
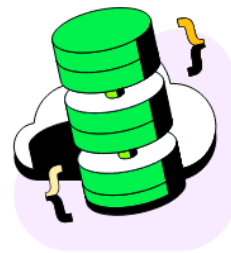
 Ngõ Lương ...  Access Manager ▾ Billing

Project 0 ▾ Data Services App Services Charts

**Overview**


NGÕ LƯƠNG THANH TRÀ'S ORG - 2024-09-02 > PROJECT 0

## Overview



### Create a cluster

Choose your cloud provider, region, and specs.

 [+ Create](#)

**DEPLOYMENT**

- Database
- Data Lake

**SERVICES**

- Device & Edge Sync
- Triggers
- Data API
- Data Federation
- Atlas Search
- Atlas Vector Search
- Stream Processing
- Migration

**SECURITY**

- Quickstart
- Backup

# Deploy your cluster

Use a template below or set up advanced configuration options. You can also edit these configuration options once the cluster is created.

☐ M10

\$0.10/hour

For production applications with sophisticated workload requirements.

STORAGE

10 GB

RAM

2 GB

vCPU

2 vCPUs

☐ Serverless

For application development and testing, or workloads with variable traffic.

STORAGE

Up to 1 TB

RAM

Auto-scale

vCPU

Auto-scale

☒ M0

Free

For learning and exploring MongoDB in a cloud environment.

STORAGE

512 MB

RAM

Shared

vCPU

Shared

✓ **Free forever!** Your M0 cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.

## Configurations

### Name

You cannot change the name once the cluster is created.

FGWeb2

### Provider

aws



Google Cloud



Azure

### Region



Hong Kong (ap-east-1)



★ Recommended



🌿 Low carbon emissions



## Quick setup

☒ Automate security setup



☐ Preload sample dataset



I'll do this later

Go to Advanced Configuration

Create Deployment

## Connect to FGWeb2Cluster ✕



You need to secure your MongoDB Atlas cluster before you can use it. Set which users and IP addresses can access your cluster now. [Read more](#)

### 1. Add a connection IP address

✓ Your current IP address (171.243.48.42) has been added to enable local connectivity. Add another later in [Network Access](#).

### 2. Create a database user

This first user will have [atlasAdmin](#) permissions for this project.

We autogenerated a username and password. You can use this or create your own.

**i** You'll need your database user's credentials in the next step. Copy the database user password.

Username

userid

Password

userid

HIDE

Copy

Create Database User

Close

Choose a connection method

## Connect to FGWWeb2



### Connect to your application



#### Drivers

Access your Atlas data using MongoDB's native drivers (e.g. Node.js, Go, etc.)



### Access your data through tools



#### Compass

Explore, modify, and visualize your data with MongoDB's GUI



#### Shell

Quickly add & update data using MongoDB's Javascript command-line interface



#### MongoDB for VS Code

Work with your data in MongoDB directly from your VS Code environment



#### Atlas SQL

Easily connect SQL tools to Atlas for data analysis and visualization



Go Back

Close

# Connect to FGWeb2Cluster



## Connecting with MongoDB Driver

### 1. Select your driver and version

We recommend installing and using the latest driver version.

Driver	Version
<div>Node.js</div>	<div>5.5 or later</div>

### 2. Install your driver

Run the following on the command line

```
npm install mongodb
```

[View MongoDB Node.js Driver installation instructions.](#)

### 3. Add your connection string into your application code

Use this connection string in your application

☐ View full code sample ☒ Show Password ⓘ

```
mongodb+srv://userid:userid@fgweb2cluster.2g1yn.mongodb.net/?  
retryWrites=true&w=majority&appName=FGWeb2Cluster
```

The password for **userid** is included in the connection string for your first time setup. This password will not be available again after exiting this connect flow.

#### RESOURCES

[Get started with the Node.js Driver](#)

[Node.js Starter Sample App](#)

[Access your Database Users](#)

[Troubleshoot Connections](#)

Go Back

Done

# Overview

## Clusters

Create cluster



FGWeb2

Connect

Edit configuration



Add data



Migrate data



Load sample data



Data modeling templates



+ Add Tag

## Connect to FGWWeb2



### Connect to your application



#### Drivers

Access your Atlas data using MongoDB's native drivers (e.g. Node.js, Go, etc.)



### Access your data through tools



#### Compass

Explore, modify, and visualize your data with MongoDB's GUI



#### Shell

Quickly add & update data using MongoDB's Javascript command-line interface



#### MongoDB for VS Code

Work with your data in MongoDB directly from your VS Code environment



#### Atlas SQL

Easily connect SQL tools to Atlas for data analysis and visualization



Go Back

Close

## Connect to FGWeb2 ✕



### Connecting with MongoDB Driver

#### 1. Select your driver and version

We recommend installing and using the latest driver version.

Driver

Version

Node.js

5.5 or later

#### 2. Install your driver

Run the following on the command line

```
npm install mongodb
```

[View MongoDB Node.js Driver installation instructions.](#)

#### 3. Add your connection string into your application code

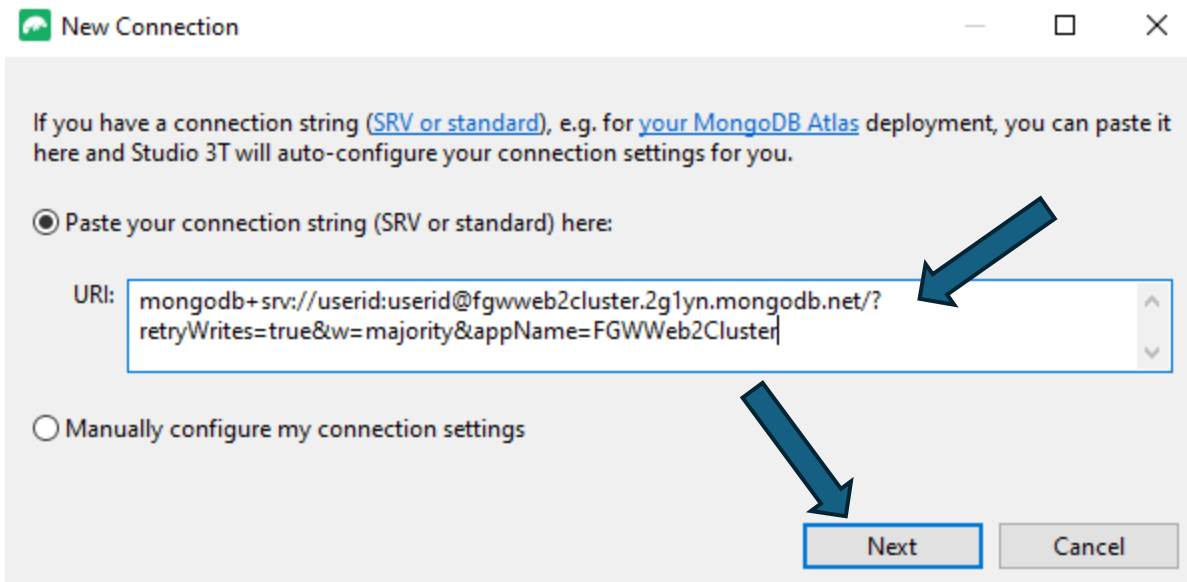
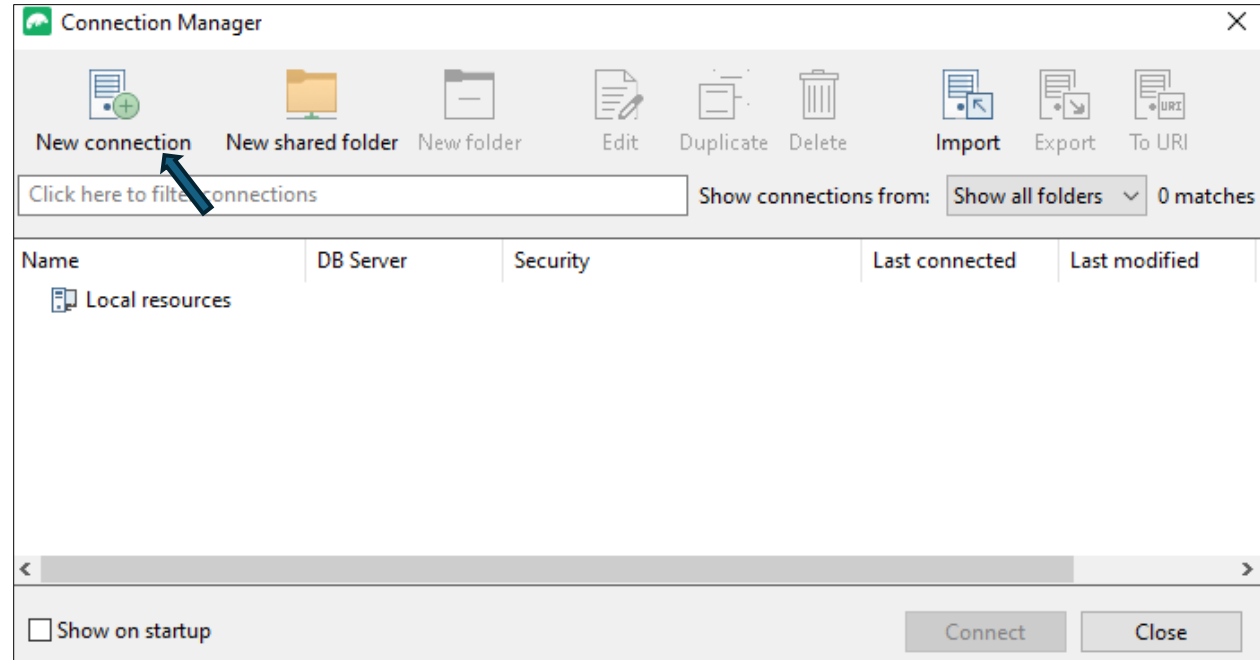
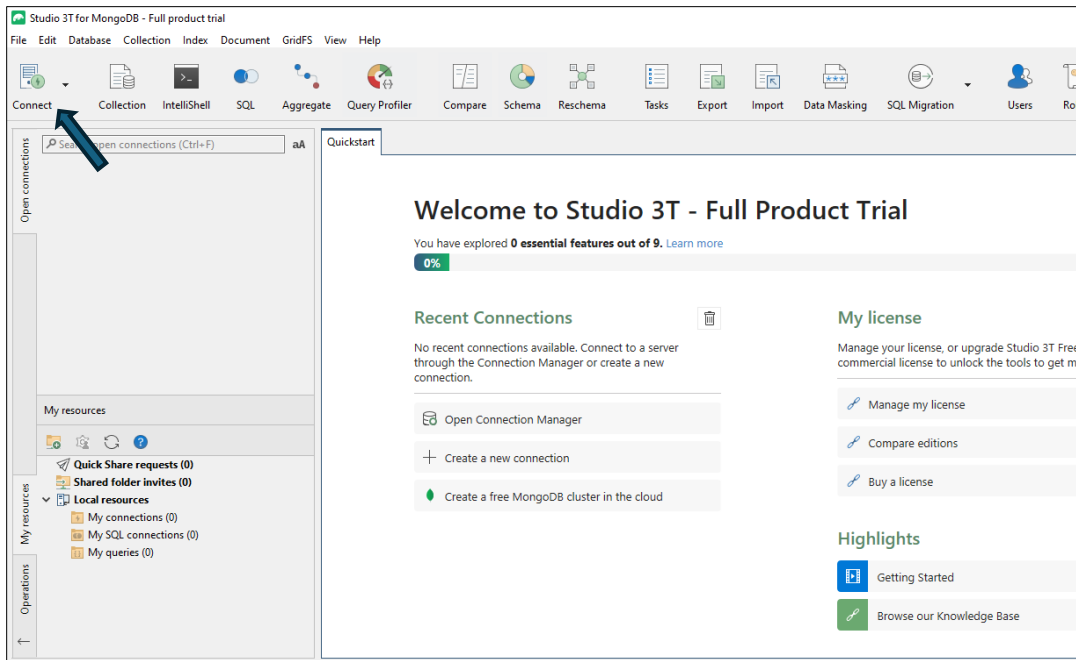
Use this connection string in your application

☐ View full code sample

```
mongodb+srv://dbw6:<db_password>@fgweb2.vvpqd.mongodb.net/?  
retryWrites=true&w=majority&appName=FGWeb2
```







New Connection

Connection name: atlas-vojlz-shard-0

Connection folder: Local resources

Server Authentication SSL SSH Proxy IntelliShell MongoDB Tools Advanced Collection History

Connection Type: DNS Seedlist (mongo+srv connections)

Server: fgwwweb2.vvpgd.mongodb.net

Enter the MongoDB [DNS Seedlist \(mongo+srv connections\)](#) server name you want to use.

New Connection

Your connection has been successfully auto-configured.

OK

☐ Read-Only Lock

From URI... Use this option to import connection details from a connection string / URI

To URI... Use this option to export complete connection details to a connection string / URI

Test Connection Save Cancel

New Connection

Connection name: atlas-huxs2o-shard-0

Connection folder: Local resources

Server Authentication SSL SSH Proxy IntelliShell MongoDB Tools Advanced Collection History

Connecting

Testing connection

atlas-huxs2o-shard-0

Operation	Status
Initializing connection	OK
Retrieving DNS Records	OK
Configuring SSL connection	OK
Connecting to server	OK
Authenticating as userid	OK
Reading server status from connection	OK
Detecting accessible databases	OK
Detecting MongoDB server version	OK
Detecting MongoDB server feature compatibility version	n/a
Connected	OK

Hide details OK Cancel

From URI... Use this option to import connection details from a connection string / URI

To URI... Use this option to export complete connection details to a connection string / URI

Test Connection Save Cancel

New Connection

Connection name: atlas-huxs2o-shard-0

Connection folder: Local resources

Server

Authentication

SSL

SSH

Proxy

IntelliShell

MongoDB Tools

Advanced

Collection History

Connection Type: DNS Seedlist (mongo+srv connections)

Server: fgwweb2cluster.2g1yn.mongodb.net

Enter the MongoDB [DNS Seedlist \(mongo+srv connections\)](#) server name you want to use.

Replica Set Name: atlas-huxs2o-shard-0

Read Preference: Primary

SRV Service Name (optional): mongodb

☐ Read-Only Lock

From URI...

To URI...

Use this option to import connection details from a connection string / URI

Use this option to export complete connection details to a connection string / URI

Test Connection

Save

Cancel

Connection Manager

New connection

New shared folder

New folder

Edit

Duplicate

Delete

Import

Export

To URI

Click here to filter connections

Show connections from: Show all folders 1 match

Name	DB Server	Security	Last connected	Last modified
Local resources				
atlas-huxs2o-shard-0	fgwweb2cluster.2...	[SSL] userid @ admin		02 Sep 2024, 13:2...

Show on startup

Studio 3T for MongoDB - Full product trial

Connect

Close

File

Edit

Database

Collection

Index

Document

GridFS

Connect

Collection

IntelliShell

SQL

Aggregate

Open connections

Search open connections (Ctrl+F)

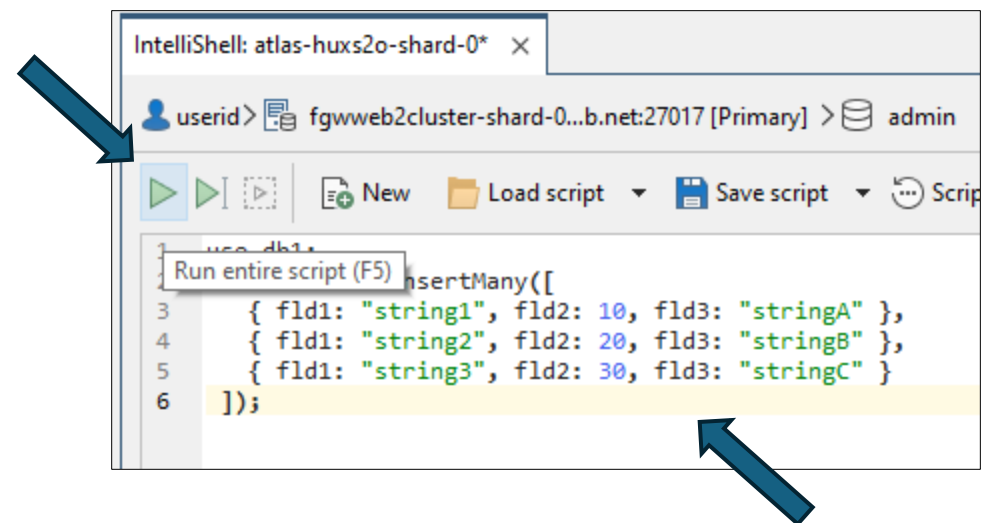
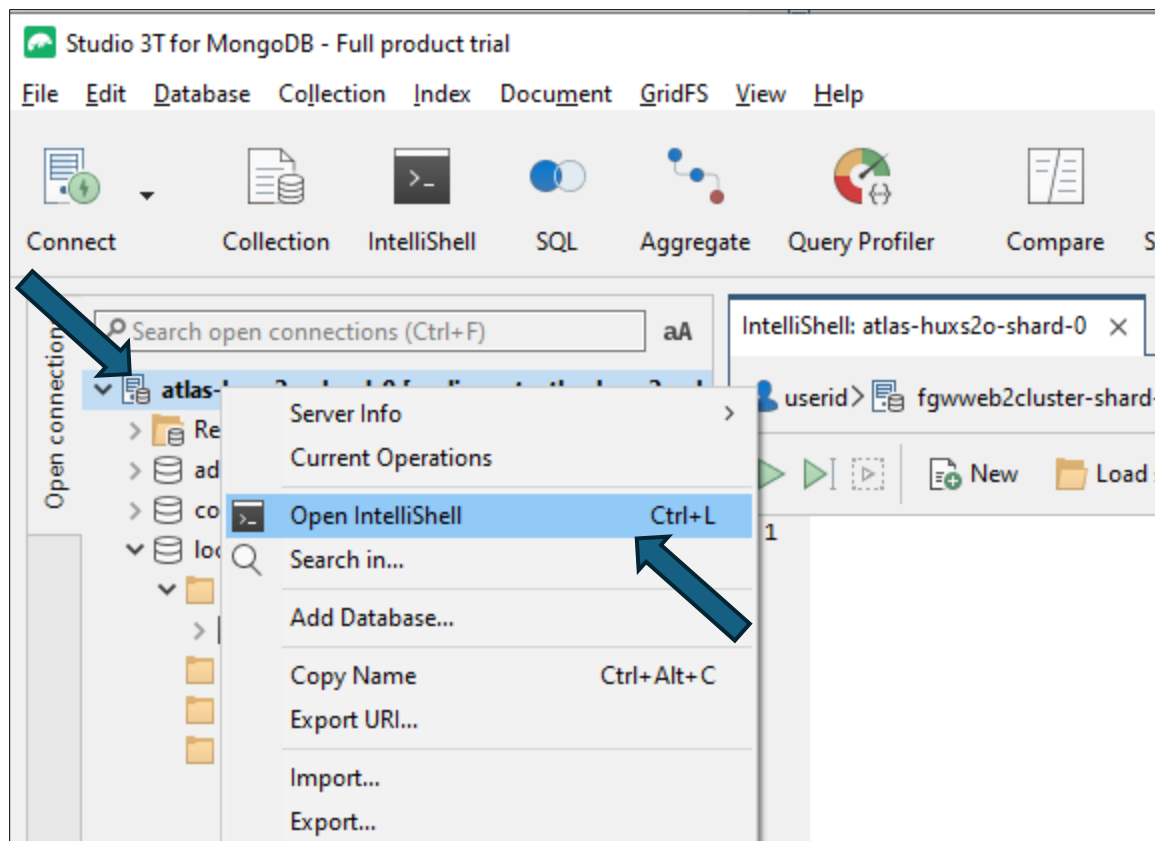
atlas-huxs2o-shard-0 [replica set: atlas-huxs2o-sh


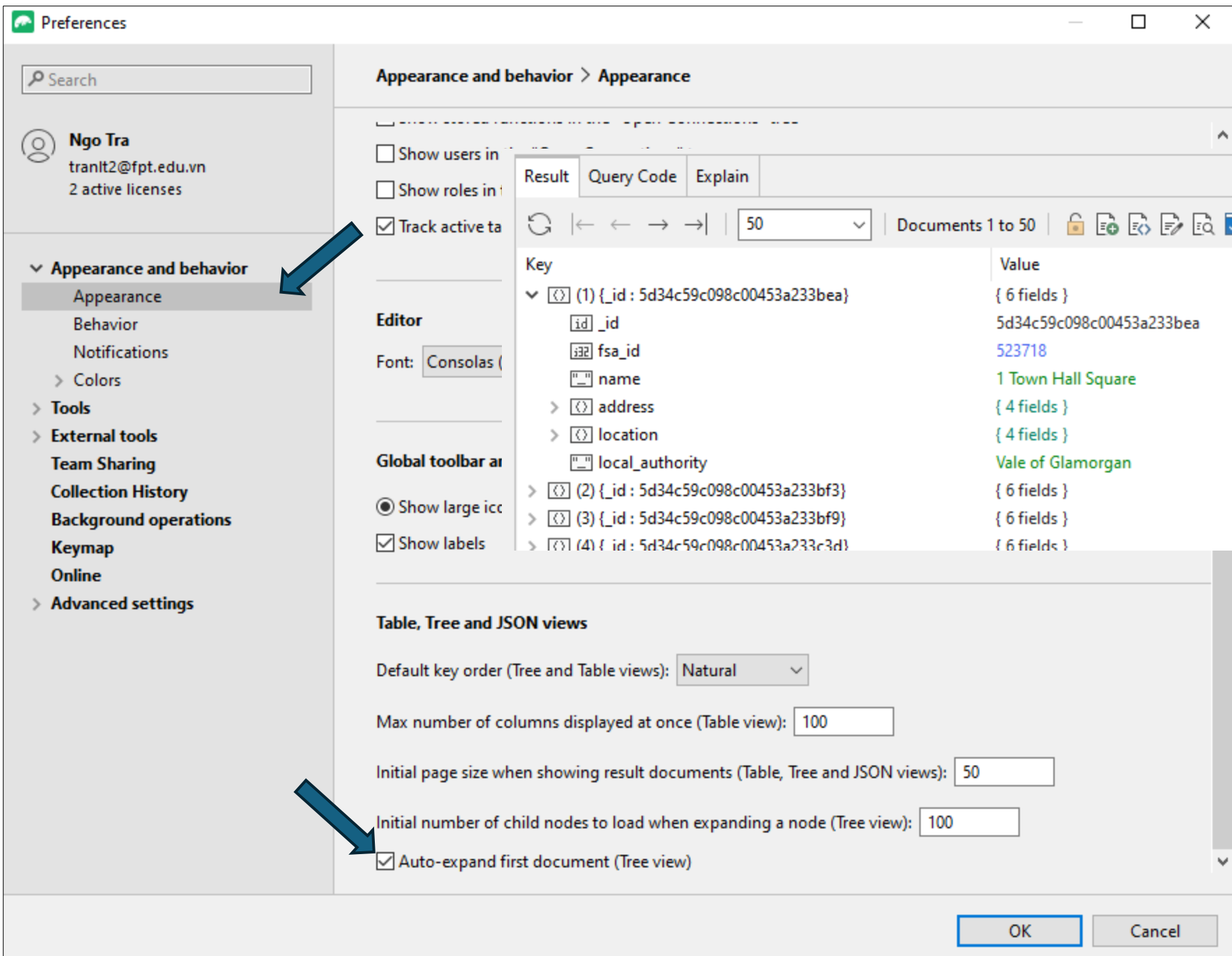
Replica Set Members

admin

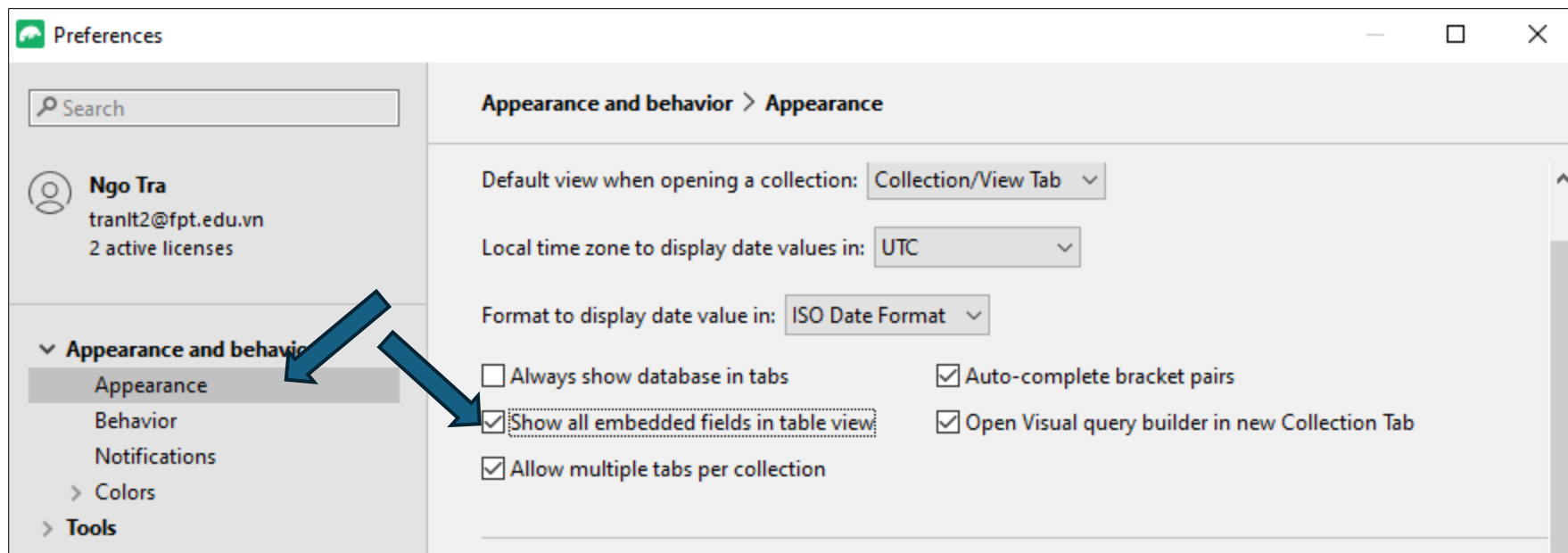
config

local





Result	Query Code	Explain
Documents 1 to 50		
Key	Value	Type
(1) { _id : 5d34c59c098c00453a233bea }	{ 6 fields }	Document
id _id	5d34c59c098c00453a233bea	ObjectId
fsa_id	523718	Int32
name	1 Town Hall Square	String
address	{ 4 fields }	Object
location	{ 4 fields }	Object
local_authority	Vale of Glamorgan	String
(2) { _id : 5d34c59c098c00453a233bf3 }	{ 6 fields }	Document
(3) { _id : 5d34c59c098c00453a233bf9 }	{ 6 fields }	Document
(4) { _id : 5d34c59c098c00453a233c3d }	{ 6 fields }	Document



Query

{ "name" : /. \*snooker.\*/i, "local\_authority" : { \$ne : "Caerphilly" } }

Projection

{ "fsa\_id": 1, "name": 1, "local\_authority": 1, "address.city": 1 }

Skip

Sort

{ "address.city": 1 }

Limit

Result

Query Code

Explain

Query

Match all (\$and)

Where

name

contains

snooker

And

local\_authority

doesn't equal

Caerphilly

Drag and drop field here or double-click

Add AND/OR group

Projection

fsa\_id

include

name

include

local\_authority

include

address.city

include

Drag and drop fields here or double-click

Sort

address.city

ascending

Drag and drop fields here or double-click

welsh\_pubs × SQL: pubs\* ×

userid> atlas-huxs2o-shard-0 (userid@fgwwweb2)

▶ Run statement

💡 Explain statement

📄 New

```
1 select local_authority, count(*)
2 from [welsh_pubs ]
3 group by local_authority
4 order by count(*) desc;
```



The **Query Code** tab shows the mongo shell code for the current aggregation. The tab can also convert the code to other languages such as JavaScript, Java, C#, Python, PHP and Ruby.

The screenshot displays the MongoDB Atlas query editor interface. At the top, the breadcrumb navigation shows the user 'userid' connected to the 'atlas-huuxs2...o-shard-0' cluster, with the database 'pubs' and collection 'welsh\_pubs' selected. Below this, a toolbar includes a 'Run' button, a 'Load query' dropdown, a 'Save query' button, and links for 'Query history', 'Copy', and 'Paste'. The main workspace is divided into two panes. The left pane, titled 'Pipeline', lists three stages: 'Stage 1: \$group', 'Stage 2: \$project', and 'Stage 3: \$sort', all of which are checked. The right pane, titled 'Operator: \$group Stage 1', shows the JSON query definition for the \$group stage: 

```
{  "_id" : {    "local_authority" : "$local_authority"  },  "COUNT(*)" : {    "$sum" : NumberInt(1)  }}
```

. At the bottom, a tabbed interface contains 'Pipeline output', 'Stage input/output', 'Query Code', and 'Explain'. The 'Query Code' tab is active, showing the MongoDB Shell code for the aggregation: 

```
1 db = db.getSiblingDB("pubs");
2 db.getCollection("welsh_pubs").aggregate(
3   [
4     {
5       "$group" : {
6         "_id" : {
7           "local_authority" : "$local_authority"
8         },
9         "COUNT(*)" : {
10          "$sum" : NumberInt(1)
11        }
12      }
13    ]
14  )
```

 A blue arrow points to the 'Language: MongoDB Shell' dropdown menu, which also has a 'Copy code' button and an 'Open in IntelliShell' button next to it.

Pipeline

+ Add stage ▾

🗑️

📄

⬆️

⬇️

☒ Stage 1: \$match

☒ Stage 2: \$group

☒ Stage 3: \$project

☒ Stage 4: \$sort

Operator: 

\$match ▾

 Stage 1 

✎ Remove stage

1 {

2 "name" : /. \*horse.\*/i

3 }

🔍

📄

⬆️

⬇️

?

Lin 3, Col 2

Pipeline output

Stage input/output 

≡

Query Code

Explain

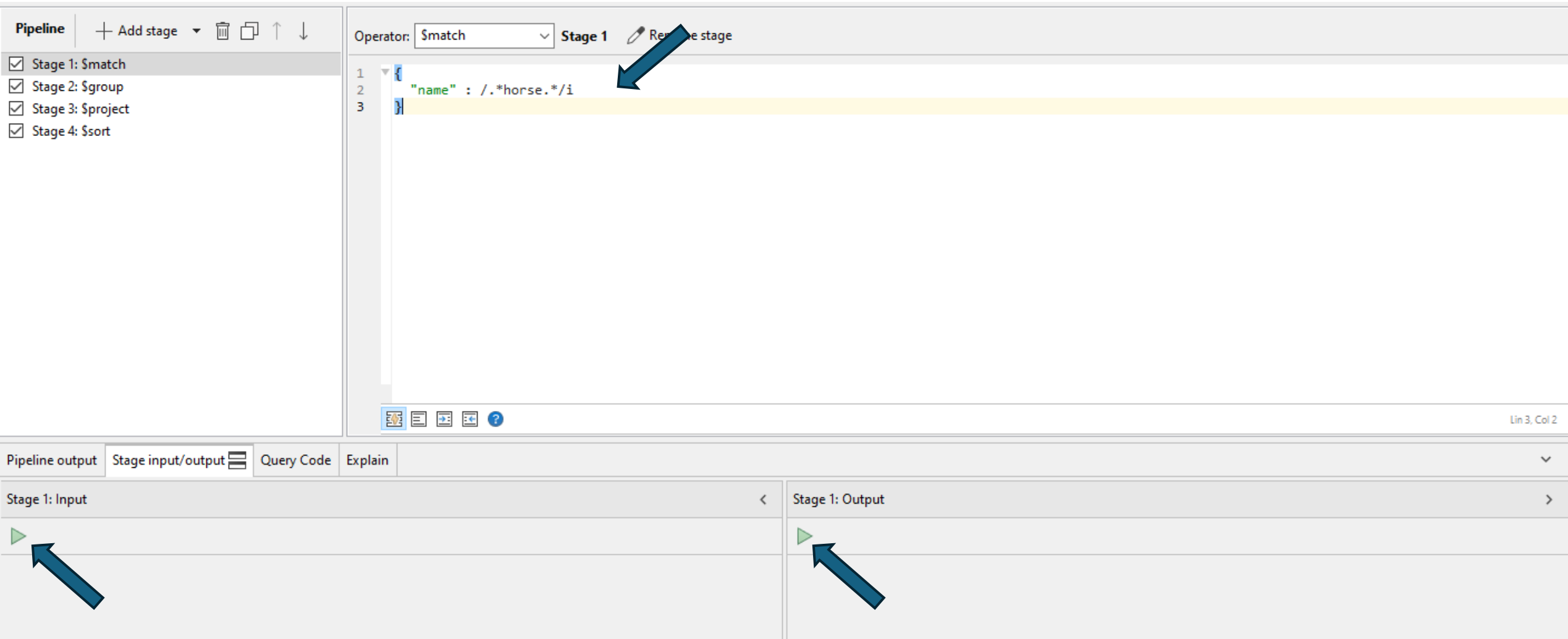
▾

Stage 1: Input

<

Stage 1: Output

>



The screenshot displays the MongoDB Atlas pipeline editor interface. On the left, a 'Pipeline' sidebar lists four stages: Stage 1: \$match, Stage 2: \$group, Stage 3: \$project, and Stage 4: \$sort, all of which are checked. The main editor area shows the configuration for Stage 1, which is a '\$match' operator. The query is defined as `{ "name" : /. *horse.*/i }`. A blue arrow points to the `"name"` field in the query. Below the query editor, there are tabs for 'Pipeline output', 'Stage input/output', 'Query Code', and 'Explain'. The 'Stage input/output' tab is active, showing the 'Stage 1: Input' and 'Stage 1: Output' sections. A blue arrow points to the 'Stage 1: Input' header, and another blue arrow points to the 'Stage 1: Output' header. The 'Stage 1: Input' section shows a green triangle icon, and the 'Stage 1: Output' section also shows a green triangle icon.

welsh\_pubs x SQL: pubs\* x Aggregation: welsh\_pubs\* x

userid> atlas-huxs2...o-shard-0] > pubs > welsh\_pubs

Run Load query Save query Query history Copy Paste

Pipeline+ Add stage Add stage Remove stage Move up Move down

☒ Stage 1: \$match

☒ Stage 2: \$group

☒ Stage 3: \$project

☒ Stage 4: \$sort

Operator: \$match Stage 1 Rename stage

```
1 {
2   "name" : /.*/horse.*/i
3 }
```

Icons

Pipeline output Stage input/output Query Code Explain

Pipeline Output

Refresh Navigation 50 Documents 1 to 16 Search

Output > amount

local_authority	amount
Powys	9
Vale of Glamorg...	7
Monmouthshire	5
Pembrokeshire	3
Wrexham	3
Bridgend	3
Carmarthenshire	3
Blaenau Gwent	2
Newport	2
Caerphilly	1
...	1

1 document selected

scottish\_pubs × british\_pubs × Export ×

Load task

COLLECTION  
Database: scottish\_info  
Collection: scottish\_pubs

Schedule

Run

General export settings

+ Add unit

Edit unit

Duplicate unit

Remove unit

Export overview

Export source

pubs > british\_pubs > Find query | ↔ Change source ▾

Load query ▾ Copy to clipboard Paste from clipboard

Export target

pubs > british\_pubs\_exported | ↔ Change ta

