



**ENTERPRISE DEVELOPER TO  
DEVOPS ENGINEER**

**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.

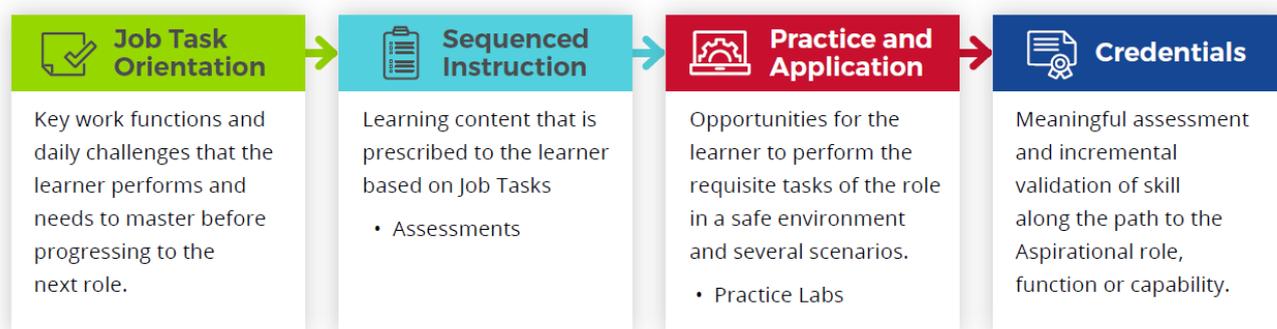
**Skillsoft Aspire Journey** stanowi odpowiedź na pytanie, jakie szkolenia muszą ukończyć, aby być przygotowanym do swojej wymarzonej pracy. Spośród kilkuset kanałów tematycznych dostępnych na naszej platformie szkoleniowej nasi specjaliści wybrali te, które naszym zdaniem najlepiej wyposażą uczących 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 poświadczane 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 – Enterprise Developer to DevOps Engineer

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

### DEVOPS JOURNEY

### ENTERPRISE DEVELOPER TO DEVOPS ENGINEER



9 courses  
9h 41m 2s

- DevOps mindset,
- DevOps vocabulary,
- DevOps Agile Development,
- failing fast in DevOps,
- collaboration tools for DevOps.



15 courses  
16h 37m 16s

- choosing DevOps tools
- Git, Docker, Ansible, Chef, Puppet, Jenkins, and SaltStack with DevOps.



12 courses  
13h 0m 49s

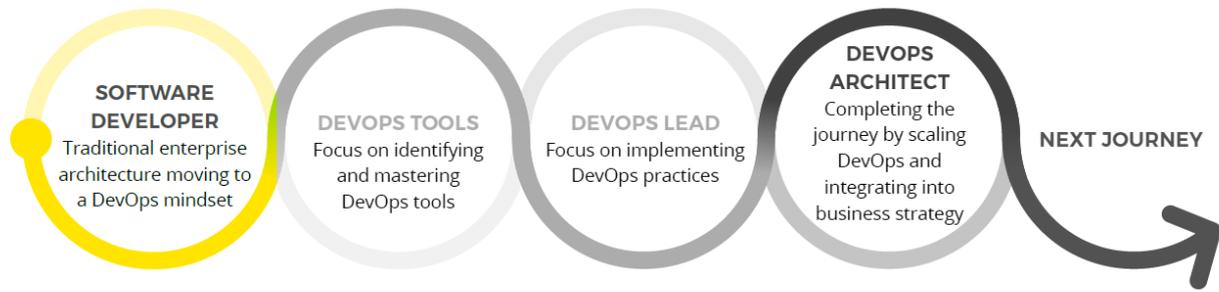
- design best practices for DevOps,
- cloud automation, DevOps automation implementation,
- using Docker, implementing CI/CD for DevOps,
- and continuous testing.



7 courses  
8h 42m 40s

- scaling DevOps for the enterprise,
- using Docker for the DevOps enterprise,
- continuous DevOps feedback,
- establishing DevOps success metrics,
- security considerations for DevOps

**ENTERPRISE DEVELOPER TO DEVOPS ENGINEER**



**Track 1: Track 1: Enterprise Developer (duration: 9h 41m 2s)**

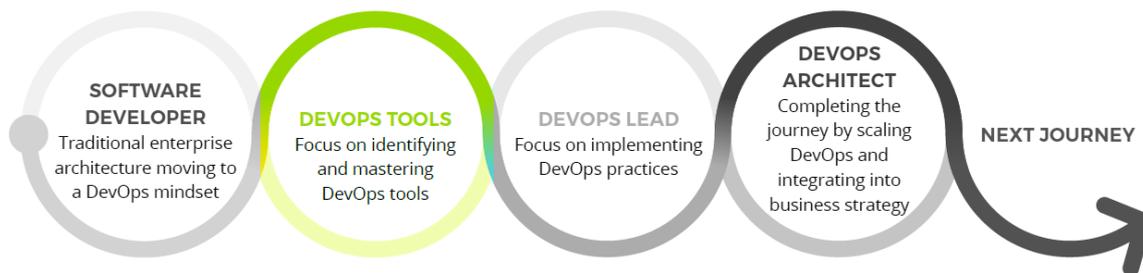
 <p><b>Niranjan Pandey</b> Software Engineer and Big Data Expert</p>	<p>DevOps Mindset: DevOps Principles &amp; Implementation</p>	 <p><b>Niranjan Pandey</b> Software Engineer and Big Data Expert</p>	<p>DevOps Mindset: DevOps Implementation Using Tools</p>
<p>Objectives:</p> <ul style="list-style-type: none"> <li>recognize the problems associated with traditional project management methodologies and approaches that can be managed by adopting the DevOps culture</li> <li>define DevOps and its principles, including the principles of flow, feedback, and continual experimentation</li> <li>describe the technical principles that are adopted with the implementation of the DevOps culture</li> <li>recognize the role of people, process, and technologies in the DevOps adoption process</li> <li>describe the benefits associated with the adoption of integrated approaches of DevOps, Agile, and ISTM</li> <li>recall the benefits of adopting DevOps in the cloud and recognize the DevOps tools provided by AWS and Azure</li> <li>recognize the key responsibilities of developers and operations in ensuring flawless DevOps implementation</li> <li>list the tools that are frequently used to facilitate collaboration and encourage team-oriented process management</li> <li>name the primary tools that a developer can use to implement continuous code build and adopt a team-driven development approach</li> <li>specify the objectives of continuous integration and continuous deployment and recognize the tools that can be used to implement them to achieve automation</li> <li>recall the benefits associated with the implementation of continuous monitoring and reporting</li> <li>demonstrate the approach of implementing AWS DevOps delivery pipeline</li> <li>recall the DevOps tools that are provided by AWS and Azure and implement an AWS DevOps delivery pipeline</li> </ul>		<p>Objectives:</p> <ul style="list-style-type: none"> <li>recognize the benefits of adopting a container-based deployment approach as compared to a VM-based approach</li> <li>demonstrate the essential tasks involved in setting up and managing code repository using Git</li> <li>recall the essential features afforded by Docker for container management and prominent Docker components</li> <li>use Maven to implement continuous code build</li> <li>implement code coverage and automated testing using SonarQube and prominent testing tools</li> <li>implement DevOps practices in the cloud using CodeCommit and CodeDeploy</li> <li>recognize the configuration management and orchestration capabilities provided by Puppet and Chef</li> <li>demonstrate how to configure Jenkins and use it to build pipelines</li> <li>work with Jenkins to implement DevOps pipeline for continuous integration, deployment, and monitoring</li> <li>recall the benefits provided by a container-based deployment approach, configure Jenkins for building pipelines, and implement DevOps pipeline using Jenkins</li> </ul>	

	<p>The Language of DevOps: DevOps Principles &amp; Practices</p>		<p>The Language of DevOps: DevOps Tools &amp; Processes</p>
<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ define the key principles of DevOps</li> <li>▪ recognize the key DevOps terms, which serve as a foundation to adopting the DevOps culture</li> <li>▪ specify the critical benefits afforded by the adoption of the DevOps principles</li> <li>▪ describe the lifecycle of DevOps from the development and operational perspectives</li> <li>▪ list tools that can be used to manage software releases, from the development stage to the actual software release</li> <li>▪ recognize the essential project management practices that are adopted to implement DevOps principles</li> <li>▪ describe the different phases involved in the DevOps process and the tools that can be used to facilitate DevOps processes and principles</li> <li>▪ describe DevOps assessment and establish the DevOps maturity model</li> <li>▪ identify the reference architectures that provide templates of proven solutions, using a set of preferred methods and capabilities</li> <li>▪ identify tools that can be used to build and adopt an end-to-end DevOps culture and practices in the enterprise</li> <li>▪ recall the key principles of DevOps, list the benefits associated with the adoption of DevOps principles, and specify the tools that can be used to adopt DevOps in the enterprise</li> </ul>		<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ recognize the essential terminology and DevOps adoption techniques</li> <li>▪ describe the roles of prominent automation tools that can be used to design, deploy, and reuse environments in establishing DevOps culture</li> <li>▪ set up DevOps tools to facilitate the practices of DevOps culture</li> <li>▪ recall the DevOps lifecycle phases from the perspective of continuous integration, continuous development, and process implementation</li> <li>▪ set up different phases of DevOps using Atlassian Jira and Confluence while specifying the role of collaborative development and operations</li> <li>▪ list the different stages of a typical DevOps delivery pipeline, from development to production</li> <li>▪ recall the end-to-end development and testing process of DevOps that are adapted for the cloud</li> <li>▪ define DevSecOps and illustrate a typical DevOps and DevSecOps workflow</li> <li>▪ list the various DevOps lifecycle phases and recall the different stages of the DevOps delivery pipeline</li> </ul>	

 <p><b>Niranjan Pandey</b> Software Engineer and Big Data Expert</p>	<p><b>DevOps Agile Development: Agile Processes for DevOps</b></p>	 <p><b>Niranjan Pandey</b> Software Engineer and Big Data Expert</p>	<p><b>DevOps Agile Development: DevOps Methodologies for Developers</b></p>
<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ describe the DevOps principles of flow, feedback, and continuous learning and how they are used to derive DevOps behavior</li> <li>▪ recognize the need for DevOps cultural transformation for productive team collaboration</li> <li>▪ recall the design thinking approaches that can be applied to compliment DevOps design practices</li> <li>▪ describe the Lean software development process and recognize the use of Kanban and Scrum Agile implementation practices</li> <li>▪ recognize the process of initiating DevOps transformation with focus on Greenfield and Brownfield DevOps projects</li> <li>▪ describe the approaches of Value Stream mapping, building a DevOps transformation team, and integrating Ops into Dev</li> <li>▪ recognize the steps involved in the DevOps and Agile DevOps implementation processes with the roadmap for DevOps developers</li> <li>▪ describe the approach of designing DevOps strategies that enumerate plans from transformation to implementation</li> <li>▪ build Kanban workflow to illustrate the entire process of transformation to the DevOps culture</li> <li>▪ define the DevOps principle of modular design, and specify the principles, characteristics, and benefits of microservices</li> <li>▪ list design thinking approaches that can compliment DevOps design practices and recall the principles, characteristics, and benefits of microservices</li> </ul>		<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ list the steps involved in implementing continuous integration workflow and the risks that can be mitigated using continuous integration</li> <li>▪ demonstrate how to version and control source codes using Git</li> <li>▪ specify the best practices and patterns for implementing continuous integration</li> <li>▪ demonstrate how to implement continuous build using Maven and MSBuild</li> <li>▪ recall the best practices for implementing continuous testing along with the importance of continuous testing in DevOps</li> <li>▪ demonstrate how to implement automated testing from the perspective of functional and load testing</li> <li>▪ describe the process of implementing continuous deployment with focus on deployment strategies like Blue/Green and Rolling Upgrade</li> <li>▪ set up end-to-end continuous delivery pipelines and implementations using open source DevOps tools</li> <li>▪ recognize the benefits of implementing continuous monitoring in DevOps pipelines</li> <li>▪ list the essential frameworks and tools that can be used to implement infrastructure as code</li> <li>▪ implement infrastructure as code using Puppet to automate infrastructure deployment and configuration management</li> <li>▪ recall the steps involved in implementing continuous integration workflow, list the prominent frameworks and tools that can be used to implement infrastructure as code, and implement infrastructure as code using Puppet</li> </ul>	

 <p><b>Niranjan Pandey</b> Software Engineer and Big Data Expert</p>	<p><b>DevOps Smart Failure: Fail Fast &amp; DevOps</b></p>	 <p><b>April Sidorski</b> IT Trainer/Consultant</p>	<p><b>DevOps Collaboration Tools: Tools for Continuous Integration</b></p>
<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ describe adoption approaches for failure prevention and the fail fast approach</li> <li>▪ describe the circumstances and scenarios leading to the adoption of fail fast systems</li> <li>▪ describe fail fast as a DevOps principle and illustrate the role of feedback cycle</li> <li>▪ list the essential principles driving the Agile manifesto and describe Agile fail fast</li> <li>▪ recognize software development techniques that can be used to build software to fail fast</li> <li>▪ recognize the implementation of the fail fast principle from the perspective of test-driven development and continuous integration</li> <li>▪ set up the application development and operations environment to adopt the fail fast approach</li> <li>▪ demonstrate the steps involved in static code analysis and generating test coverage report to aid implementations of fail fast and continuous improvement</li> <li>▪ demonstrate use cases of implementing Jenkins to enforce fail fast in Maven</li> <li>▪ set up Jenkins pipelines for fail fast management</li> <li>▪ recall DevOps testing strategies that can be adopted to benefit from the benefits of the fail fast design methodology and enable continuous improvement</li> <li>▪ recall the scenarios and circumstances leading to the adoption of the fail fast approach, specify software development techniques that can be used to build software to fail fast, and set up Jenkins pipeline for fail fast management</li> </ul>		<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ recall the prominent tools that are widely used to implement DevOps pipelines</li> <li>▪ set up Jira Confluence and integrate Slack and Bitbucket for team collaboration</li> <li>▪ recognize the need for source code management and list the essential features afforded by popular source code management tools</li> <li>▪ set up the Bitbucket environment for collaboration and code management</li> <li>▪ work with JFrog Artifactory to facilitate artifact management</li> <li>▪ list and compare the prominent continuous integration tools that can be used to build integration pipelines</li> <li>▪ install and set up Jenkins to build DevOps pipelines</li> <li>▪ recall the prominent tools that can be used to automate testing, along with their associated features</li> <li>▪ integrate build and testing tools in DevOps pipeline</li> <li>▪ recall the prominent configuration management tools, along with their associated features</li> <li>▪ set up Puppet environments for configuration management</li> <li>▪ recall source code management tools, list configuration management tools, and set up a Puppet environment for configuration management</li> </ul>	

 <p><b>April Sidorski</b> IT Trainer/Consultant</p>	<p><b>DevOps Collaboration Tools: Tools for Continuous Delivery</b></p>		<p><b>Final Exam: Enterprise Developer</b></p>
<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ list the prominent tools that can be used to configure automated deployment, along with their associated features</li> <li>▪ recognize the prominent tools that can be used to set up the container environment</li> <li>▪ automate release management using Puppet pipelines</li> <li>▪ list the prominent tools that can be used to monitor DevOps processes, along with their associated features</li> <li>▪ demonstrate the steps involved in implementing dashboards using Kibana</li> <li>▪ demonstrate the implementation of end-to-end continuous delivery processes</li> <li>▪ recognize the tool selection criteria that can help select the right tools to build the DevOps implementation stack</li> <li>▪ list tools that can be used to configure automated deployment, monitor DevOps processes, and implement end-to-end continuous delivery</li> </ul>		<p><b>Objectives</b></p> <p>This final exam will test your understanding and application of the skills covered in the Enterprise Developer track of the Skillsoft Aspire Enterprise Developer to DevOps Engineer Journey.</p>	



**Track 2: Track 2: DevOps Developer (duration: 16h 37m 16s)**

 <p><b>April Sidorski</b> IT Trainer/Consultant</p>	<p><b>DevOps Tools: Selecting the Right Tools</b></p>	 <p><b>Niranjn Pandey</b> Software Engineer and Big Data Expert</p>	<p><b>Using Git for DevOps: Using Git Effectively</b></p>
<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>describe DevOps strategies and practices that can be applied to streamline development and operations in enterprises</li> <li>describe DevOps processes and specify steps for selecting the right DevOps tools</li> <li>specify rules that should be considered when selecting DevOps tools for different phases of DevOps</li> <li>describe the impact of selecting the wrong tools in the DevOps lifecycle</li> <li>list and compare the popular source control and versioning tools</li> <li>describe the process of continuous integration and the advantages and disadvantages of continuous integration tools</li> <li>install Bamboo and Jenkins to illustrate their comparative capabilities</li> <li>describe the concept and benefits of test automation and compare prominent test automation tools</li> <li>install and demonstrate the capabilities of JMeter and API testing tools</li> <li>compare the capabilities of provisioning and change management tools, with a focus on configuration management and infrastructure as code implementation</li> <li>describe container management and compare various containerization tools that can be used to implement continuous deployment and delivery</li> <li>describe release orchestration and compare various tools that can be used to implement it</li> <li>compare logging and continuous monitoring tools and describe how they are used to implement feedback loops in the DevOps process</li> <li>install and demonstrate the capabilities of Nagios as a comprehensive continuous monitoring tool</li> <li>describe the impacts of selecting the wrong tools to implement DevOps, list continuous integration tools, list container management tools that can be used to implement continuous delivery, and describe test automation tools that can be used to automate test executions</li> </ul>		<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>describe the role of Git in the DevOps lifecycle and list the products and frameworks that can help you use Git effectively</li> <li>recall the essential terminologies that are used in Git for version control and source code management</li> <li>set up Git and use Git commands to create repositories, track files, and commit files</li> <li>set up Git using Git GUI and create repositories, track files, and commit files</li> <li>work with and manage Git repositories, including repository initialization, cloning, and recording changes to the repository</li> <li>describe how to use and work with Git server and list the essential protocols that can be used to transfer data</li> <li>recognize the features of Distributed Workflows and recall the various types of Distributed Workflows that can be used to implement a flexible approach of project collaboration</li> <li>describe the concept of Branching and its benefits</li> <li>work with Git commands to manage branches with a focus on creating, switching, and merging branches</li> <li>recall the frameworks that can be used with Git, create a Git repository, and commit files to the repository</li> </ul>	

 <p><b>Niranjan Pandey</b> Software Engineer and Big Data Expert</p>	<h3>Using Git for DevOps: Managing Conflict &amp; Effectively Using Git Workflow</h3>	 <p><b>Niranjan Pandey</b> Software Engineer and Big Data Expert</p>	<h3>Using Docker for DevOps: Introduction to Docker</h3>
<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ demonstrate the steps involved in managing and tracking branches in remote repositories</li> <li>▪ describe the concept of change integration and compare the differences between Git rebase and Git merge</li> <li>▪ work with Git merge and Git rebase to integrate changes from one branch to another</li> <li>▪ recall the prominent workflow patterns that can be used to integrate contributed works in a project and maintain overall workflow in the project</li> <li>▪ illustrate scenarios leading to merge conflict and the essential approaches that can be adopted to handle merge conflicts</li> <li>▪ demonstrate how to manage merge conflicts by editing conflicting files</li> <li>▪ implement three way merge using the merge tool</li> <li>▪ work with SourceTree to manage repositories and workflows</li> <li>▪ recall the essential features of GitFlow and GitHub Flow and recognize how they simplify branch management</li> <li>▪ manage Git workflows using GitFlow with a focus on managing feature branches</li> <li>▪ recall approaches for resolving merge conflicts, integrate changes from one branch to another using Git merge, and manage merge conflict by editing conflicting files</li> </ul>		<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ recognize containerization features and the different technologies that drive containerization</li> <li>▪ describe Docker features and the architecture of Docker engine</li> <li>▪ recall Docker use cases that drive DevOps processes</li> <li>▪ install and configure Docker on Windows and test the installed components</li> <li>▪ install and configure Docker on Linux and test the installed components</li> <li>▪ recall the features and benefits associated with deploying and working with Docker in AWS</li> <li>▪ deploy Docker Community Edition in AWS</li> <li>▪ recognize the Docker components that can be used to manage applications, data, and networks</li> <li>▪ list the important clauses that are used in Dockerfile to manage images</li> <li>▪ build images using Dockerfile</li> <li>▪ create repositories in Docker Hub and push container images to the Docker Hub from the local file system</li> <li>▪ recall the prominent features of Docker, create repositories in Docker Hub, and push container images to the Docker Hub</li> </ul>	

 <p><b>Niranjan Pandey</b> Software Engineer and Big Data Expert</p>	<p>Using Docker for DevOps: Configuring Docker for Continuous Delivery</p>	 <p><b>Niranjan Pandey</b> Software Engineer and Big Data Expert</p>	<p>Ansible: Ansible for DevOps Provisioning</p>
<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ recognize the features of various storage drivers provided by Docker</li> <li>▪ demonstrate how to manage data volumes using Docker the command line interface</li> <li>▪ classify virtual machines, containers, and images and recall the benefits of customizing images in DevOps</li> <li>▪ use the Compose tool to run multi-container Docker applications</li> <li>▪ describe the essential features of networking containers and list the default networks provided for Docker</li> <li>▪ use and work with Docker CLI commands to create and manage networks in Docker</li> <li>▪ install, configure, and build projects using Docker Assemble</li> <li>▪ set up Docker clusters using Swarm</li> <li>▪ list the prominent DevOps and continuous delivery reference architectures for Docker</li> <li>▪ describe the differences between virtual machines, containers, and images, manage data volumes, and create a network in Docker using Docker command line</li> </ul>		<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ recognize the role of Ansible in the implementation of DevOps principles and the key phases of Ansible automation for DevOps</li> <li>▪ list the key features and the critical roles played by the essential components of Ansible</li> <li>▪ demonstrate how to install Control Machine and Managed Nodes to set up Ansible</li> <li>▪ configure Ansible Sandbox using virtualization on local machines</li> <li>▪ recall the essential configuration parameters of Ansible, with focus on configuration file and environmental configuration</li> <li>▪ create inventory files to enable communication with Ansible Server</li> <li>▪ demonstrate how to use important administrative commands in Ansible to manage users, servers, and files</li> <li>▪ demonstrate the steps involved in setting up Ansible and powering GUI capability using AWX</li> <li>▪ recognize the features of Ansible playbooks and identify the important tags used in YAML to create playbooks for configuration management</li> <li>▪ demonstrate how to run Ansible playbooks on AWX server</li> <li>▪ recall the essential configuration parameters of Ansible, install Control Machine and Managed Node to set up Ansible and run Ansible Playbook on Ansible AWX server</li> </ul>	

 <p><b>Niranjan Pandey</b> Software Engineer and Big Data Expert</p>	<p><b>Ansible: Continuous Delivery and Monitoring Using Ansible Playbook and Tower</b></p>	 <p><b>Niranjan Pandey</b> Software Engineer and Big Data Expert</p>	<p><b>Chef for DevOps: Managing Infrastructure Using Chef</b></p>
<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ demonstrate a case study of working with Ansible Playbook to implement server configuration, deployment, and application launch</li> <li>▪ implement continuous delivery using Ansible Playbook to deliver software</li> <li>▪ create AWS EC2 instances using Ansible Role and Ansible Playbook</li> <li>▪ describe Ansible Tower features that help to simplify Ansible configuration management</li> <li>▪ install Ansible Tower on Linux</li> <li>▪ launch Ansible Tower in the cloud</li> <li>▪ set up and work with the various components of Ansible Tower dashboard</li> <li>▪ create projects and manage jobs using Ansible Tower</li> <li>▪ recognize the logging mechanism adopted by Ansible Tower</li> <li>▪ configure and enable Logging Aggregator Services to configure the logging service in Ansible Tower</li> <li>▪ recall Ansible Tower configuration management features, install Ansible Tower on Linux, and create a project using Ansible Tower</li> </ul>		<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ recognize the characteristics and importance of implementing Infrastructure as Code using Chef and DevOps</li> <li>▪ identify the capabilities, core principles, and benefits of Chef</li> <li>▪ list the components of Chef that are used to implement Infrastructure as Code</li> <li>▪ set up Git to create a Chef repository and implement Infrastructure as Code</li> <li>▪ recognize the features of Chef Workstation and list the various components and tools of Chef Workstation</li> <li>▪ recall the features of Cookbook and list the various components that make up a Cookbook</li> <li>▪ classify the various types of nodes that can be managed using Chef</li> <li>▪ demonstrate the steps involved in working with Hosted Chef</li> <li>▪ install Chef server and set up a Workstation</li> <li>▪ recall the components of Chef used to implement Infrastructure as Code, list the types of nodes that can be managed using Chef, and install Chef server</li> </ul>	

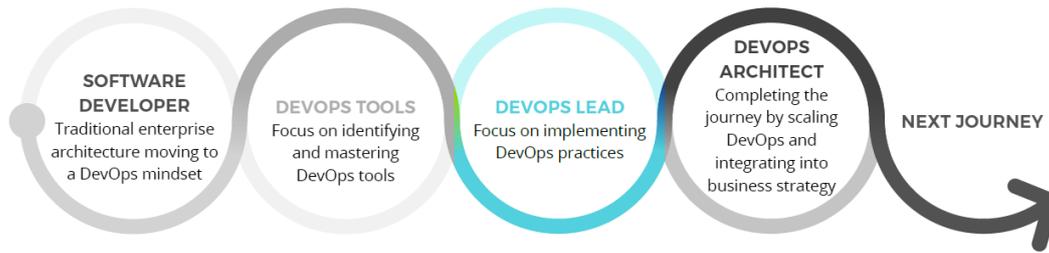
	<b>Chef for DevOps: Automate Infrastructure Using Chef Tools &amp; Templates</b>		<b>Puppet for DevOps: Installing &amp; Working with Puppet Components</b>
<b>Objectives</b> <ul style="list-style-type: none"> <li>▪ set up Chef Client and Knife</li> <li>▪ generate Chef cookbooks and configure Knife</li> <li>▪ bootstrap a node to pull and apply configuration changes</li> <li>▪ install Kitchen and use and work with Test Kitchen</li> <li>▪ describe the concept of environment in Chef, its attributes, and the sequence in which the attributes are applied in the environment</li> <li>▪ recognize the essential features of Chef Habitat and illustrate the package build flow, from automated build to package deployment</li> <li>▪ install Chef Habitat, work with Chef Habitat Builder to facilitate automated builds, and deploy Chef Habitat packages</li> <li>▪ describe Chef templates and the components that can be used to build them</li> <li>▪ create recipes using template files</li> <li>▪ recall the features and capabilities provided by Chef Automate for infrastructure automation and orchestration</li> <li>▪ work with Chef Automate to facilitate infrastructure automation and orchestration</li> <li>▪ describe scenarios where Chef may not be the best candidate for configuration management</li> <li>▪ describe Chef Habitat, install Chef Habitat, and use Chef Habitat Builder to implement automated deployment of Chef Habitat packages</li> </ul>		<b>Objectives</b> <ul style="list-style-type: none"> <li>▪ describe the implementation of DevOps practices using Puppet and user stories</li> <li>▪ illustrate the architecture of Puppet and list the prominent open-source products it provides</li> <li>▪ list the core components of Puppet and describe the application and infrastructure management capabilities of Puppet</li> <li>▪ Install and configure Puppet components including Puppet server, Puppet master, and Puppet agent</li> <li>▪ recognize the key puppet configuration settings for Puppet master server and Puppet agents</li> <li>▪ demonstrate how to use and work with essential Puppet commands</li> <li>▪ recognize the concept, structure, and essential elements of the Puppet module</li> <li>▪ demonstrate how to manage Puppet modules using commands</li> <li>▪ recall the essential components and characteristics of the Puppet language that are used to configure network and operating system resources</li> <li>▪ install the Puppet Development Kit and use it to create and test modules</li> <li>▪ recall the open-source products provided by Puppet, install the Puppet Development Kit, and create modules using the Puppet Development Kit</li> </ul>	

 <p><b>Niranjan Pandey</b> Software Engineer and Big Data Expert</p>	<p><b>Puppet for DevOps: Working with Puppet Bolt &amp; Continuous Delivery for PE</b></p>	 <p><b>Satyendra Maurya</b> Project Manager and E-Learning Content Development Specialist</p>	<p><b>Jenkins for DevOps: Jenkins Configuration for DevOps</b></p>
<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ describe the concepts of data and fact management using Puppet Hiera and Facter</li> <li>▪ demonstrate how to configure and work with Hiera to manage data in Puppet</li> <li>▪ describe the essential characteristics of Puppet Bolt and its features that can be used to orchestrate and automate manual tasks</li> <li>▪ install Bolt and create configuration files to automate workflows with plans and tasks</li> <li>▪ describe plans and tasks in Puppet Bolt and specify the structure of modules with plans and tasks</li> <li>▪ demonstrate how to use Puppet Bolt to work with plans and tasks for manual task automation</li> <li>▪ install and configure Continuous Delivery for PE</li> <li>▪ set up and build continuous delivery pipelines using Continuous Delivery for PE</li> <li>▪ define the concept of Jobs in Puppet and recall the various pre-built jobs and essential required fields that can be used to create new jobs</li> <li>▪ demonstrate the approach of using the Impact Analysis Tool to identify the impact of Puppet code on PE managed infrastructures</li> <li>▪ recognize the potential challenges of using Puppet as a continuous delivery tool</li> <li>▪ recall the potential challenges of using Puppet as a continuous delivery tool, install and configure Continuous Delivery for PE, and build continuous delivery pipelines using Continuous Delivery for PE</li> </ul>		<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ identify the role of Jenkins in implementing DevOps processes</li> <li>▪ recognize the use cases of implementing Jenkins to build continuous integration and continuous delivery pipelines</li> <li>▪ install and configure Jenkins in distributed environments</li> <li>▪ list the different types of jobs that can be created using Jenkins, along with their features</li> <li>▪ specify the various types of build triggers and build steps that can be configured in Jenkins</li> <li>▪ configure and implement freestyle jobs in Jenkins</li> <li>▪ implement pipelines in Jenkins using WebUI</li> <li>▪ implement pipelines in Jenkins using a Jenkinsfile</li> <li>▪ recall the different types of Jenkins jobs, list the different types of build triggers that can be configured in Jenkins, and implement pipelines in Jenkins using WebUI</li> </ul>	

 <p><b>Satyendra Maurya</b> Project Manager and E-Learning Content Development Specialist</p>	<p><b>Jenkins for DevOps: Automated Testing &amp; Advanced Jobs Using Jenkins</b></p>	 <p><b>Niranjan Pandey</b> Software Engineer and Big Data Expert</p>	<p><b>SaltStack for DevOps: Working with SaltStack Components</b></p>
<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ describe automated testing and the benefits of using Jenkins for test automation</li> <li>▪ configure Jenkins for building Java projects using Maven to implement build and test automation</li> <li>▪ automate unit testing Java projects using Jenkins</li> <li>▪ configure and implement performance testing using JMeter and Jenkins</li> <li>▪ configure and execute TestNG tests in Jenkins</li> <li>▪ generate TestNG reports using Jenkins</li> <li>▪ build parameterized build jobs in Jenkins</li> <li>▪ configure multi-configuration build jobs in Jenkins</li> <li>▪ configure Jenkins jobs to run in batches</li> <li>▪ recognize the most common scenarios where Jenkins can't be used</li> <li>▪ describe the benefits of using Jenkins for test automation, build parameterized build jobs in Jenkins, and configure multi-configuration build jobs in Jenkins</li> </ul>		<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ list the various products offered by SaltStack and their implementation scenarios</li> <li>▪ recognize the essential characteristics of Salt and the role Salt can play in implementing DevOps practices</li> <li>▪ describe the architecture of Salt along with its various components</li> <li>▪ install Salt and illustrate the basic configuration of Salt master and Salt minion</li> <li>▪ use Salt agentless to run Salt commands without installing Salt minion</li> <li>▪ demonstrate the various operations that can be performed using Grains in Salt</li> <li>▪ specify the essential functions in SaltUtil and job runner</li> <li>▪ specify the essential components that are required to use Salt effectively</li> <li>▪ demonstrate the steps involved in writing a Salt runner and the different modes of running Salt runners</li> <li>▪ write and configure Salt engine to enhance external processes</li> <li>▪ demonstrate the steps involved in writing execution modules and calling them on Salt master</li> <li>▪ recall the various architectural components of SaltStack, write a Salt runner, and write execution modules</li> </ul>	

 <p><b>Niranjan Pandey</b> Software Engineer and Big Data Expert</p>	<p>SaltStack for DevOps: Configuration Management with SaltStack</p>		<p>Final Exam: DevOps Developer</p>
<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ describe the configuration management methodology used in SaltStack and the default execution order of managing custom task executions in Salt</li> <li>▪ describe the concepts of Salt states and Salt formulas and list the official Salt formulas</li> <li>▪ configure a system to be managed by Salt state</li> <li>▪ demonstrate how to reuse Salt state and use &lt;include&gt; in Salt state management</li> <li>▪ add Salt formulas as a GitFS remote and add Salt formulas manually to the Salt master</li> <li>▪ describe the features of Jinja2 templating engine</li> <li>▪ describe the concept of pillars and differentiate in-memory pillar data from on-demand pillar data</li> <li>▪ work with pillar data</li> <li>▪ describe the capability of the event-driven architecture of Salt with focus on the event system and event representation</li> <li>▪ describe the concept and usages of beacons and reactors in Salt</li> <li>▪ recall the official Salt formulas, list the features of the Jinja2 templating engine, and add Salt formulas as a GitFS remote</li> </ul>		<p>This final exam will test your understanding and application of the skills covered in the DevOps Developer track of the Skillsoft Aspire Enterprise Developer to DevOps Engineer Journey.</p>	

**ENTERPRISE DEVELOPER TO DEVOPS ENGINEER**



**Track 3: Track 3: DevOps Lead (duration: 13h 0m 49s)**

 <p><b>Niranjan Pandey</b> Software Engineer and Big Data Expert</p>	<p><b>Best Practices for DevOps Implementation</b></p>	 <p><b>Niranjan Pandey</b> Software Engineer and Big Data Expert</p>	<p><b>DevOps Cloud Automation: AWS DevOps Tools</b></p>
<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ identify key elements that drive the implementation of DevOps culture in the enterprise</li> <li>▪ recall the challenges of transforming traditional approaches to DevOps and describe guidelines and rules that can be adopted when transforming to DevOps</li> <li>▪ identify the significance of adopting the Single Source Repository pattern and list the tools and methods that can be used to implement collaborative artifact sharing</li> <li>▪ recognize best practices and guidelines that should be followed when implementing and maintaining CI/CD systems in order to best serve organizational project goals and requirements</li> <li>▪ describe the concept, process, workflow, and need for an artifact repository, along with tools that can be used to implement it</li> <li>▪ configure JFrog Artifact repository to store sharable artifacts with the DevOps team and enhance the collaborative development and delivery mechanism and pipelines</li> <li>▪ recall the benefits of implementing automation testing and the benefits of using test automation in CI/CD pipelines</li> <li>▪ describe the Chaos Engineering principle, how it can help identify weaknesses in information systems, and the guiding principles of Chaos Engineering that can impact DevOps adoption</li> <li>▪ list prominent tools that can be used to implement DevOps at scale along with their associated features</li> <li>▪ recognize the deployment strategies adopted in DevOps and compare strategy adoption scenarios</li> <li>▪ configure and implement Blue-Green deployment using OpenShift</li> <li>▪ recognize the best practices for implementing automation at scale, with a focus on infrastructure, application, and compliance</li> <li>▪ describe the Twelve-Factor App methodology that can be used as a guideline to build Software-as-a-Service applications</li> <li>▪ describe the features and benefits of employing cloud-native technologies for development and deployment of applications</li> <li>▪ build cloud-native applications with Spring Boot on OpenShift</li> <li>▪ summarize the key concepts covered in this course</li> </ul>		<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ describe the benefits of DevOps principles and cloud computing as they relate to the implementation of enterprise-grade applications</li> <li>▪ describe the DevOps workflow, including the cloud components used to help build robust delivery pipelines</li> <li>▪ install the AWS CLI and PowerShell to access AWS and Azure resources</li> <li>▪ list AWS components that play important roles in setting up DevOps pipelines to build, test, and deploy applications</li> <li>▪ list Azure components that play important roles in setting up DevOps pipelines to build, test, and deploy applications</li> <li>▪ set up CodeCommit to provision a code repository and use CodeCommit for code management and versioning</li> <li>▪ set up Cloud9 to enable code writing, running, and debugging from CodeCommit</li> <li>▪ use CodeBuild to implement continuous integration for compiling source code, running tests, and producing software packages</li> <li>▪ automate release pipelines using CodePipeline</li> <li>▪ recognize scenarios where CodeDeploy can be used to automate deployment on various target environments</li> <li>▪ automate software deployments to Amazon EC2 using CodeDeploy</li> <li>▪ summarize the key concepts covered in this course</li> </ul>	

 <p><b>DevOps Cloud Automation: Advanced AWS Pipelines and DevOps Using Azure</b></p>	 <p><b>DevOps Cloud Automation: DevOps with Google Cloud Platform</b></p>
<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ automate cloud deployments using CodeBuild and CodePipeline, specifying the roles of ECR, ECS, and containers</li> <li>▪ differentiate between AWS Pipeline and Jenkins for automating code build and deployment</li> <li>▪ monitor and audit AWS resources using OpsWorks, System Manager, CloudWatch, CloudTrail, and Xray</li> <li>▪ describe the features and benefits of the CodeStar development workflow that can be used to quickly develop, build, and deploy applications on AWS</li> <li>▪ work with AWS CodePipeline in the AWS Cloud9 Integrated Development Environment</li> <li>▪ describe the role of CloudTrail in enabling governance, compliance, and operational and risk auditing of AWS accounts used for automated deployments</li> <li>▪ automate the software delivery process using the AWS Continuous Integration and Delivery pipelines</li> <li>▪ describe the features and components of Azure DevOps that teams can use to plan work, collaborate on code development, and build and deploy applications</li> <li>▪ demonstrate how to set up project, team, and GitHub to select appropriate processes and Agile process workflows using Azure Board</li> <li>▪ use Azure Pipelines to build GitHub repositories and configure pipelines to build, test, and manage releases</li> <li>▪ build multi-stage continuous deployment pipelines</li> <li>▪ add the Azure DevOps extension for the Azure CLI to implement Azure DevOps Services from the CLI</li> <li>▪ use Azure DevOps tools to collaborate and enable faster shipping through the use of a set of modern dev services</li> <li>▪ use Azure Artifacts to share Maven artifacts from public and private sources</li> <li>▪ use the Azure App Service to build and host applications</li> <li>▪ summarize the key concepts covered in this course</li> </ul>	<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ describe the Google Cloud Platform DevOps capabilities that can be used to adopt CI/CD processes when automating application lifecycle</li> <li>▪ describe Google's reference pipeline, which helps with automatically building, testing, and deploying code changes across different platforms and providing continuous integration and delivery capabilities</li> <li>▪ describe the benefits of using Google Cloud Platform for configuration management and list the tools provided by Google Cloud Platform for configuration management</li> <li>▪ set up the Cloud Source Repositories provided by GCP to store, manage, and track code</li> <li>▪ use CodeBuild to build workflows for building, testing, and deploying applications</li> <li>▪ automatically deploy applications stored in Cloud Source Repositories to the App Engine after new commits</li> <li>▪ implement Pub/Sub notifications in Cloud Source Repositories to facilitate continuous monitoring</li> <li>▪ deploy cloud functions from Cloud Source Repositories</li> <li>▪ summarize the key concepts covered in this course</li> </ul>

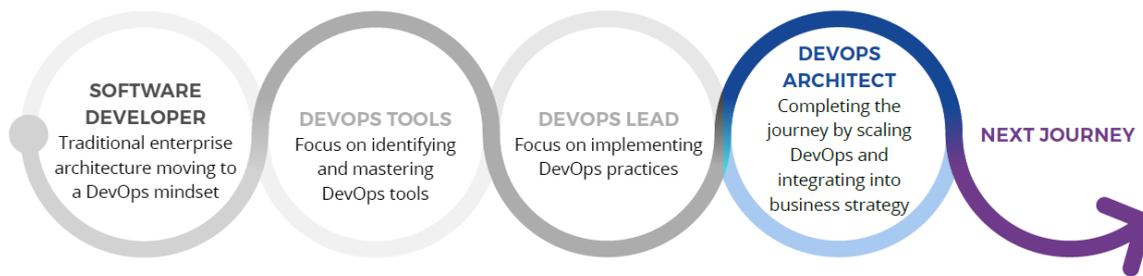
 DevOps Automation: Configuration Management	 DevOps Automation: Delivery Mechanisms
<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ recognize the DevOps tasks that can help you adopt fundamental principles of implementing DevOps automation</li> <li>▪ describe the infrastructure required to manage configuration and enable artifact management and deployment automation from the perspective of cloud and container</li> <li>▪ describe the automation benefits of DevOps principles</li> <li>▪ describe practices that help organizations adopt the CAMS principle for DevOps automation</li> <li>▪ list the steps of involved in implementing DevOps, including the practices and technologies that are used for each step</li> <li>▪ list use cases that demonstrate the implementation of automation for virtualization, deployment, release, and cloud management</li> <li>▪ automate creating EC2 instances and setup RDS with the use of Infrastructure as Code</li> <li>▪ demonstrate the role of configuration management and DevOps automation using CFEngine</li> <li>▪ describe the release management tools that can be used to improve releases using DevOps delivery pipelines</li> <li>▪ configure and implement release management for applications using Git and Jenkins pipelines</li> <li>▪ summarize the key concepts covered in this course</li> </ul>	<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ identify delivery models and describe the benefits of adopting modern delivery models and application delivery platforms</li> <li>▪ describe tools that can be used to automate builds and their benefits</li> <li>▪ automate build workflows using Maven and Jenkins</li> <li>▪ automate deployment using deployment workflows and describe the benefits of automated deployment</li> <li>▪ describe DevOps test automation workflows and the DevOps testing strategy</li> <li>▪ develop test automation frameworks from scratch</li> <li>▪ describe infrastructure automation and methods and tools that can be used to implement it</li> <li>▪ list DevOps tools that can be used to implement the functional and automation features of the DevOps workflow</li> <li>▪ recall the steps involved in implementing a "single source of truth", with focus on managing development and deployment artifacts</li> <li>▪ summarize the key concepts covered in this course</li> </ul>

 <p><b>Niranjan Pandey</b> Software Engineer and Big Data Expert</p>	<h3>CI/CD Implementation for DevOps</h3>	 <p><b>Niranjan Pandey</b> Software Engineer and Big Data Expert</p>	<h3>DevOps with Docker: Container Management</h3>
<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ describe CI/CD and its role in implementing DevOps to produce well tested software</li> <li>▪ differentiate between CI and CD based on their roles in delivering software to production</li> <li>▪ set up deployment pipelines using BitBucket to gain visibility into the production readiness of applications</li> <li>▪ describe deployment pipelines and how they help implement continuous delivery</li> <li>▪ list the pillars of continuous integration, the phases involved in the implementation of continuous integration, and best practices</li> <li>▪ implement continuous integration using Atlassian tools</li> <li>▪ set up the tools required to implement continuous integration</li> <li>▪ demonstrate the steps of implementing continuous delivery using Jenkins Workflow and its components</li> <li>▪ use Jenkins Workflow and Jenkins DSL to integrate with other tools, jobs, and environments</li> <li>▪ implement CI/CD pipelines using GitLab to illustrate the build, test, and deploy stages</li> <li>▪ create continuous delivery pipelines in Jenkin2 to build, test, and publish with Docker</li> <li>▪ build declarative pipelines using Jenkins</li> <li>▪ work with Jenkins to orchestrate building simple Node.js and React applications with the Node Package Manager</li> <li>▪ describe the concept of Blue Ocean and differentiate between Red and Blue Ocean strategies</li> <li>▪ set up Blue Ocean in Jenkins, access the Blue Ocean interface, and create pipeline projects in Jenkins</li> <li>▪ summarize the key concepts covered in this course</li> </ul>		<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ compare the traditional virtualization with container architecture and list the benefits of adopting containerization in DevOps workflows</li> <li>▪ describe major Docker components and the features of the Runc driver</li> <li>▪ list the supporting Docker technologies that can be used to manage networking, service discovery, orchestration, and clusters</li> <li>▪ work with Docker commands to manage containers, associate containers with IPs, and link containers in Docker using the self-discovery approach</li> <li>▪ implement networking in Docker using network drivers to setup container networking</li> <li>▪ set up custom bridges for Docker and use the Open vSwitch virtual switch instead of the standard Linux bridge</li> <li>▪ recall essential commands that are used in Dockerfile to support the build and run phases of managing images</li> <li>▪ create Docker images with Dockerfile for creating and deploying applications in isolated environments</li> <li>▪ describe image distribution and versioning using repositories and Docker Hub</li> <li>▪ summarize the key concepts covered in this course</li> </ul>	

	<b>DevOps with Docker: Implementing DevOps Using Docker</b>		<b>DevOps with Docker: Management in the Cloud</b>
<b>Objectives</b> <ul style="list-style-type: none"> <li>▪ discover the key concepts covered in this course</li> <li>▪ run local registries and push and store images to local registries</li> <li>▪ implement tunnels to network containers that are present on multiple hosts using Docker</li> <li>▪ describe the activities required to configure Docker for development</li> <li>▪ use Docker Remote API to automate Docker tasks</li> <li>▪ set up an end-to-end Docker-based local development environment and build applications powered by Docker</li> <li>▪ work with Docker and Jenkins to create the continuous integration workflow for building and testing applications</li> <li>▪ provision new resources and run containers on the new resources using Docker Machine for continuous deployment</li> <li>▪ facilitate the continuous monitoring features of DevOps by implementing ELK for continuous logging and monitoring</li> <li>▪ describe Sonatype Nexus and Artifactory repository manager features</li> <li>▪ install and configure Sonatype Nexus to set up a repository for artifact management</li> <li>▪ summarize the key concepts covered in this course</li> </ul>		<b>Objectives</b> <ul style="list-style-type: none"> <li>▪ discover the key concepts covered in this course</li> <li>▪ describe the tools and their features that can be used to simplify networking and service discovery in Docker</li> <li>▪ connect containers across hosts using Ambassadors</li> <li>▪ implement comprehensive Docker container network solutions using Weave</li> <li>▪ recognize the features of tools used for Docker orchestration, clustering, and management</li> <li>▪ describe the architecture of Kubernetes and the important role it plays in implementing CI/CD using containerized Docker components</li> <li>▪ create and configure Minikube clusters and set up the Minikube environment to communicate with the Docker daemon</li> <li>▪ use Compose to deploy applications on Minikube</li> <li>▪ describe tools that can be used to adopt DevOps continuous delivery principles for managing container-based architectures</li> <li>▪ create GKE clusters and install Jenkins X</li> <li>▪ build CI/CD pipelines using Jenkins X</li> <li>▪ summarize the key concepts covered in this course</li> </ul>	

 <p><b>Niranjan Pandey</b> Software Engineer and Big Data Expert</p>	<h3>DevOps Continuous Testing: Testing Approaches</h3>	 <p><b>Niranjan Pandey</b> Software Engineer and Big Data Expert</p>	<h3>Testing: Testing Methodologies</h3>
<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ discover the key concepts covered in this course</li> <li>▪ recognize the challenges associated with an Agile environment and approaches for testing DevOps artifacts</li> <li>▪ describe continuous testing and compare continuous testing and test automation</li> <li>▪ list the disadvantages of late testing and describe shift left testing and how to implement it</li> <li>▪ describe the concept, types, and common causes of technical debt</li> <li>▪ recognize differences between unit testing, behavior driven development, and test driven development</li> <li>▪ describe the BDD-based Agile workflow that is applied in DevOps and list the benefits of BDD for programs and release management</li> <li>▪ describe tools that can be used to implement continuous testing and their features</li> <li>▪ describe unit testing, its essential elements, and the tools that can be used to implement it</li> <li>▪ implement unit testing for Java programs using JUnit, with focus on TestCase and TestSuite</li> <li>▪ test SOAP-based web services using the SOAP UI</li> <li>▪ recognize the role and features of integration testing</li> <li>▪ summarize the key concepts covered in this course</li> </ul>		<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ discover the key concepts covered in this course</li> <li>▪ list system testing approaches and the steps involved in the process</li> <li>▪ recognize project management methodologies that are used to efficiently manage projects</li> <li>▪ configure projects using Scrum and Kanban methodologies</li> <li>▪ create backlogs, and plan and start Sprint with issues</li> <li>▪ recognize the need for user acceptance testing and describe how to conduct it</li> <li>▪ use QMetry Test Management for Jira</li> <li>▪ list performance testing components and types of performance testing for deployed applications</li> <li>▪ recognize the need for load testing and list tools that can be used to test application loads and collect essential application metrics</li> <li>▪ use JMeter to test static and dynamic resources</li> <li>▪ recognize the role and significance of automated testing in DevOps pipelines</li> <li>▪ summarize the key concepts covered in this course</li> </ul>	

	<h3>Final Exam - DevOps Lead</h3>
<p>This final exam will test your understanding and application of the skills covered in the DevOps Lead track of the Skillsoft Aspire Enterprise Developer to DevOps Engineer Journey.</p>	



Track 4: DevOps Engineer (duration: 8h 42m 40s)

 <p>Niranjan Pandey Software Engineer and Big Data Expert</p>	Scaling DevOps: Pipelines at Scale	 <p>Niranjan Pandey Software Engineer and Big Data Expert</p>	Scaling DevOps: Infrastructure at Scale
	<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>list features and factors that should be considered when adopting DevOps in scaling environments</li> <li>recall considerations for designing scalable DevOps architectures</li> <li>recognize the potential implementation challenges when scaling DevOps architectures and the impact of DevOps adoption on businesses</li> <li>recall the key elements of building optimized DevOps pipelines for managing DevOps at scale</li> <li>list the DevOps tools that can be used to build DevOps pipelines to manage infrastructure at scale, with a focus on DevOps automation</li> <li>describe the approach of scaling DevOps with loosely coupled architectures, with a focus on barriers and guard rails</li> <li>list approaches for building architectures used to set up DevOps pipelines at scale</li> <li>set up continuous build and continuous integration to manage DevOps architecture at scale</li> <li>describe the components and their integration approaches that are used in designing robust DevOps pipelines with Agile principles</li> <li>recall the DevOps patterns that can be used to scale web applications in the cloud</li> <li>recognize the essential features of application lifecycle management, with a focus on the phases of ALM</li> <li>summarize the key concepts covered in this course</li> </ul>		<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>describe Infrastructure as Code and list the tools that can be used to implement it, while applying best principles for provisioning infrastructure for scaling DevOps</li> <li>recognize the challenges involved provisioning the right levels of operational maturity and describe evolving provisioning practices should be integrated in DevOps workflows</li> <li>install and configure OpenShift container platform to manage IT infrastructure and evaluate its enterprise applicability</li> <li>list the features of Kubernetes that enable deployment automation, and the scaling and management of containerized applications</li> <li>recall tools that can be used to log, monitor, and configure alerts for scaling DevOps architectures</li> <li>use Sentry to monitor errors and help all software teams discover and prioritize errors in real-time</li> <li>identify the various types of application whitelisting and the steps involved in planning and implementing whitelisting</li> <li>implement code coverage with JaCoCo and SonarQube using appropriate quality gates</li> <li>install Sonatype and manage code and infrastructure artifacts</li> <li>implement build pipelines using Jenkins Blue Ocean</li> <li>recognize the standard approach of setting up teams to manage projects at scale, with a focus on adopting Agile and DevOps standards</li> <li>use git-secret to manage secure commits in a repository</li> <li>describe multi-account AWS architecture and the AWS landing zone that speeds up large-scale migrations</li> <li>manage Docker images locally and remotely on a container registry</li> <li>summarize the key concepts covered in this course</li> </ul>

 <p><b>Enterprise DevOps with Docker</b></p>	 <p><b>DevOps Continuous Feedback: Implementing Continuous Feedback</b></p>
<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ recall the Docker containerization strategy that can help leverage existing IT resources</li> <li>▪ describe the roadmap for building modern applications and the steps and benefits of implementing containerization using Docker</li> <li>▪ describe the Docker Maturity Model and Docker features that can help scale application maturity with the implementation of containerization</li> <li>▪ describe the general guidelines and recommendations for implementing containerization</li> <li>▪ illustrate the architecture of Docker Enterprise and recognize the roles of essential Docker Enterprise components</li> <li>▪ recall the key products and services of Docker Enterprise Edition and their associated features</li> <li>▪ set up and configure hosted Docker Enterprise</li> <li>▪ add nodes and provision repositories to store and manage Docker images using Docker Enterprise</li> <li>▪ build a multi-OS application, and deploy and scale it to Docker Enterprise using Swarm</li> <li>▪ build a multi-OS application, and deploy and scale it to Docker Enterprise using Kubernetes</li> <li>▪ recall approaches of securing Docker Enterprise and list the security best practices that help eliminate risks</li> <li>▪ recognize the benefits of audit logs and describe log levels that can be configured using Docker Enterprise to capture security-relevant activities</li> <li>▪ use UCP logs in Docker Enterprise</li> <li>▪ manage Docker Enterprise by backing up and restoring configuration and work with the telemetry plugin in Docker Enterprise</li> <li>▪ summarize the key concepts covered in this course</li> </ul>	<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ recall the primary goals of adopting DevOps practices in the enterprise and its impact on people, process, and product</li> <li>▪ recall approaches of implementing DevOps that adhere to all essential principles and patterns of DevOps</li> <li>▪ recall complexities that can impact enterprise systems and the role of DevOps in building a stable system</li> <li>▪ describe the DevOps Feedback Loop and identify the maturity model curve for building well-defined feedback mechanisms in the DevOps lifecycle</li> <li>▪ list essential elements of Feedback Loop and compare positive and negative Feedback Loops</li> <li>▪ recognize best practices for creating and implementing Feedback Loops in DevOps, with focus on continuous improvement</li> <li>▪ describe the phases of the DevOps lifecycle and the benefit of implementing Feedback Loops in each phase</li> <li>▪ recognize the features of Jira Issue Collector and its role in building and implementing Feedback Loops</li> <li>▪ list Feedback Loop implementation tools and enable continuous monitoring</li> <li>▪ recognize the role of monitoring in the DevOps delivery mechanism</li> <li>▪ list various types of monitors that can be used to implement Feedback Loops and create action plans to maintain system robustness</li> <li>▪ describe monitoring target categories and the role of automated monitoring and alerting in building better software</li> <li>▪ identify the DevOps metrics that can be used for monitoring throughout the different phases of the DevOps lifecycle</li> <li>▪ recognize tools that can be used for feedback monitoring and build alerts to implement continuous Feedback Loops</li> <li>▪ describe telemetry and its role in the DevOps software development lifecycle</li> <li>▪ summarize the key concepts covered in this course</li> </ul>

 <p><b>Niranjan Pandey</b> Software Engineer and Big Data Expert</p>	<p><b>DevOps Continuous Feedback: Loop Tools</b></p>	 <p><b>Niranjan Pandey</b> Software Engineer and Big Data Expert</p>	<p><b>DevOps Security Considerations: Securing DevOps Pipeline</b></p>
<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ recognize the DevOps metrics that can be used to evaluate successful implementation of DevOps practices</li> <li>▪ work with Datadog and set up agents to collect metrics and events from systems and applications</li> <li>▪ integrate issue tracking tools with Datadog to create issues from triggered alerts in Datadog</li> <li>▪ recognize the tools provided by Grafana Labs to query, visualize, alert on, and understand metrics</li> <li>▪ install Grafana and add CloudWatch as a data source to build dashboards for CloudWatch metrics</li> <li>▪ set up New Relic and explore the capabilities of New Relic using the New Relic dashboard</li> <li>▪ use New Relic to securely retrieve monitoring data from AWS accounts</li> <li>▪ configure and use New Relic to monitor application performance of Node.js applications</li> <li>▪ configure and use New Relic to monitor the performance of SQL and NoSQL databases</li> <li>▪ define alert policies for incident management using New Relic</li> <li>▪ use the New Relic Digital Intelligence Platform to get end-to-end visibility into the performance of web sites and applications</li> <li>▪ list the features of Relic APM that provide better end-to-end visibility of applications</li> <li>▪ work with the New Relic Infrastructure UI and the components that enable flexible and dynamic server monitoring</li> <li>▪ describe the features of New Relic Synthetics and its advanced testing capabilities</li> <li>▪ work with New Relic's Digital Intelligence Solution to enable end-to-end visibility into the performance of web sites and applications using service maps and health maps</li> <li>▪ establish application baselines and create key transactions in APM using New Relic</li> <li>▪ create and share custom dashboards, build charts, query data, and explore data using the New Relic Logs UI</li> <li>▪ enable AWS Serverless application for New Relic Logs by integrating with AWS CloudWatch</li> <li>▪ install and configure Trac to build wiki and issue tracking systems for software development projects</li> <li>▪ summarize the key concepts covered in this course</li> </ul>		<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ describe policy requirements and the guidelines that set the implementation standard for the security requirements specified in the baseline IT security policy</li> <li>▪ recognize the role of security in DevOps when building and developing the DevOps workflow to implement CI/CD processes</li> <li>▪ describe the implementation of continuous security in DevOps pipelines, with a focus on test-driven development, monitoring, responding to attacks, assessing risks, and maturing security</li> <li>▪ recognize the need for applying test-driven security on applications, infrastructure, and automated DevOps pipelines</li> <li>▪ describe components that can be used to build logging and auditing pipelines</li> <li>▪ recall the tools and processes that can be used to apply security for DevOps-compliant processes in order to manage intrusion and incident responses</li> <li>▪ control permissions granted to users in GitHub and manage permissions in Docker Hub</li> <li>▪ configure Jenkins security features to secure continuous build pipelines</li> <li>▪ recognize the steps involved in implementing mature a DevOps security model that can help assess risks and facilitate continuous security</li> <li>▪ summarize the key concepts covered in this course</li> </ul>	

	<p>DevOps Security Considerations: DevSecOps Principles</p>		<p>Final Exam: DevOps Engineer</p>
<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>▪ compare DevOps and DevSecOps and describe the benefits of adopting the DevSecOps paradigm</li> <li>▪ list the essential phases of the DevSecOps workflow and describe the critical checklists that can help secure DevOps implementations</li> <li>▪ describe DevSecOps tools that are used by development teams to integrate security throughout the DevOps pipeline</li> <li>▪ describe the DevSecOps Maturity Model and the dimensions that can help automate security checks</li> <li>▪ describe how security and testing can be integrated into a DevSecOps environment without compromising speed, security, or quality</li> <li>▪ use the Jenkins OAuth plugin to securely pull from GitHub</li> <li>▪ recall best practices for securing Jenkins Secrets to store and manage credentials</li> <li>▪ list the categories of tools provided by OWASP to secure applications</li> <li>▪ use penetration testing tools to conduct penetration testing on deployed application to identify vulnerabilities</li> <li>▪ summarize the key concepts covered in this course</li> </ul>		<p>This final exam will test your understanding and application of the skills covered in the DevOps Engineer track of the Skillsoft Aspire Enterprise Developer to DevOps Engineer Journey.</p>	

## Business & Leadership for DevOps Engineer

### Business & Leadership for DevOps Engineer ⓘ Optional

 <p>COURSE The Essential Role of the Agile Product Owner</p> <p>100</p>	 <p>COURSE Managing Pressure and Stress to Optimize Your...</p> <p>322</p>	 <p>COURSE Defining Alternative Solutions to a Problem</p> <p>104</p>	 <p>COURSE Developing a Growth Mind-set</p> <p>482</p>	 <p>COURSE Cultivating Cross-functional Team Collaboration</p> <p>25</p>
 <p>COURSE Confronting Your Assumptions</p> <p>184</p>	 <p>COURSE Managing a Project to Minimize Risk and Maximiz...</p> <p>93</p>	 <p>COURSE Using Strategic Thinking to Consider the Big Picture</p> <p>187</p>	 <p>COURSE Reaching Efficient Solutions with Computational Thinking</p> <p>20</p>	 <p>COURSE Running Meetings in Better Directions</p> <p>49</p>
 <p>COURSE Investigating Arguments</p> <p>93</p>	 <p>COURSE Innovating with Lean Product Management</p> <p>92</p>			

## Productivity Tools for DevOps Engineer

### Productivity Tools for DevOps Engineer ⓘ Optional

 <p>COURSE Signing in &amp; Navigating within Spaces</p> <p>11</p>	 <p>COURSE Setting Up &amp; Managing Spaces</p> <p>9</p>	 <p>COURSE Working with Spaces</p> <p>7</p>	 <p>COURSE Working with Team Members</p> <p>27</p>	 <p>COURSE Configuring Spaces</p> <p>7</p>
 <p>COURSE Signing in &amp; Setting Up</p> <p>14</p>	 <p>COURSE Using Channels</p> <p>6</p>	 <p>COURSE Private Messaging &amp; Communication Tools</p> <p>9</p>	 <p>COURSE Creating, Finding &amp; Sharing Information</p> <p>4</p>	 <p>COURSE Configuring Slack</p> <p>23</p>
 <p>COURSE Using the iOS App</p> <p>3</p>				

## Bookshelf

### Bookshelf ⓘ Optional

<p>BOOK DevOps for Dummies 18</p>	<p>BOOK Achieving DevOps: A Novel About Delivering the Best o... 1</p>	<p>BOOK DevOps for the Modern Enterprise: Winning... 2</p>	<p>BOOK The Phoenix Project: A Novel about IT, DevOps, and... 77</p>	<p>BOOK Beyond The Phoenix Project: The Origins and Evolution o... 18</p>
<p>AUDIOBOOK The Phoenix Project: A Novel about IT, DevOps, and... 44</p>	<p>BOOK Enterprise DevOps Framework: Transforming I... 2</p>	<p>BOOK Generic Pipelines Using Docker: The DevOps Guide... 1</p>	<p>BOOK Ansible: From Beginner to Pro 51</p>	<p>BOOK Accelerate: The Science of Lean Software and DevOps... 20</p>
<p>AUDIOBOOK Accelerate: The Science of Lean Software and DevOps... 7</p>	<p>BOOK Docker in Production: Lessons from the Trenches 4</p>	<p>BOOK The Kitty Hawk Venture: A Novel About Continuous... 4</p>	<p>AUDIOBOOK DevOps for the Modern Enterprise: Winning... 4</p>	<p>BOOK The DevOps Adoption Playbook: A Guide to... 38</p>
<p>BOOK Reinventing ITIL in the Age of DevOps: Innovative... 9</p>				

FOLLOW US ON:



[www.skilltech.pl](http://www.skilltech.pl)

email: [biuro@skilltech.pl](mailto:biuro@skilltech.pl)

tel. +48 22 44 88 827

**SkillTech**  
Technology hired for excellence