Questions :-SE1

1. Create a nginx docker image which will run local sites on port 8080.
2. Write a shell script to check where the file or directory is located in given location if it is not then it should create it with the given name.
3. Create S3 bucket via terraform/cloudformation which will have following properties
   A . versioning and replication.
   B. Block all public access
   C. Default encryption
   D. lifecycle policy for 10 days
4. Set up a job in Jenkins which will ask for a git tag as an input while executing it (for this create some tags on git branch).
5. Create ALB which will have path based routing. For example if we put the /test as url suffix (abc.com/test) it should redirect to a different page or server or if we put nothing it should have it's default page or server.

Questions :-SE2

1. Create a lambda function and deploy code to this function and expose it to API Gateway?
   (run terraform scripts without using a profile)
2. What would be the steps to mount an EFS with EC2 Instance?
   (Mount EFS to different instances and use it as a focal point to store the data)
3. What would be the recovery process of EC2 keys?
   (Change the EC2 Keys without removing that EBS volume)
4. Set up a VPC and launch machines in a private network, additionally private should have the internet connection?(Don't use internet gateway for private instance)
5. Overwrite the CMD process with ENTRYPOINT?
6. Setup a server and implement path based routing through the Load Balancing.
7. Create a pipeline to deploy code at EC2 instance.
8. Allow users to login to EC2 instances using their password.
9. Implement ECS and deploy containers through ECR.
   (use multistage dockerfile)

Questions :-SSE

1. Create a Jenkins pipeline with a slave node, if a branch is “dev” then it should use that node or else it should use master node.
2. Create a multi stage docker image for running a code(could be java,php or else) on nginx.
3. Write a terraform script which will have validation that if the resource name consists of the environment name “dev”,”tst” and “prod” in its name it should create it or else it should not. Create resources (for example s3 bucket) for all environments with a single resource script and make sure that the stat file should be different for all environments.
4. Create a kubernetes cluster to deploy a nginx container, and deploy code through jenkins at this cluster.
5. Create an assumed role and use it in the shell script to create an EC2 instance.
6. Create a shell script to put daily logs on S3 bucket and remove the logs in s3 older than 2 days.
7. Create a webpage(could be anything) with docker image and run it on ECS, it should be behind an ALB.
8. Create Jenkins pipeline to deploy a static site in AWS cloudfront and send build notification to slack channel. Pipeline should get executed on every push to the branch.