

Why DevOps is Important

DevOps is important because it's a software development and operations approach that enables faster development of new products and easier maintenance of existing deployments.

DevOps—the amalgamation of development (Dev) and operations (Ops) teams—is an organizational approach that enables faster development of applications and easier maintenance of existing deployments. By enabling organizations to create stronger bonds between Dev, Ops and other stakeholders in the company, DevOps promotes shorter, more controllable iterations through the adoption of best practices, automation and new tools.

DevOps is not a technology per se, but it covers everything from the organization to culture, processes and tooling. Initial steps usually include Continuous integration and continuous delivery (CI/CD), real-time monitoring, incident response systems and collaboration platforms.

DevOps adoption is growing rapidly. IDC forecasts the worldwide DevOps software market to reach \$6.6 billion in 2022, up from \$2.9 billion in 2017. The forces driving DevOps adoption include enterprise investments in software-driven innovation, adoption of micro services-based architectures and associated development methodologies, and increased investment by CTOs and CEOs in collaborative and automated application development and operational processes, says IDC analyst Stephen Elliot.

Benefits of DevOps

Maximizes Efficiency with Automation

The late DevOps authority Robert Stroud said DevOps is all about "fueling business transformation" that encompasses people, process and culture change. The most effective strategies for DevOps transformation focus on structural improvements that build community.

A successful DevOps initiative requires a culture—or mind-set—change that brings greater collaboration between multiple teams—product, engineering, security, IT, operations and so on—as well as automation to better achieve business goals.

What kind of tangible benefits can DevOps bring? By managing engineering processes end to end, DevOps emphasizes deploying software more often, in a reliable and secure way through automation.

Optimizes the Entire Business

System architect Patrick Debois, best known as the creator of the DevOps movement, says the biggest advantage of DevOps is the insight it provides. It forces organizations to "optimize for the whole system," not just IT siloes, to improve the business as a whole. In other words, be more adaptive and data-driven for alignment with customer and business needs.

Improves Speed and Stability of Software Development and Deployment

A multi-year analysis in the annual Accelerate State of DevOps Report has found that top-performing DevOps organizations do far better on software development/deployment speed and stability, and also achieve the key operational requirement of ensuring that their product or service is available to end users. But given the somewhat fuzzy definition of DevOps, how can an organization determine if its DevOps initiative is paying off? The 2019 Accelerate report also names five performance metrics—lead time (i.e., the time it takes to go from code committed to code successfully running in production), deployment frequency, change fail, time to restore and availability—that deliver a high level view of software delivery and performance, and predict the likelihood of DevOps success.

Gets You to Focus on What Matters Most: People

People, not tools, are the most important component of a DevOps initiative. Key role-players (i.e., humans) can greatly increase your odds of success, such as a DevOps evangelist, a persuasive leader who can explain the business benefits brought by the greater agility of DevOps practices and eradicate misconceptions and fears. And since automated systems are

crucial to DevOps success, an automation specialist can develop strategies for continuous integration and deployment, ensuring that production and pre-production systems are fully software-defined, flexible, adaptable and highly available.

INTRODUCTION TO DEVOPS

- What is DevOps
- Why DevOps
- DevOps Principles
- DevOps Ecosystem
- Opportunities for DevOps Engineer
- DevOps Skills in demand
- Important tools used in DevOps

AWS

- EC2
- Vpc subnet elastic IP
- S3
- RDS
- Lamda
- Iam
- EFS
- EBS



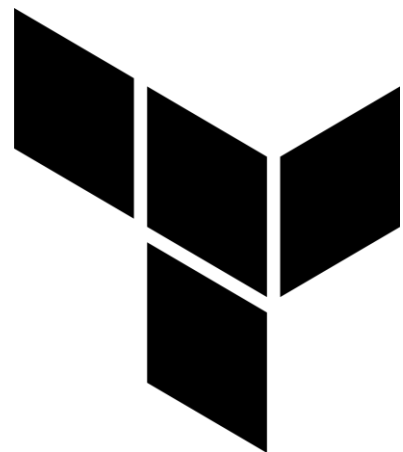
Azure

- VMS
- Azure Intra l'd
- Vnet
- Subnet
- Peering
- Storage
- Account container
- Fileshare
- Azure Network watcher
- Logs and metrics



Terraform

- Intro to iaac
- Install and configure Terraform
- Provider
- Build infra
- Change infra
- Destroy infra
- Variable
- State files and back-end



SOURCE CONROL USING GIT:

- Introduction to Version Control.
- Introduction to GIT.
- Installation and Server setup.
- Important Git Commands.
- Working with Repositories:
 - ✚ Creating a repository (git init)
 - ✚ Checking status (git status)
 - ✚ Adding files to a repository (git add)
 - ✚ Committing files (git commit)
 - ✚ Removing staged files (git reset)
- Branching in GIT.
- Merging in GIT.



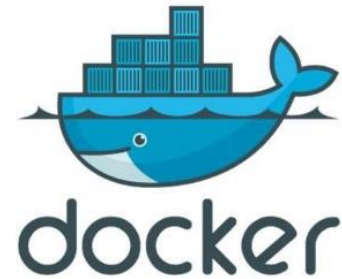
CONTINOUS INTEGRATION USING JENKINS:

- Introduction to Continuous Integration
- Continuous Integration with Jenkins Overview.
- Jenkins Architecture.
- Installation of Jenkins.
- Jenkins management
- Support for the Git version control systems
- Different types of Jenkins Jobs
- Setting up a Jenkins job
- Scheduling build Jobs.
- Securing Jenkins.
- Distributed builds with Jenkins
- Backup and Restore of Jenkins.



DOCKER

- Docker Concepts
- Installing Docker.
- Build Docker Images by using Docker Commit Command.
- Build Docker Images by Writing Dockerfile.
- Push Docker Images to Docker Hub.
- Docker Networking, Links and Volumes.
- Create Dockerized Web Applications.



NAGIOS

- Overview of Continuous Monitoring
- Fundamentals of Nagios
- Installing Nagios
- Nagios Plugins(NRPE) and Objects
- Nagios Commands
- Notification in Nagios



KUBERNETES:

- Introduction to Kubernetes
- Creating a Service
- Installing Kubernetes
- Working with Dashboard in Kubernetes
- Deploying an App using Dashboard
- Using Rolling Updates in Kubernetes
- Containers and Container Orchestration



PROJECT

CI/CD using Hashicorp Terraform and Jenkins

