CONTAINER ORCHESTRATION

An end of summer recap
By Janaki

THE PROJECT

- Extend the containers to the project
- Automate tests
- “Fail gracefully”

THE GOAL

01 INTRO

Docker images, containers, redhawk, waveforms + components

02 TESTING

Automated tests in python

03 EFFICIENCY

Add in checks and clean up tests

04 CLUSTER PLUGINS

Allow other users to access images on their device

SEE IT IN ACTION

```
def test_bad_image(self):
    self.assertNotEqual(self._domMgr, None)
    self.assertEqual(len(self._domMgr._get_applicationFactories()), 0)
    self.assertEqual(len(self._domMgr._get_applications()), 0)

    self.assertRaises(CF.ApplicationFactory.CreateApplicationError, self._domMgr.createApplication,
                      "/waveforms/bad_image_wf/bad_image_wf.sad.xml", "bad_image_app", [], [])

    #check that domain manager has not crashed
    try:
        self._domMgr._get_identifier()
    except:
        self.assertTrue(False)
```

```
2021-08-19 11:57:10 INFO  ApplicationFactory:2688 - NOT Executing a cluster yet
2021-08-19 11:57:10 TRACE ApplicationFactory:2741 - Waiting 60s for all components to register
2021-08-19 11:57:10 TRACE Application:172 - Component is of cluster type and using cluster logic
2021-08-19 11:57:10 TRACE Cluster:133 - Check terminated bad_image_1:bad_image_app_1 badimage1badimageapp1
2021-08-19 11:57:10 TRACE systemCallLib:44 - Getting cmd docker container ls --filter "name=badimage1badimageapp1" --format '{{json .Status}}' 2>&1
2021-08-19 11:57:10 TRACE Cluster:138 - Status with an output of ""
2021-08-19 11:57:10 ERROR ApplicationFactory:998 - Failed to create application 'bad_image_app': Executing component bad_image_1 implementation cpp: component terminated before registering with application
2021-08-19 11:57:10 TRACE Application:312 - terminate is attempting to delete pod bad_image_1:bad_image_app_1
2021-08-19 11:57:10 TRACE Application:172 - Component is of cluster type and using cluster logic
2021-08-19 11:57:10 TRACE Cluster:133 - Check terminated bad_image_1:bad_image_app_1 badimage1badimageapp1
2021-08-19 11:57:10 TRACE systemCallLib:44 - Getting cmd docker container ls --filter "name=badimage1badimageapp1" --format '{{json .Status}}' 2>&1
2021-08-19 11:57:10 TRACE Cluster:138 - Status with an output of ""
```

A JAVA COMPONENT

```
def test_container_launches_java(self):
    self.assertNotEqual(self._domMgr, None)
    app=self._domMgr.createApplication(
        "/waveforms/JavaContainerComp_wf/JavaContainerComp_wf.sad.xml", "java_wf", [], [])

    self.container_check(app)

    #check that domain manager has not crashed
    try:
        self._domMgr._get_identifier()
    except:
        self.assertTrue(False)
```

```
[[redacted]@localhost testing]$ docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
[[redacted]@localhost testing]$
```

```
[[redacted]@localhost testing]$ docker container ls
CONTAINER ID   IMAGE      COMMAND                  CREATED                  STATUS                  PORTS     NAMES
a4c08b205d4c   javacomp   "/bin/bash -lc '/var…"   Less than a second ago   Up Less than a second             javacomponentjavawf1
```

A CLOSER LOOK

```
def container_check(self,app):
    num_components = len(self._domMgr._get_applications()[0]._get_registeredComponents())
    #loops through each component in list
    for component in self._domMgr._get_applications()[0]._get_registeredComponents():
        identifier=component.identifier

        identifier=identifier.replace("_","")
        identifier=identifier.replace(":","")
        command="docker ps -q -f name="+identifier
        timeout=time.time() + 10
        output=subprocess.Popen(command.split(" "), stdout=subprocess.PIPE).communicate()[0]
        #updates output until there is a component
        while output == "" :
            #fails if function reaches a timeout
            if time.time() > timeout:
                app.releaseObject()
                self.assertTrue(False)
            output=subprocess.Popen(command.split(" "), stdout=subprocess.PIPE).communicate()[0]

        app.releaseObject()
        timeout=time.time() + 10
        #updates output until there is no component
        while output != "":
            #fails if function reaches a timeout
            if time.time() > timeout:
                assertTrue(False)
            output=subprocess.Popen(command.split(" "), stdout=subprocess.PIPE).communicate()[0]
```

MAKEFILE.AM

```
CLUSTERLIST = sdr/dom/components/JavaContainerComp \
              sdr/dom/components/cluster_launch_success \
              sdr/dom/components/bad_image \
              sdr/dom/components/SimpleGain \
              sdr/dom/components/java_component \
              sdr/dom/components/test_python


IMAGE_BUILD = docker build --rm \
                -f ./$1.Dockerfile \
                --build-arg $2="$3" \
                -t $(call lc,$4):$(VERSION) \
                ./

all-local: custom plugins
        python setup.py $(OSSIE_V_pysetup) build

custom:
        $(info INFO:$(foreach CUSTOM,$(CLUSTERLIST),$(call IMAGE_BUILD,$@,custom_asset,$(CUSTOM),$(call dirname,$(CUSTOM)));))
        $(foreach CUSTOM,$(CLUSTERLIST),$(call IMAGE_BUILD,$@,custom_asset,$(CUSTOM),$(call dirname,$(CUSTOM)));)
```

SPD FILE

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE softpkg PUBLIC "-//JTRS//DTD SCA V2.2.2 SPD//EN" "softpkg.dtd">
<softpkg id="DCE:7131e885-86ec-42aa-a9ae-307644466758" name="JavaContainerComp" type="2.0.6">
  <title></title>
  <author>
    <name>null</name>
  </author>
  <propertyfile type="PRF">
    <localfile name="JavaContainerComp.prf.xml"/>
  </propertyfile>
  <descriptor>
    <localfile name="JavaContainerComp.scd.xml"/>
  </descriptor>
  <implementation id="java">
    <description>The implementation contains descriptive information about the template for a software resource.</description>
    <code type="Container">
      <localfile name="java"/>
      <entrypoint>java/startJava.sh::javacontainercomp</entrypoint>
    </code>
    <compiler name="/usr/bin/javac" version="1.7"/>
    <programminglanguage name="Java"/>
    <humanlanguage name="EN"/>
    <runtime name="/usr/bin/java" version="1.7"/>
    <os name="Linux"/>
  </implementation>
</softpkg>
```

WHAT I LEARNED

Testing

C++, Python, Gitlab, Docker

Redhawk

9-5

How to ask for help

Collaboration

CHALLENGES

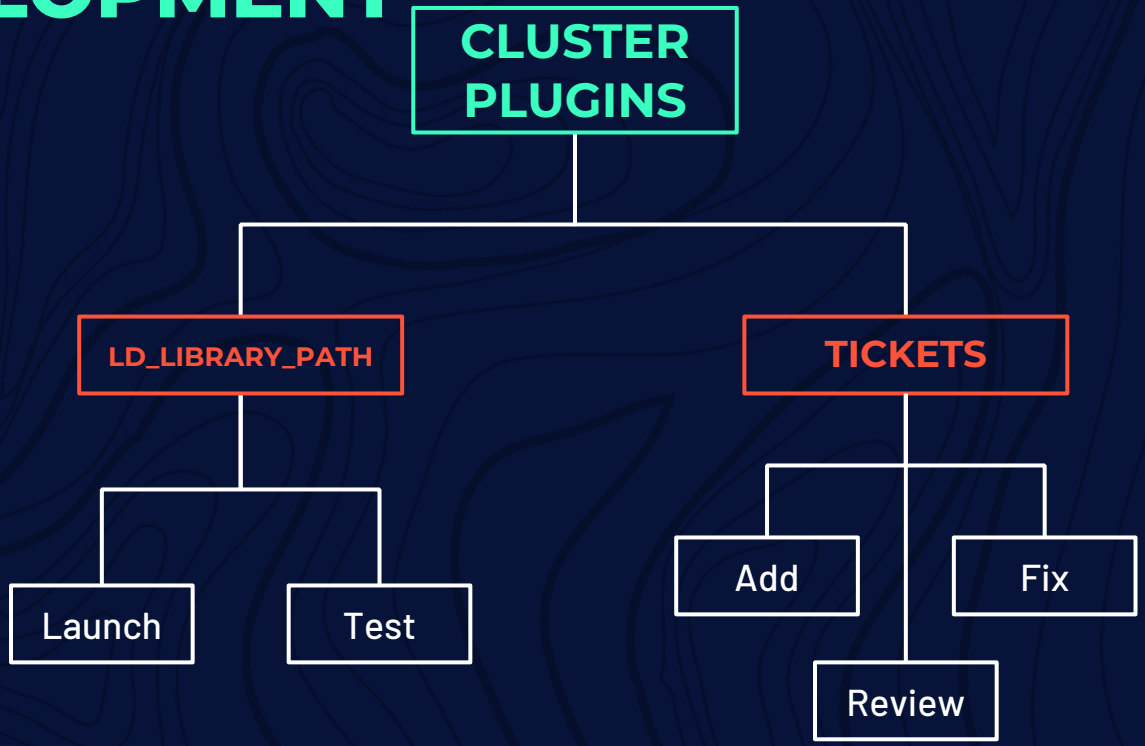
Git Remote

Google

Testing errors

Cluster Plugins

FUTURE DEVELOPMENT



THANKS

Do you have any questions?