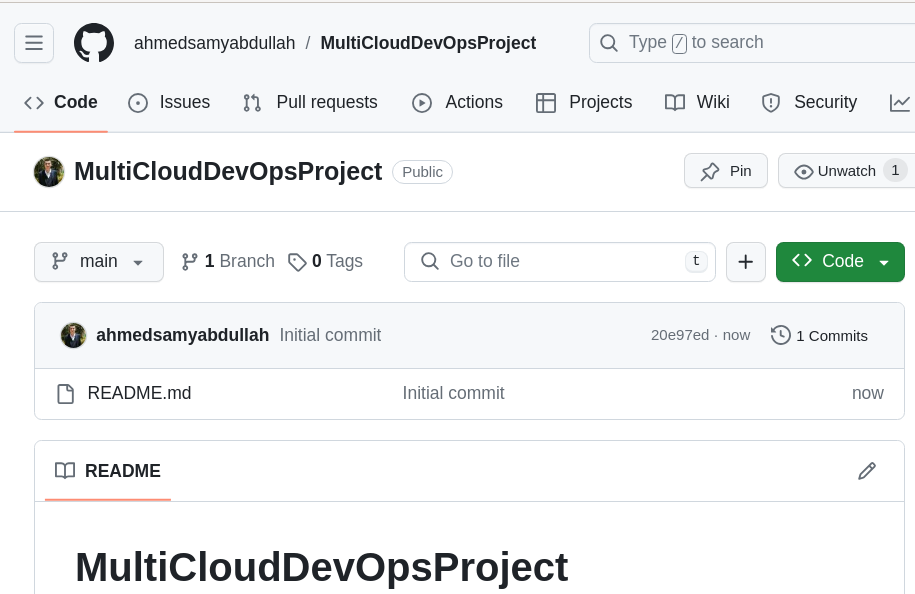
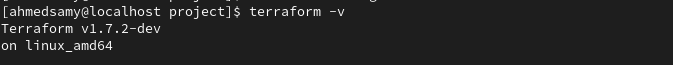
**Documentation of graduation project**

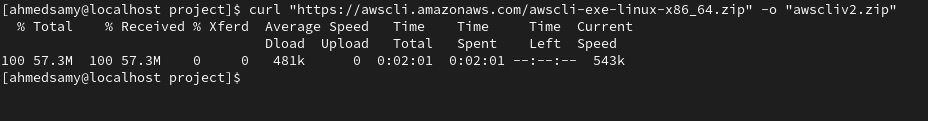
* Task GitHub Repository Setup:
* I created repository ( MultiCloudDevOpsProject ) with README.
* I created dev and main branches and push to dev banch :



* Task Infrastructure Provisioning with Terraform :
* Terraform modules : allows you to group resources together and reuse this group later.



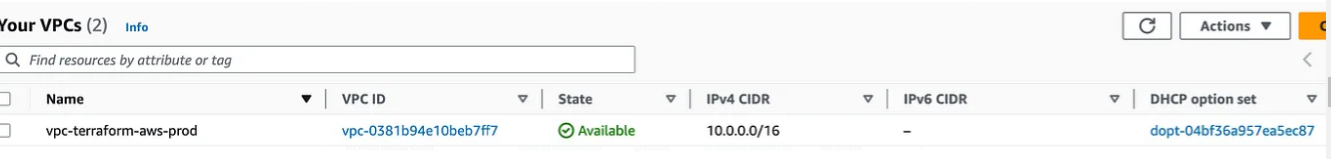
* Install aws cli



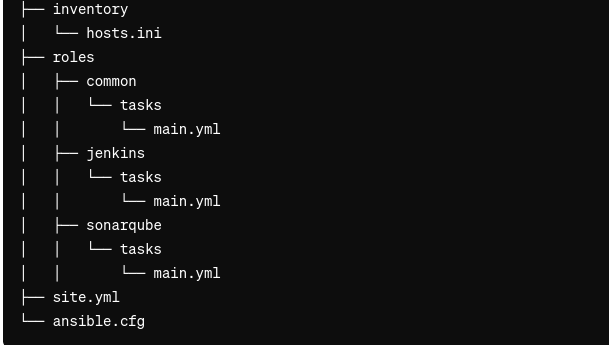
* Now run aws



* After creating vpc



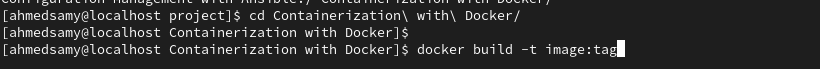
* After that I Using commands :
* Terraform init
* Terraform apply
* **Configuration Management with Ansible:**
* Ansible Vault to store credentials and to create it
* Ansible-vault create cred.yml => enter password
* We need boto to allow us to create EC2 instances
* pip install boto
* Install the role:
* ansible-galaxy install -r requirements.yml -p ./roles
* I use this Ansible structure



* At the end run ansible-playbook



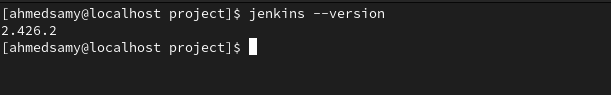
* **Containerization with Docker:**
* To build docker image



* After building the image, can run it with this command:

**docker run -p 8080:8080 image:tag**

* **Continuous Integration with Jenkins:**
* First check Jenkins is installed



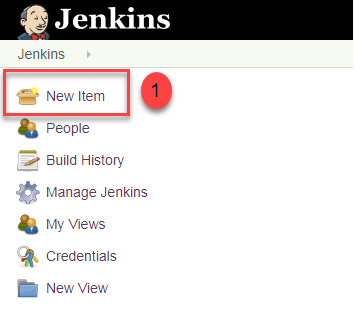
* **Install Jenkins using Docker**

****

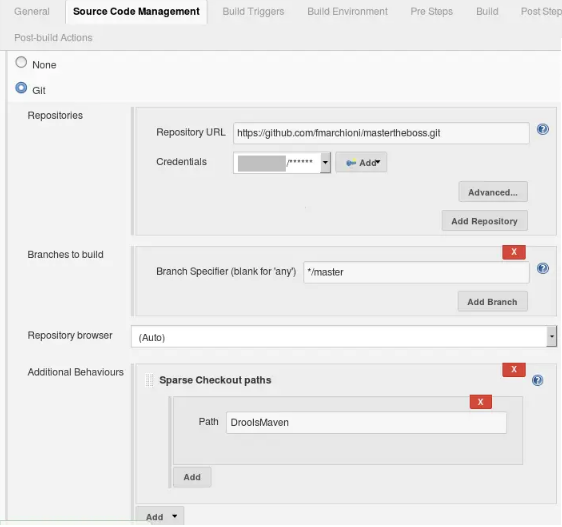


* Docker Plugin Installed: Install the "Docker" plugin in Jenkins. can do this by navigating to "Manage Jenkins" -> "Manage Plugins" -> "Available" tab, search for "Docker," and install the plugin.
* Jenkins Job Configuration Steps:

1. Create a New Jenkins Job and enter new item:



1. Configure Source Code Management:

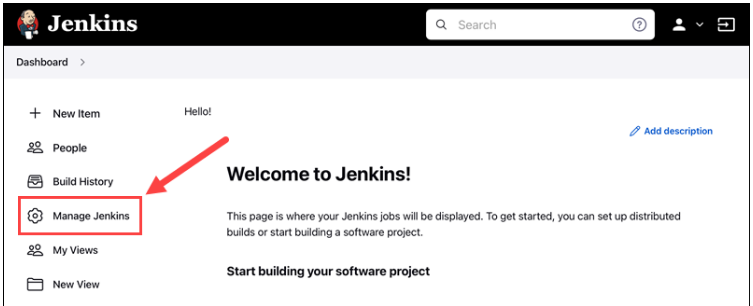


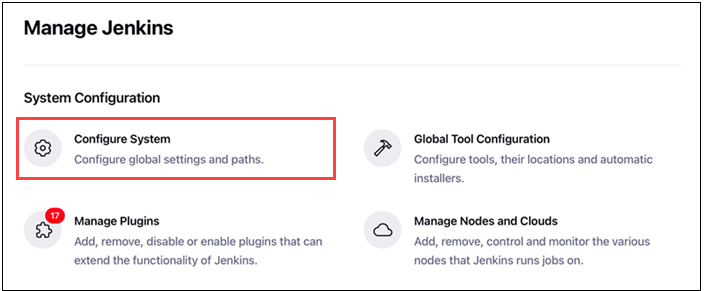
1. Configure Build Triggers
2. Configure Build Steps

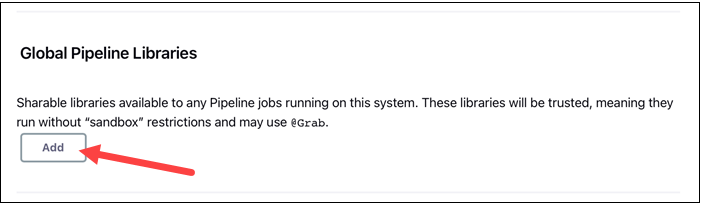


1. Configure Post-Build Actions
2. Save the Jenkins Job Configuration
3. Run the Jenkins Job Manually

* **Automated Deployment Pipeline:**
* What Is a Shared Library in Jenkins?
* A shared library in Jenkins is a collection of Groovy scripts shared between different Jenkins jobs. To run the scripts, they are pulled into a Jenkinsfile.
* Why Use a Jenkins Shared Library?
* Developers use shared libraries to avoid writing the same code from scratch for multiple projects
* How to Create a Shared Library in Jenkins?
* Step 1: Create a Groovy Script
* Step 2: Add the Script to a Git Repository
* Step 3: Add a Shared Library in Jenkins







* **Monitoring and Logging**
* First we need Access to an OpenShift Cluster



* Steps :

1. Deploy Elasticsearch Operator



1. Deploy Elasticsearch Cluster

* I created elastic-cluster.yml

1. Deploy Fluentd for Log Collection

* I created flunted.yml file

1. Deploy Kibana
   * I created kibana.yml
2. Access Kibana Dashboard:



* **AWS Integration**
* Steps

1. Set Up S3 Bucket for Terraform Backend:

Create an S3 bucket that will be used as the backend to store the Terraform state file.

1. Configure Terraform Backend in Main Configuration
2. Integrate CloudWatch for Monitoring
3. Install Terraform CloudWatch Provider:



1. Run terraform commands

