Automated Application Deployment Using Jenkins, Docker, and AWS EC2

Objective:

- Check out the application source code from GitHub.
- Build a Docker image from the source code.
- Push the Docker image to Docker Hub.
- Connect to an AWS EC2 instance via SSH, pull the Docker image, and run the container.

Implementation Overview:

1. Created application files (index.html and style.css) on the local machine.
2. Wrote a Dockerfile to containerize the application.
3. Created a Jenkins pipeline script (deploy.groovy) for automation.
4. Pushed all files to a GitHub repository.
5. Configured Jenkins to automate the build, push, and deployment process.

Prerequisites:

Local Machine (Ubuntu with Jenkins and Docker)

a. Install Jenkins: to automate the CI/CD pipeline.

b. Install Docker: to build and run container images.

c. Configure Jenkins User for Docker: To allow Jenkins to run Docker commands without sudo-

```
sudo usermod -aG docker Jenkins && sudo systemctl restart Jenkins
```

d. Install Required Jenkins Plugins: Docker Pipeline, Docker, SSH Agent, Credentials Binding

e. Configure Jenkins Credentials: credentials were added to Jenkins-

- Docker Hub credentials (username and password)
- EC2 SSH private key (Base64 encoded)

f. Verify Jenkins to EC2 Connectivity: Ensure the EC2 security group allows inbound SSH (port 22) from the Jenkins server's IP address.

EC2 Instance (Ubuntu)

a. Install Docker: Docker is installed on the EC2 instance to run containers.

b. Start and Enable Docker: Docker is started and enabled to run on system boot.

c. Configure Docker Permissions: The ubuntu user is added to the Docker group-

```
sudo usermod -aG docker ubuntu
```

d. Open Required Ports: following ports are allowed in the EC2 security group:

SSH: 22

HTTP: 80

HTTPS: 443

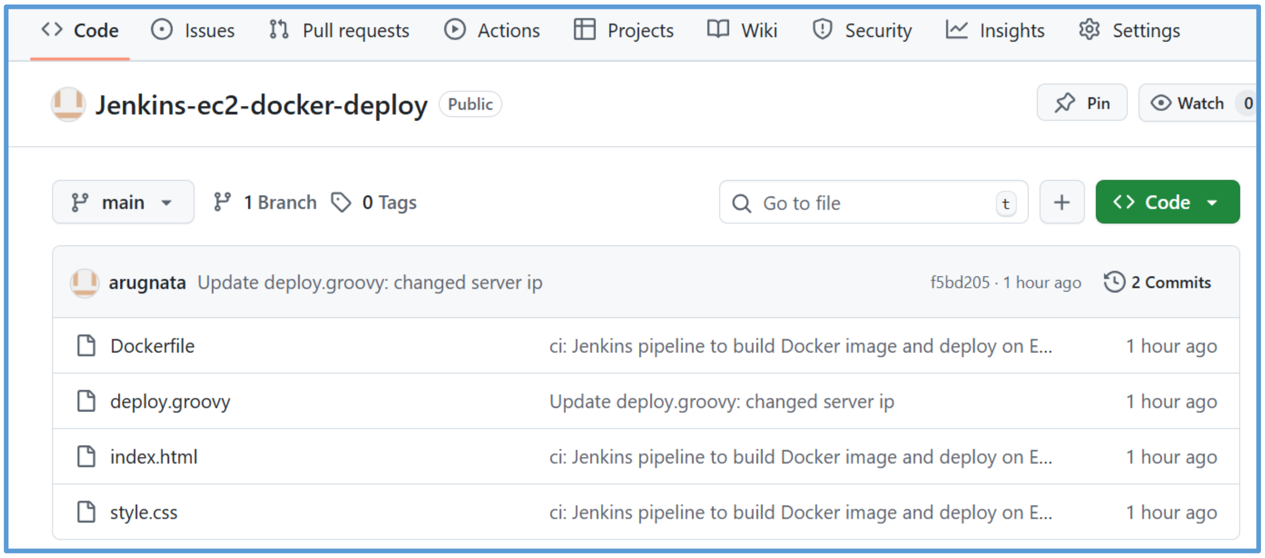
GitHub Repository

1. A GitHub repository was created containing the following files:
   - index.html
   - style.css
   - Dockerfile
   - deploy.groovy
   -
2. The repository was configured to be accessible by Jenkins (public or using credentials).

Jenkins Pipeline Configuration

- A new Jenkins Pipeline job was created.
- The pipeline was configured to fetch the pipeline script (deploy.groovy) from the GitHub repository.
- The pipeline automates:
  - Source code checkout
  - Docker image build
  - Image push to Docker Hub
  - Deployment on the EC2 instance via SSH

Proof of Concept:



Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

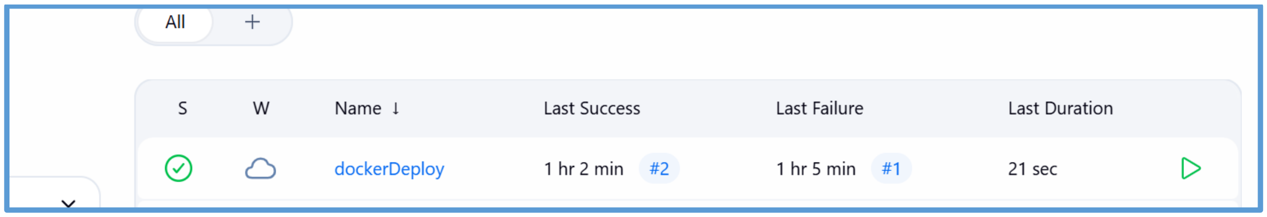
Jenkins-ec2-docker-deploy Public

main | 1 Branch | 0 Tags | Go to file | Code

arugnata Update deploy.groovy: changed server ip · f5bd205 · 1 hour ago · 2 Commits

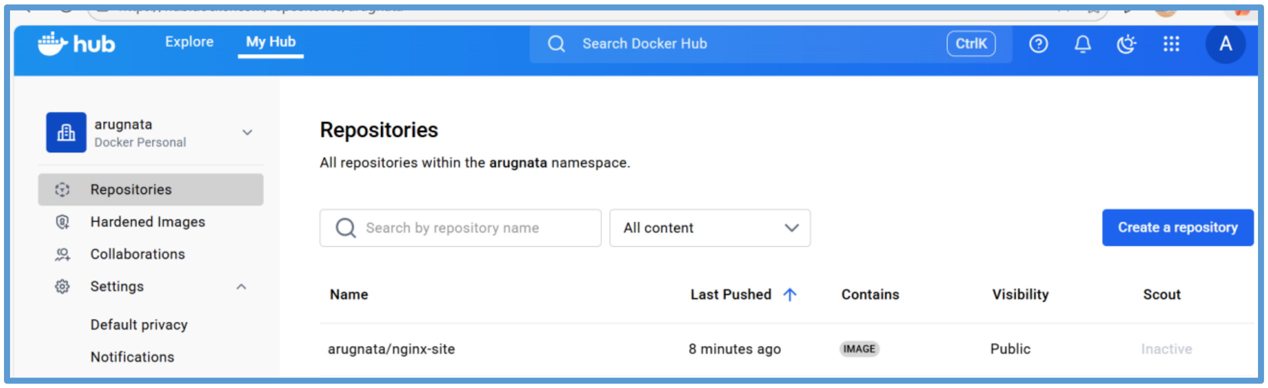
Dockerfile	ci: Jenkins pipeline to build Docker image and deploy on E...	1 hour ago
deploy.groovy	Update deploy.groovy: changed server ip	1 hour ago
index.html	ci: Jenkins pipeline to build Docker image and deploy on E...	1 hour ago
style.css	ci: Jenkins pipeline to build Docker image and deploy on E...	1 hour ago

Figure 1: Source code pushed to the GitHub repository.



S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☁	dockerDeploy	1 hr 2 min #2	1 hr 5 min #1	21 sec

Figure 2: Jenkins pipeline job creation and configuration.



hub | Explore | My Hub | Search Docker Hub

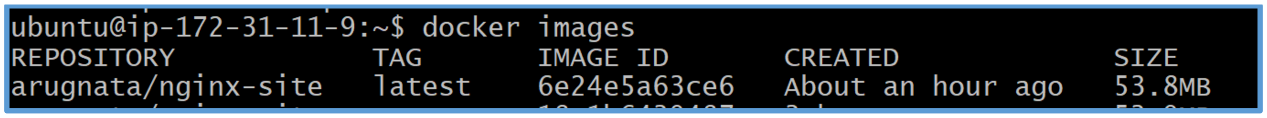
arugnata Docker Personal

Repositories

All repositories within the **arugnata** namespace.

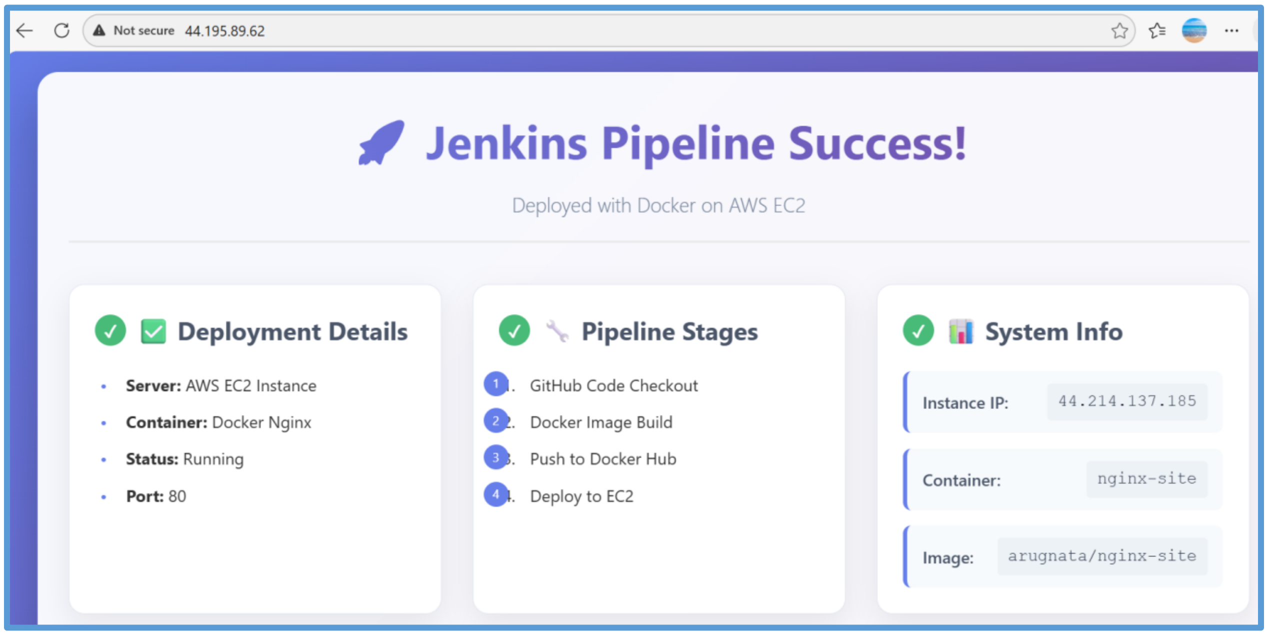
Name	Last Pushed	Contains	Visibility	Scout
arugnata/nginx-site	8 minutes ago	IMAGE	Public	Inactive

Figure 3: Docker image successfully pushed to Docker Hub.



```
ubuntu@ip-172-31-11-9:~$ docker images
REPOSITORY            TAG       IMAGE ID       CREATED              SIZE
arugnata/nginx-site   latest    6e24e5a63ce6   About an hour ago    53.8MB
```

Figure 4: Docker image pulled and container running on the EC2 instance.



Jenkins Pipeline Success!

Deployed with Docker on AWS EC2

Deployment Details

- **Server:** AWS EC2 Instance
- **Container:** Docker Nginx
- **Status:** Running
- **Port:** 80

Pipeline Stages

1. GitHub Code Checkout
2. Docker Image Build
3. Push to Docker Hub
4. Deploy to EC2

System Info

Instance IP:	44.214.137.185
Container:	nginx-site
Image:	arugnata/nginx-site

Figure 5: Successful execution of the Jenkins CI/CD pipeline.