JON BONSO AND GEROME PAGATPATAN

AZURE CERTIFIED

AZ-104 Microsoft Azure Administrator

Tutorials Dojo Study Guide

TABLE OF CONTENTS

INTRODUCTION

With the rapid advancement of technology, enterprises are adopting newer technologies that will help their businesses transform and grow. Microsoft Azure is one of the emerging technologies that you can leverage in this age since a lot of companies are shifting their existing infrastructures in the cloud. Unlike the traditional setup, cloud computing allows you to obtain resources on-demand with just one click on their platform, including the servers, storage, databases, networking, analytics, artificial intelligence, and a lot more.

Microsoft Azure offers a range of cloud services, depending on your business needs. These services are continuously upgrading, and new features are being added every year to deliver customer satisfaction. Since Azure's resources and services are too vast, the **Microsoft Azure Certification** program offers different certification paths that will help aspiring candidates and IT professionals validate their skills and knowledge to maximize the solutions created in the cloud.

Microsoft Azure is the second biggest cloud service provider in the market next to AWS, and a lot of companies are now adopting a **multicloud** strategy, which makes it all the more beneficial for IT professionals like you to expand your skill set and learn multiple cloud technologies. Learning is a lot more fun if you merge it with various cloud services. It will be an exciting and enjoyable journey for you, and the first step is to become **AZ-104 Microsoft Azure Administrator** certified. This eBook will help familiarize yourself with the basic cloud concepts as well as the core services of Microsoft Azure, which are the building blocks that will help you pass the exam and make a successful career shift to cloud computing.

Note: We took extra care to come up with these study guides and cheat sheets, however, this is meant to be just a supplementary resource when preparing for the exam. We highly recommend working on **hands-on sessions** and **practice exams** to further expand your knowledge and improve your test-taking skills.

AZ-104 MICROSOFT AZURE ADMINISTRATOR EXAM OVERVIEW

The Microsoft Azure Certification Program validates the technical skills and knowledge for building secure and reliable cloud-based applications using the Azure platform. By successfully passing the Microsoft Azure exam, individuals can prove their expertise to their current and future employers.

Exam Details

The AZ-104 Microsoft Azure Administrator examination is intended for IT Professionals who implement, manage and monitor an organization's cloud infrastructure. You can take this exam from a local testing center or online from the comfort of your home. The exam is composed of different types of questions.

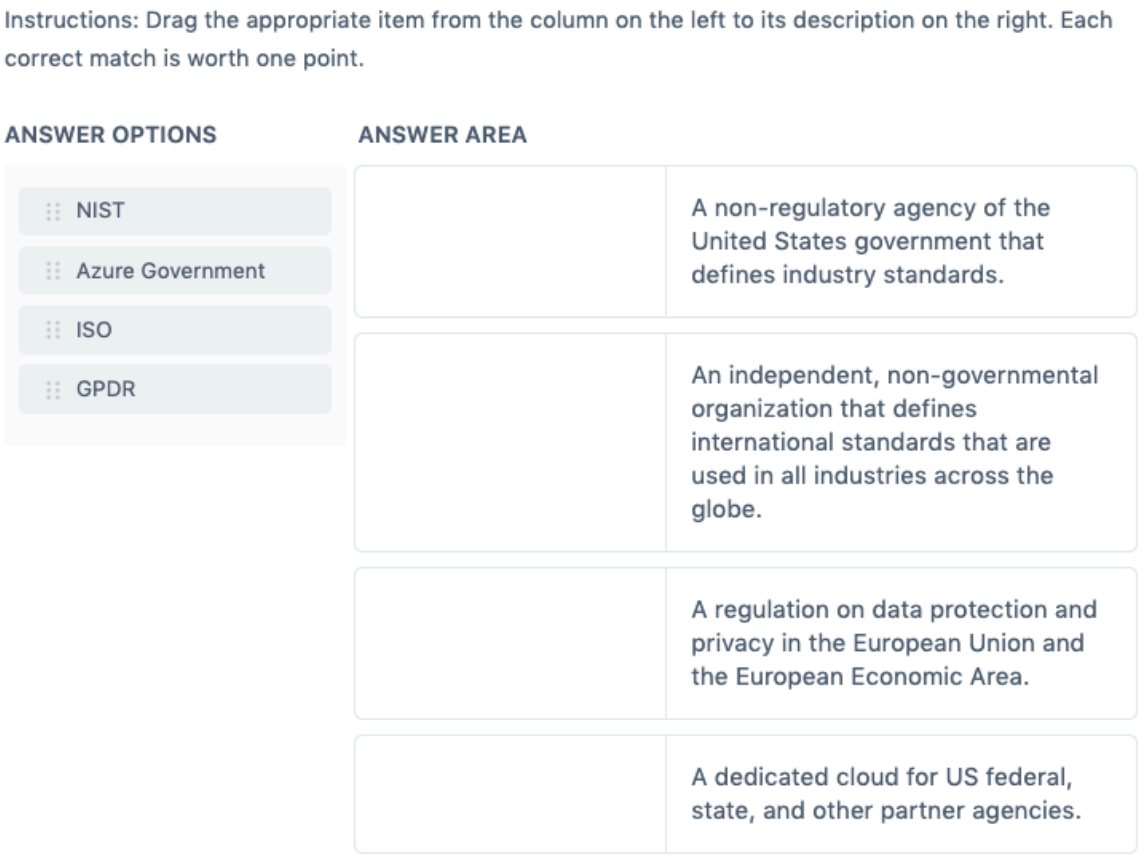
For multiple-choice types of questions, you will have to choose one correct response out of four options.

A company is planning to deploy its suite of enterprise applications to Microsoft Azure, where each application has several dependencies and subcomponents. The company must also control and manage the patching activities of the underlying operating system of the servers.

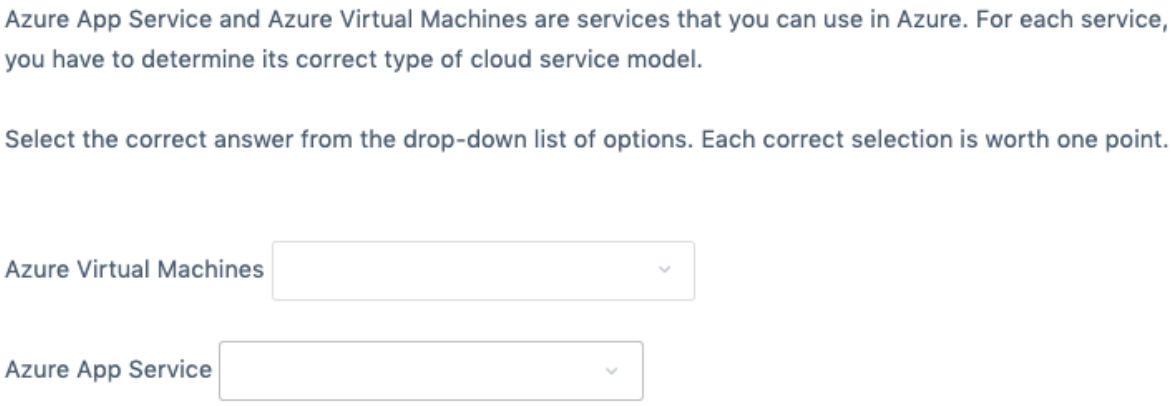
What type of cloud deployment solution should you recommend?

- ○ Infrastructure as a Service (IaaS)
- ○ Platform as a Service (PaaS)
- ○ Software as a Service (SaaS)
- ○ Functions as a service (FaaS)

For Drag and Drop questions, match the items by dragging them to their correct descriptions.



For Dropdown types of questions, select the correct answer from the drop-down list of options.



For Hotspot types of questions such as multiple Yes/No, evaluate whether the presented statements relating to a certain topic are correct/incorrect.

For each of the following items, choose **Yes** if the statement is true or choose **No** if the statement is false. Take note that each correct item is worth one point.

Questions	Yes	No
Azure virtual machines are billed on a per-hour basis.	○	○
When you delete a virtual machine in Azure, by default, any disks that are attached to the VM are deleted.	○	○
Disks attached to stopped virtual machines do not incur costs.	○	○

You can take the exam via online proctoring or from a testing center close to you.

Exam Code:	AZ-104
Prerequisites:	None
No. of Questions:	50-60
Score Range:	100-100
Cost:	0
Passing Score:	165 USD
Time Limit:	700 180 minutes

Exam Domains

The AZ-104 Microsoft Azure Administrator exam has five areas to assess your skills, each with a corresponding weight and topic coverage. The skills measured are: Manage Azure identities and governance (15–20%), Implement and manage storage (15–20%), Deploy and manage Azure compute resources (20–25%), Configure and manage virtual networking (20–25%), and Monitor and back up Azure resources (10–15%).

Manage Azure identities and governance

- Microsoft Entra ID objects
- Manage role-based access control (RBAC)
- Manage subscriptions and governance

Implement and manage storage

- Configure access to storage
- Manage data in Azure storage accounts
- Configure Azure Files and Azure Blob Storage

Deploy and manage Azure compute resources

- Automate deployment of resources by using Azure Resource Manager templates
- Create and configure virtual machines
- Create and configure containers
- Create and configure an Azure App Service

Configure and manage virtual networking

- Configure virtual networks
- Configure secure access to virtual networks
- Configure load balancing
- Monitor virtual networking

Monitor and backup Azure resources

- Monitor resources by using Azure Monitor
- Implement backup and recovery

Exam Scoring System

You can get a score from 100 to 1,000 with a minimum passing score of 700 when you take the AZ-104 Microsoft Azure Administrator exam. Microsoft uses a scaled scoring model to associate scores across multiple exam types that may have different levels of difficulty. Your complete score report will be sent to you by email 1 - 5 business days after your exam. However, as soon as you finish your exam, you'll immediately see a pass or fail notification on the testing screen.

For individuals who unfortunately do not pass their exams, you must wait 24 hours before you are allowed to retake the exam. There is no hard limit on the number of attempts you can retake an exam.

Once you receive your score report via email, the result should also be saved in your Microsoft Certification account. The score report contains a table of your performance in each domain and it will indicate whether you have met the level of competency required for these. Take note that you do not need to achieve competency in all areas for you to pass the exam. In the first part of the report, there will be a performance summary by exam section that highlights your strengths and weaknesses, which can help you determine the areas you need to improve on.

Exam Benefit

If you successfully pass any Microsoft Certification exam, you will receive a **Certified Digital Badge**. You can showcase your achievements to your colleagues and employers by adding these digital badges to your email signatures, Linkedin profile, or on your social media accounts. To view your badges, simply go to the “Dashboard” section of your Acclaim Account.

You can visit the official Microsoft Certification FAQ page to view the frequently asked questions about getting certified and other information about the Microsoft Certification: https://docs.microsoft.com/en-us/learn/certifications/certification-exam-policies.

AZ-104 MICROSOFT AZURE ADMINISTRATOR EXAM - STUDY GUIDE AND TIPS

The AZ-104 Microsoft Azure Administrator certification exam is designed for people who have experience in implementing, managing, and monitoring a Microsoft Azure environment. The exam will test your technical skills in implementing solutions based on different scenarios. Having prior experience in infrastructure management will help you understand the concepts and services easily.

The content of the exam will test your ability to perform the following:

- Manage Azure identities and governance
- Implement and manage storage
- Deploy and manage Azure compute resources
- Configure and manage virtual networking
- Monitor and backup Azure resources

For more information about the AZ-104 exam, you can check out this exam skills outline. This study guide will provide you with comprehensive review materials to help you pass the exam with flying colors.

Study Materials

For the Microsoft Azure Administrator exam, we recommend that you check out these study materials first before you take the actual exam. These resources will help you understand complex concepts and services that will be useful on your exam day.

1. **Microsoft Learn** – this website provides different learning paths for various Microsoft certifications. For the AZ-104 certification exam, you can focus on the following modules:
    - Prerequisites for Azure administrators
    - Manage identities and governance in Azure
    - Implement and manage storage in Azure
    - Deploy and manage Azure compute resources
    - Configure and manage virtual networks for Azure administrators

2. **Azure Documentation** – these documentations contain an overview, tutorials, examples, and how-to guides that will help broaden your knowledge on different Azure services.

3. **Azure Blog** – to get updated on new technologies and offerings of Microsoft Azure, you can subscribe to their newsletter.

4. **Azure FAQs** – you can find the FAQs section on the Azure documentation. The FAQs section is a compiled list of commonly asked questions, use cases, and a comparison of several Azure services.

5. **Azure free account** – the Azure portal will help you get hands-on experience with its 12-month trial. You'll also get free credits that you can spend for the first 30 days.

6. **Tutorials Dojo's Azure Cheat Sheets** – with the help of our cheat sheets, you can easily understand the information found in the Azure documentation. These are presented in bullet point format to highlight the important concepts.

7. **Tutorials Dojo's AZ-104 Microsoft Azure Administrator Practice Exams** – our practice exams have always been regarded as the best in the market. Each question in our practice tests contains detailed explanations at the end of each set to help you digest important concepts that will help you pass your Microsoft Azure certification exam on your first try.

Azure Services to Focus On

Your primary source of information when studying for the AZ-104 certification exam is the Azure documentation. To comprehend the different scenarios in the exam, you should have a thorough understanding of the following services:

1. **Azure Virtual Network** – you should know how to create a VNet peering, security rules, configuration of private/public IP addresses, network interface, subnets, and virtual networks.

2. **Azure DNS** – the configuration of custom DNS, private, and public DNS zone.

3. **Azure Application Gateway** – you should know when to use a load balancer and a web traffic load balancer, and how to create a web application firewall.

4. **Azure Load Balancer** – the types of load balancing rules, the difference between a public load balancer, and an internal load balancer.

5. **Azure VPN Gateway** – know how to configure VPN and VPN gateway.

6. **Azure ExpressRoute** – understand the concepts of ExpressRoute and how you would implement it in your environment.

7. **Azure Virtual Machines** – learn how to deploy and configure a VM, scale sets, highly available solutions, moving and redeploying of VM, creating a backup, backup policy, and recovery services vaults.

8. **Azure App Service** – learn how to create an app service plan and what run time can be put in the same app service plan.

9. **Azure Container Instances** – understand the concepts of containers and how to use ACI.

10. **Azure Kubernetes Service** – the difference between ACI and AKS, the configuration of AKS.

11. **Azure Blob** – you need to learn how to configure storage accounts, import/export of data, storage tiers, replication, and authentication.

12. **Azure Files** – learn how to create a file share, file sync, copy data using AZCopy.

13. **Microsoft Entra ID** – you should know how to manage a user, group, guest accounts, joined devices, device settings, and best practices.

14. **Azure RBAC** – learn how to create and assign a role and the types of built-in roles.

15. **Azure Policy** – you need to learn how to read and create a policy.

16. **Azure Monitor** – you should know how to interpret metrics, the configuration of log analytics, query and analyze logs, set up alerts and actions, and other service features.

We suggest that you check out **Tutorials Dojo's Azure Cheat Sheets**, which provide bullet-point summaries of the most important concepts on different Azure services.

Validate Your Knowledge

If you're feeling confident because you've followed the recommended materials above, it's time to test your knowledge of various Azure concepts and services. For high-quality practice exams, you can use the Tutorials Dojo **AZ-104 Microsoft Azure Administrator Associate practice exams**.

These practice tests cover the relevant topics that you can expect from the real exam. It also contains different types of questions such as single choice, multiple responses, hotspot, yes/no, drag and drop, and case studies. Every question on these practice exams has a detailed explanation and adequate reference links that help you understand why the correct answer is the most suitable solution. After you've taken the exams, it will highlight the areas that you need to improve on. Together with our cheat

sheets, we're confident that you'll be able to pass the exam and have a deeper understanding of how Azure works.

Sample Practice Test Questions:

Question 1

Your company has an Microsoft Entra tenant named tutorialsdojo.onmicrosoft.com and a public DNS zone for tutorialsdojo.com.

You added the custom domain name tutorialsdojo.com to Microsoft Entra ID. You need to verify that Azure can verify the domain name.

What DNS record type should you use?

1. SRV
2. NSEC
3. NSEC3
4. MX

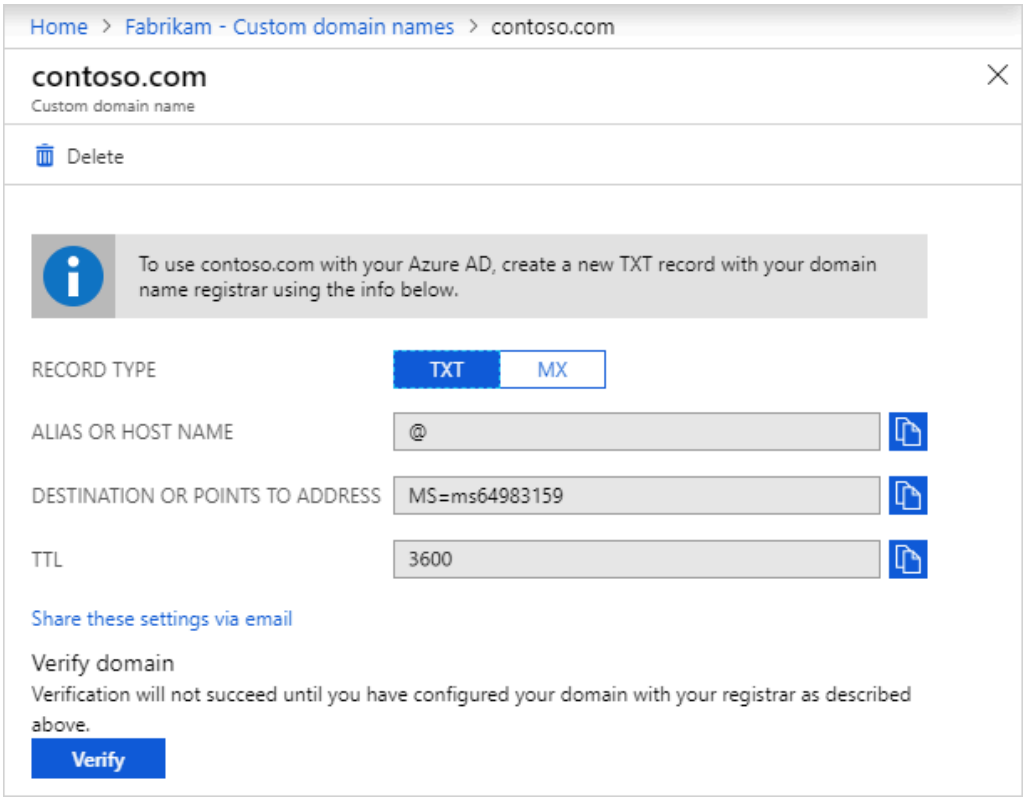
Correct Answer: 4

Microsoft Entra ID is Microsoft's cloud-based identity and access management service, which helps your employees sign in and access resources in:

– External resources, such as Microsoft Office 365, the Azure portal, and thousands of other SaaS applications.

– Internal resources, such as apps on your corporate network and intranet, along with any cloud apps developed by your own organization.

Microsoft Online business services, such as Office 365 or Microsoft Azure, require Microsoft Entra ID for sign-in and to help with identity protection. If you subscribe to any Microsoft Online business service, you automatically get Microsoft Entra ID with access to all the free features.



Every new Microsoft Entra ID tenant comes with an initial domain name, *<domainname>.onmicrosoft.com*. You can't change or delete the initial domain name, but you can add your organization's names. Adding custom domain names helps you to create user names that are familiar to your users, such as *azure@tutorialsdojo.com*.

You can verify your custom domain name by using TXT or MX record types.

Hence, the correct answer is: **MX.**

SRV, NSEC, and NSEC3 are incorrect because these record types are not supported by Microsoft Entra ID for verifying your custom domain. Only TXT and MX record types are supported.

References:

https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/add-custom-domain

https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-whatis

Check out this Microsoft Entra ID Cheat Sheet:

https://tutorialsdojo.com/microsoft-entra-id/

Question 2

You plan to automate the deployment of Windows Servers using a virtual machine scale set.

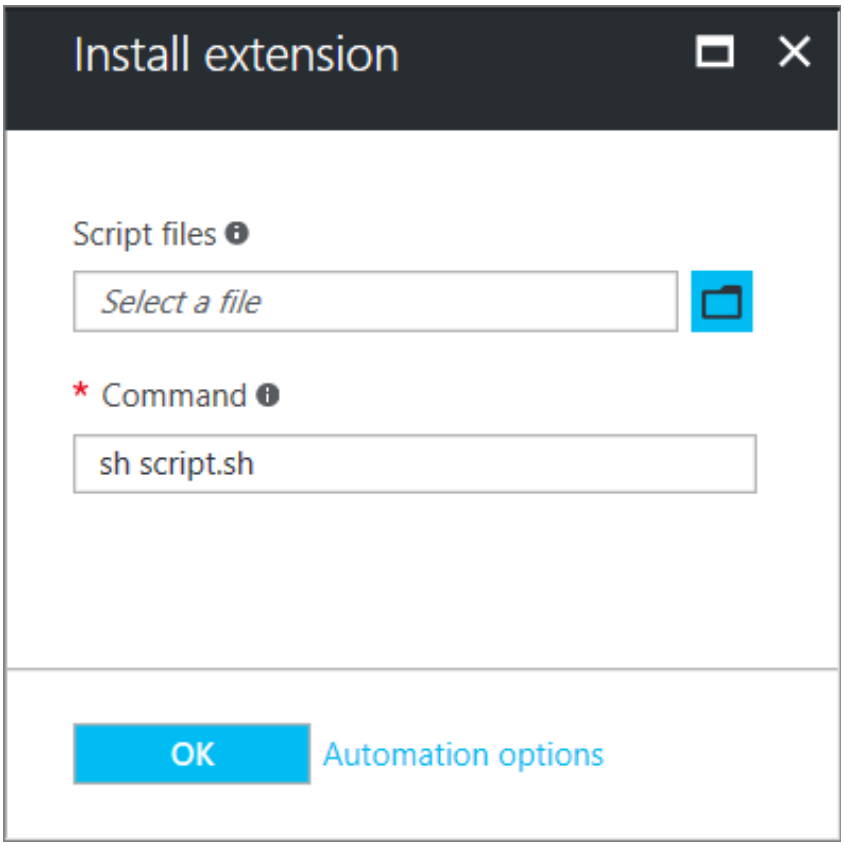
You need to make sure that the web components are installed in the virtual machines.

Which two actions should you perform?

1. Create a configuration script.
2. Create an automation account.
3. Create a policy.
4. Configure the extensionProfile section of the ARM template.
5. Create a new scale set.

Correct Answer: 1, 4

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



The Custom Script Extension downloads and executes scripts on Azure virtual machines. This extension is useful for post-deployment configuration, software installation, or any other configuration or management tasks.

Hence, the correct answers are:

– Create a configuration script.

– Configure the extensionProfile section of the ARM template.

The option that says: **Create an automation account** is incorrect because an automation account wouldn't help you automatically install web components. You still need to create a configuration script and extensionProfile in the ARM template.

The option that says: **Create a policy** is incorrect because this option only evaluates resources in Azure. Take note that you don't need to create a policy to install web components.

The option that says: **Create a new scale set** is incorrect because this wouldn't install the required web components. Instead of creating a new scale set, you should use a custom script extension to install the web components in the VMs.

References:

https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/virtual-machine-scale-sets-deploy-app

https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/tutorial-install-apps-template#what-is-the-azure-custom-script-extension

https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/virtual-machine-scale-sets-deploy-app#already-provisioned

Check out this Azure Virtual Machines Cheat Sheet:

https://tutorialsdojo.com/azure-virtual-machines/

For more **AZ-104 practice exam** questions with detailed explanations, check out the **Tutorials Dojo Portal**:

AZ-900 Microsoft Azure Fundamentals Practice Exams

AZ-104 Microsoft Azure Administrator Practice Exams

Final Remarks

It is not enough to understand the concepts at a high level. You also need to get hands-on experience by using the Microsoft Azure Portal. Simulate different scenarios that will help you deepen your understanding of various services. The combination of practical and theoretical knowledge will help you analyze difficult questions in the exam.

A few reminders that we can give is to always check the time and review your answers before proceeding to the next question (especially in the case study and yes/no questions). Before your scheduled exam day, don't forget to take a good rest. If you're not feeling confident yet, there's always an option to reschedule your exam. Good luck, and we wish you all the best.

CLOUD COMPUTING CONCEPTS

Cloud computing is the delivery of services over the Internet that helps you reduce your operating costs, run your infrastructure efficiently, and scale as business requirements change.

Cloud Service Models

- The three cloud computing service models are IaaS, PaaS, and SaaS.
- You can also use serverless computing to eliminate the need to manage infrastructure.
- The shared responsibility model determines the security tasks that are handled by the cloud provider and handled by the customer.
    - Azure is responsible for protecting the infrastructure such as hosts, network, and data center.
    - The customer is responsible for protecting their data, endpoints, account, and access management.
- IaaS, PaaS, and SaaS have different levels of managed services:

IaaS	PaaS	SaaS
Applications	Applications	Applications
Data	Data	Data
Runtime	Runtime	Runtime
MIddleware	MIddleware	MIddleware
O/S	O/S	O/S
Virtualization	Virtualization	Virtualization
Servers	Servers	Servers
Storage	Storage	Storage
Networking	Networking	Networking

You Manage Vendor Manages

Infrastructure as a service (IaaS)

- Most user management
- You are responsible for managing the **operating systems, data,** and **applications.**
- IaaS helps you to extend resources rapidly to meet the spikes required for your application.
- Used in the following scenarios:
    - **Migrating workloads** – move existing applications to the cloud.
    - **Test and development** – quickly set up and dismantle test and development environments. IaaS makes scaling development and testing environments fast and economical.
    - **Storage, backup, and recovery** – simplify the planning and management of backup and recovery systems.
    - **Website hosting** – less expensive than traditional web hosting.
    - **High-performance computing (HPC)** – clusters of computers that help solve complex problems involving millions of variables or calculations.
    - **Big data analysis** – for massive data sets that require a huge amount of processing power.

Platform as a service (PaaS)

- Less user management
- The operating systems are managed by the cloud provider, while the user is responsible for the applications and data they run and store.
- PaaS offers all the functionality you need to support the entire lifecycle of web applications: **building, testing the application**, **deploying the source code**, **managing**, and **updating** within the same integrated environment.
- Used in the following scenarios:
    - **Development framework** – a framework for creating or customizing cloud-based applications.
    - **Analytics or business intelligence** – find insights and patterns, and predict outcomes to improve business decisions.

Software as a service (SaaS)

- Least amount of management
- The cloud provider is responsible for managing everything, and the end-user just uses the software.

Serverless Computing

- Function as a Service (FaaS)
- You simply deploy the code with a serverless platform, and it runs at high availability.
- Dynamically scales up and down to meet the demands of each workload within seconds.

- A **pay-per-execution model** that charges sub-second billing only for the time and resources required to execute the code.

Cloud Architecture Models

- Three deployment methods of cloud computing: **Public vs Private vs Hybrid.**
- The model you choose for cloud deployment depends on your budget, security, scalability, and maintenance needs.

Public Cloud

- Focus on maintaining your applications without having to worry about purchasing, managing, or maintaining the hardware on which it runs.
- You can use multiple public cloud providers of varying scale.

Advantages	Disadvantages
High scalability/agility.	Specific security requirements.
Pay-as-you-go pricing.	Government policies, industry standards, or legal requirements.
You are not responsible for the updates and maintenance of the hardware.	You don't own the hardware or services and you also can't manage them as you may want to.
The required technical knowledge is minimal.	Maintaining a legacy application might be hard to meet.

Private Cloud

- A dedicated on-premises datacenter configured to be a cloud environment that provides users in your organization with self-service access to compute resources.
- You are responsible for the purchase and maintenance of the hardware and software services.
- You can use a private cloud when an organization has data that cannot be put in the public cloud, perhaps for legal reasons.

Advantages	Disadvantages
Any scenario or legacy application configuration is supported.	CapEx involved – principal cost is the procurement of the equipment.
You have control (and responsibility) over security.	To scale, you must buy, install, and set up new hardware.

Compliance or security requirements in your organization.	Private clouds require IT skills and expertise.

Hybrid Cloud

- Data and applications can move between **private** and **public clouds**.
- When there is a spike in demand in your private cloud, you can "burst through" to the public cloud for additional computing resources.

Advantages	Disadvantages
Maintain a private infrastructure for sensitive assets.	More expensive than selecting one deployment model since it involves some CapEx cost upfront
Take advantage of the resources in the public cloud when needed.	It can be more complicated to set up and manage
With the ability to scale to the public cloud, you pay for extra computing power only when needed.	
Allows you to use your own equipment to meet the security and compliance requirements in your organization.	

AZURE BASICS

Azure Overview

Azure is a cloud computing platform that was introduced by Microsoft in 2010. It enables you to create, manage, and deploy applications across a large global network. Microsoft Azure also provides a variety of services to assist your business in addressing current and potential business challenges in your infrastructure and applications.

Today, Microsoft Azure has the second-largest share in the cloud industry. It also has specialized regions for compliance or legal purposes.

Advantages of Azure Cloud Computing

- **Cost** – Eliminate the capital expense of buying hardware, software, and setting up of data centers. The principle of the cloud is, you will only pay for the computing resources you have consumed.
- **Global scale** – One of the benefits of cloud computing is the ability to scale elastically. This means that you can easily add resources such as compute and storage capacity in different regions with just a few clicks.
- **Performance** – Cloud computing services are hosted on a global network of secure data centers that are upgraded with the latest generation of computing hardware on a regular basis. Compared to a single corporate datacenter, this has several advantages, including lower application network latency and greater economies of scale.
- **Security** – Cloud service providers offer a broad set of policies, technologies, and controls to protect your data and infrastructure against potential threats.
- **Speed** – In a cloud computing environment, you can provision computing resources in minutes with just a few clicks. Providing businesses with a great deal of flexibility and relieving capacity planning pressure.
- **Productivity** – The cloud provides a lot of convenience to your IT teams since it reduces the time needed to obtain additional resources, allowing them to focus solely on achieving more important business goals.
- **Reliability** – With cloud computing, you can easily manage backup data, disaster recovery and business continuity since the data can be mirrored at multiple redundant sites.

Azure Global Infrastructure

Regions

- Each region has more than one data center, which is a physical location.
- A group of data centers deployed in a latency-defined perimeter and connected through a dedicated regional low-latency network.
- Criteria in choosing a Region:
    - **Location** – a region closest to your users minimizes the latency.
    - **Features** – some features are not available in all regions.
    - **Price** – the price of services varies from region to region.
- Each Region is paired within the same geographic area.
- If the primary region has an outage, you can **failover** to the secondary region.
- You can use paired regions for **replication.**
- Regions that are unique when it comes to compliance:
    - **Azure Government Cloud** – only US federal, state, local, and tribal governments and their partners have access to this dedicated instance.
    - **China Region** – data center is physically located within China and has no connection outside of China, including other Azure regions.

Availability Zones

- Each availability zone is a physical location within a region.
- A zone is composed of one or more data centers with independent power, cooling, and networking facilities.
- Azure services that support Availability Zones fall into two categories:
    - **Zonal services** – a resource is pinned to a specific zone.
    - **Zone-redundant services** – replicates automatically across zones.

Azure Security and Compliance

In the cloud, the responsibility of security is a shared one. Microsoft Azure secures what they can on their end, while you secure what you can on your end. Only this way can everyone protect their valuable data. Also as a customer, you inherit all the best practices of Azure policies, architecture, and operational processes built to satisfy the requirements of their most security-sensitive customers.

Microsoft Azure has also developed multiple tools and services to help you achieve your security objectives. You can also review the numerous audits and certifications that third-party auditors have conducted on Azure so that whenever you need to fulfill strict compliance with the use of a service, you can simply verify its status through the catalog.

Azure Pricing

- Azure offers pay-as-you-go and reserved instances for pricing.
- Azure Pricing Factors:
    - Resource size and resource type.
    - Different Azure locations have different prices for services.
    - The bandwidth of your services.
    - Any data transfer between two different billing zones is charged.
        - **Ingress (data in)** = free
        - **Egress (data out)** = charged based on data going out of Azure datacenters.
- Factors that can reduce costs:
    - By purchasing a **reserved instance** (one-year or three-year terms), you can significantly reduce costs by up to 72 percent compared to pay-as-you-go pricing.
    - A **reserved capacity** is a commitment for a period of one or three years for SQL Database and SQL Managed Instance.
    - **Hybrid Benefit** allows you to use your on-premises Software Assurance-enabled Windows Server and SQL Server licenses on Azure.
    - If you purchase an unused compute capacity, you can get deep discounts up to 90 percent compared to pay-as-you-go pricing. **A spot virtual machine** is for workloads that can tolerate interruptions.
- All resources belong to a **subscription**.
    - An Azure account can have multiple subscriptions.
    - Organize your resources and subscriptions using **Azure management groups**.
- **Azure Cost Management** gives you a detailed view of current and projected costs.
- For new accounts, the **Azure Free Tier** is available.
    - Free Tier offers limited usage of Azure products at no charge for 12 months.
    - You also get $200 credit that you can spend during the first 30 days.
    - More details at https://azure.microsoft.com/en-us/free/

  - Estimate your expected monthly costs using **Azure Pricing Calculator.**
- **Total Cost of Ownership (TCO) Calculator**
  - Estimate total savings over a period of time by using Azure.
  - Compares costs and savings against on-premises and co-location environments.
- **Azure Support Plans:**
  - **Basic** – included for all Azure customers.
  - **Developer** – recommended for non-production environments. Limited access to technical support during business hours by email only.
  - **Standard** – appropriate for production workload environments. Has 24/7 access to Azure's technical support engineers by phone or email.
  - **Professional Direct** – suitable for business-critical workloads. Has 24/7 access to Azure's technical support engineers by phone or email. Provides access to Operations Support, ProDirect delivery managers, and Support APIs.

Service Level Agreement (SLA)

- It is the commitment of Microsoft for the uptime and connectivity of a service.
- You could obtain a service credit if the service level agreement is not met by Microsoft.
- Composite SLAs include several resources (*with different availability levels*) to support an application.
- SLAs for multi-region deployments distribute the application in more than one region for high availability and use Azure Traffic Manager for failover if one region fails.

Service Lifecycle

- **Private Preview** is only available to a few customers for early access to new technologies and features.
- **Public Preview** makes the service in the public phase and can be used by any customers to evaluate the new features but SLA does not apply.
- **General Availability** is the release of service to the general public and is fully supported by SLAs.
- Azure updates allow you to get the latest updates on any Azure products and features.

Azure Well-Architected Framework - Five Pillars

- **Operational Excellence** - run and monitor systems to deliver business value and to continually improve supporting processes and procedures.
- **Reliability** - recover the system from infrastructure or service disruptions, dynamically acquire computing resources to meet demand, and mitigate disruptions such as misconfigurations or transient network issues.
- **Performance Efficiency** - use computing resources efficiently to meet system requirements and to maintain that efficiency as demand changes and technologies evolve.
- **Cost Optimization** - avoid or eliminate unneeded costs or suboptimal resources.
- **Security** - protect information, systems, and assets while delivering business value through risk assessments and mitigation strategies.

Best Practices when Architecting in the Cloud

- **Design for self healing** - Failures occur in a distributed system. Design your application to be self-healing in the event of failure.
- **Make all things redundant** - Design a resilient and highly available application to avoid single points of failure.
- **Minimize coordination** - To achieve scalability, you must minimize coordination between application services.
- **Design to scale out** - Design an application that can scale horizontally (adding or removing new instances) as needed.
- **Partition around limits** - Use partition for database, network, and compute limits
- **Design for operations** - The operations team must be able to access the tools they need for the application.
- **Use managed services** - When designing an application, use PaaS rather than IaaS.
- **Use the best data store for the job** - Select the storage technology that is most appropriate for your data and its intended use.
- **Design for evolution** - An evolutionary design is required for continuous innovation.
- **Build for the needs of business**- Always consider the business requirements when designing an application.

THE DIFFERENT AZURE SERVICES

Compute is the processing power required by applications and systems to carry out computational tasks.
Services: Virtual Machine, App Service, Functions, and Kubernetes Service

Storage in the cloud is used to store different types of data, such as objects, files, and backups.
Services: Blob, Disk, and Files

Database is a system to store and manage structured and unstructured information.
Services: SQL Database and Cosmos DB

Networking provides a global link to distribute the application all over the world.
Services: Virtual Network, Load Balancer, CDN, and DNS

Security allows you to authenticate and authorize users and services to access your applications.
Services: Active Directory, RBAC, and Security Center

Management and Governance is a tool to control and monitor your infrastructure services.
Services: Monitor, Policy, and Advisor

DEEP DIVE

Azure Virtual Machines

Components of a Virtual Machine

1. When creating a virtual machine, you always start off by choosing a **subscription** and **resource group**. A subscription is a container where you can provision Azure resources. Before you can deploy resources, you also need to create a new resource group. This is a logical group to organize and manage all your resources in your subscription.

2. After you have chosen the resource group, you configure the **availability option** of your virtual machine. You can choose between the availability zone, availability set or no infrastructure redundancy option. The option you selected here would determine the availability and resiliency of your applications.

3. The **image** of your virtual machine contains the OS, settings, and other applications that you will use in your server. In the Azure Marketplace, you can choose between images provided by Microsoft or your own custom image

4. Once you have chosen the image of your virtual machine, select the **type and size** of your virtual machine. This will determine the physical properties of your instance, such as vCPUs, RAM, disks, and more.

5. During the creation of your virtual machine, you can also specify whether you'd like to launch it in a **spot instance** or use another instance billing type (pay as you go or reserved).

6. To access your virtual machine, you will need to use a **key pair**. It is generated after you launch your virtual machine. Make sure to secure your copy of your public key. Once you delete your public key, you wouldn't be able to directly access your instance.

7. After you have configured the basic settings, you need to add **storage** for your virtual machine. The disks that can be added are the operating system disk, data disk, and temporary disk. Encryption for your disks is automatically configured.

8. You also need to configure which **virtual network** the virtual machine should be launched in. And the **network security group** will serve as a firewall to your servers. It contains rules that allow or deny network traffic coming to or from your firewall.

9. When you have configured the network settings of your virtual machines, you can also enable **monitoring, auto-shutdown, and backup** in the management options.

10. In the advanced configuration option of your virtual machine, you can add **extensions** for post-deployment configuration, **custom and user data** to execute certain commands while the instance is being provisioned, and **proximity placement** group to enable you to group your resources closer in the same region.

11. Lastly, you can add **tags** to easily identify and classify your resources.

12. Once you have reviewed the configuration of your instance, proceed with the launch. Wait for your virtual machine to finish preparing itself, and you should be able to connect to it if there aren't any issues.

13. If you are having difficulties connecting to a virtual machine, you can try redeploying the VM to move it to a new node in the Azure infrastructure. Don't worry, all of the existing configurations in the resource will still be there after completing the redeployment.

Types of Virtual Machines

1. **General Purpose** - provides a balanced CPU-to-memory ratio. This instance is ideal for testing, development, and low to medium-traffic web servers. The B-series have burstable performance that allows the VM to use the build-up credits when the application requires higher CPU performance.

2. **Compute Optimized** - designed to have a high CPU-to-memory ratio. Instances belonging to this family are well suited for medium-traffic web servers, network appliances, batch processes, analytics, application, and gaming servers.

3. **Memory Optimized** - offers a high memory-to-CPU ratio. Ideal for relational database servers, medium to large caches, and in-memory analytics.

4. **Storage Optimized** - provides high disk throughput and IO. This VM size is ideal for SQL, NoSQL databases, big data, data warehousing, and large transactional databases.

5. **GPU** - designed for compute-intensive, graphics-intensive, and visualization workloads. It is available in single, multiple, or fractional GPUs.

6. **High-performance compute** - the HPC VM size is the most powerful and fastest CPU with high throughput network interfaces. It is optimized for fluid dynamics, explicit and implicit finite element analysis, weather modeling, seismic processing, reservoir simulation, and RTL simulation.

Virtual Machine Disks

The disks of a virtual machine are block-level storage volumes. This storage is managed by Azure and mainly used for Azure VMs. With managed disks, all you have to do is specify the type and size of the disk and provision it.

In Azure, there are three types of disk roles:

1. **Operating system (OS) disk** - in order for the virtual machine to operate, it must have an OS disk. There are a variety of images that you can choose from in the Azure Marketplace. An example of images that you can use are Windows Server, Ubuntu, Debian, RHEL, etc. There are two types of OS disk:
    a. Persistent OS disk - this type of disk supports all sizes of VM, and the data is preserved even if you upgrade your OS disk and VM size.
    b. Ephemeral OS disk - use ephemeral OS disks if you need lower read/write latency and faster VM reimage. This type of disk is ideal for stateless applications, and it can be stored on VM cache or VM temp/resource disk if sufficient space is available.

2. **Data disk** - this disk is also managed by Azure, and you can store your application data or any other data that you need to keep. Before you use a data disk, there are two options that you can select:
    a. Create and attach a new disk - you have the option to create the new disk from a snapshot, storage blob, or an empty disk.
    b. Attach an existing disk - allows you to add the disks you've already created. It's also important to know that the number of data disks that you can attach will depend on the size of your VM.

3. **Temporary disk** - provides you short-term storage to store pages and swap files. Take note that the data on this disk may be lost when you redeploy a VM or during a maintenance event. Also, to configure a server-side encryption on this disk, you need to enable encryption at host.

The available disks that you can choose from are:

1. **Ultra Disk** - ideal for IO-intensive workloads, top-tier databases, and other transaction-heavy workloads. This storage has the highest disk size, throughput, and IOPS.
2. **Premium SSD** - designed for production and performance-sensitive workloads.
3. **Standard SSD** - used for web servers and dev/test environments.
4. **Standard HDD** - ideal for backup, non-critical data, and infrequent access.

Detail	Standard HDD	Standard SSD	Premium SSD	Ultra Disk
Disk type	HDD	SSD	SSD	SSD
Scenario	Backup, non-critical, infrequent access	Web servers, and light applications of enterprise	Production and performance sensitive workloads	IO-intensive workloads, top tier databases, and other transaction-heavy workloads
Max Disk Size	32,767 GiB	32,767 GiB	32,767 GiB	65,536 GiB
Max Throughput	500 MB/s	750 MB/s	900 MB/s	2,000 MB/s
Max IOPS	2,000	6,000	20,000	160,000

It's also very important to understand how you can secure your data inside your virtual machine disks. Let's now take a look at disk encryption, Azure managed disks supports three types of encryption:

1. **Server Side Encryption (SSE)** - the data stored on managed disks are automatically encrypted at rest by default when persisting it to the cloud.
    a. Platform-managed keys - the keys are managed by Azure. The data, images, and snapshots written to an existing managed disks are automatically encrypted-at-rest.
    b. Customer-managed keys - since you are providing your own keys, you also manage the level of encryption on each managed disk. To manage your own keys, you can use Azure Key Vault. This service enables you to import your own RSA keys or generate a new ones.

2. **Azure Disk Encryption (ADE)** - provides volume encryption on both OS and data disks of Azure VMs. The encryption for Windows is done using BitLocker. On the other hand, the encryption for Linux is done using DM-Crypt.

3. **Encryption at host** - this type of encryption is different from SSE. The encryption of data is provided by the server hosting your virtual machine and the encrypted data flows into the Azure Storage service.

	Encryption at rest	Temp Disk Encryption	Encryption of Caches	Encrypted Data Flows	Customer Keys	Encryption Status
Encryption at rest with PMK	✓	-	-	-	-	Unhealthy
Encryption at rest with CMK	✓	-	-	-	✓	Unhealthy
Azure Disk Encryption	✓	✓	✓	✓	✓	Healthy
Encryption at Host	✓	✓	✓	✓	✓	Unhealthy

Note:
- The encrypted data flows are between Compute and Storage service.
- The disk encryption status is labeled by Azure Security Center.

When creating a copy of your managed disks, there are comparisons between images and snapshots. As discussed earlier in data disks, snapshots allow you to create a point in time recovery. But how is it different from images?

Let's look at the differences between the two:
1. **Snapshots** - a full, read-only copy of your virtual hard drive. It can be taken at any point in time. The existence of a managed disk snapshot is independent of the source disk. This means that it applies only to one disk. You can also use snapshots to create a new disk and attach it to a virtual machine.
2. **Images** - contain all the managed disks associated with the virtual machine. The created image can be used to launch hundreds of virtual machines without managing any storage accounts.

To conclude the comparison, a snapshot is only aware of the disk that it contains. For scenarios that require the coordination of multiple disks, like striping, snapshot wouldn't be able to meet this requirement. Therefore, this is where you would want to use custom images.

When talking about how the virtual machine handles unexpected disk traffic, Azure offers a feature called **bursting**. This will grant the virtual machine and disk the ability to boost the IOPS and MB/s performance for a period of time. In other words, it will allow you to get more use out of your disk and also helps you avoid upgrading the disk just to accommodate traffic spikes. The bursting on virtual machines and disks are independent from one another. So if you need to burst the disk performance, you don't need to burst the virtual machine. Bursting is enabled by default for both virtual machine and disk.

The following resources support bursting:

1. **Burstable Virtual Machines:**
    a. **General Purpose:** B, Dsv3, Dasv4, Ddsv4, and Dsv4 series
    b. **Compute Optimized:** Fsv2 series
    c. **Memory Optimized:** Esv3, Easv4, Edsv4, and Esv4 series
    d. **Storage Optimized:** Lsv2 series
2. **Burstable Disk:**
    a. Premium SSD
    b. Standard SSD

Payment options for Virtual Machines

Azure provides you with a variety of options to pay for compute capacity. Here are the following payment options:

1. **Pay as you go** - you are billed on a per-second basis. You can start or stop anytime, and you only pay for what you use. This payment option is ideal for users who prefer flexibility or have unpredictable workloads that cannot be interrupted.
2. **Reserved Instance** - you get up to 72 percent price savings compared to pay-as-you-go, but in return, you need to pay the upfront cost and be committed for one or three years in a specified region. There are three options to scope a reservation:
    a. **Single resource group** - the reservation discount applies solely to the corresponding resources in the resource group you've chosen. Keep in mind that discounts will not be applied if the resource group is moved or deleted.
    b. **Single subscription** - the reservation discount applies only to the corresponding resources in the subscription you've selected.
    c. **Shared** - the reservation discount is applied to the corresponding resources in eligible subscriptions within the billing context. If the subscription is moved to a different billing

context, the discounts no longer apply to that subscription but will continue to apply to the remaining subscriptions in the billing context.

i. The billing context for Enterprise Agreement customers is enrollment. In an enrollment, the reservation shared scope contains multiple Active Directory tenants.
ii. The billing scope for Microsoft Customer Agreement customers is billing profile.
iii. The billing scope for individual subscriptions with pay-as-you-go rates is all eligible subscriptions.

After purchasing a reservation, you can always update the scope. Go to the reservation, click **Configuration**, and then rescope the reservation. Rescoping a reservation won't change the reservation term.

3. **Spot** - save up to 90 percent when you purchase unused compute capacity. This is only ideal for workloads that can tolerate interruptions. Discounts may vary based on:
    a. Region
    b. Virtual machine type
    c. Compute capacity

Since Azure Spot Virtual Machines are unused capacity, at any point in time, Azure infrastructure can evict Spot VMs with 30 seconds notice. Eviction is based on the capacity or the max price you've set. When creating a Spot VMs, you can set the eviction policy to Deallocate (default) or Delete.

The Deallocate policy moves your virtual machine to the stopped-deallocated state, allowing you to redeploy it later. However, there is no assurance that the allocation will be successful. Your quota will be depleted by the deallocated VMs, and you will be charged for the underlying disks.

If you want your virtual machines to be deleted when it is evicted, you can set the eviction policy to Delete. The underlying disks are also deleted, so you won't be charged for the storage. In the portal, you can look up the eviction rates by size in a certain region. Go to **View pricing history and compare prices in nearby regions** to see a table or graph of pricing for a specific size.

Availability Options for Virtual Machines

There are two ways to manage the availability and resiliency of your applications in a virtual machine:

1. **Availability zones** - to protect your resources from an entire data center failure, you need to deploy the VMs to a minimum of three Availability Zones to ensure resiliency. Azure services that support Availability Zones are classified into two types:
    a. Zonal services - resources are pinned to a specific Availability Zone.
    Examples: Virtual machines, Managed disks, Standard IP addresses
    b. Zone-redundant services - replicate resources automatically across Availability Zones to protect from single points of failure.

Examples: Zone-redundant storage, SQL Database

2. **Availability sets** - to protect from hardware failures within a data center, you can deploy the virtual machine to an availability set. Each VM in an availability set is assigned to an update domain and fault domain. This option ensures that at least one is available during planned or unplanned maintenance events.
    a. Update domains (planned maintenance)
        i. A logical group of virtual machines that can undergo maintenance at the same time. By default, it has five non-user-configurable update domains. It can be increased up to 20 update domains and given 30 minutes to recover before maintenance is initiated on a different update domain.
    b. Fault domains (unplanned maintenance)
        i. A logical group of virtual machines that share a common power source and network switch. By default, VMs within an availability set are separated up to three fault domains.

Virtual Machine Scale Sets

When you need to improve the performance of your applications and also provide redundancy, you should scale your resources horizontally. Horizontal scaling means you are adding more servers to the system. By doing this, the workload will be distributed across multiple resources and accommodate the increasing demand. Take note that this type of scaling is different from vertical scaling. When you say scale vertically, you are increasing or decreasing the resources of a single server instead of adding new servers to the system.

The horizontal scaling service in Azure is called **virtual machine scale sets**. A VM scale set allows you to create and manage a group of load-balanced VMs. Since the workload is distributed, if one VM fails, you can still continue to access your application through other VMs with minimal interruption. You can also distribute VMs in a scale set within a single data center or across various data centers. This service supports both layer 4 basic traffic distribution and layer 7 advanced traffic distribution and TLS termination.

Virtual Machine Scale Sets provide the following key benefits:

- By creating scaling policies, you can automatically add or remove virtual machines based on host metrics. A host metric provides you visibility into the performance of the virtual machines in a scale set without the need to install and configure agents. An example of host metrics can be CPU Utilization, Network In, and many more.
- You can create health checks and set a repair policy to automatically replace unhealthy virtual machines. Unhealthy instances are reported by Application Health extension or Load Balancer health probes.

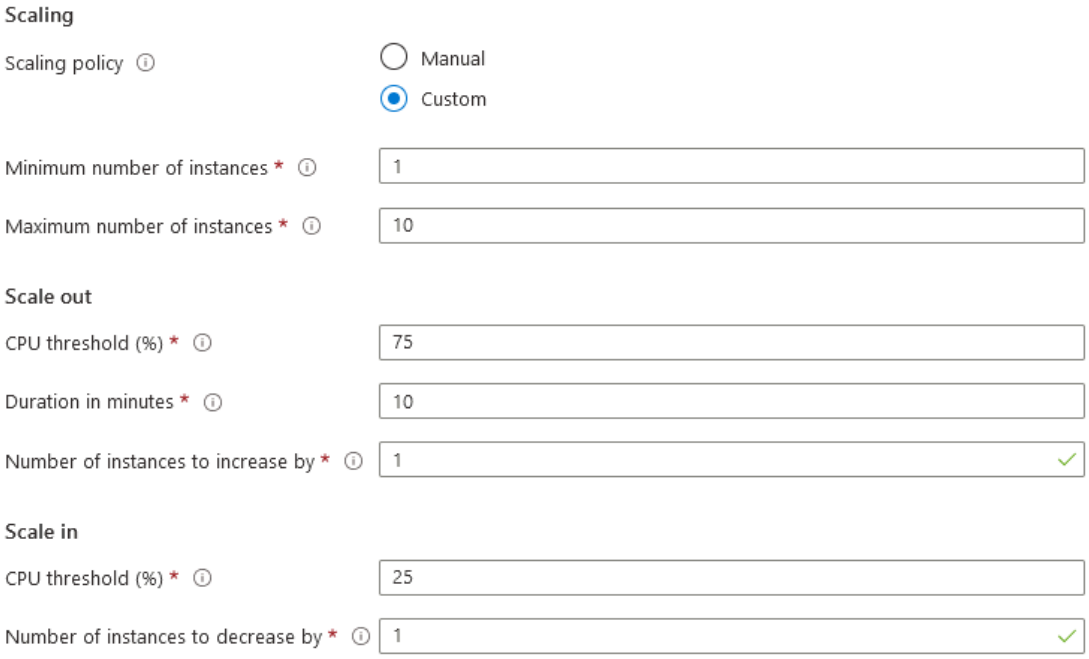
- You can associate virtual machine scale sets with a load balancer. This will allow you to distribute virtual machines across Availability Zones. By implementing this practice, you can make your application redundant and highly available.
- Lastly, virtual machine scale sets allow you to scale hundreds or even thousands of virtual machines.

Now that we know scale sets can be associated with load balancers, this will help us implement one of the best practices on architecting in the cloud by evenly distributing the virtual machines across different Availability Zones. The main reason why you need to configure it with a load balancer is to give you high availability. An application that can run continuously even if one of the virtual machines fails. Aside from distributing the load across AZs, one of the added benefits is you can use Load Balancer health probes for more robust health checks.

When associating scale sets with a load balancer, you have two options:

1. **Azure Application Gateway** - is an HTTP/HTTPs web traffic load balancer that has the capability to do the following: URL-based routing, SSL termination, session persistence, and web application firewall.
2. **Azure Load Balancer** - a TCP/UDP network traffic load balancer that supports port forwarding and outbound flows.

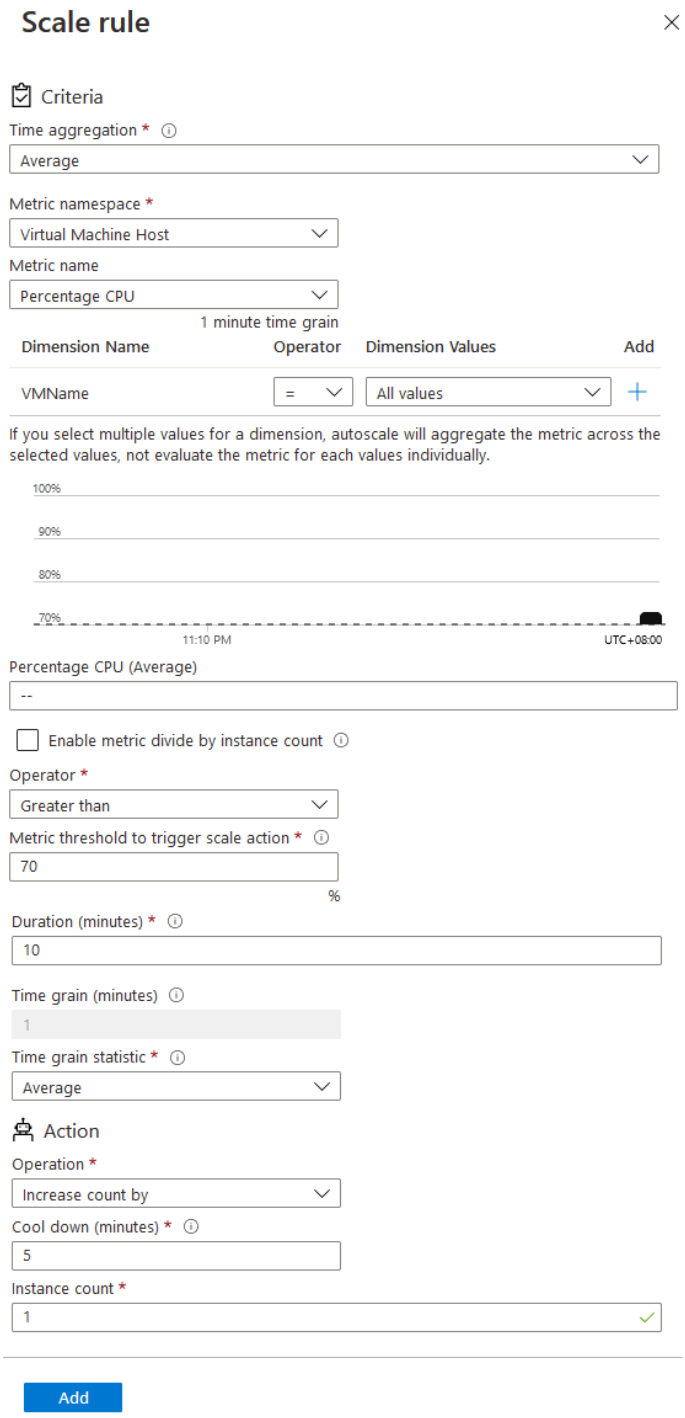
After going through load balancing, let's now talk about the scaling policy and how it works. A **scaling policy** can determine when a virtual machine should be added or removed to meet the current capacity requirements of your application. When you create a virtual machine scale set, you would see this configuration in the portal.



These configurations can only be seen if you select the custom scaling policy option. The first thing that you can set is the number of instances. But let's focus on the remaining two options, the scaling out and the scaling in. **Scale out** is when you need to add virtual machines to the scale set to increase the current capacity. In order to scale out, you should input certain values on the following fields:

- CPU threshold - is the CPU usage percentage threshold on when to trigger the scale out rule.
- Duration in minutes - is the amount of time that the autoscale will check the threshold again.
- Number of instances to increase by - this will determine how many virtual machines should be added when the scale out rule is triggered.

On the other hand, the **scale in** rule is when should the scale sets remove a virtual machine in order to decrease the capacity. Unlike scale out, you only need to input two values in the scale in fields. After you create a virtual machine scale set, you will see a lot of options available that you can configure in the scaling policy.



As seen in the image above, you can still configure other options in order to meet certain requirements on when to scale your virtual machines. Here are the options that you can customize:

1. **Metric Name** - allows you to set the metric that will be collected to your virtual machine. Some of the metrics that you can choose from are:
    - Percentage CPU

    - Network In or Out
    - Disk Read or Write Bytes
    - Disk Read or Write Operations/Sec
    - CPU Credits Consumed or Remaining
2. **Aggregates** - it is how you want to collect the data. For example, TimeAggregation = “Sum” will aggregate the sampled metrics by taking the sum. The methods that you can select from are:
    - Average
    - Minimum
    - Maximum
    - Sum
    - Last
    - Count
3. **Operators** - this will determine when to trigger scale action.
    - Greater than
    - Greater than or equal to
    - Less than
    - Less than or equal to
    - Equal to
    - Not equal to
4. **Actions** - what should the scaling policy do after it is triggered.
    - Increase count by
    - Increase percent by
    - Increase count to
    - Decrease count by
    - Decrease percent by
    - Decrease count to

If you want to collect more information based on different metrics, you need to install the following:

- **App Insights** - when you want to collect application metrics such as page load performance and session counts, you can install app insights in your application, and it will monitor your app and send telemetry to Azure.
- **Azure Diagnostic Extension** - when you want detailed Host-based metrics, you can install this extension. This agent will run inside your virtual machine. It will monitor and save performance metrics to an Azure storage service to collect more detailed information.

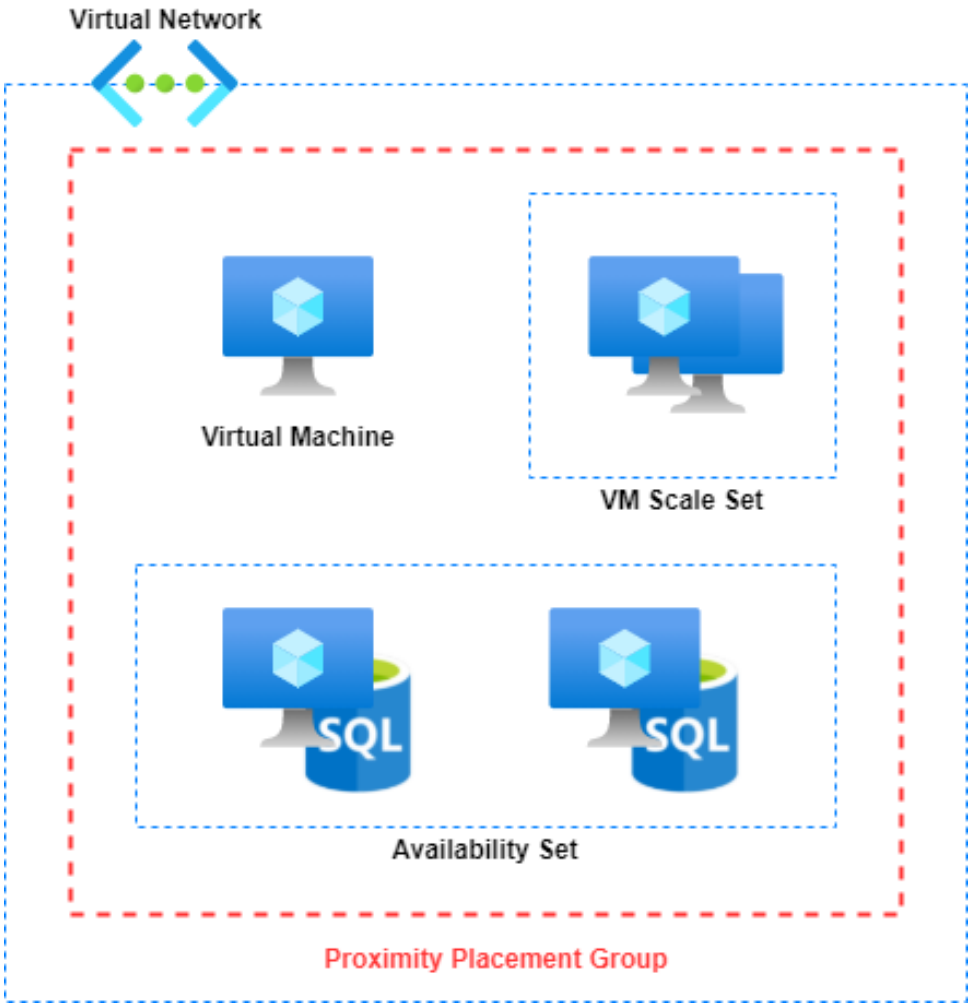
There are also other options that you can configure when creating a virtual machine scale set:

1. **Scale-In Policy** - allows you to specify the order in which virtual machines are deleted during a scale-in operation. The options that you can select from are:
    a. Default
        - Balance across availability zones and fault domains
        - Deletes the VM with the highest instance ID
    b. Newest VM
        - Balance across availability zones
        - Deletes the newest created virtual machine
    c. Oldest VM
        - Balance across availability zones
        - Deletes the oldest created virtual machine
2. **Update Policy** - allows you to set how you can upgrade your virtual machines to the latest scale set model.
    a. Automatic - upgrades will start immediately in random order.
    b. Manual - the existing virtual machines must be manually upgraded.
    c. Rolling - upgrades are rolled out in batches with the option to pause.
3. **Automatic OS Upgrades** - by enabling this option, the upgrades on the OS disk will be done automatically for all virtual machines.
4. **Health Monitoring** - helps you determine if your resources are healthy or unhealthy. There are two modes that you can select:
    a. Application Health Extension - pings an HTTP/HTTPs request with a specific path and returns an HTTP status.
    b. Load Balancer Probe - checks are done through TCP/UDP or HTTP/HTTPs requests. You can only select this option if you have an associated load balancer.
5. **Automatic Repair Policy** - automatically replace unhealthy virtual machines with a new one.
6. **Allocation Policy** - allows you to scale beyond 100 instances (default).

Proximity Placement Groups

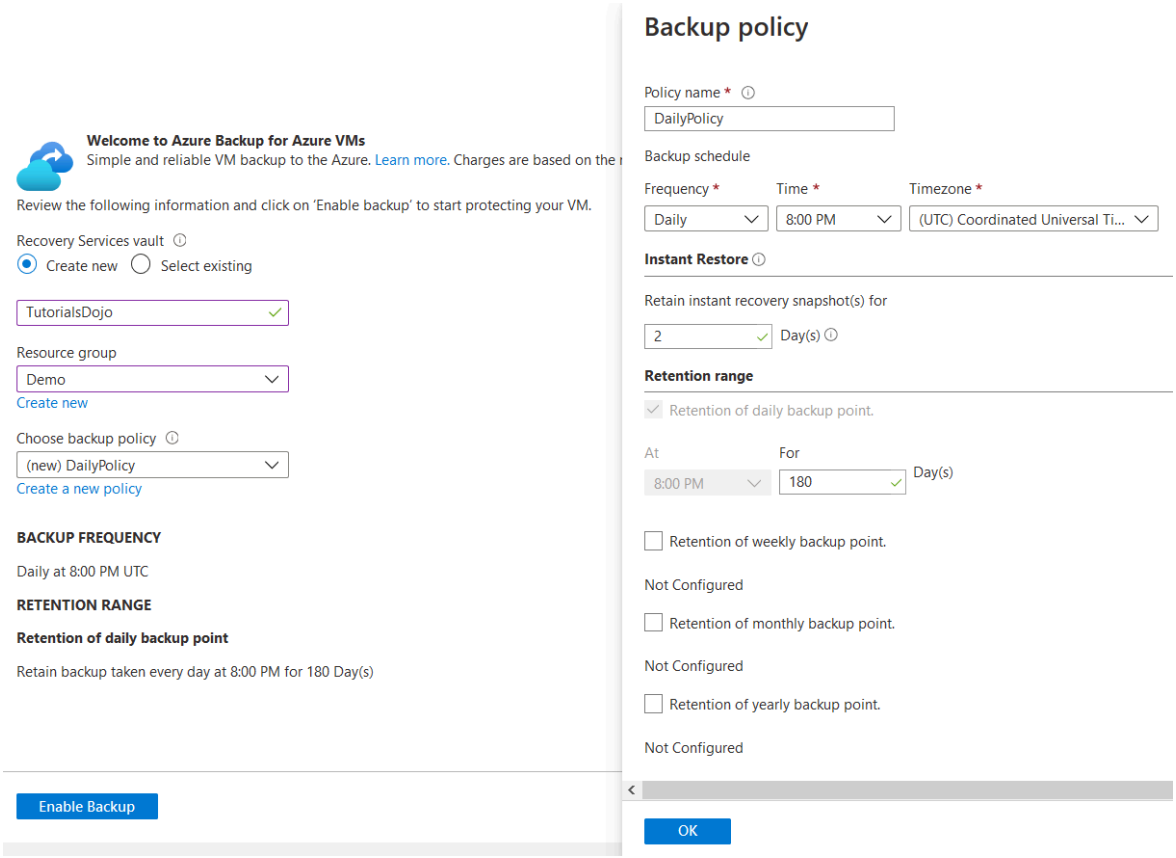
If you need to reduce the physical distance between virtual machines, you place it in a single region. To bring VMs physically close together, you deploy it in a single availability zone. However, a single AZ may span in different data centers, which can result in network latency that impacts your application. To achieve the lowest possible latency, you need to deploy the virtual machines in a proximity placement group.

This is a logical grouping to make sure that the resources are physically close to each other. It could be used with VMs, availability sets, and VM scale sets. You can also move your existing resources into a proximity placement group. This configuration is helpful if latency is your first priority. But, if you need to have resiliency, you should spread your virtual machines across availability zones. Remember that a single proximity placement group cannot span zones, and by default, it can only hold 100 virtual machines. You can scale beyond the limit if the **singlePlacementGroup** property is set to false. Therefore, multiple placement groups can hold up to 1,000 virtual machines.



Backup Azure Virtual Machines

To prevent unintended destruction or deletion of data on a virtual machine, you need to create a backup of your data. With the **Azure Backup** service, you can back up on-premises machines, workloads, and Azure VMs. The backup schedule can be configured on a daily or weekly basis. While the backup retention can be set on a daily, weekly, monthly, and yearly schedule.



If you would recall, the VM in a stopped/deallocated state only stops the virtual machine and Azure Backup only takes snapshots of the VM disks. This means that even if the VM status is running or stopped, you can still create a backup as long as the disk is attached to the VM. Remember that you can only backup data sources or virtual machines that are in the same region as the Recovery Services vault. You can also back up virtual machines that have different resource groups or operating systems as long as they are in the same region as the vault.

To understand how Azure Backup works, let's take a look at its components:

1. A **Recovery Services vault** is a storage entity that stores data and recovery points that have been created over time. The data stored in the vault is a copy of data or configuration information for

virtual machines, workloads or servers. It also supports service integration like System Center DPM, Windows Server, Azure Backup Server, and many more.

A Recovery Services vault can't be deleted if it contains the following dependencies:
- Protected data sources (IaaS VMs, SQL databases, Azure file shares)
- Backup data
- Backup data (soft-deleted state)
- Registered storage accounts

Therefore, if you try to delete the vault without removing the dependencies, you will receive an error message "Failed to delete resource group." To resolve this problem, you must stop the backup first, then disable soft delete and delete the resource group.

2. The **Microsoft Azure Recovery Services (MARS)** agent allows you to backup data from Windows virtual machines and on-premises machines. The backups are directly stored to the Recovery Services vault. In order to install the MARS agent and perform backups, you need to have the following:
   - Recovery Services vault
   - Backup policy
   - Secure route (ExpressRoute or Private Endpoint)

3. Azure Backup Vault is an important component of the Azure Backup service. It serves as a storage entity that houses backup data for various Azure services. It's designed to support newer workloads and services like Azure Database for PostgreSQL servers. The vault simplifies the organization of backup data, reducing management overhead. It incorporates enhanced security features to protect cloud backups and ensure safe data recovery, even in compromised environments. The vault is compatible with Azure role-based access control (RBAC), providing fine-grained access management. It also ensures data isolation by storing backup data in a Microsoft-managed Azure subscription and tenant, separate from the production environment. The vault handles storage settings and encryption settings, offering options for platform-managed keys and customer-managed keys for backup data encryption.

4. The **Azure Storage Explorer** is a standalone application that offers a unified interface for managing Azure Storage data. It simplifies the process of managing and navigating Azure Storage accounts, including Blob Containers, File Shares, Queues, and Tables. Users can easily upload, download, and manage data across multiple subscriptions using this tool. It also supports advanced features such as blob snapshots, blob versioning, and setting up blob access tiers. Azure Storage Explorer is available on Windows, macOS, and Linux, which makes it a convenient choice for developers and administrators working on different platforms.

5. To create a **Backup Policy**, you need to select a *datasource type* first. You can choose between Azure Virtual Machines or Azure Database for PostgreSQL servers. Then choose *frequency*, whether it is daily or weekly. After that, you input how many days you want to retain the recovery snapshots. At most, you can only retain the instant recovery snapshot for 5 days. Lastly, choose the *retention range* of your backups, if it's daily, weekly, monthly, or yearly.

6. For hybrid backup solutions for site-to-site recovery or business continuity and disaster recovery (BCDR) strategy, you can use **Azure Site Recovery.** It replicates workloads from a primary site to the secondary site. For example, your primary site suddenly suffers outages. Since the primary site becomes unavailable, Azure Site Recovery will automatically failover to the secondary site and ensure that your services are still working. The following resources can be replicated:
    - Azure Virtual Machines (Cross Region Replication)
    - Any operating system
    - On-premise to Azure
    - Other Cloud Service Providers to Azure
    - VMWare, Hyper-V, or Physical Servers

7. Lastly, it's important to know these Disaster Recovery terms:
    - **Recovery Time Objective (RTO)** is the time it takes after a disruption to restore a business process to its service level.
    - **Recovery Point Objective** (RPO) is the acceptable amount of data loss measured in time before the disaster occurs.

vCPU quotas

The vCPU quotas for VMs and VM scale sets are arranged in two tiers for each region in a subscription.

- Total Regional vCPUs
- VM size family cores

Every time you deploy a new VM, the vCPUs must not exceed the vCPU quota for the VM size family or the total regional vCPU. If either of those quotas has been exceeded, the VM deployment will not be allowed. Take note that there is also a quota for the overall number of virtual machines in the region. The quota is calculated based on the total number of cores in use, both allocated and deallocated. If you need additional cores, you can request a quota increase or delete VMs that are no longer needed.

References:
https://docs.microsoft.com/en-us/azure/virtual-machines/
https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/overview
https://docs.microsoft.com/en-us/azure/backup/backup-azure-vms-introduction
https://azure.microsoft.com/en-us/pricing/details/virtual-machines/windows/
https://learn.microsoft.com/en-us/azure/storage/storage-explorer/vs-azure-tools-storage-manage-with-storage-explorer?tabs=windows
https://learn.microsoft.com/en-us/azure/backup/backup-vault-overview

Azure App Service

App Service Plans

In App Service (Web Apps, API Apps, or Mobile Apps), an application always runs in an App Service plan. An **App Service plan** is a collection of compute resources needed for a web app to run. If you have one or more apps, you can set them up to share the same computing resources (or in the same App Service plan). Each plan consists of a **region, number & size of virtual machines,** and **pricing tier**.

The pricing tiers that you can choose from are:

1. **Shared Compute** – **Free** and **Shared** are the two base tiers. These tiers allocate CPU quotas to every app running on the shared resources, but the resources cannot scale-out.
2. **Dedicated Compute** – It is composed of **Basic, Standard, Premium,** and **PremiumV2** tiers. As the tier gets higher, you will have more VMs to scale-out.
3. **Isolated** – A dedicated virtual machine that provides maximum scale-out capabilities.

	Free	Shared	Basic
Description	Shared environment for dev/test		Dedicated environment for dev/test
App Service plan	10 per Region	10 per Resource Group	100 per Resource Group
Web, mobile, or API apps per App Service plan	10	100	Unlimited
Compute Type	Shared	Shared	Dedicated
Maximum Instances	1	1	up to 3
Storage Size	1 GB	1 GB	10 GB
Memory	1 GB	1 GB	up to 7 GB
Staging Slots	-	-	-
Scheduled Backups	-	-	-
Auto Scale	-	-	Supported
Custom Domain	-	Supported	Supported

Hybrid Connectivity	-	-	Supported
VNet Connectivity	-	-	-
Private Endpoints	-	-	-
SLA	-	-	99.95%
	Standard	**Premium**	**Isolated**
Description	Run production workloads	Enhanced performance and scale	High-Performance, Security and Isolation
App Service plan	100 per Resource Group	100 per Resource Group	100 per Resource Group
Web, mobile, or API apps per App Service plan	Unlimited	Unlimited	Unlimited
Compute Type	Dedicated	Dedicated	Isolated
Maximum Instances	up to 10	Up to 20 for v1 and v2 up to 30 for v3	up to 100
Storage Size	50 GB	250 GB	1 TB
Memory	up to 7 GB	up to 32 GB	up to 14 GB
Staging Slots	5 per app	20 per app	20 per app
Scheduled Backups	Supported	Supported	Supported
Auto Scale	Supported	Supported	Supported
Custom Domain	Supported	Supported	Supported
Hybrid Connectivity	Supported	Supported	Supported
VNet Connectivity	Supported	Supported	Supported
Private Endpoints	-	Supported	Supported
SLA	99.95%	99.95%	99.95%

Before you launch a web app in Azure App Service, you must also select the Operating System that will be used in the App Service plan. Take note that some runtime stacks will only work on Windows such as ASP.NET while Ruby will only work with Linux.

If you created a Linux web app in a Windows App Service plan, you would receive an error message: "The template deployment is not valid according to the validation procedure". To resolve this error, you must create a new App Service plan. Conversely, you will also receive the same error if you run an ASP.NET V4.8 application in a Linux App Service plan.

With Azure App services, you can choose the following runtime environment:

Runtime stack	Linux	Windows
.NET		
.NET 5	Yes	Yes
ASP.NET V.3.5 & V.4.8		Yes
.NET Core		
.NET Core 2.1 & 3.1	Yes	Yes
Java		
Java SE 8 & 11	Yes	Yes
Node		
Node 10, 10.1 & 14	Yes	
Node 10.6, 10.14 & 12	Yes	Yes
Node 10.10		Yes
PHP		
PHP 7.2, 7.3 & 7.4	Yes	Yes
Python		
Python 3.7 & 3.8	Yes	
Python 3.6	Yes	Yes
Ruby		
Ruby 2.5 & 2.6	Yes	

When you create your app, you must also choose a unique web app name because it will become a fully qualified domain name (FQDN). After you have configured your web app, the default domain name of your web app is: **<web-app-name>**.azurewebsites.net.

Lastly, the Azure app service can also run Docker containers (single/multi-container). You can define custom containers for Windows or Linux operating systems and push the image to Azure Container Registry (ACR). This Azure service will handle all your private Docker container images and other related artifacts. Then, App Service will pull the image from ACR and takes care of all of the tasks associated with deploying container-based web apps, such as OS patching, capacity provisioning, and load balancing.

Deployment Slots

When you deploy a web app, API app, or mobile app to Azure App Service, the default slot is the production slot. With deployment slots, you can set up different environments for your application, and the created slot will have its own hostname. This is very useful when you need to have a staging or testing environment. Aside from creating environments, you can also swap environments. This means that you can change the staging environment into a production environment.

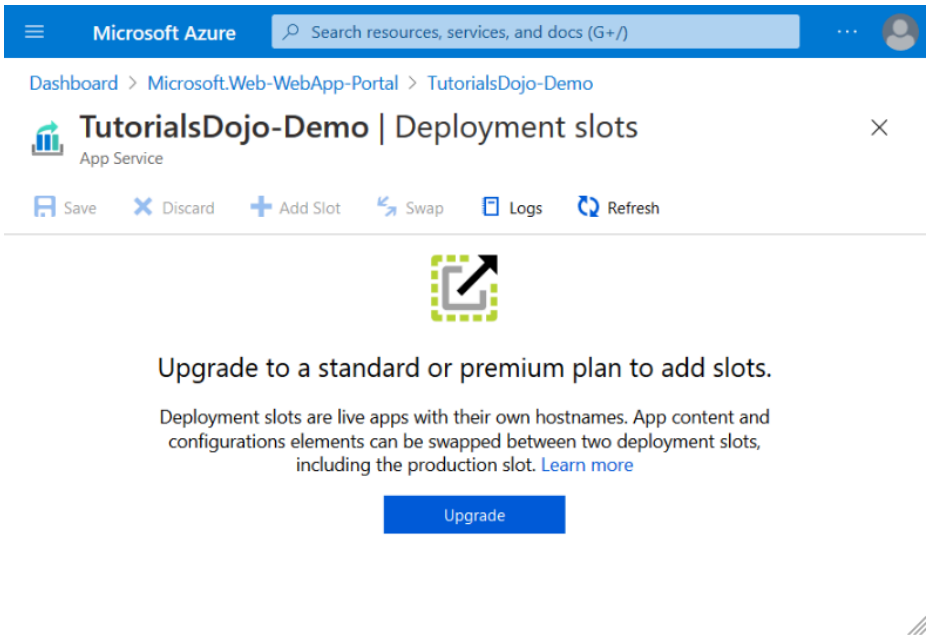
When you perform the swap operation, the following settings are swapped:

- General settings
- App settings
- Connection strings
- Handler mappings
- Public certificates
- WebJobs content

Settings that aren't swapped are:

- Publishing endpoints
- Custom domain names
- Non-public certificates and TLS/SSL settings
- Scale settings
- WebJobs schedulers
- IP restrictions
- Always On
- Diagnostic settings
- Cross-origin resource sharing (CORS)
- Virtual network integration

- Managed identities
- Settings that end with the suffix _EXTENSION_VERSION



If you encountered the image shown above, this means that your App Service plan does not have the capability to add a staging slot for your application. To solve this problem, you must upgrade your App Service plan to a **Standard** or **Premium** tier. After you successfully upgraded your plan, you can now add a slot in the deployment slots.

To understand deployment slots better, let's create an example:

You have a web app named tutorialsdojo-portal that is hosted in Azure App Services. The provisioned deployment slots for tutorialsdojo-portal are shown in the table below:

Name	Environment
tutorialsdojo-dev	Development
tutorialsdojo-staging	Staging
tutorialsdojo	Production

You configured several settings in the tutorialsdojo-dev and tutorialsdojo-staging.

You performed a swap operation between the production and staging slots. Upon testing the tutorialsdojo-portal app, it was discovered that the new features are not working properly.

A possible reason why the tutorialsdojo-portal web app is not working properly is that several settings are configured in the tutorialsdojo-dev and tutorialsdojo-staging. If you recall, when you perform swap operations, various settings are swapped.

To fix this issue, you can revert the tutorialsdojo-portal app to its previous state by swapping the slots of the tutorialsdojo-staging and tutorialsdojo environments. Since the slots have been swapped again, the app will no longer experience any performance issues.

Each App Service plan tier supports a different number of deployment slots, and there's no additional charge for using it. Also, take note that when you already have a Premium tier with more than five slots, you can't scale it down to a Standard tier because this tier only supports five deployment slots.

Diagnostics Logging

Diagnostics logging helps you access the information logged by Azure. There are five built-in diagnostics tools to assist you with debugging an App Service app:

1. **Application logging**
    - The generated log messages by your application. Each message has the following level and categories:
        - Disabled: None
        - Error: Error, Critical
        - Warning: Warning, Error, Critical
        - Information: Info, Warning, Error, Critical
        - Verbose: Trace, Debug, Info, Warning, Error, Critical
    - You can also specify the disk quota (MB) and retention period (days) for the application logs.
    - The logs can be found on the App Service file system or Azure Storage blobs.
2. **Web server logging**
    - This log message contains an HTTP method, resource URI, client IP, client port, user agent, and response code.
    - You can set the retention period (days) for the web server logs.
    - The logs are stored in Azure Storage blobs or App Service file systems.
3. **Detailed Error Messages**
    - A copy of the .htm error page. The page contains information on why the server returns an error code (HTTP code 400 or greater).
    - The logs are stored in the App Service file system.
4. **Failed request tracing**
    - Detailed information on failed requests. The information you can find here helps you improve the site performance and isolate a specific HTTP error.
    - For each failed request, one folder is generated, which contains the XML log file and XSL stylesheet.
    - The logs can be found on the App Service file system.
5. **Deployment logging**
    - This log is created when you publish content to your app.
    - You can also use this log to determine why the deployment failed. For example, if you use a custom deployment script and it fails, you can determine why the script is failing through deployment logs.
    - Like Detailed Error Messages and Failed request tracing, the logs are also stored in the App Service file system only.

App Service Environments

Let's talk about another feature of Azure App Service called **App Service Environment (ASE).** It provides a fully isolated and dedicated environment to securely run apps at a high scale. You can host Windows/Linux web apps, Docker containers, mobile apps, and Azure functions. Multiple ASEs in a single region or across multiple regions are ideal for horizontally scaling stateless application tiers in support of high requests per second workloads.

There are two types of deployment for ASE:

1. **External ASE** - it exposes the ASE-hosted apps on an internet-accessible IP address. In other words, if the virtual network is connected to your on-premises network, the apps in your ASE will have direct access to the on-premises resources. And since the ASE is inside the virtual network, it can also access the resources within it.
2. **Internal Load Balancer (ILB) ASE** - almost identical with External ASE, but the key difference is that ILB can expose ASE-hosted apps with an IP address in your virtual network. The internal endpoint is an internal load balancer.

Azure resources can be placed in a non-internet-routable network using Azure Virtual Network. To access these resources, you can use the VNet Integration feature. It allows your app to access resources in your virtual network. This feature is mainly used in multi-tenant apps. But if your app is in the App Service Environment, then you don't need to use the VNet Integration feature to reach the resources in the same VNet since ASE is already inside a virtual network.

The VNet Integration feature has two variations:

1. **Regional VNet Integration**
    - A dedicated subnet to the services that you integrate with.
    - Block outbound traffic using network security groups.
    - Route table allows you to send outbound traffic.
2. **Gateway-required VNet Integration**
    - It allows access to resources in the target virtual network.
    - Sync network allows you to sync certificates and network information.
    - You can add routes for outbound traffic.

When it comes to application deployment, Azure App Service offers several options:

1. **Run from package** - instead of copying the package files directly to the wwwroot directory, the ZIP package will be mounted directly as read-only to the wwwroot directory.
2. **Deploy ZIP or WAR** - uses Kudu service to deploy ZIP and WAR files. Kudu is the engine behind git deployments in Azure App Service. It supports the following functionality for ZIP file deployment:

   a. Leftover files from the previous deployment are deleted.
   b. Default build process, which includes package restore
   c. Deployment customization and logs
3. **Deploy via FTP** - in order to use the FTP protocol to upload files, you will need to have your own FTP client. Then go to the deployment center and get the FTP credentials for your FTP client.
4. **Deploy via cloud sync** - allows you to use Dropbox or OneDrive to deploy using cloud sync. When you turn on Sync, it will create a folder in your cloud storage service.

The following directory is where the deployment will occur:

- Windows - D:\home\site\wwwroot
- Linux - /home/site/wwwroot

To run background tasks in Azure App Service, you can use **WebJobs** to upload an executable script. This feature will enable you to run a program in the same instance as a web app, API app or mobile app without additional cost.

There are two types of WebJobs:

1. **Continuous**
   - Starts the script immediately after the WebJob is created.
   - Runs on all instances that the web app runs on.
   - Supports remote debugging.
   - Uses WebJobs Scale:
     - Single Instance - keeps a single copy of WebJob running regardless of instance count.
     - Multi-Instance - scale all WebJobs across all instances.
2. **Triggered**
   - Only starts when triggered: manually or scheduled (CRON expression)
   - Runs on a single instance.
   - Remote debugging is not supported.

The supported file types for scripts:

- Windows cmd: **.cmd, .bat, .exe**
- PowerShell: **.ps1**
- Bash: **.sh**
- PHP: **.php**
- Python: **.py**
- Node.js: **.js**
- Java: **.jar**

Reference:
https://docs.microsoft.com/en-us/azure/app-service/overview

Azure Container Instances (ACI)

Sizing and Scaling

ACI is a serverless container platform that allows you to run Docker containers on-demand without the need for infrastructure management. When you create an ACI instance, you need to specify the size of CPU and memory:

- **CPU** - the number of CPU cores you want to allocate to your container. The maximum number of vCPUs you can assign is determined by the region and SKU you select.
- **Memory** - the amount of memory you want to allocate to your container. You can start an ACI instance with as little as 1 GiB of memory. Similar to CPU, the region and SKU you choose to determine the maximum amount of memory you can assign.

To ensure that you have enough capacity to handle your workload at all times, you can use auto-scaling to define rules that automatically increase or decrease the number of container instances based on your workload's CPU or memory utilization. You can also manually scale your ACI instances using the Azure Portal, CLI, or PowerShell.

Container Groups

In Azure, a **container group** is a collection of containers that are assigned to run on the same host machine, and which share the same lifecycle, resources, local network, and storage volumes. This means that you can use ACI to deploy a multi-container application as a single unit. Multi-container groups are useful when you need to split a single functional job into a few container images.

The minimum number of CPU and memory that you can allocate to a container group is 1 CPU and 1 GiB of memory. While the maximum resources in a container group can be found in the resource availability for ACI in the deployment region.

Container groups can share an externally accessible IP address, as well as one or more ports on that address and a DNS label with a fully qualified domain name (FQDN). To enable external clients to connect to a container in the group, you need to expose the port on the IP address and the container. Take note that when you delete the container group, its IP address and FQDN are released.

The supported external volumes that you can mount within a container group are:

- Azure file share
- Secret
- Empty directory
- Cloned git repo

Configuring Container Apps

Containers, cloud-native, and microservices are all used in modern software development and deployment. A **container** is a standalone executable package that contains everything needed to run a piece of software, creating an isolated environment for the application. While **microservices** are an architectural paradigm for developing applications that are composed of small, independent services. This enables teams to autonomously build, deploy, and grow their services, increasing the speed and agility of the development process.

In Microsoft Azure, there is a fully managed service called **Azure Container Apps,** where you can deploy, manage, and scale multi-container applications and microservices. You can run your containers without worrying about the challenges of managing cloud infrastructure and complex container orchestration solutions. To understand how this service works, we're gonna take a look at a series of step-by-step tutorials on creating Container Apps.

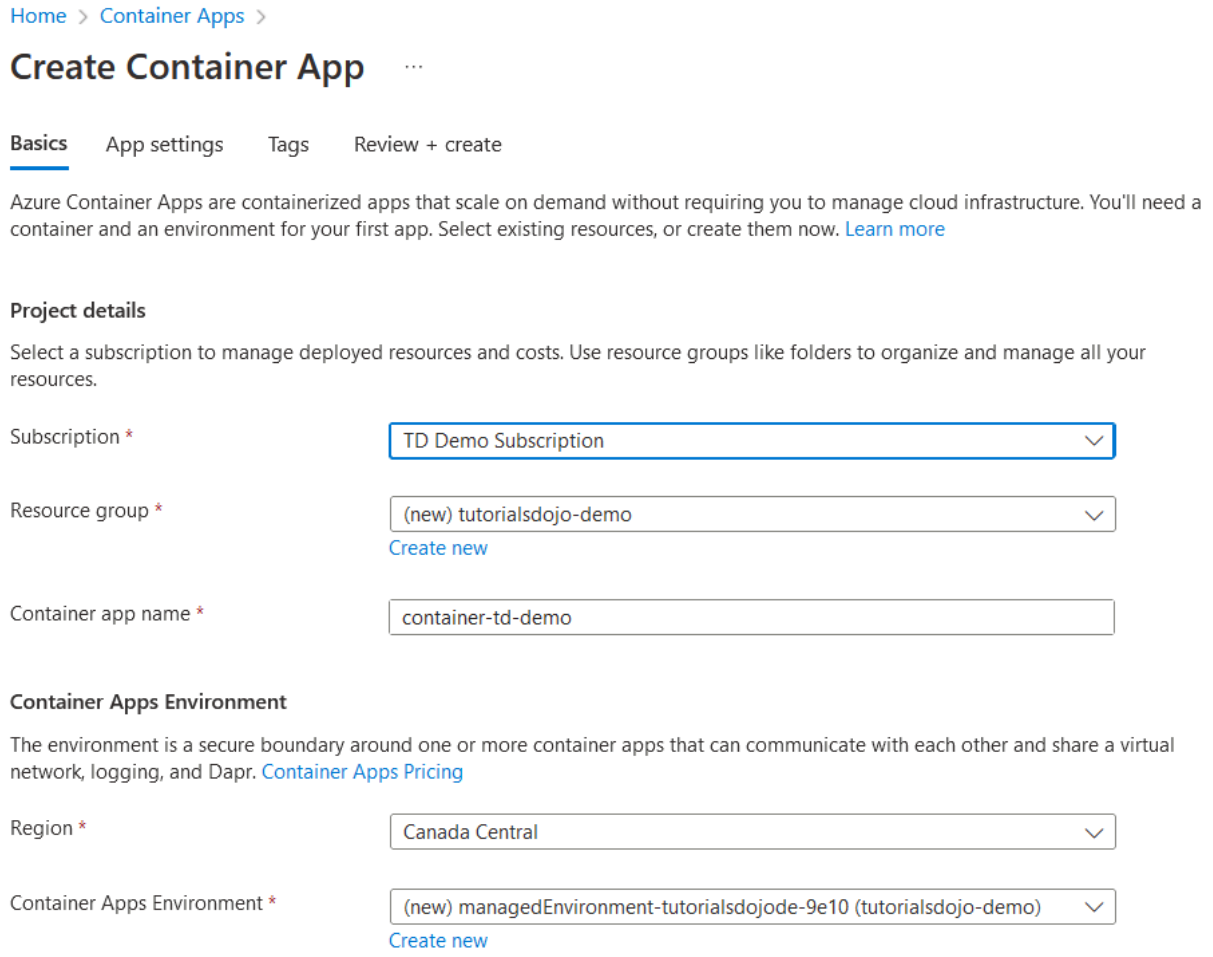
In this section, we'll learn how to create, configure and deploy Azure Container Apps using Docker images. The container images will be retrieved from Docker Hub, which is a repository of container images from software vendors, open-source projects, and the community.

There are several methods for deploying container images:

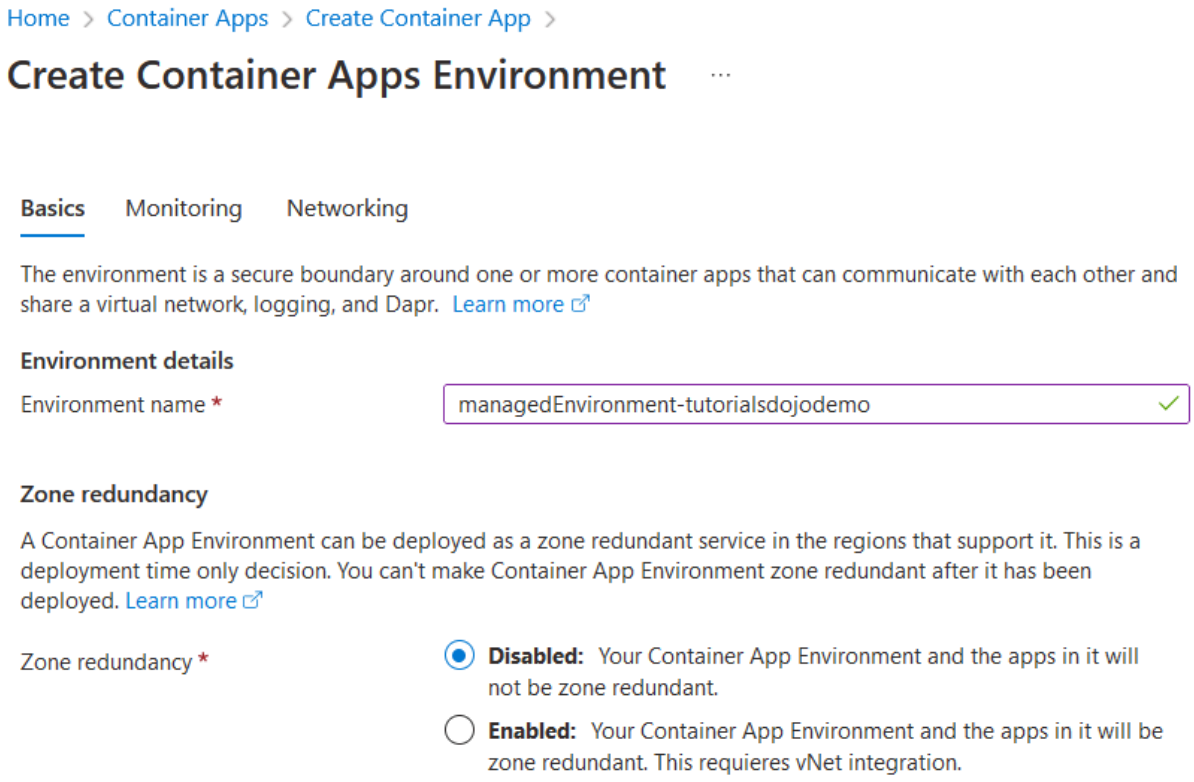
- Azure Portal
- Azure CLI
- VS Code

However, for this tutorial, we will configure Azure Container Apps using the Azure Portal. The first step you need to do is search the keyword "container apps" and click **"Create container app".**

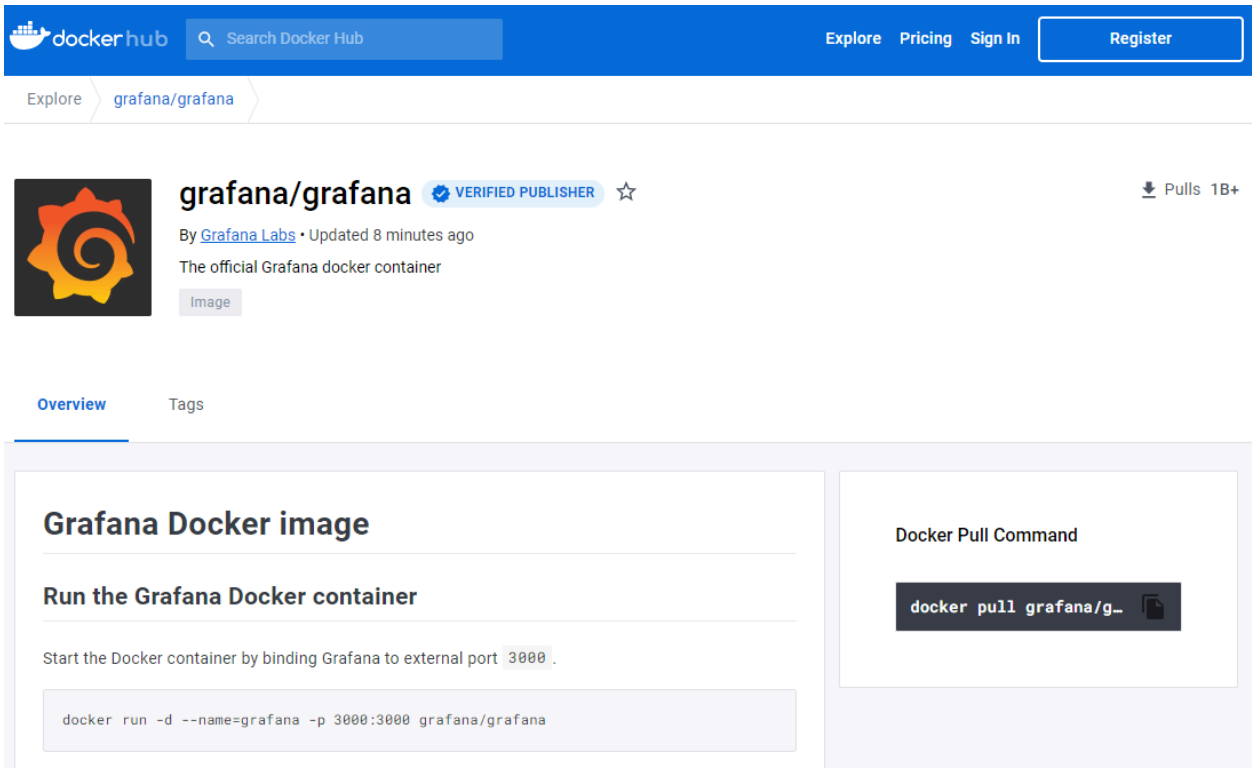
1. Once you're in the configuration settings, you must fill in the following details that have a red asterisk.



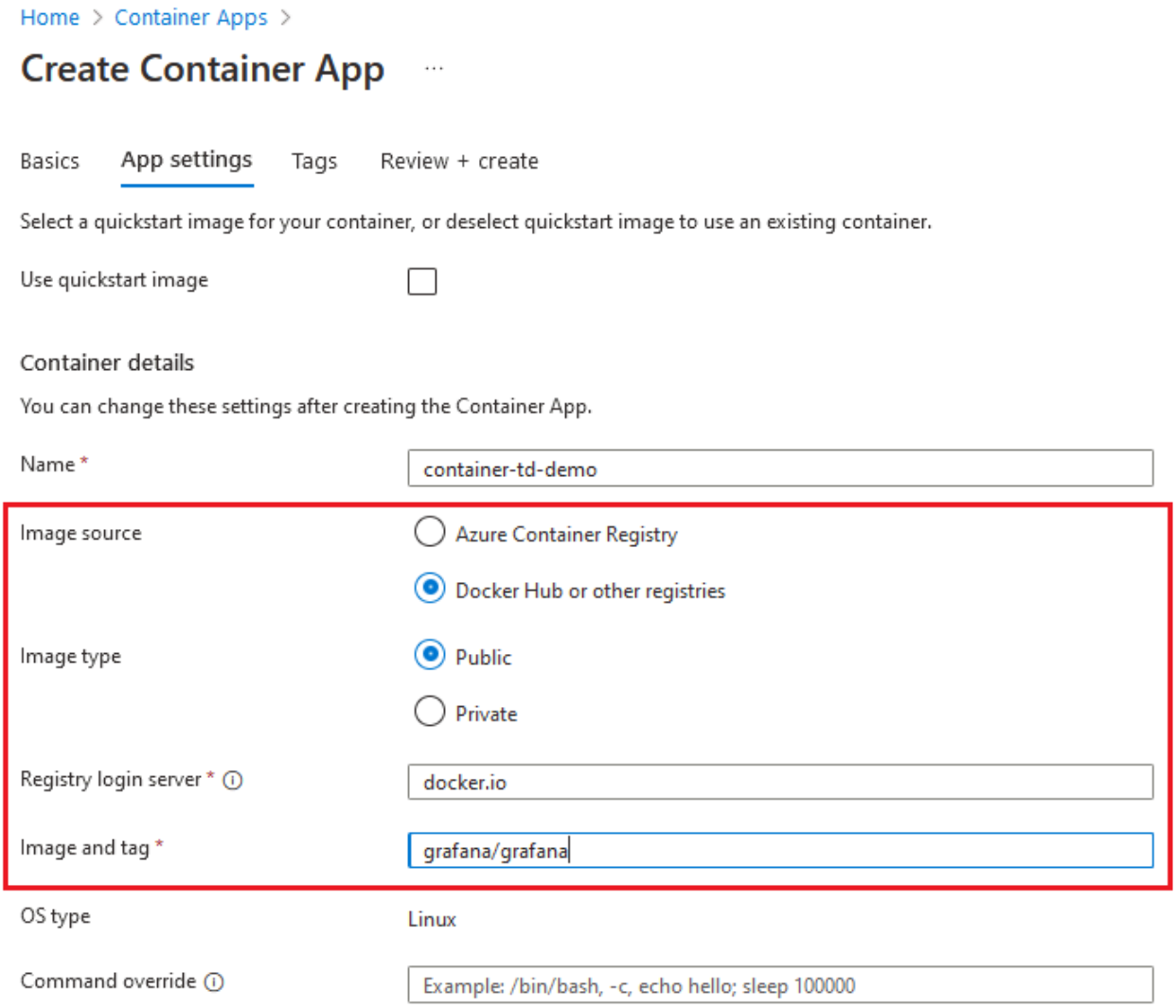
2. The **Container Apps Environment** field will be created automatically by default, but you can also configure it based on your needs. When you click **Create new**, you'll be taken to the Create Container Apps Environment page, which includes the following tabs:
    a. Basics - configure environment name and zone redundancy.
    b. Monitoring - create a log analytics workspace to store application logs.
    c. Networking - select the default or custom virtual network.



3. Next is the app settings tab; we need to untick the **"Use quickstart image"** to use a custom image from Docker Hub. After that, we need to select an image from Docker and the container image that we will use is Grafana.

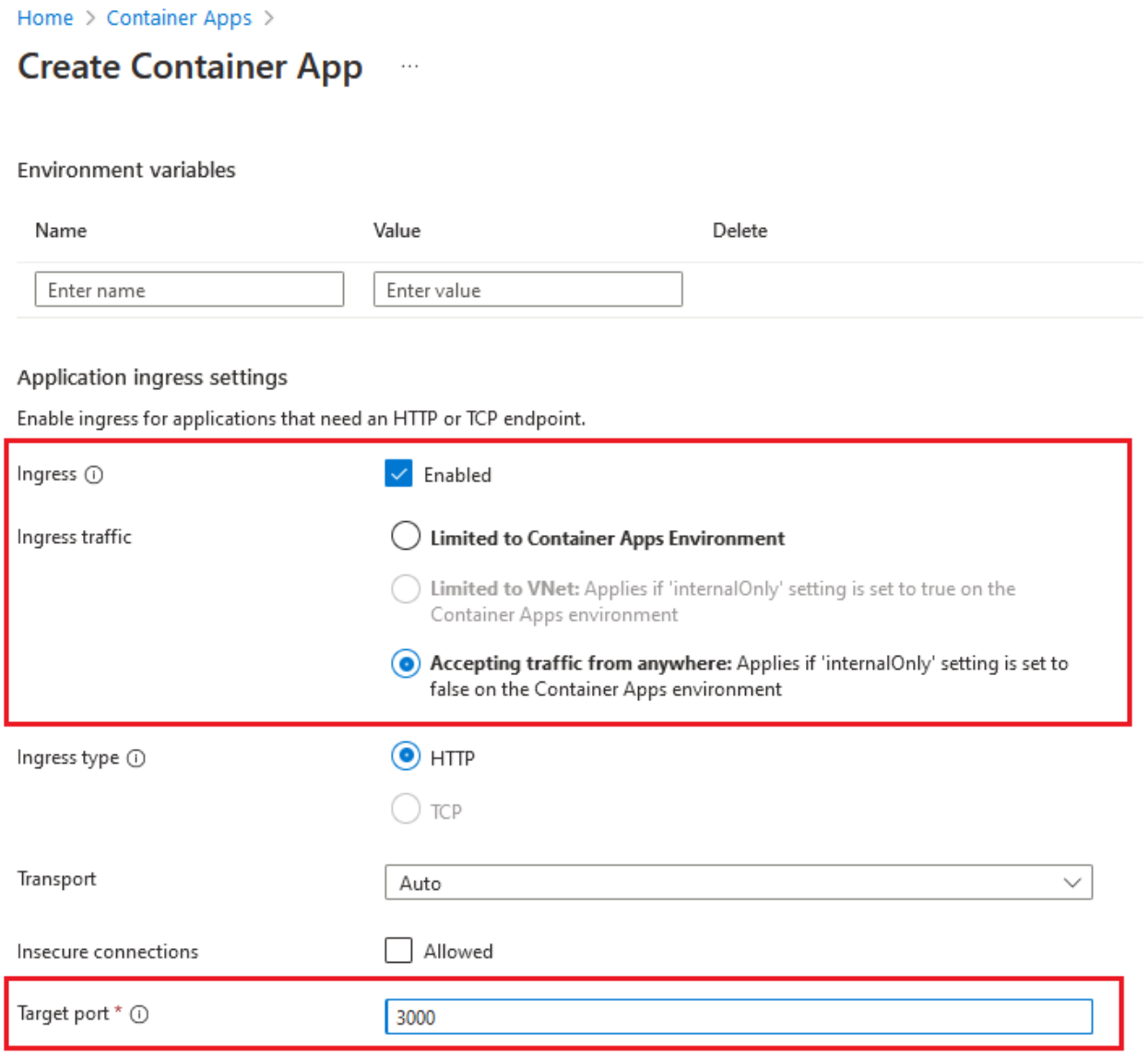


4. After you copy the Grafana image and tag, fill in the following details:



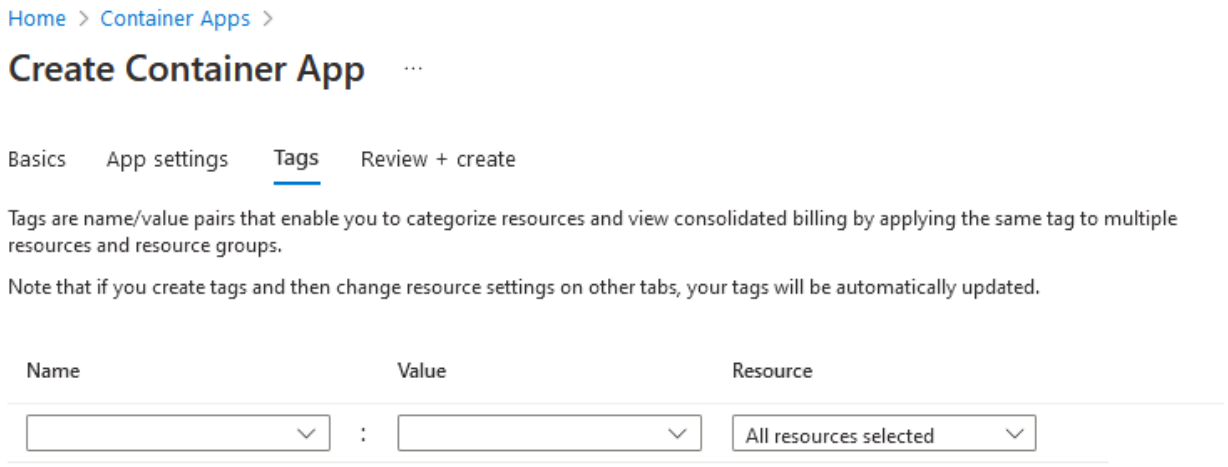
Note: If you don't specify the image's tag version, you'll always get the latest version.

5. If you scroll down, you'll see an **Application ingress settings** section. Don't forget to enable the ingress, select "**Accepting traffic from anywhere**", and the target port of the container. This is the port your container is listening on that will receive traffic.

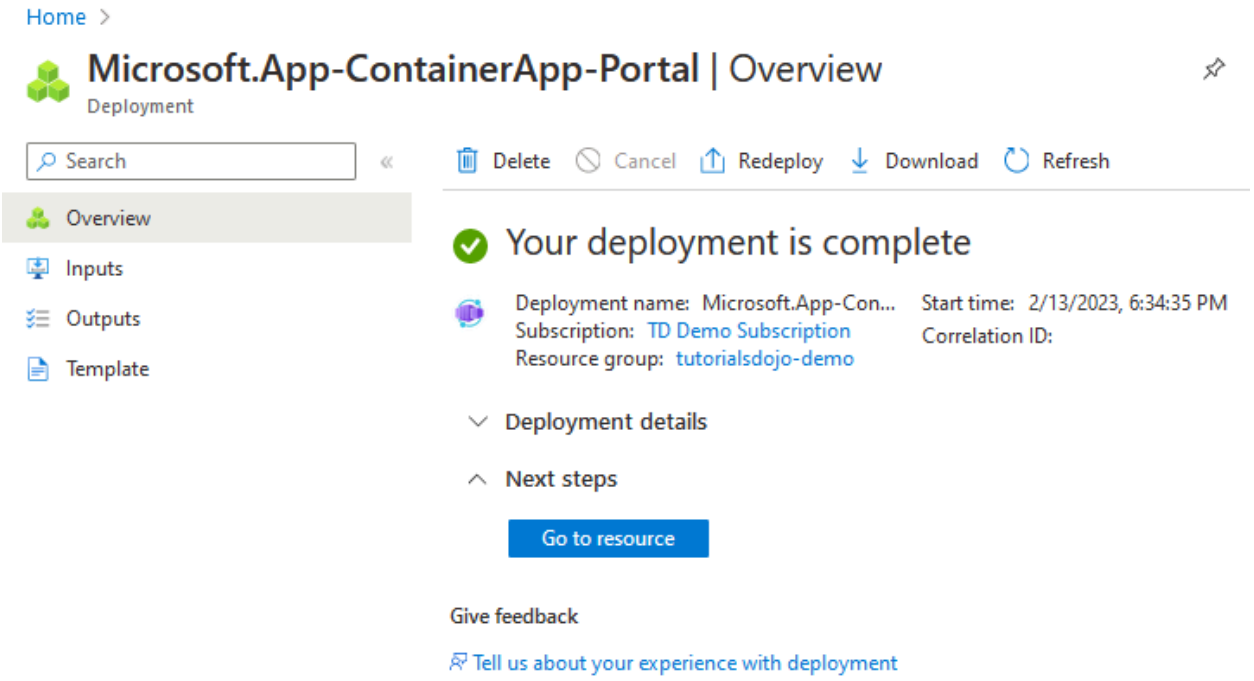


The main reason why we need to enable ingress is so we can generate an application URL. Also, the insecure connections option will just generate an HTTP URL.

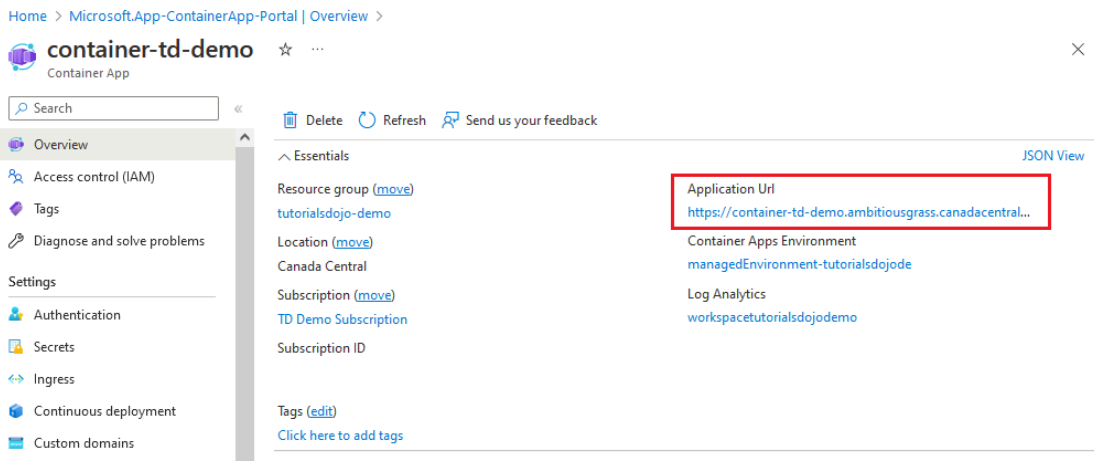
6. For the tags tab, this is optional, but for best practices, Azure recommends that you should add always add tags to organize your Azure resources.



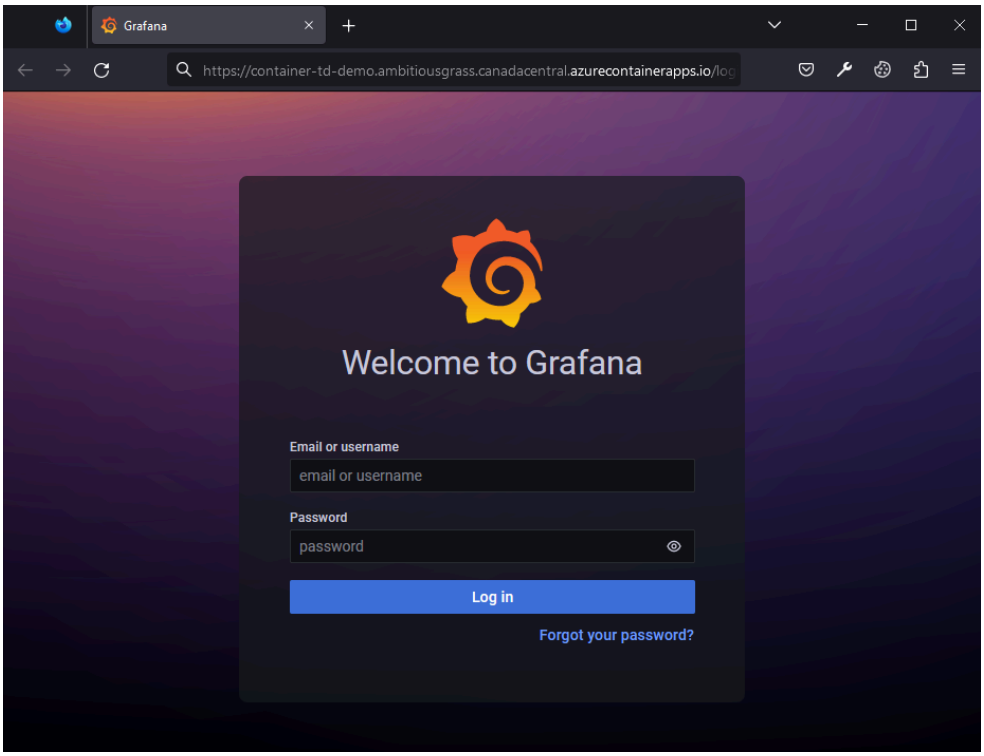
7. Before creating the container app, review all the details, and once you're done, click **Create**. Then, you'll be redirected to the **Deployment is in progress** page. The deployment will take a few minutes to be completed.



8. To verify the app that you've just deployed, go to resource and find the application URL.



9. After clicking the application URL, you'll be redirected to the Grafana app container.



10. That's it! Now you've deployed a Grafana image from Docker Hub with a few steps. Grafana is basically a dashboard to monitor the health and performance of all your resources in one platform. If you want to know about its features and how it works, then check out this article.

Reference:
https://learn.microsoft.com/en-us/azure/container-instances/container-instances-overview

Azure Kubernetes Service (AKS)

AKS is an open-source tool for orchestrating and managing many container images and applications. It lets you deploy a managed Kubernetes cluster in Azure. You use clusters and pods to deploy and scale applications. It also supports horizontal scaling, self-healing, load balancing, and secret management. Automatic monitoring of application load to determine when to scale the number of containers used.

Components

- A **control plane** is a managed Azure resource. It is where the components run, including API server and cluster database (etcd).
    - kube-apiserver – allows communication for management tools (kubectl).
    - etcd – a key-value store within Kubernetes.
    - kube-scheduler – defines what nodes should run in the workload.
    - kube-controller-manager – it oversees the smaller controllers that handle node operations and replication of pods.
- Kubernetes runs an application in your instance using **pods**.
- A **node** is made up of several pods, and **node pools** are a group of nodes with the same configuration.
- Use a **node selector** to control where a pod should be placed.
- You can run at least 2 nodes in the default node pool to ensure your cluster operates reliably.
- Multi-container pods are placed on the same node and allow containers to share the related resources.
- You can specify maximum resource limits that prevent a given pod from consuming too much compute resources from the underlying node.
- A **deployment** determines the number of replicas (pods) to be created, but you must define a manifest file in YAML format first.
- With **StatefulSets**, you can maintain the application's state within a single pod life cycle.
- The resources are logically grouped into a **namespace**, and a user may only interact with resources within their assigned namespaces.

Storage

A **volume** is used for storing, retrieving, and persisting data across pods and throughout the application lifecycle. You can either manually create data volumes to be assigned to pods or have Kubernetes create them for you. The data volumes that you can use are:

- **Azure Disks** - only available to a single node.
- **Azure Files** - share data across multiple nodes and pods.
- **Azure NetApp Files**
- **Azure Blobs** - mounted using NFS v3.0 protocol or BlobFuse

Common volume types in Kubernetes:

- **emptyDir** - data written to this volume type persists only for the duration of the pod's lifespan.
- **secret** - passwords and other sensitive data can be injected into pods.
- **configMap** - inject key-value pair properties (app configuration) into pods.

A **persistent volume** is a storage resource created and managed by the Kubernetes API that can exist beyond the lifetime of a single pod. To provide the PersistentVolume, you can use either Azure Disks or Azure Files. A PersistentVolume can be created statically by a cluster administrator or dynamically thru Kubernetes API. Keep in mind that Windows and Linux pods cannot share persistent volumes because of differences in file system.

If you need to define different tiers of storage (Premium or Standard) and reclaimPolicy, you can create a **StorageClass**. When you delete a persistent volume, the underlying Azure storage resource is controlled by the reclaimPolicy. The storage can be deleted or kept for future use with a pod.

Lastly, a **persistent volume claim** requests storage of a particular StorageClass, access mode, and size. If no existing resource can fulfill the claim based on the defined StorageClass, the Kubernetes API server can dynamically provision the underlying Azure storage resource.

Scaling

When running applications in a Kubernetes cluster, there will be times when you need to increase or decrease the amount of compute resources to handle a specific workload. In AKS, you can **manually scale pods or nodes** to maintain a fixed amount of resources and cost. You define the replica or node count when manually scaling. Based on that replica or node count, the Kubernetes API schedules the creation of additional pods or the draining of nodes.

If the requirement is scale based on the demand, then use **horizontal pod autoscaler (HPA)** to monitor the resource and automatically scale the number of replicas. By default, the HPA checks the Metrics API every 15 seconds for any changes in the replica count, but the Metrics API retrieves data from the Kubelet every 60 seconds. You need to define the min and max number of replicas that can run when you configure HPA. You also specify the metric on which to base any scaling decisions, such as CPU usage.

Since HPA is for scaling pods, Kubernetes has a **cluster autoscaler** that adjusts the number of nodes in the node pool based on the requested compute resources. By default, the cluster autoscaler checks the Metrics API every 10 seconds. It is common practice to combine horizontal pod autoscalers and cluster autoscalers. The former changes the number of pods in response to application demand, while the latter changes the number of nodes required to accommodate those additional pods.

If you need to rapidly scale your AKS cluster, then integrate it with Azure Container Instances, The virtual nodes component is installed in your AKS cluster and presents ACI as a virtual Kubernetes node. Kubernetes can then schedule pods that run as ACI instances via virtual nodes, rather than pods that run directly on VM nodes in your AKS cluster. You don't need to modify your application using virtual nodes. As the cluster autoscaler deploys new nodes in your AKS cluster, deployments can scale across AKS and ACI with no delay.

Network Connections

Each pod in a Kubernetes cluster is assigned a unique IP address, and pods can communicate with one another using these IP addresses. Kube-proxy is a component in Kubernetes networking that enables efficient and reliable communication between different components in a cluster. Services in Kubernetes provide a consistent endpoint for accessing a collection of pods and can be used to load balance traffic across multiple pods.

The following ServiceTypes are:

- **ClusterIP** - creates an internal IP address that will be used within the cluster.
- **NodePort** - creates a port mapping on the underlying node, allowing the application to be accessed directly using the node's IP address and port.
- **LoadBalancer -** creates a load balancer resource, assigns an external IP address, and connects the requested pods to the backend pool.
- **ExternalName** - creates a unique DNS entry to make application access easier.

When you create a LoadBalancer-type Service, you also create an underlying Azure load balancer resource. The load balancer is configured to distribute traffic at layer 4. With **ingress controllers**, you can distribute application traffic (layer 7) based on the inbound URL.

In AKS, you can create a cluster that uses one of the following network models:

- **Kubenet networking** - network resources are typically created and configured as the AKS cluster is deployed.
- **Azure CNI networking** - AKS cluster is connected to existing virtual network resources and configurations.

Capability	Kubenet	Azure CNI
Deploy cluster in existing or new virtual network	Supported - UDRs manually applied	Supported
Pod-pod connectivity	Supported	Supported

Pod-VM connectivity; VM in the same virtual network	Works when initiated by pod	Works both ways
Pod-VM connectivity; VM in peered virtual network	Works when initiated by pod	Works both ways
On-premises access using VPN or Express Route	Works when initiated by pod	Works both ways
Access to resources secured by service endpoints	Supported	Supported
Expose Kubernetes services using a load balancer service, App Gateway, or ingress controller	Supported	Supported
Default Azure DNS and Private Zones	Supported	Supported
Support for Windows node pools	Not Supported	Supported

In order to filter network traffic in the AKS node, you need to have a **network security groups**. When you create Services, the Azure platform automatically configures any network security group rules that are required. Since the Azure platform creates the NSG, you only need to define the required ports and forwarding as part of your Kubernetes Service manifests.

By default, all pods in an AKS cluster can send and receive traffic without restriction. To improved security, you can use **network policies** to apply traffic filter rules to pods. Network policy is a Kubernetes feature that allows you to manage the flow of traffic between pods in AKS. You can allow or restrict traffic to specific pods by specifying criteria such as assigned labels, namespace, or traffic port. While network security groups are better suited to AKS nodes, network policies are a more cloud-native approach to controlling pod traffic flow. Since pods are created dynamically in AKS clusters, necessary network policies can be implemented automatically.

Reference:
https://learn.microsoft.com/en-us/azure/aks/intro-kubernetes

Azure Resource Manager (ARM)

- A service that allows you to create, update, and delete resources in your Azure account.
- Enables you to manage access control, locks, and tags for your resources after they have been deployed.
- All requests are authenticated and authorized by ARM before being routed to the appropriate Azure service.
- Manage infrastructure using declarative templates and deploy it in a repeatable manner.
- Deploy, manage, and monitor all resources as a group.
- Tag resources to logically organize all the resources in your subscription.
- You can check the costs for a group of resources sharing the same tag.
- Define the dependencies between resources, so they're deployed in the correct order.

Resource groups

- A container that holds related resources.
- You can create a resource group using the Azure Portal, PowerShell, CLI, or an ARM template.
- Each resource can only exist in a single resource group.
- You can add or remove resources to any resource group at any time.
- Allows you to move a resource from one resource group to another.
- Resources from multiple regions can be in one resource group.
- You can give users access to a resource group.
- Resources can interact with other resources in different resource groups.
- A resource group has a location or region, as it stores metadata about the resources.
- When you delete a resource group, it also deletes all of its resources.

ARM templates

- The template is a JSON file with declarative syntax that defines the properties and configuration of your resources. It is divided into the following sections:
    - Parameters - values that allow the same template to be used in multiple environments.
    - Variables - values that can be reused in templates.
    - User-defined functions - customized functions to simplify the template.
    - Resources - define the resources to be deployed.
    - Outputs - values from deployed resources.
- When a template is deployed, ARM converts it into REST API operations.
- You can specify an apiVersion so that you can reuse the template without worrying about breaking changes introduced in later versions.

- To make sure your template adheres to suggested best practices, use an ARM template toolkit (arm-ttk).
- Before deploying the template, you can preview changes using the what-if operation.
- To deploy a template, you can use the following:
    - Azure Portal
    - Azure CLI
    - Azure Cloud Shell
    - PowerShell
    - REST API
    - Button in GitHub repository
- An application can be defined in a single template or divided into a purpose-specific template (modular files). You can also create a parent template that links all the nested templates.
- You can share the template using template specs and manage access using role-based access control (RBAC).
    - Link template - a different template file that is linked from the primary template.
    - Nested template - an embedded template syntax within the main template.
- You can also get the template of an existing resource group by exporting it.
- With Azure Pipelines, you can continuously build and deploy ARM template projects.
- You are only charged for the resources deployed by the ARM template.

Infrastructure as Code, YAML & JSON

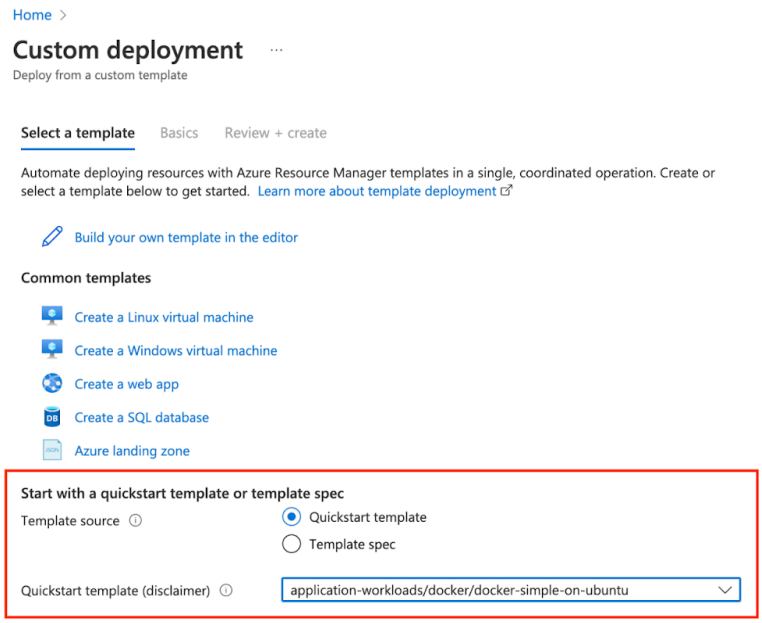
Infrastructure as Code (IaC) is a method of running IT infrastructure that automates, configures, and manages systems and networks using scripts or code. It can work with a variety of file formats, including JSON and YAML. **YAML (YAML Ain't Markup Language)** is a data serialization format that is commonly used in Ansible, Kubernetes, and other tools. While **JSON (JavaScript Object Notation)** is a popular data interchange format that is frequently used in AWS CloudFormation, Terraform, and other tools. Both YAML and JSON are simple to understand and can be used in a variety of IaC tools.

Azure Resource Manager (ARM) templates is a service provided by Microsoft Azure that allows you to provision, manage, and delete Azure resources using declarative syntax. These templates can be used to deploy and manage resources such as virtual machines, storage accounts, and virtual networks in a consistent and reliable manner. To deploy the template, you can use the Azure Portal, Azure CLI, or Azure PowerShell.

In this section, we'll use the Azure Portal to create, deploy, and export resources using ARM templates:

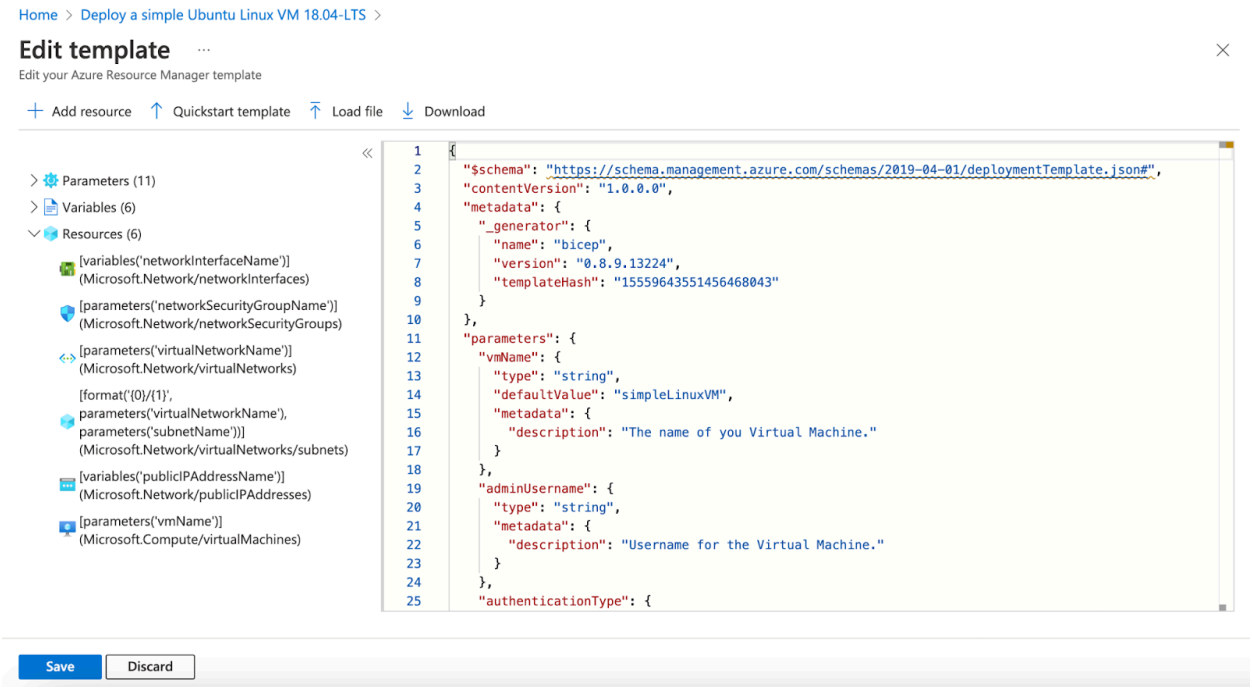
Deploying ARM templates

1. The first thing you need to do is to search for the service *"Deploy a custom template"*. After selecting the service, you'll see the options:
    - Build your own template in the editor
    - Common templates
    - Start with a quickstart template or template spec

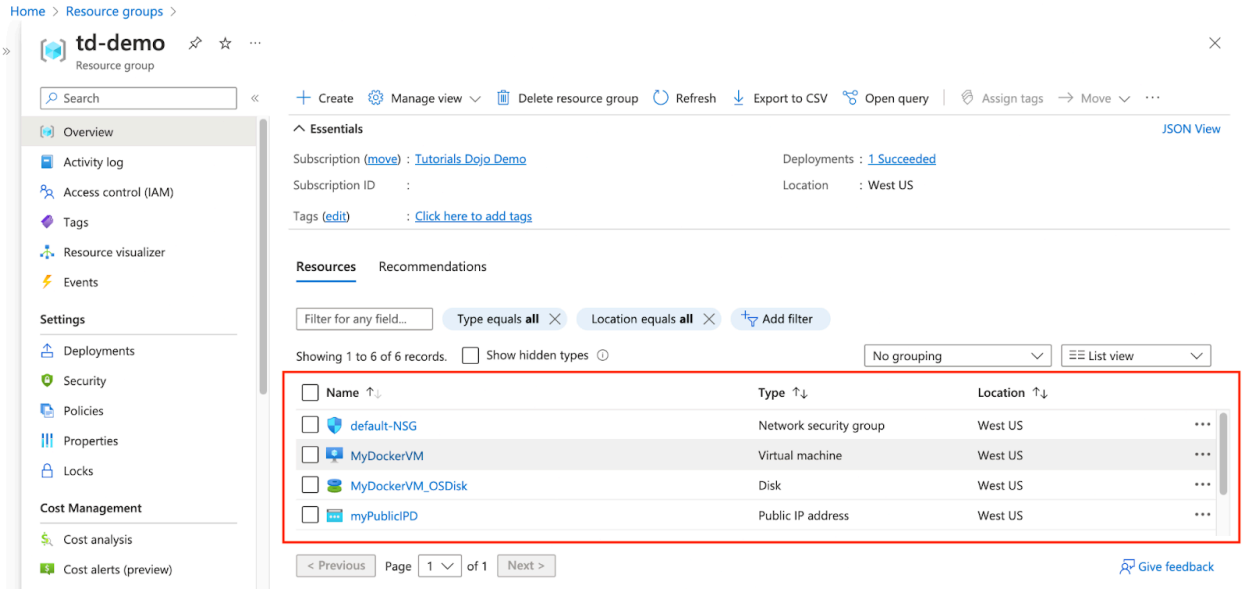


2. Begin with a quickstart template, then type *"docker-simple-on-ubuntu"* and press **Next**. Add the required parameters, such as resource group, username, dns, and password, then click **Create**.

3. You can also modify the template based on your requirements before creating the resources.



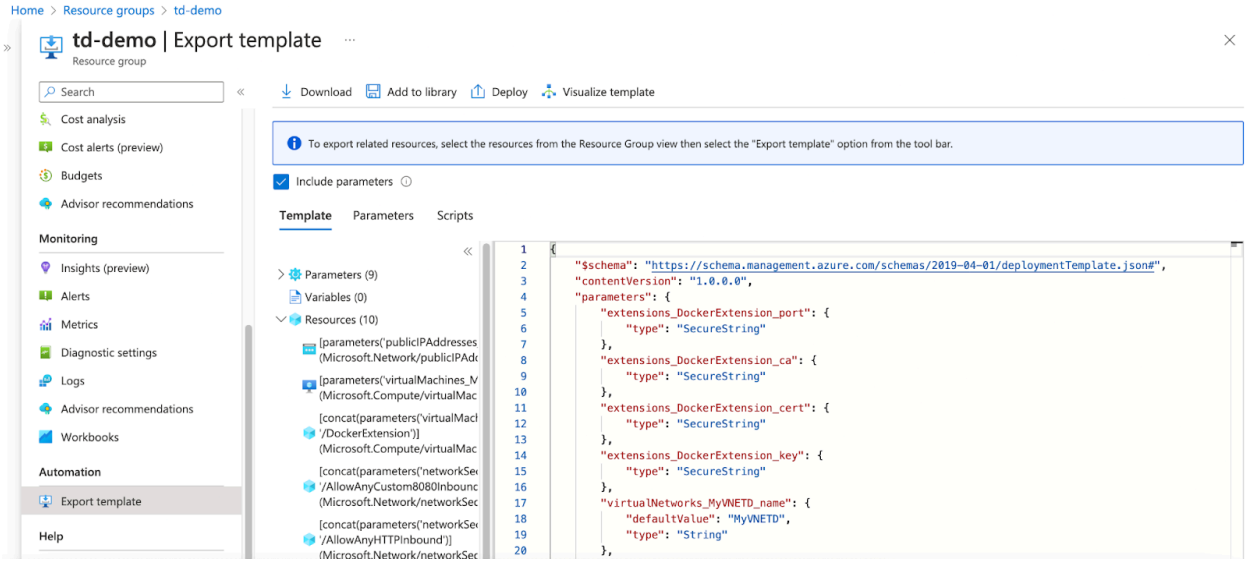
4. Once the deployment is successful, you can go to the selected resource group, and you'll see the created resources from the ARM template.



5. When the Azure resources are no longer needed, you can clean up the resources by deleting the resource group.

Exporting Template

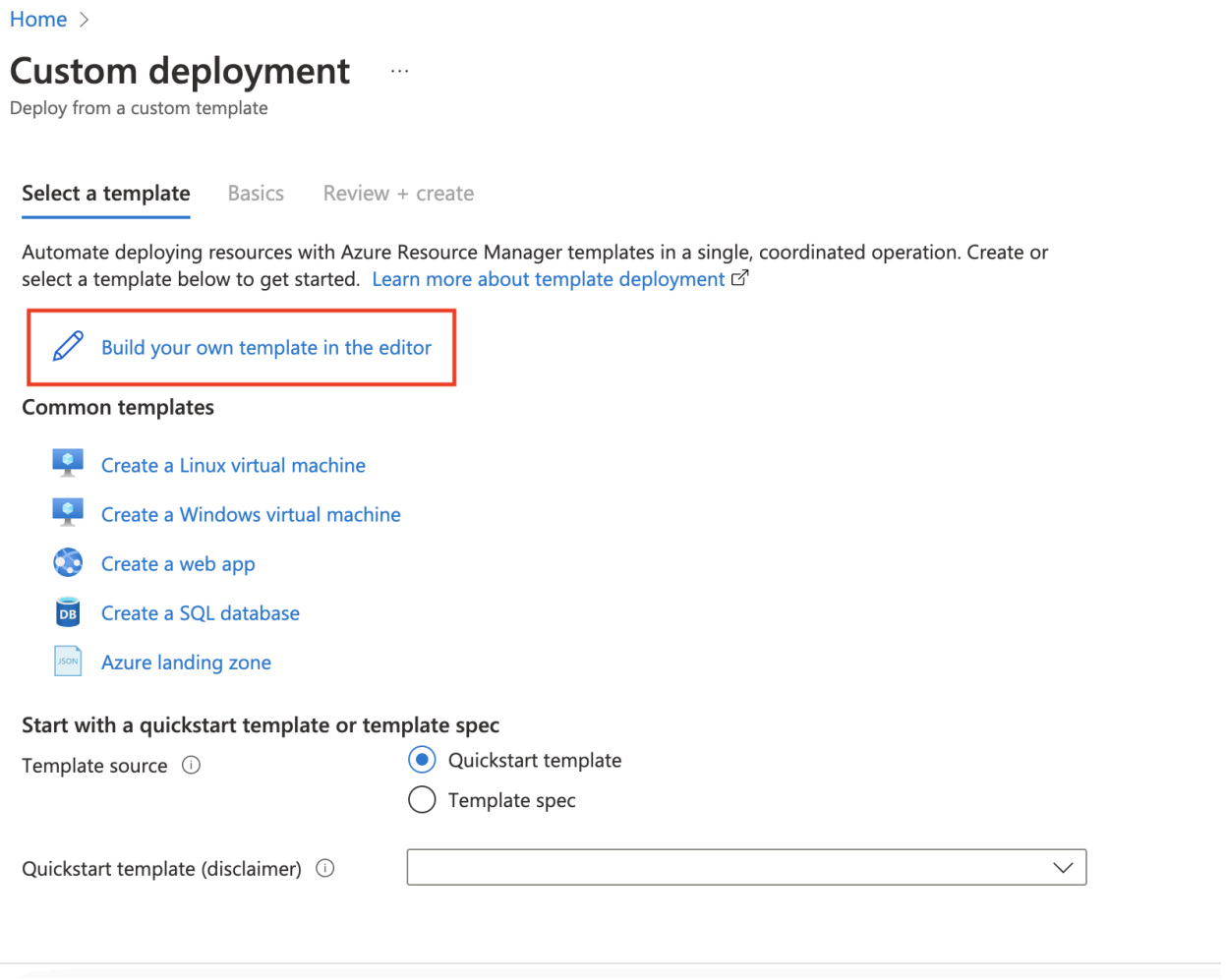
If you want to save the current state of the resource group, scroll down to the Automation section and click Export template, then click the download button.



Before re-using the template for production deployments, you may want to revise it since the template we used is a quickstart template.

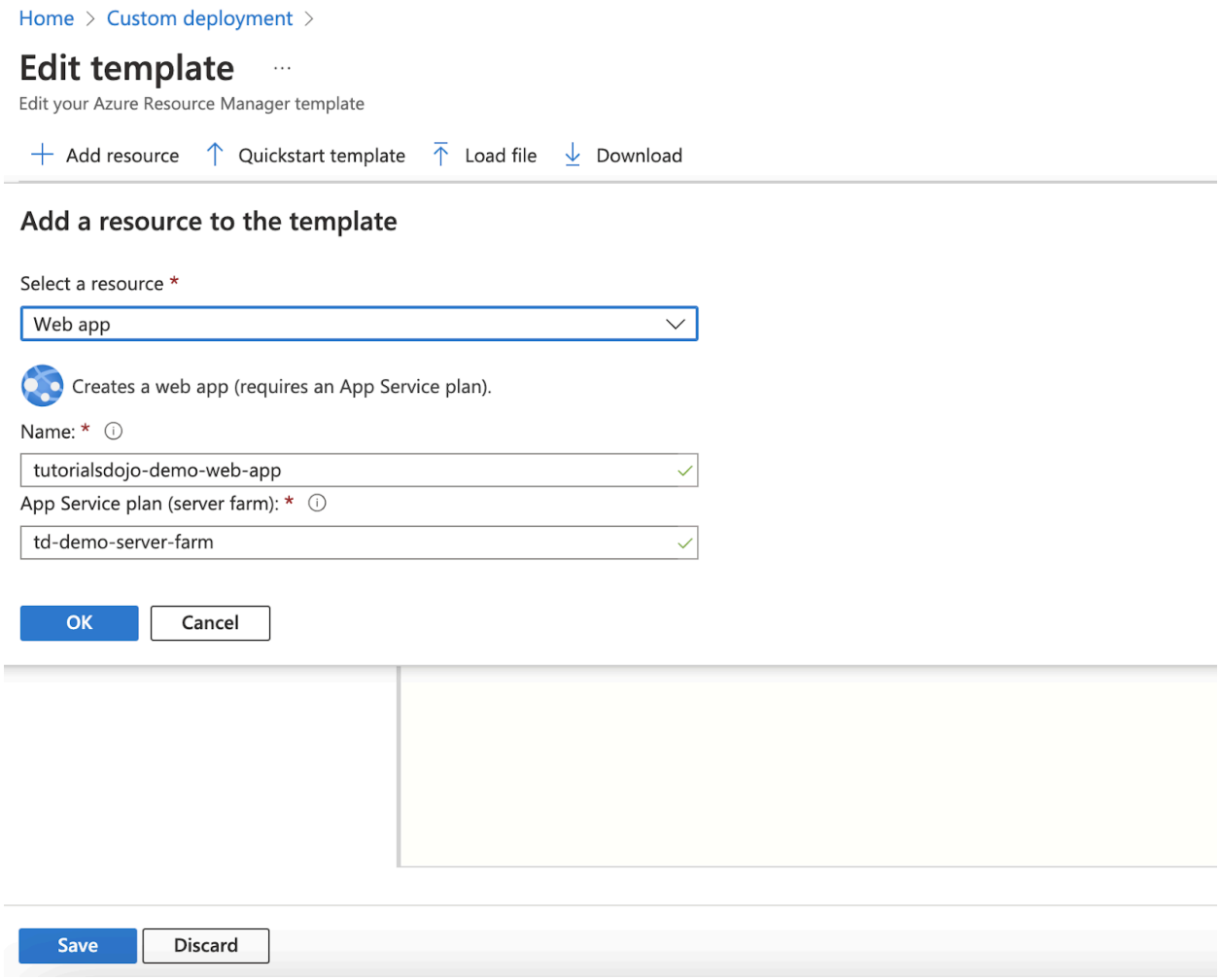
Creating ARM templates

1. Now that you've learned how to deploy using a quickstart template, let's now try creating an ARM template from scratch. Go back to the *"Deploy a custom template"* page and click **Build your own template in the editor**.

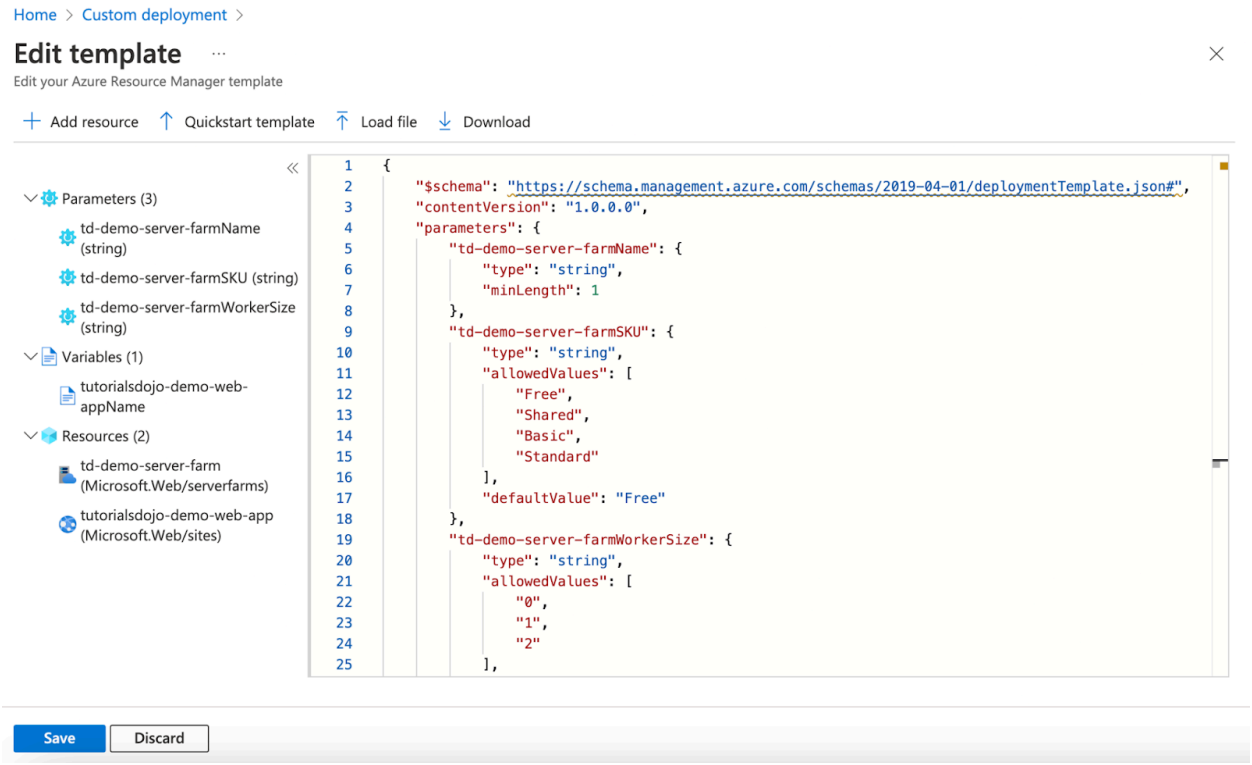


2. Once you are redirected to the editor page, you'll see that there are no parameters, variables, and resources. In this section, we'll create and deploy an Azure web app template.

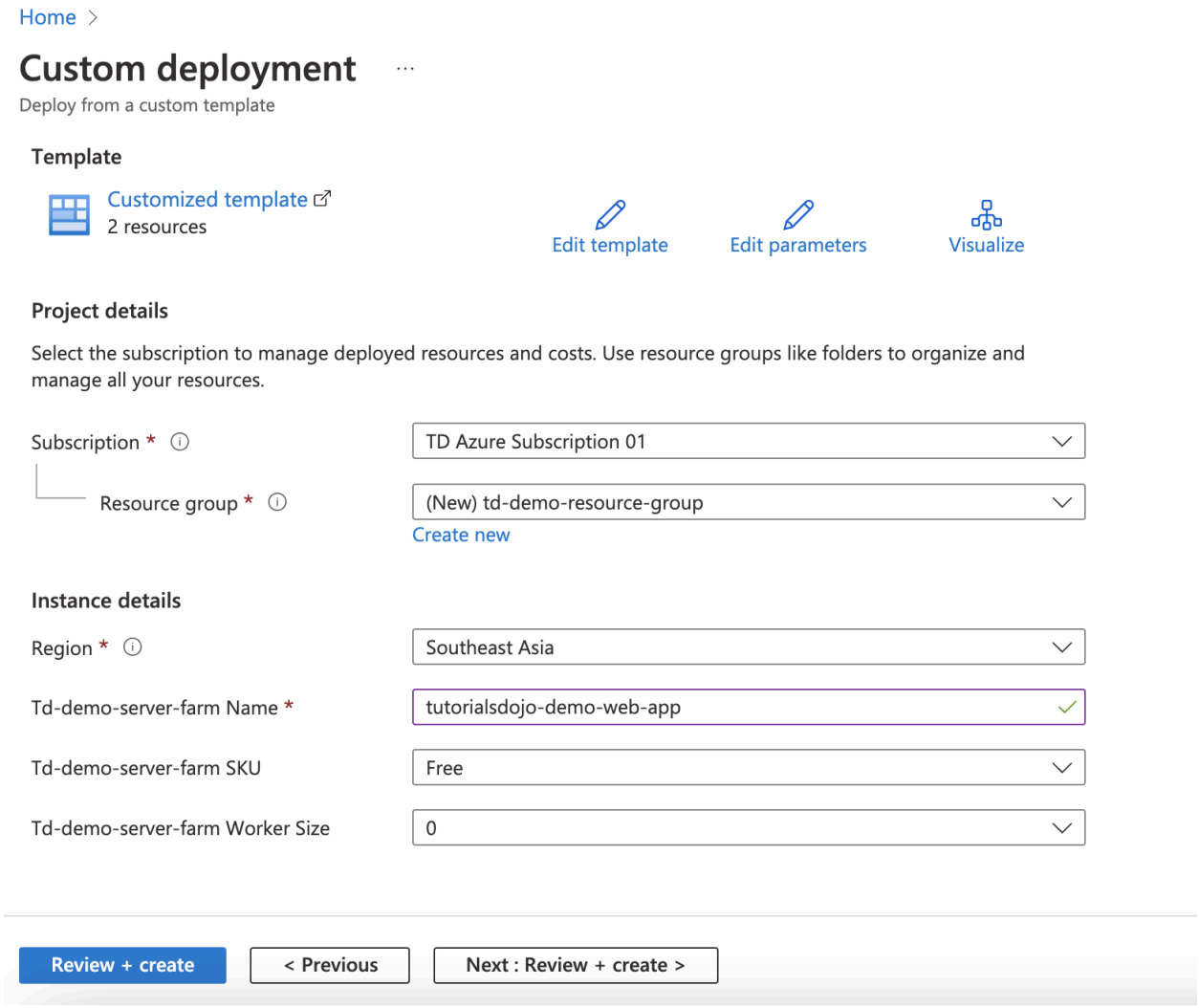
3. Click **Add resource**, select Web app, fill up the remaining fields, and press **OK**.



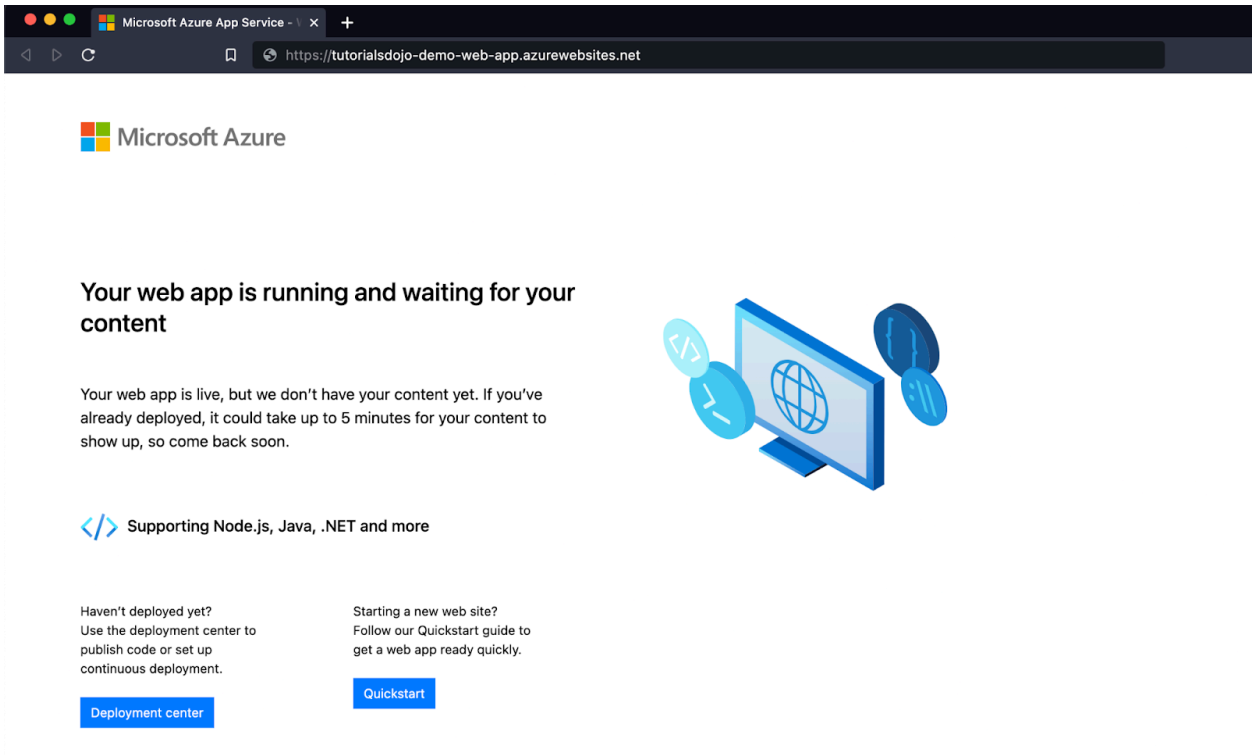
4. In the editor page, you'll notice that a JSON has been created. Feel free to change any of the parameters based on your requirements.



5. If you want to save the template you've created, just click the **Download** button. Now let's save the template and you'll be redirected to the configuration of the resource. Just fill up the required fields and click **Create**.



6. After the deployment is completed, you can go to the resource group and select the web app you've created. To confirm, if the web app is running, you can access the URL generated by the App Service. The format is: *https://**<web-app-name>**.azurewebsites.net/*



7. That's how simple it is to create and deploy a custom ARM template for your own project. Again, if you no longer need the resource, don't forget to delete the resource group to avoid unexpected billing in your account.

Azure Storage Accounts

Types of Storage Accounts

1. **General-purpose v2 accounts**
    - Supports Data Lake Gen2, Blobs, Files, Disks, Queues, Tables.
    - Delivers the lowest per-gigabyte capacity prices for Azure Storage.
2. **General-purpose v1 accounts**
    - Supports Blobs, Files, Disks, Queues, Tables.
    - You can upgrade a general-purpose v1 account to a general-purpose v2 account with no downtime and without copying the data.
    - You can use general-purpose v1 accounts since the General-purpose v2 accounts and Blob storage accounts only support the Azure Resource Manager deployment model.
    - If you don't need a large capacity for transaction-intensive or significant geo-replication bandwidth, GPv1 is a suitable choice.
3. **BlockBlobStorage accounts**
    - Provides low, consistent latency and higher transaction rates.
    - Upgrading a Blob storage account to a general-purpose v2 account has no downtime and you don't need to copy the data.
    - It doesn't support hot, cool, and archive access tiers.
    - You can use BlockBlobStorage for storing unstructured object data as block blobs or append blobs.
4. **FileStorage accounts**
    - Only support file shares.
    - Offers IOPS bursting.
5. **BlobStorage accounts**
    - Only supports block and append blobs.
    - BlobStorage account offers standard performance, while the BlockBlobStorage account supports premium performance.

Storage Account Type	Services	Performance Tiers	Access Tiers	Replication Options	Deployment Model	Encryption
General-purpose V2	Blob, File, Queue, Table, Disk, and Data Lake Gen2	Standard, Premium	Hot, Cool, Archive	LRS, GRS, RA-GRS, ZRS, GZRS, RA-GZRS	Resource Manager	Encrypted
General-purpose V1	Blob, File, Queue, Table, and Disk	Standard, Premium	N/A	LRS, GRS, RA-GRS	Resource Manager, Classic	Encrypted
Block Blob Storage	Blob (block and append blobs only)	Premium	N/A	LRS, ZRS	Resource Manager	Encrypted
FileStorage	File only	Premium	N/A	LRS, ZRS	Resource Manager	Encrypted

BlobStorage	Blob (block and append blobs only)	Standard	Hot, Cool, Archive	LRS, GRS, RA-GRS	Resource Manager	Encrypted

Storage Account Endpoint

A storage account in Azure gives your data a unique namespace. Every object you save to Azure Storage has a unique address that includes your account name. The endpoints for your storage account are formed by the account name and the Azure Storage service endpoint.

- **Blob storage:** https://*tutorialsdojo*.blob.core.windows.net
- **Table storage:** https://*tutorialsdojo*.table.core.windows.net
- **Queue storage**: https://*tutorialsdojo*.queue.core.windows.net
- **Azure Files**: https://*tutorialsdojo*.file.core.windows.net
- **Azure Data Lake Storage Gen2:** https://*tutorialsdojo*.dfs.core.windows.net

Storage Account Redundancy

Azure Storage keeps several copies of your data to protect it from both planned and unexpected events, such as transient hardware failures, network or power outages, and massive natural disasters. Even in the event of a breakdown, redundancy guarantees that your storage account fulfills its availability and durability goals. The greater level of redundancy, the more expensive the cost of replication.

Primary Region Redundancy

The data in the storage account is always replicated three times in the primary region. Azure offers two options on how your data will be replicated:

- **Locally Redundant Storage (LRS)**
  - A low-cost redundancy strategy.
  - Your data is copied synchronously three times within the primary region.
- **Zone-Redundant Storage (ZRS)**
  - Redundancy for high availability.
  - The data is copied synchronously across three Azure availability zones in the primary region.

Secondary Region Redundancy

The data in the storage account is copied to a secondary region. This redundancy option is mainly used for applications that require high durability since the data is durable even if there's a complete regional outage or disaster in the primary region. Also, the paired secondary region is determined based on the selected primary region, and can't be changed.

- **Geo-Redundant Storage (GRS)**
  - Cross-regional redundancy
  - In the primary region, data is synchronously copied three times and then asynchronously copied to the secondary region.
- **Geo-Zone-Redundant Storage (GZRS)**
  - Redundancy for both high availability and maximum durability
  - Data is copied synchronously across three Azure availability zones in the primary region, then copied asynchronously to the secondary region.

Secondary Region Redundancy with Read Access

The data is replicated to another location in the secondary region, and it is only available for read access. For example, if the primary region becomes unavailable, your data is still available to be read at all times in the secondary region. You can only enable read access to the following redundancy options:

- **Read-Access Geo-Redundant Storage (RA-GRS)**
  - The data is copied synchronously in the primary region and secondary region.
- **Read-Access Geo-Zone-Redundant Storage (RA-GZRS)**
  - The data is copied synchronously across three availability zones in the primary region, then synchronously copied to the secondary region.

Parameter	LRS	ZRS	GRS/RA-GRS	GZRS/RA-GZRS
Durability of objects over a given year	at least 99.99999999% (11 9's)	at least 99.99999999% (12 9's)	at least 99.99999999999999% (16 9's)	at least 99.99999999999999% (16 9's)
Read requests availability	At least 99.9% (99% for cool access tier)	At least 99.9% (99% for cool access tier)	- At least 99.9% (99% for cool access tier) for GRS	- At least 99.9% (99% for cool access tier) for GZRS

			- At least 99.99% (99.9% for cool access tier) for RA-GRS	- At least 99.99% (99.9% for cool access tier) for RA-GZRS
Write requests availability	At least 99.9% (99% for cool access tier)	At least 99.9% (99% for cool access tier)	At least 99.9% (99% for cool access tier)	At least 99.9% (99% for cool access tier)
Number of copies of data maintained on separate nodes	Three copies within a single region	Three copies across separate availability zones within a single region	- Three in the primary region and - Three in the secondary region	- Three across separate availability zones in the primary region - Three locally redundant copies in the secondary region

The availability and durability of your data during outages is entirely dependent on the type of redundancy configured for your storage account.

Outage Scenario	LRS	ZRS	GRS/RA-GRS	GZRS/RA-GZRS
Available if a node went down within a data center?	✓	✓	✓	✓
Available if the entire data center (zonal or non-zonal) went down?	-	✓	✓	✓
Available on region-wide outage in the primary region?	-	-	✓	✓

Has read access to the secondary region if the primary region is unavailable	-	-	✓ (with RA-GRS)	✓ (with RA-GZRS)

Storage Encryption

- All storage accounts are encrypted using Storage Service Encryption (SSE) for data at rest.
- The type of encryption used is 256-bit AES encryption.
- Storage redundancy options support the encryption of data.
- You can use a Microsoft-managed key, a customer-managed key, and a customer-provided key to manage the encryption of your data.

Key management parameter	Microsoft-managed keys	Customer-managed keys	Customer-provided keys
Encryption/decryption operations	Azure	Azure	Azure
Azure Storage services supported	All	Blob storage, Azure Files1,2	Blob storage
Key storage	Microsoft key store	Azure Key Vault or Key Vault HSM	Customer's own key store
Key rotation responsibility	Microsoft	Customer	Customer
Key control	Microsoft	Customer	Customer

Reference:
https://docs.microsoft.com/en-us/azure/storage/common/storage-account-overview

Azure Blob Storage

Blob Storage Resources

The blob storage is composed of three types of resources:

1. **Storage account** which provides a unique namespace in Azure for your data. The name of the storage account is also the endpoint of your blob storage. For example, if the storage account name is "tutorialsdojo-manila" then the endpoint is: *http://tutorialsdojo-manila.blob.core.windows.net*.
2. **Container** - organizes a set of blobs that are similar to a directory in a file system. A storage account can have an unlimited number of containers, and each container can hold an unlimited number of blobs.
3. **Blob** - a collection of binary data stored as a single entity. Examples of blobs are images, videos, audios, log files, backups, and many more. Azure supports three types of blobs:
    - **Block**
        i. Made up of blocks of data that can be managed individually
        ii. Store binary and text data up to 4.7 TB.
        iii. Preview larger block blobs up to 190.7 TiB.
    - **Append**
        i. Optimized for append operations.
        ii. Ideal for logging data from virtual machines.
    - **Page**
        i. Store random-access files up to 8 TB in size.
        ii. Store virtual hard drive (VHD) files and serve as disks for Azure VMs.

It's also important to understand the features available in a blob storage:

Versioning - when you enable blob versioning, you can easily restore an earlier version of a blob to recover your data. If you disable the versioning of the blob, it does not delete existing blobs, versions, or snapshots.

Snapshots - a read-only version of a blob that was taken at a given point in time. The snapshots persist until they are explicitly deleted.

Object Replication - copies block blobs asynchronously between a source Storage account and a destination account. A source account can have up to two destination accounts. But there can be no more than two source accounts in the destination account.

Static Website - serve your static website directly from a storage container named *$web*. You can grant read-only access in your resources with a public access level. If you want to configure a custom domain endpoint for your website, you can use Azure CDN.

Access Tiers

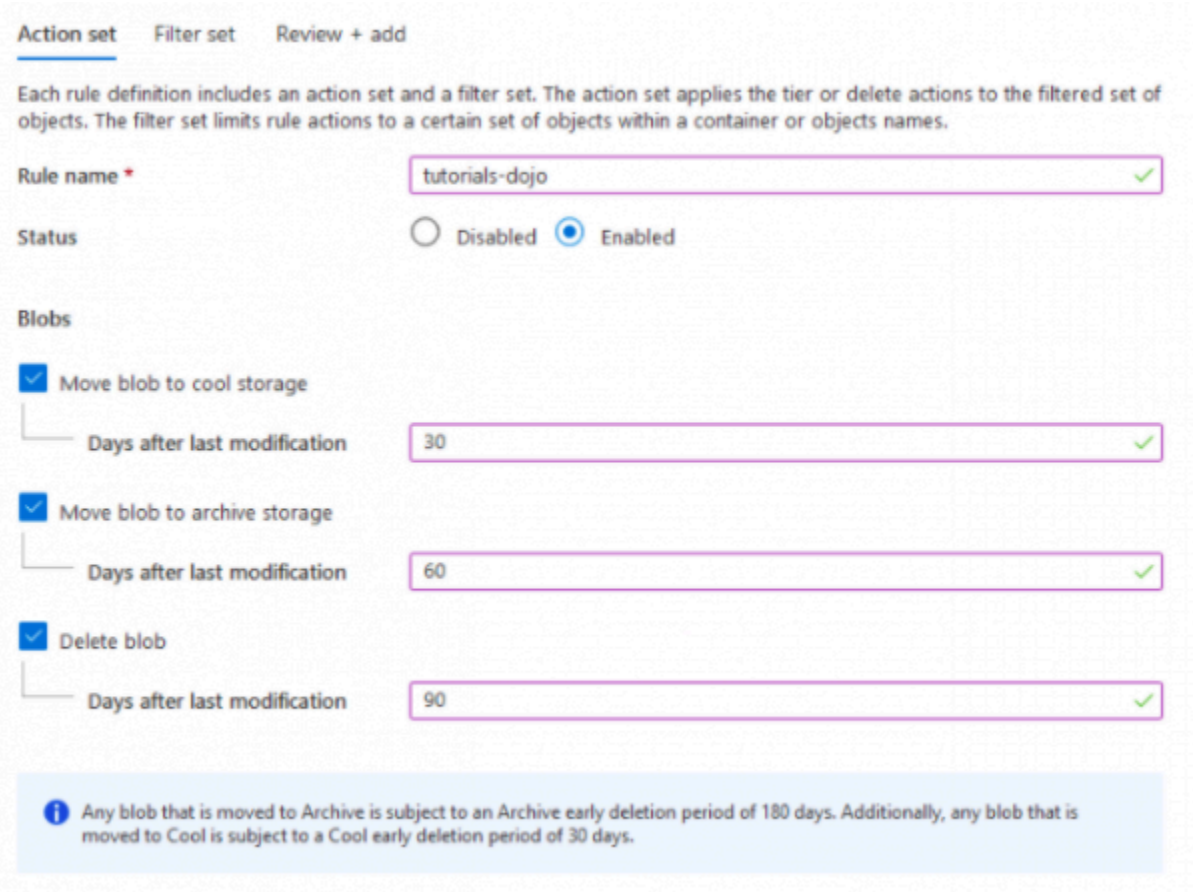
1. **Hot**
    - Highest storage costs, but lowest access costs
    - Store data that is accessed frequently
    - By default, new storage accounts are created in the hot tier
2. **Cool**
    - Lower storage costs, but higher access costs
    - Store data that is infrequently accessed (at least 30 days)
    - You can use a cool access tier for short-term backup.
3. **Archive**
    - Lowest storage costs, but the highest retrieval costs
    - Store data that is rarely accessed (at least 180 days)
    - Data needs to be stored for a long time.

	Hot tier	Cool tier	Archive tier
Availability	99.90%	99%	Offline
Availability (RA-GRS reads)	99.99%	99.90%	Offline
Usage charges	Higher storage costs, lower access, and transaction costs	Lower storage costs, higher access, and transaction costs	Lowest storage costs, highest access, and transaction costs
Minimum storage duration	N/A	30 days	180 days
Latency	milliseconds	milliseconds	hours
Use case	Data that is staged for processing and eventual migration to cool access tier	Large data sets for future processing, backup for disaster recovery	Data that needs to be preserved and hardly ever accessed

When using access tiers, there are important tiering concepts that you also need to know like **account-level tiering** and **blob-level tiering**. Let's describe first what is account level tiering. If a blob doesn't have an explicitly assigned tier, it infers the tier from the storage account access tier settings. Take note that you can only set hot and cool access tiers as the default account access tier since the archive tier can only be set at the object level.

The next tiering concept allows you to upload data to the access tier of your choice. This means that you can select your preferred access tier then change the blob access tier as your usage patterns change. All tier change requests are processed immediately, and the changes between hot and cool tiers are done instantly. But if you move a blob from the archive tier into another tier, this would take several hours. This process is called **rehydrating**.

If you want to transition your data to the appropriate access tiers, you can configure a lifecycle rule in the blob lifecycle management. The tiers that you can transition are: blob to cool storage, blob to archive storage, and delete the blob at the end of its lifecycle.



In blob-level tiering, when a blob is uploaded or moved to a different tier, you are charged at the new tier's rate immediately. Let's say you move a blob to a cooler tier, the operation billed to you is a **write operation** to the destination tier. But if you moved to a hotter tier, the operation billed to you is a read operation from the source tier. There are also charges for early deletion if a blob is moved out of the cool or archive tier. To conclude how tier changes are billed, let's take a look at the table below:

	Write Charges (Operation + Access)	Read charges (Operation + Access)
Set Blob Tier	Hot -> Cool	Archive -> Cool
	Hot -> Archive	Archive -> Hot
	Cool -> Archive	Cool -> Hot

Transfer Data with AzCopy

- AzCopy is a command-line utility that allows you to transfer blobs or files to or from a storage account.
- You can use Microsoft Entra ID and SAS tokens to provide authorization credentials.
- These are the tasks that you can do using AzCopy:
    - Upload files
    - Download blobs and directories
    - Copy blobs, directories, and containers between accounts.
    - Synchronize local storage
- You can run AzCopy on Windows, Linux, and macOS.
- AzCopy method of authorization
    - **Azure Blob storage** - Microsoft Entra ID and Shared Access Signature
    - **Azure Files** - Shared Access Signature only
- A **shared access signature (SAS)** is a uniform resources identifier (URI) that grants restricted access to your storage account. You can share the URI to grant users temporary access to a specific set of permissions. There are three types of shared access signatures:
    - **User Delegation SAS** - provides access to storage accounts using Microsoft Entra ID credentials. This SAS is only applicable to Blob storage.
    - **Service SAS** - grants access to one Azure storage service (Blob, Queue, Table, Files) using a storage account key.
    - **Account SAS** - provides access to one or more storage services using a storage account key.
- A shared access signature can be in different forms:
    - **Ad hoc SAS** - start time, expiry time, and permissions are specified in the URI. The three types of SAS can be an ad hoc SAS.
    - **Service SAS with stored access policy** - the stored access policy is defined on a resource container (blob container, file share, table or queue). The policy can be used to manage constraints to multiple SAS.

Import/Export Data to and from Azure

The **Azure Import/Export service** allows you to import large amounts of data to Blob storage and Files by shipping the disk drives to an Azure data center. You can also use this service to transfer data from a Blob storage to disk drives and then ship it to your on-premises environment. The data from one or more drives can be imported to a Blob storage or Files.

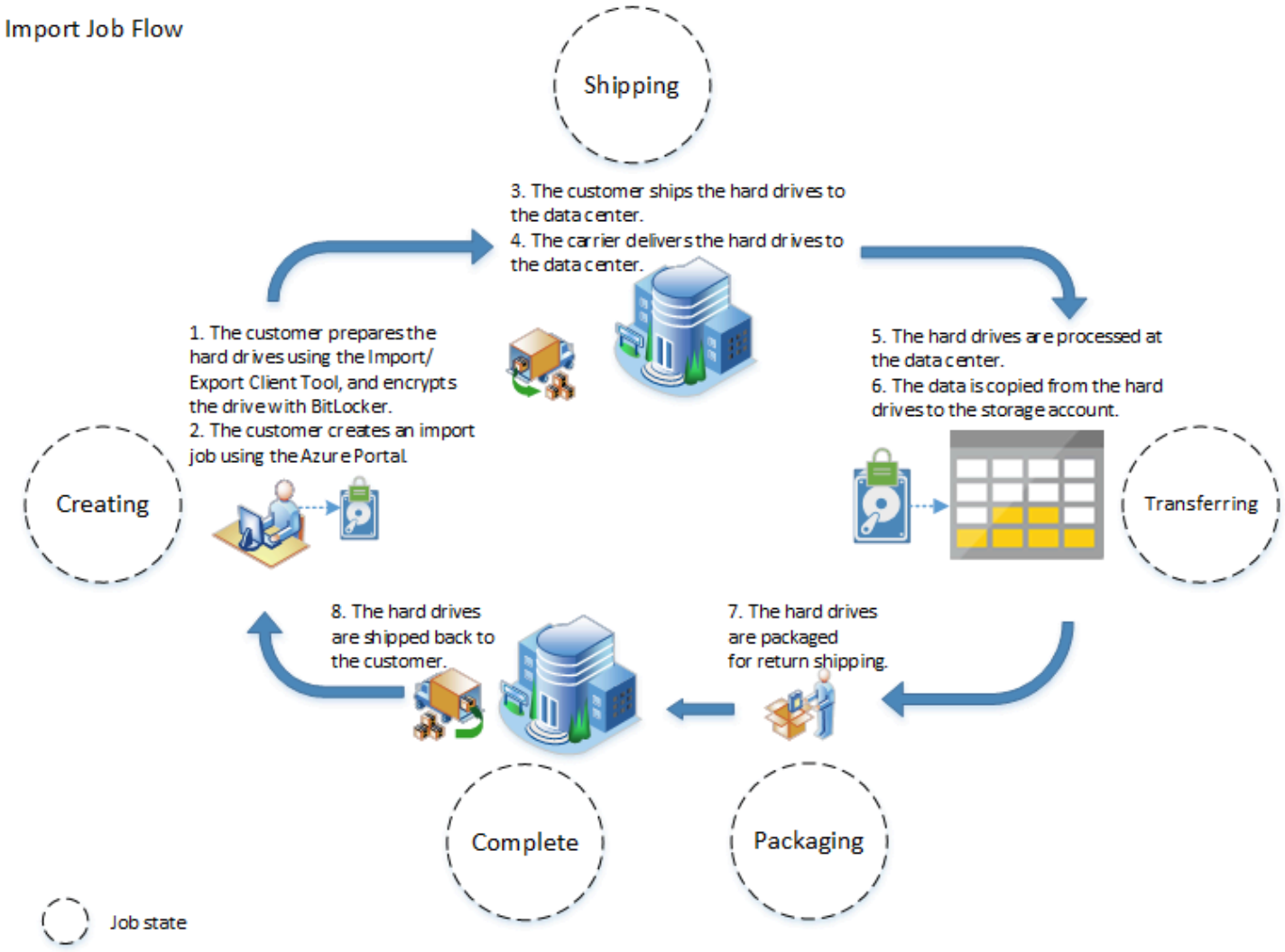
A **WAImportExport tool** is a command-line tool that you can use to prepare your disk drives that are shipped for import. By using this tool, it will be easier to copy your data to the drive. The data on the drive is encrypted with AES 256-bit BitLocker. To protect your BitLocker key, you can use an external key protector. You can also use this tool to generate the drive journal files used during import creation and identify the number of drives needed for export jobs.

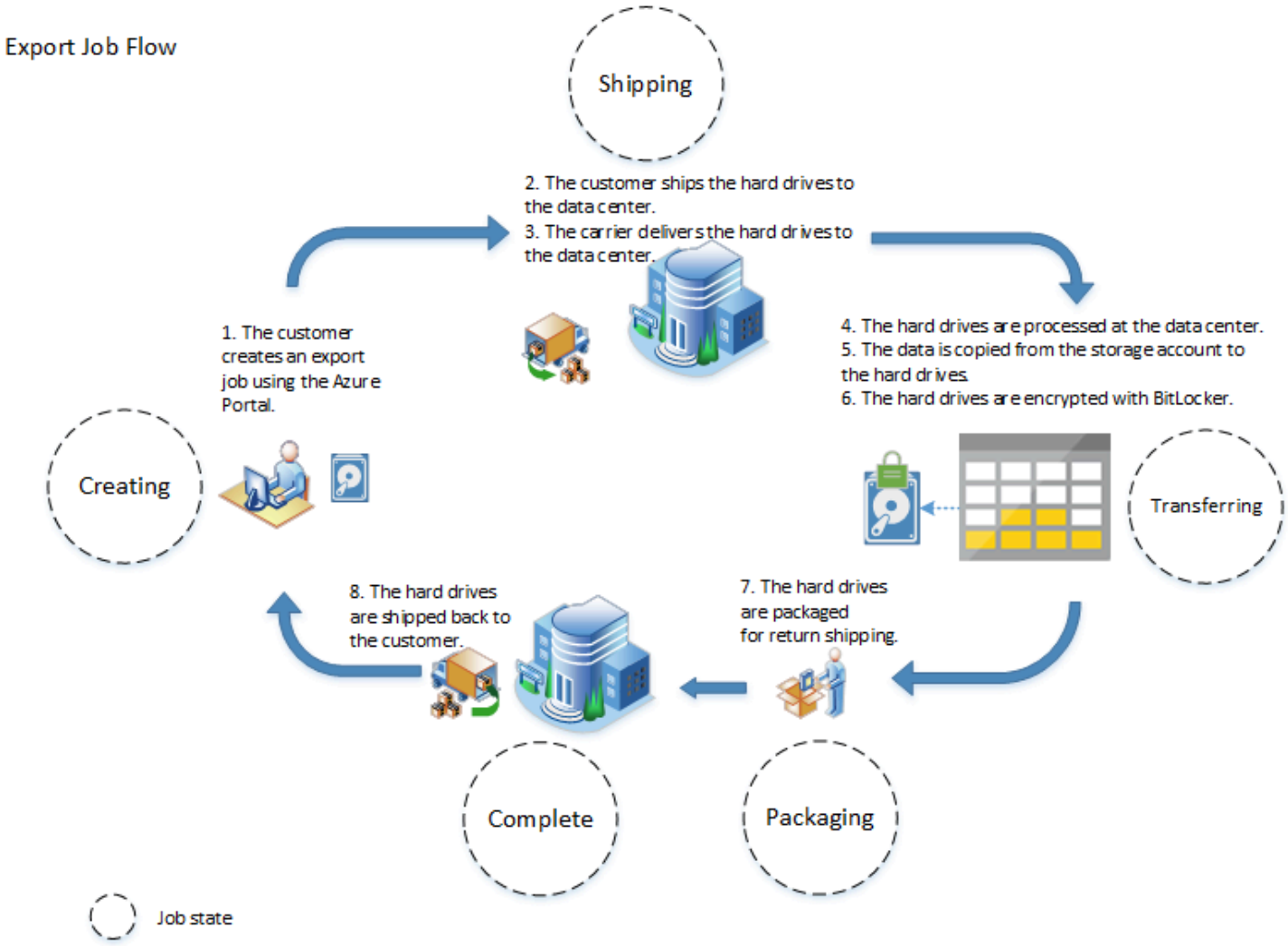
The disk drives that you can ship to the Azure data center can be solid-state drives (SSDs) or hard disk drives (HDDs). When you create an import job, the disk drives you ship contain data. But when you create an export job, the drives you ship to the data center are empty disk drives.

An **import** job allows you to import data into Azure Blobs or Azure files, whereas the **export** job allows data to be exported from Azure Blobs. For an import job, you ship drives containing your data. When you create an export job, you ship empty drives to an Azure datacenter. In each case, you can ship up to 10 disk drives per job.

Use this service on the following scenarios:

- **Data migration to the cloud** - transfer large amounts of data to Azure in a timely and cost-effective manner.
- **Content distribution** - send data to your clients' websites in a timely manner.
- **Backup** - backup your on-premises data and save it on Azure Storage.
- **Data recovery** - recover a significant quantity of data and have it delivered to your on-premises location.


Import Job Flow
Shipping
3. The customer ships the hard drives to the data center.
4. The carrier delivers the hard drives to the data center.
1. The customer prepares the hard drives using the Import/ Export Client Tool, and encrypts the drive with BitLocker.
2. The customer creates an import job using the Azure Portal
Creating
5. The hard drives are processed at the data center.
6. The data is copied from the hard drives to the storage account.
Transferring
8. The hard drives are shipped back to the customer.
7. The hard drives are packaged for return shipping.
Complete
Packaging
Job state



References:

https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-storage-tiers
https://docs.microsoft.com/en-us/azure/storage/common/storage-use-azcopy-v10
https://docs.microsoft.com/en-us/azure/import-export/storage-import-export-service

Azure Files

Storage Tiers

- **Premium file shares (SSD)**
  - High performance & low latency, within single-digit milliseconds for most IO operations.
  - For IO-intensive workloads.
- **Standard file shares (HDD)**
  - Reliable performance for IO workloads which are less latency-sensitive.
- If you created either a premium or a standard file share, you cannot automatically convert it to the other tier.

Detail	Premium	Standard
Billing model	Provisioned Billing Model, pay for how much storage you provision rather than how much storage you actually ask for.	Pay-As-You-Go Model, the bill will increase if you use (read/write/mount) the Azure file share more.
Redundancy options	It is available for locally redundant (LRS) and zone redundant (ZRS) storage.	It is available for locally redundant, zone redundant, geo-redundant (GRS), and geo-zone redundant (GZRS) storage.
Maximum size of file share	Provisioned for up to 100 TiB.	5 TiB by default, 100 TiB for locally redundant or zone redundant storage accounts.
Regional availability	File shares are not available in each region, but zone redundant support is available in a smaller subset of regions.	Available in every Azure region.

Azure File Sync

- Transform an on-premises (or cloud) Windows Server into a quick cache of your Azure file share.
- Use Azure File Sync agent to synchronize files from a server to an Azure file share.
- To create sync groups, you need to deploy a Storage Sync Service.
- A sync group defines the sync relationship between a cloud endpoint and a server endpoint.
    - **Cloud endpoint** – represents an Azure file share and multiple server endpoints.
    - **Server endpoint** – a path registered on the Windows Server.
    - When you make changes to your cloud endpoint or server endpoint, your files are automatically synced to your sync group's remaining endpoints.
    - When you make a change directly to the cloud endpoint, Azure files must first detect it via a change detection job, which only happens once every 24 hours.
    - A change detection job enumerates all the files in the file share and compares it to the sync version of that file. When the change detection job determines that there are changes, Azure File sync will initiate a sync session.
- The sync group you created should only have one cloud endpoint.
- A sync group may have server endpoints with different Active Directory memberships, even if they are not domain-joined.
- The storage accounts used for Azure Files deployments are:
    - General purpose version 2 (GPv2) storage accounts
    - FileStorage storage accounts
- You can use cloud tiering to cache frequently accessed files locally on the server.
- The service supports interop with DFS Namespaces (DFS-N) and DFS Replication (DFS-R).
    - DFS-N allows you to group shared folders located on multiple servers into one or more logically structured namespaces.
    - DFS-R enables you to replicate folders across multiple servers and sites.
- Azure File Sync has three layers of encryption:
    - Encryption at rest (Windows Server)
    - Encryption in transit
    - Encryption at rest (Azure file share)

Reference:
https://docs.microsoft.com/en-us/azure/storage/files/storage-files-introduction

Azure Virtual Network

Components of a Virtual Network

- Before you can create a virtual network, you must first select a **subscription**, **resource group** and **region**. Keep in mind that when choosing a region, you must consider the following factors: service availability, costs, compliance, and data residency.
- When you've finished setting up the basic details, you can move on to the selection of **IPv4 and IPv6 (dual-stack) address space**. A virtual network can have one or more address prefixes in CIDR notation.
- A **subnet** is a range of IP addresses in your VNet. You can also create multiple subnets and launch Azure resources into a specified subnet. Use a **public subnet** for resources that need to connect to the Internet, and a **private subnet** for resources that do not require Internet access.
- To allow the resources from a subnet to have an outbound connection, you must deploy a **NAT gateway**. By assigning a NAT gateway to a subnet, you can simplify the connectivity to the internet without having a load balancer or a public IP address attached to a virtual machine.
- If you need private IP addresses in your virtual network to reach the endpoint of an Azure service without the need for a public IP address, you can create a **service endpoint**. These endpoints will help you secure your critical Azure resources by only allowing traffic from your virtual network.
- Lastly, you can use **tags** to easily identify and categorize your resources.
- Once you're satisfied with the configuration of your virtual network, proceed with the launch. Wait for your virtual network to complete its preparations, and then you should be able to deploy resources in the subnet.

Network Security Group (NSG) and Application Security Group (ASG)

A **network security group** controls the inbound and outbound traffic of Azure resources. While an **application security group** allows you to define a group of virtual machines and define network security policies based on those groups.

- The rules are processed from lowest to highest numbers.
- You can set a number between 100 and 4096.
- The rules can be applied to both inbound or outbound traffic.
- You can allow or deny incoming or outgoing traffic.
- When you create a network security group, Azure assigns default security rules for inbound and outbound traffic.
- NSG can be attached to a subnet or a network interface. Refrain from attaching a network security group to both subnet and network interface.

Rules are processed in priority order, with lower numbers processed before higher numbers since lower numbers have higher priority. Once traffic matches a rule, processing stops. As a result, any rules that

exist with lower priorities (higher numbers) that have the same attributes as rules with higher priorities are not processed.

Virtual Network Peering

VNet peering enables you to connect two virtual networks seamlessly. Plan accordingly before initiating a peer, and ensure that your VNet address ranges do not overlap with one another.

There are two types of peering:

1. **Virtual Network Peering** - connect virtual networks in the same Azure region.
2. **Global Virtual Network Peering** - connect virtual networks across different Azure regions.

Here are some of the advantages of using VNet peering:

- Provides a high-bandwidth and low-latency connection between resources in different virtual networks.
- The resources in one virtual network can communicate with resources in another virtual network.
- Enables you to transmit data between virtual networks across Azure subscriptions, Azure regions, Microsoft Entra tenants ID, and deployment models.
- There is no downtime in either virtual network when creating the peering or after the peering is created.
- Traffic between virtual networks is private. This means that the communication between the virtual networks does not require the use of the public Internet, gateways, or encryption.

Reference:
https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-overview

Azure Load Balancer

Components of a Load Balancer

- A load balancer distributes incoming network traffic across multiple targets.There are two types of load balancers:
  - **Public load balancer** - allows outbound connections for your virtual machines.
  - **Internal load balancer** - controls the flow of traffic inside your private virtual network.
- A group of VMs or instances in a VM scale set serving the incoming request is called a **backend pool**.
- Determine the health status of backend pool instances with **health probes**.
  - Health probe down behavior – if the probes in a backend pool fail, it will stop receiving traffic until it starts passing health probes again.
- Use Azure Monitor to check the metrics, alerts, and resource health of Azure Load Balancer.
- High Availability (HA) ports enable load balancing on all ports of TCP and UDP protocols.
- With multiple frontends, you can load balance services on multiple ports and multiple IP addresses.
- SLA guarantees that two or more healthy VMS will always be available.
- The load balancer tiers are: **Basic** and **Standard**
- Standard load balancer **availability zones**:
  - **Zonal** = single zone
  - **Zone-redundant** = multiple zones

Details	Basic Load Balancer	Standard Load Balancer
Backend pool size	Supports up to 300 instances.	Supports up to 1000 instances.
Backend pool endpoints	A single availability set for VMs or VM scale set.	A single virtual network for any VMs or VM scale sets.
Health probes	TCP, HTTP	TCP, HTTP, HTTPS
Health probe down behavior	TCP connections stay alive on an instance probe down. All TCP connections terminate when all probes are down.	TCP connections stay alive on an instance probe down and on all probes down.
Availability Zones	Not available	Zone-redundant and zonal frontends for inbound and outbound traffic.

Diagnostics	Azure Monitor logs	Azure Monitor multi-dimensional metrics
HA Ports	Not available	Available for Internal Load Balancer
Secure by default	Open by default. Network security group optional.	Closed to inbound flows unless allowed by a network security group. Please note that internal traffic from the VNet to the internal load balancer is allowed.
Outbound Rules	Not available	Declarative outbound NAT configuration
TCP Reset on Idle	Not available	Available on any rule
Multiple frontends	Inbound only	Inbound and outbound
Management Operations	60-90+ seconds typical	Most operations < 30 seconds
SLA	Not available	99.99%

Load Balancing Algorithm

A load balancing rule distributes the incoming traffic to the resources in the backend pool. Health probes can determine which VMs in the backend pool can receive the load-balanced traffic. The load-balancing decision is based on the following tuple connection:

1. **Source IP address and port**
2. **Destination IP address and port**
3. **Protocol**

Session persistence maintains the traffic from a client to the same virtual machine.

1. **None** – any virtual machine can handle successive requests from the same client.
2. **Client IP** – the same virtual machine will handle successive requests from the same client IP address.

3. **Client IP and protocol** – the same virtual machine will handle successive requests from the same client IP address and protocol combination.

Reference:
https://docs.microsoft.com/en-us/azure/load-balancer/load-balancer-overview

Azure DNS

Public and Private DNS

The **Public DNS** handles the domains, DNS zones, DNS records, and record sets. A DNS zone is used to host the DNS records of a particular domain. An example of a domain is "tutorialsdojo.com". It may contain a number of DNS records such as "mail.tutorialsdojo.com" (for a mail server) and "www.tutorialsdojo.com" (for a website). A record set is a collection of DNS records in a zone that have the same name and type. Here is an example of a record set:

```
www.tutorialsdojo.com.    3600    IN    A    12.238.154.93
www.tutorialsdojo.com.    3600    IN    A    12.238.157.186
```

On the other hand, a **Private DNS** allows you to manage and resolve domain names in a virtual network without adding a custom DNS. The records in a private DNS zone are not reachable through the Internet and DNS resolution against a private DNS zone is only possible from virtual networks linked to it.

DNS Record Types

- **A** - maps the host to an IPv4 address.
- **AAAA** - maps the host to an IPv6 address.
- **CNAME** - indicates that the name it refers to is not an alias. In other words, all other records that you use are all aliases.
- **MX** - used for mail exchange and maps mail requests to your mail servers.
- **PTR** - maps the IP address to a domain or host name.
- **SOA** - stores important information about a domain or zone.
- **SRV** - helps you specify server locations.
- **TXT** - often associated with text strings in a domain name and used for verifying a domain name.

Import/Export a DNS Zone File

The importing and exporting of zone files is only applicable using the Azure CLI. If a zone file is imported, it generates a new zone in Azure Private DNS. If the zone already exists, the zone file's record sets must be merged with the existing record sets.

- The CLI command to import a zone file is:
  `az network dns zone import -g **<resource-group-name>** -n **<zone-name>**.com -f **<filename>**.txt`
- If you need to export a zone file, the command is:
  `az network dns zone export -g **<resource-group-name>** -n **<zone-name>**.com -f **<file-name>**.txt`

Reference:
https://docs.microsoft.com/en-us/azure/dns/dns-overview

Azure VPN Gateway

VPN Gateway Connections

1. **Site-to-Site** - creates a secure connection from your on-premises network to an Azure virtual network.

2. **VNet-to-VNet** - connection automatically routes to the updated address space, if you updated the address space on the other VNet.

3. **Point-to-Site** - establish a connection to your virtual network from a remote location.

In an **active-active configuration**, each Azure VPN gateway instance will establish S2S VPN tunnels, and the traffic will be routed to multiple tunnels. For **active-passive configuration**, the standby instance would only take over if a disruption happens on the active instance.

Details	Site-to-Site	Point-to-Site
Supported Services	Cloud Services and Virtual Machines	Cloud Services and Virtual Machines
Bandwidths	Typically < 1 Gbps aggregate	Based on the gateway SKU
Protocols	IPsec	Secure Sockets Tunneling Protocol (SSTP), OpenVPN and IPsec
Routing	Support both PolicyBased (static routing) and RouteBased (dynamic routing VPN)	RouteBased (dynamic)
Connection resiliency	active-passive or active-active	active-passive
Use case	Dev / test / lab scenarios and small scale production workloads for cloud services and virtual machines	Prototyping, dev / test / lab scenarios for cloud services and virtual machines

VPN Types

1. **Policy-based gateway** implements a policy-based VPN. The policy-based VPNs are used to encrypt and direct packets to IPsec tunnels. The policy or traffic selector is defined as an access list in the VPN configuration. You cannot change a policy-based VPN to a route-based VPN and vice versa.

2. **Route-based gateway** - implements a route-based VPN. The route-based VPNs use routes in the routing table to direct packets to tunnel interfaces. Tunnel interfaces can encrypt and decrypt packets. The policy or traffic selector are configured as wild cards (any-to-any).

Reference:
https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-about-vpngateways

Microsoft Entra ID

Managing Users, Groups, Roles and Devices

1. Users

- You can create a new user in your organization or a guest user.
- By enabling Multi-Factor Authentication, you provide additional security by requiring the user a second form of authentication. The additional forms that can be used with Microsoft Entra ID MFA are:
    - Microsoft Authenticator app
    - OATH hardware token
    - SMS
    - Voice call
- You can also perform the following bulk operations:
    - Bulk create
    - Bulk invite
    - Bulk delete
    - Bulk restore
    - Download users
- Self-service password reset enables users to manage their passwords from any device, at any time, and from any location.
- In the device settings, you can change the maximum number of devices per user.
- You can assign licenses to multiple users or groups to allow them to use the licensed Microsft Entra ID services. Licenses are applied per tenant, and you can't transfer them to other tenants.

2. Groups

- A collection of users, devices, groups, and service principals.
- You can easily manage access to your resources by creating an Microsoft Entra group.
- A user can belong to multiple groups.
- Groups do not have security credentials.
- Group Types:
    - **Security** – it contains users, devices, groups, and service principals as its members. The users and service principals are the owners of this group.
    - **Microsoft 365** – it contains users as its members. Both the users and service principals can be owners of this group.
- Membership type:
    - **Assigned** – manually add users to be members of the group.
    - **Dynamic user** – automatically add and remove members using the dynamic membership rules.

- **Dynamic device** – automatically add and remove members using the dynamic group rules.

3. Roles

- With external identities, you can allow users outside your organization to sign in using an external identity provider like Facebook and Google.
- Administrative roles can be used to grant access to Microsft Entra ID and other Microsoft services. There are two types of role definitions:
  - **Built-in roles** – it has a fixed set of permissions.
  - **Custom roles** – you can select permissions from a preset list. To create a custom role, you need to have an Microsoft Entra ID P1 or P2 plan.
- A Microsft Entra resource that can be a container for other Microsft Entra resources is called an administrative unit. It can only contain users and groups.

4. Devices

- **Microsoft Entra registered**
  - The devices registered are personally owned devices (bring your own device or mobile device). These devices are signed in with a personal Microsoft account.
  - A Mobile Device Management (MDM) helps you enforce configurations like storage must be encrypted, password complexity, and up-to-date security software.
  - Key capabilities:
    - Single sign-on (SSO) to cloud resources.
    - Conditional access when enrolled in Microsoft Intune or via App protection policy.
    - Enables phone sign-in with Microsoft Authenticator app.
- **Microsoft Entra joined**
  - The devices and accounts are owned by an organization. It only exists in the cloud.
  - Microsoft Entra join is primarily used for organizations that do not have an on-premises Windows Server AD infrastructure.
  - Key capabilities:
    - SSO to both cloud and on-premises resources.
    - Conditional access through MDM enrollment and MDM compliance evaluation.
    - Self-service password reset and Windows Hello PIN reset on the lock screen.
    - Enterprise State Roaming across devices.
- **Microsoft Entra hybrid joined**
  - The devices and Active Directory Domain Services account are owned by an organization. It exists both in the cloud and on-premises resources.
  - You can implement hybrid joined devices if you have an existing on-premises AD footprint and you want to benefit from the capabilities provided by Microsoft Entra.

    - Key capabilities:
        - SSO to both cloud and on-premises resources.
        - Conditional access through Domain join or through Microsoft Intune if co-managed.
        - Self-service password reset and Windows Hello PIN reset on the lock screen.
        - Enterprise State Roaming across devices.
- If you register your application to use Microsoft Entra ID, the users in your organization can do the following:
    - Get an identity for their application that is recognized by Microft Entra ID.
    - Get secrets/keys that the application will use for authentication.
    - Create a custom name and logo for your application.
    - Apply Microsoft Entra authorization (RBAC and oAuth)
    - Declare the necessary permissions for the application.
- With application proxy, you can provide SSO, and remote access for web apps hosted on-premises.

Reference:
https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-whatis

Azure RBAC

How Permissions are Enforced

Before we dive in on how to control access to resources using Azure role-based access control, let's first go through what can you do with this authorization system. Azure RBAC allows you to manage user's access to Azure resources, including what they can do with those resources and what areas they can access. For example, if you want to allow a group of database administrators to only manage SQL databases, then you have to assign an Azure role to that group.

Attaching a role definition to a user, group, service principal, and managed identity to grant access to a particular scope is called **role assignment**. You can assign a role using Azure Portal, PowerShell, CLI, SDKs, or REST APIs.

A role assignment is composed of three elements:

1. **Security Principal** – an object representing a user, group, service principal, and managed identity that requests access to Azure resources.
    a. **User** - an entity that you create in Microsoft Entra ID to represent a person who interacts with Azure services.
    b. **Group** - a collection of users created in Microsoft Entra ID.
    c. **Service Principal** - an identity that applications use to gain access to different Azure resources.
    d. **Managed Identity** - an identity in Microsoft Entra ID that is managed by Azure. There are two types of managed identities:
        - **System-assigned** - when you enable this option, an identity will be created in Microsoft Entra ID and this identity will be tied to the lifecycle of the service instance. Therefore, if the resource is deleted, the identity will automatically be deleted by Azure.
        - **User-assigned** - since system-assigned is tied with the service instance, the user-assigned identity is managed separately from the resource. You can also assign the created user-assigned managed identity to one or more instances of an Azure service.
2. **Role Definition** – a list of permissions that can be performed, such as read, write and delete. You can either use the Azure built-in roles or construct your own custom roles if the provided roles don't meet the specific needs of your organization.
    a. **Built-in roles** - since it takes time to create custom roles, you can use the built-in roles managed by Azure to easily grant the permissions needed by the principal.
    b. **Custom roles** - this role's permissions are defined by you. It can also be shared between subscriptions that trust the same Microsoft Entra ID.

3. **Scope** – set of resources to which access applies. When you assign a role, you can define a scope to further limit the actions that are permitted. Scopes are structured in a parent-child relationship. You can assign roles at any of these levels:
    a. Management Group
    b. Subscription
    c. Resource Group

Remember that role assignments are transitive for groups. If a user is a member of Group A, and Group A is a member of Group B with a role assignment, then the user will inherit the role assignment's permissions. For multiple overlapping role assignments, the effective permissions are the sum of your role assignments. For example, a user has a Contributor role at the subscription scope and a Reader role in a resource group. The sum of these two permissions is effectively the Contributor role for the subscription.

You can also attach a set of deny actions to a user, group, service principal, or managed identity at a particular scope using **deny assignments**. Take note that deny assignments take precedence over role assignments. In other words, deny assignments can restrict users from performing a specified action even if it has a role assignment.

Different Types of Roles

Classic Subscription Administrator Roles - have full access to an Azure subscription like managing resources using the Azure Portal, Resource Manager API, and the classic deployment model APIs. The three classic subscription administrative roles are:

1. **Account Administrator** - this role is the billing owner of the Azure subscription. It can manage subscriptions and billings in the account. You can only have 1 Account Administrator per Azure account.
2. **Service Administrator** - you can only have 1 Service Administrator per Azure subscription. In new subscriptions, the Account Administrator also serves as the Service Administrator. This role has full access to the Azure portal and it can assign users with a Co-Administrator role.
3. **Co-Administrator** - you can only create 200 Co-Administrator per Azure subscription. This role has the same privileges as the Service Administrator, but it can't change the association of subscriptions to Azure directories. A user with this role can only assign a Co-Administrator role to other users.

Azure Roles – provides fine-grained access management of Azure resources. The following are the four fundamental Azure built-in roles:

1. **Owner** - provides full access to all Azure resources. It can also delegate access to other users.
2. **Contributor** - allows the user to create and manage all types of resources in Azure. The role can also create a new tenant in Microsoft Entra ID, but it cannot grant access to other users.

3. **Reader** - a user with this role can only view Azure resources.
4. **User Access Administrator** - this role has permission to manage user access to all types of resources.

Built-in Roles	Read	Grant	Create, Update, Delete
Owner	✓	✓	✓
Contributor	✓		✓
Reader	✓		
User Access Administrator		✓	

Microsoft Entra Roles – provide access to manage Microsoft Entra resources in a directory such as create users, assign administrative roles to others, manage licenses, reset passwords, and manage domains. The important Microsoft Entra built-in roles are:

1. **Global Administrator** - this role can manage access to all the administrative features in Microsoft Entra. It can assign administrator roles to the users in your organization and reset the password of users or administrators in your account.
2. **User Administrator** - allows the user to create and manage different types of users and groups in Azure. Manage support tickets and monitor service health. Also, this role can only change the passwords of users and administrators.
3. **Billing Administrator** - this role has permission to make purchases, monitor service health, manage subscriptions, and support tickets in Azure.

You can also create custom roles, but you need to upgrade your Microsoft Entra ID to P1 or P2.

	Microsoft Entra Roles	Azure Roles
Definition	Manage access to Microsoft Entra resources: Users, Groups, Billing, Licensing, Application Registration, etc.	Manage access to Azure resources: Virtual Machine, Database, Storage, Networking, etc.
Custom Role	Supported	Supported
Scope	The scope is only at the tenant level.	The scope can be specified at multiple levels: Management group,

		Subscription, and Resource group
Role Information	It can be accessed through Azure Admin Portal, Microsoft 365 Admin Center, Microsoft Graph, and Azure AD PowerShell.	It can be accessed through Azure Portal, CLI, PowerShell, Resource Manager templates, and REST APIs.

Role Definition Structure

If you are planning to create your own Azure custom role, it is important to understand how roles are defined. The following structure is displayed when using Azure PowerShell:

```
{
"Name": ,
"Id": ,
"IsCustom": ,
"Description" ,
"Actions": [ ],
"NotActions": [ ],
"DataActions": [ ],
"NotDataActions" [ ],
"AssignableScopes" [ ]
}
```

The following structure is displayed when using Azure Portal, CLI, or the REST API:

```
{
"roleName": ,
"name": ,
"type": ,
"description" ,
"actions": [ ],
"notActions": [ ],
"dataActions": [ ],
"notDataActions" [ ],
"assignableScopes" [ ]
}
```

- **Name / roleName** - the display name of the created role.

- **Id / name** - the auto-generated unique ID of the role.
- **IsCustom / type** - indicates whether the role is a CustomRole (true) or a BuiltInRole (false).
- **Description / description** - the description of the custom role.
- **Actions / actions** - an array of strings that defines the management operations that can be performed by the role.
- **NotActions / notActions** - an array of strings that are excluded from the allowed Actions.
- **DataActions / dataActions** - an array of strings that defines the data operations that can be performed to the data within an object.
- **NotDataActions / notDataActions** - an array of strings that are excluded from the allowed DataActions.
- **AssignableScopes / assignableScopes** - an array of strings that defines the scope that the role is available for assignment. Take note that you can only define one management group in a custom role.

The operations that you can perform are specified in a string:

- ***** - a wildcard allows you to apply all the operations that match the string.
- **read** - allows read (GET) operations.
- **write** - allows write (PUT or PATCH) operations.
- **action** - allows custom operations like restarting a virtual machine (POST).
- **delete** - allows delete (DELETE) operations.

Reference:
https://docs.microsoft.com/en-us/azure/role-based-access-control/overview

Azure Policy

Policy Components

In order to implement governance for resource consistency, regulatory compliance, security, cost and management, you need to enforce a set of rules by creating a policy. But before you create a policy, you need to understand the following components:

1. The **policy definition** is created in a JSON format. It is used to describe resource compliance conditions and the effect to take if the conditions are met. You can assign a built-in policy or define your own rules by creating a custom policy.
2. When creating a policy definition, it is also a good practice to make it flexible or reusable by reducing the number of policy definitions. **Policy parameters** allow you to define parameters when assigning the policy definition. A parameter is composed of a name and optionally, a given value. Let's say, you define a parameter for a policy titled "location". Then you can give it different values, such as "East US" or "West US" when assigning a policy.
3. Once the policy is created, you can proceed to **policy assignment**. This is where the policy will take place within a specific scope. The scope refers to management groups, subscriptions, and resource groups. Policy assignments are inherited by child resources. This means that if the policy is applied to a resource group, it is also applied to all the resources in that resource group.
4. The **initiative definition** is a collection of policy definitions that you can assign. This will help you simplify a group of policies as one single item. One example is, create an initiative titled “Enable Monitoring in Azure Security Center”. The goal of this initiative is to monitor all the available security recommendations. There are three policy definitions under this initiative: Monitor unencrypted SQL Database, Monitor OS Vulnerabilities, and Monitor missing endpoint.
5. Lastly, once the policy is assigned, it will now evaluate for the compliance state. You can find the non-compliant resources in the Compliance tab.

Policy Definition Structure

We will be breaking down what constitutes a policy definition and what conditions you can add to your policies. The structure is as follows:

```
{
   "properties": {
      "displayName": " ",
      "description": " ",
      "mode": " ",
      "metadata": {},
      "parameters": {},
      "policyRule": {}
```

```
    }
}
```

- **Display Name** and **Description** - helps you identify the policy definition and when it's used.
- **Mode** - it determines which resource types are evaluated for a policy definition. There are two types of modes:
    a. all - evaluates subscription, resource group and all resource types.
    b. Indexed - evaluates only the resource types that support tags and location.
- **Metadata** - although this property is optional, it is useful if you need to store information about the policy definition. The common metadata properties are:
    a. version (string)
    b. category (string)
    c. preview (boolean)
    d. deprecated (boolean)
- **Parameters** - enables you to reuse the policy in different scenarios by using different values. To understand it better, think of parameters like a field on a form (name, address, city, state). These parameters are always the same, but their values change depending on who fills out the form. The following properties are used in the policy definition:
    a. name - the name of your parameter.
    b. type - it can be string, integer, float, boolean, array, object, or datetime.
    c. metadata - the subproperties used by the Azure portal to display the following information:
        ■ description - defines what the parameter is used for.
        ■ displayName - the name shown in the portal
        ■ strongType (optional) - provides a multi-select list of options in the Azure Portal. Use *Get-AzResourceProvider* to determine whether a resource type is valid for strongType.
        ■ assignPermissions (optional) - allows you to assign permissions outside the assignment scope.
    d. defaultValue (optional) - if parameter value is not defined, it sets the default value.
    e. allowedValues (optional) - an array of values that the parameter will accept.
- **Policy Rule** - this is where you apply the logical operators and conditions. The rules consist of “if” and “then” statements.
    a. The supported logical operators are:
        ■ not
        ■ allOf
        ■ anyOf
    b. The conditions that you should be aware of are:
        ■ less, lessOrEquals, greater, and greaterOrEquals - an error is thrown if the property type does not match the condition type.
        ■ like and notLike - you only need to provide one wildcard (*) value.

- match and notMatch - the condition is case sensitive
  - # to match a digit
  - ? for letter
  - . to match any character
- matchInsensitively and notMatchInsensitively - case-insensitive alternatives

Policy Effects

A policy definition has a single effect and this will define what happens when the policy rule is evaluated to match. The effects operate differently depending on whether they are applied to a new resource, an updated resource, or an existing resource.

The supported effects in a policy definition are:

- **Append** – add additional fields to the requested resource.
- **Audit** – a warning event for a non-compliant resource.
- **AuditIfNotExists** – audit the resources when the condition is met.
- **Deny** – prevents the request before being sent to the Resource Provider.
- **DeployIfNotExists** – if the condition is met, it allows you to execute a template deployment.
- **Disabled** – allows you to disable a single assignment, rather than disabling all assignments under that policy.
- **Modify** – manage tags of resources (add, update, or remove).
- The order of evaluation is managed by the Resource Provider and forward the results back to Azure Policy. When a resource fails to meet the designated governance controls of Azure Policy, this order will prevent unnecessary processing.
- **Disabled** - check this effect first to see if the policy rule should be evaluated.
- **Append** and **Modify** - since these two could alter the request, making a change could prevent an audit or deny effect from being triggered.
- **Deny** - by evaluating this effect before audit, double logging of an unwanted resource is avoided.
- **Audit** - the last evaluation

Reference:
https://docs.microsoft.com/en-us/azure/governance/policy/overview

Azure Monitor

Log Analytics

- All log data obtained by Azure Monitor shall be stored in a Log Analytics workspace
- Query simple to advanced logs.
- The data is retrieved from a workspace using a log query written using Kusto Query Language (KQL).
- The queries that you can run are:
    a. **Table-based queries** – the query organizes log data into tables.
    b. **Search queries** – use this query if you need to find a specific value in your table.
    c. **Sort and top** – to display the results in a particular order, you must sort the preferred column. To get the latest records in the entire table, you can use top.
    d. **Where** – this operator allows you to add a filter to a query. You can use different expressions when writing filter conditions.
    e. **Time filter in query** – you can define a specific time range by adding the time filter to the query.
    f. **Project and Extend** – project allows you to select specific columns, and extend will add additional columns.
    g. **Summarize** – you can identify a group of records and apply aggregations using the summarize operator.
- If the query includes workspaces in 20 or more regions, your query will be blocked from running.
- Log Analytics results are limited to a maximum of 10,000 records.
- With a log analytics agent, you can collect logs and performance data from virtual or physical devices outside Azure.
- A log analytics agent cannot send data to Azure Monitor Metrics, Azure Storage, or Azure Event Hubs.

Alert Rules and Action Groups

Action rules help you define or suppress actions at any Azure Resource Manager scope (Azure subscription, resource group, or target resource). It has various filters that can help you narrow down the specific subset of alert instances that you want to act on.

An **action group** is a collection of notification preferences defined by the owner of an Azure subscription. Azure Monitor and Service Health alerts use action groups to notify users that an alert has been triggered. Various alerts may use the same action group or different action groups depending on the user's requirements.

To understand these concepts better, let's see the example below:

Your company has an Azure subscription that contains the following resources:

Name	Type	Location
tdvm	Virtual machine	East US
tdstorage	Storage account	West US
tdapp	App Service	East US
tdsql	SQL server	North Central US

You are instructed to monitor the storage account and configure an SMS notification for the following signals.

Signal name	Signal type	Users that will be notified
Availability	Metric	User 1, User 2, and User 3
Transactions	Metric	User 2 and User 3
Create/Update Storage Account	Activity Log	User 1, User 2, and User 3
Regenerate Storage Account Keys	Activity Log	User 3

How many alert rules and action groups should you create?

The requirement in the scenario is to identify how many alert rules and action groups should be created. Based on the given signal types, you should create four alert rules. Take note that you need to create one alert rule per signal type.

For the action groups, you only need to create 3 action groups because the users that will be notified for Availability and Create/Update Storage Account are the same (User 1, User 2, and User 3). Remember that action groups are created for each unique set of users that will be notified.

Reference:
https://docs.microsoft.com/en-us/azure/azure-monitor/overview

Azure Network Watcher

Network Connectivity Monitoring

The designated tool to monitor the communication between a VM and an endpoint is called **connection monitor**. An endpoint can be an IPv4 address, uniform resource identifier (URI), the fully qualified domain name (FQDN), or a virtual machine. The minimum, average, and maximum latency detected over time are displayed in the connection monitor. After you've determined the latency for a connection, you can reduce the latency by relocating Azure resources to different Azure regions.

Diagnosing Virtual Machine Network Traffic

When you deploy a virtual machine, default security rules are applied to the virtual machine that allow/deny network traffic. Since Azure applies these rules by default, you might override the default rules and may prevent the virtual machine from communicating with other resources. To diagnose network traffic problems to or from a virtual machine, you can use **IP flow verify**.

This tool will help you specify the source and destination of an IPv4 address, protocol, port and the traffic direction. After the test is conducted, it informs you if the connection succeeds or fails. If the connection fails, IP flow verify will tell you which security rule allowed or denied the communication.

Verify a TCP connection from a Virtual Machine

The tool for diagnosing outbound connections from a virtual machine is called **connection troubleshoot**. It allows you to test the connection between a virtual machine and IPv4 address, URI, FQDN and another virtual machine. The information returned by connection troubleshoot is the connection at a specific point in time. If an endpoint becomes unreachable, this tool will inform you of the possible cause.

The type of issues that connection troubleshoot can detect are:

- DNS resolution failures
- High VM CPU and memory utilization
- NSG and firewall rules that blocks the traffic
- Misconfigured or missing routes

Analyze the ingress and egress IP traffic through a Network Security Group

To allow/deny an inbound or outbound network traffic to a network interface, you need to create a network security group (NSG). It is also vital to monitor the traffic flowing through the NSG, and flow logs can help you optimize network flows, detect intrusions, monitor throughput, verifying compliance and many more.

The NSG flow logs allow you to log the source and destination of an IP address, protocol, port, and whether traffic was allowed or denied by an NSG. If you need to analyze the logs, you can use tools like PowerBI or Traffic Analytics.

Reference:
https://docs.microsoft.com/en-us/azure/network-watcher/network-watcher-monitoring-overview

COMPARISON OF AZURE SERVICES

Azure Virtual Machine vs Web App

	Azure Virtual Machine	Azure Web App
Description	Infrastructure as a service, if you need to have full control over your computing environment.	Platform as a service, it allows you to integrate the app without managing the underlying infrastructure.
Deploy	Uses an OS image.	Uses a runtime stack.
State Management	Stateful or stateless	Stateless
Autoscaling	You need to use VM scale sets to support auto scaling in virtual machines.	Autoscaling is a built-in service in App Service.
Scale Limit	1000 nodes per scale set for platform image and 600 nodes per scale set for custom image	20 instances and 100 with App Service Environment
Traffic Distribution	Distribute the incoming network traffic using Azure load balancer.	Load balancing is integrated into App Service.
Architecture Styles	The supported architecture styles are N-Tier and Web-Queue-Worker.	The supported architecture styles are N-Tier and Big compute (HPC).

Azure Container Instances (ACI) vs Azure Kubernetes Service (AKS)

	ACI	AKS
Description	Run containers without managing servers.	Orchestrate and manage multiple container images and applications.
Deployment	For event-driven applications, quickly deploy from your container development pipelines, run data processing, and build jobs.	Uses clusters and pods to scale and deploy applications.
Web Apps (Monolithic)	Yes	Yes
N-Tier Apps (Services)	Yes	Yes
Cloud-Native (Microservices)	Yes	Yes, recommended for Linux containers
Batch/Jobs (Background tasks)	Yes	Yes
Use cases	• Dev/Test scenarios • Task automation • CI/CD agents • Small/scale batch processing • Simple web apps	• Containers and application configuration portability • Enables you to select the number of hosts, size, and orchestrator tools • Transfer container workloads to the cloud without changing your current management practices.
Major Difference	You should use AKS if you need full container orchestration, such as service discovery across multiple containers, automatic scaling, and	

	coordinated application upgrades.

Azure Scale Set vs Availability Set

	Scale Set	Availability Set
Description	A group of identically configured virtual machines spread across fault domains.	A group of discrete virtual machines spread across fault domains.
Workloads	Use Scale Set for unpredictable workloads (autoscale).	Use Availability Set for predictable workloads.
Domain default	Has 5 fault domains and 5 update domains by default	Has 3 fault domains and 5 update domains by default
Configuration	Virtual machines are created from the same image and configuration.	Virtual machines are created from different images and configurations.
Distribution	Virtual machine scale sets can be distributed within a single datacenter or across multiple data centers.	Virtual machines are automatically distributed within a data center.
Number of VMs	Scale sets can increase the number of virtual machines based on demand.	You can only add a virtual machine to the availability set when it is created.
Pricing	Scale sets have no additional charge. You only pay for the computing resources.	Availability set has no additional charge. You only pay for the computing resources.

Azure Blob vs Disk vs File Storage

	Blob Storage	Disk Storage	File Storage
Type of storage	Object storage to store all types of data formats.	Block storage for virtual machines.	File system across multiple machines.
Max Storage Size	Same as maximum storage account capacity	65,536 GiB for ultra disk 32,767 GiB for standard and premium drives	Scale up to 100 TiB
Max File Size	190.7 TiB for block blob 195 GiB for append blob 8 TiB for page blob	Equivalent to the maximum size of your volumes	4 TiB for a single file
Performance (Throughput)	500 requests per second for a single blob	Up to 2000 MBps per disk.	6,204 MiB/s for egress 4,136 MiB/s for ingress
Data Accessing	Objects can be accessed via HTTP/HTTPs.	A single virtual machine in a single AZ.	Share your files either on-premises or in the cloud.
Encryption Methods	Encrypt your data using Azure SSE (256-bit AES)	SSE by storage service and ADE for OS and data disks.	Encrypt your data using Azure SSE (256-bit AES)
Backup and Restoration	Versioning, snapshots and object replication	You can back up your managed disks at any point in time using snapshots.	Uses file share snapshots

Pricing	You are billed based on the stored data per month, operations performed, data transfer, and redundancy.	You pay for the disk size, snapshots, and number of transactions.	You pay for the provisioned GiB per month and the number of servers connected to the cloud endpoint.
Use Cases	Static website, media and log files, backups, analytics workloads	Boot volumes and transaction-intensive workloads	Central location of your files, monitoring logs and applications

Locally Redundant Storage vs Zone-Redundant Storage vs Geo-Redundant Storage

	Locally-Redundant Storage (LRS)	Zone-Redundant Storage (ZRS)	Geo-Redundant Storage (GRS)
Replication	Replicates your data 3 times within a single physical location synchronously in the primary region.	Replicates your data across 3 Azure Availability Zones synchronously in the primary region	Replicates your data in your storage account to a secondary region
Redundancy	Low	Moderate	High
Cost	Provides the least expensive replication option	Costs more than LRS but provides higher availability	Costs more than ZRS but provides availability in the event of regional outages
Percent durability of objects over a given year	At least 99.999999999% (11 9's)	At least 99.9999999999% (12 9's)	at least 99.99999999999 99% (16 9's)
Availability SLA for read requests	At least 99.9% (99% for cool access tier)	At least 99.9% (99% for cool access tier)	At least 99.9% (99% for cool access tier) for GRS At least 99.99% (99.9% for cool access tier) for RA-GRS
Availability SLA for write requests	At least 99.9% (99% for cool access tier)	At least 99.9% (99% for cool access tier)	At least 99.9% (99% for cool access tier)
Available if a node went down within a data center?	Yes	Yes	Yes

Available if the entire data center (zonal or non-zonal) went down?	No	Yes	Yes
Available on region-wide outage in the primary region?	No	No	Yes
Has read access to the secondary region if the primary region is unavailable?	No	No	Yes
Supported storage account types	General-purpose v2 General-purpose v1 Block blob storage Blob storage File storage	General-purpose v2 Block blob storage File storage	General-purpose v2 General-purpose v1 Blob storage

Azure Load Balancer vs App Gateway vs Traffic Manager vs Front Door

	Load Balancer	Application Gateway	Traffic Manager	Front Door
Service	Network load balancer.	Web traffic load balancer.	DNS-based traffic load balancer.	Global application delivery
Network Protocols	Layer 4 (TCP or UDP)	Layer 7 (HTTP/HTTPS)	Layer 7 (DNS)	Layer 7 (HTTP/HTTPS)
Type	Internal and Public	Standard and WAF	–	Standard and Premium
Routing	Hash-based, Source IP affinity	Path-based	Performance, Weighted, Priority, Geographic, MultiValue, Subnet	Latency, Priority, Weighted, Session Affinity
Global/Regional Service	Global	Regional	Global	Global
Recommended Traffic	Non-HTTP(S)	HTTP(S)	Non-HTTP(S)	HTTP(S)
Endpoints	NIC (VM/VMSS), IP address	IP address/FQDN, Virtual machine/VMSS, App services	Cloud service, App service/slot, Public IP address	App service, Cloud service, Storage, Application Gateway, API Management, Public IP address, Traffic Manager,

				Custom Host
Endpoint Monitoring	Health probes	Health probes	HTTP/HTTPS GET requests	Health probes
Redundancy	Zone redundant and Zonal	Zone redundant	Resilient to regional failures	Resilient to regional failures
SSL/TLS Termination	–	Supported	–	Supported
Web Application Firewall	–	Supported	–	Supported
Sticky Sessions	Supported	Supported	–	Supported
VNet Peering	Supported	Supported	–	–
SKU	Basic and Standard	Standard and WAF (v1 & v2)	–	Standard and Premium
Pricing	Standard Load Balancer – charged based on the number of rules and processed data.	Charged based on Application Gateway type, processed data, outbound data transfers, and SKU.	Charged per DNS queries, health checks, measurements, and processed data points.	Charged based on outbound/inbound data transfers, and incoming requests from client to Front Door POPs.

Network Security Group (NSG) vs Application Security Group (ASG)

	Network Security Group	Application Security Group
Description	A network security group is used to enforce and control network traffic.	An application security group is an object reference within an NSG.
Features	Controls the inbound and outbound traffic at the subnet level.	Controls the inbound and outbound traffic at the network interface level.
Rules	Rules are applied to all resources in the associated subnet.	Rules are applied to all ASGs in the same virtual network.
Direction	Has separate rules for inbound and outbound traffic.	Has separate rules for inbound and outbound traffic.
Limits	NSG has a limit of 1000 rules.	ASGs that can be specified within all security rules of an NSG have a limit of 100 rules.
Action	Supports ALLOW and DENY rules.	Supports ALLOW and DENY rules.
Constraints	You are not allowed to specify multiple IP addresses and IP address ranges in the NSG created by the classic deployment model.	You are not allowed to specify multiple ASGs in the source or destination.

Azure Policy vs Azure Role-Based Access Control (RBAC)

	Azure Policy	Role-based Access Control (RBAC)
Description	Ensure resources are compliant with a set of rules.	Authorization system to provide fine-grained access controls.
Focus	Policy is focused on the properties of resources.	RBAC focuses on what resources the users can access.
Implementation	You specify a set of rules to prevent over-provisioning of resources.	You grant permission on what users can create.
Default access	By default, rules are set to ALLOW.	By default, all access is denied.
Scope	Policy within the resource group or subscription.	Grant access to users or groups within a subscription.
Integration	Both services work hand-in-hand to provide governance around your environment.	

Microsoft Entra ID vs Azure Role-Based Access Control (RBAC)

	Microsoft Entra ID	Azure RBAC
Description	An identity and access management service that helps you access internal and external resources.	An authorization system that manages users' access to Azure resources, including what they can do with those resources and what areas they can access.
Focus	Grants permissions to manage access to Microsoft Entra ID resources.	Grants permissions to manage access to Azure resources.
Scope	Tenant level	Specify at multiple levels (management group, subscription, resource group, and resource)
Roles	Important Entra ID built-in roles: 1. Global Administrator – manage access to all the administrative features in Microsoft Entra. 2. User Administrator – create and manage different types of users and groups in Azure. 3. Billing Administrator – it can manage subscriptions, support tickets, make purchases, and monitor service health. Supports custom roles. You can assign multiple roles to a user.	Fundamental Azure RBAC built-in roles: 1. Owner – full access to all Azure resources. 2. Contributor – create and manage all types of resources in Azure. 3. Reader – a user with this role can only view Azure resources 4. User Access Administrator – it has permissions to manage user access to all types of resources. Supports custom roles in P1 and P2 licenses. You can assign multiple roles on a user.
Role information	You can access the role information in the Azure Portal, Microsoft 365 admin center, Microsoft Graph, and Azure AD PowerShell.	You can access the role information in the Azure Portal, CLI, PowerShell, Resource Manager templates, and REST API.

Pricing	Microsoft Entra ID has three editions: Free, P1, and P2. For the P1 and P2 licenses, you are charged on a monthly basis.	Azure RBAC is free and included in your Azure subscription.

ABOUT THE AUTHORS

Jon Bonso

Born and raised in the Philippines, Jon is the Co-Founder of **Tutorials Dojo**. Now based in Sydney, Australia, he has over a decade of diversified experience in Banking, Financial Services, and Telecommunications. He's 10x AWS Certified and has worked with various cloud services such as Google Cloud, and Microsoft Azure. Jon is passionate about what he does and dedicates a lot of time creating educational courses. He has given IT seminars to different universities in the Philippines for free and has launched educational websites using his own money and without any external funding.

Gerome Pagatpatan

Gerome Gerome currently works as a software engineer and holds 5 cloud certifications from Amazon Web Services, Microsoft Azure, and Oracle. He also co-authored high-quality educational materials in the cloud computing space, which have been used by over a quarter-million people worldwide. He is passionate about education, and now it's his turn to share his knowledge, experiences, and passion for cloud computing.