DEVOXX FRANCE 2024

MAIS AU FAIT ?

/ ÇA MARCHE COMMENT LES SERVICE ACCOUNTS ?

JULIEN WITTOUCK
Freelance @CodeKaio
Associé @Ekit3
Teacher @univ-lille
Team @Cloud-Nord

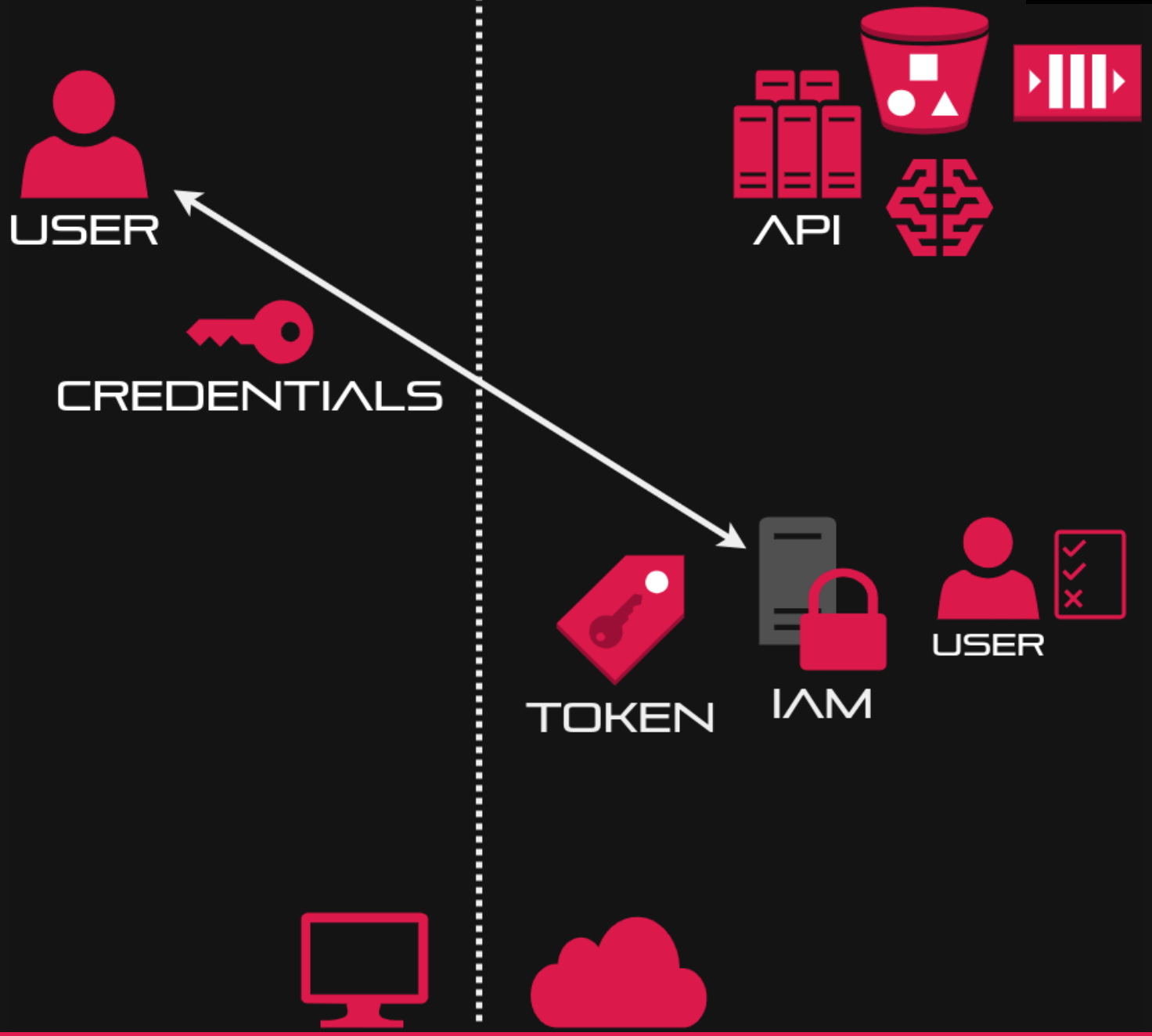
USER

API

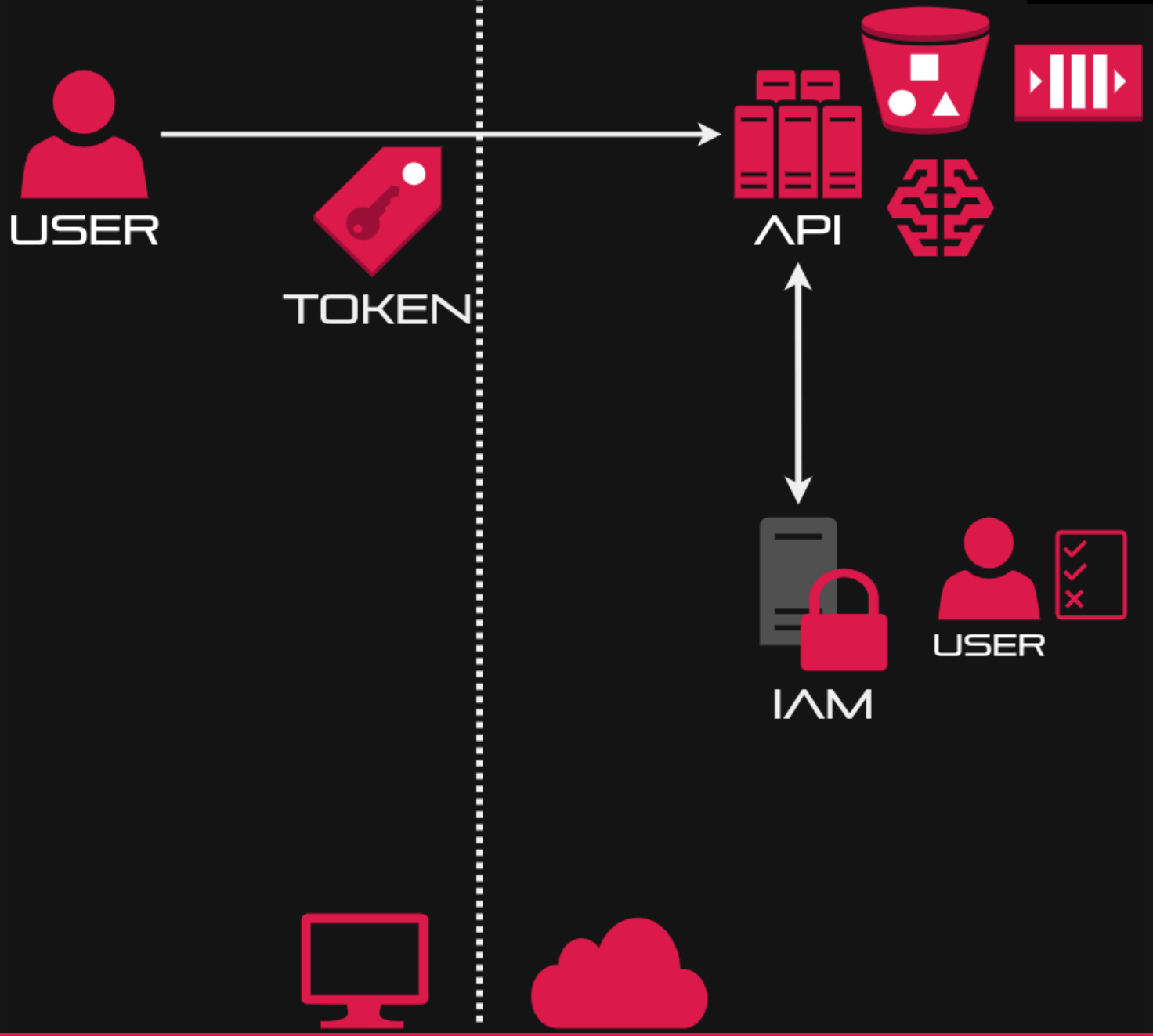
IAM

INTRODUCTION

INTRODUCTION


USER
CREDENTIALS
API
TOKEN
IAM
USER

INTRODUCTION



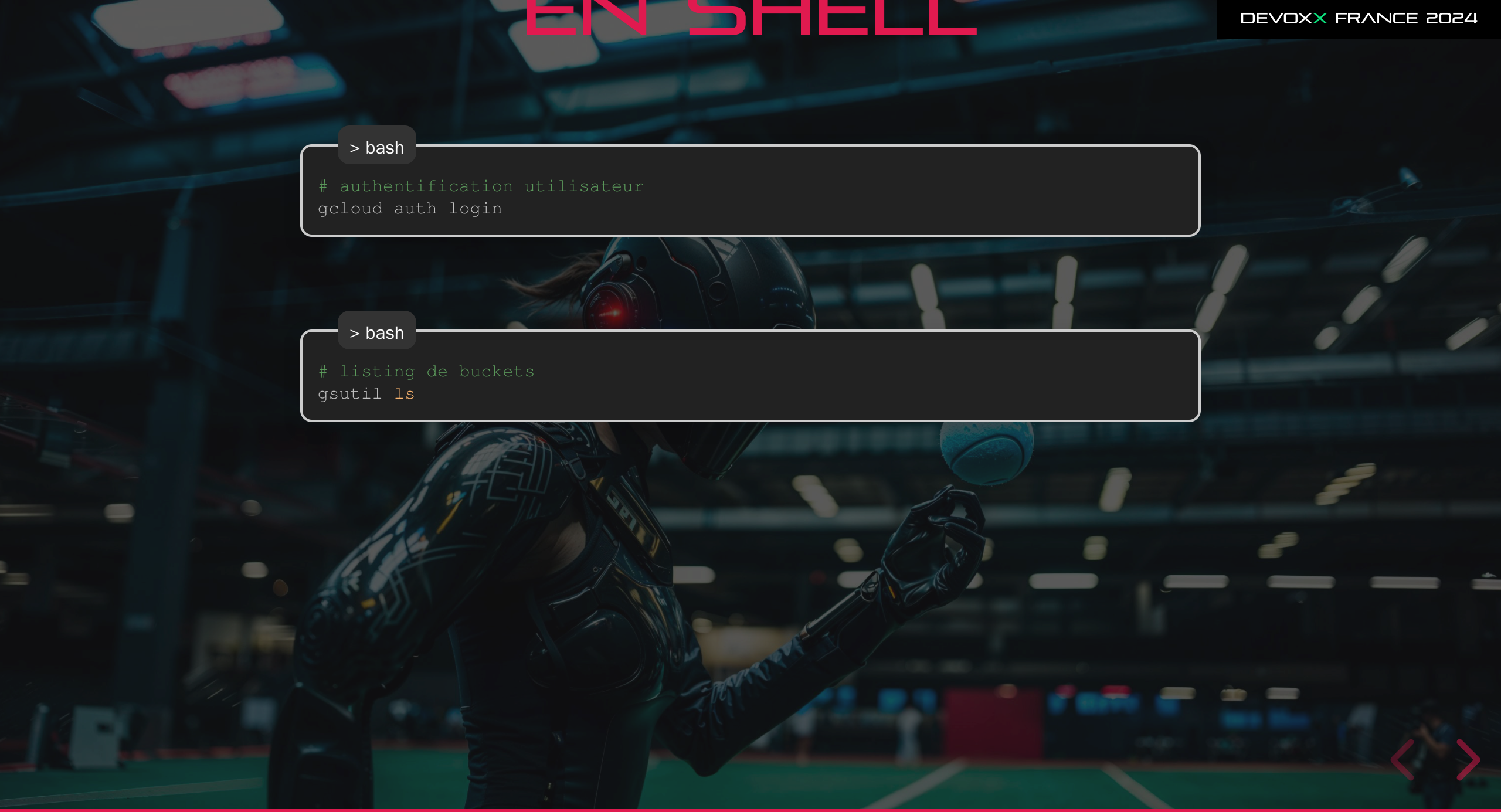
EN SHELL

> bash

```
# authentification utilisateur
gcloud auth login
```

> bash

```
# listing de buckets
gsutil ls
```



EN SHELL

> bash

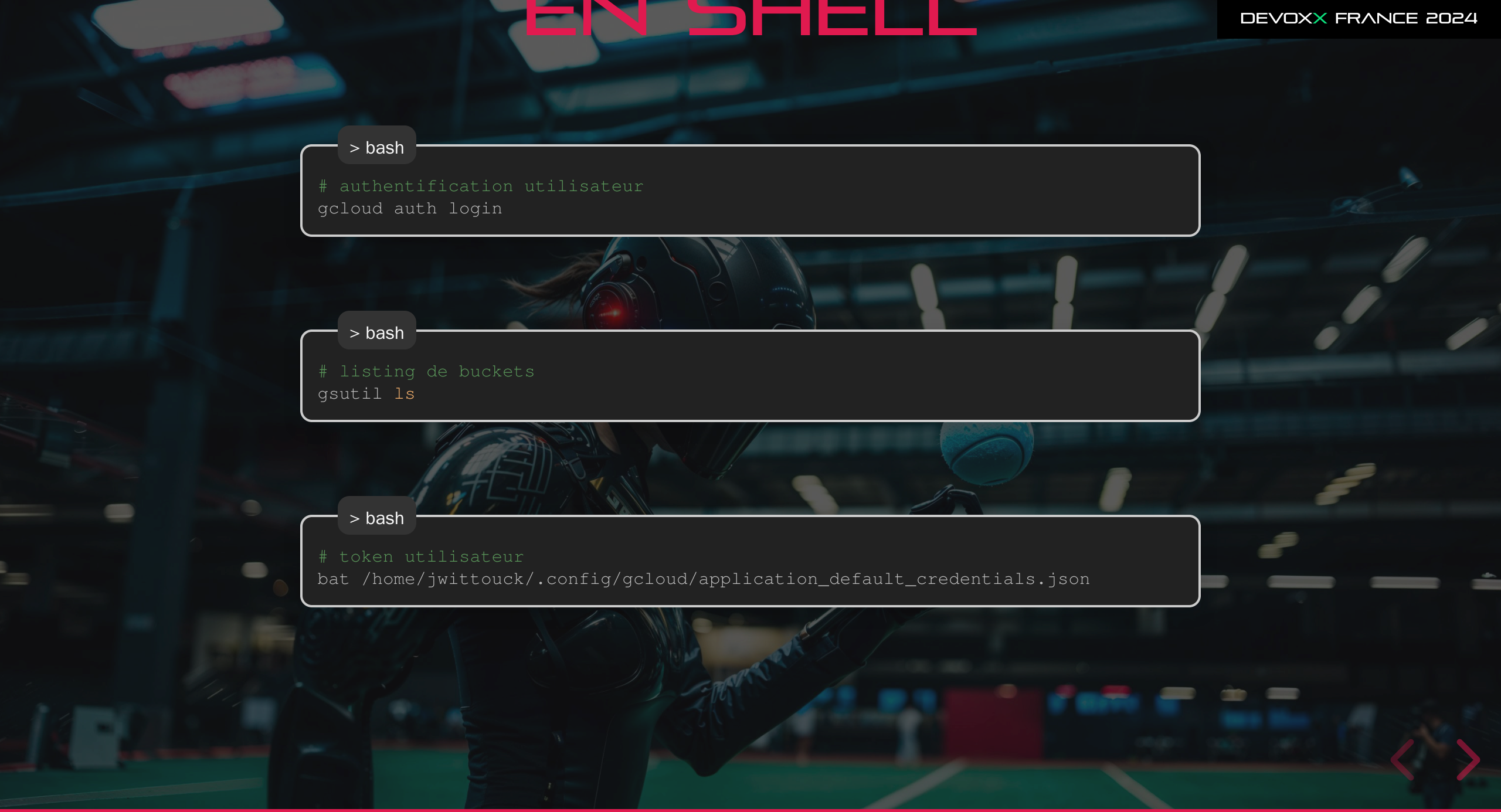
```
# authentification utilisateur
gcloud auth login
```

> bash

```
# listing de buckets
gsutil ls
```

> bash

```
# token utilisateur
bat /home/jwittouck/.config/gcloud/application_default_credentials.json
```



EN SHELL

> bash

```
# authentification utilisateur
gcloud auth login
```

> bash

```
# listing de buckets
gsutil ls
```

> bash

```
# token utilisateur
bat /home/jwittouck/.config/gcloud/application_default_credentials.json
```

> bash

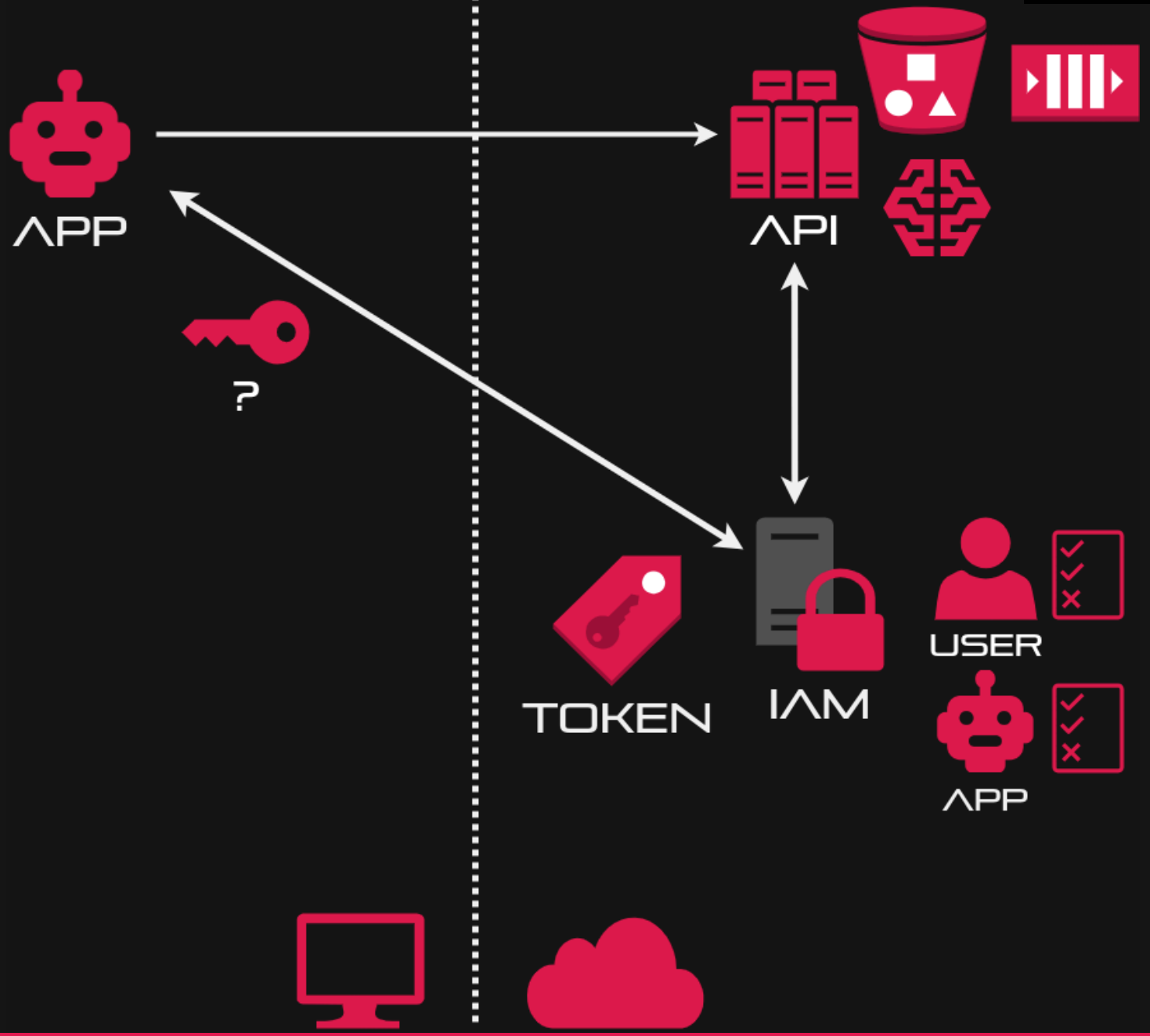
```
# requête à la main
ACCESS_TOKEN=$(gcloud auth application-default print-access-token)
curl -s -H "Authorization: Bearer $ACCESS_TOKEN" "https://storage.googleapis.com/
```

EN JAVA

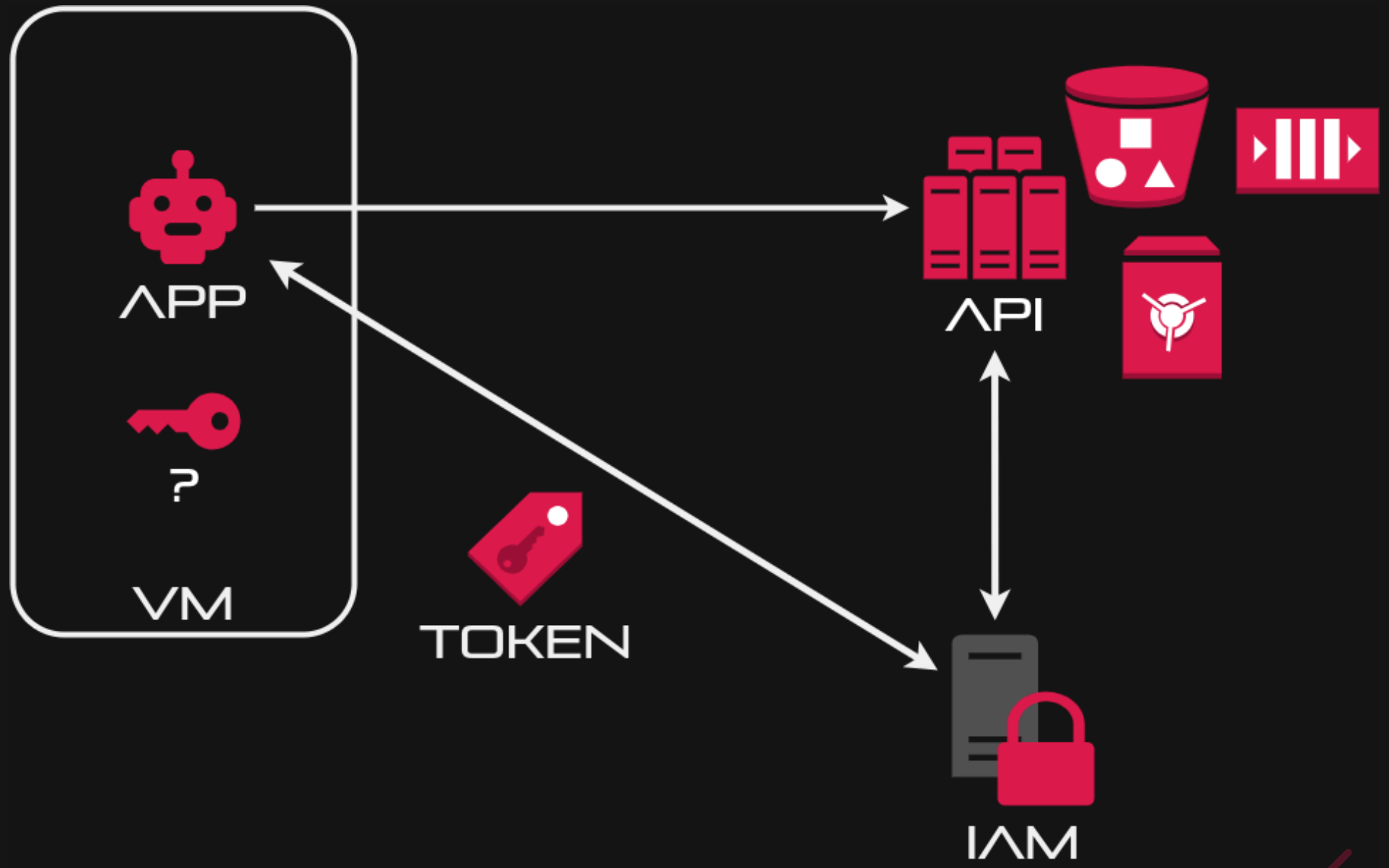
java

```java
// lister les buckets
public Stream<Bucket> stream() {
    var storage = StorageOptions.getDefaultInstance().getService();
    return storage.list().streamValues();
}
```

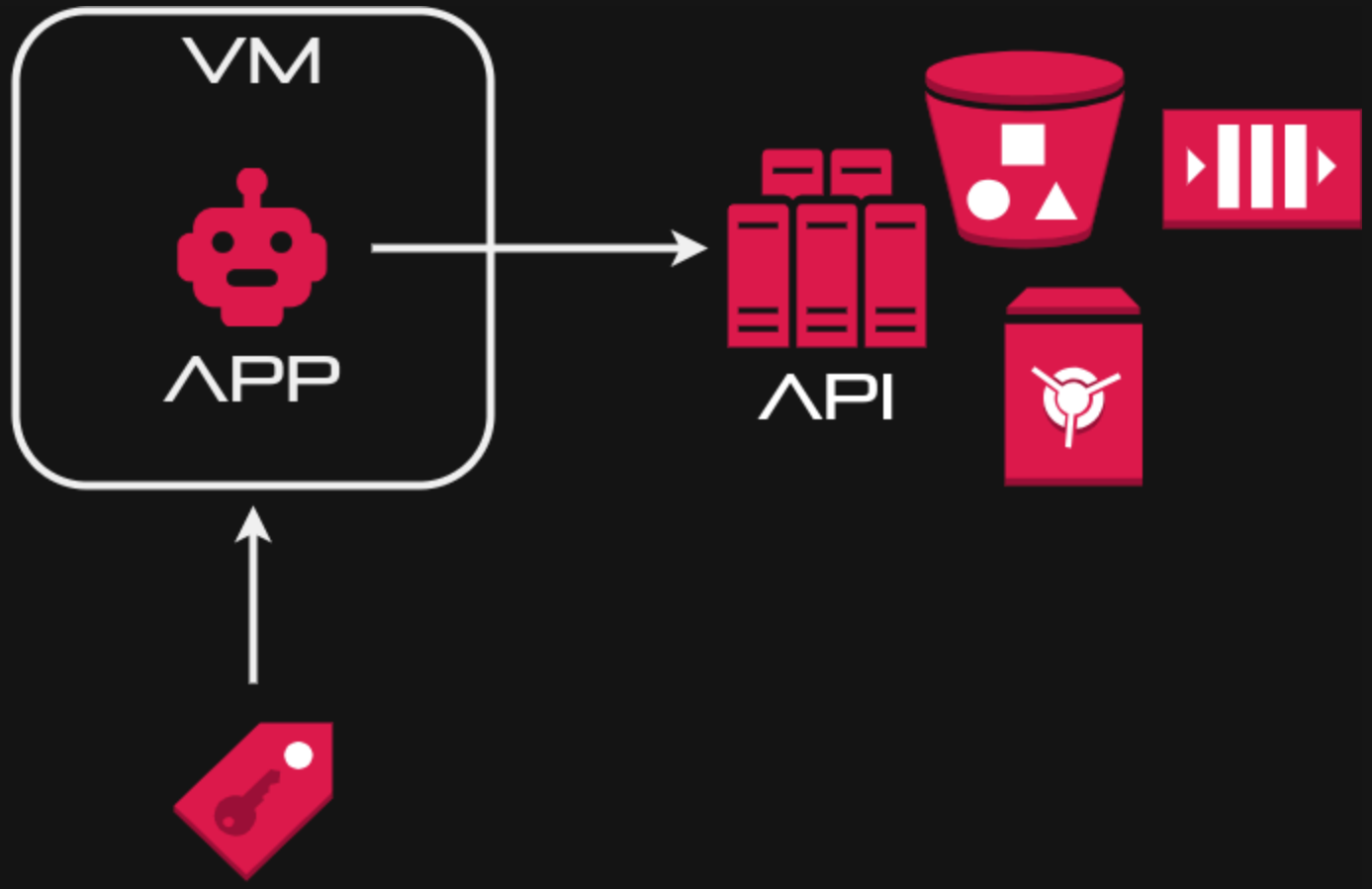
INTRODUCTION



SUR UNE VM



MONTER UN SERVICE ACCOUNT



SUR UNE VM

SERVICE ACCOUNT MONTÉ

> bash

```
$ gcloud compute instances create devoxx-demo-instance-2024 \
    --project=devoxx-2024 \
    --zone=europe-west9-a \
    --machine-type=n2-standard-2 \
    --subnet=default \
    --service-account=devoxx-demo-sa@devoxx-2024.iam.gserviceaccount.com \
    --image=ubuntu-2310-mantic-amd64
```

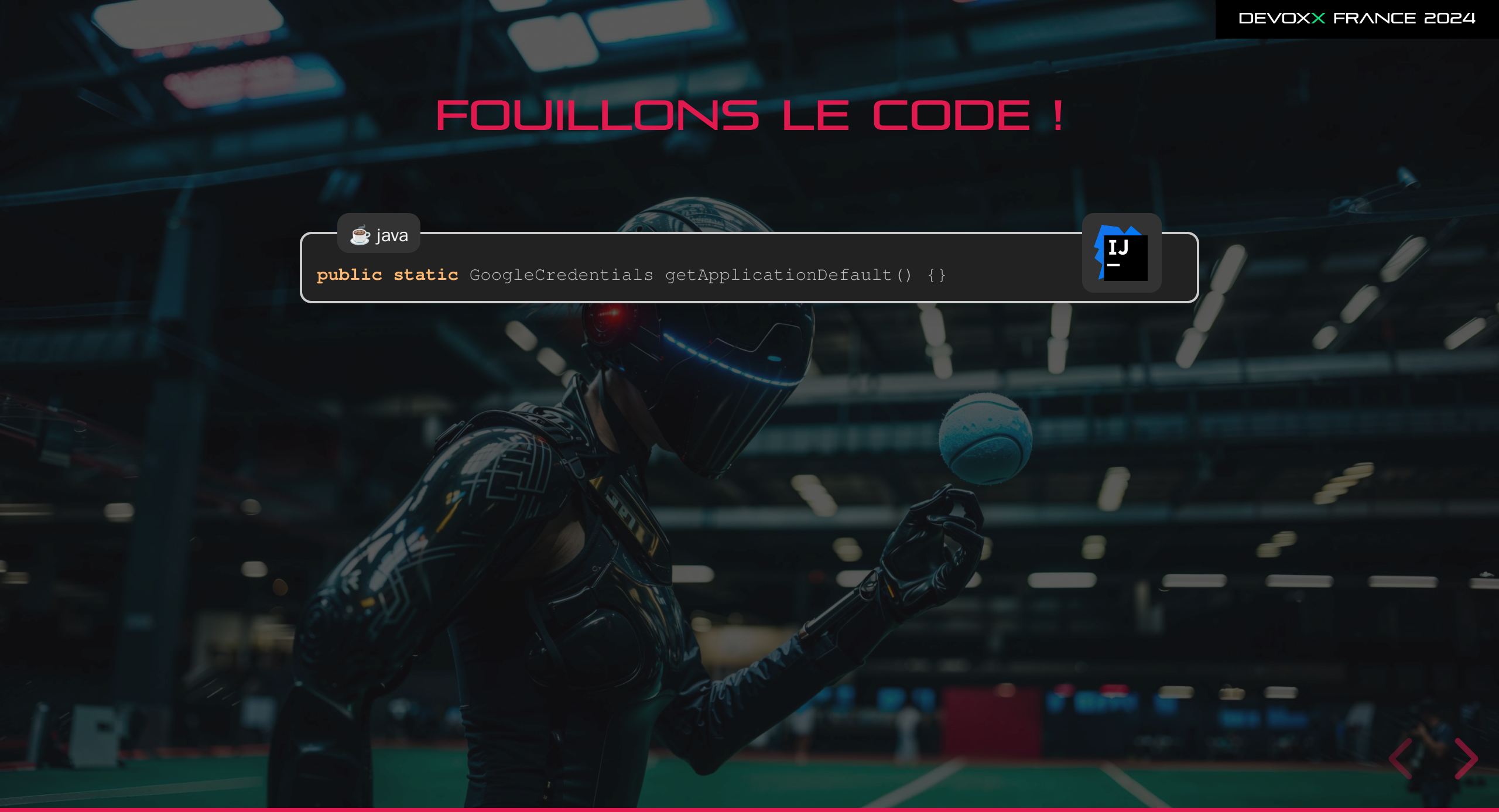
> bash

```
# connection SSH à la VM
gcloud compute ssh devoxx-demo-instance
```

> bash

```
# upload du java native
gcloud compute scp \
    $DEMO_CODE/target/list-buckets devoxx-demo-instance:/home/jwittouck
```

FOUILLONS LE CODE !



FOUILLONS LE CODE !

☕ java

```java
public static GoogleCredentials getApplicationDefault() {}
```

> bash

```bash
gcloud compute ssh devoxx-demo-instance

curl http://metadata.google.internal
```

FOUILLONS LE CODE !

☕ java

```java
public static GoogleCredentials getApplicationDefault() {}
```

> bash

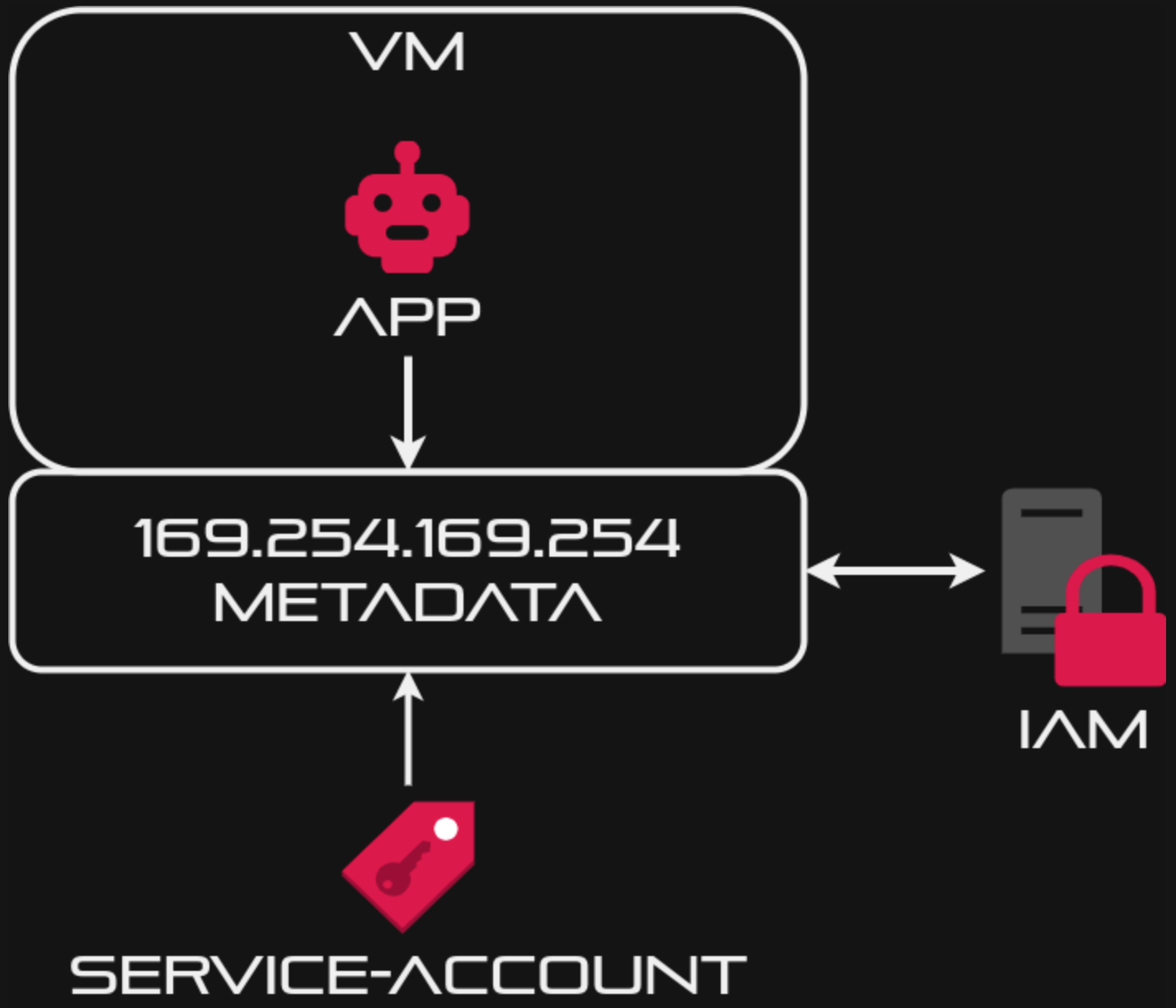
```bash
gcloud compute ssh devoxx-demo-instance

curl http://metadata.google.internal
```

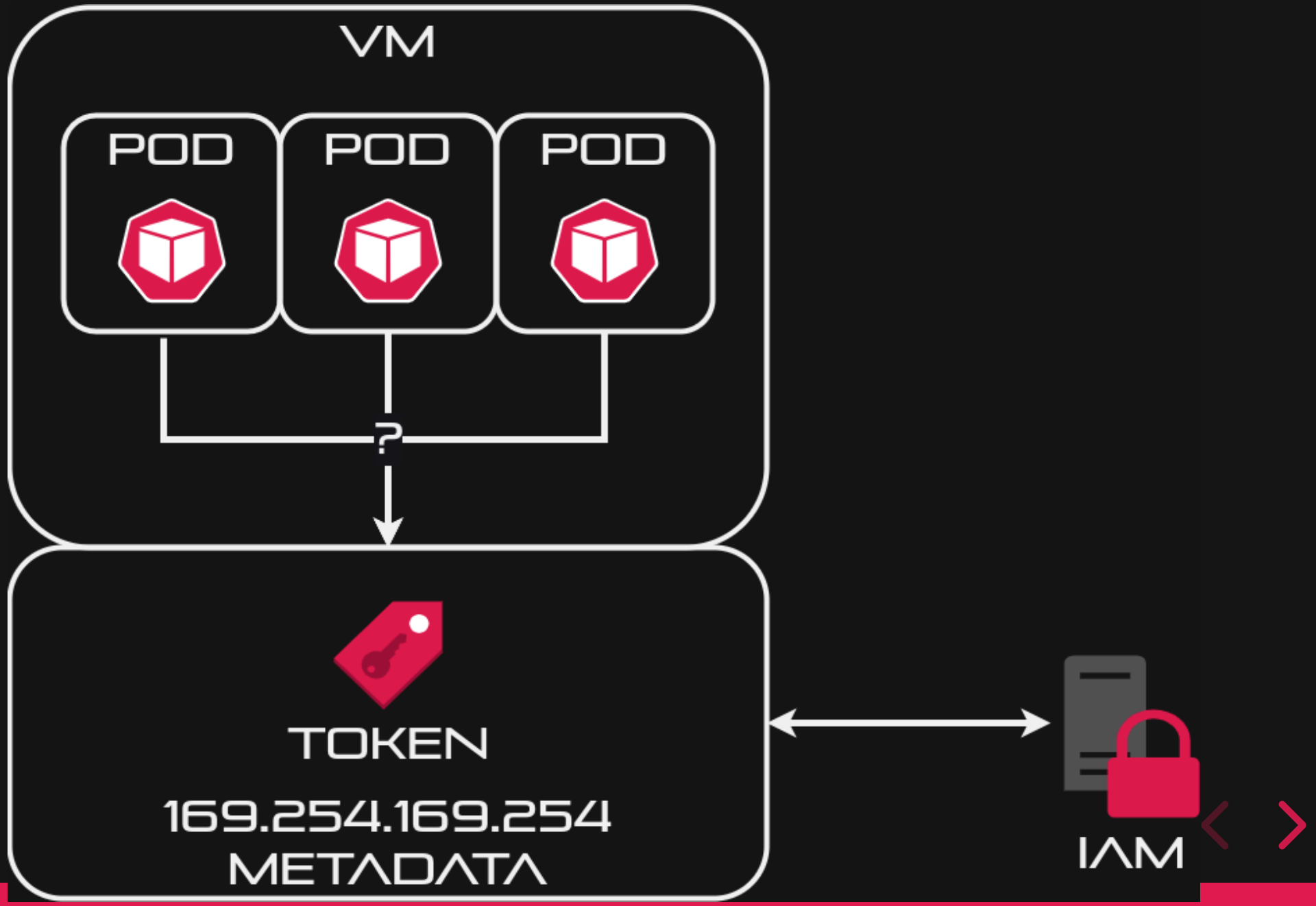
> bash

```bash
ACCESS_TOKEN=$(curl -s -H "Metadata-Flavor:Google" http://metadata.google.interna
curl -s -H "Authorization: Bearer $ACCESS_TOKEN" "https://storage.googleapis.com/
```

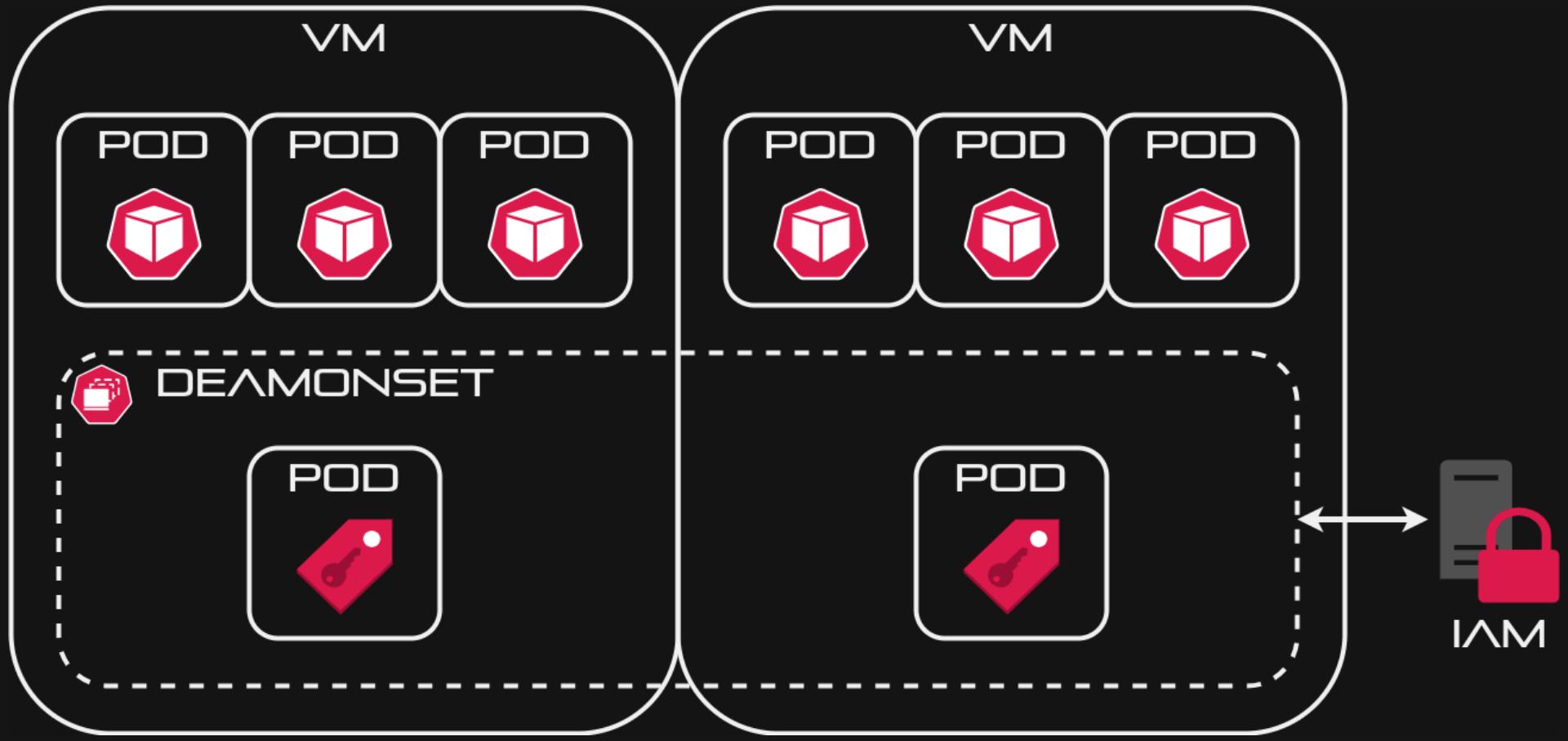
L'API METADATA



SUR KUBE



METADATA DAEMON SET



yaml

```yaml
apiVersion: v1
kind: Namespace
metadata:
  name: demo-devoxx-2024
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: k8s-demo-app-sa
  namespace: demo-devoxx-2024
---
apiVersion: v1
kind: Pod
metadata:
  name: demo-app
  namespace: demo-devoxx-2024
spec:
  containers:
    - name: gcloud
      image: gcr.io/google.com/cloudsdktool/google-cloud-cli:latest
      imagePullPolicy: Always
      stdin: true
```

> bash

```bash
k apply -f k8s/demo-app.yml

k9s -n demo-devoxx-2024 -c pod
```

DÉMO SUR K8S

> bash

```
curl -s -H "Metadata-Flavor:Google" http://metadata.google.internal/computeMetada
```

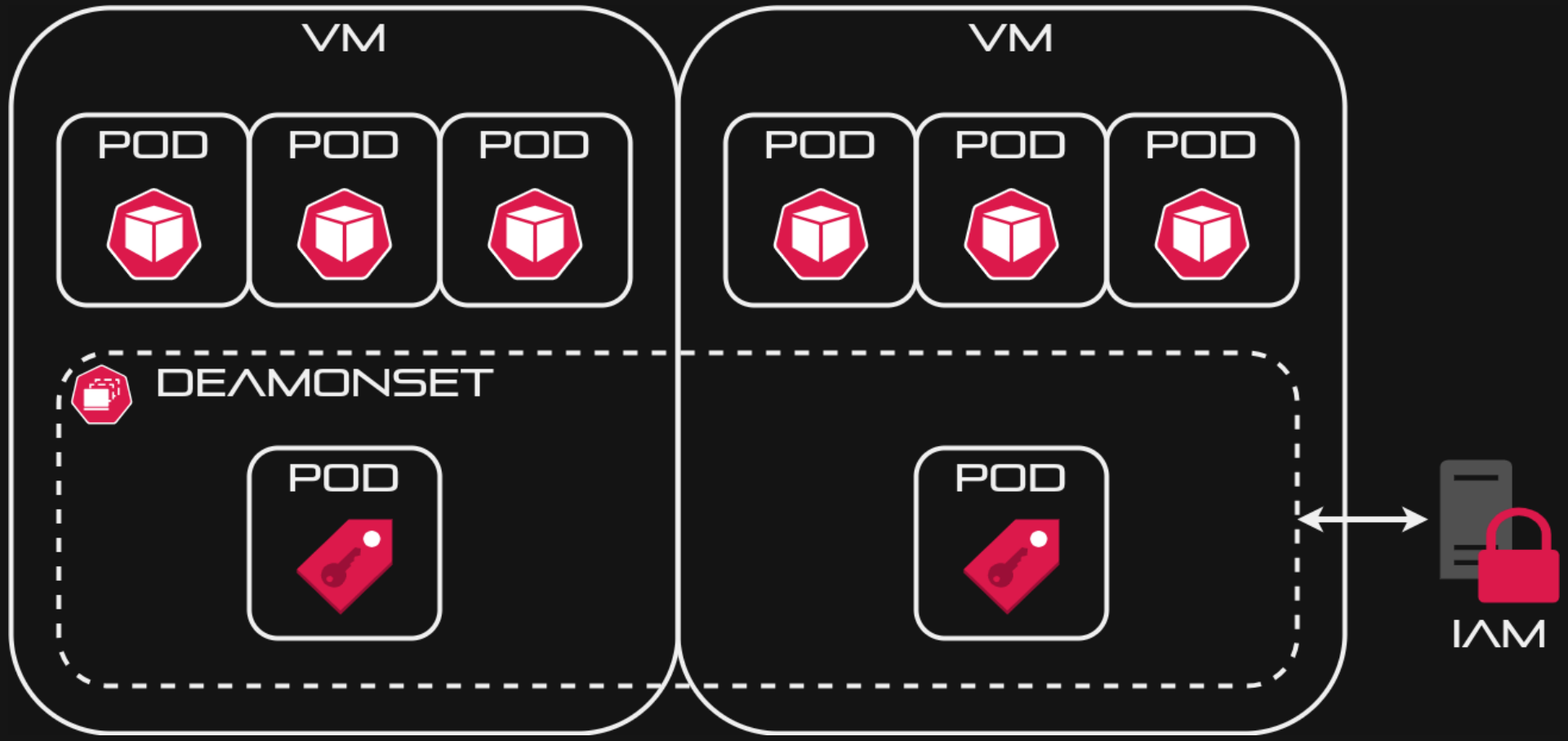
> bash

```
# on attache donne les droits au SA K8S
gcloud projects add-iam-policy-binding devoxx-2024 \
--member=principal://iam.googleapis.com/projects/623887975779/locations/global/wo
--role=roles/storage.admin
```

> bash

```
k exec -n demo-devoxx-2024 -it pod-demo-app -- bash
```

METADATA DAEMON SET



CONCLUSION

RIEN DE MAGIQUE 

CONCLUSION

RIEN DE MAGIQUE 🪄

- infra 💙 code

CONCLUSION

RIEN DE MAGIQUE 🪄

- infra 💙 code
- ⚠️ vendor lock-in (mono-cloud)

CONCLUSION

RIEN DE MAGIQUE 🪄

- infra 💙 code
- ⚠️ vendor lock-in (mono-cloud)
- pas toujours besoin d'un 🔓 vault

CONCLUSION

RIEN DE MAGIQUE 🪄

- infra 💙 code
- ⚠️ vendor lock-in (mono-cloud)
- pas toujours besoin d'un 🔓 vault
- ⚠️ tout le monde peut lire les metadata ⏩ least-privilege


MERCI !

JULIEN WITTOUCK

@CodeKaio

julien-wittouck