Assignment - 2_1 - Jenkins

Installing Blue Ocean

- Go to `Manage Jenkins > Manage Plugins`.
- Choose the `Available` tab and search for `blue ocean`.
- Select the `Blue Ocean (BlueOcean Aggregator)` plugin. The docs recommend installing only this one since all other Blue Ocean plugins will be installed automatically (as dependencies).
- Blue Ocean will be activated when Jenkins restarts.
- Choose the `Blue Ocean` option from the left menu to use the Blue Ocean UI.

Building a private GitHub Repository

(A `freestyle project` has been used here)

- Create a new freestyle project.
- Under `Source Code Management` select `Git`.
- Add the private repository's URL in the `Repository URL` text box.
- Since the repository is private, Jenkins will not be able to access the repo without credentials. Click on `Add` button next to `Credential` text box. Choose `Jenkins` as the Credential Provider in the dropdown menu. A popup should appear asking for the details.
- Choose `Username with Password` under the `Kind` option. Enter GitHub username and password under their respective text boxes.
- After adding your credentials choose the newly added credentials in the dropdown menu that appears next to `Credentials` label.
- Now the private repository can be used with Jenkins like any other repository.

Using Git SCM Poll

(The previous project has been used here)

Configuring build triggers

- Go to `Configuration` page of project.
- Under `Build Triggers` choose `GitHub hook trigger for GITScm polling` and save configuration.

Adding Jenkins service to GitHub

Now we need to set up our GitHub repo to make a request to Jenkins webhook so

that the polling logic can be applied.

- Go to GitHub repo and navigate to `Settings`.
- Choose `Integrations & services` from the submenu.
- Click on `Add service` and choose `Jenkins (GitHub plugin)` from dropdown menu.
- Add `http://<public-ip or URL>/github-webhook` under `Jenkins hook url`.
- Make sure the service is active by selective the `Active` checkbox and click on `Add Service` button.
- Now GitHub should make a request to the Jenkins webhook and cause a build (if required) to occur.

Post Build Actions - Extended Email Notification

Steps to setup Extended Email Notifications as a post build action have been documented below. Extended Email Notifications allow us to send customized email notifications after the build process.

Configuring Email

We need to first add the details of the SMTP server and the mail account so that Jenkins can send the mail.

- Go to `Manage Jenkins > Configure Systems` and scroll down to the `Extended E-mail Notification` section.
- Add the SMTP server details e.g. `smtp.gmail.com` for a `Gmail account`.
- Click on the `Advanced` button to configure the mail account that will be used to send the mails.
- Select `Use SMTP Authentication` if required e.g. Gmail uses a password for authentication and so this must be selected.
- Enter the account username under `User Name` and password under `Password`.
- Select `Use SSL` and specify the `SMTP port` (465 in most cases).
- Select `Allow sending to unregistered users` if mail should be sent to non-Jenkins users.
- With this the basic email configuration is complete.
- One can use the other options to customize the mail sent. For e.g. we can specify the list of default recipients and default content.
- Jenkins injects certain variables like `$PROJECT_NAME`, `$BUILD_NUMBER` and `$BUILD_STATUS` which can be used in the content to get dynamic values.

Adding the post build action

- Go to `Configuration` page of project and scroll down to `Post-build Actions`.
- Click on `Add post-build action` and choose `Editable Email Notification` from

dropdown menu.

- Jenkins once again provides the default settings through variables e.g. `$DEFAULT_RECIPIENTS`, `$DEFAULT_SUBJECT` and `$DEFAULT_CONTENT`.
- The options can be used for project based customizations of the email notification.
- The `Advanced` settings can be used for specifying triggers i.e. conditions under which the notification should be sent e.g. On every failed build.

With this the Post build action for extended email notification is complete.