

Lab5

06 February 2024 18:18

Setting up AWS server

EC2 > Instances > i-0083e04a3a1936f2f

Instance summary for i-0083e04a3a1936f2f (lab5) [Info](#) [Refresh](#) [Connect](#) [Instance state](#) [Actions](#)

Updated less than a minute ago

Instance ID i-0083e04a3a1936f2f (lab5)	Public IPv4 address 13.42.14.20 Open address	Private IPv4 addresses 172.31.34.193
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-13-42-14-20.eu-west-2.compute.amazonaws.com Open address
Hostname type IP name: ip-172-31-34-193.eu-west-2.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-34-193.eu-west-2.compute.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 13.42.14.20 [Public IP]	VPC ID vpc-0029ad6a3ee6759a2	Auto Scaling Group name -
IAM Role DynamoDBAccessEC2	Subnet ID subnet-0a0c1b82088446293	
IMDSv2 Required		

[Details](#) [Status and alarms](#) [Monitoring](#) [Security](#) [Networking](#) [Storage](#) [Tags](#)

Instance details [Info](#)

Platform Ubuntu (Inferred)	AMI ID ami-0ff1c68c6e837b183	Monitoring disabled
Platform details Linux/UNIX	AMI name ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-	Termination protection Disabled

Connecting via putty and installing python

```
ubuntu@ip-172-31-34-193: ~  
  
System information as of Tue Feb 6 18:21:20 UTC 2024  
  
System load: 0.0          Processes: 99  
Usage of /: 23.9% of 7.57GB Users logged in: 0  
Memory usage: 18%        IPv4 address for eth0: 172.31.34.193  
Swap usage: 0%  
  
Expanded Security Maintenance for Applications is not enabled.  
  
84 updates can be applied immediately.  
58 of these updates are standard security updates.  
To see these additional updates run: apt list --upgradable  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
New release '22.04.3 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Tue Feb 6 17:26:32 2024 from 146.169.187.24  
ubuntu@ip-172-31-34-193:~$
```

Testing python program

```
ubuntu@ip-172-31-34-193: ~  
System load:  0.0          Processes:    99  
Usage of /:   23.9% of 7.57GB  Users logged in: 0  
Memory usage: 18%          IPv4 address for eth0: 172.31.34.193  
Swap usage:  0%  
  
Expanded Security Maintenance for Applications is not enabled.  
  
84 updates can be applied immediately.  
58 of these updates are standard security updates.  
To see these additional updates run: apt list --upgradable  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
New release '22.04.3 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Tue Feb  6 17:26:32 2024 from 146.169.187.24  
ubuntu@ip-172-31-34-193:~$ python3 simpleHello.py  
test test 123  
ubuntu@ip-172-31-34-193:~$
```

Using TCP:

```
ubuntu@ip-172-31-34-193: ~  
Expanded Security Maintenance for Applications is not enabled.  
  
84 updates can be applied immediately.  
58 of these updates are standard security updates.  
To see these additional updates run: apt list --upgradable  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
New release '22.04.3 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Tue Feb  6 17:26:32 2024 from 146.169.187.24  
ubuntu@ip-172-31-34-193:~$ python3 simpleHello.py  
test test 123  
ubuntu@ip-172-31-34-193:~$ python3 tcpserver2.py  
File "tcpserver2.py", line 19  
    cmsg = "Not alphanumeric";  
    ^  
IndentationError: expected an indented block  
ubuntu@ip-172-31-34-193:~$ python3 tcpserver.py  
Server running on port 12000
```

Didn't work initially

```
IDLE Shell 3.12.1  
File Edit Shell Debug Options Window Help  
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: C:\Users\sbarb\Downloads\tcpclient(1).py =====  
Traceback (most recent call last):  
  File "C:\Users\sbarb\Downloads\tcpclient(1).py", line 15, in <module>  
    client_socket.connect((server_name, server_port))  
TimeoutError: [WinError 10060] A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond  
pen  
>>>  
===== RESTART: C:\Users\sbarb\Downloads\tcpclient(1).py =====  
Traceback (most recent call last):  
  File "C:\Users\sbarb\Downloads\tcpclient(1).py", line 15, in <module>  
    client_socket.connect((server_name, server_port))  
TimeoutError: [WinError 10060] A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond  
pen  
>>>  
===== RESTART: C:\Users\sbarb\Downloads\tcpclient(1).py =====  
Traceback (most recent call last):  
  File "C:\Users\sbarb\Downloads\tcpclient(1).py", line 15, in <module>  
    client_socket.connect((server_name, server_port))  
TimeoutError: [WinError 10060] A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond  
hh  
>>>
```

Had to reconfigure outbound TCP security settings in AWS. Somehow reverted to old settings?

```

EC2 > Security Groups > sg-0da880e9141b6a93d - launch-wizard-1
sg-0da880e9141b6a93d - launch-wizard-1

ubuntu@ip-172-31-34-193:~$
System load:  0.17      Processes:    105
Usage of /:   30.4% of 7.57GB
Memory usage: 25%      Users logged in: 0
Swap usage:   0%        IPv4 address for eth0: 172.31.34.193

Expanded Security Maintenance for Applications is not enabled.
26 updates can be applied immediately.
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo psb status
New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Thu Feb  8 10:44:13 2024 from 3.8.37.27
ubuntu@ip-172-31-34-193:~$ python3 tcpserver.py
Server running on port 12000

[Send me Message]
client_socket.send(msg)

#Return values from t
msg = client_socket.r
print(msg.decode())
client_socket.close()

===== RESTART: C:\Users\sbarb\Downloads\tcpclient(1).py =====
Traceback (most recent call last):
  File "C:\Users\sbarb\Downloads\tcpclient(1).py", line 15, in <modu
    client_socket.connect((server_name, server_port))
TimeoutError: [WinError 10060] A connection attempt failed because t
party did not properly respond after a period of time, or establish
n failed because connected host has failed to respond
pen

===== RESTART: C:\Users\sbarb\Downloads\tcpclient(1).py =====
Traceback (most recent call last):
  File "C:\Users\sbarb\Downloads\tcpclient(1).py", line 15, in <modu
    client_socket.connect((server_name, server_port))
TimeoutError: [WinError 10060] A connection attempt failed because t
party did not properly respond after a period of time, or establish
n failed because connected host has failed to respond
pen

===== RESTART: C:\Users\sbarb\Downloads\tcpclient(1).py =====
Traceback (most recent call last):
  File "C:\Users\sbarb\Downloads\tcpclient(1).py", line 15, in <modu
    client_socket.connect((server_name, server_port))
TimeoutError: [WinError 10060] A connection attempt failed because t
party did not properly respond after a period of time, or establish
n failed because connected host has failed to respond
pen

===== RESTART: C:\Users\sbarb\Downloads\tcpclient(1).py =====
Traceback (most recent call last):
  File "C:\Users\sbarb\Downloads\tcpclient(1).py", line 15, in <modu
    client_socket.connect((server_name, server_port))
TimeoutError: [WinError 10060] A connection attempt failed because t
party did not properly respond after a period of time, or establish
n failed because connected host has failed to respond
pen

===== RESTART: C:\Users\sbarb\Downloads\tcpclient(1).py =====
TCP client running...
Connecting to server at IP: 18.130.170.115 PORT: 12000
Enter a string to test if it is alphanumeric: hh
Alphanumeric

```

Now it works

Now we want to make the service run in the background so we don't have to keep the ssh connection live for the program to stay active

[Unit]

Description=TCP server service

After=multi-user.target

[Service]

Type=simple ExecStart=/usr/bin/python3 /home/ubuntu/tcpserver.py

[Install]

WantedBy=multi-user.target

This code sets the program to start on boot

```

root    761      1  0 12:17 tty1      00:00:00 /sbin/agetty -L -p -- \u --noclear tty1 1
root    773     604  0 12:17 ?                00:00:00 sshd: ubuntu [priv]
ubuntu  776      1  0 12:17 ?                00:00:00 /lib/systemd/systemd --user
ubuntu  777     776  0 12:17 ?                00:00:00 (sd-pam)
ubuntu  873     773  0 12:17 ?                00:00:00 sshd: ubuntu@pts/0
ubuntu  874     873  0 12:17 pts/0            00:00:00 -bash
root    1046     1  0 12:22 ?                00:00:00 /usr/bin/python3 /home/ubuntu/tcpserver.p
ubuntu  1047     874  0 12:22 pts/0            00:00:00 ps -ef
ubuntu@ip-172-31-34-193:~$

```

Still works

```

===== RESTART: C:\Users\sbarb\Downloads\tcpclient(1).py =====
TCP client running...
Connecting to server at IP: 13.42.50.83 PORT: 12000
Enter a string to test if it is alphanumeric: dfs
Alphanumeric

```

Ending process

```

root    1046     1  0 12:22 ?                00:00:00 /usr/bin/python3 /home/ubuntu/tcpserver.p
ubuntu  1047     874  0 12:22 pts/0            00:00:00 ps -ef
ubuntu@ip-172-31-34-193:~$ sudo kill -9 1046
ubuntu@ip-172-31-34-193:~$ ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD

```

Challenge: Computing RTT (Round Trip time) for 500 communications

The image shows two side-by-side Python IDE windows. The left window, titled 'RTTClient.py - C:/Users/sbarb/Downloads/RTTClient.py (3.12.1)', contains the following code:

```

import socket
import time

SERVER_HOST = '13.42.50.83'
SERVER_PORT = 12000
NUM_COMMUNICATIONS = 500

def tcp_client(data):
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as sock:
        sock.connect((SERVER_HOST, SERVER_PORT))
        start_time = time.time()
        sock.sendall(data.to_bytes(4, byteorder='big'))
        response = sock.recv(1024)
        end_time = time.time()
        rtt = end_time - start_time
        return rtt

if __name__ == "__main__":
    running_average = 0
    for i in range(NUM_COMMUNICATIONS):
        rtt = tcp_client(69)
        running_average = (running_average * i + rtt) / (i + 1)
        print(f"Communication {i+1}/{NUM_COMMUNICATIONS}: RTT = {rtt:.6f}s, Running Average = {running_average:.6f}s")

```

The right window, titled 'RTTServer.py - C:/Users/sbarb/Downloads/RTTServer.py (3.12.1)', contains the following code:

```

import socket

SERVER_HOST = '0.0.0.0'
SERVER_PORT = 12000

def tcp_server():
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as sock:
        sock.bind((SERVER_HOST, SERVER_PORT))
        sock.listen(1)
        while True:
            conn, addr = sock.accept()
            with conn:
                data = conn.recv(1024)
                if not data:
                    break
                conn.sendall(data)

if __name__ == "__main__":
    tcp_server()

```

Client sends an integer and the server sends back whatever integer it received. Time between it being sent and received and received again is measured and then average RTT is computed by the client.

Basic python code to send communications and compute RTT

The image shows a terminal window with the following output:

```

Communication 496/500: RTT = 0.010723s, Running Average = 0.011416s
Communication 497/500: RTT = 0.008909s, Running Average = 0.011411s
Communication 498/500: RTT = 0.007024s, Running Average = 0.011402s
Communication 499/500: RTT = 0.013177s, Running Average = 0.011405s
Communication 500/500: RTT = 0.009109s, Running Average = 0.011401s

```

Average RTT was 0.011401 over 500 communications

Sending accelerometer data to server via TCP

The image shows a Python IDE window with the following code:

```

import subprocess
import socket

SERVER_HOST = '13.42.50.83'
SERVER_PORT = 12000
NUM_COMMUNICATIONS = 500
NIOS_CMD_SHELL_BAT = "C:/intelFPGA_lite/18.1/nios2eds/Nios II Command Shell.bat"

# create a TCP client socket
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

def send_on_jtag(cmd):
    # check if atleast one character is being sent down
    assert (len(cmd) >= 1), "Please make the cmd a single character"

    # create a subprocess which will run the nios2-terminal
    process = subprocess.Popen(
        NIOS_CMD_SHELL_BAT,
        bufsize=0,
        stdin=subprocess.PIPE,
        stdout=subprocess.PIPE,
        stderr=subprocess.PIPE,
    )

    # send the cmd string to the nios2-terminal, read the output and terminate the process
    try:
        vals, err = process.communicate(
            bytes("nios2-terminal <<< {} ".format(cmd), "utf-8")
        )
        process.terminate()
    except subprocess.TimeoutExpired:
        vals = "Failed"
        process.terminate()
    return vals

def perform_computation():
    res = send_on_jtag("1") # 1 get accelerometer value for after filter 1
    # you can process the output here
    print(res)
    return res

if __name__ == "__main__":
    value = perform_computation()
    # create a TCP client socket
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    # Set up a TCP connection with the server
    # a connection socket will be assigned to this client on the server side
    client_socket.connect((server_name, server_port))

    print("TCP client running...")
    print("Connecting to server at IP: ", server_name, " PORT: ", server_port)

    # send the message to the udp server
    client_socket.send(value.encode())

    # return values from the server
    msg = client_socket.recv(1024)
    print(msg.decode())
    client_socket.close()

```

```

RRTServer.py - C:\Users\lsarb\Downloads\RRTServer.py (3.12.2)
File Edit Format Run Options Window Help
import socket

# Server configuration
SERVER_HOST = '0.0.0.0'
SERVER_PORT = 12000

def tcp_server():
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as sock:
        sock.bind((SERVER_HOST, SERVER_PORT))
        sock.listen(1)
        print(f"Server listening on {SERVER_HOST}:{SERVER_PORT}...")

        while True:
            conn, addr = sock.accept()
            with conn:
                print(f"Connected to {addr}")
                data = conn.recv(1024)
                if not data:
                    break
                msg = "Received: " + str(data)
                conn.sendall(msg.encode())

if __name__ == "__main__":
    tcp_server()

```

Sends value 500 times to the server. Server responds with 'Received:(Value received)'