CHEOPS TECHNOLOGY

The Cloud & Data Centric Company

Arrêtez le Full Admin dans k8s !

Mettez de l'OIDC dans votre vie...

Qui suis-je?

Philippe Durand
Consultant K8S, IA, Performance applicative et infrastructure
@ Cheops Technology

philippe-durand-work

J’accompagne les clients « From Zero to Hero »

CHEOPS TECHNOLOGY
The Cloud & Data Centric Company

Authentification vs Autorisation

Authentification

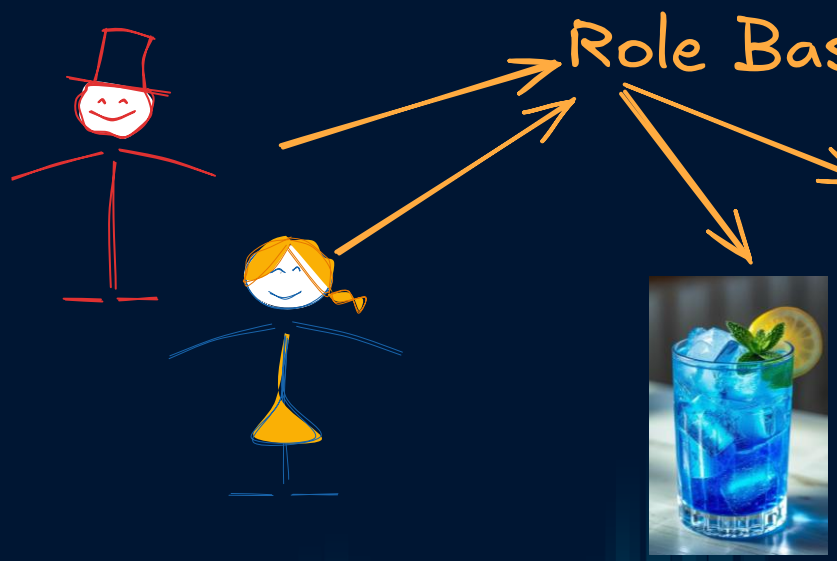
Le serveur me croit quand je dis être Philippe Durand.

Multi Factor Authentication

Authentification vs Autorisation

Autorisation :

Philippe a le droit de lecture seule.



Authentification

Avec des certificats

Authentification

Certificats Utilisateur

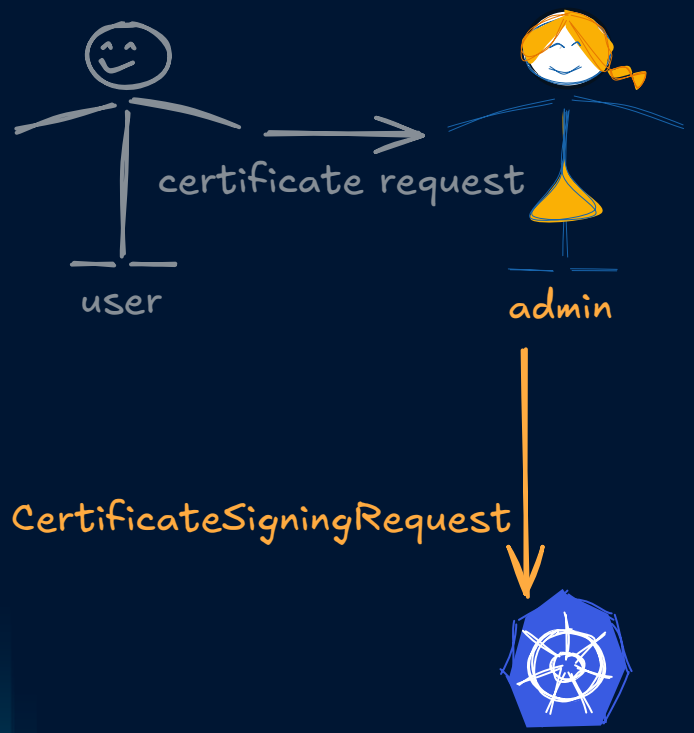
Terminal

```
$  openssl genrsa -out philippe.key 2048
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQC4wn3Oqxx5J5ir
JFD4fkK+1ul5sXsSYkN5S2RoCq67kbb4ucMJqdjtI6bYzyOWV8cj4AR3/TThCcTh
6pPWZz5BxW+K32xvWJLB+j3R
-----END PRIVATE KEY-----


$  openssl req -new -key philippe.key -subj "/CN=philippe.durand" -out philippe.csr
-----BEGIN CERTIFICATE REQUEST-----
MIICXzCCAUcCAQAwGjEYMBYGA1UEAwwPcGhpbGlwcGUuZHVyYW5kMIIBIjANBgkq
hkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAuMJ9zqsceSeYqyRQ+H5CvtbpebF7EmJD
6oD0dvRnFEYKQNKYVhprihVITKT8g9TMbKJga9SAinc7sdM=
-----END CERTIFICATE REQUEST-----
```


CHEOPS TECHNOLOGY
The Cloud Customized For You !

Authentification

Certificats Utilisateur



Terminal

```
$  cat philippe.csr | base64 | tr -d '\n'; echo ;
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSBSRVFVRVNULS0tLS0KTUlJQ1h6Q0NBVWNDQVFBd0dqRVlNQllHQTFVRUF
3d1BjR2hwYkdsd2NHVXVaSFZ5WVc1a01JSUJakFOQmdrcQpoa2lHOXcwQkFRUZBQWluYzdzZE09Ci0tLS0tRU
5EIENFUlRJRklDQVRFIFJFUVVFU1QtLS0tLQo=
```

csr.yaml

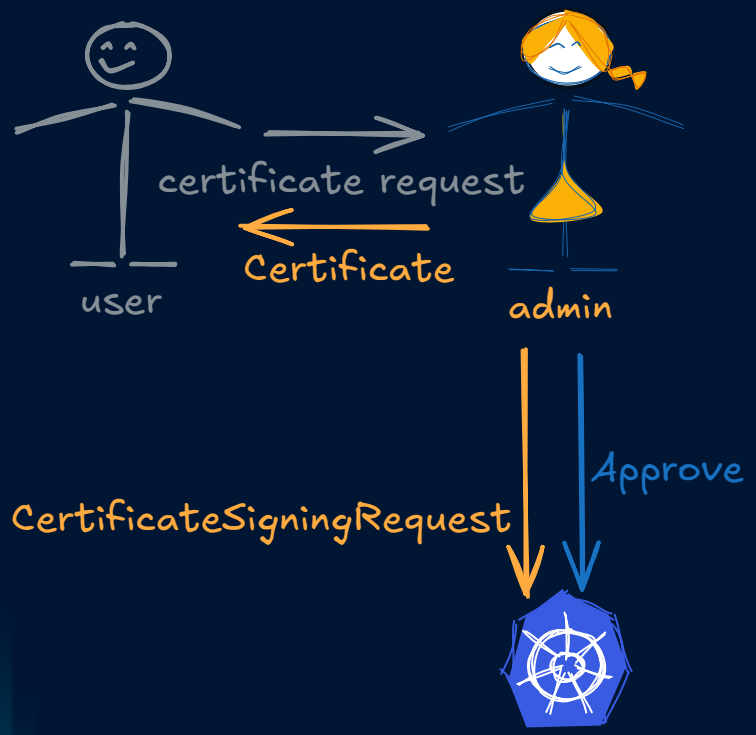
```yaml
apiVersion: certificates.k8s.io/v1
kind: CertificateSigningRequest
metadata:
  name: philippe-durand
spec:
  request: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSBSRVFVRVNULS0tLS0KTUlJQ1h6Q0NBVWNDQVFBd0dqRVlNQ
  signerName: kubernetes.io/kube-apiserver-client
  expirationSeconds: 86400
  usages:
  - client auth
```

Terminal

```
$  k apply -f csr.yaml
certificatesigningrequest.certificates.k8s.io/philippe-durand created
$  k get csr philippe-durand
NAME              AGE   SIGNERNAME                            REQUESTOR   REQUESTEDDURATION   CONDITION
philippe-durand   68s   kubernetes.io/kube-apiserver-client   admin       24h                 Pending
```

Authentification

Certificats Utilisateur



Terminal

```
$  k certificate approve philippe-durand

certificatesigningrequest.certificates.k8s.io/philippe-durand approved

$  k get csr philippe-durand -o jsonpath="{.status.certificate}"; echo;

LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUNQekNDQWVXZ0F3SUJBZ0lSQU0vRW92V0xGN0YwNVpltCZ
2dxaGtqT1BRUURBZ05JQURCRkFpQk52NlNrZjJESEpha2c5clgydC96SwpyZE5XSS8ydytmRkMvM1JoeWJsMXVB
SWhBTXdUV21uOXB1U1UzTk1qY0ZaOVpzQVZ5YlJDQVJFdFhySC9zSVNPCkJNZVAKLS0tLS1FTkQgQ0VSVElGSUN
BVEUtLS0tLQo=

$  k get csr philippe-durand -o jsonpath="{.status.certificate}" | base64 -d

-----BEGIN CERTIFICATE-----
MIICPzCCAeWgAwIBAgIRAM/EovWLF7F05ZeR0e8ltNowCgYIKoZIzj0EAwIwFTET
MBEGA1UEChMKa3ViZXJuZXRlczAeFw0yNjAyMDgwNjU0MTlaFw0yNjAyMDkwNjU0
MTlaMBoxGDAWBgNVBAMTD3BoaWxpcHBlLmR1cmFuZDCCASIwDQYJKoZIhvcNAQEB
BQADggEPADCCAQoCggEBALjCfc6rHHknmKskUPh+Qr7W6XmxexJiQ3lLZGgKrruR
tvi5wwmp2O0jptjPI5ZXxyPgBHf9NOEJxOGNvFCkmnofupGGmYsi1CmLPS72QJVL
Vm7bCgzMgwJj3zaP0Vnzh+SLAP3ztzM+8wzaxa+Wb3Dv+eGUofSiFfr0VztLhLmP
/7Dyx0ZJb0983poVySVbmSiV3BEf62lO7PoCXci6FNqpjHX5llYTDqH+1irQjvcT
VyOHRSgUF0UaoZrdyl973R2RwsNwO/bf1ZZ8YiK5RkEbTYttjL3NvsqMBMp99act
rJv+Yrt0vKaX1VKicmFm5M2wclv3alcggOE2o5T+sXECAwEAAaNGMEQwEwYDVR0l
BAwwCgYIKwYBBQUHAwIwDAYDVR0TAQH/BAIwADAfBgNVHSMEGDAWgBT7uHnXAqiD
0zFR0GvObWbtGLcyZTAKBggqhkjOPQQDAgNIADBFAiBNv6Skf2DHJakg9rX2t/zK
rdNWI/2w+fFC/3Rhybl1uAIhAMwTWmn9puSU3NMjcFZ9ZsAVybRCAREtXrH/sISO
BMeP
-----END CERTIFICATE-----
```


CHEOPS TECHNOLOGY
The Cloud Customized For You !

Authentification

Certificats Utilisateur

~/.kube/config

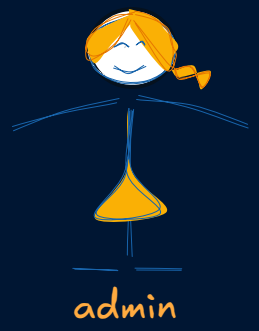
```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLSW8xQXhFblU3RmkrcXhpUzlQUG5UMGpFWEg2M3VQBDRVJUSUZJQ0FURS0tLS0tCg==
    server: https://10.5.0.2:6443
  name: talos-debian-docker
contexts:
- context:
    cluster: talos-debian-docker
    namespace: default
    user: admin@talos-debian-docker
  name: admin@talos-debian-docker
- context:
    cluster: talos-debian-docker
    namespace: default
    user: philippe-durand@talos-debian-docker
  name: philippe-durand@talos-debian-docker
current-context: admin@talos-debian-docker
kind: Config
users:
- name: admin@talos-debian-docker
  user:
    client-certificate-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0mRZbz0KLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo=
    client-key-data: LS0tLS1CRUdJTiBFQyBQUklWQVRFIEtFWS0tLS0tCkKy0tLS1FTkQgRUMgUFJJVkFURSBLRVktLS0tLQo=
- name: philippe-durand@talos-debian-docker
  user:
    client-certificate-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCkVJFdFhySC9zSQ0VSVElGSUNBVEUtLS0tLQo=
    client-key-data: LS0tLS1CRUdJTiBQNnBQV1p6NUJ4VLMzJ4dldKTEIrajNSCi0tLS0tRU5EIFBSSVZBVEUgS0VZLS0tLS0K
```

Authentification

Certificats Utilisateur

Terminal

```
$  k --context philippe-durand@talos-debian-docker auth whoami
ATTRIBUTE                                          VALUE
Username                                           philippe.durand
Groups                                             [system:authenticated]
Extra: authentication.kubernetes.io/credential-id  [X509SHA256=f69611bd084675aba59747d756125f98c5c77e03f3c52c79471c2dbab85531a2]
$  k --context philippe-durand@talos-debian-docker auth can-i get pods
no
```



Terminal

```
$  k create clusterrolebinding pdu-view --clusterrole view --user philippe.durand
clusterrolebinding.rbac.authorization.k8s.io/pdu-view created
```

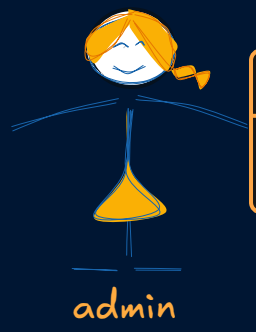
Terminal

```
$  k --context philippe-durand@talos-debian-docker auth can-i get pods
yes
$  k --context philippe-durand@talos-debian-docker auth can-i delete pods
no
```


CHEOPS TECHNOLOGY
The Cloud Customized For You !

Authentification

Certificats Utilisateur - révocation



Terminal

```
> k delete csr philippe-durand
certificatesigningrequest.certificates.k8s.io "philippe-durand" deleted
```

Terminal

```
> k --context philippe-durand@talos-debian-docker auth whoami
ATTRIBUTE                                           VALUE
Username                                            philippe.durand
Groups                                              [system:authenticated]
Extra: authentication.kubernetes.io/credential-id   [X509SHA256=f69611bd084675aba59747d756125f98c5c77e03f3c52c79471c2dbab85531a2]
> k --context philippe-durand@talos-debian-docker auth can-i get pods
yes
```

Authentification

Certificats Utilisateur

En pratique

- On ajoute les groupes dans le certificat csr initial

Terminal

```
> openssl req -new -key philippe.key -subj "/CN=philippe.durand/O=devops/O=platform" -out philippe.csr
```

- On fait le RBAC sur les groupes … c'est quand même plus facile

- On ne demande pas à un utilisateur
  - de faire tout le travail key / csr / kubeconfig
  - de penser à re-certifier sa clé

➔ On va gérer tout ce travail côté admin et produire régulièrement un kubeconfig

Authentification

Résumé

	Certificats
Acteur nominatif	100%
Gestion des groupes	50%
Révocation possible	0%
Simplicité de gestion	20%
Passe les reverse proxy L7 APIServer	0%

CHEOPS TECHNOLOGY
The Cloud Customized For You !

Authentification

Avec des Tokens

Authentification

Token

Idée : utiliser les comptes de service (ServiceAccount)

- On peut générer plusieurs secrets de type token par compte de service
  - Ces tokens n'expirent pas mais sont révoquables en supprimant le secret
- On peut faire du RBAC pour les comptes de service

Organisation possible

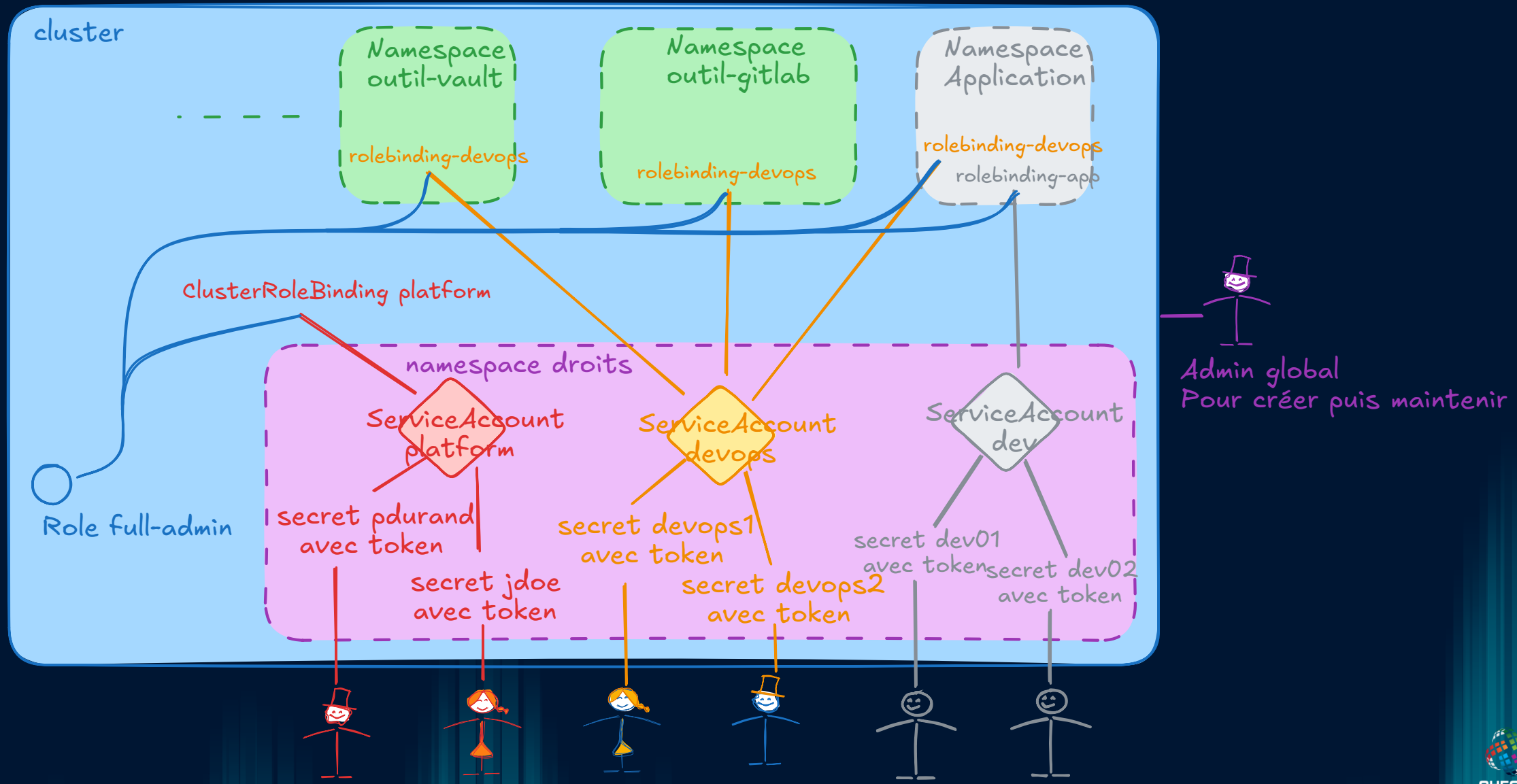
- ServiceAccount == profil métier ➔ plusieurs rôles
- Un token par personne == un identifiant par personne

Industrialisation

- Il est possible d'industrialiser la création / modification des comptes avec des outil comme, par exemple, helm.

Authentification

Token – organisation Démo



CHEOPS TECHNOLOGY
The Cloud Customized For You !

Authentification

Token : Demo time !

Terminal

```
$  k create clusterrole full-admin --verb='*' --resource='*'
clusterrole.rbac.authorization.k8s.io/full-admin created
$  k create ns droits; k create ns outil-vault; k create ns outil-gitlab; k create ns application
namespace/droits created
namespace/outil-vault created
namespace/outil-gitlab created
namespace/application created
$  k -n droits create sa platform; k -n droits create sa devops; k -n droits create sa dev
serviceaccount/platform created
serviceaccount/devops created
serviceaccount/dev created
$  k create clusterrolebinding platform --clusterrole cluster-admin --serviceaccount droits:platform
clusterrolebinding.rbac.authorization.k8s.io/platform created
$  k -n droits apply -f serviceaccount-secret-pdurand.yaml; k -n droits apply -f serviceaccount-secret-jdoe.yaml
secret/pdurand created
secret/jdoe created
$  k -n droits get secret pdurand -o jsonpath='{.data.token}' | base64 -d; echo;
eyJhbGciOiJSUzI1NiIsImtpZCI6IkRsTWhWcnNIci1yVTJxY3VhYXpGUG9WdVNDNHdwTUdza1BSYXh5Nld5YnMifQ.pLlu3LDQuoKGWpZKEzRsqvFSHUmpkcyefM_kOsQv2NH2zxSHBan1rM9sCl2IH1lXQNhcpKlRQoIPh_TMfKIJooyMY2AtWzbhY6SNBYxJ3am7TVyQxkGFj1-rOGGDQTUcO6pbc6sVLmlI
```

serviceaccount-secret-pdurand.yaml

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: pdurand
  namespace: droits
  annotations:
    kubernetes.io/service-account.name: "platform"
type: kubernetes.io/service-account-token
```

serviceaccount-secret-jdoe.yaml

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: jdoe
  namespace: droits
  annotations:
    kubernetes.io/service-account.name: "platform"
type: kubernetes.io/service-account-token
```

Authentification

Token : Demo time !

~/.kube/config

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLSW8xQXhFblU3RmkrcXhpUzlQUG5UMGpFWEg2M3VQBDRVJUSUZJQ0FURS0tLS0tCg==
    server: https://10.5.0.2:6443
  name: talos-debian-docker
contexts:
...
- context:
    cluster: talos-debian-docker
    namespace: default
    user: philippe-durand@talos-debian-docker
  name: philippe-durand@talos-debian-docker
...
current-context: admin@talos-debian-docker
kind: Config
users:
...
- name: philippe-durand@talos-debian-docker
  user:
    client-certificate-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCkVJFdFhySC9zSQ0VSVElGSUNBVEUtLS0tLQo=
    client-key-data: LS0tLS1CRUdJTiBQNnBQV1p6NUJ4VLMzJ4dldKTEIrajNSCi0tLS0tRU5EIFBSSVZBVEUgS0VZLS0tLS0K
- name: pdurand@talos-debian-docker
  user:
    token: eyJhbGciOiJSUzI1NiIsImtpZCI6IkRsTWhWcnNIci1yVTJxY3VhYXpGUG9WdVNDNHdwTUdza1BSYXh5Nld5YnMifQ.p
- name: jdoe@talos-debian-docker
  user:
    token: eyJhbGciOiJSUzI1NiIsImtpZCI6IkRsTWhWcnNIci1yVTJxY3VhYXpGUG9WdVNDNHdwTUdza1BSYXh5Nld5YnMifQ.e
```


CHEOPS TECHNOLOGY
The Cloud Customized For You !

Authentification

Token : Demo time !

Terminal

```
$  k --context pdurand@talos-debian-docker auth whoami
ATTRIBUTE   VALUE
Username    system:serviceaccount:droits:platform
UID         e4027fe1-669f-4e2d-afa7-a16c4dc01c9e
Groups      [system:serviceaccounts system:serviceaccounts:droits system:authenticated]
$  k --context pdurand@talos-debian-docker auth can-i get pods
yes
$  k --context pdurand@talos-debian-docker auth can-i delete ns
yes
```

Terminal

```
$  k -n droits delete secret pdurand
secret "pdurand" deleted from droits namespace
```

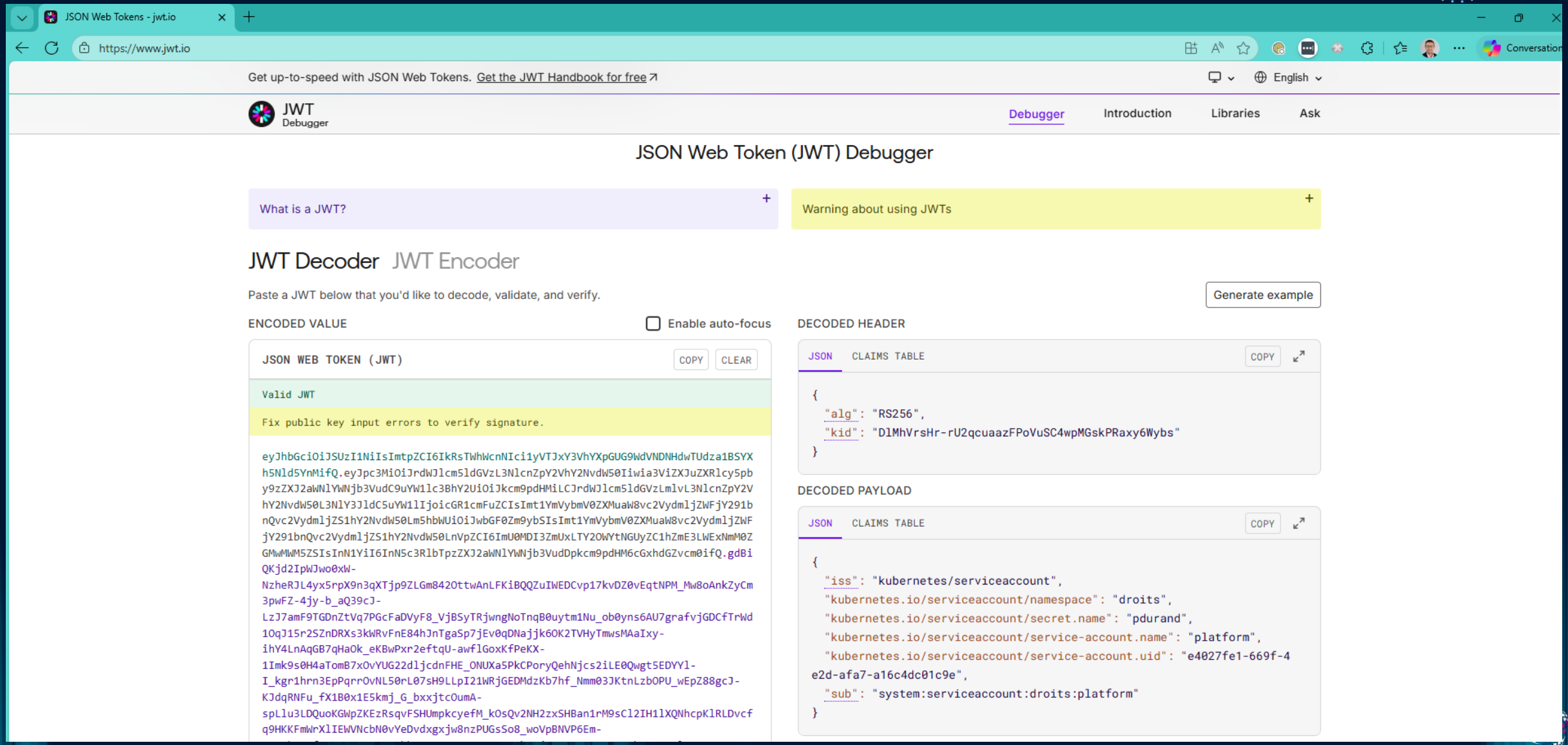
Terminal

```
$  k --context pdurand@talos-debian-docker auth whoami
error: You must be logged in to the server (Unauthorized)
$  k --context jdoe@talos-debian-docker auth can-i delete ns
yes
```


CHEOPS TECHNOLOGY
The Cloud Customized For You !

Authentification

Il contient quoi ce token ???



Authentification

Résumé

	Certificats	Tokens de SA
Acteur nominatif	100%	0%
Gestion des groupes	50%	80%
Révocation possible	0%	100%
Simplicité de gestion	20%	50%
Passe les reverse proxy L7 APIServer	0%	100%

Authentification

Bilan intermédiaire

C’est complexe, non?

Constat : **TROP** de clusters

- Sont accédés avec des droits full-admin (le certificat full-admin a été diffusé)
- Sont accessibles par des personnes ne faisant plus partie de l’entreprise
- Voire, sont accessibles en anonymous authent :
  - User : system:anonymous
  - Group : system:unauthenticated

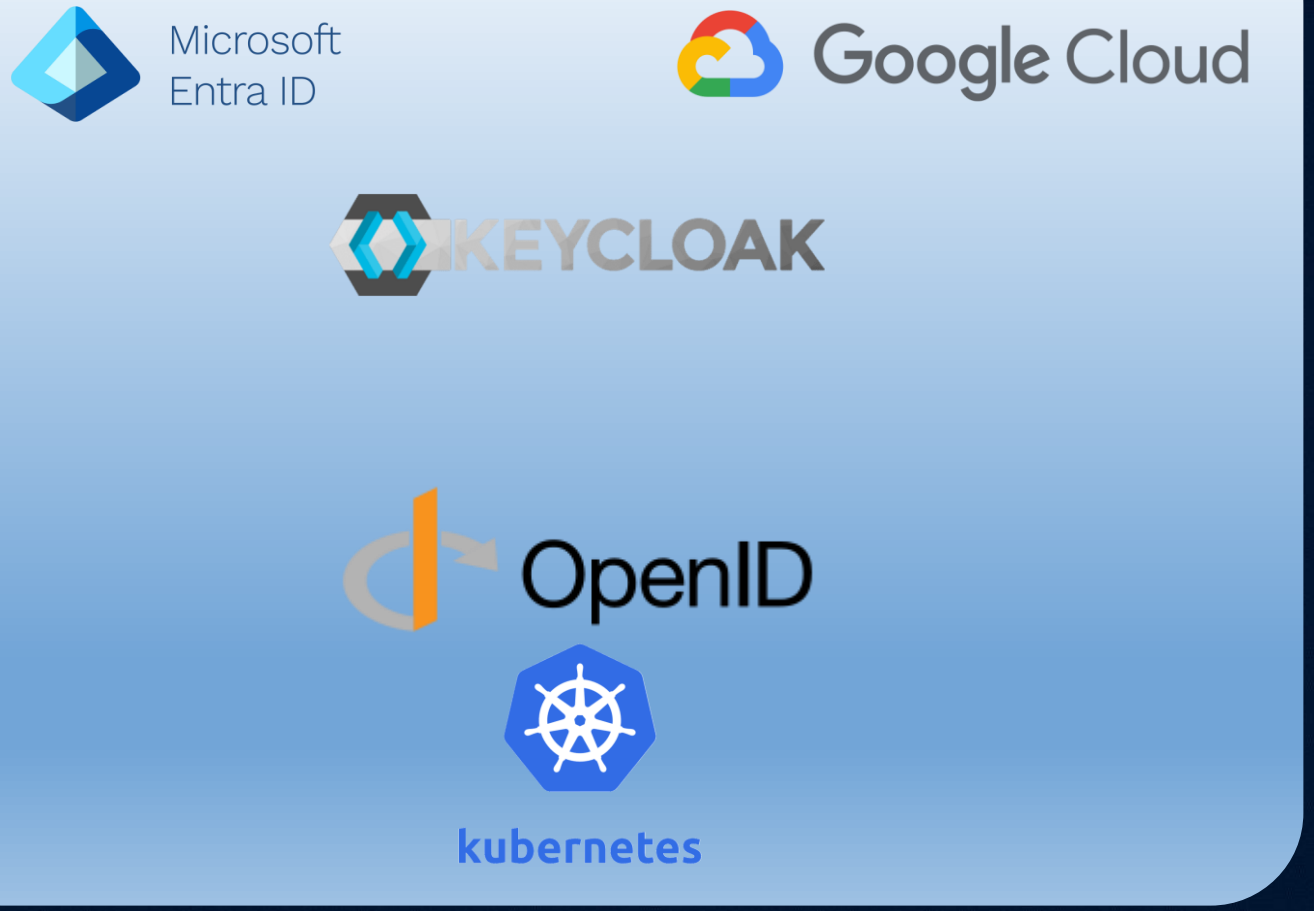
CHEOPS TECHNOLOGY
The Cloud Customized For You !

Authentification

Et maintenant?

What if ???

- K8S faisait confiance à une autorité d'authentification externe ?
- Qui fournirait un token JWT contenant user et des groupes externes au cluster ??
- Et que l'on puisse faire le RBAC sur ces groupes ???



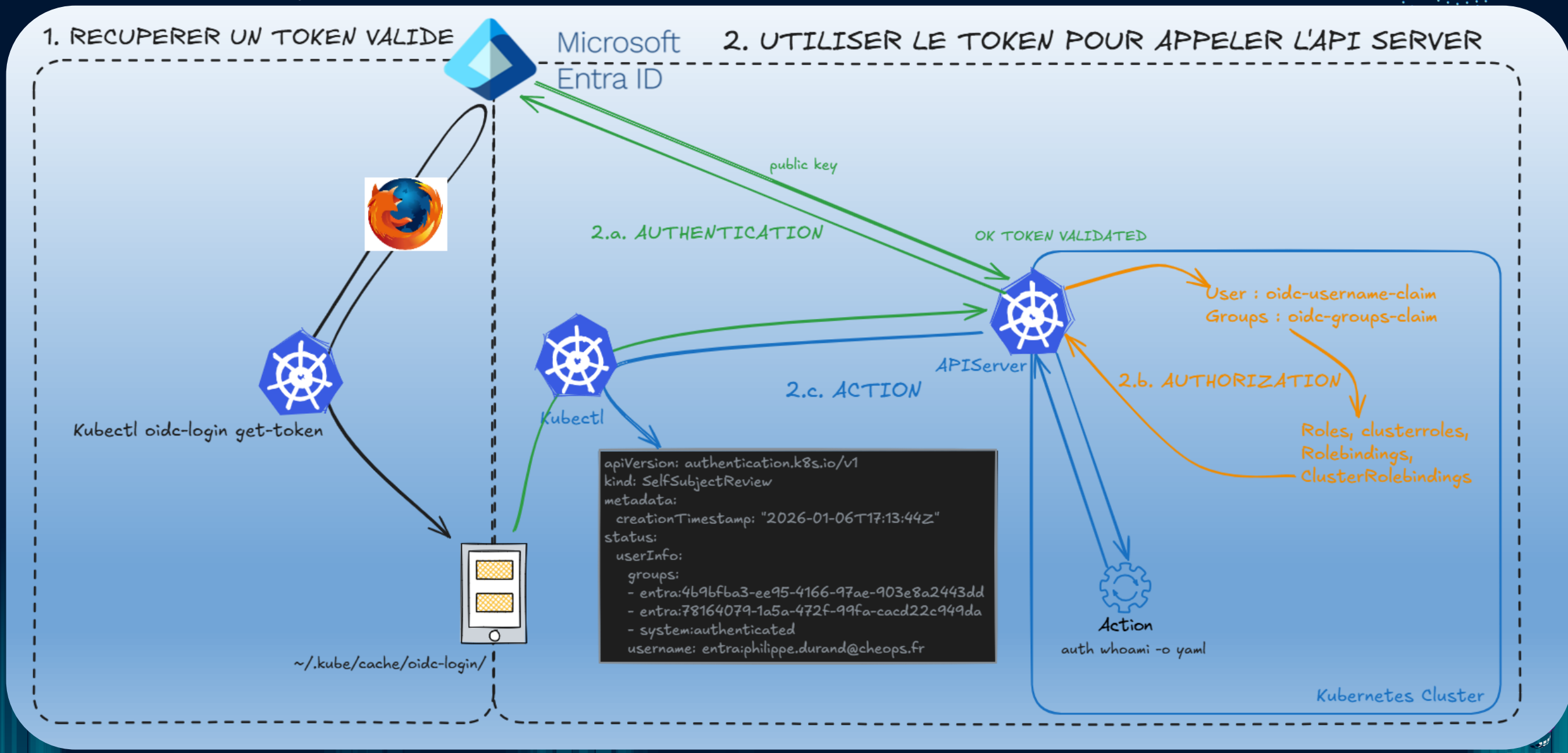
Authentification

OpenID Connect (OIDC)

- Une couche d’identité au-dessus d’OAuth 2.0
- Standard depuis ~2014

K8S OIDC avec Microsoft Entra

Comment ça fonctionne?



CHEOPS TECHNOLOGY
The Cloud Customized For You !

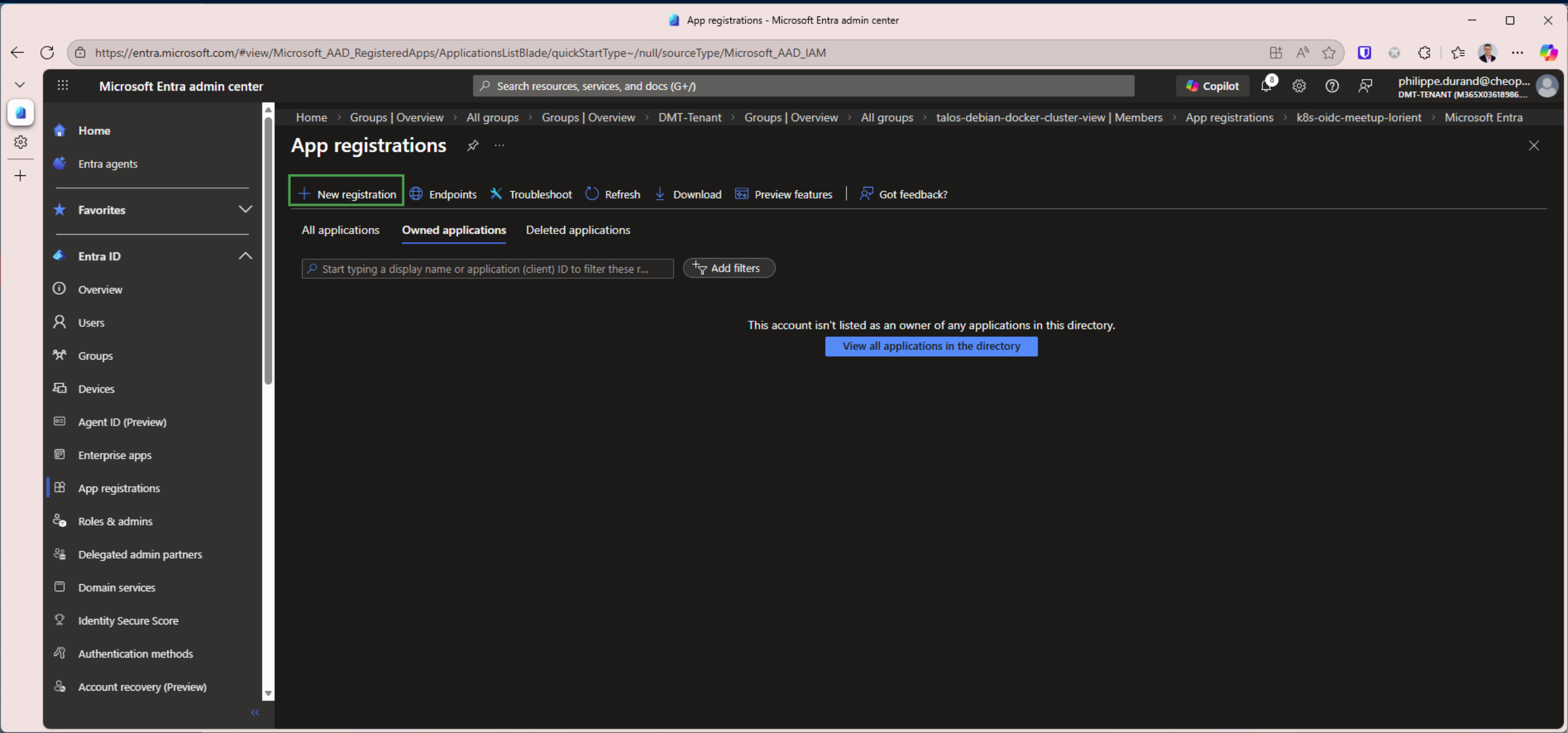
K8S OIDC avec Microsoft Entra

Les étapes de paramétrage ...

CHEOPS TECHNOLOGY
The Cloud Customized For You !

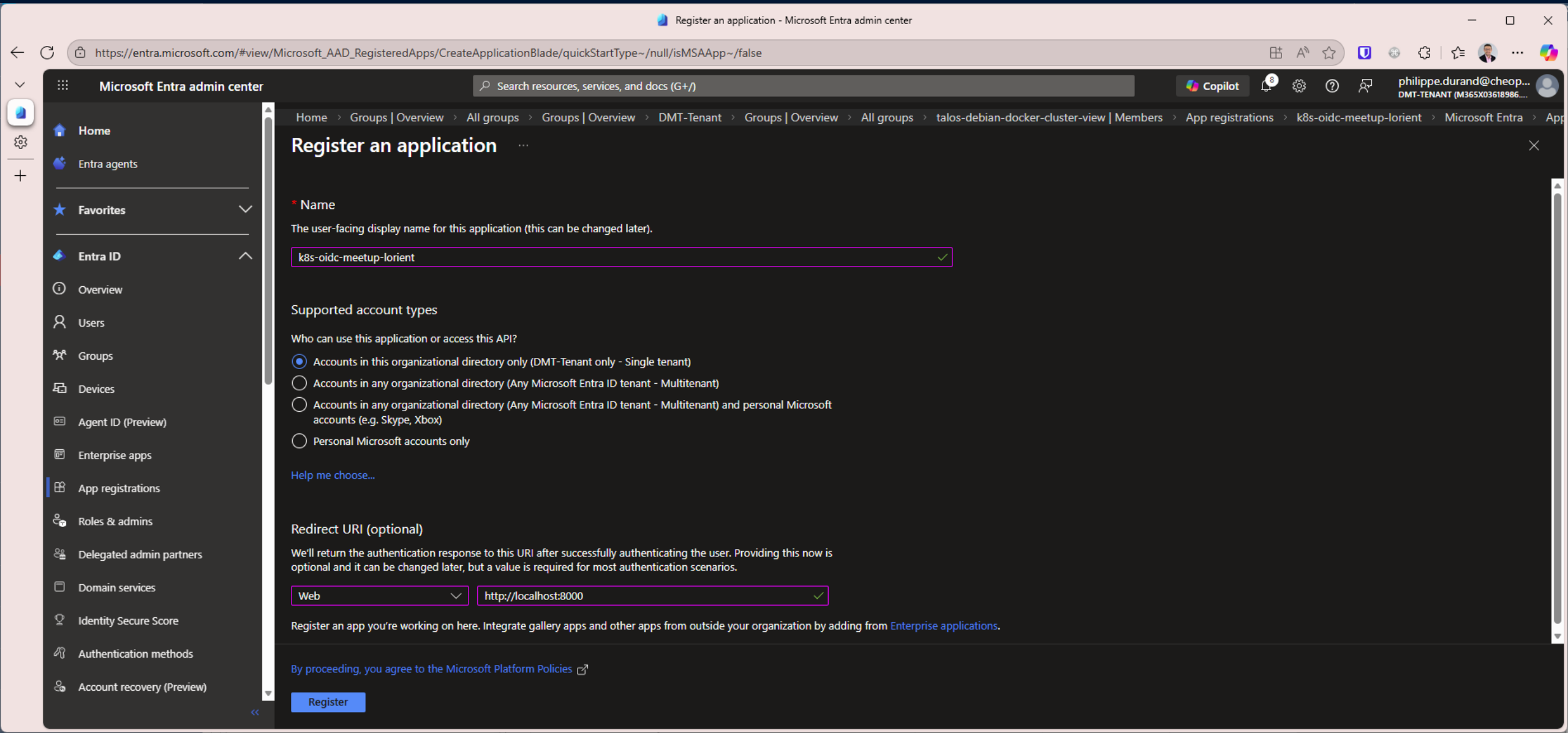
K8S OIDC avec Microsoft Entra

Paramétrage Entra : Créer une application



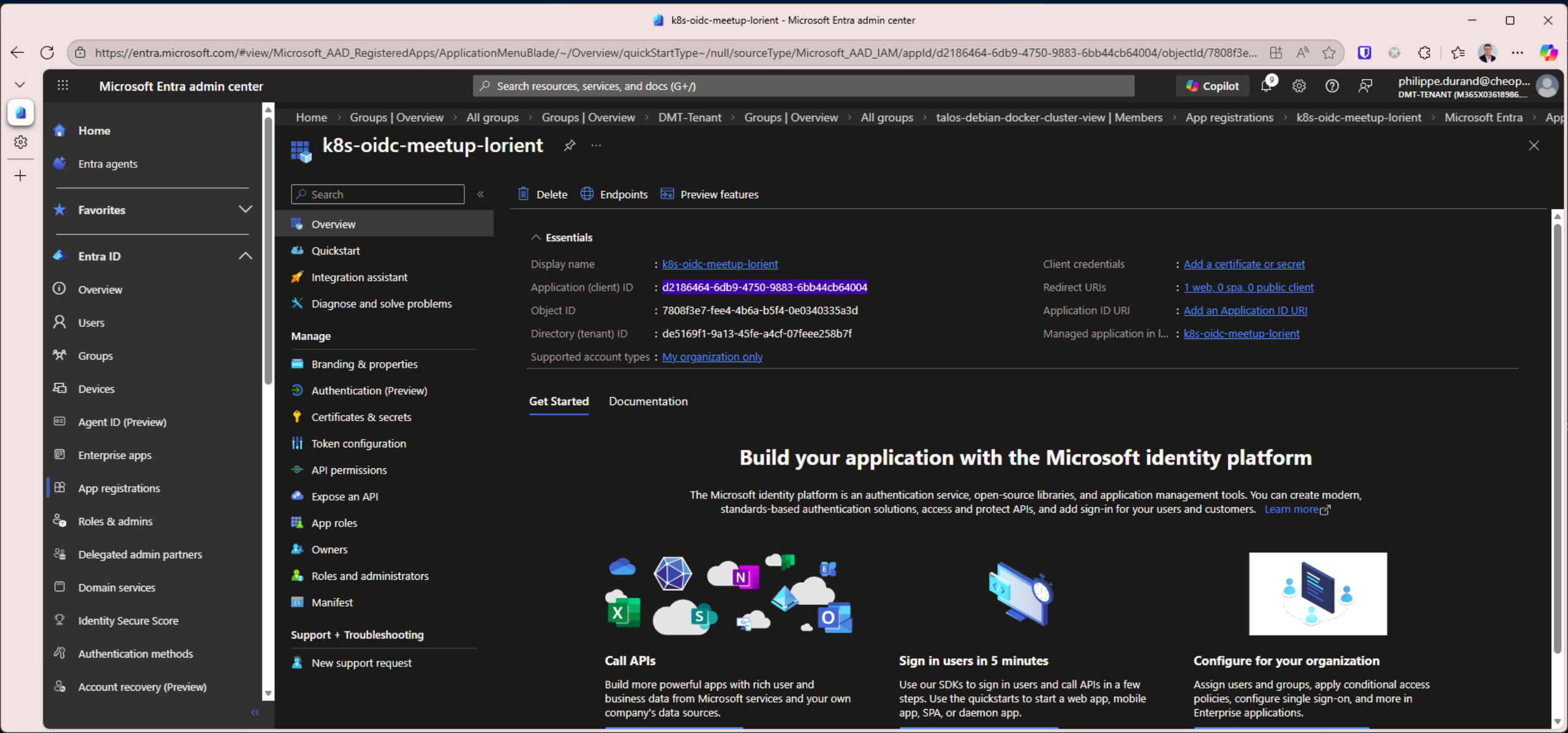
K8S OIDC avec Microsoft Entra

Paramétrage Entra : 1. Créer une application



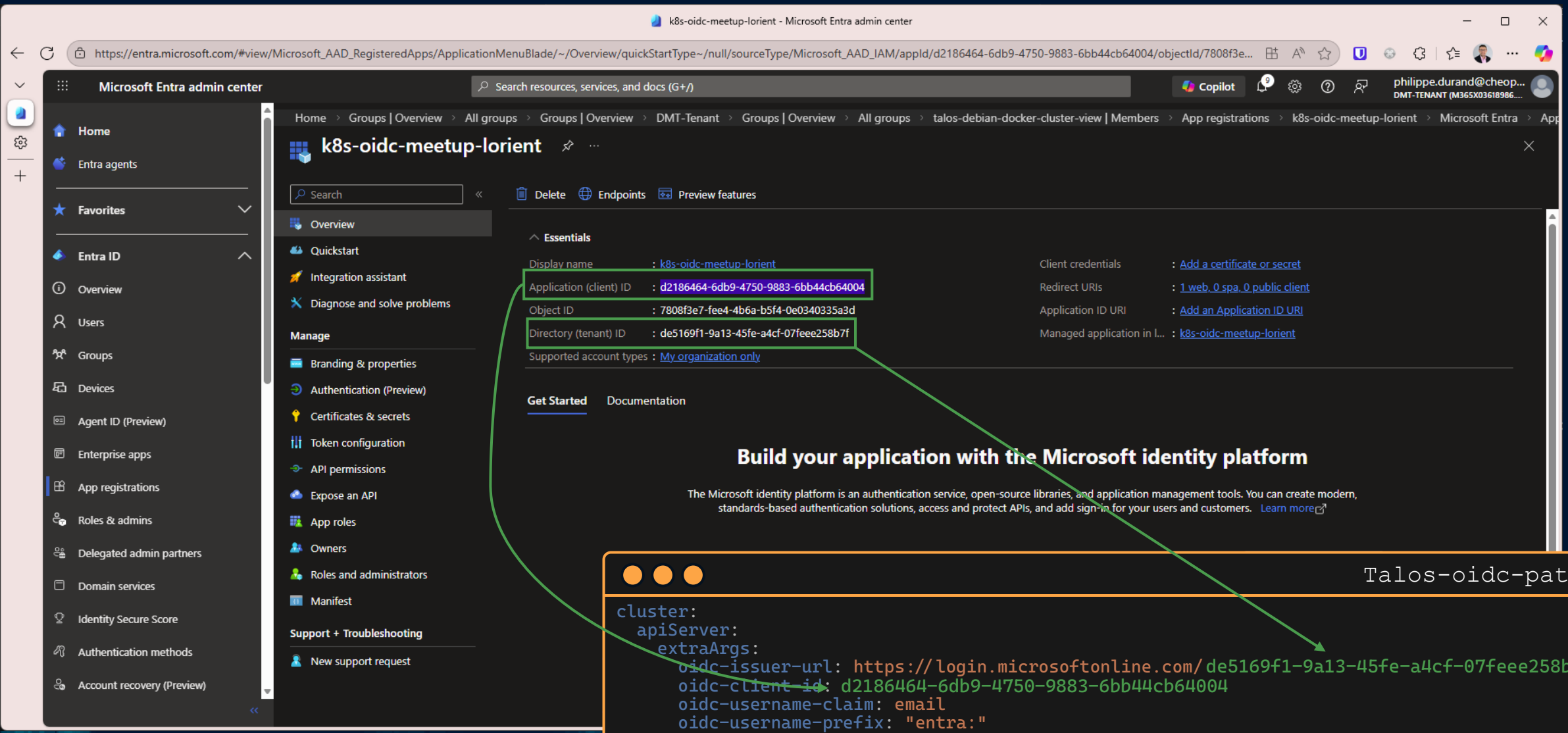
K8S OIDC avec Microsoft Entra

Paramétrage Entra : 1. Créer une application



K8S OIDC avec Microsoft Entra

Paramétrage Entra : Récupérer les identifiants d'application

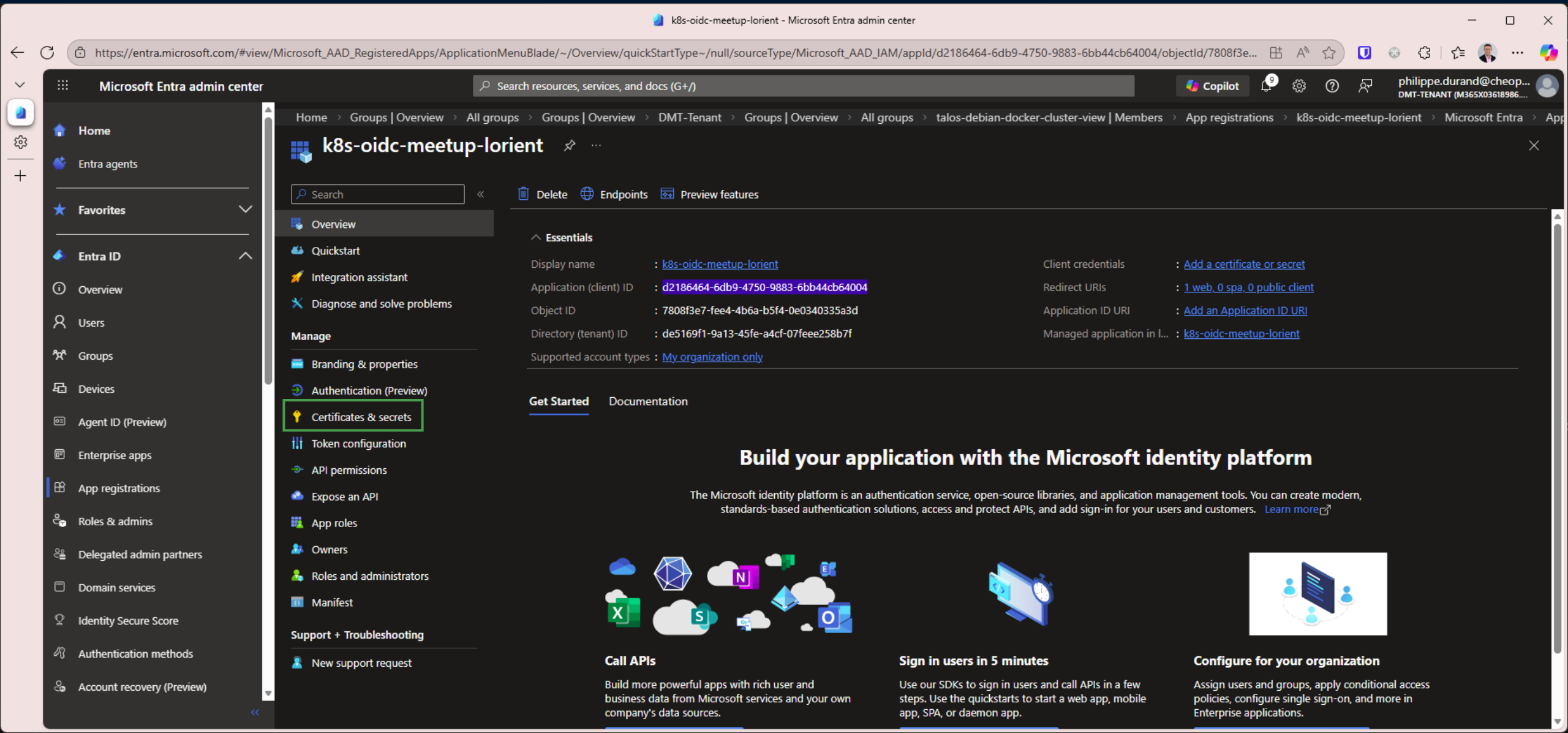


Talos-oidc-patch.yaml

```yaml
cluster:
  apiServer:
    extraArgs:
      oidc-issuer-url: https://login.microsoftonline.com/de5169f1-9a13-45fe-a4cf-07feee258b7f/v2.0
      oidc-client-id: d2186464-6db9-4750-9883-6bb44cb64004
      oidc-username-claim: email
      oidc-username-prefix: "entra:"
      oidc-groups-claim: groups
      oidc-groups-prefix: "entra:"
```

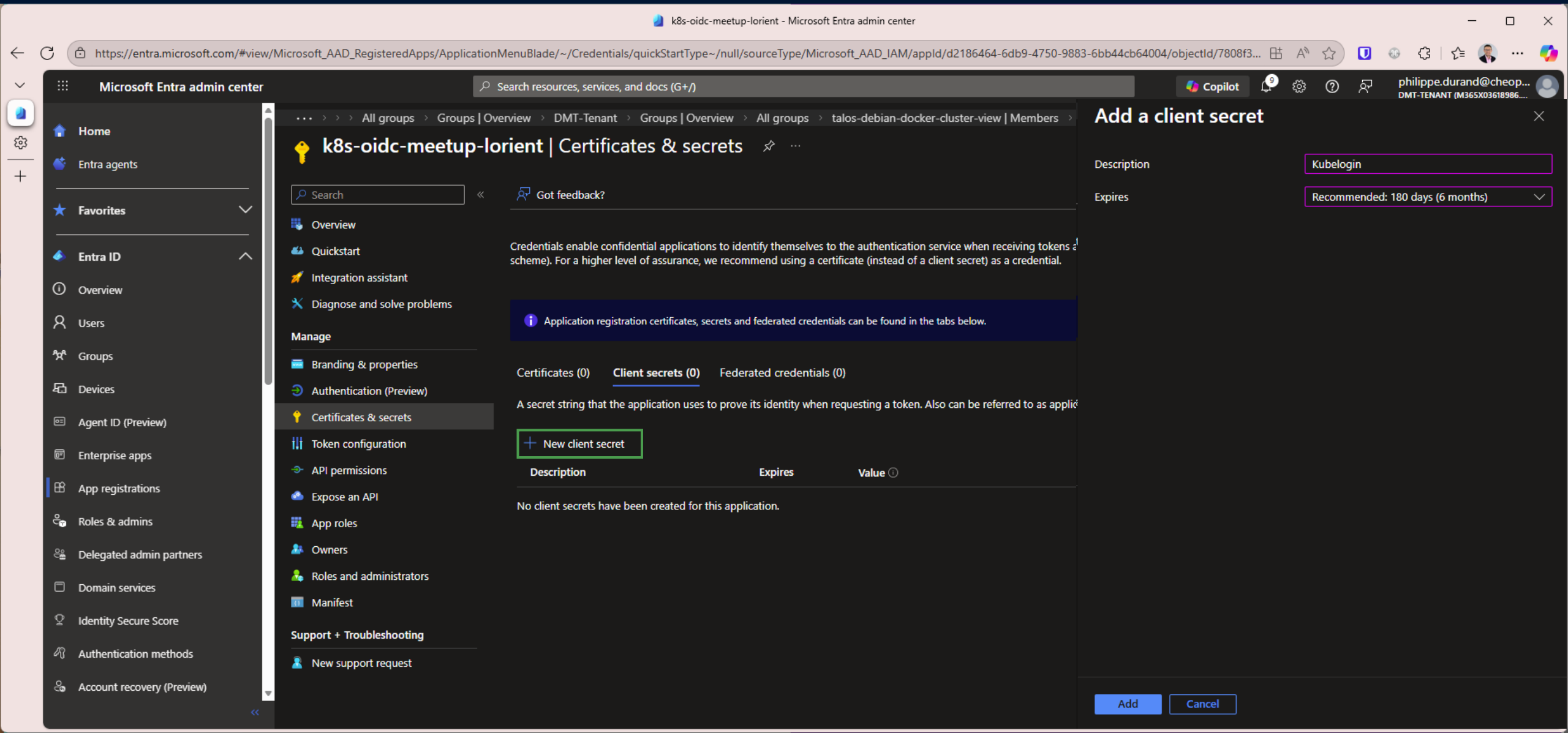
K8S OIDC avec Microsoft Entra

Paramétrage Entra : Ajouter un secret à l'application



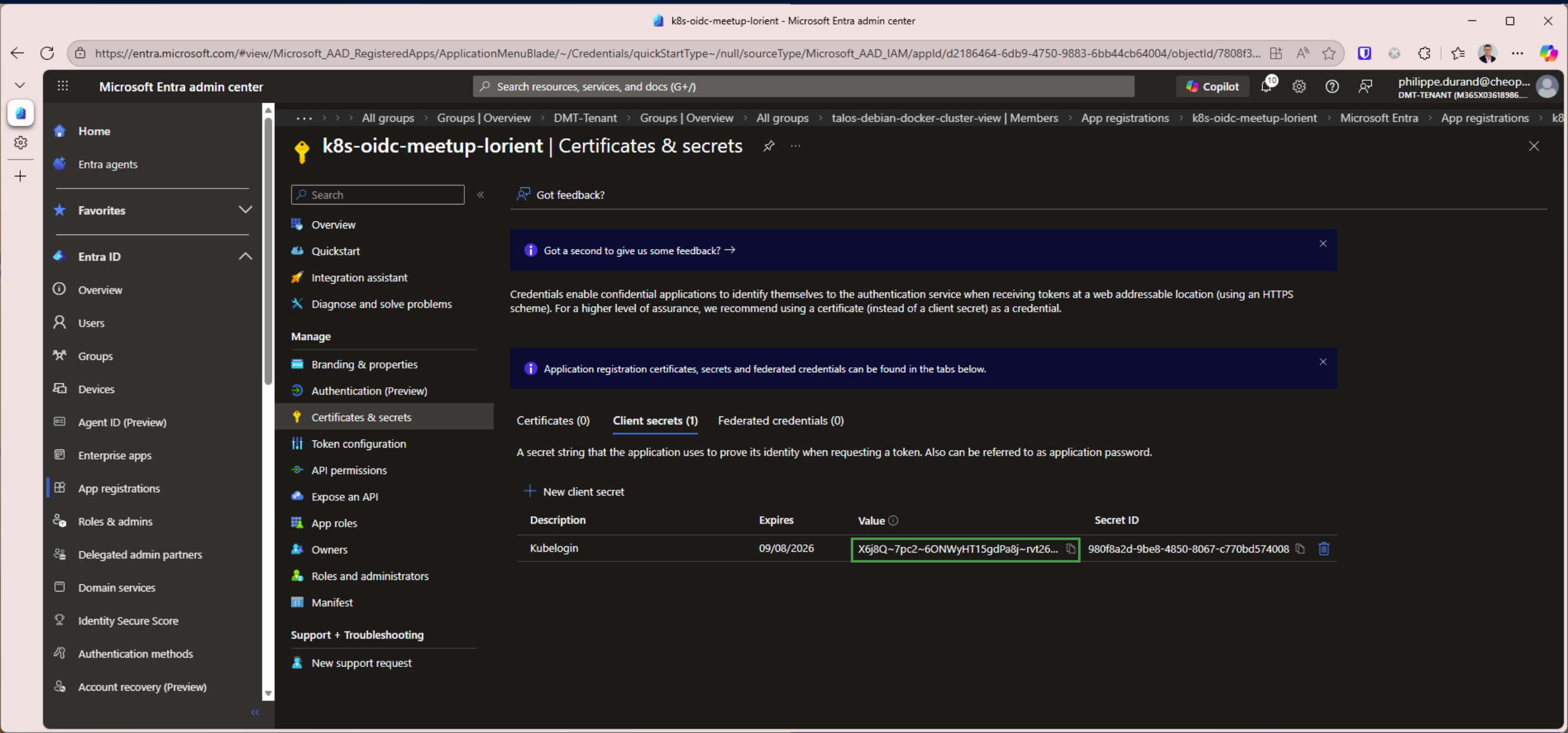
K8S OIDC avec Microsoft Entra

Paramétrage Entra : Ajouter un secret à l'application



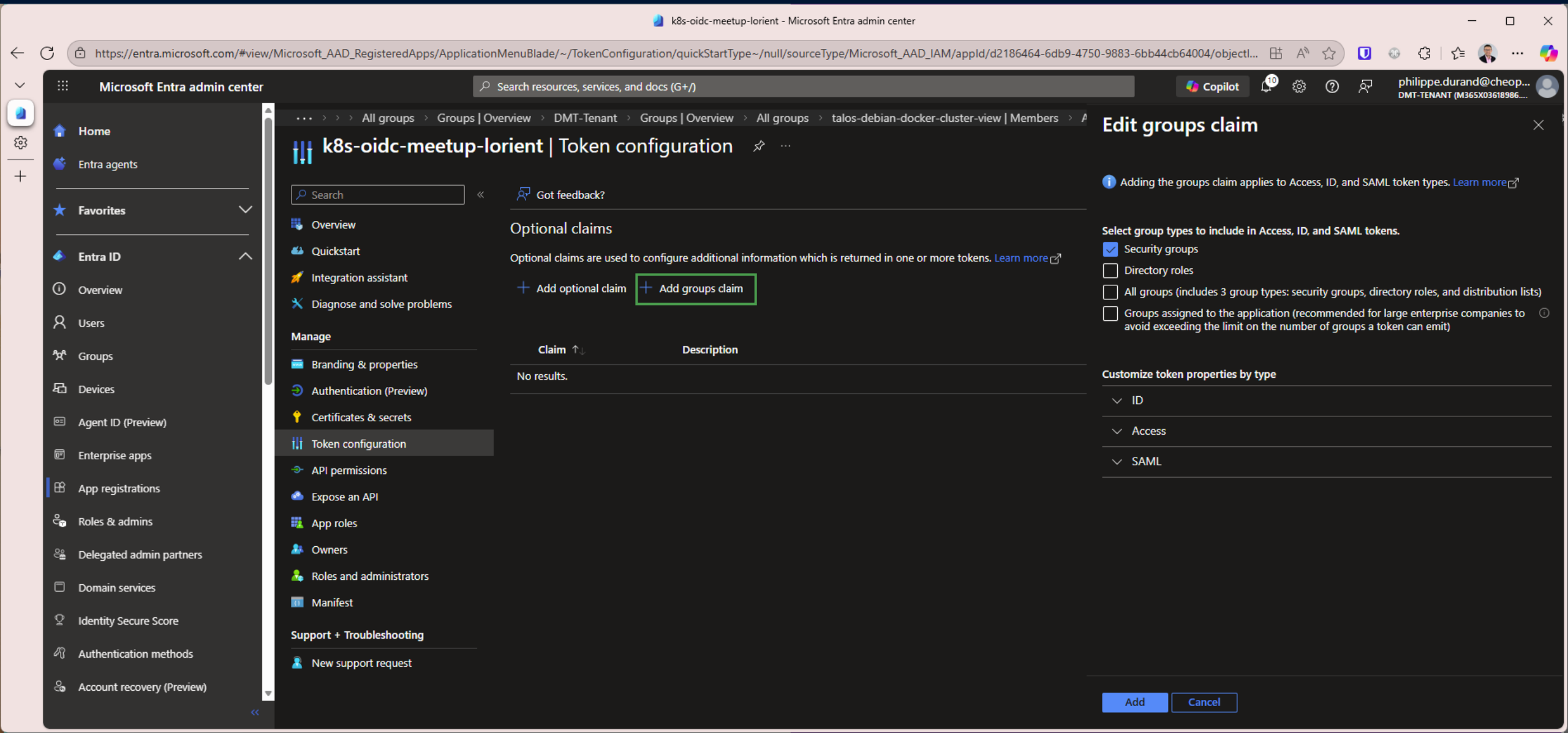
K8S OIDC avec Microsoft Entra

Paramétrage Entra – Récupérer le secret



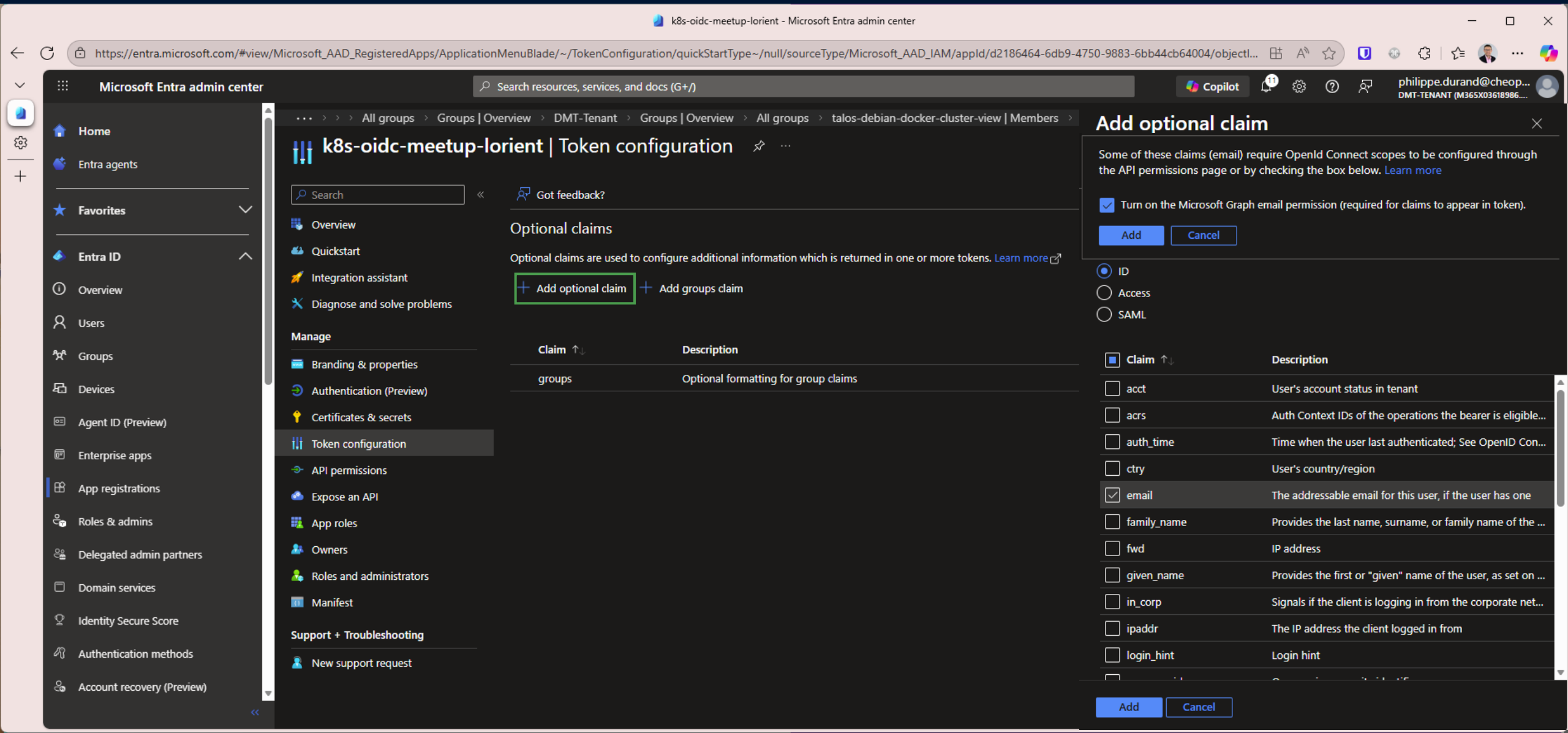
K8S OIDC avec Microsoft Entra

Paramétrage Entra – Paramétrage du token



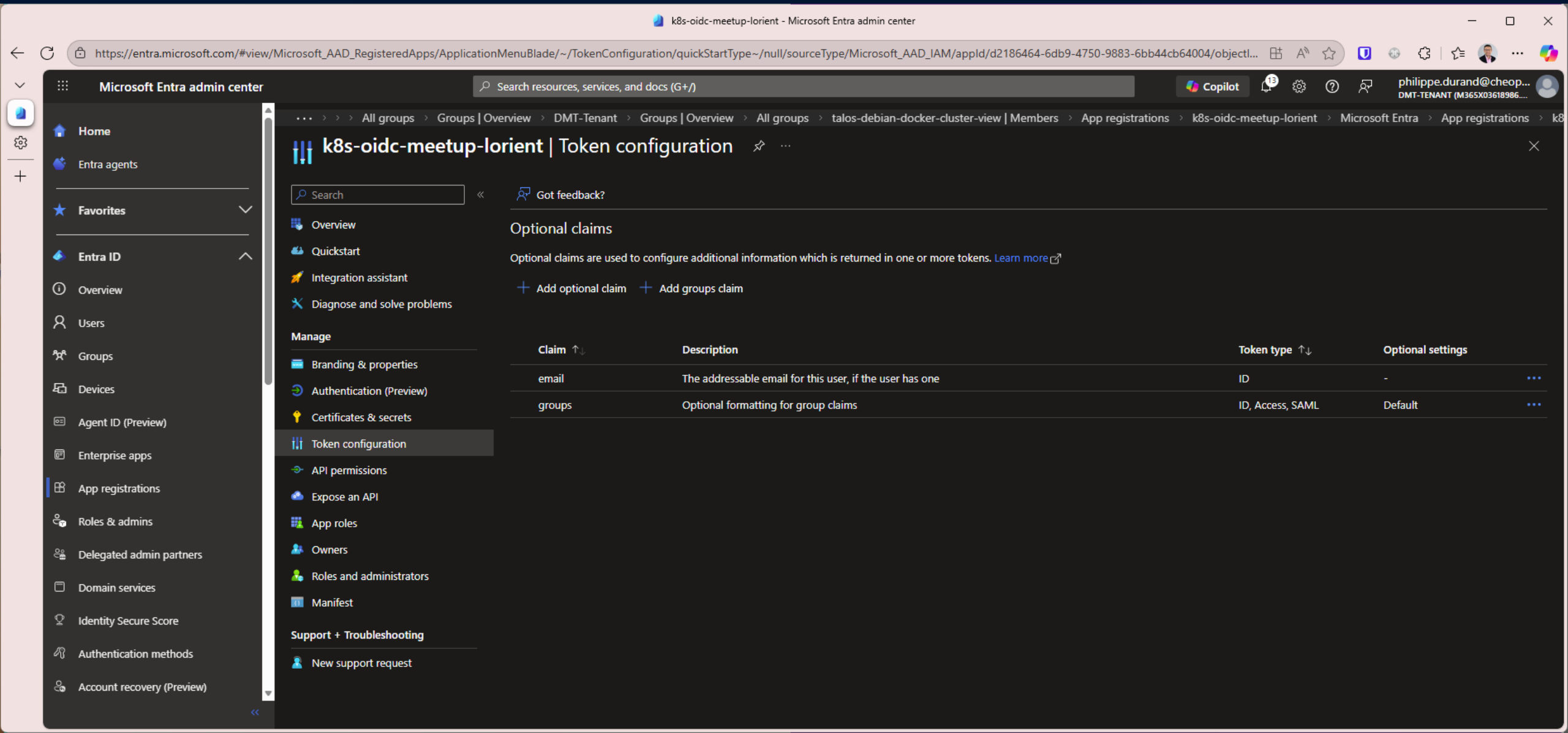
K8S OIDC avec Microsoft Entra

Paramétrage Entra – Paramétrage du token



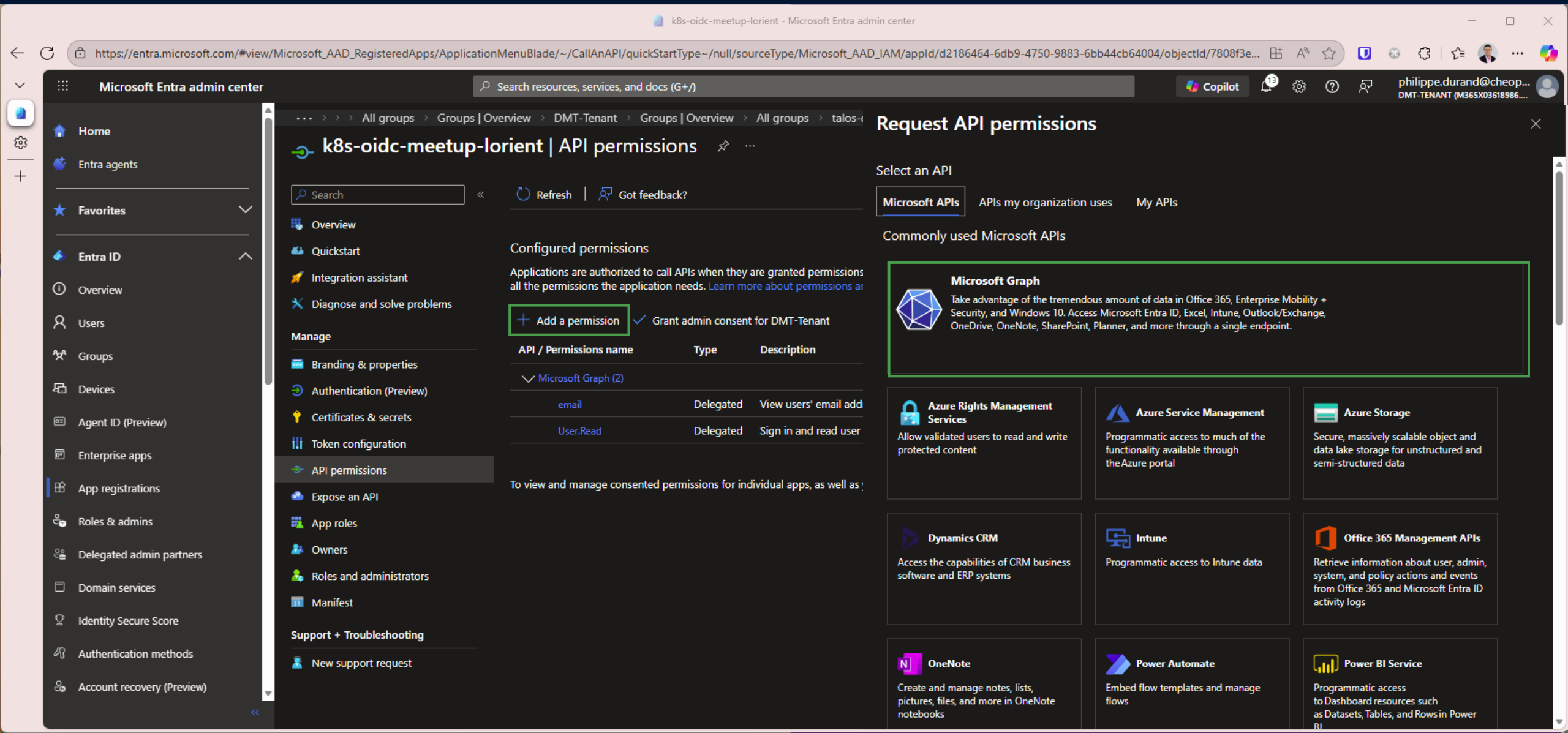
K8S OIDC avec Microsoft Entra

Paramétrage Entra – Paramétrage du token



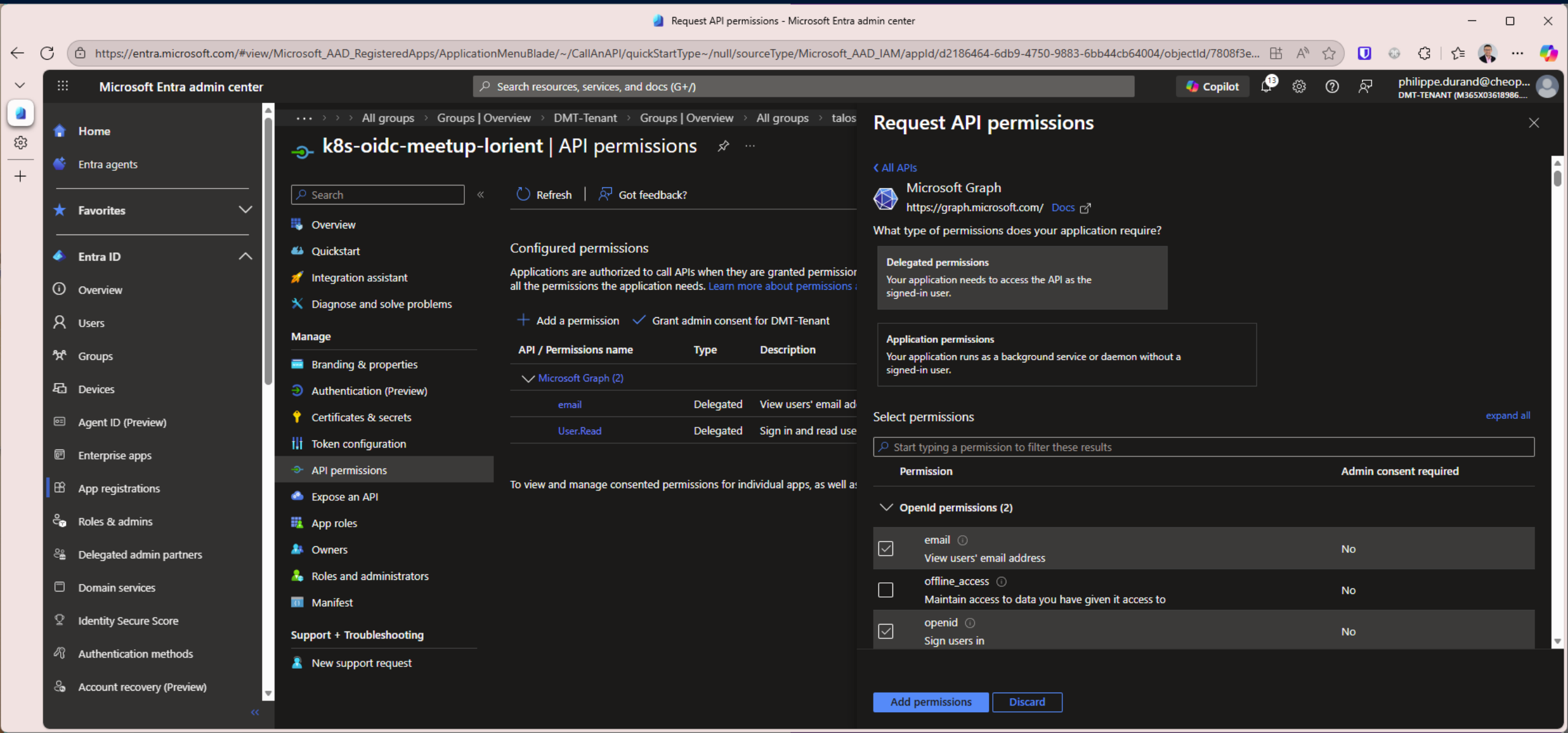
K8S OIDC avec Microsoft Entra

Paramétrage Entra – Paramétrage de l'api



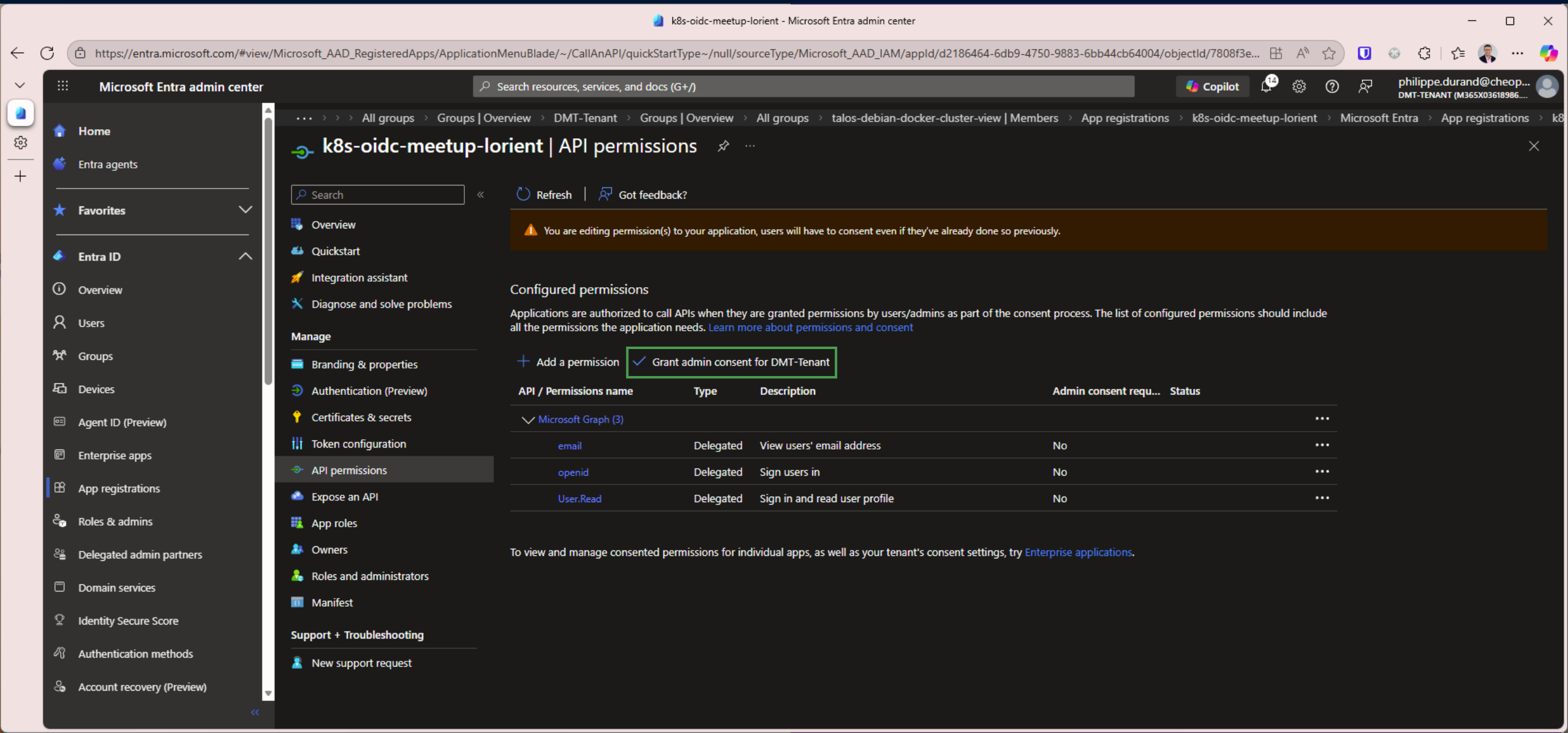
K8S OIDC avec Microsoft Entra

Paramétrage Entra – Paramétrage de l'api



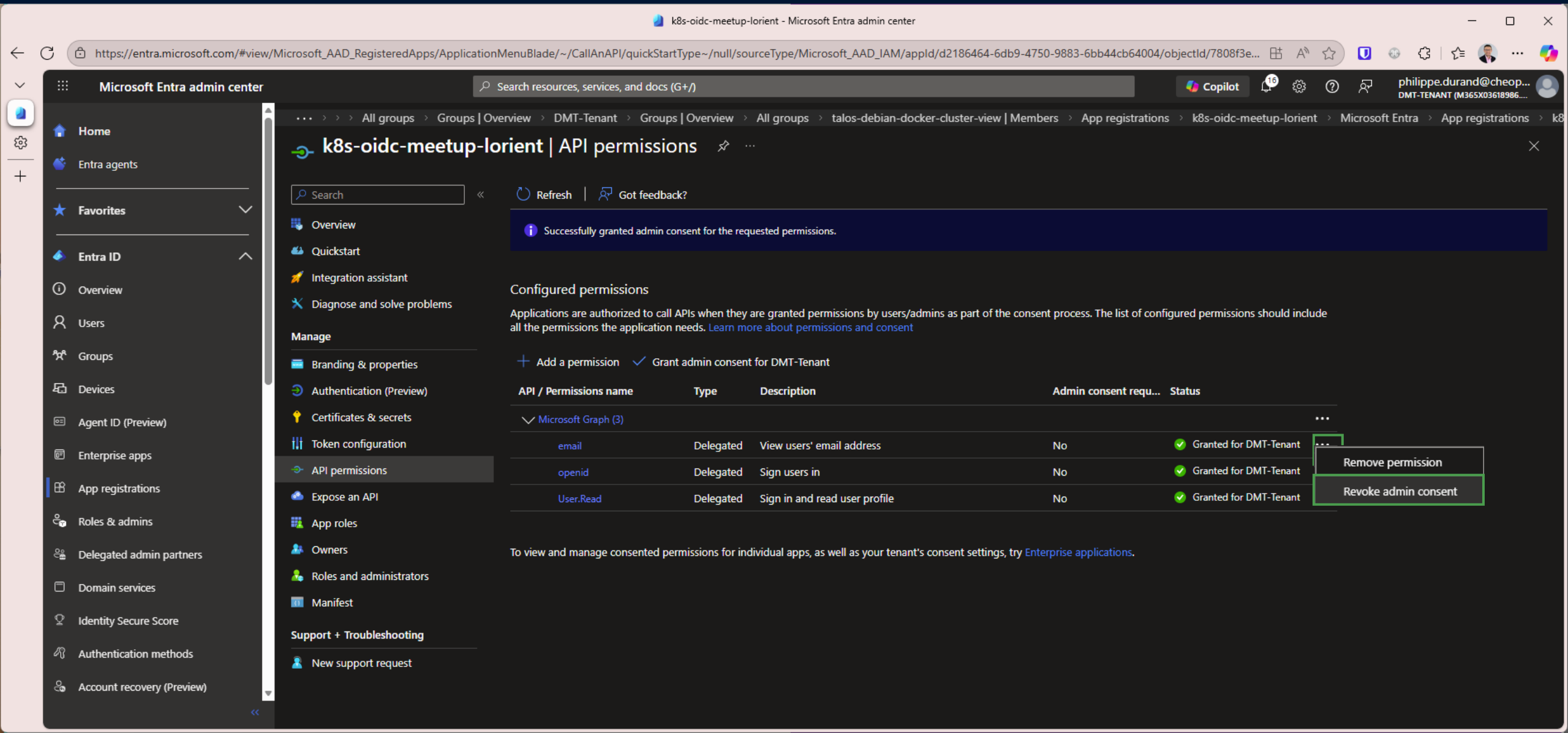
K8S OIDC avec Microsoft Entra

Paramétrage Entra – Paramétrage de l'api



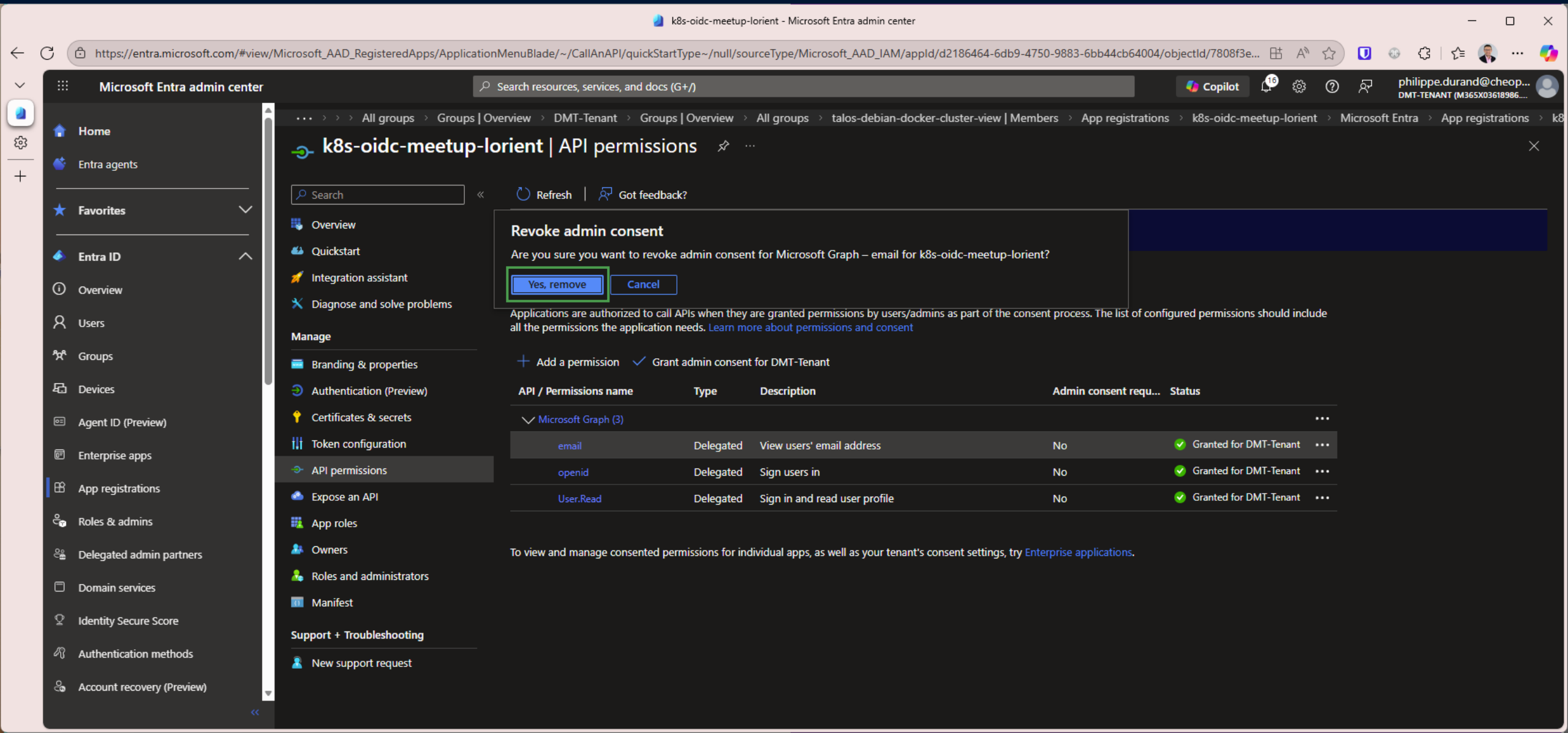
K8S OIDC avec Microsoft Entra

Paramétrage Entra – Paramétrage de l'api



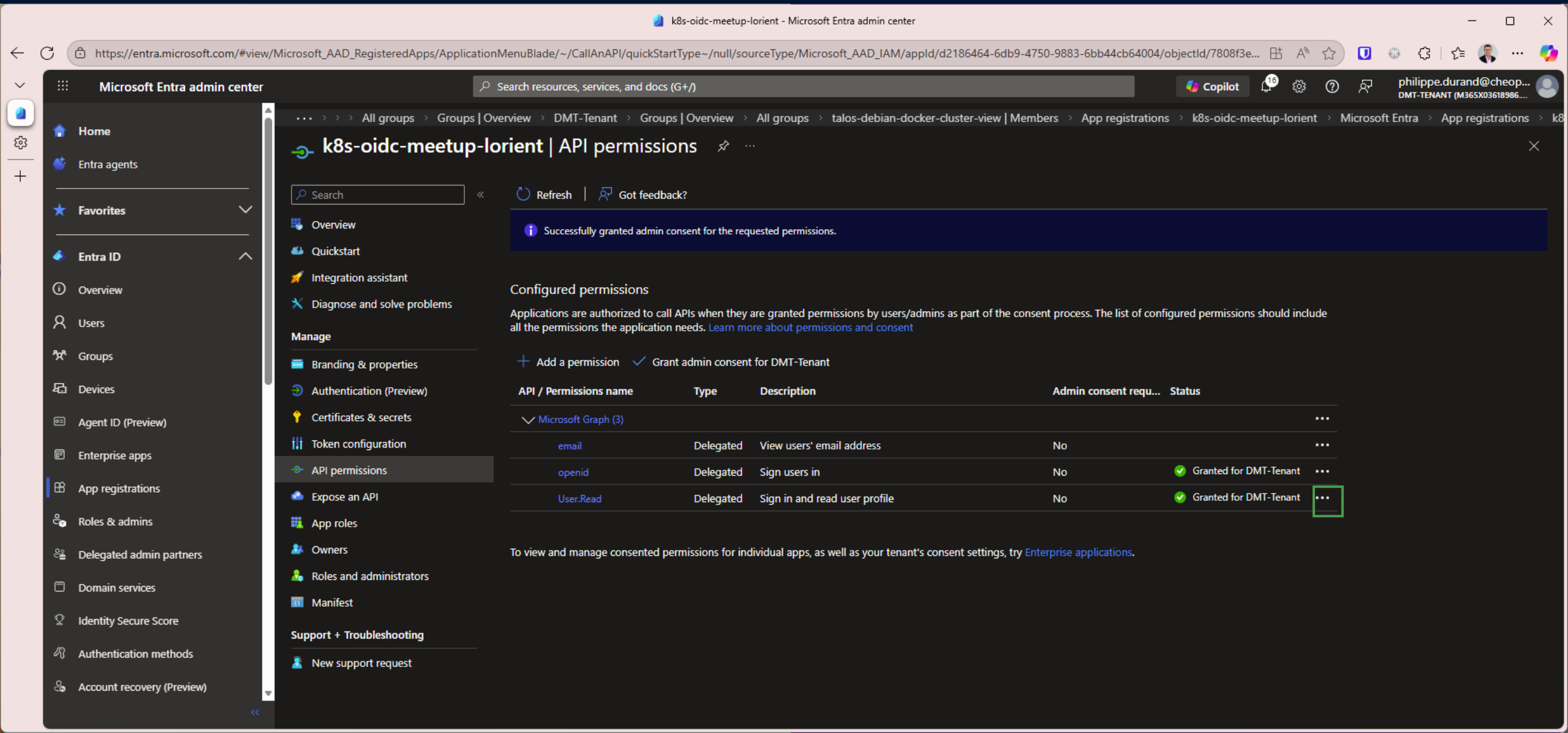
K8S OIDC avec Microsoft Entra

Paramétrage Entra – Paramétrage de l'api



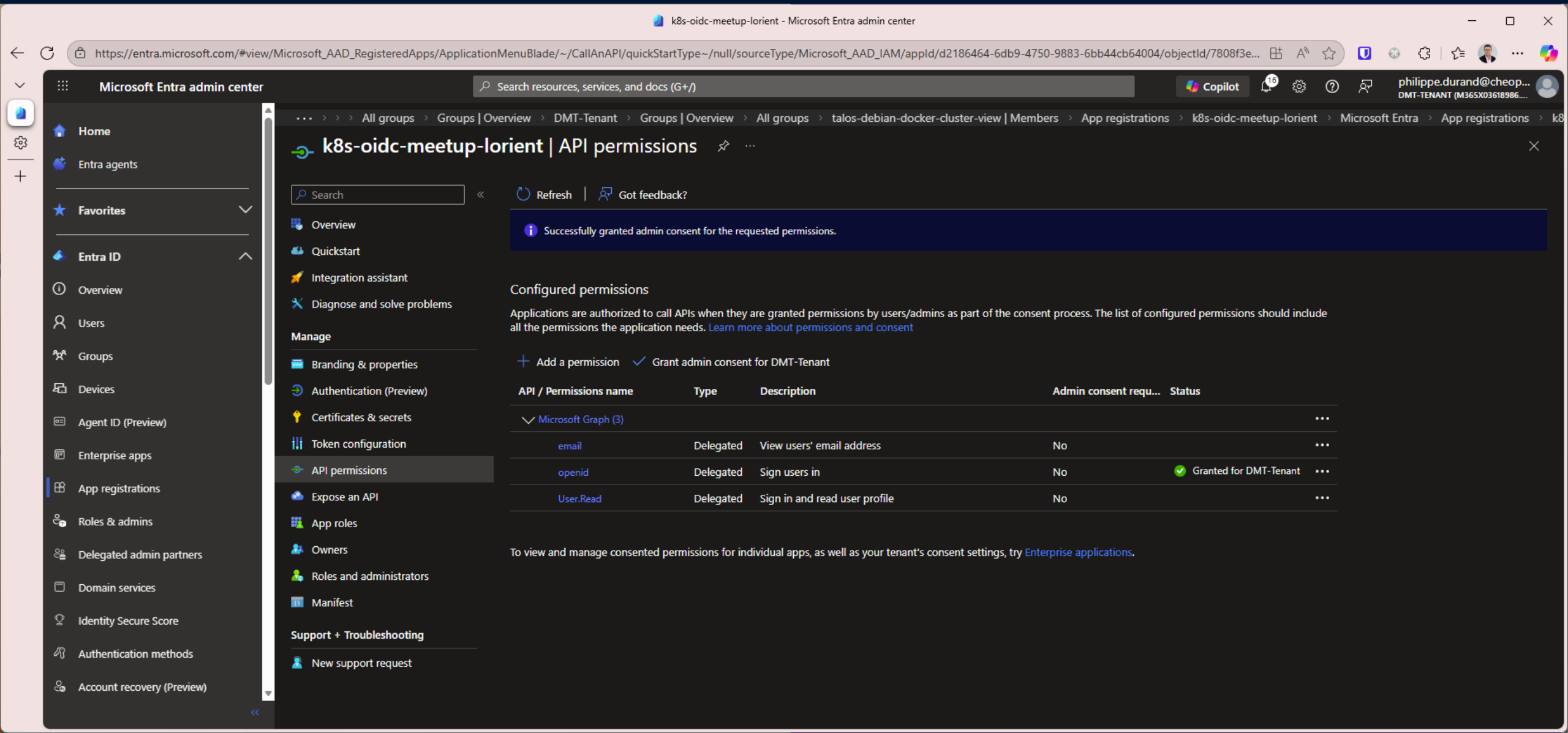
K8S OIDC avec Microsoft Entra

Paramétrage Entra – Paramétrage de l'api



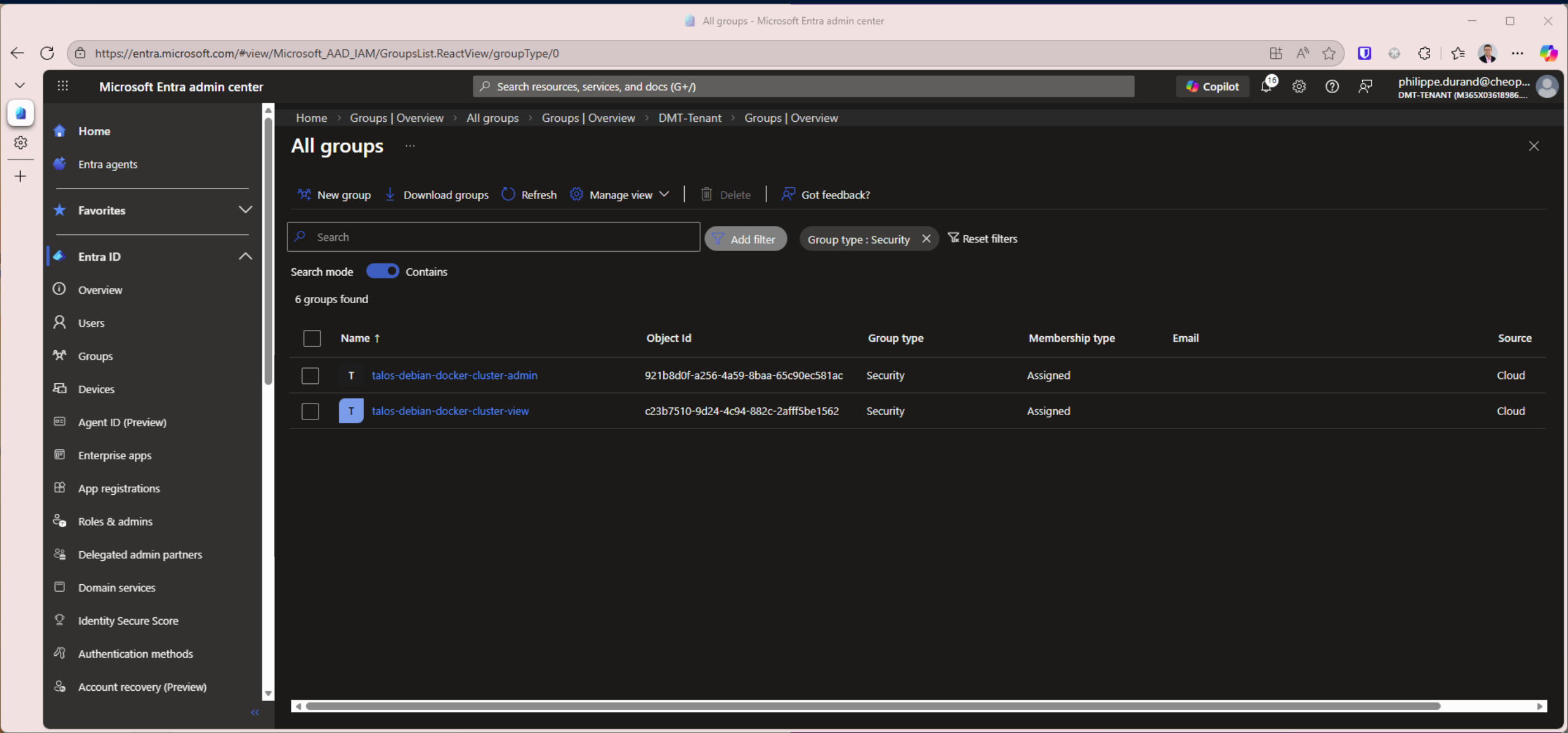
K8S OIDC avec Microsoft Entra

Paramétrage Entra – Paramétrage de l'api



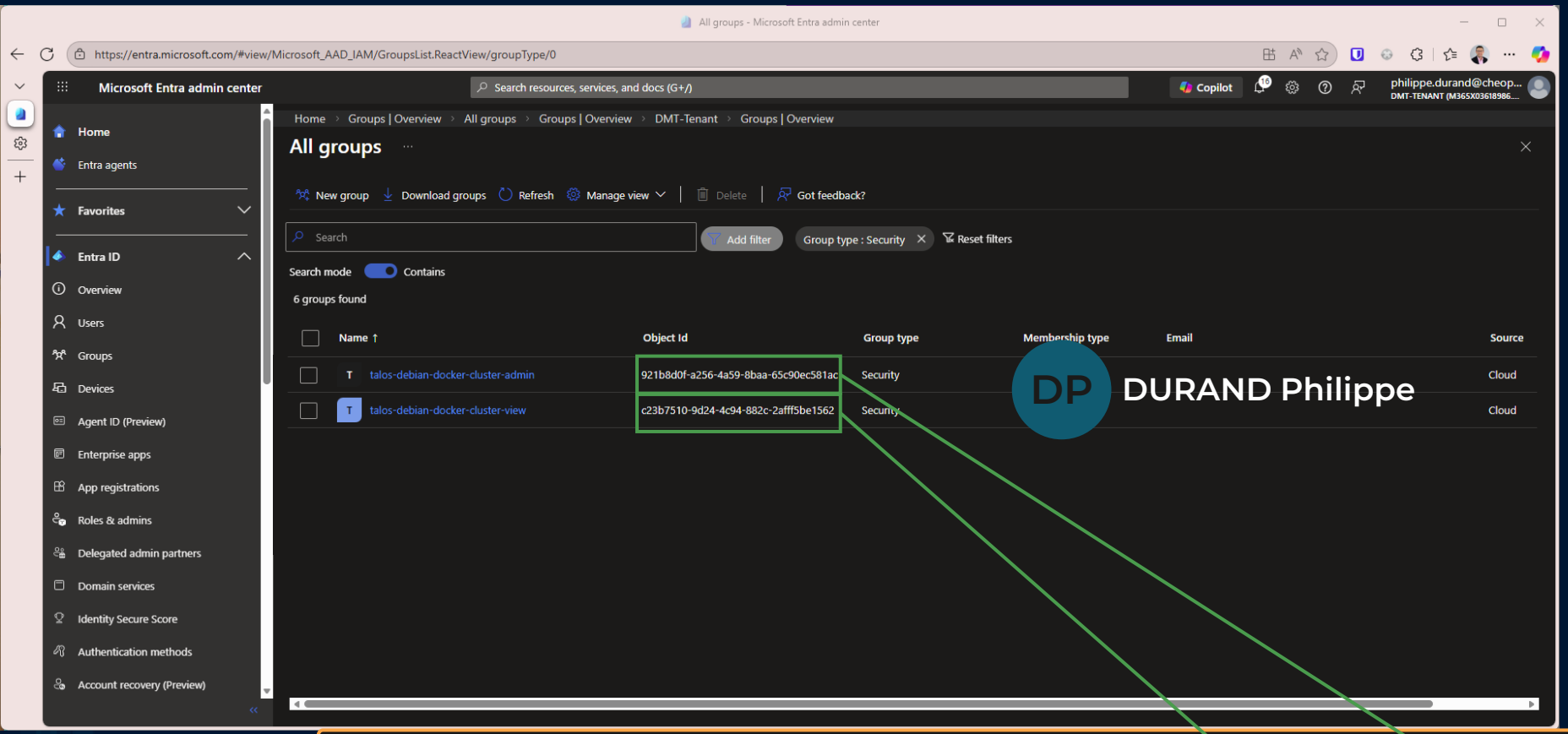
K8S OIDC avec Microsoft Entra

Paramétrage Entra – Groupes et affectations groupes ←→ users



K8S OIDC avec Microsoft Entra

Paramétrage Entra – Groupes et RBAC K8S



Admin global

```
Terminal
$  k create clusterrolebinding entra-admin --clusterrole cluster-admin --group entra:921b8d0f-a256-4a59-8baa-65c90ec581ac
clusterrolebinding.rbac.authorization.k8s.io/entra-admin created
$  k create clusterrolebinding entra-view --clusterrole view --group entra:c23b7510-9d24-4c94-882c-2afff5be1562
clusterrolebinding.rbac.authorization.k8s.io/entra-view created
```

K8S OIDC avec Microsoft Entra

Le paramétrage ... avance.

CHEOPS TECHNOLOGY
The Cloud Customized For You !

K8S OIDC avec Microsoft Entra

Paramétrage K8S

Talos-oidc-patch.yaml

```yaml
cluster:
  apiServer:
    extraArgs:
      oidc-issuer-url: https://login.microsoftonline.com/de5169f1-9a13-45fe-a4cf-07feee258b7f/v2.0
      oidc-client-id: d2186464-6db9-4750-9883-6bb44cb64004
      oidc-username-claim: email
      oidc-username-prefix: "entra:"
      oidc-groups-claim: groups
      oidc-groups-prefix: "entra:"
```

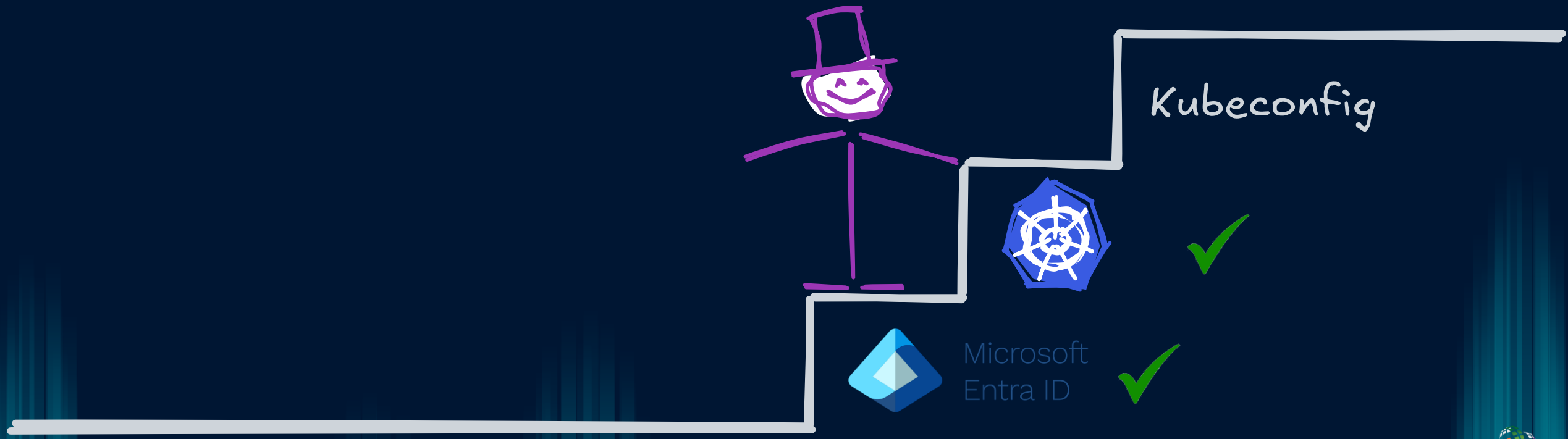
Terminal

```
$  talosctl edit machineconfig --nodes 10.5.0.2
Applied configuration without a reboot
$  docker restart talos-debian-docker-controlplane-1
talos-debian-docker-controlplane-1
```


CHEOPS TECHNOLOGY
The Cloud Customized For You !

K8S OIDC avec Microsoft Entra

Le paramétrage ... on y est presque!



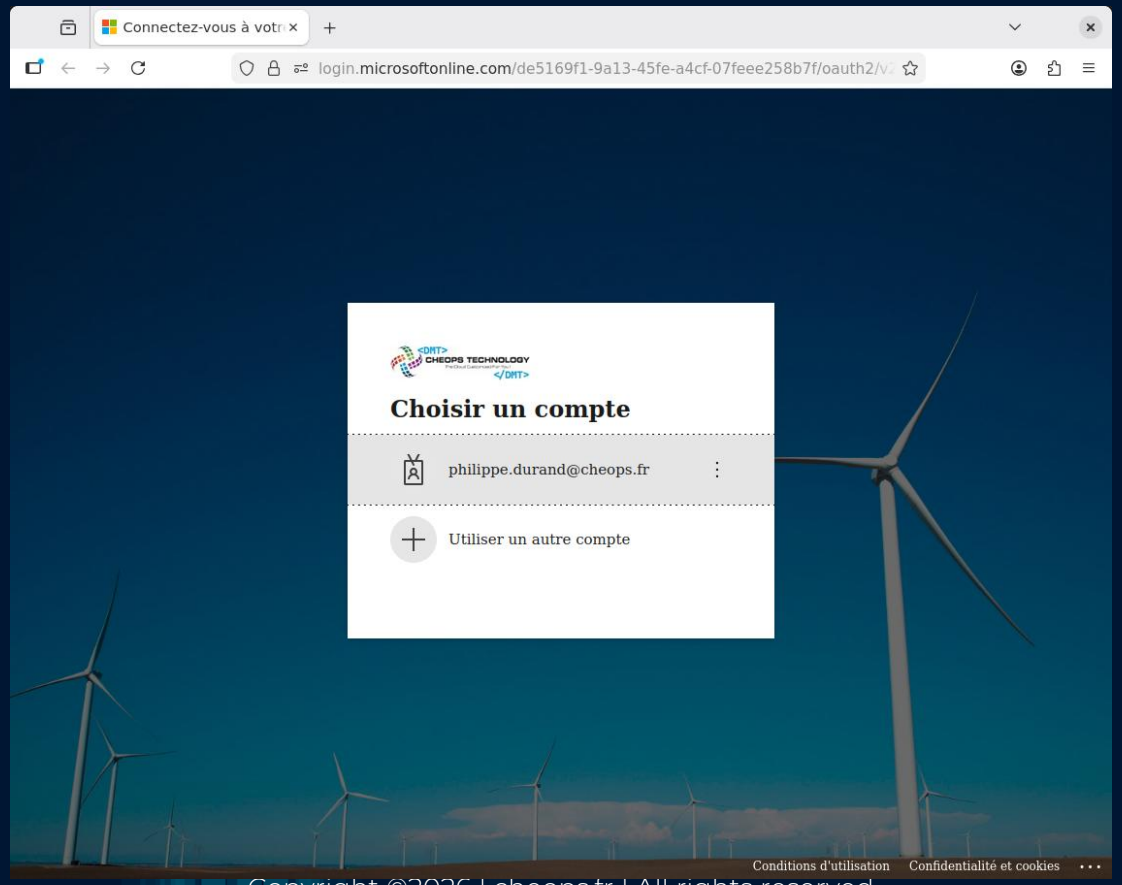
CHEOPS TECHNOLOGY
The Cloud Customized For You !

K8S OIDC avec Microsoft Entra

Paramétrage kubectl

Terminal

```
$ brew install int128/kubelogin/kubelogin
$ kubelogin get-token --oidc-issuer-url https://login.microsoftonline.com/de5169f1-9a13-45fe-a4cf-07feee258b7f/v2.0 --oidc-client-id d2186464-6db9-4750-9883-6bb44cb64004 --oidc-client-secret X6j8Q~7pc2~6ONWyHT15gdPa8j~rvt26NGGKhacB
```

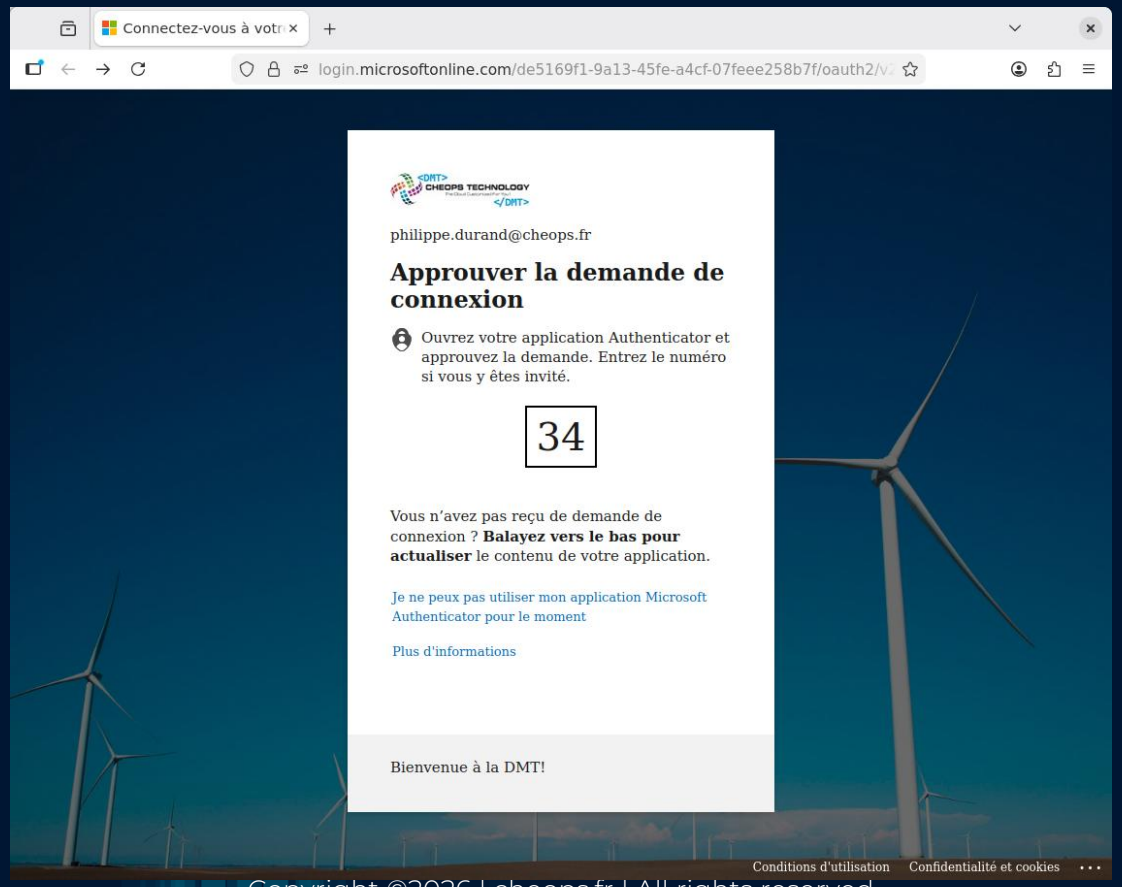


K8S OIDC avec Microsoft Entra

Paramétrage kubectl

Terminal

```
$ brew install int128/kubelogin/kubelogin
$ kubelogin get-token --oidc-issuer-url https://login.microsoftonline.com/de5169f1-9a13-45fe-a4cf-07feee258b7f/v2.0 --oidc-client-id d2186464-6db9-4750-9883-6bb44cb64004 --oidc-client-secret X6j8Q~7pc2~6ONWyHT15gdPa8j~rvt26NGGKhacB
```

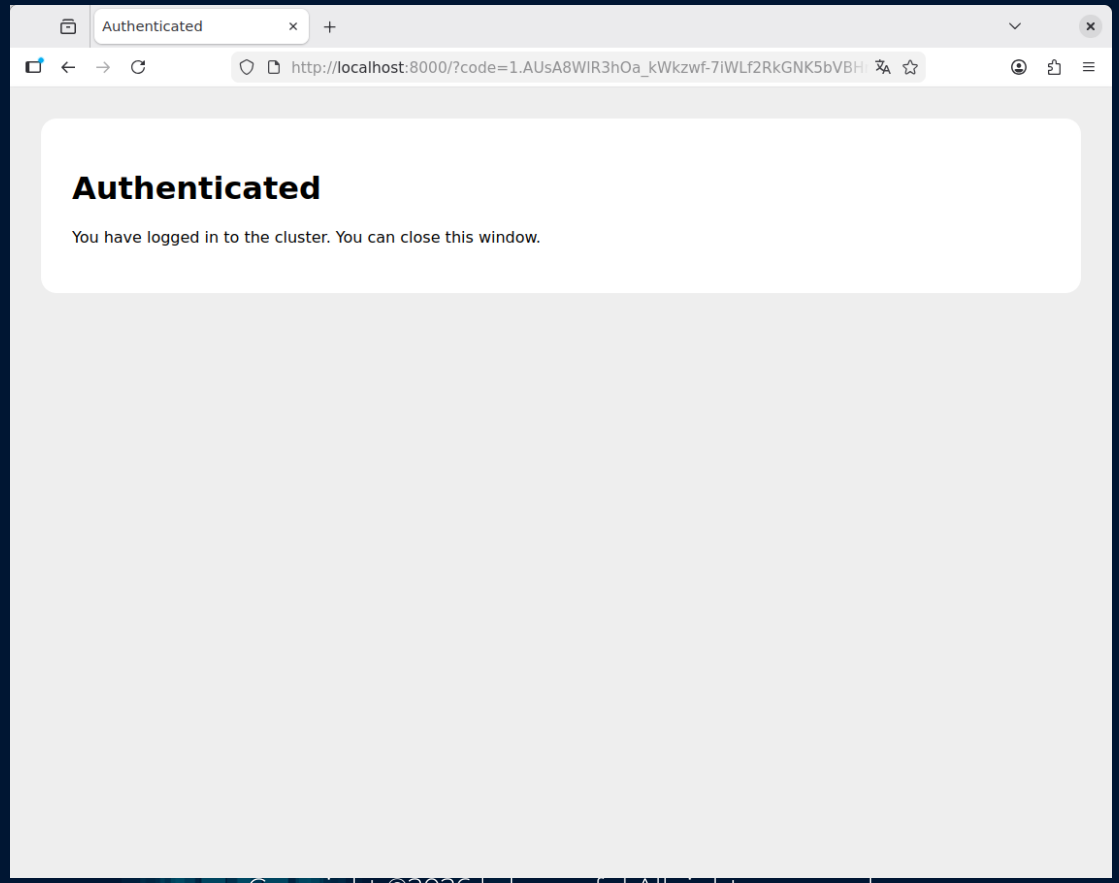


K8S OIDC avec Microsoft Entra

Paramétrage kubectl

Terminal

```
$ brew install int128/kubelogin/kubelogin
$ kubelogin get-token --oidc-issuer-url https://login.microsoftonline.com/de5169f1-9a13-45fe-a4cf-07feee258b7f/v2.0 --oidc-client-id d2186464-6db9-4750-9883-6bb44cb64004 --oidc-client-secret X6j8Q~7pc2~6ONWyHT15gdPa8j~rvt26NGGKhacB
```



CHEOPS TECHNOLOGY
The Cloud Customized For You !

K8S OIDC avec Microsoft Entra

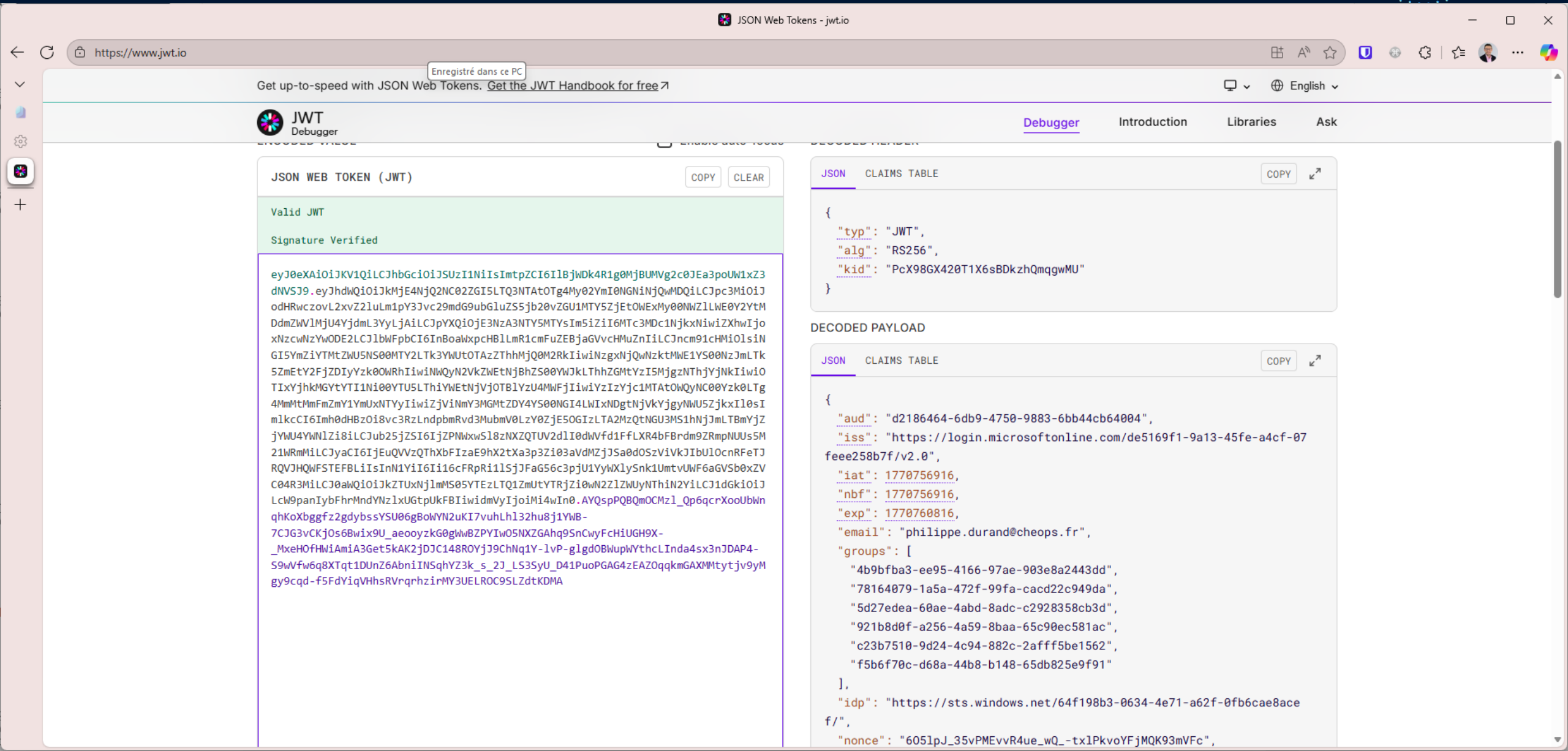
Paramétrage kubectl

Terminal

```
$  brew install int128/kubelogin/kubelogin

$  kubelogin get-token --oidc-issuer-url https://login.microsoftonline.com/de5169f1-9a13-45fe-a4cf-07feee258b7f/v2.0 --oidc-
   client-id d2186464-6db9-4750-9883-6bb44cb64004 --oidc-client-secret X6j8Q~7pc2~6ONWyHT15gdPa8j~rvt26NGGKhacB

{"kind":"ExecCredential","apiVersion":"client.authentication.k8s.io/v1beta1","spec":{"interactive":false},"status":{"expirationTi
mestamp":"2026-02-
10T20:23:13Z","token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IlBjWDk4R1g0MjBUMVg2c0JEa3poUW1xZ3dNVSJ9.eyJhdWQiOiJkMjE4NjQ2N
C02ZGI5LTQ3NTAtOTg4My02YmI0NGNiNjQwMDQiLCJpc3MiOiJodHRwczovL2xvZ2luLm1pY3Jvc29mdG9ubGluZS5jb20vZGU1MTY5ZjEtOWExMy00NWZlLWE0Y2YtMD
dmZWVlMjU4YjdmL3YyLjAiLCJpYXQiOjE3NzA3NTEwOTMsIm5iZiI6MTc3MDc1MTA5MywiZXhwIjoxNzcwNzU0OTkzLCJlbWFpbCI6InBoaWxpcHBlLmR1cmFuZEBjaGV
vcHMuZnIiLCJncm91cHMiOlsiNGI5YmZiYTMtZWU5NS00MTY2LTk3YWUtOTAzZThhMjQ0M2RkIiwiZjViNmY3MGMtZDY4YS00NGI4LWIxNDgtNjVkYjgyNWU5ZjkxIiwi
OTIxYjhkMGYtYTI1Ni00YTU5LThiYWEtNjVjOTBlYzU4MWFjIiwiYzIzYjc1MTAtOWQyNC00Yzk0LTg4MmMtMmFmZmY1YmUxNTYyIiwiNzgxNjQwNzktMWE1YS00NzJmL
Tk5ZmEtY2FjZDIyYzk0OWRhIiwiNWQyN2VkZWEtNjBhZS00YWJkLThhZGMtYzI5MjgzNThjYjNkIl0sImlkcCI6Imh0dHBzOi8vc3RzLndpbmRvd3MubmV0LzY0ZjE5OG
IzLTA2MzQtNGU3MS1hNjJmLTBmYjZjYWU4YWNlZi8iLCJub25jZSI6IlJjZmFYbVk4SENYbEFkNk1LZ1dfTkx0WnF2NkE1UXM3MjlnRmVSR3VUb0UiLCJyaCI6IjEuQVV
zQThXbFIzaE9hX2tXa3p3Zi03aVdMZjJSa0dOSzViVkJIbUlOcnRFeTJRQVJHQWFSTEFBLiIsInN1YiI6Ii16cFRpRi1lSjJFaG56c3pjU1YyWXlySnk1UmtvUWF6aGVS
b0xZVC04R3MiLCJ0aWQiOiJkZTUxNjlmMS05YTEzLTQ1ZmUtYTRjZi0wN2ZlZWUyNThiN2YiLCJ1dGkiOiJXM2NpU0p0VGVVNllCdEZFZkFFVUFBIiwidmVyIjoiMi4wI
n0.F-Jy5InwUNTx2UgZQdwPihDGnc6V8gvzIRo-4NjVbDkJZLxpfUGAGlTquVnlaME4S5sr2jQTPlqTId-
QwYJC4qxcvrxYrv7shzjFWgX3lTe6mv3l6tzOkcmz79W1YyWIIVretqUX2uRdOoewwReCZecgaJnVv-BBYhCjxkXV8s-
PLlWI7SGvjLgQDd_UdfjWVAZcgnKAyFR8NR5rZtX0JialZguxeEOtzj45dxZj4oRWFc1IZLHc7Fq-4Yitq4WhfYEDYK-
IofQ14lR2nDVM3Y29Qw9XELgNWAXcx8CymlJCSJ2DbEftdqVzEJ5K4K3BmnH2F5T2h1cg4s5EdGOf9w"}}

$  export TOKEN=$(kubelogin get-token \
   --oidc-issuer-url https://login.microsoftonline.com/de5169f1-9a13-45fe-a4cf-07feee258b7f/v2.0 \
   --oidc-client-id d2186464-6db9-4750-9883-6bb44cb64004 \
   --oidc-client-secret X6j8Q~7pc2~6ONWyHT15gdPa8j~rvt26NGGKhacB \
   | jq -r '.status.token')

$  KUBECONFIG=/dev/null k --server=https://10.5.0.2:6443 --insecure-skip-tls-verify --token="$TOKEN" auth whoami

ATTRIBUTE   VALUE
Username    entra:philippe.durand@cheops.fr
Groups      [entra:4b9bfba3-ee95-4166-97ae-903e8a2443dd entra:78164079-1a5a-472f-99fa-cacd22c949da entra:5d27edea-60ae-4abd-8adc-
c2928358cb3d entra:921b8d0f-a256-4a59-8baa-65c90ec581ac entra:c23b7510-9d24-4c94-882c-2afff5be1562 entra:f5b6f70c-d68a-44b8-b148-
65db825e9f91 system:authenticated]

$  KUBECONFIG=/dev/null k --server=https://10.5.0.2:6443 --insecure-skip-tls-verify --token="$TOKEN" auth can-i delete ns
yes
```

K8S OIDC avec Microsoft Entra

Paramétrage kubectl



K8S OIDC avec Microsoft Entra

Paramétrage kubectl

~/.kube/config

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLSW8xQXhFblU3RmkrcXhpUzlQUG5UMGpFWEg2M3VQBDRVJUSUZJQ0FURS0tLS0tCg==
    server: https://10.5.0.2:6443
  name: talos-debian-docker
contexts:
...
- context:
    cluster: talos-debian-docker
    namespace: default
    user: philippe-durand@talos-debian-docker
  name: philippe-durand@talos-debian-docker
...
current-context: admin@talos-debian-docker
kind: Config
users:
...
- name: philippe-durand@talos-debian-docker
  user:
    client-certificate-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCkVJFdFhySC9zSQ0VSVElGSUNBVEUtLS0tLQo=
    client-key-data: LS0tLS1CRUdJTiBQNnBQV1p6NUJ4VLMzJ4dldKTEIrajNSCi0tLS0tRU5EIFBSSVZBVEUgS0VZLS0tLS0K
- name: entra@talos-debian-docker
  user:
    token: eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IlBjWDk4R1g0MjBUMVg2c0JEa3poUW1xZ3dNVSJ9.eyJhdWQ
```


CHEOPS TECHNOLOGY
The Cloud Customized For You !

K8S OIDC avec Microsoft Entra

Paramétrage kubectl

Terminal

```
$  k --context entra@talos-debian-docker auth can-i delete ns
yes
$  k config set-credentials oidc \
  --exec-api-version=client.authentication.k8s.io/v1 \
  --exec-interactive-mode=Never \
  --exec-command=kubectl \
  --exec-arg=oidc-login \
  --exec-arg=get-token \
  --exec-arg="--oidc-issuer-url=https://login.microsoftonline.com/de5169f1-9a13-45fe-a4cf-07feee258b7f/v2.0" \
  --exec-arg="--oidc-client-id=d2186464-6db9-4750-9883-6bb44cb64004" \
  --exec-arg="--oidc-client-secret=X6j8Q~7pc2~6ONWyHT15gdPa8j~rvt26NGGKhacB“
User "oidc" set.
```

~/.kube/config

```
...
current-context: admin@talos-debian-docker
kind: Config
users:
...
- name: oidc
  user:
    exec:
      apiVersion: client.authentication.k8s.io/v1
      args:
      - oidc-login
      - get-token
      - --oidc-issuer-url=https://login.microsoftonline.com/de5169f1-9a13-45fe-a4cf-07feee258b7f/v2.0
      - --oidc-client-id=d2186464-6db9-4750-9883-6bb44cb64004
      - --oidc-client-secret=X6j8Q~7pc2~6ONWyHT15gdPa8j~rvt26NGGKhacB
      command: kubectl
      env: null
      interactiveMode: Never
      provideClusterInfo: false
```

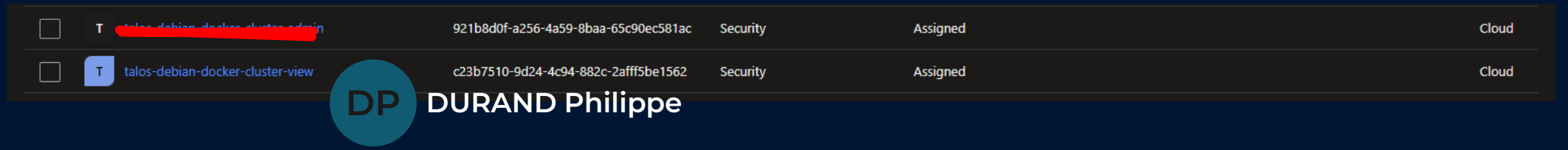
CHEOPS TECHNOLOGY
The Cloud Customized For You !

K8S OIDC avec Microsoft Entra

Paramétrage kubectl

Admin global

```
Terminal
$  k --context oidc auth can-i delete ns
yes
$  k --context oidc auth whoami
ATTRIBUTE   VALUE
Username    entra:philippe.durand@cheops.fr
Groups      [entra:4b9bfba3-ee95-4166-97ae-903e8a2443dd entra:78164079-1a5a-472f-99fa-cacd22c949da entra:5d27edea-60ae-4abd-8adc-c2928358cb3d entra:921b8d0f-a256-4a59-8baa-65c90ec581ac entra:c23b7510-9d24-4c94-882c-2afff5be1562 entra:f5b6f70c-d68a-44b8-b148-65db825e9f91 system:authenticated]
```

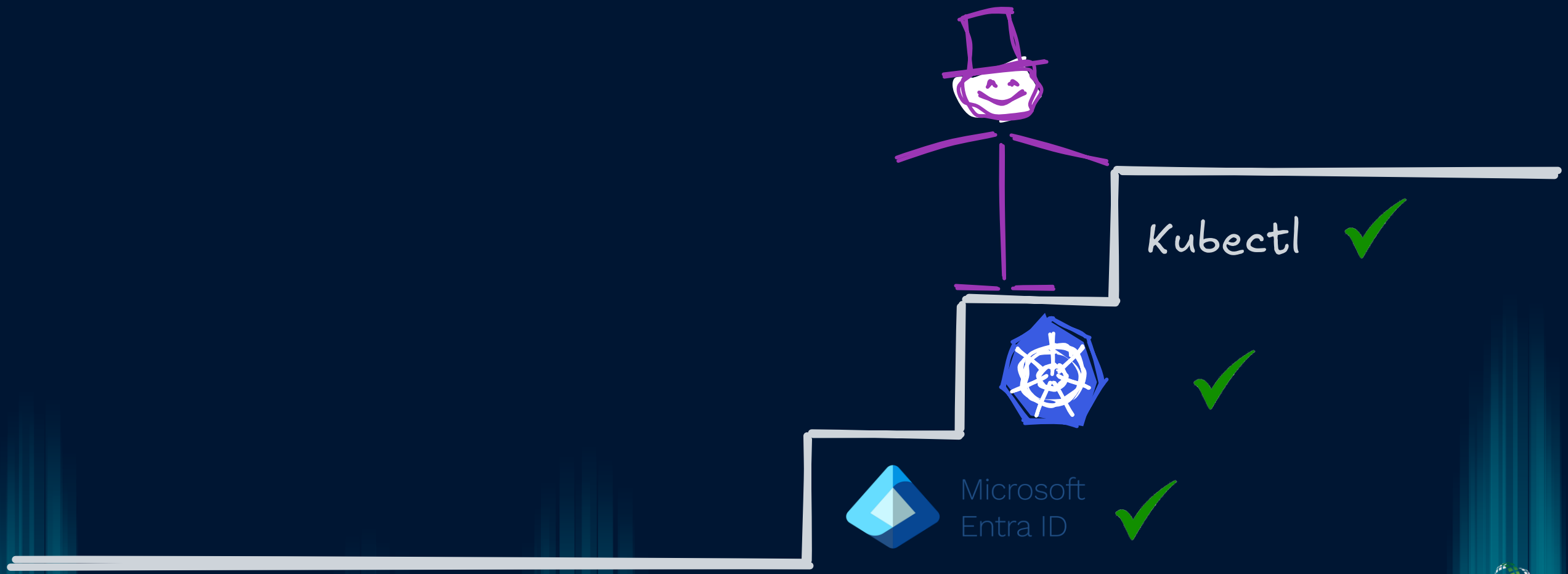



```
Terminal
$  rm ~/.kube/cache/oidc-login/*
$  k --context oidc auth can-i get pods
yes
$  k --context oidc auth can-i delete ns
no
```


CHEOPS TECHNOLOGY
The Cloud Customized For You !

K8S OIDC avec Microsoft Entra

Le paramétrage ... est fini!



CHEOPS TECHNOLOGY
The Cloud Customized For You !

Authentification

Résumé

	Certificats	Tokens de SA	OIDC
Acteur nominatif	100%	0%	100%
Gestion des groupes	50%	80%	100%
Révocation possible	0%	100%	100%
Simplicité de gestion	20%	50%	90%
Passe les reverse proxy L7 APIServer	0%	100%	100%

Authenfication

Alors ... convaincu?

Questions
Réponses

philippe-durand-work