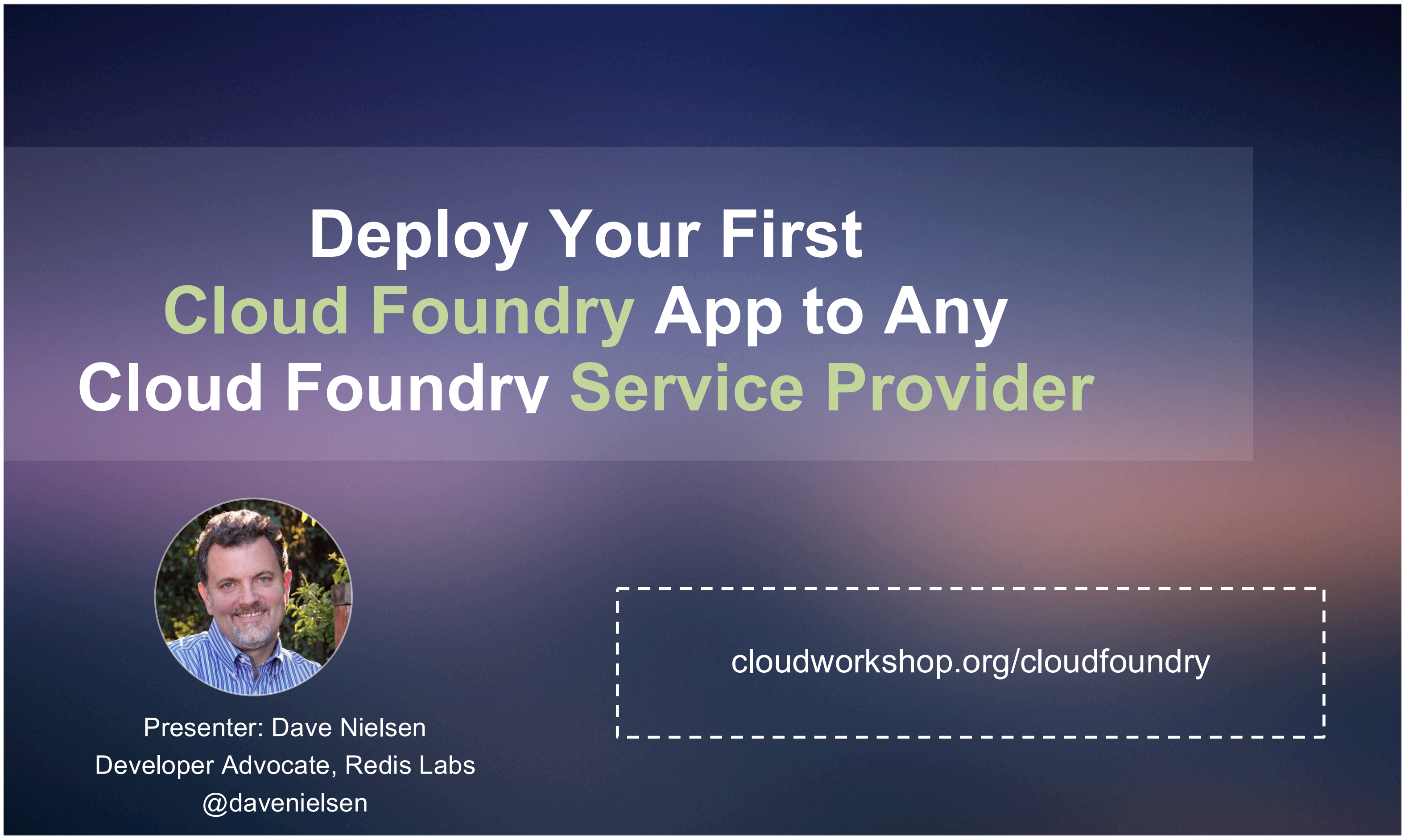

Deploy Your First Cloud Foundry App to Any Cloud Foundry Service Provider

Presenter: Dave Nielsen
Developer Advocate, Redis Labs
@davenielsen

cloudworkshop.org/cloudfoundry

CLOUD FOUNDRY™

Oct 2015

Table of Contents:

1. About This Workshop:

This workshop will teach anyone with basic IT understanding how to deploy his or her first Cloud Foundry Web app.

2. Target Audience:

Other than installing software and typing commands in a terminal, we don't use any advanced tools, so IT pros, developers, architects, and project managers should be able to complete this workshop.

3. Pre-requisites:

For this workshop, you will need:

- Your own Windows, Mac, or Linux computer with internet access
- A Cloud Foundry Service Provider (CFSP): Pivotal Web Services or IBM Bluemix. Other Cloud Foundry service providers may be used, but these instructions have not been tested with them:
    - Pivotal Web Services: run.pivotal.io (30-day trial)
    - IBM Bluemix: bluemix.net (30-day trial)
    - Stackato (by ActiveState): activestate.com/stackato/sandbox (30 day trial)
    - HP Helion Dev Platform (coming soon)
    - CenturyLink (coming soon)
- An e-mail address. Like many free Cloud Computing services, Cloud Foundry service providers require that you have a valid e-mail.

Note: Pivotal Web Services also requires that you provide an SMS-enabled phone number, to verify that you are not a bot.

Exercise 1: Deploy and Scale the "Hello Cloud" app

1. Prepare your application:

   a. Download (or create) your first application
      - You can download the "Hello Cloud" app created for this exercise from http://cloudworkshop.org/cloudfoundry
      - Unzip the "hellocloud.zip" archive into any directory you want (such as *c:/workshop* (or *~/workshop*).
      - Navigate to the new hellocloud app folder (such as *c:/workshop/hellocloud* (or *~/workshop/hellocloud*).
      - Modify the text in the *manifest.yml* file as required to deploy your app.
          - Change host: property from *cf-node-hellocloud-dcn* to a unique hostname with your initials. For example, you might change the last three letters to that of your own initials, such as 'xyz' (such as *cf-node-hellocloud-xyz)*.

Quiz: What happens if you run this app code on a server that is not Cloud Foundry?

2. Create your account:

   a. Sign-up for your free trial:

      - Pivotal Web Services, 60-day trial: http://run.pivotal.io
      - IBM Bluemix, 30-day trial: http://bluemix.net

   b. Check your e-mail. Click the link to verify your e-mail address. Pivotal will also require that you confirm your mobile phone number via SMS text message.

   c. Create an org, e.g., *mydevteam*.
      Note: Bluemix will create an org for you based on your e-mail address.

3. Setup your development space:

   a. Download the Cloud Foundry CLI:
      - Visit http://github.com/cloudfoundry/cli
      - Scroll down to the "Downloads" section of the github page
      - Download the appropriate (stable) installer

   b. Install the Cloud Foundry CLI:

      - For Windows:

        - Unzip the installer-windows.zip file you downloaded above.
        - Run the cf_installer.exe that was unzipped by the previous step
        - This will create a folder: C:\Program Files (x86)\CloudFoundry
        - Add the C:\Program Files (x86)\CloudFoundry to your %PATH% variable
        - Open the Command app (cmd.exe)
        - Type the following command to make sure the CLI was installed correctly:

        ```
        > cf -help
        ```

      - For Mac:

        - Open the Terminal application.
        - Type the following command to make sure the CLI was installed correctly:

        ```
        $ cf -help
        ```

   c. Run the following command to log into your Cloud Foundry service provider account:

      ```
      > cf login -a https://api.run.pivotal.io
      ```
      or

      ```
      > cf login -a https://api.ng.bluemix.net
      ```

      When prompted, enter the e-mail address and password you used to create your Cloud Foundry Service Provider account.

   d. Run the following commands to see details about your org and space:

```
> cf apps
> cf orgs
> cf spaces
> cf target -o [org] -s [space]
   example: > cf target -o mydevteam -s development
```

e. Push your app (run this command from within your hellocloud app folder):

```
> cf push
```

f. Run the ‘apps’ command again and see your new app listed:

```
> cf apps
```

g. Run the ‘app’ command to see health and status details about a specific app:

```
> cf app [app name]
  example: > cf app hellocloud
```

h. Run the ‘scale’ command to increase the number of instances running your new app:

```
> cf scale [app name] -i [# of instances]
  example: > cf scale hellocloud –i 2
```

i. Open your Web browser to the URL provided by the output of the *cf push* process. It should look something like this:

- For Pivotal Web Services: `cf-node-hellocloud-<your initials>.cfapps.io`
- For IBM Bluemix: `cf-node-hellocloud-<your initials>.mybluemix.net`

j. Open the Web console and view the dashboard for your app:

- See your app listed within the *developer* space dashboard.
- Click on your app to see the app’s dashboard.
- Click the Plus (+) button two more times to add a couple more instances to your app.
- View your app in the Web browser again to see the new instances in use. You may have to refresh the Web browser a few times.

k. Go back to your terminal and shut down your app:

- Reduce the number of app instances to one:

```
> cf scale [app name] -i 1
```

- Then stop your instance, so you don’t use up any more of your free trial hours:

```
> cf stop [app name]
> cf apps
```

- Finally, delete your instance, so you don’t take up any space at all

```
> cf delete [app name]
> cf apps
```

l. View the logs and events you have generated so far:

- To view Events:

```
> cf events [app name]
```

- To view Logs:

```
> cf logs [app name]
```

m. View other information of interest, such as buildpacks and quotas:

```
> cf buildpacks
> cf quotas
```

Exercise 2: Deploy Contact Form and a Database Service

1. Prepare your application:

a. Download (or create) your application (http://www.cloudworkshop.org/cloudfoundry/)

- Unzip it into *c:/workshop/contactform* (or *~/workshop/contactform*).
- Navigate into this new app folder.
- Modify the source code in the *manifest.yml* file.
    - o Change host property from cf-node-*contactform* to a unique hostname. For example, you might add your own initials to *contactform* to create a unique hostname, such as *contactform-dcn*.

b. (Optional) Setup a runtime environment on your computer and test your app:

- Install Node.js: http://nodejs.org/download
- Install MySQL: http://dev.mysql.com/downloads/
- Install MySQL Workbench: http://dev.mysql.com/downloads/workbench/

c. Push your app:

```
> cf push
```

d. Test your app:

- Open your Web browser to the URL provided by the output of the '*push*' command. It should look something like this:

    - o For Pivotal Web Services: *cf-node-contactform-dcn.cfapps.io*
    - o For IBM Bluemix: *cf-node-contactform-dcn.mybluemix.net*

- Enter any name, e-mail, and message into the form fields. Press Enter to submit the Web form. You will see an error message. Can you guess why? It's because you don't have a database setup for your app yet.

2. Add a database service to your application:

   a. View the list of marketplace services provided by your CFSP and add the ClearDB service. ClearDB is a highly available MySQL service which will be used by this contact form app.

      - Command line:

        ```
        > cf marketplace
        > cf marketplace –s cleardb
        ```

      - Web console:

        Click Marketplace on the Web console dashboard.

   b. Add a database service to your app. There are three ways of consuming service instance credentials within your application: Auto-configuration, CFRuntime, and manual. We will use the manual process.

      - Use the Web Console to bind a ClearDB service to your app:

        - In your web browser, select the “Development” space on the menu on the left side of the screen.
        - Select “Add Service.” Browse the different services available. Some of these may be hosted by your CFSP, others may be hosted outside of your CFSP.
        - Select “ClearDB MySQL Database.”
        - Click “Select this plan” to select the **free** plan.
        - Enter an Instance Name, e.g., *cf-node-contactform-mysql-dcn*.
        - Bind the service to the app:

          - Select your app, e.g., *cf-node-contactform-dcn*.
          - Click “Add.”

      - Use the command line:

        ```
        > cf start cf-node-contactform
        > cf create-service cleardb spark cleardb-mine
        > cf bind-service cf-node-contactform clear-db-mine
        ```

   c. Find the service variables:

      - In your terminal window, type:

        ```
        > cf env [app name]
        example: > cf env cf-node-contactform
        ```

      - Look for the following ClearDB properties:

        - *hostname:*
        - *username:*
        - *password:*
        - *name:*

- Find and replace these property values in the *server.js* file and replace them with the environmental variables from above:

    - *hostname -> host*
    - *username -> user*
    - *password -> password*
    - *name -> database*

- Use MySQL Workbench or any other tool you are familiar with to connect to your ClearDB database service. Then create a database table called '*contact'* with the following fields:

    - name varchar(50)
    - email varchar(50)
    - message text(50)

3. Re-open your Web browser to the URL provided by the output of the *cf push* process. Use cf app [app name] to view the URL again. Enter your name, e-mail, and message into the contact form. Press Enter to submit the form.

You will no longer see an error message because your database now works.

Exercise 3: Registration Form with e-mail verification

Coming soon

Exercise 4: Registration Form with SMS verification

Coming soon

For more information visit http://cloudworkshop.org/cloudfoundry