

Tutorial 0 - Getting Started with AWS

Version 0.11

Obtaining a User Account

Objective

The purpose of this tutorial is to describe how to establish AWS account(s) for supporting work in TCSS 462/562.

For the course, please create a personal AWS account using your UW NET ID email or other email account, or obtain an instructor provided account. An AWS account will be required to provide cloud computing resources for tutorials and the Term Project.

Use of a Linux environment is strongly recommended for AWS access.

For Windows 10/11 users, there is an Ubuntu “App” that can be installed onto Windows directly. This provides an Ubuntu Linux environment without the use of Oracle Virtualbox. For this classes, Windows and Mac users should install a Ubuntu virtual machine. Windows users can install Oracle Virtual Box to create virtual machines under Windows 10/11, and then install an Ubuntu virtual machine. Mac users with use UTM instead of Oracle Virtual Box. See Tutorial 1 for more details. **For more information regarding obtaining a Ubuntu environment, please refer to Tutorial #1.**

Task 1 – Creating an AWS account

Create a standard AWS account using your UW email address. This option requires providing a credit card as a backup if the account runs out of cloud computing credits. For this option navigate to the website (<https://aws.amazon.com/>) and click the “Create Account” button:



Create an AWS Account

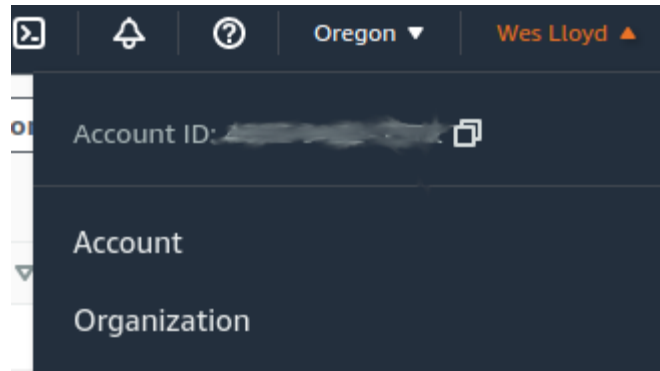
Complete the registration following all instructions.

Once the account is created, please complete the AWS Cloud Credits survey and/or provide your AWS account ID to the instructor to request cloud computing credits. If contacting the instructor by email use **“AWS CREDIT REQUEST”** as the email subject.

In the email, please include the email address used to create the AWS account as well as the 12-digit account ID to identify the account.

The instructor will work to then provide credits which will be directly loaded into the account.

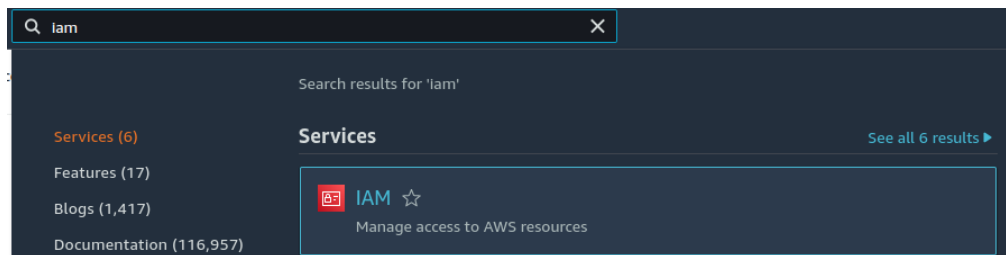
To enter the credit code, in the upper-right hand corner, select your name, and “My Billing Dashboard”. The account ID appears after “Account ID:”. The example is blurred out:



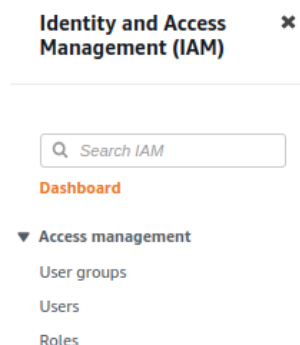
Task 2 – Create AWS Account Credentials

Once having access to AWS, create AWS account credentials to work with virtual machines on EC2, if you have not already done so. Credentials are required to access virtual machines by remote shell (SSH), and also to use the AWS command line interface, and programming APIs.

From the AWS services drop-down list, search for “IAM”, which stands for Identity Access Management. This is under the “Security” group of service, select it:



Optionally, IAM can be accessed by clicking on your name in the upper-right hand corner, and selecting “Security Credentials”. Once in the IAM dashboard, on the left hand-side select “Users”:



You may need to press “Add user”, if there are no users.
If there are users, select your user.

For creating a new users, provide a user account name. Here I am using “tcss462” as an example:

Specify user details

User details

User name

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

☐ Provide user access to the AWS Management Console - *optional*

If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

i If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel
Next

Be sure to select the **“Access key - Programmatic access”** checkbox. If this is the default account “AWS Management Console access” may already be checked. If it is not checked, check it, and follow instructions to configure a password.

Then click the “Next” button...

Now select the “Attach policies directly” option:

Permissions options

☐ **Add user to group**
 Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ **Copy permissions**
 Copy all group memberships, attached managed policies, and inline policies from an existing user.

☒ **Attach policies directly**
 Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

And down below on the screen, using the search box, search for, find, and add by selecting the checkbox the following policies:

* **AmazonEC2FullAccess**

If you plan to use this user account to explore additional Amazon’s services, then admin access should be added:

* **AdministratorAccess**

This will allow you, via the CLI, to explore and do just about everything with this AWS account.

Later in the class you may need to attach additional policies directly to your user account by accessing users under IAM.

Now click the “Next” button.

The screenshot shows the 'Review and create' page for a new IAM user. The page is divided into three main sections: 'User details', 'Permissions summary', and 'Tags - optional'. The 'User details' section shows the user name 'tcss462_f23', console password type 'None', and 'Require password reset' set to 'No'. The 'Permissions summary' section shows two permissions: 'AdministratorAccess' (AWS managed - job function) and 'AmazonEC2FullAccess' (AWS managed), both used as 'Permissions policy'. The 'Tags - optional' section shows 'No tags associated with the resource' and an 'Add new tag' button. At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Create user'.

Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

User name	Console password type	Require password reset
tcss462_f23	None	No

Permissions summary < 1 >

Name	Type	Used as
AdministratorAccess	AWS managed - job function	Permissions policy
AmazonEC2FullAccess	AWS managed	Permissions policy

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tags.

Cancel Previous **Create user**

Now review the settings, make sure they are correct, and click “Create user”.

Now from the users list, select your newly created user account, here it was called “tcss462_f23”:

The screenshot shows the 'Users (6)' list in the AWS IAM console. The table has two columns: 'User name' and 'Path'. There are two users listed: 'tcss462' and 'tcss462_f23', both with a path of '/'. Each row has a checkbox on the left.

<input type="checkbox"/>	User name	Path
<input type="checkbox"/>	tcss462	/
<input type="checkbox"/>	tcss462_f23	/

Click on the ‘Security Credentials’ tab.

Scroll down to the Access Keys widget:

The screenshot shows the 'Access keys (0)' widget in the AWS IAM console. It contains a 'Create access key' button at the top right. Below the button, there is a message: 'No access keys. As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. Learn more'. At the bottom, there is another 'Create access key' button.

Access keys (0)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. [Learn more](#)

[Create access key](#)

No access keys. As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. [Learn more](#)

[Create access key](#)

Click on the ‘Command Line Interface (CLI) button, and select the checkbox that says “I understand the above recommendation and want to proceed to crease an access key.”

Access key best practices & alternatives [Info](#)

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

☒ **Command Line Interface (CLI)**

You plan to use this access key to enable the AWS CLI to access your AWS account.

☐ **Local code**

You plan to use this access key to enable application code in a local development environment to access your AWS account.

☐ **Application running on an AWS compute service**

You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

☐ **Third-party service**

You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.

☐ **Application running outside AWS**

You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.

☐ **Other**

Your use case is not listed here.



Alternatives recommended

- Use [AWS CloudShell](#), a browser-based CLI, to run commands. [Learn more](#)
- Use the [AWS CLI V2](#) and enable authentication through a user in IAM Identity Center. [Learn more](#)

Confirmation

- ☒ I understand the above recommendation and want to proceed to create an access key.

Cancel

Next

Why are we ignoring the advice? For Ubuntu 22.04 LTS, AWS CLI V2 is not yet standard. Ubuntu 22.04 integrates support for AWS CLI V1. With the release of Ubuntu 24.04 LTS (in April 2024) we will migrate to AWS CLI V2. For now, since most folks will use Ubuntu 22.04, we will stick with AWS CLI V1.

Proceed to press “Next”.

The description tag is optional and can be skipped.;

Proceed to press “Create access key”

There is one, and only one opportunity to download your Access key and Secret access key pairs. You can press the **Show** button and copy and paste the key somewhere safe, or download the .csv file. If you fail to download or copy the key values now, they will be lost forever, and this process will need to be repeated.

The keys must be copied, or the .csv file download to save the values:

Retrieve access keys [Info](#)

Access key
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIAWYW5J4AKNH6M5CZ5	***** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

[Download .csv file](#) [Done](#)

Once you've downloaded these keys, be sure to **never** publish these key values in a source code repository such as github where your account credentials could be exposed. **Protect these keys as if they were your credit card or wallet!**

Task 3 – Install and Configure the AWS Command Line Interface (CLI)

On your Ubuntu machine, after obtaining your access key and secret key, install the AWS command line interface:

```
sudo apt install awscli
```

After installing, configure the CLI to use your credentials using "aws configure".
Provide your access key and secret key from Task 2.
Specify "us-east-2" (Ohio) as the default region.
Leave the default output as none. Most output will be returned in JSON format.

```
$ aws configure  
AWS Access Key ID [None]: XXXXXXXXXXXXXXXXXXXX  
AWS Secret Access Key [None]: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
Default region name [None]: us-east-2  
Default output format [None]:
```

Check the version of the AWS CLI that's been installed:

```
#value shown is for Ubuntu 20.04  
$ aws --version  
aws-cli/1.18.69 Python/3.8.10 Linux/5.11.0-37-generic botocore/1.16.19
```

This version will likely be 1.22.34 if using Ubuntu 22.04.
Now try inspecting the available AWS CLI commands:

```
$ aws help
```

Now try lists the default Virtual Private Clouds (VPCs) that are preconfigured in your account to provide networking for virtual machines:

```
$ aws ec2 describe-vpcs
```

Once launching a virtual machine in Tutorial #3, you can inspect any running virtual machines from the Ubuntu command line with the command:

```
$ aws ec2 describe-instances
```

Document History:

v.10 Initial version

v.11 updated version to show new user interface