**Gitlab Ci- creating a pipeline with java springboot code with sonarqube as my testing tool- used ec2 instance as gitlab runner - installed docker executor**

Steps:

* A screenshot of a computer

  Description automatically generated

1. Create a project on the **Git lab for the Continuous Integration**- in the project add, a **Java Springboot application code** along with the a **Gitlab Ci runner.**
2. For runner we are using **Amazon EC2 instance.** For configuring it we are using **t2.medium, create a key-value pair, and in the security groups** on the **inbound and outbound traffic rules allow traffic from the ports 9000, 80 and 443-** uses **9000 on the sonarqube, 80 for the information from the remains in plain text between the browser and the server, and for the 443 is HTTPS protocol means all the information travels between the server and the browser remains encrypted**. Same for the outbound rules as well.
3. For using **maven** as the **build** tool, im using the **docker image** which already has maven installed in it and i’ve used **Docker Executor**, In **GitLab CI/CD**, the **Docker executor** is a specific type of **runner executor** that enables you to run your **CI/CD jobs inside Docker container**s. Usually in Jenkins we use docker agents as a worker node, we use docker executor in **Gitlab Ci.**
4. Installing docker on your **EC2 instance**, by ssh into your **ec2 ssh -i /path/to/your/key.pem ec2-user@your-instance-ip, l**ogin into it using key value pair and public ip for installing the following commands are used sudo apt update**, sudo apt install** [**docker.io**](http://docker.io)
5. For installing the Sonarqube the following steps needs to be done,  **apt install unzip**

* **adduser sonarqube**
* **wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-9.4.0.54424.zip**
* **unzip \***
* **chmod -R 755 /home/sonarqube/sonarqube-9.4.0.54424**
* **chown -R sonarqube:sonarqube /home/sonarqube/sonarqube-9.4.0.54424**
* **cd sonarqube-9.4.0.54424/bin/linux-x86-64/**
* **./sonar.sh wrapper**  java is the pre-requisite for sonarqube so java needs to be installed and **java -jar target/spring-boot-web.jar,** this command is used to access it on port 8080
* **./sonar.sh start,**  for accessing your sonarqube, we should put the public ip of ec2 followed by 9000

1. **Generate a token from the admin of the sonarqube and paste it in Gitlab Ci repo of Ci along with Docker ID and Docker Password**
2. Last step would be installing the gitlab runner using # Download the binary for your system

* sudo curl -L --output /usr/local/bin/gitlab-runner https://gitlab-runner-downloads.s3.amazonaws.com/latest/binaries/gitlab-runner-linux-amd64
* # Give it permission to execute
* sudo chmod +x /usr/local/bin/gitlab-runner
* # Create a GitLab Runner user
* sudo useradd --comment 'GitLab Runner' --create-home gitlab-runner --shell /bin/bash
* # Install and run as a service
* sudo gitlab-runner install --user=gitlab-runner --working-directory=/home/gitlab-runner
* sudo gitlab-runner start

1. For the last step use the registration token of the gitlab Ci to the command line with your gitlab url and git lab token
2. Provide the runner as docker and pipeline is built.