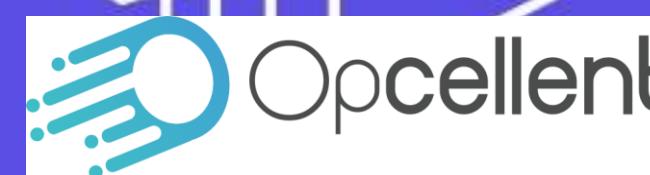
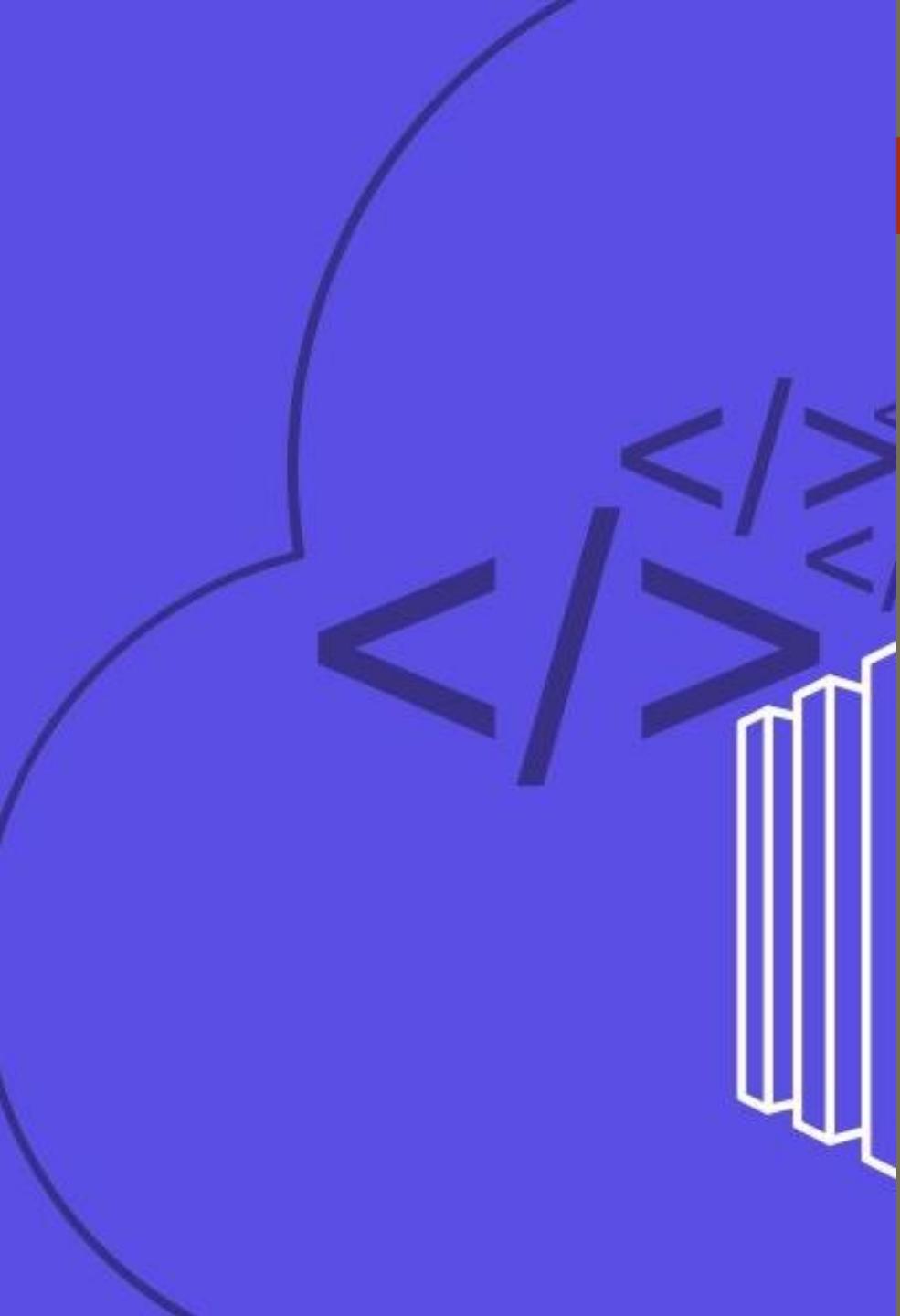




Infrastructure as Code(IaC) with Terraform (Day1)

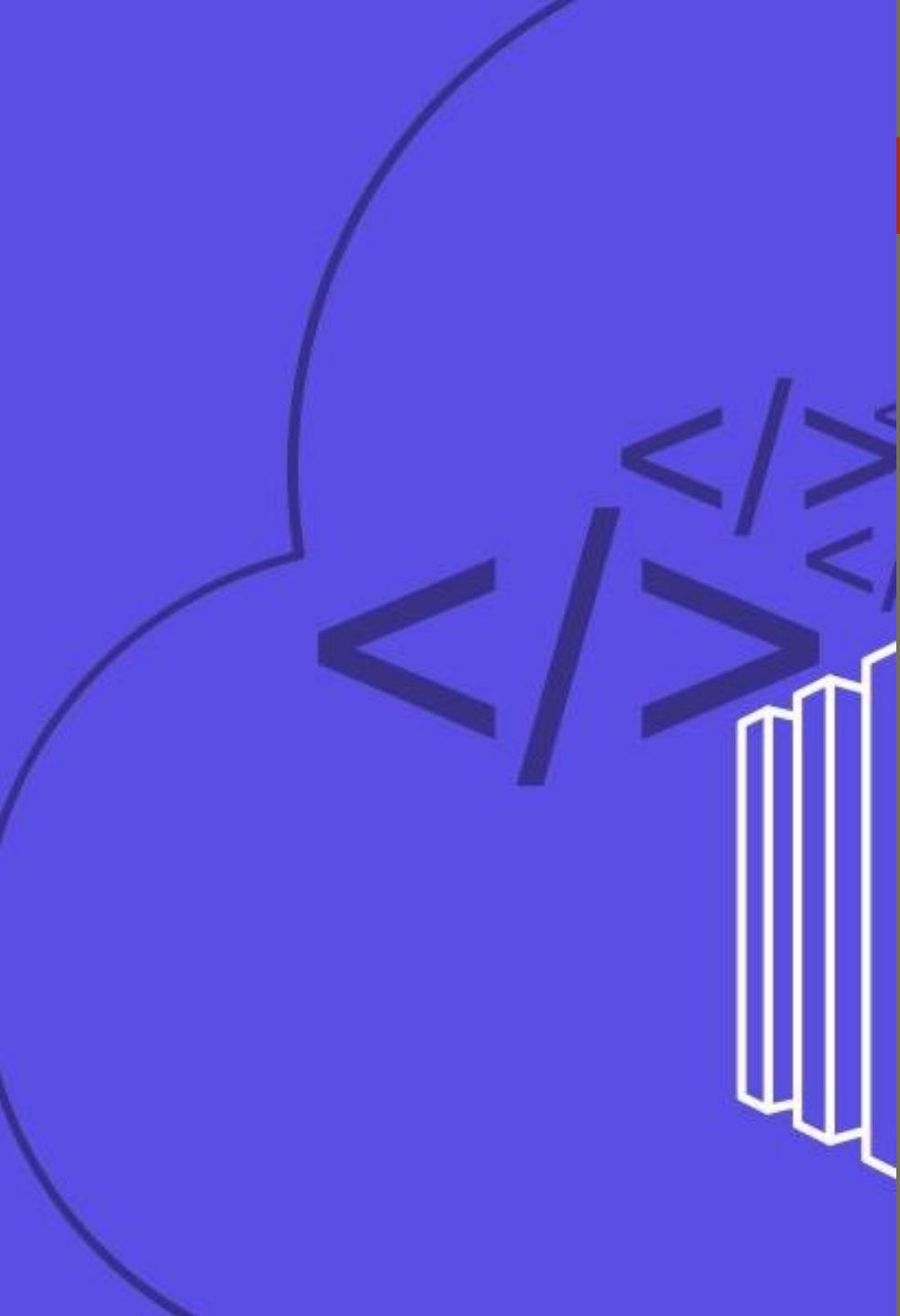
Praparn Luangphoonlap





Agenda (Day 1)

- ▶ Principle of IaC
 - ▶ Challenge to IT infrastructure today
 - ▶ Multiple cloud infrastructure
 - ▶ IaC well-known on the market
 - ▶ Orchestration vs Configuration Management
 - ▶ Feature and use-case on terraform
 - ▶ Terraform product
- ▶ Dive into HCL (Hashicorp Configuration Language) for terraform
 - ▶ Naming, Argument, Blocks and Expressions
 - ▶ Example file structure
 - ▶ System architecture
 - ▶ Terraform state management
 - ▶ Resource and dependency
 - ▶ Provisioner



Agenda (Day 2)

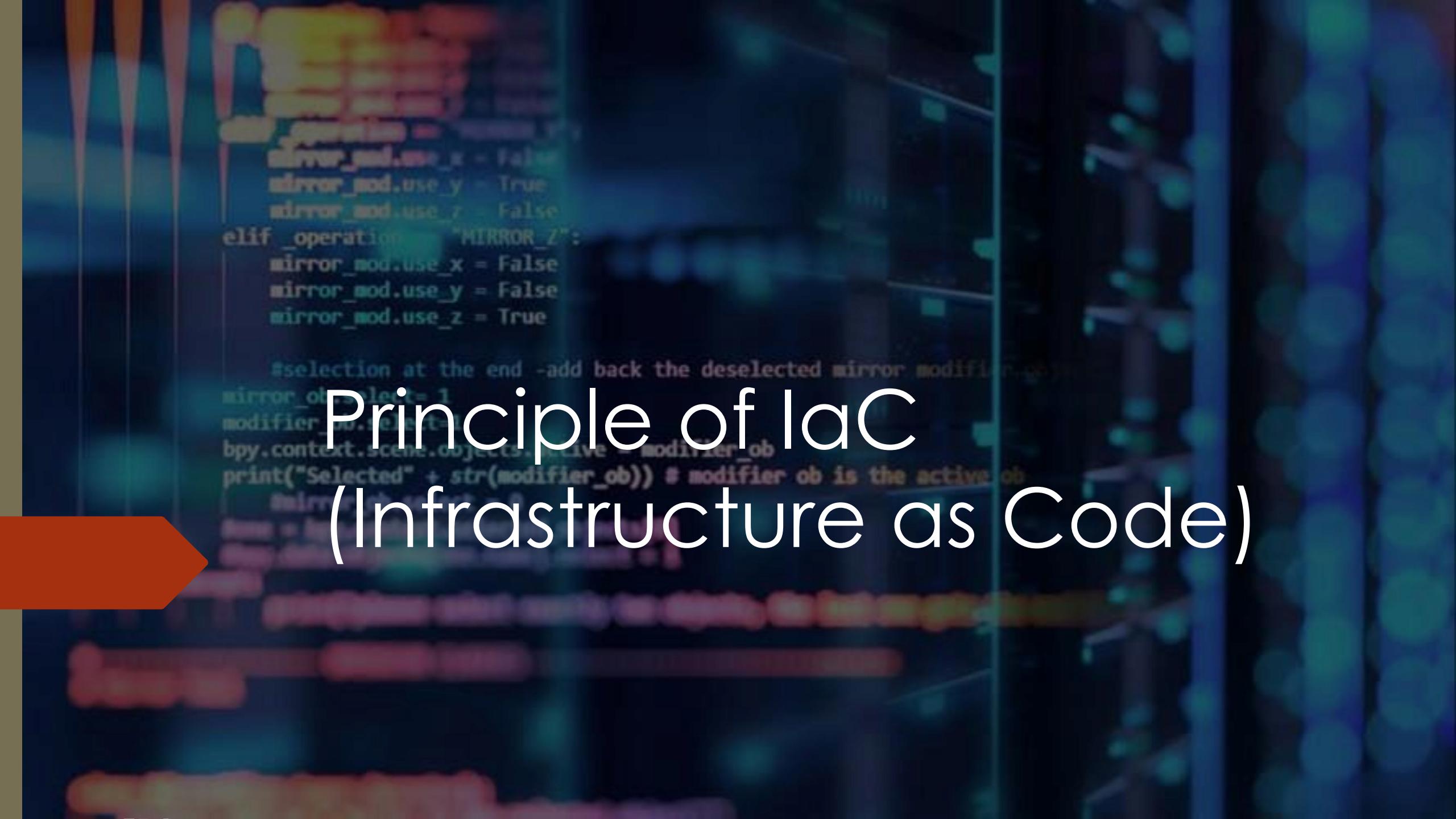
- ▶ Dive to HCL (Hashicorp Configuration Language) for terraform (Cont.)
 - ▶ Variable (Input & Output)
 - ▶ Registry and Module
- ▶ Terraform cloud for team collaboration
 - ▶ Introduction to terraform cloud
 - ▶ Setup workspace and test drive
- ▶ Team workshop (Final Workshop)
- ▶ Q&A

Class Description

- ▶ Class duration 2 days
- ▶ Course period: 08.30 – 18.30
- ▶ Lunch break: 12.00 – 13.00
- ▶ Student need to BYOD for workshop
- ▶ 30% Theory / 70% Workshop , Discussion
- ▶ Cloud resource was provided for workshop
- ▶ Electricity and Internet service is base facility required

Prerequisite for Workshop

- ▶ Notebook for join workshop (Windows 10, MAC OS X, Linux etc) with wireless card
- ▶ Basic understand for linux and vim
- ▶ Basic understand cloud concept and design practice
- ▶ Email Address and Account on GitHub (1 Set per Group)
- ▶ Basic git knowledge for operate registry
- ▶ IDE as your favorite (VSCode, Atom, PHPStrom etc)
- ▶ Email address for register with terraform cloud (1 Set per Group)
- ▶ Internet access for operate workshop



Principle of IaC (Infrastructure as Code)

```
    #operations = "MIRROR_X"
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
elif _operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

#selection at the end -add back the deselected mirror modifier
mirror_obj.select= 1
modifier_obj.select=1
bpy.context.scene.objects.active = modifier_obj
print("Selected" + str(modifier_obj)) # modifier ob is the active ob
#mirror_mod.select=1
#bpy.context.scene.objects.active = mirror_mod
```



Challenge to IT infrastructure today

- ▶ DevOps are change business's vision for IT infrastructure
- ▶ IT infrastructure will dramatically change every in sprint
- ▶ Developer need “Infrastructure as Service” for create infrastructure as request
- ▶ Operation wish to define standard infrastructure for every cloud/on-prem
- ▶ Infrastructure's agility was need with lower workforce for IT MNG



Multiple cloud infrastructure

- ▶ Multiple cloud is strategic on year 2020
- ▶ Business user a lot from multi cloud such as.
 - ▶ EKS, S3 on AWS
 - ▶ Active Directory on Azure
 - ▶ Firebase DB on Gcloud
 - ▶ Cloudflare for CDN
 - ▶ Docker on Private Cloud
- ▶ More cloud to reach. More skill to need



Multiple cloud infrastructure

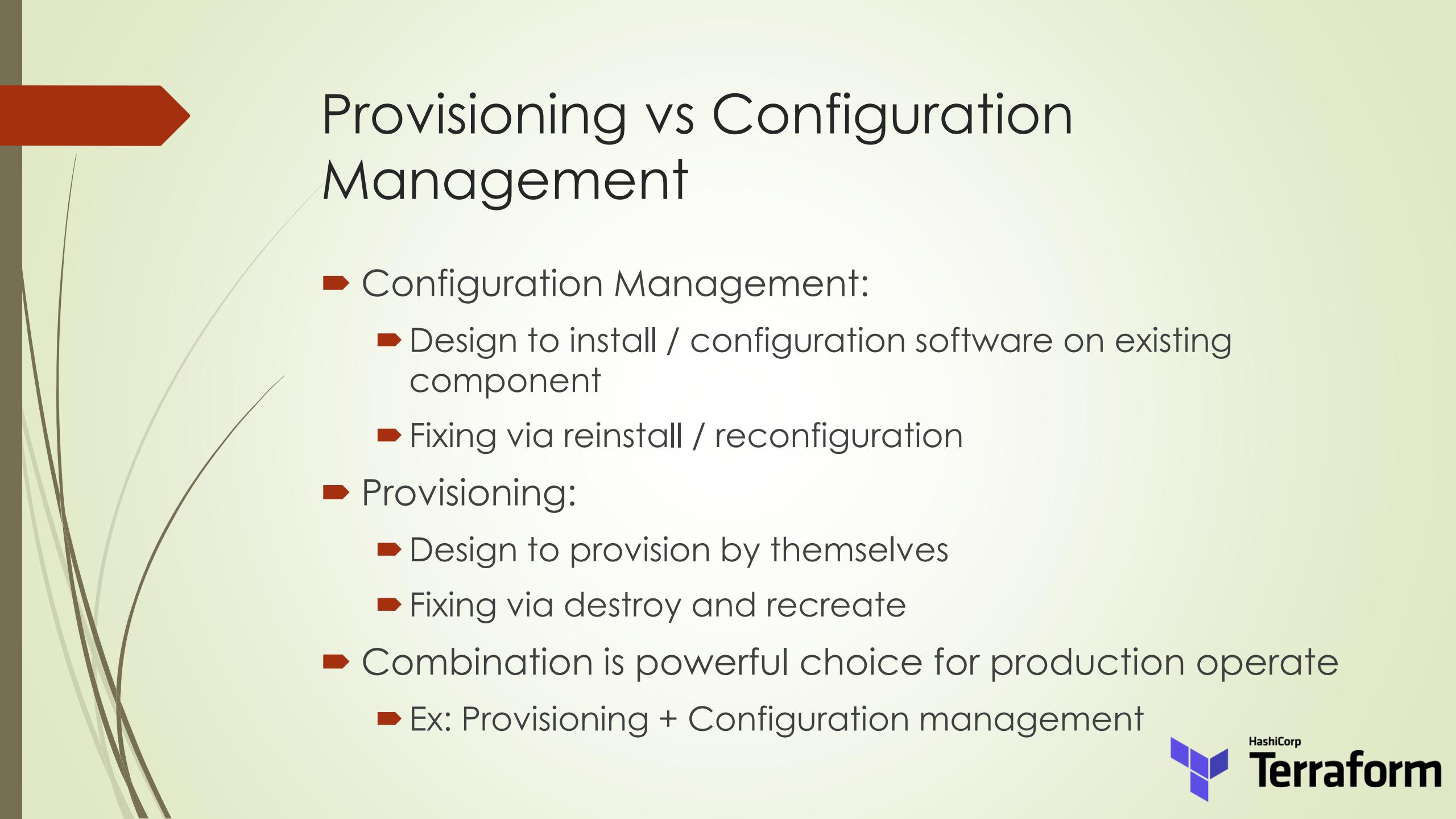
IAC well-known on market



	Source	Cloud	Type	Infrastructure	Language	Agent	Master	Community	Maturity
Chef	Open	All	Config Mgmt	Mutable	Procedural	Yes	Yes	Large	High
Puppet	Open	All	Config Mgmt	Mutable	Declarative	Yes	Yes	Large	High
Ansible	Open	All	Config Mgmt	Mutable	Procedural	No	No	Huge	Medium
SaltStack	Open	All	Config Mgmt	Mutable	Declarative	Yes	Yes	Large	Medium
CloudFormation	Closed	AWS	Provisioning	Immutable	Declarative	No	No	Small	Medium
Heat	Open	All	Provisioning	Immutable	Declarative	No	No	Small	Low
Terraform	Open	All	Provisioning	Immutable	Declarative	No	No	Huge	Low

Ref: <https://blog.gruntwork.io/why-we-use-terraform-and-not-chef-puppet-ansible-saltstack-or-cloudformation-7989dad2865c>

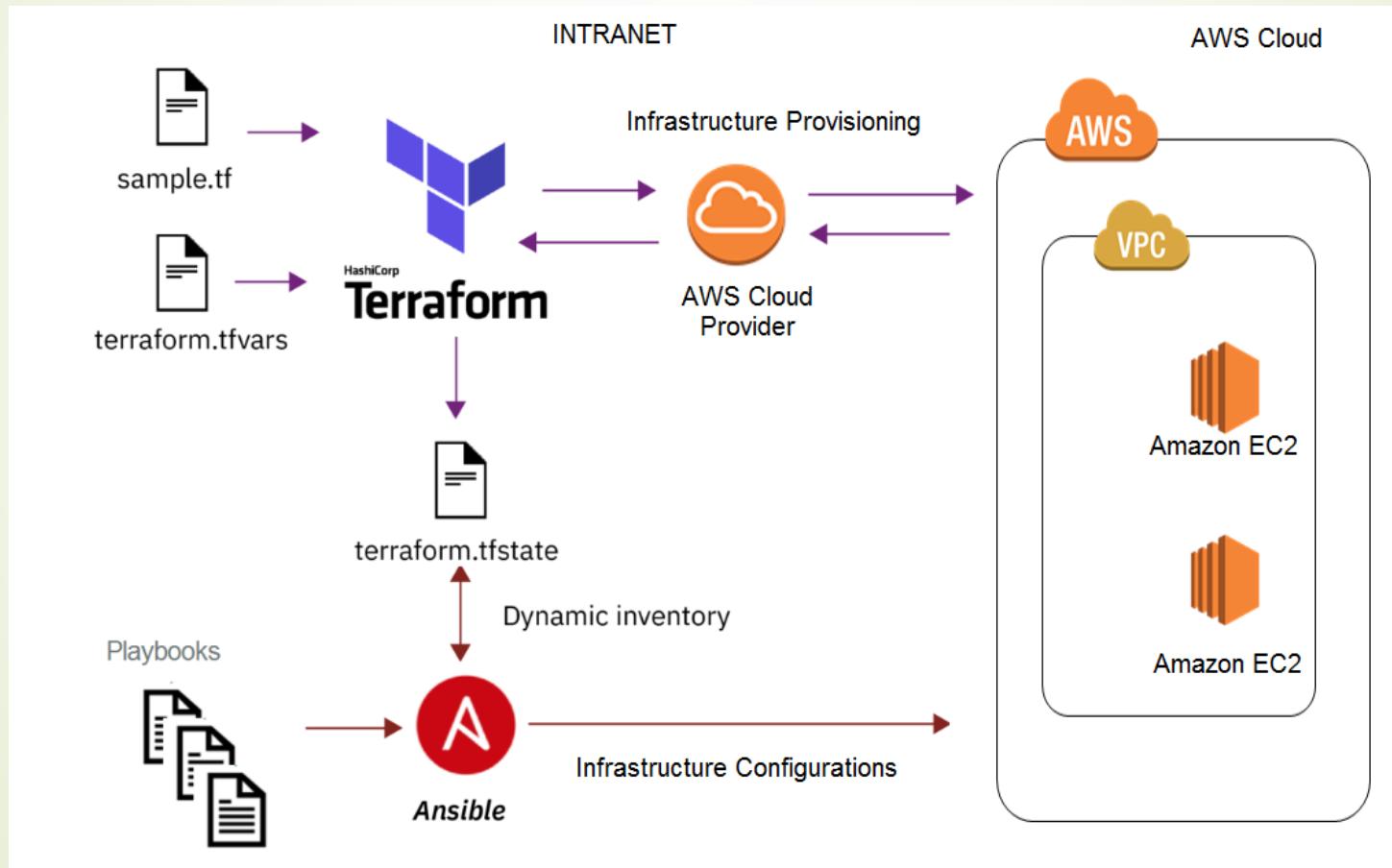




Provisioning vs Configuration Management

- ▶ Configuration Management:
 - ▶ Design to install / configuration software on existing component
 - ▶ Fixing via reinstall / reconfiguration
- ▶ Provisioning:
 - ▶ Design to provision by themselves
 - ▶ Fixing via destroy and recreate
- ▶ Combination is powerful choice for production operate
 - ▶ Ex: Provisioning + Configuration management

Provisioning vs Configuration Management



Feature and use-case terraform

- ▶ **Terraform** is an open-source infrastructure as code software tool created by HashiCorp. It enables users to define and provision a datacenter infrastructure using a high-level configuration language known as Hashicorp Configuration Language (HCL), or optionally JSON.[3]



Feature and use-case terraform

- ▶ IaC with terraform script support more than 121+ platform (Official) and more on community
- ▶ Ex: Compute Provider
 - ▶ AWS
 - ▶ Alibaba
 - ▶ Azure
 - ▶ Digital Ocean
 - ▶ Openstack
 - ▶ vSphere
 - ▶ etc

Feature and use-case terraform

Providers

Terraform is used to create, manage, and update infrastructure resources such as physical machines, VMs, network switches, containers, and more. Almost any infrastructure type can be represented as a resource in Terraform.

A provider is responsible for understanding API interactions and exposing resources. Providers generally are an IaaS (e.g. Alibaba Cloud, AWS, GCP, Microsoft Azure, OpenStack), PaaS (e.g. Heroku), or SaaS services (e.g. Terraform Cloud, DNSimple, Cloudflare).

Use the navigation to the left to find available providers by type or scroll down to see all providers.

- ACME
- Akamai
- Alibaba Cloud
- Archive
- Arukas
- Avi Vantage
- Aviatrix
- AWS
- Azure
- Azure Active Directory
- Azure Stack
- A10 Networks
- Bitbucket
- GitLab
- Google Cloud Platform
- Grafana
- Gridscale
- Hedvig
- Helm
- Heroku
- Hetzner Cloud
- HTTP
- HuaweiCloud
- HuaweiCloudStack
- Icinga2
- Ignition
- Packet
- PagerDuty
- Palo Alto Networks
- PostgreSQL
- PowerDNS
- ProfitBricks
- Pureport
- RabbitMQ
- Rancher
- Rancher2
- Random
- RightScale
- Rundeck

- | | | | | | |
|--------------------|-----------------|--------------------------|------------------|-------------------------------|------------------|
| • Brightbox | • InfluxDB | • RunScope | • F5 BIG-IP | • OpenTelekomCloud | • VMware vRA7 |
| • CenturyLinkCloud | • JDCloud | • Scaleway | • Fastly | • OpsGenie | • VMware vSphere |
| • Chef | • Kubernetes | • Selectel | • FlexibleEngine | • Oracle Cloud Infrastructure | • Vultr |
| • CherryServers | • LaunchDarkly | • SignalFx | • FortiOS | • Oracle Cloud Platform | • Yandex |
| • Circonus | • Librato | • Skytap | • Genymotion | • Oracle Public Cloud | |
| • Cisco ASA | • Linode | • SoftLayer | • GitHub | • OVH | |
| • Cisco ACI | • Local | • Spotinst | | | |
| • Cloudflare | • Logentries | • StackPath | | | |
| • CloudScale.ch | • LogicMonitor | • StatusCake | | | |
| • CloudStack | • Mailgun | • TelefonicaOpenCloud | | | |
| • Cobbler | • MongoDB Atlas | • Template | | | |
| • Consul | • MySQL | • TencentCloud | | | |
| • Datadog | • Naver Cloud | • Terraform | | | |
| • DigitalOcean | • Netlify | • Terraform Cloud | | | |
| • DNS | • New Relic | • TLS | | | |
| • DNSimple | • Nomad | • Triton | | | |
| • DNSMadeEasy | • NS1 | • UCloud | | | |
| • Docker | • Null | • UltraDNS | | | |
| • Dome9 | • Nutanix | • Vault | | | |
| • Dyn | • 1&1 | • Venafi | | | |
| • Exoscale | • OpenNebula | • VMware NSX-T | | | |
| • External | • OpenStack | • VMware vCloud Director | | | |

Feature and use-case terraform

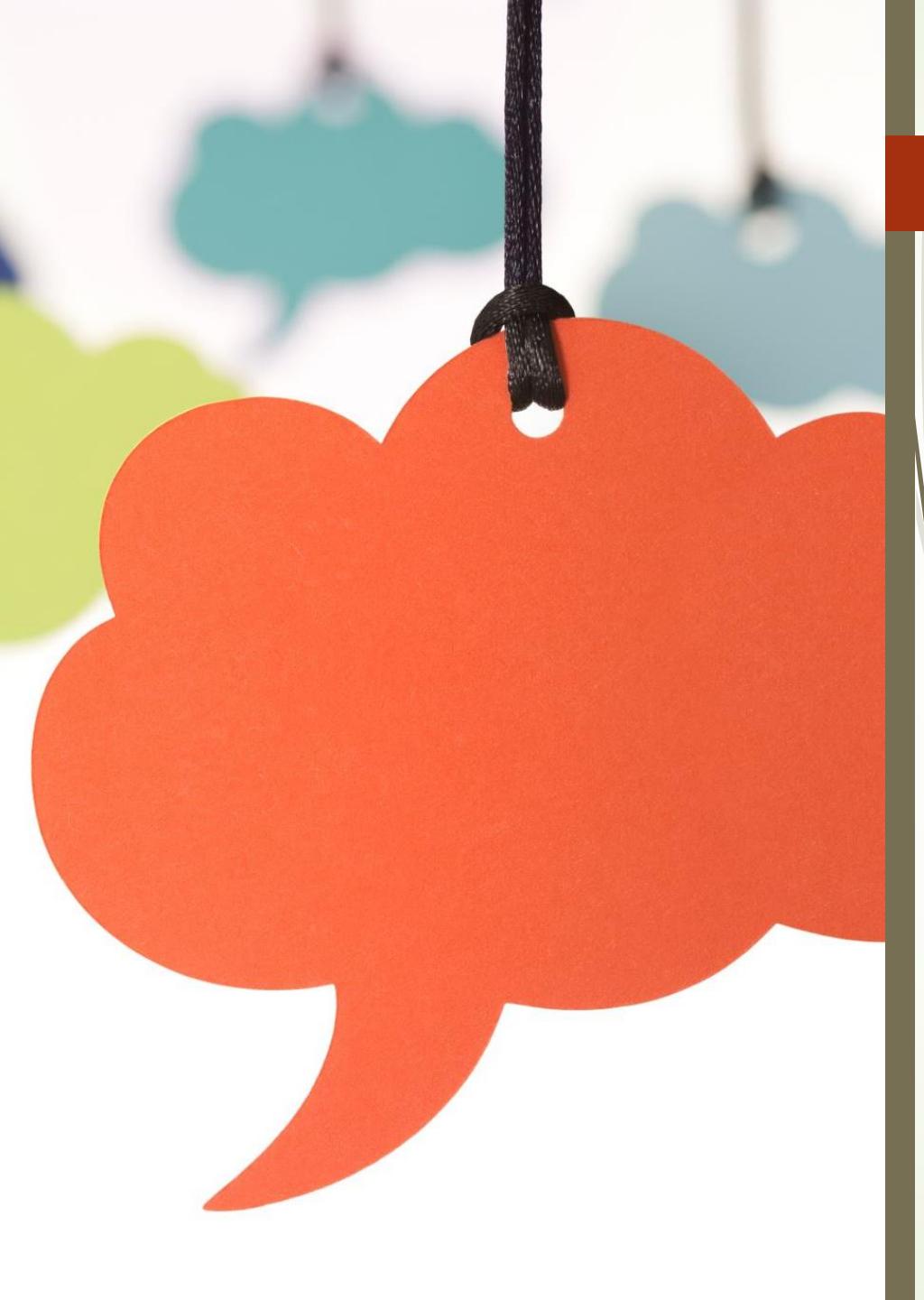
Community Providers

The providers listed below have been built by the community of Terraform users and vendors. These providers are not tested nor officially maintained by HashiCorp, and are listed here in order to help users find them easily.

If you have built a provider and would like to add it to this community list, please fill out this [community providers form](#).

- | | | |
|--|--------------------------------------|--|
| • 1Password | • Foreman | • OpenCloud |
| • Abiquo | • Gandi | • oVirt |
| • Active Directory - adlerrobert | • Generic Rest API | • Pass |
| • Active Directory - GSLabDev | • Git | • PHPIPAM |
| • Aiven | • GitHub Code Owners | • Pingdom |
| • AlienVault | • GitHub File | • Pivotal Tracker |
| • AnsibleVault | • GitInfo | • Proxmox |
| • Apigee | • Glue | • Puppet CA |
| • Artifactory | • GoCD | • PuppetDB |
| • Auth | • Google Calendar | • Purestorage Flasharray |
| • Auth0 | • Google G Suite | • QingCloud |
| • Automic Continuous Delivery | • GorillaStack | • Qiniu |
| • AVI | • Hiera | • Redshift |
| • Aviatrix | • HPE OneView | • RKE |
| • AWX | • HTTP File Upload | • Rollbar |
| • Azure Devops | • IBM Cloud | • SakuraCloud |
| • Bitbucket Server | • IIJ GIO | • SCVMM |
| • Centreon | • Infoblox | • Sendgrid |

- | | | | | | |
|--|-----------------------------------|-------------------------------|--|----------------------------------|---------------------------------------|
| • Checkly | • InsightOPS | • Sensu | • Drone | • Name | • vRealize Automation |
| • Cherry Servers | • Jira | • Sentry | • Dropbox | • Nelson | • Vultr |
| • Citrix ADC | • Jira (Extended) | • Sewan | • Duo Security | • NetApp | • Wavefront |
| • Cloud Foundry | • JumpCloud | • Shell | • EfficientIP | • NSX-V | • Win DNS |
| • CloudAMQP | • Kafka | • Smartronix | • Elastic Cloud Enterprise (ECE) | • Okta | • XML |
| • CloudKarafka | • Kafka Connect | • Snowflake | • Elasticsearch | • Online.net | • YAML |
| • CloudMQTT | • Keboola | • snowflakedb | • ElephantSQL | • Open Day Light | • Zendesk |
| • CloudPassage Halo | • Keycloak | • sops | • Enterprise Cloud | • OpenAPI | • ZeroTier |
| • CodeClimate | • Keyring | • Spinnaker | • ESXI | • OpenFaaS | • Zipper |
| • Confidant | • Kibana | • SQL | | | |
| • Confluent Cloud | • Kong | • Stateful | | | |
| • Consul ACL | • Ksyun | • Statuspage | | | |
| • CoreOS Container Linux Configs | • Kubectl | • Stripe | | | |
| • Coveo Cloud | • Kubernetes | • Sumo Logic | | | |
| • CouchDB | • libvirt | • TeamCity | | | |
| • Credhub | • Logentries | • Transloadit | | | |
| • Cronitor | • Logz.io | • Trello | | | |
| • Databricks | • LXD | • tumblr | | | |
| • Dead Man's Snitch | • Manifold | • Updown.io | | | |
| • Digital Rebar | • Matchbox | • Uptimerobot | | | |
| • Docker Machine | • MongoDB Atlas | • Vaulted | | | |
| | • Nagios XI | • Venafi | | | |



Feature and use-case terraform

- ▶ Terraform has built-in workflows for deployment step
 - ▶ **Collection** all components needed to create on project
 - ▶ **Develop** HCL script base on support provider
 - ▶ **Initialize** project for prepare config and relate library to project with "terraform init"
 - ▶ **Verify** HCL script with "terraform plan"
 - ▶ **Run** HCL script with "terraform apply"



Feature and use-case terraform

- ▶ Among many IaC script on market. Terraform was aimed for support a lot of advantage below
 - ▶ **Platform Agnostic**
 - ▶ **State Management** (source of truth)
 - ▶ **Operate Confidence**

Terraform Product

FOR INDIVIDUALS

Open Source

Infrastructure as code provisioning and management for any infrastructure.

Free

 Download

FOR TEAMS

Terraform Cloud

Collaboration and automation for practitioners and small teams using Terraform.

[Get Started](#)

FOR ORGANIZATIONS

Terraform Enterprise

Private installation of Terraform with collaboration, policy & governance, and self-service infrastructure for organizations.

[Contact Sales](#)

Terraform product

Free	Team	Team & Governance	Self-Hosted
\$0 Up to 5 Users	\$20 User / Month	\$70 User / Month	Contact Us Enterprise Sales
UNLIMITED WORKSPACES EMAIL SUPPORT	UNLIMITED WORKSPACES EMAIL SUPPORT	UNLIMITED WORKSPACES EMAIL SUPPORT	100+ WORKSPACES SLA SUPPORT
✓ VCS Integration ⓘ			
✓ Workspace Management ⓘ			
✓ Secure Variable Storage ⓘ			
✓ Remote Runs & Applies ⓘ			
✓ Full API Coverage ⓘ			
✓ Private Module Registry ⓘ			
Roles / Team Management ⓘ	✓ Roles / Team Management ⓘ	✓ Roles / Team Management ⓘ	✓ Roles / Team Management ⓘ
Sentinel ⓘ	Sentinel ⓘ	✓ Sentinel ⓘ	✓ Sentinel ⓘ
Cost Estimation ⓘ	Cost Estimation ⓘ	✓ Cost Estimation ⓘ	✓ Cost Estimation ⓘ
SAML / SSO ⓘ	SAML / SSO ⓘ	SAML / SSO ⓘ	✓ SAML / SSO ⓘ
Clustering ⓘ	Clustering ⓘ	Clustering ⓘ	✓ Clustering ⓘ
Private DC Installation ⓘ	Private DC Installation ⓘ	Private DC Installation ⓘ	✓ Private DC Installation ⓘ
Private Network Connectivity ⓘ	Private Network Connectivity ⓘ	Private Network Connectivity ⓘ	✓ Private Network Connectivity ⓘ
Self-Hosted ⓘ	Self-Hosted ⓘ	Self-Hosted ⓘ	✓ Self-Hosted ⓘ
Audit Logs ⓘ	Audit Logs ⓘ	Audit Logs ⓘ	✓ Audit Logs ⓘ



Dive to HCL for Terraform



Dive to HCL for Terraform

- ▶ HCL is stand for "HashiCorp configuration language"
- ▶ HCL is not “YAML” or “JSON” but full compatible with JSON also
- ▶ HCL are balance between human-friendly language and machine-friendly language
- ▶ HCL is heavily inspired by libucl (universal control language), nginx configuration, and others similar.
- ▶ HCL was growth 213% (Base on github information)

Ref: <https://www.businessinsider.com/fastest-growing-programming-languages-github-2019-11>

Ref: <https://github.com/hashicorp/hcl>



Dive to HCL for Terraform



Vsevolod Stakhov

vstakhov

Follow

📍 Cambridge, UK

✉️ vsevolod@highsecure.ru

Block or report user

Organizations



Fatih Arslan

fatih

Follow

Software Engineer. Gopher and Coffee geek. Creator of vim-go. Tool maker.

👤 GitHub STAFF

📍 California, USA

🔗 <https://arslan.io>

Block or report user

Organizations



Dive to HCL for Terraform

- ▶ Single line comment start with “#”, “//”
- ▶ Multiple line command start with “/*” and end with “*/”
- ▶ Value are assigned by syntax: key = value
- ▶ HCL operate with “UTF-8” format
- ▶ Variable type can be: String, Number, Boolean, Object, List
- ▶ String: “XXXXXX”
- ▶ Multiple Line String: <<EOF

YYYYYYYY

XXXXXXX

EOF

Dive to HCL for Terraform

- ▶ Number: 999 (Base 10)

- ▶ Boolean: true, false

- ▶ Array:

- ▶ ["xxx", "yyy", "999"]

- ▶ List:

```
service {
```

```
    key = value
```

```
}
```

Dive to HCL for Terraform

► Nest Object:

```
provider "aws" {  
    access_key = "${var.access_key}"  
}  
provider "aws" {  
    access_key = var.access_key  
}
```

➔ (Obsolete for terraform 0.11)

➔ terraform 0.12

► JSON

```
{  
    "provider": {  
        "aws": {  
            "access_key": "${var.access_key}"  
        }  
    }  
}
```

Naming, Arguments, Block and Expression

Naming Variable

- Allow character, digit, underscores (_), hyphen (-)
- First character cannot be “digit”

Arguments

- <argument's name> = <value>
- vpc_name = “MyVPC”
- region_zone = ["us-west-1a", "us-west-1c"]
- delete_flag = true

Naming, Arguments, Block and Expression

Block

- Block have type (variable, resource, provider, module, output)
- Delimited by “{“ and “}”
- Contain multiple expression on same block

Example

```
• variable "terraform_var02" {  
•   description = "Complex variable for operate"  
•   type = "map"  
•   default = {  
•     "0"      = "Zero to Hero"  
•     "1"      = "Once Upon a Time"  
•     "2"      = "Two Town"  
•   }  
• }
```

Naming, Arguments, Block and Expression

Expression (Type)

- Type
 - string: “xxxxxx”
 - number: 12, 6.3922
 - bool: true, false
 - list: [“name1”, 23, false]
 - map: {name1=“x”, name2=342}
 - null

Expression (Convert)

- Terraform try to automatic convert type when expect type required
 - true (boolean) \leftrightarrow “true”
 - false (boolean) \leftrightarrow “false”
 - 234 (number) \leftrightarrow “234”

Naming, Arguments, Block and Expression

Expression (Reference)

- aws_instance.eks1.ami → ami-ss
- aws_instance.eks1.id → xxxx
- aws_instance.eks1.ebs_block_device[0].device_name → sda2

Example

```
• Resource "aws_instance" "eks1"{
  •   ami="ami-ss"
  •   instance_type="t2.micro"
  •   ebs_block_device {
    •     device_name="sda2"
    •   }
  •   ebs_block_device {
    •     device_name="sda3"
    •   }
}
```

Naming, Arguments, Block and Expression

Arithmetic Operator

- A + B
- A - B
- A * B
- A / B
- A % B
- -A

Order of Operation

- ! , - (multiplication by -1)
- *, /, %
- +, - (subtraction)
- >, >= , < , <=
- ==, !=
- &&
- ||

Naming, Arguments, Block and Expression

Build-In Function

- Numeric Function
- String Function
- Collection Function
- Encoding Function
- Filesystem Function
- Date and Time Function
- Hash and Crypto Function
- IP Network Function
- Type Conversion Function

Example

- `abs(-982.20)` → 982.20
- `trim ("##VAV##","##")` → VAV
- `contains(["#", "w"], "w")` → true
- `base64encode("secret")` → c2VjcmV0
- `pathexpand("~/")` → /home/ubuntu
- `formatdate("YYYYMMDD",timestamp())` → 20200101
- `md5("secret")` → 5ebe2294ecd0e0f08eab7690d2a6ee69
- `cidrnetmask("10.38.23.0/23")` → 255.255.254.0
- `tobool("false")` → false
- `tobool("terraform")` → error

Naming, Arguments, Block and Expression

Build-In Function

- Numeric Function
- String Function
- Collection Function
- Encoding Function
- Filesystem Function
- Date and Time Function
- Hash and Crypto Function
- IP Network Function
- Type Conversion Function

Example

- replace("this is terraform", "terraform", "kkk") → this is kkk
- title("WIKIPedia") → Wikipedia
- split("@","@39@32@12@90") →
- [
- "39",
- "32",
- "12",
- "90"
-]
- signum("-3242343") → -1
- pow(10,2) → 100

Naming, Arguments, Block and Expression

For Expression

- Loop for operate on variable
- For will depend on type of variable
- Use-case for decision/modified value before next operate

Example

- variable "nu" {
 - type = "list"
 - default = ["n1", "n2", "n3"]
 - }
- [for s in var.nu : upper(s) if s != "n1"] → N2 N3

Naming, Arguments, Block and Expression

String Literal

- \n → New Line
- \r → Carrier Return
- \t → Tab
- \" → "
- \\ → \
- \uNNNN → Unicode hex 4 digits
- \uNNNNNNNN → Unicode hex 8 digits

String Template

- Interpolation:
- variable "ec" {default = "docker-1"}
- "Instance: *\${var.ec} *" → Instance : *docker-1*
- Directive:
- "Instance: %{if var.ec == ""}Non %{ else } var.ec %{ endif }

Workshop: HCL practice



Workshop HCL practice

► Question:

► `replace(lower("\nHELLO, TERRAFORM. \nNo HELLO WORLD
ANYMORE!!!\n"), "terraform", upper("<your name>"))`

► Ans:

► "

► hello, <your name>.

► no hello world anymore !!!

► "

Workshop HCL practice

► Question:

```
► split("-", title("Provider-azure-gcloud-aws-digitalocen-huiweicloud-rundesk"))
```

► Ans:

```
► [  
►   "Provider",  
►   "Azure",  
►   "Gcloud",  
►   "Aws",  
►   "Digitalocen",  
►   "Huiweicloud",  
►   "Rundesk",  
► ]
```

Workshop HCL practice

► **Question:**

► `trimspace(lower(strrev("MROFARRET MORF DLROW OLLEH ")))`

► **Ans:**

► hello world from terraform

► **Question:**

► `ceil(parseint("1111",2)) + abs(max(3215,2, -234134))`

► **Ans:**

► 3230

► **Question:**

► `signum(min(23244,0,-2232,123)) + pow(3,2)`

► **Ans:**

► 8

Workshop HCL practice

► **Question:**

► $\text{abs}(8\%5+10*(-10))$

► **Ans:**

► 97

► **Question:**

► `cidrhost("10.21.0.0/16", 2048)`

► **Ans:**

► 10.21.8.0

► **Question:**

► `cidrnetmask("10.101.23.0/32")`

► **Ans:**

► 255.255.255.255

Workshop HCL practice

► **Question:**

- `abspath(path.root)`

► **Ans:**

- `/home/ubuntu`

► **Question:**

- `formatdate ("YYYY-MM-DD HH-mm-ss AA", "2020-01-10T23:01:15Z")`

► **Ans:**

- `2020-01-10 11-01-15 PM`

► **Question:**

- `formatdate ("YYYY-MM-DD HH-mm-ss AA", timeadd("2020-01-10T23:01:15Z", "12h"))`

► **Ans:**

- `2020-01-10 11-01-15 AM`

Workshop HCL Practice

```
ubuntu@ip-172-31-25-11:~$ terraform console
> parseint(1011111111,2)

>
Error: Invalid function argument

[ on <console-input> line 1:
  (source code not available)

Invalid value for "number" parameter: first argument must be a string, not
number.

> ceil(parseint("1011111111",2))
767
> ceil(parseint("1111",2)) + abs(max(3234,2, -234134))
3249
> ceil(parseint("1111",2)) + abs(max(3234,2, -234134))
[3249
>
[
>

> replace(lower("HELLO, TERRAFORM. No HELLO WORLD ANYMORE!!!\n"), "terraform", upper("<your name>"))
hello, <YOUR NAME>. no hello world anymore!!!

> replace(lower("HELLO, TERRAFORM. \n No HELLO WORLD ANYMORE!!!\n"), "terraform", upper("praparn"))
hello, PRAPARN.
  no hello world anymore!!!

> replace(lower("HELLO, TERRAFORM. \n No HELLO WORLD ANYMORE!!!\n"), "terraform", upper("praparn"))
hello, PRAPARN.
  no hello world anymore!!!
[
> ceil(parseint("1111",2)) + abs(max(3215,2, -234134))
[3230
> signum(min(23244,0,-2232,123)) + pow(3,2)
[8
> abs(5%8)
5
> 5%8
5
> 5/8
0.625
> 5 mod 8
```

Q&A



Terraform file structure

```
variable "xxxxx" {  
    key = value  
}  
  
provider "xxxxx" {  
    key = value  
}  
  
resource "xxxxx" "yyyyy" {  
    key = value  
}  
  
module "xxxxxx" {  
    key = value  
}  
  
data "xxxxxx" "yyyyy" {  
    key = value  
}  
  
output "xxxxxx" {  
    value = xxxx.yyyyy.id  
}
```

Example file structure

```
Users > praparnlueangphoonlap > Work > Terraform > Resource > 🗂 example.tf
1  variable "terraform_var01" {}
2  variable "terraform_var02" {
3    description = "Predefine variable for operate"
4    default = "DevOps was Here"
5  }
6  variable "terraform_var02" {
7    description = "Complex variable for operate"
8    type = "map"
9    default = {
10      "0"      = "Zero to Hero"
11      "1"      = "Once Upon a Time"
12      "2"      = "Two Town"
13    }
14  }
15  |
16  provider "google" {
17    zone = "us-central1"
18    project = "my-project-id"
19    region = "us-central1"
20  }
21
22  resource "google_compute_instance" "vm_instance" {
23    name        = "terraform-instance"
24    machine_type = "f1-micro"
25
26    boot_disk {
27      initialize_params {
28        | image = "debian-cloud/debian-9"
29      }
30    }
31  }
```

```
31  network_interface {
32    network = google_compute_network.vpc_network.name
33    access_config {
34    }
35  }
36  }
37
38  module "network" {
39    source  = "terraform-google-modules/network/google"
40    version = "1.1.0"
41    network_name = "terraform-vpc-network"
42    project_id   = var.project
43
44    subnets = [
45      {
46        subnet_name   = "subnet-01"
47        subnet_ip     = var.cidrs[0]
48        subnet_region = var.region
49      },
50      {
51        subnet_name   = "subnet-02"
52        subnet_ip     = var.cidrs[1]
53        subnet_region = var.region
54
55        subnet_private_access = "true"
56      }
57    ]
58  }
```

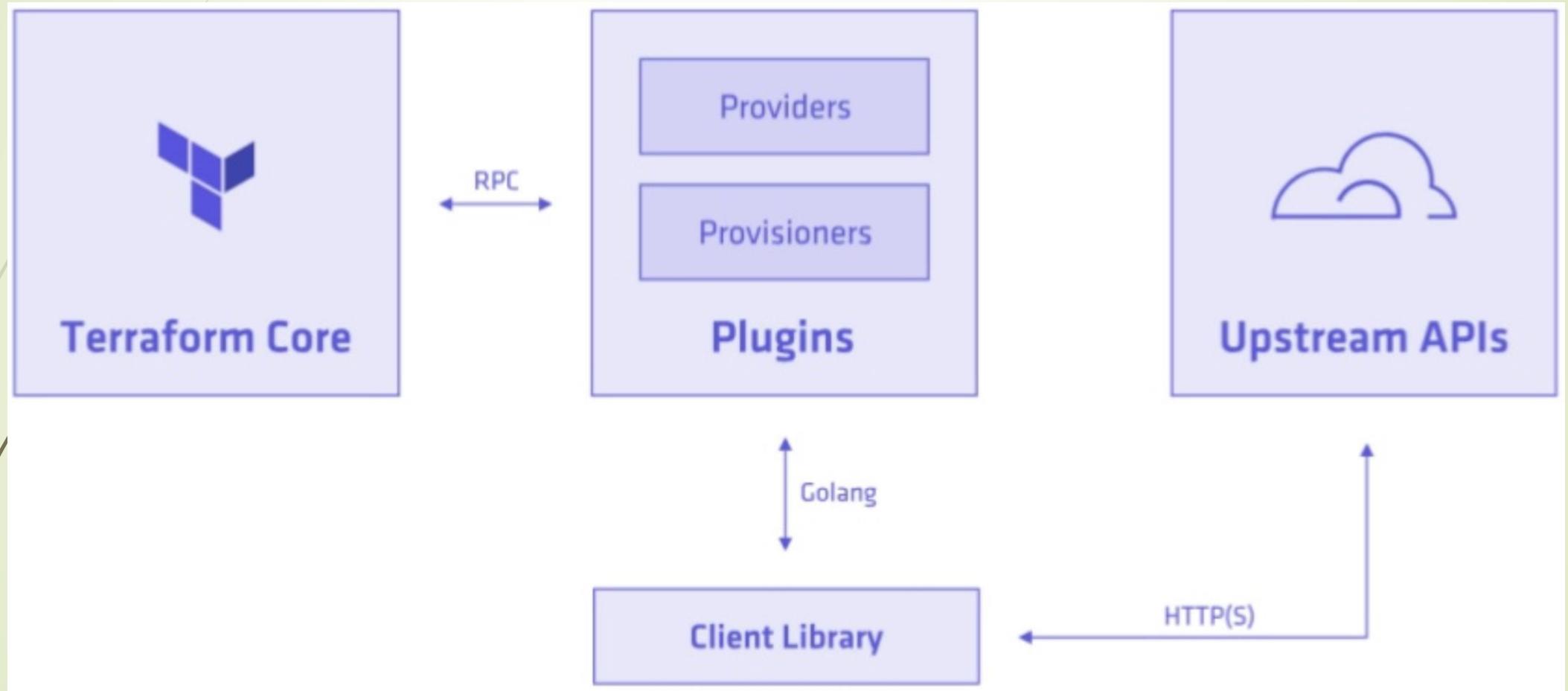
Terraform file structure

- ▶ File extension meaningful for terraform
 - ▶ Normally terraform interest all file in folder with extension *.tf or *.tf.json (Consider as working file)
 - ▶ If any file on same folder try to identify same object. Terraform will error
 - ▶ Static define terraform.tfvars or terraform.tfvars.json for store variable
 - ▶ Also consider file extension *.auto.tfvars or *.auto.tfvars.json for variable (Also load if it appear on folder)
- ▶ Terraform also generate state file call “terraform.tfstate” , “terraform.tfstate.backup” for keep status of terraform and latest result
- ▶ Terraform also generate file “tfplan” from command “terraform plan” (Obsolete on version 0.12)

Example file structure

0.IAM	00_Standard_EC2	.terraform
1.Network	01_JumpSVR_Linux	00_provider.tf
2.Instance	02_Jenkin_Master	01_ServerAZ1.tf
3.Database	03_Jenkin_Slave	01_ServerAZ3.tf
99.Test_Module	04_CMS_Build_Server	02_ServerAZ2.tf
global_variable.auto.tfvars	05_Kubernetes_Farm	99_automate.sh
	05_Kubernetes_Farm_APIConnect	destroy_terraform.sh
	05_Kubernetes_Farm_APIConnect_MNG	global_variable.auto.tfvars
	05_Kubernetes_Farm_APIConnect_MultipleAZ_MNG	log
	05_Kubernetes_Farm_APIConnect_MultipleAZ_Worker	start_terraform_auto.sh
	05_Kubernetes_Farm_CAPCO	start_terraform.sh
	05_Kubernetes_Farm_CIS_Test	terraform.tfstate
	06_Mobile_SaleTools	terraform.tfvars
	07_Pfsense_VPN	tfplan
	08_Docker_GroupLife	variables.tf
	destroy_ec2_terraform_all.sh	
	start_ec2_terraform_all.sh	

Terraform Architecture





Terraform Architecture (Provider)

- ▶ **Provider is source of resources**
- ▶ Terraform will use provider as translate API to interaction resource
- ▶ Provider will define resource and data to operate
- ▶ Multi provider is possible on same project. Terraform will automate check resource dependency by itself

Terraform Architecture (Provider)

```
provider "aws" {  
    version = "~> 2.0"  
    access_key = "XXXXXXXXXX"  
    secret_key = "YYYYYYYYYYYY"  
    region = "ap-southeast-1"  
}
```

Terraform Architecture (Provider)

The screenshot shows the Terraform AWS Provider page. At the top, there's a navigation bar with links for Intro, Learn, Docs, Community, Enterprise, Download, GitHub, and Sign In. Below the navigation is a sidebar with sections like EXPAND ALL, FILTER, All Providers, AWS Provider, Guides (including AWS Provider Version 2 Upgrade, AWS Provider Version 3 Upgrade, Custom Service Endpoints, AWS Provider Track on HashiCorp Learn), Provider Data Sources, Access Analyzer, ACM, ACM PCA, API Gateway, and Application Autoscaling. The main content area has a title "AWS Provider" and a sub-section "Example Usage" containing Terraform configuration code:

```
# Configure the AWS Provider
provider "aws" {
  version = "~> 2.0"
  region  = "us-east-1"
}

# Create a VPC
resource "aws_vpc" "example" {
  cidr_block = "10.0.0.0/16"
}
```

The screenshot shows the Terraform Resource: aws_instance page. At the top, there's a navigation bar with links for Intro, Learn, Docs, Community, Enterprise, Download, GitHub, and Sign In. Below the navigation is a sidebar with sections like EXPAND ALL, FILTER, All Providers, AWS Provider, Guides (including AWS Provider Version 2 Upgrade, AWS Provider Version 3 Upgrade, Custom Service Endpoints, AWS Provider Track on HashiCorp Learn), Provider Data Sources, Access Analyzer, ACM, ACM PCA, API Gateway, Application Autoscaling, AppMesh, AppSync, Athena, Autoscaling, Backup, Batch, Budgets, Cloud9, CloudFormation, CloudFront, and CloudHSM v2. The main content area has a title "Resource: aws_instance" and a sub-section "Example Usage" containing Terraform configuration code:

```
# Create a new instance of the latest Ubuntu 14.04 on an
# t2.micro node with an AWS Tag naming it "HelloWorld"
provider "aws" {
  region = "us-west-2"
}

data "aws_ami" "ubuntu" {
  most_recent = true

  filter {
    name   = "name"
    values = ["ubuntu/images/hvm-ssd/ubuntu-trusty-14.04-amd64-server-*"]
  }

  filter {
    name   = "virtualization-type"
    values = ["hvm"]
  }

  owners = ["099720109477"] # Canonical
}

resource "aws_instance" "web" {
  ami          = "${data.aws_ami.ubuntu.id}"
  instance_type = "t2.micro"

  tags = {
    Name = "HelloWorld"
  }
}
```

Terraform Architecture (Provider)

- ▶ Provider need authentication section for defined authority
- ▶ Depend on which provider we choose to operate
- ▶ Ex:
 - ▶ AWS:
 - ▶ Static authentication
 - ▶ Environment variable
 - ▶ Input variable
 - ▶ Share credential file
 - ▶ EC2 Role
 - ▶ vSphere
 - ▶ Username/Password
 - ▶ Digital Ocean
 - ▶ Tokenize

Terraform Architecture (Provider)

► Example AWS:

```
► provider "aws" {  
    ►   access_key = "AKIAV4FGGXKRESHGYPSONO"  
    ►   secret_key = "9vuXOnm/P8wvirYliBQgH41Rqk74X4ZWpVenSXm"  
    ►   version = "~> 2.0"  
    ►   region = var.region  
    ► }
```

Terraform Architecture (Provider)

► Example AWS:

```
► provider "aws" {  
    ►   access_key = "XXXXXXXXXXXXXXXXXX"  
    ►   secret_key = "XXXXXXXXXXXXXXXXXX"  
    ►   version = "~> 2.0"  
    ►   region = var.region  
    ► }
```

Terraform Architecture (Resource)

- ▶ Resource use to support create/read/update/delete every component that was defined from provider
- ▶ Attribute that allow to configure on each resource will operate on document
- ▶ Argument reference: Defined argument's configurable on resource

Terraform Architecture (Resource)

Resource: aws_instance

JUMP TO SECTION ▾

Provides an EC2 instance resource. This allows instances to be created, updated, and deleted. Instances also support [provisioning](#).

Example Usage

```
# Create a new instance of the latest Ubuntu 14.04 on an
# t2.micro node with an AWS Tag naming it "HelloWorld"
provider "aws" {
  region = "us-west-2"
}

data "aws_ami" "ubuntu" {
  most_recent = true

  filter {
    name   = "name"
    values = ["ubuntu/images/hvm-ssd/ubuntu-trusty-14.04-amd64-server-*"]
  }

  filter {
    name   = "virtualization-type"
    values = ["hvm"]
  }

  owners = ["099720109477"] # Canonical
}

resource "aws_instance" "web" {
  ami           = "${data.aws_ami.ubuntu.id}"
  instance_type = "t2.micro"

  tags = {
    Name = "HelloWorld"
  }
}
```

Argument Reference

The following arguments are supported:

- [ami](#) - (Required) The AMI to use for the instance.
- [availability_zone](#) - (Optional) The AZ to start the instance in.
- [placement_group](#) - (Optional) The Placement Group to start the instance in.
- [tenancy](#) - (Optional) The tenancy of the instance (if the instance is running in a VPC). An instance with a tenancy of dedicated runs on single-tenant hardware. The host tenancy is not supported for the import-instance command.
- [host_id](#) - (optional) The Id of a dedicated host that the instance will be assigned to. Use when an instance is to be launched on a specific dedicated host.
- [cpu_core_count](#) - (Optional) Sets the number of CPU cores for an instance. This option is only supported on creation of instance type that support CPU Options [CPU Cores and Threads Per CPU Core Per Instance Type](#) - specifying this option for unsupported instance types will return an error from the EC2 API.
- [cpu_threads_per_core](#) - (Optional - has no effect unless `cpu_core_count` is also set) If set to to 1, hyperthreading is disabled on the launched instance. Defaults to 2 if not set. See [Optimizing CPU Options](#) for more information.

NOTE: Changing `cpu_core_count` and/or `cpu_threads_per_core` will cause the resource to be destroyed and re-created.

- [ebs_optimized](#) - (Optional) If true, the launched EC2 instance will be EBS-optimized. Note that if this is not set on an instance type that is optimized by default then this will show as disabled but if the instance type is optimized by default then there is no need to set this and there is no effect to disabling it. See the [EBS Optimized section](#) of the AWS User Guide for more information.
- [disable_api_termination](#) - (Optional) If true, enables [EC2 Instance Termination Protection](#)
- [instance_initiated_shutdown_behavior](#) - (Optional) Shutdown behavior for the instance. Amazon defaults this to `stop` for EBS-backed instances and `terminate` for instance-store instances. Cannot be set on instance-store instances. See [Shutdown Behavior](#) for more information.
- [instance_type](#) - (Required) The type of instance to start. Updates to this field will trigger a stop/start of the EC2 instance.

Ref: <https://www.terraform.io/docs/providers/aws/r/instance.html>

Terraform Architecture (Data)

- ▶ Data is information that read from provider support in project
- ▶ Source may be:
 - ▶ Terraform enterprise
 - ▶ Provider resource (Ex: AMI owner by aws account)
 - ▶ Output from resource that exportable from resource

Terraform Architecture (Data)

Resource: aws_instance

JUMP TO SECTION ▾

Provides an EC2 instance resource. This allows instances to be created, updated, and deleted. Instances also support [provisioning](#).

Example Usage

```
# Create a new instance of the latest Ubuntu 14.04 on an
# t2.micro node with an AWS Tag naming it "HelloWorld"
provider "aws" {
  region = "us-west-2"
}

data "aws_ami" "ubuntu" {
  most_recent = true

  filter {
    name   = "name"
    values = ["ubuntu/images/hvm-ssd/ubuntu-trusty-14.04-amd64-server-*"]
  }

  filter {
    name   = "virtualization-type"
    values = ["hvm"]
  }

  owners = ["099720109477"] # Canonical
}

resource "aws_instance" "web" {
  ami      = "${data.aws_ami.ubuntu.id}"
  instance_type = "t2.micro"

  tags = {
    Name = "HelloWorld"
  }
}
```

Attributes Reference

In addition to all arguments above, the following attributes are exported:

- [id](#) - The instance ID.
- [arn](#) - The ARN of the instance.
- [availability_zone](#) - The availability zone of the instance.
- [placement_group](#) - The placement group of the instance.
- [key_name](#) - The key name of the instance
- [password_data](#) - Base-64 encoded encrypted password data for the instance. Useful for getting the administrator password for instances running Microsoft Windows. This attribute is only exported if `get_password_data` is true. Note that this encrypted value will be stored in the state file, as with all exported attributes. See [GetPasswordData](#) for more information.
- [public_dns](#) - The public DNS name assigned to the instance. For EC2-VPC, this is only available if you've enabled DNS hostnames for your VPC
- [public_ip](#) - The public IP address assigned to the instance, if applicable. **NOTE:** If you are using an `aws_eip` with your instance, you should refer to the EIP's address directly and not use `public_ip`, as this field will change after the EIP is attached.
- [ipv6_addresses](#) - A list of assigned IPv6 addresses, if any
- [primary_network_interface_id](#) - The ID of the instance's primary network interface.
- [private_dns](#) - The private DNS name assigned to the instance. Can only be used inside the Amazon EC2, and only available if you've enabled DNS hostnames for your VPC
- [private_ip](#) - The private IP address assigned to the instance
- [security_groups](#) - The associated security groups.
- [vpc_security_group_ids](#) - The associated security groups in non-default VPC
- [subnet_id](#) - The VPC subnet ID.
- [credit_specification](#) - Credit specification of instance.
- [instance_state](#) - The state of the instance. One of: `pending`, `running`, `shutting-down`, `terminated`, `stopping`, `stopped`. See [Instance Lifecycle](#) for more information.

Terraform Architecture (Provisioner)

- ▶ Provisioner was designed to inject some operation to any resource in terraform
- ▶ Support both local-provisioner (terraform's machine) or remote-provisioner (target's machine)
- ▶ Provisioner will start when resource was created
- ▶ Plug with other CM
- ▶ Use-Case: Bootstrap script for provisioner component

Terraform Architecture (Provisioner)

- ▶ Built-In Provisioner
 - ▶ Chef Provisioner
 - ▶ File Provisioner
 - ▶ Habitat Provisioner
 - ▶ Local-exec Provisioner
 - ▶ Puppet Provisioner
 - ▶ Remote-exec Provisioner
 - ▶ Salt-masterless Provisioner

Terraform Architecture (Provisioner)



Intro Learn Docs Community Enterprise Download GitHub Sign In

Terraform CLI

- EXPAND ALL | FILTER
- > Configuration Language
- > Commands (CLI)
- > Import
- > State
- > Providers
- Prov
▶ **Provisioners**
 - ▶ Provisioner Connections
 - ▶ Provisioners Without a Resource
- ▶ **Built-in Provisioners**
 - ▶ chef Provisioner
 - ▶ file Provisioner
 - ▶ habitat Provisioner
 - ▶ local-exec Provisioner
 - ▶ puppet Provisioner
 - ▶ remote-exec Provisioner
 - ▶ salt-masterless Provisioner
- > Modules
- > Backends
- > Plugins
- > Internals

local-exec Provisioner

JUMP TO SECTION ▾

The `local-exec` provisioner invokes a local executable after a resource is created. This invokes a process on the machine running Terraform, not on the resource. See the [remote-exec provisioner](#) to run commands on the resource.

Note that even though the resource will be fully created when the provisioner is run, there is no guarantee that it will be in an operable state – for example system services such as `sshd` may not be started yet on compute resources.

Note: Provisioners should only be used as a last resort. For most common situations there are better alternatives. For more information, see the [main Provisioners page](#).

Example usage

```
resource "aws_instance" "web" {
  # ...

  provisioner "local-exec" {
    command = "echo ${aws_instance.web.private_ip} >> private_ips.txt"
  }
}
```

Argument Reference

The following arguments are supported:

- `command` – (Required) This is the command to execute. It can be provided as a relative path to the current working directory or as an absolute path. It is evaluated in a shell, and can use environment variables or Terraform variables.
- `working_dir` – (Optional) If provided, specifies the working directory where `command` will be executed. It can be provided as a relative path to the current working directory or as an absolute path. The directory must exist.



Intro Learn Docs Community Enterprise Download GitHub Sign In

Terraform CLI

- EXPAND ALL | FILTER
- > Configuration Language
- > Commands (CLI)
- > Import
- > State
- > Providers
- Prov
▶ **Provisioners**
 - ▶ Provisioner Connections
 - ▶ Provisioners Without a Resource
- ▶ **Built-in Provisioners**
 - ▶ chef Provisioner
 - ▶ file Provisioner
 - ▶ habitat Provisioner
 - ▶ local-exec Provisioner
 - ▶ puppet Provisioner
 - ▶ remote-exec Provisioner
 - ▶ salt-masterless Provisioner
- > Modules
- > Backends
- > Plugins
- > Internals

remote-exec Provisioner

JUMP TO SECTION ▾

The `remote-exec` provisioner invokes a script on a remote resource after it is created. This can be used to run a configuration management tool, bootstrap into a cluster, etc. To invoke a local process, see the [local-exec provisioner](#) instead. The `remote-exec` provisioner supports both `ssh` and `winrm` type [connections](#).

Note: Provisioners should only be used as a last resort. For most common situations there are better alternatives. For more information, see the [main Provisioners page](#).

Example usage

```
resource "aws_instance" "web" {
  # ...

  provisioner "remote-exec" {
    inline = [
      "puppet apply",
      "consul join ${aws_instance.web.private_ip}",
    ]
  }
}
```

Argument Reference

The following arguments are supported:

- `inline` – This is a list of command strings. They are executed in the order they are provided. This cannot be provided with `script` or `scripts`.
- `script` – This is a path (relative or absolute) to a local script that will be copied to the remote resource and then executed. This cannot be provided with `inline` or `scripts`.
- `scripts` – This is a list of paths (relative or absolute) to local scripts that will be copied to the remote resource and then executed. They are executed in the order they are provided. This cannot be provided with `inline` or `script`.



Terraform State Management

- ▶ Terraform have file called: “terraform.tfstate” for keep source of through and state of current operation
- ▶ Before start any operate. Terraform will check for tfstate first. If it not available (first time operate). It will create by itself.
- ▶ Double operate was prohibit by feature “State Locking”
- ▶ Terraform need state for several reason.
 - ▶ Mapping to the Real World
 - ▶ Meta Data
 - ▶ Performance
 - ▶ Syncing

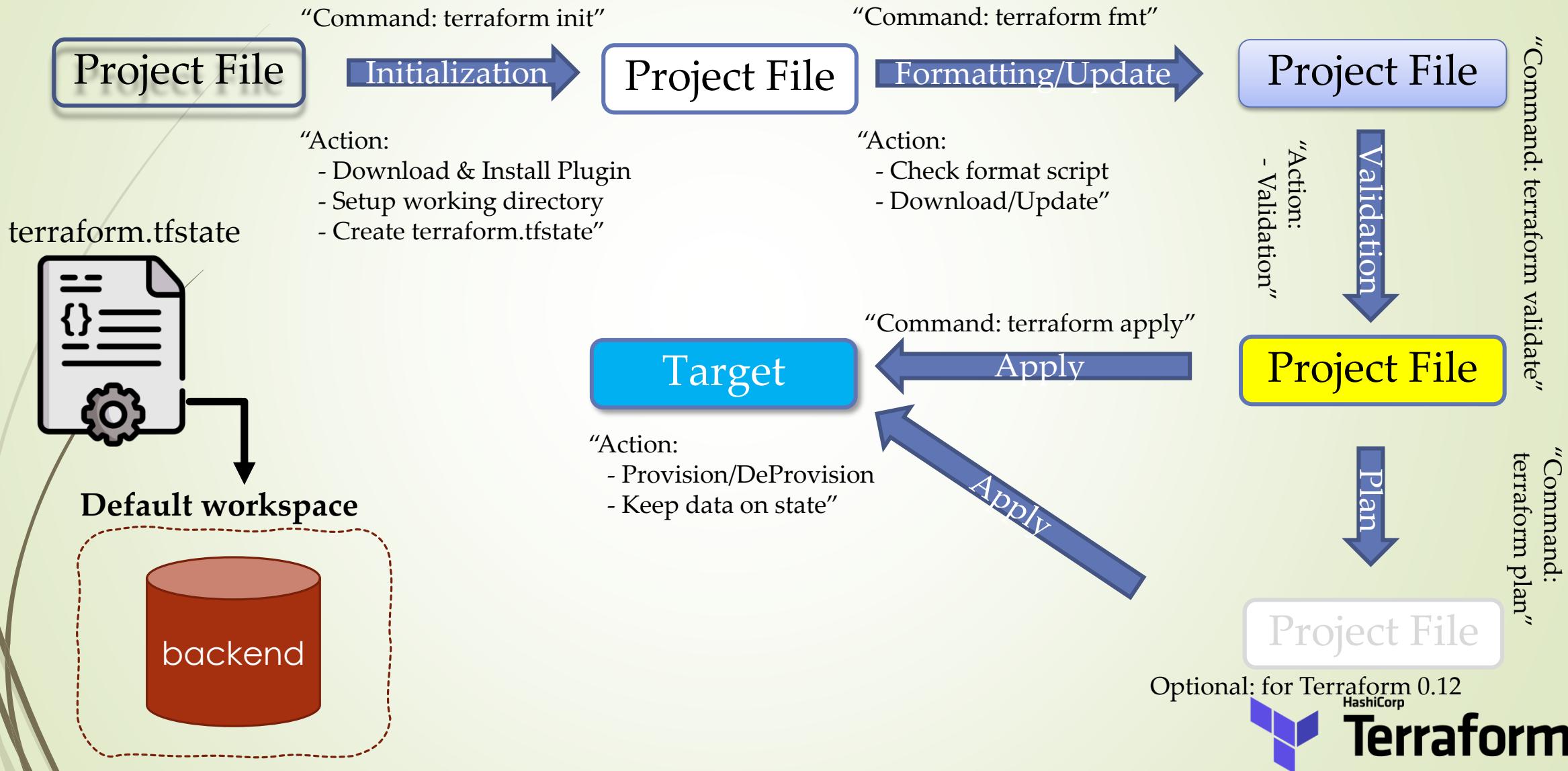
Terraform State Management

- ▶ Normally terraform need **Backend** to keep all persistence data storage(include state file)
- ▶ All backend will belong to logical unit call **Workspace**
- ▶ By default terraform initial only single workspace call “default” and have only 1 state associate with this configuration
- ▶ Anyway we can operate multiple workspace to single backend
- ▶ This will get multiple state file reuse single backend and share authentication credentials

Terraform State Management

- ▶ Multiple workspaces support several backends:
 - ▶ AzureRM
 - ▶ Consul
 - ▶ GCS
 - ▶ Local
 - ▶ Manta
 - ▶ Postgres
 - ▶ Remote
 - ▶ S3

Terraform State Management



Terraform State Management

```
[ubuntu@ip-172-31-20-46:~/terraform_202003/Workshop_1.2_Provider/01_static_provider$ terraform init
Initializing the backend...
Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "aws" (hashicorp/aws) 2.44.0...
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.
[ubuntu@ip-172-31-20-46:~/terraform_202003/Workshop_1.2_Provider/01_static_provider$ terraform fmt -check
01_static_provider.tf
[ubuntu@ip-172-31-20-46:~/terraform_202003/Workshop_1.2_Provider/01_static_provider$ terraform validate
Success! The configuration is valid.

[ubuntu@ip-172-31-20-46:~/terraform_202003/Workshop_1.2_Provider/01_static_provider$ terraform apply
data.aws_vpc.vpc: Refreshing state...

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

Outputs:

aws_vpc_arn = arn:aws:ec2:ap-southeast-1:404075494050:vpc/vpc-4483e921
aws_vpc_cidr = 172.31.0.0/16
ubuntu@ip-172-31-20-46:~/terraform_202003/Workshop_1.2_Provider/01_static_provider$ █
```

Workshop: Provider



Workshop Provider

► Example 1: Static Provider

```
1  variable "region" {}

2

3  provider "aws" {
4    access_key = "<access key>"
5    secret_key = "<secret key>"
6    version = "~> 2.0"
7    region = var.region
8  }

9

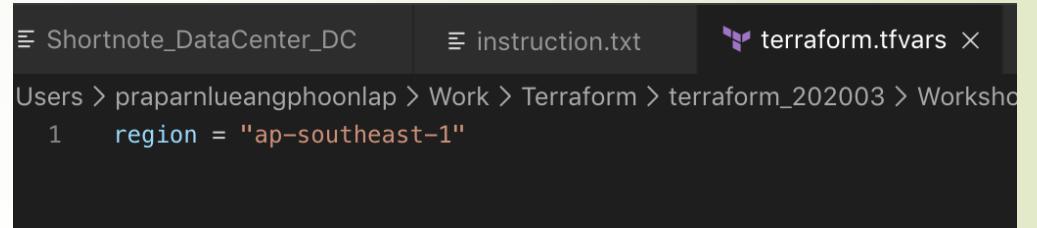
10 data "aws_vpc" "vpc" {
11   id = "vpc-4483e921"
12 }

13

14 output "aws_vpc_arn" {
15   value = data.aws_vpc.vpc.arn
16 }

17

18 output "aws_vpc_cidr" {
19   value = data.aws_vpc.vpc.cidr_block
20 }
```



The screenshot shows a terminal window with the following structure:

```
Users > paparnlueangphoonlap > Work > Terraform > terraform_202003 > Workshop
```

Two files are visible:

- `instruction.txt`: An empty file.
- `terraformer.tfvars`: A file containing the following content:

```
1  region = "ap-southeast-1"
```

Workshop Provider

► Example 1: Static Provider

```
[ubuntu@ip-172-31-20-46:~/terraform_202003/Workshop_1.2_Provider/01_static_provider$ terraform init
```

```
Initializing the backend...
```

```
Initializing provider plugins...
```

- Checking for available provider plugins...
- Downloading plugin for provider "aws" (hashicorp/aws) 2.44.0...

```
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.

```
[ubuntu@ip-172-31-20-46:~/terraform_202003/Workshop_1.2_Provider/01_static_provider$ terraform fmt -check  
01_static_provider.tf
```

```
[ubuntu@ip-172-31-20-46:~/terraform_202003/Workshop_1.2_Provider/01_static_provider$ terraform validate  
Success! The configuration is valid.
```

```
[ubuntu@ip-172-31-20-46:~/terraform_202003/Workshop_1.2_Provider/01_static_provider$ terraform apply  
data.aws_vpc.vpc: Refreshing state...
```

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

```
Outputs:
```

```
aws_vpc_arn = arn:aws:ec2:ap-southeast-1:404075494050:vpc/vpc-4483e921  
aws_vpc_cidr = 172.31.0.0/16  
ubuntu@ip-172-31-20-46:~/terraform_202003/Workshop_1.2_Provider/01_static_provider$ █
```

Workshop Provider

► Example 2: Environment Provider

```
1 provider "aws" {
2   |   version = "~> 2.0"
3 }
4
5 data "aws_vpc" "vpc" {
6   |   id = "vpc-4483e921"
7 }
8
9 output "aws_vpc_arn" {
10  |   value = data.aws_vpc.vpc.arn
11 }
12
13 output "aws_vpc_cidr" {
14  |   value = data.aws_vpc.vpc.cidr_block
15 }
```

Workshop Provider

► Example 2: Environment Provider

```
[ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/02_environment_provider$ export AWS_ACCESS_KEY_ID="AKI  
[ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/02_environment_provider$ export AWS_SECRET_ACCESS_KEY=  
[ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/02_environment_provider$ export AWS_DEFAULT_REGION="ap-southeast-1"  
[ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/02_environment_provider$ terraform init
```

Initializing the backend...

Initializing provider plugins...

- Checking for available provider plugins...
- Downloading plugin for provider "aws" (hashicorp/aws) 2.44.0...

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.

```
[ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/02_environment_provider$ terraform fmt -check  
01_environment_provider.tf  
[ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/02_environment_provider$ terraform validate  
Success! The configuration is valid.
```

```
[ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/02_environment_provider$ terraform apply  
data.aws_vpc.vpc: Refreshing state...
```

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

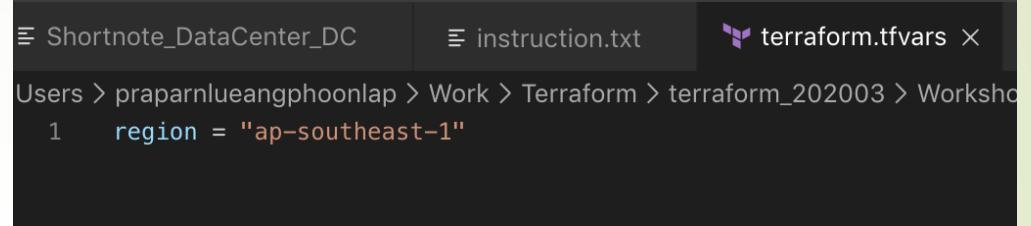
Outputs:

```
aws_vpc_arn = arn:aws:ec2:ap-southeast-1:404075494050:vpc/vpc-4483e921  
aws_vpc_cidr = 172.31.0.0/16
```

Workshop Provider

► Example 3: Input Variable Provider

```
1  variable "region" {}
2  variable "access_key" {}
3  variable "secret_key" {}
4  provider "aws" {
5    access_key = var.access_key
6    secret_key = var.secret_key
7    version = "~> 2.0"
8    region = var.region
9  }
10
11 data "aws_vpc" "vpc" {
12   id = "vpc-4483e921"
13 }
14
15 output "aws_vpc_arn" {
16   value = data.aws_vpc.vpc.arn
17 }
18
19 output "aws_vpc_cidr" [
20   value = data.aws_vpc.vpc.cidr_block
21 ]
```



The screenshot shows a terminal window with the following directory structure:

```
Users > paparnlueangphoonlap > Work > Terraform > terraform_202003 > Workshop
```

And the contents of the `terraform.tfvars` file:

```
1  region = "ap-southeast-1"
```

Workshop Provider

► Example 3: Input Variable Provider

```
[ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/03_inputvariable_provider$ export TF_VAR_access_key="[REDACTED]"
[ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/03_inputvariable_provider$ export TF_VAR_secret_key="[REDACTED"]
[ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/03_inputvariable_provider$ terraform init
```

Initializing the backend...

Initializing provider plugins...

- Checking for available provider plugins...
- Downloading plugin for provider "aws" (hashicorp/aws) 2.44.0...

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.

```
ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/03_inputvariable_provider$ terraform fmt -check
```

```
01_inputvariable_provider.tf
```

```
ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/03_inputvariable_provider$ terraform validate
```

```
Success! The configuration is valid.
```

```
ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/03_inputvariable_provider$ terraform apply
```

```
data.aws_vpc.vpc: Refreshing state...
```

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

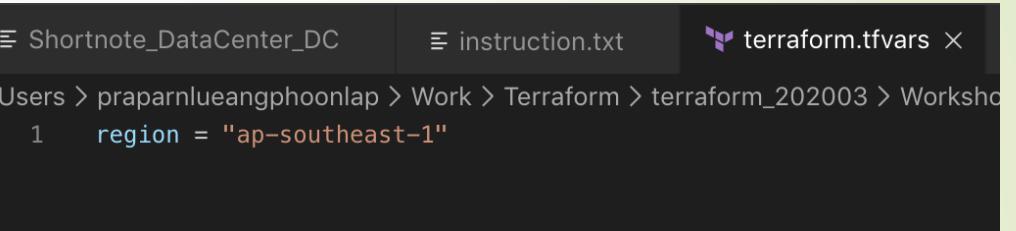
Outputs:

```
aws_vpc_arn = arn:aws:ec2:ap-southeast-1:404075494050:vpc/vpc-4483e921
aws_vpc_cidr = 172.31.0.0/16
```

Workshop Provider

► Example 4: Credential File Provider

```
1 variable "region" {}
2 provider "aws" {
3   profile = "default"
4   version = "~> 2.0"
5   region = var.region
6 }
7
8 data "aws_vpc" "vpc" {
9   id = "vpc-4483e921"
10 }
11
12 output "aws_vpc_arn" {
13   value = data.aws_vpc.vpc.arn
14 }
15
16 output "aws_vpc_cidr" {
17   value = data.aws_vpc.vpc.cidr_block
18 }
```



```
Shortnote_DataCenter_DC instruction.txt terraform.tfvars
Users > praparnlueangphoonlap > Work > Terraform > terraform_202003 > Workshop
  1   region = "ap-southeast-1"
```

Workshop Provider

► Example 4: Credential File Provider

```
[ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/04_credentialfile_provider$ aws configure
[AWS Access Key ID [None]:
[AWS Secret Access Key [Nor
[Default region name [None]: ap-southeast-1
[Default output format [None]: json
[ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/04_credentialfile_provider$ more ~/.aws/credentials
[default]
aws_access_key_id = AK:
aws_secret_access_key =
[ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/04_credentialfile_provider$ more ~/.aws/config
[default]
output = json
region = ap-southeast-1
ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/04_credentialfile_provider$ ]
```

Workshop Provider

► Example 4: Credential File Provider

```
[ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/04_credentialfile_provider$ terraform init

Initializing the backend...

Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "aws" (hashicorp/aws) 2.44.0...

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.
ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/04_credentialfile_provider$ terraform fmt -check
01_environment_provider.tf
ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/04_credentialfile_provider$      terraform validate
Success! The configuration is valid.

[ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/04_credentialfile_provider$      terraform apply
data.aws_vpc.vpc: Refreshing state...

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

Outputs:

aws_vpc_arn = arn:aws:ec2:ap-southeast-1:404075494050:vpc/vpc-4483e921
aws_vpc_cidr = 172.31.0.0/16
ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/04_credentialfile_provider$ ]
```

Workshop Provider

► Example 5: EC2 Role Provider

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	
ec2-lab-master	i-05b88a54a3a5ee762	t3.micro	ap-southeast-1a	running	2/2 checks ...	None	ec2-54-179-128-234.ap...	54.179.128.234	-	
Instance: i-05b88a54a3a5ee762 (ec2-lab-master) Public DNS: ec2-54-179-128-234.ap-southeast-1.compute.amazonaws.com										
Description	Status Checks	Monitoring	Tags							
Instance ID	i-05b88a54a3a5ee762				Public DNS (IPv4)	ec2-54-179-128-234.ap-southeast-1.compute.amazonaws.com				
Instance state	running				IPv4 Public IP	54.179.128.234				
Instance type	t3.micro				IPv6 IPs	-				
Elastic IPs					Private DNS	ip-172-31-23-74.ap-southeast-1.compute.internal				
Availability zone	ap-southeast-1a				Private IPs	172.31.23.74				
Security groups	secgroup_labserver	view inbound rules	view outbound rules		Secondary private IPs					
Scheduled events	No scheduled events				VPC ID	vpc-4483e921				
AMI ID	ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20180522 (ami-81cefcd)				Subnet ID	subnet-41742436				
Platform	-				Network interfaces	eth0				
IAM role	iamrole_ec2				Source/dest. check	True				
Key pair name	keypair				T2/T3 Unlimited	Enabled				

Workshop Provider

► Example 5: EC2 Role Provider

Roles > iamrole_ec2

Summary

Role ARN: arn:aws:iam::404075494050:role/iamrole_ec2 [Edit](#)

Role description: [Edit](#)

Instance Profile ARNs: arn:aws:iam::404075494050:instance-profile/iaminstanceprofile_ec2 [Edit](#)

Path: /

Creation time: 2020-01-08 22:37 UTC+0700

Last activity: Not accessed in the tracking period

Maximum CLI/API session duration: 1 hour [Edit](#)

[Permissions](#) [Trust relationships](#) [Tags](#) [Access Advisor](#) [Revoke sessions](#)

▼ Permissions policies (1 policy applied)

[Attach policies](#) [Add inline policy](#)

Policy name	Policy type	X
iampolicy_ec2	Managed policy	X

[Policy summary](#) [JSON](#) [Edit policy](#) [Simulate policy](#)

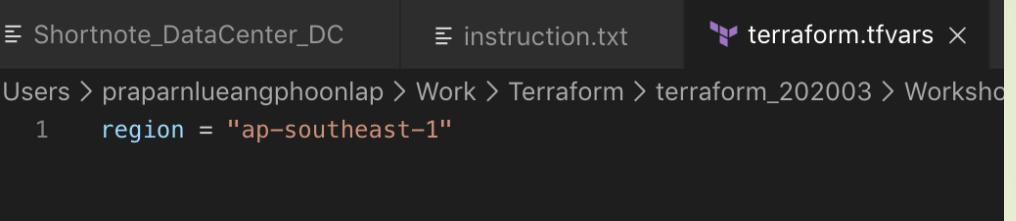
```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Action": "*",  
7       "Resource": "*"  
8     }  
9   ]  
10 }
```

aform

Workshop Provider

► Example 5: EC2 Role Provider

```
1  variable "region" {}
2  provider "aws" {
3    region = var.region
4  }
5
6  data "aws_vpc" "vpc" {
7    id = "vpc-4483e921"
8  }
9
10 output "aws_vpc_arn" {
11   value = data.aws_vpc.vpc.arn
12 }
13
14 output "aws_vpc_cidr" {
15   value = data.aws_vpc.vpc.cidr_block
16 }
```



```
Shortnote_DataCenter_DC instruction.txt terraform.tfvars
Users > praparnlueangphoonlap > Work > Terraform > terraform_202003 > Workshop
  1   region = "ap-southeast-1"
```

Workshop Provider

► Example 5: EC2 Role Provider

```
ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/05_ec2role_provider$ terraform init
Initializing the backend...
Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "aws" (hashicorp/aws) 2.44.0...
The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.aws: version = "~> 2.44"

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.
ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/05_ec2role_provider$ terraform fmt --check
01_ec2role_provider.tf
ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/05_ec2role_provider$ terraform validate
Success! The configuration is valid.

ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/05_ec2role_provider$ terraform apply
data.aws_vpc.vpc: Refreshing state...

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

Outputs:

aws_vpc_arn = arn:aws:ec2:ap-southeast-1:404075494050:vpc/vpc-4483e921
aws_vpc_cidr = 172.31.0.0/16
ubuntu@ip-172-31-23-74:~/terraform_202003/Workshop_1.2_Provider/05_ec2role_provider$ █
```

Q&A



Resource and dependency

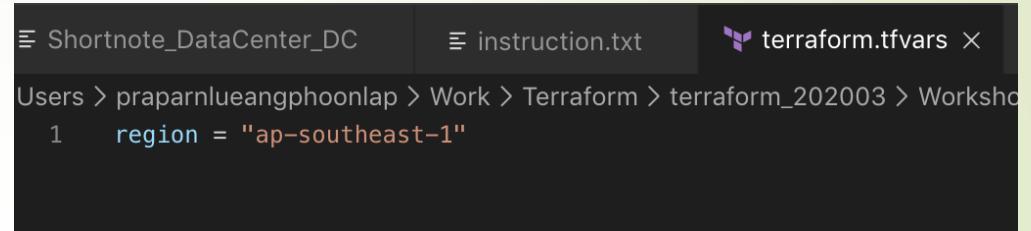
► Basic Resource

- Basically resource was defined to create object in scope of provider in terraform
- Output from resource will be what we need to provision via terraform
- Resource's dependency will take care by terraform when some resource need input from another resource (implicit dependency)
- Anyway if some resource that not relate with the other. But we need terraform for wait dependency resource. We can define "depend_on" keyword for terraform handle this consequence. (explicit dependency)

Resource and dependency

► Example1: Create EC2 Instance by resource

```
1  variable "region" {}
2  provider "aws" {
3    |  region = var.region
4  }
5
6  resource "aws_instance" "lab" {
7    |  ami          = "ami-81cefcd"
8    |  instance_type = "t3a.nano"
9    |  key_name = "keypair"
10   }
11
12  output "aws_instance_lab_id" {
13    |  value = aws_instance.lab.id
14  }
15
16  output "aws_instance_lab_public_ip" {
17    |  value = aws_instance.lab.public_ip
18  }
19
20  output "aws_instance_lab_public_dns" {
21    |  value = aws_instance.lab.public_dns
22  }
23
24  output "aws_instance_lab_keyname" [
25    |  value = aws_instance.lab.key_name
26  ]
```



The screenshot shows a terminal window with the following structure:

- File: Shortnote_DataCenter_DC
- File: instruction.txt
- File: terraform.tfvars (highlighted)

The content of the terraform.tfvars file is:

```
1  region = "ap-southeast-1"
```

Ref: <https://www.terraform.io/docs/providers/aws/r/instance.html>

Resource and dependency

The screenshot shows the Terraform documentation for the `aws_instance` resource. The page has a blue header with the Terraform logo and navigation links for Intro, Learn, Docs, Community, Enterprise, Download, GitHub, and Sign In. The main content area has a title "Resource: aws_instance" and a sub-section "Example Usage" containing a Terraform configuration snippet.

```
# Create a new instance of the latest Ubuntu 14.04 on an
# t2.micro node with an AWS Tag naming it "HelloWorld"
provider "aws" {
  region = "us-west-2"
}

data "aws_ami" "ubuntu" {
  most_recent = true

  filter {
    name   = "name"
    values = ["ubuntu/images/hvm-ssd/ubuntu-trusty-14.04-amd64-server-*"]
  }

  filter {
    name   = "virtualization-type"
    values = ["hvm"]
  }

  owners = ["099720109477"] # Canonical
}

resource "aws_instance" "web" {
  ami           = "${data.aws_ami.ubuntu.id}"
  instance_type = "t2.micro"

  tags = {
    Name = "HelloWorld"
  }
}
```

Argument Reference

The following arguments are supported:

- `ami` - (Required) The AMI to use for the instance.
- `availability_zone` - (Optional) The AZ to start the instance in.
- `placement_group` - (Optional) The Placement Group to start the instance in.
- `tenancy` - (Optional) The tenancy of the instance (if the instance is running in a VPC). An instance with a tenancy of dedicated runs on single-tenant hardware. The host tenancy is not supported for the import-instance command.
- `host_id` - (optional) The Id of a dedicated host that the instance will be assigned to. Use when an instance is to be launched on a specific dedicated host.
- `cpu_core_count` - (Optional) Sets the number of CPU cores for an instance. This option is only supported on creation of instance type that support CPU Options [CPU Cores and Threads Per CPU Core Per Instance Type](#) - specifying this option for unsupported instance types will return an error from the EC2 API.
- `cpu_threads_per_core` - (Optional - has no effect unless `cpu_core_count` is also set) If set to to 1, hyperthreading is disabled on the launched instance. Defaults to 2 if not set. See [Optimizing CPU Options](#) for more information.

NOTE: Changing `cpu_core_count` and/or `cpu_threads_per_core` will cause the resource to be destroyed and re-created.

- `ebs_optimized` - (Optional) If true, the launched EC2 instance will be EBS-optimized. Note that if this is not set on an instance type that is optimized by default then this will show as disabled but if the instance type is optimized by default then there is no need to set this and there is no effect to disabling it. See the [EBS Optimized](#) section of the AWS User Guide for more information.
- `disable_api_termination` - (Optional) If true, enables [EC2 Instance Termination Protection](#)
- `instance_initiated_shutdown_behavior` - (Optional) Shutdown behavior for the instance. Amazon defaults this to `stop` for EBS-backed instances and `terminate` for instance-store instances. Cannot be set on instance-store instances. See [Shutdown Behavior](#) for more information.
- `instance_type` - (Required) The type of instance to start. Updates to this field will trigger a stop/start of the EC2 instance.

Resource and dependency

- `instance_type` - (Required) The type of instance to start. Updates to this field will trigger a stop/start of the EC2 instance.
- `key_name` - (Optional) The key name of the Key Pair to use for the instance; which can be managed using the [aws_key_pair](#) resource.
- `get_password_data` - (Optional) If true, wait for password data to become available and retrieve it. Useful for getting the administrator password for instances running Microsoft Windows. The password data is exported to the `password_data` attribute. See [GetPasswordData](#) for more information.
- `monitoring` - (Optional) If true, the launched EC2 instance will have detailed monitoring enabled. (Available since v0.6.0)
- `security_groups` - (Optional, EC2-Classic and default VPC only) A list of security group names (EC2-Classic) or IDs (default VPC) to associate with.

NOTE: If you are creating Instances in a VPC, use `vpc_security_group_ids` instead.

- `vpc_security_group_ids` - (Optional, VPC only) A list of security group IDs to associate with.
- `subnet_id` - (Optional) The VPC Subnet ID to launch in.
- `associate_public_ip_address` - (Optional) Associate a public ip address with an instance in a VPC. Boolean value.
- `private_ip` - (Optional) Private IP address to associate with the instance in a VPC.
- `source_dest_check` - (Optional) Controls if traffic is routed to the instance when the destination address does not match the instance. Used for NAT or VPNs. Defaults true.
- `user_data` - (Optional) The user data to provide when launching the instance. Do not pass gzip-compressed data via this argument; see `user_data_base64` instead.
- `user_data_base64` - (Optional) Can be used instead of `user_data` to pass base64-encoded binary data directly. Use this instead of `user_data` whenever the value is not a valid UTF-8 string. For example, gzip-encoded user data must be base64-encoded and passed via this argument to avoid corruption.
- `iam_instance_profile` - (Optional) The IAM Instance Profile to launch the instance with. Specified as the name of the Instance Profile. Ensure your credentials have the correct permission to assign the instance profile according to the [EC2 documentation](#), notably `iam:PassRole`.
- `ipv6_address_count` - (Optional) A number of IPv6 addresses to associate with the primary network interface. Amazon EC2 chooses the IPv6 addresses from the range of your subnet.

- `volume_tags` - (Optional) A mapping of tags to assign to the devices created by the instance at launch time.
- `root_block_device` - (Optional) Customize details about the root block device of the instance. See [Block Devices](#) below for details.
- `ebs_block_device` - (Optional) Additional EBS block devices to attach to the instance. Block device configurations only apply on resource creation. See [Block Devices](#) below for details on attributes and drift detection.
- `ephemeral_block_device` - (Optional) Customize Ephemeral (also known as "Instance Store") volumes on the instance. See [Block Devices](#) below for details.
- `network_interface` - (Optional) Customize network interfaces to be attached at instance boot time. See [Network Interfaces](#) below for more details.
- `credit_specification` - (Optional) Customize the credit specification of the instance. See [Credit Specification](#) below for more details.

Resource and dependency

Timeouts

The `timeouts` block allows you to specify `timeouts` for certain actions:

- `create` – (Defaults to 10 mins) Used when launching the instance (until it reaches the initial `running` state)
- `update` – (Defaults to 10 mins) Used when stopping and starting the instance when necessary during update – e.g. when changing instance type
- `delete` – (Defaults to 20 mins) Used when terminating the instance

Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` – The instance ID.
- `arn` – The ARN of the instance.
- `availability_zone` – The availability zone of the instance.
- `placement_group` – The placement group of the instance.
- `key_name` – The key name of the instance
- `password_data` – Base-64 encoded encrypted password data for the instance. Useful for getting the administrator password for instances running Microsoft Windows. This attribute is only exported if `get_password_data` is true. Note that this encrypted value will be stored in the state file, as with all exported attributes. See [GetPasswordData](#) for more information.
- `public_dns` – The public DNS name assigned to the instance. For EC2-VPC, this is only available if you've enabled DNS hostnames for your VPC
- `public_ip` – The public IP address assigned to the instance, if applicable. **NOTE:** If you are using an `aws_eip` with your instance, you should refer to the EIP's address directly and not use `public_ip`, as this field will change after the EIP is attached.
- `ipv6_addresses` – A list of assigned IPv6 addresses, if any
- `primary_network_interface_id` – The ID of the instance's primary network interface.
- `private_dns` – The private DNS name assigned to the instance. Can only be used inside the Amazon EC2, and only available if you've enabled DNS hostnames for your VPC
- `private_ip` – The private IP address assigned to the instance
- `security_groups` – The associated security groups.
- `vpc_security_group_ids` – The associated security groups in non-default VPC
- `subnet_id` – The VPC subnet ID.
- `credit_specification` – Credit specification of instance.
- `instance_state` – The state of the instance. One of: `pending`, `running`, `shutting-down`, `terminated`, `stopping`, `stopped`. See [Instance Lifecycle](#) for more information.

Resource and dependency

► Example: Create: Provision EC2 by terraform script

```
[ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/01_basic_resource]$ terraform init

Initializing the backend...

Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "aws" (hashicorp/aws) 2.45.0...

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.aws: version = "~> 2.45"

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.
[ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/01_basic_resource]$ terraform fmt -check
01_basic_resource.tf
[ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/01_basic_resource]$ terraform validate
Success! The configuration is valid.

[ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/01_basic_resource]$ █
```

Resource and dependency

➡ Example: Create: Check and Confirm for Operate

```
ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/01_basic_resource$ terraform apply

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_instance.lab will be created
+ resource "aws_instance" "lab" {
    + ami                               = "ami-81cefcfd"
    + arn                             = (known after apply)
    + associate_public_ip_address     = (known after apply)
    + availability_zone                = (known after apply)
    + cpu_core_count                  = (known after apply)
    + cpu_threads_per_core           = (known after apply)
    + get_password_data              = false
    + host_id                         = (known after apply)
    + id                             = (known after apply)
    + instance_state                 = (known after apply)
    + instance_type                  = "t3a.nano"
    + ipv6_address_count             = (known after apply)
    + ipv6_addresses                 = (known after apply)
    + key_name                        = "keypair"
    + network_interface_id           = (known after apply)
    + password_data                  = (known after apply)
    + placement_group                = (known after apply)
    + primary_network_interface_id   = (known after apply)
    + private_dns                     = (known after apply)
    + private_ip                      = (known after apply)
    + public_dns                      = (known after apply)
    + public_ip                       = (known after apply)
    + security_groups                = (known after apply)
    + source_dest_check              = true
    + subnet_id                      = (known after apply)
    + tenancy                         = (known after apply)
    + volume_tags                     = (known after apply)
    + vpc_security_group_ids         = (known after apply)
```

```
+ ebs_block_device {
    + delete_on_termination = (known after apply)
    + device_name          = (known after apply)
    + encrypted            = (known after apply)
    + iops                 = (known after apply)
    + kms_key_id           = (known after apply)
    + snapshot_id          = (known after apply)
    + volume_id             = (known after apply)
    + volume_size           = (known after apply)
    + volume_type           = (known after apply)
}

+ ephemeral_block_device {
    + device_name      = (known after apply)
    + no_device        = (known after apply)
    + virtual_name     = (known after apply)
}

+ network_interface {
    + delete_on_termination = (known after apply)
    + device_index       = (known after apply)
    + network_interface_id = (known after apply)
}

+ root_block_device {
    + delete_on_termination = (known after apply)
    + encrypted           = (known after apply)
    + iops                 = (known after apply)
    + kms_key_id           = (known after apply)
    + volume_id             = (known after apply)
    + volume_size           = (known after apply)
    + volume_type           = (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: █
```

Resource and dependency

➡ Example: Create: Check output result

```
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

aws_instance.lab: Creating...
aws_instance.lab: Still creating... [10s elapsed]
aws_instance.lab: Creation complete after 12s [id=i-0725fc7414fcf401a]

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

Outputs:

aws_instance_lab_id = i-0725fc7414fcf401a
aws_instance_lab_keyname = keypair
aws_instance_lab_public_dns = ec2-13-229-47-208.ap-southeast-1.compute.amazonaws.com
aws_instance_lab_public_ip = 13.229.47.208
ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/01_basic_resource$ █
```

Resource and dependency

➡ Example: Create: Check output result via aws cli

```
ubuntu@ip-172-31-27-186:~$ aws configure
AWS Access Key ID [*****06WZ]:
AWS Secret Access Key [*****117]:
Default region name [ap-southeast-1]:
Default output format [json]: table
ubuntu@ip-172-31-27-186:~$ aws ec2 describe-instances --instance-ids i-0725fc7414fcf401a

|             DescribeInstances
+-----+
|             Reservations
+-----+
|             OwnerId | 404075494050
|             ReservationId | r-0143ecad3e17fbe99
+-----+
|             Instances
+-----+
|             AniLaunchIndex | 0
|             Architecture | x86_64
|             ClientToken |
|             EbsOptimized | False
|             EnaSupport | True
|             Hypervisor | xen
|             ImageId | ami-81cecfcd
|             InstanceId | i-0725fc7414fcf401a
|             InstanceType | t3a.nano
|             KeyName | keypair
|             LaunchTime | 2020-01-19T13:59:18.000Z
|             PrivateDnsName | ip-172-31-19-122.ap-southeast-1.compute.internal
|             PrivateIpAddress | 172.31.19.122
|             PublicDnsName | ec2-13-229-47-208.ap-southeast-1.compute.amazonaws.com
|             PublicIpAddress | 13.229.47.208
|             RootDeviceName | /dev/sda1
|             RootDeviceType | ebs
|             SourceDestCheck | True
|             StateTransitionReason |
|             SubnetId | subnet-41742436
|             VirtualizationType | hvm
|             VpcId | vpc-4483e921
+-----+
|             BlockDeviceMappings
+-----+
|             DeviceName | /dev/sda1
+-----+
|             Ebs
+-----+
|             AttachTime | 2020-01-19T13:59:18.000Z
|             DeleteOnTermination | True
|             Status | attached
|             VolumeId | vol-01e22299a2f2baa1f
+-----+
|             CapacityReservationSpecification
+-----+
|             CapacityReservationPreference | open
+-----+
|             CpuOptions
+-----+
|             CoreCount | 1
|             ThreadsPerCore | 2
+-----+
|             HibernationOptions
+-----+
|             Configured | False
+-----+
```

```
+-----+-----+-----+
|             MetadataOptions
+-----+-----+-----+
|             HttpEndpoint | enabled
|             HttpPutResponseHopLimit | 1
|             HttpTokens | optional
|             State | applied
+-----+-----+-----+
|             Monitoring
+-----+-----+-----+
|             State | disabled
+-----+-----+-----+
|             NetworkInterfaces
+-----+-----+-----+
|             Description | interface
|             InterfaceType | interface
|             MacAddress | 06:62:d8:ab:53:aa
|             NetworkInterfaceId | eni-05310d02effc27410
|             OwnerId | 404075494050
|             PrivateDnsName | ip-172-31-19-122.ap-southeast-1.compute.internal
|             PrivateIpAddress | 172.31.19.122
|             SourceDestCheck | True
|             Status | in-use
|             SubnetId | subnet-41742436
|             VpcId | vpc-4483e921
+-----+-----+-----+
|             Association
+-----+-----+-----+
|             IpOwnerId | amazon
|             PublicDnsName | ec2-13-229-47-208.ap-southeast-1.compute.amazonaws.com
|             PublicIp | 13.229.47.208
+-----+-----+-----+
|             Attachment
+-----+-----+-----+
|             AttachTime | 2020-01-19T13:59:18.000Z
|             AttachmentId | eni-attach-02af6f2e9282f5e63
|             DeleteOnTermination | True
|             DeviceIndex | 0
|             Status | attached
+-----+-----+-----+
|             Groups
+-----+-----+-----+
|             GroupId | sg-fe2d3c9b
|            GroupName | default
+-----+-----+-----+
|             PrivateIpAddresses
+-----+-----+-----+
|             Primary | True
|             PrivateDnsName | ip-172-31-19-122.ap-southeast-1.compute.internal
|             PrivateIpAddress | 172.31.19.122
+-----+-----+-----+
|             Association
+-----+-----+-----+
|             IpOwnerId | amazon
|             PublicDnsName | ec2-13-229-47-208.ap-southeast-1.compute.amazonaws.com
|             PublicIp | 13.229.47.208
+-----+-----+-----+
|             Placement
+-----+-----+-----+
|             AvailabilityZone | ap-southeast-1a
|             GroupName | default
|             Tenancy | default
+-----+-----+-----+
```

```
+-----+-----+-----+
|             SecurityGroups
+-----+-----+-----+
|             GroupId | sg-fe2d3c9b
|             GroupName | default
+-----+-----+-----+
|             State
+-----+-----+-----+
|             Code | 16
|             Name | running
+-----+-----+-----+
```

Resource and dependency

➡ Example: Create: Test connect via ssh to new EC2 machine

```
[praparn-MacBook-Pro:~ ssh praparn@13.229.47.208
The authenticity of host '13.229.47.208' (13.229.47.208) can't be established.
ECDSA key fingerprint is SHA256:DWD3aGau4ZLaHKq23o5cxCKUk0VJvnMD+NsxstneLqk.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '13.229.47.208' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-1060-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
 http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

-----WARNING! Your environment specifies an invalid locale.
The unknown environment variables are:
LC_CTYPE=UTF-8 LC_ALL=
This can affect your user experience significantly, including the
ability to manage packages. You may install the locales by running:
sudo apt-get install language-pack-UTF-8
or
sudo locale-gen UTF-8
-----To see all available language packs, run:
apt-cache search "^language-pack-[a-z][a-z]$"
To disable this message for all users, run:
sudo touch /var/lib/cloud/instance/locale-check.skip
-----ubuntu@ip-172-31-19-122:~$
```

Resource and dependency

➡ Example: Modify: Change instance type from script and apply again.

```
ubuntu@ip-172-31-27-186:~$ cd ~/terraform_202003/Workshop_1.3_Resource/01_basic_resource
ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/01_basic_resource$ terraform apply
aws_instance.lab: Refreshing state... [id=i-0725fc7414fcf401a]

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
 ~ update in-place

Terraform will perform the following actions:

# aws_instance.lab will be updated in-place
~ resource "aws_instance" "lab" {
    ami
    arn
    associate_public_ip_address = true
    availability_zone
    cpu_core_count
    cpu_threads_per_core
    disable_api_termination
    ebs_optimized
    get_password_data
    id
    instance_state
    ~ instance_type
    ipv6_address_count
    ipv6_addresses
    key_name
    monitoring
    primary_network_interface_id = "eni-05310d02effc27410"
    private_dns
    private_ip
    public_dns
    public_ip
    security_groups
        "default",
}
```

```
    ] source_dest_check = true
    subnet_id = "subnet-41742436"
    tags = {}
    tenancy = "default"
    volume_tags = {}
    vpc_security_group_ids = [
        "sg-fe2d3c9b",
    ]

    credit_specification {
        cpu_credits = "unlimited"
    }

    root_block_device {
        delete_on_termination = true
        encrypted = false
        iops = 100
        volume_id = "vol-01e22299a2f2baa1f"
        volume_size = 8
        volume_type = "gp2"
    }
}

Plan: 0 to add, 1 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

aws_instance.lab: Modifying... [id=i-0725fc7414fcf401a]
aws_instance.lab: Still modifying... [id=i-0725fc7414fcf401a, 10s elapsed]
aws_instance.lab: Still modifying... [id=i-0725fc7414fcf401a, 20s elapsed]
aws_instance.lab: Modifications complete after 25s [id=i-0725fc7414fcf401a]

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

Outputs:

aws_instance_lab_id = i-0725fc7414fcf401a
aws_instance_lab_keyname = keypair
aws_instance_lab_public_dns = ec2-52-77-210-220.ap-southeast-1.compute.amazonaws.com
aws_instance_lab_public_ip = 52.77.210.220
ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/01_basic_resource$
```



Terraform

Resource and dependency

► Example: **Modify:** Check information from resource's output and cross-check via aws cli

```
ubuntu@ip-172-31-27-186:~$ aws ec2 describe-instances --instance-ids i-0725fc7414fcf401a
{
    "Reservations": [
        {
            "OwnerId": "404075494050",
            "ReservationId": "r-0143cad3e17fbe99",
            "Instances": [
                {
                    "AmiLaunchIndex": 0,
                    "Architecture": "x86_64",
                    "ClientToken": null,
                    "EbsOptimized": false,
                    "EnaSupport": true,
                    "Hypervisor": "xen",
                    "ImageId": "ami-81cecfcd",
                    "InstanceId": "i-0725fc7414fcf401a",
                    "InstanceType": "t2.nano",
                    "KeyName": "keypair",
                    "LaunchTime": "2020-01-19T14:12:03.000Z",
                    "PrivateDnsName": "ip-172-31-19-122.ap-southeast-1.compute.internal",
                    "PrivateIpAddress": "172.31.19.122",
                    "PublicDnsName": "ec2-52-77-210-220.ap-southeast-1.compute.amazonaws.com",
                    "PublicIpAddress": "52.77.210.220",
                    "RootDeviceName": "/dev/sda1",
                    "RootDeviceType": "ebs",
                    "SourceDestCheck": true,
                    "StateTransitionReason": null,
                    "SubnetId": "subnet-41742436",
                    "VirtualizationType": "hvm",
                    "VpcId": "vpc-4483e921"
                }
            ],
            "BlockDeviceMappings": [
                {
                    "DeviceName": "/dev/sda1",
                    "Ebs": {
                        "AttachTime": "2020-01-19T13:59:18.000Z",
                        "DeleteOnTermination": true,
                        "Status": "attached",
                        "VolumeId": "vol-01e22299a2f2ba1ff"
                    }
                }
            ],
            "CapacityReservationSpecification": {
                "CapacityReservationPreference": "open"
            },
            "CpuOptions": {
                "CoreCount": 1,
                "ThreadsPerCore": 1
            },
            "HibernationOptions": {
                "Configured": false
            }
        }
    ]
}
```

```
ubuntu@ip-172-31-27-186:~$ aws ec2 describe-instances --instance-ids i-0725fc7414fcf401a
{
    "Reservations": [
        {
            "OwnerId": "404075494050",
            "ReservationId": "r-0143cad3e17fbe99",
            "Instances": [
                {
                    "MetadataOptions": {
                        "HttpEndpoint": "enabled",
                        "HttpPutResponseHopLimit": 1,
                        "HttpTokens": "optional",
                        "State": "applied"
                    },
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "NetworkInterfaces": [
                        {
                            "Description": "interface",
                            "InterfaceType": "ena-05310d02effc27410",
                            "MacAddress": "06:62:d8:ab:53:a8",
                            "NetworkInterfaceId": "eni-05310d02effc27410",
                            "OwnerId": "404075494050",
                            "PrivateDnsName": "ip-172-31-19-122.ap-southeast-1.compute.internal",
                            "PrivateIpAddress": "172.31.19.122",
                            "SourceDestCheck": true,
                            "Status": "in-use",
                            "SubnetId": "subnet-41742436",
                            "VpcId": "vpc-4483e921"
                        }
                    ],
                    "Association": {
                        "IpOwnerId": "amazon",
                        "PublicDnsName": "ec2-52-77-210-220.ap-southeast-1.compute.amazonaws.com",
                        "PublicIp": "52.77.210.220"
                    },
                    "Attachment": {
                        "AttachTime": "2020-01-19T13:59:18.000Z",
                        "AttachmentId": "eni-attach-02af6f2e9282f5e63",
                        "DeleteOnTermination": true,
                        "DeviceIndex": 0,
                        "Status": "attached"
                    },
                    "Groups": [
                        {
                            "GroupId": "sg-fe2d3c9b",
                            "GroupName": "default"
                        }
                    ],
                    "PrivateIpAddresses": [
                        {
                            "Primary": true,
                            "PrivateDnsName": "ip-172-31-19-122.ap-southeast-1.compute.internal",
                            "PrivateIpAddress": "172.31.19.122"
                        }
                    ],
                    "Association": {
                        "IpOwnerId": "amazon",
                        "PublicDnsName": "ec2-52-77-210-220.ap-southeast-1.compute.amazonaws.com",
                        "PublicIp": "52.77.210.220"
                    }
                }
            ]
        }
    ]
}
```

```
ubuntu@ip-172-31-27-186:~$ aws ec2 describe-instances --instance-ids i-0725fc7414fcf401a
{
    "Reservations": [
        {
            "OwnerId": "404075494050",
            "ReservationId": "r-0143cad3e17fbe99",
            "Instances": [
                {
                    "Placement": {
                        "AvailabilityZone": "ap-southeast-1a",
                        "GroupName": null,
                        "Tenancy": "default"
                    },
                    "SecurityGroups": [
                        {
                            "GroupId": "sg-fe2d3c9b",
                            "GroupName": "default"
                        }
                    ],
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    }
                }
            ]
        }
    ]
}
```

Resource and dependency

- ▶ **Example: Modify:** Test connect via ssh to new EC2 machine

```
[praparn-MacBook-Pro:~] praparn$ ssh -i terraformlab ubuntu@52.77.210.220
The authenticity of host '52.77.210.220 (52.77.210.220)' can't be established.
ECDSA key fingerprint is SHA256:DWD3aGau4ZLaHKq23o5cxCKUk0VJvnMD+NsXstneLqk.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '52.77.210.220' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-1060-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 Get cloud support with Ubuntu Advantage Cloud Guest:
 http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

New release '18.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Jan 19 14:06:59 2020 from 184.22.242.147
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

[ubuntu@ip-172-31-19-122:~]$ lsb_release
No LSB modules are available.
[ubuntu@ip-172-31-19-122:~]$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 16.04.4 LTS
Release:        16.04
Codename:      xenial
ubuntu@ip-172-31-19-122:~$ ]
```

Resource and dependency

➡ Example: Destroy: Delete EC2 by terraform script

```
ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/01_basic_resource$ terraform destroy
aws_instance.lab: Refreshing state... [id=i-0725fc7414fcf401a]

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
- destroy

Terraform will perform the following actions:

# aws_instance.lab will be destroyed
- resource "aws_instance" "lab" {
    - ami
    - arn
    - associate_public_ip_address
    - availability_zone
    - cpu_core_count
    - cpu_threads_per_core
    - disable_api_termination
    - ebs_optimized
    - get_password_data
    - id
    - instance_state
    - instance_type
    - ipv6_address_count
    - ipv6_addresses
    - key_name
    - monitoring
    - primary_network_interface_id
    - private_dns
    - private_ip
    - public_dns
    - public_ip
    - security_groups
    - source_dest_check
    - subnet_id
    - tags
    - tenancy
    - volume_tags
    - vpc_security_group_ids
}
```

```
- credit_specification {
  - cpu_credits = "unlimited" -> null
}

- root_block_device {
  - delete_on_termination = true -> null
  - encrypted = false -> null
  - iops = 100 -> null
  - volume_id = "vol-01e22299a2f2baa1f" -> null
  - volume_size = 8 -> null
  - volume_type = "gp2" -> 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

aws_instance.lab: Destroying... [id=i-0725fc7414fcf401a]
aws_instance.lab: Still destroying... [id=i-0725fc7414fcf401a, 10s elapsed]
aws_instance.lab: Still destroying... [id=i-0725fc7414fcf401a, 20s elapsed]
aws_instance.lab: Destruction complete after 30s

Destroy complete! Resources: 1 destroyed.
ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/01_basic_resource$
```

Resource and dependency

► Example: Destroy: Verify instance by aws cli

DescribeInstances		
Reservations		
OwnerId	404075494050	
ReservationId	r-0143ecad3e17fbe99	
Instances		
AmiLaunchIndex	0	
Architecture	x86_64	
ClientToken		
EbsOptimized	False	
EnaSupport	True	
Hypervisor	xen	
ImageId	ami-81cefccf	
InstanceId	i-0725fc7414fcf401a	
InstanceType	t2.nano	
KeyName	keypair	
LaunchTime	2020-01-19T14:12:03.000Z	
PrivateDnsName		
PublicDnsName		
RootDeviceName	/dev/sda1	
RootDeviceType	ebs	
StateTransitionReason	User initiated (2020-01-19 14:31:06 GMT)	
VirtualizationType	hvm	
CapacityReservationSpecification		
CapacityReservationPreference	open	
CpuOptions		
CoreCount	1	
ThreadsPerCore	1	
HibernationOptions		
Configured	False	

MetadataOptions		
HttpEndpoint		enabled
HttpPutResponseHopLimit	1	
HttpTokens	optional	
State	pending	
Monitoring		
State	disabled	
Placement		
AvailabilityZone	ap-southeast-1a	
GroupName		
Tenancy	default	
State		
Code	48	
Name	terminated	
StateReason		
Code	Client.UserInitiatedShutdown	
Message	Client.UserInitiatedShutdown: User initiated shutdown	

Workshop: Basic Resource



Workshop: Basic Resource

```
ubuntu@ip-172-31-27-186:~$ aws configure
AWS Access Key ID [*****06WZ*]:
AWS Secret Access Key [*****117*]:
Default region name [ap-southeast-1]:
Default output format [json]: table
ubuntu@ip-172-31-27-186:~$ aws ec2 describe-instances --instance-ids i-0725fc7414fcf401a

          DescribeInstances
+-----+
| Reservations
+-----+
| OwnerId      | 404075494050
| ReservationId| r-0143ecad3e17fbe99
+-----+
| Instances
+-----+
| AmiLaunchIndex | 0
| Architecture   | x86_64
| ClientToken    |
| EbsOptimized   | False
| EnaSupport     | True
| Hypervisor     | xen
| ImageId        | ami-81cefcfd
| InstanceId     | i-0725fc7414fcf401a
| InstanceType   | t3a.nano
| KeyName        | keypair
| LaunchTime     | 2020-01-19T13:59:18.000Z
| PrivateDnsName| ip-172-31-19-122.ap-southeast-1.compute.internal
| PrivateIpAddress| 172.31.19.122
| PublicCnName   | ec2-13-229-47-208.ap-southeast-1.compute.amazonaws.com
| PublicIpAddress| 13.229.47.208
| RootDeviceName | /dev/sda1
| RootDeviceType | ebs
| SourceDestCheck| True
| StateTransitionReason|
| SubnetId       | subnet-41742436
| VirtualizationType| hvm
| VpcId          | vpc-4483e921
+-----+
| BlockDeviceMappings
+-----+
| DeviceName    | /dev/sda1
+-----+
| Ebs
+-----+
| AttachTime    | 2020-01-19T13:59:18.000Z
| DeleteOnTermination| True
| Status        | attached
| VolumeId      | vol-01e2299a2f2ba1f
+-----+
| CapacityReservationSpecification
+-----+
| CapacityReservationPreference| open
+-----+
| CpuOptions
+-----+
| CoreCount     | 1
| ThreadsPerCore| 2
+-----+
| HibernationOptions
+-----+
| Configured    | False
+-----+
```

```
          MetadataOptions
+-----+
| HttpEndpoint | enabled
| HttpPutResponseHopLimit| 1
| HttpTokens   | optional
| State        | applied
+-----+
| Monitoring
+-----+
| State        | disabled
+-----+
| NetworkInterfaces
+-----+
| Description   |
| InterfaceType| interface
| MacAddress   | 06:62:08:ab:53:aa
| NetworkInterfaceId| eni-05310d02effc27410
| OwnerId      | 404075494050
| PrivateDnsName| ip-172-31-19-122.ap-southeast-1.compute.internal
| PrivateIpAddress| 172.31.19.122
| SourceDestCheck| True
| Status        | in-use
| SubnetId     | subnet-41742436
| VpcId         | vpc-4483e921
+-----+
| Association
+-----+
| IpOwnerId    | amazon
| PublicCnName | ec2-13-229-47-208.ap-southeast-1.compute.amazonaws.com
| PublicIp     | 13.229.47.208
+-----+
| Attachment
+-----+
| AttachTime   | 2020-01-19T13:59:18.000Z
| AttachmentId| eni-attach-02af6f2e9282f5e63
| DeleteOnTermination| True
| DeviceIndex  | 0
| Status       | attached
+-----+
| Groups
+-----+
| GroupId      | sg-fed2d3c9b
| GroupName    | default
+-----+
| PrivateIpAddresses
+-----+
| Primary      | True
| PrivateDnsName| ip-172-31-19-122.ap-southeast-1.compute.internal
| PrivateIpAddress| 172.31.19.122
+-----+
| Association
+-----+
| IpOwnerId    | amazon
| PublicCnName | ec2-13-229-47-208.ap-southeast-1.compute.amazonaws.com
| PublicIp     | 13.229.47.208
+-----+
| Placement
+-----+
| AvailabilityZone| ap-southeast-1a
| GroupName    | default
| Tenancy       | default
+-----+
```

```
          SecurityGroups
+-----+
| GroupId      | sg-fed2d3c9b
| GroupName    | default
+-----+
| State
+-----+
| Code         | 16
| Name         | running
+-----+
```

Q&A



Resource and dependency

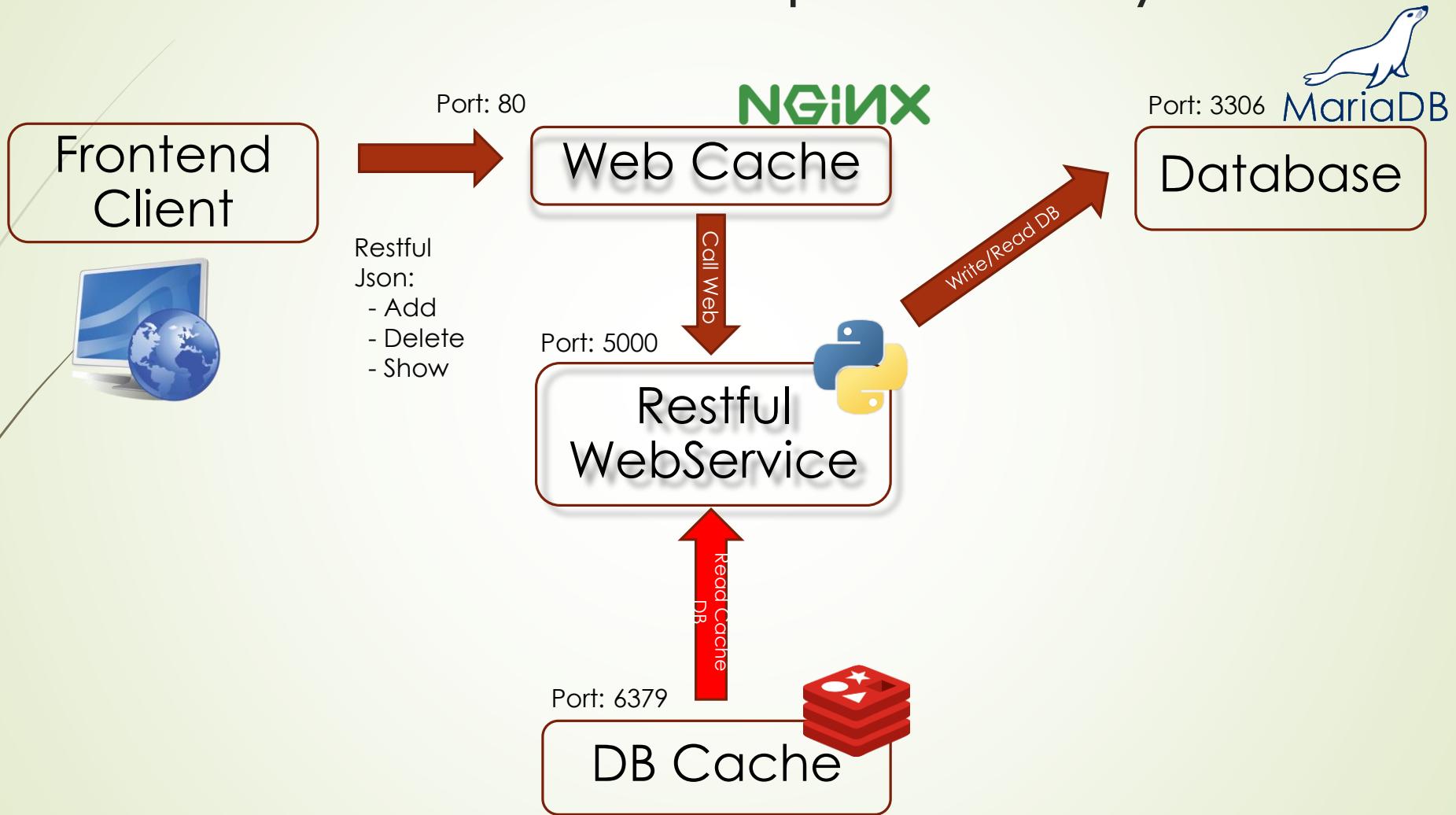
- ▶ **Resource dependency**
- ▶ In actual provision environment. We also need multiple resource to create for fulfill environment. (Ex: Server, Storage, DNS record, Database etc)
- ▶ If we have multiple resource/resource type/provider defined on same workspace. Normally terraform will create all resource parallel.
- ▶ Anyway in many case a set that our resource have some dependency with other resource



Resource and dependency

- ▶ **Resource dependency**
- ▶ Example: General WebServer
 - ▶ Web server for housing rest api
 - ▶ Database server for housing RDBMS
 - ▶ CacheDB server for cache database information
 - ▶ Cache server for cache static content from webserver
 - ▶ Dns record need information of cache server
 - ▶ Web server need information of database server and cachedb server

Resource and Dependency





Resource and dependency

- ▶ **implicit dependency**
- ▶ Terraform support to define argument in resource by reference from other resource in same workspace
- ▶ When we defined argument by referencing. Terraform will automatically detect dependency's chain between resource
- ▶ So it will order list to create resource by dependency's chain by default
- ▶ We call this dependency as “implicit dependency”

Resource and dependency

► implicit dependency

```
1  resource "aws_volume_attachment" "ebs_att" {
2    device_name = "/dev/sdh"
3    volume_id   = "${aws_ebs_volume.labebs.id}"
4    instance_id = "${aws_instance.lab.id}"
5  }
6
7  1 references
8  resource "aws_instance" "lab" {
9    ami           = "ami-21f78e11"
10   availability_zone = "us-west-2a"
11   instance_type     = "t1.micro"
12
13   tags = {
14     Name = "HelloWorld"
15   }
16
17  1 references
18  resource "aws_ebs_volume" "labebs" [
19    availability_zone = "ap-southeast-1c"
20    size             = 10
21  ]
```

Resource and dependency

- ▶ **Example:** Create multiple component with dependency (Implicit)
 - ▶ Requirement:
 - ▶ Need to create instance server from image (ami): Ubuntu server 18.04 last build of Jan,2020
 - ▶ This instance server will have base disk for "/" about 10 GB disk
 - ▶ Additional disk for keep data is about 10 GB disk
 - ▶ Add Tag for input description

Resource and dependency

- ▶ **Example:** Create multiple component with dependency (Implicit)
 - ▶ Tag Description:
 - ▶ Environment = <rd/dev/sit/uat/stag/mot/prd/other>
 - ▶ Region = ap-southeast-1
 - ▶ AZ = <ap-southeast-1a/ap-southeast-1b/ap-southeast-1c/none>
 - ▶ Category = <network/iam/compute/storage>
 - ▶ Name = XXXX
 - ▶ Zone = <public/private/none>
 - ▶ Module = <module name/none>
 - ▶ Billing = <project name / bau / rd>

Resource and dependency

- ▶ **Example:** Create multiple component with dependency (Implicit)
 - ▶ Brake Down
 - ▶ AMI-ID: data source: aws_ami
 - ▶ EC2: resource: aws_instance
 - ▶ Get ami-id from "aws_ami"
 - ▶ Configure disk storage from block device
 - ▶ Configure tag as request
 - ▶ EBS disk: resource: aws_ebs_volume
 - ▶ Attach disk: resource: aws_volume_attachment

Resource and Dependency

Data Source:

aws_ami

ami-id

Resource:

aws_instance

block storage: "/" (10 GB)

Tag ec2 by standard tag

Attach Volume

Resource:

aws_ebs_volume

block storage: (10 GB)

Resource:

aws_volume_attachment

device name: xxxxxxxx

instance id: xxxxxxxx

ebs volume id: xxxxxxxx

Resource and dependency

► **Example:** Target AMI: (ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-202001**

The screenshot shows the AWS EC2 console interface. On the left, there's a navigation sidebar with sections for Instances, Images, and Elastic Block Store. The main area has tabs for Launch and Actions, with Launch selected. A search bar at the top right shows the query "ubuntu/images/hvm-ssd/ubuntu-bionic-18....". Below the search is a table of AMIs, with one row highlighted. At the bottom, there's a detailed view of the selected AMI.

Name	AMI Name	AMI ID	Source	Owner
ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20200112	ami-09a4a9ce71ff3f20b	099720109477/ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20200112	099720109477	F
ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20191113-3b73ef...	ami-02c4c96e0a37b397f	aws-marketplace/ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20191113	67959333241	F
ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20191113	ami-07febfb4080320e	099720109477/ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20191113	099720109477	F
ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20191021	ami-0c199cae95cea87f0	099720109477/ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20191021	099720109477	F
ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20191010	ami-0bba750772355e50d	099720109477/ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20191010	099720109477	F

Image: ami-09a4a9ce71ff3f20b

Details	Tags
AMI ID: ami-09a4a9ce71ff3f20b Owner: 099720109477	AMI Name: ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20200112 Source: 099720109477/ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20200112
Status: available	State Reason: -
Creation date: January 15, 2020 at 7:31:04 AM UTC+7	Platform: Ubuntu
Architecture: x86_64	Image Type: machine
Virtualization type: hvm	Description: Canonical, Ubuntu, 18.04 LTS, amd64 bionic image build on 2020-01-12
Root Device Name: /dev/sda1	Root Device Type: ebs
RAM disk ID: -	Kernel ID: -
Product Codes: -	Block Devices: /dev/sda1=snap-08016d4ab116eb784:8:true:gp2, /dev/sdb=ephemeral0,

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Resource and dependency

► Example: Target AMI (aws_ami):



Terraform

Intro Learn Docs Community Enterprise Download GitHub Sign In

Data Source: aws_ami

JUMP TO SECTION ▾

- EXPAND ALL
- FILTER
- All Providers
- AWS Provider
- Guides
 - AWS Provider Version 2 Upgrade
 - AWS Provider Version 3 Upgrade
 - Custom Service Endpoints
 - AWS Provider Track on HashiCorp Learn
- Provider Data Sources
- Access Analyzer
- ACM
- ACM PCA
- API Gateway
- Application Autoscaling
- AppMesh
- AppSync
- Athena
- Autoscaling
- Backup

Example Usage

```
data "aws_ami" "example" {  
  executable_users = ["self"]  
  most_recent     = true  
  name_regex      = "^myami-\d{3}"  
  owners          = ["self"]  
  
  filter {  
    name  = "name"  
    values = ["myami-*"]  
  }  
  
  filter {  
    name  = "root-device-type"  
    values = ["ebs"]  
  }  
  
  filter {  
    name  = "virtualization-type"  
    values = ["hvm"]  
  }  
}
```

Argument Reference

- `owners` - (Required) List of AMI owners to limit search. At least 1 value must be specified. Valid values: an AWS account ID, `self` (the current account), or an AWS owner alias (e.g. `amazon`, `aws-marketplace`, `microsoft`).
- `most_recent` - (Optional) If more than one result is returned, use the most recent AMI.
- `executable_users` - (Optional) Limit search to users with *explicit* launch permission on the image. Valid items are the numeric account ID or `self`.
- `filter` - (Optional) One or more name/value pairs to filter off of. There are several valid keys, for a full reference, check out [describe-images in the AWS CLI reference](#).
- `name_regex` - (Optional) A regex string to apply to the AMI list returned by AWS. This allows more advanced filtering not supported from the AWS API. This filtering is done locally on what AWS returns, and could have a performance impact if the result is large. It is recommended to combine this with other options to narrow down the list AWS returns.

NOTE: If more or less than a single match is returned by the search, Terraform will fail. Ensure that your search is specific enough to return a single AMI ID only, or use `most_recent` to choose the most recent one. If you want to match multiple AMIs, use the `aws_ami_ids` data source instead.

Resource and dependency

► Example: Target AMI:

```
0 references
1 data "aws_ami" "ubuntu" {
2   most_recent = true
3   filter {
4     name    = "architecture"
5     values  = ["x86_64"]
6   }
7   filter {
8     name    = "name"
9     values  = ["ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-202001*"]
10  }
11  filter {
12    name    = "virtualization-type"
13    values  = ["hvm"]
14  }
15  filter {
16    name    = "root-device-type"
17    values  = ["ebs"]
18  }
19  owners = ["099720109477"] # Recommend Image of AWS
20 }
```

Resource and dependency

► Example: EC2 Resource (aws_instance)

Argument Reference

The following arguments are supported:

- `ami` - (Required) The AMI to use for the instance.
- `availability_zone` - (Optional) The AZ to start the instance in.
- `tags` - (Optional) A mapping of tags to assign to the resource.
- `volume_tags` - (Optional) A mapping of tags to assign to the devices created by the instance at launch time.
- `root_block_device` - (Optional) Customize details about the root block device of the instance. See [Block Devices](#) below for details.

Block devices

Each of the `*_block_device` attributes control a portion of the AWS Instance's "Block Device Mapping". It's a good idea to familiarize yourself with [AWS's Block Device Mapping docs](#) to understand the implications of using these attributes.

The `root_block_device` mapping supports the following:

- `volume_type` - (Optional) The type of volume. Can be `"standard"`, `"gp2"`, `"io1"`, `"sc1"`, or `"st1"`. (Default: `"standard"`).
- `volume_size` - (Optional) The size of the volume in gibibytes (GiB).
- `iops` - (Optional) The amount of provisioned IOPS. This is only valid for `volume_type` of `"io1"`, and must be specified if using that type
- `delete_on_termination` - (Optional) Whether the volume should be destroyed on instance termination (Default: `true`).
- `encrypted` - (Optional) Enable volume encryption. (Default: `false`). Must be configured to perform drift detection.
- `kms_key_id` - (Optional) Amazon Resource Name (ARN) of the KMS Key to use when encrypting the volume. Must be configured to perform drift detection.

Resource and dependency

► Example: EC2 Resource

Volume Characteristics

The following table describes the use cases and performance characteristics for each volume type. The default volume type is General Purpose SSD (gp2).

	Solid-State Drives (SSD)		Hard Disk Drives (HDD)	
Volume Type	General Purpose SSD (gp2)	Provisioned IOPS SSD (io1)	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Description	General purpose SSD volume that balances price and performance for a wide variety of workloads	Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads	Low-cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads
Use Cases	<ul style="list-style-type: none">Recommended for most workloadsSystem boot volumesVirtual desktopsLow-latency interactive appsDevelopment and test environments	<ul style="list-style-type: none">Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/s of throughput per volumeLarge database workloads, such as:<ul style="list-style-type: none">MongoDBCassandraMicrosoft SQL ServerMySQLPostgreSQLOracle	<ul style="list-style-type: none">Streaming workloads requiring consistent, fast throughput at a low priceBig dataData warehousesLog processingCannot be a boot volume	<ul style="list-style-type: none">Throughput-oriented storage for large volumes of data that is infrequently accessedScenarios where the lowest storage cost is importantCannot be a boot volume
API Name	gp2	io1	st1	sc1

Resource and dependency

► Example: EBS Disk (aws_ebs_volume)

The screenshot shows the Terraform documentation for the `aws_ebs_volume` resource. The page has a purple header with the Terraform logo and navigation links for Intro, Learn, Docs, Community, Enterprise, Download, GitHub, and Sign In. The main content area has a title "Resource: aws_ebs_volume" and a subtitle "Manages a single EBS volume." On the left is a sidebar with a "GUIDES" section containing links to AWS Provider Version 2 Upgrade, AWS Provider Version 3 Upgrade, Custom Service Endpoints, AWS Provider Track on HashiCorp Learn, and several other provider-specific guides like AppMesh, AppSync, and CloudFront.

Example Usage

```
resource "aws_ebs_volume" "example" {
  availability_zone = "us-west-2a"
  size              = 40

  tags = {
    Name = "HelloWorld"
  }
}
```

NOTE: One of `size` or `snapshot_id` is required when specifying an EBS volume

Argument Reference

The following arguments are supported:

- `availability_zone` - (Required) The AZ where the EBS volume will exist.
- `encrypted` - (Optional) If true, the disk will be encrypted.
- `iops` - (Optional) The amount of IOPS to provision for the disk.
- `size` - (Optional) The size of the drive in GiBs.
- `snapshot_id` - (Optional) A snapshot to base the EBS volume off of.
- `type` - (Optional) The type of EBS volume. Can be "standard", "gp2", "io1", "sc1" or "st1" (Default: "standard").

Resource and dependency

► Example: Attach Disk (aws_volume_attachment)

The screenshot shows the Terraform documentation for the `aws_volume_attachment` resource. The page has a purple header with the Terraform logo and navigation links for Intro, Learn, Docs, Community, Enterprise, Download, GitHub, and Sign In. The main content area has a title "Resource: aws_volume_attachment" and a description: "Provides an AWS EBS Volume Attachment as a top level resource, to attach and detach volumes from AWS Instances." Below this is a note about EBS block devices: "NOTE on EBS block devices: If you use `ebs_block_device` on an `aws_instance`, Terraform will assume management over the full set of non-root EBS block devices for the instance, and treats additional block devices as drift. For this reason, `ebs_block_device` cannot be mixed with external `aws_ebs_volume` + `aws_ebs_volume_attachment` resources for a given instance." At the bottom, there is a section titled "Example Usage" containing Terraform code:

```
resource "aws_volume_attachment" "ebs_att" {
  device_name = "/dev/sdh"
  volume_id   = "${aws_ebs_volume.example.id}"
  instance_id = "${aws_instance.web.id}"
}

resource "aws_instance" "web" {
  ami           = "ami-21f78e11"
  availability_zone = "us-west-2a"
  instance_type = "t1.micro"

  tags = {
    Name = "HelloWorld"
  }
}

resource "aws_ebs_volume" "example" {
  availability_zone = "us-west-2a"
  size             = 1
}
```

Resource and dependency

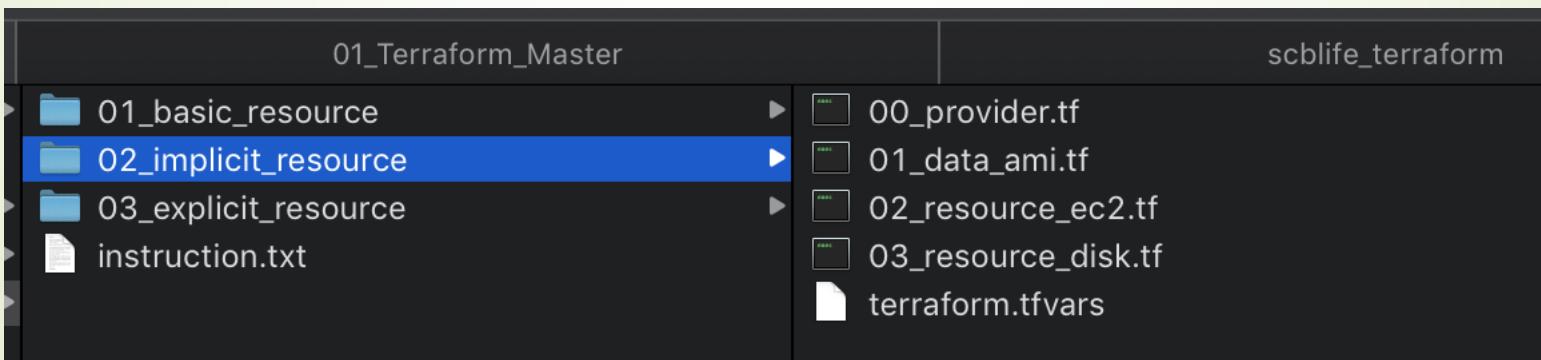
► Example: EBS Disk and Attach

```
1  resource "aws_ebs_volume" "labebs" {
2    availability_zone = "ap-southeast-1c"
3    size              = "10"
4    type              = "gp2"
5
6    tags = {
7      Environment = "rd"
8      Region     = "ap-southeast-1"
9      AZ          = "ap-southeast-1c"
10     Categories  = "storage"
11     Name        = "EBSlabresource-student-X"
12     Zone        = "public"
13     Module      = "ebs"
14     Billing     = "terraform-workshop"
15   }
16 }
17
18 resource "aws_volume_attachment" "labebsattach" {
19   device_name = "/dev/sdf"
20   volume_id   = aws_ebs_volume.labebs.id
21   instance_id = aws_instance.lab.id
22 }
23
24 output "aws_ebs_volume_labebs_id" {
25   value = aws_ebs_volume.labebs.id
26 }
```

Resource and dependency

► **Example:** From brake down. We can create terraform file (*.tf) and manage each module like below

- 00_provider.tf (Provider)
- 01_data_iam.tf (Data → IAM)
- 02_resource_ec2.tf (Resource → EC2)
- 03_resource_disk.tf (Resource → Disk EBS, Attach)



Resource and dependency

► Example: Create: Provision terraform script

```
ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/02_implicit_resource$ terraform init

Initializing the backend...

Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "aws" (hashicorp/aws) 2.45.0...

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.aws: version = "~> 2.45"

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.
ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/02_implicit_resource$     terraform fmt -check
00_provider.tf
02_resource_ec2.tf
ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/02_implicit_resource$     terraform validate
Success! The configuration is valid.

ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/02_implicit_resource$ terraform apply -auto-approve -parallelism=5 -refresh=true ./
```

Resource and dependency

► Example: Create: Provision terraform script

```
[ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/02_implicit_resource$ terraform apply -auto-approve -parallelism=5 -refresh=true ./
data.aws_ami.ubuntu: Refreshing state...
aws_ebs_volume.labebs: Creating...
aws_instance.lab: Creating...
aws_ebs_volume.labebs: Still creating... [10s elapsed]
aws_instance.lab: Still creating... [10s elapsed]
aws_ebs_volume.labebs: Creation complete after 10s [id=vol-06287f1ebc3ee22a1]
aws_instance.lab: Creation complete after 12s [id=i-0586c1cc00fe01f1a]
aws_volume_attachment.labebsattach: Creating...
aws_volume_attachment.labebsattach: Still creating... [10s elapsed]
aws_volume_attachment.labebsattach: Still creating... [20s elapsed]
aws_volume_attachment.labebsattach: Creation complete after 20s [id=vai-2615738051]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Outputs:

aws_ebs_volume_labebs_id = vol-06287f1ebc3ee22a1
aws_instance_lab_id = i-0586c1cc00fe01f1a
aws_instance_lab_keyname = keypair
aws_instance_lab_public_dns = ec2-13-251-129-151.ap-southeast-1.compute.amazonaws.com
aws_instance_lab_public_ip = 13.251.129.151
ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/02_implicit_resource$ █
```

Resource and dependency

► Example: Create: Check information from resource's output and cross-check via aws cli

DescribeInstances		CapacityReservationSpecification		Groups	
Reservations		CapacityReservationPreference		GroupId	
OwnerId		open		sg-fe2d3c9b	
ReservationId		CpuOptions		GroupName	
Instances		CoreCount		default	
AmiLaunchIndex	0	ThreadsPerCore			
Architecture	x86_64				
ClientToken		HibernationOptions			
EbsOptimized	False	Configured			
EnaSupport	True	MetadataOptions			
Hypervisor	xen	HttpEndpoint			
ImageId	ami-09a4a9ce71ff3f20b	HttpPutResponseHopLimit			
InstanceId	i-0586c1cc0fe0f1fa	HttpTokens			
InstanceType	t3a.nano	State			
KeyName	keypair				
LaunchTime	2020-01-19T15:18:52.000Z	Monitoring			
PrivateDnsName	ip-172-31-33-132.ap-southeast-1.compute.internal	State			
PrivateIpAddress	172.31.33.132	disabled			
PublicDnsName	ec2-13-251-129-151.ap-southeast-1.compute.amazonaws.com	NetworkInterfaces			
PublicIpAddress	13.251.129.151	Description			
RootDeviceName	/dev/sda1	InterfaceType			
RootDeviceType	efs	MacAddress			
SourceDestCheck	True	NetworkInterfaceId			
StateTransitionReason		OwnerId			
SubnetId	subnet-5a1e2c1c	PrivateDnsName			
VirtualizationType	hvm	PrivateIpAddress			
VpcId	vpc-4483e921	SourceDestCheck			
BlockDeviceMappings		Status			
DeviceName		SubnetId			
/dev/sda1		VpcId			
Ebs					
AttachTime	2020-01-19T15:18:53.000Z	Association			
DeleteOnTermination	True	IpOwnerId			
Status	attached	PublicDnsName			
VolumeId	vol-00a9aa56bd992baa0	PublicIp			
BlockDeviceMappings		Attachment			
DeviceName		AttachTime			
/dev/sdf		AttachmentId			
Ebs		DeleteOnTermination			
AttachTime	2020-01-19T15:19:13.000Z	DeviceIndex			
DeleteOnTermination	False	Status			
Status	attached				
VolumeId	vol-06287f1ebc3ee22a1				
Tags		Key		Value	
		Categories		compute	
		Billing		terraform-workshop	
		Module		ec2	
		Name		labresource-student-35	
		Environment		rd	
		Zone		public	
		Region		ap-southeast-1	
		AZ		ap-southeast-1c	

ubuntu@ip-172-31-27-186:~/terraform 202003/Workshop_1.3 Resource/02_implicit_resource\$

Resource and dependency

► **Example: Create:** Check information from resource's output and cross-check via aws cli

```
ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/02_implicit_resource$ aws ec2 describe-volumes --region ap-southeast-1 --filters Name=attachment.instance-id,Values=i-0586c1cc00fe01f1a
DescribeVolumes
+-----+
| Volumes
+-----+
| AvailabilityZone | CreateTime | Encrypted | Iops | Size | SnapshotId | State | VolumeId | VolumeType |
| ap-southeast-1c | 2020-01-19T15:18:51.827Z | False | 100 | 20 | null | in-use | vol-06287f1ebc3ee22a1 | gp2 |
+-----+
Attachments
+-----+
| AttachTime | DeleteOnTermination | Device | InstanceId | State | VolumeId |
| 2020-01-19T15:19:13.000Z | False | /dev/sdf | i-0586c1cc00fe01f1a | attached | vol-06287f1ebc3ee22a1 |
+-----+
Tags
+-----+
| Key | Value |
| Environment | rd |
| Zone | public |
| Billing | terraform-workshop |
| Name | EBSlabresource-student-35 |
| AZ | ap-southeast-1c |
| Region | ap-southeast-1 |
| Categories | storage |
| Module | ebs |
+-----+
Volumes
+-----+
| AvailabilityZone | CreateTime | Encrypted | Iops | Size | SnapshotId | State | VolumeId | VolumeType |
| ap-southeast-1c | 2020-01-19T15:18:53.287Z | False | 100 | 10 | snap-08016d4ab116eb784 | in-use | vol-00a9aa56bd992baa0 | gp2 |
+-----+
Attachments
+-----+
| AttachTime | DeleteOnTermination | Device | InstanceId | State | VolumeId |
| 2020-01-19T15:18:53.000Z | True | /dev/sda1 | i-0586c1cc00fe01f1a | attached | vol-00a9aa56bd992baa0 |
+-----+
ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/02_implicit_resource$
```

Resource and dependency

► Example: Create: Test connect via ssh to new EC2 machine and list disk

```
[praparns-MacBook-Pro:~ ssh praparn$ ssh -i terraformlab ubuntu@13.251.129.151
The authenticity of host '13.251.129.151 (13.251.129.151)' can't be established.
ECDSA key fingerprint is SHA256:WJuH4N+YN+Vn/7/YqwD70/d1XStJXBFI+FSACJCy1wM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '13.251.129.151' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1057-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 System information as of Sun Jan 19 15:31:51 UTC 2020

 System load:  0.03           Processes:      92
 Usage of /:   11.0% of 9.63GB Users logged in:  0
 Memory usage: 32%
 Swap usage:   0%
 IP address for ens5: 172.31.33.132

0 packages can be updated.
0 updates are security updates.

Last login: Sun Jan 19 15:31:22 2020 from 184.22.242.147
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-33-132:~$ ]
```

```
[ubuntu@ip-172-31-33-132:~$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0    7:0    0 89.1M  1 loop /snap/core/8268
loop1    7:1    0 18M   1 loop /snap/amazon-ssm-agent/1480
nvme0n1  259:0  0 10G   0 disk 
└─nvme0n1p1 259:1  0 10G   0 part /
nvme1n1  259:2  0 20G   0 disk
[ubuntu@ip-172-31-33-132:~$ sudo file -s /dev/nvme1n1
/dev/nvme1n1: data
ubuntu@ip-172-31-33-132:~$ sudo mkfs -t xfs /dev/nvme1n1
meta-data=/dev/nvme1n1          isize=512    agcount=4, agsize=1310720 blks
                           = sectsz=512  attr=2, projid32bit=1
                           =         crc=1    finobt=1, sparse=0, rmapbt=0, reflink=0
data     =               bsize=4096  blocks=5242880, imaxpct=25
                           =         sunit=0  swidth=0 blks
naming   =version 2            bsize=4096  ascii-ci=0 ftype=1
log      =internal log         bsize=4096  blocks=2560, version=2
                           =         sectsz=512  sunit=0 blks, lazy-count=1
realtime =none                extsz=4096  blocks=0, rtextents=0
ubuntu@ip-172-31-33-132:~$ sudo mkdir /additionalstorage
ubuntu@ip-172-31-33-132:~$ sudo mount /dev/nvme1n1 /additionalstorage
[ubuntu@ip-172-31-33-132:~$ df -kh
Filesystem      Size  Used Avail Use% Mounted on
udev            219M    0  219M  0% /dev
tmpfs           47M  736K  46M  2% /run
/dev/nvme0n1p1  9.7G  1.1G  8.6G 12% /
tmpfs           231M    0  231M  0% /dev/shm
tmpfs           5.0M    0  5.0M  0% /run/lock
tmpfs           231M    0  231M  0% /sys/fs/cgroup
/dev/loop0       90M   90M    0 100% /snap/core/8268
/dev/loop1       18M   18M    0 100% /snap/amazon-ssm-agent/1480
tmpfs           47M    0  47M  0% /run/user/1000
/dev/nvme1n1    20G   53M  20G  1% /additionalstorage
ubuntu@ip-172-31-33-132:~$ ]
```

Resource and dependency

➡ Example: Destroy: Delete EC2/EBS by terraform script

```
[ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/02_implicit_resource]$ terraform destroy -auto-approve
aws_ebs_volume.labebs: Refreshing state... [id=vol-06287f1ebc3ee22a1]
data.aws_ami.ubuntu: Refreshing state...
aws_instance.lab: Refreshing state... [id=i-0586c1cc00fe01f1a]
aws_volume_attachment.labebsattach: Refreshing state... [id=vai-2615738051]
aws_volume_attachment.labebsattach: Destroying... [id=vai-2615738051]
aws_volume_attachment.labebsattach: Still destroying... [id=vai-2615738051, 10s elapsed]
aws_volume_attachment.labebsattach: Destruction complete after 10s
aws_ebs_volume.labebs: Destroying... [id=vol-06287f1ebc3ee22a1]
aws_instance.lab: Destroying... [id=i-0586c1cc00fe01f1a]
aws_ebs_volume.labebs: Destruction complete after 1s
aws_instance.lab: Still destroying... [id=i-0586c1cc00fe01f1a, 10s elapsed]
aws_instance.lab: Still destroying... [id=i-0586c1cc00fe01f1a, 20s elapsed]
aws_instance.lab: Still destroying... [id=i-0586c1cc00fe01f1a, 30s elapsed]
aws_instance.lab: Still destroying... [id=i-0586c1cc00fe01f1a, 40s elapsed]
aws_instance.lab: Destruction complete after 50s

Destroy complete! Resources: 3 destroyed.
ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/02_implicit_resource$ █
```

Resource and dependency

➡ Example: Destroy: Delete EC2/EBS by terraform script

DescribeInstances	
Reservations	
OwnerId	404075494050
ReservationId	r-0331ad8e1c63c0512
Instances	
AmiLaunchIndex	0
Architecture	x86_64
ClientToken	
EbsOptimized	False
EnaSupport	True
Hyperervisor	xen
ImageId	ami-09a4a9ce71ff3f20b
InstanceId	i-058cc1cc0fe01f1a
InstanceType	t3a.nano
KeyName	keypair
LaunchTime	2020-01-19T15:18:52.000Z
PrivateDnsName	
PublicDnsName	
RootDeviceName	/dev/sda1
RootDeviceType	ebs
StateTransitionReason	User initiated (2020-01-19 15:36:19 GMT)
VirtualizationType	hvm
CapacityReservationSpecification	
CapacityReservationPreference	open
CpuOptions	
CoreCount	1
ThreadsPerCore	2
HibernationOptions	
Configured	False
MetadataOptions	
HttpEndpoint	enabled
HttpPutResponseHoplimit	1
HttpTokens	optional
State	pending
Monitoring	
State	disabled
Placement	
AvailabilityZone	ap-southeast-1c
GroupName	
Tenancy	default
State	
Code	48
Name	terminated

```
[ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/02_implicit_resource$ aws ec2 describe-volumes -
-----
|DescribeVolumes|
+-----+
ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/02_implicit_resource$ █
```

Workshop: Implicit Resource



Workshop: Implicit Resource

```
ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/02_implicit_resource$ aws ec2 describe-volumes --region ap-southeast-1 --filters Name=attachment.instance-id,Values=i-0586c1cc00fe01f1a
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
|       |   DescribeVolumes   |       |       |       |       |       |       |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
|       |       Volumes      |       |       |       |       |       |       |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
| AvailabilityZone | CreateTime | Encrypted | Iops | Size | SnapshotId | State | VolumeId | VolumeType |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ap-southeast-1c | 2020-01-19T15:18:51.827Z | False     | 100  | 20   |           | in-use | vol-06287f1ebc3ee22a1 | gp2    |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
|       |       Attachments    |       |       |       |       |       |       |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
| AttachTime | DeleteOnTermination | Device   | InstanceId | State | VolumeId |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2020-01-19T15:19:13.000Z | False     | /dev/sdf | i-0586c1cc00fe01f1a | attached | vol-06287f1ebc3ee22a1 |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
|       |       Tags          |       |       |       |       |       |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
|       |       Key            |       |       Value        |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Environment | rd             |           |
| Zone        | public          |           |
| Billing     | terraform-workshop |
| Name        | EBSlabresource-student-35 |
| AZ          | ap-southeast-1c   |
| Region      | ap-southeast-1    |
| Categories  | storage          |           |
| Module      | ebs              |           |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
|       |       Volumes      |       |       |       |       |       |       |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
| AvailabilityZone | CreateTime | Encrypted | Iops | Size | SnapshotId | State | VolumeId | VolumeType |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ap-southeast-1c | 2020-01-19T15:18:53.287Z | False     | 100  | 10   | snap-08016d4ab116eb784 | in-use | vol-00a9aa56bd992baa0 | gp2    |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
|       |       Attachments    |       |       |       |       |       |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
| AttachTime | DeleteOnTermination | Device   | InstanceId | State | VolumeId |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2020-01-19T15:18:53.000Z | True      | /dev/sda1 | i-0586c1cc00fe01f1a | attached | vol-00a9aa56bd992baa0 |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/02_implicit_resource$
```

Q&A



Resource and dependency

- ▶ **explicit dependency**
- ▶ Some scenario we need to create some resource that not have argument reference between each other at all. Anyway we need it to create with dependency
 - ▶ Ex: Create server1 before server2
 - ▶ Ex: Create database (RDS) before webserver
 - ▶ For this purpose. Terraform allow us to add syntax: `depends_on = [<resource>]` for operate dependency
 - ▶ We call this dependency as “explicit dependency”

Resource and dependency

► explicit dependency

```
1  resource "aws_s3_bucket" "backupoperation" {
2      bucket = "s3_bucket_terraform_bkk2020_lab<ID>"
3      acl    = "private"
4  }
5
6  resource "aws_instance" "lab" [
7      ami        = "ami-2757f631"
8      instance_type = "t2.micro"
9      depends_on = [aws_s3_bucket.backupoperation]
10 ]
```

Resource and dependency

- **Example:** Create multiple component with dependency (Implicit and Explicit)
 - Requirement:
 - Need to create 2 instance server
 - Instance server no.1: install nginx server as front-end
 - Allow access http/https/ssh from internet
 - Allow access all for private ip address
 - Instance server no.2: install docker for run back-end
 - Allow access only service ssh from internet
 - Allow access all for private ip address
 - Create S3 Bucket after both instance server create done

Resource and dependency

- ▶ **Example:** Create multiple component with dependency (Implicit and Explicit)
 - ▶ Brake Down
 - ▶ Subnet Mask of VPC: data: aws_vpc
 - ▶ Security Group: resource: aws_security_group_rule
 - ▶ Security Group: nginx web server
 - ▶ Security Group: docker server
 - ▶ EC2: resource: aws_instance
 - ▶ Configure automatic script setup for nginx/docker
 - ▶ Configure tag as standard naming
 - ▶ S3 bucket: resource: aws_s3_bucket

Resource and dependency

Resource

aws_security_group_rule1

Ingress Rule:
Allow from x.x.x.x with yyyy

Egress Rule:
Allow from x.x.x.x with yyyy

Resource

aws_security_group_rule2

Ingress Rule:
Allow from x.x.x.x with yyyy

Egress Rule:
Allow from x.x.x.x with yyyy

Security_group_id

Resource:

aws_instance1

User data: <script setup nginx>
Tag ec2 by standard tag

Private subnet

Data:

aws_vpc

Vpc ID: XXX

Private subnet

Security_group_id

Resource:

aws_instance2

User data: <script setup nginx>
Tag ec2 by standard tag

depend_on

depend_on

Resource:

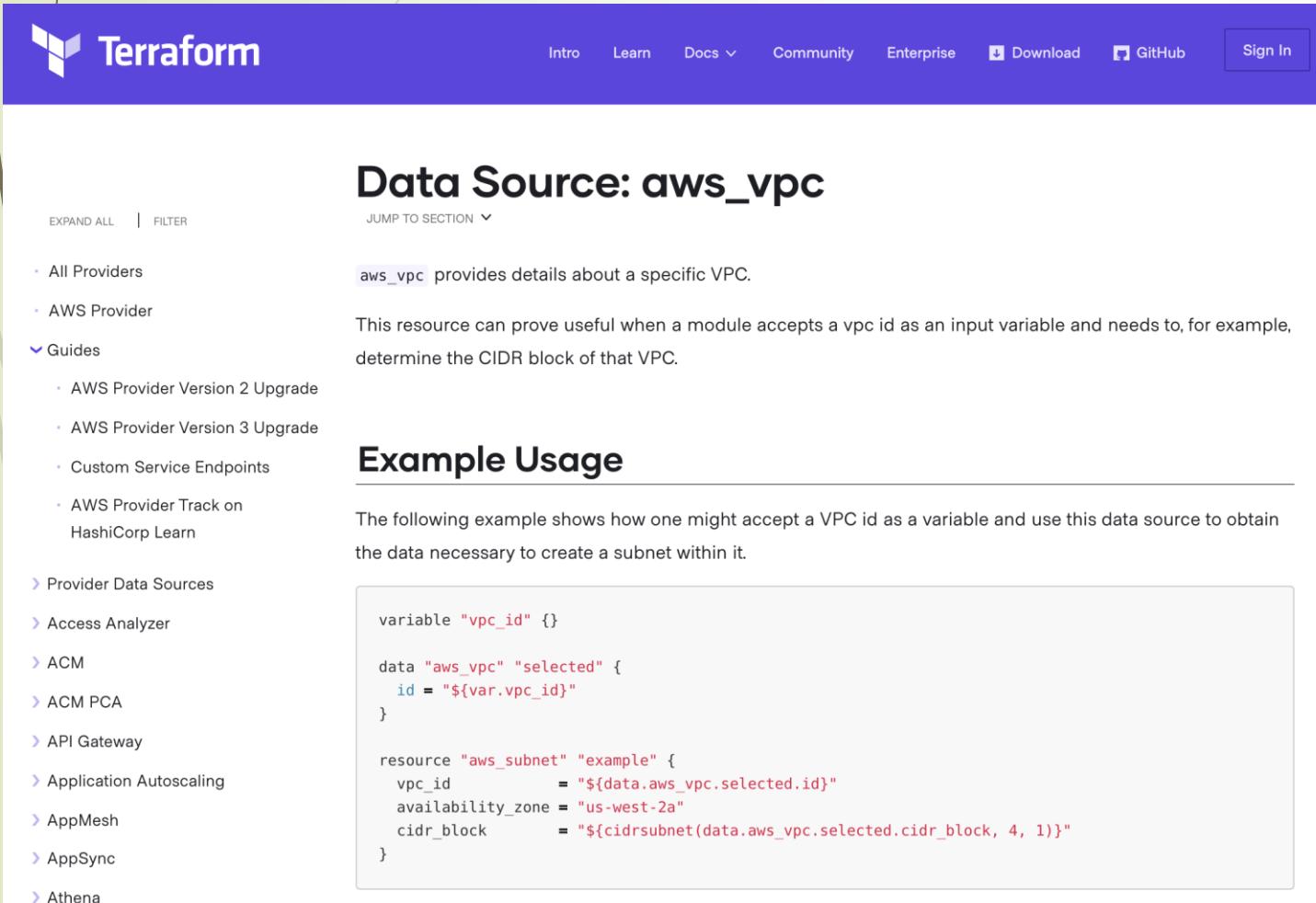
aws_s3_bucket

Name: xxxx

Bucket policy: yyyy

Resource and dependency

► Example: data subnet (aws_vpc)



The screenshot shows the Terraform documentation for the `aws_vpc` data source. The page title is "Data Source: aws_vpc". It includes a sidebar with links to various provider guides and a main content area with a description of the `aws_vpc` resource and its usage examples.

Code Example:

```
1  data "aws_vpc" "labvpc" {
2      id = "vpc-4483e921"
3  }
4
5  output "aws_vpc_cidr" [
6      value = data.aws_vpc.labvpc.cidr_block
7  ]
```

Example Usage:

```
variable "vpc_id" {}

data "aws_vpc" "selected" {
    id = "${var.vpc_id}"
}

resource "aws_subnet" "example" {
    vpc_id      = "${data.aws_vpc.selected.id}"
    availability_zone = "us-west-2a"
    cidr_block   = "${cidrsubnet(data.aws_vpc.selected.cidr_block, 4, 1)}"
}
```

Resource and dependency

► Example: Security Group (aws_security_group_rule)

The screenshot shows the Terraform documentation page for the `aws_security_group_rule` resource. The page has a purple header with links for Intro, Learn, Docs, Community, Enterprise, Download, GitHub, and Sign In. Below the header, there's a sidebar with links for All Providers, AWS Provider, Guides (including AWS Provider Version 2 Upgrade, AWS Provider Version 3 Upgrade, Custom Service Endpoints, AWS Provider Track on HashiCorp Learn), Provider Data Sources, Access Analyzer, ACM, ACM PCA, API Gateway, Application Autoscaling, and AppMesh. The main content area has a title **Resource: aws_security_group_rule**. It includes a note about security group rules and a note about protocol settings. There are also notes about security group references across VPC peering and security group IDs.

Resource: aws_security_group_rule

Provides a security group rule resource. Represents a single `ingress` or `egress` group rule, which can be added to external Security Groups.

NOTE on Security Groups and Security Group Rules: Terraform currently provides both a standalone Security Group Rule resource (a single `ingress` or `egress` rule), and a [Security Group resource](#) with `ingress` and `egress` rules defined in-line. At this time you cannot use a Security Group with in-line rules in conjunction with any Security Group Rule resources. Doing so will cause a conflict of rule settings and will overwrite rules.

NOTE: Setting `protocol = "all"` or `protocol = -1` with `from_port` and `to_port` will result in the EC2 API creating a security group rule with all ports open. This API behavior cannot be controlled by Terraform and may generate warnings in the future.

NOTE: Referencing Security Groups across VPC peering has certain restrictions. More information is available in the [VPC Peering User Guide](#).

Example Usage

Basic usage

```
resource "aws_security_group_rule" "allow_all" {  
  type      = "ingress"  
  from_port = 0  
  to_port   = 65535  
  protocol  = "tcp"  
  # Opening to 0.0.0.0/0 can lead to security vulnerabilities.  
  cidr_blocks = # add a CIDR block here  
  prefix_list_ids = ["pl-12c4e678"]  
  
  security_group_id = "sg-123456"  
}
```

Resource and dependency

► Example: Security Group (aws_security_group_rule)

```
1 resource "aws_security_group" "secgroup_server1" {
2   name      = "secgroup_server1"
3   description = "security group for web server (nginx)"
4   vpc_id    = "vpc-4483e921"
5   #Incoming Rule
6   #Public IP Rule
7   ingress {
8     from_port  = 22
9     to_port    = 22
10    protocol   = "tcp"
11    cidr_blocks = ["0.0.0.0/0"]
12  }
13
14  ingress {
15    from_port  = 80
16    to_port    = 80
17    protocol   = "tcp"
18    cidr_blocks = ["0.0.0.0/0"]
19  }
20
21  ingress {
22    from_port  = 443
23    to_port    = 443
24    protocol   = "tcp"
25    cidr_blocks = ["0.0.0.0/0"]
26  }
27  #Private IP Rule
28  ingress {
29    from_port  = 0
30    to_port    = 0
31    protocol   = "-1"
32    cidr_blocks = [data.aws_vpc.labvpc.cidr_block]
33  }
34  #Outgoing Rule
35  egress {
36    from_port  = 0
37    to_port    = 0
38    protocol   = "-1"
39    cidr_blocks = ["0.0.0.0/0"]
40  }
```

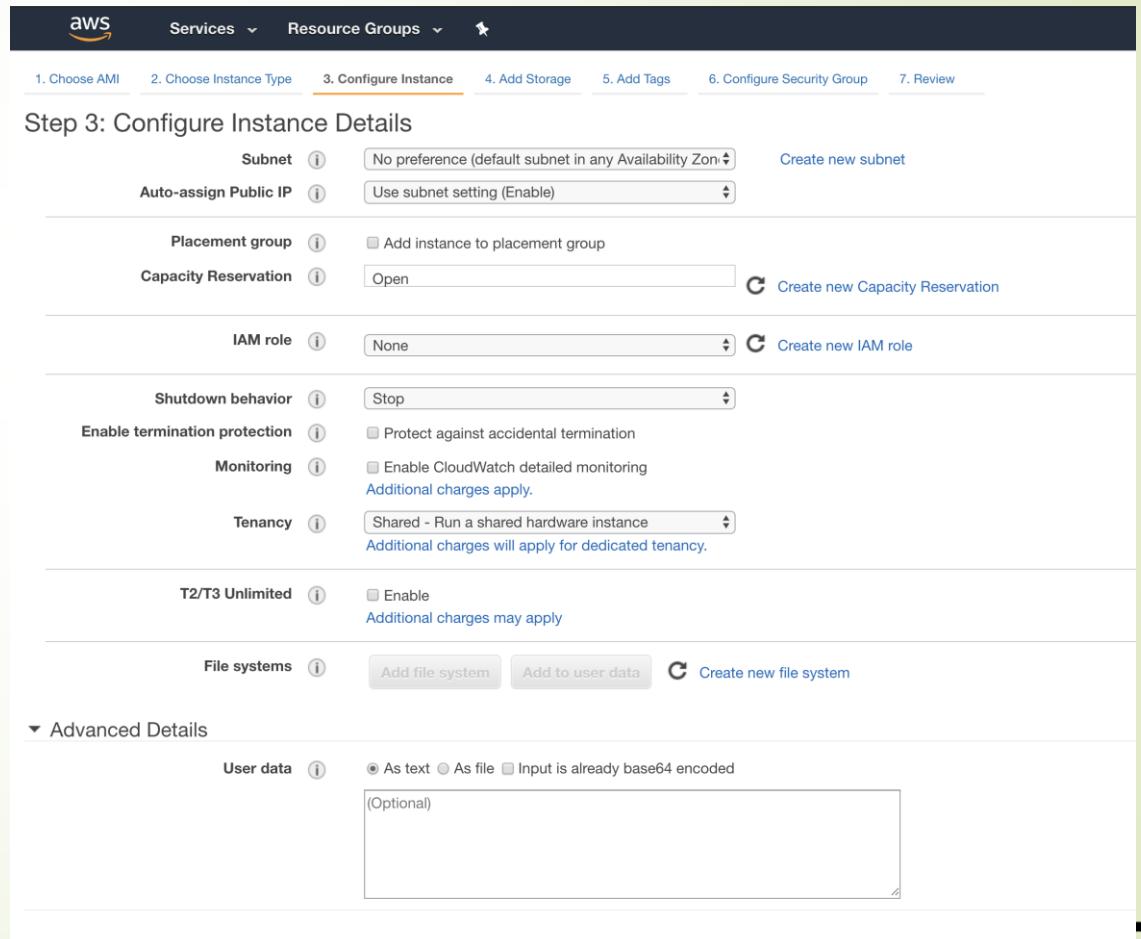
```
42   tags = {
43     Environment = "rd"
44     Region      = "ap-southeast-1"
45     AZ          = "none"
46     Categories  = "compute"
47     Name        = "securitygroup-server1-student-X"
48     Zone        = "public"
49     Module      = "secgroup"
50     Billing     = "terraform-workshop"
51   }
52 }
53
54 resource "aws_security_group" "secgroup_server2" {
55   name      = "secgroup_server1"
56   description = "security group for docker server"
57   vpc_id    = "vpc-4483e921"
58   #Incoming Rule
59   #Public IP Rule
60   ingress {
61     from_port  = 22
62     to_port    = 22
63     protocol   = "tcp"
64     cidr_blocks = ["0.0.0.0/0"]
65   }
66   #Private IP Rule
67   ingress {
68     from_port  = 0
69     to_port    = 0
70     protocol   = "-1"
71     cidr_blocks = [data.aws_vpc.labvpc.cidr_block]
72   }
73   #Outgoing Rule
74   egress {
75     from_port  = 0
76     to_port    = 0
77     protocol   = "-1"
78     cidr_blocks = ["0.0.0.0/0"]
79   }
```

```
80
81   tags = {
82     Environment = "rd"
83     Region      = "ap-southeast-1"
84     AZ          = "none"
85     Categories  = "compute"
86     Name        = "securitygroup-server1-student-X"
87     Zone        = "public"
88     Module      = "secgroup"
89     Billing     = "terraform-workshop"
90   }
91 }
```

Resource and dependency

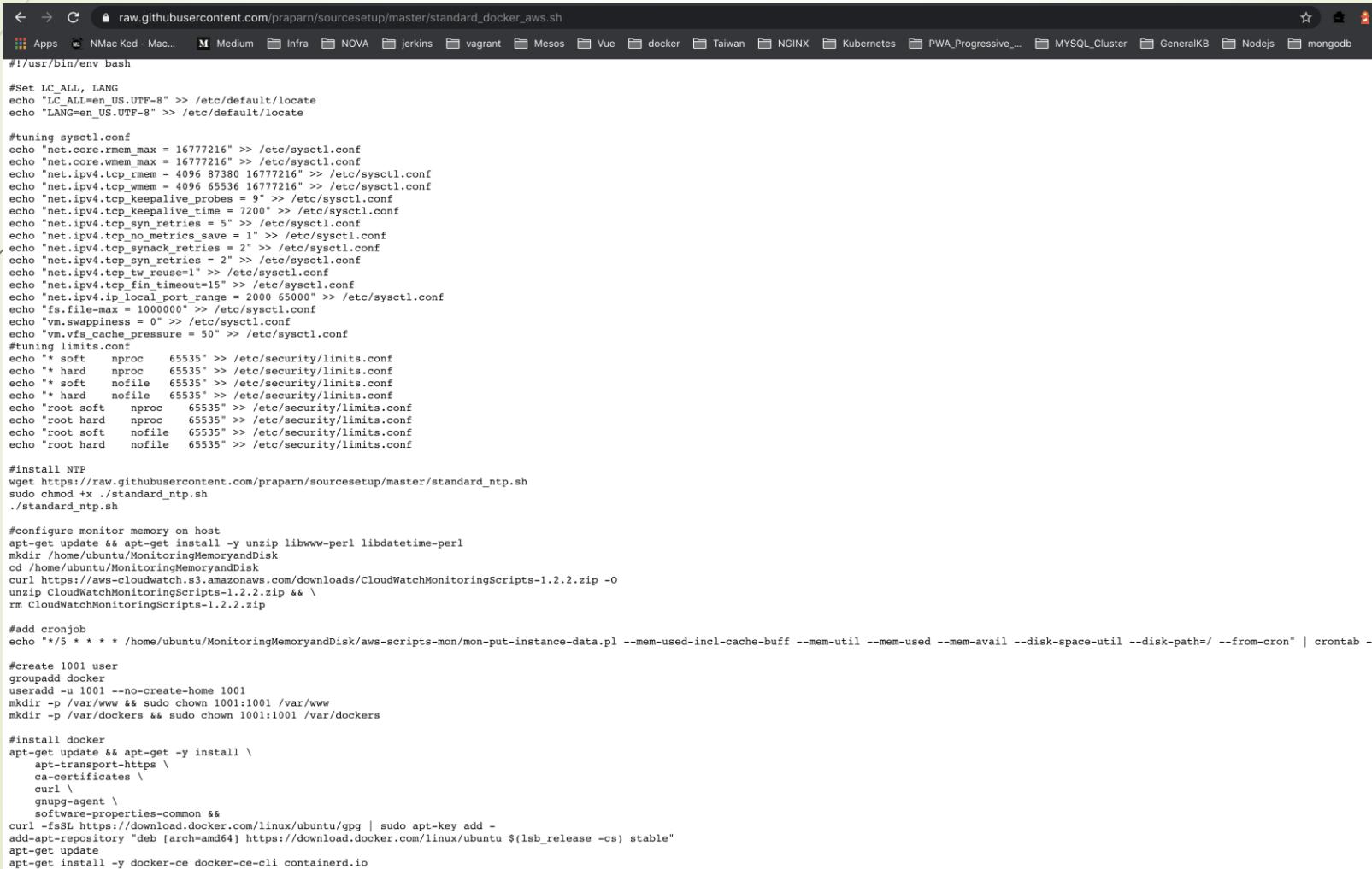
► Example: EC2 instance with user data (aws_instance)

- `user_data` - (Optional) The user data to provide when launching the instance. Do not pass gzip-compressed data via this argument; see `user_data_base64` instead.
- `user_data_base64` - (Optional) Can be used instead of `user_data` to pass base64-encoded binary data directly. Use this instead of `user_data` whenever the value is not a valid UTF-8 string. For example, gzip-encoded user data must be base64-encoded and passed via this argument to avoid corruption.



Resource and dependency

► Example: EC2 instance with user data (aws_instance)



```
#!/usr/bin/env bash

#Set LC_ALL, LANG
echo "LC_ALL=en_US.UTF-8" >> /etc/default/locate
echo "LANG=en_US.UTF-8" >> /etc/default/locate

#tuning sysctl.conf
echo "net.core.rmem_max = 16777216" >> /etc/sysctl.conf
echo "net.core.wmem_max = 16777216" >> /etc/sysctl.conf
echo "net.ipv4.tcp_rmem = 4096 87380 16777216" >> /etc/sysctl.conf
echo "net.ipv4.tcp_wmem = 4096 65536 16777216" >> /etc/sysctl.conf
echo "net.ipv4.tcp_keepalive_probes = 9" >> /etc/sysctl.conf
echo "net.ipv4.tcp_keepalive_time = 7200" >> /etc/sysctl.conf
echo "net.ipv4.tcp_syn_retries = 5" >> /etc/sysctl.conf
echo "net.ipv4.tcp_no_metrics_save = 1" >> /etc/sysctl.conf
echo "net.ipv4.tcp_syn_retries = 2" >> /etc/sysctl.conf
echo "net.ipv4.tcp_syn_retries = 2" >> /etc/sysctl.conf
echo "net.ipv4.tcp_tw_reuse=1" >> /etc/sysctl.conf
echo "net.ipv4.tcp_fin_timeout=15" >> /etc/sysctl.conf
echo "net.ipv4.ip_local_port_range = 2000 65000" >> /etc/sysctl.conf
echo "fs.file-max = 1000000" >> /etc/sysctl.conf
echo "vm.swappiness = 0" >> /etc/sysctl.conf
echo "vm.vfs_cache_pressure = 50" >> /etc/sysctl.conf
#tuning limits.conf
echo "soft    nproc      65535" >> /etc/security/limits.conf
echo "hard    nproc      65535" >> /etc/security/limits.conf
echo "soft    nofile     65535" >> /etc/security/limits.conf
echo "hard    nofile     65535" >> /etc/security/limits.conf
echo "root soft    nproc      65535" >> /etc/security/limits.conf
echo "root hard   nproc      65535" >> /etc/security/limits.conf
echo "root soft    nofile     65535" >> /etc/security/limits.conf
echo "root hard   nofile     65535" >> /etc/security/limits.conf

#install NTP
wget https://raw.githubusercontent.com/praparn/sourcesetup/master/standard_ntp.sh
sudo chmod +x ./standard_ntp.sh
./standard_ntp.sh

#configure monitor memory on host
apt-get update && apt-get install -y unzip libwww-perl libdatetime-perl
mkdir /home/ubuntu/MonitoringMemoryandDisk
cd /home/ubuntu/MonitoringMemoryandDisk
curl https://aws-cloudwatch.s3.amazonaws.com/downloads/CloudWatchMonitoringScripts-1.2.2.zip -O
unzip CloudWatchMonitoringScripts-1.2.2.zip && \
rm CloudWatchMonitoringScripts-1.2.2.zip

#add cronjob
echo "*5 * * * * /home/ubuntu/MonitoringMemoryandDisk/aws-scripts-mon/mon-put-instance-data.pl --mem-used-incl-cache-buff --mem-util --mem-used --mem-avail --disk-space-util --disk-path=/ --from-cron" | crontab -
#create 1001 user
groupadd docker
useradd -u 1001 --no-create-home 1001
mkdir -p /var/www && sudo chown 1001:1001 /var/www
mkdir -p /var/docker && sudo chown 1001:1001 /var/docker

#install docker
apt-get update && apt-get -y install \
  apt-transport-https \
  ca-certificates \
  curl \
  gnupg-agent \
  software-properties-common &&
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
apt-get update
apt-get install -y docker-ce docker-ce-cli containerd.io
```

Resource and dependency

► Example: EC2 instance with user data (aws_instance)

```
1 resource "aws_instance" "labserver1" {
2   ami           = data.aws_ami.ubuntu.id
3   instance_type = "t3a.nano"
4   key_name      = "keypair"
5   availability_zone = "ap-southeast-1a"
6   root_block_device {
7     volume_type      = "gp2"
8     volume_size       = "10"
9     delete_on_termination = "true"
10  }
11 vpc_security_group_ids = [aws_security_group.secgroup_server1.id]
12 user_data            = <<-EOF
13 #!/bin/bash
14 curl https://raw.githubusercontent.com/praparn/sourcesetup/master/standard_nginx.sh > /tmp/setup.sh
15 chmod +x /tmp/setup.sh
16 /tmp/setup.sh
17 EOF
18
19 tags = {
20   Environment = "rd"
21   Region      = "ap-southeast-1"
22   AZ          = "ap-southeast-1a"
23   Categories  = "compute"
24   Name        = "labresource-server1-student-X"
25   Zone        = "public"
26   Module      = "ec2"
27   Billing     = "terraform-workshop"
28 }
29
30
31 resource "aws_instance" "labserver2" {
32   ami           = data.aws_ami.ubuntu.id
33   instance_type = "t3a.nano"
34   key_name      = "keypair"
35   availability_zone = "ap-southeast-1a"
36   root_block_device {
37     volume_type      = "gp2"
38     volume_size       = "10"
39     delete_on_termination = "true"
40   }
41 vpc_security_group_ids = [aws_security_group.secgroup_server1.id]
42 user_data            = <<-EOF
43 #!/bin/bash
44 curl https://raw.githubusercontent.com/praparn/sourcesetup/master/standard_docker.sh > /tmp/setup.sh
45 chmod +x /tmp/setup.sh
46 /tmp/setup.sh
47 EOF
```

```
49   tags = {
50     Environment = "rd"
51     Region      = "ap-southeast-1"
52     AZ          = "ap-southeast-1a"
53     Categories  = "compute"
54     Name        = "labresource-server2-student-X"
55     Zone        = "public"
56     Module      = "ec2"
57     Billing     = "terraform-workshop"
58   }
59
60 output "aws_instance_labserver1_id" {
61   value = aws_instance.labserver1.id
62 }
63
64 output "aws_instance_labserver2_id" {
65   value = aws_instance.labserver2.id
66 }
67
68 output "aws_instance_labserver1_ip" {
69   value = aws_instance.labserver1.public_ip
70 }
71
72 output "aws_instance_labserver2_ip" {
73   value = aws_instance.labserver2.public_ip
74 }
75
76 output "aws_instance_labserver1_public_dns" {
77   value = aws_instance.labserver1.public_dns
78 }
79
80 output "aws_instance_labserver2_public_dns" {
81   value = aws_instance.labserver2.public_dns
82 }
83
84 output "aws_instance_lab_keyname" {
85   value = aws_instance.labserver1.key_name
86 }
```

Resource and dependency

► Example: S3 Bucket (aws_s3_bucket)

The screenshot shows the Terraform documentation for the `aws_s3_bucket` resource. The page has a purple header with the Terraform logo and navigation links for Intro, Learn, Docs, Community, Enterprise, Download, GitHub, and Sign In. The main content area has a white background. At the top left, there's a sidebar with links to All Providers, AWS Provider, Guides (including AWS Provider Version 2 Upgrade, AWS Provider Version 3 Upgrade, Custom Service Endpoints, AWS Provider Track on HashiCorp Learn), Provider Data Sources, Access Analyzer, ACM, ACM PCA, API Gateway, and Application Autoscaling. The main content starts with the title "Resource: aws_s3_bucket" and a sub-section "Provides a S3 bucket resource." Below this is a section titled "Example Usage" with a heading "Private Bucket w/ Tags". A code block shows the Terraform configuration for creating a private S3 bucket with tags:

```
resource "aws_s3_bucket" "b" {
  bucket = "my-tf-test-bucket"
  acl    = "private"

  tags = {
    Name      = "My bucket"
    Environment = "Dev"
  }
}
```

Resource and dependency

► Example: S3 Bucket (aws_s3_bucket)

```
1 resource "aws_s3_bucket" "labbucket" {
2   bucket = "labbucketstudent"
3   acl    = "public-read-write"
4   depends_on = [aws_instance.labserver1, aws_instance.labserver2]
5   tags = {
6     Environment = "rd"
7     Region      = "ap-southeast-1"
8     AZ          = "none"
9     Categories  = "storage"
10    Name        = "labresource-bucket-student-X"
11    Zone        = "public"
12    Module      = "s3"
13    Billing     = "terraform-workshop"
14  }
15 }
16
17 output "aws_s3_labbucket_id" {
18   value = aws_s3_bucket.labbucket.id
19 }
20
21 output "aws_s3_labbucket_domainname" {
22   value = aws_s3_bucket.labbucket.bucket_domain_name
23 }
24
25 output "aws_s3_labbucket_websitedomain" [
26   value = aws_s3_bucket.labbucket.website_domain
27 ]
```

Resource and dependency

► Example: Create: Provision terraform script

```
ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource$ terraform init

Initializing the backend...

Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "aws" (hashicorp/aws) 2.45.0...

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.aws: version = "~> 2.45"

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.
ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource$     terraform fmt -check
00_provider.tf
04_resource_ec2.tf
05_resource_bucket.tf
ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource$     terraform validate
Success! The configuration is valid.
```

Resource and dependency

► Example: Create: Provision terraform script

```
[ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource$    terraform apply -auto-approve -parallelism=5 -refresh=true ./
data.aws_vpc.labvpc: Refreshing state...
data.aws_ami.ubuntu: Refreshing state...
aws_security_group.secgroup_server2: Creating...
aws_security_group.secgroup_server1: Creating...
aws_security_group.secgroup_server1: Creation complete after 1s [id=sg-0014787fae5fb5485]
aws_security_group.secgroup_server2: Creation complete after 1s [id=sg-0264447f546711a0e]
aws_instance.labserver1: Creating...
aws_instance.labserver2: Creating...
aws_instance.labserver1: Still creating... [10s elapsed]
aws_instance.labserver2: Still creating... [10s elapsed]
aws_instance.labserver1: Creation complete after 12s [id=i-065bf9b30149c35b0]
aws_instance.labserver2: Creation complete after 12s [id=i-03adf6a2a47ec0715]
aws_s3_bucket.labbucket: Creating...
aws_s3_bucket.labbucket: Creation complete after 2s [id=labbucketstudent]

Apply complete! Resources: 5 added, 0 changed, 0 destroyed.

Outputs:

aws_instance_lab_keyname = keypair
aws_instance_labserver1_id = i-065bf9b30149c35b0
aws_instance_labserver1_ip = 54.255.246.18
aws_instance_labserver1_public_dns = ec2-54-255-246-18.ap-southeast-1.compute.amazonaws.com
aws_instance_labserver2_id = i-03adf6a2a47ec0715
aws_instance_labserver2_ip = 52.221.214.86
aws_instance_labserver2_public_dns = ec2-52-221-214-86.ap-southeast-1.compute.amazonaws.com
aws_s3_labbucket_domainname = labbucketstudent.s3.amazonaws.com
aws_s3_labbucket_id = labbucketstudent
aws_vpc_cidr = 172.31.0.0/16
ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource$ ]
```

Resource and dependency

► Example: Create: Check information from resource's output and cross-check via aws cli (Server1: Nginx)

DescribeInstances	
Reservations	
OwnerId	404075494050
ReservationId	r-0baef3e3748bf73718
Instances	
AmiLaunchIndex	0
Architecture	x86_64
ClientToken	
EbsOptimized	False
EnaSupport	True
Hypervisor	xen
ImageId	ami-81cefccfd
InstanceId	i-065bf9b30149c35b0
InstanceType	t3a.nano
KeyName	keypair
LaunchTime	2020-01-20T13:56:32.000Z
PrivateDnsName	ip-172-31-20-232.ap-southeast-1.compute.internal
PrivateIpAddress	172.31.20.232
PublicDnsName	ec2-54-255-246-18.ap-southeast-1.compute.amazonaws.com
PublicIpAddress	54.255.246.18
RootDeviceName	/dev/sda1
RootDeviceType	ebs
SourceDestCheck	True
StateTransitionReason	
SubnetId	subnet-41742436
VirtualizationType	hvm
VpcId	vpc-4483e921
BlockDeviceMappings	
DeviceName	/dev/sda1
Ebs	
AttachTime	2020-01-20T13:56:33.000Z
DeleteOnTermination	True
Status	attached
VolumeId	vol-051a09a0bdaba3af6

CapacityReservationSpecification	
CapacityReservationPreference	open
CpuOptions	
CoreCount	1
ThreadsPerCore	2
HibernationOptions	
Configured	False
MetadataOptions	
HttpEndpoint	enabled
HttpPutResponseHopLimit	1
HttpTokens	optional
State	applied
Monitoring	
State	disabled
NetworkInterfaces	
Description	interface
InterfaceType	ebs
MacAddress	06:46:7d:d6:6f:14
NetworkInterfaceId	eni-02b0a66e306708f33
OwnerId	404075494050
PrivateDnsName	ip-172-31-20-232.ap-southeast-1.compute.internal
PrivateIpAddress	172.31.20.232
SourceDestCheck	True
Status	in-use
SubnetId	subnet-41742436
VpcId	vpc-4483e921
Association	
IpOwnerId	amazon
PublicDnsName	ec2-54-255-246-18.ap-southeast-1.compute.amazonaws.com
PublicIp	54.255.246.18

Attachment		
AttachTime	2020-01-20T13:56:32.000Z	
AttachmentId	eni-attach-0fab1cd4943ac8a5	
DeleteOnTermination	True	
DeviceIndex	0	
Status	attached	
Groups		
GroupId	sg-0014787fae5fb5485	
GroupName	secgroup_server1	
PrivateIpAddresses		
Primary	True	
PrivateDnsName	ip-172-31-20-232.ap-southeast-1.compute.internal	
PrivateIpAddress	172.31.20.232	
Association		
IpOwnerId	amazon	
PublicDnsName	ec2-54-255-246-18.ap-southeast-1.compute.amazonaws.com	
PublicIp	54.255.246.18	
Placement		
AvailabilityZone	ap-southeast-1a	
GroupName		
Tenancy	default	
SecurityGroups		
GroupId	sg-0014787fae5fb5485	
GroupName	secgroup_server1	
State		
Code	16	
Name	running	
Tags		
Key		Value
Categories	compute	
Module	ec2	
Environment	rd	
Zone	public	
AZ	ap-southeast-1a	
Billing	terraform-workshop	
Name	labresource-server1-student-35	
Region	ap-southeast-1	

Resource and dependency

➡ Example: Create: Check information from resource's output and cross-check via aws cli (Security Group: Server1)

DescribeSecurityGroups							
SecurityGroups							
Description	GroupId	GroupName	OwnerId	VpcId			
security group for web server (nginx)							
sg-0d00738f4139b4fc	secgroup_server1	404075494050	vpc-4483e921				
IpPermissions							
FromPort	IpProtocol	ToPort					
80	tcp	80					
IpRanges							
CidrIp							
0.0.0.0/0							
IpPermissions							
FromPort	IpProtocol	ToPort					
-1	tcp						
IpRanges							
CidrIp							
172.31.0.0/16							
IpPermissions							
FromPort	IpProtocol	ToPort					
22	tcp	22					
IpRanges							
CidrIp							
0.0.0.0/0							

IpPermissions		
FromPort	IpProtocol	ToPort
443	tcp	443
IpRanges		
CidrIp		
0.0.0.0/0		
IpPermissionsEgress		
IpProtocol		
-1		
IpRanges		
CidrIp		
0.0.0.0/0		
Tags		
Key	Value	
AZ	none	
Environment	rd	
Zone	public	
Module	secgroup	
Region	ap-southeast-1	
Categories	compute	
Billing	terraform-workshop	
Name	securitygroup-server1-student-34	

Resource and dependency

► Example: Create: Check information from resource's output and cross-check via aws cli (Server2: Docker)

DescribeInstances	
Reservations	
OwnerId	404075494050
ReservationId	r-0febbbc2a5bb07b8e7
Instances	
AmiLaunchIndex	0
Architecture	x86_64
ClientToken	
EbsOptimized	False
EnaSupport	True
Hypervisor	xen
ImageId	ami-81cefccfd
InstanceId	i-03adf6a2a47ec0715
InstanceType	t3a.nano
KeyName	keypair
LaunchTime	2020-01-20T13:56:32.000Z
PrivateDnsName	ip-172-31-22-61.ap-southeast-1.compute.internal
PrivateIpAddress	172.31.22.61
PublicDnsName	ec2-52-221-214-86.ap-southeast-1.compute.amazonaws.com
PublicIpAddress	52.221.214.86
RootDeviceName	/dev/sda1
RootDeviceType	ebs
SourceDestCheck	True
StateTransitionReason	
SubnetId	subnet-41742436
VirtualizationType	hvm
VpcId	vpc-4483e921
BlockDeviceMappings	
DeviceName	/dev/sda1
Ebs	
AttachTime	2020-01-20T13:56:33.000Z
DeleteOnTermination	True
Status	attached
VolumeId	vol-049b8084cd0d177b7
CapacityReservationSpecification	
CapacityReservationPreference	
CpuOptions	open
CoreCount	1
ThreadsPerCore	2
HibernationOptions	
Configured	False
MetadataOptions	
HttpEndpoint	enabled
HttpPutResponseHopLimit	1
HttpTokens	optional
State	applied
Monitoring	
State	disabled
NetworkInterfaces	
Description	interface
InterfaceType	eni-0e5b528b613099ff2
MacAddress	06:37:22:50:68:c4
NetworkInterfaceId	eni-0e5b528b613099ff2
OwnerId	404075494050
PrivateDnsName	ip-172-31-22-61.ap-southeast-1.compute.internal
PrivateIpAddress	172.31.22.61
SourceDestCheck	True
Status	in-use
SubnetId	subnet-41742436
VpcId	vpc-4483e921
Association	
IpOwnerId	amazon
PublicDnsName	ec2-52-221-214-86.ap-southeast-1.compute.amazonaws.com
PublicIp	52.221.214.86
Attachment	
AttachTime	2020-01-20T13:56:32.000Z
AttachmentId	eni-attach-09154404c9f17be59
DeleteOnTermination	True
DeviceIndex	0
Status	attached
Groups	
GroupId	sg-0014787fae5fb5485
GroupName	secgroup_server1
PrivateIpAddresses	
Primary	True
PrivateDnsName	ip-172-31-22-61.ap-southeast-1.compute.internal
PrivateIpAddress	172.31.22.61
Association	
IpOwnerId	amazon
PublicDnsName	ec2-52-221-214-86.ap-southeast-1.compute.amazonaws.com
PublicIp	52.221.214.86
Placement	
AvailabilityZone	ap-southeast-1a
GroupName	
Tenancy	default
SecurityGroups	
GroupId	sg-0014787fae5fb5485
GroupName	secgroup_server1
State	
Code	16
Name	running
Tags	
Key	Value
AZ	ap-southeast-1a
Environment	rd
Zone	public
Billing	terraform-workshop
Name	labresource-server2-student-35
Region	ap-southeast-1
Categories	compute
Module	ec2

Resource and dependency

➡ Example: Create: Check information from resource's output and cross-check via aws cli (Security Group: Server2)

DescribeSecurityGroups				
SecurityGroups				
Description	GroupId	GroupName	OwnerId	VpcId
security group for docker server	sg-0a737ceeb5bc84d51	securitygroup-server2-student-34	464075494050	vpc-4483e921
IpPermissions				
FromPort	IpProtocol	ToPort		
	-1			
IpRanges				
CidrIp				
172.31.0.0/16				
IpPermissions				
FromPort	IpProtocol	ToPort		
22	tcp	22		
IpRanges				
CidrIp				
0.0.0.0/0				

IpPermissionsEgress	
IpProtocol	
-1	
IpRanges	
CidrIp	
0.0.0.0/0	
Tags	
Key	Value
Module	secgroup
Region	ap-southeast-1
Name	securitygroup-server2-student-34
Categories	compute
AZ	none
Environment	rd
Zone	public
Billing	terraform-workshop

Resource and dependency

► **Example: Create:** Check information from resource's output and cross-check via aws cli (S3 Bucket)

```
ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource$ aws s3api list-buckets
+-----+
|          ListBuckets                         |
+-----+
|          Buckets                            |
+-----+
|          CreationDate           Name        |
+-----+
| 2020-01-20T14:33:35.000Z | labresource-bucket-student-35 |
+-----+
|          Owner                               |
+-----+
| DisplayName | ID                         |
+-----+
| eva10409 | 0711825c10848d11f2d6c9d2187859a97d90be3dbfa4dc01a659a7c79925a863 |
+-----+
ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource$ aws s3api list-buckets --query "Buckets[].Name" | grep labresource-bucket-student-35
| labresource-bucket-student-35 |
ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource$
```

Resource and dependency

► **Example: Create:** Test connect via ssh to new EC2 machine and check nginx process (Server1)

```
[praparns-MacBook-Pro:~] praparn$ ssh -i terraformlab ubuntu@54.254.232.237
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-1060-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
 http://www.ubuntu.com/business/services/cloud

201 packages can be updated.
125 updates are security updates.

New release '18.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

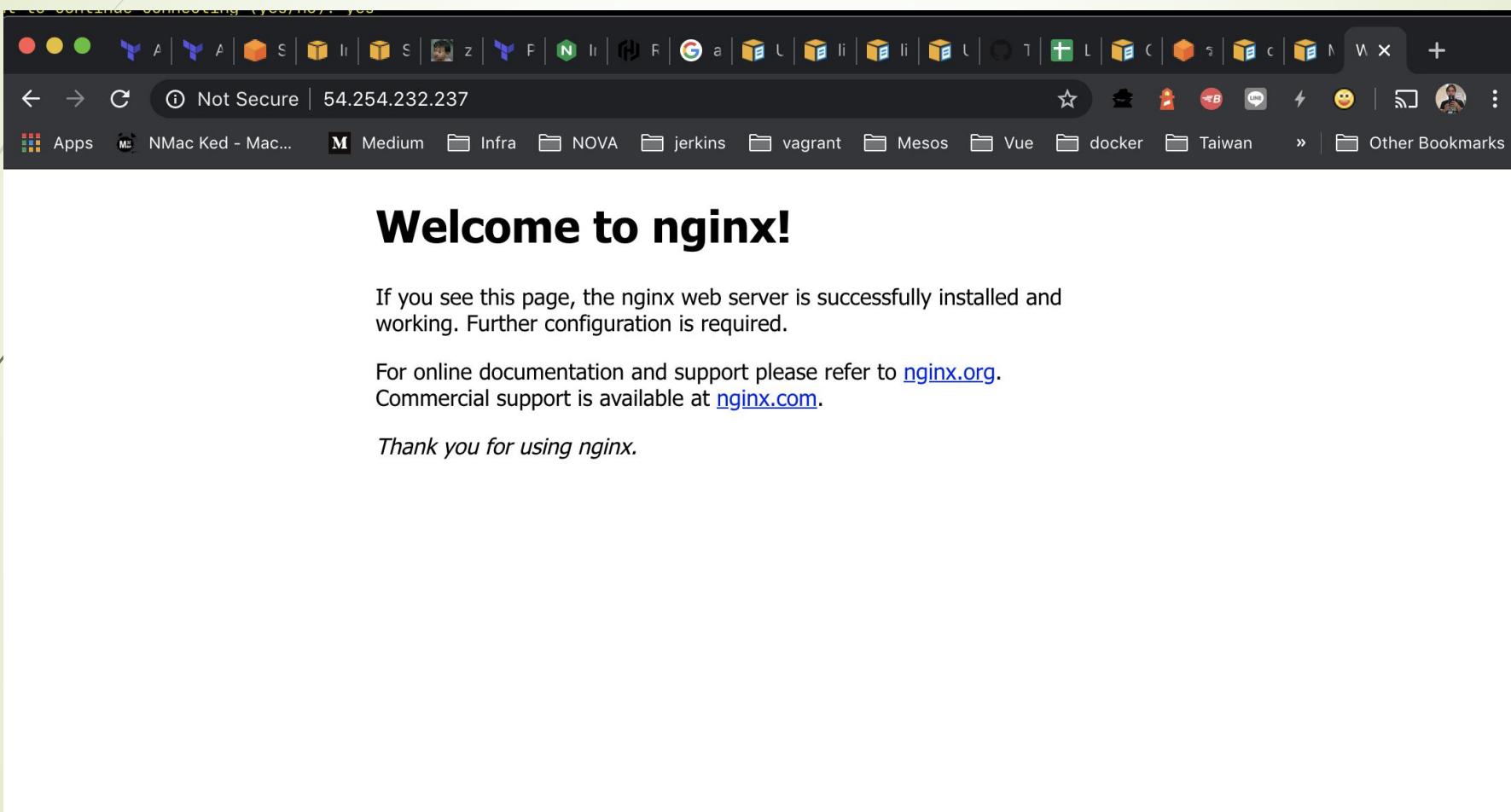
Last login: Mon Jan 20 21:45:58 2020 from 184.22.251.169
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-22-174:~]$
```

```
[ubuntu@ip-172-31-22-174:~]$ ps -ef | grep nginx
root    6642     1  0 21:34 ?        00:00:00 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
www-data 6643   6642  0 21:34 ?        00:00:00 nginx: worker process
www-data 6644   6642  0 21:34 ?        00:00:00 nginx: worker process
ubuntu   6890   6874  0 21:49 pts/0    00:00:00 grep --color=auto nginx
[ubuntu@ip-172-31-22-174:~]$ curl http://localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
}
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
ubuntu@ip-172-31-22-174:~]$
```

Resource and dependency

➡ Example: Create: Test web page of nginx



Resource and dependency

► Example: Create: Test connect via ssh to new EC2 machine and check docker command (Server2)

```
[praparn-MacBook-Pro:~ praparn$ ssh -i terraformlab ubuntu@54.254.157.254
The authenticity of host '54.254.157.254 (54.254.157.254)' can't be established.
ECDSA key fingerprint is SHA256:xfDgpK7Vp1Vop6YoyJbvFZ3VcA/beOP50DPtcLI29CM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '54.254.157.254' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-1060-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
 http://www.ubuntu.com/business/services/cloud

195 packages can be updated.
122 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

WARNING! Your environment specifies an invalid locale.
The unknown environment variables are:
 LC_CTYPE=UTF-8 LC_ALL=
This can affect your user experience significantly, including the
ability to manage packages. You may install the locales by running:

 sudo apt-get install language-pack-UTF-8
 or
 sudo locale-gen UTF-8

To see all available language packs, run:
 apt-cache search "^language-pack-[a-z][a-z]$"
To disable this message for all users, run:
 sudo touch /var/lib/cloud/instance/locale-check.skip

ubuntu@ip-172-31-29-22:~$ ]
```

```
ubuntu@ip-172-31-30-215:~$ docker version
Client: Docker Engine - Community
 Version:          19.03.5
 API version:     1.40
 Go version:      go1.12.12
 Git commit:      633a0ea838
 Built:           Wed Nov 13 07:50:12 2019
 OS/Arch:         linux/amd64
 Experimental:   false

Server: Docker Engine - Community
Engine:
 Version:          19.03.5
 API version:     1.40 (minimum version 1.12)
 Go version:      go1.12.12
 Git commit:      633a0ea838
 Built:           Wed Nov 13 07:48:43 2019
 OS/Arch:         linux/amd64
 Experimental:   false
containerd:
 Version:          1.2.10
 GitCommit:        b34a5c8af56e510852c35414db4c1f4fa6172339
runc:
 Version:          1.0.0-rc8+dev
 GitCommit:        3e425f80a8c931f88e6d94a8c831b9d5aa481657
docker-init:
 Version:          0.18.0
 GitCommit:        fec3683
ubuntu@ip-172-31-30-215:~$ ]
```

```
GitCommit: fec3683
ubuntu@ip-172-31-30-215:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:9572f7cdcee8591948c2963463447a53466950b3fc15a247fcad1917ca215a2f
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
ubuntu@ip-172-31-30-215:~$ ]
```

Resource and dependency

► Example: Create: Test push/delete file to s3 and display

```
[ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource$ curl https://raw.githubusercontent.com/praparn/sourcesetup/master/standard_docker_aws.sh > ~/script_docker.sh
% Total    % Received % Xferd  Average Speed   Time     Time      Time  Current
          Dload  Upload   Total Spent    Left  Speed
100  3361  100  3361    0     0  6936      0 --:--:-- --:--:-- --:--:-- 6929
[ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource$ aws s3api list-buckets --query "Buckets[].Name" | grep labresource-bucket-student-34
| labresource-bucket-student-34 |
[ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource$ aws s3 ls s3://labresource-bucket-student-34
[ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource$ aws s3 ls s3://labresource-bucket-student-34
[ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource$ aws s3 cp ~/script_docker.sh s3://labresource-bucket-student-34
upload: ../../script_docker.sh to s3://labresource-bucket-student-34/script_docker.sh
[ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource$ aws s3 ls s3://labresource-bucket-student-34
2020-01-20 16:03:12      3361 script_docker.sh
ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource$ ]
```

```
[ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource$ aws s3 rm s3://labresource-bucket-student-34/script_docker.sh
delete: s3://labresource-bucket-student-34/script_docker.sh
[ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource$ aws s3 ls s3://labresource-bucket-student-34
ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource$ ]
```

Resource and dependency

➡ Example: Destroy: Delete EC2/EBS by terraform script

DescribeInstances	
Reservations	
OwnerId	404075494050
r-04533a81f5e2f10ac	
Instances	
AmiLaunchIndex	0
Architecture	x86_64
ClientToken	
EbsOptimized	False
EnaSupport	True
Hypervisor	xen
ImageId	ami-81cefcd
InstanceId	i-09840dce566acc2ed
InstanceType	t3a.nano
KeyName	keypair
LaunchTime	2020-01-20T15:45:56.000Z
PrivateDnsName	
PublicDnsName	
RootDeviceName	/dev/sda1
RootDeviceType	ebs
StateTransitionReason	User initiated (2020-01-20 16:11:31 GMT)
VirtualizationType	hvm
CapacityReservationSpecification	
CapacityReservationPreference	open
CpuOptions	
CoreCount	1
ThreadsPerCore	2
HibernationOptions	
Configured	False
MetadataOptions	
HttpEndpoint	enabled
HttpPutResponseHopLimit	1
HttpTokens	optional
State	pending
Monitoring	
State	disabled
Placement	
AvailabilityZone	ap-southeast-1a
GroupName	
Tenancy	default
State	
Code	48
Name	terminated

DescribeInstances	
Reservations	
OwnerId	404075494050
r-075de01970a72fe5b	
Instances	
AmiLaunchIndex	0
Architecture	x86_64
ClientToken	
EbsOptimized	False
EnaSupport	True
Hypervisor	xen
ImageId	ami-81cefcd
InstanceId	i-003bffd4045a7a89c
InstanceType	t3a.nano
KeyName	keypair
LaunchTime	2020-01-20T15:45:56.000Z
PrivateDnsName	
PublicDnsName	
RootDeviceName	/dev/sda1
RootDeviceType	ebs
StateTransitionReason	User initiated (2020-01-20 16:11:31 GMT)
VirtualizationType	hvm
CapacityReservationSpecification	
CapacityReservationPreference	open
CpuOptions	
CoreCount	1
ThreadsPerCore	2
HibernationOptions	
Configured	False
MetadataOptions	
HttpEndpoint	enabled
HttpPutResponseHopLimit	1
HttpTokens	optional
State	pending
Monitoring	
State	disabled
Placement	
AvailabilityZone	ap-southeast-1a
GroupName	
Tenancy	default
State	
Code	48
Name	terminated

Resource and dependency

► Example: Destroy: Delete EC2/EBS by terraform script

```
[ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource]$ aws s3api list-buckets --query "Buckets[].Name" | grep labresource-bucket-student-34
[ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource]$ aws s3 ls s3://labresource-bucket-student-34

An error occurred (NoSuchBucket) when calling the ListObjectsV2 operation: The specified bucket does not exist
ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource$ █
```

Workshop: Explicit Resource



Workshop: Explicit Resource

```
[ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource$    terraform apply -auto-approve -parallelism=5 -refresh=true ./
data.aws_vpc.labvpc: Refreshing state...
data.aws_ami.ubuntu: Refreshing state...
aws_security_group.secgroup_server2: Creating...
aws_security_group.secgroup_server1: Creating...
aws_security_group.secgroup_server1: Creation complete after 1s [id=sg-0014787fae5fb5485]
aws_security_group.secgroup_server2: Creation complete after 1s [id=sg-0264447f546711a0e]
aws_instance.labserver1: Creating...
aws_instance.labserver2: Creating...
aws_instance.labserver1: Still creating... [10s elapsed]
aws_instance.labserver2: Still creating... [10s elapsed]
aws_instance.labserver1: Creation complete after 12s [id=i-065bf9b30149c35b0]
aws_instance.labserver2: Creation complete after 12s [id=i-03adf6a2a47ec0715]
aws_s3_bucket.labbucket: Creating...
aws_s3_bucket.labbucket: Creation complete after 2s [id=labbucketstudent]

Apply complete! Resources: 5 added, 0 changed, 0 destroyed.

Outputs:

aws_instance_lab_keyname = keypair
aws_instance_labserver1_id = i-065bf9b30149c35b0
aws_instance_labserver1_ip = 54.255.246.18
aws_instance_labserver1_public_dns = ec2-54-255-246-18.ap-southeast-1.compute.amazonaws.com
aws_instance_labserver2_id = i-03adf6a2a47ec0715
aws_instance_labserver2_ip = 52.221.214.86
aws_instance_labserver2_public_dns = ec2-52-221-214-86.ap-southeast-1.compute.amazonaws.com
aws_s3_labbucket_domainname = labbucketstudent.s3.amazonaws.com
aws_s3_labbucket_id = labbucketstudent
aws_vpc_cidr = 172.31.0.0/16
ubuntu@ip-172-31-21-38:~/terraform_202003/Workshop_1.3_Resource/03_explicit_resource$ ]
```

Q&A



Provisioner

- ▶ **Provisioner**
- ▶ Ideally for provisioner will be use for do some action on local-machine (terraform), remote-machine or trigger to other configuration management (CM)
- ▶ Benefit is for provide next step of state after terraform apply done. Or you can't execute consequence step from existing state.(Ex: install & configure chef agent on target machine)
- ▶ Provisioner is always the last result from terraform script
- ▶ Multiple provisioner was supported (Order by written on script)
- ▶ Provisioner can set operate before destroyed resource (when="destroy")
- ▶ Warning: Provisioner can root-cause of terraform failure. Even though other component was success (on_failure = "continue/fail")
- ▶ Some provisioner need connection (ssh, winrm)

Provisioner

- ▶ **Type of Provisioner**
- ▶ chef Provisioner
- ▶ file Provisioner
- ▶ habitat Provisioner
- ▶ local-exec Provisioner
- ▶ puppet Provisioner
- ▶ remote-exec Provisioner
- ▶ salt-masterless Provisioner

Provisioner

- ▶ **Some use-case is not recommended for use provisioner**
- ▶ Passing data into virtual machine and other computer resource
 - ▶ Ex:
 - ▶ AWS EC2: user_data, user_data_base64
 - ▶ Azure: customer_data
 - ▶ Gcloud: meta_data
- ▶ Running configuration management software
- ▶ First-class terraform provider was available
 - ▶ Ex:
 - ▶ PostgreSQL
 - ▶ Docker
 - ▶ Kubernetes

Provisioner

- ▶ **Example:** Create multiple component with dependency (Implicit)
 - ▶ Requirement:
 - ▶ Need to create instance server from image (ami): Ubuntu server 18.04 last build of Jan,2020
 - ▶ This instance server will have base disk for "/" about 10 GB disk
 - ▶ Additional disk for keep data is about 20 GB disk
 - ▶ Add Tag for input description

Provisioner

Data Source:

aws_ami

ami-id →

Resource:

aws_instance

block storage: "/" (10 GB)

Tag ec2 by standard tag

← Attach Volume

Resource:

aws_ebs_volume

block storage: (20 GB)

Resource:

aws_volume_attachment

device name: xxxxxxxx

instance id: xxxxxxxx

ebs volume id: xxxxxxxx

Resource and dependency

- **Example:** Create multiple component with dependency (Implicit)
 - Brake Down
 - Subnet Mask of VPC: data: aws_vpc
 - Security Group: resource: aws_security_group_rule
 - Security Group: nginx web server
 - Security Group: docker server
 - EC2: resource: aws_instance
 - Configure automatic script setup for nginx/docker
 - Configure tag as standard naming
 - S3 bucket: resource: aws_s3_bucket

Resource and dependency

► **Example:** Create multiple component with dependency (Implicit)

► Brake Down

► We will use remote-exec for format and mount disk as command below:

► `sudo mkfs -t xfs /dev/nvme1n1`

► `sudo mkdir /additionalstorage`

► `sudo mount /dev/nvme1n1 /additionalstorage`

► If terraform got destroy. We need to unmount all disk as command below:

► `sudo umount -a`

Resource and dependency

► **Example:** Create multiple component with dependency (Implicit)

```
1  resource "aws_instance" "lab" {
2    ami           = data.aws_ami.ubuntu.id
3    instance_type = "t3a.nano"
4    key_name     = "keypair"
5    availability_zone = "ap-southeast-1c"
6    provisioner "remote-exec" {
7      when      = destroy
8      connection {
9        type      = "ssh"
10       user      = "ubuntu"
11       private_key = "~/.terraformlab"
12       host      = self.public_ip
13     }
14     inline = [
15       "sudo umount -a"
16     ]
17     on_failure = continue
18   }
19   root_block_device {
20     volume_type      = "gp2"
21     volume_size      = "10"
22     delete_on_termination = "true"
23   }
24
25   tags = {
26     Environment = "rd"
27     Region      = "ap-southeast-1"
28     AZ          = "ap-southeast-1c"
29     Categories  = "compute"
30     Name        = "labresource-student-X"
31     Zone        = "public"
32     Module      = "ec2"
33     Billing     = "terraform-workshop"
34   }
35 }
```

```
18  resource "aws_volume_attachment" "labebsattach" {
19    device_name = "/dev/sdf"
20    volume_id   = aws_ebs_volume.labebs.id
21    instance_id = aws_instance.lab.id
22    provisioner "remote-exec" {
23      connection {
24        type      = "ssh"
25        user      = "ubuntu"
26        private_key = file("/home/ubuntu/terraformlab")
27        host      = aws_instance.lab.public_ip
28      }
29      inline = [
30        "sudo mkfs -t xfs /dev/nvme1n1",
31        "sudo mkdir /additionalstorage",
32        "sudo mount /dev/nvme1n1 /additionalstorage"
33      ]
34      on_failure = fail
35    }
36    provisioner "local-exec" {
37      when      = destroy
38      command = "echo 'umount done for destroy (event)'"
39    }
40  }
```

Resource and dependency

► Example: Create: Provision terraform script

```
ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/02_implicit_resource$ terraform init

Initializing the backend...

Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "aws" (hashicorp/aws) 2.45.0...

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.aws: version = "~> 2.45"

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.
ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/02_implicit_resource$     terraform fmt -check
00_provider.tf
02_resource_ec2.tf
ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/02_implicit_resource$     terraform validate
Success! The configuration is valid.

ubuntu@ip-172-31-27-186:~/terraform_202003/Workshop_1.3_Resource/02_implicit_resource$ terraform apply -auto-approve -parallelism=5 -refresh=true ./
```

Resource and dependency

► Example: Create: Provision terraform script

```
ubuntu@ip-172-31-19-113:~/terraform_202003/Workshop_1.4_Provisioner$ terraform apply -auto-approve -parallelism=5 -refresh=true .
data.aws_ami.ubuntu: Refreshing state...
[aws_ebs_volume.labebs: Creating...
aws_instance.lab: Creating...
aws_ebs_volume.labebs: Still creating... [10s elapsed]
aws_instance.lab: Still creating... [10s elapsed]
aws_ebs_volume.labebs: Creation complete after 10s [id=vol-0e5e772a20d140cea]
aws_instance.lab: Creation complete after 12s [id=i-0a16c706816ceadad]
aws_volume_attachment.labebsattach: Creating...
aws_volume_attachment.labebsattach: Still creating... [10s elapsed]
aws_volume_attachment.labebsattach: Still creating... [20s elapsed]
aws_volume_attachment.labebsattach: Provisioning with 'remote-exec'...
aws_volume_attachment.labebsattach (remote-exec): Connecting to remote host via SSH...
aws_volume_attachment.labebsattach (remote-exec): Host: 13.251.124.175
aws_volume_attachment.labebsattach (remote-exec): User: ubuntu
aws_volume_attachment.labebsattach (remote-exec): Password: false
aws_volume_attachment.labebsattach (remote-exec): Private key: true
aws_volume_attachment.labebsattach (remote-exec): Certificate: false
aws_volume_attachment.labebsattach (remote-exec): SSH Agent: false
aws_volume_attachment.labebsattach (remote-exec): Checking Host Key: false
aws_volume_attachment.labebsattach (remote-exec): Connected!
aws_volume_attachment.labebsattach (remote-exec): meta-data=/dev/nvme1n1           isize=512    agcount=4, agsize=1310720 blks
aws_volume_attachment.labebsattach (remote-exec):          =                     sectsz=512   attr=2, projid32bit=1
aws_volume_attachment.labebsattach (remote-exec):          =                     crc=1      finobt=1, sparse=0, rmapbt=0, reflink=0
aws_volume_attachment.labebsattach (remote-exec): data        =                     bsize=4096  blocks=5242880, imaxpct=25
aws_volume_attachment.labebsattach (remote-exec):          =                     sunit=0    swidth=0 blks
aws_volume_attachment.labebsattach (remote-exec): naming     =version 2       bsize=4096  ascii-ci=0 ftype=1
aws_volume_attachment.labebsattach (remote-exec): log        =internal log   bsize=4096  blocks=2560, version=2
aws_volume_attachment.labebsattach (remote-exec):          =                     sectsz=512  sunit=0 blks, lazy-count=1
aws_volume_attachment.labebsattach (remote-exec): realtime   =none          extsz=4096  blocks=0, rtextents=0
aws_volume_attachment.labebsattach: Creation complete after 22s [id=vai-1096346766]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Outputs:

aws_ebs_volume_labebs_id = vol-0e5e772a20d140cea
aws_instance_lab_id = i-0a16c706816ceadad
aws_instance_lab_keyname = keypair
aws_instance_lab_public_dns = ec2-13-251-124-175.ap-southeast-1.compute.amazonaws.com
aws_instance_lab_public_ip = 13.251.124.175
ubuntu@ip-172-31-19-113:~/terraform_202003/Workshop_1.4_Provisioner$
```

Resource and dependency

➡ Example: Create: Check information from resource's output and cross-check via aws cli

DescribeInstances	
Reservations	
OwnerId	404075494050
ReservationId	r-07e6c93e6387724c
Instances	
AniLaunchIndex	0
Architecture	x86_64
ClientToken	
EbsOptimized	False
EnaSupport	True
Hypervisor	xen
ImageId	ami-09a4a9ce71ff3f20b
InstanceId	i-018ba99df07d91bc7
InstanceType	t3a.nano
KeyName	keypair
LaunchTime	2020-01-21T15:48:48.000Z
PrivateDnsName	ip-172-31-36-165.ap-southeast-1.compute.internal
PrivateIpAddress	172.31.36.165
PublicDnsName	ec2-3-1-51-49.ap-southeast-1.compute.amazonaws.com
PublicIpAddress	3.1.51.49
RootDeviceName	/dev/sda1
RootDeviceType	ebs
SourceDestCheck	True
StateTransitionReason	
SubnetId	subnet-5a1e2c1c
VirtualizationType	hvm
VpcId	vpc-4483e921
BlockDeviceMappings	
DeviceName	/dev/sda1
Ebs	
AttachTime	2020-01-21T15:48:49.000Z
DeleteOnTermination	True
Status	attached
VolumeId	vol-0ae9ee75e53e630a9
BlockDeviceMappings	
DeviceName	/dev/sdf
Ebs	
AttachTime	2020-01-21T15:49:10.000Z
DeleteOnTermination	False
Status	attached
VolumeId	vol-05705e4be2f817326

Monitoring	
State	disabled
NetworkInterfaces	
Description	
MacAddress	0a:ea:51:d7:c4:04
NetworkInterfaceId	eni-03e433dacb936fe80
OwnerId	404075494050
PrivateDnsName	ip-172-31-36-165.ap-southeast-1.compute.internal
PrivateIpAddress	172.31.36.165
SourceDestCheck	True
Status	in-use
SubnetId	subnet-5a1e2c1c
VpcId	vpc-4483e921
Association	
IpOwnerId	amazon
PublicDnsName	ec2-3-1-51-49.ap-southeast-1.compute.amazonaws.com
PublicIp	3.1.51.49
Attachment	
AttachTime	2020-01-21T15:48:000Z
AttachmentId	eni-attach-0a79fb99548eeee3dd
DeleteOnTermination	True
DeviceIndex	0
Status	attached
Groups	
GroupId	sg-fe2d3c9b
GroupName	default
PrivateIpAddresses	
Primary	True
PrivateDnsName	ip-172-31-36-165.ap-southeast-1.compute.internal
PrivateIpAddress	172.31.36.165
Association	
IpOwnerId	amazon
PublicDnsName	ec2-3-1-51-49.ap-southeast-1.compute.amazonaws.com
PublicIp	3.1.51.49

Placement	
AvailabilityZone	ap-southeast-1c
GroupName	default
SecurityGroups	
GroupId	sg-fe2d3c9b
GroupName	default
State	
Code	16
Name	running
Tags	
Key	Value
Billing	terraform-workshop
Region	ap-southeast-1
AZ	ap-southeast-1c
Zone	public
Categories	compute
Environment	rd
Module	ec2
Name	labresource-student-30

Resource and dependency

► **Example: Create:** Check information from resource's output and cross-check via aws cli

```
ubuntu@ip-172-31-19-113:~/terraform_202003/Workshop_1.4_Provisioner$ aws ec2 describe-volumes --region ap-southeast-1 --filters Name=attachment.instance-id,Values=i-018ba99df07d91bc7
```

DescribeVolumes									
Volumes									
AvailabilityZone	CreateTime	Encrypted	Iops	Size	SnapshotId	State	VolumeId	VolumeType	
ap-southeast-1c	2020-01-21T15:48:49.824Z	False	100	10	snap-08016d4ab116eb784	in-use	vol-0ae9ee75e53e630a9	gp2	
Attachments									
AttachTime	DeleteOnTermination	Device	InstanceId	State	VolumeId				
2020-01-21T15:48:49.000Z	True	/dev/sda1	i-018ba99df07d91bc7	attached	vol-0ae9ee75e53e630a9				
Volumes									
AvailabilityZone	CreateTime	Encrypted	Iops	Size	SnapshotId	State	VolumeId	VolumeType	
ap-southeast-1c	2020-01-21T15:48:48.057Z	False	100	20		in-use	vol-05705e4be2f817326	gp2	
Attachments									
AttachTime	DeleteOnTermination	Device	InstanceId	State	VolumeId				
2020-01-21T15:49:10.000Z	False	/dev/sdf	i-018ba99df07d91bc7	attached	vol-05705e4be2f817326				
Tags									
Key	Value								
Module	ebs								
Billing	terraform-workshop								
Region	ap-southeast-1								
AZ	ap-southeast-1c								
Categories	storage								
Environment	rd								
Name	EBSlabresource-student-30								
Zone	public								

Resource and dependency

► **Example: Create:** Test connect via ssh to new EC2 machine and list disk

```
[praparns-MacBook-Pro:~] praparn$ ssh -i terraformlab ubuntu@13.251.124.175
The authenticity of host '13.251.124.175 (13.251.124.175)' can't be established.
ECDSA key fingerprint is SHA256:fPHiZEP/ziXtCzyGX+XjuYqL/yZ7e3T4BQVevBHoP+k.
Are you sure you want to continue connecting (yes/no)? ~ n
Please type 'yes' or 'no': yes
Warning: Permanently added '13.251.124.175' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1057-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 System information as of Tue Jan 21 16:25:05 UTC 2020

 System load:  0.0          Processes:      117
 Usage of /:   11.0% of 9.63GB   Users logged in:  0
 Memory usage: 34%
 Swap usage:   0%
 IP address for ens5: 172.31.43.11

 * Overheard at KubeCon: "microk8s.status just blew my mind".
   https://microk8s.io/docs/commands#microk8s.status

0 packages can be updated.
0 updates are security updates.

Last login: Tue Jan 21 16:22:02 2020 from 13.229.60.55
ubuntu@ip-172-31-43-11:~]$
```

```
[ubuntu@ip-172-31-43-11:~]$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0    7:0    0 89.1M  1 loop /snap/core/8268
loop1    7:1    0  18M  1 loop /snap/amazon-ssm-agent/1480
nvme0n1  259:0  0  10G  0 disk 
└─nvme0n1p1 259:1  0  10G  0 part /
nvme1n1  259:2  0  20G  0 disk /additionalstorage
[ubuntu@ip-172-31-43-11:~]$ df -kh
Filesystem      Size  Used Avail Use% Mounted on
udev            219M    0  219M  0% /dev
tmpfs           47M  736K  46M  2% /run
/dev/nvme0n1p1  9.7G  1.1G  8.6G 12% /
tmpfs           231M    0  231M  0% /dev/shm
tmpfs           5.0M    0  5.0M  0% /run/lock
tmpfs           231M    0  231M  0% /sys/fs/cgroup
/dev/loop0       90M   90M    0 100% /snap/core/8268
/dev/loop1       18M   18M    0 100% /snap/amazon-ssm-agent/1480
/dev/nvme1n1     20G   53M   20G  1% /additionalstorage
tmpfs            47M    0  47M  0% /run/user/1000
ubuntu@ip-172-31-43-11:~]$
```

Resource and dependency

➡ Example: Destroy: Delete EC2/EBS by terraform script

```
[ubuntu@ip-172-31-19-113:~/terraform_202003/Workshop_1.4_Provisioner$ terraform destroy -auto-approve
data.aws_ami.ubuntu: Refreshing state...
aws_ebs_volume.labebs: Refreshing state... [id=vol-0e5e772a20d140cea]
aws_instance.lab: Refreshing state... [id=i-0a16c706816ceadad]
aws_volume_attachment.labebsattach: Refreshing state... [id=vai-1096346766]
aws_volume_attachment.labebsattach: Destroying... [id=vai-1096346766]
aws_volume_attachment.labebsattach: Provisioning with 'local-exec'...
aws_volume_attachment.labebsattach (local-exec): Executing: ["/bin/sh" "-c" "echo 'umount done for destroy (event)'"']
aws_volume_attachment.labebsattach (local-exec): umount done for destroy (event)
aws_volume_attachment.labebsattach: Still destroying... [id=vai-1096346766, 10s elapsed]
aws_volume_attachment.labebsattach: Destruction complete after 10s
aws_instance.lab: Destroying... [id=i-0a16c706816ceadad]
aws_instance.lab: Provisioning with 'remote-exec'...
aws_ebs_volume.labebs: Destroying... [id=vol-0e5e772a20d140cea]

aws_ebs_volume.labebs: Destruction complete after 1s
aws_instance.lab: Still destroying... [id=i-0a16c706816ceadad, 10s elapsed]
aws_instance.lab: Still destroying... [id=i-0a16c706816ceadad, 20s elapsed]
aws_instance.lab: Still destroying... [id=i-0a16c706816ceadad, 30s elapsed]
aws_instance.lab: Still destroying... [id=i-0a16c706816ceadad, 40s elapsed]
aws_instance.lab: Still destroying... [id=i-0a16c706816ceadad, 50s elapsed]
aws_instance.lab: Destruction complete after 1m0s

Destroy complete! Resources: 3 destroyed.
ubuntu@ip-172-31-19-113:~/terraform_202003/Workshop_1.4_Provisioner$ ]
```

Resource and dependency

➡ Example: Destroy: Delete EC2/EBS by terraform script

DescribeInstances	
Reservations	
OwnerId	404075494050
ReservationId	r-005cdaf3409954e8c
Instances	
AmiLaunchIndex	0
Architecture	x86_64
ClientToken	
EbsOptimized	False
EnaSupport	True
Hypervisor	xen
ImageId	ami-09a4a9ce71ff3f20b
InstanceId	i-0a16c706816ceadad
InstanceType	t3a.nano
KeyName	keypair
LaunchTime	2020-01-21T16:21:29.000Z
PrivateDnsName	
PublicDnsName	
RootDeviceName	/dev/sda1
RootDeviceType	ebs
StateTransitionReason	User initiated (2020-01-21 16:30:35 GMT)
VirtualizationType	hvm
Monitoring	
State	disabled
Placement	
AvailabilityZone	ap-southeast-1c
GroupName	
Tenancy	default
State	
Code	48
Name	terminated

```
[ubuntu@ip-172-31-19-113:~/terraform_202003/Workshop_1.4_Provisioner$ aws ec2 describe-volumes --filter=VolumeIds=vol-0a16c706816ceadad
-----
|DescribeVolumes|
+-----+
```

Workshop: Provisioner



Workshop: Provisioner

```
ubuntu@ip-172-31-19-113:~/terraform_202003/Workshop_1.4_Provisioner$ terraform apply -auto-approve -parallelism=5 -refresh=true .
data.aws_ami.ubuntu: Refreshing state...
[aws_ebs_volume.labebs: Creating...
[aws_instance.lab: Creating...
aws_ebs_volume.labebs: Still creating... [10s elapsed]
aws_instance.lab: Still creating... [10s elapsed]
aws_ebs_volume.labebs: Creation complete after 10s [id=vol-0e5e772a20d140cea]
aws_instance.lab: Creation complete after 12s [id=i-0a16c706816ceadad]
aws_volume_attachment.labebsattach: Creating...
aws_volume_attachment.labebsattach: Still creating... [10s elapsed]
aws_volume_attachment.labebsattach: Still creating... [20s elapsed]
aws_volume_attachment.labebsattach: Provisioning with 'remote-exec'...
aws_volume_attachment.labebsattach (remote-exec): Connecting to remote host via SSH...
aws_volume_attachment.labebsattach (remote-exec): Host: 13.251.124.175
aws_volume_attachment.labebsattach (remote-exec): User: ubuntu
aws_volume_attachment.labebsattach (remote-exec): Password: false
aws_volume_attachment.labebsattach (remote-exec): Private key: true
aws_volume_attachment.labebsattach (remote-exec): Certificate: false
aws_volume_attachment.labebsattach (remote-exec): SSH Agent: false
aws_volume_attachment.labebsattach (remote-exec): Checking Host Key: false
aws_volume_attachment.labebsattach (remote-exec): Connected!
aws_volume_attachment.labebsattach (remote-exec): meta-data=/dev/nvme1n1      isize=512    agcount=4, agsize=1310720 blks
aws_volume_attachment.labebsattach (remote-exec):          =      sectsz=512    attr=2, projid32bit=1
aws_volume_attachment.labebsattach (remote-exec):          =      crc=1      finobt=1, sparse=0, rmapbt=0, reflink=0
aws_volume_attachment.labebsattach (remote-exec): data      =      bsize=4096   blocks=5242880, imaxpct=25
aws_volume_attachment.labebsattach (remote-exec):          =      sunit=0     swidth=0 blks
aws_volume_attachment.labebsattach (remote-exec): naming    =version 2   bsize=4096   ascii-ci=0 ftype=1
aws_volume_attachment.labebsattach (remote-exec): log       =internal log  bsize=4096   blocks=2560, version=2
aws_volume_attachment.labebsattach (remote-exec):          =      sectsz=512   sunit=0 blks, lazy-count=1
aws_volume_attachment.labebsattach (remote-exec): realtime  =none      extsz=4096   blocks=0, rtextents=0
aws_volume_attachment.labebsattach: Creation complete after 22s [id=vai-1096346766]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

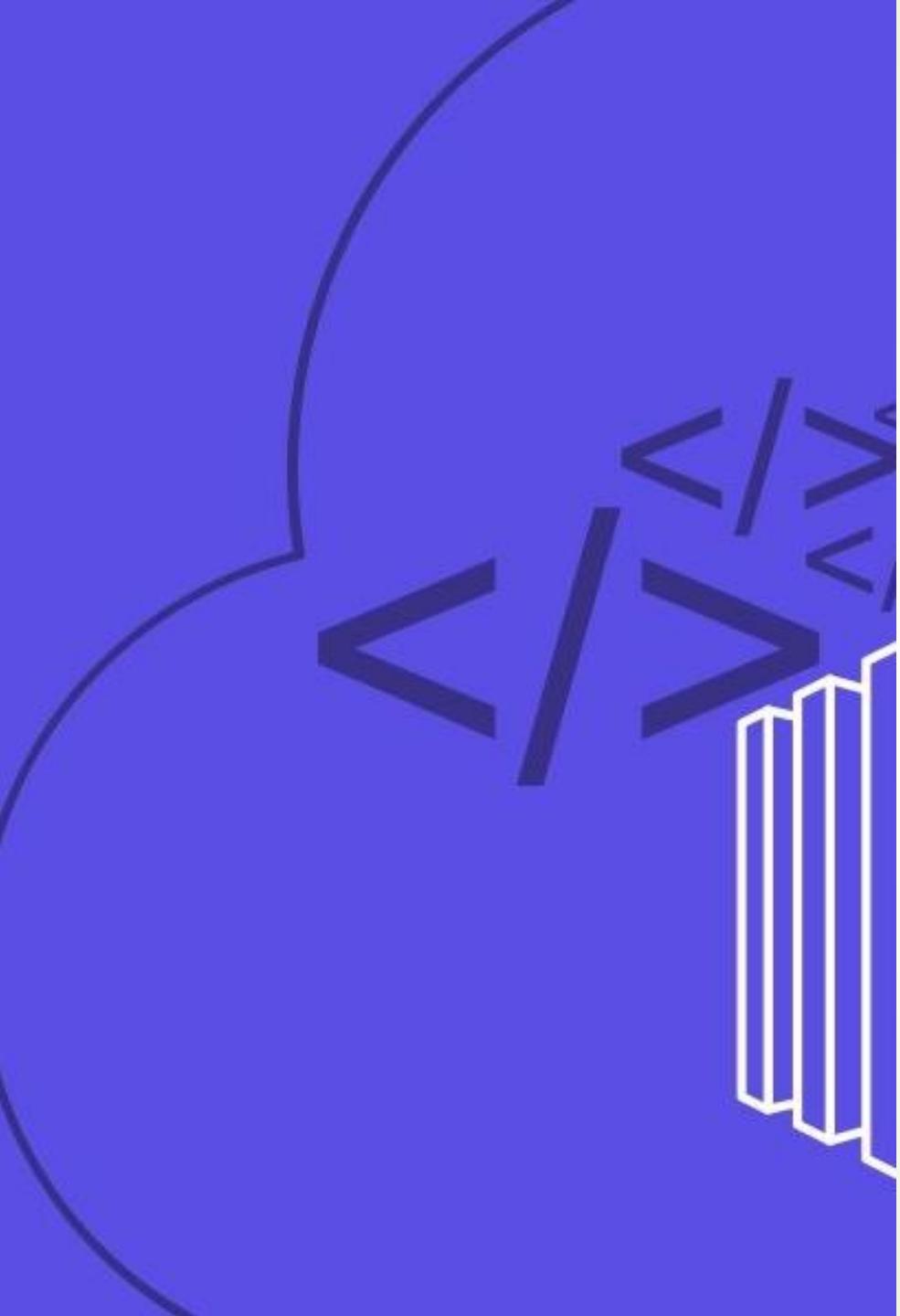
Outputs:

aws_ebs_volume_labebs_id = vol-0e5e772a20d140cea
aws_instance_lab_id = i-0a16c706816ceadad
aws_instance_lab_keyname = keypair
aws_instance_lab_public_dns = ec2-13-251-124-175.ap-southeast-1.compute.amazonaws.com
aws_instance_lab_public_ip = 13.251.124.175
ubuntu@ip-172-31-19-113:~/terraform_202003/Workshop_1.4_Provisioner$
```



Q&A





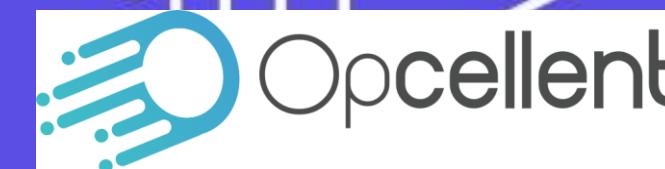
Agenda (Day 1)

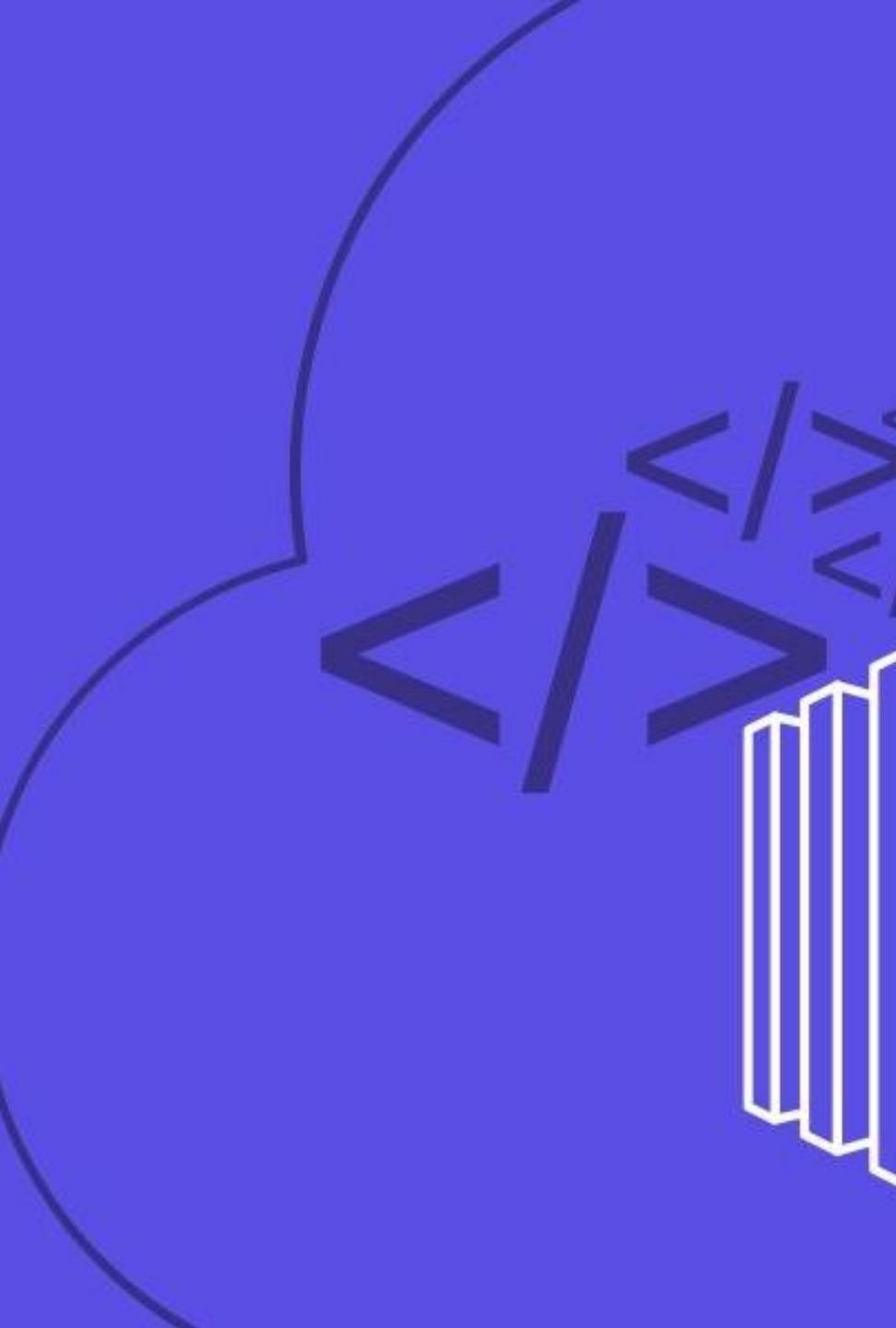
- ▶ Principle of IaC
 - ▶ Challenge to IT infrastructure today
 - ▶ Multiple cloud infrastructure
 - ▶ IaC well-known on the market
 - ▶ Orchestration vs Configuration Management
 - ▶ Feature and use-case on terraform
 - ▶ Terraform product
- ▶ Dive into HCL (Hashicorp Configuration Language) for terraform
 - ▶ Naming, Argument, Blocks and Expressions
 - ▶ Example file structure
 - ▶ System architecture
 - ▶ Terraform state management
 - ▶ Resource and dependency
 - ▶ Provisioner



Infrastructure as Code(IaC) with Terraform (Day2)

Praparn Luangphoonlap





Agenda (Day 2)

- ▶ Dive to HCL (Hashicorp Configuration Language) for terraform (Cont.)
 - ▶ Variable (Input & Output)
 - ▶ Registry and Module
- ▶ Terraform cloud for team collaboration
 - ▶ Introduction to terraform cloud
 - ▶ Setup workspace and test drive
- ▶ Team workshop (Final Workshop)
- ▶ Q&A

Variable (Input & Output)

► Do you familiar with this?

```
1  variable "region" {}
2  provider "aws" {
3    region = var.region
4  }
5
6  resource "aws_instance" "lab" {
7    ami           = "ami-81cefefcfd" #ubuntu-xenial-16.04-amd64-server
8    instance_type = "t3a.nano"
9    key_name = "keypair"
10   }
11
12  output "aws_instance_lab_id" {
13    value = aws_instance.lab.id
14  }
15
16  output "aws_instance_lab_public_ip" {
17    value = aws_instance.lab.public_ip
18  }
19
20  output "aws_instance_lab_public_dns" {
21    value = aws_instance.lab.public_dns
22  }
23
24  output "aws_instance_lab_keyname" [
25    value = aws_instance.lab.key_name
26  ]
```

Input Variable

- ▶ Terraform script have many value need to define for operate.
- ▶ Multiple script have duplicate value / reuse in many place in script
- ▶ Terraform support define input variable for this purpose
 - ▶ **Method1:** Command Line (-var)
 - ▶ **Method2:** Environment Variable
 - ▶ **Method3:** UI (Support only “String”)*Not recommend
 - ▶ **Method4:** File Variable (terraform.tfvars, terraform.tfvars.json, *.auto.tfvar, *.auto.tfvar.json)
- ▶ Define by command: -var
- ▶ Specific file by command: -var-file=

Input Variable

- ▶ Input variable can share many workspace for act like “global variable”
 - ▶ Terraform Cloud Space
 - ▶ Defined by “-var” command line option
 - ▶ Variable definition file “.tfvars” and automatic load or select by “-var-file”
 - ▶ Environment variable

Input Variable

- ▶ Type of constrain allow for variable
 - ▶ Basic type
 - ▶ String
 - ▶ Number
 - ▶ Boolean (bool)
 - ▶ Collection type
 - ▶ Lists (Set of value on same type, identify by number 0-x)
 - ▶ Maps (Set of value on same type, Identify by string)
 - ▶ Set (Set of unique value)
 - ▶ Structural type
 - ▶ Object (Set of value that have their own type)
 - ▶ Tuple (Set of value identify by sequence with their own type)

Input Variable

- ▶ String / Number / Boolean
 - ▶ Variable “vpc_id”
 - ▶ {type= **string**} //vpc_id=“vpc-4383e921”
- ▶ Variable “volume_size”
- ▶ {type=**number**} //volume_size=50
- ▶ Variable “delete_flag”
- ▶ {type= **bool**} //delete_flag=false

Input Variable

- ▶ List / Map

- ▶ Variable “vpc_cidr”
 - ▶ {type= **list**} // vpc_cidr = ["10.0.0.0/16", "10.1.0.0/16"]

- ▶ Variable “ec2_ip”

- ▶ {type=**map**}

- ▶ /*

- ▶ ec2_ip = {

- ▶ "i-0b689d3af307f4bdd" = "54.251.157.24"

- ▶ "i-0f1abda2ddd53824" = "54.169.78.6"

- ▶ }

- ▶ */

Input Variable

- ▶ Set
 - ▶ Variable “ec2_id”
 - ▶ {type= **set**} // ec2_id = [" i-0b689d3af307f4bdd ", " i-0f1abda2dddc53824 "]
- ▶ Object
 - ▶ Variable “route53_property”
 - ▶ {type= **object**}
 - ▶ /*
 - ▶ route53_property = {
 - ▶ name = “route53.terraform.local” (string)
 - ▶ public_soa= “58.97.109.43” (string)
 - ▶ ttl = 60 (number)
 - ▶ regional = “ap-southeast-1” (string)
 - ▶ delete_on_terminate = true (boolean)
 - ▶ */

Output Variable

- ▶ Terraform script can extract output/resource from provisioning for collect to use in next workspace via “output”.
- ▶ Output was declared by specific value from data/resource/looping <advance loop>
- ▶ All output was written on script and execute will operate and keep in terraform (state file) and it can exportable via command “terraform output”
- ▶ Output from terraform output can show in screen or export to file with basic pipe “terraform output >> <file name>”

Output Variable

```
1 resource "aws_instance" "labserver" {
2   count          = var.instancecount
3   ami            = data.aws_ami.ubuntu.id
4   instance_type = lookup(var.instance_type, var.server_type)
5   key_name       = var.keypair
6   availability_zone = var.availability_zone
7   root_block_device {
8     volume_type      = var.rootblockdevice_volume_type
9     volume_size      = var.rootblockdevice_volume_size
10    delete_on_termination = var.rootblockdevice_delete_on_termination
11  }
12  vpc_security_group_ids = [aws_security_group.secgroup_server.id]
13  user_data           = lookup(var.user_data, count.index)
14  tags = {
15    Environment = var.tag_environment
16    Region     = var.tag_region
17    AZ          = var.tag_az
18    Categories = var.tag_category
19    Name        = lookup(var.tag_name, count.index)
20    Zone        = var.tag_zone
21    Module      = var.tag_module
22    Billing     = var.tag_billing
23  }
24 }
25
26 output "aws_instance_labserver_ip" {
27   value = {
28     for instance in aws_instance.labserver:
29       instance.id => instance.public_ip
30   }
31 }
```

Output Variable

```
[ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/01_ec2_instance$ terraform apply -auto-approve -parallelism=5 -refresh=true -var-file=../global_variable.auto.tfvars
s ./
data.aws_ami.ubuntu: Refreshing state...
data.aws_vpc.labvpc: Refreshing state...
aws_security_group.secgroup_server: Creating...
aws_security_group.secgroup_server: Creation complete after 1s [id=sg-0205c8bc062f4b183]
aws_instance.labserver[0]: Creating...
aws_instance.labserver[1]: Creating...
aws_instance.labserver[0]: Still creating... [10s elapsed]
aws_instance.labserver[1]: Still creating... [10s elapsed]
aws_instance.labserver[0]: Creation complete after 12s [id=i-071aca1891566ad7a]
aws_instance.labserver[1]: Creation complete after 12s [id=i-0266f98b2582bbe2b]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Outputs:

aws_instance_labserver_ip = {
  "i-0266f98b2582bbe2b" = "54.255.219.157"
  "i-071aca1891566ad7a" = "13.229.214.197"
}
aws_vpc_cidr = 172.31.0.0/16
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/01_ec2_instance$ ]
```

```
Outputs:

aws_instance_labserver_ip = {
  "i-0da8541014abb05c2" = "54.255.128.121"
  "i-0f0419f97be843d2d" = "54.254.238.230"
}
aws_vpc_cidr = 172.31.0.0/16
[ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/01_ec2_instance$ terraform output >> ../global_variable.auto.tfvars
```

```
[ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/02_route53$ more ../global_variable.auto.tfvars
vpc_id="vpc-4483e921"
region = "ap-southeast-1"
availability_zone="ap-southeast-1a"
zone_id="Z10002633BJEGNCUNP41E"
aws_instance_labserver_ip = {
  "i-0da8541014abb05c2" = "54.255.128.121"
  "i-0f0419f97be843d2d" = "54.254.238.230"
}
```

Input&Output Variable (Loop)

► Count

- In terraform we can define looping with specific number of loop (Same for `(i=1; i<=10; i++)`)

- Reference from count number in each loop for select some property to operate

► For_Each (New on 0.12)

- Loop by variable type “List” or “Map”

- Operate for each member in variable and get key/value to provision

► Each (New on 0.12)

- Use to loop output that type of List or Map/List

Input&Output Variable (Loop)

► Count / Output

```
Users > praparnlueangphoonlap > Work > Terraform > terraform_202003 > Workshop_2.1_InputVariable  
1 resource "aws_instance" "labserver" {  
2   count          = var.instancecount  
3   ami            = data.aws_ami.ubuntu.id  
4   instance_type = lookup(var.instance_type, var.server_type)  
5   key_name       = var.keypair  
6   availability_zone = var.availability_zone  
7   root_block_device {  
8     volume_type    = var.rootblockdevice_volume_type  
9     volume_size     = var.rootblockdevice_volume_size  
10    delete_on_termination = var.rootblockdevice_delete_on_termination  
11  }  
12  vpc_security_group_ids = [aws_security_group.secgroup_server.id]  
13  user_data          = lookup(var.user_data, count.index)  
14  tags = {  
15    Environment = var.tag_environment  
16    Region      = var.tag_region  
17    AZ           = var.tag_az  
18    Categories   = var.tag_category  
19    Name         = lookup(var.tag_name, count.index)  
20    Zone         = var.tag_zone  
21    Module        = var.tag_module  
22    Billing        = var.tag_billing  
23  }  
24}  
25  
26 output "aws_instance_labserver_ip" {  
27   value = {  
28     for instance in aws_instance.labserver:  
29       instance.id => instance.public_ip  
30   }  
31 }
```

```
Users > praparnlueangphoonlap > Work > Terraform > terraform_202003  
1   instance_type = {  
2     "nginx" = "t3a.nano"  
3     "docker" = "t2a.nano"  
4     "other" = "t3a.nano"  
5   }  
6   server_type = "nginx"  
7   keypair = "keypair"  
8   instancecount=2  
9   rootblockdevice_volume_type = "gp2"  
10  rootblockdevice_volume_size = 10  
11  rootblockdevice_delete_on_termination = true  
12  tag_environment="rd"  
13  tag_region="ap-southeast-1"  
14  tag_az="ap-southeast-1a"  
15  tag_category="compute"  
16  tag_name={  
17    "0"    = "labresource-server1-student-X"  
18    "1"    = "labresource-server2-student-X"  
19  }  
20  tag_zone="public"  
21  tag_module="ec2"  
22  tag_billing="terraform-workshop"  
23 }
```

Input&Output Variable (Loop)

► For_Each / Output

```
[ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/02_route53$ more ..../global_variable.auto.tfvars
vpc_id="vpc-4483e921"
region = "ap-southeast-1"
availability_zone="ap-southeast-1a"
zone_id="Z10002633BJEGNCUNP41E"
aws_instance_labserver_ip = {
    "i-0da8541014abb05c2" = "54.255.128.121"
    "i-0f0419f97be843d2d" = "54.254.238.230"
}
```

```
1 resource "aws_route53_record" "lab_route53" {
2   for_each = var.aws_instance_labserver_ip
3   zone_id = var.zone_id
4   name    = each.key
5   type    = "A"
6   ttl     = "30"
7   records = [each.value]
8 }
9 output "aws_route53_fqdn" {
10   value = {
11     for dns in aws_route53_record.lab_route53:
12       dns.name => dns.fqdn
13   }
14 }
```

```
Outputs:
aws_route53_fqdn = {
    "i-0da8541014abb05c2" = "i-0da8541014abb05c2.terraform.local"
    "i-0f0419f97be843d2d" = "i-0f0419f97be843d2d.terraform.local"
}
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/02_route53$
```

Variable (Input & Output)

► **Example:** Rewrite this script to use variable for operate

```
1  variable "region" {}
2  provider "aws" {
3    | |   region = var.region
4  }
5
6  resource "aws_instance" "lab" {
7    | |     ami           = "ami-81cefefcf" #ubuntu-xenial-16.04-amd64-server
8    | |     instance_type = "t3a.nano"
9    | |     key_name      = "keypair"
10 }
11
12 output "aws_instance_lab_id" {
13   | value = aws_instance.lab.id
14 }
15
16 output "aws_instance_lab_public_ip" {
17   | value = aws_instance.lab.public_ip
18 }
19
20 output "aws_instance_lab_public_dns" {
21   | value = aws_instance.lab.public_dns
22 }
23
24 output "aws_instance_lab_keyname" [
25   | value = aws_instance.lab.key_name
26 ]
```

Variable (Input & Output)

► **Example:** Rewrite this script to use variable for operate

```
1  variable "region" {}
2  provider "aws" {
3    | | region = var.region
4  }
5
6  resource "aws_instance" "lab" {
7    | ami           = "ami-81cefef" #ubuntu-xenial-16.04-amd64-server
8    | instance_type = "t3a.nano"
9    | key_name      = "keypair"
10   }
11
12  output "aws_instance_lab_id" {
13    | value = aws_instance.lab.id
14  }
15
16  output "aws_instance_lab_public_ip" {
17    | value = aws_instance.lab.public_ip
18  }
19
20  output "aws_instance_lab_public_dns" {
21    | value = aws_instance.lab.public_dns
22  }
23
24  output "aws_instance_lab_keyname" [
25    | value = aws_instance.lab.key_name
26  ]
```

```
1  provider "aws" {
2    | | region = var.region
3  }
4
5  resource "aws_instance" "lab" {
6
7    | ami           = var.ami #ubuntu-xenial-16.04-amd64-server
8    | instance_type = varinstancetype
9    | key_name      = var.keypair
10   }
11
12  output "aws_instance_lab_id" [
13    | value = aws_instance.lab.id
14  ]
15
16  output "aws_instance_lab_public_ip" {
17    | value = aws_instance.lab.public_ip
18  }
19
20  output "aws_instance_lab_public_dns" {
21    | value = aws_instance.lab.public_dns
22  }
23
24  output "aws_instance_lab_keyname" {
25    | value = aws_instance.lab.key_name
26  }
```

Variable (Input & Output)

► **Example:** Rewrite this script to use variable for operate

```
cd /Users/priyanshu/Work/Terraform/terraform_202008/Workshop_2.1_Input  
1 #####  
2 # Define Variable to Operate #####  
3  
4 variable "region" {  
5   type = string  
6   description = "Defined region for operate"  
7 }  
8  
9 variable "ami" {  
10  type = string  
11  description = "Defined AMI ID"  
12 }  
13  
14 variable "instancetype" {  
15  type = string  
16  description = "Defined EC2 Instance Type"  
17 }  
18  
19 variable "keypair" {  
20  type = string  
21  description = "Keypair for Authentication"  
22 }  
23
```

Variable (Input & Output)

► Example (Command Line):

```
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/01_commandline_variable$ terraform init

Initializing the backend...

Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "aws" (hashicorp/aws) 2.46.0...

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.aws: version = "~> 2.46"

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.
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/01_commandline_variable$     terraform fmt -check
01_commandline_resource.tf
variables.tf
[ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/01_commandline_variable$     terraform validate
Success! The configuration is valid.
```

Variable (Input & Output)

► Example (Command Line):

```
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/01_commandline_variable$ terraform apply -auto-approve -parallelism=5 \
>     -var 'region=ap-southeast-1' \
>     -var 'ami=ami-81cefccfd' \
>     -var 'instancetype=t3a.nano' \
>     -var 'keypair=keypair' \
[>     -refresh=true ./
aws_instance.lab: Creating...
aws_instance.lab: Still creating... [10s elapsed]
aws_instance.lab: Creation complete after 12s [id=i-0eeb2339131d2f6ce]

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

Outputs:

aws_instance_lab_id = i-0eeb2339131d2f6ce
aws_instance_lab_keyname = keypair
aws_instance_lab_public_dns = ec2-13-251-110-185.ap-southeast-1.compute.amazonaws.com
aws_instance_lab_public_ip = 13.251.110.185
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/01_commandline_variable$ █
```

Variable (Input & Output)

► Example (Command Line):

DescribeInstances	
Reservations	
OwnerId	404075494050
ReservationId	r-86c6c6a226d0d8660
Instances	
AmiLaunchIndex	0
Architecture	x86_64
ClientToken	
EbsOptimized	False
EnaSupport	True
Hypervisor	xen
ImageId	ami-81cefefcfd
InstanceId	i-0eeb2339131d2f6ce
InstanceType	t3a.nano
KeyName	keypair
LaunchTime	2020-01-29T12:15:48.000Z
PrivateDnsName	ip-172-31-40-238.ap-southeast-1.compute.internal
PrivateIpAddress	172.31.40.238
PublicDnsName	ec2-13-251-110-185.ap-southeast-1.compute.amazonaws.com
PublicIpAddress	13.251.110.185
RootDeviceName	/dev/sda1
RootDeviceType	ebs
SourceDestCheck	True
StateTransitionReason	
SubnetId	subnet-5a1e2c1c
VirtualizationType	hvm
VpcId	vpc-4483e921
BlockDeviceMappings	
DeviceName	/dev/sda1
Ebs	
AttachTime	2020-01-29T12:15:49.000Z
DeleteOnTermination	True
Status	attached
VolumeId	vol-0d471f71870cd5cd2
CapacityReservationSpecification	
CapacityReservationPreference	open
CpuOptions	
CoreCount	1
ThreadsPerCore	2
HibernationOptions	
Configured	False
MetadataOptions	
HttpEndpoint	enabled
HttpPutResponseHopLimit	1
HttpTokens	optional
State	applied
Monitoring	
State	disabled
NetworkInterfaces	
Description	interface
InterfaceType	ebs
MacAddress	0:a:b9:95:8e:bd:16
NetworkInterfaceId	eni-03e884b1e02975fab
OwnerId	404075494050
PrivateDnsName	ip-172-31-40-238.ap-southeast-1.compute.internal
PrivateIpAddress	172.31.40.238
SourceDestCheck	True
Status	in-use
SubnetId	subnet-5a1e2c1c
VpcId	vpc-4483e921
Association	
IpOwnerId	amazon
PublicDnsName	ec2-13-251-110-185.ap-southeast-1.compute.amazonaws.com
PublicIp	13.251.110.185
Attachment	
AttachTime	2020-01-29T12:15:48.000Z
AttachmentId	eni-attach-03bf99a863f7cc3b2
DeleteOnTermination	True
DeviceIndex	0
Status	attached
SecurityGroups	
GroupId	sg-fe2d3c9b
GroupName	default
State	
Code	16
Name	running

Variable (Input & Output)

▶ Example (Command Line):

```
[praparns-MacBook-Pro:~] praparns$ ssh praparns@13.251.110.185
The authenticity of host '13.251.110.185 (13.251.110.185)' can't be established.
ECDSA key fingerprint is SHA256:b4NodsQDD7D8ikPMZYWVUDp+o0Qfcad625uiP/Q6CPg.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '13.251.110.185' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-1060-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 Get cloud support with Ubuntu Advantage Cloud Guest:
   http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

-----
WARNING! Your environment specifies an invalid locale.
The unknown environment variables are:
  LC_CTYPE=UTF-8  LC_ALL=
This can affect your user experience significantly, including the
ability to manage packages. You may install the locales by running:

    sudo apt-get install language-pack-UTF-8
    or
    sudo locale-gen UTF-8

To see all available language packs, run:
    apt-cache search "^language-pack-[a-z][a-z]$"
To disable this message for all users, run:
    sudo touch /var/lib/cloud/instance/locale-check.skip
```

Variable (Input & Output)

► Example (Command Line):

```
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/01_commandline_variable$ terraform destroy -auto-approve \
>     -var 'region=ap-southeast-1' \
>     -var 'ami=ami-81cefcfd' \
>     -var 'instancetype=t3a.nano' \
>     -var 'keypair=keypair' \
[>     -refresh=true ./
aws_instance.lab: Refreshing state... [id=i-0eeb2339131d2f6ce]
aws_instance.lab: Destroying... [id=i-0eeb2339131d2f6ce]
aws_instance.lab: Still destroying... [id=i-0eeb2339131d2f6ce, 10s elapsed]
aws_instance.lab: Still destroying... [id=i-0eeb2339131d2f6ce, 20s elapsed]
aws_instance.lab: Still destroying... [id=i-0eeb2339131d2f6ce, 30s elapsed]
aws_instance.lab: Still destroying... [id=i-0eeb2339131d2f6ce, 40s elapsed]
aws_instance.lab: Destruction complete after 50s

Destroy complete! Resources: 1 destroyed.
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/01_commandline_variable$ █
```

Variable (Input & Output)

► Example (Command Line):

DescribeInstances	
Reservations	
OwnerId	404075494050
ReservationId	r-06c6c6a226d0d8660
Instances	
AmiLaunchIndex	0
Architecture	x86_64
ClientToken	
EbsOptimized	False
EnaSupport	True
Hypervisor	xen
ImageId	ami-81cefefcd
InstanceId	i-0eeb239131d2f6ce
InstanceType	t3a.nano
KeyName	keypair
LaunchTime	2020-01-29T12:15:48.000Z
PrivateDnsName	
PublicDnsName	
RootDeviceName	/dev/sda1
RootDeviceType	ebs
StateTransitionReason	User initiated (2020-01-29 12:24:47 GMT)
VirtualizationType	hvm
CapacityReservationSpecification	
CapacityReservationPreference	open
CpuOptions	
CoreCount	1
ThreadsPerCore	2
HibernationOptions	
Configured	False
MetadataOptions	
HttpEndpoint	enabled
HttpPutResponseHopLimit	1
HttpTokens	optional
State	pending
Monitoring	
State	disabled

Placement	
AvailabilityZone	ap-southeast-1c
GroupName	
Tenancy	default
State	
Code	48
Name	terminated
StateReason	
Code	Client.UserInitiatedShutdown
Message	Client.UserInitiatedShutdown: User initiated shutdown

Variable (Input & Output)

► Example (Environment Variable):

```
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$ export TF_VAR_region=ap-southeast-1
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$      export TF_VAR_ami=ami-81cefcfd
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$      export TF_VAR_instancetype=t3a.nano
[ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$      export TF_VAR_keypair=keypair
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$ echo $TF_VAR_region
ap-southeast-1
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$      echo $TF_VAR_ami
ami-81cefcfd
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$      echo $TF_VAR_instancetype
t3a.nano
[ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$      echo $TF_VAR_keypair
keypair
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$ terraform init

Initializing the backend...

Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "aws" (hashicorp/aws) 2.46.0...

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.aws: version = "~> 2.46"

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.
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$      terraform fmt -check
02_environment_resource.tf
variables.tf
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$      terraform validate
Success! The configuration is valid.
```

Variable (Input & Output)

► Example (Environment Variable):

```
[ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$ terraform apply -auto-approve -parallelism=5 -refresh=true .
aws_instance.lab: Creating...
aws_instance.lab: Still creating... [10s elapsed]
aws_instance.lab: Creation complete after 12s [id=i-0eef1cdfe700ec0b7]

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

Outputs:

aws_instance_lab_id = i-0eef1cdfe700ec0b7
aws_instance_lab_keyname = keypair
aws_instance_lab_public_dns = ec2-13-251-42-135.ap-southeast-1.compute.amazonaws.com
aws_instance_lab_public_ip = 13.251.42.135
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$ ]
```

Variable (Input & Output)

► Example (Environment Variable):

```
[praparns-MacBook-Pro:~ ssh praparn$ ssh -i terraformlab ubuntu@13.251.42.135
The authenticity of host '13.251.42.135 (13.251.42.135)' can't be established.
ECDSA key fingerprint is SHA256:TZpy0Nc/dNaIpeanI51Pedp3sq9h+MUzwP5TJcDY/Ng.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '13.251.42.135' (ECDSA) to the list of known hosts.

Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-1060-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

WARNING! Your environment specifies an invalid locale.
The unknown environment variables are:
LC_CTYPE=UTF-8 LC_ALL=
This can affect your user experience significantly, including the
ability to manage packages. You may install the locales by running:

  sudo apt-get install language-pack-UTF-8
  or
  sudo locale-gen UTF-8

To see all available language packs, run:
  apt-cache search "^language-pack-[a-z][a-z]$"
To disable this message for all users, run:
  sudo touch /var/lib/cloud/instance/locale-check.skip
```

Variable (Input & Output)

► Example (Environment Variable):

```
[ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$ terraform destroy -auto-approve
aws_instance.lab: Refreshing state... [id=i-0eef1cdfe700ec0b7]
aws_instance.lab: Destroying... [id=i-0eef1cdfe700ec0b7]
aws_instance.lab: Still destroying... [id=i-0eef1cdfe700ec0b7, 10s elapsed]
aws_instance.lab: Still destroying... [id=i-0eef1cdfe700ec0b7, 20s elapsed]
aws_instance.lab: Still destroying... [id=i-0eef1cdfe700ec0b7, 30s elapsed]
aws_instance.lab: Destruction complete after 39s

Destroy complete! Resources: 1 destroyed.
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$ █
```

```
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$ export TF_VAR_region=
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$ export TF_VAR_ami=
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$ export TF_VAR_instancetype=
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$ export TF_VAR_keypair=
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$ echo $TF_VAR_region

ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$ echo $TF_VAR_ami

ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$ echo $TF_VAR_instancetype

[ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$ echo $TF_VAR_keypair

ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/02_environment_variable$ █
```

Variable (Input & Output)

► Example (File Variable):

```
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/03_file_variable$ terraform init  
Initializing the backend...
```

```
Initializing provider plugins...
```

- Checking for available provider plugins...
- Downloading plugin for provider "aws" (hashicorp/aws) 2.46.0...

```
The following providers do not have any version constraints in configuration,  
so the latest version was installed.
```

```
To prevent automatic upgrades to new major versions that may contain breaking  
changes, it is recommended to add version = "..." constraints to the  
corresponding provider blocks in configuration, with the constraint strings  
suggested below.
```

```
* provider.aws: version = "~> 2.46"
```

```
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.
```

```
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/03_file_variable$ terraform fmt -check  
03_file_resource.tf  
variables.tf
```

```
[ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/03_file_variable$ terraform validate  
Success! The configuration is valid.
```

```
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/03_file_variable$ █
```

Variable (Input & Output)

► Example (File Variable):

```
[ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/03_file_variable$ terraform apply -auto-approve -parallelism=5 -refresh=true ./aws_instance.lab: Creating...aws_instance.lab: Still creating... [10s elapsed]aws_instance.lab: Creation complete after 12s [id=i-0afea74194b4e37f3]

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

Outputs:

aws_instance_lab_id = i-0afea74194b4e37f3
aws_instance_lab_keyname = keypair
aws_instance_lab_public_dns = ec2-18-138-224-169.ap-southeast-1.compute.amazonaws.com
aws_instance_lab_public_ip = 18.138.224.169
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/03_file_variable$ ]
```

Variable (Input & Output)

► Example (File Variable):

```
praparns-MacBook-Pro:~ ssh praparns@18.138.224.169
The authenticity of host '18.138.224.169' (18.138.224.169) can't be established.
ECDSA key fingerprint is SHA256:y8Zs7P1EiFva2NvzGz/03PzbeF2AuVZkqwwuQoeFmdY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '18.138.224.169' (ECDSA) to the list of known hosts.

Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-1060-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 Get cloud support with Ubuntu Advantage Cloud Guest:
   http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

-----  
WARNING! Your environment specifies an invalid locale.
The unknown environment variables are:
  LC_CTYPE=UTF-8 LC_ALL=
This can affect your user experience significantly, including the
ability to manage packages. You may install the locales by running:

  sudo apt-get install language-pack-UTF-8
  or
  sudo locale-gen UTF-8

To see all available language packs, run:
  apt-cache search "^language-pack-[a-z][a-z]$"
To disable this message for all users, run:
  sudo touch /var/lib/cloud/instance/locale-check.skip
```

ubuntu@ip-172-31-44-28:~\$ █

Variable (Input & Output)

► Example (File Variable):

```
[ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/03_file_variable$ terraform destroy -auto-approve
aws_instance.lab: Refreshing state... [id=i-0afea74194b4e37f3]
aws_instance.lab: Destroying... [id=i-0afea74194b4e37f3]
aws_instance.lab: Still destroying... [id=i-0afea74194b4e37f3, 10s elapsed]
aws_instance.lab: Destruction complete after 20s

Destroy complete! Resources: 1 destroyed.
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/03_file_variable$ ]
```

Variable (Input & Output)

Resource

aws_security_group_rule1

Ingress Rule:
Allow from x.x.x.x with yyyy

Egress Rule:
Allow from x.x.x.x with yyyy

Security_group_id

Resource:

aws_instance1

User data: <script setup nginx>

Tag ec2 by standard tag

Resource:

aws_route53 (dns)

Name:

Tag ec2 by standard tag

Data:

aws_vpc

Vpc ID: XXX

Data:

aws_ami

AMI ID: XXX

Security_group_id

Resource:

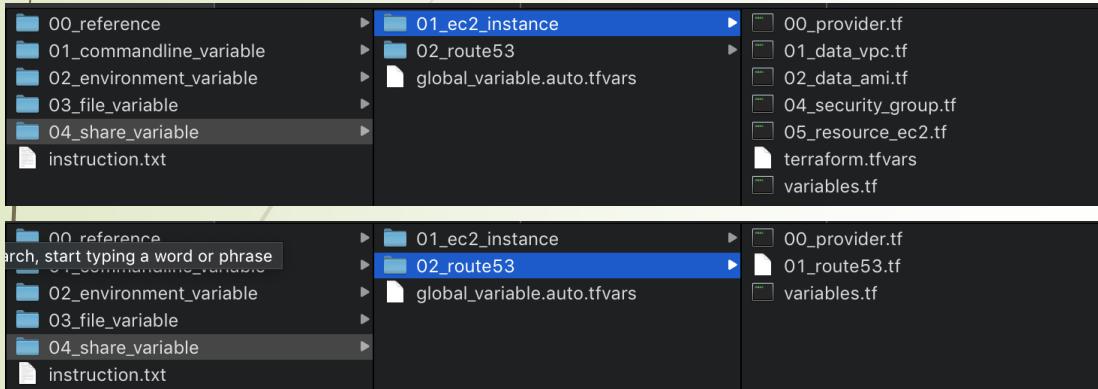
aws_instance2

User data: <script setup nginx>

Tag ec2 by standard tag

Variable (Input & Output)

➡ Example (Share Variable): Rewrite Script



```
ormCLI   instruction.txt ~.../Workshop_2.1_InputVariable  global_variable.auto.tfvars X
Users > paparnlueangphoonlap > Work > Terraform > terraform_202003 > Workshop_2.1_InputVar
1   vpc_id="vpc-4483e921"
2   region = "ap-southeast-1"
3   availability_zone="ap-southeast-1a"
4   zone_id="Z10002633BJEGNCUNP41E"
5
```

```
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/02_route53$ terraform apply -auto-approve -parallelism=5 -refresh=true -var-file=../global_variable.auto.tfvars ./
aws_route53_record.lab_route53["i-0f1abda2dddc53824"]: Creating...
aws_route53_record.lab_route53["i-0b689d3af307f4bdd"]: Creating...
aws_route53_record.lab_route53["i-0b689d3af307f4bdd"]: Still creating... [10s elapsed]
aws_route53_record.lab_route53["i-0f1abda2dddc53824"]: Still creating... [10s elapsed]
aws_route53_record.lab_route53["i-0b689d3af307f4bdd"]: Still creating... [20s elapsed]
aws_route53_record.lab_route53["i-0b689d3af307f4bdd"]: Still creating... [20s elapsed]
aws_route53_record.lab_route53["i-0f1abda2dddc53824"]: Still creating... [30s elapsed]
aws_route53_record.lab_route53["i-0b689d3af307f4bdd"]: Still creating... [30s elapsed]
aws_route53_record.lab_route53["i-0f1abda2dddc53824"]: Still creating... [40s elapsed]
aws_route53_record.lab_route53["i-0b689d3af307f4bdd"]: Still creating... [40s elapsed]
aws_route53_record.lab_route53["i-0f1abda2dddc53824"]: Still creating... [50s elapsed]
aws_route53_record.lab_route53["i-0b689d3af307f4bdd"]: Still creating... [50s elapsed]
aws_route53_record.lab_route53["i-0f1abda2dddc53824"]: Still creating... [1m0s elapsed]
aws_route53_record.lab_route53["i-0b689d3af307f4bdd"]: Still creating... [1m0s elapsed]
aws_route53_record.lab_route53["i-0f1abda2dddc53824"]: Creation complete after 1m4s [id=Z10002633BJEGNCUNP41E_i-0f1abda2dddc53824_A]
aws_route53_record.lab_route53["i-0b689d3af307f4bdd"]: Creation complete after 1m4s [id=Z10002633BJEGNCUNP41E_i-0b689d3af307f4bdd_A]
```

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

```
aws_route53_fqdn = {
  "i-0b689d3af307f4bdd" = "i-0b689d3af307f4bdd.terraform.local"
  "i-0f1abda2dddc53824" = "i-0f1abda2dddc53824.terraform.local"
```

Variable (Input & Output)

► Example (Share Variable): Rewrite Script (EC2)

```
1 resource "aws_instance" "labserver1" {
2   ami           = data.aws_ami.ubuntu.id
3   instance_type = "t3a.nano"
4   key_name      = "keypair"
5   availability_zone = "ap-southeast-1a"
6   root_block_device {
7     volume_type      = "gp2"
8     volume_size       = "10"
9     delete_on_termination = "true"
10  }
11 vpc_security_group_ids = [aws_security_group.secgroup_server1.id]
12 user_data              = <<<-EOF
13 #!/bin/bash
14 curl https://raw.githubusercontent.com/praparn/sourcesetup/master/standard_nginx.sh > /tmp/setup.sh
15 chmod +x /tmp/setup.sh
16 /tmp/setup.sh
17 EOF
18
19 tags = {
20   Environment = "rd"
21   Region      = "ap-southeast-1"
22   AZ          = "ap-southeast-1a"
23   Categories  = "compute"
24   Name        = "labresource-server1-student-X"
25   Zone        = "public"
26   Module      = "ec2"
27   Billing     = "terraform-workshop"
28 }
29
30
31 resource "aws_instance" "labserver2" {
32   ami           = data.aws_ami.ubuntu.id
33   instance_type = "t3a.nano"
34   key_name      = "keypair"
35   availability_zone = "ap-southeast-1a"
36   root_block_device {
37     volume_type      = "gp2"
38     volume_size       = "10"
39     delete_on_termination = "true"
40  }
41 vpc_security_group_ids = [aws_security_group.secgroup_server2.id]
42 user_data              = <<<-EOF
43 #!/bin/bash
44 curl https://raw.githubusercontent.com/praparn/sourcesetup/master/standard_docker_aws.sh > /tmp/setup.sh
45 chmod +x /tmp/setup.sh
46 /tmp/setup.sh
47 EOF
48 }
```

```
1 resource "aws_instance" "labserver" {
2   count           = var.instancecount
3   ami             = data.aws_ami.ubuntu.id
4   instance_type   = lookup(var.instance_type, var.server_type)
5   key_name        = var.keypair
6   availability_zone = var.availability_zone
7   root_block_device {
8     volume_type      = var.rootblockdevice_volume_type
9     volume_size       = var.rootblockdevice_volume_size
10    delete_on_termination = var.rootblockdevice_delete_on_termination
11  }
12 vpc_security_group_ids = [aws_security_group.secgroup_server.id]
13 user_data           = lookup(var.user_data, count.index)
14 tags = {
15   Environment = var.tag_environment
16   Region      = var.tag_region
17   AZ          = var.tag_az
18   Categories  = var.tag_category
19   Name        = lookup(var.tag_name, count.index)
20   Zone        = var.tag_zone
21   Module      = var.tag_module
22   Billing     = var.tag_billing
23 }
24
25
26 output "aws_instance_labserver_ip" [
27   value = {
28     for instance in aws_instance.labserver:
29       instance.id => instance.public_ip
30   }
31 ]
```

Variable (Input & Output)

► Example (Share Variable): Rewrite Script (EC2)

```
Users > paparnlueangphoonlap > Work > Terraform > terraform_202003 > Workshop_2.1_InputVariable > 04_share_variable > 01_ec2_instance > terraform.tfvars
1  instance_type = {
2    "nginx" = "t3a.nano"
3    "docker" = "t2a.nano"
4    "other" = "t3a.nano"
5  }
6  server_type = "nginx"
7  keypair = "keypair"
8  instancecount=2
9  rootblockdevice_volume_type = "gp2"
10 rootblockdevice_volume_size = 10
11 rootblockdevice_delete_on_termination = true
12 tag_environment="rd"
13 tag_region="ap-southeast-1"
14 tag_az="ap-southeast-1a"
15 tag_category="compute"
16 tag_name={
17   "0" = "labresource-server1-student-X"
18   "1" = "labresource-server2-student-X"
19 }
20 tag_zone="public"
21 tag_module="ec2"
22 tag_billing="terraform-workshop"
23
24 user_data = {
25   "0" = <<-EOF
26     #!/bin/bash
27     curl https://raw.githubusercontent.com/paparn/sourcesetup/master/standard\_nginx.sh > /tmp/setup.sh
28     chmod +x /tmp/setup.sh
29     /tmp/setup.sh
30     EOF
31   "1"= <<-EOF
32     #!/bin/bash
33     curl https://raw.githubusercontent.com/paparn/sourcesetup/master/standard\_docker\_aws.sh > /tmp/setup.sh
34     chmod +x /tmp/setup.sh
35     /tmp/setup.sh
36     EOF
37 }
```

Variable (Input & Output)

➡ Example (Share Variable): Rewrite Script (EC2)

```
file:   Untitled-1 •  terraform.tfvars  variables.tf ×  □
Users > paparnlueangphoonlap > Work > Terraform > terraform_202003 > Worksh
1 ##### Define Variable to Operate #####
2 #           Define Variable to Operate      #
3 #####
4 variable "vpc_id" {
5   type = string
6   description = "VPC ID"
7 }
8
9 variable "zone_id" {
10  type = string
11  description = "DNS Zone ID"
12 }
13
14 variable "region" {
15  type = string
16  description = "Defined region for operate"
17 }
18
19 variable "instancecount" {
20  type = number
21  description = "Total instance to create"
22 }
23
24 variable "server_type" {
25  type = string
26  description = "Defined server type for operate"
27  default= "other"
28 }
29
30 variable "instance_type" {
31  type = map
32  description = "Defined EC2 instance type"
33 }
34
35 variable "user_data" []
36  type = map
37  description = "User data script"
38 }
39
```

```
40 variable "availability_zone" {
41   type = string
42   description = "Availability zone for operate"
43 }
44
45 variable "keypair" {
46   type = string
47   description = "Keypair for authentication"
48 }
49
50 variable "rootblockdevice_volume_type" {
51   type = string
52   description = "Define volume type of root stroage"
53 }
54
55 variable "rootblockdevice_volume_size" {
56   type = number
57   description = "Define volume size of root storage"
58 }
59
60 variable "rootblockdevice_delete_on_termination" {
61   type = bool
62   description = "Flag for delete storage when ec2 delet
63 }
64
65 variable "tag_environment" {
66   type = string
67   description = "Define Environment for Operate"
68 }
69
70 variable "tag_region" {
71   type = string
72   description = "Define region for operate"
73 }
74
75 variable "tag_az" {
76   type = string
77   description = "Define availability zone for operate"
78 }
```

```
79
80 variable "tag_category" {
81   type = string
82   description = "Define category of object to create"
83 }
84
85 variable "tag_name" {
86   type = map
87   description = "Define name of ec2"
88 }
89
90 variable "tag_zone" {
91   type = string
92   description = "Define public or private zone"
93 }
94
95 variable "tag_module" {
96   type = string
97   description = "Define module to create"
98 }
99
100 variable "tag_billing" {
101  type = string
102  description = "Define billing project to operate"
103 }
```

Variable (Input & Output)

➡ Example (Share Variable): Rewrite Script (EC2)

```
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/01_ec2_instance$ terraform init
Initializing the backend...
Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "aws" (hashicorp/aws) 2.46.0...
The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.aws: version = "~> 2.46"

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.
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/01_ec2_instance$ terraform fmt -check
00_provider.tf
05_resource_ec2.tf
terraform.tfvars
variables.tf
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/01_ec2_instance$ terraform validate
Success! The configuration is valid.

ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/01_ec2_instance$ terraform apply -auto-approve -parallelism=5 -refresh=true -var-file=../global_variable.auto.tfvars
data.aws_ami.ubuntu: Refreshing state...
data.aws_vpc.labvpc: Refreshing state...
aws_security_group.secgroup_server: Creating...
aws_security_group.secgroup_server: Creation complete after 1s [id=sg-0205c8bc062f4b183]
aws_instance.labserver[0]: Creating...
aws_instance.labserver[1]: Creating...
aws_instance.labserver[0]: Still creating... [10s elapsed]
aws_instance.labserver[1]: Still creating... [10s elapsed]
aws_instance.labserver[0]: Creation complete after 12s [id=i-071aca1891566ad7a]
aws_instance.labserver[1]: Creation complete after 12s [id=i-0266f98b2582bbe2b]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Outputs:

aws_instance.labserver_ip = {
  "i-0266f98b2582bbe2b" = "54.255.219.157"
  "i-071aca1891566ad7a" = "13.229.214.197"
}
aws_vpc.cidr = 172.31.0.0/16
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/01_ec2_instance$
```

Variable (Input & Output)

► Example (Share Variable): Rewrite Script (EC2)

```
Outputs:  
aws_instance_labserver_ip = {  
    "i-0da8541014abb05c2" = "54.255.128.121"  
    "i-0f0419f97be843d2d" = "54.254.238.230"  
}  
[ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/01_ec2_instance$ terraform output >> ../global_variable.auto.tfvars
```

```
[ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/02_route53$ more ../global_variable.auto.tfvars  
vpc_id="vpc-4483e921"  
region = "ap-southeast-1"  
availability_zone="ap-southeast-1a"  
zone_id="Z10002633BJEGNCUNP41E"  
aws_instance_labserver_ip = {  
    "i-0da8541014abb05c2" = "54.255.128.121"  
    "i-0f0419f97be843d2d" = "54.254.238.230"  
}
```

Variable (Input & Output)

➡ Example (Share Variable): Rewrite Script (Route 53)

```
Workshop_2.1_InputVariable variables.tf 01_route53.tf
```

Users > paparnlueangphoonlap > Work > Terraform > terraform_202003 > V

```
1 resource "aws_route53_record" "lab_route53" {
2   for_each = var.aws_instance_labserver_ip
3   zone_id = var.zone_id
4   name    = each.key
5   type    = "A"
6   ttl     = "30"
7   records = [each.value]
8 }
9 output "aws_route53_fqdn" {
10  value = {
11    for dns in aws_route53_record.lab_route53:
12      dns.name => dns.fqdn
13  }
14 }
```

```
[ubuntu@ip-172-31-28-165:~$ nslookup
[> i-0da8541014abb05c2.terraform.local
Server:          172.31.0.2
Address:         172.31.0.2#53

Non-authoritative answer:
Name:  i-0da8541014abb05c2.terraform.local
Address: 54.255.128.121
[> i-0f0419f97be843d2d.terraform.local
Server:          172.31.0.2
Address:         172.31.0.2#53

Non-authoritative answer:
Name:  i-0f0419f97be843d2d.terraform.local
Address: 54.254.238.230
> ]
```

```
instruction.txt ~.../Workshop_2.1_InputVariable variables.tf
```

Work > Terraform > terraform_202003 > Workshop_2.1_InputVariable >

```
1 ##### Define Variable to Operate #####
2 #                                     Define Variable to Operate
3 #####
4 variable "vpc_id" {
5   type = string
6   description = "VPC ID"
7 }
8
9 variable "region" {
10  type = string
11  description = "Defined region for operate"
12 }
13
14 variable "zone_id" {
15   type = string
16   description = "DNS Zone ID"
17 }
18
19 variable "availability_zone" {
20   type = string
21   description = "Availability zone for operate"
22 }
23
24 variable "aws_instance_labserver_ip" {
25   type = map
26   description = "Setup of machine ip and id"
27 }
```

Variable (Input & Output)

► Example (Share Variable): Rewrite Script

```
[ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/02_route53$ terraform apply -auto-approve -parallelism=5 -refresh=true -var-file=../global_variable.auto.tfvars ./aws_route53_record.lab_route53["i-0f0419f97be843d2d"]: Creating...
aws_route53_record.lab_route53["i-0da8541014abb05c2"]: Creating...
aws_route53_record.lab_route53["i-0da8541014abb05c2"]: Still creating... [10s elapsed]
aws_route53_record.lab_route53["i-0f0419f97be843d2d"]: Still creating... [10s elapsed]
aws_route53_record.lab_route53["i-0da8541014abb05c2"]: Still creating... [20s elapsed]
aws_route53_record.lab_route53["i-0f0419f97be843d2d"]: Still creating... [20s elapsed]
aws_route53_record.lab_route53["i-0da8541014abb05c2"]: Still creating... [30s elapsed]
aws_route53_record.lab_route53["i-0f0419f97be843d2d"]: Still creating... [30s elapsed]
aws_route53_record.lab_route53["i-0da8541014abb05c2"]: Still creating... [40s elapsed]
aws_route53_record.lab_route53["i-0f0419f97be843d2d"]: Still creating... [40s elapsed]
aws_route53_record.lab_route53["i-0f0419f97be843d2d"]: Still creating... [50s elapsed]
aws_route53_record.lab_route53["i-0da8541014abb05c2"]: Still creating... [50s elapsed]
aws_route53_record.lab_route53["i-0f0419f97be843d2d"]: Still creating... [1m0s elapsed]
aws_route53_record.lab_route53["i-0da8541014abb05c2"]: Still creating... [1m0s elapsed]
aws_route53_record.lab_route53["i-0f0419f97be843d2d"]: Creation complete after 1m4s [id=Z10002633BJEYNCUNP41E_i-0f0419f97be843d2d_A]
aws_route53_record.lab_route53["i-0da8541014abb05c2"]: Creation complete after 1m5s [id=Z10002633BJEYNCUNP41E_i-0da8541014abb05c2_A]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

aws_route53_fqdn = {
  "i-0da8541014abb05c2" = "i-0da8541014abb05c2.terraform.local"
  "i-0f0419f97be843d2d" = "i-0f0419f97be843d2d.terraform.local"
}

ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/02_route53$ █
```

Variable (Input & Output)

► Example (Share Variable): Destroy (Route53)

```
[ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/02_route53$ cd ~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/02_route53$ terraform destroy -auto-approve -var-file=../global_variable.auto.tfvars ./
aws_route53_record.lab_route53["i-0f0419f97be843d2d"] Refreshing state... [id=Z10002633BJEGNCUNP41E_i-0f0419f97be843d2d_A]
aws_route53_record.lab_route53["i-0da8541014abb05c2"] Refreshing state... [id=Z10002633BJEGNCUNP41E_i-0da8541014abb05c2_A]
aws_route53_record.lab_route53["i-0f0419f97be843d2d"] Destroying... [id=Z10002633BJEGNCUNP41E_i-0f0419f97be843d2d_A]
aws_route53_record.lab_route53["i-0da8541014abb05c2"] Destroying... [id=Z10002633BJEGNCUNP41E_i-0da8541014abb05c2_A]
aws_route53_record.lab_route53["i-0da8541014abb05c2"] Still destroying... [id=Z10002633BJEGNCUNP41E_i-0da8541014abb05c2_A, 10s elapsed]
aws_route53_record.lab_route53["i-0f0419f97be843d2d"] Still destroying... [id=Z10002633BJEGNCUNP41E_i-0f0419f97be843d2d_A, 10s elapsed]
aws_route53_record.lab_route53["i-0da8541014abb05c2"] Still destroying... [id=Z10002633BJEGNCUNP41E_i-0da8541014abb05c2_A, 20s elapsed]
aws_route53_record.lab_route53["i-0f0419f97be843d2d"] Still destroying... [id=Z10002633BJEGNCUNP41E_i-0f0419f97be843d2d_A, 20s elapsed]
aws_route53_record.lab_route53["i-0f0419f97be843d2d"] Still destroying... [id=Z10002633BJEGNCUNP41E_i-0f0419f97be843d2d_A, 30s elapsed]
aws_route53_record.lab_route53["i-0da8541014abb05c2"] Still destroying... [id=Z10002633BJEGNCUNP41E_i-0da8541014abb05c2_A, 30s elapsed]
aws_route53_record.lab_route53["i-0f0419f97be843d2d"] Still destroying... [id=Z10002633BJEGNCUNP41E_i-0f0419f97be843d2d_A, 40s elapsed]
aws_route53_record.lab_route53["i-0da8541014abb05c2"] Still destroying... [id=Z10002633BJEGNCUNP41E_i-0da8541014abb05c2_A, 40s elapsed]
aws_route53_record.lab_route53["i-0f0419f97be843d2d"] Still destroying... [id=Z10002633BJEGNCUNP41E_i-0f0419f97be843d2d_A, 50s elapsed]
aws_route53_record.lab_route53["i-0da8541014abb05c2"] Still destroying... [id=Z10002633BJEGNCUNP41E_i-0da8541014abb05c2_A, 50s elapsed]
aws_route53_record.lab_route53["i-0f0419f97be843d2d"] Still destroying... [id=Z10002633BJEGNCUNP41E_i-0da8541014abb05c2_A, 1m0s elapsed]
aws_route53_record.lab_route53["i-0f0419f97be843d2d"] Still destroying... [id=Z10002633BJEGNCUNP41E_i-0f0419f97be843d2d_A, 1m0s elapsed]
aws_route53_record.lab_route53["i-0da8541014abb05c2"] Destruction complete after 1m2s
aws_route53_record.lab_route53["i-0f0419f97be843d2d"] Destruction complete after 1m2s

Destroy complete! Resources: 2 destroyed.
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/02_route53$
```

Variable (Input & Output)

► Example (Share Variable): Destroy (EC2)

```
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/02_route53$ cd ~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/01_ec2_instance
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/01_ec2_instance$      terraform destroy -auto-approve -var-file=../global_variable.auto.tfvars ./
data.aws_vpc.labvpc: Refreshing state...
data.aws_ami.ubuntu: Refreshing state...
aws_security_group.secgroup_server: Refreshing state... [id=sg-0d8829d2c5eaaf29e]
aws_instance.labserver[1]: Refreshing state... [id=i-0f0419f97be843d2d]
aws_instance.labserver[0]: Refreshing state... [id=i-0da8541014abb05c2]
aws_instance.labserver[0]: Destroying... [id=i-0da8541014abb05c2]
aws_instance.labserver[1]: Destroying... [id=i-0f0419f97be843d2d]
aws_instance.labserver[1]: Still destroying... [id=i-0f0419f97be843d2d, 10s elapsed]
aws_instance.labserver[0]: Still destroying... [id=i-0da8541014abb05c2, 10s elapsed]
aws_instance.labserver[1]: Still destroying... [id=i-0f0419f97be843d2d, 20s elapsed]
aws_instance.labserver[0]: Still destroying... [id=i-0da8541014abb05c2, 20s elapsed]
aws_instance.labserver[0]: Still destroying... [id=i-0da8541014abb05c2, 30s elapsed]
aws_instance.labserver[1]: Still destroying... [id=i-0f0419f97be843d2d, 30s elapsed]
aws_instance.labserver[1]: Still destroying... [id=i-0f0419f97be843d2d, 40s elapsed]
aws_instance.labserver[0]: Still destroying... [id=i-0da8541014abb05c2, 40s elapsed]
aws_instance.labserver[1]: Still destroying... [id=i-0f0419f97be843d2d, 50s elapsed]
aws_instance.labserver[0]: Still destroying... [id=i-0da8541014abb05c2, 50s elapsed]
aws_instance.labserver[1]: Destruction complete after 1m0s
aws_instance.labserver[0]: Still destroying... [id=i-0da8541014abb05c2, 1m0s elapsed]
aws_instance.labserver[0]: Still destroying... [id=i-0da8541014abb05c2, 1m10s elapsed]
aws_instance.labserver[0]: Destruction complete after 1m20s
aws_security_group.secgroup_server: Destroying... [id=sg-0d8829d2c5eaaf29e]
aws_security_group.secgroup_server: Destruction complete after 0s

Warning: Value for undeclared variable

The root module does not declare a variable named "aws_instance_labserver_ip"
but a value was found in file "../global_variable.auto.tfvars". To use this
value, add a "variable" block to the configuration.

Using a variables file to set an undeclared variable is deprecated and will
become an error in a future release. If you wish to provide certain "global"
settings to all configurations in your organization, use TF_VAR_...
environment variables to set these instead.

Destroy complete! Resources: 3 destroyed.
ubuntu@ip-172-31-17-19:~/terraform_202003/Workshop_2.1_InputVariable/04_share_variable/01_ec2_instance$ █
```

Workshop: Variable (Input & Output)



Workshop: Variable (Input & Output)

```
ubuntu@ip-172-31-19-113:~/terraform_202003/Workshop_1.4_Provisioner$ terraform apply -auto-approve -parallelism=5 -refresh=true ./
data.aws_ami.ubuntu: Refreshing state...
[aws_ebs_volume.labebs: Creating...
[aws_instance.lab: Creating...
aws_ebs_volume.labebs: Still creating... [10s elapsed]
aws_instance.lab: Still creating... [10s elapsed]
aws_ebs_volume.labebs: Creation complete after 10s [id=vol-0e5e772a20d140cea]
aws_instance.lab: Creation complete after 12s [id=i-0a16c706816ceadad]
aws_volume_attachment.labebsattach: Creating...
aws_volume_attachment.labebsattach: Still creating... [10s elapsed]
aws_volume_attachment.labebsattach: Still creating... [20s elapsed]
aws_volume_attachment.labebsattach: Provisioning with 'remote-exec'...
aws_volume_attachment.labebsattach (remote-exec): Connecting to remote host via SSH...
aws_volume_attachment.labebsattach (remote-exec): Host: 13.251.124.175
aws_volume_attachment.labebsattach (remote-exec): User: ubuntu
aws_volume_attachment.labebsattach (remote-exec): Password: false
aws_volume_attachment.labebsattach (remote-exec): Private key: true
aws_volume_attachment.labebsattach (remote-exec): Certificate: false
aws_volume_attachment.labebsattach (remote-exec): SSH Agent: false
aws_volume_attachment.labebsattach (remote-exec): Checking Host Key: false
aws_volume_attachment.labebsattach (remote-exec): Connected!
aws_volume_attachment.labebsattach (remote-exec): meta-data=/dev/nvme1n1      isize=512    agcount=4, agsize=1310720 blks
aws_volume_attachment.labebsattach (remote-exec):          =      sectsz=512   attr=2, projid32bit=1
aws_volume_attachment.labebsattach (remote-exec):          =      crc=1     finobt=1, sparse=0, rmapbt=0, reflink=0
aws_volume_attachment.labebsattach (remote-exec): data      =      bsize=4096  blocks=5242880, imaxpct=25
aws_volume_attachment.labebsattach (remote-exec):          =      sunit=0    swidth=0 blks
aws_volume_attachment.labebsattach (remote-exec): naming    =version 2   bsize=4096  ascii-ci=0 ftype=1
aws_volume_attachment.labebsattach (remote-exec): log       =internal log  bsize=4096  blocks=2560, version=2
aws_volume_attachment.labebsattach (remote-exec):          =      sectsz=512  sunit=0 blks, lazy-count=1
aws_volume_attachment.labebsattach (remote-exec): realtime  =none      extsz=4096  blocks=0, rtextents=0
aws_volume_attachment.labebsattach: Creation complete after 22s [id=vai-1096346766]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Outputs:

aws_ebs_volume_labebs_id = vol-0e5e772a20d140cea
aws_instance_lab_id = i-0a16c706816ceadad
aws_instance_lab_keyname = keypair
aws_instance_lab_public_dns = ec2-13-251-124-175.ap-southeast-1.compute.amazonaws.com
aws_instance_lab_public_ip = 13.251.124.175
ubuntu@ip-172-31-19-113:~/terraform_202003/Workshop_1.4_Provisioner$
```



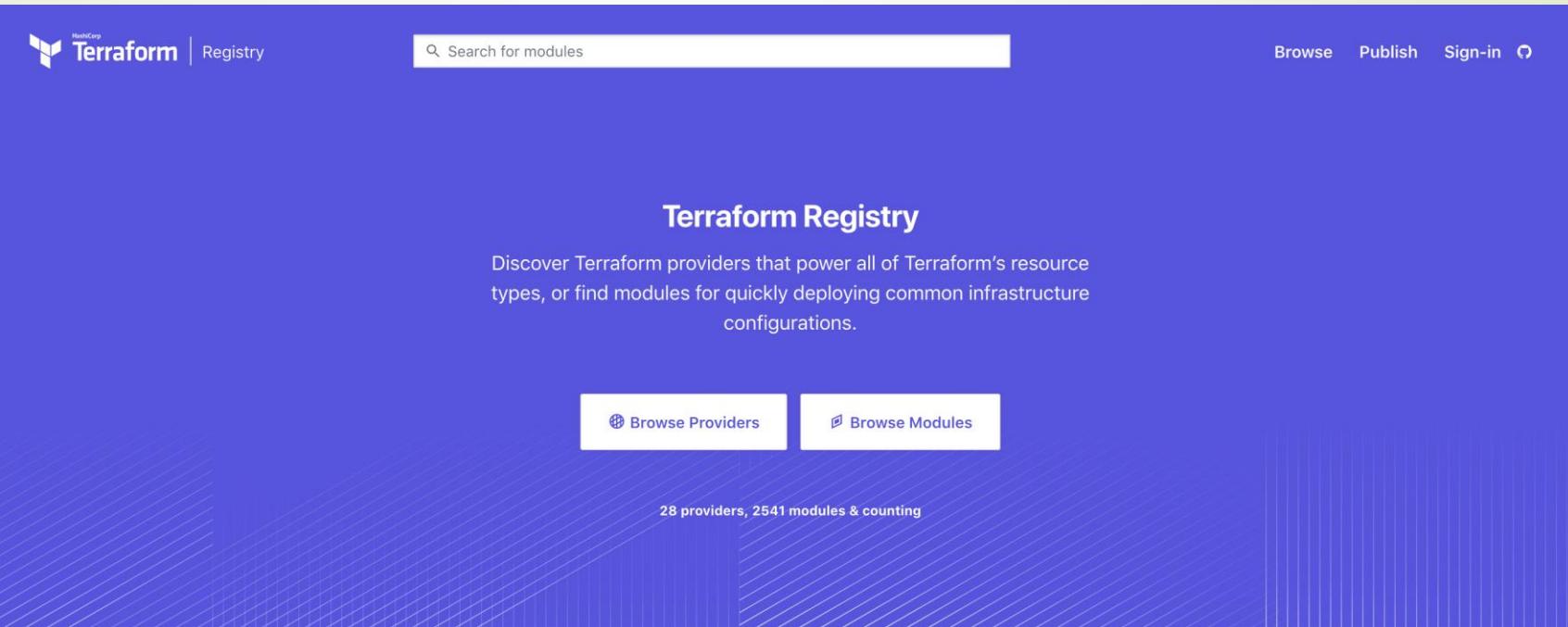
Q&A



Registry and Module

- ▶ A lot of activity on terraform need to create multiple resource that can contain with single “module”
- ▶ Some cloud/platform have a huge component need to play attention / create. But the product owner also provide “module” for speed-up provisioning
- ▶ Terraform had been provided platform for all terraforms community to share them module and provider. And provide official repository
- ▶ Terraform will use GitHub as backend for keep all module
- ▶ We can check base on “provider” or “module”

Registry and Module



Providers NEW

Providers are **plugins that implement resource types**. Find providers for the cloud platforms and services you use, add them to your configuration, then use their resources to provision infrastructure.

[Browse All](#)

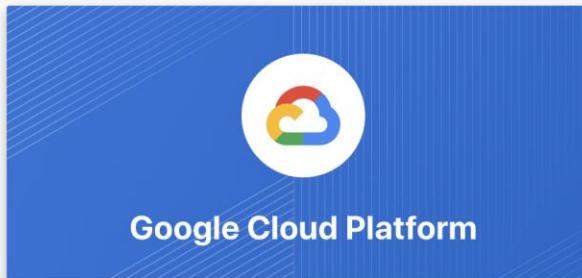
[Learn More >](#)

Registry and Module

Official cloud providers

[See all >](#)

These providers are owned, tested and maintained by HashiCorp. This group includes hyper-scale cloud providers that offer a range of services including IaaS, SaaS, and PaaS.



 [Learn about providers](#)

 [Write a provider](#)

 [Publish a provider](#)

Registry and Module

The screenshot shows the Terraform Registry interface. At the top, there's a navigation bar with the HashiCorp logo, the word "Terraform", a search bar containing "Search for modules", and links for "Browse", "Publish", and "Sign-in". Below the navigation, a breadcrumb trail shows "Providers / hashicorp / azurerm / Version 1.42.0" and a "Latest Version" button. The main content area features the "azurerm" provider logo, its name, an "Official" badge, and the publisher "HashiCorp". It displays the version 1.42.0 was published 3 days ago, has 63765 installs, and provides a source link. On the right side, there are "HELPFUL LINKS" for "Using Providers" and "Learn Terraform", and a "Report an issue" button.

Registry and Module

The screenshot shows the HashiCorp Terraform Registry interface. At the top, there's a navigation bar with the Terraform logo, a search bar labeled "Search for modules", and links for "Browse", "Publish", "Sign-in", and a user icon. Below the navigation, a breadcrumb trail shows "Providers / hashicorp / azurerm / Version 1.42.0" and a "Latest Version" button. The main content area has tabs for "Overview", "Documentation" (which is selected), and "USE PROVIDER". The "Documentation" tab contains sections for "ON THIS PAGE" (Authenticating to Azure, Example Usage, Features and Bug Requests, Argument Reference) and a "Give Feedback" link. The "Azure Provider" section itself discusses its use for configuring Microsoft Azure infrastructure via the Azure Resource Manager API. It mentions Data Sources and Resources supported by the provider. A sidebar on the left lists various Azure services like Guides, API Management, Analysis Services, App Configuration, App Service (Web Apps), Application Insights, Authorization, Automation, Azure Active Directory, Base, Batch, Beta, Bot, CDN, Cognitive Services, Compute, Container, and CosmosDB (DocumentDB). The "Authenticating to Azure" section details four methods: using the Azure CLI, Managed Service Identity, a Service Principal and Client Certificate, or a Service Principal and Client Secret.

Registry and Module

The screenshot shows the HashiCorp Terraform Registry's "Modules" page. At the top, there is a purple square icon with a white square inside. Below it, the word "Modules" is centered in a bold, dark font. A descriptive paragraph follows: "Modules are small, reusable Terraform configurations that let you manage a group of related resources as if they were a single resource." Two buttons are present: "Browse All" and "Learn More >". Below these are four module cards arranged in a 2x2 grid. Each card includes a provider logo (Cloud SQL, AzureRM, AWS, or Azure), the module name, a brief description, the last updated time, the number of downloads, and the provider type (provider).

Module cards:

- gruntwork-io / sql**
Terraform modules for deploying Cloud SQL (e.g. MySQL, PostgreSQL) in GCP
🕒 7 months ago, 346 downloads, Cloud provider
- hashicorp / nomad**
A Terraform Module for how to run Nomad on AzureRM using Terraform and Packer
🕒 a year ago, 25 downloads, Azure provider
- hashicorp / vault**
A Terraform Module for how to run Vault on AWS using Terraform and Packer
🕒 a day ago, 13596 downloads, AWS provider
- hashicorp / vault**
A Terraform Module for how to run Vault on AzureRM using Terraform and Packer
🕒 a year ago, 108 downloads, Azure provider

At the bottom of the page are three links: "Learn about modules", "Write a module", and "Publish a module".

Registry and Module

The screenshot shows the HashiCorp Terraform Registry interface. At the top, there's a navigation bar with the Terraform logo, a search bar labeled "Search for modules", and links for "Browse", "Publish", and "Sign-in". Below the navigation, there are tabs for "Providers" and "Modules", with "Modules" being the active tab. A sub-header "Modules" is followed by a sub-instruction "Modules are self-contained packages of Terraform configurations that are managed as a group.". On the right side of the main content area, there's a "FILTER BY" dropdown set to "azurerm" and a checked checkbox for "Verified". The main content area displays a grid of nine Terraform modules:

- loadbalancer** (azurerm) - Terraform Azure RM Module for Load Balancer. Version 1.2.1 · By Azure.
- network** (azurerm) - Terraform Azure RM Module for Network. Version 2.0.0 · By Azure.
- computegroup** (azurerm) - Terraform Azure RM Compute Group Module. Version 2.1.0 · By Azure.
- compute** (azurerm) - Terraform Azure RM Compute Module. Version 2.0.0 · By Azure.
- database** (azurerm) - Terraform Azure RM Module for Database. Version 1.1.0 · By Azure.
- consul** (azurerm) - A Terraform Module for how to run Consul on AzureRM using Terraform... Version 0.0.5 · By hashicorp.
- vault** (azurerm) - A Terraform Module for how to run Vault on AzureRM using Terraform and... Version 0.0.2 · By hashicorp.
- kubernetes** (azurerm) - Install a Kubernetes cluster the CoreOS Tectonic Way: HA, self-hosted, RBAC,... Version 1.8.9-tectonic.1 · By coreos.
- nomad** (azurerm) - A Terraform Module for how to run Nomad on AzureRM using Terraform... Version 0.0.1 · By hashicorp.

Registry and Module

The screenshot shows the HashiCorp Terraform Registry interface. At the top, there is a purple header bar with the Terraform logo, a search bar labeled "Search for modules", and navigation links for "Browse", "Publish", and "Sign-in". Below the header, a module card is displayed for the "compute" module by AZURERM. The card includes the module icon (a blue triangle), the name "compute" with a blue checkmark badge, the provider "AZURERM", a version dropdown set to "Version 2.0.0", and a brief description: "Terraform Azure RM Compute Module". Below the card, detailed information is provided: "Published January 13, 2020 by Azure", "Module managed by dtzar", "Total provisions: 21,838", and a link to the source code on GitHub: "github.com/Azure/terraform-azurerm-compute (report an issue)". At the bottom of the card, there are links for "Readme", "Inputs (31)", "Outputs (9)", "Dependencies (0)", and "Resources (12)". A "Provision Instructions" section on the right contains a code snippet for Terraform configuration:

```
module "compute" {  
  source = "Azure/compute/azurerm"  
  version = "2.0.0"  
  # insert the 2 required variables here  
}
```

Registry and Module

Readme Inputs (31) Outputs (9) Dependencies (0) Resources (12)

Resources

This is the list of resources that the module *may* create. The module can create zero or more of each of these resources depending on the `count` value. The count value is determined at runtime. The goal of this page is to present the types of resources that may be created.

This list contains all the resources this plus any submodules may create. When using this module, it may create less resources if you use a submodule.

This module defines **12** resources .

- `azurerm_availability_set.vm`
- `azurerm_network_interface.vm`
- `azurerm_network_security_group.vm`
- `azurerm_network_security_rule.vm`
- `azurerm_public_ip.vm`
- `azurerm_resource_group.vm`
- `azurerm_storage_account.vm-sa`
- `azurerm_virtual_machine.vm-linux`
- `azurerm_virtual_machine.vm-linux-with-disk`
- `azurerm_virtual_machine.vm-windows`
- `azurerm_virtual_machine.vm-windows-with-disk`
- `random_id.vm-sa`

Registry and Module

```
HCL
1 module "linuxservers" {
2   source      = "Azure/compute/azurerm"
3   location    = "West US 2"
4   vm_os_simple = "UbuntuServer"
5   public_ip_dns = ["linsimplevmips"] // change to a unique name per datacenter region
6   vnet_subnet_id = "${module.network.vnet_subnets[0]}"
7 }
8
9 module "windowsservers" {
10   source      = "Azure/compute/azurerm"
11   location    = "West US 2"
12   vm_hostname = "mywinvm" // line can be removed if only one VM module per resource group
13   admin_password = "ComplxP@ssw0rd!"
14   vm_os_simple = "WindowsServer"
15   is_windows_image = "true"
16   public_ip_dns = ["winsimplevmips"] // change to a unique name per datacenter region
17   vnet_subnet_id = "${module.network.vnet_subnets[0]}"
18 }
19
20 module "network" {
21   source      = "Azure/network/azurerm"
22   version     = "~> 1.1.1"
23   location    = "West US 2"
24   allow_rdp_traffic = "true"
25   allow_ssh_traffic = "true"
26   resource_group_name = "terraform-compute"
27 }
28
29 output "linux_vm_public_name"{
30   value = "${module.linuxservers.public_ip_dns_name}"
31 }
32
33 output "windows_vm_public_name"{
34   value = "${module.windowsservers.public_ip_dns_name}"
35 }
```

Registry and Module

Readme Inputs (31) Outputs (9) Dependencies (0) Resources (12)

Required Inputs

These variables must be set in the `module` block when using this module.

location `string`
Description: The location/region where the virtual network is created. Changing this forces a new resource to be created.

vnet_subnet_id `string`
Description: The subnet id of the virtual network where the virtual machines will reside.

Optional Inputs

These variables have default values and don't have to be set to use this module. You may set these variables to override their default values.

admin_password `string`
Description: The admin password to be used on the VMSS that will be deployed. The password must meet the complexity requirements of Azure
Default: `!!`

admin_username `string`
Description: The admin username of the VM that will be deployed

Readme Inputs (31) Outputs (9) Dependencies (0) Resources (12)

availability_set_id

Description: id of the availability set where the vms are provisioned.

network_interface_ids

Description: ids of the vm nics provisioned.

network_interface_private_ip

Description: private ip addresses of the vm nics

network_security_group_id

Description: id of the security group provisioned

network_security_group_name

Description: name of the security group provisioned

public_ip_address

Description: The actual ip address allocated for the resource.

public_ip_dns_name

Description: fqdn to connect to the first vm provisioned.

raform

Registry and Module

► **Example:** Create VM on Azure

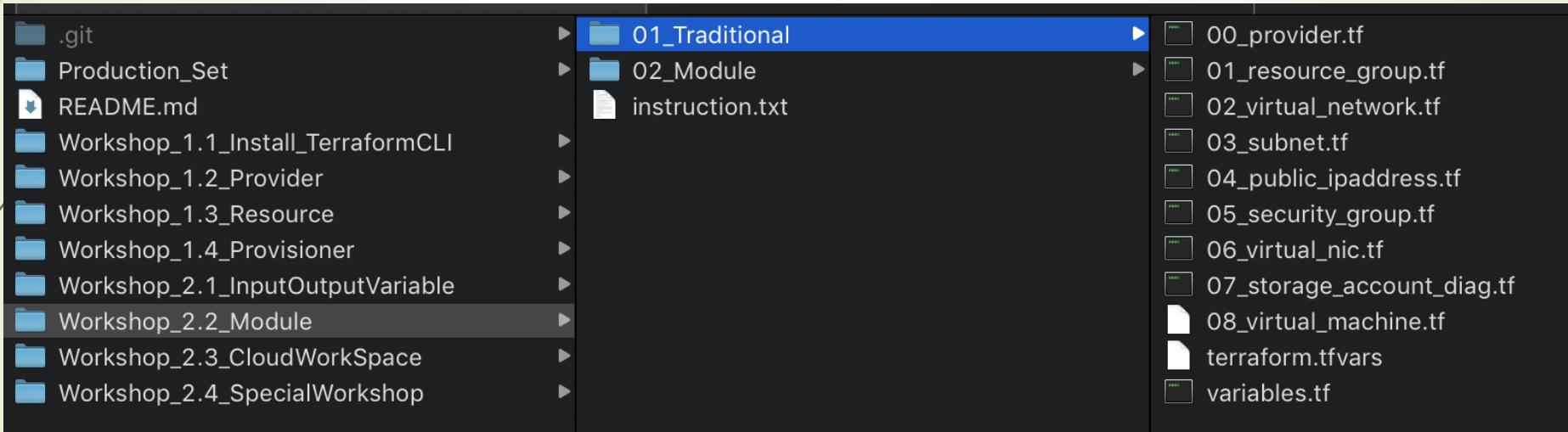
► **Traditional Way(List of Resource):**

- azurerm_resource_group
- azurerm_virtual_network
- azurerm_subnet
- azurerm_public_ip
- azurerm_network_security_group
- azurerm_network_interface
- azurerm_storage_account
- azurerm_virtual_machine

Registry and Module

► **Example:** Create VM on Azure

► **Traditional Way(List of Resource):**



Registry and Module

► **Example:** Create VM on Azure

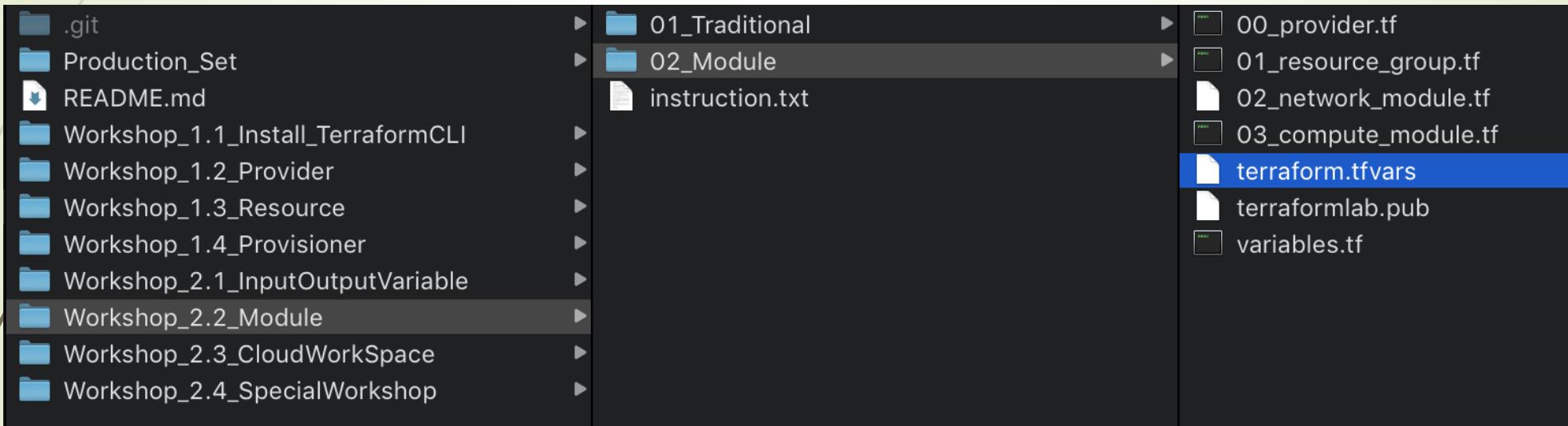
► **Module Way(List of Module):**

- azurerm_resource_group (resource)
- network (module)
- linuxservers (module)

Registry and Module

► **Example:** Create VM on Azure

► **Module Way(List of Module):**



Registry and Module

► Example: Module Way(List of Module):

```
[ubuntu@ip-172-31-27-251:~/terraform_202003/Workshop_2.2_Module/01_Module$ terraform init
Initializing modules...
Downloading Azure/compute/azurerm 2.0.0 for linuxservers...
- linuxservers in .terraform/modules/linuxservers/Azure-terraform-azurerm-compute-9952198
- linuxservers.os in .terraform/modules/linuxservers/Azure-terraform-azurerm-compute-9952198/os
Downloading Azure/network/azurerm 1.1.1 for network...
- network in .terraform/modules/network/Azure-terraform-azurerm-network-d44b1dc

Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "azurerm" (hashicorp/azurerm) 1.38.0...
- Downloading plugin for provider "random" (hashicorp/random) 2.2.1...

Warning: Skipping backend initialization pending configuration upgrade

The root module configuration contains errors that may be fixed by running the
configuration upgrade tool, so Terraform is skipping backend initialization.
See below for more information.

Terraform has initialized, but configuration upgrades may be needed.

Terraform found syntax errors in the configuration that prevented full
initialization. If you've recently upgraded to Terraform v0.12, this may be
because your configuration uses syntax constructs that are no longer valid,
and so must be updated before full initialization is possible.

Terraform has installed the required providers to support the configuration
upgrade process. To begin upgrading your configuration, run the following:
  terraform 0.12upgrade

To see the full set of errors that led to this message, run:
  terraform validate
```

Registry and Module

► Example: Module Way(List of Module):

```
ubuntu@ip-172-31-27-251:~/terraform_202003/Workshop_2.2_Module/01_Module$ ls -R .terraform/
.terraform/:
modules plugins

.terraform/modules:
linuxservers modules.json network

.terraform/modules/linuxservers:
Azure-terraform-azurerm-compute-9952198

.terraform/modules/linuxservers/Azure-terraform-azurerm-compute-9952198:
Dockerfile Gemfile Gopkg.lock Gopkg.toml LICENSE README.md Rakefile bug_report.md main.tf os outputs.tf test variables.tf

.terraform/modules/linuxservers/Azure-terraform-azurerm-compute-9952198/os:
outputs.tf variables.tf

.terraform/modules/linuxservers/Azure-terraform-azurerm-compute-9952198/test:
fixture terraform_ssh_example_test.go

.terraform/modules/linuxservers/Azure-terraform-azurerm-compute-9952198/test/fixture:
main.tf outputs.tf terraform.tfvars variables.tf

.terraform/modules/network:
Azure-terraform-azurerm-network-d44b1dc

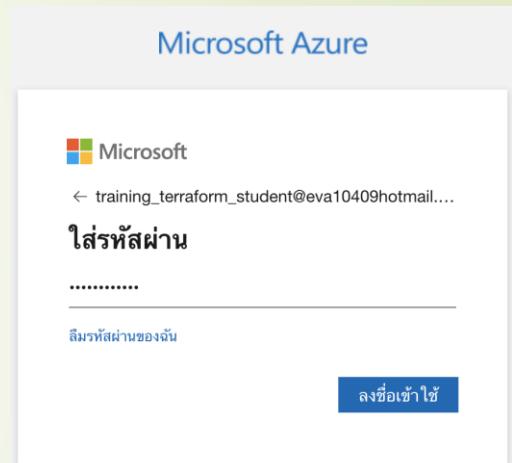
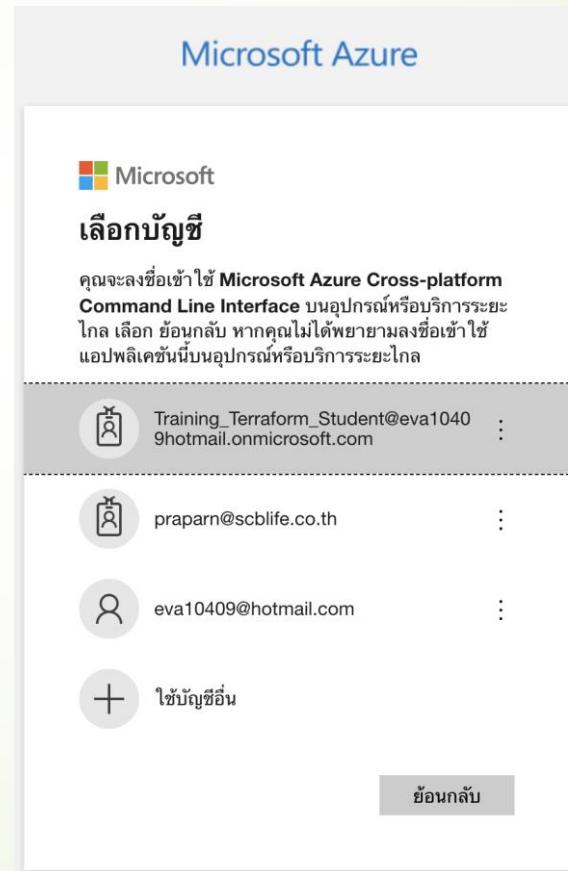
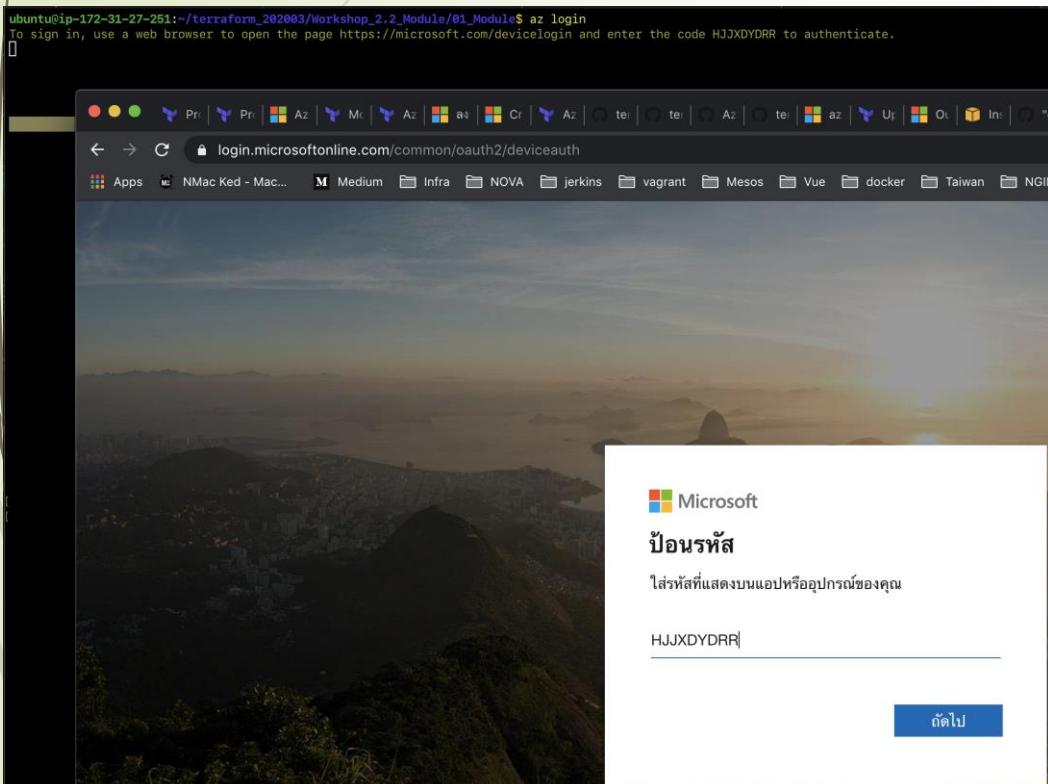
.terraform/modules/network/Azure-terraform-azurerm-network-d44b1dc:
README.md main.tf outputs.tf variables.tf

.terraform/plugins:
linux_amd64

.terraform/plugins/linux_amd64:
lock.json terraform-provider-azurerm_v1.38.0_x4 terraform-provider-random_v2.2.1_x4
ubuntu@ip-172-31-27-251:~/terraform_202003/Workshop_2.2_Module/01_Module$
```

Registry and Module

► Example: Module Way(List of Module):



Registry and Module

► Example: Module Way(List of Module):

```
[ubuntu@ip-172-31-27-251:~/terraform_202003/Workshop_2.2_Module/01_Module]$ terraform fmt -check
00_provider.tf
01_resource_group.tf
02_network_module.tf
03_compute_module.tf
terraform.tfvars
variables.tf
[ubuntu@ip-172-31-27-251:~/terraform_202003/Workshop_2.2_Module/01_Module]$ terraform validate
terraform: command not found
[ubuntu@ip-172-31-27-251:~/terraform_202003/Workshop_2.2_Module/01_Module]$ terraform validate

Warning: Interpolation-only expressions are deprecated

  on .terraform/modules/network/Azure-terraform-azurerm-network-d44b1dc/main.tf line 3, in resource "azurerm_resource_group" "network":
    3:   name      = "${var.resource_group_name}"

Terraform 0.11 and earlier required all non-constant expressions to be
provided via interpolation syntax, but this pattern is now deprecated. To
silence this warning, remove the "${" sequence from the start and the "}"
sequence from the end of this expression, leaving just the inner expression.

Template interpolation syntax is still used to construct strings from
expressions when the template includes multiple interpolation sequences or a
mixture of literal strings and interpolations. This deprecation applies only
to templates that consist entirely of a single interpolation sequence.

(and 22 more similar warnings elsewhere)

Warning: Quoted type constraints are deprecated

  on .terraform/modules/network/Azure-terraform-azurerm-network-d44b1dc/variables.tf line 38, in variable "tags":
    38:   type = "map"

Terraform 0.11 and earlier required type constraints to be given in quotes,
but that form is now deprecated and will be removed in a future version of
Terraform. To silence this warning, remove the quotes around "map" and write
map(string) instead to explicitly indicate that the map elements are strings.

Error: Unsupported block type

  on variables.tf line 144:
144: variable "custom_data" {

Blocks of type "variable" are not expected here. Did you mean "variable"?

[ubuntu@ip-172-31-27-251:~/terraform_202003/Workshop_2.2_Module/01_Module$ ]
```

Registry and Module

► Example: Module Way(List of Module):

```
ubuntu@ip-172-31-20-216:~/terraform_202003/Workshop_2.2_Module/02_Module$ terraform apply -auto-approve -parallelism=5
module.linuxservers.random_id.vm-sa: Creating...
module.linuxservers.random_id.vm-sa: Creation complete after 0s [id=KFyqAPM-]
azurerm_resource_group.labrscgrp: Creating...
azurerm_resource_group.labrscgrp: Creation complete after 0s [id=/subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22]
module.network.azurerm_resource_group.network: Creating...
module.linuxservers.azurerm_resource_group.vm: Creating...
module.network.azurerm_resource_group.network: Creation complete after 1s [id=/subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22]
module.network.azurerm_virtual_network.vnet: Creating...
module.network.azurerm_network_security_group.security_group: Creating...
module.linuxservers.azurerm_resource_group.vm: Creation complete after 1s [id=/subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22]
module.linuxservers.azurerm_availability_set.vm: Creating...
module.linuxservers.azurerm_network_security_group.vm: Creating...
module.linuxservers.azurerm_storage_account.vm-sa[0]: Creating...
module.linuxservers.azurerm_availability_set.vm: Creation complete after 0s [id=/subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Compute/availabilitySets/labserver-student-22-avset]
module.linuxservers.azurerm_public_ip.vm[0]: Creating...
module.network.azurerm_network_security_group.security_group: Creation complete after 0s [id=/subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Network/networkSecurityGroups/acctsecgrp]
module.network.azurerm_network_security_rule.security_rule_ssh[0]: Creating...
module.linuxservers.azurerm_public_ip.vm[0]: Creation complete after 5s [id=/subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Network/publicIPAddresses/labserver-student-22-0-publicIP]
module.network.azurerm_virtual_network.vnet: Still creating... [10s elapsed]
module.linuxservers.azurerm_network_security_group.vm: Still creating... [10s elapsed]
module.linuxservers.azurerm_storage_account.vm-sa[0]: Still creating... [10s elapsed]
module.network.azurerm_network_security_rule.security_rule_ssh[0]: Still creating... [10s elapsed]
module.linuxservers.azurerm_network_security_group.vm: Creation complete after 10s [id=/subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Network/networkSecurityGroups/labserver-student-22-22-nsg]
module.linuxservers.azurerm_network_security_rule.vm: Creating...
module.network.azurerm_virtual_network.vnet: Creation complete after 10s [id=/subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Network/virtualNetworks/acctvnet]
module.network.azurerm_subnet.subnet[0]: Creating...
module.network.azurerm_network_security_rule.security_rule_ssh[0]: Creation complete after 13s [id=/subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Network/networkSecurityGroups/acctsecgrp/securityRules/ssh]
module.linuxservers.azurerm_storage_account.vm-sa[0]: Still creating... [20s elapsed]
module.linuxservers.azurerm_storage_account.vm-sa[0]: Creation complete after 20s [id=/subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Storage/storageAccounts/bootdiag285caa00f33e]
module.linuxservers.azurerm_network_security_rule.vm: Still creating... [10s elapsed]
module.network.azurerm_subnet.subnet[0]: Still creating... [10s elapsed]
module.linuxservers.azurerm_network_security_rule.vm: Creation complete after 11s [id=/subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Network/networkSecurityGroups/labserver-student-22-22-nsg/securityRules/allow_remote_22_in_all]
module.network.azurerm_subnet.subnet[0]: Creation complete after 14s [id=/subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Network/virtualNetworks/acctvnet/subnets/subnet1]
module.linuxservers.azurerm_network_interface.vm[0]: Creating...
module.linuxservers.azurerm_network_interface.vm[0]: Creation complete after 0s [id=/subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Network/networkInterfaces/nic-labserver-student-22-0]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-datadisk[0]: Creating...
```

Registry and Module

► Example: Module Way(List of Module):

```
azurerm_resource_group.labrscgrp: Modifications complete after 0s [id=/subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22]
module.linuxservers.azurerm_network_interface.vm[0]: Modifications complete after 1s [id=/subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Network/networkInterfaces/nic-labserver-student-22-0]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-disk[0]: Creating...
module.linuxservers.azurerm_virtual_machine.vm-linux-with-disk[0]: Still creating... [10s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-disk[0]: Still creating... [20s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-disk[0]: Still creating... [30s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-disk[0]: Still creating... [40s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-disk[0]: Still creating... [50s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-disk[0]: Still creating... [1m0s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-disk[0]: Still creating... [1m10s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-disk[0]: Still creating... [1m20s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-disk[0]: Still creating... [1m30s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-disk[0]: Still creating... [1m40s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-disk[0]: Still creating... [1m50s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-disk[0]: Still creating... [2m0s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-disk[0]: Still creating... [2m10s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-disk[0]: Still creating... [2m20s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-disk[0]: Still creating... [2m30s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-disk[0]: Creation complete after 2m30s [id=/subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Compute/virtualMachines/labserver-student-220]
```

Registry and Module

```
[praparns-MacBook-Pro:~] praparn$ ssh -i terraformlab ubuntu@labserver202003-thailand-student-8f34c833ab88.southeastasia.cloudapp.azure.com
The authenticity of host 'labserver202003-thailand-student-8f34c833ab88.southeastasia.cloudapp.azure.com (207.46.228.144)' can't be established.
ECDSA key fingerprint is SHA256:eCm6e0Vkn9+zNcZaSzwyIJB8tyhhn5M+U/L3/6xe1c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'labserver202003-thailand-student-8f34c833ab88.southeastasia.cloudapp.azure.com,207.46.228.144' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.0.0-1028-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information disabled due to load higher than 1.0

21 packages can be updated.
14 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

[ubuntu@labserver-student-30:~]$ top
```

```
[ubuntu@labserver-student-30:~]$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:9572f7cdcee8591948c2963463447a53466950b3fc15a247fcad1917ca215a2f
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

[ubuntu@labserver-student-30:~]\$ █

Registry and Module

```
[ubuntu@ip-172-31-27-251:~/terraform_202003/Workshop_2_2_Module/01_Module$ terraform destroy -auto-approve
module.linuxservers.random_id.vm-sa: Refreshing state... [id=PrdWVkj]
azurerm_resource_group.labresgrp: Refreshing state... [id=subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22]
module.network.azurerm_resource_group.network: Refreshing state... [id=subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22]
module.linuxservers.azurerm_resource_group.vm: Refreshing state... [id=subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22]
module.network.azurerm_network_security_group.security_group: Refreshing state... [id=subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Network/networkSecurityGroups/acctsecgrp]
module.network.azurerm_virtual_network.vnet: Refreshing state... [id=subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Network/virtualNetworks/acctvnet]
module.linuxservers.azurerm_storage_account.vm-sa[0]: Refreshing state... [id=subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Storage/storageAccounts/bootdisk3eb756495923]
module.linuxservers.azurerm_availability_set.vm: Refreshing state... [id=subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Compute/availabilitySets/labserver-student-22-avset]
module.linuxservers.azurerm_network_security_group.vm: Refreshing state... [id=subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Network/networkSecurityGroups/labserver-student-22-22-nsg]
module.linuxservers.azurerm_public_ip.vm[0]: Refreshing state... [id=subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Network/publicIPAddresses/labserver-student-22-0-publicIP]
module.linuxservers.azurerm_network_security_rule.vm: Refreshing state... [id=subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Network/networkSecurityGroups/labserver-student-22-22-nsg/securityRules/allow_remote_22_in_all]
module.network.azurerm_network_security_rule_ssh[0]: Refreshing state... [id=subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Network/networkSecurityGroups/acctsecgrp/securityRules/ssh]
module.network.azurerm_subnet.subnet[0]: Refreshing state... [id=subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Network/virtualNetworks/acctvnet/subnets/subnet1]
module.linuxservers.azurerm_network_interface.vm[0]: Refreshing state... [id=subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Network/networkInterfaces/nic-labserver-student-22-0]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-datadisk[0]: Refreshing state... [id=subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Compute/virtualMachines/labserver-student-220]
module.network.azurerm_network_security_rule.security_rule_ssh[0]: Destroying... [id=subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Network/networkSecurityGroups/acctsecgrp/securityRules/ssh]
module.network.azurerm_network_security_rule.security_rule_ssh[0]: Destruction complete after 0s
module.linuxservers.azurerm_network_security_rule.vm: Destroying... [id=subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Network/networkSecurityGroups/labserver-student-22-22-nsg/securityRules/allow_remote_22_in_all]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-datadisk[0]: Destroying... [id=subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Compute/virtualMachines/labserver-student-220]
module.linuxservers.azurerm_network_security_rule.vm: Still destroying... [id=subscriptions/017499d3-a685-4954-b727-.../securityRules/allow_remote_22_in_all, 10s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-datadisk[0]: Still destroying... [id=subscriptions/017499d3-a685-4954-b727-.../virtualMachines/labserver-student-220, 10s elapsed]
module.linuxservers.azurerm_network_security_rule.vm: Destruction complete after 10s
module.linuxservers.azurerm_virtual_machine.vm-linux-with-datadisk[0]: Still destroying... [id=subscriptions/017499d3-a685-4954-b727-.../virtualMachines/labserver-student-220, 20s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-datadisk[0]: Still destroying... [id=subscriptions/017499d3-a685-4954-b727-.../virtualMachines/labserver-student-220, 30s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-datadisk[0]: Still destroying... [id=subscriptions/017499d3-a685-4954-b727-.../virtualMachines/labserver-student-220, 40s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-datadisk[0]: Still destroying... [id=subscriptions/017499d3-a685-4954-b727-.../virtualMachines/labserver-student-220, 50s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-datadisk[0]: Still destroying... [id=subscriptions/017499d3-a685-4954-b727-.../virtualMachines/labserver-student-220, 1m0s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-datadisk[0]: Still destroying... [id=subscriptions/017499d3-a685-4954-b727-.../virtualMachines/labserver-student-220, 1m10s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-datadisk[0]: Still destroying... [id=subscriptions/017499d3-a685-4954-b727-.../virtualMachines/labserver-student-220, 1m20s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-datadisk[0]: Still destroying... [id=subscriptions/017499d3-a685-4954-b727-.../virtualMachines/labserver-student-220, 1m30s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-datadisk[0]: Still destroying... [id=subscriptions/017499d3-a685-4954-b727-.../virtualMachines/labserver-student-220, 1m40s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-datadisk[0]: Still destroying... [id=subscriptions/017499d3-a685-4954-b727-.../virtualMachines/labserver-student-220, 1m50s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-datadisk[0]: Still destroying... [id=subscriptions/017499d3-a685-4954-b727-.../virtualMachines/labserver-student-220, 2m0s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-datadisk[0]: Still destroying... [id=subscriptions/017499d3-a685-4954-b727-.../virtualMachines/labserver-student-220, 2m10s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-datadisk[0]: Still destroying... [id=subscriptions/017499d3-a685-4954-b727-.../virtualMachines/labserver-student-220, 2m20s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-datadisk[0]: Still destroying... [id=subscriptions/017499d3-a685-4954-b727-.../virtualMachines/labserver-student-220, 2m30s elapsed]
module.linuxservers.azurerm_virtual_machine.vm-linux-with-datadisk[0]: Destruction complete after 2m31s]
```

Registry and Module

Azure / [terraform-azurerm-compute](#)

Watch 25 Star 47 Fork 134

Code Issues 25 Pull requests 8 Actions Projects 0 Wiki Security Insights

Modules - No provider "azurerm" plugins meet the constraint "getting started - azure" example does not work with terraform 0.12 #104

Closed bgarcial opened this issue on Jun 13, 2019 · 4 comments

bgarcial commented on Jun 13, 2019 · edited

- Terraform Version:
) terraform -v
Terraform v0.12.1
- OS Version:
Linux Mint

Bug description:

I am using [the Getting Started Azure guideline](#), and I want to create a compute and a network [modules](#) of this way:

- Azure RM Network Module to create a vnet and subnet.
- Azure RM Compute Module to create a Linux vm.

```
# Use the network module to create a vnet and subnet
module "network" {
    source      = "Azure/network/azurerm"
    version     = "2.0.0"
    location    = "${var.location}"
    resource_group_name = "${azurerm_resource_group.rg.name}"
    address_space   = "10.0.0.0/16"
    subnet_names    = ["mySubnet"]
    subnet_prefixes = ["10.0.1.0/24"]
}
```

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Notifications Customize

Subscribe

You're not receiving notifications from this thread.

5 participants



Registry and Module

► Publish Module:

- Terraform openness for all community to publish their module to github and register to registry on terraform
- GitHub's repository for compatibility with registry need to follow constrain below
 - Public repository (Case public registry)
 - Naming: `terraform-<provider>-<module>-<xxx>`
 - Ex: `terraform-aws-rds-praparn2020`
 - Description: For inform to use this module
 - Consist required file

Registry and Module

► Publish Module:

- Requirement for file on repository
 - All resource file on root directory
 - Include: README
 - Include: LICENSE
 - Include: main.tf, variable.tf, output.tf
 - <Nest Module>: under folder ./module
 - <Example> : under folder ./example

Ref: <https://www.terraform.io/docs/modules/index.html#standard-module-structure>

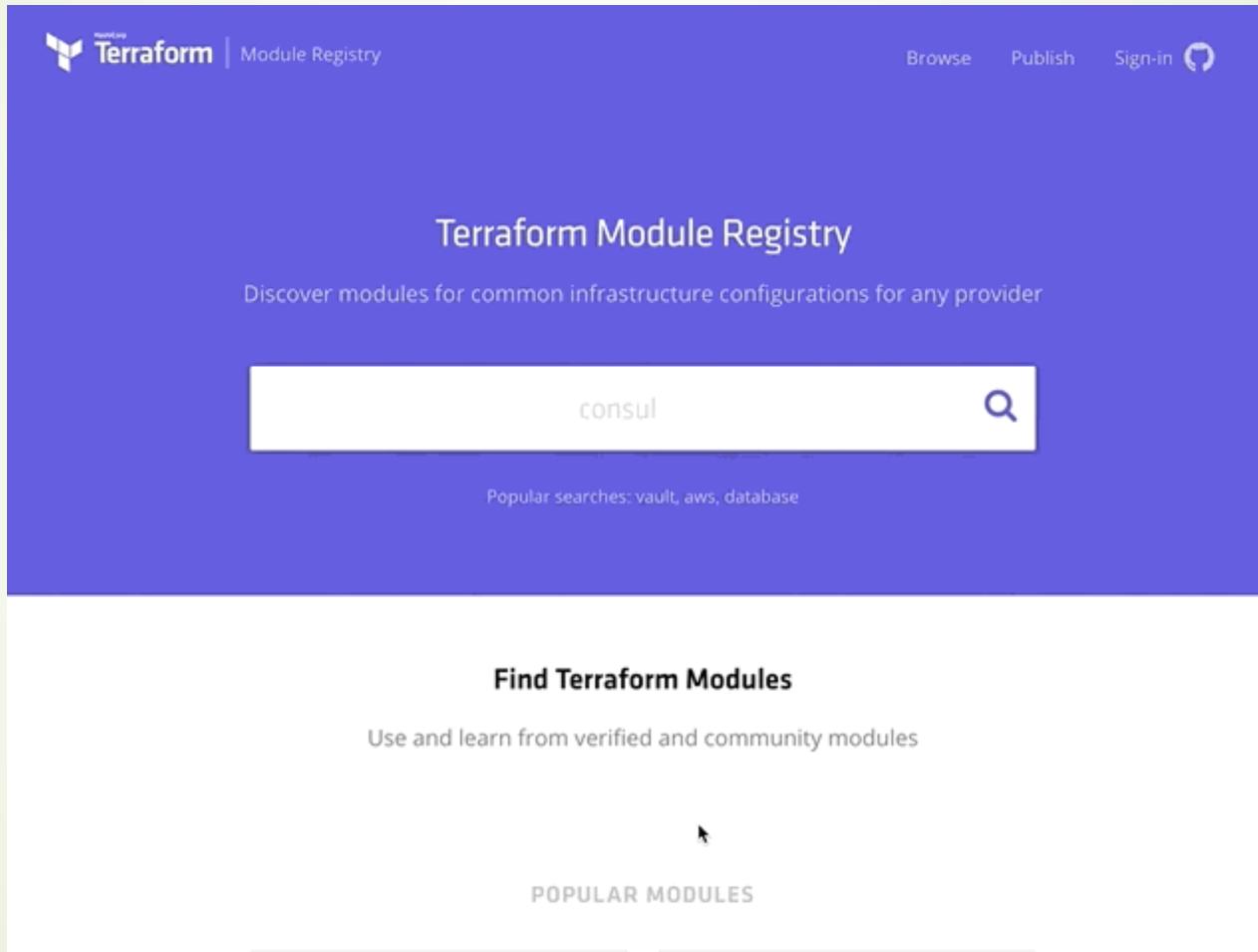
Registry and Module

```
$ tree minimal-module/
.
├── README.md
└── main.tf
└── variables.tf
└── outputs.tf
```

```
$ tree complete-module/
.
├── README.md
└── main.tf
└── variables.tf
└── outputs.tf
└── ...
└── modules/
    └── nestedA/
        ├── README.md
        ├── variables.tf
        ├── main.tf
        └── outputs.tf
    └── nestedB/
        ├── ...
    └── examples/
        └── exampleA/
            └── main.tf
        └── exampleB/
            └── ...
            └── .../
```

Ref: <https://www.terraform.io/docs/modules/index.html#standard-module-structure>

Registry and Module



Registry and Module

Azure / [terraform-azurerm-compute](#)

Code Issues 25 Pull requests 8 Actions Projects 0 Wiki Security Insights

Terraform Azure RM Compute Module <https://registry.terraform.io/modules...>

terraform-modules terraform vms

73 commits 2 branches 0 packages 17 releases 22 contributors MIT

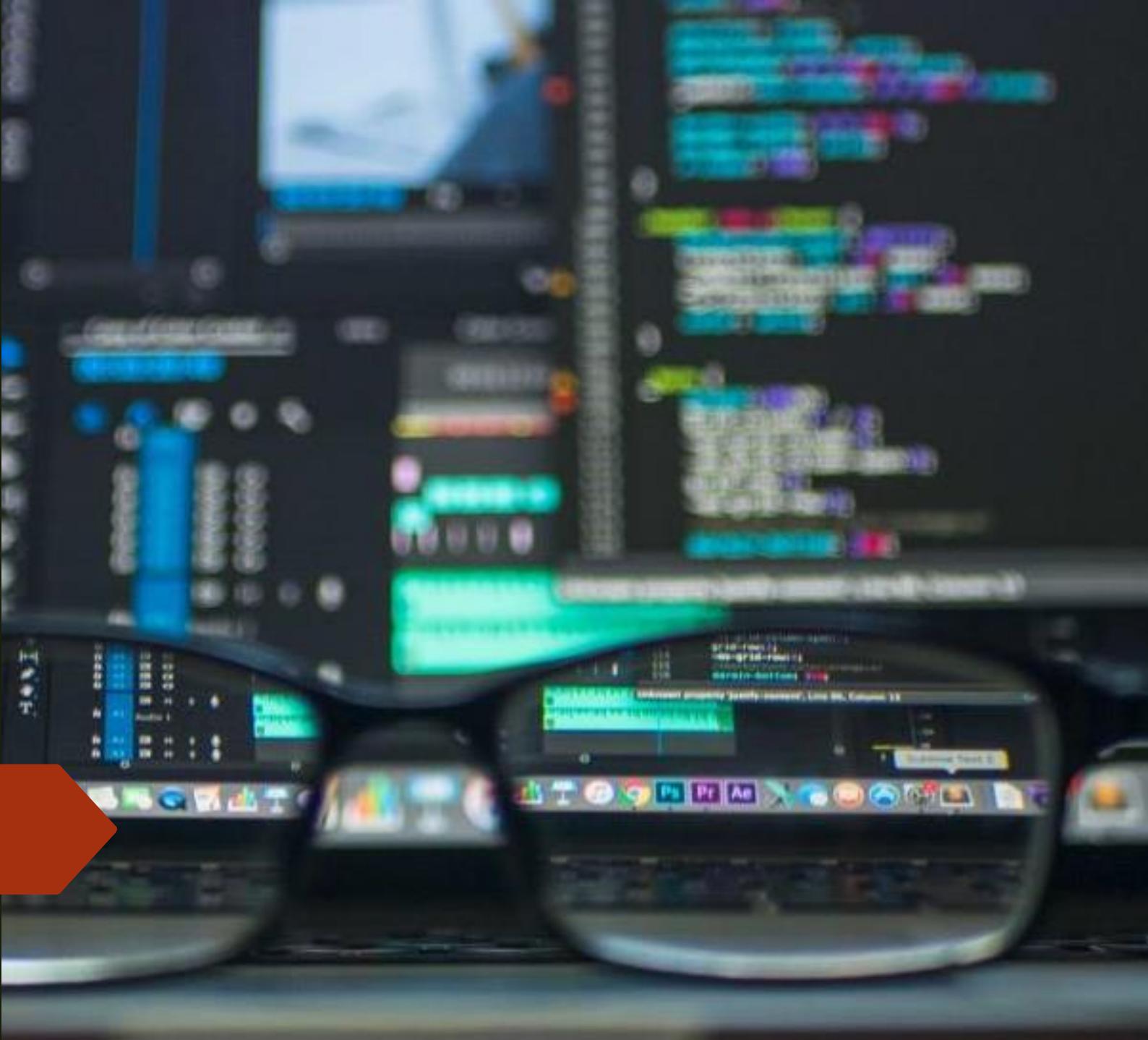
Branch: master New pull request Create new file Upload files Find file Clone or download ▾

yupwei68 upgrade to Terraform v0.12 (#122) ... ✓ Latest commit 9952198 25 days ago

File	Description	Time Ago
.github	Configure files for terraformbot (#82)	15 months ago
os	bumb os version (#94)	10 months ago
test	upgrade to Terraform v0.12 (#122)	25 days ago
.gitignore	Add Terratest Example for Compute Module (#66)	2 years ago
.travis.yml	upgrade to Terraform v0.12 (#122)	25 days ago
Dockerfile	upgrade to Terraform v0.12 (#122)	25 days ago
Gemfile	upgrade to Terraform v0.12 (#122)	25 days ago
Gopkg.lock	Add Terratest Example for Compute Module (#66)	2 years ago
Gopkg.toml	Add Terratest Example for Compute Module (#66)	2 years ago
LICENSE	Initial Release (#1)	3 years ago
README.md	updated var and readme for windows vm example (#96)	10 months ago
Rakefile	Add Terratest Example for Compute Module (#66)	2 years ago
bug_report.md	Zunlihu issue template (#75)	17 months ago
main.tf	upgrade to Terraform v0.12 (#122)	25 days ago
outputs.tf	upgrade to Terraform v0.12 (#122)	25 days ago
variables.tf	upgrade to Terraform v0.12 (#122)	25 days ago

Ref: <https://github.com/Azure/terraform-azurerm-compute>

Workshop: Registry and Module



Workshop: Module

```
[ubuntu@ip-172-31-27-251:~/terraform_202003/Workshop_2.2_Module/01_Module$ az vm image list --output table  
You are viewing an offline list of images, use --all to retrieve an up-to-date list
```

Offer	Publisher	Sku	Urn	UrnAlias	Version
CentOS	OpenLogic	7.5	OpenLogic:CentOS:7.5:latest	CentOS	latest
CoreOS	CoreOS	Stable	CoreOS:CoreOS:Stable:latest	CoreOS	latest
debian-10	Debian	10	Debian:debian-10:10:latest	Debian	latest
openSUSE-Leap	SUSE	42.3	SUSE:openSUSE-Leap:42.3:latest	openSUSE-Leap	latest
RHEL	RedHat	7-LVM	RedHat:RHEL:7-LVM:latest	RHEL	latest
SLES	SUSE	15	SUSE:SLES:15:latest	SLES	latest
UbuntuServer	Canonical	18.04-LTS	Canonical:UbuntuServer:18.04-LTS:latest	UbuntuLTS	latest
WindowsServer	MicrosoftWindowsServer	2019-Datacenter	MicrosoftWindowsServer:WindowsServer:2019-Datacenter:latest	Win2019Datacenter	latest
WindowsServer	MicrosoftWindowsServer	2016-Datacenter	MicrosoftWindowsServer:WindowsServer:2016-Datacenter:latest	Win2016Datacenter	latest
WindowsServer	MicrosoftWindowsServer	2012-R2-Datacenter	MicrosoftWindowsServer:WindowsServer:2012-R2-Datacenter:latest	Win2012R2Datacenter	latest
WindowsServer	MicrosoftWindowsServer	2012-Datacenter	MicrosoftWindowsServer:WindowsServer:2012-Datacenter:latest	Win2012Datacenter	latest
WindowsServer	MicrosoftWindowsServer	2008-R2-SP1	MicrosoftWindowsServer:WindowsServer:2008-R2-SP1:latest	Win2008R2SP1	latest



```
azurerm_resource_group.labrscgrp: Modifications complete after 0s [id=/subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22]  
module.linuxservers.azurevm_network_interface.vm[0]: Modifications complete after 1s [id=/subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Network/networkInterfaces/nic-labserver-student-22-0]  
module.linuxservers.azurevm_virtual_machine.vm-linux-with-datadisk[0]: Creating...  
module.linuxservers.azurevm_virtual_machine.vm-linux-with-datadisk[0]: Still creating... [10s elapsed]  
module.linuxservers.azurevm_virtual_machine.vm-linux-with-datadisk[0]: Still creating... [20s elapsed]  
module.linuxservers.azurevm_virtual_machine.vm-linux-with-datadisk[0]: Still creating... [30s elapsed]  
module.linuxservers.azurevm_virtual_machine.vm-linux-with-datadisk[0]: Still creating... [40s elapsed]  
module.linuxservers.azurevm_virtual_machine.vm-linux-with-datadisk[0]: Still creating... [50s elapsed]  
module.linuxservers.azurevm_virtual_machine.vm-linux-with-datadisk[0]: Still creating... [1m0s elapsed]  
module.linuxservers.azurevm_virtual_machine.vm-linux-with-datadisk[0]: Still creating... [1m10s elapsed]  
module.linuxservers.azurevm_virtual_machine.vm-linux-with-datadisk[0]: Still creating... [1m20s elapsed]  
module.linuxservers.azurevm_virtual_machine.vm-linux-with-datadisk[0]: Still creating... [1m30s elapsed]  
module.linuxservers.azurevm_virtual_machine.vm-linux-with-datadisk[0]: Still creating... [1m40s elapsed]  
module.linuxservers.azurevm_virtual_machine.vm-linux-with-datadisk[0]: Still creating... [1m50s elapsed]  
module.linuxservers.azurevm_virtual_machine.vm-linux-with-datadisk[0]: Still creating... [2m0s elapsed]  
module.linuxservers.azurevm_virtual_machine.vm-linux-with-datadisk[0]: Still creating... [2m10s elapsed]  
module.linuxservers.azurevm_virtual_machine.vm-linux-with-datadisk[0]: Still creating... [2m20s elapsed]  
module.linuxservers.azurevm_virtual_machine.vm-linux-with-datadisk[0]: Still creating... [2m30s elapsed]  
module.linuxservers.azurevm_virtual_machine.vm-linux-with-datadisk[0]: Creation complete after 2m30s [id=/subscriptions/017499d3-a685-4954-b727-273a4a7f939f/resourceGroups/labresourcegroup-student-22/providers/Microsoft.Compute/virtualMachines/labserver-student-22]
```

Q&A





Terraform Cloud for Team Collaboration



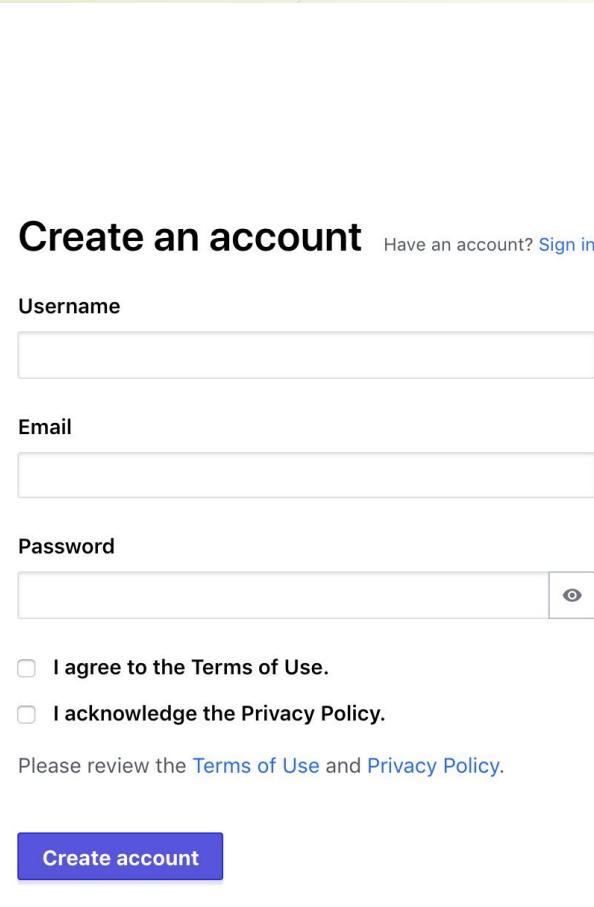
Introduction to terraform cloud

- ▶ On previous chapter. We all done operate for provision via remote machine.
- ▶ Anyway on enterprise environment. We have a huge of challenge for provision
 - ▶ Many member on team need to provision and keep workspace on single place
 - ▶ Centralize track change and log result
 - ▶ Remove credential from devops local machine
 - ▶ Approval workflow for operate via reviewed
 - ▶ etc

Introduction to terraform cloud

- ▶ Terraform have cloud platform for total free for support this purpose call “Terraform Cloud”
- ▶ Provide full feature of provisioning life cycle
 - ▶ Signup and create organization
 - ▶ Version Control System (VCS) (Optional) for integrate with
 - ▶ GitHub / GitHub Enterprise
 - ▶ GitLab.com / GitLab EE / GitLab CE
 - ▶ Bitbucket Cloud / Bitbucket server
 - ▶ Azure DevOps Server / Azure DevOps Service
 - ▶ Create workspace for collaboration
 - ▶ Operate credential and variable

Introduction to terraform cloud



The image shows a screenshot of the Terraform Cloud 'Create an account' page. It features a large orange arrow pointing right on the left side. The main form has fields for 'Username', 'Email', and 'Password'. Below the password field is a visibility toggle icon. There are two checkboxes for 'I agree to the Terms of Use.' and 'I acknowledge the Privacy Policy.', both of which are unchecked. A note at the bottom says 'Please review the [Terms of Use](#) and [Privacy Policy](#)'. A blue 'Create account' button is at the bottom.

Create an account Have an account? [Sign in](#)

Username

Email

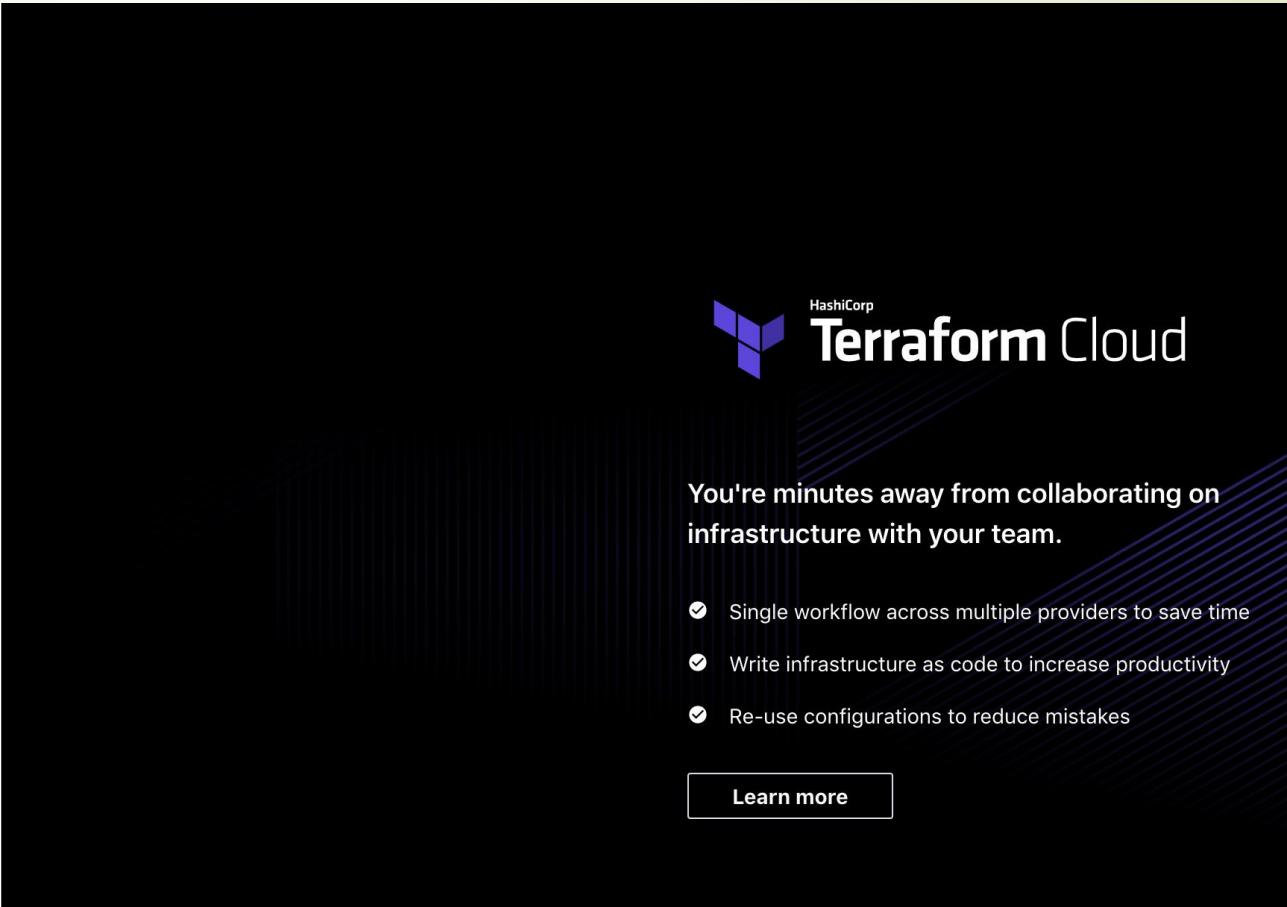
Password 👁

I agree to the [Terms of Use](#).

I acknowledge the [Privacy Policy](#).

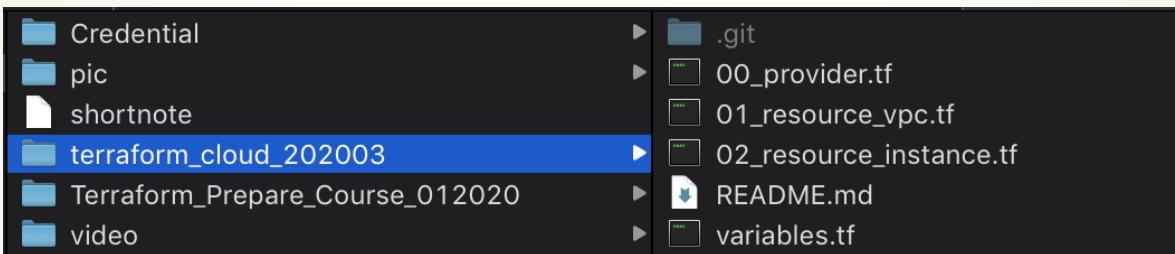
Please review the [Terms of Use](#) and [Privacy Policy](#).

[Create account](#)



Introduction to terraform cloud

- ▶ Step 1: Create Github's repository from source folder “terraform_cloud_xxxxxx”



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner / Repository name

Great repository names are short and memorable. Need inspiration? How about [sturdy-adventure](#)?

Description (optional)

Public Anyone can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

Initialize this repository with a README This will let you immediately clone the repository to your computer.

Add .gitignore: Add a license:

Create repository

© 2020 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

praparn / laC-TerraformThailand-Master

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Quick setup — if you've done this kind of thing before
Set up in Desktop or HTTPS SSH https://github.com/praparn/laC-TerraformThailand-Master.git
Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line
`echo "# laC-TerraformThailand-Master" > README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/praparn/laC-TerraformThailand-Master.git
git push -u origin master`

...or push an existing repository from the command line
`git remote add origin https://github.com/praparn/laC-TerraformThailand-Master.git
git push -u origin master`

...or import code from another repository
You can initialize this repository with code from a Subversion, Mercurial, or TFS project.
Import code

ProTip! Use the URL for this page when adding GitHub as a remote.

© 2020 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

Introduction to terraform cloud

- ▶ Step 1: Create Github's repository from source folder “terraform_cloud_xxxxxx”

```
Users > paparnlueangphoonlap > Work > Terraform > Resource > terraform_cloud_202003 > 00_provider.tf
1 provider "google" {
2   credentials = file("GCloud-Credential.json")
3   project     = var.project_id
4   region      = var.region
5   zone        = var.zone
6 }
```

```
Users > paparnlueangphoonlap > Work > Terraform > Resource > terraform_cloud_202003 > 02_resource_instance.tf
1 resource "google_compute_instance" "labserver" {
2   name          = var.compute
3   machine_type = var.compute_type
4   zone         = var.zone
5   hostname     = var.compute_name
6   tags          = [var.compute]
7   can_ip_forward = true
8   boot_disk {
9     initialize_params {
10       image = var.image
11       size  = var.disk_size
12     }
13   }
14
15   network_interface {
16     network = google_compute_network.lab_vpc_network.name
17     access_config {
18     }
19   }
20   metadata_startup_script = var.user_data
21   metadata = {
22     ssh-keys = "ubuntu:${file("terraform_gcloud.pub")}"
23   }
24 }
```

Introduction to terraform cloud

- ▶ Step 1: Create Github's repository from source folder “terraform_cloud_xxxxxx”

```
[praparns-MacBook-Pro:terraform_cloud_202003 praparn$ ls
00_provider.tf      02_resource_instance.tf  terraform.tfvars
01_resource_vpc.tf   README.md                variables.tf
[praparns-MacBook-Pro:terraform_cloud_202003 praparn$ echo "# IaC-TerraformThailand-Master" >> README.md
[praparns-MacBook-Pro:terraform_cloud_202003 praparn$ git init
Initialized empty Git repository in /Users/praparnlueangphoonlap/Work/Terraform/Resource/terraform_cloud_202003/.git/
[praparns-MacBook-Pro:terraform_cloud_202003 praparn$ git add -A
[praparns-MacBook-Pro:terraform_cloud_202003 praparn$ git commit -m "First Commit"
[master (root-commit) bc7be00] First Commit
 6 files changed, 116 insertions(+)
 create mode 100755 00_provider.tf
 create mode 100755 01_resource_vpc.tf
 create mode 100755 02_resource_instance.tf
 create mode 100644 README.md

[praparns-MacBook-Pro:terraform_cloud_202003 praparn$ git remote add origin https://github.com/praparn/IaC-TerraformThailand-Master.git
[praparns-MacBook-Pro:terraform_cloud_202003 praparn$ git push -u origin master
```

```
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 1.49 KiB | 1.49 MiB/s, done.
Total 8 (delta 0), reused 0 (delta 0)
To https://github.com/praparn/IaC-TerraformThailand-Master.git
 * [new branch]    master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
praparns-MacBook-Pro:terraform_cloud_202003 praparn$
```

Introduction to terraform cloud

- ▶ Step 1: Create Github's repository from source folder “terraform_cloud_xxxxxx”

The screenshot shows a GitHub repository page. At the top, the repository name is 'paparn / laC-TerraformThailand-Master'. To the right are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). Below the header is a navigation bar with links: 'Code' (highlighted in orange), 'Issues 0', 'Pull requests 0', 'Actions', 'Projects 0', 'Wiki', 'Security', 'Insights', and 'Settings'. The main content area displays the repository details: 'laC-TerraformThailand-Master', 'Manage topics', 'Edit', '1 commit', '1 branch', '0 packages', '0 releases', and '1 contributor'. Below this, there are buttons for 'Branch: master ▾', 'New pull request', 'Create new file', 'Upload files', 'Find file', and a green 'Clone or download ▾' button. The commit history lists a single commit by 'paparn' titled 'First Commit' made 1 minute ago. The commit details show files: '00_provider.tf', '01_resource_vpc.tf', '02_resource_instance.tf', 'README.md', 'terraform.tfvars', and 'variables.tf', all with first commits made 1 minute ago.

File	Commit	Time
00_provider.tf	First Commit	1 minute ago
01_resource_vpc.tf	First Commit	1 minute ago
02_resource_instance.tf	First Commit	1 minute ago
README.md	First Commit	1 minute ago
terraform.tfvars	First Commit	1 minute ago
variables.tf	First Commit	1 minute ago

Introduction to terraform cloud

► Step 2: Access url: <https://app.terraform.io/> and signup for free account (Include confirm email-back)

Create an account Have an account? Sign in

Username

Email

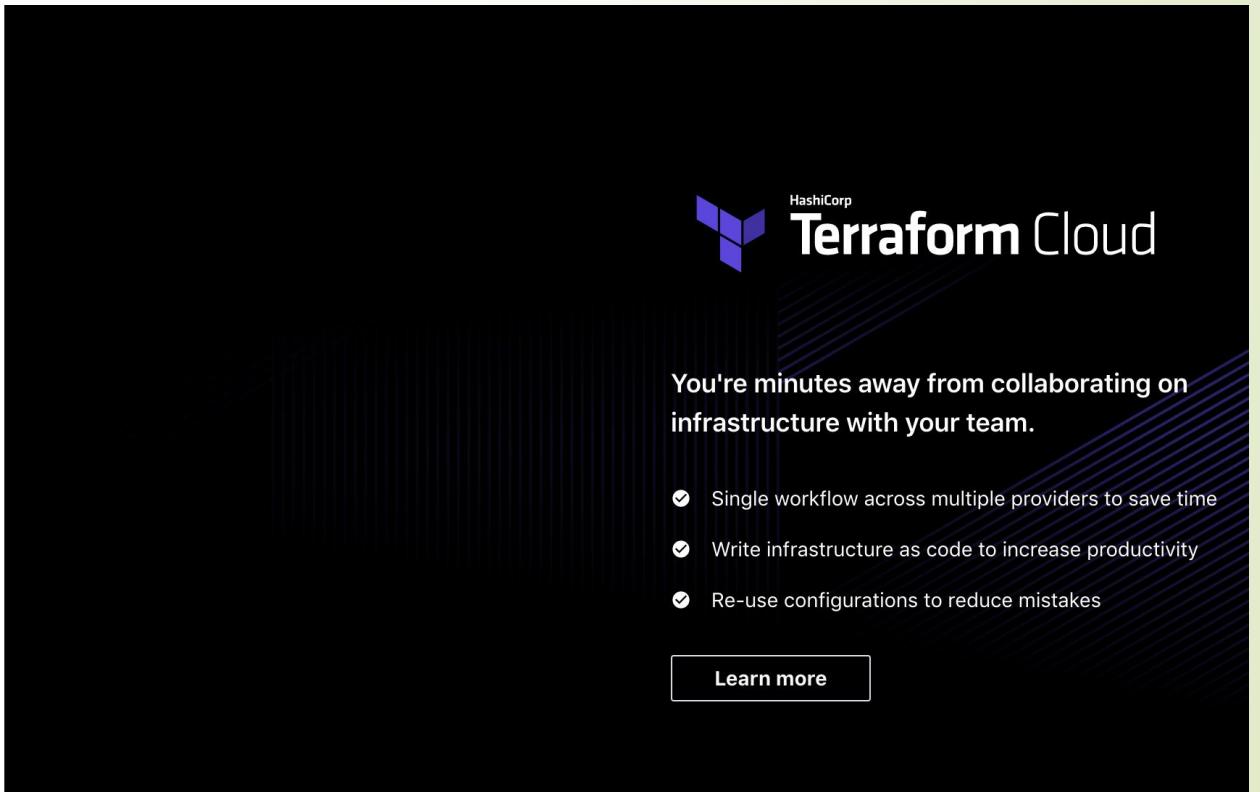
Password

I agree to the [Terms of Use](#).

I acknowledge the [Privacy Policy](#).

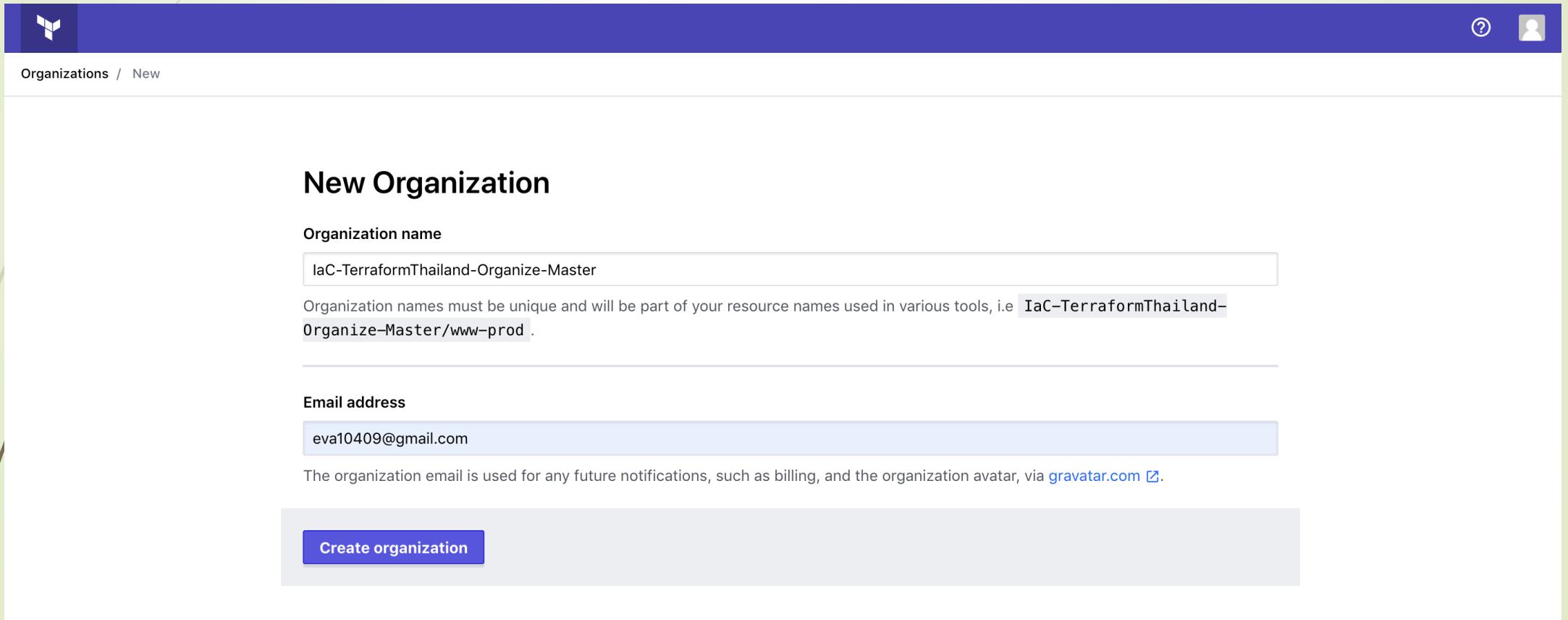
Please review the [Terms of Use](#) and [Privacy Policy](#).

Create account



Introduction to terraform cloud

► Step 3: Login and create organization name



The screenshot shows the 'New Organization' creation interface. At the top, there's a blue header bar with the Terraform logo, a help icon, and a user profile icon. Below the header, the URL 'Organizations / New' is visible. The main section is titled 'New Organization'. It has two input fields: 'Organization name' containing 'IaC-TerraformThailand-Organize-Master' and 'Email address' containing 'eva10409@gmail.com'. A note below the email field states: 'The organization email is used for any future notifications, such as billing, and the organization avatar, via [gravatar.com](#)'. At the bottom is a blue 'Create organization' button.

Organizations / New

New Organization

Organization name

Organization names must be unique and will be part of your resource names used in various tools, i.e. IaC-TerraformThailand-Organize-Master/www-prod .

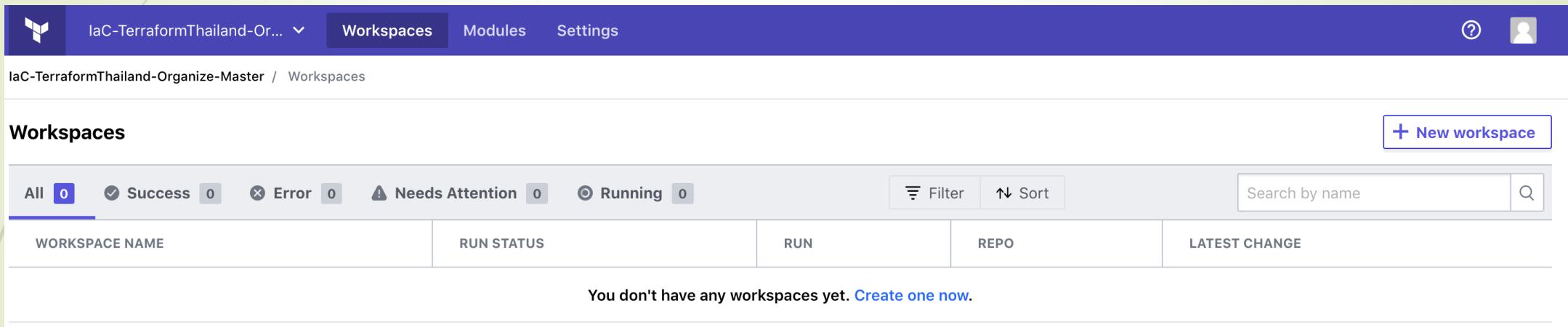
Email address

The organization email is used for any future notifications, such as billing, and the organization avatar, via [gravatar.com](#).

Create organization

Introduction to terraform cloud

► Step 4: Create workspace with VCS (Version Control Provider)



The screenshot shows the Terraform Cloud interface for managing workspaces. At the top, there's a navigation bar with a logo, the workspace name "IaC-TerraformThailand-Or...", and links for "Workspaces", "Modules", and "Settings". On the far right of the bar are a help icon and a user profile icon. Below the bar, the breadcrumb navigation shows "IaC-TerraformThailand-Organize-Master / Workspaces". The main area is titled "Workspaces" and features a "New workspace" button with a plus sign. A filter bar at the top allows selecting "All" (0), "Success" (0), "Error" (0), "Needs Attention" (0), or "Running" (0) workspaces, along with "Filter" and "Sort" options and a search bar. The table below has columns for "WORKSPACE NAME", "RUN STATUS", "RUN", "REPO", and "LATEST CHANGE". A message in the center of the table says "You don't have any workspaces yet. [Create one now.](#)".

Introduction to terraform cloud

► Step 4: Create workspace with VCS (Version Control Provider)

Create a new Workspace

Workspaces allow you to organize infrastructure and collaborate on Terraform runs.

1 Connect to VCS

2 Choose a repository

3 Configure settings

Connect to a version control provider

Choose the version control provider that hosts the Terraform configuration for this workspace.

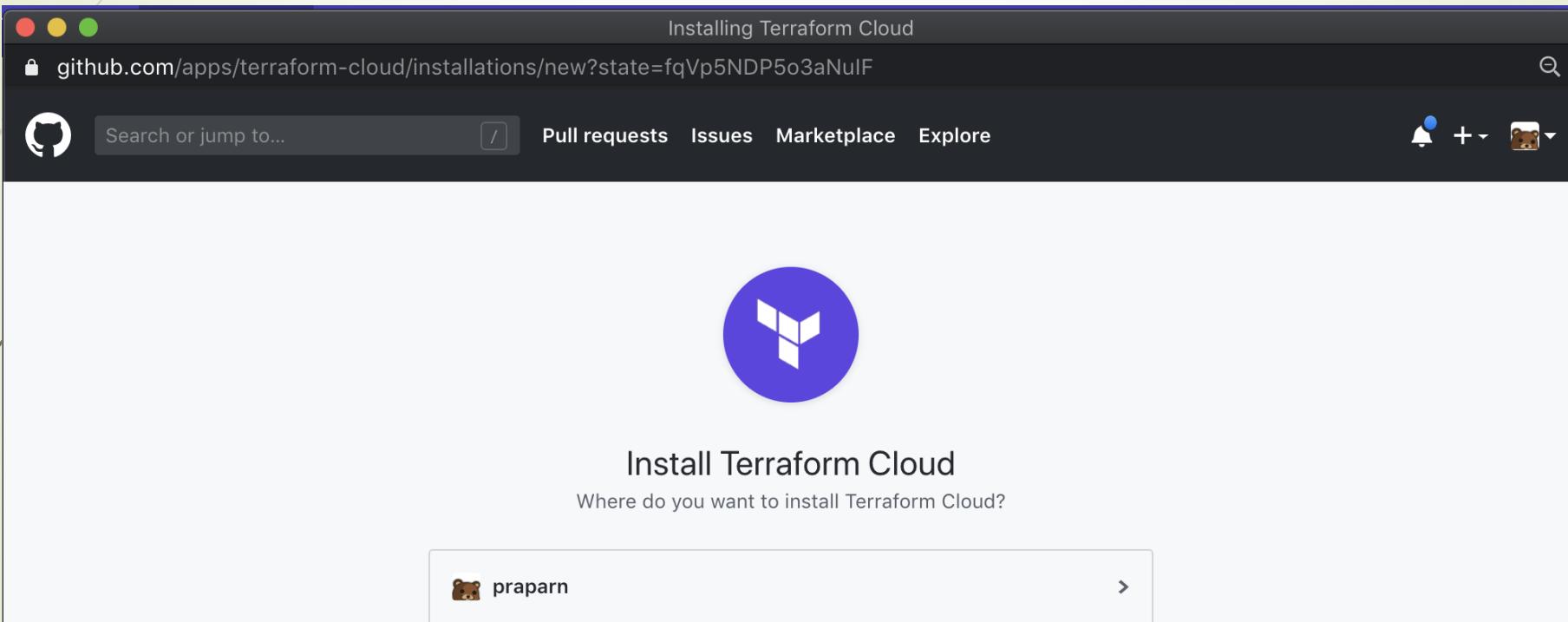


GitHub

[Connect to a different VCS](#)

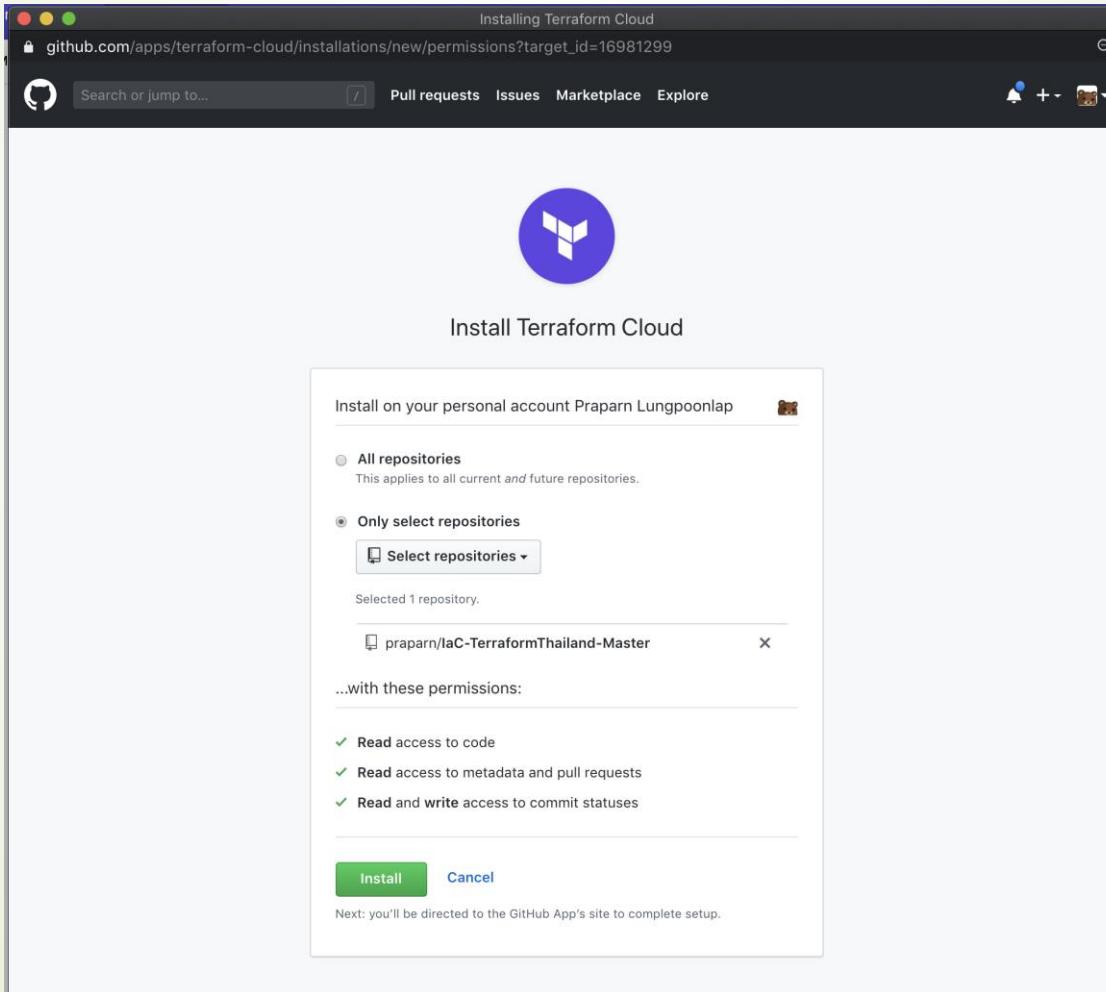
Introduction to terraform cloud

► Step 4: Create workspace with VCS (Version Control Provider)



Introduction to terraform cloud

► Step 4: Create workspace with VCS (Version Control Provider)



Introduction to terraform cloud

► Step 4: Create workspace with VCS (Version Control Provider)

Create a new Workspace

Workspaces allow you to organize infrastructure and collaborate on Terraform runs.

 Connect to VCS

 Choose a repository

 Configure settings

Choose a repository

Choose the repository that hosts your Terraform source code. We'll watch this for commits and pull requests.



praparn ▾ 1 repository

Filter

IaC-TerraformThailand-Master >

Can't see your repository? Enter its ID below, e.g. acme-corp/infrastructure :

acme-corp/infrastructure >

Introduction to terraform cloud

► Step 4: Create workspace with VCS (Version Control Provider)

Create a new Workspace

Workspaces allow you to organize infrastructure and collaborate on Terraform runs.

Connect to VCS

Choose a repository

3 Configure settings

Configure settings

Workspace Name

IaC-TerraformThailand-Master

The name of your workspace is unique and used in tools, routing, and UI. Dashes, underscores, and alphanumeric characters are permitted. Learn more about [naming workspaces](#).

▼ Advanced options

Create workspace

Cancel

Introduction to terraform cloud

► Step 5: Configure Variable as below

Terraform Variables

These [Terraform variables](#) are set using a `terraform.tfvars` file. To use interpolation or set a non-string value for a variable, click its HCL checkbox.

Key	Value
Key	<input type="text" value="vpc_id"/>
Value	<input type="text" value="vpc-4483e921"/> <input type="checkbox"/> HCL ⓘ <input type="checkbox"/> Sensitive ⓘ
Description	<input type="text" value="description (optional)"/>
<button>Save variable</button> <button>Cancel</button>	

Terraform Variables

These [Terraform variables](#) are set using a `terraform.tfvars` file. To use interpolation or set a non-string value for a variable, click its HCL checkbox.

Key	Value	...
vpc_id	vpc-4483e921	...

Introduction to terraform cloud

► Step 5: Configure Variable as below

No	Variable	Value	Remark
1	project_id	<Reference in Control Sheet>	Terraform Variable
2	region	<Reference in Control Sheet>	
3	zone	<Reference in Control Sheet>	
4	gcompute_network_name	<Reference in Control Sheet>	
5	gcompute_createsubnet	<Reference in Control Sheet>	
6	compute_name	<Reference in Control Sheet>	
7	compute	<Reference in Control Sheet>	
8	compute_type	<Reference in Control Sheet>	
9	image	<Reference in Control Sheet>	
10	disk_size	<Reference in Control Sheet>	
11	user_data	<Reference in Control Sheet>	

Introduction to terraform cloud

► Step 5: Configure Variable as below

Variables

These variables are used for all plans and applies in this workspace. Workspaces using Terraform 0.10.0 or later can also load default values from any `*.auto.tfvars` files in the configuration.

Sensitive variables are hidden from view in the UI and API, and can't be edited. (To change a sensitive variable, delete and replace it.) Sensitive variables can still appear in Terraform logs if your configuration is designed to output them.

When setting many variables at once, the [Terraform Cloud Provider](#) or the [variables API](#) can often save time.

Terraform Variables

These [Terraform variables](#) are set using a `terraform.tfvars` file. To use interpolation or set a non-string value for a variable, click its HCL checkbox.

Key	Value	...
project_id	polar-decorator-267303	...
region	asia-southeast1	<input type="checkbox"/> HCL ⓘ <input type="checkbox"/> Sensitive ⓘ

Description
description (optional)

Save variable **Cancel**

Introduction to terraform cloud

► Step 5: Configure Variable as below

Variables

These variables are used for all plans and applies in this workspace. Workspaces using Terraform 0.10.0 or later can also load default values from any `*.auto.tfvars` files in the configuration.

Sensitive variables are hidden from view in the UI and API, and can't be edited. (To change a sensitive variable, delete and replace it.) Sensitive variables can still appear in Terraform logs if your configuration is designed to output them.

When setting many variables at once, the [Terraform Cloud Provider](#) or the [variables API](#) can often save time.

Terraform Variables

These [Terraform variables](#) are set using a `terraform.tfvars` file. To use interpolation or set a non-string value for a variable, click its HCL checkbox.

Key	Value	
project_id	polar-decorator-267303	...
region	asia-southeast1	...
zone	asia-southeast1-a	...
gcompute_network_name	lab-vpc-student-7	...

Key	Value	HCL	Sensitive
gcomputeCreatesubnet	true	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Description	<input type="text" value="description (optional)"/>
-------------	---

Save variable **Cancel**

Introduction to terraform cloud

➡ Step 5: Configure Variable as below

Terraform Variables		
These Terraform variables are set using a <code>terraform.tfvars</code> file. To use interpolation or set a non-string value for a variable, click its HCL checkbox.		
Key	Value	
<code>project_id</code>	polar-decorator-267303	...
<code>region</code>	asia-southeast1	...
<code>zone</code>	asia-southeast1-a	...
<code>gcompute_network_name</code>	lab-vpc-student-g7	...
<code>gcomputeCreatesubnet</code> <small>HCL</small>	true	...
<code>compute_name</code>	lab-computer-student-g7.terraform.local	...
<code>compute</code>	lab-computer-student-g7	...
<code>compute_type</code>	f1-micro	...
<code>image</code>	ubuntu-1804-lts	...
<code>disk_size</code> <small>HCL</small>	10	...
<code>user_data</code>	<pre>curl https://raw.githubusercontent.com/praparn/sourcesetup/master/standard_docker_aws.sh > /tmp/setup.sh && chmod +x /tmp/setup.sh && /tmp/setup.sh</pre>	...

Introduction to terraform cloud

- ▶ Step 6-7: Add find credential and Git commit and push with comment “Git Action Credential”

```
[praparns-MacBook-Pro:terraform_cloud_202003 praparn$ git add -A
[praparns-MacBook-Pro:terraform_cloud_202003 praparn$ git commit -m "Git Action Credential"
[master 3d5a723] Git Action Credential
 2 files changed, 13 insertions(+)
   create mode 100644 GCloud-Credential.json
[praparns-MacBook-Pro:terraform_cloud_202003 praparn$ git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 2.01 KiB | 2.01 MiB/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/praparn/IaC-TerraformThailand-Master.git
 7cf4f5c..3d5a723  master -> master
praparns-MacBook-Pro:terraform_cloud_202003 praparn$ ]
```

Introduction to terraform cloud

► Step 8: Back to Terraform Cloud. Choose Tab “Run”

The screenshot shows the Terraform Cloud interface. At the top, there is a navigation bar with a logo, the workspace name "laC-TerraformThailand-Or...", and tabs for "Workspaces", "Modules", and "Settings". Below the navigation bar, the breadcrumb path shows "laC-TerraformThailand-Organize-Master / Workspaces / laC-TerraformThailand-Master / Runs". The main content area is titled "laC-TerraformThailand-Master" and has tabs for "Runs", "States", "Variables", "Settings", and "Queue plan". The "Runs" tab is currently selected. Below the tabs, there is a section titled "Current Run" which displays a card for a "Git Action Credential". The card includes a small bear icon, the text "Git Action Credential CURRENT", and the identifier "#run-DBe7fT5BPZMvWjxv | praparn triggered from GitHub | Branch master | 3d5a723". To the right of the card, there is an orange button with the text "NEEDS CONFIRMATION" and a timestamp "2 minutes ago".

Introduction to terraform cloud

► Step 9: Verify terraform plan and click “Confirm & Apply”

IaC-TerraformThailand-Master ⓘ

Runs States Variables Settings Queue plan CURRENT

! NEEDS CONFIRMATION Git Action Credential

paparn triggered a run from GitHub 3 minutes ago Run Details

Plan finished 4 minutes ago Resources: 3 to add, 0 to change, 0 to destroy

Started 3 minutes ago > Finished 3 minutes ago

Download Sentinel mocks Sentinel mocks can be used for testing your Sentinel policies

View raw log Top Bottom Expand Full screen

```
}
}

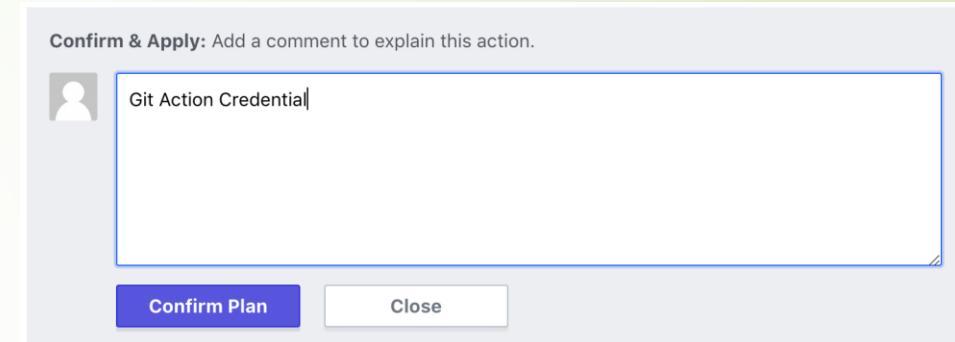
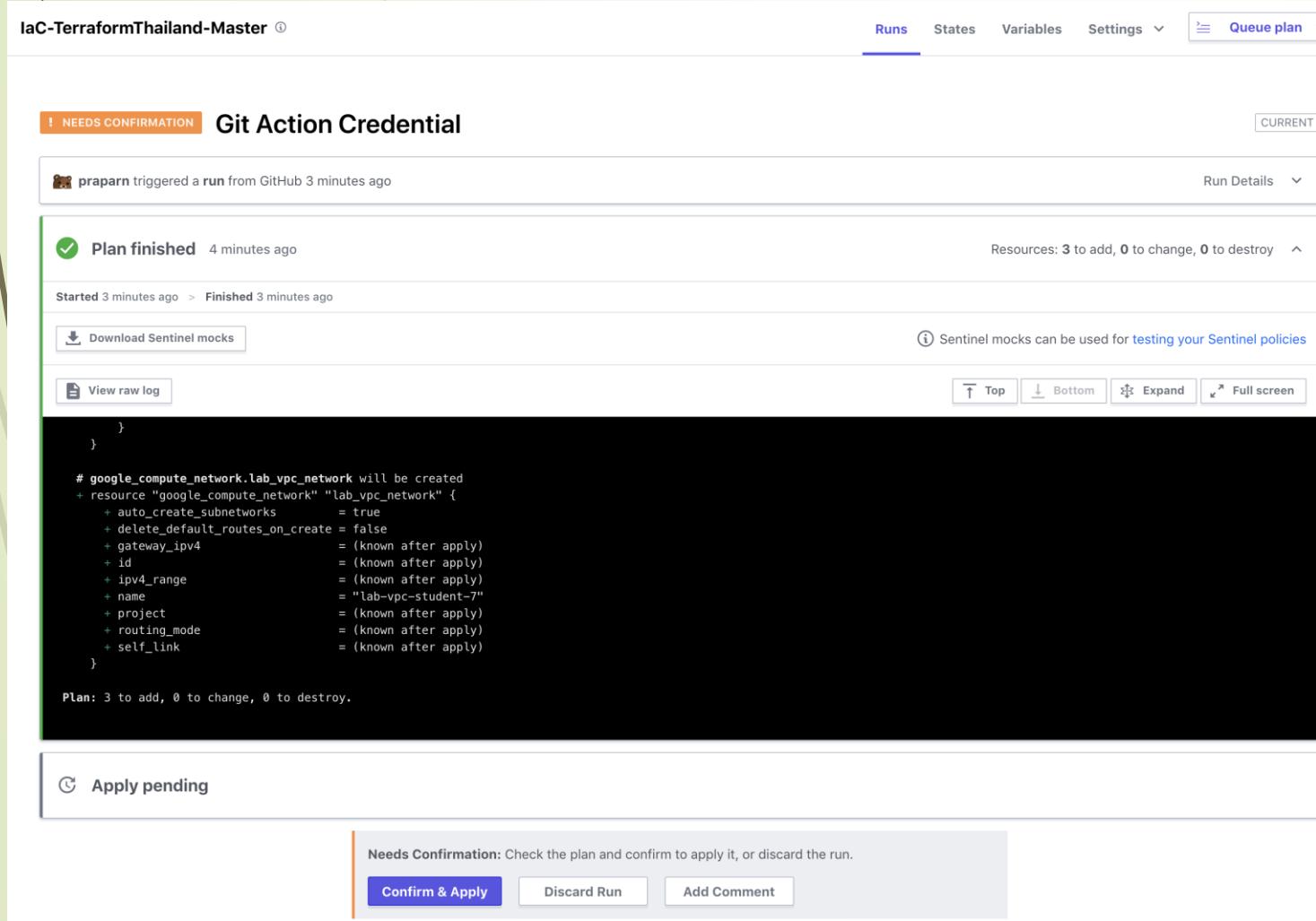
# google_compute_network.lab_vpc_network will be created
+ resource "google_compute_network" "lab_vpc_network" {
    + auto_create_subnetworks      = true
    + delete_default_routes_on_create = false
    + gateway_ipv4                = (known after apply)
    + id                           = (known after apply)
    + ipv4_range                   = (known after apply)
    + name                         = "lab-vpc-student-7"
    + project                      = (known after apply)
    + routing_mode                 = (known after apply)
    + self_link                     = (known after apply)
}
```

Plan: 3 to add, 0 to change, 0 to destroy.

Apply pending

Needs Confirmation: Check the plan and confirm to apply it, or discard the run.

Confirm & Apply Discard Run Add Comment



Introduction to terraform cloud

► Step 10: Monitor terraform progress

The screenshot shows the Terraform Cloud interface for a "Git Action Credential" run. The top bar indicates the status is "APPLYING" and the current run is "CURRENT".

- Trigger:** praparn triggered a run from GitHub 3 minutes ago. [Run Details](#)
- Status:** Plan finished 6 minutes ago. Resources: 3 to add, 0 to change, 0 to destroy.
- Apply:** Apply running a few seconds ago. Started a few seconds ago.
 - [View raw log](#)
 - Log controls: Following log, Top, Bottom, Expand, Full screen

The log output shows the following Terraform activity:

```
Terraform v0.12.20
Initializing plugins and modules...
2020/02/11 13:58:05 [DEBUG] Using modified User-Agent: Terraform/0.12.20 TFC/664d7ed310
google_compute_network.lab_vpc_network: Creating...
google_compute_network.lab_vpc_network: Still creating... [10s elapsed]
google_compute_network.lab_vpc_network: Still creating... [20s elapsed]
```

Introduction to terraform cloud

► Step 10: Monitor terraform progress

✓ APPLIED **Git Action Credential** CURRENT

 praparn triggered a run from GitHub 3 minutes ago Run Details ▾

✓ Plan finished 7 minutes ago Resources: 3 to add, 0 to change, 0 to destroy ▾
Started 7 minutes ago > Finished 7 minutes ago

[Download Sentinel mocks](#)  Sentinel mocks can be used for testing your Sentinel policies

[View raw log](#)  Top  Bottom  Expand  Full screen

```
}
}

# google_compute_network.lab_vpc_network will be created
+ resource "google_compute_network" "lab_vpc_network" {
  + auto_create_subnetworks = true
  + delete_default_routes_on_create = false
  + gateway_ipv4 = (known after apply)
  + id = (known after apply)
  + ipv4_range = (known after apply)
  + name = "lab-vpc-student-7"
  + project = (known after apply)
  + routing_mode = (known after apply)
  + self_link = (known after apply)
}
```

Plan: 3 to add, 0 to change, 0 to destroy.

✓ Apply finished a few seconds ago Resources: 3 added, 0 changed, 0 destroyed ▾



Introduction to terraform cloud

► Step 10: Monitor terraform progress

Apply finished a few seconds ago

Started 2 minutes ago > Finished a minute ago

Resources: 3 added, 0 changed, 0 destroyed ^

[View raw log](#)

2020/02/11 13:58:05 [DEBUG] Using modified User-Agent: Terraform/0.12.20 TFC/664d7ed310
google_compute_network.lab_vpc_network: Creating...
google_compute_network.lab_vpc_network: Still creating... [10s elapsed]
google_compute_network.lab_vpc_network: Still creating... [20s elapsed]
google_compute_network.lab_vpc_network: Still creating... [30s elapsed]
google_compute_network.lab_vpc_network: Creation complete after 37s [id=projects/polar-decorator-267303/global/networks/lab-vpc-student-7]
google_compute_firewall.lab_firewall: Creating...
google_compute_instance.labserver: Creating...
google_compute_firewall.lab_firewall: Creation complete after 7s [id=projects/polar-decorator-267303/global/firewalls/allow-ssh]
google_compute_instance.labserver: Creation complete after 9s [id=projects/polar-decorator-267303/zones/asia-southeast1-a/instances/lab-computer-student-7]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Outputs:

google_compute_instance_labserver_id = 8064195906681900538
google_compute_instance_labserver_nat_ip = 34.87.80.107

State versions created:
laC-TerraformThailand-Organize-Master/laC-TerraformThailand-Master#sv-UfHokXKqR3qAVpNf (Feb 11, 2020 20:58:53 pm)

↑ Top ↓ Bottom ⌂ Expand ⌄ Full screen

Introduction to terraform cloud

► Step 11: Check result from gcloud-cli and Test access

```
[ubuntu@ip-172-31-31-216:~$ gcloud compute instances list --filter="name='lab-computer-student-7'"
NAME          ZONE      MACHINE_TYPE  PREEMPTIBLE  INTERNAL_IP  EXTERNAL_IP  STATUS
lab-computer-student-7  asia-southeast1-a  f1-micro        10.148.0.2   34.87.80.107  RUNNING
[ubuntu@ip-172-31-31-216:~$ ssh -i ~/terraform_gcloud ubuntu@34.87.80.107
The authenticity of host '34.87.80.107 (34.87.80.107)' can't be established.
ECDSA key fingerprint is SHA256:kkCQRL5QUI90472LjQXgvxiqBHjqtGRd8dMe3yxJ6Os.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '34.87.80.107' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.0.0-1029-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 System information as of Tue Feb 11 14:12:44 UTC 2020

System load:  0.0          Processes:      92
Usage of /:   11.8% of 9.52GB  Users logged in:   0
Memory usage: 37%          IP address for ens4: 10.148.0.2
Swap usage:   0%

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@lab-computer-student-7:~$ ]
```

Introduction to terraform cloud

► Step 12: Test run docker command from script setup

```
[ubuntu@lab-computer-student-g7:~$ sudo docker run hello-world && exit
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:9572f7cdcee8591948c2963463447a53466950b3fc15a247fcad1917ca215a2f
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

logout
Connection to 34.87.80.107 closed.
ubuntu@ip-172-31-31-216:~$ ]
```

Introduction to terraform cloud

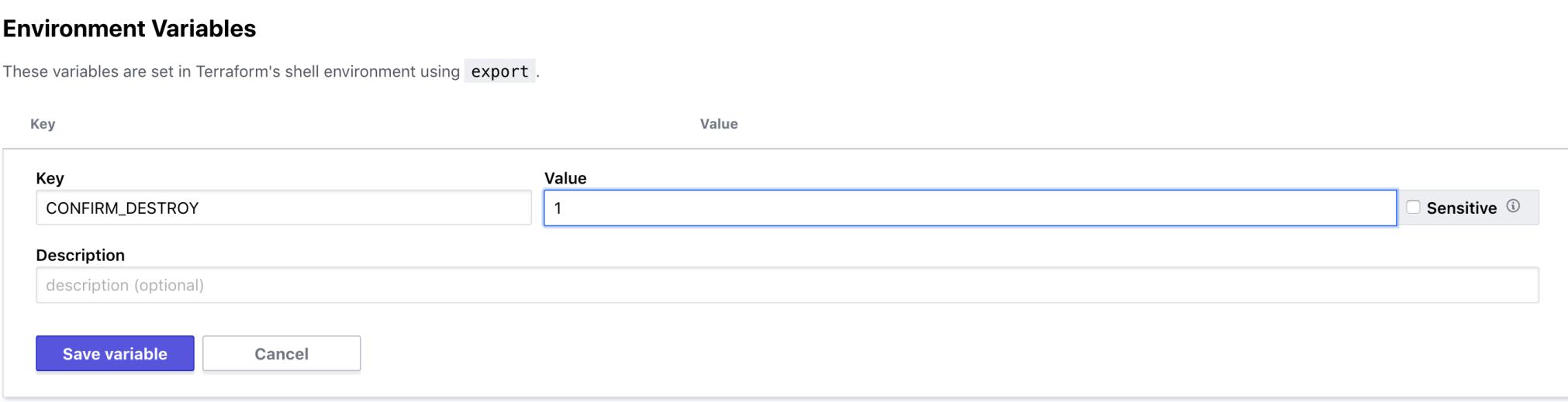
- ▶ Step 12: Set environment variable for “CONFIRM_DESTROY” to 1

Environment Variables

These variables are set in Terraform's shell environment using `export`.

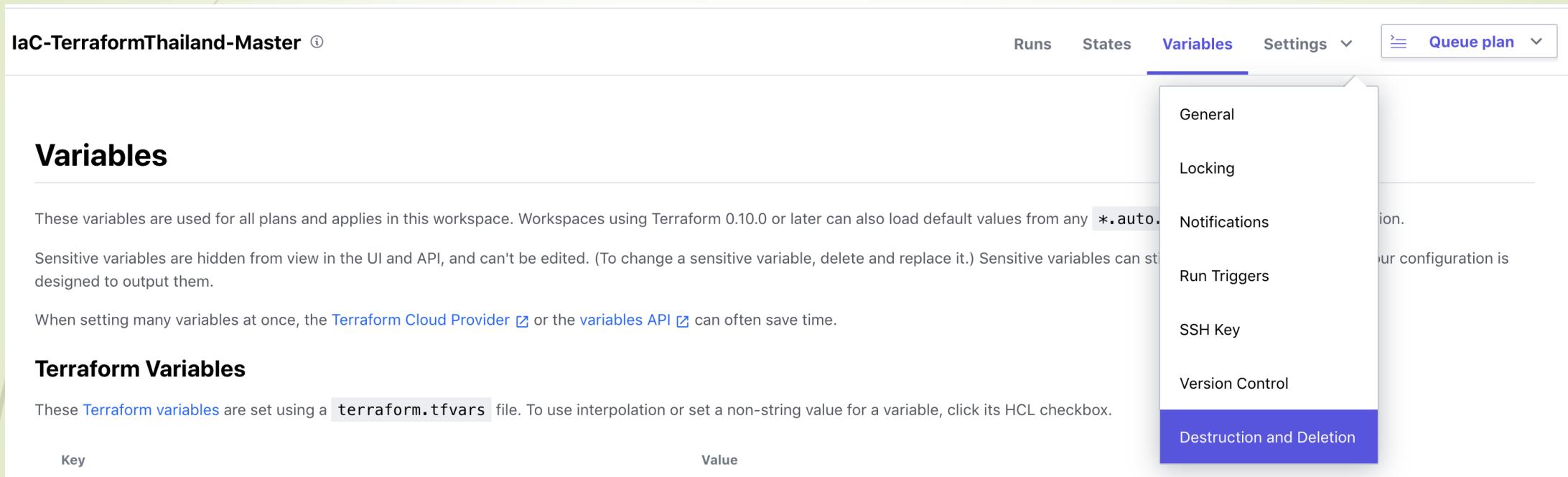
Key	Value	
Key CONFIRM_DESTROY	Value 1	<input type="checkbox"/> Sensitive ⓘ
Description description (optional)		

Save variable **Cancel**



Introduction to terraform cloud

► Step 13: Go to Settings → Destruction and Deletion



The screenshot shows the 'Variables' page of a workspace named 'laC-TerraformThailand-Master'. The 'Variables' tab is selected. A dropdown menu from the 'Settings' button contains several options: General, Locking, Notifications, Run Triggers, SSH Key, Version Control, and **Destruction and Deletion**, which is highlighted with a blue background.

Variables

These variables are used for all plans and applies in this workspace. Workspaces using Terraform 0.10.0 or later can also load default values from any `*.auto`.

Sensitive variables are hidden from view in the UI and API, and can't be edited. (To change a sensitive variable, delete and replace it.) Sensitive variables can be designed to output them.

When setting many variables at once, the [Terraform Cloud Provider](#) or the [variables API](#) can often save time.

Terraform Variables

These [Terraform variables](#) are set using a `terraform.tfvars` file. To use interpolation or set a non-string value for a variable, click its HCL checkbox.

Key	Value
-----	-------

Introduction to terraform cloud

► Step 14: Select “Queue destroy plan”

Destruction and Deletion

There are two independent steps for destroying this workspace and any infrastructure associated with it. First, any Terraform infrastructure should be destroyed. Second, the workspace in Terraform Cloud, including any variables, settings, and alert history can be deleted.

Queueing a destroy Plan will redirect to a new Plan that will destroy all of the infrastructure managed by Terraform. It is equivalent to running `terraform plan --destroy -out=destroy.tfplan` followed by `terraform apply destroy.tfplan` locally.

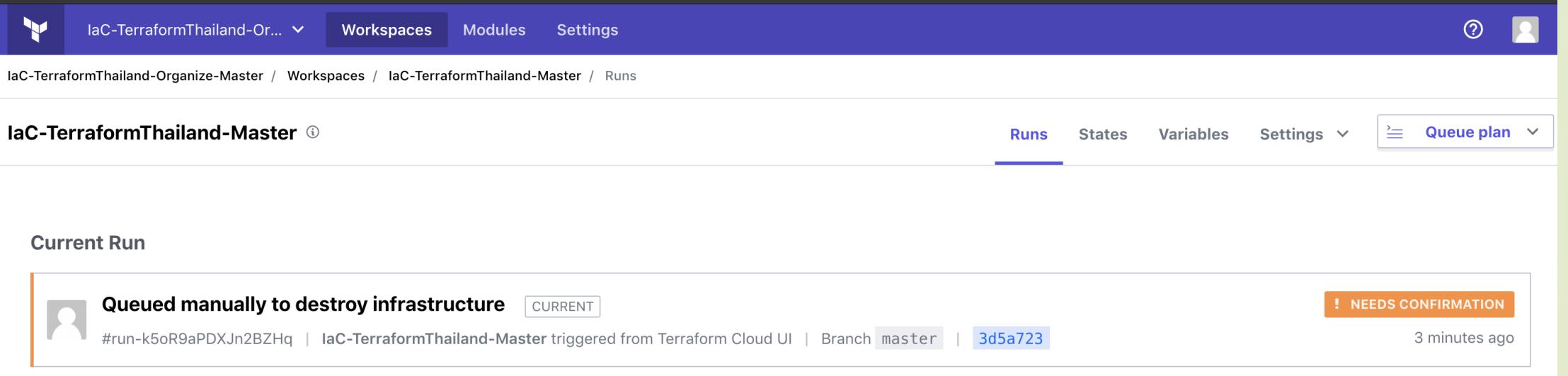
Queueing a destroy Plan will be disabled until there is an environment variable set named `CONFIRM_DESTROY` with a value of `1`. You can use the variables page to set it.

[Queue destroy plan](#)

[Delete from Terraform Cloud](#)

Introduction to terraform cloud

► Step 15: Back to Tab “Run”



The screenshot shows the Terraform Cloud web interface. At the top, there's a navigation bar with a logo, the workspace name "IaC-TerraformThailand-Or...", a dropdown menu, and tabs for "Workspaces", "Modules", and "Settings". On the far right are a help icon and a user profile icon. Below the navigation, the breadcrumb path shows "IaC-TerraformThailand-Organize-Master / Workspaces / IaC-TerraformThailand-Master / Runs". The main content area is titled "IaC-TerraformThailand-Master" with a "Runs" tab selected. Other tabs include "States", "Variables", and "Settings". To the right of the tabs is a "Queue plan" button. The main section is titled "Current Run" and displays a single run card. The run card has a user icon, the status "Queued manually to destroy infrastructure", a "CURRENT" badge, and an orange "NEEDS CONFIRMATION" button. Below the status are the run ID "#run-k5oR9aPDXJn2BZHq", the workspace name "IaC-TerraformThailand-Master triggered from Terraform Cloud UI", the branch "master", the commit hash "3d5a723", and the timestamp "3 minutes ago".

Introduction to terraform cloud

► Step 16: Verify terraform and click “Confirm & Apply”

! NEEDS CONFIRMATION Queued manually to destroy infrastructure

IaC-TerraformThailand-Master triggered a run from Terraform Cloud UI 4 minutes ago

Plan finished 4 minutes ago

Started 4 minutes ago > Finished 3 minutes ago

Download Sentinel mocks

View raw log

Resources: 0 to add, 0 to change, 3 to destroy

Sentinel mocks can be used for testing your Sentinel policies

Top Bottom Expand Full screen

```
- on_host_maintenance = "MIGRATE" -> null
- preemptible      = false -> null
}
}

# google_compute_network.lab_vpc_network will be destroyed
resource "google_compute_network" "lab_vpc_network" {
- auto_create_subnetworks      = true -> null
- delete_default_routes_on_create = false -> null
- id                           = "projects/polar-decorator-267303/global/networks/lab-vpc-student-7" -> null
- name                          = "lab-vpc-student-7" -> null
- project                      = "polar-decorator-267303" -> null
- routing_mode                 = "REGIONAL" -> null
- self_link                     = "https://www.googleapis.com/compute/v1/projects/polar-decorator-267303/global/networks/lab-vpc-student-7" -> null
}

Plan: 0 to add, 0 to change, 3 to destroy.
```

Apply pending

Needs Confirmation: Check the plan and confirm to apply it, or discard the run.

Confirm & Apply Discard Run Add Comment



Introduction to terraform cloud

► Step 17: Monitor terraform progress

APPLYING Git Action Credential CURRENT

 **paparn** triggered a **run** from GitHub 3 minutes ago [Run Details](#)

 **Plan finished** 6 minutes ago Resources: **3** to add, **0** to change, **0** to destroy

 **Apply running** a few seconds ago

Started a few seconds ago

[View raw log](#) [Following log](#) [Top](#) [Bottom](#) [Expand](#) [Full screen](#)

```
Terraform v0.12.20
Initializing plugins and modules...
2020/02/11 13:58:05 [DEBUG] Using modified User-Agent: Terraform/0.12.20 TFC/664d7ed310
google_compute_network.lab_vpc_network: Creating...
google_compute_network.lab_vpc_network: Still creating... [10s elapsed]
google_compute_network.lab_vpc_network: Still creating... [20s elapsed]
```

Introduction to terraform cloud

► Step 17: Monitor terraform progress



```
Apply running a few seconds ago
Started a few seconds ago
View raw log Following log Top Bottom Expand Full screen
Terraform v0.12.20
Initializing plugins and modules...
2020/02/11 14:26:14 [DEBUG] Using modified User-Agent: Terraform/0.12.20 TFC/664d7ed310
google_compute_firewall.lab_firewall: Destroying... [id=projects/polar-decorator-267303/global/firewalls/allow-ssh]
google_compute_instance.labserver: Destroying... [id=projects/polar-decorator-267303/zones/asia-southeast1-a/instances/lab-computer-student-7]
google_compute_firewall.lab_firewall: Still destroying... [id=projects/polar-decorator-267303/global/firewalls/allow-ssh, 10s elapsed]
google_compute_instance.labserver: Still destroying... [id=projects/polar-decorator-267303/zones/asia-southeast1-a/instances/lab-computer-student-7, 10s elapsed]
google_compute_firewall.lab_firewall: Destruction complete after 17s
google_compute_instance.labserver: Still destroying... [id=projects/polar-decorator-267303/zones/asia-southeast1-a/instances/lab-computer-student-7, 20s elapsed]
```

Introduction to terraform cloud

► Step 17: Monitor terraform progress

Apply finished a few seconds ago

Started 7 minutes ago > Finished 4 minutes ago

Resources: 0 added, 0 changed, 3 destroyed

[View raw log](#) [Top](#) [Bottom](#) [Expand](#) [Full screen](#)

```
google_compute_instance.labserver: Still destroying... [id=projects/polar-decorator-267303/zones/a...st1-a/instances/lab-computer-student-7, 30s elapsed]
google_compute_instance.labserver: Still destroying... [id=projects/polar-decorator-267303/zones/a...st1-a/instances/lab-computer-student-7, 40s elapsed]
google_compute_instance.labserver: Still destroying... [id=projects/polar-decorator-267303/zones/a...st1-a/instances/lab-computer-student-7, 50s elapsed]
google_compute_instance.labserver: Still destroying... [id=projects/polar-decorator-267303/zones/a...st1-a/instances/lab-computer-student-7, 1m0s elapsed]
google_compute_instance.labserver: Still destroying... [id=projects/polar-decorator-267303/zones/a...st1-a/instances/lab-computer-student-7, 1m10s elapsed]
google_compute_instance.labserver: Still destroying... [id=projects/polar-decorator-267303/zones/a...st1-a/instances/lab-computer-student-7, 1m20s elapsed]
google_compute_instance.labserver: Still destroying... [id=projects/polar-decorator-267303/zones/a...st1-a/instances/lab-computer-student-7, 1m30s elapsed]
google_compute_instance.labserver: Still destroying... [id=projects/polar-decorator-267303/zones/a...st1-a/instances/lab-computer-student-7, 1m40s elapsed]
google_compute_instance.labserver: Still destroying... [id=projects/polar-decorator-267303/zones/a...st1-a/instances/lab-computer-student-7, 1m50s elapsed]
google_compute_instance.labserver: Destruction complete after 1m57s
google_compute_network.lab_vpc_network: Destroying... [id=projects/polar-decorator-267303/global/networks/lab-vpc-student-7]
google_compute_network.lab_vpc_network: Still destroying... [id=projects/polar-decorator-267303/global/networks/lab-vpc-student-7, 10s elapsed]
google_compute_network.lab_vpc_network: Still destroying... [id=projects/polar-decorator-267303/global/networks/lab-vpc-student-7, 20s elapsed]
google_compute_network.lab_vpc_network: Still destroying... [id=projects/polar-decorator-267303/global/networks/lab-vpc-student-7, 30s elapsed]
google_compute_network.lab_vpc_network: Destruction complete after 36s

Apply complete! Resources: 0 added, 0 changed, 3 destroyed.
```

State versions created:
laC-TerraformThailand-Organize-Master/laC-TerraformThailand-Master#sv-CvQ7nYSyM3emmR18 (Feb 11, 2020 21:28:48 pm)

Introduction to terraform cloud

► Step 18: Verify on Gcloud-CLI

```
[ubuntu@ip-172-31-31-216:~$ gcloud compute instances list --filter="name='lab-computer-student-g7'"  
Listed 0 items.  
ubuntu@ip-172-31-31-216:~$ ]
```

► Step 19: Destroy workspace via menu “Destruct&Destroy” and choose “Delete from Terraform Cloud”

Destruction and Deletion

There are two independent steps for destroying this workspace and any infrastructure associated with it. First, any Terraform infrastructure should be destroyed. Second, the workspace in Terraform Cloud, including any variables, settings, and alert history can be deleted.

Queueing a destroy Plan will redirect to a new Plan that will destroy all of the infrastructure managed by Terraform. It is equivalent to running `terraform plan -destroy -out=destroy.tfplan` followed by `terraform apply destroy.tfplan` locally.

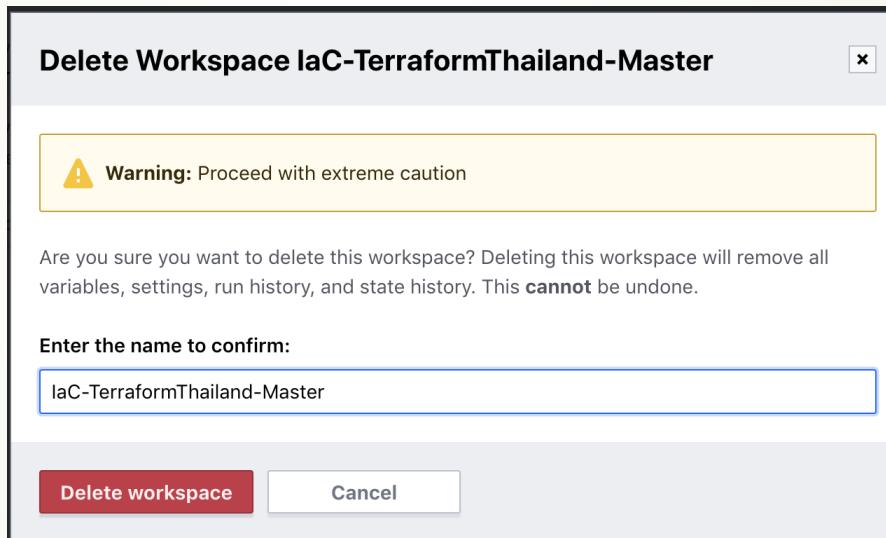
Queueing a destroy Plan will be disabled until there is an environment variable set named `CONFIRM_DESTROY` with a value of `1`. You can use the variables page to set it.

[Queue destroy plan](#)

[Delete from Terraform Cloud](#)

Introduction to terraform cloud

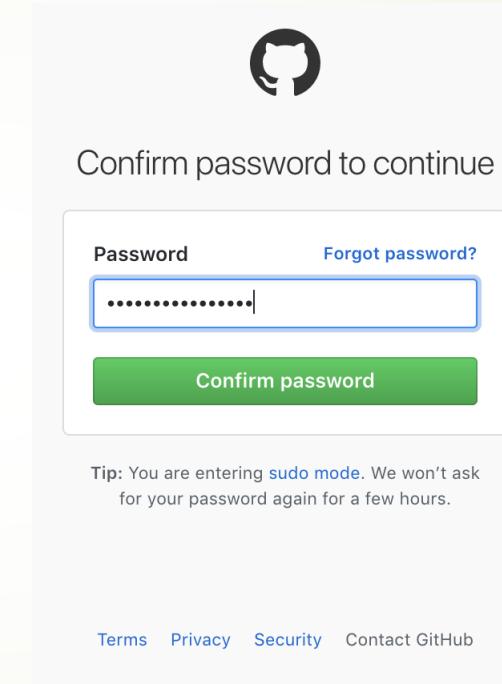
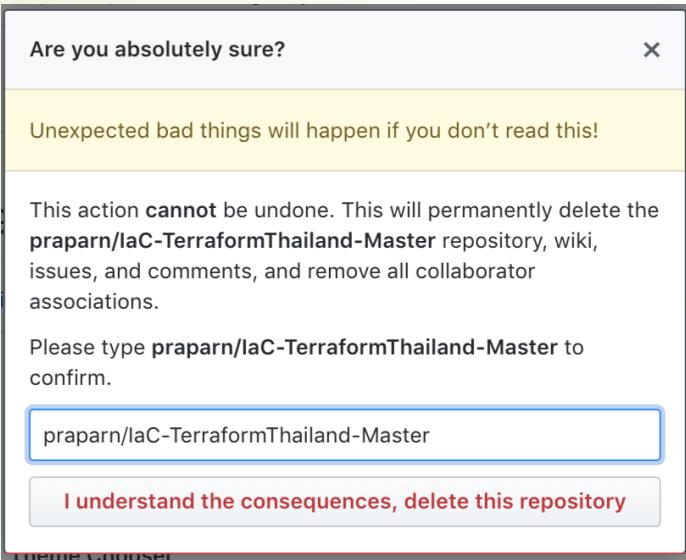
- ▶ Step 20: Destroy workspace via menu “Destruct&Destroy” and choose “Delete from Terraform Cloud”



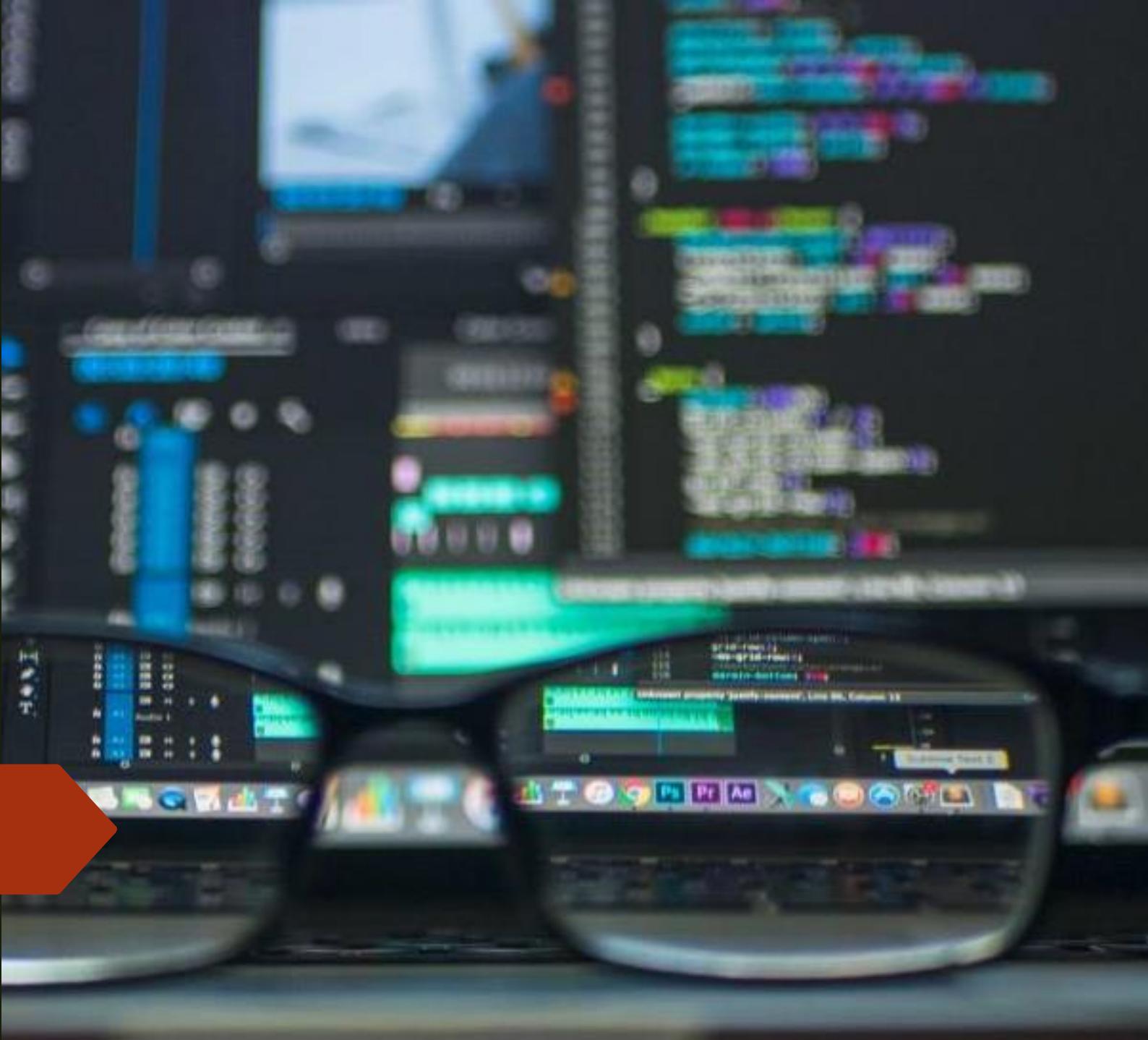
A screenshot of the Terraform Cloud interface showing the "Workspaces" page. The top navigation bar includes "Workspaces", "Modules", and "Settings". The main area displays a table with columns: WORKSPACE NAME, RUN STATUS, RUN, REPO, and LATEST CHANGE. A message at the bottom states "You don't have any workspaces yet. Create one now." A "New workspace" button is located in the top right of the workspace list area.

Introduction to terraform cloud

► Step 21: Delete github repository



Workshop: Terraform Cloud



Workshop: Terraform Cloud (Native)

```
[ubuntu@ip-172-31-31-216:~/terraform_202003/Workshop_2.3_TerraformCloud/NativeWorkSpace$ terraform init
Initializing the backend...
Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "google" (hashicorp/google) 3.8.0...
The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.google: version = "~> 3.8"

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.
ubuntu@ip-172-31-31-216:~/terraform_202003/Workshop_2.3_TerraformCloud/NativeWorkSpace$     terraform fmt -check
00_provider.tf
02_resource_instance.tf
terraform.tfvars
variables.tf
ubuntu@ip-172-31-31-216:~/terraform_202003/Workshop_2.3_TerraformCloud/NativeWorkSpace$     terraform validate
Success! The configuration is valid.

ubuntu@ip-172-31-31-216:~/terraform_202003/Workshop_2.3_TerraformCloud/NativeWorkSpace$     terraform apply -auto-approve -parallelism=5
google_compute_network.lab_vpc_network: Creating...
google_compute_network.lab_vpc_network: Still creating... [10s elapsed]
google_compute_network.lab_vpc_network: Still creating... [20s elapsed]
google_compute_network.lab_vpc_network: Still creating... [30s elapsed]
google_compute_network.lab_vpc_network: Still creating... [40s elapsed]
google_compute_network.lab_vpc_network: Creation complete after 48s [id=projects/polar-decorator-267303/global/networks/lab-vpc-student-g37]
google_compute_firewall.lab_firewall: Creating...
google_compute_instance.labserver: Creating...
google_compute_firewall.lab_firewall: Creation complete after 9s [id=projects/polar-decorator-267303/global/firewalls/allow-ssh]
google_compute_instance.labserver: Still creating... [10s elapsed]
google_compute_instance.labserver: Creation complete after 11s [id=projects/polar-decorator-267303/zones/asia-southeast1-a/instances/lab-computer-student-g37]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Outputs:
google_compute_instance_labserver_id = 7518454955287783674
google_compute_instance_labserver_nat_ip = 34.87.80.107
ubuntu@ip-172-31-31-216:~/terraform_202003/Workshop_2.3_TerraformCloud/NativeWorkSpace$ ]
```



Google Cloud Platform



Workshop: Terraform Cloud (Demo)

APPLIED **Git Action Credential**

praparn triggered a run from GitHub 3 minutes ago Run Details

Plan finished 7 minutes ago

Started 7 minutes ago > Finished 7 minutes ago

[Download Sentinel mocks](#)

[View raw log](#)

Resources: 3 to add, 0 to change, 0 to destroy

Sentinel mocks can be used for testing your Sentinel policies

Top Bottom Expand Full screen

```
}
}

# google_compute_network.lab_vpc_network will be created
+ resource "google_compute_network" "lab_vpc_network" {
    + auto_create_subnetworks      = true
    + delete_default_routes_on_create = false
    + gateway_ipv4                = (known after apply)
    + id                           = (known after apply)
    + ipv4_range                   = (known after apply)
    + name                         = "lab-vpc-student-7"
    + project                      = (known after apply)
    + routing_mode                 = (known after apply)
    + self_link                     = (known after apply)
}
```

Plan: 3 to add, 0 to change, 0 to destroy.

Apply finished a few seconds ago

Resources: 3 added, 0 changed, 0 destroyed



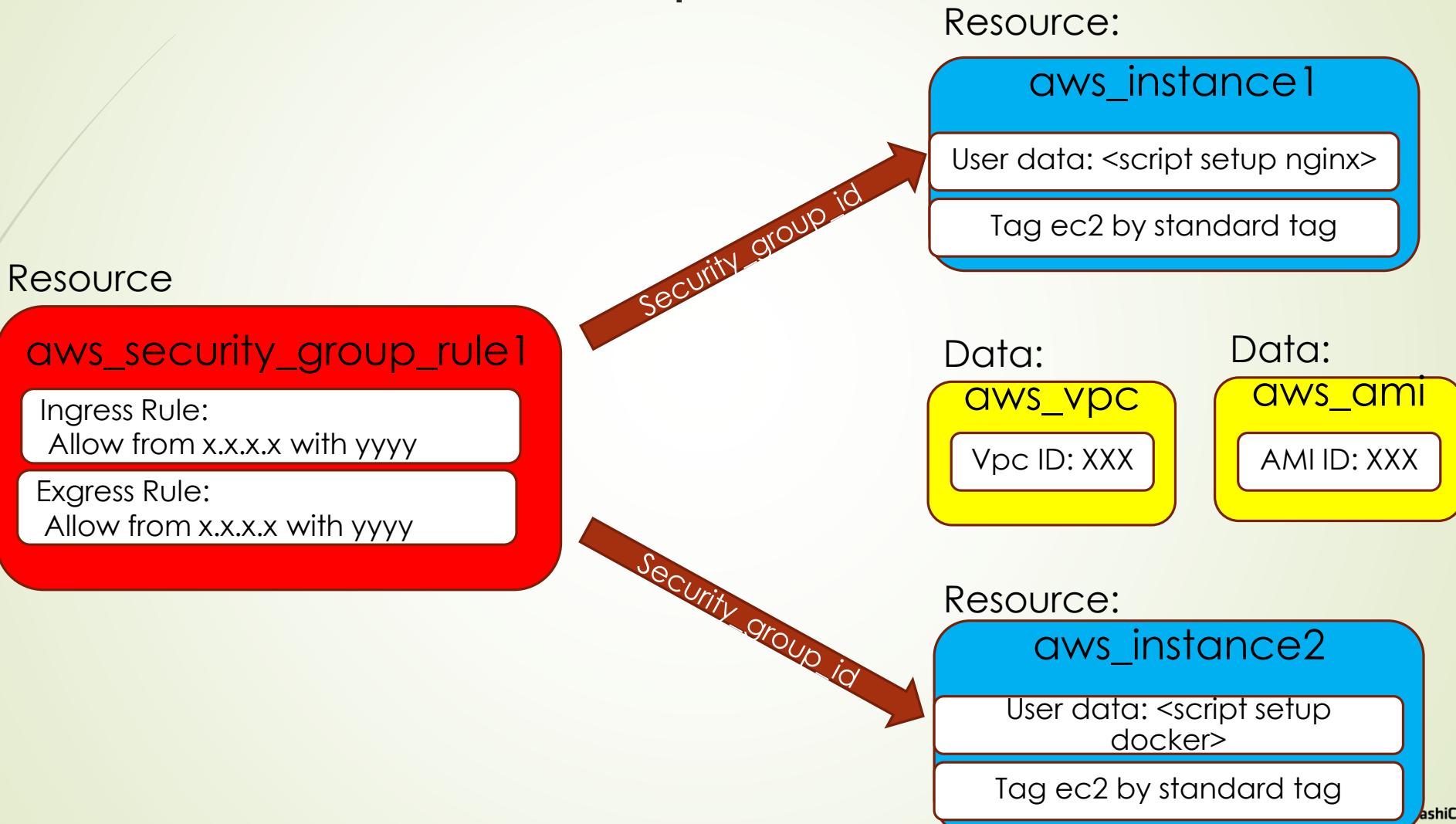
Q&A



Workshop: Team Workshop

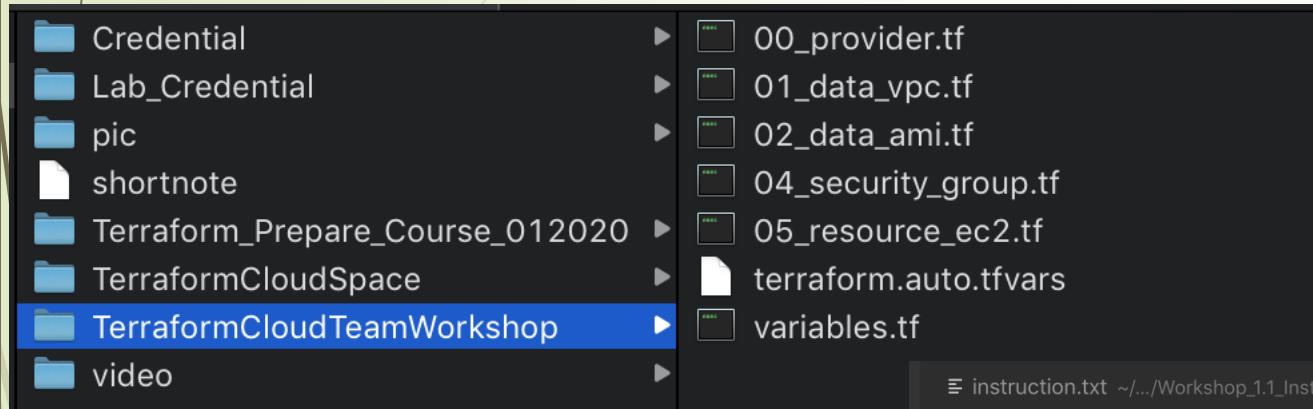


Team Workshop



Team WorkShop

► Source of terraform file



A code editor window showing the content of the 'terraform.auto.tfvars' file. The file contains configuration for AWS Lambda functions and their dependencies.

```
1 instance_type = {
2     "nginx" = "t3a.nano"
3     "docker" = "t2a.nano"
4     "other" = "t3a.nano"
5 }
6
7 user_data = {
8     "0" = <<-EOF
9         #!/bin/bash
10        curl https://raw.githubusercontent.com/praparn/sourcesetup/master/standard_nginx.sh > /tmp/setup.sh
11        chmod +x /tmp/setup.sh
12        /tmp/setup.sh
13        EOF
14     "1"= <<-EOF
15         #!/bin/bash
16         curl https://raw.githubusercontent.com/praparn/sourcesetup/master/standard_docker_aws.sh > /tmp/setup.sh
17         chmod +x /tmp/setup.sh
18         /tmp/setup.sh
19         EOF
20 }
```

Team WorkShop

► Step1: Create private repository on github and add terraform to git

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner **Repository name ***

paparn / IaC-TerraformThailand-User-13 ✓

Great repository names are short and memorable. Need inspiration? How about [special-funicular](#)?

Description (optional)

IaC-TerraformThailand-User-13

Public
Anyone can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: **None** ▾ Add a license: **None** ▾ ⓘ

Create repository

Search or jump to... Pull requests Issues Marketplace Explore

paparn / IaC-TerraformThailand-User-13 Private

Unwatch ▾ 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Security Insights Settings

Quick setup — if you've done this kind of thing before

Set up in Desktop or **HTTPS** **SSH** <https://github.com/paparn/IaC-TerraformThailand-User-13.git> ⌂

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# IaC-TerraformThailand-User-13" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/paparn/IaC-TerraformThailand-User-13.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/paparn/IaC-TerraformThailand-User-13.git
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

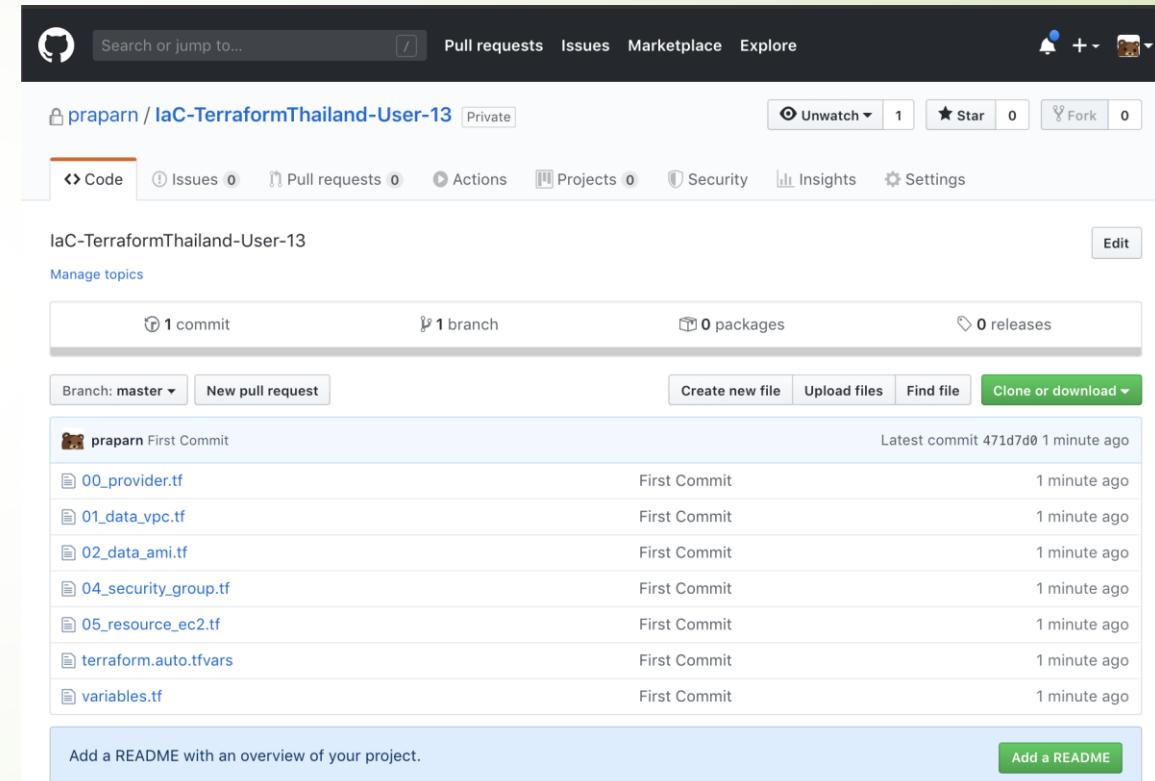
Import code

ProTip! Use the URL for this page when adding GitHub as a remote.

Team WorkShop

► Step1: Create private repository on github and add terraform to git

```
praparn-MBP:TerraformCloudTeamWorkshop praparn$ ls -l
total 56
-rwxr-xr-x 1 praparn staff 42 Jan 29 20:49 00_provider.tf
-rwxr-xr-x 1 praparn staff 49 Jan 29 22:39 01_data_vpc.tf
-rwxr-xr-x 1 praparn staff 423 Jan 19 15:20 02_data_ami.tf
-rwxr-xr-x 1 praparn staff 1115 Jan 27 21:39 04_security_group.tf
-rwxr-xr-x 1 praparn staff 1081 Jan 29 22:42 05_resource_ec2.tf
-rw-r--r-- 1 praparn staff 616 Feb 15 00:17 terraform.auto.tfvars
-rwxr-xr-x@ 1 praparn staff 1948 Feb 14 23:59 variables.tf
praparn-MBP:TerraformCloudTeamWorkshop praparn$ git init
Initialized empty Git repository in /Users/praparnlueangphoonaipaparn/Work/Terraform/Resource/TerraformCloudTeamWorkshop/.git/
praparn-MBP:TerraformCloudTeamWorkshop praparn$ git add -A
praparn-MBP:TerraformCloudTeamWorkshop praparn$ git commit -m "First Commit"
[master (root-commit) 471d7d0] First Commit
 7 files changed, 227 insertions(+)
 create mode 100755 00_provider.tf
 create mode 100755 01_data_vpc.tf
 create mode 100755 02_data_ami.tf
 create mode 100755 04_security_group.tf
 create mode 100755 05_resource_ec2.tf
 create mode 100644 terraform.auto.tfvars
 create mode 100755 variables.tf
praparn-MBP:TerraformCloudTeamWorkshop praparn$ git remote add origin https://github.com/praparn/IaC-TerraformThailand-User-13.git
praparn-MBP:TerraformCloudTeamWorkshop praparn$ git push -u origin master
Username for 'https://github.com/praparn/IaC-TerraformThailand-User-13.git': eva10409@gmail.com
Password for 'https://eva10409@gmail.com@github.com/praparn/IaC-TerraformThailand-User-13.git':
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (9/9), 2.23 KiB | 2.23 MiB/s, done.
Total 9 (delta 0), reused 0 (delta 0)
To https://github.com/praparn/IaC-TerraformThailand-User-13.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
praparn-MBP:TerraformCloudTeamWorkshop praparn$
```



Team WorkShop

- ▶ Step2: Access terraform cloud (<https://app.terraform.io/signup/account>) and signup

The screenshot shows the 'Create an account' page for Terraform Cloud. At the top, there's a dark banner with the HashiCorp logo and the text 'Terraform Cloud'. Below it, a message says 'You're minutes away from collaborating on infrastructure with your team.' The main form has fields for 'Username', 'Email', and 'Password'. There are two checkboxes at the bottom: 'I agree to the Terms of Use.' and 'I acknowledge the Privacy Policy.'. A note at the bottom says 'Please review the [Terms of Use](#) and [Privacy Policy](#)'. A blue 'Create account' button is at the bottom.

The screenshot shows the 'Sign in to Terraform Cloud' page. It features the HashiCorp logo and 'Terraform Cloud' text. It includes fields for 'Username or email' (containing 'laC-TerraformThailand-Master') and 'Password'. A 'Forgot password?' link and a blue 'Sign in' button are present. Below the form, text encourages users to sign up with a 'Create your free account' link and mentions 'Terraform Offerings'.

Team WorkShop

- ▶ Step3: Create organization (Reference from control sheet)

The screenshot shows the 'New Organization' creation interface. At the top, there's a blue header bar with the HashiCorp logo and a three-line menu icon. Below the header, the URL 'Organizations / New' is visible. The main section is titled 'New Organization'. It contains two input fields: 'Organization name' with the value 'IaC-TerraformThailand-Organize-13' and 'Email address' with the value 'eva10409@gmail.com'. A note below the email field states: 'The organization email is used for any future notifications, such as billing, and the organization avatar, via [gravatar.com](#)'. At the bottom is a large blue 'Create organization' button.

Organizations / New

New Organization

Organization name

Organization names must be unique and will be part of your resource names used in various tools, i.e `IaC-TerraformThailand-Organize-13/www-prod`.

Email address

The organization email is used for any future notifications, such as billing, and the organization avatar, via [gravatar.com](#).

Create organization

© 2020 HashiCorp, Inc. [Support](#) [Terms](#) [Privacy](#) [Security](#)

Team WorkShop

- ▶ Step4: Create workspace and connect vcs (version control system) via github's repository that we just create

IaC-TerraformThailand-Or...

Create a new Workspace

Workspaces allow you to organize infrastructure and collaborate on Terraform runs.

1 Connect to VCS 2 Choose a repository 3 Configure settings

Connect to a version control provider

GitHub

Connect to a different VCS

If you only plan to use Terraform CLI or the API to perform runs in this workspace, you don't need to connect it to version control. You can add a VCS connection later if you change your mind.

No VCS connection
Requires CLI or API

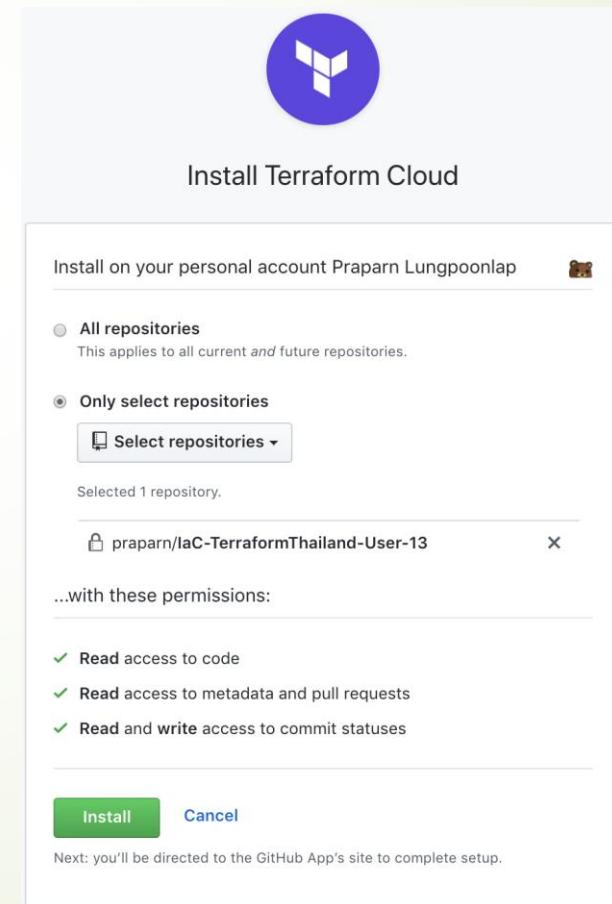
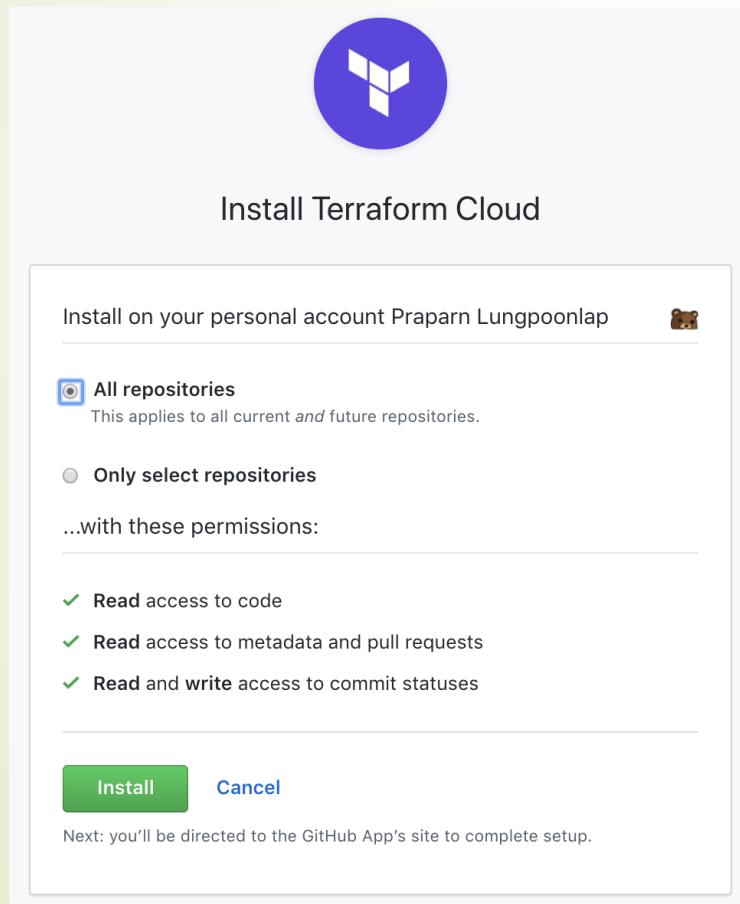
Install Terraform Cloud

Where do you want to install Terraform Cloud?

praparn

Team WorkShop

- ▶ Step4: Create workspace and connect vcs (version control system) via github's repository that we just create



Team WorkShop

- ▶ Step4: Create workspace and connect vcs (version control system) via github's repository that we just create

Create a new Workspace

Workspaces allow you to organize infrastructure and collaborate on Terraform runs.

1 Connect to VCS 2 Choose a repository 3 Configure settings

Choose a repository

Choose the repository that hosts your Terraform source code. We'll watch this for commits and pull requests.

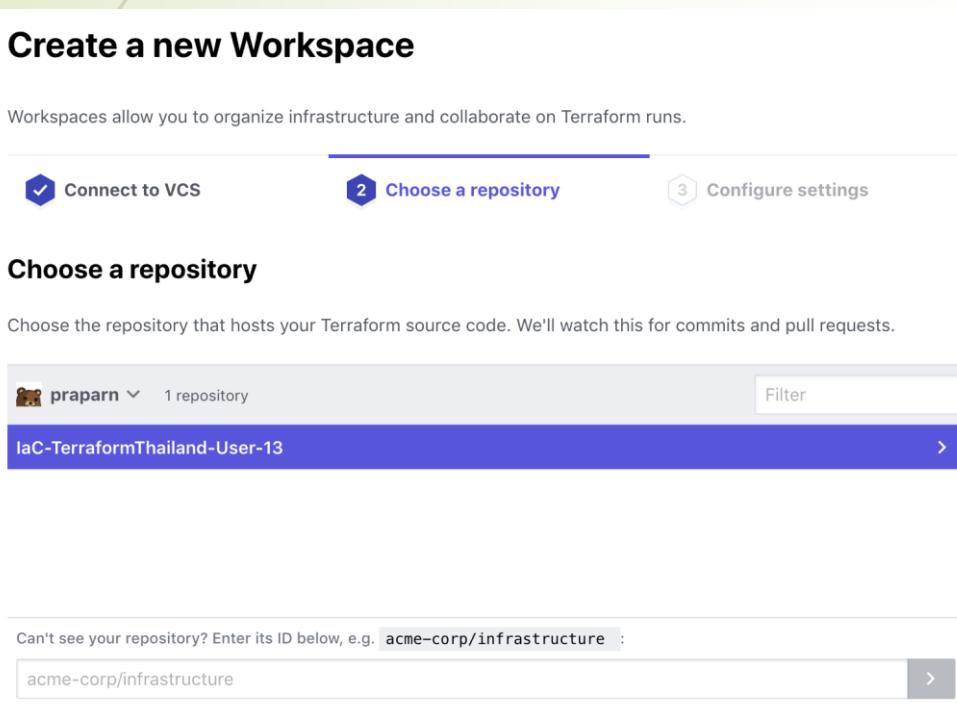
praparn ▾ 1 repository

Filter

laC-TerraformThailand-User-13 >

Can't see your repository? Enter its ID below, e.g. acme-corp/infrastructure :

acme-corp/infrastructure >



Create a new Workspace

Workspaces allow you to organize infrastructure and collaborate on Terraform runs.

1 Connect to VCS 2 Choose a repository 3 Configure settings

Configure settings

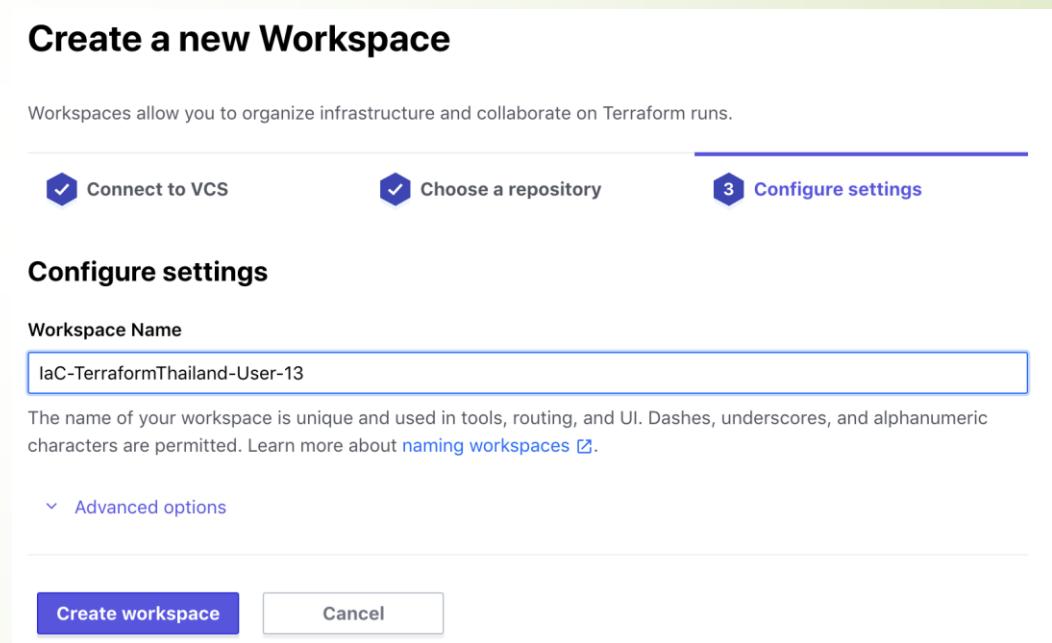
Workspace Name

laC-TerraformThailand-User-13

The name of your workspace is unique and used in tools, routing, and UI. Dashes, underscores, and alphanumeric characters are permitted. Learn more about [naming workspaces](#).

Advanced options

Create workspace Cancel



Team WorkShop

- ▶ Step5: Input variable for operate with terraform (Reference on Control Sheet)

 Configuration uploaded successfully

Your configuration has been uploaded. Next, you probably want to configure variables (such as access keys or configuration values). If your configuration doesn't require variables, you can queue your first plan now.

[Configure variables](#) [Queue plan](#)

laC-TerraformThailand-User-13 ⓘ

Runs States [Variables](#) Settings Queue plan

Variables

These variables are used for all plans and applies in this workspace. Workspaces using Terraform 0.10.0 or later can also load default values from any `*.auto.tfvars` files in the configuration.

Sensitive variables are hidden from view in the UI and API, and can't be edited. (To change a sensitive variable, delete and replace it.) Sensitive variables can still appear in Terraform logs if your configuration is designed to output them.

When setting many variables at once, the [Terraform Cloud Provider](#) or the [variables API](#) can often save time.

Terraform Variables

These [Terraform variables](#) are set using a `terraform.tfvars` file. To use interpolation or set a non-string value for a variable, click its HCL checkbox.

Key	Value
There are no variables set.	

[+ Add variable](#)

Environment Variables

These variables are set in Terraform's shell environment using `export`.

Key	Value
There are no variables set.	

[+ Add variable](#)

Team WorkShop

- ▶ Step5: Input variable for operate with terraform (Reference on Control Sheet)

Terraform Variables

These [Terraform variables](#) are set using a `terraform.tfvars` file. To use interpolation or set a non-string value for a variable, click its HCL checkbox.

Key	Value
Key	<input type="text" value="vpc_id"/>
Value	<input type="text" value="vpc-4483e921"/> <input type="checkbox"/> HCL ⓘ <input type="checkbox"/> Sensitive ⓘ
Description	<input type="text" value="description (optional)"/>
<button>Save variable</button> <button>Cancel</button>	

Terraform Variables

These [Terraform variables](#) are set using a `terraform.tfvars` file. To use interpolation or set a non-string value for a variable, click its HCL checkbox.

Key	Value	...
vpc_id	vpc-4483e921	...

Team WorkShop

- ▶ Step5: Input variable for operate with terraform (Reference on Control Sheet)

Terraform Variables

These Terraform variables are set using a `terraform.tfvars` file. To use interpolation or set a non-string value for a variable, click its HCL checkbox.

Key	Value	...
vpc_id	vpc-4483e921	...
region	ap-southeast-1	...
availability_zone	ap-southeast-1a	...
server_type	nginx	...
keypair	keypair	...
instancecount	2	...
rootblockdevice_volume_type	gp2	...
rootblockdevice_volume_size	10	...
rootblockdevice_delete_on_termination	true	...
tag_environment	rd	...
tag_region	ap-southeast-1	...

tag_az	ap-southeast-1a	...
tag_category	compute	...
tag_name <small>HCL</small>	{ "0" = "labresource-server1-student-13" "1" = "labresource-server2-student-13" }	...
tag_zone	public	...
tag_module	ec2	...
tag_billing	terraform-workshop	...

[+ Add variable](#)

Team WorkShop

- ▶ Step6: Setup environment variable for access key and secret key
(Reference from control sheet)

Environment Variables

These variables are set in Terraform's shell environment using `export`.

Key	Value	...
AWS_ACCESS_KEY_ID	AKIAV4FGGXKRDU62DMNS	...
AWS_SECRET_ACCESS_KEY	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	...

[+ Add variable](#)

Team WorkShop

- ▶ Step7: Go to tab “run” and submit “queue plan” for start planning process

✓ Configuration uploaded successfully

Your configuration has been uploaded. Next, you probably want to configure variables (such as access keys or configuration values). If your configuration doesn't require variables, you can queue your first plan now.

The screenshot shows the Terraform Cloud UI interface. On the left, there are two buttons: "Configure variables" (blue) and "Queue plan" (white). In the center, the title "Queued from Terraform Cloud UI" is displayed above three status boxes. The top box shows a user icon and the text "IaC-TerraformThailand-Master triggered a run from Terraform Cloud UI a few seconds ago" with a "Run Details" dropdown arrow. The middle box shows a circular progress icon and the text "Plan queued". The bottom box shows a circular progress icon and the text "Apply pending". At the bottom of the screen, a message states "Cancelable: You can cancel this run to stop it from executing." with "Cancel Run" and "Add Comment" buttons.

Configure variables Queue plan

PLANNING Queued from Terraform Cloud UI CURRENT

IaC-TerraformThailand-Master triggered a run from Terraform Cloud UI a few seconds ago Run Details

Plan queued

Apply pending

Cancelable: You can cancel this run to stop it from executing.

Cancel Run Add Comment

Team WorkShop

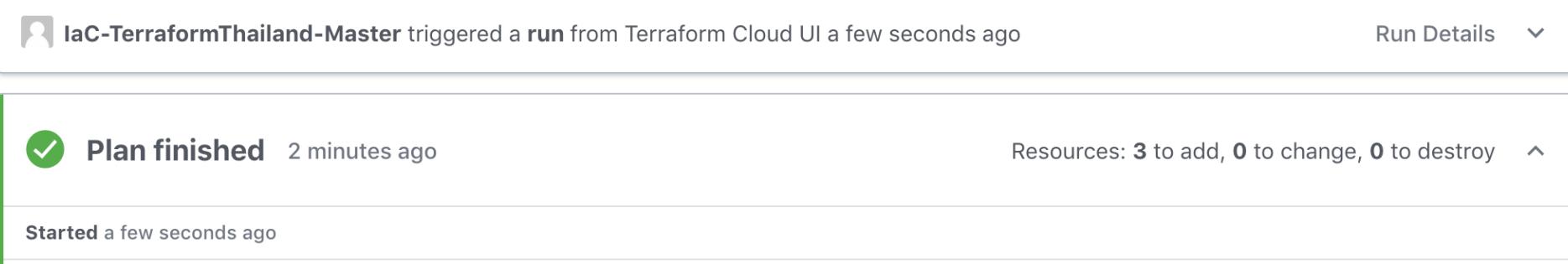
- ▶ Step8: Monitor progress for terraform plan and review item to create

! NEEDS CONFIRMATION **Queued from Terraform Cloud UI** **CURRENT**

 **IaC-TerraformThailand-Master** triggered a **run** from Terraform Cloud UI a few seconds ago [Run Details](#)

 **Plan finished** 2 minutes ago Resources: **3** to add, **0** to change, **0** to destroy [^](#)

Started a few seconds ago



Team WorkShop

► Step8: Monitor progress for terraform plan and review item to create

```
+ name                  = "securitygroup-server1-student-X"
+ owner_id              = (known after apply)
+ revoke_rules_on_delete = false
+ tags                  = {
    + "AZ"        = "none"
    + "Billing"   = "terraform-workshop"
    + "Categories" = "compute"
    + "Environment" = "rd"
    + "Module"    = "secgroup"
    + "Name"      = "securitygroup-server1-student-X"
    + "Region"    = "ap-southeast-1"
    + "Zone"      = "public"
}
+ vpc_id                = "vpc-4483e921"
```

Plan: 3 to add, 0 to change, 0 to destroy.

⌚ Apply pending

Needs Confirmation: Check the plan and confirm to apply it, or discard the run.

Confirm & Apply

Discard Run

Add Comment

Team WorkShop

- ▶ Step9: Choose “Confirm & Apply” with commit “Git Action Credential”

The screenshot shows the HashiCorp Terraform interface. At the top, there is a terminal-like window displaying a Terraform plan:

```
+ name = "securitygroup-server1-student-X"
+ owner_id = (known after apply)
+ revoke_rules_on_delete = false
+ tags = {
    + "AZ" = "none"
    + "Billing" = "terraform-workshop"
    + "Categories" = "compute"
    + "Environment" = "rd"
    + "Module" = "secgroup"
    + "Name" = "securitygroup-server1-student-X"
    + "Region" = "ap-southeast-1"
    + "Zone" = "public"
}
+ vpc_id = "vpc-4483e921"
```

Below the plan, a message says: **Plan: 3 to add, 0 to change, 0 to destroy.**

Below the terminal, a modal dialog is open with the title **⌚ Apply pending**. It contains a text input field labeled **Confirm & Apply: Add a comment to explain this action.** A user icon next to the input field shows the text **Git Action Credential**. At the bottom of the dialog are two buttons: **Confirm Plan** and **Close**.

Team WorkShop

- ▶ Step9: Choose “Confirm & Apply” with commit “Git Action Credential”

The screenshot shows the HashiCorp Terraform interface. At the top, there is a terminal-like window displaying a Terraform plan:

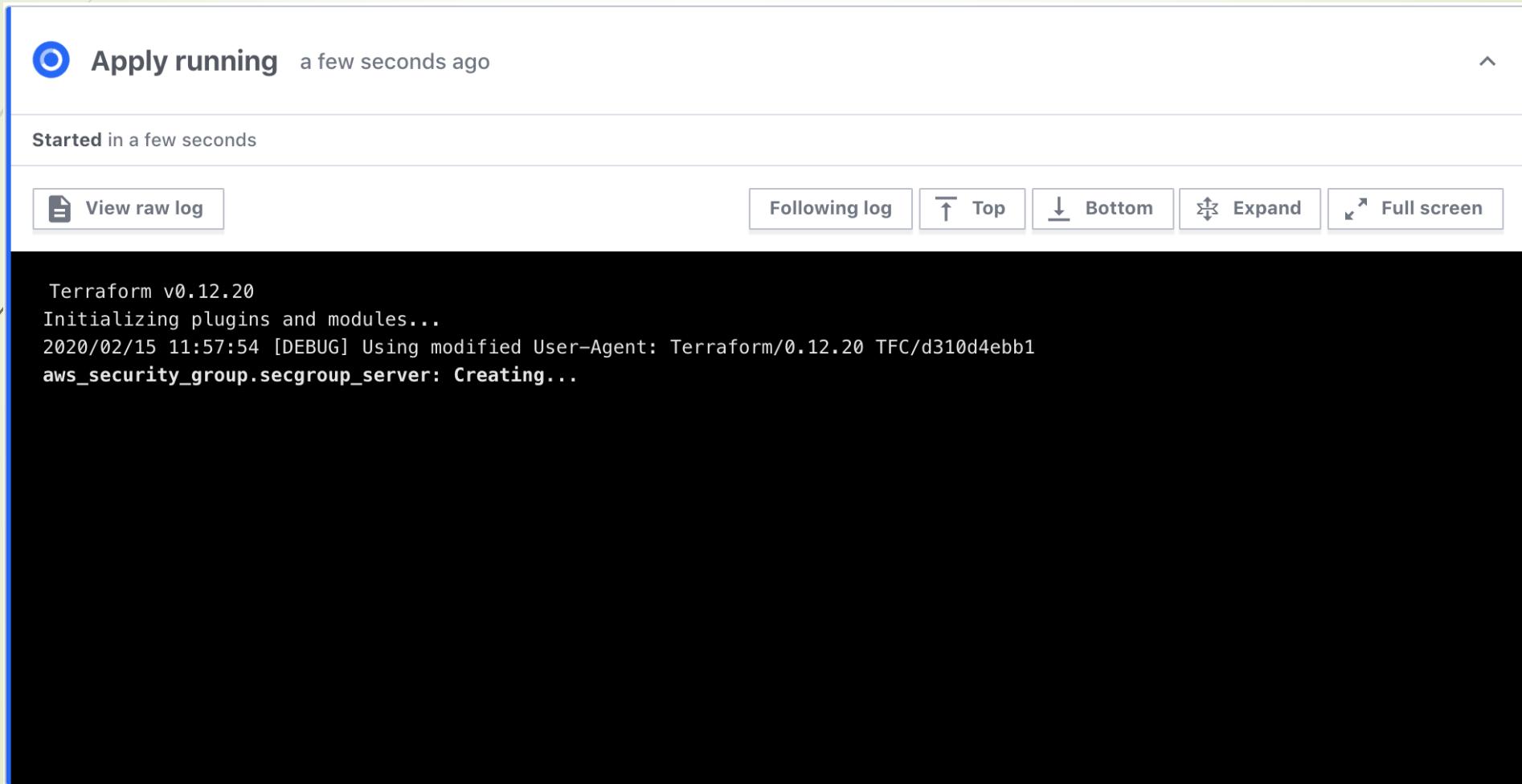
```
+ name = "securitygroup-server1-student-X"
+ owner_id = (known after apply)
+ revoke_rules_on_delete = false
+ tags = {
    + "AZ" = "none"
    + "Billing" = "terraform-workshop"
    + "Categories" = "compute"
    + "Environment" = "rd"
    + "Module" = "secgroup"
    + "Name" = "securitygroup-server1-student-X"
    + "Region" = "ap-southeast-1"
    + "Zone" = "public"
}
+ vpc_id = "vpc-4483e921"
```

Below the plan, a message says: **Plan: 3 to add, 0 to change, 0 to destroy.**

Below the terminal, a modal dialog is open with the title **⌚ Apply pending**. It contains a text input field labeled **Confirm & Apply: Add a comment to explain this action.** A user icon next to the input field shows the text **Git Action Credential**. At the bottom of the dialog are two buttons: **Confirm Plan** and **Close**.

Team WorkShop

► Step10: Monitor result from terraform cloud



raform

Team WorkShop

► Step10: Monitor result from terraform cloud

 **Apply finished** a few seconds ago Resources: 3 added, 0 changed, 0 destroyed ^

Started a minute ago > Finished a few seconds ago

[View raw log](#) Top Bottom Expand Full screen

```
aws_instance.labserver[1]: Creating...
aws_instance.labserver[0]: Creating...
aws_instance.labserver[1]: Still creating... [10s elapsed]
aws_instance.labserver[0]: Still creating... [10s elapsed]
aws_instance.labserver[1]: Still creating... [20s elapsed]
aws_instance.labserver[0]: Still creating... [20s elapsed]
aws_instance.labserver[1]: Creation complete after 21s [id=i-07d42934b3fabfdb9]
aws_instance.labserver[0]: Creation complete after 21s [id=i-003702e0d19772b95]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Outputs:

aws_instance_labserver_ip = {
  "i-003702e0d19772b95" = "54.255.194.134"
  "i-07d42934b3fabfdb9" = "54.254.138.29"
}
```

State versions created:
[laC-TerraformThailand-Organize-13/laC-TerraformThailand-User-13#sv-QxaXBxMpsNiVtGbp](#) (Feb 15, 2020 18:58:25 pm)

Team WorkShop

► Step11: Verify ec2 status by aws (Both Server)

```
ubuntu@ip-172-31-23-231:~$ aws ec2 describe-instances --instance-ids i-003702e0d19772b95
{
    "DescribeInstances": {
        "Reservations": [
            {
                "OwnerId": "404075494050",
                "ReservationId": "r-09c55d3fe73a4306b"
            }
        ],
        "Instances": [
            {
                "AmiLaunchIndex": 0,
                "Architecture": "x86_64",
                "ClientToken": null,
                "EbsOptimized": false,
                "EnaSupport": true,
                "Hypervisor": "xen",
                "ImageId": "ami-81cefccfd",
                "InstanceId": "i-003702e0d19772b95",
                "InstanceType": "t3a.nano",
                "KeyName": "keypair",
                "LaunchTime": "2020-02-15T11:58:07.000Z",
                "PrivateDnsName": "ip-172-31-19-90.ap-southeast-1.compute.internal",
                "PrivateIpAddress": "172.31.19.90",
                "PublicDnsName": "ec2-54-255-194-134.ap-southeast-1.compute.amazonaws.com",
                "PublicIpAddress": "54.255.194.134",
                "RootDeviceName": "/dev/sda1",
                "RootDeviceType": "ebs",
                "SourceDestCheck": true,
                "StateTransitionReason": null,
                "SubnetId": "subnet-41742436",
                "VirtualizationType": "hvm",
                "VpcId": "vpc-4483e921"
            }
        ],
        "BlockDeviceMappings": [
            {
                "DeviceName": "/dev/sda1"
            }
        ],
        "Ebs": [
            {
                "AttachTime": "2020-02-15T11:58:08.000Z",
                "DeleteOnTermination": true,
                "Status": "attached",
                "VolumeId": "vol-06de50479c5e0c311"
            }
        ],
        "CapacityReservationSpecification": [
            {
                "CapacityReservationPreference": "open"
            }
        ]
    }
}
```

CpuOptions	
CoreCount	1
ThreadsPerCore	2
HibernationOptions	
Configured	False
MetadataOptions	
HttpEndpoint	enabled
HttpPutResponseHopLimit	1
HttpTokens	optional
State	applied
Monitoring	
State	disabled
NetworkInterfaces	
Description	interface
InterfaceType	eni
MacAddress	06:c2:5e:03:31:02
NetworkInterfaceId	eni-015f39d48df393d20
OwnerId	404075494050
PrivateDnsName	ip-172-31-19-90.ap-southeast-1.compute.internal
PrivateIpAddress	172.31.19.90
SourceDestCheck	true
Status	in-use
SubnetId	subnet-41742436
VpcId	vpc-4483e921
Association	
IpOwnerId	amazon
PublicDnsName	ec2-54-255-194-134.ap-southeast-1.compute.amazonaws.com
PublicIp	54.255.194.134

Attachment	
AttachTime	2020-02-15T11:58:07.000Z
AttachmentId	eni-attach-0e20f2998f99dbe1e
DeleteOnTermination	true
DeviceIndex	0
Status	attached
Groups	
GroupId	sg-05ae8d4816c74a111
GroupName	securitygroup-server1-student-X
PrivateIpAddresses	
Primary	true
PrivateDnsName	ip-172-31-19-90.ap-southeast-1.compute.internal
PrivateIpAddress	172.31.19.90
Association	
IpOwnerId	amazon
PublicDnsName	ec2-54-255-194-134.ap-southeast-1.compute.amazonaws.com
PublicIp	54.255.194.134
Placement	
AvailabilityZone	ap-southeast-1a
GroupName	
Tenancy	default
SecurityGroups	
GroupId	sg-05ae8d4816c74a111
GroupName	securitygroup-server1-student-X
State	
Code	16
Name	running
Tags	
Key	Value
Categories	compute
Module	ec2
Name	labresource-server1-student-13
Region	ap-southeast-1
Billing	terraform-workshop
Environment	rd
AZ	ap-southeast-1a
Zone	public

Team WorkShop

► Step12: SSH to target server and test (Both Server)

```
[ubuntu@ip-172-31-23-231:~$ ssh -i ~/terraformlab ubuntu@54.255.194.134
The authenticity of host '54.255.194.134 (54.255.194.134)' can't be established.
ECDSA key fingerprint is SHA256:FRuzmNK5n7V8G8A14Y+khltEhs8MCNl73g0WVJqQw.
[Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '54.255.194.134' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-1060-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 Get cloud support with Ubuntu Advantage Cloud Guest:
   http://www.ubuntu.com/business/services/cloud

204 packages can be updated.
133 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

WARNING! Your environment specifies an invalid locale.
The unknown environment variables are:
  LC_CTYPE=UTF-8 LC_ALL=
This can affect your user experience significantly, including the
ability to manage packages. You may install the locales by running:

  sudo apt-get install language-pack-UTF-8
  or
  sudo locale-gen UTF-8

To see all available language packs, run:
  apt-cache search "^language-pack-[a-z][a-z]$"
To disable this message for all users, run:
  sudo touch /var/lib/cloud/instance/locale-check.skip
```

```
[ubuntu@ip-172-31-19-90:~$ ps -ef | grep nginx && curl http://localhost
root      6655      1  0 18:59 ?          00:00:00 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
www-data  6656  6655  0 18:59 ?          00:00:00 nginx: worker process
www-data  6657  6655  0 18:59 ?          00:00:00 nginx: worker process
ubuntu    6840  6814  0 19:17 pts/0        00:00:00 grep --color=auto nginx
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
```

Team WorkShop

► Step12: SSH to target server and test (Both Server)

```
[ubuntu@ip-172-31-23-231:~$ ssh -i ~/terraformlab ubuntu@54.254.138.29
The authenticity of host '54.254.138.29 (54.254.138.29)' can't be established.
ECDSA key fingerprint is SHA256:oxjbYpBKVa3rvySg5qAMN8omoCEbYo3008+k/kAjnkc.
[Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '54.254.138.29' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-1060-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
 http://www.ubuntu.com/business/services/cloud

197 packages can be updated.
129 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

-----
WARNING! Your environment specifies an invalid locale.
The unknown environment variables are:
LC_CTYPE=UTF-8 LC_ALL=
This can affect your user experience significantly, including the
ability to manage packages. You may install the locales by running:

  sudo apt-get install language-pack-UTF-8
  or
  sudo locale-gen UTF-8

To see all available language packs, run:
  apt-cache search "^language-pack-[a-z][a-z]$"
To disable this message for all users, run:
  sudo touch /var/lib/cloud/instance/locale-check.skip
-----
```

```
ubuntu@ip-172-31-31-241:~$ docker version && docker run hello-world && exit
Client: Docker Engine - Community
 Version:          19.03.6
 API version:      1.40
 Go version:       go1.12.16
 Git commit:       369ce74a3c
 Built:            Thu Feb 13 01:28:06 2020
 OS/Arch:          linux/amd64
 Experimental:    false

Server: Docker Engine - Community
 Engine:
  Version:          19.03.6
  API version:      1.40 (minimum version 1.12)
  Go version:       go1.12.16
  Git commit:       369ce74a3c
  Built:            Thu Feb 13 01:26:38 2020
  OS/Arch:          linux/amd64
  Experimental:    false
 containerd:
  Version:          1.2.10
  GitCommit:        b34a5c8af56e510852c35414db4c1f4fa6172339
 runc:
  Version:          1.0.0-rc8+dev
  GitCommit:        3e425f80a8c931f88e6d94a8c831b9d5aa481657
 docker-init:
  Version:          0.18.0
  GitCommit:        fec3683
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:9572f7cdcee8591948c2963463447a53466950b3fc15a247fcad1917ca215a2f
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

logout
Connection to 54.254.138.29 closed.
ubuntu@ip-172-31-23-231:~$
```



Team WorkShop

- ▶ Step13: Edit file `terraform.auto.tfvars`, change value of all `instance_type` to `t2.nano` and commit change to git with comment “Edit instance type”

```
Users > paparnlueangphoonlap > Work > Terraform > Resource > TerraformCloudTeamWorkshop > terraform.auto.tfvars
1  instance_type = {
2    "nginx" = "t2.nano"
3    "docker" = "t2.nano"
4    "other" = "t2.nano"
5  }
6
7  user_data = {
8    "0" = <<-EOF
9      #!/bin/bash
10     curl https://raw.githubusercontent.com/praparn/sourcesetup/master/standard\_nginx.sh > /tmp/setup.sh
11     chmod +x /tmp/setup.sh
12     /tmp/setup.sh
13     EOF
14   "1"= <<-EOF
15     #!/bin/bash
16     curl https://raw.githubusercontent.com/praparn/sourcesetup/master/standard\_docker\_aws.sh > /tmp/setup.sh
17     chmod +x /tmp/setup.sh
18     /tmp/setup.sh
19     EOF
20 }
```

Team WorkShop

- ▶ Step13: Edit file `terraform.auto.tfvars`, change value of all `instance_type` to `t2a.nano` and commit change to git with comment “Edit instance type”

```
[praparns-MBP:TerraformCloudTeamWorkshop praparn$ git add -A
[praparns-MBP:TerraformCloudTeamWorkshop praparn$ git commit -m "Edit instance type"
[master 0bc9d8c] Edit instance type
 1 file changed, 2 insertions(+), 2 deletions(-)
[praparns-MBP:TerraformCloudTeamWorkshop praparn$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 297 bytes | 297.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/praparn/IaC-TerraformThailand-User-13.git
 471d7d0..0bc9d8c  master -> master
praparns-MBP:TerraformCloudTeamWorkshop praparn$ ]
```

praparn / IaC-TerraformThailand-User-13 Private

Code Issues 0 Pull requests 0 Actions Projects 0 Security Insights Settings

IaC-TerraformThailand-User-13 Edit

Manage topics

2 commits 1 branch 0 packages 0 releases

Branch: master New pull request Create new file Upload files Find file Clone or download

praparn Edit instance type Latest commit 0bc9d8c 35 seconds ago

File	Commit Type	Time Ago
00_provider.tf	First Commit	1 hour ago
01_data_vpc.tf	First Commit	1 hour ago
02_data_ami.tf	First Commit	1 hour ago
04_security_group.tf	First Commit	1 hour ago
05_resource_ec2.tf	First Commit	1 hour ago
terraform.auto.tfvars	Edit instance type	35 seconds ago
variables.tf	First Commit	1 hour ago

Add a README with an overview of your project. Add a README

Team WorkShop

- ▶ Step14: Check result on terraform cloud (tab run). Review result from terraform plan

Current Run

 Edit instance type	CURRENT	! NEEDS CONFIRMATION
#run-ycCgkZ1Y8GosGwca paparn triggered from GitHub Branch master 0bc9d8c		
2 minutes ago		

Run List

 Edit instance type	CURRENT	! NEEDS CONFIRMATION
#run-ycCgkZ1Y8GosGwca paparn triggered from GitHub Branch master 0bc9d8c		
2 minutes ago		

Queued from Terraform Cloud UI

 Edit instance type	CURRENT	✓ APPLIED
#run-btP2VvFfkjhJAF3H IaC-TerraformThailand-Master triggered from Terraform Cloud UI Branch master 471d7d0		
36 minutes ago		

Team WorkShop

- ▶ Step14: Check result on terraform cloud (tab run). Review result from terraform plan

The screenshot shows a Terraform Cloud interface. At the top, there are buttons for "View raw log", "Top", "Bottom", "Expand", and "Full screen". The main area displays a black terminal-like window containing Terraform configuration code and a plan summary. The code includes sections for `credit_specification` and `root_block_device`. Below the code, a message says "Plan: 0 to add, 2 to change, 0 to destroy.". At the bottom, a button labeled "Apply pending" is shown, along with a confirmation message: "Needs Confirmation: Check the plan and confirm to apply it, or discard the run." with three buttons: "Confirm & Apply", "Discard Run", and "Add Comment".

```
]
credit_specification {
  cpu_credits = "unlimited"
}

root_block_device {
  delete_on_termination = true
  encrypted             = false
  iops                  = 100
  volume_id              = "vol-090ce93ecc04d91b5"
  volume_size            = 10
  volume_type            = "gp2"
}
}

Plan: 0 to add, 2 to change, 0 to destroy.
```

⌚ Apply pending

Needs Confirmation: Check the plan and confirm to apply it, or discard the run.

Confirm & Apply Discard Run Add Comment

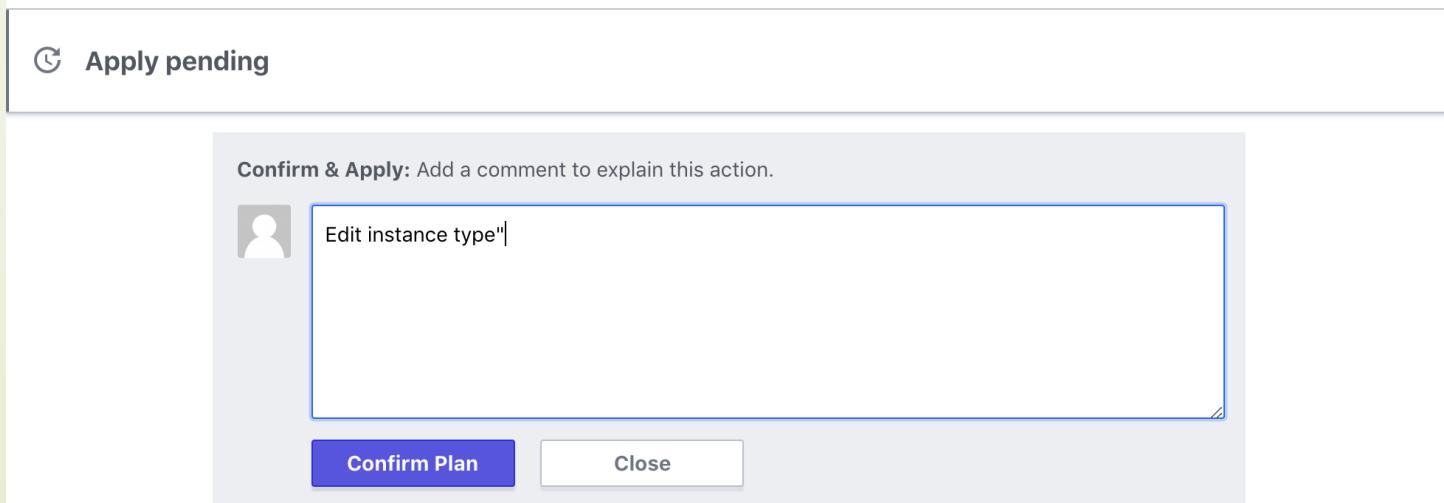
Team WorkShop

- ▶ Step15: Choose “Confirm & Apply” with comment “Edit instance type”

```
]
credit_specification {
    cpu_credits = "unlimited"
}

root_block_device {
    delete_on_termination = true
    encrypted            = false
    iops                 = 100
    volume_id            = "vol-090ce93ecc04d91b5"
    volume_size          = 10
    volume_type          = "gp2"
}
}

Plan: 0 to add, 2 to change, 0 to destroy.
```



Team WorkShop

- ▶ Step16: Monitor progress on terraform cloud until finished

✓ APPLIED Edit instance type CURRENT

 **paparn** triggered a **run** from GitHub a minute ago Run Details ▾

✓ Plan finished 2 minutes ago Resources: 0 to add, 2 to change, 0 to destroy ▾

✓ Apply finished a few seconds ago Resources: 0 added, 2 changed, 0 destroyed ▾

Started a minute ago > Finished a few seconds ago

[View raw log](#) [Top](#) [Bottom](#) [Expand](#) [Full screen](#)

```
aws_instance.labserver[1]: Still modifying... [id=i-07d42934b3fabfdb9, 10s elapsed]
aws_instance.labserver[0]: Still modifying... [id=i-003702e0d19772b95, 20s elapsed]
aws_instance.labserver[1]: Still modifying... [id=i-07d42934b3fabfdb9, 20s elapsed]
aws_instance.labserver[0]: Still modifying... [id=i-003702e0d19772b95, 30s elapsed]
aws_instance.labserver[1]: Still modifying... [id=i-07d42934b3fabfdb9, 30s elapsed]
aws_instance.labserver[1]: Modifications complete after 32s [id=i-07d42934b3fabfdb9]
aws_instance.labserver[0]: Still modifying... [id=i-003702e0d19772b95, 40s elapsed]
aws_instance.labserver[0]: Modifications complete after 42s [id=i-003702e0d19772b95]

Apply complete! Resources: 0 added, 2 changed, 0 destroyed.

Outputs:

aws_instance_labserver_ip = {
  "i-003702e0d19772b95" = "52.77.219.0"
  "i-07d42934b3fabfdb9" = "13.229.121.116"
}
```

Team WorkShop

► Step17: Verify ec2 status by aws (Both Server)

```
ubuntu@ip-172-31-23-231:~$ aws ec2 describe-instances --instance-ids i-07d42934b3fabfdb9
{
    "Reservations": [
        {
            "OwnerId": "404075494050",
            "ReservationId": "r-0c4f23923f708c74a",
            "Instances": [
                {
                    "AmiLaunchIndex": 0,
                    "Architecture": "x86_64",
                    "ClientToken": "False",
                    "EbsOptimized": "True",
                    "EnaSupport": "True",
                    "Hypervisor": "xen",
                    "ImageId": "ami-81cefccfd",
                    "InstanceId": "i-07d42934b3fabfdb9",
                    "InstanceType": "t2.nano",
                    "KeyName": "keypair",
                    "LaunchTime": "2020-02-15T12:41:28.000Z",
                    "PrivateDnsName": "ip-172-31-31-241.ap-southeast-1.compute.internal",
                    "PrivateIpAddress": "172.31.31.241",
                    "PublicDnsName": "ec2-13-229-121-116.ap-southeast-1.compute.amazonaws.com",
                    "PublicIpAddress": "13.229.121.116",
                    "RootDeviceName": "/dev/sda1",
                    "RootDeviceType": "ebs",
                    "SourceDestCheck": "True",
                    "StateTransitionReason": "None",
                    "SubnetId": "subnet-41742436",
                    "VirtualizationType": "hvm",
                    "VpcId": "vpc-4483e921",
                    "BlockDeviceMappings": [
                        {
                            "DeviceName": "/dev/sda1",
                            "Ebs": {
                                "AttachTime": "2020-02-15T11:58:08.000Z",
                                "DeleteOnTermination": "True",
                                "Status": "attached",
                                "VolumeId": "vol-090ce93ecc04d91b5"
                            }
                        }
                    ],
                    "CapacityReservationSpecification": {
                        "CapacityReservationPreference": "open"
                    },
                    "CpuOptions": {
                        "CoreCount": 1,
                        "ThreadsPerCore": 1
                    },
                    "HibernationOptions": {
                        "Configured": "False"
                    },
                    "MetadataOptions": {
                        "HttpEndpoint": "enabled",
                        "HttpPutResponseHopLimit": 1,
                        "HttpTokens": "optional",
                        "State": "applied"
                    },
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "NetworkInterfaces": [
                        {
                            "Description": "interface",
                            "InterfaceType": "ena-03d819bd6b4c99d2b",
                            "MacAddress": "06:37:6d:34:81:fa",
                            "NetworkInterfaceId": "eni-03d819bd6b4c99d2b",
                            "OwnerId": "404075494050",
                            "PrivateDnsName": "ip-172-31-31-241.ap-southeast-1.compute.internal",
                            "PrivateIpAddress": "172.31.31.241",
                            "SourceDestCheck": "True",
                            "Status": "in-use",
                            "SubnetId": "subnet-41742436",
                            "VpcId": "vpc-4483e921"
                        }
                    ],
                    "Association": {
                        "IpOwnerId": "amazon",
                        "PublicDnsName": "ec2-13-229-121-116.ap-southeast-1.compute.amazonaws.com",
                        "PublicIp": "13.229.121.116"
                    },
                    "Placement": {
                        "AvailabilityZone": "ap-southeast-1a",
                        "GroupName": null,
                        "Tenancy": "default"
                    },
                    "SecurityGroups": [
                        {
                            "GroupId": "sg-05ae8d4816c74a111",
                            "GroupName": "securitygroup-server1-student-X"
                        }
                    ],
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    },
                    "Tags": [
                        {
                            "Key": "Region",
                            "Value": "ap-southeast-1"
                        },
                        {
                            "Key": "Name",
                            "Value": "labresource-server2-student-13"
                        },
                        {
                            "Key": "Categories",
                            "Value": "compute"
                        },
                        {
                            "Key": "AZ",
                            "Value": "ap-southeast-1a"
                        },
                        {
                            "Key": "Billing",
                            "Value": "terraform-workshop"
                        },
                        {
                            "Key": "Environment",
                            "Value": "rd"
                        },
                        {
                            "Key": "Zone",
                            "Value": "public"
                        },
                        {
                            "Key": "Module",
                            "Value": "ec2"
                        }
                    ]
                }
            ]
        }
    ]
}
```

Team WorkShop

► Step18: Verify ec2 status by aws (Both Server)

```
[ubuntu@ip-172-31-23-231:~$ ssh -i ~/terraformlab ubuntu@52.77.219.0
The authenticity of host '52.77.219.0 (52.77.219.0)' can't be established.
ECDSA key fingerprint is SHA256:FRUzmNK5n7Y8GBA14Y+khltENS8MCNl73g0WyVwJqQw.
[Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '52.77.219.0' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-1060-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 Get cloud support with Ubuntu Advantage Cloud Guest:
   http://www.ubuntu.com/business/services/cloud

204 packages can be updated.
133 updates are security updates.

New release '18.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat Feb 15 19:17:22 2020 from 52.221.230.248
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

```
[ubuntu@ip-172-31-19-90:~$ ps -ef | grep nginx && curl http://localhost && exit
root      1220      1  0 19:41 ?          00:00:00 nginx: master process /usr/sbin/nginx -g daemon on; master_process
www-data  1221  1220  0 19:41 ?          00:00:00 nginx: worker process
ubuntu    1403  1387  0 19:49 pts/0    00:00:00 grep --color=auto nginx
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
}
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
logout
Connection to 52.77.219.0 closed.
ubuntu@ip-172-31-23-231:~$ ]
```

Team WorkShop

► Step18: Verify ec2 status by aws (Both Server)

```
[ubuntu@ip-172-31-23-231:~$ ssh -i ~/terraformlab ubuntu@13.229.121.116
The authenticity of host '13.229.121.116 (13.229.121.116)' can't be established.
ECDSA key fingerprint is SHA256:0XjBYpBKVa3rvySg5qAMN8omoCEbY03008+k/kAjnkc.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '13.229.121.116' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-1060-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
 http://www.ubuntu.com/business/services/cloud

197 packages can be updated.
129 updates are security updates.

New release '18.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat Feb 15 19:19:23 2020 from 52.221.230.248
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

[ubuntu@ip-172-31-23-241:~$ docker version && docker run hello-world && exit
Client: Docker Engine - Community
 Version:          19.03.6
 API version:     1.40
 Go version:      go1.12.16
 Git commit:      369ce74a3c
 Built:            Thu Feb 13 01:28:06 2020
 OS/Arch:          linux/amd64
 Experimental:    false

Server: Docker Engine - Community
 Engine:
  Version:          19.03.6
  API version:     1.40 (minimum version 1.12)
  Go version:      go1.12.16
  Git commit:      369ce74a3c
  Built:            Thu Feb 13 01:26:38 2020
  OS/Arch:          linux/amd64
  Experimental:    false
 containerd:
  Version:          1.2.10
  GitCommit:        b34a5c8af56e510852c35414db4c1f4fa6172339
 runc:
  Version:          1.0.0-rc8+dev
  GitCommit:        3e425f80a8c931f88e6d94a8c831b9d5aa481657
 docker-init:
  Version:          0.18.0
  GitCommit:        fec3683
```

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub. (amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
\$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
<https://hub.docker.com/>

For more examples and ideas, visit:
<https://docs.docker.com/get-started/>

logout
Connection to 13.229.121.116 closed.
ubuntu@ip-172-31-23-231:~\$ █

Team WorkShop

- ▶ Step19: Setup environment variable " CONFIRM_DESTROY=1" for allow terraform destroy

Environment Variables

These variables are set in Terraform's shell environment using `export`.

Key	Value	...
AWS_ACCESS_KEY_ID	AKIAV4FGGXKRDU62DMNS	...
AWS_SECRET_ACCESS_KEY	P/7UgRJ9UjbbcOJgpbvVvAWQGJQrG4xT3/vUL2Jpu	...
CONFIRM_DESTROY	1	...
+ Add variable		

Team WorkShop

- ▶ Step20: Go to tab “Settings” → “Destruction and Deletion”, Choose “Queue destroy plan”

The screenshot shows the Terraform Cloud interface for workspace "laC-TerraformThailand-User-13". The top navigation bar includes tabs for "Runs", "States", "Variables" (which is the active tab), and "Settings". A dropdown menu under "Settings" shows options like "Queue plan", "General", "Locking", "Notifications", "Run Triggers", "SSH Key", "Version Control", and "Destruction and Deletion". The "Destruction and Deletion" tab is highlighted with a blue background. The main content area is titled "Variables" and contains sections for "General", "Locking", "Notifications", "Run Triggers", "SSH Key", "Version Control", and "Destruction and Deletion". The "Destruction and Deletion" section is expanded, showing sub-options: "from any", "variables", and "Destruction and Deletion". Below this, there is a large heading "Destruction and Deletion" and descriptive text about the two-step destruction process. At the bottom right, there are two buttons: "Queue destroy plan" and "Delete from Terraform Cloud".

laC-TerraformThailand-User-13 ⓘ

Runs States Variables Settings Queue plan General Locking Notifications Run Triggers SSH Key Version Control Destruction and Deletion

Variables

These variables are used for all plans and applies in this workspace. Workspaces using Terraform 0.10.0 or later can have `*.auto.tfvars` files in the configuration.

Sensitive variables are hidden from view in the UI and API, and can't be edited. (To change a sensitive variable, you must use the Terraform Cloud Provider or the variables API.)

When setting many variables at once, the Terraform Cloud Provider or the variables API can often save time.

Terraform Variables

These Terraform variables are set using a `terraform.tfvars` file. To use interpolation or set a non-string value, use the Terraform Cloud Provider or the variables API.

Destruction and Deletion

There are two independent steps for destroying this workspace and any infrastructure associated with it. First, any Terraform infrastructure should be destroyed. Second, the workspace in Terraform Cloud, including any variables, settings, and alert history can be deleted.

Queueing a destroy Plan will redirect to a new Plan that will destroy all of the infrastructure managed by Terraform. It is equivalent to running `terraform plan --destroy -out=destroy.tfplan` followed by `terraform apply destroy.tfplan` locally.

Queueing a destroy Plan will be disabled until there is an environment variable set named `CONFIRM_DESTROY` with a value of `1`. You can use the variables page to set it.

Queue destroy plan Delete from Terraform Cloud

Team WorkShop

- ▶ Step21: Go to tab “run”, review terraform plan for destroy

! NEEDS CONFIRMATION **Queued manually to destroy infrastructure** CURRENT

IaC-TerraformThailand-Master triggered a run from Terraform Cloud UI a few seconds ago Run Details ▾

Plan finished a minute ago Resources: 0 to add, 0 to change, 3 to destroy ^

Started a few seconds ago

[Download Sentinel mocks](#) (i) Sentinel mocks can be used for testing your Sentinel policies

[View raw log](#) Following log Top Bottom Expand Full screen

```
- name = "securitygroup-server1-student-X" -> null
- owner_id = "404075494050" -> null
- revoke_rules_on_delete = false -> null
- tags = {
    - "AZ" = "none"
    - "Billing" = "terraform-workshop"
    - "Categories" = "compute"
    - "Environment" = "rd"
    - "Module" = "secgroup"
    - "Name" = "securitygroup-server1-student-X"
    - "Region" = "ap-southeast-1"
    - "Zone" = "public"
  } -> null
- vpc_id = "vpc-4483e921" -> null
}
```

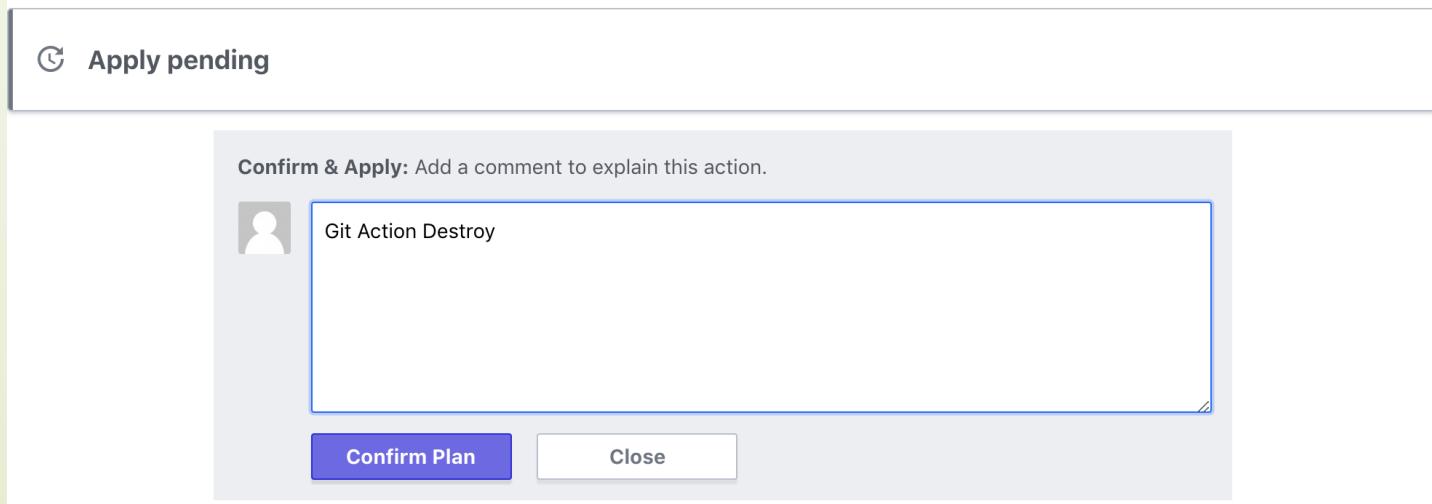
Plan: 0 to add, 0 to change, 3 to destroy.

Team WorkShop

- ▶ Step22: Choose “Confirm & Apply” with comment “Git Action Destroy”

```
- name          = "securitygroup-server1-student-X" -> null
- owner_id     = "404075494050" -> null
- revoke_rules_on_delete = false -> null
- tags          = {
  - "AZ"          = "none"
  - "Billing"     = "terraform-workshop"
  - "Categories" = "compute"
  - "Environment" = "rd"
  - "Module"      = "secgroup"
  - "Name"        = "securitygroup-server1-student-X"
  - "Region"      = "ap-southeast-1"
  - "Zone"        = "public"
} -> null
- vpc_id       = "vpc-4483e921" -> null
}

Plan: 0 to add, 0 to change, 3 to destroy.
```

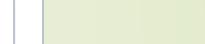


Team WorkShop

► Step23: Monitor Result until finished

 **Apply running** a few seconds ago

Started a few seconds ago

```
Terraform v0.12.20
Initializing plugins and modules...
2020/02/15 13:10:25 [DEBUG] Using modified User-Agent: Terraform/0.12.20 TFC/d310d4ebb1
aws_instance.labserver[1]: Destroying... [id=i-07d42934b3fabfdb9]
aws_instance.labserver[0]: Destroying... [id=i-003702e0d19772b95]
aws_instance.labserver[1]: Still destroying... [id=i-07d42934b3fabfdb9, 10s elapsed]
aws_instance.labserver[0]: Still destroying... [id=i-003702e0d19772b95, 10s elapsed]
aws_instance.labserver[1]: Still destroying... [id=i-07d42934b3fabfdb9, 20s elapsed]
aws_instance.labserver[0]: Still destroying... [id=i-003702e0d19772b95, 20s elapsed]
aws_instance.labserver[1]: Destruction complete after 22s
```

Team WorkShop

► Step23: Monitor Result until finished



Apply finished a few seconds ago Resources: 0 added, 0 changed, 3 destroyed ^

Started 2 minutes ago > Finished a few seconds ago

[View raw log](#) [Top](#) [Bottom](#) [Expand](#) [Full screen](#)

```
aws_instance.labserver[1]: Destroying... [id=i-07d42934b3fabfdb9]
aws_instance.labserver[0]: Destroying... [id=i-003702e0d19772b95]
aws_instance.labserver[1]: Still destroying... [id=i-07d42934b3fabfdb9, 10s elapsed]
aws_instance.labserver[0]: Still destroying... [id=i-003702e0d19772b95, 10s elapsed]
aws_instance.labserver[1]: Still destroying... [id=i-07d42934b3fabfdb9, 20s elapsed]
aws_instance.labserver[0]: Still destroying... [id=i-003702e0d19772b95, 20s elapsed]
aws_instance.labserver[1]: Destruction complete after 22s
aws_instance.labserver[0]: Still destroying... [id=i-003702e0d19772b95, 30s elapsed]
aws_instance.labserver[0]: Destruction complete after 33s
aws_security_group.secgroup_server: Destroying... [id=sg-05ae8d4816c74a111]
aws_security_group.secgroup_server: Destruction complete after 2s

Apply complete! Resources: 0 added, 0 changed, 3 destroyed.
```

Outputs:

```
aws_instance_labserver_ip = {}
```



Team WorkShop

► Step24: Verify ec2 status on aws cli (Both Server)

DescribeInstances	
Reservations	
OwnerId	404075494050
ReservationId	r-09c55d3fe73a4306b
Instances	
AmiLaunchIndex	0
Architecture	x86_64
ClientToken	
EbsOptimized	False
EnaSupport	True
Hypervisor	xen
ImageId	ami-81cefccfd
InstanceId	i-003702e0d19772b95
InstanceType	t2.nano
KeyName	keypair
LaunchTime	2020-02-15T12:41:28.000Z
PrivateDnsName	
PublicDnsName	
RootDeviceName	/dev/sda1
RootDeviceType	ebs
StateTransitionReason	User initiated (2020-02-15 13:10:29 GMT)
VirtualizationType	hvm

State	
Code	48
Name	
	terminated
StateReason	
Code	Client.UserInitiatedShutdown
Message	Client.UserInitiatedShutdown: User initiated shutdown
Tags	
Key	
Value	
Categories	compute
Module	ec2
Name	labresource-server1-student-13
Region	ap-southeast-1
Billing	terraform-workshop
Environment	rd
AZ	ap-southeast-1a
Zone	public

ubuntu@ip-172-31-23-231:~\$ █

Team WorkShop

- ▶ Step25: Go to tab “Settings” → “Destruction and Deletion”, Choose “Delete from Terraform Cloud”

The screenshot shows the Terraform Cloud interface for a workspace named "laC-TerraformThailand-User-13". The top navigation bar includes tabs for "Runs", "States", "Variables" (which is the active tab), and "Settings". A dropdown menu for "Queue plan" is also present. On the left, there's a sidebar with sections for "Variables" and "Terraform Variables". The main content area is titled "Destruction and Deletion". It contains text explaining the two-step destruction process and information about queueing destroy plans. At the bottom right of the content area are two buttons: "Queue destroy plan" and "Delete from Terraform Cloud".

laC-TerraformThailand-User-13 ⓘ

Runs States Variables Settings Queue plan

Variables

These variables are used for all plans and applies in this workspace. Workspaces using Terraform 0.10.0 or later can have `*.auto.tfvars` files in the configuration.

Sensitive variables are hidden from view in the UI and API, and can't be edited. (To change a sensitive variable, you must use the Terraform Cloud Provider or the variables API.)

When setting many variables at once, the Terraform Cloud Provider or the variables API can often save time.

Terraform Variables

These Terraform variables are set using a `terraform.tfvars` file. To use interpolation or set a non-string value, use the Terraform Cloud Provider or the variables API.

General
Locking
Notifications
Run Triggers
SSH Key
Version Control
Destruction and Deletion

Destruction and Deletion

There are two independent steps for destroying this workspace and any infrastructure associated with it. First, any Terraform infrastructure should be destroyed. Second, the workspace in Terraform Cloud, including any variables, settings, and alert history can be deleted.

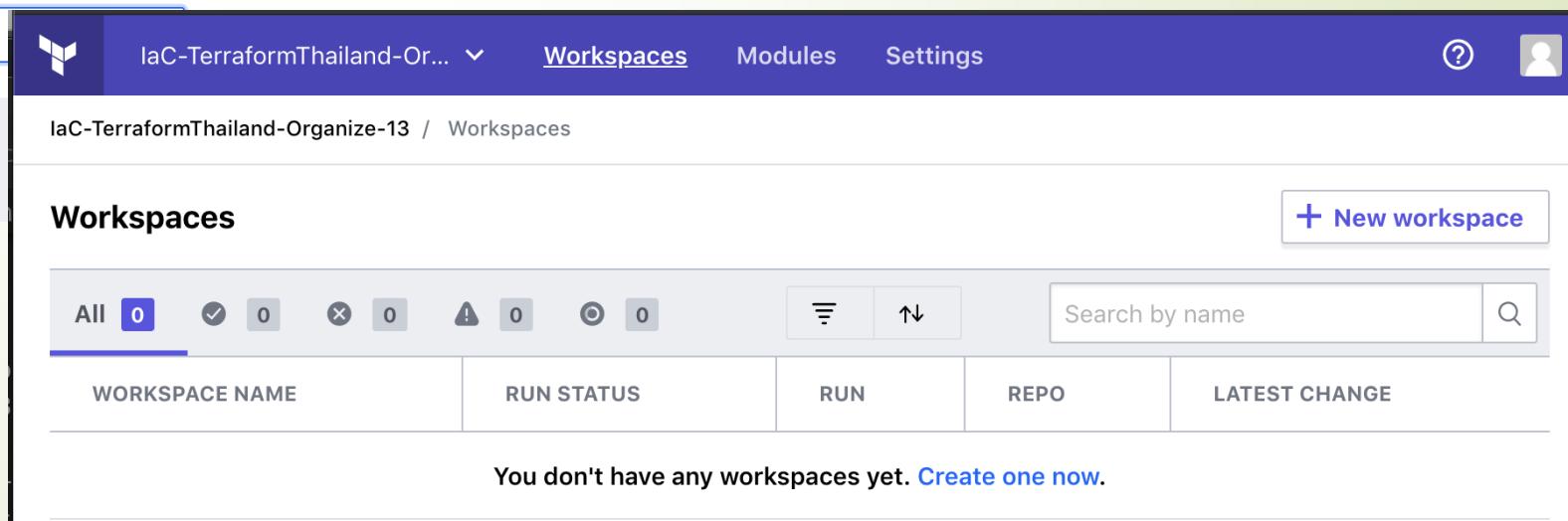
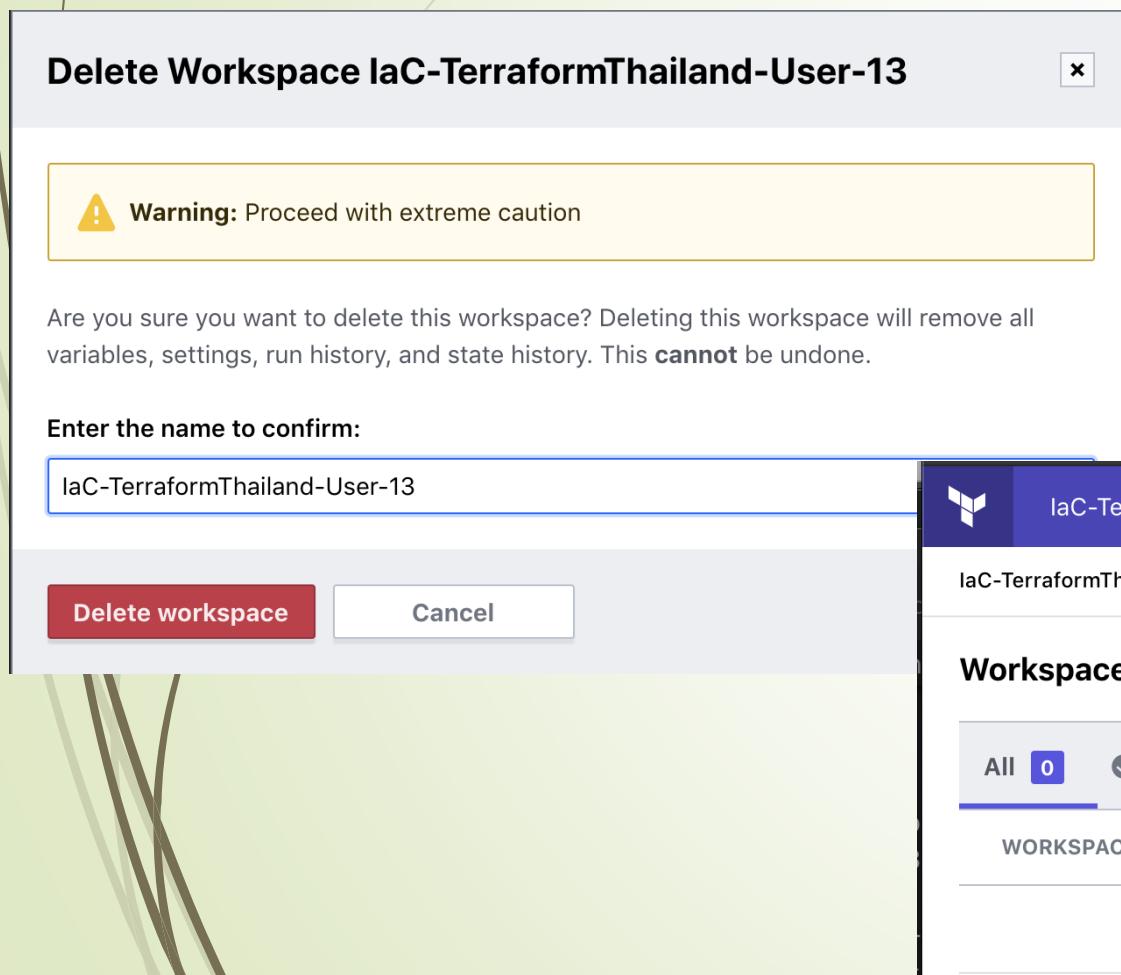
Queueing a destroy Plan will redirect to a new Plan that will destroy all of the infrastructure managed by Terraform. It is equivalent to running `terraform plan --destroy --out=destroy.tfplan` followed by `terraform apply destroy.tfplan` locally.

Queueing a destroy Plan will be disabled until there is an environment variable set named `CONFIRM_DESTROY` with a value of `1`. You can use the variables page to set it.

Queue destroy plan Delete from Terraform Cloud

Team WorkShop

► Step26: Input workspace name for delete



The image shows the main workshop interface. At the top, there is a navigation bar with a logo, the text "IaC-TerraformThailand-Or...", and tabs for "Workspaces", "Modules", and "Settings". To the right of the tabs are a help icon and a user profile icon. The main area is titled "IaC-TerraformThailand-Organize-13 / Workspaces". A "Workspaces" section displays a table with columns: WORKSPACE NAME, RUN STATUS, RUN, REPO, and LATEST CHANGE. The table currently shows "All 0" workspaces. A button "+ New workspace" is located in the top right of this section. Below the table, a message says "You don't have any workspaces yet. Create one now."

Team WorkShop

► Step27: Delete repository from github

The screenshot shows the GitHub repository settings page for 'paparn / IaC-TerraformThailand-User-13'. The 'Settings' tab is selected. On the right, a modal window titled 'Are you absolutely sure?' displays a warning message: 'Unexpected bad things will happen if you don't read this!', 'This action cannot be undone. This will permanently delete the paparn/IaC-TerraformThailand-User-13 repository, wiki, issues, and comments, and remove all collaborator associations.', and 'This will not change your billing plan. If you want to downgrade, you can do so in your Billing Settings.' A text input field contains 'paparn/IaC-TerraformThailand-User-13' and a button labeled 'I understand the consequences, delete this repository'. Below the modal, sections for 'Danger Zone' include 'Make this repository public', 'Transfer ownership', 'Archive this repository', and 'Delete this repository'.

Search or jump to... / Pull requests Issues Marketplace Explore

paparn / IaC-TerraformThailand-User-13 Private

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Security Insights Settings

Options

Manage access
Branches
Webhooks
Notifications
Integrations & services
Deploy keys
Secrets
Actions

Settings

Repository name
IaC-TerraformThailand-User-13 Rename

Template repository
Template repositories let users generate new repositories with the same directory structure and files. Indicate if paparn/IaC-TerraformThailand-User-13 can be used as a template for creating other repositories.

Social preview

⚠️ You can upload a social image, but it will not be visible publicly while paparn/IaC-TerraformThailand-User-13 is private.
Upload an image to customize your repository's social media preview.

Are you absolutely sure?

Unexpected bad things will happen if you don't read this!

This action cannot be undone. This will permanently delete the paparn/IaC-TerraformThailand-User-13 repository, wiki, issues, and comments, and remove all collaborator associations.

This will not change your billing plan. If you want to downgrade, you can do so in your Billing Settings.

Please type paparn/IaC-TerraformThailand-User-13 to confirm.

paparn/IaC-TerraformThailand-User-13

I understand the consequences, delete this repository

Upgrade to GitHub Pro or make this repository public to enable Pages.

Make this repository public

Transfer ownership

Archive this repository

Delete this repository

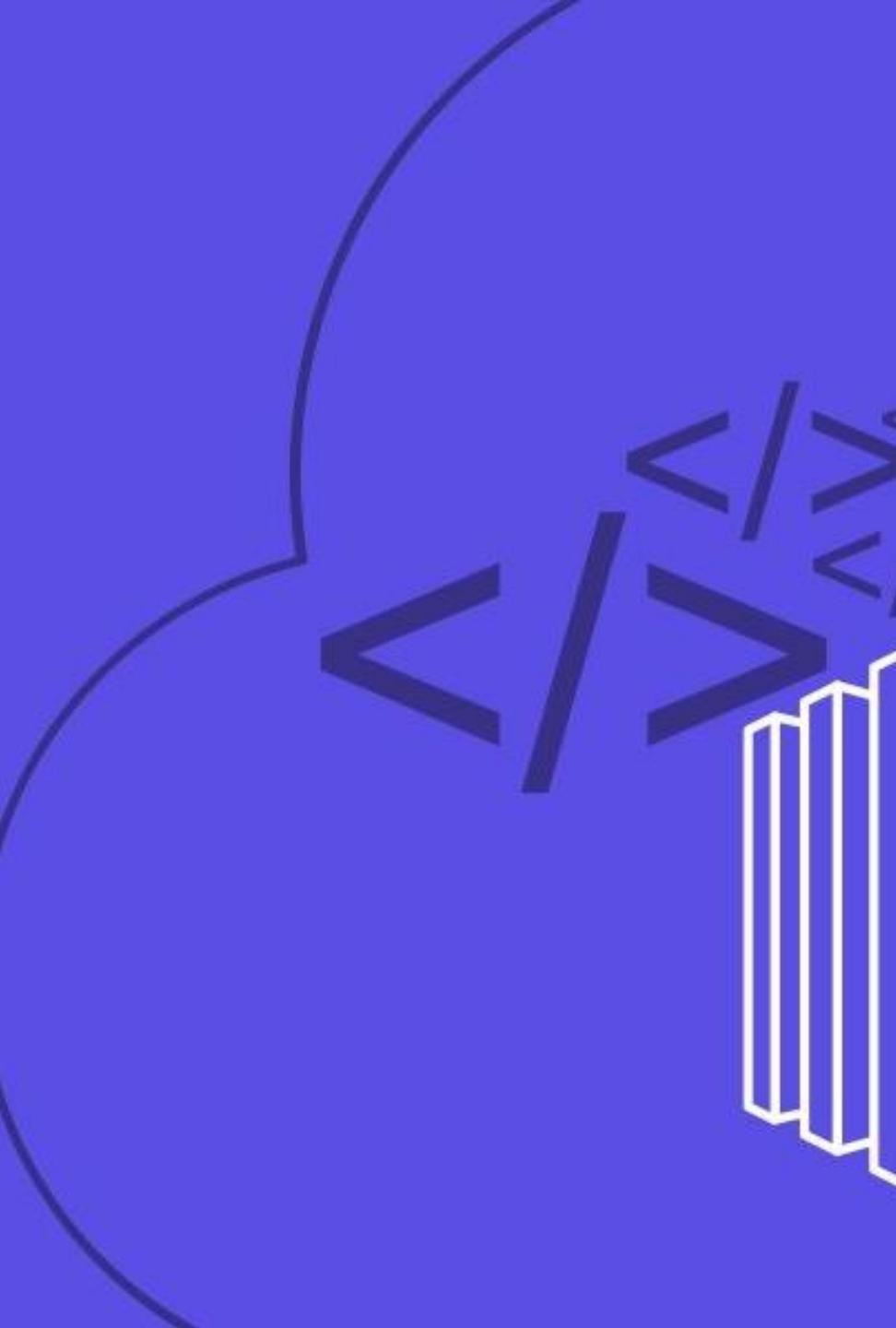
Team WorkShop

► Step28: Cleanup Lab and shutdown remote server

```
ubuntu@ip-172-31-23-231:~$ cd ~/  
ubuntu@ip-172-31-23-231:~$ rm -rf /*  
[ubuntu@ip-172-31-23-231:~$ sudo shutdown  
Shutdown scheduled for Sat 2020-02-15 13:24:02 UTC, use 'shutdown -c' to cancel.  
ubuntu@ip-172-31-23-231:~$ █
```

Q&A





Agenda (Day 2)

- ▶ Dive to HCL (Hashicorp Configuration Language) for terraform (Cont.)
 - ▶ Variable (Input & Output)
 - ▶ Registry and Module
- ▶ Terraform cloud for team collaboration
 - ▶ Introduction to terraform cloud
 - ▶ Setup workspace and test drive
- ▶ Team workshop (Final Workshop)
- ▶ Q&A



Q&A

Praparn Luangphoonlap

