# DevOps Engineer to Cloud Architect

## SKILLSOFT ASPIRE JOURNEY

skillsoft

Głównym wyzwaniem przed którym stają dziś organizacje na całym świecie jest konieczność ciągłego podnoszenia umiejętności i poziomu wiedzy w ślad za gwałtownym rozwojem nowych technologii i zmian na globalnym rynku.

Stały rozwój i podnoszenie kwalifikacji w IT od dawna jest już rzeczą oczywistą, a możliwość zapewnienia wsparcia specjalistom chcącym stale się rozwijać jest jedną z głównych kart przetargowych w walce o pracownika.

Na rynku liczą się dziś ludzie, którzy posiadają konkretne kompetencje i zestaw umiejętności pozwalający im wykonywać zadania efektywnie, a nie Ci z najdłuższym stażem pracy.

Dziś, bardziej niż kiedykolwiek w cenie jest umiejętność budowania ścieżki kariery dla profesjonalistów IT, którzy wciąż chcą się liczyć na rynku pracy.

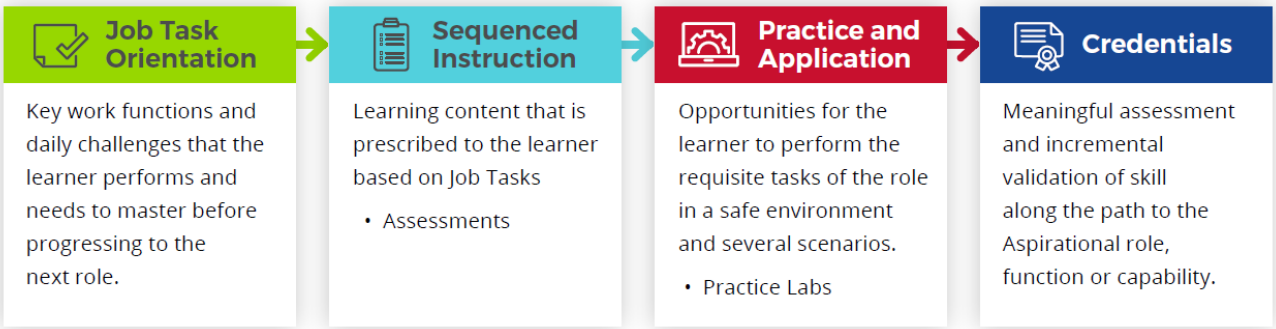
**Skillsoft Aspire Journey** stanowi odpowiedź na pytanie, jakie szkolenia muszę ukończyć, aby być przygotowanym do mojej wymarzonej pracy. Spośród kilkuset kanałów tematycznych dostępnych na naszej platformie szkoleniowej nasi specjaliści wybrali te, które naszym zdaniem najlepiej wyposażą uczących się w narzędzia potrzebne do realizacji zadań w nowej roli.

Skillsoft Aspire Journey to zestawy szkoleń i ćwiczeń w języku angielskim, które metodycznie, krok po kroku pozwalają specjalistom przejść od poziomu podstawowego do zaawansowanego.

Każda ścieżka zawiera szkolenia, laboratoria wirtualne, video i książki, które pomogą uczącym się osiągnąć pożądane kompetencje poświadczone certyfikatem.

**Aspire Journey Model**

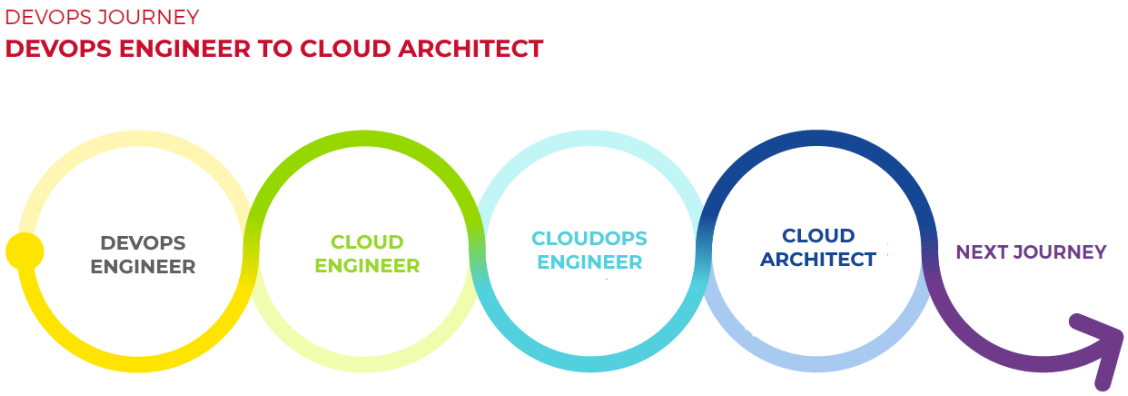
Cała ścieżka opiera się na 4-elementowym cyklu powtarzanym na kolejnych etapach nauki.




1. Określenie kluczowych funkcji i wyzwań, z którymi musi poradzić sobie uczący się w chwili obecnej, jak i tymi, z którymi przyjdzie mu się zmierzyć w nowej pracy.
2. Przejście zaprojektowanych ścieżek w proponowanej kolejności, wykonanie ćwiczeń i zaliczenie testów.
3. Przećwiczenie nowych umiejętności w kontrolowanym środowisku w oparciu o gotowe scenariusze działań. Laboratoria wirtualne Skillsoft
4. Certyfikat – zaliczenie testu końcowego na poziomie co najmniej 70% i uzyskanie certyfikatu potwierdzającego ukończenie danego etapu nauki.

# Aspire Journey – DevOps Engineer to Cloud Architect

Analizując trendy opisujące zachowanie użytkowników na naszych platformach szkoleniowych i współpracując ściśle z naszymi klientami na całym świecie Skillsoft wyselekcjonował najlepsze materiały szkoleniowe i ułożył je w ustrukturalizowaną ścieżkę rozwoju. Ścieżka zawiera ponad 72 godzin szkoleniowych.




10 courses
11h 45m 38s

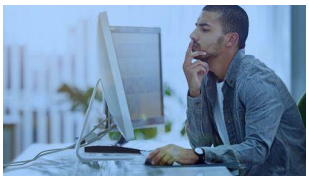
- DevOps mindset,
- promoting DevOps disciplines,
- Kanban for operations,
- Agile and DevOps, DevOps in continuous delivery and continuous integration,
- DevOps and AWS,
- Azure and DevOps,
- Google Cloud platform and Devops,
- infrastructure as code

10 courses
12h 31m 42s

- DevOps automation architect,
- DevOps Automation strategies and design patterns,
- AWS DevOps automation,
- Azure DevOps automation,
- Google Cloud Platform DevOps automation,
- DevOps automation Across platforms

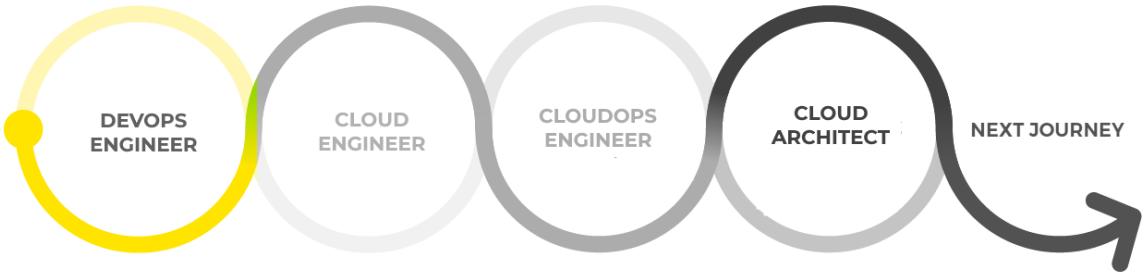
13 courses
13h 30m 17s

- moving from DevOps to CloudOps;
- elements of IaaS;
- public, private, and hybrid CloudOps interoperability;
- Docker and Kubernetes in multi-cloud environments;
- OpenStack in CloudOps;
- securing CloudOps deployment;
- CloudOps in software defined WANs



6 courses
5h 41m 7s

- cloud architecting,
- transitioning to cloud operations,
- CloudOps solutioning,
- CloudOps explainability,
- future trends in cloud computing




## Track 1: Cloud Engineer (duration: 11h 45m 38s)


### Adopting the DevOps Mindset

Objectives

- define the concept of DevOps and how DevOps practices can help eliminate the problems of traditional software development approaches
- describe different approaches of embracing the DevOps mindset and the value DevOps brings to software development projects
- differentiate between the mindsets that drive traditional and DevOps software development
- list key elements of the DevOps mindset and describe the roles of design thinking, Lean, and Agile in enabling and facilitating DevOps in the enterprise
- describe values of the DevOps mindset that play key roles in transforming software development practices to DevOps
- specify the benefits of transforming software development approaches with the DevOps mindset with a focus on agility and automation
- recognize the critical role played by the cloud in realizing the benefits the DevOps mindset
- recall the criteria for selecting on-premise, cloud, hybrid or multi-cloud architectures when adopting the continuous deployment principle of DevOps
- describe the DevOps methodologies, principles, and strategies that are used to build the end-to-end DevOps delivery lifecycle
- list tools that can be used to set up and implement end-to-end DevOps and CloudOps practices
- set up a DevOps framework to enable DevOps adoption in the cloud
- configure Bitbucket to implement code collaboration using code repository, build, and automated deployment
- set up configuration management tools to manage deployments on diversified targets, including on-premise and cloud


### DevOps Practices for the Enterprise

Objectives

- recall common application management roadblocks and describe how the adoption DevOps practices and principles can help resolve them
- compare DevOps and traditional IT management approaches from the perspective of driving software and system innovation
- describe Disciplined DevOps and the workflow of Disciplined DevOps, with focus on how Disciplined DevOps addresses challenges faced by modern organizations
- list critical change management elements and phases that enable enterprises to achieve better outcomes with the right cultural shift
- describe the paradigms used to provide complete management layers for delivering applications using the DevOps landscape
- describe how DevOps can be utilized to bring people, process, and technology together for digital transformation in every layer
- recognize prominent deployment strategies and differentiate between traditional, modern, and dynamic deployment approaches
- list the features of prominent tools and platforms used to set up traditional and DevOps deployment environments
- set up deployment projects using Atlassian tools
- apply DevOps principles to setting up modern deployment environments that are powered by containerization in the cloud
- recall the general release scheduling strategies that potentially support the Disciplined DevOps mindset
- set up release management solutions that can help track cross-project releases on a single board with custom release workflows

<table>
<tr>
<td> 

Kanban for Operations: Managing Projects Using Kanban</td>
<td> 

Agile and DevOps: Adopting Agile Methodology</td>
</tr>
<tr>
<td>Objectives<ul>
<li>recall the history of Kanban and recognize Kanban strategies for helping IT operations teams meet project timelines</li>
<li>describe the guiding principles and core practices of Kanban</li>
<li>recognize key terms and goals that are critical for a successful rollout of Kanban for operations</li>
<li>describe Kanban elements that are used to manage project execution, with focus on role, prioritization, blocker, and feedback</li>
<li>describe project tracking and forecasting techniques in Kanban project management</li>
<li>describe how Kanban enables value stream visualization and ensures stable flow of DevOps practices</li>
<li>recall approaches Waterfall teams can use to transition to Kanban practices</li>
<li>create Kanban projects and configure the projects to prioritize and visualize project workflows</li>
<li>list project management tools that can be used to implement Kanban for managing value streams in project implementations</li>
<li>create Kanban boards and configure the columns to monitor project work and implement continuous delivery of work</li>
<li>list important Kanban board metrics for tracking and measuring progress and optimizing productivity</li>
<li>use the Kanban project management approach to demonstrate Agile metrics for optimizing delivery</li>
<li>create and configure Kanban boards and visualize the flow of work to map teams' workflow stages</li>
<li>set WIP limits to constrain work in progress while using Kanban projects</li>
<li>apply Kanban approaches to manage DevOps practices for a productive DevOps team, with focus on work in progress, value stream mapping, and work items</li>
<li>describe the quality control and continuous improvement processes that are applied in Kanban</li>
</ul></td>
<td>Objectives<ul>
<li>recognize the objectives of Agile and the core Agile principles based on the Agile Manifesto</li>
<li>recognize Agile Manifesto values that can help elevate businesses to the next level</li>
<li>recognize the combined benefits of DevOps and Agile in managing large scale enterprise projects</li>
<li>differentiate between Agile and DevOps practices</li>
<li>create Scrum boards to implement the Agile project management methodology</li>
<li>create, configure, and start Sprints to manage projects and Backlogs</li>
<li>describe Scrum and the Scrum roles used for implementing Agile projects</li>
<li>list the types of reports that can help track Work in Progress, including Burndown reports and Velocity charts</li>
<li>recognize the steps involved in the visual requirements gathering process for Agile project management</li>
<li>describe the influence of architecture vision on team velocity and software quality, with a focus on the benefits of architecture vision</li>
<li>set up Agile product development projects and practices using Jira, while also specifying the product vision, goal, and project estimation with the use of the objective criteria technique</li>
<li>implement Agile practices using Microsoft Azure DevOps and TFS</li>
<li>list the key performance indicators and metrics that are used in Agile to evaluate the progress of development projects</li>
</ul></td>
</tr>
</table>

<table>
<tr>
<td> 

Adopting the DevOps CI/CD Paradigm</td>
<td> 

DevOps & AWS: CloudOps Implementation</td>
</tr>
<tr>
<td>Objectives<ul>
<li>describe the DevOps lifecycle and the patterns used to complement the Agile methodology and move towards adopting DevOps practices</li>
<li>describe continuous delivery and the benefits of adopting it</li>
<li>list the key principles and foundational practices that are essential in implementing continuous delivery</li>
<li>describe the role of architecture, culture, and patterns in adopting successful continuous delivery</li>
<li>list tools used to implement continuous delivery and their features</li>
<li>set up the continuous delivery architecture using required toolsets</li>
<li>describe continuous integration and the principles that can be applied to implement it</li>
<li>describe key practices for effectively implementing continuous integration</li>
<li>differentiate between continuous integration, continuous delivery, continuous deployment, and value stream mapping</li>
<li>use Bitbucket to set up repositories and pipelines to adopt continuous delivery workflows</li>
<li>set up continuous integration workflows in Bitbucket Pipelines using sample Node.js applications</li>
<li>build continuous delivery workflows with the use of the Branch-per-Issue model and Atlassian tools</li>
<li>implement continuous deployment pipelines with Bitbucket Pipelines</li>
<li>describe Git branching strategies that can help achieve continuous delivery</li>
<li>recognize how the Continuous Delivery Maturity Model can help build and implement Enterprise Continuous Delivery, with a focus on the Build, Deploy, Test, and Report phases</li>
<li>list Application Release Automation components and benefits</li>
<li>recognize the criteria for selecting code repository solutions that can compliment organizational strategies</li>
<li>build GitFlow workflows to design branching models and enable scheduled release cycles</li>
<li>demonstrate the use of GitLab as a complete DevOps platform for end-to-end DevOps lifecycle implementation</li>
<li>demonstrate use of CI/CD over containerization to drive down pre-production costs</li>
</ul></td>
<td>Objectives<ul>
<li>recognize the relationship between AWS and DevOps</li>
<li>recognize the benefits of using AWS for CloudOps and DevOps implementation</li>
<li>list AWS services that provide tooling support for implementing end-to-end CloudOps lifecycle</li>
<li>describe AWS services that are used to provision, configure, and manage AWS infrastructure resources using code and templates, while also monitoring and enforcing infrastructure compliance</li>
<li>describe continuous integration and continuous delivery implementation approaches using AWS services</li>
<li>set up continuous integration workflows with AWS CodePipeline to build code with AWS CodeBuild</li>
<li>use CodeCommit to manage code repositories that can be used in CloudOps implementation</li>
<li>use AWS CodeStar to develop, build, and deploy applications in AWS</li>
<li>recognize application deployment targets in AWS</li>
<li>implement AWS CodePipeline in the AWS Cloud9 Integrated Development Environment</li>
<li>work with AWS X-Ray to visualize application components, and identify and troubleshoot performance bottlenecks</li>
<li>build dynamic and enterprise-ready CI/CD environments for serverless application deployment in AWS</li>
</ul></td>
</tr>
</table>


CloudOps with Azure DevOps Tools

Objectives

- recall DevOps and CloudOps concepts and describe the DevOps application lifecycle and the Azure DevOps reference model
- list Azure DevOps tool features that can be used for software development with source control, work tracking, and continuous integration and delivery
- use Azure Boards to manage software projects by tracking stories, Backlogs, features, and associated bugs
- install and configure Azure Boards for GitHub to manage the connection of Azure Board projects with GitHub repositories
- create local Git repositories and synchronize them with centralized Git repositories in Azure DevOps
- configure continuous integration and continuous deployment for applications using build and release in Azure Pipelines
- work with Azure Artifacts to discover, install, and publish NuGet, NPM, and Maven packages in Azure DevOps
- use Azure DevOps to create test plans for validating applications in order to manage the software development project testing lifecycle
- monitor application performance using Application Insights
- deploy applications to AWS through Azure DevOps
- recognize the approach of shifting security mindsets to DevSecOps culture with focus on various security strategy components
- describe the approach of applying Azure DevOps principles to transform monolithic solutions to cloud solutions


CloudOps with Google Cloud Platform Tools

Objectives

- describe the technical, procedural, measurement, and cultural capabilities of Google Cloud Platform that drive higher software delivery with CloudOps
- recognize Google Cloud Platform solutions for end-to-end CloudOps implementation
- specify the benefits of and approaches for implementing CloudOps on Google Cloud Platform
- create and manage a source code repository using the Google Cloud Console and gcloud command line tool
- describe Cloud Build features that can be used to implement continuous integration
- list Google Cloud Platform configuration management tools and their features
- build configuration management workflows using a single machine running Chef Client
- describe the Google Cloud Platform reference pipeline for continuous integration that helps in delivering quality software and systems
- use Cloud Build and GitHub to automate continuous integration workflows for serverless applications
- describe continuous delivery pipelines and the benefits of using Google Cloud Platform for continuous delivery
- automate App Engine deployments with Cloud Build and Cloud Source Repositories
- create continuous delivery pipelines using Google Kubernetes Engine, Cloud Source Repositories, Cloud Build, and Spinnaker for Google


CloudOps: Infrastructure as Code

Objectives

- recognize the deficiencies associated with manual resource configuration and how to use Infrastructure as Code to eliminate those deficiencies
- recall the essential stages involved in the lifecycle of Infrastructure as Code and with the roles played by each of those stages
- recall the history of Infrastructure as Code and describe the transformation of Infrastructure as Code from DevOps to the CloudOps paradigm
- describe the concept of Policy as Code, including the benefits of using Policy as Code to implement Infrastructure as Code
- list benefits and best practices for implementing Infrastructure as Code to realize CloudOps
- specify the important CloudOps principles that need to be adopted to organize Infrastructure as Code
- list prominent tools and services that can be used to implement Infrastructure as Code while applying the required DevOps and CloudOps policies
- define the concept of Continuous Configuration Automation and recognize tools that can be used to automate and configure various layers of infrastructure
- describe Chef features and its components and tools that can be used to implement Continuous Configuration Automation
- install Chef and write cookbooks that can be used to manage provisioning of resources
- describe Infrastructure as a Service and differentiate between Infrastructure as a Service and Infrastructure as Code

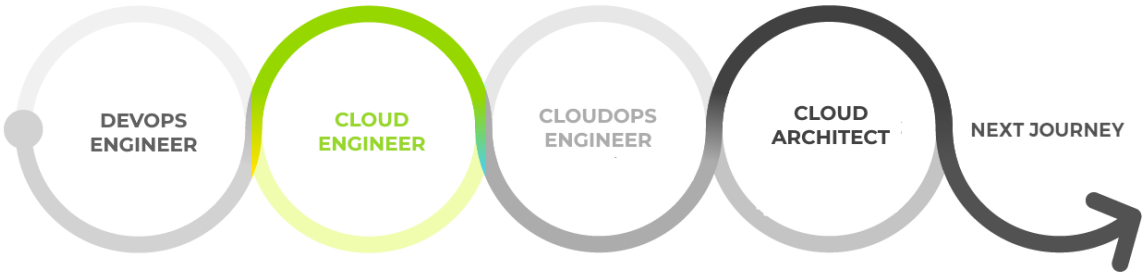
implement of Infrastructure as Code using CloudFormation templates in AWS

Final Exam: DevOps Engineer

Objectives

- build configuration management workflows using a single machine running Chef Client
- compare DevOps and traditional IT management approaches from the perspective of driving software and system innovation
- create and manage a source code repository using the Google Cloud Console and gcloud command-line tool
- create, configure, and start Sprints to manage projects and Backlogs
- create Kanban projects and configure the projects to prioritize and visualize project workflows
- create local Git repositories and synchronize them with centralized Git repositories in Azure DevOps
- define the concept of DevOps and how DevOps practices can help eliminate the problems of traditional software development approaches
- demonstrate the use of GitLab as a complete DevOps platform for end-to-end DevOps lifecycle implementation
- describe Chef features and its components and tools that can be used to implement Continuous Configuration Automation
- describe Cloud Build features that can be used to implement continuous integration
- describe continuous delivery and the benefits of adopting it
- describe continuous integration and continuous delivery implementation approaches using AWS services
- describe continuous integration and the principles that can be applied to implement it
- describe different approaches of embracing the DevOps mindset and the value DevOps brings to software development projects
- describe Disciplined DevOps and the workflow of Disciplined DevOps, with focus on how Disciplined DevOps addresses challenges faced by modern organizations
- describe Git branching strategies that can help achieve continuous delivery
- describe Kanban elements that are used to manage project execution, with a focus on role, prioritization, blocker, and feedback
- describe project tracking and forecasting techniques in Kanban project management
- describe Scrum and the Scrum roles used for implementing Agile projects

- describe the DevOps methodologies, principles, and strategies that are used to build the end-to-end DevOps delivery lifecycle
- describe the guiding principles and core practices of Kanban
- describe the paradigms used to provide complete management layers for delivering applications using the DevOps landscape
- describe the technical, procedural, measurement, and cultural capabilities of Google Cloud Platform that drive higher software delivery with CloudOps
- differentiate between Agile and DevOps practices
- differentiate between continuous integration, continuous delivery, continuous deployment, and value stream mapping
- differentiate between the mindsets that drive traditional and DevOps software development
- install and configure Azure Boards for GitHub to manage the connection of Azure Board projects with GitHub repositories
- install Chef and write cookbooks that can be used to manage the provisioning of resources
- list AWS services that provide tooling support for implementing end-to-end CloudOps lifecycle
- list Azure DevOps tool features that can be used for software development with source control, work tracking, and continuous integration and delivery
- list key elements of the DevOps mindset and describe the roles of design thinking, Lean, and Agile in enabling and facilitating DevOps in the enterprise
- list project management tools that can be used to implement Kanban for managing value streams in project implementations
- list prominent tools and services that can be used to implement Infrastructure as Code while applying the required DevOps and CloudOps policies
- list the features of prominent tools and platforms used to set up traditional and DevOps deployment environments
- list the key principles and foundational practices that are essential in implementing continuous delivery
- list tools used to implement continuous delivery and their features
- recall approaches Waterfall teams can use to transition to Kanban practices
- recall common application management roadblocks and describe how the adoption DevOps practices and principles can help resolve them
- recall DevOps and CloudOps concepts and describe the DevOps application lifecycle and the Azure DevOps reference model
- recall the essential stages involved in the lifecycle of Infrastructure as Code and with the roles played by each of those stages
- recall the history of Infrastructure as Code and describe the transformation of Infrastructure as Code from DevOps to the CloudOps paradigm
- recall the history of Kanban and recognize Kanban strategies for helping IT operations teams meet project timelines
- recognize Agile Manifesto values that can help elevate businesses to the next level
- recognize how the Continuous Delivery Maturity Model can help build and implement Enterprise Continuous Delivery, with a focus on the Build, Deploy, Test, and Report phases
- recognize key terms and goals that are critical for a successful rollout of Kanban for operations
- recognize the approach of shifting security mindsets to DevSecOps culture with a focus on various security strategy components
- recognize the benefits of using AWS for CloudOps and DevOps implementation
- recognize the combined benefits of DevOps and Agile in managing large scale enterprise projects
- recognize the criteria for selecting code repository solutions that can complement organizational strategies
- recognize the deficiencies associated with manual resource configuration and how to use Infrastructure as Code to eliminate those deficiencies
- recognize the objectives of Agile and the core Agile principles based on the Agile Manifesto
- recognize the relationship between AWS and DevOps
- set up deployment projects using Atlassian tools
- set up the continuous delivery architecture using required toolsets
- specify the benefits of transforming software development approaches with the DevOps mindset with a focus on agility and automation
- use AWS CodeStar to develop, build, and deploy applications in AWS
- use Azure Boards to manage software projects by tracking stories, Backlogs, features, and associated bugs
- use Cloud Build and GitHub to automate continuous integration workflows for serverless applications
- use Cloud Build to automate continuous integration workflows for serverless applications
- use CodeCommit to manage code repositories that can be used in CloudOps implementation




## Track 2: Cloud Engineer (duration: 12h 31m 42s)

<table>
<tr>
<td> 

Adopting IT Automation</td>
<td> 

Applying Design Patterns in DevOps & CloudOps</td>
</tr>
<tr>
<td>Objectives<br>▪ recognize the IT automation process and potential applications that can be automated<br>▪ describe the advantages, disadvantages, and best practices of automation that should be applied for productive IT operations<br>▪ recognize how to develop enterprise automation capabilities that can help deliver controlled self-service capabilities for managing IT processes and infrastructure<br>▪ define key processes that need to be followed to implement effective IT automation strategy<br>▪ list the key principles of DevOps and the approach of automating DevOps process using integration technologies<br>▪ identify the role of automation in DevOps and the benefits of enabling standardization in DevOps automation<br>▪ recognize the role of the DevOps Automation Architect along with their critical responsibilities<br>▪ list the essential benefits of deployment automation and prominent approaches of implementing deployment automation<br>▪ describe the benefits of cloud automation and common cloud tasks that are a good candidate for automation<br>▪ recall the prominent tools that can be used to automate DevOps practises with focus on CI/CD and continuous monitoring<br>▪ install and configure SaltStack to automate Infrastructure provisioning<br>▪ automate static site deployment using AWS CloudFormation</td>
<td>Objectives<br>▪ recall common application management roadblocks and describe how the adoption DevOps practices and principles can help resolve them<br>▪ compare DevOps and traditional IT management approaches from the perspective of driving software and system innovation<br>▪ describe Disciplined DevOps and the workflow of Disciplined DevOps, with focus on how Disciplined DevOps addresses challenges faced by modern organizations<br>▪ list critical change management elements and phases that enable enterprises to achieve better outcomes with the right cultural shift<br>▪ describe the paradigms used to provide complete management layers for delivering applications using the DevOps landscape<br>▪ describe how DevOps can be utilized to bring people, process, and technology together for digital transformation in every layer<br>▪ recognize prominent deployment strategies and differentiate between traditional, modern, and dynamic deployment approaches<br>▪ list the features of prominent tools and platforms used to set up traditional and DevOps deployment environments<br>▪ set up deployment projects using Atlassian tools<br>▪ apply DevOps principles to setting up modern deployment environments that are powered by containerization in the cloud<br>▪ recall the general release scheduling strategies that potentially support the Disciplined DevOps mindset<br>▪ set up release management solutions that can help track cross-project releases on a single board with custom release workflows</td>
</tr>
</table>

<table>
<tr>
<td> 

Using AWS to Set Up DevOps and CloudOps Automation</td>
<td> 

Applying Automation Using AWS Tools</td>
</tr>
<tr>
<td>Objectives<ul>
<li>recall the history of Kanban and recognize Kanban strategies for helping IT operations teams meet project timelines</li>
<li>describe the guiding principles and core practices of Kanban</li>
<li>recognize key terms and goals that are critical for a successful rollout of Kanban for operations</li>
<li>describe Kanban elements that are used to manage project execution, with focus on role, prioritization, blocker, and feedback</li>
<li>describe project tracking and forecasting techniques in Kanban project management</li>
<li>describe how Kanban enables value stream visualization and ensures stable flow of DevOps practices</li>
<li>recall approaches Waterfall teams can use to transition to Kanban practices</li>
<li>create Kanban projects and configure the projects to prioritize and visualize project workflows</li>
<li>list project management tools that can be used to implement Kanban for managing value streams in project implementations</li>
<li>create Kanban boards and configure the columns to monitor project work and implement continuous delivery of work</li>
<li>list important Kanban board metrics for tracking and measuring progress and optimizing productivity</li>
<li>use the Kanban project management approach to demonstrate Agile metrics for optimizing delivery</li>
<li>create and configure Kanban boards and visualize the flow of work to map teams' workflow stages</li>
<li>set WIP limits to constrain work in progress while using Kanban projects</li>
<li>apply Kanban approaches to manage DevOps practices for a productive DevOps team, with focus on work in progress, value stream mapping, and work items</li>
<li>describe the quality control and continuous improvement processes that are applied in Kanban</li>
</ul></td>
<td>Objectives<ul>
<li>recognize the objectives of Agile and the core Agile principles based on the Agile Manifesto</li>
<li>recognize Agile Manifesto values that can help elevate businesses to the next level</li>
<li>recognize the combined benefits of DevOps and Agile in managing large scale enterprise projects</li>
<li>differentiate between Agile and DevOps practices</li>
<li>create Scrum boards to implement the Agile project management methodology</li>
<li>create, configure, and start Sprints to manage projects and Backlogs</li>
<li>describe Scrum and the Scrum roles used for implementing Agile projects</li>
<li>list the types of reports that can help track Work in Progress, including Burndown reports and Velocity charts</li>
<li>recognize the steps involved in the visual requirements gathering process for Agile project management</li>
<li>describe the influence of architecture vision on team velocity and software quality, with a focus on the benefits of architecture vision</li>
<li>set up Agile product development projects and practices using Jira, while also specifying the product vision, goal, and project estimation with the use of the objective criteria technique</li>
<li>implement Agile practices using Microsoft Azure DevOps and TFS</li>
<li>list the key performance indicators and metrics that are used in Agile to evaluate the progress of development projects</li>
</ul></td>
</tr>
</table>

<table>
<tr>
<td> 

Azure DevOps: Repository & Pipeline Management</td>
<td> 

Azure DevOps: Managing Agile Lifecycle</td>
</tr>
<tr>
<td>Objectives<ul>
<li>recognize the objectives of the Topology and Orchestration Specification for Cloud Application and methodical frameworks that can be used for DevOps artifact transformation with the TOSCA standard</li>
<li>recognize the DevOps workflow and the Azure-specific patterns used for automating DevOps workflows</li>
<li>list Azure tools and technologies that help with implementing end-to-end DevOps and CloudOps practices in each phase of the lifecycle</li>
<li>recognize the general workflow of version control and the different types of version control provided by Azure Repos</li>
<li>list the key elements and artifacts that are used in Azure Repos to manage version control and workflows</li>
<li>create local Git repositories and synchronize them with centralized Git repositories in Azure DevOps</li>
<li>create and manage branches using Azure DevOps</li>
<li>build GitHub repositories and applications using Azure Pipelines from the command line</li>
<li>manage technical debts using Azure DevOps and SonarQube to implement continuous inspection of code quality and review</li>
<li>implement CI/CD for Node.js applications using Azure Pipelines</li>
<li>automate and execute test cases for web applications as a part of the Azure DevOps release pipelines</li>
<li>create Azure CI/CD pipelines to enable end-to-end automation with Azure DevOps</li>
</ul></td>
<td>Objectives<ul>
<li>set up continuous deployment of containerized applications for Azure Kubernetes Service using Azure Pipelines</li>
<li>recall Azure Boards capabilities that can be used to plan, track, and monitor Sprints following the Agile methodology</li>
<li>demonstrate Agile planning and portfolio management using Azure Boards tools and processes that can be used to help plan, manage, and track work across teams</li>
<li>describe the concept of Azure Test Plans from the perspective of manual and automated testing</li>
<li>create various test artifacts that include elements like test plan, test suite, test case, test configuration, and parameters for managing backlog items</li>
<li>demonstrate the exploratory testing and feedback management capabilities of Azure Test Plans</li>
<li>recall features of Azure Artifacts and best practices that can be used to manage and share packages</li>
<li>discover, install, and publish NuGet, NPM, and Maven packages in Azure DevOps</li>
<li>deploy multi-container applications to Azure Kubernetes Services</li>
<li>recognize the configuration management and monitoring capabilities and tools of Azure DevOps</li>
<li>use Ansible Solution Templates to configure Ansible instances on virtual machines along with Ansible and a suite of tools configured to work with Azure</li>
<li>configure Azure Kubernetes Service clusters in Azure using Ansible and use Playbooks to create resource groups and AKS clusters within the resource groups</li>
<li>describe the various DevOps Solution Architectures that are derived using various Azure DevOps tools and technologies</li>
<li>use Azure Blueprint to create, deploy, and update compliant environments</li>
</ul></td>
</tr>
</table>


## GCP DevOps: CloudOps with Google Cloud Platform

Objectives

- recognize Google Cloud Platform features and why Google Cloud Platform is a secondary cloud provider
- describe the key features of Google Kubernetes Engine and how it can be used to set up CloudOps to manage operations that scale and manage workload
- list best practices for operating containers inspired by the Twelve-Factor App methodology and recognize how to build cloud-native applications using the CloudOps methodology
- create and configure Google Kubernetes Engine clusters in default VPC and enable alias IP addresses
- deploy applications across multiple Kubernetes clusters using an Istio multi-cluster service mesh
- describe the features of Google Cloud Source Repository and compare its capabilities with GitHub
- create repositories using the Cloud Source Repository and manage code using the checkin/checkout and merge strategies
- deploy applications from the Cloud Source Repository to App Engine
- compare the automated deployment capabilities of Google Cloud Deployment Manager and Hashicorp Terraform
- use the Google Cloud Deployment Manager to automate the configuration of Google Cloud Platform resources to deploy applications on targeted environments
- use Terraform to create virtual machine instances on the Google Cloud Platform and build applications on the instances
- recognize the machine learning, artificial intelligence, and analytical capabilities of Google Cloud Platform, with focus on the toolsets that enable edge feature implementations like artificial intelligence, IoT, and business intelligence
- configure AutoML and BigQuery to manage large scale of data and provide high quality machine learning models following the CloudOps paradigm
- configure Cloud Pub/Sub to set up fully-managed real-time messaging environments to send and receive messages between independent applications
- recognize the concept of multi-cloud design along with the essential design patterns for connecting Google Cloud Platform with other cloud platforms
- establish connectivity and transfer data between Google Cloud Platform and other cloud service providers using external IP addresses


## DevOps Automation Across Platforms: CloudOps for Multi-cloud Deployments

Objectives

- define multi-cloud and list the benefits of adopting the multi-cloud strategy
- describe multi-cloud architecture design, which can be used to derive effective cloud strategy for application management on multi-cloud environments
- recognize the opportunities and challenges of hybrid and multi-cloud deployments, along with the critical drivers for hybrid cloud and multi-cloud setups
- recognize common hybrid and multi-cloud architecture patterns and scenarios where they can be applied
- list the common network topologies that can be used for hybrid and multi-cloud setup
- recognize the critical challenges associated with multi-cloud environments and the frameworks that can be used to eliminate these challenges
- describe CloudOps characteristics that can be applied to build a cloud management platform for multi-cloud environments
- list the AWS tools that can be used to implement multi-cloud environment CloudOps and their associated features
- list the Azure tools that can be used to implement multi-cloud environment CloudOps and their associated features
- list the Google Cloud Platform tools that can be used to implement multi-cloud environment CloudOps and their associated features
- create a multi-cloud environment with Azure and AWS
- describe the challenges associated with multi-cloud in driving CI/CD across public and private clouds
- identify the critical patterns that can be used to build cloud-specific automations in DevOps pipelines and list the steps involved in setting up DevOps-style multi-cloud deployments
- describe the multi-cloud DevOps Framework that provides a DevOps solution for managing multi-cloud distributed environments


DevOps Automation Across Platforms: Working with Multi-cloud Tools

Objectives

- list the major DevOps tools that can be used or adapted for cross-cloud multi-cloud deployments
- recognize the process of implementing CI/CD DevOps pipelines that employ multiple cloud environments, including source code, monitoring, and tools that are used in each step
- integrate Azure Active Directory with multiple Amazon Web Services accounts
- use Azure Pipelines to realize the full DevOps cycle with multi-cloud support
- setup and configure the CloudOps CI/CD process using Spinnaker to implement the multi-cloud environment
- install and configure Jenkins to support multi-cloud environments and deploy code to multi-cloud environments using containerization
- list and compare the tools that can be used to monitor metrics across multi-cloud platforms and environments
- create secure, private, and site-to-site connections between Google Cloud Platform and Amazon Web Services using Virtual Private Networks to setup multi-cloud environments
- set up and monitor multi-cloud architecture involving Azure and AWS using New Relic
- recognize the challenges associated with multi-cloud disaster recovery setup and patterns that can be used to eliminate those challenges
- set up cloud disaster recovery using Google Cloud Platform and AWS

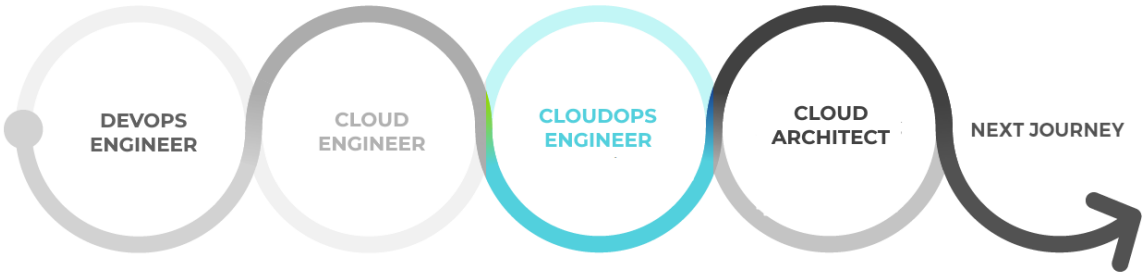
Final Exam: Cloud Engineer

Objectives

- build GitHub repositories and applications using Azure Pipelines from the command line
- configure and manage Amazon EC2 systems with the AWS Systems Manager
- create and manage branches using Azure DevOps
- create local Git repositories and synchronize them with centralized Git repositories in Azure DevOps
- create repositories using the Cloud Source Repository and manage code using the check-in/checkout and merge strategies
- create various test artifacts that include elements like test plan, test suite, test case, test configuration, and parameters for managing backlog items
- define key processes that need to be followed to implement effective IT automation strategy
- define multi-cloud and list the benefits of adopting the multi-cloud strategy
- demonstrate the exploratory testing and feedback management capabilities of Azure Test Plans
- describe CloudOps characteristics that can be applied to build a cloud management platform for multi-cloud environments
- describe multi-cloud architecture design, which can be used to derive an effective cloud strategy for application management on multi-cloud environments
- describe the advantages, disadvantages, and best practices of automation that should be applied for productive IT operations
- describe the benefits of cloud automation and common cloud tasks that are a good candidate for automation
- describe the concept of Azure Test Plans from the perspective of manual and automated testing
- describe the evolution of design patterns and list the elements that are used to share feasible solutions for generic problems
- describe the features of Google Cloud Source Repository and compare its capabilities with GitHub
- describe the key features of Google Kubernetes Engine and how it can be used to set up CloudOps to manage operations that scale and manage workload
- identify the role of automation in DevOps and the benefits of enabling standardization in DevOps automation
- implement continuous security monitoring using AWS GuardDuty
- install and configure Jenkins to support multi-cloud environments and deploy code to multi-cloud environments using containerization

- integrate Azure Active Directory with multiple Amazon Web Services accounts
- list Azure tools and technologies that help with implementing end-to-end DevOps and CloudOps practices in each phase of the lifecycle
- list best practices for operating containers inspired by the Twelve-Factor App methodology and recognize how to build cloud-native applications using the CloudOps methodology
- list prominent AWS services that are used to facilitate the configuration management of multiple instances in AWS along with their associated features
- list the AWS tools that can be used to implement multi-cloud environment CloudOps and their associated features
- list the Azure tools that can be used to implement multi-cloud environment CloudOps and their associated features
- list the design patterns than can be adopted to enable DevOps and CloudOps in the enterprise
- list the factors driving the adoption of DevOps and CloudOps along with the approaches that are used to implement automation
- list the features and benefits of AWS Stack that compliment AWS DevOps and CloudOps adoption, as well as critical automation patterns of implementing AWS DevOps and CloudOps
- list the Google Cloud Platform tools that can be used to implement multi-cloud environment CloudOps and their associated features
- list the major DevOps tools that can be used or adapted for cross-cloud multi-cloud deployments
- manage applications using AWS CodeStar, source repository, a continuous deployment toolchain, and a project dashboard
- recall Azure Boards capabilities that can be used to plan, track, and monitor Sprints following the Agile methodology
- recall critical DevOps and CloudOps continuous deployment patterns and recognize how to implement continuous deployment pipelines
- recall DevOps topologies and tools that are used to set up DevOps architecture to improve value delivery for customers and the business
- recall features of Azure Artifacts and best practices that can be used to manage and share packages
- recall the challenges associated with DevOps and CloudOps and describe the correct approach of using AWS to implement DevOps and CloudOps in the enterprise to realize a positive impact on business
- recall the concept of Infrastructure as Code and the role of AWS CloudFormation in implementing and managing Infrastructure as Code
- recall the critical features of AWS CodeDeploy that can help automate software deployments to a variety of AWS compute services
- recognize AWS CodeStar features that can help facilitate developing, building, and deploying applications in AWS
- recognize AWS Config features that are used to assess, audit, and evaluate the configuration of AWS resources
- recognize critical DevOps and CloudOps patterns that are prominently used to scale applications using cloud services
- recognize effective approaches to automating IT processes using the AWS DevOps Reference Architecture
- recognize Google Cloud Platform features and why Google Cloud Platform is a secondary cloud provider
- recognize how to develop enterprise automation capabilities that can help deliver controlled self-service capabilities for managing IT processes and infrastructure
- recognize the challenges associated with multi-cloud disaster recovery setup and patterns that can be used to eliminate those challenges
- recognize the concept of multi-cloud design along with the essential design patterns for connecting Google Cloud Platform with other cloud platforms
- recognize the critical challenges associated with multi-cloud environments and the frameworks that can be used to eliminate these challenges
- recognize the general workflow of version control and the different types of version control provided by Azure Repos
- recognize the IT automation process and potential applications that can be automated
- recognize the objectives of the Topology and Orchestration Specification for Cloud Application and methodical frameworks that can be used for DevOps artifact transformation with the TOSCA standard
- recognize the process of implementing CI/CD DevOps pipelines that employ multiple cloud environments, including source code, monitoring, and tools that are used in each step
- recognize the strategy for implementing and improving DevOps practices and recognize the tools that are used across various development and operations processes in the enterprise
- set up AWS CodeDeploy to automate the deployment of applications

- set up cloud disaster recovery using Google Cloud Platform and AWS
- set up continuous deployment of containerized applications for Azure Kubernetes Service using Azure Pipelines
- use AWS Config and Config rules to monitor and enforce compliance for infrastructures
- use Azure Blueprint to create, deploy, and update compliant environments
- use Terraform to create virtual machine instances on the Google Cloud Platform and build applications on the instances
- use the Google Cloud Deployment Manager to automate the configuration of Google Cloud Platform resources to deploy applications on targeted environments




## Track 3: CloudOps Engineer (duration: 13h 30m 17s)

|  DevOps to CloudOps for Multi-cloud |  Implementing IaaS & Orchestration for Multi-cloud Environments |
|---|---|
| Objectives:<br>▪ compare the differences between DevOps and CloudOps along with challenges and best practices associated with the implementation of CloudOps<br>▪ recognize the cloud capabilities that needs to be integrated in existing DevOps implementations to transform DevOps to CloudOps<br>▪ identify the characteristics of prominent cloud management platforms that drives successful implementation and adaption of CloudOps<br>▪ describe the architectural designs that can lead to effective multi-cloud strategy along with the challenges managed by multi-cloud strategy<br>▪ compare the differences between hybrid cloud and multi-cloud and describe the pattern-based multi-cloud architecture migration<br>▪ list the prominent multi-cloud management tools that we can use to implement CloudOps along with the recommended criteria for selecting the right multi-cloud management tool<br>▪ describe the approach of implementing and optimizing multi-cloud strategy in the enterprise<br>▪ recognize the role played by CloudOps automation to drive business transformation in multi-cloud<br>▪ describe the CloudOps design scenarios that are used in the enterprise to manage critical architectural and operational requirement of multi-cloud strategy<br>▪ recall the approaches of building DevOps pipelines for multi-cloud application deployment<br>▪ specify the key features of the multi-cloud management model and define the architecture of Cloud Service Brokerage platform that are used to eliminate operational challenges of multi-cloud<br>▪ recognize the challenges of designing CloudOps solution for multi-cloud and describe the optimized design of multi-cloud CloudOps framework to mitigate the challenges from the perspective of release management | Objectives:<br>▪ define IaaS along with the architecture of IaaS and the key benefits that we can realize with the adoption of IaaS with CloudOps<br>▪ recall the core cloud services and compare the capabilities of IaaS with SaaS and PaaS<br>▪ list the core components of IaaS platform and architecture and compare the capabilities of IaaS with Bare Metal as a Service<br>▪ describe the features of the declarative and imperative approaches of implementing Infrastructure as Code along with the right fit scenarios of adopting them<br>▪ list the popular IaaS providers that provides infrastructure resources like servers, network connections, storage along with critical features like content delivery networks and load balancing<br>▪ recognize the role of orchestration in implementing CloudOps for heterogeneous deployments<br>▪ recognize the relation between orchestration and automation along with the benefits of cloud orchestration from the perspective of multi-cloud CloudOps adoption<br>▪ identify the categories of cloud automation and orchestration tools from the perspective of cloud provider-native and third-party tools<br>▪ recall the multi-cloud strategies and tools that we can use to plan and architect Infrastructure as a Service and orchestration in order to optimize CloudOps resource provisioning and costs<br>▪ define the concept of containers along with the benefits of running containers in multi-cloud environments<br>▪ list the APIs and tools that are used in running containers in multi-cloud environments along with the key aspects that we have to consider when choosing cloud infrastructures for containerized applications<br>▪ recognize the role of Kubernetes in container management for mult-icloud and the impact of containerization on multi-cloud and edge computing requirements like AI and IoT<br>▪ install and configure OpenStack on Ubuntu LTS to create multi-cloud models<br>▪ install OpenNebula and configure the OpenNebula IaaS cloud installation on Ubuntu LTS to create multi-cloud models |

<table>
<tr>
<td> 

CloudOps Interoperability: Modeling Cloud Computing for Integration</td>
<td> 

CloudOps Interoperability: Inter-cloud Integration & Implementation</td>
</tr>
<tr>
<td>Objectives:<ul>
<li>describe the role of standards in cloud computing interoperability and list cloud computing interoperability use cases supported by standards</li>
<li>list the categories of cloud computing portability and interoperability with focus on data, application, platform, and infrastructure components</li>
<li>describe the interfaces of the Distributed Computing Reference Model along with the standardization required for cloud computing portability and interoperability</li>
<li>recall application design principles that can help reduce complexity and provide better interoperability for diversified applications that are designed for cloud computing</li>
<li>describe the Cloud Ecosystem Reference Model and its taxonomy to design interoperable cloud architectures</li>
<li>recognize the cloud ecosystem Enterprise Architecture Principles that ensure consistency and integrity of the enterprise architecture in managing the life cycle of cloud services across the enterprise</li>
<li>list the prominent cloud deployment models and describe scenarios for adopting them</li>
<li>specify the benefits of achieving cloud interoperability and maximizing portability, along with the challenges of achieving interoperability between diversified cloud components</li>
<li>describe the key elements of interoperability and portability for cloud services, as well as scenarios that depict interoperability and portability considerations and requirements and how to address them</li>
<li>compare the features exhibited by public, private, and hybrid cloud that helps CloudOps Architects decide the right environment for application deployment</li>
<li>recall important considerations for defining hybrid cloud strategy, with focus on hybrid cloud integration, management, and CloudOps adoption</li>
<li>recognize the process-oriented classification of cloud migration that can help migrate selected architectural components from on-premise to cloud</li>
</ul></td>
<td>Objectives:<ul>
<li>recognize considerations for adopting cloud deployment strategies and appropriate scenarios for adopting private and public cloud for application deployment</li>
<li>list and compare the prominent cloud offerings that enable enterprises to deploy and manage a wide array of private, public, and hybrid cloud environments</li>
<li>describe the prominent types of cloud integration and their benefits in creating and operating cloud-to-cloud integration</li>
<li>describe recommended approaches of cloud application integration and list the features of prominent cloud-based integration tools that enable handling the increasing complexity of IT frameworks by connecting diverse and disparate data and services</li>
<li>recognize the benefits of using Integration Platform as a Service to integrate multi-cloud environments</li>
<li>install and configure the CloudHub platform service to deploy applications with a fully-managed and highly available cloud infrastructure</li>
<li>set up the Talend environment to build portable multi-cloud integration workflows</li>
<li>describe the features of the prominent types of inter-cloud implementations and list the popular topologies of different cloud architectures</li>
<li>describe architectural patterns that can be adopted to plan multi-cloud migration and deployments</li>
<li>recognize the multi-cloud implementation workflow that can help design automated and portable architectures for application delivery</li>
<li>demonstrate the steps involved in integrating AWS with Azure Active Directory</li>
<li>demonstrate how to use Aviatrix to integrate Azure with AWS and Google Cloud Platform</li>
</ul></td>
</tr>
</table>

<table>
<tr>
<td> 

Docker & Multi-cloud: Managing Multi-cloud with Docker</td>
<td> 

Docker & Multi-cloud: Multi-host, Multi-cloud Management with Docker Enterprise</td>
</tr>
<tr>
<td>Objectives:<ul>
<li>recall the architecture of Docker along with the features afforded by Docker components</li>
<li>recall the benefits afforded by Docker and list the prominent open source tools that can be used by architects for productive container implementations</li>
<li>implement container linking to allow multiple containers to link with each other</li>
<li>recognize the critical challenges of setting up multi-cloud container management along with the essential features that are required for multi-cloud management platform with containerization</li>
<li>define the multi-cloud strategy and the role of Docker Container Management in implementing multi-cloud containerization</li>
<li>list the critical Docker services that needs to be managed while setting up multi-cloud containerization architecture with Docker</li>
<li>describe the features of Docker Cloud and Docker Hub along with the benefits of combining Docker and multi-cloud computing technologies</li>
<li>demonstrate the steps involved in integrating Docker Cloud with AWS</li>
<li>set up nodes on Docker Cloud and deploy services on the nodes</li>
<li>create multi-cloud Docker clusters using Docker Swarm, Docker Machine, and popular cloud platforms</li>
</ul></td>
<td>Objectives:<ul>
<li>set up overlay network driver to create distributed networks among multiple Docker daemon hosts</li>
<li>define the federated application management process along with the benefits of adopting federated application pattern</li>
<li>recognize the components and features of Docker Enterprise along with the role of Docker Enterprise in implementing federated application management architecture</li>
<li>compare the capabilities of Docker and Kubernetes from the perspective of multi-cloud orchestration and management</li>
<li>describe the Kubernetes federation architecture and recognize the role of Kubernetes and Docker that follows topology and orchestration specification for cloud applications</li>
<li>connect Docker local client with remote AWS Docker host to set up hybrid containerized environment</li>
<li>demonstrate the implementation of Multi-host Overlay Networking using Etcd</li>
<li>recognize the objective of Cloud Native Computing Foundation and describe the CNCF's recommended path using the Cloud Native Landscape</li>
<li>create AWS and Azure clusters using the Containership.io platform</li>
<li>set up federated application management with Docker Enterprise and Amazon EKS</li>
</ul></td>
</tr>
</table>

<table>
<tr>
<td> 

Managing Multi-cloud Containers Using Kubernetes</td>
<td> 

OpenStack in CloudOps: Managing Multi-cloud with OpenStack</td>
</tr>
<tr>
<td>Objectives:<ul>
<li>recall the architecture and components of Kubernetes along with the different types of pods that can be used to host vertically-integrated applications</li>
<li>list the prominent tools and add-ons that are used to manage the deployment and life cycle of Kubernetes clusters and describe their associated features</li>
<li>describe the essential patterns that can be used with Kubernetes to create production-level and heterogeneous deployments</li>
<li>recall the challenges of multi-cloud architectures and illustrate how Kubernetes can help reduce the complexity and risks of multi-cloud strategy</li>
<li>compare the features of Kubernetes and Docker and describe the solution architectures that can be built using Kubernetes and Docker in production</li>
<li>recognize the essential Kubernetes deployment patterns that can be used to deploy application stacks using the Declarative deployment method</li>
<li>implement Fixed and Recreate deployment strategies using Kubernetes</li>
<li>implement Zero Downtime and No Concurrent Versions deployment applying the Blue/Green deployment strategy using Kubernetes</li>
<li>implement Canary testing using Kubernetes Deployments and Services</li>
<li>describe the Kubernetes Cluster Federation, which allows coordinating the configuration of multiple Kubernetes clusters</li>
<li>install and configure multi-cloud single node Kubernetes clusters on Ubuntu to deploy all Kubernetes services</li>
<li>install and configure multi-cloud Kubernetes on AWS to manage multi-cloud orchestration</li>
<li>describe the Extension and Operator patterns that are used in Kubernetes to improve Kubernetes cluster management capabilities</li>
</ul></td>
<td>Objectives:<ul>
<li>recognize the role of OpenStack and the OpenStack components that provide the capability to access infrastructure resources</li>
<li>install OpenStack and demonstrate the steps to create and launch virtual machine instances using the UI</li>
<li>install and configure storage nodes to operate account, container, and object services</li>
<li>describe topologies of multi-cloud orchestration involving Azure, AWS and OpenStack</li>
<li>differentiate between OpenStack and Kubernetes and describe the benefits of using them together</li>
<li>list tools that can be used to manage OpenStack deployment and life cycle</li>
<li>recognize the role of shared services that are used by OpenStack to provide critical services like identification, imaging, indexing, and placement</li>
<li>describe the features of Heat that are used to orchestrate composite cloud applications through OpenStack</li>
<li>install and configure Heat on OpenStack nodes</li>
<li>recall the different approaches of deploying OpenStack and getting started with the most commonly used OpenStack services</li>
</ul></td>
</tr>
</table>

<table>
<tr>
<td> 

OpenStack in CloudOps: Automation</td>
<td> 

Securing CloudOps Deployments: Security Standards for Multi-cloud</td>
</tr>
<tr>
<td>Objectives:<ul>
<li>recognize the enterprise and operational factors that impacts the design of OpenStack Cloud</li>
<li>describe the impact of SLA considerations on the design of OpenStack cloud to provide redundancy and high availability</li>
<li>recall use cases of implementing OpenStack container integration and open source projects that are used to implement OpenStack container integration points</li>
<li>list the prominent containers and Platform as a Service tools that can be deployed on OpenStack Cloud</li>
<li>recognize how Magnum helps manage container orchestration engine provisioning</li>
<li>launch OpenStack instances using Docker Machine and then install and configure Docker on the launched instances</li>
<li>run Kubernetes clusters in OpenStack ecosystem and deploy a simple application server</li>
<li>describe the challenges of implementing OpenStack multi-cloud along with the associated solutions for eliminating those challenges</li>
<li>install and configure Jenkins Automation Server and create jobs to install OpenStack Cloud</li>
<li>install OpenStack-Ansible in test environments using general workflows that include hosts preparation, deployment configuration, playbook execution, and verification</li>
</ul></td>
<td>Objectives:<ul>
<li>recognize the key principles and strategies for securing the enterprise cloud</li>
<li>describe recommended practices and models for perimeter security that can help enterprises develop consistent and effective approaches to cloud security</li>
<li>recall the role and characteristics of a cloud security framework and classify the key cloud audit methods</li>
<li>specify the appropriate choices for security mechanisms and recognize options that need to be configured for security mechanisms on both the service and client side</li>
<li>recognize prominent intrusion detection techniques that are used to manage cloud computing and security challenges</li>
<li>recall approaches that can be adopted to overcome the impacts of security challenges in multi-cloud environments</li>
<li>identify security challenges that need to be considered when using multiple cloud providers and recognize best practices that enterprises can adopt to significantly improve the security of multi-cloud deployments</li>
<li>recognize approaches of securing CloudOps platforms that can help in building secure pipelines, middleware, and infrastructure for multi-cloud applications</li>
<li>recall security issues of public, private, and hybrid cloud models that need to be considered by cloud architects when designing cloud solutions</li>
<li>specify the steps that cloud service customers should take to evaluate and manage the security and privacy of the cloud services from the perspective of mitigating risks and delivering the appropriate level of support</li>
</ul></td>
</tr>
</table>

<table>
<tr>
<td> 

Securing CloudOps Deployments: Implementing</td>
<td> 

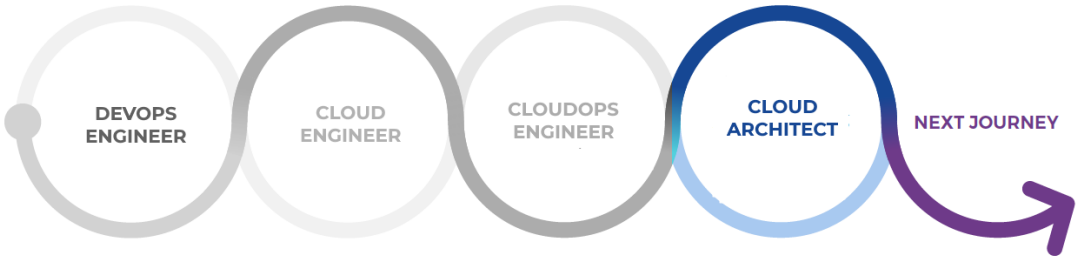
CloudOps: Implementing SD-WAN to Optimize</td>
</tr>
<tr>
<td>Objectives:<ul>
<li>recall AWS Cloud Security services and features that can be used to secure workloads and applications in the cloud</li>
<li>configure AWS Security Hub to automate security checks, manage security findings, and identify the highest priority security issues across AWS environments</li>
<li>create and configure symmetric and asymmetric keys using AWS KMS to control encryption across AWS</li>
<li>list the security products provided by Azure that can be used to protect data, applications, and infrastructure when implementing layered security and defense in depth strategies across identity, data, hosts, and networks</li>
<li>implement Azure Active Directory and Seamless Single Sign-On to automatically sign in users</li>
<li>recognize the Google Security Model and Google services that can be used to fulfil the security, policy, and regulatory compliance requirements of cloud deployments</li>
<li>set up VPN between a Check Point Security Gateway and Cloud VPN on Google Cloud Platform</li>
<li>recognize the need for securing containers and recommended cloud container security best practices to consider when securing container usage in the cloud</li>
<li>harden security and restrict network access to the control plane and nodes of Google Kubernetes Engine</li>
<li>encrypt Kubernetes Secrets at the application layer using keys that are managed in Cloud Key Management Service</li>
</ul></td>
<td>Objectives:<ul>
<li>recall the evolution of wide area networks</li>
<li>differentiate between traditional and software-defined wide area networks (SD-WANs) and describe the advantages of using SD-WAN</li>
<li>describe the solution architecture of an SD-WAN and the primary components that are used to build SD-WAN solutions</li>
<li>differentiate between a software defined network and SD-WAN and describe the working mechanisms of the SD-WAN</li>
<li>describe SD-WAN overlays and the design of various topologies and deployment models using SD-WAN overlays</li>
<li>describe how CloudOps can be applied to networks to remove the complexity of configuring, managing, and delivering SD-WANs</li>
<li>list tools and applications that can be used to build, deploy, and maintain SD-WANs</li>
<li>recognize security challenges and best practices for SD-WAN secure deployments</li>
<li>recall the prominent Cisco SD-WAN architectures that can be implemented on AWS</li>
<li>configure AWS Transit Gateway using the Amazon VPC Console</li>
<li>recognize challenges of legacy network design and SD-WAN approaches for transforming complex legacy networks into easy-to-manage and scalable networks</li>
<li>configure the AWS-provided Cisco Cloud Services Router 1000V - BYOL for SD-WAN</li>
<li>configure VMware SD-WAN using VeloCloud Gateway to facilitate SD-WAN capabilities to service provider environments</li>
<li>list common techniques and solutions for optimizing WAN to speed up important elements of the network</li>
</ul></td>
</tr>
</table>

Final Exam: CloudOps Engineer

Objectives:

- classify the prominent types of cloud integration along with their benefits in creating and operating cloud-to-cloud integration
- compare the differences between DevOps and CloudOps along with challenges and best practices associated with the implementation of CloudOps
- compare the differences between Software Defined Network and Software-Defined Wide Area Network along with the working mechanisms of Software-Defined Wide Area Network
- compare the features of Kubernetes and Docker and describe the solution architectures that can be built using Kubernetes and Docker in production
- configure AWS Security Hub to automate security checks, manage security findings and identify the highest priority security issues across AWS environments
- configure the AWS provided Cisco Cloud Services Router 1000V - BYOL for Software-Defined Wide Area Network
- connect Docker local client with remote AWS Docker host to set up a hybrid containerized environment
- create and configure Symmetric and Asymmetric keys using AWS KMS to control encryption across AWS
- create AWS and Azure clusters using the Containership.io platform
- define IaaS along with the architecture of IaaS and the key benefits that we can realize with the adoption of IaaS with CloudOps
- define the concept of containers along with the benefits of running containers in multi-cloud environments
- define the multi-cloud strategy and the role of Docker Container Management in implementing multi-cloud containerization
- demonstrate the steps involved in configuring AWS Transit Gateway using the Amazon VPC Console
- demonstrate the steps involved in integrating AWS with Azure Active Directory
- describe the features of Heat that are used to orchestrate composite cloud applications through OpenStack
- describe the impact of SLA considerations on the design of OpenStack cloud to provide redundancy and high availability
- describe the interfaces of the Distributed Computing Reference Model along with the standardization required for cloud computing portability and interoperability
- describe the role of standards in cloud computing interoperability and list cloud computing interoperability use cases supported by standards
- encrypt Kubernetes Secrets at the application layer using keys that are managed in Cloud Key Management Service
- identify the categories of cloud automation and orchestration tools from the perspective of cloud provider-native and third-party tools
- identify the characteristics of prominent cloud management platforms that drives successful implementation and adaption of CloudOps
- identify the characteristics of Software-Defined Wide Area Network
- implement container linking to allow multiple containers to link with each other
- implement Zero Downtime and No Concurrent Versions deployment applying the Blue/Green deployment strategy using Kubernetes
- install and configure Jenkins Automation Server and create jobs to install OpenStack Cloud
- install and configure multi-cloud single-node Kubernetes clusters on Ubuntu to deploy all Kubernetes services
- install and configure storage nodes to operate the account, container and object services
- install and configure the CloudHub platform service to deploy applications with a fully-managed and highly available cloud infrastructure
- install OpenNebula and configure the OpenNebula IaaS cloud installation on Ubuntu LTS to create multi-cloud models
- install OpenStack and demonstrate the steps to create and launch virtual machine instances using UI
- list and compare the prominent cloud offerings that enable enterprises to deploy and manage a wide array of private, public and hybrid cloud environments
- list the categories of cloud computing portability and interoperability with a focus on data, application, platform, and infrastructure components

- list the core components of IaaS platform and architecture and compare the capabilities of IaaS with Bare Metal as a Service
- list the prominent containers and Platform as a Service tool that can be deployed on OpenStack Cloud
- list the prominent multi-cloud management tools that we can use to implement CloudOps along with the recommended criteria for selecting the right multi-cloud management tool
- list the prominent tools and add-ons that are used to manage the deployment and life cycle of Kubernetes clusters and describe their associated features
- list the prominent tools that can be used to manage OpenStack deployment and lifecycle
- list the security products provided by Azure that can be used to protect data, applications, and infrastructures when implementing layered and defense-in-depth strategy across identity, data, hosts, and networks
- recall application design principles that can help reduce complexity and provide better interoperability for diversified applications that are designed for cloud computing
- recall the approaches that can be adopted to overcome the impacts of security challenges in Multicloud environments
- recall the architecture and components of Kubernetes along with the different types of pods that can be used to host vertically-integrated applications
- recall the architecture of Docker along with the features afforded by Docker components
- recall the benefits afforded by Docker and list the prominent open-source tools that can be used by architects for productive container implementations
- recall the challenges of multi-cloud architectures and illustrate how Kubernetes can help reduce the complexity and risks of a multi-cloud strategy
- recall the core cloud services and compare the capabilities of IaaS with SaaS and PaaS
- recall the essentials of Cloud Security Framework and classify the prominent Cloud Audit methods
- recall the prominent Cisco Software-Defined Wide Area Network architectures that can be implemented on AWS
- recall the security services provided by AWS Cloud Security to secure workloads and applications in the cloud along with their associated features and capabilities
- recall the various use cases of implementing OpenStack container integration along with the prominent open-source projects that are used to implement OpenStack container integration points
- recognize the approaches of securing CloudOps platform that can help build secure pipelines, middleware, and infrastructures for Multicloud applications
- recognize the challenges of designing CloudOps solution for multi-cloud and describe the optimized design of multi-cloud CloudOps framework to mitigate the challenges from the perspective of release management
- recognize the cloud capabilities that need to be integrated into existing DevOps implementations to transform DevOps to CloudOps
- recognize the components and features of Docker Enterprise along with the role of Docker Enterprise in implementing federated application management architecture
- recognize the critical challenges of setting up multi-cloud container management along with the essential features that are required for multi-cloud management platform with containerization
- recognize the security challenges and the best practices that need to be applied for Software-Defined Wide Area Network secure deployments
- run Kubernetes clusters in the OpenStack ecosystem and deploy a simple application server
- set up overlay network driver to create distributed networks among multiple Docker daemon hosts
- specify the benefits of achieving cloud interoperability and maximizing portability, along with the challenges of achieving interoperability between diversified cloud components
- specify the factors that need to be considered while adopting cloud deployment strategies along with the appropriate scenarios of adopting private and public cloud for application deployment
- specify the steps that Cloud Service Customers should take to evaluate and manage the security and privacy of the cloud services from the perspective of mitigating risks and delivering the appropriate level of support




## Track 4: Cloud Architect (duration: 5h 41m 7s)

|  Role of a Cloud Architect |  Cloud Transition: Adopting & Moving to Cloud & Multi-cloud Environments |
|---|---|
| Objectives:<br>▪ define the roles and responsibilities of a Cloud Architect in architecting cloud practices to manage multi-cloud service deployment and management<br>▪ describe the cloud strategies and architecture services used by cloud architects to deliver multi-cloud operations management<br>▪ differentiate between Solution Architects and Cloud Architects and identify the deliverables they need to deliver to deploy and manage multi-tier applications on multi-cloud environments<br>▪ recognize cloud challenges that need to be managed by Cloud Architects to ensure Agile delivery mechanisms and optimization of IT services and workloads that can run in multi-cloud environments<br>▪ classify the solutions that need to be planned by Cloud Architects to effectively implement security, traffic management, automation, and orchestration of multi-cloud applications<br>▪ describe the Cloud Computing Reference Model and Cloud Governance Lifecycle that need to be implemented by Cloud Architects to build cloud frameworks<br>▪ recognize the key success factors that need to be considered as benchmarks by Cloud Architects to identify business values of suggested cloud architectures<br>▪ describe the team structure and service delivery model that helps Cloud Architects address key challenges and enumerate multi-cloud delivery model<br>▪ specify standards and compliance that a Cloud Architect needs to assess to fulfil the compliance requirements that are to be applied on multi-cloud practices<br>▪ list tools and frameworks used by Cloud Architects to assess cloud operations and identify current capabilities of multi-cloud<br>▪ describe the Target Operating Model that needs to be proposed by Cloud Architects to derive vision to deliver value through appropriate cloud strategies<br>▪ describe the Multi-cloud Maturity Model and transformation workflows that need to be adopted by Cloud Architects to build solutions for cloud delivery mechanisms | Objectives:<br>▪ recall the mindset differences between DevOps and CloudOps and describe what CloudOps offers to optimize speed, security, and operational efficiency of DevOps teams<br>▪ describe the roadmap that can be adopted to transition from DevOps to CloudOps, along with the role played by cloud architects in facilitating the transition<br>▪ identify the role of the Cloud Operating Model in helping cloud architects and target audiences define operational processes that align business and IT to accomplish strategic business outcomes<br>▪ recall the role of the people, organization, and operations domain in determining the transition process of building a cloud-enabled enterprise through the application of appropriate cloud enablement and a well-designed CloudOps process model<br>▪ list the best practices that can be adopted by cloud architects to transform enterprise capabilities and describe the benefits of multi-cloud transition<br>▪ describe approaches cloud architects can use to manage data and information in order to implement the right cloud transition mechanisms<br>▪ list common cloud transition strategies that can be used to build a transition roadmap, with focus on migrated critical applications and infrastructures<br>▪ identify the critical systemic principles that cloud architects need to adopt when planning multi-cloud transition, with focus on disaster recovery and SLAs<br>▪ describe features of CloudOps solution planning and deployment tools, as well as usage scenarios from the perspective of audience and role<br>▪ recognize the processes, SLAs, and KPIs associated with running cloud operations that can help create operational baseline capabilities to be used as benchmarks in transition<br>▪ list the pivotal organizational roles that help to facilitate smooth process, data, and platform transition in multi-cloud environments<br>▪ describe the maturity model of CloudOps and recognize the transition path that needs to be used to identify the right maturity model to eliminate negative impacts on business and users |

<table>
<tr>
<td> 

CloudOps Solutioning Strategies</td>
<td> 

Applying the Explainability Approach to Guide Cloud Implementation</td>
</tr>
<tr>
<td>Objectives:<ul>
<li>list the components that play key roles in solutioning CloudOps projects</li>
<li>recognize the role of a layered approach in managing enterprise CloudOps applications and describe multiple viewpoint elements from the perspective of users, operators, engineers, and managers</li>
<li>describe the role-based CloudOps solutioning architecture reference models that are guided by DMTF, IBM, and NIST</li>
<li>describe the layer-based CloudOps solutioning architecture reference models that are guided by Cisco, IEFT, and ITU</li>
<li>recognize the different types of cloud engineers and their roles in defining vision and providing logical and physical architecture during the CloudOps solutioning process</li>
<li>recognize the evolutionary steps that are used to define successful CloudOps adoption and to ensure processes have close alignment with business goals and strategies</li>
<li>recognize the role of a cloud governance framework in addressing gaps and deficiencies and in providing comprehensive cloud-centric governance for each phase of the cloud transformation life cycle used in CloudOps</li>
<li>recognize how the Responsible, Accountable, Consulted, and Informed matrix defines the roles played by all stakeholders engaged in solutioning CloudOps</li>
<li>list the measures and metrics of each step involved in solutioning CloudOps strategies for enterprises</li>
<li>describe the anatomy of Cloud Service Agreements that must be considered by CloudOps architects when defining an SLA or Fair Use Policy for cloud services</li>
<li>recognize prominent CloudOps solutioning tools that can be used to design, plan, architect, implement, and estimate costs to ensure CloudOps solutioning provides the expected ROI</li>
<li>describe the monitoring, metering, and billing components of CloudOps architecture and recognize factors that can help derive cloud services pricing models for building CloudOps solutions</li>
</ul></td>
<td>Objectives:<ul>
<li>describe the concept of AI Explainability and differentiate between AI Explainability and CloudOps Explainability</li>
<li>describe CloudOps Explainability and the role it plays in CloudOps implementation for managing multi-cloud solutions</li>
<li>define explanatory systems and evaluate them from functional, operational, usability, security, and validation perspectives</li>
<li>identify properties that are used to define systems to accommodate explainability approaches and recognize how users interact with explainable systems and what is expected of them</li>
<li>recall the effect of explainability on the robustness, security, and privacy aspects of predictive systems and describe approaches of evaluating how well the explanation is understood using qualitative and quantitative validation approaches</li>
<li>list explainability techniques that can be used to define operational and functional derivatives of CloudOps including leave one column out, permutation impact, and local interpretable model-agnostic explanations</li>
<li>recognize the role of explainability and how it can be applied throughout the process of operating cloud environments and infrastructures to ensure efficient service delivery following the CloudOps paradigm</li>
<li>describe the three stages of AI Explainability along with the methodologies that are used in each stage to derive the right CloudOps model for implementation guidance</li>
<li>recognize the role of explainability in defining AI-assisted Cloud Managed Services that can be used to manage large cloud enterprise distributed applications</li>
<li>list the architectures that can be derived using Explainable Models and that can help share CloudOps or DevOps Model Explainability with the stakeholders to establish better collaboration</li>
<li>recognize the role of Explainable AI reasoning paths in building CloudOps workflows that can be trusted by customers, employees, regulators, and other key stakeholders</li>
<li>describe the role of CloudOps and DevOps Explainability in mitigating challenges along with the need for management and governance of AI frameworks in CloudOps architectures</li>
</ul></td>
</tr>
</table>


Cloud Future: Adapting Cloud Innovation

Objectives:

- recognize the future trends in cloud computing that a CloudOps architect should be aware of in order to conduct impact analysis on existing CloudOps practices
- recognize emerging technologies and their associated impact on CloudOps
- recall solutioning strategies that can be adopted by a CloudOps architect to accommodate the changes that are driven by emerging technologies in existing CloudOps architectures
- recognize the key resource types that can help a CloudOps architect keep track of the future CloudOps developments and adopt the right trending features to improve existing CloudOps architectures
- recognize critical cloud application design considerations that a CloudOps architect needs to adhere to in order to ensure they can adopt and adapt to any futuristic upgrades to the existing practices
- describe the inside-out development strategy that can help CloudOps teams focus on implementing domain models before defining how they are to be used by the external world to ensure they can adapt to futuristic interfaces
- describe the outside-in development strategy that can help CloudOps teams focus on solutioning and reduce accidental complexity by removing speculative work
- recognize the role of states as applied in scalable multi-cloud architectures and how to handle states at the service level in order to move into futuristic systems integration
- describe the system architecture design considerations that a CloudOps team must apply to be able to build architectures that can adopt and adapt to futuristic innovations with the use of the Adapter pattern
- recognize the Minimum Valuable Increment design approach used by CloudOps teams to drive Agile processes and ensure business and IT alignment
- list research firms that work on forecasting the future of cloud computing and their impact on the future of the enterprise
- describe the Manifesto for Future Generation Cloud Computing and the areas that a CloudOps architect should be familiar with to align with future research directions

Final Exam: Cloud Architect

Objectives:

- categorize the different types of Cloud Engineers along with their roles in defining the vision and providing logical and physical architecture during CloudOps Solutioning process
- classify the solutions that need to be planned by Cloud Architects to effectively implement security, traffic management, automation and orchestration of Multicloud applications
- compare the differences between the roles played by Solution Architects and Cloud Architects and identify the deliverables that need to be delivered by them to deploy and manage Multi-tier applications on Multicloud environments
- define the concept of AI Explainability and describe the analogy between AI Explainability and CloudOps Explainability
- define the concept of Explanatory System and evaluate the Explanatory Systems from the perspective of various dimensions that includes Functional, Operational, Usability, Security and Validation
- define the maturity model of CloudOps and recognize the transition path that needs to be used to identify and apply the right maturity model to eliminate negative impacts on business and users
- define the processes, SLAs and KPIs associated with running Cloud operations that can help create Operational Baseline capabilities to be used as benchmarks in transition
- define the role of Cloud Governance Framework in addressing gaps, deficiencies and providing comprehensive Cloud-centric governance for each phase of the cloud transformation lifecycle used in CloudOps
- define the roles and responsibilities of a Cloud Architect in architecting CloudOps Practices to manage Multicloud service deployment and management

- describe the anatomy of Cloud Service Agreement that must be factored by Cloud Architects to define SLAs and Fair Use Policy of Cloud services
- describe the approaches that need to be adopted by Cloud Architects to manage data and information to enumerate the right Cloud transition mechanisms
- describe the Cloud Computing Reference Model and Cloud Governance Lifecycle that need to be implemented by Cloud Architects to build CloudOps frameworks
- describe the CloudOps Explainability and specify the role it plays in CloudOps implementation to manage Multicloud solutions
- describe the Cloud Strategy & Architecture Services enumerated by Cloud Architect to deliver Multicloud operations management
- describe the features of the prominent tools that can be used for planning and deploying CloudOps Solutions along with the usage scenarios from the perspective of audience and role
- describe the Inside-out Development strategy that can help CloudOps teams focus on implementing Domain Model before defining how they are to be used by the external world to ensure they can adapt to futuristic interfaces
- describe the Layer-based CloudOps Solutioning architecture reference models that are guided by CISCO, IEFT, and ITU
- describe the logical CloudOps Architecture with the Monitoring, Metering, and Billing components and list the essential factors that can help derive pricing model of the Cloud services that are used to build CloudOps solutions
- describe the Multicloud Maturity Model and Transformation Workflows that need to be adopted by a Cloud Architect to build a solution for CloudOps delivery mechanism
- describe the Outside-in Development strategy that can help the CloudOps team focus on Solutioning and reduce accidental complexity by removing speculative work
- describe the roadmap that can be adapted to transition from DevOps to CloudOps along with the role played by Cloud Architects in facilitating the transition
- describe the Role-based CloudOps Solutioning architecture reference models that are guided by DMTF, IBM and NIST
- describe the role of CloudOps and DevOps Explainability in mitigating challenges along with the need for management and governance of AI Frameworks in CloudOps architectures
- describe the System Architecture Design considerations that a CloudOps team must adapt to be able to build architectures that adopt and adapt to futuristic innovations with the use of the Adapter pattern
- describe the Target Operating Model that needs to be proposed by a Cloud Architect to derive vision to deliver value through appropriate cloud strategies
- describe the team structure and Service delivery Model that helps Cloud Architect to address key challenges and enumerate Multicloud delivery model
- describe the three stages of AI Explainability along with the methodologies that are used in each stage to derive the right CloudOps model for implementation guidance
- identify the critical systemic principles that Cloud Architects need to adopt while planning Multi-cloud transition with a focus on Disaster Recovery and SLAs
- identify the Manifesto for Future Generation Cloud along with the areas that are relevant in the Cloud horizon that a Cloud Architect need to be prepared to be able to align with the future research directions
- identify the properties that are used to define systems to accommodate Explainability approaches and illustrate how users interact with Explainable systems and what is expected of them
- identify the role of Cloud Operating Model that helps Cloud Architects and target audiences define operational processes that align Business and IT to accomplish strategic business outcomes
- list the architectures that can be derived using Explainable Models, and that can help share CloudOps or DevOps Model Explainability with the stakeholders to establish better collaboration
- list the architectures that can be derived using Explainable Models, and that can help share CloudOps or DevOps Model Explainability with the stakeholders to establish better collaboration
- list the best practices that can be adopted by Cloud Architects to transform enterprise capabilities along with the benefits of Multi-cloud transition
- list the common and prominent Cloud transition strategies that can be used to build transition roadmap with a focus on migrated critical applications and infrastructures
- list the critical roles that are pivotal for any organization to facilitate the smooth process, data and platform transition in Multi-cloud environments
- list the essential components that play key roles in solutioning CloudOps projects

- list the Explainability techniques that can be used to define operational and functional derivatives of CloudOps including Leave One Column Out, Permutation Impact, and Local Interpretable Model-agnostic Explanations
- list the prominent emerging technologies along with their associated impact on CloudOps
- list the prominent tools and frameworks used by a Cloud Architect to assess cloud operations and identify current capabilities of Multicloud
- recall the critical Cloud Application Design considerations that a Cloud Architect need to adhere to to ensure they can adopt and adapt to any futuristic upgrades to the existing practices
- recall the effect of Explainability on the robustness, security and privacy aspects of Predictive Systems and describe the approaches of evaluating how well Explanation is understood using Qualitative and Quantitative validation approaches
- recall the Manifesto for Future Generation Cloud along with the areas that are relevant in the Cloud horizon that a Cloud Architect need to be prepared to be able to align with the future research directions
- recall the mindset difference between DevOps and CloudOps and describe what CloudOps offers to optimize speed, security and operational efficiency of DevOps teams
- recall the Minimum Valuable Increment design approach suggested CloudOps teams drive agile processes and ensure business and IT alignment
- recall the prominent CloudOps Solutioning tools that can be used to design, plan, architect, implement and estimate costs to ensure CloudOps solutioning provides expected ROI
- recall the role of the People, Organization and Operations domain in determining the transition process that can help build Cloud-enabled enterprise with the application of appropriate cloud enablement using well designed CloudOps process model
- recall the Solutioning Strategies that can be adopted by a Cloud Architect to accommodate the changes that are driven by emerging technologies in the existing CloudOps architectures
- recognize the architectures that can be derived using Explainable Models, and that can help share CloudOps or DevOps Model Explainability with the stakeholders to establish better collaboration
- recognize the challenges of CloudOps that need to be managed by Cloud Architects to ensure Agile Delivery Mechanism and optimization of IT services and workloads that can run in Multicloud environments
- recognize the Evolutionary steps that need to be used to define successful CloudOps adoption to ensure processes have a close alignment with business goals and strategies
- recognize the future trends in Cloud Computing that a Cloud Architect should be aware of to be able to conduct impact analysis on existing CloudOps practices
- recognize the key Resource Types that can help a Cloud Architect keep track of the future CloudOps developments and adopt the right trending feature to improve existing CloudOps architectures
- recognize the key success factors that need to be considered as a benchmark by Cloud Architect to identify business values of suggested CloudOps architectures
- recognize the role of Explainability that can be applied throughout the process of operating Cloud environments and infrastructures to ensure efficient service delivery following the CloudOps paradigm
- recognize the role of Layered approach in managing Enterprise CloudOps applications and specify the Multiple Viewpoint elements from the perspective of users, operators, engineers, and managers
- recognize the role of states that are applied in scalable architectures of Multi-cloud and that provides approaches of handing states at service level and move into futuristic systems integration
- recognize the stages of building CloudOps
- specify the Responsible, Accountable, Consulted and Informed Matrix of Roles played by all the stakeholders engaged in solutioning CloudOps
- specify the standards and compliances that a Cloud Architect need to assess to fulfill the compliance requirements that are to be applied on Multicloud practices

## Productivity Tools for CloudOps Architects

Optional

COURSE

**Signing in & Setting up a Team**

11

COURSE

**Using the Conversation Tools**

9

COURSE

**Creating & Managing Projects**

11

COURSE

**Finding & Sharing Items**

6

COURSE

**Running Reports & Configuring Projects**

5

COURSE

**Signing In & Setting Up**

18

COURSE

**Using the Team Communication Tools**

49

COURSE

**Setting Up & Tracking Projects**

14

COURSE

**Managing your Project Tasks & Assets**

12

COURSE

**Using the Calendar Tools**

11

COURSE

**Using Basecamp for iOS**

12

COURSE

**Sign-in & Setup**

6

COURSE

**Communication Tools**

10

COURSE

**Working with Groups**

5

COURSE

**Creating, Finding, & Sharing Information**

6

COURSE

**Configuring Convo**

4

COURSE

**The Convo iOS App**

7

COURSE

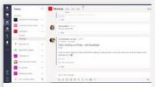
**Microsoft Teams: Getting to know the application**

784

COURSE

**Microsoft Teams: Using Teams & Channels**

614



COURSE

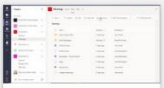
**Microsoft Teams: Communicating via the App**

573

COURSE

**Microsoft Teams: Formatting, Illustrating & Reacting to...**

457



COURSE

**Microsoft Teams: Creating, Finding & Organizing Files**

430

COURSE

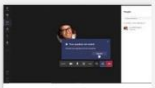
**Microsoft Teams: Working with Apps, Tabs & Wiki**

401

COURSE

**Microsoft Teams: Making calls, Organizing Contacts ...**

395



COURSE

**Microsoft Teams: Creating, Joining & Managing Meetings**

389


COURSE

**Signing in & Setting Up**

35

COURSE

**Using Channels**

24

COURSE

**Private Messaging & Communication Tools**

26


COURSE

**Creating, Finding & Sharing Information**

11

COURSE

**Configuring Slack**

32

COURSE

**Using the iOS App**

8

COURSE

**Sign-in & Setup**

33

COURSE

**Creating Teams & Boards**

17

COURSE

**Managing Cards**

11

COURSE

**Finding & Sharing Information**

12

COURSE

**Setting Up**

43

COURSE

**Posting & Reacting to Status Updates**

33

COURSE

**Using Groups**

10

COURSE

**Collaborating & Communicating**

58

COURSE

**Configuring Networks**

21



COURSE

**JIRA Cloud: Creating & Setting Up Projects**

71



COURSE

**JIRA Cloud: Configuring & Managing Boards**

48



COURSE

**JIRA Cloud: Planning & Working on a Software...**

40



COURSE

**JIRA Cloud: Reporting in Jira Software**

38

# Bookshelf

Optional

BOOK

**Leading the Transformation: Applying Agile and DevOps...**

47

AUDIOBOOK

**Leading the Transformation: Applying Agile and DevOps...**

9

BOOK

**Kanban Change Leadership: Creating a Culture of...**

8

BOOK

**The Kanban Playbook**

29

BOOK

**Business-Driven IT-Wide Agile (Scrum) and Kanban...**

21

BOOK

**Hyper-Productive Knowledge Work...**

5

BOOK

**DevOps for Azure Applications: Deploy Web...**

5

BOOK

**Pro DevOps with Google Cloud Platform: With...**

13

BOOK

**DevOps for Dummies**

28

BOOK

**Pro Powershell for Amazon Web Services: DevOps for...**

100

BOOK

**Agile, DevOps and Cloud Computing with Microsoft...**

8

BOOK

**Agile Project Management with Azure DevOps:...**

8

BOOK

**Hands-on Azure Repos: Understanding Centralized...**

1

BOOK

**Cloud Computing**

21

BOOK

**Cryptography for Security and Privacy in Cloud...**

11

BOOK

**Pro Docker**

64

BOOK

**OpenStack Cloud Application Development**

5

BOOK

**OpenStack in Action**

3

BOOK

**Applied OpenStack Design Patterns: Design Solutions...**

BOOK

**Developing Interoperable and Federated Cloud...**

7

BOOK

**Migrating Large-Scale Services to the Cloud: A...**

6

BOOK

**Cloud Computing: A Self-Teaching Introduction**

49

BOOK

**The Basics of Cloud Computing: Understanding...**

70

## Business & Leadership for CloudOps Architects

Optional

COURSE

**Thinking Strategically as a Manager**

381

COURSE

**Improving Your Technical Writing Skills**

303

COURSE

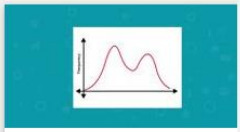
**Agile Principles and Methodologies**

1424

COURSE

**Taking Your Team to the Next Level with Delegation**

151



COURSE

**Data Analysis and Root Cause Analysis in Six Sigma**

168


COURSE

**Agile Project Planning**

543


COURSE

**Agile Project Scheduling and Monitoring**

367

## FOLLOW US ON:

**www.skilltech.pl**

**email: biuro@skilltech.pl**

**tel. +48 22 44 88 827**

SkillTech

Technology hired for excellence