

PIVOTAL CLOUD FOUNDRY

LAB ASSIGNMENTS

Lab 3:

Logging and Troubleshooting

Steps:

In this section you will push an application to Cloud Foundry that has a fatal flaw. By accessing logs you will attempt to diagnose the issue, repair the application, and re-push.

Warning

If you have trouble viewing logs and events using `cf` commands, view the logs in the Apps Manager console instead. This is usually due to proxy or firewall restrictions on the websocket port 4443. The typical errors are:

```
Error dialing loggregator server: dial tcp 1.2.3.4:4443: connection timed out.  
websocket.Dial wss://loggregator.run.pivotal.io:4443/tail/?... connection timed  
out.
```

Debugging Push Problems - IDE Users

IDE (STS /Eclipse) :

1. If you have not done so already, import the `cf-spring-mvc-trouble` project into your IDE. If you have any other projects open, you might like to close them all so you don't work with the wrong project by mistake.
2. Deploy the `cf-spring-mvc-trouble` application on Cloud Foundry
 - a. Be sure to alter the value of subdomain to produce a unique route.
 - b. Watch the console / log output carefully during the deployment. Note that the application appears to be having difficulty starting. Run the following commands to see if you can diagnose the issue (replace "[app]" with your application's name):

i. `cf events [app]`

ii. `cf logs [app] --recent`

iii. `cf logs [app]`

3. Once you have a general feeling for the problem, return to the IDE and examine this Java class:

`gopivotal.cf.workshop.Config` - an easy way to do this is to use `Ctrl-T` or `Cmd-T` to popup the Open Type dialog and enter `Config` - pick the `Config` class in the current project. Find the lines of code marked

with the "BAD CODE" comment. This is a Spring application and the `@PostConstruct` method is automatically called on startup. Do you see why it is failing?

- Comment out the code that is causing the problem. (Java comments are lines starting with the “//” characters.) Save your work (File / Save), return to the “Pivotal Cloud Foundry” tab, and use “Update and Restart” to repush and restart the application.

Note

When using an IDE (Eclipse/STS/IntelliJ) there is no need to ‘build’ the application. Saving causes an automatic build on any modified files.

- The application should now start successfully. Run the cf events and cf logs commands again and note the differences in the logged output.

CLI:

Try the following command in CLI

- cf events <<app-name>>
- cf logs <<app-name>>
- cf logs <<app-name>> --recent

Pivotal Web Console:

In pivotal web console you can see the log details under App→logs link as shown below:

Overview Services Route (1) Env Variables **Logs** Settings Buildpack: https://github.com/

Logs

```

2016-08-10T19:54:08.987+05:30 [CELL/0] [OUT] Starting health monitoring of container
2016-08-10T19:54:09.510+05:30 [CELL/0] [OUT] Container became healthy
2016-08-10T19:55:10.311+05:30 [RTR/5] [OUT] static-demo-cap-studs.cfapps.io - [10/08/2016:14:25:10.309 +0000] "GET / HTTP/1.1" 200 0 532 "-" "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.84 Safari/537.36" 10.10.2.195:55093 x_forwarded_for:"1.39.99.41" x_forwarded_proto:"http" vcap_request_id:9556d2cf-dec8-4df3-7185-43d99e6b9cb0 response_time:0.002442855 app_id:a5e9baab-3704-43d9-a667-d2a04f5da59f
2016-08-10T19:55:10.315+05:30 [APP/0] [OUT] 1.39.99.41, 10.10.2.195 - - [10/Aug/2016:14:25:10 +0000] "GET / HTTP/1.1" 200 544
2016-08-10T19:55:10.664+05:30 [RTR/4] [OUT] static-demo-cap-studs.cfapps.io - [10/08/2016:14:25:10.662 +0000] "GET /js/modernizr.js HTTP/1.1" 200 0 4093 "http://static-demo-cap-studs.cfapps.io/" "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.84 Safari/537.36" 10.10.2.195:57829 x_forwarded_for:"1.39.99.41" x_forwarded_proto:"http" vcap_request_id:e0b431ee-4677-4294-5c99-f2a617c4330c response_time:0.002675904 app_id:a5e9baab-3704-43d9-a667-d2a04f5da59f
2016-08-10T19:55:10.667+05:30 [RTR/0] [OUT] static-demo-cap-studs.cfapps.io - [10/08/2016:14:25:10.662 +0000] "GET /css/foundation.css HTTP/1.1" 200 0 18041 "http://static-demo-cap-studs.cfapps.io/" "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.84 Safari/537.36" 10.10.2.195:57829 x_forwarded_for:"1.39.99.41" x_forwarded_proto:"http" vcap_request_id:29ef00a9-df97-4fa0-5f00-5c583014ebca response_time:0.004970429 app_id:a5e9baab-3704-43d9-a667-d2a04f5da59f
2016-08-10T19:55:10.668+05:30 [APP/0] [OUT] 1.39.99.41, 10.10.2.195 - http://static-demo-cap-studs.cfapps.io/ - [10/Aug/2016:14:25:10 +0000] "GET /css/foundation.css HTTP/1.1" 200 18054
2016-08-10T19:55:10.669+05:30 [APP/0] [OUT] 1.39.99.41, 10.10.2.195 - http://static-demo-cap-studs.cfapps.io/ - [10/Aug/2016:14:25:10 +0000] "GET /js/modernizr.js HTTP/1.1" 200 4105
2016-08-10T19:55:10.978+05:30 [RTR/5] [OUT] static-demo-cap-studs.cfapps.io - [10/08/2016:14:25:10.963 +0000] "GET /js/jquery.js HTTP/1.1" 200 0 73062 "http://static-demo-cap-studs.cfapps.io/" "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.84 Safari/537.36" 10.10.2.195:54962 x_forwarded_for:"1.39.99.41" x_forwarded_proto:"http" vcap_request_id:08e06fc3-e017-4dd2-53e2-c593d33a8691 response_time:0.01458421 app_id:a5e9baab-3704-43d9-a667-d2a04f5da59f

```

Debugging Push Problems - Using cf

In this section we will do the lab using the cf utility.

1. In your terminal or command window, make sure that **cf-spring-mvc-trouble** is the current directory.
2. Deploy the cf-spring-mvc-trouble application on Cloud Foundry. There is no manifest so you will need to specify the -i, -m, -n and -p flags. The application you need to deploy is pre-built/cf-spring-mvc-trouble-0.0.1-SNAPSHOT.war.
 - a. Be sure to specify the value of subdomain (-n option) to produce a unique route.
 - b. Watch the console / log output carefully during the deployment. Note that the application appears to be having difficulty starting. Run the following commands to see if you can diagnose the issue (replace “[app]” with your application’s name):
 - i. **cf events [app]**
 - ii. **cf logs [app] --recent**
 - iii. **cf logs [app]**
3. Once you have a general feeling for the problem, examine the Java code in the following file: gopivotal.cf.workshop.Config.java. An easy way to do this is to enter
 - more src/main/java/com/gopivotal/cf/workshop/Config.java (Linux/MacOS)
 - more src\main\java\com\gopivotal\cf\workshop\Config.java (Windows)in your command/terminal window to view the file. Use the tab key auto-complete feature to avoid typing the path in full. Or feel free to use the editor of your choice.

Note the lines of code marked with the “BAD CODE” comment:

```
@PostConstruct
public void consumeAllMemory() {
// TODO: LOOK HERE! LOOK HERE! LOOK HERE!
// BAD CODE! BAD CODE! BAD CODE!
// The code you see here will consume all available memory allocated
// to the application instance. Comment out this code and re-push
List<Double> list = new ArrayList<Double>();
while (true) {
list.add(Math.random());
}
// End of bad code.
}
```

This program has been deliberately setup to fail - the consumeAllMemory method sits in an infinite loop until all memory is consumed and the application fails to start.

4. Let's assume you contacted a developer colleague and got them to remove the offending code. The new application is pre-built/fixed-cf-spring-mvc-trouble-0.0.1-SNAPSHOT.war. Rerun your cf push command but specify this file instead (using the -p option).
5. The application should now start successfully. Run the cf events and cf logs commands again and note the differences in the logged output.

Conclusion:

In this lab we learnt how to view log reports in CLI and web console.

Congratulations, you have completed this exercise!