**Deploy and manage Azure compute resources**

**Create a Linux virtual machine in Azure**

* new vm machines network security settings by default has Outbound request are allowed. Inbound traffic is only allowed from within the virtual network.

Vm creation scenario sizes

|  |  |
| --- | --- |
| General use | B,Dsv2-3, Dv2 |
| Heavy computing tasks | Fsv2, Fs, F |
| Large memory | Esv3, ev3, M, Gs, Dsv2, Dv2 |
| Data storage and processing | NV, NC, NCv2, NCv3, ND |
| High performance computing | H |

When you create a VM with a single OS disk and data disk by default they are both encrypted using SSE, storage service encryption

Map storage disks

* OS disk; primary drive that stores the operating system. Labelled as /dev/sda by default
* Temporary disk; used to store swap files

Data disk

* Can storage data in primary drive along with OS but recommended to create dedicated disks. Each disk can hold up to 33,000gb

Unmanaged disk

* Manually managed

Managed disk

* Managed by azure and is recommended choice

Network communication

* Communicates with external networks using ‘Vnets’

SSH authentication

* Encrypted connection over terminal shell
* Sign in using username and password or SSH key pair
* Public key or private key

Network security group

* Tool used to enforce and control traffic
* Security layer that provides a firewall to filter inbound and outbound traffic
* Can be created in azure portal or command line tool
* Network rules are in priority order starting from lowest

**Manage VM’s using Azure CLI**

* Command line tool used to manage azure resources

VM storage options

* OS Storage
* Temporary storage
* Data storage

VHD files are stored as ‘page blobs’

* General purpose and premium both support page blobs
* Standard is used if your concern is about cost and premium if it’s about performance

Data disk

* Can be attached to a VHD at any time, cannot be deleted from storage while attached
* If an there’s an existing VHD in storage you can add to VM using powershell
* Azure portal can create a new empty data disk to a VM

Config data disk to VM

az configure --defaults location=eastus

attach empty data disk to vm

az vm disk attach \

--vm-name <name of vm> \

--name <data disk name> \

--size-gb 64 \

--sku Premium\_LRS \

--new

* When creating a new data disk, it must me initialized and formatted before use
* Can be a one-time operation when connecting using SSH
* Using a script to automate the task can save time, use ‘custom script extension’
* Store the scripts using ‘azure storage’ or a public storage like ‘github’

Run az vm show to get your VM's public IP address and save the IP address as a Bash variable

ipaddress=$(az vm show \

--name support-web-vm01 \

--show-details \

--query [publicIps] \

--output tsv)

What premium storage to use?

* Ultra-disks; best performance, can only be used as a data disk
* Premium SSD; can be used with VMs that include ‘s’ in the series name like ‘Dv3-series’
* Standard SSD; limited to specific VM sizes. Available in specific regions and only work with ‘managed disks’
* Standard HDD; can be used with any VM and billed at a lower rate

**Resize virtual disks**

* To resize or modify vm sizes you need to deallocate the vm
* After deallocating then update the vm size, the restarting to update changes

1.

az vm deallocate \

--resource-group <resource-group-name> \

--name <vm-name>

2.

az disk update \

--resource-group <resource-group-name> \

--name <disk-name> \

--size-gb 200

3.

az vm start \

--resource-group <resource-group-name> \

--name <vm-name>

* After resizing the size of the vm you have to let it know about the extra space
* If you have a reserved IP address and is deallocated and restarted by default it will be given a new IP address

**Secure Virtual Disk**

Encryption

* A way to secure information by encrypting using a bunch of random letters and numbers
* Uses a form of key that is asymmetric or asymmetric
* symmetric is the same key used to decrypt and encrypt whereas asymmetric is using different keys for both
* symmetric is faster whereas asymmetric is slower

Azure disk encryption technologies

* SSE, storage service encryption; performed on the physical disk
* ADE, azure disk encryption; managed by the owner using bitlocker. Encrypts on VHD
* Both use 256-bit encryption

Azure key vault

* Encryption keys are used by ADE and stored in a azure ‘key vault’
* It is a tool used to store passwords and accessing secrets
* Access policies needs to be enabled in order to use them

The policies are

* Disk encryption; needed for ADE
* Deployment; enable to retrieve secrets from key vault when its referenced, like when you create a vm
* Template deployment; same as deployment but when using a template

Encryption is also possible on an existing disk

* Windows allows you to encrypt the whole disk or just the OS
* Some Linux distributions, only the data disk can be encrypted
* Can be decrypted too using azure cli or powershell

**Encryption Using Azure Resource Manager Templates**

* Templates are json files used to define a set of azure resources
* Templates can be reused to create resources
* Can be created manually or through azure portal
* The templates are stored in github which can be deployed from azure

**Keeping VMs Updated**

Update management solution

* Tool use to manage and update OS and patches for Windows and Linux on-premises and cloud

Advantages

* No agents needed for config within the vm
* Can run updates without logging in to vm
* Lists missing updates and solution lists in an easy read format

Components Used

* Microsoft monitoring agent
* Powershell desired state config
* Automation hybrid runbook
* Microsoft update

Compliance scan

* Performed every 12 hours on windows and 3 on Linux machine
* After updates it will forward to azure log analytics

Recurring updates

* Can created a scheduled deployment of updates. To run on specific pcs

**Resource Manager Templates**

* A template that defines all resources in an azure deployment
* A JSON file that is a form of declarative automation. Defines what resources are needed but not how to create them. Basically, say what you need but the let manager create them for you

Why use templates

* Makes deployments faster and more repeatable.
* Decreases errors when creating manually
* Can be linked together
* Can be reused
* Consistency and it provides a common language to describe deployments.
* Better to use an existing template that to write from scratch

Azure quickstart templates

* Templates provided by azure community and available on github

Before deploying a template, you need to verify first using ‘linter’ to verify JSON syntax

* Then visualize the template using ‘az resource manager visualizer’ to see how it relates to our other resources
* Then test template on a test development, it does not create the resources but provides feedback on what would happen if it does

Extend templates

* Can do so by creating multiple templates or modify the existing one to suit your needs
* To use json template with azure use the ‘az vm extension set’ command.
* Resource manager uses the ‘dependsOn’ element to determine the correct order to deploy resources.

**Create VM using VHD templates**

VHD Disk is a virtual disk created and hosted on cloud

* Can contain multiple VHDs
* Durable, highly available as its on cloud
* Stored on cloud and can be manged securely with all its features that azure offers
* Scalability
* Cost and performance vary to suit any need

Virtual machine image

* A template used to create the VHDs to run the virtual machine
* Images created using Hyper-V can be uploaded to azure and created also

Generalized Image

* If created using Hyper-V a blank disk needs to be created first then create the vm will be created using this disk. From there you can add the OS when starting the vm
* If using Azure marketplace, the vm will be created with an existing image with an OS. From there you add your own software, os updates, and other packages required
* Windows tool uses ‘sysprep’
* Linux tool uses ‘waagent’

After building the image you must reset some things because when you create the image it comes with the

* Hostname of vm
* Log files
* Username and password when prompted during creation
* Security identifiers

They must be reset to default state before you create more vms. Otherwise you might end up will multiple vms with the same credentials. This process is called ‘generalized image’

Specialized image

* A virtual image copy of a live vm after it has reached a specific state
* Good option for creating a snapshot of a vm or backup. You can restore your vm using this image
* If used to create a new vm, the new vm will retain all data from that image

**Generalize a server**

Windows vm using ‘sysprep’

Linux using ‘waagent’

* Shut down vm and deallocate
* Then using command line tool generalize the image
* Create new image using vm’s generalized image

**Build a scalable application with virtual machine scale sets**

* A way to adjust how resources are used when demand starts to overload
* In azure you’re able to do this using ‘machine scale set’
* Uses a load balancer to distribute requests across VM instances by using health probe. It uses ‘pings’ to determine what to do with the instances. Like if it doesn’t respond to a ping then it will stop sending requests

Horizontal scaling

* Process of adding and removing machines, depending on demand

Vertical scaling

* Adding resources such as memory, cpu power, or disk space to vms

Scale sets

* Scheduled scaling, can adjust if there’s a spike in traffic and scale back down if it decreases
* Auto scaling; basically, if you can’t do schedules scaling, but based on metric threshold scaling

Reducing cost using low-priority scales

* Scale option that can save up to 80%
* Option when there are a lot of unused resources

Two kinds of removal

* Delete;
* deallocate

Autoscale conditions

Schedule

* increase workload at a specific date or time period
* when time falls during the time period the scale condition is evaluated before anything is done to add or remove instances.
* Default scale condition pretty much has no scale rules and is like a ‘null scale’

Metrics

* Scaling is based on the resources to monitor such as CPU, memory usage, or response time
* It removes the instances based on the rules of these metrics

Commonly used metrics include

* Percentage cpu
* Inbound flows and outbound flows
* Disk read/write operations
* Data disk queue depth

Scale

* One-minute interval is too short to make autoscaling worth so minimum is 5 minutes. This is called ‘duration’
* When a scale detects it has gone past the threshold it will perform a ‘scale action’. This can be ‘scale in’ where it decreases or ‘scale out’ where it increases
* Scale action uses ‘greater than’ or ‘less than’ or ‘equal’ to determine how to reach to the threshold
* ‘cool down’ period is when a scale action is stabilized determined in minutes. Minimum cool down is 5 minutes

**Install and update apps with VM scale sets**

You can use custom script extension to roll out automatic updates on your app. In this case it can automate the same tasks on a scale set

Install using custom script extension

The process is the same as using a custom script extension

* Use the ‘az vmss extension set’

Update an app using scale set

The updates are according to a policy such as

* Automatic; no specific time/date for update. Could all be updated at the same time
* Rolling; updates in batches with an optional pause to minimize or eliminate service outage. Requires a ‘health probe’ or ‘application health extension’
* Manual; default mode and change must be done manually

**Protect VM settings with Azure Automation State Configuration**

Azure automation

* Makes sure that the VMS in a cluster are in a consistent state. Should have the same software for all VMS
* Built on powershell, make sure modules are up to date

Why?

* To maintain a consistent config across all resources
* Has built in pull server so azure can automatically get update configs from the server.

Powershell DSC

* A desired state configuration, is declarative that describes the desired state but not the steps needed to achieve that state
* Push mode; admin manually or auto sends the configs to one or mode nodes.
* Pull mode; pull server holds all the configurations. Pulls at 15 minute intervals.

LCM

* Local configuration manager
* Responsible for updating the state of a node, like a VM, to match the desired state
* It goes in steps get, set, and set.

**Deploy and run a containerized web app with Azure App Service**

Build and store images using container registry

Container registry

* Tool used to create docker registries like docker hub
* Like docker hub but on azure it offers more security

Build docker image

1. Create container registry in azure portal
2. Download source code to build image from, like from github repo
3. Move to source folder
4. az acr build --registry <container\_registry\_name> --image webimage .
5. once built check on registry resource to confirm

Deploy web app

* you can deploy web app directly from container registry
* create a web app resource and in the basics, tabs select required options and point to container registry and created image.

Update the image automatically using web hook

* service used by azure to receive notifications about updates to images in the registry
* for azure this feature is called ‘tasks’ it works to rebuild your image every time the source code is changed.
* This can be set up in container registry and if your web app is set up for continuous integration in app service, it receives a notification via webhook and updates the app

Azure container registry is used to hold docker images and can create web apps using the images. Webhook can configure continuous deployment for web apps. It monitors any changes made from the source and is updated accordingly.

**Scale a web app service with scale in and scale out**

Creation of a new web app comes with creating an app service plan or using an

existing one.

* If using an existing plan all web apps will share resources with your app
* If they have different requirements, then a different service plan is needed
* Key to scaling is when to scale and by how much. You monitor by using the metrics available for the app service

**Azure container instances**

Why use azure container instances?

* Fast start-up
* Per second billing
* hyperV level security
* custom sizes
* persistent storage
* linux and windows

create a container instance

az container create \

--resource-group deploy-container \

--name mycontainer \

--image microsoft/aci-helloworld \

--ports 80 \

--dns-name-label $DNS\_NAME\_LABEL \

--location australiaeast

Check

az container show \

--resource-group deploy-container \

--name mycontainer \

--query "{FQDN:ipAddress.fqdn,ProvisioningState:provisioningState}" \

--output table

**Control Restart Behaviour**

Restart policies

* always;
* never; runs one-time only, never restarted
* OnFailure;

Troubleshoot container instances

* Use container logs & container events

az container logs \

--resource-group <group name> \

--name <container name>

az container attach \

--resource-group learn-deploy-aci-rg \

--name mycontainer

**Kubernetes**

* A service solution to make running multiple container instances easier to manage and deploy
* not monolithic; deployment, management, scaling, and such are optional and not grouped together
* Kubernetes deployments are done in clusters with a master machine and one or more workers.

Core pieces

* Pod; collection of containers co-located on a single machine
* Service; load balancer to reduce traffic
* Deployment; used for replicating a container multiple times

Azure Dev Spaces

Basically, helps teams be more productive on Kubernetes and allows you to

* Minimize local dev machine set up
* Code quickly using Visual studio code or visual studio
* Generate docker and Kubernetes config code from dev to production
* Develop code in isolation and do integrated testing with other components

When to use Kubernetes

