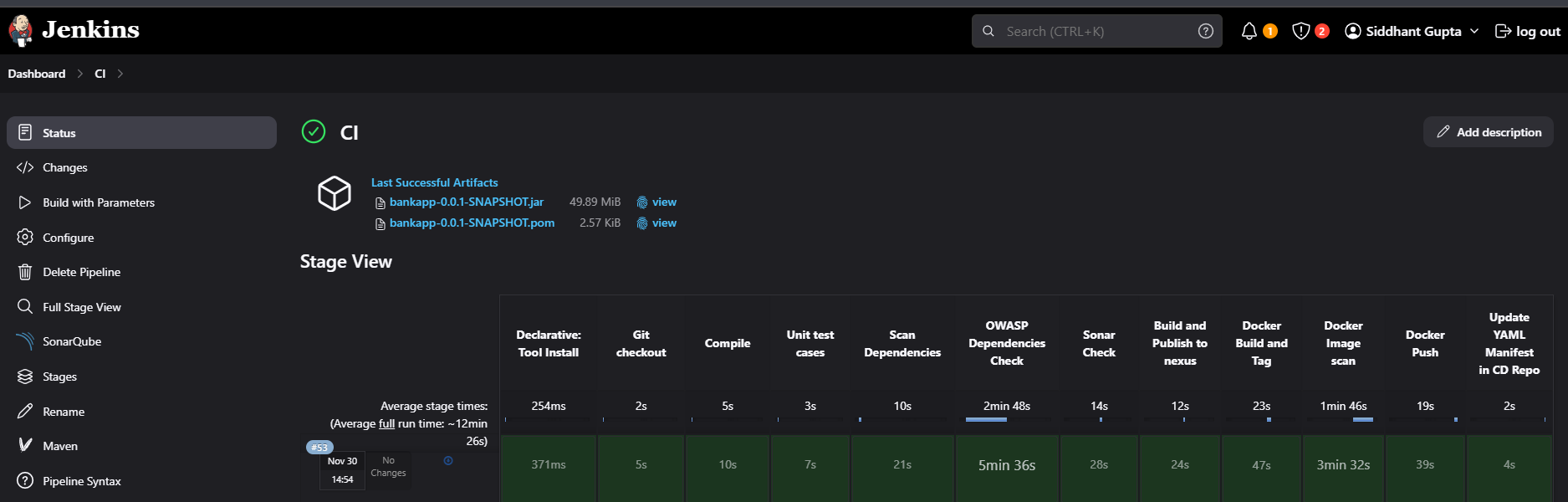
# Project Documentation

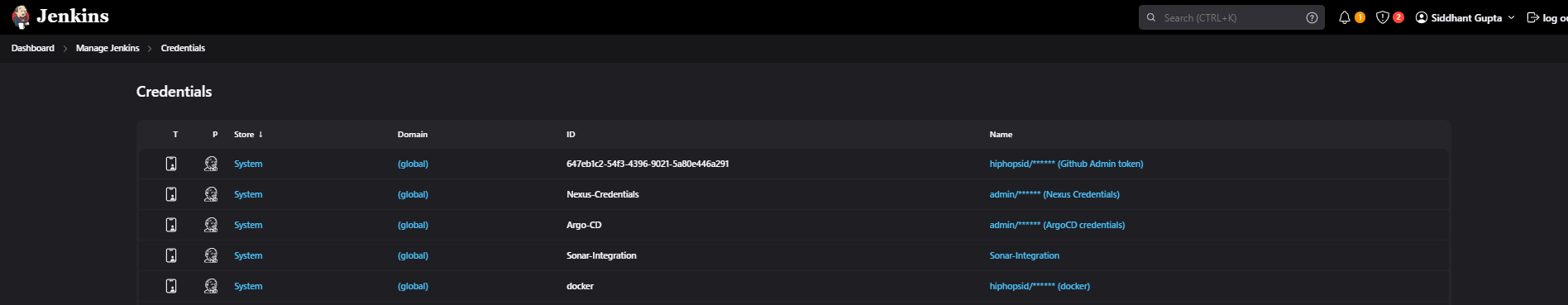
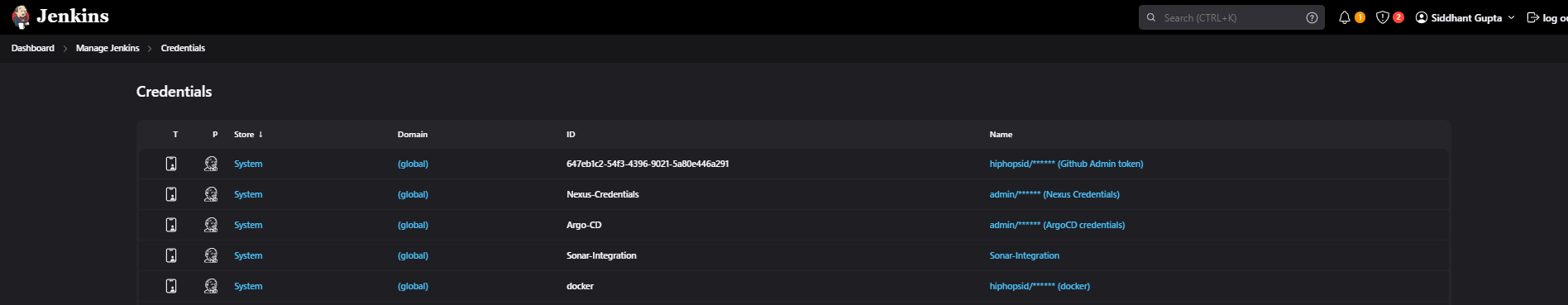
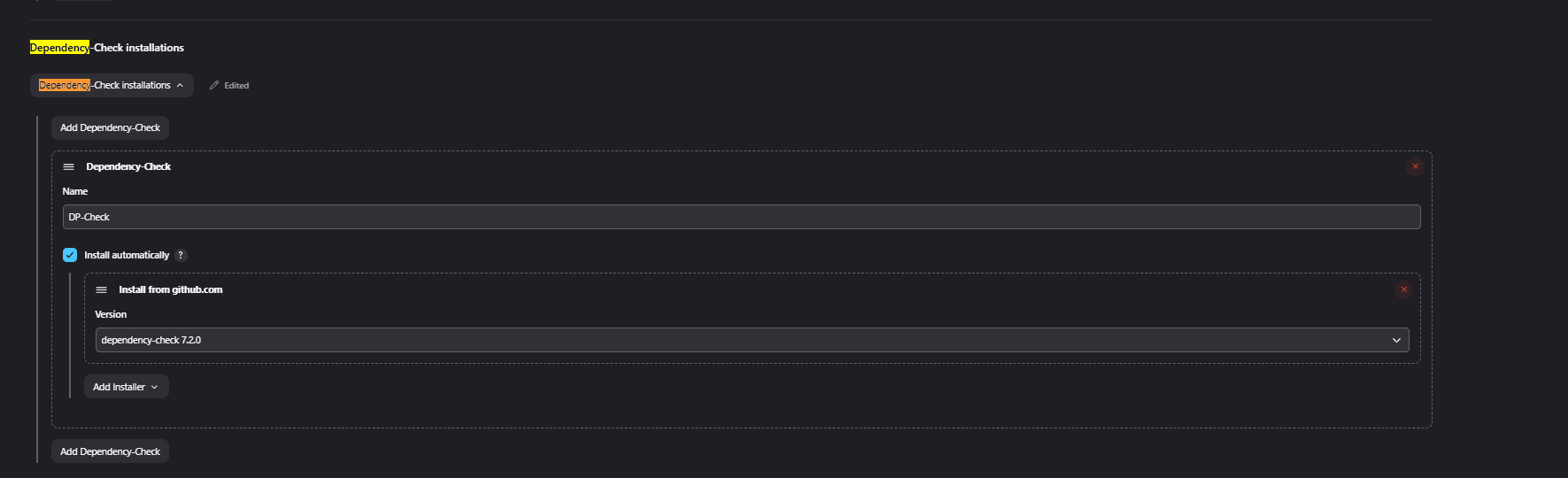


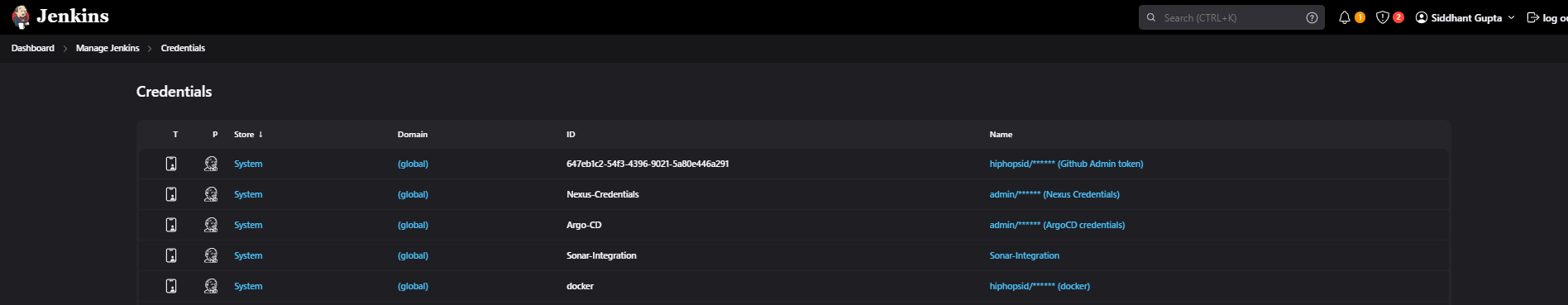
* CI Tool: Jenkins
* CD Tool: ArgoCD

**Installation and configuration process:**

* Jenkins Installation: We are running Jenkins as a service.
  + wget -q -O - <https://pkg.jenkins.io/debian-stable/jenkins.io.key> |sudo gpg --dearmor -o /usr/share/keyrings/jenkins.gpg\
  + sudo sh -c 'echo deb [signed-by=/usr/share/keyrings/jenkins.gpg] <http://pkg.jenkins.io/debian-stable> binary/ > /etc/apt/sources.list.d/jenkins.list'
  + sudo apt update
  + sudo apt install jenkins
  + sudo systemctl start jenkins.service
  + sudo systemctl status jenkinssudo usermod -aG docker jenkins, then restart your jenkins
* SonarQube Installation: We will run it as container
  + docker run -d -p 9000:9000 sonarqube:lts-community
  + Access: localhost:9000
* Nexus Installtaion:
  + docker run -d -p 8081:8081 sonatype/nexus3
  + access nexus: localhost:8081
  + docker exec -it 5e15c75afc6e cat /nexus-data/admin.password --> To get first time password
* Install trivy:
  + https://aquasecurity.github.io/trivy/v0.18.3/installation/
  + sudo apt-get install wget apt-transport-https gnupg lsb-release
  + wget -qO - <https://aquasecurity.github.io/trivy-repo/deb/public.key> | sudo apt-key add -
  + echo deb <https://aquasecurity.github.io/trivy-repo/deb> $(lsb\_release -sc) main | sudo tee -a /etc/apt/sources.list.d/trivy.list
  + sudo apt-get update
  + sudo apt-get install trivy
* Configuring Jenkins

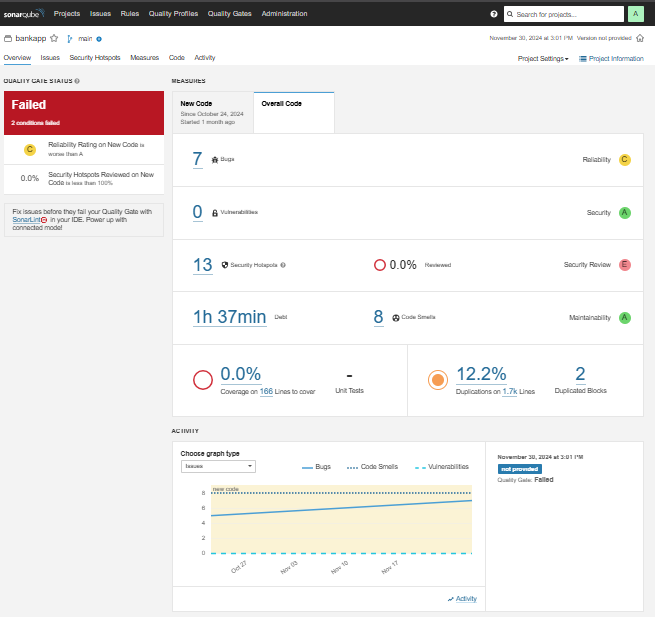
1. Install necessary plugins:

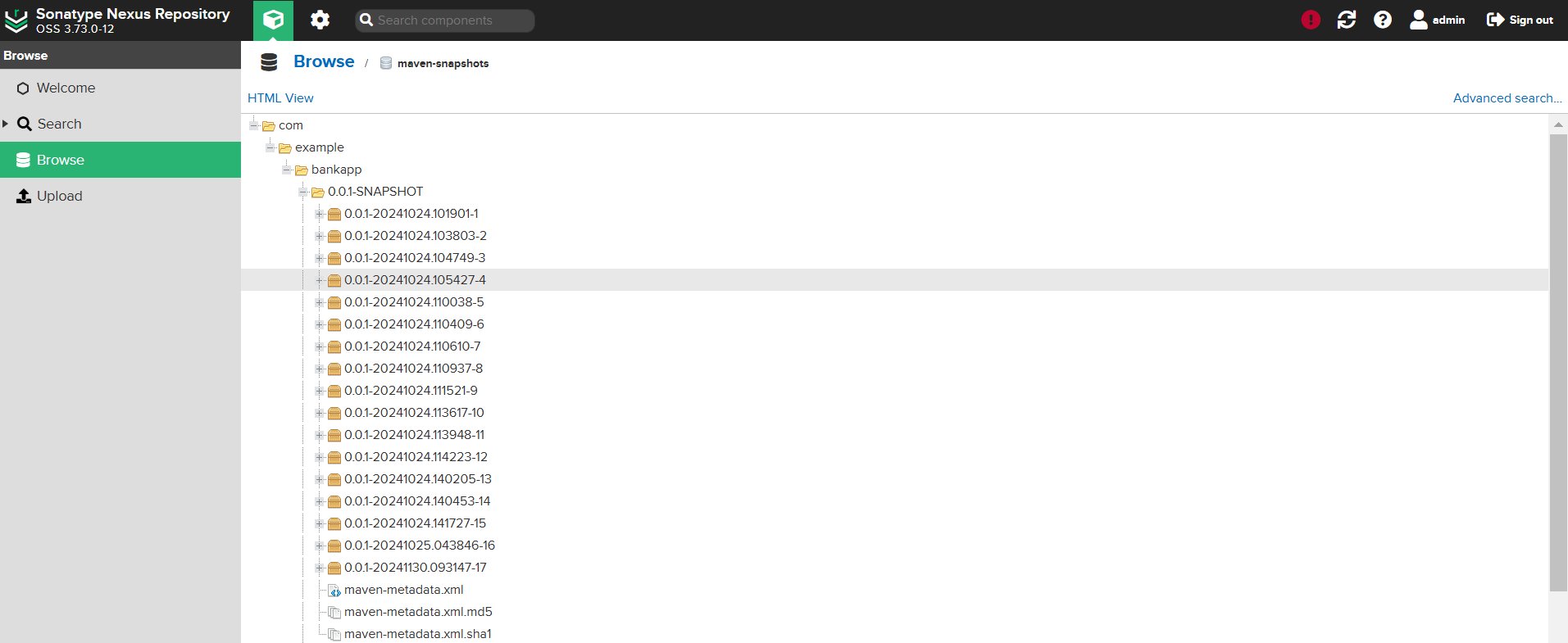
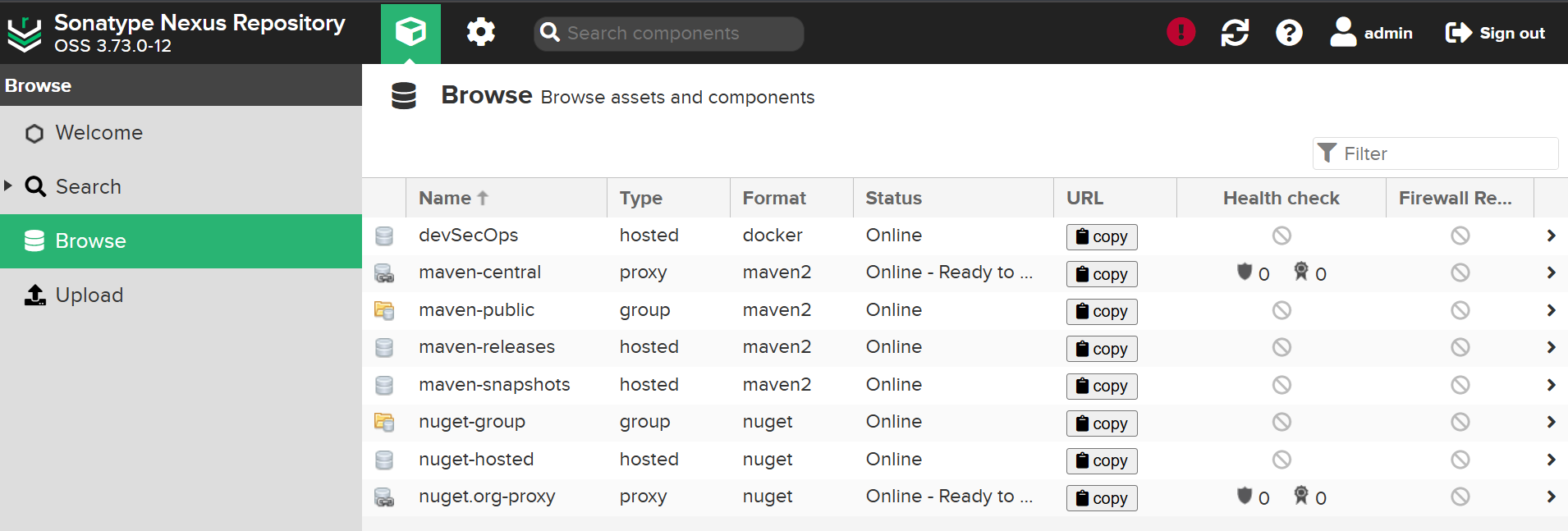
* Sonarqube Scanner
  + 
* Maven
  + 
* Maven integration Plugin
* Docker
* Pipeline: Stage View
* Config File Provider: to store credentials\
* Dependency Check
  + 

1. Configure required Credentials:
   1. 

**Covering CI Steps:**

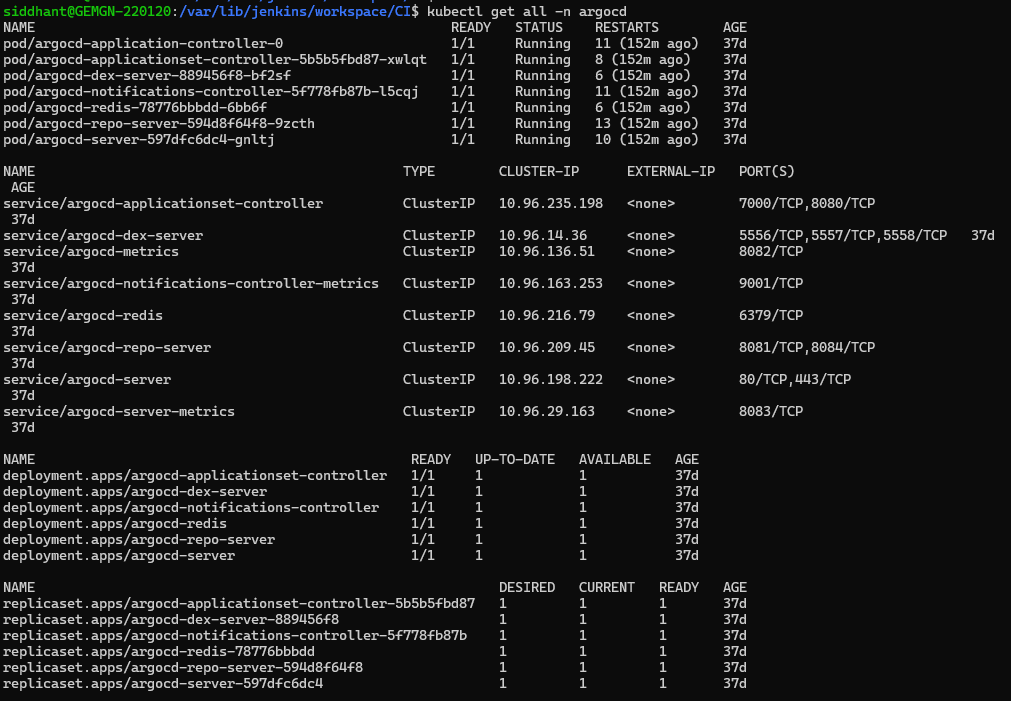
The pipeline includes 12 stages in total to complete the CI/CD pipelines:

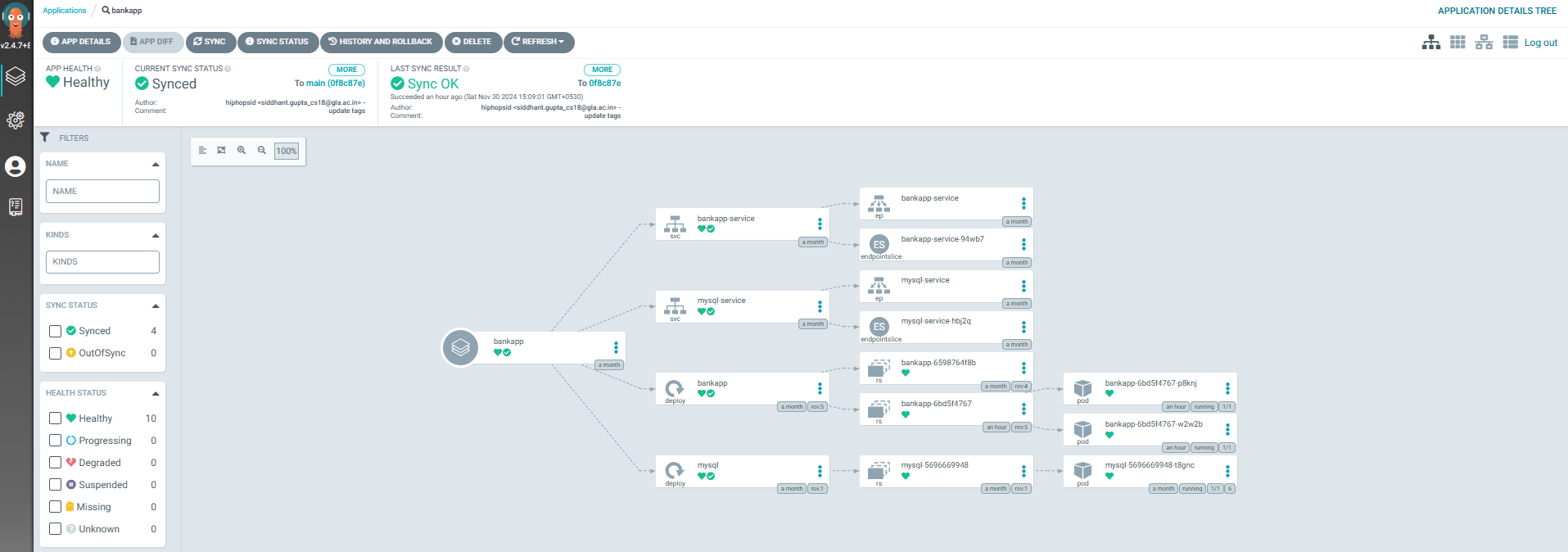
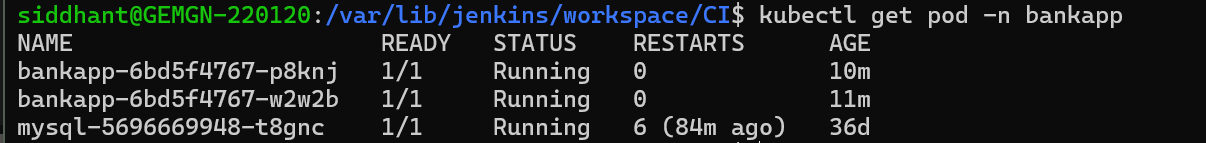
* Declarative: Tool Install
  + Installing all required tools, in my case maven, as application is Java based.
* Git Checkout
  + To checkout the repo and clone it into our jenkins workspace
* Compile
  + “mvn compile”-- Building/Compiling the applictaion
* Unit test cases
  + mvn test -DskipTests=true
* OWASP Dependencies Check
* Sonar Check
  + 
* Build and Publish to nexus
  + Using nexus as Artifact to store my docker images and application artifacts.



* Docker Build and Tag
  + Containerizing the image with the required build tag.
* Docker Image scan-- Doing trivy Scan to identify vulnerabilities
* Docker Push
  + To push the image on the artifactory.
* Update YAML Manifest in CD Repo
  + Doing following steps in the pipeline:
    - Cloning the CD repository
    - Updating iamge tag in the deployment.yaml file
    - And, pushing again the updated changes to the repo

**Covering CD steps:**

1. ArgoCD Configuration:
   1. 

* Synced the application’s CD repo containing yaml files under ArgoCD to automate the deployment.
  + 
  + 
  + 