

The Render CLI

Manage your Render resources from the command line.

Use the Render CLI to manage your Render services and datastores directly from your terminal:

```
Services > app-frontend (My App - Production) > Restart

Are you sure you want to restart resource app-frontend?

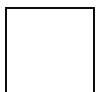
Yes (y) No (n)

Navigation: [Ctrl+C]: Quit [Ctrl+D]: Back [Ctrl+S]: Copy command to clipboard
```

Among many other capabilities, the CLI supports:

- Triggering service deploys, restarts, and one-off jobs
- Opening a psql session to your database
- Viewing and filtering live service logs

The CLI also supports non-interactive use in scripts and CI/CD.



Please submit bugs and feature requests on the CLI's [public GitHub repository](#).

Setup

+ Did you install the Render CLI before November 2024?

1. Install

Homebrew Linux/macOS Direct download Build from source

Run the following commands:

```
$ brew update
$ brew install render
```

SHELL

After installation completes, open a new terminal tab and run `render` with no arguments to confirm.

2. Log in

The Render CLI uses a **CLI token** to authenticate with the Render platform. Generate a token with the following steps:

- 1 Run the following command:

```
$ render login
```

SHELL

Your browser opens a confirmation page in the Render Dashboard.

2 Click **Generate token**.

The CLI saves the generated token to its local configuration file.

3 When you see the success message in your browser, close the tab and return to your terminal.

4 The CLI prompts you to set your active workspace.

You can switch workspaces at any time with `render workspace set`.

You're ready to go!

Common commands

This is not an exhaustive list of commands.

- Run `render` with no arguments for a list of all available commands.
- Run `render help <command>` for details about a specific command.

COMMAND	DESCRIPTION
<code>login</code>	<p>Opens your browser to authorize the Render CLI for your account. Authorizing generates a CLI token that's saved locally.</p> <p>If the CLI already has a valid CLI token or <u>API key</u>, this command instead exits with a zero status.</p>
<code>workspace set</code>	<p>Sets the CLI's active workspace. CLI commands always operate on the active workspace.</p>
<code>services</code>	<p>Lists all services and datastores in the active workspace. Select a service to perform actions like deploying, viewing logs, or opening an SSH/psql session.</p>

COMMAND	DESCRIPTION
<code>deploys list</code> <code>[SERVICE_ID]</code>	<p>Lists deploys for the specified service. Select a deploy to view its logs or open its details in the Render Dashboard.</p> <p>If you don't provide a service ID in interactive mode, the CLI prompts you to select a service.</p>
<code>deploys create</code> <code>[SERVICE_ID]</code>	<p>Triggers a deploy for the specified service.</p> <p>If you don't provide a service ID in interactive mode, the CLI prompts you to select a service.</p> <p>In <u>non-interactive mode</u>, helpful options include:</p> <ul style="list-style-type: none"> ▪ <code>--wait</code> to block until the deploy completes (a failed deploy exits with a non-zero status) ▪ <code>--commit [SHA]</code> to deploy a specific commit (Git-backed services only) ▪ <code>--image [URL]</code> to deploy a specific Docker image tag or digest (image-backed services only)
<code>psql</code> <code>[DATABASE_ID]</code>	<p>Opens a psql session to the specified PostgreSQL database.</p> <p>If you don't provide a database ID in interactive mode, the CLI prompts you to select a database.</p>
<code>ssh</code> <code>[SERVICE_ID]</code>	<p>Opens an SSH session to a running instance of the specified service.</p> <p>If you don't provide a service ID in interactive mode, the CLI prompts you to select a service.</p>

Non-interactive mode

By default, the Render CLI uses interactive, menu-based navigation. This default is great for manual use, but not for scripting or automation.

Configure the CLI for non-interactive use in CI/CD and other automated environments with the following steps:

1. Authenticate via API key

The Render CLI can authenticate using an API key instead of `render login`. Unlike CLI tokens, API keys do not periodically expire. For security, use this authentication method only for automated environments.

- 1 Generate an API key with [these steps](#).
- 2 In your automation's environment, set the `RENDER_API_KEY` environment variable to your API key:

```
export RENDER_API_KEY=rnd_RUExp...
```

BASH

If you provide an API key this way, it always takes precedence over CLI tokens you generate with `render login`.

2. Set non-interactive command options

Set the following options for *all* commands you run in non-interactive mode:

FLAG	DESCRIPTION
<code>-o / --output</code>	Sets the output format. For automated environments, specify <code>json</code> or <code>yaml</code> . Also supports <code>text</code> for unstructured text output, along with the default value <code>interactive</code> .
<code>--confirm</code>	Skips any confirmation prompts that the command would otherwise display.

For example, to list the active workspace's services in JSON format:

```
$ render services --output json --confirm
```

SHELL

Example: GitHub Actions

This example action provides similar functionality to Render's [automatic Git deploys](#). You could disable auto-deploys and customize this action to trigger deploys with different conditions.

To use this action, first set the following secrets in your repository:

SECRET	DESCRIPTION
<code>RENDER_API_KEY</code>	A valid Render API key .
<code>RENDER_SERVICE_ID</code>	The ID of the service you want to deploy

YAML

```
name: Render CLI Deploy
run-name: Deploying via Render CLI
# Run this workflow when code is pushed to the main branch.
on:
  push:
    branches:
      - main
jobs:
  Deploy-Render:
    runs-on: ubuntu-latest
    steps:
      # Downloads the Render CLI binary and adds it to the PATH.
      # To prevent breaking changes in CI/CD, we pin to a
      # specific CLI version (in this case 1.1.0).
      - name: Install Render CLI
        run: |
          curl -L https://github.com/render-oss/cli/releases/download
          unzip render.zip
          sudo mv cli_v1.1.0 /usr/local/bin/render
      - name: Trigger deploy with Render CLI
        env:
          # The CLI can authenticate via a Render API key without log
          RENDER_API_KEY: ${ secrets.RENDER_API_KEY }
          CI: true
```

```
run: |  
  render deploys create ${{ secrets.RENDER_SERVICE_ID }} --ou
```

Local config

By default, the Render CLI stores its local configuration at the following path:

```
$HOME/.render/cli.yaml
```

TEXT

You can change this file path by setting the `RENDER_CLI_CONFIG_PATH` environment variable.

Managing CLI tokens

For security, CLI tokens periodically expire. If you don't use the Render CLI for a while, you might need to re-authenticate with `render login`.

View a list of your active CLI tokens from your [Account Settings page](#) in the Render Dashboard. You can manually revoke a CLI token that you no longer need or that might be compromised. Expired and revoked tokens do not appear in the list.



X LinkedIn GitHub

© Render 2025

