3.5 Schedule Automatic Builds and Cron Jobs



This section will guide you to:

* Setup a recurring Schedule in cron format

This guide has five subsections, namely:

3.5.1 Setting up Jenkins in Windows 10

3.5.2 Setting up a .NET Build Project in Jenkins

3.5.3 Setting up a Schedule to generate a build

3.5.4 Setting up a recurring Schedule in cron format

3.5.5 Pushing the code to your GitHub repositories

* Jenkins is installed in your practice lab. Refer to **DotNet Lab guide: Phase 4** for more information.

**Step 3.5.1:** Setting up Jenkins in Windows 10

* Jenkins is already installed in your practice lab. Refer **DotNet Lab guide:Phase 4** for more information.

**Step 3.5.2:** Setting up a .NET Build Project in Jenkins for continuous integration

* Open Jenkins dashboard page on <http://localhost:8080> in your browser.
* Login as admin user.
* From the left bar, click **Manage Jenkins.**
* In the next page click **Manage Plugins.**
* Click on the **Available** tab.
* Check **Team Foundation Server,** **VS Team Services Continuous Deployment,** **MS Build** and **Schedule Build.**
* At the bottom of the page click **Install without Restart.**
* Go back to the Dashboard page.
* In the Dashboard page, click on **New Item** from the left bar.
* Enter name as NET\_BUILD and select **FreeStyle Project** andclick **Ok.**
* This will open the Configuration screen.
* In the **General** tab enter a description of your choice and check **Discard Old builds.**
* In **Log Rotation,** set **Days to Keep Builds** to 3.
* Click **Source Code Management** tab and check **Team Foundation Version Control.**
* Enter the TFVC **Collection URL** eg. <https://team.myprojects.com/>. (You can see this URL in the TFS Admin console. The URL should comprise of the details mentioned. - <http://servername:8080/tfs/thiscollectionname> )
* Enter the **Project Path.** Eg. $/MyTeam.
* In the **Credentials** dropdown select **Manual** andenter your VSTS username and password(This can be set to **Automatic** also).
* Click the **Build Environment** tab.
* Check **Delete workspace before build starts**
* Click **Add Build Step** option and choose **Build a Visual Studio Project or Solution using MS Build.**
* Fill in the relevant values for **MSBuild Version, MSBuild Build Path** and **Command Line Arguments.**
* Under **Post Build Actions** click **Add Post Build Action.**
* Choose **Email Notification** and enter the recipient email ids in the form.
* Click **Save.**
* From the left bar, click **Build Now** to build the project. This will generate a build in the Jenkins workspace. Once the **Build Number** is displayed click on it to see the workspace. This will show all the files that have been generated by MSBuild.

**Step 3.5.3:** Setting up a one-time Schedule to generate a build

* Open Jenkins dashboard page on <http://localhost:8080>.
* Login as admin user.
* Click on **NET\_BUILD** from the list of displayed projects.
* From the left bar, click **Schedule Build.**
* Enter a date and time and click **Schedule.**
* This will show the queued build item in the left bar for this project.

**Step 3.5.4:** Setting up a recurring Schedule in cron format

* Click on **Configure** in the left bar.
* Click the **Build Triggers** tab.
* Check **Build Periodically.**
* In the **Schedule** textarea enter the recurring time slot for generating a build, in cron format. Eg for running it on the 1st and 15th of every month put **H H 1,15 1-11 \* .** You can enter multiple schedules for the project by entering multiple rows.
* Click **Save.**

**Step 3.5.5:** Pushing the code to your GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add .

Commit the changes using the following command:

git commit -m “Changes have been committed.”

Push the files to the folder you created initially using the following command:

git push -u origin master