Cloud Controller API

Goal

Use cf curl, the Cloud Controller API, cf CLI tracing and globally unique identifiers (guids).

Approximate time: 10 minutes

Exercises

Make sample Cloud Controller API calls using cf curl

The Cloud Controller is a component of the Elastic Runtime. It is the primary component that operators and developers interact with when using Cloud Foundry. The cf CLI is a text-based client that makes calls to the Cloud Controller API behind the scenes. Here you will directly make Cloud Controller API calls using `cf curl`.

> **TIP** Using `cf curl` to make Cloud Controller API calls is convenient, because the cf CLI automatically works with the User Account and Authentication (UAA) service to get access tokens. If you ever need to call the Cloud Controller API outside of the context of the cf CLI (such as inside of an app instance), you will need to obtain an access token from the UAA.

1. Browse the Cloud Controller API documentation at http://apidocs.cloudfoundry.org.
2. View the Cloud Controller API documentation for listing applications under Apps > List all Apps.

3. Using the `cf curl` command below, make an /apps API call to view the details of the apps in your Cloud Foundry installation.

```
cf curl v2/apps
```

4. Explore the details of the result. Notice that you are viewing the details of all of the apps in the installation. You never specified any organization or space related to this Cloud Controller API call.
5. Make another Cloud Controller API call to view the details of a single application. Do this by adding a parameter to the query string of the previous call, as shown in the curl statement below. You will need to change `my_sample_app` to a valid app name in your installation.

```
cf curl v2/apps?q=name:my_sample_app
```

6. Verify that you see one result in the output.

Congratulations, you have made Cloud Controller API calls. You can see that the cf CLI is easier to use than the Cloud Controller API because it abstracts away some details for you. Use the cf CLI whenever you can, but there may be occasions where you need to use the Cloud Controller API. For example, this may be needed if you are customizing the Cloud Foundry installation.

Use CLI tracing

cf CLI tracing is easy to use and can provide valuable information when troubleshooting or developing with Cloud Foundry. cf CLI tracing is turned on with the CF_TRACE=true environment variable.

1. Use the cf CLI to view the apps in your development space of your installation. You will **turn on CLI tracing** to see the Cloud Controller API calls that the cf CLI is making behind the scenes.

On MacOS or Linux the following command turns on CLI tracing, then calls cf apps.

```
CF_TRACE=true cf apps
```

On MS Windows this has to be two commands:

```
set CF_TRACE=true
cf apps
```

2. Notice that the CLI is making a Cloud Controller API call behind the scenes. The request is in the REQUEST: block and the response is in the RESPONSE: block. The request looks something like this:

```
REQUEST: [2015-08-30T15:41:16-07:00]
GET /v2/spaces/f38b8e81-1f87-4ce4-9d16-25cdaf0c0e70/summary HTTP/1.1
Host: api.system.sX.edu.pcfdemo.com
Accept: application/json
Authorization: [PRIVATE DATA HIDDEN]
Content-Type: application/json
User-Agent: go-cli 6.12.2-24abed3 / darwin
```

3. You can see that behind the scenes, the `cf apps` call is making a call to the `spaces/[space guid]/summary` Cloud Controller API.
4. View the Cloud Controller API for Spaces > Get Space summary at http://apidocs.cloudfoundry.org. Almost every resource in Cloud Foundry has a globally unique identifier (guid) associated with it. You will often see these in logs. To get a summary of the information related to the apps in a space, the guid for the space must be included in the Cloud Controller API request. The cf CLI has a target already set, so that is how it knows what the guid of the space is.

5. Notice that the cf CLI uses information from the `spaces/[space guid]/summary` Cloud Controller API request to display the nicely formed results for `cf apps`. That information is under the RESPONSE section.
6. The cf CLI knew the guid of the current space. In some circumstances, you will need to make a Cloud Controller API request to determine the guid for a resource. To obtain the guid of a space, you can use the `spaces` Cloud Controller API. This is in the Cloud Controller API documentation under Spaces > List all Spaces. **Make this API call** and verify that the guid for your current space is the same as the guid in REQUEST block when tracing the `cf apps` call.
7. Turn on tracing and run a `cf push` for any simple app. Notice that the cf CLI is making many Cloud Controller API requests behind the scenes.
8. If you have time, you can try out other Cloud Controller API calls.
9. The cf CLI has a plugin interface, allowing you to make custom client-side commands. Brows the cf plugin documentation at https://docs.cloudfoundry.org/devguide/installcf/use-cli-plugins.html. You can explore the community CLI plugins at http://plugins.cloudfoundry.org/ui/.

Congratulations, you have completed this lab. You have used `cf curl` to make Cloud Controller API requests, used cf CLI tracing and used guids in your requests.

Last updated 2015-12-10 22:38:00 PST