**Tools Setup**

* Git(2.10.1)

Git can installed Ubuntu linux distribution.

apt-get install git

* Ansible(2.1.2.0)

Installing ansible on Ubuntu linux distribution is easy with a single command.

sudo yum install ansible

To work with Ansible we need to create *ansible.hosts* file in the home directory. Add hosts in the hosts file.

-- localhost

Create ssh keys for you system if not created already.

ssh-keygen -t rsa

Append the public keys to the *authorized\_keys.*

cat id\_rsa.pub >> authorized\_keys

Ping the hosts using Ansible to test the Setup.

ansible -i ~/ansible.hosts -m ping all

Configuration files:

config file = /etc/ansible/ansible.cfg

configured module search path = Default w/o overrides

* Jenkins(2.24)

Jenkins can be installed as a service or as a webapp. To use as web app download the *jenkins.war* file and deploy it any application server like tomcat. To install it as a service.

apt-get install jenkins

Jenkins can also be installed using Ansible playbook. Clone the Ansible jenkins role in the ‘/etc/ansible/roles/’ directory.

git clone <https://github.com/ICTO/ansible-jenkins.git>

Create a host file in home directory ( ~/ansible.hosts).

Mention the configuration of the tools as defined below.

[gitlab]

localhost http\_port=8081

[jenkins]

localhost http\_port=8083

[jira]

localhost http\_port=8085

Create ssh keys with blank passphrase.

ssh-keygen -t rsa

Add key to authorized keys.

cat id\_rsa.pub >> authorized\_keys

Test the setup with the command.

ansible -i ~/ansible.hosts -m ping all

Create a jenkins playbook file in the home directory (jenkins\_playbook.yml).

- hosts: jenkins

vars:

jenkins\_hostname: localhost

roles:

- ansible-jenkins

Install Jenkins by the command.

ansible-playbook -i ~/ansible.host ~/jenkins\_playbook.yml

* Gitlab(8.13.0)

Gitlab can installed as a service by using a procedure mentioned on the website specific to the distribution.

1. Add the GitLab package server and install the package

curl -sS https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.deb.sh | sudo bash

sudo apt-get install gitlab-ce

2. Configure and start GitLab

sudo gitlab-ctl reconfigure

3. Browse to the hostname and login. On your first visit, you'll be redirected to a password reset screen to provide the password for the initial administrator account. Enter your desired password and you'll be redirected back to the login screen.

While downgrading from Gitlab EE to Gitlab CE some issues may be encountered.

ActionView::Template::Error (The single-table inheritance mechanism failed to locate the subclass: 'JenkinsService'. This  
error is raised because the column 'type' is reserved for storing the class in case of inheritance. Please rename this  
column if you didn't intend it to be used for storing the inheritance class or overwrite Service.inheritance\_column to  
use another column for that information.)

sudo gitlab-rails runner "Service.where(type: ['JenkinsService', 'JenkinsDeprecatedService']).delete\_all"

The configuration file is located at:

/etc/gitlab/gitlab.rb

* Maven(3.3.9)

Maven can downloaded from the apache website. The zipped archive can be downloaded and extracted specific to the operating system. Add the bin directory to the system PATH.

* JIRA(7.2.2)

Jira can be installed using the executable available from Atlassian. Download the bundle from the jira download page.

Extract it in a directory.

tar -xzf atlassian-jira-software-X.X.X.tar.gz -C <installation-directory>

Edit <installation-directory>\atlassian-jira\WEB-INF\classes\jira-application.properties file in any text editor.After jira.home add the absolute path to your home directory (not a symlink), for example:

jira.home=/var/jirasoftware-home

By default JIRA listens on port 8080. If you have another application running on your server that uses the same ports, you'll need to tell JIRA to use a different port. Edit <installation-directory>\conf\server.xml

<Server port="5005" shutdown="SHUTDOWN">

...

<Service name="Catalina">

<Connector port="5050"

maxThreads="150"

minSpareThreads="25"

connectionTimeout="20000"

enableLookups="false"

maxHttpHeaderSize="8192"

protocol="HTTP/1.1"

useBodyEncodingForURI="true"

redirectPort="8443"

acceptCount="100"

disableUploadTimeout="true"/>

Run <installation-directory>/bin/start-jira.sh to start the setup process.

./start-jira.sh

Go to <http://localhost:8080/> to launch JIRA in your browser (change the port if you've updated the Connector port).