

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

July 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 **Artifacts** 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, BitBucket

Languages - Bash , Python

Cloud Platform - AWS

Containerization and Orchestration - Docker, Kubernetes

Web Server - Apache, Nginx

Infrastructure as a code - Terraform

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

Configuration Management Tool - Ansible

Build Tools - Maven, Gradle, NPM

SAST & DAST Tools - SonarQube, OWASP ZAP, SonarCloud

CI/CD Tools - Jenkins, GitHub actions, Gitlab

Monitoring & Logging - Grafana, Prometheus, DataDog

Programming Language - Java, SQL, Python

Databases - MySQL, DynamoDB

PROJECTS

Cooperative Banking Solution Deployment

- 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. **This 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.

Infrastructure Development

- **Architectural Design** : Developed detailed architectural diagrams to align with client specifications, ensuring clarity in project scope and technical requirements.
- **Infrastructure Provisioning** : Provisioned AWS infrastructure using Terraform templates for key components including VPC, NACL, EKS, ElastiCache Redis, IGW, NAT Gateway, RDS, ALB, CloudFront, Route 53, ACM, enabling a robust and scalable architecture.
- **CI/CD Automation** : Streamlined CI/CD processes by automating Docker image builds and deployments to Amazon ECR using AWS CodePipeline, resulting in faster and more reliable deployment cycles.
- **Application Monitoring** : Integrated AWS monitoring tools (CloudWatch, X-Ray) to enhance application observability and performance tracking, providing actionable insights for performance optimization.
- **Microservices Deployment** : Automated the deployment of microservices into an EKS cluster with ArgoCD,

ensuring continuous deployment of image builds and improving operational efficiency.

Integrated Business Management Solution

- Containerized Java Based microservices, deploying them in a container orchestration environment.
- Constructed Terraform scripts for automating infrastructure provisioning, boosting operational efficiency by 40%.
- Implemented multiple CI/CD pipelines using Jenkins, integrating with Git for version control, SonarQube for code quality checks, Trivy for image vulnerability scanning, and OWASP ZAP for dynamic security testing to streamline the security of deployments.
- Contributed to cloud bill optimization efforts, resulting in a 30% reduction in overall costs.
- Deployed Prometheus for monitoring and Grafana for visualization of application performance metrics.
- Documented a comprehensive runbook and operations manual, providing clear guidelines for operational procedures.

End-to-End DevOps Automation for Cloud-Hosted Application

- Led the development and deployment of a Java-based application with a **CI/CD pipeline** integrated into **AWS** for seamless automation and cloud management.
- Managed source code using **Bitbucket** and integrated **SonarCloud** for continuous **code quality** analysis.
- Utilized **AWS CodeArtifact** to download necessary dependencies during the build process with Maven.
- Built application artifacts using AWS CodeBuild and stored them in **S3** for further use.
- Automated deployment to **AWS Elastic Beanstalk**, with **AWS RDS** handling database connectivity.
- Implemented a post-deployment process to execute Selenium test suites using CodeBuild.
- Designed and managed the pipeline using **AWS CodePipeline**, incorporating CodeDeploy for **efficient** and **reliable** application delivery.

CI/CD Pipeline Implementation and DevOps Practices

- Ensured effective version control and collaboration by managing the source code repository with Git.
- Created Dockerfiles 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.

CERTIFICATES

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

EDUCATION
