

Elwin Bassaget

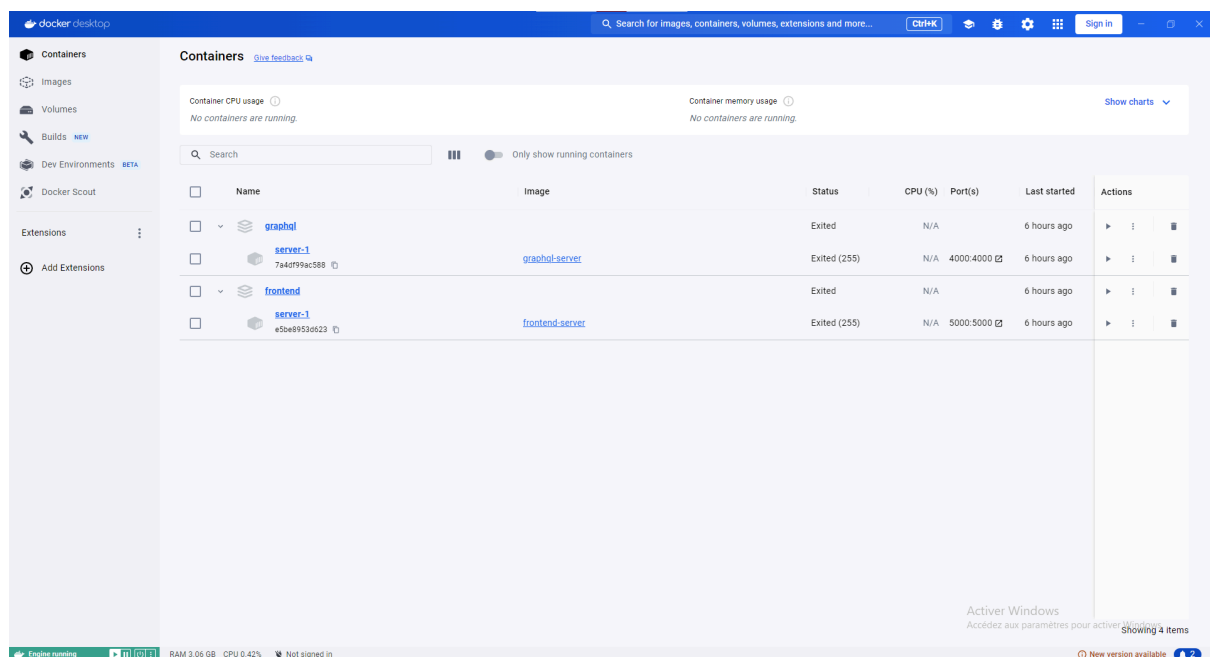
Projet Cloud-AWS

Documentation du projet

Docker : Conteneurisation

Pour partir sur ce projet je suis parti sur un projet que j'ai fait cette année, un projet GraphQL en Backend codé en TS avec une base de donnée Sqlite, et en Frontend un projet en Vue 3 (typé TS). J'ai rencontré notamment une difficulté avec la conteneurisation de l'application GraphQL.

J'ai mis en place deux docker-compose pour pouvoir les déployer, séparant les deux parties.



Orchestration avec Docker Swarm

J'ai manipulé des Docker Swarm avec multipass et sans, mais je n'ai pas compris le but final de l'utilisation de ce procédé, ou plutôt je n'ai pas compris l'aboutissement final que cela peut donner.

AWS EC2 : Préparation du Déploiement avec Terraform

J'ai mis en place les fichiers nécessaires pour l'utilisation de Terraform et Ansible, malheureusement j'ai eu beaucoup de difficulté pour la mise en place de ceux-ci tant bien par ma méconnaissance que par la difficulté de compatibilité avec Windows, j'ai passé une grande partie à essayer de faire fonctionner le tout, pour au final ne pas avoir assez de temps et pouvoir faire fonctionner l'instance

```
root@DESKTOP-8700H9I: /mnt/d/travail_ecole/Efrei/graphql-project/terraform

private_dns_name_options {
  enable_resource_name_dns_a_record = false -> null
  enable_resource_name_dns_aaaa_record = false -> null
  hostname_type = "ip-name" -> null
}

root_block_device {
  delete_on_termination = true -> null
  device_name = "/dev/sda1" -> null
  encrypted = false -> null
  iops = 100 -> null
  tags = {} -> null
  tags_all = {} -> null
  throughput = 0 -> null
  volume_id = "vol-0a79fcae063f3a59e" -> null
  volume_size = 8 -> null
  volume_type = "gp2" -> null
  # (1 unchanged attribute hidden)
}

}

Plan: 1 to add, 0 to change, 1 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.instance: Destroying... [id=i-029328d5ae475f5f0]
aws_instance.instance: Still destroying... [id=i-029328d5ae475f5f0, 10s elapsed]
aws_instance.instance: Still destroying... [id=i-029328d5ae475f5f0, 20s elapsed]
aws_instance.instance: Still destroying... [id=i-029328d5ae475f5f0, 30s elapsed]
aws_instance.instance: Still destroying... [id=i-029328d5ae475f5f0, 40s elapsed]
aws_instance.instance: Destruction complete after 42s
aws_instance.instance: Creating...
aws_instance.instance: Still creating... [10s elapsed]
aws_instance.instance: Still creating... [20s elapsed]
aws_instance.instance: Still creating... [30s elapsed]
aws_instance.instance: Provisioning with 'local-exec'...
aws_instance.instance (local-exec): Executing: ["/bin/sh" "-c" "ansible-playbook -i '3.85.137.220,' --user ubuntu --private-key ~/.keys/myKey playbook.yml"]
aws_instance.instance (local-exec): [DEPRECATION WARNING]: community.docker.docker_compose has been deprecated.
aws_instance.instance (local-exec): This module uses docker-compose v1, which is End of Life since July 2022.
aws_instance.instance (local-exec): Please migrate to community.docker.docker_compose_v2. This feature will be
aws_instance.instance (local-exec): removed from community.docker in version 4.0.0. Deprecation warnings can be
aws_instance.instance (local-exec): disabled by setting deprecation_warnings=False in ansible.cfg.
aws_instance.instance (local-exec): [WARNING]: Could not match supplied host pattern, ignoring: ec2_instances

aws_instance.instance (local-exec): PLAY [ec2_instances] *****
aws_instance.instance (local-exec): skipping: no hosts matched

aws_instance.instance (local-exec): PLAY RECAP *****

aws_instance.instance: Creation complete after 33s [id=i-02acac0e63d24ddeb]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
root@DESKTOP-8700H9I: /mnt/d/travail_ecole/Efrei/graphql-project/terraform#
```

Utilisation de WSL pour lancer Terraform et Ansible

The screenshot displays the AWS Management Console interface. The left sidebar contains navigation links for various AWS services. The main content area shows the 'Instances' page, which includes a table of EC2 instances. The instance 'i-029328d5ae475f5f0' is selected and highlighted in blue. Below the table, the details for this instance are displayed, showing its state as 'Terminated'.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
Cloud-AWS-Project	i-01bee771dded16801	Terminated	t2.micro	-	View alarms	us-east-1b	-	-	-
Cloud-AWS-Project	i-098be83726040d42	Terminated	t2.micro	-	View alarms	us-east-1b	-	-	-
Cloud-AWS-Project	i-0ac1255c86b62dcea	Terminated	t2.micro	-	View alarms	us-east-1c	-	-	-
Cloud-AWS-Project	i-006a5f532bd389316	Terminated	t2.micro	-	View alarms	us-east-1b	-	-	-
Cloud-AWS-Project	i-030136cfa75d9e10	Terminated	t2.micro	-	View alarms	us-east-1b	-	-	-
Cloud-AWS-Project	i-081efac96c3d903	Terminated	t2.micro	-	View alarms	us-east-1b	-	-	-
Cloud-AWS-Project	i-029328d5ae475f5f0	Terminated	t2.micro	2/2 checks passed	View alarms	us-east-1c	-	-	-

i-029328d5ae475f5f0 (Cloud-AWS-Project)

Instance summary

Instance ID: i-029328d5ae475f5f0 (Cloud-AWS-Project)

Public IPv4 address: -

Private IPv4 addresses: -

Public IPv4 DNS: -

Instance state: Terminated

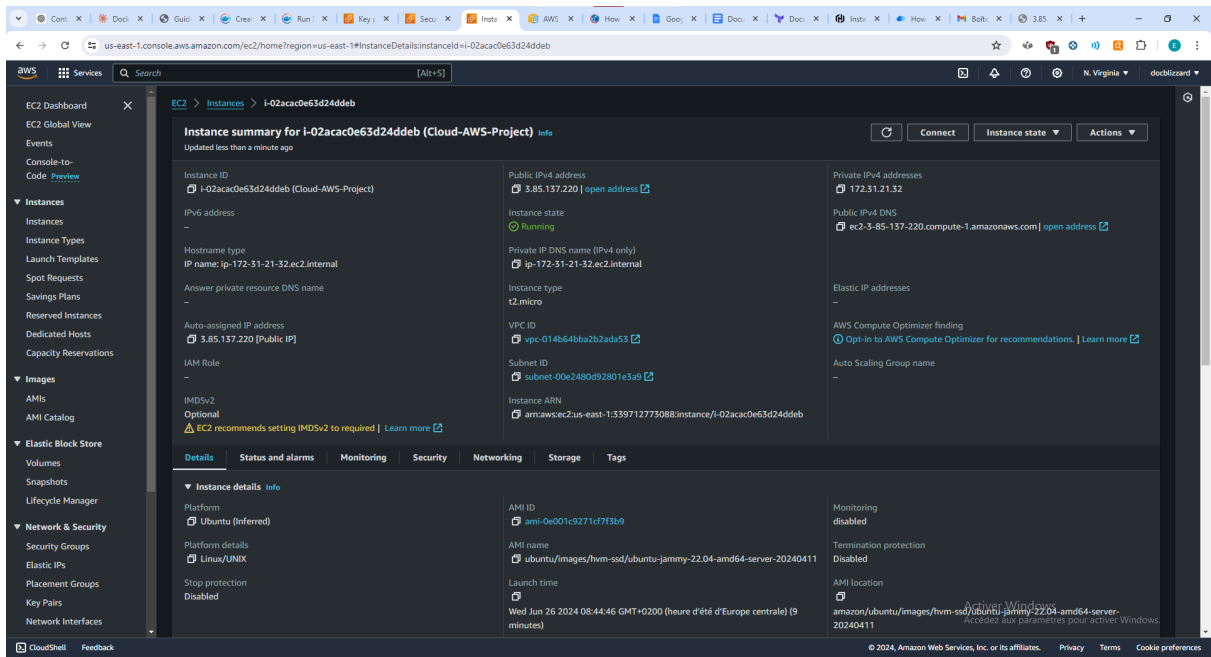
IPv6 address: -

Public IPv4 DNS: -

Hostname type: -

Activier Windows
Accédez aux paramètres pour activer Windows.

Instance EC2 sur AWS



Je n'ai pas compris non plus comment intégrer les variables depuis le fichier .env pour pouvoir les réutiliser.