Anushka Pawar

Site Reliability | AWS Cloud | DevOps
panushkapawar92@gmail.com | https://anushka-92.github.io/Portfolio/ | +91-9307781766

Education

MIT ACADEMY OF ENGINEERING
B.Tech in ENTC
2019-2023 Pune, India
CGPA - 8.23/10

RAJASHRI SHAHU COLLEGE
2018-2019 Latur, India

KESHAVRAJ MAHAVIDYALAY
2016-2017 Latur, India
Class X - 96.20%

Links
GitHub **Anushka-Pawar**
LinkedIn **Anushka-Pawar**

Skills
CLOUD INFRASTRUCTURE
• AWS-Console, CLI and SDK(Python Boto3)

CONTAINERS AND CONTAINER ORCHESTRATION
• Docker(ECS) • Kubernetes (EKS)

AWS SERVICES

INFRASTRUCTURE AS CODE
• Terraform

PROGRAMMING
• C • Python

LINUX
GIT
DBMS

Extra Curricular Activities
SOCIAL VOLUNTEERING
Worked with an NGO (RHA) ON DIFFERENT DRIVES LIKE FOOD DISTRIBUTION,BLOOD DONATION AND PALKHI.

Hobbies
• Mandala Art
• Yoga
• Travelling

Projects
PROVISIONED AWS INFRASTRUCTURE
KEY SKILLS : VPC,Security Group, EC2,Terraform, CLI
Terraform-AWS project link
- Created VPC and Subnet along with custom Route Table
- Configured Default/Main Route Table and Created Security Group. Created EC2 instance.
- Configured ssh key pair in Terraform config file
- Configured Terraform to install Docker and run nginx image and Configured Terraform to install Docker and run nginx image

SCALABLE WEB APPLICATION DEPLOYMENT ON AWS
KEY SKILLS : AWS EC2, ELB, AUTOSCALING , IAM
AWS project link
- Different target groups have been created according to needs.
- Application load balancer is created and different rules are set to redirect traffic to particular target group.
- EC2 instance is launched with help of Template and user-data.

CLOUD NATIVE MONITORING APP
KEY SKILLS : EKS, ECR, Docker, Boto3 ,AWS-CLI
Cloud-monitoring project link
- Created cpu, memory utilisation application
- Dockerised the monitoring application by configuring containers , dockerfile creation.
- Pushed image to ECR repository
- Deployement of application using kubernestes cluster

DEVELOPING WITH DOCKER
KEY SKILLS :Docker, JENKINS, GIT ,MONGODB
Docker project link
- Developing an javascript application.
- Used docker container of Mongo DB dependency from dockerhub
- Commited code to git and used CI tool jenkins
- Pushed docker image into a private Registry on AWS
- Deploying our containerized application
- Configured persistence using volume for application