Exercise – A GitLab server of our very own

Objective

This exercise looks at setting up our own GitLab server and allowing other people access to it. We will set up the server on AWS, which has its own problems. The GitLab server can be used in the case study if you want to investigate how to create a fully home controlled system.

Part 1: Setup

We will need to create a new server in aws to run GitLab for us. Create a new EC2 instance with the following properties:

- AMI – Centos 6 (ami-03a2ad65)
- Size – t2.medium
- Storage – 20gig
- Tag – name: GitLab
- Security group: Open port 80/tcp for access from anywhere

Connect to your server using SSH as a **CENTOS** user. If you need help with any of these steps, please see the quick guides for creating a new instance, generating the key and connecting via putty or the command line.

Part 2: Installing GitLab

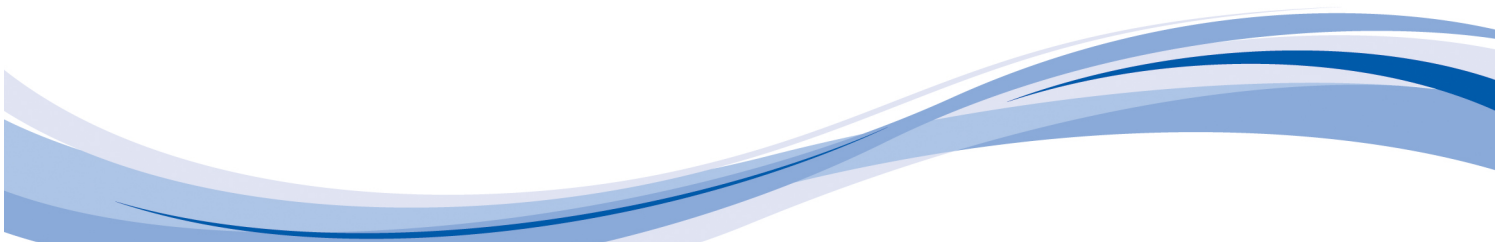
All how to install GitLab is on official webpage: https://about.gitlab.com/installation/. To install GitLab on our machine we need to do the following:

1. Run the first two lines of the script. It will open HTTP and SSH access in the system firewall.

```
$ sudo yum install -y curl policycoreutils-python \
                                        openssh-server cronie

$ sudo lokkit -s http -s ssh
```

2. Next commands will install the services required for GitLab, start postfix (a mail server) tell postfix to automatically start.

```
$ sudo yum install postfix -y

$ sudo service postfix start

$ sudo chkconfig postfix on
```



3. Next we want to grab a copy of the installation script with curl and pipe it through bash to run.

```
  $ curl
https://packages.gitlab.com/install/repositories/gitlab/gitlab-
ee/script.rpm.sh | sudo bash
```

4. Once the script has run we install gitlab-ee through yum. But first we need to change `http://gitlab.example.com` to the URL at which we want to access the GitLab instance. It will automatically configure and start GitLab at that URL.

```
  $ sudo EXTERNAL_URL="http://gitlab.example.com" yum install
gitlab-ee -y
```

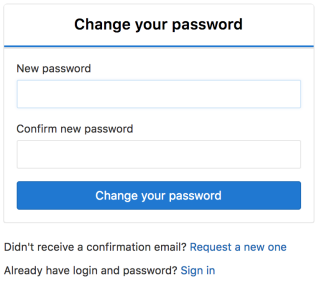
5. Configure gitlab

```
  $ sudo gitlab-ctl reconfigure
```

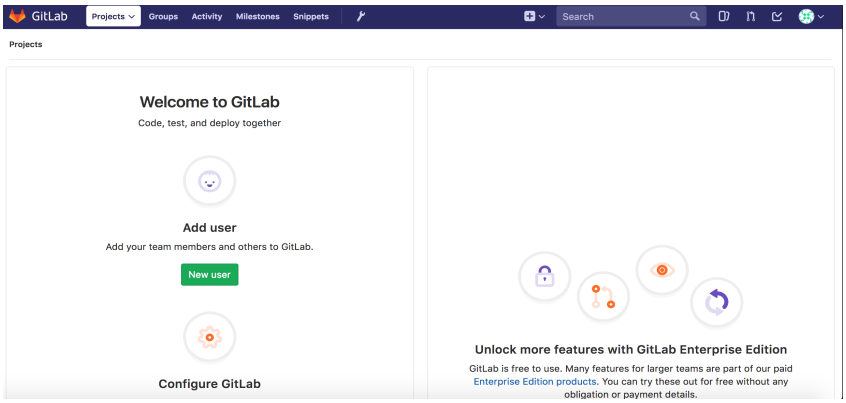
GitLab uses chef-solo to configure and set itself up. When everything has finished have a look at your new GitLab server at *http:///[yourip]*

Note: If the reconfigure command fails, run it a second time. There is a bug in the GitLab script that assumes that you will have the bridge module enabled in your OS kernel. The second reconfigure fixes this problem.

1. The first step is to change password immediately. Your username will be root.

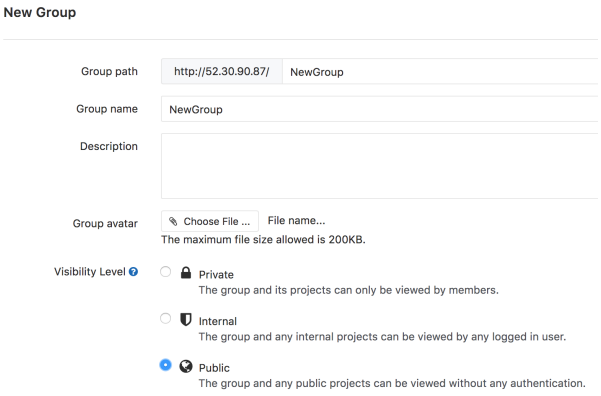


2. Once you have done this you will be rewarded with the main dashboard:

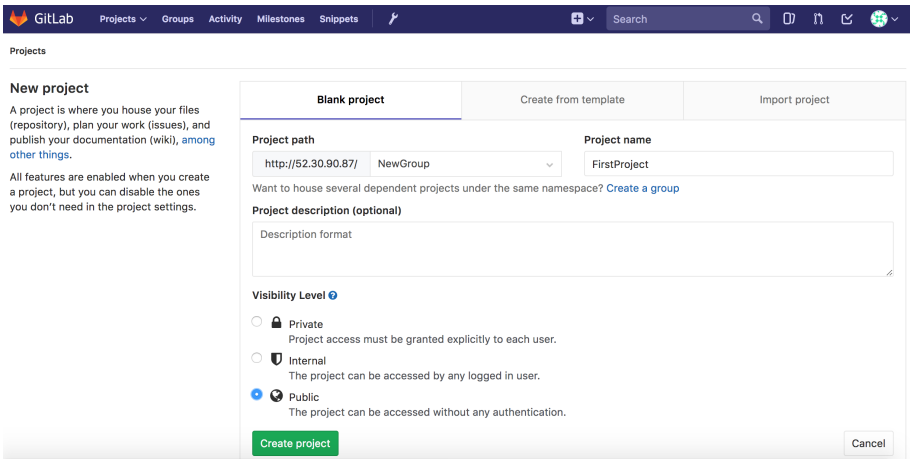


Part 3: Create a group and project

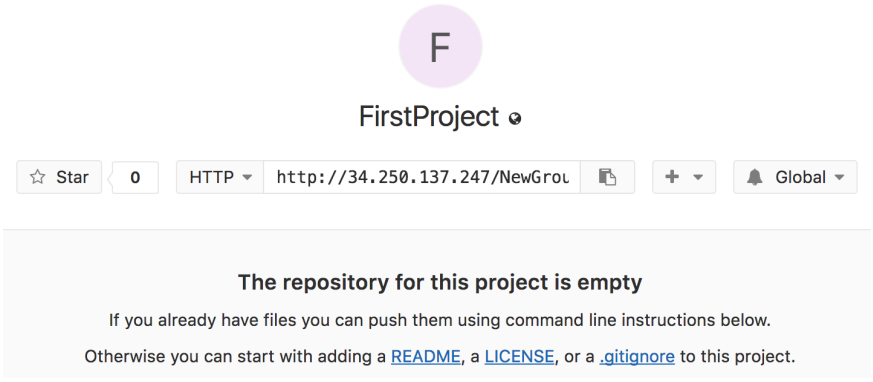
3. Now we have a server we want to create a new group. Click on the **New Group** button. Give your group a name and make it public. Then click the **Create group** button.



4. Now we can create a new project. Click on the **New Project** button. Give your project a name, a description and make it public for this time. Any name will do.



5. Then click the **Create project** button.



6. Change from SSH access to using HTTP access for your project and then copy the git repository link.
7. We should be able to clone this repository to any machine. Try cloning the your repository to your windows machine.

```
$ git clone [repo address]
```

8. Create a file in your new repository, commit and push it. Your username and password will be “root” and whatever password you set when you first logged in. See if your files appear on your server.

Part 3: Automate the installation process

The main question that we should be asking whenever we install a new system is:

> “How can I automate that?”

AWS allows you to pass it a script that is run on start up. For linux this will be a bash script. It should have the extension of *.sh* and start with the line

```
#!/bin/bash
```

Write a script which goes through each of the steps required to install and setup a GitLab server. Create a new machine via aws and pass it your script and see if you can get GitLab to be installed without any help.

Part 4: Document

Write down the steps you took over this exercise in your documentation git repository. Add your script from part 3. Remember to write down any problems you had and how you solved them, so that you can refer back to this later.

If you have time…

- Have a look at creating a backup for your GitLab server
- Create a brand new server and restore your setup from this backup
- Look at how to automate the backup with cron tab