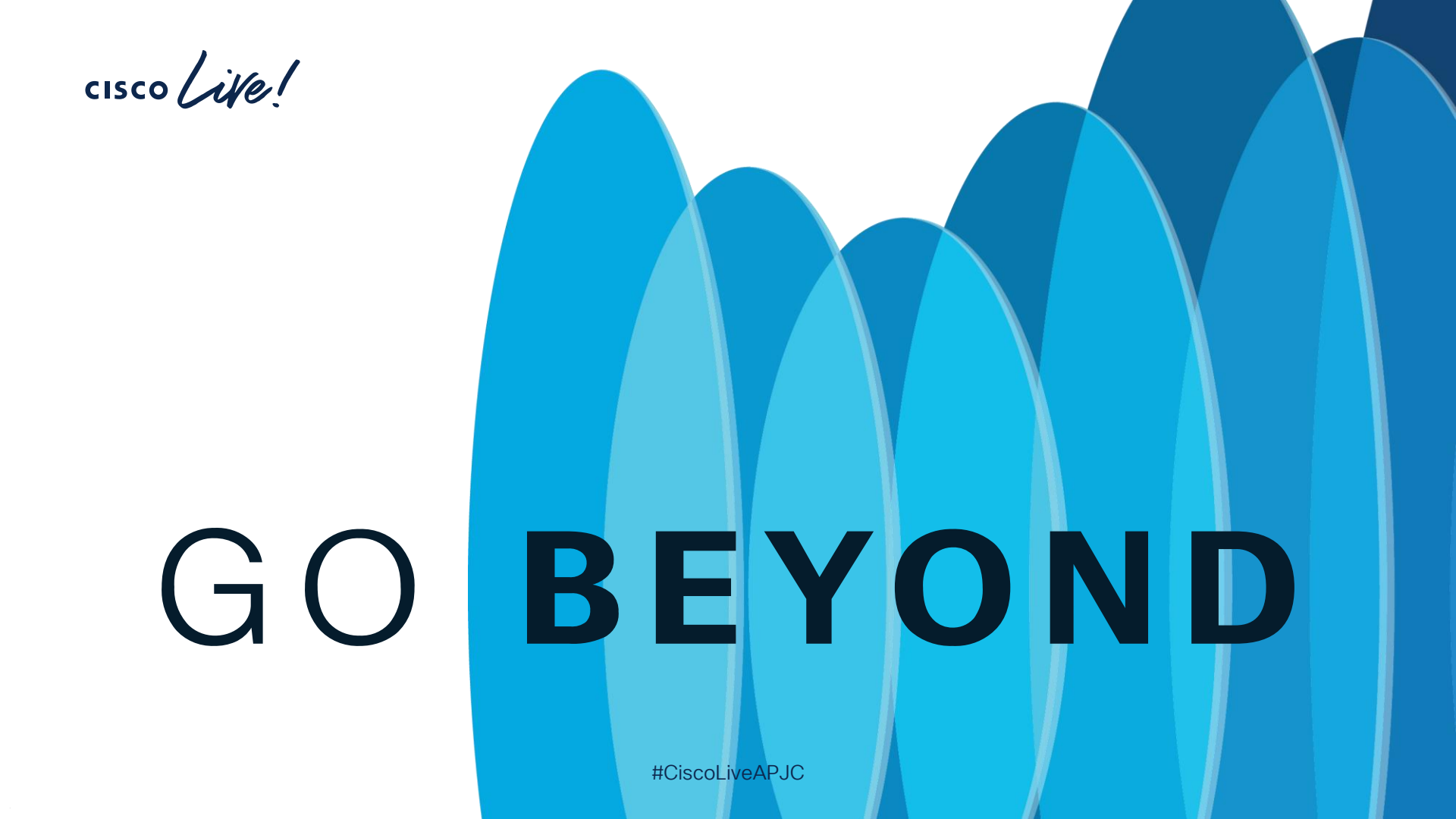

cisco Live!
GO BEYOND
#CiscoLiveAPJC

The power of Terraform and Meraki

Yuji Terada, Solutions Engineer
@terada_yuji
DEVNET-2557

Cisco Webex App

https://ciscolive.ciscoevents.com/ciscolivebot/#DEVNET-2557

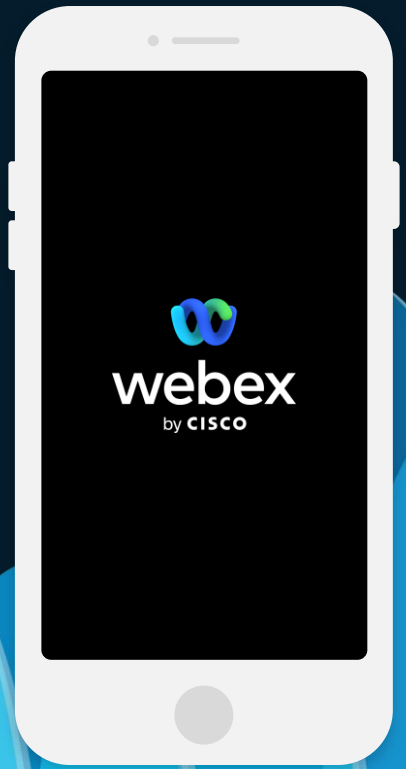
Questions?

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 November 15, 2024.



cisco Live!

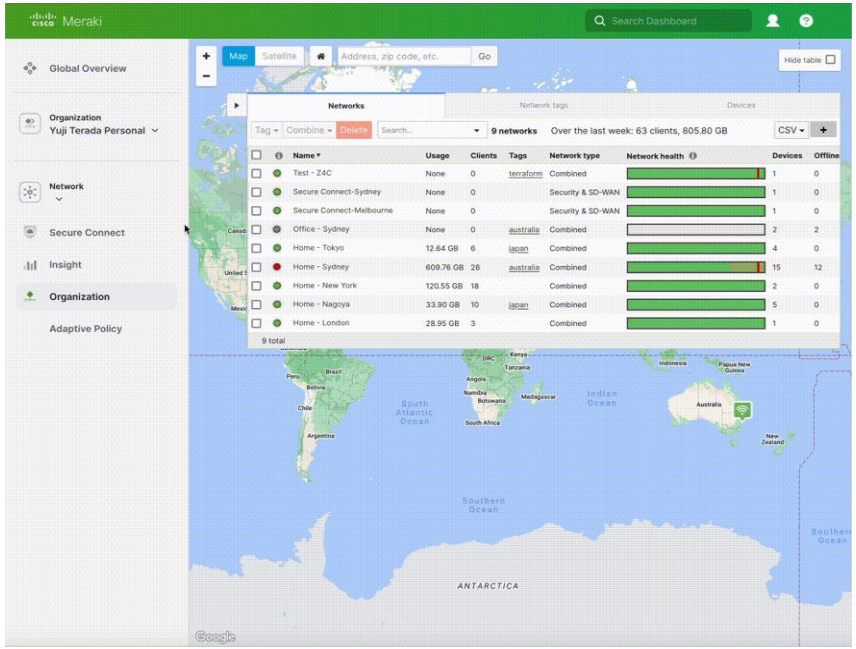
© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Simplifying powerful technology to free passionate people to focus on their mission

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Simplicity to the next level




```
> terraform init

> terraform apply
```


© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

About me

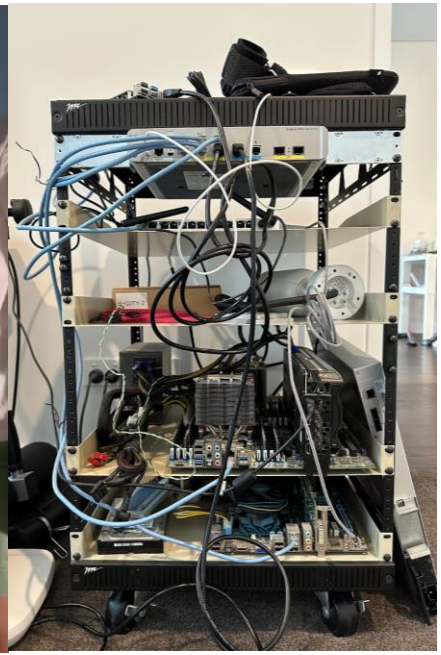
Yuji Terada

Career

Apr 2014	Support Engineer @TAC
Feb 2018	Solutions Engineer @Meraki
	Meraki API Ninja
	Meraki Geo Lead

Certifications

- CCIE R&S, Wireless
- CCNP DevNet



© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

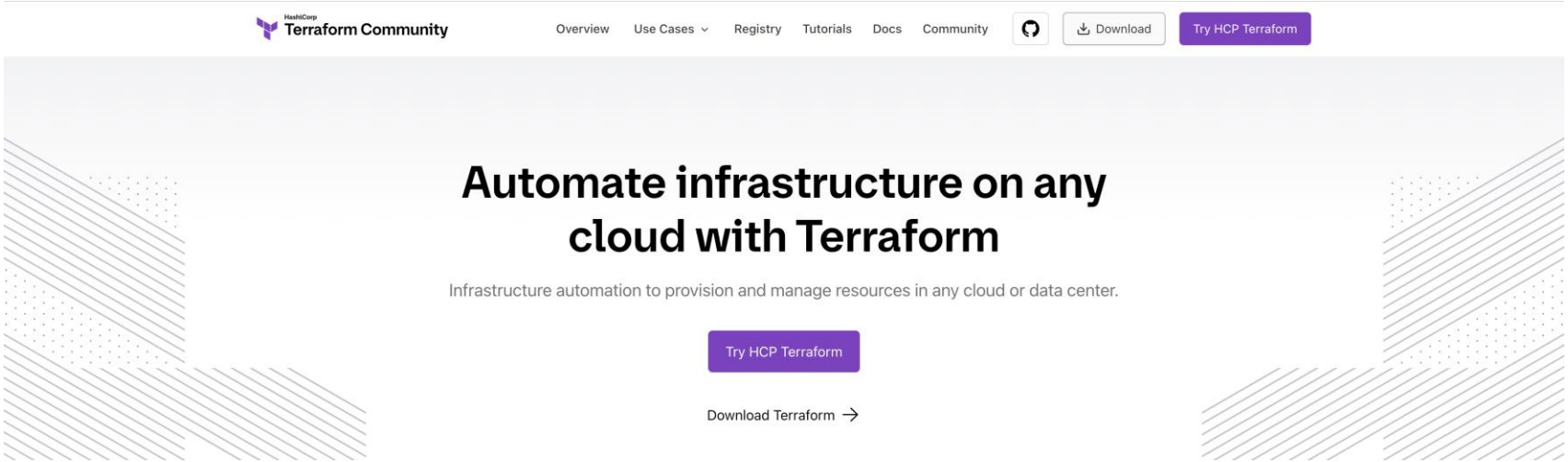
Agenda

- Terraform basics
- Deploying Z4C
- Deploying vMX in AWS
- Conclusion

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Terraform

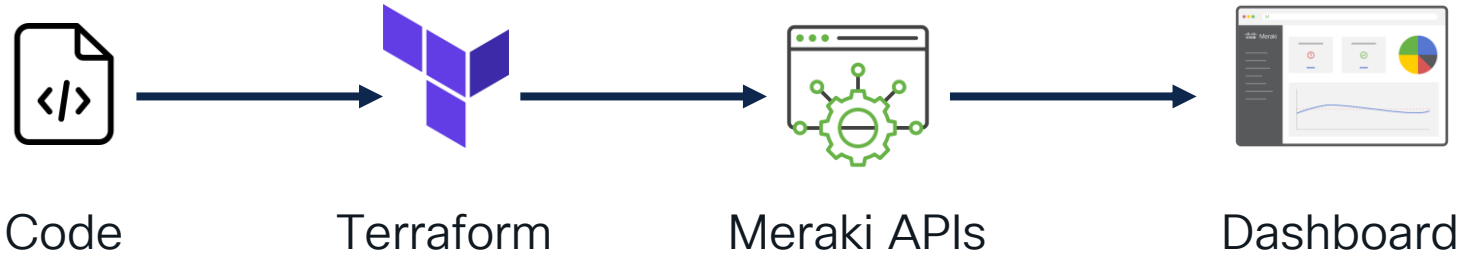
Terraform is an infrastructure as code tool that lets you build, change, and version cloud and on-prem resources safely and efficiently.



https://www.terraform.io

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Terraform under the hood



Ability is limited to the availability of APIs

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Hello world!

```
terraform {
  required_providers {
    meraki = {
      source  = "cisco-open/meraki"
      version = "0.2.11-alpha"
    }
  }
}

provider "meraki" {
  meraki_dashboard_api_key = "12345"
}

resource "meraki_networks" "test_network" {
  organization_id = "734417"
  name            = "Test network"
  product_types   = ["appliance", "switch", "wireless"]
}
```


© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Command: terraform init

```
Initializing the backend...

Initializing provider plugins...
- Finding cisco-open/meraki versions matching "0.2.11-alpha"...
- Installing cisco-open/meraki v0.2.11-alpha...
- Installed cisco-open/meraki v0.2.11-alpha (signed by a HashiCorp
partner, key ID 535815BB4B6133C1)

Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about
it here:
https://www.terraform.io/docs/cli/plugins/signing.html

Terraform has created a lock file .terraform.lock.hcl to record the
provider
selections it made above. Include this file in your version control
repository
so that Terraform can guarantee to make the same selections by
default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform
plan" to see
any changes that are required for your infrastructure. All Terraform
commands
should now work.

If you ever set or change modules or backend configuration for
Terraform,
rerun this command to reinitialize your working directory. If you
forget, other
commands will detect it and remind you to do so if necessary.
```


© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Command: terraform plan

```
Terraform used the selected providers to generate the following
execution plan. Resource actions are indicated with the following
symbols:
  + create

Terraform will perform the following actions:

  # meraki_networks.test_network will be created
  + resource "meraki_networks" "test_network" {
      + enrollment_string           = (known after apply)
      + id                          = (known after apply)
      + is_bound_to_config_template = (known after apply)
      + name                        = "Test network"
      + network_id                  = (known after apply)
      + notes                       = (known after apply)
      + organization_id             = "734417"
      + product_types               = [
          + "appliance",
          + "switch",
          + "wireless",
        ]
      + tags                        = (known after apply)
      + time_zone                   = (known after apply)
      + url                         = (known after apply)
    }

Plan: 1 to add, 0 to change, 0 to destroy.
```

```
─────────────────────────────────────────────────────────────────────
─────────────────────────────────────────────────────────────────────
──────────────────────────────

Note: You didn't use the -out option to save this plan, so Terraform
can't guarantee to take exactly these actions if you run "terraform
apply" now.
```


© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Command: terraform apply

```
Terraform used the selected providers to generate the following
execution plan. Resource actions are indicated with the following
symbols:
  + create

Terraform will perform the following actions:

  # meraki_networks.test_network will be created
  + resource "meraki_networks" "test_network" {
      + enrollment_string           = (known after apply)
      + id                          = (known after apply)
      + is_bound_to_config_template = (known after apply)
      + name                        = "Test network"
      + network_id                  = (known after apply)
      + notes                       = (known after apply)
      + organization_id             = "734417"
      + product_types               = [
          + "appliance",
          + "switch",
          + "wireless",
        ]
      + tags                        = (known after apply)
      + time_zone                   = (known after apply)
      + url                         = (known after apply)
    }

Plan: 1 to add, 0 to change, 0 to destroy.
```

```
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

meraki_networks.test_network: Creating...
meraki_networks.test_network: Creation complete after 6s
[id=L_726205439913499888]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```


© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Command: terraform destroy

```
terraform destroy
meraki_networks.test_network: Refreshing state...
[id=L_726205439913499889]

Terraform used the selected providers to generate the following
execution plan. Resource actions are indicated with the following
symbols:
  - destroy

Terraform will perform the following actions:

  # meraki_networks.test_network will be destroyed
  - resource "meraki_networks" "test_network" {
      - id                          = "L_726205439913499889" -> null
      - is_bound_to_config_template = false -> null
      - name                        = "Test network" -> null
      - network_id                  = "L_726205439913499889" -> null
      - organization_id             = "734417" -> null
      - product_types               = [
          - "appliance",
          - "switch",
          - "wireless",
        ] -> null
      - time_zone                   = "America/Los_Angeles" -> null
      - url                         = "https://n290.meraki.com/Test-
network-app/n/sn4RBdIe/manage/usage/list" -> null
    }

Plan: 0 to add, 0 to change, 1 to destroy.
```

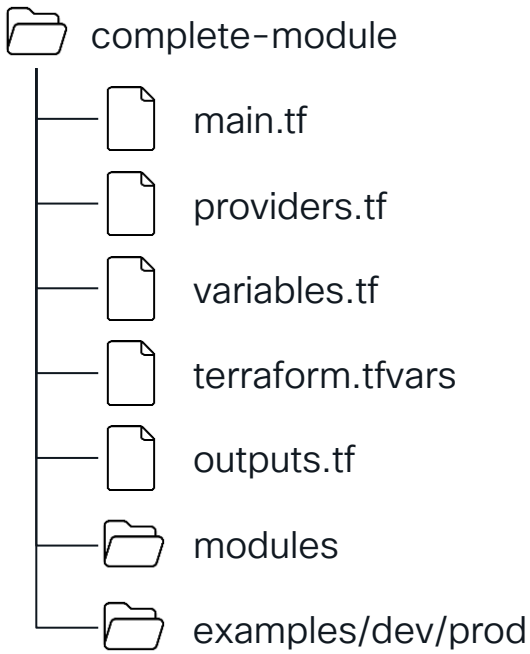
```
Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown
above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

meraki_networks.test_network: Destroying... [id=L_726205439913499889]
meraki_networks.test_network: Destruction complete after 4s

Destroy complete! Resources: 1 destroyed.
```


© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Organise the file

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Folder structure



© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

File: main.tf

```
resource "meraki_networks" "test_network" {
  organization_id = var.organization_id
  name            = var.network_name
  product_types   = ["appliance", "switch", "wireless"]
}
```


© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

File: providers.tf

```
terraform {
  required_providers {
    meraki = {
      source  = "cisco-open/meraki"
      version = "0.2.11-alpha"
    }
  }
}

provider "meraki" {
  meraki_dashboard_api_key = var.meraki_dashboard_api_key
}
```


© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

File: variables.tf

```
variable "meraki_dashboard_api_key" {
  type        = string
  description = "The API key used to authenticate with the Meraki Dashboard"
}

variable "organization_id" {
  type        = string
  description = "The ID of the Meraki organization"
}

variable "network_name" {
  type        = string
  description = "The name of the Meraki network to be created"
}
```

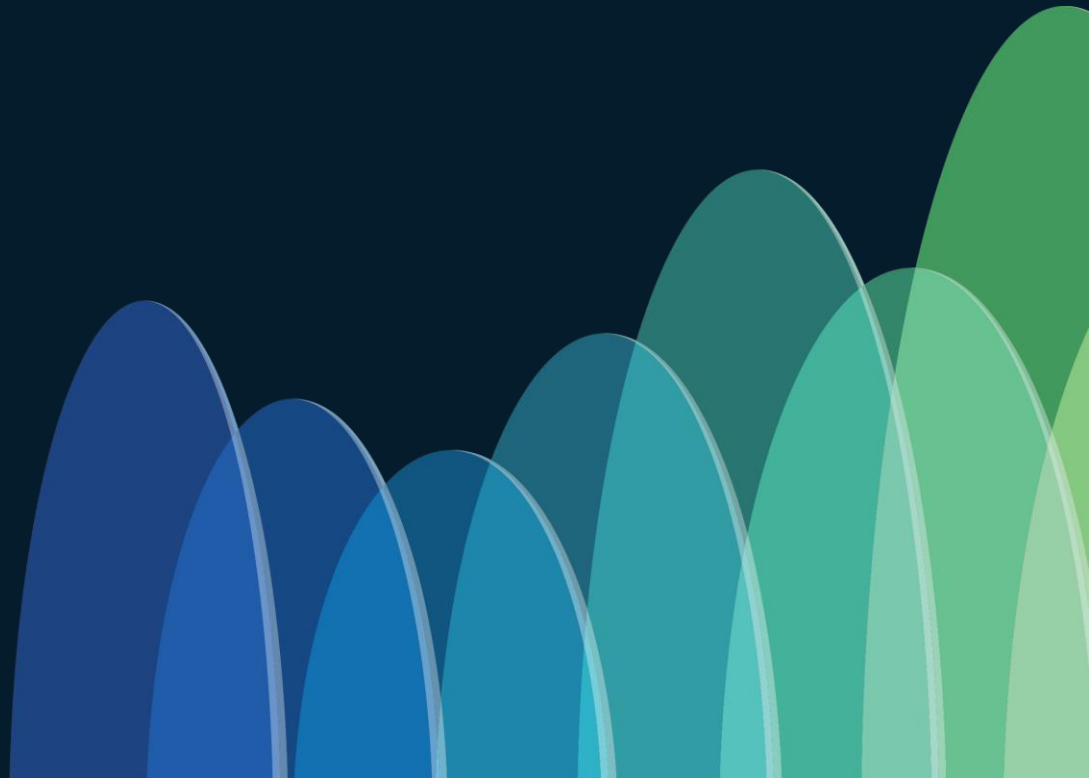

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

File: terraform.tfvars

```
meraki_dashboard_api_key = "12345"
organization_id          = "734417"
network_name             = "Test network"
```


© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Example 1: Create Network with Z4C



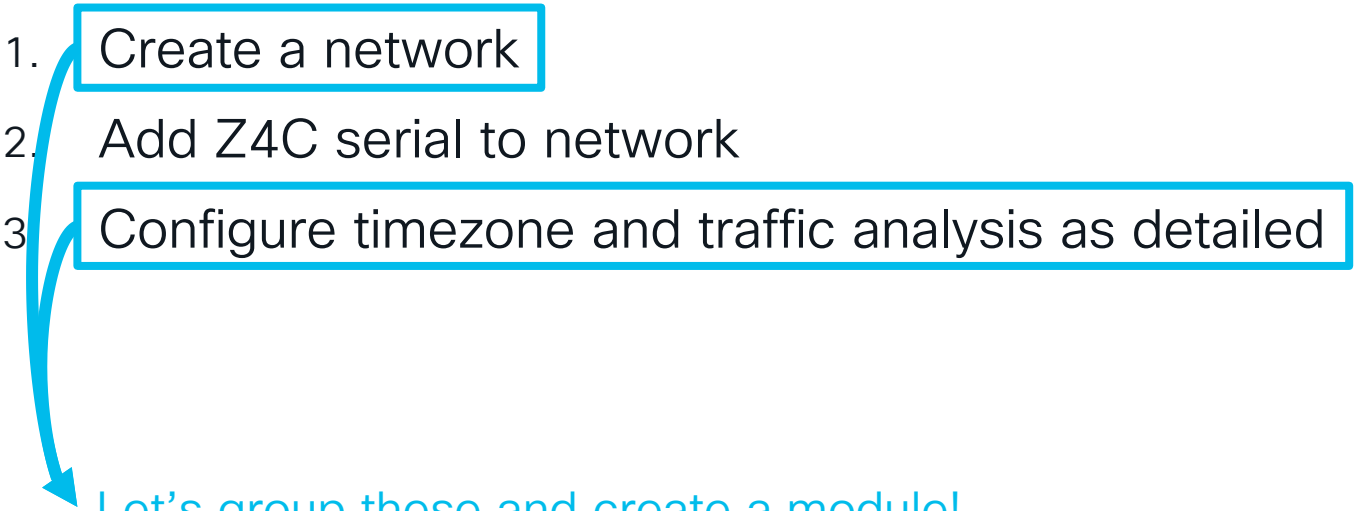
Steps to deploy Z4C

1. Create a network
2. Add Z4C serial to network
3. Configure timezone and traffic analysis as detailed

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Create a module for reusability

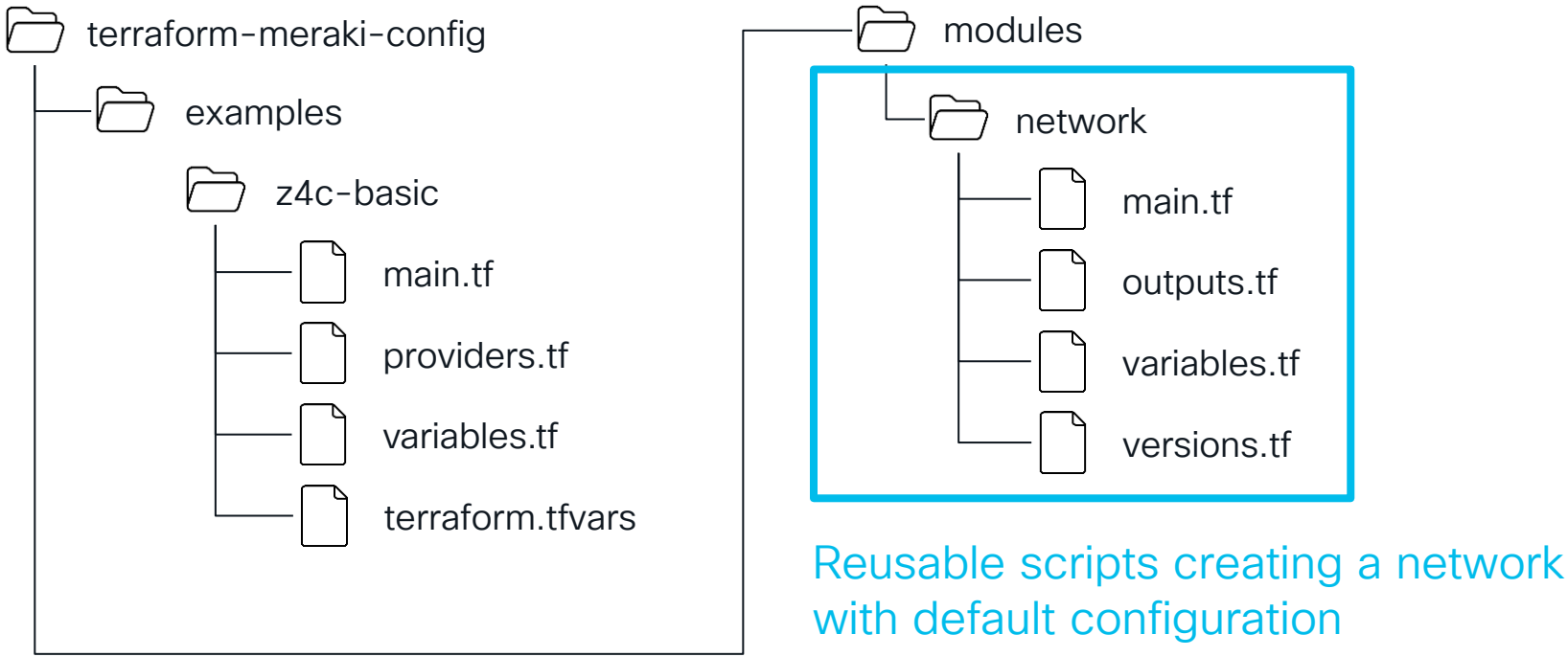
1. Create a network
2. Add Z4C serial to network
3. Configure timezone and traffic analysis as detailed



Let’s group these and create a module!

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Folder structure



Reusable scripts creating a network with default configuration

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

File: ./terraform-meraki-config/modules/network/main.tf

```
resource "meraki_networks" "network" {
  name            = var.network_name
  organization_id = var.organization_id
  notes           = var.notes
  product_types   = var.product_types
  tags            = var.tags
}

resource "meraki_networks_traffic_analysis" "traffic_analysis" {
  mode       = "detailed"
  network_id = meraki_networks.network.id
}
```

Constant because it should always be “detailed”

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

File: ./terraform-meraki-config/modules/network/variables.tf

```
variable "network_name" {
  description = "The name of the Meraki network"
  type        = string
}

...

variable "product_types" {
  description = "List of product types to include in the network"
  type        = list(string)
  default     = ["appliance", "camera", "cellularGateway", "sensor",
"switch", "systemsManager", "wireless"]
}

...
```

Constant and chucking everything in for reusability

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

File: ./terraform-meraki-config/dev/z4c-basic/main.tf

```
module "network" {
  source                  = "../../modules/network"
  network_name            = var.network_name
  organization_id         = var.organization_id
}

locals {
  network_id = module.network.id
}

resource "meraki_networks_devices_claim" "claimed_devices" {
  network_id = local.network_id
  parameters = {
    serials = var.serials
  }
}
```

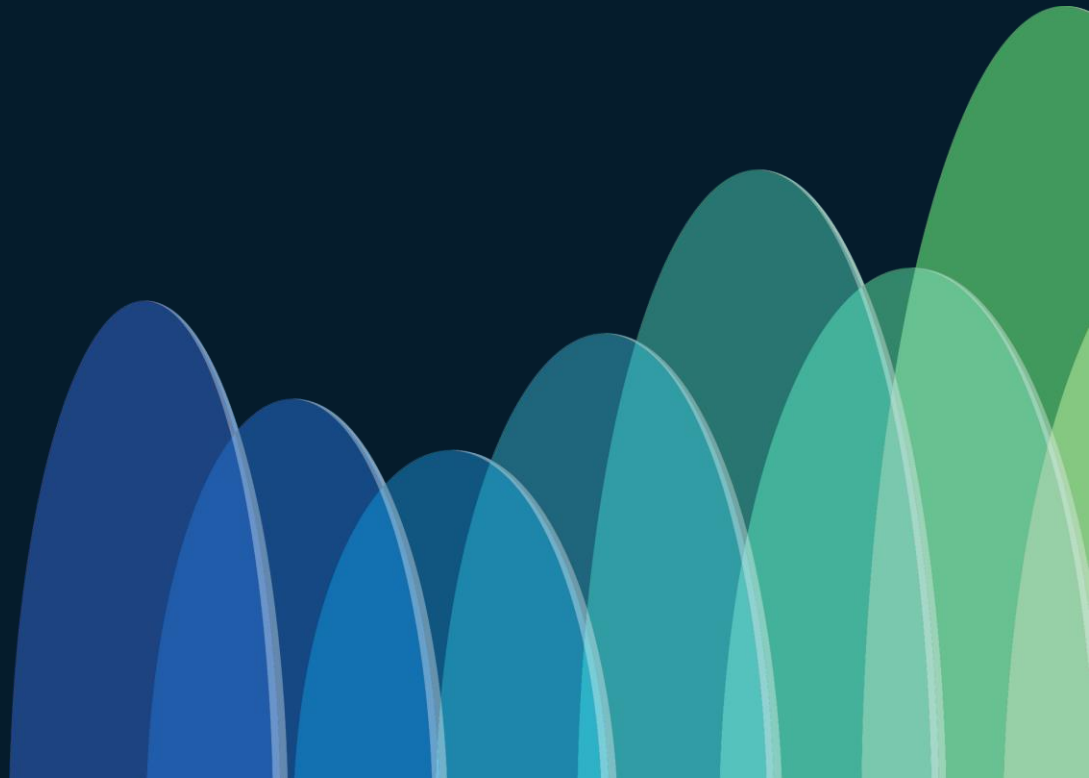

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Adding more configuration

1. Create a network
2. Add Z4C serial to network
3. Configure timezone and traffic analysis as detailed
4. Configure corp and guest VLAN
5. Configure internet only ACL for guest VLAN

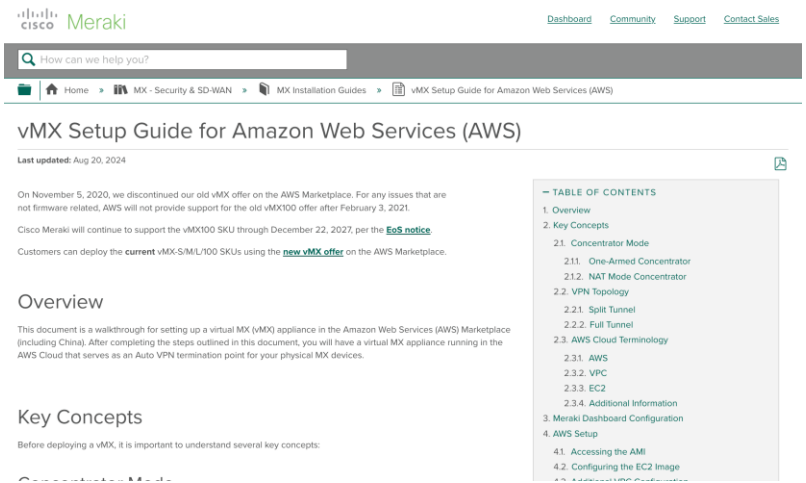
© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Example 2: Deploy vMX in AWS

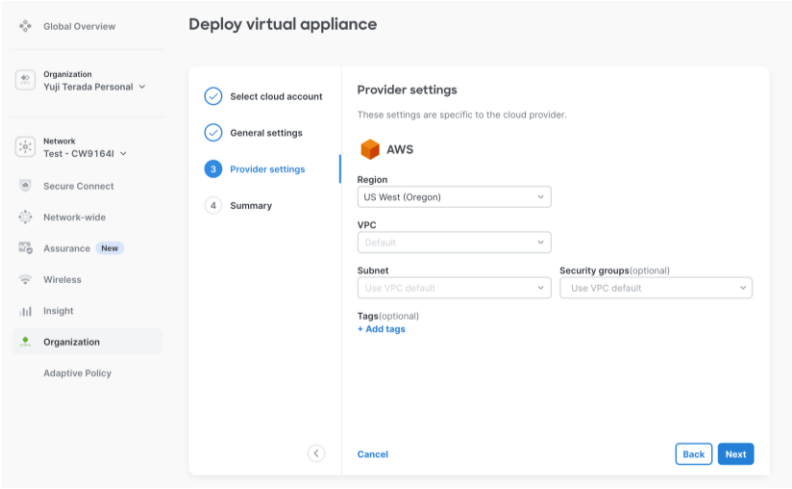


How do you deploy vMX?

Read document and run steps



Use Cloud Integration



These options require manual configuration on both dashboards

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Steps to deploy vMX in AWS

Meraki Dashboard

1. Create network
2. Add vMX to network
3. Generate auth token
4. Configure mode as passthrough
5. Configure type as hub
6. Configure local networks

AWS Management Console

1. Create VPC in AWS
2. Create private and public subnets in AWS
3. Create internet gateway
4. Create NAT gateway
5. Configure route table for each
6. Deploy vMX EC2 instance
7. Configure security group for vMX
8. Disable source destination check

https://documentation.meraki.com/MX/MX_Installation_Guides/vMX_Setup_Guide_for_Amazon_Web_Services_(AWS)
https://registry.terraform.io/providers/hashicorp/aws/latest/docs

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Steps to deploy vMX in AWS

Meraki Dashboard

1. Create network
2. Add vMX to network
3. Generate auth token
4. Configure mode as pa
5. Configure type as hub
6. Configure local netwo

AWS Management Console

AWS

and public subnets in AWS

t gateway

ateway

te table for each

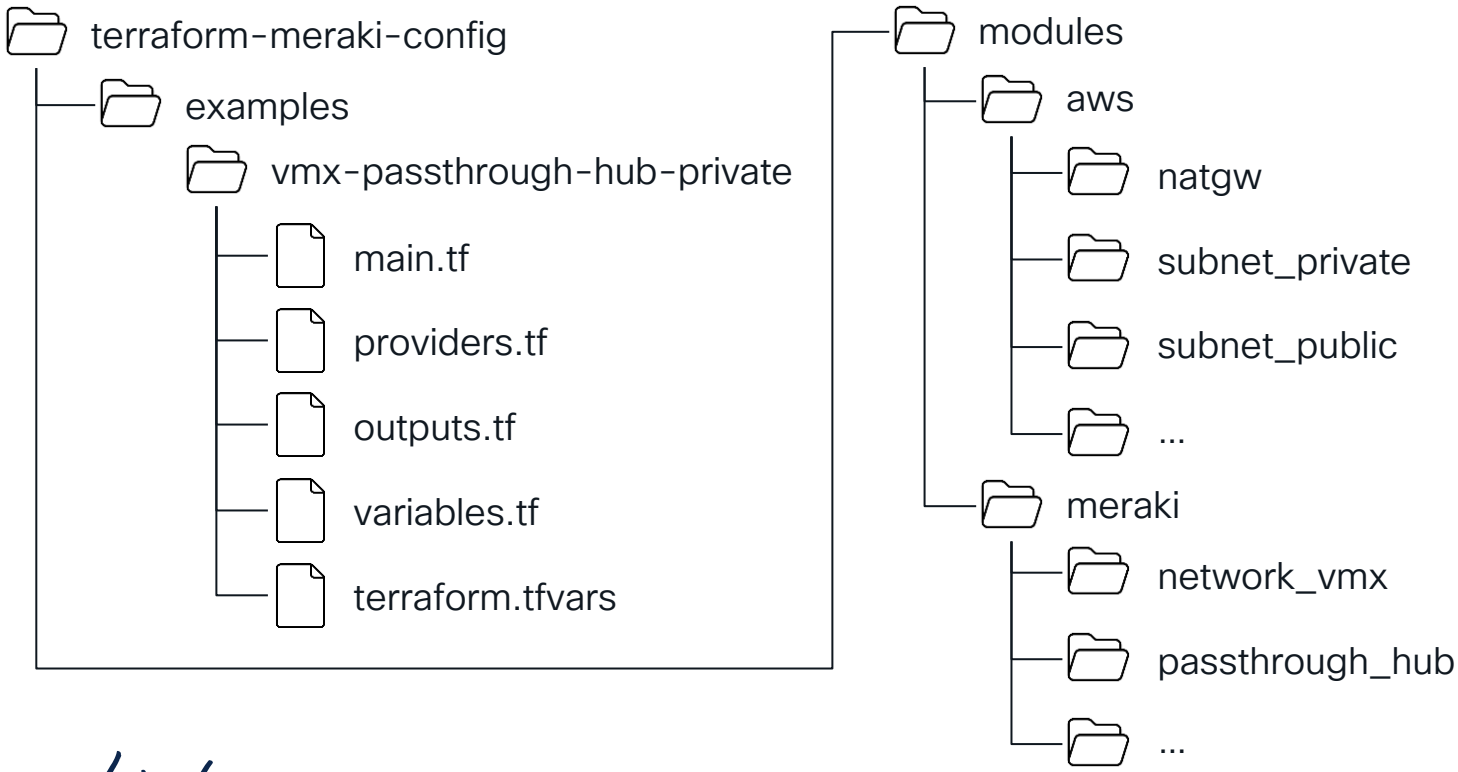
C2 instance

7. Configure security group for vMX
8. Disable source destination check

https://documentation.meraki.com/MX/MX_Installation_Guides/vMX_Setup_Guide_for_Amazon_Web_Services_(AWS)

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Folder structure



© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Summary

Z4C demo

- Dashboard logic dictates script behavior

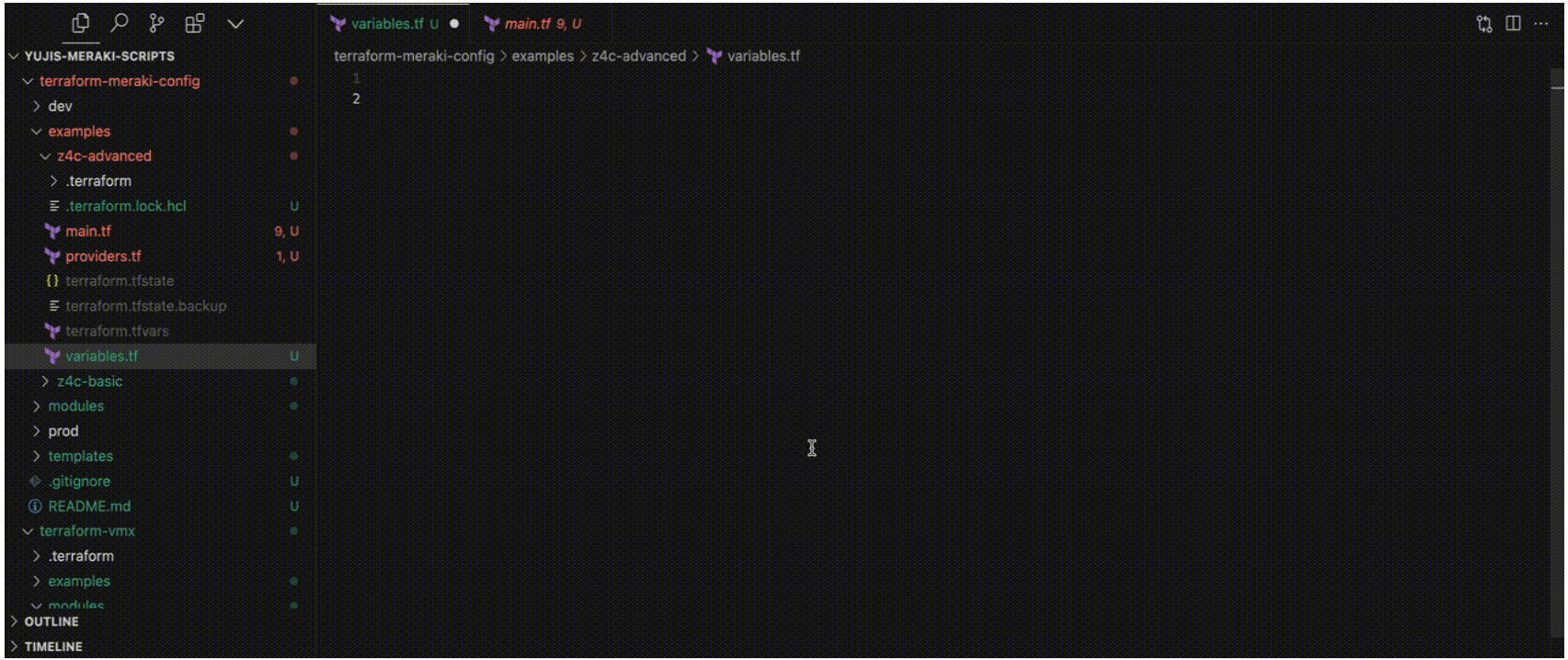
vMX demo

- Dashboard API availability equals script ability
- Terraform supports Meraki and other platforms

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Cursor in action



© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

A junior dev is going to come along and be like "Why are you not just using Cursor and making this responsive? Why are you wasting so much time?"

https://youtu.be/3wsVLG-dxg8?si=F_naK_MZJiBXyNCn&t=473

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Complete Your Session Evaluations

Complete a minimum of 4 session surveys and the Overall Event Survey to claim a **Cisco Live T-Shirt**.

Complete your surveys in the **Cisco Live mobile app**.

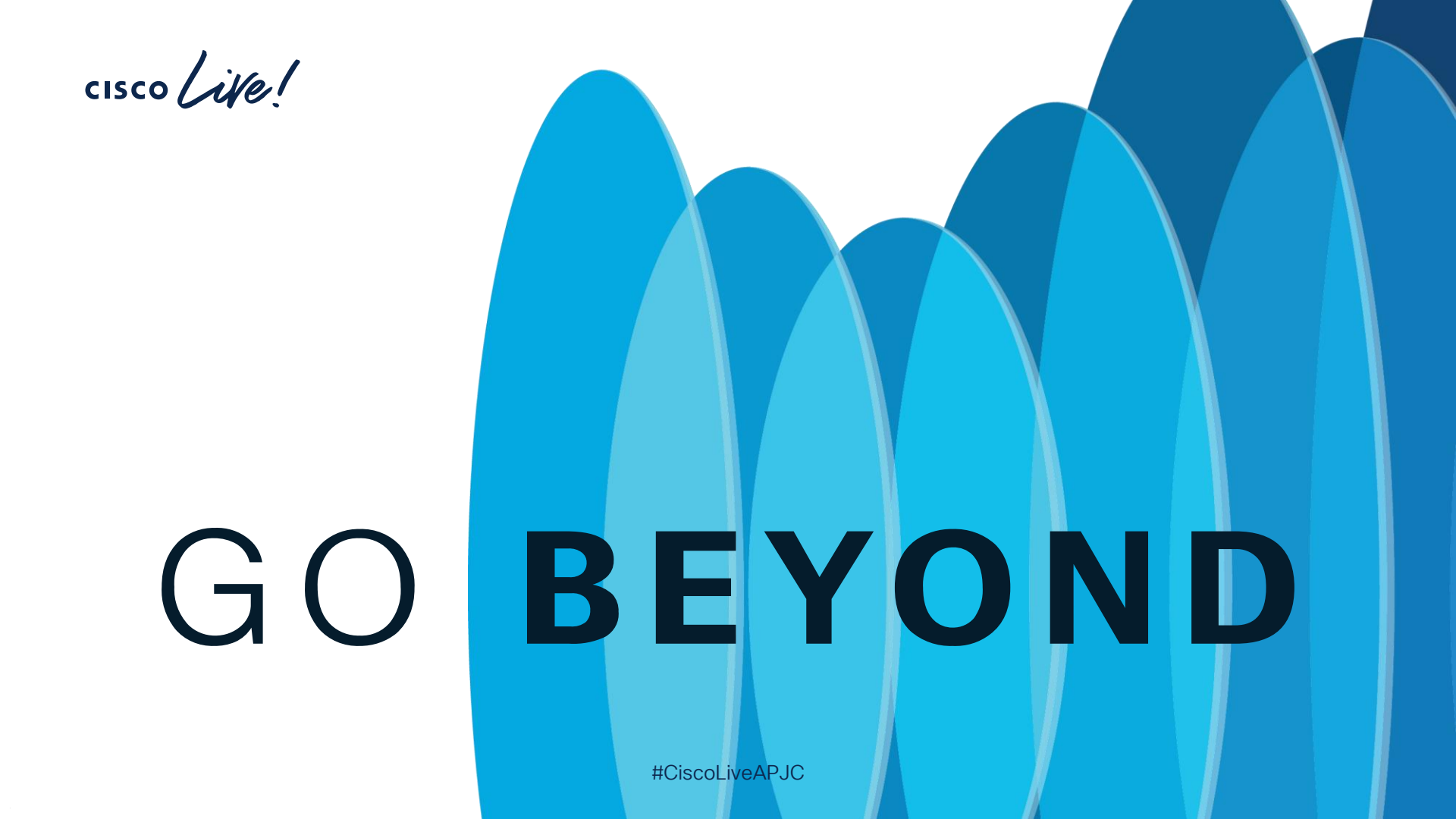
© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

Continue your education

- Visit the Cisco Stand for related demos
- Book your one-on-one Meet the Expert 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

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

cisco
Thank you
cisco Live!
#CiscoLiveAPJC


cisco Live!
GO BEYOND
#CiscoLiveAPJC