SATYAM ARYA

MERN Stack Developer | Cloud & DevOps Enthusiast

✉ satyam19arya@gmail.com

+91 9452851038

❖ GitHub

❖ LinkedIn

❖ Tech Blogs

Profile

Currently in 4th year pursuing BTech in Computer Science and Engineering with specialization in Cloud Computing.

Skills

Languages: C++
Skills: Html, CSS, JavaScript, React, Redux, Bootstrap, MongoDB, Express, Insomnia, Strapi, Linux, Docker, Terraform, Ansible, Jenkins, CircleCI, ArgoCD, Kubernetes
Cloud: AWS
Others: Git/GitHub, Lucid chart

Achievements

- AWS Certified Cloud Practitioner
- Microsoft Certified: Azure Fundamentals
- Microsoft Certified: Azure Data Fundamentals
- HashiCorp Certified: Terraform Associate
- Hackcbs5.0 Hackathon
- Yuva's Knack Hackathon

Education

BTech in CSE with specialization in Cloud Computing
SRM Institute of Science and Technology
9.05 CGPA

12th (Science)
Guru Har Rai Academy | Kanpur, UP
Percentage: 83%

Projects

GitOps-Driven MERN App (social media) on AWS EKS with Prometheus Monitoring

- Implemented user authentication using JWT tokens, ensuring secure access to the application
- Create and Update Account with Profile Picture, Name & Bio
- Create Post with Text and Image
- Implemented a like functionality for posts
- Utilized Redux for state management
- Leveraged MongoDB Atlas for efficient and scalable data storage.
- Integrated Cloudinary for image storage
- Provisioned infrastructure on AWS using Terraform (AWS EKS)
- Set up a CI/CD pipeline using CircleCI, automatically triggered upon code commits.
- Implemented a GitOps workflow by synchronizing ArgoCD with the GitHub manifest repository.

Ecommerce Application using Strapi with Multi-AZ AWS Deployment

- Design frontend using ReactJs and utilized Redux for state management.
- Utilized Strapi, a headless CMS, for efficient backend management, ensuring smooth content updates and product additions.
- Integrated the Stripe payment gateway to facilitate secure payments.
- Utilized Cloudinary for image storage.
- Deployed the application on AWS to ensure high availability, scalability, and security.
- Created a custom VPC to isolate and secure resources.
- Launched and managed EC2 instances to host the application.
- Configured security groups to control inbound and outbound traffic.
- Implemented auto-scaling of EC2 instances based on traffic demands to maintain performance and cost-efficiency.
- Utilized RDS (PostgreSQL) to store and manage the application's data, ensuring high availability and scalability.
- Utilized Route53 for domain management, AWS Certificate Manager for a valid SSL/TLS certificate and Amazon CloudFront distribution to serve website globally and AWS WAF to protect against common web-based attacks.