EDUSUM

#1 Online Certification Guide

MICROSOFT AZ-400

Microsoft DevOps Engineer Certification Questions & Answers

Exam Summary – Syllabus –Questions

AZ-400

Microsoft Certified - DevOps Engineer Expert

40-60 Questions Exam – 700/1000 Cut Score – Duration of 150 minutes

Table of Contents:

Know Your AZ-400 Certification Well:

The AZ-400 is best suitable for candidates who want to gain knowledge in the Microsoft Azure. Before you start your AZ-400 preparation you may struggle to get all the crucial DevOps Engineer materials like AZ-400 syllabus, sample questions, study guide.

But don't worry the AZ-400 PDF is here to help you prepare in a stress free manner. The PDF is a combination of all your queries like-

- What is in the AZ-400 syllabus?
- How many questions are there in the AZ-400 exam?
- Which Practice test would help me to pass the AZ-400 exam at the first attempt?

Passing the AZ-400 exam makes you Microsoft Certified - DevOps Engineer Expert. Having the DevOps Engineer certification opens multiple opportunities for you. You can grab a new job, get a higher salary or simply get recognition within your current organization.

Microsoft AZ-400 DevOps Engineer Certification Details:

Exam Name	Microsoft Certified - DevOps Engineer Expert
Exam Code	AZ-400
Exam Price	$165 (USD)
Duration	150 mins
Number of Questions	40-60
Passing Score	700 / 1000
Books / Training	**AZ-400T00-A: Designing and Implementing Microsoft DevOps solutions**
Schedule Exam	**Pearson VUE**
Sample Questions	**Microsoft DevOps Engineer Sample Questions**
Practice Exam	**Microsoft AZ-400 Certification Practice Exam**

AZ-400 Syllabus:

Topic	Details
Develop an Instrumentation Strategy (5-10%)	
Design and implement logging	- assess and configure a logging framework - design a log aggregation and storage strategy (e.g., Azure storage) - design a log aggregation and query strategy (e.g., Azure Monitor, Splunk, Exabeam, LogRhythm) - manage access control to logs (workspace-centric/resource-centric) - integrate crash analytics (App Center Crashes, Crashlytics)
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)
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 extension, Log Analytics agent, Azure Platform Logs, Event Grid) - manage Access control to the monitoring platform
Develop a Site Reliability Engineering (SRE) strategy (5-10%)	
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)

Topic	Details
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 - leverage Application Insights Smart Detection and Dynamic thresholds in Azure Monitor
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 - design approach for piecemeal recovery (e.g., to improve recovery time objective strategies) - integrate health check with compute environment - implement different types of health checks (container liveness, startup, shutdown)
Develop a security and compliance plan (10-15%)	
Design an authentication and authorization strategy	- design an access solution (Azure AD Privileged Identity Management (PIM), Azure AD Conditional Access, MFA, Azure AD B2B, etc.) - implement Service Principals and Managed Identity - design an application access solution using Azure AD B2C - configure service connections
Design a sensitive information management strategy	- evaluate and configure vault solution (Azure Key Vault, Hashicorp Vault) - manage security certificates - design a secrets storage and retrieval strategy (KeyVault secrets, GitHub secrets, Azure Pipelines secrets) - formulate a plan for deploying secret files as part of a release
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 Code scanning, GitHub Secret scanning, pipeline-based scans, Git hooks, SonarQube, Dependabot, etc.)
Design governance enforcement mechanisms	- implement Azure policies to enforce organizational requirements - implement container scanning (e.g., static scanning, malware, crypto mining)

Topic	Details
	- design and implement Azure Container Registry Tasks - design break-the-glass strategy for responding to security incidents
Manage source control (10-15%)	
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 - design approach for efficient code reviews (e.g., GitHub code review assignments, schedule reminders, Pull Analytics)
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
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
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) - integrate GitHub Codespaces
Facilitate communication and collaboration (10-15%)	
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, ServiceNow)

Topic	Details
	- integrate GitHub as repository with Azure Boards - communicate user analytics
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)
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 and project management platforms (e.g., Email, SMS, Slack, Teams, ServiceNow, etc.) - integrate build and release with communication platforms (e.g., build fails, release fails) - integrate GitHub pull request approvals via mobile apps
Define and implement continuous integration (20-25%)	
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)
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
Design an application infrastructure management strategy	- assess a configuration management mechanism for application infrastructure - define and enforce desired state configuration for environments

Topic	Details
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)
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
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 (10-15%)	
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.)
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, Azure App Configuration) - design and implement release gates and approval processes

Topic	Details
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

Microsoft AZ-400 Sample Questions:

Question: 1

Your company releases a new iOS app using App Center. You plan to release and distribute the beta version of this app to a small group of internal users.
You want to notify users when a new version is released to test. You need to configure this release with the minimal administrative effort necessary to maintain this process.
Which two actions should you perform to configure this process?
Each correct answer presents part of the solution.

a) Install TestFlight on user phones.
b) Modify your build to distribute to the distribution group.
c) Modify your build to distribute to the store.
d) Create a private distribution group and invite the internal users.
e) Send users the install QR code via email.

Answer: b, d

Question: 2

You have a GitHub repository. You create a new repository in Azure DevOps. You need to recommend a procedure to clone the repository from GitHub to Azure DevOps. What should you recommend?

a) Create a webhook.
b) Create a service connection for GitHub.
c) From Import a Git repository, click Import
d) Create a pull request.
e) Create a personal access token in Azure DevOps.

Answer: c

Question: 3

A company decides to build an Azure DevOps pipeline. System monitoring is a key point, so your team uses Azure Monitor. You need to make sure that an alert is sent when a system resource goes down. What should you do?

a) Configure Azure Monitor to monitor to run on a separate build agent.
b) Configure Azure Monitor to monitor Azure resources and virtual machines (VMs).
c) Configure Azure Monitor to monitor Azure resources and operating system events.
d) Configure Azure Monitor to monitor Azure resources only.

Answer: d

Question: 4

Your company uses a Git repository in Azure Repos to manage the source code of a web application. The master branch is protected from direct updates. Developers work on new features in the topic branches.

Because of the high volume of requested features, it is difficult to follow the history of the changes to the master branch. You need to enforce a pull request merge strategy.

The strategy must meet the following requirements:

- Consolidate commit histories

- Merge tie changes into a tingle commit

Which merge strategy should you use in the branch policy?

a) Git fetch
b) no-fast-forward merge
c) squash merge
d) fast-forward merge

Answer: c

Question: 5

A company uses several open-source libraries in their Azure pipeline. These libraries have their own terms and conditions.

You need to make sure that the company is using software with approved terms and conditions. To meet this requirement, you decide to use WhiteSource.

Which two actions should you perform to achieve this goal?

Each correct answer presents part of the solution.

a) Define a black list of automatically rejected licenses.
b) Let WhiteSource automatically detect licenses any time an open- source component is added.
c) Define a white list of automatically approved licenses.
d) Create scripts that run before adding any open-source components.

Answer: a, c

Question: 6

Your company uses Azure DevOps. Only users who have accounts in Azure Active Directory can access the Azure DevOps environment.

You need to ensure that only devices that are connected to the on-premises network can access the Azure DevOps environment.

What should you do?

a) Assign the Stakeholder access level all users.
b) In Azure Active Directory, configure risky sign-ins.
c) In Azure DevOps, configure Security in Project Settings.
d) In Azure Active Directory, configure conditional access.

Answer: d

Question: 7

Your company uses Service Now for incident management. You develop an application that runs on Azure. The company needs to generate a ticket in Service Now when the application fails to authenticate.

Which Azure Log Analytics solution should you use?

a) Automation & Control
b) IT Service Management Connector (ITSM)
c) Application lmiQ.hu Connector
d) insight & Analytics

Answer: b

Question: 8

A company uses Azure DevOps as their development platform. The DevOps team needs to speed up the development process. What step should they perform to increase the speed of development?

a) Do all development locally and then push to the cloud.
b) Use serverless applications.
c) Create a script in the pipeline that merges local changes into feature branches.
d) Use virtual machines (VMs) instead of physical servers in the workflow.

Answer: b

Question: 9

You plan to share packages that you wrote, tested, validated, and deployed by using Azure Artifacts. You need to release multiple builds of each package by using a single feed.

The solution must limit the release of packages that are in development. What should you use?

a) global symbols
b) local symbols
c) upstream sources
d) views

Answer: c

Question: 10

Which branching strategy should you recommend for the investment planning applications suite?

a) release isolation
b) main only
c) development isolation
d) feature isolation

Answer: c

Study Guide to Crack Microsoft DevOps Engineer AZ-400 Exam:

- Getting details of the AZ-400 syllabus, is the first step of a study plan. This pdf is going to be of ultimate help. Completion of the syllabus is must to pass the AZ-400 exam.
- Making a schedule is vital. A structured method of preparation leads to success. A candidate must plan his schedule and follow it rigorously to attain success.
- Joining the Microsoft provided training for AZ-400 exam could be of much help. If there is specific training for the exam, you can discover it from the link above.
- Read from the AZ-400 sample questions to gain your idea about the actual exam questions. In this PDF useful sample questions are provided to make your exam preparation easy.
- Practicing on AZ-400 practice tests is must. Continuous practice will make you an expert in all syllabus areas.

Reliable Online Practice Test for AZ-400 Certification

Make EduSum.com your best friend during your Designing and Implementing Microsoft DevOps Solutions exam preparation. We provide authentic practice tests for the AZ-400 exam. Experts design these online practice tests, so we can offer you an exclusive experience of taking the actual AZ-400 exam. We guarantee you 100% success in your first exam attempt if you continue practicing regularly. Don't bother if you don't get 100% marks in initial practice exam attempts. Just utilize the result section to know your strengths and weaknesses and prepare according to that until you get 100% with our practice tests. Our evaluation makes you confident, and you can score high in the AZ-400 exam.

Start Online practice of AZ-400 Exam by visiting URL
https://www.edusum.com/microsoft/az-400-designing-and-implementing-microsoft-devops-solutions