IoT Resource Tree Representation for Massive Data Set

Team member:

Qian Chen , Yingchao Zhu

Niklas Kunkel ,Qingqing Li

Tasks Outlined for Sprint 2

Sprint 5 (eow 9) 34 38

Figure out how to use online AWS instance of ODL (shared with other IoT team)

3 5 3/3 3 3 Mar 20

Learn Python

3 5

Write Python to populate the AWS Tree with correct ODL properties (ie: resourceType, oncologyRef, name, label, ect.)

3 1/1 13 13 Mar 25

Write Python script to pull entire AWS Tree data so we can parse it later.

1/1 Mar 25 13 13

Deliver design doc to guide final

Add a card…

sprint 6 (eow 10) [NoBurn] 38

Find sigma js template for demo purposes

Mar 27 2

Study formatting of JSON that sigma.js takes as valid input.

Mar 27 5

Write python script to encode AWS Tree Data into valid JSON

Mar 29 13

Use pre-made sigma.js template to load up encoded JSON file as Proof-of-Concept

Apr 1 5

Learn Javascript

Apr 1 10

Download and build the YANG visualizer

1 3

Add a card…

AWS CRUD (REST)

POST

- Create AE Container & contextInstance

GET

- Retrieve AE Container & contextInstance

PUT

- Update AE Container & contextInstance

DELETE

- Delete AE Container & contextInstance

Generating M2M Tree

```
#This tree builder will let you choose how many container you want to add in an AE
#and how many contentInstance you want to add in each container
#You should have already created the AE in the tree and change the AE_url of your AE name
import json
import requests

def print_container_Stats():
    print container_output.url
    print container_output.status_code
    print container_output.text

def print_contentInstance_Stats():
    print contentInstance_output.url
    print contentInstance_output.status_code
    print contentInstance_output.text

AE_url = 'http://52.10.62.166:8282/InCSE1/Team2AEx'

Parameter = {'from': 'http:localhost:10000', 'requestIdentifier': '12345'}
Header = {'Content-Type': 'application/json', 'Accept': 'application/json'}

container_number = 10
#This is the number how many container you want to add in AE
contentInstance_number = 2
#This is the number how many contentInstance you want to add in each container
for container_count in xrange(0,container_number):
    #This for loop will create container in AE
    temp1 = str(container_count)
    container_name = 'container' + temp1
    #Create container
    Data_container =  "{\"from\":\"http:localhost:10000\",\"requestIdentifier\":\"12345\",\"resourceType\":\"container\",\"content\":{\"labels\":\"T
    print 'Post Request Creating container'
    container_output = requests.post(AE_url, params= Parameter, headers = Header, data= Data_container)
    print_container_Stats()
    for contentInstance_count in xrange(0,contentInstance_number):
    #This for loop will create contentInstance in each container
        temp2 = str(contentInstance_count)
        contentInstance_name = 'contentInstance' + temp2
        #Create contentInstance
        container_url = AE_url + '/%s' %(container_name)
        Data_contentInstance = "{\"from\":\"http:localhost:10000\",\"requestIdentifier\":\"12345\",\"resourceType\":\"contentInstance\",\"content\":
        print 'Post Request Creating contentInstance'
        contentInstance_output = requests.post(container_url, params= Parameter, headers = Header, data= Data_contentInstance)
        print_contentInstance_Stats()
```

Scraping AWS Tree Data

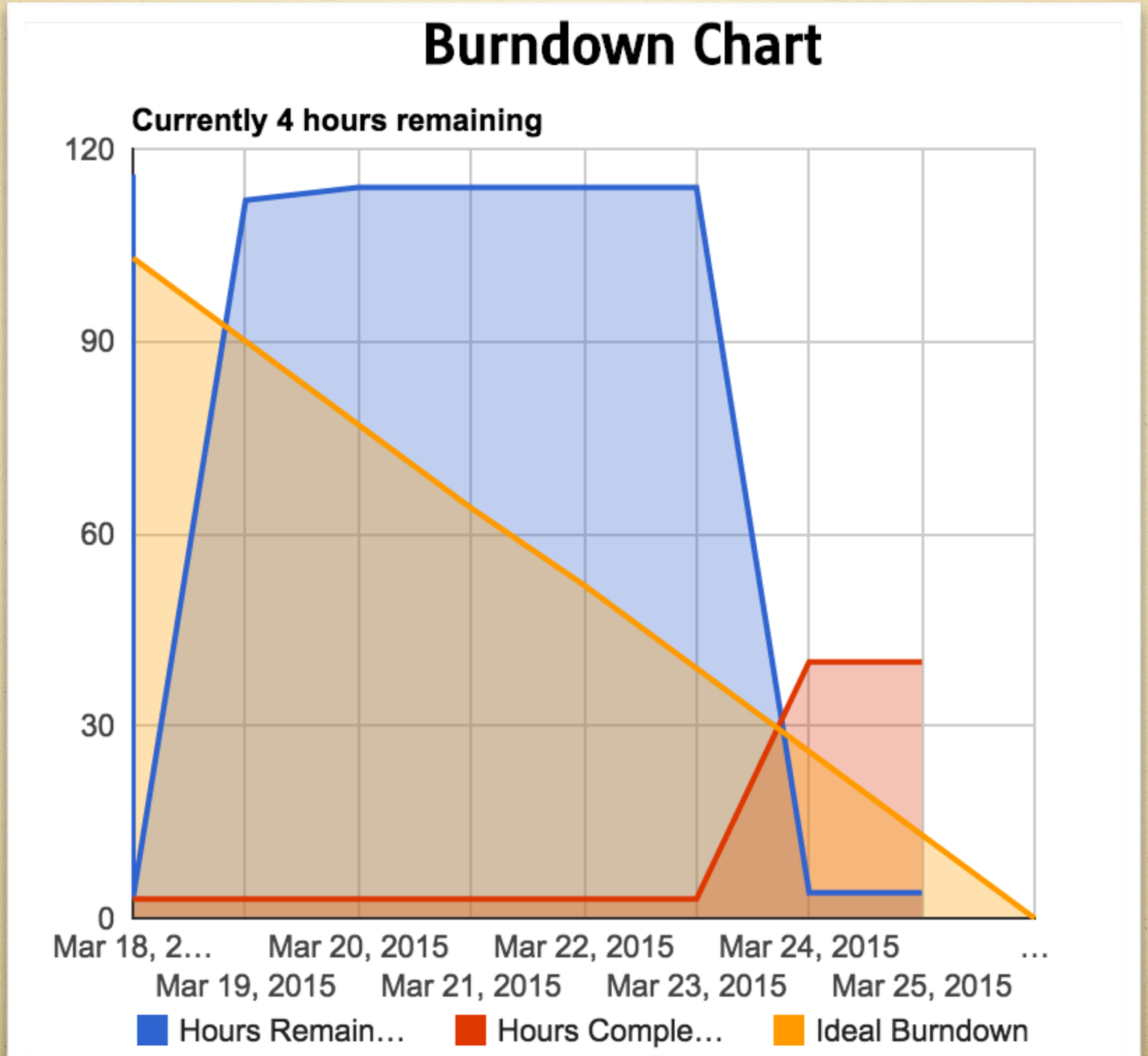
iot-ui-bigdata — bash — 80×24


```
nstance(latest-allAttributes)","resourceID":"contentInstance0","Attributes":[{"a
ttributeName":"resourceID","attributeValue":"645378569"},{"attributeName":"resou
rceName","attributeValue":"contentInstance0"},{"attributeName":"resourceType","a
ttributeValue":"class org.opendaylight.yang.gen.v1.http.cisco.com.ns.onem2m.coma
tt.rev140727.ContentInstance"},{"attributeName":"creationTime","attributeValue":
"23-03-15T22:15:337"},{"attributeName":"lastModifiedTime","attributeValue":"23-0
3-15T22:15:337"},{"attributeName":"labels","attributeValue":"Test_For_Python_Cod
e"},{"attributeName":"parentID","attributeValue":"InCSE1/Team2AEx/container9"},{
"attributeName":"stateTag","attributeValue":"1"}]}]}}

------------------------------
{"output":{"responseStatusCode":2002,"ResourceOutput":[{"resourceType":"contentI
nstance(latest-allAttributes)","resourceID":"contentInstance1","Attributes":[{"a
ttributeName":"resourceID","attributeValue":"645378568"},{"attributeName":"resou
rceName","attributeValue":"contentInstance1"},{"attributeName":"resourceType","a
ttributeValue":"class org.opendaylight.yang.gen.v1.http.cisco.com.ns.onem2m.coma
tt.rev140727.ContentInstance"},{"attributeName":"creationTime","attributeValue":
"23-03-15T22:15:502"},{"attributeName":"lastModifiedTime","attributeValue":"23-0
3-15T22:15:502"},{"attributeName":"labels","attributeValue":"Test_For_Python_Cod
e"},{"attributeName":"parentID","attributeValue":"InCSE1/Team2AEx/container9"},{
"attributeName":"stateTag","attributeValue":"2"}]}]}}

------------------------------
wireless1x-155-41-85-188:iot-ui-bigdata Christina$
```

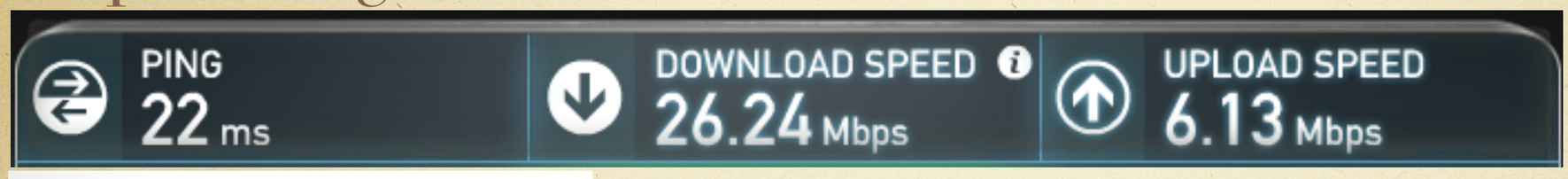
Burndown Chart



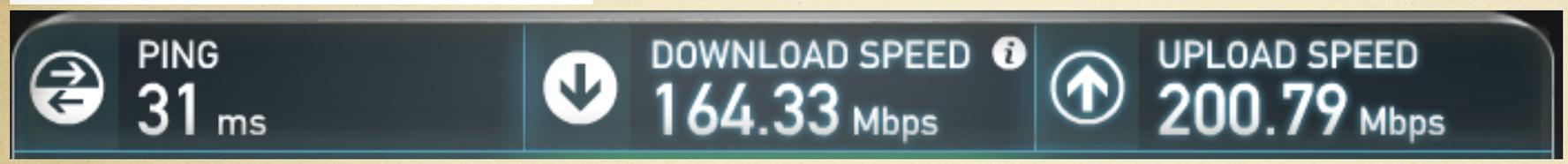
“Demo”

About the time

Populating Tree /w 100 containers & 100 CI in each



2303.38758397 = 38.38 mins



1693.16622496 = 28.21 mins

Localhost

86.2315950394

Retrieve Only Container

16.9789791107

Retrieve All

1777.47971702 = 29.61 mins

Localhost

49.7392141819

Work for the next Sprint