Anish Singh

anishsengarr123@gmail.com 9305204647 Varanasi, U.P github.com/anishsengarr
linkedin.com/in/anish-singh-devops https://anishsengarr.github.io/

PROFESSIONAL SUMMARY

Proficient in DevOps, Multi-Cloud (AWS,Azure) including analysis, design, scripting, testing, automation, version control and, documentation.

PROFESSIONAL EXPERIENCE

Analyst 02/2019 – 12/2023
GlobalLogic Technologies Limited Gurugram

- Designed end-to-end CI/CD workflows using Jenkins, Terraform, and AWS Lambda for rapid and reliable software deployments.
- Worked on multi-tier applications, managing web server configurations with NGINX and Apache to optimize performance and load balancing.
- Responsible for end-to-end build and deployments process for both Non-Prod and Prod environments.

Desktop Support Engineer 10/2017 – 12/2019
AGS Transact Technologies Ltd Jaipur

- Created Docker file and automated docker image creation using Jenkins and Docker.
- Collaborated with developers and analysts to meet project requirements. Effectively managed time and prioritized multiple projects.
- Supported the application in production and worked closely with offshore and onshore teams.

EDUCATION

B.Tech in Electronics & Comm. Engineering 04/2013 – 06/2017
BBDNIIT ,Lucknow UP

SKILLS

Technical Skills

- Containerization : Docker, Kubernetes, Helm
- Cloud Platforms : AWS, Azure
- CI/CDTools : Jenkins, GitHub, GitLab CI, GitHub Actions
- Configuration Management : Terraform, Ansible
- Scripting Languages : Python, Bash
- Monitoring : Prometheus, Grafana, Cloudwatch, ELK Stack
- Security : SonarQube, Trivy, IAM, Load Balancer
- Other : CloudFormation, ALB, Ingress, EKS, EC2,S3, Argo CD, Lambda, VPC, Linux

TECHNICAL SUMMARY

DEVOPS

- Configuration Management with Ansible: Implemented Ansible playbooks to automate configuration tasks on multiple servers.
- Proficient with version control systems like Git.
- Capable of integrating Git/GitHub with CI/CD pipelines, leveraging hooks, and implementing automated testing and deployment processes.
- Knowledge in Docker, adept at creating, managing, and optimizing containerized applications for efficient deployment and scalability.
- Skilled in creating and managing Kubernetes resources such as pods, deployments, and services,

AWS

VPC, EC2, S3, Route 53, CloudFront, Elastic Load Balancing, Amazon EBS, AWS Backup Service, CloudWatch, CloudTrail, RDS, IAM, Security groups, NACLs, VPC peering, EFS, AWS Shield
AWS Lambda

LINUX

- Linux Security: User permissions, access controls.
- System Administration: User management, file system management, system configuration.
- Collaboration Tools: SSH, scp, rsync, cron.
- Storage Management: Managing file systems, partitions, and disk storage.
- Package Management: Proficient in using package managers for software installation, updates,and dependency management.

PROJECTS

CICD Pipeline for Java Application to deploy on Kubernetes Cluster using Jenkins

- Integration of Jenkins & Sonarqube for static code analysis.
- Dockerizing Application and pushing the image to private registry.
- Identifying the misconfiguration in HELM charts using datree.
- Pushing the helm charts to nexus for re-usability.
- Manual approval for deployment.
- Deploying the hhapplication on k8s cluster using the helm charts.
- Configuring mail server and Enabling pull request trigger.

Automated CI/CD Pipeline for Django Web Application using AWS, Docker, Jenkins and Kubernetes

- A Real time CI/CD Pipeline on Jenkins with GitHub Integration and a system designed to handle Database, Server and orchestration using Docker, Docker Compose, Ansible.

Integration of AWS SDK wrapper and Disk Schedular with Product Backend

- Accessing AWS and Python using BOTO3. Formation of AWS wrapper to compose S3 Bucket using AWS CLI and Python to automate task of peculiar acclaimed services of AWS.
- Automate disk usage scheduling using the Python schedule library.

Automating the Delivery of Highly Scalable Reddit Clone Application using CI/CD Pipeline with Docker, Docker Hub, and Kubernetes Minikube

- For the CI component, I set up an automated pipeline that integrated code changes from the development team and tested them before pushing the updated container images to Docker Hub. This ensured that any changes made were thoroughly tested before deployment, thus improving the overall quality of the application.
- For the CD component, I used Kubernetes Minikube to deploy the container images to the appropriate environment based on the results of the tests. The application was designed to be highly available and scalable, leveraging Kubernetes features such as services and ingress.
- To ensure optimal performance, I also implemented monitoring and logging functionalities to identify and troubleshoot any issues that may arise. The pipeline was designed with automation in mind, minimizing manual interventions and reducing the likelihood of human errors.

CERTIFICATES

Internet of Things(IOT)

17/08/2019 -03/10/2019