# Drupal Deployment on Acquia Cloud with GitHub

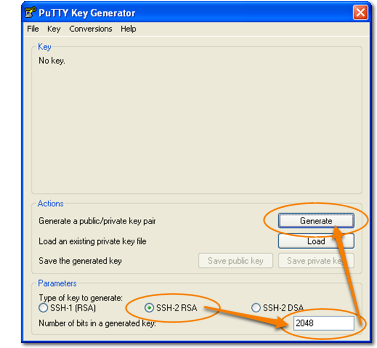
1. Overview

You can use GitHub to deploy your code in Acquia Cloud server. To connect to an Acquia Cloud server, you need to have a private/public key pair and to register the public key in Acquia Cloud.

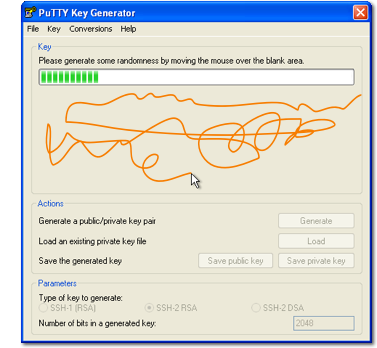
# Generating an SSH public key (Windows)

To generate an SSH public key:

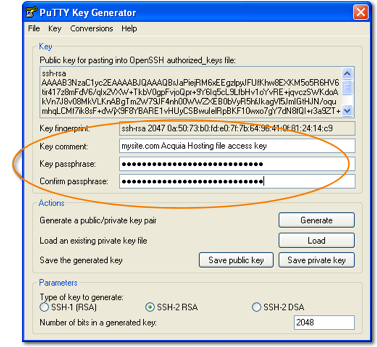
1. Download [PuTTYgen](http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html) to your computer.
2. To run the application, double-click the PuTTYgen icon.
3. Select the **SSH-2 RSA** option, set the number of bits in the generated key to **2048**, and then click**Generate**.



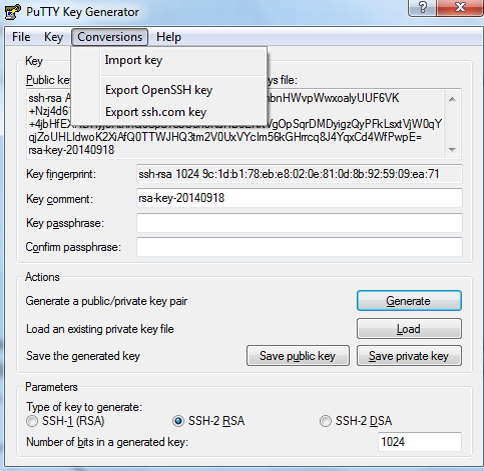
1. Move your mouse over the blank gray area of the PuTTY Key Generator pane. This adds random elements to the key generation process so that you obtain a unique key.



1. Enter a key comment and passphrase in the appropriate fields.
   * **Key comment** - Identify an individual key with a comment. This is useful if you have multiple keys for different accounts.
   * **Key passphrase** - Enter the key passphrase to authenticate your key. Every time you access your Acquia Cloud account you will need to enter this. Some SFTP/SCP/RSYNC programs allow you to save the passphrase for easier repeated access.



1. To save your key in OpenSSH format, which is necessary to ensure that the key is compatible with Acquia Dev Desktop and Acquia Cloud, select **Conversions > Export OpenSSH key**.



1. Click **Save private key**.

Remember where you save this key on your system. You will need it to access your directories on Acquia Cloud. Keep your private key safe. Do not disclose it or publish it where others can find it.

1. Copy to your clipboard the public key displayed on the PuTTYgen panel including **ssh-rsa** at the beginning and your key comment at the end. You can then paste it in the Acquia UI when you add the public key to your Acquia user profile.
2. Click **Save public key**.

# Adding a public key

An SSH public key is required to SSH to servers and environments of sites. You can also use your public key to access your servers using SFTP or rsync, or to access your Git repository.

## 3.1 Adding an SSH key to your Acquia account

You can add a public key to your Acquia account. If you have the proper role and permission as a member of a team, you can then use your public key to SSH to servers and environments of sites your team is assigned to. You can also use your public key to access your servers using SFTP or rsync, or to access your Git repository.

After you have added your SSH public key to your account, you can use it to access an Acquia Cloud server or environment, if:

* You are a member of a team that is assigned to the server's site, and you have a role that includes the appropriate SSH permissions, or
* You are either be a primary or technical contact for the site.

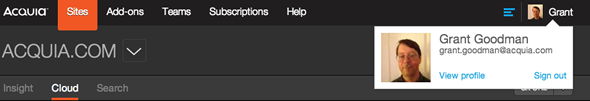
Acquia provides the following permissions related to SSH keys:

* Add SSH key to Git repository
* Add SSH key to non-Production environments
* Add SSH key to the Production environment

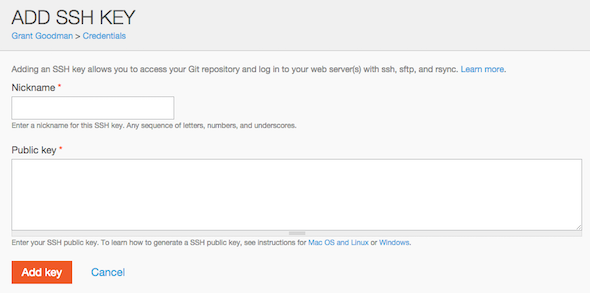
So, if you have just the **Add SSH key to git repository** and **Add SSH key to non-Production environments**permissions, you can check code into and out of your site's Git repository, as well as use SSH to connect to your site's Dev and Stage environments, but you cannot connect to your site's Production environment using SSH. Learn more about [teams and permissions](https://docs.acquia.com/network/teams).

To add a public key to your account:

1. Click your name in the upper right of the Acquia user interface, then click **View profile** to open your Acquia profile.



1. In your Acquia profile, click **Credentials**.
2. On the Credentials page, under **SSH keys**, click **Add SSH key**.
3. Enter the SSH key nickname, such as the name of the person who owns the key.



3.2 Entering the key

1. Using a text editor, open your SSH public key file, and then copy the contents of the file to the clipboard, ensuring that you don't add any extra lines or spaces to the copied key.

By default, the file name is ~/.ssh/id\_dsa.pub or ~/.ssh/id\_rsa.pub.

#### **If you can't either view or edit the public key file**

If you can't either view or edit your public key file using a text editor, use a command like one of the following to copy the contents of the key file directly to the clipboard:

* + **Mac**

pbcopy < ~/.ssh/id\_dsa.pub

* + **Linux**

sudo apt-get install xclip  
xclip -sel clip < ~/.ssh/id\_dsa.pub

* + **Windows with GitBash**

clip < ~/.ssh/id\_dsa.pub

1. Paste the OpenSSH-formatted key into the **SSH public key** field. OpenSSH public keys start with ssh-dssor ssh-rsa, and appear similar to the following example:

ssh-dss  AAAAG1bB0us3MAAACBALFF6+dpSkO6bwbJ6BCCwbGavQPqR3JSwGWWm1ZCg2  
i43xzDTonY6+PZavGYbgbYgGySDVBbSxKIKSMGUWE8EVHiYzwiUYYaFdTYpkEyqOw/6FlDN  
sVjL+hb454dPgdYOhvjVCI683KrvTP6OMmQTCxInQpeYmyYql7dhhbg4B7AAAAFQDjLv0eP  
hqNrlPyX6j76nxF0dAf3wAAAIB4boChX4eU8YQT0Og023q44f0dlTvJFgKHa6UZDVUBDpw9  
ZsVvkk703HBjFPxDaOJPjurZtMuGIkw8XhA4a8gWj5v9WppY8EQZcmxHoI73czcCJ53WfRr  
qOwM3HHZddoxEcwz0sTdQ3BkG7G1z0ln92raOnFPC0Ju7YCCV82yswAAAIBjGPas8fU7ycf  
T0dMwtQmUetcj+5qUam7imzwNZ9EB29JVLbo90oVSjWJHrGMst2tGEw3VQm+a1o/ICq+nSG  
I9/trLbbEoTISO8MnDi/5UEiPApo4636EKkIahE8QKZlhlqtGZPfp0hDmn1vgKgFkp95em+  
Zb6r1IZJmx+/ORjcg== user@hostname

1. Click **Add key**.

After you add a key, there can be a delay of approximately a minute before you can use the key to connect to your servers or repository.

## GitHub Basics Introduction

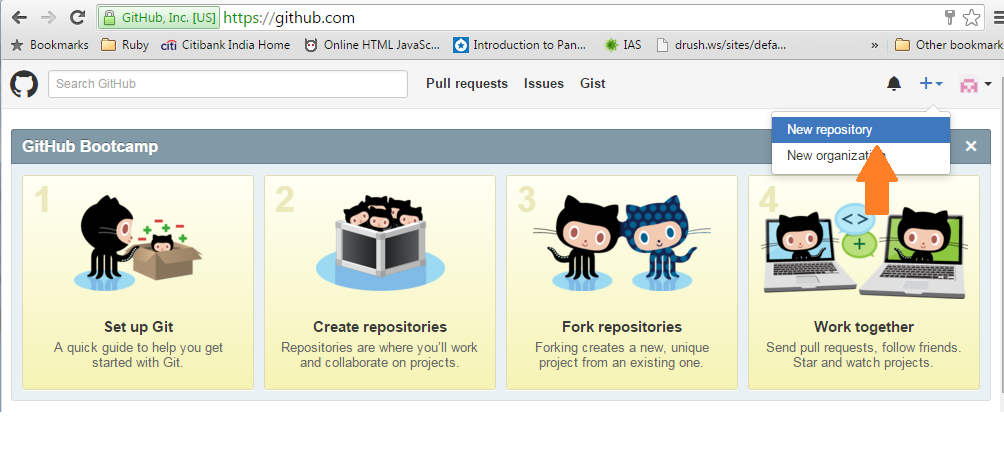
GitHub is a web-based open-source version control system. When developers are creating an application they will make constant changes to the code and release new versions and do couple of releases. Git keeps the code revision straight and store the modifications in a central repository. This allows developers to easily collaborate, as they can download a new version of the software, make changes, and upload the newest revision.

Prerequisites :

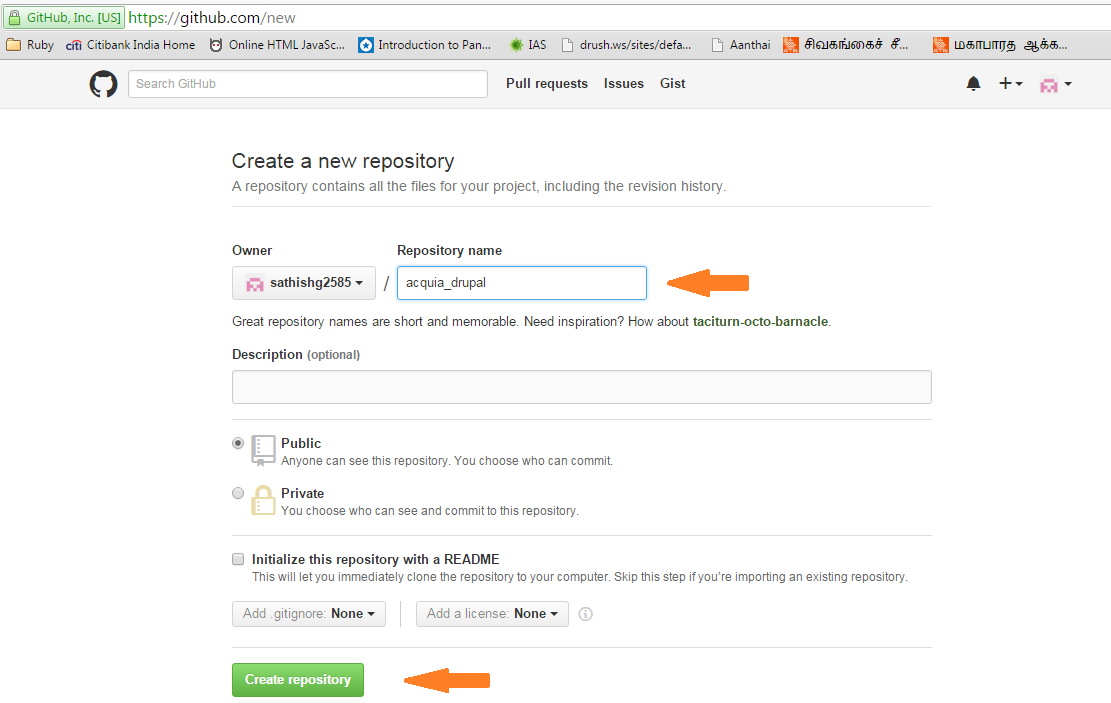
1. Need to have an account in Github.com.
2. Download and Install Git version control software from The Git Development Community <https://git-scm.com/downloads>.

Creating a New GitHub Repository

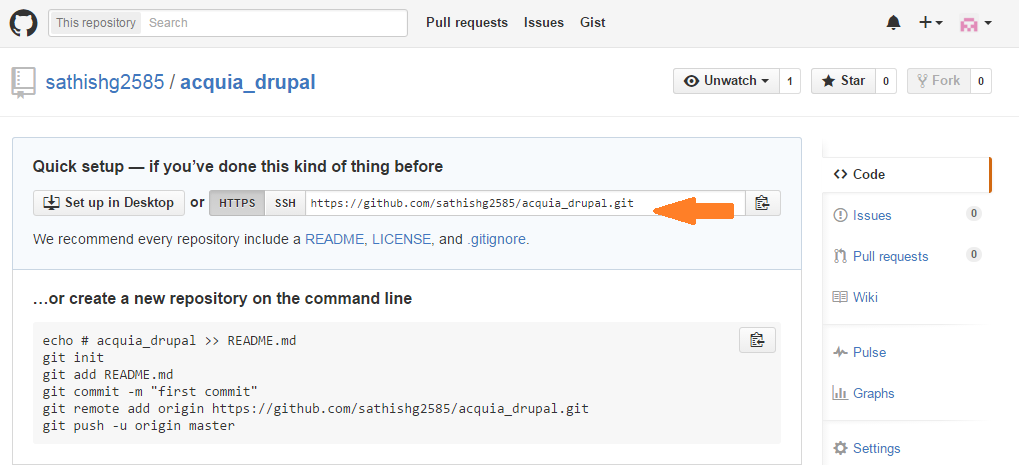
1. Open git bash terminal and go to a path where you want to create a new repository. Below commands will help you to create and push your code to Git.
2. git init -> To initialize the new repo type this command. This will Initialize empty Git repository
3. Copy you drupal project to this newly created Git repository.
4. git add \* -> add all files and folders to the repo
5. git commit -m "YOUR\_COMMENT" -> To make the commit
6. Create a remote repository.



Provide a name to your repository and click on Create Repository



Highlighted is your remote repository URL

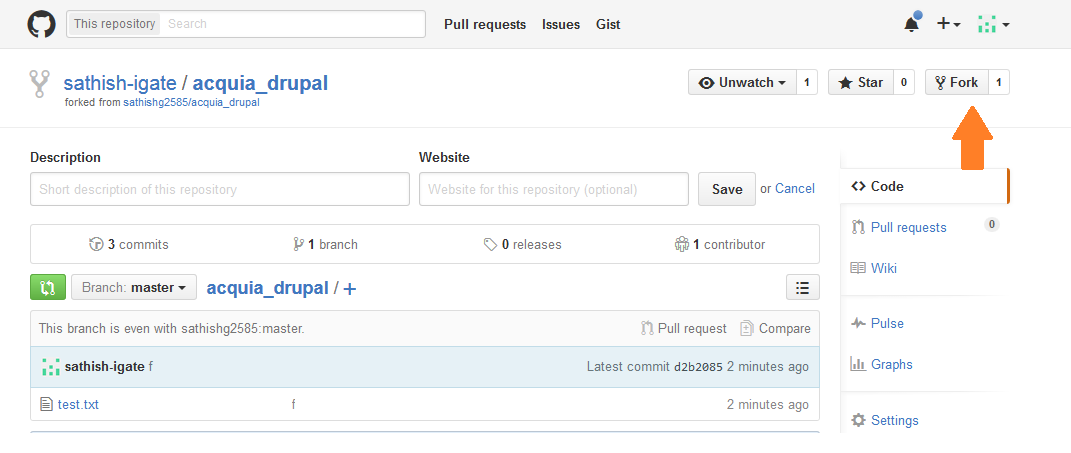


Now you have created the remote repository and you are ready to push your code from local to remote

1. git remote add **origin** **URL-OF-REMOTE-REPO** -> connect local repo to remote repo. origin is the alias name of your repo
2. git push origin **BRANCH\_NAME** -> push your code to a branch of your repo

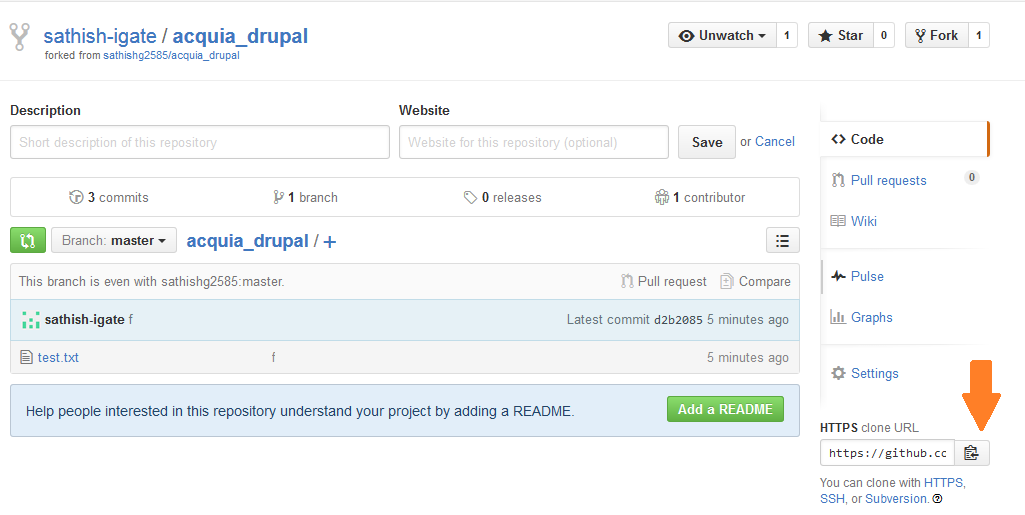
Working in existing repository

1. First fork the existing repository where other developers may be working on it.



Once forked, the repository will be copied under your account for further commits.

1. Clone the repository from your fork for that copy the Clone URL from your account.



1. git clone **FORK\_CLONE\_URL** **FOLDER\_NAME** -> clone the repo to local
2. git remote add **upstream** **MAIN\_REPO\_URL** -> Connect local repo to main remote repo. upstream is the alias for main repository.
3. git fetch **upstream** -> fetch all branches and tags from main remote repo
4. git checkout -b BRANCH\_NAME upstream/BRANCH\_NAME
5. git pull upstream BRANCH\_NAME -> For getting updates from main remote branch before starting your updates
6. git add \* -> add all files and folders to the repo
7. git commit -m "YOUR\_COMMENT" -> To make the commit
8. git push origin BRANCH\_NAME-> push your code to a branch of your repo

Acquia