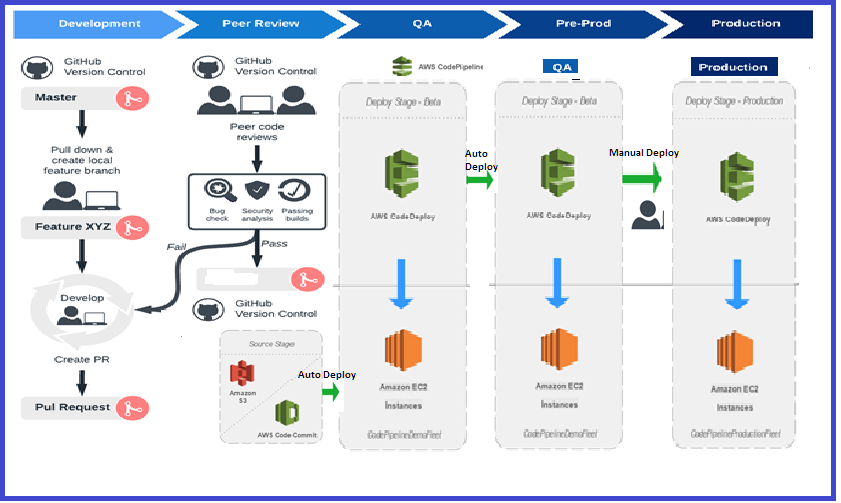
Q-

How would you ensure any change made to this Dockerfile is source controlled, approved, tested and deployed. Explain which tools you will use as if this was going into a production environment.

Explanation-

|  |  |
| --- | --- |
| Git, Github Repository : Version Control System tool | Amazon Route 53 for DNS Services |
| Jenkins : Continuous Integration tool | ELB for load balancing, Amazon ECS for cluster |
| Selenium : Continuous Testing tool | Amazon EC2 ,EBS to launch instances |
| Ansible Configuration Management and knowledge on Puppet | Identity Access Manager (IAM) To manage users/Groups |
| Nagios : Continuous Monitoring tool | AmazonEC2, ElasticBeanStackto launch instances |
| Docker, Kubernetees : Containerization tool | Amazon ECR for |
| Maven as Build Tool | Simple Storage Service (S3) for |
| Sonar Qube, Nexus Artifactory | Amazon CloudFront for content delivery, Edge Caching, Edge Security Amazon CloudSearch for search workloads, Amazon Simple Email Service (Amazon SES) for sending and receiving emails27  MangoDB, Amazon DynamoDB for NoSQL databases |



* + 1. Pull and create a local feature branch from Github repository
    2. Develop the feature and test locally
    3. Create a Pull request for review
    4. Based on the review results if Passed then merge to Git hub master repository
    5. To be fixed if any changes required as per review feedback and follow the process 3.
    6. Now code commit will trigger auto deployment on AWS for ETE (Dev) environment
    7. Sanity and functional testing will run if no cases fails then moved to next stage.
    8. Code will trigger auto deployment on AWS QA environment
    9. Sanity and functional testing will be performed if all good then will be planned for production release
    10. Manual process will follow to deploy in production environment.