Creating CI/CD pipelines for docker Images

Challenges:

We are creating the CI/CD pipelines which create the docker images using the Jenkins job and push that images inside the private docker repository or public repository. This private repository is either s3, bucket, artifactory or nexus repo and docker hub public repo . In order to build the CI/CD pipelines we are using the below tools.

1. Terraform
2. Packer
3. Jenkins
4. Git / Bitbucket
5. Nexus
6. Private repo. (S3, ECR,Nexus,Artifactory)
7. Docker hub

Use cases we solved with current pipelines:

1. Create CI/CD pipelines for the enterprise application where e2e flow is working fine. Developers can build , test and deploy code automatically..
2. Create the docker images using the Jenkins job which will push to s3 bucket or docker hub and using deploy job deploy on the swarm cluster.

Scope:

In order to keep the project scope simple currently we are building the docker images using Jenkins jobs and pushing it to docker hub repo. In order to implement the pipelines for enterprise application and data companies we are scanning the docker images before pushing it to private repos. Also in future we will deploy the same docker images using jenkins pipeline on swarm cluster and kubernetes cluster.

Pipeline workflow and its implementation:

Main goal of the pipelines is when developers push or commit the code in the scm repository that code should be compiled, build, tested and released in the production in an automated way. In the current case we are building the docker images using jenkins pipeline. Also this pipeline is used with other use cases where we need to create artifacts and push on the nexus.

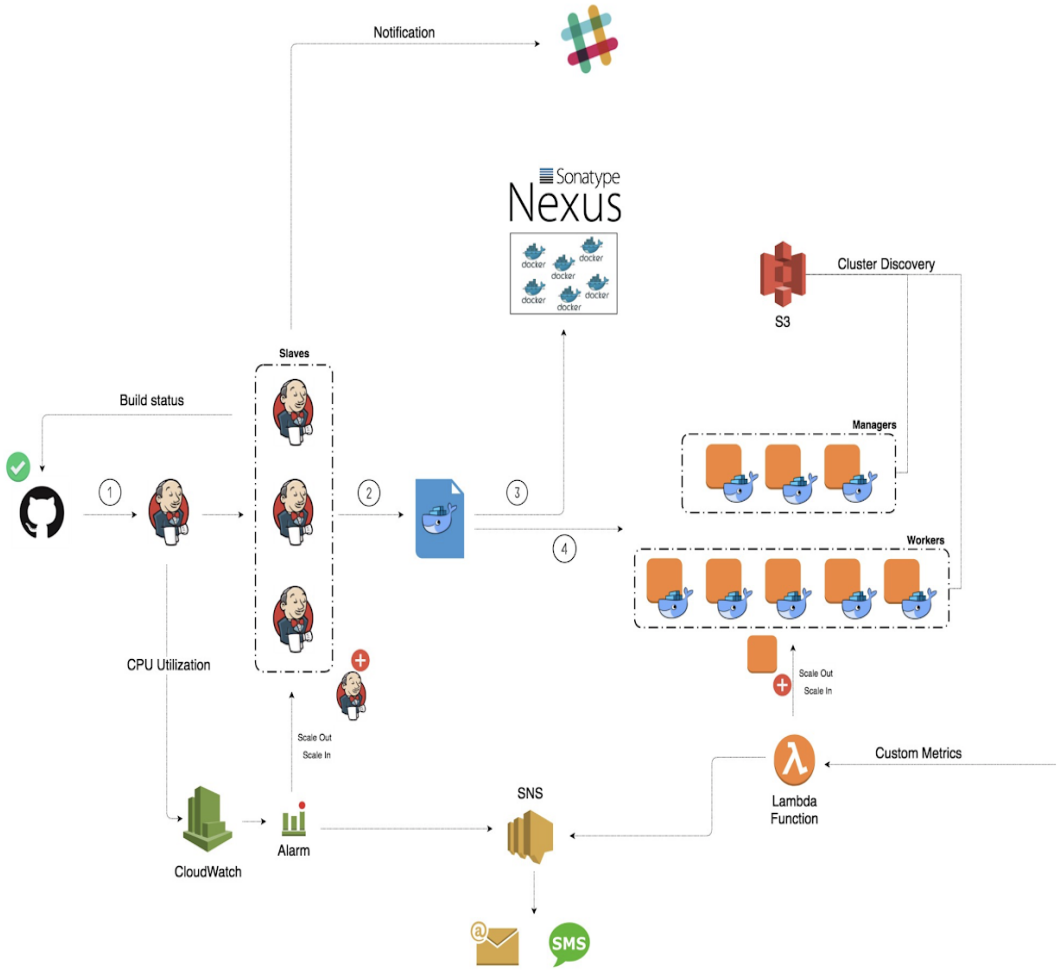
Implementation Details:

1. We are using AWS Cloud for implementing these pipelines.
2. We are using terraform in order to create AWS infra and doing Jenkins setup .
3. We are creating a Jenkins setup in master slave architecture in autoscaling.
4. Jenkins slave scale based on the load.
5. In order to create Jenkins master-slave we are creating pre bake AMI using packer.
6. In the current project we are building the docker images using the Jenkins pipeline job.
7. We will also create a Jenkins pipeline which will pull the code from git and build the respective artifact and push that either on an artifactory or targeted server.

Tools Used in Project:

1. Terraform 12.24.0 version
2. Packer Latest Version
3. Jenkins latest version
4. AWS-vault
5. Nexus and JDK as per the requirement.
6. Docker (19.03.8)

Platform Diagram:

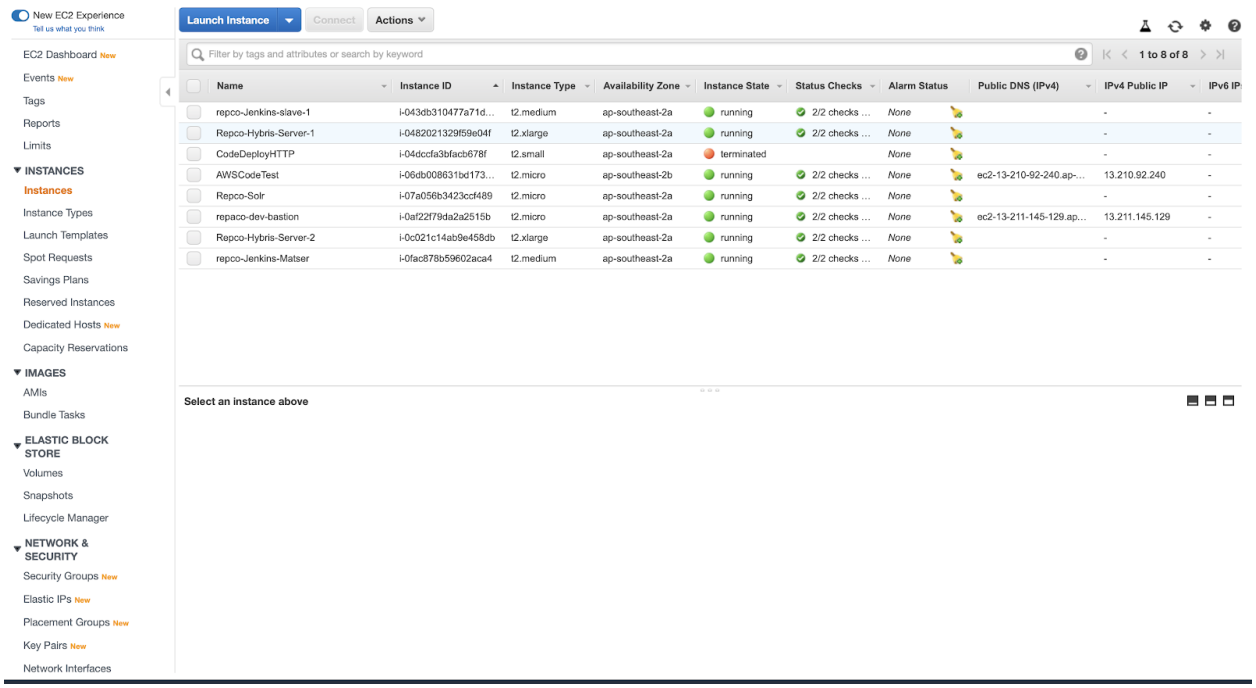


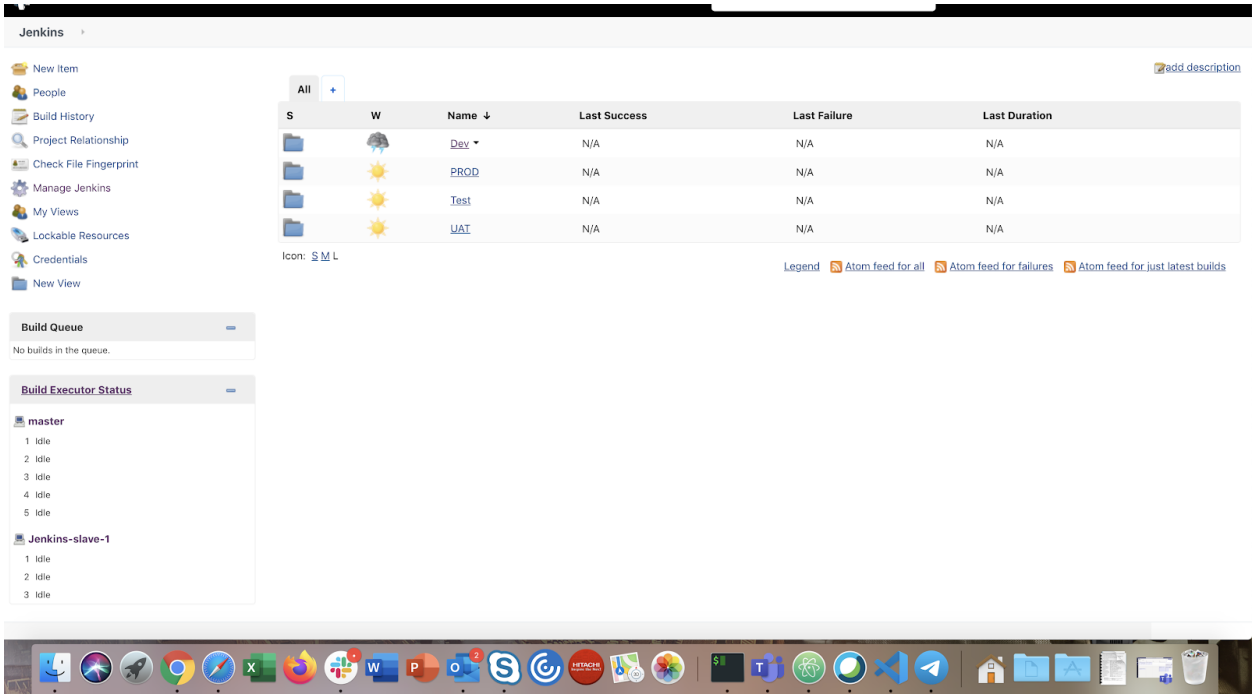
Project snippet:

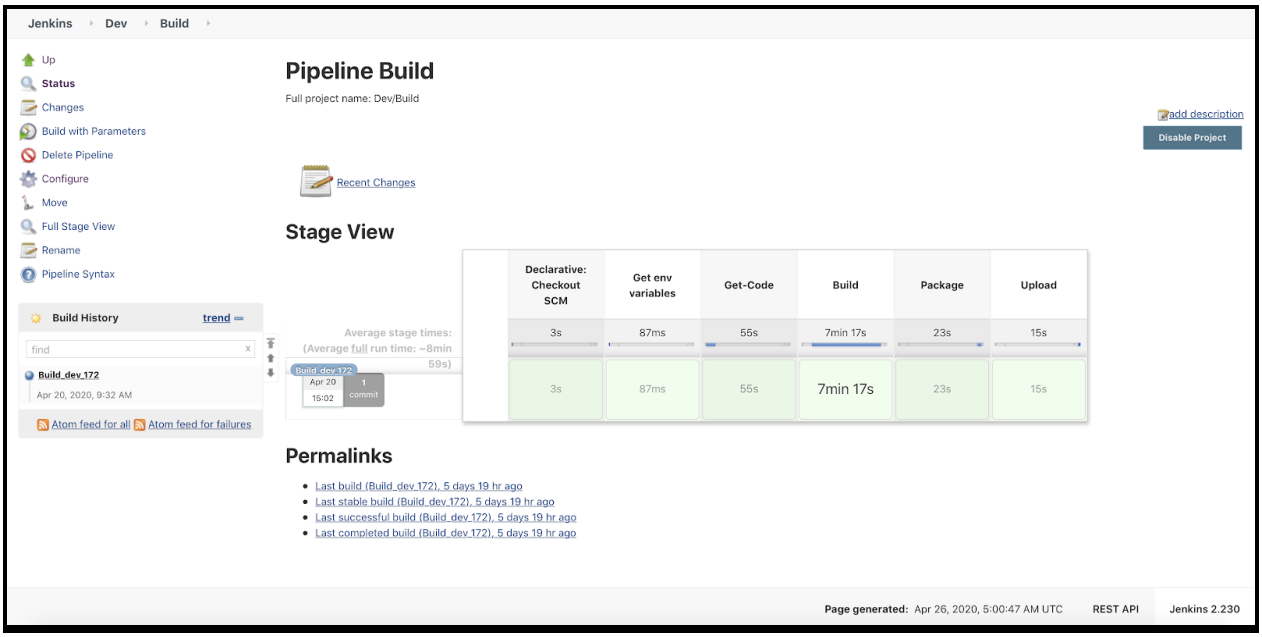
1. **Real time use case** “CI/CD for spring boot application”

   In this pipeline Java Enterprise application code is placed on the bitbucket. Using Declarative pipelines
   1. we are fetching that code
   2. compile it
   3. building the artifacts using ant
   4. pushing the artifacts inside s3 bucket
   5. Deploy the artifacts on the targeted cluster.

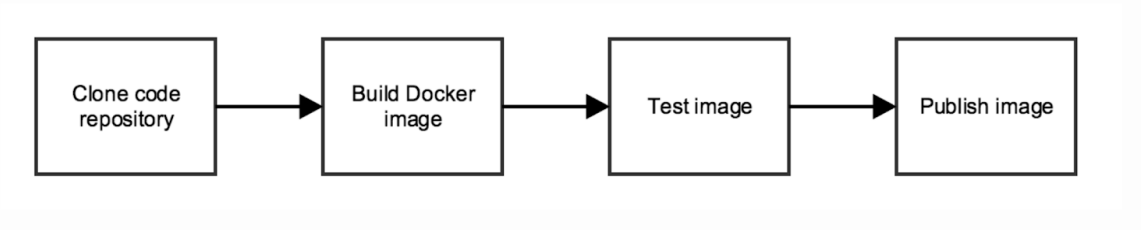
All this happens on a single click of jenkins job.





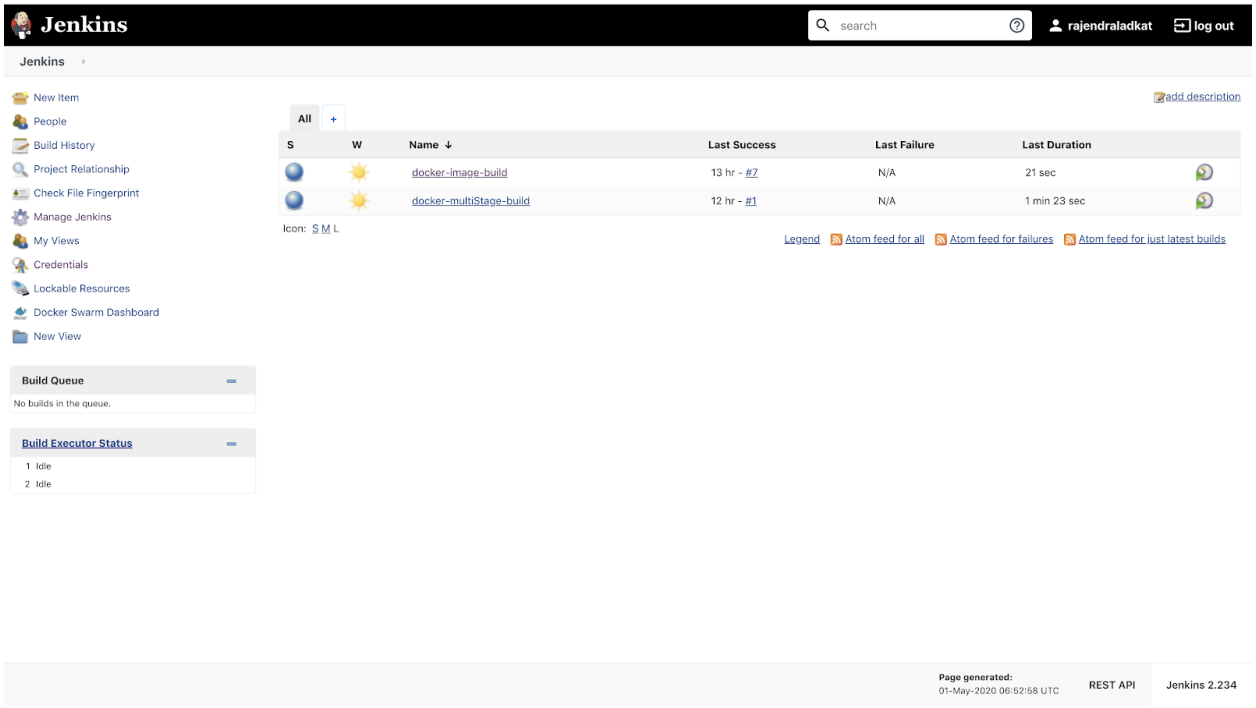


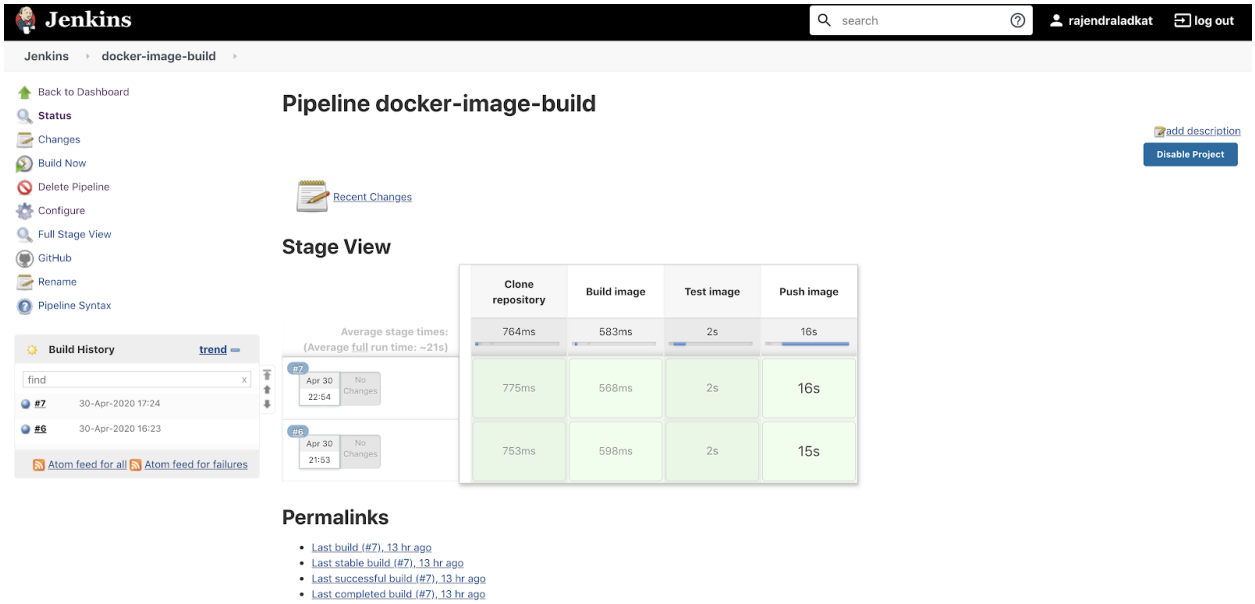
2. Real time use case “Building docker images using the Jenkins job”.

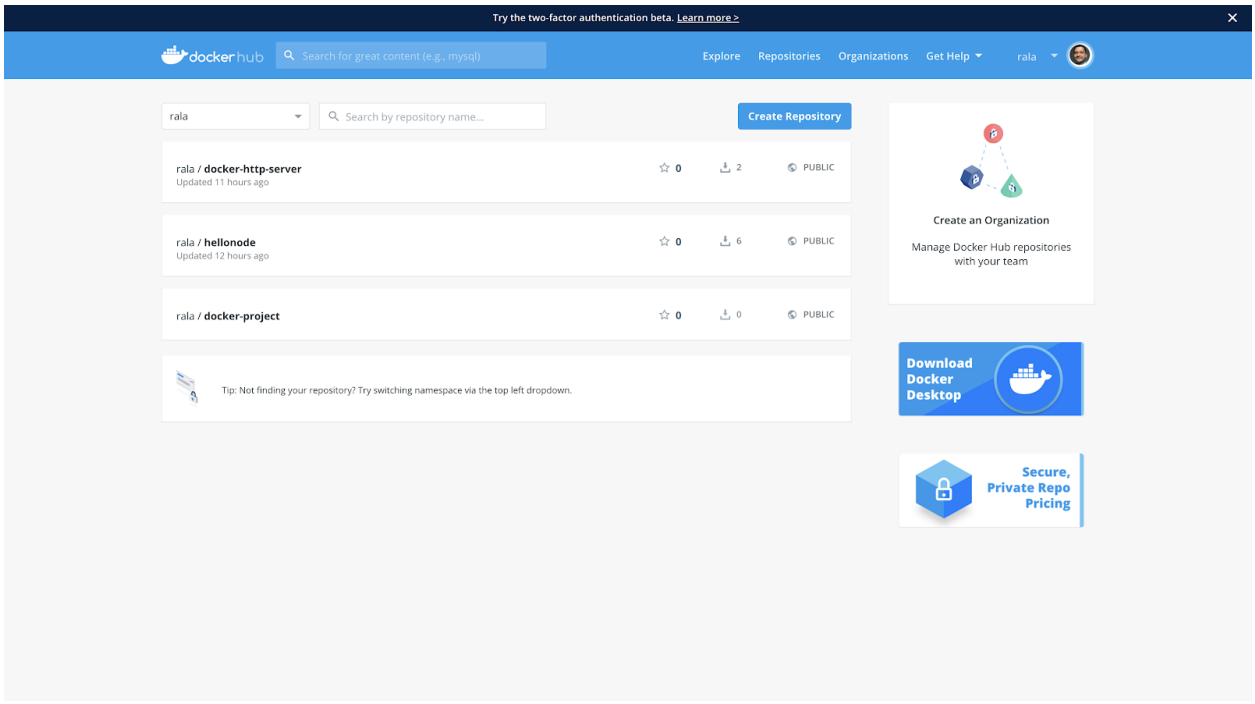


1. We have set up Jenkins on the EC2 instance in which docker,docker-compose and jenkins is installed.
2. We have set up jenkins with required docker plugins.
3. We have created the pipelines which we do the following things in multiple stages.

1. Get the code from git scm
   2. This git repository contain the Jenkins file as well as required other files along with dockerfile
   3. Code is fetch into jenkins workspace and using dockerfile docker images is build Then image is tested with customize test cases
4. Finally it will be pushed to docker hub repo.
5. We have created two Jenkins jobs.
   1. Single build
   2. Multi-stage-build.




Jenkins
search
rajendraladkat
log out
Jenkins
docker-image-build
Back to Dashboard
Status
Changes
Build Now
Delete Pipeline
Configure
Full Stage View
GitHub
Rename
Pipeline Syntax
Build History
trend
find
#7
30-Apr-2020 17:24
#6
30-Apr-2020 16:23
Atom feed for all
Atom feed for failures
Pipeline docker-image-build
add description
Disable Project
Recent Changes
Stage View
Average stage times:
(Average full run time: ~21s)
Clone repository
Build image
Test image
Push image
764ms
583ms
2s
16s
#7
Apr 30
22:54
No Changes
775ms
568ms
2s
16s
#6
Apr 30
21:53
No Changes
753ms
598ms
2s
15s
Permalinks
Last build (#7), 13 hr ago
Last stable build (#7), 13 hr ago
Last successful build (#7), 13 hr ago
Last completed build (#7), 13 hr ago


Try the two-factor authentication beta. Learn more >
dockerhub
Search for great content (e.g., mysql)
Explore
Repositories
Organizations
Get Help
rala
rala
Search by repository name...
Create Repository
rala / docker-http-server
Updated 11 hours ago
0
2
PUBLIC
rala / hellonode
Updated 12 hours ago
0
6
PUBLIC
rala / docker-project
0
0
PUBLIC
Tip: Not finding your repository? Try switching namespace via the top left dropdown.
Create an Organization
Manage Docker Hub repositories with your team
Download Docker Desktop
Secure, Private Repo Pricing

Important links:

Jenkins:

"http://ec2-13-210-70-195.ap-southeast-2.compute.amazonaws.com:8080"

Git hub:

1. https://github.com/Rajendraladkat1919/docker-multi-stage
2. https://github.com/Rajendraladkat1919/docker-demo
3. https://github.com/Rajendraladkat1919/ci-cd

Docker hub:

Image Name:

1. rala/docker-http-server
2. rala/hellonode

Pros:

- We are using open source tools like terraform, packer , docker , jenkins in order to deploy stack.
- It will save the cost
- Cloud agnostic
- Blue-green deployment is possible.
- We can deploy on any cloud
- Integrate with multiple tool like ansible and chef.

Cons:

- Some infrastructure cost for the pipelines.

Future scope:

- Cloud agnostic
- Implement with data intelligence for better resource utilization
- Can implement to test infra for security compliance.
- Deployment on K8s cluster
- Can integrate with multiple open source tool.
- Use for zero deployment.

References:

1. Terraform: https://www.terraform.io/intro/index.html
2. Packer : https://packer.io/intro/index.html
3. Jenkins: https://jenkins.io/doc/
4. Nexus : https://www.sonatype.com/product-nexus-repository
5. AWS-vault: https://github.com/99designs/aws-vault
6. Artifactory: https://jfrog.com/artifactory/
7. AWS cloud
8. Doc.docker.com
9. https://appfleet.com
10. https://releaseworksacademy.com/