# Ansible F5 Agility 2021 Workshop

## Overview

We make every effort to ensure the lab guide has step by step instructions to complete the tasks, however some topics are not covered in depth due to a lack of time. Throughout the labs we hope to share features, elements, good and bad practices to using Ansible Tower for Automation.

You will be required to modify some files during the course of this workshop. You will not be required to write your own playbooks. The playbooks used are open source and thus free to use and modify. That being said, writing playbooks and running them in a test environment is the best way to learn.

***Outline***

* Part 1: Getting setup
  + Fork Git Repo
  + Connect to Ansible Tower
  + Run a job to spin up your student F5 in AWS
  + Review the Ansible playbooks
  + Connecting to your student F5
* Part 2: Tower lab 1 (Imperative)
  + Connect to Ansible Tower
  + Explore Ansible Tower
  + Create a project
  + Create an inventory
  + Create job templates
  + Create a workflow
  + Add rescue logic to a workflow
* Part 3: Tower lab 2 – AS3 (Declarative)
  + Overview of AS3 Ansible Components
  + Install AS3 plugin
  + Create job templates
  + Create a workflow

**Note: Assume the output shown in the examples below will be different to yours.**

## Part 1: Getting setup

### ***Overview***

* Fork the lab github repository to your own repository so you can edit and modify.

## *Fork the Sirius ansible networking GitHub repository*

### Step 1

## Login in to Github at <https://github.com>

### Step 2

## Go to <https://github.com/mysidlabs/f5-agility-2021-tower-labs>

### Step 3

## Click on the Fork button in upper right.

## Graphical user interface, text, application Description automatically generated

## Note: Once forked you can modify all files within GitHub

*Connecting to Ansible Tower*

### Step 1

## Open your web browser (Chrome or Firefox)

### Step 2

Put the address of the lab Ansible Tower

## https://tower.mysidlabs.com

### Step 3

## Login with your student ID and password provided in the chat window

*Graphical user interface, application

Description automatically generated*

### Step 4

## Navigate to Templates and click the rocket ship icon next to “Student F5 Lab Self Setup”

**Graphical user interface, text, application, email

Description automatically generated**

### Step 5

Click Next

**Graphical user interface, text, application

Description automatically generated**

### Step 6

Enter your student ID number provided in chat and click next. Your student ID will be different than the below screenshot.

**Graphical user interface, text, application

Description automatically generated**

### Step 7

Click Launch

**Graphical user interface, text, application

Description automatically generated**

### Step 8

Wait for a minute or so for the job to run, validate it was successful

Graphical user interface, text, application

Description automatically generated

*Review the Ansible Playbooks*

### Step 1

## Open your web browser (Chrome or Firefox)

### Step 2

Go to <https://github.com/mysidlabs/f5-agility-2021-tower-labs> or your forked version of this repo. Review the folder structure. Go to setup > deploy-student-instance.yml to review the code used to spin up the student F5’s in AWS.

**Tip: Cloud Init is your friend**

<https://clouddocs.f5.com/cloud/public/v1/shared/cloudinit.html#cloudconfig-tmosdecl>

**A picture containing clipart

Description automatically generated**

*Connecting to your F5*

### Step 1

## Open your web browser (Chrome or Firefox)

### Step 2

Put the address of your F5 in the address bar including the port number 8443. NOTE: Your student F5 will have a different number!

## https://siduser155.f5.mysidlabs.com:8443

### Step 3

## Proceed through the warnings regarding the SSL certificate not being valid

## Note: Some browsers will not allow you to connect/bypass the security warning due to settings and the fact that the SSL certificate is a self-signed certificate and cannot be validated. You may have to change browsers in order to bypass the security warning.

### Step 4

Record the F5 Private IP address

A screenshot of a social media post

Description automatically generated

### Step 5

## Login into the F5 BigIP

## Username: admin

## Password: Mys1dlabspw!

## Part 2: Basic F5 labs

### Topology

The topology is simple for the sake of learning some ansible basics. The diagram below is an example, the XXX in the hostname is your Student ID. If you are student 199 then the hostname for the F5 would be siduser199.f5.mysidlabs.com.

Diagram

Description automatically generated

## Part 2: Tower lab 1

Explore Ansible Tower

### Step 1:

Open web browser and go to [https://tower.mysidlabs.com](https://tower.mysidlabs.com/) and enter in your username and password.

A screenshot of a cell phone

Description automatically generated

### Step 2:

Once logged in explore the interface

A screenshot of a computer

Description automatically generated

### Step 3:

Click on the “i” information button on the top right of the user interface.



You will get something similar to the following:

A picture containing bird, flower

Description automatically generated

### Step 4:

Click on the Inventories Tab on the left side of the page

A close up of a logo

Description automatically generated

Click on Network-Lab-Instructor-Inventory



You will see the following

A screenshot of a cell phone

Description automatically generated

### Step 5:

Click on the Projects Tab on the left side of the page. Note that projects point to your source code repository where your playbooks exist.



Click on any of the available project. Below I selected Instructor-ansible-network-labs as an example, but it may not exist when you do this.



You will see the following

A screenshot of a cell phone

Description automatically generated

### Step 6:

Click on the Credentials Tab on the left side of the page

A picture containing knife

Description automatically generated

Type network in the search field and click on the magnifying glass icon

A screenshot of a cell phone

Description automatically generated

Click on network-lab-key



You will see the following

A screenshot of a cell phone

Description automatically generated

### Step 7:

Click on the Templates Tab on the left side of the page

A picture containing table

Description automatically generated

Click on any of the available templates. Below I selected Instructor-SNMP as an example, but it may not exist when you do this.



You will see the following

A screenshot of a computer

Description automatically generated

### Lab 1.1: Create an Ansible Tower job template

### Step 1:

Create a Credential

Click on Credentials section using the left navigation bar.

A close up of a logo

Description automatically generated

Click on the green plus button on the right-hand side to create a new Credential

templates link

Fill out the following fields as follows, and click SAVE

|  |  |
| --- | --- |
| Name | <<siduserID>>.f5.mysidlabs.com |
| Credential Type | Network |
| Username | admin |
| Password | Mys1dlabspw! |

A screenshot of a cell phone

Description automatically generated

### Step 2:

Create a Project

Click on Projects section using the left navigation bar.

A picture containing table

Description automatically generated

Click on the green plus button on the right-hand side

templates link

Fill out the following fields as follows, and click SAVE

|  |  |
| --- | --- |
| Name | <<siduserID>>.f5.mysidlabs.com |
| Organization | sid-org |
| SCM Type | git |
| SCM URL | https://github.com/<<your-github-user>>/f5-agility-2021-tower-labs |
| Clean/Delete on Update/  Update Revision on Launch | Checked |

A screenshot of a cell phone

Description automatically generated

### Step 3:

Create an Inventory

Click on Inventories section using the left navigation bar.

A close up of a logo

Description automatically generated

Click on the green plus button on the right-hand side and click Inventory to create a new inventory

A picture containing clock

Description automatically generated

Fill out the following fields and click SAVE

|  |  |
| --- | --- |
| Name | <<siduserID>>.f5.mysidlabs.com |
| Organization | sid-org |

A screenshot of a cell phone

Description automatically generated

Click on the Sources tab, click the green plus button

A screenshot of a cell phone

Description automatically generated

Fill out the following fields and click SAVE

|  |  |
| --- | --- |
| Name | <<siduserID>>.f5.mysidlabs.com |
| Source | Amazon EC2 |
| Credentials | aws-api |
| Regions | US East Ohio |
| Instance Filters | tag:Name=F5-Lab-<<siduserID>>  ex: tag:Name=F5-Lab-siduser155 |
| Only Group By | Tags |
| Overwrite/Update on Launch | Checked |

A screenshot of a social media post

Description automatically generated

Click on the Sources link at the top left



Click on the green plus button on the right-hand side to add another source

A screenshot of a cell phone

Description automatically generated

Fill out the following fields and click SAVE

|  |  |
| --- | --- |
| Name | <<siduserID>>.web.mysidlabs.com |
| Source | Amazon EC2 |
| Credentials | aws-api |
| Regions | US East Ohio |
| Instance Filters | tag:Name=F5-Lab-Web\* |
| Only Group By | Tags |
| Overwrite/Update on Launch | Checked |

Graphical user interface, application

Description automatically generated

Click on Sources > Sync All

Graphical user interface, text, application, email

Description automatically generated

You should have green clouds next to your sources once sync is complete

Click on the <<siduserID>>.f5.mysidlabs.com link at the top left



Click on Hosts and verify you have 4 hosts

Graphical user interface

Description automatically generated

### Step 4:

Create a Job Template

Click on Inventories section using the left navigation bar.



Click on the green plus button on the right-hand side and click Job Template to create a new inventory

A screenshot of a cell phone

Description automatically generated

Fill out the following fields and click SAVE

|  |  |
| --- | --- |
| Name | <<siduserID>>-create-nodes |
| Job Type | Run |
| Inventory | <<siduserID>>.f5.mysidlabs.com |
| Project | <<siduserID>>.f5.mysidlabs.com |
| Playbook | 1.1-create-nodes.yml |
| Credentials | Select Credential Type: Network  <<siduserID>>.f5.mysidlabs.com |

Graphical user interface

Description automatically generated

Click Save

### Step 5:

Create a Job Template

Click on Inventories section using the left navigation bar.



Click on the green plus button on the right-hand side and click Job Template to create a new inventory

A screenshot of a cell phone

Description automatically generated

Fill out the following fields and click SAVE

|  |  |
| --- | --- |
| Name | <<siduserID>>-create-vip |
| Job Type | Run |
| Inventory | <<siduserID>>.f5.mysidlabs.com |
| Project | <<siduserID>>.f5.mysidlabs.com |
| Playbook | 1.3-create-vip.yml |
| Credentials | Select Credential Type: Network  <<siduserID>>.f5.mysidlabs.com |

Graphical user interface, application

Description automatically generated

### Step 6:

Create a template for cleanup

Create a Job Template

Click on Inventories section using the left navigation bar.



Click on the green plus button on the right-hand side and click Job Template to create a new inventory

A screenshot of a cell phone

Description automatically generated

Fill out the following fields and click SAVE

|  |  |
| --- | --- |
| Name | <<siduserID>>-cleanup |
| Job Type | Run |
| Inventory | <<siduserID>>.f5.mysidlabs.com |
| Project | <<siduserID>>.f5.mysidlabs.com |
| Playbook | 1.4-cleanup.yml |
| Credentials | Select Credential Type: Network  <<siduserID>>.f5.mysidlabs.com |

Graphical user interface

Description automatically generated

### Step 7:

Create a workflow template. The instructors will walk through the steps, so pay attention, but at the end you want your workflow to look like this:

Graphical user interface, text, application

Description automatically generated

Diagram

Description automatically generated

### Step 8:

Create a survey for the workflow

Click add survey and fill out per screen shots below. Note that variable names are case sensitive and use camel case.

Graphical user interface, application

Description automatically generated

Graphical user interface, application, website

Description automatically generated

### Step 9:

Launch the workflow template, fill out the survey (remember what you name your vip and pool, maybe even write it down).

### Step 10:

Verifying that the playbook did what you expected. Login to the F5 with your web browser to see what was configured.

Click on the Local Traffic on the left-hand menu

Click on Virtual Servers.

The Virtual Server will be displayed.

This time it will be Green (Available (Enabled) - The virtual server is available)

Check Pools for app\_pool that both web servers are set to port 80 for their service\_port

If everything looks good from an F5 standpoint, verifying the web servers

Go to:

https://<<siduserID>>.f5.mysidlabs.com/

Each time you refresh the host will change between web1, web2 and web3.

### Step 10:

Cleanup your F5 before running the next lab

Click on templates, search for your user number and select “siduserXXX-clean” job template.

In the extra variables box edit to include your VIP name and pool name like below:

Graphical user interface, text, application, email

Description automatically generated

Click save and launch. Validate that your F5 no longer has nodes, pools and vips.

## Part 4: F5 AS3 labs

Lab 2.0: The way better AS3 method

### Step 1:

**Make sure the BIG-IP configuration is clean!!!**

Click on templates, search for your user number and select “siduserXXX-clean” job template.

In the extra variables box edit to include your VIP name and pool name like below:

Graphical user interface, text, application, email

Description automatically generated

Click save and launch. Validate that your F5 no longer has nodes, pools and vips.

If your AS3 playbook has errors remember what Moss and Roy always do. Did you turn it off and on?

Reboot your F5 > System > Configuration > Device > General

A couple of men looking at a computer

Description automatically generated with low confidence

Graphical user interface, text, application, email

Description automatically generated

### Step 2:

Validate that your F5 has the AS3 package installed. Navigate to iApps > Package Management LX. You should see 2 packages for AS3 installed.

A picture containing graphical user interface

Description automatically generated

Note: AS3 can be downloaded here: <https://github.com/F5Networks/f5-appsvcs-extension/releases>

### Step 3:

Before starting we look at the Playbooks, it is important to understand how AS3 works. AS3 requires a JSON template to be sent as an API call to the F5. You do not need to fully understand every parameter or create these templates from scratch.

Take some time to review the documentation here: <https://clouddocs.f5.com/products/extensions/f5-appsvcs-extension/latest/refguide/understanding-the-json-schema.html>

Examples are located here: <https://clouddocs.f5.com/products/extensions/f5-appsvcs-extension/latest/userguide/examples.html>

And some really good info here on using AWAF with AS3: https://clouddocs.f5.com/training/fas-ansible-workshop-101/3.3-as3-asm.html

### Step 4:

Look at the file called 2.1-as3.yml

Review the output and see what information is in the output.

**connection:** - local tells the Playbook to run locally (rather than SSHing to itself)

**uri:** - this task is calling the uri module

**url: "https://{{ ansible\_host }}:8443/mgmt/shared/appsvcs/declare"** - webURL (API) for AS3

**method: POST** - HTTP method of the request, must be uppercase. Module documentation page has list of all options.

**body: "{{ lookup('template','j2/ as3\_template.j2', split\_lines=False) }}"** - This passed the template as the body for the API request

**status\_code: 200** - A valid, numeric, HTTP status code that signifies success of the request. This can be comma separated list of status codes. In this instance we use 200 which means OK, this is a standard response for successful HTTP requests

Look at the file called AS3\_template.j2

Look at the file called awaf.xml

Step 5:

Create a new job in Ansible Tower to run the AS3 playbook:

Fill out the following fields and click SAVE

|  |  |
| --- | --- |
| Name | <<siduserID>>-AS3 |
| Job Type | Run |
| Inventory | <<siduserID>>.f5.mysidlabs.com |
| Project | <<siduserID>>.f5.mysidlabs.com |
| Playbook | Lab2/2.1-as3.yml |
| Credentials | Select Credential Type: Network  <<siduserID>>.f5.mysidlabs.com |

Graphical user interface

Description automatically generated

Add a survey to the job template

**Question 1**

Prompt: What is the tenant name?

Description: tenant

Answer variable name: tenantName

Answer type: text

Default Answer: workshopExample

**Question 2**

Prompt: What is the user name?

Description: username

Answer variable name: f5UserName

Answer type: text

Default Answer: admin

**Question 3**

Prompt: What is the password?

Description: password

Answer variable name: f5Password

Answer type: password

Default Answer: Mys1dlabspw!

Graphical user interface, application

Description automatically generated

Step 6:

Launch the job, fill out the survey and then login to the F5 with your web browser to see what was configured.

Finally, test and see if you get dancing polar bears.

F5: <https://siduserXXX.f5.mysidlabs.com:8443>

Website: <https://siduserXXX.f5.mysidlabs.com>

Step 7:

Copy or clone your siduserXXX-AS3 job from the previous task. Edit the copy to rename and change the playbook. By copying the previous job template you will be able to use the same survey questions.



|  |  |
| --- | --- |
| Name | <<siduserID>>-AS3-cleanup |
| Job Type | Run |
| Inventory | <<siduserID>>.f5.mysidlabs.com |
| Project | <<siduserID>>.f5.mysidlabs.com |
| Playbook | Lab2/2.2-as3-delete.yml |
| Credentials | Select Credential Type: Network  <<siduserID>>.f5.mysidlabs.com |

Graphical user interface

Description automatically generated

Run the job and verify your Big IP configuration is clean

## Success - Congratulations.

## Picture 2095273455

**Appendix A:**

Useful resource links and information

Links:

Ansible Best Practices

<https://docs.ansible.com/ansible/latest/user_guide/playbooks_best_practices.html>

Ansible Network troubleshooting

<https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html>

Ansible cli\_command module information

https://www.ansible.com/blog/deep-dive-on-cli-command-for-network-automation

Variable precedence

<https://docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html#variable-precedence-where-should-i-put-a-variable>

Additional Notes:

* Remember YAML is very sensitive to correct indentation
* **Hostvars** allow us to access meta-data about our inventory hosts.
* The use of an Ansible role is best practice when there is a well-defined scope with a high possibility of re-use.
* If you copy and paste text for a playbook you may get indentation issues. Ansible provides a simple syntax checker, try ansible-playbook --syntax-check backup.yml to verify. A Best Practice is to use a linter, for example ansible-review. Ansible provides excellent online documentation, which is also available from the command line, for example ansible-doc ios\_config. For a full list of modules try ansible-doc –l
* There where multiple ways of implementing a playbook where specific tasks or groups of tasks execute against specific hosts. For example, we could have used 1 playbook for configuring every router in the lab utilizing the “when:” statement to ensure specific tasks are only applied to a specific router. Although this is not necessarily following best practices.
* The use of handlers: which can be used in any playbook. A handler is a special way of calling a task whenever an action needs to be taken after a previous task. For example, both installing and configuring an application may require a restart. A handler would be notified by both tasks but would only run once when the playbook finishes.