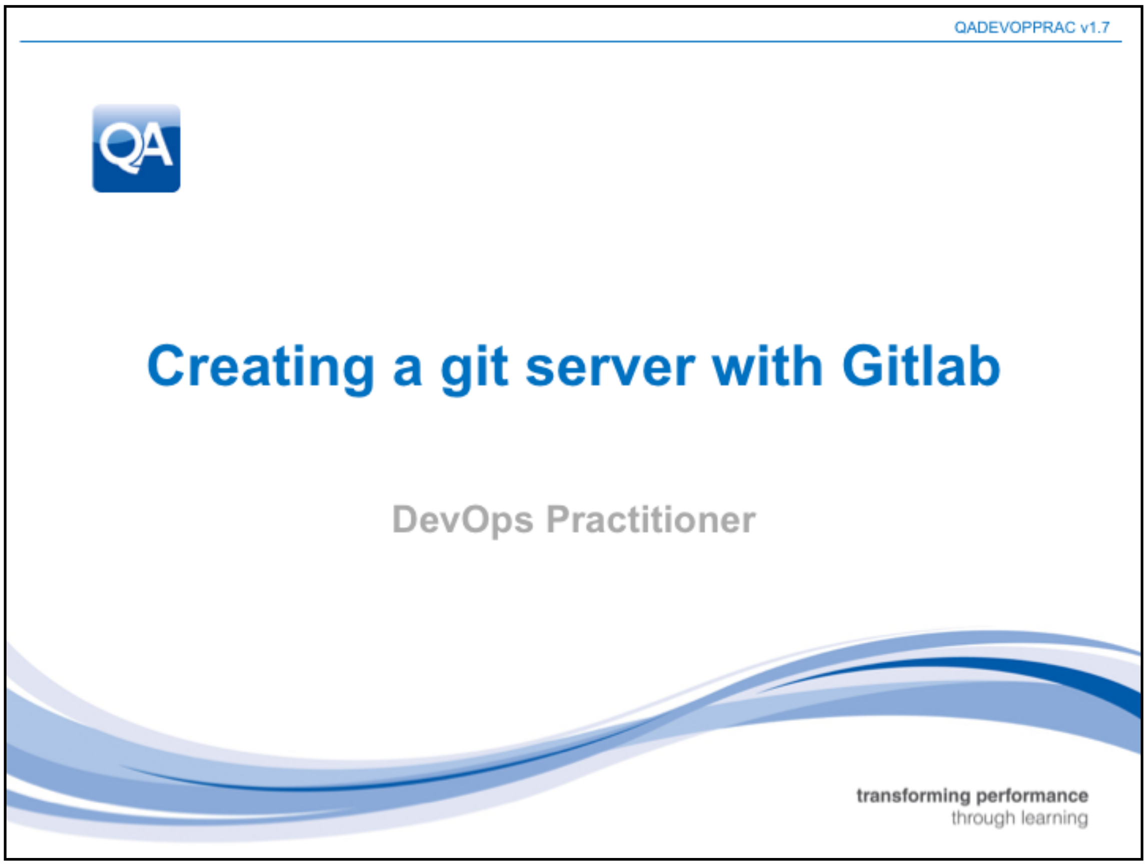
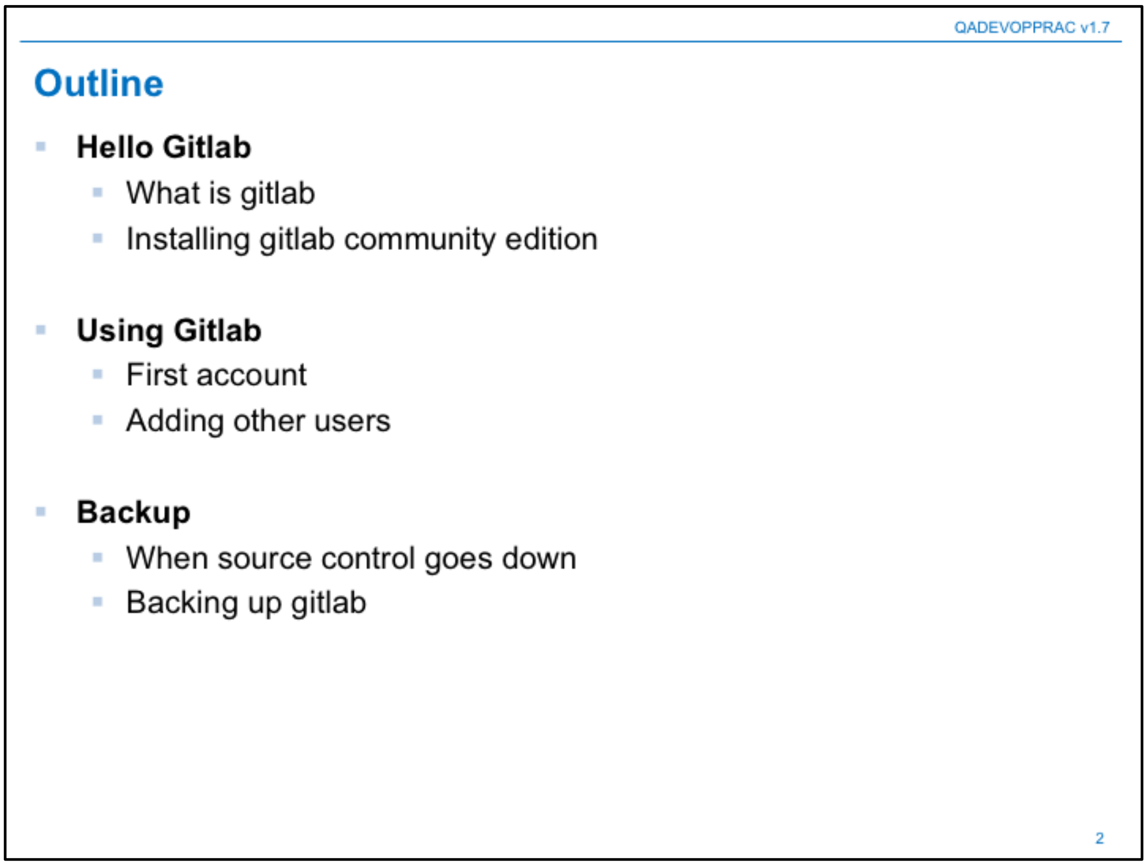


QADEVOPPRAC v1.7

Creating a git server with Gitlab

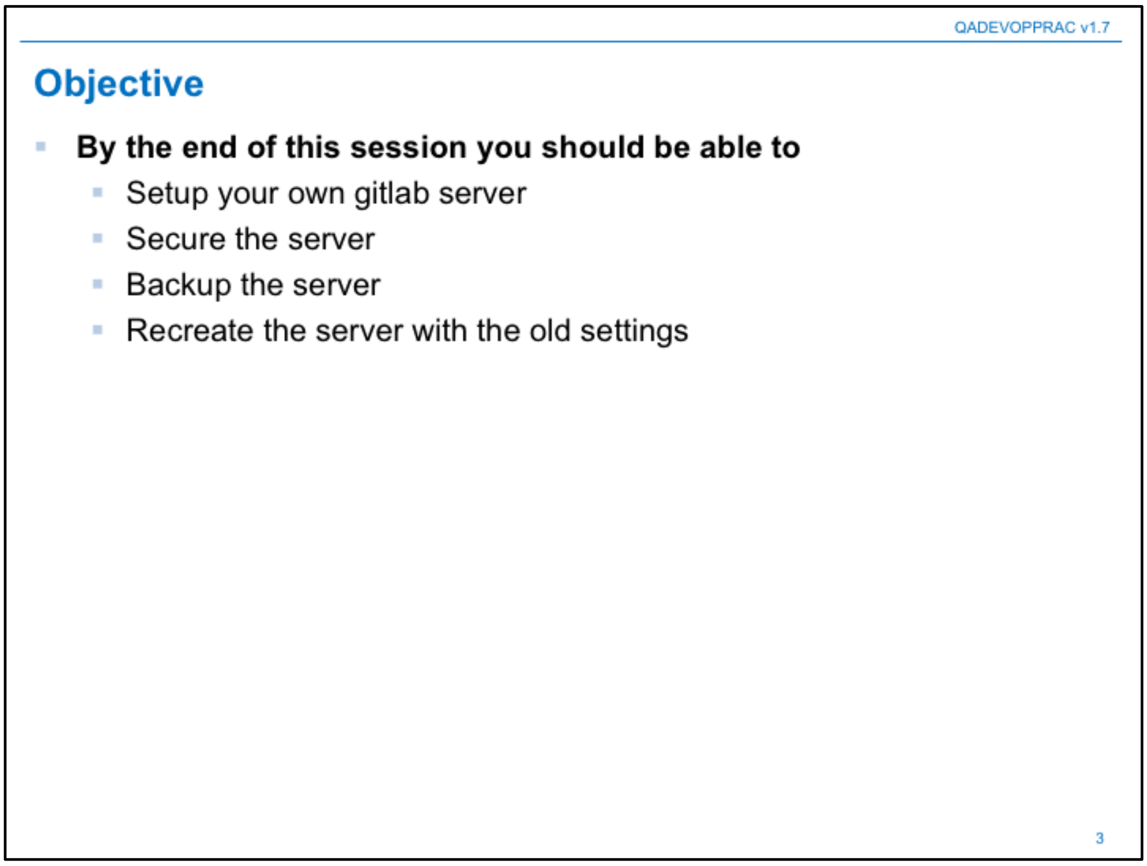
DevOps Practitioner

transforming performance
through learning



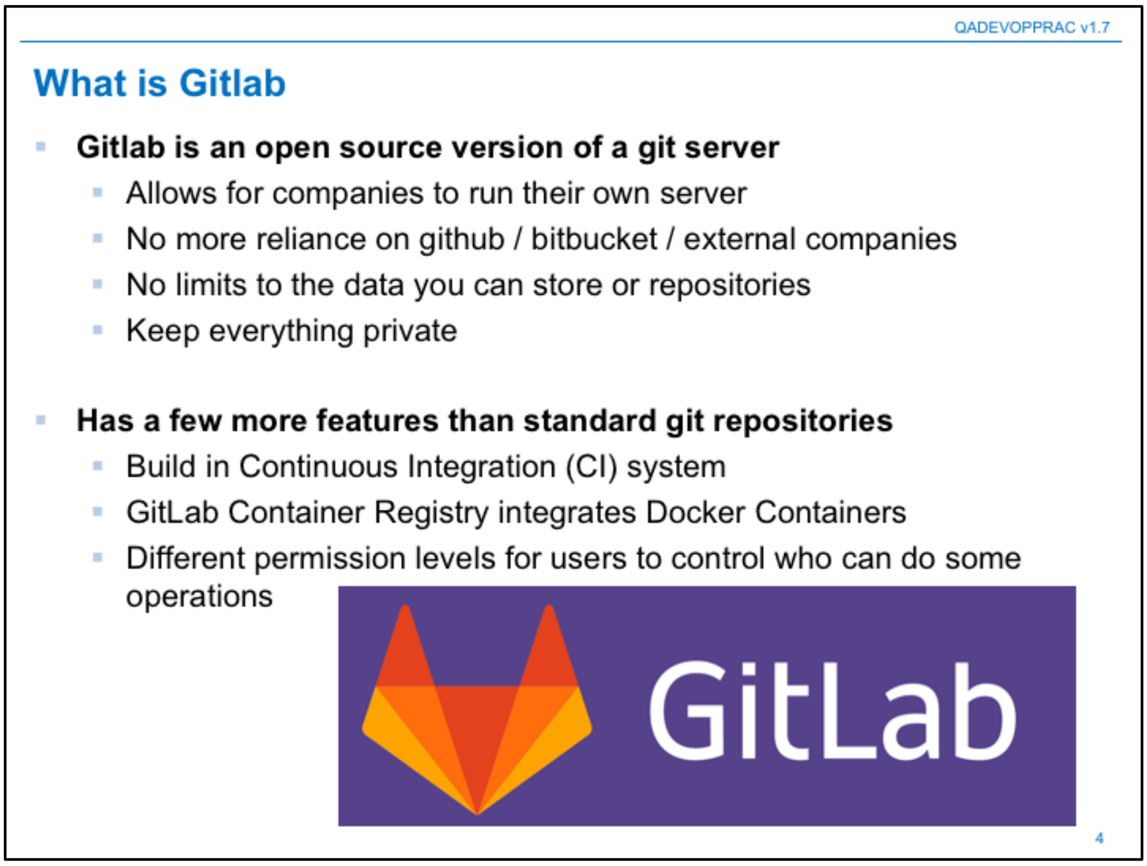
Outline

- **Hello Gitlab**
  - What is gitlab
  - Installing gitlab community edition
- **Using Gitlab**
  - First account
  - Adding other users
- **Backup**
  - When source control goes down
  - Backing up gitlab



Objective

- **By the end of this session you should be able to**
  - Setup your own gitlab server
  - Secure the server
  - Backup the server
  - Recreate the server with the old settings



What is Gitlab

- **Gitlab is an open source version of a git server**
  - Allows for companies to run their own server
  - No more reliance on github / bitbucket / external companies
  - No limits to the data you can store or repositories
  - Keep everything private
- **Has a few more features than standard git repositories**
  - Build in Continuous Integration (CI) system
  - GitLab Container Registry integrates Docker Containers
  - Different permission levels for users to control who can do some operations

GitLab

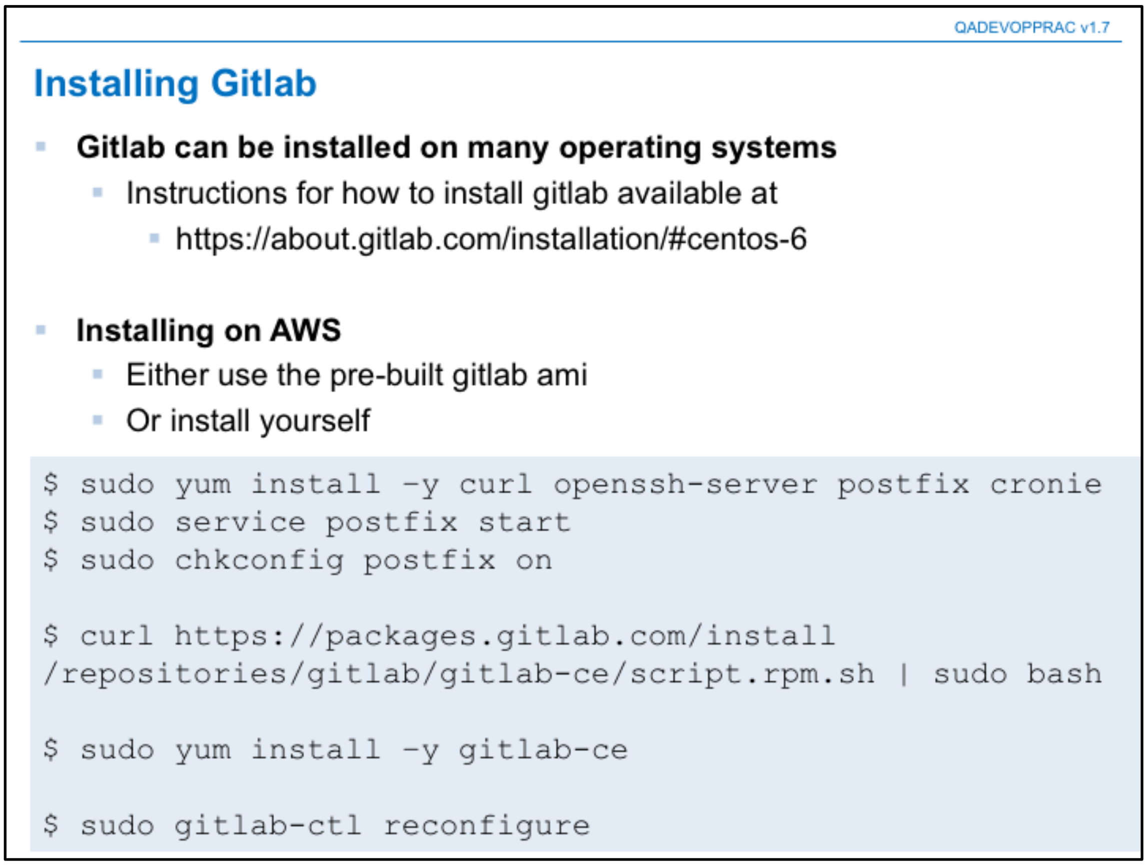
4

On the gitlab website there is even a page about why it is better than github - https://about.gitlab.com/better-than-github/

Community vs Enterprise Edition

https://about.gitlab.com/features/#compare

We will be using the community edition as this is the open source (and free) version



QADEVOPPRAC v1.7

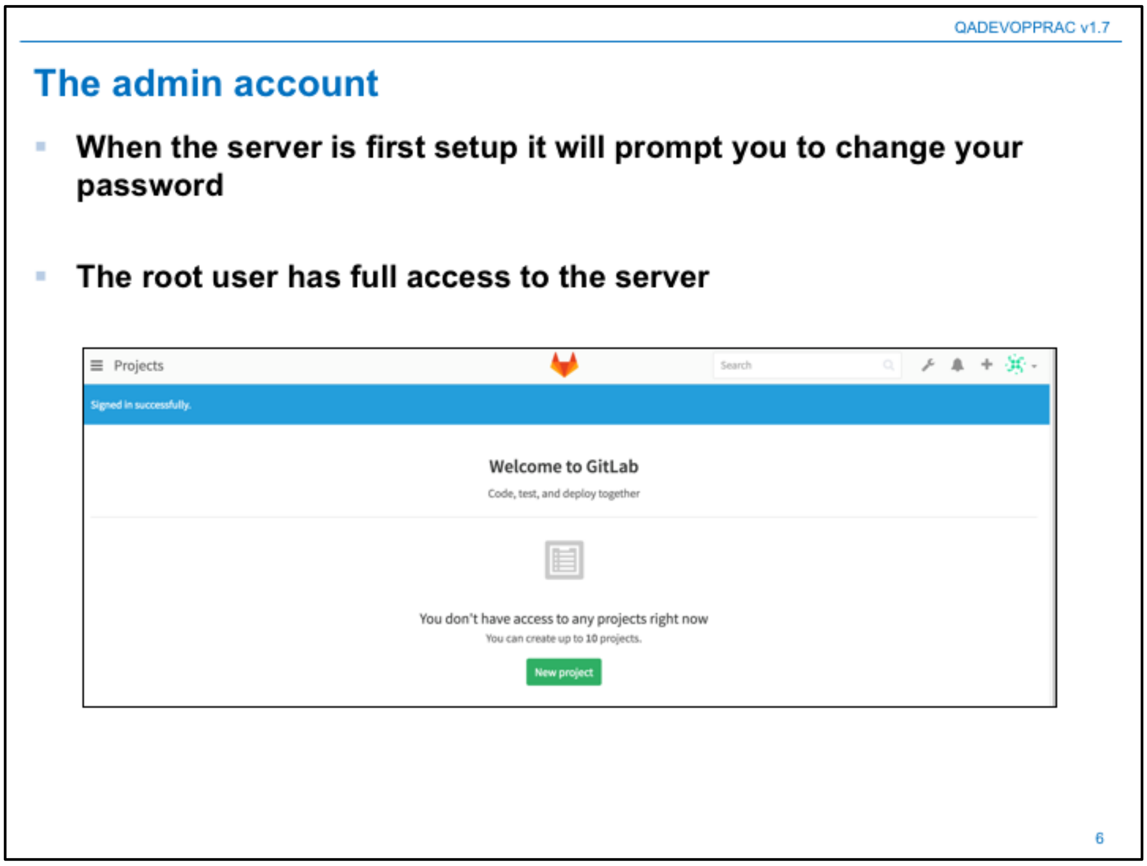
Installing Gitlab

- **Gitlab can be installed on many operating systems**
  - Instructions for how to install gitlab available at
    - https://about.gitlab.com/installation/#centos-6
- **Installing on AWS**
  - Either use the pre-built gitlab ami
  - Or install yourself

```
$ sudo yum install -y curl openssh-server postfix cronie
$ sudo service postfix start
$ sudo chkconfig postfix on

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

$ sudo yum install -y gitlab-ce

$ sudo gitlab-ctl reconfigure
```

Often you will need to run the reconfigure line twice.
Gitlab uses chef to configure the server, but the script assumes that you have the bridge module available in your kernal.
To enable this ahead of time type:

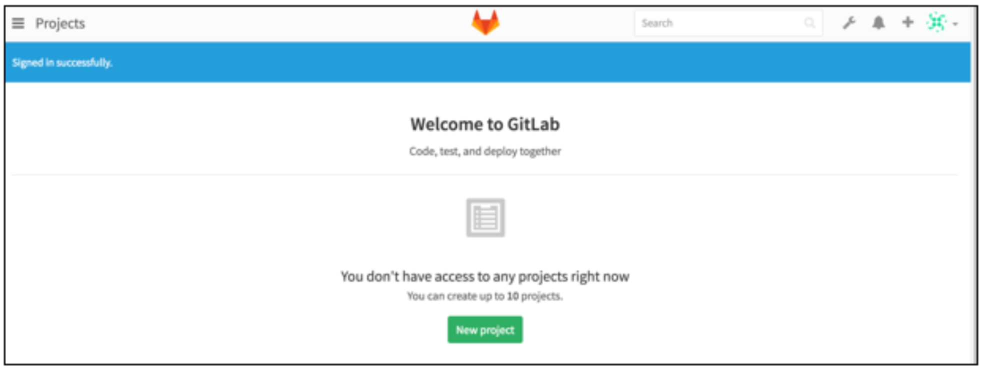
```
sudo modprobe bridge
```



QADEVOPPRAC v1.7

The admin account

- When the server is first setup it will prompt you to change your password
- The root user has full access to the server



Welcome to GitLab

Code, test, and deploy together

You don't have access to any projects right now

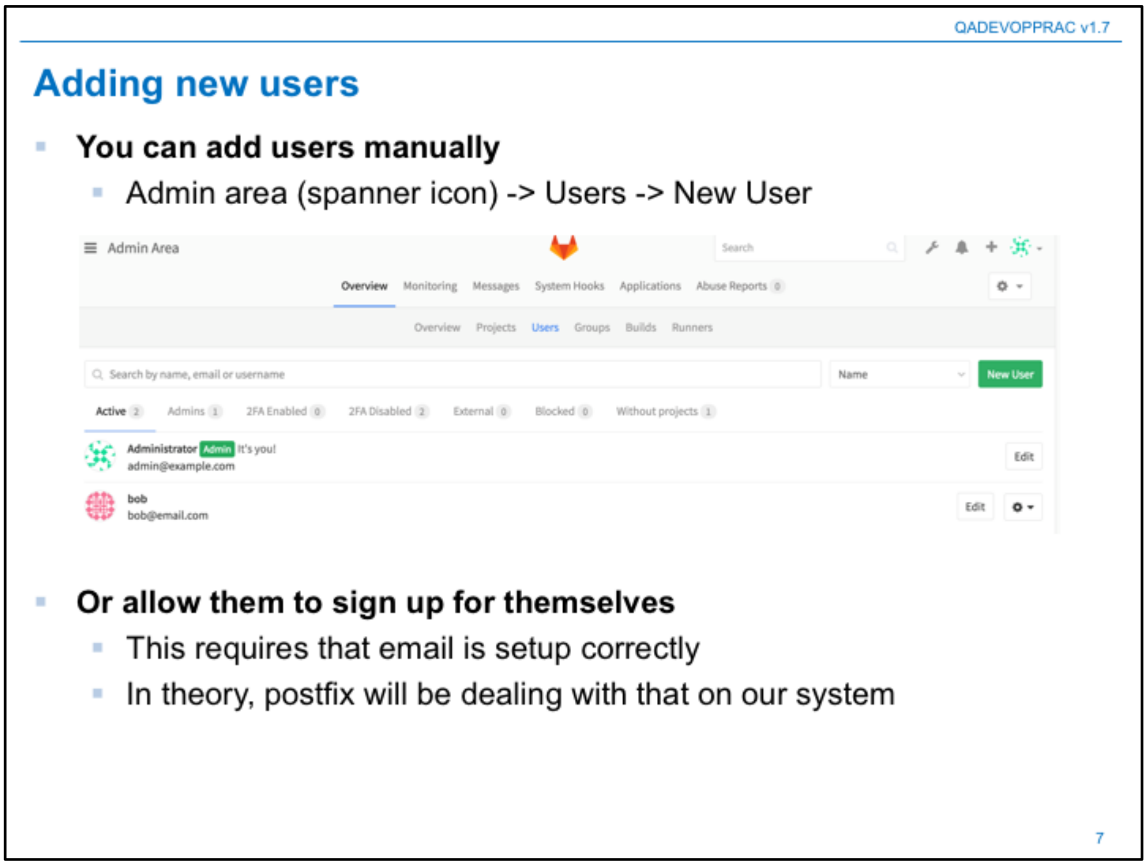
You can create up to 10 projects.

6

In previous versions you needed a username and password:

Username: root

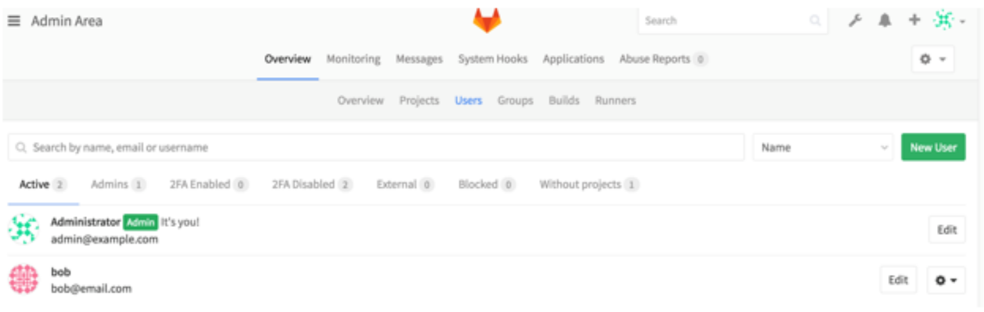
Password: 5iveL!fe



QADEVOPPRAC v1.7

Adding new users

- **You can add users manually**
  - Admin area (spanner icon) -> Users -> New User

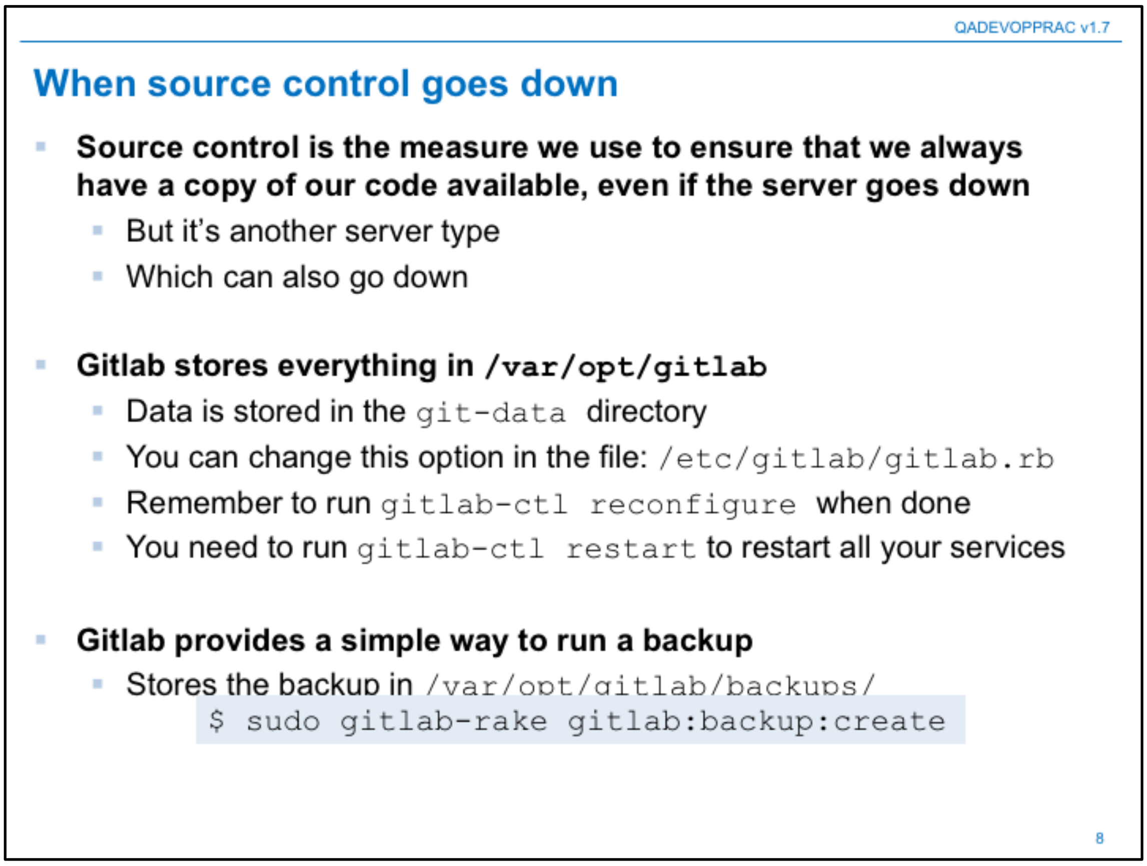


- **Or allow them to sign up for themselves**
  - This requires that email is setup correctly
  - In theory, postfix will be dealing with that on our system

7

You can see if postfix has sent an email via the command line by reading the logs:

```
$ sudo cat /var/log/maillog
```



QADEVOPPRAC v1.7

When source control goes down

- **Source control is the measure we use to ensure that we always have a copy of our code available, even if the server goes down**
  - But it's another server type
  - Which can also go down
- **Gitlab stores everything in `/var/opt/gitlab`**
  - Data is stored in the `git-data` directory
  - You can change this option in the file: `/etc/gitlab/gitlab.rb`
  - Remember to run `gitlab-ctl reconfigure` when done
  - You need to run `gitlab-ctl restart` to restart all your services
- **Gitlab provides a simple way to run a backup**
  - Stores the backup in `/var/opt/gitlab/backups/`

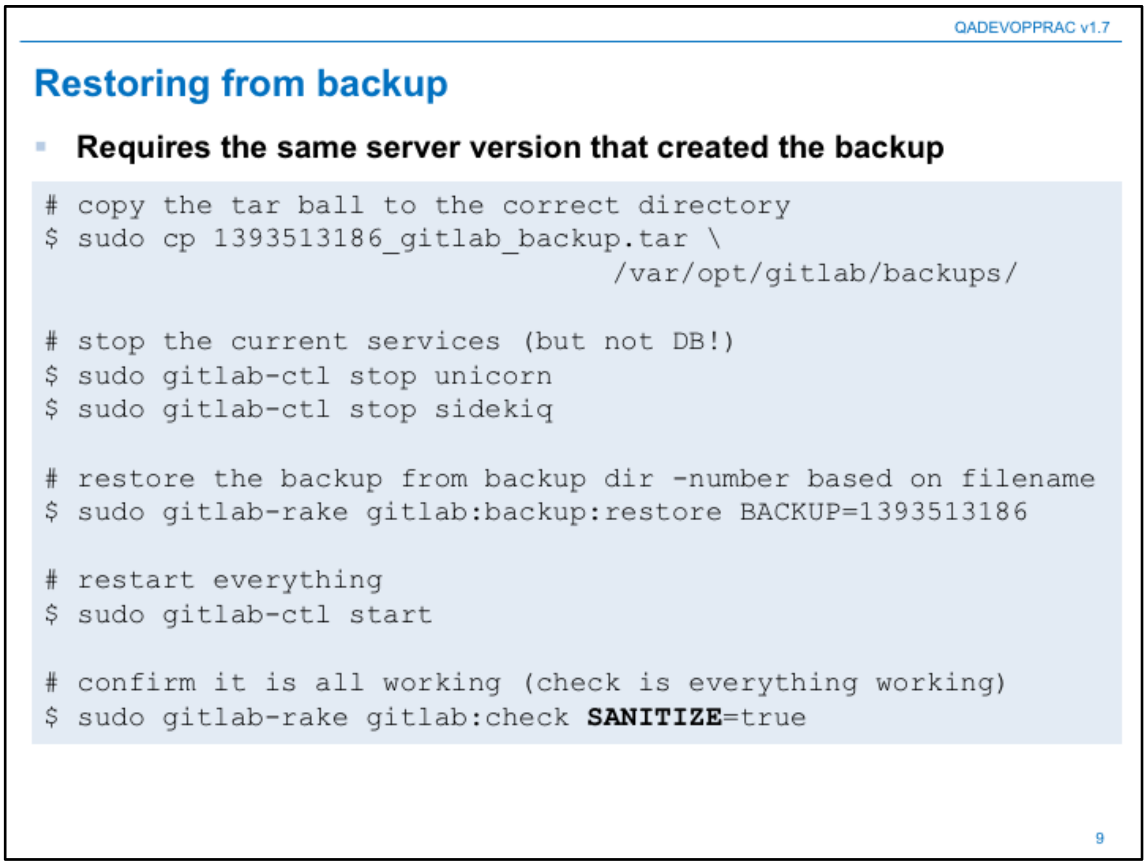
```
$ sudo gitlab-rake gitlab:backup:create
```

8

If you have two factor authentication you need to manually back up the /etc/gitlab/gitlab-secrets.json file. This contains the database encryption key used for TFA users.

For config backups: https://gitlab.com/gitlab-org/omnibus-gitlab/blob/master/doc/settings/backups.md#backup-and-restore-omnibus-gitlab-configuration

`$ sudo gitlab-rake gitlab:backup:create` – grabs every single info, creates backup zip file WITHOUT restarting servers (work done in milliseconds)

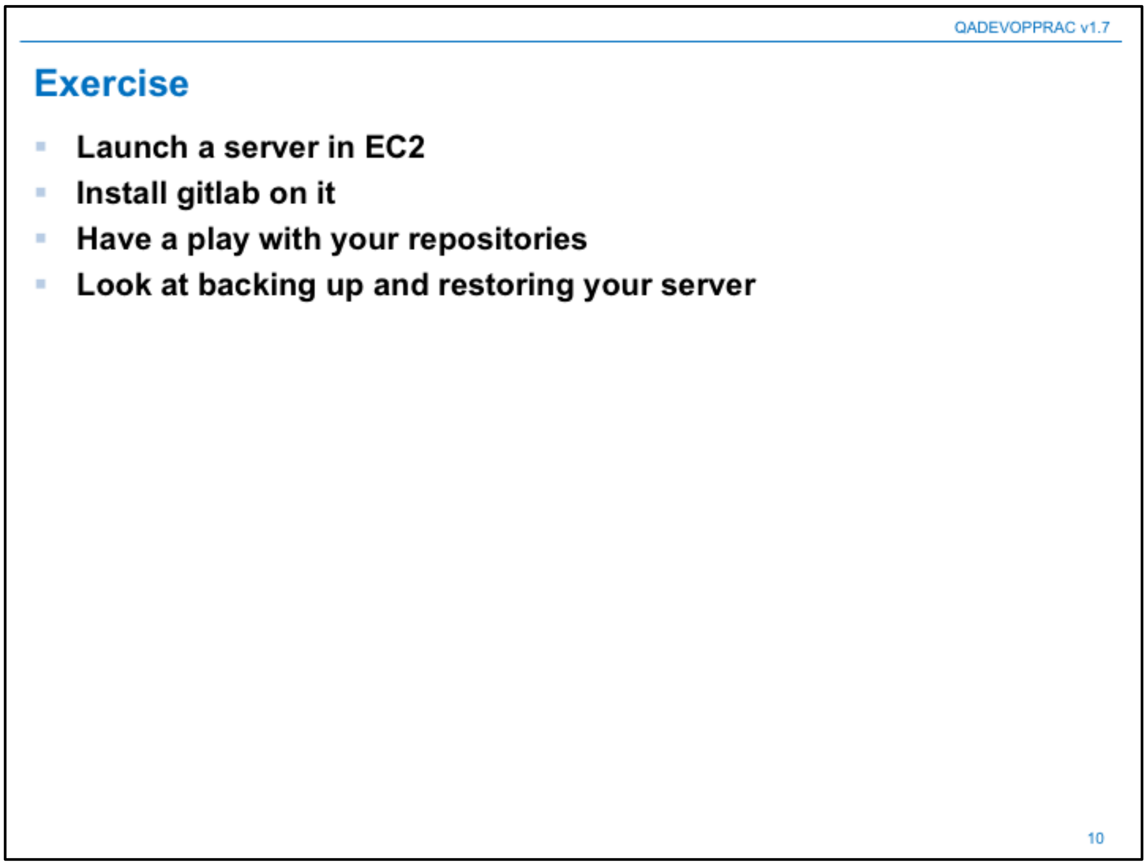


Restoring from backup

- **Requires the same server version that created the backup**

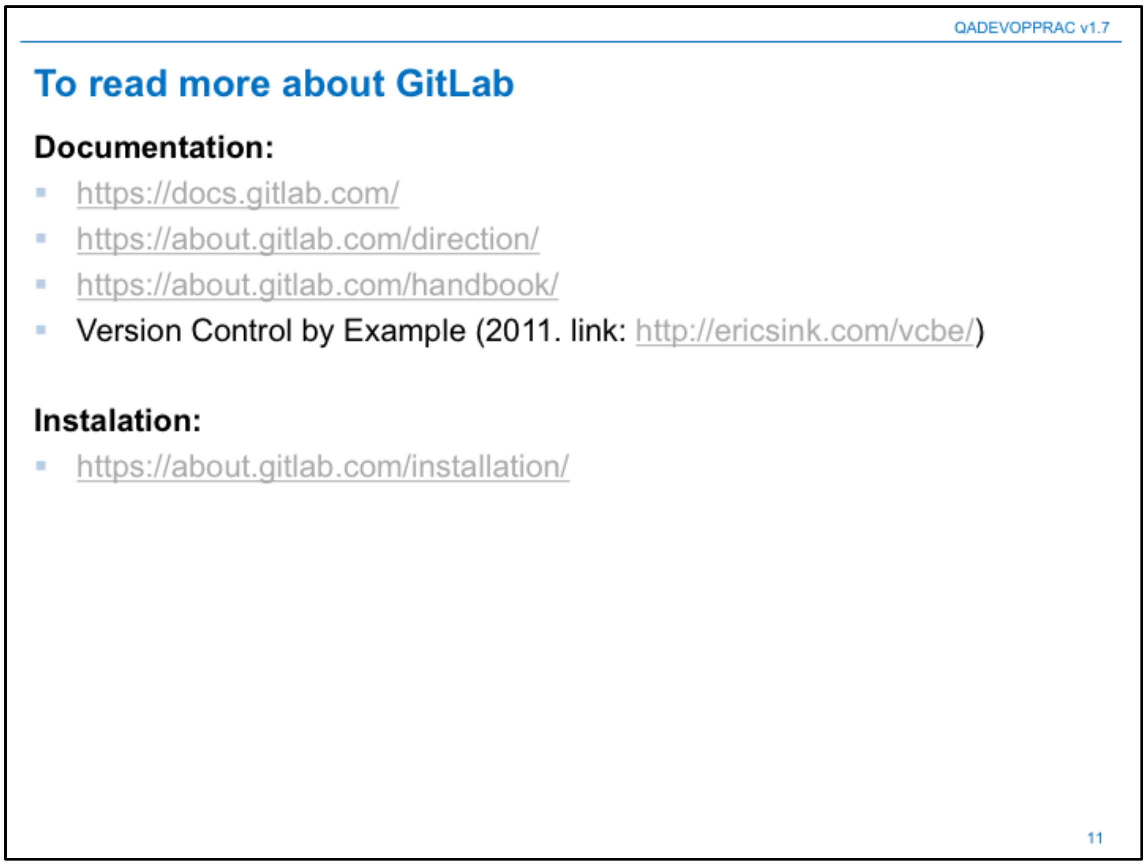
```
# copy the tar ball to the correct directory
$ sudo cp 1393513186_gitlab_backup.tar \
                                /var/opt/gitlab/backups/

# stop the current services (but not DB!)
$ sudo gitlab-ctl stop unicorn
$ sudo gitlab-ctl stop sidekiq

# restore the backup from backup dir -number based on filename
$ sudo gitlab-rake gitlab:backup:restore BACKUP=1393513186

# restart everything
$ sudo gitlab-ctl start

# confirm it is all working (check is everything working)
$ sudo gitlab-rake gitlab:check SANITIZE=true
```

See https://gitlab.com/gitlab-org/gitlab-ce/blob/master/doc/raketasks/backup_restore.md for more information



Exercise

- Launch a server in EC2
- Install gitlab on it
- Have a play with your repositories
- Look at backing up and restoring your server



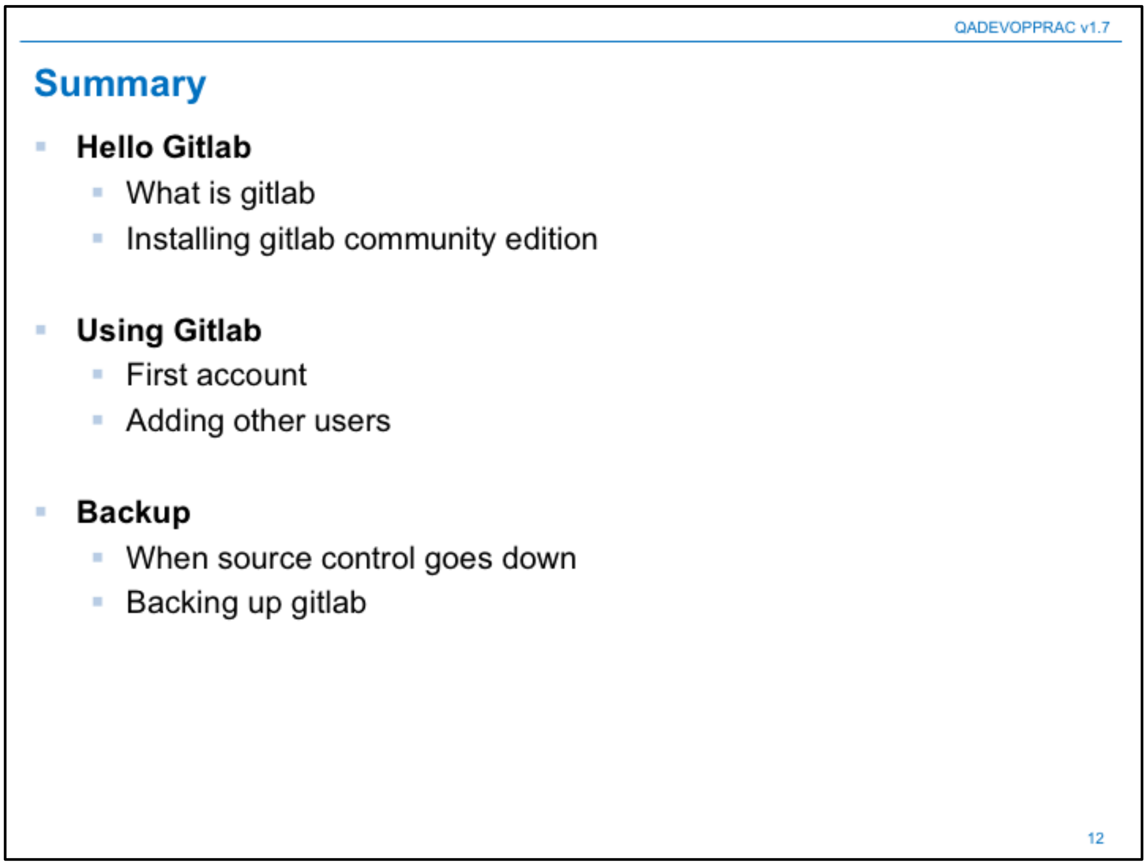
To read more about GitLab

Documentation:

- https://docs.gitlab.com/
- https://about.gitlab.com/direction/
- https://about.gitlab.com/handbook/
- Version Control by Example (2011. link: http://ericsink.com/vcbe/)

Instalation:

- https://about.gitlab.com/installation/



Summary

- **Hello Gitlab**
  - What is gitlab
  - Installing gitlab community edition
- **Using Gitlab**
  - First account
  - Adding other users
- **Backup**
  - When source control goes down
  - Backing up gitlab