



Command reference
For cloud / database courses

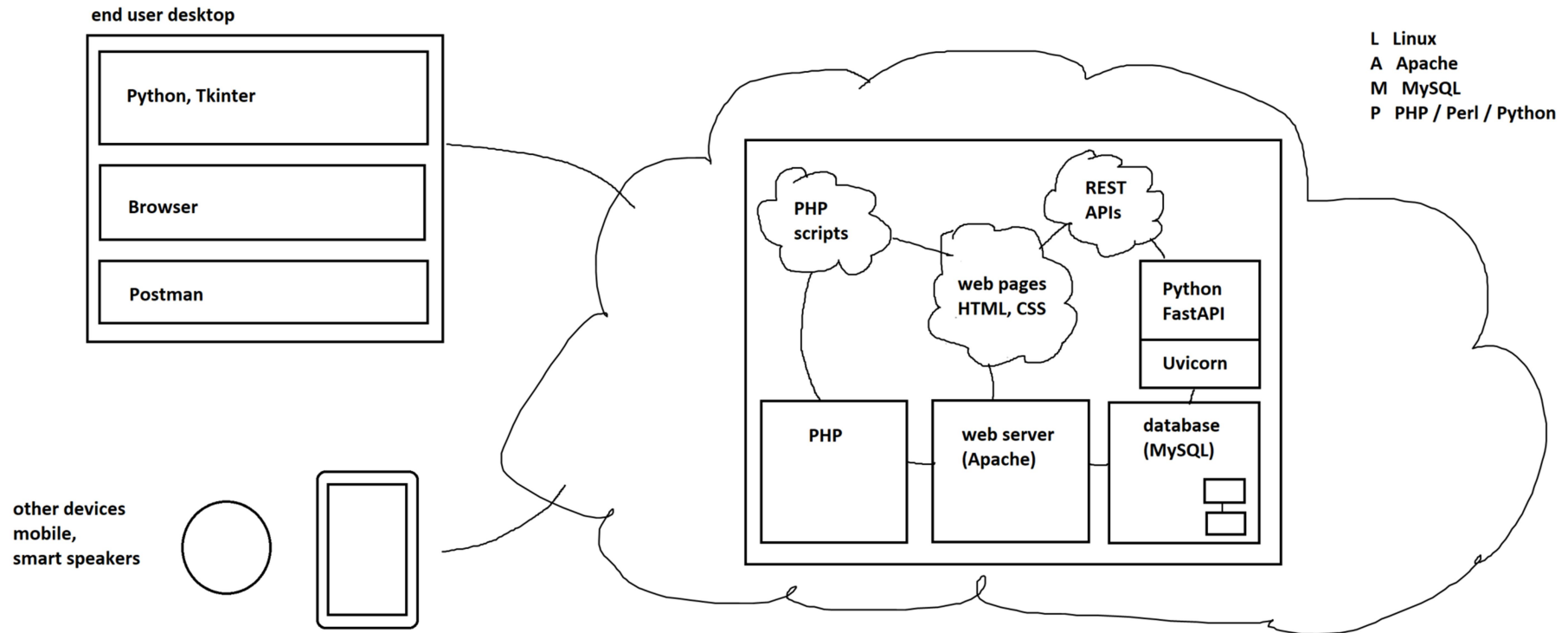
Table of contents

Contents

| | | |
|------|--|----|
| 1 | The big picture | 4 |
| 2 | HTML file structure | 5 |
| 3 | CSS styling tags..... | 7 |
| 4 | AWS S3 - steps for creating a static website..... | 8 |
| 5 | AWS EC2 - steps for creating a machine instance..... | 10 |
| 6 | Azure – steps for creating a static website | 12 |
| 7 | Azure – steps for creating a machine instance | 14 |
| 8 | Setting up Apache, MySQL, PHP | 16 |
| 9 | Copying files from the local machine to the remote machine | 22 |
| 10 | A few Linux command line commands | 23 |
| 11 | MySQL commands | 27 |
| 12 | SQL statements | 31 |
| 12.1 | Standard CRUD commands..... | 31 |
| 12.2 | Joining tables..... | 33 |
| 12.3 | Aggregate commands | 33 |
| 13 | Foreign key constraints..... | 35 |
| 14 | Creating a hello world API with FastAPI..... | 37 |

| | | |
|----|--|----|
| 15 | Create an API that returns data from a MySQL table | 40 |
| 16 | phpMyAdmin and MySQL Workbench | 42 |
| 17 | Other stuff..... | 43 |

1 The big picture



2 HTML file structure

Basic skeleton of an HTML file

```
<!DOCTYPE html>    first thing on all web pages
<html>             all contents of a page

  <header>
    <style>
      any imported styles such as fonts
      for example...
      @import url('https://fonts.googleapis.com/css?family=Bebas+Neue');
    </style>

    <title>
      title of the page on the browser tab
    </title>
  </header>

  <body>
    <h1>
      main title of the page
    </h1>

    <p>
      paragraphs of text
    </p>

    <div>
      an invisible container used for styling
    </div>
```

```
<a href="file being linked to">link text</a>  
    
</body>  
</html>
```

3 CSS styling tags

Styling tags go into `style=""` attributes on HTML tags

`text-align: center left right`

`font-size: 70pt`

`font-family: 'my font'`

`background-color: (colour name)`

`color: (colour name)`

`margin: 10px` the margin around the outside of an item

`padding: 10px` the padding inside the item

`display: inline-block` to e.g. make a row of pictures centred on the page

4 AWS S3 - steps for creating a static website

1 in VS Code - add Live Server extension

2 make a folder for creating a website, open the folder in VS Code

3 add a file called **index.html**

4 create a web page about anything you like
your favourite food
an animal
a flower
find some pictures and add them to the page

5 add some styling
change some colours, fonts, sizes
find fonts on google fonts - fonts.google.com

6 in AWS console- jump into S3

7 create a bucket

8 upload the files into the bucket

1 permissions tab - switch off block all public access

2 permissions tab - bucket policy- paste in the following:


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::PUT YOUR BUCKET NAME HERE/*"
    }
  ]
}
```

Save the changes.

3 properties tab - static website hosting - change to **enabled**
set index page to the name of your page

5 AWS EC2 - steps for creating a machine instance

1 open aws.amazon.com, sign in

2 go to EC2 service, launch instance

3 go through steps of wizard

Name : whatever you want (e.g. web server)

AMI : pick Ubuntu - make sure it is FREE TIER ELIGIBLE

Instance type : t2.micro - FREE TIER ELIGIBLE

key pair - download file to your machine

network settings - allow HTTP traffic from internet

everything else - defaults

4 click launch instance - wait for "2/2 checks passed" then the machine is ready to connect

5 copy .pem file from Downloads to a private folder on your machine- for example : c:\users\<username>\Documents\pemfiles
remember where you put it!!!! you will not be able to retrieve it again if you lose it.

6 connect to machine - on command line :

```
ssh -i <pemfile>.pem ubuntu@<public IP address>
```

You can get the public IP address from the AWS console window.

If the connection fails it is likely because the pem file is too open. Make sure it is in a folder that is accessible only by the file owner.

If your file is too open....

On Windows – in a file browser change to My Documents, make a new folder called pemfiles, copy the pem file into this folder. On Windows 11 this will probably be enough.

If it still complains - Right click the file. Pick Properties. Change to Security tab. Click Advanced. Make sure the owner of the file is yourself (change it if necessary). Remove control from everyone except yourself. Click Disable inheritance. OK everything and try again.

On Mac – open a terminal window (press Command + spacebar, type in “terminal”)

Change to the folder where you put your pem file

Use the command: **sudo chmod 400 myfile.pem** (use your own file name...)

6 Azure – steps for creating a static website

In VS Code – install Azure storage extension

Sign up for an account

Start at azure.microsoft.com

Click free account

Click start free

Get a Microsoft sign in

Get a new account

You will probably need your phone to verify your identity and a debit/credit card

Set up a static website

Start on portal.azure.com

Under Azure services – click storage accounts

Create a storage account

Choose a subscription – there should be one by default

Resource group – Create new (if there are none already) – call it e.g. MyResourceGroup

Storage account name – this must be unique across all of Azure – use something long – random words and numbers – note that this will be visible in the endpoint (URL) you use to access the site

Region – pick one close to you e.g. UK South

Performance, redundancy – leave as defaults

Go through the other pages of the wizard – review the settings but don't change anything

Click Create

Wait until deployment is complete – should take <1 minute – click refresh if it doesn't auto update

Go to storage account overview

In menu on the left – under data management – choose static website

Select enabled

Index document name – index.html

Error document name – 404.html

Save

Note primary endpoint (keep a copy of this URL)

Create a new folder on your machine called azure-workspace

Then a new folder inside that called MyStaticWebsite (or whatever)

Create and deploy the website

In VS Code...

Open the folder

Create file index.html

Make a hello world type page

Right click in blank area

Choose deploy to static website

Sign in to Azure

Select subscription

Select resource group

Message : deploying – wait until deployment complete

Browse to website using URL saved earlier

See <https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blob-static-website-host>

7 Azure – steps for creating a machine instance

Start at portal.azure.com

Pick virtual machines

Create>Azure virtual machine

Basics tab:

Choose subscription, resource group (create new resource group if there are none shown)

Enter VM name

Set region

Image – choose latest Ubuntu server

Size – pick B1S – this is the one which you can create for free

Authentication type – SSH

Username – azureuser (can replace with your own but make a note)

SSH key source – create a new key pair

Key pair name – something sensible – note default has VM name in it which is a good idea

Inbound port rules – enable 22 (SSH), 80 (HTTP) – note we can make more precisely defined rules later (which we will have to do)

Go through pages of the wizard:

Disks

Networking

Monitoring

Advanced

Tags

Review all the settings but leave everything as the defaults

Click Create

Generate a new key pair – NOTE this can only be done once!!!!!!

Click download private key and create resource

Save pem file somewhere sensible – Documents folder

Machine is created – deployment in progress

Takes 1-2 mins

Wait until your deployment is complete

Click go to resource – goes to overview page

Note public IP address

To connect to the machine – open a command prompt

Use the command : `ssh -i mykeypair.pem azureuser@public-ip-address`

Continue connecting yes

...and we're connected

See <https://learn.microsoft.com/en-us/azure/virtual-machines/linux/quick-create-portal?tabs=ubuntu>

8 Setting up Apache, MySQL, PHP

0 update everything

First thing to do once you are connected -

```
sudo apt-get update
```

1 install software

on remote machine :

```
sudo apt-get install apache2 php libapache2-mod-php mysql-server php-mysql
```

2 check the packages are installed

```
dpkg --get-selections | grep apache  
dpkg --get-selections | grep php  
dpkg --get-selections | grep mysql
```

3 restart Apache2

This will make sure that apache is up and running

```
sudo service apache2 restart
```


4 open Apache config file

```
sudo nano /etc/apache2/sites-enabled/000-default.conf
```

5 Edit the config file

find the line that begins **DocumentRoot** - add this line below it

```
AddType application/x-httpd-php .html
```

save the file - Ctrl-X, Y, Enter

6 restart Apache again

```
sudo service apache2 restart
```

7 open MySql

```
sudo mysql -u root
```

8 in MySql change the root password:

```
alter user 'root'@'localhost' identified with mysql_native_password by 'root123';  
quit
```

9 try logging in with the new password

```
mysql -u root -p
```

then quit again

10 configure MySql installation

```
sudo mysql_secure_installation
```

answer the questions as follows:

validate password component no
change root password no
remove anonymous users yes
disallow root login remotely yes
remove test database yes
reload privilege tables now yes

11 change to the Apache folder

```
cd /var/www/html
```

12 create a PHP test script

```
sudo nano test.php
```

add one line to the file:

```
<?php phpinfo(); ?>
```

save and exit : Ctrl-X, Y, Enter

13 test Apache + PHP

- on your own machine - open browser
- enter public ip address of remote machine
- should show Apache welcome page
- enter ipaddress/test.php
- should show php information

14 test MySQL

on remote machine - create a test script:

```
cd /var/www/html  
sudo nano dbtest.php
```

enter the script as follows

```
<?php
```

```
$con = mysqli_connect("localhost", "root", "root123");  
  
if(!$con)  
{  
    die("Could not connect: ".mysqli_connect_errno());  
}  
echo "Connected";  
echo "<br>";  
echo mysqli_get_host_info($con);  
echo "<br>";  
mysqli_close($con);  
echo "Disconnected";  
?>
```

save the file (Ctrl-X, Y, Enter)

15 try the database test

- on your own machine - open a browser
- enter your IP address /dbtest.php
- should show Connected + some info + Disconnected

possible errors:

if a message appears like "This page isn't working right now" (or similar) - it is probably a typo in the script

if a message appears like "Could not connect" followed by a number - it is probably a problem with MySQL

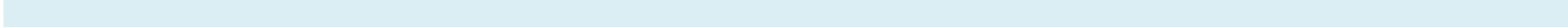
for example error 1698 is an authentication problem - the user name or password is not correct

16 change the owner of the /var/www/html folder

(this will allow us to copy files into the folder more easily)

on the remote machine: - note, **username** will be **ubuntu** on AWS, **azureuser** on Azure

```
sudo chown -R -v username /var/www/html
```



sudo - execute as root user
chown - command to change ownership of a folder
-R - recursive (acts on all subfolders as well as specified folder)
-v - verbose - more output messages
username - name of new owner
/var/www/html - name of folder whose ownership we are changing

9 Copying files from the local machine to the remote machine

```
scp -r -i <pemfile>.pem c:\<folder to copy across> username@<ip address>:/var/www/html
```

scp - secure copy command

-r - recursive (copies subfolders as well as main folder)

-i - specify identity file (pem file) to use

then the name of the source folder

then the name of the destination folder

note username will be **ubuntu** on AWS, **azureuser** on Azure

10 A few Linux command line commands

ls - lists files in a folder

ls -l long format (more detail)

ls -lS order by size (note uppercase S)

ls -lt order by date/time

ls -lh human readable file sizes

ls -r reverse order

ls -a show all including hidden files

or you can put them all together: ls -lShr

ls /usr lists files in another folder

ls /usr /etc list files in multiple folders

access rights shown when using long format:

e.g.

-rw-r--r--

First character : - for file, d for folder

Next 3 : access for file owner

Next 3 : access for members of file owner's group

Next 3 : access for everyone else

r w x : read write execute

file <filename> shows information about a file (even if it does not have a file extension) – Linux does not enforce file extensions so some files may not have an extension even if they are e.g. a picture file etc

less <filename> file viewer - pgup/pgdn, arrow keys, q to quit

nano text editor

ctrl-x exit

ctrl-x, y, enter - exits and saves the current file

cd change directory

df free disk space df -h human readable

free free memory -h again

date current date and time

cal calendar for this month

man documentation for any command

e.g. man free

pwd show current folder (this is normally shown in the prompt anyway)

cd change directory

cd /var/www/html change directly to a folder

cd .. up a folder

cd ~ go to home folder

cd - go to previous folder

mv move files

rm delete files + directories rm -r remove recursively (folder and all subfolders)

cp copy files

mkdir make a directory

`sudo` run command as root user (you will need to enter your password)

`exit` end a terminal session (e.g. disconnect from remote machine to go back to local machine)

Linux folders

`/bin` - binaries (approx. = c:\Windows\Program files)

`/boot` - linux kernel (approx. = c:\Windows\system)

`/dev` - devices

`/etc` - system config files, shell scripts, user accounts (`/etc/passwd`) approx. = registry / .ini files

`/home` - user folders approx. = Users folder

`/lib` - shared libraries (= dlls)

`/lost+found` - failure recovery - should usually be empty

`/media` - mounted drives etc on newer systems (auto mounted drives)

`/mnt` - manually mounted drives

`/opt` – “optional” additional software (commercial products etc)

`/proc` - aspects of kernel

`/root` - home folder for root account

`/sbin` - system binaries

`/tmp` - temporary files (should usually be empty or at least contain nothing permanent)

`/usr` - normal programs etc (approx. = Program Files)

`/usr/bin` - executables

`/usr/lib` - shared libraries

`/usr/local`

`/usr/sbin` - system admin programs

`/usr/share` - shared data - e.g. media, config files, icons etc

/usr/share/doc - documentation

/var – “varies” - data that is likely to change- databases, mail, web pages

/var/www/html home folder for Apache web server pages

/var/log - logs of system activity

11 MySQL commands

Launch MySQL on the command line : `mysql -u root -p`

Show all existing databases:

```
show databases;
```

Create a new database:

```
create database ____;
```

switch to a database:

```
use ____;
```

show tables in current database:

```
show tables;
```

Show the structure of a table:

```
describe _____;
```

Show the command that was used to create a table (also shows any constraints that were set up when the table was created)

```
show create table _____;
```

Create a table....

```
create table Customers(  
  cust_id integer not null auto_increment,  
  cust_name varchar(50) not null,  
  cust_address varchar(80),  
  cust_city varchar(50),  
  cust_country varchar(50),  
  cust_postcode varchar(20),  
  cust_contact varchar(50),  
  cust_added timestamp default current_timestamp on update current_timestamp,  
  primary key(cust_id)  
);
```

Another table...

```
create table AccountManagers(  
  mgr_id integer not null auto_increment,  
  mgr_firstname varchar(50) not null,  
  mgr_lastname varchar(50) not null,  
  mgr_email varchar(50),  
  mgr_phone varchar(20),  
  mgr_added timestamp default current_timestamp on update current_timestamp,
```

```
primary key(mgr_id)
);
```

A table with foreign keys...

```
create table Transactions(
  trx_id integer not null auto_increment primary key,
  trx_amount decimal(8, 2) not null,
  trx_description varchar(100),
  trx_added timestamp default current_timestamp on update current_timestamp,
  trx_cust_id integer,
  trx_mgr_id integer,
  constraint fk_customer foreign key (trx_cust_id) references Customers(cust_id),
  constraint fk_accountmanager foreign key (trx_mgr_id) references AccountManagers(mgr_id)
);
```

Modifying a foreign key constraint - to do this it is necessary to drop the current constraint and create a new one

To drop a constraint....

```
alter table Transactions drop foreign key fk_accountmanager;
```

To add a new one....

```
alter table Transactions
add constraint fk_account manager
foreign key (trx_mgr_id) references AccountManagers(mgr_id)
on update cascade
on delete set null;
```


12 SQL statements

12.1 Standard CRUD commands

Inserting a record into a table...

```
insert into Customers
(cust_name, cust_address, cust_city, cust_country, cust_postcode, cust_contact)
values(
'Big Corporation Inc',
'1 Corporation Boulevard',
'Anytown',
'Anyland',
'ABC 123',
'Alice Smith');
```

Show all the records in a table...

```
select * from _____;
```

“*” means “all columns”. Otherwise specify the columns you want...

```
select cust_id, cust_name cust_city from Customers;
```

finding records based on a condition....

```
select * from Transactions
where trx_mgr_id = 16;
```

Ordering the output...

```
select cust_name from Customers
order by cust_name;
```

Note, by default records are returned in ascending order (A-Z, 0-9). To return in descending order (Z-A, 9-0), add **desc** to the end of the **order by** part:

```
order by cust_name desc;
```

Updating a record...

```
update Transactions
set trx_mgr_id = 50
where trx_mgr_id = 16;
```

Deleting a record...

```
delete from AccountManagers
where mgr_id = 16;
```


12.2 Joining tables

Joining two tables using join...

```
select l.cust_name, l.cust_country, r.trx_amount, r.trx_description
from Customers as l join Transaction as r
on l.cust_id = r.trx_cust_id
where r.trx_amount > 9000.00 and l.cust_country = 'Brazil'
order by r.trx_amount desc;
```

Note that using the join keyword by itself indicates an inner join (the most common type of join)

Joining three tables using where...

```
select c.cust_name, c.cust_country, t.trx_amount, t.trx_description, a.mgr_firstname, a.mgr_lastname,
a.mgr_email
from Transactions as t, Customers as c, AccountManagers as a
where t.trx_cust_id = c.cust_id and t.trx_mgr_id = a.mgr_id
and c.cust_country = 'Vietnam'
order by t.trx_description;
```

12.3 Aggregate commands

Example of count, sum, avg, max, min and group by

Also note where doesn't work on aggregates- use having instead

```
select l.mgr_id, l.mgr_firstname, l.mgr_lastname,
count(r.trx_amount) as number,
sum(r.trx_amount) as total,
avg(r.trx_amount) as average,
```

```
max(r.trx_amount) as highest,  
min(r.trx_amount) as lowest  
from AccountManagers as l join Transactions as r  
on l.mgr_id = r.trx_mgr_id  
where mgr_id <= 10  
group by l.mgr_id  
having total > 250000  
order by lowest;
```

A second example - this would return the list of account managers with the person with the highest total value of transactions at the top:

```
select l.mgr_id, l.mgr_firstname, l.mgr_lastname, count(r.trx_amount), sum(r.trx_amount) as total  
from AccountManagers as l join Transactions as r  
on l.mgr_id = r.trx_mgr_id  
group by l.mgr_id  
order by total desc;
```

13 Foreign key constraints

Foreign key constraints can be added when a table is created

To add a constraint afterwards, drop any existing constraint and add a new one using alter table (see the MySQL commands section for an example)

Constraints can be set on update or on delete:

```
on update restrict
on update cascade
on update set null

on delete restrict
on delete cascade
on delete set null
```

Note, if the constraints are not specified, **restrict** is the default for both **on update** and **on delete**.

If there are e.g. 1000 customers and 20 account managers

this query will work

```
insert into Transactions(trx_amount, trx_description, trx_cust_id, trx_mgr_id) values(123.45, 'Purchase of carrots', 55, 44);
```

this one will fail a constraint

```
insert into Transactions(trx_amount, trx_description, trx_cust_id, trx_mgr_id) values(123.45, 'Purchase of carrots', 55, 120);
```

so will this one

```
insert into Transactions(trx_amount, trx_description, trx_cust_id, trx_mgr_id) values(123.45, 'Purchase of carrots', 5555, 120);
```

This one will fail too

```
update customer set cust_id = 99999 where cust_id = 2;
```

14 Creating a hello world API with FastAPI

Installing FastAPI - on AWS machine:

Note, the command to install Pip is only necessary if you don't already have Pip. Try `pip --version` or `pip3 --version` on a command line to see if the tool is present.

```
sudo apt-get install python3-dev
sudo apt install python3-pip
pip install --upgrade setuptools
pip install starlette
pip install fastapi
pip install "uvicorn[standard]"
pip install pydantic
```

Note if you get error messages about the wrong distro information, run this command:

```
pip install --upgrade --user setuptools==58.3.0
```

Then try the installation again

After the installation is done, reboot

```
sudo reboot
```

Try Uvicorn

```
uvicorn --version
```

This should report the version number

If the command is not recognised try the full path of Uvicorn:

```
/home/ubuntu/.local/bin/uvicorn --version
```

Note - this should not really be necessary- Uvicorn should be added to your system path during installation.

Create a file called helloworldapi.py:

```
from fastapi import FastAPI
app = FastAPI()
@app.get("/hello")
def api_hello():
    return {"Message": "Hello world"}
```

You could create this directly on the AWS machine or on your own machine.

If you create it on your own machine you will need to transfer it to the AWS machine using scp:

(replace the pem file, folder name and IP address with your own)

```
scp -r -i c:\Users\User\Documents\pemfiles\monday.pem c:\MyBankAPI ubuntu@18.170.26.194:/var/www/html
```

Open port 8000 on the AWS machine (port 8000 is Uvicorn's default port so it needs to allow incoming traffic)

To do this:

Open a browser, go to aws.amazon.com, log in

Go to the EC2 service, find your machine, open the machine's properties

Click the security group

Add a new incoming rule: TCP traffic, on port 8000, from anywhere. Add "Uvicorn traffic" or similar as a description

Save everything

On the AWS machine run Uvicorn

```
uvicorn helloworldapi:app --host 0.0.0.0
```

Open a browser and enter the URL of the API

```
http://your_ip_address:8000/hello
```

This should show the hello world message

If the message doesn't appear, check the Uvicorn screen for any error messages

If there are no error messages, but the browser cannot connect to the API, it is probably a problem with the AWS security settings

15 Create an API that returns data from a MySQL table

This API uses the PyMySQL library. Install the library on the AWS machine:

```
pip install pymysql
```

Make sure it is installed:

```
pip list
```

Look for PyMySQL in the output.

In VS Code on your local machine create a file called main.py

Code for the API - replace the MySQL username/password/database name with your own:

```
from fastapi import FastAPI
import pymysql

app = FastAPI()

@app.get("/customer")
def api_customer(n : int):
    con = pymysql.connect(host="localhost", user="root", password="root123", database="MyBank")
    cur = con.cursor()

    sql = f"select * from Customers where cust_id = '{n}'"
    cur.execute(sql)
```



```
cust = cur.fetchone()

cur.close()
con.close()

return {"cust_id": cust[0],
        "cust_name": cust[1],
        "cust_address": cust[2],
        "cust_city": cust[3],
        "cust_country": cust[4],
        "cust_postcode": cust[5],
        "cust_contact": cust[6],
        "cust_dateadded": cust[7]}
```

Transfer the file to the AWS machine (use your own details)

```
scp -r -i c:\Users\User\Documents\pemfiles\monday.pem c:\MyBankAPI ubuntu@18.170.26.194:/var/www/html
```

On the AWS machine, start Uvicorn

```
uvicorn main:app --host 0.0.0.0 --reload
```

Open a browser and call the API using a valid customer number (use your own details):

```
http://18.170.26.194:8000/customer?n=222
```

Make sure the customer data is returned correctly.

16 phpMyAdmin and MySQL Workbench

These are alternatives to using the standard MySQL command line interface

In Ubuntu on the AWS machine phpMyAdmin can be installed using the command

```
sudo apt-get install phpmyadmin
```

During installation it will prompt you for which web server to use - pick Apache

It will also ask you if you want to configure a database (for phpMyAdmin itself to use) - pick yes

Enter your root password and confirm

Note, the installation is also a command line interface- you can't use the mouse- use Tab to move between options and press Space to select an option (a * appears when an option is selected)

Once it is installed it can be accessed from <http://your ip address/phpmyadmin>

MySQL Workbench can be installed from <https://www.mysql.com/products/workbench/>

It works on Windows, Linux and Mac

17 Other stuff

Website for generating test data:

<https://www.mockaroo.com/>

Website for royalty free pictures:

<https://pixabay.com/>

Website for checking JSON is correctly formatted

<https://jsonlint.com/>