Managing access control and authorization

- Managing access control and authorization
  - Go to `Manage Jenkins` > `Configure Global Security` > `Enable security`.
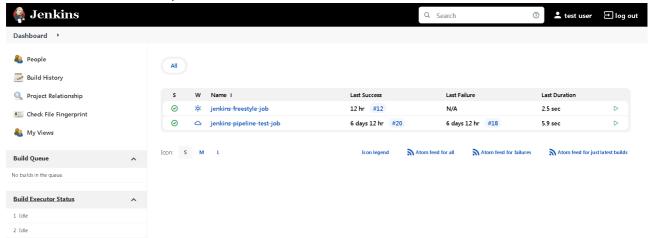
Creating Users in Jenkins

- Select `Manage Jenkins` > `Manage Users` > `Create a user`

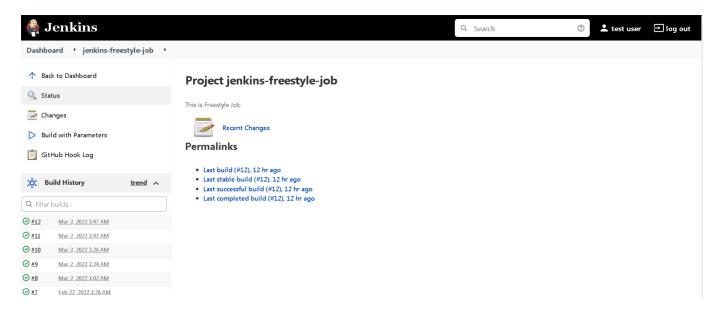## Maintaining roles and project-based security

- Lets configure some users in Jenkins, create a read only user `readonlyuser`
- Go to `Manage Jenkins` > `Configure Global Security` > under `Authorization` > Select `Project-based Matrix Authorization Strategy`.
  - Add `User/Group` configure security based on different sections such as `Overall,Credentials,Job,View and so on`.
  - Click on `Apply and Save`.

- Use another browser and try to access the Jenkins dashboard with a newly added user. This User will not be able to see all the options in Jenkins Dashboard.
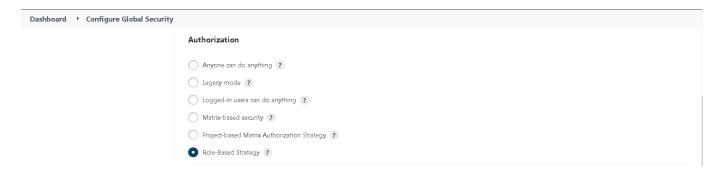
- Here, the user is not able to see the `Configure` Option under a specific Job, because Job Specific Permissions are granted.
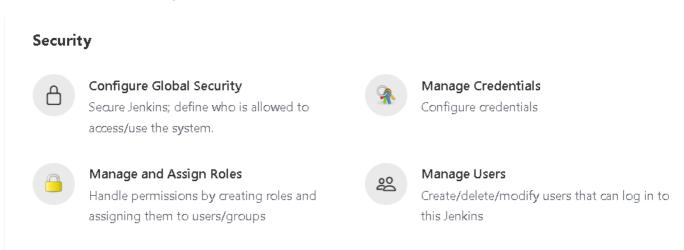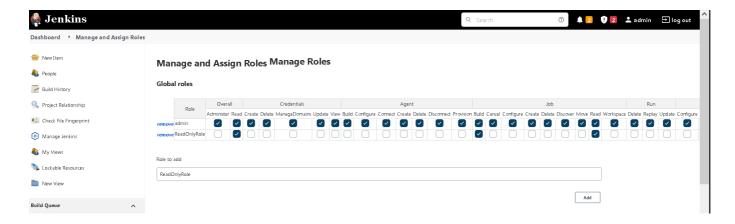

Role-Based-Authorization Strategy

Manage Jenkins > Manage Plugins > Install the `Role Based Authorized Strategy` Plugin.

- Add plugin from available tab in Plugins Manager i.e `Role Based Authorized Strategy`
- To enable `Role-Based Strategy`, Go to `Manage Jenkins` > `Configure Global Security` > Select the `Role-based Strategy`
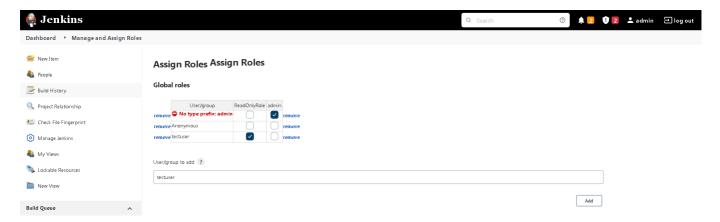

- To Create a Role, Go to `Manage Jenkins` > `Manage and Assign Roles` > `Manage Roles` > `Role to add` > Enter `ReadOnlyRole`.

- Click on `Apply and Save`.

- To Assign this Role to a user, Go to `Manage Jenkins` > `Manage and Assign Roles` > `Assign Roles` > `Enter Username` > Select Role to be assigned `ReadOnlyRole`.

- Use another browser and try to access the Jenkins dashboard with a user to check permissions in the Role are applied.
- Item Roles:
  - This can be created for specific Jobs access and can be assigned to user/group.

Jenkins Github Webhook

- Integrate jenkins with github so automatically CICD works when any commit is made to the repo Go to `Jenkins` > `Manage Jenkins` > `Configure System` > `Add a Github Server` > Add a Name : `jenkins-github-webhook-server` and Enter API URL : `http://public-ip:8080/github-webhook/`
- Lets add a webhook in Github to point to Jenkins URL
- In `Github Repository > Go to Repository > Settings > Go to webhook and addnew webhook > Specify http://public-ip:8080/github-webhook/`
- Go to Jenkins Project, Select the `GitHub hook trigger for GITScm polling` checkbox under Build Triggers tab > `Save`.
- For Webhook to work, open port `8080` in security group for IP address of Github.
- Now if we make some changes to some file in Github, this Jenkins Project should be triggered.
- Jenkins receives a Github Payload similar to this Github Push Webhook Event Payload

When Jenkins receives a GitHub push hook, GitHub Plugin checks to see whether the hook came from a GitHub repository which matches the Git repository defined in SCM/Git section of this job. If they match and this option is enabled, GitHub Plugin triggers a one-time polling on GITScm. When GITScm

> polls GitHub, it finds that there is a change and initiates a build. The last sentence describes the behavior of Git plugin, thus the polling and initiating the build is not a part of GitHub plugin.

Audit Trail Plugin

- Manage Jenkins > Manage Plugins > Install the `Audit Trail` Plugin.
- Go to `Manage Jenkins` > `Configure Systems` > `Audit Trail` > `Add Logger` > Select `Log File`
- Provide the Log Location as `/var/log/jenkins/audit-%g.log`, provide Log File Size as `50` and Log File Count `10`.
- After executing some build job for some Jenkins Project, check the content of the audit file.

```
ls -ltr /var/log/jenkins/
```

> Audit Trail Plugin keeps a log of users who performed particular Jenkins operations, such as configuring jobs. This plugin adds an Audit Trail section in the main Jenkins configuration page. Here you can configure log location and settings (file size and number of rotating log files), and a URI pattern for requests to be logged. The default options select most actions with significant effect such as creating/configuring/deleting jobs and views or delete/save-forever/start a build. The log is written to disk as configured and recent entries can also be viewed in the Manage / System Log section.

Reference

- [Reset Jenkins Admin Password from CLI](#)