Developers workstation setup Guide.

Developer Workstation Setup Guide step-by-step:

In this series I will introduce you to the tools of the trade through the development of Developer workstation.

The tools of the DevOps engineers are new and most of them are in the active development phase with frequent releases. Some of the new versions have show stopping bugs in them, so it is a good idea to test all new versions of the tools before you uninstall the old one.

To work as a DevOps engineer you need a development environment with multiple tools. Luckily all of them are available for free and easy to set up.

- Version control system - GIT.
- IDE's -Eclipse, VS code, Intellij.
- putty, win scp.
- Bitbucket and jira
- jenkins, nexus.

You can do all development and testing on your workstation for free, but to see your scripts running in a real cloud, you can set up an account at a cloud provider.

Amazon Web Services (AWS) offers a free tier where you can launch small server instances for free.

You can use Mac, Windows, or Linux computer as a workstation. I have separated the Windows, and Linux development tool setup.

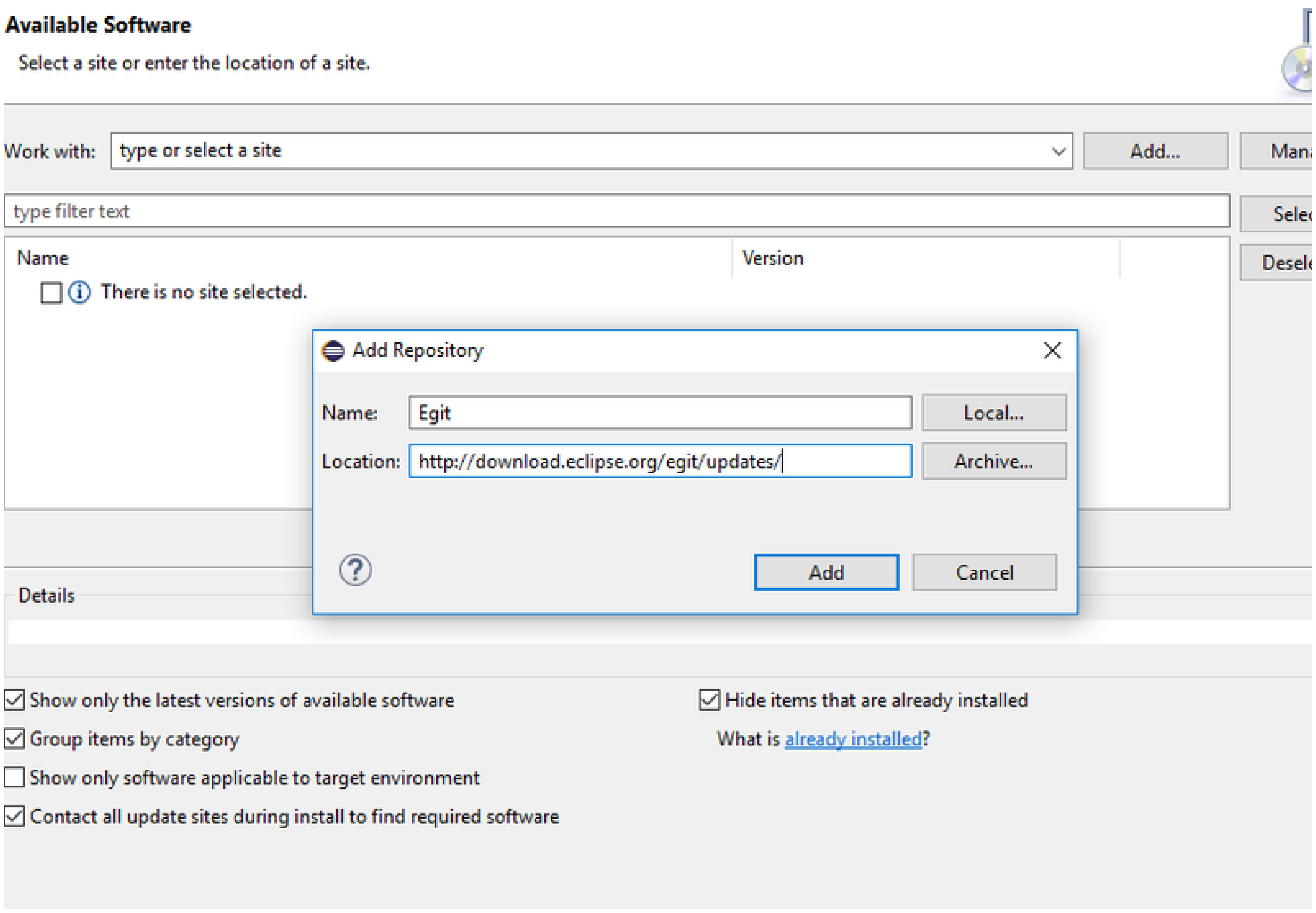
Setup the IDE tools:-

Install the IDE software tools i.e eclipse, VS code, Intellij tools.

You need to install the git extension in the tool itself.

Eclipse IDE installation and setup

1. Install Eclipse IDE and version is latest. link https://www.eclipse.org/downloads/.
2. Install new software in eclipse i.e git.



Available Software

Select a site or enter the location of a site.

Work with: type or select a site

type filter text

Name	Version
☐ There is no site selected.	

Add Repository

Name: Egit

Location: http://download.eclipse.org/egit/updates/

Details

☑ Show only the latest versions of available software

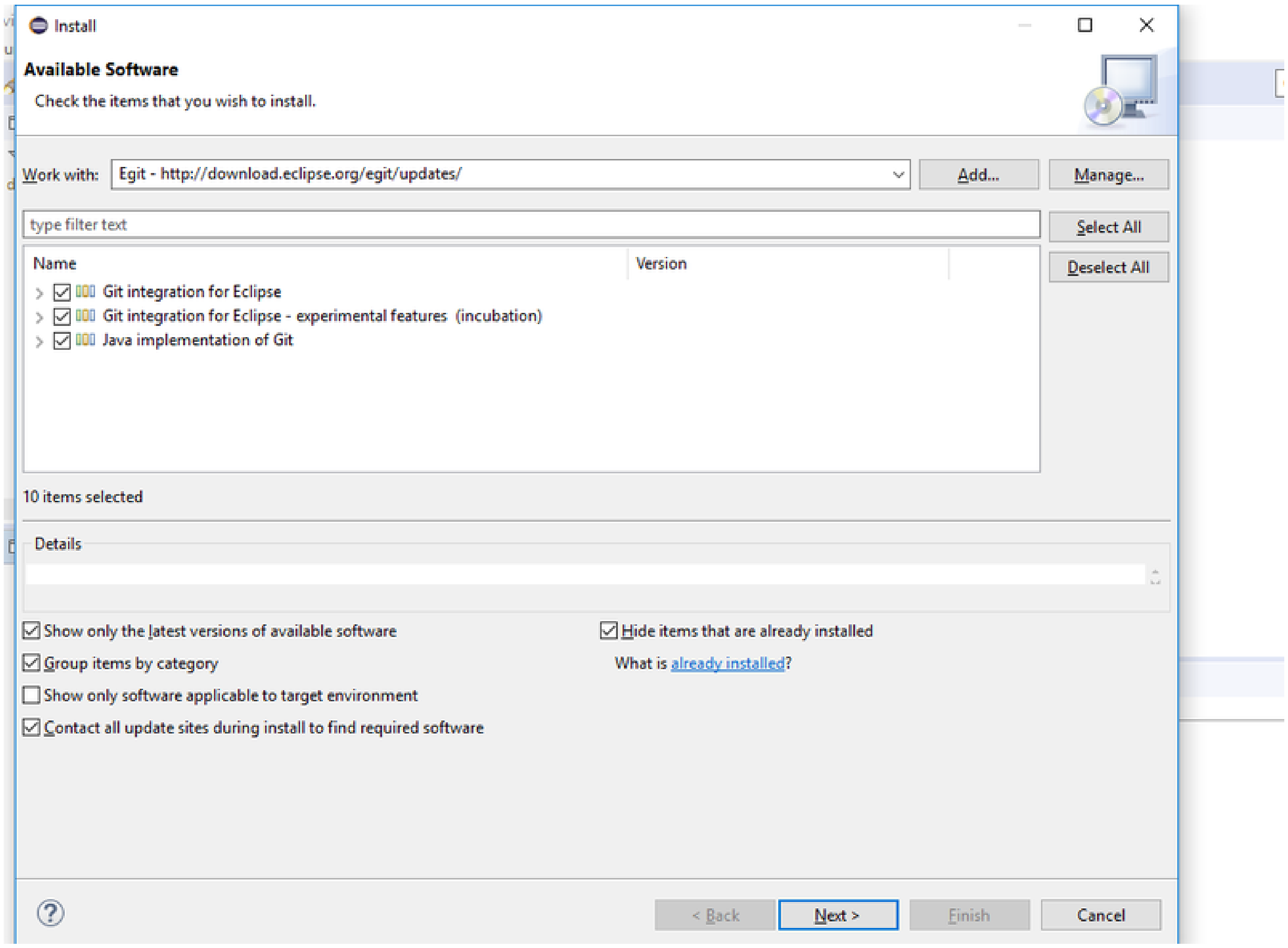
☑ Group items by category

☐ Show only software applicable to target environment

☑ Contact all update sites during install to find required software

☑ Hide items that are already installed

What is already installed?



Available Software

Check the items that you wish to install.

Work with: Egit - http://download.eclipse.org/egit/updates/

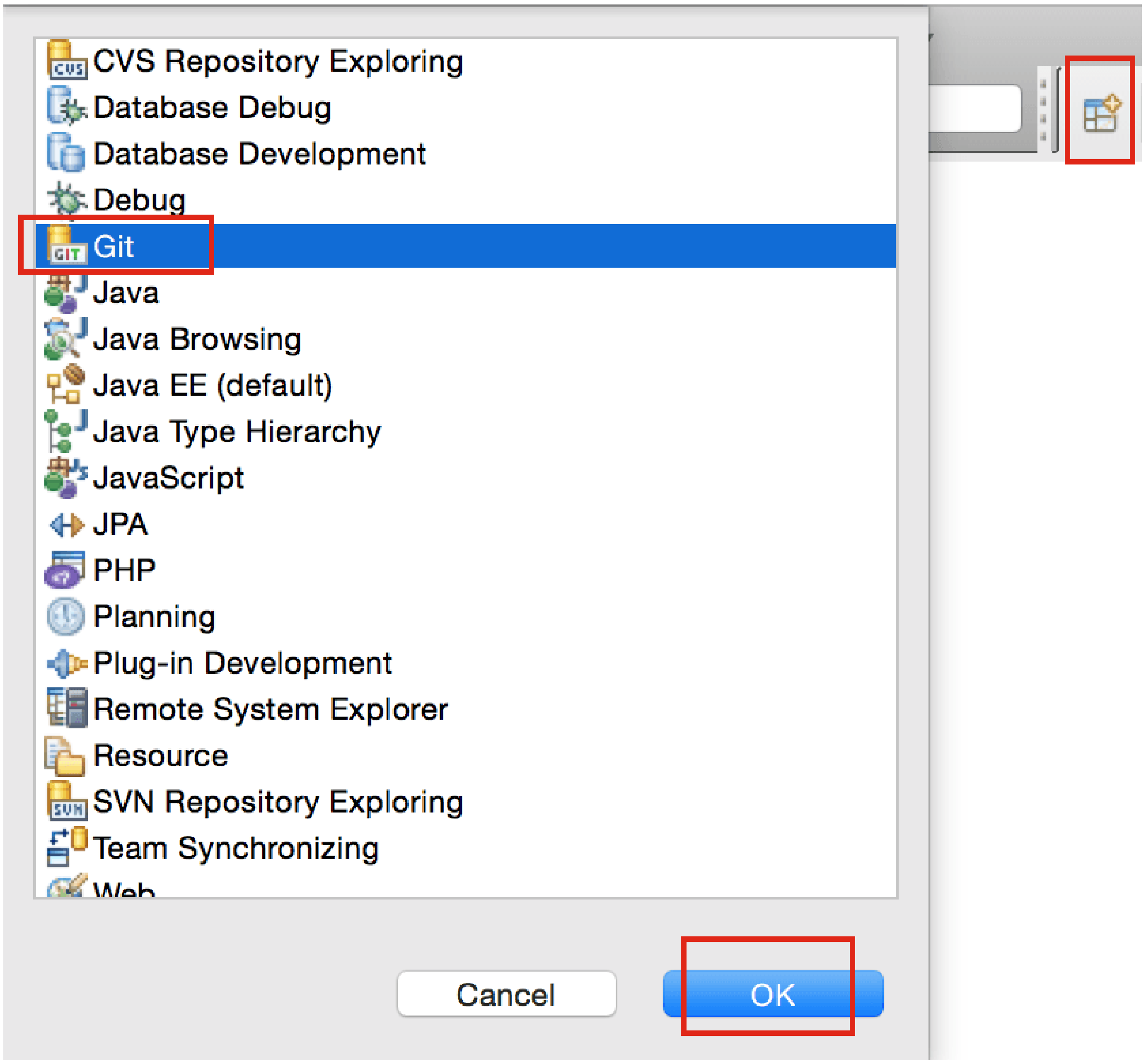
Name	Version
☑ Git integration for Eclipse	
☑ Git integration for Eclipse - experimental features (incubation)	
☑ Java implementation of Git	

10 items selected

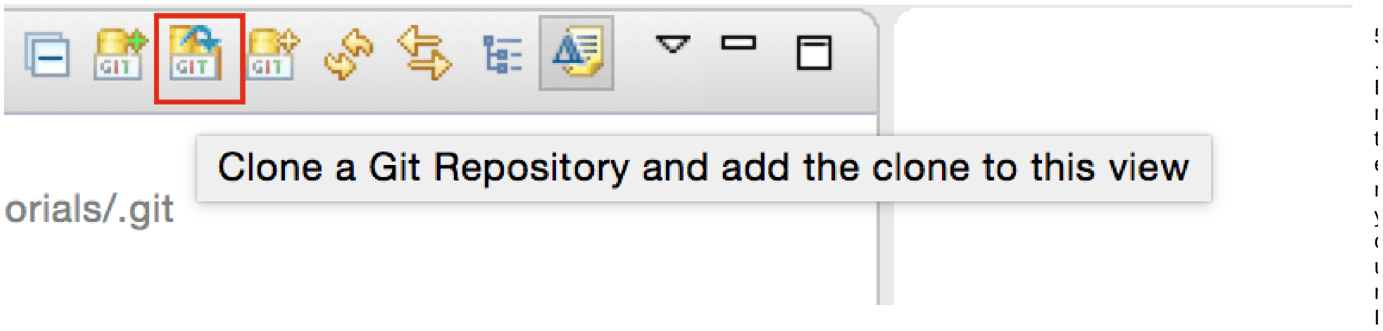
Details

☑ Show only the latest versions of available software
☑ Group items by category
☐ Show only software applicable to target environment
☑ Contact all update sites during install to find required software
☑ Hide items that are already installed
What is already installed?

3. Now Open `Perspective` and choose `Git` from list.

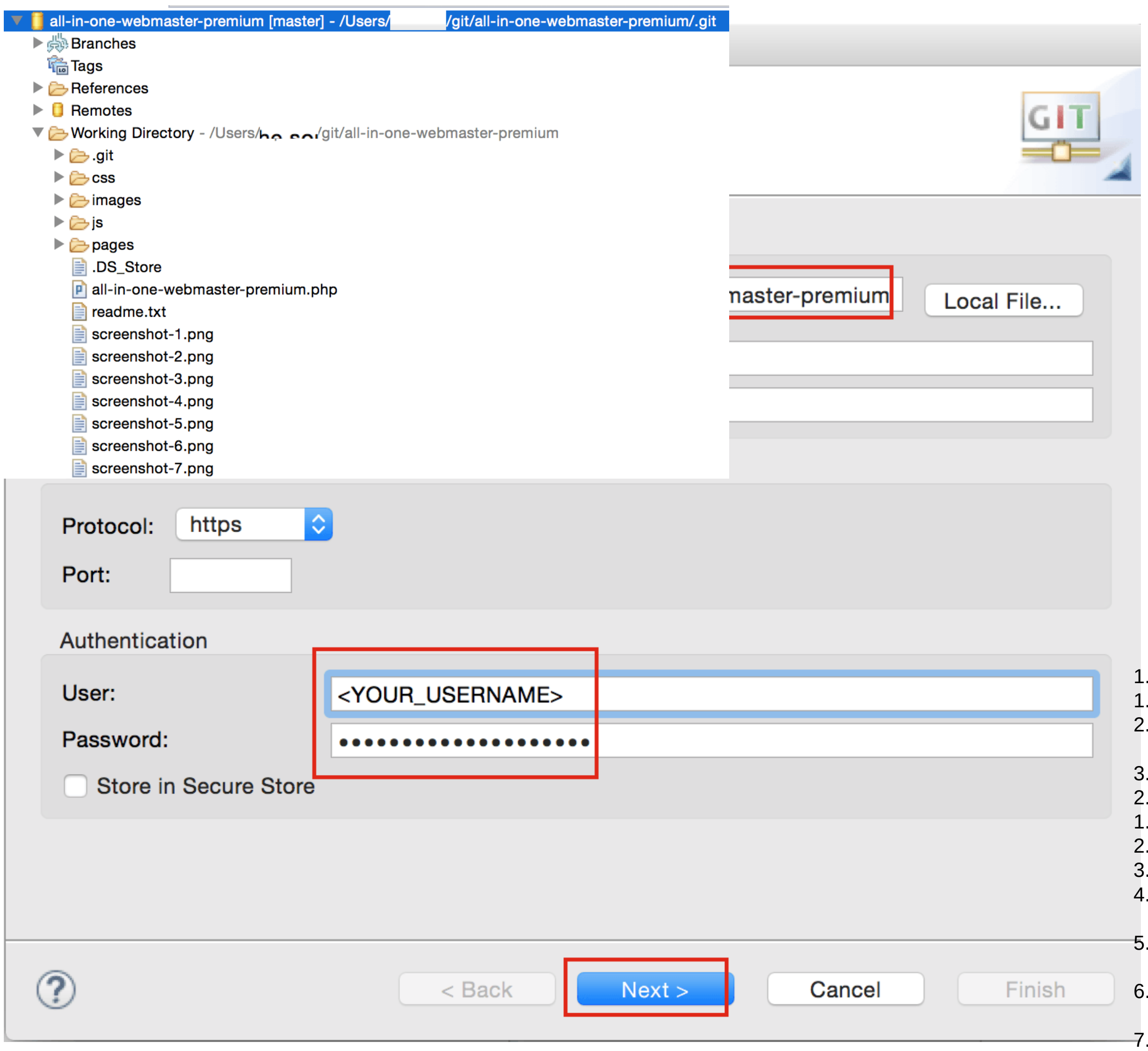


4.Click Clone Repository.



itbucket URL and User Information as mentioned in below diagram. Click `Next` and `Finish`. No need to change other configuration in next window.

6.You should see your Bitbucket repository now in eclipse.


all-in-one-webmaster-premium [master] - /Users/[redacted]/git/all-in-one-webmaster-premium/.git
Branches
Tags
References
Remotes
Working Directory - /Users/[redacted]/git/all-in-one-webmaster-premium
.git
css
images
js
pages
.DS_Store
all-in-one-webmaster-premium.php
readme.txt
screenshot-1.png
screenshot-2.png
screenshot-3.png
screenshot-4.png
screenshot-5.png
screenshot-6.png
screenshot-7.png

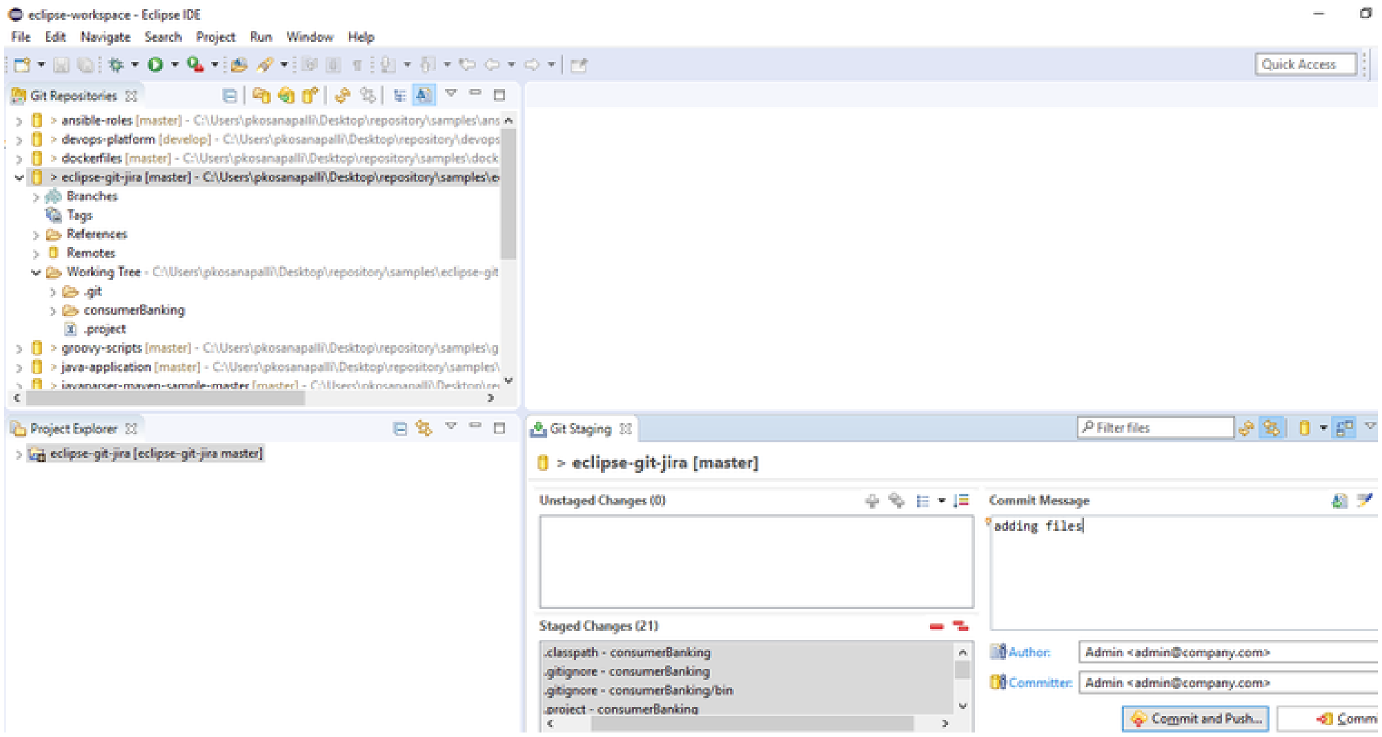
master-premium

Protocol: https
Port:

Authentication
User: <YOUR_USERNAME>
Password: ••••••••••••••••••••
Store in Secure Store

7.Now you may want to import the project so you can work on the source code.

1. Click
   1. ‘Windows’ >
   2. ‘Open Perspective’ >
   3. ‘Resource’
2. Click
   1. ‘File’ >
   2. ‘Import’ >
   3. ‘Git’ >
   4. ‘Projects from Git’ >
   5. ‘Existing local repository’ >
   6. ‘Select a Git Repository’ >
   7. ‘Import as General Project’ >
   8. ‘Next’ >
   9. ‘Finish’
3. The code should then appear in your ‘Project Explorer’ window as a normal project
4. Now make changes to your file as you want
5. Look at ‘Git Staging’ view to see your changed files and Click ‘`Commit and Push`‘



```
> ansible-roles [master] - C:\Users\pkosanapalli\Desktop\repository\samples\ans
> devops-platform [develop] - C:\Users\pkosanapalli\Desktop\repository\devops
> dockerfiles [master] - C:\Users\pkosanapalli\Desktop\repository\samples\dock
> eclipse-git-jira [master] - C:\Users\pkosanapalli\Desktop\repository\samples\e
    Branches
    Tags
    References
    Remotes
    Working Tree - C:\Users\pkosanapalli\Desktop\repository\samples\eclipse-git
        .git
        consumerBanking
        .project
> groovy-scripts [master] - C:\Users\pkosanapalli\Desktop\repository\samples\g
> java-application [master] - C:\Users\pkosanapalli\Desktop\repository\samples\
> javaparser-maven-sample-master [master] - C:\Users\pkosanapalli\Desktop\re
```

eclipse-git-jira [eclipse-git-jira master]

> eclipse-git-jira [master]

Staged Changes (21):

```
.classpath - consumerBanking
.gitignore - consumerBanking
.gitignore - consumerBanking/bin
.project - consumerBanking
```

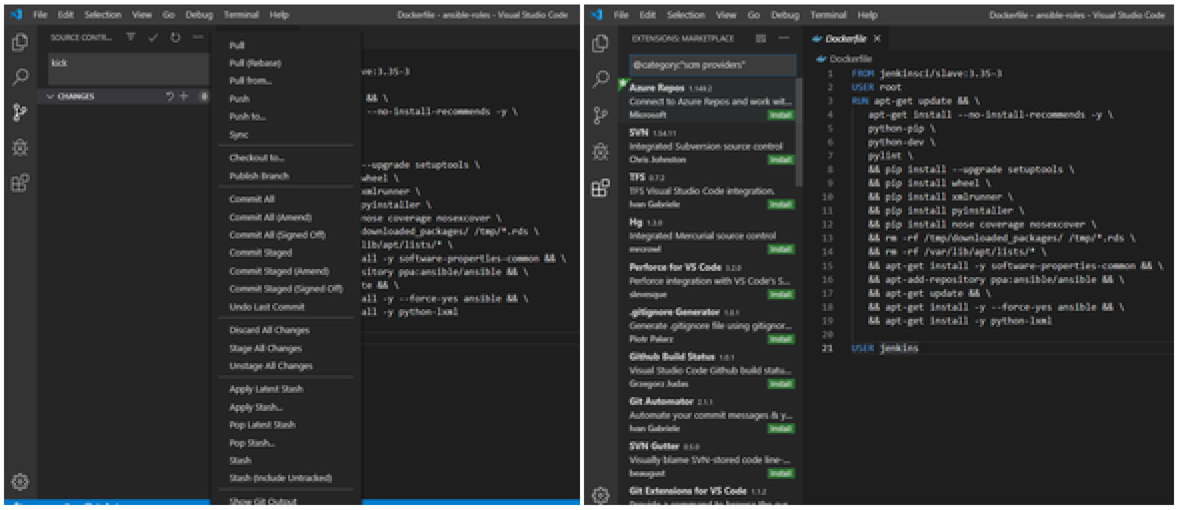
Commit Message: adding files

Author: Admin <admin@company.com>

Committer: Admin <admin@company.com>

VS code IDE installation and setup

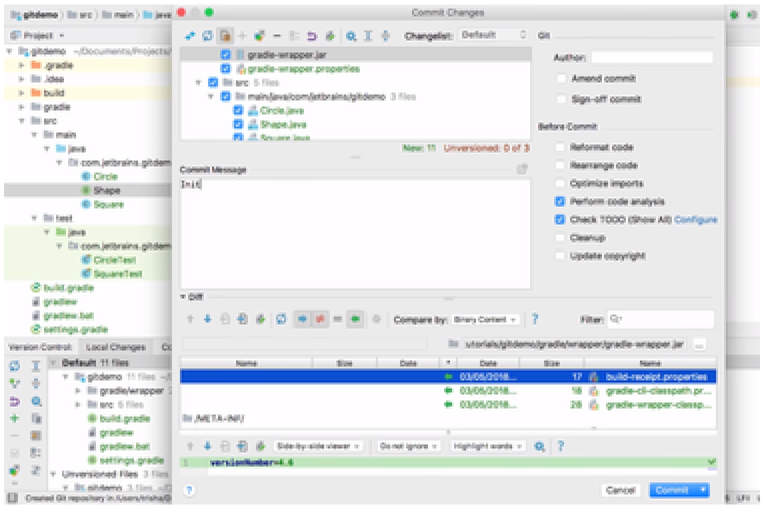
1. Install VS Code IDE and version is latest. link https://code.visualstudio.com/download.
2. Add the VCS extersion in vs code i.e git.



```
FROM jenkinsci/slave:3.35-3
USER root
RUN apt-get update && \
    apt-get install --no-install-recommends -y \
    python-pip \
    python-dev \
    pylint \
    && pip install --upgrade setuptools \
    && pip install wheel \
    && pip install xmlrunner \
    && pip install pyinstaller \
    && pip install nose coverage nosexcover \
    && rm -rf /tmp/downloaded_packages/ /tmp/*.rds \
    && rm -rf /var/lib/apt/lists/* \
    && apt-get install -y software-properties-common && \
    && apt-add-repository ppa:ansible/ansible && \
    && apt-get update && \
    && apt-get install -y --force-yes ansible && \
    && apt-get install -y python-lxml

USER jenkins
```

Intelliji IDE installation and setup

1. Install intelliji IDE and version is latest. link https://www.jetbrains.com/idea/download/#section=windows
2. Install new software in intelliji i.e git.
3. Add the git VCS in the project explorer.


Commit Changes

Changelist: Default

gradle-wrapper.jar
gradle-wrapper.properties
src 5 files
main/java/com/jetbrains/gitdemo 3 files
Circle.java
Shape.java
Square.java

New: 11 Unversioned: 0 of 3

Commit Message

```
Init
```

Author:
Amend commit
Sign-off commit

Before Commit
Reformat code
Rearrange code
Optimize imports
[✓ Perform code analysis]
[✓ Check TODO (Show All)] Configure
Cleanup
Update copyright

Compare by: Binary Content

Name	Size	Date	Date	Size	Name
			03/05/2018...	17	build-receipt.properties
			03/05/2018...	18	gradle-cli-classpath.pr...
			03/05/2018...	28	gradle-wrapper-classp...

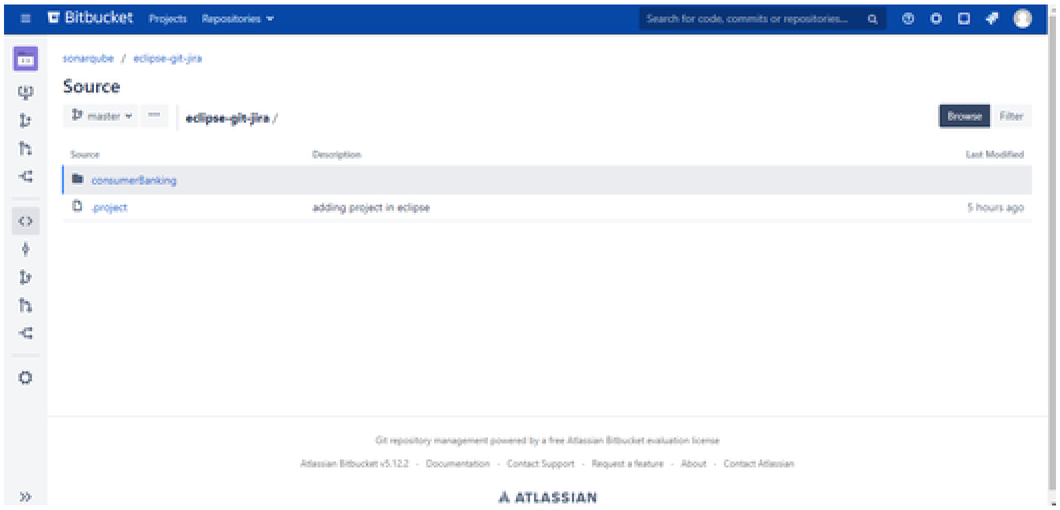
```
versionNumber=4.6
```

Cancel Commit

Setup the project repo:-

Here we need to create project and configure the repo based on our requirements.

1. Login to repo with credentials, and create a project create/import repo clone into local desk or in local IDE.
2. for more details please go to this document How to - Bitbucket repo setup - Create sample scala repo for sample scala project in intelliji ide.


sonarqube / eclipse-git-jira

Source

master eclipse-git-jira /

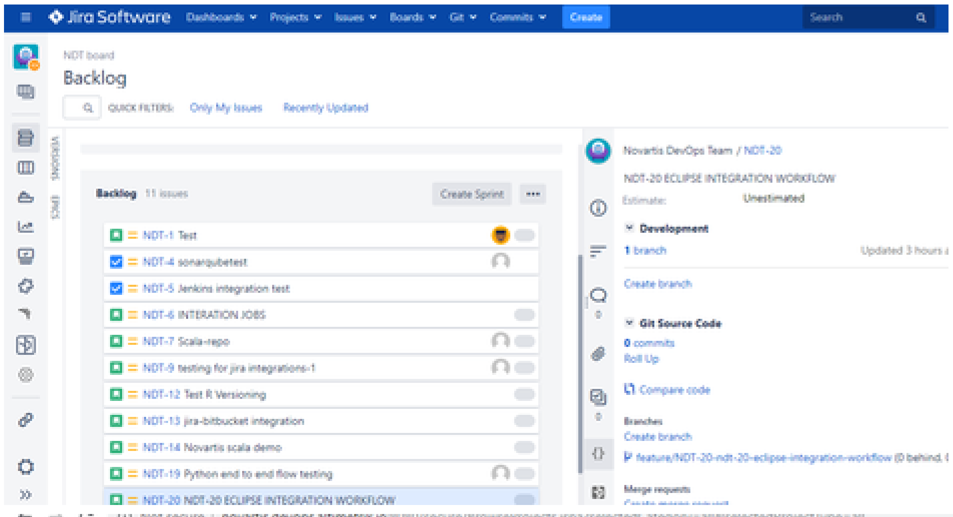
Source	Description	Last Modified
consumerBanking		
.project	adding project in eclipse	5 hours ago

Git repository management powered by a free Atlassian Bitbucket evaluation license

Atlassian Bitbucket v5.12.2 · Documentation · Contact Support · Request a feature · About · Contact Atlassian

Setup the Jira tool:-

1. first step is to install Jira software in the instance. for reference How To - Jira Configuration .
2. You need to create a project and board in the Jira dashbord.


NDT board

Backlog

QUICK FILTERS: Only My Issues Recently Updated

Backlog 11 issues

- NDT-1 Test
- NDT-4 sonarqubetest
- NDT-5 Jenkins integration test
- NDT-6 INTERATION JOBS
- NDT-7 Scala-repo
- NDT-9 testing for jira integrations-1
- NDT-12 Test R Versioning
- NDT-13 jira-bitbucket integration
- NDT-14 Novartis scala demo
- NDT-19 Python end to end flow testing
- NDT-20 NDT-20 ECLIPSE INTEGRATION WORKFLOW

Novartis DevOps Team / NDT-20

NDT-20 ECLIPSE INTEGRATION WORKFLOW

Estimate: Unestimated

Development

1 branch Updated 3 hours a

Create branch

Git Source Code

0 commits

Roll Up

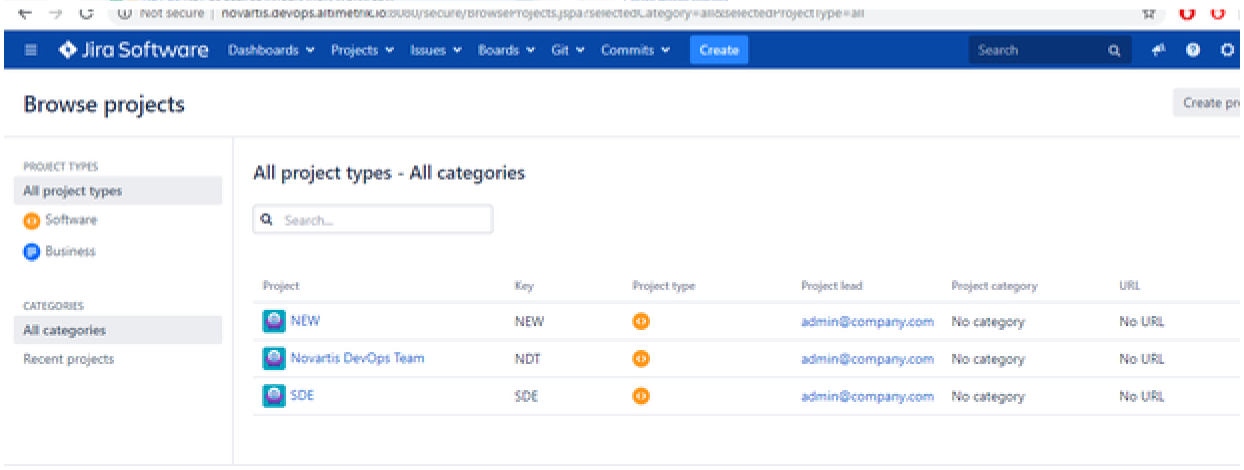
Compare code

Branches

Create branch

feature/NDT-20-ndt-20-eclipse-integration-workflow (0 behind, (

Merge requests


Not secure | novartis.devops.altimetrik.io:8080/secure/BrowseProjects.jspa?selectedCategory=all&selectedProjectType=all

Browse projects

PROJECT TYPES: All project types, Software, Business

CATEGORIES: All categories, Recent projects

All project types - All categories

Project	Key	Project type	Project lead	Project category	URL
NEW	NEW	●	admin@company.com	No category	No URL
Novartis DevOps Team	NDT	●	admin@company.com	No category	No URL
SDE	SDE	●	admin@company.com	No category	No URL

3. You need to create a workflow for the issues in a board, so that we can be smooth transition of issue progress status. for reference - How to Create a New Workflow in Jira

4. You can do the background tasks for auto transition of issue - How to - Jira Auto-transition.

5. We need to configure the issue model screen for smooth visualization& field flow details- How To - Jira - Screen Definitions.

6. Now we need to create a issue- How To Create Epic and Initiative workflows and screens in Jira..

7.For complete Jira workflow - How to - Jira Full developer flow and types of Issues (Epic, Story and Bugs) .

Setup the Jenkins:-

1. Install the Jenkins server in the instance and configure the url – How To - Jenkins Configuration.
2. Install the plugins and configure the global settings for the tools in manage jenkins. --How to - Install jenkins and configuration on Cloud.
3. Create the pipeline job for continuous integration and continuous deployment for java/nodejs/python/asp.net application.
4. specify the repository url and authentication and build tool.
5. Specify the target for destination deployment of the application.
6. The package is deployed in the server.

Setup the Nexus server:-

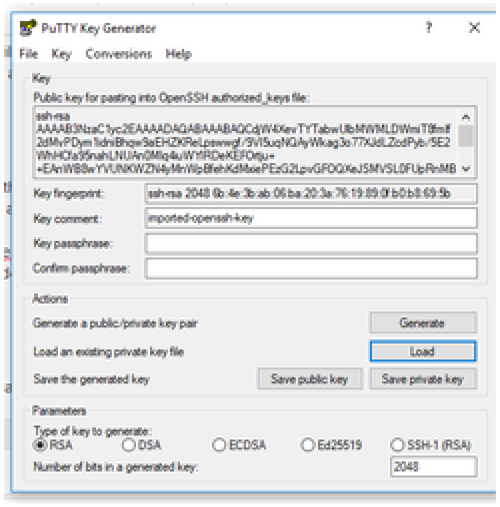
1. Install the nexus server in the instance and login with the user credentials.
2. create a deployment repository in the nexus server and specify the type of application i.e maven, nodejs, python, docker etc -Nexus Artifactory Configurations With Basic Maven Repository.
3. Add the nexus plugins and configure the server in jenkins.
4. Now integrate the jenkins and nexus for continous deployment- How To - Nexus Jenkins Integration .

Setup the putty& Winscp tools:-

1. install putty from the https://www.putty.org/ .
2. create a instance from the EC2 instace and download the putty key.
3. Provide the full access for pem file linux command.

```
chmod 700 pemfile.pem
```

4. Convert the pem file into private key and save it.



Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCdjW4XevTYTabwUbMWMLDWmiT8mf
2dMvPDym1idniBhqw3aEHZKRelpswwgf/9Vl5uqNQAyWkag3o77XJdLZodPyb/5E2
WhHCfa95nahLNUAn0Mlq4uWYtRDeKEFOtju+
+EAnWB8wYVUNKWZN4yMnWpBfehKdMxePEzG2LpvGFOQXeJSMVSL0FUpRnMB
```

Key fingerprint: ssh-rsa 2048 6b:4e:3b:ab:06:ba:20:3a:76:19:89:0f:b0:b8:69:5b

Key comment: imported-openssh-key

Key passphrase:

Confirm passphrase:

Actions

Generate a public/private key pair

Load an existing private key file

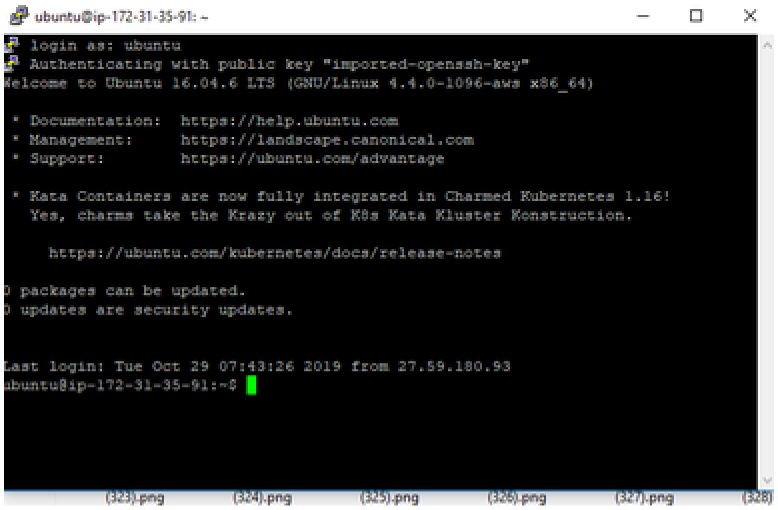
Save the generated key

Parameters

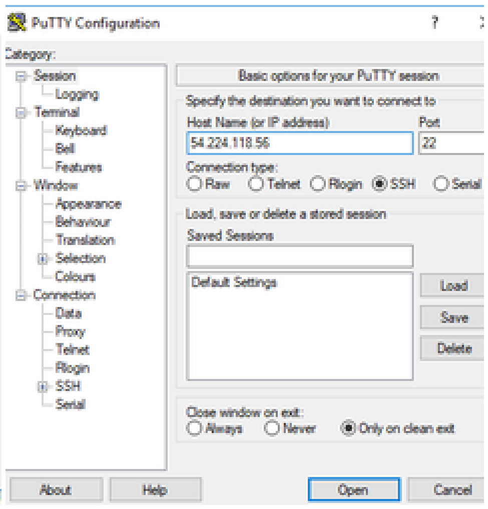
Type of key to generate:
[✓ RSA] DSA ECDSA Ed25519 SSH-1 (RSA)

Number of bits in a generated key: 2048

5. Now open putty and enter ip address of instance and insert the private key, login to your server. A new screen appers, thats you ec2 instance.



```
 login as: ubuntu
 Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1096-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Kata Containers are now fully integrated in Charmed Kubernetes 1.16!
   Yes, charms take the Krazy out of K8s Kata Kluster Konstruction.

     https://ubuntu.com/kubernetes/docs/release-notes

0 packages can be updated.
0 updates are security updates.


Last login: Tue Oct 29 07:43:26 2019 from 27.59.180.93
ubuntu@ip-172-31-35-91:~$
```



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 54.224.118.56

Port: 22

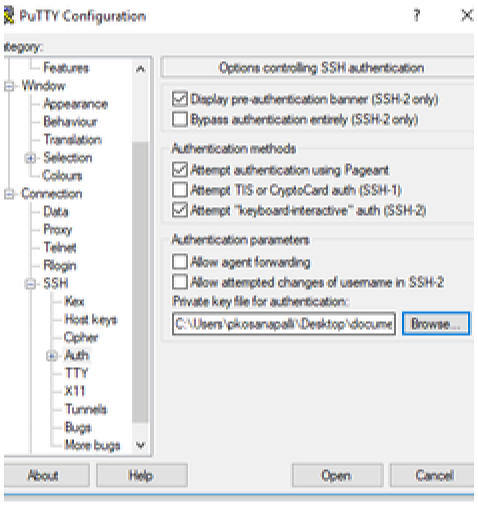
Connection type:
Raw Telnet Rlogin [✓ SSH] Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit:
Always Never [✓ Only on clean exit]



PuTTY Configuration

Options controlling SSH authentication

- [✓] Display pre-authentication banner (SSH-2 only)
- [ ] Bypass authentication entirely (SSH-2 only)

Authentication methods

- [✓] Attempt authentication using Pageant
- [ ] Attempt TIS or CryptoCard auth (SSH-1)
- [✓] Attempt "keyboard-interactive" auth (SSH-2)

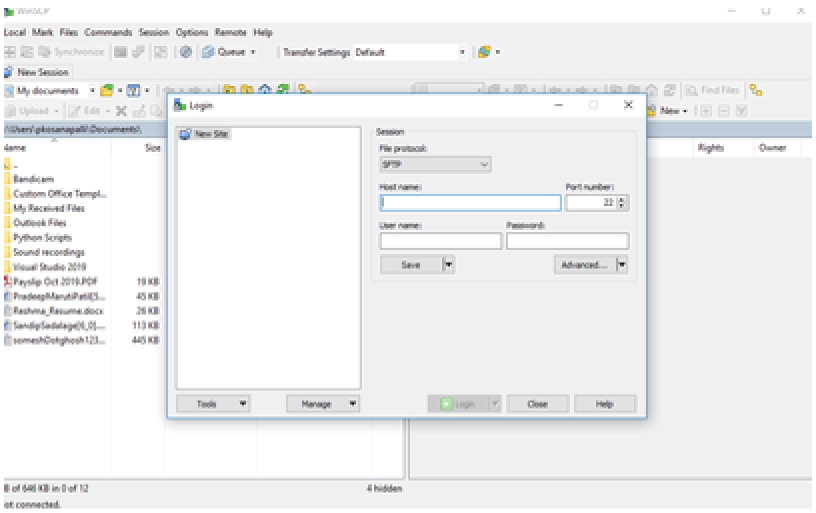
Authentication parameters

- [ ] Allow agent forwarding
- [ ] Allow attempted changes of username in SSH-2

Private key file for authentication: C:\Users\pkosanapalli\Desktop\docume

6. Install winscp from https://winscp.net/download/WinSCP-5.15.5-Setup.exe.

7. winscp is similar to putty, you need to give login credentials. the advantage is you can able to see the file explorer and download to locally.



Login

Session

File protocol: SFTP

Host name:

Port number: 22

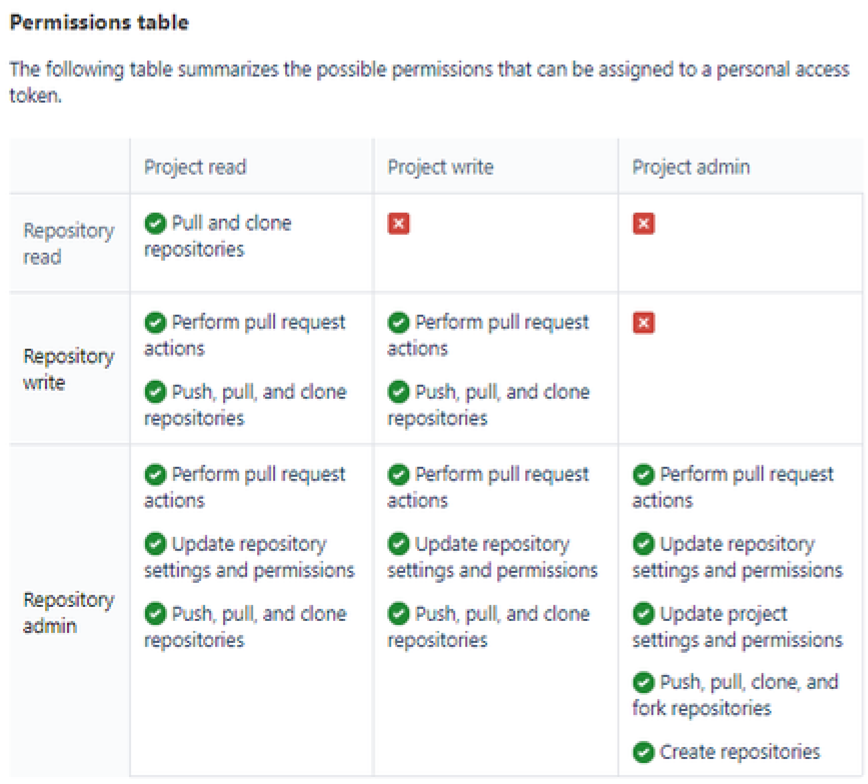
User name:

Password:

8.login to instance for rectification of Linux servers.

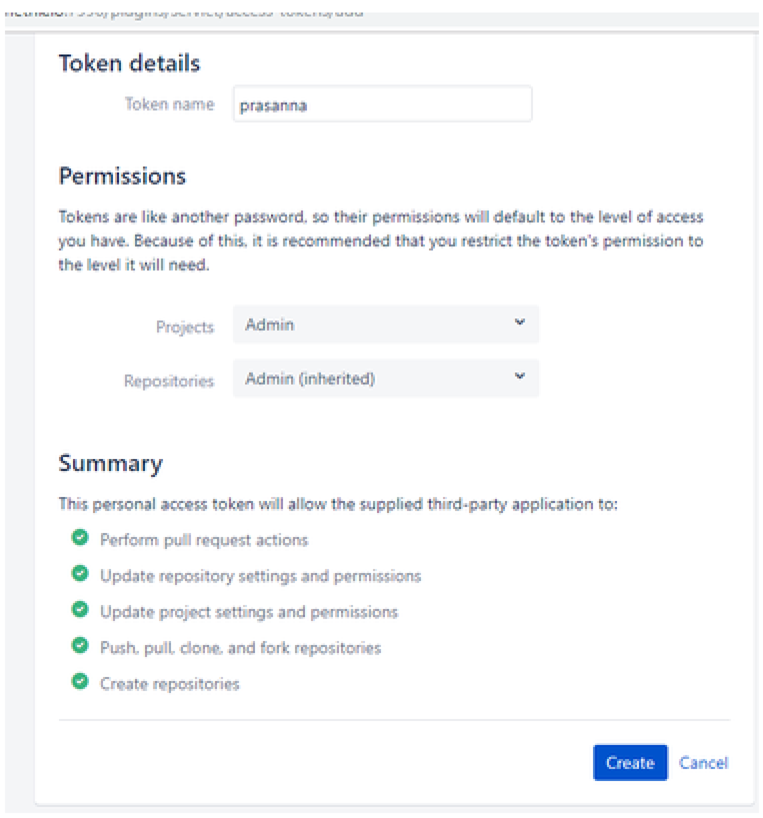
Authentication Bit-bucket &GIT:-

1. Go to Bitbucket server manage account   persnoal access token create token add permissions.


Permissions table

The following table summarizes the possible permissions that can be assigned to a personal access token.

	Project read	Project write	Project admin
Repository read	✓ Pull and clone repositories	✗	✗
Repository write	✓ Perform pull request actions ✓ Push, pull, and clone repositories	✓ Perform pull request actions ✓ Push, pull, and clone repositories	✗
Repository admin	✓ Perform pull request actions ✓ Update repository settings and permissions ✓ Push, pull, and clone repositories	✓ Perform pull request actions ✓ Update repository settings and permissions ✓ Push, pull, and clone repositories	✓ Perform pull request actions ✓ Update repository settings and permissions ✓ Update project settings and permissions ✓ Push, pull, clone, and fork repositories ✓ Create repositories


Token details

Token name: prasanna

Permissions

Tokens are like another password, so their permissions will default to the level of access you have. Because of this, it is recommended that you restrict the token's permission to the level it will need.

Projects: Admin

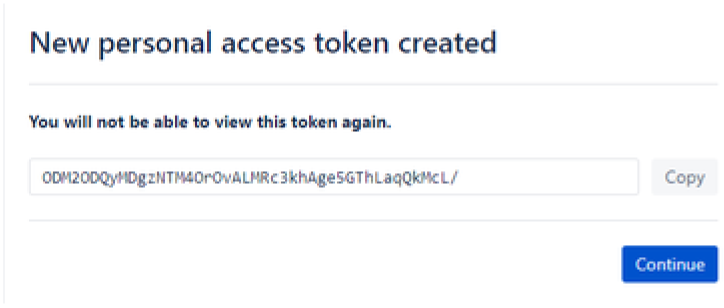
Repositories: Admin (inherited)

Summary

This personal access token will allow the supplied third-party application to:

- ✓ Perform pull request actions
- ✓ Update repository settings and permissions
- ✓ Update project settings and permissions
- ✓ Push, pull, clone, and fork repositories
- ✓ Create repositories

2. copy the personal token key and paste where your requires git clone & push,pull etc.

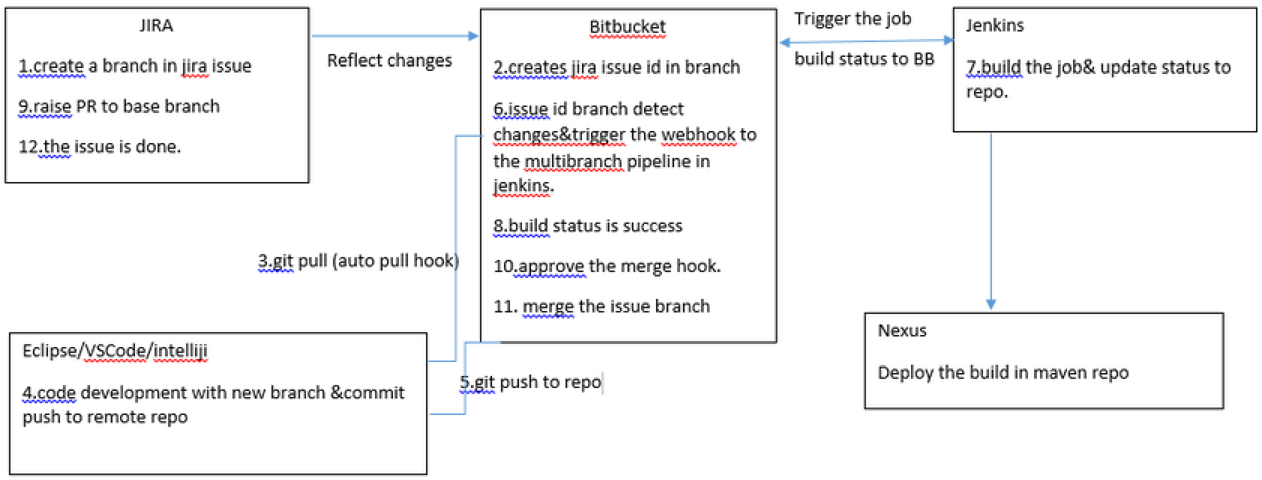


New personal access token created

You will not be able to view this token again.

```
ODM2ODQyMDgzNTM4OrOvALMRc3khAge5GThLaqQkMcL/
```

End to end workflow of developer:


JIRA
1.create a branch in jira issue
9.raise PR to base branch
12.the issue is done.
Reflect changes
Bitbucket
2.creates jira issue id in branch
6.issue id branch detect changes&trigger the webhook to the multibranch pipeline in jenkins.
8.build status is success
10.approve the merge hook.
11. merge the issue branch
Trigger the job
build status to BB
Jenkins
7.build the job& update status to repo.
3.git pull (auto pull hook)
Eclipse/VSCode/intellij
4.code development with new branch &commit push to remote repo
5.git push to repo
Nexus
Deploy the build in maven repo

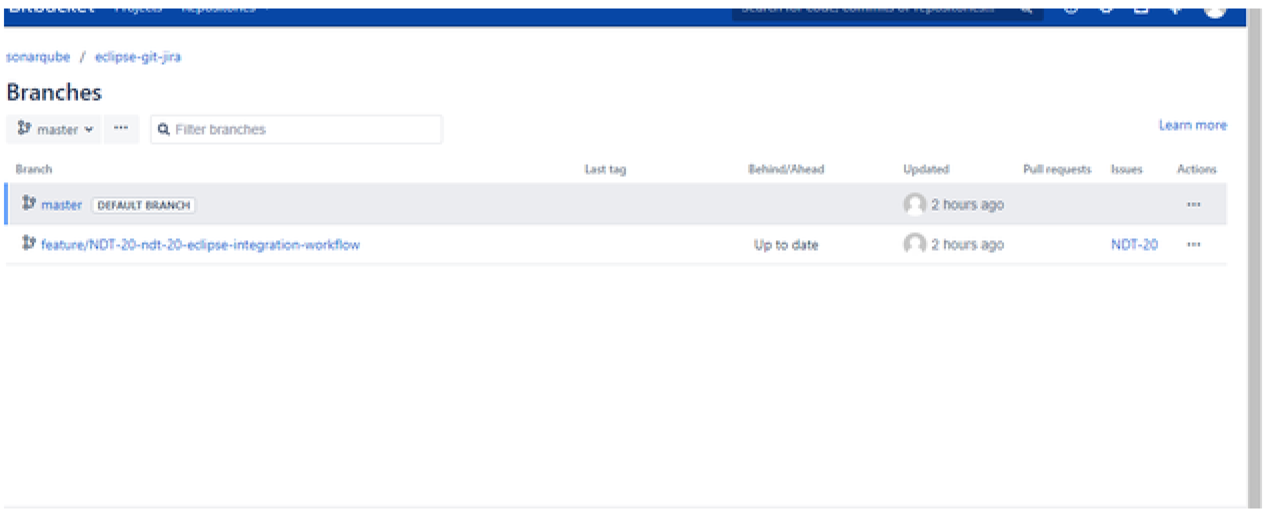
Steps:

1. Create a issue in the Jira and create the branch from the corresponding repository. note: integrate the Jira and Bitbucket for repo&issues reflection - Bitbucket- Jira integration.

2. The feature branch which you created is reflected in the Bitbucket branches with issue id.

3. Now open the eclipse and import the repository and configure the eclipse as i shown in above setup of Eclipse.

4. You can able to see the feature branch.


sonarqube / eclipse-git-jira

Branches

Branch	Last tag	Behind/Ahead	Updated	Pull requests	Issues	Actions
master DEFAULT BRANCH			2 hours ago			···
feature/NDT-20-ndt-20-eclipse-integration-workflow		Up to date	2 hours ago		NDT-20	···

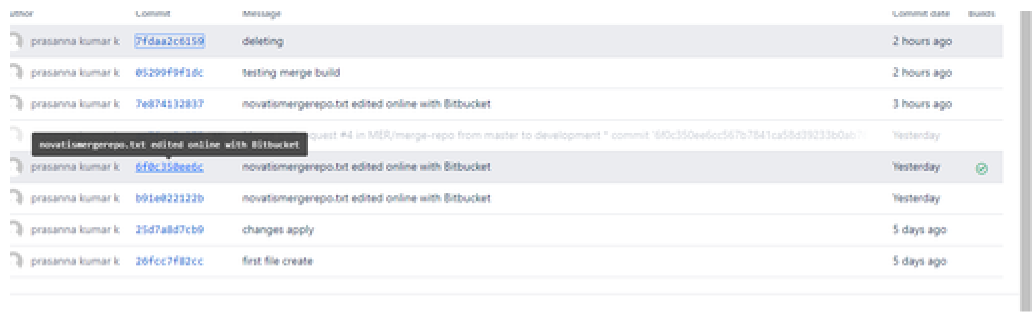
5.Develop the code changes in Eclipse and commit and push the changes to remote feature branch.

6. Now a multi branch pipeline job is triggered in the jenkins with the help of webhooks in the bitbucket.

reference-1 : Bitbucket and jenkins integration - Jenkins integration- job status & merge restrict.

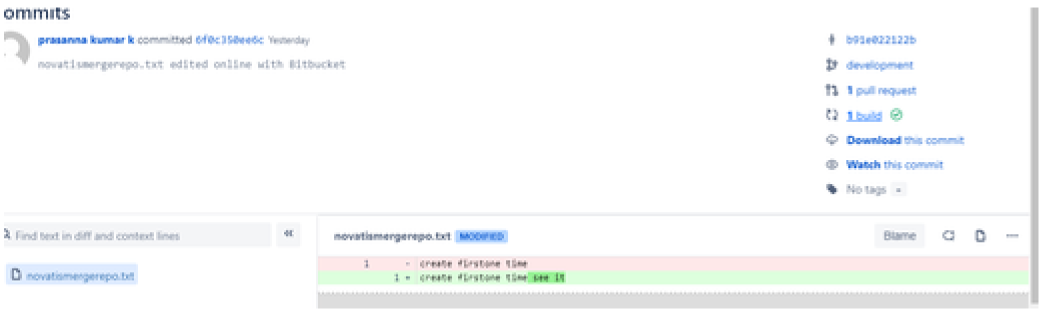
reference-2 :multi branch pipeline job in jenkins - How to - Integrate multi branch pipeline in jenkins

7.Now you can see the build status in the Bitbucket status bar.



Author	Commit	Message	Commit date	Builds
prasanna kumar k	7fdaa2c6159	deleting	2 hours ago	
prasanna kumar k	05299f9f1dc	testing merge build	2 hours ago	
prasanna kumar k	7e874132837	novatismergerepo.txt edited online with Bitbucket	3 hours ago	
		[illegible] quest #4 in MER/merge-repo from master to development * commit '6f0c350ee6cc567b7841ca58d39233b0ab'	Yesterday	
prasanna kumar k	6f0c350ee6c	novatismergerepo.txt edited online with Bitbucket	Yesterday	✓
prasanna kumar k	b91e022122b	novatismergerepo.txt edited online with Bitbucket	Yesterday	
prasanna kumar k	25d7a8d7cb9	changes apply	5 days ago	
prasanna kumar k	26fcc7f82cc	first file create	5 days ago	

novatismergerepo.txt edited online with Bitbucket


ommits

prasanna kumar k committed 6f0c350ee6c Yesterday

novatismergerepo.txt edited online with Bitbucket

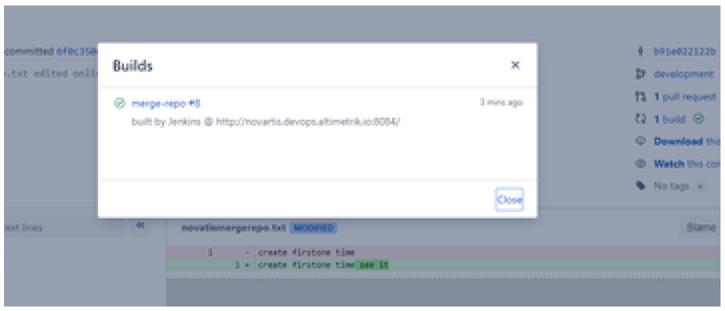
b91e022122b
development
1 pull request
1 build ✓
Download this commit
Watch this commit
No tags

Find text in diff and context lines

novatismergerepo.txt

novatismergerepo.txt MODIFIED

```
1     - create firstone time
    1 + create firstone time see it
```

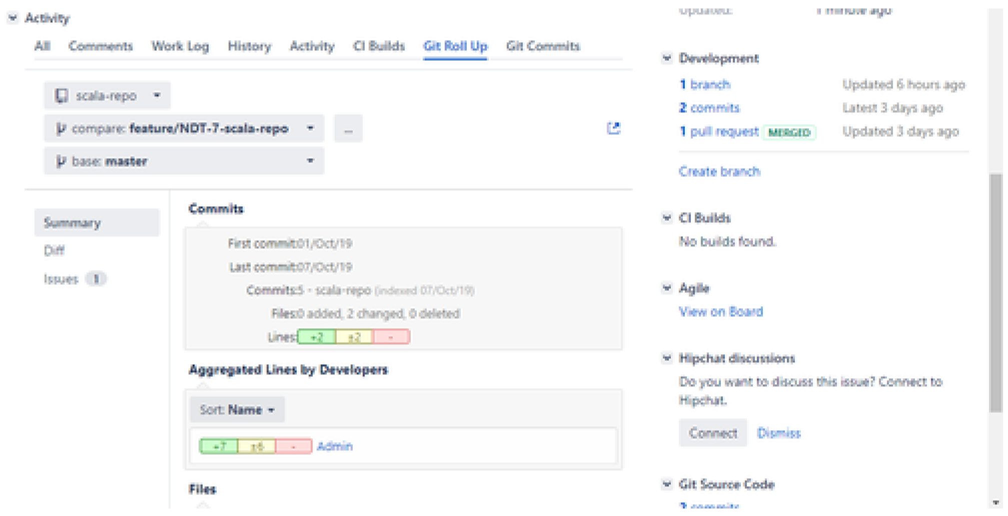

Builds

merge-repo #8 3 mins ago

built by Jenkins @ http://novartis.devops.altimetrik.io:8084/

Close

8. Now you can see the build status in the Jira also and you can see the git roll up and commits, feature branches related to the issue.



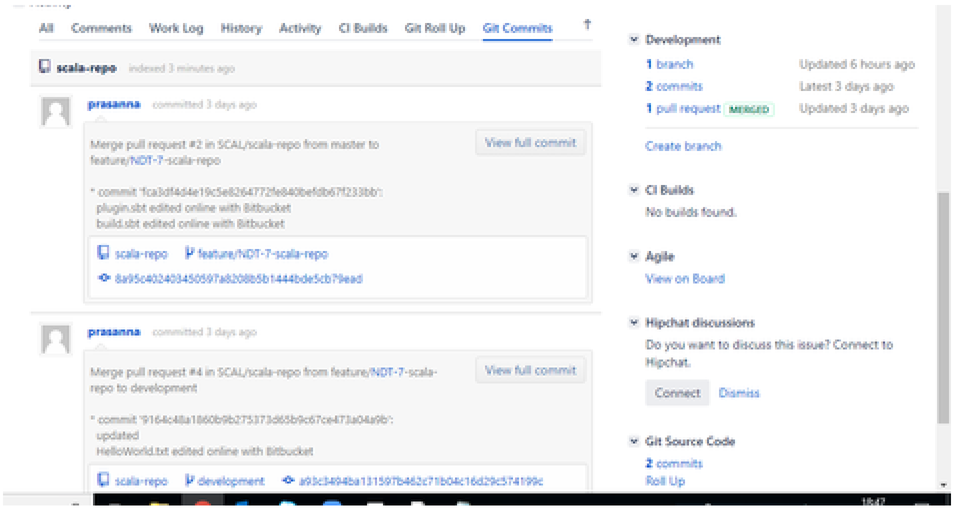
First commit:01/Oct/19
Last commit:07/Oct/19
Commits:5 - scala-repo (indexed 07/Oct/19)
Files:0 added, 2 changed, 0 deleted
Lines: +2 +2 -

+7 +6 - Admin

1 branch Updated 6 hours ago
2 commits Latest 3 days ago
1 pull request MERGED Updated 3 days ago
Create branch

No builds found.

Do you want to discuss this issue? Connect to Hipchat.



scala-repo indexed 3 minutes ago

prasanna committed 3 days ago

Merge pull request #2 in SCAL/scala-repo from master to feature/NDT-7-scala-repo

* commit 'fca3df4d4e19c5e8264772fe840befdb67f233bb':
plugin.sbt edited online with Bitbucket
build.sbt edited online with Bitbucket

scala-repo feature/NDT-7-scala-repo
8a95c402403450597a8208b5b1444bde5cb79ead

prasanna committed 3 days ago

Merge pull request #4 in SCAL/scala-repo from feature/NDT-7-scala-repo to development

* commit '9164c48a1860b9b275373d65b9c67ce473a04a9b':
updated
HelloWorld.txt edited online with Bitbucket

scala-repo development a93c3494ba131597b462c71b04c16d29c574199c

1 branch Updated 6 hours ago
2 commits Latest 3 days ago
1 pull request MERGED Updated 3 days ago
Create branch

No builds found.

Do you want to discuss this issue? Connect to Hipchat.

9. Once code is freezes, now you can raise the PR in the Jira issue and same is reflected in the bitbucket.

10. Once the lead is approved for the merge. You can able to merge.

reference 1: How to- repository configuration for default branch& merge hook-bit bucket. for merge hooks

reference 2: Merge hook-branch (reviewer & build successful) for builds status.

11. In the Jira , you need to set the workflow as auto transition of the ticket, once the PR is merged.

12.After merging, the issue is transfer to review stage.

13. Jenkins while deploy the build application in nexus.

14.Any changes you need you can comment in the issue section of Jira ticket in bit bucket.

15. Also you can able to raise a issue from Bitbucket itself and same will be reflects in the Jira board.

Whats next:

Trigger the build in Gitlab(ee) when changes detect in bitbucket:

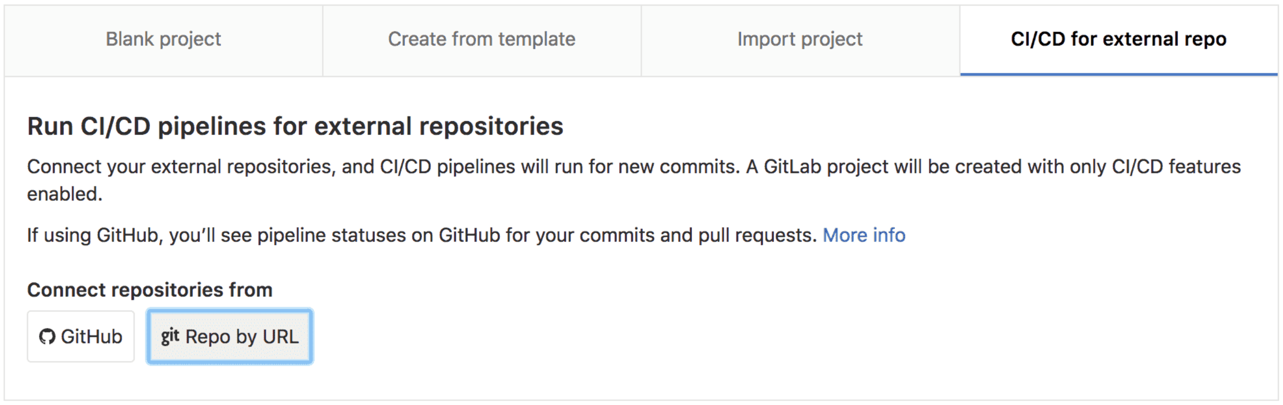
Using GitLab CI/CD with a Bitbucket Cloud repository

As i show in the workflow use gitlab instead of jenkins. GitLab CI/CD can be used with Bitbucket Cloud by:

1. Creating a CI/CD project.
2. Connecting your Git repository via URL.

To use GitLab CI/CD with a Bitbucket Cloud repository:

1. In GitLab create a **CI/CD for external repo**, select **Repo by URL** and create the project.



Run CI/CD pipelines for external repositories

Connect your external repositories, and CI/CD pipelines will run for new commits. A GitLab project will be created with only CI/CD features enabled.

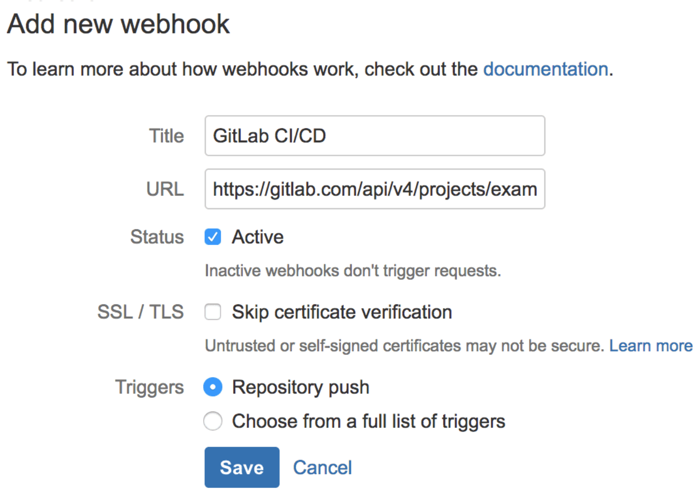
If using GitHub, you'll see pipeline statuses on GitHub for your commits and pull requests. More info

GitLab will import the repository and enable Pull Mirroring.

2. In GitLab create a Personal Access Token with `api` scope. This will be used to authenticate requests from the web hook that will be created in Bitbucket to notify GitLab of new commits.
3. In Bitbucket, from **Settings > Webhooks**, create a new web hook to notify GitLab of new commits.

   The web hook URL should be set to the GitLab API to trigger pull mirroring, using the Personal Access Token we just generated for authentication.
   `https://gitlab.com/api/v4/projects/<NAMESPACE>%2F<PROJECT>/mirror/pull?private_token=<PERSONAL_ACCESS_TOKEN>`

   The web hook Trigger should be set to 'Repository Push'.

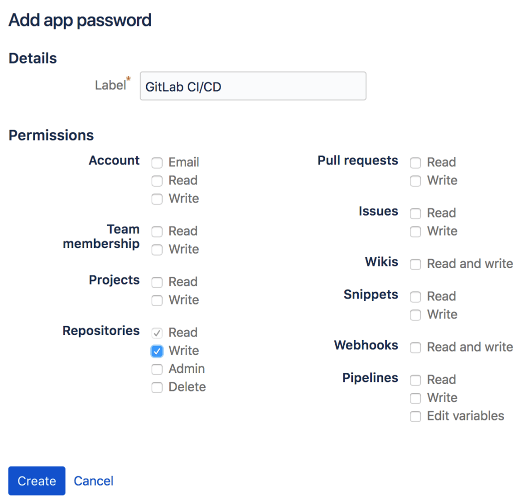


Add new webhook

To learn more about how webhooks work, check out the documentation.

Title	GitLab CI/CD
URL	https://gitlab.com/api/v4/projects/exam
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

   After saving, test the web hook by pushing a change to your Bitbucket repository.
4. In Bitbucket, create an **App Password** from **Bitbucket Settings > App Passwords** to authenticate the build status script setting commit build statuses in Bitbucket. Repository write permissions are required.



Add app password

Details

Label* GitLab CI/CD

Permissions

Account: [ ] Email [ ] Read [ ] Write

Team membership: [ ] Read [ ] Write

Projects: [ ] Read [ ] Write

Repositories: [✓] Read [✓] Write [ ] Admin [ ] Delete

Pull requests: [ ] Read [ ] Write

Issues: [ ] Read [ ] Write

Wikis: [ ] Read and write

Snippets: [ ] Read [ ] Write

Webhooks: [ ] Read and write

Pipelines: [ ] Read [ ] Write [ ] Edit variables

5. In GitLab, from **Settings > CI/CD > Environment variables**, add variables to allow communication with Bitbucket via the Bitbucket API:

   `BITBUCKET_ACCESS_TOKEN`: the Bitbucket app password created above.

   `BITBUCKET_USERNAME`: the username of the Bitbucket account.

   `BITBUCKET_NAMESPACE`: set this if your GitLab and Bitbucket namespaces differ.

   `BITBUCKET_REPOSITORY`: set this if your GitLab and Bitbucket project names differ.
6. In Bitbucket, add a script to push the pipeline status to Bitbucket. Note: changes made in GitLab will be overwritten by any changes made upstream in Bitbucket.

   Create a file`build_status`and insert the script below and run`chmod +x build_status`in your terminal to make the script executable.

   ```
   Still in Bitbucket, create a.gitlab-ci.ymlfile to use the script to push pipeline success and
   failures to Bitbucket.
   ```

7.
   ```
   stages:
     - test
     - ci_status

   unit-tests:
     script:
       - echo "Success. Add your tests!"

   success:
     stage: ci_status
     before_script:
       - ""
     after_script:
       - ""
     script:
       - BUILD_STATUS=passed BUILD_KEY=push ./build_status
     when: on_success

   failure:
     stage: ci_status
     before_script:
       - ""
     after_script:
       - ""
     script:
       - BUILD_STATUS=failed BUILD_KEY=push ./build_status
     when: on_failure
   ```

GitLab is now configured to mirror changes from Bitbucket, run CI/CD pipelines configured in `.gitlab-ci.yml` and push the status to Bitbucket thats all done !!