ANKITH B V

DevOps Enthusiast

https://github.com/Ankithv007 ankithbv007@gmail.com

https://www.linkedin.com/in/ankithbv **+91 7406552044**

Motivated DevOps Engineer with a focus on automation and cloud technologies. Experienced in building CI/CD pipelines and managing scalable infrastructure to enhance development workflows. Committed to improving team collaboration and delivering efficient solutions in fast-paced environments.

Technical Skills

Cloud Technologies	AWS
AWS Services	AWS VM,AWS IAM, AWS CI/CD, AWS EKS, AWS PipeLine,AWS ECR, AWS CLI …..
Container Technologies	Docker, Kubernetes
Scripting	Shell and Python
Version Control System	Git and GitHub
Configuration Management	Ansible
Infrastructure as Code	Terraform, AWS CloudFormation
CI/CD	Jenkins
Observabilty	Prometheus and grafana

PROJECTS

Wanderlust-Mega-Project (end-to-end CI/CD)

https://github.com/Ankithv007/Wanderlust-Mega-Project

- Designed and implemented a CI/CD pipeline using Jenkins for continuous integration (CI) and continuous deployment (CD) across stages like code quality checks, security scans, and deployment.
- Automated pipeline triggering when developers push code to GitHub, integrating changes seamlessly.
- Integrated OWASP to perform vulnerability analysis and dependency checks to ensure application security.
- Used SonarQube to analyze the codebase for quality gates, ensuring coding standards are met and potential issues are flagged.
- Integrated Trivy to conduct security scanning during Docker image builds, ensuring no vulnerabilities in containerized applications.

- Automated Docker image builds, versioning, and pushing to a container registry using Jenkins.
- Managed deployments on Kubernetes to ensure high availability and scalability.
- Set up Prometheus and Grafana for monitoring application performance and resource usage, with visual dashboards for proactive management.
- Configured Jenkins to send real-time email notifications on build and deployment status for team awareness.

Highly Available Web Application on AWS

https://github.com/Ankithv007/aws-/blob/main/README.md

- Architected a Virtual Private Cloud (VPC) with public and private subnets distributed across multiple Availability Zones for enhanced availability and fault tolerance.
- Configured an Application Load Balancer (ALB) to route incoming traffic efficiently across EC2 instances, improving application scalability.
- Deployed EC2 instances within private subnets to secure backend services, with NAT gateways in public subnets allowing outbound internet access.
- Automated instance scaling using an Auto Scaling group to dynamically adjust resources based on traffic load, optimizing performance and cost.
- Designed a secure network by implementing security groups to control access to EC2 instances, defining both inbound and outbound traffic rules.
- Created and configured NAT gateways in the public subnet to provide secure internet access to instances in private subnets without exposing them directly to the internet.
- Ensured redundancy and high availability by distributing infrastructure components across multiple availability zones.
- Utilized private subnets to protect sensitive backend services while enabling communication with the public-facing components.
- Configured routing tables to ensure proper traffic flow between public and private subnets within the VPC.
- Applied AWS best practices for network architecture, including separating public and private resources for better security management.
- Integrated an Application Load Balancer with health checks to automatically reroute traffic away from unhealthy instances, ensuring continuous service availability.
- Designed the architecture to be highly scalable and resilient to potential failures by leveraging AWS's multi-AZ capabilities.
- Configured the Auto Scaling group to launch new EC2 instances as traffic increased, ensuring a seamless user experience even under heavy loads.
- Defined security group rules to minimize the attack surface by allowing only necessary traffic, such as HTTP/HTTPS to web servers and SSH for secure access

Game 2048 on Amazon EKS

https://github.com/Ankithv007/aws-/tree/main/day22

- Deployed the classic 2048 game on Amazon EKS (Elastic Kubernetes Service) to manage and scale Kubernetes clusters effectively.

- Used Kubernetes Services and Ingress with an Ingress controller for managing traffic routing and external access to the 2048 game application.
- Utilized AWS Application Load Balancer (ALB) as the Ingress controller for efficient traffic distribution across game services.
- Configured AWS Load Balancer Controller to automate provisioning and management of AWS resources like ALBs in the Kubernetes cluster
- Implemented NodePort Service to expose the game application on a specific port, enabling external access.
- Ensured high availability by deploying the game in Kubernetes pods with 5 replicas, balancing traffic across instances.
- The use of ALB as the ingress controller provided seamless load balancing and traffic management,simplifying operations and improving application performance.
- Integrated AWS services such as IAM for enhanced security and identity management within the Kubernetes environment.

EDUCATION

MVJ COLLEGE OF ENGINEERING - B.E (CSE) [2019-2023]

NATIONAL COLLEGE BASAVANAGUDI - 12th [2016 - 2018]

GOVERNMENT HIGH SCHOOL BHUVANAHALLI - 10th [2016]

ACHIEVEMENTS

- Attended Tata NQT exam.
- Completed 3-month Java course with certification.
- Team Lead in final year project.

ADDRESS

#113
3RD Cross Pragathipura
Banashankari
Bengaluru - 560070