



# DataOps Certified Professional



## About DevOpsSchool

DevOpsSchool is a unit of "Cotocus PVT Ltd" and a leading platform which helps IT organizations and professionals to learn all the emerging technologies and trend which helps them to learn and embrace all the skills, intelligence, innovation and transformation which requires to achieve the end result, quickly and efficiently. We provide over 40 specialized programs on DevOps, Cloud, Containers, Security, AI, ML and on Big data that are focused on industry requirement and each curriculum is developed and delivered by leading experts in each domain and aligned with the industry standards.

## About Course

DataOps can yield an order of magnitude improvement in quality and cycle time when data teams utilize new tools and methodologies. DevOps optimizes the software development pipeline.

DataOps is a collaborative data management practice focused on improving the communication, integration and automation of data flows between data managers and data consumers across an organization. The goal of DataOps is to deliver value faster by creating predictable delivery and change management of data, data models and related artifacts.

DataOps uses technology to automate the design, deployment and management of data delivery with appropriate levels of governance, and it uses metadata to improve the usability and value of data in a dynamic environment.



Co-ordinator – Akanksha

Call/WhatsApp: - +91 1800 889 7977

Mail Address: - [contact@DevOpsSchool.com](mailto:contact@DevOpsSchool.com)

Secondary contact – Patrick

Call/WhatsApp: - +91 7004 215 841

Mail Address: - [contact@DevOpsSchool.com](mailto:contact@DevOpsSchool.com)

<b>Duration</b>	<b>72 Hrs</b>
<b>Mode</b>	Online (Instructor-led, live & Interactive)
<b>Projects (Real time scenario based)</b>	<b>1</b>

FEATURES	DEVOPSSCHOOL	OTHERS
Lifetime Technical Support	✓	✗
Lifetime LMS access	✓	✗
Top 30 Tools	✓	✗
Interview KIT (Q&A)	✓	✗
Training Notes	✓	✗
Step by Step Web Based Tutorials	✓	✗
Training Slides	✓	✗
Training + Additional Videos	✓	✗



## Training

---

Upon completion of this program you will get 360-degree understanding of DataOps methodology. This course will give you thorough learning experience in terms of understanding the concepts, mastering them thoroughly and applying them in real work environment.

## Projects

---

You will be given industry level real time projects to work on and it will help you to differentiate yourself with multi-platform fluency, and have real-world experience with the most important tools and platforms.

## Interview

---

As part of this, you would be given complete interview preparations kit, set to be ready for the DataOps role. This kit has been crafted by 200+ years industry experience and the experiences of nearly 10000 DevOpsSchool's DevOps learners worldwide.



## Agenda/Course outline of the DataOps Training

### Models & Architecture - DataOps Concept and Foundation

- Introduction to DataOps: definition, history, and evolution of data management.
- Key principles of DataOps: agility, collaboration, automation, and feedback loops.
- Comparison of DataOps, DevOps, and Agile methodologies.
- Challenges of traditional data management approaches and benefits of DataOps.
- DataOps roles and responsibilities: data engineers, data scientists, analysts, and operators.
- DataOps tools and technologies: data integration, ETL/ELT, data virtualization, and data quality.
- Introduction to DataOps platforms: Databricks, Dataiku, StreamSets, and others.
- DataOps automation and orchestration: using Airflow, Kubernetes, and other tools.
- Continuous Integration and Continuous Deployment (CI/CD) in data environments.
- Infrastructure as Code (IaC) and Configuration Management applied to data systems.
- Data Quality and Testing: data validation, profiling, and cleansing.
- Monitoring and Observability in DataOps: metrics, alerts, and dashboards.
- Security and Compliance considerations in DataOps: data protection, access control, and auditing.
- Data Governance and Metadata Management: data catalog, lineage, and ownership.
- Best practices for DataOps: iterative development, version control, and collaboration.
- DataOps in cloud-based environments: AWS, Azure, Google Cloud, and hybrid scenarios.
- Use cases and case studies: real-world examples of successful DataOps implementations.
- Future trends in DataOps: machine learning, data streaming, and edge computing.

### Platform - Operating Systems - Centos/Ubuntu & VirtualBox & Vagrant

#### Ubuntu

- Installing CentOS7 and Ubuntu
- Accessing Servers with SSH
- Working at the Command Line
- Reading Files
- Using the vi Text Editor
- Piping and Redirection
- Archiving Files
- Accessing Command Line Help
- Understanding File Permissions
- Accessing the Root Account
- Using Screen and Script
- Overview of Hypervisor
- Introduction of VirtualBox
- Install VirtualBox and Creating CentOS7 and Ubuntu Vms

## Vagrant

- Understanding Vagrant
- Basic Vagrant Workflow
- Advance Vagrant Workflow
- Working with Vagrant VMs
- The Vagrantfile
- Installing Nginx
- Provisioning
- Networking
- Sharing and Versioning Web Site Files
- Vagrant Share
- Vagrant Status
- Sharing and Versioning Nginx Config Files
- Configuring Synced Folders

## Platform - Cloud - AWS - 4 hrs

- Introduction of AWS
- Understanding AWS infrastructure
- Understanding AWS Free Tier
- IAM: Understanding IAM Concepts
- IAM: A Walkthrough IAM
- IAM: Demo & Lab
- Computing:EC2: Understanding EC2 Concepts
- Computing:EC2: A Walkthrough EC2
- Computing:EC2: Demo & Lab
- Storage:EBS: Understanding EBS Concepts
- Storage:EBS: A Walkthrough EBS
- Storage:EBS: Demo & Lab
- Storage:S3: Understanding S3 Concepts
- Storage:S3: A Walkthrough S3
- Storage:S3: Demo & Lab
- Storage:EFS: Understanding EFS Concepts
- Storage:EFS: A Walkthrough EFS
- Storage:EFS: Demo & Lab
- Database:RDS: Understanding RDS MySql Concepts
- Database:RDS: A Walkthrough RDS MySql
- Database:RDS: Demo & Lab
- ELB: Elastic Load Balancer Concepts
- ELB: Elastic Load Balancer Implementation
- ELB: Elastic Load Balancer: Demo & Lab
- Networking:VPC: Understanding VPC Concepts
- Networking:VPC: Understanding VPC components



- Networking:VPC: Demo & Lab

## Platform - Containers - Docker - 4 hrs

---

- What is Containerization?
- Why Containerization?
- How Docker is good fit for Containerization?
- How Docker works?
- Docker Architecture
- Docker Installations & Configurations
- Docker Components
- Docker Engine
- Docker Image
- Docker Containers
- Docker Registry
- Docker Basic Workflow
- Managing Docker Containers
- Creating our First Image
- Understanding Docker Images
- Creating Images using Dockerfile
- Managing Docker Images
- Using Docker Hub registry
- Docker Networking
- Docker Volumes
- Deepdive into Docker Images
- Deepdive into Dockerfile
- Deepdive into Docker Containers
- Deepdive into Docker Networks
- Deepdive into Docker Volumes
- Deepdive into Docker Volume
- Deepdive into Docker CPU and RAM allocations
- Deepdive into Docker Config
- Docker Compose Overview
- Install & Configure Compose
- Understanding Docker Compose Workflow
- Understanding Docker Compose Services
- Writing Docker Compose Yaml file
- Using Docker Compose Commands
- Docker Compose with Java Stake
- Docker Compose with Rails Stake
- Docker Compose with PHP Stake
- Docker Compose with Nodejs Stake



## Planning and Designing - Jira & Confluence (2 + 2 = 4 hrs)

### JIRA

- Overview of Jira
- Use cases of Jira
- Architecture of Jira
- Installation and Configuration of Jira in Linux
- Installation and Configuration of Jira in Windows
- Jira Terminologies
- Understanding Types of Jira Projects
- Working with Projects
- Working with Jira Issues
- Adding Project Components and Versions
- Use Subtasks to Better Manage and Structure Your Issues
- Link Issues to Other Resources
- Working in an Agile project
- Working with Issues Types by Adding/Editing/Deleting
- Working with Custom Fields by Adding/Editing/Deleting
- Working with Screens by Adding/Editing/Deleting
- Searching and Filtering Issues
- Working with Workflow basic
- Introduction of Jira Plugins and Addons.
- Jira Integration with Github

### Confluence

- Exploring Confluence benefits and resources
- Configuring Confluence
- Navigating the dashboard, spaces, and pages
- Creating users and groups
- Creating pages from templates and blueprints
- Importing, updating, and removing content
- Giving content feedback
- Watching pages, spaces, and blogs
- Managing tasks and notifications
- Backing up and restoring a site
- Admin tasks
  - Add/Edit/Delete new users
  - Adding group and setting permissions
  - Managing user permissions
  - Managing addons or plugins
  - Customizing confluence site



- Installing Confluence
  - Evaluation options for Confluence
  - Supported platforms
  - Installing Confluence on Windows
  - Activating Confluence trial license
  - Finalizing Confluence Installation

## Backend Programming Language 2 - Python/Flask with MySQL DB - 4 hrs.

- Planning - Discuss some of the Small Project Requirement which include Login/Registration with Some Students records CRUD operations.
- Design a Method --> Classes -> Interface using Core Python
- Fundamental of Core Python with Hello-world Program with Method --> Classes
- Coding in Flask using HTML - CSS - JS - MySQL
- Fundamental of Flask Tutorial of Hello-World APp
- UT - 2 Sample unit Testing using Python test
  - Package a Python App
  - AT - 2 Sample unit testing using Selenium

Deploy to Some env.

-----END OF THE CLASS -----

Technology Demonstration

- Software Planning and Designing using JAVA
- Core Python
- Flask
- MySQL
- pytest
- Selenium
- HTML
- CSS
- JS.

## Source Code Versioning - Git using Github - 4 hrs

- Introduction of Git
- Installing Git
- Configuring Git
- Git Concepts and Architecture
- How Git works?
- The Git workflow
- Working with Files in Git
  - Adding files
  - Editing files
  - Viewing changes with diff
  - Viewing only staged changes
  - Deleting files
  - Moving and renaming files
  - Making Changes to Files
- Undoing Changes
  - Reset
  - Revert
- Amending commits
- Ignoring Files
- Branching and Merging using Git
- Working with Conflict Resolution
- Comparing commits, branches and workspace
- Working with Remote Git repo using Github
- Push - Pull - Fetch using Github
- Tagging with Git

# Container Orchestration - Kubernetes & Helm Introduction - 4 hrs

## Kubernetes

- Understanding the Need of Kubernetes
- Understanding Kubernetes Architecture
- Understanding Kubernetes Concepts
- Kubernetes and Microservices
- Understanding Kubernetes Masters and its Component
  - kube-apiserver
  - etcd
  - kube-scheduler
  - kube-controller-manager
- Understanding Kubernetes Nodes and its Component
  - kubelet
  - kube-proxy
  - Container Runtime
- Understanding Kubernetes Addons
  - DNS
  - Web UI (Dashboard)
  - Container Resource Monitoring
  - Cluster-level Logging
- Understand Kubernetes Terminology
- Kubernetes Pod Overview
- Kubernetes ReplicationController Overview
- Kubernetes Deployment Overview
- Kubernetes Service Overview
- Understanding Kubernetes running environment options
- Working with first Pods
- Working with first ReplicationController
- Working with first Deployment
- Working with first Services
- Introducing Helm
  - Basic working with Helm

## Database - Mysql

- Introduction to MySQL: history, features, and benefits.
- Understanding Relational Databases: data modeling, normalization, and relational algebra.
- Installation and Configuration of MySQL: setting up a local or remote server.
- MySQL Workbench: an overview of the graphical user interface (GUI) and its features.
- MySQL Data Types: numeric, string, date/time, and spatial data types.
- Creating Tables in MySQL: defining columns, constraints, and indexes.
- Inserting Data into MySQL: using SQL INSERT statements and other tools.
- Retrieving Data from MySQL: using SQL SELECT statements, filtering, and sorting data.
- Updating and Deleting Data in MySQL: using SQL UPDATE and DELETE statements.
- Joins and Subqueries in MySQL: combining data from multiple tables and queries.
- Grouping and Aggregating Data in MySQL: using SQL GROUP BY and aggregate functions.
- Data Manipulation and Transactions in MySQL: locking, rollback, and commit operations.
- MySQL Security: user authentication, authorization, and encryption.
- Backup and Recovery in MySQL: backup strategies, tools, and techniques.
- Advanced MySQL Features: stored procedures, triggers, and views.
- MySQL Optimization and Performance Tuning: indexing, query optimization, and configuration.
- MySQL and Web Development: using PHP, Python, or other programming languages with MySQL.
- MySQL and Big Data: integrating MySQL with Hadoop, Spark, and other big data platforms.

## Database - Postgresql

- Introduction to PostgreSQL: history, features, and benefits.
- Relational Database Concepts: data modeling, normalization, and SQL basics.
- Installation and Configuration of PostgreSQL: setting up a local or remote server.
- PostgreSQL Tools: an overview of the command-line interface (CLI) and graphical user interface (GUI).
- PostgreSQL Data Types: numeric, string, date/time, and spatial data types.
- Creating Tables in PostgreSQL: defining columns, constraints, and indexes.
- Inserting Data into PostgreSQL: using SQL INSERT statements and other tools.
- Retrieving Data from PostgreSQL: using SQL SELECT statements, filtering, and sorting data.
- Updating and Deleting Data in PostgreSQL: using SQL UPDATE and DELETE statements.
- Joins and Subqueries in PostgreSQL: combining data from multiple tables and queries.
- Grouping and Aggregating Data in PostgreSQL: using SQL GROUP BY and aggregate functions.
- Data Manipulation and Transactions in PostgreSQL: locking, rollback, and commit operations.

- PostgreSQL Security: user authentication, authorization, and encryption.
- Backup and Recovery in PostgreSQL: backup strategies, tools, and techniques.
- Advanced PostgreSQL Features: stored procedures, triggers, and views.
- PostgreSQL Optimization and Performance Tuning: indexing, query optimization, and configuration.
- PostgreSQL and Web Development: using PHP, Python, or other programming languages with PostgreSQL.
- PostgreSQL and Big Data: integrating PostgreSQL with Hadoop, Spark, and other big data platforms.
- PostgreSQL Replication: high availability and disaster recovery solutions.
- PostgreSQL Extensions: advanced features and functionalities.

## Data Analytics Engine - Apache Spark

- Introduction to Apache Spark: history, features, and benefits.
- Overview of Data Analytics: understanding data analytics concepts and workflows.
- Installing and Configuring Apache Spark: setting up a local or remote cluster.
- Apache Spark Architecture: understanding the building blocks of Spark.
- Resilient Distributed Datasets (RDDs): the fundamental data structure in Spark.
- DataFrames and Datasets: high-level APIs for structured data processing in Spark.
- Spark SQL: processing structured data using SQL queries in Spark.
- Spark Streaming: processing real-time data streams in Spark.
- Machine Learning with Spark: building and evaluating machine learning models in Spark.
- Graph Processing with Spark: analyzing and visualizing graph data using Spark.
- Advanced Spark Concepts: caching, partitioning, and serialization in Spark.
- Spark Performance Tuning: optimizing Spark jobs for faster processing and better resource utilization.
- Spark Ecosystem: an overview of Spark's integration with other big data technologies.
- Data Visualization in Spark: creating visualizations of Spark data using third-party libraries.
- Spark and Cloud Computing: deploying and managing Spark on cloud platforms like AWS and Azure.
- Real-world Applications of Spark: use cases and case studies of Spark in various industries.

## Reporting - Grafana

- Introduction to Grafana: history, features, and benefits.
- Overview of Data Visualization: understanding data visualization concepts and workflows.
- Installing and Configuring Grafana: setting up a local or remote server.
- Grafana Architecture: understanding the building blocks of Grafana.
- Data Sources in Grafana: connecting to various data sources like Prometheus, InfluxDB, Elasticsearch, and others.
- Creating and Customizing Dashboards in Grafana: building dashboards using various visualization panels and options.
- Templating in Grafana: creating dynamic dashboards using template variables.
- Alerts in Grafana: setting up alerts for monitoring and notification purposes.
- Annotations in Grafana: adding and managing annotations for better context in dashboards.
- Plugins in Grafana: extending the functionality of Grafana using plugins.
- Grafana API: using Grafana APIs for automating tasks and integrating with other systems.
- Security in Grafana: managing user authentication, authorization, and permissions in Grafana.
- Advanced Grafana Concepts: dashboard version control, backup and restore, and Grafana cloud.
- Integrations with Grafana: integrating Grafana with other tools like Prometheus, InfluxDB, and Elasticsearch.
- Best Practices for Grafana: tips and tricks for better performance, usability, and maintainability.

## Reporting - Grafana

### Apache NiFi:

- NiFi is an open-source data integration tool that can be used to ingest data from various sources, transform it, and load it into Apache Kafka.

### StreamSets Data Collector:

- StreamSets Data Collector is an ETL tool that provides a drag-and-drop interface to ingest data from multiple sources, transform it, and load it into Apache Kafka.

### Confluent Kafka Connect:

- Kafka Connect is a component of the Confluent Platform that provides a framework for building and running connectors between external systems and Apache Kafka.

### Talend:

Talend is a commercial ETL tool that provides connectors for Apache Kafka, allowing users to ingest data from various sources, transform it, and load it into Kafka.

- Apache Oozie:
  - Oozie is a workflow scheduler system used to manage Hadoop jobs and coordinate dependencies between them. It provides a web-based interface and supports job scheduling, job control, and job coordination.

## Bigdata - Apache Hadoop

- Lets understand Continuous Integration
- What is Continuous Integration
- Benefits of Continuous Integration
- What is Continuous Delivery
- What is Continuous Deployment
- Continuous Integration Tools
  
- What is Jenkins
- History of Jenkins
- Jenkins Architecture
- Jenkins Vs Jenkins Enterprise
- Jenkins Installation and Configurations
  
- Jenkins Dashboard Tour
- Understand Freestyle Project
- Freestyle General Tab
- Freestyle Source Code Management Tab
- Freestyle Build Triggers Tab
- Freestyle Build Environment
- Freestyle Build
- Freestyle Post-build Actions
- Manage Jenkins
- My Views
- Credentials
- People
- Build History
- Creating a Simple Job
- Simple Java and Maven Based Application
- Simple Java and Gradle Based Application
- Simple DOTNET and MSBuild Based Application
  
- Jobs Scheduling in Jenkins
- Manually Building
- Build Trigger based on fixed schedule
- Build Trigger by script
- Build Trigger Based on pushed to git
- Useful Jobs Configuration
- Jenkins Jobs parameterised
- Execute concurrent builds

- Jobs Executors
- Build Other Projects
- Build after other projects are built
- Throttle Builds
  
- Jenkins Plugins
- Installing a Plugin
- Plugin Configuration
- Updating a Plugin
- Plugin Wiki
- Top 20 Useful Jenkins Plugins
- Using Jenkins Plugins Best Practices
- Jenkins Node Management
- Adding a Linux Node
- Adding a Windows Nodes
- Nodes Management using Jenkins
- Jenkins Nodes High Availability
  
- Jenkins Integration with other tools
- Jira
- Git
- SonarQube
- Maven
- Junit
- Ansible
- Docker
- AWS
- Jacoco
- Coverity
- Selenium
- Gradle
  
- Reports in Jenkins
- Junit Report
- SonarQube Reports
- Jacoco Reports
- Coverity Reports
- Selenium Reports
- Test Results
- Cucumber Reports
  
- Jenkins Node Management
- Adding a Linux Node
- Adding a Windows Nodes
- Nodes Management using Jenkins



- Jenkins Nodes High Availability
- Notification & Feedback in Jenkins
- CI Build Pipeline & Dashboard
- Email Notification
- Advance Email Notification
- Slack Notification
- Jenkins Advance - Administrator
- Security in Jenkins
- Authorization in Jenkins
- Authentication in Jenkins
- Managing folder/subfolder
- Jenkins Upgrade
- Jenkins Backup
- Jenkins Restore
- Jenkins Command Line

## Big Data Tools for Visualization - Microsoft PowerBI

- Connect to Big Data Sources:
  - Connect Power BI to your big data sources, such as Hadoop, Spark, SQL Server, or Azure Data Lake Storage. Use Power BI's built-in connectors or install custom connectors to connect to other data sources.
- Load and Transform Big Data:
  - Load your big data into Power BI using Power Query, Power Pivot, or the Power BI Desktop. Transform your data using Power Query to clean, reshape, and combine data from multiple sources.
- Create Visualizations:
  - Use Power BI's visualization tools to create charts, graphs, tables, and maps that represent your big data. Use interactive filters, slicers, and drill-downs to explore and analyze your data.
- Apply Advanced Analytics:
  - Apply advanced analytics techniques to your big data using Power BI's machine learning algorithms, such as clustering, forecasting, and sentiment analysis. Use Power BI's R and Python integration to write custom scripts and perform advanced analytics.
- Collaborate and Share:
  - Collaborate with your team and share your big data insights using Power BI's sharing and collaboration features. Publish your reports and dashboards to the Power BI Service and share them with your team or embed them in your applications.

- Monitor and Optimize Performance:
  - Monitor and optimize the performance of your big data visualizations using Power BI's performance monitoring and optimization tools. Use the Query Diagnostics feature to identify slow queries and optimize them for better performance.
- Explore Big Data with AI:
  - Explore your big data using Power BI's AI capabilities, such as the AI-powered visuals, which use machine learning to uncover insights and patterns in your data.
- Secure Your Big Data:
  - Secure your big data using Power BI's security and compliance features. Use role-based access control, data protection, and auditing to ensure that your data is secure and compliant with regulations.
- Scale Your Big Data Visualizations:
  - Scale your big data visualizations to handle large volumes of data using Power BI's performance optimization and scalability features. Use features such as DirectQuery and Azure Analysis Services to handle large volumes of data.
- Extend Power BI with Custom Solutions:
  - Extend Power BI with custom solutions using the Power BI API and the Power BI Embedded SDK. Use these tools to create custom visualizations, embed Power BI in your applications, and integrate with other systems and data sources.

## Big Data Tools for Visualization - Tableau

---

- Connect to Big Data Sources:
  - Connect Tableau to your big data sources, such as Hadoop, Spark, SQL Server, or Amazon Redshift. Use Tableau's built-in connectors or install custom connectors to connect to other data sources.
- Load and Transform Big Data:
  - Load your big data into Tableau using Tableau's data connectors or the Tableau Desktop. Transform your data using Tableau Prep to clean, reshape, and combine data from multiple sources.
- Create Visualizations:
  - Use Tableau's visualization tools to create charts, graphs, tables, and maps that represent your big data. Use interactive filters, slicers, and drill-downs to explore and analyze your data.

- Apply Advanced Analytics:
  - Apply advanced analytics techniques to your big data using Tableau's machine learning algorithms, such as clustering, forecasting, and sentiment analysis. Use Tableau's R and Python integration to write custom scripts and perform advanced analytics.
- Collaborate and Share:
  - Collaborate with your team and share your big data insights using Tableau's sharing and collaboration features. Publish your reports and dashboards to Tableau Server or Tableau Online and share them with your team or embed them in your applications.
- Monitor and Optimize Performance:
  - Monitor and optimize the performance of your big data visualizations using Tableau's performance monitoring and optimization tools. Use Tableau's Query Performance Optimization feature to identify slow queries and optimize them for better performance.
- Explore Big Data with AI:
  - Explore your big data using Tableau's AI capabilities, such as Ask Data, which uses natural language processing to allow users to ask questions and receive visualizations.
- Secure Your Big Data:
  - Secure your big data using Tableau's security and compliance features. Use Tableau's role-based access control, data protection, and auditing to ensure that your data is secure and compliant with regulations.
- Scale Your Big Data Visualizations:
  - Scale your big data visualizations to handle large volumes of data using Tableau's performance optimization and scalability features. Use features such as Tableau Data Engine, Tableau Data Extracts, and Tableau Hyper to handle large volumes of data.
- Extend Tableau with Custom Solutions:
  - Extend Tableau with custom solutions using Tableau's APIs and SDKs. Use these tools to create custom visualizations, embed Tableau in your applications, and integrate with other systems and data sources.



# Thank you!

Connect with us for more info

Call/WhatsApp: - +91 700 483 5930 Mail: -  
[contact@DevOpsSchool.com](mailto:contact@DevOpsSchool.com)  
[www.DevOpsSchool.com](http://www.DevOpsSchool.com)