Birkbeck

Cloud computing Course Work

Luiz Peres –

Documentation:

1. Setup:

* Node.js:  
  The V8 JavaScript engine in Chrome serves as the foundation for the JavaScript runtime Node.js.   
  It makes backend development accessible by enabling JavaScript to operate on the server side.  
  The Homebrew package manager was used for installation, and the following command was used to confirm the version:
* node –version
* npm –version
* Express:  
    
  Express is a flexible and lightweight Node.js framework for creating REST APIs and web applications. It makes managing routes, middleware, and HTTP requests easier.  
  Installed using:
* npm install express
* Mongoose  
    
  Mongoose is a MongoDB object data modelling (ODM) library. For modelling application data, it provides a simple schema-based approach.  
  Installed using:
* npm install mongoose
* Visual Studio Code (VS Code):  
    
  For developing and managing code, Visual Studio Code provides a robust code editor. VS Code was utilised to open the project folder using:
* code .

Project initialisation:

* A new directory named cloud-computing-cw was created to organize all project files:
* mkdir ~/cloud-computing-cw
* cd ~/cloud-computing-cw
* The project was initialized with a package.json file using:
* npm init -y
* This file tracks project dependencies and metadata.

Installed the express and mongoose packages to enable REST API development and MongoDB integration:

* npm install express mongoose

A screenshot of a computer

Description automatically generated

1. Running the API on Postman

Objective: To verify the functionality of the API by testing it in Postman.

1. Verified API Setup:

* The server was started using the command:

node server.js

* Output confirmed the server was running and connected to MongoDB:

Connected to MongoDB

Server is running on http://localhost:3000

1. Configured Postman for API Testing:

* Opened Postman and created a new request.
* Selected GET as the request method.
* Entered the API endpoint:

http://localhost:3000

1. Tested the API:

* Sent the request and received the expected response:

Welcome to the Cloud Computing Coursework API!

1. Resolved Issues:

* Addressed issues with Postman, such as:

Ensuring the Desktop Agent was running.

* Checking URL formatting to avoid invalid characters.
* Adjusting headers in Postman to prevent conflicts.

A screenshot of a computer program

Description automatically generated

Postman test

A screenshot of a computer

Description automatically generated

1. Setting Up MongoDB

Steps:

1. Add the MongoDB 6.0 repository:

echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-6.0.list

1. Update apt:

sudo apt update

1. Install MongoDB:

sudo apt install -y mongodb-org

1. Clearing previous data and fixing ownership:

sudo chown -R mongodb:mongodb /var/lib/mongodb

sudo chown -R mongodb:mongodb /var/log/mongodb

sudo rm -rf /var/lib/mongodb/\*

sudo mkdir -p /var/lib/mongodb

sudo chown -R mongodb:mongodb /var/lib/mongodb

1. Start MongoDB:

sudo systemctl start mongod

sudo systemctl enable mongod

sudo systemctl status mongod

Running the API server

Steps:

1. Navigate to the directory:

cd ~/lp-coursework-cc

1. Check NodeJS connection with MongoDB:

node server.js

Output: Server is running on <http://localhost:3000>

Connected to MongoDB

A screen shot of a computer program

Description automatically generated

Testing Locally

Steps:

1. Open a new terminal and run:

curl <http://127.0.0.1:3000/>

Response: Welcome to the Cloud Computing Coursework API!

A screenshot of a computer

Description automatically generated

Configure Firewall Rules on Google Could

Steps:

1. Log in to the GCP console.
2. Navigate to VPC network – Firewall Rules.
3. Create a new firewall rule:

Name: allow-http-3000.

Priority: Default (1000).

Direction: Ingress.

Action: Allow.

Targets: Apply to all instances in the network.

Source: 0.0.0.0/0

Protocol/Port: tcp:3000.

A screenshot of a web page

Description automatically generated

External testing with Postman and Browser

Steps:

1. In Postman, create a GET request with:

URL: <http://34.30.151.101:3000/>.

Response: Welcome to the Cloud Computing Coursework API!.

A screenshot of a computer

Description automatically generated

1. Open a browser and access the same URL: <http://34.30.151.101:3000/>.

Response: Welcome to the Cloud Computing Coursework API!.

A screenshot of a computer

Description automatically generated