Discussion #1:

Using Your Computer / AWS for OpenMP Labs

Karl Marrett, Dan Bochen, Jason Lau, Atefeh Sohrabizadeh, Eddie Huang, Yuze Chi, Peng Wei and Jason Cong

Outline

- Academic Integrity
- Setup the OpenMP Environment on Your Computer
- Sign-up an AWS Educate Account
- Create an AWS EC2 instance
- Labs Requirements

Academic Integrity

- CITE if in doubt! Ask me if still in doubt
- NEVER copy and paste code except from the starter kit
- NEVER show other students your code, even incidentally
- Say “NO” and protect yourself
- Students were sent to the dean’s office every year

OpenMP Labs Environments (Local)

- OpenMP is a specification
- May be implemented differently by different vendors
- What you need is a modern compiler that supports OpenMP
    - GCC 4.2+
    - LLVM Clang 3.8+
    - Intel C Compiler 11+
    - Microsoft Visual C++ Compiler 2017+
- Add compiler options to enable the support

OpenMP Labs Environments (Local)

- macOS users & Linux users 😄

OpenMP Labs Environments (Local)

- **macOS users & Linux users** 😄
- **Windows users** 😖
    - Haven’t been using Windows for 12 years 🙈
    - Try this: docs.microsoft.com/en-us/cpp/parallel/openmp/openmp-in-visual-cpp
        - Compiler option to enable OpenMP: `/openmp`
    - Or use Windows Subsystem for Linux
    - Or use AWS for development (use a cheap instance please)
    - Or even better: Install a virtual machine

OpenMP Labs Environments (macOS)

- **Install Xcode CLT if you haven’t:** `xcode-select --install`
- **Install Homebrew if you haven’t:** https://brew.sh/
- **Install LLVM Clang by invoking:** `brew install llvm`
- **Compile with**
  ```
  /usr/local/opt/llvm/bin/clang++ -fopenmp \
  -L/usr/local/opt/llvm/lib your_code.cpp
  ```

- **Nix Users:** `nix-shell -p gcc    # g++ -fopenmp your_code.cpp`

OpenMP Labs Environments (Linux)

- **Install compilers if you haven't:**
  - **(Ubuntu)** `sudo apt install build-essential`
  - **(CentOS)** `sudo yum groupinstall 'Development Tools'`
  - **Other OS: I assume you know what you need :-)**
- **Compile with** `g++ -fopenmp your_code.cpp`

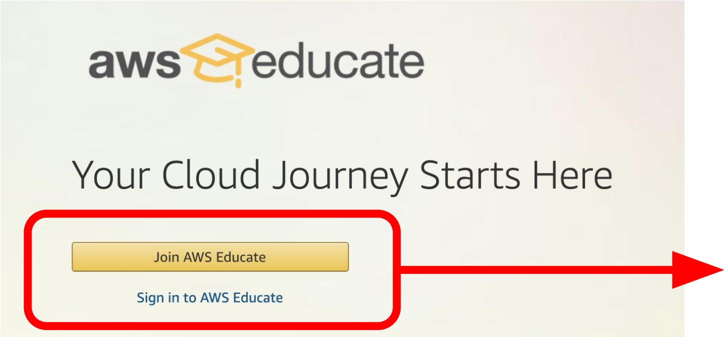
- **Nix Users:** `nix-shell -p gcc   # g++ -fopenmp your_code.cpp`

Sign-up an AWS Educate Account

- Why use AWS?
  - Powerful CPUs
  - Fair comparison of performance
- We will use AWS for grading :-)

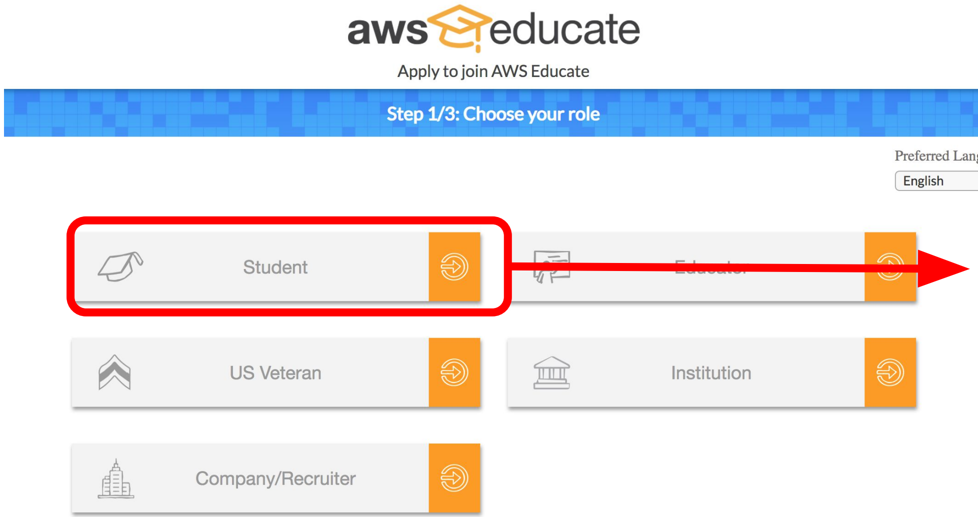
Sign-up an AWS Educate Account

- AWS Educate Starter Account with a $100 in AWS credits
- You must signup AWS Educate with your @ucla.edu email
- Use this link for AWS Educate signup or search “AWS Educate”
    - https://aws.amazon.com/education/awseducate/



Sign-up or Login

Sign-up an AWS Educate Account



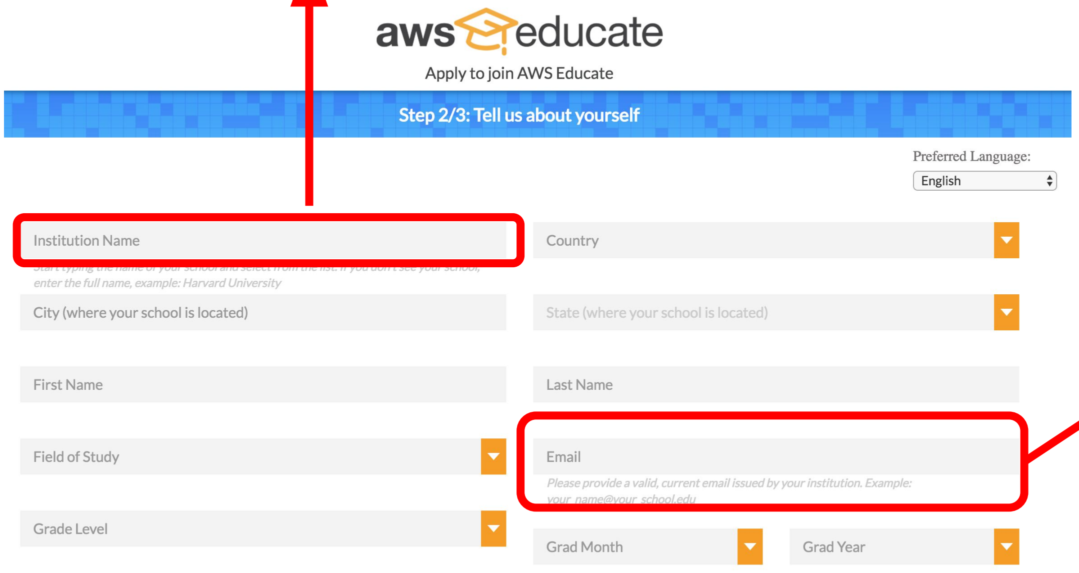
Apply to join AWS Educate

Step 1/3: Choose your role

Preferred Language: English

Sign-up as a Student

Write the complete name (**without comma**) to get the full $100 credit
"University of California Los Angeles"   IMPORTANT: **do NOT enter "UCLA"**



aws educate

Apply to join AWS Educate

Step 2/3: Tell us about yourself

Preferred Language: English

Institution Name

Start typing the name of your school and select from the list. If you don't see your school, enter the full name, example: Harvard University

Country

City (where your school is located)

State (where your school is located)

First Name

Last Name

Field of Study

Email

Please provide a valid, current email issued by your institution. Example: your_name@your_school.edu

Grade Level

Grad Month

Grad Year

Use your **@ucla.edu** email, do NOT use @g.ucla.edu or @cs.ucla.edu or anything else, or you will NOT receive the correct amount of credits

Sign-up an AWS Educate Account

- An email will be sent to your @ucla.edu address for verification
- And then, the application will be reviewed
- Just wait for good news :-)

Oops!!!! What if

I have used up my credits this year for

another course

my personal project

a hackathon

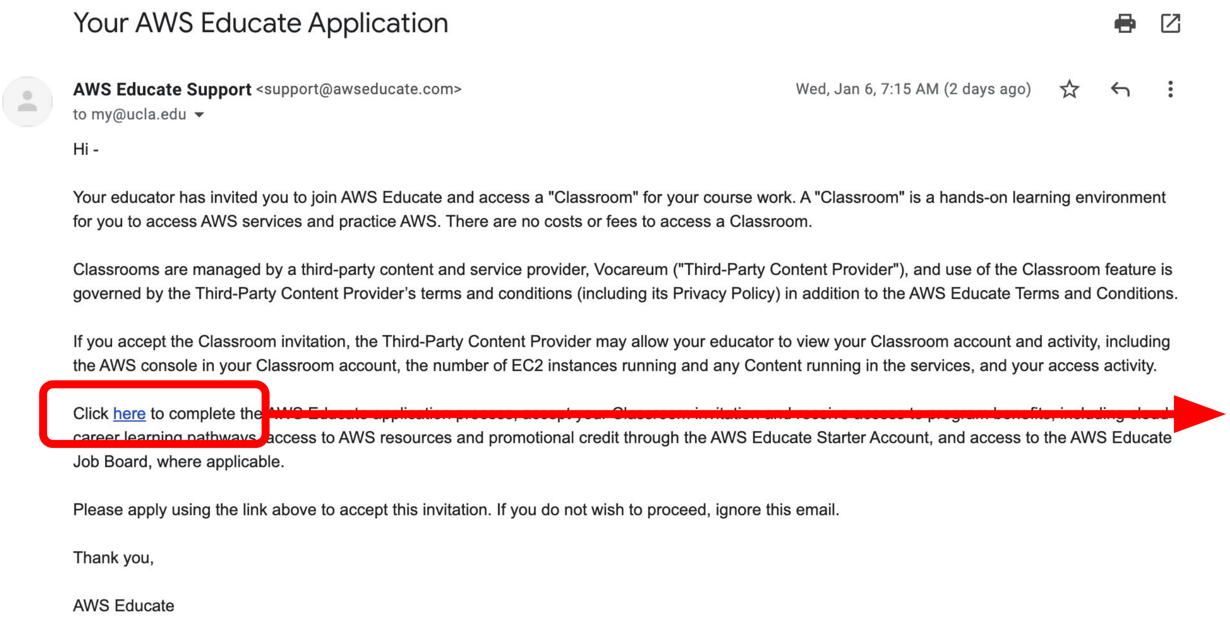
bitcoin mining

fun

……

Use our Classroom credits

- You will probably receive an email by next Wednesday like the following:



Your AWS Educate Application

AWS Educate Support <support@awseducate.com>
to my@ucla.edu

Wed, Jan 6, 7:15 AM (2 days ago)

Hi -

Your educator has invited you to join AWS Educate and access a "Classroom" for your course work. A "Classroom" is a hands-on learning environment for you to access AWS services and practice AWS. There are no costs or fees to access a Classroom.

Classrooms are managed by a third-party content and service provider, Vocareum ("Third-Party Content Provider"), and use of the Classroom feature is governed by the Third-Party Content Provider's terms and conditions (including its Privacy Policy) in addition to the AWS Educate Terms and Conditions.

If you accept the Classroom invitation, the Third-Party Content Provider may allow your educator to view your Classroom account and activity, including the AWS console in your Classroom account, the number of EC2 instances running and any Content running in the services, and your access activity.

Click here to complete the AWS Educate application process, accept your Classroom invitation and receive access to program benefits, including cloud career learning pathways, access to AWS resources and promotional credit through the AWS Educate Starter Account, and access to the AWS Educate Job Board, where applicable.

Please apply using the link above to accept this invitation. If you do not wish to proceed, ignore this email.

Thank you,

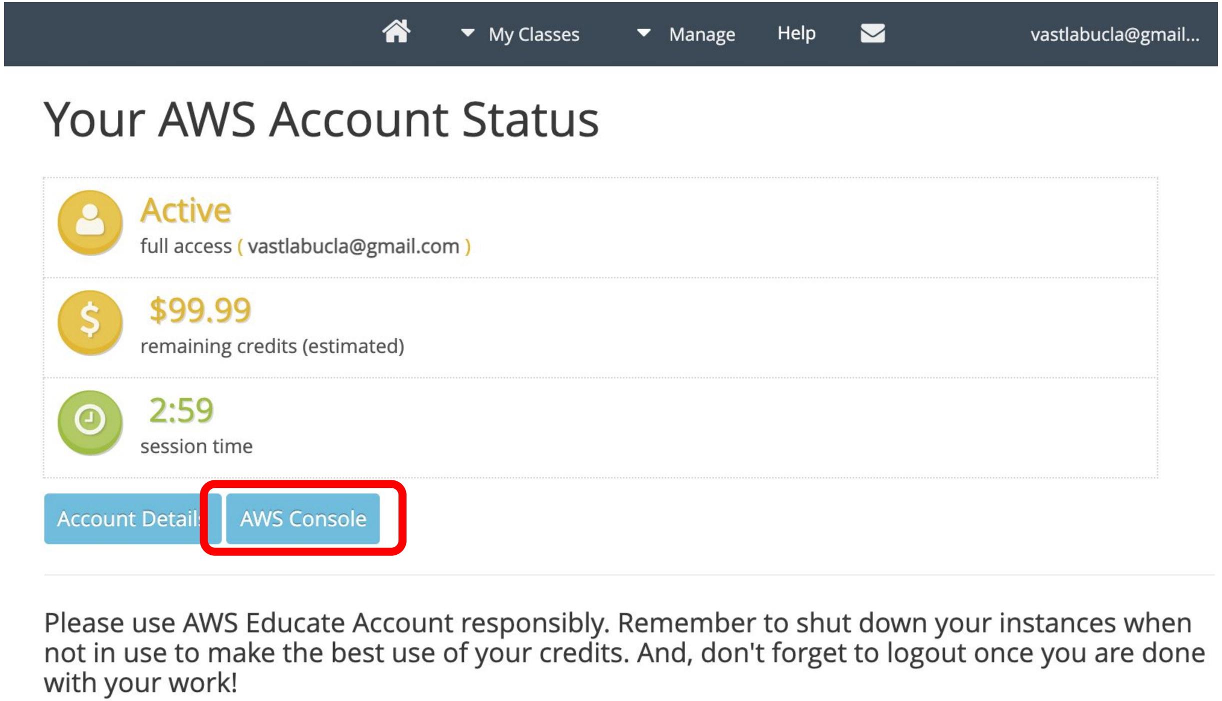
AWS Educate

click here

Use our Classroom credits

- Expect the email by next Wednesday when Lab 1 is released
- If not and if you need the credits:
  - Send an email to kdmarrett@gmail.com

Create an AWS EC2 instance



Your AWS Account Status

Active
full access (vastlabucla@gmail.com)

$99.99
remaining credits (estimated)

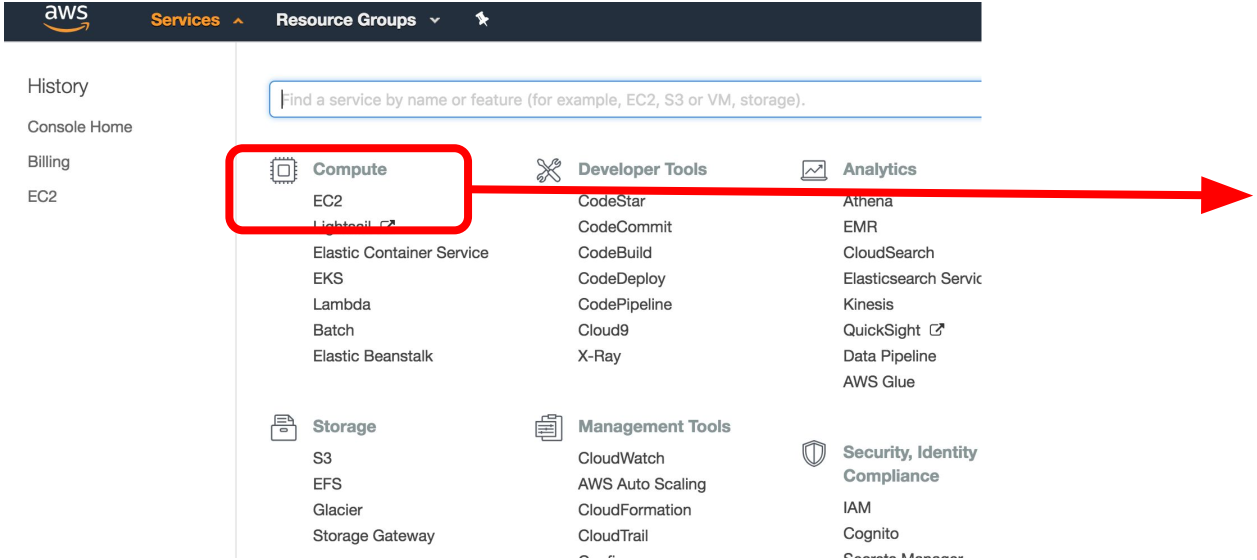
2:59
session time

Please use AWS Educate Account responsibly. Remember to shut down your instances when not in use to make the best use of your credits. And, don't forget to logout once you are done with your work!

log out of your own AWS account first

Create an AWS EC2 instance

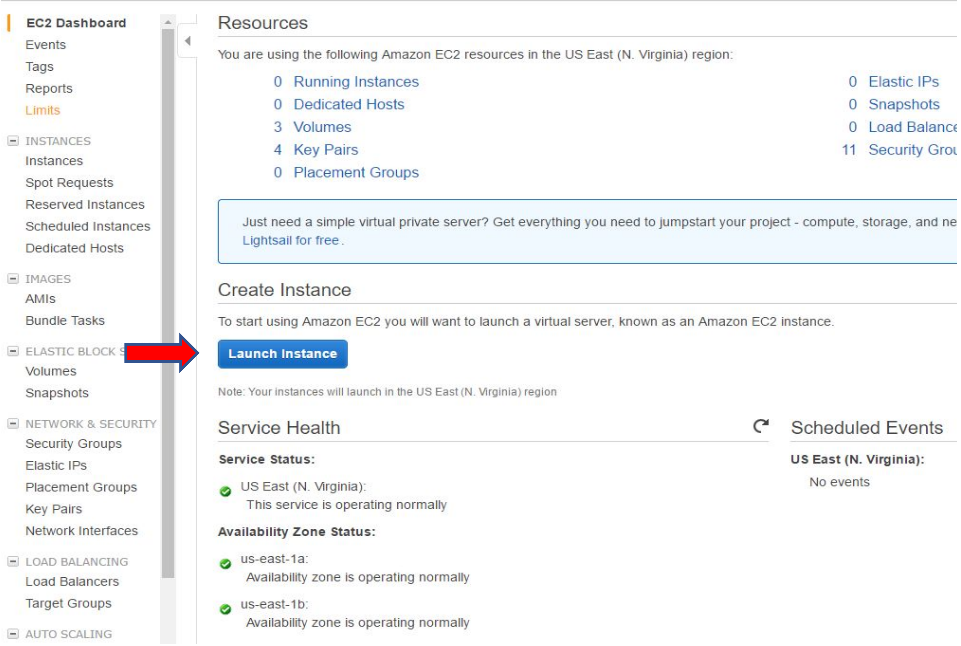
- **Once your account is active, head over to the AWS console, and click on EC2.**



Under “Services”, find and click EC2

Create an AWS EC2 instance

- Then click “Launch Instance”.



Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

0	Running Instances	0	Elastic IPs
0	Dedicated Hosts	0	Snapshots
3	Volumes	0	Load Balance
4	Key Pairs	11	Security Grou
0	Placement Groups		

Just need a simple virtual private server? Get everything you need to jumpstart your project - compute, storage, and ne
Lightsail for free .

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US East (N. Virginia) region

Service Health

Service Status:

US East (N. Virginia):
This service is operating normally

Availability Zone Status:

us-east-1a:
Availability zone is operating normally

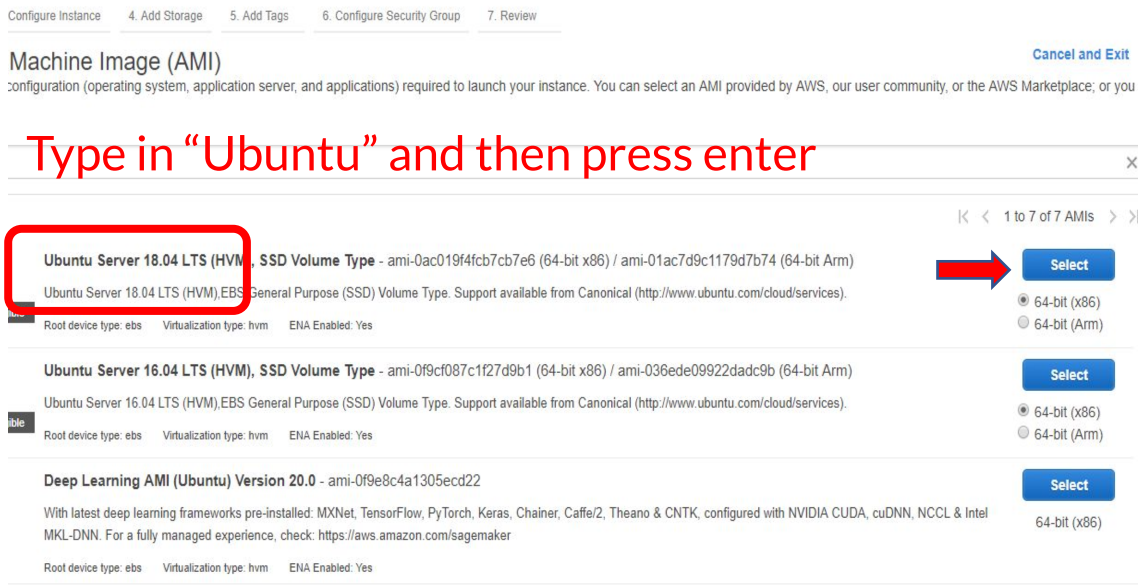
us-east-1b:
Availability zone is operating normally

Scheduled Events

US East (N. Virginia):
No events

Create an AWS EC2 instance: Select AMI

- We'll use different AMIs for different purposes
- For CPU and GPU we use Ubuntu 18.04
- For FPGA we use FPGA Developer AMI or Merlin AMI



Configure Instance   4. Add Storage   5. Add Tags   6. Configure Security Group   7. Review

Machine Image (AMI)

Cancel and Exit

configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you

Type in "Ubuntu" and then press enter

1 to 7 of 7 AMIs

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0ac019f4fcb7cb7e6 (64-bit x86) / ami-01ac7d9c1179d7b74 (64-bit Arm)

Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Root device type: ebs   Virtualization type: hvm   ENA Enabled: Yes

Select

● 64-bit (x86)
○ 64-bit (Arm)

Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-0f9cf087c1f27d9b1 (64-bit x86) / ami-036ede09922dadc9b (64-bit Arm)

Ubuntu Server 16.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Root device type: ebs   Virtualization type: hvm   ENA Enabled: Yes

Select

● 64-bit (x86)
○ 64-bit (Arm)

Deep Learning AMI (Ubuntu) Version 20.0 - ami-0f9e8c4a1305ecd22

With latest deep learning frameworks pre-installed: MXNet, TensorFlow, PyTorch, Keras, Chainer, Caffe/2, Theano & CNTK, configured with NVIDIA CUDA, cuDNN, NCCL & Intel MKL-DNN. For a fully managed experience, check: https://aws.amazon.com/sagemaker

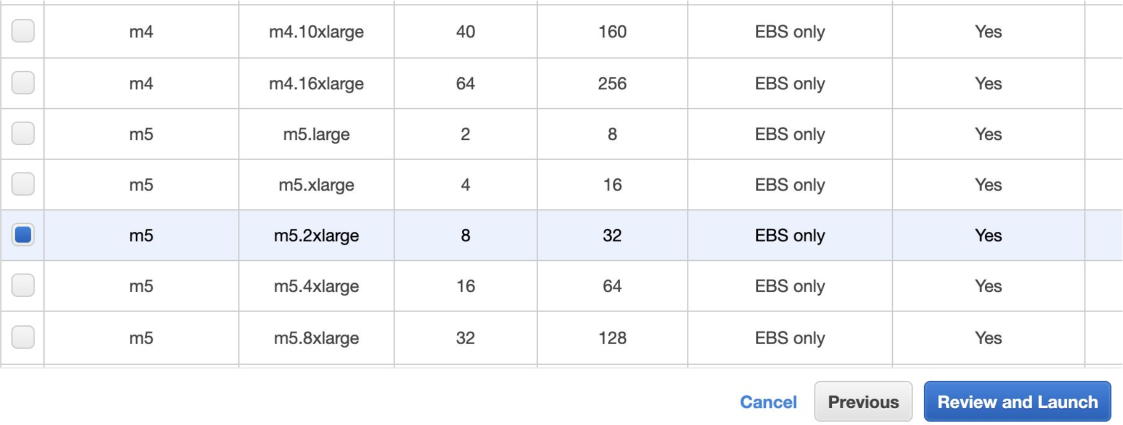
Root device type: ebs   Virtualization type: hvm   ENA Enabled: Yes

Select

64-bit (x86)

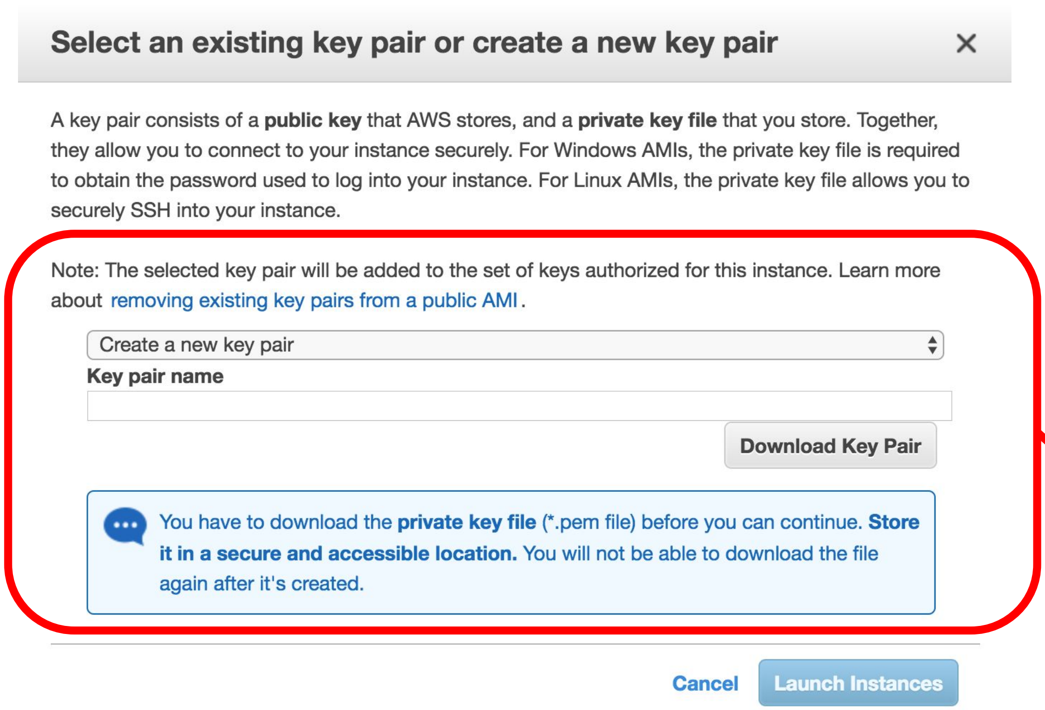
Create an AWS EC2 instance: Select the type

- We’ll use different instance types for different purposes
- For CPU we use m5.2xlarge
- For GPU we use g3s.xlarge
- For FPGA development we use m5.2xlarge
- For FPGA deployment we use f1.2xlarge



□	m4	m4.10xlarge	40	160	EBS only	Yes
□	m4	m4.16xlarge	64	256	EBS only	Yes
□	m5	m5.large	2	8	EBS only	Yes
□	m5	m5.xlarge	4	16	EBS only	Yes
☑	m5	m5.2xlarge	8	32	EBS only	Yes
□	m5	m5.4xlarge	16	64	EBS only	Yes
□	m5	m5.8xlarge	32	128	EBS only	Yes

Create an AWS EC2 instance: Key pair



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI .

Create a new key pair

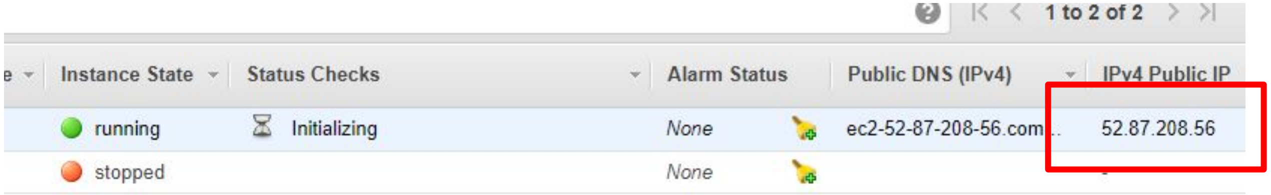
Key pair name

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Create a key pair and use it for SSH

Create an AWS EC2 instance: Log into instance

- **Type in your console:**
- `chmod 600 <your_keypair_file.pem>`
- `ssh -i <your_keypair_file.pem> ubuntu@<ip address>`



e	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
	🟢 running	⌛ Initializing	*None*	ec2-52-87-208-56.com...	52.87.208.56
	🔴 stopped		*None*		-

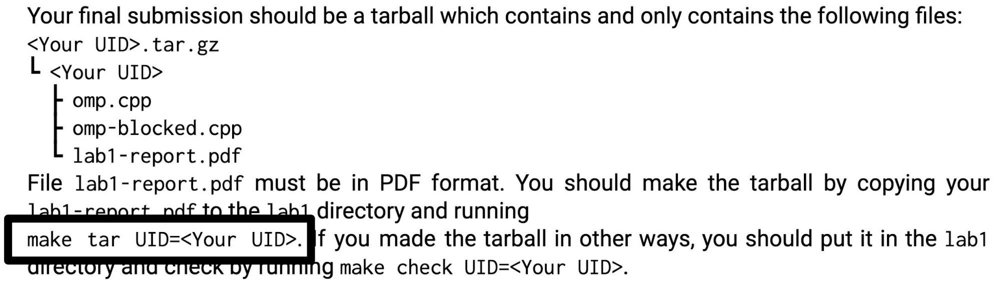
- **Windows users:**
  - ○ http://www.edamamecourse.org/docs/mobaxterm.html

Create an AWS EC2 instance

- Check the price of the instance your are using.
- Keep it on only when you are experimenting with the performance.
- When you need to take a break, make sure you stop the instance!
  - You can start the instance at any time.
- For Ubuntu the default username is ‘ubuntu’.
  FPGA AMI uses CentOS, the default username is ‘centos’.
- When you’re done for the lab, make sure you terminate the instance!
  - You’ll not be charged anymore and your data will be erased.

Labs Requirements

- **Follow the instructions to generate the correct tarball format**



Your final submission should be a tarball which contains and only contains the following files:

```
<Your UID>.tar.gz
└ <Your UID>
  ├ omp.cpp
  ├ omp-blocked.cpp
  └ lab1-report.pdf
```

File `lab1-report.pdf` must be in PDF format. You should make the tarball by copying your `lab1-report.pdf` to the `lab1` directory and running `make tar UID=<Your UID>`. If you made the tarball in other ways, you should put it in the `lab1` directory and check by running `make check UID=<Your UID>`.

- **We will use m5.2xlarge instances for grading**
- **You should submit your tarball to Gradescope and it will be automatically graded**
- **Make sure your code produces correct results**

Use GitHub for code backup

- Register a GitHub account if you haven’t
- Use GitHub PRIVATE repos to manage your code
- Avoid data loss & ease code migration
- We will provide detailed instructions in lab specs