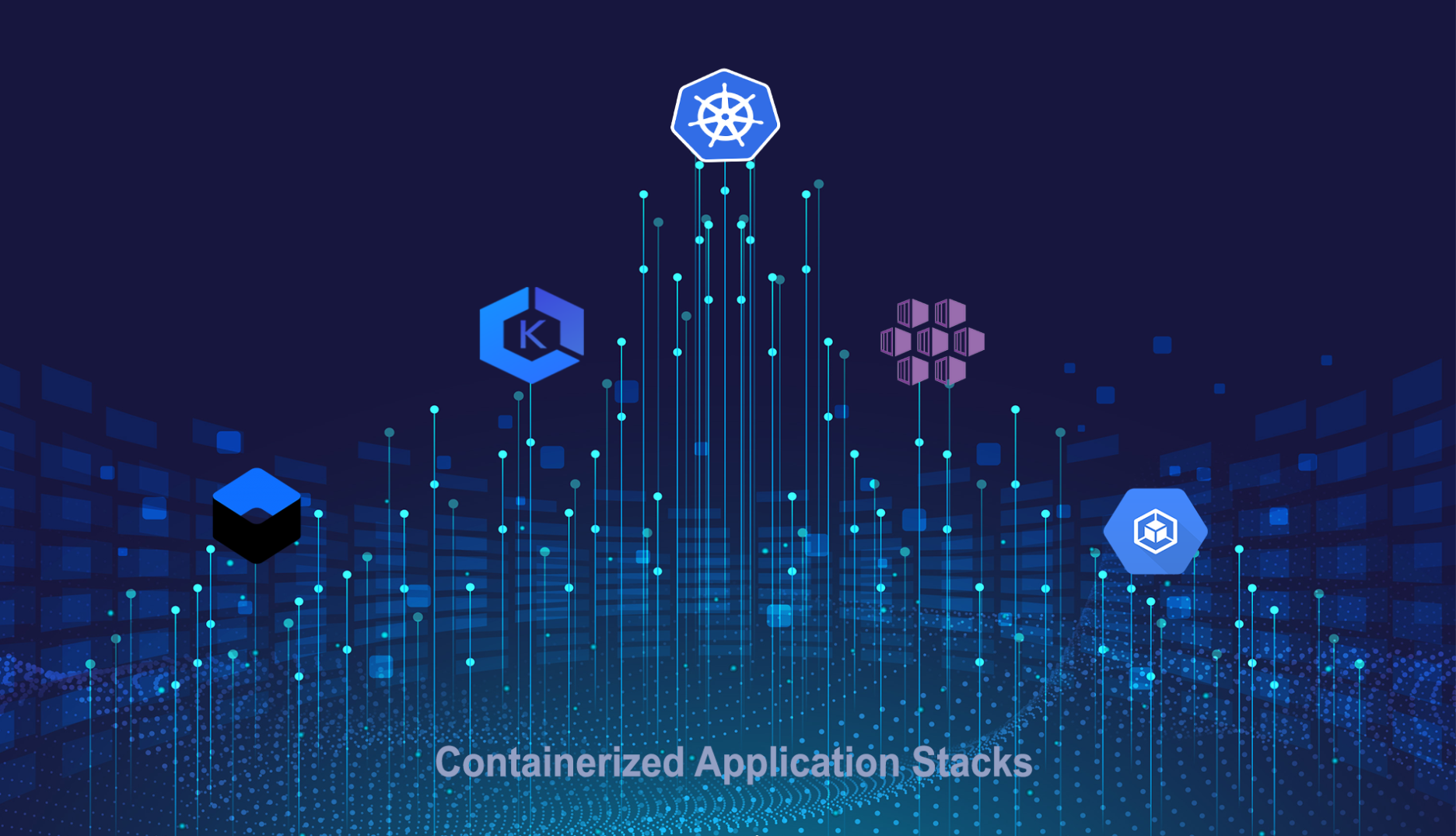
**Automated CICD Jenkins with pre build SonarQube and OWASP** 

**Technical Usage Manual – Automating Jenkins AMI with SonarQube and OWASP**

**Jenkins** offers a simple way to set up a continuous integration and continuous delivery environment for almost any combination of languages and source code repositories

Jenkins achieves Continuous Integration with the help of plugins and allows the integration of various DevOps stages. In this deployment scenario if you want to test and build your application, also the same development code should go through multiple stages of verification manually or with verified softwares to pass the defined metrics, the scenarios will be achieved with the help of installing plugins for that software.

Here, we are using GitHub as the repository for source code, connecting Jenkins with SonarQube Scanner for code test and OWASP for security scanning of the built image after deployment. Appropriate docker plugins and necessary tools such as Email extensions will be integrated to Jenkins and allow users to have a seamless experience in UI. .

Yobitel audits and validates that its Cloud-Native Stacks are stable, secure and scalable, meeting production grade standards with best practices. Yobitel stacks ensure a few click deployments over the instance with version upgrade resilience.

**Abstract:**

Jenkins with pre-built SonarQube and OWASP is available for code analysis and application testing. You can configure your pipeline using our pre-built options, including SonarQube and OWASP. We offer pre-configured pipelines that support code testing in any language and automated deployment of code in AWS ECS Fargate with ECR image. These pipelines are designed for SMEs and startups, facilitating continuous testing and deployment in ECS.

**Technical Usage Manual**

**Prerequisites**

1. Credentials - Add AWS Access Key & Secret Key, Github Fine Grained Access Token, Email credentials with Jenkins

2. Configuring Jenkins

**1.Credentials:**

**1.1 AWS Access Key &** **Secret Key:**

Obtain the [AWS Access Key & Secret Keys](https://docs.aws.amazon.com/keyspaces/latest/devguide/access.credentials.html#create.keypair) from your **AWS Account**

**1.2.Github Fine Grained Access Token:**

You can get [Github Fine Grained Access Token](https://github.blog/2022-10-18-introducing-fine-grained-personal-access-tokens-for-github/) from Developer Settings on your

Github account.

**1.3.Email Credential:**

Email credential must be Generated from [App Password](https://support.google.com/mail/answer/185833?hl=en#) on Google.

**2. Configuring Jenkins**

2.1 Jenkins EC2 instance **Public IP**  is available in **CloudFormation stack**

**Outputs,**  and connect Jenkins dashboard with (Eg: 18.212.180.159:8080).

2.2 Login to Jenkins with username: **admin** and password:

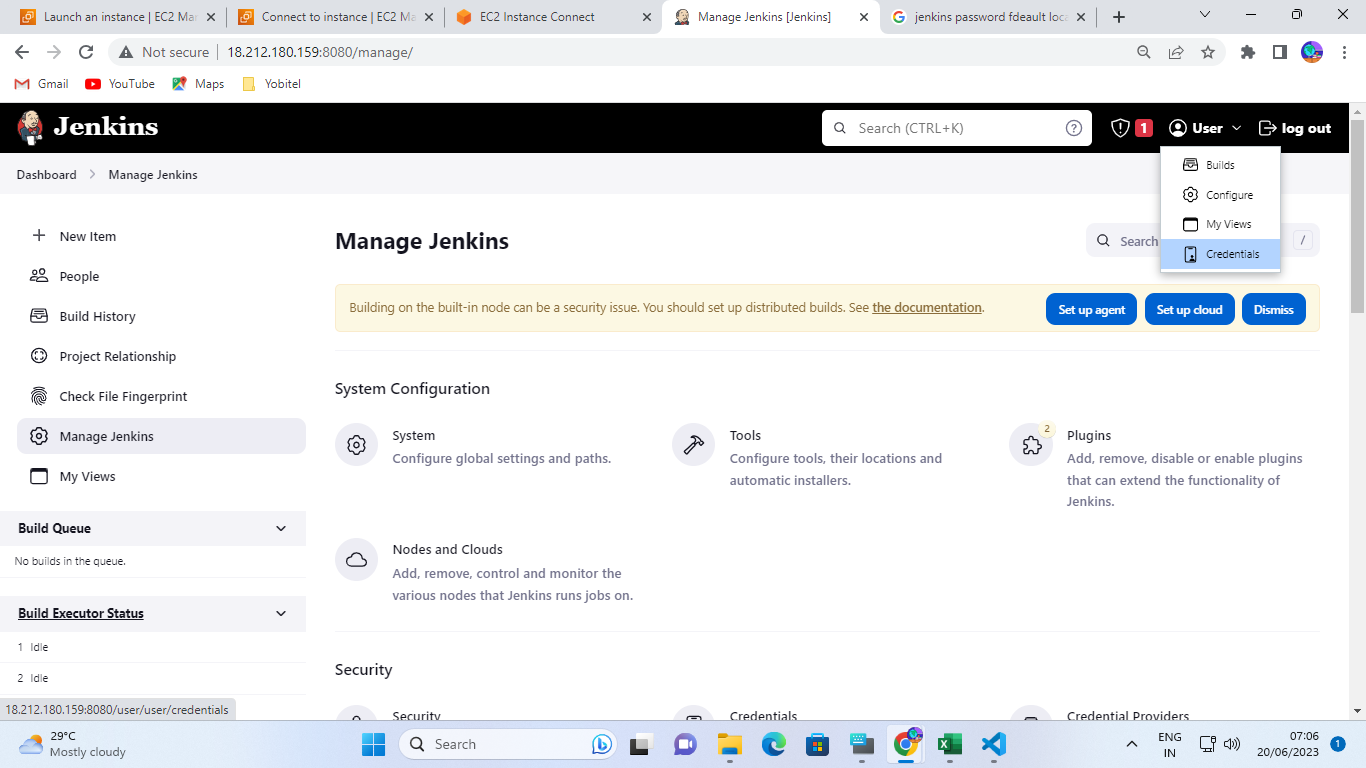
**dc05e43dcb6340d3add94e1d2c01d020** (or) use the command

**cat /var/jenkins/secrets/initialAdminPassword** in the **AWS CLI.**

2.3 Go to **Manage Jenkins >>** **Security>>User,** Create an **New User** with

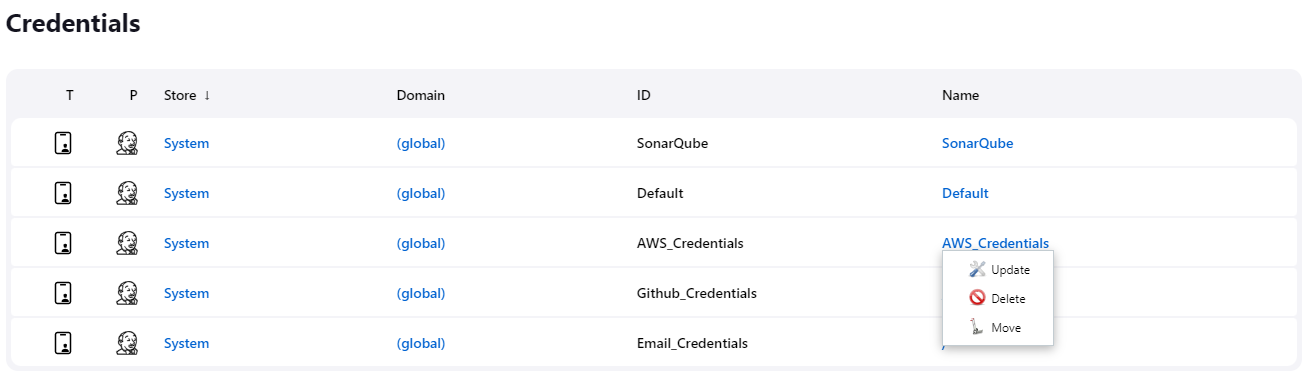
**Credentials** and Login as **User.**

2.4 Click **Credentials** under the **User**

****

2.5 Update the **AWS Credential** , **GitHub Credential** and **Email Credential** with your

own.



2.6 Go to **Manage Jenkins >> System** **>>Jenkins Location,** update your Jenkins

Instance Public IP in the Jenkins URL (Eg: http://<Public IP>:8080).

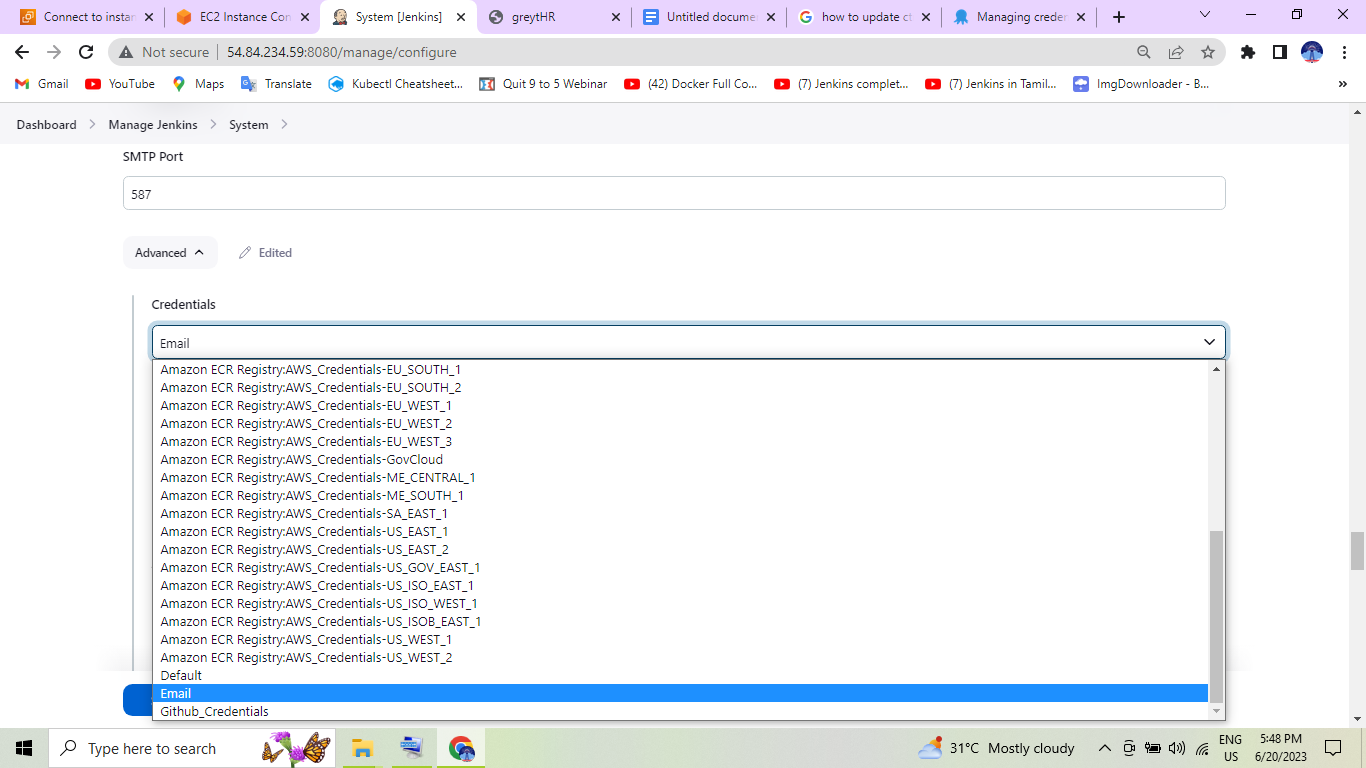
2.7 Go to **Extended Email Notification** and choosethe **Advanced,** choose your

**Email Credential.**

2.8 Choose to  **Email Notification >>** **Advanced,** Selectthe **Use SMTP**

**Authentication** and Enter **User Name** and **Password.** Finally Apply and Save the

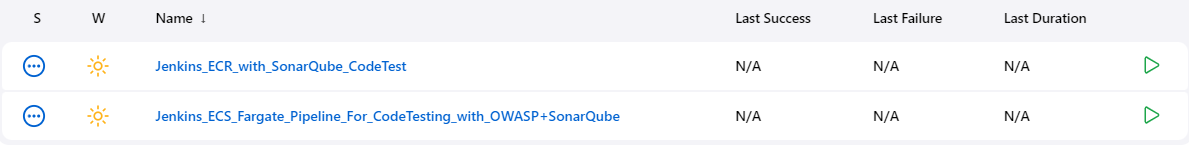
configuration.



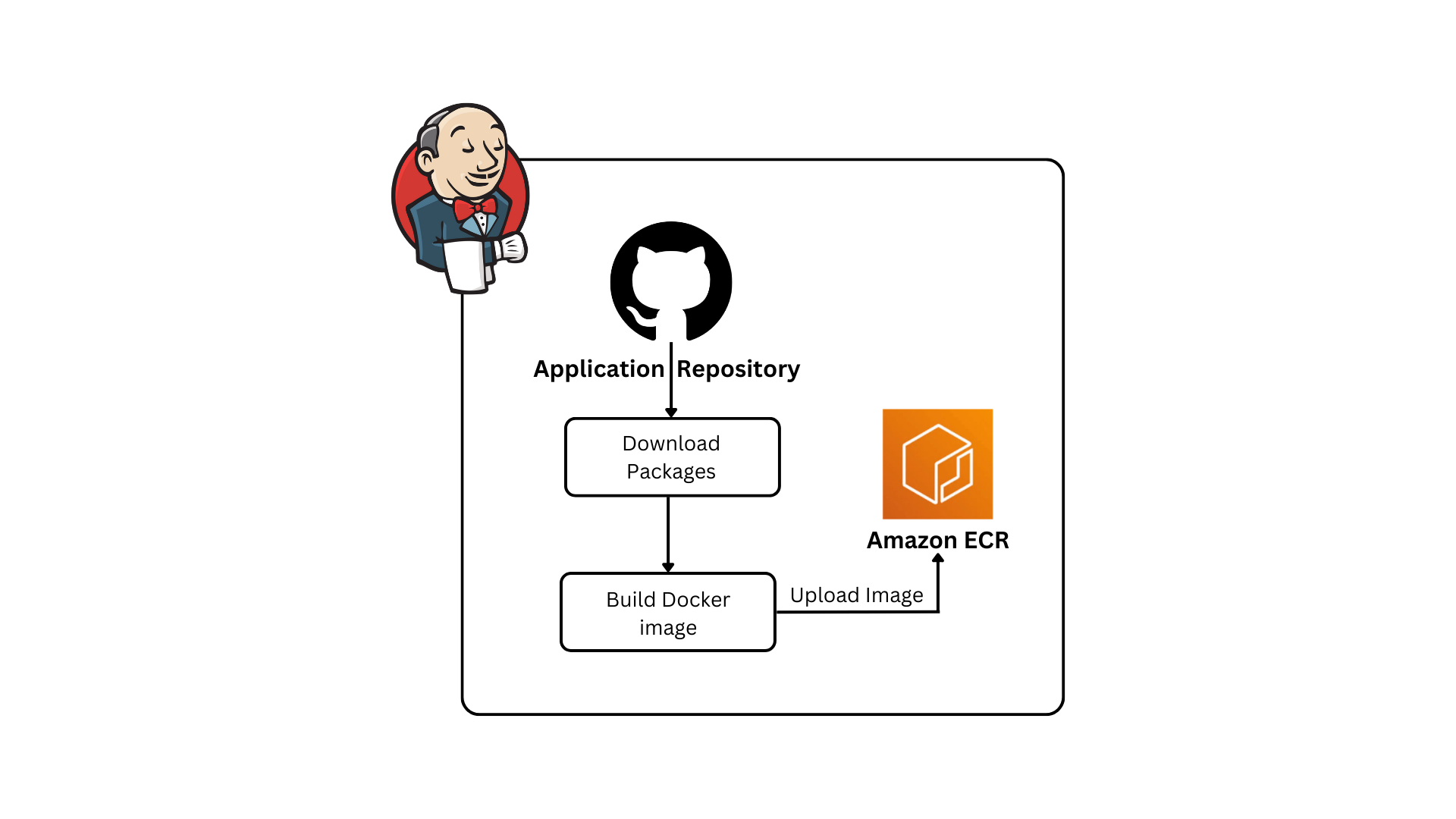
2.9 Go to **Dashboard**

On the Jenkins Dashboard, preloaded pipelines will be shown for different methods of CICD operations.

**Available Pipelines:**

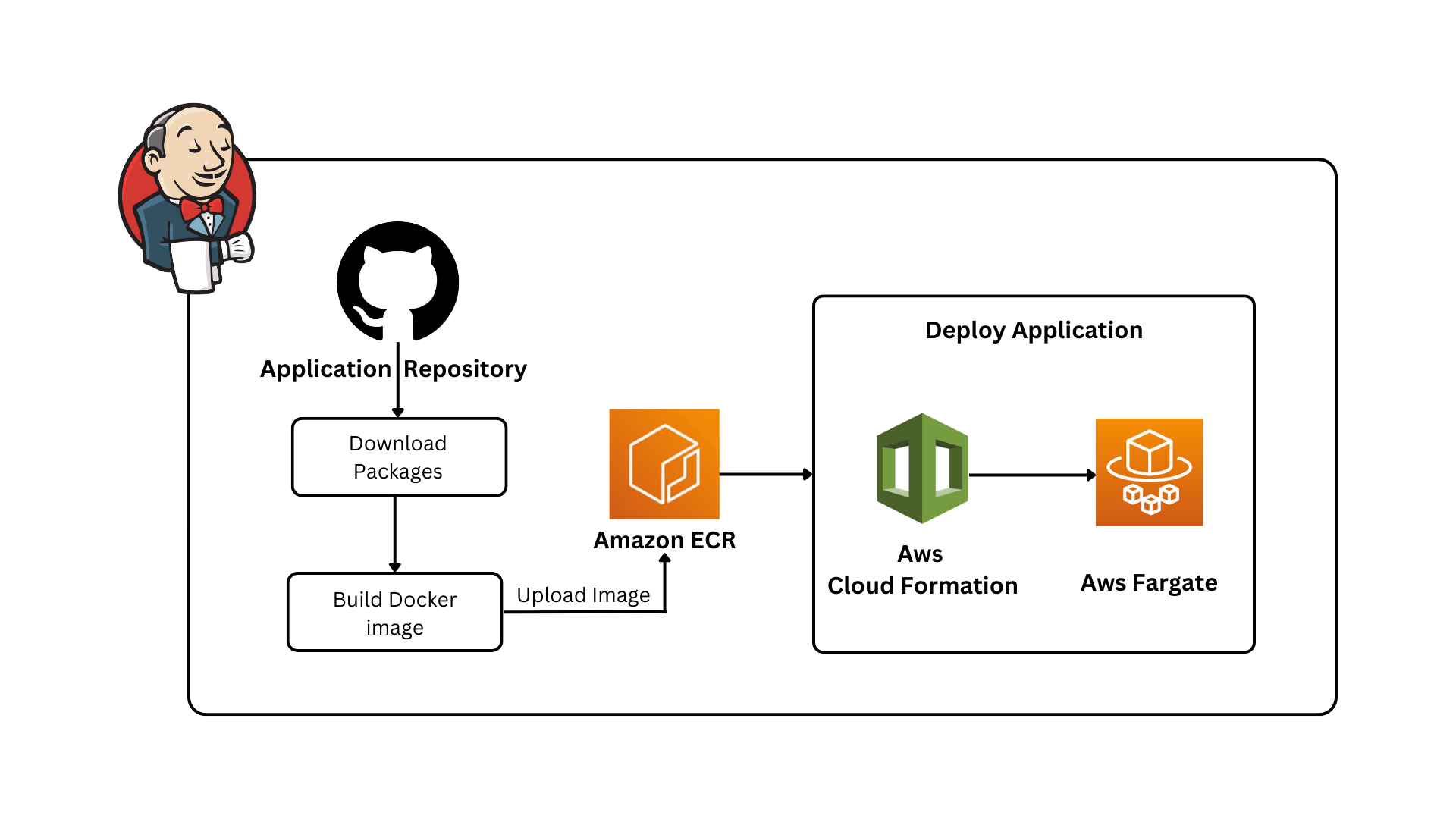
****

**Jenkins\_ECR\_with\_SonarQube\_CodeTest:** Automated CICD Pipeline to analyse source code using SonarQube, build it as docker image and push into **ECR.**



2.1 Jenkins\_ECR\_with\_SonarQube\_CodeTest

**Jenkins\_ECS\_Fargate\_Pipeline\_For\_CodeTesting\_with\_OWASP+SonarQube:** Automated CICD Pipeline to analyse source code using SonarQube, build it as docker image and push into **ECR.** Deploy the Application build Image in **AWS ECS Fargate**  from ECR repository and analyse the application using **OWASP.**



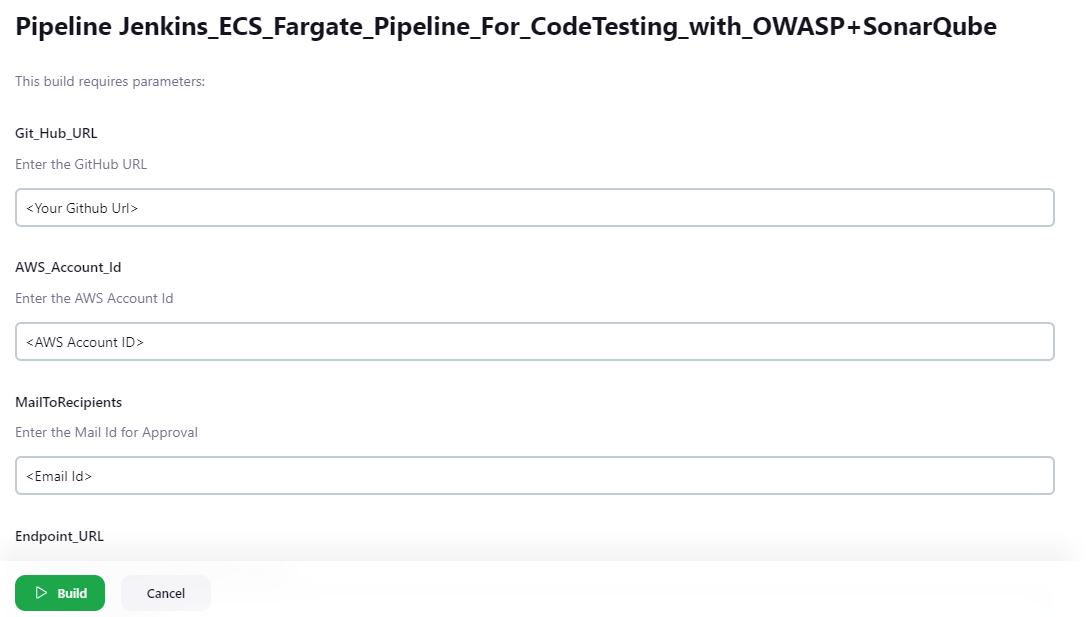
2.2 Jenkins\_ECS\_Fargate\_Pipeline\_For\_CodeTesting\_with\_OWASP+SonarQube

2.10 Choose the Pipeline which you need and **▶ to Schedule a Build.**

2.11 Enter the required parameters in your pipeline to build.

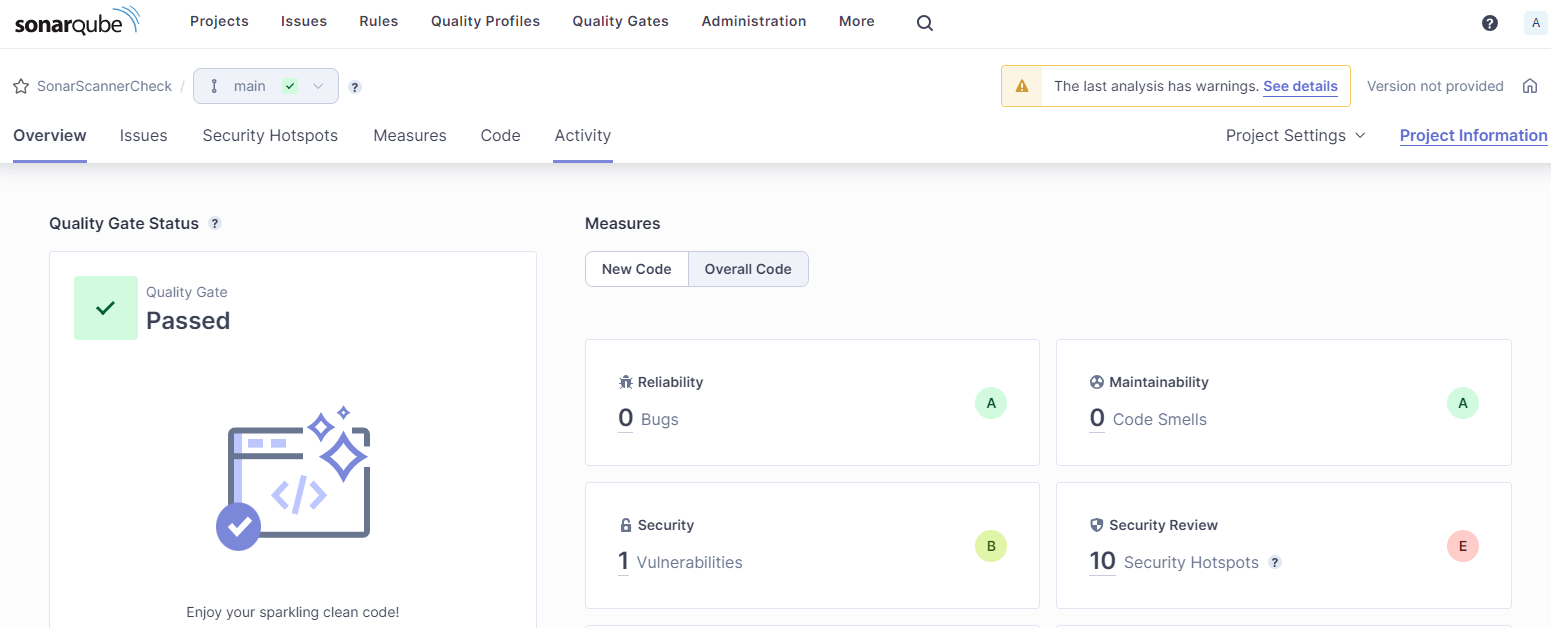
(Note: Some parameters are given as default, If you need you can change it by

your own)

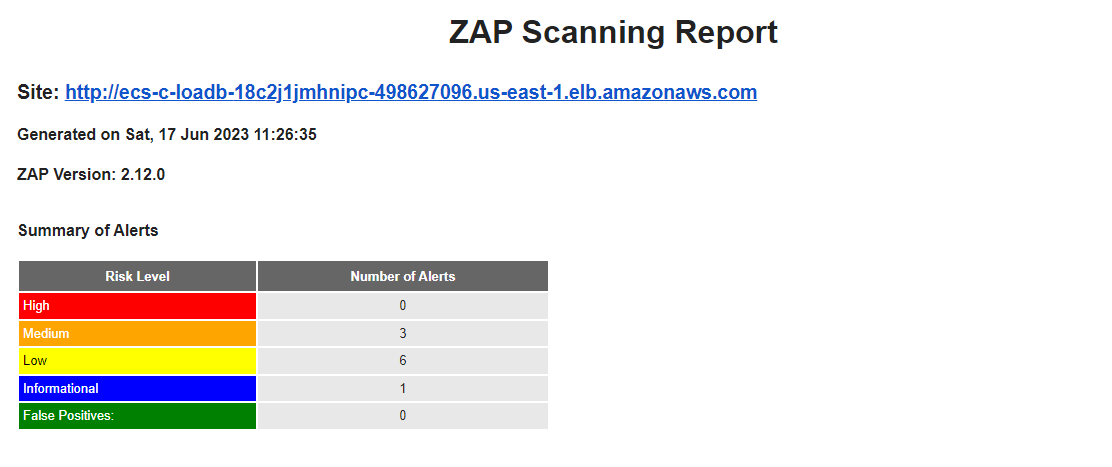


**Appendix:**

While building these pipelines you will receive the SonarQube and OWASP ZAP Report which will look like below shown.



SonarQube Analyse Report



OWASP ZAP Report

**Note**: These Pipelines consist of several stages for Continuous Integration and Continuous Deployment with Sonarqube and OWASP which was given by default.In addition we configured mail service which will send the code analysis report and OWASP Zap report to respective email you provided. Our pipeline supports any type of code to analyse and deploy your application in ECS Fargate. For the further improvement of the pipeline, you can contact our support team.

**Insights & Support:**

* For further details about Jenkins and its uses, refer [Jenkins](https://www.jenkins.io/) document
* We will do our best to respond to your questions within the next 24 hours in business days.  For any technical support or query you can drop a mail to [support@yobitel.com](mailto:support@yobitel.com).
* Check our other Containerized Cloud-Native application stacks in [AWS Marketplace](https://aws.amazon.com/marketplace/seller-profile?id=5ac1a1e2-3686-4310-ba24-520bd32259a5) .