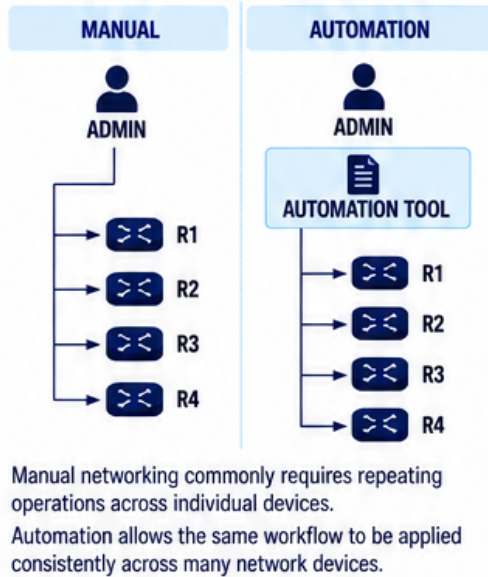


NETWORK AUTOMATION QUICK FACTS

Concept	Purpose
 Automation	Execute Network Tasks Automatically
 Programmability	Control Networks Through Software
 Python	Scripting & Automation Logic
 Ansible	Agentless Automation
 APIs	Programmatic Communication
 Git	Version Control

MANUAL vs AUTOMATED NETWORKING



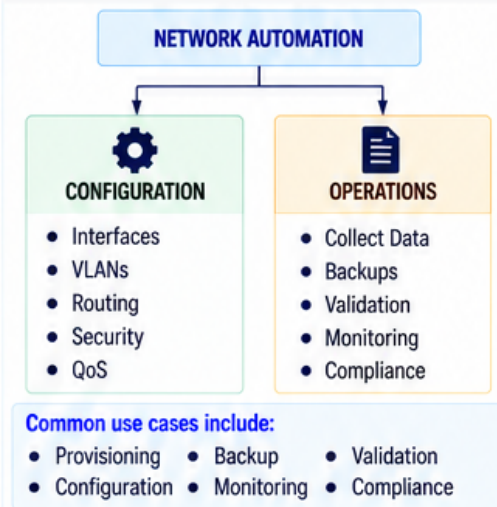
WHY NETWORK AUTOMATION?

Manual Networking	Network Automation
Repetitive Tasks	Repeatable Workflows
Device-by-Device	Multi-Device
Slower at Scale	Faster at Scale
Human Error Risk	Greater Consistency
Harder to Standardize	Standardized Processes

Automation improves speed, scale, consistency and repeatability, but changes still need testing and validation.

 Speed  Scale  Consistency  Repeatability

COMMON AUTOMATION TASKS



COMMON AUTOMATION TOOLS

Tool	Main Role
 Python	Scripting & Logic
 Ansible	Configuration Automation
 Git	Version Control
 Jinja2	Configuration Templates
 APIs	Programmatic Interaction
 YAML / JSON	Structured Data

Different tools can be combined into a complete automation workflow.

PYTHON IN NETWORK AUTOMATION

Python is commonly used to build automation logic and interact with devices, controllers and APIs.

PYTHON SCRIPT

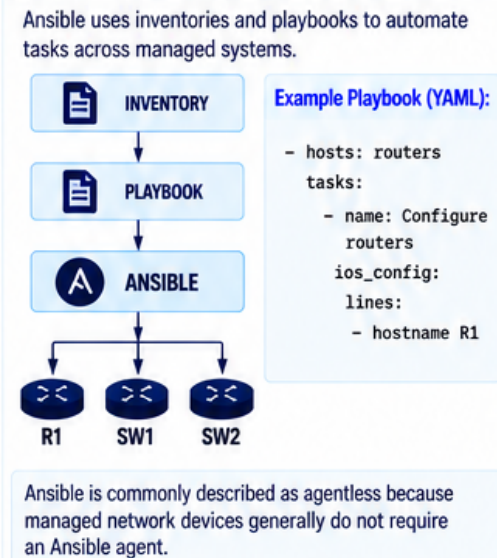
- Read Data
- Apply Logic
- Connect / Call API
- Make Changes
- Verify Results

Example:

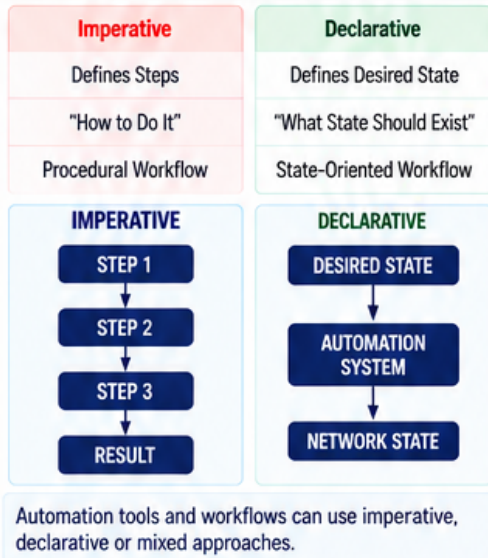
```
for device in devices:  
    configure(device)  
  
    # verify result  
  
    print("Done")
```

Python provides flexibility for custom workflows, data processing and integration.

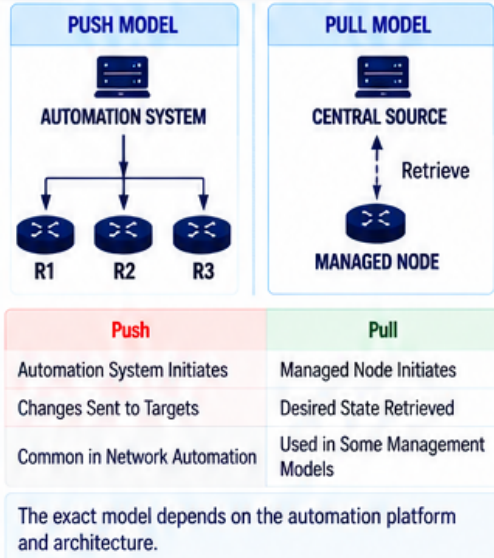
ANSIBLE AUTOMATION



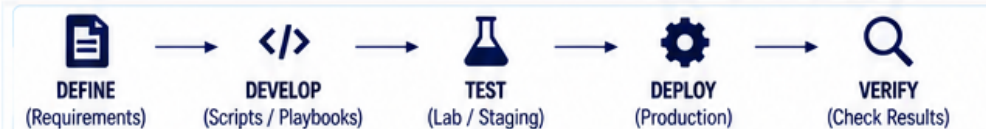
IMPERATIVE vs DECLARATIVE



PUSH vs PULL



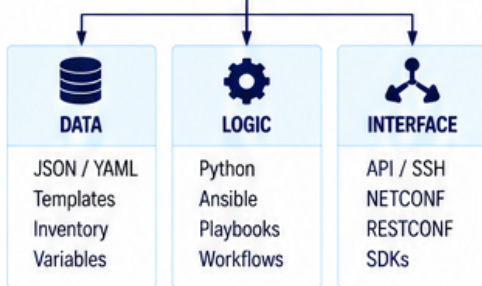
AUTOMATION WORKFLOW



Testing and verification are essential parts of network automation; successful execution does not automatically mean the network reached the intended state.

AUTOMATION BUILDING BLOCKS

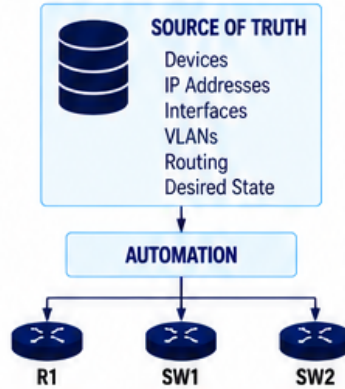
AUTOMATION



A network automation system combines data, automation logic and methods for interacting with network infrastructure.

SOURCE OF TRUTH

A source of truth stores authoritative information used by automation workflows.



The source of truth helps keep automation data centralized and consistent.

STRUCTURED DATA

Automation commonly represents network information using structured data formats.

JSON	YAML
Common with APIs	Common with Automation Tools
{ } Syntax	Indentation-Based
Machine-Friendly	Human-Readable
Key / Value Structure	Key / Value Structure

JSON Example:

```
{
  "vlan": 10,
  "name": "Users"
}
```

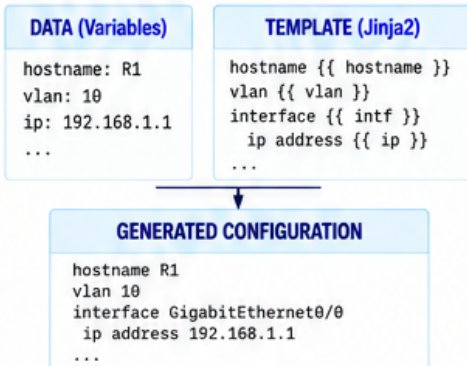
YAML Example:

```
vlan: 10
name: Users
```

Structured data makes information easier for software to parse and process.

CONFIGURATION TEMPLATES

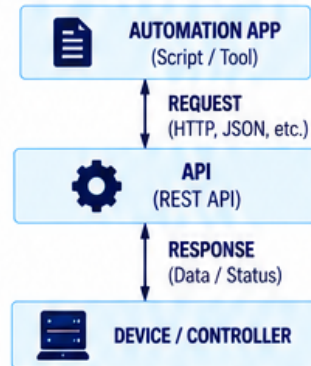
Templates separate variable data from reusable configuration structure.



Jinja2 is commonly used for configuration templating in network automation.

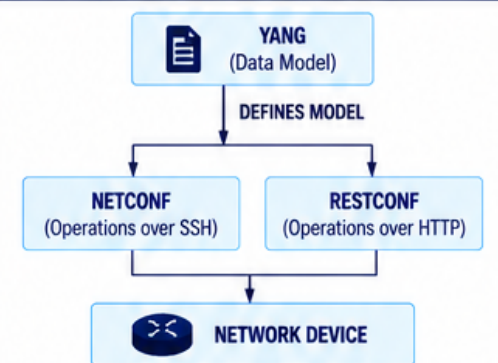
APIs IN AUTOMATION

APIs allow automation software to interact programmatically with devices, controllers and services.



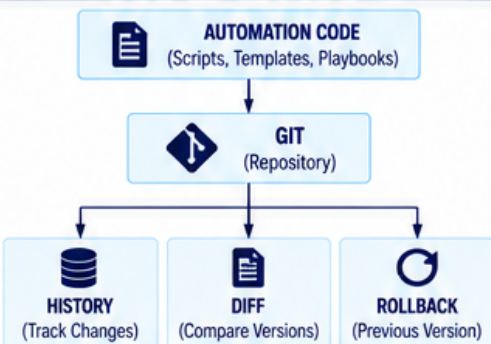
Common API operations include retrieving information, changing configuration and triggering actions.

MODEL-DRIVEN AUTOMATION



YANG defines structured network data models. NETCONF and RESTCONF can provide programmatic access to YANG-modeled data. This enables automation to work with structured data rather than relying only on CLI text.

GIT & VERSION CONTROL



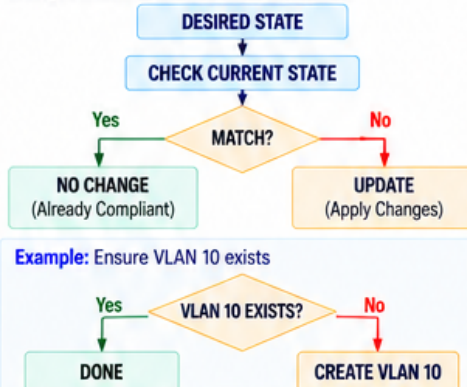
Common files stored in Git:

Scripts • Playbooks • Templates • Data Files

Git can track automation code and provides history, comparison and the ability to revert to earlier versions. (Does not automatically roll back live network configuration.)

IDEMPOTENCY

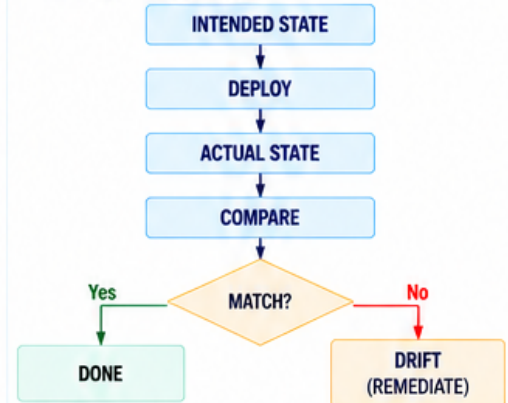
An idempotent automation operation can be executed repeatedly without producing unnecessary additional changes once the desired state is reached.



Idempotency is especially useful in state-oriented configuration automation.

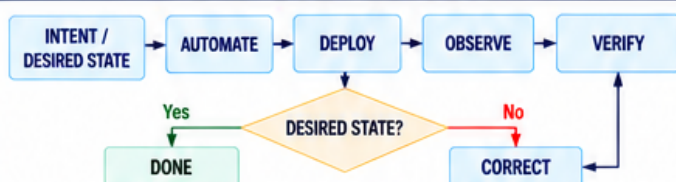
VALIDATION & STATE

Automation should verify that the actual network state matches the intended state.



Configuration success alone does not guarantee operational success.

CLOSED-LOOP AUTOMATION



Closed-loop automation combines deployment with observation, validation and corrective action.

Not every automation system is closed-loop; this is an advanced automation model.

COMPLETE NETWORK AUTOMATION FLOW

