Guide to Git - Deployment

Overview

The Git™ Version Control feature allows you to deploy your cPanel-managed repositories. Generally, deployment sends finished code into production. You can use different configurations to automatically (push deployment) or manually (pull deployment) deploy changes.

- For example, you could use deployment to make changes to your website locally. Then, automatically send them to a directory on your cPanel account.
- For more information about how to deploy changes, read our Git Version Control documentation.
- For more information about how to troubleshoot problems with this feature, read our Guide to Git - For System Administrators documenta tion.

Requirements

Before deployment, repositories **must** meet the following requirements:

- A valid checked-in `.cpanel.yml` file in the top-level directory.
- One or more local or remote branches.
- A clean working tree.

If a repository does **not** meet these requirements, the system will **not** display deployment information. Also, it will disable deployment functionality.

The .cpanel.yml file

The `.cpanel.yml` file determines how and where the changed files deploy. You **must** check a `.cpanel.yml` file in to the top-level directory for each cPanel-managed repository that you deploy. `.cpanel.yml` files **must** use the format in the examples below.

Important:

- The files below are only **examples**. You **must** update them to suit your needs. These files will **not** allow you to deploy a repository successfully.
- **Don't** use a wildcard character, such as an asterisk (`*`), to deploy all files. This could deploy items like the `.git` directory and cause serious problems.
- **Don't** use characters that are invalid in YAML files. For more information, read the Escaped Characters section of yaml.org's YAML Specification.

Deploy individual filesDeploy an entire directory

The following `.cpanel.yml` file deploys the `index.html` and `style.css` files to the `example` account's `public_html` directory:

```
---
deployment:
  tasks:
    - export DEPLOYPATH=/home/user/public_html/
    - /bin/cp index.html $DEPLOYPATH
    - /bin/cp style.css $DEPLOYPATH
```

- Line 1 signifies the beginning of a YAML file.
- Lines 2 and 3 add the `deployment` and `tasks` keys, respectively.
- Lines 4 through 6 specify an array of BASH commands to run during deployment. You can add as many commands to this array as you wish.

The following `.cpanel.yml` file deploys all files in the `images` directory to that directory in the `example` account's `public_html` directory:

```
---
deployment:
  tasks:
    - export DEPLOYPATH=/home/user/public_html/
    - /bin/cp -R images $DEPLOYPATH
```

- Line 1 signifies the beginning of a YAML file.
- Lines 2 and 3 add the `deployment` and `tasks` keys, respectively.
- Lines 4 and 5 specify an array of BASH commands to run during deployment. You can add as many commands to this array as you wish.

Note:
To add comments to this file, add a line that begins with the hash character (#).

Automatic or push deployment

Important:
cPanel's *Git Version Control* feature (*cPanel >> Home >> Files >> Git Version Control*) automatically adds a `post-receive` hook to all cPanel-managed repositories.

- When you push changes **directly** to a cPanel-managed repository that includes a `.cpanel.yml` file, the hook deploys those changes **automatically**.
- For more information, read Git's githooks documentation.

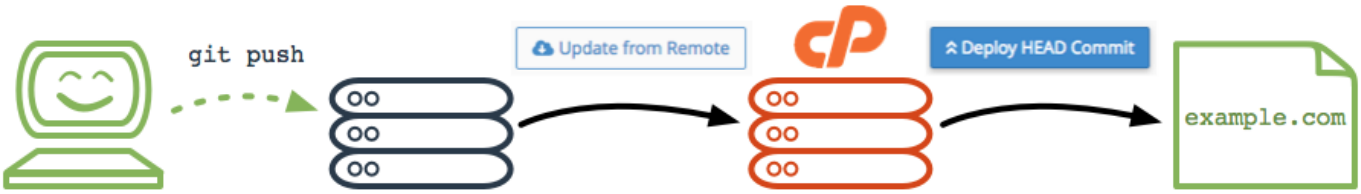
Automatic push deployment of a website.

With push deployment, a single `git push` command sends changes from your local computer to your cPanel-managed repository. The system then automatically runs the commands in your `.cpanel.yml` file. This configuration will send changes from the cPanel-managed repository to a

production directory. (For example, to the directory that contains your website's public files.)

> **Note:**
> You can use manual deployment to deploy your repository again without new changes.

Manual or pull deployment



Manual pull deployment of a website with a remote repository.

With pull deployment, the `git push` command sends changes from your local computer to a remote repository. When you click *Update from Remote* in the *Pull or Deploy* tab of the *Manage* section of the *Git Version Control* interface (*cPanel >> Home >> Files >> Git Version Control*), the system retrieves changes from the remote repository and applies them to the cPanel-managed repository. When you click *Deploy HEAD Commit*, the system runs the commands in your `.cpanel.yml` file to send changes from the cPanel-managed repository to a production directory. (For example, to the directory that contains your website's public files.)

Additional documentation

Suggested documentationFor cPanel usersFor WHM usersFor developers

- Guide to Git - Deployment
- Guide to Git
- Guide to Git - Common Git Commands
- Guide to Git - Host Git Repositories on a cPanel Account
- Guide to Git - Git Terms

Error rendering macro 'contentbylabel' : parameters should not be empty

Error rendering macro 'contentbylabel' : parameters should not be empty

Content by label

There is no content with the specified labels