

AWS re/Start Summer School 2021

IT Fundamentals (CDIS MC5101) Assignment

Auckland (NZAKL2) Due 5:00PM Friday January 21st 2022 – 150 Marks Total

Assessment Mapping

After completing this assessment, the student will have met the following learning outcomes related to the graduate profile outcome.

1. Apply basic IT concepts and commands to install, configure and use a Linux operating system.
2. Apply networking and security fundamentals to enable computer systems to communicate with each other over a network.
3. Apply programming and database concepts to setup a database and create a simple program that communicates with a database.

Assessment Instructions

- You will work individually for this assignment.

Assessment Submission Instructions

- You will use GitHub for this assignment. The instructor will create a repository for you to submit your code. Update this frequently to ensure you've always got a backup copy.

Part A – Learning Outcome 3 – 70 Marks

Task 1 - Database – 35 Marks

Use the SQLite version of the Sakila database from <https://github.com/krp/sakila-sqlite>

Use SQLite Browser (<https://sqlitebrowser.org/>) to help you create twenty (20) unique queries to display and manipulate data from each of the tables.

Hint: SELECT * FROM city; is one unique query.

- a. Edit the file named **queries.py** to add **twenty (20) Python functions** that will be used to run the queries using Python's **sqlite3** module. [20 marks]

SQL Commands:

- b. Demonstrate the user of ORDER BY in at least one (1) table. [2 marks]
- c. Demonstrate the use of WHERE in at least three (1) tables. [1 mark]
- d. Demonstrate the use of DELETE in at least one (1) table. [1 mark]
- e. Demonstrate the use of UPDATE in at least one (1) table. [1 mark]
- f. Demonstrate the use of INSERT INTO in at least 5 tables. [5 marks]
- g. Demonstrate the use of JOIN in at least two (2) tables. [5 marks]

Task 2 - Web API – 35 Marks

Create a file named **main.py** which will be used to create an API using FastAPI (<https://fastapi.tiangolo.com/> - installed with *pip install fastapi*).

- a. Use FastAPI and **main.py** to create twenty (20) routes, one for each of your queries, reusing the code from the previous section with Python's **import** statement. [10 marks]
- b. Use the HTTP GET method for queries which use SELECT statements. [5 marks]
- c. Use the HTTP POST method for queries which use INSERT INTO statements. [5 marks]
- d. Use the HTTP DELETE method for queries which use the DELETE statement. [5 marks]
- e. Use the HTTP PUT method for queries which use the UPDATE statement. [5 marks]

Hint: `@app.get("/pathname")` is how you use the HTTP GET method.

- f. Create a `/hash_password` route which accepts a password via HTTP POST and returns a hexdigest using an algorithm from Python's **hashlib** library. [5 marks]

Verify your API works by running the **uvicorn** webserver (installed with *pip install uvicorn*) and testing it with your browser at <http://localhost:8000/docs> .

Ensure you upload your finished version of `main.py` to GitHub.

Part B – Learning Outcomes 1 & 2 – 80 Marks

Deployment (Linux, Networking & Security)

Deploy your API to the cloud.

Using instructions from **52 - [NF] - Lab – Build your VPC and Launch a Web Server** from Canvas, create a VPC, Subnets, Security Groups, and then deploy a server to EC2 which uses these.

NOTE: You only need to create a single public subnet and a single private subnet. Use the last 4th and 5th digits of your student ID number as the public subnet IP and the last two digits as the private subnet address.

e.g. If your student ID number is 555**1234**

Use 10.0.**12**.0/24 as your Public Subnet and 10.0.34.0/24 as your Private Subnet.

The webserver running your API **must use the last 5 digits of your student ID number for its port.**

Use **scp** to copy the *sakila.db* file from your computer to the EC2 instance.

Task 3 - Security – 20 Marks

In the **Security.md** file in your GitHub repository add the following details.

- The SSH public key located in `~/.ssh/authorized_keys` and the algorithm it uses. [5 marks]
- The command used to change the API server port number (last 5 digits of your student ID number) which needs to be opened under security group settings. [5 marks]
- Changing your SSH server port number can reduce the amount of bruteforce attempts that spam your SSH logs. Change your SSH server port number to the first 5 digits of your student ID number. Write down the name of the file and the line you would need to change in order to change the SSH. [5 marks]
- The hashing algorithm you chose earlier for the `/hash_password` route, and depending on which you chose, list its strengths and weaknesses. [5 marks]

Task 4 - Networking – 10 Marks

In the **Networking.md** file in your GitHub repository add the following details.

- VPC Name and VPC ID: (Include your first name in the VPC name) [1 mark]
- Public & Private Subnet CIDR Blocks [1 mark]

Upload screenshots of the following as evidence that you have completed this section.

- Screenshot of VPC details [2 mark]
- Screenshots of Subnet details [2 marks]
- Screenshots of Security Group details [2 marks]
- Screenshot of Internet Gateway details [2 marks]

Upload screenshots to the repository as proof. The screenshots should match the details in **Networking.md**.

Task 5 - Shell Scripting – 10 Marks

Add the following files to your GitHub repository.

- Create a simple shell script named **install.sh** which installs the latest version of Python and the required Python libraries (fastapi & uvicorn). [5 marks]
- Create a simple shell script named **runme.sh** which runs the uvicorn webserver on a publically accessible IP address (0.0.0.0:54321) when the user types **./runme.sh** into a shell. [5 marks]

Hint: Don't forget to include the **#!** at the top of the file.

Task 6 - Linux/Cloud - 30 Marks

Set up a 2nd EC2 instance and install the Python **httpie** library on it.

Using the **http** command it installed, query one of the routes your API provides.

In the **Cloud.md** file in your GitHub repository add the following details.

- The name of the Linux distribution your EC2 instance uses. [2 marks]
- The hardware architecture your server uses (e.g. 64-bit x86, 64-bit ARM). [2 marks]
- The Amazon Machine Image (AMI) ID (e.g. ami-052432ead9b0a1234) used. [2 marks]
- The EC2 instance type (e.g. t3.xlarge) you're using. [2 marks]
- The number of vCPUs your instance has. [2 marks]
- The amount of RAM/memory (GB) your instance has. [2 marks]
- The storage size (GB) your instance has. [2 marks]
- Your VPC network id (e.g. vpc-0784bdc0c0866c695). [2 marks]
- Your Public/Elastic IP address (your server's IP address). [2 marks]
- The Private IP address of your server (begins with 10.0.). [2 marks]
- The Network Address of the subnet your server is using (Hint: Run **ip addr**). [2 marks]
- The Broadcast Address of the subnet your server is using. [2 marks]
- The Default Gateway Address configured for your server. (Hint: Run **ip route**). [2 marks]
- The client output from the **http** command. [2 marks]
- The logs displayed by **uvicorn** on the server. [2 marks]

Task 7 - Database Cloud Deployment – 10 Marks

Using the **162 - [DF] - Lab - [Challenge] Build and Access an RDS Server** lab from Canvas, deploy a database server using RDS and connect to it from the command line.

Find a way to migrate your SQLite database to your RDS database of choice.

In the **RDS.md** file, add the following details:

- A paragraph on about chosen database software (e.g. PostgreSQL, MySQL, Aurora) and why you chose it over an alternative. [2 marks]
- A link to the resource(s) you used when learning how to do this. [2 marks]
- The command used to connect a CLI client to the database. [2 marks]
- The IP address of your database and the subnet it is in. [2 marks]
- Export the sakila schema & data from this database (not-SQLite) to a **.sql** file and add it to GitHub. [2 marks]

Marking Schedule

Task	Marking Criteria	Marks
1 – Database		
a	Has 20 queries in queries.py	20
b	Uses ORDER BY	2
c	Uses WHERE	1
d	Uses DELETE	1
e	Uses UPDATE	1
f	Uses INSERT INTO (5 times)	5
g	Uses JOIN (2 times)	5
2 – Python		
a	main.py with 20 views	10
b	Uses @app.get	5
c	Uses @app.post	5
d	Uses @app.delete	5
e	Uses @app.put	5
f	Working /hash_password view	5
3 – Security		
a	Pasted SSH pubkey and algorithm it uses	5
b	Command used to change API server port	5
c	Changing SSH server port details	5
d	Hashing algorithm pros & cons	5
4 – Networking		
a	VPC Name & VPC ID in Networking.md	1
b	Public & Private Subnet CIDR Blocks	1
c	Screenshot of VPC details	2
d	Screenshot of Subnet details	2

e	Screenshot of Security Group details	2
f	Screenshot of Internet Gateway details	2
5 – Shell Scripting		
a	Working install.sh script	5
b	Working runme.sh script	5
6 – Linux/Cloud		
a	EC2 distro name	2
b	Hardware architecture	2
c	AMI ID	2
d	EC2 instance type	2
e	Number of vCPUs	2
f	RAM amount	2
g	Storage amount	2
h	VPC ID	2
i	Public IP	2
j	Private IP	2
k	Network Address	2
l	Broadcast Address	2
m	Default Gateway Address	2
n	Output from http command	2
o	unicorn logs from client request	2
7 – DB Deployment		
a	Paragraph about chosen DB software	2
b	Link(s) to resources used	2
c	Command used to connect via CLI to DB server	2
d	IP address of DB and subnet it is in (from earlier)	2
e	.sql file exported which contains schema/data.	2
TOTAL		/150