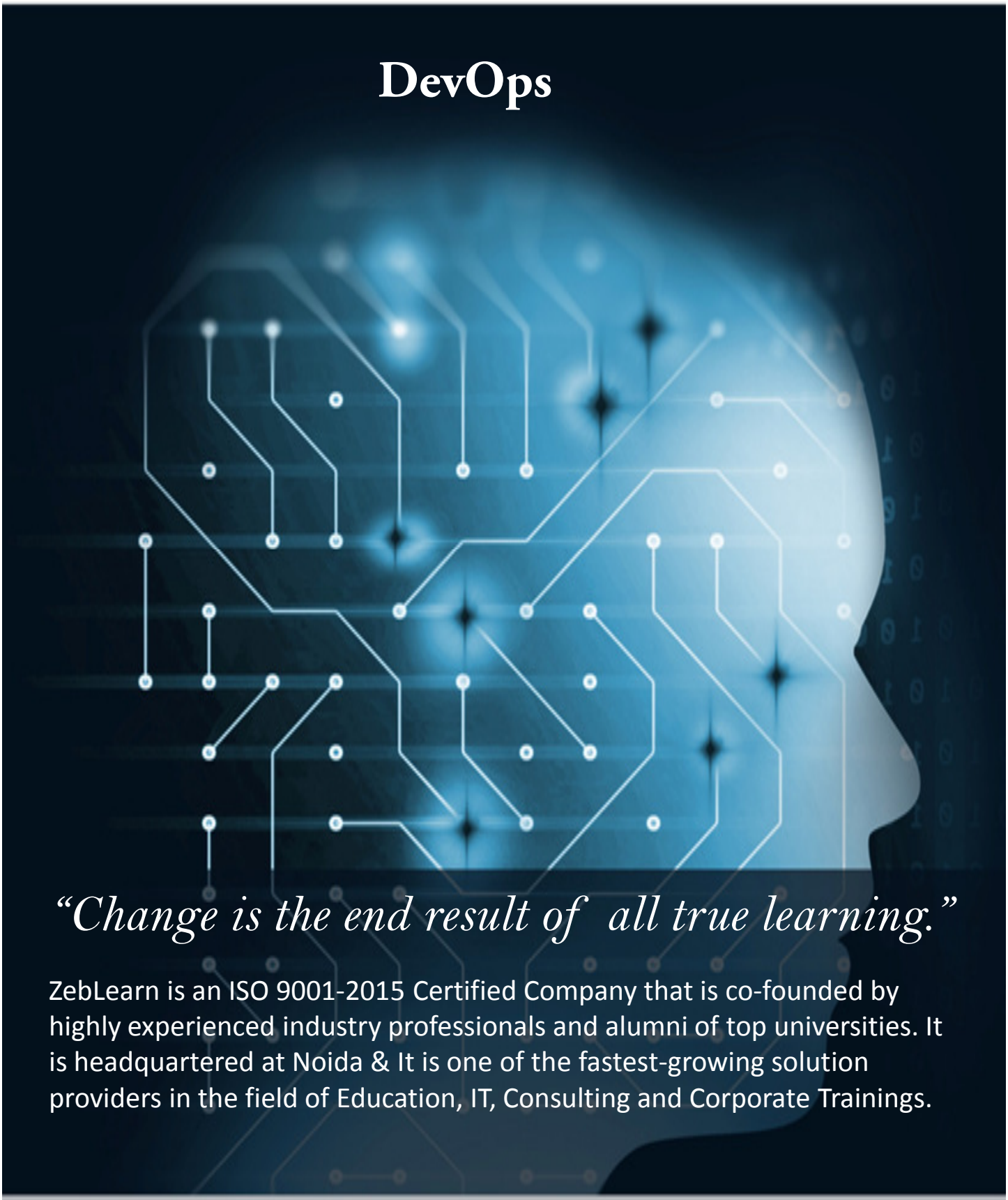
ZebLearn

DevOps



"Change is the end result of all true learning."

ZebLearn is an ISO 9001-2015 Certified Company that is co-founded by highly experienced industry professionals and alumni of top universities. It is headquartered at Noida & It is one of the fastest-growing solution providers in the field of Education, IT, Consulting and Corporate Trainings.

+91 72778 77778 info@zeblearn.com www.zeblearn.com

DevOps Syllabus

Infrastructure Setup

- EC2 Walkthrough
- Installation of DevOps Tools on cloud
  - Git
  - Docker
  - Selenium
  - Maven
  - Jenkins
  - Puppet
  - Ansible
  - Kubernetes
  - Nagios

Introduction to DevOps

- What is Software Development
- Software Development Life Cycle
- Traditional Models for SDLC
- Why DevOps?
- What is DevOps?
- DevOps Lifecycle
- DevOps Tools

Continuous Testing

- What is Continuous Testing?
- What is Maven?
- Running Test Cases on Chromium Web Driver
- What is Headless Mode?

DevOps Syllabus

Hands-on Exercise –

- Using Maven to import dependencies in Eclipse
- Implementing a headless test using Chrome WebDriver

Continuous Integration using Jenkins

- Introduction to Continuous Integration
- Jenkins Master Slave Architecture
- Understanding CI/CD Pipelines
- Creating an end to end automated CI/CD Pipeline

Hands-on Exercise –

- Creating a Jenkins Master Slave on AWS
- Installing Plug-ins in Jenkins
- Creating Jenkins Builds
- Creating Scheduled Builds
- Triggering Jobs using Git Web Hooks
- Using the Pipeline Plugin In Jenkins

Software Version Control

- What is Version Control
- Types of Version Control System
- Introduction to SVN
- Introduction to Git
- Git Lifecycle
- Common Git Commands
- Working with Branches in Git
- Merging Branches
- Resolving Merge Conflicts
- Git Workflow

DevOps Syllabus

Hands-on Exercise –

- Git Life cycle Commands
- Pushing Code to Github
- Stashing Code in git
- Creating, Deleting Git Branches
- Reverting a Push to GitHub
- Merging branches using git merge
- Merging branches using git rebase.
- Resolving merge conflicts using git merge tool

Continuous Deployment: Containerization with Docker

- Introduction to Docker
- Understanding Docker Lifecycle
- Components of Docker Ecosystem
- Common Docker Operations
- Creating a DockerHub Account
- Committing changes in a Container
- Pushing a Container Image to DockerHub
- Creating Custom Docker Images using Dockerfile

Hands-on Exercise –

- Common Docker Operations
- Creating a DockerHub Account
- Committing Changes to a Container
- Pushing container to DockerHub
- Creating Local Image Repository
- Building an Image using Dockerfile

DevOps Syllabus

Containerization with Docker: Ecosystem and Networking

- What are Docker Volumes
- Deploying a Multi-Tier Application using Docker Network
- Using Docker Compose to deploy containers
- What is Container Orchestration
- Container Orchestration Tools
- Introduction to Docker Swarm
- Deploying a 2-Node Cluster using Docker Swarm

Hands-on Exercise –

- Creating Docker Volumes
- Using Docker Compose to deploy multiple containers
- Deploying a Multi Node Cluster using Docker Swarm
- Deploying a multi-service app on Docker Swarm

Configuration Management using Puppet

- Need of Configuration Management
- Configuration Management Tools
- What is Puppet
- Puppet Architecture
- Setting up Master Slave using Puppet
- Puppet Manifests
- Puppet Modules
- Applying configuration using Puppet
- Puppet File Server

DevOps Syllabus

Containerization with Docker: Ecosystem and Networking

- What are Docker Volumes
- Deploying a Multi-Tier Application using Docker Network
- Using Docker Compose to deploy containers
- What is Container Orchestration
- Container Orchestration Tools
- Introduction to Docker Swarm
- Deploying a 2-Node Cluster using Docker Swarm

Hands-on Exercise –

- Creating Docker Volumes
- Using Docker Compose to deploy multiple containers
- Deploying a Multi Node Cluster using Docker Swarm
- Deploying a multi-service app on Docker Swarm

Configuration Management using Puppet

- Need of Configuration Management
- Configuration Management Tools
- What is Puppet
- Puppet Architecture
- Setting up Master Slave using Puppet
- Puppet Manifests
- Puppet Modules
- Applying configuration using Puppet
- Puppet File Server

DevOps Syllabus

Hands-on Exercise –

- Setting up Master Slave on AWS
- Testing Connection of nodes with Puppet
- Creating a Manifest
- Deploying Manifest on Node
- Creating a Module
- Deploying sample software on nodes using Puppet Modules and Manifests
- Implementing a File Server Module on Puppet

Configuration Management using Ansible

- What is Ansible?
- Ansible vs Puppet
- Ansible Architecture
- Setting up Master Slave using Ansible
- Ansible Playbook
- Ansible Roles
- Applying configuration using Ansible

Hands-on Exercise –

- Installing Ansible on AWS
- Creating a Playbook using YAML
- Creating an Ansible Role
- Using Roles in Playbook

DevOps Syllabus

Continuous Orchestration using Kubernetes

- Introduction to Kubernetes
- Docker Swarm vs Kubernetes
- Kubernetes Architecture
- Deploying Kubernetes using Kubeadms
- Alternate ways of deploying Kubernetes
- YAML Files
- Creating a Deployment in Kubernetes using YAML
- Services in Kubernetes
- Ingress in Kubernetes
- Case Study – Kubernetes Architecture

Hands-on Exercise –

- Setting up Kubernetes using kubeadm
- Installing Kubernetes using kops and GCK
- Creating a Deployment
- Creating Services
- Creating an Ingress
- Demonstrating the use of Ingress, services and deployments together

Continuous Monitoring using Nagios

- What is Continuous Monitoring
- Introduction to Nagios
- Nagios Architecture
- Monitoring Services in Nagios
- What are NRPE Plugins
- Monitoring System Info using NRPE plugins

DevOps Syllabus

Hands-on Exercise –

- Installing Nagios
- Monitoring of different servers using Nagios

Terraform Modules & Workspaces

- What is Infrastructure as a code
- Iac vs Configuration Management
- Introduction to Terraform
- Installing Terraform on AWS
- Basic Operations in terraform
- (init, plan, apply, destroy)
- Terraform Code Basics
- Deploying and end-to-end architecture on AWS using Terraform

Thanks you

Now or Never