

Docker OPP Demonstration App Deployment Guide

Overview

Demonstration of integrating an old legacy application with Okta. It is setup to run all necessary components within docker containers- allowing the entire codebase to be portable.

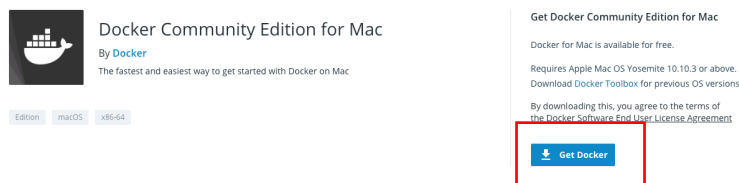
- This application, while running in docker, has currently only been tested while running on a MacOS host

Install Docker for Mac

If you don't already have Docker running on your Mac, download and install Docker. For official instructions on installation see: <https://docs.docker.com/docker-for-mac/install/>

For the purposes of this lab, I installed Docker Community Edition for Mac.

1. Navigate to <https://store.docker.com/editions/community/docker-ce-desktop-mac>



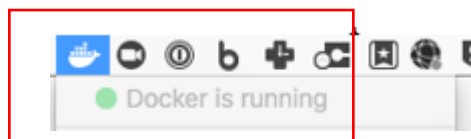
2. Click the **Get Docker** button to download
3. Open the **docker.dmg** file



4. Drag & Drop as instructed
5. Navigate to your Applications directory
6. Double click to launch Docker

Verify Docker install

7. Verify that there is a whale icon in the top status bar which indicates that Docker is running, and accessible from a terminal.



- Optionally, you can open a terminal window and check the version of docker, docker-compose, and docker-machine using the following commands.

```
$ docker --version
```

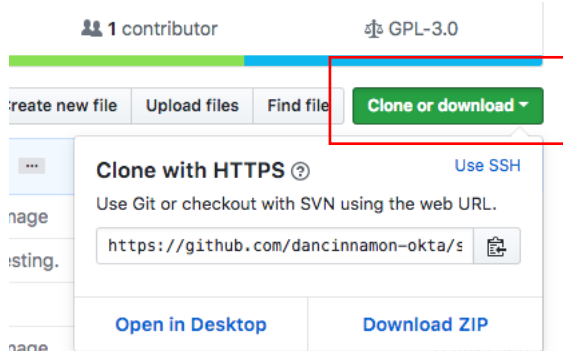
```
$ docker-compose --version
```

```
$ docker-machine --version
```

Download the SWA Demo zip file from GitHub

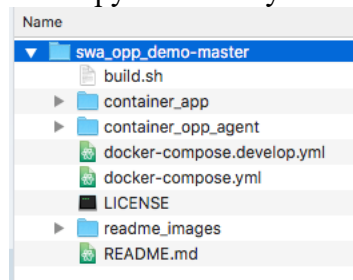
- Navigate to: - https://github.com/dancinnamon-okta/swa_opp_demo

- Click the **Clone or Download** button



- Click **Download ZIP**

- Copy the file to your working directory (e.g. Documents/) and extract it.



Create a “Catalog Admin” Group

The application is looking for a group called "Catalog Admin". If a user is a member of this group, then they can view the admin pages.

- Create a Directory group with the following values:

Field	Value
-------	-------

Name	Catalog Admin
Group Description	Catalog Admins with User View Access

Add Group

Add groups so you can quickly perform actions across large sets of people.

Name

Group Description

14. Click **Add Group**

Create User Accounts

15. Create a person with the following values:

Field	Value
First name	Tim
Last name	Taylor
Username	tim.taylor@oktaice.com
Primary email	tim.taylor@oktaice.com
Groups	Catalog Admin
Password	Set by admin
Enter Password	Tra!nme4321
User must change password at first login	Unchecked

Add Person

First name

Tim

Last name

Taylor

Username

tim.taylor@strenoline.com

Primary email

tim.taylor@strenoline.com

Secondary email (optional)

Groups (optional)

Catalog Admin x

Password ⓘ

Set by admin

☐ User must change password on first login

Save

Save and Add Another

Cancel

16. Click **Save and Add Another**

Field	Value
First name	Al
Last name	Borland
Username	al.borland@oktaice.com
Primary email	al.borland @oktaice.com
Groups	<none>
Password	Set by admin
Enter Password	Tra!nme4321
User must change password at first login	Unchecked

17. Click **Save**

Edit User Profile attributes

18. Directory > People > Search **“Tim Taylor”** > select > Profile tab > Edit

19. Add the following values

Field	Value
Primary phone	800-555-1212
Country code	USA

Organization	Binford Tools
Department	Outdoors

20. Repeat the process for “**Al Borland**”.

Create SWA Application

21. Log in to your Okta tenant and create an application with Platform = **Web** and Sign On Method is **Secure Web Authentication (SWA)**

22. Click **Create** and enter the following values:

Field	Value
App Name	Legacy App – Product Catalog
App’s Login Page URL	https://localhost:9000/login
App Type	Internal (Checked)
Who sets the credentials	Administrator sets username and password
Application username	Okta username

23. Click **Finish**

Add Application Custom Attributes and Mappings

24. Click **Profile**

25. For each of the *String* attributes listed below, click **Add Attribute** > Enter the Display Name and Variable Name and then click **Save and Add Another** until all five custom attributes are added

Display Name	Variable Name
Phone Number	phone_number
Department	department
Country	country
Opt In For Marketing	opt_in
Company Name	company_name

Attributes

[+ Add Attribute](#) [Map Attributes](#)

FILTERS	Display Name	Variable Name	Data Type	Attribute Type	
All	Phone Number	phone_number	string	Custom	✎ ✕
Base	Department	department	string	Custom	✎ ✕
Custom	Country	country	string	Custom	✎ ✕
	Opt In For Marketing	opt_in	string	Custom	✎ ✕
	Company Name	company_name	string	Custom	✎ ✕

26. Click **Map Attributes**

27. Click **Okta to Legacy App**

Legacy App – Product Catalog User Profile Mappings

Legacy App – Product Catalog to Okta **Okta to Legacy App – Product Catalog**

Okta User Profile
user

Legacy App – Product Catalog User Profile
appuser


28. Map the following

user.primaryPhone	→	phone_number	string
user.department	→	department	string
user.countryCode	→	country	string
"Yes"	→	opt_in	string
user.organization	→	company_name	string

29. Click **Save Mappings**.

30. Click **Apply Updates Now**

31. Copy the Okta application name (as shown below). You will need this value later for **OKTA_APP_NAME** field in Docker **.env** file.

 **strenolinejohnzebrowskisdemoorg_legacyappproductcatalog_1** [Profile](#) [Mappings](#)

Legacy App – Product Catalog User

Edit Docker Environment Variables

Open the **.env** file in your working directory. NOTE: This file is not visible in Finder, but you should see it if you open the *swa_opp_demo-master* directory in Atom

Field	Value
OKTA_ORG	<The subdomain of your Okta org>
OKTA_ORG_TYPE	Specify either oktapreview or okta >
APP_CODEBASE	https://github.com/dancinnamon-okta/simple_swa_demo.git

API_KEY	< putareallylongvaluehere> e.g. <i>abcdefghijklmnop123456789</i>
OKTA_APP_NAME	< nameofyourappinokta> e.g. <i>strenolinejohnzebworskisdemoorg_legacyappproductcatalog_1</i>

```

1 #####Demo environment#####
2 #Include your organization's instance here in the file, and which git repository
3 #Okta org - provide only the subdomain name here, so if your org is https://your
4 #Org Type - specify either 'okta', or 'oktapreview' here. This is used to compi
5 #App codebase - specify a git repository for pulling the latest webUI to host.
6 OKTA_ORG=strenoline
7 OKTA_ORG_TYPE=okta
8 APP_CODEBASE=https://github.com/dancinamon-okta/simple_swa_demo.git
9 API_KEY=abcdefghijklmnop123456789
10 OKTA_APP_NAME=strenolinejohnzebworskisdemoorg_legacyappproductcatalog_1
11

```

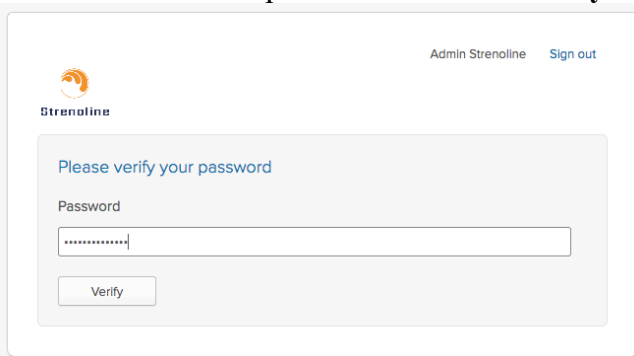
Run build.sh

32. Open up a terminal
33. Navigate to you working directory e.g. *Documents/swa_opp_demo-master/*
34. Run the shell script **.\build.sh**
35. After some amount of time, you will be provided a URL to be used for authentication and registration. (Example shown below) Copy and paste the URL in a browser.

~~Please visit the URL:~~

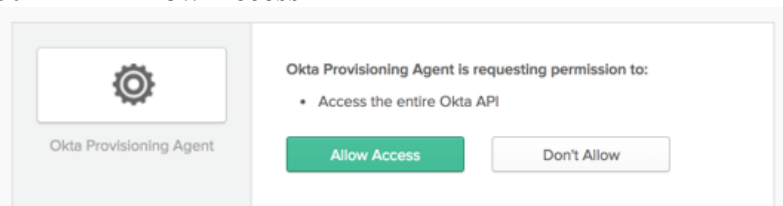
<https://strenoline.okta.com/oauth2/auth?code=cy95pz87> before Wed Nov 07 20:22:46 UTC 2018 to authenticate and continue agent registration

35. Enter the Admin password and click **Verify**



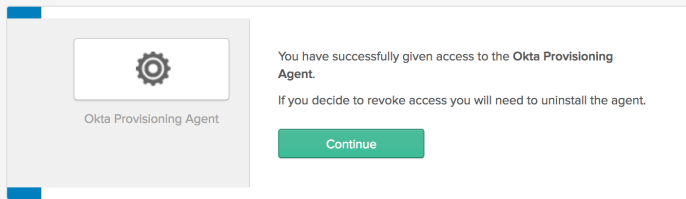
The screenshot shows the Strenoline admin interface. At the top, there is a header with the Strenoline logo and the text 'Admin Strenoline' and 'Sign out'. Below the header, there is a section titled 'Please verify your password'. Inside this section, there is a 'Password' label and a password input field with a masked password '*****'. Below the input field is a 'Verify' button.

36. Click **Allow Access**



The screenshot shows a dialog box for the Okta Provisioning Agent. On the left, there is a gear icon and the text 'Okta Provisioning Agent'. On the right, there is a section titled 'Okta Provisioning Agent is requesting permission to:' followed by a list of permissions: 'Access the entire Okta API'. At the bottom, there are two buttons: 'Allow Access' (green) and 'Don't Allow' (grey).

37. Click **Continue**



38. Check the terminal window. If it is continuing to prompt you to enter the URL, copy and paste the URL a second time and allow access again.

39. You should see messages like the following in the terminal upon completion of the build process

```
Successfully built fb31d65c17d0
Successfully tagged swa_opp_demo-master_opp_agent:latest
```

Verify self-signed certificate creation

40. Verify **swa_opp_demo.cert** and **swa_opp_demo.key** are in your working directory

```
$ ls -la
```

Verify Docker images

41. At a command prompt enter the following

```
$ docker image ls
```

You should have three images as a result of the build process

- **swa_opp_demo-master_opp_agent**
- **swa_opp_demo-master_demowebapp**
- **centos**

42. At a command prompt enter the following

```
$ docker container ls -a
```

At this point, you should not have any containers

Bring the containers up

43. At the terminal run the following command

```
$ docker-compose up
```



```
swa_opp_demo-master — docker-compose - docker-c
demowebapp_1      Uninstalling Django-2.1.3:
demowebapp_1      Successfully uninstalled Django-2.1.3
demowebapp_1      Successfully installed django-2.1.2
demowebapp_1      You are using pip version 9.0.1, however version 18.1 is available.
demowebapp_1      You should consider upgrading via the 'pip install --upgrade pip' command.
demowebapp_1      Operations to perform:
demowebapp_1      Apply all migrations: auth, contenttypes, sessions, swa_app
demowebapp_1      Running migrations:
demowebapp_1      Applying contenttypes.0001_initial... OK
demowebapp_1      Applying contenttypes.0002_remove_content_type_name... OK
demowebapp_1      Applying auth.0001_initial... OK
demowebapp_1      Applying auth.0002_alter_permission_name_max_length... OK
demowebapp_1      Applying auth.0003_alter_user_email_max_length... OK
demowebapp_1      Applying auth.0004_alter_user_username_opts... OK
demowebapp_1      Applying auth.0005_alter_user_last_login_null... OK
demowebapp_1      Applying auth.0006_require_contenttypes_0002... OK
demowebapp_1      Applying auth.0007_alter_validators_add_error_messages... OK
demowebapp_1      Applying auth.0008_alter_user_username_max_length... OK
demowebapp_1      Applying auth.0009_alter_user_last_name_max_length... OK
demowebapp_1      Applying sessions.0001_initial... OK
demowebapp_1      Applying swa_app.0001_initial... OK
demowebapp_1      Applying swa_app.0002_auto_20180911_0306... OK
demowebapp_1      Applying swa_app.0003_profile... OK
```

Upon completion, you should see the message: **Applying swa_app.0003_profile... OK**
Keep this terminal window open and available so you can view the SCIM traffic.

Verify Docker containers

44. Open another terminal window and enter the following

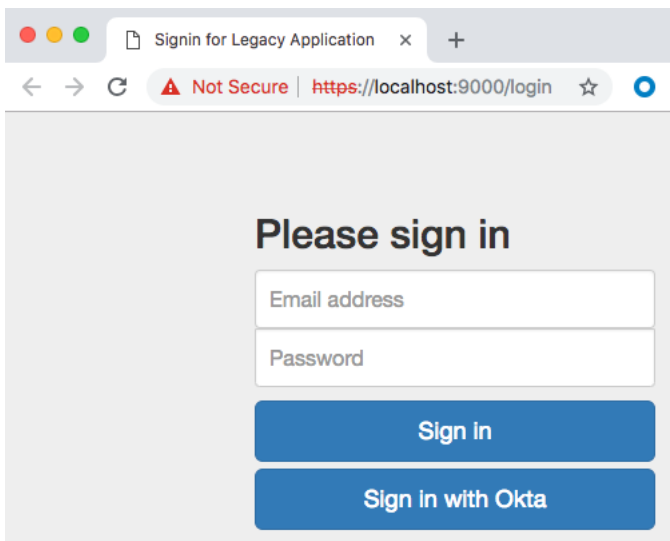
```
$ docker container ls -a
```

At this point, you should have two running containers with the **STATUS** of **Up**

NOTE: The container created from the **swa_opp_demo-master_demowebapp** image is mapped to port **9000**

Verify demowebapp deployment



45. Verify that the demowebapp is successfully deployed – <https://localhost:9000/login>



Verify OPP Agent Deployment - Green

46. In your Okta tenant, navigate to **Dashboard > Agents**

On-Premises Agents

On-Premises Agent	Last Heard From	Used By	
 a66027d0506d	a few seconds ago	0 apps	Activate 

Configure SCIM Connector Configuration

47. Edit the application created earlier in Okta to enable on-premises provisioning for the app.
48. In the App Setting section – Provisioning - Enable on-premises provisioning (**Checked**)

App Settings

Cancel

Application label

Legacy App – Product Catalog

This label displays under the app on your home page

Application visibility

☐ Do not display application icon to users

☐ Do not display application icon in the Okta Mobile App

Browser plugin auto-submit

☐ Automatically log in when user lands on login page

Provisioning

☒ Enable on-premises provisioning

Auto-launch

☐ Auto-launch the app when user signs into Okta.

Application notes for end users

This note will be accessible to all end users via their dashboard

Application notes for admins

This note will only be accessible to admin on this page

Save

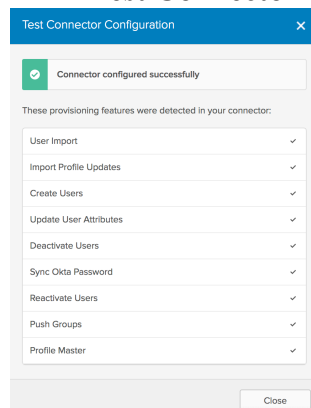
49. Click **Save**.
50. On the Provisioning tab, you should see the message that you have a provisioning agent active
51. Click **Configure SCIM Connector** and specify the following

Field	Value
SCIM connector base URL	https://demowebapp:9000/scim/v2/
Authorization type	HTTP Header
HTTP header name	Authorization

HTTP header value	<Value of API_KEY set in .env file>
Unique user field name	userName
Accept User updates	Store updates to the user's app profile returned by the connector. (Check)
Connect to these agents	The agent listed should match the agent registered earlier (Check)

- NOTE : The SCIM connector URL **MUST BE**: <https://demowebapp:9000/scim/v2/>. "demowebapp" is the internal docker name of the web-app container (remember the network communication is just between the 2 docker containers here). "demowebapp" is also the CN on the SSL certificate.
- The unique user name field must be "userName"
- Ensure that you select the OPP agent that you wish to use. It's likely you'll only have 1 option here.

52. Click **Test Connector configuration**



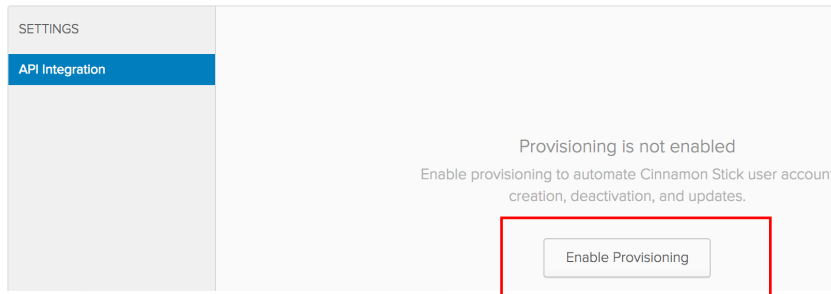
53. Click **Close**

54. Click **Save**

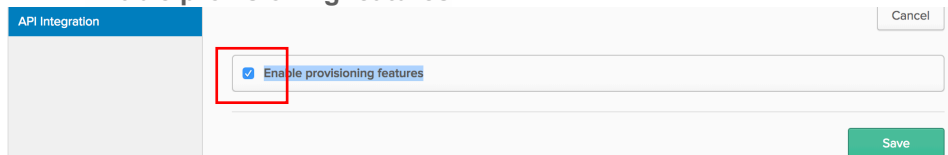
55. View the terminal output. You should see the following message:

```
""GET /scim/v2/ServiceProviderConfigs HTTP/1.1" 200 844"
```

56. On the Application provisioning tab, click the Enable Provisioning button



57. Check **Enable provisioning features**



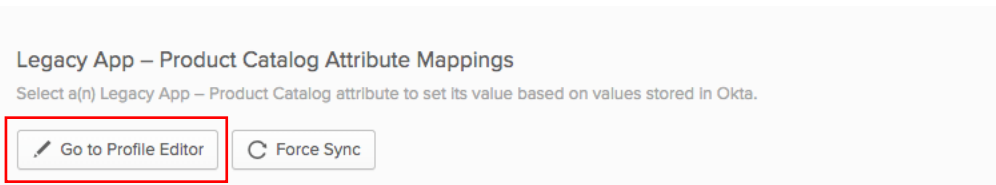
58. Click **Save**

59. Enable the following provisioning options: **Create users, Update User attributes, Deactivate users, Sync Passwords** (Sync a randomly generated password).

60. Click **Save**.

61. On the Provisioning tab, scroll down to the Application Attribute Mappings section

62. Click **Go to Profile Editor**



Assign Users & Groups

63. On the Assignments tab, assign **Al Borland** to the application.

64. View the terminal output to see the SCIM provisioning messages:

```
"GET /scim/v2/Users?filter=userName+eq+%22al.borland%40strenoline.com%22&startIndex=1&count=200 HTTP/1.1" 200 116
```

```
"POST /scim/v2/Users HTTP/1.1" 201 578
```

65. Assign the Group “**Catalog Admin**” to the application

66. View the terminal output to see the SCIM provisioning messages:

Push Groups

67. On the **Push Group** tab, click **Push Groups** and then **Find Groups by Name**.

68. Search for the “Catalog Admin” group

69. Check **Push group memberships immediately**

70. Click **Save**

Assign Legacy App – Product Catalog to Groups

i Extra info is needed to assign this app to a group.
The attributes below will apply to all people assigned to this group.

Phone Number

Department

Country

Opt In For Marketing

Company Name

[Save and Go Back](#) [Cancel and Go Back](#)

71. View the terminal output to see the SCIM provisioning messages

```
"POST /scim/v2/Groups HTTP/1.1" 201 249
```

Verify Application Assignments

72. Verify the application assignment by logging into your Okta tenant as **Al Borland** and launching the *Legacy App – Product Catalog*

Product Catalog

Home Electrical Fasteners Heating


Al Borland's Profile

User Login ID	al.borland@strenoline.com
Opt In Marketing?	Yes
Department	Contractors
Phone Number	810-225-0005
Country	US
Company	Binford Tools

Welcome to our Catalog - Al!
Browse our selection of top quality products.

[Get started today](#)

Featured Product
A revolutionary new design!

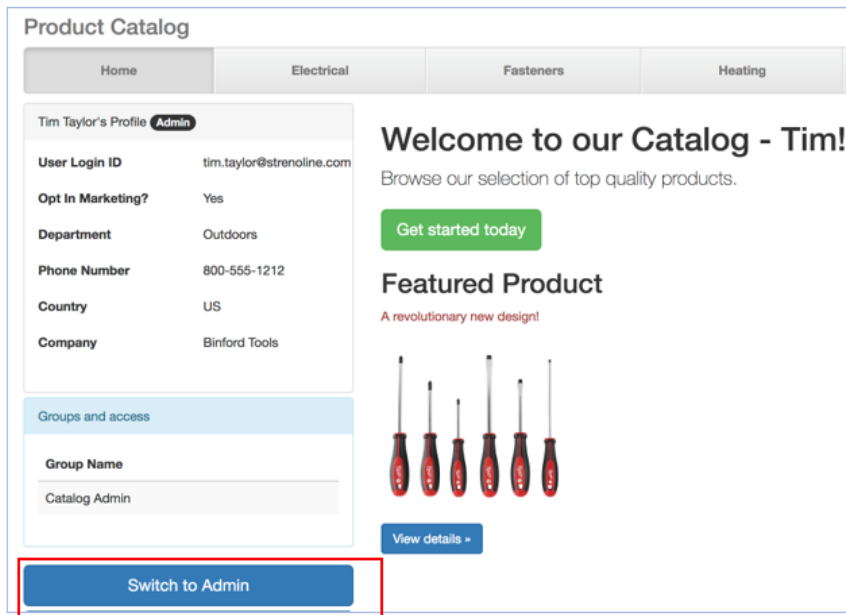


Groups and access

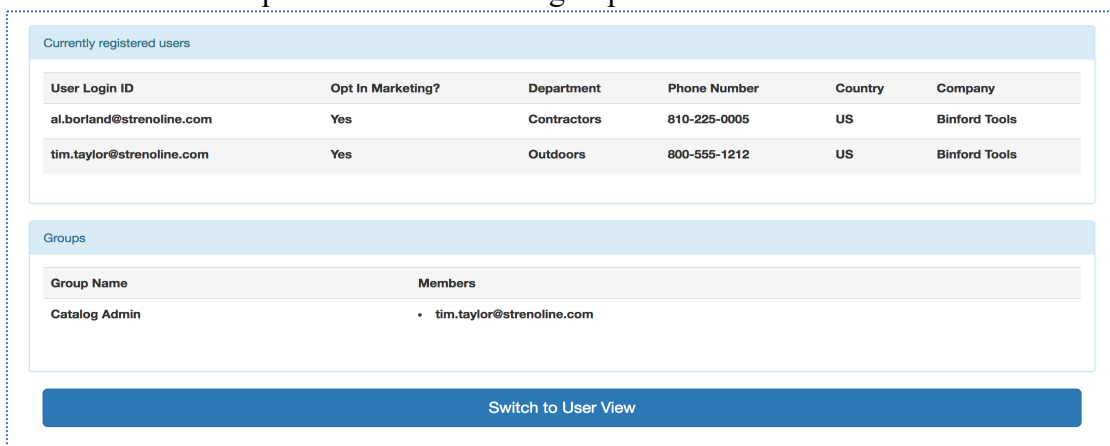
Group Name

73. Close the *Legacy App – Product Catalog* tab and log out of your tenant.

74. Now log in as **Tim Taylor** and launch the *Legacy App – Product Catalog*



75. Since Tim is a member of the Catalog Admins group, you can click the **Switch to Admin** button to see all the provisioned users and groups.



Congratulations! You have completed the Docker OPP App Deployment.

Post Demo: Managing the containers

When you are finished with the demo application, you can stop the containers by entering the following command in a terminal window:

```
$ docker-compose stop
```

To resume running the containers later, enter the following in a terminal window:

```
$ docker-compose up
```