

EXPLORING JENKINS

- **Installing BlueOcean :**

Blue Ocean rethinks the user experience of Jenkins. Designed from the ground up for Jenkins Pipeline but still compatible with freestyle jobs, Blue Ocean reduces clutter and increases clarity for every member of the team. Blue Ocean's main features include Sophisticated visualizations, Pipeline editor, Personalisation.

1. If required, ensure you are logged in to Jenkins.
2. From the Jenkins home page, click **Manage Jenkins** on the left and then **Manage Plugins** in the center.
3. Click the **Available** tab and type blue ocean into the **Filter** text box.
4. Select the **Blue Ocean** plugin's check box near the top of the the **Install** column and then click **Install without restart** button at the the end of the page.

- **Building a private GitHub Repository :**

For Jenkins to access GitHub repository it should be Public else we have to give access to Jenkins.

1. Create a Private GitHub repository.
2. Start a new freestyle project in Jenkins.
3. Under Source code management select git.
4. Paste the url of private repository in the url section.
5. Since our repository is private we have to add credentials of our github using the add button.
6. Then select the credentials in the dropdown menu.
7. Use poll SCM in build and schedule it with required time.
8. Use execute shell in build and write the commands to be executed.

- **GIT Scm Poll :**

Using SCM Poll, Jenkins will check periodically (with mentioned time) if new commits are made and if they are made then it will build the changed version. But whereas using Git SCM polling it will create a link between GitHub and Jenkins, Jenkins will come to know when new commits are made and will build only then, it is more efficient as it need not periodically check if commits are made or not.

1. Configuring Build triggers:
 - i. Go to the project for which you need to implement Git SCM polling.
 - ii. Go to configure and scroll down to Build Triggers.
 - iii. Select GitHub hook trigger for GITscm polling.
 - iv. Write respective build commands and apply then save.

2. To get public hook url install ngrok and run the command `./ngrok http 8080`. This will give our local host a public ip address which should be used at webhook in settings of jenkins.
3. Activating Jenkins(GitHub Plugin) service to GitHub:
 - i. Go to that particular repository for which we need GITscm polling.
 - ii. Go to settings and then to Integrations and Services.
 - iii. Then select add service and search for Jenkins(GitHub plugin)
 - iv. Give your local url at Jenkins Hook url given by ngrok.
(`<hook given by ngrok>/github-webhook/`)
 - v. Select Active and click on Add service.

- **Post Build Step Email Notification:**

1. Configuring email address:
 - i. Go to manage jenkins then to configure system.
 - ii. Scroll down to Email Notification.
 - iii. Select use authentication (since it is required for mail).
 - iv. Go to advanced and enter the credentials required(You might have to give permission for jenkins to access Gmail).
 - v. Save the changes made.
2. Post build action:
 - i. Go to the project for which you require updates in email.
 - ii. Scroll down to post build actions.
 - iii. Select email notifications in the dropdown menu.
 - iv. Enter the mail id's of the recipients in the Recipients box given.
 - v. Choose either send email for every unstable build or send separate emails to individuals who broke the build depending on your requirement.
 - Send email for every unstable build will send the emails to the recipients given in the recipients box.
 - vi. Apply and save changes.