GitLab Community Edition

Version:	1.0.0
Created by:	cloudimg

Table of Contents

1.) Overview

This document is provided as a user guide for the GitLab Community Edition product offering on the AWS Marketplace. Please reach out to support@cloudimg.co.uk if any issues are encountered following this user guide for the chosen product offering.

2.) Access & Security

Please update the security group of the target instance to allow the below ports and protocols for access and connectivity.

Protocol	Type	Port	Description
SSH	TCP	22	SSH connectivity
TCP	TCP	80	GitLab Front End URL
TCP	TCP	5432	PostgreSQL Database

3.)System Requirements

The minimum system requirements for the chosen product offering can be found below

Minimum CPU	Minimum RAM	Required Disk Space
2	4 GB	20 GB

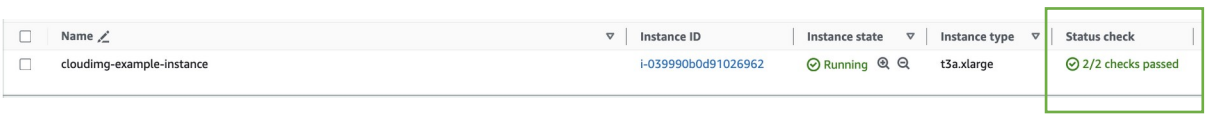
4.) Connecting to the Instance

Once launched in the Amazon EC2 Service, please connect to the instance via an SSH client using the **ec2-user** with the key pair associated at launch. Once connected as the **ec2-user** user, you will be able to sudo to the **root** user by issuing the below command.

Switch to the root user.

```
sudo su -
```

NOTE: Please allow the EC2 Instance to reach 2/2 successful status checks to ensure you will be able to connect successfully with the ec2-key pair assigned at launch. Upon attempting to SSH to early you may receive errors such as below, this is expected with an early SSH connection. Allow the EC2 instance to reach 2/2 status checks and you will be able successfully connect with the ec2-key pair assigned at launch as the ec2-user.



Example errors you may receive with an early SSH connection.

```
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).

ec2-user@your-instance-ip's password:
```


cloudimg
(+44) 02045382725
3rd Floor 86-90 Paul Street London EC2A 4NE
support@cloudimg.co.uk
https://cloudimg.co.uk

5.) On Startup

An OS package update script has been configured to run on boot to ensure the image is fully up to date at first use. You can disable this feature by removing the script from /stage/scripts/ and deleting the entry in crontab for the root user.

Disable the OS update script from running on reboot

```
rm -f /stage/scripts/initial_boot_update.sh

crontab -e

#DELETE THE BELOW LINE. SAVE AND EXIT THE FILE.
@reboot /stage/scripts/initial_boot_update.sh
```

6.) Filesystem Configuration

Please see below for a screenshot of the server disk configuration and specific mount point mappings for software locations.

```
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        1.9G     0  1.9G   0% /dev
tmpfs           1.9G   28K  1.9G   1% /dev/shm
tmpfs           1.9G   13M  1.9G   1% /run
tmpfs           1.9G     0  1.9G   0% /sys/fs/cgroup
/dev/nvme0n1p2   38G  5.2G   31G  15% /
/dev/nvme1n1    9.8G  313M  8.9G   4% /var/opt/gitlab
/dev/nvme0n1p1  2.0G  121M  1.7G   7% /boot
tmpfs           390M     0  390M   0% /run/user/1002
tmpfs           390M     0  390M   0% /run/user/0
```

Mount Point	Description
/boot	Operating System Kernel files
/var/opt/gitlab	GitLab Data Directory

7.) Server Components

Please see below for a list of installed server components and their respective installation paths. The below versions are subject to change on initial boot based on the initial_boot_update.sh script finding new versions of the software in the systems package repositories.

Component	Software Home
GitLab Community Edition	/etc/gitlab/gitlab.rb
PostgreSQL Database	/var/opt/gitlab/postgresql

8.) Scripts and Log Files

The below table provides a breakdown of any scripts & log files created to enhance the useability of the chosen offering.

Script/Log	Path	Description
Initial_boot_update.sh	/stage/scripts	Update the Operating System with the latest updates available.
Initial_boot_update.log	/stage/scripts	Provides output for initial_boot_update.sh
gitlab-ce-set-private-url.sh	/stage/scripts	Configure GitLab Front End URL to be the Private IP address of the EC2 Instance.
gitlab-ce-set-public-url.sh	/stage/scripts	Configure GitLab Front End URL to be the Public IP address of the EC2 Instance.
gitlab-ce-set-dns-hostname-url-http.sh	/stage/scripts	Configure GitLab Front End URL to be the DNS address of the EC2 Instance.

9.) Using System Components

Instructions can be found below for using each component of the server build mentioned in section 7 of this user guide document.

GitLab

3x GitLab installation scripts have been created and located under the /stage/scripts directory for setting the GitLab front end URL based on each of the below use cases. Execute one of the below scripts matching the needs of your use case.

Each script will reconfigure the GitLab Front End URL set in the /etc/gitlab/gitlab.rb file and run gitlab-ctl reconfigure.

The average runtime for the configuration scripts mentioned below is 3minutes, 20seconds. This timing has been collected by running each of the scripts on a t3.medium, t3.large & t3.xlarge instance type to gather the average time required.

/stage/scripts/gitlab-ce-set-private-url.sh

Please execute this script via the below command as the **root** user if you wish to access the GitLab front end application via the URL: ***http://PRIVATE_IP_OF_YOUR_EC2_INSTANCE***

```
/stage/scripts/gitlab-ce-set-private-url.sh
```

/stage/scripts/gitlab-ce-set-public-url.sh

Please execute this script via the below command as the **root** user if you wish to access the GitLab front end application via the URL: ***http://PUBLIC_IP_OF_YOUR_EC2_INSTANCE***

```
 /stage/scripts/gitlab-ce-set-public-url.sh
```

/stage/scripts/gitlab-ce-set-hostname-url-http.sh

Please execute this script via the below command as the **root** user if you wish to access the GitLab front end application via the URL: ***http://DNS_HOSTNAME_OF_YOUR_CHOOSING***

```
/stage/scripts/gitlab-ce-set-dns-hostname-url-http.sh
```

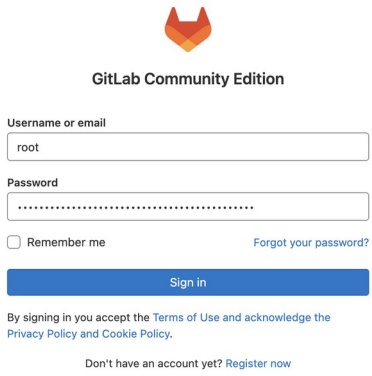
Upon successful completion of any script mentioned above, output resembling the below code block should be present as output.

```
Notes:

Default admin account has been configured with following details:

Username: root

Password: You didn't opt-in to print initial root password to STDOUT.

Password stored to /etc/gitlab/initial_root_password. This file will be cleaned up in first
reconfigure run after 24 hours.
```

```
NOTE: Because these credentials might be present in your log files in plain text, it is highly
recommended to reset the password following
https://docs.gitlab.com/ee/security/reset_user_password.html#reset-your-root-password.


gitlab Reconfigured!
```

You will now be able to access the GitLab Front End via the URL set based on the script used for your given use case. This example will assume the use of the public-url script.

Navigate to a Web Browser and navigate to URL: http://PUBLIC_IP



Username – root

Password - Please refer to the file generated on the instance: /etc/gitlab/initial_root_password for the randomly generated password value. This value will only be stored for 24hours so please make a record of this value if you do not intend to set a custom value via the below instructions.

Setting custom GitLab root user password
Follow the steps below to set a custom password value for the **root** user of the GitLab Application Front End.

From the EC2 Instance command prompt as the **root** user. Run the below command if you wish to set a new password value.

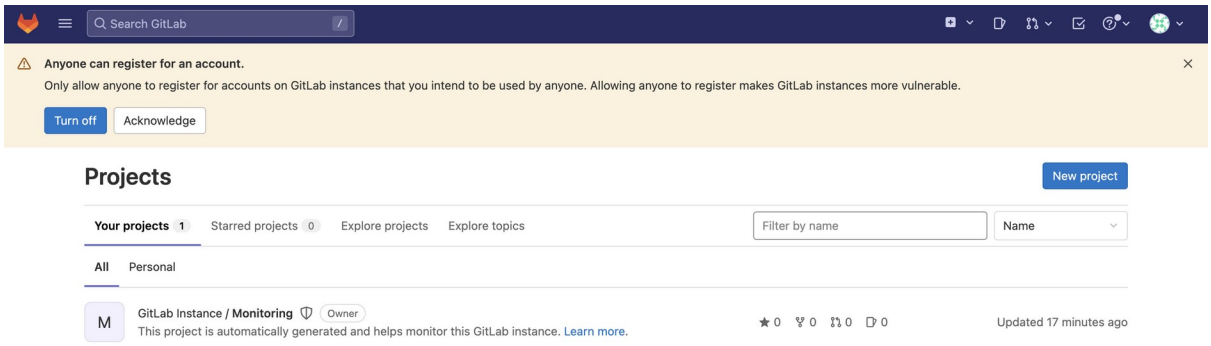
```
```

```
gitlab-rake "gitlab:password:reset[root]"
```

Allow 30 seconds for the command prompt to return, prompting for the new password.

```
[root@ip-172-31-87-42 scripts]# gitlab-rake "gitlab:password:reset[root]"
Enter password:
Confirm password:
Password successfully updated for user with username root.
[root@ip-172-31-87-42 scripts]#
```

You will now be able to login into the GitLab Front End URL with the username **root** and newly set password.



GitLab is now available and ready for use.