

Configuration Management Tools

Video – Configuration Management Tools

As mentioned in the introduction to this module, setting up a network can be very time consuming. Configuration management tools can help you to automate the configuration of routers, switches, firewalls and many other aspects of your network.

Click play in the video to learn about configuration management tools.

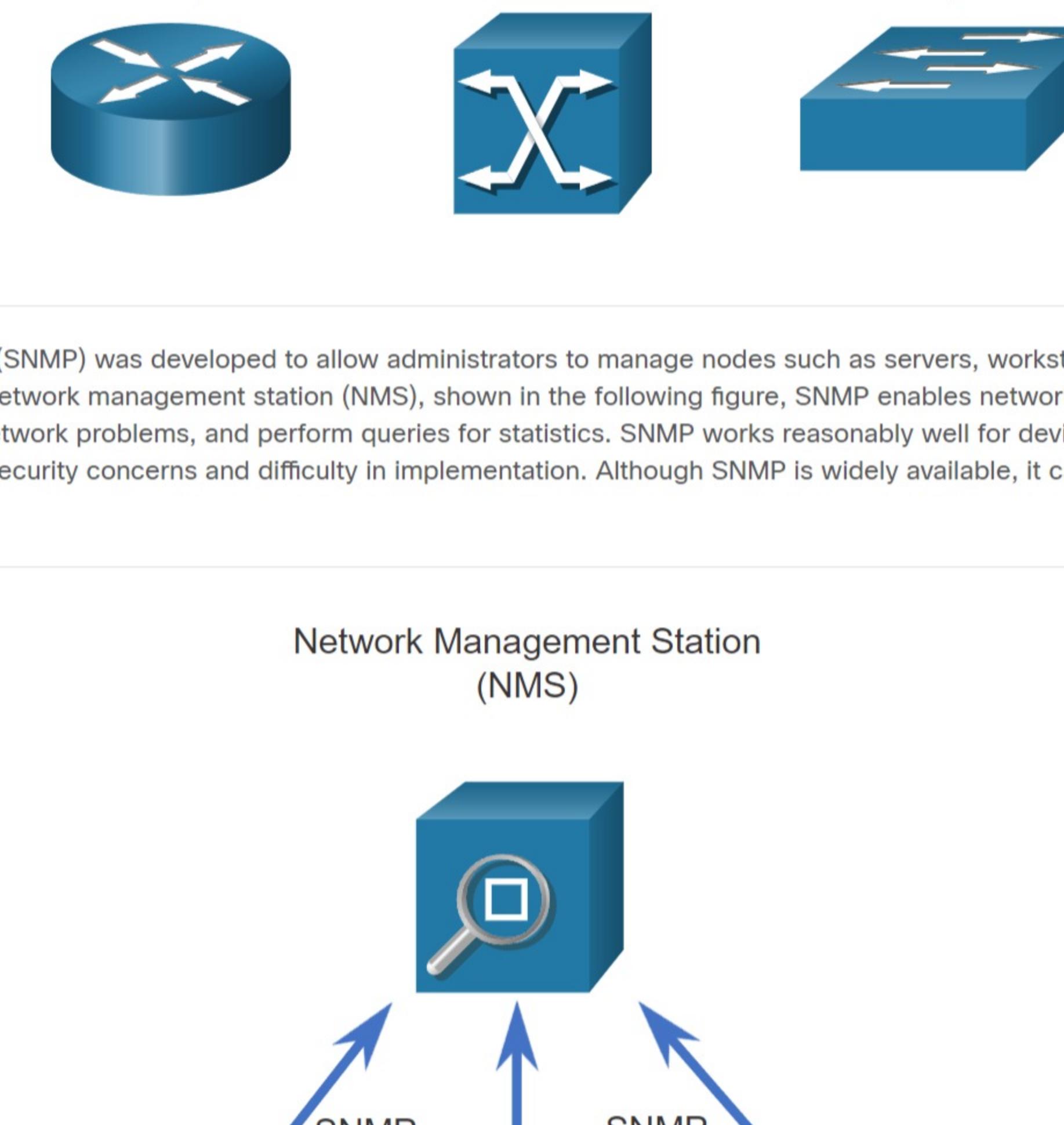
Video – Configuration Management

This video will cover the following:

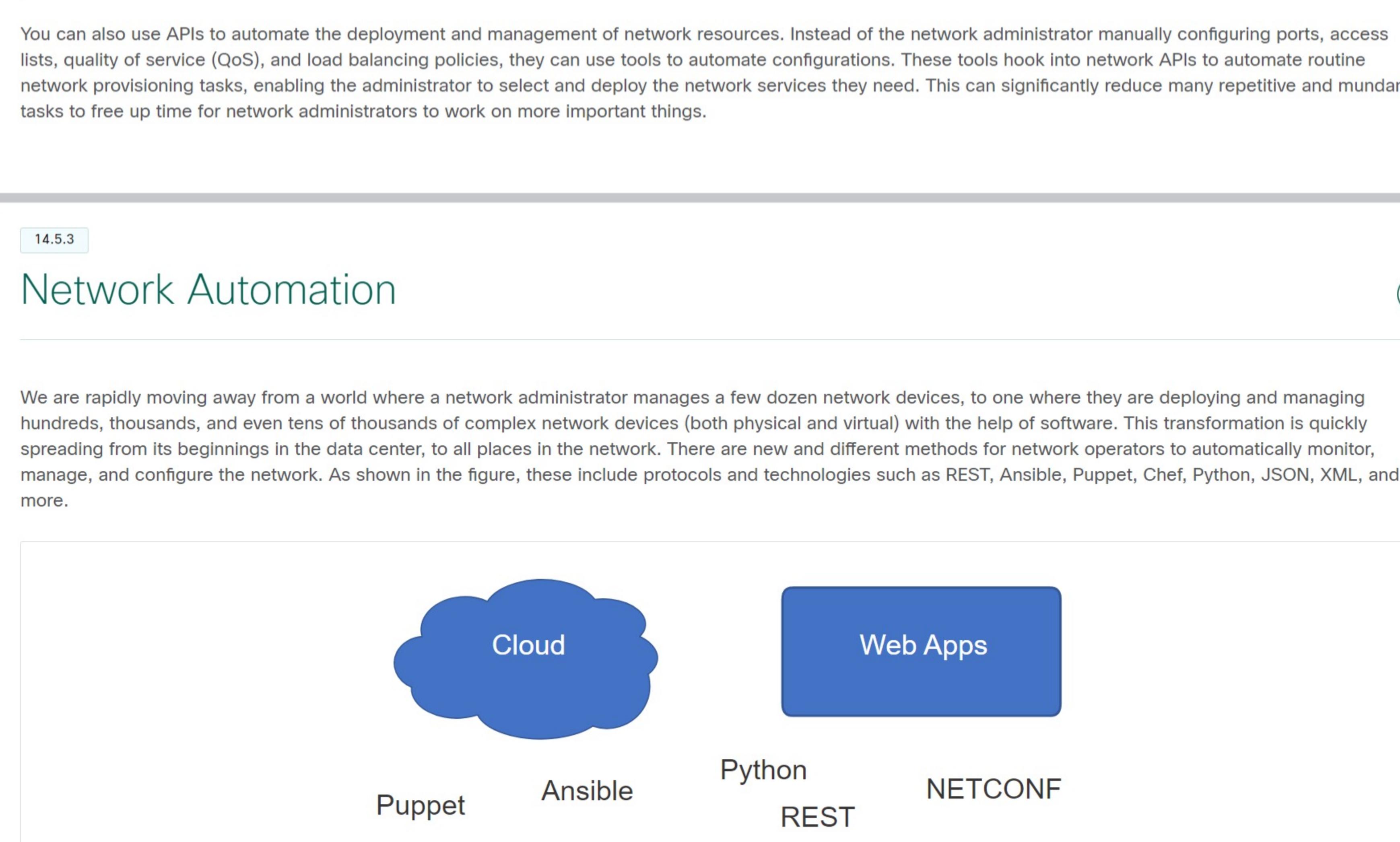
- Compare configuration management tools including Ansible, Puppet, Chef and SaltStack
- Review plays, tasks, modules, parameters, and variables in a sample playbook

Traditional Network Configuration

Network devices such as router, switches, and firewalls have traditionally been configured by a network administrator using the CLI, as shown in the figure. Whenever there is a change or new feature, the necessary configuration commands must be manually entered on all of the appropriate devices. In many cases, this is not only time-consuming, but can also be prone to errors. This becomes a major issue on larger networks or with more complex configurations.



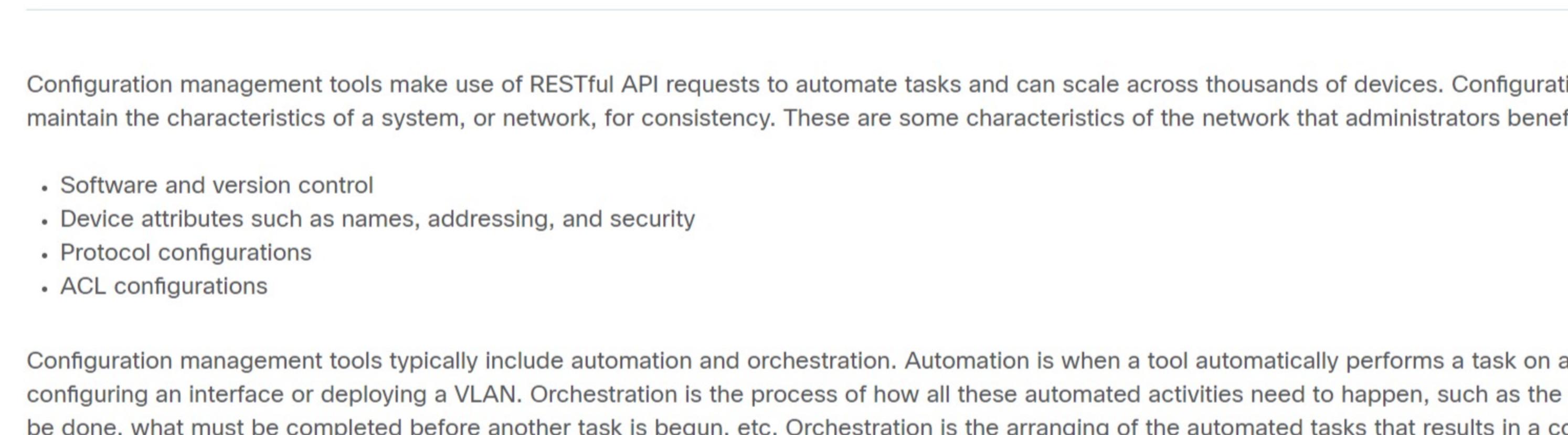
Simple Network Management Protocol (SNMP) was developed to allow administrators to manage nodes such as servers, workstations, routers, switches, and security appliances, on an IP network. Using a network management station (NMS), shown in the following figure, SNMP enables network administrators to monitor and manage network performance, find and solve network problems, and perform queries for statistics. SNMP works reasonably well for device monitoring. However, it is not typically used for configuration due to security concerns and difficulty in implementation. Although SNMP is widely available, it cannot serve as an automation tool for today's networks.



You can also use APIs to automate the deployment and management of network resources. Instead of the network administrator manually configuring ports, access lists, quality of service (QoS), and load balancing policies, they can use tools to automate configurations. These tools hook into network APIs to automate routine network provisioning tasks, enabling the administrator to select and deploy the network services they need. This can significantly reduce many repetitive and mundane tasks to free up time for network administrators to work on more important things.

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 hundreds, thousands, and even tens of thousands of complex network devices (both physical and virtual) with the help of software. This transformation is quickly spreading from its beginnings in the data center, to all places in the network. There are new and different methods for network operators to automatically monitor, manage, and configure the network. As shown in the figure, these include protocols and technologies such as REST, Ansible, Puppet, Chef, Python, JSON, XML, and more.



Configuration Management Tools

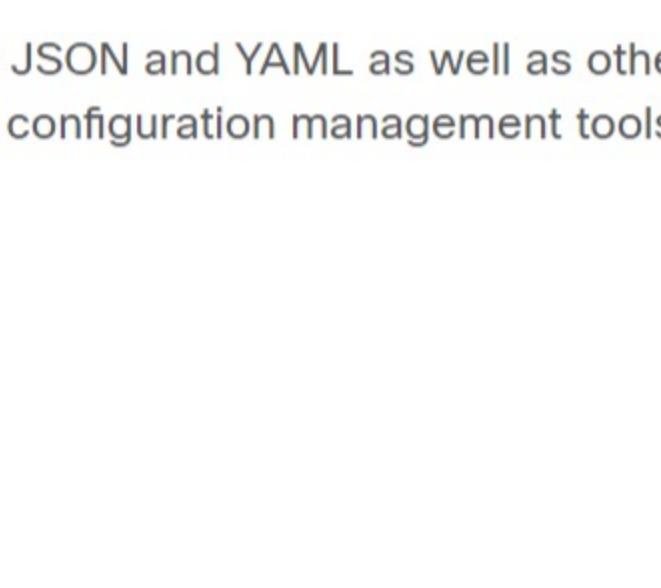
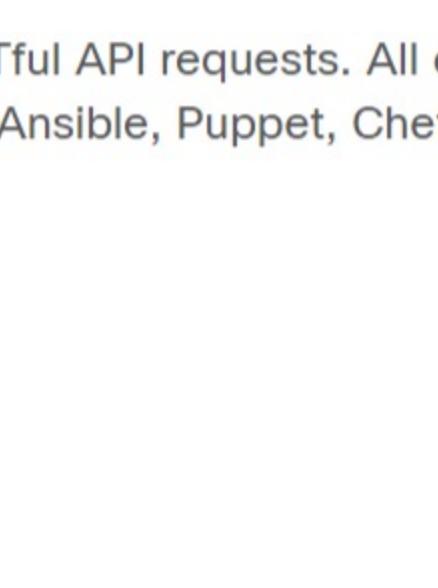
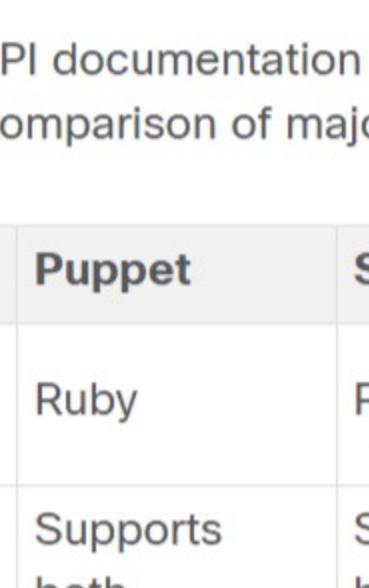
Configuration management tools make use of RESTful API requests to automate tasks and can scale across thousands of devices. Configuration management tools maintain the characteristics of a system, or network, for consistency. These are some characteristics of the network that administrators benefit 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. This might be configuring an interface or deploying a VLAN. Orchestration is the process of how all these automated activities need to happen, such as the order in which they must be done, what must be completed before another task is begun, etc. Orchestration is the arranging of the automated tasks that results in a coordinate process or workflow.

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

• What programming language? – Ansible and SaltStack are both built on Python whereas Puppet and Chef are built on Ruby. Similar to Python, Ruby is an open-source programming language that is cross-platform. However, Ruby is typically considered a more difficult language to learn than Python.

• Agent-based or agentless? – Configuration management is either agent-based or agentless. Agent-based configuration management is "pull-based", meaning the agent on the managed device periodically connects with the master for its configuration information. Changes are done on the master and pulled down and executed by the device. Agentless configuration management is "push-based." A configuration script is run on the master. The master connects to the device and executes the tasks in the script. Of the four configuration tools in the table, only Ansible is agentless.

• How are devices managed? – This lies with a device called the Master in Puppet, Chef and SaltStack. However, because Ansible is agentless, any computer can be the controller.

• What is created by the tool? – Network administrators use configuration management tools to create a set of instructions to be executed. Each tool has its own name for these instructions: Playbook, Cookbook, Manifest, and Pillar. Common to each of this is specification of a policy or a configuration that is to be applied to devices. Each device type might have its own policy. For example, all Linux servers might get the same basic configuration and security policy.

Check Your Understanding - Configuration Management

Check your understanding of configuration management by choosing the BEST answer to the following questions.

1. Which of the following are not typically used as configuration tools? (Choose two.)

- API
- Ansible
- Chef
- Puppet
- SaltStack
- SNMP

2. Identify the term for the following definition in relationship to configuration management tools: Programmatically performing a task on a system such as configuring an interface or deploying a VLAN.

- Version control
- Automation
- Orchestration
- Network management

3. Identify the term for the following definition in relationship to configuration management tools: The process of how all automated activities need to happen, such as the order they must be done and what must be completed before another task is begun.

- Version control
- Automation
- Orchestration
- Network management

4. True or False: Agentless means that the controller or master pushes the configuration to the controlled device.

- True
- False

5. Which of the following configuration management tools use Python? (Choose all that apply.)

- Ansible
- Chef
- Puppet
- SaltStack

Check

Show Me

Reset