
cisco Live!
ALL IN

The bridge to possible

Infrastructure as Code for ACI with Terraform

Thomas Renzy – Technical Leader Customer Experience
@ThomasRenzy
BRKDCN-1811

Cisco Webex App

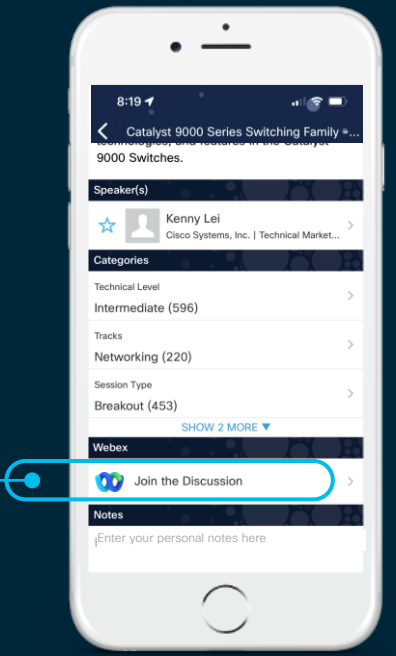
Questions?

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

How

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

Webex spaces will be moderated by the speaker until June 17, 2022.



https://ciscolive.ciscoevents.com/ciscolivebot/#BRKDCN-1811

Agenda

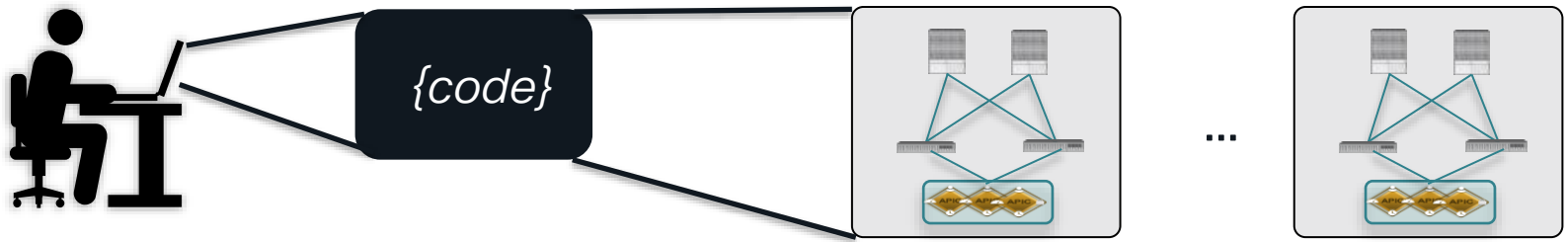
- Introduction to Terraform
- Terraform with ACI
- ACI Terraform Demos
- Next steps

Introduction to Terraform

Infrastructure as Code

- Using/Writing code to describe infrastructure
- Automate provisioning/repeatable tasks

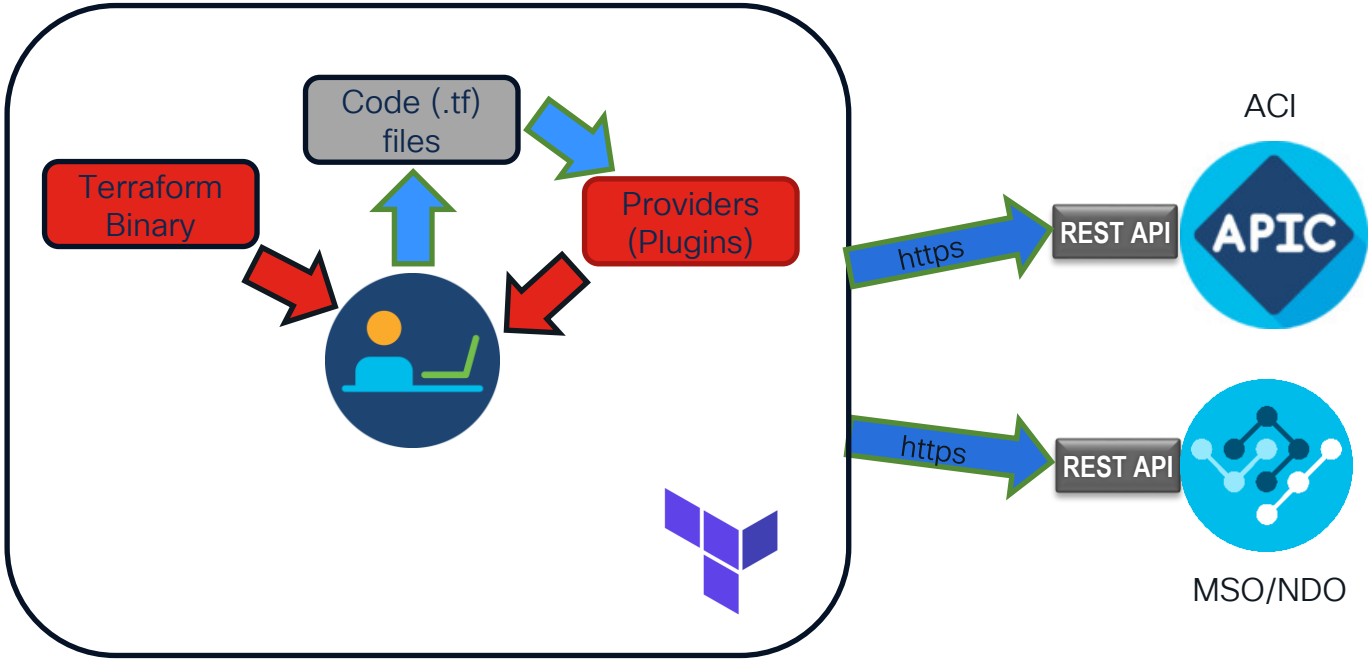
- Leverages Software Tools
  - Version control
  - Documentation
  - Testing
- Provides Speed & Scale



What is Terraform?

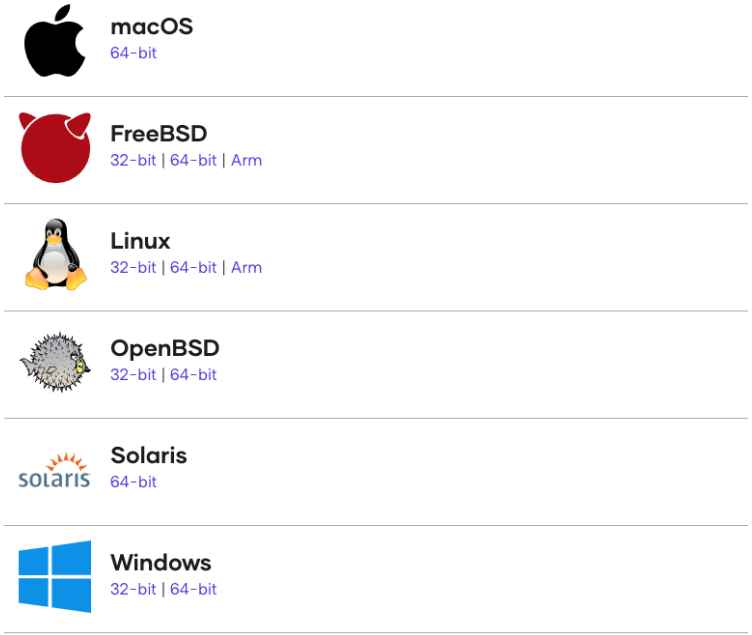
- Open Source
- Infrastructure provisioning tool
- Single binary – Linux, Windows, MacOS
- HashiCorp Configuration Language (HCL)
- Written in Go
  - No programming skills needed
- Declarative
- Leverages Plugins (providers)

Terraform overview



Terraform Installation

- https://www.terraform.io/downloads.html
- Pick your platform
- Unzip and install in your PATH
  - `/usr/local/bin`
  - `C:\Program Files (x86)`
- Run Terraform – That's it.



Terraform with ACI

Terraform Providers

- create/read/update/destroy infrastructure
- Relies on specific plugins
  - Downloaded Dynamically via initialization (via `terraform init`)
- Understands API interactions
  - APIC and MSO REST API calls

- Can use more than one vendor's providers
- Built by Vendor – certified by HashiCorp
- Can write your own providers
  - Terraform is open source
  - Written in Go

Types of Terraform Providers

Official

Owned & maintained by HashiCorp

Ex. AWS, Azure,GCP

Verified

Owned & maintained by partners.

Ex. ACI, MSO, ASA

Community

Published by individual groups or maintainers in the community

```
terraform {
  required_providers {
    aci = {
      source  = "CiscoDevNet/aci"
      version = "2.2.1"
    }
  }
}
```

```
terraform {
  required_providers {
    mso = {
      source = "CiscoDevNet/mso"
    }
  }
}
```

Terraform Resources & Data Sources

Resources

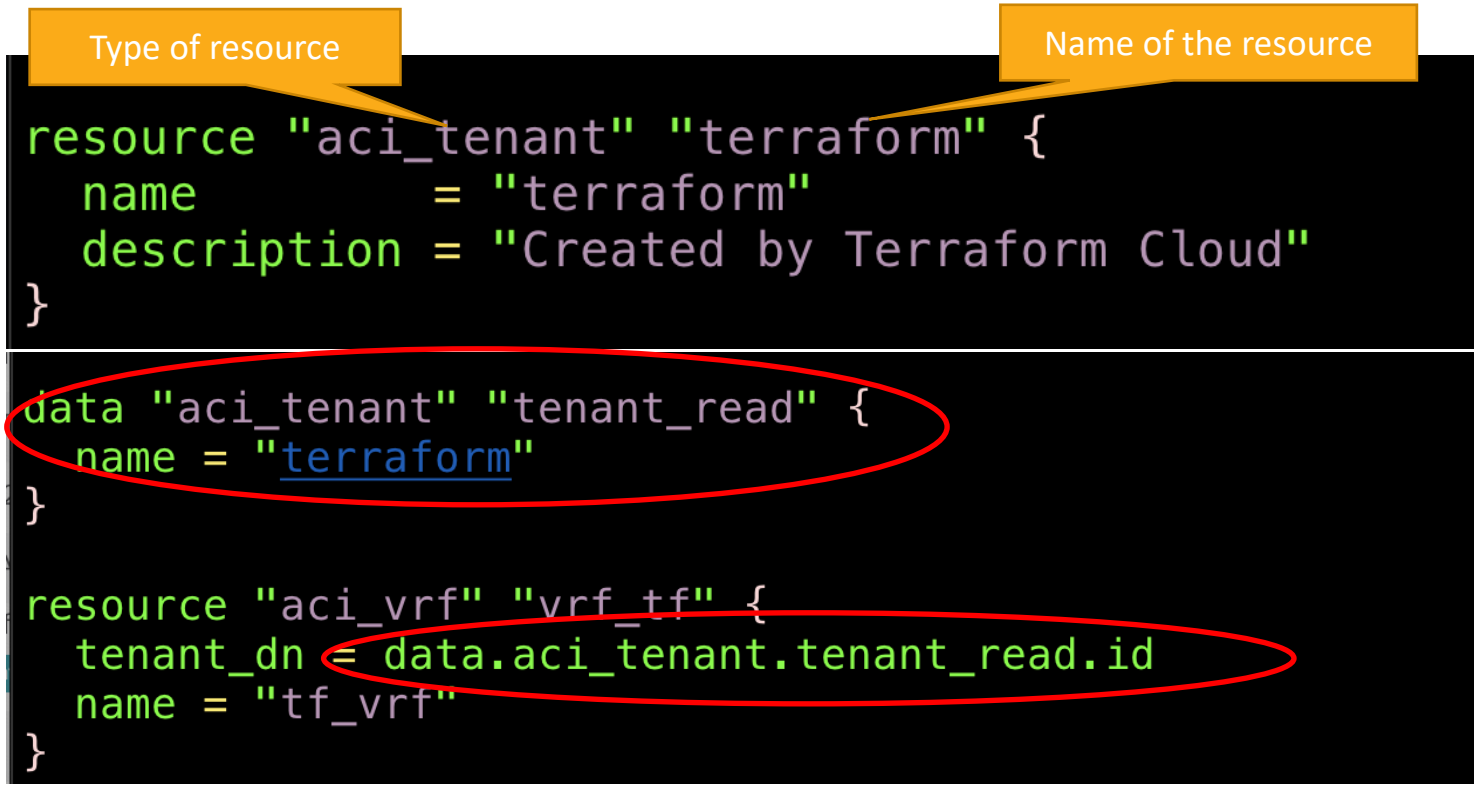
- Resources specific to a given provider
- Always Read/Write
- Terraform apply/destroy modifies resource
- Describes your intent for a particular infrastructure object

Data Sources

- Allow data to be fetched or computed for use elsewhere in Terraform configuration
- Always Read Only
- Terraform apply/destroy does not modify resource
- Query info outside of Terraform

Over 400 ACI Resources/Data Sources
Over 90 MSO Resources/Data Sources

Terraform Resources & Data Sources



Terraform Dependency Mapping

- Keeps track of dependencies and correct order of deployment
- Keeps a graph and state of infrastructure

```
resource "aci_tenant" "terraform" {
  name        = "terraform"
  description = "Created by Terraform Cloud"
}

resource "aci_vrf" "vrf1"{
  tenant_dn = aci_tenant.terraform.id
  name      = "tf-vrf"
}
```

Terraform State

- Records information about infrastructure it created
- Maps resources to configurations – terraform.tfstate
- Backends
  - State storage
  - Local
  - Terraform Cloud
  - AWS S3 bucket


```
{
  "version": 4,
  "terraform_version": "1.2.0",
  "serial": 1,
  "lineage": "9faa9d32-3d38-41b9-f0c5-b901370871ef",
  "outputs": {},
  "resources": [
    {
      "mode": "managed",
      "type": "aci_tenant",
      "name": "terraform",
      "provider": "provider[\"registry.terraform.io/ciscodevnet/aci\"]",
      "instances": [
        {
          "schema_version": 1,
          "attributes": {
            "annotation": "orchestrator:terraform",
            "description": "Created by Terraform Cloud",
            "id": "uni/tn-terraform",
            "name": "terraform",
            "name_alias": "",
            "relation_fv_rs_tenant_mon_pol": "",
            "relation_fv_rs_tn_deny_rule": null
          },
```

Variables

- Can use variables for value substitution
  - Specify type – string, number, boolean
  - Inside plans or external file – `variables.tf/variables.tfvar`

```
variable "tenant_name" {
        default = "terraform" # Name of our Tenant
}
```

```
resource "aci_tenant" "terraform" {
  name        = var.tenant_name
  description = "Created by Terraform Cloud"
}
```

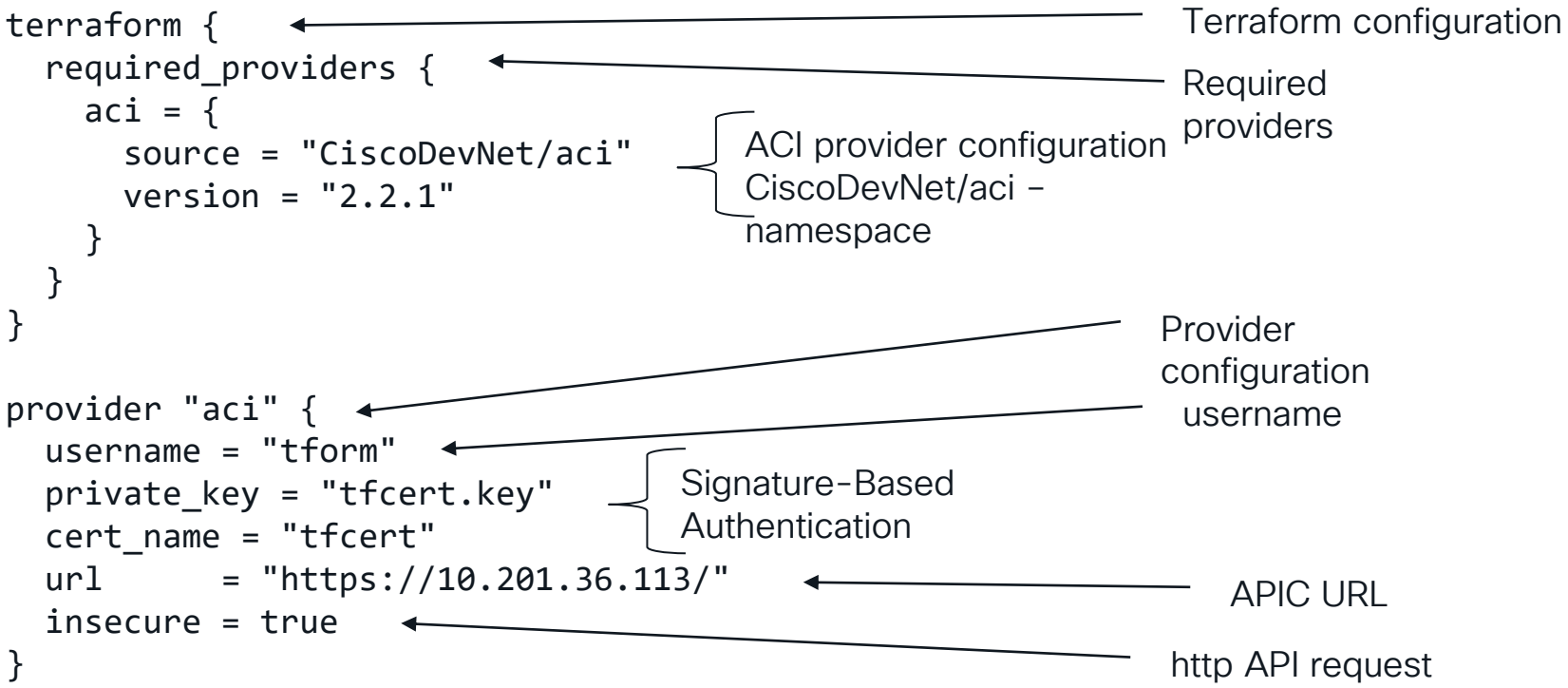
Terraform Plans/Configuration Files

- Collection of HCL instructions
  - What do you want to provision
- Can be in a singular file – main.tf
  - Can be broken up into smaller *.tf
- Can comment lines

  # This is a single line comment

  /* Can use this for a multiline comment */

Terraform Plan example

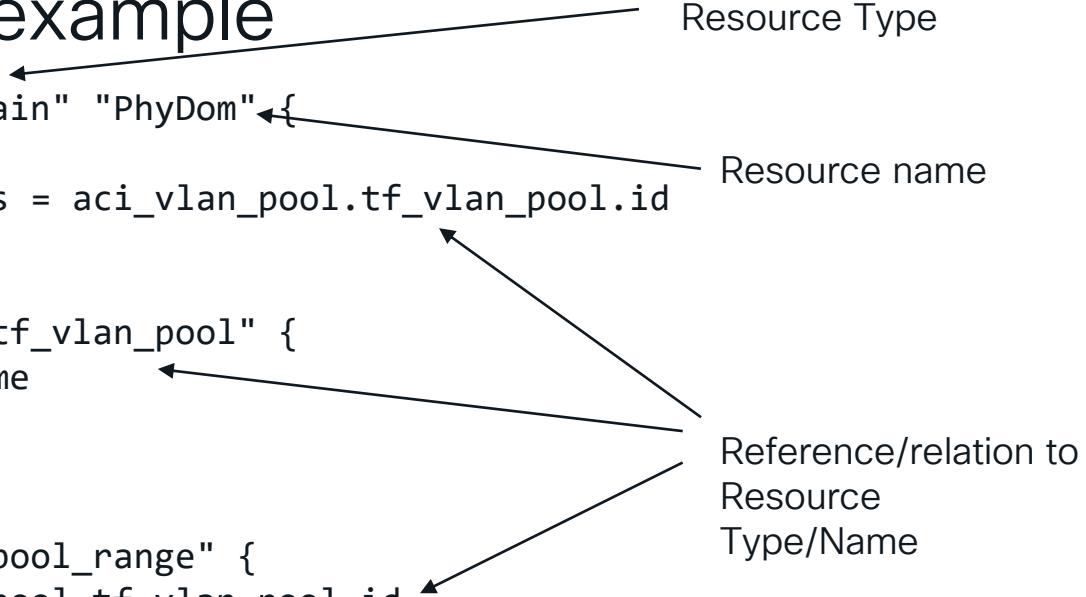


Terraform Plan example

```
resource "aci_physical_domain" "PhyDom" {
  name       = "PhyDom"
  relation_infra_rs_vlan_ns = aci_vlan_pool.tf_vlan_pool.id
}

resource "aci_vlan_pool" "tf_vlan_pool" {
  name       = var.pool_name
  alloc_mode = var.alloc
}

resource "aci_ranges" "tf_pool_range" {
  vlan_pool_dn = aci_vlan_pool.tf_vlan_pool.id
  from         = var.vlan_start
  to           = var.vlan_end
  alloc_mode   = "inherit"
  role         = "external"
}
```



When there isn't a Resource – aci_rest_managed

- Manages Objects via REST API calls with no provider
- Can reconcile state information
- Terraform does not track aci_rest content
- API calls can be captured via API Inspector/APIC GUI
- mso_rest for MSO

```
resource "aci_rest_managed" "aaaUserDomain_all" {
  dn         = "${aci_local_user.tform_user.id}/userdomain-all"
  class_name = "aaaUserDomain"
  content = {
    name = "all"
  }
}
```

Terraform commands

- `terraform init`
  - Installs plugins for configured providers
  - Must initialize before plan/apply
- `terraform plan`
  - determines what actions are necessary to achieve the desired state
- `terraform apply (-auto-approve)`
  - scans the current directory for the configuration
  - Applies the configuration to targets
- `terraform destroy`
  - Infrastructure managed by Terraform will be destroyed.
  - This will ask for confirmation before destroying

Terraform with ACI Demos

A Sample Three Tier Application with Terraform

- We want to do the following:
  - Create a new Tenant – Cisco
  - New VRF – cisco_vrf
  - New BDs – web-bd, app-bd, db-bd
  - Application Profile – cisco_ap
  - 3 EPGs:
    - web_epg , app_epg, db_epg
- 2 Contracts (and associated subjects/filters)
  - web_to_app – Communication between Web EPG and App EPG on http (tcp 80)
  - app_to_db – Communication between App EPG and DB EPG on sql (tcp 1433)

Demo – Three Tier Application

Creating Fabric Access Policies

- We want to do the following:
  - Create a new VLAN pool and VLAN Ranges
    - TF-VLAN-Pool, vlan range 121-130
  - New Physical Domain
  - Fabric policies
    - Link policy – 10G on
    - LLDP on
    - CDP Enable
  - Access port policy group
  - Leaf policies

Demo - Create Fabric Access Policies

Next Steps

Infrastructure as Code with Terraform

- Install Terraform
  - Available for most platforms
- Think big.....start small
  - Automate the simple, then build into more complex tasks
- Ease of writing Infrastructure as code with Terraform
- No special programming skills needed
- Resources/Data Sources for most common tasks
- Terraform and Robust APIC/MSO REST API makes automation easy and scalable

More information

- Walk in Lab – LABDCN-1776 (Intro to Terraform with ACI)
- https://www.terraform.io/
- https://registry.terraform.io/providers/CiscoDevNet/aci/latest/docs
- https://github.com/CiscoDevNet/terraform-provider-aci
- https://github.com/CiscoDevNet/terraform-provider-mso
- https://github.com/trenzy/
- https://developer.cisco.com/automation-terraform/

Technical Session Surveys

- Attendees who fill out a minimum of four session surveys and the overall event survey will get Cisco Live branded socks!
- Attendees will also earn 100 points in the Cisco Live Game for every survey completed.
- These points help you get on the leaderboard and increase your chances of winning daily and grand prizes.

Cisco Learning and Certifications

From technology training and team development to Cisco certifications and learning plans, let us help you empower your business and career. www.cisco.com/go/certs

Pay for Learning with Cisco Learning Credits

(CLCs) are prepaid training vouchers redeemed directly with Cisco.

Learn

Cisco U.
IT learning hub that guides teams and learners toward their goals

Cisco Digital Learning
Subscription-based product, technology, and certification training

Cisco Modeling Labs
Network simulation platform for design, testing, and troubleshooting

Cisco Learning Network
Resource community portal for certifications and learning

Train

Cisco Training Bootcamps
Intensive team & individual automation and technology training programs

Cisco Learning Partner Program
Authorized training partners supporting Cisco technology and career certifications

Cisco Instructor-led and Virtual Instructor-led training
Accelerated curriculum of product, technology, and certification courses

Certify

Cisco Certifications and Specialist Certifications
Award-winning certification program empowers students and IT Professionals to advance their technical careers

Cisco Guided Study Groups
180-day certification prep program with learning and support

Cisco Continuing Education Program
Recertification training options for Cisco certified individuals

Here at the event? Visit us at **The Learning and Certifications lounge at the World of Solutions**

Continue your education

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

cisco Live!

Thank you


cisco Live!
ALL IN