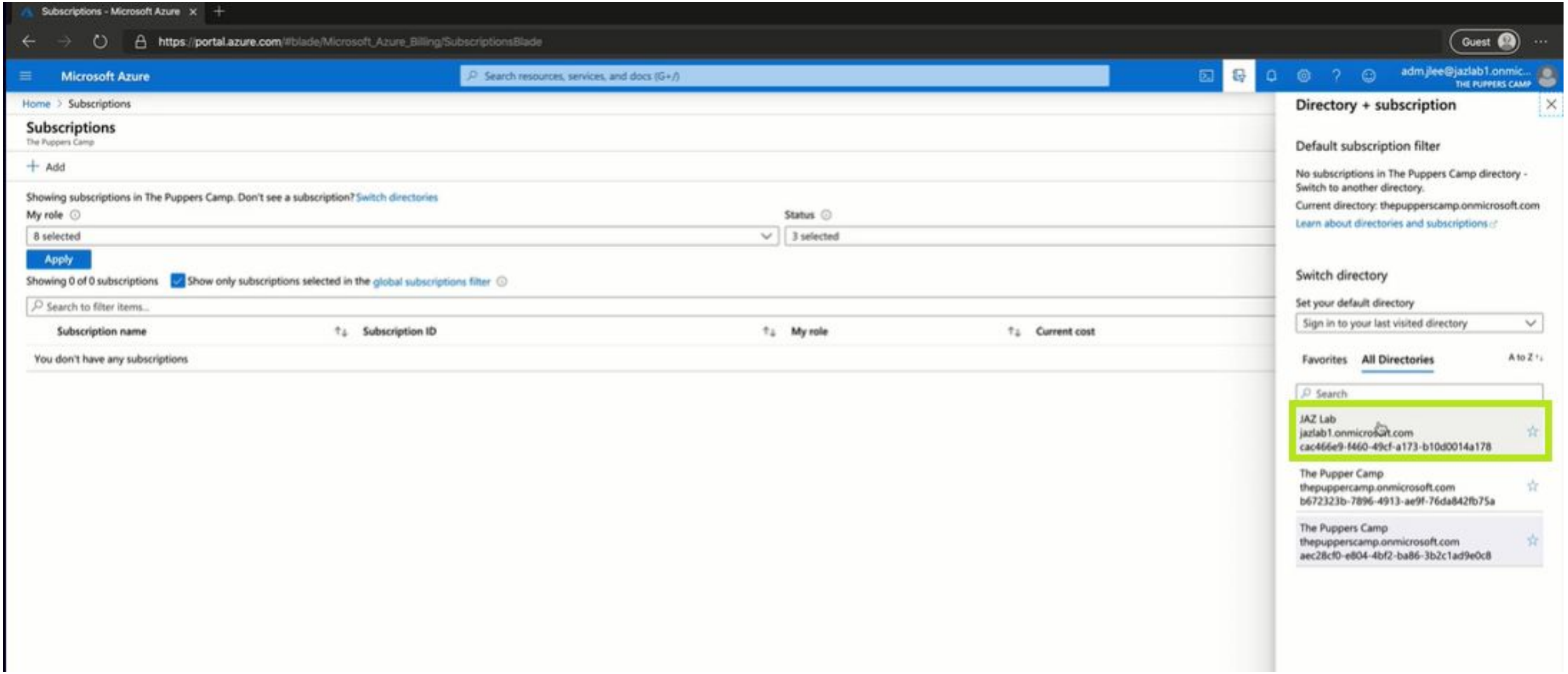
Create a new Directory



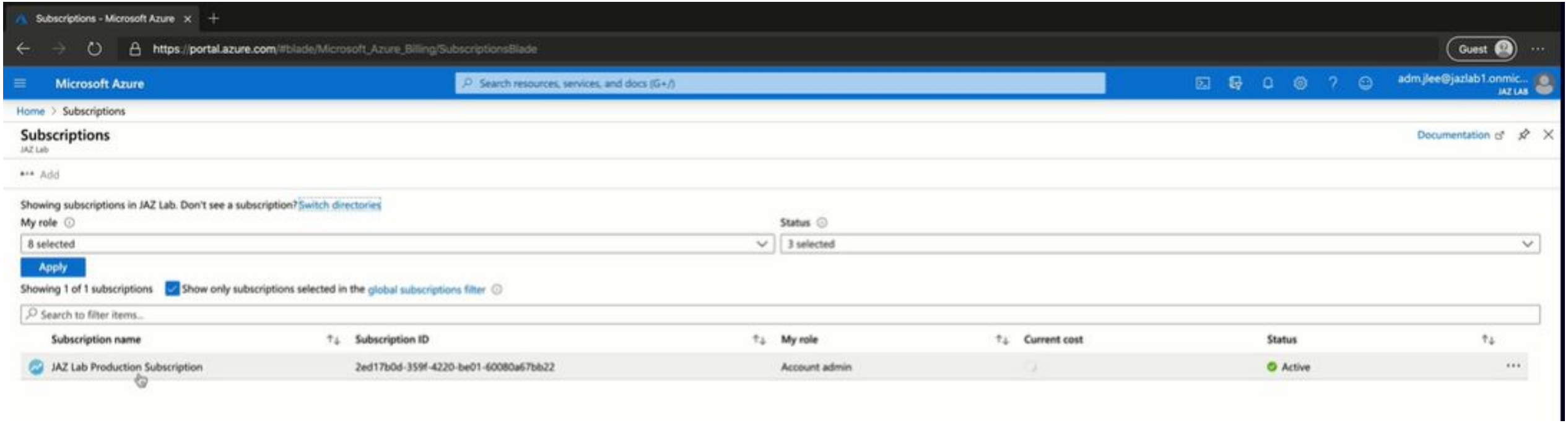
New tenant with no subscription



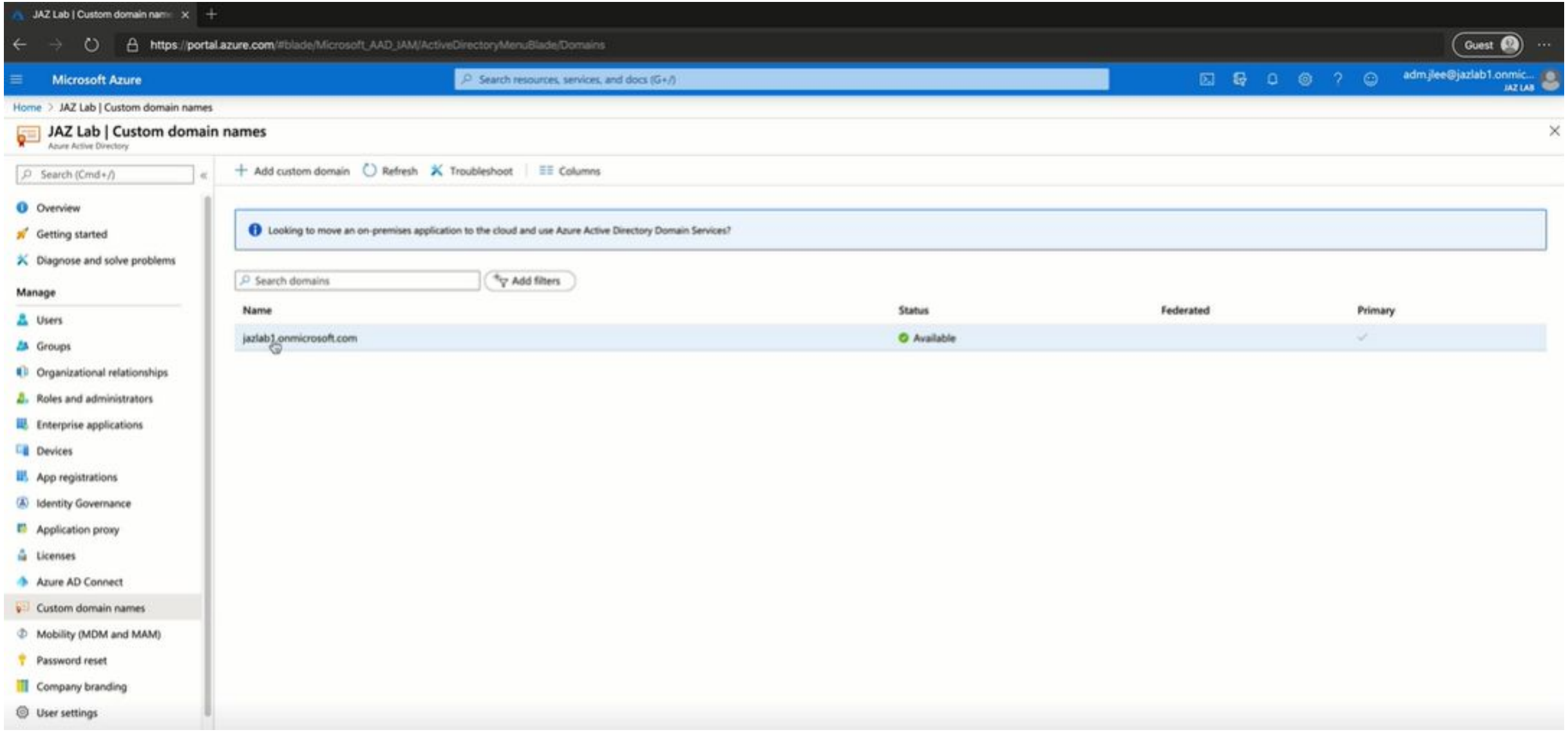
Switching Directory



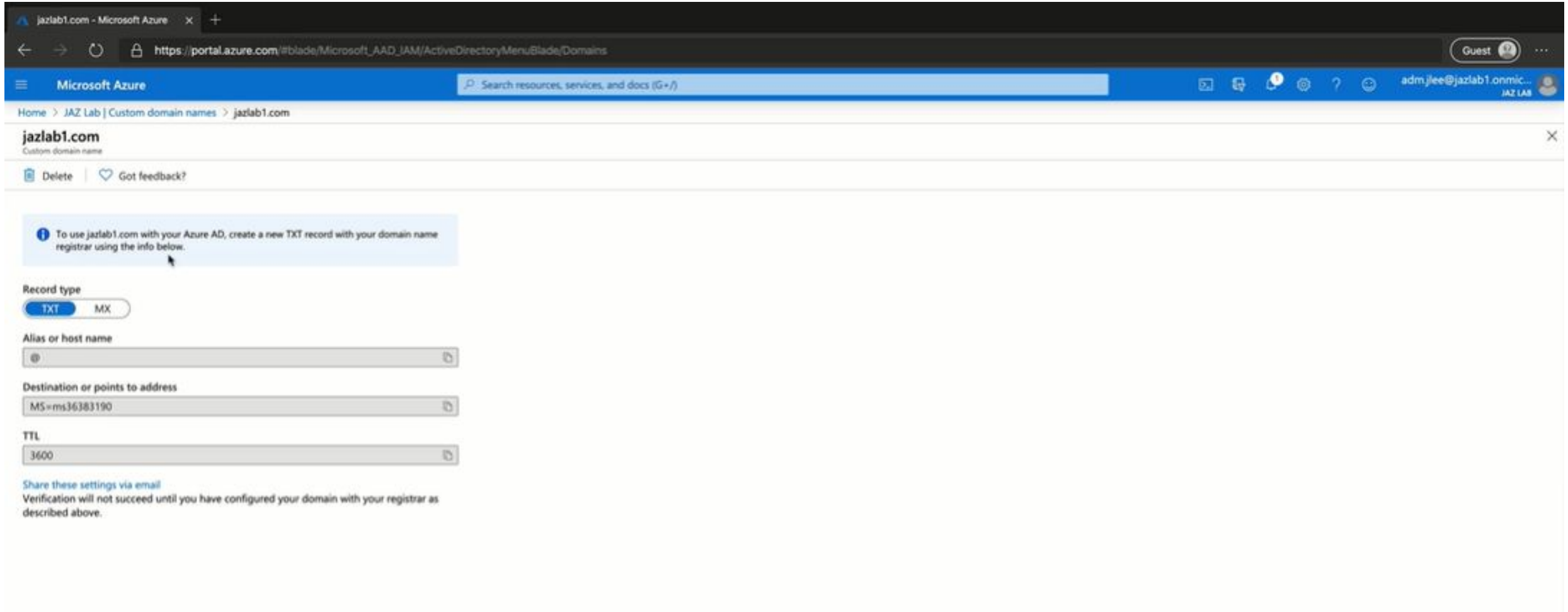
Switching Directory



Important components



Custom Domain name



Custom Domain name

Thank You