

Configuration Management with Ansible and Terraform (CMAT)



Getting Started with Configuration Management



Learning Objectives

By the end of this lesson, you will be able to:

- Illustrate the fundamentals of configuration management and its benefits to improve system reliability and consistency
- Identify the different configuration management tools to enhance security and ensure compliance with regulatory requirements
- Outline the concept of Infrastructure as Code (IaC) to enhance the organizations operational efficiency and enable more agile and reliable IT operations
- Identify the features of Ansible to automate repetitive tasks, which boosts efficiency and reduces the potential for human error

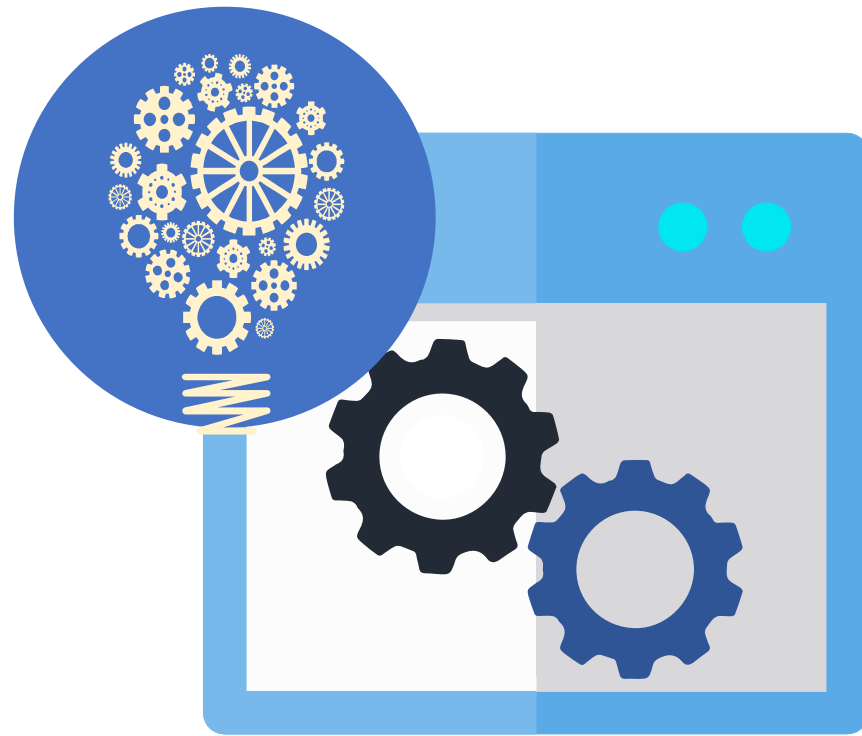




Understanding Configuration Management (CM)

Introduction to Configuration Management (CM)

It is a process in systems engineering that ensures a product's performance and its functional and physical attributes align with its requirements, design, and operational information.



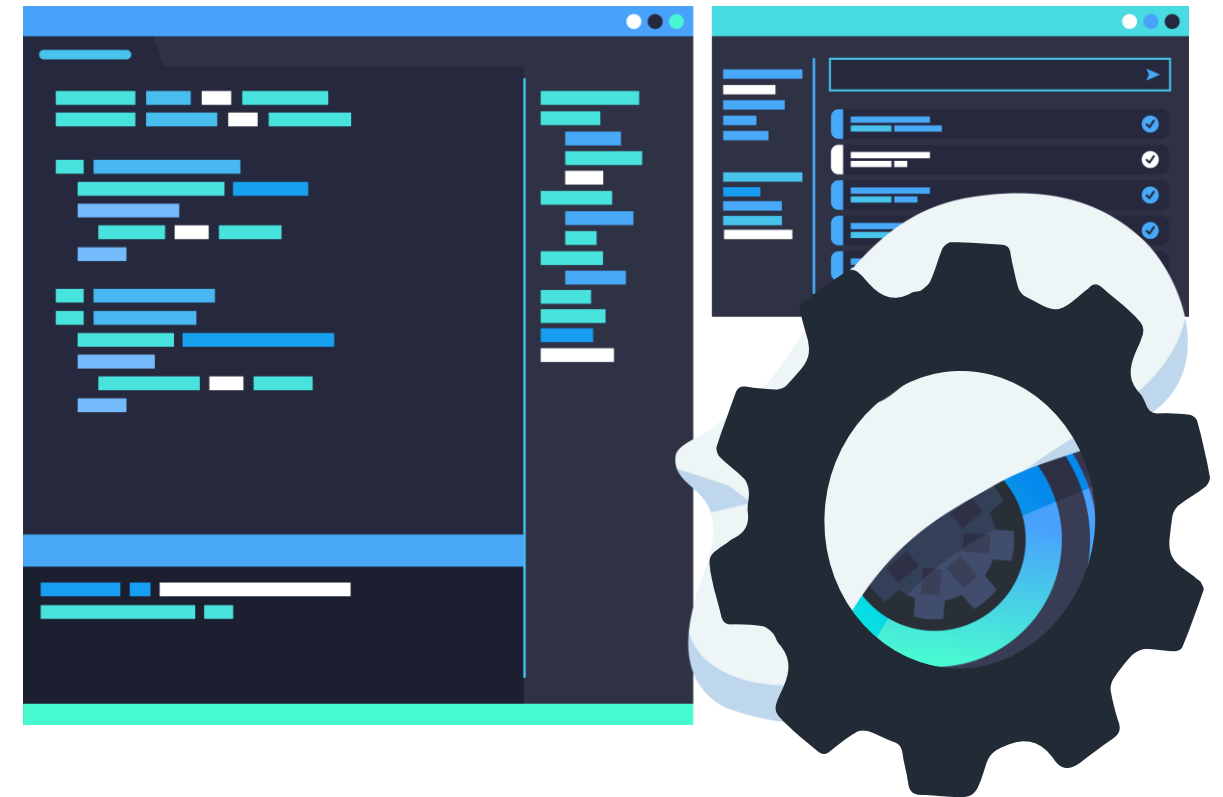
With this process, the system's expected performance can be assured, even after undergoing numerous changes over time.

Why Configuration Management?

Minor inconsistencies in IT systems can lead to significant configuration drift. Configuration management addresses these issues by implementing controlled and documented changes to the systems.

CM processes provide:

1. Consistency and standardization
2. Change management
3. Traceability
4. Enhanced collaboration
5. Improved quality control
6. Scalability
7. Disaster recovery



It ensures consistent and accurate configurations of systems across complex environments, where changes are frequent, collaboration is necessary, and compliance with regulations is crucial.

Configuration Management Scope

Software and Server Configurations: Configuration management tools (CMTs) manage both software and server settings for comprehensive configuration management.

Efficiency: CMTs save time and boost productivity by automating configuration management.

Configuration Items: CMTs handle various items like application files, software packages, and installations.

Configuration Management Process

Three operations comprise the configuration management process:

Configuration identification:

It identifies the correct configuration that the configuration management tool needs to manage.



Configuration regulation:

It governs how changes to the configuration are implemented for the application software.



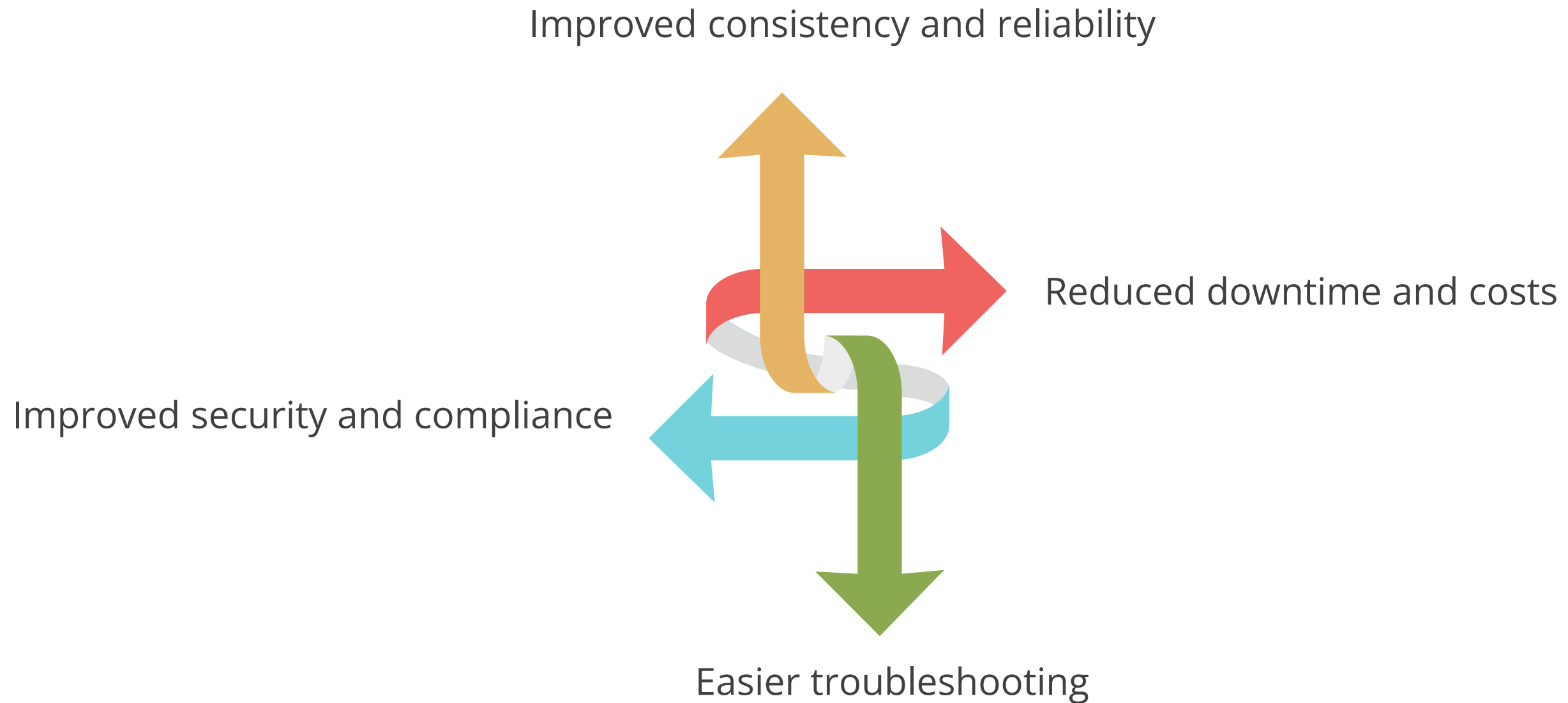
Configuration compliance:

It audits and ensures compliance with configuration changes made to the application software.



Benefits of Configuration Management

Configuration management offers the following benefits:



Automating Configuration Management

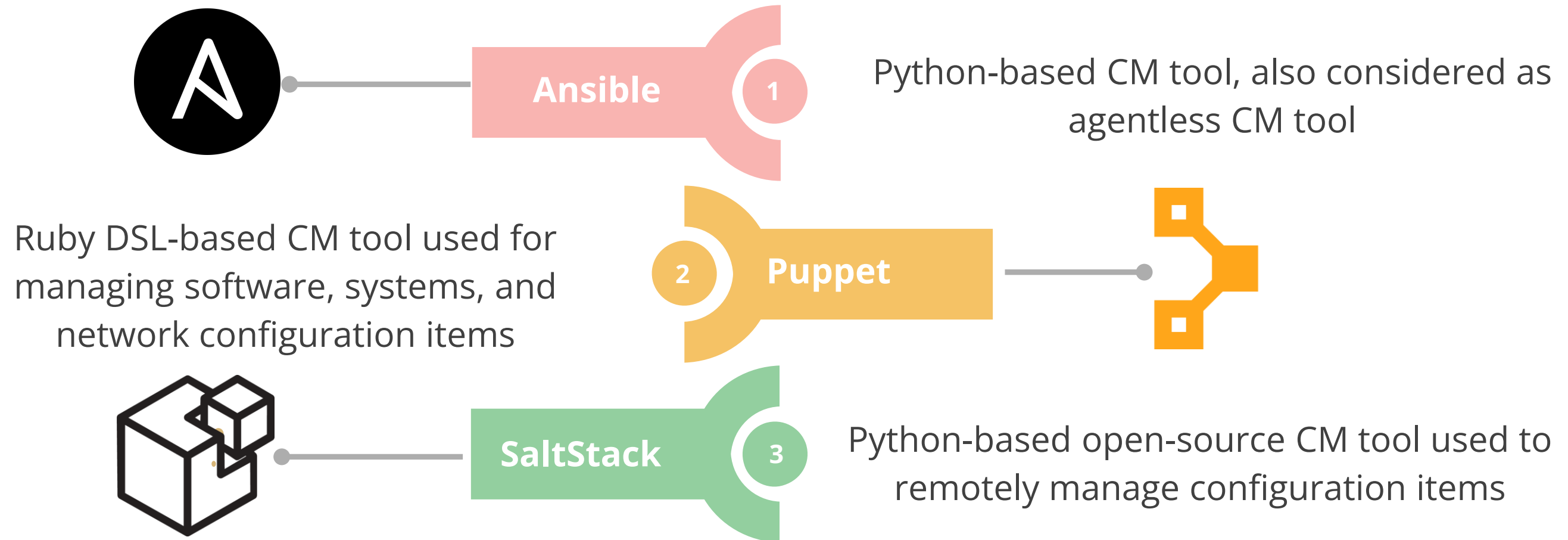
The objective of configuration management is to maintain computers in their desired state. This was traditionally managed manually or through proprietary programming by system administrators.



Automation involves using software to perform operations such as configuration management to reduce costs, time, complexity, and effort.

Configuration Management Tools

The following are the configuration management tools:



Configuration Management Tools



Chef

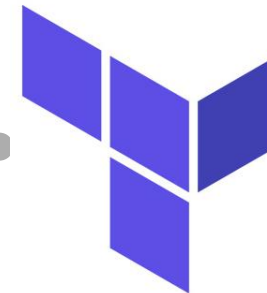
4

Ruby-based CM tool, which has integration with most of the cloud-based platforms

CM tool, designed to define and manage infrastructure resources in a declarative way

5

Terraform



Rudder

6

CM tool that builds on top of CFEngine, providing a web interface for policy management and reporting

Features of Configuration Management Tools

Version control

Tracks change to configurations, rolls back if necessary, and audits historical states

Declarative configuration

Defines the desired state of systems, which the tool automatically configures

Policy enforcement

Ensures compliance with organizational policies and standards, which all systems must adhere to

Inventory management

Keeps a centralized record of all managed systems and their configurations

Features of Configuration Management Tools

- Patch management

Automates the deployment of security patches and updates for software

- Reporting and auditing

Generates reports on system configurations and monitors compliance

- Application deployment

Automates the deployment and setup of applications

- Reducing Errors

Reduces errors caused by manual configurations

Quick Check



You are an IT manager responsible for managing multiple environments (development, testing, and production) in your organization. Recently, you have noticed discrepancies between these environments, leading to deployment issues. Which of the following actions would most effectively address this problem through automation?

- A. Increasing the budget for manual quality assurance testing
- B. Hiring more IT staff to manually configure each environment
- C. Implementing an automated configuration management tool to ensure consistent setup
- D. Scheduling weekly meetings to review environmental discrepancies



Understanding Infrastructure as Code

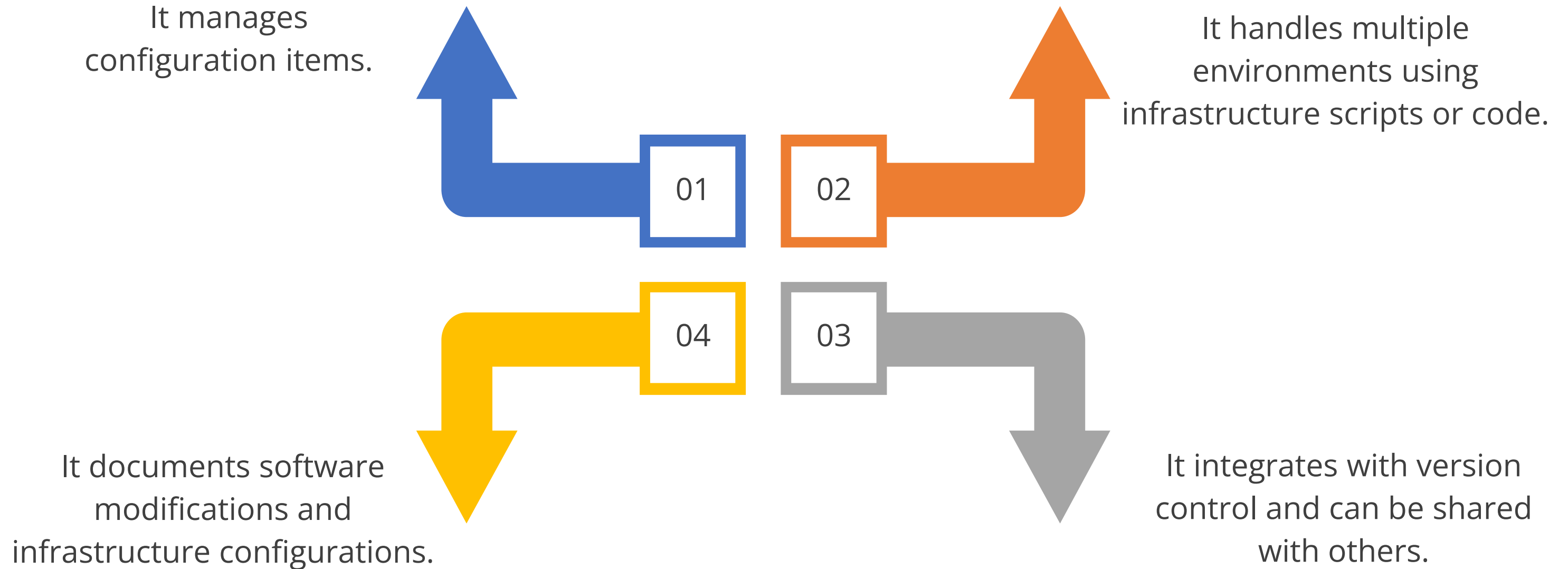
Infrastructure as Code (IaC)

It is the approach of managing and provisioning computer data centers through machine-readable definition files.



Role of Infrastructure as Code

The following are the key roles of IaC:



Declarative vs. Imperative Approaches

Declarative approach

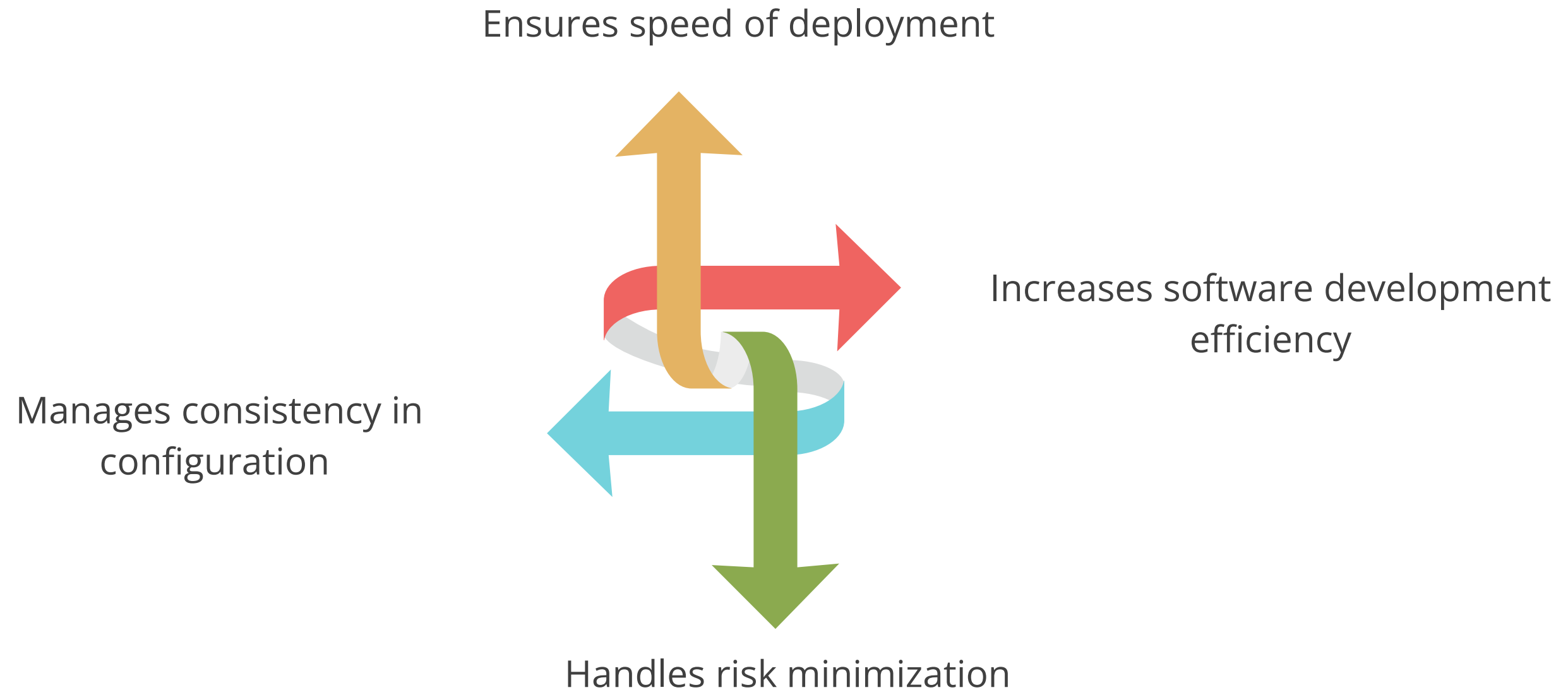
In a declarative approach, the IaC tool will set up the system for the user, defining the ideal system state, including the services required and any qualities they must have.

Imperative approach

An imperative approach defines the specific commands that must be executed in a precise order to create the intended configuration.

Benefits of Infrastructure as Code

The IaC approach offers the following benefits:



Quick Check



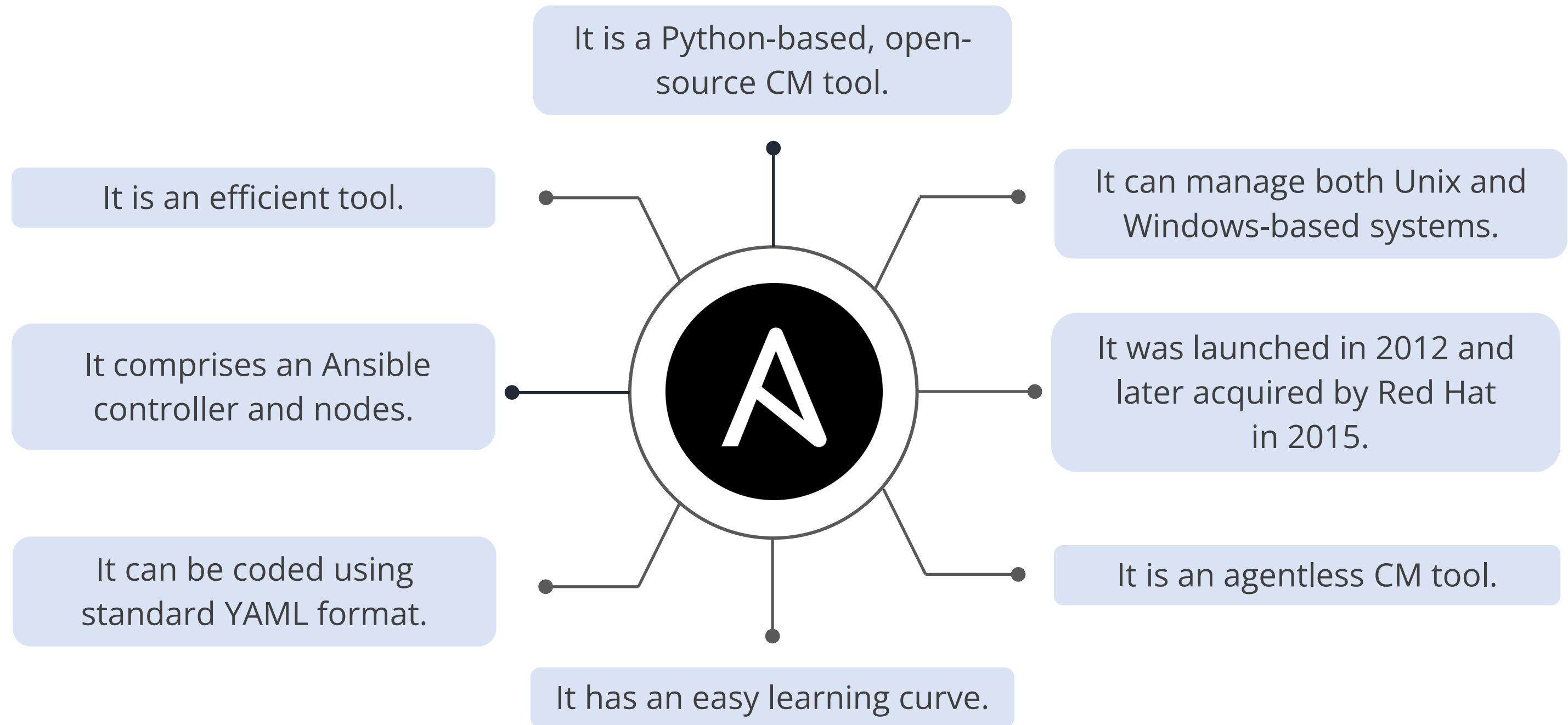
You are tasked with managing your company's infrastructure using Infrastructure as Code (IaC). To ensure that the infrastructure is always in a desired state, which approach should you use, and why?

- A. Imperative approach, because it provides step-by-step instructions
- B. Declarative approach, because it defines the desired state of the infrastructure
- C. Manual configuration, because it allows for more flexibility
- D. Ad-hoc scripting, because it can be quickly modified for immediate needs



Introduction to Ansible

Introduction to Ansible



What Is Ansible?

It is an open-source IT engine that is used to automate application deployment, service orchestration, cloud provisioning, and other IT tools.



ANSIBLE

Why Use Ansible?

Ansible is used for several reasons, including:

It is easy to structure the release process with distributed and continuous development.

It results in lower system maintenance costs.

1

2

3

4

5

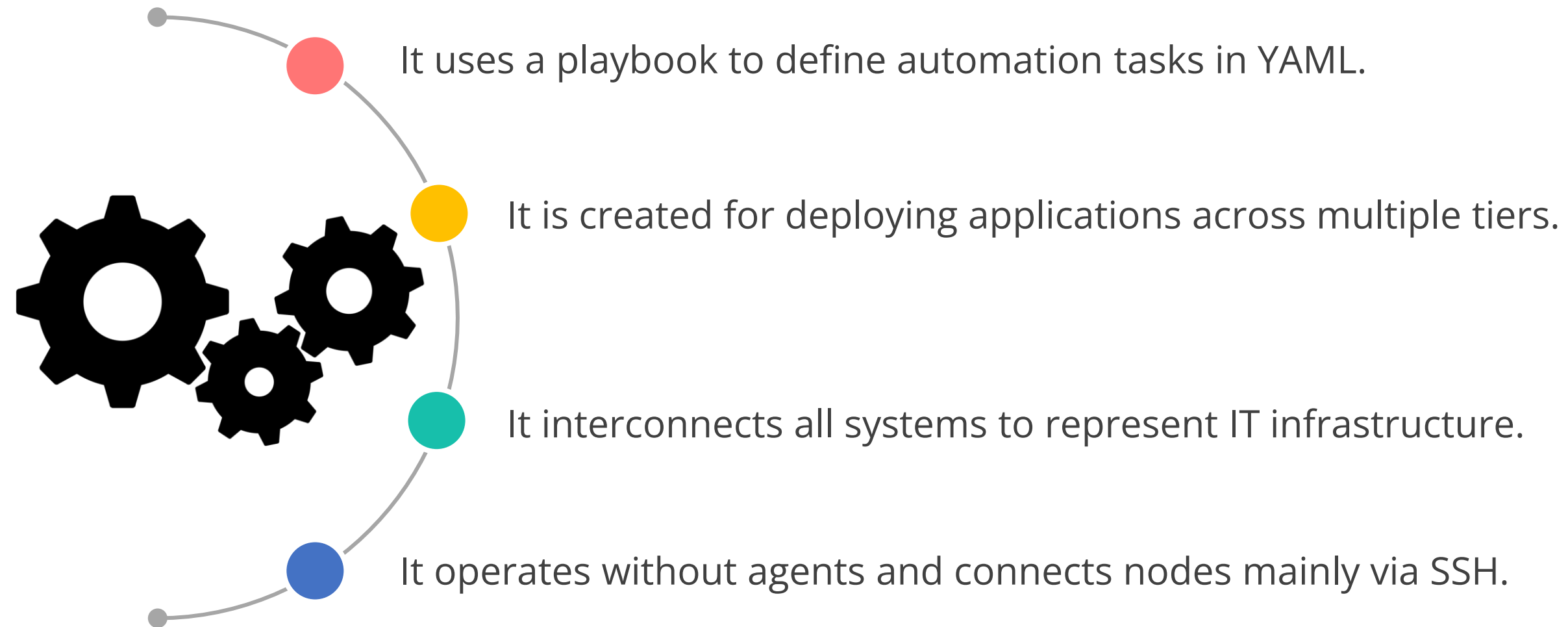
It is easy to audit and verify processes.

It is compatible with all infrastructures.

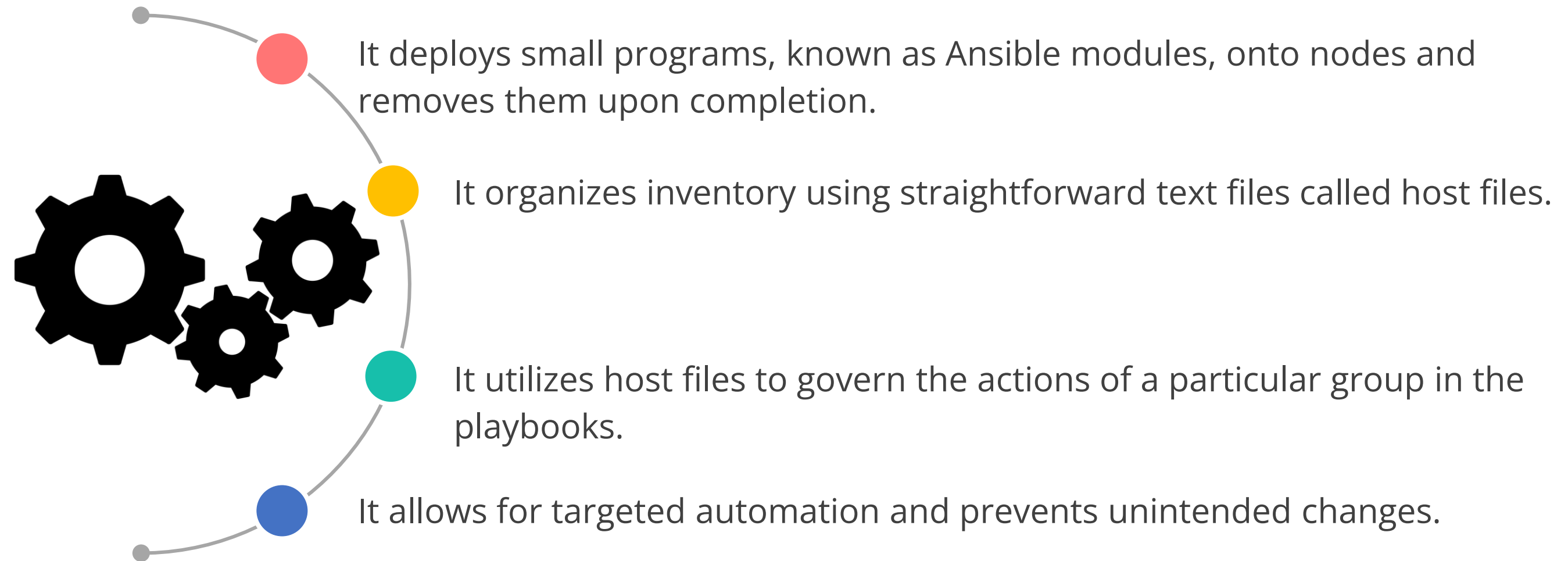
It leads to faster recovery time.

Ansible Features and Capabilities

It is a powerful tool with several distinctive features and capabilities.



Ansible Features and Capabilities



Ansible Components

The following are the key components of Ansible:

Control Machine

It is a machine where Ansible is installed, and it functions as a server.

Inventory

It is a host file that contains information about the managed nodes.

Playbook

It is the entry point of Ansible provisioning, written in YAML.

Task

It is a block of code that defines a single process.

Ansible Components

Module

It is an abstract of a system task, such as creating and changing files.

Role

It is a framework that organizes playbooks, facilitating the sharing and reuse of provisioning.

Facts

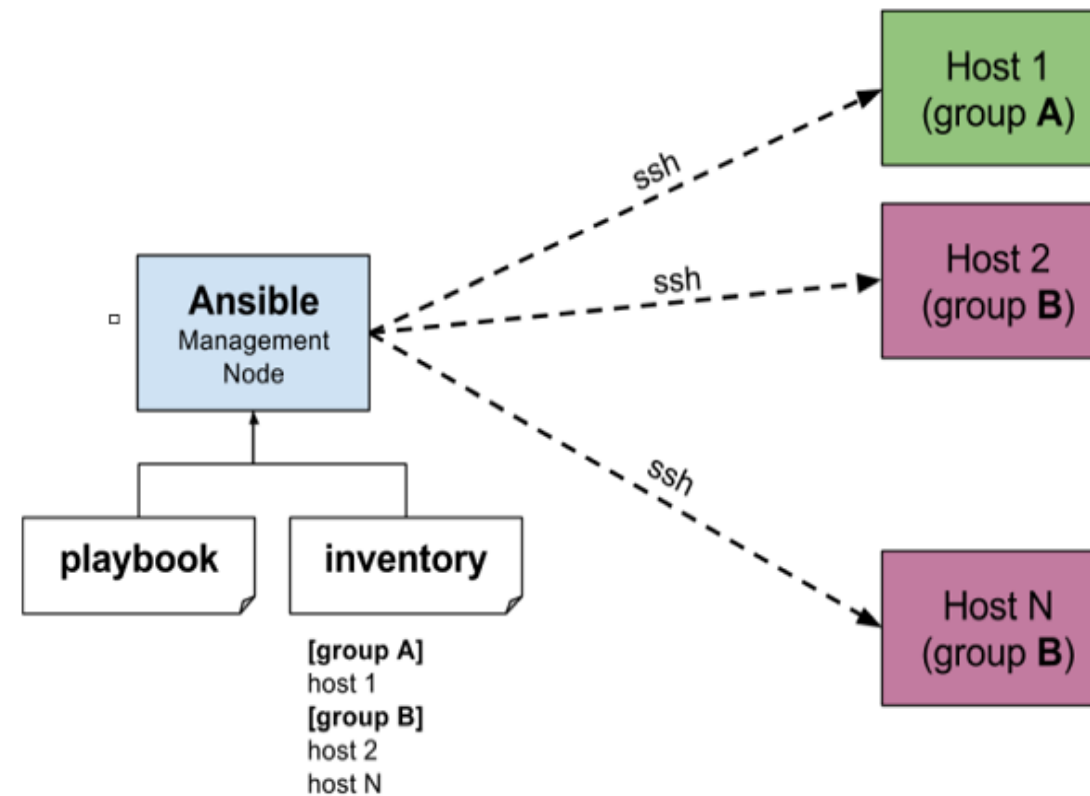
It refers to global variables that contain information about the system.

Handlers

It refers to tasks that trigger changes in the status of a service.

Ansible Working and Architecture

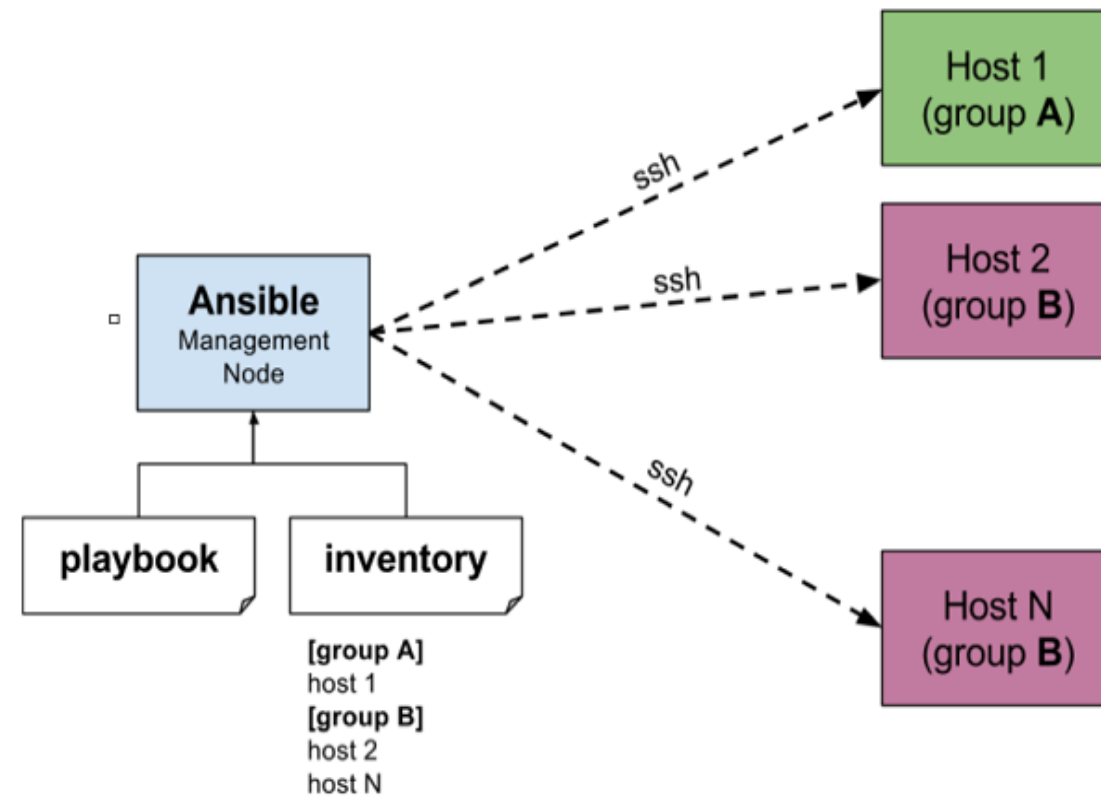
Ansible operates by establishing connections to nodes and deploying small programs, referred to as Ansible modules. By default, it runs these modules over SSH and removes them upon completion.



Modules can be stored on any machine, eliminating the need for servers, daemons, or databases.

Ansible Working and Architecture

The management node acts as the controlling node, overseeing the execution of the playbook. This playbook is YAML code designed to carry out small tasks on client machines.

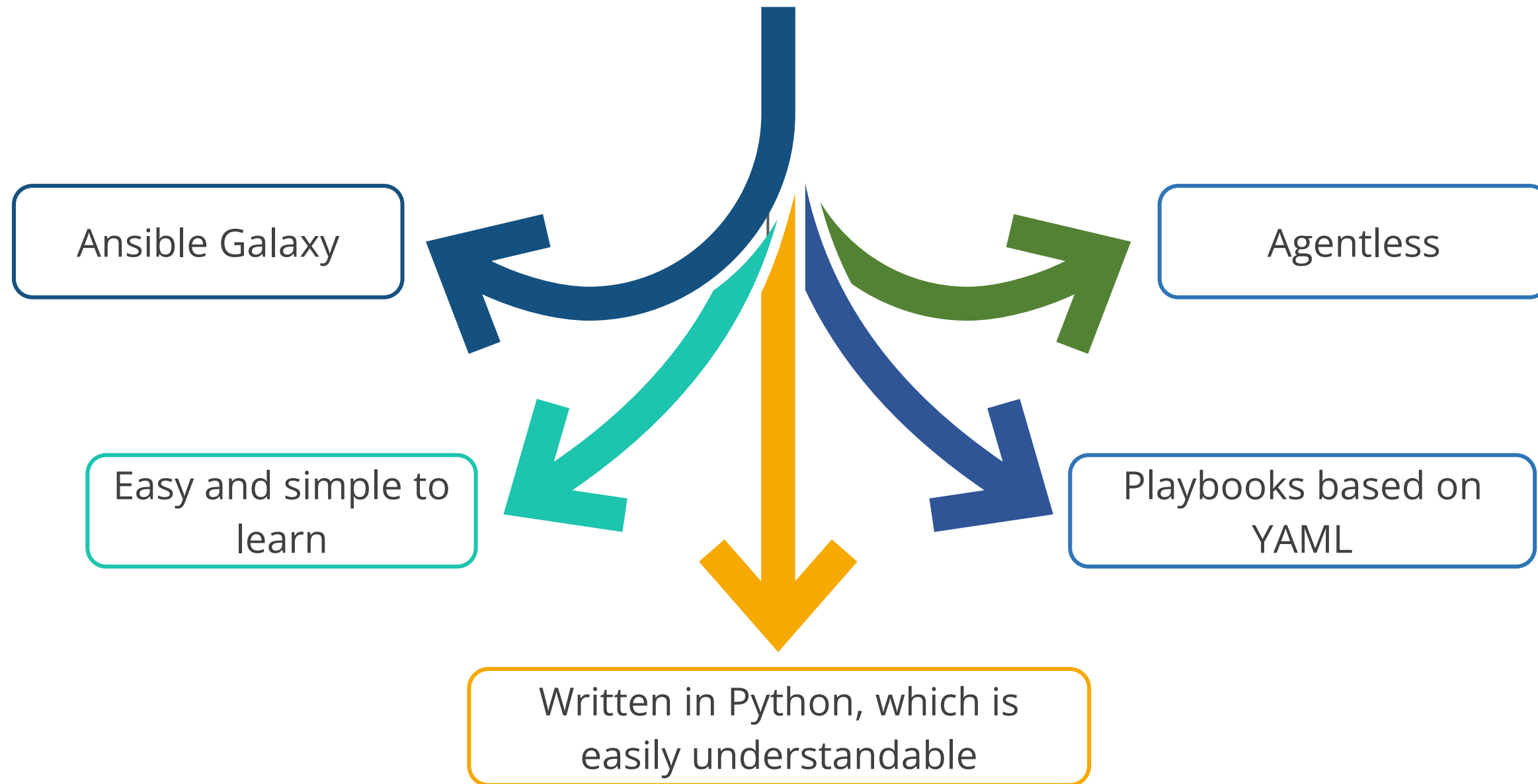


The inventory file is a list of hosts where the Ansible modules will operate.

The management node establishes an SSH connection and executes modules on the hosts.

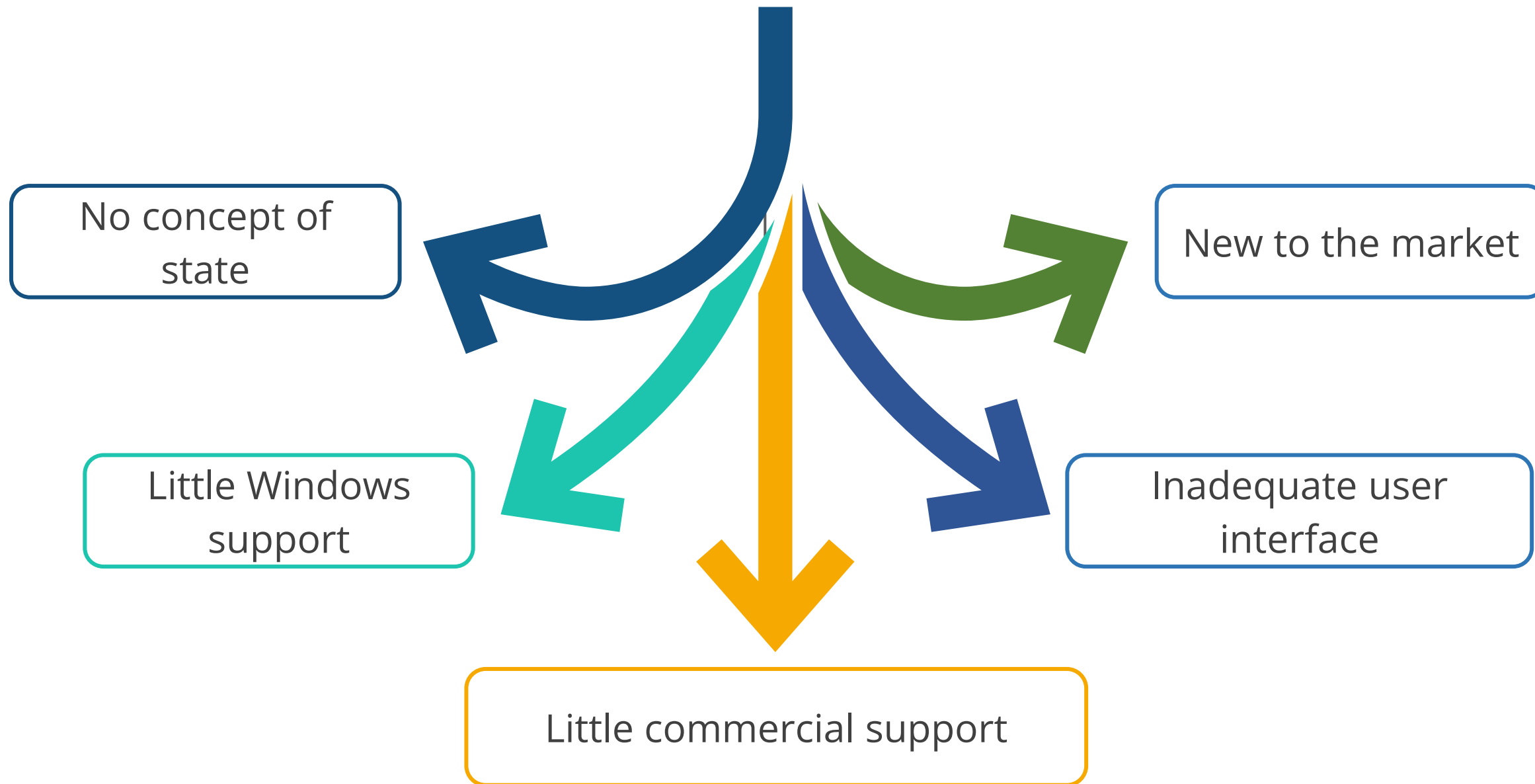
Advantages of Ansible

The following are the advantages of Ansible:



Drawbacks of Ansible

The following are the drawbacks of Ansible:



Assisted Practice



Setting up LAMP stack using Ansible playbook

Duration: 15 Min.

Problem Statement:

You have given a task to set up and configure LAMP Stack dependencies with Ansible to pre-configure the infrastructure for new developers and automate the configuration management for the same.

Outcome:

Users will have a streamlined, efficient, and scalable process for setting up and managing LAMP stack environments, ultimately enhancing developer productivity and maintaining consistency across all development platforms.

Note: Refer to the demo document for detailed steps:
01_Setting_Up_LAMP_Stack_Using_Ansible

Assisted Practice: Guidelines



Steps to be followed:

1. Configure the Ansible playbook
2. Execute the LAMP playbook

Quick Check



As an IT manager, you are evaluating Ansible for its configuration management capabilities. Which of the following best describes how Ansible executes tasks on managed nodes?

- A. By installing a client agent on each managed node
- B. By using a master-slave architecture
- C. By leveraging SSH for agentless communication
- D. By requiring a dedicated control server for each managed node

Key Takeaways

- ❖ Configuration management is a process in systems engineering that ensures a product's performance, functional, and physical attributes consistently align with its requirements, design, and operational information.
- ❖ The objective of configuration management is to maintain computers in their desired state. This was traditionally managed manually or through proprietary programming by system administrators.
- ❖ Ansible is an open-source IT engine that is used to automate application deployment, service orchestration, cloud provisioning, and other IT tools.
- ❖ In a declarative approach, the IaC tool will set up the system for the user, defining the ideal system state, including the services required and any qualities they must have.





Thank you