Cloud Foundry CLI Cheat Sheet

ALTOROS

About Cloud Foundry

Cloud Foundry is an app-centric cloud-native platform that aims to automate and streamline the life cycle of cloud applications. It is highly extendable, works with most popular programming languages, and runs on private and public cloud infrastructures. Besides the open source version, there are several commercial distributions, e.g., IBM Bluemix, HP Helion Stackato, GE Predix, and Pivotal CF. The documentation can be found at: http://docs.cloudfoundry.org

The basics: multi-tenancy

Cloud Foundry supports multi-tenancy with fine-grained user-based access control and resource quotas. Development accounts are called **organizations** (or orgs). Each org has one or more **spaces**, to which apps and services are scoped. Collaborators access orgs through **user accounts** and can have different roles in different orgs and spaces. All users in an org share a resource quota plan, applications, services availability, and custom domains.

Installing the Cloud Foundry CLI

Whether you are using open source Cloud Foundry in a private datacenter or a managed proprietary version, you will most likely need the Cloud Foundry CLI (Command Line Interface) to interact with the PaaS and applications. The CF CLI is written in Go and is easy to install on most platforms. Links to installers, binaries, and instructions for Mac, Linux, and Windows can be found at: https://paas.ly/cli-install

Getting started

After getting credentials to a Cloud Foundry deployment, you need to target your CLI and login. The following commands will help you with these steps.

`cf help <cf-cli-command>` Displays CF CLI help with all the available commands. Specifying a command shows its help entry and options.

`cf api <target-cf-api-url>` Sets or shows the target API endpoint URL.

`cf login` Logs a user in. Shows prompts for username and password. It is possible to pass your API URL, username, password, org, and space as arguments, although it is not encouraged for security reasons. Alias: `l`

`cf target -o <org> -s <space>` Sets or shows the targeted org or space.

`cf logout` Logs a user out. Alias: `lo`

Managing applications

Next, are the most frequently used app management commands.

`cf apps` Lists all applications in the current space. Alias: `a`

`cf app <app-name>` Displays information, such as health and status, for the specified app.

`cf push <app-name> -s <stack> -b <buildpack-name>` Pushes a new app or syncs changes to an existing app with or without a manifest. It it is also possible to simultaneously push several apps with a manifest:

```
cf push -f /path/to/manifest.yml
```

You are highly encouraged to view `cf help push` for the full list of options.

`cf scale <app-name> -i <instances> -k <disk> -m <memory>` Changes or displays the instance count, disk space limit, and memory limit for an app.

`cf delete <app-name>` Deletes the specified application. The `-r` option also deletes any mapped routes and `-f` forces deletion without confirmation.

`cf start <app-name>` Starts an app. Alias: `st`

`cf stop <app-name>` Stops an app. Alias: `sp`

`cf restart <app-name>` Restarts an app (the application process). Alias: `rs`

`cf restage <app-name>` Re-builds application sources and re-runs the application in a new container. Alias: `rg`

Environment variables and logs

Use these commands to set/delete environment variables and view or tail application logs.

`cf env <app-name>` Shows all environment variables for an app. Alias: `e`

`cf set-env <app-name> <variable-name>` Sets an environment variable for the specified app. Alias: `se`

`cf unset-env <app-name> <variable-name>` Deletes an environment variable.

`cf logs <app-name>` Tails logs for the specified app. Use the `--recent` option to dump recent app logs instead.

`cf events <app-name>` Shows recent app events.

Services

Cloud Foundry **services** provide applications with data storage and other resources, often referred to as **service instances**. The relationship between Cloud Foundry services and service instances is somewhat similar to the relationship between RDBMS and the databases they manage. This section contains basic commands for creating CF service instances and binding them to apps.

`cf marketplace` Lists all services available in the marketplace. Alias: `m`

`cf services` Lists service instances in the current space. Alias: `s`

`cf service <service-name>` Shows information about a service instance.

`cf create-service <service-name> <plan> <service-instance> -c <parameters-json>` Creates a service instance. Use the `-c` option to pass additional configuration parameters as a JSON object. Example:

```
cf create-service database
altoros-plan mydb -c '{"ram_gb":8}'
```

`cf delete service <service-instance>` Deletes a service instance.

`cf bind-service <app-name> <service-instance>` Binds a service instance to the specified app. Use the `-c` option to pass additional configuration parameters as a JSON object.

`cf unbind-service <app-name> <service-instance>` Unbinds a service instance from an app. Alias: `us`

Routes and domains

Next, are the most frequent commands for managing routes and domains. For more information, see this guide: https://paas.ly/routes-domains

`cf routes` Lists all routes in the current space or organization. Alias: `r`

`cf create-route <space> <domain> -n <hostname>` Creates a URL route in a space. Example for app-host.altoros.com:

```
cf create-route my-space
altoros.com --hostname app-host
```

`cf map-route <app-name> <domain> -n <hostname>` Adds a previously created URL route to an app. Example for app-host.altoros.com:

```
cf map-route app altoros.com
--hostname app-host
```

`cf domains` Lists all domains in the target organization.

`cf create-domain <org> <domain>` Attaches a domain to an organization.

Spaces

Spaces provide users within an org with access to a separate, shared location for building, deploying, and otherwise working with apps. Each app and service is scoped to a certain space.

`cf spaces` Lists all spaces in the target org.

`cf space <space-name>` Shows available information on a space (apps, domains, services, etc.)

`cf create-space <space-name> -o <org>` Creates a new space with default quota in the specified organization.

`cf delete space <space-name>` Deletes a space.

`cf rename-space <old-space-name> <new-space-name>` Renames a space.