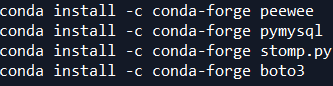
網路程式設計 HW5-report 0410024 余東儒

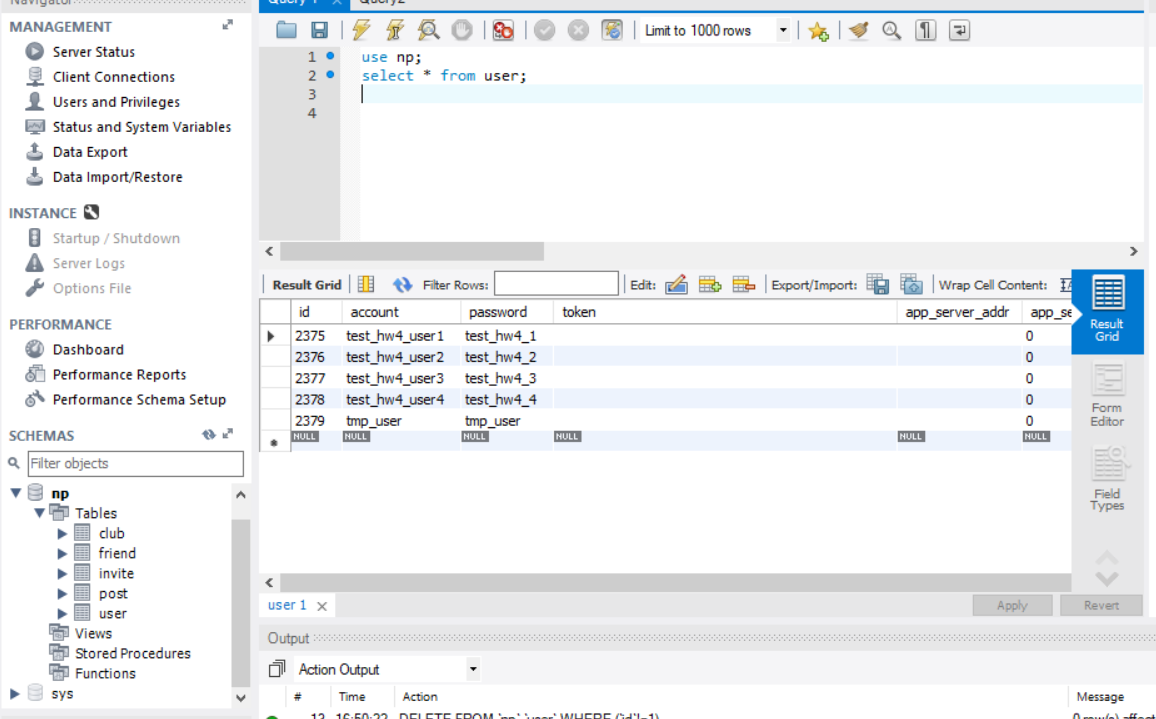
一、環境設定

首先先辦一個AWS的帳號，然後創建一組access key，再來就是在先開一個ec2的 instance，我這邊是用ubuntu的OS，然後需要設定的地方為security group，在inbound跟outbound我設定為All traffic 都可以access。創立好instance之後，用ssh連上instance，下載我需要的套件，我用anaconda作為套件管理，下載peewee、pymysql、Stomp和boto3，另外再載activemq的安裝程式並安裝，另外也有下載AWS的CLI，用-aws configure這個指令將一開始下載的access key輸入，這樣才能連到我的aws account。



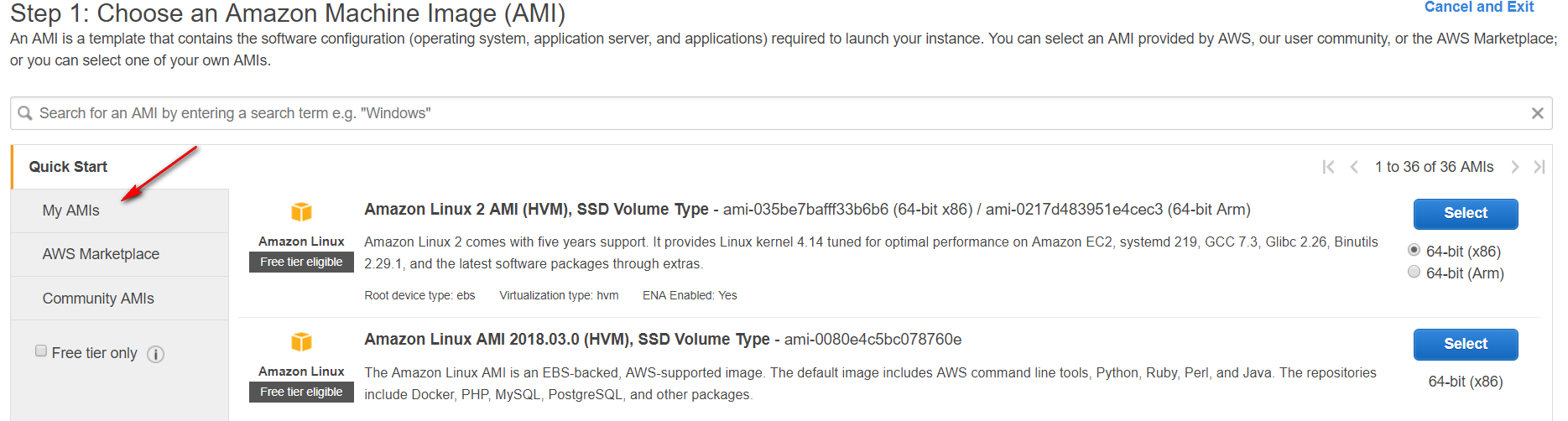
接下來將我的login\_server 和app\_server分別丟到兩個instance，再來將兩個instance分別做成AMI，以供之後程式創立instance用，

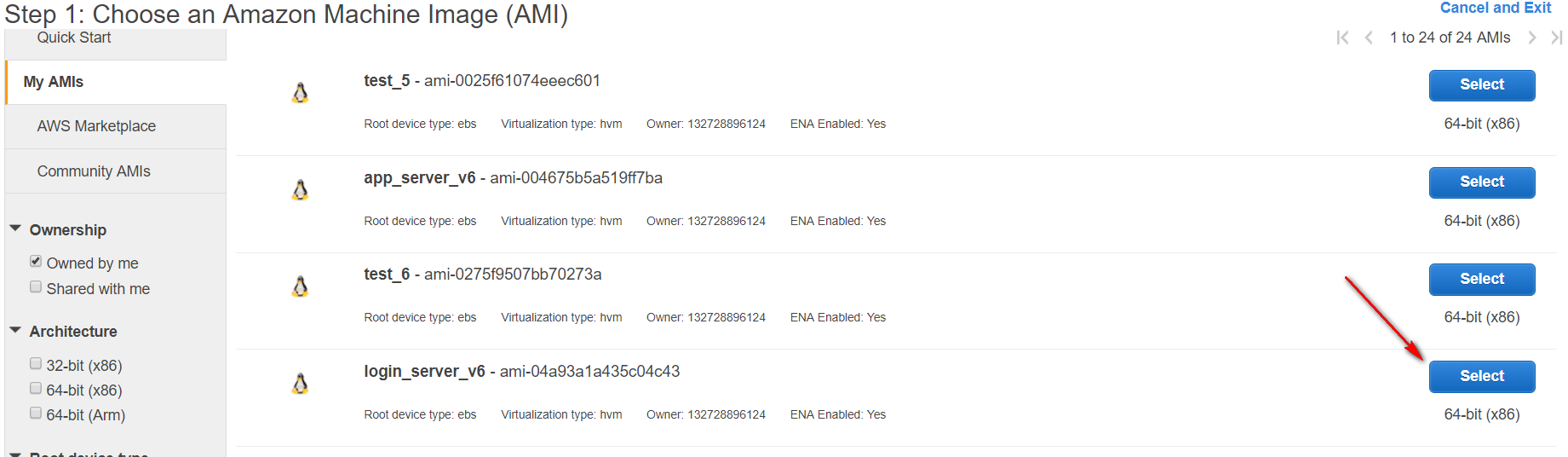
我的database也是放在AWS上，我在RDS上create一個MySQL的database，然後用workbench連線到AWS的MySQL，進行database的建立。



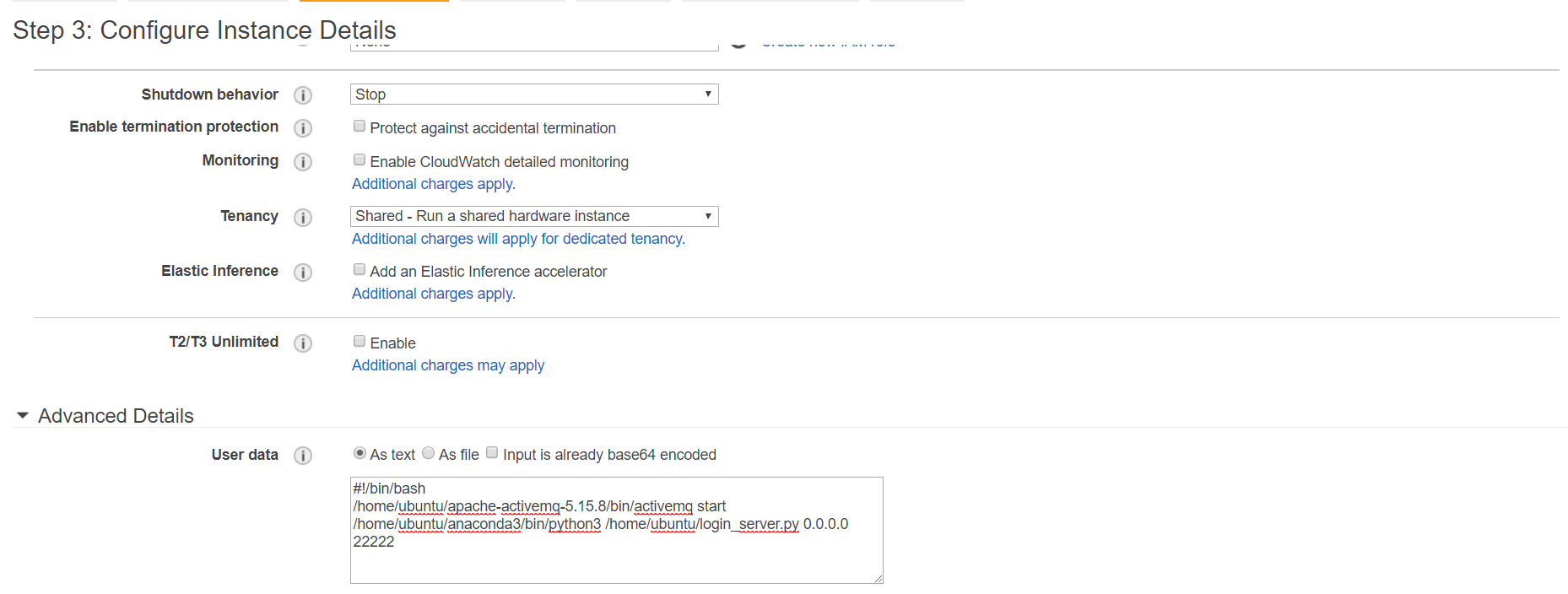
二、程式執行方式

Login\_server的部分我直接在aws的console create，選擇MY AMIS，選擇先前創立好給login\_server的AMI，

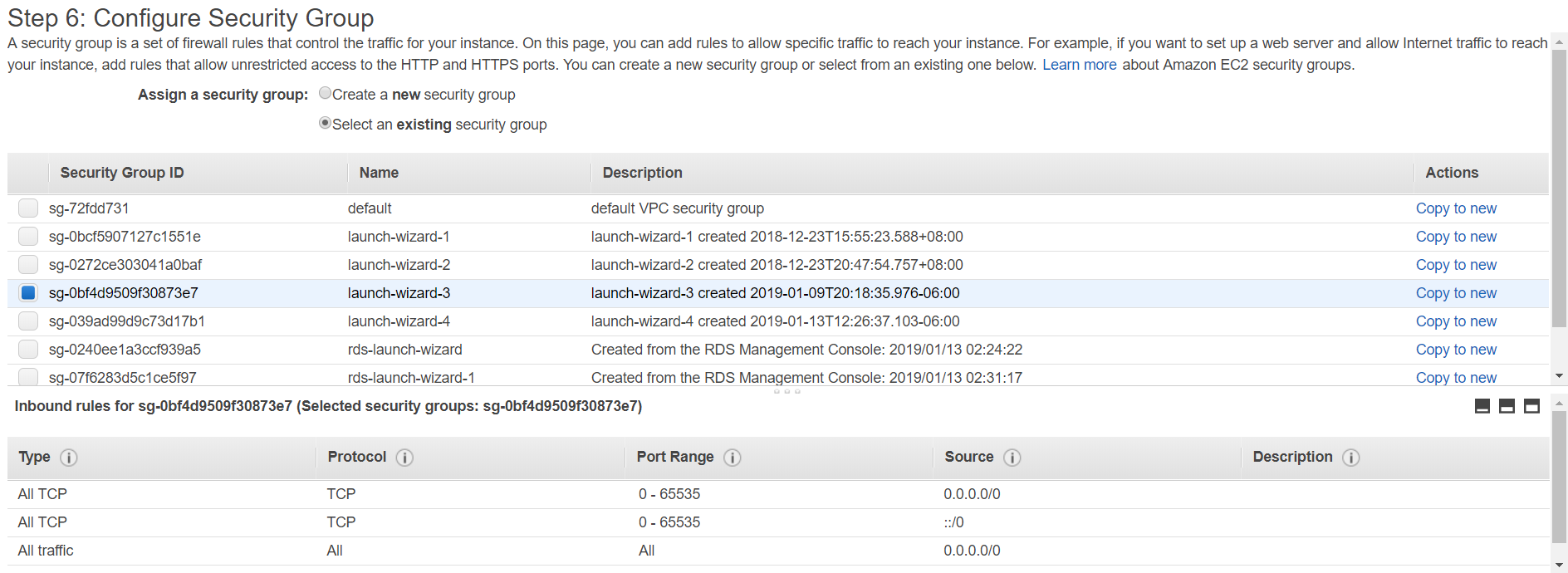




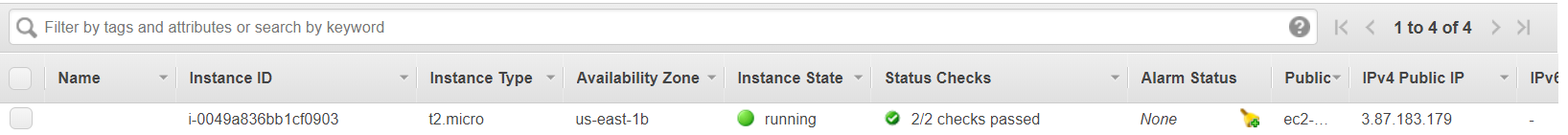
再來在step 3這裡的User data輸入我們在開機時要的指令，這邊第一行指令是將activemq開啟，第二行則是執行我的login\_server python檔，ip為0.0.0.0是因為要開放給全部人連線。



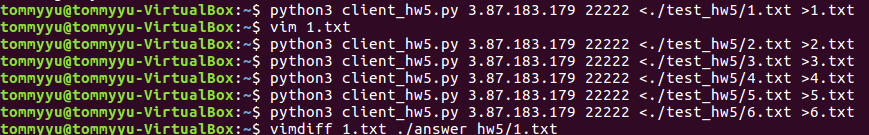
在Step 6這邊則要選擇剛剛創立好的security group，這樣才能讓其他人連到這台server。



最後創建好的這個instance就是我的login\_server，記下public IP以供我的client能夠連線。

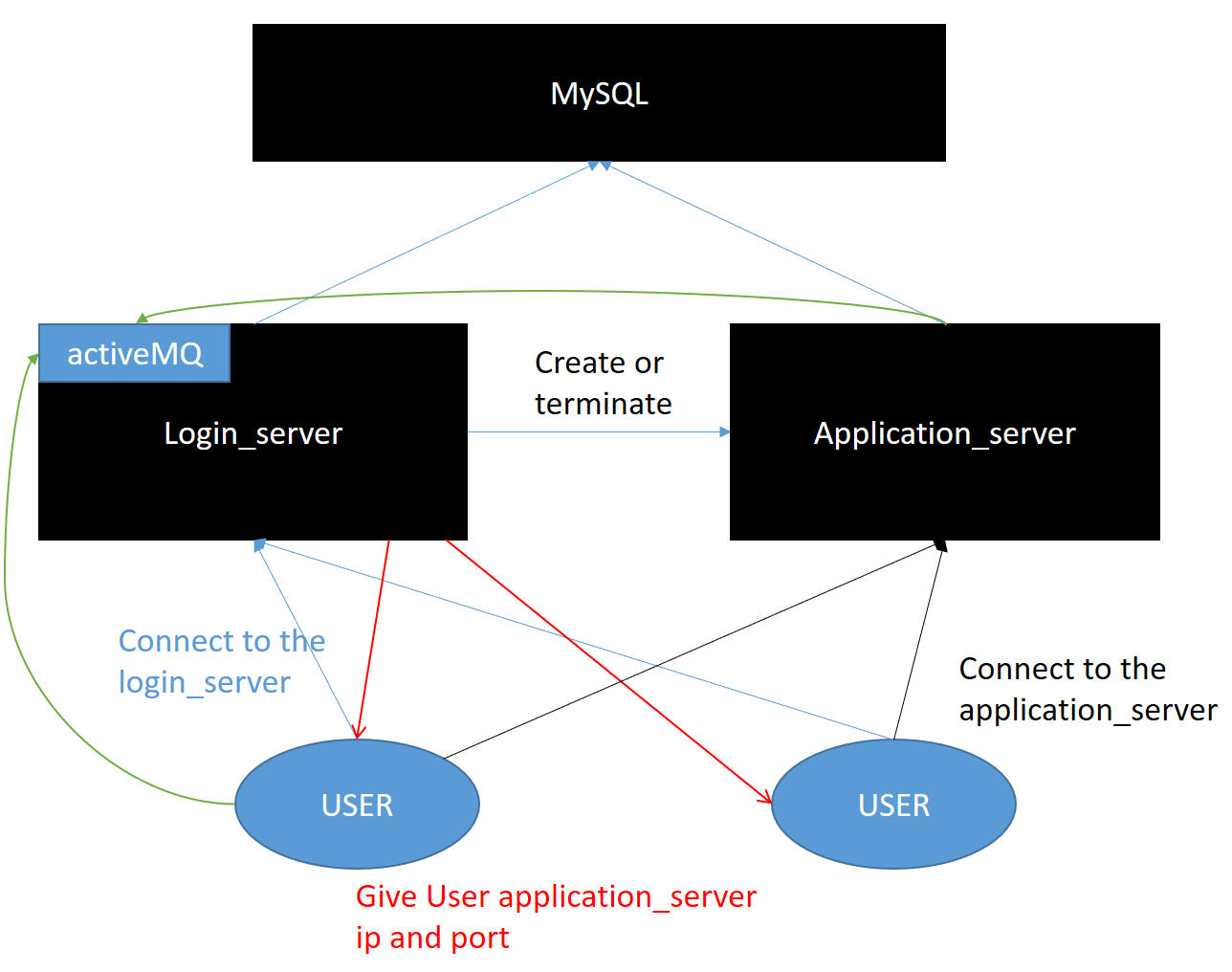


只要在cmd上直接執行client程式就好 後面的arg為login\_server的連線ip和port。



三、程式流程圖

三個黑色的區塊是架在AWS，application\_server是login\_server由boto3控制建立或是中止，login\_server同時也會記錄applicatoin\_server的ip，當user向login\_server要求login時，login\_server同時也會分配一個ip給他，而所有的資料。ActiveMQ則是架在login\_server，供User、application\_server和login\_server使用。



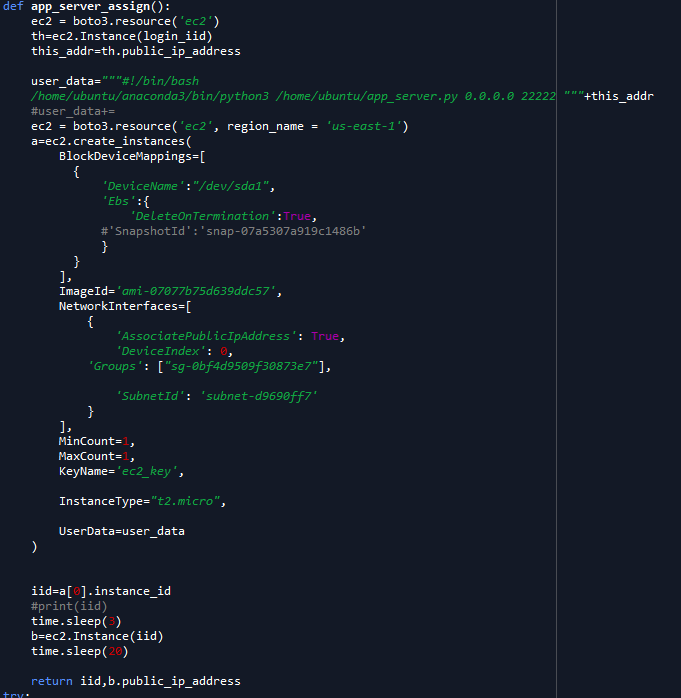
四、程式碼說明

這邊主要說明為了將之前作業放到AWS所增加或改變的code

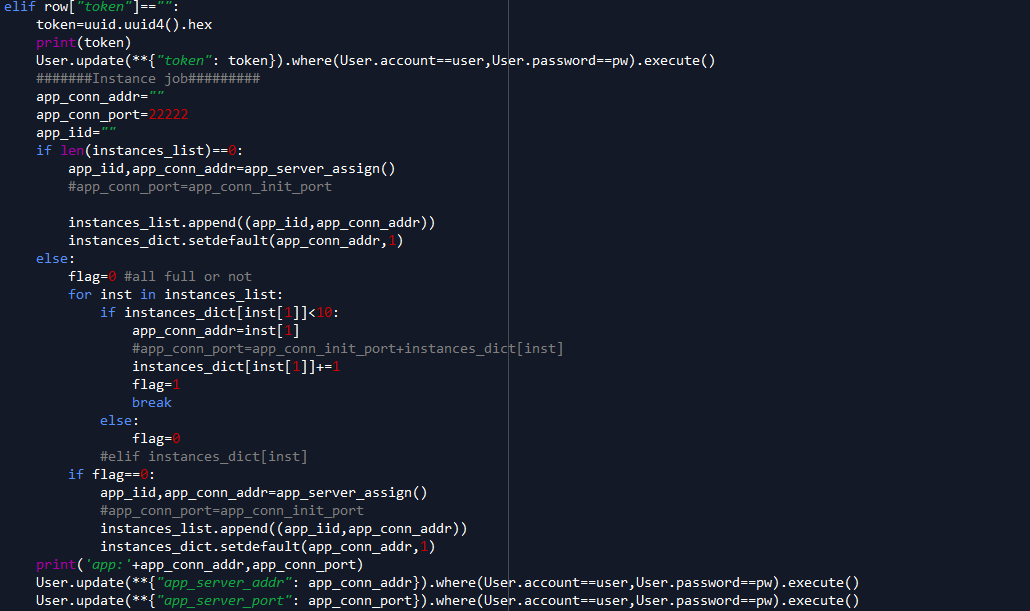
1. Client端在login的時候，會用一個dictionary將user和對應的application\_ip和port存下來，每一個cmd連結到application server或是login server。



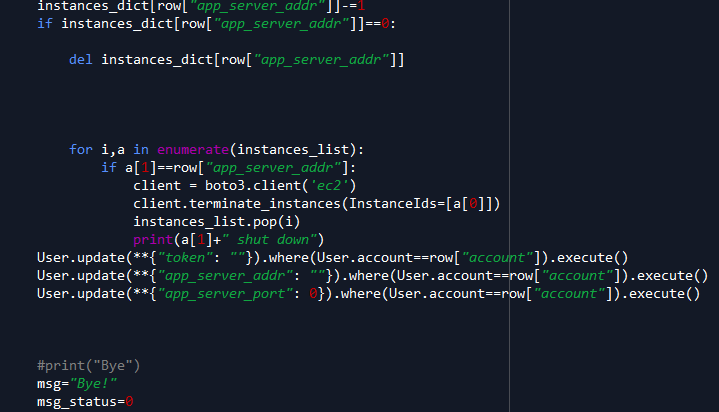
1. login\_server.py 函式app\_server\_assign是當login\_server發現沒有app server存在或是沒空位的時候，就會互叫這個函式，用boto3新建instance當作app server，一樣用user data讓server一開始就run command。



1. 這個部分則是當user login時，login server判斷是否有app server空位，login\_server.py用instances\_list和instances\_dict紀錄曾創建過的app server和其連線數，也會將user和其連線到的ip記錄到database中，以防client中止前沒有將user logout。



而logout的時候也會判斷instance是否已經idle，若idle則直接中止，delete同理。



1. Application\_server的部分則沒太多變動，就只是把先前的server code切到另一個程式去。