Cisco FSO Labs

Two Tracks:

1. Developing Code to FSO APIs Fastrack

In this track, students will be enabled to immediately start developing their own code to our APIS with the use of Concourse Ephemeral OCI Build Containers and Tasks

Code has already been developed to handle the automation of programmatic authentication to the FSO APIs and is guaranteed to work in any environment via the curated OCI build container images. This means all the students need to do to develop code is run tasks and update URL strings and work with the returned JSON from the API.

Documentation that explains how to work through each module of the lab is contained in a README.md in each lab-tasks subdirectory.

Once the student clones the git repo and checks out their assigned branch, they will have all the documentation and code to run the lab self paced, however it is recommended to attend the in person training to “write code” live in a code pairing manner with the Instructor to work through the lab challenges.

https://github.com/devops-ontap/cisco-fso-labs

The Students will not need to install anything to run the lab except git and fly. Students can participate in the lab from any operating system.

All Documentation and Steps are contained in the Docs folder and the respective lab-tasks for the lab are contained in the lab repo here:

<https://github.com/devops-ontap/cisco-fso-labs/tree/main/Docs>

Each lab task module takes approximately 5 min to explain and demo and then the students will be allocated 5-10 min to repeat the steps that the Instructor has demo’d (also documented in the README.md)

The Students will be encouraged to perform trial and error code development so they can feel comfortable in failing in their code. It is actually recommended to write the code to fail first. When we fail our code, we are actually writing in tests for our code and following the best practice in regards to standard Software Development Lifecyle processes.

Students will practice rapid iteration(rapid development) of code in very small chunks and test their code in tasks ( building their code in the ephemeral build containers).

The lab teaches Students the best practice in regard to maintaining any variable data: variables, config files, ssh-keys, tokens, api keys, passwords, secrets etc in a secure vault. We are using Hashicorp Vault as it is free and as Cisco.

The lab teaches students NOT how to run one command against an API to do one unit of work using Postman, RATHER it teaches students how to automate bulk operations so they can do thousands of things at once time…..which meets the needs of the real world of DEVOPS.

Sample Recordings…..

Webex meeting recording: Automation - Programmatically Auth to FSO APIs

Password: gQSPhrT9

Recording link: <https://cisco.webex.com/cisco/ldr.php?RCID=6c4ca1edec6eafe4f3e88b9e57961a2f>

Webex meeting recording: Devops - Automate Kubernetes Cluster Deploys to AWS

Password: MrmshGd7

Recording link: <https://cisco.webex.com/cisco/ldr.php?RCID=993c4247b9057d6babcec9249f07595f>

Webex meeting recording: AppD API Auth Kickstart - Vault Integration via OCI Build Container

Password: cRUZpCJ6

Recording link: <https://cisco.webex.com/cisco/ldr.php?RCID=670a35e97ba392bb7567f0e5e37f11b1>

Webex meeting recording: DEVOPS Way of Working - Coding in Build Containers

Password: iXGa82UP

Recording link: <https://cisco.webex.com/cisco/ldr.php?RCID=f22c97bcadbd07fe1fdc52ce4b4d5372>

Webex meeting recording: Lab Day 1 - ThousandEyes-Rapid Iteration - Developing Code using Tasks and Params

Password: MhRrnPQ5

Recording link: <https://cisco.webex.com/cisco/ldr.php?RCID=0a497284f3785953e7c886e8d347060b>

Webex meeting recording: Quick Tips - OCI Ephemeral Build Containers in Concourse

Password: 6Sh6m9CE

Recording link: <https://cisco.webex.com/cisco/ldr.php?RCID=a0d08a622aec2d15339aa0d5db6318d5>

1. Automatic Deployment of the entire lab underlying infrastructure to a Cloud Provider(AWS) using a pipeline

For Partner Students who like this lab, they want to deploy the underlying lab infrastructure to demo or teach their customers – they can deploy the entire lab environment to their own Cloud Provider environment via a pipeline deployment.

The code is in the same repo.