Cloud Architect using Microsoft Azure - Scholarship

Lesson 1: Introduction to Designing Infrastructure and Managing Migration

IaaS vs PaaS vs SaaS:

IaaS	PaaS	SaaS
• Physical Data Center	• OS X	• Programs, Websites, Apps
• Servers, Storage, Networking	• Database management tools	

Desing Infrastructure:

Advantages of cloud over on-premise -

1. On-demand availability
2. scale quickly - automation
3. pay what you use
4. security focus infrastructure
5. easy setup.

Issues IT staff regularly deals with for on-premise -

1. Licensing - multiple softwares, paperwork, POs.
2. Lack of solutions available
   - solution from 3rd party provider
   - compatibility issues
3. - Lack of skilled IT staff to integrate
4. Infrastructure limitations
   - servers going down, performance issues
   - limited staff to maintain physical servers
   - global server presence

Design Principles:
1. design for self-healing
2. redundancy
3. use managed services
4. design to scale out
5. build for needs of business

Potential stumbling blocks:
1. migrating resources into solution that is not cost-effective.
2. underestimate time needed to migrate to cloud, including modifying on-premise application.
3. not having strategy to revert changes back to on-premise if cloud solution does not work out - e.g. drop in performance, latency issues, higher costs.
4. underestimating costs associated with migrating services into cloud.
5. not properly setting up resources to ensure high availability.
6. not understanding service interruptions covered by SLA and setup / configure appropriately.
7. not properly explaining value + advantages of migrating resources into cloud.

Key Stakeholders:

CEO	Sales	
CFO	Accounts/Finance	→ [ ]
CTO	HR	
COO	R+D	→ [ ]
	IT	→ [ ]

Subscription

Resource Group	Services
[ ]	web App, SQL storage
[ ]	web App, VM, SQL storage, AKS
[ ]	" " "

Criteria for a Succesful cloud migration:

1. identify + involve key stakeholders.
2. create strategic plan
   - what model - IaaS, PaaS, SaaS?
   - what workload to migrate ?
   - which cloud provider is best ?
   - will be locked into cloud provider ? cloud agnostic ?
3. calculate cost of ownership
   - cost of migration
   - ongoing cost of cloud services.
4. discover + assess on-premise resources
   - software requirements
   - hardware
   - security on-prem vs cloud.
   - data type
5. training
   - users, maintainers, developers
6. plan for decommission on-premise resources when not in use.
7. plan for optimization after migration
   - performance tests + comparisons.

Reasons not to migrate to cloud:

1. not ready for temporary initial dual cost of on-premise + cloud resources during migration.
2. business applications are not straightforward, require extensive modification to work in cloud.
3. some business applications run best locally, better performance. High latency issues if in cloud.

Reasons to migrate to cloud:

1. lower cost, pay as you go
2. easier secure collaboration for users due to built-in security protocols to restrict access when applicable.
3. no expense for buying equipment/hardware.
4. global presence with world-wide data centers
5. improved security deployment of resources following best practices and implemented via infrastructure-as-code.
6. reduction of end-of-life concerns.
7. grow as business grows.

Ready to migrate:

1. invested heavily in applicat development that are not cloud ready.
2. no clear understanding what needs to be migrated.
3. no clear method to determine successful implementation.
4. not identified cost related to migration or performance implications.
5. not able to answer why migration now.
6. compliancy limitations/restrictions.

When to migrate:

1. most of business applications are cloud ready.
2. identified total costs + cost benefits.
3. age of existing hardware reaching near end of life.
4. company requires more remote + mobile access.
5. clear strategic plan for all applications + dependencies.

Virtual Machine (VM)
- emulation of computing system that uses software instead of physical computer to run programs + applications.
- web servers - host web pages + handle HTTP
- database servers - provide database service
- email servers

→ Azure Portal - Create VM

High Availability:
- Azure resources available during
  1. scheduled / unscheduled maintainer
  2. natural disaster.
- Azure availability configuration
  1. availability zone in Region
  2. VM Scale sets — Fault domains
  3. Availability sets — Update domains

- rack of servers
- point of failure at one data center.

scheduled maintaince
for each instance sequentially but not all at same time.

Load Balancing:
1. application gateway - routing based on specific url instead of VMs.
2. front door - encrypti/decrypti.*
3. Azure load balancer - evenly distribute across VMs.
4. Traffic manager
   └ DNS load balancer to distribute traffic from around world.

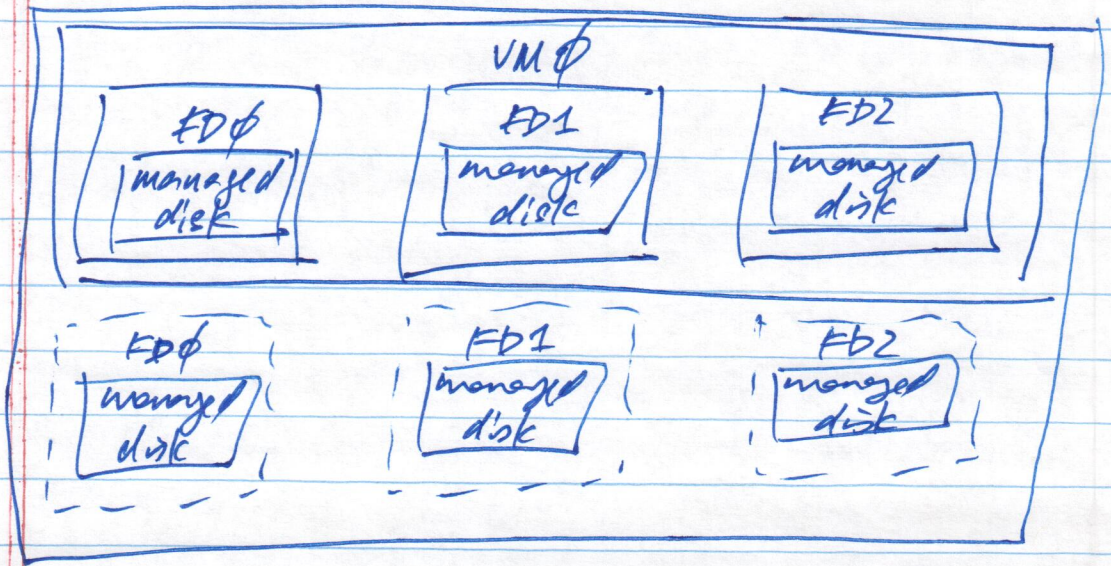
VM Scale Sets:

- combines groups of VMs to run as single unit.
  - provides high availability by creating + managing identical VMs.
  - use load balancing + auto scaling to keep apps up in VMs.
  - available after increase in resource use.

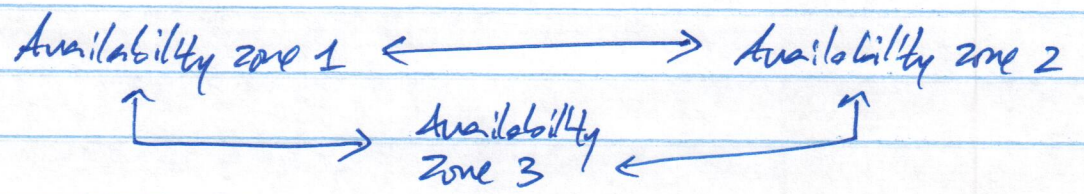
- advantages
  - maintain high availability + applicati resiliency.
  - apps automatically scale as resource usage changes.
  - works at scale.

Availability Sets:

- strategic group of VMs to ensure no single point of failure within data center.
  - may not protect if entire data center goes down.
  - update domains + fault domains.
  - need at least 2 VMs.



Availability Zones:
- provides unique physical locations within an Azure region.
- spread out across multiple zones to ensure high availability.
- continue accessibility to apps even after one zone fails.



Load Balancer to Distribute Web Traffic:
Azure Portal → Virtual Networks → Subscription
Resource group
Name
Region
→ IP Addresses
→ Create

Azure Portal → Load Balancer → Subscription
Resource Group
Name
Region
SKU - standard
Type - public
Tier - regional
→ frontend IP → Name
IPv4
IP
IP address ← name
availability zone
zone-redundant

→ Create

Azure Portal → Virtual Machine → Subscripti
Resource Group
VM Name
Regin
Availability optim - Availability Zone
\ Availability set
Image
Size
Username / password
Inbound port rules - RDP, SSH

Networking → virtual network - select
Subnet
public IP
NIC - Advanced
Configure network security group - create new

Management → Create

Network Security group → Inbound security rules
→ Add → Source - Service Tag
Source - Internet
Destinati port range - 80, 443
Protocol - TCP
Acti - Allow
Priority - 1010
Name

VM #1 → Networking → Load Balancing → Add →
- Load balancing opti - Azure
- Select load balancer - select
- Backend pool - create new
  - name

VM #2 → Networking → Loading Balancing → Add → ...
Backend pool - use existing

Load Balancer → Health Probes → Add
- Name
- Protocol - HTTP
- Port - 80
- Path - /
- Interval - every 5 seconds
- Unhealthy threshold - 2 consecutive failures

→ Load Balancing Rules → Add
- Name
- IPv4
- Frontend IP address
- Protocol - TCP
- Port - 80
- Backend port - 80
- Backend pool - select
- Health probe
- Session persistence - persist cookies
- Idle timeout

VM → connect → RDP → login → Add Roles + Features
#1 #2 · check web server → Install (repeat for VM #2)
restart if necessary

→ Tools → IIS Manager → VM name
└ Sites
└ Default website
Advance settings
└ explorer - path [physical]
└ update html → index.html
web browser - load balancer frontend IP

App Service:
- HTTP service for hosting REST API, web, mobile apps.
- Key features
1. global scale with high availability.
2. serverless code - server independent.
3. use applicat templates - e.g. wordpress, DNN.
4. security + compliance already managed at server level.
5. managed productn environment - upgrades + patches
6. multiple languages + framework.

Create DNN web app:
Azure Portal → DNN platform → Name
Subscript
Resource group
App Service plan / Locati
SQL Database - name, server - login, password
location

Go to website → installati → username / password / email

database – custom
- SQL Server Express Database
- servername – name.database.windows.net
- databasename
- username / password

note: use standard tier for database as free tier is too slow for efficient installation.

after installati → login → customize.

Service Level Agreement (SLA):
- App Services – 99.95%
- cloud services – 99.5%
- VM – 99.9%
- Azure AD – 99.9%

Azure cloud Design Principles:
- design for self-healing
- design for redundancy
- design to scale out
- design for evolution
- design for operations
- managed services
- focus on business needs rather than infrastructure.

Azure Cloud Design Styles:
- big compute
- big data
- event-driven architecture - e.g. IoT
- microservices
- N-tier applicat - UI layer, logic layer, data layer - # 3

Communicat Between On-premise + Cloud:
Factors to consider -
1. latency
2. security - HIPAA, GDPR, accounts privacy
3. speed
4. redundancy

Tools for communicat -
1. VNET / VNET Peering - between different regions + subscripts
2. Gateway - between on-premise to Azure
3. VPN -
4. Network watcher - troubleshoot issues

Methods to communicat between Azure Resources -
1. VNET - e.g. Kubernetes, App services
2. virtual network endpoints
3. VNET Peering - connect different VNETS
4. private link - connect between Azure resources

Communication with on-prem - specific
1. point-to-site VPN - connect devices via VPN
2. site-to-site VPN - connect any devices on network
3. Azure ExpressRoute - private internet direct to Azure
- expensive, but secure.

Policy-based VPN:
1. uses combinations of prefixes from both networks (on-prem, cloud) to define how traffic is encrypted/decrypted through IP tunnel.

2. allows for multiple VPNs via single VNET gateway.

3. does not support VPN diagnosis in Azure.

Route-based VPN:
1. use any wild card traffic selectors and routing/forwarding tables to direct traffic to different IPsec tunnels.

2. better than policy-based VPN if device supports it.

3. can perform VPN diagnostic in Azure.

Deploy VPN gateways:
Required resources -
1. compatible VPN device on-prem.
2. public IPv4 address on-prem.
3. virtual network within Azure.

fault tolerance configurations -

1. Active / Standby - planned / unplanned maintenance of active will failover to standby.
2. Active / Active - failover to active gateway.
3. Express Route failover - failover of Express Route to site-site VPN.
4. Zone redundant gateways - failure of one gateway zone to another

Azure VPN:

1. Secure communication with Azure resources to on-prem network.
2. filter network traffic.
3. integrate with Azure resources.
4. route network traffic.

Create VPN:

Azure Portal → Create virtual network
→ Create virtual network gateways
└ Subscription
Name
Region
Gateway type - VPN or Express Route
VPN type - Route-based or Policy-based
SKU
Generation
Virtual network - select previously created vnet
Gateway subnet
Public IP address - create new
- name
- basic

Vnet gateway → Point-to-site configuration
(Address pool - list of IP address
Tunnel type - IKEv2 SSTP (SSL)
Authentication type - Azure certificate
- name, public certificate data
↑ copy + paste

Save
Download VPN client.
↳ install VPN client
go to VPN to see + select VPN

Network security group (NSG):
Secure traffic flow between -
1. application and internet
2. applications
3. users and application
note: NSG rules take precedence over Windows Firewall rules.

e.g. configure FTP server on VM
Add Inbound Port Rule
Destination Port Range
Protocol - Any
Name - FTP
Azure VM → networking → inbound security rule
Azure VM → RDP → add roles + features
↳ add websrvr
↳ add FTP server, telnet client
→ IIS → add web index.html
→ add FTP site
→ FTP firewall support - data channel port range
IP address of server
(cmdR) control firewall.cpl → advanced settings → inbound rules → new rule

↳ Port

↳ TCP, Specific local ports

cmdR → services.msc → Microsoft FTP Service

↳ right click - restart

Common Issues

1. If VMs behind load balancer not responding to health probes, make sure IP addresses are added to NSG allow set.
2. Load balancer stuck in failed state, edit to change state.
3. Can't change backend ports for existing load balancer rule
   - delete health probe by updating VM scale set.
   - update port after removing VM scale set.
   - reconfigure probe again after port has been updated.

Lesson 3: Designing for Backup and Recovery

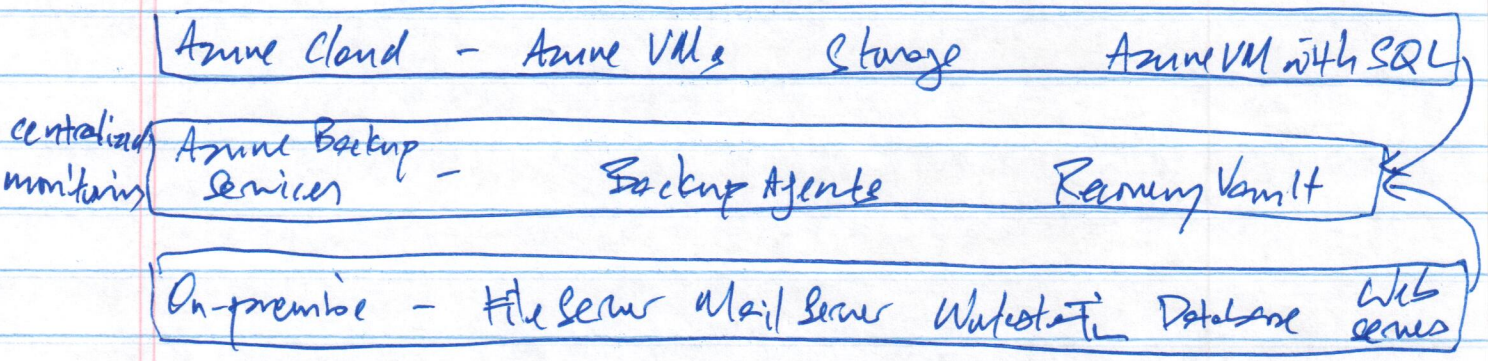
Things to consider

1. identify workloads + usage.
2. plan for usage patterns (e.g. workhours, weekdays)
3. establish availability metrics
   - mean time to recovery (MTTR)
   - mean time between failures (MTBF)
4. establish recovery metrics
   - recovery time objective (RTO)
   - recovery point objective (RPO)
5. determine workload availability targets
6. understand SLA

Best practices for implementing a backup plan:

1. perform failure mode analysis. (e.g. chaos test)
2. have a redundancy plan.
3. utilize resiliency strategies when applicable
   - (system's recovery ability after failure
   - prioritize which applications critical to business.
4. make availability a consideration when designing solution
   - SLA - what percent of time resources available.
   - + backups
5. have plan + documentation on how to store, backup + replicate data.

Azure Backups:



Azure Backup Services -

1. runs in cloud + store recovery points.
2. enforces policies created/modified in cloud
3. ready for backup → backup agent → VSS snapshot → VM storage

Items that can be backed up -

1. On-prem resources
2. Azure VM
3. Azure Managed Disks
4. Azure file shares
5. SQL server in Azure VM
6. Azure Database for postgreSQL

Backup features –

1. block level incremental backups – only saves changes to reduce size, time.
2. data integrity verified in cloud.
3. configurable retention policies for storing data in cloud.

Advantages –

1. full flexibility for when backups are taken – manually or scheduled.
2. Support for VMs on both Hyper-V and VM Ware
3. no special licensing required.

Disadvantages –

1. unable to backup Oracle workload

Azure Portal → Recovery Services Vault → Add
- Subscription
- resource group
- name
- region

vault name → Backup → where workload running – Azure, on-prem
what to backup – VM, files, SQL, ...
follow steps – download client + install

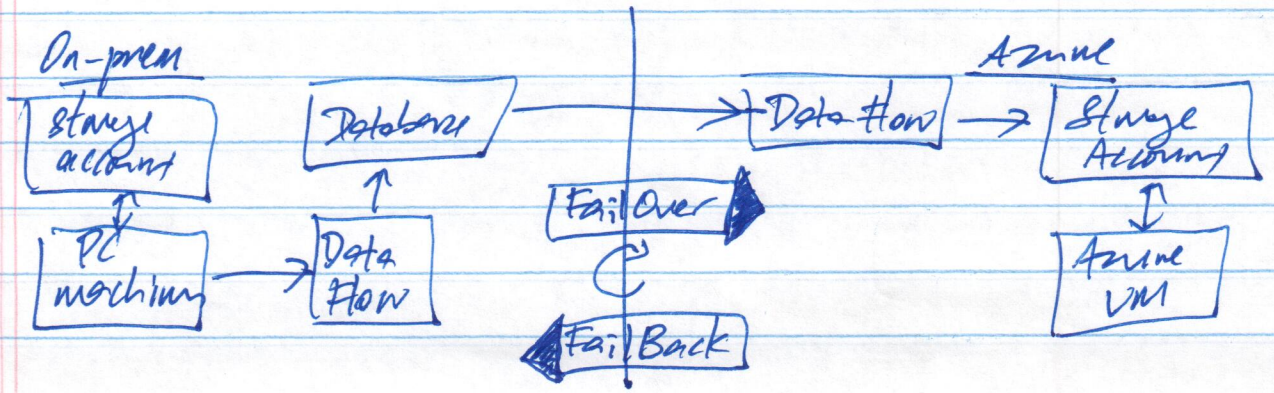
or VM → Backup → Recovery Services Vault
choose backup policy – name
- frequency
- instant restore – days
- retention range – days

vault name → Backup item → select backup item – restore / recover
→ Backup jobs

Site Recovery:
- Azure VM replication during failure
- On-prem replication – cost of equipment, electricity, network, labor, ...

- Site recovery replicat policy
  - define number of recovery points.
  - how often snapshots taken.

- Site recovery capacity
  - bandwidth limitation
    - ASR scans + generate report of needs.
    - tools to properly size ASR deployment.
    - report includes – bandwidth required for ongoing replicat / initial synchronize.
      - amount of storage space.
      - # of storage accounts
      - # of configurat servers needed on-prem.



- Site recovery network
  - networking options
    - VNET - connect from one region to another.
    - Expressroute / VPN - connect from on-prem to Azure.
  - outbound connectivity for URLs that needs to be allowed within firewall:
    1. *blob.core.windows.net
    2. login.microsoftonline.com
    3. *.hypervrecoverymanager.windowsazure.com
    4. *.servicebus.windows.net
    5. *.vault.azure.net

Azure VM → Site Recovery → Choose Region → Host Replic
Azure Services → Recovery Services Vault → Choose Name

- Test Failover ← Replicated items
  - Failover direction - Region 1 to 2
    - recovery point
    - Azure VNET
- → Site Recovery Jobs
- → Replicated items - status
  - clean up test failover
- → Overview

Failover

- Failover ← Replicated items
  - Choose recovery point
- → Replicated items - status
  - replicated machine → commit

FailBack { Replicated items → disable replicati
→ unprotect
Failover - choose RPO
unprotect

Azure → Recovery Vault → Backup items
restore VM
- select restore point
- create new or replace existing
↳ shutdown old VM

→ Backup Jobs

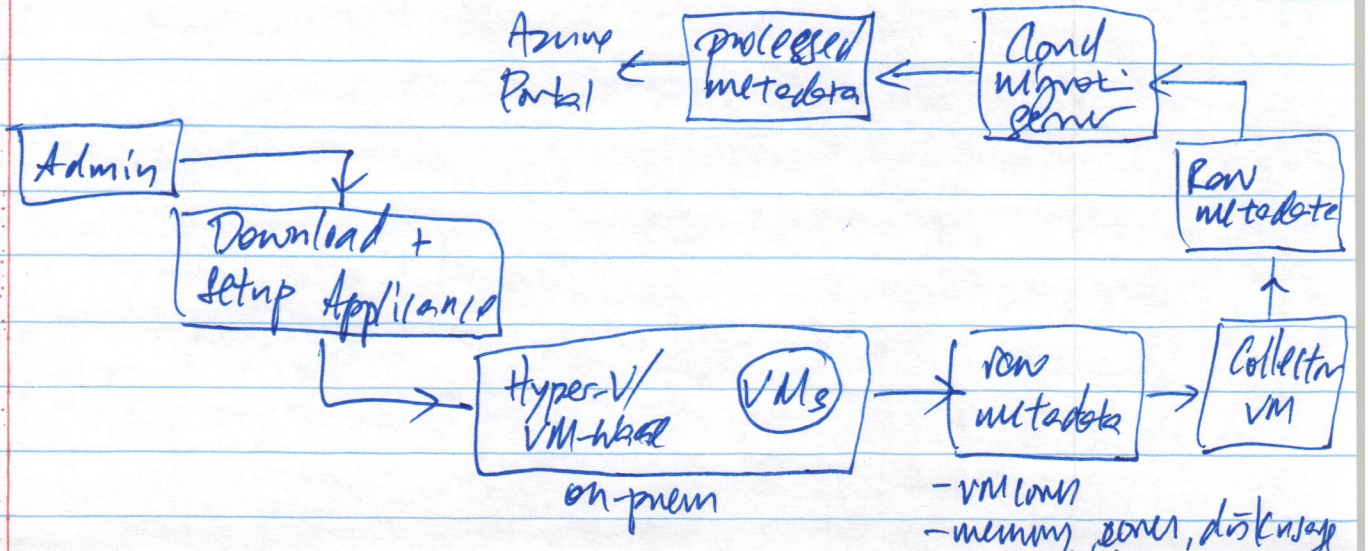
Installation of Azure Recovery Services Agent
- schedule backup
- backup now

Azure Migrate has 4 main stages for migrat
1. assess - reviews environment for potential issues
which workloads can be migrated.
2. migrate - match current workload
3. optimize - discover areas to optimize
4. monitor - continuous system to look out for unexpected
anomalies, issues, missed dependencies to
maintain health + performance of systems.

1. Assess Stage:
- identify servers, apps, resources to be migrated.
- inventory of on-prem computing resources + dependencies.

- develop a map of how different pieces communicate + interact with one another.
- Azure Service MAP
- best migration option
  1. rehost - minimal
  2. refactor - optimize
  3. re-architect - significant changes
  4. re-build - from ground up.
  5. replace - alternative solution.

- Assessment option in Azure Migrate
  - provide performance-based sizing calculations
    - VM sizing
    - compute, storage
  - provide estimate of on-going cost of running resources with Azure
  - provide visualization of dependencies
  - assist in creating groups of devices to be assessed together and migrated to Azure at same time.



Azure Portal - review assessment
- Azure readiness - ready, not ready, unknown.
- monthly costs
- monthly storage costs

- cost analysis
  - tools to keep within budget
    - Azure Total Cost of Ownership (TCO)
    - Azure Advisor
    - Azure Cost Management

- compute products
  - virtual machine scale sets - manage groups of load balanced VM. - automatically/manually add VMs as needed.
  - Azure Kubernetes Service (AKS) - containerized app that are managed & scalable.
  - Service Fabric - distributed system platform to package, deploy, manage microservices & containers.
  - App Services - run applications via managed VMs.
  - Container Instances - on demand managed serverless Azure environment.
  - Batch - run & scale large number of computing jobs in a batch pool.
  - Azure dedicated host - dedicated physical server to host Azure VMs

Azure Portal → Azure Migrate → Servers → Discover Server Assessment
- type of virtualization

- name
- download Azure Migrate appliance
  (Azure Migrate Appliance - turn on + run
→ Server Assessment - Assess
  (select machines - name
  - select or create group name
  - select machines
  → create assessment
→ Server Assessment - Assessment
  (report

Total cost of ownership (TCO) - Azure website calculator
- hardware / software costs
- implementation time
- user training
- support + maintenance

2. Migrate
- Things to consider
  - Active Directory vs. Azure Active Directory.
  - List of resources to migrate.
  - Projected timeline.
  - migrate as many applications to PaaS or SaaS to minimize patches + updates.
  - Automate backup strategy for on-prem and cloud.

-Tools - ASR
  - Azure Data Migration Service.

(Azure Site Recovery)

- Migrate workloads using ASR
1. prepare source + target environment.
2. start replication process.
3. test to make sure works.
4. failover from source servers to Azure.

- Database migration tool
- on-prem to Azure database
- review environment, provide report of databases can be migrated.
- use tool to migrate database.

- Potential limitations -
- max 100 VMs simultaneously.
- amount of time depends on VM size, connection speed, number of VMs.

- Post migration
- review security settings of VM
- backup schedule
- use ASR to replicate VMs to another region.

Azure Portal → Migrate → Servers → Server Migration → Database Migration
- virtualized?
- Download + install

Migrate ← Replicate ←

3. Optimize
- Review cost using Azure Cost Management
  - breakdown of resources for subscription
  - review cost by resource group
- Azure Advisor
  - resize VM size up or down based usage.
  - discounts
  - reserving resources for discounts.
  - security + performance advisories.

4. Monitor
- Azure Monitor
  - provide health + performance details
  - automate review of data + filter only relevant data.
  - create alerts to notify when thresholds exceeded.
    e.g. autoscale metric, CPU exceeds threshold

Azure Portal → Monitor → Metrics → select scope
  - resource group
  - refine scope
    - location
    - storage account
    - capacity - metric
    - availability

→ Logs → Log Analytics
→ Activity log → filters, timespan
  - select event → new alert rule

Resource
└ Conditi

Actions – add action group
↳ Create action group
└ subscripti
resource group
acti group name
└ notificati
- email/sms
- name
└ actions
- action type – Azure Functi
Automati Runbook
└ ....
– name

→ Review + Create
└ Alert rule name
· same alert rule to – select resource group.
→ Review + Create

Azure Portal → Azure Advisor – Cost
– Security
– Reliability

Common issues
– not seeing performance data for VMs in assessment reports
↳ make sure to allow outbound traffic on port 443.
– low confidence ratings on assessment report
↳ make sure VM powered on, Hyper-V-VM dynamic memory enabled.

Lesson 5: Automati

ARM template:

Azure Portal → JSON → VMs templates

↓ contain template.json parameter.json

- why use
  1. Consitency within deployments.
  2. complex deployments made easy.
  3. reduce errors.
  4. easy to re-use.
  5. speed.

- elements of ARM template:
  1. Schema
  2. contentVersion
  3. parameters
  4. variables - unique things, globally defined
  5. functions - pre-existing or user-defined
  6. resources
  7. outputs

Azure Portal → Custom Deployment → Select template
- Create Linux VM
- " Windows " → subscripti
- " Web app → resource group
- " SQL datase → regio
- ... admin username
  admin password
  → Create

→ Download template for automati

→ Custom Deployment → Build own template in editor
- load file (template.json)
edit parameters - load file (parameters.json)
- subscription
storage name
→ Create

Deployment options for ARM templates
1. VS Code
2. Azure Portal (template editor)
3. Azure CLI
4. Powershell
5. GitHub
6. Cloud shell

ARM template best practices
1. template limitations - maximum
- parameters - 256
- variables - 256
- resources - 800
- output values - 64
- characters in template expression - 24576
2. minimize parameters when possible
3. secure data - secure string
4. proper use of variables

template.json - default values
"defaultValue": "[uniqueString('storage', resourceGroup().id)]",
"metadata": {
"description": "Generate unique storage name" }

Azure Portal → Template spec → Create template spec
- name
- subscription
- resource group
- version
- location
- edit template - copy + paste

(keep track of changes ←)

→ Create
↳ view, edit, delete, deploy

Desired State Configuration (DSC)
- specify exactly software environment setup.
- makes maintenance easier, checks for compliants + notifies or makes correct.
- DSC script:
  1. configuration block
  2. node block
  3. resource block

e.g. Configuration DeployHVRole {
Node Backoffice.localdomain {
WindowsFeature HyperVRole {
Name = "Hyper-V"
Ensure = "Present"
LogPath = "C:\...\Logs.txt"
}
}
}
DeployHVRole

DSC requirements

1. Automat account
2. Supported OS - Windows, Linux
3. Windows Nodes running in Azure require WMF 5.1
   - if Windows Server 2012 and Windows 7, enable WinRM

Windows Management Framework

Windows Remote Management

Azure Portal → Automat account → Add
- name
- Subscript
- Resource group
- Azure Run as Account → No
→ Create

DSC script - Configuration IISInstall {
node "localhost" {
WindowsFeature webserver {
Ensure = "Present"
Name = "Web-Server"
}
}
}

Automat Account → DSC → Add
- select file to upload
- name

Configuration → select DSC → compile

Nodes → Add → select VM → connect → select node configuration name
refresh frequency
configuration mode - apply + monitor
- apply + autocorrect

Allow module override - ✓
Reboot node if needed - ✓
Action after reboot - continue configuration

VM → Connect → RDP → Window Server Manager
↳ IIS installed.

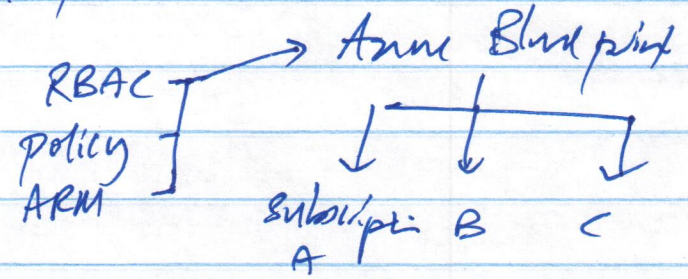
Automation Account → DSC → Nodes → Status review

Azure Policies :
- handles - multiple subscriptions
  - manages regulatory compliance
  - multiple teams
  - standardized of cloud resources configuration.
- apply policies at scale

Azure Portal → all services → management + governance
Assign Policy ← Assignments ← policy ↵
- scope - Subscription
  - resource group
- basics - policy definition - types - search - e.g size
- parameters - allowed size skills
- non-compliance message
- remediation
→ Create

Azure Blueprint
- repeatable resources, implemented, following organizational standards + requirements.
- enables auditing, traceability, compliance of deployments.

- continues to exist + have functionality after deployment.
  - vs. ARM template no relationship after deployment.
  - Resource Manager templates removable in Blueprint
- can include ARM template, Azure policy.
- can orchestrate deployment of
  1. role assignment
  2. policy assignments
  3. ARM templates
  4. resource groups



Azure Portal → Blueprints - PCI compliance.

ARM template common errors:
1. validation - syntax errors or quota limits
   - use template validation tool in PowerShell
2. deployment - review logs for details - e.g. permissions