Práctica 4.

Tarea 1.

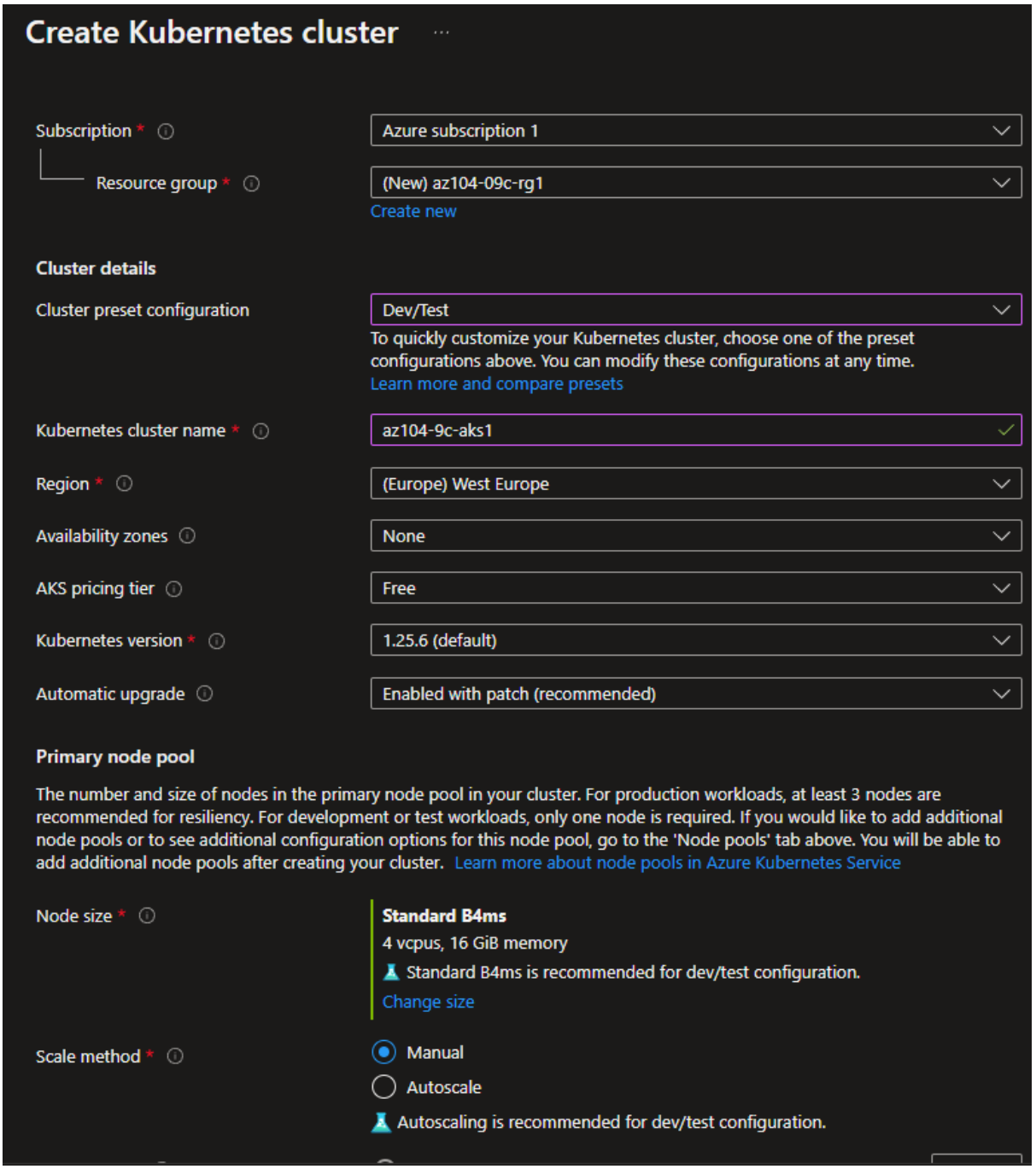
Abrimos el cloud powershell de azure e introducimos los comandos:

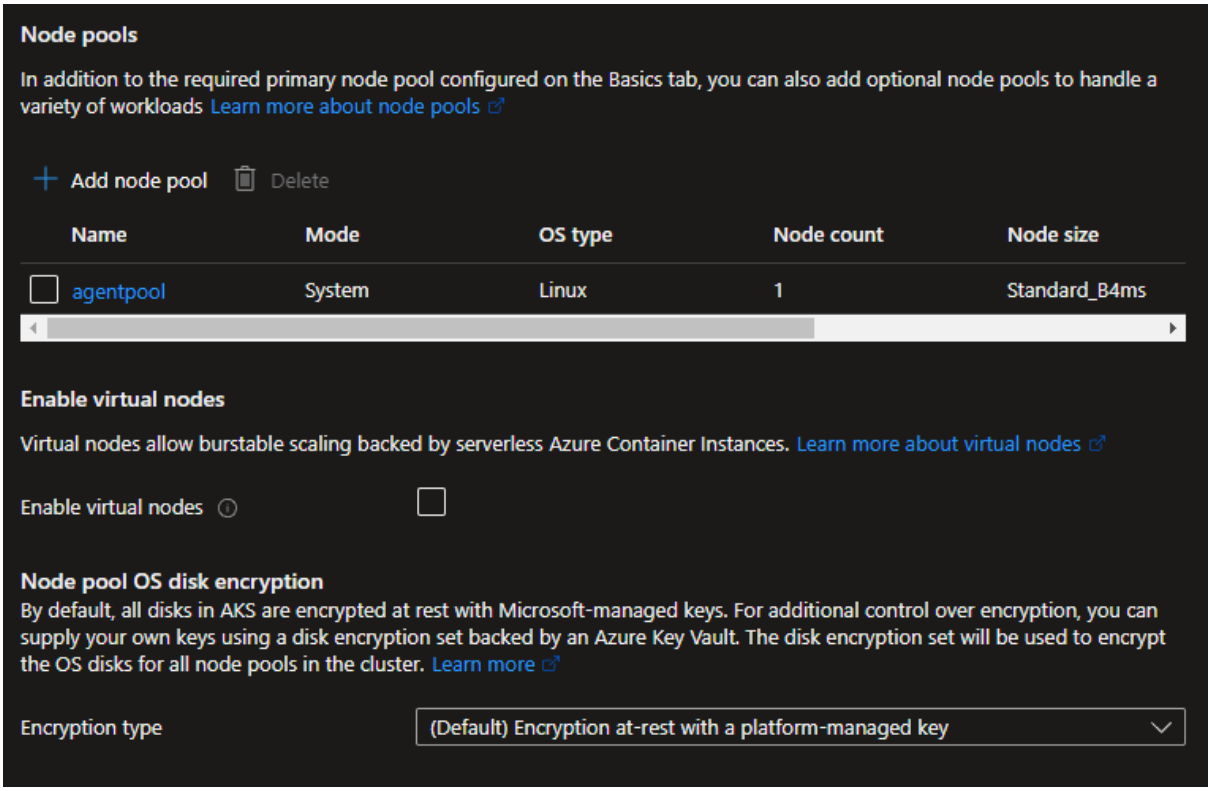
```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell


MOTD: Azure Cloud Shell now includes Predictive IntelliSense! Learn more: https://aka.ms/CloudShell/IntelliSense

VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
PS /home/adrian> Register-AzResourceProvider -ProviderNamespace Microsoft.Kubernetes
```

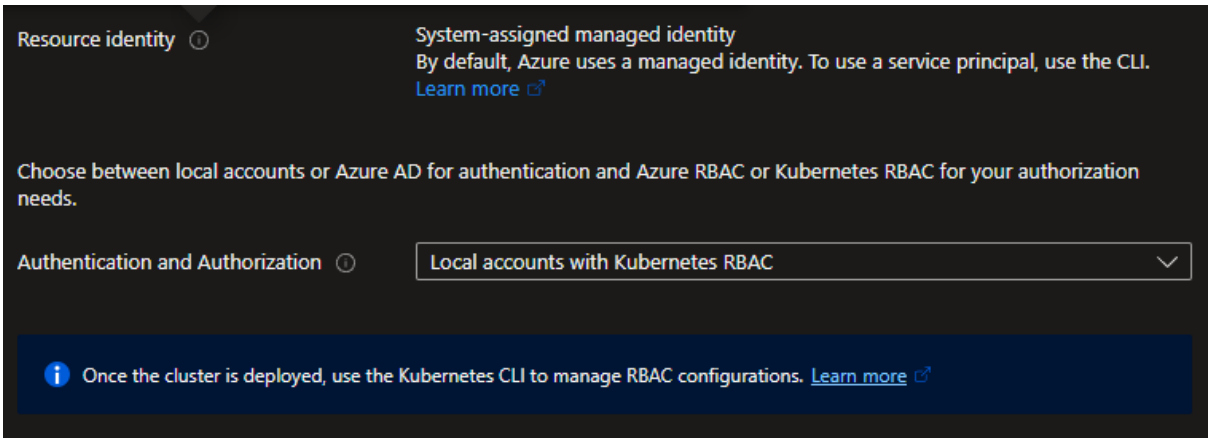
```
PS /home/adrian> Register-AzResourceProvider -ProviderNamespace Microsoft.KubernetesConfiguration

ProviderNamespace : Microsoft.KubernetesConfiguration
RegistrationState : Registering
ResourceTypes     : {sourceControlConfigurations, extensions, fluxConfigurations, operations…}
Locations         : {East US, West Europe, West Central US, West US 2…}
```

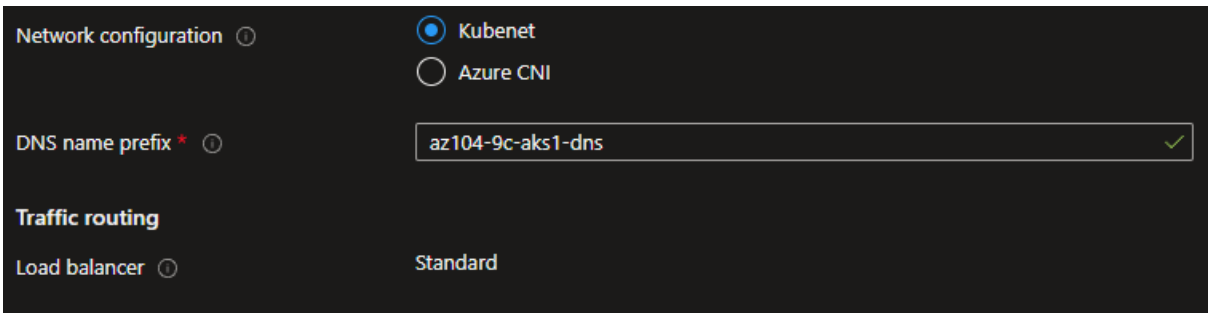
Tarea 2.

Creacion del cluster de kubernetes

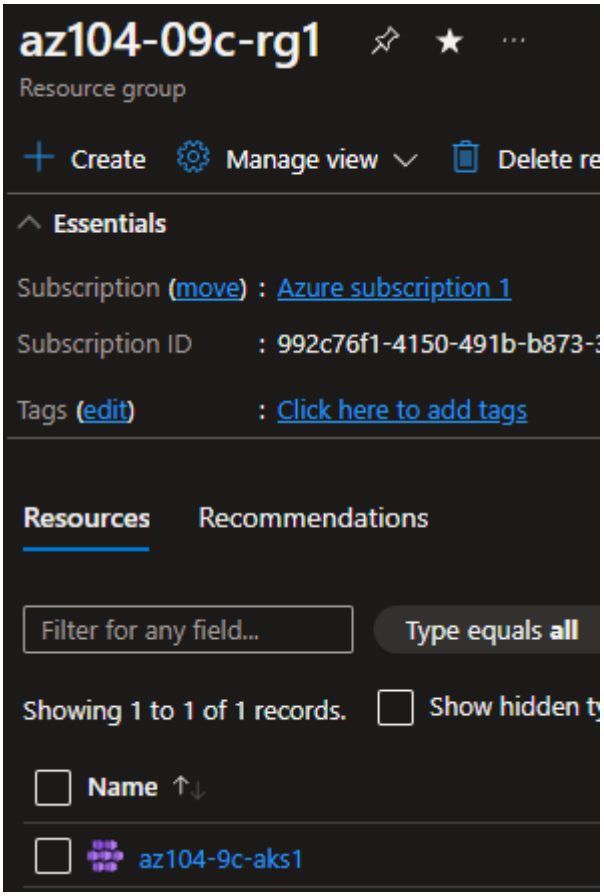



Node pools
In addition to the required primary node pool configured on the Basics tab, you can also add optional node pools to handle a variety of workloads Learn more about node pools
Add node pool
Delete
Name
Mode
OS type
Node count
Node size
agentpool
System
Linux
1
Standard_B4ms
Enable virtual nodes
Virtual nodes allow burstable scaling backed by serverless Azure Container Instances. Learn more about virtual nodes
Enable virtual nodes
Node pool OS disk encryption
By default, all disks in AKS are encrypted at rest with Microsoft-managed keys. For additional control over encryption, you can supply your own keys using a disk encryption set backed by an Azure Key Vault. The disk encryption set will be used to encrypt the OS disks for all node pools in the cluster. Learn more
Encryption type
(Default) Encryption at-rest with a platform-managed key


Resource identity
System-assigned managed identity
By default, Azure uses a managed identity. To use a service principal, use the CLI.
Learn more
Choose between local accounts or Azure AD for authentication and Azure RBAC or Kubernetes RBAC for your authorization needs.
Authentication and Authorization
Local accounts with Kubernetes RBAC
Once the cluster is deployed, use the Kubernetes CLI to manage RBAC configurations. Learn more

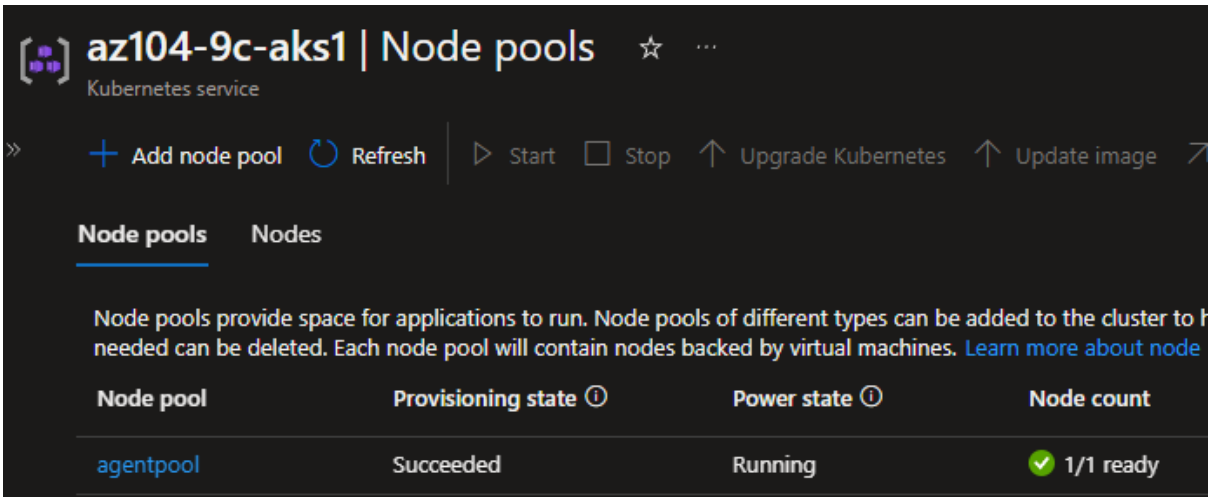

Network configuration
Kubenet
Azure CNI
DNS name prefix *
az104-9c-aks1-dns
Traffic routing
Load balancer
Standard

Ya se ha creado el resource group con el cluster dentro.



Tarea 3.

Abrimos el grupo de nodos:



y la consola en azure cloud console en bash y verificamos la conexión.

```
adrian [ ~ ]$ RESOURCE_GROUP='az104-09c-rg1'
AKS_CLUSTER='az104-9c-aks1'
az aks get-credentials --resource-group $RESOURCE_GROUP --name $AKS_CLUSTER
kubectl get nodes
Merged "az104-9c-aks1" as current context in /home/adrian/.kube/config
NAME                                STATUS   ROLES   AGE   VERSION
aks-agentpool-48599477-vmss000000   Ready    agent   22m   v1.25.6
```

Creamos un pod:

```
adrian [ ~ ]$ kubectl create deployment nginx-deployment --image=nginx
deployment.apps/nginx-deployment created
```

Verificamos la creación del pod:

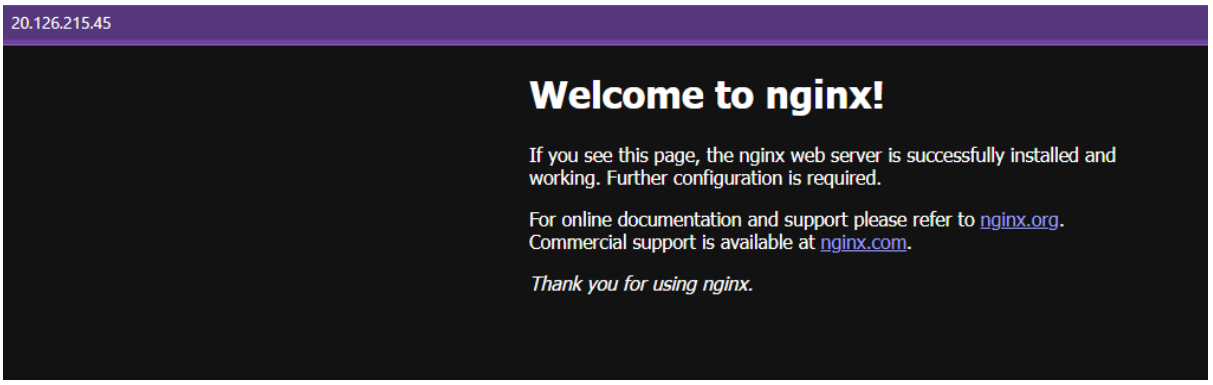
```
adrian [ ~ ]$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-5fbdf85c67-7pfrf   1/1     Running   0          32s
```

Y lo hacemos accesible desde internet:

```
adrian [ ~ ]$ kubectl expose deployment nginx-deployment --port=80 --type=LoadBalancer
service/nginx-deployment exposed
```

Después vemos la ip publica del pod para poder acceder al servicio.

```
adrian [ ~ ]$  kubectl get service
NAME               TYPE           CLUSTER-IP     EXTERNAL-IP     PORT(S)        AGE
kubernetes         ClusterIP      10.0.0.1       <none>          443/TCP        45m
nginx-deployment   LoadBalancer   10.0.61.224    20.126.215.45   80:30038/TCP   20m
```



Tarea 4.

Escalamos el número de pods de 1 a 2.

```
adrian [ ~ ]$ kubectl scale --replicas=2 deployment/nginx-deployment
deployment.apps/nginx-deployment scaled
```

Verificamos si se ha creado el pod nuevo.

```
adrian [ ~ ]$  kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-5fbdf85c67-cxhtk   0/1     Pending   0          6s
nginx-deployment-5fbdf85c67-vtpgv   1/1     Running   0          57s
```

Creamos un nuevo nodo en el cluster pasando de 1 a 2.

```
adrian [ ~ ]$   kubectl get nodes
NAME                                STATUS   ROLES   AGE   VERSION
aks-agentpool-11425770-vmss000000   Ready    agent   14m   v1.25.6
aks-agentpool-11425770-vmss000001   Ready    agent   72s   v1.25.6
```

Escalamos el número de pods a 10.

```
adrian [ ~ ]$  kubectl scale --replicas=10 deployment/nginx-deployment
deployment.apps/nginx-deployment scaled
adrian [ ~ ]$  kubectl get pods
NAME                                READY   STATUS              RESTARTS   AGE
nginx-deployment-5fbdf85c67-2ksjt   1/1     Running             0          12m
nginx-deployment-5fbdf85c67-2plsq   1/1     Running             0          10s
nginx-deployment-5fbdf85c67-2scgw   0/1     ContainerCreating   0          10s
nginx-deployment-5fbdf85c67-84jnh   1/1     Running             0          9m7s
nginx-deployment-5fbdf85c67-9gwp7   0/1     ContainerCreating   0          10s
nginx-deployment-5fbdf85c67-b2vzb   1/1     Running             0          10s
nginx-deployment-5fbdf85c67-gzn5n   0/1     ContainerCreating   0          10s
nginx-deployment-5fbdf85c67-hfkfk   0/1     ContainerCreating   0          10s
nginx-deployment-5fbdf85c67-n4pnt   1/1     Running             0          10s
nginx-deployment-5fbdf85c67-x276t   1/1     Running             0          10s
```

Vemos la distribución de los pods en los nodos.

```
adrian [ ~ ]$  kubectl get pod -o=custom-columns=NODE:.spec.nodeName,POD:.metadata.name
NODE                                POD
aks-agentpool-11425770-vmss000000   nginx-deployment-5fbdf85c67-2ksjt
aks-agentpool-11425770-vmss000000   nginx-deployment-5fbdf85c67-2plsq
aks-agentpool-11425770-vmss000001   nginx-deployment-5fbdf85c67-2scgw
aks-agentpool-11425770-vmss000000   nginx-deployment-5fbdf85c67-84jnh
aks-agentpool-11425770-vmss000001   nginx-deployment-5fbdf85c67-9gwp7
aks-agentpool-11425770-vmss000000   nginx-deployment-5fbdf85c67-b2vzb
aks-agentpool-11425770-vmss000001   nginx-deployment-5fbdf85c67-gzn5n
aks-agentpool-11425770-vmss000001   nginx-deployment-5fbdf85c67-hfkfk
aks-agentpool-11425770-vmss000000   nginx-deployment-5fbdf85c67-n4pnt
aks-agentpool-11425770-vmss000001   nginx-deployment-5fbdf85c67-x276t
```

Y borramos la implementación.

```
adrian [ ~ ]$  kubectl delete deployment nginx-deployment
deployment.apps "nginx-deployment" deleted
```

Y terminamos limpiando los recursos creados en esta práctica.

```
adrian [ ~ ]$ az group list --query "[?starts_with(name,'az104-09c')].name" --output tsv
az104-09c-rg1
adrian [ ~ ]$ az group list --query "[?starts_with(name,'az104-09c')].[name]" --output tsv | xargs -L1 bash -c 'az group delete --name $0 --no-wait --yes'
```