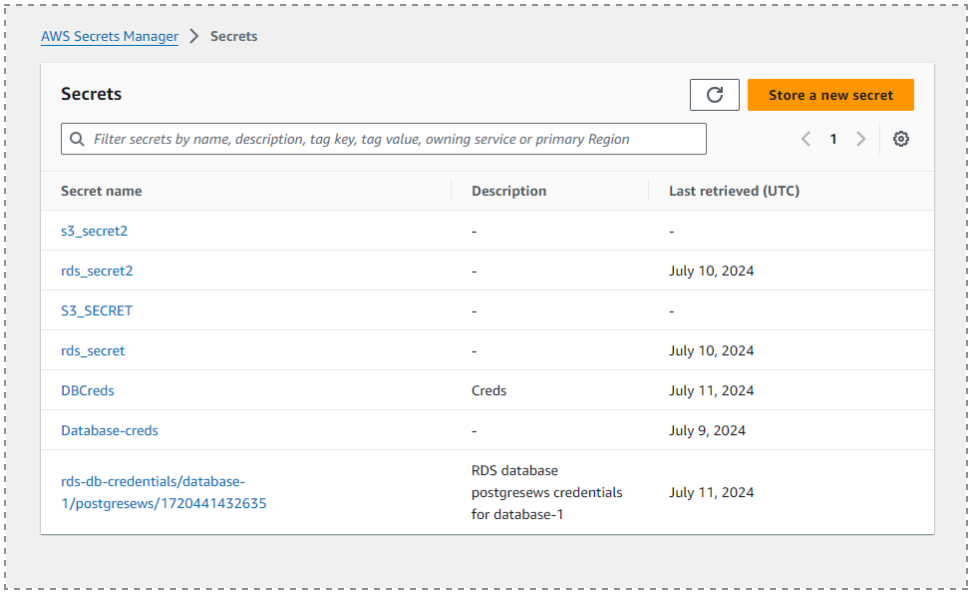
AWS Secrets - Read in Credentials Securely

Go to AWS and go to Secret Manager

1. Click **Store a new secret**



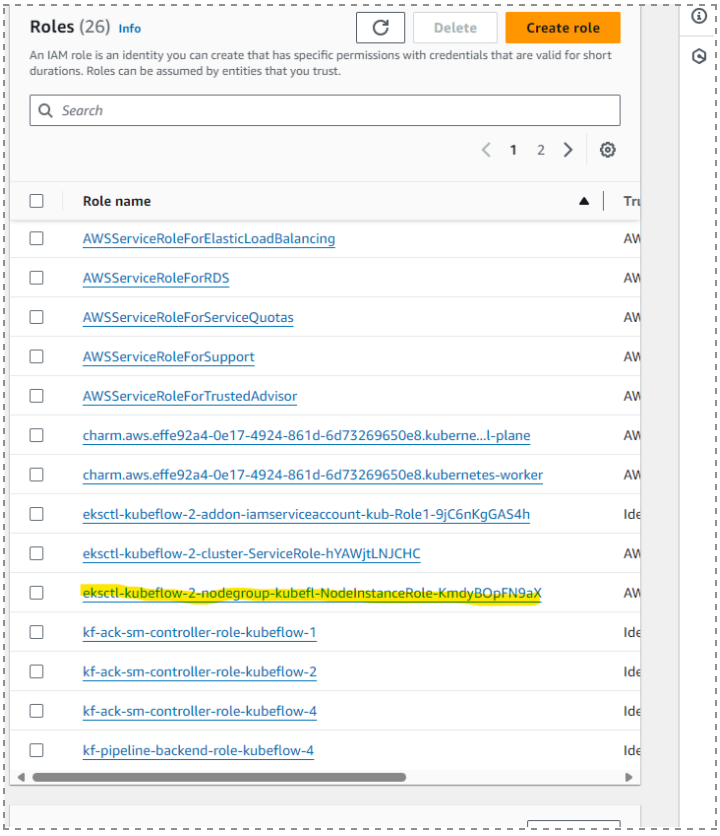
2. Choose type of secret and enter in values. For generic postgresql/s3 bucket choose **Other type of secret** and add a key/value for username and password. If using an RDS database, you can store an RDS secret.

Here is code to use for reading in the secret:

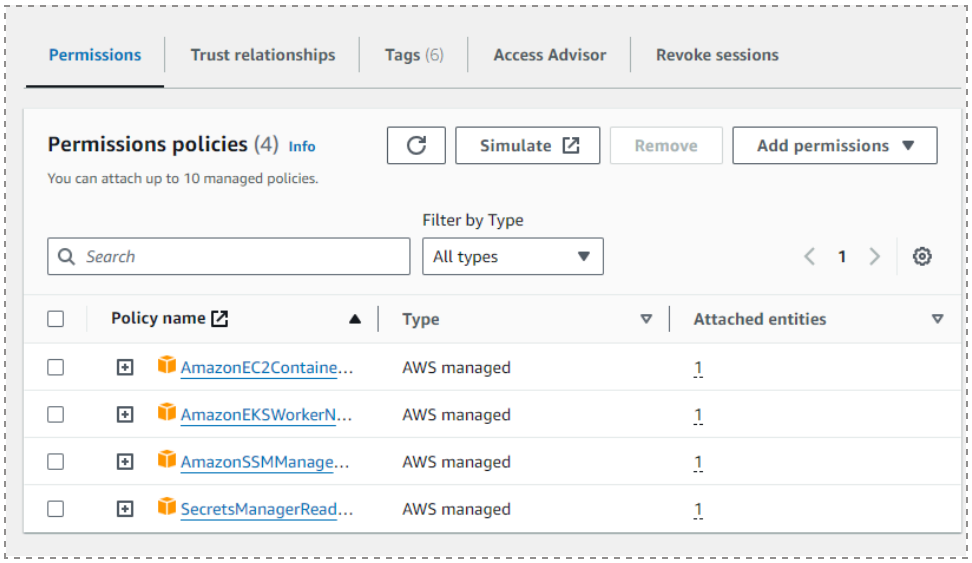
```
import boto3 from botocore.exceptions
import ClientError
import json
def get_secret():
   secret_name = "DBCreds"
   region_name = "us-east-1" # Create a Secrets Manager client
   session = boto3.session.Session()
   client = session.client(
      service_name='secretsmanager',
```

```
    region_name=region_name
)
try:
    get_secret_value_response = client.get_secret_value(
    SecretId=secret_name )
except ClientError as e: # For a list of exceptions thrown, see #
https://docs.aws.amazon.com/secretsmanager/latest/apireference/API
_GetSecretValue.html

raise e secret = get_secret_value_response['SecretString'] # Parse
the secret string to get the credentials secret_dict =
json.loads(secret)
username = secret_dict['username']
password = secret_dict['password']
host = secret_dict['host'] port = secret_dict['port']
dbname = secret_dict['dbname']

return username, password, host, port, dbname
```

If you get an error, go to Roles in the IAM dashboard, select the NodeInstance Role of the cluster
(Roles should already be added for both of the clusters, but in case something got changed)



Check the permissions policies that the SecretManagerReadWrite permissions policy was added.



If not, select add permissions, and then attach policies
Search SecretManagerReadWrite and add it