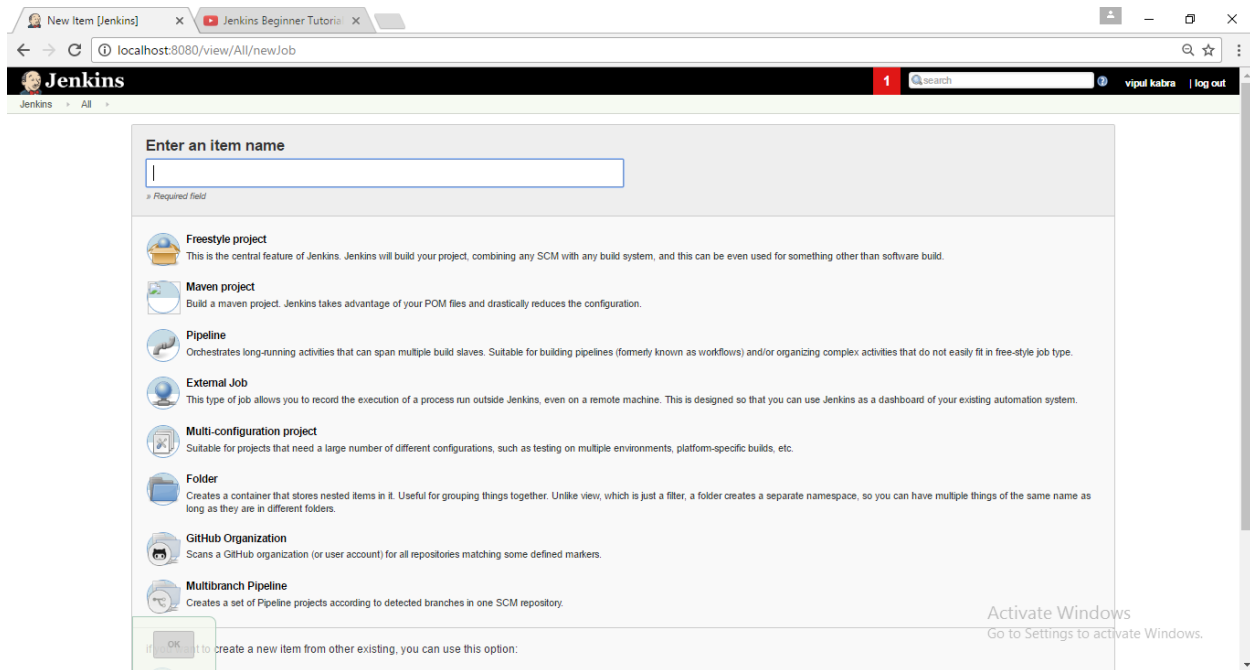
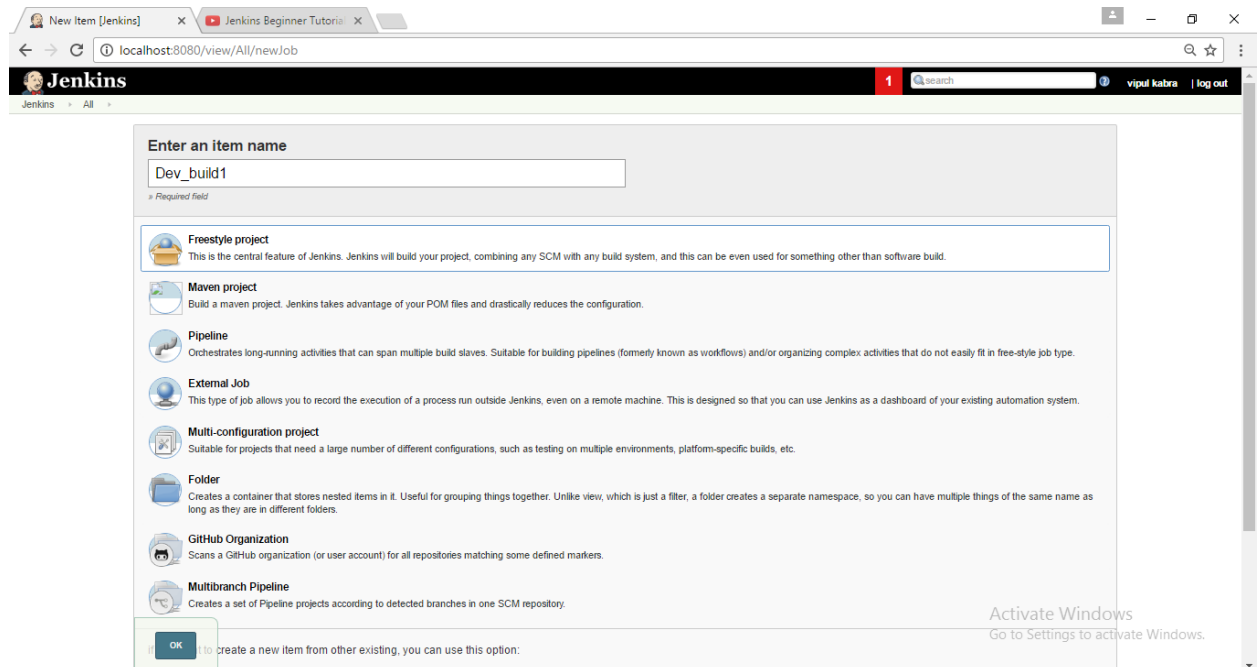


Creating Job in Jenkins and configuring the same and playing with build

➔ Create Job/build – Login in to Jenkin and Click on new Item from top left hand option



➔ Enter Job name and select project type



There are 5 tabs that need to be provided for the job they are as below

General → Provide the description of Job/build

Source code Management → If source code will be pulled from any repository you can select this. Note make sure those plugin are installed to get them added

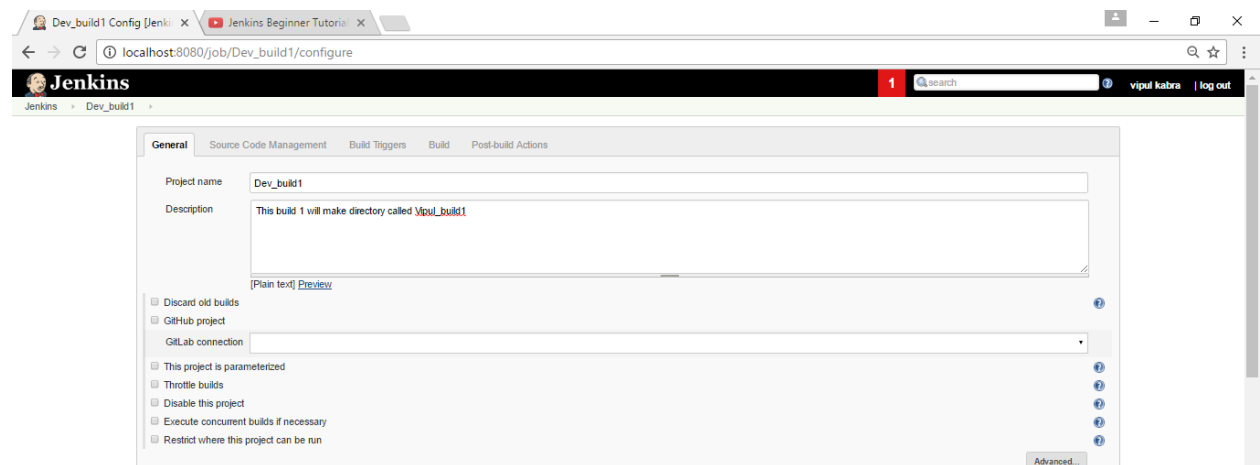
Build Triggers → Define how build want to be triggered using script, should do if any other build is completed, schedule build or Poll SCM which is one when any new source code is checked in your repository

Build → This describe how you want to execute build using command prompt or using any script

Post Build Actions → This describe once build is completed what do we want to do next we can ask to run other build , can ask to send notification and so on

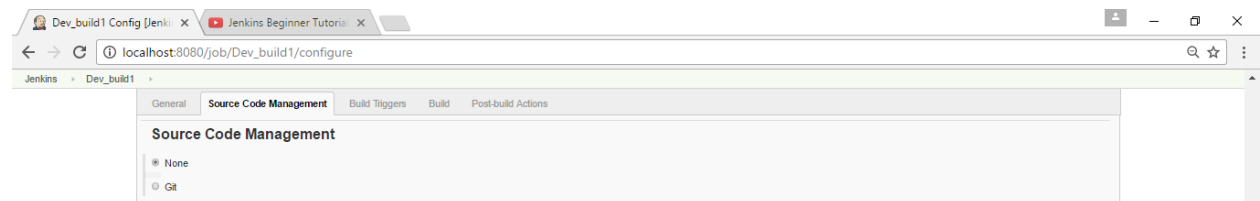
General

We creating job here and provided with short description , we will not be using build with any repository hence have not selected any. You have many other option that you can do with this job like disabled ,where this build to run and so on



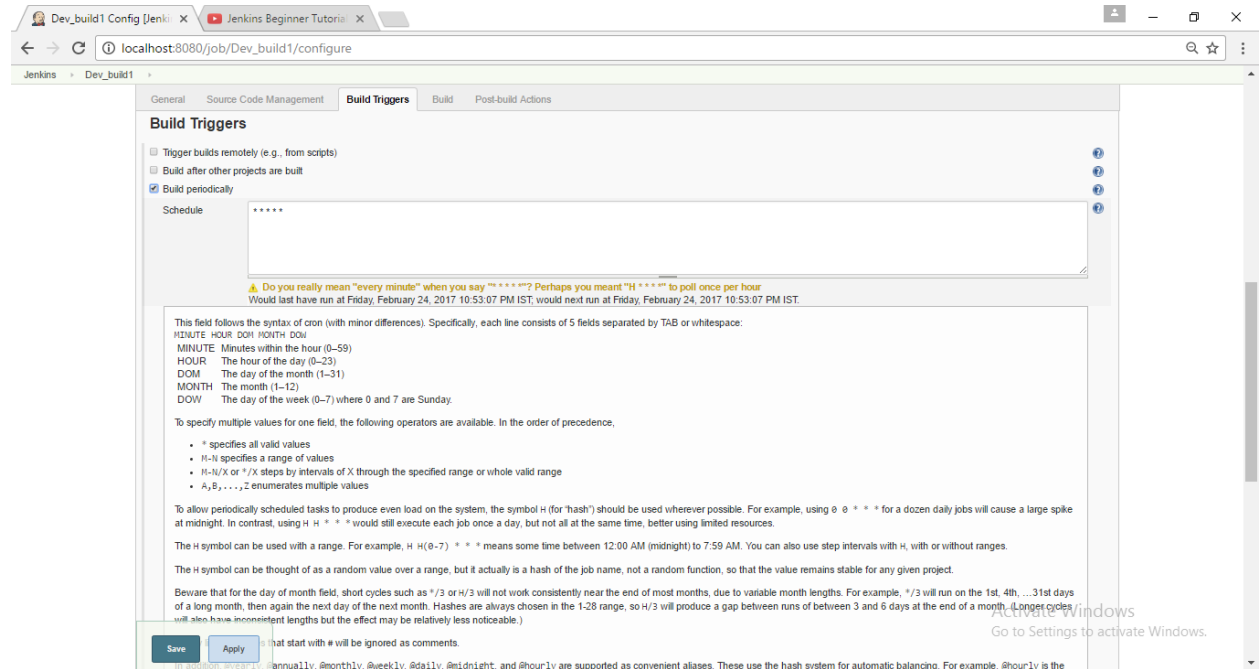
Source Code Management

If you want to pull your code for job build from any repository need to provide those details
If you using any repository make sure those plugin are installed



Build Triggers

How you want to execute your build, here I am using build periodically given 5 star
First position denotes Minute, Second Hour , third Day of Month, Fourth for Month, Fifth
for Day of Week . Giving star means valid value



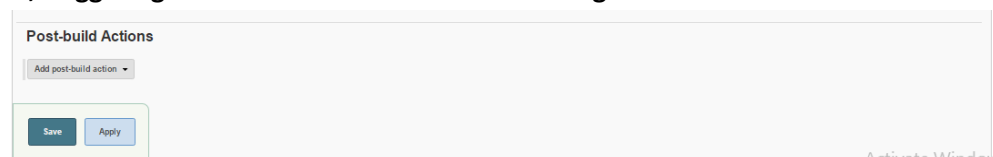
Build

How you want to execute your build through any script or from Git hub or will be from
any command, here we are using Windows batch command and we using Date command



Post Build Action

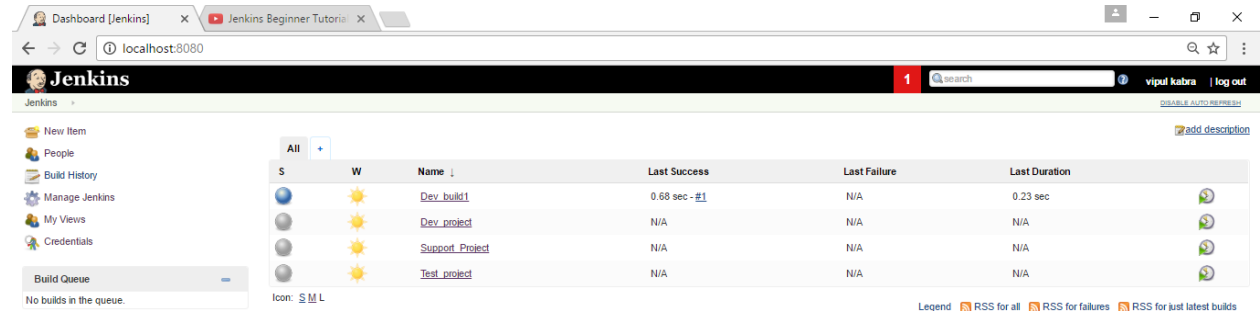
Once build is completed what is needed that needs to be configure here like sending email
notification, Triggering next build in this case we are taking no action



Click on Apply to create job

Your build/project entry is created on Dashboard. Since you have schedule to run every minute it has executed and been successful the color blue denote success. On failure it turns to red

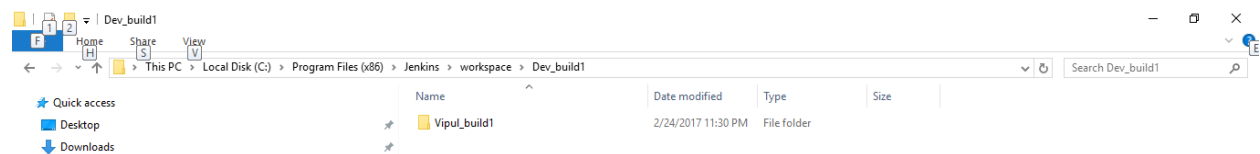
So now as per build command it has executed and created directory



See below console output of build successful



Showing Vipul_build1 directory created



Now our schedule was to run every minute to create same directory, since we already had this directory created when it tries to create again this is obvious it will fail.

Let see what happen after 1 success build

You can see on dashboard getting failure

The screenshot shows the Jenkins Dashboard at localhost:8080. The 'All' filter is selected, and the table lists four jobs: Dev_build1, Dev_project, Support_Project, and Test_project. Dev_build1 is marked as failed (red circle) and shows a last failure of 45 sec - #8. The left sidebar shows the 'Build Queue' with 'No builds in the queue'.

S	W	Name	Last Success	Last Failure	Last Duration
Failed	Failed	Dev_build1	7 min 45 sec - #1	45 sec - #8	0.23 sec
Success	Success	Dev_project	N/A	N/A	N/A
Success	Success	Support_Project	N/A	N/A	N/A
Success	Success	Test_project	N/A	N/A	N/A

Let's check the reason of failure

Saying directory exist

The screenshot shows the Jenkins Console Output for 'Dev_build1 #2'. The output indicates that the build failed because a subdirectory or file 'vipul_build1' already exists. The console output is as follows:

```

Started by timer
Building on master in workspace C:\Program Files (x86)\Jenkins\workspace\Dev_build1
[Dev_build1] $ cmd /c call C:\Windows\TEMP\udson6410488624261797013.bat
C:\Program Files (x86)\Jenkins\workspace\Dev_build1>mkdir vipul_build1
A subdirectory or file vipul_build1 already exists.
C:\Program Files (x86)\Jenkins\workspace\Dev_build1>exit 1
Build step 'Execute windows batch command' marked build as failure
Finished: FAILURE
  
```

Let see if Develop user login can he check this build since we have define pattern to view Dev jobs for him

The screenshot shows the Jenkins Dashboard at localhost:8080, filtered by the 'Develop' user. The table lists two jobs: Dev_build1 and Dev_project. Dev_build1 is marked as failed (red circle) and shows a last failure of 34 sec - #12. The left sidebar shows the 'Build Queue' with 'No builds in the queue'.

S	W	Name	Last Success	Last Failure	Last Duration
Failed	Failed	Dev_build1	11 min - #1	34 sec - #12	0.23 sec
Success	Success	Dev_project	N/A	N/A	N/A

Since this is developer build when Tester login he should not see this build of Dev_build1

The screenshot shows the Jenkins Dashboard at localhost:8080, filtered by the 'Tester' user. The table lists one job: Test_project. Test_project is marked as failed (red circle) and shows a last failure of N/A. The left sidebar shows the 'Build Queue' with 'No builds in the queue'.

S	W	Name	Last Success	Last Failure	Last Duration
Failed	Failed	Test_project	N/A	N/A	N/A

→Trigger job remotely

We will go to configure job

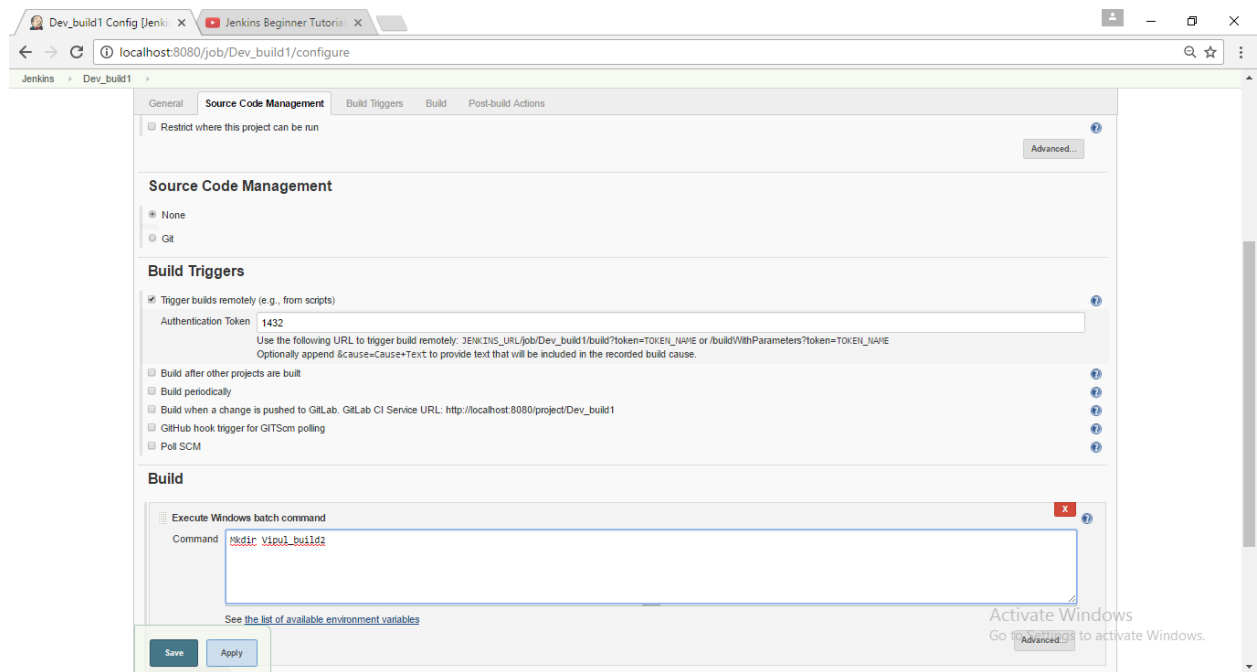
In Build Triggers we will uncheck Schedule and will check to trigger builds remotely

When we do this need to provide token number as well

We will also change Mkdir since that already exist as shown below

Jenkins will provide URL you need to use that to run this

`/job/Dev_build1/build?token=TOKEN_NAME` (Token Name is 1432)

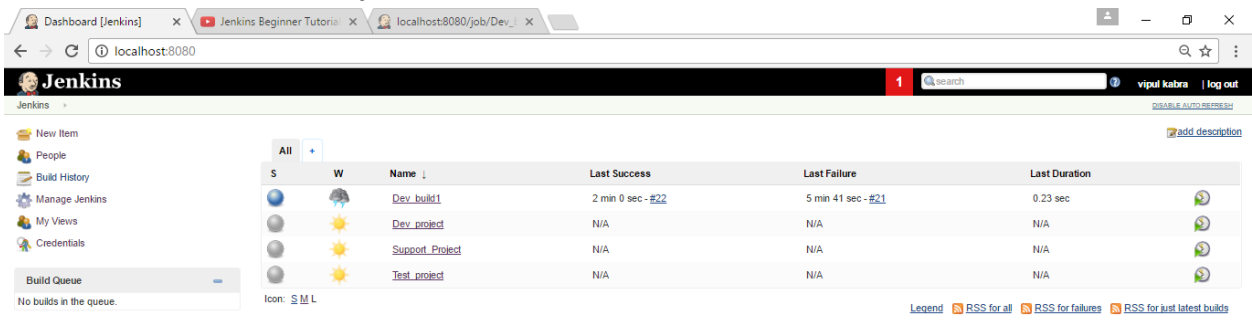


Save and Apply

Let's use link provide and token name and run it



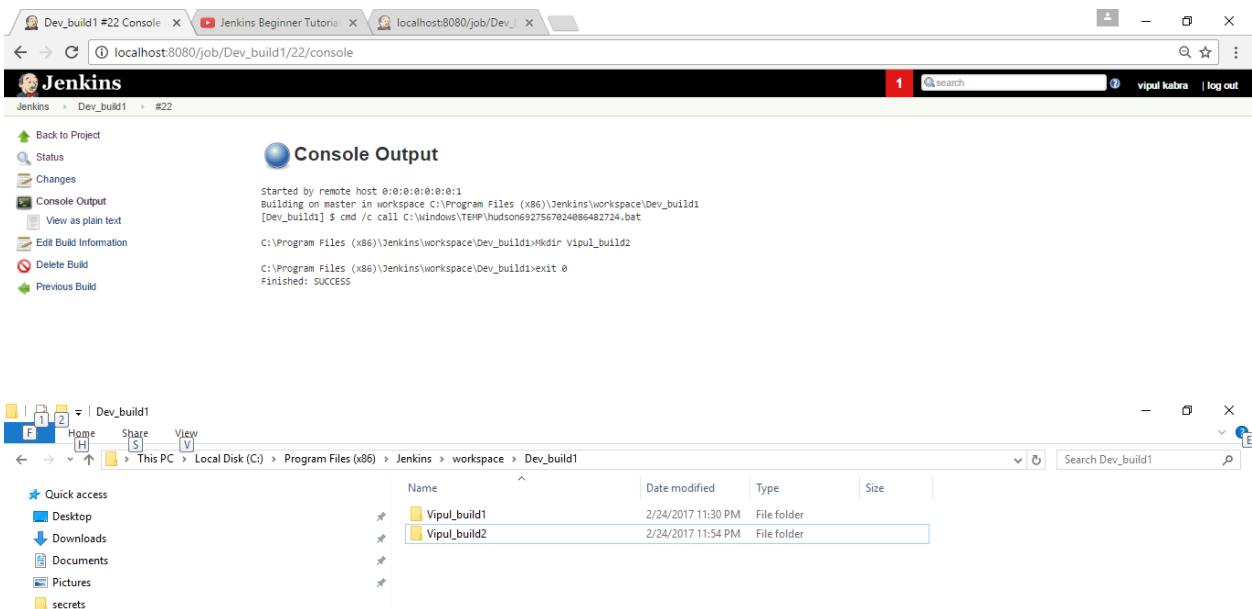
After build executed remotely let's see on Jenkin dashboard



The screenshot shows the Jenkins dashboard at localhost:8080. The left sidebar contains links for New Item, People, Build History, Manage Jenkins, My Views, and Credentials. The main area displays a table of builds for the 'Dev_build1' job. The table has columns for Status (S), Web icon (W), Name, Last Success, Last Failure, and Last Duration. The builds listed are 'Dev_build1', 'Dev_project', 'Support_Project', and 'Test_project'. Below the table, there is a 'Build Queue' section showing 'No builds in the queue.' and a 'Legend' section with links for RSS feeds.

S	W	Name	Last Success	Last Failure	Last Duration
		Dev_build1	2 min 0 sec - #22	5 min 41 sec - #21	0.23 sec
		Dev_project	N/A	N/A	N/A
		Support_Project	N/A	N/A	N/A
		Test_project	N/A	N/A	N/A

Let's see console output if they have created new directory called Vipul_build2 which we used to run remote trigger



The top screenshot shows the Jenkins console output for 'Dev_build1 #22'. The output text is as follows:

```
Started by remote host 0:0:0:0:0:0:1
Building on master in workspace C:\Program Files (x86)\jenkins\workspace\Dev_build1
[Dev_build1] $ cmd /c call C:\windows\TEMP\hudson6927567024086482724.bat
C:\Program Files (x86)\jenkins\workspace\Dev_build1>mkdir Vipul_build2
C:\Program Files (x86)\jenkins\workspace\Dev_build1>exit 0
Finished: SUCCESS
```

The bottom screenshot shows a Windows File Explorer window displaying the contents of the 'Dev_build1' directory. It lists two folders: 'Vipul_build1' and 'Vipul_build2', both created on 2/24/2017.

Name	Date modified	Type	Size
Vipul_build1	2/24/2017 11:30 PM	File folder	
Vipul_build2	2/24/2017 11:54 PM	File folder	

➔ **Trigger Test1_build after Build1_ is success (Trigger job based on dependency)**

We will trigger Dev_build1 (MkDIR VIPUL_CODE_Complete)

On Success

We will Trigger TEST_Build1 (MkDIR VIPUL_TEST_Complete)

On success

We will Trigger Support_Build1 (MkDIR SUPPORT_Complete)

Let's Create 2 new Job

Test_Build1 which will run after Dev_build1 is succeed

Support_build1 which will run after Test_build1 is succeed

Configure Dev_build1 so that it runs success

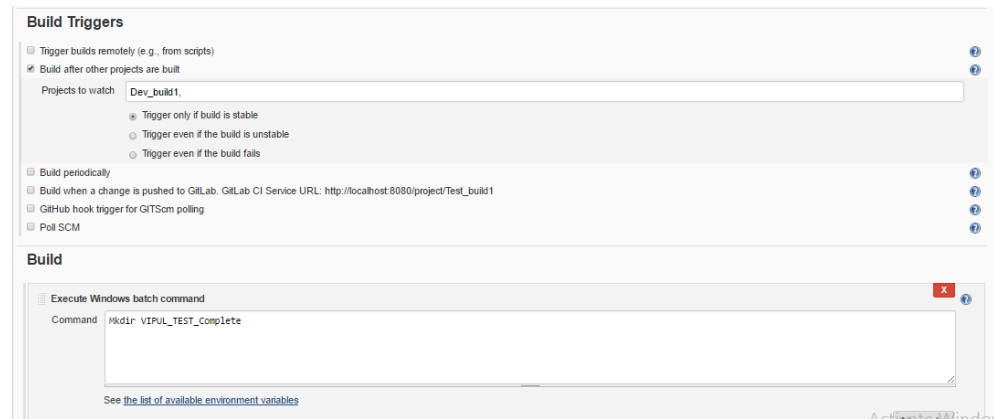
The screenshot shows the Jenkins Dashboard at localhost:8080. The left sidebar contains navigation links: New Item, People, Build History, Manage Jenkins, My Views, and Credentials. The main area displays a table of jobs with their build history.

S	W	Name ↓	Last Success	Last Failure	Last Duration
●	☀	Dev_build1	24 min - #22	27 min - #21	0:23 sec
●	☀	Dev_project	N/A	N/A	N/A
●	☀	Support_build1	N/A	N/A	N/A
●	☀	Support_Project	N/A	N/A	N/A
●	☀	Test_build1	N/A	N/A	N/A
●	☀	Test_project	N/A	N/A	N/A

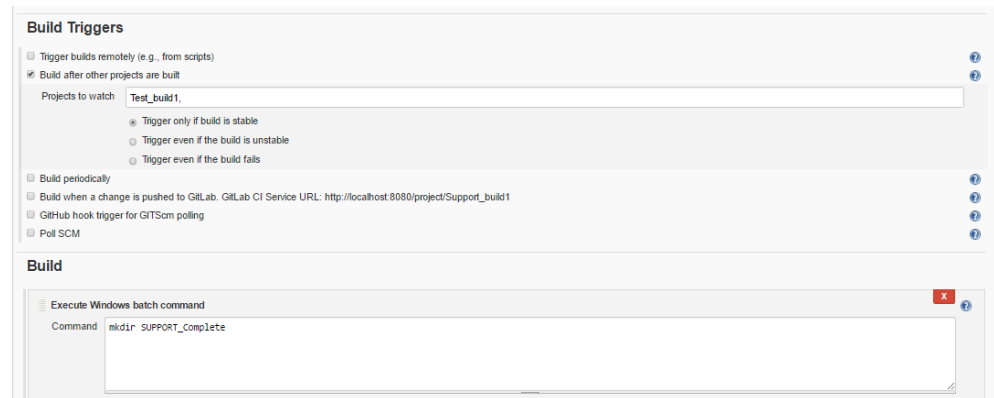
Below the table, there is a legend and links for RSS feeds: Legend, RSS for all, RSS for failures, and RSS for just latest builds.

Dev_build1 job Configuration

The screenshot shows the Jenkins job configuration for 'Dev_build1'. The 'Build' section is selected, and the 'Execute Windows batch command' option is chosen. The command field contains the text: `mkdir VIPUL_CODE_Complete`. Below the command field, there is a link to 'See the list of available environment variables' and an 'Advanced...' button.

TEST_Build1 job configuration

The screenshot shows the Jenkins configuration page for a job named 'TEST_Build1'. Under the 'Build Triggers' section, the option 'Build after other projects are built' is selected. The 'Projects to watch' field contains 'Dev_build1'. Three radio buttons are present: 'Trigger only if build is stable' (selected), 'Trigger even if the build is unstable', and 'Trigger even if the build fails'. Other triggers like 'Build periodically', 'Build when a change is pushed to GitLab', 'GitHub hook trigger for GITScm polling', and 'Poll SCM' are unchecked. The 'Build' section shows 'Execute Windows batch command' with the command 'mkdir VIFUL_TEST_Complete'. A link 'See the list of available environment variables' is at the bottom.

Support_Build1 job configuration

The screenshot shows the Jenkins configuration page for a job named 'Support_Build1'. Under the 'Build Triggers' section, the option 'Build after other projects are built' is selected. The 'Projects to watch' field contains 'Test_build1'. Three radio buttons are present: 'Trigger only if build is stable' (selected), 'Trigger even if the build is unstable', and 'Trigger even if the build fails'. Other triggers like 'Build periodically', 'Build when a change is pushed to GitLab', 'GitHub hook trigger for GITScm polling', and 'Poll SCM' are unchecked. The 'Build' section shows 'Execute Windows batch command' with the command 'mkdir SUPPORT_Complete'.

Let's Trigger Dev_build1 through remote giving URL and token

Jenkin Dashboard stating Dev Build 1 in Queue

The screenshot shows the Jenkins Dashboard at localhost:8080. The 'Build Queue (1)' section on the left indicates that 'Dev_build1' is in the queue. The main table lists the following jobs:

S	W	Name ↓	Last Success	Last Failure	Last Duration
🟢	🌤️	Dev_build1	4 min 42 sec - #24	44 min - #21	0.17 sec
🟡	🌤️	Dev_project	N/A	N/A	N/A
🔴	🌤️	Support_build1	4 min 32 sec - #3	4 min 22 sec - #4	0.47 sec
🟢	🌤️	Support_Project	N/A	N/A	N/A
🟢	🌤️	Test_build1	4 min 32 sec - #2	N/A	0.47 sec
🟢	🌤️	Test_project	N/A	N/A	N/A

The 'Build Executor Status' section shows the 'master' node with 1 idle executor and 2 offline executors.

Jenkin Dashboard Dev build 1 succeed and Test Build1 in Queue

The screenshot shows the Jenkins Dashboard at localhost:8080. The 'Build Queue (1)' section on the left indicates that 'Test_build1' is in the queue. The main table lists the following jobs:

S	W	Name ↓	Last Success	Last Failure	Last Duration
🟢	🌤️	Dev_build1	1.2 sec - #25	44 min - #21	0.17 sec
🟡	🌤️	Dev_project	N/A	N/A	N/A
🔴	🌤️	Support_build1	4 min 41 sec - #3	4 min 31 sec - #4	0.47 sec
🟢	🌤️	Support_Project	N/A	N/A	N/A
🟢	🌤️	Test_build1	4 min 41 sec - #2	N/A	0.47 sec
🟢	🌤️	Test_project	N/A	N/A	N/A

The 'Build Executor Status' section shows the 'master' node with 4 idle executors.

Jenkin Dashboard stating all 3 Job completed

The screenshot shows the Jenkins Dashboard at localhost:8080. The 'Build Queue' section on the left indicates 'No builds in the queue.' The main table lists the following jobs:

S	W	Name ↓	Last Success	Last Failure	Last Duration
🟢	🌤️	Dev_build1	22 sec - #25	45 min - #21	0.17 sec
🟢	🌤️	Dev_project	N/A	N/A	N/A
🟢	🌤️	Support_build1	2.3 sec - #5	4 min 52 sec - #4	0.24 sec
🟢	🌤️	Support_Project	N/A	N/A	N/A
🟢	🌤️	Test_build1	12 sec - #3	N/A	0.21 sec
🟢	🌤️	Test_project	N/A	N/A	N/A

The 'Build Executor Status' section shows the 'master' node with 4 idle executors.

Since all build succeeded MKdir command executed

The screenshot shows a file explorer view of the workspace directory. The structure is as follows:

- workspace
 - Dev_build1
 - Vipul_build1
 - Vipul_build2
 - VIPUL_CODE_Complete
 - Support_build1
 - SUPPORT_Complete
 - Test_build1
 - VIPUL_TEST_Complete