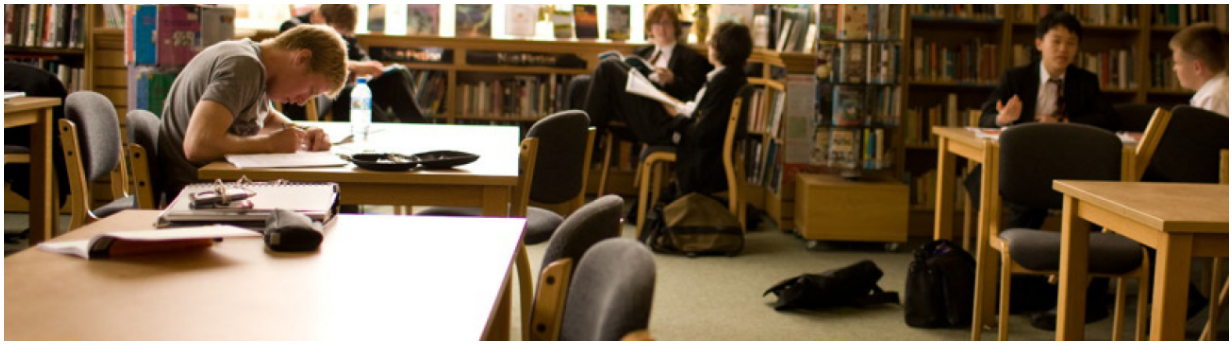
Cisco Intercloud Fabric Provider Platform Test Harness, Release 2.3.1



2015-11-13

Contents

Cisco ICFPP Test Harness

The Cisco Intercloud Fabric Provider Platform (ICFPP) test harness is a test suite that enables you to perform API-level testing of Cisco ICFPP. The test harness is a Python-based REST client that simulates the northbound interface for Cisco ICFPP. You can use the APIs to validate Cisco ICFPP functionality without an actual northbound interface.

Requirements

The Cisco ICFPP test harness requires:

- A Linux system
- Python 2.7.x installed on the Cisco ICFPP target system
- Access to the Cisco ICFPP URL via HTTPS

Note

The Cisco ICFPP test harness is not supported on Windows systems.

Installing the Test Harness

The test harness package contains the following files:

File	Description
userInput.txt	Configuration file
icpep_rclient.py	Base implementation of Cisco ICFPP REST APIs
icfpp_func_tests.py	Implementation file that executes major Cisco ICFPP operations
parallel_tests.py	Implementation file that executes parallel operations for stress tests
tenant_conninfo.py	Helper file
vmops-main.py	Main test file to execute
wrapper_rc.py	Wrapper state machine implementation
xmltodict.py	Helper file

To install the test harness, copy the test harness package to the Linux system and extract the contents using the **tar** utility.

Configuring the Test Harness

Configure the test harness by modifying the userInput.txt file as described in the following table.

For more information about the tenant-provisioning fields, see the Cisco ICFPP architecture and northbound APIs sections in the *Cisco Intercloud Fabric Provider Platform Administrator Guide*.

Field	Description
targetIp	The Cisco ICFPP IP address. In a large-scale setup with a load balancer, use the load balancer IP address or URI. If there is no load balancer, use the primary Cisco ICFPP IP address.
cloudIp	Back-end cloud infrastructure IP address or URI.
cloudInstanceName	Unique instance name for the cloud being provisioned.
moduleName	The adapter being used.
cloud_param_name	Any parameter that must be passed for the specified cloud instance. For example, for some back ends, the service provider must provide an FTPS name that must be specified. If you are not sure, leave the parameter empty.
tenantName	Name of the tenant to be provisioned.
userName	User associated with the tenant.
paword	Password for the user.

Field	Description
apikey	API key. Used only for providers that do not use username and password credentials.
secretkey	Secret key. Used only for providers that do not use username and password credentials.
orgName	Required for VMware vCloud Director. Optional for other cloud providers.
vpcName	Name of the virtual private cloud (VPC) for provisioning the virtual machine (VM). This field is ignored if the providerVpcManagement cloud capability indicates CREATE and an automatically-generated VPC name is used.
networkName	Name of the network in which the VM is to be instantiated. This field is ignored if providerNetworkManagement cloud capability indicates CREATE an automatically-generated network name is used.
The following information pertains to images to be uploaded into Cisco ICFPP. The information is mandatory for testing template functionality. The imageInfo section (included below) is in JavaScript Object Notation (JSON) format.	

	multiLineProperty imageInfo = [
	{ "Basic": Any name that associates the Image
	{
	# Enabled for uploads to cloud
	"enabled" : 1,
	# Name of file
	"filename" : "basic-image.ova",
	# Directory path to where the file is present
	"dirname" : "/home/test/harness",
	"type" : "VMDK",
	"serverType" : "APPLICATION",
	#Info about Servers to be created from this image
	"serverInfo" :
	{
	#Number of servers to create from this image
	"count" : 1,
	"cpu" : "1",
	"memory" : "2048",
	"publicIP" : "false"
	}
	},
	"ICS":
	{
	"enabled" : 1,
	"filename" : "ics-test.ova",
	"dirname" : "/home/test/harness",
	"type" : "VMDK",
	"serverType" : "INFRA_ICS",
	#Info about Servers to be created from this image
	"serverInfo" :
	{
	#Number of servers to create from this image
	"count" : 1,
	"cpu" : "1",
	"memory" : "2048",

Field	Description
	"publicIP" : "true" } } }]
imageAlgorithmVersion	Specify V2.
imageTransferBlockSize	Specify 524288.
destPath	Destination directory in which the downloaded files reside. This directory must be as large as the file being tested.

Test Harness API Details

The test harness APIs are organized as follows:

- Cloud provisioning and tenant configuration APIs

  The cloud and tenant configuration and query APIs, which the test harness exercises, are available only to a service provider administrator. For the complete set of APIs, see the *Cisco Intercloud Fabric Provider Platform Administrator Guide*.

- VM life cycle preparation APIs

  These APIs are used to discover details about the back-end platform and perform any additional configuration required for VM life cycle operations. These APIs are used by a tenant administrator and simulate Cisco Intercloud Fabric Director. These APIs can discover locations, capabilities, and VPC capabilities of the back-end cloud platform. These APIs can also be used to create networks in the back end that are used later for VM operations.

- VM life cycle APIs

  These APIs are used to upload images and templates, and to create VMs using the uploaded templates. You can also use these APIs to perform operations such as start, stop, reboot, and delete on the templates and VMs.

Preparing to Use the Test Harness

Before using the test harness, complete the following tasks:

1. Install and configure Cisco ICFPP for use.
2. Confirm that Python 2.7 is being used on the test harness machine. If Cisco ICFPP is used as the test harness, confirm that Python 2.7 is located at /usr/local/bin/python2.7.
3. Ensure that the back-end cloud platform is accessible from Cisco ICFPP.
4. Set up any required resources on the cloud.

Using the Test Harness

A Linux system with Python 2.7.x is required to use the test harness.

Note

- *The Cisco ICFPP test harness is not supported on Windows systems.*
- *If desired, you can use Cisco ICFPP itself as the test harness machine. If you do so, in the userInput.txt file, change the targetIp field to localhost (127.0.0.1) before performing the following steps.*

To run the test harness, perform the following steps:

1. Copy the test harness package into a directory.
2. Modify the userInput.txt configuration file using the setup information. For more information about the userInput.txt file, see Configuring the Test Harness.

3. Run the test harness code by entering the following command:

   ./vmops-main.py –i userInput.txt

   By default, this command runs all tests that are defined in the file vmops_main.py. You can customize this file to disable and enable a set of tests.

   After the tests are run, the tests and responses from Cisco ICFPP are logged in a file

4. To view test results, navigate to the directory from which vmops-main.py was executed. The tests and their results are logged in this directory with the filename logtest*yyyy-mm-dd*T*hh*:*mm*:*ss*.*nnnnnn*, where the values are based on the date and time of the tests, such as logtest2015-07-10T16:00:23.377981.