TechyEdz Solutions

Training | Consulting | Developement | Outsourcing

Azure Cloud Engineer Masters Program

9032803832

contact@techyedz.com

9032803832

www.techyedz.com

Azure Cloud Engineer Masters Program

Course Overview:

Microsoft Azure Cloud Engineer Masters Program will prepare you for three certification exams: AZ-104, which is required to attain Azure Administrator Badge (Associate-level), AZ-303, which is required to attain Azure Solutions Architect Badge (Expert-level) and AZ-400 which is required to attain Azure DevOps Badge. You will be able to implement advanced networking configurations, plan authentication and security of the infrastructure, use IaaS solutions and Storage Services to deploy end-to-end cloud solutions.

Azure Administrator (AZ -104)

Course Outline:

Manage Azure identities and governance

1. Manage Azure AD objects

- create users and groups
- manage user and group properties
- manage device settings
- perform bulk user updates
- manage guest accounts
- configure Azure AD Join
- configure self-service password reset
- NOT: Azure AD Connect; PIM

2. Manage role-based access control (RBAC)

- create a custom role
- provide access to Azure resources by assigning roles
  - subscriptions
  - resource groups
  - resources (VM, disk, etc.)
- interpret access assignments
- manage multiple directories

3. Manage subscriptions and governance

- configure Azure policies
- configure resource locks
- apply tags
- create and manage resource groups
    - move resources
    - remove RGs
- manage subscriptions
- configure Cost Management
- configure management groups

Implement and manage storage

1. Manage storage accounts

- configure network access to storage accounts
- create and configure storage accounts
- generate shared access signature
- manage access keys
- implement Azure storage replication
- configure Azure AD Authentication for a storage account

2. Manage data in Azure Storage

- export from Azure job
- import into Azure job
- install and use Azure Storage Explorer
- copy data by using AZCopy

3. Configure Azure files and Azure blob storage

- create an Azure file share
- create and configure Azure File Sync service
- configure Azure blob storage
- configure storage tiers for Azure blobs

Deploy and manage Azure compute resources

1. Configure VMs for high availability and scalability

- configure high availability

- deploy and configure scale sets

2. Automate deployment and configuration of VMs

- modify Azure Resource Manager (ARM) template
- configure VHD template
- deploy from template
- save a deployment as an ARM template
- automate configuration management by using custom script extensions

3. Create and configure VMs

- configure Azure Disk Encryption
- move VMs from one resource group to another
- manage VM sizes
- add data discs
- configure networking
- redeploy VMs

4. Create and configure containers

- create and configure Azure Kubernetes Service (AKS)
- create and configure Azure Container Instances (ACI)
- NOT: selecting an container solution architecture or product; container registry settings

5. Create and configure Web Apps

- create and configure App Service
- create and configure App Service Plans
- NOT: Azure Functions; Logic Apps; Event Grid

Configure and manage virtual networking

1. Implement and manage virtual networking

- create and configure VNET peering
- configure private and public IP addresses, network routes, network interface, subnets, and virtual network

2. Configure name resolution

- configure Azure DNS
- configure custom DNS settings

- configure a private or public DNS zone

3. Secure access to virtual networks

- create security rules
- associate an NSG to a subnet or network interface
- evaluate effective security rules
- deploy and configure Azure Firewall
- deploy and configure Azure Bastion Service
- NOT: Implement Application Security Groups; DDoS

4. Configure load balancing

- configure Application Gateway
- configure an internal load balancer
- configure load balancing rules
- configure a public load balancer
- troubleshoot load balancing
- NOT: Traffic Manager and FrontDoor and PrivateLink

5. Monitor and troubleshoot virtual networking

- monitor on-premises connectivity
- use Network Performance Monitor
- use Network Watcher
- troubleshoot external networking
- troubleshoot virtual network connectivity

6. Integrate an on-premises network with an Azure virtual network

- create and configure Azure VPN Gateway
- create and configure VPNs
- configure ExpressRoute
- configure Azure Virtual WAN

Monitor and back up Azure resources

1. Monitor resources by using Azure Monitor

- configure and interpret metrics

- analyze metrics across subscriptions
- configure Log Analytics
  - implement a Log Analytics workspace
  - configure diagnostic settings
- query and analyze logs
  - create a query
  - save a query to the dashboard
  - interpret graphs
- set up alerts and actions
  - create and test alerts
  - create action groups
  - view alerts in Azure Monitor
  - analyze alerts across subscriptions
- configure Application Insights
- NOT: Network monitoring

2. Implement backup and recovery

- configure and review backup reports
- perform backup and restore operations by using Azure Backup Service
- create a Recovery Services Vault
  - use soft delete to recover Azure VMs
- create and configure backup policy
- perform site-to-site recovery by using Azure Site Recovery
- NOT: SQL or HANA

Azure - Microsoft Azure Architect Technologies (AZ-303)

Course Outline:

Implement and Monitor an Azure Infrastructure

1. Implement cloud infrastructure monitoring

- monitor security
- monitor performance
  - configure diagnostic settings on resources
  - create a performance baseline for resources
  - monitor for unused resources
  - monitor performance capacity
  - visualize diagnostics data using Azure Monitor
- monitor health and availability

    - monitor networking
    - monitor service health
- monitor cost
    - monitor spend
    - report on spend
- configure advanced logging
    - implement and configure Azure Monitor insights, including App Insights, Networks, Containers
    - configure a Log Analytics workspace
- configure logging for workloads
    - initiate automated responses by using Action Groups
- configure and manage advanced alerts
    - collect alerts and metrics across multiple subscriptions
    - view Alerts in Azure Monitor logs

2. Implement storage accounts

- select storage account options based on a use case
- configure Azure Files and blob storage
- configure network access to the storage account
- implement Shared Access Signatures and access policies
- implement Azure AD authentication for storage
- manage access keys
- implement Azure storage replication
- implement Azure storage account failover

3. Implement VMs for Windows and Linux

- configure High Availability
- configure storage for VMs
- select virtual machine size
- implement Azure Dedicated Hosts
- deploy and configure scale sets
- configure Azure Disk Encryption

4. Automate deployment and configuration of resources

- save a deployment as an Azure Resource Manager template
- modify Azure Resource Manager template
- evaluate location of new resources
- configure a virtual disk template
- deploy from a template
- manage a template library
- create and execute an automation runbook

5. Implement virtual networking

- implement VNet to VNet connections
- implement VNet peering

6. Implement Azure Active Directory

- add custom domains
- configure Azure AD Identity Protection
- implement self-service password reset
- implement Conditional Access including MFA
- configure user accounts for MFA
- configure fraud alerts
- configure bypass options
- configure Trusted IPs
- configure verification methods
- implement and manage guest accounts
- manage multiple directories

7. Implement and manage hybrid identities

- install and configure Azure AD Connect
- identity synchronization options
- configure and manage password sync and password writeback
- configure single sign-on
- use Azure AD Connect Health

Implement Management and Security Solutions

1. Manage workloads in Azure

- migrate workloads using Azure Migrate
  - assess infrastructure
  - select a migration method
  - prepare the on-premises for migration
  - recommend target infrastructure
- implement Azure Backup for VMs
- implement disaster recovery
- implement Azure Update Management

2. Implement load balancing and network security

- implement Azure Load Balancer

- implement an application gateway
- implement a Web Application Firewall
- implement Azure Firewall
- implement the Azure Front Door Service
- implement Azure Traffic Manager
- implement Network Security Groups and Application Security Groups
- implement Bastion

3. Implement and manage Azure governance solutions

- create and manage hierarchical structure that contains management groups, subscriptions and resource groups
- assign RBAC roles
- create a custom RBAC role
- configure access to Azure resources by assigning roles
- configure management access to Azure
- interpret effective permissions
- set up and perform an access review
- implement and configure an Azure Policy
- implement and configure an Azure Blueprint

4. Manage security for applications

- implement and configure KeyVault
- implement and configure Azure AD Managed Identities
- register and manage applications in Azure AD

Implement Solutions for Apps

1. Implement an application infrastructure

- create and configure Azure App Service
- create an App Service Web App for Containers
- create and configure an App Service plan
- configure an App Service
- configure networking for an App Service
- create and manage deployment slots
- implement Logic Apps
- implement Azure Functions

2. Implement container-based applications

- create a container image

- configure Azure Kubernetes Service
- publish and automate image deployment to the Azure Container Registry
- publish a solution on an Azure Container Instance

Implement and Manage Data Platforms

1. Implement NoSQL databases

- configure storage account tables
- select appropriate CosmosDB APIs
- set up replicas in CosmosDB

2. Implement Azure SQL databases

- configure Azure SQL database settings
- implement Azure SQL Database managed instances
- configure HA for an Azure SQL database
- publish an Azure SQL database

Azure - Designing and Implementing Microsoft DevOps Solutions (AZ-400)

Course Outline:

Develop an Instrumentation Strategy

1. Design and implement logging

- assess and Configure a log framework
- design a log aggregation and storage strategy (e.g. Azure storage)
- design a log aggregation using Azure Monitor
- manage access control to logs (workspace-centric/resource-centric)
- integrate crash analytics (App Center Crashes, Crashlytics)

2. Design and implement telemetry

- design and implement distributed tracing

- inspect application performance indicators
- inspect infrastructure performance indicators
- define and measure key metrics (CPU, memory, disk, network)
- implement alerts on key metrics (email, SMS, webhooks, Teams/Slack)
- integrate user analytics (e.g. Application Insights funnels, Visual Studio App Center, TestFlight, Google Analytics)

3. Integrate logging and monitoring solutions

- configure and integrate container monitoring (Azure Monitor, Prometheus, etc.)
- configure and integrate with monitoring tools (Azure Monitor Application Insights, Dynatrace, New Relic, Naggios, Zabbix)
- create feedback loop from platform monitoring tools (e.g. Azure Diagnostics VM extensions, Azure Platform Logs, Event Grid)
- manage Access control to the monitoring platform

Develop a Site Reliability Engineering (SRE) strategy

1. Develop an actionable alerting strategy

- identify and recommend metrics on which to base alerts
- implement alerts using appropriate metrics
- implement alerts based on appropriate log messages
- implement alerts based on application health checks
- analyze combinations of metrics
- develop communication mechanism to notify users of degraded systems
- implement alerts for self-healing activities (e.g. scaling, failovers)

2. Design a failure prediction strategy

- analyze behavior of system with regards to load and failure conditions
- calculate when a system will fail under various conditions
- measure baseline metrics for system
- recommend the appropriate tools for a failure prediction strategy

3. Design and implement a health check

- analyze system dependencies to determine which dependency should be included in health check
- calculate healthy response timeouts based on SLO for the service
- design approach for partial health situations
- integrate health check with compute environment

- implement different types of health checks (liveness, startup, shutdown)

Develop a security and compliance plan

1. Design an authentication and authorization strategy

- design an access solution (Azure AD Privileged Identity Management (PIM), Azure AD Conditional Access, MFA)
- organize the team using Azure AD groups
- implement Service Principals and Managed Identity
- configure service connections

2. Design a sensitive information management strategy

- evaluate and configure vault solution (Azure Key Vault, Hashicorp Vault)
- generate security certificates
- design a secrets storage and retrieval strategy
- formulate a plan for deploying secret files as part of a release

3. Develop security and compliance

- automate dependencies scanning for security (container scanning, OWASP)
- automate dependencies scanning for compliance (licenses: MIT, GPL)
- assess and report risks
- design a source code compliance solution (e.g. GitHub security, pipeline-based scans, Git hooks, SonarQube)

4. Design governance enforcement mechanisms

- implement Azure policies to enforce organizational requirements
- implement container scanning (e.g. static scanning, malware, crypto mining)
- design and implement Azure Container Registry Tasks (eg. Azure Policy)
- design break-the-glass strategy for responding to security incidents

Manage source control

1. Develop a modern source control strategy

- integrate/migrate disparate source control systems (e.g. GitHub, Azure Repos)
- design authentication strategies
- design approach for managing large binary files (e.g. Git LFS)

- design approach for cross repository sharing (e.g. Git sub-modules, packages)
- implement workflow hooks

2. Plan and implement branching strategies for the source code

- define Pull Requests (PR) guidelines to enforce work item correlation
- implement branch merging restrictions (e.g. branch policies, branch protections, manual, etc.)
- define branch strategy (e.g. trunk based, feature branch, release branch, GitHub flow)
- design and implement a PR workflow (code reviews, approvals)
- enforce static code analysis for code-quality consistency on PR

3. Configure repositories

- configure permissions in the source control repository
- organize the repository with git-tags
- plan for handling oversized repositories
- plan for content recovery in all repository states
- purge data from source control

4. Integrate source control with tools

- integrate GitHub with DevOps pipelines
- integrate GitHub with identity management solutions (Azure AD)
- design for GitOps
- design for ChatOps
- integrate source control artifacts for human consumption (e.g. Git changelog)

Facilitate communication and collaboration

1. Communicate deployment and release information with business stakeholders

- create dashboards combining boards, pipelines (custom dashboards on Azure DevOps)
- design a cost management communication strategy
- integrate release pipeline with work item tracking (e.g. AZ DevOps, Jira)
- integrate GitHub as repository with Azure Boards
- communicate user analytics

2. Generate DevOps process documentation

- design onboarding process for new employees
- assess and document external dependencies (e.g. integrations, packages)
- assess and document artifacts (version, release notes)

3. Automate communication with team members

- integrate monitoring tools with communication platforms (e.g. Teams, Slack, dashboards)
- notify stakeholders about key metrics, alerts, severity using communication platforms (e.g. Email, SMS, Slack, Teams)
- integrate build and release with communication platforms (e.g. build fails, release fails)
- Define and implement continuous integration

4. Design build automation

- integrate the build pipeline with external tools (e.g., Dependency and security scanning, Code coverage)
- implement quality gates (e.g. code coverage, internationalization, peer review)
- design a testing strategy (e.g. integration, load, fuzz, API, chaos)
- integrate multiple tools (e.g. GitHub Actions, Azure Pipeline, Jenkins)

5. Design a package management strategy

- recommend package management tools (e.g. GitHub Packages, Azure Artifacts, Azure Automation Runbooks Gallery, Nuget, Jfrog, Artifactory)
- design an Azure Artifacts implementation including linked feeds
- design versioning strategy for code assets (e.g. SemVer, date based)
- plan for assessing and updating and reporting package dependencies (GitHub Automated Security Updates, NuKeeper, GreenKeeper)
- design a versioning strategy for packages (e.g. SemVer, date based)
- design a versioning strategy for deployment artifacts

6. Design an application infrastructure management strategy

- assess a configuration management mechanism for application infrastructure
- define and enforce desired state configuration for environments

7. Implement a build strategy

- design and implement build agent infrastructure (include cost, tool selection, licenses, maintainability)

- develop and implement build trigger rules
- develop build pipelines
- design build orchestration (products that are composed of multiple builds)
- integrate configuration into build process
- develop complex build scenarios (e.g. containerized agents, hybrid, GPU)

8. Maintain build strategy

- monitor pipeline health (failure rate, duration, flaky tests)
- optimize build (cost, time, performance, reliability)
- analyze CI load to determine build agent configuration and capacity
- manage pipeline health
- identify the number of agents and jobs to run in parallel
- investigate test failures

9. Design a process for standardizing builds across organization

- manage self-hosted build agents (VM templates, containerization, etc.)
- create reuseable build subsystems (YAML templates, Task Groups, Variable Groups, etc.)

Define and implement a continuous delivery and release management strategy

1. Develop deployment scripts and templates

- recommend a deployment solution (e.g. GitHub Actions, Azure Pipelines, Jenkins, CircleCI, etc.)
- design and implement Infrastructure as code (ARM, Terraform, PowerShell, CLI)
- develop application deployment process (container, binary, scripts)
- develop database deployment process (migrations, data movement, ETL)
- integrate configuration management as part of the release process
- develop complex deployments (IoT, Azure IoT Edge, mobile, App Center, DR, multi- region, CDN, sovereign cloud, Azure Stack, etc.)

2. Implement an orchestration automation solution

- combine release targets depending on release deliverable (e.g., Infrastructure, code, assets, etc.)
- design the release pipeline to ensure reliable order of dependency deployments
- organize shared release configurations and process (YAML templates, variable groups)
- design and implement release gates and approval processes

3. Plan the deployment environment strategy

- design a release strategy (blue/green, canary, ring)
- implement the release strategy (using deployment slots, load balancer configurations, Azure Traffic Manager, feature toggle, etc.)
- select the appropriate desired state solution for a deployment environment (PowerShell DSC, Chef, Puppet, etc.)
- plan for minimizing downtime during deployments (VIP Swap, Load balancer, rolling
    - deployments, etc.)
- design a hotfix path plan for responding to high priority code fixes

Prerequisites:

- There are no prerequisites for enrollment in this Microsoft Azure Masters Program. Basic understanding of cloud computing concepts and Linux will help.

Who Should Attend:

- While the Microsoft Azure Cloud Engineer Masters Program is designed for professionals who are planning to attain any or all of the three badges – Azure Administrator Associate Badge, Azure Architect Expert Badge and Azure DevOps Badge.

Number of Hours: 100hrs

Certification: AZ-104, AZ-303 & AZ-400

Key Features:

- One to One Training
- Online Training
- Fastrack & Normal Track
- Resume Modification
- Mock Interviews
- Video Tutorials
- Materials
- Real Time Projects

- Virtual Live Experience
- Preparing for Certification