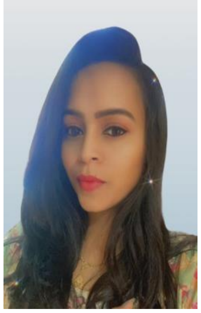
Curriculum Vitae

Likhitha M. S

DevOps Engineer with 4+ years of experience in handling the complete end to end automated DevOps process using multiple tools and technologies like **Shell Scripting, Git, GitHub, Jenkins, AWS DevOps, Terraform, Docker, Kubernetes, Cloud Technologies (AWS)**.



Overall Experience
Roles and responsibilities
I have been involved in various engagements in AWS DevOps mainly working on the above-mentioned activities:

- Implementing and handling the **automated process** of **DevOps** and Integration of the same with **Cloud** as per different Business Requirements.
- Managed GitHub repositories and permissions, including branching and tagging.
- Implemented infrastructure as code using Git, CI/CD pipelines, Ansible, and Terraform, reducing manual configuration efforts by 70%.
- Having knowledge in encryption mechanisms and key management strategies for data at rest and in transit using AWS Key Management Service (KMS)
- Automated security compliance checks using AWS Config Rules and AWS Lambda functions, reducing manual effort and ensuring continuous adherence to security standards.
- Implement robust monitoring tools to track key metrics, ensuring continuous measurement against SLOs.
- Designed and deployed cloud solutions in AWS, ensuring high availability and scalability for client applications.
- Ensured efficient data storage and adherence to security policies, implementing IAM roles, security groups, and network ACLs to protect sensitive data.
- experienced Cloud Engineer with designing and supporting cloud environments, including IaaS and PaaS

System Engineer

EVRY India Private Limited

a Tieto EVRY Company

August 2023- Present

Languages Known

Kannada

English

Telugu

Technical Skills:

Cloud Platforms: AWS

Cloud Services: EC2, S3, RDS, VPC,

Functions Infrastructure as Code: Git,

Jenkins CI/CD pipelines, Ansible,

Terraform.

containerization: Kubernetes, Docker

Security: Identity and Access Management (IAM), Security Groups, Network ACLs, Cloud trail

Knowledge of AWS config, AWS service catalog, cost explorer,

- Implemented end-to-end DevOps practices, integrating CI/CD pipelines for rapid and reliable software delivery.
- Created DevOps best practice documents in confluence and shared with team to follow best practices.
- Collaborated with infrastructure partners to establish robust test data backup and recovery tools, ensuring data integrity and business continuity.
- Automate the build of containerized systems with CI/CD tooling, Helm charts, and more.
- Designing and deploying new infrastructure in AWS with S3, VPC, subnets, Load Balancer, Cloud Watch
- Created technical design recommendations for developing and integrating new software and system technologies, ensuring alignment with written specifications.
- Led the adoption of serverless computing by implementing AWS Lambda for specific workloads, reducing infrastructure costs and improving scalability.
- Led the containerization initiative by setting up a Kubernetes cluster and migrating multiple applications to Docker containers, improving scalability, resource utilization, and enabling seamless application deployment and management.
- Troubleshot and supported containerization technologies such as Kubernetes and Docker, enabling seamless application deployment and management.
- Build and configure Kubernetes-based infrastructure, networking policies, LBs, and cluster security, Define -auto scaling and cost strategies.
- Collaborated with cross-functional teams to implement best practices in the software development lifecycle, including unit and integration testing, source code management, and continuous integration.
- Implemented robust monitoring solutions like CloudWatch, Prometheus, or Grafana for proactive system health management.

Projects: 1
About Client: Gainwell Technology are responsible for building and operating critical internal and external services.

Responsibilities:

- Designed and implemented workflows to automate the creation of custom Amazon Machine Images (AMIs) using **AWS EC2 Image Builder** to standardize and accelerate the provisioning of infrastructure across environments.

Automation: Cloud Formation,

Monitoring: CloudWatch, Prometheus

Networking: DNS, VPN, Load Balancers

Scripting: Python, Bash

Operating Systems: Linux, Windows.

- Developed and maintained **CloudFormation templates** to automate the entire lifecycle of EC2 instances, from provisioning EC2 instances to creating AMIs using EC2 Image Builder and deploying the images to multiple environments.
- Configured EC2 Image Builder to include security updates, patch management, and compliance controls (e.g., CIS benchmarks, security patches) as part of the image creation process, ensuring secure and compliant base images.
- Implemented designed CloudFormation stack to support Mult environment deployments with environment-specific configurations, ensuring scalability and flexibility in the cloud infrastructure.
- Set up AWS CloudWatch metrics and alarms to monitor the image creation process and track the health of AMIs, reducing failures in the build process and improving reliability.
- Managed AWS CloudFormation stacks, including updates, rollbacks, and stack deletions, to ensure proper infrastructure management and minimal downtime during deployment.
- Collaborated with the Ansible team to automate application configuration on CloudFormation-provisioned infrastructure, improving application deployment speed and consistency.
- Integrated AWS CloudWatch alarms and custom metrics with CloudFormation to automatically monitor and notify the team about infrastructure health.

Projects: 2
About Client: DXC Technology are responsible for building and operating critical internal and external services.

Technical/Business Challenge

Mission Critical application used by bank customers in FAB. Currently, being provisioned on **AWS** Environment and facing challenges like:

AWS DevOps Engineer Valley Infosystem

Sep 2019 – August 2023

- Developers were building the application code locally and deployed to AWS environments (DEV, QA, PROD) and it was time consuming and prompt to manual error.
- Manual Release Process through Change Sets in AWS leads to the longer deployment cycle.
- Worked on Create API for Scanning Docker Image using Python.

Solution

- Introduced **Best Practices** of using Version Control System like **Git – GitHub.**
- Configured Jenkin to build and deploy application and IAC code on AWS environment using CI/CD
- Implemented **Best Security Practices** in **GitHub** by creating Groups for multiple teams like Admin, Dev, QA, Operations etc. and providing appropriate **permissions** to the respective groups for accessing **Project, Repos** etc.
- Migrated the on-premises infrastructure to **AWS** by using various DevOps and Cloud tools.

Benefits

- **Better Management** of **Code Base** through **Repos.**
- **Faster Releases** of Salesforce Application with the help of **Jenkin CI/CD Pipelines**.
- **Secured** the **GitHub** Project by assigning multiple **permissions** to **Users** and **Groups.**

Project:3

About Client: The client's mission revolves around leveraging cutting-edge technology to create a positive impact on Banking services.

Location: Bangalore

Environment: GIT, Maven, Jenkins, shell, Tomcat, Linux, Ansible, Docker, Terraform, Monitoring Tools (Grafana, Prometheus).

Technical/Business Challenge

Mission Critical applications are used by customers globally. Currently, it is being provisioned on-premises and facing challenges like:

- ➔ Developers were creating resources in AWS manually.
- ➔ Manual Release Process leads to the longer deployment cycle.
- ➔ No real time monitoring of the applications.

Solution

- ➔ Introduced **Terraform** to automate the creation of resources in AWS.
- ➔ Deployed their Application - **Frontend (React)** & **Backend (Python)** in **Docker** containers and automated it using **Jenkin CI/CD Pipelines**.
- ➔ Design the application from monolithic to microservice.
- ➔ **Docker Images** were deployed to **ECR** to **store** and **manage** the **container images.**

- ➔ **ECS for Containers** was configured to pull **container images** from **ECR.**
- ➔ **AWS S3** was used to store and reference application related files, images etc.
- ➔ Application related **Secrets** and **Certificates** were stored and referenced by the secret **manager.**

Benefits

- ➔ **Reduced cost** of idle infrastructure as everything is being managed by AWS and we have the **flexibility** to **choose pricing tier** as per our requirement.
- ➔ **Faster Releases** of Application with the help of **Jenkin CI/CD Pipelines**.
- ➔ Realtime Monitoring through different services in **AWS CloudWatch**

Education

B.E.
KNS Institute of Technology, Yelahanka, Bangalore
August 2016 – June 2019, **CGPA** – 7.5/10

Diploma
Sridevi Polytechnic, Sira gate, Tumkur

April 2014– April 2016, **Percentage** – 75%	