Implementing DevOps Solutions and Practices using Cisco Platforms v1.1 (300-910)

Exam Description: Implementing DevOps Solutions and Practices using Cisco Platforms v1.1 (DEVOPS 300-910) is a 90-minute exam associated with the DevNet Professional Certification. This exam certifies a candidate's knowledge of DevOps practices as it pertains to deployment automation that enables automated configuration, management, and scalability of cloud microservices and infrastructure processes on Cisco platforms. The course, Implementing DevOps Solutions and Practices using Cisco Platforms, helps candidates to prepare for this exam.

The following topics are general guidelines for the content likely to be included on the exam. However, other related topics may also appear on any specific delivery of the exam. To better reflect the contents of the exam and for clarity purposes, the guidelines below may change at any time without notice.

20%	1.0	**CI/CD Pipeline**

1. Describe characteristics and concepts of build /deploy tools such as Jenkins, GitLab CI/CD, or Travis CI

2. Identify the sequence, components, and integrations to implement a CI/CD pipeline for a given scenario

3. Troubleshoot issues with a CI/CD pipeline such as code-based failures, pipeline issues, and tool incompatibility

4. Identify tests to integrate into a CI/CD pipeline for a given scenario

5. Identify release deployment strategy (canary, rollbacks, and blue/green) for a given scenario

6. Diagnose code dependency management issues including API, tool chain, and libraries

7. Interpret a GitLab CI/CD configuration file

15%	2.0	**Packaging and Delivery of Applications**

1. Identify the steps to containerize an application

2. Identify steps to deploy multiple microservice applications

3. Evaluate microservices and container architecture diagrams based on technical and business requirements (security, performance, stability, and cost)

4. Identify safe handling practices for configuration items, application parameters, and secrets

5. Construct a Docker file to address application specifications

6. Describe the usage of golden images to deploy applications

20%   3.0   Automating Infrastructure

1.   Describe how to integrate DevOps practices into an existing organization structure

2.   Describe the use of configuration management tools to automate infrastructure services such as Ansible and Terraform

3.   Construct an Ansible playbook to automate an application deployment of infrastructure services

4.   Construct a Terraform configuration to automate an application deployment of infrastructure services

5.   Describe the practice and benefits of Infrastructure as Code

6.   Design a pre-check validation of the network state in a CI/CD pipeline for a given scenario

7.   Design a pre-check validation of the application infrastructure in a CI/CD pipeline for a given scenario

8.   Describe the concepts of extending DevOps practices to the network for NetDevOps

9.   Identify the requirements such as memory, disk I/O, network, and CPU needed to scale the application or service

15%   4.0   Cloud and Multicloud

1.   Describe the concepts and objects of Kubernetes

2.   Deploy applications to a Kubernetes cluster

3.   Utilize objects of Kubernetes to build a deployment to meet requirements

4.   Validate the success of an application deployment on Kubernetes

5.   Describe method and considerations to deploy an application to multiple environments such as multiple cloud providers, high availability configurations, disaster recovery configurations, and testing cloud portability

6.   Describe the process of tracking and projecting costs when consuming public cloud

7.   Describe benefits of infrastructure as code for repeatable public cloud consumption

8.   Compare cloud services strategies (build versus buy)

20%   5.0   Logging, Monitoring, and Metrics
1.   Identify the elements of log and metric systems to facilitate application troubleshooting such as performance issues and streaming telemetry logs

2.   Implement a log collection and reporting system for applications

   a.   aggregate logs from multiple related applications

   b.   search capabilities

c.   reporting capabilities

2.   Troubleshoot a distributed application using AppDyanmics with Application Performance Monitoring

3.   Describe the principles of chaos engineering

4.   Construct Python scripts that use APIs to accomplish these tasks

    a.   build a monitoring dashboard

    b.   notify Webex space

    c.   responding to alerts and outages

    d.   creating notifications

    e.   health check monitoring

    f.   opening and closing incidents

2.   Identify additional application requirements to provide visibility into application health and performance

3.   Describe Kubernetes capabilities related to logging, monitoring, and metrics

4.   Describe the integration of logging, monitoring and alerting in a CI/CD pipeline design

10%   6.0   Security
1.   Identify methods to secure an application and infrastructure during production and testing in a CI/CD pipeline

2.   Identify methods to implement a secure software development life cycle