70-532 Syllabus

The following the is syllabus for 70-532 exam.

1. Create and manage Azure Resource Manager Virtual Machines (30−35%)

- Deploy workloads on Azure Resource Manager (ARM) Virtual Machines (VMs)
    - Identify workloads that can and cannot be deployed; run workloads, including Microsoft and Linux; create VMs
- Perform configuration management
    - Automate configuration management by using PowerShell Desired State Configuration and VM Agent (custom script extensions); configure VMs using a configuration management tool, such as Puppet or Chef; enable remote debugging
- Configure ARM VM networking
    - Configure static IP addresses, Network Security Groups (NSG), DNS, User Defined Routes (UDRs), external and internal load balancing with HTTP and TCP health probes, public IPs, firewall rules and direct server return; design and implement Application Gateway
- Scale ARM VMs
    - Scale up and scale down VM sizes, deploy ARM VM Scale Sets (VMSS), configure ARM VMSS auto-scale
- Design and implement ARM VM storage
    - Configure disk caching, plan for storage capacity, configure shared storage using Azure File service, configure geo-replication, implement ARM VMs with Standard and Premium Storage
- Monitor ARM VMs
    - Configure ARM VM monitoring, configure alerts, configure diagnostic and monitoring storage location
- Manage ARM VM availability
    - Configure multiple ARM VMs in an availability set for redundancy, configure each application tier into separate availability sets, combine the Load Balancer with availability sets

2. Design and implement a storage and data strategy (25−30%)

- Implement Azure Storage blobs and Azure files
    - Read data, change data, set metadata on a container, store data using block and page blobs, stream data using blobs, access blobs securely, implement async blob copy, configure Content Delivery Network (CDN), design blob hierarchies, configure custom domains, scale blob storage
- Implement Azure storage tables and queues
    - Implement CRUD with and without transactions; design and manage partitions; query using OData, scale tables and partitions, add and process messages, retrieve a batch of messages, scale queues
- Manage access and monitor storage
    - Generate shared access signatures, including client renewal and data validation; create stored access policies; regenerate storage account keys;

configure and use Cross-Origin Resource Sharing (CORS); set retention policies and logging levels; analyse logs

- Implement Azure SQL Databases
    - Choose the appropriate database tier and performance level, configure and perform point-in-time recovery, enable geo-replication, import and export data and schema, scale Azure SQL databases
- Implement Azure DocumentDB
    - Create databases and collections, query documents, run DocumentDB queries
- Implement Redis caching
    - Choose a cache tier, implement data persistence, implement security and network isolation, tune cluster performance
- Implement Azure Search
    - Create a service index, add data, search an index, handle search results

3. Manage identity, application and network services (15−20%)

- Integrate an app with Azure Active Directory (Azure AD)
    - Develop apps that use WS-federation, OAuth, and SAML-P endpoints; query the directory using Graph API
- Design and implement a communication strategy
    - Implement hybrid connections to access data sources on-premises, leverage site-to-site (S2S) VPN and ExpressRoute to connect to an on-premises infrastructure
- Design and implement a messaging strategy
    - Develop and scale messaging solutions using service bus queues, topics, relays and notification hubs; monitor service bus queues, topics, relays, event hubs and notification hubs
- Develop apps that use Azure AD B2C and Azure AD B2B
    - Design and implement .NET MVC, Web API and Windows desktop apps that leverage social identity provider authentication, including Microsoft account, Facebook, Google+, Amazon and LinkedIn; leverage Azure AD B2B to design and implement applications that support partner-managed identities

4. Design and implement Azure PaaS compute and web and mobile services (25–30%)

- Design Azure App Service Web Apps
    - Define and manage App Service plans; configure Web Apps settings, certificates and custom domains; manage Web Apps by using the API, Azure PowerShell and Xplat-CLI; implement diagnostics, monitoring and analytics; implement web jobs; design and configure Web Apps for scale and resilience
- Implement Azure Functions
    - Create Azure Functions, implement a webhook function, create an event processing function, implement an Azure-connected function
- Implement API management
    - Create managed APIs, configure API management policies, protect APIs with rate limits, add caching to improve performance, monitor APIs, customise the Developer Portal

- Design Azure App Service API Apps
    - Create and deploy API Apps, automate API discovery by using Swashbuckle, use Swagger API metadata to generate client code for an API app, monitor API Apps
- Develop Azure App Service Logic Apps
    - Create a Logic App connecting SaaS services, create a Logic App with B2B capabilities, create a Logic App with XML capabilities, trigger a Logic App from another app, create custom and long-running actions, monitor Logic Apps
- Develop Azure App Service Mobile Apps
    - Create a Mobile App, add offline sync to a Mobile App, add authentication to a Mobile App, add push notifications to a Mobile App
- Design and implement Azure Service Fabric apps
    - Create a Service Fabric application; build an Actors-based service; add a web front end to a Service Fabric application; monitor and diagnose services; migrate apps from cloud services; create, secure, upgrade and scale Service Fabric Cluster in Azure; scale a Service Fabric app