CREDO SYSTEMZ
Simplifying IT

# START YOUR AZ 400 CAREER TODAY!!

CREDO SYSTEMZ

## Azure DevOps Engineer Program

# Capstone Projects :

Real Time Business Scenario using
Azure DevOps Engineer

### CI/CD Pipeline with Azure DevOps

Set up source control, build triggers, automated testing, and deployment stages for a sample .NET or Node.js web application using Azure DevOps.

### Infrastructure as Code with ARM/Bicep

Write and deploy infrastructure templates for resources like App Services, Azure SQL, and Storage Accounts. Integrate into a pipeline for version-controlled deployments.

### Containerized Application Deployment with AKS

Containerize an app, push to Azure Container Registry, and deploy to AKS using Helm charts. Implement rolling updates and scaling.

### Automated Testing and Quality Gates

Integrate unit, integration, and UI tests into the CI pipeline. Set up code quality gates with SonarCloud and enforce branch policies using Azure Repos.

### Azure Monitoring and Alerting

Add telemetry to a deployed application. Create dashboards, define metrics, set up alerts, and integrate incident management (e.g., via email or ITSM tools).

### Blue-Green Deployment with Traffic Routing

Configure a blue-green deployment using App Service slots or AKS namespaces. Use Azure Front Door for traffic routing and gradual rollout strategies.

# AZURE DEVOPS ENGINEER
# COURSE SYLLABUS

**Duration : 40 hrs**

## Section 1 : Design a DevOps strategy

**Migration and consolidation strategy - DevOps tools**

- Analyze existing artifact - deployment packages, NuGet, Maven, npm
- Container repositories
- Test management tools
- Work management tools
- Recommend migration and integration strategies
  - Artifact repositories
  - Source control
  - Test management
  - Work management

## Section 2 : Understanding Agile work management approach

- Identify and recommend project metrics, KPIs, and DevOps measurements
- Agile work management
- Mentor team members on Agile techniques and practices
- Scaling Agile practices
- Understanding in-team and cross-team collaboration mechanisms

## Section 3 : Design a quality strategy

- Analyze existing quality environment
- Working quality metrics
- Feature flag lifecycle
- Measuring and managing technical debt
- Changes to team structure to optimize quality
- Recommend performance testing strategy

## Section 4 : Design a secure development process

- Inspect and validate code base for compliance
- Inspect and validate infrastructure for compliance
- Secure development strategy
- Integrate code security validation - static code analysis
- Integrate infrastructure security validation

## Section 5 : Design a tool integration strategy

To design a license management strategy

  - VSTS users
  - concurrent pipelines
  - test environments,
  - open source software licensing
  - third-party DevOps tools and services
  - package management licensing
- Design a strategy for end-to-end traceability from work items to working software
- Integrating monitoring and feedback to development teams
- Authentication and access strategy
- Integrating on-premises and cloud resources

## Section 6 : Implement DevOps development processes

**Design a version control strategy**

- Working with Branching models
- Introduction to Version control systems
- Understaning Code flow strategy

## Section 7 : Implement and integrate source control

- External source control
- Integrate source control into third-party continuous integration and continuous
- deployment (CI/CD) systems

## Section 8 : Implement and manage build infrastructure

- Private and hosted agents
- Working with third party build systems
- Concurrent pipelines
- Manage Azure pipeline configuration
  - Agent queues
  - Service endpoints
  - Pools
  - Webhooks

## Section 9 :Implement code flow

- Pull request strategies
- Branch and fork strategies
- configure branch policies

## Section 10 : Implement a mobile DevOps strategy

- Manage mobile target device sets and distribution groups
- Target UI test device sets
- Provision tester devices for deployment
- Create public and private distribution groups

## Section 11 : Managing application configuration and secrets

- Secure and compliant development process
- General (non-secret) configuration data
- secrets, tokens, and certificates
- applications configurations
  - Web App
  - Azure Kubernetes Service
  - containers
- Secrets management
  - Web App
  - Azure Kubernetes Service
  - containers
  - Azure Key Vault
- Managing security and compliance in the pipeline

## Section 12 : Implement continuous integration

- Manage code quality and security policies
- Monitor code quality
- Configure build to report on code coverage
- Automated test quality
- Test suites and categories
- Monitor quality of tests
- Security analysis tools
  - SonarQube,
  - White Source Bolt
  - Open Web Application Security Project

## Section 13 : Implement a container build strategy

- create deployable images
  - Docker
  - Hub
  - Azure Container Registry
- Docker multi-stage builds

## Section 14 : Implement a build strategy

- Design build triggers, tools, integrations, and workflow
- Hybrid build process
- Multi-agent builds
- Build tools and configuration (e.g. Azure Pipelines, Jenkins)
- set up an automated build workflow

## Section 15 : Implement continuous delivery

**Design a release strategy**

- Release tools
- Identify and recommend release approvals and gates
- Measuring quality of release and release process
- Recommend strategy for release notes and documentation
- select appropriate deployment pattern

## Section 16 : Set up a release management workflow

- Automate inspection of health signals for release approvals by using release gates
- Configure automated integration and functional test execution
- Create a release pipeline
  - Azure Kubernetes Service
  - Service Fabric
  - WebApp
- Create multi-phase release pipelines
- Integrate secrets with release pipeline
- Provision and configure environments
- Manage and modularize tasks and templates - task and variable groups

## Section 17 : Implement an appropriate deployment pattern

- Implement blue-green deployments
- Implement canary deployments
- Implement progressive exposure deployments
- Scale a release pipeline to deploy to multiple endpoints
  - Deployment groups
  - Azure Kubernetes Service
  - Service Fabric

## Section 18 : Implement dependency management

**Design a dependency management strategy**

- Artifact management tools and practices (Azure Artifacts, npm, Maven, Nuget)
- Abstract common packages to enable sharing and reuse
- Inspect codebase to identify code dependencies that can be converted to packages
- Standardized package types and versions across the solution
- Refactor existing build pipelines to implement version strategy that publishes packages

## Section 19 : Manage security and compliance

- Inspect open source software packages for security and license compliance to align with
- corporate standards (e.g., GPLv3)
- Configure build pipeline to access package security and license rating (e.g., Black Duck,
- White Source)
- Configure secure access to package feeds

## Section 20 : Implement application infrastructure

**Design an infrastructure and configuration management strategy**

- Existing and future hosting infrastructure
- Infrastructure as Code (IaC) technologies
- managing technical debt on templates
- Transient infrastructure for parts of a delivery lifecycle
- Mitigate infrastructure state drift

## Section 21 : Implement Infrastructure as Code (IaC)

- Create nested resource templates
- Manage secrets in resource templates
- Provision Azure resources
- Recommend an Infrastructure as Code (IaC) strategy
- Recommend appropriate technologies for configuration management
  - ARM Templates
  - Terraform
  - Chef
  - Puppet
  - Ansible

## Section 22 : Manage Azure Kubernetes Service infrastructure

- Provision Azure Kubernetes Service - ARM templates, CLI

- Create deployment file for publishing to Azure Kubernetes Service - kubectl, Helm
- Develop a scaling plan

## Section 23 : Implement infrastructure compliance and security

- Compliance and security scanning
- Prevent drift by using configuration management tools
- Automate configuration management by using PowerShell Desired State Configuration (DSC)
- Automate configuration management by using a VM Agent with custom script
- extensions
- Set up an automated pipeline to inspect security and compliance

## Section 24 : Implement continuous feedback

**Recommend and design system feedback mechanisms**

- Design practices to measure end-user satisfaction - Send a Smile, app analytics
- Design processes to capture and analyze user feedback from external sources - Twitter,
- Reddit, Help Desk
- Design routing for client application crash report data
- Monitoring tools and technologies
- Feature usage tracking tools

## Section 25 : Implement process for routing system feedback to development teams

- 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 system

- IT Service Management connector
- ServiceNow Cloud Management
- App Insights work items

## Section 26 : Optimize feedback mechanisms

- 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

# SKILLS AND TOOLS

## Tools Covered

Azure Boards

Azure Pipelines

Azure DevOps auth

Azure ML

Repo Scaling

## Skills Covered

Process

Source Control

Pipelines

Security

Compliance

Instrumentation

Communication

# Earn your AZ 400 Course Completion Certificate

Credo Systemz's certificate is highly recognized by 30K Global companies around the world.

CREDO SYSTEMZ
Simplifying IT

CERTIFICATE
OF EXCELLENCE

THIS IS TO CERTIFY THAT

VETRIVEL. S

CANDIDATE ID : [illegible]
has successfully completed
AZURE DEVOPS ENGINEER
Course Conducted During February 2025 to April 2025

Date: 04-05-2025

Authorised Signatory

Plot No.8 , Vinayaga Ave, Rajiv Gandhi Salai, OMR ,Chennai - 42. Ph : 98844 12301 / 96001 12302, Web : www.credosystemz.com

# WHAT OUR TRAINEE SAYS?

## Jhanvi

**4.7** ★★★★½

The trainer explained cloud concepts clearly with real-life examples, making it easy to understand for beginners. Practice sessions and doubt clearing were really helpful for building a strong foundation.

## Nirosha

**4.2** ★★★★½

The course was well-structured with hands-on labs for each topic, which improved my practical skills. The trainer's guidance on managing Azure resources and services was excellent.

## Karthik M

**5.0** ★★★★★

The sessions focused on design patterns, architecture scenarios, and case studies that matched the exam syllabus. The instructor made complex concepts simple with diagrams and live use cases.

## Srinivasan

**4.9** ★★★★½

The training covered pipelines, CI/CD, and DevOps practices in a very practical way. Tools like Azure DevOps and GitHub Actions were taught with real-time project examples.

## Priya

**4.0** ★★★★

Security topics like IAM, Defender for Cloud, and Key Vault were explained with clarity. Lab exercises boosted my confidence to handle security tasks in a real environment.

## Renuka Devi

**4.5** ★★★★½

Networking concepts like VNets, VPNs, and routing were made easy with step-by-step labs. The trainer's deep knowledge and support throughout the course were very helpful.

# CHENNAI

## VELACHERY

New # 30, Old # 16A, Third Main Road, Rajalakshmi Nagar, Velachery, (Opp. to Murugan Kalyana Mandapam), Chennai – 600 042.

+91 98844 12301

## OMR

Plot No.8, Vinayaga Avenue, Rajiv Gandhi Salai, (OMR), Okkiampettai, (Behind Okkiyampet Bus Stop) Chennai – 600 097.

+91 96001 12302

# OVERSEAS

## USA

Houchin Drive, Franklin, TN -37064. Tennessee

## UAE

Sima Electronic Building, LLH Opposite, Electra Street – Abu Dhabi