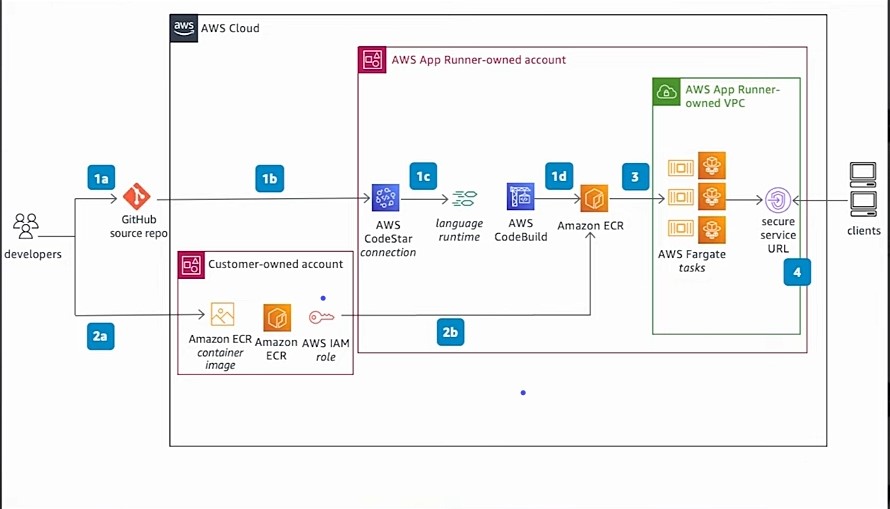
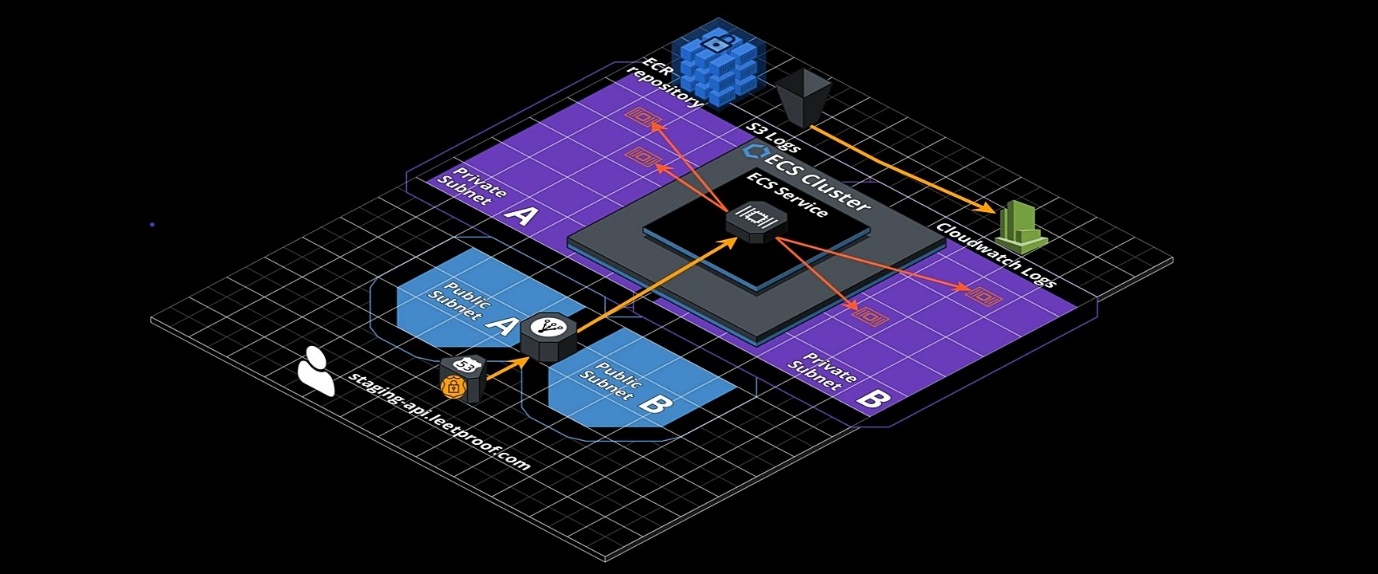
**CLOUD ENGINEERING AND DevOps SHORT ROAD MAP**

In our road map we will be combining cloud engineering and DevOps in our projects. After the projects you will be equipped with required skills to be employed as junior cloud or DevOps engineer. We will need these three steps to know what concepts fits where:

|  |  |  |
| --- | --- | --- |
| INFRASTRUCTURE | AUTOMATION | MONITORING |
| SOFTEWARE DEVELOPER | DevOps ENGINEER | CLOUD ENGINEER |
| Creating software using different infrastructure can be physical or on cloud. | Automating the process using different pipelines and CI/CD mechanism. | Offers services, platform and infrastructure to create, store and monitor the software. |





|  |  |
| --- | --- |
| Project |  |
| Amazon S3 project: Deploy a static website on AWS using Amazon S3 | * Use HTTPs for security (Amazon CloudFront), use Amazon Route 53 for DNS, Amazon DynamoDB, AWS API Gate away, Lambda, python(boto3), Certificate manager * Simple 2-tier, 3-tier calculator App. * Tools: AWS Amplify – used to build and host websites. Lambda- code that runs (serverless) upon trigger. API Gateway- used to build APIs. DynamoDB – NoSQL Database. IAM – creates security to the Lambda. * Use lift and shift strategy |
| Infrastructure as Code (Iac) project: Deploy a static website on AWS using AWS Serverless Application Model (SAM) template and deploy them using AWS SAM CLI | * Use the previous projects examples. * Tools: Terraform or SAM * Using past/new projects, finish this “ [https://cloudresumechallenge.dev/docs/the-challenge/aws/](https://cloudresumechallenge.dev/docs/the-challenge/aws/%20) “challenge steps and you will be ready for the next phase * Security: Firewall, Reverse Proxy and Forward Proxy * Version control: Git * Master Linux: Linux admin, ssh, ftp |
| EC2 project: Deploy a Dynamic Web App on AWS and EC2 | * Visualize Data with Amazon Quicksight * Integrate Amazon Lex Chatbot with Facebook Messenger * Fortune Teller Application * Search on AWS “advanced module” * GitHub: 5 Mini AWS Cloud Project Ideas * Serverless Web Application on AWS |
| Docker Project: Deploy a Dynamic Web application on AWS using Docker, ECR, and ECS Fargate | * GitHub: WillButton/trustified-api * Containerization, Image, Ps, ECR Registry, ECS Fargate Service in ECS Cluster * Networking: Subnets, ssl * Load balancers * Create DNS in Route53 * Logging Facility |
| Kubernetes project: Deploy a Dynamic Web application on AWS using Docker, ECR, and Kubernetes | * GitHub: N4si/cloud-native-monitoring-app. Tools: Boto3, Docker, KubeCtl, AWS account, Python3(Preferred language), ECR, EKS * Container Orchestration * Docker Swarm or Open Shift can be used instead * Architecture, Pods, Manifest, Deployment, Replicas * Microservices |
| Terraform Project: use Terraform To deploy the Dynamic Web application from EC2, Docker and Kubernetes Projects above | * Infrastructure provision * IaC |
| Ansible Project: Deploy a web App on AWS using Ansible | * Configuration management * Puppet or Chef can be used instead. * Service, Shell and Debug |
| Python Project: Use Python to Automate AWS-related Tasks and Solutions | * Python as scripting language |
| CI/CD Project: Deploy a Dynamic Web application on AWS using a CI/CD Pipeline then implement Testing, Monitoring and Logging | * GitHub: Jenkins-Zero-To-Hero, Azure-Samples/contoso-real-estate * Site: <chtrmbl.github.io> * Tools: Cypress for testing, S3, AWS X-Ray, CloudWatch, Git, Jenkins, Splunk used for Monitoring, GitLab Ci, Docker.io, Minicube, AlgoCd, Kubernetes , GitHub * Launch EC2 Instance * Jenkins: install (Docker Plugin attached with Maven, SonerQube Scanner Plugin) * Site: <operatorhub.io> * EC2 lab offers an environment for K8 (Kubernetes) |
| Re-factoring projects: Migrate existing application to new architecture | * Stack (Frontend): VM for Tomcat (App Server), Application load balancer, Autoscaling for VM scaling, S3 for storage. * Stack (Backend): RDS instance, Active MQ, Elastic Cache, Route53, CloudFront. |
| Networking Concept | * VPN, Public & Private DNS, NAT, Proxy, Firewall, Static/Dynamic Ips, CIDR Notations, Public/Private Network, Subnets * Protocols: http, https, ftp, tcp, ssl/tls, ssh, smtp |
| Resume | * Often upload your work to LinkedIn, GitHub, personal blogs. Check out:   Site: <adityacprtm.dev>, GitHub: Mo ROBLE, Site: <marikabergman.com> |