Anand Pattanashetti

DevOps Engineer

EMAIL - ashetti.devops@gmail.com
Ph.no - +917411292959
GitHub PROFILE - GitHub
BLOGGING - Blogs
LINKEDIN - LinkedIn
PORTFOLIO - https://anandshetty.cloud
Bangalore Karnataka India - 560040

PROFESSIONAL SUMMARY

AWS Certified Solutions Architect - Associate and a DevOps Engineer with 2+ years of experience in designing and optimizing CI/CD pipelines, infrastructure automation, and deployments. Skilled in cloud tech, containerization, and orchestration tools. Proven in enhancing efficiency, streamlining workflows, and boosting team collaboration.

WORK EXPERIENCE

Associate DevOps Engineer : Cloud Kinetics Technology Solutions Private Ltd **April 2022 - Present**

- Collaborated closely with customers to understand their requirements and recommended feasible solutions, minimizing application downtime and reducing infrastructure costs.
- Led the **containerization** efforts using the **Docker** platform and managed **orchestration** with **Kubernetes** on **AWS EKS** Spearheaded the design, architecture, and implementation of scalable cloud-based web applications using AWS leveraging advanced **CI/CD pipelines.**
- Developers push code to a shared repository like **GitHub**, triggering a webhook that initiates a Jenkins build. Jenkins pulls the code,by using build tool compiles and unit test cases and then integrate **sonarqube** for code coverage and code quality check and we do **vulnerabilities** scan by using **Trivy** vulnerability scan and then build or package application and then building **Aritfacts** we push into the nexus repository and it push into **docker** it builds a **container** and pushes the image to an AWS or Docker registry. The pipeline includes security best practices, with **Trivy** scanning for **vulnerabilities** before the **image** is pushed. **Jenkins** then **deploys** the updated container to a **Kubernetes cluster** for **continuous deployment (CD)**. Post-deployment, **Grafana** visualizes the app's performance using data from AWS Monitor, providing insights into system health and stability.
- Led the deployment of a Java-based banking application, integrating Jenkins, Docker, and Kubernetes to establish efficient CI/CD pipelines. **This effort resulted in a 30% reduction in infrastructure costs** and a **50% improvement in system performance through advanced containerization and orchestration.** The project also involved **SonarQube** for code quality assurance, **OWASP dependency checks** for enhanced security, and Ansible for infrastructure automation, ensuring a secure and efficient deployment process.
- Designed and built advanced CI/CD pipelines using Jenkins, integrating tools like Git, SonarQube, Docker, Terraform, Kubernetes, Argo CD, and security tools **(SAST & DAST).** This modernized customer architecture, **leading to a 60% reduction in infrastructure costs** and a **50% enhancement in system performance.**
- Developed production-grade Terraform scripts utilizing conditions and functions to provision services on AWS automating infrastructure provisioning through **Terraform** and **CI/CD** tools. This approach optimized deployment processes, **eliminating 70% of manual work while ensuring high-quality code via continuous integration.**
- Implemented DevSecOps practices, integrating security tools **(SAST & DAST) into CI/CD pipelines** and at the application level, **enhancing system security by 60%.** Additionally, **implemented monitoring** solutions with **Prometheus, Grafana, Datadog, enhancing observability.**
- Implemented monitoring with tools like Prometheus, Grafana, Datadog, enhancing observability
- Successfully automated repetitive tasks using **Bash Script** and maximizing efficiency and ensuring consistent automation performance.

SKILLS

Versioning Tool - Git, GitHub, GitLab

Languages - Bash (Intermediate), Python (Beginner)

Cloud Platform - AWS

Containerization and Orchestration - Docker, Kubernetes

Web Server - Apache, Nginx

Infrastructure as a code - Terraform, Cloudformation (NoVoice)

Platforms/Operating System - Linux (Ubuntu, Amazon), Windows

Configuration Management Tool - Ansible

Build Tools - Maven, Gradle, NPM

SAST & DAST Tools - SonarQube, OWASP ZAP

CI/CD Tools - Jenkins,GitHub actions, Gitlab

Monitoring & Logging - Grafana, Prometheus,DataDog

Programming Language - Java, SQL

Databases - MySQL, DynamoDB

PROJECTS

Banking Application

- Deployed a Java-based banking application using Jenkins as the CI/CD tool, **achieving a 30% decrease in deployment time compared to manual methods**.
- Configured Docker containers to deploy the application, streamlining the deployment process and **reducing the time to launch new instances by 50%.**
- Orchestrated Kubernetes clusters to host the banking application, providing scalable infrastructure and high availability for production workloads.
- CI/CD Automation Implemented CI/CD pipelines with Jenkins, automating the build, test, and deployment processes. T**his automation led to a 30% reduction in manual effort**, improved development cycles, and a more secure deployment process through the integration of SAST and DAST tools.
- Enabled flexible deployment strategies by setting up Docker and Kubernetes environments, allowing for both standalone Docker containers and Kubernetes-managed clusters. This flexibility increased scalability by 50% in real-time operations.
- By integrating Jenkins with Docker and Kubernetes, it achieved a robust and efficient automated deployment process, significantly enhancing the team's ability to deliver and maintain the application.

CI/CD Pipeline For Container-Based Workloads: A DevOps Strategy

- After coding, developers push the code to a shared repository such as GitHub. Frequently merging the code and validating it is one way to ensure CI is error-free. To start the process, a GitHub webhook triggers a Jenkins project build. When code changes are made and committed to the repository, the pipeline gets activated. It downloads the code and triggers a build process.
- In this step, the code is compiled, artifacts are built, dependencies are sorted out and stored in the repository. Environments are created, containers are built and images are stored for roll out. This is followed by the testing processes. The Jenkins build job uses a dynamic build agent in AWS Elastic Kubernetes Service (EKS) to perform a container build process.
- A container image is created from the code in source control and is then pushed to an AWS/Docker Container Registry.
- Using the process of CD, Jenkins deploys an updated container image to the Kubernetes cluster..
- A Grafana instance provides visual dashboards of the application performance based on the data from AWS

Monitor.

CI/CD Pipeline Implementation and DevOps Practices

- Ensured effective version control and collaboration by managing the source code repository with Git.
- Created Docker files to build custom Docker images, facilitating consistent and reproducible application environments.
- Automated the build, test, and deployment processes, leading to faster and more reliable software delivery.
- Configured GitLab webhooks to trigger Jenkins jobs automatically, streamlining the CI/CD process.
- Incorporated SonarQube into the CI/CD pipeline to enforce code quality standards and provide continuous feedback to developers.
- Enhanced application security by integrating OWASP Dependency Check to identify vulnerabilities in third-party libraries and Trivy to scan Docker images for security issues.
- Set up Nexus3 for artifact management, enabling efficient storage, versioning, and retrieval of build artifacts.
- Automated application deployment and lifecycle management in Kubernetes environments with ArgoCD.
- Developed Grafana dashboards to monitor system performance, providing real-time insights into application health and stability.
- Produced comprehensive documentation detailing the installation, configuration, and operation of the CI/CD pipeline and associated tools.

USE-CASE - End-To-End DevOps Tools On AWS

- **CodeCommit:** A Git-based source code repository in AWS used to store, manage, and track the codebase, allowing for collaboration and version control.
- **CodeBuild**: A continuous integration service that compiles code and builds artifacts like executables or Docker images. It fetches code from repositories like CodeCommit and provides various build environments.
- **CodeDeploy:** A service that automates code deployment to AWS resources like EC2 instances or Lambda functions. It helps manage releases, track deployment history, and roll back if needed.
- **CodePipeline:** A service that orchestrates the entire CI/CD workflow. It integrates with CodeCommit, CodeBuild, and CodeDeploy to automate the pipeline from code commit to deployment, allowing the definition of stages for building, testing, and deploying applications.

CERTIFICATES

1. **AWS Certified Solutions Architect - Associate - AWS**
2. **DevOps With AWS - MicroDegree**

INTEREST

- TECH BLOGGING
- READ BOOKS
- GYMNASTICS

EDUCATION

BLDEA's V P Dr P.G. Halakatti College of Engineering & Technology *University : VTU Belagavi*
Bachelor of Engineering