===
=

Intercomparison of Historical Temperature Anomalies in Climate Models

GUIDE:
Configuration of AWS AMI for CS205 Group 7 Project

By Eimy Bonilla, Peter Sherman, Matt Stewart

May 9, 2018

===
=

Abstract

This is a step-by-step guide of configuring and installing necessary dependencies and software packages required to run Dask and AWS Simple System Manager (SSM) on an AWS instance. This instance was then used to create an Amazon Machine Image (AMI) that was used in the analysis of multiple NASA CMIP5 climate models.

Notes

- This guide has been prepared for connecting with remote instance using Linux or Mac OS.
- The Amazon Machine Image (AMI) used for this guide was an **Ubuntu Server 16.04 LTS(HVM), SSD Volume type**-64-bit.
- The instance used for this process was a **t2.2xlarge**.
- Edit storage for instance, by setting aside at least 50 GB.
- CS205 course-key was used.

1. Preparing EC2 instance with updates and configuring

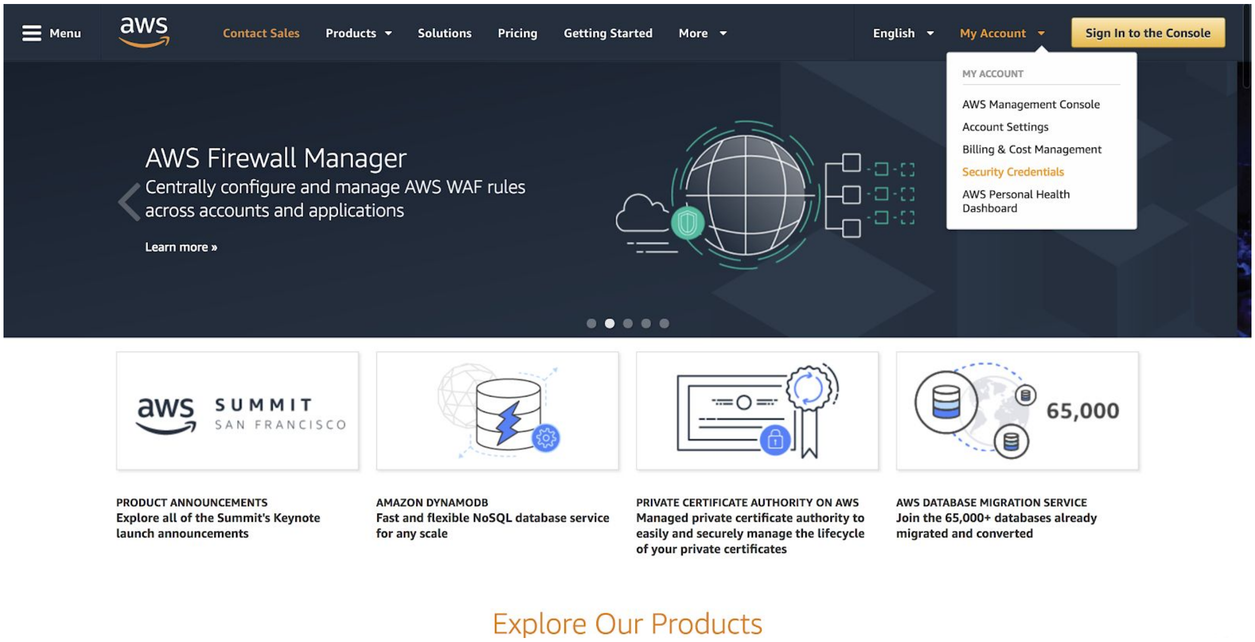
```
$ sudo apt-get update

$ sudo apt-get install gcc gfortran

$ sudo apt install awscli

$ aws configure
```

CLICK “CREATE NEW ACCESS KEY” AND USE THE KEY ID AND SECRET ID IN THE CONFIGURE BY FOLLOWING THE NEXT STEPS ON AWS



AWS Firewall Manager

Centrally configure and manage AWS WAF rules across accounts and applications

Learn more »

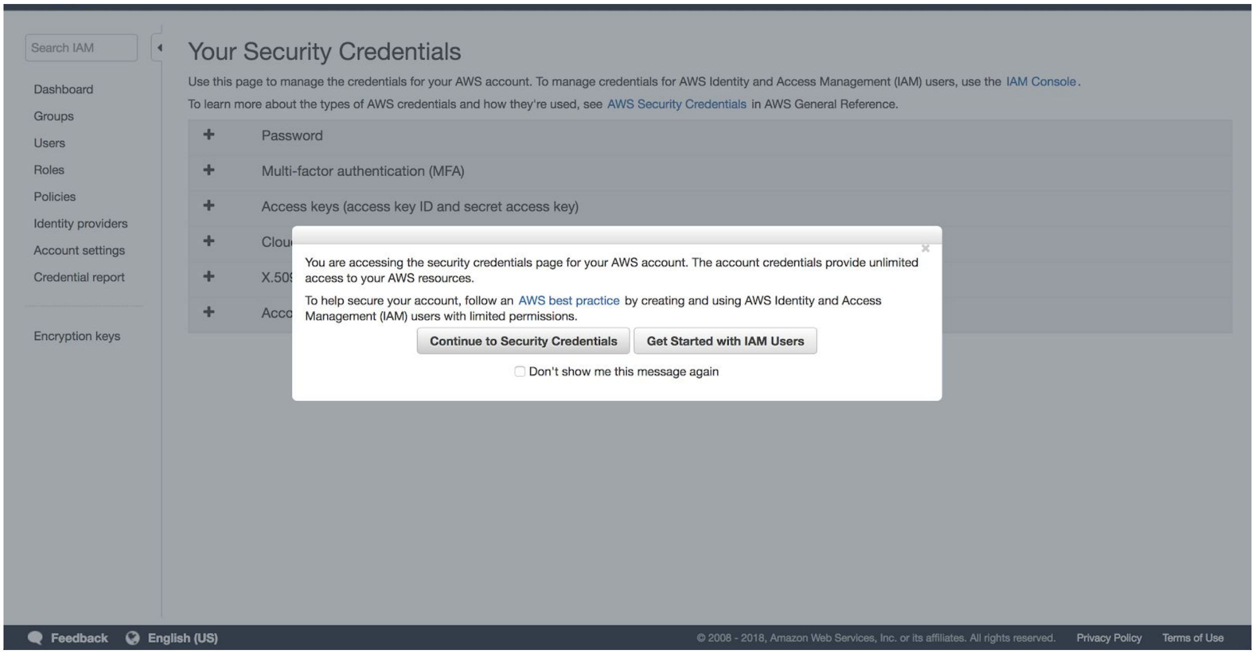
PRODUCT ANNOUNCEMENTS
Explore all of the Summit's Keynote launch announcements

AMAZON DYNAMODB
Fast and flexible NoSQL database service for any scale

PRIVATE CERTIFICATE AUTHORITY ON AWS
Managed private certificate authority to easily and securely manage the lifecycle of your private certificates

AWS DATABASE MIGRATION SERVICE
Join the 65,000+ databases already migrated and converted

Explore Our Products



Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Console.

To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference.

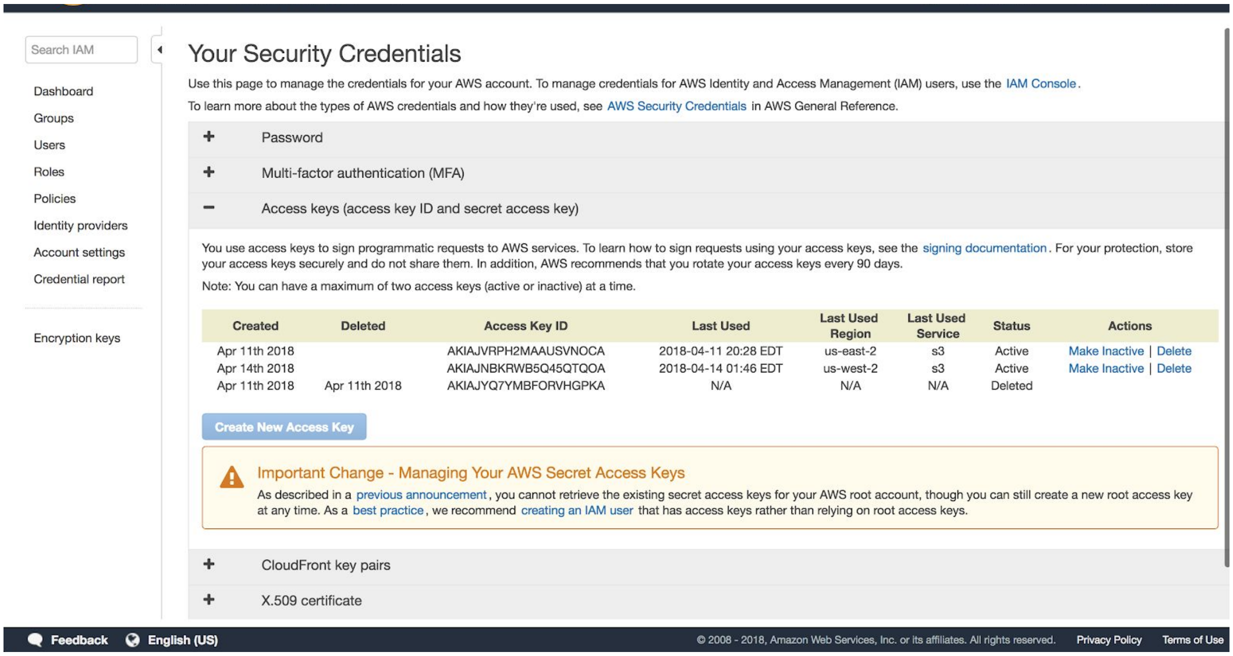
+ Password
+ Multi-factor authentication (MFA)
+ Access keys (access key ID and secret access key)
+ Clou
+ X.50
+ Acco

You are accessing the security credentials page for your AWS account. The account credentials provide unlimited access to your AWS resources.

To help secure your account, follow an AWS best practice by creating and using AWS Identity and Access Management (IAM) users with limited permissions.

□ Don't show me this message again

Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Console.

To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference.

+ Password

+ Multi-factor authentication (MFA)

− Access keys (access key ID and secret access key)

You use access keys to sign programmatic requests to AWS services. To learn how to sign requests using your access keys, see the signing documentation. For your protection, store your access keys securely and do not share them. In addition, AWS recommends that you rotate your access keys every 90 days.

Note: You can have a maximum of two access keys (active or inactive) at a time.

Created	Deleted	Access Key ID	Last Used	Last Used Region	Last Used Service	Status	Actions
Apr 11th 2018		AKIAJVRPH2MAAUSVNOCA	2018-04-11 20:28 EDT	us-east-2	s3	Active	Make Inactive | Delete
Apr 14th 2018		AKIAJNBKRWB5Q45QTQOA	2018-04-14 01:46 EDT	us-west-2	s3	Active	Make Inactive | Delete
Apr 11th 2018	Apr 11th 2018	AKIAJYQ7YMBFORVHGPKA	N/A	N/A	N/A	Deleted	

Important Change - Managing Your AWS Secret Access Keys

As described in a previous announcement, you cannot retrieve the existing secret access keys for your AWS root account, though you can still create a new root access key at any time. As a best practice, we recommend creating an IAM user that has access keys rather than relying on root access keys.

+ CloudFront key pairs

+ X.509 certificate

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

2. Installing libnetdff and linetcdff packages on the instance to be able to create, access, and sharing of scientific data in Fortran

```
$ sudo apt-get install libnetcdf-dev libnetcdff-dev
```

3. Installing Anaconda to Instance

```
$ wget
https://repo.continuum.io/archive/Anaconda2-4.1.1-Linux-x86_64
.sh

$ bash Anaconda2-4.1.1-Linux-x86_64.sh

$ source .bashrc
```

Installing packages to instance using conda command. Also, installing dask and common dependencies

```
$ conda install numpy pandas h5py Pillow scipy toolz pytables
fastparquet xarray dask

$ pip install netcdf4
```

4. Attempt to access S3 bucket to see if AWSCLI is configured correctly

```
$ aws s3 cp s3://nasanex/NEX-GDDP/BCSD… ./
```

To recursively download data for an entire model, the following command can be run and ACCESS1-0 can be changed with the appropriate model name. Years can also be filtered in a similar manner.

```
$ aws s3 cp
s3://nasanex/NEX-GDDP/BCSD/rcp45/day/atmos/tasmax/r1i1p1/v1.0/
~/mean_folder/ --recursive --exclude “*” --include
“*ACCESS1-0*” --exclude "*.json"
```

5. Install other required packages

```
$ pip install zarr
$ pip install tqdm
```

6. Install SSM agent onto instance

```
$ sudo apt-get install upstart-sysv -y
$ sudo update-initramfs -u
$ sudo reboot

$ wget
https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest
/debian_amd64/amazon-ssm-agent.deb

$ sudo dpkq -i amazon-ssm-agent.deb
```

7. To check the SSM agent is running on the instance, run the following:

```
$ sudo status amazon-ssm-agent
```

8. Then attach the IAM policy ‘CS205’ to the instance which contains policies allowing full access to SSM and S3, then run the following and replace the instance ID, region, and such, with a running instance to check commands can be sent via SSM:

```
$ aws ssm send-command --document-name "AWS-RunShellScript"
--instance-ids "i-04801b9e6a59f0dc5" --parameters
'{"commands":["bash test.sh"],"executionTimeout":["3600"]}'
--timeout-seconds 600 --region us-west-2
```

Instances can be started using AWSCLI as an SSM command with the following command, given a preconfigured AMI.

```
 $ instance_id=( ${instance_id[@]} $(aws ec2 run-instances
--region us-west-2 --key cs205-HWB --instance-type m4.4xlarge
--subnet-id subnet-2662aa5f --security-group-ids sg-2eddfb50
--count 1 --image-id ami-f1334289 --output text --query
'Instances[*].InstanceId') )
```

This command creates a new instance and concatenates the instance ID to an already declared array of instance IDs.