AWS – Project management in the Cloud

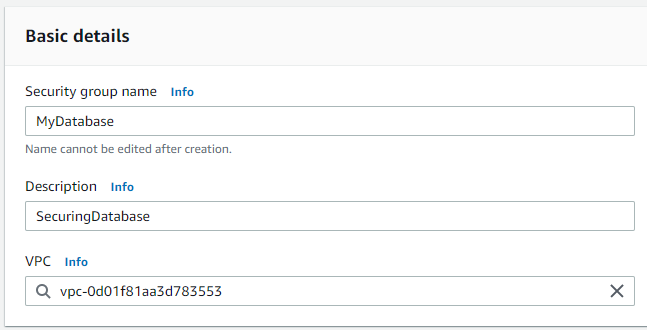
|  |
| --- |
| **Learning Outcomes** |
| At the end of this practical lesson, students will be able to:   * Setup the Python Flask Web Application environment * Setup the AWS environment * Setup/Configuration * Going remote |

**Step 1:** at the top right of your screen click on the dropdown button on the left of your “account” dropdown button, take note of your selected region.

**Step 2:** in your search bar search and select “EC2” amazon services.

**Step 3:** at the left sidebar, click on “Network & Security” dropdown button and select “Security Groups” to create two Security Groups, one for Website and another for Database. Click on “Create security group” and name as “MyWebApp” and “MyDatabase”.

Graphical user interface, text, application, email

Description automatically generated 

(MyWebApp)

Graphical user interface, application

Description automatically generated

(MyDatabase)

A screenshot of a computer

Description automatically generated

**Step 4:** at the left sidebar, click on “Instances” dropdown button and select “Instances” to create two instances, one for Website and another for Database. Click on “Launch instances” and name the two servers as “MyWebApp” and “MyDatabase”.

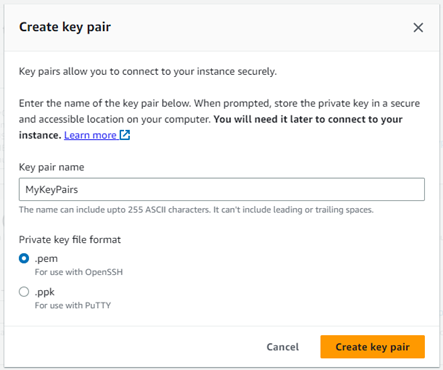
1. “MyWebApp” Server
   1. Select “Windows Microsoft” as your Amazon Machine Image Operating System
   2. Under Key pair (login) select “Create new key pair”. Key pair name as “MyKeyPairs”, file format as ‘.pem’. and click “Create key pair”.

Graphical user interface, text, application

Description automatically generated 

* 1. Select “Select existing security group”. Click on the dropdown button and select “MyWebApp” as the website Security Group.
  2. Select “Launch instance” on the right of the screen.

1. “MyDatabase” Server
   1. Select “Amazon Linux” as your Amazon Machine Image Operating System
   2. Under Key pair (login) select “Create new key pair”. Key pair name as “MyKeyPairs”, file format as ‘.pem’. and click “Create key pair”.

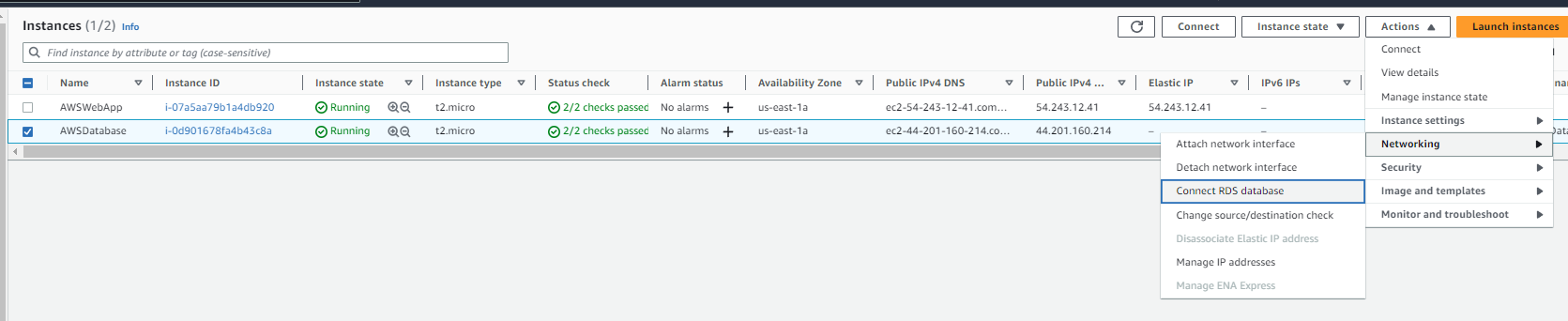
* 1. Select “Select existing security group”. Click on the dropdown button and select “MyDatabase” as the website Security Group.
  2. Select “Launch instance” on the right of the screen.

**Step 5:** in your search bar search and select “RDS” amazon services.

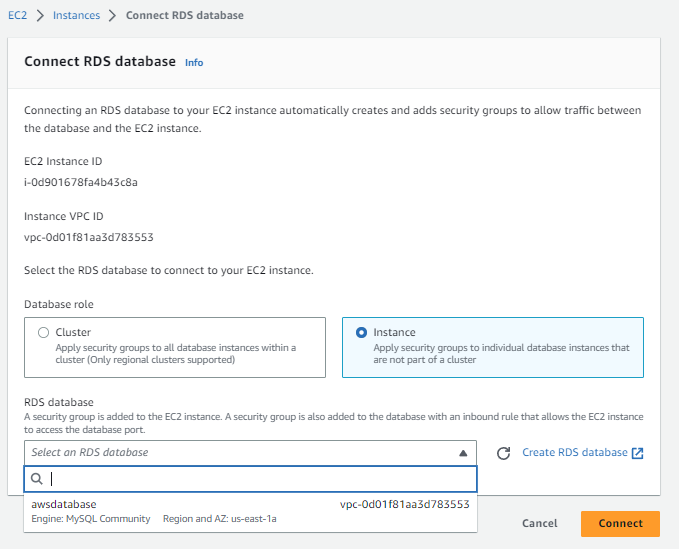
**Step 6:** at the left sidebar, click on “Databases” to create one database. Click on “Create database” and name as “mydatabase”.

1. Under “Engine options”, select “MySQL” (referring to MySQL Workbench)
2. In “Template” select “Free tier”
3. In “Settings”, under “DB instance identifier” input “mydatabase”
4. In “Credentials Settings”
   1. under “Master username” input “admin”
   2. under “Master password” input “Pa$$w0rd”
5. In “Connectivity”
   1. Under “Compute resource” select “Don’t connect to an EC2 compute resource”
   2. Under “Public access” select “Yes”
   3. Under “VPC security group (firewall)” select “Choose existing”
   4. Under “Existing VPC security groups” select “MyDatabase”
6. Select “Create database”

**Step 7:** go back to “EC2” service and connect the selected instance, “MyDatabase” instance with “mydatabase” RDS.

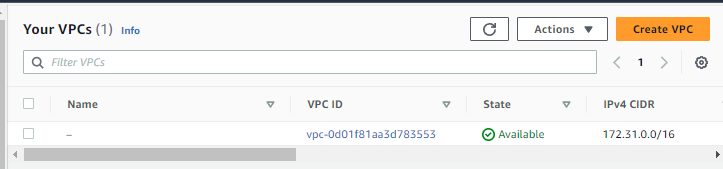


1. Select “Instance” and click on the dropdown arrow and select “awsdatabase” and “Connect”

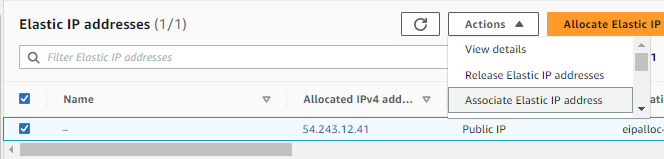


**Step 8:** at the left sidebar, click on “Network & Security” dropdown button and select “Elastic IPs” to create a public ip address and associate it with “MyWebApp” so that the public can access the website which is by default created in the VPC private ip address range.

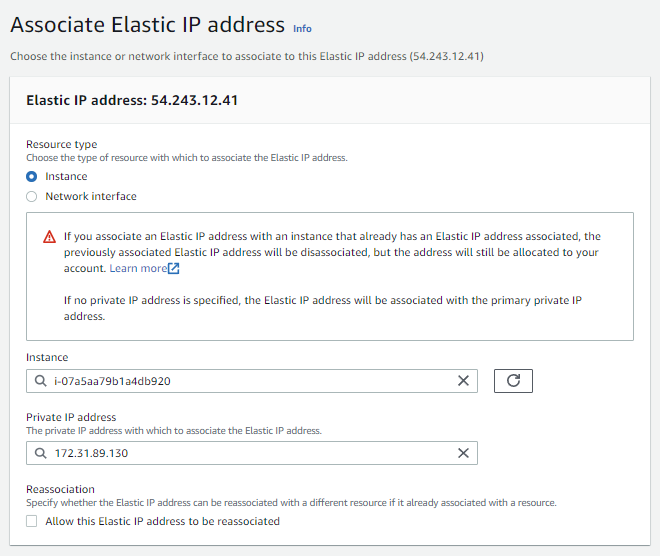
1. In order to know if your VPC is public or private, search for “VPC” service and select “Your VPC” at the left sidebar under “Virtual private cloud” dropdown arrow.



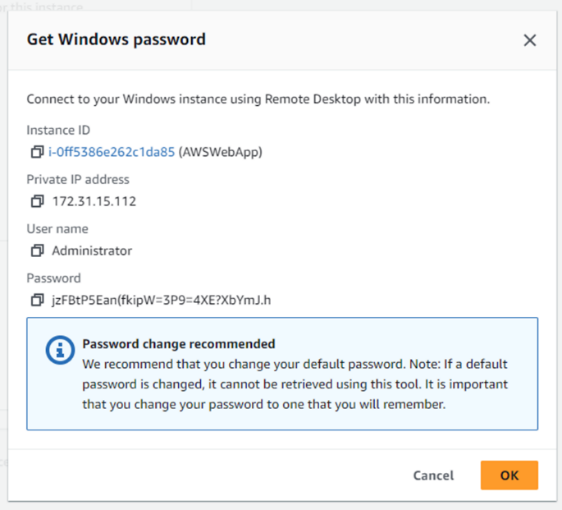
1. Back to “Elastic IPs” select “Allocate Elastic IP address”, select the created public address and select “Associate Elastic IP address”



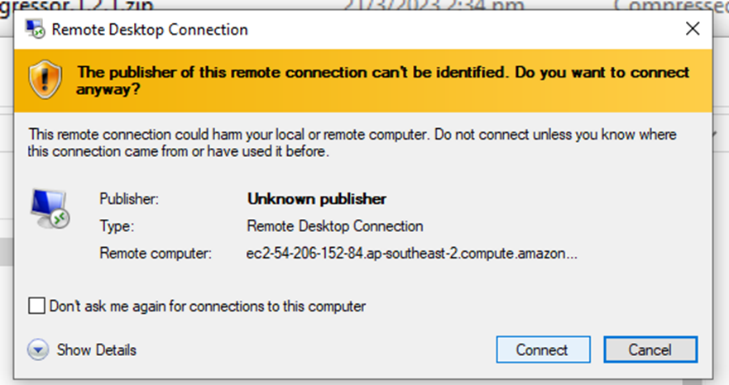
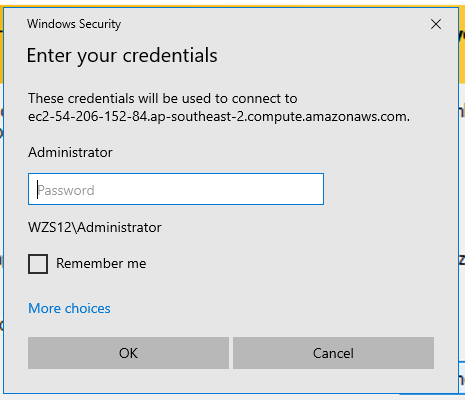
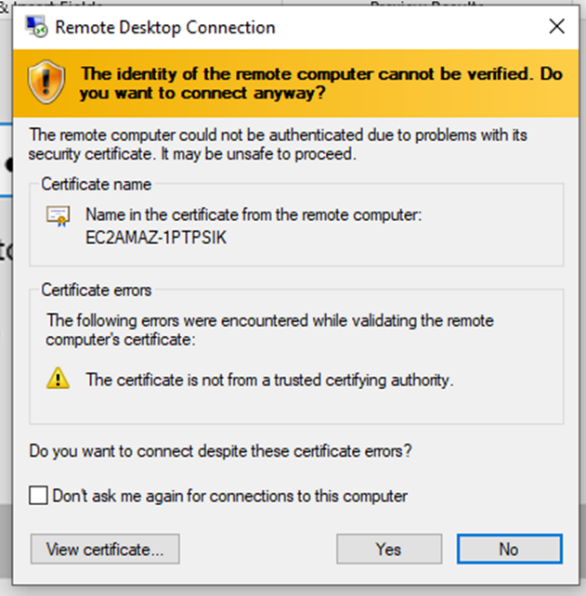
1. Under “Instance” and “Private IP address”, select “MyWebApp” and its private address and click “Associate”.



**Step 9:** go back to where we create our instances, and select “MyWebApp” to connect. Select “RPP client” and download remote desktop file. To get the password in order to connect to the remote host, select “Get password”. The private key file that we are supposed to upload is “MyKeyPairs.pem” file to decrypt and get the password.



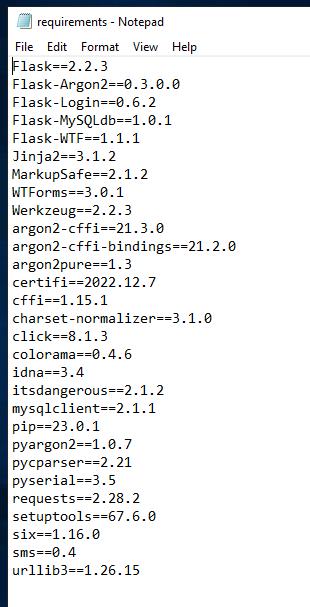
**Step 10:** open the downloaded “MyWebApp.rdp” file and click connect. It will show you a pop up screen requiring you enter the password you copied. And click “okay” and lastly “yes”.

**Step 11:** after rdp have loaded, in the windows platform, download Python 3.(X - 1).X (newest version) and GitHub Desktop. The windows firewall advanced setting should have public profile inbound rule as “allow”

1. For python, remember to “Add Python to PATH”
2. Clone your repository file which contains the website you want to run
3. Open command prompt and do the following commands
   1. pip install --upgrade pip
   2. pip install --upgrade wheel
   3. cd C:\Users\Administrator\Documents\GitHub\GitShare (where your app.py is located)
   4. pip install -r requirements.txt

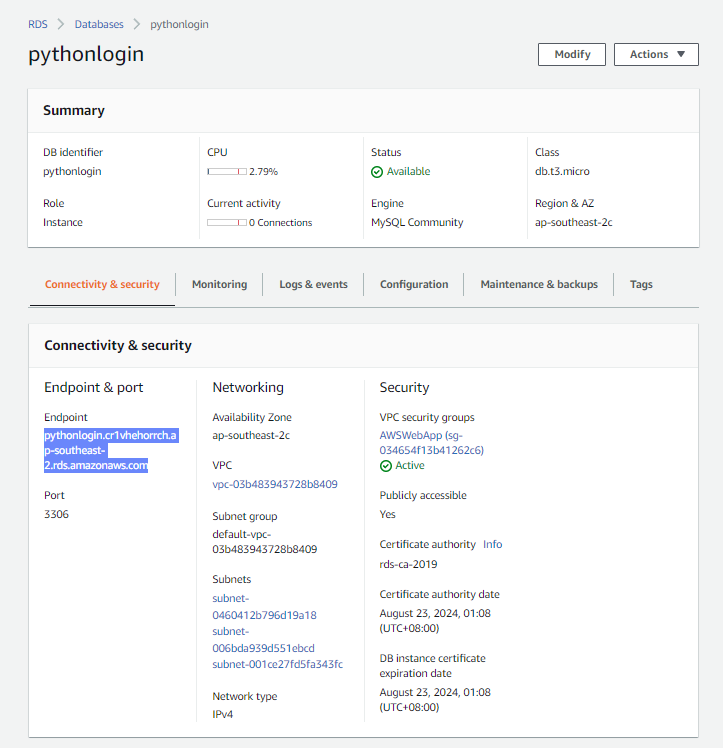
the txt should look like this with all the packages you needed. The file must be placed in the same directory as your app.py



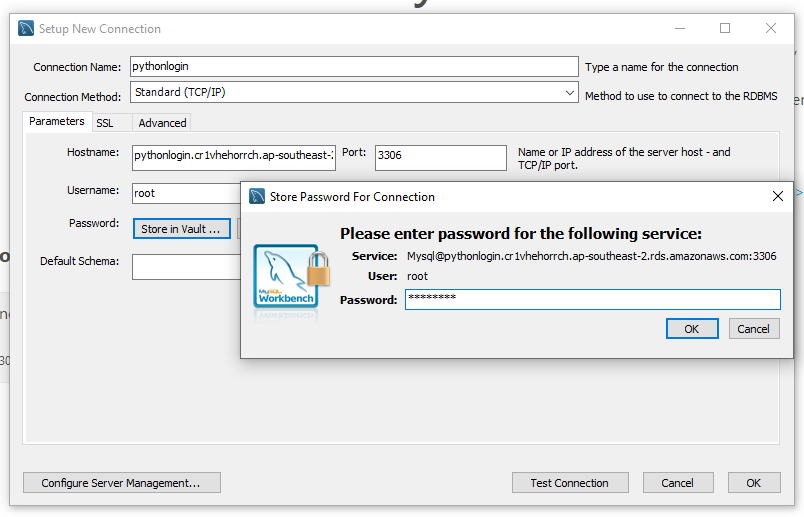
* 1. for packages that have an error, download them manually by running the command “py app.py” to check for missing packages

**Step 12:** to connect MySQL Workbench with AWS RDS services, setup a new connection.

1. Under connection name input “mydatabase”
2. Under hostname, copy the endpoint in RDS database “mydatabase” and put into the box.



1. Username as “admin”
2. And press Test Connection
3. A pop up will appear asking you to input the password “Pa$$w0rd” in order for the connection to be successful

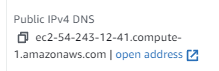


**Step 13:** Next, check that your RDPs app.py have this line.



1. If yes, in the command prompt, run the following command
   1. cd C:\Users\Administrator\Documents\GitHub\GitShare (where your app.py is located)
   2. py app.py

Finally, go and select your “MyWebApp” EC2 and click “open address” under “Public IPv4 DNS”



Change https to http and your webpage will load

