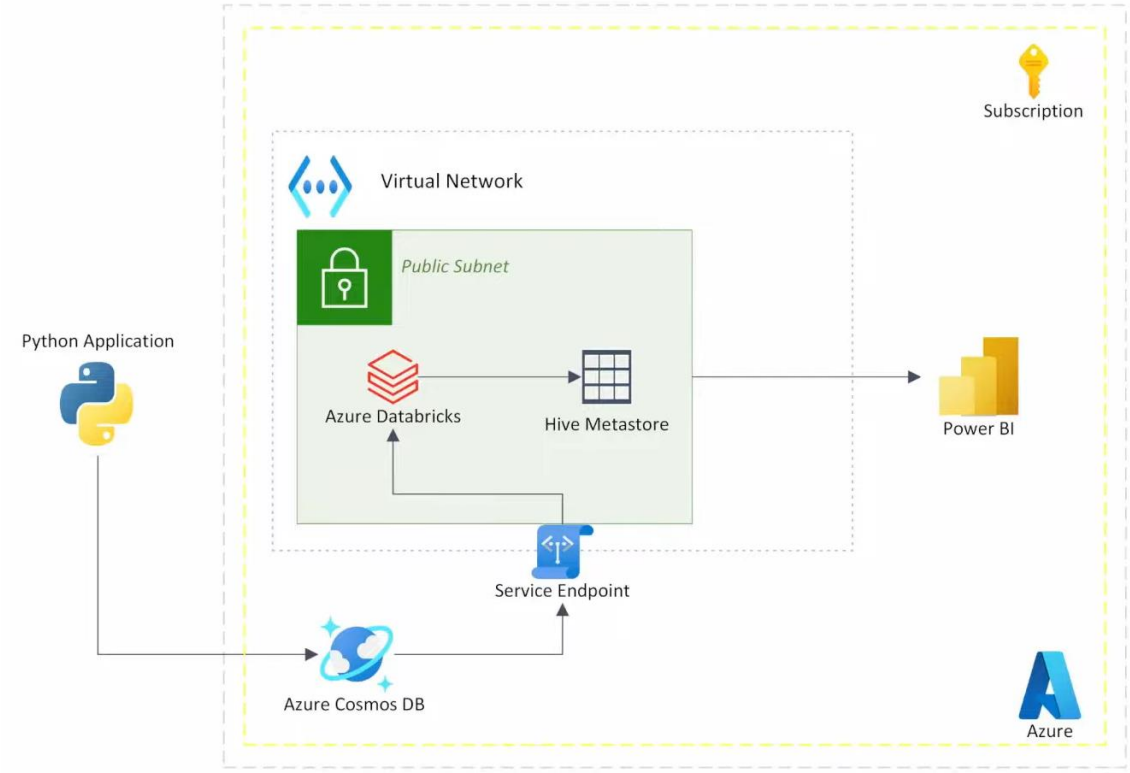
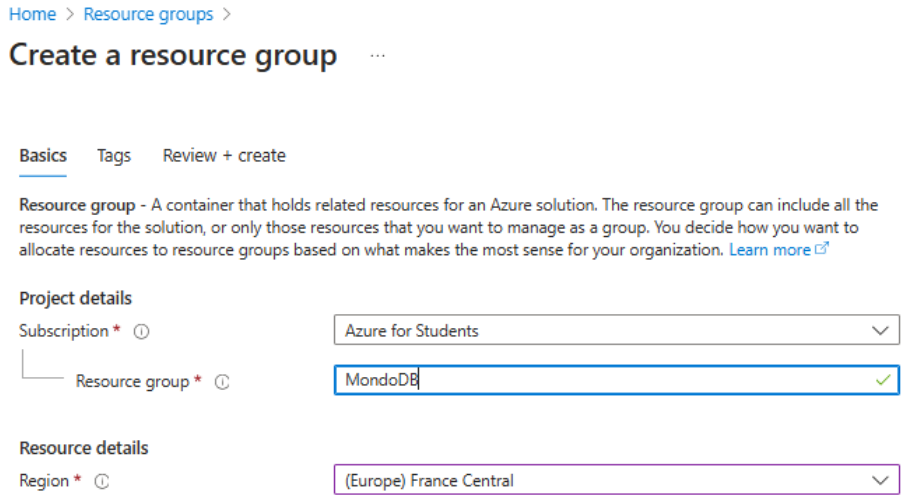
Azure Cosmos DB API para MongoDB utilizando Spark y Databricks con PowerBI mediante Partner Connect

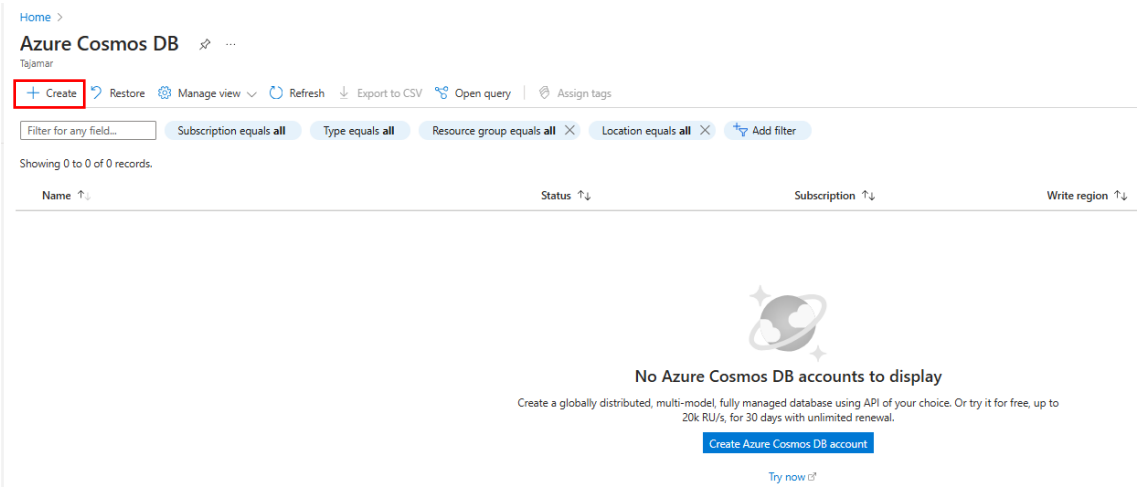
Arquitectura:



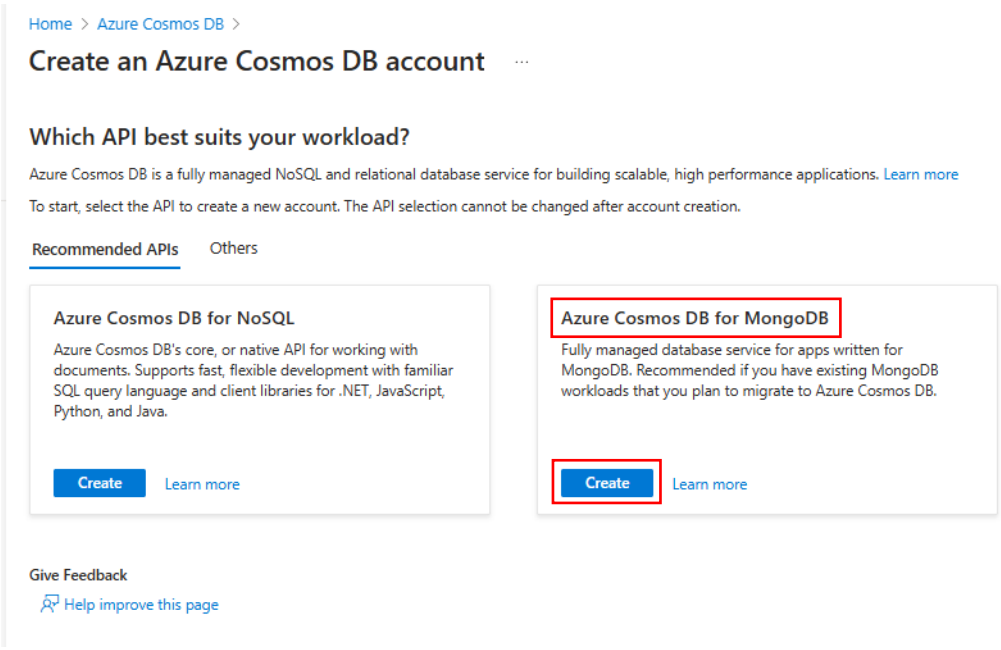
Creamos el grupo de recursos



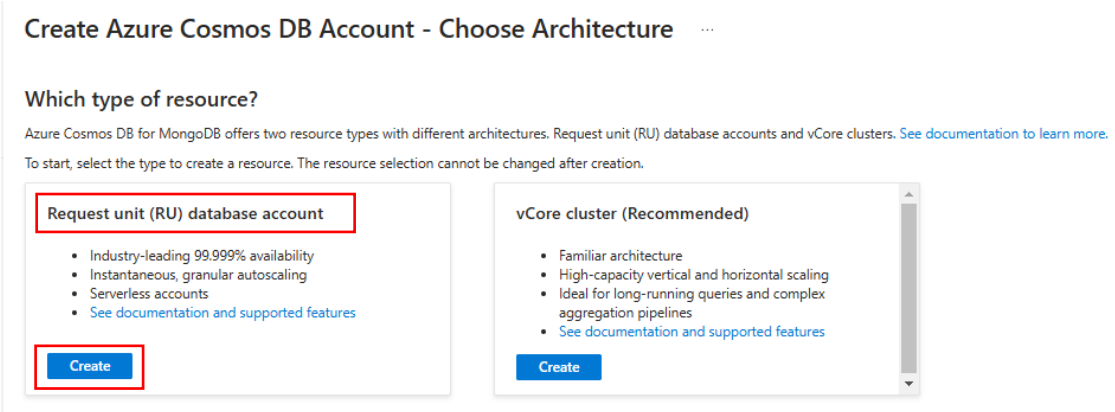
Azure Cosmos DB



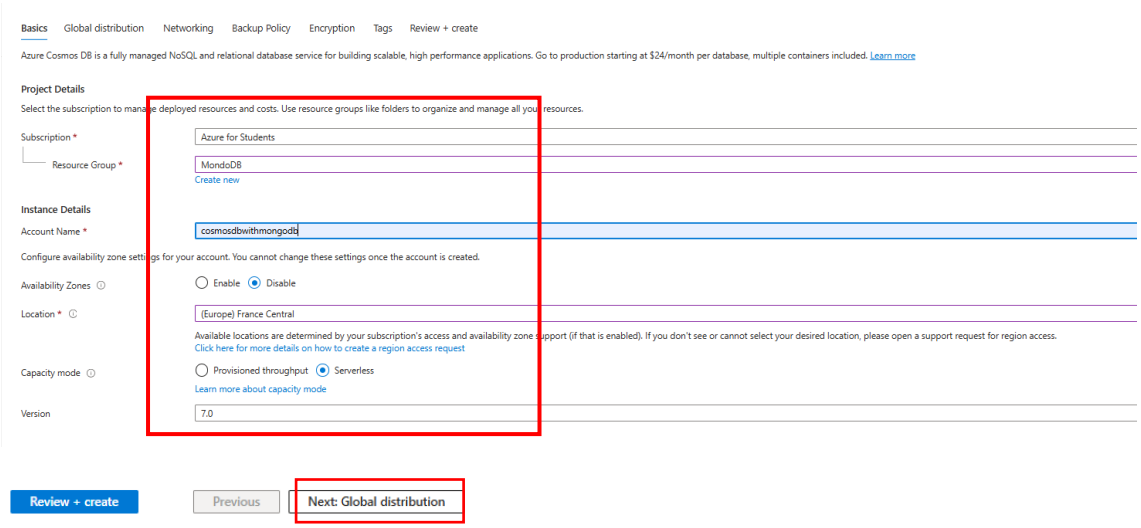
Click en create



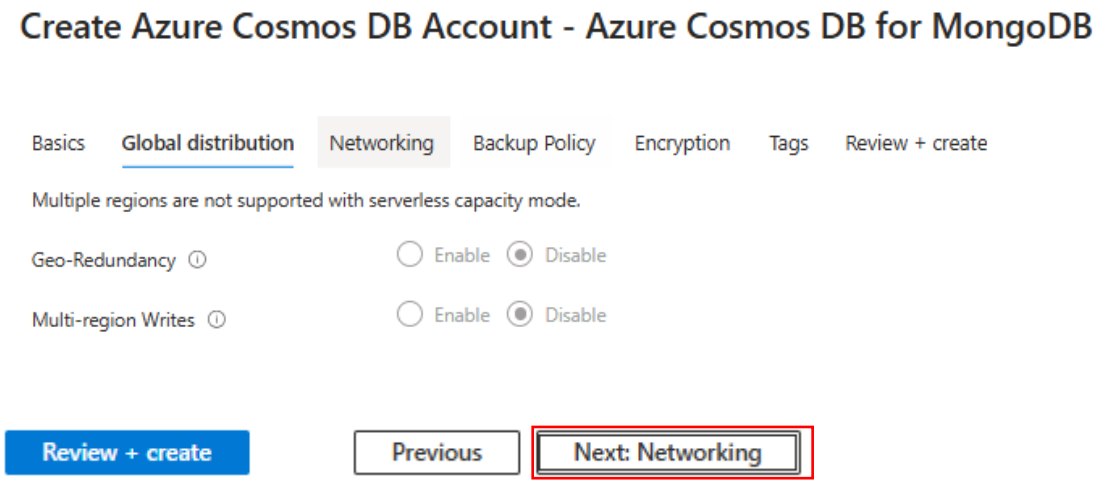
Escoger Azure Cosmos DB for MongoDB luego create



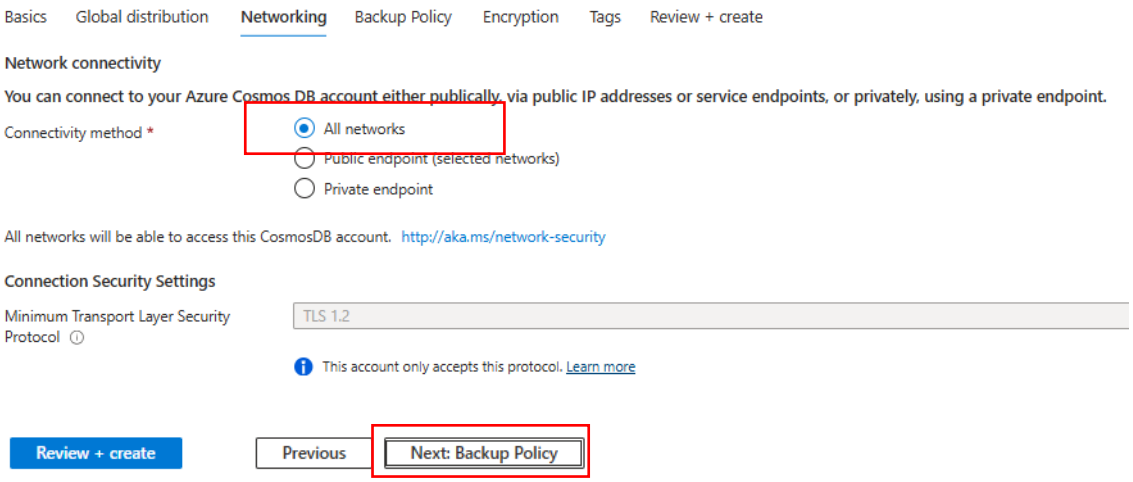
Por ahora, seleccionamos el Serverless (Request unit – RU database account)



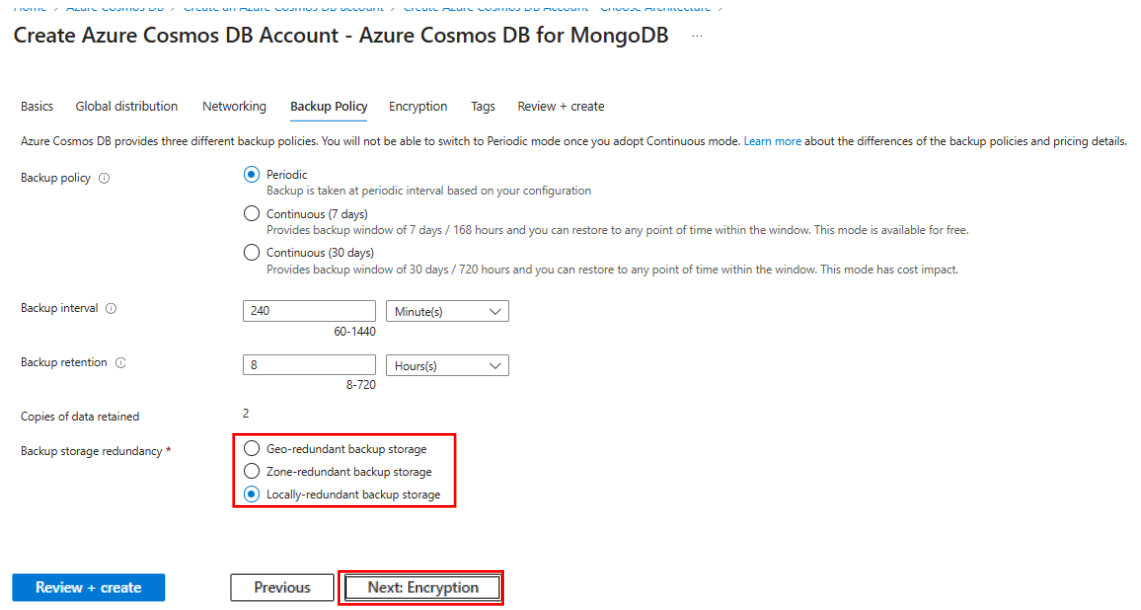
Next: Global distribution



Disable, luego Next: Networking



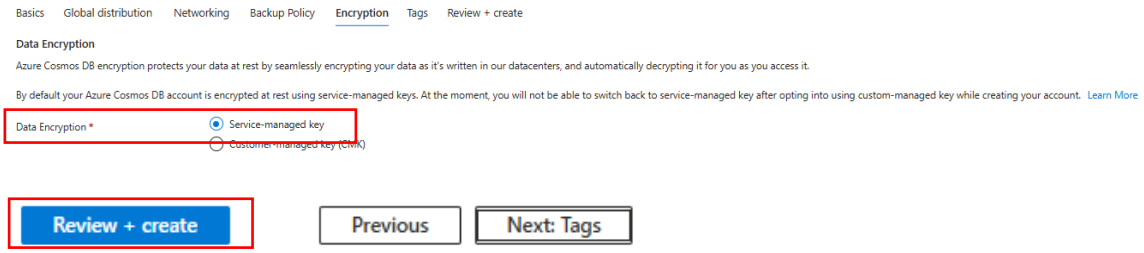
All networks, luego Next: Backup Policy



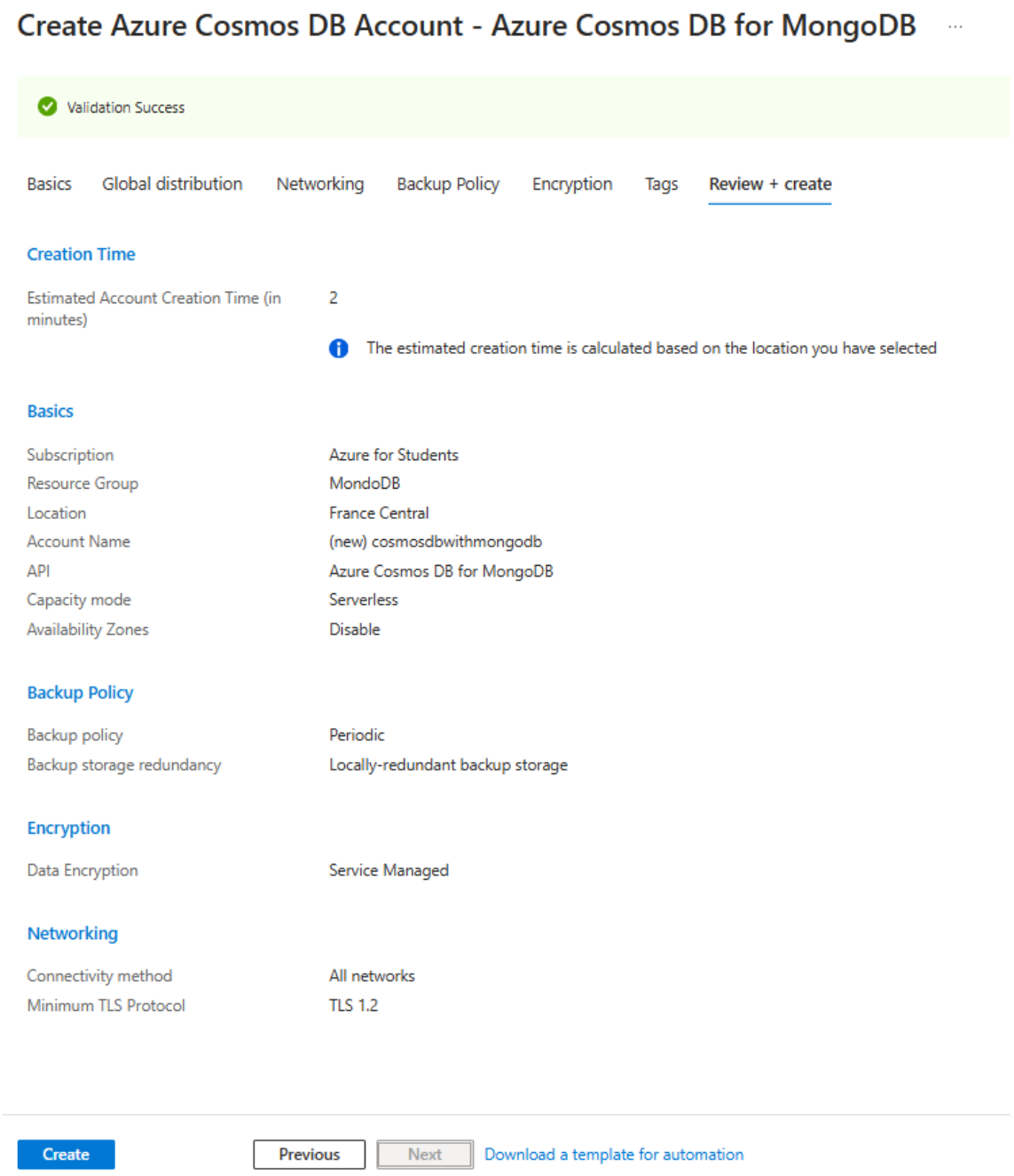
Periodic

Locally-redundant backup storage

Lo demás lo que se muestra por defecto. Luego click en Next: Encryption

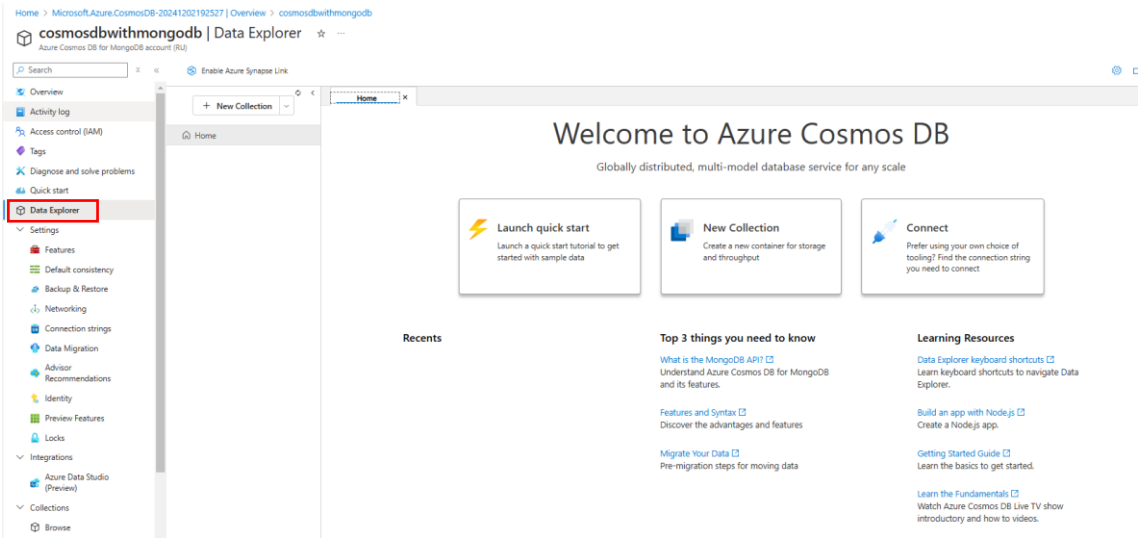


Service-managed key, luego click en Review + Create

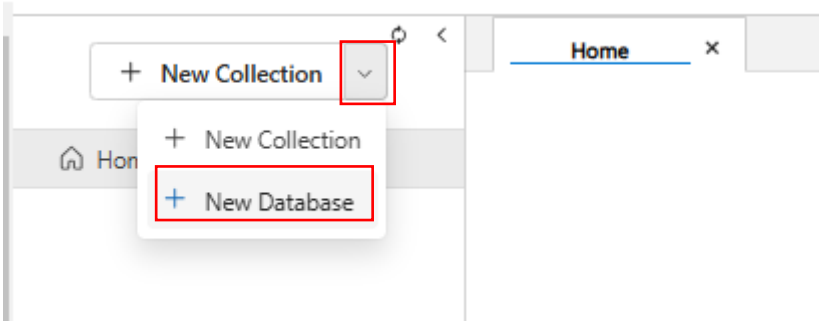


Create

Una vez efectuado el deploy ir a go to ressources luego click Data Explorer

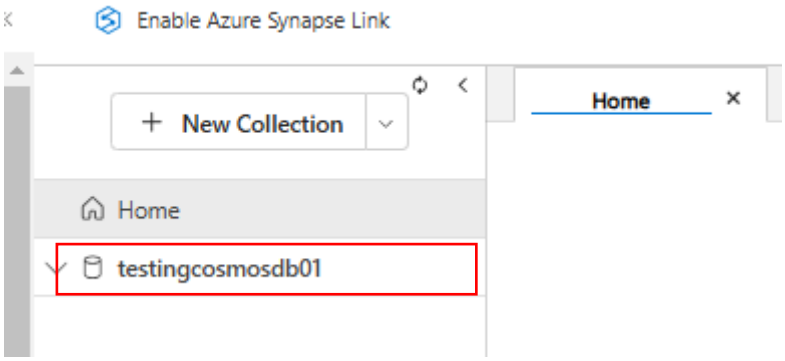


Luego click en +New Collection y seleccionar New Database

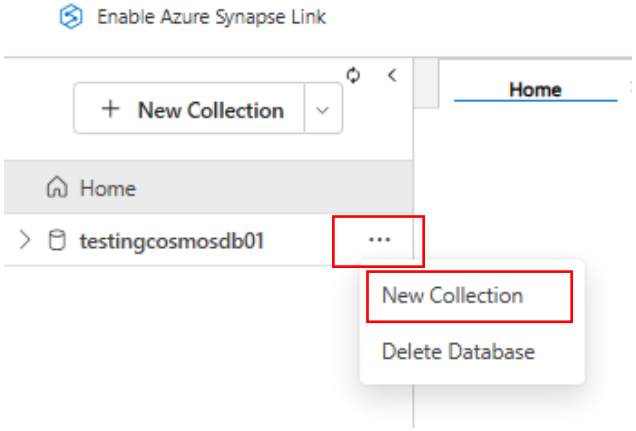


En Database id asignamos un nombre, luego OK

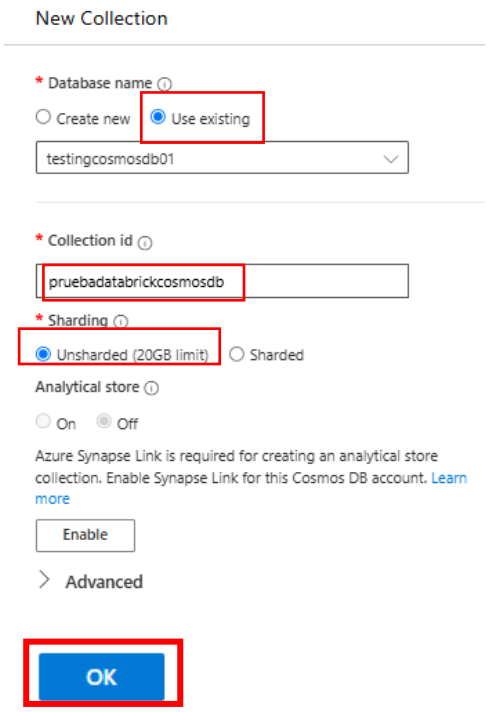
Una vez creado debes verlo así:



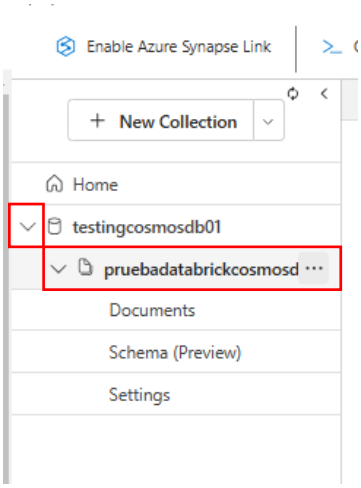
Click en los tres puntos y luego new collection:



Use existing, añadir un nombre en Collection id, seleccionar Unsharded



Una vez creado debería verse esto:



Entra al sitio web OpenWeather y regístrate.

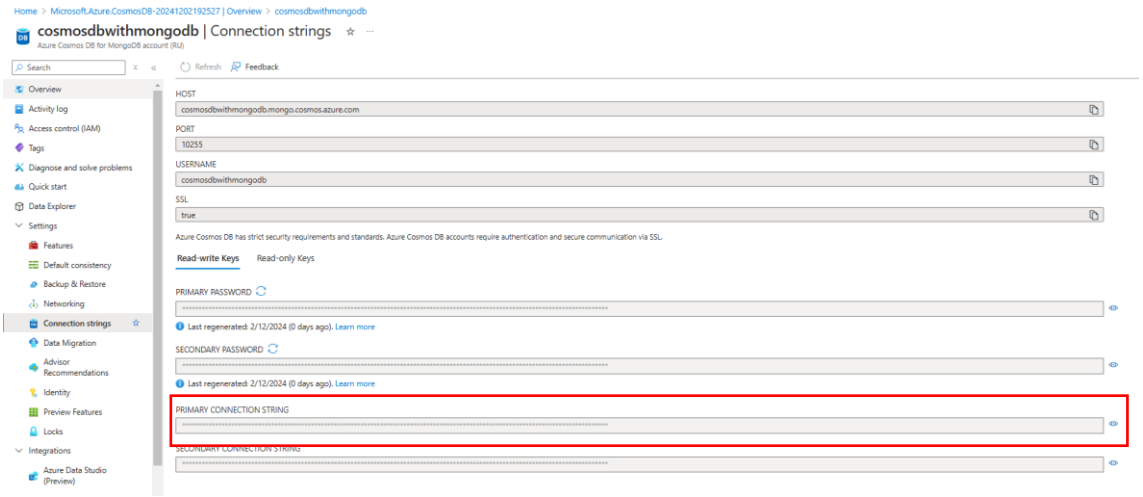
Accede a la sección de API Keys en tu perfil después de registrarte.

Copia la API Key que se te proporciona

Reemplaza tu_api_key en el ***script*** con la Key que has copiado.

```
# Configuración de OpenWeather API
API_KEY = "8b73515bcec2ae54c33e6d422f504f2a"  # Reemplaza con tu API Key
CITY = "Toronto"  # Cambia por la ciudad deseada
WEATHER_URL = f"http://api.openweathermap.org/data/2.5/weather?q={CITY}&appid={API_KEY}&units=metric"
```

En Azure, en tu CosmosDB , ve a Connection strings y copia PRIMARY CONNECTION STRING:



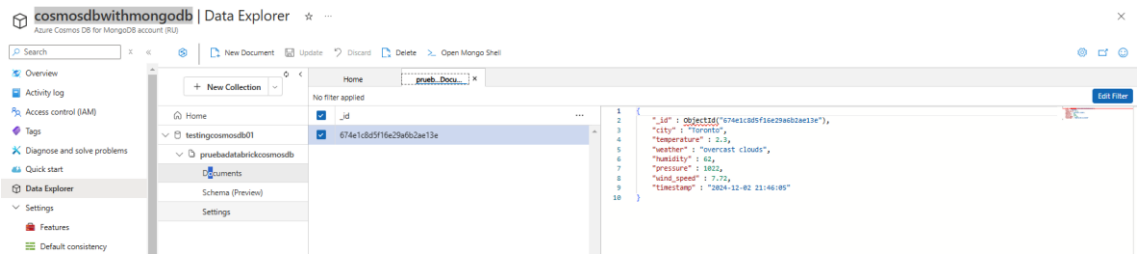
Pegar tu PRIMARY CONNECTION STRING en el script (weather_to_cosmodb.py)

```
# Configuración de Cosmos DB
DB_NAME = "testingcosmosdb01"  # Nombre de tu base de datos
COLLECTION_NAME = "pruebadatabrickcosmosdb"  # Nombre de tu colección
CONNECTION = "mongodb://cosmosdbwithmongodb:fUTd69EFQvoWifX0yn4sBTJ3e0yJoMaRAC3hUriFSZnAKlheVJwezrgr01Jw5GVnFjLWnho3YkRcACDbPhcU3g==@cosmosdbwithr
```

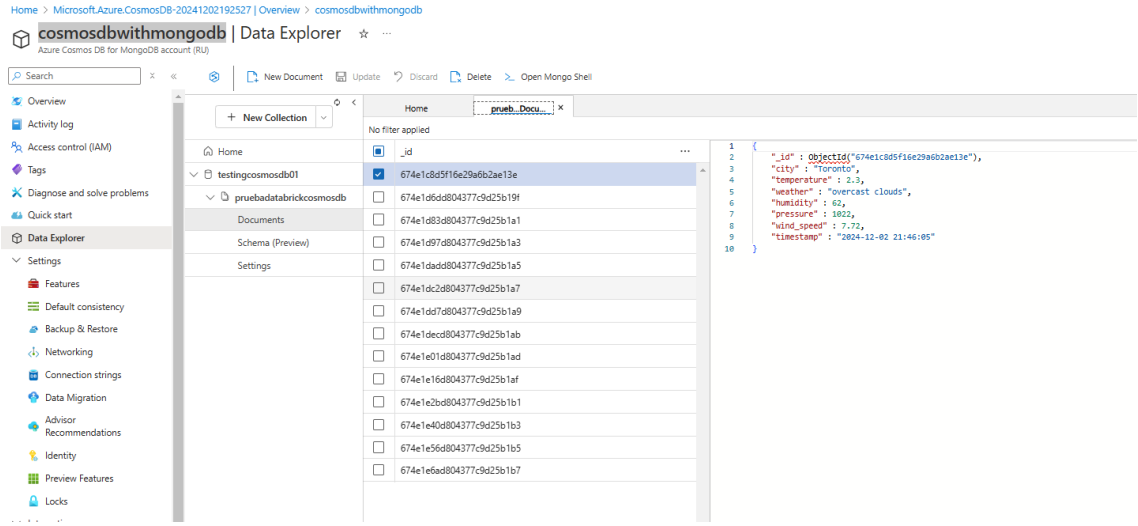
En DB_NAME cambiar por el nombre de tu base de datos.

En COLLECTION_NAME cambiar por el nombre de colección.

Guardar y ejecutar el script, debería imprimir los datos del clima obtenidos y confirmar que se han insertado en tu base de datos Cosmos DB



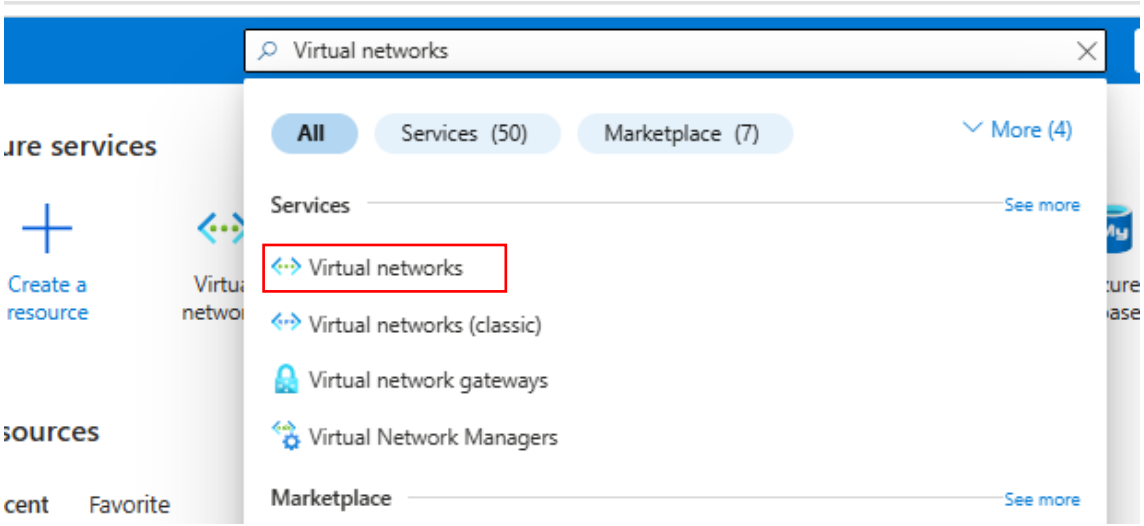
Cada 20 segundos está capturando datos del clima y enviado a tu Cosmos DB:

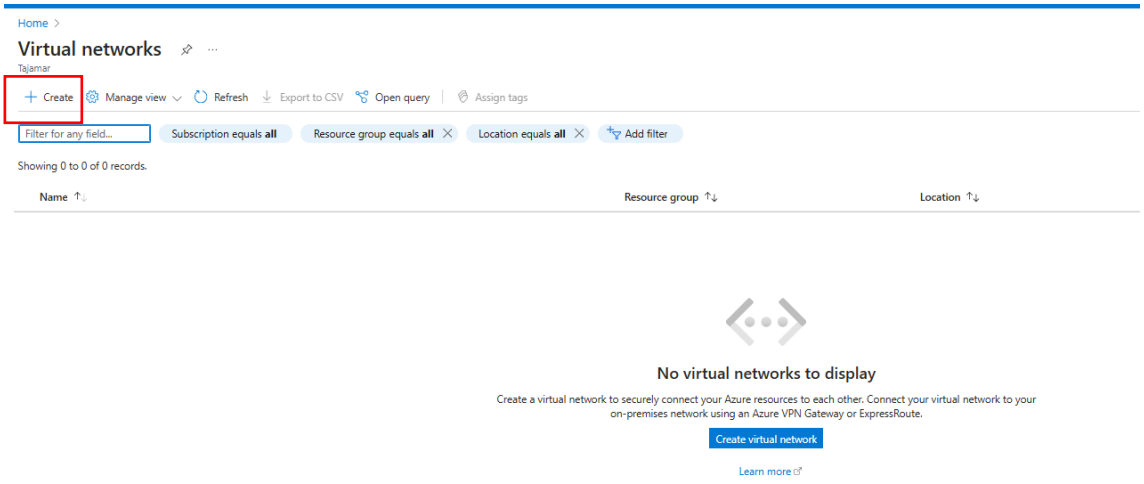


Para detener la captura de datos escribimos CTRL + C en el terminal

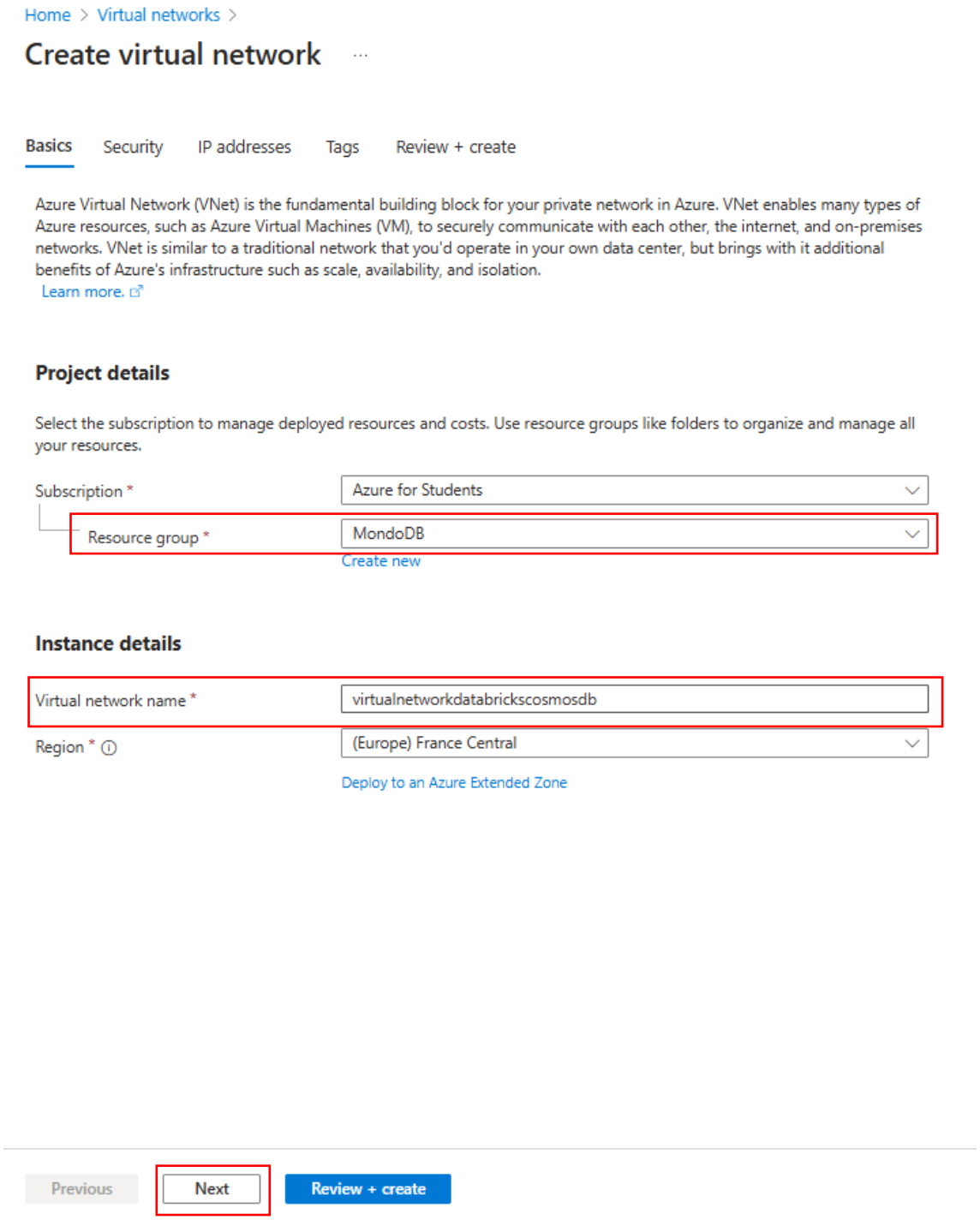
Crear una red virtual.

En el buscador de Azure escribimos Virtual Networks:

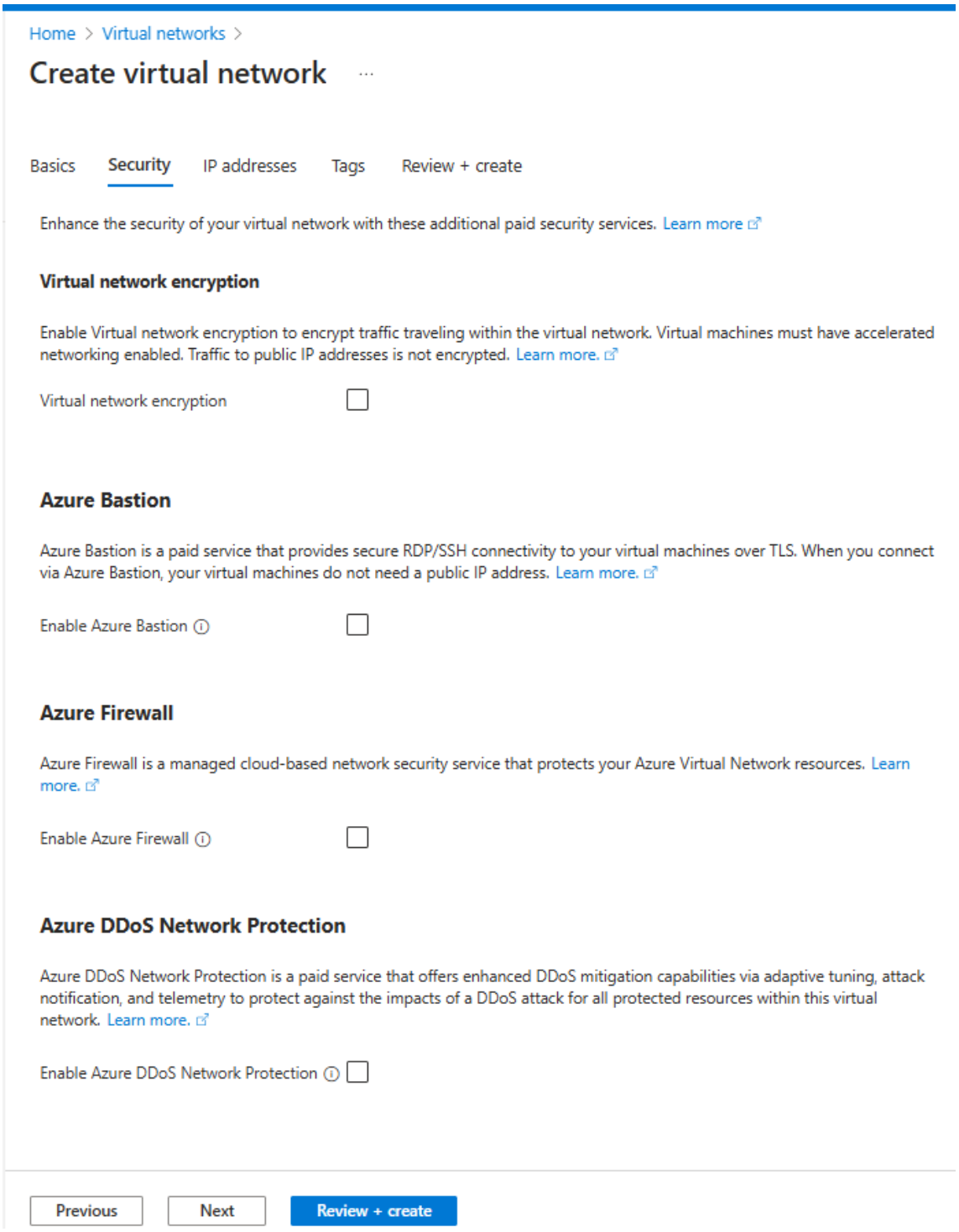




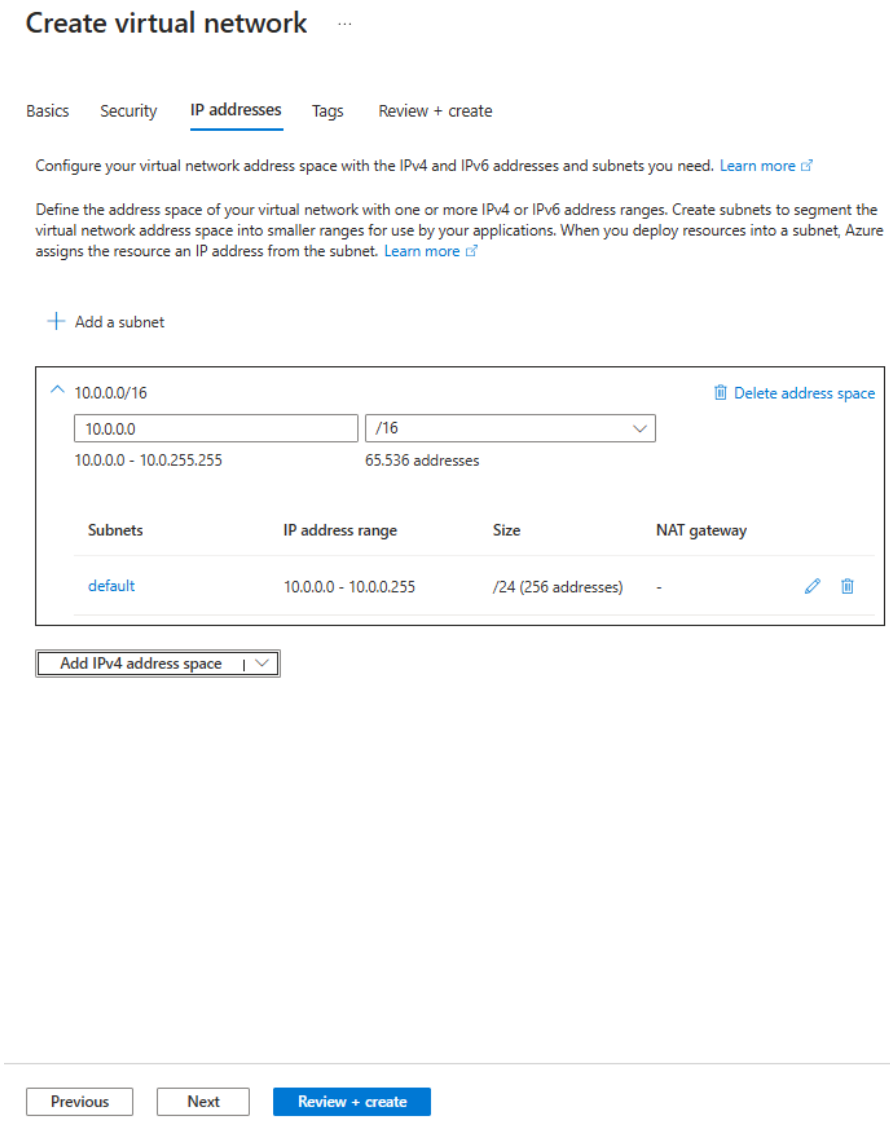
Click en Create.



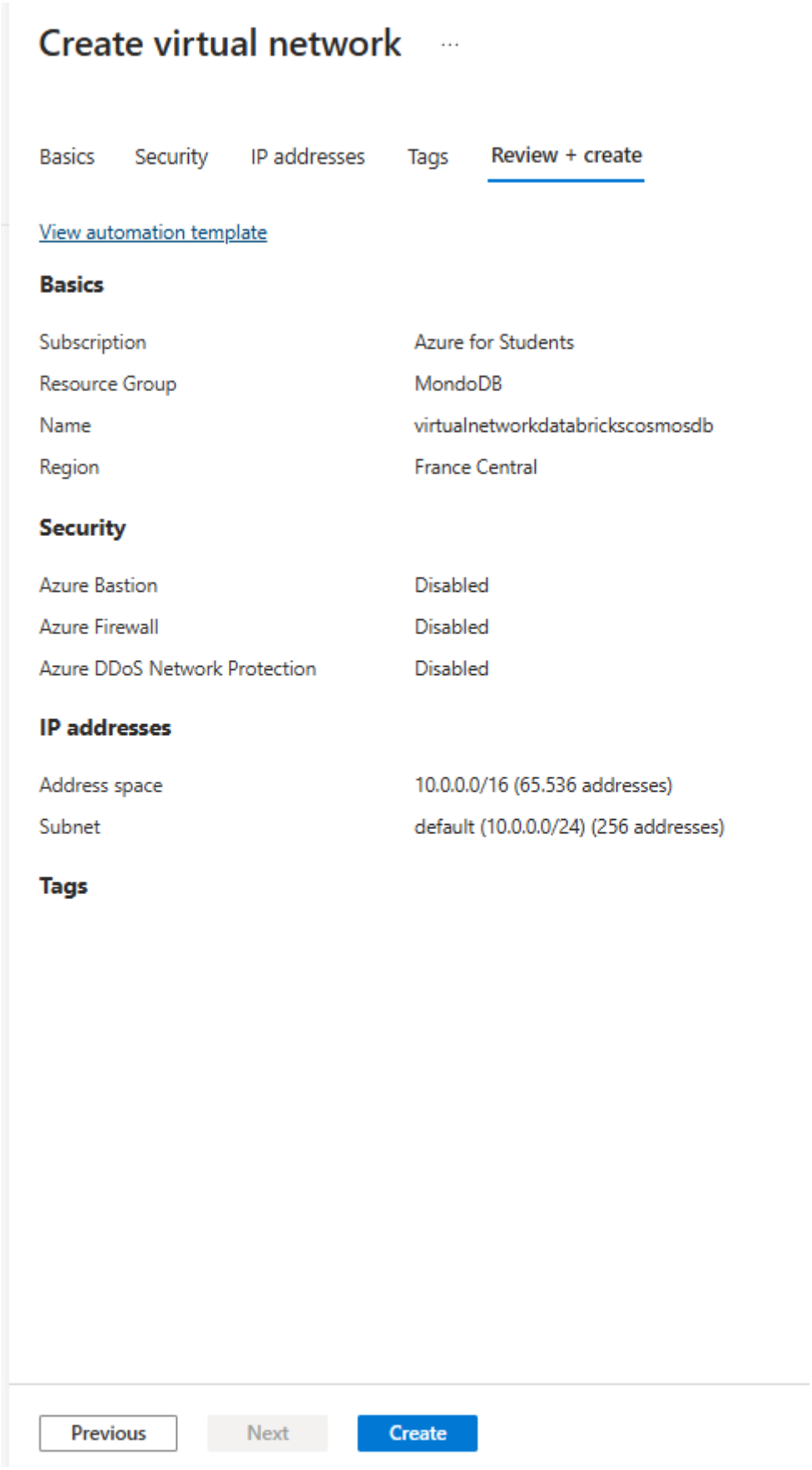
Darle nombre a tu Virtual network, luego Next:



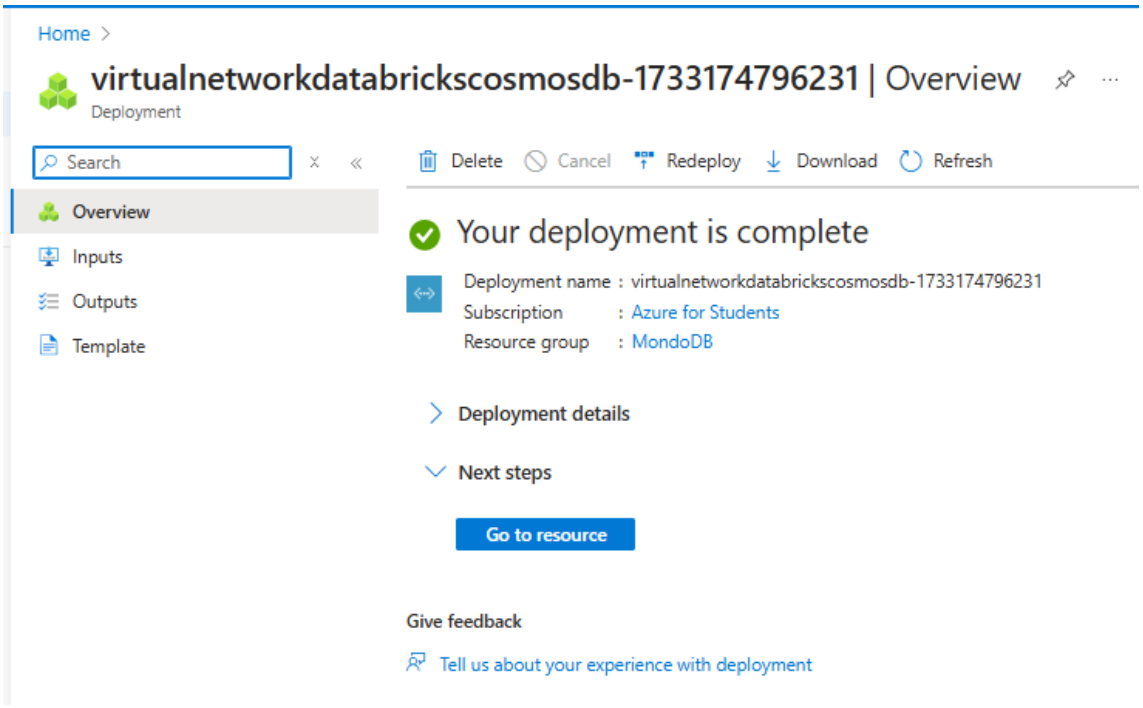
Dejar todas las casillas desactivadas, luego click en Next:



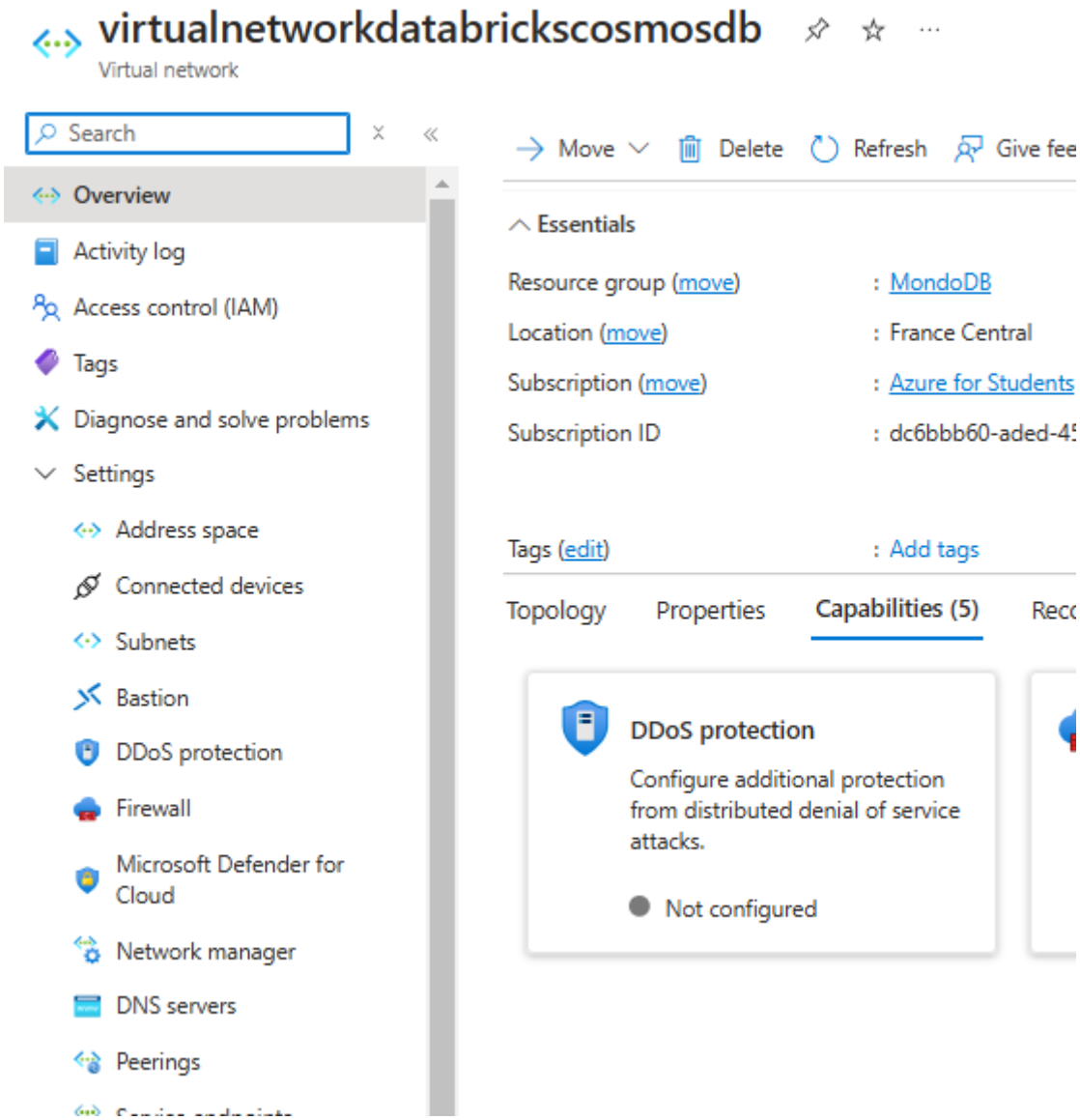
Dejar lo que aparece por defecto, luego Review + Create


Create virtual network
...
Basics
Security
IP addresses
Tags
Review + create
View automation template
Basics
Subscription
Azure for Students
Resource Group
MondoDB
Name
virtualnetworkdatabrickscosmosdb
Region
France Central
Security
Azure Bastion
Disabled
Azure Firewall
Disabled
Azure DDoS Network Protection
Disabled
IP addresses
Address space
10.0.0.0/16 (65.536 addresses)
Subnet
default (10.0.0.0/24) (256 addresses)
Tags
Previous
Next
Create

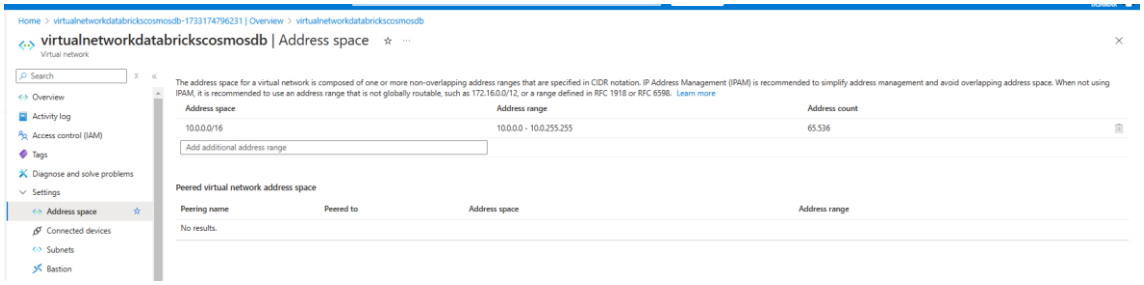
Create



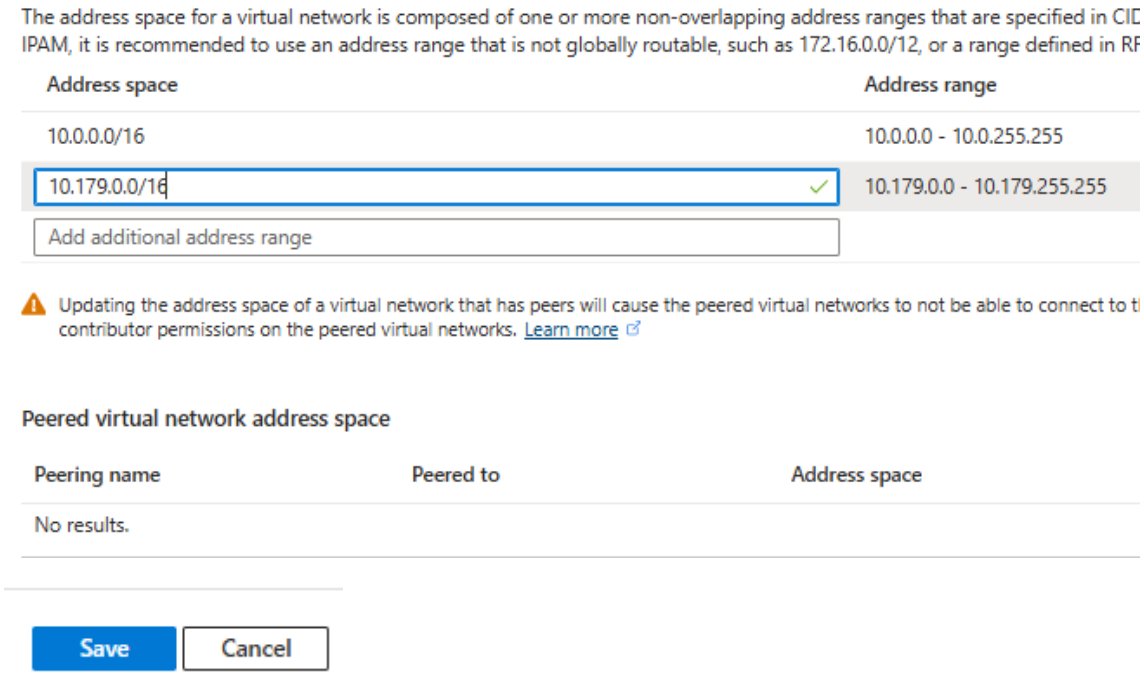
Click en Go to resource



Click en Address space

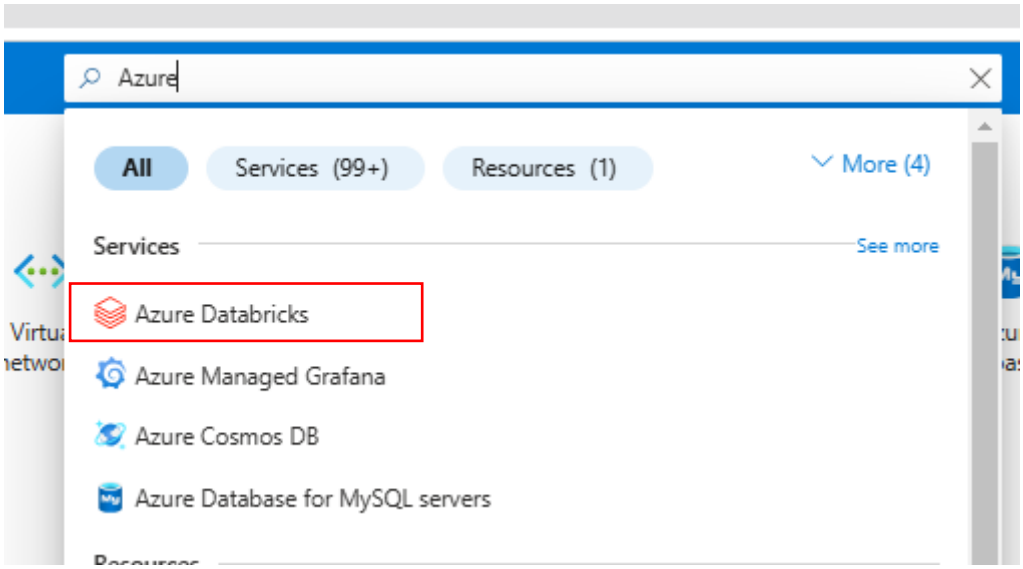


Añadimos otra subred para los servicios de Databricks:

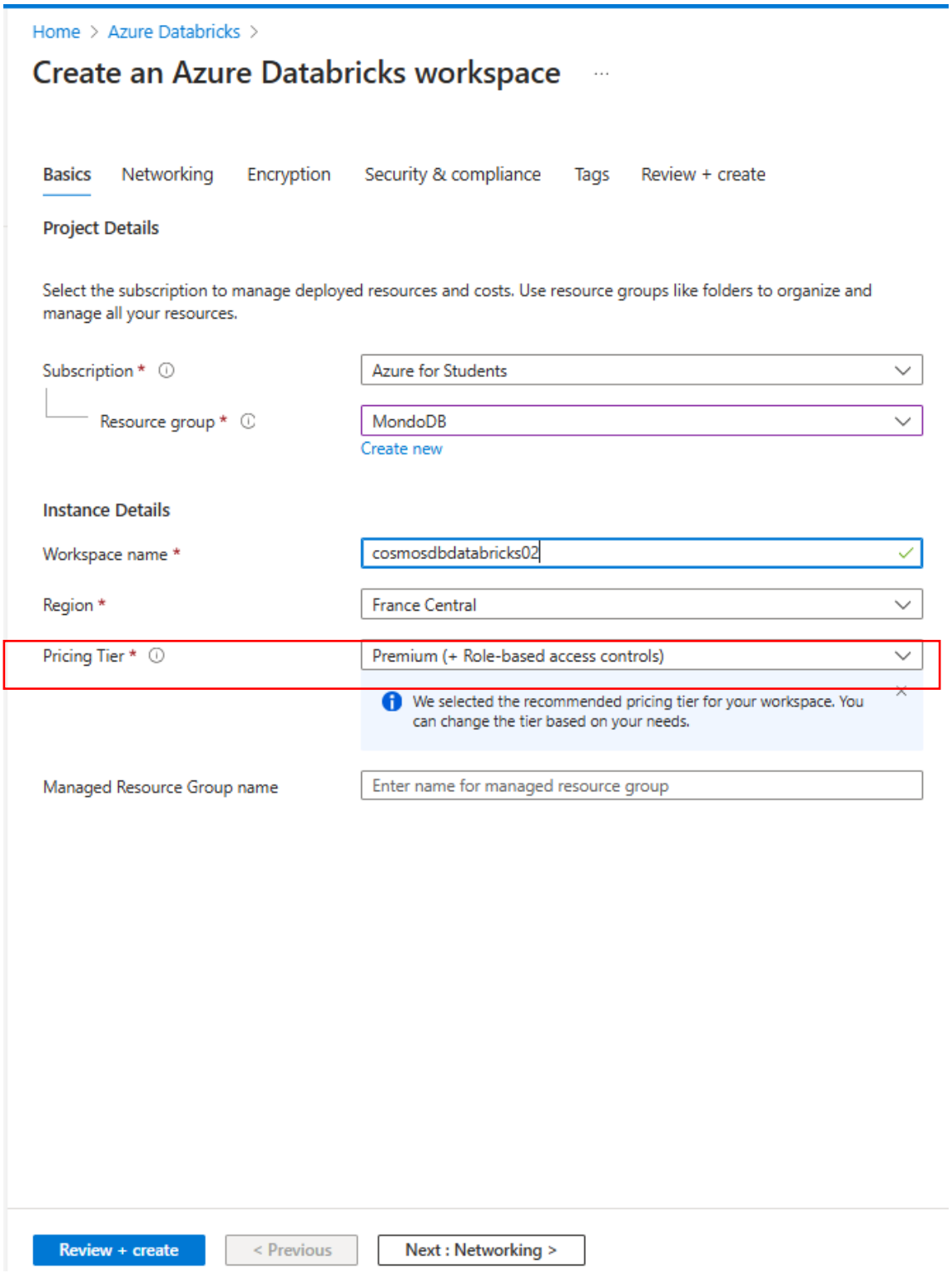


Click en Save

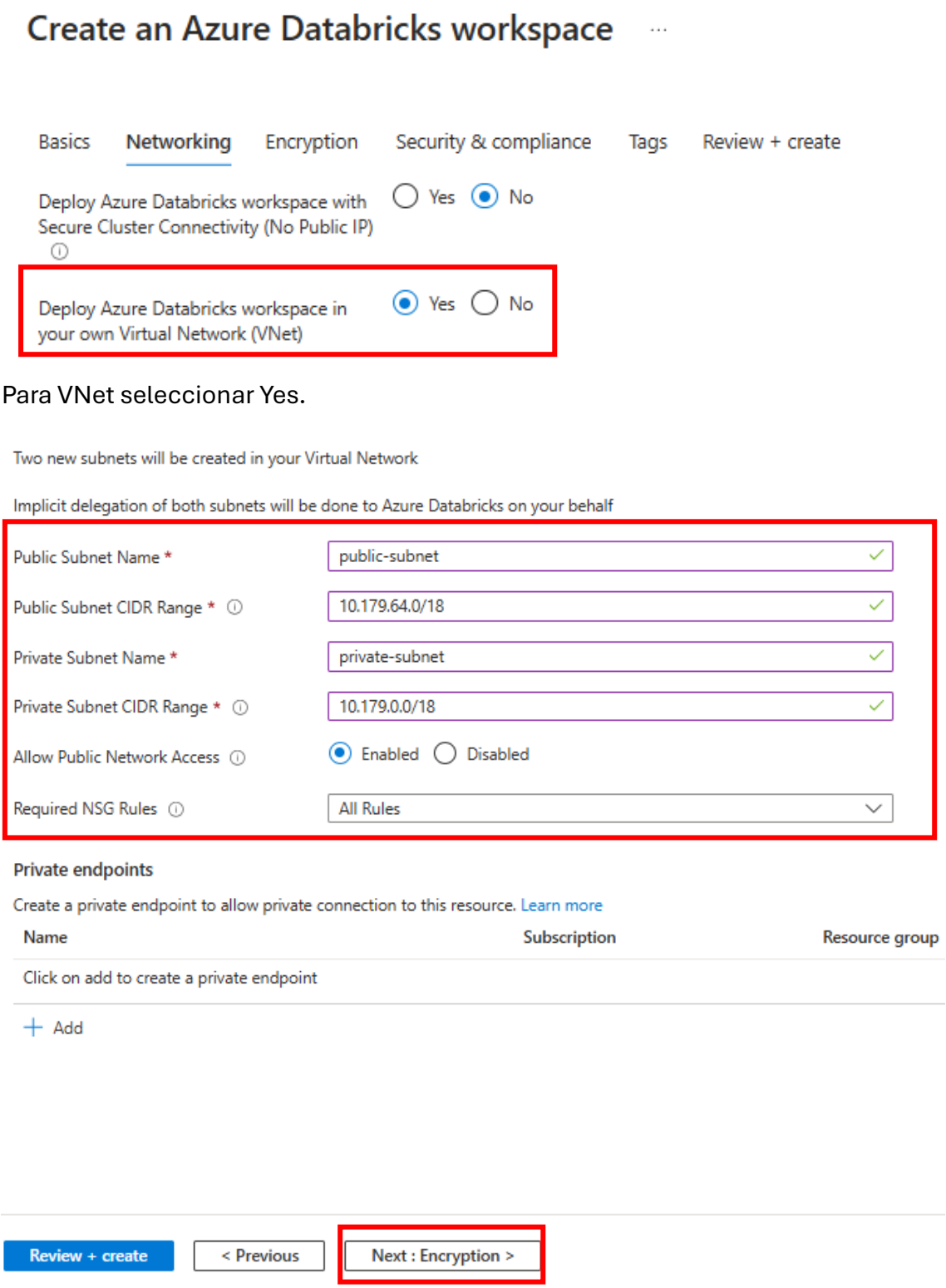
Crear un recursos de Azure Databricks:



En Pricing Tier seleccionar Premium para poder usar Power BI:

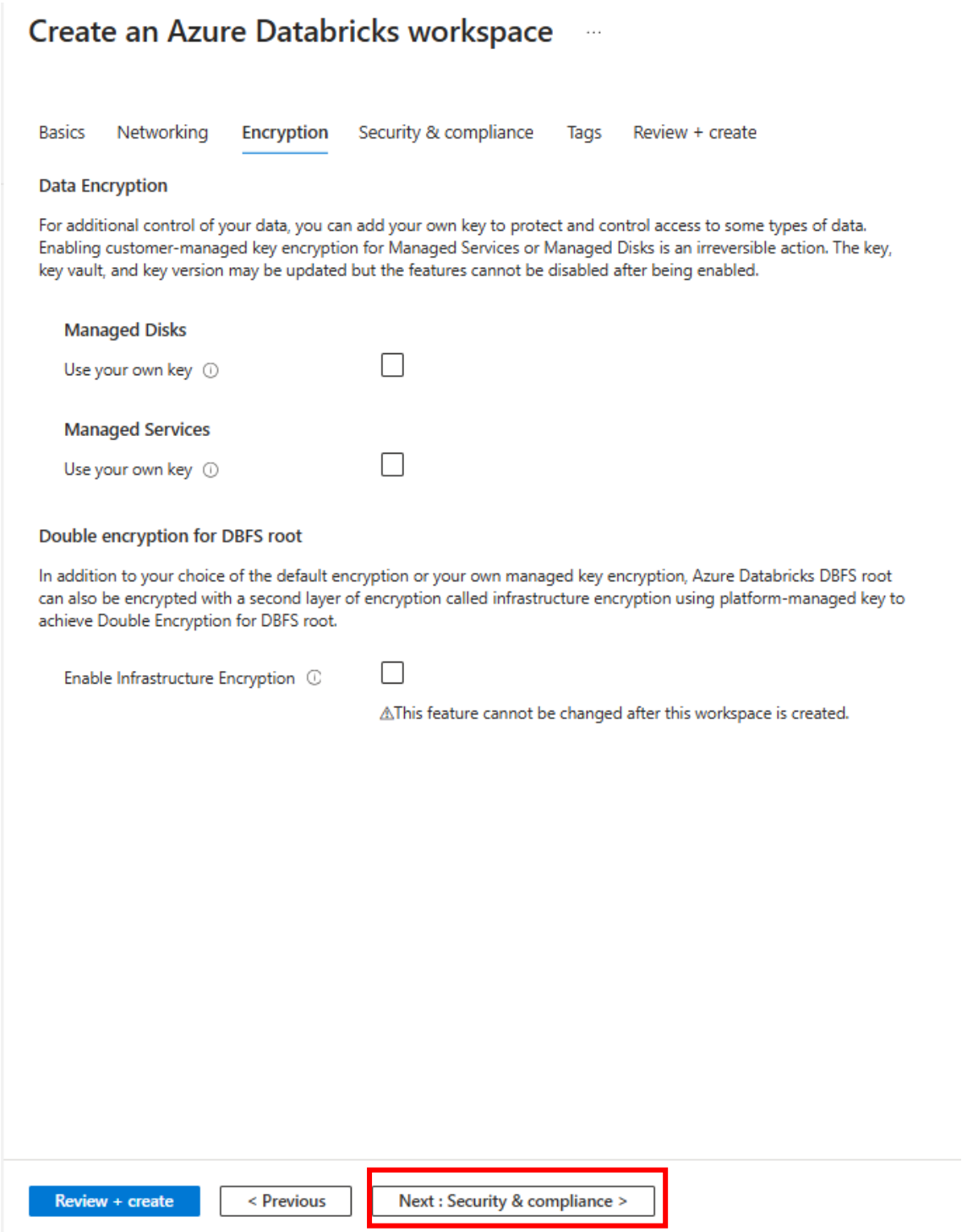


Click en Next: Networking

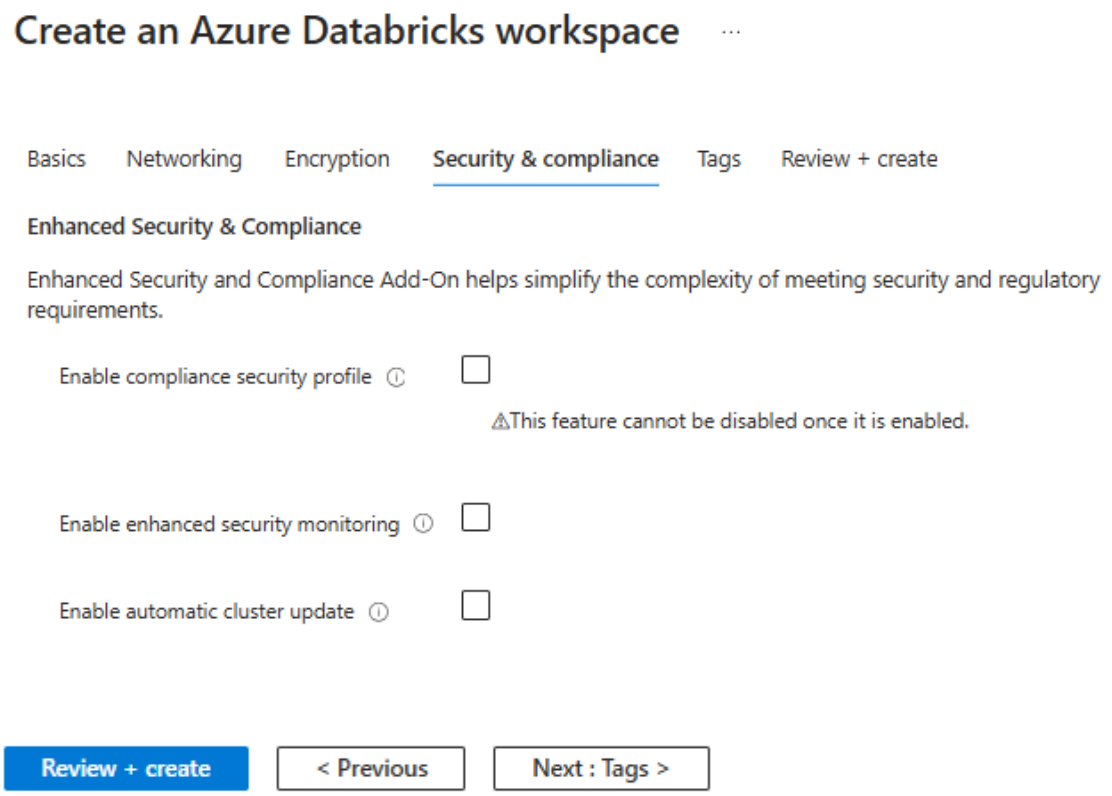


Para VNet seleccionar Yes.

Click en Next: Encription



No habilitar ninguna opción. Click en Next: Security...



No habilitar ninguna opción, luego click en Review + Create

Create an Azure Databricks workspace ...

Validation Succeeded

Basics | Networking | Encryption | Security & compliance | Tags | Review + create

Summary

Basics

Workspace name	cosmosdbdatabricks02
Subscription	Azure for Students
Resource group	MondoDB
Region	France Central
Pricing Tier	premium
Managed Resource Group name	

Networking

Deploy Azure Databricks workspace with Secure Cluster Connectivity (No Public IP)	No
Deploy Azure Databricks workspace in your own Virtual Network (VNet)	Yes
Virtual Network	virtualnetworkdatabrickscosmosdb
Public Subnet Name	public-subnet
Public Subnet CIDR Range	10.179.64.0/18
Private Subnet Name	private-subnet
Private Subnet CIDR Range	10.179.0.0/18
Allow Public Network Access	Enabled
Required NSG Rules	All Rules

Encryption

Enable Infrastructure Encryption	No
Enable CMK for Managed Disks	No
Enable CMK for Managed Services	No

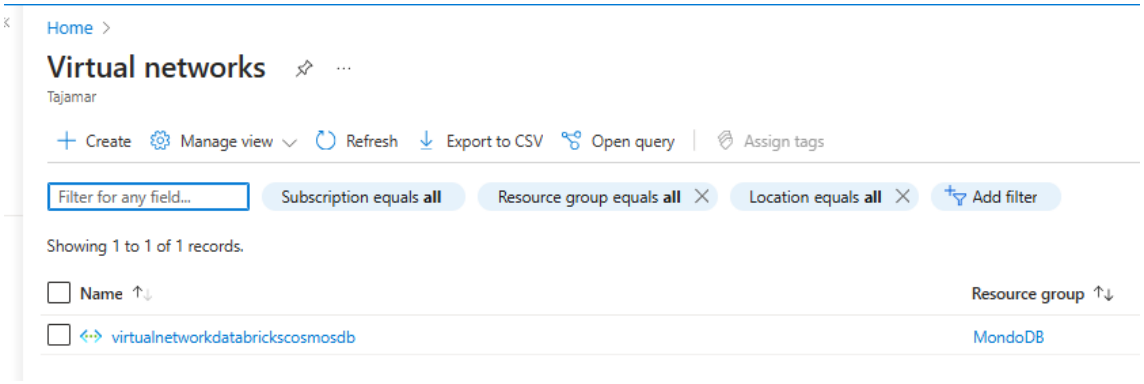
Security & compliance

Create | < Previous | Download a template for automation

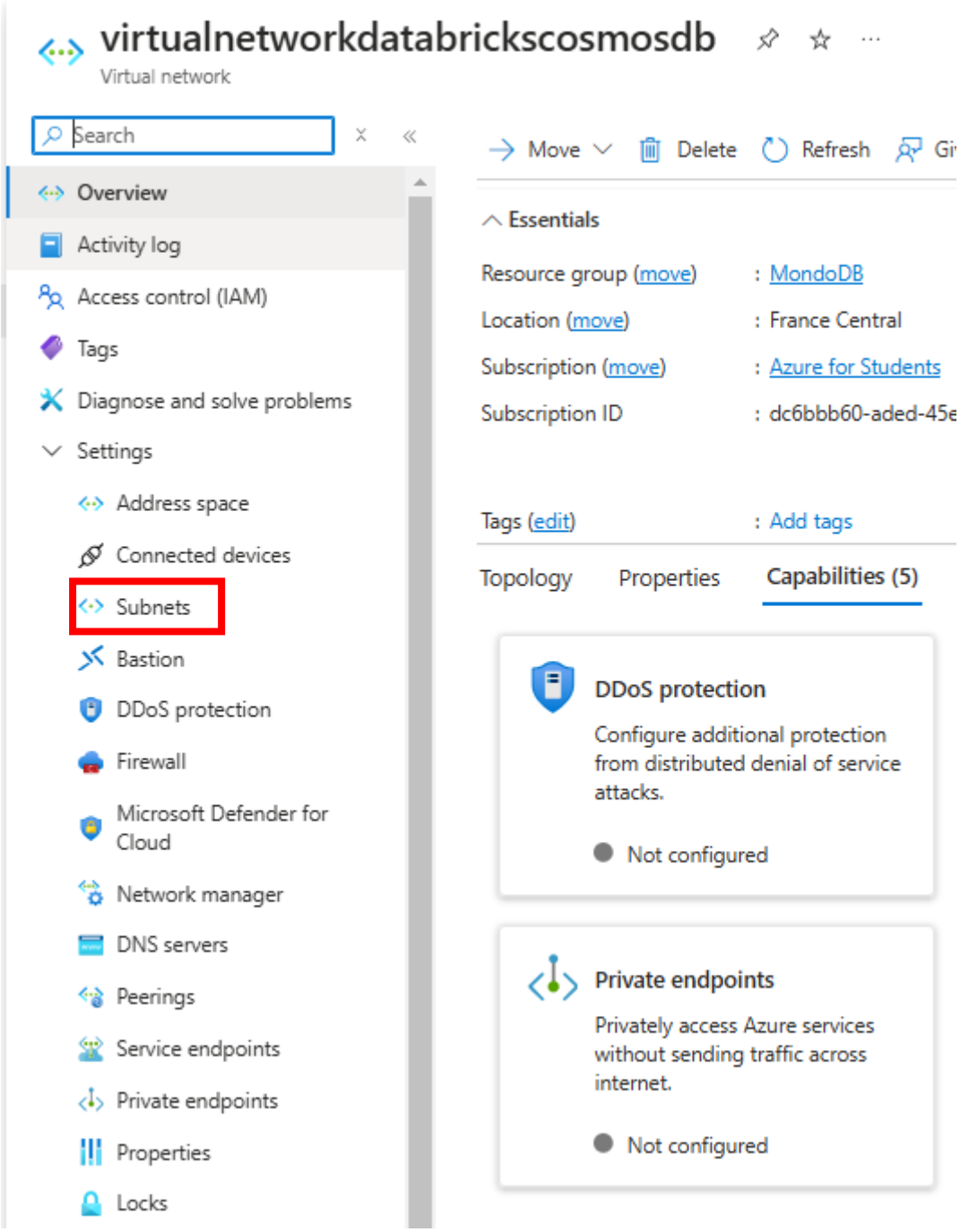
Click en Create

Crear un Service Endpoint para Cosmos DB

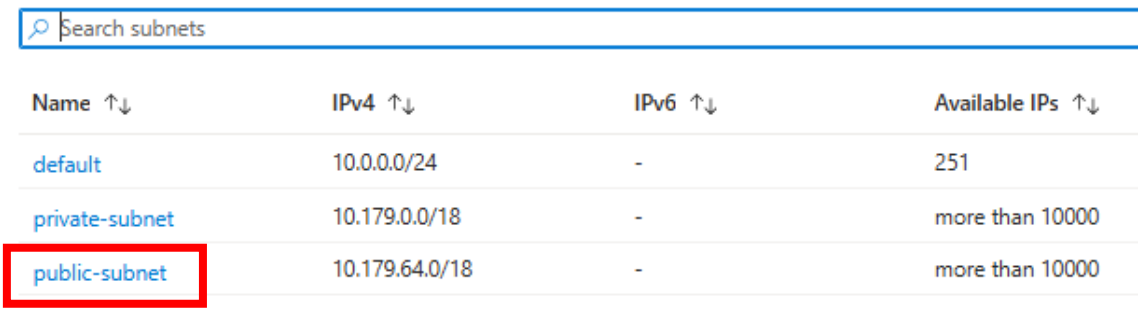
Vamos a nuestra virtual network:



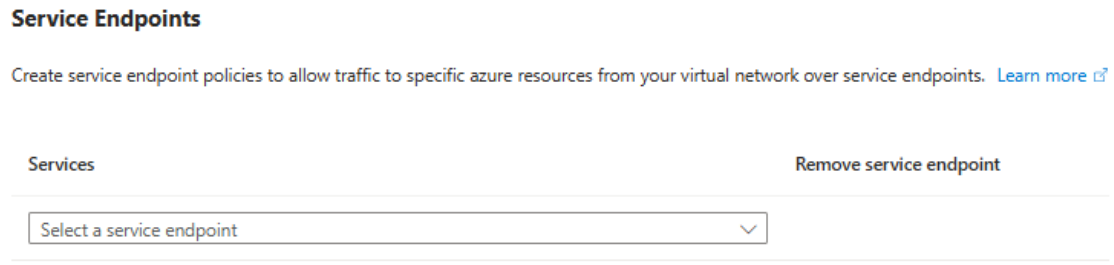
Luego clic en subnets:

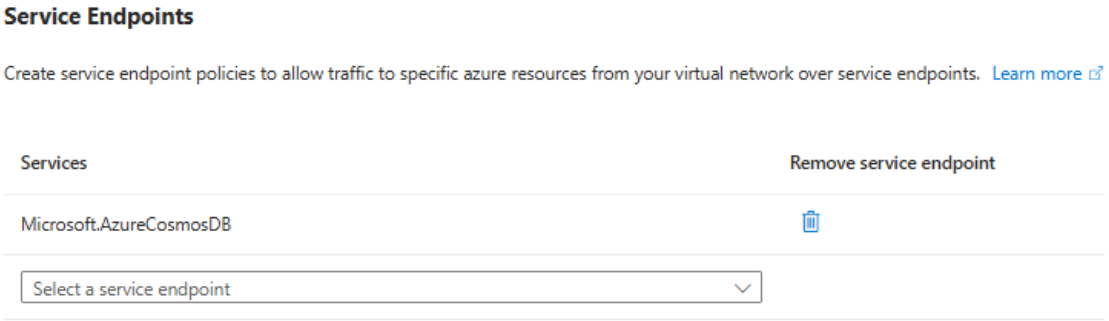


Click en public-subnet:



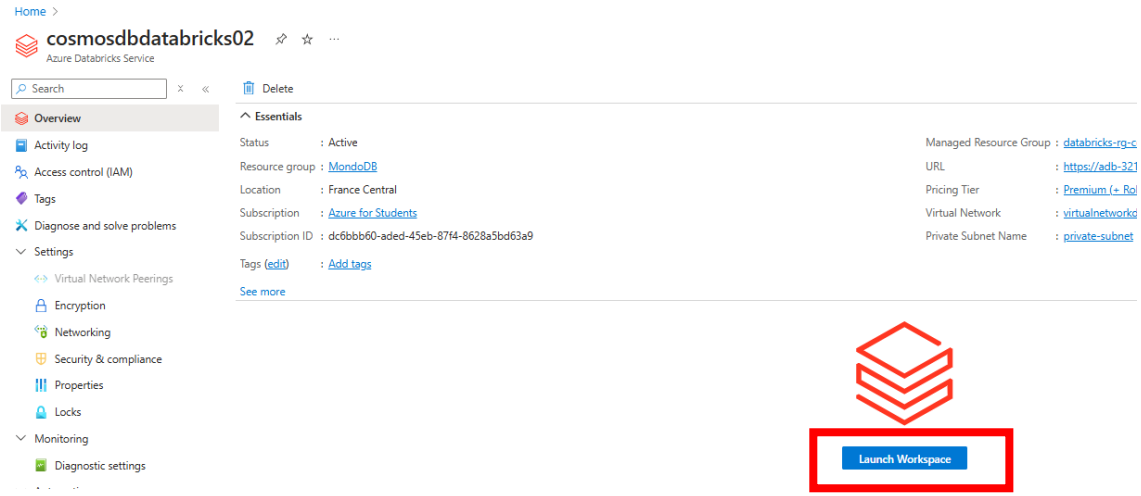
En Services Endpoint seleccionar Micorsoft AzureCosmosDB



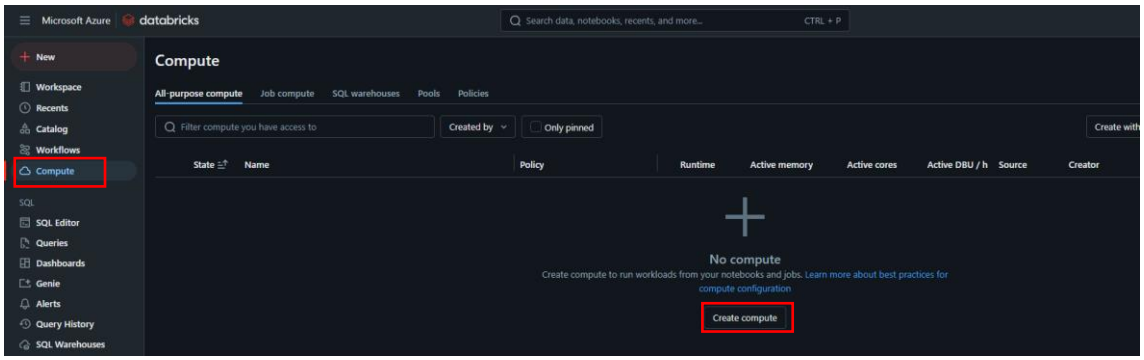


Click en Save.

Vamos al servicio de Azure Databricks y damos click en Launch Workspace:



Creamos un cluster, click en Compute y luego Create Compute:



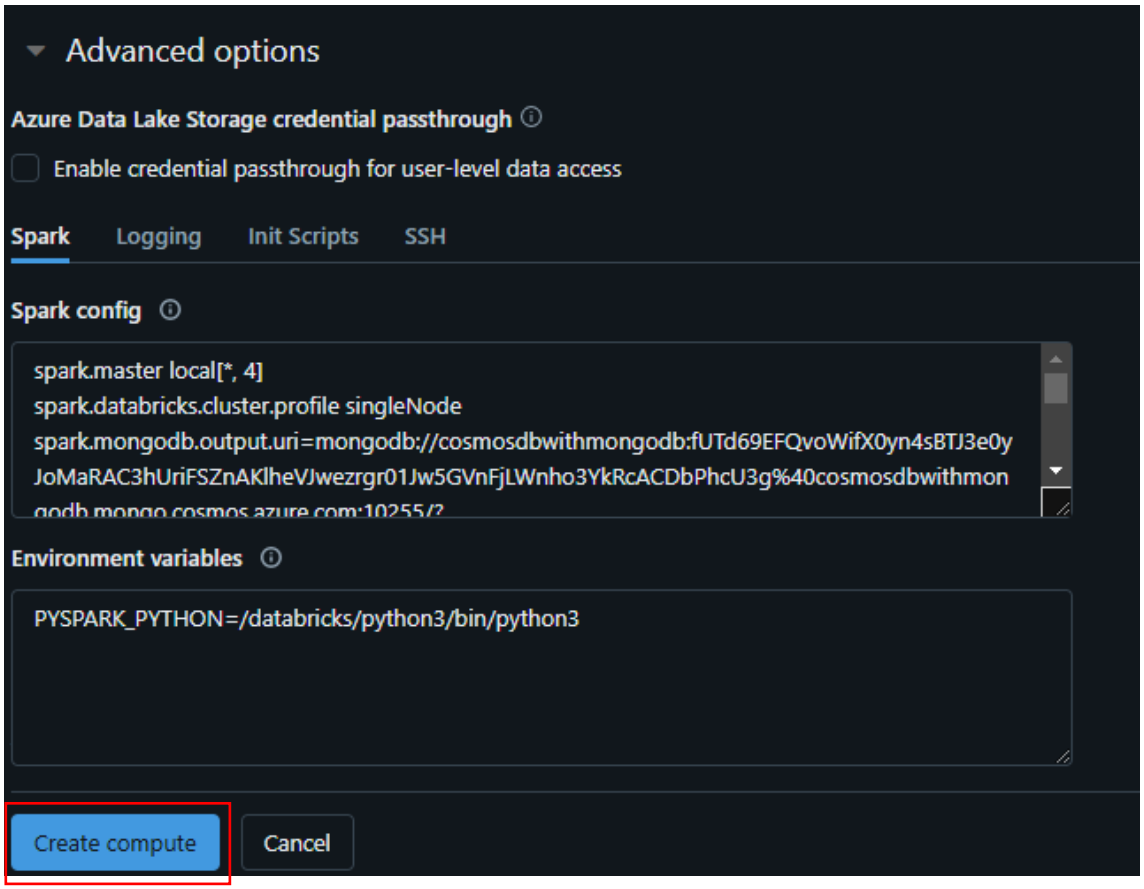
Importante, Single node

Click en Advanced options:

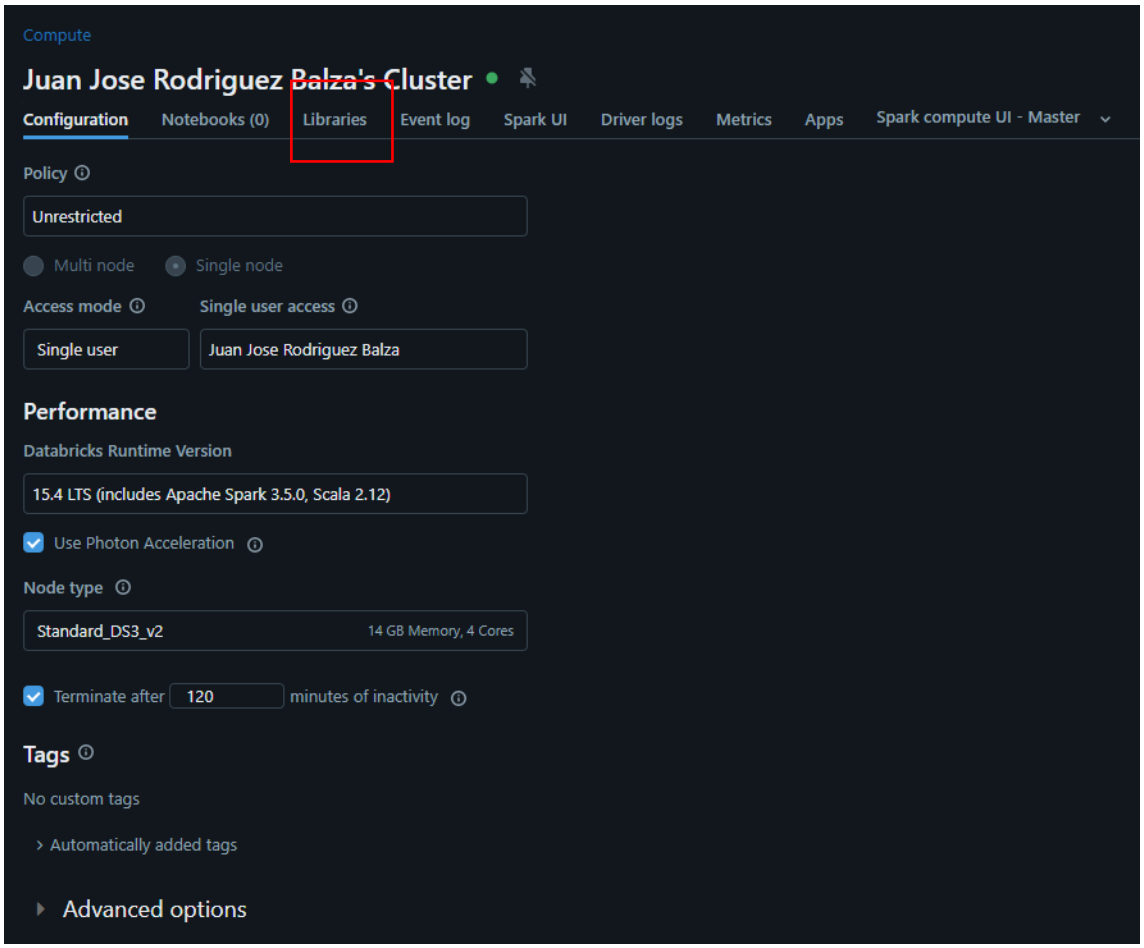
En Spark config escribimos:

spark.master local[*, 4]
spark.databricks.cluster.profile singleNode
spark.mongodb.output.uri=AQUITUCADENADECONEXIONUSADAENELSCRIPT
spark.mongodb.input.uri= AQUITUCADENADECONEXIONUSADAENELSCRIPT

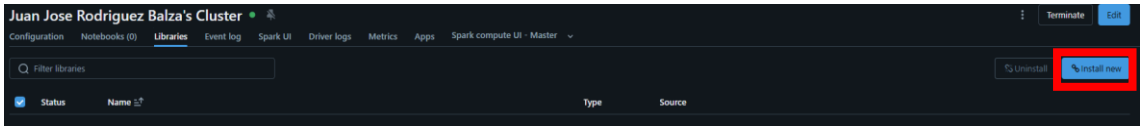
```
# Configuración de Cosmos DB
DB_NAME = "testingcosmosdb01"  # Nombre de tu base de datos
COLLECTION_NAME = "pruebadatabrickcosmosdb"  # Nombre de tu colección
CONNECTION = "mongodb://cosmosdbwithmongodb:fUTd69EFQvoWifX0yn4sBTJ3e0yJoMaRAC3hUriFSZnAKlheVJwezrgr01Jw5GVnFjLWnho3YkRcACDbPhcU3g==@cosmosdbwith
```



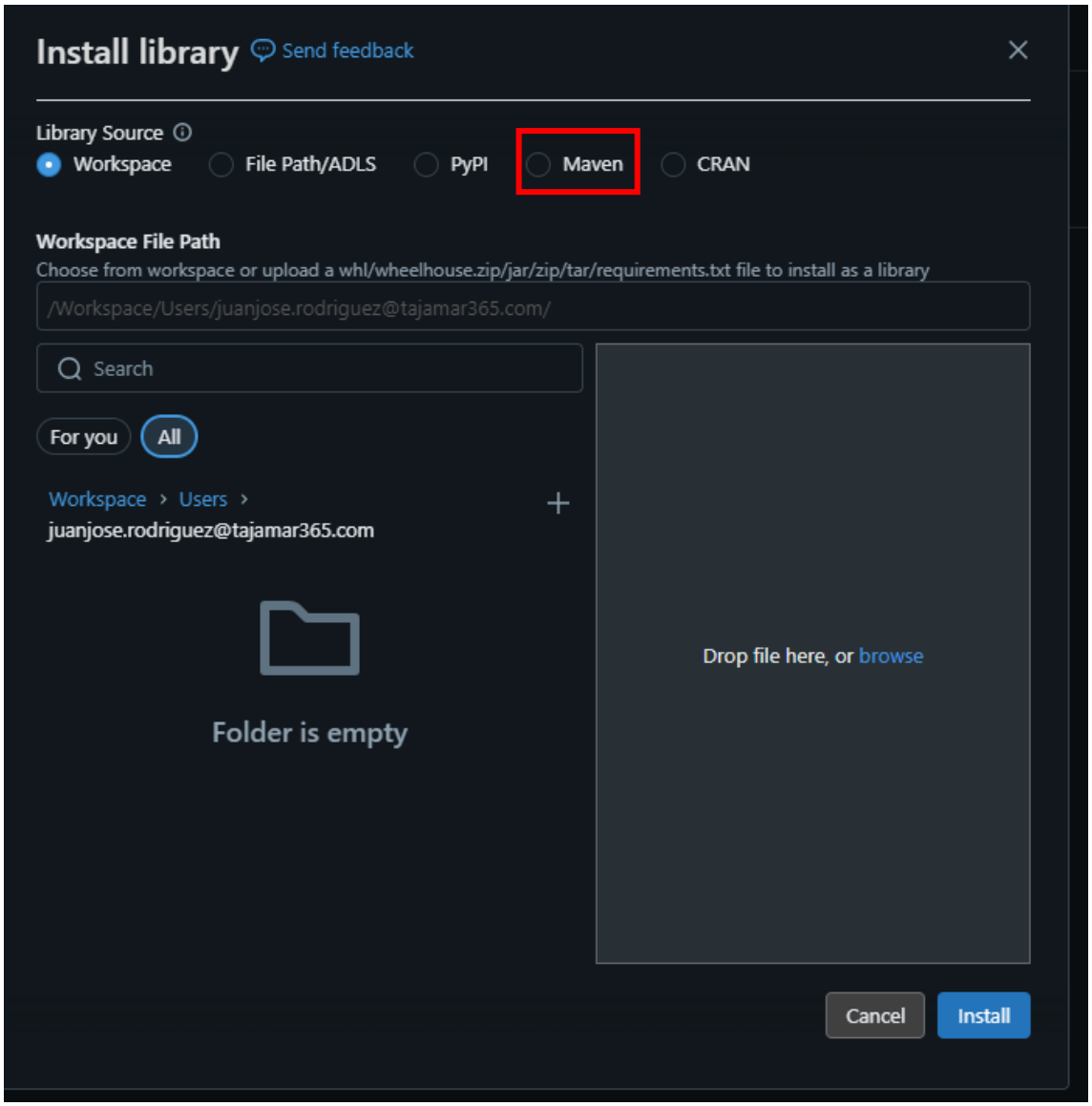
Una vez creado el cluster damos click en Libraries:



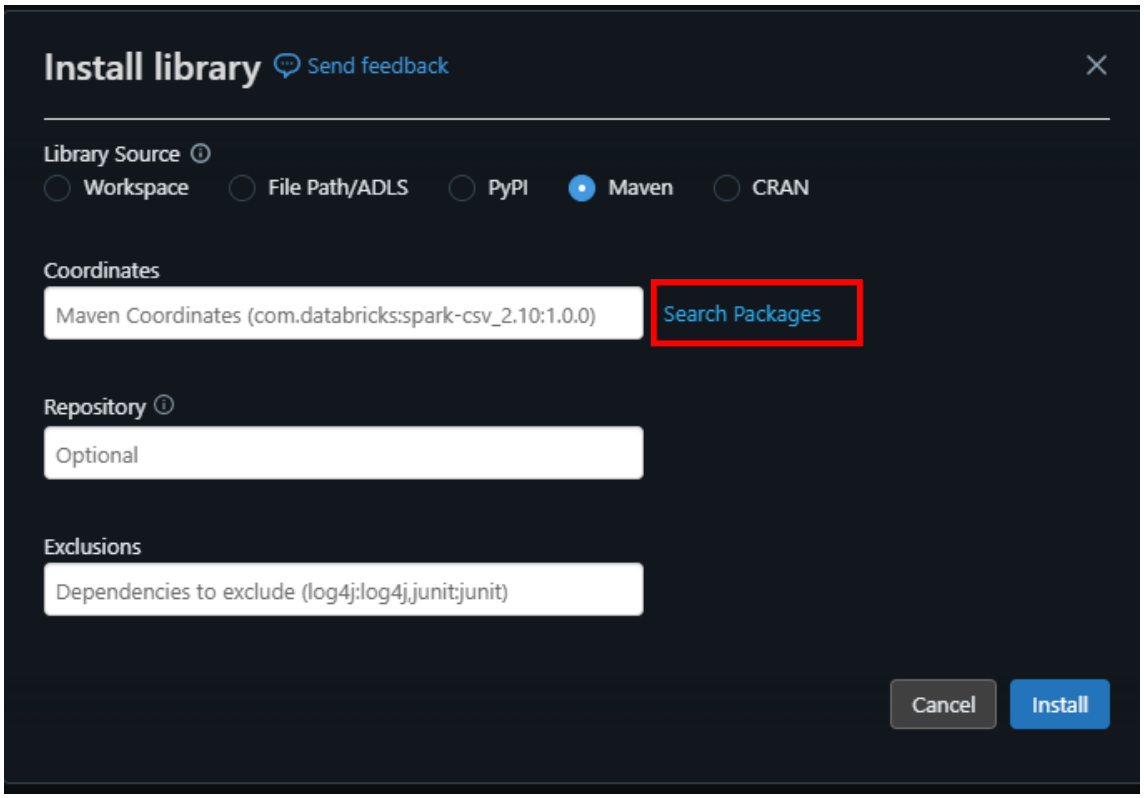
Click en install new:



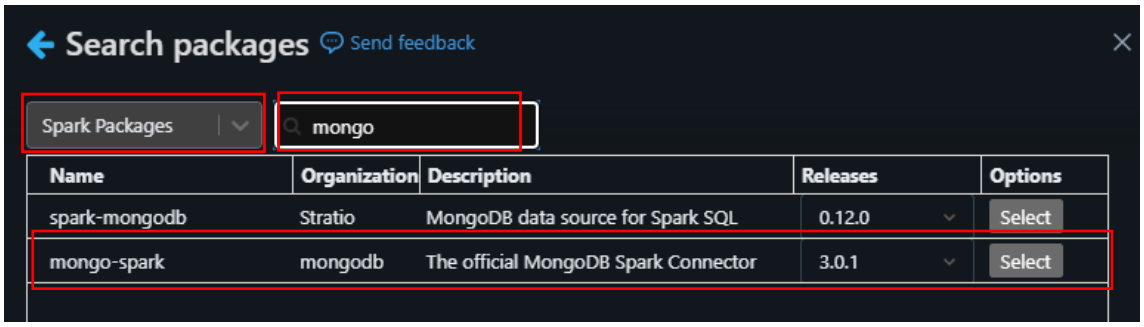
Click en Maven:



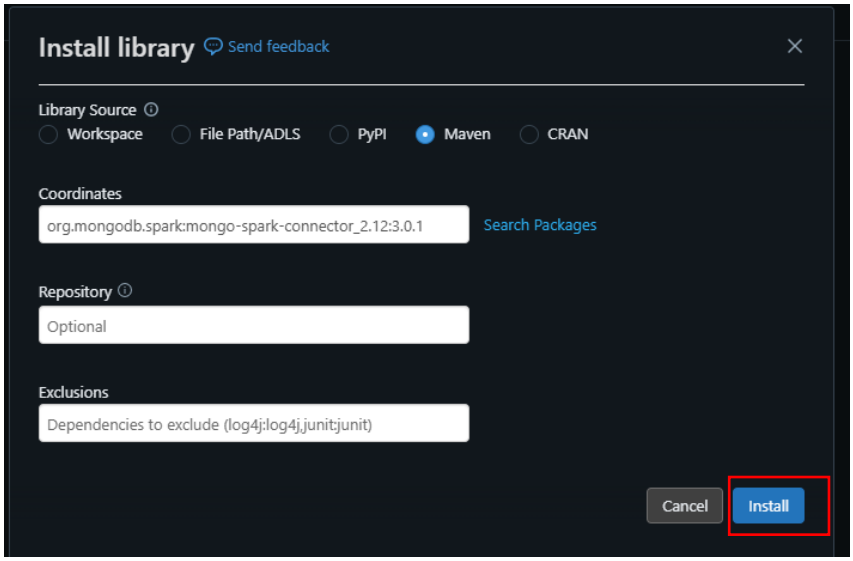
Click en Search Packages:



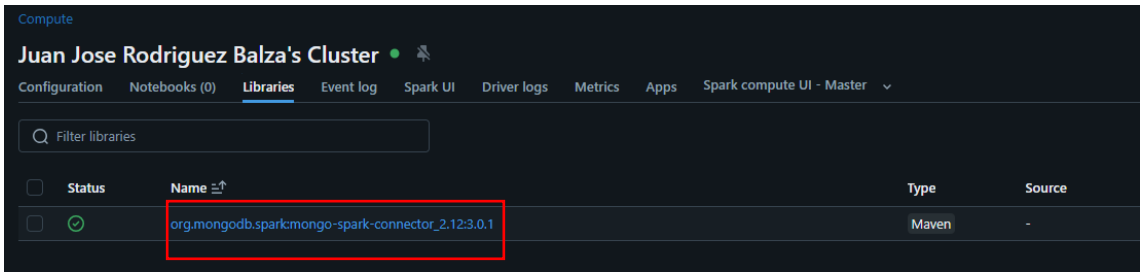
Spark Packages, escribir mongo y seleccionar el conector de spark oficial:



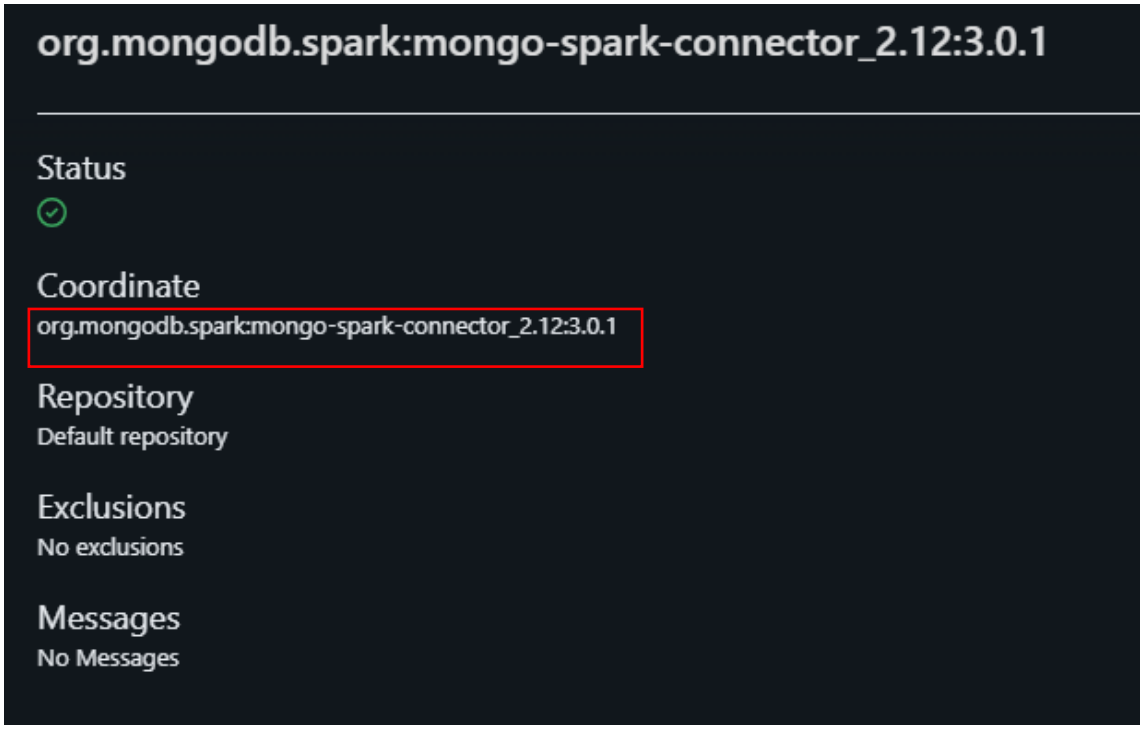
Click en install



Una vez instalado damos click en el conector:



Copiamos el Coordinate:



Abrimos un notebook y escribimos:

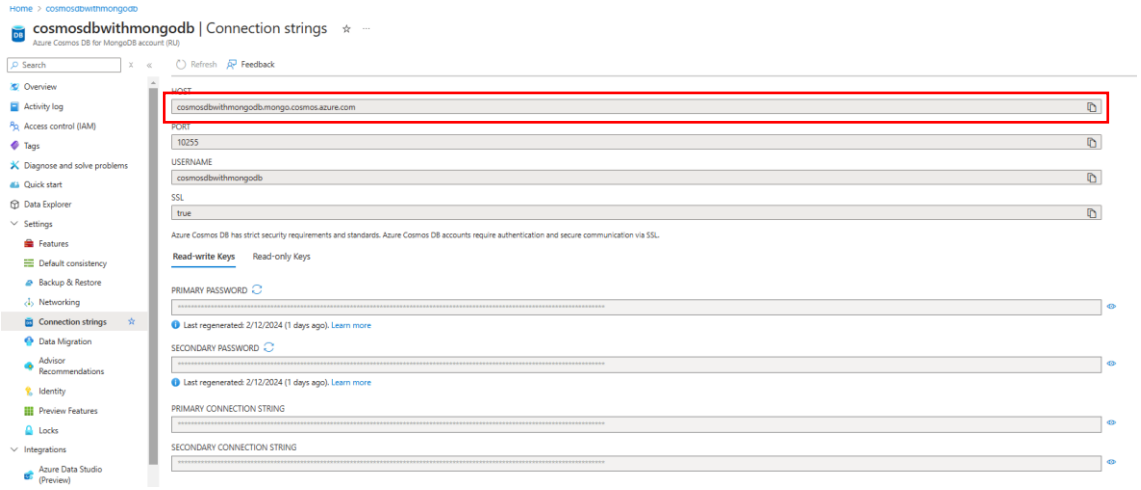
```
from pyspark.sql import SparkSession
databasename = 'testingcosmosdb01' # cambiar por el nombre de tu base de datos
collection = 'pruebadatabrickcosmosdb' # CAmbiar por el nombre de tu colleccion
```

```
uri = "mongodb://cosmosdbwithmongodb:fUTd69EFQvoWifX0yn4sBTJ3e0yJoMaRAC3hUriFSZnAKlheVJwezrgr01Jw5GVnFjLWnho3YkRcACDbPhcU3g==@cosmosdbwithmongodb.mongo.cosmos.azure.com:10255/<database>.<collection>?ssl=true&replicaSet=globaldb&retrywrites=false&maxIdleTimeMS=120000&appName=Databricks"
```

Cambia la uri del notebook por tu propia URI. Esta tiene una estructura como:

mongodb://<username>:<password>@**cosmosdbwithmongodb**.mongo.cosmos.azure.com:10255

Lo que va después del @ (que también es el USERNAME) lo encuentras haciendo click en connection string en tu base de datos de Cosmos DB:



Mientras que:

Usuario y contraseña serían:

- **USERNAME**: cosmosdbwithmongodb
- **PASSWORD**: Debes usar el valor del **Primary Password** visible en la misma sección.

Luego en otra celda escribes:

```
spark = SparkSession.builder.appName("TestingApp").config('spark.jars.packages', 'org.mongodb.spark:mongo-spark-connector_2.12:3.0.1').getOrCreate()
```

Pegar tu 'Coordinate'

Ahora creamos un dataframe:

```
01:12 AM (4s)

df = spark.read.format("com.mongodb.spark.sql.DefaultSource") \
    .option("uri", uri) \
    .option("database", databasename) \
    .option("collection", collection) \
    .load()

(1) Spark Jobs

df: pyspark.sql.dataframe.DataFrame = [_id: struct, city: string ... 6 more fields]
```

Luego df.show()

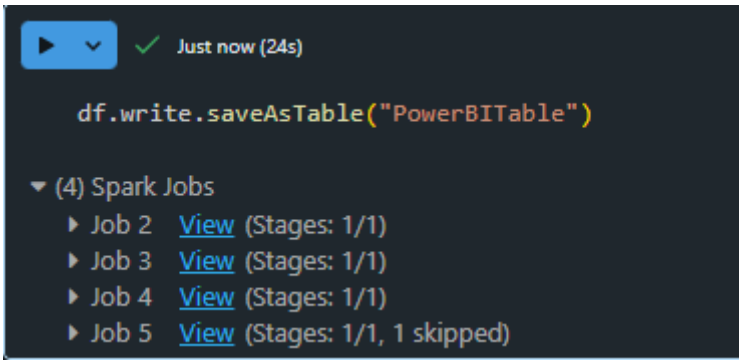
```
df.show()
```

(1) Spark Jobs

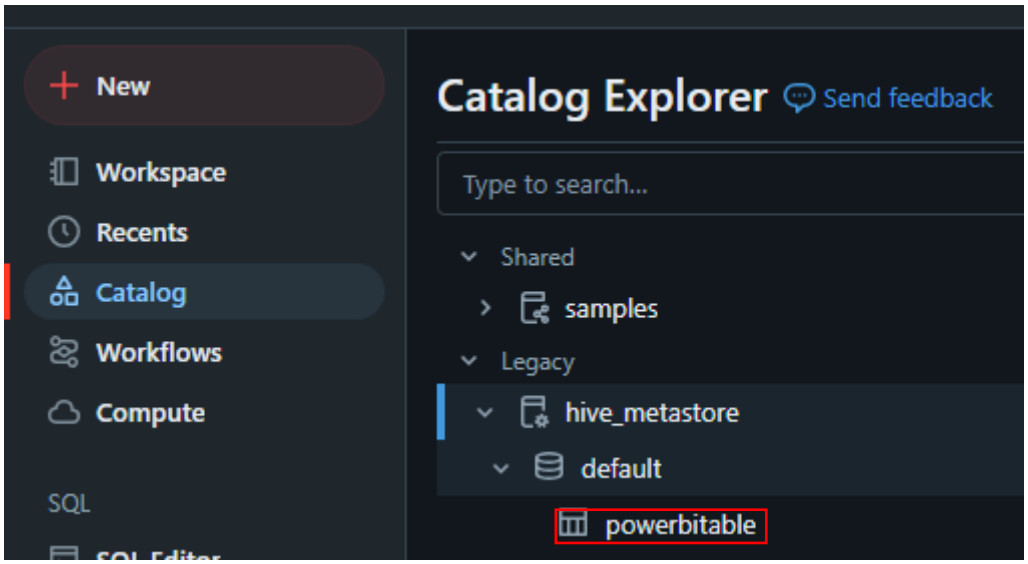
_id	city	humidity	pressure	temperature	timestamp	weather	wind_speed
{674e1c8d5f16e29a...	Toronto	62	1022	2.3	2024-12-02 21:46:05	overcast clouds	7.72
{674e1d6dd804377c...	Toronto	62	1022	2.3	2024-12-02 21:49:49	overcast clouds	7.72
{674e1d83d804377c...	Toronto	62	1022	2.3	2024-12-02 21:50:11	overcast clouds	7.72
{674e1d97d804377c...	Toronto	62	1022	2.19	2024-12-02 21:50:31	overcast clouds	7.72
{674e1dadd804377c...	Toronto	62	1022	2.19	2024-12-02 21:50:53	overcast clouds	7.72
{674e1dc2d804377c...	Toronto	62	1022	2.19	2024-12-02 21:51:14	overcast clouds	7.72
{674e1dd7d804377c...	Toronto	62	1022	2.19	2024-12-02 21:51:35	overcast clouds	7.72
{674e1decd804377c...	Toronto	62	1022	2.19	2024-12-02 21:51:56	overcast clouds	7.72
{674e1e01d804377c...	Toronto	62	1022	2.19	2024-12-02 21:52:17	overcast clouds	7.72
{674e1e16d804377c...	Toronto	62	1022	2.19	2024-12-02 21:52:38	overcast clouds	7.72
{674e1e2bd804377c...	Toronto	62	1022	2.19	2024-12-02 21:52:59	overcast clouds	7.72
{674e1e40d804377c...	Toronto	62	1022	2.19	2024-12-02 21:53:20	overcast clouds	7.72
{674e1e56d804377c...	Toronto	62	1022	2.19	2024-12-02 21:53:42	overcast clouds	7.72
{674e1e6ad804377c...	Toronto	62	1022	2.3	2024-12-02 21:54:02	overcast clouds	7.72
{674e1e7fd804377c...	Toronto	62	1022	2.3	2024-12-02 21:54:23	overcast clouds	7.72
{674e1e94d804377c...	Toronto	62	1022	2.3	2024-12-02 21:54:44	overcast clouds	7.72
{674e1ea9d804377c...	Toronto	62	1022	2.3	2024-12-02 21:55:05	overcast clouds	7.72
{674e1ebed804377c...	Toronto	62	1022	2.19	2024-12-02 21:55:26	overcast clouds	7.72

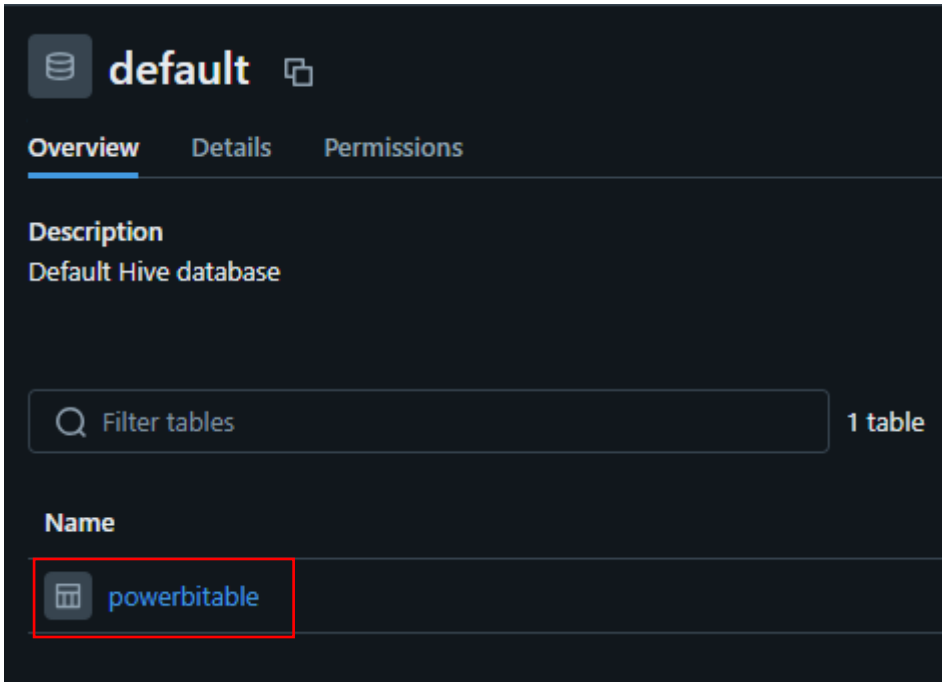
Vamos a usar un Partner Connect de Databricks para poder utilizar Power BI:

En Databricks, vamos ahora a Partner Connect. Para ello vamos a almacenar el dataframe de arriba en un 'Hive Metastore' :

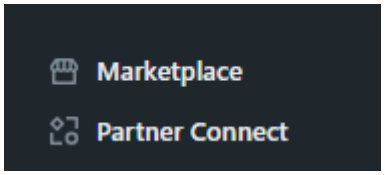


Si damos click en Catalog encontramos la tabla:

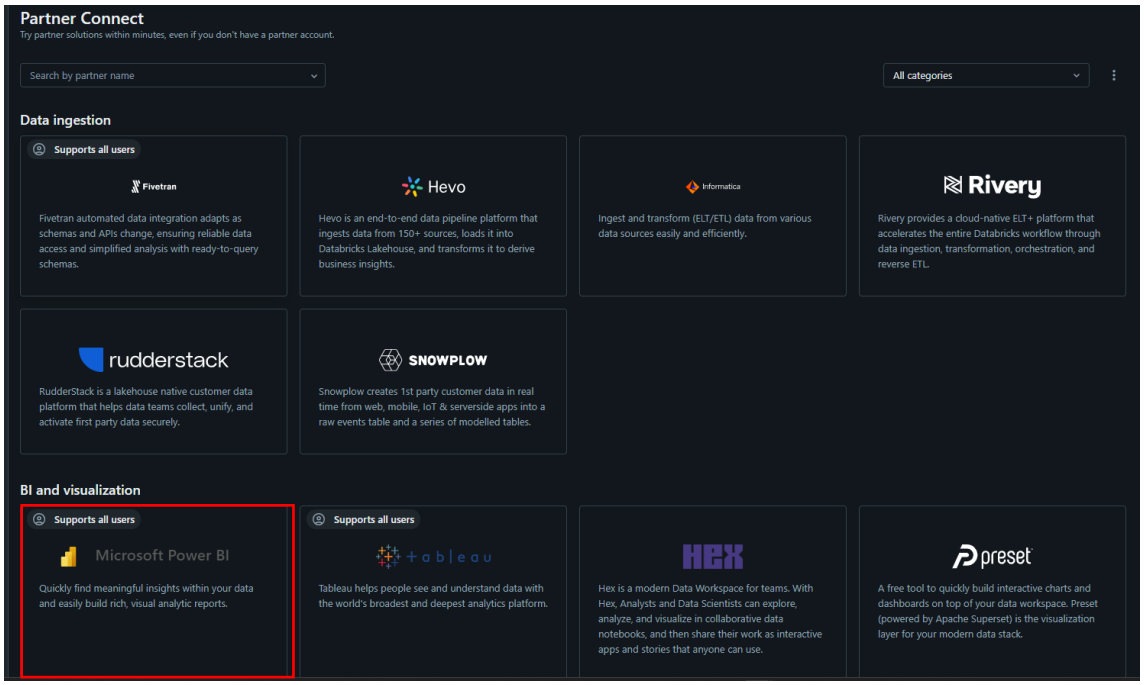




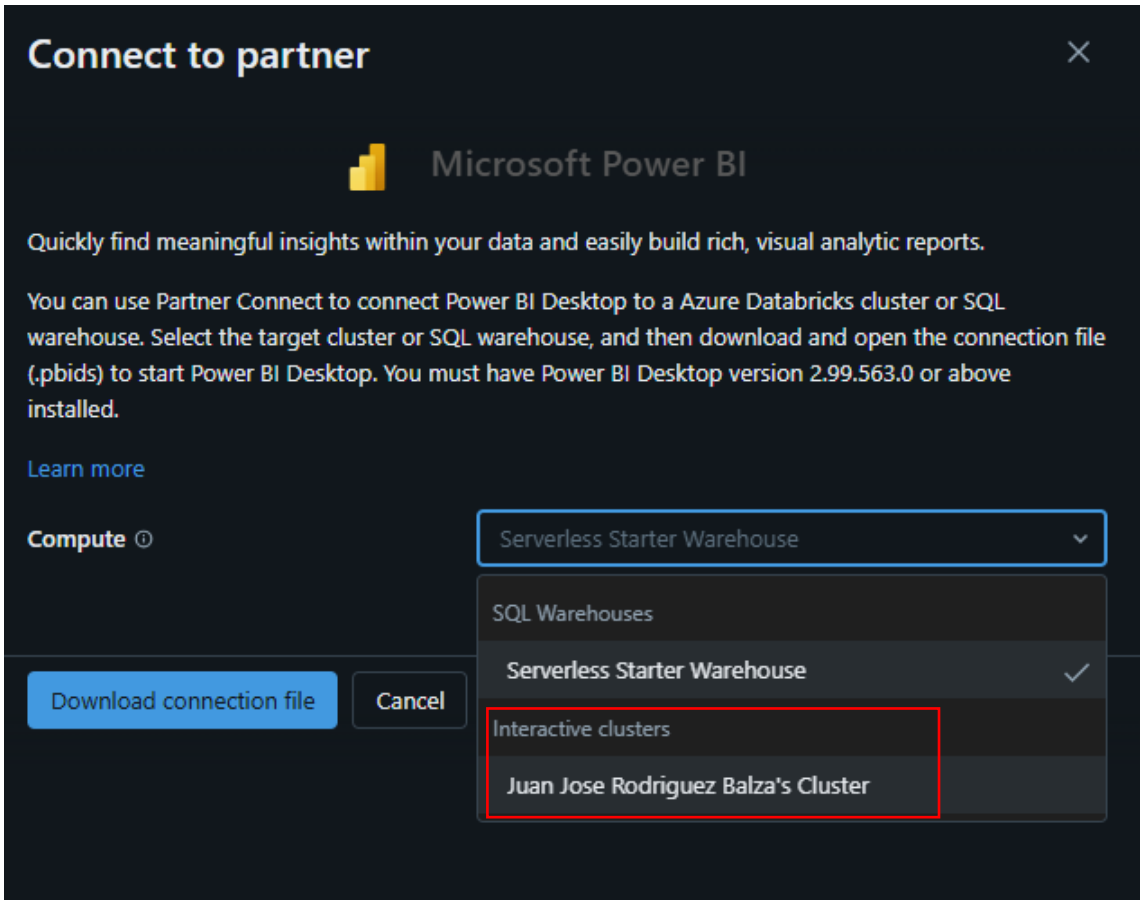
Ahora nos vamos a Partner Connect:



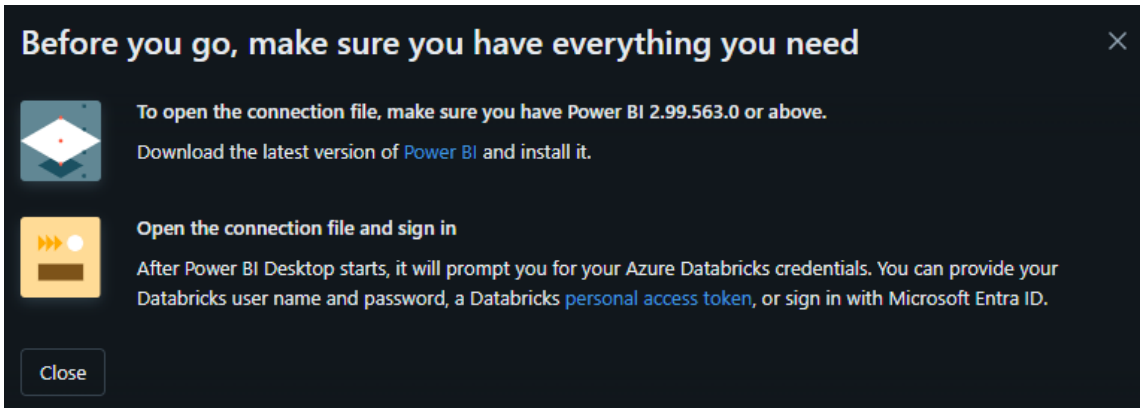
Seleccionamos Power BI:



Seleccionamos el cluster:

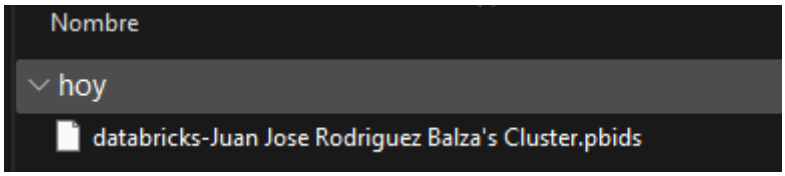


Click en Download connection file



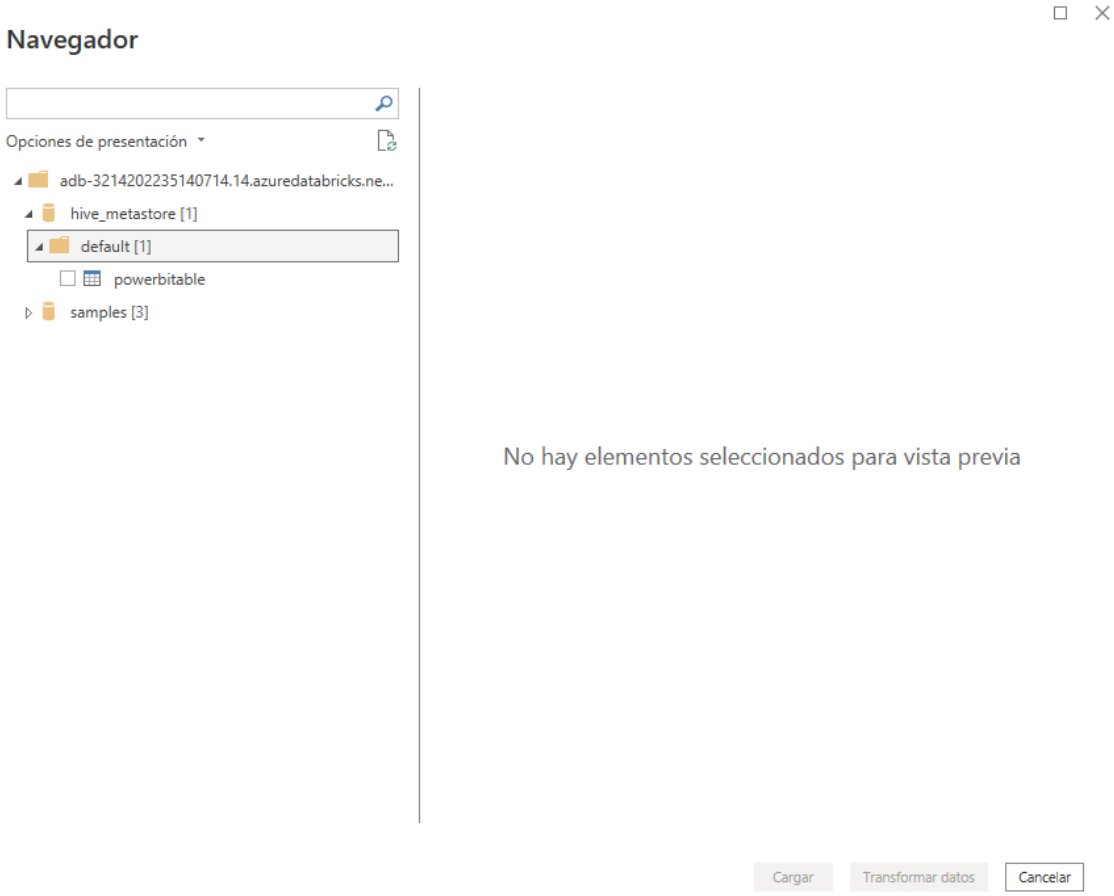
Verifica que tienes Power BI instalado en el equipo con la versión que se especifica arriba.

Una vez que has verificado la instalación de Power BI ve a tu carpeta de descargas y buscas el 'archivo de conexión' (conection file) que descargaste desde el Partner connect.

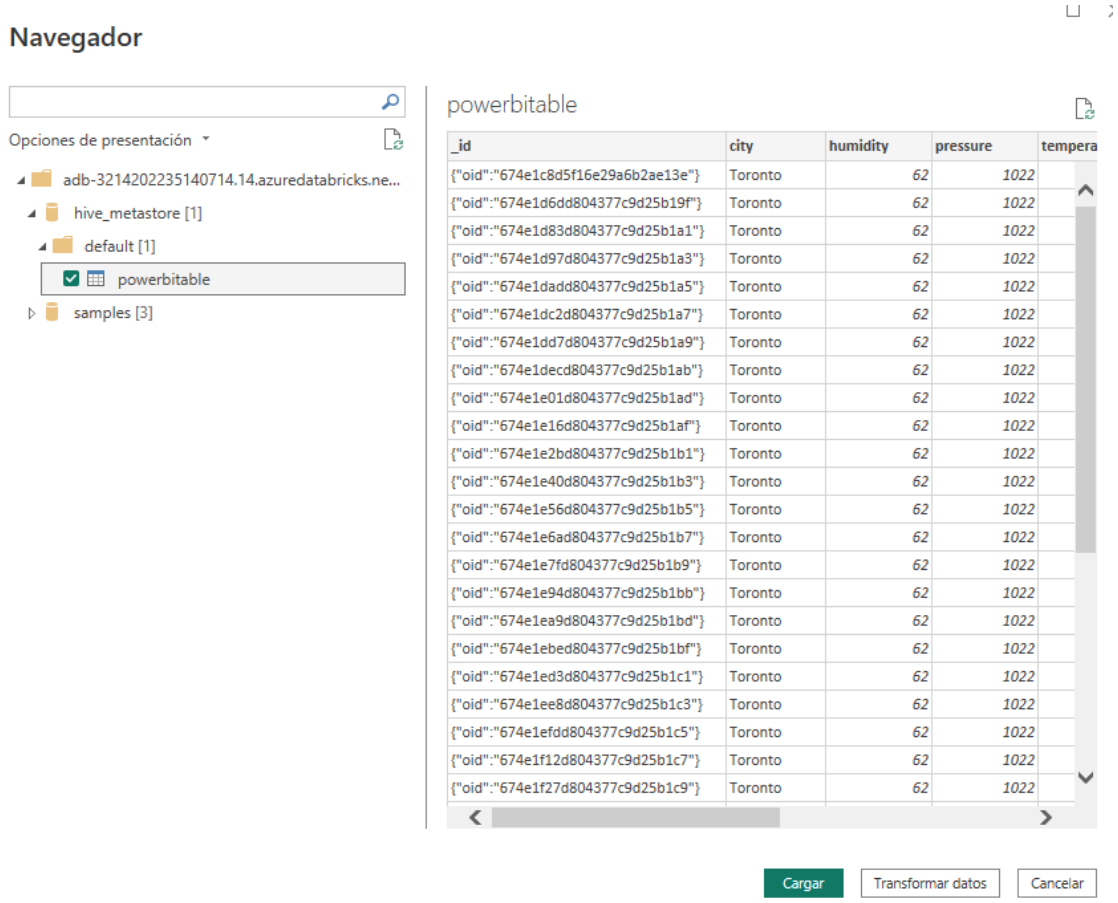


El archivo tiene extensión .pbids.

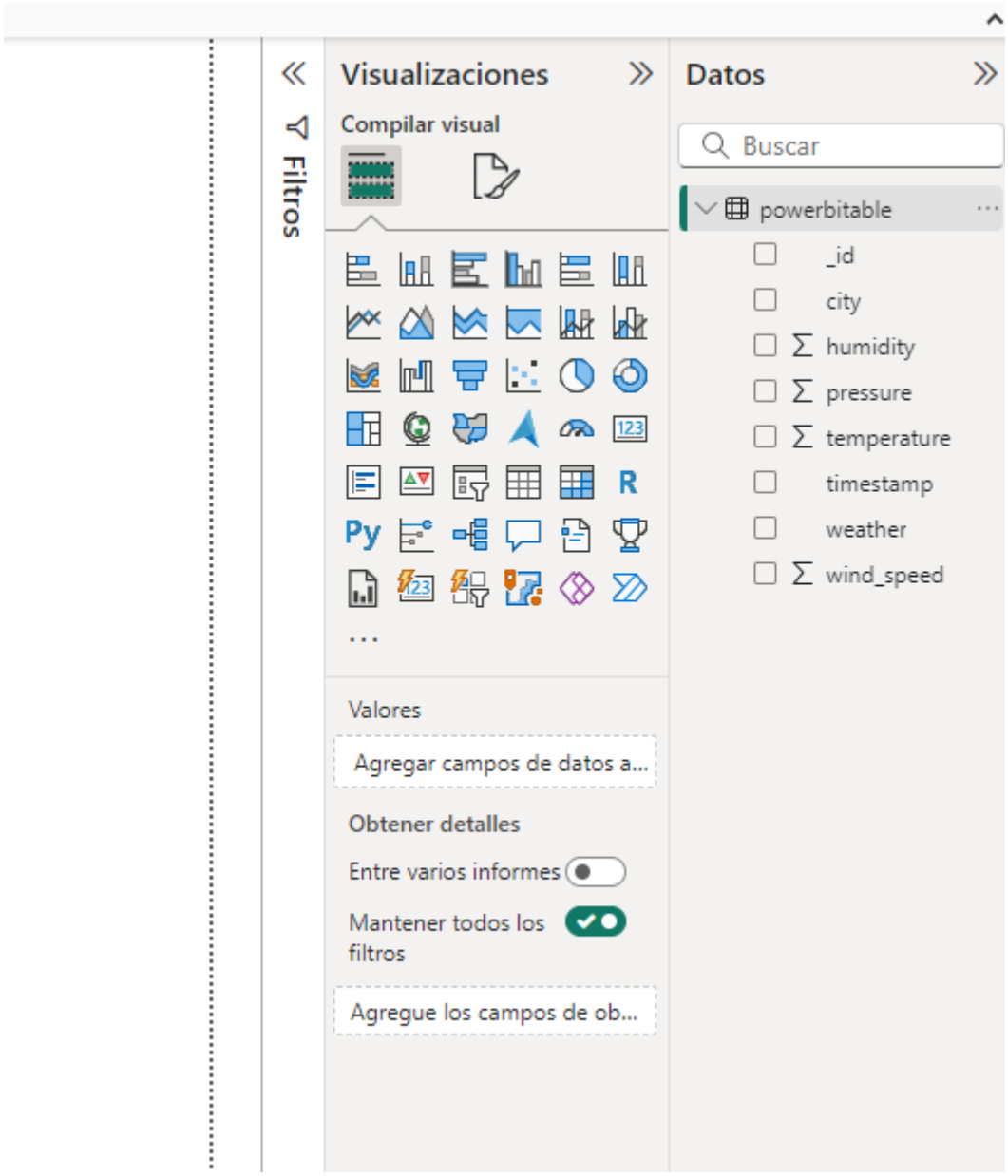
Abrir el archivo con Power BI Desktop. Power BI te pedirá clave y contraseña, selecciona Azure Active Directory e inicias sesión con el correo que tienes registrado en Azure, luego click en Conectar:



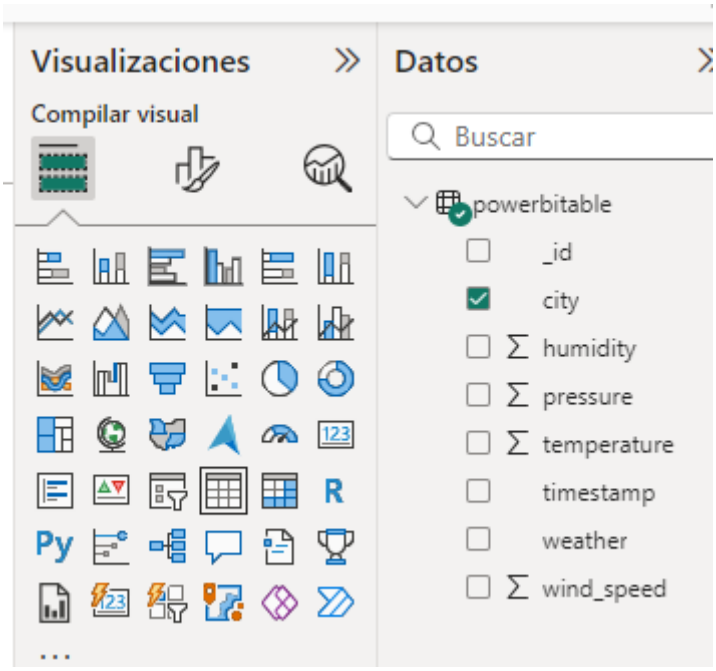
Seleccionamos powerbitable, luego Cargar:

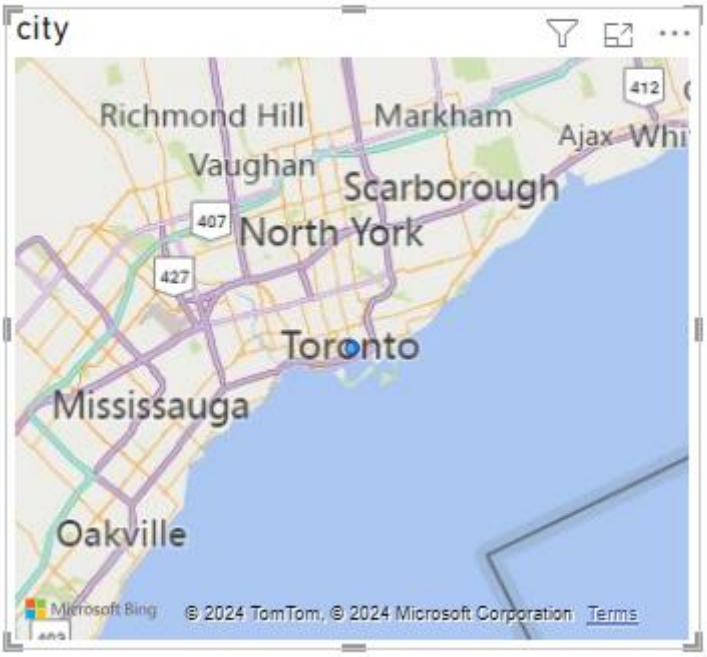


En el panel principal ya se puede ver la tabla:



Damos click en City y luego click en 


Visualizaciones
Compilar visual
Datos
Buscar
powerbitable
_id
city
humidity
pressure
temperature
timestamp
weather
wind_speed


city
Richmond Hill
Markham
Ajax
Vaughan
Scarborough
North York
Toronto
Mississauga
Oakville
412
407
427
Microsoft Bing
© 2024 TomTom, © 2024 Microsoft Corporation
Terms

Dedica un buen tiempo ha interaccionar con Power BI haciendo varios plots.

Una vez finalizado eliminar todos los recursos que has creado para evitar cargos adicionales.