



Fundamentals of Network Automation Using PYTHON

Course Intro

ine.com





Rohit Pardasani

CCIEx5 #21282, CCSI #34999



rpardasani@ine.com



[@RohitPardasani](https://twitter.com/RohitPardasani)



linkedin.com/in/rohit21282



CCIE Routing & Switching

CCIE Security

CCIE Voice

CCIE Service Provider

CCIE Collaboration

What Will You Learn?

- Setting up the LAB
- Installing PYTHON3, NETMIKO, ATOM and VIM
- Print Command
- Numbers and Math
- Variables
- Built-in Methods
- Format String
- Lists
- Dictionaries
- Nested Objects
- Installing Operating System Modules
- Reading, Writing , and Working with Files
- WITH Statement
- Functions
- Conditionals
- For Loop
- Connecting to Cisco Routers and Executing Commands using Netmiko



LAB Setup

ine.com

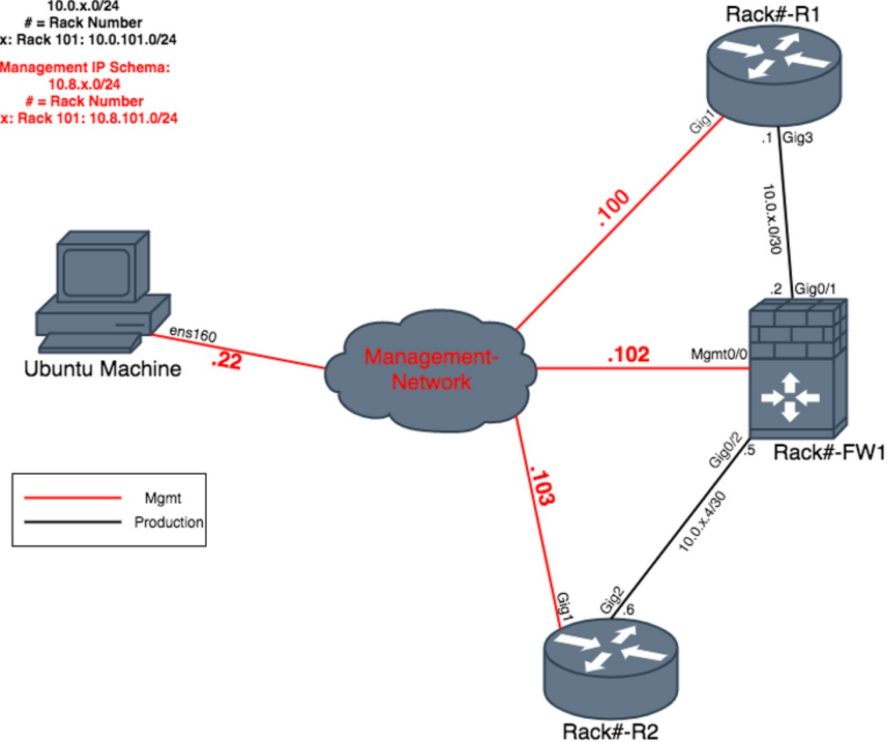


Diagram

Lab Topology:  racks@unifiednetworklearning.com

Production IP Schema:
10.0.x.0/24
= Rack Number
Ex: Rack 101: 10.0.101.0/24

Management IP Schema:
10.8.x.0/24
= Rack Number
Ex: Rack 101: 10.8.101.0/24



How to Access Rack

Connect via AnyConnect to 198.37.120.237

****Note:** If you do not have the AnyConnect client, navigate via Chrome or Firefox to 198.37.120.238 and download the client.

SSH to the Ubuntu client:

10.8.#.22 (Where # is Rack number)

Ex: Rack101 → 10.8.101.22

Username: rack#-ubuntu (Where # is Rack number)

Password: cisco

SSH to the Cisco Devices:

R1: 10.8.#.100 (Where # is Rack number)

R2: 10.8.#.103 (Where # is Rack number)

FW1: 10.8.#.102 (Where # is Rack number)

Username: cisco Password: cisco

Use this Ubuntu machine for python Interpreter, Cisco devices are also connected to this subnet and will be used in later labs.





Using the Print command - Lab Task

Lab Task

- Create a file named Lab2.py on Atom.
- This file must print the following:
 - show ip interface brief
 - show ip route | include 192.168.1.0
 - show interface status
 - This lab belongs to Rohit's class
 - The password is "cisco"



Numbers and Math with Python - Lab Task

Lab Task

- Create a file named Lab3.py on Atom.
- This file must print the following:

```
How much is 3 + 2?  
5  
If 1 Dollar is 75 Rupees, then how much is 10 Dollars in Rupees?  
750  
What is 3 + 2 + 5 - 2 * 4?  
The answer is 2  
What is 4 + 2 + 5 * 2 / 4 + 2? 10.5  
Is 3 + 2 = 5?  
True  
Is 3 + 3 = 5?  
False  
Is 3 + 2 >= 5?  
True  
Is 3 + 3 <= 5?  
False
```



Creating, Verifying, and Printing Variables – Lab Task

Lab Task

- Create a file named Lab4.py on Atom.
- Create a variable of class string named "ccie1" with value as "CCIE Enterprise Infrastructure"
- Create a variable of class string named "ccie2" with value as "CCIE Service Provider"
- Create a variable of class string named "ccie3" with value as "CCIE Security"
- Create a variable of class string named "ccie4" with value as "CCIE Collaboration"
- Create a variable of class int named "ei" with value as "40000"
- Create a variable of class int named "sp" with value as "10000"
- Create a variable of class int named "sec" with value as "15000"
- Create a format string variable named "total_ei" with value as "There are 40000 CCIE Enterprise Infrastructure certified people in the world"
- Create a format string variable named "total_sp" with value as "There are 10000 CCIE Service Provider certified people in the world"
- Create a format string variable named "total_sec" with value as "There are 15000 CCIE Security certified people in the world"
- Print total_ei, total_sp and total_sec in separate lines.
- Print "The total number of CCIE's in the world = 65000" (This value must come from the format string variable)



Creating, Verifying, and Printing Built-in Methods – Lab Task

Lab Task

- Create a file named Lab5.py on Atom.
- Create a variable of class string named "ccie" with value as "CCIE Enterprise Infrastructure\n CCIE Service Provider\n CCIE Security"
- Create a variable of class string named "ip_address1" with value as "10.1.2.100"
- Create a variable of class string named "ip_address2" with value as "10.3.4.200"
- Store the contents of the variable "ccie" in a new variable called as "lower_ccie". The value must be in lower-case.
- Split the contents of the variable "ip_address1" into a list called "new_ip_address1". Use the split built-in method.
- Split the contents of the variable "ip_address2" into a list called "new_ip_address2". Use the split built-in method.
- Split the contents of the variable "lower_ccie" into a list called "new_lower_ccie". Use the splitlines built-in method.
- Print the third octet of "new_ip_address1".
- Print the fourth octet of "new_ip_address2".
- Print "new_lower_ccie".



Creating, Verifying, and Printing Format and Replace Built-in Method – Lab Task

Lab Task

- Create a file named Lab6.py on Atom.
- Create a variable of class string named "ip_address1" with value as "10.1.5.5"
- Use the replace built-in method to replace "5" with "2" and store this in a new variable called as "temp_ip_address".
- Print "temp_ip_address".
- Use the replace built-in method to replace the first occurrence of "2" with "100" and store this in a new variable called as "new_ip_address".
- Create a new variable called as "csr_ip_address" with value as "The IP address of the gateway router is {}".
- Print "csr_ip_address". This should print " The IP address of the gateway router is 10.1.100.2"



Creating, Verifying, and Printing Lists – Lab Tasks

Lab Task

- Create a file named Lab7.py on Atom.
- Create a variable of class string named "ip_address" with value as "10.1.5.5"
- Create a variable of class string named "interface" with value as "G0/0/0/0"
- Create a variable of class list named "List" with index value 0 set as the variable "interface" and index value 1 set as the variable ip_address and index value 2 set as "description connected via Python" and index value 3 set as "shut".
- Print each index value in a separate line.
- Print List.
- Modify List index value 3 with "no shut"
- Print "The IP address of the router is 10.1.5.5 and the management interface is G0/0/0/0". Ensure the IP address and interface name comes from the list.



Creating, Verifying, and Printing Dictionaries – Lab Tasks

Lab Task

- Create a file named Lab8.py on Atom.
- Create a variable of class dict named “info1” with key and value set as:
 - hostname = R1
 - mgmt-ip = 10.1.1.1
 - username = rohit
 - password = cisco
- Create a variable of class dict named “info2” with key and value set as:
 - hostname = R2
 - mgmt-ip = 10.1.1.2
 - username = rohit
 - password = cisco
- Create a variable of class list named “info” with index value 0 set as a info1 and index value 1 set as info2.
- Print info.



Creating, Verifying, and Pretty Printing Nested Objects – Lab Task

Lab Task

- Create a file named Lab9.py on Atom.
- Create a variable of class dict named “dict1” with key and value set as:
 - hostname = R1
 - mgmt-ip = 10.1.1.1
 - username = rohit
 - password = cisco
- Create a variable of class dict named “dict2” with key and value set as:
 - hostname = R2
 - mgmt-ip = 10.1.1.2
 - username = rohit
 - password = cisco
- Create a variable of class dict named “interfaces_r1” with key and value set as:
 - interface = G1
 - ip_address = 192.168.1.1
- Create a variable of class dict named “interfaces_r2” with key and value set as:
 - interface = G1
 - ip_address = 192.168.1.2
- Create a variable of class list named “data_center” with index value 0 set as a dict1 and index value 1 set as dict2. Ensure the variable interfaces_r1 is merged with dict1 and interfaces_r2 is merged with dict2 within the list data_center.
- Import JSON and pretty print IP address of R1
- Import JSON and pretty print data_center



Reading, Writing, and Printing Files – Lab Task

ine.com

Lab Task

- Create a variable of class list named “vlans” which contains dictionary items with key and value set as:
 - {‘id’: ‘10’, ‘name’: ‘DATA’}
 - {‘id’: ‘20’, ‘name’: ‘VOICE’}
 - {‘id’: ‘30’, ‘name’: ‘MGMT’}
- Create a new variable named “modify_vlans” which creates a new file with write permission called “vlans.cfg”.
- Write the contents of the list “vlans” to the vlans.cfg files as follows:
 - vlan 10
 name DATA
 - vlan 20
 name VOICE
 - vlan 30
 name MGMT
- Verify the contents of the file vlans.cfg in Python3 and also verify it in Ubuntu.



Writing Files using WITH Statement – Lab Task

Lab Task

- Create a variable of class list named “intf” which contains dictionary items with key and value set as:
 - {'int': 'interface', 'name': 'G0/0'}
 - {'int': 'interface', 'name': 'G0/1'}
 - {'int': 'interface', 'name': 'G0/2'}
 - {'desc': 'description', 'name': 'Connected via Python'}
 - {'cmd': 'no', 'status': 'shut'}
- Create a new variable named “modify_intf” which creates a new file with write permission called “r1.interfaces.cfg”. Use the WITH statement to achieve this.
- Write the contents of the list “intf” to the r1.interfaces.cfg files as follows:
 - Interface G0/0
description Connected via Python
no shut
 - Interface G0/1
description Connected via Python
no shut
 - Interface G0/2
description Connected via Python
no shut
- Ensure **print(R1)** prints the contents of the file.



Conditionals and Functions – Lab Task

Lab Task

- Create a variable of class dict named “dict1” which contain key, and value set as:
 - {"hostname": "R1", "OS": "IOS-XE", "mgmt-ip": "10.1.1.1"}
- Create a variable of class dict named “dict2” which contain key, and value set as:
 - {"hostname": "R2", "OS": "IOS-XR", "mgmt-ip": "10.1.1.2"}
- Create a function name “device_ip” to display the following the management IP of the device.
- The management IP must be displayed based on the following conditions:
 - If the OS is IOS-XE, then it must print “The management IP of "hostname" is "mgmt-ip"”
 - If the OS is IOS-XR, then it must print “The management IP of "hostname" is "mgmt-ip"”
 - If the OS is NEXUS, then it must print “This device has an unknown image”
- Save the function as my_functions.py in ubuntu machine in the /home/rack102-ubuntu/Desktop/rohit folder.
- Ensure you are able to call this function by importing using the command “from my_functions import device_ip”



Functions, Conditional, and For Loop – Lab Task

Lab Task

- Create a file named “**device.py**” on Atom.
- Create a variable of class dict named “dev1” which contain key, and value set as:
 - **{"hostname": "R1", "OS": "IOS-XE", "mgmt-ip": "10.1.1.1"}**
- Create a variable of class dict named “dev2” which contain key, and value set as:
 - **{"hostname": "R2", "OS": "IOS-XR", "mgmt-ip": "10.1.1.2"}**
- Create a variable of class dict named “dev3” which contain key, and value set as:
 - **{"hostname": "R3", "OS": "IOS-XE", "mgmt-ip": "10.1.1.3"}**
- Create a variable of class dict named “dev4” which contain key, and value set as:
 - **{"hostname": "R4", "OS": "IOS-XR", "mgmt-ip": "10.1.1.4"}**
- Create a variable of class dict named “dev5” which contain key, and value set as:
 - **{"hostname": "R5", "OS": "NEXUS", "mgmt-ip": "10.1.1.5"}**
- Create a variable of class list named “**device_list**” which contains all the dictionaries.
- Create a function name “**device_info**” to display the management IP of the devices based on the OS platform.

Lab Task

- The management IP must be displayed based on the following conditions:
 - If the OS is IOS-XE, then it must print “The management IP of "hostname" running IOS-XE is "mgmt-ip"”
 - If the OS is IOS-XR, then it must print “The management IP of "hostname" running IOS-XR is "mgmt-ip"”
 - If the OS is NEXUS, then it must print “The management IP of "hostname" running NEXUS is "mgmt-ip"”
- Save the function as `device_info.py` in ubuntu machine in the `/home/rack102-ubuntu/Desktop/rohit` folder.
- Ensure you are able to call this function by importing it from ubuntu machine and print the information required as per the question.
- Use must use For Loop and Conditionals to achieve this requirement.



Executing Commands on a Cisco Router Using a File – Lab Task

Lab Task

- Create a file named “**Lab10.py**” on Atom. This script should be executed from the Ubuntu machine using the “python3 Lab10.py” command.
- Lab10.py must establish a connection to R1 and R2 using ConnectHandler and must check if the connection to R1 and R2 is alive.
- Lab10.py must send configs to R1 from the R1.cfg file.
- Lab10.py must send configs to R2 from the R2.cfg file.
- Lab10.py must print the commands sent to the routers.
- Ensure the connections are disconnected after printing the commands sent to the router.
- Create a file R1.cfg and R2.cfg on the Ubuntu machine in the folder /home/rack102-ubuntu/Desktop/rohit. This file must have the following config:

```
hostname Rack102-Router1
router ospf 1
router-id 1.1.1.1
network 0.0.0.0 255.255.255.255 area 0
```

```
hostname Rack102-Router2
router ospf 1
router-id 2.2.2.2
network 0.0.0.0 255.255.255.255 area 0
```