รหัสนักศึกษา :

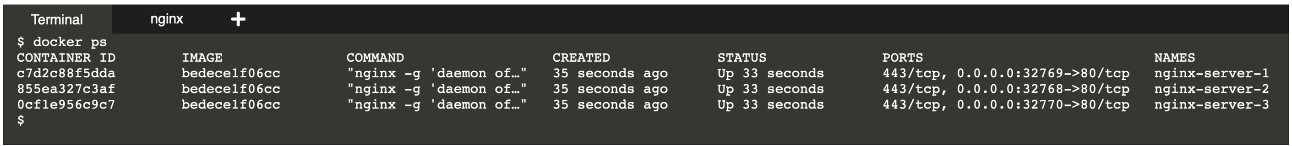
Lab environment

https://www.katacoda.com/saranonuan/scenarios/terraform101

หากไม่สามารถเข้าได้ ให้ลองเข้า URL

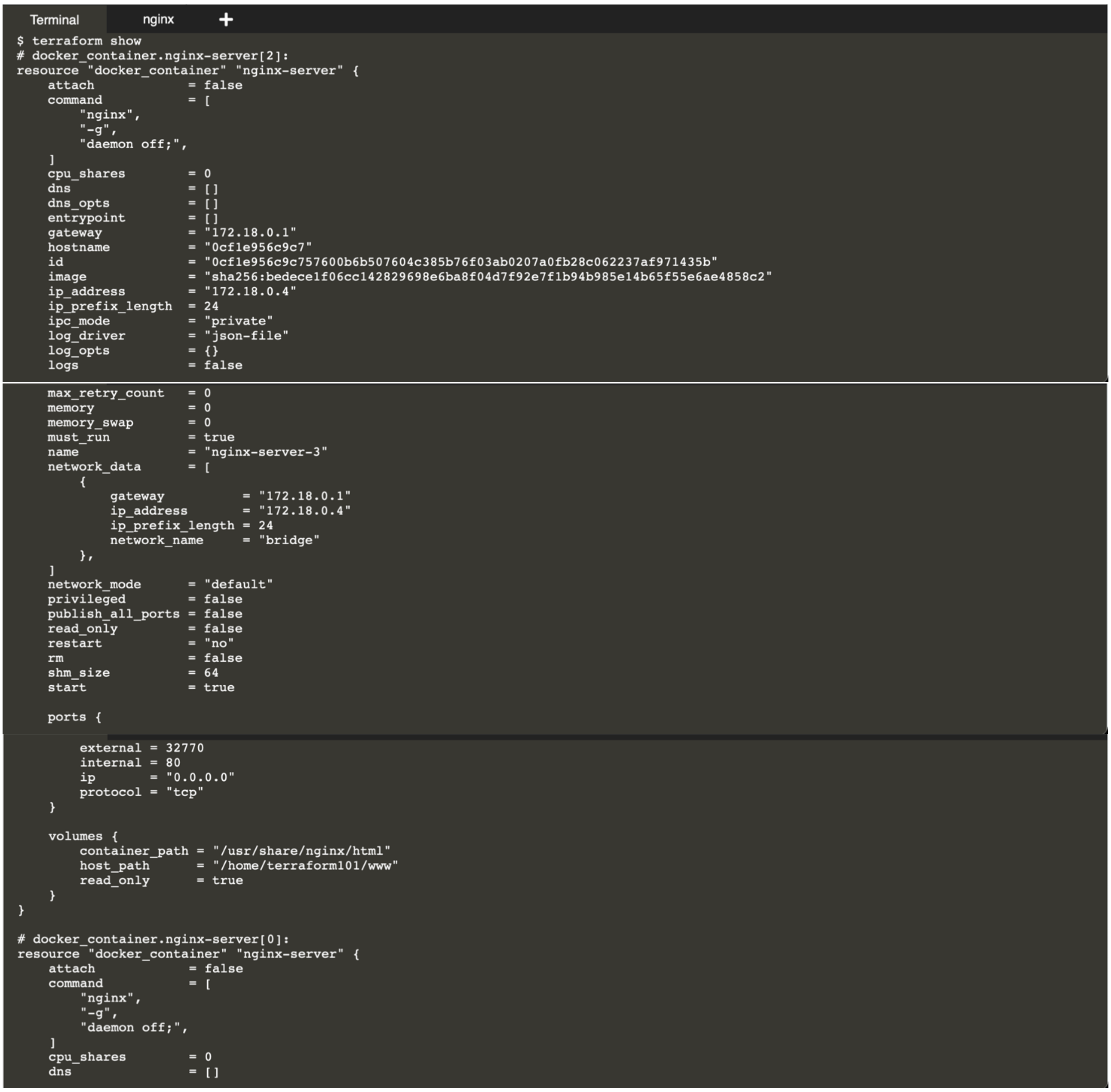
https://www.katacoda.com/saranonuan/training/terraform

A. Capture Screen output เมื่อรัน docker ps ใน Step 9

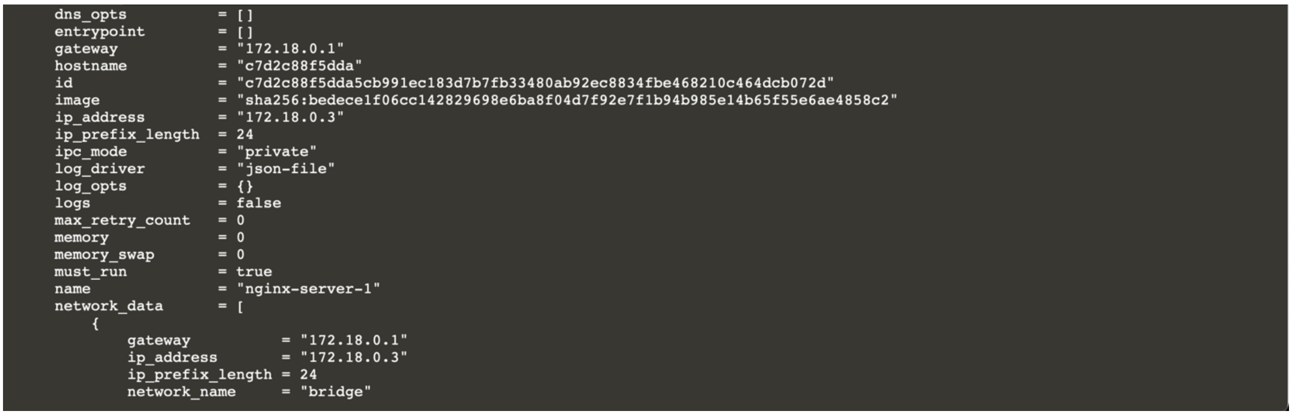


```
$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED              STATUS              PORTS                                   NAMES
c7d2c88f5dda        bedece1f06cc        "nginx -g 'daemon of…"   35 seconds ago       Up 33 seconds       443/tcp, 0.0.0.0:32769->80/tcp          nginx-server-1
855ea327c3af        bedece1f06cc        "nginx -g 'daemon of…"   35 seconds ago       Up 33 seconds       443/tcp, 0.0.0.0:32768->80/tcp          nginx-server-2
0cf1e956c9c7        bedece1f06cc        "nginx -g 'daemon of…"   35 seconds ago       Up 33 seconds       443/tcp, 0.0.0.0:32770->80/tcp          nginx-server-3
$
```

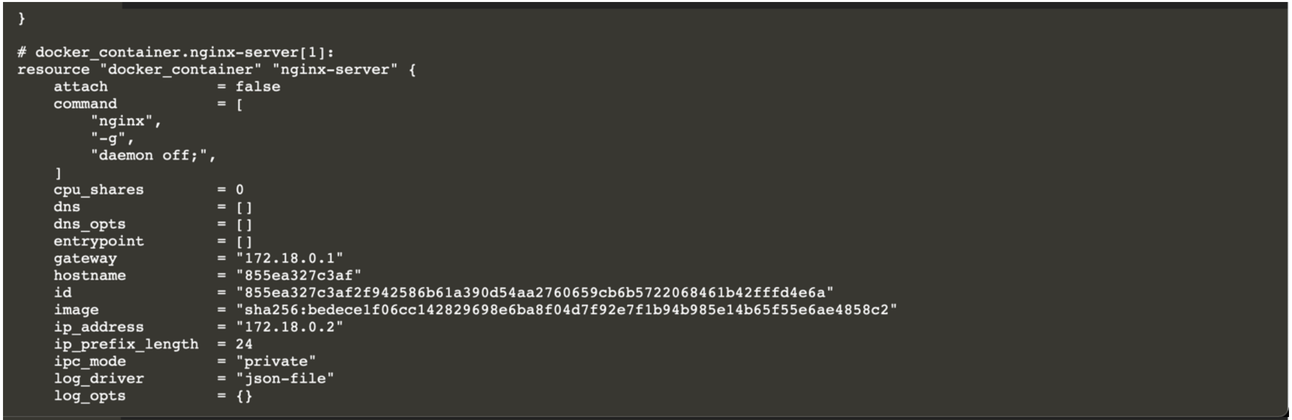
B. Capture Screen output เมื่อรัน terraform show ใน Step 9



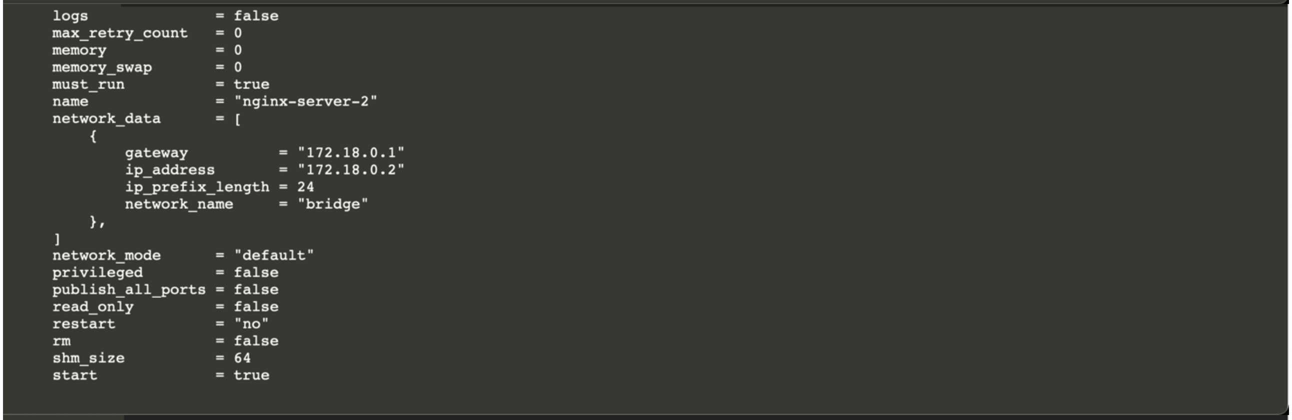
```
$ terraform show
# docker_container.nginx-server[2]:
resource "docker_container" "nginx-server" {
    attach            = false
    command           = [
        "nginx",
        "-g",
        "daemon off;",
    ]
    cpu_shares        = 0
    dns               = []
    dns_opts          = []
    entrypoint        = []
    gateway           = "172.18.0.1"
    hostname          = "0cf1e956c9c7"
    id                = "0cf1e956c9c757600b6b507604c385b76f03ab0207a0fb28c062237af971435b"
    image             = "sha256:bedece1f06cc142829698e6ba8f04d7f92e7f1b94b985e14b65f55e6ae4858c2"
    ip_address        = "172.18.0.4"
    ip_prefix_length  = 24
    ipc_mode          = "private"
    log_driver        = "json-file"
    log_opts          = {}
    logs              = false
    max_retry_count   = 0
    memory            = 0
    memory_swap       = 0
    must_run          = true
    name              = "nginx-server-3"
    network_data      = [
        {
            gateway          = "172.18.0.1"
            ip_address       = "172.18.0.4"
            ip_prefix_length = 24
            network_name     = "bridge"
        },
    ]
    network_mode      = "default"
    privileged        = false
    publish_all_ports = false
    read_only         = false
    restart           = "no"
    rm                = false
    shm_size          = 64
    start             = true

    ports {
        external = 32770
        internal = 80
        ip       = "0.0.0.0"
        protocol = "tcp"
    }

    volumes {
        container_path = "/usr/share/nginx/html"
        host_path      = "/home/terraform101/www"
        read_only      = true
    }
}

# docker_container.nginx-server[0]:
resource "docker_container" "nginx-server" {
    attach            = false
    command           = [
        "nginx",
        "-g",
        "daemon off;",
    ]
    cpu_shares        = 0
    dns               = []
```



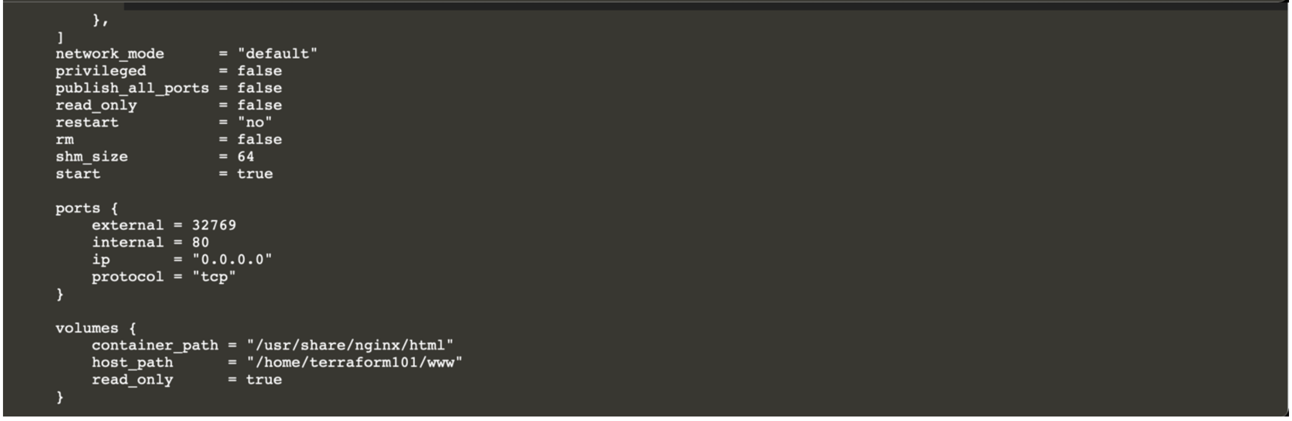
```
    dns_opts          = []
    entrypoint        = []
    gateway           = "172.18.0.1"
    hostname          = "c7d2c88f5dda"
    id                = "c7d2c88f5dda5cb991ec183d7b7fb33480ab92ec8834fbe468210c464dcb072d"
    image             = "sha256:bedece1f06cc142829698e6ba8f04d7f92e7f1b94b985e14b65f55e6ae4858c2"
    ip_address        = "172.18.0.3"
    ip_prefix_length  = 24
    ipc_mode          = "private"
    log_driver        = "json-file"
    log_opts          = {}
    logs              = false
    max_retry_count   = 0
    memory            = 0
    memory_swap       = 0
    must_run          = true
    name              = "nginx-server-1"
    network_data      = [
        {
            gateway          = "172.18.0.1"
            ip_address       = "172.18.0.3"
            ip_prefix_length = 24
            network_name     = "bridge"
```



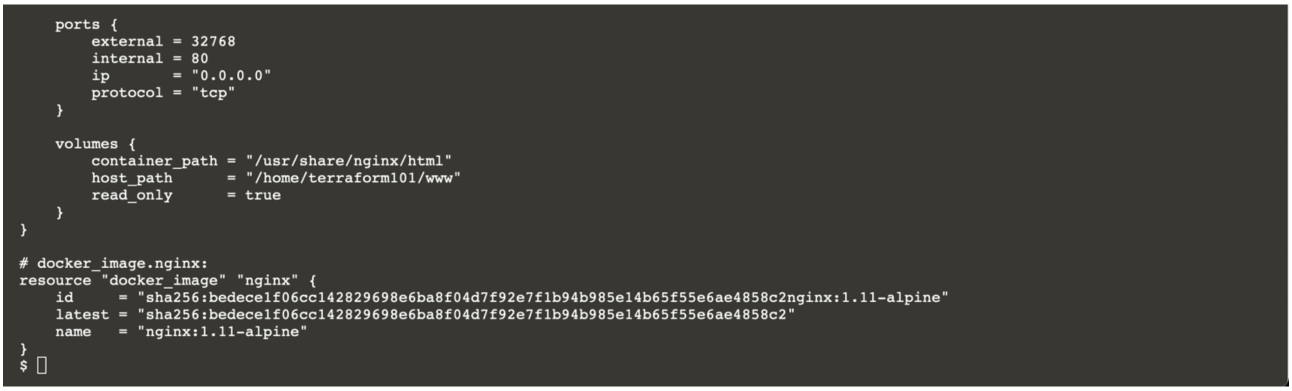
```
}

# docker_container.nginx-server[1]:
resource "docker_container" "nginx-server" {
    attach            = false
    command           = [
        "nginx",
        "-g",
        "daemon off;",
    ]
    cpu_shares        = 0
    dns               = []
    dns_opts          = []
    entrypoint        = []
    gateway           = "172.18.0.1"
    hostname          = "855ea327c3af"
    id                = "855ea327c3af2f942586b61a390d54aa2760659cb6b5722068461b42fffd4e6a"
    image             = "sha256:bedece1f06cc142829698e6ba8f04d7f92e7f1b94b985e14b65f55e6ae4858c2"
    ip_address        = "172.18.0.2"
    ip_prefix_length  = 24
    ipc_mode          = "private"
    log_driver        = "json-file"
    log_opts          = {}
```



```
    logs              = false
    max_retry_count   = 0
    memory            = 0
    memory_swap       = 0
    must_run          = true
    name              = "nginx-server-2"
    network_data      = [
        {
            gateway          = "172.18.0.1"
            ip_address       = "172.18.0.2"
            ip_prefix_length = 24
            network_name     = "bridge"
        },
    ]
    network_mode      = "default"
    privileged        = false
    publish_all_ports = false
    read_only         = false
    restart           = "no"
    rm                = false
    shm_size          = 64
    start             = true
```

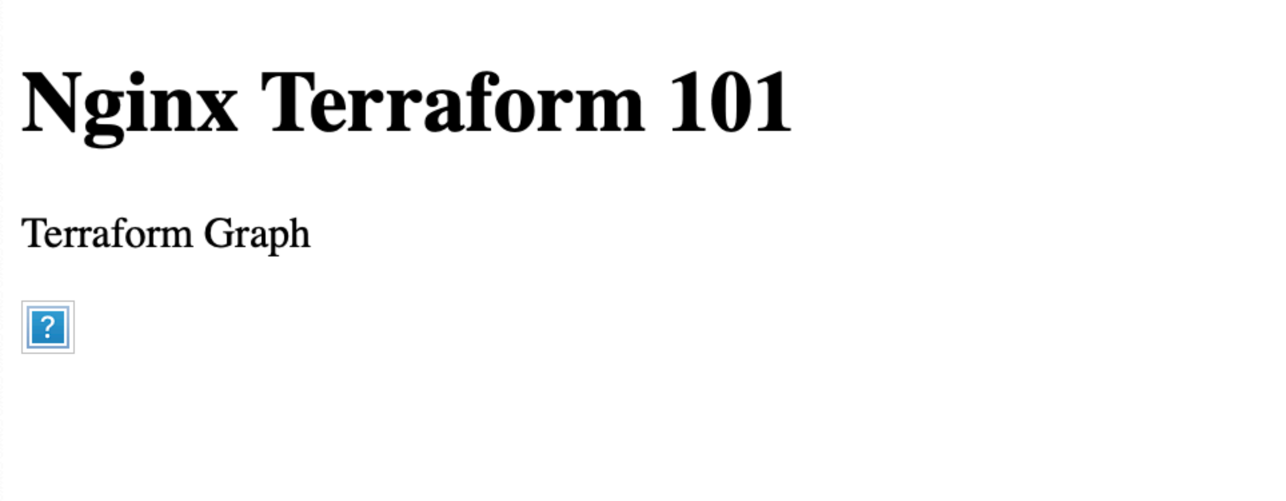


```
        },
    ]
    network_mode      = "default"
    privileged        = false
    publish_all_ports = false
    read_only         = false
    restart           = "no"
    rm                = false
    shm_size          = 64
    start             = true

    ports {
        external = 32769
        internal = 80
        ip       = "0.0.0.0"
        protocol = "tcp"
    }

    volumes {
        container_path = "/usr/share/nginx/html"
        host_path      = "/home/terraform101/www"
        read_only      = true
    }
```



```
    ports {
        external = 32768
        internal = 80
        ip       = "0.0.0.0"
        protocol = "tcp"
    }

    volumes {
        container_path = "/usr/share/nginx/html"
        host_path      = "/home/terraform101/www"
        read_only      = true
    }
}

# docker_image.nginx:
resource "docker_image" "nginx" {
    id     = "sha256:bedece1f06cc142829698e6ba8f04d7f92e7f1b94b985e14b65f55e6ae4858c2nginx:1.11-alpine"
    latest = "sha256:bedece1f06cc142829698e6ba8f04d7f92e7f1b94b985e14b65f55e6ae4858c2"
    name   = "nginx:1.11-alpine"
}
$ 
```

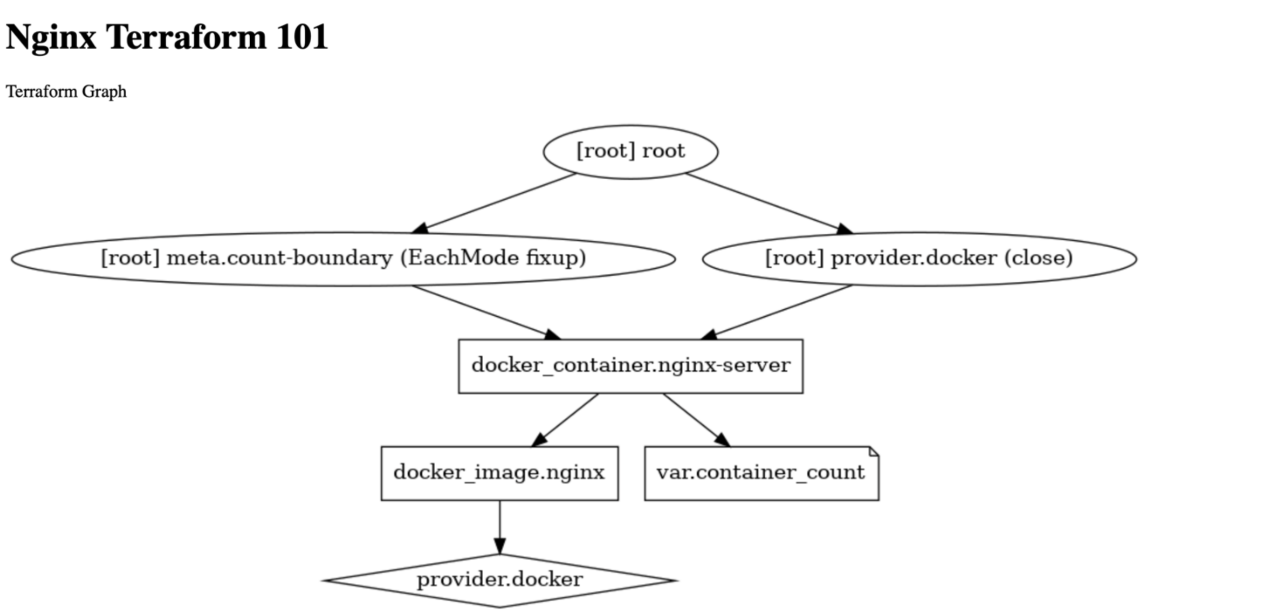
C. Capture Screen output ของหน้า Nginx ใน Step 9



Nginx Terraform 101

Terraform Graph

D. Capture Screen output ของหน้า Nginx หลังจากสร้าง Terraform Graph ใน Step 11

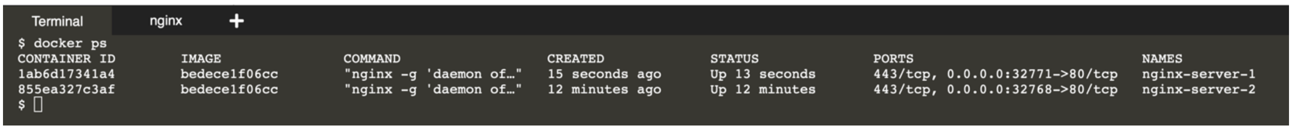


Nginx Terraform 101

Terraform Graph

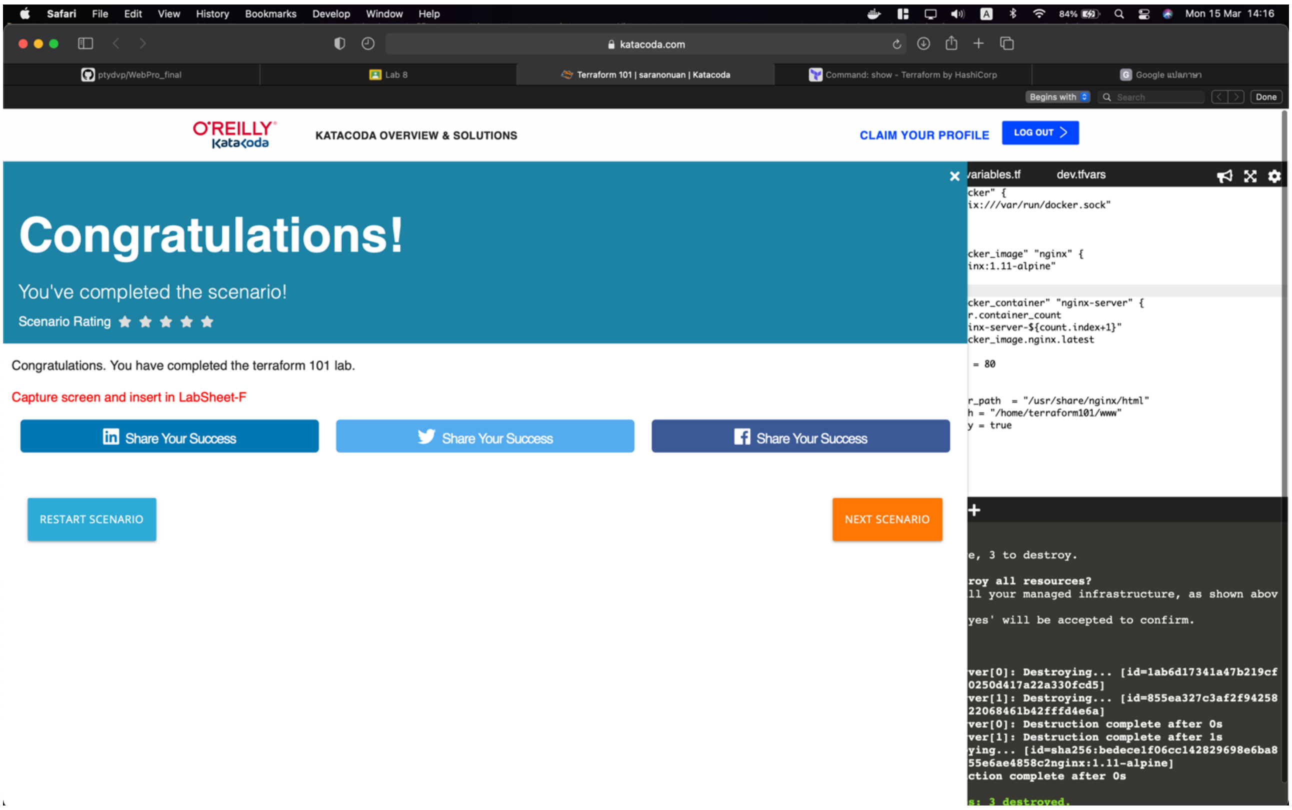
[root] root
[root] meta.count-boundary (EachMode fixup)
[root] provider.docker (close)
docker_container.nginx-server
docker_image.nginx
var.container_count
provider.docker

E. Capture Screen output เมื่อทำการ apply จำนวน Container ใหม่เสร็จแล้ว และรัน docker ps ใน Step 12



```
$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                              NAMES
1ab6d17341a4        bedece1f06cc        "nginx -g 'daemon of…"   15 seconds ago      Up 13 seconds       443/tcp, 0.0.0.0:32771->80/tcp     nginx-server-1
855ea327c3af        bedece1f06cc        "nginx -g 'daemon of…"   12 minutes ago      Up 12 minutes       443/tcp, 0.0.0.0:32768->80/tcp     nginx-server-2
$ 
```

F. Capture Screen หน้า Finish หลัง Quiz เสร็จ


Safari File Edit View History Bookmarks Develop Window Help
84% Mon 15 Mar 14:16
katacoda.com
ptydvp/WebPro_final
Lab 8
Terraform 101 | saranonuan | Katacoda
Command: show - Terraform by HashiCorp
Google แปลภาษา
Begins with
Search
Done
O'REILLY Katacoda
KATACODA OVERVIEW & SOLUTIONS
CLAIM YOUR PROFILE
LOG OUT
variables.tf
dev.tfvars

Congratulations!

You've completed the scenario!

Scenario Rating ★ ★ ★ ★ ★

Congratulations. You have completed the terraform 101 lab.

Capture screen and insert in LabSheet-F

```
cker" {
ix:///var/run/docker.sock"


cker_image" "nginx" {
inx:1.11-alpine"


cker_container" "nginx-server" {
r.container_count
inx-server-${count.index+1}"
cker_image.nginx.latest

 = 80


r_path  = "/usr/share/nginx/html"
h = "/home/terraform101/www"
y = true
```

```
e, 3 to destroy.

roy all resources?
ll your managed infrastructure, as shown abov

yes' will be accepted to confirm.



ver[0]: Destroying... [id=1ab6d17341a47b219cf
0250d417a22a330fcd5]
ver[1]: Destroying... [id=855ea327c3af2f94258
22068461b42fffd4e6a]
ver[0]: Destruction complete after 0s
ver[1]: Destruction complete after 1s
ying... [id=sha256:bedece1f06cc142829698e6ba8
55e6ae4858c2nginx:1.11-alpine]
ction complete after 0s

s: 3 destroyed.
```