

Logo of the project

# Phabricator

Phabricator is a suite of web-based software development collaboration tools, including the *Differential* code review tool, the *Diffusion* repository browser, the *Herald* change monitoring tool, the *Maniphest* bug tracker and the *Phriction* wiki. Phabricator integrates with Git, Mercurial, and Subversion. It is available as free software under the Apache License 2.0. Phabricator was originally developed as an internal tool at Facebook. Phabricator’s principal developer is Evan Priestley. Priestley left Facebook to continue Phabricator’s development in a new company called Phacility.

### GitLab vs Phabricator:

**GitLab:** Open source self-hosted Git management software. GitLab offers git repository management, code reviews, issue tracking, activity feeds and wikis. Enterprises install GitLab on-premise and connect it with LDAP and Active Directory servers for secure authentication and authorization. A single GitLab server can handle more than 25,000 users but it is also possible to create a high availability setup with multiple active servers.

**Phabricator:** Open Source, Software Development Platform. Phabricator is a collection of open source web applications that help software companies build better software.

GitLab and Phabricator are primarily classified as “Code Collaboration & Version Control” and “Code Review” tools respectively.

Some of the features offered by GitLab are:

Manage git repositories with fine grained access controls that keep your code secure  
Perform code reviews and enhance collaboration with merge requests  
Each project can also have an issue tracker and a wiki

On the other hand, Phabricator provides the following key features:

reviewing code before it hits master  
auditing code after it hits master  
hosting Git/Hg/SVN repositories

“Self hosted” is the primary reason why developers consider GitLab over the competitors, whereas “Open Source” was stated as the key factor in picking Phabricator.

GitLab is an open source tool with 20.1K GitHub stars and 5.33K GitHub forks.

### Phabricator / GitLab Integration:

GitLab allows you to import all tasks from a Phabricator instance into GitLab issues. The import creates a single project with the repository disabled.

Currently, only the following basic fields are imported:

Title  
Description  
State (open or closed)  
Created at  
Closed at

#### Users

The assignee and author of a user are deducted from a Task’s owner and author: If a user with the same username has access to the namespace of the project being imported into, then the user will be linked. Enabling this feature

While this feature is incomplete, a feature flag is required to enable it so that we can gain early feedback before releasing it for everyone. To enable it:

1. Run the following command in a Rails console:

\*Feature.enable(:phabricator\_import)\*

1. Enable Phabricator as an import source in the Admin Area.

## Assumptions

This procedure pulls artifacts the following repositories : -

amzn2-core  
 httpd  
 amazon-linux-extras  
 epel  
 webtatic  
 php55w  
github.com/phacility  
 libphutil  
 arcanist

These artifacts are necessary for the installation of Phabricator, but are not available on Artifactory. It is assumed that it is permitted to pull these artifacts in lieu of their being made available on Artifactory.

## Installing / Getting started

These are the basic install steps to get Phabricator up and running …

1. Update the Operating System   
2. Install MariaDB  
3. Install Apache  
4. Install PHP 7.3  
5. Download and install Phabricator  
6. Setup a virtual host for Phabricator  
7. Setup the MariaDB credentials for Phabricator  
8. Modify firewall rules and setup a Phabricator admin account  
9. Resolve Issues

You should be presented with the Phabricator registration web page. Here you can create an administrator account for daily management.

### Initial Configuration

#### 1. Update the Operating System

sudo yum --disablerepo="\*" --enablerepo="amzn2-core"  
sudo yum update -y   
sudo yum clean all

#### 2. Install MySQL

##### Check Prerequisites

yum list installed | grep libaio  
sudo groupadd -r -g 1007 mysql  
sudo useradd -r -u 1007 -g 1007 -s /bin/false mysql  
mkdir phabricator  
cd phabricator

##### Download MySQL

curl -L -o mysql-8.0.19-linux-glibc2.12-x86\_64.tar.xz \  
http://10.176.72.70:5558/artifactory/cdos-tariff/dev\_software/mysql/mysql-8.0.19-linux-glibc2.12-x86\_64.tar.xz  
xz -dv mysql-8.0.19-linux-glibc2.12-x86\_64.tar.xz  
tar xvf mysql-8.0.19-linux-glibc2.12-x86\_64.tar  
rm -f mysql-8.0.19-linux-glibc2.12-x86\_64.tar

##### Install MySQL

sudo mv mysql-8.0.19-linux-glibc2.12-x86\_64 /usr/local  
sudo chown root:root /usr/local/mysql-8.0.19-linux-glibc2.12-x86\_64  
sudo ln -s /usr/local/mysql-8.0.19-linux-glibc2.12-x86\_64 /usr/local/mysql  
sudo mkdir /usr/local/mysql/mysql-files  
sudo chown mysql:mysql /usr/local/mysql/mysql-files  
sudo chmod 750 /usr/local/mysql/mysql-files

##### Configure MySQL

sudo /usr/local/mysql/bin/mysqld --initialize-insecure --user=mysql  
sudo /usr/local/mysql/bin/mysql\_ssl\_rsa\_setup  
sudo /usr/local/mysql/bin/mysqld\_safe --user=mysql &  
sudo cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysql.server  
sudo systemctl enable mysql.server  
sudo systemctl start mysql.server

##### Test MySQL & Set Root Passowrd

mysql -h localhost -u root --skip-password  
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'root'; FLUSH PRIVILEGES;

#### 3. Install Apache

##### Install httpd from amzn2-core

sudo yum --disablerepo="\*" --enablerepo="amzn2-core" install -y httpd

##### Configure httpd

sudo systemctl enable httpd  
sudo systemctl start httpd

##### Test Apache

curl http://localhost

#### 4. Install PHP

##### Install PHP

yum --disablerepo="\*" --enablerepo="amzn2-core" install -y amazon-linux-extras  
amazon-linux-extras install epel  
rpm -Uvh https://mirror.webtatic.com/yum/el7/webtatic-release.rpm  
yum --disablerepo="\*" --enablerepo="webtatic" install -y php55w php55w-cli php55w-mysql php55w-process php55w-common php55w-pdo php55w-mbstring  
sudo yum install -y php55w php55w-cli php55w-mysql php55w-process php55w-common php55w-pdo php55w-mbstring  
cat > phpinfo.php <<EOF  
<?php  
  
// Show all information, defaults to INFO\_ALL  
phpinfo();  
  
?>  
EOF  
sudo mv phpinfo.php /var/www/html  
sudo chown apache:apache /var/www/html/phpinfo.php

##### Restart Apache

sudo systemctl restart httpd

##### Test PHP

curl http://localhost/phpinfo.php

#### 5. Download and install Phabricator

##### Clone Arcanist and libphutil from Phacility’s github repository

mkdir phabricator  
git clone https://github.com/phacility/libphutil.git phabricator/libphutil  
git clone https://github.com/phacility/arcanist.git phabricator/arcanist

##### Download Phabricator from Artifactory

curl -L -o phabricator-master.zip \  
http://10.176.72.70:5558/artifactory/cdos-tariff/dev\_software/phabricator/phabricator-master.zip  
unzip phabricator-master.zip  
rm -f phabricator-master.zip

##### Install Phabricator

mv phabricator-master phabricator/phabricator  
sudo cp -r phabricator/phabricator /var/www/html  
sudo chown -R apache:apache /var/www/html/phabricator

#### 6. Setup a virtual host for Phabricator

##### Configure Phabricator

cat > phabricator.conf <<'EOF'  
<VirtualHost \*:80>  
 ServerAdmin tim.haw@hmrc.gov.uk  
 DocumentRoot /var/www/html/phabricator/phabricator/webroot/  
 ServerName phabricator.hmrc.gov.uk  
 ServerAlias www.phabricator.hmrc.gov.uk  
 RewriteEngine on  
 RewriteRule ^/rsrc/(.\*) - [L,QSA]  
 RewriteRule ^/favicon.ico - [L,QSA]  
 RewriteRule ^(.\*)$ /index.php?\_\_path\_\_=$1 [B,L,QSA]  
 <Directory /var/www/html/phabricator/phabricator/webroot/>  
 AllowOverride All  
 </Directory>  
 ErrorLog /var/log/httpd/phabricator.hmrc.gov.uk-error\_log  
 CustomLog /var/log/httpd/phabricator.hmrc.gov.uk-access\_log common  
</VirtualHost>  
EOF  
sudo mv phabricator.conf /etc/httpd/conf.d/

##### Restart Apache

sudo systemctl restart httpd

#### 7. Setup the MySQL credentials for Phabricator

##### Run the MySQL configuration scripts

sudo /var/www/html/phabricator/phabricator/bin/config set mysql.host localhost  
sudo /var/www/html/phabricator/phabricator/bin/config set mysql.port 3306  
sudo /var/www/html/phabricator/phabricator/bin/config set mysql.user root  
sudo /var/www/html/phabricator/phabricator/bin/config set mysql.pass root

##### Populate Phabricator Schemas

sudo /var/www/html/phabricator/phabricator/bin/storage upgrade --force

##### Create storage directory /var/repo

sudo mkdir /var/repo  
sudo chown apache:apache /var/repo

#### 8. Setup a Phabricator admin account

sudo systemctl restart mysql.service  
sudo systemctl restart httpd.service  
sudo /var/www/html/phabricator/phabricator/bin/phd start

Open a web browser, and browse to : -

http://<your\_d4d\_ip\_address>/auth/register/

Enter a username (e.g. admin)  
Enter a Full Name (e.g. Administrator)  
Enter an E-mail address (e.g. admin@hmrc.gov.uk)  
  
Click Auth -> Add Provider -> Username/Password -> Add Provider  
Click on Phabricator  
Click on People -> admin (Administrator) -> Manage -> Send Welcome Email -> Phabricator Welcome Email -> Send Email

As the mailer is not set up yet, we need another way to intercept the welcome e-mail that was just sent to the admin user. To do this, in a command shell, enter the following commands : -

sudo /var/www/html/phabricator/phabricator/bin/mail list-outbound  
sudo /var/www/html/phabricator/phabricator/bin/mail show-outbound --id 1

The content of the welcome e-mail will be displayed. Back to the web browser window, and follow the link given in the welcome e-mail Follow the link (e.g. http://10.176.149.2/login/once/welcome/1/o3y7vqb5poix54hjl6hejmg43sd63a7o/1/)

Enter Password: (e.g. Ph@br1c@t0r) Confirm Password: Ph@br1c@t0r -> Set Account Password

### Import or Observe an Existing Repository

Click on Diffusion -> New Repository -> Import or Observe an Existing repository

Create a repository in Diffusion, but do not activate it yet.

Click on Create a new Git repository  
Name: (e.g. Phabricator Software Development Tools)  
Callsign: (e.g. PHAB \*note all upper-case\*)  
Short Name: (e.g. phabricator)  
Description: (e.g. Phabricator is a suite of web-based software development collaboration tools, including the Differential code review tool, the Diffusion repository browser, the Herald change monitoring tool, the Maniphest bug tracker and the Phriction wiki. Phabricator integrates with Git, Mercurial, and Subversion.)  
Tags: (e.g. DTO - C&IT RoRo)  
Click on Create Repository

Add the URI for the existing GitLab repository you wish to observe in the URIs section, in Observe mode.

Click on URIs -> Add New URI  
URI: (e.g. git@10.102.83.38:7839703/phabricator.git)  
I/O Type: Observe: Copy from a remote  
Display Type: Visible: Show as a clone URI  
Click on Create Repository URI  
Click on Set Credential -> Add New Credential  
Name: (e.g. COLUMBUS\u.7839703)  
Description: (copy & paste public key)  
Visible To: Credential Author  
Editable By: Credential Author  
Login/Username: git  
Private Key: (copy & paste private key)  
Password for Key: (leave blank)  
Leave \*\*Lock Permanently\*\* unchecked.  
Click on Create Credential -> Set Credential

Activate the repository in Diffusion.

Click on Diffusion -> All Repositories -> Phabricator ...  
Click on Actions -> Manage Repository  
Click on Activate Repository -> Activate Repository

To import the repository, once the observed repository is fully synced up, then change the "I/O Type" on the Observe URI to "No I/O".  
  
To push to an empty repository, create and activate an empty repository, then push all of your changes to the empty repository. In Git, this is done with "git push"

Click on Diffusion -> New Repository -> Create a new Git repository

Show issues: http://10.176.149.2/config/issue/

http://10.176.149.2/auth/config/new/

Add Auth Provider: Username/Password

sudo /var/www/html/phabricator/phabricator/bin/accountadmin

sudo /var/www/html/phabricator/phabricator/bin/mail list-outbound sudo /var/www/html/phabricator/phabricator/bin/mail show-outbound –id 1

http://10.176.149.2/login/once/welcome/2/lriev7jozytuggblnabryfvzrxj42vc3/2/

sudo /var/www/html/phabricator/phabricator/bin/auth recover admin ``` #Required PHP extensions are not installed ‘mbstring’ sudo yum install -y php55w-mbstring sudo systemctl restart httpd

##### Start the Phabricator daemons

sudo /var/www/html/phabricator/phabricator/bin/phd start

## Final Steps

#### Resolve Issues

##### Show Issues

http://10.176.149.2/config/issue/

##### 1. Phabricator Daemons Are Not Running

sudo su -  
cd /var/www/html/phabricator/phabricator  
./bin/phd start

##### 2. Base URI Not Configured

sudo su -  
cd /var/www/html/phabricator/phabricator  
./bin/config set phabricator.base-uri http://10.176.149.2/

##### 3. No Authentication Providers Configured

http://10.176.149.2/auth/config/new/

##### 4. Disable PHP always\_populate\_raw\_post\_data

/etc/php.ini always\_populate\_raw\_post\_data "-1"

##### 5. MySQL Native Driver Not Available

sudo apt-get install php5-mysqlnd

##### 6. Unsafe PHP “Local Infile” Configuration

/etc/php.ini mysqli.allow\_local\_infile = 0

##### 7. PHP post\_max\_size Not Configured

/etc/php.ini post\_max\_size "32M"

##### 8. Small MySQL “max\_allowed\_packet”

mysql config max\_allowed\_packet "33554432"

##### 9. MySQL STRICT\_ALL\_TABLES Mode Not Set

mysql config add sql\_mode=STRICT\_ALL\_TABLES

##### 10. MySQL May Run Slowly

my.cnf innodb\_buffer\_pool\_size=1600M

##### 11. Unsafe MySQL “local\_infile” Setting Enabled

my.cnf local\_infile=0

##### 12. Install Pygments to Improve Syntax Highlighting

sudo pip install Pygments  
http://10.176.149.2/config/edit/pygments.enabled Use Pygments

##### 13. Large File Storage Not Configured

https://secure.phabricator.com/book/phabricator/article/configuring\_file\_storage/

##### 14. Alternate File Domain Not Configured

http://10.176.149.2/config/edit/security.alternate-file-domain/?issue=security.security.alternate-file-domain  
Use ./bin/config in phabricator/ to edit it

##### 15. Server Timezone Not Configured

/etc/php.ini date.timezone Europe/London  
http://10.176.149.2/config/edit/phabricator.timezone/?issue=config.timezone Europe/London

##### 16. Zend OPcache Not Installed

sudo yum install php55w-Zend OPcache

##### 17, PHP Extension “APCu” Not Installed

sudo yum install php55w-apcu

##### 18. Missing ‘gd’ Extension

sudo yum install php-gd

##### 19. Mailers Not Configured

http://10.176.149.2/config/edit/cluster.mailers  
https://secure.phabricator.com/book/phabricator/article/configuring\_outbound\_email/

#### Optional script to solve most issues

sudo su -  
  
yum install -y python-pygments  
  
mkdir /var/repo  
chown apache: /var/repo  
  
yum install -y php55w-pear php55w-devel php55w-pecl-apcu httpd-devel pcre-devel  
  
cd /var/www/html/phabricator/phabricator/  
./bin/config set phabricator.base-uri 'http://10.176.149.2'  
./bin/config set security.alternate-file-domain https://files.hmrc.gov.uk  
  
sed -i "s/post\_max\_size = 8M/post\_max\_size = 32M/" /etc/php.ini  
sed -i "s/;date.timezone =/date.timezone = Europe\/London/" /etc/php.ini  
sed -i "/; End:/a apc.stat = Off" /etc/php.ini  
sed -i "/; End:/a apc.slam\_defense = Off" /etc/php.ini  
sed -i "/; End:/a apc.write\_lock = On" /etc/php.ini  
sed -i "/; End:/a extension=apcu.so" /etc/php.ini  
  
sed -i "/^socket=/a innodb\_buffer\_pool\_size=1600M # about 40% of your system memory" /etc/my.cnf  
sed -i "/^socket=/a ft\_boolean\_syntax=' |-><()~\*:\"\"&^'" /etc/my.cnf  
sed -i "/^socket=/a ft\_min\_word\_len=3" /etc/my.cnf  
sed -i "/^socket=/a ft\_stopword\_file=/var/www/html/phabricator/phabricator/resources/sql/stopwords.txt" /etc/my.cnf  
sed -i "/^socket=/a sql\_mode=STRICT\_ALL\_TABLES" /etc/my.cnf  
sed -i "/^socket=/a max\_allowed\_packet=32M" /etc/my.cnf  
  
exit

## Developing

Here’s a brief intro about what a developer must do in order to start developing the project further:

git clone https://github.com/your/awesome-project.git  
cd awesome-project/  
packagemanager install

And state what happens step-by-step.

### Building

If your project needs some additional steps for the developer to build the project after some code changes, state them here:

./configure  
make  
make install

Here again you should state what actually happens when the code above gets executed.

### Deploying / Publishing

In case there’s some step you have to take that publishes this project to a server, this is the right time to state it.

packagemanager deploy awesome-project -s server.com -u username -p password

And again you’d need to tell what the previous code actually does.

## Features

What’s all the bells and whistles this project can perform? \* What’s the main functionality \* You can also do another thing \* If you get really randy, you can even do this

## Configuration

Here you should write what are all of the configurations a user can enter when using the project.

#### Argument 1

Type: String  
Default: 'default value'

State what an argument does and how you can use it. If needed, you can provide an example below.

Example:

awesome-project "Some other value" # Prints "You're nailing this readme!"

#### Argument 2

Type: Number|Boolean  
Default: 100

Copy-paste as many of these as you need.

## Contributing

When you publish something open source, one of the greatest motivations is that anyone can just jump in and start contributing to your project.

These paragraphs are meant to welcome those kind souls to feel that they are needed. You should state something like:

“If you’d like to contribute, please fork the repository and use a feature branch. Pull requests are warmly welcome.”

If there’s anything else the developer needs to know (e.g. the code style guide), you should link it here. If there’s a lot of things to take into consideration, it is common to separate this section to its own file called CONTRIBUTING.md (or similar). If so, you should say that it exists here.

## Links

Even though this information can be found inside the project on machine-readable format like in a .json file, it’s good to include a summary of most useful links to humans using your project. You can include links like:

* Project homepage: https://your.github.com/awesome-project/
* Repository: https://github.com/your/awesome-project/
* Issue tracker: https://github.com/your/awesome-project/issues
  + In case of sensitive bugs like security vulnerabilities, please contact my@email.com directly instead of using issue tracker. We value your effort to improve the security and privacy of this project!
* Related projects:
  + Your other project: https://github.com/your/other-project/
  + Someone else’s project: https://github.com/someones/awesome-project/

## Licensing

One really important part: Give your project a proper license. Here you should state what the license is and how to find the text version of the license. Something like:

“The code in this project is licensed under MIT license.”