

Practical Worksheet 2

Create an EC2 instance using awscli

Create a security group

```
>_ Labs/lab2 $ aws ec2 create-security-group --group-name 23344153 --description "security group"
{
  "GroupId": "sg-0f2edade19b4a2c3a"
}
```

Security group named **23344153** is created with the description of **security group** and the group ID is returned from the command.

Authorise inbound traffic for ssh

```
>_ Labs/lab2 $ aws ec2 authorize-security-group-ingress --group-name 23344153 --protocol tcp --port 22 --cidr 0.0.0.0/0
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-035221eac70b4b88e",
      "GroupId": "sg-0f2edade19b4a2c3a",
      "GroupOwnerId": "523265914192",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 22,
      "ToPort": 22,
      "CidrIpv4": "0.0.0.0/0"
    }
  ]
}
```

An ingress rule that allows any **IPv4** address to connect under the **TCP** protocol on port **22** which is **SSH** is added to the security group **23344153**.

Create a key pair that will allow you to ssh to the EC2 instance

```
>_ Labs/lab2 $ aws ec2 create-key-pair --key-name 23344153 --query 'KeyMaterial' --output text > ~/.ssh/aws
>_ Labs/lab2 $ ls -l ~/.ssh/aws
-rw-r--r-- 1 kaiqiliang staff 1679 10 Aug 11:47 /Users/kaiqiliang/.ssh/aws
>_ Labs/lab2 $ chmod 400 $ _
>_ Labs/lab2 $ ls -l $ _
-r----- 1 kaiqiliang staff 1679 10 Aug 11:47 /Users/kaiqiliang/.ssh/aws
```

A key pair with the name **23344153** is created and the output is the private key which is redirected to the file **~/.ssh/aws** with its permission changed to readonly to the owner of the file as shown by the **ls -l** command.

Create the instance and note the instance id

```
>_ Labs/lab2 $ aws ec2 run-instances --image-id ami-d38a4ab1 --security-group-ids 23344153 --count 1 --instance-type t2.micro --key-name 23344153 --query 'Instances[0].InstanceId'
"i-066a51f736be366a9"
>_ Labs/lab2 $ aws ec2 create-tags --resources i-066a51f736be366a9 --tags Key=Name,Value=23344153
```

A **t2.micro** type of **ec2** instance of a particular Ubuntu image (specified by the **ami**) is created with the security group and key pair specified to be **23344153** created earlier. A tag of **23344153** is also attached

to the instance so that it can be identified easily.

Get the public IP address

```
>_ Labs/lab2 $ aws ec2 describe-instances --instance-ids i-066a51f736be366a9 --query 'Reservations[0].Instances[0].PublicIpAddress'
"54.253.109.98"
```

Using the instance ID output from the previous step the public IP address of the instance can be queried and outputted to the terminal.

Connect to the instance

```
>_ Labs/lab2 $ ssh -i ~/.ssh/aws ubuntu@54.253.109.98
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

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Aug 10 03:52:49 2022 from 130.95.254.133
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-47-161:~$ exit
logout
Connection to 54.253.109.98 closed.
```

Using the `ssh` command the `ec2` instance can be connected to remotely through its public IP address and authenticated using the private key saved in the `.ssh` directory.

Look at the instance using the AWS console

aws Services Search for services, features, blogs, docs, and more [Option+S] Sydney 23344153@student.uwa.edu.au @ 5232-6591-4192

New EC2 Experience Tell us what you think

EC2 Dashboard
EC2 Global View
Events
Tags
Limits

▼ Instances
Instances **New**
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances **New**
Dedicated Hosts
Capacity Reservations

EC2 > Instances > i-066a51f736be366a9

Instance summary for i-066a51f736be366a9 (23344153) Info

Updated less than a minute ago

Connect Instance state Actions

Instance ID i-066a51f736be366a9 (23344153)	Public IPv4 address 54.253.109.98 open address	Private IPv4 addresses 172.31.47.161
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-54-253-109-98.ap-southeast-2.compute.amazonaws.com open address
Hostname type IP name: ip-172-31-47-161.ap-southeast-2.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-47-161.ap-southeast-2.compute.internal	Elastic IP addresses -
Answer private resource DNS name -	Instance type t2.micro	AWS Compute Optimizer finding ⊗
Auto-assigned IP address 54.253.109.98 [Public IP]	VPC ID vpc-0b754f714cd1af245	

aws Services Search for services, features, blogs, docs, and more [Option+S] Sydney 23344153@student.uwa.edu.au @ 5232-6591-4192

New EC2 Experience Tell us what you think

EC2 Dashboard
EC2 Global View
Events
Tags
Limits

▼ Instances
Instances **New**
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances **New**
Dedicated Hosts
Capacity Reservations

▼ Images
AMIs **New**
AMI Catalog

Details Security Networking Storage Status checks Monitoring Tags

▼ Instance details Info

Platform Ubuntu (Inferred)	AMI ID ami-d38a4ab1	Monitoring disabled
Platform details Linux/UNIX	AMI name ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20180306	Termination protection Disabled
Stop protection Disabled	Launch time Wed Aug 10 2022 11:49:09 GMT+0800 (AWST) (5 minutes)	AMI location 099720109477/ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20180306
Instance auto-recovery Default	Lifecycle normal	Stop-hibernate behavior disabled
AMI Launch index 0	Key pair name 23344153	State transition reason -
Credit specification standard	Kernel ID -	State transition message -
Usage operation RunInstances	RAM disk ID -	Owner 523265914192
ClassicLink -	Enclaves Support -	Boot mode -

The Public IPv4 address, instance ID and security group of the **ec2** instance created through the terminal all match the information found on the aws console.

Terminate the instance

```
>_ Labs/lab2 $ aws ec2 terminate-instances --instance-ids i-066a51f736be366a9
{
  "TerminatingInstances": [
    {
      "CurrentState": {
        "Code": 32,
        "Name": "shutting-down"
      },
      "InstanceId": "i-066a51f736be366a9",
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
  ]
}
```

After terminating the **ec2** instance it outputs the previous state as **running** and the current state as **shutting-down** because it does not terminate immediately, it takes some time to shut down the instance.

Create an EC2 instance with Python Boto script

Repeat the steps above using the equivalent Boto commands in a python script. The script should output the IP address to connect to

```
'''lab2.py'''
import boto3

client = boto3.client('ec2')
ec2 = boto3.resource('ec2')

# create a security group given the group name and description
security_group_id = client.create_security_group(
    GroupName='23344153-sg',
    Description='Security Group'
)['GroupId']

# print out security group ID and Vpc ID
vpc_id = client.describe_vpcs().get('Vpcs', [{}])[0].get('VpcId', '')
print(f'Security Group Created {security_group_id} in vpc {vpc_id}.',
end='\n\n')

# Add inbound rule for the given security group ID
data = client.authorize_security_group_ingress(
    GroupId=security_group_id,
    IpPermissions=[{
        'IpProtocol': 'tcp',
        'FromPort': 22,
        'ToPort': 22,
        'IpRanges': [{'CidrIp': '0.0.0.0/0'}]
    }]
)
print(f'Ingress Successfully Set {data}', end='\n\n')

# create key pair given a key name
keyname = '23344153-key'
print(client.create_key_pair(KeyName=keyname), end='\n\n')

# create an ec2 instance given ami, the number of instances, type, key and security group
instance = ec2.create_instances(
    ImageId='ami-d38a4ab1',
    MaxCount=1,
    MinCount=1,
    InstanceType='t2.micro',
    KeyName=keyname,
    SecurityGroupIds=[security_group_id]
)[0]
print(instance.id, end='\n\n')
```

```
# wait for the instance to launch
instance.wait_until_running()
response = client.describe_instances(InstanceIds=[instance.id])

# once its running print out the public IP address
print(response['Reservations'][0]['Instances'][0]['PublicIpAddress'])
```

Run `lab2.py`.

```
(venv) >_ Labs/lab2 $ python lab2.py
Security Group Created sg-0e0a45c0d732e8675 in vpc vpc-0b754f714cd1af245.

Ingress Successfully Set {'ResponseMetadata': {'RequestId': 'ce1639ed-bc41-45e7-8c76-d50432f0a652',
'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid': 'ce1639ed-bc41-45e7-8c76-d50432f0a652', '
cache-control': 'no-cache, no-store', 'strict-transport-security': 'max-age=31536000; includeSubDoma
ins', 'content-type': 'text/xml; charset=UTF-8', 'content-length': '719', 'date': 'Wed, 10 Aug 2022 0
8:51:45 GMT', 'server': 'AmazonEC2'}, 'RetryAttempts': 0}}

{'KeyFingerprint': '4d:58:90:61:18:c1:aa:cf:21:d0:bf:ed:d1:af:be:fc:55:e8:ba:a6', 'KeyMaterial': '---
---BEGIN RSA PRIVATE KEY-----\nMIIIEPgIBAAKCAQEA0TwdLxLkqKUQ9Mo1dgEippGA2aiyDbPgtrMQwLhuHHcRAfV\ni4C
ea5TqGd151AJ1dJhE+oe4RGulGst01AwPfoX3rTlBklgEtK00hZVdq7xDM5r\nVssDW7++5QmXWKC4sJFnC9en2ApFy8J03mKSq
azjCXbZjUFq802o2IWz2b5Vze0T\nVDrWRIfWqzh0jXQu60ts2YpXmt0vqTgQUSdQylaG/bW2BwQp0EtAy6W7oPqkve4a\nYkbbkL
R/qLHkwyRKpEfKq3/WNdUJ1Z38EjBd4P0sAj/jxp0kvbc42nxKBM7oKMAiC\nT0qLvNBryzw/V8eanjFDXQxICnPIGF6f0euJoQI
DAQABAOIBAQC+82AF2cY4arx4\nynnnTweF/plazTZO8keHInuYw38UA8ViV76nEcJMj13m2MZVPeRb8W2LISUhPmPXK\nn6Fcs1fz
HD2gj/JbtGcrhmG1NZC1iZ6RBN1EbmbpsCqgKmqA20Cb9bRVf52ZUcbuU\n41iJA+jZVOC2aHLOyR70ALHR00QUGESGheKRIRyUE
x3kQRA8N0umEZsbc0u90QbZ\nbRvaTUNcX4h00ZQf2MZrtznX7AuLUyNuEPx7z0U2qOx73w+X4bp7Piv0Qgle0E81\nnhjrM1tDLK
PbsZsnVKVbcytBr1fFpjiZuuax5qN4kBBrrWBczNgj172qEaxuzzvUy\nnFORt3kWBaOGBAPfMSnKILNEEFjSVvW2dAcyEua0eGV
BNq2YuvBohZSiB4w+b158\nneB13PNFkS9bzLWDizybJQx1vfm84+cfXPfVer2T93cnRI3rv0Xc3ZMHfib2bp7fv\nnLC6P4DHo6Nv
9dw9Haw8VeIaE0waFD9a2j1wxGrgixMPdP02iAP9q43ZAoGBANgp\nD44YxCMK6ywmJs0R5d3b7Jw+S8aixTqIC0hMpa7L0VNly
KucHQ+beEURQ18YjmB3\n5SHJVLGwNAEkKD3DAhbky4VYG2v0rSQ8Do6/T7z3N0IgcGSbQ69JDEY9LSiKMtov\nn2jSbGqOz8hDsa
bUr8ebngC7meVBPXTocsxSoDtUJAoGBANWk1SsaLUmohADQmRkQ\nnTclGavVt87+ZEKnpwwMzmrjgWs6Xpf2Y4xbEjzMZfim9UAq
Qgnm2XJ8sWZ4bSX/7\nnG/hJaNSx7F7saeK8kaRvpfdhe6h2Y4w6MawTdoCRhvLk/Zm2wDh+EoQoUKj2YEmG\nn7ONXkuxzzinTMhtk
qAust8pNBAoGBAI4SHrkKfmr5nmz8ZijBNPwsiLXBvLdFigt\nn1kyEo/jI2SHZKhDq82bIpjmu6FKcTye0ZZLZQqd0dxmw7An72
6m0YteXw52TnIc\nnTu0fbIckbNs3eueE7O20M1U07F1Sy61/+gjNUKr7oEPjPm7nHSzIRwWMUXupTfrd\nn1g3jyywRAoGBAPQL+
+tf/edLjg2qN5FdMQSdFNkpV1/XNY2hhOXIGnsVD4ItInDT\nnsHcm00S3aemnmVMoeX2ngpiY0x9iPcOG0GYq210XwxG0w67JMR4k
G3pT0r8/vtLFw\nn7C/oqpQaSk9VcUr9/voF5hYDac5+xWewGuhOP3l3dsHvL/LXSMYYRrGN\nn-----END RSA PRIVATE KEY---
--', 'KeyName': '23344153-key', 'KeyPairId': 'key-03f14dee5d142e7a4', 'ResponseMetadata': {'RequestI
d': 'c096020a-f53e-460e-8ef9-f84de29960f6', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid
': 'c096020a-f53e-460e-8ef9-f84de29960f6', 'cache-control': 'no-cache, no-store', 'strict-transport-
security': 'max-age=31536000; includeSubDomains', 'vary': 'accept-encoding', 'content-type': 'text/x
ml; charset=UTF-8', 'content-length': '2093', 'date': 'Wed, 10 Aug 2022 08:51:45 GMT', 'server': 'Ama
zonEC2'}, 'RetryAttempts': 0}}

i-0f7103e1d7e9af329

3.25.240.167
```

Using Docker

Check the version

```
>_ Labs/lab2 $ docker --version
Docker version 20.10.17, build 100c701
```

Docker version 20.10.17 has been installed on the system and added to the path.

Build and run an httpd container

Create a directory called `html`

```
>_ Labs/lab2 $ mkdir html
>_ Labs/lab2 $ cd html
```

A `html` directory has been created and changed into that directory.

Edit a file index.html inside the html directory

```
>_ lab2/html $ cat index.html
<html>
  <head> </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

An `index.html` file is created with the content of `Hello World!` in a `p` tag.

Create a file called `Dockerfile` outside the html directory

```
>_ Labs/lab2 $ cat Dockerfile
FROM httpd:2.4
COPY ./html/ /usr/local/apache2/htdocs/
```

A `Dockerfile` is created outside the `html` directory with the content of 2 instructions. First to create a `httpd` image of version 2.4. Second to copy the `html` directory to the `usr/local/apache/htdocs` directory in filesystem of the image.

Build the docker image

```
>_ Labs/lab2 $ docker build -t myapache2 .
[+] Building 11.8s (7/7) FINISHED
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 97B                                              0.0s
=> [internal] load .dockerignore                                                0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/httpd:2.4                    4.5s
=> [internal] load build context                                                0.0s
=> => transferring context: 145B                                                0.0s
=> [1/2] FROM docker.io/library/httpd:2.4@sha256:343452ec820a5d59eb3ab9aaa6201d193f91c3354f8 7.0s
=> => resolve docker.io/library/httpd:2.4@sha256:343452ec820a5d59eb3ab9aaa6201d193f91c3354f8 0.0s
=> => sha256:a9fe95647e78b5516c7e2327355b6996e2ea295cd76ae242cbfe87f016b4e 30.05MB / 30.05MB 4.0s
=> => sha256:040f6930c687a462b7e20f12d1fc8aae2134bebb87dbb4f23e3426b510a4e707 146B / 146B 0.4s
=> => sha256:cea7e99357d2a4bc983b615d4202b9fb48f8489e687878c0db26e020918b51e 1.70MB / 1.70MB 1.8s
=> => sha256:343452ec820a5d59eb3ab9aaa6201d193f91c3354f8c4f29705796d9353d4cc 1.86kB / 1.86kB 0.0s
=> => sha256:368f508197c6f79495d2a82d5a2a9db12c15d11c8da6e44cc48fd57c035dd0e 1.37kB / 1.37kB 0.0s
=> => sha256:4ddf3128a12761820ec2b1874bd8db0422a885e6f8d9e0eb6c824267b1fa0c4 9.06kB / 9.06kB 0.0s
=> => sha256:3773f1101b216afb716d84999a1e69189173e4d09bb619515c7cfbbb544c1 23.68MB / 23.68MB 6.2s
=> => sha256:fda5683342e44ff0ed8ef35deedaeae77a6886538cbeb7c52808c60324e64102 297B / 297B 2.8s
=> => extracting sha256:a9fe95647e78b5516c7e2327355b6996e2ea295cd76ae242cbfe87f016b4e760 1.2s
=> => extracting sha256:040f6930c687a462b7e20f12d1fc8aae2134bebb87dbb4f23e3426b510a4e707 0.0s
=> => extracting sha256:cea7e99357d2a4bc983b615d4202b9fb48f8489e687878c0db26e020918b51e7 0.1s
=> => extracting sha256:3773f1101b216afb716d84999a1e69189173e4d09bb619515c7cfbbb544c15ce 0.7s
=> => extracting sha256:fda5683342e44ff0ed8ef35deedaeae77a6886538cbeb7c52808c60324e64102 0.0s
=> [2/2] COPY ./html/ /usr/local/apache2/htdocs/                               0.1s
=> exporting to image                                                            0.0s
=> => exporting layers                                                            0.0s
=> => writing image sha256:1fc943ef591cb2c2ef33af661cf0f071cfa61d938965b48daeed596e90dd4fc2 0.0s
=> => naming to docker.io/library/myapache2                                     0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

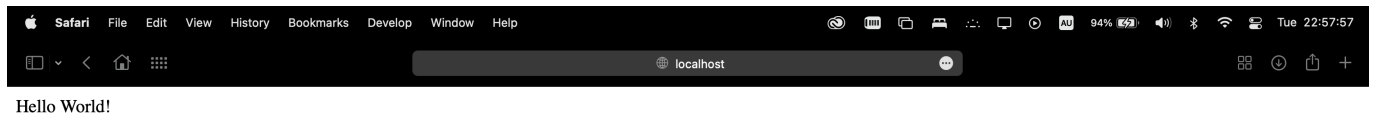
The 2 instructions specified in the `Dockerfile` is executed successfully step by step and a docker image named `myapache2` is created.

Run the image

```
>_ Labs/lab2 $ docker run -p 80:80 -dit --name my-app my-apache2
a7b30b705e8347bec40b3d85bbf89cce096e567b16111c1bf851856ca4b8fb6b
```

A container of the docker image named `my-app` is launched on port 80.

Open a browser and access address <http://localhost> or <http://127.0.0.1> Confirm you get Hello World



The default port `http` uses is 80 so through the URL `localhost` on a browser the docker container can be accessed.

Other commands

```
>_ Labs/lab2 $ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
a7b30b705e83   my-apache2     "httpd-foreground"      2 hours ago   Up 2 hours   0.0.0.0:80->80/tcp      my-app
>_ Labs/lab2 $ docker stop my-app
my-app
>_ Labs/lab2 $ docker rm my-app
my-app
>_ Labs/lab2 $ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
```

After stopping and removing the docker container the name `my-app` is no longer visible.