- **Blue Ocean:**

    Blue Ocean is designed from the ground up for Jenkins Pipeline. Blue Ocean rethinks the user experience of Jenkins. Blue Ocean reduces clutter and increases clarity for every member of the team. Blue Ocean's main features include Sophisticated visualizations, Pipeline editor, Personalization, Pinpoint precision, Native integration for branch and pull requests

    Installing Blue Ocean :
    - ➢ Ensure that you are logged-in to jenkins , if required.
    - ➢ Jenkin homepage
        - ✔ goto **Manage Jenkins**
        - ✔ then **Manage plugins**
    - ➢ In the Filter text box search for blue ocean and then check in **Blue Ocean** plugin's check box.
    - ➢ Then click **Install without restart** button and then plugin gets installed

- **Private Github Repository:**
    - ➢ In jenkins create a select **New Item** choose then choose freestyle project and then click **OK**.
    - ➢ In **Source Code Management** choose **Git** option
    - ➢ Create a private repository in Git. Download student package if you want to create it for free else give your credit card details.
    - ➢ Copy-paste the private repository's URL in Repository URL's box
    - ➢ As the repository is private, Jenkins can't access it without the credentials so click on **Add** button below the credential text box, choose Jenkins and then give the required details in the popup generated (choose **Username with Password** under the **Kind** option. Enter GitHub username and password under their respective text boxes).
    - ➢ Now choose the newly added credentials in the dropdown menu which is next to Credentials label. The private repository can be used with Jenkins.

- **Using Git SCM Poll:**
    - ➢ goto configurations of the project in **Build Triggers** choose **GitHub hook trigger for GITScm polling** option and save it.

    - ➢ Now goto github repository and then to **Settings**

    - ➢ From sub-menu choose **Integrations & services** option.

    - ➢ Select **Add service** and choose **Jenkins (GitHub plugin)**.

    - ➢ Now install **ngrok** in your terminal and Copy-paste the **Forwarding for localhost:80** in **Jenkins hook url**

    - ➢ Tick the **Active** checkbox if it's not ticked and click on **Add Service** button.

    - ➢ Now GitHub should make a request to the Jenkins webhook when ever commit is made

- **Post Built Actions – Email Notification:**

  Email Notifications option allows us to send email notifications when ever build is failed and also notifies you when the failed build becomes success.

  ➢ **Configuring e-mail id:**
    - ✔ In the homepage of jenkins goto **Manage Jenkins**.
    - ✔ Goto **Configure system**
    - ✔ goto **E-mail Notification**
    - ✔ provide the smtp server details. If you use gmail then give **smtp.gmail.com**
    - ✔ Select the **Advanced** option which is below Default user e-mail suffix.
    - ✔ Select **Use SMTP Authentication**
    - ✔ provide Username (gmail Id) and password of your gmail account.
    - ✔ Select **Use SSL**
    - ✔ provide **SSL port** (It's mostly 465)
    - ✔ enter your gmail id in **Reply-To Address** field
    - ✔ check in **Test configuration by sending test e-mail**
    - ✔ enter your gmail id in **Test e-mail recipient** field
    - ✔ Provide jenkins to access for gmail id
    - ✔ click on **Test configuration**
    - ✔ finally click on **Apply** and then **Save**

  ➢ **Post build Action:**
    - ✔ goto configuration of the project
    - ✔ click on Add post-build action and select **E-mail Notification** option in it.
    - ✔ Enter your gmail Id in **Recipients** and save.

  Now When ever you build is failed you get a Notification to entered email-id.