DEEPAKKUMAR RATHOUR

AWS | DevOps | Linux | Terraform | Ansible | Git | GitHub | Docker | K8s | Shell | Ci/CD Pipeline

+91 7264940479 | rathourdeepakkumarjofficial@gmail.com | Linkedin ↗ | GitHub ↗ | Portfolio ↗

Summary: DevOps Engineer with 6 months of hands-on internship experience in AWS, Linux, Terraform, Ansible, Docker, Kubernetes, and Jenkins. Skilled in CI/CD pipelines, automation with Shell scripting, and Infrastructure as Code. Strong problem-solving and collaborative mindset with a passion for scalable cloud solutions.

Skills:

Technical Skills: AWS Cloud, Linux, Ansible, Terraform, Shell scripting, DevOps tools, Docker, Kubernetes, Jenkins, CI/CD Pipeline, Git, GitHub, Infrastructure Automation, Azure (App Service, CI/CD, Publish Profiles), Prometheus with Grafana, CloudWatch, ArgoCD, Helm, EKS, ECS.

Professional Skills: Problem-Solving, Collaborative abilities, Adaptability, Communication, Attention to Details, Leadership skills.

Education:

Dr. Babasaheb Ambedkar Technological University Lonere (DBATU) 2021-2024
B.Tech, Computer Science and Engineering.
CGPA - 8.36

Maharashtra State Board of Technical Education (MSBTE) 2019-2021
Diploma in Engineering, Electronics Engineering
93.56 %

Key Achivements :

- Increased Deployment Efficiency
  Improved deployment speed by 30% through Dockerized Log Analyzer Tool.
- Facilitated Team Collaboration Improved team collaboration by 25% with shared Terraform state management.
- Automated Cloud Provisioning
  Reduced manual provisioning by 40% using AWS CloudFormation.
- Optimized Storage Solution
  Lowered infrastructure costs by 20% with automated file archiving.

Deepakkumar Rathour

Date:

Experience:

CoreXtech IT Services Pvt Ltd. Remote
Cloud Engineer Intern Feb 2025 – Jul 2025

- Assisted in designing and deploying AWS infrastructure (EC2, S3, IAM, VPC, RDS), supporting VM setup, security group configurations, and implementing cloud security best practices (least privilege access, encryption, secure networking).
- Automate infrastructure provisioning with Terraform & CloudFormation, reducing manual setup by 40%, and gained hands-on experience in IAM roles, policies, and permissions for secure resource management.
- Monitored performance and resource utilization with Amazon CloudWatch, identifying issues, while contributing to team meetings, documentation, and knowledge sharing for effective project tracking.

Corestance Technology Pvt Ltd. Remote
Frontend Developer Mar 2024 – Sep 2024

- Developed responsive, cross-browser websites using HTML, CSS, JavaScript, Bootstrap, and Tailwind CSS, optimizing layouts for mobile/desktop to enhance performance, usability, and reduce costs.

Projects:

1. NGINX Deployment on AWS ECS Fargate.
   AWS | ECS | Fargate
   - Hosted a containerized NGINX application on AWS ECS Fargate, creating task definitions, clusters, and services to demonstrate scalable, serverless container deployment without managing EC2 instances.

2. Deploying a Containerized Application on Amazon EKS using Ingress and AWS Load Balancer Controller | Link ↗
   AWS | EKS | Kubernetes | Fargate | Ingress | ALB | IAM | Helm | CI/CD | Cloud Infrastructure | DevOps Practices
   - Deployed a containerized 2048 game application on Amazon EKS using AWS Fargate for serverless Kubernetes workloads.
   - Configured Kubernetes manifests Namespace, Deployment, Service, and Ingress to automate application deployment and networking.
   - Implemented Ingress-based routing with the AWS Load Balancer Controller for secure and cost-efficient external access through Application Load Balancer (ALB).
   - Integrated IAM OIDC and service accounts for fine-grained access control between AWS and Kubernetes.
   - Utilized Helm, kubectl, and AWS CLI for cluster setup, configuration, and resource management.

3. Two-Tier Application Deployment on AWS EKS using Docker, Kubernetes & Helm | Link ↗
   AWS | EKS | Kubernetes | Docker | Helm | Flask | MySQL | eksctl | CI/CD | Load Balancer | DevOps Practices
   - Deployed a scalable two-tier Flask–MySQL application on Amazon EKS capable of handling 10,000+ concurrent users using best DevOps practices.
   - Containerized application components using Docker and Docker Compose, and pushed images to DockerHub for version control and reusability.
   - Automated Kubernetes cluster setup with kubeadm and AWS EKS (eksctl) for managed orchestration and scalability.
   - Packaged and deployed Kubernetes manifests using Helm charts, enabling modular and version-controlled deployments.
   - Configured multi-node cluster and AWS Load Balancer for high availability and fault tolerance, achieving 60% downtime reduction.