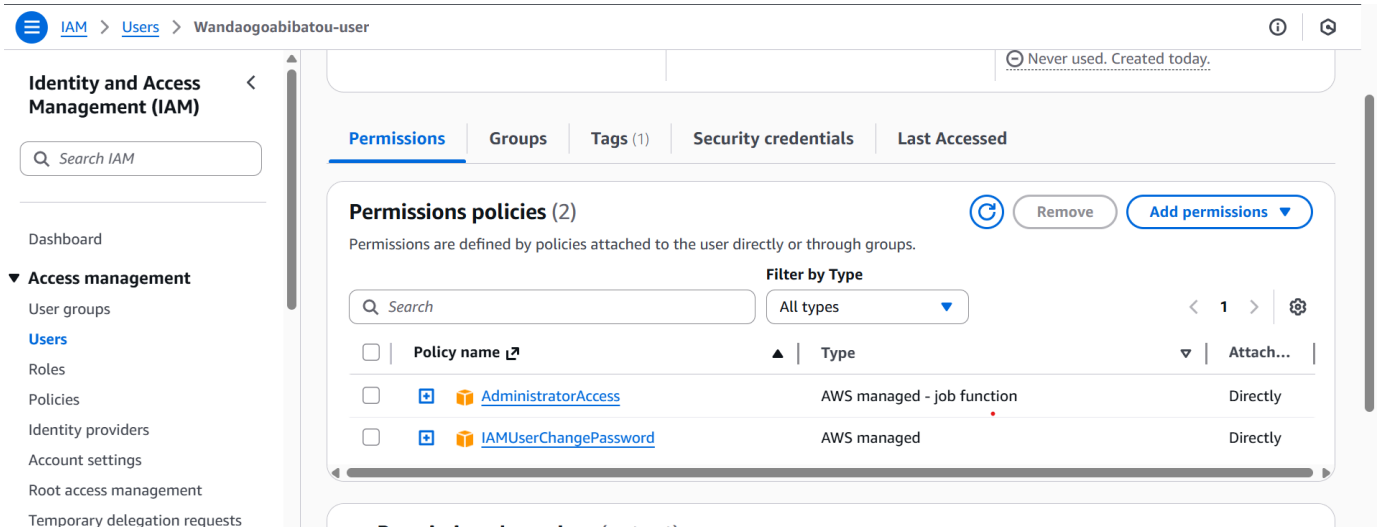
Abibatou Wandaogo

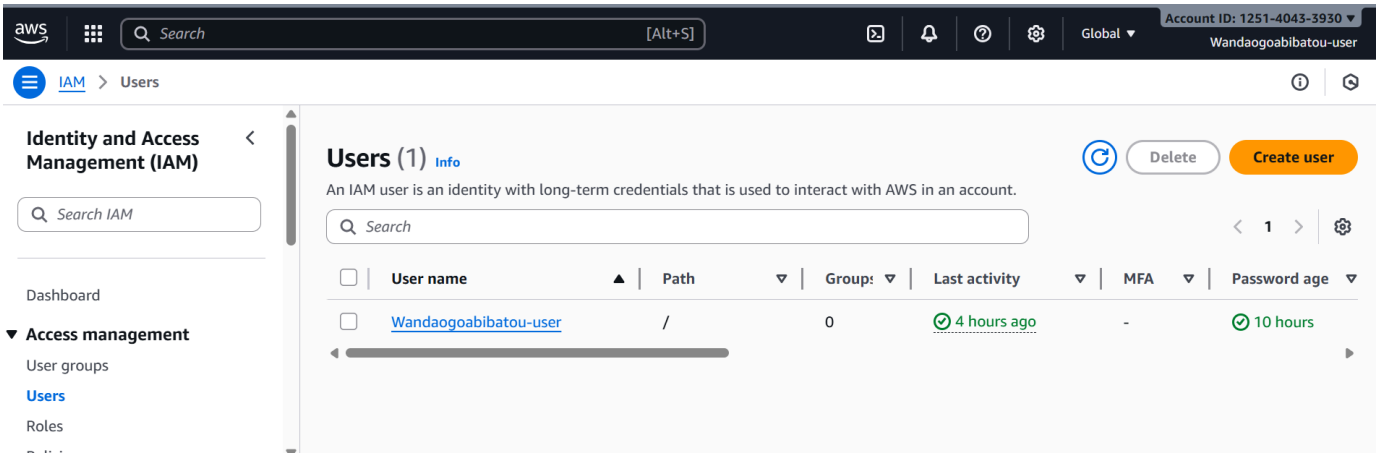
Youcef Guediri

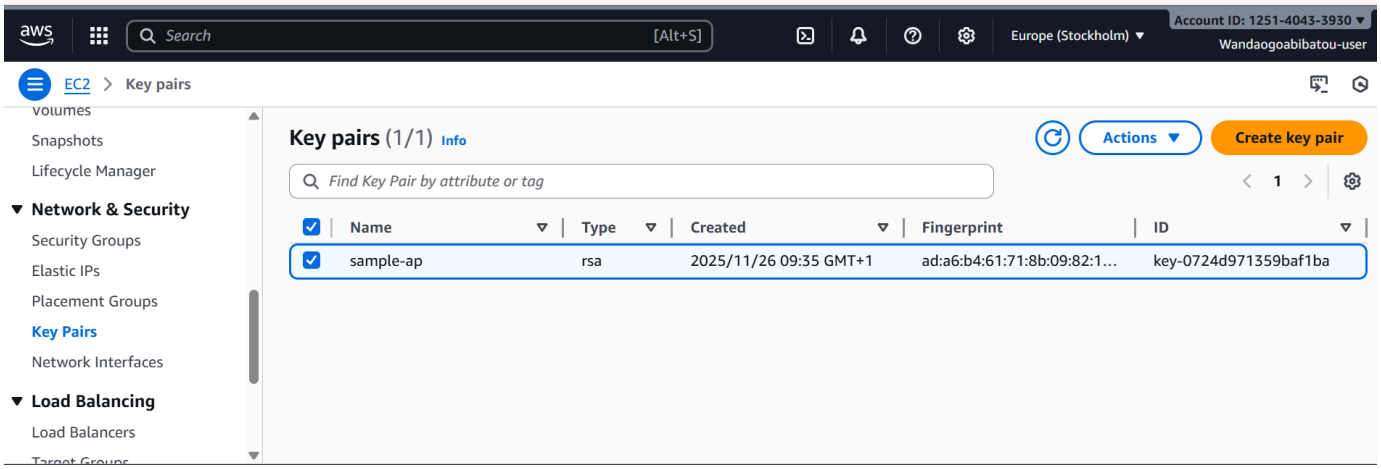
Capture d'écran des résultats des TD

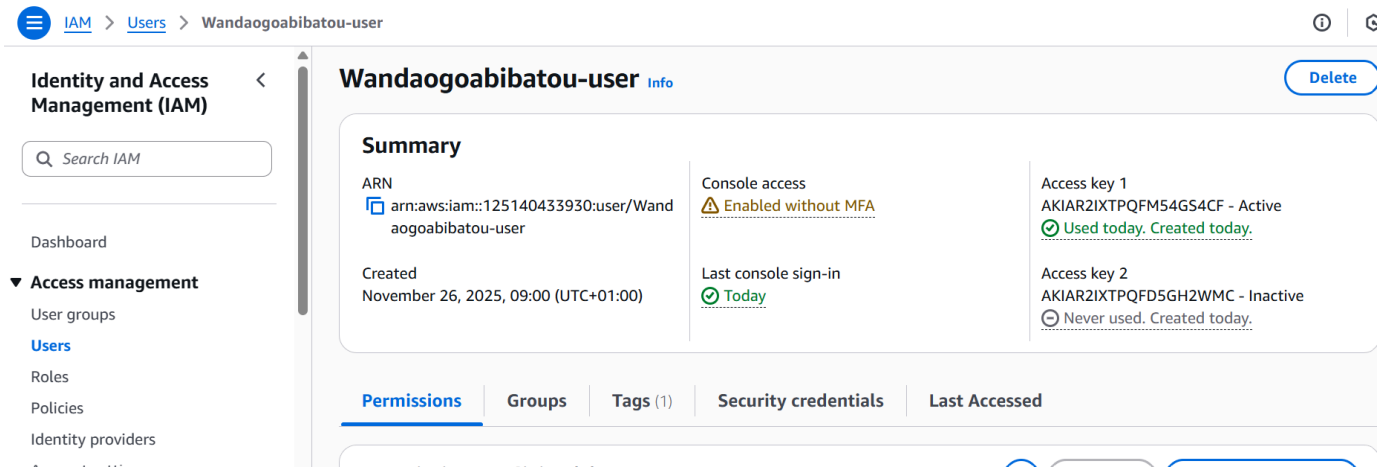
TD1



TD2


aws
Search
[Alt+S]
Global
Account ID: 1251-4043-3930
Wandaogoabibatou-user
IAM > Users
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Users (1) Info
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.
Delete
Create user
Search
1
User name
Path
Groups
Last activity
MFA
Password age
Wandaogoabibatou-user
/
0
4 hours ago
-
10 hours


aws
Search
[Alt+S]
Europe (Stockholm)
Account ID: 1251-4043-3930
Wandaogoabibatou-user
EC2 > Key pairs
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
Key pairs (1/1) Info
Actions
Create key pair
Find Key Pair by attribute or tag
1
Name
Type
Created
Fingerprint
ID
sample-ap
rsa
2025/11/26 09:35 GMT+1
ad:a6:b4:61:71:8b:09:82:1...
key-0724d971359baf1ba




```
}
(de1-env) bibawandaogo@Wandaogo:~/devops_base/td2/scripts/bash/devops_base/td2/scripts/ansible$ ./deploy-ec2-instance.sh
Instance ID = i-0abfa0ce69ed5db50
Security Group ID = sg-0c6be24e072c5f91f
Public IP = 18.222.74.175
(de1-env) bibawandaogo@Wandaogo:~/devops_base/td2/scripts/bash/devops_base/td2/scripts/ansible$
```


```
Hello, World!
```

```
(de1-env) bibawandaogo@Wandaogo:~/devops_base/td2/scripts/bash/devops_base/td2/scripts/ansible$ nano deploy-ec2-instance.sh
(de1-env) bibawandaogo@Wandaogo:~/devops_base/td2/scripts/bash/devops_base/td2/scripts/ansible$ chmod u+x deploy-ec2-instance.sh
(de1-env) bibawandaogo@Wandaogo:~/devops_base/td2/scripts/bash/devops_base/td2/scripts/ansible$ ./deploy-ec2-instance.sh
Récupération du VPC par défaut...
VPC trouvé : vpc-091401036199ad4f8
Vérification si le Security Group existe déjà...
Security Group déjà existant : sg-0c6be24e072c5f91f
Lancement de l'instance...
Instance lancée : i-0cc1800ef9f8de1e9
(de1-env) bibawandaogo@Wandaogo:~/devops_base/td2/scripts/bash/devops_base/td2/scripts/ansible$ aws ec2 describe-instances \
  --instance-ids i-0cc1800ef9f8de1e9 \
  --query "Reservations[0].Instances[0].State.Name" \
  --output text
running
(de1-env) bibawandaogo@Wandaogo:~/devops_base/td2/scripts/bash/devops_base/td2/scripts/ansible$ aws ec2 describe-instances \
  --instance-ids i-0cc1800ef9f8de1e9 \
  --query "Reservations[0].Instances[0].PublicIpAddress" \
  --output text
18.118.29.153
```

```
(de1-env) bibawandaogo@Wandaogo:~/devops_base/td2/scripts/ansible$ # Récupérer l'IP publiqu

PUBLIC_IP=$(aws ec2 describe-instances \
  --filters "Name=tag:Name,Values=sample-app-ansible" "Name=instance-state-name,Values=runn
ng" \
  --query 'Reservations[0].Instances[0].PublicIpAddress' \
  --region us-east-2 \
  --output text)

echo "Public IP: $PUBLIC_IP"

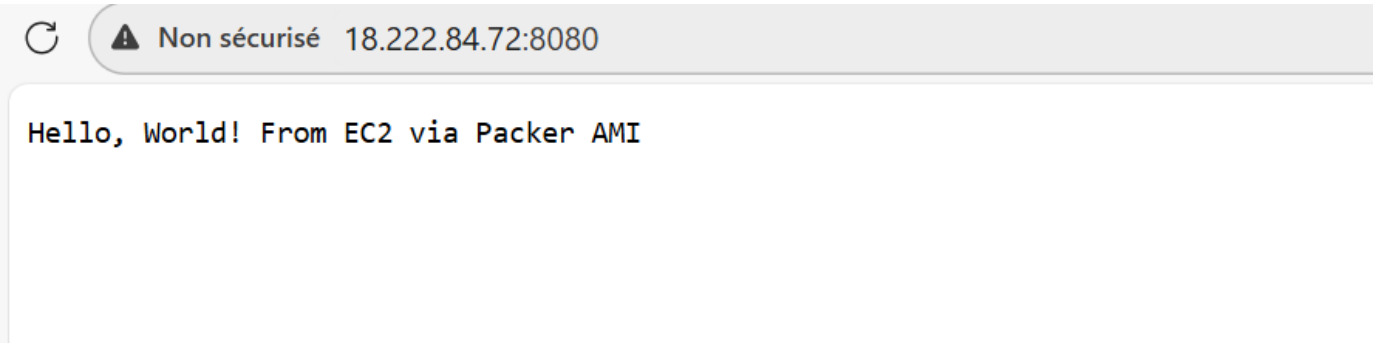
# Attendre 5s que l'app soit bien démarrée
sleep 5

# Tester l'application
curl http://$PUBLIC_IP:8080

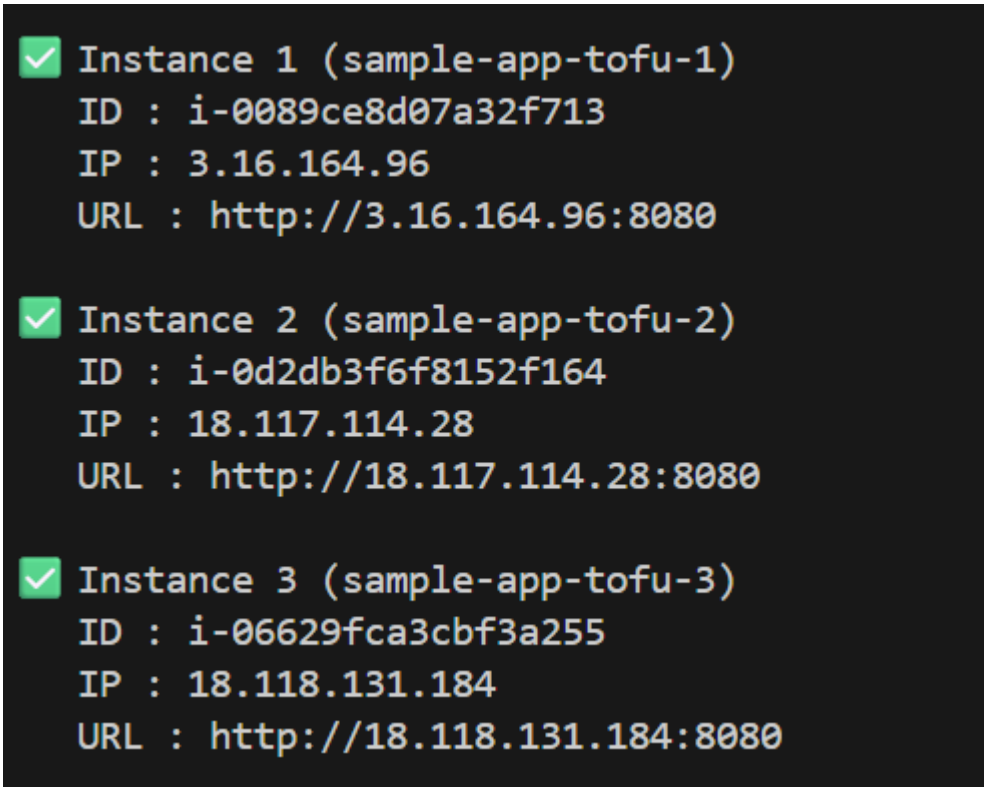
echo ""
echo "✅ Test terminé !"
Public IP: 3.143.241.36
Hello, World! From EC2 via Ansible

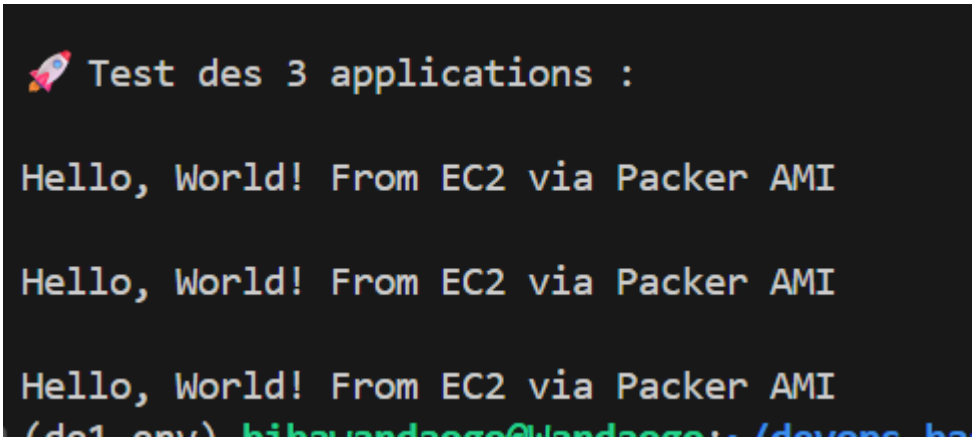
✅ Test terminé !
(de1-env) bibawandaogo@Wandaogo:~/devops_base/td2/scripts/ansible$
```

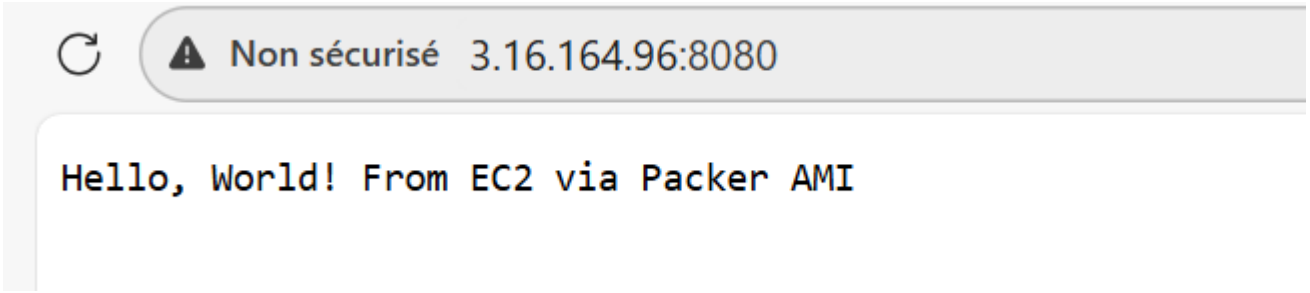
Hello, World! From EC2 via Ansible

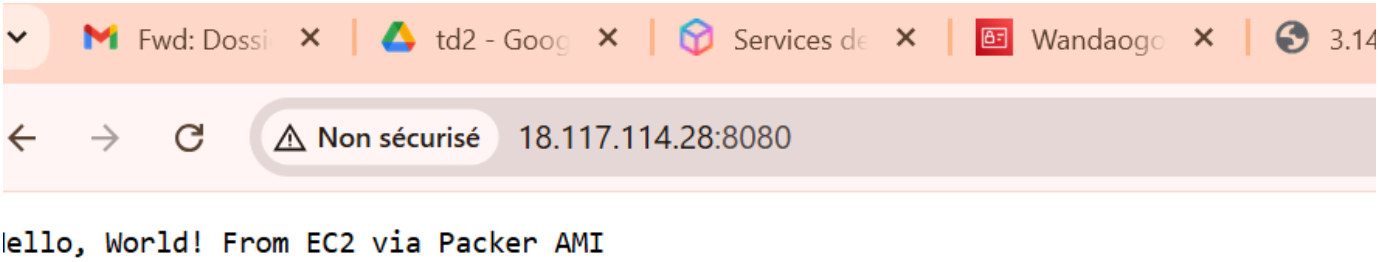


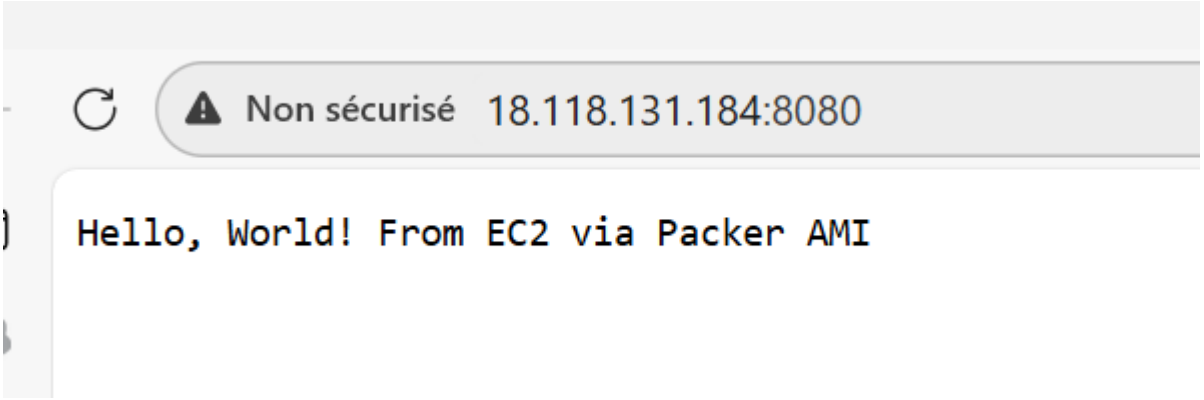
Hello, World! From EC2 via Packer AMI



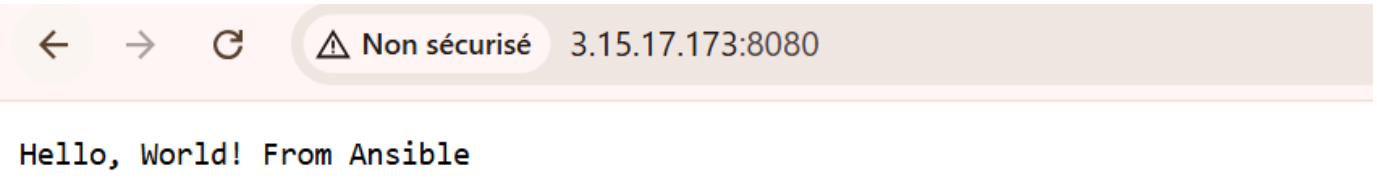

🚀 Test des 3 applications :
Hello, World! From EC2 via Packer AMI
Hello, World! From EC2 via Packer AMI
Hello, World! From EC2 via Packer AMI


Non sécurisé 3.16.164.96:8080
Hello, World! From EC2 via Packer AMI


Non sécurisé 18.117.114.28:8080
ello, World! From EC2 via Packer AMI


Non sécurisé 18.118.131.184:8080
Hello, World! From EC2 via Packer AMI

TD3



```
(de1-env) bibawandaogo@Wandaogo:~/devops_base/TD3/scripts/ansible$ cd /home/bibawanda
ogo/devops_base/TD3/scripts/ansible

# Extraire correctement une IP
IP=$(grep -v "^\[" hosts.ini | grep -v "^$" | head -1 | awk '{print $1}')

echo "IP extraite : $IP"

echo ""
echo "Test de l'application sur $IP:8080"
curl http://$IP:8080

echo ""
echo "✅ Application testée"
IP extraite : 3.15.17.173

Test de l'application sur 3.15.17.173:8080
Hello, World! From Ansible

✅ Application testée
```

```
## Architecture Déployée …
### Sample App Instances (Backend)
- i-0c5b2124b7c2b6b35 : 3.15.17.173
- i-029da7969ddfd8203 : 3.23.101.136
- i-02b77a8feab3eb9bc : 18.118.28.93

### Nginx Load Balancer
- Instance : 3.142.198.146

## Rôles Ansible Créés

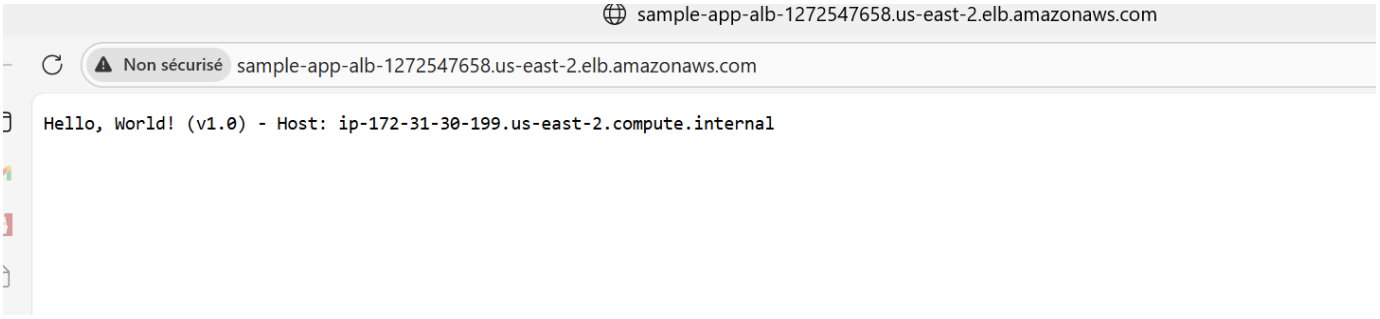
1. **nodejs-app** : Installation Node.js + PM2
2. **sample-app** : Déploiement application
3. **nginx** : Configuration load balancer
```

```
## Test

```bash
curl http://3.142.198.146
# Réponse : Hello, World! From Ansible
```

Rolling Updates

```bash
ansible-playbook -i hosts.ini configure_sample_app_playbook.yml -v
# serial: 1 = 1 instance à la fois
# max_fail_percentage: 30 = tolérance 30% erreurs
```
```




```
🌐 Application URL: http://sample-app-alb-1272547658.us-east-2.elb.amazonaws.com

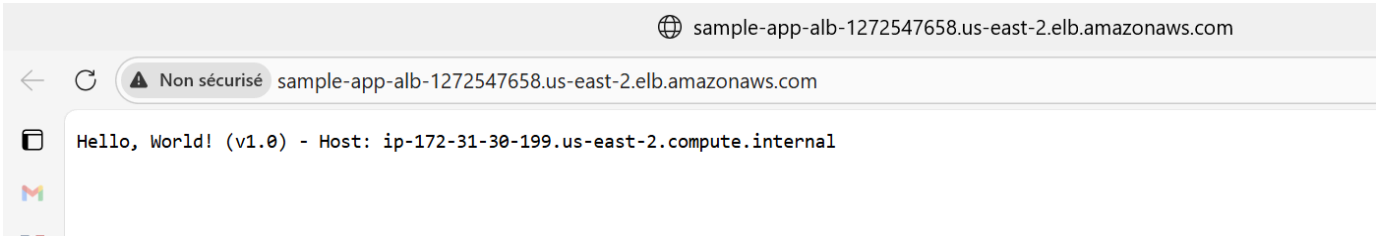
🧪 Test:

Request 1:
Hello, World! (v1.0) - Host: ip-172-31-44-157.us-east-2.compute.internal

Request 2:
Hello, World! (v1.0) - Host: ip-172-31-15-188.us-east-2.compute.internal

Request 3:
Hello, World! (v1.0) - Host: ip-172-31-15-188.us-east-2.compute.internal

✅ Infrastructure fonctionnelle !
```



```
(de1-env) bibawandaogo@Wandaogo:~/devops_base/TD3/scripts/kubernetes$ cd /home/bibawa
ndaogo/devops_base/TD3/scripts/kubernetes

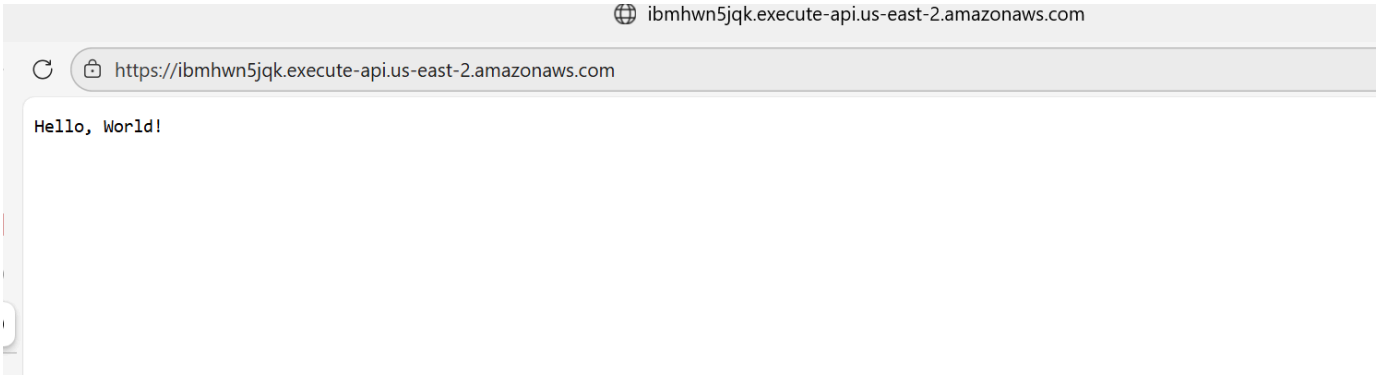
echo "=== Supprimer le déploiement cassé ==="
kubectl delete deployment sample-app-deployment …
=== Réappliquer le déploiement v2 ===
deployment.apps/sample-app-deployment created

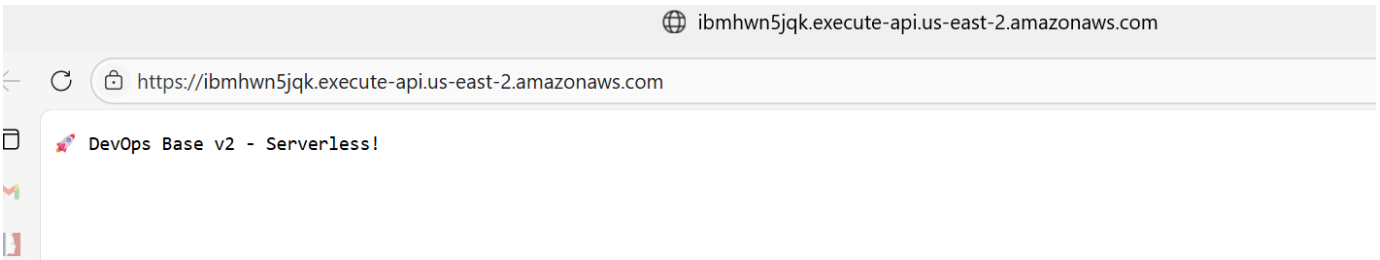
=== Attendre les pods ===

=== Vérifier ===
NAME                                   READY   STATUS             RESTARTS   AGE
sample-app-deployment-556cc6c766-9fj5n   0/1     ErrImageNeverPull   0          12s
sample-app-deployment-556cc6c766-m29kl   0/1     ErrImageNeverPull   0          12s
sample-app-deployment-556cc6c766-tshxm   0/1     ErrImageNeverPull   0          12s
sample-app-deployment-6b68c785f5-jlq69   1/1     Terminating         0          38m
sample-app-deployment-6b68c785f5-pqqst   1/1     Terminating         0          38m
sample-app-deployment-6b68c785f5-sgtlk   1/1     Terminating         0          38m

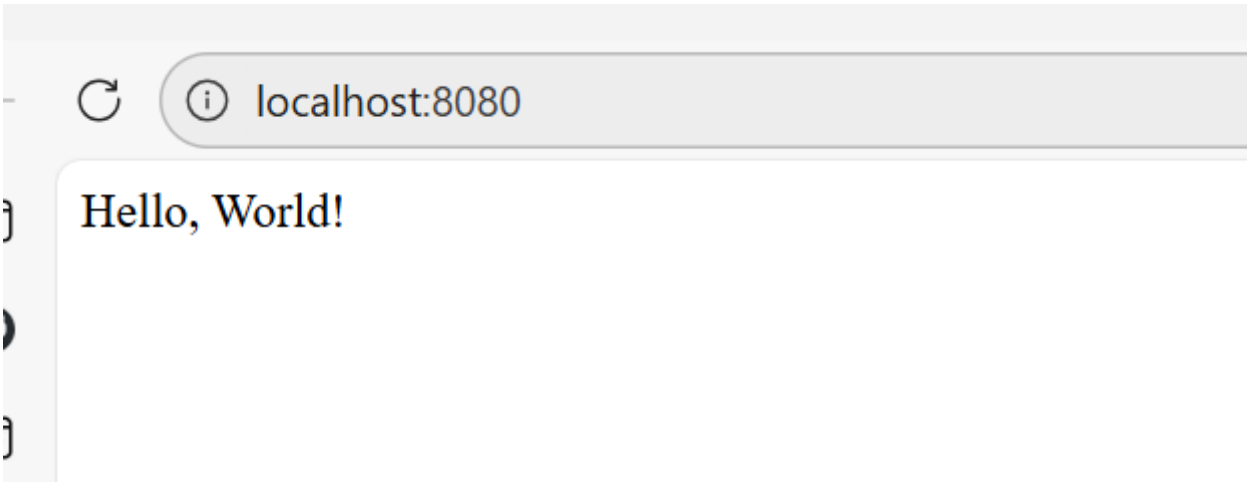
=== TEST ===
DevOps Base!

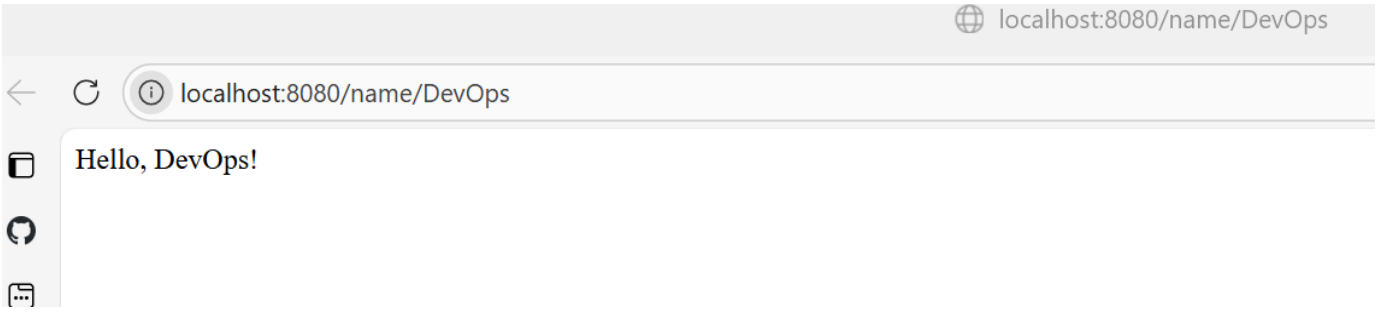
✅ Résultat attendu : 🚀 DevOps Base v2 - Rolling Update Works!
(de1-env) bibawandaogo@Wandaogo:~/devops_base/TD3/scripts/kubernetes$
```

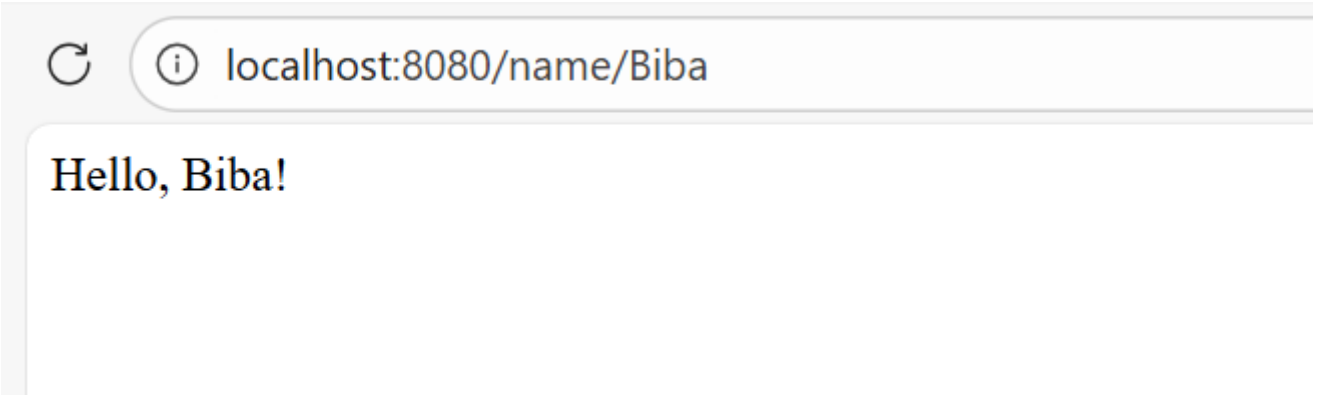


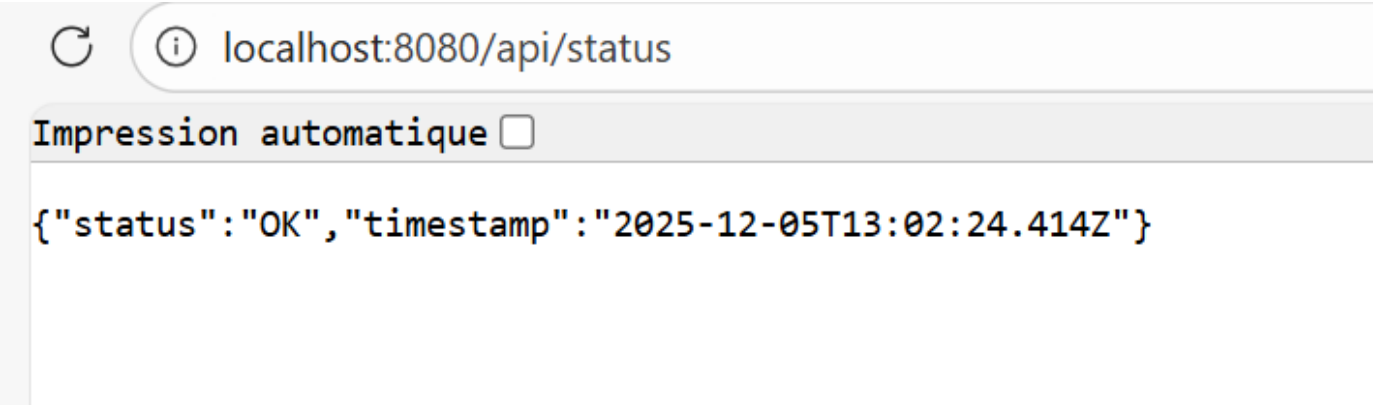


TD4


localhost:8080
Hello, World!


localhost:8080/name/DevOps
localhost:8080/name/DevOps
Hello, DevOps!


localhost:8080/name/Biba
Hello, Biba!




```
echo "✅ Tous les endpoints testés"
=== Tester les endpoints avec curl ===

=== 1. Test GET / ===
Hello, World!

=== 2. Test GET /name/DevOps ===
Hello, DevOps!

=== 3. Test GET /name/Biba ===
Hello, Biba!

=== 4. Test GET /api/status ===
{"status":"OK","timestamp":"2025-12-05T13:00:30.598Z"}
✅ Tous les endpoints testés
(de1-env) bibawandaogo@Wandaogo:~$
```

```
----------|---------|----------|---------|---------|-------------------
File      | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
----------|---------|----------|---------|---------|-------------------
All files |     100 |      100 |     100 |     100 |
 app.js   |     100 |      100 |     100 |     100 |
----------|---------|----------|---------|---------|-------------------
Test Suites: 1 passed, 1 total
Tests:       5 passed, 5 total
Snapshots:   0 total
Time:        0.789 s
Ran all test suites.
```

```
----------|---------|----------|---------|---------|-------------------
File      | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
----------|---------|----------|---------|---------|-------------------
All files |     100 |      100 |     100 |     100 |
 app.js   |     100 |      100 |     100 |     100 |
----------|---------|----------|---------|---------|-------------------
Test Suites: 1 failed, 1 total
Tests:       1 failed, 4 passed, 5 total
Snapshots:   0 total
Time:        0.704 s, estimated 1 s
Ran all test suites.

⚠️Les tests échouent comme prévu !
```

TD5


```
✅ STATUS FINAL:

   PARTIE 1: ✅ 100% COMPLÈTE
   TESTS: ✅ 6/6 RÉUSSIS
   INFRASTRUCTURE: ✅ OPÉRATIONNELLE
   WORKFLOWS: ✅ ACTIFS
```

```
echo "╔═══  RÉSUMÉ FINAL - TD5 PART 2 COMPLÈTEMENT DÉPLOYÉE          ═══╗" …
✅ AUTHENTIFICATION:

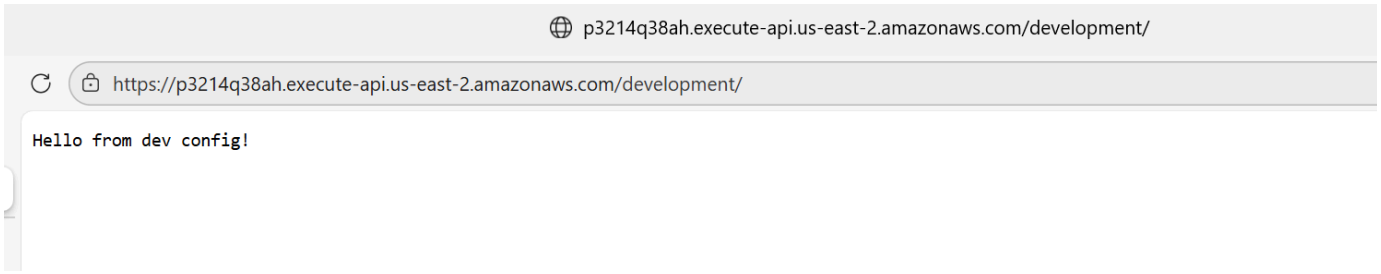
   Plan Role: lambda-sample-plan-role
   Apply Role: lambda-sample-apply-role
   - Méthode: OIDC Tokens (Sécurisé) ✅
   - Permissions: S3 + DynamoDB ✅

═══════════════════════════════════════════════════════════

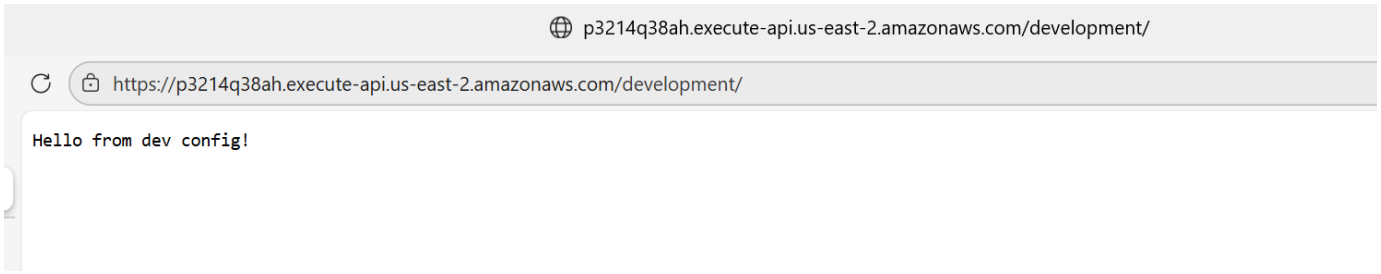
🎯 FONCTIONNALITÉS DÉPLOYÉES:

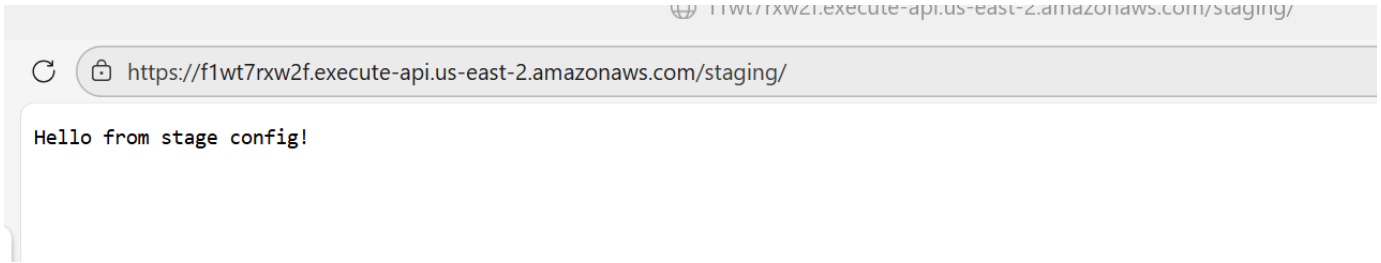
   1. ✅ Infrastructure Backend (S3 + DynamoDB)
   2. ✅ Remote State Management
   3. ✅ State Locking (Prévient les conflits)
   4. ✅ Continuous Delivery Pipelines
   5. ✅ Automatic Plans on Pull Requests
   6. ✅ Automatic Apply on Merge
   7. ✅ Secure OIDC Authentication
   8. ✅ Version Control pour l'Infrastructure
```

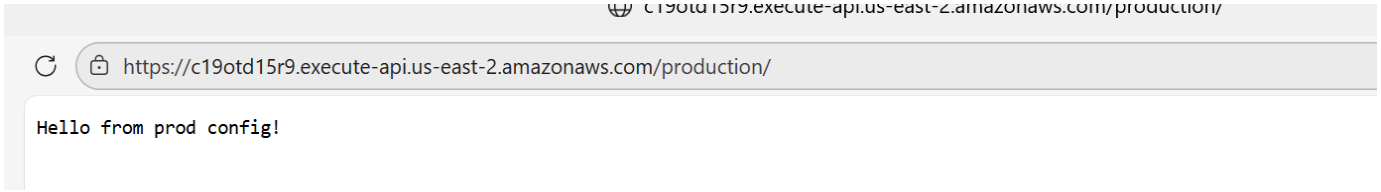
7 workflow runs	Event	Status	Branch	Actor	
✅ docs: Add comprehensive README for TD5 Part 2 - ... Application Tests #7: Pull request #2 opened by bibatou2004			test-cd-pipeline		6 minutes ago 22s
✅ feat: Test CD Pipeline with Lambda v2 (#1) Application Tests #6: Commit 2cfe0e6 pushed by bibatou2004			main		15 minutes ago 17s
✅ feat: Test CD Pipeline with Lambda v2 Application Tests #5: Pull request #1 synchronize by bibatou2004			test-cd-pipeline		21 minutes ago 23s
✅ feat: Test CD Pipeline with Lambda v2 Application Tests #4: Pull request #1 opened by bibatou2004			test-cd-pipeline		21 minutes ago 19s
✅ fix: Corriger la configuration API Gateway Application Tests #3: Commit f8b01d2 pushed by bibatou2004			main		Today at 3:54 PM 20s
✅ feat: Ajouter application Lambda sample et workflo... Application Tests #2: Commit dd08a52 pushed by bibatou2004			main		Today at 3:42 PM 18s

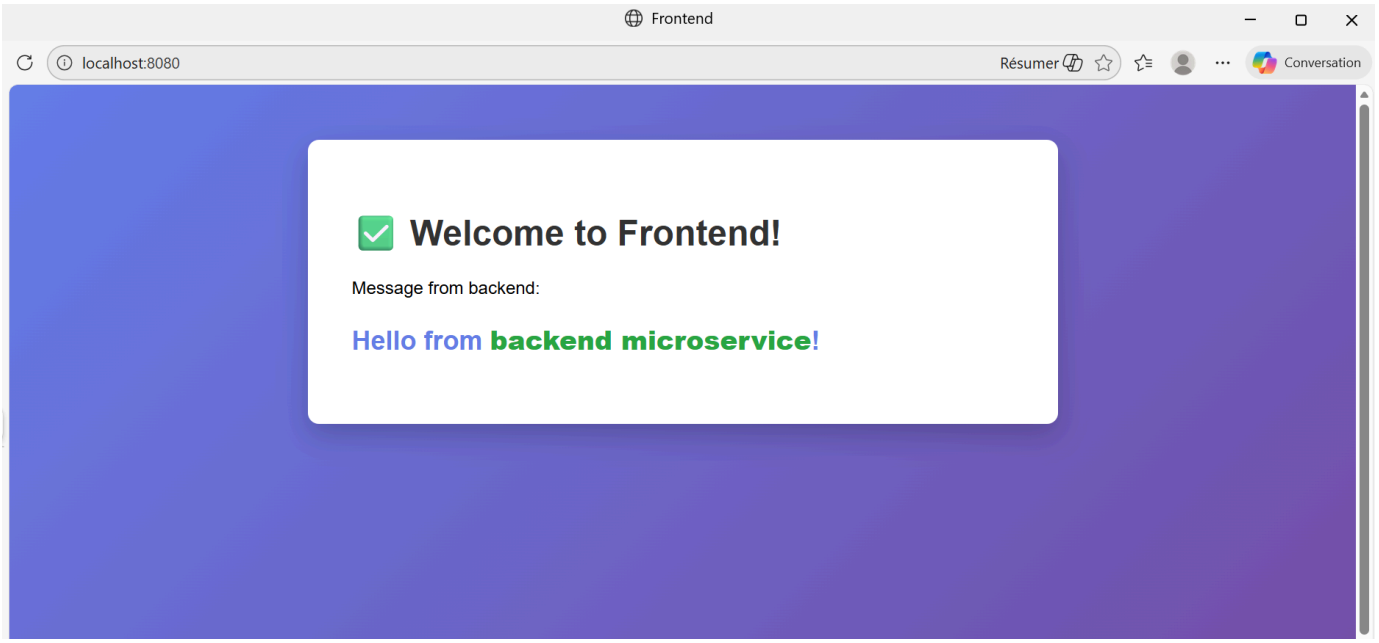

p3214q38ah.execute-api.us-east-2.amazonaws.com/development/
https://p3214q38ah.execute-api.us-east-2.amazonaws.com/development/
Hello from dev config!

TD6


p3214q38ah.execute-api.us-east-2.amazonaws.com/development/
https://p3214q38ah.execute-api.us-east-2.amazonaws.com/development/
Hello from dev config!


https://f1wt7rxw2f.execute-api.us-east-2.amazonaws.com/staging/
Hello from stage config!


https://c19otd15r9.execute-api.us-east-2.amazonaws.com/production/
Hello from prod config!


Frontend
localhost:8080
Résumer
Conversation
✅ Welcome to Frontend!
Message from backend:
Hello from backend microservice!


127.0.0.1:32771
127.0.0.1:32771
Client sent an HTTP request to an HTTPS server.