HashiCorp Brings a Cloud Operating Model to Network Engineers

Kevin Corbin

HashiCorp

Twitter: @kecorbin

Season 3, Talk 4
Hosted by Hank Preston, NetDevOps Engineer
Twitter: @hfpreston
https://developer.cisco.com/netdevops/live

Cloud Operating Model for Network Engineers

HashiCorp

- Automate the things!
- Expanding domain of "networking" in multi-cloud world
- NetDevOps Engineers are application developers!

About HashiCorp

Leading Cloud Infrastructure Automation

Our software stack enables the provisioning, securing, connecting and running of apps and the infrastructure to support them.

We unlock the cloud operating model for every business and enable their digital transformation strategies to succeed.

Founded	Employees	Funding
2012	**900**	**349M**

Nomad

Consul

Terraform

Vault

Packer

Vagrant

A generational transition is underway

Traditional datacenter
"Static"

Dedicated infrastructure

Modern datacenter
"Dynamic"

Private cloud + **Public multi-cloud**

SYSTEMS OF RECORD ⟷ **SYSTEMS OF ENGAGEMENT**

Evolving Application Workload Delivery

APP SERVER		CLOUD NATIVE	SERVERLESS	EDGE

Challenge

How to deliver applications to the cloud with consistency?

PHYSICAL	VMs	CONTAINERS	SERVICES	DEVICES

Solution

Proprietary solutions reduce flexibility and creates complexity across vendors

Run Development

Connect Networking

Secure Security
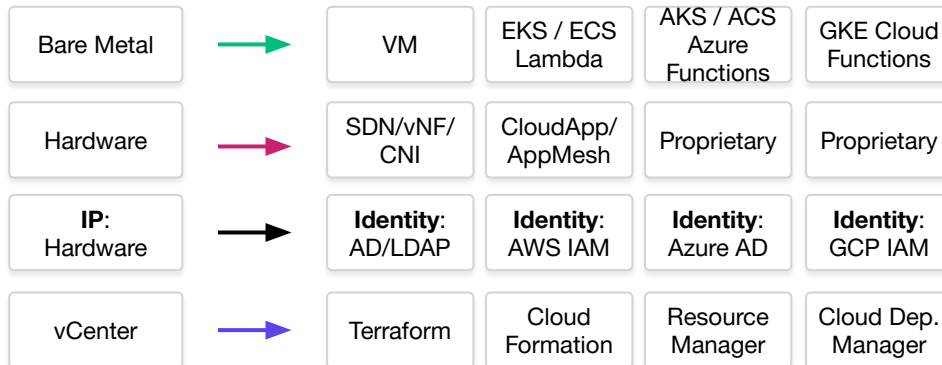
Provision Operations

	DEDICATED		MULTI CLOUD			
Run	Bare Metal	→	VM	EKS / ECS Lambda	AKS / ACS Azure Functions	GKE Cloud Functions
Connect	Hardware	→	SDN/vNF/ CNI	CloudApp/ AppMesh	Proprietary	Proprietary
Secure	**IP**: Hardware	→	**Identity**: AD/LDAP	**Identity**: AWS IAM	**Identity**: Azure AD	**Identity**: GCP IAM
Provision	vCenter	→	Terraform	Cloud Formation	Resource Manager	Cloud Dep. Manager
				AWS	Azure	GCP

# Evolving application workload delivery

Challenge

How to deliver applications to the cloud with consistency?

Solution

Establish central shared service platforms with a single control plane, and consistent workflows.

APP SERVER		CLOUD NATIVE	SERVERLESS	EDGE
PHYSICAL	VMs	CONTAINERS	SERVICES	DEVICES

Nomad   Orchestration

Consul   Networking

Vault   Security

Terraform   Infrastructure
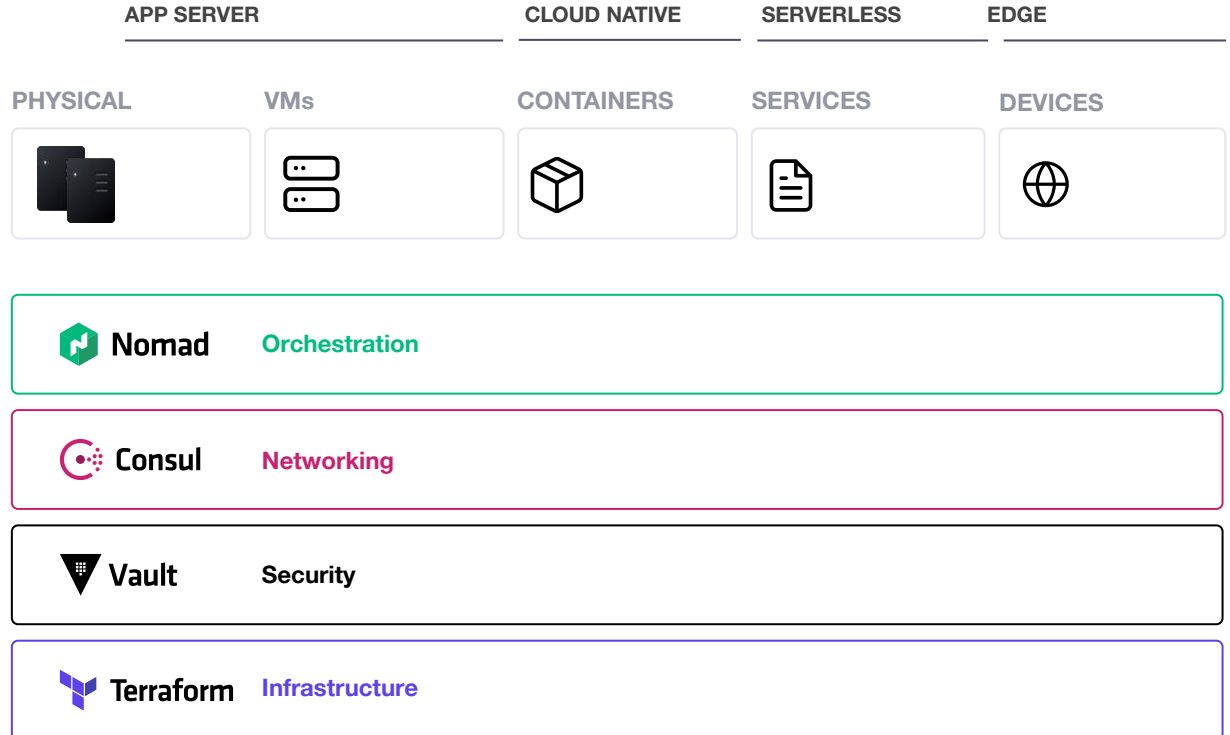
The HashiCorp Stack

A control plane for every layer of the cloud operating model

Run
Development

Connect
Networking

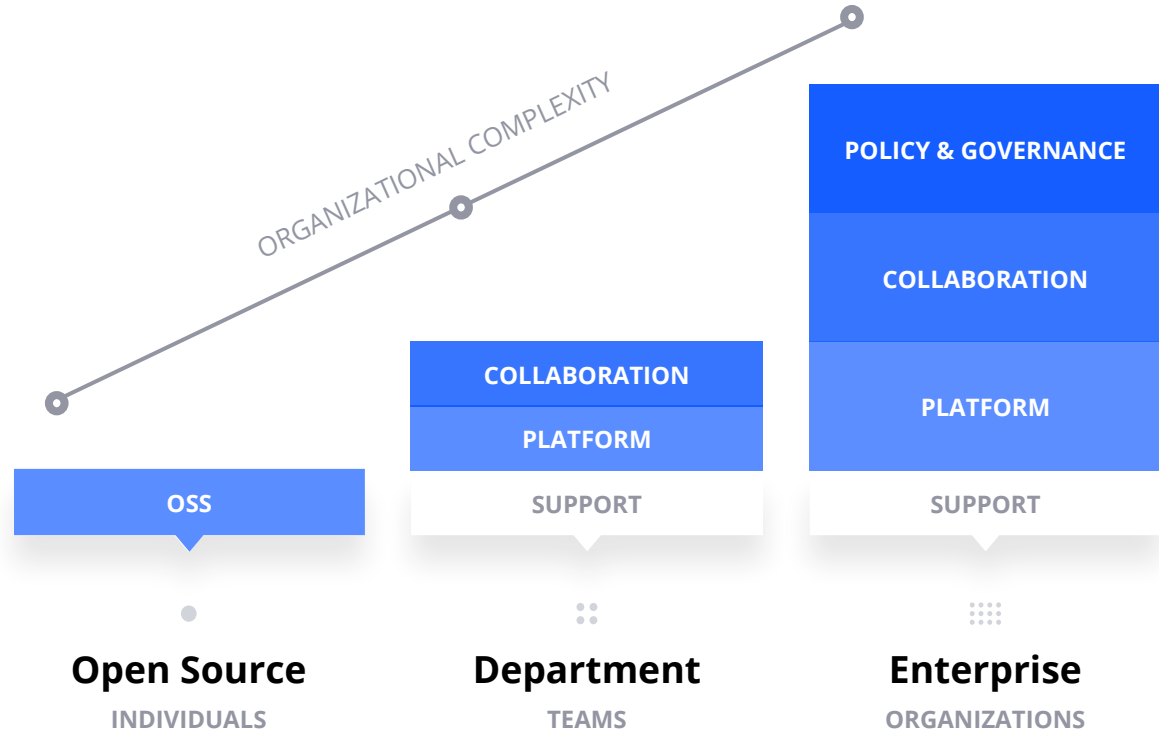**Secure**
Security

Provision
Operations
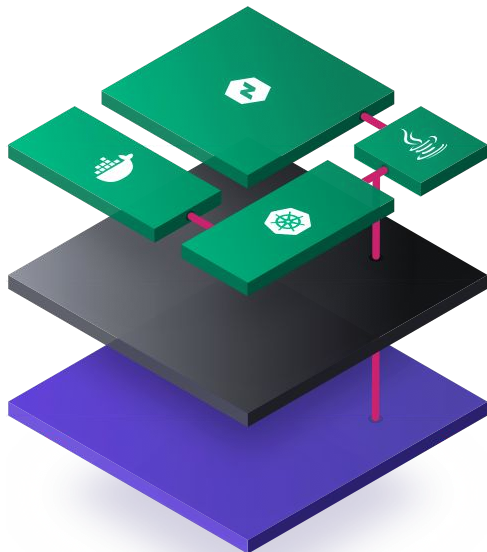
Nomad

Consul

Vault

Terraform

Open Source vs. Enterprise

Products build on open source to address organizational challenges and complexity

ORGANIZATIONAL COMPLEXITY

POLICY & GOVERNANCE

COLLABORATION

PLATFORM

SUPPORT

COLLABORATION

PLATFORM

SUPPORT

OSS

Open Source
INDIVIDUALS

Department
TEAMS

Enterprise
ORGANIZATIONS

learn.hashicorp.com

HashiCorp Learn

Learn how to provision, secure, connect, and run any infrastructure for any application.

Terraform	**Vault**	**Consul**
Learn about automated infrastructure provisioning.	Learn about secrets management and data protection.	Learn how to run service discovery and a service mesh with Consul.
Nomad	**Vagrant**	**Packer**
Learn how to deploy and manage any containerized, legacy, or batch application.	Learn to create development environments with Vagrant	Learn to build automated machine images with Packer

Terraform: Infrastructure Automation

Provides the foundation for cloud infrastructure automation using infrastructure as code for provisioning and compliance in the cloud operating model

- **Multi-Cloud Compliance & Management** to provision and manage any infrastructure with one workflow

- **Self-Service infrastructure** for users to easily provision infrastructure on-demand with a library of approved infrastructure modules
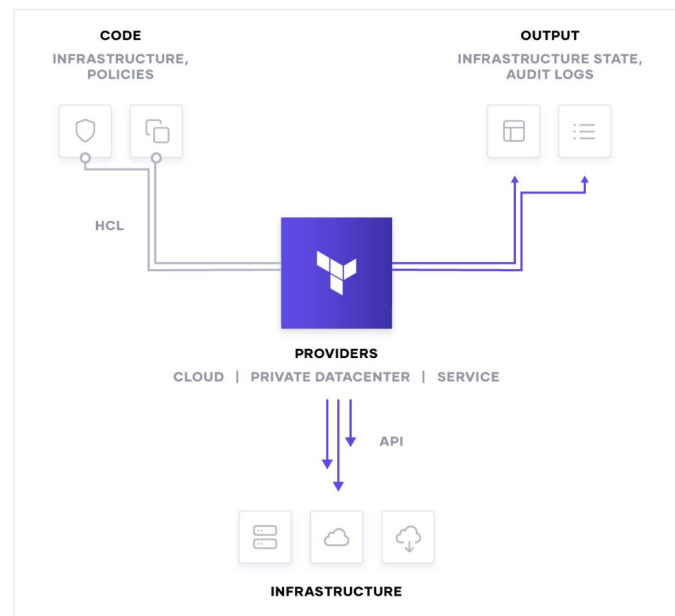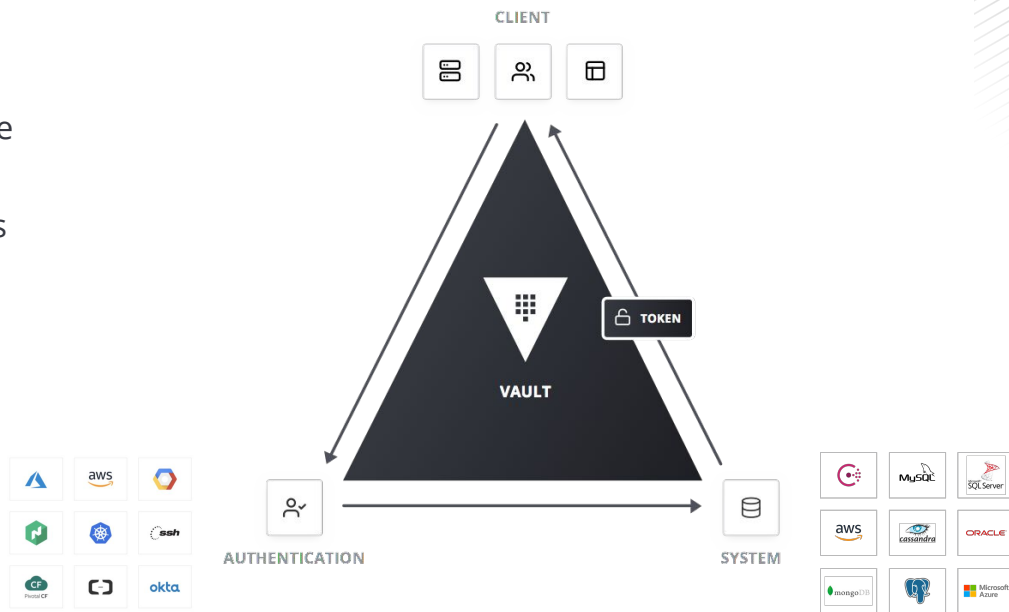
Terraform Networking Use Cases

- Stateful, Declarative network provisioning
- Providers for many common network vendors
  - Cisco, F5, Palo, Infoblox, Checkpoint
- vNF Provisioning
- DNS records as Code
  - Infoblox, Route53
- Content Delivery Networks
- Integrates well with Ansible

Vault: Security Automation

Provides the foundation for cloud security that uses trusted sources of identity to keep secrets and application data secure in the cloud operating model

- **Secrets management** to centrally store and protect secrets across clouds and applications
- **Data encryption** to keep application data secure across environments and workloads
- **Advanced Data Protection** to secure workloads and data across traditional systems, clouds, and infrastructure.

Vault Networking Use Cases

- Remove hard-coded static values from scripts, playbooks, configs
  - Python SDK, Ansible modules
- PKI certificates for infrastructure equipment
- SSH Key Management / Rotation
- local and backup logins / privilege passwords
- Vault agent for templating / injecting secrets into configs
- KMIP for storage arrays

Consul: Network Automation

Provides the foundation for cloud network automation as a central service registry for service-based networking in the cloud operating model

- **Service registry & health monitoring** to provide a real-time directory of all services with their health status
- **Network automation** with service discovery for dynamic reconfiguration of network infrastructure as application services scale up, down or move
- **Zero trust network with service mesh** to enable identity-based security enforced at the endpoints via sidecar proxies
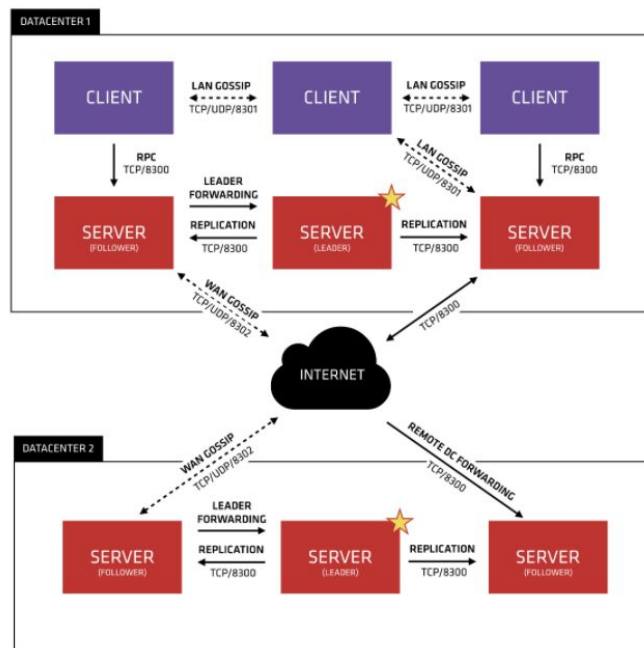
Consul Networking Use Cases

- Service Registry - real-time source of truth for location of nodes/services
- Vivaldi - Decentralized Network Coordinate System
  - Find Nearest X (Service/Node/Datacenter)
- Distributed Health Checking (w/ consul-esm for services which don't have agent)
- Dynamic updates of firewall rules / load balancer pools
- Execute scripts/playbooks based on service changes - watches
- consul-template - arbitrary config/api template rendering, notifier, and supervisor
- Service Mesh - "secure sockets as a service"

Consul + ACI Appcenter app coming soon!
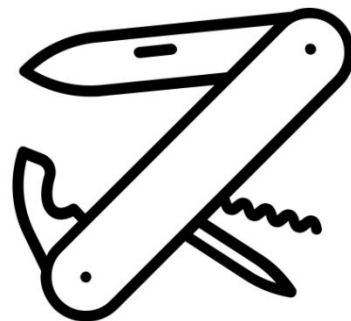
Nomad: Application Automation

Provides the foundation for cloud application automation by enabling workload orchestration in the cloud operating model

- **Container Orchestration** for deploying, managing and scaling containerized applications

- **Legacy Application Orchestration** to containerize, deploy and manage legacy apps on existing infrastructure

- **Batch Workload Orchestration** to enable ML, AI, data science and other intensive workloads in high performance computing (HPC) scenarios
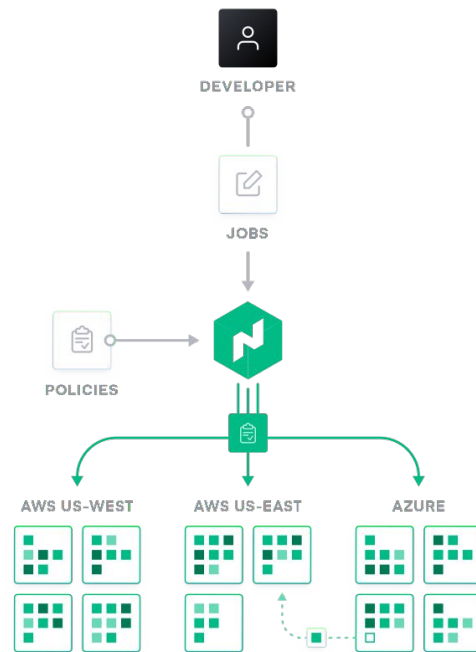
Nomad Networking Use Cases

- Dead simple task scheduler for automation scripts
- Modern replacement for the "tools" server
- Consul integration - service registry data is easily available for inventories, etc.
- Vault integration - secrets are available

Resources

- Learning Tracks
  - https://learn.hashicorp.com/
- Instruqt - Self paced, interactive lab environments
  - https://play.instruqt.com/hashicorp
- Documentation
  - https://www.terraform.io/docs/index.html
  - https://www.vaultproject.io/docs
  - https://www.consul.io/docs/index.html
  - https://www.nomadproject.io/docs/
- Additional Slides in this deck for use cases discussed

Use Case Details

Guiding Principle:
Infrastructure as Code

- Using version control and automation to reduce human error and failed builds
- Terraform infrastructure as code and policy as code to automate everything.
- Open source providers allow rapid creation and support for any infrastructure

Azure DevOps
Bitbucket
GitHub
GitLab

org/network-web:master

Link workspace to source control repository with Terraform code.

network-web-dev

org/network-web:master
tag: v0.4

Promote v.04 to UAT

network-web-uat

org/network-web:master
tag: v0.3

Promote v.03 to Prod

Fork network-web-uat to network-web-uat-fork1, config is selected from triage-bug branch.

network-web-uat-fork1

org/network-web:triage-bug

network-web-prod

org/network-web:master
tag: v0.2

Fork network-web-dev to network-web-fork1, config is selected from feature branch.

network-web-fork1

org/network-web:feature22

Fork network-web-dev to network-web-fork2 workspace, config is selected from feature branch.

network-web-fork2

org/network-web:feature25

Guiding Principle:
Immutability

- Leads to more robust systems that are simpler to operate, debug, version and visualize
- Improve confidence levels of deployments and simplified analysis of failure scenarios
- Eliminates sequential order of operations required to achieve target state
- Cattle vs Pets

im·mu·ta·ble

/iˈmyo͞otəbəl/ 🔊

Adjective
Unchanging over time or unable to be changed: "an immutable fact".

Synonyms
invariable - unalterable - constant - changeless

copy known-good-config startup-config
reload

Guiding Principle:
Infrastructure State

- Provisioning infrastructure as code is fast and efficient, but rapid changes can be difficult to track
- Implicit dependency resolution
- Independent resources can be provisioned in parallel
- Acts as Cache to optimize performance.
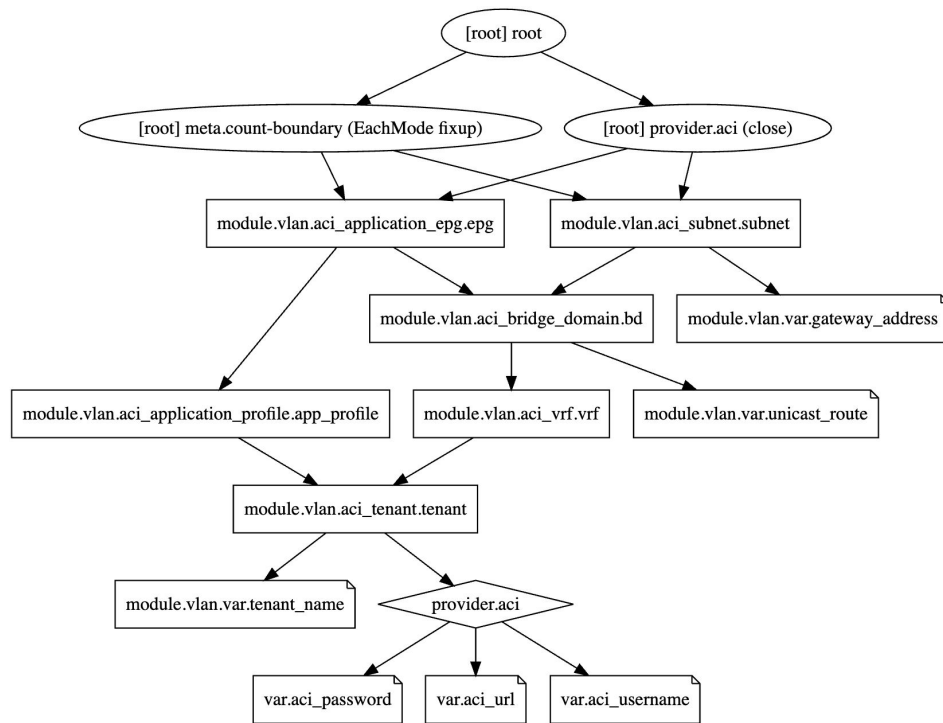
Terraform Core Engine

- OSS hosted at github.com/hashicorp/terraform
- CLI based workflow
- Loads providers as needed

Responsible for:

- Reading and Interpolating configuration files and modules
- State Management
- Executing plan
- Communicating with providers
- Constructing resource graph
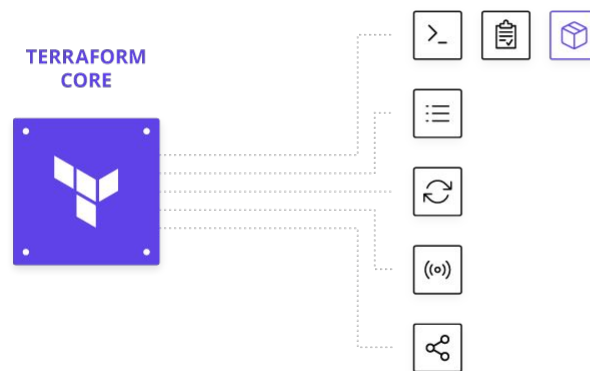
Resource Graph

- Safely provision and change infrastructure
- See planned infrastructure changes before execution
- No need to manually coordinate dependent resources
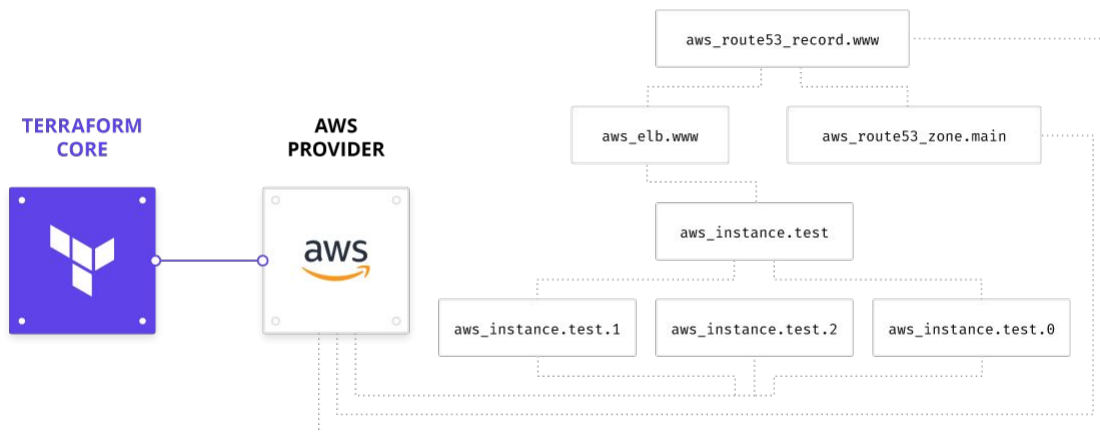
Provider Plugins

- Provider and Provisioner plugins expose implementation for specific services
- Offer extensible layer for 'Core' to learn how to talk to anything with an API without any upgrades

Responsible for:

- Initializing libraries for API calls
- Authenticating with Provider
- Defining resources that map to services
- Executing commands or scripts on designated resources

EXTENSIBLE PROVIDER MODEL

TERRAFORM CORE
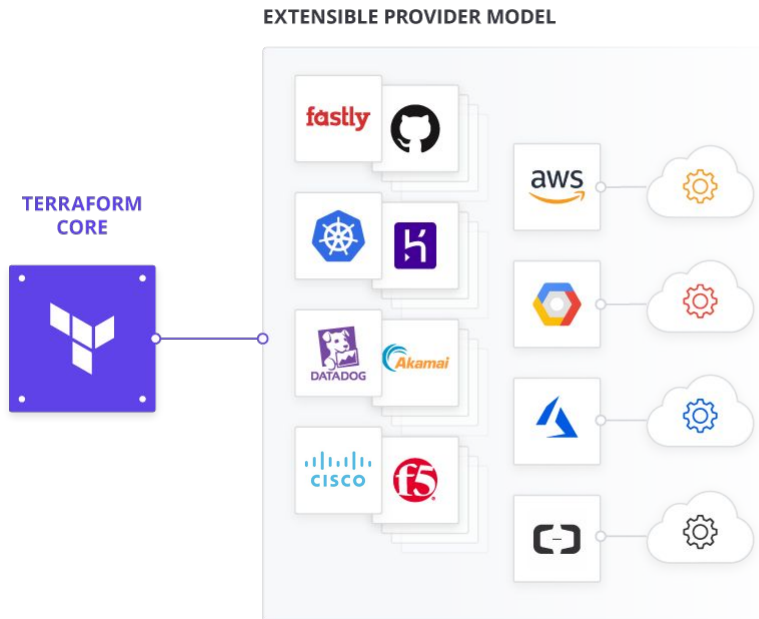
Guiding Principle:
Service Registry

- Allow networking operations to decouple from IP addresses
- Automated networking and security based on logical services
- Services can be discovered, connected, and secured with service name as identity
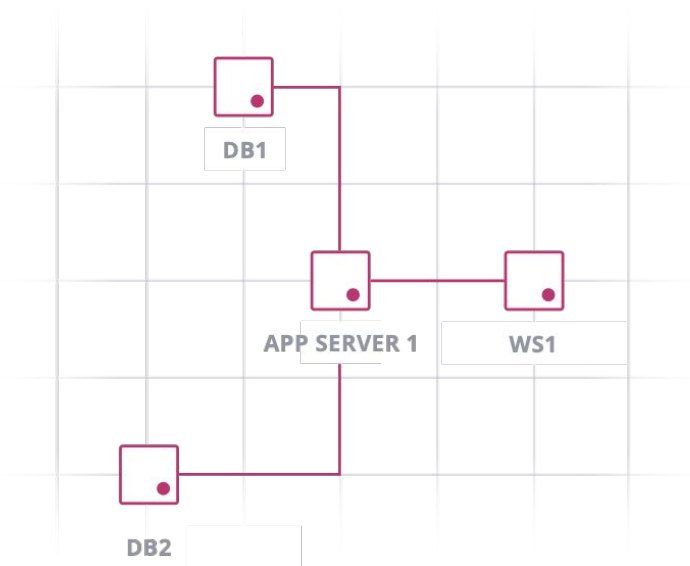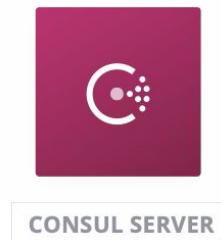
CONSUL SERVER

DB1

APP SERVER 1   WS1

DB2

Consul: Network Automation

Provides the foundation for cloud network automation as a central service registry for service-based networking in the cloud operating model

- ☑ **Service registry & health monitoring** to provide a real-time directory of all services with their health status
- ☑ **Network middleware automation** with service discovery for dynamic reconfiguration as services scale up, down or move
- ☑ **Zero trust network with service mesh** to enable identity-based security enforced at the endpoints via sidecar proxies
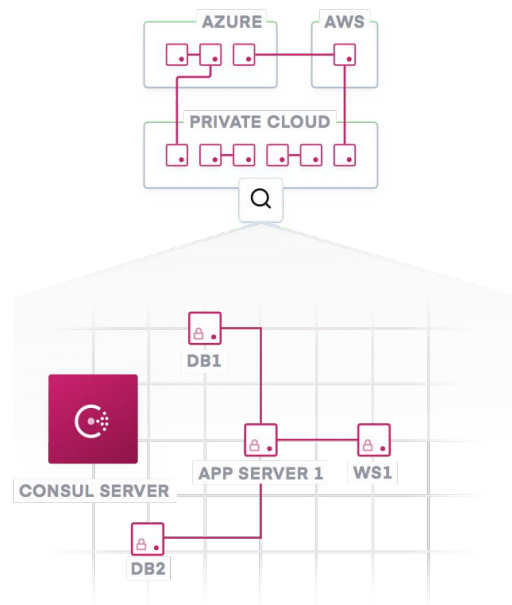
Challenges with tracking dynamic services

BEFORE

- Spreadsheets
- Load balancer dashboards
- Discrete configuration files
- Platform-specific tools
- Cloud-specific tools
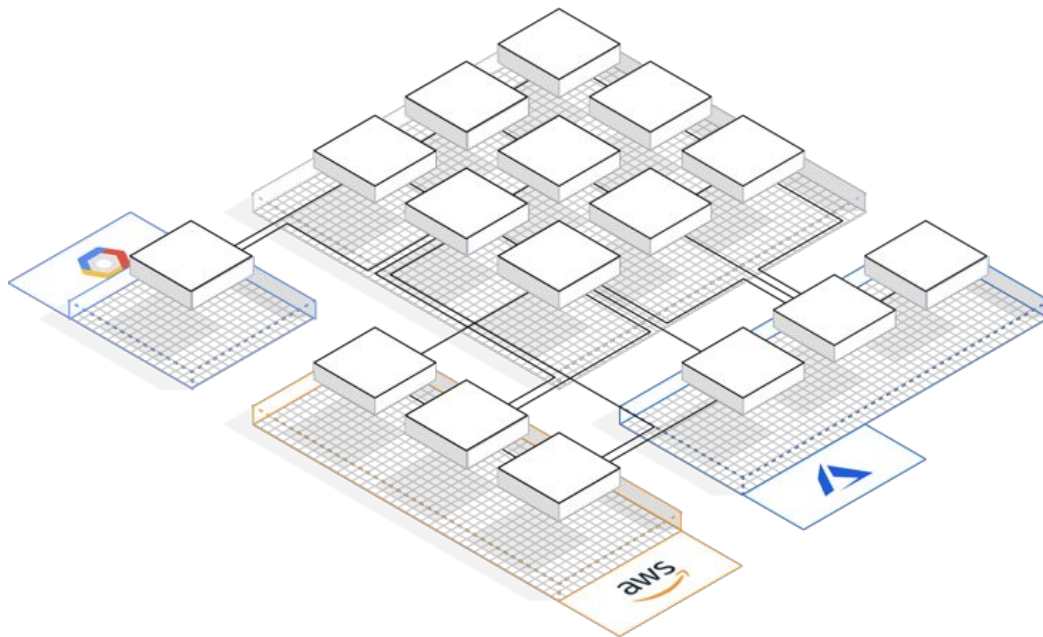
Central Service Registry

AFTER

Consul provides a real-time "directory", including

▪ What services are running
▪ service network location
▪ service health status
▪ Multi-platform & multi-region

dc1   **Services**   Nodes   Key/Value   ACL   Intentions

Services

| All (29) | ✓ Passing (28) | ⚠ Warning (0) | ✗ Critical (1) |

Service	Health Checks ⓘ	Tags	
consul	✓ 3		
postgresql	✓ 2	global	db
postgresql-proxy	✓ 2		
redis	✓ 2	global	cache
redis-proxy	✓ 2		
uuid	✓ 8	global	web
web	✓ 1 ✗ 1	global	web
web-proxy	✓ 8		

Challenges with updates to network devices

BEFORE

- Ticket-based system
- Manual approach
- Error-prone process
- Multiple handoffs between teams
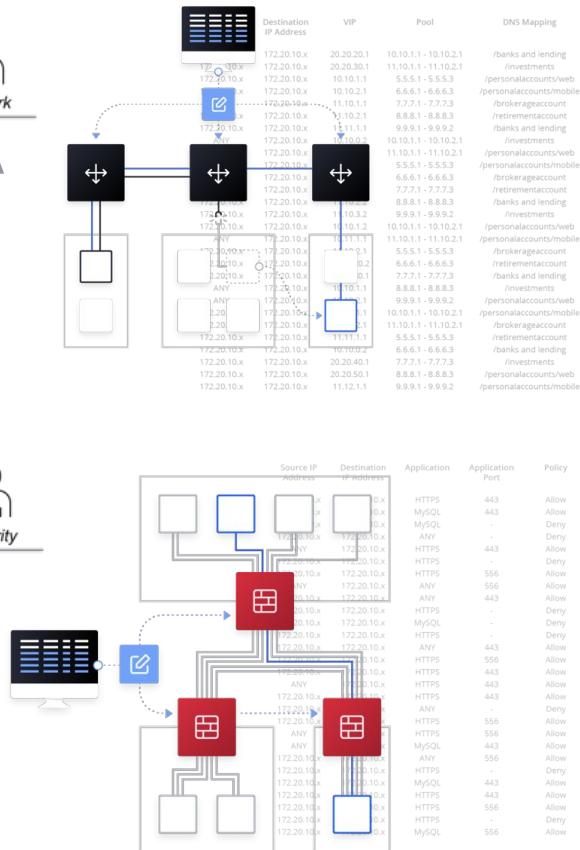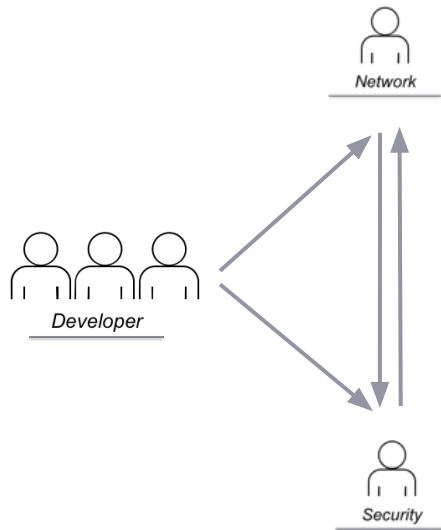
Network Automation

AFTER

Consul enables a "publisher-subscriber" model
- Services "publish" network location automatically
- Network devices "subscribe" to the service changes
- Service changes trigger dynamic reconfiguration automatically
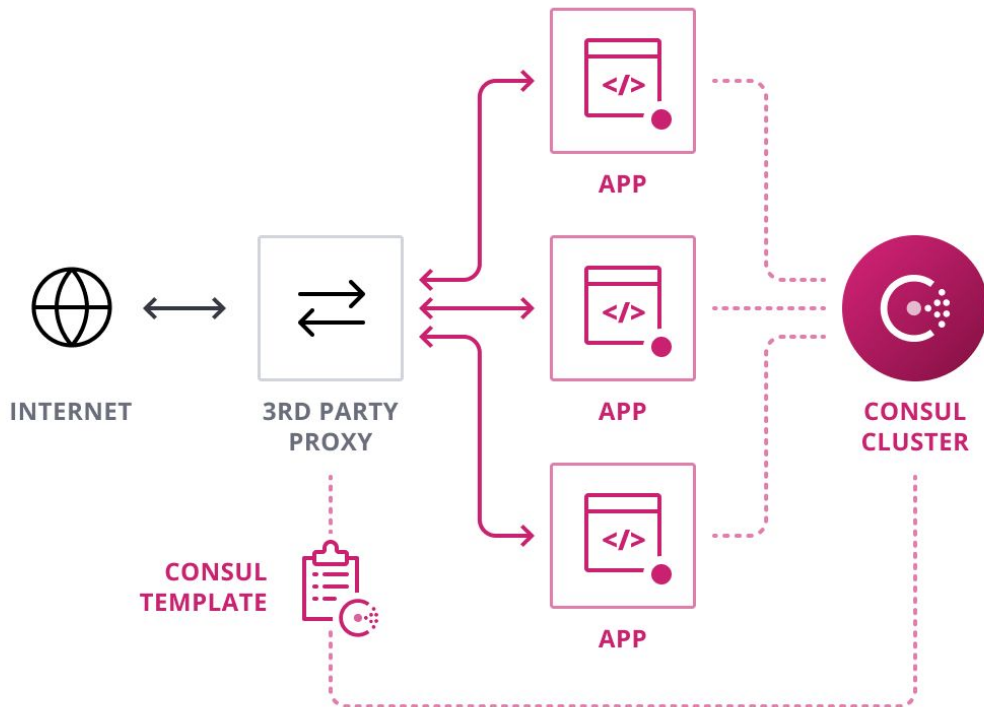
Feature: **Watches**

React to changes dynamically

Watches are the simplest way to react to changes using Consul.

- Watch for changes in K/V, services, nodes, health checks, and events

- Invoke external handlers when a change is detected. The handler can be any executable, letting operators customize behavior

```
$ consul watch -type key

{
  "type": "key",
  "key": "foo/bar/baz",
  "handler_type": "script",
  "args": ["/usr/bin/my-service-handler.sh", "-redis"]
}
```

Tool: **Consul Template**

Link 3rd party config files to Consul K/V

Consul Template is a standalone application that populates values from Consul and dynamically renders updates to any third party configurations.

Consul template automatically triggers a reload of third party tools when the template is updated.

Nodes are dynamically added and removed from Consul, and the load balancer will be immediately informed of the change without any operator intervention.
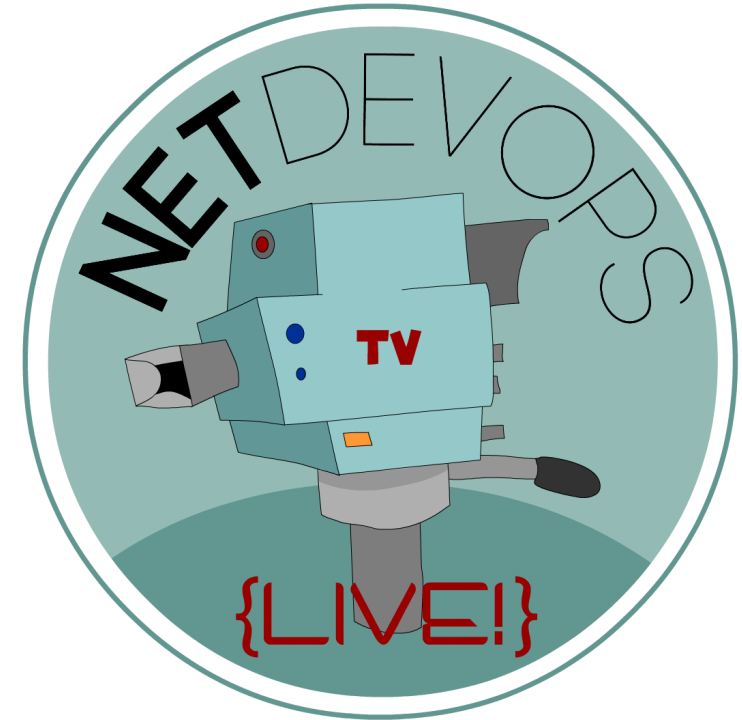
REQUEST VALUES | RETURN VALUES

CONFIG TEMPLATE

CONSUL TEMPLATE

RENDERED CONFIG FILES

APPS

PARSE TEMPLATE

WRITE CONFIGURATIONS

READ CONFIGURATIONS

NetDevOps
Tech Chat

DEVNET
developer.cisco.com

</finish>
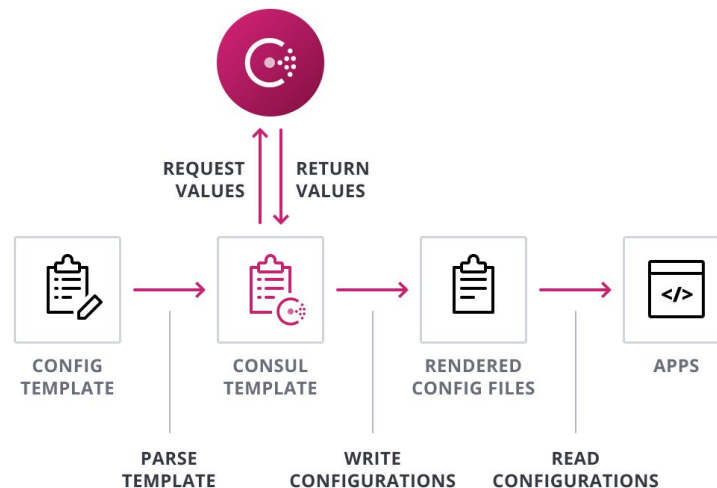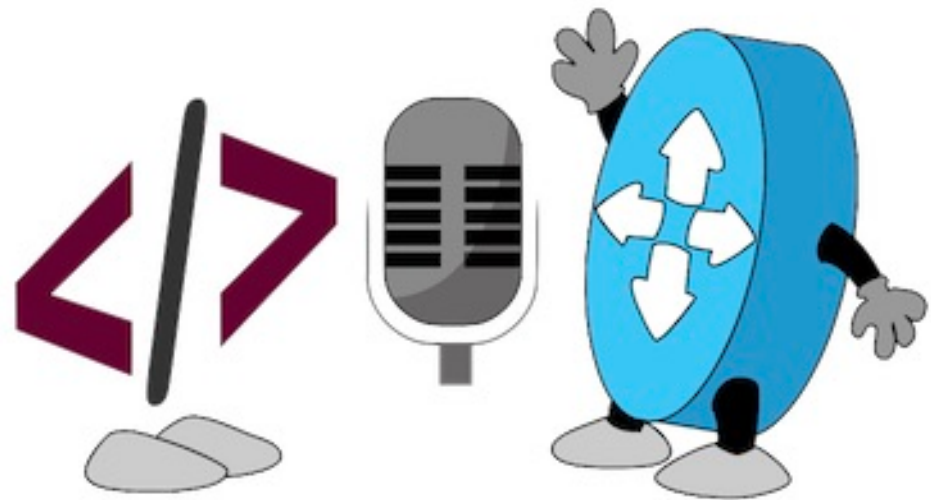
Webinar Resources on DevNet!

- Docs and Links

- Learning Labs

- DevNet Sandboxes

- Code Samples

developer.cisco.com/netdevops/live/#s03t02
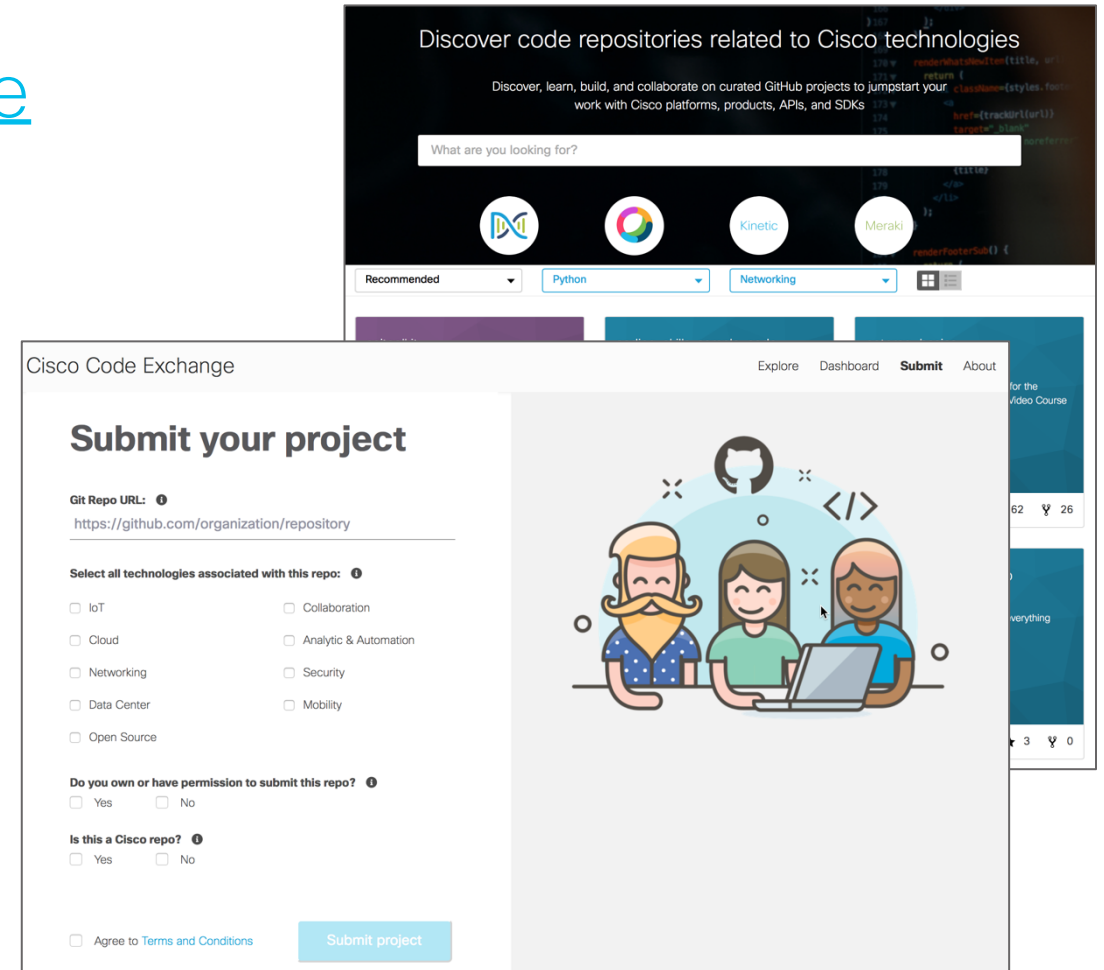
DEVNET
developer.cisco.com

NetDevOps Live! Code Exchange Challenge

developer.cisco.com/codeexchange

Use Terraform with the DevNet ACI Always On Sandbox to Configure a Tenant and Application Profile

Example: You know, everyone knows it, but can you "Infrastructure as Code" the much loved Web/App/Data 3 Tier Architecture?

DEVNET
developer.cisco.com

Looking for more about NetDevOps?

- NetDevOps on DevNet
  developer.cisco.com/netdevops

- NetDevOps Live!
  developer.cisco.com/netdevops/live

- NetDevOps Blogs
  blogs.cisco.com/tag/netdevops

- Network Programmability Basics Video Course
  developer.cisco.com/video/net-prog-basics/

DEVNET
developer.cisco.com

Join us next week!

Give your Network a REST
with Postman

May 5, 2020 with Kevin Swiber

https://developer.cisco.com/netdevops/live/#s03t05

DEVNET
developer.cisco.com

Got more questions? Stay in touch!

Hank Preston

hapresto@cisco.com

@hfpreston

http://github.com/hpreston

CISCO DEVNET
LEARN   CODE   INSPIRE   CONNECT

developer.cisco.com

@CiscoDevNet

facebook.com/ciscodevnet/

http://github.com/CiscoDevNet

DEVNET
developer.cisco.com

NETDEVOPS {LIVE!}

CISCO DEVNET

https://developer.cisco.com/netdevops/live
@netdevopslive