



Networkforyou

Subscribe to our
You Tube Channel



Networkforyou



**Welcome
To
Network for you
Network Automation**



Email us :
networkforyou4@gmail.com

1 of 32

WhatsApp Us : +918143809578



Traditional Network Management:

- The Traditional way to manage network devices is one at a time using SSH to the command line.
- Copying and pasting from a text file template is common.
- GUI tools to manage one device at a time have also been available for a long time but they have typically been slow and inefficient.

NMS Network Management Systems:

- NMS systems such as Solar Winds, CISCO Works and CISCO Prime Infrastructure have also been available for a long time.
- They use protocols such as SNMP and NetFlow to gather information and report on the state of the network.
- SNMP can also be used to push configuration to devices but it has limited functionality.

The issues with Traditional Network Management:

- Configuring one device at a time is time consuming and inefficient.
- It increases the error like typos and other mistakes.
- Individual edits to multiple devices by separate engineers over time with little version control leads to configuration drift (that is non standardized configuration).
- Having non standardized configurations and accessing one device at a time is also inefficient for troubleshooting.

Network Automation:

- Device Configuration
- Initial device provisioning
- Software Version Control
- Collecting statics from devices
- Compliance verification
- Reports
- Troubleshooting.

Let define what Network Automation is?

- Network Automation is the process of automating the configuring, managing, testing, deploying and operating of physical and virtual devices within a network.
- Or in other words we can say Network Automation is a methodology in which software automatically configures provisions, manages and tests network devices.
- It is used by enterprises and service providers to improve efficiency and reduce human error and operating expenses.
- Every day network tasks and functions are performed automatically.

Email us :
networkforyou4@gmail.com

WhatsApp Us : +918143809578



- Managing bandwidth and finding fast reroutes to implement the best computing paths.
- Automation is any process that is self-driven, that reduces and potentially eliminates the need for human intervention.
- Automation was once confined to the manufacturing industry.
- Highly repetitive tasks, such as automobile assembly, were turned over to machines and the modern assembly line was born.
- Python is widely used to perform network automation
- Python is an open source scripting language, thus used to automate anything.

Network Automation Benefits:

- Network programmability enables automation which reduces human to machine interaction.
- This greatly reduces the chance of human error such as typos.
- Modern tools have been built with monitoring, configuration and troubleshooting.
- It is much more scalable than configuration one device at a time.
- Network programmability can provide configuration version control.
- It can also provide software version control.
- Troubleshooting is more efficient.
- Events and error codes can be acted on programmatically.
- Improving configuration and troubleshooting efficiency reduces operational expense.
- With Network Automation we can ensure device have a standardized configuration.
- Provide reports on and correct any exceptions.
- Automatically take corrective action on events and error codes.
- There are multiple methods that can be used to automate network management like Python scripts, NETCONF, RESTCONF, Ansible, Puppet, SDN, and CISCO DNA Center etc.
- Not all this methods are supported by all devices.
- We can look and choose the methods which suitable for our environment skills.

Python for Network Programmability:

- It is easy to learn
- Human readable
- Open Source
- Can be installed on all OS
- Easy to find network automation code samples



JSON (JavaScript Object Notation):

- JSON stand for JavaScript Object Notation
- Easier for humans to read and work
- JSON is a standard text based format for representing structured data based on JavaScript Object syntax.
- It is commonly used for transmitting data from the server to the client.
- RESTful APIs often use JSON

JSON DATA Types:

- Object
- Array
- Number
- String
- Null
- Boolean

JSON Data Types: Object

- An object is an unordered collection of key or value pairs.
- It surrounded by curly braces {}
- Keys must be strings and values must be a valid JSON data type (string, numbers, Object, array, Boolean or null).
- Keys and values are separated by a colon
- Each key and value pair is separated by a comma.

```
{  
  "name" : "Ethernet1" ,  
  "description" : "Internet Link" ,  
  "enabled" : false  
}
```



JSON Data Types: Array:

- An array is an ordered list of values.
- They are surrounded by square brackets [].
- Values must be a valid JSON data type (string, number, and object, array, Boolean or null).

```
{  
  "name" : "Albert",  
  "age" : 35,  
  "brothers" : ["Joy", "Ali", "Roy"]  
}
```

JSON Data Types:

String: "name" : "anyname"

Number: "Output Errors" : 3

Boolean: "Enabled" : false

Null: "nsec" : null

XML (Extensible Markup Language):

- It is standardized in 1998.
- Widely used across the Internet.
- XML was designed to describe and transfer data.
- <key> value </key> contained within object tags.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<note>
```

```
<to>To</to>
```

```
<from>Raja</from>
```

```
<heading>Reminder</heading>
```

```
<body>Enjoy!</body>
```

```
</note>
```

Email us :
networkforyou4@gmail.com

WhatsApp Us : +918143809578



The first line is the XML declaration. It defines the XML version (1.0) and the encoding used (ISO-8859-1 = Latin-1/West European character set).

The next line describes the root element of the document (like saying: "this document is a note"):

The next 4 lines describe 4 child elements of the root (to, from, heading, and body).

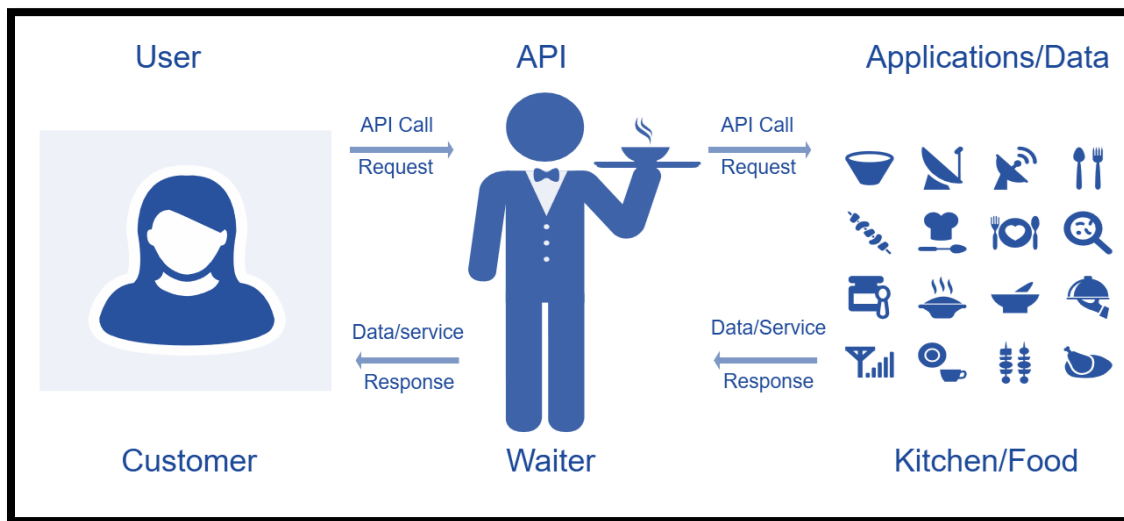
And finally the last line defines the end of the root element.

API: We, as network engineers usually use the command-line interface (CLI) or a GUI to configure or monitor our network devices. Show and debug commands with scripts is difficult because these commands are for humans. To interact with applications or network devices, we can use an Application Programming Interface (API).

- API Stand for Application Programming Interfaces.
- An API is a way for computer program to communicate directly with another computer program or in other words we can say it machine to machine communication.
- It is typically used to perform CRUD operations
- **CRUS stand for Create, Read, Update and Delete.**
- To interact with applications or network devices, we can use an Application Programming Interface (API).
- An API is a software interface which allows other applications to communicate with our application.
- APIs are found almost everywhere. Amazon Web Services, Facebook, and home automation devices such as thermostats, refrigerators, and wireless lighting systems, all use APIs.
- They are also used for building programmable network automation.
- An API is software that allows other applications to access its data or services.
- It is a set of rules describing how one application can interact with another, and the instructions to allow the interaction to occur.
- The user sends an API request to a server asking for specific information and receives an API response in return from the server along with the requested information.
- **XML and JSON are two of the most common data formats that are used with APIs.**
- An API is similar to a waiter in a restaurant.
- A customer in a restaurant would like to have some food delivered to the table.
- The food is in the kitchen where it is cooked and prepared.
- The waiter is the messenger, similar to an API.
- The waiter (the API) is the person who takes the customer's order (the request) and tells the kitchen what to do.
- When the food is read, the waiter will then deliver the food (the response) back to the customer.

Email us :
networkforyou4@gmail.com

WhatsApp Us : **+918143809578**



- REST It's an acronym for REpresentational State Transfer
- Representational means we transfer the representation of a resource between a server and a client. We use a data format for this representation, typically JSON or XML.
- State Transfer means that each operation with a REST API is self-contained. Each request carries (transfers) all information (state) to complete the operation.
- REST APIs typically use HTTP methods to retrieve or send information between applications.
- We use the same HTTP methods when we use a web browser to visit a website, but now we use them to interact with an application.
- HTTP has multiple methods, but these four are the most common ones:

GET: A read-only method to retrieve a specified resource.

POST: Submits data to the specified resource to process. The POST method can also create new resources.

PUT: Updates the specified resource by replacing the existing data.

DELETE: Deletes the specified resource.

- REST have some Constraints as given below

- **Client – Server architecture** (The Client sends a request , the server sends a response)
- **Uniform Interface:** Provides simplicity.
- **State less:** no client context is stored one the server between requests.

Email us :
networkforYou4@gmail.com

WhatsApp Us : +918143809578



- **Cacheability:** Responses must define themselves as either cacheable or non cacheable
- **Layered system:** any intermediary devices such as load balancers must be transparent to the client and server.
- **Code on demand :** It is optional that is Servers can temporarily extend or customize the functionality of a client by transferring executable code.
- NETCONF and RESETCONF are API specially designed to work with network devices.

Data Models:

Data models describe the things you can configure, monitor, and the actions you can perform on a network device. Or A data model is a well understood and agreed upon method to describe something.

YANG (Yet Another Next Generation):

- YANG Stand for Yet Another Next Generation.
- **YANG is a data modelling Language** which provides a standardized way to represent the operational and configuration data of a network device.
- It can be used both internally and when packaged for transmission.
- SNMP is widely used to monitor networks and we can configure network devices by using SNMP, But in reality we don't use SNMP much to configure Network devices. CLI Scripting is more popular for Configuration.
- YANG is a modeling language and uses data models that are similar to SNMP (MIBs)
- YANG is a standard, described in RFC 6020
- YANG uses data models that describe what you can configure on device, what you can monitor on device and administrative actions you can perform on devices (Like clearing Interface counters or resetting OSPF process).
- YANG Data Models can be Common data models or Vendor data models.
- Data models allow a uniform way for us to configure, monitor and interact with network devices.
- Network automation tools like NETCONF, RESTCONF, and gRPC require YANG data models.
- YANG uses a Hierarchical tree structure, Similar to the XML data format.

Configuration Management Tools:

- It offer an automated method to implements and monitor changes to our systems
- In other words we can say Configuration management systems are designed to make controlling large numbers of devices.
- It allows us to control many different systems in an automated way from one central location.
- **There are two types of Configuration management tools: 1)Agent based tools 2)Agentless tools**
- Agent based tools require the installation of an agent on the system we want to manage.
- Agentless tools do not require an agent or software on the system which we want to manage.

Email us :
networkforyou4@gmail.com

WhatsApp Us : +918143809578



- Ansible is an agentless tool
- Puppet and chef are two example of agent based tools

Puppet:

- Puppet is a configuration management tool used for deploying, configuring and managing servers.
- It uses an agent on the target devices
- Puppet Master runs on Linux server
- Pull model, agent checks in every 30 mins by default
- It is written in Ruby
- It is created in 2005

Chef:

- Chef is an automation platform that configures and manages our infrastructures.
- It uses a master slave architecture similar to Puppet
- An agent must installed on the target devices
- It use Pull model
- It is also written in ruby
- It is Created in 2009
- In this terminology is cook Book – Recipe

Ansible:

- It can be run from any machine with Python 2 or Python 3 Installed
- It is agentless mean no need to install any software on target machine
- It used Push Model
- It use SSH by default
- It is Originally released in 2012
- Ansible modules are pre built Python scripts
- Ansible Inventory files define all hosts that will be managed by the control workstation
- Ansible Playbooks are YAML files that outline the instructions it need to run.
- Ansible, Puppet and Chef were designed primarily for server system administration
- But Ansible is typically more suitable for network environments than Puppet and Chef because it does not require an agent.
- CISCO devices usually cannot run an agent.

SaltStack (Agent and Server Mode):

- SaltStack is another configuration management tool, in the same category as Chef and Puppet.
- SaltStack is built on Python, and it has a Python interface so a user can program directly to SaltStack by using Python code.

Email us :
networkforyou4@gmail.com

WhatsApp Us : +918143809578



- You can use Salt agentless to run Salt commands on a system without installing a Salt minion. The only requirements on the remote system are SSH and Python.
- SaltStack offers an agentless option called Salt SSH that allows users to run
- Salt commands without having to install a minion on the remote device or node.
- This is similar in concept to Puppet Bolt. The main requirements to use Salt SSH are that the remote system must have SSH enabled and Python installed.
- Salt SSH connects to a remote system and installs a lightweight version of SaltStack in a temporary directory and can then optionally delete the temporary directory and all files upon completion, leaving the remote system clean.
- These temporary directories can be left on the remote systems along with any necessary files to run Salt SSH.
- This way, the files do not have to be reinstalled to the remote device, which can be useful when time is a consideration.
- This is often useful on devices that are using Salt SSH more frequently than other devices in the environment.

Introduction Python for Network Engineer:

- Python is a programming language.
- It is free and open source.
- Developed in the 90s by Guido Van Rossum.
- We can use Python for anything from simple scripts to complex object-oriented programming.
- The syntax of Python is simple, which makes it easy to read and write code.
- Python is easy to read and write. Here we are going to Learn how we can use python for Networks.

Pythons Usages:

- If you have any repetitive tasks then you can automate with Python. You can write Python script to do it for you.
- We can connect to Our devices with Pythons scripts do any configuration with pythons.

Other usages web developments, Machine learning, Data analysis, Games and Desktop applications

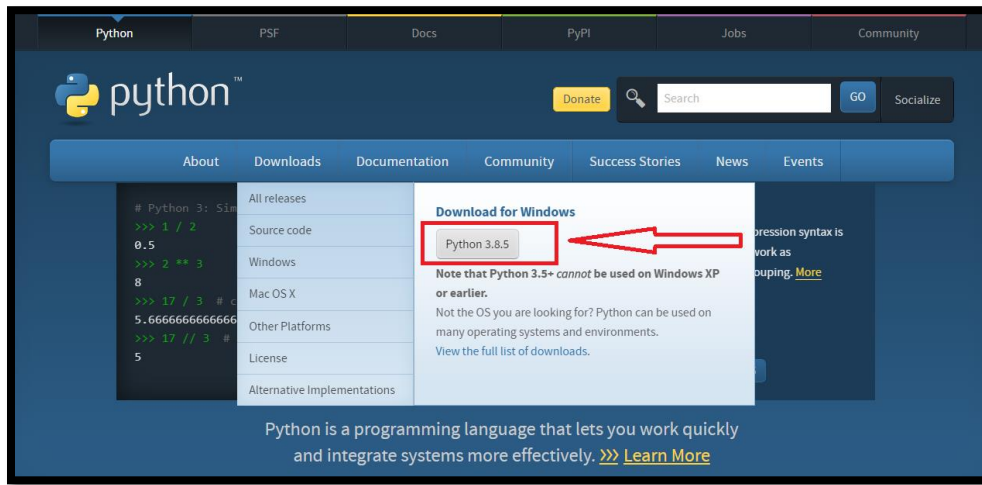
How to Install Python:

- Go to browser type www.python.org Click download tab website automatically detect your Operating system and Show you (In my case: download for Windows click that and download that.)

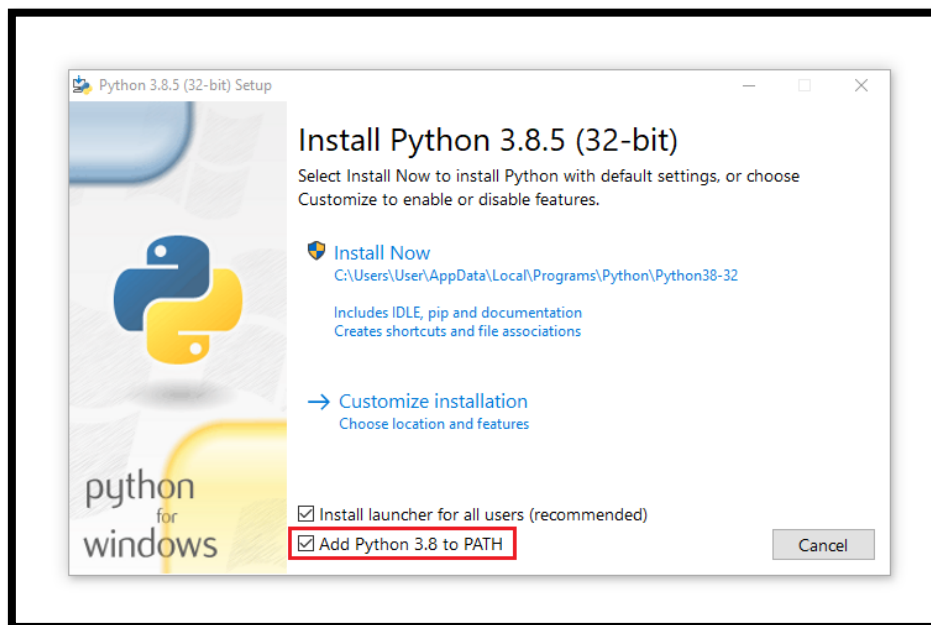
Email us :
networkforyou4@gmail.com

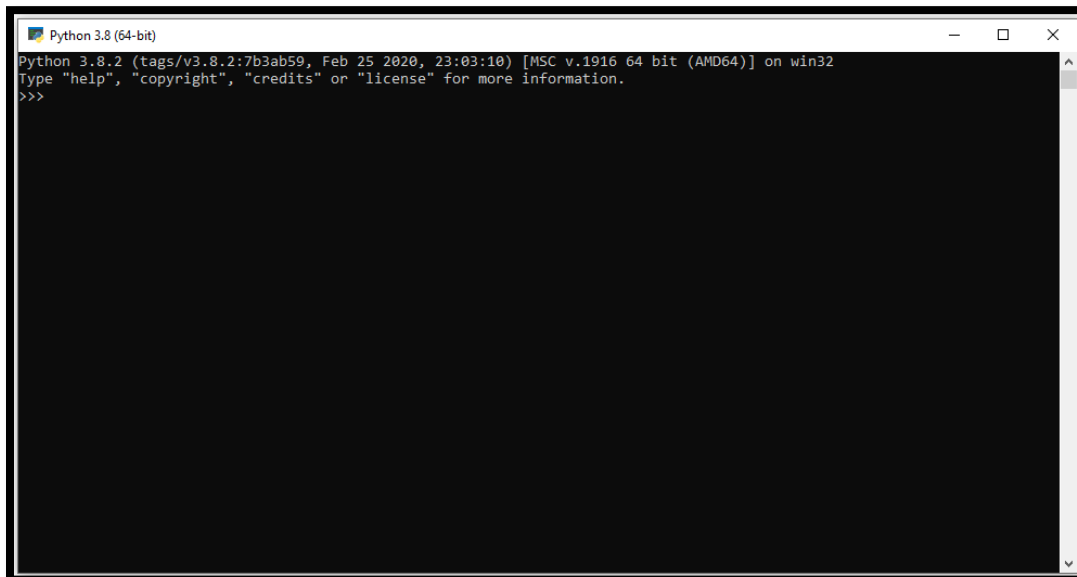
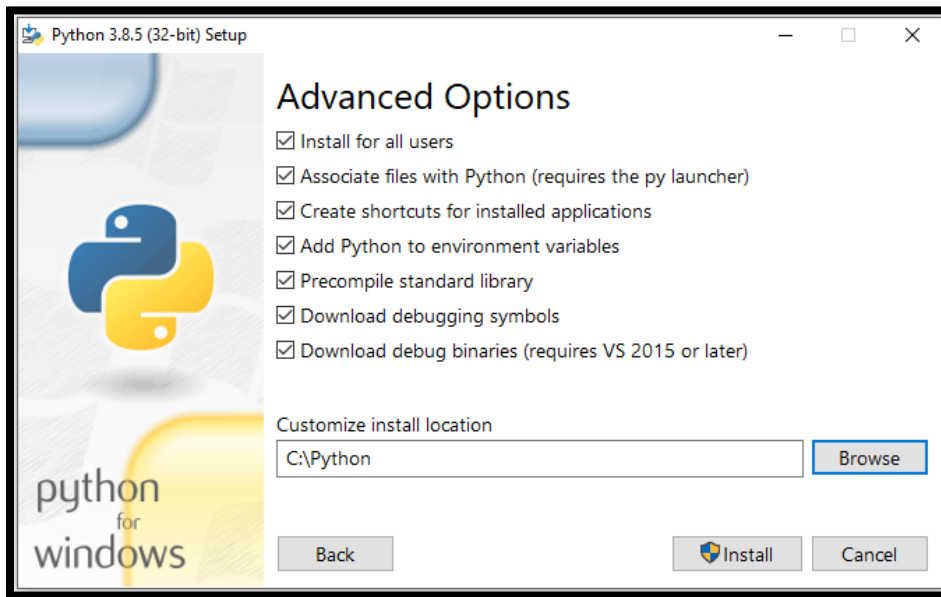
10 of 32

WhatsApp Us : +918143809578



- Open the installation file and we will see the following screen:
- The installation is straight forward but we will suggest clicking on the “Add Python 3.x to PATH” checkbox.
- **This means you can run Python anywhere from the command line.**
- Press the install now button and you only have to wait some minutes for the installation to complete.





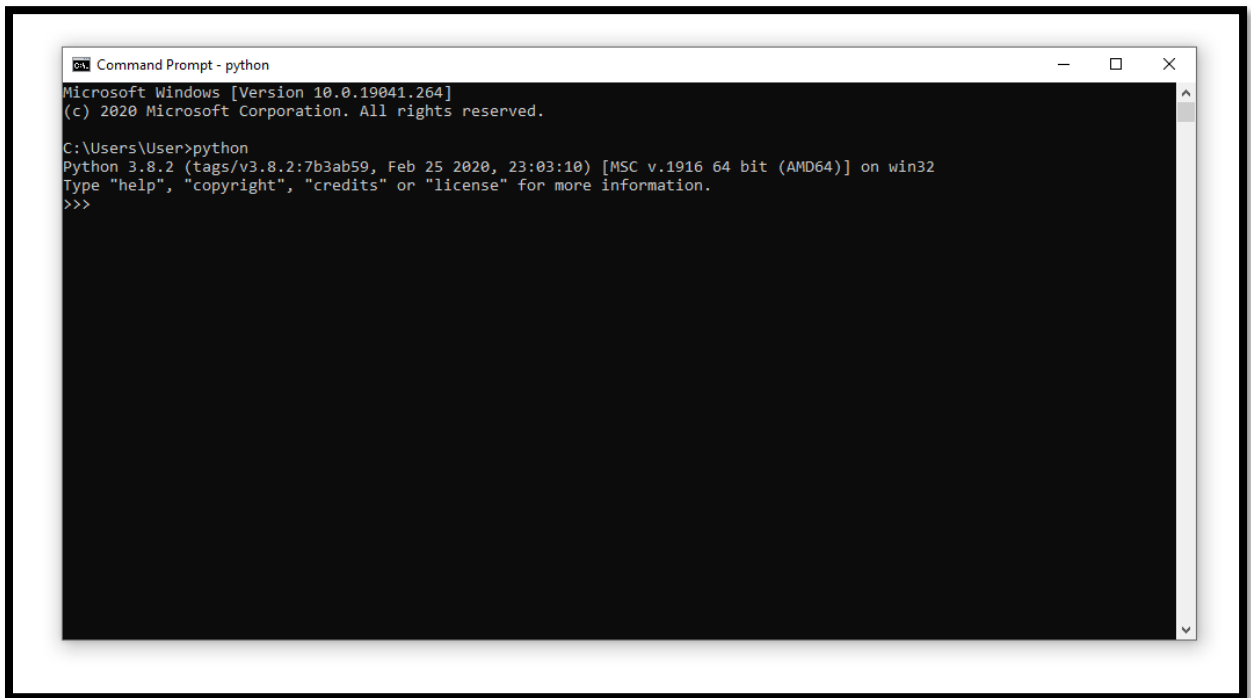


```
Python 3.8 (64-bit)
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 4 + 4
8
>>> 4*4
16
>>> 4-4
0
>>> 4/4
1.0
>>> 4%4
0
>>>
```

- We can use python in command prompt – python. Type python.
- As given below screen short.
- We can install I python. I python is mask of python
- I python make good look color coded. (Ipython is Interactive (Better look))
- We can exit python shell with **exit ()** function.
- We can Install ipython with the following command as given below.

Pip install ipython

- We use pip. It is python management system.
- To download and install any package from the pypi.org we can do but with the help of pip we can do directly.
- Example: Just write pip install ipython
- So, it will download and install.
- Pip we can say it is packages management system which manage packages.
- Package is nothing but codes like we cannot write all code for everything so we use other codes that is known as packages.



- Automatically pip install supporting packages when we install any packages. Like dependency packages.
- Dependency is like one package is depend with another package. In simple we can say one package is depend on other packages.

Print Function:

- Print function is very important to print.
- It is inbuilt function.
- It has Function name and input (Arguments)
- Print function shows information to the user of our program.
- It is very useful to show the user what the program is doing when we execute it.

Syntax is: function_name (input)

Example Print (4+4)



Python Variables:

- Assigning a value to a variable is called declaration.
- We can give a variable a name, then use the equal symbol (=) to declare the value.
- **Variables are useful when you want to reuse values in our codes.**

Strings:

Example 1: a = "Hello World"

Print (a)

b = "Here is NetworkforYou Router"

print (b)

Numbers:

We can also use this for numbers. Let declare a number to variables "a" and "b"

Example 1: a = 5

b = 10

Print (a)

Print (b)

Example 2: a = 5

b = 10

print (a+b)

Re- declare Variable:

We can also re-declare a variable.

Example:

a=1

b=2

print(a)

print (b)

a=3

print(a)



Variable within Variable:

We can assign variable within variable

Example: a=10

b=a

print (b)

Here we create variable a then we assign that a in variable b then we print b.

Functions:

We can also declare a variable to a Python Function.

Example: router = input ("this is NetworkforYou Router")

Print(router)

Conclusion:

- In this we learn how we can declare Variable.
- Declaration means we can assign a value to variable.
- We can use variables for many things like strings, numbers, Functions etc.
- Re- declaring a variable is no issues we can do like that.

Python User Input:

- Python supports user input which means program will ask user to input.
- Like when user input any data and press enter it will store in variable as a string.

Example:

Hostname = input ("Enter your Hostname")

```
In [3]: hostname = input ("Enter your Hostname:" )
Enter your Hostname:Router

In [4]: print (hostname)
Router
```

- You can use like this multiple examples to use this User Input function in Python.
- The input () function can show a string to the user, like In our case it show Enter your Hostname Otherwise the user might not even notice what they need to enter etc.



Python If, Elif, Else:

- Till now we learn how we can execute code in sequentially. Now we are going to learn how we can run code skipped some code and check if any condition is matching then only this code run else not etc.
- So, we will use the if, elif, else statements to make decisions.

If:

- Let see one example:

```
ios_version = 16
if ios_version == 16:
    print ("Yes it is Version 16")
```

See in this it will print if version is 16 if it is not then it will not print.

Shorthand If:

When our if statement consists of a single line, then we can use the shorthand notation. Let see example.

```
if 100 > 55: print ( "Router is using High CPU")
```

Nested if:

We can also use nested if statements. This can be useful if we want to check multiple conditions. Let see some example:

```
ios_version = 16
hostname = "R1"
if "R1" in hostname:
    print("This device is a Router")
    if ios_version == 16:
        print ("Router runs IOS Version 16")
```

Else:

When we do not have a match with the if statement, everything else matches the else statement. Let see example.

Email us :
networkforyou4@gmail.com

WhatsApp Us : +918143809578



```
cpu_load = 85
    if cpu_load > 90:
        print ("CPU load is over 90%")
    else:
        print("CPU load is OK")
```

Shorthand Else:

- If we want to write in one line, we can do like that also it is called Shorthand else.
- This could be useful if you only have a single action under your statements.
- Let see example:

```
print ("CPU load above threshold.") if 90 > 85 else print ("CPU load is OK")
```

Elif Statements:

If you have multiple if statements then we can add that by using the elif statement.

Let see example:

```
ios_version = 16
if ios_version == 16:
    print( "Router IOS is Version 16")
elif ios_version == 15:
    print("Router IOS is Version 15")
else:
    print("we donot have version 15 and 16")
```

Python3: Lab:

```
# Import required libraries
import telnetlib

# Define my important variables
HOST = '192.168.20.139'
user = 'admin'
password = 'cisco'

# Using this info telnet into device
tn = telnetlib.Telnet(HOST)
```

Email us :
networkforYou4@gmail.com

WhatsApp Us : +918143809578



```
# Handle the Username/Password prompt and supply our values
```

```
tn.read_until(b"Username: ")
```

```
tn.write(user.encode('ascii') + b"\n")
```

```
if password:
```

```
    tn.read_until(b"Password: ")
```

```
    tn.write(password.encode('ascii') + b'\n')
```

```
# Write the commands on the device
```

```
tn.write(b'enable\n')
```

```
tn.write(b'cisco\n')
```

```
tn.write(b'conf ter\n')
```

```
tn.write(b'int loop 0\n')
```

```
tn.write(b'ip address 1.1.1.1 255.255.255.255\n')
```

```
tn.write(b'int loop 1\n')
```

```
tn.write(b'ip address 2.2.2.2 255.255.255.255\n')
```

```
tn.write(b'end\n')
```

```
tn.write(b'exit\n')
```

```
# Print out what we have done on device
```

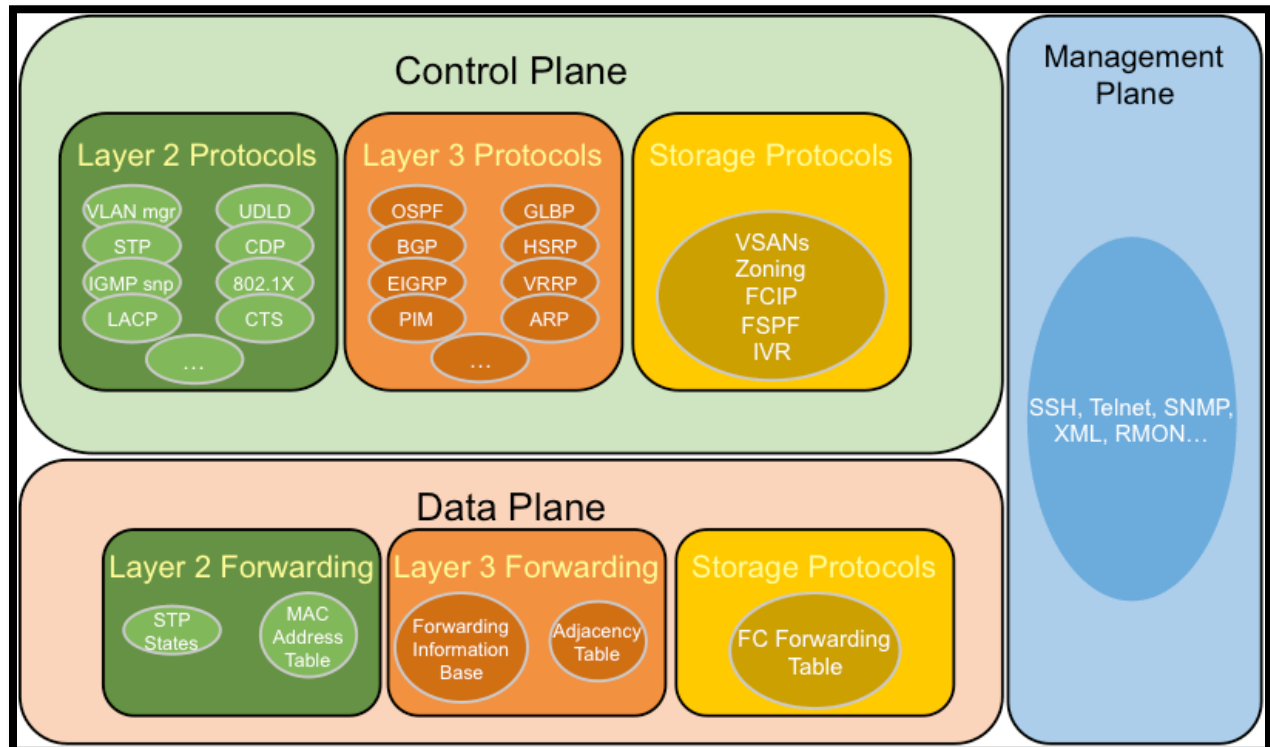
```
print(tn.read_all().decode())
```



SDN (Software Defined Networking):

Software Defined Networking (SDN) is a network architecture approach that enables the network to be intelligently and centrally controlled or programmed by using software application.

Before starting SDN we need to have idea about Router and switch Plan so let discuss about that.



Control Plane: The control plane is responsible for exchanging routing information building the arp table etc. As given below the responsible of control plan.

- Learning MAC address to build a switch MAC address table
- Running STP to create a loop free topology.
- Building ARP tables
- Running routing protocols like OSPF, BGP, EIGRP etc. to build routing table.

Data Plane: Data plane is responsible for forwarding traffic and it relies on the information that get from control plan. Here is some task that take care by Data Plan is given below.

- Encapsulate and de encapsulate packets
- Adding or removing headers like 802.1 Q header
- Dropping traffic because of ACL

Email us :
networkforYou4@gmail.com

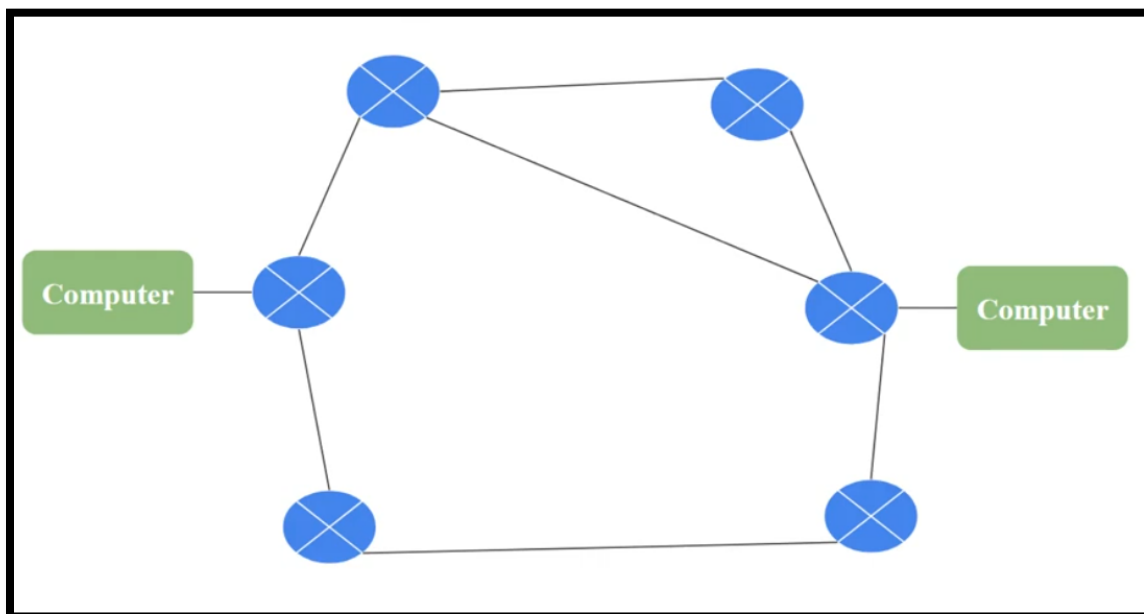
WhatsApp Us : +918143809578

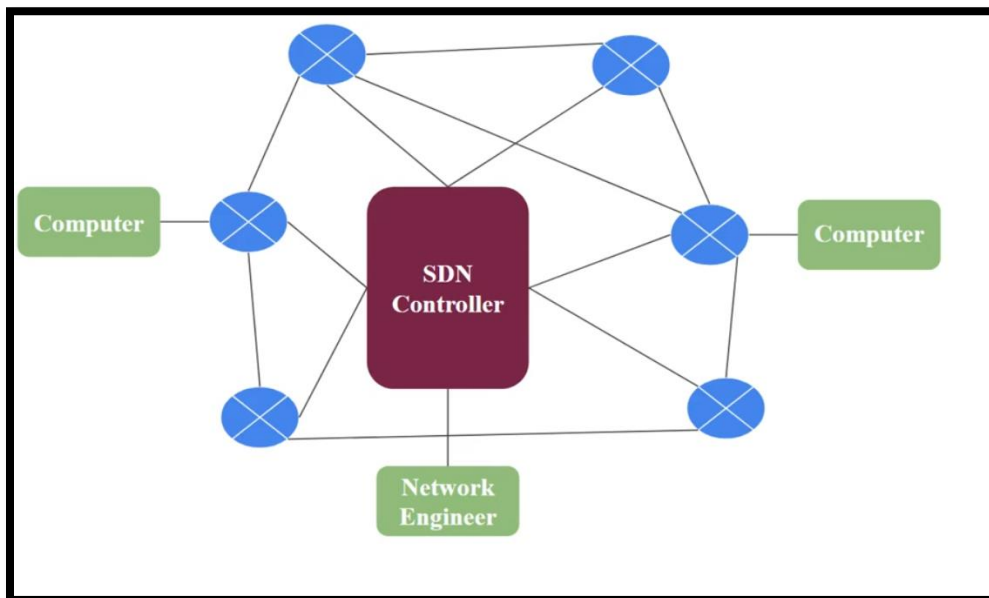
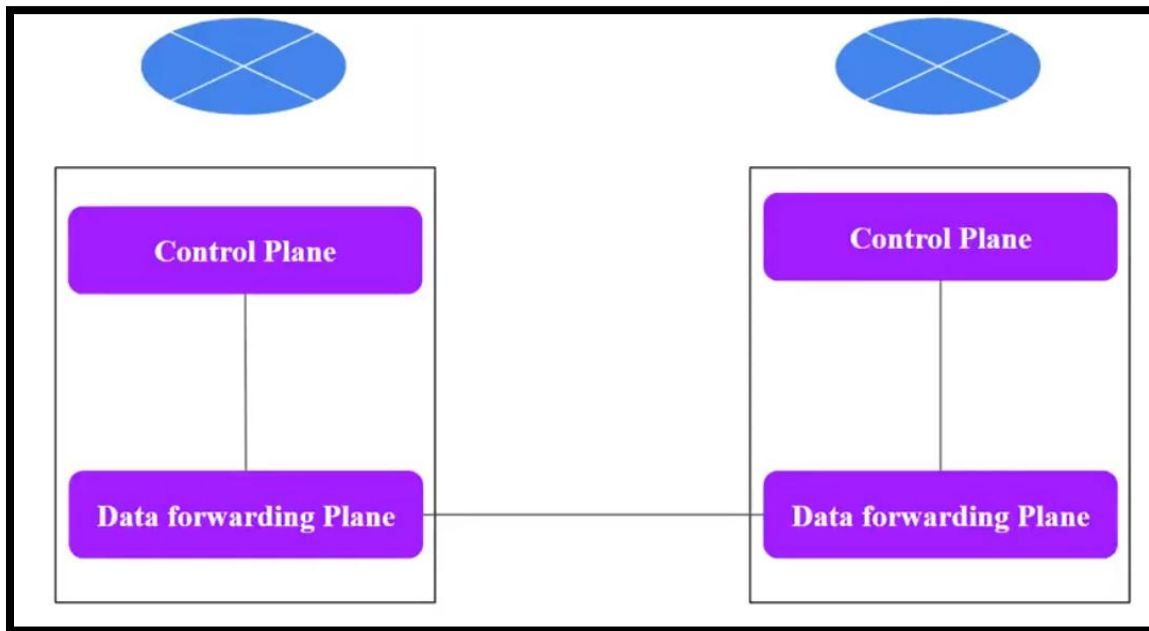


- Matching MAC addresses for forwarding
- Change source and destination address when using NAT
- Matching IP destinations in the routing tables

Management Plane: Is used for access and management of our network devices. Like using Telnet, SSH or console port etc.

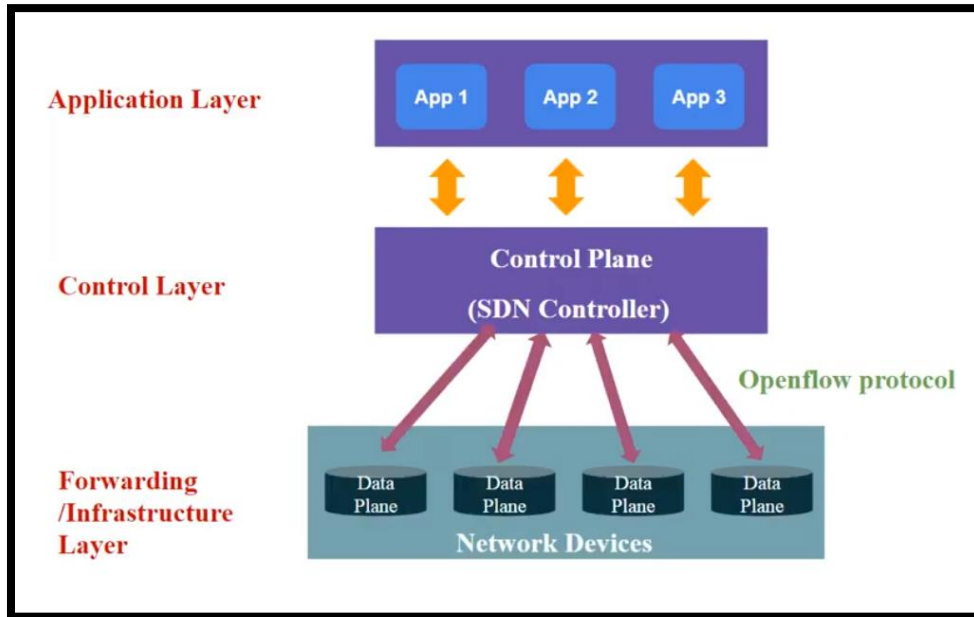
- Telnet
- SSH
- Console







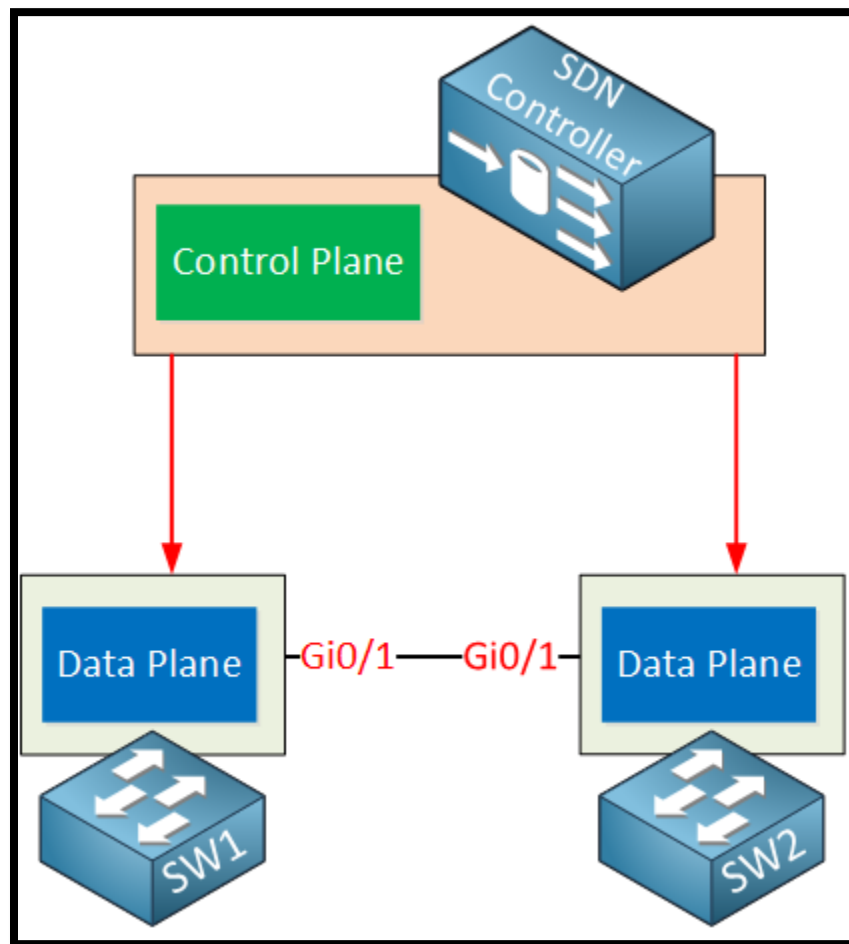
SDN Network Architecture:



- We can say with SDN we can use a central controller for the control plane.
- It can be Physical Hardware device or Virtual Machine.

Email us :
networkforyou4@gmail.com

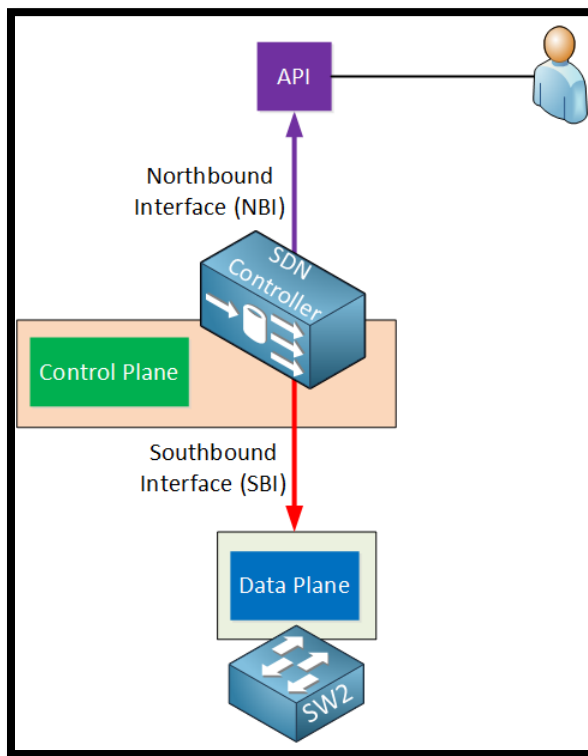
WhatsApp Us : +918143809578



- Above we can see the SDN controller which is responsible for the control plane.
- The switches are now just “dumb” devices that only have a data plane, no control plane.
- The SDN controller is responsible for feeding the data plane of these switches with information from its control plane.
- There are some advantages and disadvantages of having a distributed vs a central control plane.
- One of the advantages of having a central controller is that we can configure the entire network from a single device.
- This controller has full access and insight of everything that is happening in our network.

Email us :
networkforyou4@gmail.com

WhatsApp Us : +918143809578



- The SDN controller has to communicate with our network devices in order to program the data plane.
- This is done through the southbound interface.
- This is not a physical interface but a software interface, often an API (Application Programming Interface).
- An API is a software interface that allows an application to give access to other applications by using pre-defined functions.

Southbound Interface:

- OpenFlow
- Cisco OpFlex
- Rest
- NETCONF
- RESTCONF
- SSH

Email us :
networkforyou4@gmail.com

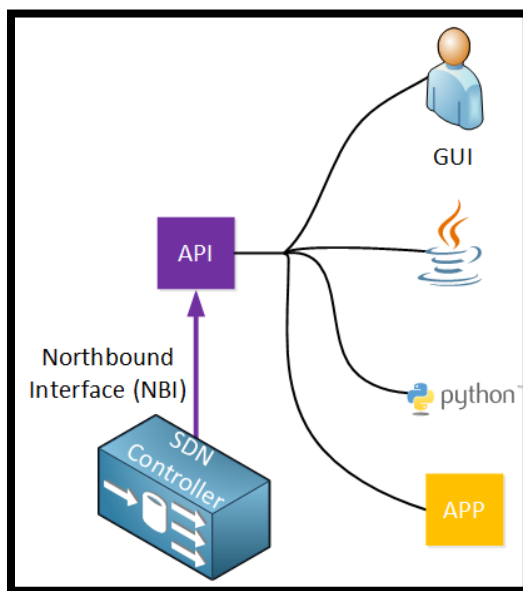
WhatsApp Us : +918143809578



Northbound Interface:

- The northbound interface is used to access the SDN controller itself.
- This allows a network administrator to access the SDN to configure it or to retrieve information from it.
- This could be done through a GUI but it also offers an API which allows other applications access to the SDN controller.
- We can use this to write scripts and automate your network administration.

Typically, REST is northbound interface.



- Open Daylight: It is an Open source SDN controller hosted by the Linux Foundations
- CISCO SDN Controllers is APIC (**Application Policy Infrastructure Controller**). Is design to manage Data center
- Cisco SDN Controllers is DNA Center. It is design to manage Enterprise environments like branch, Campus and WAN

CISCO DNA Center:

- DNA Stand for Digital Network Architecture
- Cisco DNA enables us to streamline operations and facilitate IT and Business innovation.
- IBN (Intent-based networking) built on CISCO DNA takes a software delivered approach to automating and assuring services across your wan and Your Branch and Campus networks or we can say DNA is CISCO software define Architecture solution to do our task automation.

Email us :
networkforyou4@gmail.com

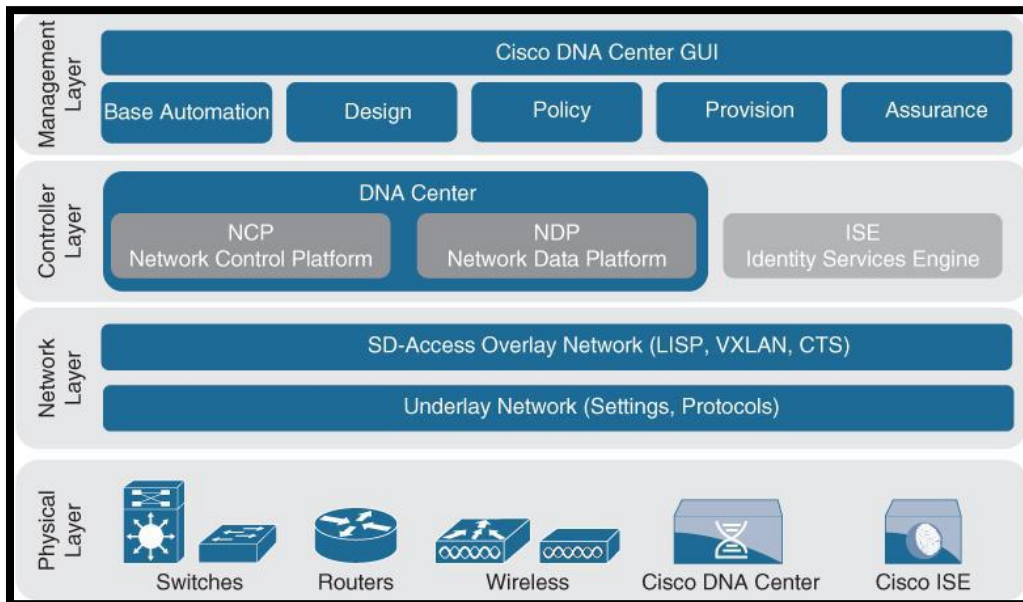
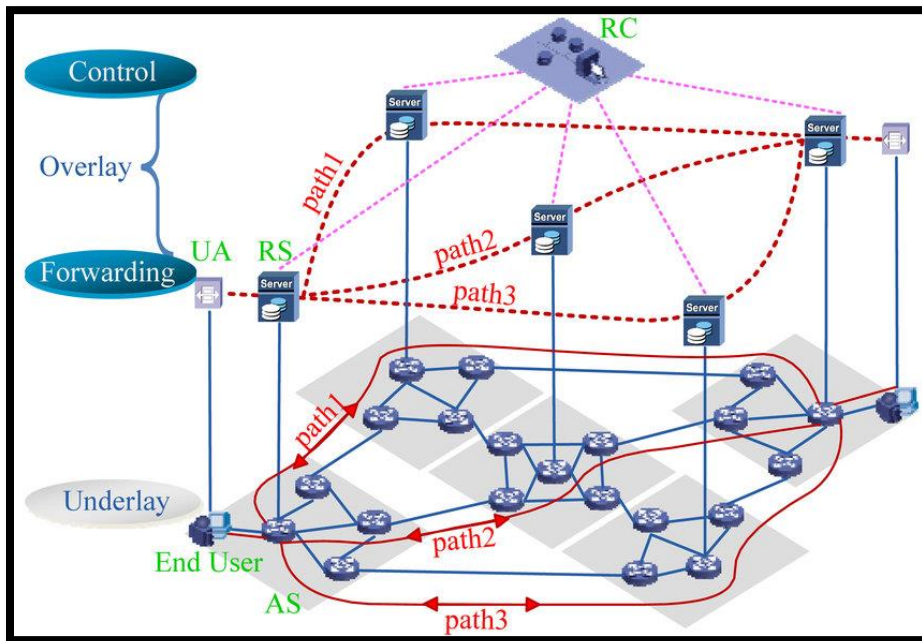
WhatsApp Us : **+918143809578**



- **Intent Based Networking (IBN) it is key part of concepts for Software Defined Networking and IBN (Intent Based Networking) Transforms a traditional manual network into a controller base network that translates the business needs into policies that can be run automated and applied across the network.**
- **The Main Aim of IBN is to continuously monitor and adjust network performance to help desired business outcomes.**
- Three of the main building blocks of CISCO DNA and Software Defined Architecture are **DNA Center, SD Access and SD WAN**
- **DNA Center is a CISCO SDN Controller** which is designed to manage enterprise network
- We can say DNA Center as an upgrade to the APIC-EM (Older CISCO SDN Controller) (Application Policy Infrastructure Controller Enterprise Module)
- DNA Center Runs as Appliance on CISCO UCS Server Hardware. Underlying Operating System Linux
- It can be Clustered for redundancy.

SD Access:

- SD Access is a newer method of network access control which solves the limitation of traditional implementation.
- It is traffic flow security is based on user identity not physical location and IP address.
- Users log in form and can move to any physical location in the network
- 2 Components are required for SD Access ISE (Identity Services Engine) and DNA Center.
- Users are authenticated by the ISE and the security policy is configured on the DNA Center.
- SD Access uses an underlay and overlay network
- **An underlay network is the underlying physical network. Physical network and it provides the underlying physical connections which the overlay network is built on top.**
- **An overlay network is a logical topology used to virtually connect devices it is build over the Physical underlay network.**
- **The combination of underlay and overlay form the SD Access Network Fabric.**
- Underlay Network is physical infrastructure above which overlay network is built.
- An Overlay network is a virtual network that is built on top of an underlying Network infrastructure/Network layer



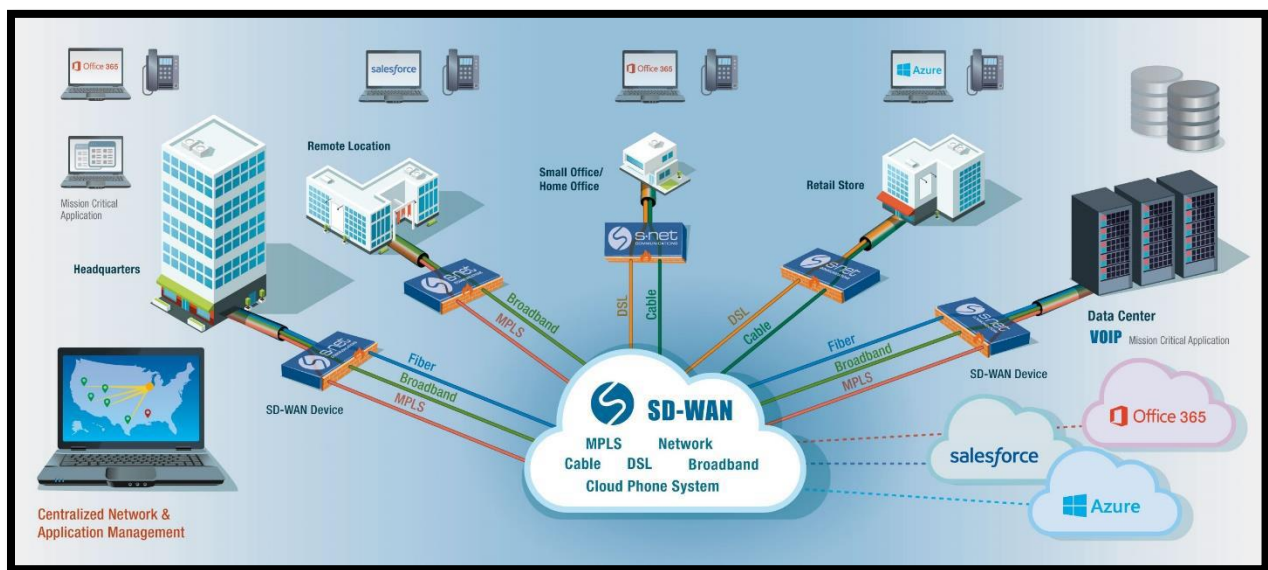
Email us :
networkforyou4@gmail.com

WhatsApp Us : +918143809578



SD- WAN:

- SD WAN Provides automated setup of WAN connectivity between sites
- Traffic flow control is application aware
- Monitoring and Failover is automated
- Automated, standardized setup of connectivity between sites
- Lower cost
- More Flexibility and easier to migrate WAN services
- Integration with the latest cloud and network technologies



DevNet DNA Center Lab URL and Username and password.

<https://sandboxdnac.cisco.com>

User: devnetuser

Password: Cisco123!

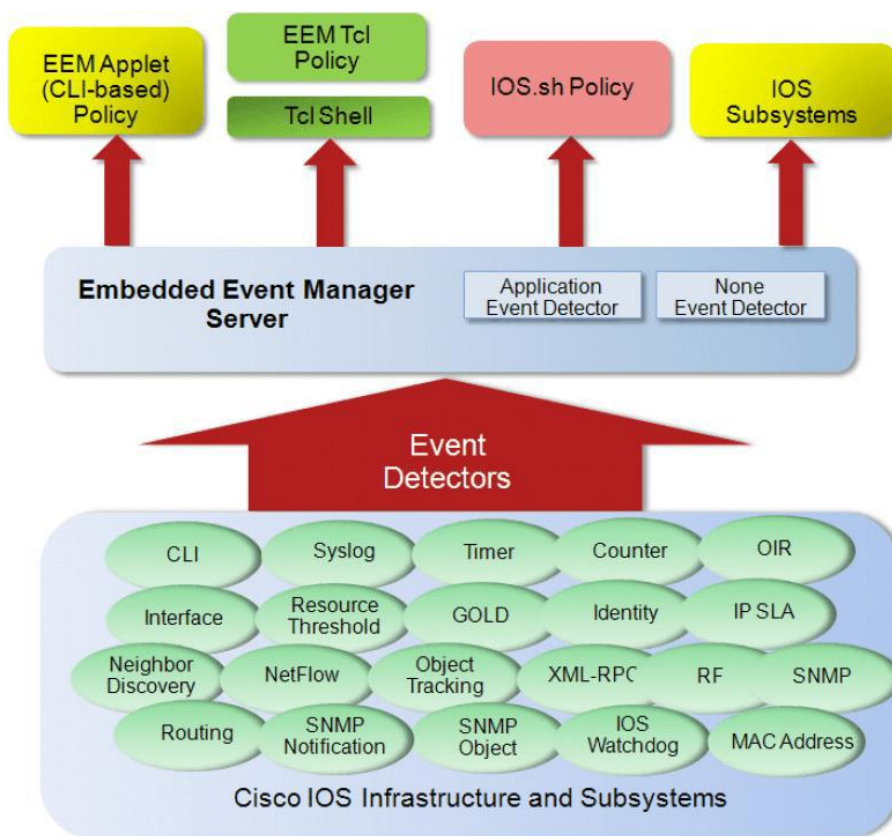
Email us :
networkforyou4@gmail.com

WhatsApp Us : +918143809578



Embedded Event Manager (EEM):

- Embedded Event Manager (EEM) is a technology on Cisco Routers that lets you run scripts.
- EEM is a technology on Cisco Routers that run commands when a certain event happens.
- EEM is powerful & flexible tool to automate tasks and customize the behavior of Cisco IOS.
- The Cisco Embedded Event Manager or Cisco EEM is a software component of Cisco IOS.
- That provides real-time network event detection and onboard automation in Cisco IOS.
- EEM allows you to automate tasks, perform minor enhancements and create workarounds.
- EEM makes life easier for network operators by tracking & classifying events that take place.
- Embedded Event Manager (EEM) providing actions options for those events in the Network.
- Embedded Event Manager uses event detectors & actions to provide notifications of events.
- Embedded Event Manager Supports more than 20 event detectors that are highly integrate.
- With different IOS Software components to trigger actions in response to network events.

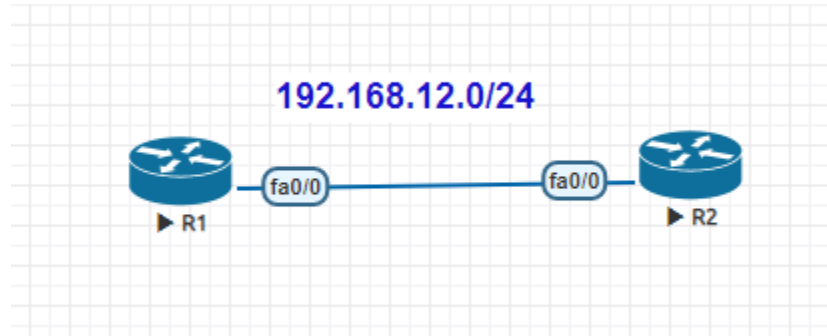


Email us :
networkforYou4@gmail.com

WhatsApp Us : +918143809578



Lab Embedded Event Manager (EEM) Lab:



R1 Configuration	R2 Configuration
<pre>en config t hostname R1 interface f0/0 ip add 192.168.12.1 255.255.255.0 no shutdown</pre>	<pre>en config t hostname R2 interface f0/0 ip add 192.168.12.2 255.255.255.0 no shutdown - event manager applet abc event syslog pattern "Interface FastEthernet0/0, changed state to administratively down" action 1 cli command "enable" action 2 cli command "config t" action 3 cli command "interface f0/0" action 4 cli command "no shutdown" - - - - - R2#debug event manager action cli event manager applet INT-Status event syslog pattern "Interface f0/0, changed state to down" action 1 cli command "enable" action 2 cli command "config t" action 3 cli command "interface e0/0" action 4 cli command "no shutdown"</pre>

Email us :
networkforyou4@gmail.com

WhatsApp Us : +918143809578



```
R2#config t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#int f0/0
R2(config-if)#sh
R2(config-if)#
*Sep 8 13:52:46.883: %LINK-5-CHANGED: Interface FastEthernet0/0, changed state to administratively down
*Sep 8 13:52:47.027: %HA_EM-6-LOG: abc : DEBUG(cli_lib) : : CTL : cli_open called.
*Sep 8 13:52:47.047: %HA_EM-6-LOG: abc : DEBUG(cli_lib) : : OUT : R2>
*Sep 8 13:52:47.051: %HA_EM-6-LOG: abc : DEBUG(cli_lib) : : IN : R2>enable
*Sep 8 13:52:47.079: %HA_EM-6-LOG: abc : DEBUG(cli_lib) : : OUT : R2#
*Sep 8 13:52:47.083: %HA_EM-6-LOG: abc : DEBUG(cli_lib) : : IN : R2#config t
*Sep 8 13:52:47.211: %HA_EM-6-LOG: abc : DEBUG(cli_lib) : : OUT : Enter configuration commands, one per line. End with C
NTL/Z.
*Sep 8 13:52:47.211: %HA_EM-6-LOG: abc : DEBUG(cli_lib) : : OUT : R2(config)#
*Sep 8 13:52:47.215: %HA_EM-6-LOG: abc : DEBUG(cli_lib) : : IN : R2(config)#interface f0/0
*Sep 8 13:52:47.339: %HA_EM-6-LOG: abc : DEBUG(cli_lib) : : OUT : R2(config-if)#
*Sep 8 13:52:47.343: %HA_EM-6-LOG: abc : DEBUG(cli_lib) : : IN : R2(config-if)#no shutdown
*Sep 8 13:52:47.467: %HA_EM-6-LOG: abc : DEBUG(cli_lib) : : OUT : R2(config-if)#
*Sep 8 13:52:47.471: %HA_EM-6-LOG: abc : DEBUG(cli_lib) : : CTL : cli_close called.
*Sep 8 13:52:47.507:
*Sep 8 13:52:47.507: tty is now going through its death sequence
*Sep 8 13:52:49.367: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
```

Email us :
networkforyou4@gmail.com

WhatsApp Us : +918143809578