**70-534 Architecting Microsoft Azure – Udemy**

Check latest exam requirement objectives consistently.

**Study Plan:**

How many hours am I going to give to this each week? GMAT/School/CCNA. Don’t start and have a break. Keep going. Don’t cram.

Create a study plan, $165.00 per exam. Exam can be proctored from home. 2 and a ½ hours with 40 questions. Look at latest test changes. Changes frequently. 1 month of dedicated studying. Official book from library and study test were not the best. One week of extra buffer.

Scenario questions. Different question types. Focused around hybrid solutions. Some vague questions. Fair test.

**ASM (Azure Services Model) – classic portal vs Azure Resource Model (ARM)** – newer portal with deployment model. Some things you can not do and some things are not easier in either portal. Default will be the new portal. Slowly migrating off the old portal

ARM templates/ARM backups were recently added. UDR (User Define Roots, Application Gateway, Azure Storage encryption, Azure Disk Encryption, SQL database TDE, Azure Scheduler,

**Network Services – Objective 1:**

**GFS (Global Foundation Services) called MCIO (Microsoft Cloud Infrastructure and Operations) Datacenters:**

Azure is all over the globe. Products are a ton.

**HA** – architected to stay up and running. 99.999% availability uptime. Two or more VM’s in a set are 99.95% availability. 10% credit or <99% is 25% credit.

Datacenters support all of Microsoft’s online businesses. Large as 3 cruise ships. Over 100 datacenters in 30 regions and 11 geos. NA, EU, AS, SA

Geo is a country but larger than a country. China region is special as a china company runs the datacenter. Needs a separate Azure subscription.

Region are paired with other region. NA/SA for example. DC’s are updated only one pair at a time. Regions don’t support all resources. Australia DC is only available to them. Same for India. Keeping data in region. 300 miles between DC.

**Triple Redundant storage** – 3 times replicated in that region. Or can elect up to 6-times and across geos. Brazil is paired with US in one instance. Pairs are in same geo for data protection laws.

**Racks** – servers are arranged in groups. Two blades per 1U. Height total of 52U. 96 servers on a single rack which would be 48U. JBOD (Just a Bunch of Disks (up to 60GB). 20 racks make up a cluster.

All pre-wired. Clusters have the same hardware. Cluster can have close to 1,000 servers.

**ITPAC** – container of servers, built as one unit and shipped to a Microsoft datacenter for plug and play

Water, power, electricity

**Active Directory:**

Identity, Roles, and Permission, Company directory, Password Policies

**AD DS** – Domain Services, employees log into windows

**AD LDS** – light weight domain services

**AD CS** – Certificate Services, PKI (Public key Infrastructure)

**AD FS** – Federation Services with SSO

**AD RMS** – Rights Management Services, protect documents

Not a replacement for on-prem AD. Azure

Azure AD is there to extend AD from on-prem AD to cloud. Identity management centered, no hierarchical object model. Basic Service. Dynamic or Static IP’s

No private IP to VPN. CNAME records can be used to point your DNS to the Azure domain. A record for DNS model. 4096 private IP per VNET, 60 public dynamic IP’s, 20 public static IP’s first 5 are free and .004/hr. 500 VM per VNET. 50 VNET

ACL’s are for endpoints not vnet/subnets. NSG (Network Security Groups), powerful than ACL. Blacklist IP’s with ACL. You can have up to 50 ACL per endpoint. Ordered rule by priority. Packets get filtered before reaching the VM. Doesn’t take CPU cycles

NSG contain ACL rules. Can be associated with vnets, subnets, or VM’s. NSG ACL applie to the VM inside the subnet. Can only be applied in region. Have priorities

**Resource Groups:**

PaaS started here. Added IaaS later.

Operate groups of VM’s at once. Can’t nest groups, resource can be part of one group

No limits to resources. Can’t rename.

**IP addresses/User Groups:**

Routing tables can be created. VPN through S2S VPN. Or blackhole loops.

These have to be defined and not automatically done.

**Azure Compute:**

Web, worker, VM role.

App Service:

* web apps (shared/dedicated) managed. Any language, powershell, etc. continuous integration
* connect to other platforms, HA, security, application templates, Streamlined visual studio
* mobile apps (IOS, Android, HTML5). SSO. Azure AD FS. Build offline ready apps, push notifications, autoscale, staging environments, continuous integration
* api apps (migration of existing API). CORS (easy consumption, protocol to talk to each other over domains). Access controls, integration with logic apps, integration into visual studio.
* logic apps (build workflow logic). If this, then that. Zapier, IFTTT, can be designed in browser, templates

A0-A4 (extra small to extra large)

A5-A7 – Larger databases memory intensive

A8-A9 – network optimized (messaging)

A10-A11 – compute intensive.

D Series: SSD

* D1-D4 – websites
* D11-D14 – memory intensive

Dv2 Series:

* SSD
* D1\_v2- everyday applications
* D11\_v2-D15\_v2 – large database

Cloud service – PaaS. You can install other things on this VM. Remote in.

* Web role – public endpoint, IIS, HTTP & HTTPS
* Worker Role – No IIS, No public endpoints, computation/data management. Background jobs.
* Perform tasks. Communicate between messaging queues. Configuration set up with instances.
* Don’t create VM. VM is created by configuration file. Updates switchover instances.

**VM role:**

Multiple predefined images. Windows Server 2012 R2. Looks like a physical server. OS running in a VHD. 400 + options for Linux. Computers/applications.

VM sizing:

Basic vs standard. Compute intensive. Price is a factor.

A-series – general purpose. Optimized to run at high performance

D-series – designed for Performance. SSD, Faster processor and memory

DS-series – premium storage instead of temporary storage. Same pricing to Dv2

Dv2-series – more powerful. 33% faster. Premium storage (faster io access)

G-series – memory intensive application, faster processor and more disk space, 2 time memory, 4 times on storage. Premium storage. Graphic processor

Basic tier doesn’t have load balancing, etc.

**Availability set** – two or more VM’s running under a load balancer

**Fault domain (FD)** – rack of servers. Entire rack of servers fails the whole rack is affected

**Update domain** – define grouping of servers that can be updated as a set. Can be updated individually and doesn’t bring the application down

**VPN & Express Route:**

Connect to a network remotely. Encrypted traffic. Secure tunnel. Connects to offices together but needs a gateway on both ends

P2S (Point to Site) – dev on a database with only one person needing access. To be able to use it as it is local. 100 Mbps, 200 Mbps. Uses SSTP protocol.

S2S (Site to Site) – entire network can be connected to Azure. Hits the Gateway first. Speed Is limited. 100 Mbps. Performance sku only offers 200 Mbps. Outbound has cost associated with it. IPSEC or RRAS

ExpressRoute – More expensive. Doesn’t use public internet. Private fiber connection. IXP by ISP. A lot faster. 500 Mbps, 1 Gbps, 2 Gbps, 50,000 a month. Servers could be hosted at the IXP. Another is MPLS. Cost more with multi locations. Know the differences between P2S, S2S and Express Route

**Azure Services:**

Cisco CSR at the marketplace with 4vNIC. Apache, PHP, Linux options.

Know all the servers in terms of networking like VPN, S2S, Express Route.

Load Balancer: spreads the load out to other servers within the pool.

Round robin – cycling through the servers one by one. Monitors every 15 seconds.

Sticky session – same server responding to the user. Single point of failure

Vertical Scaling – increasing resources up.

DNS resolution points to the public IP of the LB.

Internal Load balancers – access services behind the scene. NAT with the internal load balancers to help security.

Application Gateway: Web based LB – HTTP/HTTPS. Firewall option. Cookie based affinity. SSL offloading. URL routing. Can have the same websites behind the LB. Application matched to more than 2 VM’s. LB operates at layer 4. Site behind the actual LB

Traffic Manager – same application can run in different regions across the globe. Helps reduce latency. DNS local. DNS points to traffic manager. No global DNS in this case. Refresh of information in the cache. Checks on the health of the locations. Couple minutes if a node fails to revert to the failover instance. Does support round robbin. Primary/Secondary. Global/region routing of traffic. Nested profiles allows you to point one top-level traffic manager profile to another traffic manager profile to combine two different load balancing methods.

**Azure Media Services:**

Ondemand streaming media services. Live events. Cloud based encoding. Upload video content and encode in multiple formats at scale like DRM and CDN.

**Azure CDN:**

VM to send static content that are closest to the user. Video/images/audio/css/js. Distribute files closest to the region as well. Increases performance of application by distribution CDN places static content closest to it’s users.

**Azure AD:**

Manages user access/identify management. Tying in corporate AD. Identity management of users. Multi-Factor like RSA.

**Microsoft Reddis Cache:**

High data store. Doesn’t have information stored in the database. Basic, Standard or Premium. Databases can be split into different databases. Temporary information or information that is searched through frequently. Sits in front of the database or in front of the Load balancer. You can also have the cache on the web server if needed.

**MultiFactor Authentication:**

Phone based authentication methods. SMS/call. SDK methods to build into your application.

**Azure Service Bus:**

Messaging platform to send messages between applications. Disconnected communication between applications like a cell phone.

Assignment 1 – create a virtual network, not public available. Add two basic web VM’s in available set. Configure point to site VPN.

**Secure Resources – Objective 2:**

AD as a secure option. AD Connect or DirSync

**Managed Identities:**

Azure AD. Multi-tenant cloud based identity. Keep users not seeing users from another company. SaaS model. IDMaaS – Identity Management as a Service

Handles 5 million organizations with this detail. 1 billion authentications every day. 1 Trillion authentications since Azure AD was put in place. To work with user applications. SSO and multi-platform support and multi-factor. 99.9% uptime.

Graph API:

Allow developrs to build web and mobile apps. REST API and can integrate with any API

CRUD, Customize, Read, User, Delete. JSON. All over HTTP. Need an authentication token. Query functions. Check changes between two time periods

OAUTH/OPENID – Opensource for authentication with untrusted clients. Trusted client would be a server. Untrusted is like a phone. Requires token. Can expire. Can be integrated into the backend network

**Hybrid Identities:**

Many environments have onpremise and in the cloud. Identity for all applications no matter where the infrastructure resides. Security, cost savings, legacy apps, requirements.

SAML claim. (SAML – Security Assertions Markup Langugage). Proof of their identity. Tokes are signed. First, requests SAML token. Issues token to client along with proof key. Provider signs token. Customer needs to sign the token via proof key. Application now trust its

Azure AD FS:

Federated identities. Turning over responsibility to a third party. AD FS passes it onto your AD. Avoids DirSync. Can keep all your identities in your managed premise.

AD Application proxy:

SSO for remote employees. Allowing access inside and outside your network. Not a vpn. Authenticates through the proxy and the proxy gets in your local network. Have to install a connector inside the network. All traffic goes through Azure. AADSync, DirSync, and Azure AD Connect are ways for synchronizing onpremise to Azure AD

Azure ACS:

Azure Access Control Service. Enable multiple identity providers. Identity provider choosen. Each provider has their own security token. Intermediary that does the work for you and deals with the external providers. Google, Facebook, Yahoo, OpenID, Azure AD, Twitter, Microsoft. Azure w/FS. App sends the user to authenticate. Provider responds with token. Client sends the token to ACS and coversits it. Sends it back to the app.

ASP.NET coding:

.NET Framework can provide the code to work with ACS but code is different for different providers. Active Directory B2C – cloud identity management solution for consumer facing web applications

Data Security:

NSG (Network Security Group) – virtual firewall. Define outbound and inbound rules. 5 elements – Protocol, Source IP, Source Port Range, Destination IP and Port. Can have multiple NIC’s or subnets. Each subnet can only have one NSG. Lower numbers take priority. Traffic within network, outbound and load balancer allows, deny all other traffic.

Data in transit:

SSL/TLS:HttpS must be used. Client side encryption. VPN is another option to encrypt traffic crossing the internet or S2S/P2S. ExpressRoute doesn’t travel the internet. Azure Rights Management – service that encrypts sensitive files and data. Provides ways to access that encrypted data for authorized and authenticated users.

Data at rest:

Azure SSE (Storage Service Encryption). Automatically encrypts data and decrypts. 256-bit AES. Block blobs, page, blobs and append blobs only works here. ARM only, not classic. Client side encryption. Encrypt disks used by VM’s.

Azure Operations Management Suite:

Security audits. Dashboard style interfaces. Helps to know a real operations center can map VM’s too. Visualization tool. Detect threats, monitor network, mitigate damage.

RBAC:

Don’t give everyone admin access. Abstraction of access control. Owner, contributor, reader standard roles. Can have custom roles. Subscription can only belong to one AAD. Same for resource groups and resource’s belong to only one group. There are limits to at the resource level. Contributor can’t change security but everything else. Inheritance does exist. Can’t give access to others.

Assignment #2 – Create a user, assign that user to a role, and grant that role access to a resource. Datacenters are no longer a requirement on the test.

**Azure Data Storage**

Table Storage:

NoSQL, no schema, structured, not RBDMS, name value pair metaphor. Rows are entities and can have up to 255 properties. Partition Key and RowKey are primary key. Row key is like an id key. Timestamp is mandatory. Paritiion key is like a table name. No indexes. Entities can have different numbers of properties. Azure sorts the table and automatically moves partitions.

SQL Storage:

Relational. Elastic database pools can allow you to provision tables as needed like autoscale. Hosted by Azure. Pay per use. Geo-replicated as an option. 99.99% SLA.

Document DB:

NoSQL model. JSON. Data is stored in text format. Performance tiers for pricing. $6 per month. 25-100 dollars per month.

Blob Storage:

Container model and have security. Binary large object. Files, videos, photos, backup images, etc. 500 TB of storage capacity.

Block – optimized for streaming and storing objects in the cloud. Images, videos, documents, etc.

Append blob – optimized for appending. Log files.

Page blob – random writes. Harddrive writes. VHD are stored as page blob.

Queue Storage:

Messaging between services that is persistent. Maximum 64KB in size. Read and delete. Temporary. Invisibility behavior. Once read becomes invisible for a short time. If not deleted, visible again.

File Storage:

Network shared with SMB 3.0. Rest API. Application like storing log or backups. Helps migrate legacy systems. All other storage types have the 500 TB limit. File share has 5 TB limit on it. Can create directories inside the share. Each file can be 1 TB.

SQL Server VM:

PaaS or IaaS to be consumed. VM provisioned and install software. Need a license. Choose a template and edition. Possible data migration issues. Disaster recovery issues – can backup to a blob in Azure.

Securing SQL Server:

Ensure the right people have access, etc. owner can have access at anytime. Public access given to read the contents. Authorization can be done through ACL. White list and Encryption. Lock your database down. Data can be encrypted at rest SSE. **Shared Access Signature (SAS)** – give a temporary access to a storage asset that expires. **Shared Access Policy (SAP)** – can be given to users and revoke as needed.

**Azure Mobile Apps**

Azure App Service like fully managed PaaS. Supports data storing, authentication, push notifications, and offline sync. Web, API, logic, mobile. Mobile Apps were called Mobile Services. It can autoscale, staging environments, continuous deployment. Virtual networking, isolated environments.

Cross Platforms:

Windows, IOS, Android, JavaScript, Apache Cordova, Xamarin using C#. Microsoft does provide SDK’s for each platform. You would need new Mobile app, new SQL database, server code, and quick start application.

Offline Sync:

Allows mobile clients to store and retrieve data locally until Internet is online. Improve responsiveness, operate Wi-Fi only devices without a SIM card. Sync data between multiple devices and handle conflicts. Limit network on metered connections. Can use app while offline.

Done through a local store or storage location on your devices. Xamarin is SQLlite. API’s can be used to sync table API’s. Application must push changes up to the network for all outstanding changes. Data is pulled from the network. Implicit push(outstanding changes in the local store by changes) and incremental pull to retrieve new data like Github.

Extended Mobile Apps:

SDK for .NET and Node.js and REST API for PHP, Java, Python. Exposes a new endpoint for your code that it can call. Example – custom authentication code, database sources, web services, use queues, and connecting to legacy systems.

Mobile Security:

Application security and infrastructure and platform security. 24 hour threat management, encryption using communicating over networks. App service apps are isolated from the internet. Application security – developer must ensure operates security. Inputs are sanitized, don’t trust requests, check safeguards, and session hijacking.

Penetration testing must be done when public facing components are added to their networks. Software available.

**Notification Hub:**

Mobile application with notifications can be super important. Mobile systems have to hook into existing apple/google notifications. App hits the platform notification service via token and goes back to the app-back-end and then to the PNS. One API call. Multiple users. Through any platform via Azure notification hub. Support localization.

Notification hub fits in between appback-end to PNS(platform notification service)

Tagging:

Segment users within the notification hub. Level of service they are entitled too.

Secure Message:

Like with a bank app. Tries data and sends data, etc.

Offline Sync:

Wake the app and update itself.

**WEB API**

Developing websites with an API on .NET framework. API – way for two systems to communicate using a standard, predefined language. WEB API – way for two systems to communicate with each other over a nework.

Can be hosted in Azure. RESTful API.

HTTP request via IIS. WEB API runs on MVC model. Routing analyzes request, chooses a controller and an action and invokes it.

Controller verifies that the request is authorized

Populates a model with the results of the request

Data is organized instead of just HTML< and sent back to caller.

Can be created in Visual Studio. PaaS or IaaS

Scaling:

* Option of increasing hosting plan (Free, Basic, Standard, Premium)
* Instance size and instance count

Web Jobs:

* Designed to be long-running code or a background task without a user interface
* Runs continuously, on demand, or predefined schedule.
* Can use SDK with built-in features
* Open-source repository access with other people written.
* Example taking 1500 files from retail location to process taking 20 hours
* Files supported- exe, cmd/bat, ps1, sh, php, py, js

Secure Web API:

* Communicate over public internet via HTTP
* More powerful with direct access to DB.
* Deletion can screw things up.
* Code can be added to request authentication/authorization. Google analytics is Javascript based. API requires backend work.
* Monitoring and logging for API is important. API is server side programming.
* Don’t create your own security.

**Hybrid Apps**

Service Bus:

Design hybrid application with design strategies. Onprem and services running in azure.

Security may not allow some services to run in the cloud. External use might be rare.

Service bus relay allows apps to connect to your on premises service. WCF accepts the calls and passes the WCF service running in the corporate network. Allows you to be selective on the service that is running into your network.

BizTalk API:

Can managed hybrid connections built in. web apps can use cany language

Applications can be moved from local network to cloud without changing connection strings. No connection string required.

**Azure Media Services**

Design a media service. Media watched online everyday. Supports online streaming videos bot live and pre-recorded

On PaaS. Signal goes into Media services for live encoding. SDK’s available. (RTMP & RTP). Supports Preview Monitoring. Media indexer including closed captions and transcripts. Checks quality of stream. Archiving and storage for records. Supports DRM and AES.

Assignment #3 – Create a storage account and a container. Create a table using Table Storage. Write a small .NET program that uses Table Storage.

**Introduction to Advanced Applications**

Design an advanced application

Compute Intensive:

Need more resources. You can add them. Large job broken into smaller tasks. SETI@home. Use cases – disease research, molecular simulation, mathematic problems, etc. Job that can only be handled serially not scale. Speed of processor is the speed of that task. Split the processor to handle the jobs

Manage, scale, elastic, pay for use.

HPC pack:

* meant originally for on premises.
* Hybrid
* IaaS
* PaaS (Azure Batch)

Head node versus compute:

* Head node controls the distribution of jobs
* Tasks happen on cluster
* Head node can be hosted on prem

Azure Batch:

* Job scheduling as a service
* App lifecycle management
* Management and scheduling
* Budget, quotas and limits.

VM Options:

* Compute intensive instances (A,D)
  + A8 - 8 cores, 56 GB memory, 2 network cards
  + A9 – 16 cores, 112GB memory, 2 network cards
  + A10 – 8 cores, 56GB memory, 1 network cards
  + A11 – 16 cores, 112GB memory, 1 network card

Azure Premium Storage (AS, DS)

* Lift and shift of existing applications.
* Sustainable high-performance I/O using SSD
* Attach up to 32TB of data disks to DS series
* Attach up to 64 TB of data disks to G series

**Long Running applications**

THGTTG – 42 71/2 – 10 million years

Jobs take days even weeks to run

Machine can have hardware problems. Don’t want to lose hours and application can throw errors. Indication of progress.

Concepts of availability:

* Requires extra consideration to availability
* Microsoft SLAs
* Using multiple instances
* Stateless vs State-ful
* Avoid single points of failure
* Deploy across regions

Concept of reliability:

* Fault domains and update domains
* Handling errors (retries)
* Disconnected applications (loose coupling)
* Application health monitoring

Concepts of scaling:

* Planned scaling vs reactive scaling
* Scaling up vs scaling out
* Breaking up workload to smaller parts

Long-running applications

* Partition the workload, similar to compute-intensive
* Health monitoring or Azure Application Insights
* Logging
* Retry on errors
* Gracefully handle restarts without corruption
* Consider check pointing into storage from time to time. (snapshot almost)
* Is\_singleton:true config file for single instance to run.

Which of the following Azure App Services is best to handle long-running applications?

Azure WebJobs

**Storage Options**

* Traditional SQL databases have challenges around scaling.
* Parititioning and sharding the database is an architectural challenge
* Best to consider storage options when performance or scalability is needed
* NoSQL options – designed for large applications like Twitter and Facebook
* Temporary storage vs persistent storage.
* Consistency issues and locking (dirty read, etc)
* Data sorting and access, indexing
* Latency – length of time for message to get from compute instance to storage instance
* Cost issues

**Azure Integration**

* Azure AD
* Azure Storage
* Message Queues
* Service Bus
* Notification Services
* Focus on the following:
  + Azure Machine Learning
  + Big Data
  + Azure Media Services
  + Azure Search Services

Machine Learning:

* New branch of computer science
* Examines large amounts of data to detect patterns
* Generates code to let you recognize those patterns in new data.
* Make better predictions
* Cloud service that allows you to create jobs for these predictive analytics
* Need a data expert and more data the better

Big Data:

* Name given to the branch of computer science devoted to storing and analyzing massive amounts of data.
* SQL Server would choke
* Traditional methods don’t work
* Data accumulates and appears to be accelerating
* HDInsight service for analyzing big Data
  + Apache Hadoop clusters in the cloud
  + Provides framework for big data operations
    - Manage
    - Analyze
    - Report

Media Services:

* Scalable live media streaming or on-demand
* Securely upload and transcode video to streaming format.
* Scaling, copyright protection

Azure Search:

* PaaS – search as a service
* Managed by Microsoft
* Service to populate your data
* Integrate search into your mobile and web apps
* REST API or .NET SDK
* Support multiple languages
* Simple query syntax or lucene syntax (Lucene has to be hosted itself)
* Suggestions
* Highlight matches
* Facets and filters

**Azure Web Apps**

Scalability of Web Apps24 regions plus 100+ and 10 geos.

Scaling/Caching options:

* CDN (Content Delivery Network) – delivery static files from own network of servers to somewhere else
* Traffic Manager – direct traffic to closest regional DC by DNS request. DNS is cached.
* Traffic Manager Failover
* Choose the right data source – Redis
* Autoscale

Develop Web Apps:

SDK’s for .NET. Azure SDK requires data storage like SQL

Debug Web Apps:

**Turn off customerrors in web.config**

Publish debug version which includes symbols

Attach IDE to Azure server

Remote debugging can sometimes time out.

Azure disables the remote debug after 48 hours for security reasons

Web apps can keep logs

* Application tracing
* Web server log
* Detailed error log
* Failed request tracing log

Languages - .NET, Java, PHP, Ruby, Python, Node.js, Powershell. Tutorials for creating stuff in Azure

Web Options:

* Web Apps
  + Web Apps are pure PaaS, package code and config
  + Can autoscale and multiple instances, auto lb
* Cloud Service
  + PaaS
  + Longer to deploy and VM is created for you
  + Re-sizing can cause downtime
  + Remote desktop access
  + Alter startup scripts
  + Install software
  + Azure takes care of the OS
* VM
  + IaaS
  + VHD in Azure Storage
  + Easier to migrate
  + Responsible for everything
  + Deployements, LB, scaling yourself

Site Extensions:

* 59 site extensions
* Add custom blocks
* Examples
  + Phpmyadmin
  + Monitor storage account
* Can create your own – [www.siteextensions.net/packages](http://www.siteextensions.net/packages).

Packages:

* Preparing your code for deployment – called publishing
* FTP or FTPS can use to upload code or manually
* Kudu – opensource – repo into azure
* Azure web site has kudu companion with it. Git can clone it
* Web deploy – built in VS
  + Pre-complies and deploys complied files

Hosting Plans:

* Web Apps are hosted on App Service Plans
  + Limits and prices
    - Number of apps
    - Disk space
    - SLA
    - Auto-scale features
* Can group apps together
* Apps in same region can share a plan
* Apps run under one instance
* An app can only be part of a hosting plan
* Best to separate plans for dev, test, production
* Apps can be moved to new plans

Resource Groups:

* Apps/plans belong to resource groups
* Resource groups are restricted to a region
* Separate projects for different resource groups

Deployment Slots:

* Staging can be done in a separate place.
* You can flip a switch. Traffic is just changed
* DNS level change
* You can decide route traffic by percentage with a second slot if needed – A/B testing

**Know Deployment Slots/Extensions**

**Business Continuity**

Concept in case a disaster happens you can recover quickly with minimal downtime. Involve’s trade-offs. Designed for resilience. It handles its own failures. You’ll still need to architect your own solution to protect against known failure points.

**Scaling up** – larger hardware, more memory, bigger CPU

**Scaling out** – multiple instances of a web app

Autoscale plans

Basic plan – up to 3

Standard plan up to 10, autoscale and grs

Premium up to 50, autoscale and grs

**Elastic scale** – sql database has a scaling feature

**Sharding** – divide load to multiple instances

Global Scale:

* Consider distributed loads to other locales.
* CDN
* Move apps to other regions and geos
* Traffic manager to direct to the quickest responding site

Data Replication:

* Make sure SQL databases are distributed to other regions.
* **SQL Sync** – not replication but sync
  + Making sure transaction on primary happens on secondary
* **SQL geo-replication – for disaster recovery**
  + Standard tier, offline secondary (read-only)
    - Being retired April 2017
  + Premium tier, Active Geo-Replication
    - Up to 4 copies in the same region or others
  + Primary is always ahead, the master
  + Secondary is a bit behind.

Disaster Recovery:

Minimal downtime:

* Use deployment slots
* Use multiple instances, configured for Upgrade Domains
* Automated backups are on the standard and premium tiers
  + Standard has 2 automated backups a day
  + Premium has 50 automated backups a day
* Backups are stored as blob’s within a container

High Availability

* Must be designed in. DR
* 3-4’9’s
* Properties
  + Availability
  + Scalability
  + Fault tolerance – does it try again? Retry errors. No dependency so it doesn’t just fail
* DR must be tested and it must be designed not just happen.
* RTO (Recovery Time Objective) – maximum amount of time to restore application functionality
* Recovery Point Objective (RPO) – acceptable time window of data lost

ARM(Azure Resource Manager) Template:

Deployment script and guarantee those resources will be deployed as specified.

Templates place to pull configurations from within Microsoft’s website

**Azure templates in github**

You can also export ARM templates from resources that are already in your account in the deployment automation resource group

Assignment #5

Create a Hellow World web App, Add a deployment slot, swap the deployment and prove it works.