

# BANSIKUMAR MENDAPARA

San Diego, CA | (908) 608 4326 | [bansimendapara53@gmail.com](mailto:bansimendapara53@gmail.com)

[linkedin.com/in/bansi-mendapara](https://www.linkedin.com/in/bansi-mendapara) | [bansimendapara.com](https://bansimendapara.com) | AWS Community Builder

## EDUCATION

### Master of Science - Computer Science

San Diego State University, San Diego, CA

May 2021

GPA: 3.85

### Bachelor of Engineering - Information Technology

Gujarat Technological University, Ahmedabad, India

June 2019

GPA: 3.77

## SKILLS

Programming Languages:	Python, Java, C, C++
AWS:	EC2, S3, RDS, Lambda, CloudFront, VPC, CloudFormation, IAM
DevOps:	Git, Chef, Ansible, Docker, Kubernetes, Terraform
Database:	MySQL, SQL, Oracle
Platform:	Linux, Windows
Web Technologies:	JavaScript, Bootstrap, CSS, HTML

## WORK EXPERIENCE

### DevOps Engineer Intern, Vee Ventures (New York, US)

April 2021 - May 2021

- Designed **CloudFormation** template to create infrastructure which reduces deployment time by 30%
- Automated cloud provisioning with **infrastructure as code** which decreases manual work by 50%
- Improved performance by 7% by reducing unnecessary redundancies and improving the backend
- Achieved 40% cost reduction by migrating to a **serverless** approach
- Maintained **Continuous Integration/Continuous Delivery** systems leading to an improvement in the time to ship new feature release by 24%

## PROJECTS

### [Event-Driven Python on AWS](#) | AWS Lambda, Amazon RDS, CloudWatch Event, AWS CloudFormation

- Performed data manipulation using Lambda function and load data into **RDS** PostgreSQL database
- Configured once-daily **CloudWatch event rule** to trigger Lambda function and notify using SNS
- Converted this infrastructure into code using CloudFormation YAML file
- Built **CI/CD pipeline** using GitHub actions and visualized data using **QuickSight**

### [Cloud Portfolio/Resume](#) | Amazon S3, Amazon Route53, AWS API Gateway, Amazon CloudFront

- Deployed static website using S3 and implemented HTTPS and OAI using CloudFront
- Created a public hosted zone in Route53 to route the requests to CloudFront distribution
- Managed DynamoDB, Lambda and **API Gateway** to store, update and retrieve visitor counter
- Provisioned architecture using **AWS SAM** and set up CI/CD pipeline with GitHub actions

### Serverless Image Processing | Step Function, S3, Lambda, API Gateway, Rekognition

- Initiated face collection in **Rekognition** to store indexed face images as searchable vectors
- Launched multiple lambda functions for facial detection, validation, indexing and thumbnail generation and used step functions to sequence these lambda functions
- Enabled S3 events to start a **state machine** execution

### Highly-Available Dynamic Site-to-Site VPN | AWS Transit Gateway, AWS Site-to-Site VPN, BGP

- Created the AWS and simulated On-premises environments and 2 customer gateway objects
- Established 2 **VPN** connections using transit gateway VPN attachment
- Configured 4 IPSEC tunnels between 2 On-premises and added **BGP** capability using strongSwan and FRR

## CERTIFICATION

- [AWS Certified Developer Associate](#)
- [AWS Certified Solutions Architect Associate](#)
- [HashiCorp Certified: Terraform Associate](#)