

# CS451 Homework 1

Name: **Batkhisig Dulamsurankhor**

CWID: A20543498

Collaborator: None

1. Sign up for an AWS Account with your IIT email

<https://www.awseducate.com/registration/s/registration-detail?>

The screenshot shows the 'Create your account' section of the AWS Educate sign-up page. It includes fields for First name, Middle name (optional), Last name, Country, State or province, City, and Birth month. There is also a note about filling out fields to create an account and a link to sign up as a recruiter.

2. Login into AWS Educate <https://www.awseducate.com/>

The screenshot shows the AWS Educate homepage with a 'Your cloud journey starts here' banner. It features a 'Getting Started' section with three courses: 'Introduction to the AWS Management Console', 'Introduction to Cloud 101', and 'Getting Started with Storage'. On the left, there are filters for Course Features, Skills, Level, and Duration. On the right, there are sections for 'Explore' (with links to learn more from AWS and share feedback) and 'Fast-track your future' (with links to AWS Cloud Institute and badge-eligible courses).

- a. Take Introduction to AWS Management if you have never used AWS
- b. Also take Cloud 101 if you have never used AWS

3. Let's deploy your 1 st vm , login in to <http://aws.amazon.com>
- a. It should look like this:

The screenshot shows the AWS Console Home page. In the top left, under 'Recently visited', 'EC2' is highlighted with an orange icon. To the right, there's a section titled 'Applications (0)' with a 'Create application' button. Below it, a message says 'No applications' and 'Get started by creating an application.' At the bottom of this section is another 'Create application' button. On the far left, there's a 'Welcome to AWS' sidebar with 'Getting started with' and 'Open issues' links. On the right, there's a 'Cost and usage' section. The bottom navigation bar includes links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

- b. Search for EC2 by clicking view all services

The screenshot shows the 'All services' page. The top navigation bar has 'Console Home > All services'. The main area is titled 'All services' and contains a 'Services by category' grid. The 'Compute' section is expanded, listing services like EC2, Lightsail, Lambda, Batch, Elastic Beanstalk, Serverless Application Repository, AWS Outposts, EC2 Image Builder, AWS App Runner, AWS SimSpace Weaver, and Parallel Computing Service. Other sections shown include 'Quantum Technologies', 'Management & Governance', and 'Security, Identity, & Compliance'. The bottom navigation bar includes links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

### c. Click EC2

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links for EC2 Global View, Instances, Images, and Elastic Block Store. The main area displays 'Resources' with a summary of current usage across various categories like Instances (running), Auto Scaling Groups, Capacity Reservations, Dedicated Hosts, etc. To the right, there's a section for 'EC2 Free Tier' which indicates 0 EC2 free tier offers in use. It also shows end-of-month forecasts and exceedance alerts. Below this is a 'Service health' section and an 'Account attributes' section.

### d. Click Launch Instance and you should see

This screenshot shows the 'Launch an instance' wizard. Step 1: Set instance details. It asks for the number of instances (set to 1). It also lists the selected software image (Amazon Linux 2023 AMI 2023.5.2...), virtual server type (t2.micro), and firewall (New security group). A note about the free tier is displayed. At the bottom are 'Cancel', 'Launch instance', and 'Review commands' buttons.

e. It should look like this:

The screenshot shows the AWS Launch Wizard interface. At the top, it displays the instance type as t2.micro, which is free tier eligible. Below this, there's a section for a key pair (login) with a dropdown menu showing 'aws-auth'. Under network settings, it shows a VPC (vpc-0150d260377addf21) and a subnet. A note about additional charges for auto-assigning public IP is present. On the right side, the summary panel shows 1 instance, the software image as Amazon Linux 2023 AMI 2023.5.2, and the virtual server type as t2.micro. A callout box highlights the 'Free tier' information. At the bottom right, there are 'Launch instance' and 'Review commands' buttons.

And you will likely need to generate a new RSA key pair so you can login. It will download a PEM file. Make sure you allow 0.0.0.0/0 traUic or your IP address so you can login. Also for the storage, take the default 8gb . After you did all this, click Launch

Instance at the bottom Right.

This screenshot shows the continuation of the AWS Launch Wizard configuration. It includes sections for 'Key pair (login)' (aws-auth), 'Network settings' (VPC: vpc-0150d260377addf21, Subnet: No preference), and 'Additional charges apply' (Auto-assign public IP: Enable). On the right, the summary panel shows 1 instance, the software image as Amazon Linux 2023 AMI 2023.5.2, and the virtual server type as t2.micro. A callout box highlights the 'Free tier' information. At the bottom right, there are 'Launch instance' and 'Review commands' buttons.

4. Login.

a. Copy the IPv4 address and open the shell or Putty

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with links like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots, Lifecycle Manager). The main area is titled 'Instances (1) Info' and shows one instance: 'check' (status: initializing), 'Alarm status' (View alarms +), 'Availability Zone': us-east-2c, 'Public IPv4 DNS': ec2-3-22-98-44.us-east..., 'Public IPv4 IP': 3.22.98.44, 'Elastic IP': -, 'IPv6 IPs': -, 'Monitoring': disabled. Below this is a 'Select an instance' dropdown.

b. Type or enter – note your PEM file might be different:

- ssh -i aws-auth.pem ec2-user@3.22.98.44
- enter 'sudo -i' as seen below
- You'll see this

The terminal window title is '2024 Fall – root@ip-172-31-43-114:~ – ssh -i aws-auth.pem ec2-user@3.22.98.44 – 101x24'. The terminal output shows:

```
WARNING: UNPROTECTED PRIVATE KEY FILE!
Permissions 0644 for 'aws-auth.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "aws-auth.pem": bad permissions
ec2-user@3.22.98.44: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[batkhishig@dhcp20 2024 Fall % chmod 400 aws-auth.pem
[batkhishig@dhcp20 2024 Fall % ssh -i aws-auth.pem ec2-user@3.22.98.44
#_#
\_###_ Amazon Linux 2023
\_###\_
\###|
\#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
V~' '-->
~~_/
~~-. _/
/_m/'
```

[[ec2-user@ip-172-31-43-114 ~]\$  
[[ec2-user@ip-172-31-43-114 ~]\$  
[[ec2-user@ip-172-31-43-114 ~]\$ sudo -i  
[root@ip-172-31-43-114 ~]# ]]

c. Install Java by entering ‘yum -y install java g++ gcc php-cli php-xml php-json’

```
2024 Fall — root@ip-172-31-43-114:~ — ssh -i aws-auth.pem ec2-user@3.22.98.44 — 101x24

libXi-1.7.10-6.amzn2023.0.2.x86_64
libXinerama-1.1.4-8.amzn2023.0.2.x86_64
libXrandr-1.5.2-6.amzn2023.0.2.x86_64
libXrender-0.9.10-14.amzn2023.0.2.x86_64
libXt-1.2.0-4.amzn2023.0.2.x86_64
libXtst-1.2.3-14.amzn2023.0.2.x86_64
libbrotli-1.0.9-4.amzn2023.0.2.x86_64
libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64
libmpc-1.2.1-2.amzn2023.0.2.x86_64
libpng-2:1.6.37-10.amzn2023.0.6.x86_64
libstdc++-devel-11.4.1-2.amzn2023.0.2.x86_64
libtool-ltdl-2.4.7-1.amzn2023.0.3.x86_64
libxcb-1.13.1-7.amzn2023.0.2.x86_64
libxcrypt-devel-4.4.33-7.amzn2023.x86_64
libxslt-1.1.34-5.amzn2023.0.2.x86_64
make-1:4.3-5.amzn2023.0.2.x86_64
php8.3-cli-8.3.7-1.amzn2023.0.1.x86_64
php8.3-common-8.3.7-1.amzn2023.0.1.x86_64
php8.3-xml-8.3.7-1.amzn2023.0.1.x86_64
pixman-0.40.0-3.amzn2023.0.3.x86_64
xml-common-0.6.3-56.amzn2023.0.2.noarch

Complete!
[root@ip-172-31-43-114 ~]#
```

d. Next download a test suite ‘wget

<https://github.com/phoronix-test-suite/phoronix-test-suite/archive/refs/heads/master.zip>

```
2024 Fall — root@ip-172-31-43-114:~ — ssh -i aws-auth.pem ec2-user@3.22.98.44 — 101x24

-bash: suite/phoronix-test-suite/archive/refs/heads/master.zip: No such file or directory
[root@ip-172-31-43-114 ~]# wget https://github.com/phoronix-test-suite/phoronix-test-suite/archive/re
fs/heads/master.zip
--2024-09-04 19:45:16-- https://github.com/phoronix-test-suite/phoronix-test-suite/archive/refs/heads/
master.zip
Resolving github.com (github.com)... 140.82.114.4
Connecting to github.com (github.com)|140.82.114.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/phoronix-test-suite/phoronix-test-suite/zip/refs/heads/master [f
ollowing]
--2024-09-04 19:45:16-- https://codeload.github.com/phoronix-test-suite/phoronix-test-suite/zip/refs/
heads/master
Resolving codeload.github.com (codeload.github.com)... 140.82.113.10
Connecting to codeload.github.com (codeload.github.com)|140.82.113.10|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/zip]
Saving to: 'master.zip'

master.zip          [      =>                               ] 10.32M 10.8MB/s   in 1.0s

2024-09-04 19:45:17 (10.8 MB/s) - 'master.zip' saved [10824787]

[root@ip-172-31-43-114 ~]#
```

e. Unzip the file ‘unzip master.zip’

```
2024 Fall — root@ip-172-31-43-114:~ — ssh -i aws-auth.pem ec2-user@3.22.98.44 — 101x24
inflating: phoronix-test-suite-master/pts-core/web-interface/html/about.html
inflating: phoronix-test-suite-master/pts-core/web-interface/html/benchmark.html
inflating: phoronix-test-suite-master/pts-core/web-interface/html/early.html
inflating: phoronix-test-suite-master/pts-core/web-interface/html/settings.html
inflating: phoronix-test-suite-master/pts-core/web-interface/html/test_queue.html
creating: phoronix-test-suite-master/pts-core/web-interface/index.php
creating: phoronix-test-suite-master/pts-core/web-interface/js/
inflating: phoronix-test-suite-master/pts-core/web-interface/js/pts-web-functions.js
inflating: phoronix-test-suite-master/pts-core/web-interface/js/pts-web-interface.js
inflating: phoronix-test-suite-master/pts-core/web-interface/js/pts-web-socket.js
creating: phoronix-test-suite-master/pts-core/web-interface/web-interfaces/
inflating: phoronix-test-suite-master/pts-core/web-interface/web-interfaces/pts_webui_component.php

inflating: phoronix-test-suite-master/pts-core/web-interface/web-interfaces/pts_webui_intro.php
inflating: phoronix-test-suite-master/pts-core/web-interface/web-interfaces/pts_webui_loader.php
inflating: phoronix-test-suite-master/pts-core/web-interface/web-interfaces/pts_webui_main.php
inflating: phoronix-test-suite-master/pts-core/web-interface/web-interfaces/pts_webui_result.php
inflating: phoronix-test-suite-master/pts-core/web-interface/web-interfaces/pts_webui_results.php
inflating: phoronix-test-suite-master/pts-core/web-interface/web-interfaces/pts_webui_search.php
inflating: phoronix-test-suite-master/pts-core/web-interface/web-interfaces/pts_webui_system.php
inflating: phoronix-test-suite-master/pts-core/web-interface/web-interfaces/pts_webui_test.php
inflating: phoronix-test-suite-master/pts-core/web-interface/web-interfaces/pts_webui_tests.php
inflating: phoronix-test-suite-master/release-highlights.md
[root@ip-172-31-43-114 ~]#
```

f. cd phoronix-test-suite-master/ ; ./install-sh

```
2024 Fall — root@ip-172-31-43-114:~/phoronix-test-suite-master — ssh -i aws-auth.pem ec2-us...
inflating: phoronix-test-suite-master/pts-core/web-interface/web-interfaces/pts_webui_component.php

inflating: phoronix-test-suite-master/pts-core/web-interface/web-interfaces/pts_webui_intro.php
inflating: phoronix-test-suite-master/pts-core/web-interface/web-interfaces/pts_webui_loader.php
inflating: phoronix-test-suite-master/pts-core/web-interface/web-interfaces/pts_webui_main.php
inflating: phoronix-test-suite-master/pts-core/web-interface/web-interfaces/pts_webui_result.php
inflating: phoronix-test-suite-master/pts-core/web-interface/web-interfaces/pts_webui_results.php
inflating: phoronix-test-suite-master/pts-core/web-interface/web-interfaces/pts_webui_search.php
inflating: phoronix-test-suite-master/pts-core/web-interface/web-interfaces/pts_webui_system.php
inflating: phoronix-test-suite-master/pts-core/web-interface/web-interfaces/pts_webui_test.php
inflating: phoronix-test-suite-master/pts-core/web-interface/web-interfaces/pts_webui_tests.php
inflating: phoronix-test-suite-master/release-highlights.md
[[root@ip-172-31-43-114 ~]# cd phoronix-test-suite-master/
[root@ip-172-31-43-114 phoronix-test-suite-master]# ./install-sh
/usr/bin/which: no xdg-mime in (/root/.local/bin:/root/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/var/lib/snapd/snap/bin)

Phoronix Test Suite Installation Completed

Executable File: /usr/bin/phoronix-test-suite
Documentation: /usr/share/doc/phoronix-test-suite/
Phoronix Test Suite Files: /usr/share/phoronix-test-suite/
[root@ip-172-31-43-114 phoronix-test-suite-master]#
```

g. Run this command '/usr/bin/phoronix-test-suite install osbench'

```
2024 Fall — root@ip-172-31-43-114:~/phoronix-test-suite-master — ssh -i aws-auth.pem ec2-us...

Phoronix Test Suite v10.8.5

To Install: pts/osbench-1.0.2

Determining File Requirements .....
Searching Download Caches .....

1 Test To Install
  1 File To Download [0.01MB]
  1MB Of Disk Space Is Needed
  3 Seconds Estimated Install Time

pts/osbench-1.0.2:
  Test Installation 1 of 1
  1 File Needed [0.01 MB]
  Downloading: osbench-master-1.zip [0.01MB]
  Downloading .....
  Approximate Install Size: 1 MB
  Estimated Install Time: 3 Seconds
  Installing Test @ 19:46:54

[root@ip-172-31-43-114 phoronix-test-suite-master]#
```

h. Run the benchmark command '/usr/bin/phoronix-test-suite run osbench' and see the options below

```
2024 Fall — root@ip-172-31-43-114:~/phoronix-test-suite-master — ssh -i aws-auth.pem ec2-us...
Xen HVM domU 4.11.amazon testing on Amazon Linux 2023.5.20240819 via the Phoronix Test Suite.

Amazon Linux 2023.5.20240819:

Processor: Intel Xeon E5-2686 v4 (1 Core), Motherboard: Xen HVM domU (4.11.amazon BIOS), Chip
set: Intel 440FX 82441FX PMC, Memory: 949MB, Disk: 8GB, Graphics: Cirrus Logic GD 5446

OS: Amazon Linux 2023.5.20240819, Kernel: 6.1.102-111.182.amzn2023.x86_64 (x86_64), Compiler:
GCC 11.4.1 20230605, File-System: xfs, System Layer: Xen HVM domU 4.11.amazon

OSBench
Test: Create Files
us Per Event < Lower Is Better
Amazon Linux 2023.5.20240819 . 44.04 |=====

OSBench
Test: Create Threads
us Per Event < Lower Is Better
Amazon Linux 2023.5.20240819 . 22.92 |=====

OSBench
Test: Launch Programs
us Per Event < Lower Is Better
Amazon Linux 2023.5.20240819 . 362.59 |=====

OSBench
Test: Create Processes
us Per Event < Lower Is Better
Amazon Linux 2023.5.20240819 . 79.51 |=====

OSBench
Test: Memory Allocations
Ns Per Event < Lower Is Better
Amazon Linux 2023.5.20240819 . 114.31 |=====

Would you like to upload the results to OpenBenchmarking.org (y/n): n

[root@ip-172-31-43-114 phoronix-test-suite-master]#
```

i. Notice the results, were there functioning? Why?

Yes, some results were fluctuating, for example:

Test: Create Files:

45.251336  
43.888119  
42.862239  
44.144062

Test: Create Threads:

22.70937  
23.090839  
22.969246

Test: Launch Programs:

364.95924  
363.030434  
359.768867

Test: Create Processes:

79.898834  
78.091621  
80.530643

Test: Memory Allocations:

90.713024  
116.139889  
116.950989  
115.88192  
115.957022  
118.515015  
115.046024  
114.639044  
117.328167  
116.688013  
117.589951  
116.29796

In my opinion, the reason benchmarks are changing and not staying constant is because there are some background processes in the vm that are using some resources like CPU, memory, disk etc. That is why the numbers are changing. Also process scheduling and CPU caching can affect the performance.

5. Write and compile a simple C/C++ that displays the time of the system. Take a screen shot of the compile and run

```
2024 Fall — ec2-user@ip-172-31-43-114:~ — ssh -i aws-auth.pem ec2-user@3.22.98.44 — 101x22
[[ec2-user@ip-172-31-43-114 ~]$ vi show_time.c
[[ec2-user@ip-172-31-43-114 ~]$ cat show_time.c
#include <stdio.h>
#include <time.h>

int main() {
    time_t current_time;
    time(&current_time);

    printf("Current time: %s", ctime(&current_time));
    return 0;
}
[[ec2-user@ip-172-31-43-114 ~]$ gcc -o show_time show_time.c
[[ec2-user@ip-172-31-43-114 ~]$ ./show_time
Current time: Wed Sep  4 21:15:44 2024
[ec2-user@ip-172-31-43-114 ~]$
```

6. Shutdown the vm and delete it. Always shutdown the VM when not in use.  
a. Execute this: 'shutdown -h +0'

```
2024 Fall — root@ip-172-31-43-114:~ — zsh — 101x22
[[ec2-user@ip-172-31-43-114 ~]$ sudo -i
[[root@ip-172-31-43-114 ~]# shutdown -h +0
Shutdown scheduled for Wed 2024-09-04 21:16:47 UTC, use 'shutdown -c' to cancel.
[root@ip-172-31-43-114 ~]# Connection to 3.22.98.44 closed by remote host.
Connection to 3.22.98.44 closed.
batkhishig@dhcp20 2024 Fall %
```

Then it'll show terminated

Part2:

### In your words, what did we do today in this assignment?

In this assignment, we rented a server resource from AWS, connected to it remotely using ssh key, installed some libraries, ran benchmarks and programs. Finally, we shut it down.

**What IP address was the VM assigned?**

My vm was assigned 3.22.98.44.

**What did sudo do?**

Sudo granted me access to admin privilege. So that I don't have to write sudo every single time when sudo access is required, like installing libraries, shutting it down etc.

**What did yum do?**

Yum command helped me install libraries.

**Was deploying the vm easier to deploy then installing an OS on a physical machine?**

Yes, it was way easier. I just needed to choose options for my vm specifications, setup access key and just press the launch button. Installing an OS on a physical machine would involve much more stages. I would have to purchase hardware, create a bootable device for the OS that I want to install, install the OS on the physical machine and setup(creating user etc).

**What was the purpose of the SSH key you generated?**

The purpose was to connect to my vm remotely from another computer. Ssh is a secure key, so as long as I don't lose my private key, my vm is safe.

**Why do you think you got a random IP address?**

The cloud probably has a limited number of ip addresses, so reusing the same address when vm is terminated improves efficiency of ip address usage and is cost effective.

**What did the test suite do?**

The test suite benchmarked the hardware of the vm that I have created. The tests include, creating files, creating threads, launching programs, creating processes and memory allocations.