



K.Gunganathan – 6369283552/gugan1062001@gmail.com

Judges colony, SRP tools , kottivakkam | 6369283552 |gugan1062001@gmail.com |

Objective

Enthusiastic and highly motivated recent graduate with foundational knowledge in AWS cloud services, Linux administration, and DevOps practices. Eager to apply technical skills and cloud expertise in a challenging role to contribute to innovative projects and solutions.

Professional Summary

- Proficient in AWS services including EC2, RDS, Redshift, ELB, ASG, S3, VPC, VPC Peering, Transit Gateway, IAM, CloudWatch, Route 53, SNS & SQS, CloudFront, CloudFormation, ECS, ECR, EKS, ASM.
 - Hands-on experience with static website hosting using S3 and CloudFront.
 - Capable of deploying infrastructure as code with CloudFormation.
 - Basic Linux administration skills.
 - Completed a three-tier architecture project using AWS services.
 - Strong problem-solving abilities and a quick learner.
-

Technical Skills

Cloud Services: EC2, RDS, Redshift, ELB, ASG, S3, VPC, VPC Peering, Transit Gateway, IAM, CloudWatch, Route 53, SNS & SQS, CloudFront, ECS, ECR, EKS, ASM

DevOps Tools: Jenkins, Docker

Languages: Python, Bash, Shell Scripting

Operating Systems: Linux (Ubuntu, CentOS, Amazon Linux)

Monitoring & Logging: CloudWatch

Networking: VPC, Subnets, Security Groups, NACLs, Route Tables

Version Control: Git, GitHub

Projects

Static Website Hosting with S3 and CloudFront

- Deployed a static website using S3 for storage and CloudFront for content delivery.
- Configured S3 bucket policies and IAM roles to secure access to website content.
- Optimized website performance and availability by leveraging CloudFront's edge locations.

Three-Tier Architecture Implementation

- **Objective:** Designed and implemented a scalable, secure, and highly available threetier architecture using AWS services.
- **Architecture Overview:**
 - **Web Tier:**
 - ▢ Deployed Amazon EC2 instances in public subnets across two Availability Zones (AZ1 and AZ2) for high availability.
 - ▢ Configured an Internet Gateway to allow external access to the web servers.
 - ▢ Utilized Elastic Load Balancing (ELB) to distribute incoming traffic across multiple EC2 instances, ensuring fault tolerance and load distribution.
 - **Application Tier:**
 - ▢ Deployed Amazon EC2 instances in private subnets within both AZ1 and AZ2 to host application logic, isolating them from direct internet access.
 - ▢ Configured security groups and network access control lists (NACLs) to restrict access to the application servers.
 - ▢ Ensured communication between the web tier and the application tier via ELB.
 - **Database Tier:**
 - ▢ Utilized Amazon Aurora for the database layer, with the primary database instance deployed in a private subnet.
 - ▢ Configured an Aurora Read Replica in a separate private subnet for read scalability and high availability.
 - ▢ Implemented automated backups, snapshots, and multi-AZ replication for data durability and disaster recovery.
- **Key Features:**
 - **High Availability:** Achieved by distributing resources across multiple Availability Zones, ensuring redundancy and minimizing downtime.
 - **Security:** Implemented robust security measures, including the use of private subnets for the application and database tiers, and security groups to control inbound and outbound traffic.
 - **Scalability:** Enabled auto-scaling for both the web and application tiers to automatically adjust the number of running instances based on demand.
 - **Automation:** Used CloudFormation templates to automate the deployment and management of the entire infrastructure, ensuring consistency and repeatability.
 - **Monitoring:** Leveraged CloudWatch for monitoring system performance and setting up alarms for critical metrics.

- **Cost Efficiency:** Optimized the use of AWS resources to balance performance and cost, utilizing Reserved Instances and Spot Instances where applicable.

Integration of EC2 with S3 and CodeDeploy Automation

- **Objective:** Enhanced project automation and deployment efficiency by integrating EC2 servers with S3, automating deployments using CodeDeploy, and monitoring with CloudWatch.
- **Steps Involved:**
 - **EC2 and S3 Integration:** Configured EC2 instances to interact with S3 for storing and retrieving application data.
 - **CodeDeploy Setup:** Automated the deployment of application updates from S3 to EC2 instances using CodeDeploy.
 - **CodePipeline Automation:** Set up CodePipeline to automate the build, test, and deploy process, triggered by changes in the S3 bucket.
 - **Monitoring with CloudWatch:** Utilized CloudWatch to monitor pipeline activities, deployment success/failure, and set up alerts for critical events.
- **Key Features:**
 - **Automation:** Reduced manual deployment steps by automating the entire process from code changes to deployment.
 - **Efficiency:** Improved deployment efficiency and reduced downtime by using CodePipeline and CodeDeploy.
 - **Monitoring:** Enhanced visibility into the deployment process and pipeline performance with CloudWatch metrics and alarms.
 - **Security:** Ensured secure access between EC2 and S3, and managed deployment permissions using IAM roles and policies.

Education

Bachelor of engineering in Computer Science

[M.A.M school of engineering] | [Trichy] | [2019-2023]

CGPA:7.31%

HSC

[Kshatriya Nadar higher secondary school] [kamuthi] [2018-2019] – 66%

SSLC

[Kshatriya Nadar higher secondary school] [kamuthi] [2016-2017] – 80%