DevOps Curriculum

Introduction to DevOps:

- Basics of DevOps, its history, principles, and key concepts.
- The emergence and history of DevOps.
- Core principles and key concepts of DevOps.
- The role of DevOps in software development.
- Measuring DevOps performance with key metrics.
- DevOps compared to SRE (Site Reliability Engineering).

Linux for DevOps and Shell Scripting:

- Introduction to Linux and its architecture.
- Linux commands and shell scripting essentials.

Version Control Systems:

- Introduction to Git and its significance.
- Git basics: cloning, branching, merging, and pull requests.
- Best practices for using Git.
- Advanced Git features: rebasing, cherry-picking, and bisecting.
- Utilising GitHub and GitHub Desktop.

Web Servers:

- Introduction to web servers.
- Importance of web servers in serving applications and content.
- Overview of common web servers: Tomcat, Nginx.
- Load balancing and reverse proxy concepts.
- Configuring HTTPS and SSL/TLS.
- Best practices for securing web servers.

Build Tools:

- Role of build tools in automating code compilation.
- Introduction to build tools.
- Common build tools: Maven, Gradle.
- Automating builds and managing dependencies.
- Creating effective build pipelines.
- Integrating build tools with version control and CI/CD.
- Best practices for using build tools.

Artifact Repository Tools:

- Understanding artifact repository tools.

- Common tools: Nexus, Artifactory.
- Publishing and retrieving artefacts.
- Managing dependencies with repositories.
- Integration with build tools and CI/CD.
- Best practices for artefact repositories.

Cloud Computing:

- Basics of cloud computing.
- Overview of major cloud provider AWS.
- Exploring virtualization and containerization.
- Introduction to serverless computing and event-driven architecture.
- Understanding cloud services like logging, metrics, alarms, and SNS.
- Networking, storage, load balancing, and DNS in the cloud.
- Ensuring cloud security and compliance.

Infrastructure as Code & Configuration Management:

- Utilising infrastructure as code tools like Terraform and Ansible.
- Terraform basics: creating and managing infrastructure.
- Best practices for Terraform.
- Introduction to Ansible.
- Ansible basics: infrastructure provisioning and management.
- Best practices for Ansible.

Containerization and Container Orchestration:

- Introduction to Docker and its significance.
- Docker basics: creating and managing containers.
- Best practices for Docker.
- Introduction to Kubernetes and its role.
- Kubernetes basics: setting up and managing clusters.
- Best practices for Kubernetes.
- Advanced Kubernetes features: networking, service discovery, and scaling.

Continuous Integration and Continuous Deployment:

- Understanding CI/CD and its importance.
- Introduction to Jenkins and other CI/CD tools.
- Jenkins basics: creating and managing build pipelines.
- Best practices for Jenkins.
- Implementing test automation and deployment strategies.
- Utilising GitHub Actions for CI/CD.

Monitoring, Logging & Alerting:

- Overview of monitoring and logging tools.
- Introduction to ELK (Elasticsearch, Logstash, Kibana).
- Basics of collecting and analysing logs with ELK.
- Best practices for ELK usage.
- Exploring Application Performance Monitoring (APM) tools.

Security:

- Emphasising security best practices and tools.
- Introduction to security practices.
- Network security and encryption fundamentals.
- Managing vulnerabilities.
- Ensuring security compliance and adhering to regulations.

Advanced Topics:

- Delving into advanced subjects like service meshes, serverless computing, and advanced Kubernetes features.
- Understanding service meshes.
- Exploring serverless computing.
- Advanced features within Kubernetes.

Staying Up-to-Date:

- Keeping current with DevOps developments.
- Following blogs, newsletters, and podcasts.
- Attending meetups and conferences.
- Enrolling in online courses and certifications.

Hands-on:

- Learning through practical projects.
    - Deploying application on aws
    - Automation to whitelist paths of a website using jenkins
- Building real-world projects.
- Troubleshooting and issue resolution.
- Understanding trade-offs between solutions.