

DOCUMENTATION

Search the library



▼ Documentation

▼ Acquia Cloud

- [What's new - Acquia Cloud](#)
- [Known Issues](#)
- ▶ [Getting Started](#)
- ▶ [Acquia Cloud Free subscriptions](#)
- ▶ [Acquia Cloud in more depth](#)
- ▶ [How Acquia Cloud works](#)
- ▶ [Configuring Acquia Cloud](#)
- ▶ [Creating a new site](#)
- ▼ **Managing your workflow**
 - [Upgrading Drupal](#)
 - ▼ **Managing your code**
 - ▶ [About your code repository](#)
 - [Configuring your code for local development](#)
 - [Sending updates to your code repository](#)
 - [Installing a Drupal module on Acquia Cloud](#)
 - [Editing code on your server](#)
 - [Live Development workflow](#)
 - [Migrating from SVN to Git](#)
 - [Using GitHub with Acquia Cloud](#)
 - [Managing your files](#)
 - ▶ [Managing your](#)

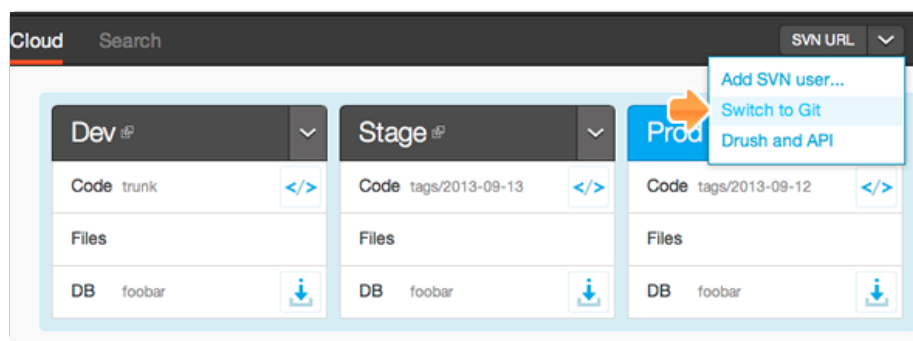
[Documentation](#) » [Acquia Cloud](#) » [Managing your workflow](#)

Managing your code

The Acquia Cloud Workflow page gives you powerful and easy-to-use tools for managing your website's code

Choosing your repository directory structure

Depending on the repository you use, Acquia Cloud creates a different directory structure. You can either choose your version control system on the **Workflow** page or [migrate to a new repository](#).



A Subversion (SVN) code repository

Acquia Cloud creates three top-level directories in your repository when you first register: `trunk`, `branches`, and `tags`. The `/trunk` directory contains two subdirectories: `/docroot` and `/library`. The `docroot` directory is for your Drupal codebase that Acquia Cloud executes on your web server(s) if you select `trunk` as the deployment directory. The `library` directory is in your PHP include path if you need to add PHP extensions. The `/branches` directory is created empty. The `tags` directory is created with a single tag named `WELCOME`, representing the initial state of `trunk`.

```

/trunk
  /docroot
    /sites
    /files
    /modules
    /backups
    /import
    /logs
  /library
  /branches
  /tags

```

where:

- `/trunk` - Contains the website code and files for your development area. In the initial configuration, Acquia Cloud deploys code from this directory to both your production and staging servers.
- `/trunk/docroot` - Stores the development copy of your Drupal codebase.
- `/trunk/library` - Contains the PHP libraries for your website. This directory is in your PHP include path.
- `/branches` - Contains the new development and production branches for your website code. After your

databases

- [Viewing tasks](#)
- [Viewing server statistics](#)
- [Backing up your site](#)
- [Publishing your website](#)
- [Using Cron to schedule tasks](#)
- [Automating with Cloud Hooks](#)
- [Additional environments for Acquia Cloud Enterprise](#)
- ▶ [Developing with the Cloud API](#)
- ▶ [Improving website performance](#)
- ▶ [Security and compliance](#)
- [Videos](#)
- ▶ [Acquia Network](#)
- ▶ [Acquia Cloud Site Factory](#)
- ▶ [Acquia Lift](#)
- ▶ [Acquia Dev Desktop](#)
- ▶ [Acquia Dev Desktop 2 beta](#)
- ▶ [Acquia Search](#)
- ▶ [Drupal Commons](#)
- ▶ [Acquia Drupal](#)
- [Drupal Gardens](#)
- ▶ [Remote Administration](#)
- ▶ [Support Users Guide](#)
- [What's new in Acquia products](#)

website is live, use `/trunk` for testing since it deployed to your staging server, and create new production branches in the `/branches` directory.

- `/tags` - Contains named copies of revisions representing the state of your code and files at some point in time. In the initial configuration, Acquia Cloud creates a tag that contains the same website code and files as the `trunk` directory.

A Git code repository

Acquia Cloud creates a master branch with three folders when you first register, `acquia-utils`, `docroot`, and `library`. The `acquia-utils` directory contains two files by default, both of which help developers view and manage the logs created by their servers on Acquia Cloud. The `docroot` directory is for your Drupal codebase that Acquia Cloud executes on your web server(s) if you select `master` as the deployment branch. The `library` directory is in your PHP include path if you need to add PHP extensions.

```
/acquia-utils
/docroot
/library
```

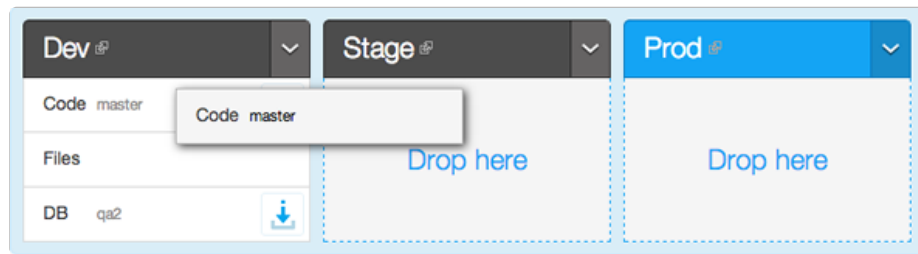
where:

- `/acquia-utils` - Contains the website code and files for your development area. In the initial configuration, Acquia Cloud deploys code from this directory to both your production and staging servers.
- `/docroot` - Stores the development copy of your Drupal codebase.
- `/library` - Contains the PHP libraries for your website. This directory is in your PHP include path.

Working with code on the Workflow page

The Code section of the **Workflow** page is a user interface for deploying code from your repository to your website's environments. When you commit a branch to your repository, Acquia Cloud updates the environment running that code branch.

To move your code between environments, drag a code branch from one environment to another environment. Acquia Cloud creates a new tag for the head version of that branch (the most recently committed code version) and then deploys the tag in the environment you selected.

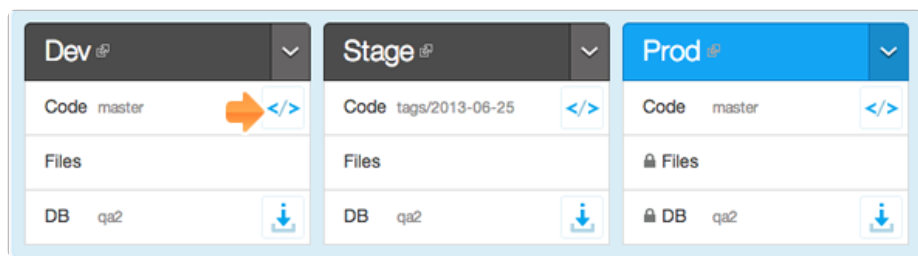


You should only make and test changes to administrative configurations, views, and other major website settings in your development or staging environments. Do not make or test changes in your production environment. If you make a change that breaks your website in the production environment, you might bring your live website down.

Deploying code

To deploy code in an environment:

1. In the code container for the environment, click the code select button.



- In the **Push code** dialog, select a tag from the **Code checkout** menu and click **Deploy**.

When you select a new branch or tag to deploy, that code is retrieved from your repository and deployed to the appropriate environment on the web servers. The deploy task appears in the Workflow page task list.

SVN only: If you want to deploy a non-standard branch or tag, enter the path to the code branch or tag in the **Code path to deploy on [env]** field, and then click **Deploy**. Git branches are well-defined objects in the code repository and are listed in the **Code checkout** menu on the **Push code** dialog.

Moving code from one environment to another

To move code across your environments:

- Check out your repository to your local computer.
- Make changes to your website in the `/trunk` directory, or whatever branch you've configured the Workflow page to deploy for **Dev**. Any environment deploying a branch will deploy all commits to that branch immediately.
- Drag your **Dev** code container to **Stage**, and then click **Deploy**.

When you drag your Dev code container to the Stage environment, Acquia Cloud creates a symbolic link between Stage and the head version (the most recently committed code version) of trunk in your Dev environment. When you click **Deploy**, Acquia Cloud creates a tag and updates the web server to that version of your code. The default tag format is `yyyy-mm-dd.version` (for example, `tags/2013-09-18`). From the example, the next tag you create would be `2013-09-18.0`.

When you have a version that you are ready to deploy to your live production website, drag your code



- To change the code on your servers, you can:
 - Commit changes to the currently deployed branch.
 - Select a different repository branch or tag to deploy.

Reverting code

To revert code to an earlier version:

- In the code container for the environment, click the code select button.
- In the Deploy code dialog, select the tag for the version you want to revert to and click **Deploy**.

When you deploy a previous branch or tag, Acquia Cloud retrieves the code snapshot from your repository and then deploys it on the appropriate environment on the web servers.

Managing your code with other tools

If you don't want to use the code management workflow tool included with Acquia Cloud, you can use the tool of your choice to manage your Acquia Cloud code repositories.

To deploy code across environments using the command line, use the following appropriate command:

For SVN:

```
svn copy trunk tags/2011-03-18
svn commit -m "Created new tag"
```

For Git:

```
git clone trunk tags/2011-03-18  
git commit -a -m "Created new tag"
```

Related topics

- [About your code repository](#)
- [Using Live Development](#)

WAS THIS HELPFUL?  

Status

Forums

Acquia.com

[◀ Upgrading Drupal](#)

[About your code repository ▶](#)