Module 3: Building Azure IaaS-Based Server Applications

Contents:

Module overview

This module identifies workloads that are ideally deployed using Infrastructure-as-a-Service services in Azure. The module focuses on the VM Scale Sets and Virtual Machine services in Azure and how to best deploy workloads to these services using best practices and features such as Availability Sets.

Objectives

After completing this module, students will be able to:

- Design an availability set for one or more virtual machines
- Describe the differences between fault and update domains.
- Author a VM Scale Set ARM template.
- Join a virtualized machine to a domain either in Azure or on a hybrid network.

Lesson 1: High Availability

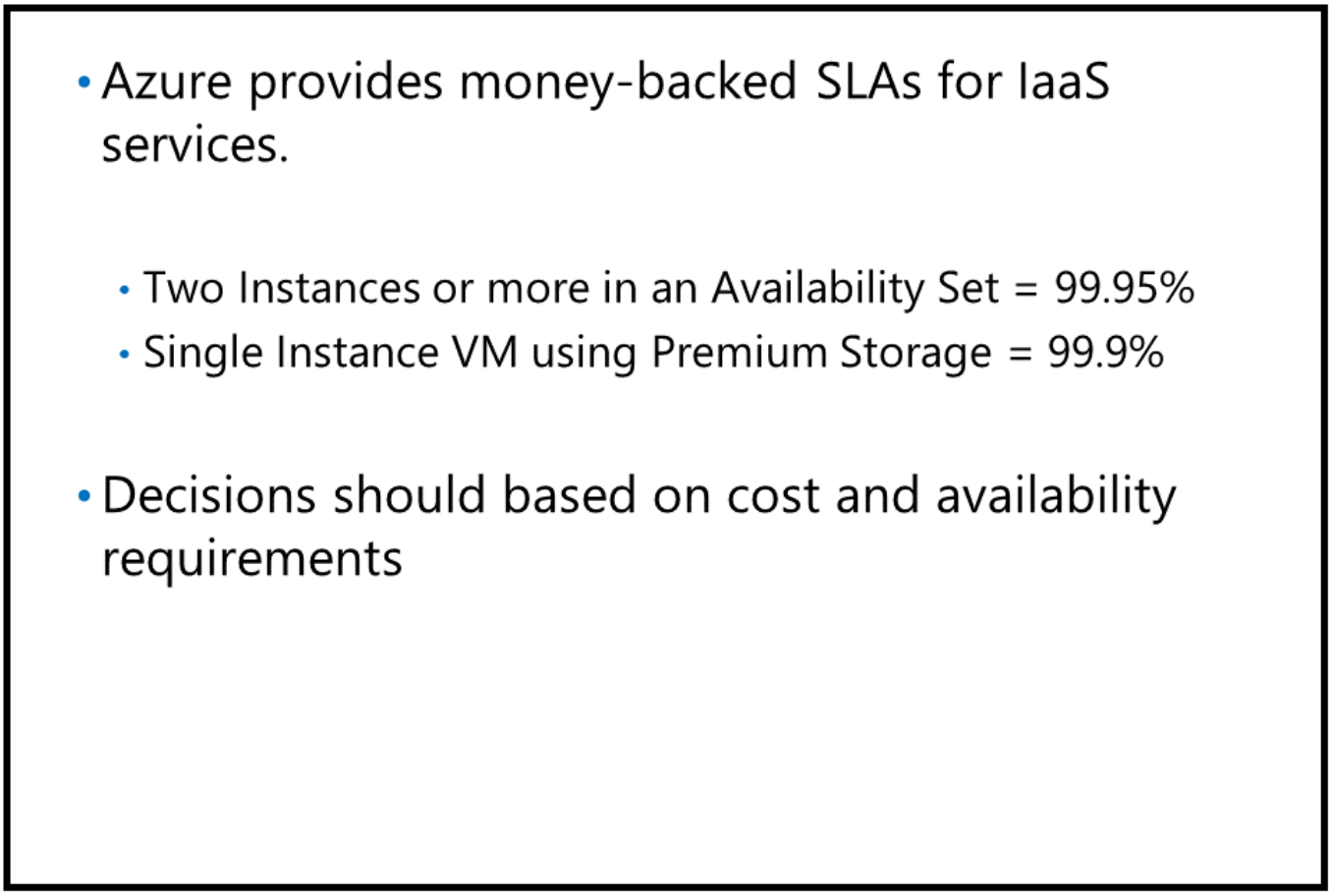
Azure has provided a money-backed Service Level Agreement (SLA) for a long time and with the recent product releases and enhancements now provides several different services and levels of service. This lesson will explain what is available and the best way to ensure the availability of your resources. This lesson will cover Availability Sets and Availability Zones.

Lesson objectives

After completing this lesson, you will be able to:

- Describe Availability features of Microsoft Azure
- Understand difference between an Availability Set and an Availability Zone
- Deploy resources into Availability Sets correctly
- Decide which applications and resources should not use Availability Sets.

Azure Availability



Microsoft Azure provides a Service Level Agreement (SLA) that is backed by a financial service credit payment for infrastructure as a Service (IaaS) Virtual Machines. The SLA depends entirely upon the deployment of the virtual machine and what resources it uses. The aim is to prevent virtual machine reboots.

The method Azure uses to ensure that the SLA can be provided is an Availability Set. An availability set ensures that all virtual machines that are added to the set are placed in such a way as to ensure that neither hardware faults or Azure fabric updates that is unplanned and planned maintenance events can bring down all of the virtual machines.

Application Availability

An Azure virtual machine can be impacted for one of three reasons.

- Unplanned hardware maintenance event
- An unexpected downtime

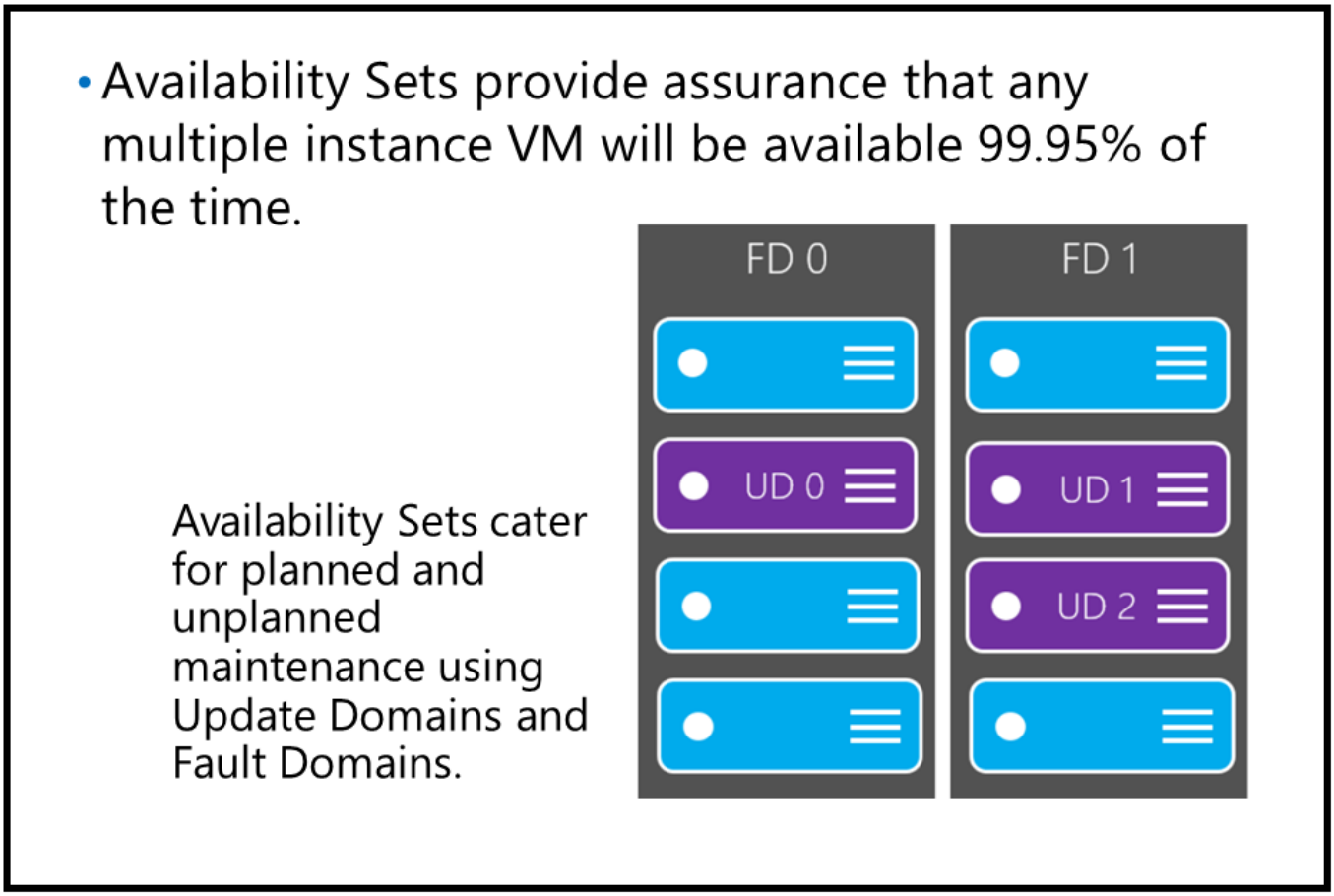
- Planned maintenance events

To reduce or remove the impact of downtime related to these events there are several steps to take, these include:

- Place virtual machines in an availability set for redundancy
- Use managed disks for all VMs placed in an availability set
- Use Scheduled Events to respond to events
- Place each tier of your application in a separate availability set
- Use a load balancer in combination with availability sets

All the above steps provide additional high availability for your application and can be used in varying situations. The fundamental building block is the Availability Set.

Availability Sets



The availability set is the recommended solution for placing multiple instance VMs, this configuration will allow for at least one VM being available at least 99.95% of the time.

> **Best Practice:** Avoid single instance VMs in an availability set. These are not subject to any SLA unless all the Operating System and Data disks are using Premium storage.

Remember Availability Sets comprise of Update Domains and Fault Domains. As shown in the image below:

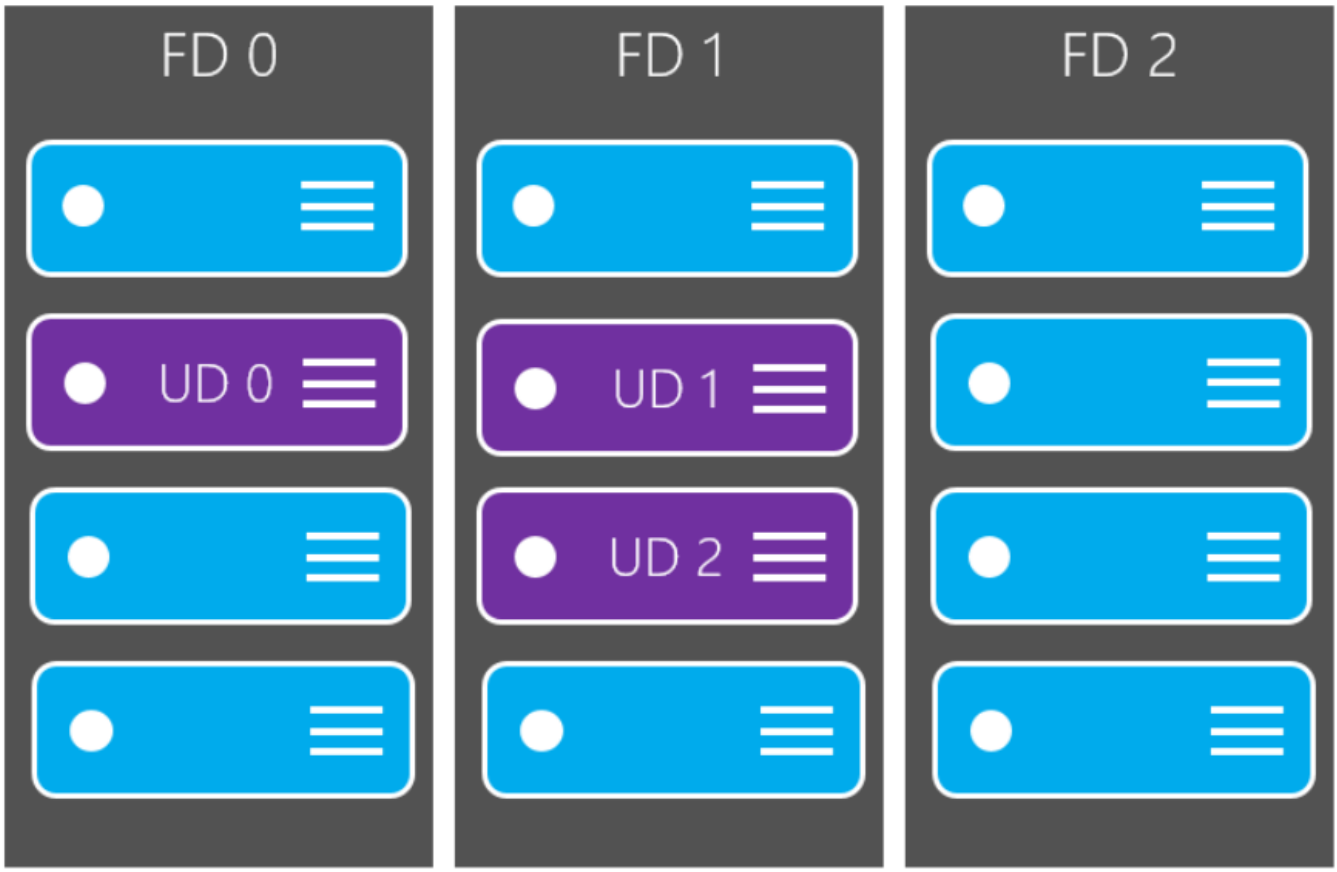


FIGURE 3.1: UPDATE AND FAULT DOMAINS

Each machine in the Availability set is placed in an Update Domain and a Fault domain.

An Availability Set is a logical grouping service that you can create and use to ensure that the VMs you deploy within an Azure datacenter are isolated from each other. Azure places the VMs you deploy within an Availability Set across multiple physical servers, racks and network switches.

In the event of an Azure hardware or software failure, only a proportion of the VMs deployed to your Availability set are impacted. The application running on the VMs will remain available. Architecturally availability sets are essential to most cloud-based solutions.

Multiple Availability Sets.

An extension of the availability set model is used logically to place individual tiers of an application into separate Availability Sets.

In this example we have a two-tier VM-based application which contains three load-balanced front-end Web servers and two back-end VMs that host a SQL Server data store. In Azure, it would be best practice to create two availability sets in preparation for the application. AVSet1 for the Web tier and AVSet2 for the data tier. Each time you create a VM for the application, you deploy it to the correct Availability set for the function it will perform.

Azure will then ensure that the VMs you create within the availability set are isolated across multiple physical hardware resources. If the physical hardware that one of your Web Server or Database Server VMs is running on has a problem, you know that the other instances of your Web Server and Database VMs remain running because they are on different hardware.

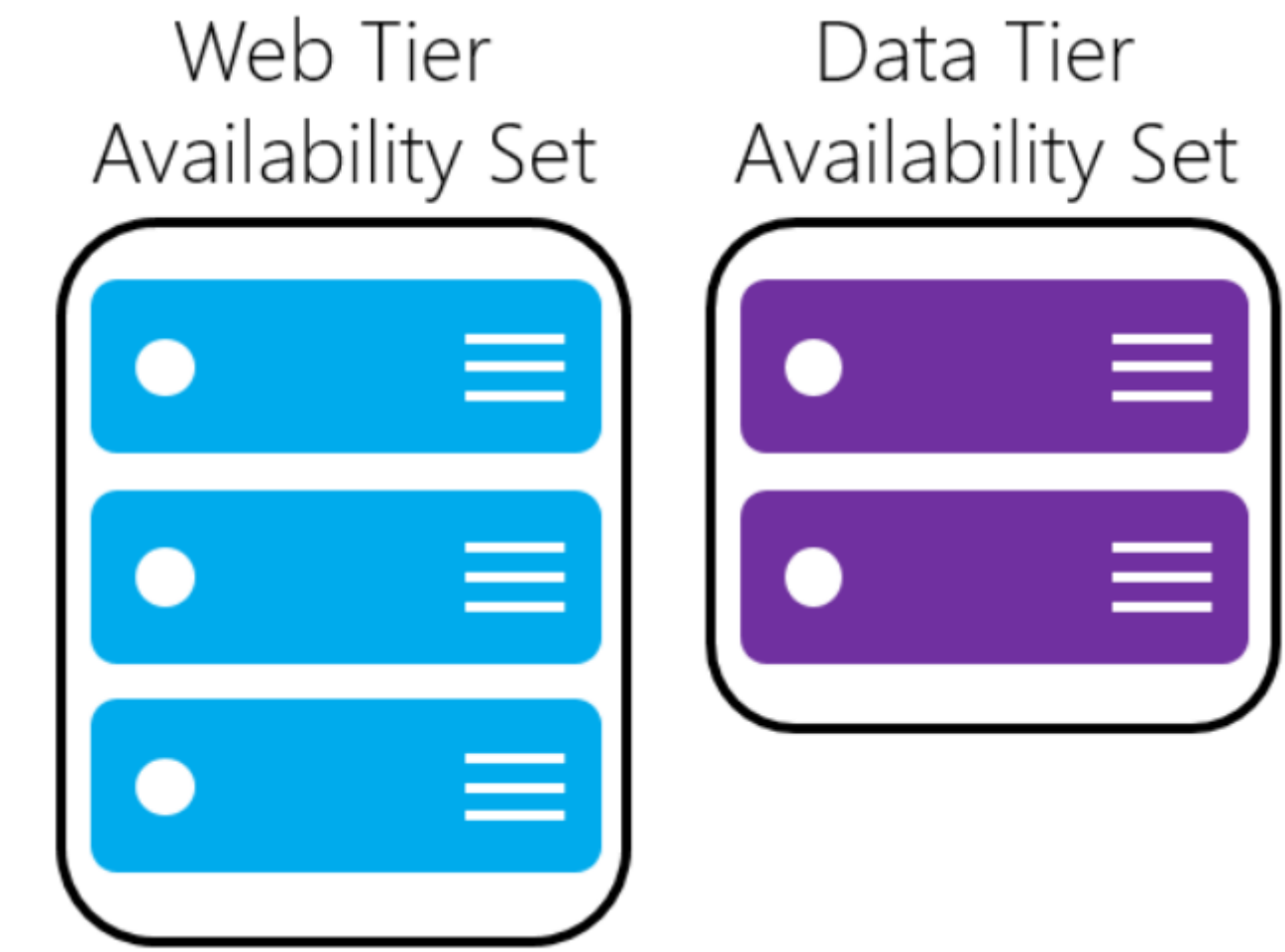
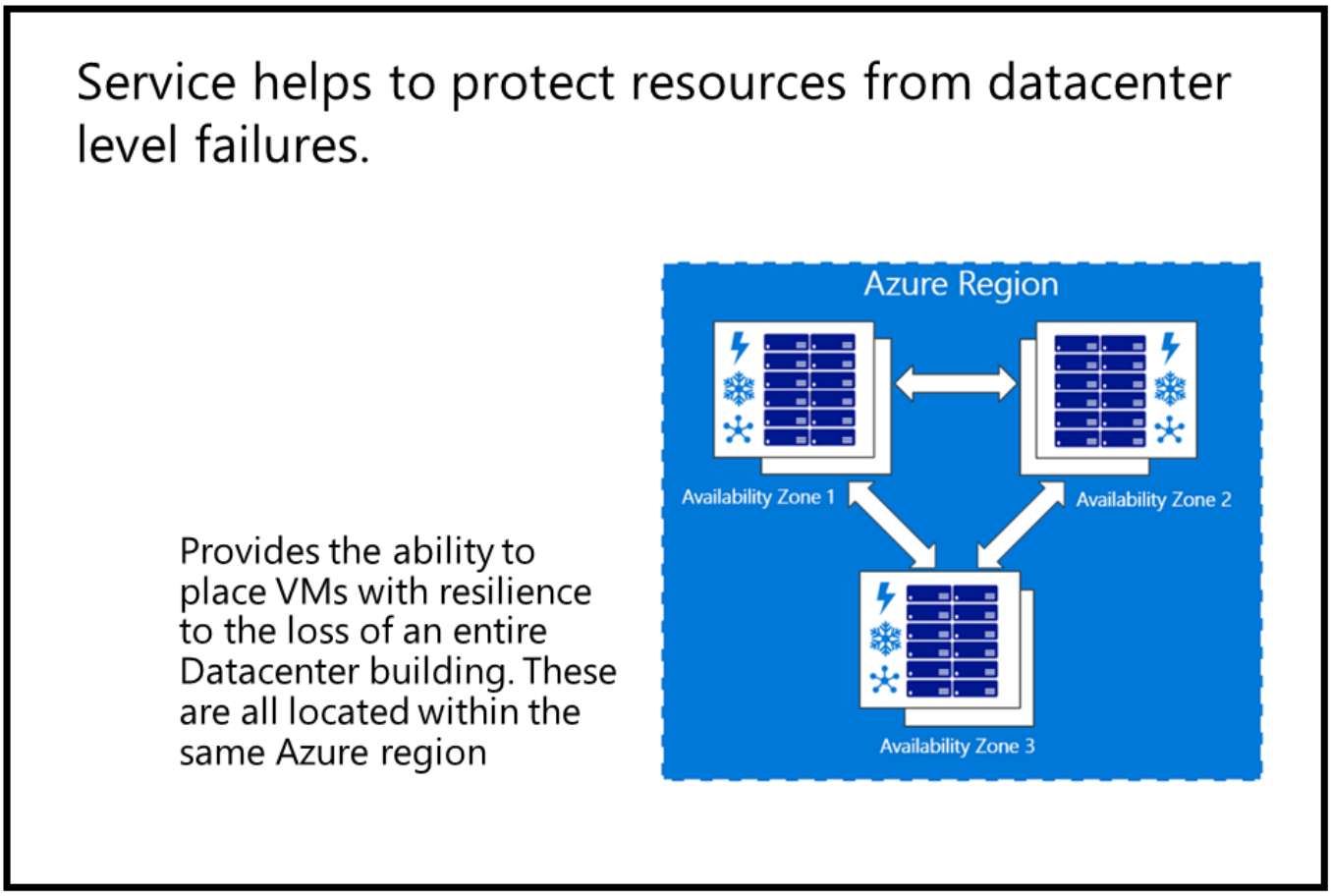


FIGURE 3.2: N-TIER AVAILABILITY SETS

Availability Zones



A new preview service further extending highly available VMs is the Availability Zone. Whilst Availability Sets can protect from individual hardware faults and even to rack-based faults, the advent of a datacenter-wide fault would prevent the Availability set from functioning.

To extend the capability further, Microsoft has released a preview service which as an alternative to Availability Sets can provide highly available VMs across datacenter buildings or Zones. In this instance, the Availability Zone feature will allow for a complete data center failure and keep your VM based application running. The title Zone indicates a separate zone or building within a single Azure region.

There is a maximum of three Availability Zones per supported Azure region. Each Zone operates on an entirely isolated power source, cooling system, and network infrastructure.

The use of Availability Zones to architect your applications to use VMs replicated in Zones provides an additional level of protection

There are few scenarios where the use of either Availability Zones or Availability Sets would not be used in a production IaaS based infrastructure or application. The use of single VMs or VM's without availability sets whilst not attracting an SLA would still be suitable for dev and test scenarios.

Lesson 2: Templated Infrastructure

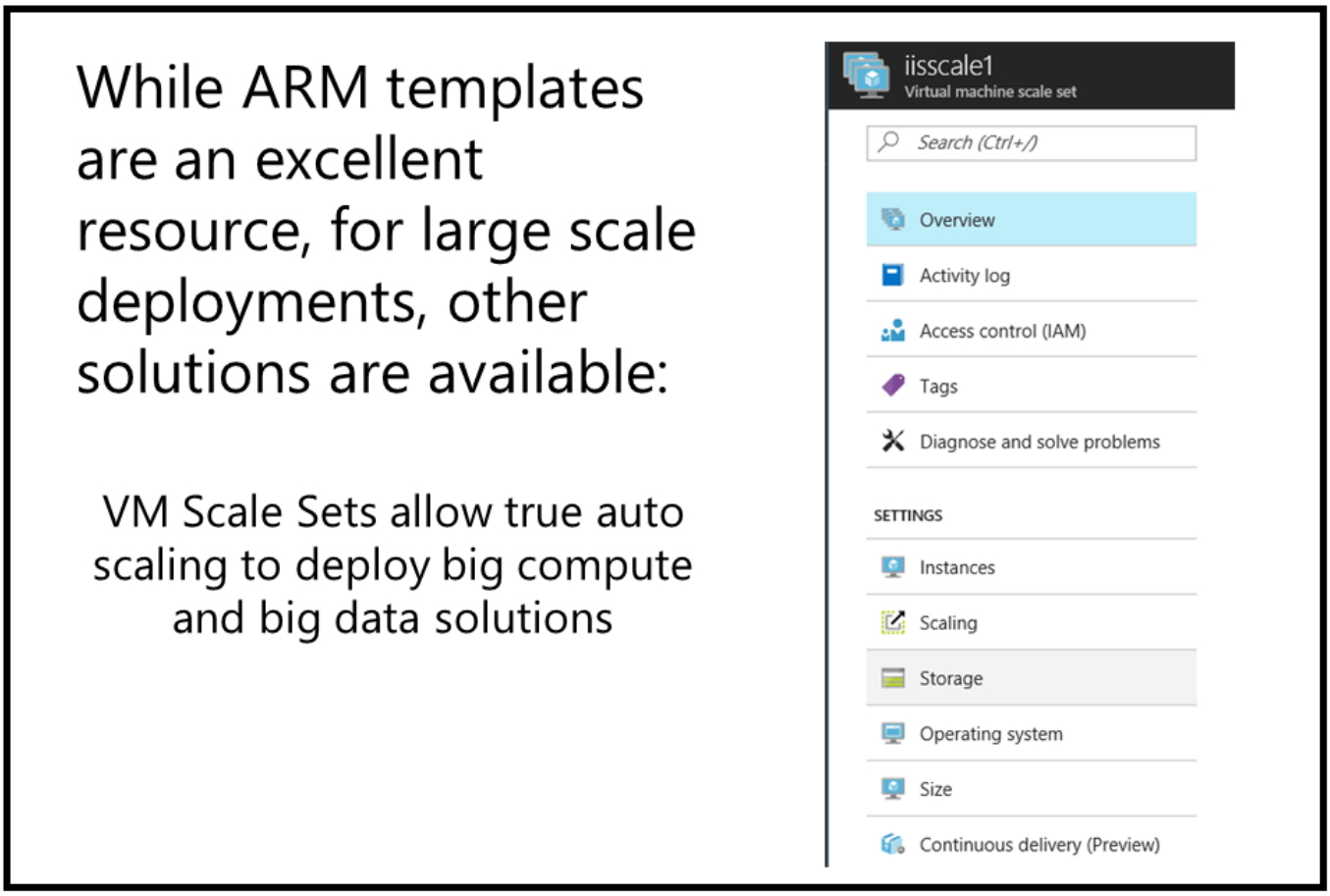
Azure provides a compute resource capable of true Autoscale without pre-provisioning Virtual machines. Azure VM Scale Sets allow for templated deployment of marketplace images and custom images in highly scalable and highly available infrastructure to provide a platform for big compute and big data IaaS based applications. This lesson provides an overview of Scale Sets and discusses the differences between deployment of multiple VMs and scale sets in your application.

Lesson objectives

After completing this lesson, you will be able to:

- Describe Azure VM Scale Set features.
- Decide how to deploy VM Scale sets.
- Decide whether VM Scale Sets or standalone VMs provide the features you require.
- Deploy an application to a VM Scale Set using VSTS.

Templated Infrastructure

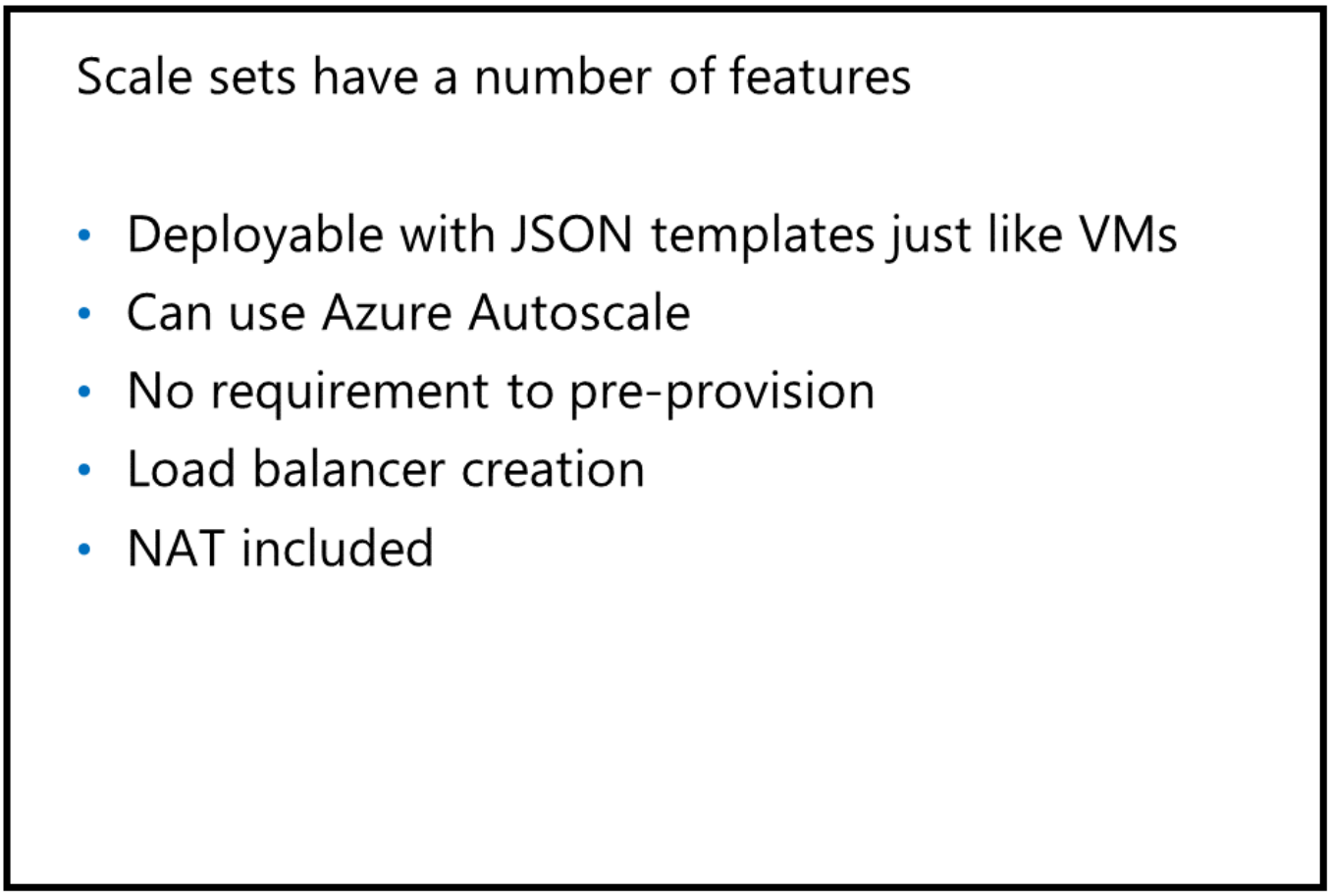


Scalable architecture in cloud platforms is nothing new, but always comes with a trade-off between complexity and control. Complexity is defined here as the difficulty in how to define and deploy the resources you need to control the individual resources once deployed. The tradeoff leads typically to a choice between a PaaS solution and an IaaS solution, neither really providing everything that is required.

Azure VM Scale Sets are an Azure compute resource that provides both a high degree of infrastructure resource control without the need to invest time and energy in managing the attendant networking, storage and compute. In addition, the built-in load balancing allows it to be controlled like IaaS but Scaled like PaaS.

It is usual when building cloud infrastructure to create storage, compute and network resources and create the dependencies between them. Azure VM Scale sets handle this resource creation for you and go one step further by managing the necessary configurations and optimizations when the scale set scales up or down, further reducing the workload required.

Virtual Machine Scale Sets



An Azure VM Scale Set has several features that make it attractive to the Azure architect. The ability to define a VM Scale Set by JSON template and deploy it using any of the standard deployment methods enables their use in many automated solutions. This extends into continuous deployment scenarios with Visual Studio Team Services.

An Azure VM Scale Set allows a Virtual machine to deploy up to 1000 times in the same subnet in a controlled and automated manner with accurate auto-scaling.

An Azure VM Scale Set also requires no pre-provisioning of the Virtual Machine before adding to the scale set. The network and load balancer are created, configured and managed automatically, including the Network Address Translation (NAT) for access to and from the VM Instances.

These features added to the ease of deployment through the portal, Azure PowerShell or Azure CLI make the Azure VM Scale Set a powerful tool for the Azure cloud architect.

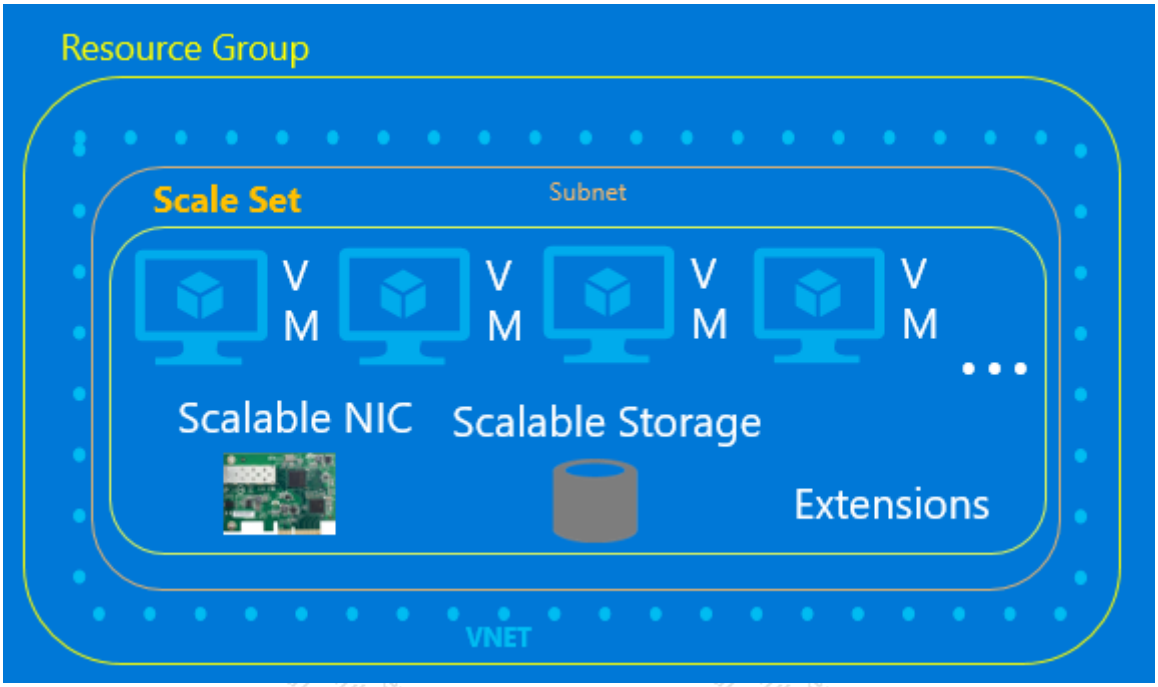


FIGURE 3.3: VIRTUAL MACHINE SCALE SET

Virtual Machines vs. Virtual Machine Scale Sets

Scale Sets
- Easy to grow and shrink on demand
- Easy to reimage
- Easy to overprovision
- Upgrade policies

VMs
- Attach disks to VMs
- Attach non-empty disks
- Snapshot a VM
- Capture a VM Image
- Migrate from native to managed disks
- Assign IPv6 public IP addresses to individual VM NICs

Azure Virtual Machines and Azure VM Scale Sets have several unique features that allow the architect to choose between their suitability for an application or deployment.

Azure VM Scale Sets

The Scale Set capacity property allows you to deploy more VMs in parallel. This is easier than writing a scripting the orchestration required to deploy individual VMs in parallel. With Azure VM Scale Sets:

- You can use Azure Autoscale to automatically scale a scale set but not individual VMs.
- You can reimage scale set VMs but not individual VMs.
- You can overprovision scale set VMs for increased reliability if a faster deployment time is required. To do this with individual VMs custom code must be written.
- Can take advantage of an upgrade policy. This makes it easy to upgrade all the VMs in your Scale. With individual VMs, this must be orchestrated.

Azure Virtual Machines

In contrast, with Azure Virtual Machines:

- You can attach data disks to individual VMs, but attached data disks in VM Scale Sets apply to all instances in the set. You can also attach non-empty data disks to individual VMs but not to VMs in a scale set.
- You can snapshot an individual VM but not a VM in a scale set. You can also capture an image from an individual VM but not from a VM in a scale set.
- You can migrate an individual VM to use managed disks from native disks, this cannot be done in a VM Scale Set.
- You can assign IPv6 public IP addresses to individual network Interface Cards in a VM but cannot do so for VMs in a scale set.

> **Note:** You can, however, assign an IPv6 public IP address to load balancers, it does not matter if the load balancer is in front of a VM or a VM Scale Set.

Virtual Machine Scale Set Considerations

- Custom Extensions can be used to configure new VM instances when scaling – this can add time to the deployment.
- Custom Images can be used to deploy all images to the scale sett, this scales VMs in a ready to use state.

When working with VM Scale Sets there are design considerations which will affect the ease of use and performance of the resultant Scale Set.

Connecting to a VM Scale Set instance VM

If using a Windows VM in the Scale Set, it is possible to connect to a specific VM instance by accessing the Load balancer inbound NAT rules and using the correct IP address and custom port. In the instance below the RDP client would be pointed at 52.166.236.225:50007.

Once you enter the correct admin user credentials access will be granted to the Virtual Machine instance:

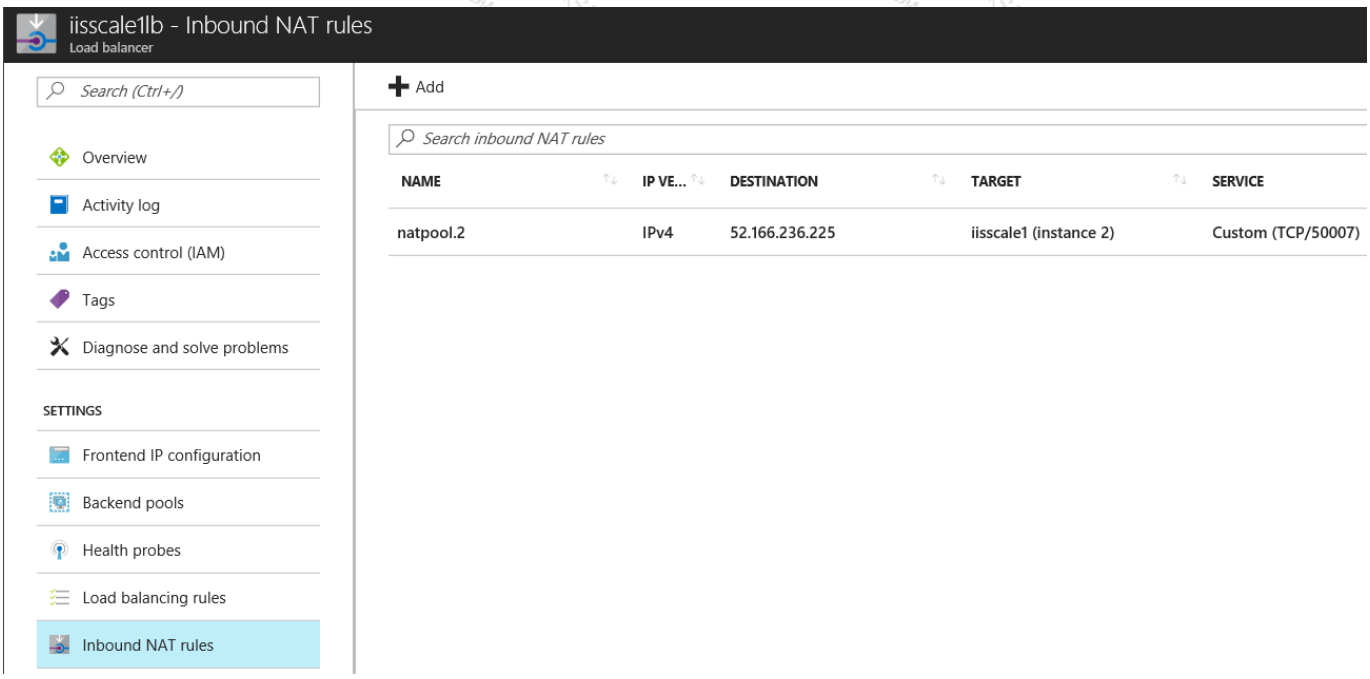


FIGURE 3.4: INBOUND NAT RULES

Note: For a Linux VM Scale Set, there is a choice between SSH key or username and password.

Continuous Delivery

Use Continuous delivery to maintain an application in a VMSS with Visual Studio Team Services. Continuous delivery in Visual Studio Team Services simplifies setting up a robust deployment pipeline for your application. By default, the pipeline builds code, and updates VM scale set with the latest version of your application. The update to VM Scale set can be done either by creating an image and using that to create/update VM scale set or by using customs script VM extension to install/update your application on VM scale set. You can quickly add another VM Scale Set to the pipeline to validate your changes before they ever get to production.

Need to provision additional Azure resources, run scripts, upgrade your database or run additional validation tests? You can easily extend this deployment automation to handle any other operations your application needs to do during deployment.

Visual Studio Team Services can be used to automated the deployment of code to a VMSS Instance.

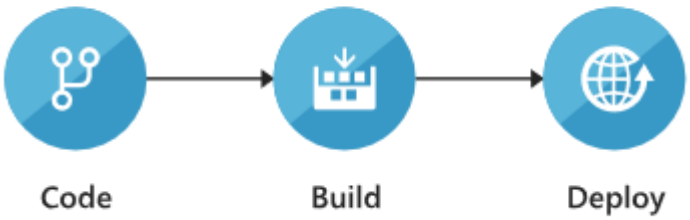


FIGURE 3.5: VISUAL STUDIO TEAM SERVICES

When deploying an application to a VM Scale Set, there are usually two ways to achieve the goal.

1. Use of VM extensions to install software to each instance at deployment time.
2. Create a custom image that already contains the OS and application in a single VHD.

Option 2 is also known as an **immutable deployment**. This method has many advantages.

- Predictability
- Easy to Scale
- Easy to roll-back
- Faster to scale (no code to install on each VM as it is deployed)

Having chosen option 2, the benefits can be further enhanced by taking advantage of the Visual Studio Team Services continuous integration toolset and the Continuous Delivery preview service in the VM Scale Set blade.

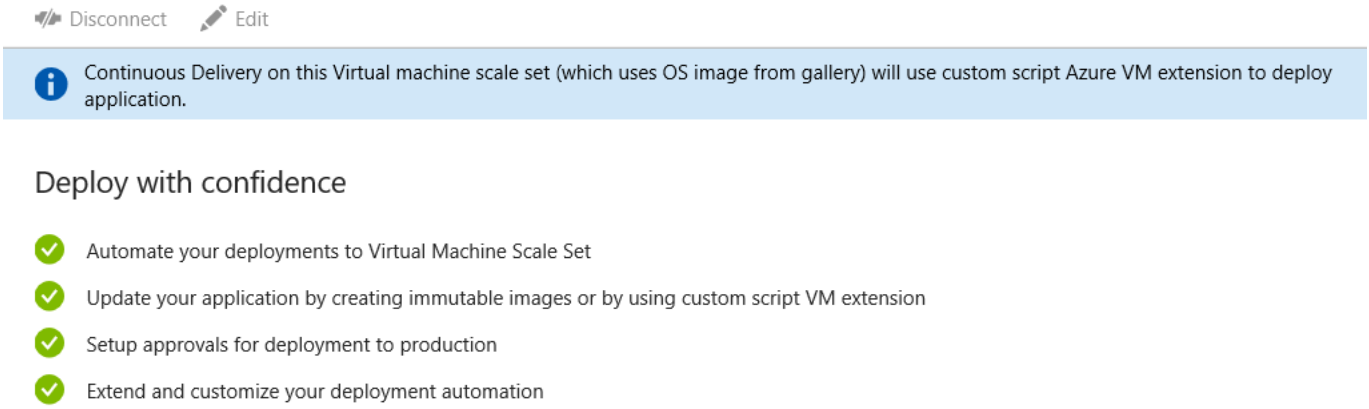


FIGURE 3.6: DEPLOY WITH CONFIDENCE

By enabling and configuring VSTS continuous delivery, the application code and deployment scripts can be deployed from either GitHub or the Visual Studio Team Service repository.

The added benefit is that any new versions of the application can be tested on a similar VM Scale set and then deployed directly into the production instances without any downtime.

Large VM Scale Sets

Azure VM Scale Sets can handle up to 1000 VMs in each Scale Set. Azure classifies a Scale Set that can scale beyond 100 VMs as a large VM Scale Set. The large VM Scale Set capability is marked by a singlePlacementGroup = false property setting. The large VM Scale set has unique requirements as well as changing the way specific aspects of a Virtual Machine deployment behave, such as load balancing and fault domains.

To decide whether your application can make efficient use of large scale sets, consider the following requirements:

- Large scale sets require Azure Managed Disks.
- Scale sets created from Azure Marketplace images can scale up to 1,000 VMs.
- Scale sets created from custom images can scale up to 300 VMs.
- Layer-4 load balancing with scale sets composed of multiple placement groups requires Azure Load Balancer Standard SKU.
- Layer-7 load balancing with the Azure Application Gateway is supported for all scale sets.
- Scale sets are defined with a single subnet – ensure subnet is large enough to handle all potential VM instances
- Ensure your compute limits are high enough, the requirement for compute cores will prevent a successful deployment if not.
- Fault Domains and Update Domains relate to a single placement group, to maintain high availability ensure there are at least two VM instances in each Fault Domain and Update Domain.

Lesson 3: Domain-Joined Virtual Machines

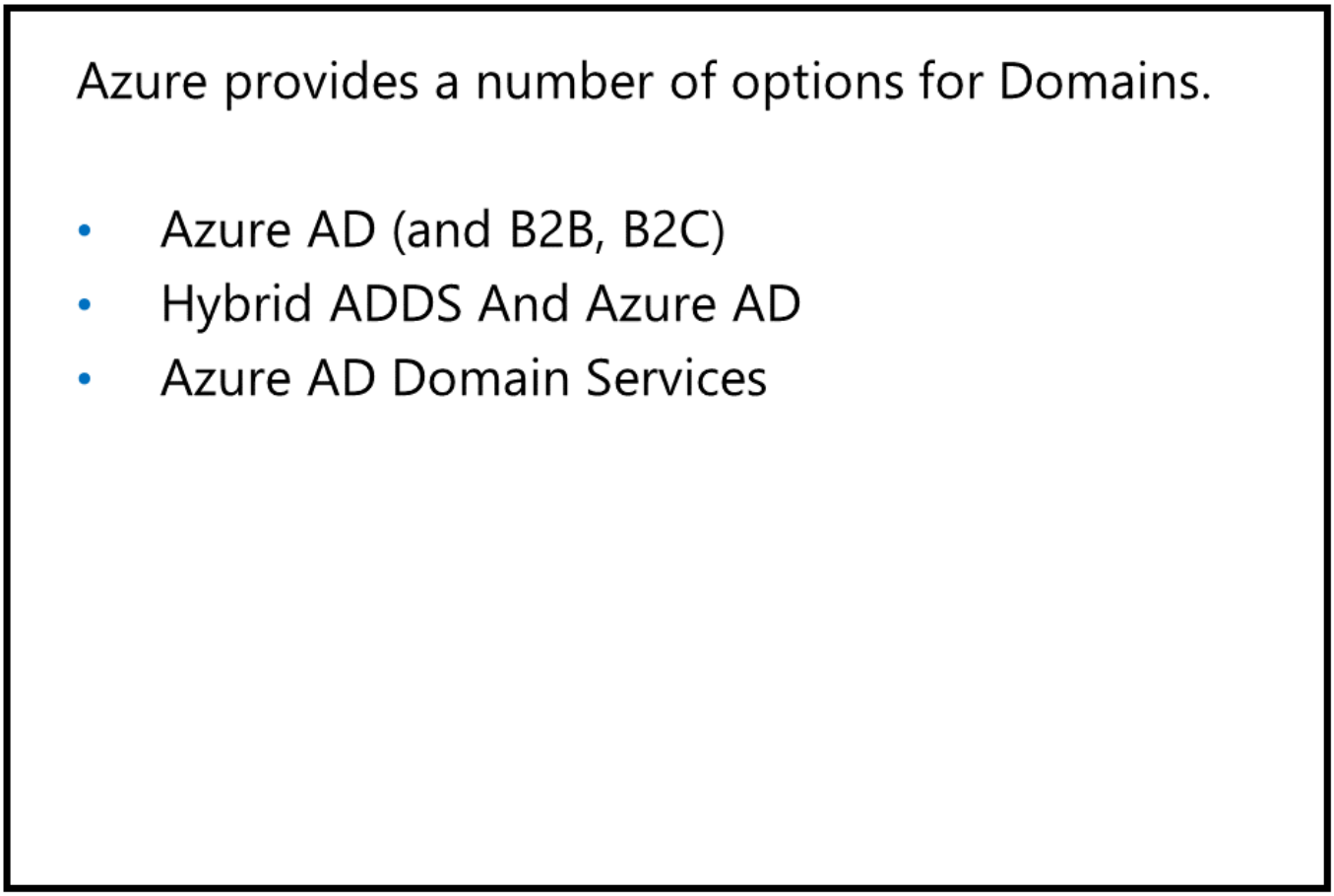
The use of Active Directory is widespread throughout the on-premises and cloud-based Windows infrastructure world. The advent of Azure AD brings many options for the Azure Architect to choose between. This lesson will examine the benefits of and differences between cloud only and hybrid solutions comprised of on-premises Active Directory Domain Services, Azure AD, and Azure AD Domain Services.

Lesson objectives

After completing this lesson, you will be able to:

- Describe Azure AD Domain Services and Hybrid AD options.
- Decide when to use Azure AD Domain Services, AD DS in an Azure VM or Hybrid on premises.
- Create an Azure AD Domain Services Managed domain.

Domain and IaaS Applications



Authentication and Authorization of users in a cloud or hybrid infrastructure require careful planning and consideration. There are several options when using Azure AD. Azure AD is a multi-tier Identity as a Service offering that provides a complete Identity and Access Management solution for your cloud or hybrid environment.

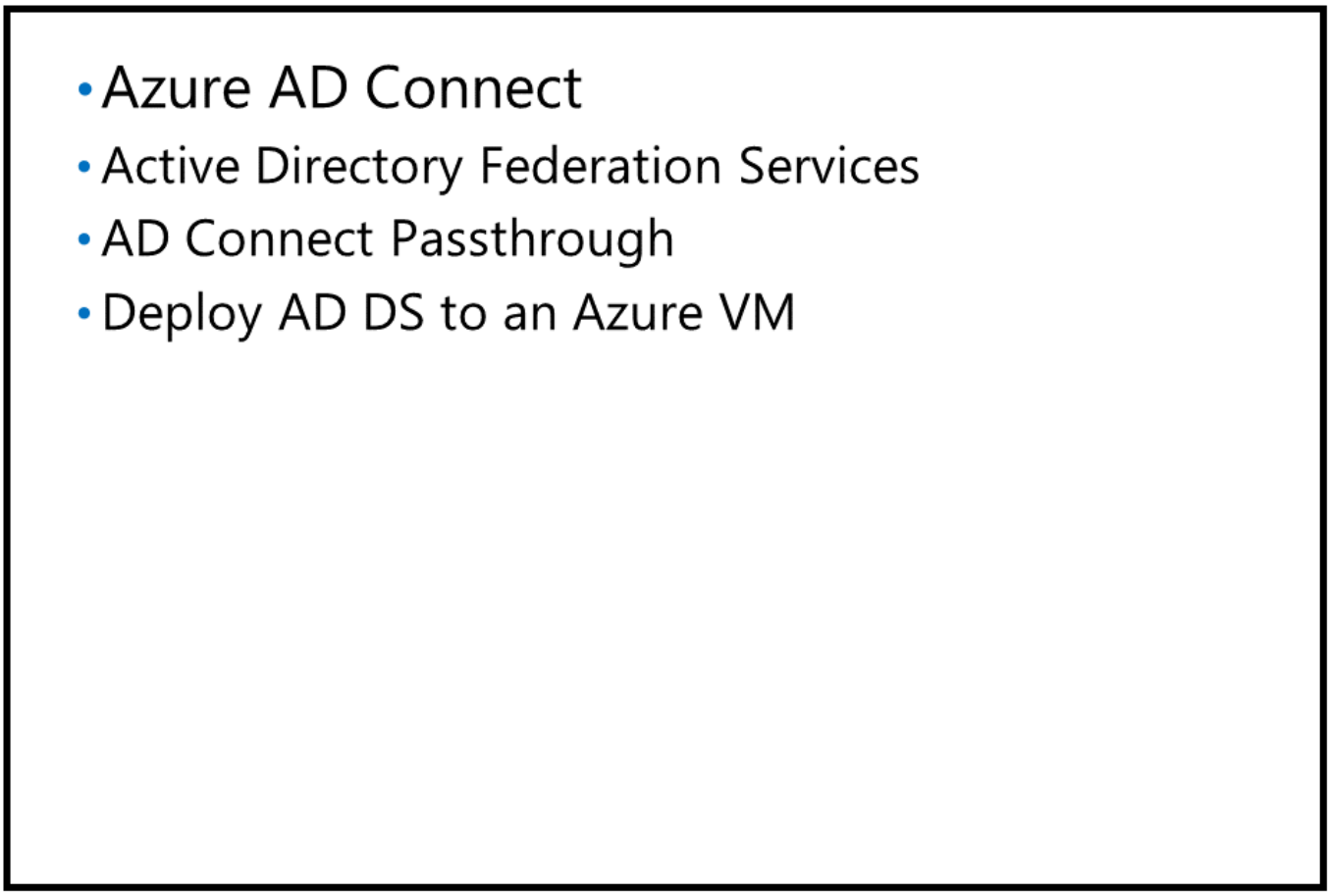
In its basic form, Azure AD is a free service that provides the ability for Single Sign-On into cloud applications. The various tiers basic, premium 1 and Premium 2 each provide additional levels of service such as Multi-Factor

Authentication and additional reporting. This lesson covers the various options available to Azure AD administrators to provide domain services to their hybrid or cloud-based networks:

- Azure AD Connect
- Azure AD Domain Services
- Azure AD pass-through Authentication

Azure AD also provides services to allow connection with consumers (Azure AD B2C) and business partners (Azure AD B2B) without deploying complex infrastructure or federation.

Hybrid Connectivity



Azure AD Connect

Azure AD Connect provides the ability to integrate your on-premises directories with Azure AD. Having deployed AD Connect, you can provide single sign-on to cloud applications such as Office 365, Azure and other SaaS applications.

Why use AD Connect? Having a common identity to access both cloud and on-premises applications and resources enables users to take advantage of a Single identity, one set of credentials to remember and a Single tool, single sign in, easy for administrators to deploy.

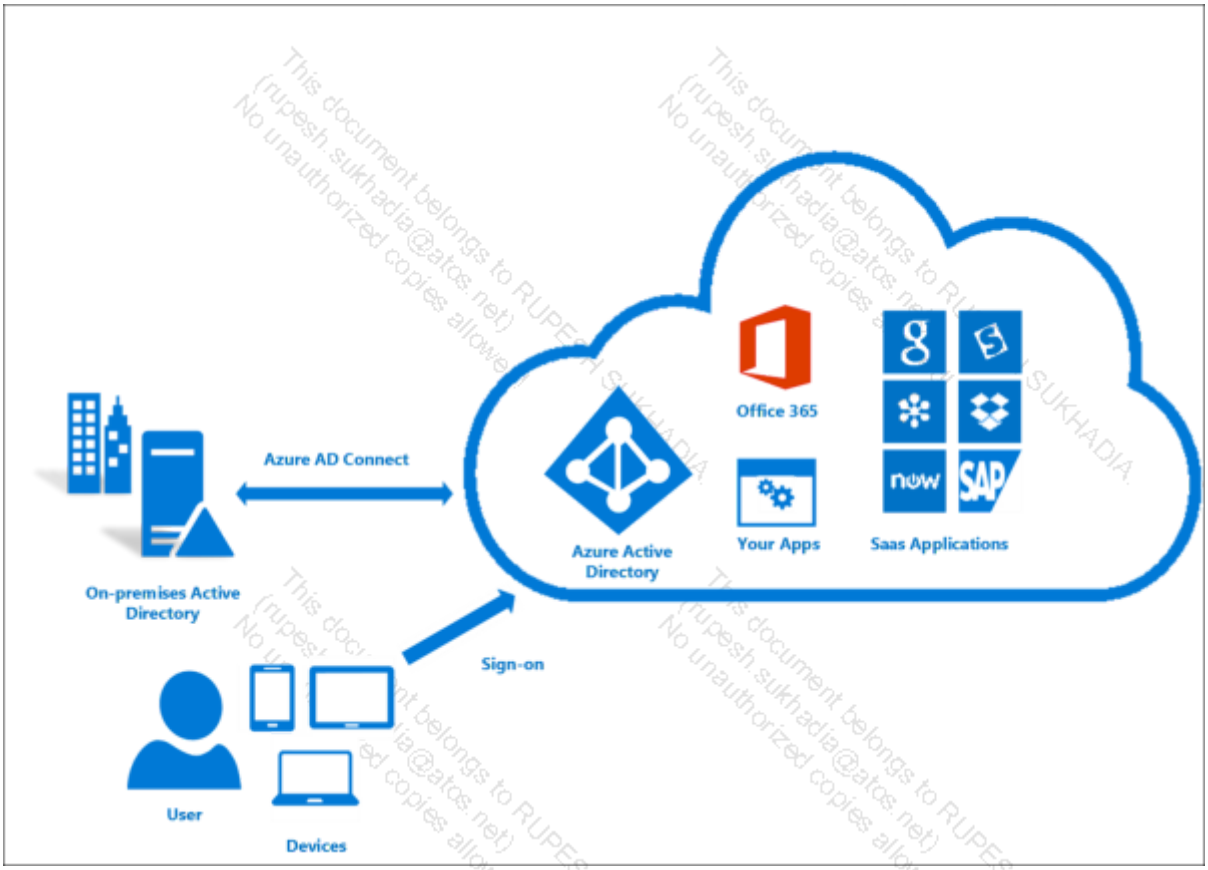


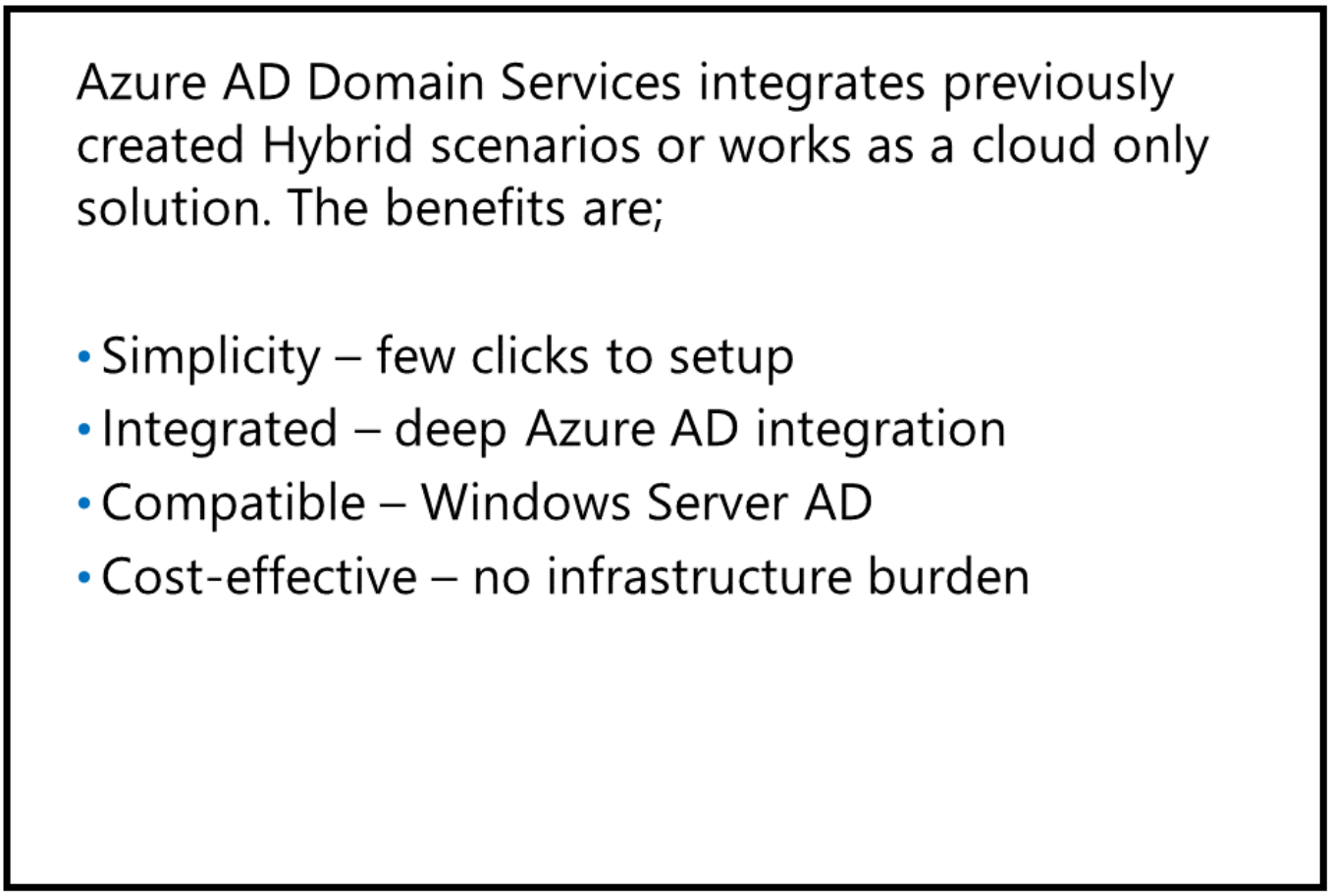
FIGURE 3.7: AZURE AD HYBRID

Azure AD Connect provides the choice of

- Password Synchronization only, the ability to synchronize users and groups
- ADFS, to allow on-premises authentication, 3rd party MFA, etc.
- Pass-through authentication, provides on-premises authentication without deploying ADFS

Whichever architecture you decide to use, if on-premises directories are included, you will need Azure AD Connect to provide the synchronization engine. The Microsoft Identity Manager (MIM) client is installed on-premises and used to configure the users, groups and attributes to be synchronized.

Azure AD Domain Services



Azure provides the ability to deploy Infrastructure solutions in many ways. To support this Azure AD Domain Services provides managed domain services such as domain join, group policy, LDAP and Kerberos authentication, all of which are entirely Windows Server Active Directory compatible.

Azure AD Domain Services allows you to take advantage of these services without having to deploy a Domain Controller to an IaaS VM in the cloud. Azure AD Domain Services is compatible with both Cloud only tenants and hybrid tenants using Azure AD Connect.

Cloud only tenant

By enabling Azure AD Domain Services on an Azure AD tenant, Azure creates a highly-available domain service which is connected to the virtual network, and all the Azure AD objects are available within this domain, but all user identities, credentials and groups, including group memberships, are created and managed in Azure AD.

The advantages of this solution are:

- The domain administrator does not need to manage this domain or any domain controllers.
- AD replication for this domain does not require management. All objects are automatically available.
- Azure manages this Domain, so the Azure AD tenant administrator has no domain or enterprise admin privileges.

Hybrid cloud tenant

By enabling Azure AD Domain Services on an Azure AD tenant that is synchronized to an on-premises directory, an additional stand-alone Domain is created by the managed service. All objects from the on-premises domain and the Azure AD tenant are available to the managed service domain. Tenant identities are still created and managed within Azure AD, and on-premises identities are still created and managed on-premises. This solution allows users to sign in to cloud services with their on-premises identities. For this to work with the Azure AD Domain Services, Azure AD Connect must be configured to allow password synchronization; this is required so resources in the cloud connected to the managed domain can use Kerberos to authenticate. The managed domain is a standalone domain and not an extension to the on-premises directory.

The advantages of this solution are:

- The domain administrator does not need to manage this domain or any domain controllers for the managed domain.
- AD replication for this domain does not require management. All objects are automatically available.
- Azure manages this Domain, so the Azure AD tenant administrator has no domain or enterprise admin privileges.

The benefits of the Azure AD Domain Services managed domain offering are;

- Simple to deploy in a few clicks– No IaaS infrastructure required to provide authentication to Virtual Machines
- Deep integration within your Azure AD tenant
- Compatible with Windows Server Active Directory – Not all features available in Windows Server AD are available in Azure AD Domain Services. The following are compatible, LDAP, Kerberos, NTLM, Group Policy, and domain join capabilities.
- Cost-effective – No need to pay for Azure IaaS Virtual Machines

Azure AD pass-through authentication

Azure AD Connect provides a new service that permits on-premises authentication without the need to deploy ADFS infrastructure. This is a considerable cost and time saver. With Azure AD Connect, you have no need for complicated certificates or trusts.

Azure AD Pass-through Authentication enables users to sign in to both on-premises and cloud applications using the same credentials. When users sign in using Azure AD, this feature validates users' passwords against your on-premises Active Directory, the same was an ADFS based solution would do.

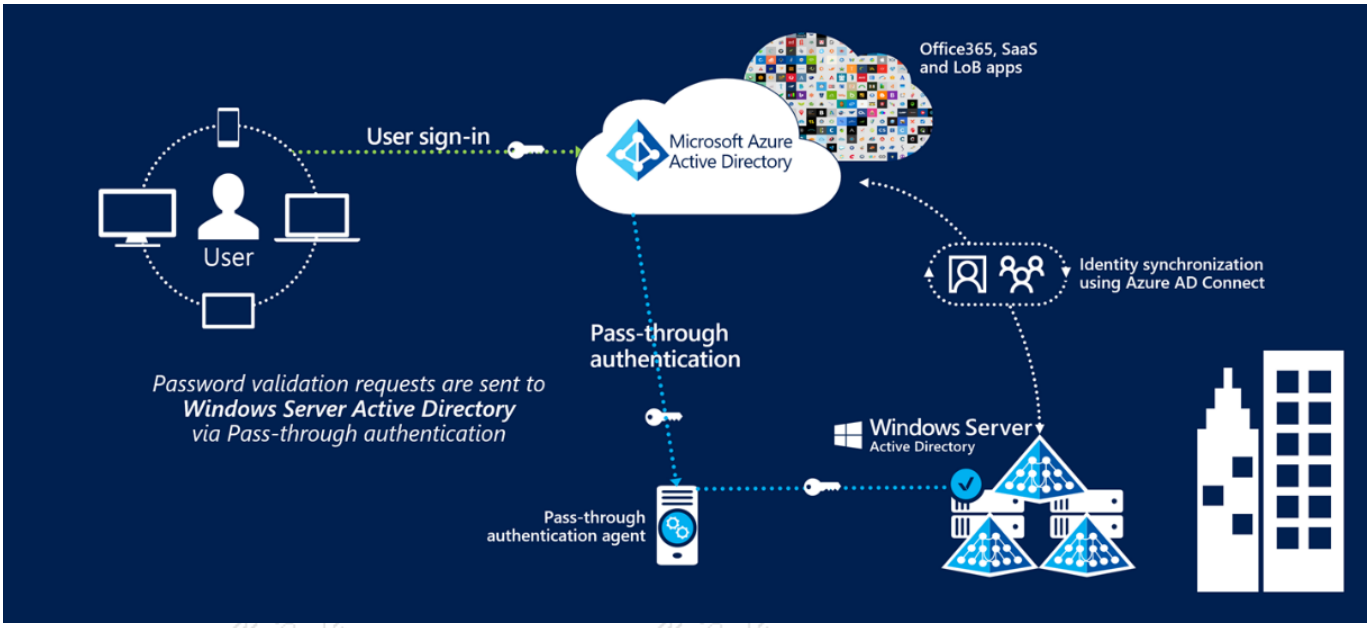


FIGURE 3.8: PASS-THROUGH AUTHENTICATION

The highlighted benefits of this solution are:

- A great user experience, the user uses the same passwords to sign into both on-premises and cloud-based applications.
- Users spend less time resolving password-related issues.
- Users can be enabled to use self-service password management from the Azure AD directly.
- Easy to deploy, there is no requirement for on-premises deployments or network configuration. Only a small agent needed on-premises.
- Secure storage of passwords since on-premises passwords are never stored in the cloud.
- No additional ports or network configuration is required since the agent communicates outbound, so no perimeter network is required.
- Takes advantage of Azure AD Conditional Access policies, including Multi-Factor Authentication (MFA)
- By installing additional agents, the service can become highly available.

This is a free feature available to all tiers of Azure AD and supports all web-based apps and those supporting modern authentication. Multi-forest environments are supported, although some routing changes may be required.

Lab: Deploying Infrastructure Workloads to Azure

Scenario

One of your clients wants to build a web application hosted on Internet Information Services (IIS) that will scale in response to increases and decreases in usage. The solution should minimize the amount of manual setup and maintenance work necessary for each virtual machine running IIS.

Objectives

- Deploy a VM using the PowerShell DSC extension.
- Deploy a VMSS using PowerShell DSC as part of the VM profile.

Lab setup

Estimated Time: 60 minutes

Virtual machine: **20535A-SEA-ARCH**

User name: **Admin**

Password: **Pa55w.rd**

The lab steps for this course change frequently due to updates to Microsoft Azure. Microsoft Learning updates the lab steps frequently, so they are not available in this manual. Your instructor will provide you with the lab documentation.

Exercise 1: Deploy a Virtual Machine using PowerShell DSC

Exercise 2: Deploy a Virtual Machine Scale Set using PowerShell DSC

Exercise 3: Cleanup Subscription

Review Question(s)

Module review and takeaways

Review Question(s)