



You make **possible**



How to programmatically migrate from traditional network to ACI with Terraform

Nicolas Vermandé
Technical Marketing Engineer
IBNG

 @nvermande

DEVNET-2585

CISCO *Live!*

Barcelona | January 27-31, 2020



Agenda

- Terraform Concepts

- Project Structure
- Import
- Variables
- Operations

- Terraform with ACI

- ACI object model
- Resource anatomy

- Migration Scenario

- Milestones Definition
- Migration as Code
- Demo

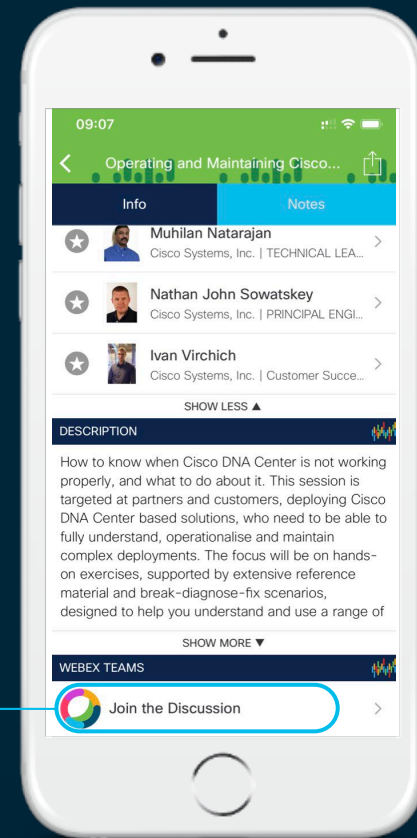
Cisco Webex Teams

Questions?

Use Cisco Webex Teams to chat with the speaker after the session

How

- 1 Find this session in the Cisco Events Mobile App
- 2 Click “Join the Discussion”
- 3 Install Webex Teams or go directly to the team space
- 4 Enter messages/questions in the team space



What is Terraform?

Terraform is an Infrastructure Resources Manager

Composable infrastructure

- Compose and combine infrastructure resources to build and maintain a desired state
- Plan and execution are distinct actions
- Manages all resources through APIs
- Resources and data can be re-used within modules
- Terraform use core and plugins components for basic functions and extensibility



How to install it?

- Go to www.terraform.io/download.html

Download Terraform

Below are the available downloads for the latest version of Terraform (0.11.11). Please download the proper package for your operating system and architecture.

You can find the SHA256 checksums for Terraform 0.11.11 online and you can verify the checksums signature file which has been signed using HashiCorp's GPG key. You can also download older versions of Terraform from the releases service.

Check out the v0.11.11 CHANGELOG for information on the latest release.



- Pick you platform
- Unzip
- Move binary somewhere in your PATH (e.g: /usr/local/bin)
- Run terraform commands

Terraform defines a high level syntax

- Resources are declared in TF file
- Syntax is HCL – HashiCorp Configuration Language
- Human understandable

```
34 provider "aci" {
35     username      = var.aci_user
36     private_key   = var.aci_private_key
37     cert_name     = var.aci_cert_name
38     url           = var.apic_url
39     insecure      = true
40 }

51 resource "aci_tenant" "terraform_ten" {
52     name = var.tenant_name
53 }

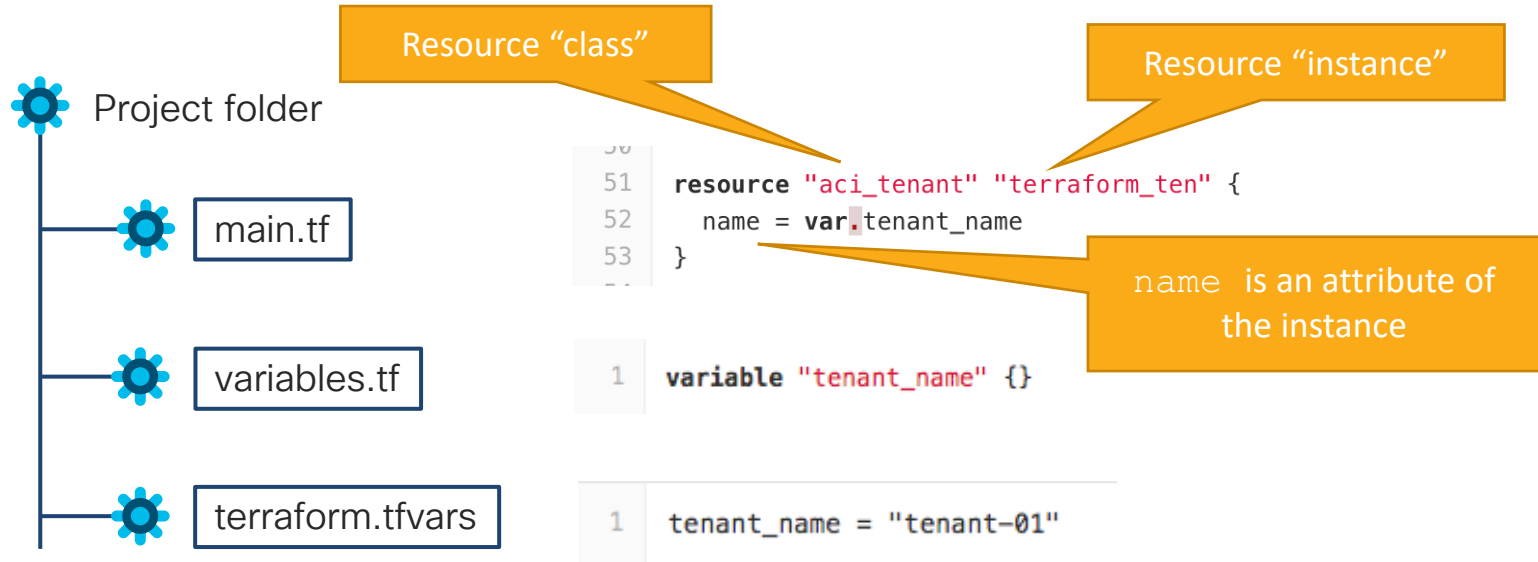
54
55 resource "aci_vrf" "vrf1" {
56     tenant_dn = aci_tenant.terraform_ten.id
57     name      = var.vrf_name
58 }
```


Terraform Providers

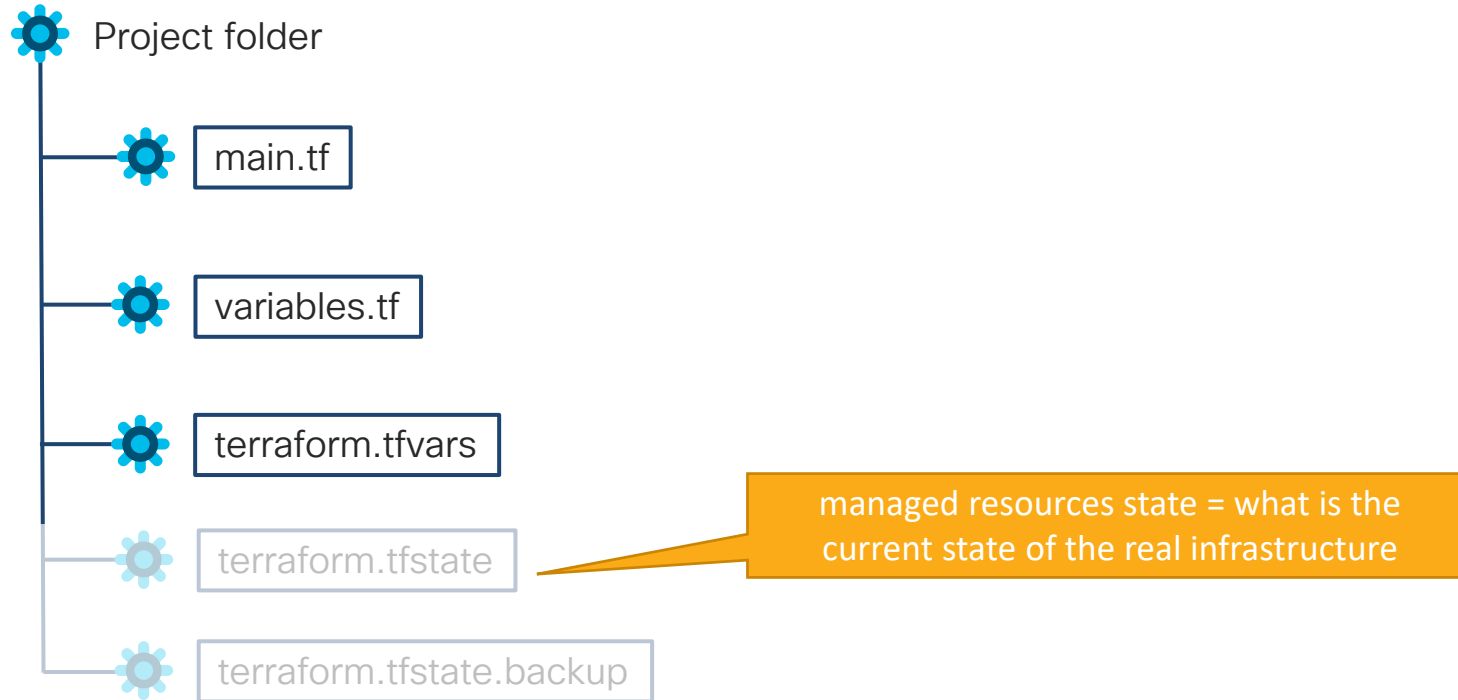
- Providers abstract the API layer of resources providers
 - New resources are available for Terraform to provision and manage
- AAA configuration is required as part of the provider definition in the TF file
- ACI supports user and X509 certificate based authentication



Arguments are used to compose Resources



Arguments are used to compose Resources



Import Resources

- Define Resources in APIC as usual

Import Resources

APIC



Tenant

VRF

Import Resources

- Define Resources in APIC as usual
- Add imported resources in your configuration file as well as additional resources to be created

Import Resources

```
resource "aci_tenant" "myTenant" {  
  }  
  
resource "aci_vrf" "vrf1" {  
  }
```

Configuration File



Import Resources

APIC



Tenant

VRF

Configuration File



Tenant

BD

VRF

EPG

Import Resources

- Define Resources in APIC as usual
- Add imported resources in your configuration file as well as additional resources to be created
- Run import command to import resources in the state file

Import Resources

```
terraform import aci_tenant.myTenant uni/tn-myTenant
```

```
terraform import aci_vrf.vrf1 uni/tn-myTenant/ctx-vrf1
```

Import Resources

APIC



Tenant

VRF

State File



Tenant

VRF

Configuration File



Tenant

VRF

BD

EPG

Import Resources

- Define Resources in APIC as usual
- Add imported resources in your configuration file as well as additional resources to be created
- Run import command to import resources in the state file
- Run terraform plan/apply to create remaining resources and update the state file

Import Resources

APIC



State File



Configuration File



Tenant

BD

VRF

EPG

Tenant

BD

VRF

EPG

Tenant

BD

VRF

EPG

Data Sources vs Resources

Resources	Data sources
<ul style="list-style-type: none">• Managed by Terraform• RW• Can be used for interpolation• terraform apply and destroy modifies resources• Displayed by terraform show and terraform state list <pre>resource "aci_tenant" "terraform_ten" { name = "terraform_ten" }</pre>	<ul style="list-style-type: none">• Managed by Provider• RO• Can be used for interpolation• terraform apply and destroy don't alter data sources lifecycle• Displayed by terraform show and terraform state list <pre>data "vsphere_datacenter" "dc" { name = "\${var.vsphere_datacenter}" }</pre>

Interpolation is used to combine Resources

```
60 resource "aci_bridge_domain" "bd1" {  
61     tenant_dn      = aci_tenant.terraform_ten.id  
62     relation_fv_rs_ctx = aci_vrf.vrf1.name  
63     name           = var.bd_name  
64 }
```

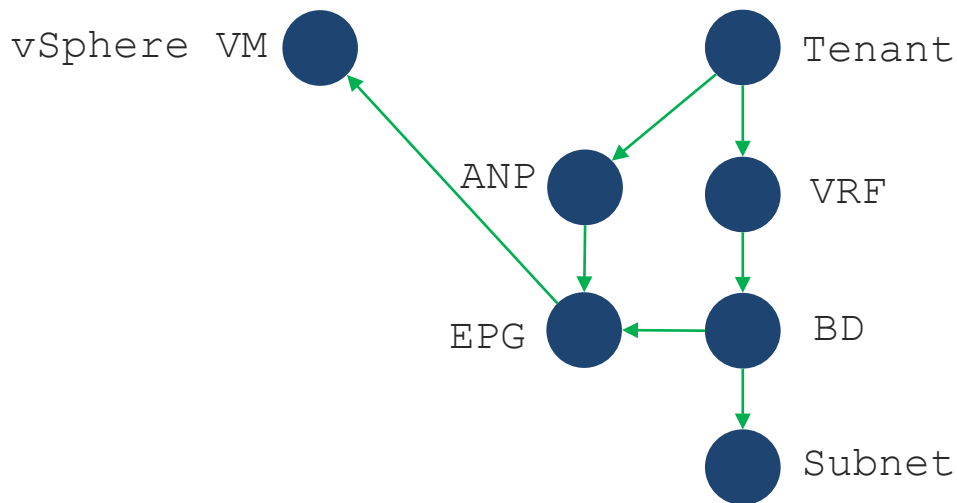


Reference to another resource
(aci_vrf.vrf1)

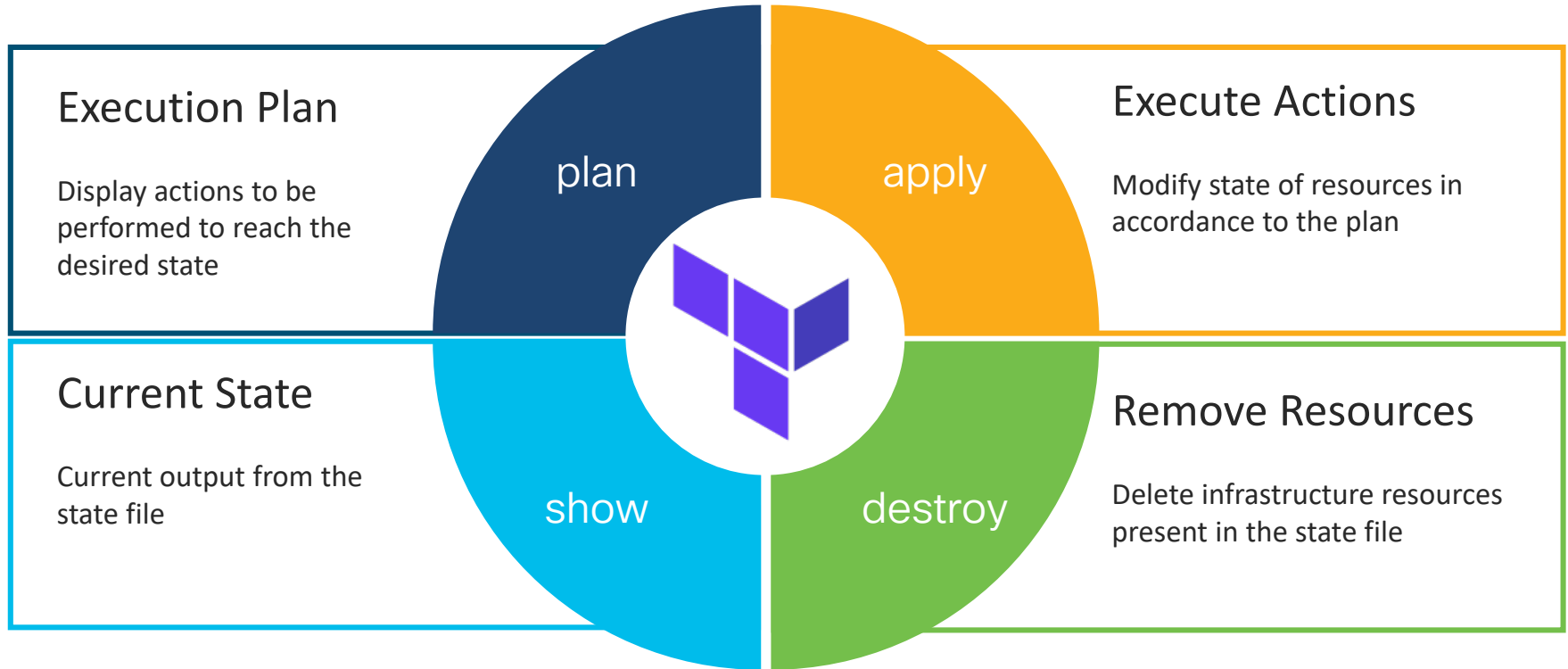
- Interpolation indicates dependency
- Used to build Direct Acyclic Graph (DAG) of dependencies
 - Determines the order of Terraform tasks
- References attributes from other resources

Terraform manages dependencies with DAG

- DAG – Direct Acyclic Graph is a directional tree without loop
- Walking through multiple graphs is handled in parallel



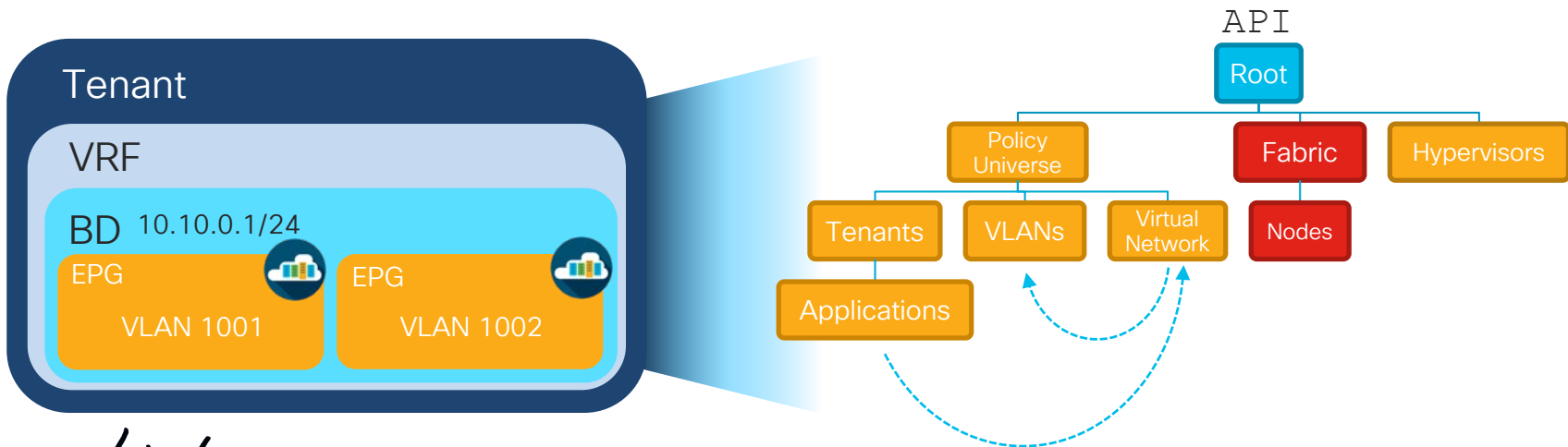
Main Commands



Anatomy of an ACI Resource in Terraform

Infrastructure as Code with ACI

- ACI has a modeled representation of everything APIC knows
- ACI object model is a distributed MIT (Management Information Tree) structure, fully accessible through REST API
- Every node is a managed object (MO) with class, attributes and a distinguished name (Dn)



Terraform and ACI Resource Mapping

- Terraform identifies ACI objects with their Dn
- Terraform resource id is the absolute path of ACI object in the DMIT

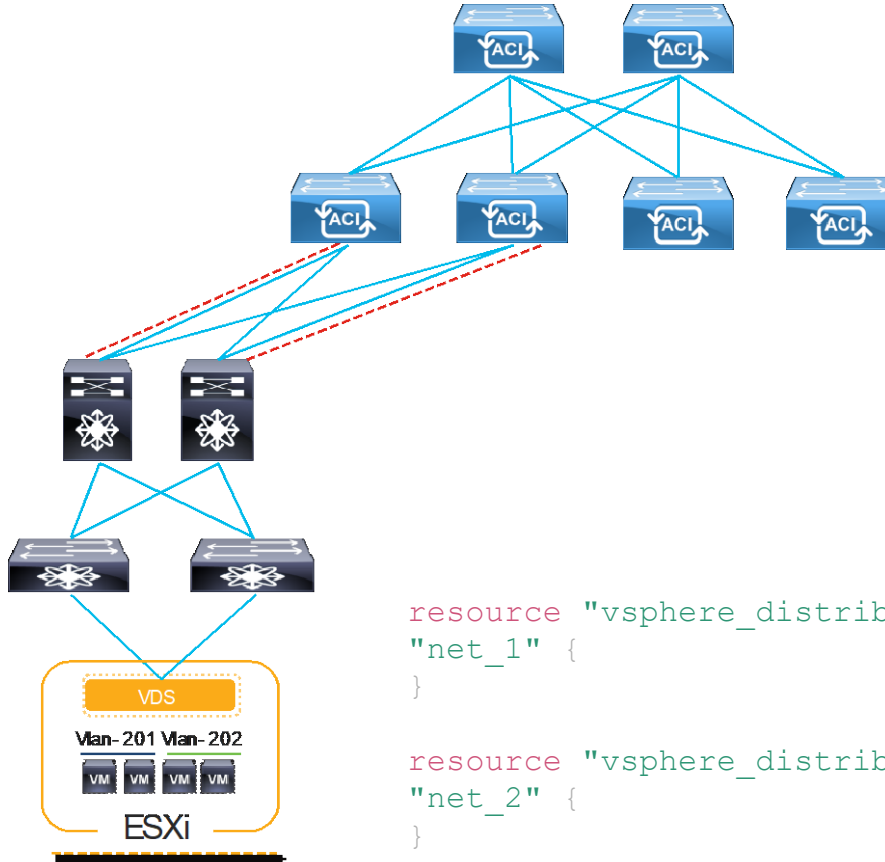
```
nvermand@nvermand-us-main:~/terraform/ACI$ terraform state show aci_filter_entry.https
```

```
id = uni/tn-terraform_ten/flt-allow_https/e-https
```

```
apply_to_frag = no
arp_opc       = unspecified
d_from_port   = https
d_to_port     = https
description   =
ether_t       = ip
filter_dn     = uni/tn-terraform_ten/flt-allow_https
icmpv4_t      = unspecified
icmpv6_t      = unspecified
match_dscp    = unspecified
name          = https
name_alias    =
prot          = tcp
s_from_port   = unspecified
s_to_port     = unspecified
stateful      = yes
tcp_rules     =
```

Migrating to ACI with terraform

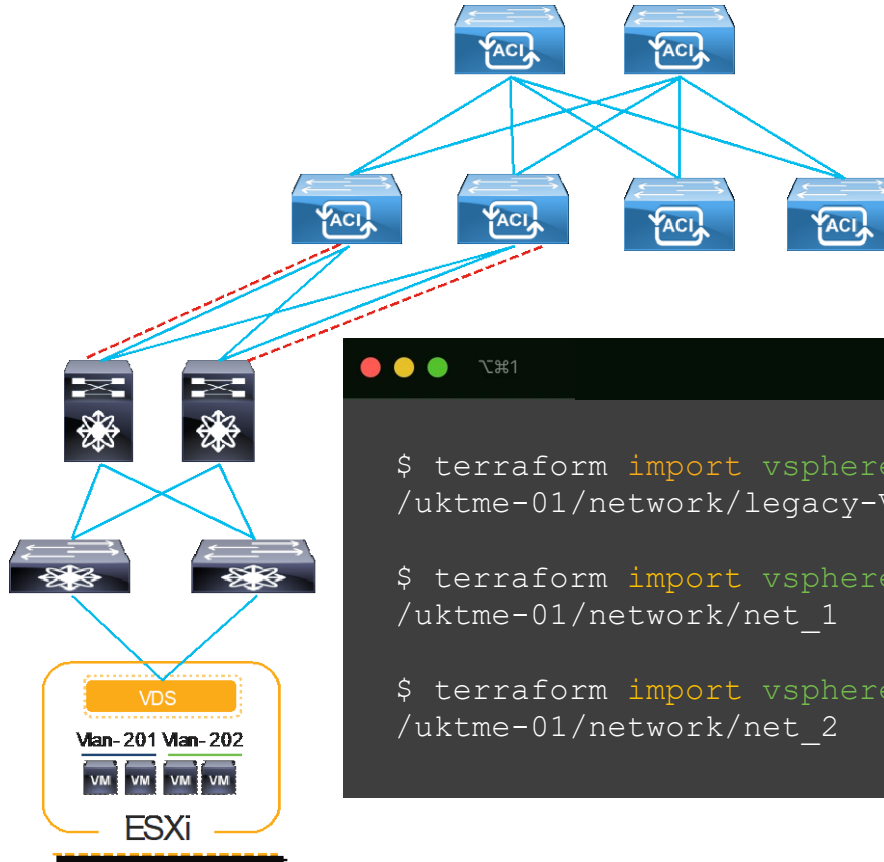
Scenario



```
resource "vsphere_distributed_port_group"
"net_1" {
}
```

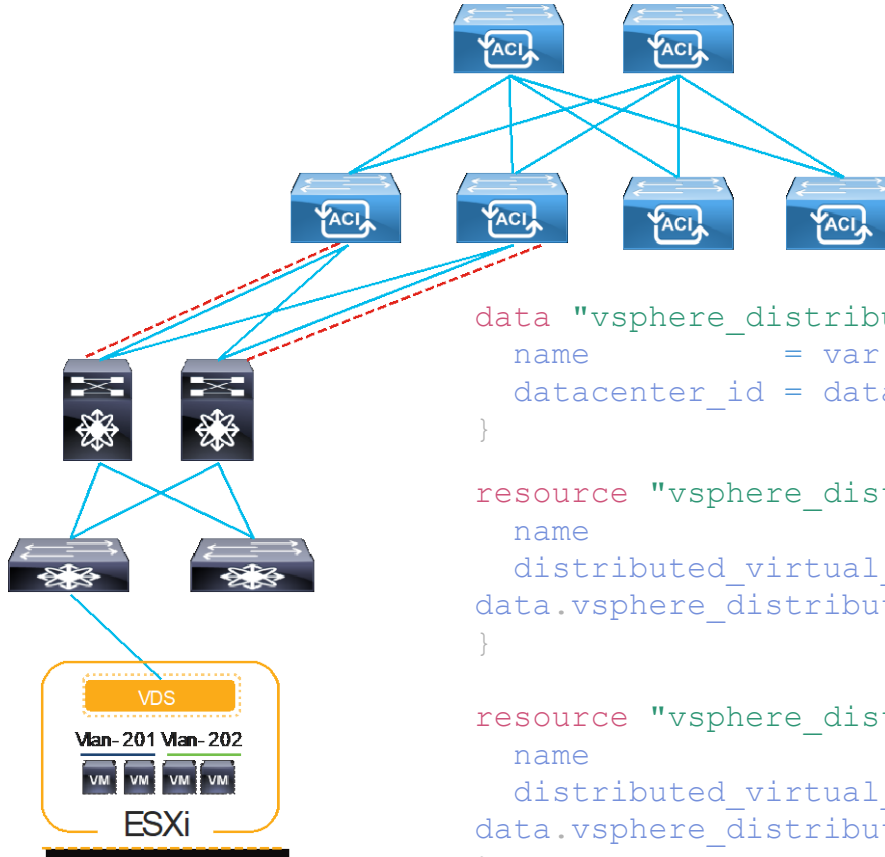
```
resource "vsphere_distributed_port_group"
"net_2" {
}
```

Scenario



```
$ terraform import vsphere_distributed_virtual_switch.legacy-VDS \  
/uktme-01/network/legacy-VDS  
  
$ terraform import vsphere_distributed_port_group.net_1 \  
/uktme-01/network/net_1  
  
$ terraform import vsphere_distributed_port_group.net_2 \  
/uktme-01/network/net_2
```

Scenario

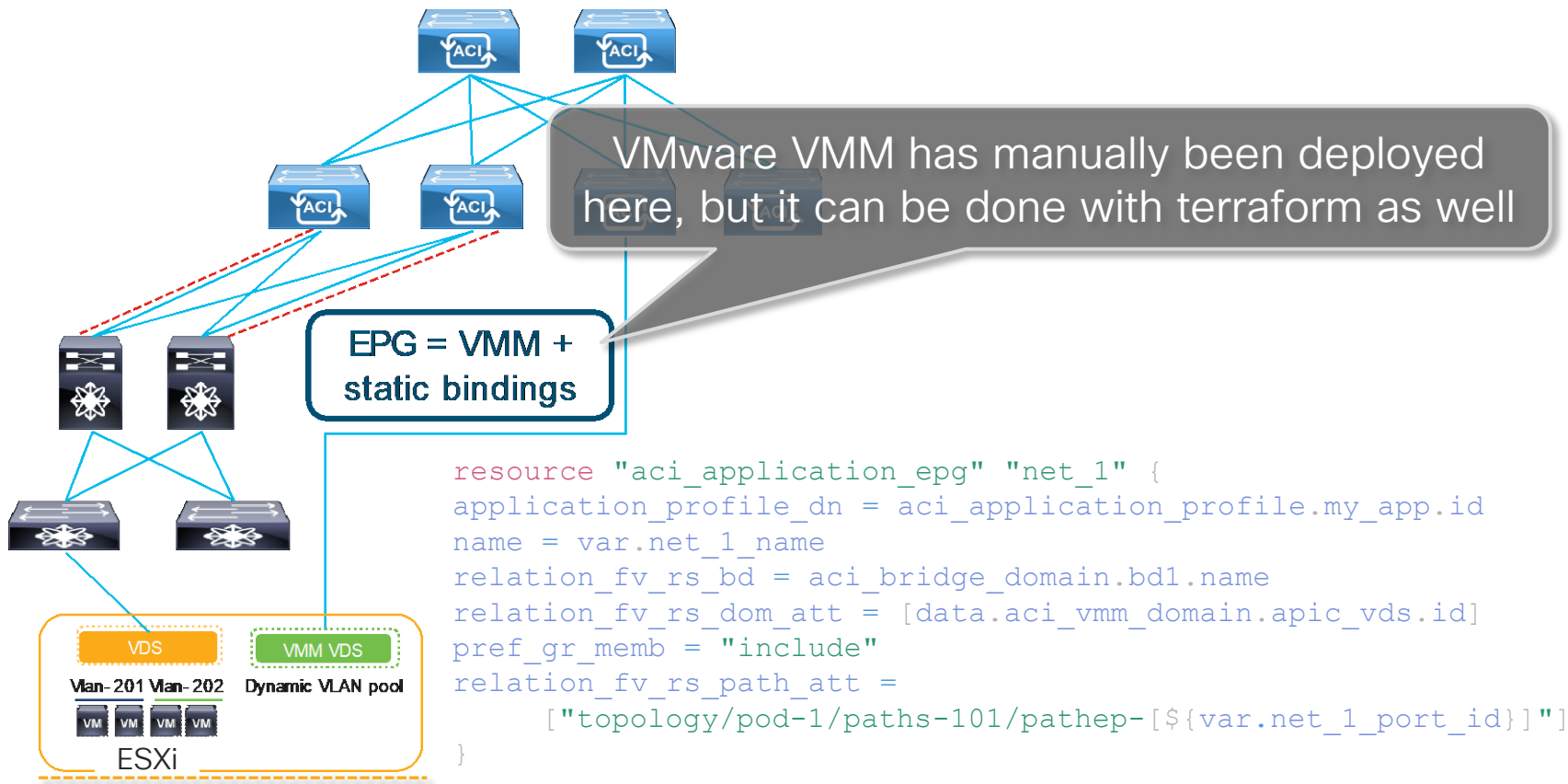


```
data "vsphere_distributed_virtual_switch" "legacy-VDS" {  
  name           = var.vsphere_dvs  
  datacenter_id = data.vsphere_datacenter.uktme-01.id  
}
```

```
resource "vsphere_distributed_port_group" "net_1" {  
  name                       = var.net_1_name  
  distributed_virtual_switch_uuid =  
    data.vsphere_distributed_virtual_switch.legacy-VDS.id  
}
```

```
resource "vsphere_distributed_port_group" "net_2" {  
  name                       = var.net_2_name  
  distributed_virtual_switch_uuid =  
    data.vsphere_distributed_virtual_switch.legacy-VDS.id  
}
```


Scenario



Remaining Tasks

- Import Virtual Machines in Terraform
- Change Virtual Machines Network Portgroup
- Delete old Portgroups when migration is done

Let's see this in action!

Final Thoughts

- Importing Resources vs Creating Resources
 - Is creation idempotent? If yes, you can safely create resources
- How to work with Terraform as a team?
 - Terraform backend can be remote (S3, etcd, consul, Azure RM, etc). It includes state file and locks
 - Authentication
 - Enable remote operations
 - Configuration file should be subject to PR

Call to Actions

- Install Terraform and declare the ACI provider
- Start doing Infrastructure as Code!
- Provide feedback or log issues/feature requests at <https://github.com/terraform-providers/terraform-provider-aci>
- The ACI provider also covers Cloud APIC!

Other Resources

DEVNET-2340 Infrastructure as Code with Terraform and Cisco ACI: don't call me a CLI junkie anymore

DEVNET-2618 Automating ACI Operations on AWS

<https://www.terraform.io/intro/index.html>

Q&A

Complete your online session survey



- Please complete your session survey after each session. Your feedback is very important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (starting on Thursday) to receive your Cisco Live t-shirt.
- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Content Catalog on ciscolive.com/emea.

Cisco Live sessions will be available for viewing on demand after the event at ciscolive.com.

Continue your education



Demos in the
Cisco Showcase



Walk-In Labs



Meet the Engineer
1:1 meetings



Related sessions



Thank you





You make **possible**