

# Terraform dla Administratora produktów VMware Zaczynamy przygodę!

Dla znajomych Seba



HashiCorp

Terraform

exeaa<sup>®</sup>

vmware<sup>®</sup>

[AkademiaDatacenter.PL/vmug](https://AkademiaDatacenter.PL/vmug) → dodatkowe mat.+ losowanie na koniec

# MATERIAŁY

[AkademiaDatacenter.pl/vmug](https://AkademiaDatacenter.pl/vmug)

- Uczestnictwo w losowaniu na końcu prelekcji
- Dostęp do prezentacji
- Link do githuba z przykładami
- Link na tajnej grupie nauki na FB do pierwszych kroków z Terraformem dla Vmware NINJAs
- Dostęp do postów na AkademiaDatacenter.pl w wersji DRAFT (o terraformie)

# Sebastian Grugel

- Senior VMware Engineer at the EXEA datacenter Toruń
- VMware + cloud computing specialization
- VMware VCP / VCAP + vExpert \*\*\*\* certifications
- Husband and father of 3 boys
  
- Cloud Datacenter User Group (leader)
- Google Cloud Developer Community (leader)
- bITConf.pl - CLOUD program board
- AkademiaDatacenter.pl – 7 lat

Początkujący w terraformie



sebastian.grugel@gmail.com



# Agenda

- Zanim o Terraformie....
- Co to jest Terraform ?
- Do czego został stworzony ?
- HCL
- Podstawowe komponenty
- Podstawowe komendy
- DEMO – Task1
- DEMO – Task2
- Pytania , szybkie losowanie 😊

# Zanim o Terraformie....

- Z czego może składać się infrastruktura ?
- Czym jest KOD ?
- Typy kodów:
  - Imperatywny
  - deklaratywny

No to co to je ten... terra..form...

- IaC - Infrastructure as code software od HashiCorp
- Go
- Ostatnia stabilna wersja 0.11.14
- Modułowość (providers)

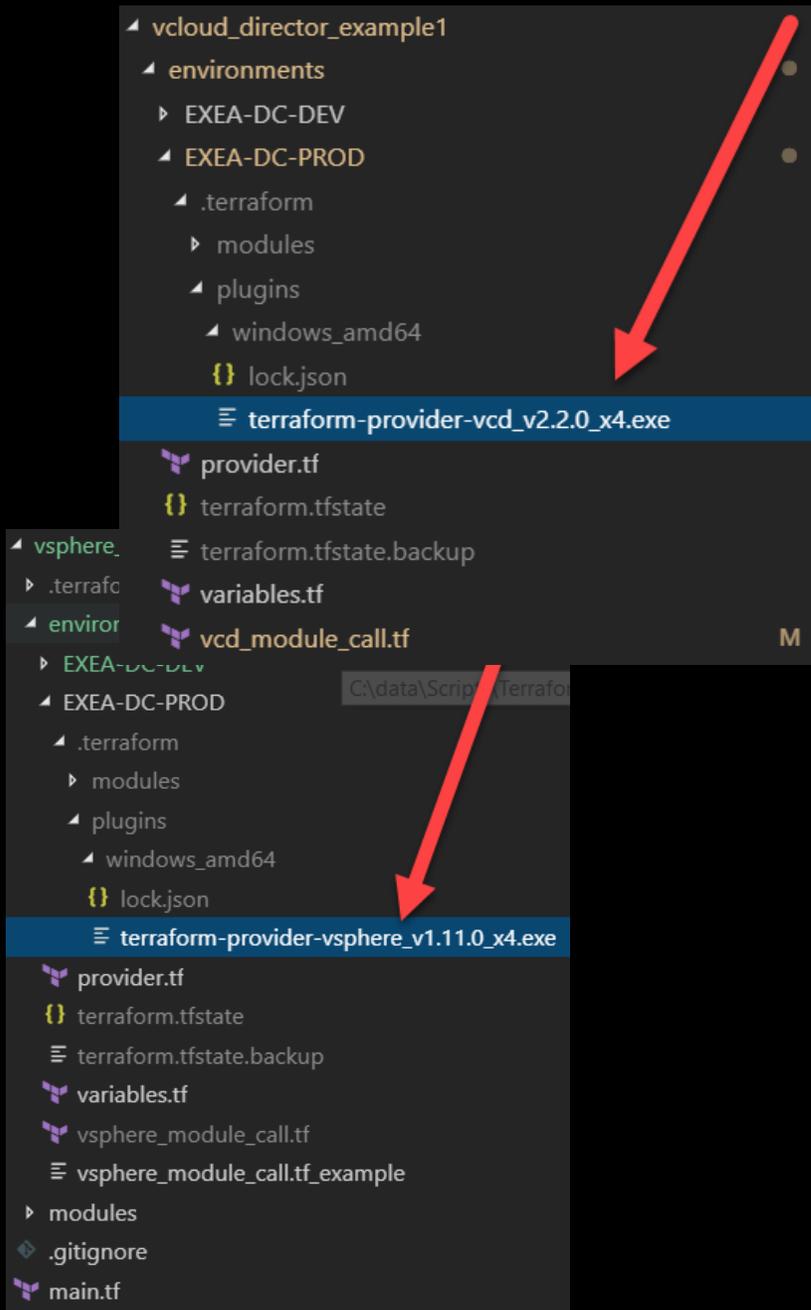
# Do czego został stworzony ?

- Budowania infrastruktury
- Zmiany Infrastruktury
- Wersjonowanie Infrastruktury

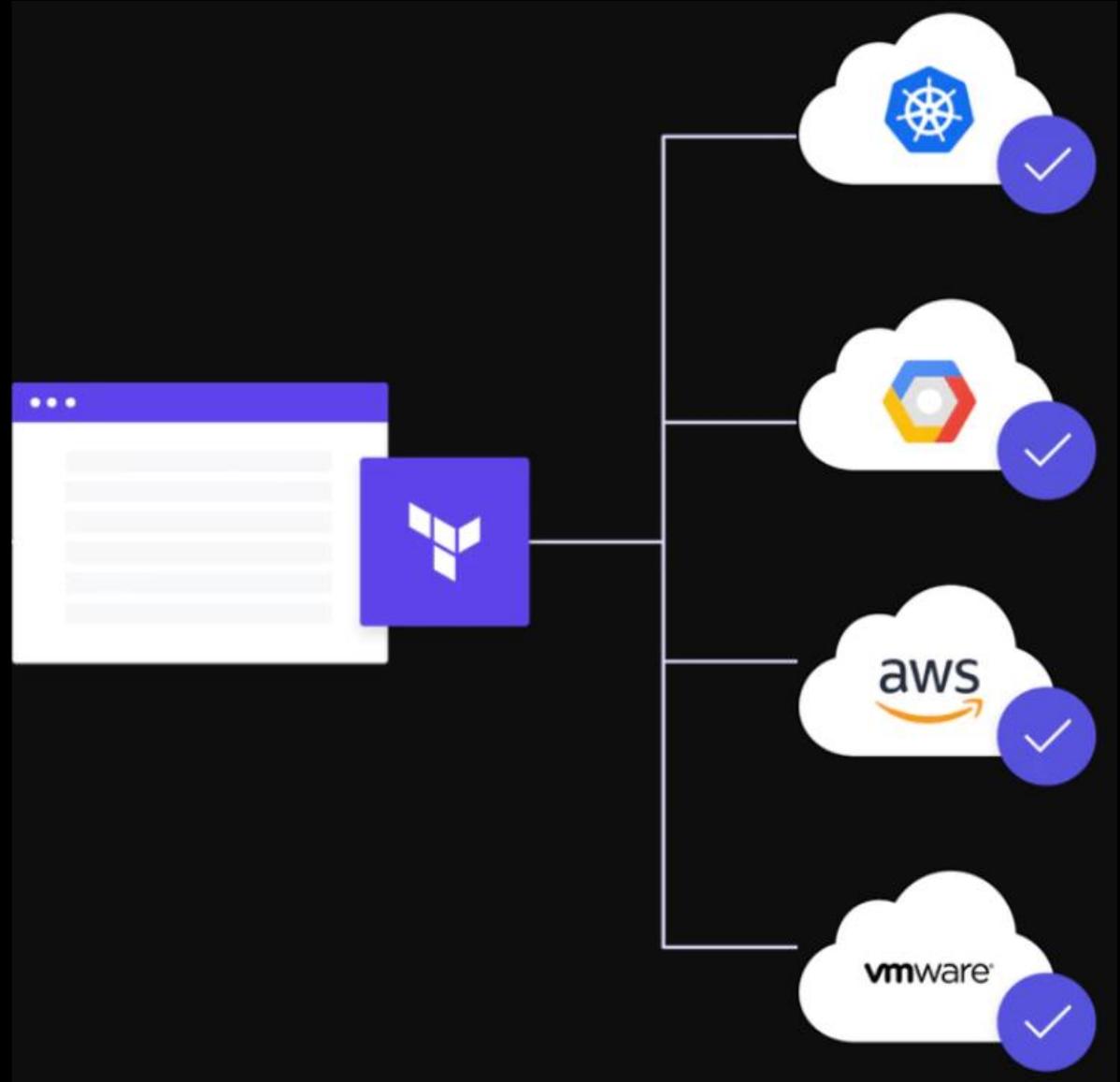
# HCL – Hashicorp Configuration Language

- Wspiera inne produkty Hashicorp
- Wspiera JSONa
- Podobno jest całkiem czytelny :D
- Komentarze
- Przykład:

```
variable "hosts" {  
  default = []  
  type = "list"  
}
```



# PROVIDER



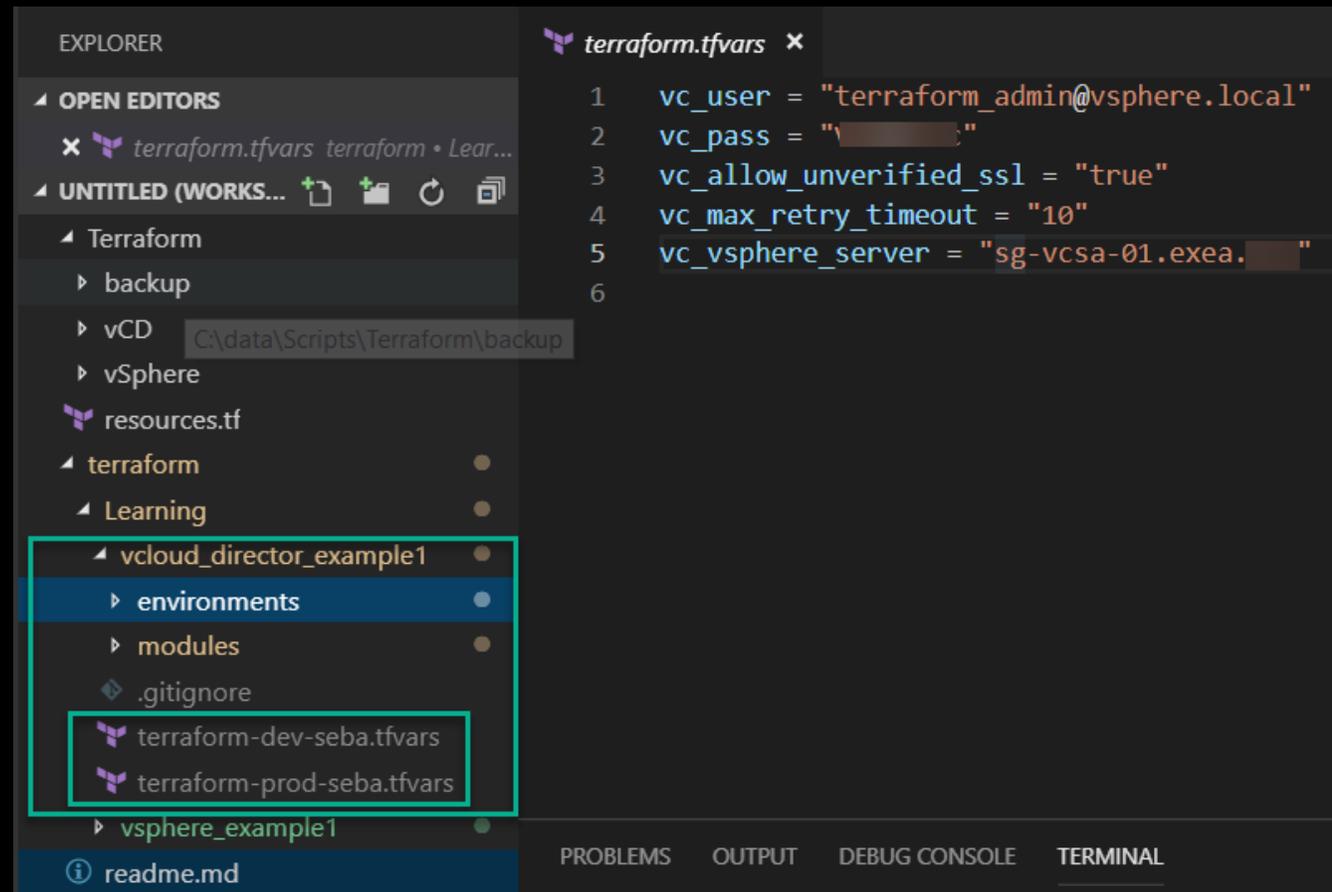
# Statefile

- Domyślnie: plik o nazwie „terraform.tfstate”
- Może być przechowywany zdalnie: np. na S3 bucket.
- Przechowuje aktualne informacje o infrastrukturze
- Służy do tworzenia planów i wprowadzania zmian w infrastrukturze
- Format JSON
- Nie edytuj bezpośrednio, użyj polecenia **terraform state**, aby manipulować tym plikiem

```
{} terraform.tfstate x
1  {
2    "version": 3,
3    "terraform_version": "0.11.14",
4    "serial": 23,
5    "lineage": "3df4442c-5646-9bb2-a21",
6    "modules": [
7      {
8        "path": [
9          "root"
10       ],
11       "outputs": {},
12       "resources": {},
13       "depends_on": []
14     },
15     {
16       "path": [
17         "root",
18         "dev_environment"
19       ],
20       "outputs": {},
21       "resources": {},
22       "depends_on": []
23     },
24     {
25       "path": [
26         "root",
```

# Pliki konfiguracyjne

- Pliki tekstowe
- Rozszerzenie .tf
- Dodatkowe pliki .tfvars – na dane „wrażliwe” .tfvars
- Napisane w HashiCorp Configuration Language



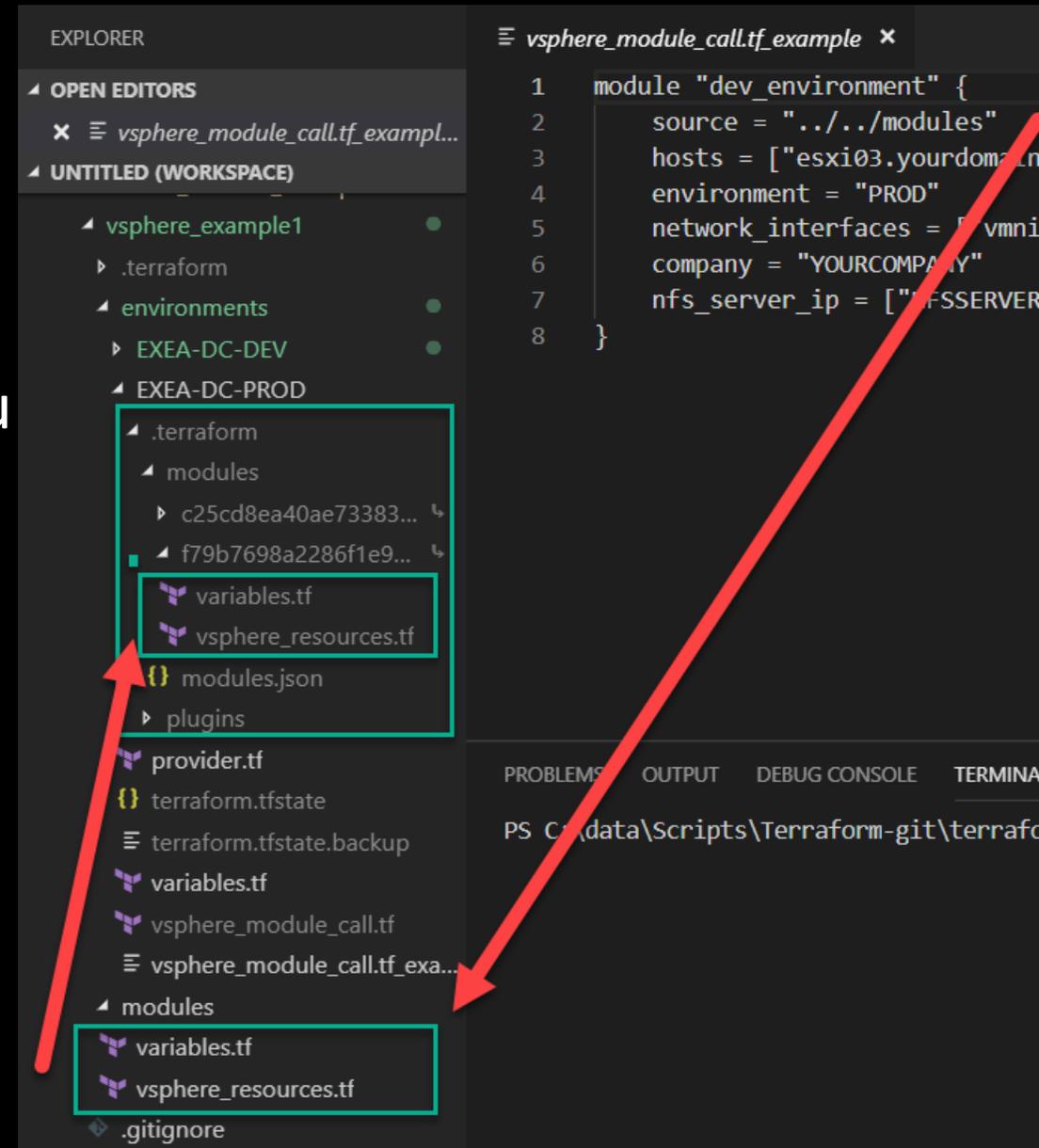
The screenshot shows the Visual Studio Code interface. On the left, the Explorer view displays a file tree for a Terraform project. The 'vcloud\_director\_example1' folder is expanded, showing subfolders 'environments' and 'modules', a '.gitignore' file, and two variable files: 'terraform-dev-seba.tfvars' and 'terraform-prod-seba.tfvars'. The 'terraform-prod-seba.tfvars' file is selected and highlighted with a red box. On the right, the editor shows the content of the selected file, 'terraform.tfvars', which contains six lines of configuration in HashiCorp Configuration Language (HCL):

```
1 vc_user = "terraform_admin@vsphere.local"
2 vc_pass = "XXXXXXXXXX"
3 vc_allow_unverified_ssl = "true"
4 vc_max_retry_timeout = "10"
5 vc_vsphere_server = "sg-vcsa-01.exea.XXXXXX"
6
```

At the bottom of the interface, the status bar shows tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL'.

# terraform init

- Inicjalizuje katalog roboczy
- Pobiera informacje o providerach
- Pobiera moduły TF do ukrytego katalogu
- Inicjuje backend ( dla terraform.tfstate)  
➔ nie testowałem



# terraform plan

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
+ module.dev_environment.vsphere_folder.teamfolder[4]
  id: <computed>
  datacenter_id: "datacenter-2"
  path: "Teams/Team-5"
  type: "vm"

+ module.dev_environment.vsphere_folder.teams
  id: <computed>
  datacenter_id: "datacenter-2"
  path: "Teams"
  type: "vm"

+ module.dev_environment.vsphere_nas_datastore.nfsdatastore
  id: <computed>
  access_mode: "readwrite"
  accessible: <computed>
  capacity: <computed>
  free_space: <computed>
  host_system_ids.#: "2"
  host_system_ids.1622726607: "host-141"
  host_system_ids.411055375: "host-134"
  maintenance_mode: <computed>
  multiple_host_access: <computed>
  name: "Datastore-DEV-NFS"
  protocol_endpoint: <computed>
  remote_hosts.#: "1"
  remote_hosts.0: "██████████"
  remote_path: "/mnt/nfs/nfs1"
  type: "NFS"
  uncommitted_space: <computed>
  url: <computed>

+ module.dev_environment.vsphere_resource_pool.resource_pool_mgmt
  id: <computed>
  cpu_expandable: "true"
  cpu_limit: "-1"
  cpu_reservation: "0"
  cpu_share_level: "normal"
  cpu_shares: <computed>
  memory_expandable: "true"
```

# terraform apply

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
wait_for_guest_ip_timeout: "0"  
wait_for_guest_net_routable: "true"  
wait_for_guest_net_timeout: "0"
```

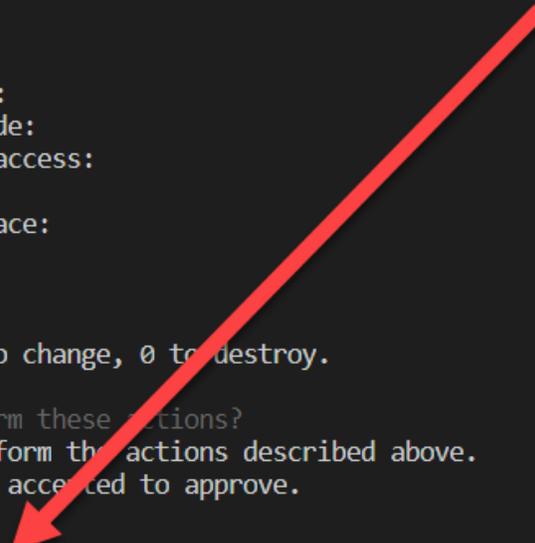
```
+ module.dev_environment.vsphere_vmfs_datastore.datastore1  
  id: <computed>  
  accessible: <computed>  
  capacity: <computed>  
  disks.#: "3"  
  disks.0: "naa.6000c291"  
  disks.1: "naa.6000c296"  
  disks.2: "naa.6000c29c"  
  free_space: <computed>  
  host_system_id: "host-141"  
  maintenance_mode: <computed>  
  multiple_host_access: <computed>  
  name: "Datastore-sg"  
  uncommitted_space: <computed>  
  url: <computed>
```

```
+ module.dev_environment.vsphere_vmfs_datastore.datastore2  
  id: <computed>  
  accessible: <computed>  
  capacity: <computed>  
  disks.#: <computed>  
  free_space: <computed>  
  host_system_id: "host-134"  
  maintenance_mode: <computed>  
  multiple_host_access: <computed>  
  name: "Datastore-sg"  
  uncommitted_space: <computed>  
  url: <computed>
```

Plan: 35 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:



# terraform destroy

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
- module.dev_environment.vsphere_nas_datastore.nfsdatastore
- module.dev_environment.vsphere_resource_pool.resource_pool_mgmt
- module.dev_environment.vsphere_resource_pool.resource_pool_priority
- module.dev_environment.vsphere_resource_pool.resource_pool_regular
- module.dev_environment.vsphere_virtual_machine.vm[0]
- module.dev_environment.vsphere_virtual_machine.vm[1]
- module.dev_environment.vsphere_virtual_machine.vm[2]
- module.dev_environment.vsphere_virtual_machine.vm[3]
- module.dev_environment.vsphere_virtual_machine.vm[4]
- module.dev_environment.vsphere_virtual_machine.vm[5]
- module.dev_environment.vsphere_virtual_machine.vm[6]
- module.dev_environment.vsphere_virtual_machine.vm[7]
- module.dev_environment.vsphere_virtual_machine.vm[8]
- module.dev_environment.vsphere_virtual_machine.vm[9]
- module.dev_environment.vsphere_vmfs_datastore.datastore1
- module.dev_environment.vsphere_vmfs_datastore.datastore2
```

Plan: 0 to add, 0 to change, 35 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:

# Task1: vCD – onboarding klienta

## Stan obecny (klient ma):

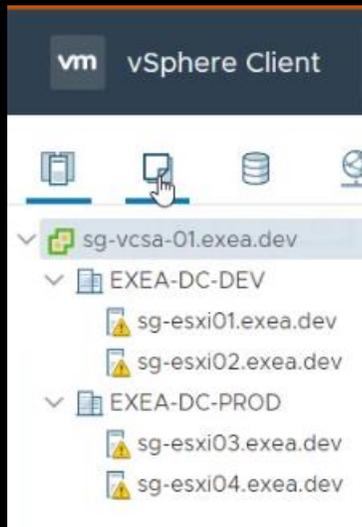
- własną organizację w vCD
- NSX EDGE
- VDC
- 1 domyślna sieć

## Stan docelowy (klient ma):

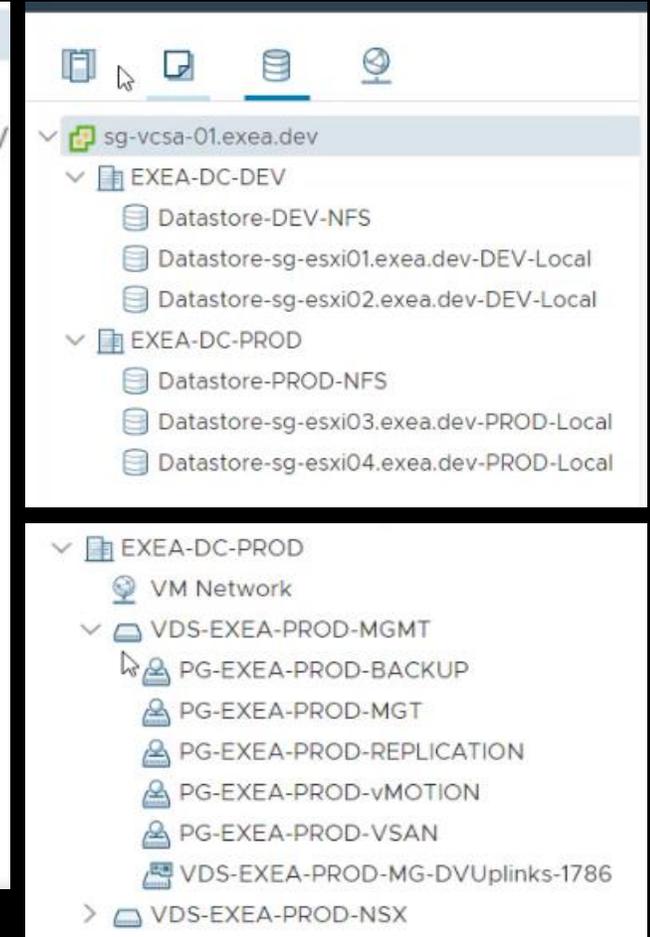
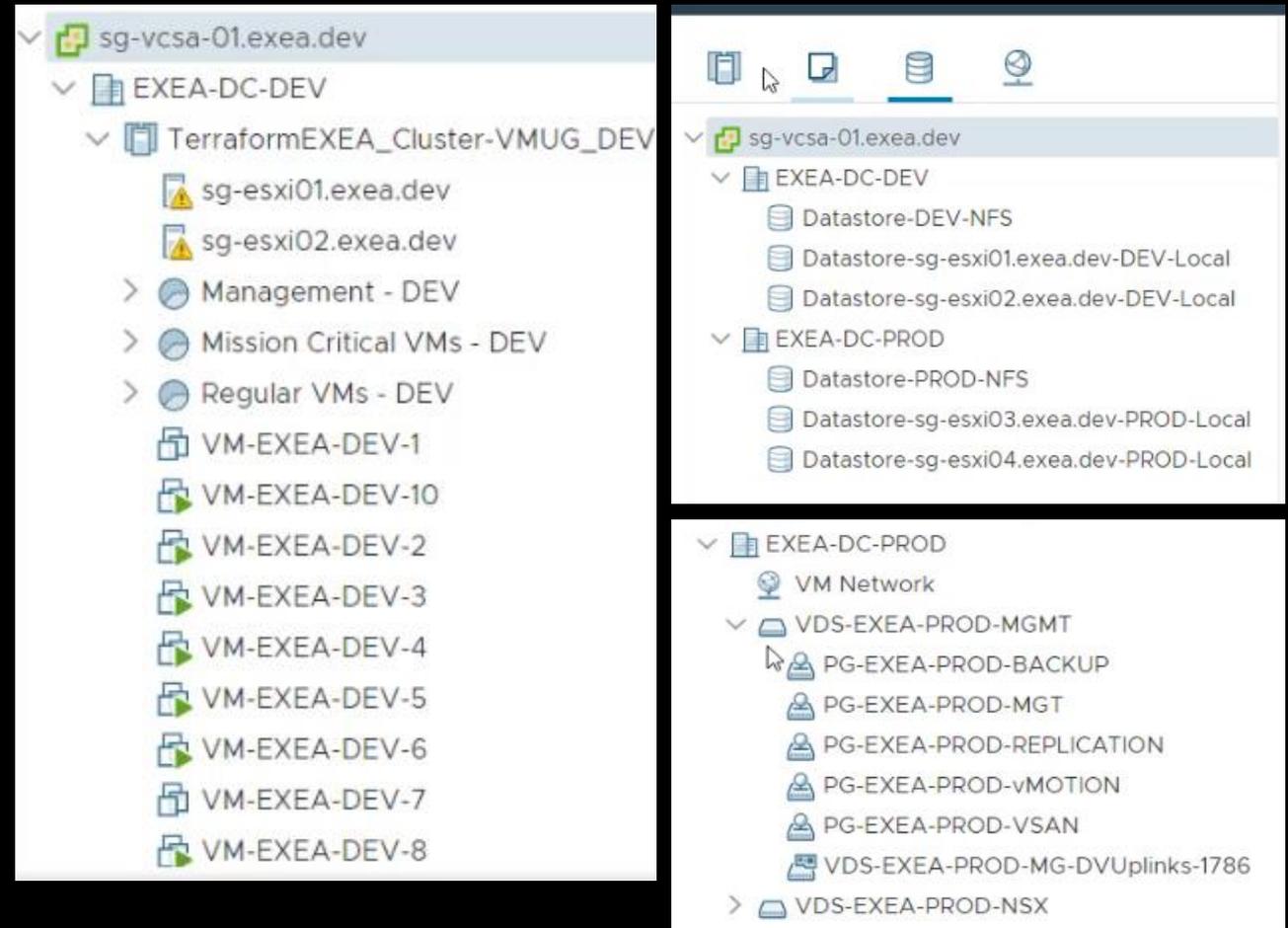
- własną organizację w vCD
- NSX EDGE
- VDC
- 1 domyślna sieć
- 1 dodatkową sieć
- 1 vApp
- 3 maszyny gotowe do startu
- Zdefiniowane rule FW
- Zdefiniowane NATy
- Dodane własne katalogi w bibliotece

# Task2: Konfiguracja vSphere na życzenie

Stan obecny (klient ma):



Stan docelowy (klient ma):



# Źródła:

[terraform.io](https://terraform.io)

[vSphere provider dokumentacja](#)

[vCloudDirector provider dokumentacja](#)

<https://vcloudvision.com/2019/04/12/provision-vsphere-vm-s-using-terraform>

<https://blogs.vmware.com/vcloud/2019/04/vcloud-director-terraform.html>

<https://anthonyspiteri.net/infrastructure-as-code-vs-restful-apis-terraform-and-everything-in-between/>

<https://anthonyspiteri.net/infrastructure-as-code-vs-restful-apis-a-working-example-with-terraform-and-vcloud-director>

<https://www.virtualtothecore.com/learning-how-to-use-terraform-in-vcloud-director/>

<https://mycloudrevolution.com/2018/01/02/vmware-vcloud-director-provider-for-terraform/>

<https://www.consdata.com/pl/blog/terraform-czyli-o-tym-jak-okielznac-chmure-od-amazona>

<https://medium.com/@amolkokje/terraform-vsphere-virtual-machines-limitations-57621a73019a>

<https://akademiadatacenter.pl/tag/terraform/>

<https://github.com/hashicorp/terraform/blob/v0.12.0/CHANGELOG.md> nowa wersja terraforma 22.05.2019r.

W sumie to już wszystko 😊

**Pytania ?  
Sugestie ?**

Wpisaleś się na [AkademiaDatacenter.pl/vmug](https://AkademiaDatacenter.pl/vmug) ? Za chwile losowanie 😊