Title: Demystifying Cisco FSO Stack APIs

Session ID: DEVWKS-2678

Speakers: Anuj Modi

Learning Objectives or Table of Contents [Heading 1]

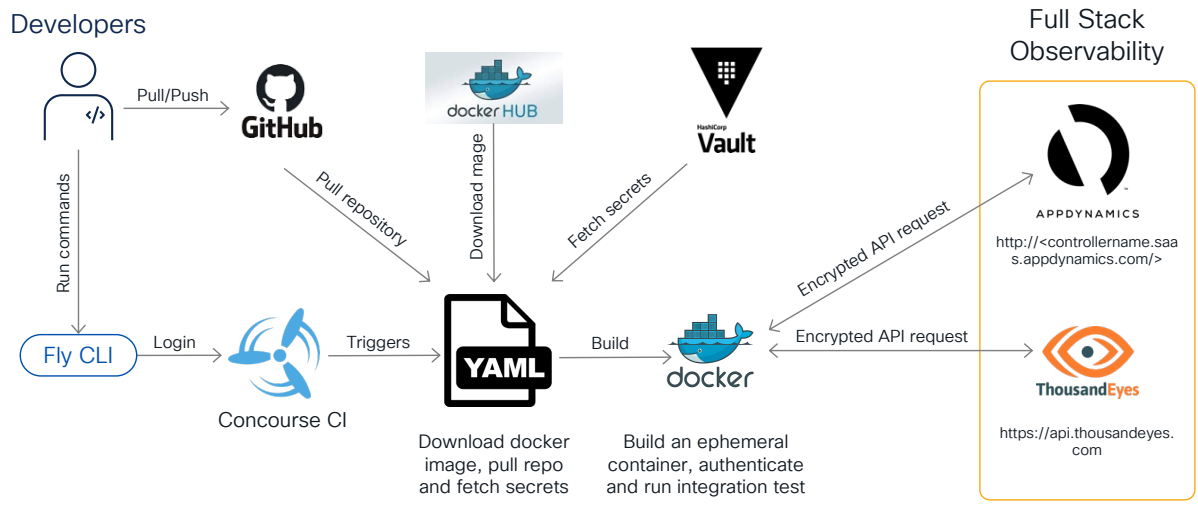
Upon completion of this lab, you will be able to:

- Cloning the FSO API repo and creating own git branch to make the changes in the code.
- Learn how to login into Concourse CI through fly and trigger the tasks in the automated pipeline.
- Run the AppDynamics and ThousandEyes tasks to build an ephemeral container using the docker image, fetch API tokens and SSH key from Vault to run the API tests.

Scenario

The Cisco FSO API lab guide invoke the API calls in most secure and automated way through integration with Hashicorp Vault and Concourse CI pipeline. The lab is deployed in Kubernetes cluster on the AWS where students can run the tasks using an ephemeral build container which automatically fetch the tokens, ssh keys and credentials from Vault. The Concourse CI pipeline provide the automation for all the tasks in the lab and provide fly cli to trigger these tasks in yaml format. This yaml file consist of resources, job, tasks, shell and python scripts. The python scripts in the container leverages these credentials to communicate with Cisco AppDynamics and ThousandEyes to get the desired information from these products. The students can watch the progress on Concourse UI for jobs and tasks.

Developers workflow



Task 1: Clone the repository in the laptop

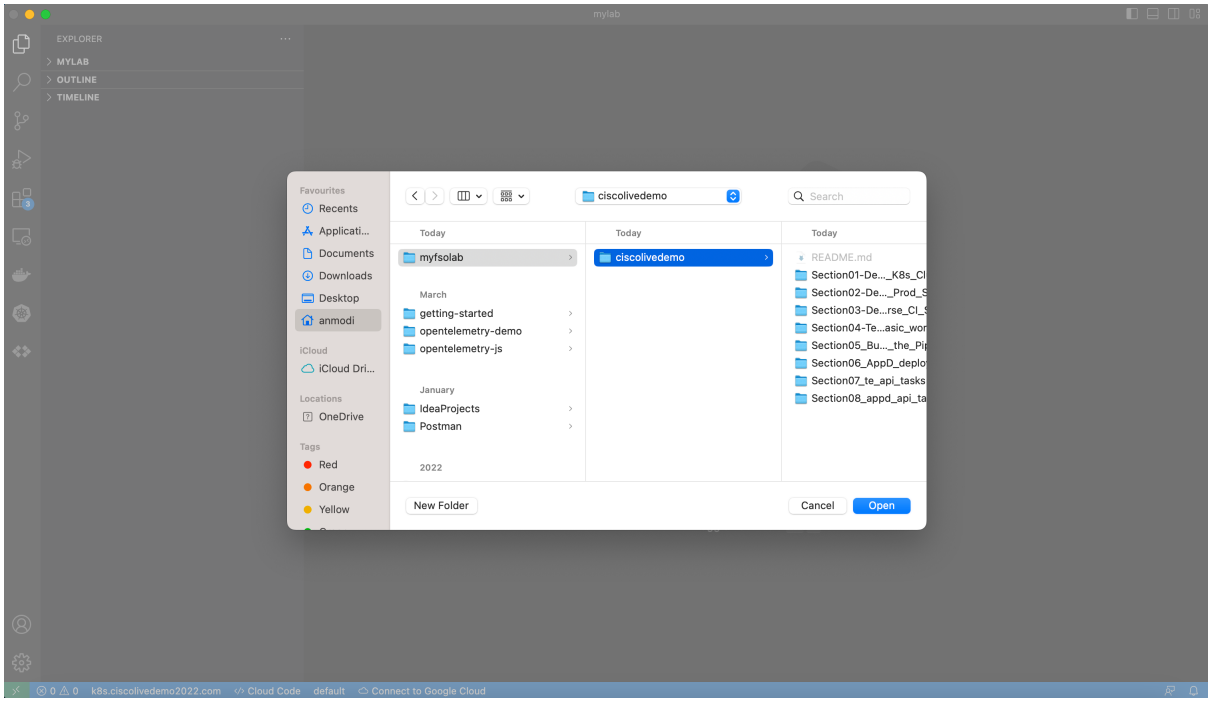
Step 1: Create a local directory and clone the repo.

Create a Directory on your local machine and change into the <dir name> directory you created and then clone the repo into the directory: https://github.com/anujmodi1/ciscolivedemo.git

#mkdir -p <dir name>
#cd <dir name>
#git clone https://github.com/anujmodi1/ciscolivedemo.git
#cd ciscolivedemo

```
anmodi@ANMODI-M-L56Q ~ % mkdir -p myfsolab
anmodi@ANMODI-M-L56Q ~ % cd myfsolab
anmodi@ANMODI-M-L56Q myfsolab % git clone https://github.com/anujmodi1/ciscolivedemo.git
Cloning into 'ciscolivedemo'...
remote: Enumerating objects: 1951, done.
remote: Counting objects: 100% (460/460), done.
remote: Compressing objects: 100% (331/331), done.
remote: Total 1951 (delta 154), reused 398 (delta 111), pack-reused 1491
Receiving objects: 100% (1951/1951), 188.36 MiB | 8.19 MiB/s, done.
Resolving deltas: 100% (984/984), done.
anmodi@ANMODI-M-L56Q myfsolab % cd ciscolivedemo
anmodi@ANMODI-M-L56Q ciscolivedemo %
```

Open the ciscolivedemo folder with Visual Studio.



Go to the terminal in Visual Studio, check if you are into ciscolivedemo folder.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

anmodi@ANMODI-M-L56Q ciscolivedemo % pwd
/Users/anmodi/myfsolab/ciscolivedemo
anmodi@ANMODI-M-L56Q ciscolivedemo %
```

Check the git status command, you will in main branch.

```
anmodi@ANMODI-M-L56Q ciscolivedemo % git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   .DS_Store

no changes added to commit (use "git add" and/or "git commit -a")
anmodi@ANMODI-M-L56Q ciscolivedemo %
```

Next step, you create your own branch <your name>
#git checkout -b <your name>

```
anmodi@ANMODI-M-L56Q ciscolivedemo % git checkout -b anuj
Switched to a new branch 'anuj'
```

Double-Check/Verify you are in the correct branch and fetch all the repo.
#git status
#git fetch --all

```
anmodi@ANMODI-M-L56Q ciscolivedemo % git status
On branch anuj
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   .DS_Store

no changes added to commit (use "git add" and/or "git commit -a")
anmodi@ANMODI-M-L56Q ciscolivedemo % git fetch --all
anmodi@ANMODI-M-L56Q ciscolivedemo %
```

Task 2: Login into Concourse via fly Cli

Step 1: Login into concourse via fly cli

Login via fly with the username and password assigned to you by your Lab Instructor

fly --target=target login --concourse-url=http://dev-ci.ciscolivedemo2022.com:8080 -n main --username=ci --password=xxxx

```
anmodi@ANMODI-M-L56Q ciscolivedemo % fly --target=target login --concourse-url=http://dev-ci.ciscolivedemo2022.com:8080 -n main --username=ci
logging in to team 'main'

target saved
anmodi@ANMODI-M-L56Q ciscolivedemo %
```

Task 3: Get the list of enterprise Agents ThousandEyes

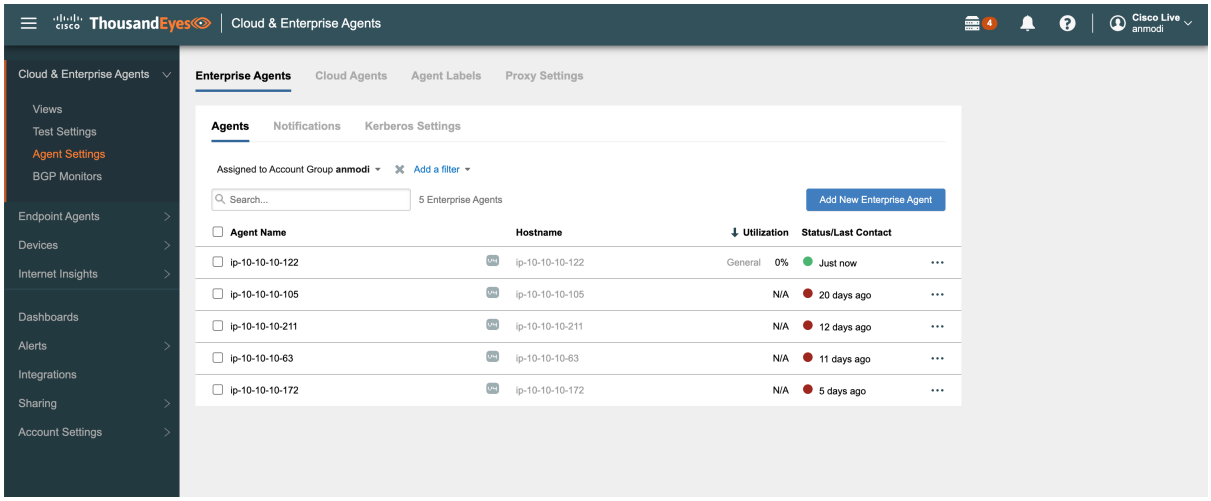
Step 1: Run the lab tasks to get the list of enterprise agents from ThousandEyes

Understand that the Instructor has already generated an API key from the GUI and entered it into the vault.

Login into thousandeyes dashboard using following credentials.

https://app.thousandeyes.com/login
Username: ciscolivedemo2022@gmail.com
Password: xxxxx

Verify you have atleast one agent is running on the TE portal.



To run this task, make sure you are in the input directory:
cd /Users/anmodi/dev/ciscolivedemo/Section07_te_api_tasks/input

```
anmodi@ANMODI-M-L56Q input % cd /Users/anmodi/dev/ciscolivedemo/Section07_te_api_tasks/input
```

Run the task and verify you can authenticate to and get a valid json response from the ThousandEyes API:
fly -t target e -c te_api_task_1.yml

```
anmodi@ANMODI-M-L56Q input % fly -t target e -c te_api_task_1.yml
uploading input done
executing build 32334 at http://dev-ci.ciscolivedemo2022.com:8080/builds/32334
initializing
initializing check: image
selected worker: dev-ci-worker-0
selected worker: dev-ci-worker-0
INFO: found existing resource cache

selected worker: dev-ci-worker-0
running /bin/sh -ce pwd
cd input
chmod a+x te_api.sh
./te_api.sh
```

You will get the enterprise agent id list output like below

```
WARNING! The VAULT_TOKEN environment variable is set! This takes precedence
over the value set by this command. To use the value set by this command,
unset the VAULT_TOKEN environment variable or set it to the token displayed
below.

<Response [200]>
973641
986261
987651
991266
997386
999591
[{'agentId': 973641}, {'agentId': 986261}, {'agentId': 987651}, {'agentId': 991266}, {'agentId': 997386}, {'agentId': 999591}]
<Response [400]>
succeeded
anmodi@ANMODI-M-L56Q input %
```

Task 4: Get the list of business transactions from AppDynamics

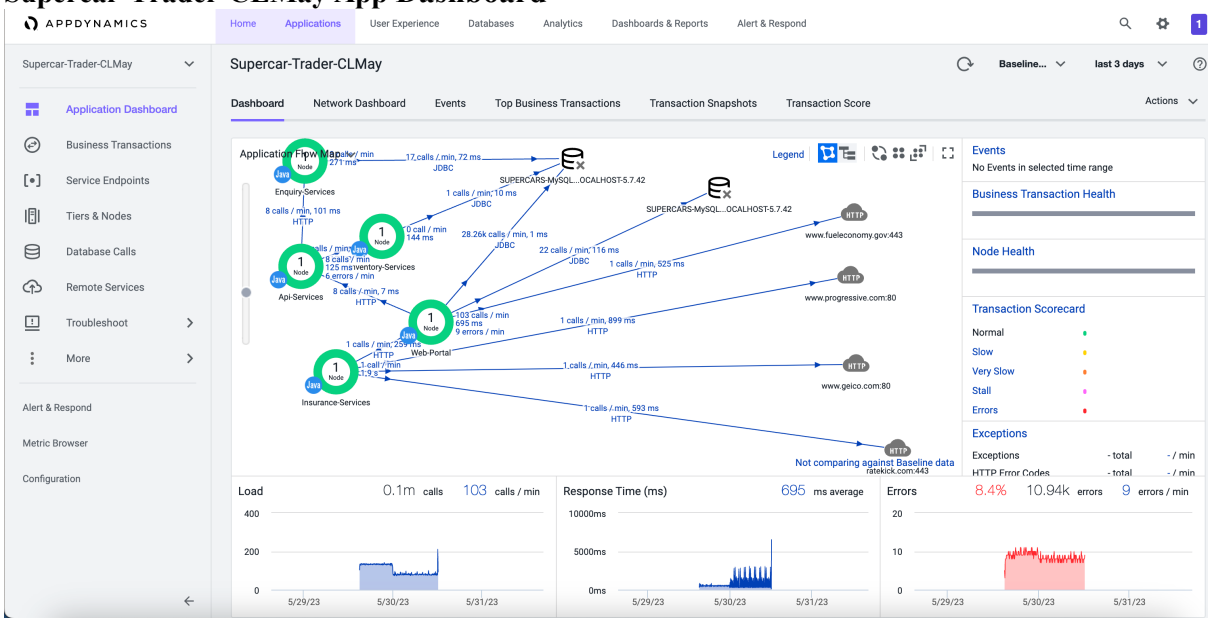
Step 1: Run the lab tasks to get the list of business transactions from AppDynamics

Understand that the Instructor has already generated an API key from the GUI and entered it into the vault

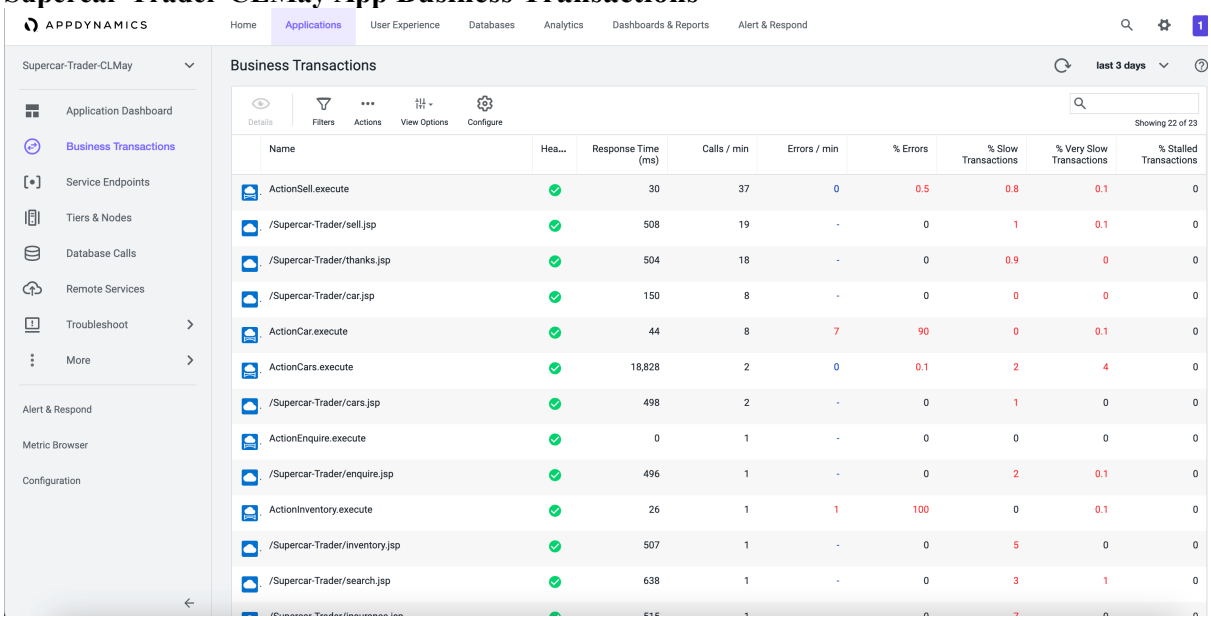
Login to Appdynamic Dashboard and check all the business transactions for **Supercar-Trader-CLMay** App.

https://kickstarter.saas.appdynamics.com/controller/
Account Name: kickstarter
Username: ciscolivedemo2022@gmail.com
Password: xxxxx

Supercar-Trader-CLMay App Dashboard



Supercar-Trader-CLMay App Business Transactions



To run this task, make sure you are in the input directory:
cd /Users/anmodi/dev/ciscolivedemo/Section08_appd_api_tasks/input

```
anmodi@ANMODI-M-L56Q input % cd /Users/anmodi/dev/ciscolivedemo/Section08_appd_api_tasks/input
anmodi@ANMODI-M-L56Q input %
```

Run the task and verify you can authenticate to and get a valid json response from the AppD API:

fly -t target e -c appd_get_token_task.yml

```
anmodi@ANMODI-M-L56Q input % fly -t target e -c appd_get_token_task.yml
uploading input done
executing build 32344 at http://dev-ci.ciscolivedemo2022.com:8080/builds/32344
initializing
initializing check: image
selected worker: dev-ci-worker-0
selected worker: dev-ci-worker-0
INFO: found existing resource cache

selected worker: dev-ci-worker-1
running /bin/sh -ce cd input
chmod a+x appd_get_token.sh
./appd_get_token.sh
```

This authenticates to the AppD Api and generates a temporary oath token and writes oauth token to the vault.

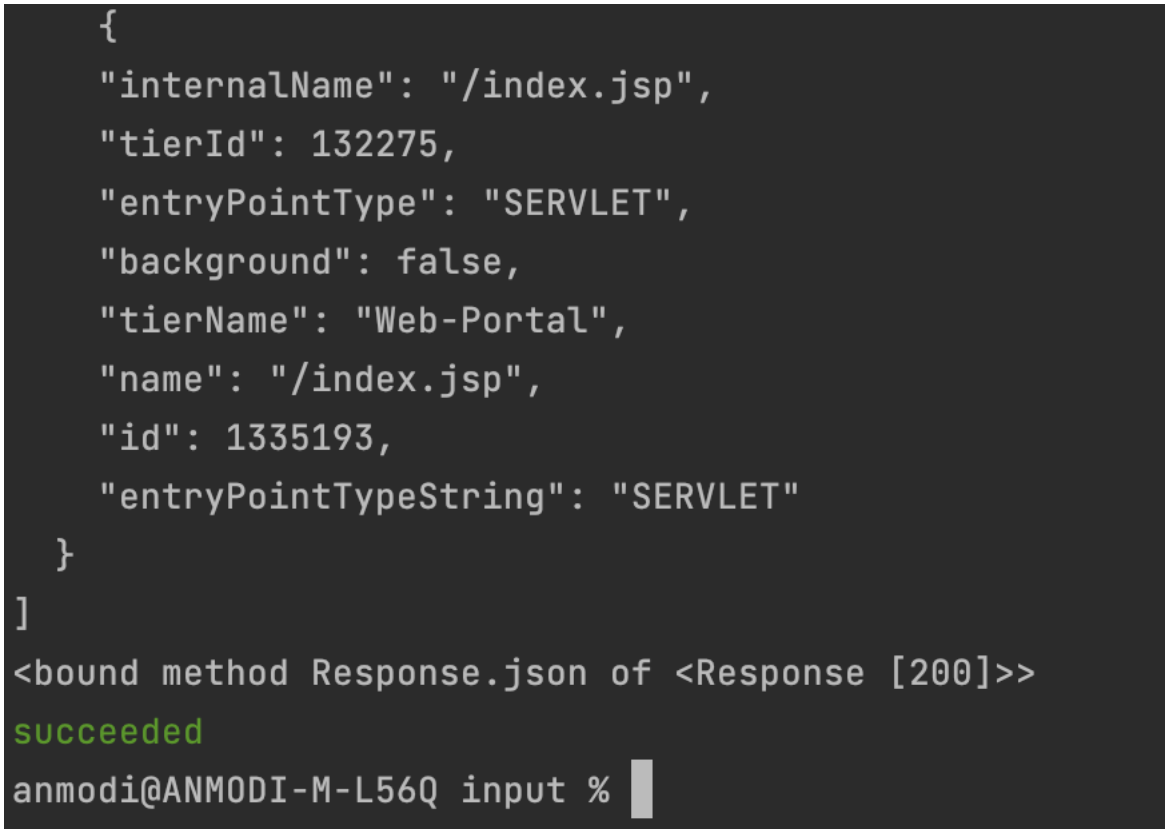
You should be getting 20x succeeded message

```
This it the temp oath token generated for appd
204
succeeded
anmodi@ANMODI-M-L56Q input %
```

fly -t target e -c appd_use_token_task.yml

```
anmodi@ANMODI-M-L56Q input % fly -t target e -c appd_use_token_task.yml
uploading input done
executing build 32348 at http://dev-ci.ciscolivedemo2022.com:8080/builds/32348
initializing
initializing check: image
selected worker: dev-ci-worker-0
selected worker: dev-ci-worker-0
INFO: found existing resource cache

selected worker: dev-ci-worker-0
running /bin/sh ./input/appd_use_token.sh
```



This task uses the temp oath token generated in the previous tasks from Vault and perform an API call that returns json from the AppD API

You should see list of business transactions for Supercar-Trader-CLMay Application.