Exercise – Jenkins and Git part 2

Objective

The objective of this exercise is to automate backing up the Jenkins machine and set up security for your Jenkins server

Overview

Part 1: Backing up Jenkins Config

To use git to back up the Jenkins configuration files we need to setup our repository and git account to use SSH keys.

On your Jenkins server command line:

1. Open a bash instance under the Jenkins user:

```
$ sudo -u jenkins bash
```

2. We want to create an SSH key pair. This includes a public and private key. To do this type

```
bash-4.1$ ssh-keygen
```

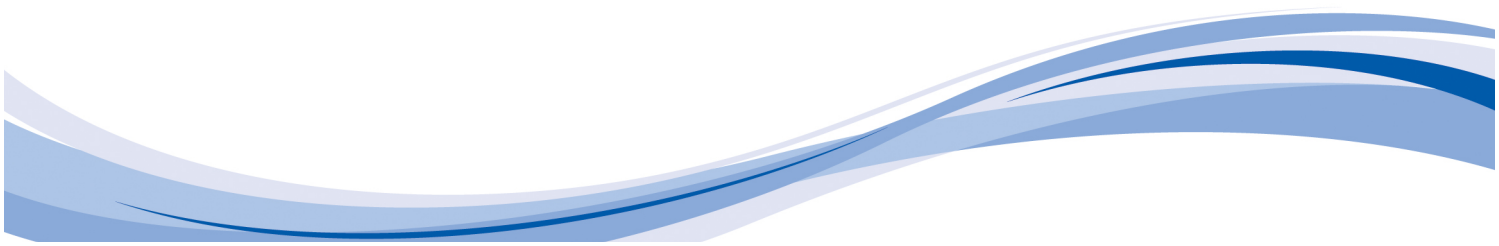
   Accept all the defaults. This will generate two files in the Jenkins home directory: id_rsa id_rsa.pub. The output should give you a message along these lines:

```
Your identification has been saved in /var/lib/jenkins/.ssh/id_rsa.
Your public key has been saved in /var/lib/jenkins/.ssh/id_rsa.pub.
```

   The .pub file is the public key; this is the one you can freely give to other people and servers. The other is the private key; this is the one that only you should use. This allows servers to identify who you are.

3. Run the following command to display a copy of the key that we can pass to your git server.

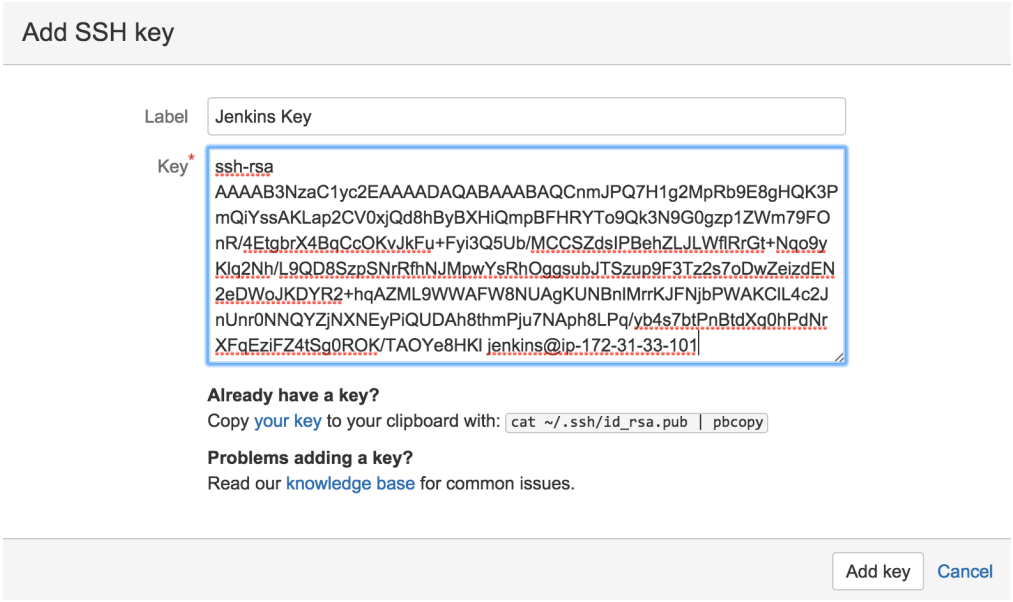
```
$ cat ~/.ssh/id_rsa.pub
```



In Bitbucket

Our next step is to setup Bitbucket to accept the SSH key as an authentication method.

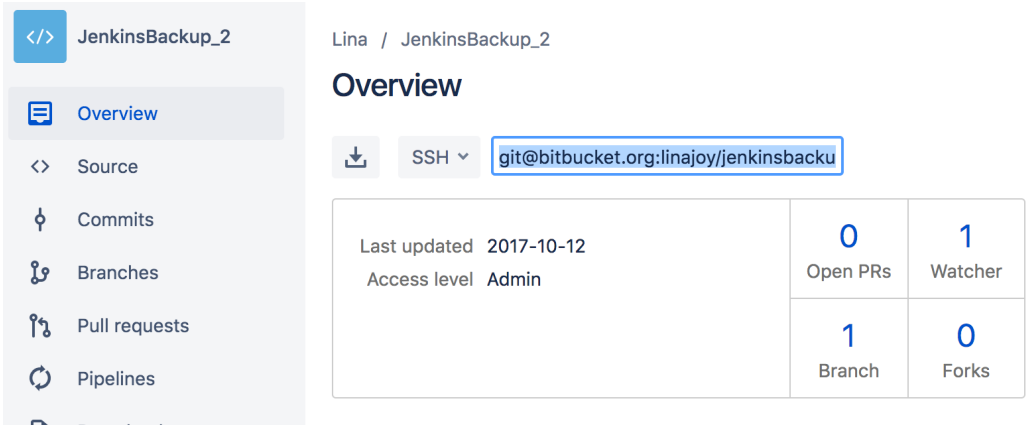
1. Go to Bitbucket and login. Click on the profile icon and go to “**Bitbucket settings**”
2. Click “**SSH Keys**” on the left hand side of the screen and then click "**Add key**"
3. Give the key any name you like and copy in the public key.



4. Finally, we want to set up Jenkins to use a git repository.

   Create a new repository called **Jenkins-Backup**. You can leave it private if you like. The key will let Jenkins login to commit changes.

5. Click on **Overview** and select "**SSH**". Copy the link given starting from "git@..."



6. We first want to check if we can connect to the repository via the SSH link from the command line. In your terminal go back to the Jenkins bash shell and type:

```
bash-4.1$ cd ~

bash-4.1$ git ls-remote git@bitbucket.org:UserName/jenkins-\
                                                    backup.git
```

7. Replace the git@bitbucket with your link. This will give you a RSA key fingerprint for the server. Type yes to continue connecting. This should list the branches on the host.

   If it works, we can add this to Jenkins.

Jenkins Dashboard

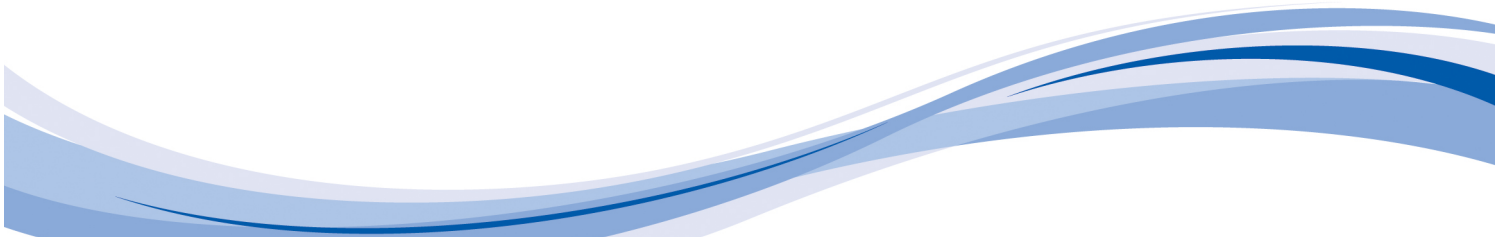
1. Go to your Jenkins Server. Click on **Manage Jenkins**.
2. Install the **SCM Sync Configuration** plugin via the plugins menu
3. Then go back to the **Manage Jenkins** menu and click on **Configure System**. We should have a new area for **SCM Sync**. In this give the *repository URL* as the SSH git url copied from your git server. This will allow Jenkins to commit the files correctly.
4. Click **save** to go back to the main dashboard.
5. To push the setup to your git server, go into **Manage Jenkins**, then **Configure System** settings and change something (the basic system message is usually a good one).

Note: if you are having issues with the sync not working you can find the reason why in **Manage Jenkins** -> **Logs**.

To fix problems with existing repositories remove the current directory and then restart Jenkins before making a new change.

```
$ sudo rm -rf /var/lib/jenkins/scm-sync-\
                            configuration/checkoutConfiguration

$ sudo service jenkins restart
```



Part 2: Security

It is not ideal to have anyone able to setup jobs on Jenkins and trigger new builds. Jenkins sets up a basic admin user by default. We can access the security settings via the **Manage Jenkins** -> **Configure Global Security** links.

Investigate the options in here (but be careful, you can lock yourself out quite easily):

- The original setting for Security was anyone can do anything
- You can allow users to sign up from the Security realm options or add users manually through the **Manage Jenkins** options
- Matrix based security allows you to specify exactly what each person can do and with what on Jenkins - it is very easy to lock yourself out of Jenkins using this!

  Note: Make sure the first user you create has access to everything!