DevOps

Experience with

* Infrastructure Automation Tools – Chef, Puppet, Ansible, SaltStack/Windows Powershell
* Web Languages – Ruby/Python/PHP/Java

1. What is the difference between Agile & DevOps
2. What is the need for DevOps?

# DevOps Tools

Git: Version Control

Jenkins: Continuous Integration

|  |  |
| --- | --- |
| **Usage** | **Tools** |
| Version Control | **GIT**, Subversion, AWS CodeCommit |
| Unit Testing | **JUnit** |
| Code Review | Codacy, Codebeat, Codeclimate, Scrutinizer |
| Static Analysis | **Checkstyle, FindBugs, PMD, FindSecurityBugs, SONAR** |
| Build | **Maven**, Gradle |
| Continuous Integration | **Jenkins**, Travis CI, Atlassian Bamboo |
| Continuous Testing | **Selenium**, TestComplete, TestingWhiz  QTP, Appium, Robot, Cucumber, Nose |
| Continuous Deployment | **Jenkins, GoCD, Codar, AWS Code Deploy** |
| Infrastructure Automation | **Puppet, Chef, Ansible** |
| Continuous Monitoring | **Nagios, Splunk, ELK (Elastic Search, Logstash, Kibana)** |
| Containerization | **Docker, Kubernetes Docker Orchastration** |
| Virtualization | OpenStack, VMWare, SoftLayer, Vagrant |
| Performance Testing | **JMeter**, Grinder, Tsung, WebLoad, BlazeMeter, NeoLoad, LoadRunner, Loadster, |
| Security Testing |  |
| Resilience Testing | Hystrix, ChaosMonkey |
| UI Testing |  |

# Devops Course – Edureka

## Overview of DevOps

Why DevOps?

What is DevOps?

DevOps Market Trends

DevOps Engineer Skills

DevOps Delivery Pipeline

DevOps Ecosystem

DevOps Case Studies –

## Version Control with GIT

What is Version Control?

What is GIT?

Why GIT for your organization?

Install GIT

Common commands in GIT

Working with Remote Repositories

Use Cases of Real time using GIT

Branching and Merging in GIT

GIT Workflows – Branching, Merging, Stashing, Rebasing, Reverting, Resetting

GIT Cheatsheet

## Continuous Integration (using Jenkins)

What is CI?

Why is CI required?

Introduction to Jenkins

Introduction to Maven

Jenkins Management

Adding a slave node to Jenkins

Building a Delivery Pipeline

Pipeline as a Code

Implementation of Jenkins in a Project

Build pipeline of Jobs using Jenkins

Create a pipeline script to deploy an application over tomcat server

## Continuous Testing with Selenium

What is Continuous Testing?

What are popular Continuous Testing Tools?

Characteristics of CT Tools and when to use which one

Introduction to Selenium

Why Selenium?

Selenium – Webdriver

Creating Test cases in Selenium Webdriver (Walts)

What and why X-Path

Handling different controls on WebPage

Framework in Selenium

Selenium Integration with Jenkins

Installing Selenium

Creating Test Cases using Selenium WebDriver

## Containerization with Docker: Ecosystem and Networking

Introduction to Docker Ecosystem

Docker Compose

Docker Swarm

Managing Containers

Running Containers

Introduction to Docker Networking

Network Types

Docker Container Networking

Use Docker Compose to create a Word press site

Start Containers on a Cluster with Docker Swarm

Deploy a Multi-tier application over a cluster

Scale an application

## Containerization using Kubernetes

Containers and Container Orchestration

Introduction to Kubernetes

Revisiting Kubernetes Cluster Architecture

Spinning up a Kubernetes Cluster on Ubuntu VMs

Exploring your Cluster

Understanding YAML

Creating a Deployment in Kubernetes using YAML

Creating a Service in Kubernetes

Installing Kubernetes Dashboard

Deploying an app using Dashboard

Using Rolling Updates in Kubernetes

## Continuous Deployment: Configuration Management with Puppet

Introduction to Puppet

Puppet Installation

Puppet Configuration

Puppet Master and Agent Setup

Node Classification

Puppet Environment

Puppet Classes

Automation and Reporting

Configure and Implement servers using Puppet

## Configuration Management with Ansible

What is Infrastructure as Code

What is Configuration management? What are its advantages?

What are the tools for Config Management?

Introduction to Ansible

Ansible Installation

Configuring Ansible Roles

Write Playbooks

Executing Ad-hoc Commands

## Continuous Monitoring with Nagios

Introduction to Continuous Monitoring

Introduction to Nagios

Installing Nagios

Nagios Plugins (NRPE) and Objects

Nagios Commands and Notification

Monitoring different Servers using Nagios

## Introduction to DevOps on Cloud

Why Cloud?

Introduction to Cloud Computing

Why DevOps on the Cloud

Introduction to AWS

Various AWS Services

DevOps using AWS

JAVA DevOps

1. IDE
   1. Use a good IDE – Eclipse or Jetbrains; Does JetBrains really Improve productivity? Does it warrant cost?
   2. IDE configuration
   3. IDE Keyboard Shortcuts
   4. IDE Plugins – GIT Integration, Auto-Formatting, Static Analysis (PMD, FindBugs, FindSecBugs, CheckStyle), Unit Testing (Junit), Code Review
2. Code Review
3. Unit Testing (Junit)
4. GIT
   1. Commit Code (Developer Cycle for all GIT Activities)
   2. GIT Triggers for CheckStyle, FindBugs, PMD,
5. Continuous Integration

Simplilearn – Projects

Project 1 Scenario:

A media company wishes to offer a website where users can upload photographs. Captions and titles can be added to the photographs. Thumbnails need to be created from photographs so that multiple images can be displayed on index pages. Customers can order prints of photographs on T-shirts, mugs, and other items. You will need to design, implement, and deploy part of the system.  
   
The objective of this project is to design and implement the business and integration tiers of this project. In the final system, image files and thumbnails will be stored in a file system accessible to the web server.

Goals of the project:

* Set up a MySQL database in a Docker container
* Design a database table structure for the data
* Write an SQL script to create the table and enter some sample data
* Create the table in the database
* Create a Java entity object and map it to the database table
* Create a DAO and its implementation
* Create a Junit test case and write integration tests

The deliverables are:

* SQL script
* Entity object code
* Hibernate configuration file
* DAO and implementation code
* Junit test case

Project 2 Scenario:

A library has a large CD collection which it needs to catalog. It needs to store information in a database and have a web front end. You will need to design, implement and deploy part of the system. The objective of this project is to produce a web application which implements the catalog. The information required for each CD is:

* Title
* Artist(s)
* Year of release
* Number of tracks
* Total playing time

Each CD will also have a list of tracks containing the following information:

* Title
* Author(s)
* Playing time

Goals of the project:  
  
The starting point for this project is the code and instructions for exercise 8.7. Take a copy of the DevOps project from lesson 8 and rename the files from Monitoring to Library. You are only required to produce the web front end. The data manager can generate a hard-coded list of CDs rather than extract them from the database.  
  
The deliverables are:

* The data manager Java file
* The library controller Java file
* The JSP which displays the CD list
* A screenshot of the web page showing the list of CDs

Project 3 Scenario:

There is a requirement to be able to convert an integer into words. For example, 57 needs to be converted into “fifty-seven”.The objective of this project is to develop a method which can convert the numbers 1 through 999 to words. If you have time to extend the range up to 2 billion. The method needs to be developed using Test Driven Development (TDD). A standalone application is also required which reads a number from the keyboard and prints out its value in words or “Invalid number”. The program should exit when the user enters 0.  
   
The application needs to be packaged as an executable jar file using Ant.  
  
 Goals of the project:

* Clone the repository and install Ant.
* Git clone https://github.com/simplilearn-devops/devops-project-3
* Curl -O http://apache.mindstudios.com//ant/binaries/apache-ant-1.10.0-bin.zip
* Unzip apache-ant-1.10.0-bin.zip
* Ins apache-ant-1.10.0 ant
* Start the VNC server
* On your local machine connect to the VNC server on port 5901
* Open the Numbers project in Eclipse
* Add the jar files in lib to the classpath
* Develop the numbers to words application using TDD
* Implement the main application to read numbers from the keyboard and print out the values
* On the server navigate to the Numbers project
* Run ant to build the project. The build will fail if the unit tests fail.
* ~/ant/bin/ant dist
* The results of the unit tests are in the report directory which got created
* Run the application and try it out
* java –j Numbers.jar

The deliverables are:

* The unit test java file
* The number of words Java file
* The main application Java file
* The test results file from the report directory
* A screenshot of the application output after running it with a representative set of numbers

Project 4 Scenario:

You were asked to put some downloadable materials to Amazon S3 so that your company’s customers all around the world can access them.   
A month later, during a massive marketing campaign, the marketing team realized that using S3 is expensive and asked you to find and use a more cost-effective way to distribute the files.

Goals of the Project:

* Create an S3 bucket
* Set its access permissions to allow all anonymous users to download the files from it
* Upload sample files to an S3 bucket
* Create a CloudFront distribution using an S3 bucket as its origin
* Verify that the files are accessible

Project 5 Scenario:

The Oscrop Corporation’s public-facing web app currently runs on an IIS web server at the company’s chosen ISP. Oscrop wants to migrate this web app to Azure. You must test the web app’s functionality by setting up a test Oscrop web app. An internal team will provide you with a test web app to deploy. You must ensure that they can continue to stage changes to the test web app before deploying those changes to the public-facing site.

The operations team at Oscrop currently uses a Microsoft SQL Server database to store back-end data for the web app on the company’s servers. You want to investigate the option of using Azure SQL Database to host this database.

The operations team is interested in monitoring the performance of this database in Azure.

Goals of the Project:

* Create a new web app
* Deploy a web app
* Manage web apps
* Create an Azure SQL Database
* Configure server firewall rules
* Use SQL Server Management Studio
* View database metrics