

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN – ĐHQG TP.HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ PHẦN MỀM



TÀI LIỆU HƯỚNG DẪN TRIỂN KHAI HỆ THỐNG
ỨNG DỤNG LUYỆN THI IELTS CHO HỆ ĐIỀU HÀNH ANDROID

GVHD:

TS. Ngô Huy Biên

Sinh viên thực hiện:

19127626 Lê Nguyễn Tú Văn

19127631 Dương Tiến Vinh

19127373 Hồ Văn Duy

19127294 Nguyễn Trần Thiện Toàn

19127607 Trần Nguyên Trung

19127597 Trần Khả Trí

Tp. Hồ Chí Minh, tháng 7 năm 2023

Mục lục

Hướng dẫn triển khai hệ thống trên môi trường Azure	3
Tổng quan	3
Công cụ	3
Cài đặt công cụ	4
Triển khai hệ thống	5
Tổng quan	5
Khởi tạo Azure Kubernetes Service (AKS) cluster	5
Cài đặt helm chart	8
Cài đặt và cấu hình vault	8
Cài đặt và cấu hình ArgoCD	9
Cài đặt hệ thống tự động	9
Cấu hình DNS:	9

Hướng dẫn triển khai hệ thống trên môi trường Azure

Tổng quan

Hệ thống Kubernetes của nhóm được triển khai trên môi trường Azure với dịch vụ Azure Kubernetes Service (AKS). Nhóm sẽ triển khai hệ thống tự động với công cụ Terraform.

Công cụ

- Hệ thống được kiểm tra & triển khai với cấu hình như sau:
 - o Hệ điều hành: Ubuntu 22.10.
 - o Terminal: zsh 5.9 (x86_64-ubuntu-linux-gnu).
- Công cụ (terminal):
 - o kubectl: Công cụ để triển khai Kubernetes resource

```
$ kubectl version --short
Client Version: v1.27.3
Kustomize Version: v5.0.1
Server Version: v1.26.4+k3s1
```

- o kustomize: Công cụ để cài đặt resource

```
$ kustomize version
v5.0.3
```

- o terraform: Công cụ để triển khai hệ thống tự động

```
$ terraform --version
Terraform v1.5.2
on linux_amd64
```

- o azure-cli: Công cụ để giao tiếp với Azure

```
$ az --version
azure-cli          2.50.0

core               2.50.0
telemetry          1.0.8

Dependencies:
msal               1.22.0
azure-mgmt-resource 23.1.0b2
```

- o helm: Công cụ để quản lý Kubernetes resource

```
$ helm version
version.BuildInfo{Version:"v3.12.1", GitCommit:"f32a527a060157990e2aa86bf45010dfb3cc8b8d", GitTreeState:"clean",
GoVersion:"go1.20.4"}
```

o helmfile: Công cụ để quản lý helm chart

```
$ helmfile version
Version      0.154.0
Git Commit   c498af3
Build Date   24 May 23 06:31 +07 (1 month ago)
Commit Date  24 May 23 06:29 +07 (1 month ago)
Dirty Build   no
Go version    1.20.4
Compiler     gc
Platform     linux/amd64
```

o argocd: Công cụ để giao tiếp với ArgoCD server

```
$ argocd version
argocd: v2.7.5+a2430af
BuildDate: 2023-06-16T15:00:03Z
GitCommit: a2430af1c356b283e5e3fc5bde1f5e2b5199f258
GitTreeState: clean
GoVersion: go1.19.10
Compiler: gc
Platform: linux/amd64
```

o Các công cụ phụ: curl, wget, jq, gpg.

Cài đặt công cụ

Clone repo gitops về máy:

```
$ git clone https://github.com/the-company-tcus/gitops.git
```

Chuyển đến thư mục scripts/install-tools:

```
$ cd scripts/install-tools
```

Cài đặt tất cả các công cụ:

```
$ ./setup.sh
```

Cài đặt công cụ thủ công:

- Cài đặt kubectl:

```
$ ./kubectl.setup.sh
```

- Cài đặt kustomize:

```
$ ./kustomize.setup.sh
```

- Cài đặt terraform:

```
$ ./terraform.setup.sh
```

- Cài đặt azure-cli:

```
$ ./azure.setup.sh
```

- Cài đặt helm:

```
$ ./helm.setup.sh
```

- Cài đặt helmfile:

```
$ ./helmfile.setup.sh
```

- Cài đặt argocd:

```
$ ./argocd.setup.sh
```

Triển khai hệ thống

Tổng quan

Để triển khai nhóm sẽ làm theo các bước sau:

1. Tạo Azure Kubernetes Service (AKS) cluster bằng terraform.
2. Cài đặt helm chart bằng helmfile.
3. Cài đặt và cấu hình vault.
4. Cài đặt và cấu hình argocd.

Khởi tạo Azure Kubernetes Service (AKS) cluster

1. Đăng nhập vào Azure:

```
$ az login
```

Người dùng sẽ được yêu cầu truy cập vào đường dẫn và bắt đầu quá trình đăng nhập.

2. Truy cập thông tin của account:

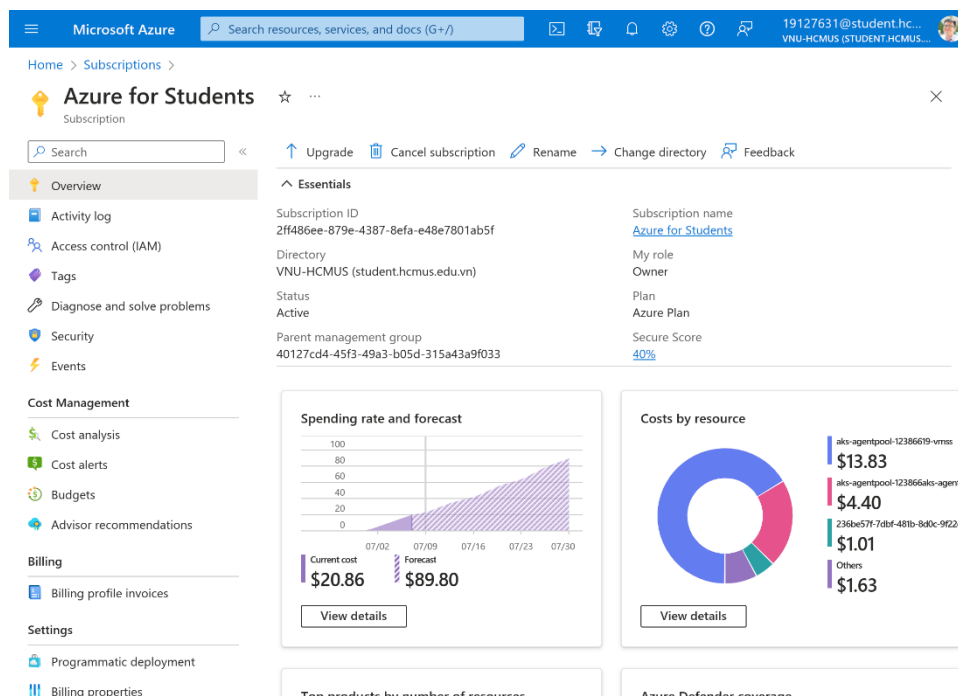
Người dùng sẽ cần hai thông tin sau để xác thực với Terraform:

- Subscription ID: ID của subscription mà người dùng muốn triển khai hệ thống lên.

```
$ az account subscription list
```

```
[
  {
    "authorizationSource": "RoleBased",
    "displayName": "Azure for Students",
    "id": "/subscriptions/2ff486ee-879e-4387-8efa-e48e7801ab5f",
    "state": "Enabled",
    "subscriptionId": "2ff486ee-879e-4387-8efa-e48e7801ab5f",
    "subscriptionPolicies": {
      "locationPlacementId": "Public_2014-09-01",
      "quotaId": "AzureForStudents_2018-01-01",
      "spendingLimit": "On"
    }
  }
]
```

- Hoặc có thể truy cập vào Home/Subscriptions/<SUBSCRIPTION-NAME> để lấy Subscription ID.



- Tenant ID: ID của tenant mà người dùng muốn triển khai hệ thống lên.

```
$ az account show
```

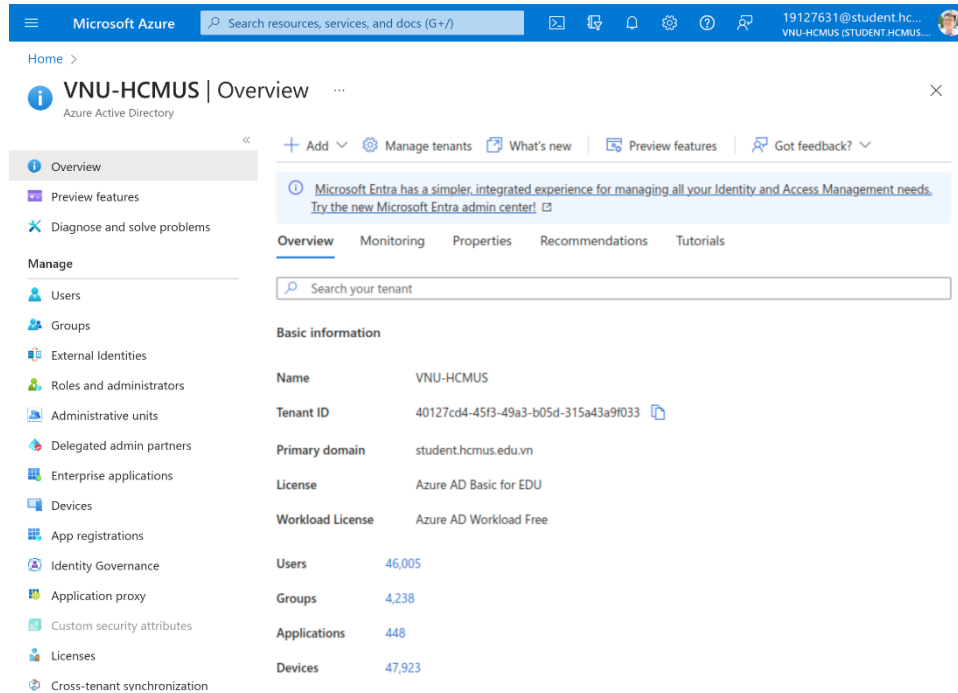
```
{
  "environmentName": "AzureCloud",
  "homeTenantId": "40127cd4-45f3-49a3-b05d-315a43a9f033",
  "id": "2ff486ee-879e-4387-8efa-e48e7801ab5f",
  "isDefault": true,
  "managedByTenants": [],
  "name": "Azure for Students",
}
```

```

"state": "Enabled",
"tenantId": "40127cd4-45f3-49a3-b05d-315a43a9f033",
"user": {
  "name": "19127631@student.hcmus.edu.vn",
  "type": "user"
}
}

```

- Hoặc có thể truy cập vào Home/Azure Active Directory để lấy Tenant ID.



3. Cập nhập thông tin xác thực cho Terraform:

Người triển khai sẽ cập nhật hai thông tin subscription_id và tenant_id vào file terraform.tfvars trong thư mục terraform/azure:

```

subscription_id = "2ff486ee-879e-4387-8efa-e48e7801ab5f"
tenant_id       = "40127cd4-45f3-49a3-b05d-315a43a9f033"

```

```

user_principal_names = [
  "19127626@student.hcmus.edu.vn",
  "19127373@student.hcmus.edu.vn",
  "19127294@student.hcmus.edu.vn",
  "19127607@student.hcmus.edu.vn",
  "19127597@student.hcmus.edu.vn",
]

```

Ngoài ra, người triển khai cần cập nhật thông tin user_principal_names với email của các thành viên trong nhóm.

4. Khởi tạo Terraform:

```
$ terraform init
```

5. Kiểm tra các tài nguyên sẽ được tạo ra:

```
$ terraform plan
```

Terraform sẽ tạo ra các tài nguyên sau:

- Resource Group: ethi-group.
 - AKS Cluster: ethi
 - Node pool: agentpool.
 - Node Count: 1.
 - VM Size: Standard_B2ms.
 - Kubernetes Version: 1.26.3.
 - Role Assignment:
 - Role Definition Name: Azure Kubernetes Service RBAC Cluster Admin.
 - Principal ID: Các Object ID của thành viên trong nhóm.
6. Triển khai hệ thống:

```
$ terraform apply
```

Cài đặt helm chart

Cài đặt các helm chart bằng helmfile:

```
$ helmfile sync
```

Cài đặt và cấu hình vault

1. Khởi tạo vault:

```
$ kubectl -n vault exec vault-0 -- vault operator init -key-shares=1 -key-threshold=1 -format=json > cluster-keys.json
```

Vault sẽ được khởi tạo với số lượng key-share là 1 và key-threshold là 1. File cluster-keys.json sẽ chứa thông tin về unseal key và root token của vault.

2. Đăng nhập vào vault

```
$ kubectl -n vault exec vault-0 -- vault login
```

Người triển khai sẽ được yêu cầu nhập root_token trong file cluster-keys.json để đăng nhập vào vault.

3. Tạo secret path


```
$ kubectl -n vault exec vault-0 -- vault secrets enable -path=secret kv-v2
```

Vault sẽ tạo ra một secret path có tên là secret để lưu trữ các secret.

4. Tạo Kubernetes secret để lưu root token

```
$ kubectl -n vault create secret generic vault-token \
--from-literal=token=<ROOT-TOKEN>
```

Người triển khai sẽ cập nhật <ROOT-TOKEN> bằng root_token trong file cluster-keys.json.

5. Import secret vào vault:

```
$ cd scripts/vault && make import-prod token="<ROOT-TOKEN>"
```

Người triển khai sẽ cập nhật <ROOT-TOKEN> bằng root_token trong file cluster-keys.json.

Vault sẽ được import các secret từ file scripts/vault/import-secret-prod.yaml vào secret path secret, với công cụ [medusa](#) CLI.

Note: Target make import-prod sẽ yêu cầu người triển khai đã cài đặt Docker.

Cài đặt và cấu hình ArgoCD

1. Cài đặt ArgoCD với Kustomize:

```
$ kubectl apply -k ./registry/ethi/components/argocd
```

2. Cài đặt các ứng dụng của ArgoCD:

```
$ kubectl apply -f ./registry/ethi
```

Cài đặt hệ thống tự động

Người triển khai có thể dùng file script của nhóm để cài đặt hệ thống tự động các bước:

2. Cài đặt helm chart bằng helmfile.

3. Cài đặt và cấu hình vault.

4. Cài đặt và cấu hình argocd.

Người triển khai có thể chạy file script bootstrap-prod.sh để cài đặt hệ thống:

```
$ ./bootstrap-prod.sh
```
















Cấu hình DNS:

Chạy lệnh sau để lấy Public IP của Traefik Load Balancer, địa chỉ này sẽ được dùng để cấu hình DNS:

```
$ kubectl -n kube-system get svc
```

```
1: ~ x +
~ kubectcl -n kube-system get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S)
AGE
kube-dns ClusterIP 10.0.0.10 <none> 53/UDP,53/TCP
23d
metrics-server ClusterIP 10.0.39.108 <none> 443/TCP
23d
traefik LoadBalancer 10.0.115.135 20.81.118.174 7432:30426/TCP,7434:30537/TCP,7433:30385/TCP,80:30493/TCP,443:32635/TCP
23d
~ ok 08:17:43 PM
```

Sau đó cấu hình DNS như hình:

Type ▲	Name	Content	Proxy status	TTL	Actions
A	 ethi-team.tech	20.81.118.174	 DNS only	Auto	Edit 
CNAME	admin	ethi-team.tech	 DNS only	Auto	Edit 
CNAME	api	ethi-team.tech	 DNS only	Auto	Edit 
CNAME	argocd	ethi-team.tech	 DNS only	Auto	Edit 
CNAME	docs	the-company-tcus.netlify.app	 Proxied	Auto	Edit 
CNAME	em9550	u32548810.wl061.sendgrid.net	DNS only	Auto	Edit 
CNAME	keycloak	ethi-team.tech	 DNS only	Auto	Edit 
CNAME	vault	ethi-team.tech	 DNS only	Auto	Edit 