

# CLOUD COMPUTING SYSTEMS

## Lab 4

Nuno Preguiça, João Resende

([nuno.preguica\\_at\\_fct.unl.pt](mailto:nuno.preguica_at_fct.unl.pt), [jresende\\_at\\_fct.unl.pt](mailto:jresende_at_fct.unl.pt))

# GOAL

In the end of this lab you should be able to:

- Create a Azure Redis for Cache account @ Azure;
- Use AzureRedisCache to store cache data.
- More professional deployment.

# GOAL

In the end of this lab you should be able to:

- **Create a Azure Redis for Cache account @ Azure;**
- Use AzureRedisCache to store cache data.
- More professional deployment.

# CREATE AZURE REDIS FOR CACHE

Microsoft Azure

redis

Azure services

Create a resource

More services

Recent resource

Name

Subscription

22 hours ago

a week ago

a week ago

a week ago

a week ago

12 months ago

Services

App registrations

Azure Cache for Redis

On-premises data gateways

SQL Server registries

Event Grid Partner Registrations

Container registries

HDInsight clusters

Resources

Marketplace

Azure Cache for Redis Enterprise & Flash

RedisEdge

Azure Cache for Redis

Redis

Documentation

Azure Cache for Redis Documentation | Microsoft Docs

What is Azure Cache for Redis? | Microsoft Docs

Use redis-cli with Azure Cache for Redis | Microsoft Docs

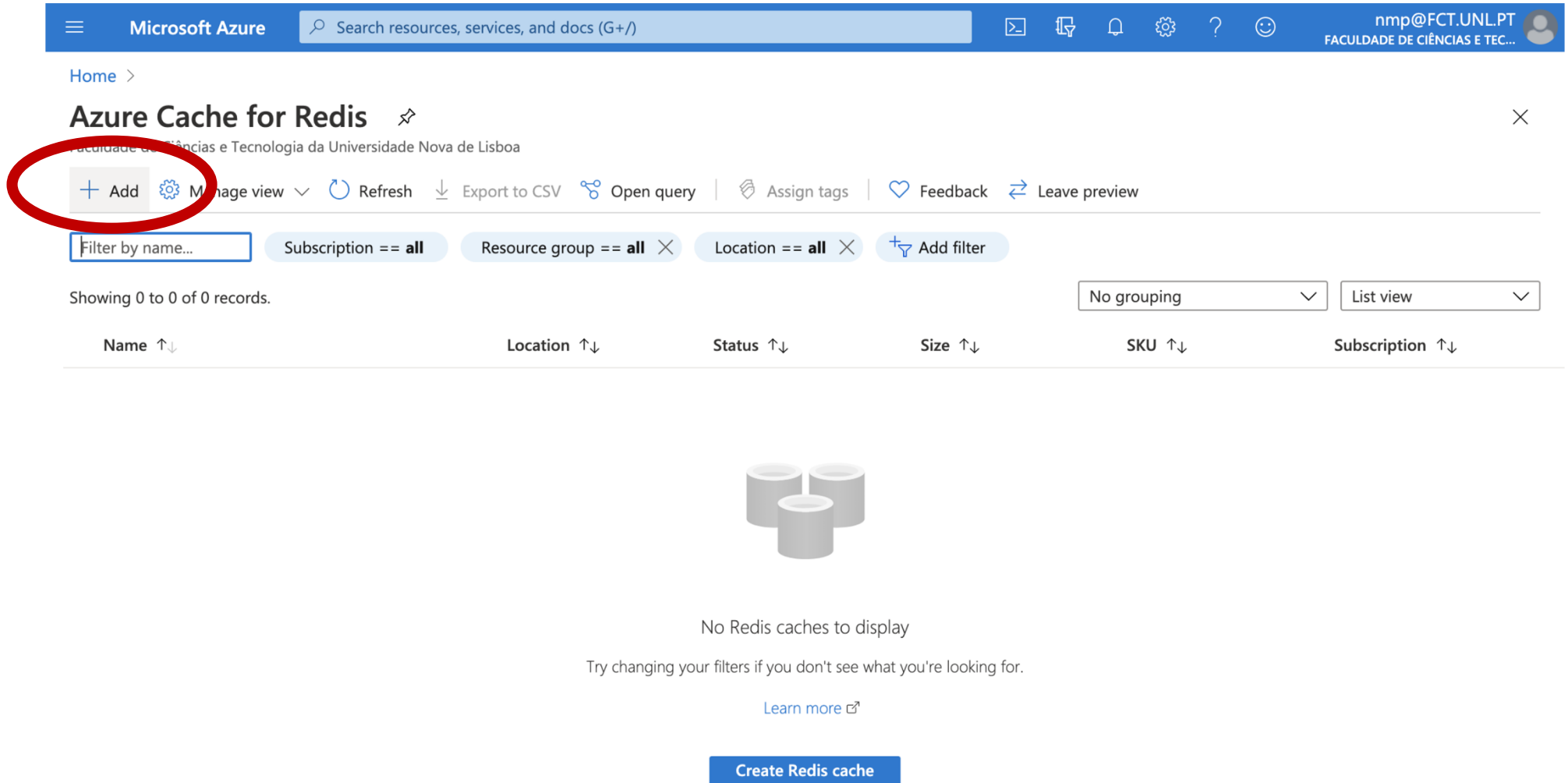
Quickstart: Use Azure Cache for Redis in .NET Framework ...

Resource Groups

Give feedback

Name	Subscription	22 hours ago
Azure para Estudantes	Subscription	22 hours ago
scc21224204	Azure Cosmos DB account	a week ago
scc-backend-rg-4204	Resource group	a week ago
scc21224204	Storage account	a week ago
scc2021-rg-westeuropa-4204	Resource group	a week ago
scc2021-4204	Resource group	12 months ago

# CREATE AZURE REDIS FOR CACHE (2)



The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with the Microsoft Azure logo, a search bar, and user information (nmp@FCT.UNL.PT). Below the navigation bar, the page title is 'Azure Cache for Redis'. A red circle highlights the '+ Add' button in the top left corner of the resource list. To the right of the '+ Add' button are links for 'Manage view', 'Refresh', 'Export to CSV', 'Open query', 'Assign tags', 'Feedback', and 'Leave preview'. Below these links is a filter bar with a search box 'Filter by name...' and several filter buttons: 'Subscription == all', 'Resource group == all', 'Location == all', and 'Add filter'. The table below the filter bar shows 'Showing 0 to 0 of 0 records.' and has columns for 'Name', 'Location', 'Status', 'Size', 'SKU', and 'Subscription'. Below the table, there's a message 'No Redis caches to display' with a sub-message 'Try changing your filters if you don't see what you're looking for.' and a link 'Learn more'. At the bottom, there's a blue button labeled 'Create Redis cache'.

Microsoft Azure

Search resources, services, and docs (G+)

Home >

## Azure Cache for Redis

Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa

+ Add Manage view Refresh Export to CSV Open query Assign tags Feedback Leave preview

Filter by name... Subscription == all Resource group == all Location == all Add filter

Showing 0 to 0 of 0 records.

No grouping List view

Name	Location	Status	Size	SKU	Subscription
------	----------	--------	------	-----	--------------

No Redis caches to display

Try changing your filters if you don't see what you're looking for.

[Learn more](#)

Create Redis cache

# CREATE AZURE REDIS FOR CACHE (3)

Microsoft Azure

nmp@FCT.UNL.PT  
FACULDADE DE CIÊNCIAS E TEC...

Home > Create a resource > Azure Cache for Redis >

## New Redis Cache ...

Basics Networking Advanced Tags Review + create

Azure Cache for Redis helps your application stay responsive even as user load increases. It does so by leveraging the low latency, high-throughput capabilities of the Redis engine. [Learn more](#)

### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \*

Azure para Estudantes

Resource group \*

scc-backend-rg-4204

[Create new](#)

### Instance Details

DNS name \*

scc2122cache4204

.redis.cache.windows.net

Location \*

West Europe

Cache type ([View full pricing details](#)) \*

Basic C0 (250 MB Cache, No SLA)

Review + create

< Previous

Next : Networking >

Check features and pricing at:  
<https://azure.microsoft.com/pt-pt/pricing/details/cache/>

# CREATE AZURE REDIS FOR CACHE (4)

Microsoft Azure

Search resources, services, and docs (G+)

nmp@FCT.UNL.PT  
FACULDADE DE CIÊNCIAS E TEC...

[Home](#) > [Create a resource](#) > [Azure Cache for Redis](#) >

## New Redis Cache ...

Basics

Networking

Advanced

Tags

Review + create

Non-TLS port

☐ Enable

Redis version

☐ 4

☒ 6

Review + create

< Previous

Next : Tags >

# CREATE AZURE REDIS FOR CACHE (5)

The screenshot displays the Microsoft Azure portal interface. At the top, the header includes the 'Microsoft Azure' logo, a search bar, and navigation icons. The user's email 'nmp@FCT.UN' and affiliation 'FACULDADE DE CIÊNCIAS E T' are visible in the top right. The main content area shows the 'Overview' page for a deployment named 'CreateRedis-scc2122cache4204-20211024182614'. A green checkmark icon indicates that the deployment is complete. Below this, deployment details are listed: 'Deployment name: CreateRedis-scc2122cache4204-2021102418...', 'Subscription: Azure para Estudantes', 'Resource group: scc-backend-rg-4204', 'Start time: 10/24/2021, 6:26:15 PM', and 'Correlation ID: 464f3473-2c04-4514-a389-b3b3ce55...'. A 'Deployment details (Download)' link is available. Under the 'Next steps' section, there is a 'Go to resource' button. The left sidebar contains navigation links for 'Overview', 'Inputs', 'Outputs', and 'Template'.

Microsoft Azure Search resources, services, and docs (G+)

Home >

**CreateRedis-scc2122cache4204-20211024182614** | Overview

Deployment

Search (Cmd+/) << Delete Cancel Redeploy Refresh

Overview

Inputs

Outputs

Template

✓ Your deployment is complete

Deployment name: CreateRedis-scc2122cache4204-2021102418... Start time: 10/24/2021, 6:26:15 PM  
Subscription: [Azure para Estudantes](#) Correlation ID: 464f3473-2c04-4514-a389-b3b3ce55...  
Resource group: [scc-backend-rg-4204](#)

Deployment details ([Download](#))

Next steps

[Go to resource](#)



# CREATE AZURE REDIS FOR CACHE: URL

The screenshot displays the Azure portal interface for an Azure Cache for Redis resource. The top navigation bar includes the Microsoft Azure logo, a search bar, and user information (nmp@FCT.UNL.PT). The breadcrumb trail shows the path: Home > CreateRedis-scc2122cache4204-20211024182614 > scc2122cache4204. The resource name 'scc2122cache4204' is prominently displayed, along with the icon for Azure Cache for Redis. A search bar (Cmd+/) is available for navigating within the resource's settings.

The left sidebar lists various management options: Overview (selected), Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events, Settings, Access keys, Advanced settings, Scale, Cluster size, Data persistence, Schedule updates, Geo-replication, and Virtual Network.

The main content area shows the 'Essentials' section, which includes a warning about TLS 1.0 and 1.1 support being retired. Below this, key resource details are listed: Resource group (scc-backend-rg-4204), Status (Running - Basic 250 MB), Location (West Europe), Subscription (Azure para Estudantes), and Subscription ID (83abecdf-8b40-49a0-bcae-b5fba4011353). A red circle highlights the 'Host name' field, which contains the URL: `scc2122cache4204.redis.cache.windows.net`.

Additional settings visible include 'Non-SSL port (6379) disabled' and links for 'Show access keys...', '\*Best practices\*' (https://aka.ms/redis/p/bestpractices), and '\*New features\*' (https://aka.ms/newfeatures). A 'JSON View' link is also present.

The 'Memory Usage' section features a horizontal bar chart showing the current memory usage level relative to the 350kB limit. The usage is currently at approximately 300kB.

# CREATE AZURE REDIS FOR CACHE: KEY

The screenshot shows the Microsoft Azure portal interface. At the top, there's a blue header with the 'Microsoft Azure' logo and a search bar. Below the header, the breadcrumb trail reads 'Home > CreateRedis-scc2122cache4204-20211024182614 > scc2122cache4204'. The main heading is 'scc2122cache4204 | Access keys' with a key icon and a three-dot menu. Below this, there's a search bar and two buttons: 'Regenerate Primary' and 'Regenerate Secondary'. The left sidebar contains navigation links: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events, Settings (highlighted), Access keys (selected), Advanced settings, Scale, Cluster size, Data persistence, Schedule updates, Geo-replication, and Virtual Network. The main content area shows the 'Access keys' details. It lists the 'Primary' and 'Secondary' access keys, each with a copy icon. The primary access key is highlighted with a red circle. Below the keys, there are sections for 'Primary connection string (StackExchange.Redis)' and 'Secondary connection string (StackExchange.Redis)', both with copy icons. At the bottom, there's a link to 'https://aka.ms/redisclients' for more information on other clients.

Microsoft Azure

Search resources, services, and docs (G+)

Home > CreateRedis-scc2122cache4204-20211024182614 > scc2122cache4204

**scc2122cache4204** | Access keys

Azure Cache for Redis

Search (Cmd+ /)

Regenerate Primary Regenerate Secondary

Primary

Secondary

Primary connection string (StackExchange.Redis)

Secondary connection string (StackExchange.Redis)

For information on other clients see: <https://aka.ms/redisclients>

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Events

Settings

Access keys

Advanced settings

Scale

Cluster size

Data persistence

Schedule updates

Geo-replication

Virtual Network

# GOAL

In the end of this lab you should be able to:

- Create a Azure Redis for Cache account @ Azure;
- **Use AzureRedisCache to store cache data.**
- More professional deployment.

# ACCESSING AZURE CACHE FOR REDIS: USEFUL LINKS

We will use the library provided by Redis.

Java Docs available at:

<https://www.javadoc.io/doc/redis.clients/jedis/4.2.3>

Overview on how to use at:

<https://docs.microsoft.com/en-us/azure/azure-cache-for-redis/cache-java-get-started>

# MAVEN DEPENDENCIES

```
<dependency>
    <groupId>redis.clients</groupId>
    <artifactId>jedis</artifactId>
    <version>4.2.3</version>
</dependency>
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-core</artifactId>
    <version>2.13.4</version>
</dependency>
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.13.4</version>
</dependency>
```

# STEP 1: CREATE CLIENT TO REDIS (1)

## Alternatives:

1. Create a simple client to Redis. A client has a connection to the Redis server and it is not thread-safe.
  - Not a good option for using in an application server.
2. Use a pool of clients to Redis, that can be shared by multiple threads.

# STEP 1: CREATE CLIENT TO REDIS (2)

```
private static final String RedisHostname = ... ;
```

```
private static final String RedisKey = ...;
```

```
private static JedisPool instance;
```

```
public synchronized static JedisPool getCachePool() {
```

```
    if( instance != null)
```

```
        return instance;
```

```
    final JedisPoolConfig poolConfig = new JedisPoolConfig();
```

```
    poolConfig.setMaxTotal(128);
```

```
    poolConfig.setMaxIdle(128);
```

```
    poolConfig.setMinIdle(16);
```

```
    ...
```

```
    poolConfig.setBlockWhenExhausted(true);
```

```
    instance = new JedisPool(poolConfig, RedisHostname, 6380, 1000, RedisKey, true);
```

```
    return instance;
```

```
}
```

```
new JedisPool( config, hostname,  
               port, timeout, key, TLS)
```

# STEP 1: CREATE CLIENT TO REDIS (2)

```
try (Jedis jedis = RedisCache.getCachePool().getResource()) {  
  
    // Execute operations with a client to the Redis server  
  
}
```



## STEP 2: SET/GET A VALUE TO A KEY

```
ObjectMapper mapper = new ObjectMapper();
```

```
try (Jedis jedis = RedisCache.getCachePool().getResource()) {
```

```
    jedis.set("user:" + id, mapper.writeValueAsString(u));
```

```
    String res = jedis.get("user:" + id);
```

```
    System.out.println(res);
```

```
    Long cnt = jedis.lpush("MostRecentUsers", mapper.writeValueAsString(u));
```

```
    if (cnt > 5)
```

```
        jedis.ltrim("MostRecentUsers", 0, 4);
```

```
    List<String> lst = jedis.lrange("MostRecentUsers", 0, -1);
```

```
    for( String s : lst)
```

```
        System.out.println(s);
```

```
    cnt = jedis.incr("NumUsers");
```

```
    System.out.println("Num users : " + cnt);
```

```
}
```

## STEP 2: ADD ELEMENT TO LIST AND KEEP LIMIT

```
ObjectMapper mapper = new ObjectMapper();
try (Jedis jedis = RedisCache.getCachePool().getResource()) {
    jedis.set("user:" + id, mapper.writeValueAsString(u));
    String res = jedis.get("user:" + id);
    System.out.println(res);

    Long cnt = jedis.lpush("MostRecentUsers", mapper.writeValueAsString(u));
    if (cnt > 5)
        jedis.ltrim("MostRecentUsers", 0, 4);

    List<String> lst = jedis.lrange("MostRecentUsers", 0, -1);
    for( String s : lst)
        System.out.println(s);

    cnt = jedis.incr("NumUsers");
    System.out.println( "Num users : " + cnt);
}
```

NOTE: this is just an example for the use of a list... in your project it probably does not make sense to have a list of Users.

## STEP 2: GET THE ELEMENTS OF A LIST

```
ObjectMapper mapper = new ObjectMapper();
try (Jedis jedis = RedisCache.getCachePool().getResource()) {
    jedis.set("user:" + id, mapper.writeValueAsString(u));
    String res = jedis.get("user:" + id);
    System.out.println(res);

    Long cnt = jedis.lpush("MostRecentUsers", mapper.writeValueAsString(u));
    if (cnt > 5)
        jedis.ltrim("MostRecentUsers", 0, 4);

    List<String> lst = jedis.lrange("MostRecentUsers", 0, 1);
    for (String s : lst)
        System.out.println(s);

    cnt = jedis.incr("NumUsers");
    System.out.println("Num users : " + cnt);
}
```

## STEP 2: INCREMENT AND GET THE VALUE OF A COUNTER

```
ObjectMapper mapper = new ObjectMapper();
try (Jedis jedis = RedisCache.getCachePool().getResource()) {
    jedis.set("user:" + id, mapper.writeValueAsString(u));
    String res = jedis.get("user:" + id);
    System.out.println(res);

    Long cnt = jedis.lpush("MostRecentUsers", mapper.writeValueAsString(u));
    if (cnt > 5)
        jedis.ltrim("MostRecentUsers", 0, 4);

    List<String> lst = jedis.lrange("MostRecentUsers", 0, -1);
    for (String s : lst)
        System.out.println(s);

    cnt = jedis.incr("NumUsers");
    System.out.println("Num users : " + cnt);
}
```

# REDIS DATATYPES

<https://redis.io/topics/data-types>

- String
- List of strings (with insertion on head or tail)
- Set of strings
- Sorted set of strings
- Hashes (similar to a struct/map)
- Bit array
- HyperLogLogs (probabilistic data structure for estimating the number of elements in a set)
- ...

# USEFUL REDIS METHOD

- `expire( key, seconds)`

Allows to set an expire time for a key. After the time for the key expires, the key is automatically deleted from the cache.

# SOME NOTES

You should use keys that are unique for a given data type.

What should you keep in cache?

- Everything that might get accessed later;
- In case of doubt, just keep everything... it is a cache: what is not being used gets discarded.

# CODE PROVIDED

The code provided (lab4.zip) is a Maven project with a single class that stores a User object in a Redis list and the number of users created in a counter.

For testing it in the command line, just run:

```
mvn compile assembly:single
```

to compile and create a single file with all compiled classes and dependencies.

Run the program as follows:

```
java -cp target/scc2223-lab4-1.0-jar-with-dependencies.jar scc.utils.TestCache
```



# TODO

Extend your backend to use the Azure Cache whenever that makes sense.

NOTE: in the end, do not forget to delete the Azure Cache for Redis resources.

# GOAL

In the end of this lab you should be able to:

- Create a Azure Redis for Cache account @ Azure;
- Use AzureRedisCache to store cache data.
- **More professional deployment.**

# PROBLEMS

- Creating resources manually is time consuming
- Setting keys in the code is bad practice

# PROBLEMS

- **Creating resources manually is time consuming**
- Setting keys in the code is bad practice

# AUTOMATED CREATION OF RESOURCES

Possible to create resources:

- Programatically using management API;
- Using az command to create scripts.

# AUTOMATED CREATION OF RESOURCES USING MANAGEMENT API

Check scc-mgt.zip code at CLIP.

Code creates:

- Blob storage accounts, containers;
- Cosmos DB accounts, DB, containers;
- Redis servers.

Configure the code to change names, regions used, etc.

**Check comment with TODO.**

# AUTOMATED CREATION OF RESOURCES USING MANAGEMENT API (2)

Program also create two file per regions – e.g. westeurope:

- `azurekeys-westeurope.props`
  - Property file with URLs and keys for created resources
- `azureprops-westeurope.sh`
  - Shell script with commands to update application settings for both app service and Azure functions.
  - After starting the web app, you can run the commands.

# APPLICATION SETTINGS

Command to set application settings in Azure:

```
az functionapp config appsettings set --name app_name --resource-group res_group_name --  
settings "PROP=VAL"
```



# PROBLEMS

- Creating resources manually is time consuming
- **Setting keys in the code is bad practice**

Application settings can be accessed using system environment variable – `System.getenv(variable name)`

# ACCESSING KEY IN APPLICATION SETTINGS

**// Get connection string in the storage access keys page**

```
String storageConnectionString = System.getenv("BlobStoreConnection");
```

**// Get container client**

```
BlobContainerClient containerClient = new BlobContainerClientBuilder()  
    .connectionString(storageConnectionString)  
    .containerName("images")  
    .buildClient();
```

# CODE PROVIDED

The code provided (scc-mgt.zip) is a Maven project with a class that creates resources in Azure. Check TODO to configure for your needs.

For testing it in the command line, just run:

```
mvn compile assembly:single
```

to compile and create a single file with all compiled classes and dependencies.

Run the program as follows to create resources:

```
java -cp target/scc2223-mgt-1.0-jar-with-dependencies.jar scc.mgt.AzureManagement
```

Run the program as follows to delete resources:

```
java -cp target/scc2223-mgt-1.0-jar-with-dependencies.jar scc.mgt.AzureManagement -delete
```