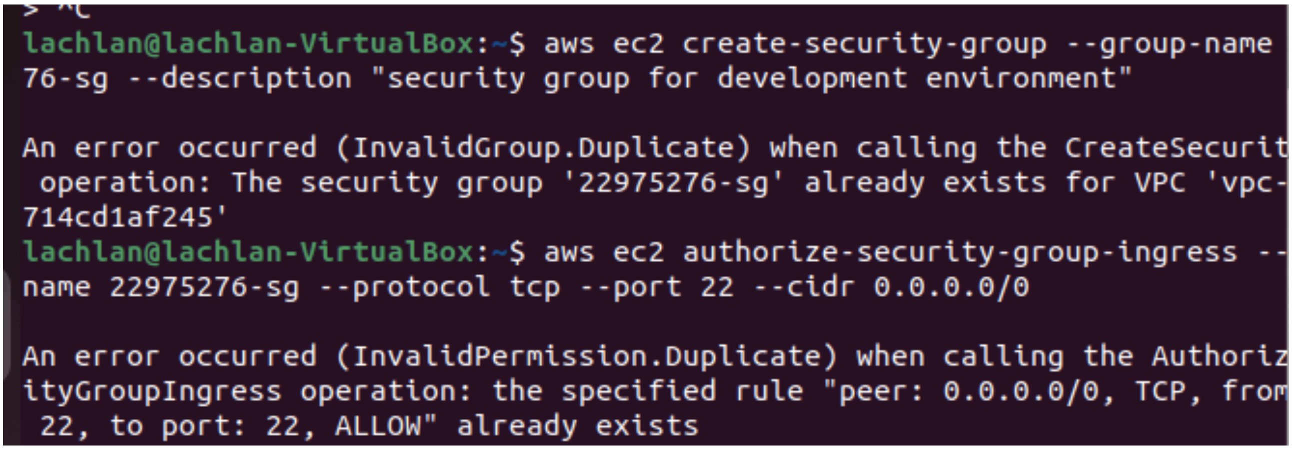
22975276

Part 1

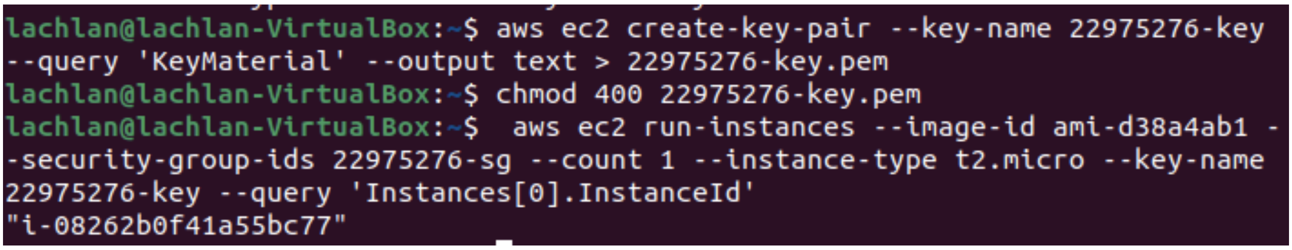
Steps 1 to 2



```
> ^C
lachlan@lachlan-VirtualBox:~$ aws ec2 create-security-group --group-name
76-sg --description "security group for development environment"

An error occurred (InvalidGroup.Duplicate) when calling the CreateSecurit
 operation: The security group '22975276-sg' already exists for VPC 'vpc-
714cd1af245'
lachlan@lachlan-VirtualBox:~$ aws ec2 authorize-security-group-ingress --
name 22975276-sg --protocol tcp --port 22 --cidr 0.0.0.0/0

An error occurred (InvalidPermission.Duplicate) when calling the Authoriz
ityGroupIngress operation: the specified rule "peer: 0.0.0.0/0, TCP, from
 22, to port: 22, ALLOW" already exists
```

There are errors that are returned but only because I had already run the commands. Just illustrating this is the process I did and it is evident that it has been completed by the fact that it's already been done.

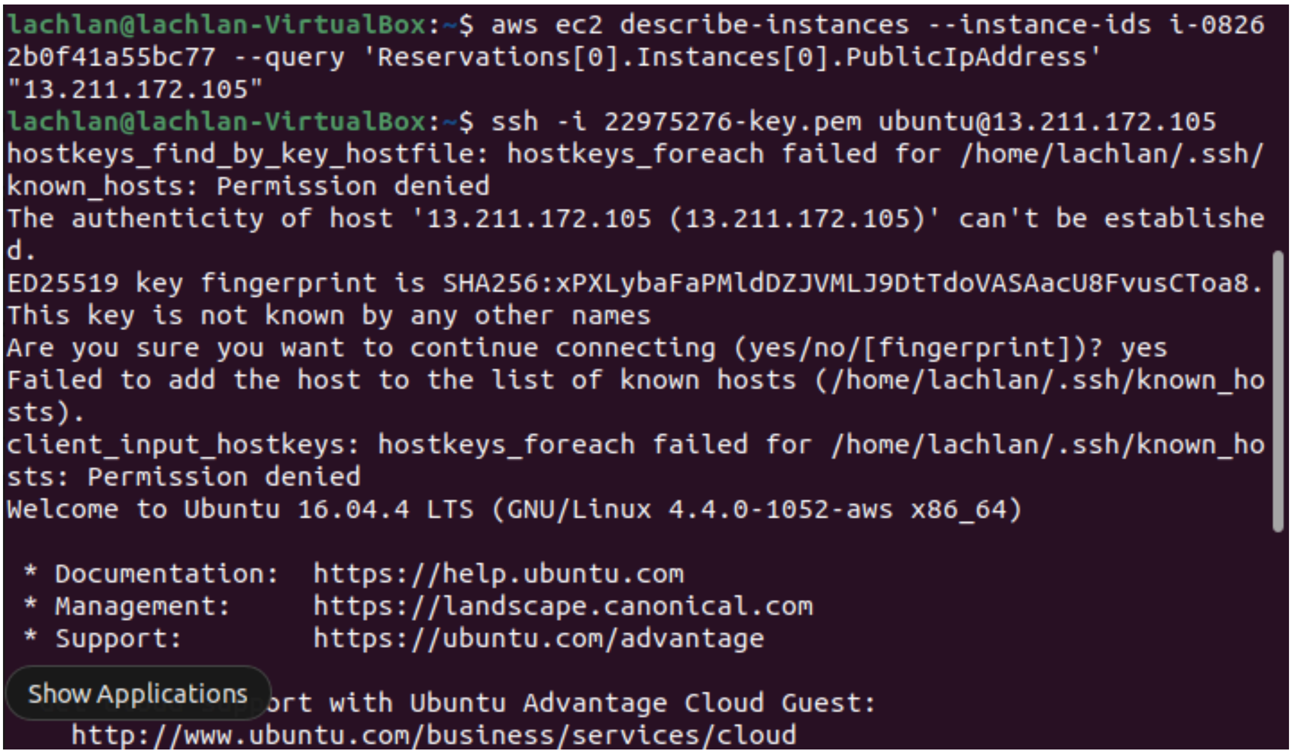
Step 3 and 4 Creating Key Pair and Launching Instance.



```
lachlan@lachlan-VirtualBox:~$ aws ec2 create-key-pair --key-name 22975276-key
--query 'KeyMaterial' --output text > 22975276-key.pem
lachlan@lachlan-VirtualBox:~$ chmod 400 22975276-key.pem
lachlan@lachlan-VirtualBox:~$  aws ec2 run-instances --image-id ami-d38a4ab1 -
-security-group-ids 22975276-sg --count 1 --instance-type t2.micro --key-name
22975276-key --query 'Instances[0].InstanceId'
"i-08262b0f41a55bc77"
```

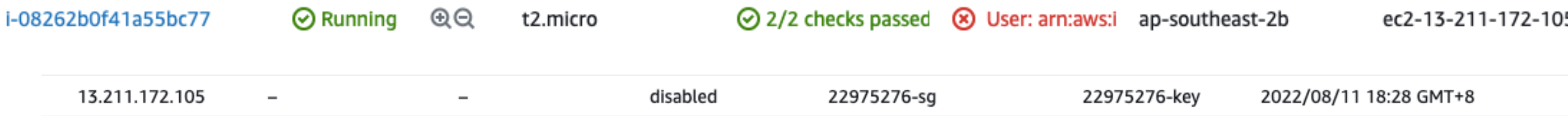
The first command creates the key pair, the second creates the permissions and the third creates the ec2 instance using the key pair and security group.

Step 5,6, and 7 Get the Public IP address, Connect to the Instance, and Look at it in AWS



```
lachlan@lachlan-VirtualBox:~$ aws ec2 describe-instances --instance-ids i-0826
2b0f41a55bc77 --query 'Reservations[0].Instances[0].PublicIpAddress'
"13.211.172.105"
lachlan@lachlan-VirtualBox:~$ ssh -i 22975276-key.pem ubuntu@13.211.172.105
hostkeys_find_by_key_hostfile: hostkeys_foreach failed for /home/lachlan/.ssh/
known_hosts: Permission denied
The authenticity of host '13.211.172.105 (13.211.172.105)' can't be establishe
d.
ED25519 key fingerprint is SHA256:xPXLybaFaPMldDZJVMLJ9DtTdoVASAacU8FvusCToa8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Failed to add the host to the list of known hosts (/home/lachlan/.ssh/known_ho
sts).
client_input_hostkeys: hostkeys_foreach failed for /home/lachlan/.ssh/known_ho
sts: Permission denied
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-1052-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

                  ort with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud
```

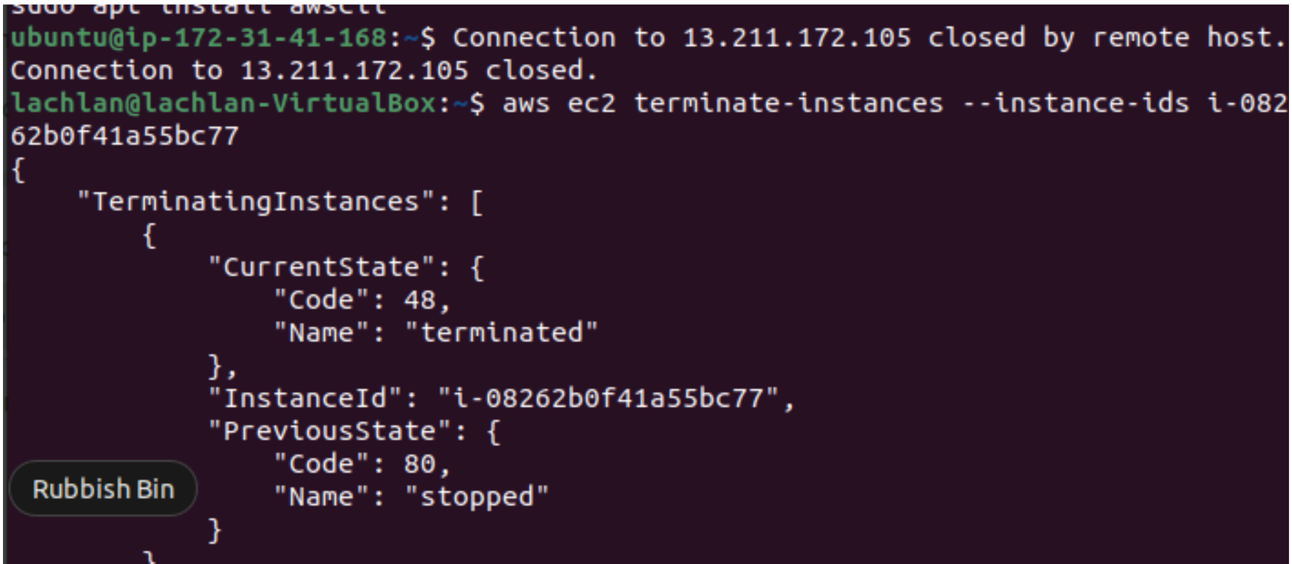
The describe instances command returns the public ip address. 13.211.172.105
The next ssh command connects to the instance using the ip. Unfortunately the screenshot cut off but from here it asked to allow permission which I did and from here I connected to the instance the screenshot below shows it running on the AWS console.



i-08262b0f41a55bc77 ✓ Running t2.micro ✓ 2/2 checks passed ⊗ User: arn:aws:i ap-southeast-2b ec2-13-211-172-105

13.211.172.105 – – disabled 22975276-sg 22975276-key 2022/08/11 18:28 GMT+8

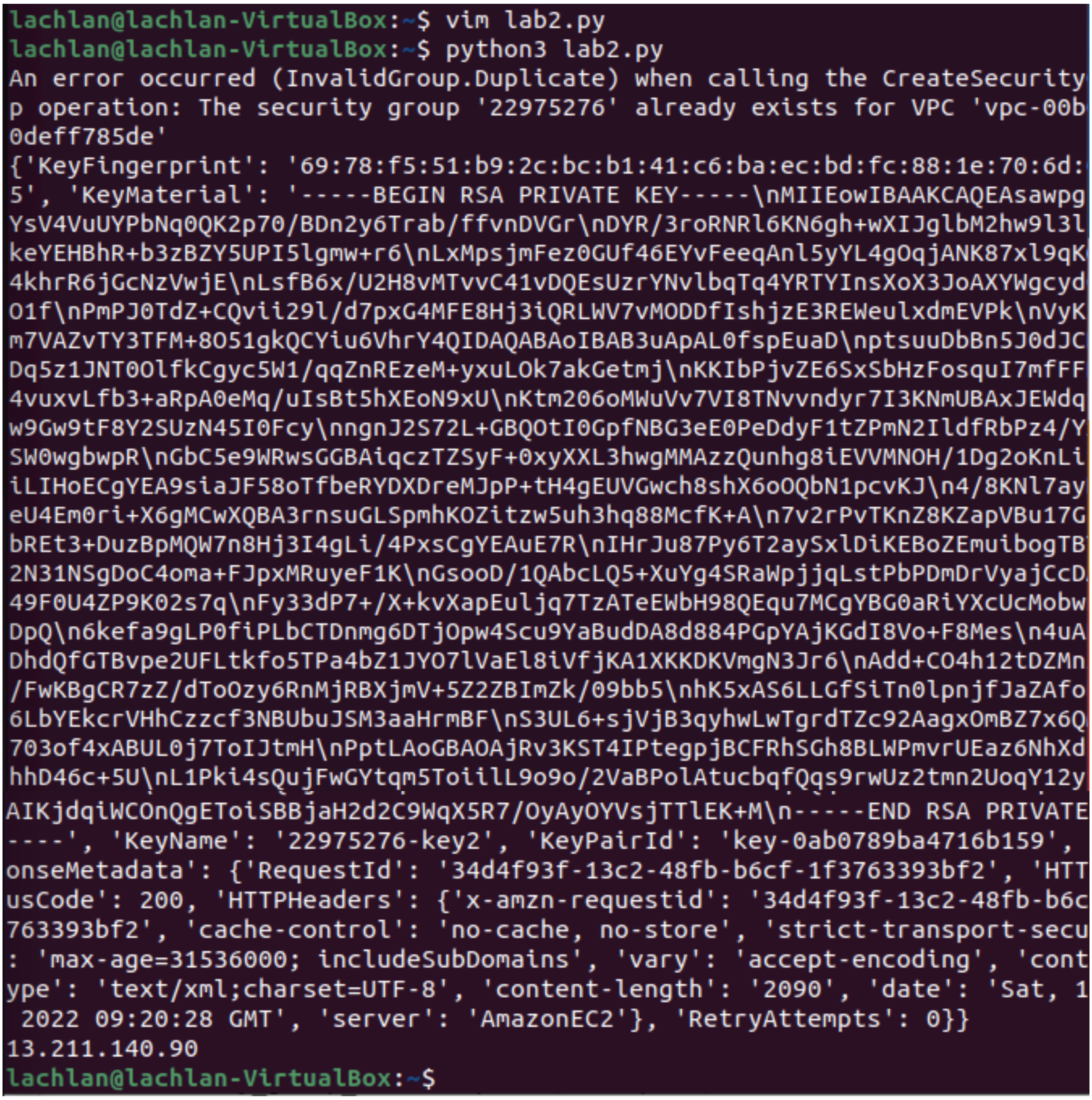
Step 8 Terminate instance



```
sudo apt install awscli
ubuntu@ip-172-31-41-168:~$ Connection to 13.211.172.105 closed by remote host.
Connection to 13.211.172.105 closed.
lachlan@lachlan-VirtualBox:~$ aws ec2 terminate-instances --instance-ids i-082
62b0f41a55bc77
{
    "TerminatingInstances": [
        {
            "CurrentState": {
                "Code": 48,
                "Name": "terminated"
            },
            "InstanceId": "i-08262b0f41a55bc77",
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
```

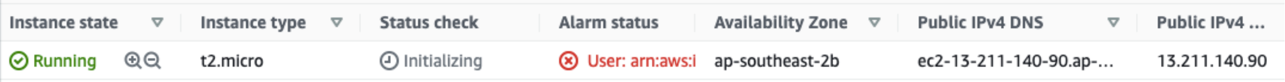
22975276

Part 2 Python Boto3



```
lachlan@lachlan-VirtualBox:~$ vim lab2.py
lachlan@lachlan-VirtualBox:~$ python3 lab2.py
An error occurred (InvalidGroup.Duplicate) when calling the CreateSecurity
p operation: The security group '22975276' already exists for VPC 'vpc-00b
0deff785de'
{'KeyFingerprint': '69:78:f5:51:b9:2c:bc:b1:41:c6:ba:ec:bd:fc:88:1e:70:6d:
5', 'KeyMaterial': '-----BEGIN RSA PRIVATE KEY-----\nMIIEowIBAAKCAQEAsawpg
YsV4VuUYPbNq0QK2p70/BDn2y6Trab/ffvnDVGr\nDYR/3roRNRl6KN6gh+wXIJglbM2hw9l3l
keYEHBhR+b3zBZY5UPI5lgmw+r6\nLxMpsjmFez0GUf46EYvFeeqAnl5yYL4gOqjANK87xl9qK
4khrR6jGcNzVwjE\nLsfB6x/U2H8vMTvvC41vDQEsUzrYNvlbqTq4YRTYInsXoX3JoAXYWgcyd
O1f\nPmPJ0TdZ+CQvii29l/d7pxG4MFE8Hj3iQRLWV7vMODDfIshjzE3REWeulxdmEVPk\nVyK
m7VAZvTY3TFM+8O51gkQCYiu6VhrY4QIDAQABAoIBAB3uApAL0fspEuaD\nptsuuDbBn5J0dJC
Dq5z1JNT0OlfkCgyc5W1/qqZnREzeM+yxuLOk7akGetmj\nKKIbPjvZE6SxSbHzFosquI7mfFF
4vuxvLfb3+aRpA0eMq/uIsBt5hXEoN9xU\nKtm206oMWuVv7VI8TNvvndyr7I3KNmUBAxJEWdq
w9Gw9tF8Y2SUzN45I0Fcy\nngnJ2S72L+GBQOtI0GpfNBG3eE0PeDdyF1tZPmN2IldfRbPz4/Y
SW0wgbwpR\nGbC5e9WRwsGGBAiqczTZSyF+0xyXXL3hwgMMAzzQunhg8iEVVMNOH/1Dg2oKnLi
iLIHoECgYEA9siaJF58oTfbeRYDXDreMJpP+tH4gEUVGwch8shX6oOQbN1pcvKJ\n4/8KNl7ay
eU4Em0ri+X6gMCwXQBA3rnsuGLSpmhKOZitzw5uh3hq88McfK+A\n7v2rPvTKnZ8KZapVBu17G
bREt3+DuzBpMQW7n8Hj3I4gLi/4PxsCgYEAuE7R\nIHrJu87Py6T2aySxlDiKEBoZEmuibogTB
2N31NSgDoC4oma+FJpxMRuyeF1K\nGsooD/1QAbcLQ5+XuYg4SRaWpjjqLstPbPDmDrVyajCcD
49F0U4ZP9K02s7q\nFy33dP7+/X+kvXapEuljq7TzATeEWbH98QEqu7MCgYBG0aRiYXcUcMobw
DpQ\n6kefa9gLP0fiPLbCTDnmg6DTjOpw4Scu9YaBudDA8d884PGpYAjKGdI8Vo+F8Mes\n4uA
DhdQfGTBvpe2UFLtkfo5TPa4bZ1JYO7lVaEl8iVfjKA1XKKDKVmgN3Jr6\nAdd+CO4h12tDZMn
/FwKBgCR7zZ/dToOzy6RnMjRBXjmV+5Z2ZBImZk/09bb5\nhK5xAS6LLGfSiTn0lpnjfJaZAfo
6LbYEkcrVHhCzzcf3NBUbuJSM3aaHrmBF\nS3UL6+sjVjB3qyhwLwTgrdTZc92AagxOmBZ7x6Q
703of4xABUL0j7ToIJtmH\nPptLAoGBAOAjRv3KST4IPtegpjBCFRhSGh8BLWPmvrUEaz6NhXd
hhD46c+5U\nL1Pki4sQujFwGYtqm5ToiilL9o9o/2VaBPolAtucbqfQqs9rwUz2tmn2UoqY12y
AIKjdqiWCOnQgEToiSBBjaH2d2C9WqX5R7/OyAyOYVsjTTlEK+M\n-----END RSA PRIVATE
----', 'KeyName': '22975276-key2', 'KeyPairId': 'key-0ab0789ba4716b159',
onseMetadata': {'RequestId': '34d4f93f-13c2-48fb-b6cf-1f3763393bf2', 'HTT
usCode': 200, 'HTTPHeaders': {'x-amzn-requestid': '34d4f93f-13c2-48fb-b6c
763393bf2', 'cache-control': 'no-cache, no-store', 'strict-transport-secu
: 'max-age=31536000; includeSubDomains', 'vary': 'accept-encoding', 'cont
ype': 'text/xml;charset=UTF-8', 'content-length': '2090', 'date': 'Sat, 1
 2022 09:20:28 GMT', 'server': 'AmazonEC2'}, 'RetryAttempts': 0}}
13.211.140.90
lachlan@lachlan-VirtualBox:~$
```

Since I had previously created the security group in a former attempt it displays an error. But the new key is created and the ip address of the new instance is returned. If running the code with no security group or key saved it will return no already exists errors.



Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
✓ Running	t2.micro	Initializing	☒ User: arn:aws:i	ap-southeast-2b	ec2-13-211-140-90.ap-...	13.211.140.90

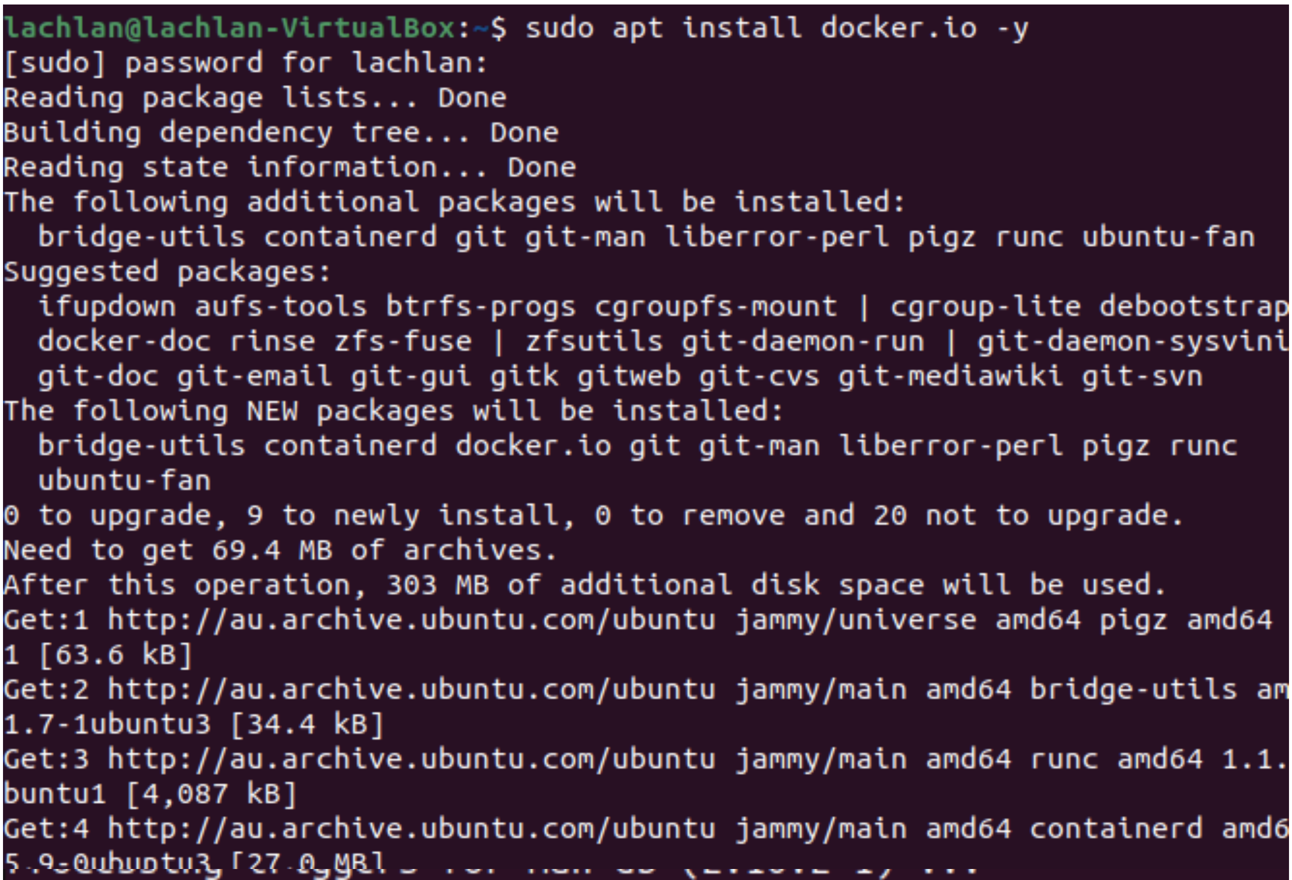
22975276

```
#Python Code
import boto3
ec2 = boto3.resource('ec2')
from botocore.exceptions import ClientError
#CREATE SECURITY GROUP AND AUTHORISE TRAFFIC ON PORT 22 1&2
ec2 = boto3.client('ec2')

response = ec2.describe_vpcs()
vpc_id = response.get('Vpcs', [{}])[0].get('VpcId', '')

try:
    response = ec2.create_security_group(GroupName='22975276',
                                         Description='security group for 
development environment',
                                         VpcId=vpc_id)
    security_group_id = response['GroupId']
#    print('Security Group Created %s in vpc %s.' % (security_group_id, vpc_id))

    data = ec2.authorize_security_group_ingress(
        GroupId=security_group_id,
        IpPermissions=[
            {'IpProtocol': 'tcp',
             'FromPort': 22,
             'ToPort': 22,
             'IpRanges': [{'CidrIp': '0.0.0.0/0'}]},
            {'IpProtocol': 'tcp',
             'FromPort': 22,
             'ToPort': 22,
             'IpRanges': [{'CidrIp': '0.0.0.0/0'}]}
        ])
#    print('Ingress Successfully Set %s' % data)

except ClientError as e:
    print(e)
#CREATE KEY PAIR 3
response = ec2.create_key_pair(KeyName='22975276-key2')
print(response)

#CREATE INSTANCE 4
import boto3
ec2 = boto3.resource('ec2')
instances = ec2.create_instances(
        ImageId="ami-d38a4ab1",
        MinCount=1,
        MaxCount=1,
        InstanceType="t2.micro",
        KeyName="22975276-key2"
    )

#GET IP ADDRESS 5
instances[0].wait_until_running()
instances[0].reload()
ip = instances[0].wait_until_running()
public_ip = instances[0].public_ip_address
print(public_ip)
```

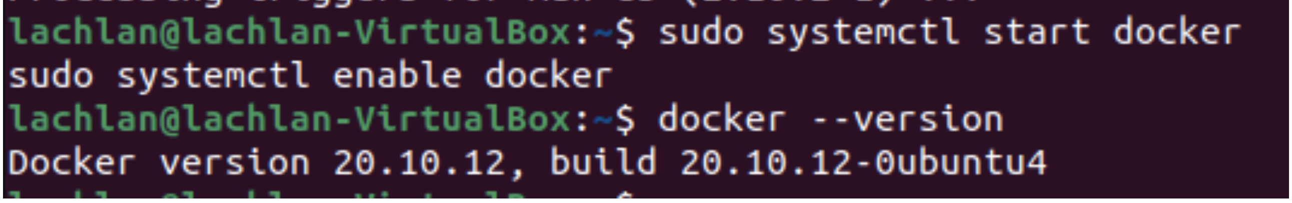
22975276

Part 3 Docker

Step 1 and 2 install docker and check version

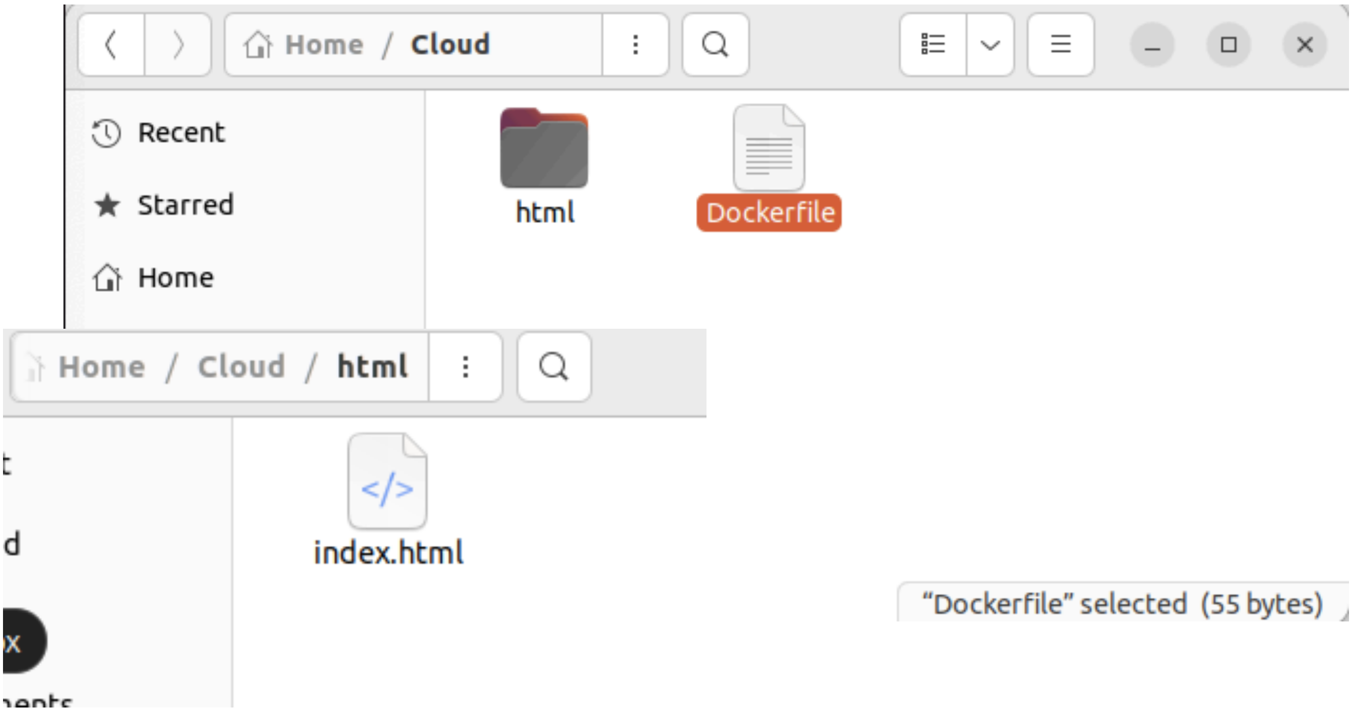


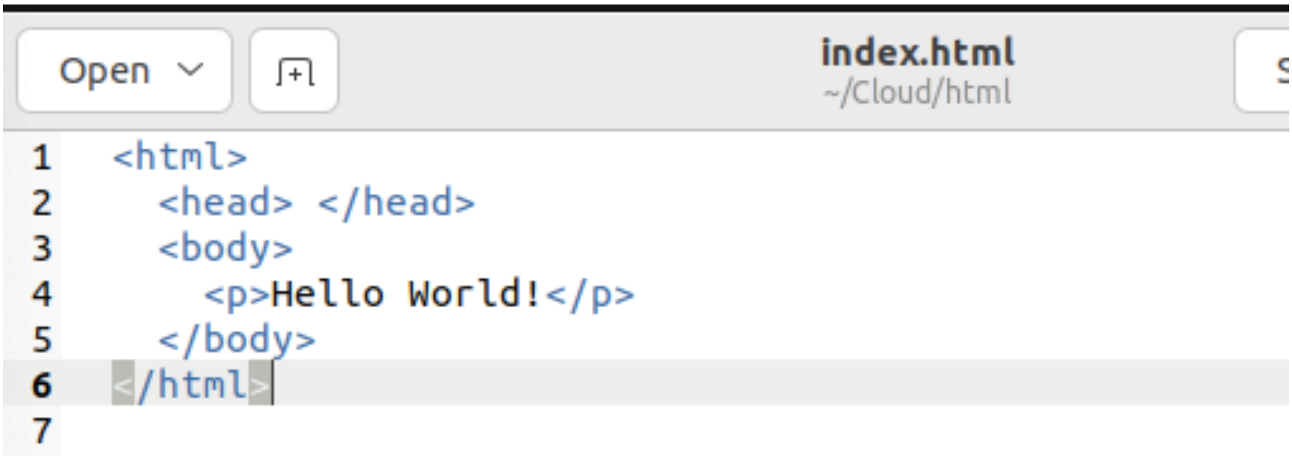
```
lachlan@lachlan-VirtualBox:~$ sudo apt install docker.io -y
[sudo] password for lachlan:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd git git-man liberror-perl pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap
  docker-doc rinse zfs-fuse | zfsutils git-daemon-run | git-daemon-sysvini
  git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  bridge-utils containerd docker.io git git-man liberror-perl pigz runc
  ubuntu-fan
0 to upgrade, 9 to newly install, 0 to remove and 20 not to upgrade.
Need to get 69.4 MB of archives.
After this operation, 303 MB of additional disk space will be used.
Get:1 http://au.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64
1 [63.6 kB]
Get:2 http://au.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils am
1.7-1ubuntu3 [34.4 kB]
Get:3 http://au.archive.ubuntu.com/ubuntu jammy/main amd64 runc amd64 1.1.
buntu1 [4,087 kB]
Get:4 http://au.archive.ubuntu.com/ubuntu jammy/main amd64 containerd amd6
5.9-0ubuntu3 [27.0 MB]
```



```
lachlan@lachlan-VirtualBox:~$ sudo systemctl start docker
sudo systemctl enable docker
lachlan@lachlan-VirtualBox:~$ docker --version
Docker version 20.10.12, build 20.10.12-0ubuntu4
```

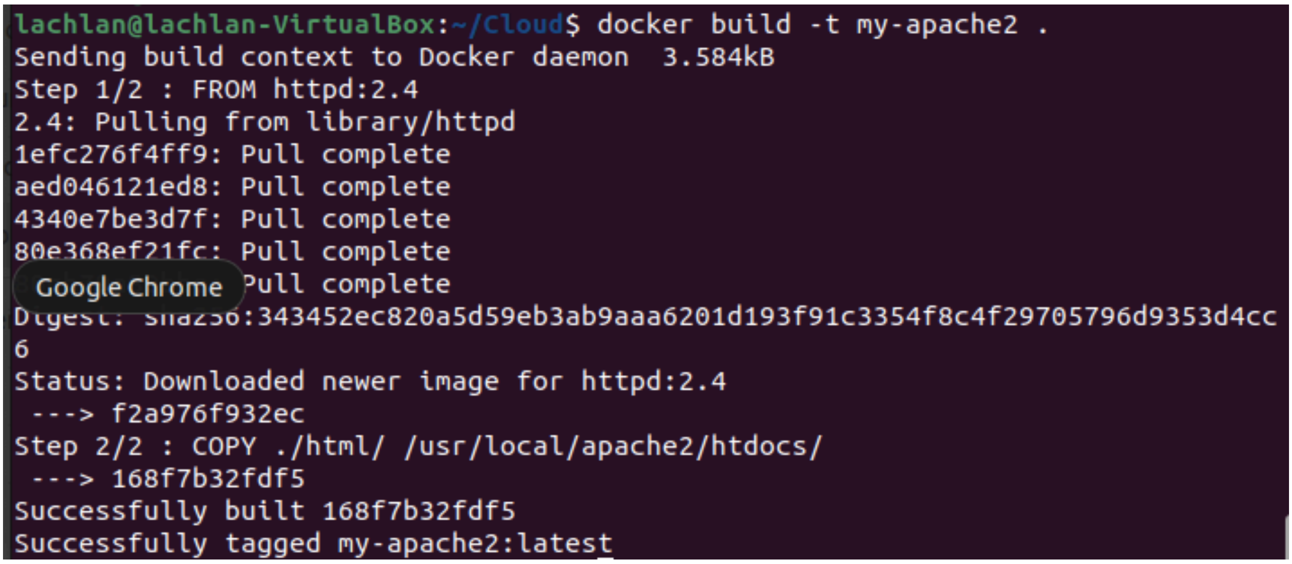
Steps 3 and 4, Build http container and Docker file





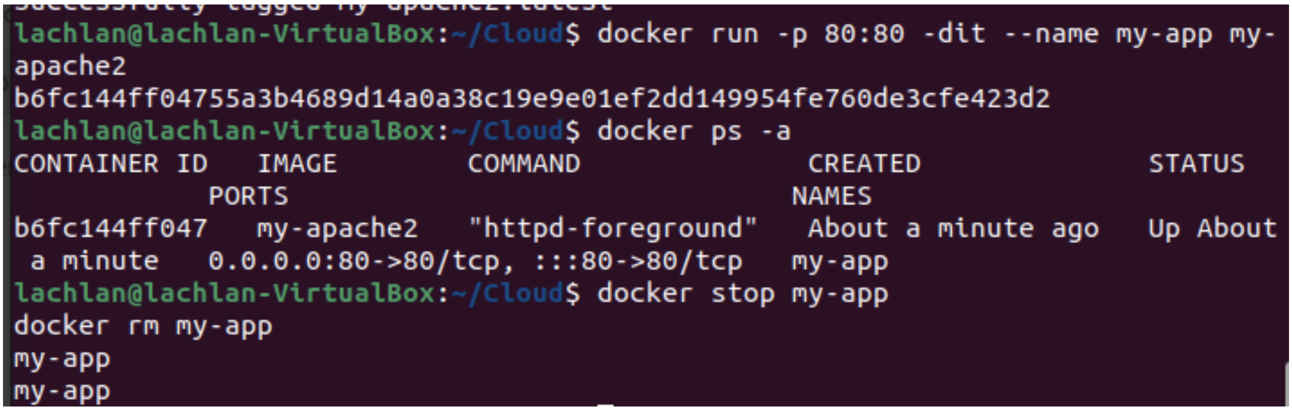
```
<html>
  <head> </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>

```

Step 5 building docker image

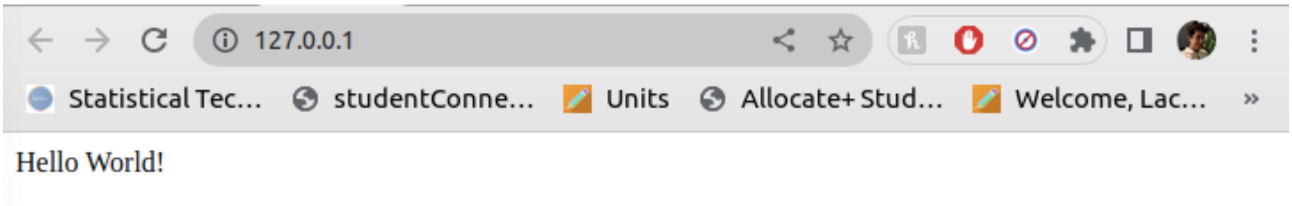


```
lachlan@lachlan-VirtualBox:~/Cloud$ docker build -t my-apache2 .
Sending build context to Docker daemon  3.584kB
Step 1/2 : FROM httpd:2.4
2.4: Pulling from library/httpd
1efc276f4ff9: Pull complete
aed046121ed8: Pull complete
4340e7be3d7f: Pull complete
80e368ef21fc: Pull complete
[illegible]: Pull complete
Digest: sha256:343452ec820a5d59eb3ab9aaa6201d193f91c3354f8c4f29705796d9353d4cc
6
Status: Downloaded newer image for httpd:2.4
 ---> f2a976f932ec
Step 2/2 : COPY ./html/ /usr/local/apache2/htdocs/
 ---> 168f7b32fdf5
Successfully built 168f7b32fdf5
Successfully tagged my-apache2:latest
```

Step 6,7, and 8 run view, and terminate the image



```
Successfully tagged my-apache2:latest
lachlan@lachlan-VirtualBox:~/Cloud$ docker run -p 80:80 -dit --name my-app my-
apache2
b6fc144ff04755a3b4689d14a0a38c19e9e01ef2dd149954fe760de3cfe423d2
lachlan@lachlan-VirtualBox:~/Cloud$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED              STATUS
            PORTS                               NAMES
b6fc144ff047   my-apache2     "httpd-foreground"       About a minute ago   Up About
 a minute   0.0.0.0:80->80/tcp, :::80->80/tcp   my-app
lachlan@lachlan-VirtualBox:~/Cloud$ docker stop my-app
docker rm my-app
my-app
my-app
```



Hello World!