

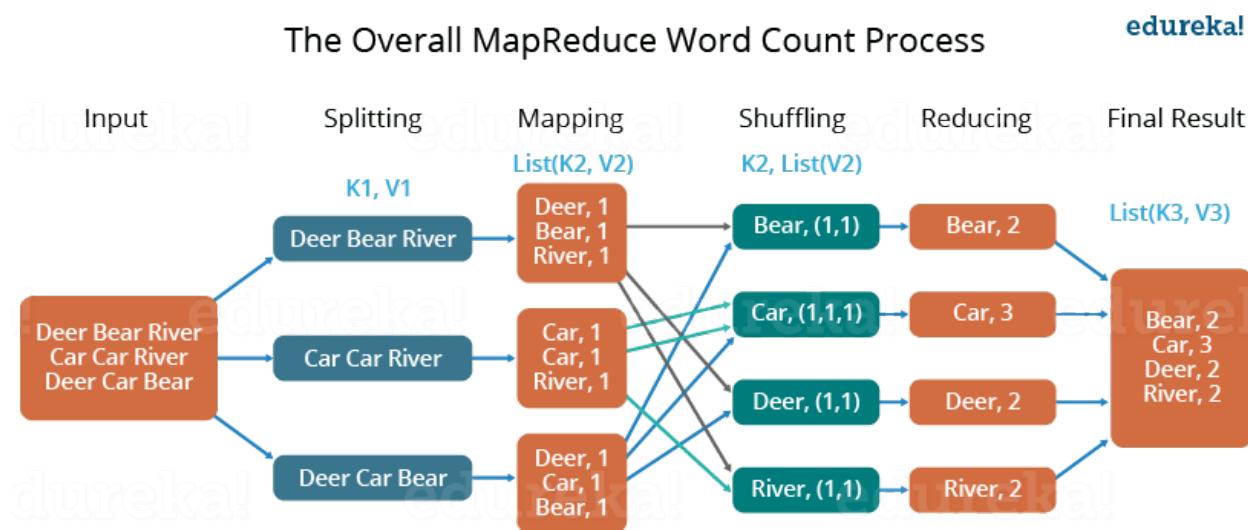
## Exercise #3: MapReduce and Replication

### Part 1. MapReduce

**Map-reduce** is a data processing paradigm for condensing large volumes of data into useful aggregated results. It typically performs distributed and parallel processing on large data sets in a distributed environment.

MapReduce consists of two distinct tasks – Map and Reduce:

- The reducer phase takes place after the mapper phase has been completed.
- In the map job, a block of data is read and processed to produce key-value pairs as intermediate outputs.
- The output of a Mapper or map job (key-value pairs) is input to the Reducer.
- The reducer receives the key-value pair from multiple map jobs.
- Then, the reducer aggregates those intermediate data tuples (intermediate key-value pair) into a smaller set of tuples or key-value pairs which is the final output.



Tasks:

- Task 1. Install CouchDB and get access
- Task 2. Create the music database
- Task 3. Insert Documents
- Task 4. Understanding MapReduce

## CouchDB



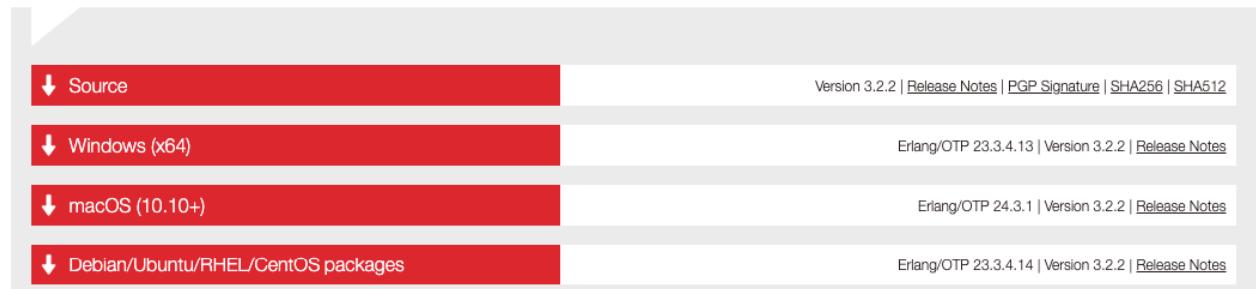
CouchDB is an open source database developed by Apache software foundation. The focus is on the ease of use, embracing the web. It is a NoSQL document store database.

It uses JSON, to store data (documents), JavaScript as its query language to transform the documents, HTTP protocol for API to access the documents, query the indices with the web browser.

### Task 1. Install CouchDB and get access

Install the latest version available <https://couchdb.apache.org/>

 Download CouchDB 3.2.2

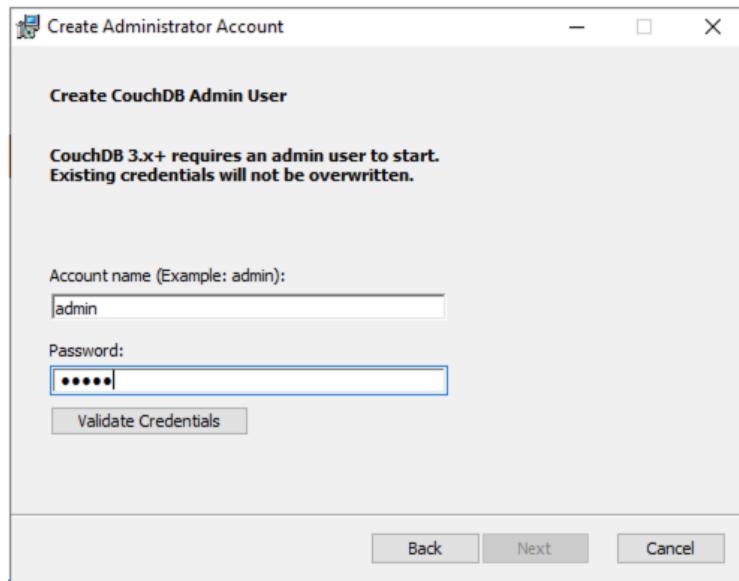


The screenshot shows the official CouchDB download page. At the top, there are four red download buttons:

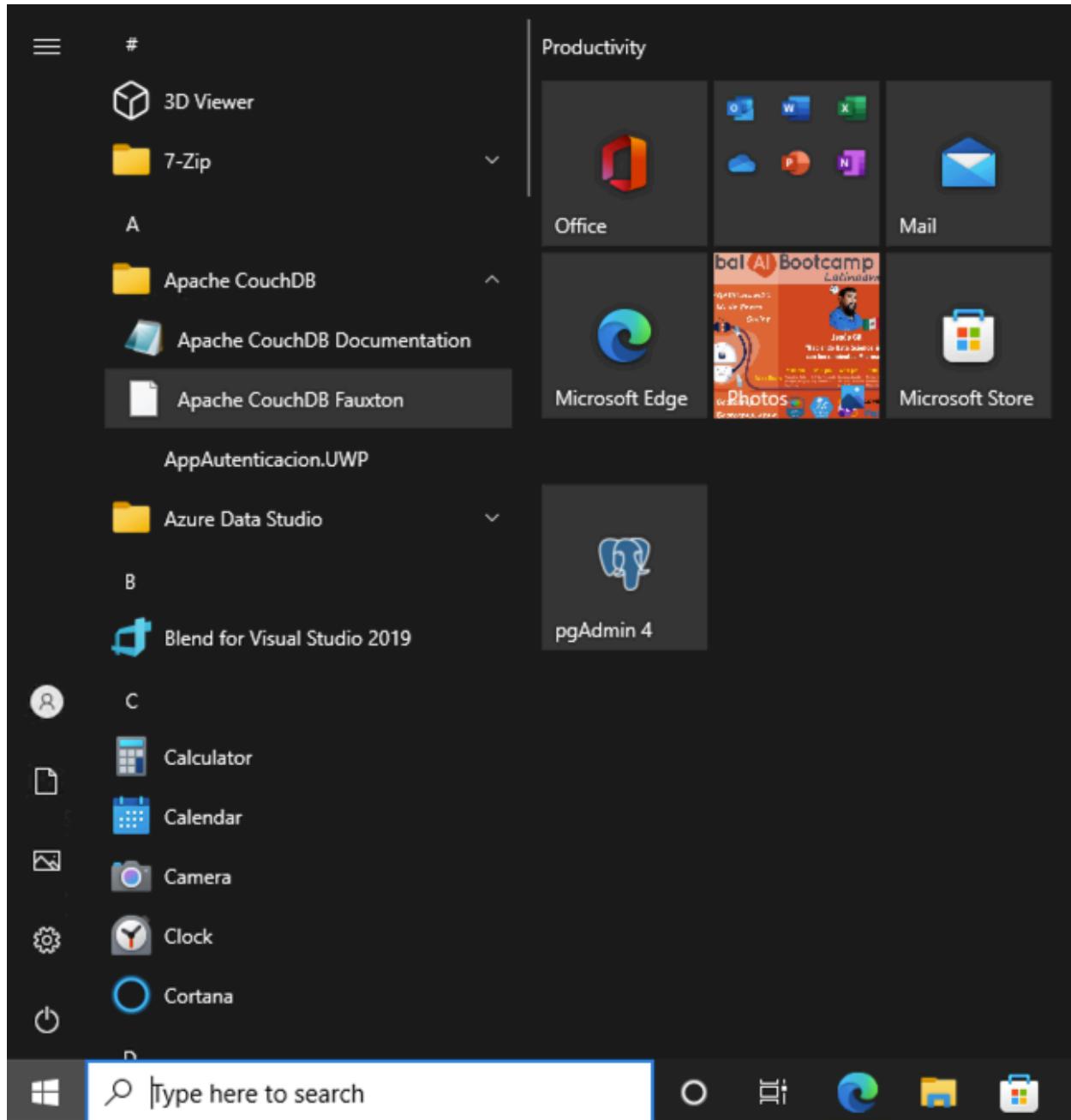
- Source (link to https://couchdb.apache.org/)
- Windows (x64) (link to https://couchdb.apache.org/)
- macOS (10.10+) (link to https://couchdb.apache.org/)
- Debian/Ubuntu/RHEL/CentOS packages (link to https://couchdb.apache.org/)

Below the buttons, there are links for Version 3.2.2 and Erlang/OTP versions.

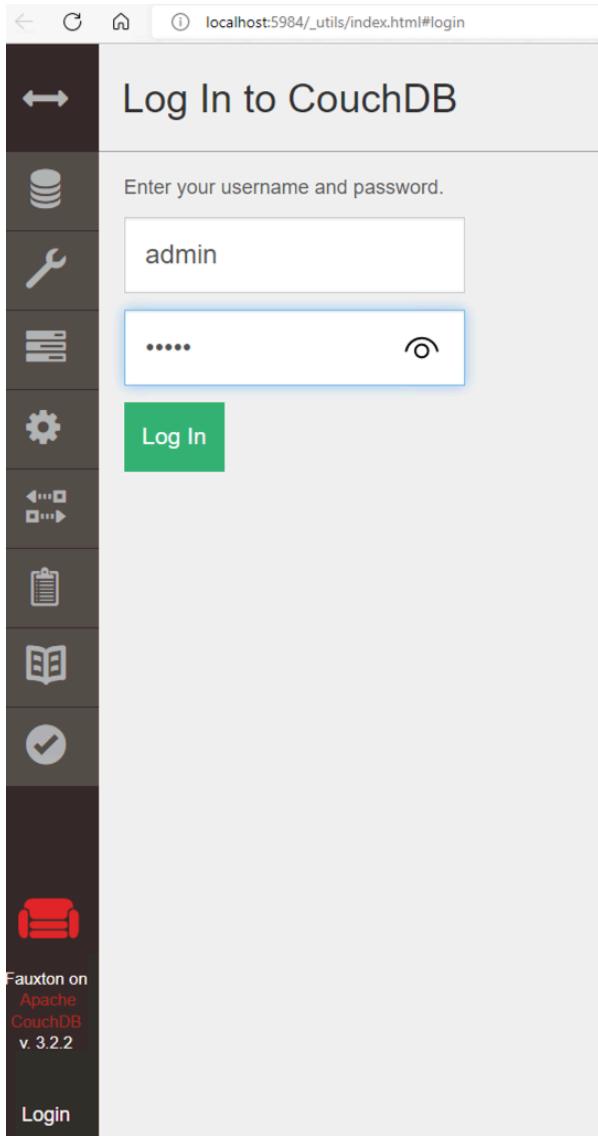
Set up a password for the “admin” user on install.



Open Apache CouchDB Fauxton



Enter your credentials



The screenshot shows a web browser window with the URL `localhost:5984/_utils/index.html#login`. The title bar says "Log In to CouchDB". On the left is a vertical sidebar with various icons: a double arrow (refresh), a wrench (settings), a grid (database), a gear (configuration), a double-headed arrow (replication), a clipboard (list), a book (view), a checkmark (status), and a red couch (Fauxton). Below the sidebar, it says "Fauxton on Apache CouchDB v. 3.2.2". At the bottom of the sidebar is a "Login" button. The main area has a heading "Enter your username and password." with two input fields: one for "admin" and one for a password (represented by four dots). To the right of the password field is a visibility icon. A green "Log In" button is at the bottom.

localhost:5984/\_utils/index.html#login

## Log In to CouchDB

Enter your username and password.

admin

.....

Log In

Fauxton on  
Apache  
CouchDB  
v. 3.2.2

Login

## Task 2. Create the music database

The screenshot shows the Fauxton interface for Apache CouchDB. On the left is a sidebar with various links: Databases (selected), Setup, Active Tasks, Configuration, Replication, News, Documentation, Verify, Your Account, and a CouchDB logo. The main area is titled "Databases" and shows a table with one entry: "\_replicator" (Size: 4.3 KB, # of Docs: 1, Partitioned: No). A modal window titled "Create Database" is open over the table. In the "Database name" input field, the value "music" is entered and highlighted with a red box. Below the input field is a "Partitioning" section with a checkbox labeled "Partitioned". To the right of the checkbox is a link "What is a Partitioned Database?". At the bottom of the modal are "Cancel" and "Create" buttons, with a red arrow pointing towards the "Create" button.

## Expected result

The screenshot shows the Fauxton interface after the database has been created. The sidebar remains the same. The main area now displays the "music" database. The "All Documents" section is visible, and a green notification bar at the top right states "Database created successfully".

### Task 3. Insert Documents

From Moodle, download the file “**MuseAlbums.json**”

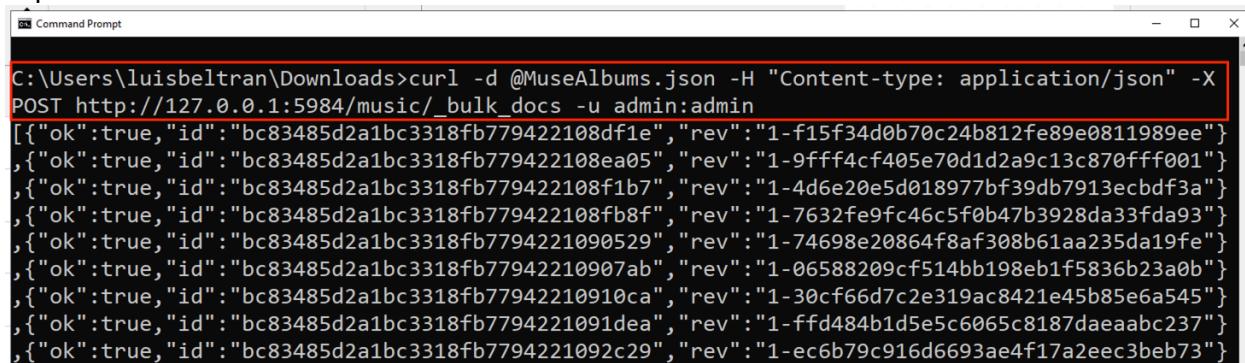
Open a Terminal (Command Prompt) and set it to the Downloads folder

Run the following command

```
curl -d @MuseAlbums.json -H "Content-type: application/json" -X  
POST http://127.0.0.1:5984/music/_bulk_docs -u user:pass
```

Replace **user** and **pass** with your credentials

Expected result:



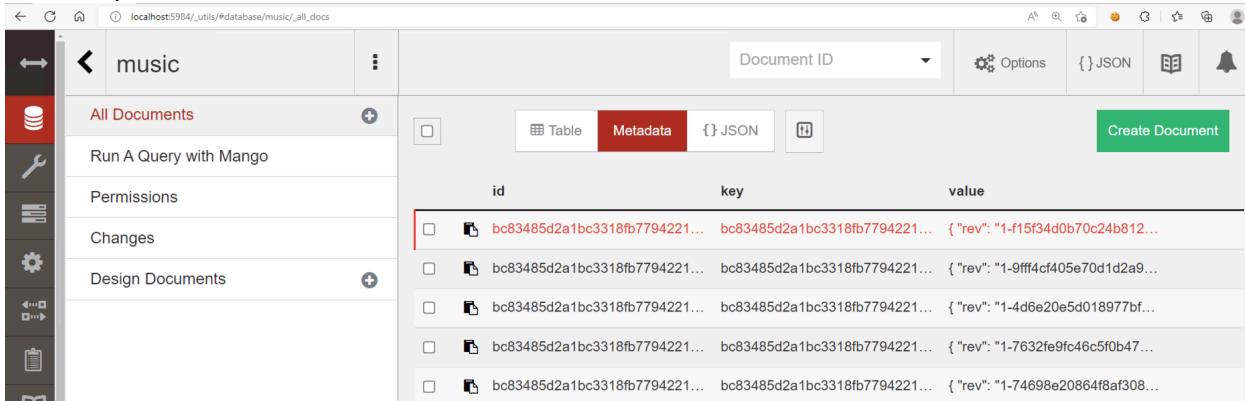
```
C:\Users\luisbeltran\Downloads>curl -d @MuseAlbums.json -H "Content-type: application/json" -X  
POST http://127.0.0.1:5984/music/_bulk_docs -u admin:admin
```

The output shows a list of 10 JSON objects, each representing a document inserted into the 'music' database. Each object contains fields: 'ok' (true), 'id' (a unique document ID), 'rev' (the revision number), and a 'value' field containing the document's content.

id	key	value
bc83485d2a1bc3318fb779422108df1e	bc83485d2a1bc3318fb779422108df1e	{ "rev": "1-f15f34d0b70c24b812fe89e0811989ee" }
bc83485d2a1bc3318fb779422108ea05	bc83485d2a1bc3318fb779422108ea05	{ "rev": "1-9fff4cf405e70d1d2a9c13c870fff001" }
bc83485d2a1bc3318fb779422108fb1b7	bc83485d2a1bc3318fb779422108fb1b7	{ "rev": "1-4d6e20e5d018977bf39db7913ecbd3a" }
bc83485d2a1bc3318fb779422108fb8f	bc83485d2a1bc3318fb779422108fb8f	{ "rev": "1-7632fe9fc46c5f0b47b3928da33fd93" }
bc83485d2a1bc3318fb7794221090529	bc83485d2a1bc3318fb7794221090529	{ "rev": "1-74698e20864f8af308b61aa235da19fe" }
bc83485d2a1bc3318fb77942210907ab	bc83485d2a1bc3318fb77942210907ab	{ "rev": "1-06588209cf514bb198eb1f5836b23a0b" }
bc83485d2a1bc3318fb77942210910ca	bc83485d2a1bc3318fb77942210910ca	{ "rev": "1-30cf66d7c2e319ac8421e45b85e6a545" }
bc83485d2a1bc3318fb7794221091dea	bc83485d2a1bc3318fb7794221091dea	{ "rev": "1-ffd484b1d5e5c6065c8187daeaabc237" }
bc83485d2a1bc3318fb7794221092c29	bc83485d2a1bc3318fb7794221092c29	{ "rev": "1-ec6b79c916d6693ae4f17a2eec3beb73" }

**NOTE: If the command doesn't work, install Git to enable bash commands (see Appendix)**

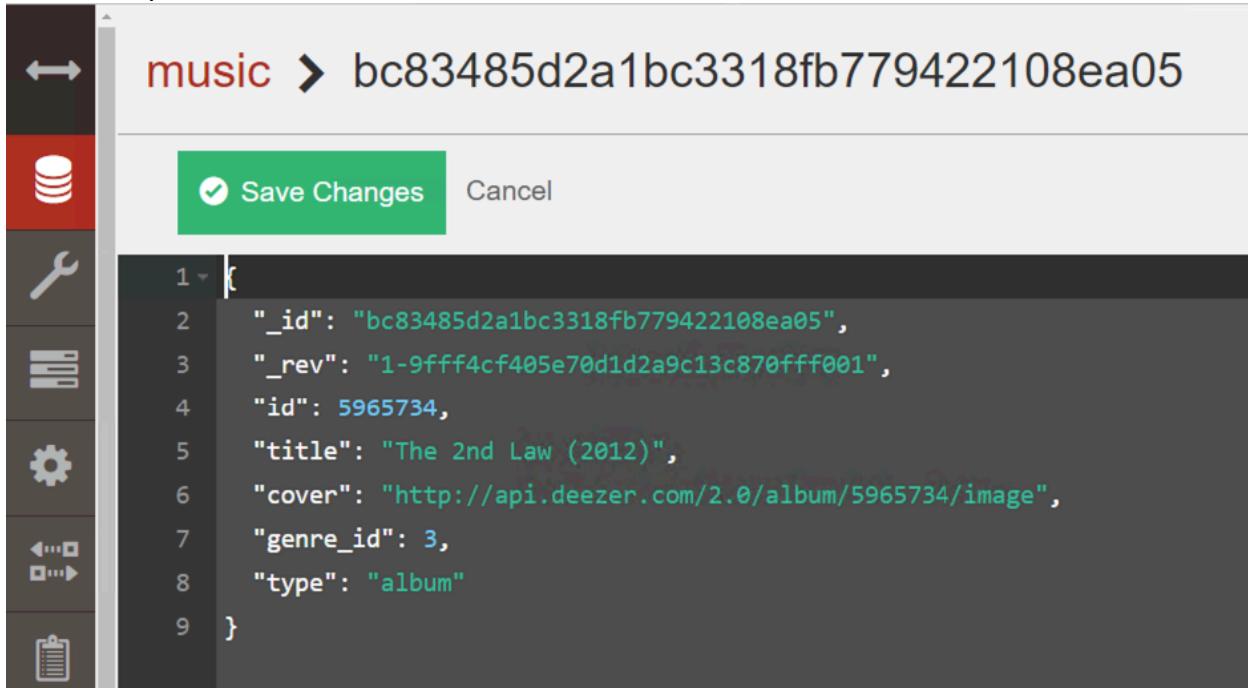
Refresh your database view on Fauxton



The Fauxton interface displays the 'music' database. On the left, there is a sidebar with icons for database management. The main area shows a table with columns: 'id', 'key', and 'value'. There are 10 rows of data, each corresponding to one of the documents inserted via the curl command. The 'value' column shows the JSON structure of each document.

id	key	value
bc83485d2a1bc3318fb779422108df1e	bc83485d2a1bc3318fb779422108df1e	{ "rev": "1-f15f34d0b70c24b812fe89e0811989ee" }
bc83485d2a1bc3318fb779422108ea05	bc83485d2a1bc3318fb779422108ea05	{ "rev": "1-9fff4cf405e70d1d2a9c13c870fff001" }
bc83485d2a1bc3318fb779422108fb1b7	bc83485d2a1bc3318fb779422108fb1b7	{ "rev": "1-4d6e20e5d018977bf39db7913ecbd3a" }
bc83485d2a1bc3318fb779422108fb8f	bc83485d2a1bc3318fb779422108fb8f	{ "rev": "1-7632fe9fc46c5f0b47b3928da33fd93" }
bc83485d2a1bc3318fb7794221090529	bc83485d2a1bc3318fb7794221090529	{ "rev": "1-74698e20864f8af308b61aa235da19fe" }
bc83485d2a1bc3318fb77942210907ab	bc83485d2a1bc3318fb77942210907ab	{ "rev": "1-06588209cf514bb198eb1f5836b23a0b" }
bc83485d2a1bc3318fb77942210910ca	bc83485d2a1bc3318fb77942210910ca	{ "rev": "1-30cf66d7c2e319ac8421e45b85e6a545" }
bc83485d2a1bc3318fb7794221091dea	bc83485d2a1bc3318fb7794221091dea	{ "rev": "1-ffd484b1d5e5c6065c8187daeaabc237" }
bc83485d2a1bc3318fb7794221092c29	bc83485d2a1bc3318fb7794221092c29	{ "rev": "1-ec6b79c916d6693ae4f17a2eec3beb73" }

Click on any document to see its details



```
1 [
2   "_id": "bc83485d2a1bc3318fb779422108ea05",
3   "_rev": "1-9fff4cf405e70d1d2a9c13c870ffff001",
4   "id": 5965734,
5   "title": "The 2nd Law (2012)",
6   "cover": "http://api.deezer.com/2.0/album/5965734/image",
7   "genre_id": 3,
8   "type": "album"
9 }
```

Other useful commands:

See information about the CouchDB environment

```
curl http://127.0.0.1:5984/
```

See the list of all databases

```
curl -X GET http://127.0.0.1:5984/_all_dbs -u user:pass
```

Create a database

```
curl -X PUT http://127.0.0.1:5984/database_name -u user:pass
```

Getting database information

```
curl -X GET http://127.0.0.1:5984/my_database -u user:pass
```

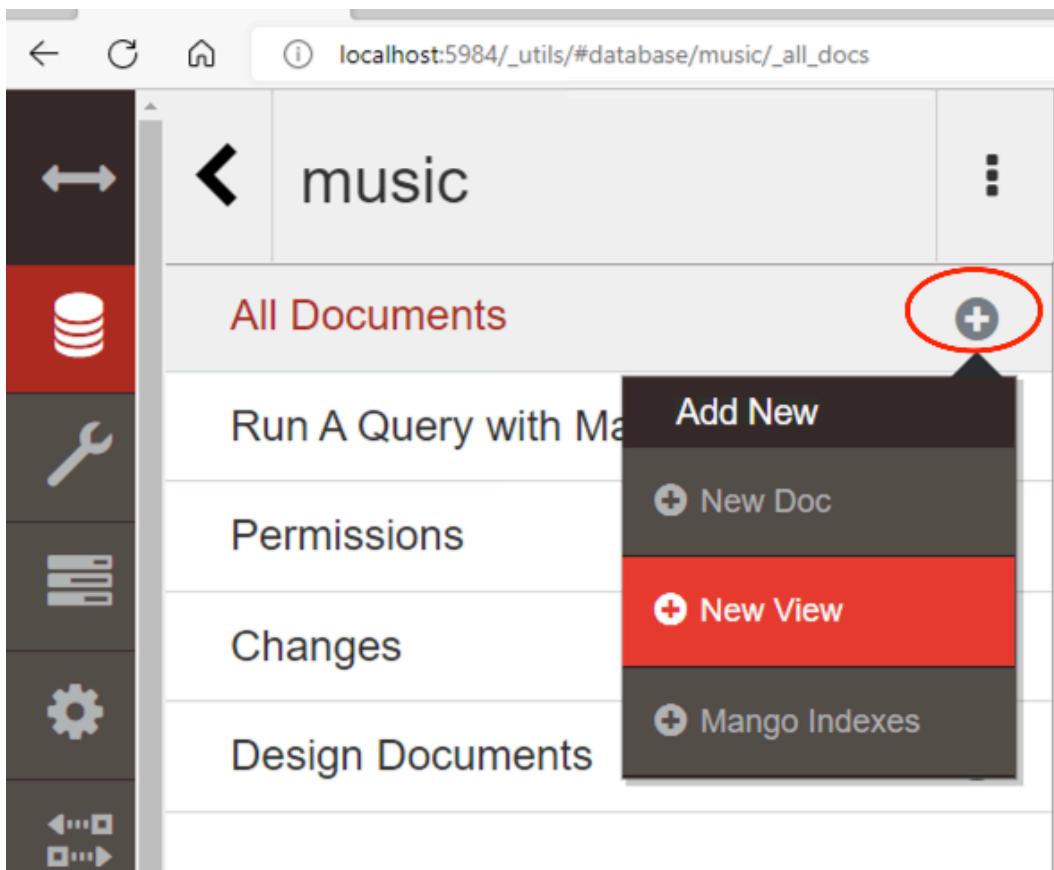
#### Task 4. Understanding MapReduce

- All CouchDB queries are MapReduce operations.
- Finding matching documents is the “Map” portion of MapReduce
- The Reduce portion involves calculating, aggregating, and returning the results of a query

Steps:

- Create a View
- Add a Map function
- Add a Reduce code

In your database page, click on the + button and choose **New View**.



Set the following:

- Design name is **MusicInfo**
- Index name is **AlbumsByGenre**
- Emit the **genre\_id** field
- Create it

New View

Design Document ?

New document ▾ \_design/ **MusicInfo**

Index name ? **AlbumsByGenre**

Map function ?

```
1 function (doc) {  
2   emit(doc.genre_id, 1);  
3 }
```

Reduce (optional) ?

NONE

 **Create Document and then Build Index** Cancel

If you check the View, a key field will be displayed (it contains the genre\_id)

The screenshot shows the Apache CouchDB Futon interface. On the left, there's a sidebar with various database management options like 'All Documents', 'Run A Query with Mango', 'Permissions', 'Changes', 'Design Documents', and a 'MusicInfo' section which is expanded to show 'Metadata' and 'Views'. The 'Views' section contains a single item: 'AlbumsByGenre', which is highlighted with a red border. On the right, the main area displays a table titled 'AlbumsByGenre' with three columns: 'id', 'key', and 'value'. The 'key' column is explicitly labeled as containing the genre\_id. The table contains 10 rows, each with a file icon, a long ID string, and the value '3'.

id	key	value
bc83485d2a1bc3318fb7794221...	2	1
bc83485d2a1bc3318fb7794221...	3	1

Click on the wrench to edit the AlbumsByGenre View

This screenshot shows the 'Edit' menu for the 'AlbumsByGenre' view. The menu is a vertical list with three items: 'Edit', 'Clone', and 'Delete'. Each item has an associated icon: a wrench for 'Edit', a copy symbol for 'Clone', and a trash can for 'Delete'. The 'Edit' button is highlighted with a red background.

- Edit
- Clone
- Delete

Add a Reduce element in your view. Select the **count** aggregate function.

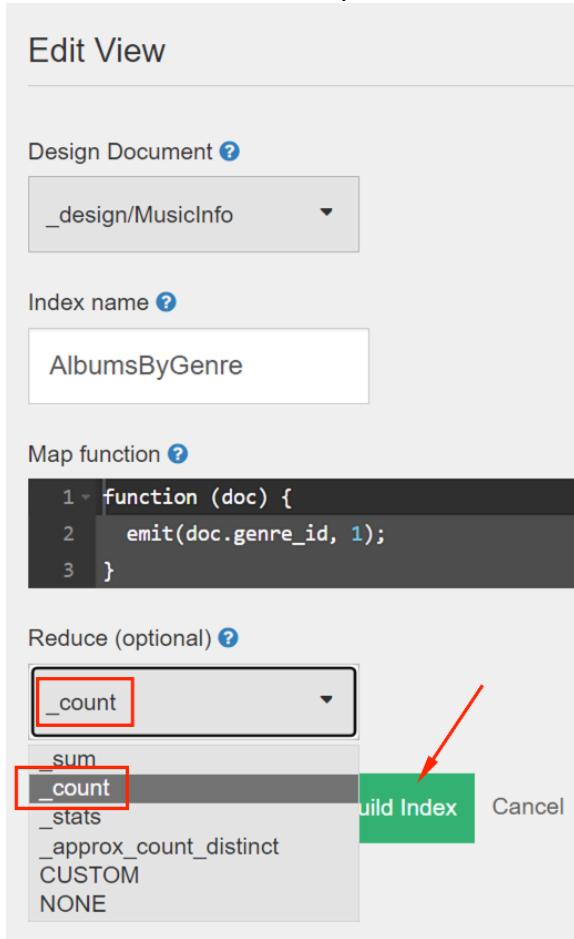
Edit View

Design Document ?  
\_design/MusicInfo

Index name ?  
AlbumsByGenre

Map function ?  
1. function (doc) {  
2. emit(doc.genre\_id, 1);  
3. }

Reduce (optional) ?  
       
Build Index Cancel



Run the following command in Terminal / Command Prompt

```
curl  
http://127.0.0.1:5984/music/_design/MusicInfo/_view/AlbumsByGenre?  
e?group=true -u user:pass
```

Expected result

```
luisbeltran@dev-vm MINGW64 ~/Downloads  
$ curl http://127.0.0.1:5984/music/_design/MusicInfo/_view/AlbumsByGenre?group=true -u admin:admin  
{"rows": [  
  {"key": 2, "value": 1},  
  {"key": 3, "value": 33},  
  {"key": 6, "value": 4},  
  {"key": 7, "value": 1},  
  {"key": 10, "value": 1},  
  {"key": 11, "value": 2}  
]}
```

Create a second view, this time selecting a Custom Reduce function. The configuration is:

- Design document: Music2013Info
- View name: Albums2013ByGenre
- Map function

```
function (doc) {  
  if (doc.title && doc.title.indexOf('2013') !== -1)  
    emit(doc.genre_id, 1);  
}
```

- Reduce: **CUSTOM**

```
function (keys, values) {  
  return sum(values);  
}
```

The screenshot shows the Apache CouchDB Futon interface. On the left, there is a sidebar with a navigation tree. The tree includes 'All Documents', 'Run A Query with Mango', 'Permissions', 'Changes', 'Design Documents', and a expanded section for 'Music2013Info' which contains 'Metadata' and 'Views'. Under 'Views', there is an entry for 'Albums2013ByGenre' which is highlighted in red. Below this, another section for 'MusicInfo' is shown, followed by 'Metadata' and 'Views' for 'AlbumsByGenre'. On the right, the main panel is titled 'Edit View'. It has several configuration fields:

- 'Design Document': A dropdown menu set to '\_design/Music2013Info'.
- 'Index name': A text input field containing 'Albums2013ByGenre'.
- 'Map function': A code editor containing the provided map function, with the entire code block highlighted in red.
- 'Reduce (optional)': A dropdown menu set to 'CUSTOM'.
- 'Custom Reduce function': A code editor containing the provided reduce function, with the entire code block highlighted in red.

At the bottom of the right panel, there are two buttons: a green 'Save Document and then Build Index' button with a checked checkbox, and a 'Cancel' button.

Test it by using the terminal:

```
curl  
http://127.0.0.1:5984/music/_design/Music2013Info/_view/Albums2013ByGenre?group=true -u user:pass
```

```
luisbeltran@dev-vm MINGW64 ~/Downloads  
$ curl http://127.0.0.1:5984/music/_design/Music2013Info/_view/Albums2013ByGenre?group=true -u admin:admin  
{"rows": [  
 {"key":3,"value":1},  
 {"key":6,"value":1},  
 {"key":7,"value":1},  
 {"key":11,"value":2}  

```

### Extend the example with a new view

This is the Map function

```
function (doc) {  
 var years = ["2012", "2013", "1999", "2001"]  
  
years.forEach(function(year) {  
 if (doc.title && doc.title.indexOf(year) !== -1)  
 emit(year, doc.title);  
});  
}
```

Reduce function: \_count

Test

```
$ curl http://127.0.0.1:5984/music/_design/MusicInfo/_view/AlbumsByYear?group=true -u admin:admin  
{"rows": [  
 {"key": "1999", "value": 3},  
 {"key": "2001", "value": 5},  
 {"key": "2012", "value": 4},  
 {"key": "2013", "value": 5}  

```

## Part 2. Replication

Click on the **Replication** menu and then click on **New Replication**

The screenshot shows the Apache Derby Replication interface. On the left is a sidebar with links: Databases, Setup, Active Tasks, Configuration, Replication (which is selected and highlighted in red), News, and Documentation. The main area has tabs: Replicator DB Activity (selected) and replicate Activity. A message says "Replications must have a replication document to display in the following table." Below is a table header with columns: Source, Target, Start Time, Type, State, and Actions. A green button labeled "New Replication" is visible at the bottom right of the table area, with a red arrow pointing to it.

Set the Source (original database) configuration as follows:

The screenshot shows the "Source" configuration dialog. It has three fields: "Type:" set to "Local database", "Name:" set to "music", and "Authentication:" set to "Username and password". The "Authentication:" field is highlighted with a black border. Below it are two input fields: "admin" and "\*\*\*\*\*" (representing a password).

Set the Target (replica) as follows

## Target

Type: New local database

New database: music\_replica

New database options:  Partitioned

Authentication: Username and password

admin

.....

Finally, set the Replication type as Continuous, then Start the replication process

## Options

Replication type: Continuous

Replication document: Custom ID (optional) X

Start Replication Clear

You will observe that now there are two databases. Please note the number of documents in each database.

Databases			
	Name	Size	# of Docs
	_replicator	4.9 KB	2 ⓘ
	music	23.6 KB	44
	music_replica	15.2 KB	44

Insert a new document in the **music** database

music ➤ New Document

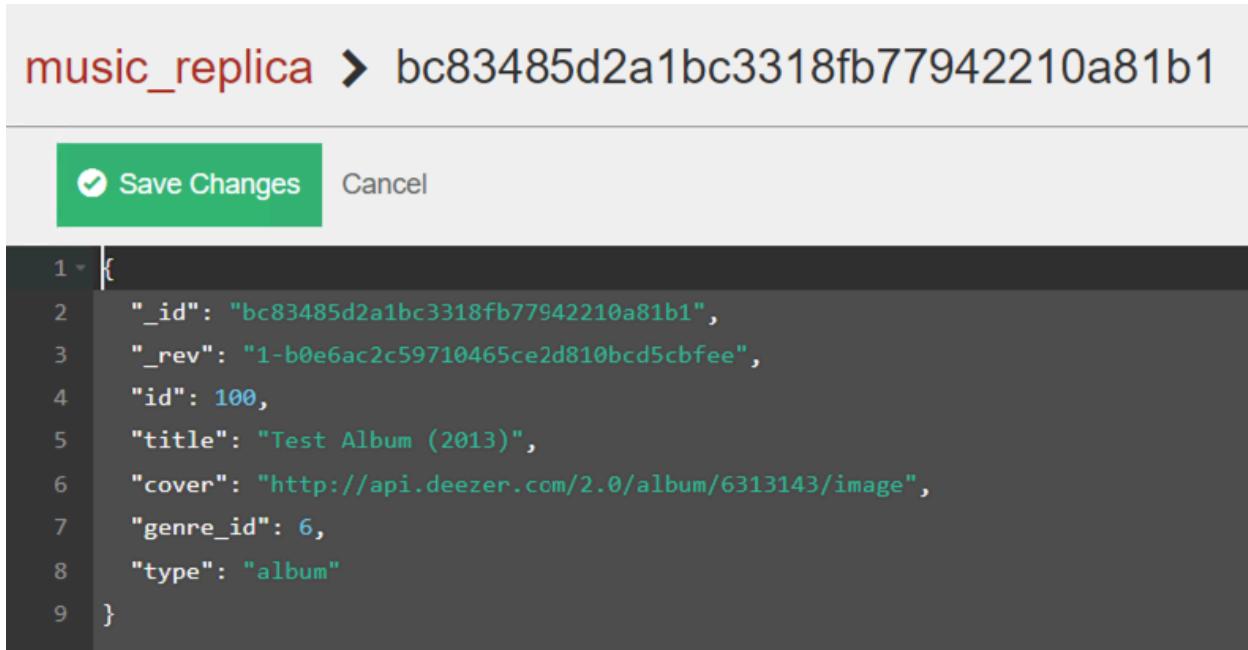
Create Document Cancel

```
1 {  
2   "id":100,  
3   "title":"Test Album (2013)",  
4   "cover":"http://api.deezer.com/2.0/album/6313143/image",  
5   "genre_id":6,  
6   "type":"album"  
7 }
```

Check the databases and observe that this document was replicated in **music\_replica** database

Databases			
	Name	Size	# of Docs
	_replicator	4.9 KB	2 ⓘ
	music	24.6 KB	45
	music_replica	16.1 KB	45

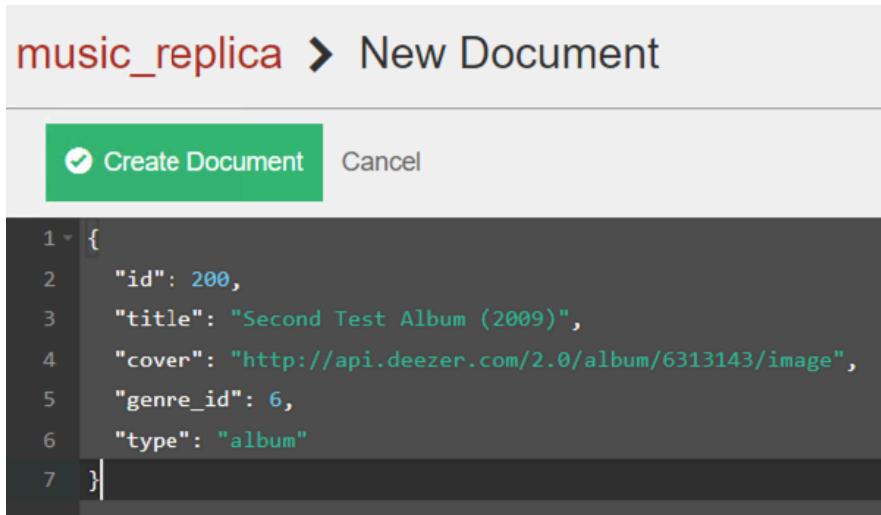
Of course you can check its details



The screenshot shows a database interface with a header "music\_replica" and a document ID "bc83485d2a1bc3318fb77942210a81b1". Below the header is a green button labeled "Save Changes" with a checkmark icon. To its right is a "Cancel" link. The main area displays a JSON document with line numbers 1 through 9 on the left. The document content is:

```
1  [
2    "_id": "bc83485d2a1bc3318fb77942210a81b1",
3    "_rev": "1-b0e6ac2c59710465ce2d810bcd5cbfee",
4    "id": 100,
5    "title": "Test Album (2013)",
6    "cover": "http://api.deezer.com/2.0/album/6313143/image",
7    "genre_id": 6,
8    "type": "album"
9 }
```

However, if you insert a new document in music\_replica, that information is not replicated to music database because the job does not work in both directions.



```
1 {  
2   "id": 200,  
3   "title": "Second Test Album (2009)",  
4   "cover": "http://api.deezer.com/2.0/album/6313143/image",  
5   "genre_id": 6,  
6   "type": "album"  
7 }
```

Now the number of documents in each database is different

Name	Size	# of Docs
_replicator	4.9 KB	2 ⓘ
music	24.6 KB	45
music_replica	16.6 KB	46

## **Appendix**

### **Azure account**

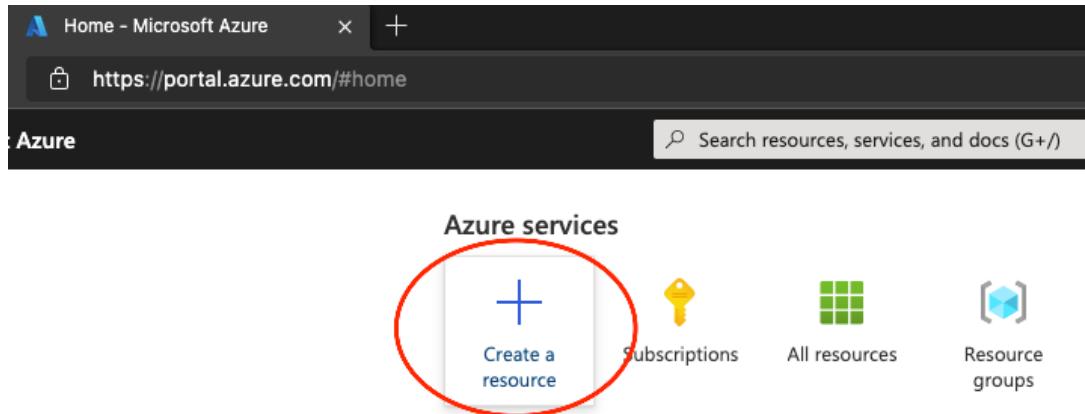
Follow the tutorial

<https://dev.to/esdanielgomez/creating-azure-for-students-account-48g>

Virtual machine

Go to the Azure Portal: <https://portal.azure.com/>

Click on **Create a resource**



Select **Virtual machine** or type it in the Search box

## Create a resource

The screenshot shows the 'Create a resource' page. At the top, there are sections for 'Get Started', 'Recently created', and a search bar labeled 'Search services and marketplace'. To the right, there's a 'Getting Started' link with a rocket icon. Below these, the 'Popular Azure services' section is displayed, with a link to 'See more in All services'. The services are listed in two columns:

SQL Database <a href="#">Create</a>   <a href="#">Docs</a>   <a href="#">MS Learn</a>	Key Vault <a href="#">Create</a>   <a href="#">Docs</a>   <a href="#">MS Learn</a>
Template deployment (deploy using custom templates) <a href="#">Create</a>   <a href="#">Learn more</a>	Web App <a href="#">Create</a>   <a href="#">Docs</a>   <a href="#">MS Learn</a>
Virtual machine <a href="#">Create</a>   <a href="#">Learn more</a>	Function App <a href="#">Create</a>   <a href="#">Docs</a>
Identity	
Integration	
Internet of Things	
IT & Management Tools	

In Project details section, click on **Create new** under Resource group and type **couchdb-rg**

#### Project details

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

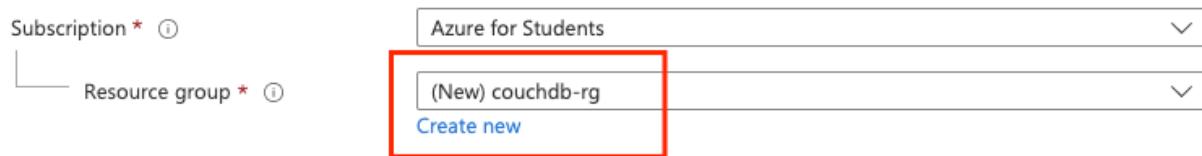
Subscription \* ⓘ

Azure for Students

Resource group \* ⓘ

(New) couchdb-rg

Create new



In Instance details, set:

- **couchdb-vm** as virtual machine name
- **North Europe** as region
- **Windows 10 Pro** as Image
- **Standard D2s v3** as size

#### Instance details

Virtual machine name \* ⓘ

couchdb-vm

Region \* ⓘ

(Europe) North Europe

Availability options ⓘ

No infrastructure redundancy required

Security type ⓘ

Standard

Image \* ⓘ

Windows 10 Pro, version 21H2 - Gen2

See all images | Configure VM generation

VM architecture ⓘ

Arm64

x64

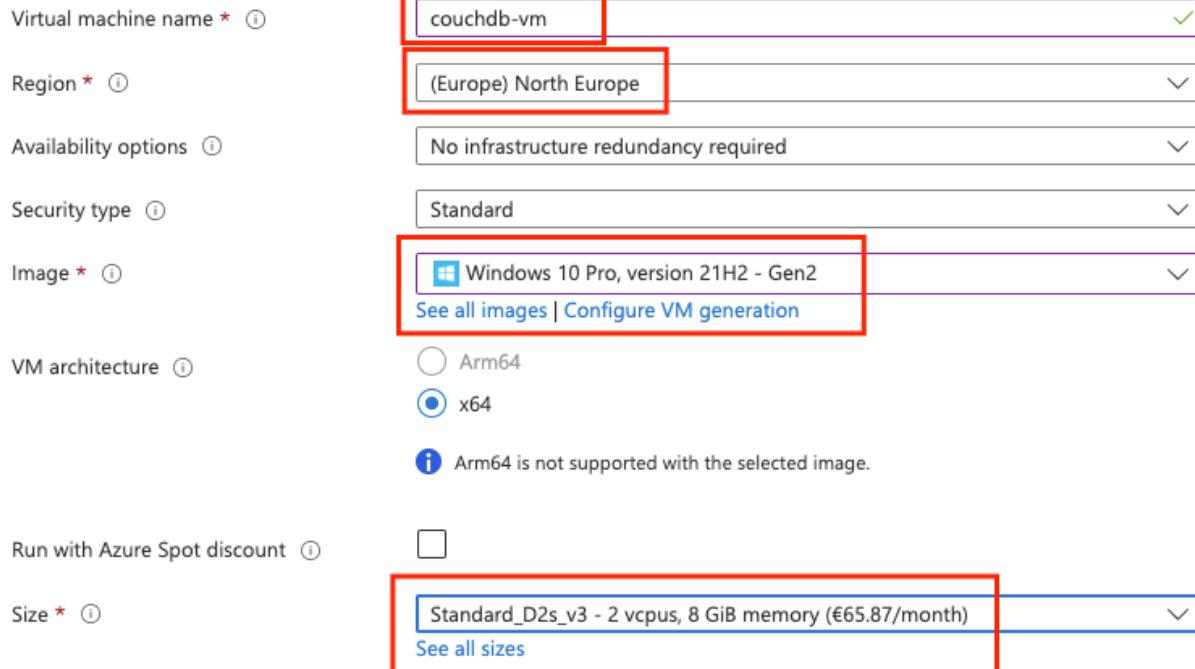
ⓘ Arm64 is not supported with the selected image.

Run with Azure Spot discount ⓘ

Size \* ⓘ

Standard\_D2s\_v3 - 2 vcpus, 8 GiB memory (€65.87/month)

See all sizes



Set **username** and a **strong password** (and confirm it) in the Administrator account section. Moreover, select the **3389 RDP port** in inbound port rules.

#### Administrator account

Username \* ⓘ

luisbeltran



Password \* ⓘ

\*\*\*\*\*



Confirm password \* ⓘ

\*\*\*\*\*



#### Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports \* ⓘ

None

Allow selected ports

Select inbound ports \*

RDP (3389)



**⚠ This will allow all IP addresses to access your virtual machine.** This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

Finally, **accept** the Licensing and click on **Review + create**

#### Licensing



I confirm I have an eligible Windows 10/11 license with multi-tenant hosting rights.



[Review multi-tenant hosting rights for Windows 10/11 compliance ↗](#)

[Review + create](#)

< Previous

Next : Disks >

Once your resource is created, click on **Connect**, download the **RDP** file, double click on it and enter your credentials (username and password) to access your virtual machine, where you will create the resources.

The screenshot shows the Azure portal interface for a virtual machine named "couchdb-vm". The "RDP" option in the top navigation bar is highlighted with a red box. The main content area displays the VM's properties, including its resource group, status, location, subscription, and tags. Below the properties, there are tabs for "Properties", "Monitoring", "Capabilities (7)", "Recommendations", and "Tutorials". A warning message at the bottom left says, "⚠ To improve security, enable just-in-time access on this VM. →". Under the "RDP" tab, there is a section titled "Connect with RDP" with a sub-section "Suggested method for connecting". It lists three prerequisites: "Checking network security group for inbound access from your client's IP address", "The VM's network interface has a Public IP address", and "The VM is running", all of which are marked as checked. Below this, instructions say, "To connect to your virtual machine via RDP, select an IP address, optionally change the port number, and download the RDP file." The "IP address" field contains "Public IP address (20.234.14.77)" and the "Port number" field contains "3389". A blue button labeled "Download RDP File" is at the bottom, with a red arrow pointing to it.

## Enter Your User Account

This user account will be used to connect to 20.234.14.77:3389  
(remote PC).

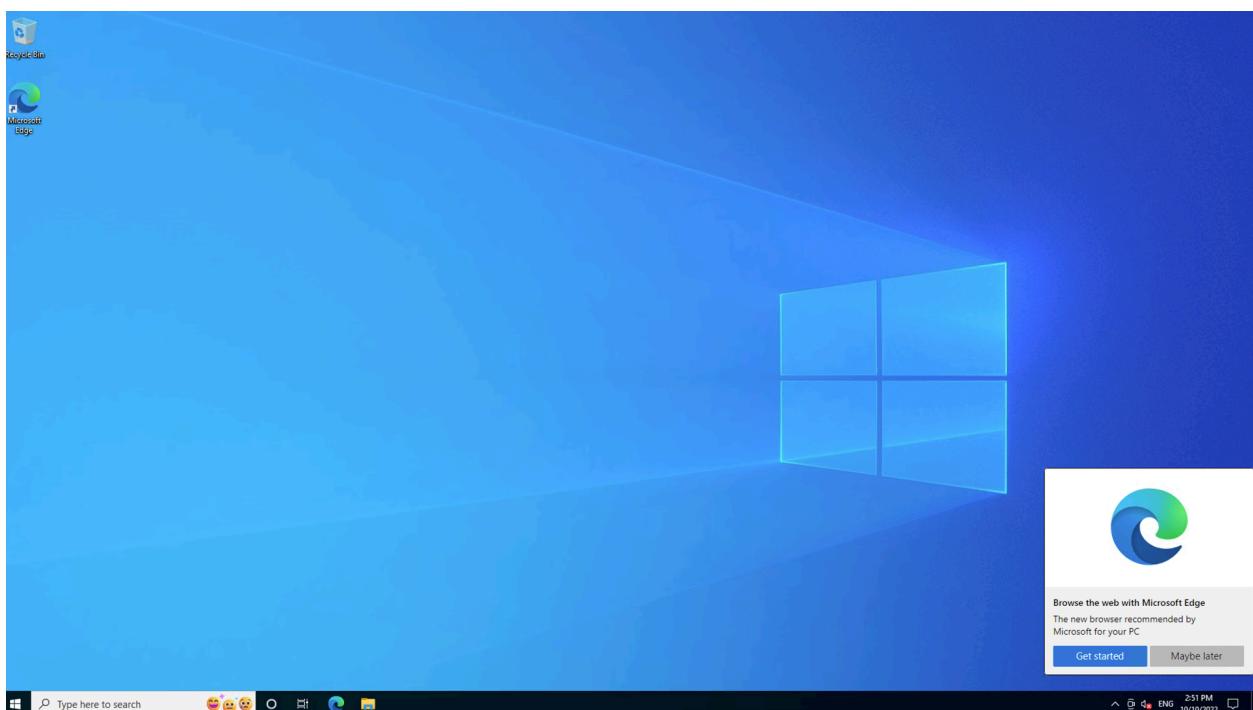
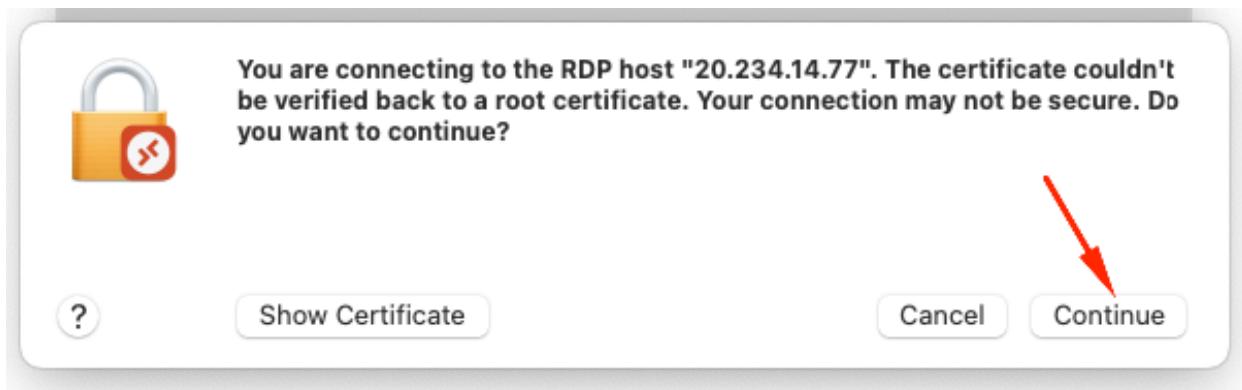
Username: luisbeltran

Password: 

Show password

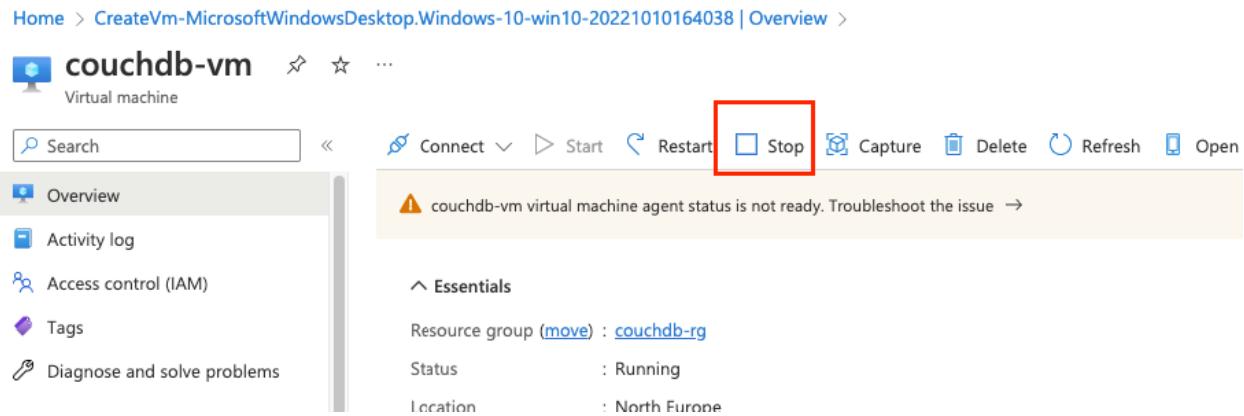
[Cancel](#)

[Continue](#)



**IMPORTANT:**

**When you don't use the virtual machine, visit the portal, access the resource and click on Stop to avoid consumption of your Azure credits.**



The screenshot shows the Azure portal interface for a virtual machine named 'couchdb-vm'. The top navigation bar includes 'Search', 'Connect', 'Start', 'Restart', a redboxed 'Stop' button, 'Capture', 'Delete', 'Refresh', and 'Open'. Below the navigation bar, there's a status message: 'couchdb-vm virtual machine agent status is not ready. Troubleshoot the issue →'. On the left, a sidebar lists 'Overview' (which is selected and highlighted in grey), 'Activity log', 'Access control (IAM)', 'Tags', and 'Diagnose and solve problems'. The main content area displays 'Essentials' information: Resource group ([move](#)) : couchdb-rg, Status : Running, and Location : North Europe.

**When you want to use the resource again, just click on Start and Connect + RDP, because a new IP will be used.**

## Git

Download Git from <https://git-scm.com/> (click on Downloads)

The screenshot shows the official Git website at <https://git-scm.com/>. The page features a large header with the Git logo and the tagline "fast-version-control". Below the header, there's a brief introduction to Git as a free and open source distributed version control system. To the right, a 3D diagram illustrates a distributed network where multiple repositories are interconnected by bidirectional arrows. On the left side of the main content area, there are four circular icons with text labels: "About" (gear icon), "Documentation" (book icon), "Downloads" (download arrow icon), and "Community" (speech bubble icon). The "Downloads" section is highlighted with a red border. On the right, a computer monitor displays a teal-colored window titled "Latest source Release 2.38.0" with a "Release Notes (2022-10-02)" link and a "Download for Windows" button.

Choose your OS

This screenshot shows the "Downloads" page of the Git website. At the top, there are four buttons for "macOS", "Windows" (which is highlighted with a red border), "Linux/Unix", and "GUI Clients". Below these buttons, a note states that older releases are available and the Git source repository is on GitHub. The "GUI Clients" section contains a paragraph about the built-in tools and third-party options, with a link to "View GUI Clients →". To the right, another computer monitor displays the same "Latest source Release 2.38.0" window as seen in the previous screenshot.

Follow the installation process. When it is successfully installed, open a new Terminal (Command Prompt) and test the **curl** command

```
cmd Command Prompt
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\luisbeltran>curl
curl: try 'curl --help' for more information

C:\Users\luisbeltran>
```