**Updated Notes for D338**

**Skill 1.1 - Manage Azure Active Directory (Azure AD Objects)**

* **Cloud-only users are created and managed exclusively in Azure AD and their attributes can be updated directly in Azure AD. They can only be created through the Azure portal, Azure PowerShell, and the Azure CLI.**
* **Important Azure AD Built-in Roles:**
  + **Global Administrator –** manage access to all the administrative features in Azure AD.
  + **User Administrator –** create and manage different types of users and groups in Azure**.**
  + **Billing Administrator** – it can manage subscriptions, support tickets, make purchases, and monitor service health.
* Must be Global Administrator or User Administrator to create new users.
* **Only a username and the user’s name (given name and surname) are mandatory when creating a new user.**
* Groups are groups of objects that make role assignments and access permissions easier to manage. Group can contain groups, users, devices, or service principles.
* **When creating groups you must first select the type of group you are creating. You have two options:**
* **Security groups** allow you to share Azure resource access to a group of users, devices, or service principals.
* **Office 365 group** allows access to a shared mailbox, calendar, SharePoint site, and so on.
* Group **Membership** types -
  + **Assigned** – Allows you to add and remove users manually
  + **Dynamic User** – Allows you to use dynamic group rules to automatically add or remove members
  + **Dynamic Device** – Allows you to use dynamic group rules to automatically add and remove devices
* **Dynamic Groups require an Azure AD Premium P1 license otherwise the Membership Type option is unavailable and is set to Assigned.**
* You **can** change the membership type for a group after it has been created.
* **For both dynamic user and dynamic device-based groups, the rules associated with the group are evaluated on an ongoing basis. If a user or device has an attribute that matches the rule, that user or device is added to the group. If an attribute changes and the user or device no longer matches the criteria for group membership, the entity will be removed.**
* Ways to manage devices -
  + Browsing to your Azure AD tenant and select devices.
  + Through the devices blade for an individual user
* **To enable and disable devices you must be a global administrator.**
* Disabling a device prevents it from accessing Azure AD resources. This does not prevent the user from access resources in general, it only prevents the user from accessing resources from that disabled device
* Deleting a device prevents it from accessing your Azure AD resources and removes all details that are attached.
* **Uploading users in bulk is a three-step process**
  + **Download the CSV template**
  + **Edit the CSV with bulk update values**
  + **Upload the CSV and submit the operation**
* All users and admins can invite a guest.
* **Azure AD Join allows you to control devices, the applications installed and accessed from them, and how those applications interact with your corporate data.**
* Associating devices with Azure AD gives you three options –
  + Registering a device – Appropriate for personal devices
  + Joining a device – Useful for corporate-owned devices
  + Hybrid AD joined – Devices that are joined to your on-premises AD and are registered with your Azure AD tenant.
* **Devices associated with Azure AD allows you to manage a devices identity like SSO and securing access using conditional access. Can be managed independently of the user’s identity.**
* Azure AD Join is an extension of device registration that changes the local state of the device. When a device is Azure AD-Joined, users can sign into the device using an organizational account instead of a personal account.
* **Registration of devices in Azure AD can be combined with Mobile Device Management (MDM) solutions.**
* Non-hybrid Azure AD join is used for Windows 10 Pro and Windows 10 Enterprise.
* **Hybrid Azure AD Join can use Windows 10, Windows 2016, Windows 7, Windows 8.1, Windows 2008, Windows 2008 R2, Windows 2012, and Windows 2012 R2.**
* Self-Service Password Reset (SSPR) License Requirements
  + Password Change – Cloud-user only, included in all editions of Azure AD
  + Password Reset – Cloud-user only, Microsoft 365 Business Standard, Microsoft 365 Business Premium, Azure AD Premium P1, Azure AD Premium P2
  + Password Change/Unlock/Reset - Hybrid-users, Microsoft 365 Business Premium, Azure AD Premium P1, Azure AD Premium P2.
* **Azure AD requires two authentication methods by default for Self-Service Password Reset (SSPR) for admins.**
* Categories of Azure AD Roles -
  + **Azure AD-Specific Roles** – These roles grant permissions to manage resources within Azure AD only.
  + **Service-specific Roles** – For major Microsoft 365 services (Non-Azure AD).
  + **Cross-Service Roles** – Roles that span services. Global Administrator and Global reader. All 365 services honor these roles.
* **Azure AD License features**
  + **Azure AD Free** – User and group management, on-premises directory synchronization, basic reports, self-service password change for cloud users, and SSO across Azure, Microsoft 365, and many popular SaaS.
  + **Azure AD Premium P1** – Lets hybrid users' access both on-premises and cloud resources. Also supports advanced administration, such as dynamic groups, self-service group management, Microsoft Identity manager, and cloud write-back capabilities, which allow self-service password to reset for your on-premises users.
  + **Azure AD Premium P2** – Offers Azure Active directory identity protection to help provide risk-based Conditional Access to your apps and critical company data and Privileged Identity Management to help discover, restrict, and monitor administrators and their access to resources and to provide just-in-time access when needed.
* Built-in roles are used to delegate permissions to perform administrative tasks within Azure AD administrative units.
* **Administrative units** let you subdivide your organization into any unit that you want, and then assign specific administrators that can manage only the members of that unit. For example, you could use administrative units to delegate permissions to administrators of each school at a large university, so they could control access, manage users, and set policies only in the School of Engineering.
* Azure CLI can be used to create and update Azure AD groups
* **Email Address should be specified for each new guest account**

**Differences between Azure AD and RBAC Roles:**

* [**https://tutorialsdojo.com/azure-active-directory-ad-vs-role-based-access-control-rbac**](https://tutorialsdojo.com/azure-active-directory-ad-vs-role-based-access-control-rbac)

**Skill 1.2 - Manage Role-Based Access Control (RBAC)**

* **Role-based access control (RBAC) - Allows you to manage the entities, called “Security Principals”, that have access to Azure resources and the actions that those entities can perform.**
* RBAC access can be granted to users, groups, service principals, and managed identities through role assignments.
* **Fundamental Azure RBAC built-in roles:**
  + **Owner** – full access to all Azure resources.
  + **Contributor –** create and manage all types of resources in Azure.
  + **Reader –** a user with this role can only view Azure resources
  + **User Access Administrator –** it has permissions to manage user access to all types of resources.
* Azure RBAC is applicable to the management of resources created in the Azure Resource Manager (ARM) deployment model.
* **A security principal is an object that represents a user, group, service principal, or managed identity that is requesting access to Azure resources. You can assign a role to any of these security principals.**
* Service principal - A security identity used by applications or services to access specific Azure resources. You can think of it as a user identity (username and password or certificate) for an application.
* **A role is the definition of what actions are allowed and/or denied. RBAC is configured by selecting a role and associated the role with a security principal, such as a user, group, or service principal**
* Most privileged access rights take precedence when you have overlapping assignments.
* **Role definition contains the list of permissions or declared permissions and those permissions define what actions can or cannot be performed against a type of resource, such as read, write, or delete.**
* RBAC roles are used to manage access and allow or restrict users to Azure resources, while Azure AD administrative roles are used to allow or restrict admins to perform identity tasks, such as creating new users, resetting passwords, and so on.
* **Management groups are not applicable in all scenarios and in some cases a subscription will be the highest scope you will work with when applying role assignments.**
* Security Principals do not have access to Azure resources until a role assignment is made.
* **Role assignments can be created and removed by people with the Owner or User Access Administrator built-in roles.**
* Just as a blueprint allows an engineer or an architect to sketch a project's design parameters, **Azure Blueprints** enables cloud architects and central information technology groups to define a repeatable set of Azure resources that implements and adheres to an organization's standards, patterns, and requirements
* Azure Blueprints and resource locks can be used to make deny assignments at a child scope.
* **Limit of 5,000 custom roles per directory, limit of 2,000 role assignments per subscription. Custom roles can be created from existing built-in roles, starting from scratch, or with a JSON file to define the custom permissions.**
* Must have write permissions on all the items in a scope to create a custom role.
* **Deny Assignments are set and controlled by applying a resource lock for resources created through Azure Blueprints**
* Built-in roles cannot be modified.
* **A custom role definition is a collection of permissions that you add from a preset list. These permissions are the same permissions used in the built-in roles.**
* Role Assignments can be created and listed in the Portal, Azure AD PowerShell, or Microsoft Graph API.
* **Custom Roles can be created to provide a set of permissions that are not available when using built-in roles.**
* Built-in roles can be cloned and then modified for small tweaks to permissions.

**Skill 1.3 - Manage Subscriptions and Governance**

* **An Azure subscription, which forms the core of an Azure environment, is a foundational component of every Azure implementation. Every resource that you create in Azure resides in an Azure subscription, which is a billing boundary for Azure resources with per-resource, role-based access controls.**
* As you build and deploy services in Azure, you will create many types of resources. For instance, when creating your first virtual machine, you will also deploy many other resources including
  + A disk for the OS
  + A network interface for the VM
  + A virtual network and subnet for that network interface to bind to
  + A network security group (in a default portal configuration)
* **It is important to understand that many services in Azure create multiple resources, and how you manage those resources will be driven by organizational Policy and the lifecycle of your infrastructure hosted in Azure.**
* **A resource in Azure is a single service instance,** which can be a virtual machine, a virtual network, a storage account or any other Azure service.
* **Resource groups are logical groupings of resources or those single-service instances**
* Each resource in Azure can only exist in one resource group and resource groups cannot be renamed. There are no limitations to the types of resources that can be logically contained within a resource group, and there are no limitations on the regions in which resources must reside when in a resource group.

**Configure Azure Policies**

* Azure Policy is an Azure service that can be used to create, assign, and manage policies that enforce governance in your Azure environment. Includes the application of rules that allow or deny a given resource type, apply tags automatically, and even enforce data sovereignty.
* **Azure RBAC and Azure Policy are often used in combination. Where Azure RBAC controls individual user access, group access, and rights to your Azure environments at a specific scope, Azure Policy provides a mechanism to express how the environment is governed for all users at a specified scope regardless of any RBAC assignments.**
* **Azure RBAC is a default deny mechanism with an explicit allow mechanism, whereas Policy is a default allow mechanism with an explicit deny system.**
* Policy definition describes your desired behavior for Azure resources at the time resources are created or updated. Through a policy definition, you declare what resources and resource features are considered compliant within your Azure environment and what should happen when a resource is non-compliant.
* **Policy definitions are authored in JSON.**
* Policy definition contains these elements:
* Mode
* Parameters
* Display Name
* Description
* Policy Rule
* Logical Evaluation
* Effect
* **When managing resource groups—and in many cases the multiple Azure services that reside within them— Azure Policy with Policy definitions and Policy assignments can be used to govern those resources. Initiative definitions and initiative assignments can be used to govern those same resources, but instead of applying multiple Policy definitions and making multiple Policy assignments, you can package or group multiple definitions into a single initiative and then assign that initiative to your desired scope. Controlling resource groups with Azure Policy is done by scoping the assignment of Policy and initiatives. Recall that Azure Policy supports multiple scopes:-** 
  + **Management Groups**: Assignments scoped at the management group (either the Tenant Root Group or a child group) apply to all child resources in the management group including child management groups, all subscriptions, resource groups, and resources.
  + **Subscriptions**: Assignments scoped to a subscription apply to all child resources in the subscription resource groups and resources.
  + **Resource Groups**: Assignments scoped to a resource group apply to all child resources in the resource group.
* When creating policy scopes, it is possible to configure excluded scopes. Allows you to model your environment with rich declarations in the form of Policy definitions that are applied exactly as required by your organizations governance needs.

**Azure Resouce Locks**

* Azure Resource Locks (Sometimes called management locks) are used to prevent the accidental deletion or modification of resources. The two lock types are
  + **CanNotDelete** – Locks prevent the deletion of a resource. A CanNotDelete lock only prevents deletion of a resource and does not impede the modification of a resource
  + **ReadOnly** – Locks prevent users from modifying a resource, which includes updating or deleting a resource.
* **Resource locks, regardless of type, can be applied to the subscription, resource group, and resource scopes. When you apply a lock to a scope, the resources within that scope inherit the lock. This means that a lock applied to the resource group scope applies to all the resources in the resource group. Resource locks apply to all service instances and resources within a scope.**
* Lock inheritance varies based on the type of lock that is applied**. ReadOnly locks are inherited by child resources, while CanNotDelete locks are also inherited by child resources, but it has a side effect. If the CanNotDelete lock is applied to one of the resources in the resource group and you attempt to delete that resource group, it will fail.** When you try to delete the resource group, the operation tries to delete all the underlying resources first and won’t be able to delete the resource with CanNotDelete lock, hence the resource group deletion would also fail.

**Apply and manage tags on resources**

* Resource Tags allow you to apply custom metadata to your Azure resources to logically organize them and to build out custom taxonomies. A tag is a name and value pair.
* **Common tag types include the environment with which a resource is associated, a cost center or billing code, and resource owner.**
* Tags are also included in the billing data for Azure Enterprise Agreement (EA) subscriptions through the EA Portal and for non-EA subscriptions through the Account Portal at <https://account.azure.com/subscriptions>..
* Tags must be applied at the resource scope to be visible in detailed usage exports. Tags applied at the resource group scope are **not inherited by child resources.** This means that as you are applying tags to your resources in Azure, you should think about applying tags to each resource to have the clearest line of sight into your usage based on your organizational tags.
* **Azure Policy can be used to automatically apply tags to resources**
* Azure Tag limitations -
  + **Resource Support** – Not all resources support tags. This means that you will not be able to apply tags to everything in Azure. For example, management groups and generalized VMs don’t support tags.
  + **Number of tags** – Resources, resource groups, and subscriptions are limited to 50 tags. Each resource can have different tags
  + **Tag Names** – Tag names cannot exceed 512 characters. Storage account tag names are limited to 128 characters.
  + **Tag Values** – Cannot exceed 256 characters
  + **Virtual Machine Tags** – VMs cannot exceed 2048 characters for all tag names and values combined
  + **Tag Inheritance** – Tags are not inherited by child resources. Tags applied to a resource group are not applied to resources in that resource group
  + **Classic Resources** – Only available for resources created in the Azure Resource Manager model.
  + **Illegal Characters** – Tag names cannot contain these characters, <, >, %, &, \, ?, /
* **To apply tags to a subscription, rg, or resource, the user Must have write access (Contributor role or higher access) to apply tags to a subscription, resource group, or resource.**
* Tags can be applied in both an imperative manner and declaratively through Resource Manager templates.
* **Tags do not have inheritance, so if you need a tag to be applied to all resources in a resource group then each resource must be tagged individually.**
* Values used with the Update-AzTag Command when using the -Operation parameter:
  + **Replace** – Replaces the specified tags in the listed resources
  + **Merge** – Merges the newly specified tags with the existing ones and overrides the conflicts for the listed resources.
  + **Delete** – Deletes the specified tags from the listed resources.

**Create and manage resource groups**

* **When creating resource groups, it is important that you consider factors such as a single resource being associated with only one resource group at a time and the following:-**
  + A resource group **cannot** be nested in another resource group
  + You can add or remove a resource from a resource group at any time
  + You can move a resource from one resource group to another
  + A resource group can be used to scope access control
  + A resource group can be used to scope Policy
  + A resource in a resource group can interact with resources in another resource group
  + A resource group is created in a location (region). The location of a resource group specifies where the metadata for the resource group is stored.
  + Microsoft recommends that all resources in a resource group share the same lifecycle
  + It is not mandatory to have all Azure resources belong to a resource group. Resources are deployed to a subscription, tenant, or management group exist outside of resource groups.

**Move resources across resource groups**

* While moving resources from one resource group to another, the resources will be locked. Both write and delete operations to the Azure resource will be blocked, but the underlying service will continue to function.
* **Moving resources between subscriptions requires both subscriptions to be associated with the same Azure AD tenant. If the subscriptions do not belong to the same tenant, then you can update the target subscription to use the source Azure AD tenant by transferring ownership of the subscription to another account.**
* **A single move operation in the Resource Manager cannot move more than 800 resources.**
* You can delete a resource group with the Azure Portal, Azure PowerShell, the Azure CLI, or the REST API.
* **Azure subscriptions have controls available that govern access to the resources within a subscription, govern cost through quotas and tagging, and govern the resources that are allowed in an environment with Azure Policy.**
* If you are moving resources between subscriptions, you must also be mindful of resource quotas. For example, if you are moving many virtual machines, you will need to make sure that the target subscription has enough vCPUs available or the move operation will fail. Make sure you validate any quotas prior to moving a resource.

**Manage Azure Subscriptions**

* There are multiple ways to obtain an Azure subscription, and a wide range of subscription types (or offers). Some common types include the following:
* **Free trial**
* **Pay-As-You-Go/Web Direct**
* **Visual Studio/MSDN subscriptions**
* **Microsoft Resellers**
* **Cloud Solution Provider**
* **Microsoft Open Licensing**
* **Enterprise Agreements**
* **The capabilities of each subscription are similar in that each subscription type allows you to create and manage resources. Some subscription types have restrictions on supported resource types and locations.**

**Assigning administrator permissions**

* **Azure has many different roles for managing access to Azure resources. These include classic subscription administrative roles like Account Administrator, Service Administrator, or Co-Administrator, as well as Azure role-based access controls (RBAC) that are available in Azure Resource Manager (ARM). When managing access to Azure subscriptions and resources, it is recommended to use Azure RBAC roles whenever possible.**
* **Classic subscription Administrators have full access to an Azure subscription. They can manage resources through the Azure Portal, Resource Manager API’s (including PowerShell and the CLI).**
* The account that is signed up for an Azure subscription is automatically set as both the account administrator and the service administrator.
  + Access to the account center and creation of a new Azure subscription and billing changes can only be performed by the **Account administrator**.
* **There can only be one account administrator per account and one service administrator per subscription**.
* Co-Administrators have the same level of access as the **service administrator** but cannot change the association of subscriptions to Azure directories. Can be up to 200 co-administrators per subscription.
* **Users with Service Administrator and Co-Administrator roles have the same access as a user who is assigned the Azure RBAC owner role at the subscription scope.**
* Azure RBAC has more than 70 built-in roles, but there are four foundational roles-
  + **Owner** – Full access to all resources. Delegate access to others. The Service Administrator and Co-Administrators are assigned the Owner role at the subscription scope. Applies to all resource types.
  + **Contributor** – Create and manage all of types of Azure resources. Cannot grant access to others. Applies to all resource types.
  + **Reader** – View Azure resources. Applies to all resource types.
  + **User Access Administrator** – Manage user access to Azure resources.

**Configure management groups**

* Management groups allow you to apply governance across subscriptions, including the application of common RBAC controls and the application of Azure policy. Benefits of management groups
  + **Reduce overhead**
  + **Enforcement**
  + **Reporting**
* **Management groups form a hierarchy that is up to six levels deep, excluding the root and subscription levels.**
* Management groups introduce an additional scope above a subscription. When applied at the management group scope, each subscription under the management group inherits the RBAC and Policy assignments of the management group

**Important RBAC and management groups**

* **RBAC applied at the management group level is inherited by all the child resources within the scope of the management group (subscriptions, resource groups, and resources). For instance, if you add a user as an Owner at the management group scope, that user will become an Owner in all the subscriptions associated with the management group.**

**Configure cost management**

* **In Azure, there are several types of quotas that are applicable to subscriptions, including resource quotas and spending quotas.**
* Resource Quotas (Limits) - Azure administrators can view the current consumption and usage of resources within an Azure subscription and understand how that consumption can be affected by Azure resource limits. Administrators can also request quota increases for certain resource types. For instance, the number of cores available for virtual machines is limited to 20 per region by default. This limit can be increased by submitting a request to Microsoft support.
* **Spending Quotas – Spending quotas allow administrators to set alerts within an Azure subscription by configuring budgets to inform the business when their Azure spending has hit a certain threshold. While a resource limit can stop resources from being created (for example, there are not enough cores available to the subscription in the desired region), a spending quota acts as an alerting mechanism and does not stop resources from being created or consumed.**
* Tags in Azure Resource Manager allow consumers of Azure to logically categorize Azure resource groups and Azure resources. As resources are tagged, they can then be queried and tracked based on the associated tags. Tags are a crucial component to implement chargeback within an Azure subscription. As resources are tagged, they can then be queried and tracked based on the associated tags.
* **Submitting a request to increase a quota is only submitting a support request to Microsoft.**
* The consumption of resources within a subscription against a resource quota can also be viewed with PowerShell. There are multiple cmdlets available in the Az (formerly AzureRm) PowerShell modules for querying per-service quota usage.

**Configure cost center quotas**

* One of the best ways to drive accountability is to make sure that the consumers of Azure resources understand their cost. Budgets in Azure Cost Management provide Azure customers subscriptions under many offer types with the ability to proactively manage cost and monitor Azure spend over time at a subscription level. Budgets are a monitoring mechanism only with set thresholds and notification rules. Users must have at least read access (Reader rights) to a subscription to view budgets and must have Contributor (or higher) rights to create and manage budgets.
  + Cost Management Contributor and Cost Management Reader are two specialized roles that can be used to grant principals access to Cost Management data.
* **Budgets can be created at the subscription, management group, and resource group scope if necessary.**
* Budget alerts can also leverage the same Action Groups that Azure Monitor supports. Action groups are a collection of notification preferences and are discussed in detail later in this doc.
* **By default, you will be creating a budget at the subscription scope, but budgets can also be created at the management group as well as resource group scope if necessary.**

**Monitor and report spend**

* While Azure Advisor and its cost recommendations provide one method for monitoring spend and unused resources, Azure has many other tools that can help you monitor the cost of your resources and report on that cost.

There are several considerations that you must account for when reporting on the cost associated with your Azure resources:

* + Azure services are available to customers in 140 countries worldwide.
  + Billing is supported across 24 major currencies.
  + Azure subscriptions are billed monthly. If you are paying by credit card, note that pre-paid cards and virtual credit cards are not accepted.
  + You can also pay for Azure by monthly invoice. To apply for invoice payment, raise an appropriate billing support ticket from the Azure management portal. Processing the request takes 5-7 days, depending on the time required for the necessary credit checks. Invoice payment is only available to business customers, and once a subscription has been moved to invoice payment, it cannot be moved back to credit card payment. If you choose invoice payment, you will get an invoice, and you will pay with a wire transfer or check.
  + Customers on an Enterprise Agreement (EA) can add up-front commitments to Azure and then create multiple subscriptions under the agreement, which draw from the monetary commitment.
  + EA commitments are billed immediately, and then consumed throughout the year against the Azure resources consumed.
  + If the committed spend is exceeded, the extra spend, or “overage,” is billed at the same discounted EA rate. Billing for overage is annual if the overspend is under 50 percent of the commitment, or quarterly if over 50 percent.
  + Azure Marketplace third party services are billed separately with a potentially different billing period, separate invoice, and separate credit card charge. Each service has its own billing model, which will be described in the Azure portal at the time of purchase. These range from pay-as-you-go per-minute billing to fixed monthly charges. Some services also offer a “bring your own license” model, which must provide a license purchased separately prior to using the service.
* **There are three portals that are used to manage Azure subscriptions that are relevant for billing and cost management. They are:**
* **The EA Portal** available at https://ea.azure.com. This is available only to customers with an Enterprise Agreement and is used for managing spend across one or more subscriptions.
* **The Account Portal** at https://account.azure.com/subscriptions. This is available for all subscriptions and accessible by Account owners. It is used to manage subscriptions, payment methods, and spending limits.
* **The Azure portal** at https://portal.azure.com. This is available for all subscriptions and includes Azure Cost Management.
* Cost Management includes features for performing cost analysis, setting per-subscription budgets and alerts, setting recommendations for optimization, and exporting cost management data to perform deeper analysis. Access to Cost Management service is dictated by scopes. A user must have at least read access to the one of the cost management scopes.
* **If you have access to more than one scope, you can filter by scope and begin interacting with the data. The data in a view can be downloaded from Cost Analysis as a CSV. Any filtering that you have applied, including groupings, are applied to the file.**

**Chapter Summary -**

* + Windows 10 can be added to Azure AD as a device to be managed, enabling BYOD or corporate cloud only deployments with Azure AD join.
  + Azure AD Join enables administrators to manage device identity independently of users.
  + Downstream Windows clients can be managed through Azure AD using Azure AD hybrid join.
  + Conditional access is a feature of Azure AD which allows administrator to control access to cloud applications through additional checks such as user location, the device the user is accessing the cloud app from, and more.
  + Multiple Azure AD tenants can be created and managed through Azure. This includes creating new directories and deleting existing directories.
  + Users and groups can be created through Azure portal, PowerShell, Azure CLI, and the Graph API. Users and groups can be managed in bulk with tools like PowerShell.
  + Self-service password resets can be combined with the password writeback features of Azure AD connect to allow users to reset their passwords from the cloud while adhering to on-premises password standards.
  + Tags in Azure can be used to logically organize resources by categories. Each tag is a name and a value pair. Tags can be shared across multiple resources and enforced with Azure Policy.
  + Azure Policy is a service that lets you create, manage, and apply Policy to Azure resources at a subscription, resource group, or resource level. Policies enforce different rules over your Azure resources, so those resources remain compliant with your organizations standards.
  + Role-based access control allows you to grant users, groups, and service principals access to Azure resources at the subscription, resource group, or resources with RBAC inheritance. The three core roles are Owner, Contributor, and Reader.
  + Resource Group template is a JSON file that allows you to declaratively describe a set of resources. These resources can then be added to a new or existing resource group.
  + Templates can simplify orchestration because you only need to deploy the template to deploy all your resources.
  + Template allows you to configure multiple resources simultaneously and use variables/parameters/functions to create dependencies between resources.
* Azure Policy can be used to enforce tagging rules and conventions by automatically applying or requiring tags to resources.
* Tags are used to add supplemental information to the resources, such as environment purpose.
* Management groups can contain multiple subscriptions and support role assignments.
* Tags at the resource level would be automatically included in the billing report available from the Azure portal, which includes per-resource cost.

**Chapter 2: Implement and manage storage**

**Skill 2.1 - Secure Storage**

* An Azure Storage account is an entity you create that is used to store Azure Storage data objects such as blobs, files, queues, tables, and disks. Data in Azure Storage account is durable and highly available, secure, massively scalable, and accessible from anywhere in the world over HTTP or HTTPS

**Configure network access to the storage accounts**

* Storage Accounts are managed through Azure Resource Manager. Management operations are authenticated and authorized using Azure Active Directory and RBAC. Each storage account service exposes its own endpoint used to manage the data in that storage service (blobs in Blob Storage, entities in tables, and so on). These service-specific endpoints are not exposed through Azure Resource Manager; instead, they are (by default) Internet-facing endpoints.
* **Access to these Internet-facing storage endpoints must be secured, and Azure Storage provides several ways to do so. In this section, we will review the network-level access controls: the storage firewall and service endpoints..** 
  + **Endpoints**
    - **Public endpoint (all networks)**
    - **Public endpoint (selected networks)**
    - **Private endpoint (VNet network)**

**Storage Firewall**

* Storage Firewall allows you to limit access to specific IP addresses or an IP address range. Applies to all storage account services (blobs, tables, queues, and files).
* **When creating a storage firewall, you must use public Internet IP address space. You cannot use IPs in the private IP address space. Service endpoints are used to restrict access to specific subnets within an Azure VNet.**
* The storage firewall includes an option to allow access from trusted Microsoft services. These services include Azure Backup, Azure Site Recovery, and Azure Networking.

**Virtual network service endpoints**

* **In some scenarios, a storage account is only accessed from within an Azure virtual network. In this case, it is desirable from a security standpoint to block all Internet access. Configuring virtual network service endpoints for your Azure Storage accounts allows you to remove access from the public Internet and only allow traffic from a virtual network for improved security.**
* Another benefit of using service endpoints is optimized routing. Service endpoints create a direct network route from the virtual network to the storage service. If forced tunneling is being used to force Internet traffic to your on-premises network or to another network appliance, requests to Azure Storage will follow that same route. By using service endpoints, you can use direct route to the storage account instead of the on-premises route, so no additional latency is incurred.
* **Service Endpoints are configured in two steps**
  + **From the virtual network subnet. Creates the route from the subnet to the storage service but does not restrict which storage account the virtual network can use.**
  + **Configuring which virtual networks can access a particular storage account.**

**Blob Storage access levels**

* Storage accounts support an additional access control mechanism that is limited only to Blob Storage. By default, no public read access is enabled for anonymous users, and only users with rights granted through RBAC or with the storage account name and key will have access to the stored blobs. To enable anonymous user access, you must change the container access level. The supported levels are as follows:
  + **Private** – With this option, only the storage account owner can access the container and its blobs.
  + **Blob** – With this option, only blobs within the container can be accessed anonymously
  + **Container** - With this option, blobs and their containers can be accessed anonymously.
* **You can change the access level through the Azure portal, Azure PowerShell, Azure CLI, programmatically using the REST API, or by using Azure Storage Explorer. The access level is configured separately on each blob container.**
* **Shared Access Signature token (SAS Token)** is a URI query string parameter that grants access to specific containers, blobs, queues, and tables. Use an SAS token to grant access to a client that should not have access to the entire contents of the storage account, but still requires secure authentication. By distributing an SAS URI to these clients, **you can grant them access to a specific resource, for a specified period of time, and with a specified set of permissions**. **Frequently, SAS tokens are used to read and write the data to users’ storage accounts. Also, SAS tokens are widely used to copy blobs or files to another storage account.**
* **When dealing with SAS tokens, you must use only the HTTPS protocol. Because active SAS tokens provide direct authentication to your storage account, you must use a secure connection, such as HTTPS, to distribute SAS token URIs.**

**Create and configure storage accounts**

* Azure Storage accounts provide a cloud-based storage service that is highly scalable, available, performant, and durable. Within each storage account, a number of separate storage services are provided:
  + **Blobs** – Provides a highly scalable service for storing arbitrary data objects such as text or binary data. Three blob types
    - Page Blobs – Used to store VHD files when deploying unmanaged disks
    - Block Blobs
    - Append Blobs
  + **Tables** – Provides a NoSQL-style store for storing structured data. Unlike a relational database, tables in Azure storage do not require a fixed schema, so different entries in the same table can have different fields
  + **Queues** – Provides a reliable messaging queueing between application components.
  + **Files** – Provides managed file shares that can be used by Azure VMs or on-premises servers.
  + **Disks** – Provides a persistent storage volume for Azure VM which can be attached as a virtual hard disk.
* **When creating a storage account, there are several options that must be set: Performance Tier, Account Kind, Replication Option, and Access Tier.**

**Naming storage accounts**

* While naming an Azure Storage Account, you need to remember these points:
  + Storage account name must be unique across all existing storage account names in Azure
  + The name must be between 3 to 24 characters and can contain only lowercase letters and numbers.
* **When creating a storage account, you must choose between the Standard and Premium performance tiers. This setting cannot be changed later.**
* **Performance Tiers –** 
  + **Standard** – Supports all storage services. Blobs, tables, files, queues, and unmanaged Azure virtual machine disks. It uses magnetic disks to provide cost-efficient and reliable storage.
  + **Premium** – Designed to support workloads with greater demands on I/O and is backed by high-performance SSD disks. Only supports General Purpose accounts with Disk Blobs and Page Blobs. It also supports Block Blobs or Append Blobs with BlockBlobStorage accounts and files with FileStorage accounts. Only supports LRS for general-purpose storage accounts. Supports LRS and ZRS for both BlockBlobStorage and FileStorage accounts.

**Account kind**

* Possible Account Kind for each tier
  + **Standard tier supports three values** General-Purpose V1 and V2, and BlobStorage.
  + **Premium tier supports** **four values** Genera-Purpose V1 and V2, BlockBlobStorage, and FileStorage.
* **Blob storage account is a specialized storage account used to store Block Blobs and Append Blobs. You can’t store Page Blobs in these accounts.**
* **Only General-Purpose V2 and Blob Storage accounts support the Hot, Cool, and Archive access tiers.**
* **General-Purpose V1 and Blob Storage accounts can both be upgraded to General-Purpose V2. This operation is irreversible. No other changes to the Account Kind are supported**
* Storage Account Types and Supported Features –
  + **General-Purpose V2** – Supports Blob, File, Table, and Queue, Supports Unmanaged disk (Page Blob). Standard and performance tiers. Support for Hot, Cool, and Archive access tiers. Replication options at LRS, ZRS, GRS, RA-GRS, GZRS, and RA-GZRS.
  + **General-Purpose V1** – Supports Blob, File, Table, and Queue, Supports Unmanaged disk (Page Blob). Standard and performance tiers. **N/A for supported access tiers (Hot, Cool, Archive)**. Supports LRS, GRS, and RA-GRS replication options
  + **Blob Storage** – Supports Blob, block and append blobs only. No unmanaged disk support. Standard performance tier. Supports Hot, Cool, and Archive access tiers. Replication options are LRS, GRS, and RA-GRS.
  + **Blob Block Storage** - Supports Blob, block and append blobs only. No unmanaged disk support. Premium performance tier. N/A for access tiers. Replication options are LRS and ZRS.
  + **File Storage** – Supports file only service. No unmanaged disk support. Supports the premium performance tier. N/A access tiers. Supported replication options are LRS and ZRS.

**Replication Options**

* When you create a storage account, you can also specify how your data will be replicated for redundancy and resistance to failure. These options are listed below
  + **Locally Redundant Storage (LRS)** – Three synchronous copies of your data within a single datacenter. Available for general-purpose or Blob storage accounts at both the standard and performance tiers.
  + **Zone Redundant Storage (ZRS)** – Makes three synchronous copies to three separate availability zones within a single region. **Available for General-Purpose V2** storage accounts only, at the Standard Performance tier only. Also available for BlockBlobStorage and FileStorage
  + **Geographically Redundant Storage (GRS)** – Same as LRS (three local copies), plus three additional asynchronous copies to a second datacenter hundreds of miles away from the primary region. Data replication typically occurs within 15 minutes, although no SLA is provided. Available for General-purpose or Blob storage accounts, at the Standard Performance tier only.
  + **Read Access Geographically Redundant Storage (RA-GRS)** – This has the same capabilities as GRS, plus you have read-only access to the data in the secondary datacenter. Available for General-purpose or Blob storage accounts, at the Standard Performance tier only.
  + **Geographically Zone Redundant Storage (GZRS)** – The same as ZRS (Three synchronous copies across multiple availability zones), plus three additional asynchronous copies to a second datacenter hundreds of miles away from the primary region. Data replication typically occurs within 15 minutes, although no SLA is provided. Available for General-purpose v2 storage accounts only, at the standard performance tier only
  + **Read Access Geographically zone redundant storage (RA-GZRS)** – This has the same capabilities as GZRS, plus you have read-only access to the data in the secondary datacenter. Available for General-Purpose V2 storage accounts only at the standard performance tier only.
* **When creating a storage account via the Azure portal, the replication and performance tier options are specified using separate settings. When creating an account using Azure PowerShell, the Azure CLI, or via a template, these settings are combined within the SKU setting. For example, to specify a Standard storage account using locally redundant storage using the Azure CLI, use --sku Standard\_LRS.**

**Access Tiers**

* **Azure Blob Storage supports three access tiers: Hot, Cool, and Archive. Each represents a trade-off of performance, availability, and cost. There is no trade-off on the durability (probability of data loss), which is extremely high across all tiers.**
* **Azure Blob Storage Tiers** –
  + **Hot**: This access tier is used to store frequently accessed objects. Relative to other tiers, data access costs are low while storage costs are higher.
  + **Cool**: This access tier is used to store large amounts of data that is not accessed frequently and that is stored for at least 30 days. The availability SLA is lower than for the Hot tier. Relative to the Hot tier, data access costs are higher and storage costs are lower.
  + **Archive**: This access tier is used to archive data for long-term storage, that is accessed rarely, can tolerate several hours of retrieval latency, and will remain in the Archive tier for at least 180 days. This tier is the most cost-effective option for storing data, but accessing that data is more expensive than accessing data in the Hot or Cool tiers.
* **ARCHIVE TIER supportability - Currently, the archive tier is not supported for ZRS, GZRS, or RA-GZRS accounts.**

**Generate shared access signatures**

* There are a few different ways you can create an SAS token. A SAS (Shared Access Signature) token is a way to granularly control how a client can access data in Azure storage account. You can also use an account-level SAS to access the account itself. You can control many things such as what services and resources the client has access to, how long the token is valid for, and more.
* **The simplest way to create one is by using the Azure portal. Also, you can create SAS tokens using Storage Explorer or the command-line tools (or programmatically using the REST APIs/SDK).**

**Using shared access signatures**

* Each SAS token is a query string parameter that can be appended to the full URI of the blob or other storage resource for which the SAS token was created.
  + The following example shows the combination in more detail. Suppose the storage account name is examref, the blob container name is examrefcontainer, and the blob path is sample-file.png. The full URI to the blob in storage is:
    - <https://examrefstorage.blob.core.windows.net/examrefcontainer/sample-file.png>
  + The combined URI with the generated SAS token is
    - https://examrefstorage.blob.core.windows.net/examrefcontainer/sample-file.png**?sv=2019-10-10&ss=bfqt&srt=sco&sp=rwdlacupx&se=2020-05-08T08:50:14Z&st=2020-05-08T00:50:14Z&spr=https&sig=65tNhZtj2lu0tih8HQtK7aEL9YCIpGGprZocXjiQ%2Fko%3D**

**Using account-level SAS**

* **You can create the SAS at the storage account–level, too. With this SAS, you can manage all the resources belonging to the storage account. You can also perform write and delete operations for all the resources (blobs, tables, and so on) of the storage account.**
* Currently, stored access policy is not supported for account-level SAS.
* **SAS can be used at the storage account-level as well. Allows management of all the resources belonging to the storage account.**

**Using user delegation SAS**

* User delegation SAS using Azure AD credentials is also possible. The user delegation is only supported by Blob Storage, and it can grant access to containers and blobs.

**Using a stored access policy**

* **An SAS token incorporates the access parameters (start and end time, permissions, and so on) as part of the token. The parameters cannot be changed without generating a new token, and the only way to revoke an existing token before its expiry time is to roll over the storage account key used to generate the token or delete the blob. In practice, these limitations can make standard SAS tokens difficult to manage.**
* Stored access policies allow the parameters for an SAS token to be decoupled from the token itself. The access policy specifies the start time, end time, and access permissions, and the access policy is created independently of the SAS tokens. SAS tokens are generated that reference the stored access policy instead of embedding the access parameters explicitly.
* **You can only have a max of five stored access policies on a container, table, queue, or file share.**
* You can revoke the SAS token by deleting the access policy, renaming it (changing the identifier), or changing the expiry time.

**Manage access keys**

* The simplest way to manage access to a storage account is to use access keys.
* **Access Keys allow full access to all data in all services within the storage account. You can create, read, update, and delete container, blobs, tables, queues, and file shares. You have full administrative access to everything other than the storage account itself (Cannot delete storage account or change settings on the storage account)**
* Applications will use the storage account name and key for access to Azure Storage.
* **Each storage account has two access keys. This allows you to modify applications to use the second key instead of the first and then regenerate the first key. This technique is known as “key rolling,” and it allows you to reset the primary key with no downtime for applications that directly access storage using an access key. Rolling a storage account access key will invalidate any SAS tokens that were generated using that key.**

**Managing access keys in Azure Key Vault**

* It is important to protect the storage account access keys because they provide full access to the storage account. Azure Key Vault helps safeguard storage account access keys as well as cryptographic keys and secrets used by cloud applications and services, such as authentication keys, storage account keys, data encryption keys, and certificate private keys.
* **Keys in Azure Key Vault can be protected in software or by using hardware security modules (HSMs). HSM keys can be generated in place or imported. Importing keys is often referred to as bring your own key or BYOK.**
* Accessing and unencrypting the stored keys is done by a developer, although keys from Key Vault can also be accessed from ARM templates during deployment. You can manage storage account keys with key vault using Azure PowerShell or CLI.

**Configure Azure AD Authentication for a storage account**

* **Azure Active Directory (AAD) authentication has been recently added as an authorization mechanism for Azure Storage. Azure Blobs and Queues are supported for Azure AD authentication. Azure Table storage is not supported right now.**
* Storage Accounts that are created with the Azure Resource Manager deployment model only support Azure AD authorization.
* **AAD authentication enables customers to leverage Azure’s RBAC for granting the required permissions to a security principal (users, groups, and applications) down to the scope of an individual blob container or queue. Azure AD returns an OAuth 2.0 token to security principal, which can be used for authorization against Azure Storage (Blob or queue)**
* SAS signatures can be signed by Azure AD credentials to provide access to storage accounts.
* **Managed Service Identity (MSI) – Can be used to access blobs or queues from an Azure entity like Azure VM, virtual machine scale set, or an Azure Functions app.**

**RBAC roles for blobs and queues**

* There are few built-in RBAC roles available in Azure for authorizing access to Blob and Queue Storage.
  + **Storage Blob Data Owner**: Sets ownership and manages POSIX access control for Azure Data Lake Storage Gen2.
  + **Storage Blob Data Contributor**: Grants read/write/delete permissions for Blob Storage.
  + **Storage Blob Data Reader**: Grants read-only permissions for Blob Storage.
  + **Storage Queue Data Contributor**: Grants read/write/delete permissions for Queue Storage.
  + **Storage Queue Data Reader**: Grants read-only permissions for Queue Storage.
  + **Storage Queue Data Message Processor**: Grants peek, retrieve, and delete permissions to messages in queues.
  + **Storage Queue Data Message Sender**: Grants add permissions to messages in queues.

**Resource scope for blobs and queues**

* It is also important to determine the scope of the access for security principal before you assign an RBAC role. You can narrow down the scope to the container or queue level. Below are the valid scopes:
  + **Container** – All blobs inside the container, the container properties, and the metadata will inherit the role assignment when this scope is selected
  + **Queue** – All the messages inside the queue, as well as queue properties and metadata will inherit the role assignment when this scope is selected.
  + **Storage Account** – Under this scope, the role assignment will be applicable at the storage account level. **All the containers, blobs, queues, and messages within the storage account** will inherit the role assignment when this scope is selected.
  + **Resource Group -** Under this scope, the role assignment will be applicable at the resource group level.
  + **Subscription -** Under this scope, the role assignment will be applicable at the subscription level.

**Configure access to Azure Files**

* **Azure Files provides managed file shares that are accessible over the SMB protocol. Azure file shares leverage Kerberos tokens to authenticate a user or application to access the file shares.**
* Azure Files – Two types of identity-based authentication
  + On-premises Active Directory Domain Services (AD DS)
  + Azure Active Directory Domain Services (Azure AD DS)
* **Steps for enabling AD DS Authentication to Azure Files**
  + **Enables AD DS authentication on your storage account**
  + **Assign share-level access permissions to an Azure AD identity**
  + **Assign directory/file-level permissions using Windows ACLs**
  + **Mount the Azure File Share**
  + **Update the password of your storage account identity in AD DS**
* Shared Access Signature token provides secure, delegated access to resources in your Azure storage account
* **To access Azure Files by using SAS, you must use the REST method.**
* Storage account access keys are 512-bit keys that can be used to authorize access to data in your storage account

**Skill 2.2 – Manage Storage**

* Azure Import/Export service allows you to ship data into or out of an Azure Storage account by physically shipping disks to an Azure datacenter.
  + Export service is only available for Blob storage. After receiving the disks from Microsoft, you will need to retrieve the Bit Locker keys from the Azure portal to unlock the disks.
  + Import service is available for Blob storage and Azure Files.
* **The first step to import data using the Azure Import/Export service is to install the Microsoft Azure Import/Export tool known as the WAImportExport tool. Two versions of this tool, version 1 is recommended for Azure Blob Storage and Version 2 is recommended for Azure Files.**
* Requirements and limitations of the Azure Import/Export Jobs tool -
  + Windows 7, Windows Server 2008 R2, or a later OS version is required. This tool only works with 64-bit operating systems and might not work with 32-bit operating systems.
  + The tool also requires a .NET Framework 4.5.1 or later and BitLocker
  + All storage account types are supported (General-Purpose V1 and V2, and Blob Storage)
  + Block, Page, and Append blobs are supported for both import and export.
  + Azure File service is only supported for import jobs but not export jobs.
* **A single import/export job can have a maximum of 10 HDDs and SSDs and a mix of HDDs and SSDs of any size.**
* The second step to import data is to prepare the drives using the WAImportExport tool. When preparing the drive, the first session requires several parameters, such as the destination storage account key, the BitLocker key, and the log directory.
* **Azure Import/Export tool creates a journal file that contains the information necessary to restore the files on the drive to the Azure Storage account, such as mapping a folder or file to a container, blob, or files. Each drive used in the import job will have a unique journal file that is created by the tool.**
* Once the drive preparation is complete, the third step in the import process is to create an import job through the Azure portal.
* **Having created the import job, the fourth step in the import process is to physically ship the disks to Microsoft using a supported courier service with a tracking number for your package**.

**Install and use Azure Storage Explorer**

* Azure Storage Explorer is a cross-platform application designed to help you quickly manage one or more Storage Accounts. It can be used with all storage : Blob Storage, Azure Tables, Queue Storage, and Azure Files. In addition, Azure Storage Explorer also supports the CosmosDB and Azure Data Lake Storage services.

**Connecting Storage Explorer to Storage Accounts**

* After Storage Explorer is installed, you can connect to Azure Storage in one of five different ways
  + **Add an Azure account** – This option allows you to sign in using a work or Microsoft account and access all your storage accounts via role-based access control.
  + **Using a connection string** – This option requires you to have access to the connection string of the storage account. The connection string is retrievable by opening the storage account blade in the Azure portal and clicking Access Keys.
  + **Use a Shared Access Signature URI** – A shared Access signature provides access to a storage account without requiring an account key to be shared. Access can be restricted for example, to read-only access for Blob Storage for one week only.
  + **Using a Storage Account Name and Key** – This option requires you to have access to the storage account name and key. These values can be accessed from the Azure portal under Access Keys.
  + **Attach to a Local Emulator** – Allows you to connect to the local Azure Storage emulator as part of the Microsoft Azure SDK.

**Using Storage Explorer**

* Storage Explorer Operations - Azure Storage Explorer provides an intuitive GUI interface for each storage service. The below info summarizes the supported operations for each service.
  + **Blob Containers** – Create, rename, copy, delete, control public access level, manage leases and create and manage shared access signatures and access policies
  + **Blobs** – Upload, download, manage folders, rename and delete blobs, copy blobs, create and manage blob snapshots, change blob access tier, and create and manage shared access signatures and access policies
  + **Tables** – Create, rename, copy, delete, and create and manage shared access signatures and access policies.
  + **Table Entities** – Import, export, view, add, edit, delete, and query
  + **Queues** – Create. Delete, create and manage shared access signatures and access policies
  + **Messages** – Add, view, dequeue, and clear all messages.
  + **File Shares** – Create, rename, copy, delete, create and manage snapshots, connect a VM to a file share, and create and manage shared access signatures and access policies
  + **Files** – Upload folders or files, download folders or files, manage folders, copy, rename, and delete files.

**Storage blob copy**

* **The Azure Storage Explorer can be used to perform a storage blob copy. To copy between storage accounts, navigate to the source storage account, select one or more files, and click the Copy button on the toolbar. Next, navigate to the destination storage account, expand the container that you want to copy to, and click Paste from the toolbar**.

**Copy data by using AzCopy**

* AzCopy is a command-line utility that you can use to perform large-scale bulk transfer of data to and from Azure Storage. AzCopy performs all the operations asynchronously and can run simultaneously. It’s fault tolerant so if the operation is interrupted it can resume from where it left off once the issue is resolved.
* **The latest version of AzCopy lets you take incremental backups of blobs and keep it synchronized in order to contain the same version of data.**
* AzCopy can be added to the system path so that you can run AzCopy from any of the folders from your system while using it in Windows PowerShell.
* **Storage Explorer is a GUI which uses AzCopy to perform all its data transfer operations in the backed.**
* AzCopy supports Azure login, service principal, SAS token, access key, managed identity, and so on as authentication types. Use *azcopy login*

**Upload/download the data using AzCopy**

* AzCopy can be used to upload data to an Azure Blob Storage. Only condition is that the storage account and destination container should already exist.
* **AzCopy can also be used to copy between storage accounts.**

**Async blob copy**

* The AzCopy application can also be used to copy between storage accounts. The following example shows how to copy the blob from source storage account’s container to destination storage account’s container using SAS token.
  + AzCopy copy [https://examref.blob.core.windows.net/ srccontainer/[blob-path]?<sas token>](https://examref.blob.core.windows.net/%20srccontainer/%5bblob-path%5d?%3csas%20token%3e)” "https://examrefdest.blob.core.windows.net/destcontainer/[blob-path]?<sas token>"

**Sync blob copy**

* **You can use the azcopy sync command to do synchronized copy between two blob containers. This command synchronizes the contents of a destination container with a source container by copying blobs if the last modified time of a blob in the destination is earlier than that of the corresponding blob in the source. By default, the recursive flag is true for the sync command and copies all subdirectories:**
  + **azcopy sync “**[**https://examref.blob.core.windows.net/srccontainer/?<sas token>**](https://examref.blob.core.windows.net/srccontainer/?%3csas%20token%3e)**” "https://examref.blob.core.windows.net/destcontainer/"**

**Changing storage account replication mode**

* Storage accounts can be moved freely between LRS, GRS, and RA-GRS replication modes. Azure will replicate the data asynchronously in the background as required. **Migration to or from ZRS, GZRS, and RA-GZRS** **works differently** and is best to simply copy the data to a new storage account with the desired replication mode using a tool like AzCopy or requesting a live data migration via Azure Support.

**Configure blob object replication**

* **Azure Storage blob object replication provides asynchronous replication of block blobs from one storage account to another. The blobs are replicated based on the defined replication rules.** **You can leverage object replication only when blob versioning is enabled for both the source and destination storage accounts and the blob change feed is enabled for the source storage account.**
* Blob versioning captures the state of a blob when it is modified or deleted, **Azure storage creates a new version ID for a blob with each change.** The blob change feed provides all the changes with the blobs and its metadata in form of transactional logs.
* **Benefits by using object replication**
  + **For large data processing jobs, you can analyze the data in a single region, and you can distribute results to additional regions as needed. This saves processing time and compute resources to perform the same in all regions**
  + **With replication, the users can read data from the replicated regions as well. Allows for reducing latency for read requests by giving them the flexibility to choose the nearest region to read the data**
  + **Compute workloads can now process the same sets of block blobs in different regions using object replication.**
  + **You can reduce the costs by moving your replicated data to the archive tiers using Lifecycle Management policies.**
* Limitations of Blob Object Replication -
  + Object replication doesn’t work with the Archive tier
  + Blob snapshots and immutable snapshots are not supported with object replication.
  + Object replication doesn’t work with accounts with a hierarchical namespace (Azure Data Lake Storage Gen2)
  + Because block blob data is replicated asynchronously, there is no SLA on when accounts are in sync. However, you can check the replication status of a blob
  + The source account can only have a maximum of two destination accounts
  + Once you create a replication policy, the destination container is read-only, and you can no longer perform write operations against it.

**Skill 2.3 - Configuring Azure Files and Azure Blob Storage**

* **Azure Files is a fully managed file share service that offers endpoints for the Server Message Block (SMB) protocol** **also known as Common Internet File System or CIFS. This allows you to create one or more file shares in the cloud (with default max size as 5 TiB per share). You can enable large file share for a storage account and create file shares up to 100 TiB..**
* Common use cases for Azure File shares
  + Migration of existing applications that require a file share for storage
  + Shared storage of files, such as web content, log files, application configuration files, or even installation media.
  + Replace an existing fileserver

**Connecting to Azure Files outside of Azure**

* Because Azure Files provides support for SMB 3.0, it is possible to connect directly to an Azure file share from a computer running outside of Azure. In this case, remember to open the outbound TCP port 445 in your local network.
* **Ways to Mount an Azure File Share -**
  + **Directly through file explorer with the map network drive option and the full path**
  + **Connect and mount with the net use command**
    - **net use x \\<storage-path>>**
  + **Connect and mount from Linux with the mount command**
    - **mount -t cifs //<storage>**

**Create and configure Azure File Sync service**

* Azure File Sync extends Azure Files to allow on-premises file services to be extended to Azure while maintaining performance and compatibility. Key features include
  + **Multi-site access** – The ability to write files across Windows and Azure Files
  + **Cloud Tiering** – Storage only recently accessed data on local servers. The rest of the data gets tiered to Azure in a storage account.
  + **Azure Backup Integration** – Backup in the cloud
  + **Fast Disaster Recovery** – Restore file metadata immediately and recall as needed.
* **An Azure Sync group to define the topology for how your file synchronization will take place. Within a sync group, you will add server endpoints, which are file servers and paths within the file server you want the sync group to sync with each other.**
* Enable cloud tiering to only store frequently accessed files locally on the server while all your other files are stored in Azure Files.

**Monitoring synchronization health**

* Open the sync group in the Azure portal. A health indicator is displayed by each of the server endpoints; green indicates a healthy status. Click the endpoint to see stats such as the number of files remaining, size, and any resulting errors

**Configure Azure Blob Storage**

* **Azure Blob Storage is used for large-scale storage of arbitrary data objects, such as media files, log files, and so on.**
* Blob Storage Layout -
  + Each storage account can have one or more blob containers and all blobs must be stored within a container.
  + Containers are similar in concept to a hard drive on your computer in that they provide a storage space for data in your storage account.
  + Within each container you can store blobs, much as you would store files on a hard drive. Blobs can be placed at the root of the container or organized into a folder hierarchy.

**Understanding blob types**

* Blob Types -
  + **Page Blobs** – Optimized for random-access read and write operations. Page Blobs are used to store VHD files which use unmanaged disks with Azure virtual machines. The maximum Page Blob size is 8TB.
  + **Block Blobs** – Optimized for efficient uploads and downloads for videos, images, and other general-purpose file storage. The maximum Block Blob size is slightly more than 4.75 TB.
  + **Append Blobs** – Optimized for append operations. Updating or deleting existing blocks in the blob is not supported. Up to 50,000 blocks can be added to each Append Blob, and each block can be up to 4MB in size, giving a maximum Append Blob size of slightly more than 195 GB. Page Blobs are most used for log files.
* **Blobs of all three types can share a single blob container.**
* The type of blob is set at creation and cannot be changed after the fact. If you need to change a blob after the fact, then you must delete it and reupload as the appropriate blob type before it can be used.
* **Azure Storage Explorer can be used to create a Blob container.**

**Soft Delete for Azure Storage blobs**

* **The default behavior of deleting a blob is that the blob is deleted and lost forever. Soft Delete is a feature that allows you to save and recover your data when blobs or blob snapshots are deleted even in the event of an overwrite. Maximum retention period for soft delete is 365 days.**

**Configure storage tiers for Azure blobs**

* Azure Blob Storage supports three access tiers: Hot, Cool, and Archive. Each represents a trade-off of performance, availability, and cost.

**Account-level tiering**

* **Storage Account blobs can coexist between the three access tiers (Hot, cool, and archive) within the same account. If it does not have an assigned** **tier then it infers the access tier from the account access tier setting by default. In such a scenario, you will see that the access tier’s Inferred blob property is set to true**
* You can change blob access tiers without having to move data between accounts. All requests to change tier will take place immediately between Hot and Cool tiers.
* **Data in the Archive storage tier is stored offline and must be rehydrated to the Cool or Hot tier before it can be accessed. This process can take up to 15 hours.**
* Changing access tier on a blob can either occur at the account level or at the individual blob level.

**Configure blob Lifecycle Management**

* **Azure Storage has a lifecycle-management capability, and it can be used to transition data to lower-access tiers automatically based on pre-configured rules. You can also delete the data at the end of its lifecycle. These rules can be executed against the storage account once per day. Specific blobs and containers can be targeted using filter sets.**
* Lifecycle management policies can take up to 24 hours to go into effect, and then the action can take an additional 24 hours to run. Overall, it can take 48 hours for policy actions to complete once you set up lifecycle management.
* Azure Blobs allow unstructured data to be stored and accessed at a massive scale in block blobs, such as an enterprise data lake on Azure.
* **Cool access tier is used for storing data for at least 30 days.**
* General-Purpose V2 storage accounts support hot, cool, and archive access tiers while General-Purpose V1 does not.
* **Name of a storage account must be globally unique**
* Dataset CSV file and Driveset CSV file are the two files that are required to prepare disks that will contain data to be imported into Azure Storage. The first of them contains the list of data files, while the second lists the disks and the corresponding drive letters.
* **Prerequisites for implementing object replication for Azure Blob Storage -**
  + **Blob Versioning**
  + **Change Feed**
* Soft delete automatically retains deleted blobs for up to 14 days.

**Chapter Summary -**

* + **Standard storage accounts use magnetic drives and provides the lowest cost per GB. This type of account is best suited for applications that require bulk storage or where data is accessed infrequently.**
  + Premium storage accounts use solid state drives and offer consistent, low-latency performance. This type of account can only be used with Azure virtual machine disks and are best for I/O-intensive applications like databases
  + **Storage accounts must specify a replication mode. The options are locally redundant, zone-redundant, geo-redundant, read-access geo-redundant storage, geo zone-redundant, and read-access geo zone-redundant.**
  + Azure Storage can be managed through several tools directly from Microsoft – Azure Portal, PowerShell, CLI, Storage Explorer, and AzCopy.
  + **Access to storage accounts can be controlled with several techniques – Azure AD authentication, storage account name and key, SAS, SAS with access policy, and using the storage firewall and virtual network service endpoints. Access to Blob Storage can also be controlled using the public access level of the storage container.**
  + You can also use AzCopy to copy files between storage accounts or from outside publicly accessible locations to your Azure Storage account.
  + **Azure Storage has a lifecycle-management capability, and it can be used to transition data to lower-access tiers automatically based on preconfigured rules. You can also delete the data at the end of its lifecycle. These rules can be executed against the storage account once per day. Specific blobs and containers can be targeted using filter sets.**
  + Azure Storage also provides blob object replication capabilities that provide asynchronous replication of Block Blobs from one storage account to another. The blobs are replicated based on the defined replication rules.

**Chapter 3 - Deploy and manage Azure compute resources**

**Microsoft Azure offers many features and services that can be used to create inventive solutions for almost any IT problem. Some of the most common services for designing these solutions are Microsoft Azure virtual machines (VM) and VM scale sets (VMSS). Virtual machines are one of the key compute options for deploying workloads in Microsoft Azure.**

**In addition to this, you have other compute services, Azure Kubernetes Service (known as AKS) and Azure Container Instances (known as ACI) which are comparatively new in the market. With the wide adoption of containerized workloads across many IT companies, Microsoft is also heavily investing in enhancing their current product set to support container-based workloads. We also have services, such as Azure App Service and its App Service Plans to manage and host Web applications.**

**Skill 3.1 - Automate Deployment of Virtual Machines (VMs) by using Azure Resource Manager Templates**

* The ability to provision virtual machines on-demand using the Azure portal is incredibly powerful. The true power of the cloud, however, is the ability to automatically deploy one or more resources defined in code, such as a script or a template. Use cases such as defining an application configuration, and automatically deploying it on-demand allow teams to be more agile by providing dev, test, or production environments in a fast and repeatable fashion. Because the configuration is stored as code, changes to infrastructure can also be tracked in a version control system.

**ARM Template Overview**

* **Azure Resource Manager templates are authored using JavaScript Object Notation (JSON) and provide the ability to define the configuration of resources, such as virtual machines, storage accounts, and so on in a declarative manner.**
* **Structure of an ARM Template**
  + **$schema** The JSON schema file is the reference to the standard structure defined for an ARM template, which can help you determine when something is wrong with your template in comparison to the schema file syntax.
  + **contentVersion** This provides source control to track the changes made in your template.
  + **parameters** Using parameters, we can define the various values that are passed at runtime without changing the exact template file. The parameters can be changed by the *azuredeploy.parameters.json* file or in the PowerShell script that is used to deploy your template. The parameters are key elements when dealing with nested templates to pass the values from parent template to the child templates.
  + **variables** Defines values which are used in your template to simplify template language. Mostly, variables are hard-coded values, but they also can be created dynamically using parameters or standard template functions.
  + **functions** Users can create functions that can be used within the template. The complex expressions that are being used multiple times in the template can be defined as a function once. You need to create your own namespace and create member functions as needed. You cannot access variables or any other user-defined functions within your function.
  + **resources** This contains resources that are deployed or updated in a resource group. You can define the condition to control the provisioning of each resource. Also, the *dependsOn* determines which resources must be deployed first before a specific resource.
  + **outputs** Here, you can define the type of values that are returned after deployment. This section is used to keep a track of resources that are being deployed or updated.

**Defining a network interface**

* **Every virtual machine has one or more network interfaces. To create one with a template, add a variable to the variables section to store the network interface resource name as the following snippet demonstrates:**
  + **"VMNicName": "VMNic"**

**Defining a virtual machine resource**

* Before creating the virtual machine resource, you will add several parameters and variables to define. Each virtual machine requires administrative credentials. To enable a user to specify the credentials at deployment time, add two additional parameters for the administrator account and the password.

"VMAdminUserName": {

"type": "string",

"minLength": 1

},

"VMAdminPassword": {

"type": "string",

"minLength": 1

}

* **To add a public IP address to the virtual machine, you must make several modifications. The first is to define a parameter that the user will use to specify a unique DNS name for the public IP. Second is to add the public IP resource itself.** 
  + **Next modification is to update the network interface resource that the public IP address is associated with. Network interface must now have a dependency on the public IP address to ensure it is created before the network interface.**
* **Several variables are needed to define the configuration of the virtual machine resource. The following variables define the VM name, operating system image, and the VM size. These should be inserted into the variables section of the template.**

**"VMName": "MyVM",**

**"VMImagePublisher": "MicrosoftWindowsServer",**

**"VMImageOffer": "WindowsServer",**

**"VMOSVersion": "WS2019-Datacenter",**

**"VMOSDiskName": "VM2OSDisk",**

**"VMSize": "Standard\_D2\_v2",**

**"VM2ImagePublisher": "MicrosoftWindowsServer",**

**"VM2ImageOffer": "WindowsServer",**

**"VM2OSDiskName": "VM2OSDisk",**

**"VMSize": "Standard\_D2\_v2"**

* There are several properties of a virtual machine resource that are critical to its configuration.
  + hardwareProfile This element is where you set the size of the virtual machine. Set the vmSize property to the desired size, such as Standard\_D2\_v2.
  + osProfile This element at a basic level is where you set the computerName and adminUsername properties. The adminPassword property is required if you do not specify an SSH key. This element also supports three sub elements: windowsConfiguration, linuxConfiguration and secrets.
  + osProfile, windowsConfiguration While the example doesn’t use this configuration, this element provides the ability to set advanced properties on Windows VMs:
  + provisionVMAgent This is enabled by default, but you can disable it. Specify whether extensions can be added.
  + enableAutomaticUpdates Specify whether windows updates are enabled.
  + timeZone Specify the time zone for the virtual machine.
  + additionalUnattendContent Pass unattended install configuration for additional configuration options.
  + winRM Configure Windows PowerShell remoting.
  + provisionVMAgent Enabled by default, but you can disable. Specify whether extensions can be added.
  + disablePasswordAuthentication If set to true you must specify an SSH key.
  + Ssh, publicKeys Specify the public key to use for authentication with the VM.
  + osProfile, secrets This element secrets is used for deploying certificates that are in Azure Key Vault.
  + storageProfile This element is where OS image is specified, and the OS and data disk configuration are set.
  + networkProfile This element is where the network interfaces for the virtual machine are specified.
* VM Scale Sets provide the unique ability to scale out certain types of workloads to handle large processing problems, and they optimize cost by only running instances when needed.
* **Prerequisite of deploying a virtual machine is a virtual network.**

**Modify an Azure Resource Manager template**

* Often you will need to modify a template that you have previously used to change the configuration. As previously mentioned, one of the key concepts of using templates to describe your infrastructure (commonly referred to as Infrastructure as Code) is so you can modify it and deploy in a versioned manner. To accommodate this behavior ARM supports two different deployment modes: complete and incremental.
  + **Incremental** – ARM leaves unchanged resources that exist in the resource group but aren’t in the template. It will update the resources in the resource group if the settings in the template differ from what is deployed.
  + **Complete** – ARM will delete resources that exist in the resource group that are not in the template. Helpful if you need to remove a resource from Azure and you want to make sure your template matches the deployment. To use Complete mode, you must use the REST API or the command-line tools with the -Mode/--mode parameter set to Complete.
* **Incremental is the default mode for ARM deployments.**

**Configure a virtual hard disk template**

* **In the storageProfile section of a virtual machine resource, you can specify the imageReference element that references an image from the Azure Marketplace**
* You also can specify a generalized VHD that you have previously created. To specify a user image for a generalized VHD, you must specify the osType property (Windows or Linux) and the URL to the VHD itself, and the URL to where the disk will be created in Azure Storage (osDiskVhdName).
* **Templates can be deployed using the Azure Portal, the command line tools, or directly using the REST API.**
* The parameters to a template can be passed to the **New-AzResourceGroupDeployment** cmdlet using the **TemplateParameterObject** parameter for values that are defined directly in the script as **.json**. The **TemplateParameterFile** parameter can be used for values stored in a local .json file. The **TemplateParameterUri** parameter for values that are stored in a .json file at an HTTP endpoint.

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

* An existing deployment can be exported as a template that you can use to regenerate the environment or to just gain a better understanding of how the deployment is configured. There are two ways of exporting a template from a deployment within a resource group.
  + **First way** is to export the actual template used for the deployment. This method exports the template exactly as it was used, including the values for parameters and variables during the original execution. To export this template, navigate to the resource group in Azure and click deployments, select the deployment to export and click view template on the top navigation. From here you can click to download the template locally or you can click to deploy to re-deploy the template using different parameters.
  + **Second Approach** is generating an ARM template is to use the Automation Script menu option for the resource group. It generates a template that represents **the current state of the resource group.** Useful for redeploying to the same resource group as it probably has hard-coded values.

**Deploy virtual machine extensions**

* Azure VMs have a variety of built-in extensions that can enable configuration management. These are the most common. Both extensions can be executed at provisioning time or after the virtual machine has already been started.
  + **Windows PowerShell DSC Extension** allows you to define the state of a virtual machine using the PowerShell Desired State Configuration Language. Allows for continuous updates when integrated with Azure Automation DSC service.
  + **Custom Script Extension** can be used to execute arbitrary commands such as batch files, regular PowerShell scripts, or a bash script. Supported on Windows and Linux-based virtual machines and is ideal for bootstrapping a VM to an initial configuration. Your script must be accessible via a URI such as an Azure Storage Account to be used and must either be accessed anonymously or passed with a shared access signature (SAS URL).
* **Azure Virtual Machine extensions are small applications that can be used to perform post-deployment configuration and automation tasks on Azure Virtual Machines**
* Automation Scripts link in the left pane of a resource group page provides the ability to generate an Azure Resource Manager template that represents the current state of that resource group
* **Shared Access Signature (SAS) generates a URL that provides secure access to an Azure Storage blob containing the script used by the Azure custom script extension.**

**Skill 3.2 - Configure VMs for High Availability and Scalability**

* **Microsoft Azure virtual machines are a flexible and powerful option for deploying workloads into the cloud. The support of both Windows and Linux-based operating systems allows for the deployment of a wide variety of workloads that traditionally run in an on-premises environment..**

**Configure high availability**

* Resiliency is a critical part of any application architecture. Azure provides several features and capabilities to make it easier to design resilient solutions. The platform helps you to avoid a single point of failure at the physical hardware level and provides techniques to avoid downtime during host updates. Using features such as availability zones, availability sets, and load balancers provides you the capabilities to build highly resilient and available systems.

**Availability zones (Used for REPLICATION)**

* Many Azure regions provide availability zones, **which are separated groups of datacenters within a region.** Availability zones are close enough to have low-latency connections to other availability zones. They're connected by a high-performance network with a round-trip latency of less than 2ms. However, availability zones are far enough apart to reduce the likelihood that more than one will be affected by local outages or weather. Availability zones have independent power, cooling, and networking infrastructure. They're designed so that if one zone experiences an outage, then regional services, capacity, and high availability are supported by the remaining zones. They help your data stay synchronized and accessible when things go wrong. To ensure resiliency, there is a minimum of three separate zones in all enabled regions.
* **Availability Zones provide a 99.99% SLA uptime when two or more VMs are deployed into two or more availability zones.**
* **There are two ways that Azure services use availability zones:**
  + **Zonal** resources are pinned to a specific availability zone. You can combine multiple zonal deployments across different zones to meet high reliability requirements. You're responsible for managing data replication and distributing requests across zones. If an outage occurs in a single availability zone, you're responsible for failover to another availability zone.
  + **Zone-redundant** resources are spread across multiple availability zones. Microsoft manages spreading requests across zones and the replication of data across zones. If an outage occurs in a single availability zone, Microsoft manages failover automatically.
* When VMs are created in three Availability Zones, those will be automatically distributed across three **fault domains** and three **update domains**
* **Fault domains represent a group of servers which have shared power, cooling and networking.**
* An update domain represents **a group of servers** that can be rebooted at the same time.
  + **NOTE: An availability set protects your azure resources from failures within data centers where as an availability zone protects from entire datacenter failures.**

**Availability sets**

* Deploying a multitier application into an availability set can provide redundancy and high availability to the virtual machines. To provide redundancy for your virtual machines, you must place at least two virtual machines in an availability set. This configuration ensures that at least one virtual machine is available in the event of a host update, or a problem. Having at least two virtual machines in an availability set is a requirement for the service level agreement (SLA) for virtual machines of 99.95 percent. You can place a single instance virtual machine in an availability set, too, but doing so provides comparatively lower SLAs. A Premium SSD provides an SLA of 99.9 percent, while Standard SSD and Standard HDD provide SLAs of 99.5 percent and 95 percent, respectively.
* **Availability sets should be deployed according to their workload or application tier. If you have a three-tier solution of web servers, middle tier, and a database tier then each should have its own availability set. IE two data tier VMs in its own availability set, two web tier VMs in its own availability set and so on.**
* Availability sets can be configured by assigning a fault domain and an update domain. Each availability set can have up to 20 update domains and 3 fault domains. This reduces the impact to VMs from physical hardware failures.
* **An availability set must be set at the creation time of the virtual machine.**
* **Proximity placement group is a logical grouping of VMs to reduce the latency by keeping them closer to each other.** If the VMs are placed in the same proximity placement group, they will be physically located closer to each other.

**Availability sets and managed disks**

* **Availability sets and managed disks complement each other. When the VM uses managed disks and is placed in an availability set (known as an aligned availability set), it ensures that the VM disks are placed in different storage fault domains. This alignment ensures that all the managed disks attached to a VM are within the same managed disk fault domain. Number of fault domains for an availability set depends on the region it belongs to, with either two or three fault domains per region.**

**Deploy and configure scale sets**

* **VMSS is especially beneficial for applications with variable or unpredictable workloads since it can automatically alter the number of VM instances based on demand. This helps to ensure that your application stays available and responsive to users even during periods of high traffic or increased activity.**
* You can use VMSS to deploy a group of VMs with identical configurations, such as the operating system, programs, and data. VMSS also provides load balancing, automatic scaling, and connection with other Azure services, making it a powerful and versatile solution for managing large-scale workloads.
* **Overall, VMSS is an important Microsoft Azure feature that provides a scalable and dependable method for deploying and managing a set of identical VMs.**
* A **Virtual Machine Scale Set (VMSS)** is a compute resource that you can use to deploy and manage a set of identical virtual machines. By default, a VMSS supports up to 100 instances but can scale up to 1,000 instances by placing instances into multiple placement groups. Using multiple placement groups is commonly referred to as a “large scale set”
* **Placement group is a construct, such as an Azure Availability set, with its own fault domains and upgrade domains.**
* Conditions of large-scale sets “multiple placement groups”
  + **If you are using a custom image, your scale set supports up to 600 instances instead of 1,000**
  + The basic SKU of the Azure Load Balancer can scale up to 300 instances
  + For a large-scale set (>100 instances), you should use the Standard SKU (Supports up to 1,000 instances) or the Azure Application Gateway.
* **All instances of a VMSS will use the same operating system disk image**
* A scale set can be deployed to an availability zone to provide higher redundancy and resiliency. If the scale set is created within a single availability zone, then all the instances will be deployed within a single zone. If the scale set is deployed in multiple availability zones (**known as zone-redundant scale set**), based on scaling rules, the instances can be deployed to multiple zones if needed.
* **Load Balancing Settings When Setting up a VMSS**
  + **Application Gateway** is an HTTP/HTTPS web traffic load balancer with IRL-based routing, SSL termination, session persistence, and web application firewall.
  + **Azure Load Balancer** support all TCP/UDP network traffic, port-forwarding and outbound flows
* During the lifecycle of running a virtual machine scale set you may need to upgrade the instances with the latest scale set model. The VMSS resource property upgrade policy determines how VMs will be upgraded once a new update is available.
  + **Automatic** – All instances are updated in the random order when an update is available, which can cause downtime
  + **Rolling** – Updates VMs in multiple batches, and you can set a pause time between two batches, which can avoid total downtime.
  + **Manual** – It is up to you to programmatically step through and update each instance using the PowerShell command.
* **Health Monitoring for a VMSS is required when you plan to use managed infrastructure and automatic OS upgrades.**
* Spreading Algorithm option for Health Monitoring decides how scale set instances will be placed in a fault domain. With max spreading, the instances are distributed in the maximum fault domains possible for each zone. Fixed Spreading restricts instances to **exactly five fault domains. If a scale set is using a fixed spreading algorithm and if there are less than five fault domains available, the deployment will fail.**

**Skill 3.3 - Configure VMs**

* There are multiple ways to create and configure virtual machines, depending on your intended use. The easiest way to create an individual virtual machine is to use the Azure portal. If you have a need for automated provisioning (or you just enjoy the command line), the Azure PowerShell cmdlets and the Azure cross-platform command line interface (CLI) are a good fit. For more advanced automation—even including orchestration of multiple virtual machines—Azure Resource Manager templates can also be used.

**Configure Azure Disk Encryption**

* **Encrypting Azure VM disk was always a difficult task, and you had to rely on the Azure AD app to perform the work. Now you have a straightforward way to encrypt your Azure VM disks using integration with Azure Key Vault..**
* VM’s can be created with customer-managed encryption keys (CMK) while creating a new VM. Before creating the VM, you need to create disk encryption set first.
* **Once a customer-managed key is used, you cannot change the selection back to platform-managed key.**
* Server-side Encryption Models -
  + **Service-managed keys** – Provides a combination of control and convenience with low overhead
  + **Customer-Managed Keys** – Gives you control over the keys, including bring your own key (BYOK) support, or allows you to generate new ones
  + **Service-managed keys in customer-controlled hardware** – Enables you to manage keys in your proprietary repository, outside of Microsoft control. This is called host your own key (HYOK), it's a complex setup and most Azure services don’t support it.

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

* **VMs can be moved to a different resource group, subscription, availability zone, and another region.**
* Because the resource group will change, any existing scripts that target resources in this resource group will no longer work until they have been updated.
* **You can retrieve the resource ID using Azure portal, PowerShell, or CLI. To get the resource ID from Azure portal, you need to go to resource and then navigate to the resource’s properties.**

**Manage VM sizes**

* **There are many situations where the amount of compute processing your workload needs varies dramatically from day to day or even hour to hour. Azure provides purpose built virtual machine sizes.**
* Different Types of Virtual Machine Sizes -
  + **General Purpose** – Small to medium scale development environments. Has a balanced CPU-to-memory ratio.
  + **Compute Optimized** – Uses for CPU intensive workloads in medium-scale environments. Ideal for network appliances or batch process in small environments.
  + **Memory optimized** – This size type provides higher memory compared to CPU and is ideal for medium-scale database servers. With high memory, these sizes can be used for caches, or it can be used in memory analytics.
  + **Storage Optimized** – Offers high disk throughput and IO, good for large transactional databases like Cassandra, MongoDB, and so on. It can be used for Big Data and data warehousing.
  + **GPU Optimized** – Provides VMs with one or many NVIDIA GPUs. It provides high compute and graphics, which are ideal for visualization workloads.
  + **High Performance Compute** – Capable of handling batch processing, molecular modeling, and fluid dynamics. This type offers substantial CPU power and diverse options for low-latency RDMA networking using FDR InfiniBand and several memory configurations to support memory-intensive computational requirements.
* **Azure VMs are relatively easy to change the size even after it has been deployed. There are a few things to consider when having to change the size:** 
  + **The first consideration is to ensure that the region your VM is deployed to supports the instance size that you want to change the VM to.**
  + **The new size must be supported in the current hardware cluster in which your VM is deployed. The second consideration is whether the new size is supported in the current hardware cluster in which your VM is deployed.**
* All VMs in an availability set must be stopped before changing the hardware cluster. All running VMs in an availability set must use the same physical hardware cluster.

**Add data disks**

* **Adding a data disk to an existing Azure virtual machine using the Azure portal is almost identical to the creation process.**
* You can use the following source types to create a new managed disk:
  + **Snapshot** – If selected, you can browse for snapshots in the current subscription and location.
  + **Storage Blob** – If selected, you can browse storage accounts in all subscriptions you have access to, so you can select the VHD.
  + **None** – If selected, a new empty VHD is created.
* **If the virtual machine is deployed into an availability zone, the disk is automatically placed into the same zone as the virtual machine using Azure CLI.**

**Configure networking**

* During the virtual machine provisioning process in the Azure portal you can set the following options using the Networking blade
  + The virtual network, subnet, and the public IP address
  + The network security group for the network interface card (NIC)
  + The public inbound ports that should be open (if any)
  + If accelerated networking should be enabled
  + If the VM should be included in an existing Azure Load Balancer back-end pool
* Network Security Group (NSG) - Is a networking filter containing a list of security rules which control network traffic when applied. These rules can manage both inbound and outbound traffic. **A network security group can be associated to a network interface, the subnet the network interface is in, or both.** To simplify management of security rules, it’s recommended that whenever possible, you associate a network security group to individual subnets, rather than individual network interfaces within the subnet. **Learn more about networking Chapter 4**
* **Application Security Groups (ASG) – Enables you to define network security policies based on workloads with rules focused on applications instead of IP and network addresses. They allow you to group virtual machines with monikers and secure applications by filtering traffic from trusted segments of your network.**

**Accelerated networking**

* **Accelerated Networking enables single root I/O virtualization (SR-IOV) to a virtual machine, which improves it networking performance. Improves performance by bypassing the virtual switch between the host VM and the physical switch.**
* **Accelerated networking** can be enabled at the time of creation or after the virtual machine is created, if the following perquisites are met
  + VM must be a supported size for accelerated networking
  + VM must use a supported Azure Gallery Image
  + **All VMs in an availability set or VMSS must be stopped/deallocated before enabling it on any NICs**
* **Supported sizes for accelerated networking -**
  + **Supported on most general-purpose and compute-optimized instance sizes with two or more vCPUs. D/DSv2 and F/Fs series are supported**
  + **On instances that support hyperthreading, supported on VM instances with four or more vCPUs. The following series are supported – D/DSv3, E/ESv3, Fsv2, and Ms/Mms.**
* The following Windows-based images from the Azure Marketplace are supported:
  + Windows Server 2019 Datacenter
  + Windows Server 2016 Datacenter
  + Windows Server 2012 R2 Datacenter

**Add network interfaces**

* A network interface enables an Azure virtual machine to communicate with the Internet, Azure, and on-premises resources. Common use cases for having multiple network interfaces are as follows:
  + **Network and Security Functions** – Multiple network interfaces enable virtual network appliances, such as load balancers, firewalls, and proxy servers
  + **Network Isolation** - Common best practices include isolating public-facing services from internal networks.
  + **Bandwidth Isolation** - In certain cases, such as heartbeat signals, it is important to have isolated traffic to guarantee the minimal amount of bandwidth is available to the workload.
* **By default, the first network interface attached is defined as the primary network interface. All others are secondary. You can control which network interface you send outbound traffic to; by default, it will be the primary network interface.**

**Connecting to virtual machines**

* **There are many ways to connect to virtual machines. You should consider options such as connecting to VMs using their public IP addresses and protecting VMs with network security groups and allowing only the port for the service you are connecting to. You should also understand how to connect to a VM on its private IP address. In this section, we’ll review the most common tools to connect and manage your VMs.**

**Authentication options**

* For **Windows-based virtual machines**, usernames can be a maximum of 20 characters in length and cannot end in a period (“.”). Passwords must be between 12 and 123 characters in length, and they must meet several complexity requirements.
* **For Linux-based virtual machines, you can specify an existing SSH public key or a password when creating a Linux VM. Linux usernames must be between 1 and 32 characters in length, and passwords must be between 6 and 72 characters.**

**Connecting to a Windows VM with remote desktop**

* The default connectivity option for a Windows-based virtual machine is to use the remote desktop protocol (RDP) and a Remote Desktop client such as mstsc.exe. The RDP service listens on TCP port 3389 and provides full access to the Windows desktop.

**Redeploy VMs**

* **Redeploying a VM might help with troubleshooting issues, such as RDP or SSH connectivity or application access. When you redeploy a VM, it moves the VM to a new host within Azure and reenables.**
  + **To redeploy a VM using PowerShell , use the “Set-AzVM” cmdlet**
  + **To redeploy using Azure CLI tools use the “az vm redeploy” command**

**Skill 3.4 - Create and Configure Containers**

* **Containers allow you to package an application and all its dependencies into a compressed package called an image. The image can then be uploaded to an image repository. A container runtime can then be installed on your computer (or VM) and point it to the image in the repository. Container runtime will download the image, extract it, and it will then create a container that hosts the application in an isolated environment.**
* A container doesn’t need an entire operating system because it uses the kernel of the host OS. For that reason, you can’t run a Linux-based container on a Windows computer or vice versa.
* **Docker is the most popular container runtime. You can install Docker on Windows, MacOS or Linux. Docker also has a repository called Docker Hub.**
* Azure has many services that utilize containers. In this chapter, we’re going to cover two of them: Azure Container Instances and Azure Kubernetes Service.
  + Azure Container Instances – Easy to create containers. Point ACI to a repository and it creates the container for you. It doesn’t require that you pay for a VM, ACI is serverless. You can access it using a public IP address or a DNS name label using the format label.azure\_region.azurecontainer.io.
  + **Azure Kubernetes Service – A cloud-based implementation of the popular container orchestration service Kubernetes. Kubernetes runs on top of the container runtime, and it can help you scale and manage a containerized deployment. Configuring Kubernetes on your own is complex and requires specialized knowledge. Using Azure Kubernetes Service is much simpler because Microsoft has done the heavy lifting for you.**
* Azure Cloud Shell is a convenient way to create a container for ACI. Cloud Shell allows you to use PowerShell or Bash, and you can use both the Azure CLI and the Az PowerShell module to create and manage Azure resources.
* **Containers can be access from the URL using port 80 or with the Public IP address.**
* When creating a container through Azure Cli or Az Powershell, you are actually creating a container group and a container to run inside of that container group. A Container Group is the top-level object in ACI, and it represents all the containers running on a particular computer. Multiple containers being used in a container group share the same URL, so you will need to specify a separate port for each container. To create multiple containers in a container group you need to use ARM templates.
* **Multi-container container groups are currently only supported on Linux. A container group that hosts a Windows container can only contain that single container. Microsoft is working on feature parity between Linux and Windows containers, so this will likely change.**
* Largest image you can host in ACI is 15 GB. You cannot create a container that uses more than 4 CPU cores and 16GB of memory.
* **A container runs inside of a container group regardless if you have one or multiple containers.**
* **A container group can have its properties updated like its DNS label. Not all properties of a container group can be modified, some require that you delete and redeploy container groups in order to change them.**
* Properties that require container delete
  + OS Type
  + CPU, memory, or GPU resources
  + Restart Policy
  + Network Profile
  + Availability Zone

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

* **As applications grow, they may span multiple services running in multiple containers, and these containers may be running on multiple servers. This complexity may become difficult to manage with Azure Container Instances. AKS includes an orchestrator that can help you to manage the complexity of a multi-container deployment.**
* AKS deployments run in a cluster and each computer in the cluster is referred to as a ***node***. One single node is responsible for the other nodes in the cluster, it is commonly referred to as the ***control plane***.
* Inside of a cluster, you will find one or more containers. These containers run inside of a ***pod***. A pod can run a single container, but it can also run multiple containers. When multiple containers are running in a pod, they share storage and a single IP address.
* **Azure Disks can be used for storage with AKS. To create and configure an Azure disk using the Asure CLI use the code below:**
  + *az disk create --resource-group AZ-104 --name 104disk --size-gb 50 --query id --output tsv*
* The resource group you specify when creating your disk is the resource group for the AKS cluster. If you don’t know the resource group of your AKS cluster, you can use the ***az aks show***.
* **Azure Disks can only be used by a single pod. If you need to access storage across multiple pods, you should use Azure Files instead. When you use Azure Files, you’ll be able to access your storage using an SMB path that any of your pods can use. Whether you use Azure Disks or Azure Files, if you create your storage as part of your pod, that storage will not exist when the pod is deleted.**
* If you require that the lifecycle of your stored data, not be tied to the lifecycle of the pod, you should use persistent volumes instead. Persistent volumes can use Azure Files or Azure Disks, and they can either be created by the AKS cluster administrator or by the Kubernetes API.
* **If you require that the lifecycle of your stored data not be tied to the lifecycle of the pod, you should use *persistent volumes* instead. Persistent Volume exists within the cluster, but outside of a pod. Kubernetes connects the persistent volume to the pod using a persistent volume claim.**

**Configure scaling for AKS**

* **AKS provides the ability to manually scale or to automatically scale, and you can scale at the cluster level when you need to scale nodes and at the pod level when you need to scale pods. You can even incorporate ACI into your AKS cluster in order to handle situations where you need to add additional nodes quickly.**

**Autoscaler components of AKS -**

* Kubernetes provides two autoscaler components to make it easy to configure auto-scaling; the horizontal pod autoscaler (HPA) and the cluster autoscaler.-
  + **Horizontal Pod Autoscaler (HPA)–** Scales to a maximum of five pods and minimum of two pods.
  + **Cluster Autoscaler** – Scales your node clusters based on the number of pending pods.
* **To use the HPA to scale your pods, use *kubectl* to run the autoscale command on your deployment.**
* Under the hood, AKS uses Virtual Machine Scale Sets (VMSS) to implement cluster autoscaling. However, **you should never attempt to interact with the VMSS directly.**

**Configure network connections for AKS**

* When creating an AKS cluster, you have two options for networking: kubenet and Azure Container Networking Interface (CNI). By default, AKS will use ***kubenet*** networking, also called basic networking. However, you can specify to use CNI (or advanced) networking if desired.
  + ***Kubenet*** – Called basic networking. Each node in the cluster gets an IP address from the VNet subnet where the cluster is deployed. Each pod within the cluster gets an internal IP address from an address space explicitly set aside for the pods. Uses NAT to establish connections to other Azure resources. Means that other VMs in Azure or On-premises can't directly establish communication with those pods.
  + **Azure Container Networking Interface (CNI)** - Called advanced networking. When you use CNI networking, both the nodes and the pods receive an IP address from the subnet
* **Kubernetes is designed to orchestrate pods, and that means that it spins them up and tears them down as needed. Because of that, the IP addresses for your pods are constantly changing. For that reason, Kubernetes implements the concept of a service that sits between incoming network traffic and one or more identical pods. The service gets an IP address from a specific IP address pool set aside for services, and because the service is always running, it’s not affected by pod lifecycle.**
* When network traffic needs to reach a particular pod (for example, a pod running a website that needs to process an HTTP request), the traffic is received by the service. The service will then balance the traffic to the pods using a round robin algorithm. **There are multiple service types**.
  + **Cluster IP** – Provides an internal IP address that can only be used within the AKS cluster
  + **NodePort** – Provides a port mapping on the node, allowing network traffic to reach the node using the specified port.
  + **LoadBalancer** – Provides an Azure Load Balancer and an external IP address to allow access to the node as per load balancing rules that are created. (Internal load balancers can be created to restrict access from the internet)
  + **ExternalName** – Provides a DNS entry for AKS nodes.
* **You can use NGINX for an ingress controller in AKS, but you can also use other methods such as the AKS HTTP application routing feature or the Application Gateway Ingress Controller (AGIC) add-on.**
* Network Security in an AKS cluster is handled using NSGs and network policy. Azure creates NSG rules for you as you create resources. Network policy is a feature in Kubernetes that enables you to control network traffic between pods.
* **Command to check for updates in AKS-** 
  + ***az aks get-updates***
* When upgrading an AKS cluster, you cannot skip minor versions. For example, you can’t upgrade from version 1.19.3 to version 1.21.1. You would first have to upgrade to a 1.20 build and then upgrade again to version 1.21.1.
* **Upgrading an AKS cluster upgrades one node at a time. First stops any pods from being scheduled on the node it’s about to upgrade, and any pods that are currently running on that node are scheduled for other nodes. A new node is then created using the version of Kubernetes you’ve specified to upgrade to, and only when that node is ready and added to the cluster will new pods be scheduled to upgrade. Once that’s done Kubernetes deletes the nodes running the older versions and begins the upgrade process on the next node in the cluster. This continues until all nodes are upgraded.**

**Skill 3.5 - Create and Configure Azure App Service**

* Azure App Service is a **PaaS** offering that makes it easy to host a web app in the cloud. However, App Service isn’t just for web apps. Any application that is designed to process HTTP requests can benefit from App Service. This makes App Service the ideal hosting platform for web apps, apps that expose REST APIs, and much more.
* **App Service consists of a front-end load balancer that uses a round robin algorithm to distribute requests to web servers. These web servers are called *workers*, and they are responsible for processing HTTP requests. You can use either Linux or Windows workers, and you can also choose the VM size used by your app.**
* App Services offers both shared workers (Shared with other App Services users) or dedicated works that host only your app. These configurations choices are part of an ***App Service Plan*** that is used to host your apps.

**Create an App Service Plan**

* **Before creating a web app in App Service, you need to create an App Service Plan. You can create it yourself or have Azure create it for you when you create your web app.**

**Configure scaling settings in an App Service plan**

* One of the greatest benefits of App Service is the flexibility it provides for scaling applications. You can scale vertically (scale to a more or less powerful VM) or horizontally (add and remove *workers*) easily and quickly.
* **When scaling in App Service, it will take steps to ensure your application remains available during the scaling process. It will first allocate one or more VMs of the selected size and it will then copy any apps you have in the App Service plan to the new VMs.**
* When an App Service plan runs on more than one instance, the front-end load balancer will use a round robin algorithm to load balance between all instances.
* **When you configure an autoscale rule to scale out for a specific metric, you should also create another rule to scale in when that metric drops below your desired threshold.**
* You can create your app using your own code or to run a Docker container in your App Service plan. If you choose Code, you can select from a wide range of runtime stacks, including .NET, Java, Node, PHP, Ruby, and Python.
* If you don’t see an existing App Service plan when creating a new web app, make sure you have selected the OS that matches the App Service plan’s OS. You also need to ensure the region you select is the region where the App Service plan is deployed.
* **You can use App Service to secure your apps with Azure Active Directory, and you can also easily implement security using Facebook, Google, and Twitter so that users can authenticate to your app using their existing logins on-premises.**
* App Service uses OAUTH authentication when configuring a third-party identity provider. Secrets that you provide to configure the provided are securely stored in Azure Key Vault.
* **You can browse to your app service app using the provided Azure domain name or using your own domain name.**
* DNS Records needed at your domain registrar –
  + A Record – Root domain record that maps a host to the IP address
  + TXT Record – Verify ownership of your root domain, create a TXT record named asuid with a value of the custom domain verification ID shown in the Azure portal.
  + CNAME Record – Used to map a subdomain.
* **App Services can be used to buy and configure a domain name. App Service domains are fully managed in Azure, and they are the easiest way to configure a custom domain.**

**Configure backup for an App Service**

* App Service provides easy backup and restore of your apps. These backups can be created manually, or they can be scheduled on a regular basis. Backups can be retained for an indefinite amount of time.
* **App Service backups are stored in Azure storage, and each backup is a complete copy of the app. Backups are not incremental.**
* App Service backups cannot exceed 10 GB, and that includes the app’s content and any databases.

**Configure networking settings**

* App Services Networking Features
  + **VNet Integration** – Enables outgoing communication from your app into your Azure virtual network
  + **Private Endpoint Connections** – Enables connectivity to your app from private endpoints using **Azure Private Link**. Enables you to connect securely to resources running in Azure VNet or on-premises resources using either VPN or ExpressRoute.
  + **Hybrid Connection** – Enables outgoing communication from your app to an endpoint using a TCP connection. The host can be located practically anywhere. Rely on the installation of the Hybrid Connection Manager (HCM) on the host you’re attempting to access. HCM handles communication between the remote host and the web app and because this communication happens over standard ports it doesn’t usually require any ports to be opened on a firewall. Typically connects to the endpoint host with the NetBIOS name. Uses Service Bus for communication.
* **Deployment slots allow you to create another app with its own hostname in your App Service Plan. These can be used to test a new version of an app, and once you are satisfied with the new version, you can easily swap the test deployment slot into production. You can even configure a percentage of live traffic to a deployment slot for testing.**

**Configure deployment settings**

* **Deployment slots allow you to create another app with its own hostname in your App Service plan. You can use a deployment slot to test a new version of an app, and once you’re satisfied with the new version, you can easily swap the test deployment slot into production. You can even configure App Service to send a percentage of live traffic to a deployment slot for testing.**
* Each deployment slot is its own web page.
* **One of the most common deployment options is to use continuous integration and continuous deployment (CI/CD). With this you can use a source repository like GitHub, or a local Git repository.**
* App Service also supports deployments using FTPS.
* **You can deploy to your web app using tools like Visual Studio, visual studio code, and others y using the publish profile. The publish profile is a configuration file that lists all the methods available for deployment as well as the credentials necessary for connective.**
* Azure DevOps should be used to implement continuous deployment strategy
* **Azure App Service plans map directly to a pricing tier.**

**Chapter Summary –**

* + You can create VMs from the Azure portal, PowerShell, the CLI, and the Azure Resource Manager templates.
  + Availability sets can only be set at provisioning time, but data disks can be added at any time.
  + You can connect to Azure VMs using a public IP address or a Private IP address with RDP, SSH, or even PowerShell. A VPN must be setup to connect using a private IP like a site-to-site, point-to-site, or ExpressRoute
  + Custom Script Extension is commonly used to execute scripts on Windows or Linux-based VMs. The PowerShell DSC extension is used to apply desired state configurations to Windows-based VMs.
  + There are unmanaged and managed disks and images. Key difference between the two is with unmanaged disks or images it is up to you to manage the storage account. With managed disks, Azure takes care of this for you
  + Azure Diagnostics agent can be enabled on Windows and Linux virtual machines to capture diagnostic, performance, logs, and boot diagnostic data.
  + Availability Zones provide high availability at the data center level, availability sets provide high availability within a data center.
  + Managed disks provide additional availability over unmanaged disks by aligning with availability sets and providing storage in redundant storage units
  + Virtual Machine Scale Sets (VMSS) can scale up to 1,000 instances. You need to ensure that you create the VMSS configured for large scale sets if you intend to go above 100 instances. There are several other limits to consider too. Using custom image, you can only create up to 300 instances. To scale above 100 instances, you must use the Standard SKU of the Azure Load Balancer or the Azure Appe Gateway.
  + Azure Resource Manager templates are authored using JSON and allow you to define the configuration of resources, such as virtual machines, storage accounts, and so on in a declarative manner.
  + Kubernetes is an open-source container management and orchestration system.
  + Azure Kubernetes Services is a PaaS offering of Kubernetes running in Azure. It reduces the configuration and operational overhead of the cluster
  + Azure Container Instances are a way to quickly deploy isolated containers without worrying about backend infrastructure.
  + App Services is a PaaS-hosting service that makes it easy to build applications that can process HTTP requests.
  + An App Service plan offers computer resources to the web application for its execution. This App Service plan can be shared with multiple web apps as well.
* New-AzResourceGroup deployment cmdlet performs a deployment of an ARM template to a resource group
* Schema element of an Azure Resource Manager templated determines whether the template supports deployment to subscriptions
* Azure VMs deployed into different availability zones qualify for 99.99% SLA
* VMs deployed into the same availability set qualify for 99.95% SLA
* 16GB is the largest amount of money that can be allocated to an ACI
* Ingress controller provides the ability to distribute traffic to a set of pods based on the incoming URL
* Azure Files provide shared storage for multiple pods on an AKS cluster
* ACI provides the fastest node scaling of AKS
* Workers are responsible for processing of incoming HTTP requests
* Azure Key Vault is used to implement Azure App Service authentication to third party identity providers are stored here.
* App Service backup requires a minimum of the Standard pricing tier
* Hybrid Connection provides communication from Azure App Service to an on-premises databases server via TCP

**Chapter 4 – Configure and manage virtual networking**

**An Azure virtual network (or VNet) provides the foundation of the Azure networking infrastructure. Virtual machines are connected to virtual networks. This connection provides inbound and outbound connectivity to other virtual machines, to on-premises networks, and to the Internet. Azure provides many networking features that will be familiar to those already experienced in networking, such as the abilities to control which network flows are permitted and to control network routing.**

**Skill 4.1 – Implement and Manage Virtual Networking**

* Virtual Networks (Vnets) provide the foundations of the Azure networking infrastructure. Each Vnet allows you to define a network space, comprising one or more IP address ranges. This network space is then carved into subnets. IP addresses for virtual machines, as well as some other services such as an internal Azure Load Balancer, are assigned from these subnets. Each subnet allows you to define which network flows are permitted (Using NSGs) and what network routes should be taken (Using user-defined routes).
* **With networking you always lose five addresses:**
  + **.0 = network address**
  + **.1 = Azure Gateway**
  + **.2 & .3 = DNS**
  + **.255 = Broadcast**
    - **/29 CIDR is the smallest you can use. It gives you 8 addresses and five are already in use so you only have three usable addresses.**

**Create and configure a virtual networks and subnets**

* **A virtual network (VNet) is an Azure resource.** When creating a VNet, the most important setting is the IP range (or ranges) the VNet will use.
* **IP ranges are defined using classless inter-domain routing (CIDR) notation. For example, the range 10.5.0.0/16 represents all IP ranges starting with 10.5.**
* Your VNet IP ranges must be taken from the private address ranges defined in RFC 1918:
  + 10.0.0.0–10.255.255.255 (10.0.0.0/8)
  + 172.16.0.0–172.31.255.255 (172.16.0.0/12)
  + 192.168.0.0– 192.168.255.255 (192.168.0.0/16)
* **You can also use public, Internet-addressable IP ranges in your VNet. However, this is not recommended because the addresses within your VNet will take priority, and virtual machines in your VNet will no longer be able to access the corresponding Internet addresses.**

**Subnets**

* **Any Azure resource in a virtual network is deployed into a subnet. Subnets are used to divide the VNet IP space. Different subnets can have different network security and routing rules, enabling applications and application tiers to be isolated and network flows between them to be controlled. For example, consider a typical three-tier application architecture comprised of a web tier, an application tier, and a database tier. By implementing each tier as a separate subnet, you can control precisely which network flows are permitted between tiers and from the Internet.**
* Name of a subnet must be unique within that VNet. You cannot change the subnet name after it has been created. Each subnet must also define a single network range (CIDR format)
* **You are required to define one subnet when creating a VNet using the Azure portal.**
* You can’t change the address range of a resources already deployed to the subnet. If you want to change a subnet’s address range, you first must delete all the objects in that subnet.
* **Subnets can only be deleted from VNets if they are empty. Once a subnet is deleted, the addresses that were part of that address range are released and available again for use within new subnets that you can create.**

**Additional virtual network settings**

* There are some additional settings and features of VNets and subnets to also be aware of.
  + Properties of a virtual network
    - **Name** – VNet name must be unique within the resource group, between 2 and 64 characters, and may contain letters (case insensitive), numbers, underscores, periods or hyphens. Must start with a letter or number and end with a letter, number, or underscore.
    - **Location** – Each VNet is tied to a single Azure region and can only be used by resources in the same region
    - **Address Space** – An array of IP address ranges is available for use by subnets.
    - **DNS Settings** – Contains an array of DNS servers. If specified, these DNS servers are configured on virtual machines in the virtual network in place of the Azure-provided DNS servers
    - **Subnets** – The list of subnets configured for this VNet
    - **Peerings** – The list of peerings configured for this VNet. Peerings are used to create network connectivity between separate VNets.

**Additional virtual network settings**

* Properties of a virtual Network (VNet)
  + **Name** - The VNet name must be unique within the resource group, is between 2 and 64 characters, and may contain letters (case insensitive), numbers, underscores, periods, or hyphens. It must start with a letter or number and end with a letter, number, or underscore.
  + **Location** - Each VNet is tied to a single Azure region and can only be used by resources (such as Virtual Machines) in the same region.
  + **Address Space** - An array of IP address ranges available for use by subnets.
  + **DNS Settings** - Contains an array of DNS servers. If specified, these DNS servers are configured on virtual machines in the virtual network in place of the Azure-provided DNS servers.
  + **Subnets** - The list of subnets configured for this VNet.
  + **Peerings** - The list of peerings configured for this VNet. Peerings are used to create network connectivity between separate VNets.
* **Supported Settings of a Virutal network subnet**
  + **Name** – Must be unique within the VNet.
  + **Address Range** – IP address range for a subnet, specified in CIDR notation. All subnets must sit within the VNet address space and cannot overlap
  + **Network Security Group** – NSGs can be associated to a subnet and are used to control which inbound and outbound traffic flows are permitted.
  + **Route Table** – Applied to a subnet and used to override the default system routes. These are used to send traffic to destination networks that are different than the routes that Azure uses by default.
  + **Service Endpoints (and Policies)** - An Array of Service Endpoints for this subnet. Service Endpoints provide a direct route to various Azure PaaS services, without requiring an Internet-facing endpoint. Service Endpoint Policies provide further control over which instances of those services may be accessed.
  + **Delegations** – An array of references to delegations on the subnet. Delegations allow subnets to be used by certain Azure services, which will then deploy managed resources into the subnet.

**Create and configure VNet peering**

* VNet peering allows virtual machines in two separate virtual networks to communicate directly by using their private IP addresses. They can either be in the same Azure region or in separate Azure regions. Peering between **VNets in different regions is called** **Global VNet peering**. **In all cases, Traffic between peered VNets travels over the Microsoft backbone infrastructure, not the public Internet.**
* **You can peer VNets in different subscriptions, even if those subscriptions are under different Azure Active Directory tenants. ou can also use VNet peering to connect Resource Manager VNets to the older “classic” VNets. However, peering between two classic VNets is not supported. (A VNet-to-VNet VPN can be used in this case.)**
* Peered VNets must have non-overlapping IP address spaces. In addition, the VNet address space cannot be modified once the VNet is peered with another VNet.
* **There is a limit of 500 peering connections per VNet. This is a hard limit.**
* No VNet gateways are required by VNet peering. This avoids the cost, throughput limitations, additional latency, and additional incurred complexity associated with using VNet gateways, though you can use VNet gateways to connect to on-premises networks using gateway transit.
* **Global Peering cannot be used to access the front-end IP of a basic internal Azure load-balancer in the remote virtual network. In these cases, a VNet-to-VNet VPN should be used instead. This limitation doesn’t apply with the standard tier load balancer.**
* There is no automatic outbound connectivity between peered VNets, and the *VirtualNetwork* service tag does not include the address space of the peered VNet.
* **You can control the connectivity between peered virtual networks using network security groups.**

**Service chaining and hub-and-spoke networks**

* **A common way to reduce duplication of resources is to use a hub-and-spoke network topology. In this approach, shared resources (such as domain controllers, DNS servers, monitoring systems, and so on) are deployed into a dedicated hub VNet. These services are accessed from multiple applications, each deployed to their own separate spoke VNets.**
* VNet peering is not transitive. That means there is no automatic connectivity between spokes in a hub-and-spoke topology. This means there is no automatic connectivity between spokes in a hub-and-spoke topology. Where such connectivity is required, one approach is to deploy additional VNet peerings between spokes. However, with a large number of spokes, this can quickly become unwieldy. **An alternative approach is to deploy a network virtual appliance (NVA)** into the hub through user-defined routes (UDRs) to route inter-spoke traffic through the NVA. This is known as **service chaining**, and it enables spoke-to-spoke communication without requiring additional VNet peerings
* **To transit traffic from one spoke VNet to another spoke VNet via an Network Virtual Appliance (NVA) in the hub VNet, the VNet peerings must be configured correctly. By default, a peering connection will only accept traffic originating from the VNet to which it is connected. This will not be the case for traffic forwarded between spoke VNets via an NVA in a hub VNet. To permit such traffic, the Allow Forwarded Traffic setting must be enabled for those VNet peerings.**
* When setting up two VNets to communicate with an external network, only the first will be permitted by default. To allow connectivity between a second VNet with an external network this must be configured
  + **Use Remote Gateways** – This setting must be enabled on the peering connection from VNET-B to VNET-A. This informs VNET-B of the availability of the gateway in VNET-A. **Note that to enable this setting, VNET-B cannot have its own virtual network gateway.**
  + **Allows Gateway transit** – This option must be enabled on the peering connection from VNET-A to VNET-B. This permits traffic from VNET-B to use VNET-A's gateway to send traffic to the external network. Gateway transit can be used for S2S, P2S, and VNet to Vnet.

**Configure private and public IP addresses and network interfaces**

* **VMs in Azure use TCP/IP to communicate with: services in Azure, other VMs you have deployed in Azure, on-premises networks, and the Internet. Just as a physical server uses a network interface card (NIC) to connect to a physical network, virtual machines use a network interface resource (also referred to as a NIC) to connect to a virtual network or the Internet.**
* There are two types of IP addresses you can use in Azure:
  + **Public IP addresses**. Used for communication with the Internet
  + **Private IP addresses**. Used for communication within Azure virtual networks and connected on-premises networks

**Network interfaces**

* Both public and private IP addresses are configured on virtual machines using network interface resources. Therefore, to understand how to use public and private IP addresses with your virtual machine, you first must understand network interfaces.
* **A Network Interface is a standalone Azure resource. Typically provisioned and deleted with its corresponding virtual machine. Just as a physical server can have more than one network card, you can associate multiple network interfaces with a single virtual machine.**
* Network Interface Settings -
  + **Name** - This is the network interface name, which must be unique within the resource group. It can be between 1-80 characters, and it may contain letters (case insensitive), numbers, underscores, periods, or hyphens. The name must start with a letter or number and end with a letter, number, or underscore.
  + **Location** – Location of the resource, and it must be the same as the location of any virtual network or any virtual machine to which the network interface will be connected.
  + **DNS Settings** – If specified, These DNS servers are configured on virtual machines in the virtual network in place of the Azure-provided DNS servers. This setting will override the VNet-level DNS settings, if both are specified.
  + **IP Forwarding** – Used to enable IP forwarding on this network interface. It allows the VM using the interface to receive traffic that is not sent to one of the IPs in the IP configurations. Also, it allows the VM to send traffic using an IP address that is not in the IP configurations.
  + **IP Configurations** – This is list of IP configurations for the network interface. These are the most important settings, it contains the public and private IP addresses.
  + **Network Security Groups** – This setting will display the name of any NSGs associated with this interface.
  + **Accelerated Networking** – This setting is used to enable accelerated networking, though it is **only supported on certain VM sizes**.

**Private IP addresses**

* **Private IP addresses are configured within the IP configurations of the network interface. They are not a separate resource.**
* First private IP to be allocated will be .4
  + Addresss 0-3 are used for other purposes.
    - **0 = network address**
    - **.1 = Azure Gateway**
    - **.2 & .3 = DNS**
    - **.255 = Broadcast**
    - **/29 CIDR is the smallest you can use. It gives you 8 addresses and five are already in use so you only have three usable addresses.**
* **There are two methods used to assign private IP addresses: dynamic or static. The default allocation method is dynamic, where the IP address is automatically allocated from the resource’s subnet (using an Azure DHCP server).**
* A Dynamically allocated IP address can change when you stop and start the associated virtual machine. To avoid this, private IP addresses can also be allocated statically.
* **Static private IP addresses are commonly used for:** 
  + **Virtual machines that act as domain controllers or DNS servers**
  + **Resources that require firewall rules using IP addresses**
  + **Resources accessed by other apps/resources through an IP address explicitly, rather than a domain name.**
* By default, when you change to static, Azure will assign the previously assigned dynamic IP address.
* **Static private IP addresses should only be configured in the Azure network interface resource. They will be assigned to the virtual machine using DHCP, just like with dynamic private IP addresses.**
* Private IPv4 address assignments can be either dynamic or static**. Private IPv6 address can only be assigned dynamically.**
* **Each network interface must have one private IPv4 address assigned as the primary IP configuration. You can add one or more IPv4 address as secondary IP configurations. Each network interface can be assigned zero or a maximum or one private IPv6 address as a secondary IP configuration.**

**Create and configure public IP addresses**

* **Associating a public IP address with a network interface creates an Internet-facing endpoint, allowing your virtual machine to receive network traffic directly from the Internet.**
* **A Public IP address is a standalone Azure resource**. This contrasts with a private IP address that exists only as a collection of settings on another resource, such as a network interface or a Load Balancer.
* **Associating a VM with a public IP requires you to update the IP configuration of the network interface to reference to the public IP address resource**.

**Public IP Basic vs Standard Pricing Tiers**

* **Public IP addresses are available at two pricing tiers (or SKUs): Basic or Standard. All Public IP Addresses created before the introduction of these tiers are mapped to the Basic tier.**
* Public IP addresses tiers -
  + **Basic** – Supports both static and dynamic allocation methods. Open by default for inbound traffic. Use NSGs to restrict inbound or outbound traffic. Not zone redundant and doesn’t support availability zone. Does not support public IP prefixes.
  + **Standard Tier** – Supports static allocation only. Closed by default for inbound traffic. Use NSG’s to allow inbound traffic and restrict outbound traffic. Zone redundant by default allows you to use availability zones. Supports public IP prefixes.

**Public IP address allocation**

* **As with private IP addresses, public IP addresses support both dynamic and static IP allocation. For the Basic tier, both static and dynamic allocation are supported, the default being dynamic. For the Standard tier, only static allocation is supported.**
* Under dynamic allocation, an actual IP address is only allocated to the public IP address resource when the resource is in use. If a virtual machine is stopped (deallocated) or deleted, the IP address assigned to the public IP address resource is released and returned to the pool of available IP addresses managed by Azure.
* **If you want to retain the IP address, the public IP address resource should be configured to use static IP allocation. An IP address will be assigned immediately (if one was not already dynamically assigned). This IP address will never change, regardless of whether the associated virtual machine is stopped or deleted. Typically, static public IP addresses are used in scenarios where a dependency is defined by a particular IP address.**
* With private IP addresses, static allocation allows you to specify the IP address to use from the available subnet address range. In contrast, static allocation of public IP addresses does not allow you to specify which public IP address to use. Azure assigns the IP address from a pool of IP addresses in the Azure region where the resource is located.

**Public IP address prefixes**

* **When using multiple public IP addresses, it can be convenient to have all of the IP addresses allocated from a single IP range or prefix. For example, when configuring firewall rules, this allows you to configure a single rule for the prefix, rather than separate rules for each IP address.**
* To support this scenario, Azure allows you to reserve a public IP address prefix. Public IP address resources associated with that prefix will have their IP addresses assigned from that range, rather than from the general-purpose Azure pool.
* **When creating an IP prefix, you must specify the prefix resource name, subnet size, and the Azure region where the IP addresses will be allocated. Once the prefix is created, individual public IP addresses can be created that are associated with this prefix. Note that only standard-tier public IP addresses support allocation from a prefix, and thus only static allocation is supported.**

**DNS Labels**

* Domain Name System (DNS) can be used to create a mapping from a domain name to an IP address. This allows you to reference IP address endpoints using a domain name, rather than using the assigned IP address directly.
* **There are 4 ways to configure a DNS label for an Azure public IP address**
  + **By specifying the DNS name label property of the public IP address resource**
  + **By creating a DNS, A record in Azure DNS or a third-part DNS service hosting a DNS domain**
  + **By creating a DNS CNAME record in Azure DNS or a third-party service hosting a DNS domain**
  + **By creating an alias record in Azure DNS**
* A DNS label is a suffix of the FQDN. DNS label you provide is concatenated with this suffix to form the FQDN, which can be used to look up the IP address via a DNS query.
* **For example, if your public IP address is deployed to the Central US region, and you specify the DNS label contoso-app, then the FQDN will be contoso-app.centralus.cloudapp.azure.com.**
  + **The major limitation of this approach is that the DNS suffix is taken from an Azure-provided DNS domain. It does not support the use of your own vanity domain, such as contoso.com. To address this, you will need to use one of the other approaches.**
  + **Approach 1 - Creating a dns a record**
    - In this approach, you will have already hosted your vanity domain either in Azure DNS or a third-party DNS service. Using your hosting service, you can create a DNS entry in your vanity domain mapping to your public IP address resource.
  + **Approach 2 - Creating a dns cname record**
    - In this approach, you start by creating a DNS label for your public IP address. You then create a CNAME record in your vanity domain, which maps your chosen domain name to the Azure-provided DNS name. For example, you might map www.contoso.com to contoso-app.centralus.cloudapp.azure.com. This approach has the advantage of avoiding the need for static IP allocation because the Azure-provided DNS entry updates automatically if the assigned IP address changes. However, the downside of this approach is that the Domain Name System does not support CNAME records at the apex (or root) of a DNS domain, which means while you can create a CNAME record for www.contoso.com, you cannot create one for contoso.com (without the www).
  + **Approach 3 - Creating an alias record**
    - In this approach, your vanity domain must be hosted in Azure DNS. You can then create an alias record, which works the same as an A record, except that rather than specifying the assigned IP address value explicitly in the DNS record, you simply reference the public IP address resource. The assigned IP address is taken from this resource and automatically configured in your DNS alias record. With alias records, the DNS record is automatically updated if the assigned IP address changes, avoiding the need for static IP allocation.

**Outbound Internet connections**

* **When a public IP address is assigned to a virtual machine’s network interface, outbound traffic to the Internet will be routed through that IP address. The recipient will see your public IP address as the source IP address for the connection. However, the virtual machine itself does not see the public IP address in its network settings—it only sees the private IP address. Traffic leaves the virtual machine via the private IP address, and Source Network Address Translation (SNAT) is used to map the outbound traffic from the private IP address to the public IP address.**
* Source Network Adress Translation (SNAT) - Used for when the traffic leaves a virtual machine via the private IP address and used SNAT to map the outbound traffic from the private IP address to the public IP address.
* **A public IP address is not required for outbound Internet traffic. Even without a public IP address assigned, virtual machines can still make outbound Internet connection. SNAT is used to map the private IP address to the Internet-facing IP address.**
* Public IP address resources can use either IPv4 or IPv6 (but not both). IPv6 support is as follows
  + **Only the basic tier is supported**
  + **Only dynamic allocation is supported**
* **Azure VMs that are in the same VNet can communicate automatically with each and with the Internet without any explicit configuration changes, even when they are in different subnets.**

**IPv4 and IPv6**

* **Public IP address resources can use either an IPv4 or IPv6 address (but not both). Note that IPv6 support is limited as follows:**
  + **Only the Basic tier is supported.**
  + **Only dynamic allocation is supported.**

**Configure network routes**

* **Network routes control how traffic is routed in your network. Azure provides default routing for common scenarios, with the ability to configure your own network routes where necessary.**

**System routes**

* Azure VMs that are in the same VNet can communicate automatically with each other and with the Internet without any explicit configuration changes, even when they are in different subnets. This is also the case for communication from the VMs to your on-premises network when a hybrid connection from Azure to your data center has been established.
* **The following are the default system routes that Azure will use and provide for you:**
  + **Within the same subnet**
  + **From one subnet to another within VNet**
  + **VMs to the Internet**
  + **A VNet to another VNet through a VPN gateway**
  + **A VNet to another VNet through VNet peering**
  + **A VNet to your on-premises network through a VPN gateway or ExpressRoute (Optional)**
  + **VirtualNetworkServiceEndpoint (Optional)**

**User-defined routes**

* **User-defined routes – Useful for when you want to send traffic through a network virtual appliance, such as load balancers, firewall, or router deployed into your VNet from the Azure marketplace.**
* To make this possible, you must create what are known as user-defined routes (UDRs). The UDR is implemented by creating a route table resource. Within the route table, a number of routes are configured. Each route specifies the destination IP range (in CIDR notation) and the next hop IP address. A variety of different types of next hop are supported:
  + **Virtual Appliance** – A VM running a network application such as a load-balancer or firewall. You would have to specify the IP address of the appliance which can be a VM or internal load-balancer for high availability virtual appliances.
  + **Virtual Network Gateway** – Used to route traffic to a VPN gateway.
  + **Virtual Network** – Used to route traffic within the Virtual Network
  + **Internet** – Used to route a specific IP address of prefix to the Internet
  + **None** – Used to drop all traffic send to a given IP address or prefix
* **You can have multiple route tables, and the same route table can be associated to one or more subnets. Each subnet can only be associated to a single route table. All VMs in a subnet use the route table associated to that subnet.**

**IP forwarding**

* **User-defined routes (UDR) allow for changing the default system routes that Azure creates for you in an Azure VNet. In the virtual appliance scenario, the UDRs forward traffic to a virtual appliance such as a firewall, which is running as an Azure virtual machine.**
* By default, a virtual machine in Azure will not accept a network packet addressed to a different IP address. For that traffic to be allowed to pass into that virtual appliance, you must enable IP forwarding on the network interface of the virtual machine. This configuration doesn’t typically involve any changes to the Azure UDR or VNet, but depending on the scenario, you might need to make some configuration changes in the VM’s OS to enable this to work correctly.

**How routes are applied**

* If multiple routes contain the same address prefix, Azure selects the route type, based on the following priorities:
  + User-defined routes
  + System routes for traffic in a virtual network, across a virtual network peering, or to a virtual network service endpoint
  + BGP routes
  + Other system routes
* **If an IP address matches two routing rules, then the longest prefix match algorithm is used to select the route.**

**Forced tunneling**

* **A special case is when routes are configured with the destination IP prefix 0.0.0.0/0. Given the precedence rules described above, this route controls traffic destined for any IP address is not covered by any other rules.**

**Configure endpoints on subnets**

* Service Endpoints are a mechanism to integrate Azure PaaS services into your virtual network and access them through a Microsoft Azure backbone network instead of over the Internet. Service Endpoints prevent the exposure of data and services to the Internet.
* **Service Endpoints can be enabled on subnets, and you can also add service endpoints to multiple subnets from the virtual network settings.**
* Multiple service endpoints can be created for Azure services on a given subnet.

**Configure private endpoints**

* Private Endpoints establish a private connection between any of the supported Azure services and your virtual network. Provides a secure channel between a VNet and a service. The private endpoint gets a private IP address from your VNet address space. The connectivity for it takes place using private links.

**Configure Azure DNS**

* **It is important to understand the different roles played by recursive and authoritative DNS servers, and how a DNS query is routed to the correct DNS name servers using DNS delegation.**
* DNS Zone is the representation of a domain name in an authoritative DNS server. It contains the collection of DNS records for a given domain name.
* **You can purchase domain names using the App Service Domains service. DNS zone hosting is provided by Azure DNS.**
* DNS Settings on a user's device points to a recursive DNS server, also sometimes known as local DNS service (or LDNS) or simply a DNS resolver. Recursive DNS service is typically hosted by your company or by your ISP.
* **NS records tell clients on the Internet where to find the name servers for a given DNS zone. NS records for a DNS zone are configured in the parent zone, and a copy of the records is also present in the child zone. Setting up these NS records is called delegating a DNS domain.**

**DNS Services in Azure –**

* **There are several DNS-related services and features in Azure—an overview of each is given below. The first three items are Azure services, which you consume by creating service-specific resources that you will be billed for. The remaining three items are Azure features, which you configure using settings on other resource types, such as a virtual network, public IP address, or network interface.**
  + **Azure DNS** – Allows you to host your DNS domains in Azure. It provides the ability to create and manage the DNS records for your domain and provides name servers, which answer DNS queries for your domain from other users on the Internet. Azure DNS also supports private DNS zones, which are used for intranet-based name resolution for VM to VM lookups, including support for some scenarios not supported by the Azure-provided DNS service, which we’ll cover shortly. Private DNS zones are currently in preview.
  + **Azure Traffic Manager** – An intelligent DNS service that uses DNS to implement global traffic management. Where Azure DNS always provides the same DNS response to a given DNS query, in Azure Traffic Manager the same query may result in one of several responses, depending on a number of factors which you control, such as where the end-user is located or which of your service endpoints is currently available.
  + **App Service Domains** – Allows purchasing of domain names, which can then be hosted in Azure DNS. This service is integrated with Azure App Service but can be used for any domain registration even if App Service is not being used.
  + **Azure-provided DNS** – Sometimes called Internal DNS, it allows the VMs in your virtual network to find each other, using DNS queries based on the hostname of each VM. The DNS queries are internal (Private) to the virtual network.
  + **Recursive DNS** – A service provided by Azure for DNS name resolution from your Azure VMs or other Azure services. You can also configure your VMs to use your own DNS server instead. This is sometimes informally called bring your own DNS. This is common when joining your VMs to a domain controller.
  + **Reverse DNS** – Provides the ability to configure the reverse DNS lookup for an Azure-assigned public IP address. (Reverse DNS lookup zones for IP blocks you own can be hosted in Azure DNS.)

**Creating and delegating a DNS Zone to Azure DNS**

* A DNS Zone is a resource in Azure DNS. Creating a DNS zone resource allocates authoritative DNS name servers to host the DNS records for that zone. Azure DNS can then be used to manage those DNS records.
* **The assigned name servers will vary between zones, so if you’re setting up multiple zones in Azure DNS you need to check the name servers on each one. Don’t assume that the name servers will be the same across all your zones.**
* A glue record is a DNS server record that is not authoritative for the zone and is used to avoid a condition of impossible dependencies for a DNS zone. These IP addresses might change in the future.
* **Azure DNS treats child zones as entirely separate zones. Therefore, delegating a child zone follows the same process as delegating the parent zone:** 
  + **Create the child zone resource**
  + **Identify the name servers for the child zone. These will be different to the name servers assigned to the parent zone.**
  + **Create NS records in the parent zone to delegate the child zone. The name of the NS records should be the child zone name and the RDATA in the NS records should be the child zone name servers**

**Managing DNS records in Azure DNS**

* Each record in the domain name system includes the following properties:
  + **Name**. The name of the DNS record is combined with the name of the DNS zone to form the fully qualified domain name (FQDN). For example, the record www in zone contoso.com corresponds to the FQDN www.contoso.com.
  + **Type**. The type of DNS record determines what data is associated with the record and what purpose it is used for. A list of record types supported by Azure DNS is provided in the next section.
  + **TTL**. The TTL (or Time-to-Live) tells recursive DNS servers how long a DNS record should be cached.
  + **RDATA**. The data returned for each DNS record. The type of data returned depends on the DNS record type. For example, an A record will return an IPv4 address, whereas a CNAME record returns another domain name.
* **The collection of records in a DNS zone with the same name and the same type is called a resource record set. Records in Azure DNS are managed using record sets. Record sets are a child resources of the DNS zone and can contain up to 20 individual DNS records.**
* Azure DNS supports all commonly used DNS record types. The full list of supported record types—together with a description of each is below
  + **A** – Maps IPv4 address
  + **AAAA** – Maps IPv6 address
  + **CAA** – Used to specify which certificate authorities can issues certificates for a domain. Must be configured using CLI or PowerShell
  + **CNAME** – Provides a mapping from one DNS name to another. The DNS standards do not allow CNAME records at the zone apex.
  + **MX** – Used for mail server configurations
  + **NS** – An NS record set at the zone apex containing the name servers for the DNS zone is required by the DNS standards.
  + **PTR** - Used for reverse DNS lookups in reverse lookup zones
  + **SOA** – Required at the apex of every zone. This is created and deleted with DNS zone resource.
  + **SRV** – Used for service discovery for Kerberos to Minecraft to SIP.
  + **TXT** – Used for a wide range of applications, including email Sender Policy Framework (SPF).
    - **NOTE**: SPF records are used to identify legitimate mail servers for a domain and help prevent spam.

**Alias Records**

* **Alias records allow you to define the target of the DNS record implicitly by referencing another Azure resources. The value of the DNS record is populated automatically based on the resource it references and is updated automatically if that resource changes.**
* Alias records can reference three different resource types:
  + An A or AAAA. These records can reference a public IP address, of type IPv4 or IPv6, respectively.
  + A, AAAA, or CNAME. These records can reference a Traffic Manager profile. This exposes the dynamic, traffic-managed name resolution of the Traffic Manager directly within a record in your DNS domain. Prior to this feature, you had to create a CNAME record from your domain to a record in the trafficmanager.net domain provided by Azure Traffic Manager.
  + An A, AAAA or CNAME. These records can also reference another record in the same DNS zone. This lets you create synchronized records with ease.
* **When creating a DNS zone, the location field only specifies the resource group location. It does not apply to the DNS zone resource itself, which is global rather than regional.**
* To set up DNS delegation for the DNS zone, these name servers must be listed in the corresponding NS records in the parent zone. If the domain was purchased using the Azure App Service Domains service, this will be done automatically.
* **VMs are configured to use Azure’s recursive DNS servers. These provide name resolution for Internet-hosted domains, plus private VM-to-VM name resolution within a virtual network.**

**Bring your own DNS**

* Alternatively, you can configure your own DNS settings, which will be configured during the DHCP exchange on the VMs instead. This enables you to specify your own DNS servers, either in Azure or running on-premises. With your own DNS servers, you can support any DNS scenario, including scenarios not supported by the Azure-provided service.
* **If you are using your own DNS servers, then you should not configure it within the VM itself because the platform is unaware of the settings you have chosen. Instead, you should configure the options within the virtual network settings.**
* DNS server settings at the virtual network level apply to all VMs in the virtual network. You can apply VM-specific DNS server settings within each network interface. Where multiple VMs are deployed in an availability set, setting DNS servers at the network interface, all VMs in the availability set are updated.
* **By using private DNS zones, you can use your own custom domains names. It provides an alternative approach to name resolution within and between virtual networks.**
* Custom DNS settings can be configured at the VNet level and the network interface level but not at the subnet level.
* **If you make changes to the DNS settings at the virtual network level, any affected virtual machines must restart to pick up the new settings. If you make changes to DNS settings and the network interface level, the affected VM (or VMs across the availability set, if used) will restart automatically to pick up the new settings.**
* Challenge of using your own DNS server is that you will need to register each VM in your DNS services. To do this, you can configure the DNS service to accept Dynamic DNS queries, which the VM will send when it boots

**Skill 4.2 - Secure Access to virtual networks**

* **Network Security Groups (NSGs) – Allow you to control which network flows are permitted into and out of your virtual networks and virtual machines. It’s a standalone Azure resource, which acts as a networking filter. Each NSG contains a list of security rules that are used to allow or deny inbound or outbound traffic.**
* NSGs are associated with a subnet or with a specific VMs network interface.
* **NSG rules are enforced based on their priority. Priority values start from 100 and go to 4096. Rules will be read and enforced starting with 100 and are followed by 101, 102, and so on. When a rule is found that matches the traffic under consideration, the rule is applied, and all further processing stops—subsequent rules are disregarded.**

**Service Tags**

* Service tags are Platform-defined shortcuts that map to the IP ranges of various Azure services. The IP ranges associated with each service tag are updated automatically whenever the IP addresses used by the service change.
* **Service tags are used in NSG rules as a quick and reliable way of creating rules that control traffic to each service.**
* Service Tags control access to the service but not to a specific resource within that service.
* **All NSGs have a set of default rules. You cannot add to, edit, or delete these default rules. However, since they have the lowest possible priority, they can be overridden by other rules which you create.** 
  + **Virtual Network** – Traffic originating and ending in a virtual network is allowed both inbound and outbound directions
  + **Internet** – Outbound traffic is allowed, but inbound traffic is blocked
  + **Load Balancer** – Allows the Azure Load Balancer to probe the health of your VMs and role instances. If you are not using a load balanced set, you can override this rule.

**Application security groups**

* As you have seen, NSG rules are like traditional firewall rules and are defined using source and destination IP blocks. They enable you to segment your network traffic into application tiers, which are segmented into separate subnets. This creates some management challenges.
* **Application security groups (ASGs) address these challenges by offering an alternative approach to network segmentation. They allow you to achieve the same goal of segmenting your application into separate tiers, and they strictly control the permitted network flows between tiers. However, ASGs do not require that you associate each tier with a separate subnet, so all the challenges associated with planning and managing subnets fall awayAll VMs can be placed into a single subnet so they can have the same NSG to define network flows between application tiers.**
* An NSG contains rules defining the permitted traffic flows between application tiers
* **ASGs allow you to configure network security as a natural extension of an application’s structure, which allows you to group virtual machines and define network security policies based on those groups.**
* Steps to configure an ASG –
  + Create an application security group resource for each server group. This resource has no properties, other than its name, resource group, and location
  + Associate the network interface from each VM with the appropriate ASG. This defines which group each VM belongs to
  + Finally, define your network security group rules using ASG names instead of explicit IP ranges. Like how rules are configured using named service tags
* **NSGs are used to define the rules of how traffic is filtered for your IaaS deployments in Azure. We have seen how to create NSG resources and define the NSG rules. However, these NSGs, by themselves, are not effective until they are associated with a resource in Azure.**
* Each NIC or subnet can only be associated with one NSG. However, a single NSG can be associated with multiple NICs and/or subnets. It is possible to have a multi-NIC VM, and you can associate the same or different NSG to each Network Interface.
* **Alternatively, NSGs can be associated with a subnet; in that case, they apply to all traffic to and from resources in that subnet. This approach is useful when applying the same rule across multiple VMs.**
* Effective Security Rules view is designed to provide insight to drill into each NSG rule and see the exact list of source and destination IP prefixes that have been applied, regardless of how the NSG rule was defined.
* **Effective Security Rules are used at the NIC level for viewing rules applied to a VM**

**How NSGs are Applied**

* Microsoft does not recommend deploying NSGs to subnets and NICs within the same subnet. However, although Microsoft does not recommend it, this configuration is supported, and it’s important to understand how NSGs are applied when deployed in this way.
  + For inbound traffic, first the NSG at the subnet is applied, followed by the NSG at the NIC. Traffic only flows if both NSGs allow the traffic to pass.
  + For outbound traffic, the sequence is reversed. First, the NSG at the NIC is applied, followed by the NSG at the subnet. Again, traffic only flows if both NSGs allow the traffic to pass.
  + In all cases, rules within each NSG are applied in priority order, with the first matching rule applicable first.

**Implement Azure Firewall**

* **Azure Firewall is a managed service that provides out-of-the-box network security for Azure resources. Because this is a managed service, its added advantages are that it offers high availability and scalability. In a typical enterprise environment, outbound network access restrictions are mandatory because the provide direct access to external web world. Azure Firewall provides an ability to limit the outbound IP addresses and ports that are allowed to communicate within an Azure subnet. It also provides additional benefits such as outbound SNAT support, Inbound DNAT support, and Azure Monitor logging.**
* Azure Firewall allows you to create and configure application and network rules. Application rules are created with the list of fully qualified names that are allowed to be accessed from a subnet. Network rules are a combination of source and destination IP addresses along with their ports and protocols.
* **Hub-and-spoke is the standard deployment model for Azure Firewall, where the Firewall is hosted on its own VNet, and other resources are placed in peered VNets in the same region with one or more subnets.**
* Azure Firewall must be hosted in a subnet named AzureFirewallSubnet with a minimum **/26** address space for the Azure Firewall to provision more VMs to accommodate scaling.
* **A default route table will need to be created to route the outbound requests through the firewall**
* To filter outbound web traffic, you will need to create an application rule.
* **By default, infrastructure FQDNs are allowed by Azure Firewall with a built-in rule collection. You can override this by creating a deny all applications rule collection.**

**Implement Azure Bastion Service**

* Azure Bastion provides secure connections to Azure Virtual Machines using the SSL channel through a browser directly without using any external client. It uses port 443
* **Azure Bastion service is provisioned within a VNet within a separate subnet. If you have multiple VNets in your environment, you will need to deploy Azure Bastion for each VNet separately.**
* To evaluate an effective security rule for a network interface of an Azure VM, that Azure VM must be started
* **Azure Firewall supports blocking traffic based on the FQDN of the destination while NSGs do not.**

**Skill 4.3 – Configure Load Balancing**

* **Azure application gateway is a type of load balancer that can manage traffic for web applications. The web traffic routing occurs at the application layer (OSI Layer 7). Azure Application Gateway offers additional features such as SSL/TLS termination, autoscaling, URL-based routing, redirection, and the like**
* Azure Load Balancer is a fully managed load-balancing service, which is used to distribute inbound traffic across a pool of back-end servers running in an Azure virtual network. Can receive traffic on either Internet-facing or Intranet-facing endpoints and supports both UDP and TCP traffic. Operates at the transport layer (OSI Layer 4) to route inbound and outbound connections at the packet level.
* **Azure Load balancer supports automatic failover between back-end servers based on health probes and enables high availability applications.**
* Application Gateway routes application web traffic to defined resources in a back-end pool.

**Basic and Standard Load Balancer tiers**

* **Azure Load Balancer is available in two pricing tiers (SKUs): Basic and Standard. These tiers offer different levels of scale, features, and pricing.** 
  + **Basic** – Does not support Availability Zones, supports up to 300 backend servers, VMs must be in the same availability set or a single VM Scale Set, supports TCP and HTTP health probes. Supports Azure Monitor for public load balancer only, alerts and backend pool health count. Open by default, can optionally restrict flow using NSGs. Single outbound IP, not configurable. No provided SLA, free though
  + **Standard** – Supports Zone-specific or zone-redundant deployments, including cross-zone load-balancing. Up to 1,000 servers any mix of VMs, availability sets, and VM scale sets in the same VNet. Supports TCP, HTTP, and HTTPS health probes. Rich metrics provide via Azure Monitor. Inbound flows closed by default; access less-permitted inbound flows using NSGs. Supports multiple outbound IP addresses that are configurable via outbound rules. Supports HA ports, TCP reset on idle timeout, and faster management operations. Based on the number of rules and data processes. 99.99% availability for a data path with two healthy VMs.
* When used with IaaS VMs, each load balancer can support multiple frontend IP configurations

**Frontend IP configuration**

* **Azure Load Balancer supports two modes: internal Load Balancer or public Load Balancer. In each case, the frontend IP configuration defines the endpoint upon which the Load Balancer receives incoming traffic.**
  + **Internal Load Balancer**. Used to load-balance traffic for Intranet-facing applications, or between application tiers. The frontend IP configuration references a subnet, and an IP address from that subnet is allocated using either dynamic or static assignment to the Load Balancer.
  + **Public Load Balancer**. Used to load-balance traffic for Internet-facing applications. The frontend IP configuration references a separate public IP address resource, which is used to receive inbound traffic.
* A public Load Balancer must be associated with a public IP address resource. If the load-balancer uses the standard pricing tier, then the public IP address must also use the standard pricing tier. Standard tier Load Balancers support both zone specific and zone redundant deployment options.

**Backend configuration**

* **The backend pool defines the backend servers over which the Load Balancer will distribute incoming traffic.**
* When using a basic-tier Load Balancer, this backend pool must comprise either a single virtual machine, virtual machines in the same availability set, or a VM scale set. (Traffic will be distributed to all virtual machines in the VM scale set.) You cannot distribute traffic to multiple virtual machines unless they are members of the same availability set or VM scale set.
* **With a standard-tier load-balancer, these restrictions are lifted. Backend pools can comprise a combination of virtual machines across availability sets and VM scale sets.**

**Health Probes**

* **Azure Load Balancer supports continual health probing of backend pool instances to determine which instances are healthy and able to receive traffic. The Load Balancer will stop sending traffic flows to any backend pool instance that is determined to be unhealthy. Unhealthy instances continue to receive health probes, so the Load Balancer can resume sending traffic to that instance once it returns to a healthy state.**
* **Azure Load Balancer Supports these health probes**
  + **TCP** – Probes attempt to initiate a connection by completing a three-way TCP handshake. If successful, the connection is then closed with a four-way handshake.
  + **HTTP** – Probes issue an HTTP GET with a specified path
  + **HTTPS** – Probes are like HTTP probes, except that a TLS/SSL wrapper is used. Only supported on the standard tier load balancer.
* **All three probe types must also specify the probe port or the interval. The minimum probe interval is five seconds, and the minimum consecutive prove failure threshold is two seconds.**
* An endpoint is marked as unhealthy in the following settings:
  + For HTTP or HTTPS probes only, the endpoint returns an HTTP status code other than 200 OK.
  + The probe endpoint closes the connection using a TCP reset.
  + The probe endpoint fails to respond during the timeout period snf for a consecutive number of requests. The number of failed requests required to mark the endpoint unhealthy is configurable.
* **Configuring a dedicated health check page, enables each backend server to implement custom application logic to decide whether it is healthy.**
* When configuring NSGs for backend servers, it is important to allow both inbound traffic and probe traffic.
* **The last setting, Floating IP (direct server return), is only recommended when load-balancing traffic for a SQL Server Always On Availability Group listener. For other scenarios, the Floating IP setting should be left disabled.**
* Standard tier load balancer supports routing diagnostics automatically to Azure Monitor.
* **Azure Application Gateway should be deployed into a dedicated subnet of an Azure virtual network**
* Log Analytics workspace can server as the destination for Azure Load Balancer diagnostics and provides the ability to run Kusto Query Language (KQL) queries directly from the Azure portal.

**Skill 4.4 – Monitor and troubleshoot virtual networking**

* **Azure Network Performance Monitor (NPM) - Is a networking monitoring solution for hybrid networks that enables you to monitor network connectivity and performance between various points in your network, both in Azure and On-premises. It can provide reports of network performance and raise alerts when network issues are detected.**
* Network Performance Monitor (NPM) provides three services -
  + **Performance Monitor** – Used to monitor connectivity between various points in your network, both in Azure and on premises. You can monitor nodes at both ends, and you can gather data about connectivity, packet loss, latency, and available network paths.
  + **Service Connectivity Monitor** – Used to monitor outbound connectivity from nodes on your network to any external service with an open TCP port, such as websites, applications, or databases. This measures latency, response time, and packet loss, enabling you to determine whether poor performance is caused by network or application issues.
  + **ExpressRoute** – Used to monitor end-to-end connectivity between your on-premises network and Azure over ExpressRoute. Can use auto-discover for your ExpressRoute network topology. It can then then track your ExpressRoute bandwidth utilization, packet loss, and latency. These are measured at the circuit, peering and Azure virtual network level.
* **NPM also provides a dashboard giving an overview of the network status. as well as detailed per-service charts and reports.**
* **NPM is a Log Analytics solution**. Log Analytics agents are installed on each node used to measure network connectivity and performance. These agents perform synthetic transactions over TCP or ICMP to measure network performance. Data gathered from these agents is channeled into a Log Analytics workspace. NPM analyzes this data to provide both reporting and alerting.
* **NPM can be installed from the Azure Marketplace. It is also available from Network Watcher, an Azure Service that acts as a hub for a wide range of network monitoring and diagnostic tools.**
* NPM requires you to create a Log Analytics workspace or select an existing workspace to use.
* **Places to install NPM Agents -**
  + **To monitor a given network link, agents should be installed on servers at both ends of that link**
  + **To monitor connections between subnets, an agent should be installed on at least one server in each subnet.**
* Installing an NPM agent on an on-premises server
  + Download the OMS agent
  + You will need the Workspace ID and Primary key to install the agent.
  + Download a PowerShell Script to open the necessary firewall ports.
  + Default port used is **TCP** **8084**

**Performance Monitor**

* **Performance Monitor enables you to monitor packet loss and latency between your endpoints, both in Azure and on-premises. A VM or server running the Log Analytics agent is required at both ends of each monitored connection. Can be setup with TCP or ICMP-based monitoring.**

**Service Connectivity Monitor**

* Service Connectivity Monitor – Used to test outbound connectivity from your network to open a TCP port, such as a website, application, or database. It supports pre-configured endpoints for Microsoft Office 365 and Dynamics. You can also configure custom tests to arbitrary endpoints. Once configured, Service Connectivity Monitor will generate packet loss and network performance charts (Showing latency and response times) for each tested endpoint

**ExpressRoute Monitor**

* **ExpressRoute Monitor allows you to monitor end-to-end network connectivity and performance between on-premises and Azure endpoints over ExpressRoute connections. It can auto-detect ExpressRoute circuits and your network topology, and track bandwidth utilization, packet loss and network latency. Reports are available for each ExpressRoute circuit or peering, and also for each Azure virtual network using ExpressRoute.** **Once configured, it takes 30-60 minutes for the first ExpressRoute reporting data to become available.**

**Use network resource monitoring**

* Azure Application Gateways and Azure Load Balancers can be used to send logs to a storage account, streamed to an EventHub, or integrated with an Azure Log Analytics workspace. Azure Application Gateway Analytics Log Analytics solution provides a pre-configured dashboard and charts showing App Gateway status.

**Use Azure Network Watcher**

* **Azure Network Watcher provides a central hub for a wide range of network monitoring and diagnostic tools. These tools are valuable across a wide range of network troubleshooting scenarios and provide access to other tools like Network Performance Monitor and Connection Monitor.**
* Network Watcher is enabled as a single instance per Azure region. It is not deployed like a conventional resource, but it does appear as a resource in a resource group.
* **Any subscription containing a virtual network resource will automatically have Network Watcher enabled. Network Watcher can also be deployed in Azure Portal or the CLI.**
* **Some of the Network Watcher tools require the Network Watcher VM extension to be installed on the VM being monitored. Its available for both Windows and Linux VMs. It is installed automatically when using Network Watcher via the Azure portal.**

**IP Flow Verify (Part of Network Watcher)**

* IP Flow Verify tool provides a quick and easy way to test whether a given network flow will be allowed into or out of an Azure virtual machine**. It will report whether the requested traffic is allowed or blocked and in the latter case, which NSG rule is blocking the flow.** Useful tool for verifying that NSGs are correctly configured. Works by simulating the requested packet flow through the NSGs applied to the VM. For this reason, the VM must be in a running state.

**Next Hop (Part of Network Watcher)**

* Next Hop Tool provides a useful way to understand how a VMs outbound traffic is being directed. For a given outbound flow, it shows the next hop IP address and type and the route table ID of any user-defined route in effect. Possible next hop types are -
  + **Internet**
  + **VirtualAppliance**
  + **VirtualNetworkGateway**
  + **VirtualNetwork**
  + **VirtualNetworkPeering**
  + **VirtualNetworkServiceEndpoint**
  + **None (Used for user-defined routes)**

**Packet Capture (Part of Network Watcher)**

* **Packet Capture tool allows you to capture network packets entering or leaving your virtual machines. It is a powerful tool for deep network diagnostics. You can capture all packets, or a filtered subset based on the protocol and local and remote IP addresses and ports. Packet Captures are stored as a file on the VM or in an Azure storage account, in which case NSGs must allow access from the VM to Azure Storage.**

**Network Topology (Part of Network Watcher)**

* Network Topology provides a diagrammatic view of the resources in your virtual network. It is not a diagnostic or alerting tool. It is a quick and easy way to review your network resources and manually check for misconfiguration. Limitation is that is only shows the topology within a single virtual network. All common network resource types are supported, although for application gateways, only the backend pool connected to the network interface is shown.
  + Underlying topology data can be downloaded in JSON format via Azure PowerShell or Azure CLI using the ***Get-AzNetworkWatcherTopology*** cmdlet or the ***az network watcher show-topology*** command, respectively.

**Troubleshoot external networking**

* **Another pair of useful tools to investigate issues with external networks are the Connection Monitor and Connection Troubleshoot tools in Network Watcher. These are discussed in the next section: “Troubleshoot virtual network connectivity.” In this section, we discuss VPN Troubleshoot, which is designed specifically to diagnose problems with VPN connections.**

**VPN Troubleshoot (Part of Network Watcher)**

* VPN Troubleshoot tool – Provides automated diagnostics of Azure VPN gateways and connections. The results provide a detailed report on gateway health and connection health, providing accurate pointers regarding common issues that might occur when enabling informed remediations. VPN troubleshoot only supports route-based VPN gates so Site-to-Site VPNs and VNet-to-VNet connections. **Does not support ExpressRoute connections or Point-to-Site connections.**
  + During the troubleshooting process, logs are written to a storage account. Account must be created before starting the troubleshooting process.

**Troubleshoot virtual network connectivity**

* **A number of the tools we have already seen can be useful for troubleshooting connectivity issues between and within virtual networks. Network Watcher offers two more tools that are particularly useful in this scenario: Connection Troubleshoot and Connection Monitor.**
  + **Connection Troubleshoot** – Is a Network Watcher feature designed to allow you to test the connectivity between an Azure VM or an App Gateway and another endpoint, either another Azure VM or an arbitrary Internet or Intranet endpoint.
  + **Connection Monitor** – Similar to connection troubleshoot in that it uses the same mechanism to test the connection between an Azure VM or App Gateway and another endpoint. The difference is that Connection Monitor provides ongoing connection monitoring, whereas Connection Troubleshoot provides a point-in-time test.

**Skill 4.5 - Integrate an On-Premises Network with an Azure Virtual Network**

* Hybrid networks are commonly used for Intranet applications, which may be hosted in Azure but only accessed from the on-premises network. They are also used by Azure applications that require access to an on-premises resource, such as a database.
* **Hybrid networks provide connectivity between the private IP space of the on-premises network and the private IP space of the Azure VNet. The VNet can be thought of as an extension of the existing on-premises network. The concept is similar to extending the on-premises network to a new office location.**

**Create and configure Azure VPN Gateway**

* Virtual Network Gateway – Allows you to create connections from your virtual network to other networks. When creating a gateway, you must specify if it will be used for VPN connections or ExpressRoute connections. Virtual Network Gateway used for VPN connections are called a VPN gateway, while those used for ExpressRoute connections are called ExpressRoute gateways.

**Gateway subnets**

* **VPN Gateways can only be deployed to a dedicated gateway subnet within the VNet. VPN gateway is implemented using Azure virtual machine. While the minimum size for the gateway subnet is a CIDR /29. The Microsoft-recommended best practice is to use a CIDR /27 address block to allow for future expansion.**
* VPN connection between an on-premises network and an Azure VNet can only be established if the network ranges do not overlap. Network address ranges should be planned carefully to avoid restricting future connectivity options.
* **VPN Gateways are available in several pricing tiers, or SKUs.**
  + **Basic** – 10 Max site-to-site VPN connections. Throughput of 100 Mbps
  + **VpnGw1 and VpnGw1Az** – 30 Max site-to-site VPN connections. Throughput of 650 Mbps.
  + **VpnGW2 and VpnGw2Az** – 30 Max site-to-site VPN connections. Throughput of 1 Gbps
  + **VpnGw3 and VpnGw3Az** – 30 Maxi site-to-site VPN connections. Throughput of 1.25 Gbps
* **You can resize a gateway between the VpnGw1, VpnGw2, and VpnGw3 tiers. You cannot however resize a basic tier gateway.**

**Border Gateway Protocol (BGP)**

* Border Gateway Protocol (BGP) is a standard used in the Internet to exchange routing information between networks. BGP can be optionally enabled on your VPN gateway, if the on-premises gateway also supports it. If used, it enables the VPN gateway and the on-premises gateway to exchange routing information automatically, avoiding the need to configure routes manually.
* **BGP also enables high availability redundant connections (see next section) advanced features such as transit routing across multiple networks. It is also used where a VPN connection is used as a failover in case the primary connection, using ExpressRoute, were to fail.**

**High Availability**

* By default, each VPN gateway is deployed as two VMs in an active-standby configuration. **To reduce downtime in the event the active instance fails, an active-active configuration can also be used.** In active-active both gateways have their own public IP addresses, and two connections are made to the on-premises VPN endpoint.
* **Dual on-premises VPN endpoints can be used but requires BGP to be enabled. Works in an active-standby or active-active VPN gateways. Active-active gateway with dual on-premises endpoints provides a fully redundant configuration, avoiding a single point of failure.**
* Before creating a VPN gateway, you will need to first create the gateway subnet. When creating the gateway subnet, there is no special parameter or cmdlet name to denote that this is a gateway subnet rather than a normal subnet. The only distinction that identifies a gateway subnet is the subnet name, ***GatewaySubnet***.

**Create and configure Azure ExpressRoute**

* **ExpressRoute is a secure and reliable private connection between your on-premises network and the Microsoft cloud. The connection is provided mostly by a third-party network provider who has partnered with Microsoft to offer this ExpressRoute services. This third party is known as the *ExpressRoute* *provider*. Alternatively, with ExpressRoute Direct, you can connect directly to the MSEE.**
* ExpressRoute provides connectivity to all Microsoft Cloud Services, unlike a Site-to-Site VPN which only provides connectivity to your Azure VNet.

**ExpressRoute Connectivity models**

* ***ExpressRoute*** connectivity can be established in one of three ways. The capabilities and features of ExpressRoute are the same in each case.
  + If your network already has a presence at a co-location facility with a cloud exchange your co-location provider can establish a virtual cross-connection with the Microsoft Cloud. This provides either a layer 2 or a managed layer 3 connection
  + Your connectivity provider may be able to provide a point-to-point ethernet connection from their network to your on-premises network. Again, this approach offers either a layer 2 or managed layer 3 connection.
  + You're existing IPVPN WAN provider may be able to integrate ExpressRoute into your WAN, if they are registered as an ExpressRoute provider.

**Circuits and peering**

* **An ExpressRoute circuit is an Azure resource used to represent the logical connection between your on-premises network and Microsoft. Each circuit is identified by GUID called a service key (s-key), which is shared with your connectivity provider. Each circuit has a fixed bandwidth, and a specific peering location.** 
  + **Bandwidth Options** – 50 Mbps, 100 Mbps, 200 Mbps, 500 Mbps, 1 Gbps, 2 Gbps, 5 Gbps, and 10 Gbps. Bandwidth can either be metered or unlimited.
    - **Metered** – All inbound data transfer is free of charge, and all outbound data transfer is charged based on a predetermined rate. Users are also charged a fixed monthly port fee (Based on high-availability dual ports)
    - **Unlimited** - All inbound and outbound data transfer is free of charge. Users are charged a single fixed monthly port fee (Based on high-availability dual ports)
* ExpressRoute Circuits offer two peering options, also known as routing domains. Each circuit can use either one or both peerings
  + **Azure Private Peeing** – Provides connectivity over the Intranet address space into your Azure virtual network. This peering is considered a trusted extension of your core network into Azure
  + **Microsoft Peering** – Provides connectivity over the Internet address space into Microsoft services such as Office 365, Dynamics 365, and Internet-facing endpoints of Azure platform (PaaS) services.
* **Older circuits can be using a third peering model, Azure Public Peering, which provides connectivity to Azure PaaS services only. This is deprecated for new circuits.**
* Each ExpressRoute circuit has two connections from your network edge to Microsoft edge routers, configured using BGP. Microsoft requires dual BGP connections from your edge to each Microsoft edge router.
* **ExpressRoute Premium add-on allows to extend connectivity to all Microsoft data centers worldwide. This add-on also raises the number of routes permitted for the Azure Private Peering 4,000 to 10,000. It also increases the number of virtual networks that can be connected to each ExpressRoute circuit from 10 to between 20 and 100 (depending on the bandwidth of the circuit)**
* When creating an ExpressRoute circuit, you must specify both the peering location and the location of the ExpressRoute circuit resource. There are independent settings, although Microsoft suggests the best practice to be nearby.
* **Billing for the ExpressRoute circuit begins immediately upon resource creation and does not depend upon completing the configuration with the ExpressRoute provider. ExpressRoute circuits can be expensive, so care is advised. It is a good practice to restrict the ability to create ExpressRoute circuits using Azure Policy.**
* The Service key will need to be shared with your ExpressRoute provider.
* **Virtual networks are connected to ExpressRoute circuits using an ExpressRoute gateway. An ExpressRoute gateway is a virtual network gateway, created with the ExpressRoute option (rather than the VPN option, used to create VPN gateways). Just as with VPN gateways, the ExpressRoute gateway must be created in the gateway subnet of the virtual network.**

**Azure Virtual WAN**

* Azure Virtual WAN is a combination of many networking, security, and routing functionalities together to provide a single operational interface for various networking solutions. Azure Virtual WAN facilitates Point-to-site, Site-to-site, ExpressRoute connectivity, and Azure Firewall configuration all at one place. The Azure Virtual WAN can be leveraged to either use one of the scenarios or use multiple of them. The Virtual WAN leverages hub-and-spoke topology. The hubs are nothing, but an Azure regions and spokes are considered as individual endpoints. The hubs relate to each other using Microsoft’s backbone network wherein for spoke connectivity various VPN devices are used. Microsoft has partnered with many VPN solution providers with ability to export the device info, configure and establish connectivity with Azure Virtual WAN.
* **With Basic WAN, you can only create Basic Hubs. Basic Hubs are only capable of creating site-to-site connections.**
* /29 is the minimum size of the virtual network subnet required to create an Azure VPN gateway
* **Azure ExpressRoute supports connectivity to Azure platform services, such as Cosmos DB.**
* Azure Virtual WAN Basic supports only Site-to-Site VPN.
* **2x500 Mbps is the minimum throughput that can be allocated to an Azure Virtual WAN hub gateway is a single scale unit, which represents gateway instances with 500 Mbps throughput each.**

**Chapter Summary -**

* + Azure VNets are isolated networks using a private IP address space
  + **Virtual Networks are divided into subnet, which allow you to isolate workloads**
  + Azure reserves the first 4 and last IP address in each subnet. The first IP address allocated to VM is therefore typically the .4 IP address.
  + **Private IP address for a VM is assigned from a subnet and configured as settings on the IP configuration of a network interface resource.**
  + A VM can be associated with one or more network interfaces, and each network interface can contain multiple IP configurations.
  + **Dynamic IP addresses are released when the VM is stopped**
  + Public IP addresses are managed as a standalone resource, which can be associated with a network interface IP configuration
  + **Public IP addresses support two pricing tiers (SKUs) - Basic tier supports dynamic and static assignment and provides open connectivity (Can be restricted using NSGs). The standard tier supports zone-redundant deployments, use static allocation only, and is closed by default (Access is enabled using NSGs)**
  + User-Defined Routes (UDRs) - Change the default behavior of subnets allowing you to direct outbound traffic to other locations. Typically, traffic is sent through a virtual appliance such as a firewall.
  + If a UDR is used to send traffic to a virtual appliance, IP forwarding must be enabled on the NIC of the virtual appliance VM.
  + Routing Outbound Internet traffic via a VPN connection to a network security device is known as forced tunneling.
  + The effective routes for each network interface can be reviewed to help diagnose routing issues.
  + VNets can be connected using either VNet peering or VNet-to-VNet VPN connections
  + To connect two VNet, they must have non-overlapping IP address spaces.
  + Virtual Networks can be connected using VNet peering. This is supported both within a region or across regions.
  + By default, peered VNets appear and perform as a single network. There is an option limit connectivity, in which case NSG rules must be used to define the permitted connections.
  + VNet peering allows VMs to see each other as one network, but their relationships are non-transitive. If VNetA and VNetB are peered and VNetB and VNetC are peered VNetA and VNetC are not peered.
  + Hub-and-spoke network architecture is a common approach where a separate spoke VNets are used by each application, peered to a hub VNet containing a network virtual appliance (NVA). The peering connections must enable Allow Forwarded traffic.
  + Using VNet peering to provide access to a central VNet containing shared services, such as Active Directory domain controllers, is known as service chaining.
  + Alternatively, virtual networks can be connected using a VNet-to-VNet VPN connection.
  + A virtual network gateway can be used to create VPN connections between virtual networks.
  + Alternatively, virtual networks can be connected using a VNet-to-VNet VPN connection.
  + Virtual Network gateway can be used to create VPN connections between virtual networks (and is then called a VPN gateway)
  + The size of the VPN gateway should be chosen based on the throughput required.
  + The GatewaySubnet is a special subnet that is only used for virtual network gateways.
  + A VPN gateway can be peered VNets. The peering connections must enable the settings to Use Remote Gateway (On the peering toward the gateway) and Allow Gateway Transit (On the peering from the gateway)
  + Both global VNet peering and VNet-to-VNet VPN connections route traffic between Azure regions over the Microsoft backbone network, not the public Internet.
  + Azure DNS provides an authoritative DNS service for hosting Internet-facing domains
  + DNS Zones in Azure DNS must be delegated from the parent domain. This is achieved by setting up appropriate NS records in the parent domain, pointing to the name servers assigned by Azure DNS.
  + DNS records in Azure DNS are managed using record sets, which are the collection of records with the same name and the same type.
  + DNS records at the zone apex use the record name @. You cannot create records with the CNAME record type at the zone apex.
  + Azure DNS Alias records allow DNS records to reference other Azure resources, such as public IP address.
  + Azure DNS also supports private DNS zones, which can also be used to enable VM-to-VM DNS lookups.
  + Network Security Groups are used to create firewall rules to control network flows.
  + NSGs can be applied at the subnet level, or on individual VM network interfaces.
  + Each NSG includes a list of default rules, which can be overridden using user-defined rules. Rules are applied in priority order.
  + Source and destination IP address ranges in NSG rules can be specified explicitly using CIDR ranges.
  + IP address ranges can also be specified using application security groups (ASGs). ASGs allow NSG rules to be defined for groups of VMs without needing to allocate the VMs into separate subnets.
  + Tools to help identifying the required NSG rules include service map and NSG flow logs.
  + Effective Security Rules can be reviewed for each network interface. This allows you to see the exact IP ranges used by each service tag and ASG
  + Azure Firewall is a managed service which provides out of the box network security to secure Azure resources. Azure Firewall allows us to create and configure application and network rules.
    - Application Rules – Created with the list of fully qualified names which allowed to be accessed from a subnet
    - Network Rules – Are combination of source and destination IP addresses along with their ports and protocols.
  + Azure Bastion is provisioned within a Virtual Network within a separate subnet called “AzureBastionSubnet”.
  + Azure Application Gateway is a type of Load Balancer which can manage traffic for web applications. The web traffic routing occurs at application layer (OSI Layer 7). Azure Application Gateway routes application web traffic to defined resources in a backend pool.
  + Azure Load Balancer (ALB) - A fully managed, high-performance load-balancing service for TCP and UDP traffic. Operates at the transport layer (OSI Layer 4). Unlike App Gateway it does not have visibility into application-level traffic.
  + ALB can be deployed with either a public (Internet) or private (Intranet) frontend IP address.
  + ALB comes in two pricing tiers – Standard supports availability zones and more flexible backend pools and several other features. Basic is free of charge.
  + ALB load-balancing configuration comprises frontend IP configuration, backend pool, health proves, and load-balancing rule.
  + ALB supports port forwarding, using inbound NAT rules. This maps a specific frontend port to a specific backend port on a specific backend server.
  + Network Performance Monitor provides monitoring for hybrid networks. It supports performance monitor, connectivity monitor, and ExpressRoute monitor to monitor ExpressRoute connections.
  + Network Watcher is a central hub providing access to a wide range of networking tools.
  + IP Flow Verify is a Network Watcher feature used to test if a given network flow is allowed in or out of an Azure VM.
  + Next Hop is used to determine the next hope address and routing rule for a given network flow.
  + Packet Captures enables network traffic on a given VM to be captured, either locally or to an Azure storage account.
  + Network Topology creates a diagrammatic representation of the resources in your virtual network.
  + VPN Troubleshoot provides automated, in-depth troubleshooting of VPN connections.
  + Connection Troubleshoot allows you to test the connectivity between two Azure VMs or between a VM and an arbitrary external endpoint.
  + Connection Monitor enables long-term connection monitoring, using similar diagnostics as used by Connection Troubleshoot.
  + Site-to-Site VPN connections provide connectivity between an on-premises network and an Azure virtual network, using an encrypted tunnel over the public Internet.
  + VPN gateways are virtual network gateways deployed with the gateway type VPN. They are used to terminate site-to-site VPN connections.
  + Site-to-Site VPNs support BGP routing and active-active gateways and connections to enable high availability.
  + A wide variety of physical (and software) devices are supported as the on-premises Site-to-Site VPN endpoint. The device must have an Internet-facing static IPv4 address.
  + A local network connection is an Azure resource used to represent the on-premises VPN device and network in Azure.
  + ExpressRoute connection provides connectivity between an on-premises network an Azure virtual network, using a dedicated connection from a connectivity provider.
  + You can connect to ExpressRoute either via your co-location facility provider, via a point-to-point ethernet connection, or by extending your IPVPN WAN.
  + ExpressRoute provides Microsoft Peering (Connectivity to Azure PaaS endpoints, and other Microsoft services) or Private Peering (Connectivity to Azure virtual networks). The former uses Internet address, and the latter uses Intranet addresses. Azure Public Peering, for Azure PaaS services only, is deprecated for new ExpressRoute circuits.
  + ExpressRoute circuits provide different levels of bandwidth, from 50Mbps to 10Gbps. They also provide redundant connections.
  + ExpressRoute circuits are connected to an Azure virtual network using an ExpressRoute gateway (A virtual network gateway of type ExpressRoute)
  + By default, ExpressRoute provides connectivity to all Microsoft data centers in each geopolitical region. The ExpressRoute Premium Add-On extends coverage to all data centers, globally. It also increases the number of private peering routes and the number of virtual networks, which can be connected to a circuit.

**Skill 5.1 - Monitor Resources by Using Azure Monitor**

* **By developing a proactive monitoring strategy, you will be able to understand the operation of your environment at a component level, including resource health and resource spend. Implementing a robust strategy will help you increase your uptime through proactive notifications, so you can resolve issues before they become problems and optimize your resources for optimal performance, which allows you to increase your ROI with the services you deploy.**
* Areas to consider when creating a monitoring strategy -
  + **Visibility into services and the Azure platform**: This is all about understanding how an application or set of services is performing across the board. You will need to understand what metrics you need to monitor and how those can be acted on in Azure through both alerts and visualizations in dashboards.
  + **Deeper insights into applications**: This is particularly true with service or dependency maps and advanced tracing. You may even use these insights to drive automation and remediations within your environments.
  + **Resource Optimization:** You need to understand which metrics are important to not just the health of your application, but also the effects on users or systems that consume those application. By using the visibility and insights you extract from the Azure platform, you can directly correlate the effects of remediations in your environment.
* **Azure Monitor maximizes the availability and performance of your applications by delivering a comprehensive solution for collecting, analyzing, and acting on telemetry from your cloud and on-premises environments. Helps you understand how your applications are performing and proactively identifies issues affecting them and the resources on which they depend.**
* Azure Monitor landing page provides a jumping off point to configure other more specific monitoring services, such as Application Insights, Network Watcher, Log Analytics, Management solutions and so on.
* **Azure Monitor for Containers is an offering for monitoring your Managed Kubernetes clusters (AKS) and Azure Container Instances (ACI).**
* Azure Monitor for VMs is an offering that provides new capabilities for monitoring your virtual machines and virtual machine scale sets.
* **Azure Monitor helps you track performance, maintain security, and identify trends by ingesting metrics and telemetry from multiple areas, including applications and the operating systems of virtual machines. It also allows you to query resources, subscriptions, tenants, and event custom sources.**
* Log Analytics must be enabled and configured before insights can be extracted or visualizations can be created that are dependent on that data.
* **Comparing Metrics and Logs surfaces some key differentiators -** 
  + **Retention** – Most metrics are retained for 93 days within the Azure service, while logs stored in Log Analytics can be retained for up to 2 years. There are opportunities to do long term retention of metrics by storing metrics in Log Analytics as well.
  + **Properties** – Metrics have a fixed set of properties (or attributes). These are time, type, resource, value, and dimensions (optional). Logs have different properties for each log type and even support rich data types, such as date and time.
  + **Data Availability** – Metrics are gathered over time (like once a minute) and available for immediate query. Logs are often gathered after being triggered by an event and can take time to process before they are available for query. While both offer near real-time query capabilities, metrics will typically be used for fast alerts, and logs used for more complex analysis.
* **Once the data is collected, Azure Monitor provides a single pane of glass, or entry point, to interacting with your metrics and logs. Interactions can include querying and alerting, building visualizations and dashboards, or even automated responses based on telemetry for functionality, such as autoscaling in virtual machines.**
* **Data stored in Log Analytics can also be queried directly through a Log Analytics Workspace, where you will have access to the same query interfaces as you have through Azure Monitor, but you also can make customizations to the configuration of the workspace and access workspace-specific solutions including visualizations and queries.**

**Configure and interpret metrics**

* Metrics are the numerical values that are output by resources and services within Azure. Metrics are available for several Azure resources, but not all resources support metrics currently.
  + In this case, we are only referring to the numerical values that the resources in Azure generates, not the logs or text-based values, such as the value of an event log that can be stored in a storage account or in a Log Analytics Workspace.
* **Metrics includes platform metrics, which are created by Azure resources and made available in Azure Monitor for querying and alerting. You can also query application metrics from Application Insights if the service is enabled and you have instrumented your applications. Regardless of whether that application is hosted on a virtual machine or even in a PaaS service.**
* Virtual Machines in Azure can also push custom metrics to the monitor service using the **Windows Diagnostic extension on Windows servers and with the InfluxData Telegraf Agent on Linux VMs. There is also an opportunity to push custom metrics from other sources through a REST API.**
* **Azure metrics are collected at one-minute intervals (unless otherwise specified) and are identified by a metric name and a namespace (or category). Most of the Azure metrics are retained for 93 days within Azure Monitor, but there are notable exceptions listed below:**
  + **Guest OS Metrics** -
    - Collected through diagnostic extensions and sent to an Azure Storage account.
    - Retention period of 14 days
  + **Guest OS metrics sent to Azure Monitor Metrics**
    - Monitored by Windows diagnostic extensions of the InfluxData Telegraf agent and are routed to an Azure Monitor data sink.
    - Retention period of 93 days.
  + **Guest OS metrics collected by Log Analytics Agents**
    - Collected by the Log Analytics agent and are sent to a Log Analytics workspace.
    - Retention period of 31 days. This retention period can be extended for up to two years.
  + **Application Insights log-based metrics**
    - Log-based metrics that are translated into log queries.
    - Retention period of 90 days.
* For longer term retention, metrics can optionally be sent to Azure storage for select resources and retained up to the configured retention policy or the storage limits of the account. They can also be sent to Log Analytics with a default retention period of 31 days.
* **As metrics are collected, each metric has the following properties:** 
  + **The time the value was collected**
  + **The type of measurement that value represents**
  + **The resource with which the value is associated**
  + **The value itself**
* Metrics can be one dimensional or multidimensional with up to 10 dimensions. A nondimensional metric can be thought of as the metric name, and the value of the metric output is collected by the Monitor service over time. A multidimensional metric (both from an Azure resource or a custom metric) is the metric name and an additional name-value pair with additional data. For example, imagine a storage account with multiple Blob containers where you need to track the consumption of storage by container. A nondimensional metric would provide only the total consumed storage for the Blob service in the storage account where a multidimensional metric would provide the consumption by container as it has the additional data stored in the metric record.
* **Azure Dashboards:** Each chart or visualization that you create in Azure Monitor can also be pinned to an Azure dashboard. You can have multiple dashboards in Azure, and you can even share a dashboard with others in your organization.

**Configure Azure Monitor logs**

* Log Analytics helps you collect, correlate, search, and act on log and performance data generated by operating systems, applications, and Azure services. It gives you operational insights using rich search and visualizations. It gives you operational insights using rich search and visualizations. Log Analytics provides a single pane of glass for interacting with the data from the entire platform and the workloads you host on it including both Linux and Windows servers. Also, Log Analytics can be used with other cloud providers.
* **Log Analytics Workspace is where logs are collected and aggregated. The logs can also be queried and visualized through Log Analytics or through Azure Monitor. A workspace is an Azure resource, meaning that RBAC can be applied for granular access to the service and the data stored within it. This also means that workspaces can be in regions that meet your organization’s regulatory requirements, data isolation, and scope. You can create multiple workspaces in a single subscription.**

**Implement Log Analytics Workspace**

* **A workspace can be created through the Azure portal, Azure PowerShell, the Azure CLI, and Resource Manager templates**
* To configure a workspace, you will need to provide -
  + A name for the workspace
  + The subscription the workspace will be associated with
  + A resource group
  + A location
  + A selection for pricing tier
* **For machines to report telemetry to Log Analytics, they must be running the Azure Log Analytics (OMS) agent. This agent was previously referred to as the Microsoft Monitoring Agent (MMA) or the OMS Linux agent. The agent binds to a workspace to collect the data defined in the workspace settings or in installed solutions.**
* For the agent to send telemetry, you must also ensure that the required ports are available, and the required URIs are added to the approved-list. The agent utilizes port 443 for all outbound communication.
* **The free tier workspace includes 5GB of log storage per month, with per-GB pricing and per-GB charges for additional storage and retention.**

**Configure diagnostic settings**

* In addition to resource logs, there are also tenant-level services, such as Azure Active Directory, which exist outside a subscription from which you might need to collect log data.
* **Diagnostics logs are one type of log data. There is also log data within the Azure Activity Log, and there is log data that can be obtained from virtual machines with the use of diagnostics agents that is separate from diagnostic logs associated with a tenant-level service or an Azure resource.**
* Both resource logs and tenant logs are considered diagnostic logs. Diagnostic logs that you configure for a tenant service, or a resource are separate from the Azure Activity Log and guest telemetry obtained with diagnostic agents.
* **Azure Activity Log surfaces data at the subscription level and can be useful for understanding actions that occur within your environment against the Resource Manager APIs. For example, when a new deployment is submitted, the events associated with that deployment such as the time it was submitted, the resources that were created, and the user that submitted the request are all tracked within the Activity Log. However, at the subscription level, you are missing any resource-level logs. For example, the Activity Log can show when a Network Security Group (or NSG) was created, but it cannot show when an NSG rule was applied to traffic that was subject to the NSG, such as when a port or protocol is blocked. Diagnostic logs provide this functionality.**
* Events in the activity log are retained for 90 days, you can retain them for a longer period by sending them to Azure Storage and/or a Log Analytics Workspace.
* **Diagnostic logs:** Each resource or tenant service on which you enable diagnostic logs will have varying controls (or settings). For example, not all resources support a retention policy in the diagnostic settings, and not all resources support sending metric data to another location.
* Azure Diagnostics agent can also be configured through resource manager templates and the command line tools by specifying a configuration file. For the exam you should be aware of the schema of this configuration and how to apply it using automated tools. You can learn more about the Azure Diagnostics schema at: https://docs.microsoft.com/azure/monitoring-and-diagnostics/azure-diagnostics-schema.

**Query and analyze logs**

* **Azure Monitor stores and surfaces two types of data -**
  + **Metrics** – Numerical values such as performance counters
  + **Logs** – Either numerical data or text

**Create a query**

* After the workspace has been configured, tenant logs, resource logs, and machines have been onboarded, you can begin to analyze and visualize data. To interact with the data in Log Analytics, you use log queries, which are used to:
  + Perform interactive analysis of log data through the Azure portal in Azure Monitor and a Log Analytics Workspace.
  + Build custom alert rules based on the logs in a workspace.
  + Generate visualizations to can be shared through Azure Dashboards.
  + Export custom data sets to Excel or Power BI.
  + Perform automation based on log data with PowerShell or the Azure CLI.
* **The query language used by Log Analytics is called Kusto (KQL). Kusto queries are used to generate read-only requests to process data and return results. This means that the logs stored in Log Analytics are immutable and are only removed from a workspace based on the retention configuration.**
* Kusto queries are case-sensitive. Typically, language keywords are written in lowercase. When using the names of tables and columns in queries, you must ensure you are using the correct case.

**Set up alerts and actions**

* **Azure Monitor brings a unified alerting experience to Azure, with a single pane of glass for interacting with metrics, the Activity Log, Log Analytics, service and resource health, and service-specific insights that provide out-of-the-box dashboards with visualizations and queries for:**
  + **Custom applications with Application Insights**
  + **Virtual Machines**
  + **Storage accounts**
  + **Containers**
  + **Networks**
  + **Key vaults (preview)**
* Alerts have multiple notification options, including:
  + Email
  + SMS
  + Push notifications to the Azure mobile app
  + Voice
* **Integration with automation services.**
* Alerts that are generated within Azure Monitor can invoke Azure Automation runbooks, Logic Apps, Azure Functions, and even generate incidents in third-party IT service Management tools such as ServiceNow.
* **Alerts in Azure Monitor are centered on alert rules. Alert Rules contain the following components:** 
  + **A Target resource (Or resource type)**
  + **Conditional Logic for the alert with criteria based on the available signals for the target resource**
  + **An Action Group, or what should happen when the alert rule condition is met**
  + **A name and description for the alert rule**
* **Alert Rules in Azure Monitor are not the same as alerts. They are the criteria used to evaluate when an alert should be generated. An alert is generated based on the rule, and then the alerts themselves are acted upon separately, even maintaining their own state such as New or Closed.**
* **Action Groups:** Action Groups are separate resources and are independent of the alert rule. This means that the same Action Group can be used across multiple alert rules.
* **In addition to sending email notifications, you can execute the following actions:**
  + **Runbook** – A set of PowerShell code that runs in the Azure Automation Service.
  + **Function Apps** – A Function App is a set of code that runs “serverless” that can respond to alerts. This functionality requires Version 2 of Function Apps, and the value of the *AzureWebJobsSecretStorageType* app setting must be set to *files*.
  + **ITSM** – You may have up to 10 IT Service Manager (ITSM) actions with an ITSM connection. The following ITSM providers are currently supported: ServiceNow, System Center Service Manager, Provance, and Cherwell.
  + **Logic Apps** – Provides a visual designer to model and automate your process as a series of steps known as a workflow. There are many connectors across the cloud and on-premises to quickly integrate across services and protocols. When an alert is triggered the Logic App can take the notification data and use it with any of the connectors to remediate the alert or start other services.
  + **Webhook** - A webhook allows you to route an Azure alert notification to other systems for post-processing or custom actions. For example, you can use a webhook on an alert to route it to services that send text messages, log bugs, notify a team via chat/messaging services, or do any number of other actions.

**View alerts in Azure Monitor**

* After an alert rule has been created, the alert rule and Action Group can be managed through Azure Monitor from the Alerts blade by selecting Manage Alert Rules. Alerts can be managed across multiple subscriptions and can be filtered by Resource Group, Resource Type, Signal Type, and Status
* **Alerts and Metric alerts from Alert Rules do not generate alerts immediately and can take up to 10 minutes. When alerts are generated, they will be distributed based on the actions defined in the Action Group.**
* You can enable and disable alert rules as needed to meet your requirements.
* **Alert can have one of three states**
  + **New** – The alert is new and has not been reviewed
  + **Acknowledged** – The issues that generated the alert is being actioned by an administrator
  + **Closed** – The issue that generated the alert has been resolved, and the alert has been marked as closed.
* **Alert State:** Alert State is not the same as the monitor condition of an alert. When the Azure platform generates an alert based on an alert rule, the alert’s monitor condition is set to *fired* and when the underlying condition clears, the monitor condition is set to *resolved*.
* **As alerts are generated, they will appear on the Alerts blade in Azure Monitor. From the Alerts blade, can you view alerts for all subscriptions, and drill into one or more specific Subscriptions, Resource Groups, and Resources. Also, you can filter by Time Range by choosing Past Hour, Past 24 Hours, Past 7 Days, or Past 30 Days from the drop-down menu**

**Configure Application Insights**

* Application Insights is used for development and as a production monitoring solution. It works by installing a package into your app, which can provide a more internal view of what’s going on with your code. Its data includes response times of dependencies, exception traces, debugging snapshots, and execution profiles. It provides powerful smart tools for analyzing all this telemetry both to help you debug an app, and to help you understand what users are doing with it. You can tell whether a spike in response times is caused by something in an app or an external resourcing issue. Application Insights provides significantly more value when your application is instrumented to emit custom events and exception information.
* **Application Insights has an extensive dashboard depicting all the aspects of your application workload. The dashboard displays application performance, usage, diagnostic, and other app data. The dashboard can be customized based on your preferences.**

**Skill 5.2 - Implement Backup and Recovery**

* Azure Backup is a service that allows you to backup on-premises servers, cloud-based virtual machines and virtualized workloads. Also supports the backup of Azure Storage file shares.

**Create a Recovery Services Vault**

* **Within Azure, a single resource is provisioned for either Azure Backup or Azure Site Recovery. This resource is called a Recovery Services Vault. It is also the resource that is used for configuration and management of both Backup and Site Recovery.**

**Use Soft Delete to recover Azure VMs**

* The default behavior of deleting a backup is that the backup is deleted and lost forever. Soft Delete is a feature that allows you to save and recover your data when backup data are deleted even in the event of an overwrite. This feature must be enabled in Recovery Services Vault by choosing Properties -> Security Settings. **When you use Soft Delete, backup data is retained for 14 days after deletion.**

**Create and configure Backup Policy**

* **To view your current backup policies in the Azure portal, open the Recovery Services Vault blade, and then click backup policies.**
* You can create four different types of policies from this view in Azure Portal
  + **Azure Virtual Machines** – Allows you to specify the backup frequency, retention period, and the backup point on a weekly, monthly, and yearly schedule.
  + **SAP HANA In Azure VM** – Allows you to use SAP HANA specific backup technology such as **full, differential, and log backup with an associated schedule for each option**.
  + **Azure File Share** – Allows you to schedule a daily backup for an Azure file share.
  + **SQL Server in Azure VM** – Allows you to use SQL server-specific backup technology, such as **full, differential, and log backups**, with an associated schedule for each option. Also, you can enable SQL backup **compression**.

**Define backup policies**

* **An Azure Backup policy defines how often backups occur and how long the backups are retained. The default policy accomplishes a daily backup at 05:30pm UTC and retains backups for 30 days and you can define custom Backup policies.**

**Perform backup and restore operations by using Azure Backup Service**

* Azure Backup service can be used to backup and restore various cloud as well as on-prem resources. Recovery Services Vault is used to enable Azure Backup and to configure the backup policies.
* **When you back up an Azure virtual machine, you can restore an entire virtual machine or you can restore individual files from the virtual machine and it is quite easy to set up.**
* When protecting IaaS VMs by using Azure Backup, only VMs in the same region as the Recovery Services Vault are available for backup. Because of this, it is a best practice to choose Geo-Redundant storage or Read Access Geo-Redundant storage to be associated with the Recovery Services Vault. This ensures that if a regional outage affects VM access, there is a replicated copy of the backup in another region.
* **To restore a virtual machine that has encrypted disks, you also need to provide the Azure Backup Service Access to the Key Vault holding the key.**

**Backup and restore with on-premises workloads**

* **Azure Backup Server is a standalone service that you install on a Windows Server operating system that stores the backed-up data in a Recovery Services Vault. Azure Backup Server inherits much of its functionality from Data Protection Manager (DPM). It does not backup to tape, and it does not integrate with system center. You should consider using Azure Backup server when you need to back up the following supported workloads:**
  + **Windows Client**
  + **Windows Server**
  + **Linux Servers**
  + **VMWare VMs**
  + **Exchange**
  + **SharePoint**
  + **SQL Server**
  + **System State and Bare Metal Recovery**
* Microsoft Azure Recovery Services (MARS) Agent can be used to backup files and folders from on-premises VM. The agent is only supported on Windows agents, you will need to download the vault credentials file which is under the Recovery Services agent download link. The vault **credentials file is needed during the installation of the MARS agent. Its only valid for 48 hours** from the time of download, so be sure to obtain them only when you are ready to install the MARS agent.
* **Data protected by Azure Backup in encrypted using the supplied passphrase. If the passphrase is lost or forgotten, any data protected by Azure Backup is not able to be recovered and is lost.**

**Perform site-to-site recovery by using Azure Site Recovery**

* **Azure Site Recovery Services enables us to replicate, failover, and failback virtual machines as needed. Azure Site Recovery solutions allows us to address the major scenarios below:**
  + **Azure VMs from one region to another**
  + **On-Premises VMs (VMWare, Hyper-V, and physical servers) to Azure**
  + **On-Premises VMs to another site**

**Configure and review backup reports**

* Azure Backup Reports provide data visualization across your Recovery Services Vaults and Azure subscriptions to provide insight into your backup activity. This reporting solution is currently widely supported for Azure virtual machine backup and file and folder backup scenarios when using the MARS (Microsoft Azure Recovery Services) agent.
* **In order to configure the backup reports, you need to create or use an existing Log Analytics Workspace to store the backup reporting data. Also, you need a Recovery Services Vault, which records all the backup operations as diagnostic data.**
* Azure Site Recovery is used to replicate physical and Azure VMs to a secondary Azure location for backup
* Azure Backup vaults support backup of Disks and Blobs.

**Chapter Summary -**

* Below are some of the key takeaways from this chapter:
* Azure Monitor is a single pane of glass for accessing Azure metrics, tenant and resource diagnostic logs, Log Analytics, service health, and alerts.
* You can configure alerts based on metric alerts (captured from Azure Metrics) to Activity Log alerts that can notify by email, webhook, SMS, Logic Apps, or even an Azure Automation Runbook.
* Azure Log Analytics can consolidate machine data from on-premises and cloud-based workloads and this data is indexed and categorized for quick searching. Data can be collected from both Windows and Linux machines.
* Azure Log Analytics has many management solutions that help administrators gain value out of complex machine data. These solutions contain pre-built visualizations and queries that help surface insights quickly.
* Queries in Log Analytics can be saved for quick access and visualized and shared using Azure Dashboards. To analyze data outside of Log Analytics you can export the data to Excel and Power BI.
* Recovery Services Vault is used for configuration and management of both Backup and Site Recovery.
* An Azure Backup policy defines how often backups occur and how long the backups are retained.
* The Azure Backup service can backup and restore and entire virtual machine and you can also use it for just file recovery to restore files from a recovery point without recreating the entire virtual machine.
* Azure Backup can be used to protect files and folders, applications, and IaaS virtual machines. This cloud-based data protection service helps organizations by providing offsite backups of on-premises servers and protection of VM workloads they have already moved to the cloud.
* The backup data is retained for 14 days after deletion by using Soft delete feature.
* Azure Site Recovery service enables us to replicate, failover, and failback virtual machines as needed.