**What is DevOps**

https://aws.amazon.com/devops/what-is-devops/

DevOps is the combination of cultural philosophies, practices, and tools that increases an organization’s ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market.

**Uses of DevOps**

Speed, Rapid Delivery, Reliability, Scale, Improved collaboration, Security,

**DevOps Practices.**

Continuous Integration, Continuous delivery, Microservices, [Infrastructure as Code](https://aws.amazon.com/devops/what-is-devops/#iac), Monitor and Logging, Communication, and collaboration

**CI** - Continuous integration is a software development practice where developers regularly merge their code changes into a central repository, after which automated builds and tests are run. The key goals of continuous integration are to find and address bugs quicker, improve software quality, and reduce the time it takes to validate and release new software updates.

**CD**- Continuous delivery is a software development practice where code changes are automatically built, tested, and prepared for a release to production. It expands upon continuous integration by deploying all code changes to a testing environment and/or a production environment after the build stage. When continuous delivery is implemented properly, developers will always have a deployment-ready build artifact that has passed through a standardized test process.

**Microservices -** The microservices architecture is a design approach to build a single application as a set of small services. Each service runs in its own process and communicates with other services through a well-defined interface using a lightweight mechanism, typically an HTTP-based application programming interface (API). Microservices are built around business capabilities; each service is scoped to a single purpose. You can use different frameworks or programming languages to write microservices and deploy them independently, as a single service, or as a group of services.



**GitHub -GitHub** SCM and VCS is a platform for hosting code that allows for version control and collaboration. It allows you and others to collaborate on projects from anywhere. This lesson will teach you the fundamentals of GitHub, such as repositories, branches, commits, and pull requests

**Uses of GitHub** - It's used for storing, tracking, managing, and collaborating on software projects. It makes it easy for developers to share code files and collaborate with fellow developers on open-source projects. GitHub also serves as a social networking site where developers can openly network, collaborate, and pitch their work

**Jenkins**

Jenkins is an open-source free automation tool used to build and test software projects. The tool makes it painless for developers to integrate changes to the project. Jenkins' primary focus is to keep track of the version control system and initiate and monitor a build system if there are any changes

**Uses**

Provides numerous plugins for integration with multiple test automation tools and frameworks into the test pipeline

Easily Integration with Other Tools

Plugins for integrations

**Docker**

Docker is an open-source platform for building, deploying, and managing containerized applications. Learn about containers, how they compare to VMs, and why Docker is so widely adopted and used.

**Uses of Docker**

Using Docker, you can quickly deploy and scale applications into any environment and know your code will run. Running Docker on AWS provides developers and admins a highly reliable, low-cost way to build, ship, and run distributed applications at any scale.

Scaling

Reliable

Low cost

Build, ship Distributed application at any Scale

**Kubernetes**

Kubernetes is an open-source container orchestration system for automating software deployment, scaling, and management, it’s used for bundling and managing clusters of containerized applications — a process known as 'orchestration' in the computing world

**Uses of Kubernetes**

Kubernetes services provide load balancing and simplify container management on multiple hosts. They make it easy for an enterprise's apps to have greater scalability and be flexible, portable, and more productive

No downtime for deployment

Automated Operations

Scalability – Scaling the deployment

Flexibility – Any cloud

Portability – Any Environments

Infrastructure abstraction – Networking and Storage

Container Management and Cluster Management tools

**Terraform**

Terraform is an open-source Automation and infrastructure as code software tool that enables you to create, change, and improve infrastructure safely and predictably.

**Uses of Terraform**

Create infrastructure, Multi cloud Deployment and infrastructure

Change infrastructure, Integrating with Existing workflows, Manage network infrastructure

Improve infrastructure

Efficiently and Safely

**Ansible**

Ansible® is an [open source](https://www.redhat.com/en/topics/open-source/what-is-open-source-software)[IT automation](https://www.redhat.com/en/topics/automation/whats-it-automation) tool that automates [provisioning](https://www.ansible.com/use-cases/provisioning), [configuration management](https://www.ansible.com/use-cases/configuration-management), [application deployment](https://www.ansible.com/use-cases/application-deployment), [orchestration](https://www.ansible.com/use-cases/orchestration), and many other manual IT processes

**Uses of Ansible**

Software installation and configuration Management

Automate daily tasks

Security

Connecting many web servers

Provisioning

**Helm**

A package manager for Kubernetes that makes it easy to take applications and services that are either highly repeatable or used in multiple scenarios and deploy them to a typical K8s cluster