DELL Technologies

Pied Piper Technical Workshop

Day 3: Automation

Abstract

This document is provided to assist attendees with completing the appropriate labs to apply the concepts and knowledge learnt throughout the technical workshop program. It is not intended to be used or distributed in isolation and may not contain all required information.

May 2020

Student Lab Guide

DELL Technologies

Revisions

Version	Date	Description
0.1	May 2020	Initial draft
0.2	June 2020	Updates- Labs documented with detailed steps and additional screenshots

Student Lab Guide

DELL Technologies

Table of Contents

DELL Technologies

Lab 1: Postman & JSON Viewer

Module Objectives:

- ❑ Learn about REST based API's
- ❑ Connect and pull data from a public API
- ❑ Learn about JSON formatting
- ❑ Parse a REST API response payload with a JSON viewer

DELL Technologies

Lab Exercise – Part A: Use Postman client to connect to a public API service

Complete the below steps to demonstrate your understanding of the tools and concepts required for the remaining lab exercises;

1. Open the Postman app
2. Sign in with your account (create a new account if not already setup)
3. Create a new request
4. Enter the following URL into the GET request field

   http://api.open-notify.org/iss-now.json

5. Click send and review the response

```
{
    "message": "success",
    "iss_position": {
        "longitude": "-47.0391",
        "latitude": "-38.0618"
    },
    "timestamp": 1585719494
}
```

6. By default, the Postman app will display the response in a nicely formatted JSON view. Toggle the view and explore what a raw response looks like.

7. What is this API service and what data is being returned?
8. What happens when you change the request type from a GET to a POST?
9. Explore the other tabs in the interface (e.g. Params, Authorisation, Headers, Body, etc…)

DELL Technologies

Lab Exercise – Part B: Use a JSON parser to nicely format the response data

Complete the below steps to demonstrate your understanding of the tools and concepts required for the remaining lab exercises;

1. Open chrome and browse to this URL

   > http://api.open-notify.org/iss-now.json

2. If the JSON Viewer extension is enabled, you will see a nicely formatted JSON

3. Use the extension controls on the right-hand side of the page to toggle between RAW and JSON formatting

4. Open Atom text editor and create a new file with the raw API data

   > {"message": "success", "iss_position": {"latitude": "23.2789", "longitude": "-131.9393"}, "timestamp": 1585777695}

5. Save the file as test.json and Atom will colour code the text as per below

6. Atom has a package that can be installed called 'pretty-json' and helps format JSON files for easy viewing. Use the packages menu to select the 'Prettify' option in the 'Pretty JSON' sub menu or use the ctrl+shift+j keyboard shortcut.

DELL Technologies

Lab Exercise – Bonus Lab: Use input parameters to interact with the API

Complete the below steps to demonstrate your understanding of the tools and concepts required for the remaining lab exercises;

1. Open the Postman app and perform a GET request on the below API

   > http://api.open-notify.org/iss-pass.json

2. The response will be a failure message as the API requires input parameters

3. Enter the following key / value parameters into the Query Params section

   > lat : -33.8

   > lon : 151.2

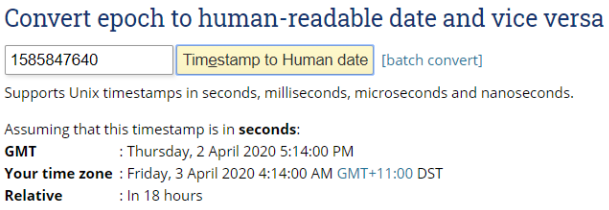
   * These are the longitude and latitude values for Sydney, Australia

3. Submit the query and observe the response

4. The response provides a list of upcoming passes that the ISS can be seen from the provided location (i.e. Sydney)

5. The 'risetime' is in EPOCH format and can be easily converted as shown below;

   https://www.epochconverter.com/




```
{
  "message": "success",
  "request": {
    "altitude": 100,
    "datetime": 1585781411,
    "latitude": -33.8,
    "longitude": 151.2,
    "passes": 5
  },
  "response": [
    {
      "duration": 656,
      "risetime": 1585781707
    },
    {
      "duration": 562,
      "risetime": 1585787545
    },
    {
      "duration": 522,
      "risetime": 1585836016
    },
    {
      "duration": 662,
      "risetime": 1585841730
    },
    {
      "duration": 531,
      "risetime": 1585847640
    }
  ]
}
```

DELL Technologies

Retrospective: Experimented and tested a public API without authentication

- ❑ Learnt about REST based API's
- ❑ Connected to the http://api.open-notify.org/ public API using the Postman development tool and natively in a web browser
- ❑ Used a browser extension plugin and text editor package to parse raw API data and format into human readable JSON formatting
- ❑ Bonus: experimented with input parameters and passed values to the API for a customised response

DELL Technologies

Lab 2: Dell EMC API's

Module Objectives:

- ❑ Learn about some of Dell EMC's REST API's
  - ➢ Isilon OneFS API
  - ➢ ECS API
- ❑ Experiment with Postman and create auth session tokens
- ❑ Explore API objects
- ❑ Test the functionality

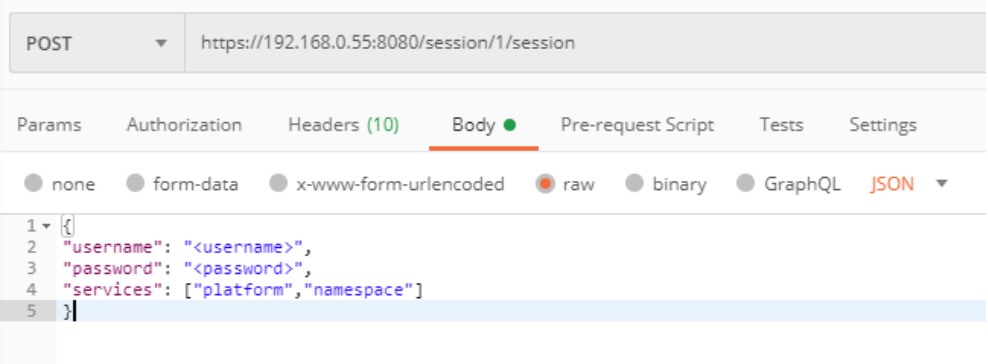
DELL Technologies

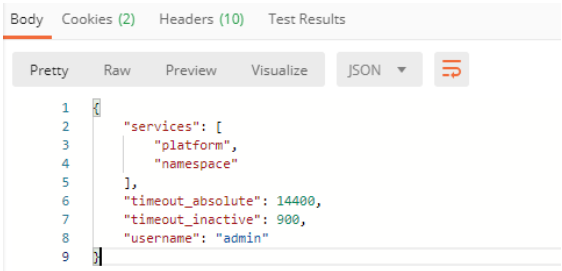
Lab Exercise – Part A: Create an Isilon login session

Complete the below steps to demonstrate your understanding of the tools and concepts required for the remaining lab exercises;

1. Open the Postman app
2. Sign in with your account (create a new account if not already setup)
3. Create a new request
4. Go to File --> Settings and disable SSL verification
5. Enter the following URL into a POST request field (change the GET dropdown to POST)
   > https://isilon.demo.local:8080/session/1/session
6. In the 'Headers' tab, add the key value pair below;
   > Content-Type : application/json
7. In the 'Body' tab, enter the following raw code;
   > {
   > "username": "<username>",
   > "password": "<password>",
   > "services": ["platform","namespace"]
   > }

   Replace with your Isilon credentials (e.g. ansible / ansible)
8. Click send and review the response





DELL Technologies

Lab Exercise – Part A: Retrieve the details of all SMB shares on Isilon, using the API

Complete the below steps to demonstrate your understanding of the tools and concepts required for the remaining lab exercises;

1. Copy the isicsrf cookie value returned in the previous session request

isicsrf	2ca1e91e-ef2f-4d0c-992f-7db22bc391d7

2. Create a new GET request
3. Enter the following URL into a request field
   > https://Isilon.demo.local:8080/platform/7/protocols/smb/shares
4. In the 'Headers' tab, add the following 2 key value pairs;
   > X-CSRF-Token : <paste the cookie value from step 1>
   > Referer : https://Isilon.demo.local:8080
5. Click send and review the response


```
"shares": [
    {
        "access_based_enumeration": false,
        "access_based_enumeration_root_only": false,
        "allow_delete_readonly": false,
        "allow_execute_always": false,
        "allow_variable_expansion": false,
        "auto_create_directory": false,
        "browsable": true,
        "ca_timeout": 120,
        "ca_write_integrity": "write-read-coherent",
        "change_notify": "norecurse",
        "continuously_available": false,
        "create_permissions": "default acl",
        "csc_policy": "manual",
        "description": "Isilon OneFS",
        "directory_create_mask": 448,
        "directory_create_mode": 0,
        "file_create_mask": 448,
        "file_create_mode": 64,
        "file_filter_extensions": [],
        "file_filter_type": "deny",
        "file_filtering_enabled": false,
        "hide_dot_files": false,
        "host_acl": [],
        "id": "ifs",
        "impersonate_guest": "never",
        "impersonate_user": "",
        "inheritable_path_acl": false,
        "mangle_byte_start": 60672,
        "mangle_map": [
            "0x01-0x1F:-1",
            "0x22:-1",
            "0x2A:-1",
            "0x3A:-1",
            "0x3C:-1",
            "0x3E:-1",
            "0x3F:-1",
            "0x5C:-1"
        ],
        "name": "ifs",
        "ntfs_acl_support": true,
        "oplocks": true,
        "path": "/ifs",
        "permissions": [
            {
                "permission": "full",
                "permission_type": "allow",
                "trustee": {
                    "id": "SID:S-1-1-0",
                    "name": "Everyone",
                    "type": "wellknown"
                }
            }
        ],
        "run_as_root": [],
        "smb3_encryption_enabled": false,
        "sparse_file": false,
```

DELL Technologies

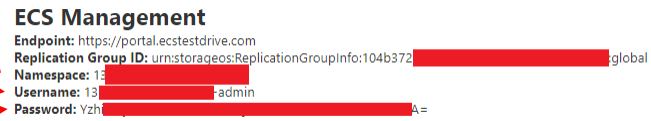
Lab Exercise – Part B: Create an ECS login session

Complete the below steps to demonstrate your understanding of the tools and concepts required for the remaining lab exercises;

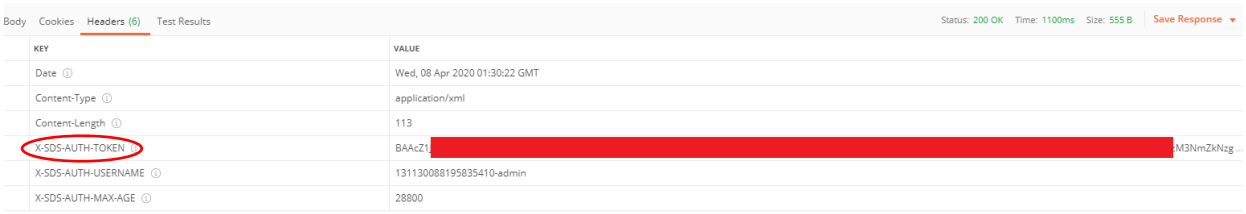
1. Open the Postman app
2. Sign in with your account (create a new account if not already setup)
3. Create a new request
4. Enter the following URL into a GET request field
   - > https://portal.ecstestdrive.com/login
5. In the 'Parameters' tab, add the key value pair below;
   - > Namespace : <namespace ID>
6. In the 'Authorization' tab, select "Basic Auth" and the enter the below;
   - > Username : <Username>
   - > Password : <Password>
7. Click send and review the response

Replace with your ECS Test Drive credentials

https://portal.ecstestdrive.com/Credentials/Overview



***Note: check that the header returned an X-SDS-AUTH-TOKEN value*

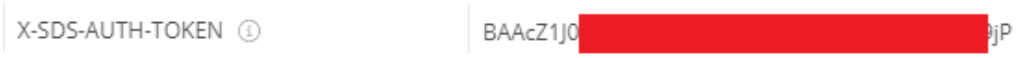


DELL Technologies

Lab Exercise – Part B: Retrieve a list of Object Storage buckets on ECS, using the API

Complete the below steps to demonstrate your understanding of the tools and concepts required for the remaining lab exercises;

1. Copy the X-SDS-AUTH-TOKEN header value returned in the previous login request



2. Create a new GET request
3. Enter the following URL into a request field
   > https://portal.ecstestdrive.com/object/bucket.json
4. In the 'Parameters' tab, add the following key value pair;
   > Namespace : <the same namespace ID as used in the session login>
5. In the 'Headers' tab, add the following key value pair;
   > X-CSRF-Token : <paste the header value from step 1>
6. Click send and review the response

DELL Technologies

Retrospective: Used Postman client to connect to Dell EMC API's with authentication

- ❑ Learnt about Dell EMC REST API's
- ❑ Experimented using Postman client to connect to Isilon and ECS REST API endpoints
- ❑ Created authentication tokens
- ❑ Explored API objects and retrieved data from endpoints


```
        "name": "ifs",
        "ntfs_acl_support": true,
        "oplocks": true,
        "path": "/ifs",
        "permissions": [
            {
                "permission": "full",
                "permission_type": "allow",
                "trustee": {
                    "id": "SID:S-1-1-0",
                    "name": "Everyone",
                    "type": "wellknown"
                }
            }
        ],
        "run_as_root": [],
        "smb3_encryption_enabled": false,
        "sparse_file": false,
```


Body Cookies Headers (3) Test Results

Pretty Raw Preview Visualize JSON


```
{
    "object_bucket": [
        {
            "name": "photo-album",
            "id": "1[redacted]0.photo-album",
            "link": {
                "rel": "self",
                "href": "/object/bucket/1[redacted]0.photo-album"
            },
            "namespace": "1[redacted]10",
            "locked": false,
            "created": "2020-03-03T09:26:56.395Z",
            "retention": 0,
            "vpool": "urn:storageos:ReplicationGroupInfo:104b3728-fba1-41b3-86
            "fs_access_enabled": false,
            "softquota": "-1",
            "is_stale_allowed": true,
            "is_tso_read_only": false,
            "default_retention": 0,
            "block_size": -1,
            "auto_commit_period": 0,
            "notification_size": -1,
            "is_encryption_enabled": "false",
```

DELL Technologies

Lab 3: Ansible with Dell EMC Isilon File Storage

Module Objectives:

- ❑ Learn about basic Ansible tasks, modules & playbooks
- ❑ Use Ansible modules to manage an Isilon storage appliance
- ❑ Use the Ansible URI module to connect to Isilon RESTAPI
- ❑ Write a playbook to create a new SMB share
- ❑ Test its functionality

DELL Technologies

Lab Exercise – Part A: Ping & SSH Modules

Complete the below steps to demonstrate your understanding of the tools and concepts required for the remaining lab exercises;

1. Clone the following repository (if not already);

   https://github.com/theocrithary/Piper-2020/tree/master/Day%203

2. Review the files contained in the Day 3 directory
   - ansible-isilon-create_smb_share.yaml
   - ansible-isilon-isi_status.yaml
3. Open the putty client from your desktop and ssh to the Linux server provided by your instructor
   - ssh root@<your linux server IP>
4. Type the following command and observe the output
   - ansible all -m ping -k
5. Create a new file using the vi editor and copy the contents provided in the github repo
   - vi ansible-isilon-isi_status.yaml
6. Type this command to run the ansible playbook and check the Isilon status
   - ansible-playbook ansible-isilon-isi_status.yaml
7. Enter the Isilon ssh password when prompted and observe the results

DELL Technologies

Lab Exercise – Part B: URI Module

Complete the below steps to demonstrate your understanding of the tools and concepts required for the remaining lab exercises;

1. Create a new file using the vi editor and copy the contents provided previously

   > vi ansible-isilon-create_smb_share.yaml
   (be sure to replace the IP address with the Isilon IP assigned to your lab)

2. Type this command to run the ansible playbook and create a new SMB share

   > ansible-playbook ansible-isilon-create_smb_share.yaml

3. Observe the results and take note of the difference between this result and the previous playbook

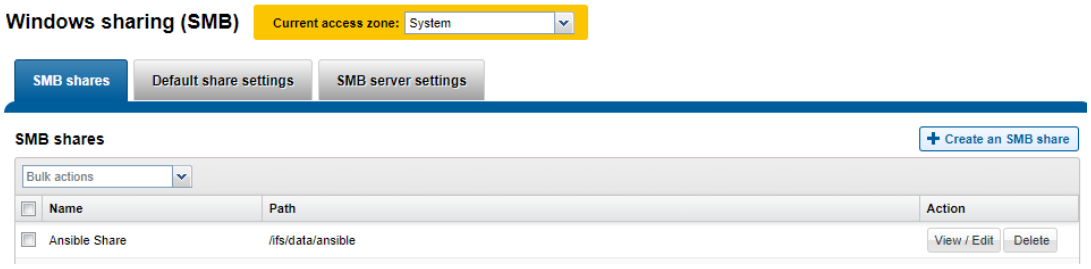
4. Open a browser and login to your Isilon appliance using the credentials provided by your instructor
   https://<Isilon IP address>:8080/

5. Navigate to the 'Protocols' -> 'Windows sharing (SMB)' dropdown menu

6. You should see a new share created by the Ansible playbook you just ran

```
Isilon_Share_Name: "Ansible Share"
Isilon_Share_Path: /ifs/data/ansible
Isilon_Share_Descr: "Share created by Ansible RESTAPI playbook"
```

```
"json": {
    "id": "Ansible Share"
},
"location": "https://192.168.0.55:8080/8/protocols/smb/shares/Ansible%20Share",
"msg": "OK (unknown bytes)",
```



DELL Technologies

Retrospective: A basis for an IaC service catalogue using Isilon File Storage

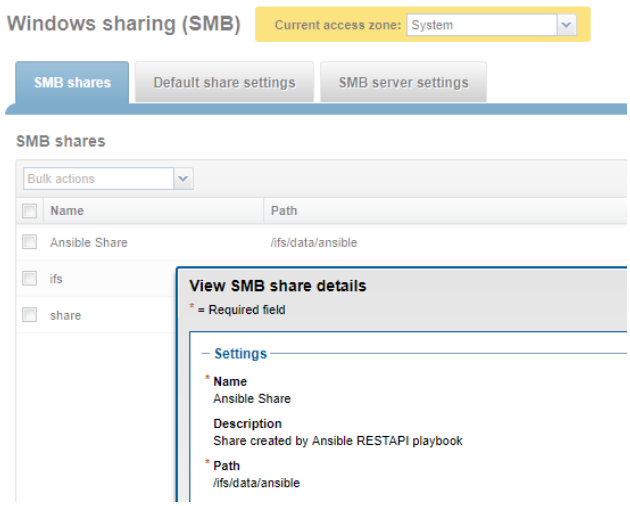
You now have an Ansible playbook that can be invoked by a workflow such as an ITSM ticket request or a self-service portal order.

- ❑ Learnt about Ansible tasks, modules & playbooks
- ❑ Used the ping and ssh modules to connect to Isilon
- ❑ Used the URI module to connect to the Isilon REST API
- ❑ Wrote a playbook to create a new SMB share

```
[root@centos ~]# ansible-playbook ansible-isilon-isi_status.yaml

PLAY [Query isilon status] *****************************************

TASK [Gathering Facts] *********************************************
Enter passphrase for key '/root/.ssh/id_rsa':
ok: [192.168.0.55]

TASK [connect to "{{ ansible_host }}" via SSH and issue "isi status"] ***
changed: [192.168.0.55]

TASK [debug] *******************************************************
ok: [192.168.0.55] => {
    "result": {
        "changed": true,
        "cmd": [
            "isi",
            "status"
        ],
```

```
[root@centos ~]# ansible-playbook ansible-isilon-create_smb_share.yaml

PLAY [Isilon API get authentication token] *************************

TASK [Gathering Facts] *********************************************
Enter passphrase for key '/root/.ssh/id_rsa':
ok: [192.168.0.55]

TASK [get Isilon API session ID] ***********************************
ok: [192.168.0.55]
```

```
TASK [create SMB share] ********************************************************
ok: [192.168.0.55]

TASK [debug] *******************************************************************
ok: [192.168.0.55] => {
    "result_create": {
        "allow": "GET, POST, DELETE, HEAD",
        "changed": false,
        "connection": "close",
        "content_type": "application/json",
        "cookies": {},
        "date": "Tue, 14 Apr 2020 12:00:48 GMT",
        "failed": false,
        "json": {
            "id": "Ansible Share"
        },
        "location": "https://192.168.0.55:8080/8/protocols/smb/shares/Ansible%20Share",
        "msg": "OK (unknown bytes)",
```



DELL Technologies

Lab 4: Ansible with Dell EMC ECS Object Storage

Module Objectives:

- ❑ Use the Ansible URI module to connect to ECS RESTAPI
- ❑ Write a playbook to create a new ECS Bucket
- ❑ Test its functionality

DELL Technologies

Lab Exercise: Create a basic Python app to run the Flask web server

Complete the below steps to demonstrate your understanding of the tools and concepts required for the remaining lab exercises;

1. Clone the following repository (if not already);

   https://github.com/theocrithary/Piper-2020/tree/master/Day%203

2. Review the file contained in the Day 3 directory
   - ansible-ecs-create_bucket.yaml
3. Open the putty client from your desktop and ssh to the Linux server provided by your instructor
   - ssh root@<your linux server IP>
4. Create a new file using the vi editor and copy the contents provided in the github repo

   > vi ansible-ecs-create_bucket.yaml
   (be sure to replace all the variables in the vars: section with your ECS Test Drive credentials)
5. Type this command to run the ansible playbook and create a new ECS Bucket

   > ansible-playbook ansible-ecs-create_bucket.yaml
6. Observe the results
7. Use the S3 Browser app to confirm the bucket was created and upload a file to test its permissions

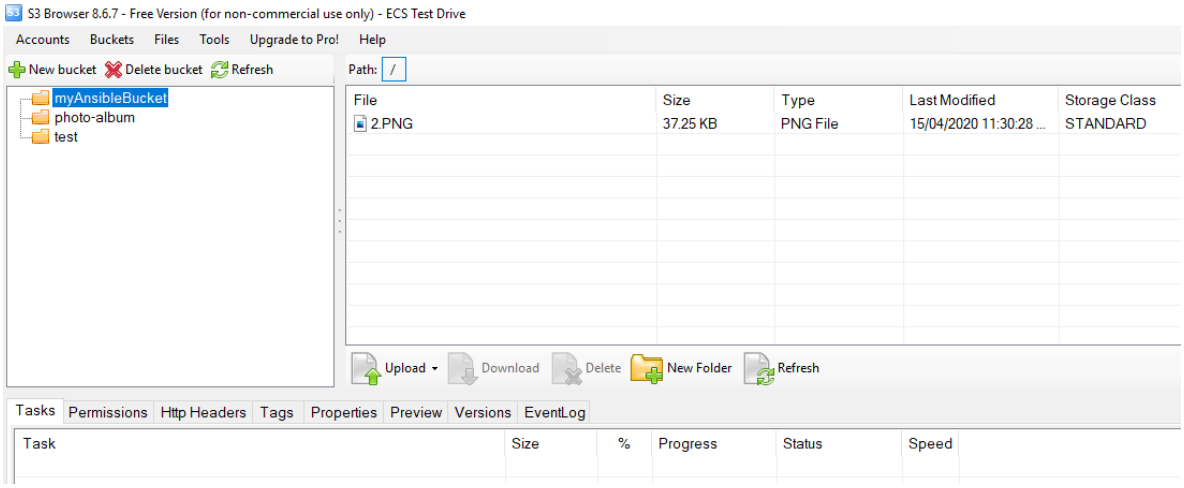
DELL Technologies

Retrospective: A basis for an IaC service catalogue using ECS Object Storage

You now have an Ansible playbook that can be invoked by a workflow such as an ITSM ticket request or a self-service portal order.

- ❑ Used the URI module to connect to ECS REST API
- ❑ Wrote a playbook to create a new ECS Bucket

```
[root@centos ~]# ansible-playbook ansible-ecs-create_bucket.yaml

PLAY [ECS create new S3 bucket] ************************************************************************

TASK [Get ECS API authentication token] ****************************************************************
ok: [localhost]

TASK [Connect to ECS using a previously stored cookie to create new bucket] ****************************
ok: [localhost]

TASK [debug] *******************************************************************************************
ok: [localhost] => {
    "result_create": {
        "changed": false,
        "connection": "close",
        "content": "{\"name\":\"myAnsibleBucket\",\"id\":\"13                0.myAnsibleBucket\",\"inactive\":f
        "content_type": "application/json",
        "cookies": {},
        "date": "Wed, 15 Apr 2020 01:27:39 GMT",
        "failed": false,
        "json": {
            "TagSet": [],
            "global": null,
            "id": "13                0.myAnsibleBucket",
            "inactive": false,
            "metaData": {
                "isEnabled": false,
                "maxKeys": 0,
                "metadata": []
            },
            "name": "myAnsibleBucket",
            "remote": null,
            "vdc": null
        },
        "msg": "OK (unknown bytes)",
        "redirected": false,
        "status": 200,
        "transfer_encoding": "chunked",
        "url": "https://portal.ecstestdrive.com/object/bucket.json"
    }
}

PLAY RECAP *********************************************************************************************
localhost                  : ok=3    changed=0    unreachable=0    failed=0
```



DELL Technologies

Lab 5: VMware vRealize Automation

Module Objectives:

- ❑ Learn about VMware's IaC tools
  - ❖ vRealize Automation
    - Cloud Assembly
    - Service Broker
    - Orchestrator
    - Code Stream
- ❑ Build a Cloud Assembly blueprint to deploy a VM
- ❑ Create a Service Broker catalogue item to publish the blueprint to the portal
- ❑ Bonus: create a CI/CD pipeline integrated into Code Stream
- ❑ Test its functionality

DELL Technologies

Lab Exercise – Part A: Create a service blueprint with Cloud Assembly

1. Open a new browser session using incognito or private browsing to avoid cookie conflicts between the Dell EMC democenter lab and the VMware HOL lab portals
2. Login to https://labs.hol.vmware.com and connect to the below lab;
    - HOL-2002-03-CMP – vRealize Automation Cloud - Getting Started
3. Launch Chrome and checkin with your Hands-on labs email account to retrieve your Lab ID
4. Click on the username and password link to open the Cloud Services console
(password: VMware1!)
5. Navigate to: Cloud Assembly -> Blueprints -> New
    - Provide a name: My First App, Select project: trading, Create
    - Add a 'Cloud Machine' to the canvas
    - Click the Properties tab on the right and toggle the 'Show all properties' switch
    - Scroll down and select the following; Image Type: ubuntu, Flavor: small
    - Add a cloud constraint by clicking the plus sign under the cloud config -> constraints section and entering: platform:azure
    - Add a 'Cloud Network' to the canvas
    - Connect the cloud machine to the network by clicking and dragging the small white circle to the left of the cloud machine box and dropping it onto the cloud network box
6. Click 'version' and create -> Click 'deploy' and give it a name: simple blueprint
(wait for it to complete: 1-2 mins)

DELL Technologies

Lab Exercise – Part B: Publish the blueprint as a service catalogue using Service Broker

1. Continue the previous lab session and find the Service Broker section in the console
2. Navigate to: Service Broker -> Content & Policies
    - Click on the ‘trading cfts’ content source, review the details and click ‘save & import’
    - Go to ‘content sharing’ and select ‘trading’ as the Project
3. Click on ‘Add Items’ and select the ‘trading cfts’ content source, then save
4. Navigate to the ‘Catalog’ tab at the top of the page and find the ‘Apache web server’ item and click ‘request’
    - Give it a name: cft1 and check that the project is ‘trading’ and key name is ‘Hands-On Labs’ before submitting

DELL Technologies

Retrospective: A taste of what VMware tools are available for Infrastructure Automation

- ❑ Learnt about VMware IaC tools
- ❑ Built a blueprint
- ❑ Created a self-service catalogue item in the portal
- ❑ Tested the functionality

This is only a small subset of VMware's capability in the area of Infrastructure Automation, please continue to explore these services and reach out to a VMware representative for additional support.

DELL Technologies

Lab 6: CI/CD with Jenkins

Module Objectives:

- ❑ Learn about Jenkins as CI/CD tool
- ❑ Configure Pipeline as a Code
- ❑ Build (Trigger) Pipeline job
- ❑ Review stages- Checkout, Build, Publish and Clean-up

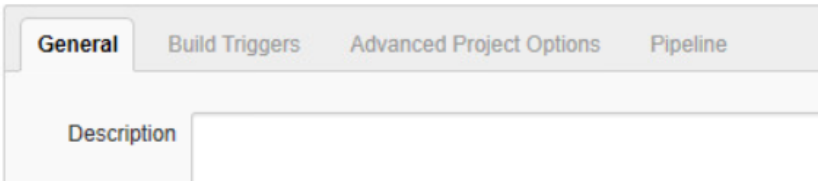
DELL Technologies

Lab Exercise: Create a build pipeline with Jenkins

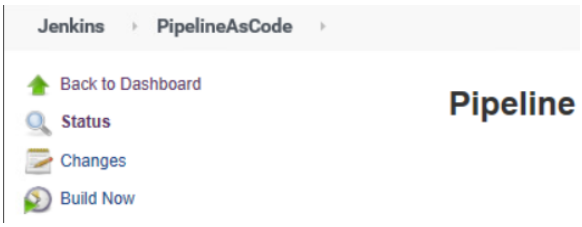
Complete the below steps to demonstrate your understanding of the tools and concepts required for the remaining lab exercises;

(Require Github & Docker hub accounts)

1. Run commands from RunDIND & RunJenkins file on Windows command prompt
2. Run to check Jenkins BlueOcean and DIND containers are running
   - docker ps
3. Login to http://localhost:8080 with Credentials (admin/Password123!)
4. Create new job & configure as below



   - Jobname- PipelineAsCode
   - Type- Pipeline
   - Build Trigers- Check “github hook trigger for GITScm polling”
   - Pipeline Definition- Pipeline Script from SCM
   - SCM- Git
   - Repository URL- https://github.com/YOURGITHUBID/Piper-2020.git
   - Credentials- github (your credentials) & Dockerhub ((your credentials)
   - Replace Jenkins file in your github repo with dockerhub login id-
     - registry = " YourDockerHubID/YourRepoName“
   - Script Path- (relative path to Jenkins file in repo) e.g. Day 03/Lab 06 – Jenkins/Jenkinsfile
5. Click on Pipeline & Click Build Now

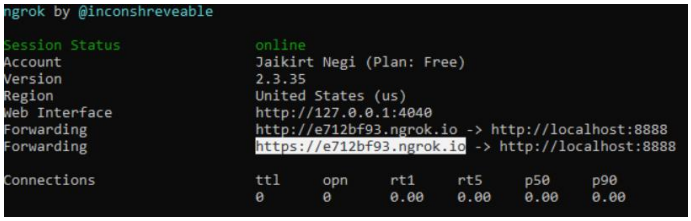


DELL Technologies

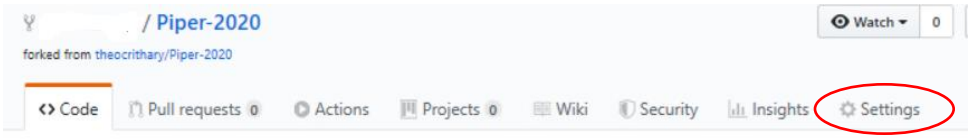
Lab Exercise: Automatically invoke the pipeline when new code is committed to Github

Complete the below steps to demonstrate your understanding of the tools and concepts required for the remaining lab exercises;

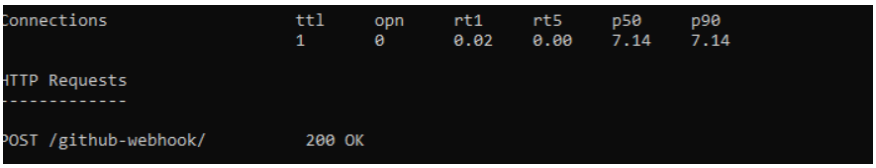
1. Run following commands from command prompt
    a. ngrok http 8888



2. Copy URL as shaded from the command prompt
3. Go to github.com and Piper-2020 repo
4. Click settings



5. Click Webhooks> paste the URL from above
6. https://XXXXXX.ngrok.io/github-webhook/ & content type as application/json
7. Save the settings
8. Create a new file in Piper-2020 repo & save it to trigger Pipeline by SCM polling
9. Verify SCM poll request on ngrok command prompt

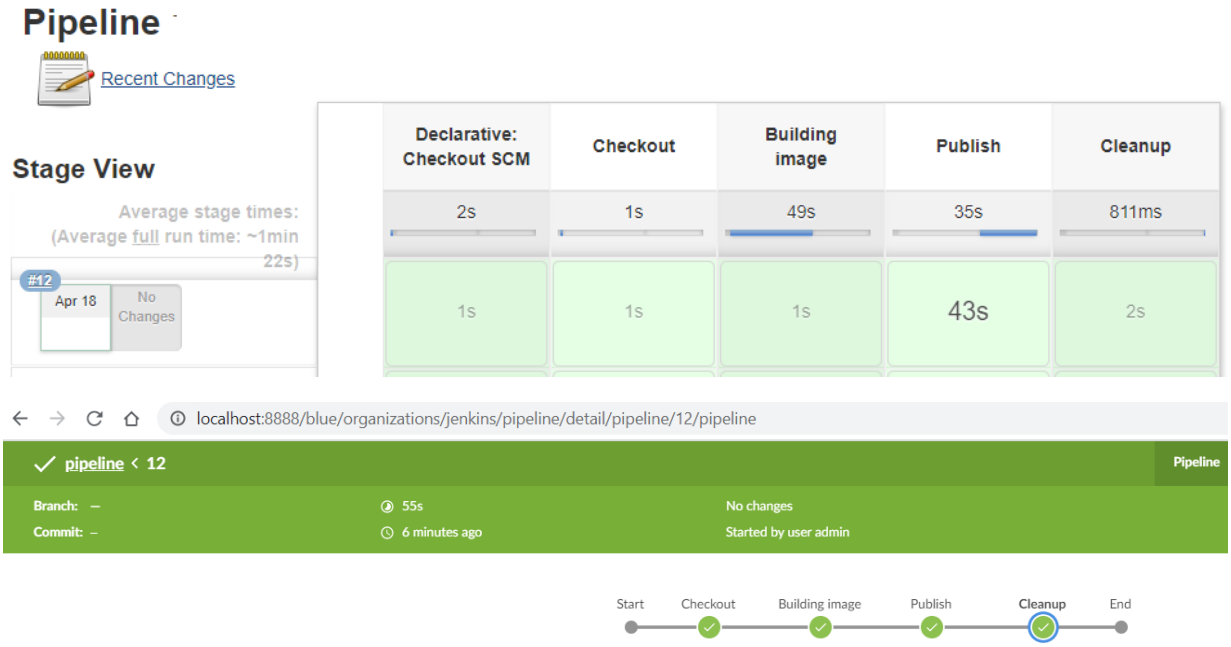


10. Got to Jenkins URL http://localhost:8888 & click pipeline to verify new job initiated via GitSCM polling

15. Console output for pipeline job shows started by github

Retrospective: Built a CI/CD pipeline with an automated trigger using Github & Jenkins

- ❑ Learnt about Jenkins as CI/CD tool
- ❑ Configured Pipeline as a Code
- ❑ Triggered Pipeline job
- ❑ Reviewed the pipeline stages



Now that our container is being built using the latest code whenever a Github commit is performed, we can extend this pipeline to include deployment and testing stages.

Stay tuned for additional enhancements to this lab, or feel free to iterate on this lab example and add your own deployment / testing workflows.