

cloud.gov Workshop Cheatsheet

2.1 Select and install the appropriate installer for your computer:

Go to <https://github.com/cloudfoundry/cli/releases> and select an **Installer** for your system. Download and go through the installation steps.

On Macs, with Homebrew, you can use:

```
brew cask install cloudfoundry-cli
```

After the installer has finished, run the command:

```
> cf
```

and you should see a list of command options.

2.2 Login to cloud.gov with the *cf* CLI

You'll enter the command below, and you'll be directed to an *authentication URL*.

```
cf login --sso -a https://api.fr.cloud.gov
```

Confirm you're logged in by seeing the *orgs* you belong to:

```
cf orgs
```

3.1: Download labs

Mac/Linux shell:

```
cd $HOME
curl -Lo cgw.zip http://bit.ly/cgw-zip
unzip cgw.zip
cd cg-workshop-master
```

Windows Powershell:

```
cd $HOME
iwr -o cgw.zip https://bit.ly/cgw-zip
7z x cgz.zip # If no 7zip, use File Explorer to
unpack
cd cg-workshop-master
```

Lab 3.2: Deploy static website

Don't literally use *myfname-lname* below. Use your own name like, *jane-doe*:

```
cf push -f lab03-site/manifest.yml myfname-lname
```

^ If you do use *myfname-lname*, you'll run into a **route conflict** with someone else trying to use that same name. `

Now try accessing your site at

<https://fname-lname.app.cloud.gov>

Lab 4.1: Review the *deployment manifest*

```
more lab04-app/manifest.yml
```

How much memory/disk are we saving compared to defaults of 512Mb RAM and 1024Mb disk quota?

Lab 4.2 Push the application

```
cf push -f lab04-app/manifest.yml cglab
```

Lab 4.3 Review the app status and health

Run:

```
cf app cglab
```

How much memory and disk is it using?

Lab 5.1 Review the available *services*

Run the command below. How many Redis *services* are there?

```
cf marketplace
```

Examine **redis32** service details with the **-s** option.

```
cf marketplace -s redis32
```

Lab 5.2: Create a Redis service with *create-service*

The format for *create-service redis32* is:

```
cf create-service redis32 PLAN NAME
```

Run:

```
cf create-service redis32 micro cglab-redis
```

Wait one minute, then check your service with:

```
cf service cglab-redis
```

Lab 5.3 Associate service and app with *bind-service*

The app, *cglab* needs to know about *cglab-redis*. The *bind-service* shares service information by setting *environment variables* in the app container.

Run:

```
cf bind-service cglab cglab-redis
```

View the environment variables in the app with:

```
cf env cglab
```

What's the first service under `VCAP_SERVICES`?

Lab 5.4 Push the new version of our app

Now we can push the version of the app that uses the data store. Run:

```
cf push cglab -f lab05-state/manifest.yml
```

Has the app's URL changed?

Visit your app at the URL. Refresh page multiple times. What does the app do?

Lab 5.5 Scaling

Since CF stores executable artifacts and runs them in containers, you can quickly *scale* your app to meet demand.

Scale *cglab* to two instances, then immediately, refresh the *cglab* webapp page multiple times

```
cf scale cglab -i 2
```

How long until a new instance was available?

Lab 6.1: View live application logs

View current app activity:

```
cf logs cglab
```

Then interact with your *cglab* webpage. Press **Ctrl-C** to stop log streaming

Do you see any logs from the router? From the app?

Lab 6.2: View historical logs in Kibana

- Visit: <https://logs.fr.cloud.gov>
- Enter **ERR** in the Search box, then search
- Click the ► triangle to expand, then seek *@message*
- What error is our *cglab* application giving?

Lab 6.3: Application Events

Events are generated by CloudFoundry, *about* your application.

View application events. Do you see any CRASH events?

```
cf events cglab
```

Lab 6.4: SSH to debug *cglab*

Connect to your *cglab* application^[^3]

```
cf ssh cglab
```

You'll be connected a Linux container. To see all processes, run the command below. How many processes are running?

```
ps -ef
```

Lab 8.1: Delete apps with *cf delete*

List all your apps with:

```
cf apps
```

Then delete each one, e.g.:

```
cf delete cglab  
cf delete myfname-lname # use the real app name
```

Lab 8.2: Delete services

List all your services with:

```
cf services
```

Then delete each one, e.g.:

```
cf delete-service cglab-redis
```

Lab 8.3: Delete unused routes

CloudFoundry automatically creates *routes* for your web application. List your routes with:

```
cf routes
```

Routes that no longer connect to apps are *orphaned*. Clean them all up with:

```
cf delete-orphaned-routes
```