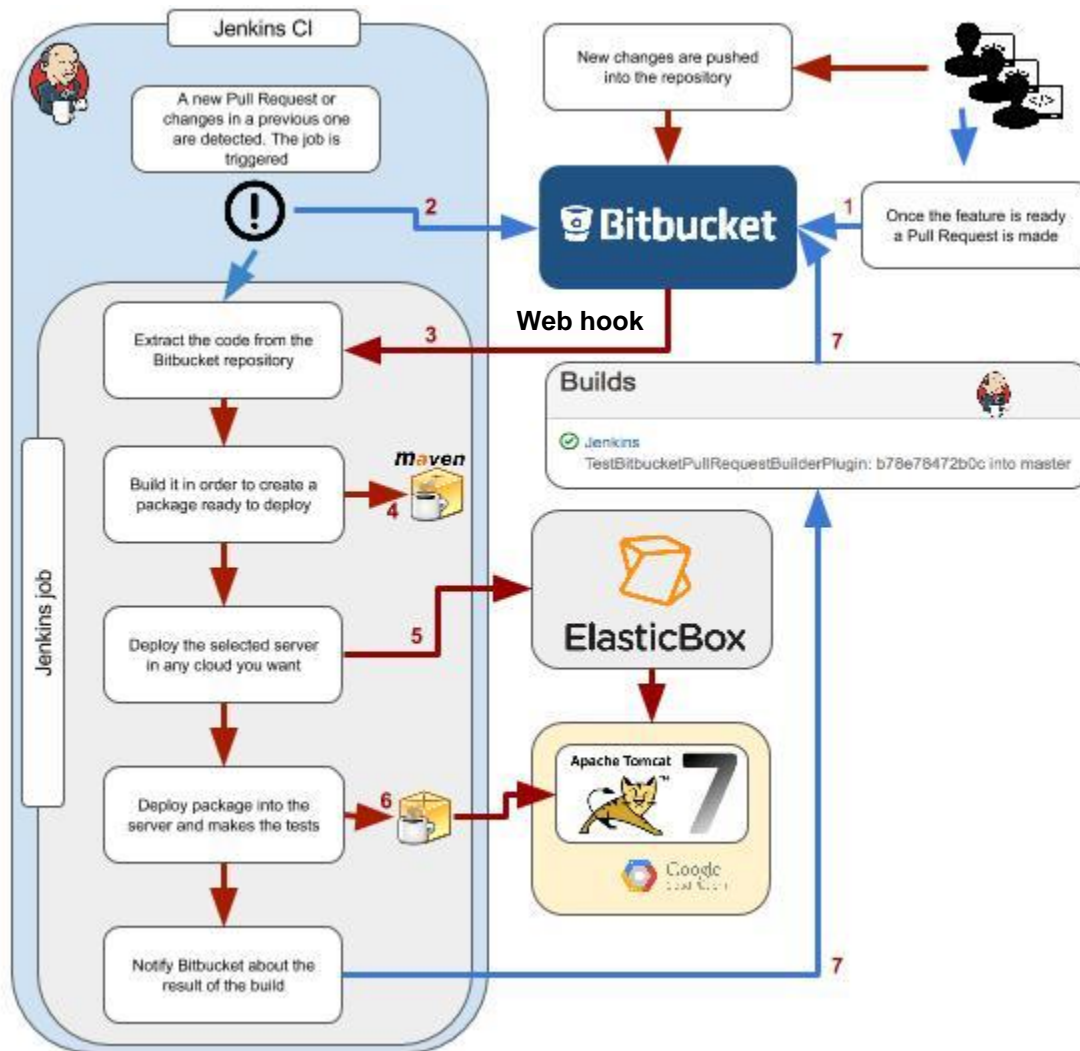


Jenkins - CI

# Jenkins Continuous Integration - WebHook

- Build is triggered automatically whenever there is a code check-in in the Repository



# Jenkins Continuous Integration – ngrok

- Goal is to kick off a Jenkins build when ever code is checked into GIT repository
- Firstly we need to expose a Jenkins server running behind a NAT or firewall to the internet
- *Note that this is done only for Training/academic purposes. You need to contact your network admins in your organization. Not recommended for Production scenarios*
- Since you may not be having a public IP we will be making use of a utility called *ngrok* to expose current IP address as a publicly available IP
- For more information on *ngrok* please visit <https://ngrok.com>
- Download the *ngrok* utility from <https://ngrok.com/download>
- Unzip the utility to a folder
- Open a command window
- Change directory to the folder containing *ngrok.exe*
- Since Jenkins is running on port 8080, lets try to expose it as a publicly available port

# Jenkins Continuous Integration - ngrok

- Run the command *ngrok http 8080*

```
C:\software\ngrok-stable-windows-amd64>ngrok http 8080_
```

- You can see that a local host 8080 is available as public IP
- <http://75aaaaf0.ngrok.io> This URL is important to configure web hook

```
C:\Administrator: Command Prompt - ngrok http 8080
ngrok by @inconshreveable (Ctrl+C to quit)

Session Status      online
Version             2.2.4
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://75aaaaf0.ngrok.io -> localhost:8080
Forwarding           https://75aaaaf0.ngrok.io -> localhost:8080

Connections         ttl    opn    rt1    rt5    p50    p90
                   0      0      0.00   0.00   0.00   0.00
```

# Bitbucket Plugin Installation

- Go to Manage Jenkins → Manage Plugins
- Search for Bitbucket under 'Available' tab
- Select the Bitbucket plugin and Install without restart



Bitbucket Plugin

Integrates with BitBucket

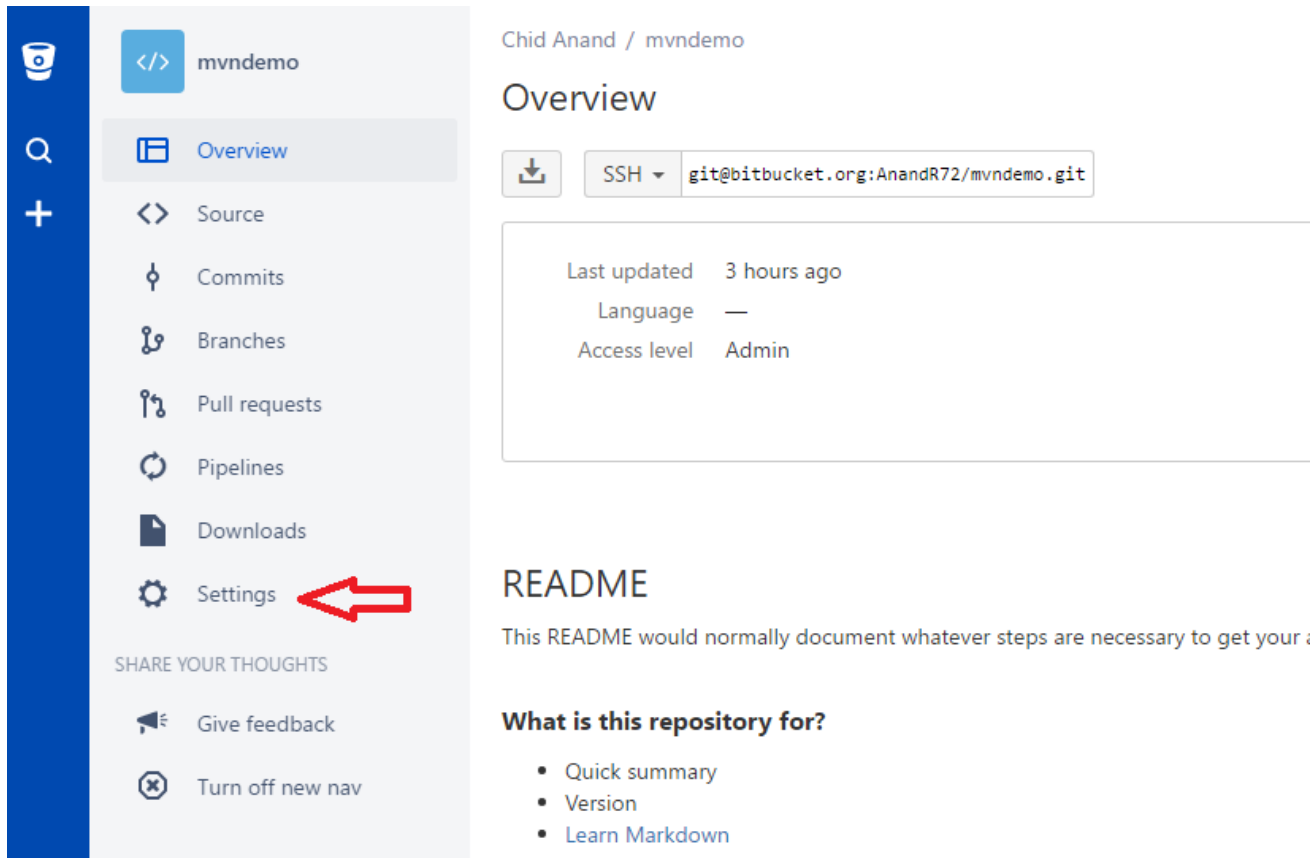
# Jenkins – Configure a job for CI

- Lets select the *mvndemo* job created in previous sessions for CI
- Select '*Configure*' on the job.
- Under *Build Triggers* – check the option, '*Build when change is pushed to BitBucket*'
- Note: This option will be available after the installation of plugin in previous step



# Bitbucket – Configure Web Hook


- Go to the Bitbucket repository and click on [Settings](#)



The screenshot displays the Bitbucket interface for a repository named 'mvndemo' by user 'Chid Anand'. The left sidebar contains a navigation menu with icons for Overview, Source, Commits, Branches, Pull requests, Pipelines, Downloads, and Settings. The 'Settings' option is highlighted with a red arrow. Below the navigation menu, there are links to 'SHARE YOUR THOUGHTS', 'Give feedback', and 'Turn off new nav'. The main content area shows the 'Overview' tab, which includes a download button, an SSH URL 'git@bitbucket.org:AnandR72/mvndemo.git', and repository details: 'Last updated 3 hours ago', 'Language —', and 'Access level Admin'. Below this is the 'README' section, which includes a heading 'What is this repository for?' and a list of links: 'Quick summary', 'Version', and 'Learn Markdown'.

Chid Anand / mvndemo

## Overview

 SSH `git@bitbucket.org:AnandR72/mvndemo.git`

Last updated	3 hours ago
Language	—
Access level	Admin

## README

This README would normally document whatever steps are necessary to get your i

### What is this repository for?

- Quick summary
- Version
- [Learn Markdown](#)

# Bitbucket – Configure Web Hook contd

- Next click on [Webhooks](#) link

The screenshot displays the Bitbucket interface for a repository named 'mvndemo'. On the left, a sidebar contains navigation links: Overview, Source, Commits, Branches, Pull requests, Pipelines, Downloads, and Settings (which is highlighted). Below these are links for 'Give feedback' and 'Turn off new nav'. The main content area is titled 'Settings' and includes a breadcrumb 'Chid Anand / mvndemo'. Under the 'GENERAL' section, there are links for 'Repository details', 'User and group access', 'Access keys', and 'Username aliases'. The 'WORKFLOW' section contains links for 'Branch permissions', 'Default reviewers', 'Webhooks' (highlighted with a red arrow), and 'Links'. The 'FEATURES' section includes links for 'Git LFS', 'Wiki', and 'Issue tracker'. On the right, the 'Repository details' section is visible, showing fields for 'Name' (mvndemo), 'Size' (410.8 KB), 'Git Large File Storage' (0 bytes of space used), 'Description', 'Access level' (checked for 'This is a private repository'), 'Forking' (set to 'Allow only private forks'), 'Landing page' (set to 'Overview'), and 'Website'.



# Bitbucket – Configure Web Hook contd

- Next click on [Add Webhook](#) Button

## Settings

### GENERAL

[Repository details](#)

[User and group access](#)

[Access keys](#)

[Username aliases](#)

### WORKFLOW

[Branch permissions](#)

[Default reviewers](#)

### Webhooks

[Links](#)

### FEATURES

[Git LFS](#)

## Webhooks

Webhooks allow you to extend what Bitbucket does when a request is merged).

To learn more about how webhooks work, check out the

[Add webhook](#)



Title	URL
Pipelines	<a href="https://bitbucket-pipelines.atlassian.io">https://bitbucket-pipelines.atlassian.io</a>
Pipelines	<a href="https://bitbucket-pipelines.atlassian.io">https://bitbucket-pipelines.atlassian.io</a>

# Bitbucket – Configure Web Hook contd

- Provide a meaningful title
- *Provide the publicly accessible URL generated by ngrok*
- *Do not forget to append **/bitbucket-hook/** at the end of the URL*
- *You can also note that the trigger is repo push*

Webhooks

Edit jenkins

To learn more about how webhooks work, check out the [documentation](#).

Title


URL

Status ☒ Active

Inactive webhooks don't trigger requests.

SSL / TLS ☐ Skip certificate verification

Untrusted or self-signed certificates may not be secure. [Learn more](#)

Triggers ☒ Repository push 

☐ Choose from a full list of triggers

# Testing the web hook

- All the configuration is done.
- Lets test the web hook trigger
- Open the GIT Bash and navigate to git repo under eclipse workspace
- Execute a git pull to bring changes from repository

MINGW64:/c/Users/HP/workspace/mvndemo

```
HP@DESKTOP-GP40G85 MINGW64 ~/workspace/mvndemo (master)
$ git pull origin master
From https://bitbucket.org/AnandR72/mvndemo
* branch      master      -> FETCH_HEAD
Already up-to-date.

HP@DESKTOP-GP40G85 MINGW64 ~/workspace/mvndemo (master)
$ |
```

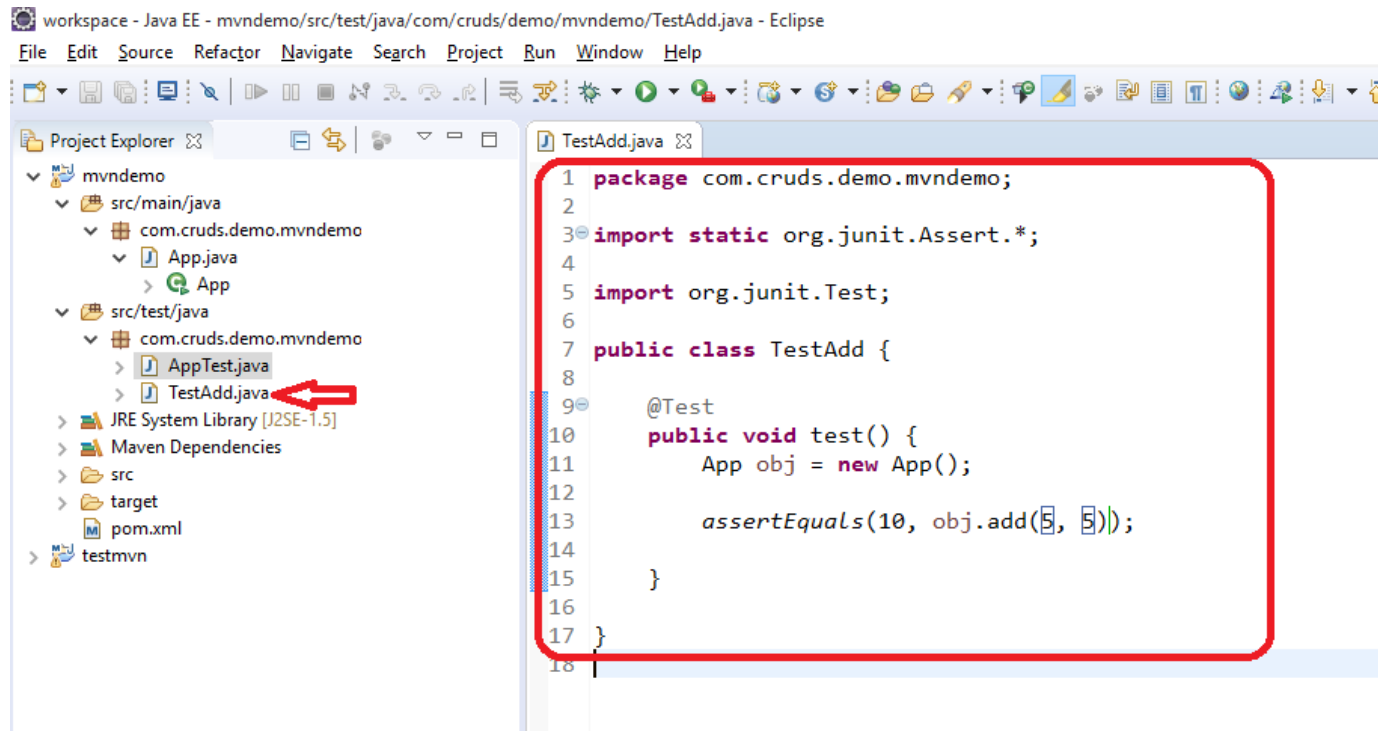
# Testing the web hook contd..

- Open Eclipse and Lets add another method to **App.java**

```
7 public class App
8 {
9     public static void main( String[] args )
10    {
11        System.out.println( "Hello World!" );
12    }
13
14    public String getGreeting()
15    {
16
17        return "Hello from Maven";
18    }
19
20    public int add(int x, int y)
21    {
22        return x + y;
23    }
24
25 }
```

# Testing the web hook contd..

- Lets also write another JUnit Test case to test the newly added method
- Right click on Test package → New → Other → Junit Test Case
- Give the name of the Test Case as *TestAdd*



The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer displays the project structure. The 'src/test/java' directory is expanded, showing 'AppTest.java' and 'TestAdd.java'. A red arrow points to 'TestAdd.java'. On the right, the TestAdd.java file is open in the editor, showing the following code:

```
1 package com.cruds.demo.mvndemo;
2
3 import static org.junit.Assert.*;
4
5 import org.junit.Test;
6
7 public class TestAdd {
8
9     @Test
10    public void test() {
11        App obj = new App();
12
13        assertEquals(10, obj.add(5, 5));
14    }
15 }
16
17
18
```

# Testing the web hook contd..

- *Now commit and push the changes to Git Repo using git bash*
- *This should trigger a build in Jenkins. Bitbucket will initiate build*

MINGW64:/c/Users/HP/workspace/mvndemo

```
HP@DESKTOP-GP40G85 MINGW64 ~/workspace/mvndemo (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   src/main/java/com/cruds/demo/mvndemo/App.java

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        src/test/java/com/cruds/demo/mvndemo/TestAdd.java

no changes added to commit (use "git add" and/or "git commit -a")

HP@DESKTOP-GP40G85 MINGW64 ~/workspace/mvndemo (master)
$ git add .

HP@DESKTOP-GP40G85 MINGW64 ~/workspace/mvndemo (master)
$ git commit -m'added new method add and test case'
[master db764fa] added new method add and test case
 2 files changed, 21 insertions(+)
 create mode 100644 src/test/java/com/cruds/demo/mvndemo/TestAdd.java

HP@DESKTOP-GP40G85 MINGW64 ~/workspace/mvndemo (master)
$ git push origin master
Counting objects: 17, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (17/17), 1.08 KiB | 0 bytes/s, done.
Total 17 (delta 2), reused 0 (delta 0)
To https://bitbucket.org/AnandR72/mvndemo.git
 3dd32cd..db764fa  master -> master
```