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

View Full Course Details including Latest Schedule Online

MICROSOFT

AZ-400T00: Designing and Implementing Microsoft DevOps solutions

BONUS! Cyber Phoenix Subscription Included: All Phoenix TS students receive complimentary ninety (90) day access to the Cyber Phoenix learning platform, which hosts hundreds of expert asynchronous training courses in Cybersecurity, IT, Soft Skills, and Management and more!

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, scale Git for an enterprise, consolidate artifacts, design a dependency management strategy, manage secrets, implement continuous integration, implement a container build strategy, design a release strategy, set up a release management workflow, implement a deployment pattern, and optimize feedback mechanisms.

What You'll Learn

- Plan for the transformation with shared goals and timelines
- Select a project and identify project metrics and KPIs
- Create a team and agile organization structure
- Describe the benefits of using Source Control
- Migrate from TFVC to Git
- Scale Git for Enterprise DevOps
- Recommend artifact management tools and practices
- Abstract common packages to enable sharing and reuse
- Migrate and consolidate artifacts
- Migrate and integrate source control measures
- Manage application config and secrets
- Develop a project quality strategy
- Plan for secure development practices and compliance rules

- Implement and manage build infrastructure
- Explain why continuous integration matters
- Implement continuous integration using Azure DevOps
- Manage code quality including: technical debt, SonarCloud, and other tooling solutions
- Manage security policies with open source, OWASP, and WhiteSource Bolt
- Implement a container strategy including how containers are different from virtual machines and how microservices use containers
- Implement containers using Docker
- Inspect open source software packages for security and license compliance to align with corporate standards
- Configure build pipeline to access package security and license rating
- Configure secure access to package feeds
- Inspect codebase to identify code dependencies that can be converted to packages
- Identify and recommend standardized package types and versions across the solution
- Refactor existing build pipelines to implement version strategy that publishes packages
- Manage security and compliance
- Differentiate between a release and a deployment
- Define the components of a release pipeline
- Explain things to consider when designing your release strategy
- Classify a release versus a release process and outline how to control the quality of both
- Describe the principle of release gates and how to deal with release notes and documentation
- Explain deployment patterns, both in the traditional sense and in the modern sense
- Choose a release management tool
- Explain the terminology used in Azure DevOps and other Release Management Tooling
- Describe what a Build and Release task is, what it can do, and some available deployment tasks
- Classify an Agent, Agent Queue, and Agent Pool
- Explain why you sometimes need multiple release jobs in one release pipeline
- Differentiate between multi-agent and multi-configuration release job
- Use release variables and stage variables in your release pipeline
- Deploy to an environment securely using a service connection
- Embed testing in the pipeline
- List the different ways to inspect the health of your pipeline and release by using alerts, service hooks, and reports
- Create a release gate
- Describe deployment patterns
- Implement Blue Green Deployment
- Implement Canary Release
- Implement Progressive Exposure Deployment
- Configure crash report integration for client applications
- Develop monitoring and status dashboards
- Implement routing for client application crash report data
- Implement tools to track system usage, feature usage, and flow
- Integrate and configure ticketing systems with development team’s work management
- Implement a mobile DevOps strategy

- Apply infrastructure and configuration as code principles.
- Deploy and manage infrastructure using Microsoft automation technologies such as ARM templates, PowerShell, and Azure CLI
- Describe deployment models and services that are available with Azure
- Deploy and configure a Managed Kubernetes cluster
- Deploy and configure infrastructure using 3rd party tools and services with Azure, such as Chef, Puppet, Ansible, SaltStack, and Terraform
- Define an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructure
- Implement compliance and security in your application infrastructure
- Design practices to measure end-user satisfaction
- Design processes to capture and analyze user feedback from external sources
- Design routing for client application crash report data
- Recommend monitoring tools and technologies
- Recommend system and feature usage tracking tools
- Analyze alerts to establish a baseline
- Analyze telemetry to establish a baseline
- Perform live site reviews and capture feedback for system outages
- Perform ongoing tuning to reduce meaningless or non-actionable alerts

Schedule

Currently, there are no public classes scheduled. Please contact a Phoenix TS Training Consultant to discuss hosting a private class at 301-258-8200.

Program Level

Advanced

Training Delivery Methods

Group Live

Duration

5 Days / 32 hours Training

CPE credits

26 NASBA CPE Credits

Field of Study

Information Technology

Advanced Prep

N/A

Course Registration

Candidates can choose to register for the course by via any of the below methods:

- Email: Sales@phoenixts.com
- Phone: 301-582-8200
- Website: www.phoenixts.com

Upon registration completion candidates are sent an automated course registration email that includes attachments with specific information on the class and location as well as pre-course study and test preparation material approved by the course vendor. The text of the email contains a registration confirmation as well as the location, date, time and contact person of the class.

Online enrolment closes three days before course start date.

On the first day of class, candidates are provided with instructions to register with the exam provider before the exam date.

Complaint Resolution Policy

To view our complete Complaint Resolution Policy policy please click here: Complaint Resolution Policy

Refunds and Cancellations

To view our complete Refund and Cancellation policy please click here: Refund and Cancellation Policy

Who Should Attend

Students in this course are interested in implementing DevOps processes or in passing the Microsoft Azure DevOps Solutions certification exam.

Prerequisites

Fundamental knowledge about Azure, version control, Agile software development, and core software development principles. It would be helpful to have experience in an organization that delivers software.

Exam Information

AZ-400: Designing and Implementing Microsoft DevOps Solutions

Note: The content of this exam was updated on June 15, 2020.

Develop an instrumentation strategy	5-10%
Develop a Site Reliability Engineering (SRE) strategy	5-10%
Develop a security and compliance plan	10-15%
Manage source control	10-15%
Facilitate communication and collaboration	10-15%
Define and implement continuous integration	20-25%
Define and implement a continuous delivery and release management strategy	10-15%

You can purchase the exam voucher separately through Phoenix TS. Phoenix TS is an authorized testing center for Pearson VUE and Prometric websites. Register for exams by calling us or visiting the Pearson VUE and Prometric websites.

Duration

5 Days

Price

$3,295

Course Outline

Module 1: Planning for DevOps

Lessons

- Transformation Planning
- Project Selection
- Team Structures
- Migrating to Azure DevOps

Lab : Agile Planning and Portfolio Management with Azure Boards

After completing this module, students will be able to:

- Plan for the transformation with shared goals and timelines
- Select a project and identify project metrics and KPIs
- Create a team and agile organizational structure
- Design a tool integration strategy
- Design a license management strategy (e.g. VSTS users)
- Design a strategy for end-to-end traceability from work items to working software
- Design an authentication and access strategy
- Design a strategy for integrating on-premises and cloud resources

Module 2: Getting started with Source Control

Lessons

- What is Source Control
- Benefits of Source Control
- Types of Source Control Systems
- Introduction to Azure Repos
- Introduction to GitHub
- Migrating from Team Foundation Version Control (TFVC) to Git in Azure Repos
- Authenticating to Git in Azure Repos

Lab : Version Controlling with Git

After completing this module, students will be able to:

- Describe the benefits of using Source Control
- Describe Azure Repos and GitHub
- Migrate from TFVC to Git

Module 3: Scaling Git for enterprise DevOps

Lessons

- How to Structure your Git Repo
- Git Branching Workflows
- Collaborating with Pull Requests in Azure Repos
- Why care about GitHooks
- Fostering Inner Source

Lab : Code Review with Pull Requests

After completing this module, students will be able to:

- Explain how to structure Git repos
- Describe Git branching workflows
- Leverage pull requests for collaboration and code reviews
- Leverage Git hooks for automation
- Use git to foster inner source across the organization

Module 4: Consolidating Artifacts & Designing a Dependency Management Strategy

Lessons

- Packaging Dependencies
- Package Management
- Migrating and Consolidating Artifacts

Lab : Updating Packages

After completing this module, students will be able to:

- Recommend artifact management tools and practices
- Abstract common packages to enable sharing and reuse
- Migrate and consolidate artifacts
- Migrate and integrate source control measures

Module 5: Implementing Continuous Integration with Azure Pipelines

Lessons

- The concept of pipelines in DevOps
- Azure Pipelines
- Evaluate use of Hosted vs Private Agents

- Agent Pools
- Pipelines and Concurrency
- Azure DevOps and Open Source Projects (Public Projects)
- Azure Pipelines YAML vs Visual Designer
- Continuous Integration Overview
- Implementing a Build Strategy
- Integration with Azure Pipelines
- Integrate External Source Control with Azure Pipelines
- Set Up Private Agents
- Analyze and Integrate Docker Multi-Stage Builds

Lab : Enabling Continuous Integration with Azure Pipelines

Lab : Integrating External Source Control with Azure Pipelines

After completing this module, students will be able to:

- Implement and manage build infrastructure
- Explain why continuous integration matters
- Implement continuous integration using Azure DevOps

Module 6: Managing Application Config and Secrets

Lessons

- Introduction to Security
- Implement secure and compliant development process
- Rethinking application config data
- Manage secrets, tokens, and certificates
- Implement tools for managing security and compliance in a pipeline

Lab : Integrating Azure Key Vault with Azure DevOps

After completing this module, students will be able to:

- Manage application config and secrets
- Implement tools for managing security and compliance in pipeline

Module 7: Managing Code Quality and Security Policies

Lessons

- Managing Code Quality
- Managing Security Policies

Lab : Managing Technical Debt with Azure DevOps and SonarCloud

After completing this module, students will be able to:

- Manage code quality including: technical debt SonarCloud, and other tooling solutions
- Manage security policies with open source and OWASP

Module 8: Implementing a Container Build Strategy

Lessons

- Implementing a Container Build Strategy

Lab : Modernizing Existing ASP.NET Apps with Azure

After completing this module, students will be able to:

- Implement a container strategy including how containers are different from virtual machines and how microservices use containers
- Implement containers using Docker

Module 9: Manage Artifact versioning, security & compliance

Lessons

- Package security
- Open source software
- Integrating license and vulnerability scans
- Implement a versioning strategy

Lab : Manage Open Source Security and License with WhiteSource

After completing this module, students will be able to:

- Inspect open source software packages for security and license compliance to align with corporate standards
- Configure build pipeline to access package security and license rating
- Configure secure access to package feeds
- Inspect codebase to identify code dependencies that can be converted to packages
- Identify and recommend standardized package types and versions across the solution
- Refactor existing build pipelines to implement version strategy that publishes packages
- Manage security and compliance

Module 10: Design a Release Strategy

Lessons

- Introduction to Continuous Delivery
- Release strategy recommendations
- Building a High-Quality Release pipeline
- Choosing a deployment pattern
- Choosing the right release management tool

After completing this module, students will be able to:

- Differentiate between a release and a deployment
- Define the components of a release pipeline
- Explain things to consider when designing your release strategy
- Classify a release versus a release process and outline how to control the quality of both
- Describe the principle of release gates and how to deal with release notes and documentation
- Explain deployment patterns, both in the traditional sense and in the modern sense
- Choose a release management tool

Module 11: Set up a Release Management Workflow

Lessons

- Create a Release Pipeline
- Provision and Configure Environments
- Manage and Modularize Tasks and Templates
- Integrate Secrets with the release pipeline
- Configure Automated Integration and Functional Test Automation
- Automate Inspection of Health

Lab : Configuring Pipelines as Code with YAML

Lab : Setting up secrets in the pipeline with Azure Key vault

Lab : Setting up and Running Functional Tests

Lab : Using Azure Monitor as release gate

Lab : Creating a release Dashboard

After completing this module, students will be able to:

- Explain the terminology used in Azure DevOps and other Release Management Tooling
- Describe what a Build and Release task is, what it can do, and some available deployment tasks
- Classify an Agent, Agent Queue, and Agent Pool
- Explain why you sometimes need multiple release jobs in one release pipeline
- Differentiate between multi-agent and multi-configuration release job
- Use release variables and stage variables in your release pipeline
- Deploy to an environment securely using a service connection

- Embed testing in the pipeline
- List the different ways to inspect the health of your pipeline and release by using alerts, service hooks, and reports
- Create a release gate

Module 12: Implement an appropriate deployment pattern

Lessons

- Introduction to Deployment Patterns
- Implement Blue Green Deployment
- Feature Toggles
- Canary Releases
- Dark Launching
- AB Testing
- Progressive Exposure Deployment

Lab : Feature Flag Management with LaunchDarkly and Azure DevOps

After completing this module, students will be able to:

- Describe deployment patterns
- Implement Blue Green Deployment
- Implement Canary Release
- Implement Progressive Exposure Deployment

Module 13: Implement process for routing system feedback to development teams

Lessons

- Implement Tools to Track System Usage, Feature Usage, and Flow
- Implement Routing for Mobile Application Crash Report Data
- Develop Monitoring and Status Dashboards
- Integrate and Configure Ticketing Systems

Lab : Monitoring Application Performance

After completing this module, students will be able to:

- Configure crash report integration for client applications
- Develop monitoring and status dashboards
- Implement routing for client application crash report data
- Implement tools to track system usage, feature usage, and flow
- Integrate and configure ticketing systems with development team’s work management

Module 14: Infrastructure and Configuration Azure Tools

Lessons

- Infrastructure as Code and Configuration Management
- Create Azure Resources using ARM Templates
- Create Azure Resources using Azure CLI
- Create Azure Resources by using Azure PowerShell
- Desired State Configuration (DSC)
- Azure Automation with DevOps
- Additional Automation Tools

Lab : Azure Deployments using Resource Manager Templates

After completing this module, students will be able to:

- Apply infrastructure and configuration as code principles
- Deploy and manage infrastructure using Microsoft automation technologies such as ARM templates, PowerShell, and Azure CLI

Module 15: Azure Deployment Models and Services

Lessons

- Deployment Modules and Options
- Azure Infrastructure-as-a-Service (IaaS) Services
- Azure Platform-as-a-Service (PaaS) services
- Serverless and HPC Computer Services
- Azure Service Fabric

Lab : Deploying a Dockerized Java app to Azure Web App for Containers

After completing this module, students will be able to:

- Describe deployment models and services that are available with Azure

Module 16: Create and Manage Kubernetes Service Infrastructure

Lessons

- Azure Kubernetes Service

Lab : Deploying a multi-container application to Azure Kubernetes Service

After completing this module, students will be able to:

- Deploy and configure a Managed Kubernetes cluster

Module 17: Third Party Infrastructure as Code Tools available with Azure

Lessons

- Chef
- Puppet
- Ansible
- Terraform

Lab : Infrastructure as Code

Lab : Automating Your Infrastructure Deployments in the Cloud with Terraform and Azure Pipelines

After completing this module, students will be able to:

- Deploy and configure infrastructure using 3rd party tools and services with Azure, such as Chef, Puppet, Ansible, and Terraform

Module 18: Implement Compliance and Security in your Infrastructure

Lessons

- Security and Compliance Principles with DevOps
- Azure security Center

Lab : Implement Security and Compliance in an Azure DevOps Pipeline

After completing this module, students will be able to:

- Define an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructure
- Implement compliance and security in your application infrastructure

Module 19: Recommend and design system feedback mechanisms

Lessons

- The inner loop
- Continuous Experimentation mindset
- Design practices to measure end-user satisfaction
- Design processes to capture and analyze user feedback
- Design process to automate application analytics

Lab : Integration between Azure DevOps and Teams

After completing this module, students will be able to:

- Design practices to measure end-user satisfaction
- Design processes to capture and analyze user feedback from external sources
- Design routing for client application crash report data
- Recommend monitoring tools and technologies
- Recommend system and feature usage tracking tools

Module 20: Optimize feedback mechanisms

Lessons

- Site Reliability Engineering
- Analyze telemetry to establish a baseline
- Perform ongoing tuning to reduce meaningless or non-actionable alerts
- Analyze alerts to establish a baseline
- Blameless Retrospectives and a Just Culture

After completing this module, students will be able to:

- Analyze alerts to establish a baseline
- Analyze telemetry to establish a baseline
- Perform live site reviews and capture feedback for system outages
- Perform ongoing tuning to reduce meaningless or non-actionable alerts

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

Starting at **$3,295**

ATTENTION

For GSA pricing or Contractor quotes call 301-258-8200 – Option 2.

GSA

Price Match Guarantee

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

10420 Little Patuxent Parkway Suite 500 Columbia, MD 21044