ELYSIUM ACADEMY

Milestone of Cognizance

Since 2007

VERSION

6

TESBO COURSE

ELYSIUM ACADEMY ELYSIUM CERTIFIED DEVOPS ENGINEER

ELYSIUM ACADEMY ELYSIUM CERTIFIED DEVOPS ENGINEER

ELYSIUM ACADEMY ELYSIUM CERTIFIED DEVOPS ENGINEER

ELYSIUM ACADEMY ELYSIUM CERTIFIED DEVOPS ENGINEER

SR. CODE

EAPL/TESBO/TSTC09

COURSE CODE

EATDE

SUB CATEGORY

CLOUD COMPUTING

TOTAL DURATION	THEORY TAKEN	PRACTICAL TAKEN
180 HOURS	50 HOURS	130 HOURS

COURSE DESCRIPTION

The Elysium Certified DevOps Engineer training course is a comprehensive program focusing on AWS Cloud, DevOps methodologies, and essential tools such as Git, GitHub, Jenkins, Maven, Ansible, Kubernetes, and Terraform. You will gain hands-on experience in deploying, automating, and managing infrastructure and applications on AWS, implementing CI/CD pipelines, and orchestrating containerized environments.

COURSE GOALS

By completion, you will master AWS services, DevOps principles, and tools necessary for efficient software delivery and infrastructure management. They will be proficient in version control with Git/GitHub, continuous integration with Jenkins, infrastructure provisioning with Terraform, container orchestration with Kubernetes, and configuration management with Ansible, enabling them to streamline development workflows and enhance organizational agility.

FUTURE SCOPE

Graduates will be well-equipped for roles as DevOps engineers, cloud architects, or automation specialists. With expertise in AWS and popular DevOps tools, they'll find opportunities in companies adopting Agile, DevOps, and cloud-native practices, ensuring a promising career trajectory in the dynamic field of DevOps and cloud computing.

ELYSIUM ACADEMY®
Milestone of Cognizance

Since 2007

SDLC AUTOMATION

01 CHAPTER

IMPLEMENT CI/CD PIPELINES

01 HRS
02 HRS

a. Software development lifecycle (SDLC) concepts, phases, and models
b. Pipeline deployment patterns for single- and multi-account environments
c. Configuring code, image, and artifact repositories
d. Using version control to integrate pipelines with application environments
e. Setting up build processes (for example, AWS CodeBuild)
f. Managing build and deployment secrets (for example, AWS Secrets Manager, AWS Systems Manager Parameter Store)
g. Determining appropriate deployment strategies (for example, AWS CodeDeploy)

02 CHAPTER

INTEGRATE AUTOMATED TESTING INTO CI/CD PIPELINES

01 HRS
03 HRS

a. Different types of tests (for example, unit tests, integration tests, acceptance tests, user interface tests, security scans)
b. Reasonable use of different types of tests at different stages of the CI/CD pipeline
c. Running builds or tests when generating pull requests or code merges (for example, AWS CodeCommit, CodeBuild)

d.Running load/stress tests, performance benchmarking, and application testing at scale

e.Measuring application health based on application exit codes

f. Automating unit tests and code coverage

g.Invoking AWS services in a pipeline for testing

03
CHAPTER

BUILD AND MANAGE ARTIFACTS

01 HRS
03 HRS

a.Artifact use cases and secure management

b.Methods to create and generate artifacts

c.Artifact lifecycle considerations

d.Creating and configuring artifact repositories (for example, AWS CodeArtifact, Amazon S3, Amazon Elastic Container Registry [Amazon ECR])

e.Configuring build tools for generating artifacts (for example, CodeBuild, AWS Lambda)

f.Automating Amazon EC2 instance and container image build processes (for example, EC2 Image Builder)

ELYSIUM ACADEMY®
Milestone of Cognizance

Since 2007

04 CHAPTER

DEPLOYMENT STRATEGIES

02 HRS

03 HRS

a. Deployment methodologies for various platforms (for example, Amazon EC2, Amazon Elastic Container Service [Amazon ECS], Amazon Elastic Kubernetes Service [Amazon EKS], Lambda)

b. Application storage patterns (for example, Amazon Elastic File System [Amazon EFS], Amazon S3, Amazon Elastic Block Store [Amazon EBS])

c. Mutable deployment patterns in contrast to immutable deployment patterns

d. Tools and services available for distributing code (for example, CodeDeploy, EC2 Image Builder)

e. Configuring security permissions to allow access to artifact repositories (for example, AWS Identity and Access Management [IAM], CodeArtifact)

f. Configuring deployment agents (for example, CodeDeploy agent)

g. Troubleshooting deployment issues

h. Using different deployment methods (for example, blue/green, canary)

ELYSIUM ACADEMY®
Milestone of Cognizance

Since 2007

CONFIGURATION MANAGEMENT AND IAC

05
CHAPTER

CLOUD INFRASTRUCTURE AND REUSABLE COMPONENTS

01 HRS

03 HRS

a. Infrastructure as code (IaC) options and tools for AWS

b. Change management processes for IaC-based platforms

c. Configuration management services and strategies

d. Composing and deploying IaC templates (for example, AWS Serverless Application Model [AWS SAM], AWS CloudFormation, AWS Cloud Development Kit [AWS CDK])

e. Applying CloudFormation StackSets across multiple accounts and AWS Regions

f. Determining optimal configuration management services (for example, AWS OpsWorks, AWS Systems Manager, AWS Config, AWS AppConfig)

g. Implementing infrastructure patterns, governance controls, and security standards into reusable IaC templates (for example, AWS Service Catalog, CloudFormation modules, AWS CDK) Task Statement

06 CHAPTER

DEPLOY AUTOMATION

01 HRS
03 HRS

a. AWS account structures, best practices, and related AWS services

b. Standardizing and automating account provisioning and configuration

c. Creating, consolidating, and centrally managing accounts (for example, AWS Organizations, AWS Control Tower)

d. Applying IAM solutions for multi-account and complex organization structures (for example, SCPs, assuming roles)

e. Implementing and developing governance and security controls at scale (AWS Config, AWS Control Tower, AWS Security Hub, Amazon Detective, Amazon GuardDuty, AWS Service Catalog, SCPs)

07 CHAPTER

DESIGN AND BUILD AUTOMATED SOLUTIONS

01 HRS
03 HRS

a. AWS services and solutions to automate tasks and processes

b. Methods and strategies to interact with the AWS software-defined infrastructure

c. Automating system inventory, configuration, and patch management (for example, Systems Manager, AWS Config)

d. Developing Lambda function automations for complex scenarios (for example, AWS SDKs, Lambda, AWS Step Functions)

e. Automating the configuration of software applications to the desired state (for example, OpsWorks, Systems Manager State Manager)

f. Maintaining software compliance (for example, Systems Manager)

RESILIENT CLOUD SOLUTIONS

08
CHAPTER

HIGHLY AVAILABLE SOLUTIONS TO MEET RESILIENCE

02 HRS
04 HRS

a. Multi-AZ and multi-Region deployments (for example, compute layer, data layer)

b. SLAs

c. Replication and failover methods for stateful services

d. Techniques to achieve high availability (for example, Multi-AZ, multi-Region)

e. Translating business requirements into technical resiliency needs

f. Identifying and remediating single points of failure in existing workloads

g. Enabling cross-Region solutions where available (for example, Amazon DynamoDB, Amazon RDS, Amazon Route 53, Amazon S3, Amazon CloudFront)

h. Configuring load balancing to support cross-AZ services

i. Configuring applications and related services to support multiple Availability Zones and Regions while minimizing downtime

09 CHAPTER

SOLUTIONS THAT ARE SCALABLE TO MEET BUSINESS REQUIREMENTS

02 HRS | 04 HRS

a. Appropriate metrics for scaling services
b. Loosely coupled and distributed architectures
c. Serverless architectures
d. Container platforms
e. Identifying and remediating scaling issues
f. Identifying and implementing appropriate auto scaling, load balancing, and caching solutions
g. Deploying container-based applications (for example, Amazon ECS, Amazon EKS)
h. Deploying workloads in multiple Regions for global scalability
i. Configuring serverless applications (for example, Amazon API Gateway, Lambda, AWS Fargate)

10 CHAPTER

AUTOMATED RECOVERY PROCESSES

01 HRS | 04 HRS

a. Disaster recovery concepts (for example, RTO, RPO)
b. Backup and recovery strategies (for example, pilot light, warm standby)
c. Recovery procedures
d. Testing failover of Multi-AZ and multi-Region workloads (for example, Amazon RDS, Amazon Aurora, Route 53, CloudFront)

e. Identifying and implementing appropriate cross-Region backup and recovery strategies (for example, AWS Backup, Amazon S3, Systems Manager)

f. Configuring a load balancer to recover from backend failure

MONITORING AND LOGGING

11 CHAPTER

COLLECTION, AGGREGATION, AND STORAGE OF LOGS AND METRICS

01 HRS

04 HRS

a. How to monitor applications and infrastructure

b. Amazon CloudWatch metrics (for example, namespaces, metrics, dimensions, and resolution)

c. Real-time log ingestion

d. Encryption options for at-rest and in-transit logs and metrics (for example, client-side and server-side, AWS Key Management Service [AWS KMS])

e. Security configurations (for example, IAM roles and permissions to allow for log collection)

f. Securely storing and managing logs

g. Creating CloudWatch metrics from log events by using metric filters

h. Creating CloudWatch metric streams (for example, Amazon S3 or Amazon Kinesis Data Firehose options)

i. Collecting custom metrics (for example, using the CloudWatch agent)
j. Managing log storage lifecycles (for example, S3 lifecycles, CloudWatch log group retention)
k. Processing log data by using CloudWatch log subscriptions (for example, Kinesis, Lambda, Amazon OpenSearch Service)
l. Searching log data by using filter and pattern syntax or CloudWatch Logs Insights
m. Configuring encryption of log data (for example, AWS KMS)

12 CHAPTER

AUDIT, MONITOR, AND ANALYZE LOGS AND METRICS TO DETECT ISSUES

02 HRS
04 HRS

a. Anomaly detection alarms (for example, CloudWatch anomaly detection)
b. Common CloudWatch metrics and logs (for example, CPU utilization with Amazon EC2, queue length with Amazon RDS, 5xx errors with an Application Load Balancer [ALB])
c. Amazon Inspector and common assessment templates
d. AWS Config rules
e. AWS CloudTrail log events
f. Building CloudWatch dashboards and Amazon QuickSight visualizations
g. Associating CloudWatch alarms with CloudWatch metrics (standard and custom)

h. Configuring AWS X-Ray for different services (for example, containers, API Gateway, Lambda)

i. Analyzing real-time log streams (for example, using Kinesis Data Streams)

j. Analyzing logs with AWS services (for example, Amazon Athena, CloudWatch Logs Insights)

13
CHAPTER

MONITORING AND EVENT MANAGEMENT

02 HRS
04 HRS

a. Event-driven, asynchronous design patterns (for example, S3 Event Notifications or Amazon EventBridge events to Amazon Simple Notification Service [Amazon SNS] or Lambda)

b. Capabilities of auto scaling for a variety of AWS services (for example, EC2 Auto Scaling groups, RDS storage auto scaling, DynamoDB, ECS capacity provider, EKS autoscalers)

c. Alert notification and action capabilities (for example, CloudWatch alarms to Amazon SNS, Lambda, EC2 automatic recovery)

d. Health check capabilities in AWS services (for example, ALB target groups, Route 53)

e. Configuring solutions for auto scaling (for example, DynamoDB, EC2 Auto Scaling groups, RDS storage auto scaling, ECS capacity provider)

f. Creating CloudWatch custom metrics and metric filters, alarms, and notifications (for example, Amazon SNS, Lambda)

g. Configuring S3 events to process log files (for example, by using Lambda) and deliver log files to another destination (for example, OpenSearch Service, CloudWatch Logs)

h. Configuring EventBridge to send notifications based on a particular event pattern

i. Installing and configuring agents on EC2 instances (for example, AWS Systems Manager Agent [SSM Agent], CloudWatch agent)

j. Configuring AWS Config rules to remediate issues

k. Configuring health checks (for example, Route 53, ALB)

MONITORING AND LOGGING

14 CHAPTER

PROCESS, NOTIFY, AND TAKE ACTION IN RESPONSE TO EVENTS

01 HRS

03 HRS

a. AWS services that generate, capture, and process events (for example, AWS Health, EventBridge, CloudTrail)

b. Event-driven architectures (for example, fan out, event streaming, queuing)

c. Integrating AWS event sources (for example, AWS Health, EventBridge, CloudTrail)

d. Building event processing workflows (for example, Amazon Simple Queue Service [Amazon SQS], Kinesis, Amazon SNS, Lambda, Step Functions)

15 CHAPTER

CONFIGURATION CHANGES IN RESPONSE TO EVENTS

01 HRS
03 HRS

a. Fleet management services (for example, Systems Manager, AWS Auto Scaling)
b. Configuration management services (for example, AWS Config)
c. Applying configuration changes to systems
d. Modifying infrastructure configurations in response to events
e. Remediating a non-desired system state Task Statement

16 CHAPTER

TROUBLESHOOT SYSTEM AND APPLICATION FAILURES

01 HRS
04 HRS

a. AWS metrics and logging services (for example, CloudWatch, X-Ray)
b. AWS service health services (for example, AWS Health, CloudWatch, Systems Manager OpsCenter)
c. Root cause analysis
d. Analyzing failed deployments (for example, AWS CodePipeline, CodeBuild, CodeDeploy, CloudFormation, CloudWatch synthetic monitoring)
e. Analyzing incidents regarding failed processes (for example, auto scaling, Amazon ECS, Amazon EKS)

SECURITY AND COMPLIANCE

17 CHAPTER

IDENTITY AND ACCESS MANAGEMENT AT SCALE

01 HRS
03 HRS

a. Appropriate usage of different IAM entities for human and machine access (for example, users, groups, roles, identity providers, identity-based policies, resource-based policies, session policies)

b. Identity federation techniques (for example, using IAM identity providers and AWS IAM Identity Center [AWS Single Sign-On])

c. Permission management delegation by using IAM permissions boundaries

d. Organizational SCPs

e. Designing policies to enforce least privilege access

f. Implementing role-based and attribute-based access control patterns

g. Automating credential rotation for machine identities (for example, Secrets Manager)

h. Managing permissions to control access to human and machine identities (for example, enabling multi-factor authentication [MFA], AWS Security Token Service [AWS STS], IAM profiles)

CHAPTER 18

AUTOMATION FOR SECURITY CONTROLS AND DATA PROTECTION

01 HRS

04 HRS

a. Network security components (for example, security groups, network ACLs, routing, AWS Network Firewall, AWS WAF, AWS Shield)

b. Certificates and public key infrastructure (PKI)

c. Data management (for example, data classification, encryption, key management, access controls)

d. Automating the application of security controls in multi-account and multi-Region environments (for example, Security Hub, Organizations, AWS Control Tower, Systems Manager)

e. Combining security controls to apply defense in depth (for example, AWS Certificate Manager [ACM], AWS WAF, AWS Config, AWS Config rules, Security Hub, GuardDuty, security groups, network ACLs, Amazon Detective, Network Firewall)

f. Automating the discovery of sensitive data at scale (for example, Amazon Macie)

g. Encrypting data in transit and data at rest (for example, AWS KMS, AWS CloudHSM, ACM)

ELYSIUM ACADEMY®
Milestone of Cognizance

Since 2007

CHAPTER 19

SECURITY MONITORING AND AUDITING SOLUTIONS

02 HRS

04 HRS

a. Security auditing services and features (for example, CloudTrail, AWS Config, VPC Flow Logs, CloudFormation drift detection)
b. AWS services for identifying security vulnerabilities and events (for example, GuardDuty, Amazon Inspector, IAM Access Analyzer, AWS Config)
c. Common cloud security threats (for example, insecure web traffic, exposed AWS access keys, S3 buckets with public access enabled or encryption disabled)
d. Implementing robust security auditing
e. Configuring alerting based on unexpected or anomalous security events
f. Configuring service and application logging (for example, CloudTrail, CloudWatch Logs)
g. Analyzing logs, metrics, and security findings

ELYSIUM ACADEMY®
Milestone of Cognizance

Since 2007

20 CHAPTER

INTRODUCTION TO DEVOPS

3.5 HRS

00 HRS

1) Why DevOps?
2) What is DevOps?
3) DevOps Market Trends?
4) DevOps Engineer Skills
5) DevOps Tools chain
6) Addressing Challenges through DevOps
7) Workflow of DevOps
8) DevOps Delivery Pipeline
9) DevOps Ecosystem

21 CHAPTER

VERSION CONTROL WITH GIT, JENKINS AND MAVEN INTEGRATION

03 HRS

04 HRS

1) What is version control?
2) What is Git?
3) Why Git for your organization
4) Install Git
5) Common commands in Git
6) Working with Remote Repositories
7) Advantages of Distributed VCS

22 CHAPTER

GIT, JENKINS & MAVEN INTEGRATION

2.5 HRS

08 HRS

1) Branching and Merging in Git
2) Git workflows
3) Git cheat sheet
4) What is CI
5) Why CI is Required
6) Introduction to Jenkins (With Architecture)
7) Introduction to Maven
8) Jenkins Management Preview
9) Adding a slave node to Jenkins
10) Build & Delivery Pipeline
11) Auto Deployment in Jenkins
12) Pipeline as a Code
13) Implementation of Jenkins in the Project

23 CHAPTER

CONTAINERIZATION WITH DOCKER

03 HRS

06 HRS

1) Docker overview
2) Installing Docker
3) Pulling images (Docker Pull)
4) Running images (Docker run)

5) Docker build and deployment- Connecting to running images (Docker exec)

6) Exposing volumes and ports

7) Inspecting system (Docker PS, docker status)

8) Using docker – compose to connect containers

9) Exposing volumes and ports

24
CHAPTER

CONFIGURATION MANAGEMENT WITH \ANSIBLE

03 HRS
06 HRS

1) Introduction to Ansible, Ansible mechanism.
2) Ansible installation in AWS instance. Ansible configuration, playing with ansible adhoc commands, creating simple play book, Playbook advanced – variables, loop, condition, roles.

ELYSIUM ACADEMY®
Milestone of Cognizance

Since 2007

25 CHAPTER

IAC TOOL – TERRAFORM

04 HRS

07 HRS

1) Introduction to Terraform
2) Installation of Terraform
3) Merging Terraform with AWS
4) Creating TF file
5) Building full cloud architecture using Terraform
6) Terraform backend
7) Terraform variables
8) Terraform state
9) Terraform locals
10) Terraform destroy

26 CHAPTER

APPLICATION MONITORING WITH PROMETHEUS & GRAFANA

11 HRS

07 HRS

1) PROMETHEUS
   a. Used for event monitoring & alerting
   b. Record real- time metrics in a time series Database (TSDB)
   c. Build using a HTTP pull model, with flexible queries and real-time alerting
   d. Node Exporter- Software that you can install on "NIX Kernel (Linux, openBSD, FreeBSD or Darwin)

O2) GRAFANA

a. Grafana is a multi-platform open source analytics and interactive visualization web application.

b. Provides charts, graphs and alerts for the web when connected to supported data sources

27
CHAPTER

LOG MONITORING WITH SPLUNK

02 HRS
06 HRS

1. Introduction to Splunk
2. Necessity of Logs
3. Why Splunk?
4. Splunk Components
5. Search Heads
6. Indexes
7. Forwarders
8. Installation of Splunk
9. Installation of Splunk Forwarder
10. Splunk Search
11. Splunk Alerts
12. Splunk Dashboards

ELYSIUM ACADEMY®
Milestone of Cognizance

Since 2007

28
CHAPTER

ORCHESTRATION TOOL – KUBERNETES

04 HRS

10 HRS

1. Kubernetes Introduction & Architecture
2. Kubernetes Installation – KOPS Method
3. Kubernetes – Clusters, Pod, Namespace
4. Deployment in Kubernetes
5. Kubernetes – Replica set, Demon Set, ConfigMap, Service
6. Services in K8s – Nodeport, ClusterIP, Load balancer, Ingress service
7. Persistent Volume, Persistent Volume Claim
8. Dashboards in Kubernetes

ELYSIUM ACADEMY®
Milestone of Cognizance

Since 2007

Placement Assistance 100%

135+ Professional Courses

Practical Sessions 90%

67+ Global Pacts

Corporate Placements 65%

170+ IT Companies Tie-Up

ELYSIUM GROUP OF COMPANIES

ELYSIUM ACADEMY PRIVATE LIMITED

Authorized International Partners

Microsoft · CompTIA · Cisco · aws · Google · SUSE · Rocheston · C++