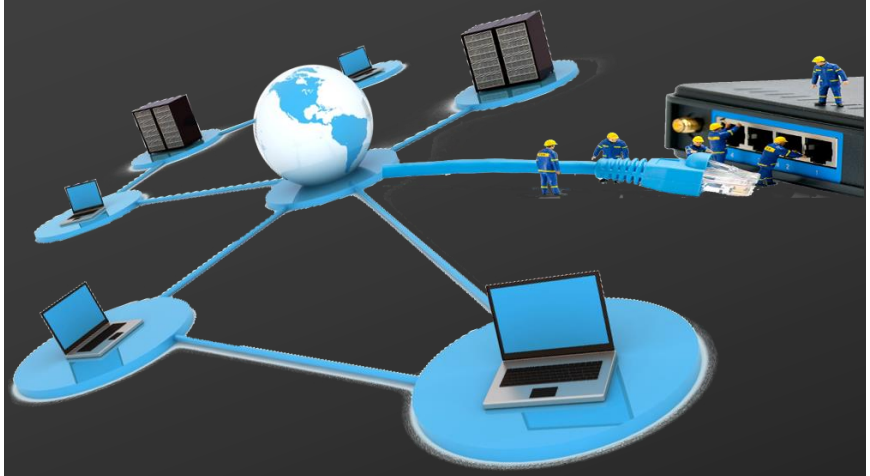
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN – ĐHQG-HCM
Khoa Mạng máy tính & Truyền thông

Cloud và Tự động hoá

NT132 – Quản trị mạng và hệ thống

GV: Đỗ Hoàng Hiển

hiendh@uit.edu.vn



Nội dung

ACL

Hôm nay học gì?

1. Cloud
2. Tự động hoá

Cloud Computing

Cloud Overview

Cloud computing addresses a variety of data management issues:

- Enables access to organizational data anywhere and at any time
- Streamlines the organization's IT operations by subscribing only to needed services
- Eliminates or reduces the need for onsite IT equipment, maintenance, and management
- Reduces cost for equipment, energy, physical plant requirements, and personnel training needs
- Enables rapid responses to increasing data volume requirements

Cloud Services

The three main cloud computing services defined by the National Institute of Standards and Technology (NIST) in their Special Publication 800-145 are as follows:

- **Software as a Service (SaaS)** - The cloud provider is responsible for access to applications and services that are delivered over the internet.
- **Platform as a Service (PaaS)** - The cloud provider is responsible for providing users access to the development tools and services used to deliver the applications.
- **Infrastructure as a Service (IaaS)** - The cloud provider is responsible for giving IT managers access to the network equipment, virtualized network services, and supporting network infrastructure.

Cloud service providers have extended this model to also provide IT support for each of the cloud computing services (ITaaS). For businesses, ITaaS can extend the capability of the network without requiring investment in new infrastructure, training new personnel, or licensing new software.

Cloud Models

There are four primary cloud models:

- **Public clouds** - Cloud-based applications and services made available to the general population.
- **Private clouds** - Cloud-based applications and services intended for a specific organization or entity, such as the government.
- **Hybrid clouds** - A hybrid cloud is made up of two or more clouds (example: part private, part public), where each part remains a separate object, but both are connected using a single architecture.
- **Community clouds** - A community cloud is created for exclusive use by a specific community. The differences between public clouds and community clouds are the functional needs that have been customized for the community. For example, healthcare organizations must remain compliant with policies and laws (e.g., HIPAA) that require special authentication and confidentiality.

Cloud Computing versus Data Center

These are the correct definitions of data center and cloud computing:

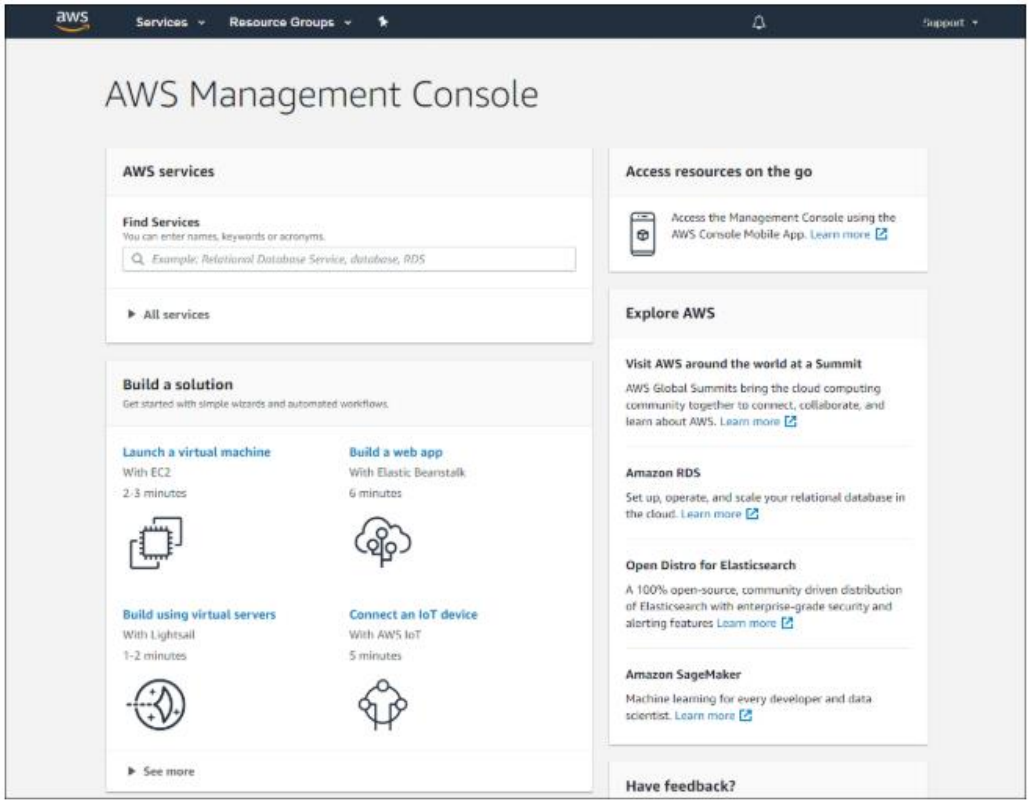
- **Data center**: Typically, a data storage and processing facility run by an in-house IT department or leased offsite. Data centers are typically very expensive to build and maintain.
- **Cloud computing**: Typically, an off-premise service that offers on-demand access to a shared pool of configurable computing resources. These resources can be rapidly provisioned and released with minimal management effort.

Data centers are the physical facilities that provide the compute, network, and storage needs of cloud computing services. Cloud service providers use data centers to host their cloud services and cloud-based resources.

Virtualization

Cloud Computing and Virtualization

- The terms “cloud computing” and “virtualization” are often used interchangeably; however, they mean different things. Virtualization is the foundation of cloud computing. Without it, cloud computing, as it is most-widely implemented, would not be possible.
- Virtualization separates the operating system (OS) from the hardware. Various providers offer virtual cloud services that can dynamically provision servers as required. These virtualized instances of servers are created on demand.



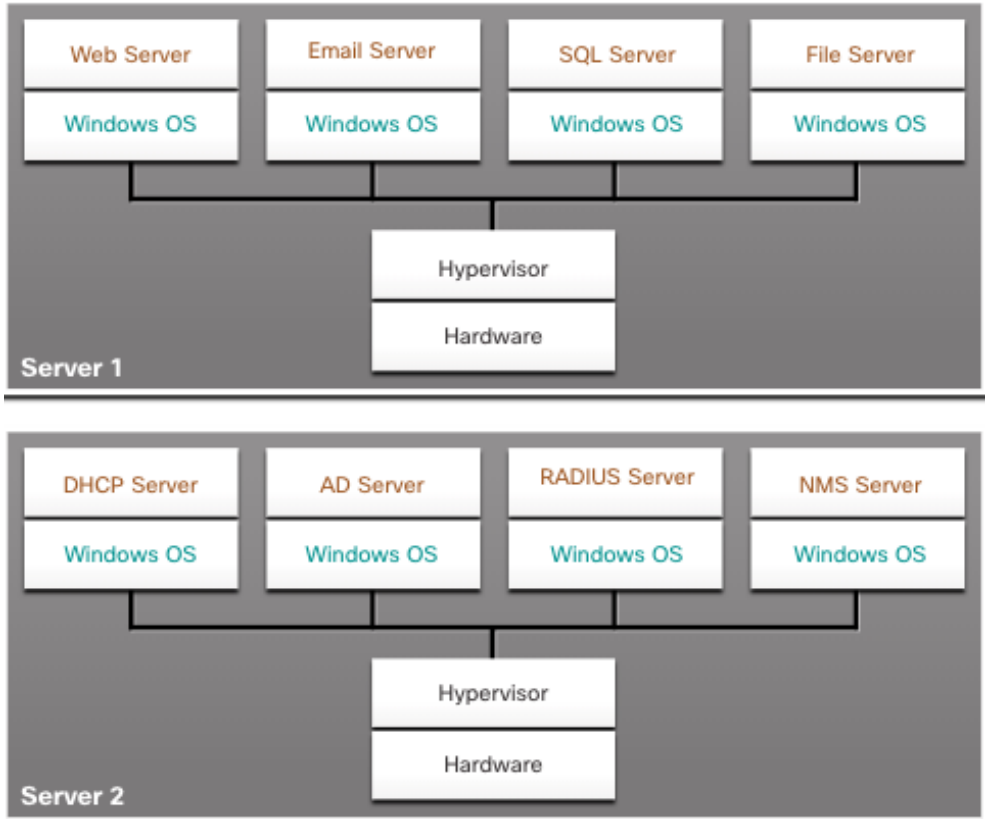
Dedicated Servers

Historically, enterprise servers consisted of a server OS, such as Windows Server or Linux Server, installed on specific hardware. All of a server's RAM, processing power, and hard drive space were dedicated to the service provided (e.g., Web, email services, etc.).

- When a component fails, the service that is provided by this server becomes unavailable. This is known as a single point of failure.
- Dedicated servers were generally underused. They often sat idle for long periods of time, waiting until there was a need to deliver the specific service they provide. These servers wasted energy and took up more space than was warranted by the amount of service provided. This is known as server sprawl.

Server Virtualization

- Server virtualization takes advantage of idle resources and consolidates the number of required servers. This also allows for multiple operating systems to exist on a single hardware platform.
- The use of virtualization normally includes redundancy to protect from a single point of failure.
- The hypervisor is a program, firmware, or hardware that adds an abstraction layer on top of the physical hardware. The abstraction layer is used to create virtual machines which have access to all the hardware of the physical machine such as CPUs, memory, disk controllers, and NICs.



Advantages of Virtualization

One major advantage of virtualization is overall reduced cost:

- Less equipment is required
- Less energy is consumed
- Less space is required

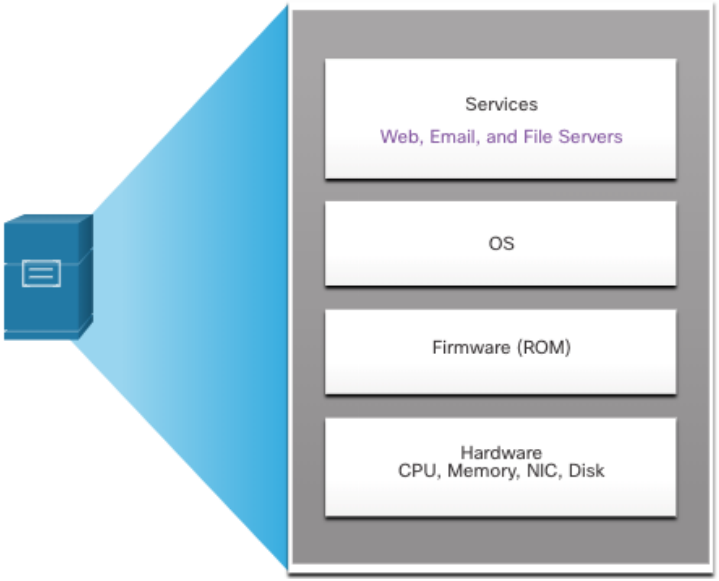
These are additional benefits of virtualization:

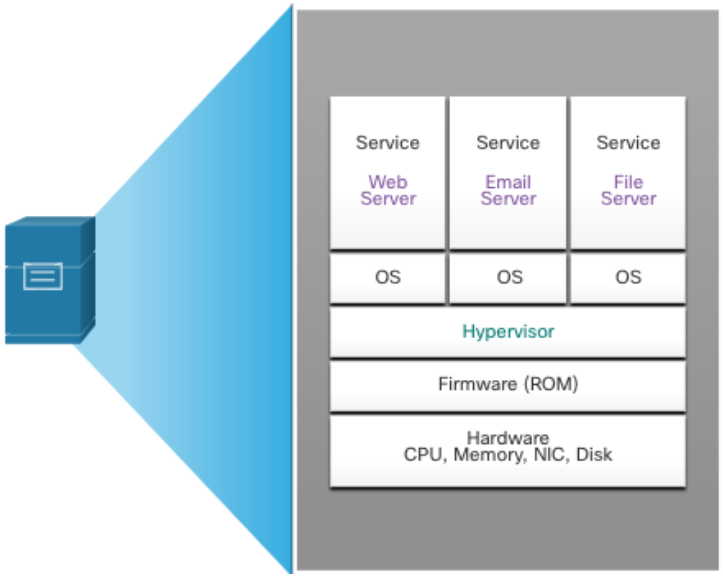
- Easier prototyping
- Faster server provisioning
- Increased server uptime
- Improved disaster recovery
- Legacy support

Abstraction Layers

A computer system consists of the following abstraction layers: Services, OS, Firmware, and Hardware.

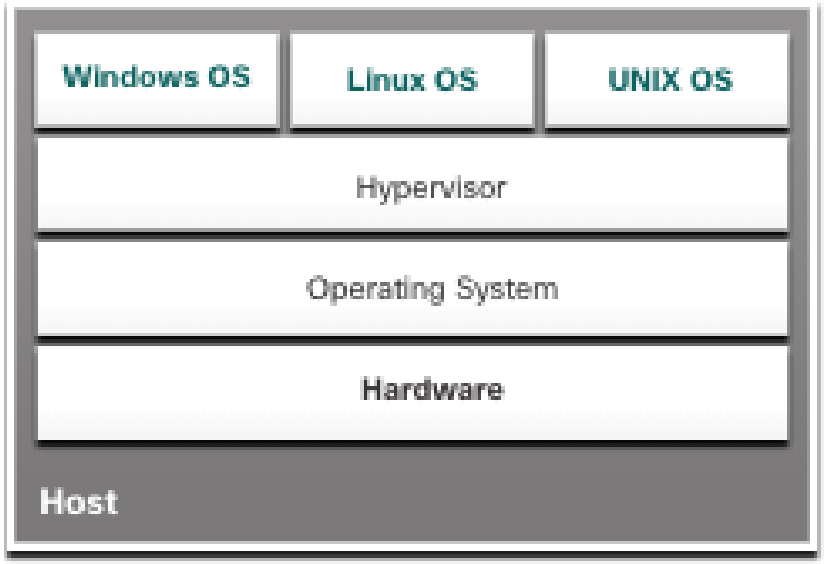
- At each of these layers of abstraction, some type of programming code is used as an interface between the layer below and the layer above.
- A hypervisor is installed between the firmware and the OS. The hypervisor can support multiple instances of OSs.





Type 2 Hypervisors

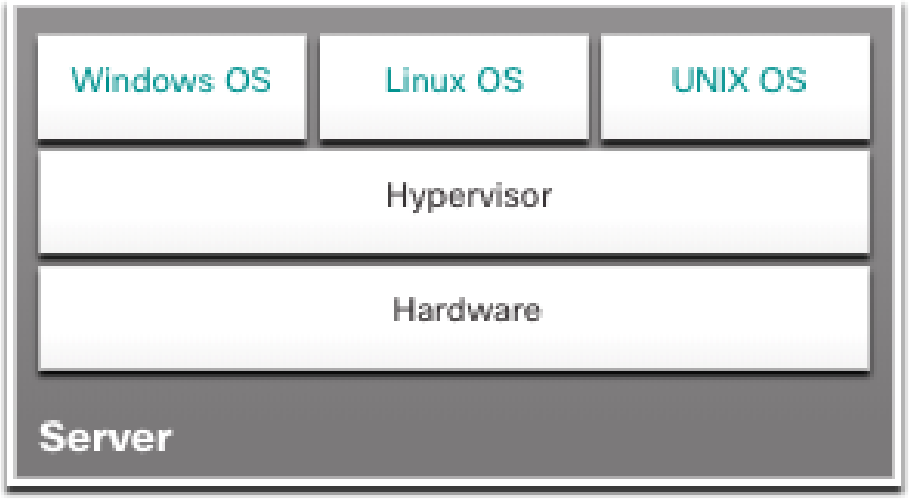
- A Type 2 hypervisor is software that creates and runs VM instances. The computer, on which a hypervisor is supporting one or more VMs, is a host machine. Type 2 hypervisors are also called hosted hypervisors.
- A big advantage of Type 2 hypervisors is that management console software is not required.



Virtual Network Infrastructure

Type 1 Hypervisors

- Type 1 hypervisors are also called the “bare metal” approach because the hypervisor is installed directly on the hardware. Type 1 hypervisors are usually used on enterprise servers and data center networking devices.
- With Type 1 hypervisors, the hypervisor is installed directly on the server or networking hardware. Then, instances of an OS are installed on the hypervisor, as shown in the figure. Type 1 hypervisors have direct access to the hardware resources. Therefore, they are more efficient than hosted architectures. Type 1 hypervisors improve scalability, performance, and robustness.

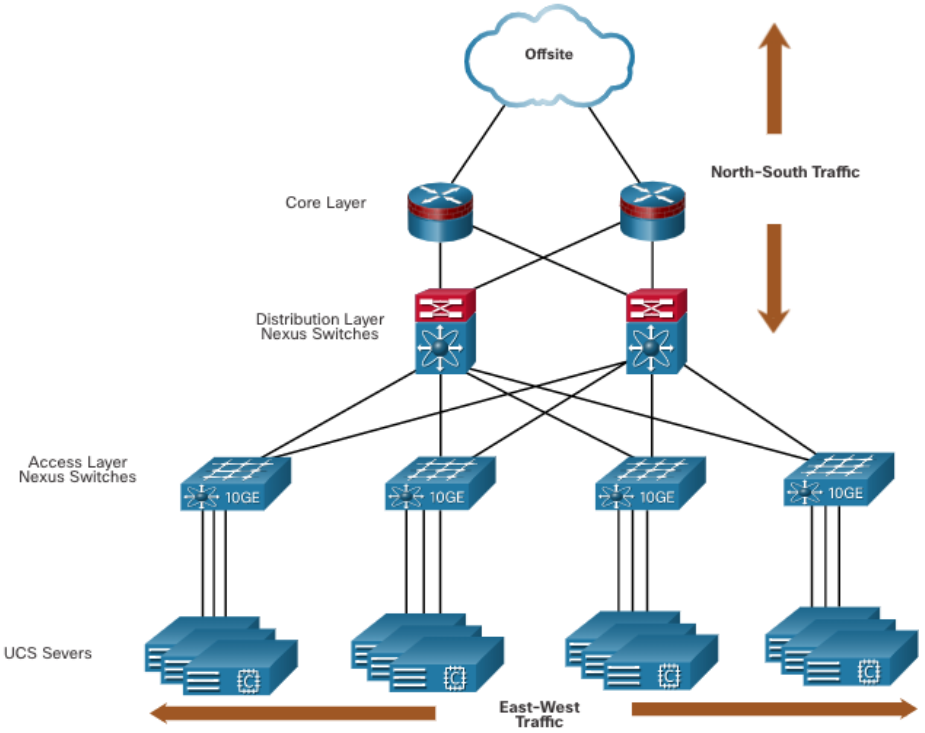


Installing a VM on a Hypervisor

- Type 1 hypervisors require a “management console” to manage the hypervisor. Management software is used to manage multiple servers using the same hypervisor. The management console can automatically consolidate servers and power on or off servers as required.
- The management console provides recovery from hardware failure. If a server component fails, the management console automatically moves the VM to another server. Cisco Unified Computing System (UCS) Manager controls multiple servers and manages resources for thousands of VMs.
- Some management consoles also allow server over allocation. Over allocation is when multiple OS instances are installed, but their memory allocation exceeds the total amount of memory that a server has. Over allocation is a common practice because all four OS instances rarely require the all their allocated resources at any one moment.

The Complexity of Network Virtualization

- Server virtualization hides server resources. This can create problems when using traditional network architectures.
- VMs are movable, and the network administrator must be able to add, drop, and change network resources and profiles to support their mobility. This process would be manual and time-consuming with traditional network switches.
- Traffic flows differ from the traditional client-server model. Typically, there is a considerable amount of traffic being exchanged between virtual servers (East-West traffic) that changes in location and intensity over time. North-South traffic is typically traffic destined for offsite locations such as another data center, other cloud providers, or the internet.



The Complexity of Network Virtualization (Cont.)

- Dynamic ever-changing traffic requires a flexible approach to network resource management. Existing network infrastructures can respond to changing requirements related to the management of traffic flows by using Quality of Service (QoS) and security level configurations for individual flows. However, in large enterprises using multivendor equipment, each time a new VM is enabled, the necessary reconfiguration can be very time-consuming.
- The network infrastructure can also benefit from virtualization. Network functions can be virtualized. Each network device can be segmented into multiple virtual devices that operate as independent devices. Examples include subinterfaces, virtual interfaces, VLANs, and routing tables. Virtualized routing is called virtual routing and forwarding (VRF).

Software-Defined Networking

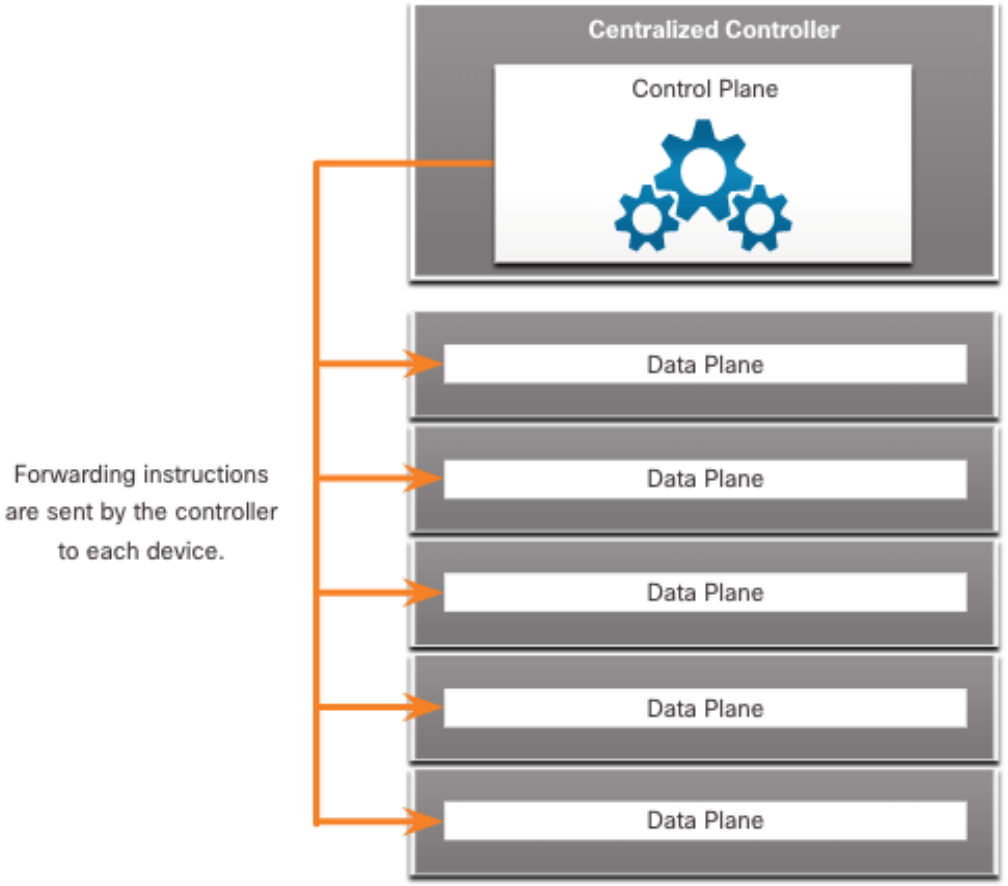
Control Plane and Data Plane

A network device contains the following planes:

- **Control plane** - This is typically regarded as the brains of a device. It is used to make forwarding decisions. The control plane contains Layer 2 and Layer 3 route forwarding mechanisms, such as routing protocol neighbor tables and topology tables, IPv4 and IPv6 routing tables, STP, and the ARP table. Information sent to the control plane is processed by the CPU.
- **Data plane** - Also called the forwarding plane, this plane is typically the switch fabric connecting the various network ports on a device. The data plane of each device is used to forward traffic flows. Routers and switches use information from the control plane to forward incoming traffic out the appropriate egress interface. Information in the data plane is typically processed by a special data plane processor without the CPU getting involved.

Control Plane and Data Plane (Cont.)

- CEF is an advanced, Layer 3 IP switching technology that enables forwarding of packets to occur at the data plane without consulting the control plane.
- SDN is basically the separation of the control plane and data plane. The control plane function is removed from each device and is performed by a centralized controller. The centralized controller communicates control plane functions to each device. Each device can now focus on forwarding data while the centralized controller manages data flow, increases security, and provides other services.



Control Plane and Data Plane (Cont.)

- The **management plane** is responsible for managing a device through its connection to the network.
- Network administrators use applications such as Secure Shell (SSH), Trivial File Transfer Protocol (TFTP), Secure FTP, and Secure Hypertext Transfer Protocol (HTTPS) to access the management plane and configure a device.
- The management plane is how you have accessed and configured devices in your networking studies. In addition, protocols like Simple Network Management Protocol (SNMP), use the management plane.

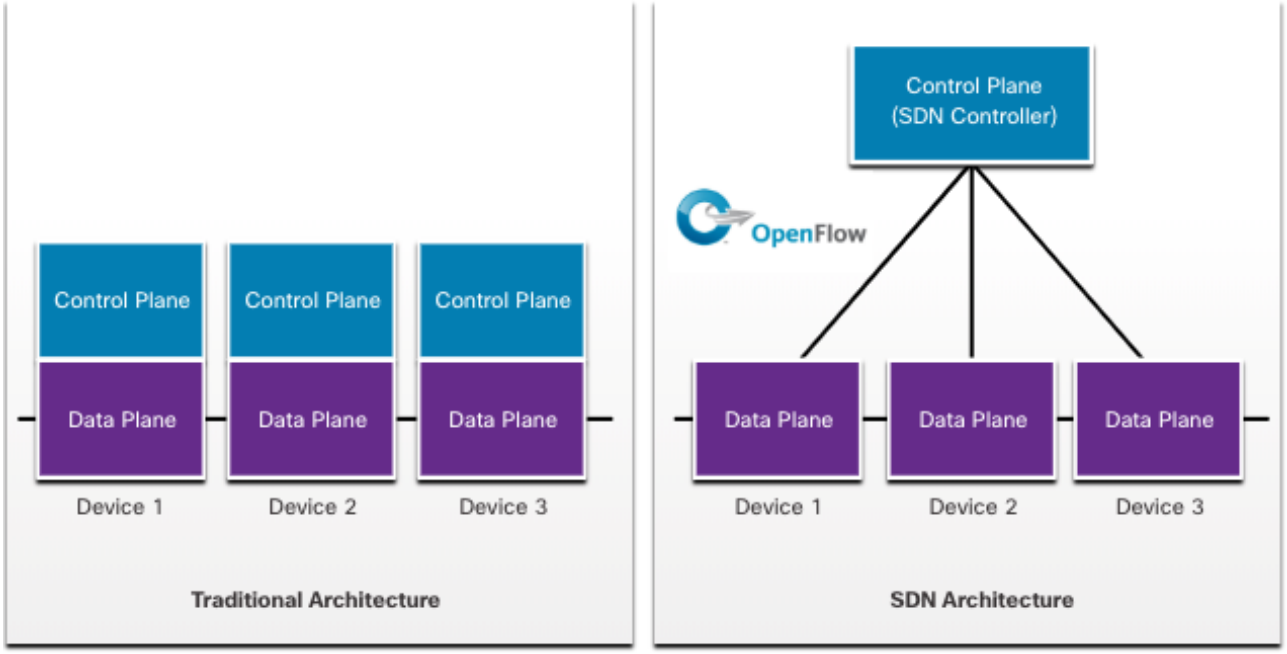
Network Virtualization Technologies

Components of SDN may include the following:

- **OpenFlow** - This approach was developed at Stanford University to manage traffic between routers, switches, wireless access points, and a controller. The OpenFlow protocol is a basic element in building SDN solutions.
- **OpenStack** - This approach is a virtualization and orchestration platform designed to build scalable cloud environments and provide an IaaS solution. OpenStack is often used with Cisco ACI. Orchestration in networking is the process of automating the provisioning of network components such as servers, storage, switches, routers, and applications.
- **Other components** - Other components include Interface to the Routing System (I2RS), Transparent Interconnection of Lots of Links (TRILL), Cisco FabricPath (FP), and IEEE 802.1aq Shortest Path Bridging (SPB).

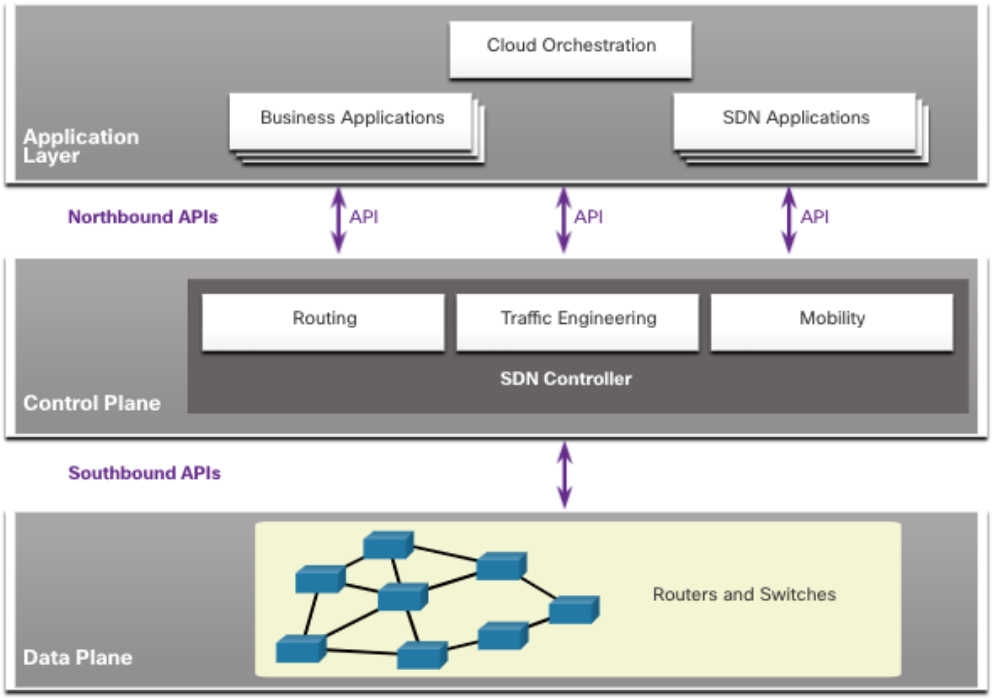
Traditional and SDN Architectures

In a traditional router or switch architecture, the control plane and data plane functions occur in the same device. Routing decisions and packet forwarding are the responsibility of the device operating system. In SDN, management of the control plane is moved to a centralized SDN controller. The figure compares traditional and SDN architectures.



Traditional and SDN Architectures (Cont.)

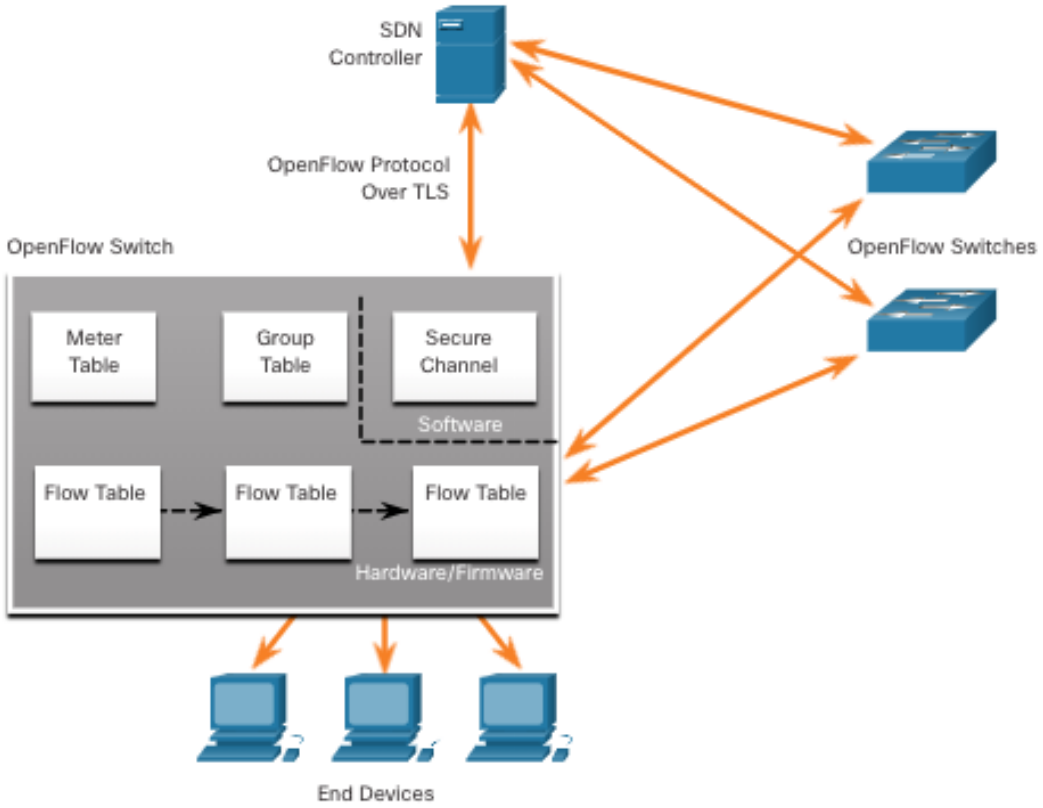
- The SDN controller is a logical entity that enables network administrators to manage and dictate how the data plane of switches and routers should handle network traffic. It orchestrates, mediates, and facilitates communication between applications and network elements.
- The complete SDN framework is shown in the figure. Note the use of Application Programming Interfaces (APIs). An API is a standardized definition of the proper way for an application to request services from another application.
- The SDN controller uses northbound APIs to communicate with the upstream applications, helping network administrators shape traffic and deploy services. The SDN controller uses southbound APIs to define the behavior of the data planes on downstream switches and routers. OpenFlow is a widely implemented southbound API.



Controllers

SDN Controller and Operations

- The SDN controller defines the data flows between the centralized control plane and the data planes on individual routers and switches.
- Each flow traveling through the network must first get permission from the SDN controller, which verifies that the communication is permissible according to the network policy.
- All complex functions are performed by the controller. The controller populates flow tables. Switches manage the flow tables.



SDN Controller and Operations (Cont.)

Within each switch, a series of tables implemented in hardware or firmware are used to manage the flows of packets through the switch. To the switch, a flow is a sequence of packets that matches a specific entry in a flow table.

The three table types shown in the previous figure are as follows:

- **Flow Table** - This table matches incoming packets to a particular flow and specifies the functions that are to be performed on the packets. There may be multiple flow tables that operate in a pipeline fashion.
- **Group Table** - A flow table may direct a flow to a Group Table, which may trigger a variety of actions that affect one or more flows.
- **Meter Table** - This table triggers a variety of performance-related actions on a flow including the ability to rate-limit the traffic.

Automation Overview

The Increase in Automation

These are some of the benefits of automation:

- Machines can work 24 hours a day without breaks, which results in greater output.
- Machines provide a more uniform product.
- Automation allows the collection of vast amounts of data that can be quickly analyzed to provide information which can help guide an event or process.
- Robots are used in dangerous conditions such as mining, firefighting, and cleaning up industrial accidents. This reduces the risk to humans.
- Under certain circumstances, smart devices can alter their behavior to reduce energy usage, make a medical diagnosis, and improve automobile driving safety.

Thinking Devices

- Many devices now incorporate smart technology to help to govern their behavior. This can be as simple as a smart appliance lowering its power consumption during periods of peak demand or as complex as a self-driving car.
- Whenever a device takes a course of action based on an outside piece of information, then that device is referred to as a smart device. Many devices that we interact with now have the word smart in their names. This indicates that the device has the ability to alter its behavior depending on its environment.
- In order for devices to “think”, they need to be programmed using network automation tools.

Data Formats

The Data Formats Concept

- Data formats are simply a way to store and exchange data in a structured format. One such format is called Hypertext Markup Language (HTML). HTML is a standard markup language for describing the structure of web pages.
- These are some common data formats that are used in many applications including network automation and programmability:
  - JavaScript Object Notation (JSON)
  - eXtensible Markup Language (XML)
  - YAML Ain’t Markup Language (YAML)
- The data format that is selected will depend on the format that is used by the application, tool, or script that you are using. Many systems will be able to support more than one data format, which allows the user to choose their preferred one.

Compare Data Formats

```
{
        "message": "success",
        "timestamp": 1560789260,
        "iss_position": {
                "latitude":
"25.9990",
        "longitude": "-132.6992"
        }
}
```

JSON Format

```
message: success
timestamp: 1560789260
iss_position:
        latitude: '25.9990'
        longitude: '-
132.6992'
```

YAML Format

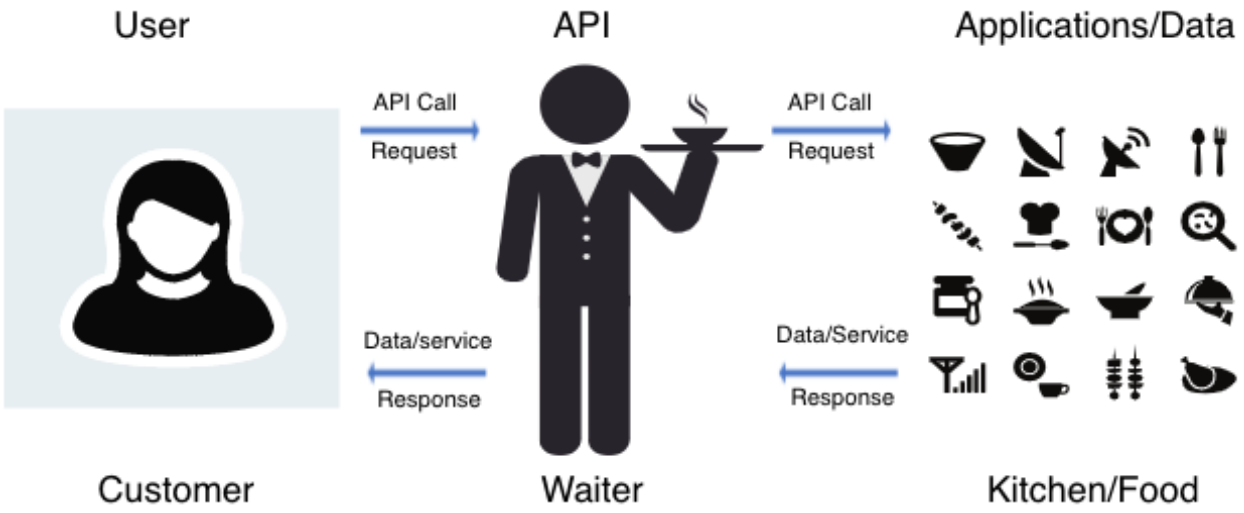
```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
<message>success</message>
<timestamp>1560789260</timestamp>
<iss_position>
   <latitude>25.9990</latitude>
   <longitude>-132.6992</longitude>
</iss_position>
</root>
```

XML Format

APIs

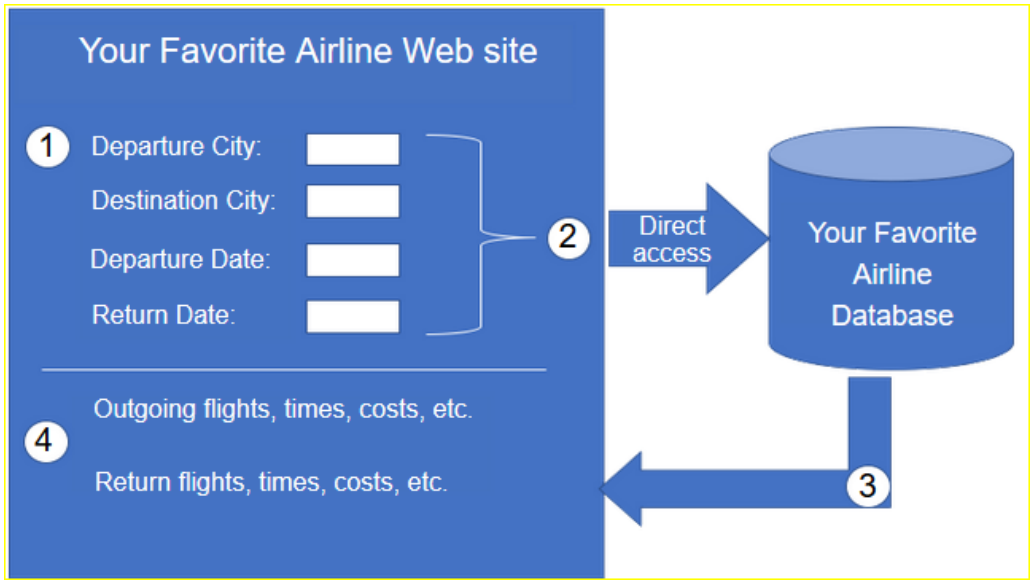
The API Concept

- 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.
- An API is similar to a waiter in a restaurant, as shown in the following figure.



An API Example

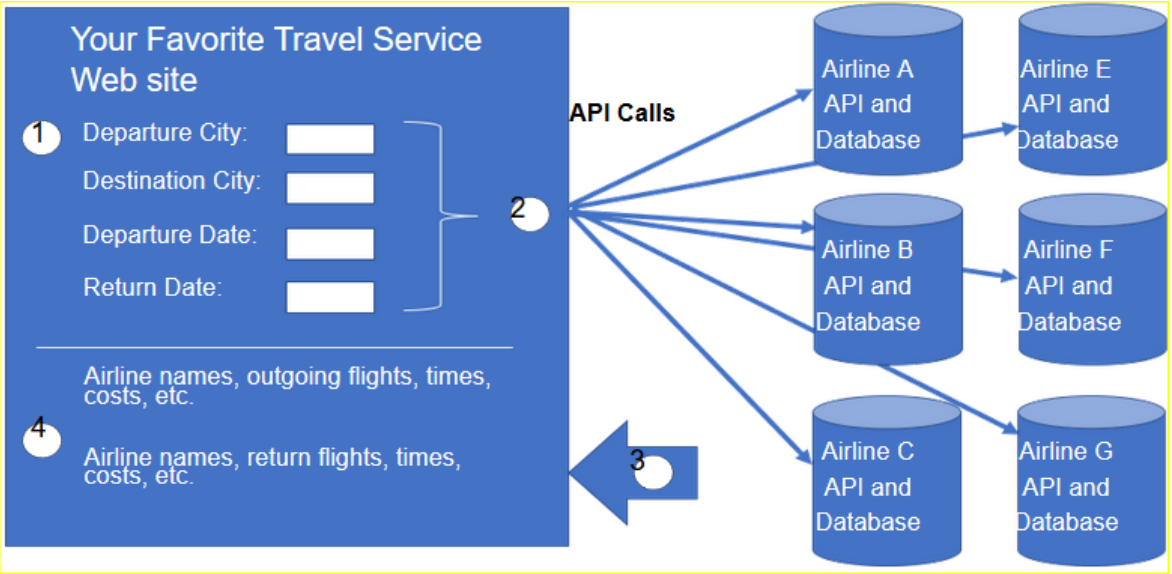
To really understand how APIs can be used to provide data and services, we will look at two options for booking airline reservations. The first option uses the web site of a specific airline. Using the airline's web site, the user enters the information to make a reservation request. The web site interacts directly with the airline's own database and provides the user with information matching the user's request.



An API Example (Cont.)

A travel site can access this same information, not only from a specific airline but a variety of airlines. In this case, the user enters in similar reservation information. The travel service web site interacts with the various airline databases using APIs provided by each airline. The travel service uses each airline API to request information from that specific airline, and then it displays the information from all the airlines on the its web page.

The API acts as a kind of messenger between the requesting application and the application on the server that provides the data or service. The message from the requesting application to the server where the data resides is known as an API call.



Types of Web Service APIs

A web service is a service that is available over the internet, using the World Wide Web. There are four types of web service APIs:

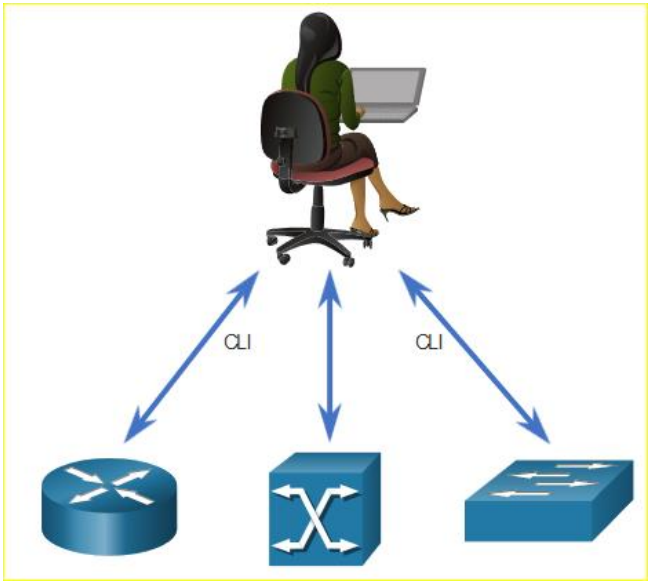
- Simple Object Access Protocol (SOAP)
- Representational State Transfer (REST)
- eXtensible Markup Language-Remote Procedure Call (XML-RPC)
- JavaScript Object Notation-Remote Procedure Call (JSON-RPC)

Characteristic	SOAP	REST	XML-RPC	JSON-RPC
Data Format	XML	JSON, XML, YAML, and others	XML	JSON
First released	1998	2000	1998	2005
Strengths	Well-established	Flexible formatting and most widely used	Well-established, simplicity	Simplicity

Configuration Management Tools

Traditional Network Configuration

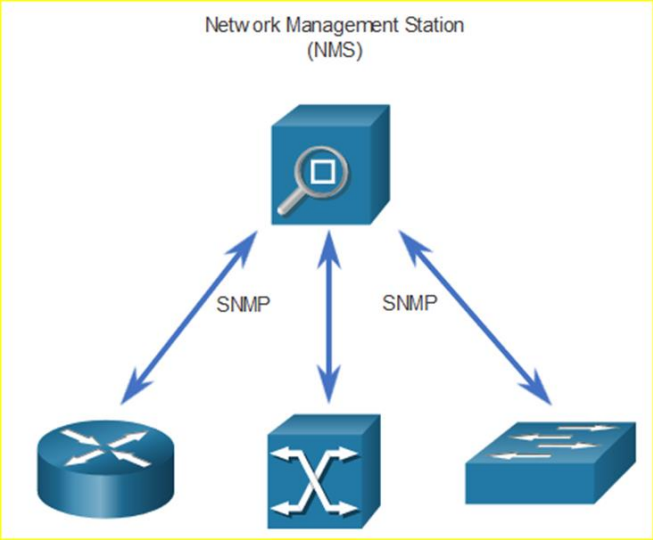
Network devices have traditionally been configured by a network administrator using the CLI. Whenever there is a change or new feature, the necessary configuration commands must be manually entered on all of the appropriate devices. This becomes a major issue on larger networks or with more complex configurations.



Traditional Network Configuration

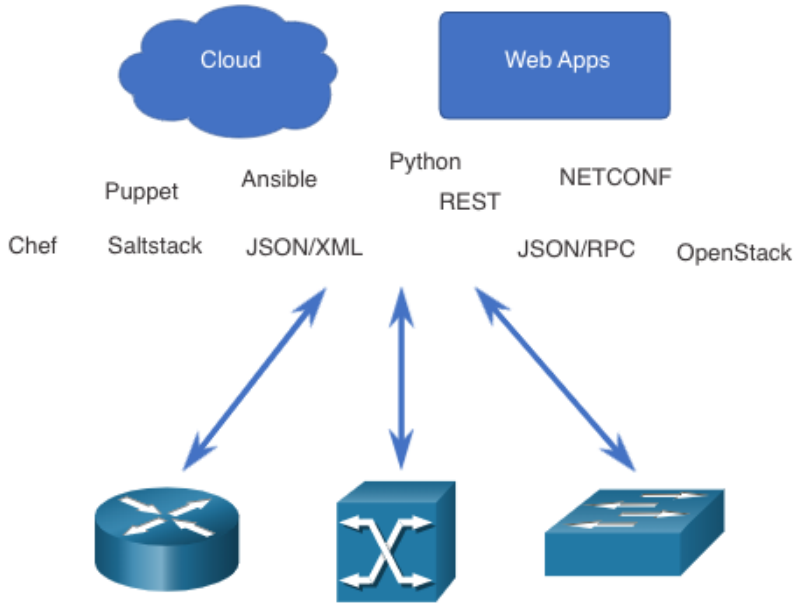
Simple Network Management Protocol (SNMP) lets administrators manage nodes on an IP network. With a network management station (NMS), network administrators use SNMP to monitor and manage network performance, find and solve network problems, and perform queries for statistics. SNMP is not typically used for configuration due to security concerns and difficulty in implementation.

You can also use APIs to automate the deployment and management of network resources. Instead of manually configuring ports, access lists, QoS, and load balancing policies, you can use tools to automate configurations.



Network Automation

We are rapidly moving away from a world where a network administrator manages a few dozen network devices, to one where they are deploying and managing a great number of complex network devices (both physical and virtual) with the help of software. This transformation is quickly spreading to all places in the network. There are new and different methods for network administrators to automatically monitor, manage, and configure the network. These include protocols and technologies such as REST, Ansible, Puppet, Chef, Python, JSON, XML, and more.



Configuration Management Tools

Configuration management tools make use of RESTful API requests to automate tasks and can scale across thousands of devices. These are some characteristics of the network that administrators benefit from automating:

- Software and version control
- Device attributes such as names, addressing, and security
- Protocol configurations
- ACL configurations

Configuration management tools typically include automation and orchestration. Automation is when a tool automatically performs a task on a system. Orchestration is the arranging of the automated tasks that results in a coordinate process or workflow.

Configuration Management Tools (Cont.)

There are several tools available to make configuration management easier:

- Ansible
- Chef
- Puppet
- SaltStack

The goal of all of these tools is to reduce the complexity and time involved in configuring and maintaining a large-scale network infrastructure with hundreds, even thousands of devices. These same tools can benefit smaller networks as well.

Compare Ansible, Chef, Puppet, and SaltStack

Ansible, Chef, Puppet, and SaltStack all come with API documentation for configuring RESTful API requests. All of them support JSON and YAML as well as other data formats. The following table shows a summary of a comparison of major characteristics of Ansible, Puppet, Chef, and SaltStack configuration management tools.

Characteristic	Ansible	Chef	Puppet	SaltStack
What programming language?	Python + YAML	Ruby	Ruby	Python
Agent-based or agentless?	Agentless	Agent-based	Supports both	Supports both
How are devices managed?	Any device can be "controller"	Chef Master	Puppet Master	Salt Master
What is created by the tool?	Playbook	Cookbook	Manifest	Pillar

Today end, **See you next week!**