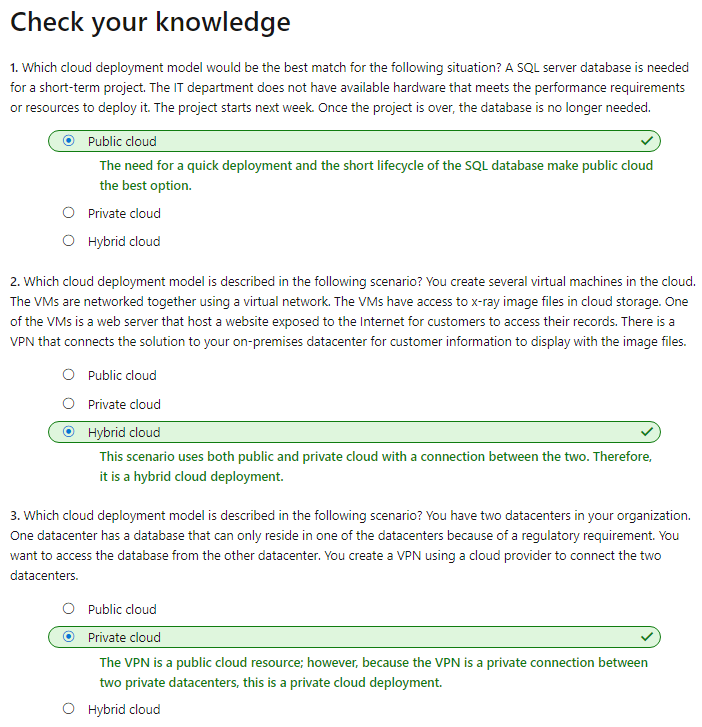
**Manage resources in Azure**

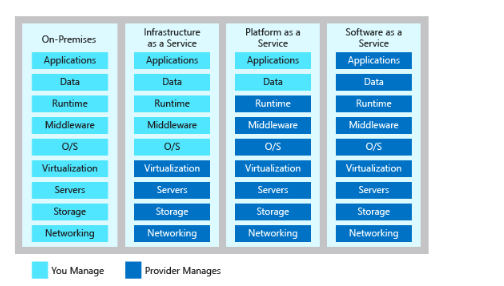
**Align requirements with cloud types and service models in Azure**

1. Azure solutions for public, private, and hybrid cloud
   1. What is cloud computing?
      1. Cloud computing is the provisioning of services and applications on demand over the internet
      2. To the user, the details of the services are abstracted
      3. Cloud computing uses virtualization
      4. There are three deployment models for cloud computing: public cloud, private cloud, and hybrid cloud
2. Public cloud
   1. Public clouds are the most common way of deploying cloud computing
   2. Available over the internet
   3. You can automatically scale up to meet demand at peak enrolment
   4. When site traffic is low, your site can scale down to save costs
   5. Your site is responsive at peak demand, and you only pay for more resources when needed
   6. Why public cloud?
      1. Public clouds can be deployed faster than on-premises infrastructures
      2. Every employee of a company can use the same application
      3. Examples of why you would use public cloud:
         1. Service consumption through on-demand or subscription model
         2. No up-front investment of hardware
         3. Automation
         4. Geographic dispersity
         5. Reduced hardware maintenance
   7. Developers can use the public cloud to quickly provision virtual machines for sandboxed environments to develop a solution
3. Private cloud
   1. A private cloud consists of computing resources used exclusively by users from one business or organization
   2. Physically located at your organization’s on-site datacentre, or it can be hosted by a third-party service provider
   3. The organization is responsible for the purchase, configuration, and maintenance of the hardware
   4. Used when data needs to remain at the other site because of regulatory compliance
   5. Why private cloud?
      1. Can provide more flexibility
      2. High levels of control and security
      3. Examples of why you would use private cloud:
         1. **Pre-existing environment**: For example an OS that can’t be replicated in the cloud
         2. **Legacy applications**
         3. **Data sovereignty and security**
         4. **Regulatory compliance / certification**
4. Hybrid cloud
   1. Combines a public cloud and a private cloud by allowing data and applications to be shared between them
   2. Hybrid cloud computing gives businesses the ability to seamlessly scale their on-premises infrastructure up to the public cloud to handle any overflow
   3. Using a hybrid cloud helps eliminate the need to make up-front capital expenditures to handle short-term spikes
   4. Flexibility to manage which resources are local versus resources in the cloud
   5. For example when a website need to be responsive and public facing, but the data needs to remain at the physical location
   6. Why hybrid cloud?
      1. **Existing hardware investment**
      2. **Regulatory requirements**
      3. **Unique operating environment**
      4. **Migration**: Move to the cloud over time



**Cloud service models**

1. Cloud computing resources are delivered using three different service models:
   1. IaaS
   2. Paas
   3. SaaS



**Infrastructure as a service**

1. Enables you to quickly scale resources to meet demand and only pay for what you use
2. Avoids the expense and complexity of buying and managing your own physical servers
3. You don't need to manage physical servers and appliances
4. Common Scenarios:
   1. Developers need to test work on many different environments
   2. Need control over **Web** **Hosting**
   3. Infrastructure to support **Web** **Apps**
   4. **Storage**, **backup** and **recovery**
   5. If you have a workload that requires **high**-**performance** **computing**
   6. **Big data analysis**
5. Advantages:
   1. Eliminates capital expense and reduces ongoing cost
   2. Improves business continuity and disaster recovery
   3. Respond quicker to shifting business condition
   4. Increase stability, reliability, and supportability

**Platform as a service**

1. Is a complete development and deployment environment in the cloud
2. Build and deploy everything from simple cloud-based apps to sophisticated, cloud-enabled enterprise applications
3. You purchase the resources from a cloud service provider on a pay-as-you-go basis
4. Like IaaS, PaaS includes infrastructure such as servers, storage, and networking
5. Common scenarios
   1. Using PaaS, your developers have the option to create a web app
   2. Details such as creating a virtual machine, installing a web server, and installing middleware are abstracted away
   3. A SQL database can be quickly provisioned
   4. PaaS provides a framework that developers can build upon to develop or customize cloud-based applications
   5. Analysis tools provided as a service allows you to analyse and mine data
6. Advantages:
   1. Reduced development time
   2. Develop for multiple platforms
   3. Use sophisticated tools affordably
   4. Support geographically distributed development teams
   5. Efficiently manage the application lifecycle

**Software as a service**

1. Common examples are email, calendaring, and office tools such as Microsoft 365
2. You can rent the use of an application for your organization
3. Your users connect to the service over the Internet, usually with a web browser
4. Outlook, Hotmail, or Yahoo! Mail are examples
5. Common scenarios
   1. SaaS CRM can be used to quickly implement a CRM solution
   2. you can rent productivity apps, such as email, collaboration, and calendaring, even ERP systems
6. Advantages
   1. **Gain access to sophisticated applications**. CRM, ERP etc.
   2. **Use free client software**: Users can run apps through Web Browser
   3. **Access** **app** **data** **from** **anywhere**

**What is the Azure CLI?**

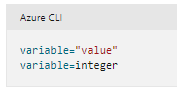
1. The Azure CLI is a command-line program to connect to Azure and execute administrative commands
2. It runs on Linux, macOS, and Windows
3. For example, to restart a virtual machine (VM), you would use a command like the following:

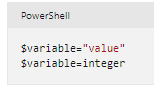


1. The Azure CLI can also be used from a browser through the Azure Cloud Shell

**How to install the Azure CLI**

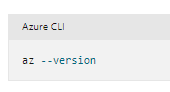
1. On both Linux and macOS, you use a package manager to install the Azure CLI:
   1. Linux: **apt**-**get** on Ubuntu, **yum** on Red Hat, and **zypper** on OpenSUSE
   2. Mac: **Homebrew**
2. Using the Azure CLI in scripts
   1. Set variables in a **Bash** **shell**:



* 1. **Powershell** set variables:
  2. 

**Exercise - Install and run the Azure CLI**

1. To check the azure CLI version:

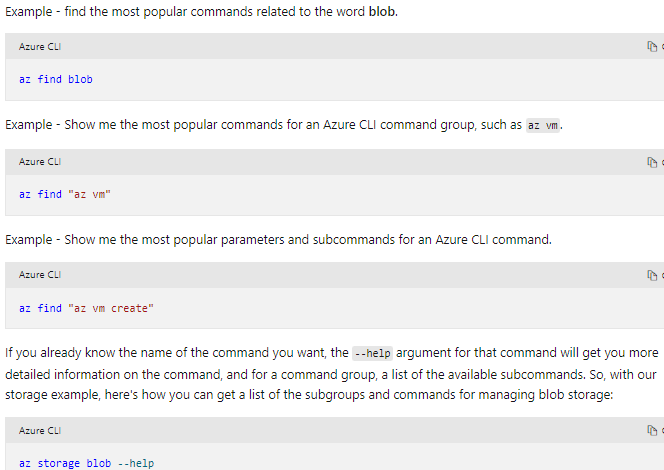


**Work with the Azure CLI**

1. The Azure CLI lets you type commands and execute them immediately

**What Azure resources can be managed using the Azure CLI?**

1. You can work with **resource** **groups**, **storage**, **virtual** **machines**, **Azure** **Active** **Directory** (**Azure** **AD**), **containers**, **machine** **learning**, and so on
2. Commands in the CLI are structured in **groups** and **subgroups**
3. Storage group contains subgroups including **account**, **blob**, and **queue**



**How to create an Azure resource**

1. When creating a new Azure resource, there are typically three steps: **connect** **to** **your** **Azure** **subscription**, create the **resource**, and verify that creation was **successful**



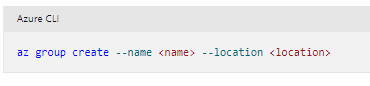
**Connect**

1. Authenticate Azure commands:



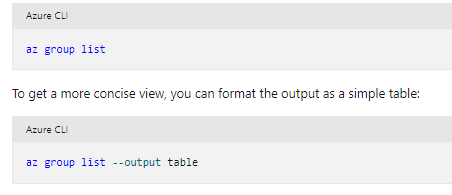
**Create**

1. The Azure CLI group create command creates a resource group
2. Example:



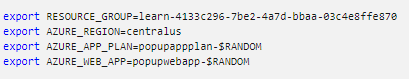
**Verify**

1. For example, the Azure CLI group list command lists your Azure resource groups

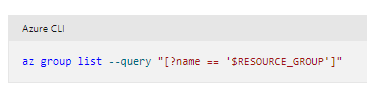


**Exercise - Create an Azure website using the CLI**

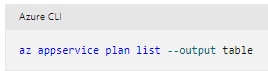
1. Commands:





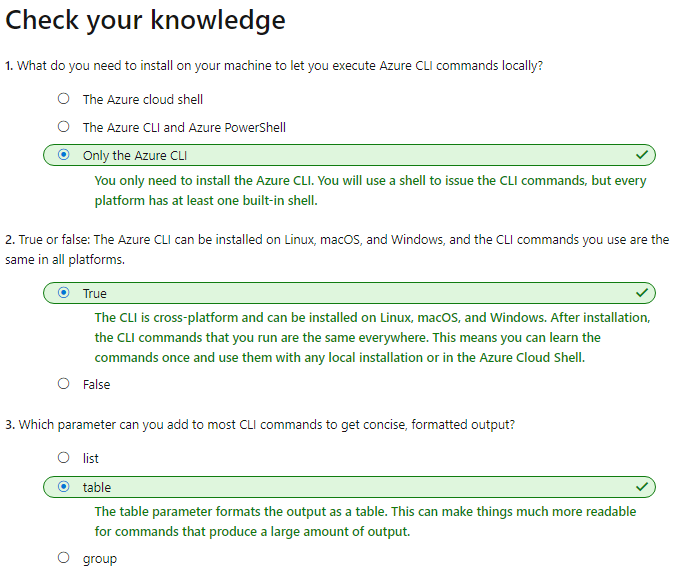


**az appservice plan create --name $AZURE\_APP\_PLAN --resource-group $RESOURCE\_GROUP --location $AZURE\_REGION --sku FREE**





**az webapp deployment source config --name $AZURE\_WEB\_APP --resource-group $RESOURCE\_GROUP --repo-url "https://github.com/Azure-Samples/php-docs-hello-world" --branch master --manual-integration**



**Decide if Azure PowerShell is right for your tasks**

1. What tools are available?
   1. **Azure** **Portal**
   2. **Azure** **CLI**
   3. **Azure** **Powershell**
   4. All three are cross-platform, running on Windows, macOS, and Linux
2. What is the Azure Portal
   1. Azure portal is a website that lets you create, configure, and alter the resources in your Azure subscription
   2. Is a Graphical User Interface (GUI) that makes it convenient to locate the resource you need
   3. The portal does not provide any way to automate repetitive tasks
3. What is the Azure CLI?
   1. The Azure CLI is a cross-platform command-line program to connect to Azure and execute administrative commands
   2. For example to create a VM:



* 1. The Azure CLI is available two ways: inside a **browser** via the Azure Cloud Shell or with a **local** **install** on **Linux**, **Mac**, or **Windows**
  2. First launch a shell such as **cmd**.**exe** on Windows or Bash on Linux or macOS
  3. To automate repetitive tasks, you assemble the commands into a shell script

1. What is Azure PowerShell?
   1. Azure PowerShell is a module that you add to **Windows** **PowerShell** or **PowerShell** **Core** to let you **connect** to your **Azure** **subscription** and manage resources
   2. Azure PowerShell provides the **New**-**AzVM** command that creates a virtual machine



* 1. Azure PowerShell is also available **two** **ways**: inside a **browser** via the **Azure** **Cloud** **Shell** or with a **local** install on **Linux**, **Mac**, or **Windows**

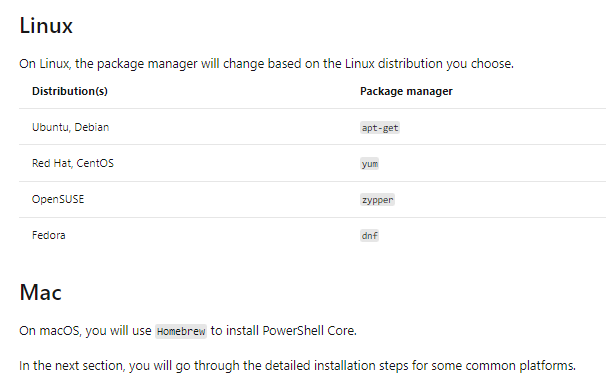
1. How to Choose an administrative tool
   1. **Automation**: Do you need to automate a set of complex or repetitive tasks?
   2. **Learning** **Curve**: Do you need to complete a task quickly without learning new commands or syntax?
   3. **Team** **Skillset**: Does your team have existing expertise?
2. Example:
   1. If you need to repeat tasks repeatably, the Azure Portal is not a good choice. Rather use Azure CLI or Azure Powershell
   2. Azure PowerShell is available on the operating systems your admin team uses, it supports automation, and should be **quick** for your team to learn

**Install PowerShell**

1. What must be installed:
   1. **The base PowerShell product**: This comes in two variants: PowerShell on Windows, and PowerShell Core on macOS and Linux
   2. **The Azure PowerShell module:** This extra module must be installed to add the Azure-specific commands to PowerShell

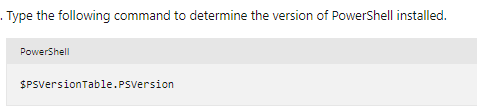
**How to install PowerShell Core**

1. On both Linux and macOS, you use a package manager to install PowerShell Core



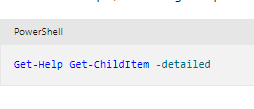
**Exercise - Install Azure PowerShell**

1. Windows
   1. PowerShell is included with Windows, however there may be an update available for your machine



**Create an Azure Resource using scripts in Azure PowerShell**

1. PowerShell lets you write commands and execute them immediately. This is known as **interactive** **mode**
2. What are PowerShell cmdlets?
   1. A PowerShell command is called a **cmdlet**
   2. The term cmdlet is intended to imply "small command"
   3. Cmdlets follow a verb-noun naming convention; for example, **Get**-**Process**, **Format**-**Table**, and **Start**-**Service**



**What is a PowerShell module?**

1. Cmdlets are shipped in modules
2. A PowerShell Module is a DLL
3. You can get a list of loaded modules using the Get-Module command:

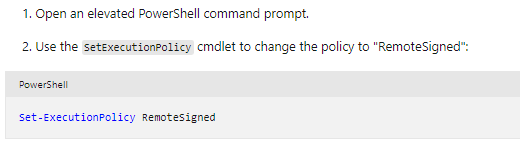


**What is the Az module?**

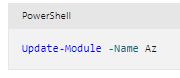
1. **Az** is the formal name for the Azure PowerShell module

**Install the Az module**

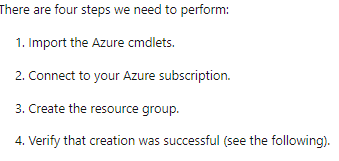
1. The Az module is available from a global repository called the PowerShell Gallery
2. Install the module onto your local machine through the Install-Module

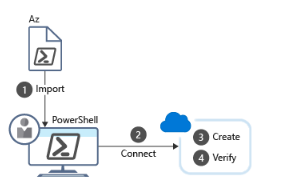


**Update a module**



**Example: How to create a resource group with Azure PowerShell**





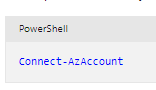
**Import the Azure cmdlets**

1. At startup, PowerShell loads only the core cmdlets by default
2. You use the Import-Module cmdlet to load modules



**Connect**

1. With a local install of Azure PowerShell, you will need to authenticate before you can execute Azure commands
2. **Connect**-**AzAccount** cmdlet prompts for your Azure credentials

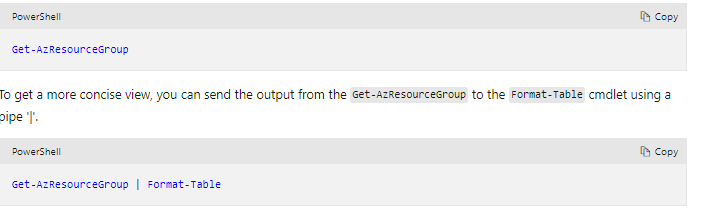


**Work with subscriptions**

1. You can only be in one subscription at a time
2. Use the Get-AzContext cmdlet to determine which subscription is active

**Get a list of all resource groups**

1. You can retrieve a list of all Resource Groups in the active subscription:



**Create a resource group**

1. You can create resource groups with the New-AzResourceGroup cmdlet
2. You must specify a name and location
3. The name must be unique within your subscription



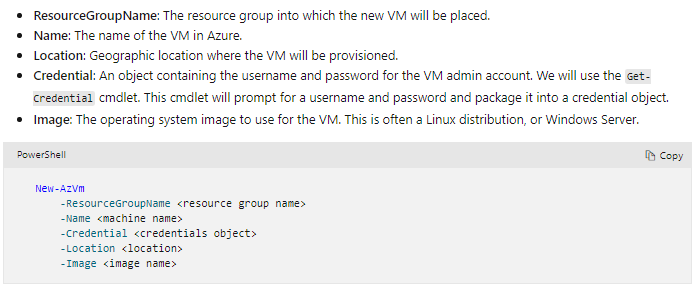
**Verify the resources**

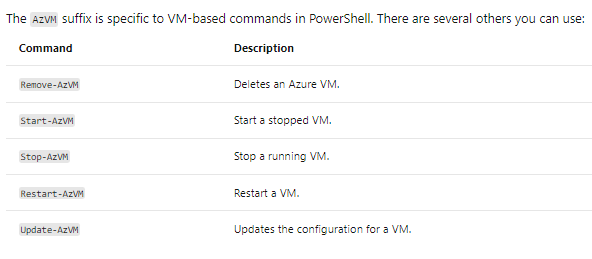
1. The Get-AzResource lists your Azure resources
2. You can get a more concise view through the Format-Table cmdlet. Here we will use a shorthand version ft:



**Create an Azure Virtual Machine**

1. Azure PowerShell provides the **New**-**AzVm** cmdlet to create a virtual machine
2. We only need to specify five things:

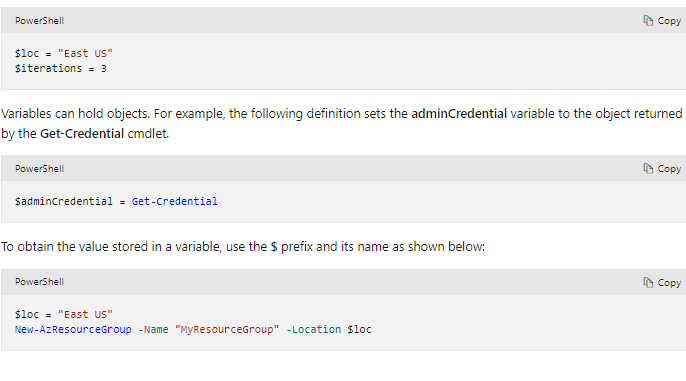


****

**Exercise - Create an Azure Resource using scripts in Azure PowerShell**

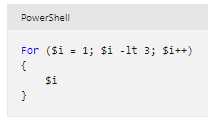
**Create and save scripts in Azure PowerShell**

1. What is a PowerShell script?
   1. A **PowerShell** **script** is a text file containing commands and control constructs
   2. The commands are invocations of cmdlets
   3. PowerShell script files have a .**ps1** file extension
2. PowerShell techniques
   1. PowerShell has many features found in typical programming languages
   2. You can define **variables**, use **branches** and **loops**, capture **command**-**line** **parameters**, write **functions**, add **comments**, and so on
3. Variables



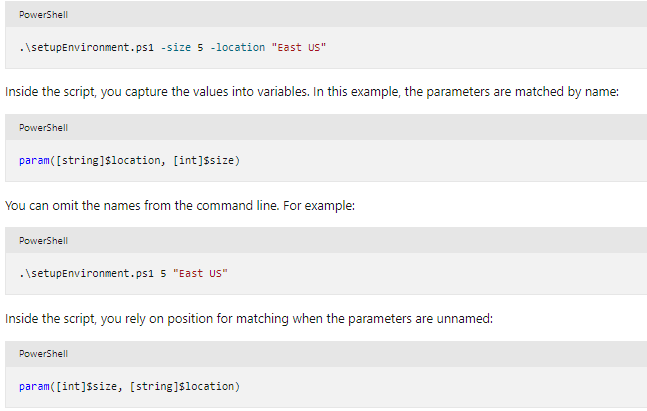
**Loops**

1. PowerShell has several loops: **For**, **Do**...**While**, **For**...**Each**, and so on
2. The For loop is the best match for our needs because we will execute a cmdlet a fixed number of time
3. The comparison operators are written -**lt** for "less than", -**le** for "less than or equal", -eq for "equal", -**ne** for "not equal”



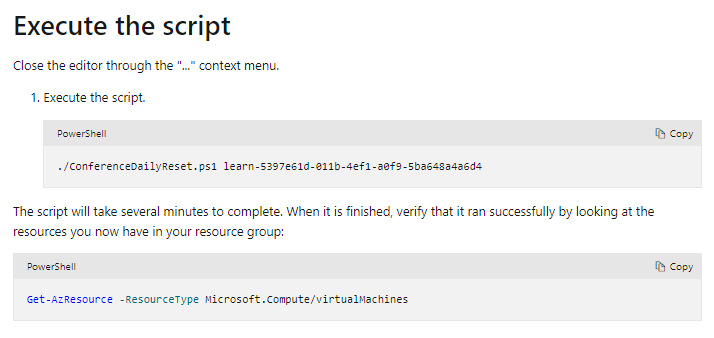
**Parameters**

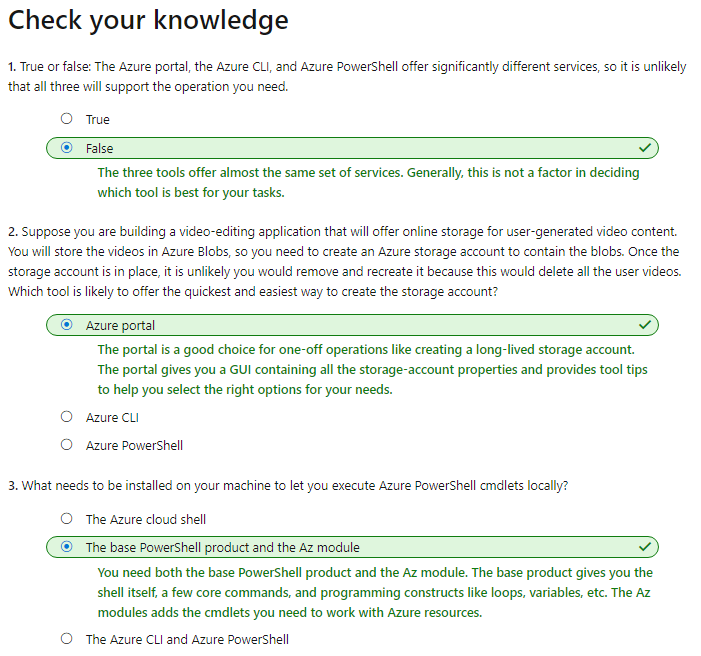
1. When you execute a script, you can pass arguments on the command line



**Exercise - Create and save scripts in Azure PowerShell**







**Compare costs by using the Total Cost of Ownership Calculator (SPEED READING)**

1. The TCO Calculator helps you estimate the cost savings of operating your solution
2. Costs can include electricity, network maintenance, and IT labour

**How does the TCO Calculator work?**



**Step 1: Define your workloads**

1. Get specifications based on: Servers, databases, storage and Networking

**Step 2: Adjust assumptions**

1. Check current on-premises licenses
2. Costs according to:
   1. Electricity price per kilowatt hour (KWh)
   2. Hourly pay rate for IT administration
   3. Network maintenance cost
3. To improve the accuracy of the TCO Calculator results, you adjust the values so that they match the costs of your current on-premises infrastructure

**Step 3: View the report**

1. Choose a time frame between one and five years

# Purchase Azure services

**What types of Azure subscriptions can I use?**

1. Azure offers:
   1. Free trial
   2. Pay as you go
   3. Member offers: Existing membership provides credits

**How do I purchase Azure services?**

1. There are three main ways to purchase services on Azure. They are:
   1. Through an Enterprise Agreement
   2. Directly from the web
   3. Through a Cloud Solution Provider

**What factors affect cost?**

1. **Resource** **type**: For example, with a **storage** **account** you specify a type (such as **block** **blob** **storage** or **table** **storage**), a **performance** **tier** (**standard** or **premium**), and an **access** **tier** (**hot**, **cool**, or **archive**)

**Usage meters**

1. Meters include:
   1. Overall CPU time
   2. Time spent with a public IP address
   3. Incoming (ingress) and outgoing (egress) network traffic
   4. Disk size and amount of disk read and disk write operations
2. Resource usage
   1. In Azure, you're always charged based on what you use
   2. Deallocating a VM means that the VM is no longer running. But the associated hard disks and data are still kept in Azure
   3. You're still billed for disk storage

**Azure subscription types**

1. Azure free trial subscription provides access to a number of Azure products that are free for 12 months
2. Also include usage allowances, which affect costs

**Azure Marketplace**

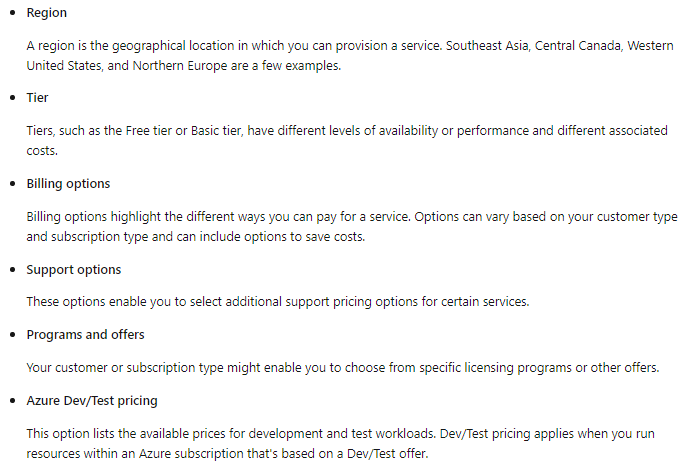
1. Examples include managed network firewall appliances or connectors to third-party backup services

**Does location or network traffic affect cost?**

1. Location
   1. Azure infrastructure is distributed globally, deployed closest to customers
   2. Different regions can have different associated prices
2. Zones for billing of network traffic
   1. Some inbound data transfers (data going into Azure datacenters) are free
   2. Zones include:



1. The Pricing calculator displays Azure products in categories
2. Keep in mind that the Pricing calculator provides estimates and not actual price quotes
3. The options that you can configure in the Pricing calculator vary between products, but they can include:



**Manage and minimize total cost on Azure**

1. Apply tags to identify cost owners
   1. Tags help you manage costs associated with the different groups of Azure products and resources
   2. You can use tags to categorize costs by department

**Resize underutilized virtual machines**

1. A common recommendation is to resize or shut down VMs that are underutilized or idle
2. Keep in mind that resizing a VM requires it to be stopped, resized, and then restarted

**Deallocate virtual machines during off hours**

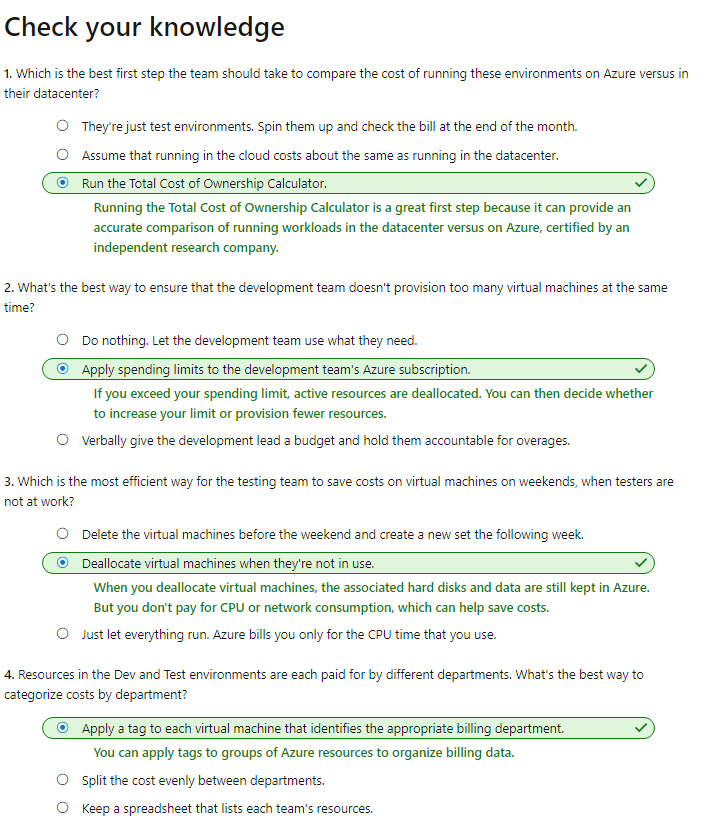
1. Recall that to deallocate a VM means to no longer run the VM, but preserve the associated hard disks
2. This approach is an excellent strategy for development and testing environments

**Migrate from IaaS to PaaS services**

1. As you move your workloads to the cloud, a natural evolution is to start with infrastructure as a service (IaaS)
2. One way to reduce costs is to gradually move IaaS workloads to run on platform as a service (PaaS) services
3. PaaS services such as Azure SQL Database often less expensive

**Save on licensing costs**

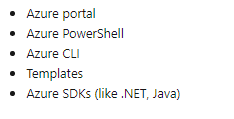
1. Choose cost-effective operating systems
2. Use Azure Hybrid Benefit to repurpose software licenses on Azure



# Principles of resource groups

1. What are resource groups?
   1. All resources must be in a resource group and a resource can only be a member of a single resource group
2. Logical grouping
   1. If you delete a resource group, all resources contained within are also deleted
3. Authorization
   1. By applying RBAC permissions to a resource group, you can ease administration

**Create a Resource Group**

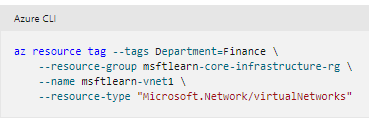


**Use resource groups for organization**

1. Consistent naming convention
   1. You can start with using an understandable naming convention
2. Organizing principles
   1. For example, put all **virtual** **networks** in one **resource** **group**, all **virtual** **machines** in another **resource** **group**, and all **Azure** **Cosmos** **DB** instances in yet another **resource** **group**
3. Organizing for authorization
   1. RBAC, you can organize resources by who needs to administer them
   2. For example, Azure SQL Database instances, putting them in the same resource group would simplify administration
4. Organizing for life cycle
   1. If you delete a resource group, you delete all the resources in it
5. Organizing for billing
   1. Placing resources in the same resource group is a way to group them for usage in billing reports

# Use tagging to organize resources

1. What are tags?
   1. Tags are name/value pairs of text data that you can apply to resources and resource groups
   2. A resource can have up to 50 tags
   3. Tags can be added and manipulated through the **Azure** **portal**, **Azure** **CLI**, **Azure** **PowerShell**, **Resource** **Manager** **templates**, and through the **REST** **API**
   4. Example:



**Use tags for organization**

1. You can use tags to group your billing data
2. You can retrieve all the resources in your subscription that have a specific tag name or value
3. Tagging resources can also help in monitoring to track down impacted resources
4. It's also common for tags to be used in automation
5. If you want to automate the shutdown and startup of virtual machines in development environments during off-hours to save costs, you can use tags to support automation

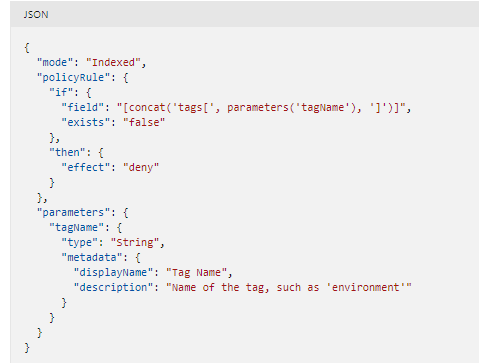
# Use policies to enforce standards

1. How do you ensure that new resources follow the rules – Azure Policy

**What is Azure Policy?**

1. Azure Policy is a service you can use to create, assign, and manage policies
2. Apply and enforce rules that your resources need to follow

**Create the policy definition**



**Create a policy assignment**

1. To enable the policy, you need to create an assignment

# Secure resources with role-based access control

1. Implementing Azure Policy ensured that all our employees with Azure access are following our internal standards for creating resources
2. How do we protect those resources once they are deployed?
3. RBAC provides fine-grained access management for Azure resources
4. Using RBAC, you can:
   1. Allow one user to manage VMs in a subscription
   2. Allow a database administrator (DBA) group to manage SQL databases
   3. Allow a user to manage all resources in a resource group, such as VMs, websites, and virtual subnets
   4. Allow an application to access all resources in a resource group
5. To view access permissions, use the **Access** **control** (**IAM**) panel

**How RBAC defines access**

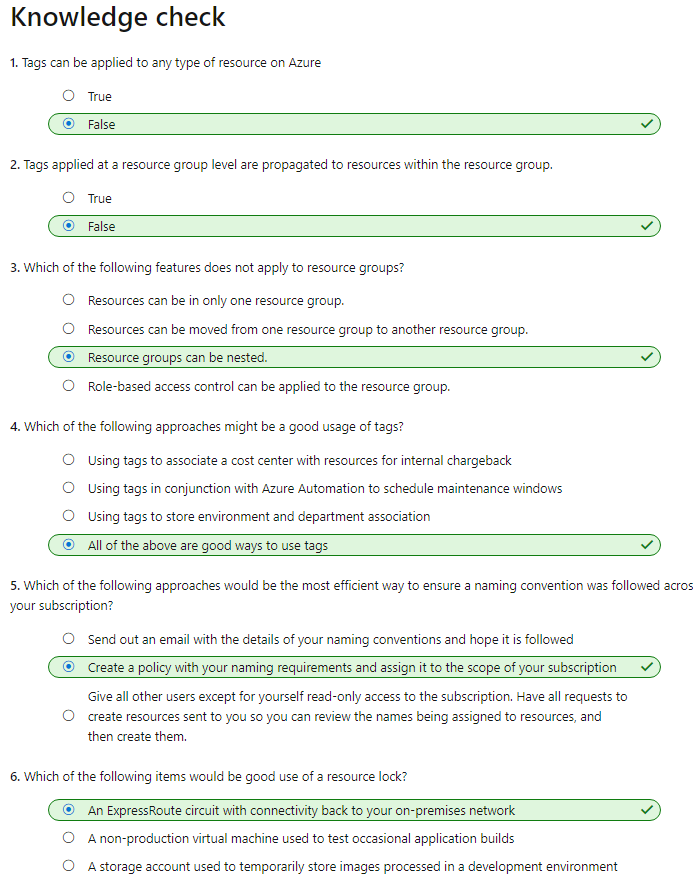
1. **RBAC** uses an allow model for access
2. RBAC allows you to perform specific actions, such as read, write, or delete
3. Can have read and write permissions on a resource group

**Best Practices for RBAC**

1. Here are some best practices:
   1. Segregate duties within your team
   2. When planning your access control strategy, grant users the lowest privilege level
   3. Use Resource Locks to ensure critical resources aren't modified or deleted

**Use resource locks to protect resources**

1. What are resource locks?
   1. **Delete** will allow all operations against the resource but **block** the ability to **delete** it
   2. **Read**-**only** will **only** allow **read** **activities** to be performed against it
   3. When a resource lock is applied, you must first remove the lock in order to perform that activity



# Identify incorrectly assigned resources in Azure

# Organize Azure resources into resource groups

# Make sure that all the resources in your resource group have a single lifecycle

# You can move resources from one resource group to another resource group in a different subscription

# Use a consistent naming convention

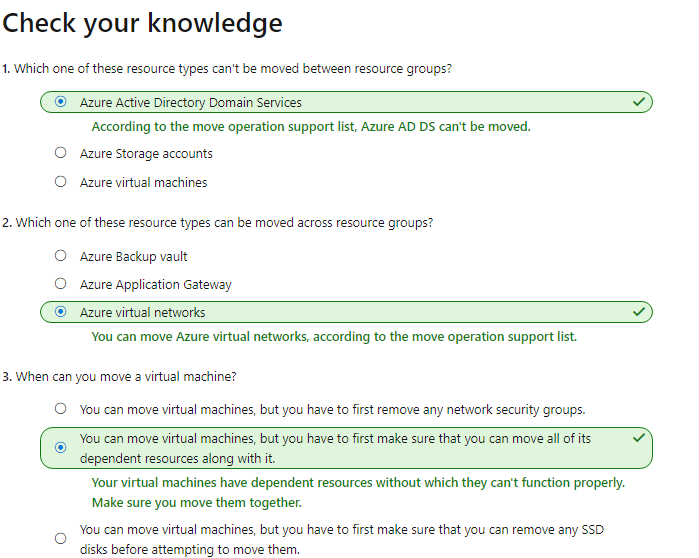
# To help identify and organize your resource groups, use a consistent naming convention

# Assess resources that can move

# 

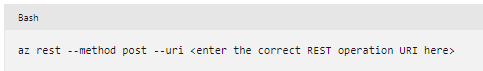
**Check the limitations on resource types**

1. For example, these resources can be moved:
   1. Azure Storage accounts
   2. Azure Virtual Machines
   3. Azure Virtual Networks
2. For example, these resources **can** be moved:
   1. Azure Storage accounts
   2. Azure virtual machines
   3. Azure virtual networks
3. These resources **can't** be moved:
   1. Azure Active Directory domain services
   2. Azure Backup vaults
   3. Azure App Service gateways
4. Virtual machines have their own limitations you must keep in mind. Here's a summary of limitations for virtual machines:
   1. If you want to move a virtual machine, all of its dependants must go with it
   2. You can't move virtual machines with certificates
   3. You can't move virtual machine scale sets
   4. You can't move any managed disks

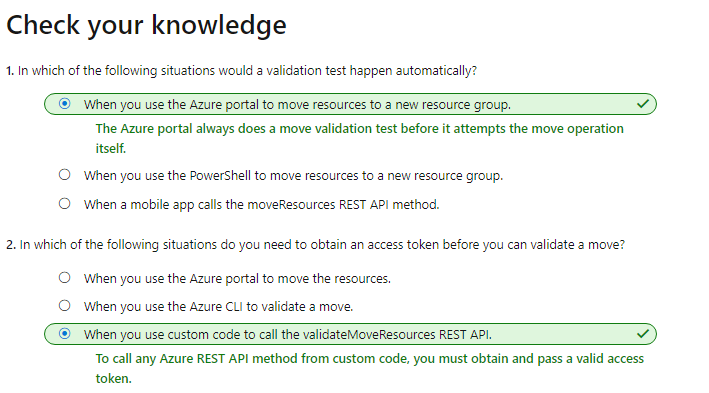


# Validate resources in Azure

1. Prepare to test your move
   1. Before attempting to move a resource, you can test whether it will be successful by calling the **validate** **move** operation from the Azure REST API.
   2. If you're trying to move resources through the Azure portal, you don't need to validate
   3. Validate Move example:



1. To formulate the correct REST URI to call, you must have:
   1. Your Azure subscription ID
   2. Name of the resource group
   3. Resource ID of each resource
   4. Resource ID of the destination resource group
   5. Account access token



# Identify steps to move resources between Azure resource groups

# When you start a move operation, the resource group holding your resources and the new destination resource group are locked

# You can't do write or delete operations on the resource groups until the move operation ends

# You’re moved resources don't change location. For example, if you have a storage account in the East US region, and you move it to another resource group, it keeps its East US region location

**Move resources between subscriptions**

# Depending on the resource type, you can move your resources between subscriptions

# 

# ORDER:

# Create resource group

# Get resource

# Move the resource

# List the resources

# 