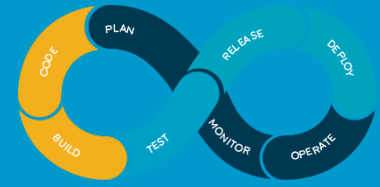# DevOps Training - Week 1

Description:

As a beginner, you would be understanding how a server environment will be and dealing with the operating systems and administering them. As per standard practice which we use in real projects we are also going to use Linux as our standard operating system to complete the course. *So, during the first week we are going to talk on creating servers in AWS environment along with some other basics in AWS Cloud followed with Linux Basics and Administration Topics.*

Topics Covered:

- Create an account in AWS
- Understanding Regions and Availability Zones in AWS
- Installing required software's in Desktop
- Setting up access to AWS cloud using SSH Keys
- Create servers in EC2 Service
- Understand Linux Command Line
- Getting Server Information using Linux Commands
- File and Directory management
- Using VI/VIM Editor
- Linux cli utilities for downloading software
- Linux Administration Topics
    - User Management
    - Package Management
    - Service Management
    - Disk Management
    - Network Management
    - File Permissions

DevOps Training - Week 2
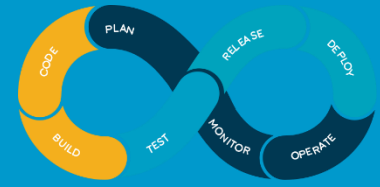
Description:

As a DevOps Engineer, you'll work collaboratively with software engineering to deploy and operate our systems. Help automate and streamline our operations and processes. *In this week, we are going to understand how the application architectures and we setup practically those applications in AWS EC2 Instances Manually. Also, we will further proceed to setup our applications in an automated way using Shell Scripting. Also, to invoke shell scripts we will start using Jenkins as an automation tool and we are going to talk few basics of Jenkins.*

Topics Covered:

- Understand different components of an application
- Understand the architecture of an application
- Installing and Configuring Web Server, Application Server and Database Servers
- Integration of Web, Application and Database servers to work as a stack
- Understanding the latest generation spring boot applications versus legacy applications
- Creation of servers in EC2 instances in and setup web and application servers
- Creation of RDS instances in AWS for application stack
- Setup Security Group firewalls to limit the database to be accessed by only Application Server.
- Introduction to Shell Scripting
- Printing messages with Shell Scripting
- Variables and Functions in Shell Scripting
- Getting Inputs from user in Shell Scripting
- Conditions and Loops in Shell Scripting
- Develop shell scripts to install all web, app and db related configs with shell scripting
- Installation of Jenkins
- Triggering the shell scripts on remote nodes using Jenkins

DevOps Training - Week 3
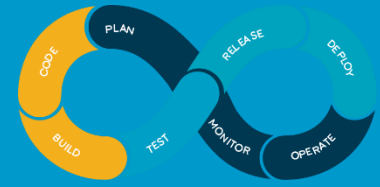
Description:

DevOps has several components that must work in unison for a team to meet its objectives. A key element, which usually serves as the center of the DevOps "machinery," is configuration management. *In this week, we are going to enhance our automation with configuration management tool Ansible and we are going to configure and setup WEB, APP and DB components on servers. We will discuss Ansible much deeper along with integration of source code management tool GIT and we track all the changes get modified to the code.*

Topics Covered:

- Introduction to VCS
- Difference between CVCS(SVN) and DVCS(GIT)
- Architecture of GIT
- Using existing GIT Repositories
- Installing GitLab Server and Create Users and delegate Repositories
- Installation of Ansible and Configuring Ansible.
- Ansible Inventory file
- YAML syntax for Ansible Playbooks
- Understanding Playbooks, Plays, Task and Modules
- Different ways of defining variables with Ansible
- Conditions and Loops in Ansible
- How TAGS are used in Ansible
- Setup Application using Ansible and update code in Git Repositories
- Run the ansible playbooks with Jenkins for automation
- Ansible Vault
- Roles in Ansible
- Ansible pull and Galaxy

DevOps Training
- Week 4
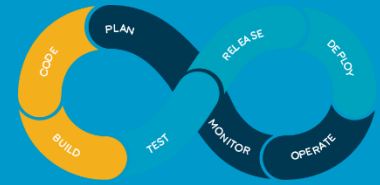
Description:

While companies look to fix their application development processes by moving from waterfall to DevOps, they also understand that DevOps alone won't save them. The latency in making capital purchases of hardware and software slows the development process, even if it's made agile. Developers end up waiting around for capital resources to be put in place before the applications can be deployed. Thus, DevOps won't have much value without the cloud, and the cloud won't have much value without DevOps. The centralized nature of cloud computing provides DevOps automation with a standard and centralized platform for testing, deployment, and production. *During this week we are going to setup our project with multiple environments and we use Elastic Beanstalk service to replicate the same thing in AWS Cloud. Along with that we also discuss about restricting the users to use particular services in AWS using IAM and manage the servers with AWS systems manager which is an alternative to Ansible in cloud. Also we will talk about basic monitoring and log monitoring using ELK.*

Topics Covered:

- Understand the importance of multiple environments for an application
- Setup multiple environment application
- Use ansible to configure and customize these environments in an automated way
- Use Jenkins to deploy the new application based on environments
- Introduction to AWS Elastic Beanstalk Service
- Setup DEV and PROD environments in Elastic Beanstalk
- Introduction to IAM
- IAM Users and Groups
- IAM Roles and Policies
- Making Custom Roles and Policies
- Introduction to AWS Systems Manager
- Deploy the application with AWS Systems Manager
- Use AWS Systems Manager as Configuration Management tool
- Setup monitoring using NewRelic
- Setup Log Monitoring using ELK, Using Elasticsearch service from AWS
- Introduction to Cloud Watch
- Monitor Performance with CloudWatch of your instances
- Using CloudWatch as an alternative to log monitoring

DevOps Training - Week 5
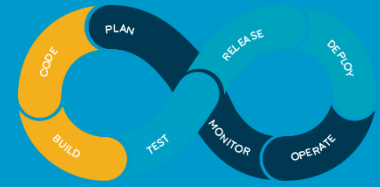
Description:

Continuous Integration and Continuous Delivery (CI/CD) are often cited as pillars of successful DevOps. To establish and optimize the CI/CD development model and reap the benefits, companies need to build an effective pipeline to automate their build, integration and testing processes. At a high level, the pipeline "includes compiling, packaging, and running basic tests prior to a code base merge. After your code is in the base, the main branch of your version control software, additional tests are run to ensure your apps work with real configuration and other services. Performance and security tests are also run at this point. From here you deploy code to staging and then to production". *In this week, we are going to cover the tools which help in making the pipelines to deliver the code to production in smooth manner. We use GIT branching strategies, We talk about build tools like Maven and Gradle, We talk about functional testing using Selenium and Python code, We use CodeQuality testing using SonarQube, We talk about Nexus artifact manager and GitLab tool a bit more. Majorly we cover Jenkins Pipeline code and Seed Jobs in Jenkins for complete automation.*

Topics Covered:

- Understanding CICD
- Best Practices of CICD
- Understanding different jobs in Jenkins
- Introduction to Jenkins Pipelines and Groovy
- Doing Jenkins automation with Seed Jobs
- Understanding GIT branching strategy
- Introduction to Maven Projects
- Understanding different Maven life cycle phases and customize them as per requirements
- Introduction to Gradle Projects and understand how to build binaries with Gradle
- Introduction Code Quality Analysis using SonarQube
- Setup SonarQube and test the development code and publish reports
- Using Selenium code for UI testing
- Writing Python scripts for API testing
- Introduction to Artifact Managers
- Setup Nexus and understand different types of repositories
- Creating repositories and limiting access to particular users to upload and download artifacts
- Setup Jenkins pipeline to include all the tools and ensure build happens automatically
- Introduction to Multibranch Pipelines
- Setup automated pipelines using Multibranch
- Deep Dive into Jenkins and all management options

DevOps Training - Week 6
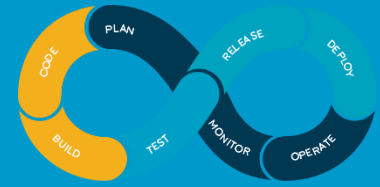
Description:

Continuous Integration and Continuous Delivery (CI/CD) techniques enable collaboration, increase agility, and deliver a high-quality product faster. It has never been easier to Integrate a Continuous Integration and Delivery pipeline without Cloud technologies. DevOps automation is becoming cloud-centric. Most public and private cloud computing providers support DevOps systemically on their platform, including continuous integration and continuous development tools. This tight integration lowers the cost associated with on-premises DevOps automation technology, and provides centralized governance and control for a sound DevOps process. Many developers who enter the process find that governance keeps them out of trouble, and it's easier to control this centrally via the cloud versus attempting to bring departments under control. *In before week we have discussed the pipeline setups using different tools and we will replace them with AWS Services in this week. We are going to use S3 buckets as artifact repository, we use Code Commit as our GIT repository, we use CodeBuild instead of Jenkins Builds, we use Code Deploy instead of Ansible, we use Code Pipeline instead of Jenkins Pipelines.*

Topics Covered:

- Introduction to Simple Storage Server (S3)
- Creating buckets using Console
- Uploading and downloading data to S3
- Building static websites using S3
- Enable version control on S3
- Getting Started with Code Commit
- Working with Repositories
- Working with Commits
- Working with Branches
- Migrate to AWS CodeCommit
- Authentication and Access Control
- Getting Started with CodeBuild
- Run AWS CodeBuild Directly
- Use AWS CodePipeline with AWS CodeBuild
- Use AWS CodeBuild with Jenkins
- Working with Build Projects and Builds
- Getting started with CodeDeploy
- Application Specification Files
- Working with the AWS CodeDeploy Agent

- Working with Instances
- Working with Deployment Configurations
- Working with Deployment Groups
- Working with Deployments
- AppSpec File Reference
- Concepts of CodePipeline
- Working with Pipelines
- Working with Actions
- Working with Stage Transitions
- Monitoring Pipelines

DevOps Training - Week 7
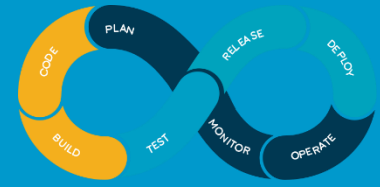
Description:

Nowadays it's critical to get your releases out fast, which requires having an automated CI/CD pipeline that takes your code from text to binaries to a deployed environment. Implementing an automated pipeline in the past has been challenging, especially when dealing with legacy applications. This is where Dockers and Kubernetes comes in. Kubernetes has revolutionized the way we deploy and manage our containerized applications. Using Helm together with Kubernetes, you gain simplified application deployment. Kubernetes is one of modern DevOps tools and component you require from infra side is declarative. Using DevOps tools, we reduced the deployments from days to hours and tools like kubernetes bring them to minutes. *In this week, we are going to discuss about containerization technologies and basics of Docker. Then we are going to talk on kubernetes in a much-detailed way and we will complete all the CICD setups with Jenkins Kubernetes integrations.*
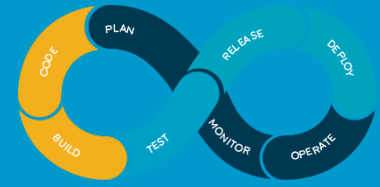
Topics Covered:

- What is Virtualization?
- What is Containerization?
- Virtualization vs Containerization
- Introduction to Docker
- Running Docker Containers
- Making Docker Images with Dockerfile and push them to Docker Registry
- Launching AWS Elastic Kubernetes Service for practice
- Get cluster details
- List all nodes associated with the cluster
- Stopping a cluster
- Deleting a cluster
- Installing & Accessing the Kubernetes dashboard
- Deploy a containerized app image in the locally setup kubernetes cluster
- List all local deployments
- Create a kubectl proxy for forwarding communication to cluster-wide private network
- Curl to verify that the app is running
- List all existing pods
- Get description of a specific pod
- View logs of the container
- Execute commands directly on the container
- Create a ephemeral volume in EBS.

- Configure Pod to store data in EBS Volumes.
- Create a new service
- Add ha-proxy to configuration file as proxy to expose the application
- Expose the service outside the cluster using ha-proxy
- List all services
- Get more details of a particular service
- Get more information about a label
- Use labels to query required pods
- Create a new label to the pod
- Scale up the above deployment to 4 replicas
- Scale down the above deployment to 2 replicas
- Update the image of the application in deployments
- Check the rollout status in deployments
- Rollback an update in deployments
- Delete the service created
- Helm charts and their need in Kubernetes
- Deploy an application with helm charts

DevOps Training - Week 8

Description:

Infrastructure as Code (IaC) is the management of infrastructure (networks, virtual machines, load balancers, and connection topology) in a descriptive model, using the same versioning as DevOps team uses for source code. Like the principle that the same source code generates the same binary, an IaC model generates the same environment every time it is applied. IaC is a key DevOps practice and is used in conjunction with continuous delivery. Tools like Terraform, AWS CloudFormation, Azure Resource Manager Templates, Google Cloud Deployment Manager Templates and OpenStack Heat are a great way to define server infrastructure for deploying software. The configuration to provision, modify, and rebuild an environment is captured in a way that is transparent, repeatable, and testable. Used right, these tools give us confidence to tweak, change, and refactor our infrastructure easily and comfortably. *During this week, we are going to provision all the AWS services using Terraform and we going to exclusively include IAC in our CICD process and we will deliver the release in blue-green deployments without any outage to the end customer.*

Topics Covered:

- Introduction to IaC
- Introduction to Terraform
- Terraform Installation
- Configuring terraform with AWS
- Create an EC2 instance with Terraform
- Variables in Terraform
- Output Attributes in Terraform
- State file
- Importance of Remote State file
- Data Sources
- Templates
- Modules in Terraform
- Best Practices of Module creation in Terraform
- Create all the resources with Terraform and launch complete infrastructure with Terraform
- Services: EC2, ALB, VPC, RDS, IAM , Beanstalk, S3, CloudWatch
- Introduction to Interpolation
- Conditionals
- Built-In Functions
- Best Practices of Terraform
- Include terraform CICD