301-258-8200 | Sales@PhoenixTS.com | www.PhoenixTS.com

MICROSOFT

Microsoft Certified: Azure DevOps Engineer Expert

Validate your technical skills and grow your career.

This certification demonstrates that the recipient is responsible for implementing a scalable, robust pragmatic and realistic cloud solution that works for their organization.

The candidate should possess people skills, along with processes and technologies that can repeatedly deliver value to their company.

Why Take The Microsoft Certified: Azure DevOps Engineer Expert AZ-400

The need AI tech professionals is going to increase dramatically in the near future and passing the exam will help you secure an excellent position in the industry as an Azure DevOps Engineer

Increase My Salary:

- The average salary for someone who holds a Microsoft Certified: Azure DevOps Engineer Expert certification is around $145,000 / year.

Be Part Of The Team:

- Write software for scaling delivery of system automation and automated deployments while working in unison with stakeholders.

Abilities Validated By The Certification:

- Develop an instrumentation strategy
- Develop a Site Reliability Engineering (SRE) strategy
- Develop a security and compliance plan
- Manage source control

10420 Little Patuxent Parkway Suite 500 Columbia, MD 21044

- Facilitate communication and collaboration
- Define and implement continuous integration
- Define and implement a continuous delivery and release management strategy

Recommended Knowledge & Experience:

- Prerequisites include a certification in either Azure Administration or Azure Development

Exam Topics & Scoring:

Exam AZ-400: Designing and Implementing Microsoft DevOps Solutions

DEVELOP AN INSTRUMENTATION STRATEGY (5-10%)

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)

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 VM extensions, 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)

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

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 (10-15%)

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

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

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)

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 (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

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)

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)
- 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 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 (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

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
- manage pipeline health
- identify the number of agents and jobs to run in parallel
- investigate test failures

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, multiregion, 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)
- design and implement release gates and approval processes

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

Prepare for your exam:

The best way to prepare is with first-hand experience. Taking advantage of the opportunities that Phoenix TS provides will assist you with gathering all the knowledge and skills you'll need for certification.

Phoenix TS Microsoft Certified: Azure DevOps Engineer Expert – Learning Pathways

- **AZ-400T00: Designing and Implementing Microsoft DevOps solutions**

  Course Overview Phoenix TS' 5-day instructor-led Microsoft Designing and Implementing Microsoft DevOps solutions training and certification boot camp in Washington, DC Metro, Tysons Corner, VA, Columbia, MD or Live Online provides you the knowledge and skills to design and implement DevOps processes and practices. Students will learn how to plan for DevOps, use source control, […]

  Click To Read More

1 –

https://www.globalknowledge.com/us-en/resources/resource-library/articles/top-paying-certifications/?utm_source=Sales-Enablement&utm_medium=White-Paper&utm_campaign=&utm_content=Top-Paying-Certs



Price Match Guarantee

We'll match any competitor's price quote. Call us at 240-667-7757.

Exam Details

- Multiple choice, multiple answers
- Testing in person or online proctored exam
- 130 Min to take the test
- $150 test fee
- Available in English, Japanese, Korean, and Simplified Chinese