DevOps Training Course Content

70 days of Live Sessions

Trainings will be conducted 6 days a week and each session is **1.5hr** of time.

Linux OS Basics & Admin

In this section we cover Linux Basics with Administration on CentOS7 OS.

- **Introduction to Linux OS**, Command Syntaxes, Hardware and Software information review.
- **Files**: Manage files like list, create, delete, copy & rename files
- **Directories**: Directory Structure, Navigate Directories, Manage Directories like list, create, delete, copy & rename dir and Move Dir & Files.
- **Editor**: VIM editor, Discuss three modes which are ESC, INSERT & COLON Modes.
- **Utilities**: Commands like finding files, Download files from internet & **Pipes**.
- **Administration Topics**: Process Management, User Management with SUDO, Package Management, Service Management, Permission Management & Network informative commands.
- **Project Setup:** Using the knowledge of Linux, we will setup the project end to end manually with the given manual steps using Linux VMs in **AWS Cloud.**
- **AWS Cloud:** EC2 instance creation along with Launch templates & Spot Instance types, Route53 Domain Creation, SSL Certificates will be discussed.
- **Nginx**: How a reverse proxy works and SSL certs works will be discussed.

CentOS

Git Repos & Shell Scripting

In this section we cover how to deal with code with Git repos. Ideally our scope of work is not the manual effort, So the effort we did in last section for project needs to be automated, So we discuss the topics of shell scripting followed with automating the effort of shell scripting.

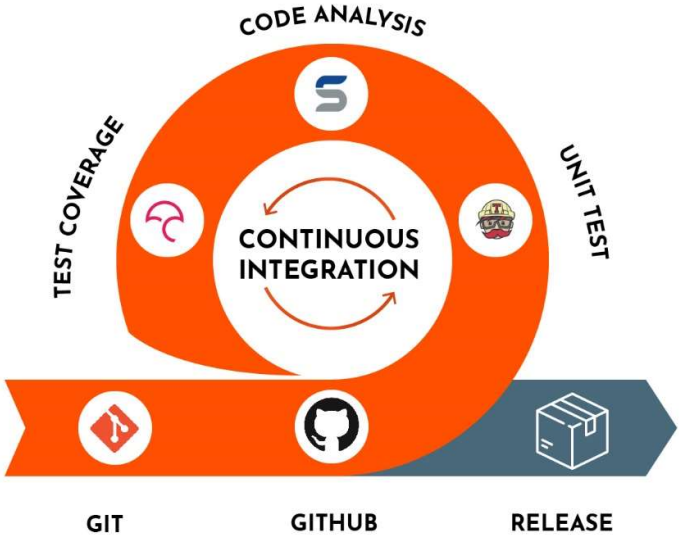
- **Git Repo:** Intro to GitHub & Azure Repos, Create account in GitHub & AZD, Create repos clone them in desktop and push the changes manually & using editor.
- **Shell Scripting:** Types of Shells, She-Bang, Comments, Printing Text, Variables, User Inputs, Special Variables, Redirectors, Exit Status, Quotes, Functions, Conditions, Loops.
- **Project automation with Shell Scripting:** We will start implementing the above topics with in our project and we will make automating the manual effort.
- **Makefile**: We will discuss about using Makefile and start building Makefiles to our shell scripting project.

Jenkins

Continuous Integration

CI is one of the practice in the modern development. It helps developers to identify the issues in much early before releasing the code. DevOps engineers usually try to make these configurations and help developers to stabilize the process.

- **CI Pipelines:** Introduction to CI & Pipelines steps involved in general.
- **SonarQube**: Setup sonarqube application and scan the code and report it.
- **Nexus**: Use this to archive the artifacts in central place.
- **Selenium**: How to do the automated tests when there is a new application has been about to release
- **Builds**: How to make the compile the code and make the builds.
- **Release Strategy**: Different best practices and its outcomes to enhance these pipelines to Continuous Delivery and then to Continuous Deployment.
- In this strategy we discuss about branching, tagging and merging the code in Git repos.



Ansible

Shell scripting is Great, yet it has certain disadvantages to use it widely. So we need to talk about more robust systems or tools which we can automate the effort with minimum efforts. There comes the **Ansible tool as handy**. Using this tool we setup the project.

- **Configuration Management:** Introduction to CM, Evolution of CM, Advantages of CM, How Ansible is leader, Ansible Installation
- **Inventory**: Ansible Inventory and how best it can be setup as per need.
- **Ansible with SSH**: Ansible using SSH Password, then SSH Keys, Overview of LDAP or AD, Ansible Configuration file
- **Ad-Hoc**: Executing Ansible Ad-Hoc commands
- **Playbooks**: Need of Playbook, Introduction to YAML, Playbook Layout, Simple Playbook that connects to the nodes.
- **Variables**: Different ways of parsing variables.
- **Roles**: Need of Roles, Role structure, Playbooks on roles
- **Misc**: Tags, Filters, Handlers Loops, Vault.
- **Ansible Pull**: Need of pull
- **Ansible ATX:** UI Management tool for Ansible.

ANSIBLE

Amazon Web Services

In this section we are going to explore different AWS services. So far we would have discussed about setting up the project but not the best practices of setting up the project like Security, Scalability, High Availability and others. In this section we are going to discuss about those concepts and implement using the services of AWS.

- **Load Balancer:** We discuss about the need to Load Balancer and we setup the ALB & NLB and understand the need of both with scenario based.
- **ALB Rules:** There are different ways of setting up the listener rules we discuss through them.
- **AutoScaling**: We discuss the need of autoscale for any application and we discuss the strategy of Instance refresh.
- **Security**: We discuss the security aspect of application with VPC in Private subnets, Load Balancers with security enabled services like WAF & Sheild.
- **Databases**: We discuss the DB components services of AWS like RDS and ElasticCache.
- **Containers:** We discuss EKS cluster and setup application inside the Cluster.
- **CI/CD:** We will discuss the CI/CD process tools in AWS like CodeBuild, Code Deploy & Code Pipeline.
- **ElasticSearch**: We will use Managed service of ElasticSearch in AWS for aggregating the logs.
- **Prometheus**: We will use the Managed Prometheus service in AWS for monitoring the servers.

Terraform

In previous section we discussed the different services of AWS. Usually as DevOps engineer we need to automate the effort and even for cloud as well and that will be done by Terraform tool.

- **IaC:** Introduction to IaC (Infrastructure as Code) tool and need for it. What Terraform is helping us to achieve.
- **HCL**: Terraform DSL which is HCL v2 (Hashicorp Configuration Language). Introduction to it, Some basics of how it looks like.
- **Introduction**: Terraform Files, Terraform Commands.
- **Outputs**: Terraform outputs and different ways of using it with root module and sub module.
- **Variables**: Different ways of parsing variables to Terraform and its precedence and best practices. Also followed with Data types & Types of Variables in Terraform.
- **Terraform Core Resources:** Providers, Resources, Provisioners
- **Modules**: Need for modules and how it helps to keep the code DRY.
- **Logical Statements**: Terraform Loops & Conditions.
- **Functions**: In Built functions of Terraform.
- **Makefiles** for Terraform commands.

HashiCorp

Terraform

Prometheus, Grafana & Elasticsearch

Until last section we made setup our project end to end with automation, Now we will be start operating the project. While we do it then monitoring comes in a key factor and we need the metrics for understanding the performance and also reporting the ongoing issues. So **Prometheus** comes as a widely used tool to collect metrics and reporting.
Along with performance metrics we need the logs of the applications also to be collected to understand the insights of the application. Hence we start using **ELK** to collect the logs and provision dashboards to get insights.

- **Introduction**: Basics of Monitoring and how Prometheus is different than others.
- **Setup**: Prometheus Monitoring Setup, Pull Metrics, Dynamic Nodes Setup with AWS.
- **Grafana**: Collect Metrics in Grafana & Create Dashboards from Prometheus Data Source.
- **Alerts**: Configure Alerts in Grafana to report any kind of application issues.
- **Log Aggregation**: Introduction to log aggregation and Importance of it in Business.
- **Components**: Understand the architecture of ELK, Components Involved in it.
- **Logs**: Understand Structured and Unstructured logs and use components like Filebeat & Logstash to handle them.
- **Kibana**: Collect log indices and create patters in Kibana, Choose data types for the logs coming in, Dashboards & Data Visualisation in Kibana.

Continous Deployment & Strategies

So far we have application setup end to end with automation and even we have alerts to identify the issue. So it is time to look into releases of how to do the releases and strategies involved over it.

- **Rolling Updates**, Do rolling updates with end to end with Azure Pipelines & Releases
- **Blue Green Deployments,** Do Blue Green Deployments end to end with Azure Pipelines & Releases
- **Canary Deployments,** Do Canary Deployments with end to end with Azure Pipelines & Releases.

Docker and Kubernetes

Until previous section we learned how to deploy services on servers. But to make all the operation faster we better do that with Containers.. So we will be talking about Docker and make Docker Images for all the components of the projects and we deploy them to Kubernetes

- **Docker:** Introduction to Containers & Anatomy of Docker, Docker Echo System, Docker container options
- **Docker Imaging:** Making Docker Images & Setup project component Images
- **Kubernetes Introduction:** Need of an Orchestration tool and why kubernetes is leading, Echo System of Kubernetes
- **Kubernetes Resources:** Pods, Services, ReplicaSets, Deployments, Secrets, ConfigMaps
- **Rolling & Blue Green Deployments:** Setup release strategy for all the components in Kubernetes with Azure Pipelines & Azure Releases.