Project: Building a Continuous Integration/Continuous Deployment (CI/CD) pipeline for a web application.

Tools used:

1. Version Control: Git

2. CI Server: Jenkins

3. Configuration Management: Ansible

4. Containerization: Docker

5. Orchestration: Kubernetes

6. Artifact Repository: Docker Registry

7. Monitoring: Prometheus + Grafana

8. Log Management: ELK Stack (Elasticsearch, Logstash, Kibana)

9. Cloud Provider: AWS (Amazon Web Services)

Project Overview:

1. \*\*Version Control (Git)\*\*:

- Set up a Git repository to store the source code of the web application.

2. \*\*Continuous Integration (CI) with Jenkins\*\*:

- Configure Jenkins to monitor the Git repository for changes.

- Set up Jenkins to build the application whenever changes are pushed to the repository.

- Run unit tests and code quality checks as part of the build process.

- Notify developers of build status via email or Slack.

3. \*\*Configuration Management with Ansible\*\*:

- Use Ansible to automate the configuration of servers.

- Define playbooks for provisioning the necessary infrastructure components (e.g., web servers, databases) required for the application.

4. \*\*Containerization with Docker\*\*:

- Dockerize the web application and its dependencies.

- Create Dockerfiles for building the application images.

- Set up a Docker Compose file for local development and testing.

5. \*\*Orchestration with Kubernetes\*\*:

- Deploy Kubernetes clusters for staging and production environments.

- Define Kubernetes manifests (Deployment, Service, Ingress, etc.) to deploy and manage the application containers.

- Implement rolling updates and scaling strategies for the application.

6. \*\*Artifact Repository (Docker Registry)\*\*:

- Set up a private Docker Registry to store Docker images.

- Push built Docker images to the registry during the CI/CD process.

7. \*\*Monitoring with Prometheus + Grafana\*\*:

- Install and configure Prometheus for collecting metrics from the application and infrastructure.

- Set up Grafana dashboards to visualize and monitor the metrics collected by Prometheus.

8. \*\*Log Management with ELK Stack\*\*:

- Deploy an ELK Stack (Elasticsearch, Logstash, Kibana) for centralized logging.

- Configure Logstash to collect logs from application containers and system components.

- Use Kibana to search, analyze, and visualize log data for troubleshooting and monitoring.

9. \*\*Cloud Provider (AWS)\*\*:

- Utilize AWS services such as EC2, EKS (Elastic Kubernetes Service), RDS (Relational Database Service), S3, etc., for hosting infrastructure components.

- Configure security groups, IAM roles, and other AWS resources as required.

With this setup, you'll have a comprehensive DevOps pipeline for building, testing, deploying, monitoring, and managing a web application efficiently.