cisco Live!

#CiscoLive

The bridge to possible

Accelerate Datacenter Infrastructure Provisioning Using Infrastructure as Code

Brandon Beck, Technical Solutions Architect - @techBeck03
Lionel Hercot, Technical Marketing Engineer - @LHercot
BRKDCN-2454

Agenda

- What is Infrastructure as Code
- Why Infrastructure as Code
- Tools of the trade
- Demo
- Cisco DC Providers and Collections
- Where to start your journey

“Automation is to modern infrastructure what blood is to the body. It is core, you cannot have modern infrastructure without it.”

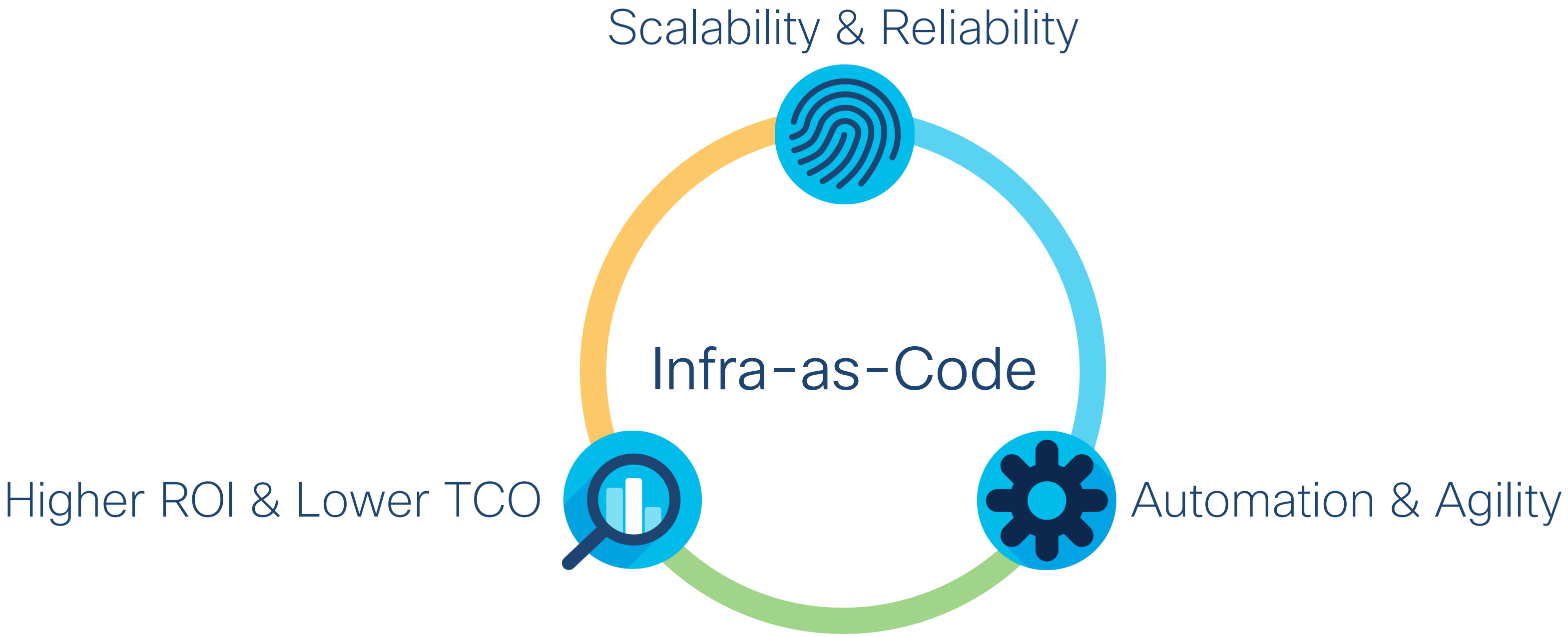
Market Guides for Infrastructure Automation and Service Orchestration and Automation Platforms by Manjunath Bhat

Gartner

Infrastructure as Code (IaC) – What/Why/How

- Automate the provisioning and management of the technology stack
- Translate manual tasks into reusable, robust, distributable code
- Rely on practices that have been successfully used for years in software development (version control, automated testing, release tagging, continuous delivery, etc.)
- Benefits: much higher delivery speed; significant reliability boost

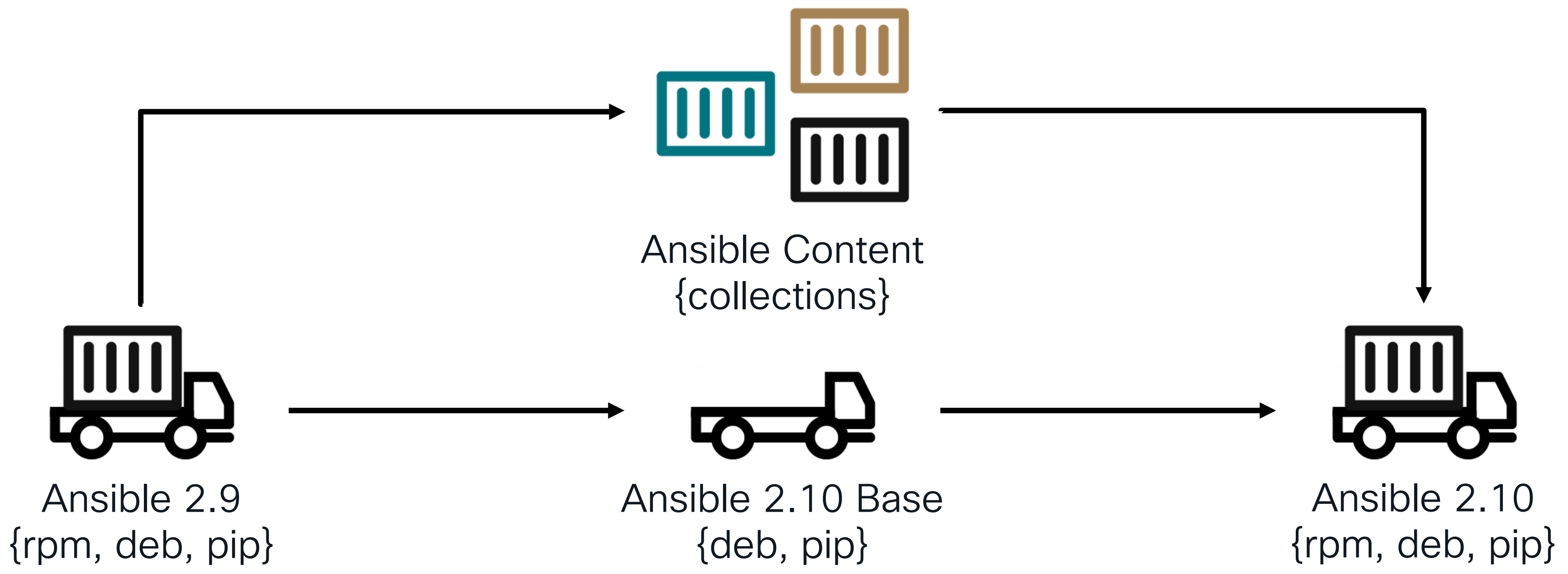
Why Infrastructure-as-Code (IaC)?



What is Ansible?

- Open-source Configuration Management Tool
- Commercial support from RedHat
- Declarative (when possible) and idempotent
- Can manage a wide range of systems:
  - VMs, network devices, cloud instances, etc.
- Agentless
- Python server-side dependencies

What are Ansible Collections?



What is Terraform?

HashiCorp
Terraform

- Open-source Infrastructure Provisioning Tool
- Commercial support from HashiCorp
- Declarative and idempotent
- Immutable infrastructure concept
- Can manage a wide range of systems:
  VMs, network devices, cloud instances, etc.
- Agentless, single binary file
- Zero server-side dependencies

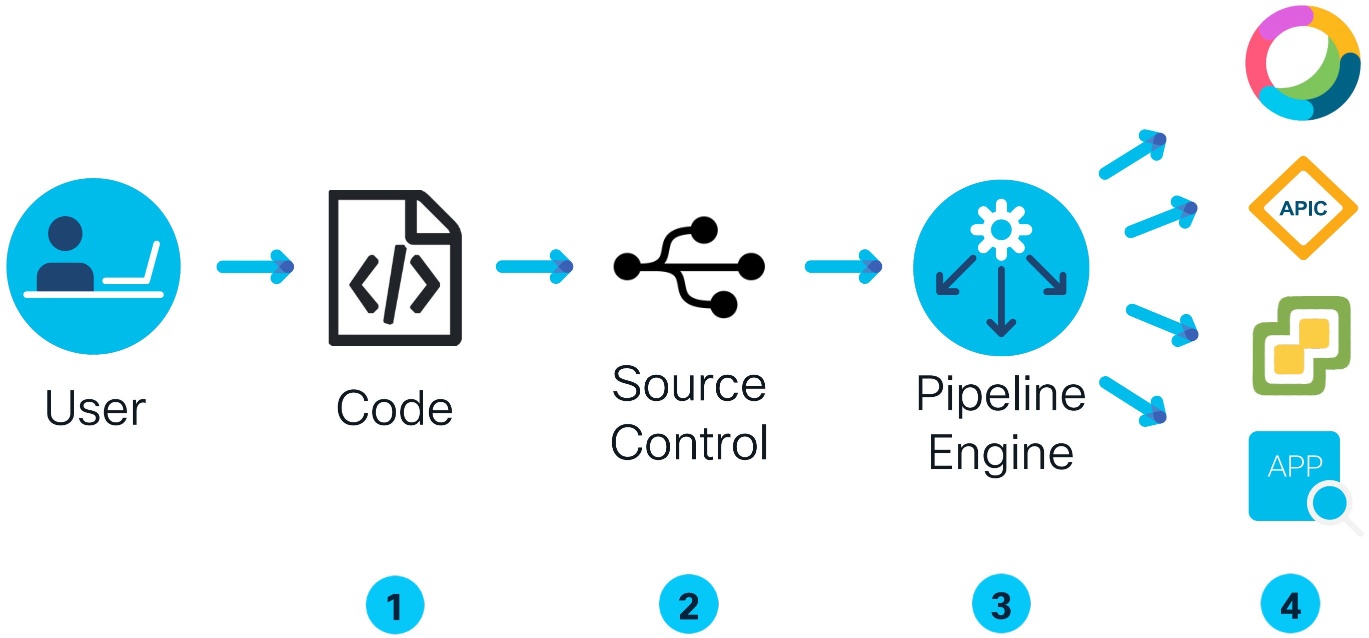
Ansible or Terraform?

- Both Ansible and Terraform can coexist
  - It’s not an either/or story
  - Terraform can call Ansible for ad-hoc tasks after deploying a VM
- Terraform keeps state locally
  - It knows what is configured vs desired end-state
  - Can automatically destroy / recreate resources
- Ansible mutate the infrastructure
  - Need to re-run everything
  - Might need to create advanced controls to avoid long running scripts

CI/CD Pipeline

- Continuous Integration (CI)
  - Practice of merging all developer changes to a shared repo several times a day
  - It main include the creation and test of artifacts (executable, app, ...)
- Continuous Deployment (CD)
  - Approach to deliver new software functionalities frequently through automated deployments
  - Rely on Continuous Integration for tracking changes

What a CI/CD workflow looks like



Which tool / language / orchestrator should I use?

- Normalization of the construct definition is the goal
  - But probably not possible today
- CI/CD Pipeline allow to use the best tool for each case
  - That means multiple tools in the same pipeline
  - Gartner estimates that on average 8 different tools are used in a CI/CD pipeline

Common components of a CI/CD Pipeline

Code	Source Control	Pipeline / Orchestrator
HashiCorp Terraform	GitLab	HashiCorp Terraform Cloud
ANSIBLE	GitHub	GitHub Actions
	Bitbucket	Travis CI

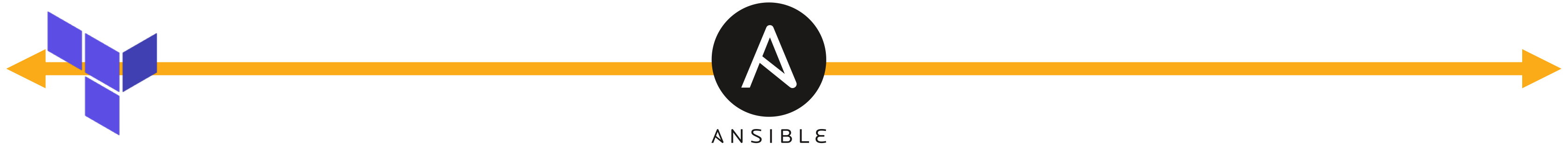
Declarative vs Imperative

- Define *what* the eventual target configuration should be

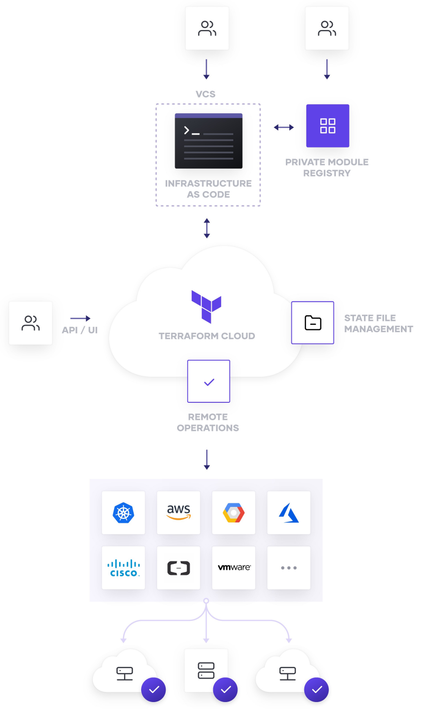
  e.g., 1 Tenant with 2 BDs and 2 EPGs
- Define the desired state
- Automation is responsible for the desired state to be reflected in the infrastructure

- Define *how* the infrastructure should be changed

  e.g., Add BD X and EPG Y
- Automate a common use case (e.g. add a network segment)
- Automation defines steps (workflow) to end with the desired conclusion



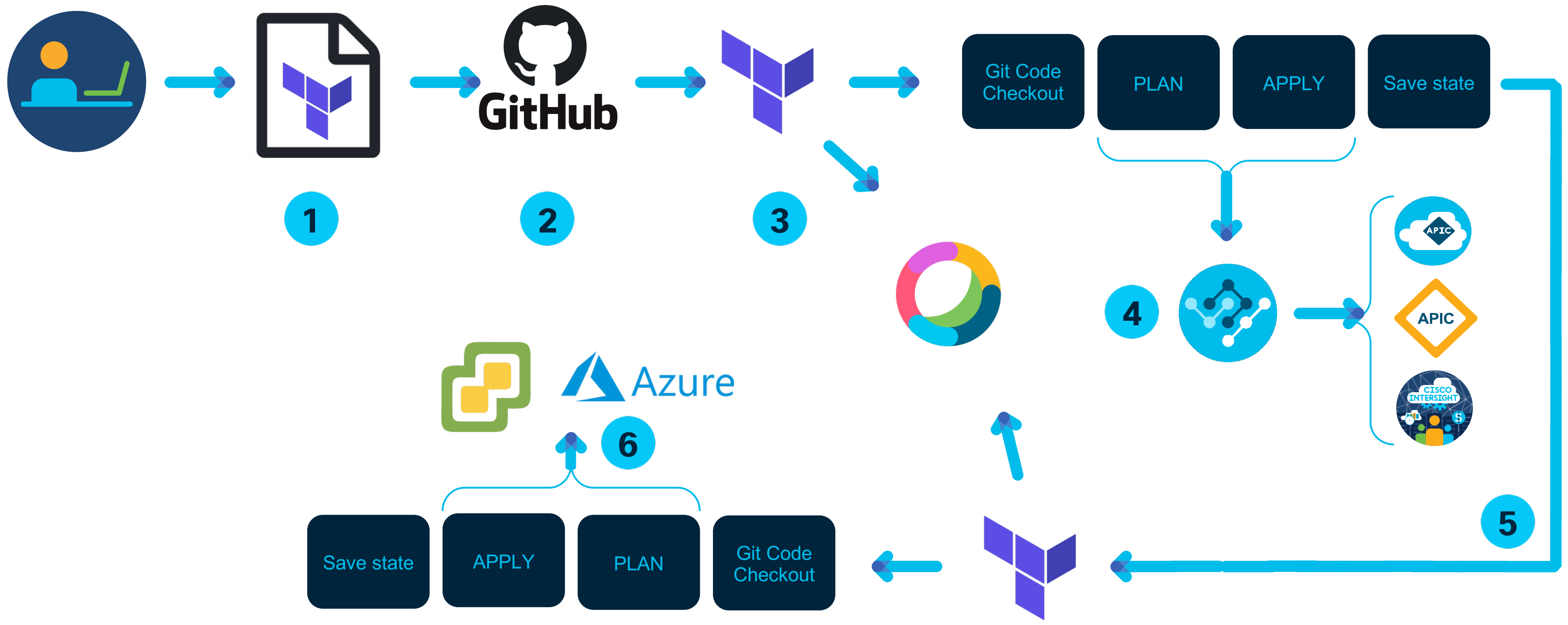
What is Terraform Cloud?



- HashiCorp Infrastructure as Code Cloud Service
- Can trigger plan, apply and destroy any Terraform plan.
- Can be triggered by a source control hook
  - A commit / PR to a repository can be used as a hook
- Provides Enterprise level features:
  - State File Management and Sharing
  - Private Module Registry
  - Config Compliance Checks (Sentinel)

Let’s see it in action!

What a Terraform Cloud workflow looks like



Cisco Data Center Collections

ANSIBLE

Cisco NXOS	80+ modules in cisco.nxos
ACI Cloud / Onprem	80+ modules in cisco.aci
Multi Site Orchestrator	45+ modules in cisco.mso
Network Assurance Engine	6+ modules in cisco.nae
Data Center Network Manager	5+ modules in cisco.dcnm
Intersight	8+ modules in cisco.intersight
UCS Manager	29+ modules in cisco.ucs

Available Today

Cisco Data Center Providers

HashiCorp Terraform

ACI
Cloud / Onprem

Multi Site Orchestrator

Data Center Network Manager

Intersight

APIC

DCNM

CISCO INTERSIGHT

Available Today

How to start?

- Start simple
- Pick a task you want to automate
  - Interface Configuration (Fabric Access Policies / Interfaces)
  - Cookie-cutter Tenant / VRF / EPG templating
  - Cookie-cutter VNI / VRF / Interface templating
  - EPG to VLAN assignment
  - Cookie-cutter server policies/profiles
  - Cookie-cutter Kubernetes cluster deployments
- Automate these tasks (individually)
- Build on it (stitch them together)
- Verify your changes with NAE

New DevNet Resources available!

New MSO Sandbox

New Terraform Learning Labs

[More Info on DevNet: https://developer.cisco.com/nexusapi/](https://developer.cisco.com/nexusapi/)

Key Takeaways

- Infrastructure as Code is a journey. Start it today!
- Cisco products are designed to be automated
- Ansible and Terraform can work together
- Go learn with our DEVNET learning labs

Continue your education

- Demos in the Cisco campus
- Meet the engineer 1:1 meetings
- Walk-in labs
- Related sessions

The bridge to possible

Thank you

cisco Live!

#CiscoLive