**Use case#1:  Implement Continuous Integration (CI) and Continuous Deployment (CD) for spring based micro services.**

·         Create Jenkin server

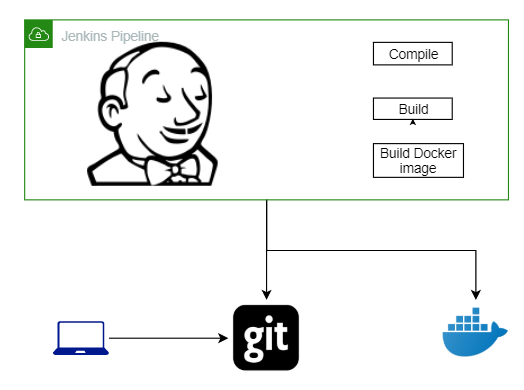
·         Create Kubernetes cluster

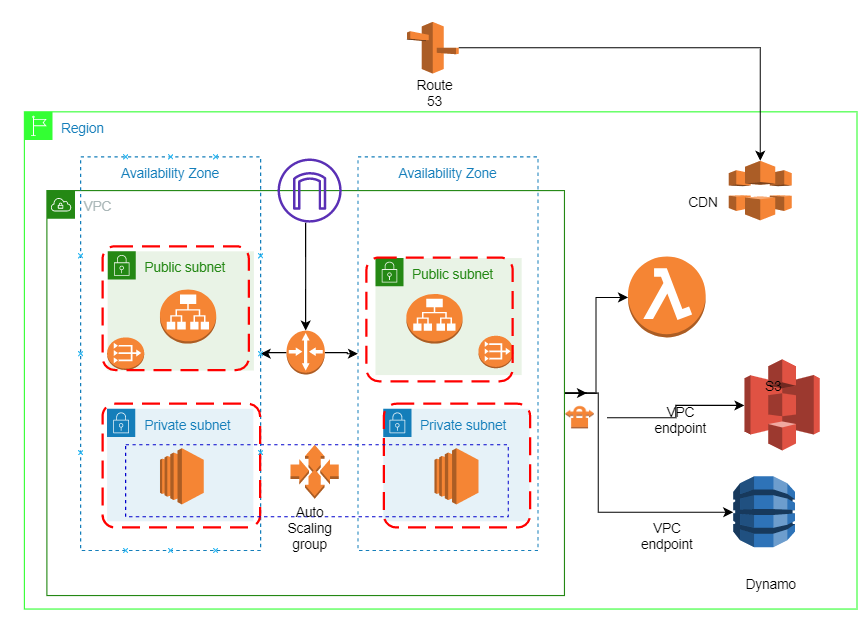
·         Create Jenkins file

·         Using pipeline create docker image

·         Upload to docker website

·         Deploy to Kubernetes.





Solution:

              Stage 1#:

CI -> Jenkins pipeline

1. Get microservice app artifact/s from Git
2. Compile, build & package
3. Create docker image and pushed to docker hub

CD -> AWS Infra provision

1. Get docker image from docker hub
2. Deploy into AWS EC2 and EC2 should be configured under VPC along with public & private subnets
3. Run the application
4. Access the application via public network.

Stage 2#:

              High availability & fault tolerance ->

1. Attach autoscaling & load balancer facilities to above step#5 activity.
2. Configure domain via Route 53
3. Access the application via public domain network

Stage 3#:

              S3 & Dynamo DB integration ->

1. Implement DynamoDB integration in our app and get & push data.
2. Implement a task to upload files to S3 via app
3. App should be accessible via public domain network and support above 11 & 12 tasks.

Stage 4#:

              Integration with SQS and Lambda ->

1. Lambda should be triggered as soon as new file to be added in S3  and push payload to SQS (ext. : step #12)
2. Read the payload from SQS via our own app and add an entry into DynamoDB.

**Use case#2:  Implement Complete serverless application in AWS services.**

Solution:

1. Implement REST services in Lambda via python boto3
2. Integrate with API gateway
3. Create website in S3 and configure above REST services with website.
4. Integrate CloudFront (CDN) for static data like images which are available in website.
5. Data/events to be loaded into dynamo DB and SQS via website.
6. Configure REST endpoints with Route 53

