Pranay Singh Parihar

PHONE+917568591982•E-MAIL: PRANAYSPARIHAR@gmail. com

Work Experience

JP Morgan Chase & Co

May2020

Software Engineering Intern

- I was responsible for optimizing and troubleshooting a website.
- I had to get familiar with JPMorgan Chase frameworks and tools
- Understanding the concepts of git was also part of the job.
- Display data visually for traders (Trader's Dashboard)
- Use JP Morgan's perspective framework.

Deloitte

May2020

Technology Consultant Internship

- Understanding Cloud Computing
- CloudFeasibilityAssessment
- CloudReadinessAssessment
- Client Discovery
- Design a Business Case
- Considerations for Mobilization
- Define the project approach
- Conduct a market scan
- Further analysis & solution presentation

Tivona Global

October 2020

DevOps Intern

- Continuous deployment automation using bash scripting.
- Automated AWS cloud deployment using Terraform from scratch.
- Implemented terraform functions on Terraform cloud.
- Implemented sentinel policies on Terraform cloud.
- Using dynamic blocks to dynamically create multiple resources of block within a resource from a complex value such as a list of map.
- Create IAM policies for users.
- Lambda function to attach the IAM policy to an IAM user and save the CloudTrail logs for IAM user in DynamoDB, using Boto.

Education

Electrical and Electronics Engineering, Vellore Institute Of Technology

2017 - 2021

- Major in **Electrical and Electronics Engineering,** School of Electrical Engineering-coursework
- included electrical machines, Control Systems, and Signals and Systems.

Certifications

Amazon Web Services

Certified Cloud Practitioner

- Define what the AWS Cloud is and the basic global infrastructure
- Describe basic AWS Cloud architectural principles
- Describe the AWS Cloud value proposition
- Describe key services on the AWS platform and their common use cases (for example, compute and analytics)
- Describe basic security and compliance aspects of the AWS platform and the shared security model
- Define the billing, account management, and pricing models
- Identify sources of documentation or technical assistance (for example, whitepapers or support tickets)
- Describe basic/core characteristics of deploying and operating in the AWSCloud

IBM

Istio and IBM Cloud Kubernetes Service

- Configure Istio to receive telemetry data
- Istio Ingress controller
- Perform A/B testing with Istio
- Set up Istio Certificate Authority (CA)

Aviatrix

Aviatrix Certified Engineer

- Multi-Cloud Networking Architecture (MCNA)
- Multi-Cloud Connectivity
- Egress Filtering
- Network Automation

GitLab

GitLab 101

- GitLab CI/CD
- GitLab Collaboration
- GitLab Mergify

Projects

Portfolio Website with AWS Amplify, Lambda and Terraform

- **AWS Amplify:** This is where my static website is hosted and where all of your HTML, CSS, JavaScript and assets will live.
- **API Gateway**: My AWS Amplify website will make an API call when a form is processed and when this call is made to API Gateway, it will trigger a Lambda function.
- **Lambda**: The Lambda function can do whatever you want but in our case, it simply sends the data from the form to an email address using AWS Simple Email Service (SES).
- **Terraform:** Terraform will be used to automate the process of creating lambda functions The `AWS::Lambda::Function` resource creates a Lambda function. To create a function, you need a deployment package and an execution role. The deployment package contains your function code. The execution role grants the function permission to use AWS services, such as Amazon CloudWatch Logs for log streaming and AWS X-Ray for request tracing. VCS Integration was done using BitBucket. GitOps workflow was used for CI/CD pipeline. Sentinel Check was used to setup resource setup policy.

Serverless App with S3, Lambda and Terraform

- **Simple Storage Service(S3):** This is where my static website is hosted and where all of my HTML, CSS, JavaScript and assets will live.
- **API Gateway**: Your S3 website will make an API call when a form is processed and when this call is made to API Gateway, it will trigger a Lambda function.
- **Lambda**: The Lambda function can do whatever you want but in our case, it simply sends the data from the form to an email address using AWS Simple Email Service (SES).
- **Terraform:** Terraform will be used to automate the process of creating lambda functions The `AWS::Lambda::Function` resource creates a Lambda function. To create a function, you need a deployment package and an execution role. The deployment package contains your function code. The execution role grants the function permission to use AWS services, such as Amazon CloudWatch Logs for log streaming and AWS X-Ray for request tracing.

AWS Serverless Ecommerce Platform

Communication/Messaging:

- AWS AppSync **for interactions between users and the ecommerce platform.**
- Amazon API Gateway **for service-to-service synchronous communication (request/response).**
- Amazon EventBridge **for service-to-service asynchronous communication (emitting and reacting to events).**

Authentication/Authorization:

- Amazon Cognito **for managing and authenticating users, and providing JSON web tokens used by services.**
- AWS Identity and Access Management **for service-to-service authorization, either between microservices (e.g. authorize to call an Amazon API Gateway REST endpoint), or within a microservice (e.g. granting a Lambda function the permission to read from a DynamoDB table).**

Compute:

- AWS Lambda **as serverless compute either behind APIs or to react to asynchronous events.**

Storage:

- Amazon DynamoDB **as a scalable NoSQL database for persisting information.**

CI/CD:

- Terraform **with** AWS Serverless Application Model **for defining AWS resources as code in most services.**
- AWS Cloud Development Kit (CDK) **for defining AWS resources as code in the payment-3p service.**
- Amazon CodeCommit **as a repository to trigger the CI/CD pipeline.**
- Amazon CodeBuild **for building artifacts for microservices and running tests.**
- Amazon CodePipeline **for orchestrating the CI/CD pipeline to production.**

Monitoring:

- **Amazon CloudWatch for metrics, dashboards, log aggregation.**
- **AWS X-Ray for tracing across AWS services and across microservices.**

Achievements

My course got accepted and published on Udemy

Udemy

Udemy Instructor

- I authored practice questions for the Terraform associate exam.
- Second largest question bank for practice available online.
- My course got recognized as tier-1 course on the Udemy platform.

Technical Skills

- **Programming Languages –** HTML, Scripting, Python, JavaScript, Bash, AWS, Azure,CSS, MongoDB, NoSQL, SQL, Docker, Bash, Terraform, Google Cloud Platform
- **Software -** Adobe Photoshop, Adobe Premier Pro, MS Excel, MS PowerPoint, Docker

English Language Proficiency

EF Standard English Test (EF SET)

78% Overall

- Listening 80 - Reading 75
- CEFR Rank- C2(Proficient)