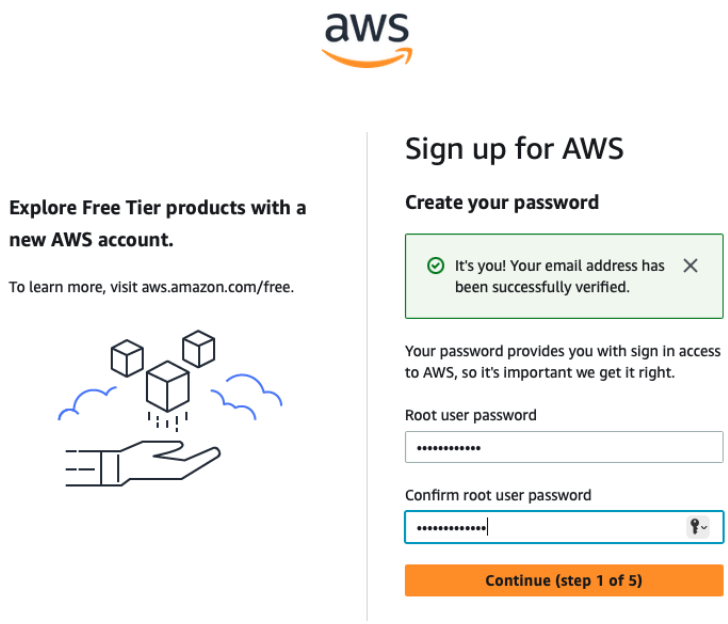
Vicente De Leon

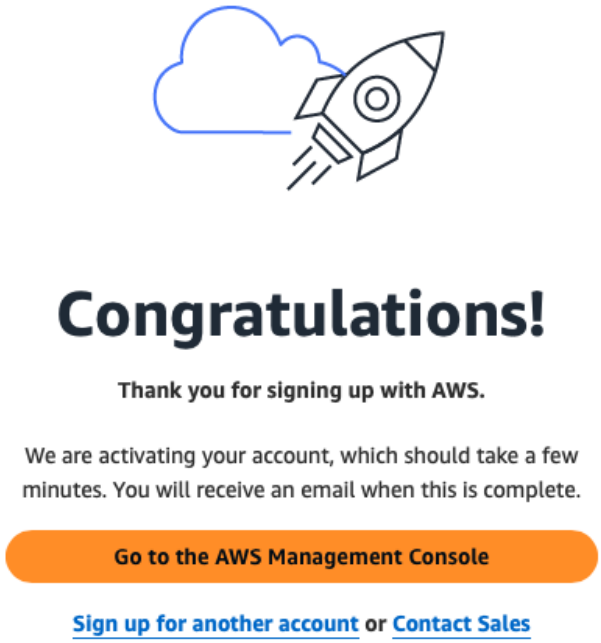
Big Data Applications

Indiana University

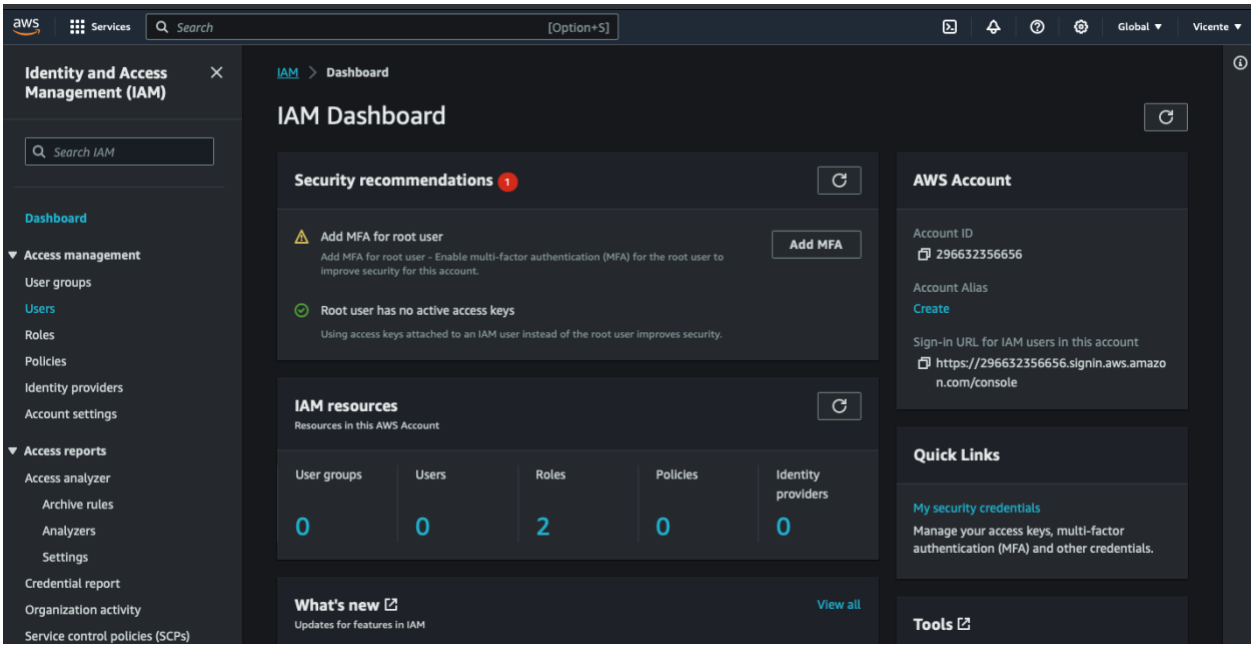
Homework 6 – AWS

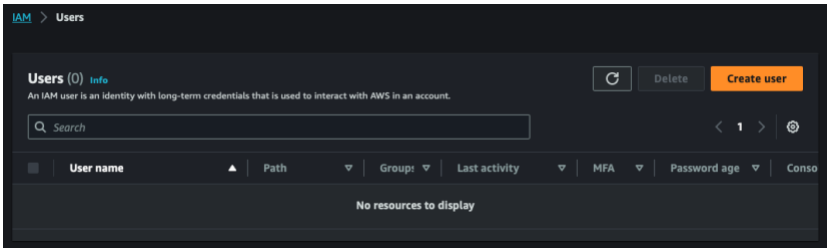
1. Creating AWS Account:

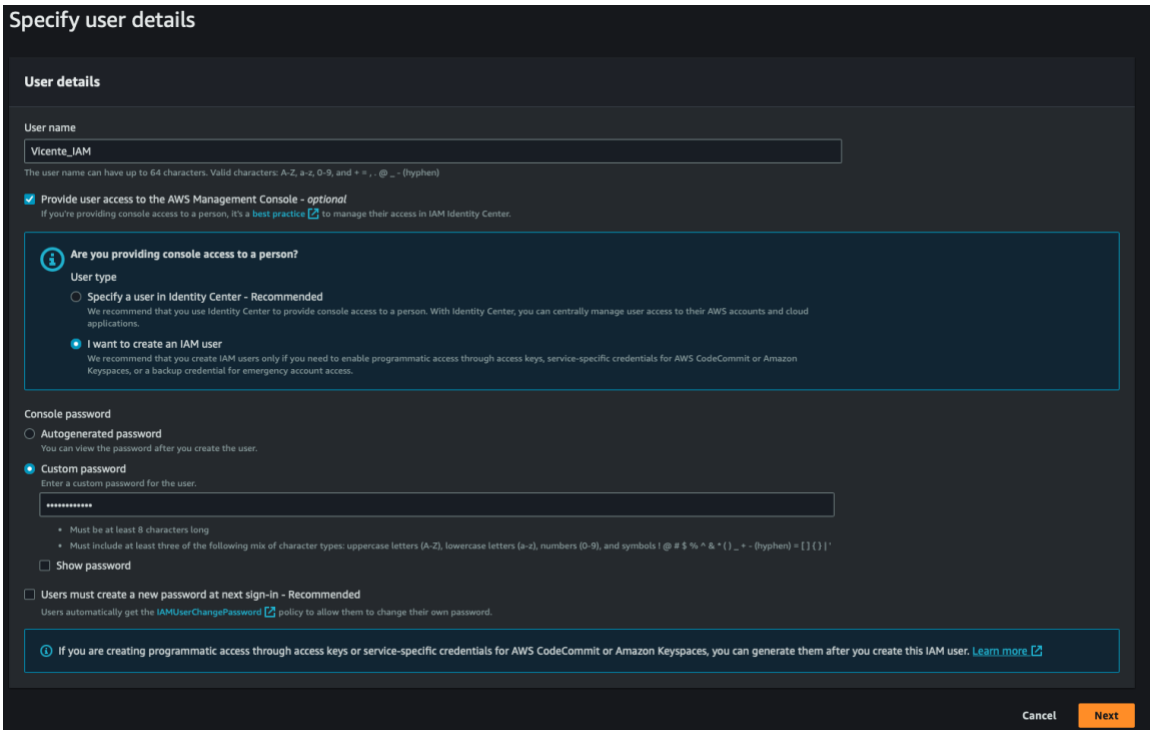




2. Create IAM user with pragmatic and console access:

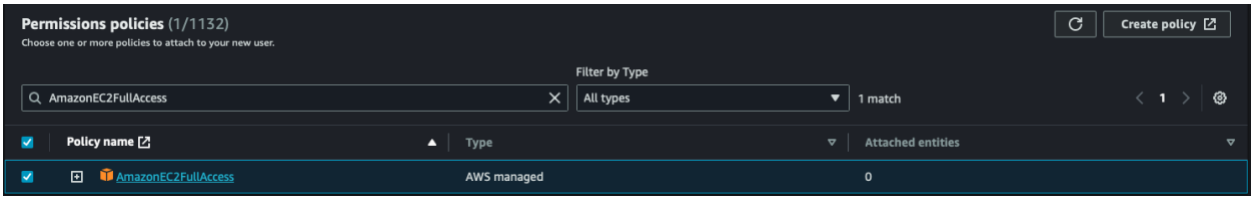


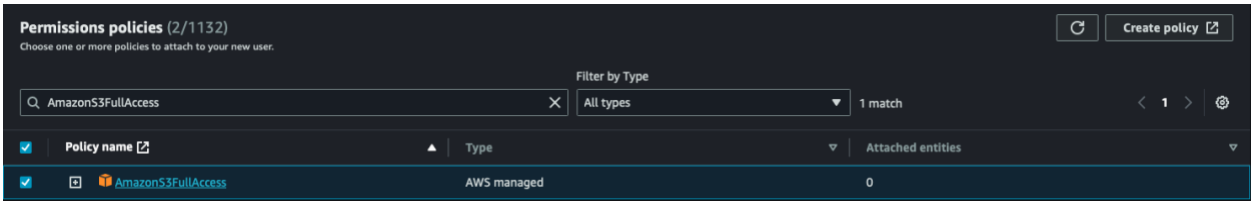


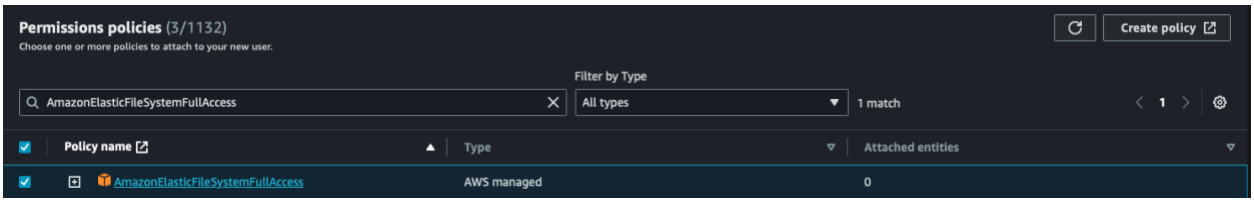


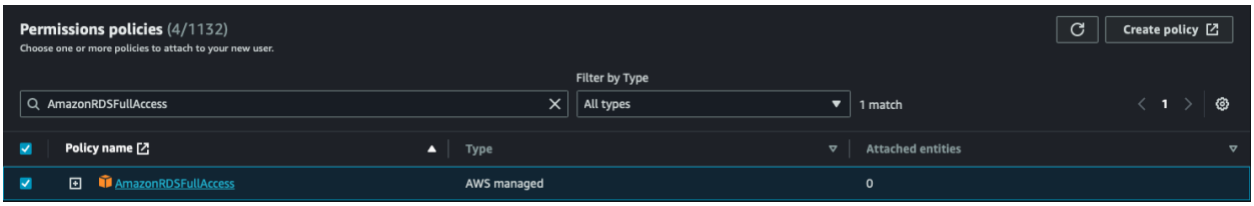
For the “Attach policies directly” permission policies I think I will grant permission to the following (this is what I have in my CV – related to this class). Since there are many policies, I’m just going to go ahead with the “FullAccess”:

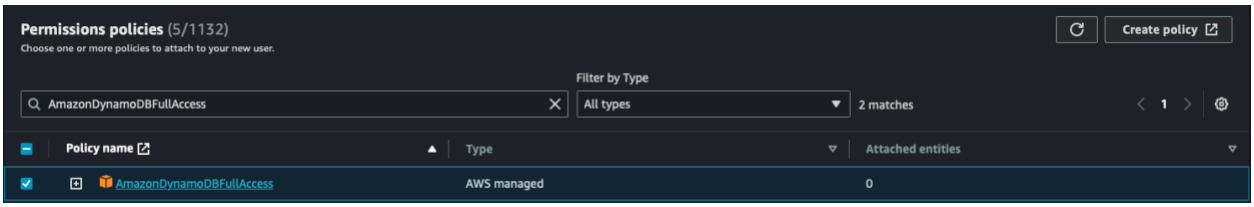
AWS (EC2, S3, EBS, EFS, RDS, DynamoDB, SageMaker).

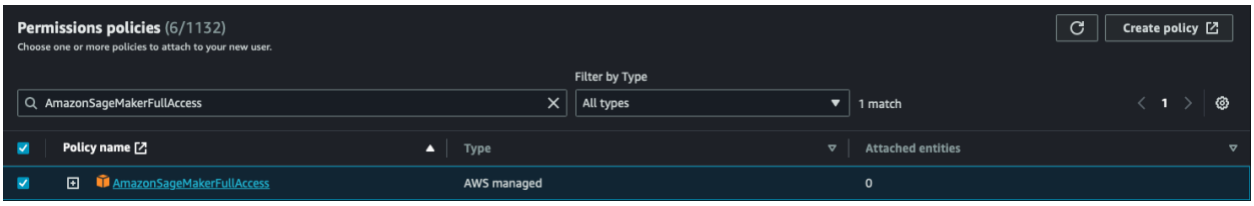


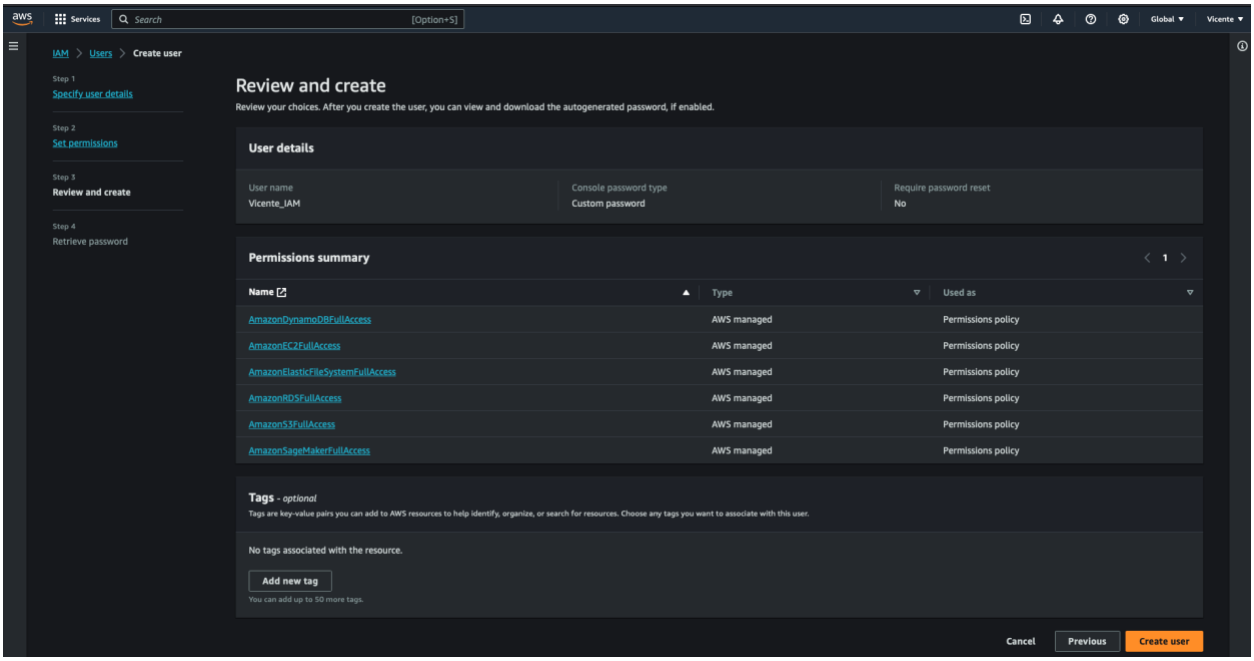


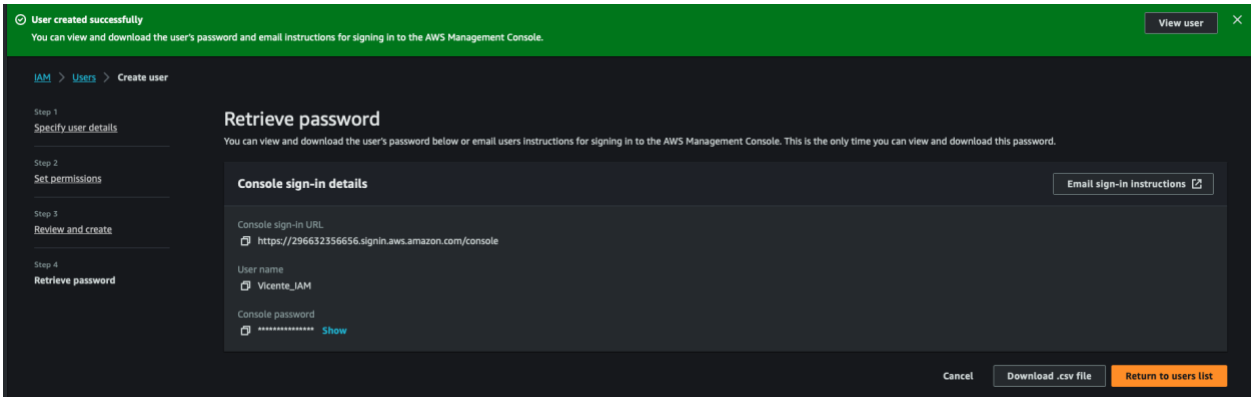






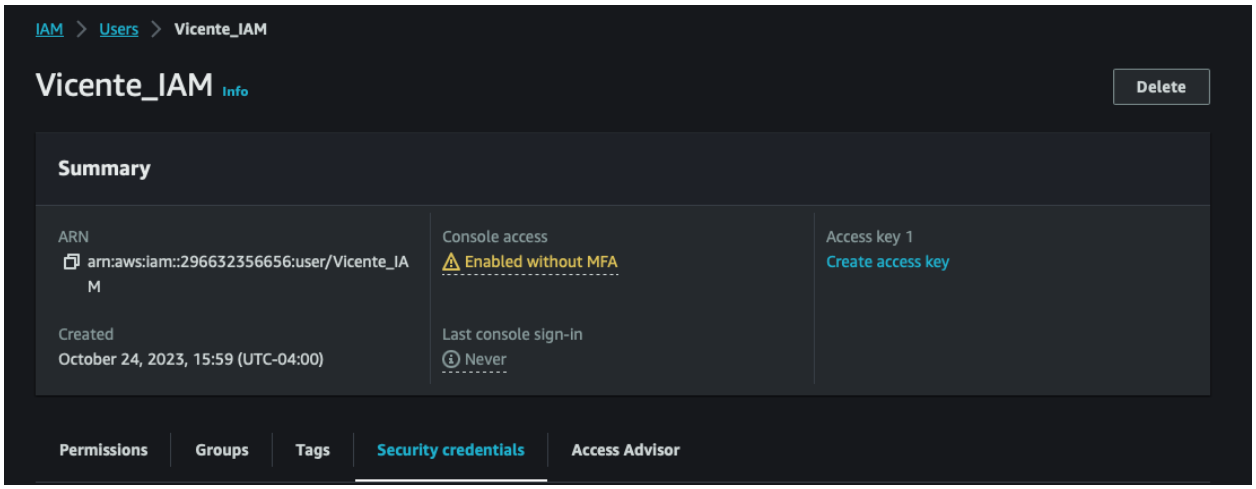


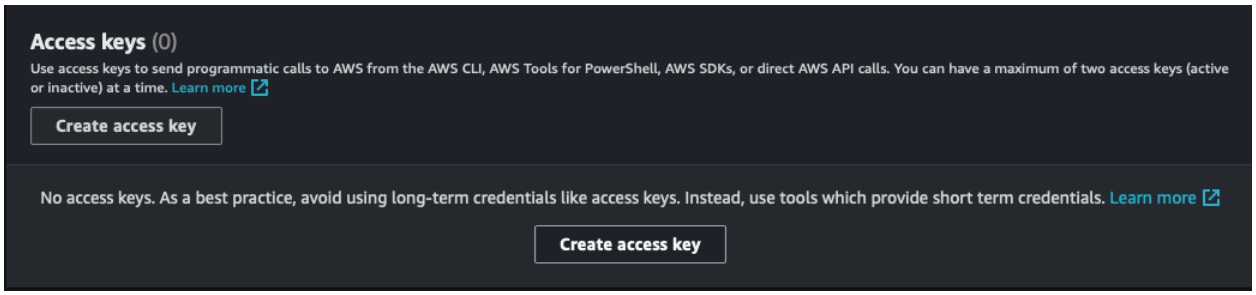




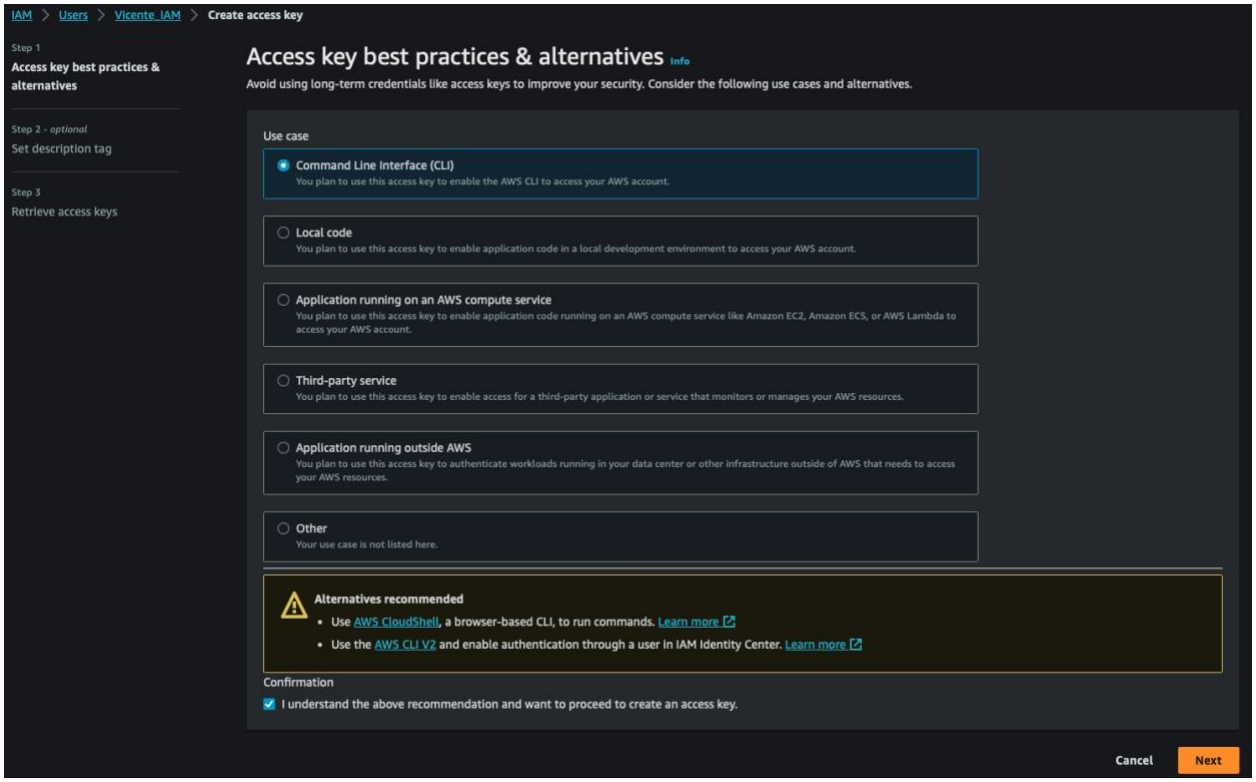
Reference: https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html

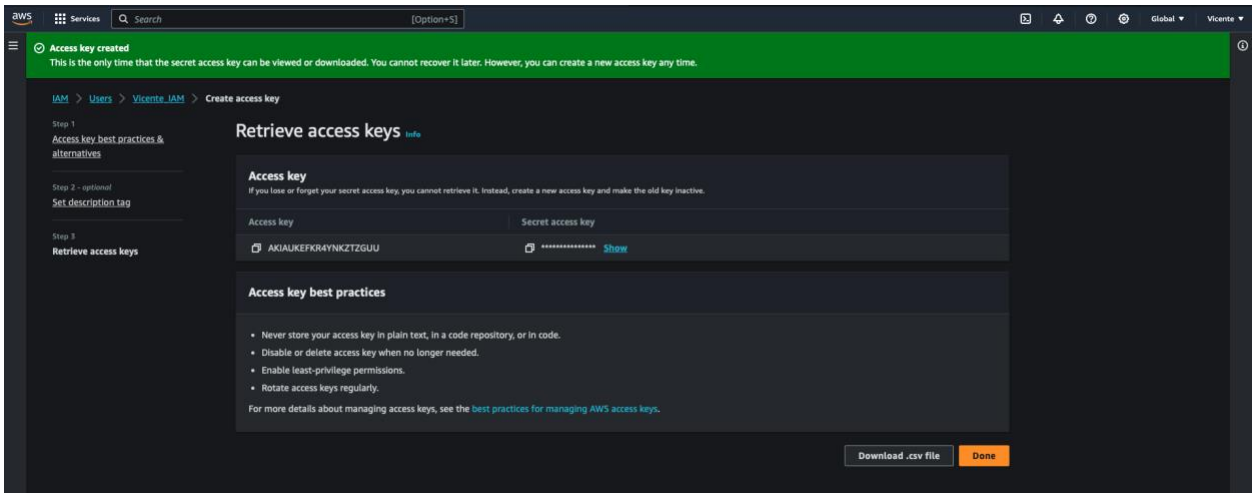
3. Generate an access key and secret access key for the IAM user.





Based on HW 6 – CLI option:




aws
Services
Search
[Option+S]
Global
Vicente
Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.
IAM > Users > Vicente_IAM > Create access key
Step 1
Access key best practices & alternatives
Step 2 - optional
Set description tag
Step 3
Retrieve access keys
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
AKIAUKEFKR4YNKZTZGUU
********** 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

4. Install the AWS Command Line Interface (CLI) in local machine:

- We can use Homebrew to install awscli.
- We can also use pip to install awscli.

I think I will try and use pip for this task.

```
(base) deleonv@Vicentes-MacBook-Air ~ % pip install awscli
Collecting awscli
  Downloading awscli-1.29.70-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: botocore==1.31.70 in ./opt/anaconda3/lib/python3.9/site-packages (from awscli) (1.31.70)
Requirement already satisfied: docutils<0.17,>=0.10 in ./opt/anaconda3/lib/python3.9/site-packages (from awscli) (0.16)
Requirement already satisfied: s3transfer<0.8.0,>=0.7.0 in ./opt/anaconda3/lib/python3.9/site-packages (from awscli) (0.7.0)
Requirement already satisfied: PyYAML<6.1,>=3.10 in ./opt/anaconda3/lib/python3.9/site-packages (from awscli) (6.0)
Requirement already satisfied: colorama<0.4.5,>=0.2.5 in ./opt/anaconda3/lib/python3.9/site-packages (from awscli) (0.4.4)
Requirement already satisfied: rsa<4.8,>=3.1.2 in ./opt/anaconda3/lib/python3.9/site-packages (from awscli) (4.7.2)
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in ./opt/anaconda3/lib/python3.9/site-packages (from botocore==1.31.70->awscli) (0.10.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in ./opt/anaconda3/lib/python3.9/site-packages (from botocore==1.31.70->awscli) (2.8.2)
Requirement already satisfied: urllib3<1.27,>=1.25.4 in ./opt/anaconda3/lib/python3.9/site-packages (from botocore==1.31.70->awscli) (1.26.11)
Requirement already satisfied: pyasn1>=0.1.3 in ./opt/anaconda3/lib/python3.9/site-packages (from rsa<4.8,>=3.1.2->awscli) (0.4.8)
Requirement already satisfied: six>=1.5 in ./opt/anaconda3/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->botocore==1.31.70->awscli) (1.16.0)
Downloading awscli-1.29.70-py3-none-any.whl (4.3 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.3/4.3 MB 15.8 MB/s eta 0:00:00
Installing collected packages: awscli
Successfully installed awscli-1.29.70
(base) deleonv@Vicentes-MacBook-Air ~ % aws --version
aws-cli/1.29.70 Python/3.9.12 Darwin/22.6.0 botocore/1.31.70
```

5. Configure AWS CLI using Access Key ID and Secret Access Key:

```
(base) deleonv@Vicentes-MacBook-Air ~ % aws configure
AWS Access Key ID [None]: [redacted]
AWS Secret Access Key [None]: [redacted]
Default region name [None]:
Default output format [None]:
(base) deleonv@Vicentes-MacBook-Air ~ %
```

References:

Installation: https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html

AWSCLI: https://stackoverflow.com/questions/72496253/aws-cli-m1-chip-installation

AWCLI Homebrew: https://formulae.brew.sh/formula/awscli

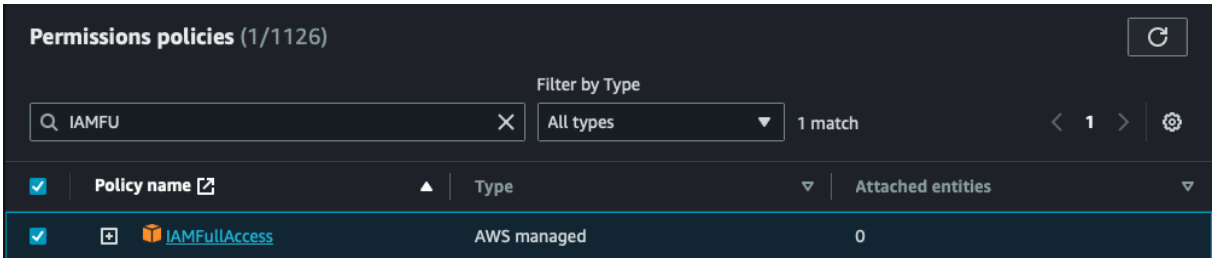
AWS CLI PIP: https://medium.com/@yogeshdarji/steps-to-install-awscli-on-mac-5bad783483a

AWS CLI PIP: https://pypi.org/project/awscli/

AWS CLI configuration: https://docs.aws.amazon.com/cli/latest/reference/configure/

6. Use the AWS CLI to create a new IAM group and add the IAM user to that group:

I was having trouble creating the group, but then I grant permission:

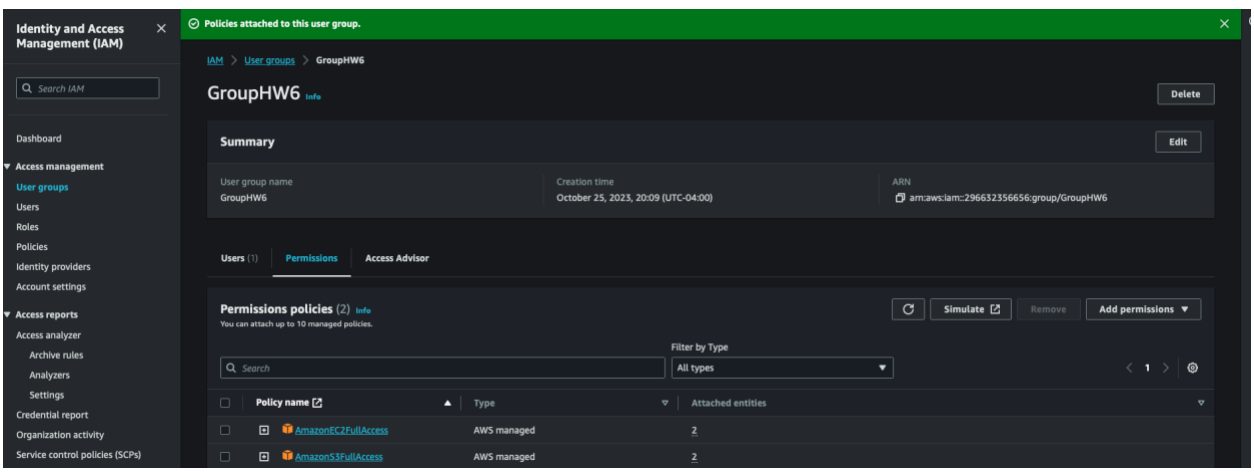


After adding “AWSFullAccess”, then I created the group “GroupHW6” and added “Vicente_IAM” as user:

```
(base) deleonv@Vicentes-MacBook-Air ~ % aws iam create-group --group-name GroupHW6

An error occurred (AccessDenied) when calling the CreateGroup operation: User: arn:aws:iam::296632356656:user/Vicente_IAM is not authorized to
perform: iam:CreateGroup on resource: arn:aws:iam::296632356656:group/GroupHW6 because no identity-based policy allows the iam:CreateGroup acti
on
(base) deleonv@Vicentes-MacBook-Air ~ % aws iam create-group --group-name GroupHW6
{
    "Group": {
        "Path": "/",
        "GroupName": "GroupHW6",
        "GroupId": "AGPAUKEFKR4YJACUI2PIQ",
        "Arn": "arn:aws:iam::296632356656:group/GroupHW6",
        "CreateDate": "2023-10-26T00:09:53Z"
    }
}
(base) deleonv@Vicentes-MacBook-Air ~ % aws iam add-user-to-group --user-name Vicente_IAM --group-name GroupHW6
```

Adding the two permissions we might use in the following steps (AmazonS3FullAccess) and (AmazonEC2FullAccess):

```
(base) deleonv@Vicentes-MacBook-Air ~ % aws iam attach-group-policy --group-name GroupHW6 --policy-arn arn:aws:iam::aws:policy/AmazonEC2FullAccess
(base) deleonv@Vicentes-MacBook-Air ~ % aws iam attach-group-policy --group-name GroupHW6 --policy-arn arn:aws:iam::aws:policy/AmazonS3FullAccess
(base) deleonv@Vicentes-MacBook-Air ~ %
```



References:

Creating groups using CLI: https://docs.aws.amazon.com/cli/latest/reference/iam/create-group.html

Adding user into group: https://docs.aws.amazon.com/IAM/latest/UserGuide/id_groups_manage_add-remove-users.html

Adding policies: https://docs.aws.amazon.com/cli/latest/reference/iam/attach-group-policy.html

7. Creating an S3 bucket using the AWS CLI. Configure the bucket to allow public read access for object.

Since I have “us-east-2” by default I will do the following:

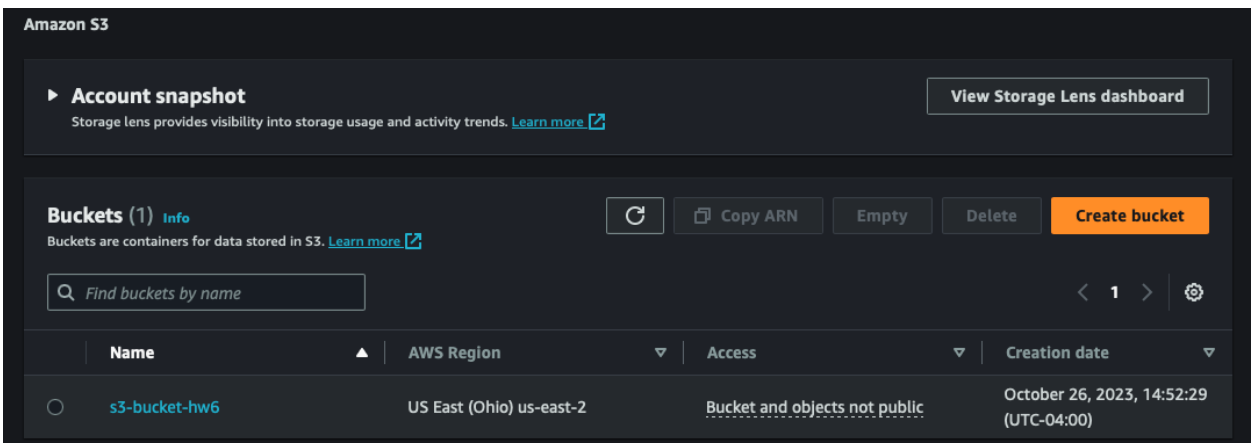
Example 3: To create a bucket outside of the ``us-east-1`` region

The following `create-bucket` example creates a bucket named `my-bucket` in the `eu-west-1` region. Regions outside of `us-east-1` require the appropriate `LocationConstraint` to be specified in order to create the bucket in the desired region.

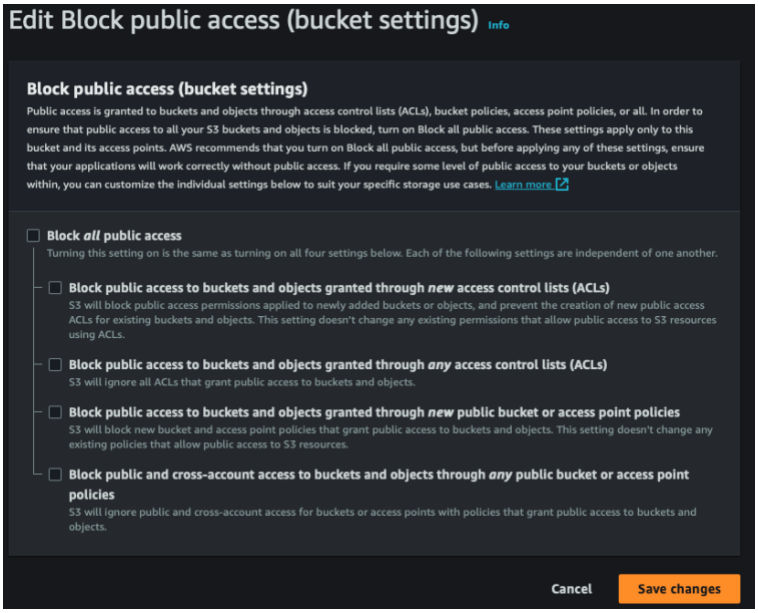
```
aws s3api create-bucket \
    --bucket my-bucket \
    --region eu-west-1 \
    --create-bucket-configuration LocationConstraint=eu-west-1
```

```
(base) deleonv@Vicentes-MacBook-Air ~ % aws s3api create-bucket --bucket s3-bucket-hw6 --region us-east-2 --create-bucket-configuration LocationConstraint=us-east-2
{
    "Location": "http://s3-bucket-hw6.s3.amazonaws.com/"
}
(base) deleonv@Vicentes-MacBook-Air ~ %
```

How to view this in dashboard? Go to services, click on Storage, S3:



I had to unblock public access due to “Access Denied”:



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
      "Condition": {
        "StringEquals": {
          "s3:ExistingObjectTag/public": "yes"
        }
      }
    }
  ]
}
```

Granting public access using JSON inline command (after trying couple times, this might be simpler):

```
type for bucket public access (JSON inline command - easier/simpler?):
aws s3api put-bucket-policy --bucket s3-bucket-hw6 --policy '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::s3-bucket-hw6/*"
    }
  ]
}'
```

I used to be blocked (ON) and I was getting “Access Denied” after running the above CLI command:

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

Edit

Block *all* public access

Off

▶ **Individual Block Public Access settings for this bucket**

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more

Edit Delete

Copy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::s3-bucket-hw6/*"
    }
  ]
}
```

References:

Creating bucket: https://docs.aws.amazon.com/cli/latest/reference/s3api/create-bucket.html

Region information: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html

Allow public read access for objects: https://repost.aws/knowledge-center/read-access-objects-s3-bucket

Allow public read access for objects: https://docs.aws.amazon.com/cli/latest/reference/s3api/put-bucket-policy.html

Bucket policy: https://docs.aws.amazon.com/AmazonS3/latest/userguide/S3OutpostsBucketPolicyEdit.html

8. Create a security group in the EC2 (Elastic Compute Cloud) service. Define inbound and outbound rules to control incoming and outgoing traffic.

```
(base) deleonv@Vicentes-MacBook-Air ~ % aws ec2 create-security-group --group-name HW6SecurityGroup --description "HW6 Security Group"
You must specify a region. You can also configure your region by running "aws configure".
(base) deleonv@Vicentes-MacBook-Air ~ % aws ec2 create-security-group --group-name HW6SecurityGroup --description "HW6 Security Group" --region us-east-2
{
    "GroupId": "sg-09e29c694e37e30b9"
}
(base) deleonv@Vicentes-MacBook-Air ~ %
```

Getting IP address (IPv4) for inbound:

```
deleonv — -zsh — 117×24
Last login: Thu Oct 26 19:38:39 on ttys001
(base) deleonv@Vicentes-MacBook-Air ~ % curl checkip.amazonaws.com
73.57.1.79
(base) deleonv@Vicentes-MacBook-Air ~ %
```

Example 1: To add a rule that allows inbound SSH traffic

The following `authorize-security-group-ingress` example adds a rule that allows inbound traffic on TCP port 22 (SSH).

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-1234567890abcdef0 \
    --protocol tcp \
    --port 22 \
    --cidr 203.0.113.0/24
```

Output:

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-01afa97ef3e1bedfc",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "203.0.113.0/24"
        }
    ]
}
```

Inbound - for SSH Traffic:

```
(base) deleonv@Vicentes-MacBook-Air ~ % aws ec2 authorize-security-group-ingress --group-name HW6SecurityGroup --protocol tcp --port 22 --cidr 73.57.1.79/32 --region us-east-2
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0a9544a42af2bfb73",
            "GroupId": "sg-09e29c694e37e30b9",
            "GroupOwnerId": "296632356656",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "73.57.1.79/32"
        }
    ]
}
```

Inbound - for HTTP Traffic:

```
(base) deleonv@Vicentes-MacBook-Air ~ % aws ec2 authorize-security-group-ingress --group-name HW6SecurityGroup --protocol tcp --port 80 --cidr 0.0.0.0/0 --region us-east-2
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0996aa910e722176d",
            "GroupId": "sg-09e29c694e37e30b9",
            "GroupOwnerId": "296632356656",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "CidrIpv4": "0.0.0.0/0"
        }
    ]
}
```

Security Groups (2) Info

Actions | Export security groups to CSV | Create security group

Filter security groups

Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count	Outbound rules co...
–	sg-0dd07b8e91b213022	default	vpc-00cc3bf492505007e	default VPC security gr...	296632356656	1 Permission entry	1 Permission entry
–	sg-09e29c694e37e30b9	HW6SecurityGroup	vpc-00cc3bf492505007e	HW6 Security Group	296632356656	2 Permission entries	1 Permission entry

Group ID came from EC2 dashboard, Security Groups, sg-09e29c694e37e30b9 (HW6SecurityGroup)

Outbound – for HTTP permission to anywhere regular:

```
(base) deleonv@Vicentes-MacBook-Air ~ % aws ec2 authorize-security-group-egress --group-id sg-09e29c694e37e30b9 --protocol tcp --port 80 --cidr 0.0.0.0/0 --region us-east-2
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0b96c60ac1c2e938b",
            "GroupId": "sg-09e29c694e37e30b9",
            "GroupOwnerId": "296632356656",
            "IsEgress": true,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "CidrIpv4": "0.0.0.0/0"
        }
    ]
}
```

Outbound – for HTTPS permission to anywhere secure:

```
(base) deleonv@Vicentes-MacBook-Air ~ % aws ec2 authorize-security-group-egress --group-id sg-09e29c694e37e30b9 --protocol tcp --port 443 --cidr 0.0.0.0/0 --region us-east-2
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0cfb5b2852af92f3f",
            "GroupId": "sg-09e29c694e37e30b9",
            "GroupOwnerId": "296632356656",
            "IsEgress": true,
            "IpProtocol": "tcp",
            "FromPort": 443,
            "ToPort": 443,
            "CidrIpv4": "0.0.0.0/0"
        }
    ]
}
```

Observations:

For the inbound:

- SSH – allowing SSH traffic only from IPv4 address.
- HTTP – allowing HTTP traffic from anywhere, which is something common for web servers.

For the outbound:

- HTTP and HTTPS traffic can go to any destination.

References:

Security group: https://docs.aws.amazon.com/cli/latest/reference/ec2/create-security-group.html

Security group: https://docs.aws.amazon.com/cli/latest/userguide/cli-services-ec2-sg.html

Security group ingress: https://docs.aws.amazon.com/cli/latest/reference/ec2/authorize-security-group-ingress.html

getting IP address: https://www.turais.de/get-your-public-ip-from-commandline/

EC2 connection: https://repost.aws/knowledge-center/connect-http-https-ec2

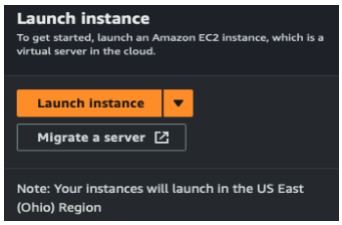
HTTP: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/security-group-rules-reference.html

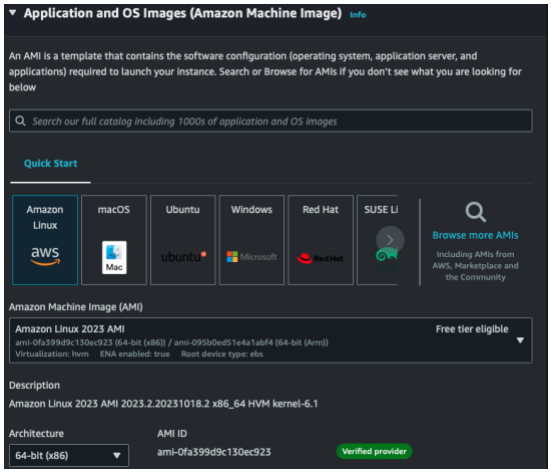
Ipv4 address: https://docs.aws.amazon.com/vpc/latest/userguide/security-group-rules.html

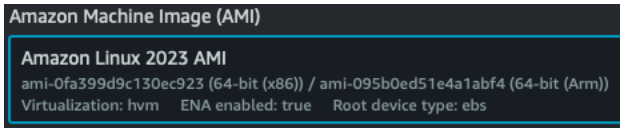
Outbound: https://awscli.amazonaws.com/v2/documentation/api/2.3.2/reference/ec2/authorize-security-group-egress.html

9. Launch a new EC2 instance using Amazon Linux 2 as the base AMI (Amazon Machine Image). Assign the security group created in step 8 to the instance.

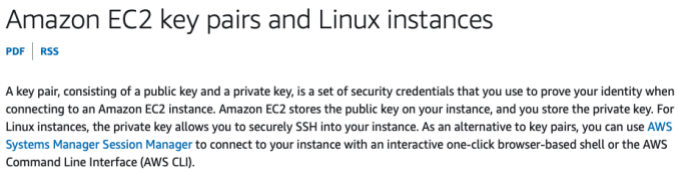
EC2 Dashboard:

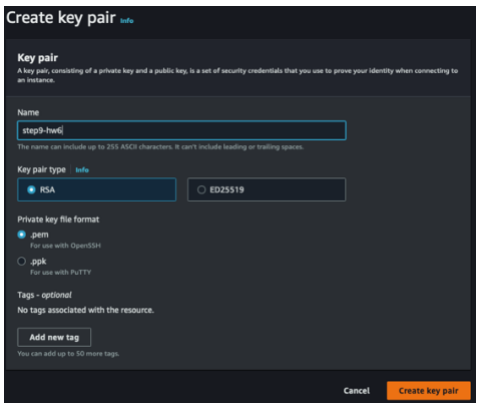






I seems I must create a key pair to launch an EC2 instance:





Using CLI to create an EC2 instance:

To create the EC2 instance in the AWS CLI with the minimum recommended set of parameters, use the following commands:

```
aws ec2 run-instances \
--image-id <ami-id> \
--instance-type <instance-type> \
--subnet-id <subnet-id> \
--security-group-ids <security-group-id> <security-group-id> … \
--key-name <ec2-key-pair-name>
```

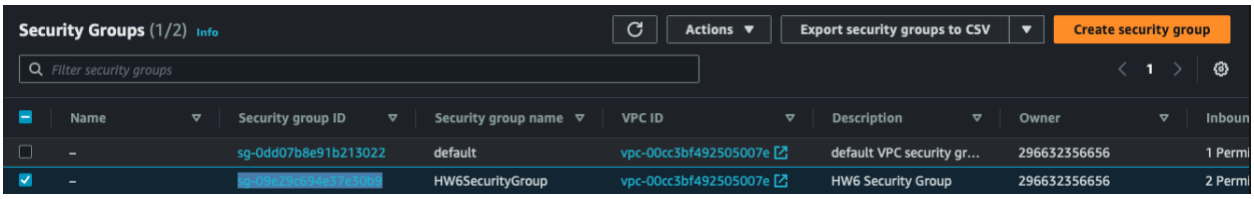
CLI Command:

```
(base) deleonv@Vicentes-MacBook-Air ~ % aws ec2 run-instances --image-id ami-0fa399d9c130ec923 --region us-east-2 --instance-type t2.micro --security-group-ids sg-09e29c694e37e30b9 --key-name step9-hw6
{
    "Groups": [],
    "Instances": [
        {
            "AmiLaunchIndex": 0,
            "ImageId": "ami-0fa399d9c130ec923",
            "InstanceId": "i-03d4345d964e0dad1",
            "InstanceType": "t2.micro",
            "KeyName": "step9-hw6",
            "LaunchTime": "2023-10-27T21:27:02.000Z",
            "Monitoring": {
                "State": "disabled"
            },
            "Placement": {
                "AvailabilityZone": "us-east-2a",
                "GroupName": "",
                "Tenancy": "default"
            },
            "PrivateDnsName": "ip-172-31-5-209.us-east-2.compute.internal",
            "PrivateIpAddress": "172.31.5.209",
            "ProductCodes": [],
            "PublicDnsName": "",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "StateTransitionReason": "",
            "SubnetId": "subnet-0857c6e5bafceb4e2",
            "VpcId": "vpc-00cc3bf492505007e",
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "b1615338-4976-4bec-ac55-dc88bd1499a3",
            "EbsOptimized": false,
            "EnaSupport": true,
            "Hypervisor": "xen",
            "NetworkInterfaces": [
                {
                    "Attachment": {
                        "AttachTime": "2023-10-27T21:27:02.000Z",
                        "AttachmentId": "eni-attach-0d87d2af5a7693054",
                        "DeleteOnTermination": true,
                        "DeviceIndex": 0,
                        "Status": "attaching",
                        "NetworkCardIndex": 0
                    },
                    "Description": "",
                    "Groups": [
                        {
                            "GroupName": "HW6SecurityGroup",
                            "GroupId": "sg-09e29c694e37e30b9"
                        }
                    ],
                    "Ipv6Addresses": [],
                    "MacAddress": "02:ae:c1:4c:7d:15",
                    "NetworkInterfaceId": "eni-08a7b5f775dee72b0",
                    "OwnerId": "296632356656",
                    "PrivateDnsName": "ip-172-31-5-209.us-east-2.compute.internal",
                    "PrivateIpAddress": "172.31.5.209",
                    "PrivateIpAddresses": [
                        {
                            "Primary": true,
                            "PrivateDnsName": "ip-172-31-5-209.us-east-2.compute.internal",
                            "PrivateIpAddress": "172.31.5.209"
                        }
                    ],
                    "SourceDestCheck": true,
                    "Status": "in-use",
                    "SubnetId": "subnet-0857c6e5bafceb4e2",
                    "VpcId": "vpc-00cc3bf492505007e",
                    "InterfaceType": "interface"
                }
            ],
            "RootDeviceName": "/dev/xvda",
            "RootDeviceType": "ebs",
            "SecurityGroups": [
```

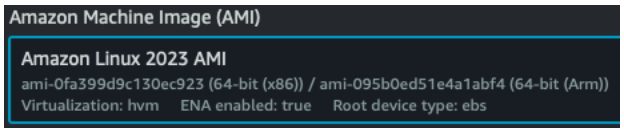
```
Step 9:
# image-id ami-0fa399d9c130ec923 (came from EC2 Dashboard -> Launch Instances)
# region -> Ohio -> us-east-2 ( I have no info about subnet_id)
# instance type -> Free tier instance (t2.micro)
# security group id -> HW6SecurityGroup ID I just created EC2 dahsboard (Security Groups)
# key name -> Key Pair name

type: aws ec2 run-instances --image-id ami-0fa399d9c130ec923 --region us-east-2 --instance-type t2.micro --security-group-ids sg-09e29c694e37e30b9 --key-name step9-hw6
```

Reviewing CLI command:

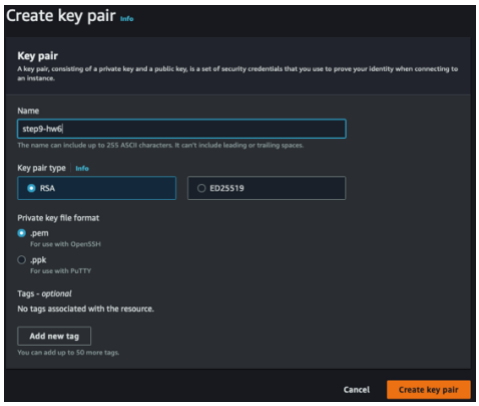
HW6SecurityGroup ID:



AMI ID:



Key Name:



References:

Linux AMI: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/finding-an-ami.html

Ami: https://docs.aws.amazon.com/cli/latest/reference/ec2/describe-images.html

Amazon EC2 instances and Key pairs: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html

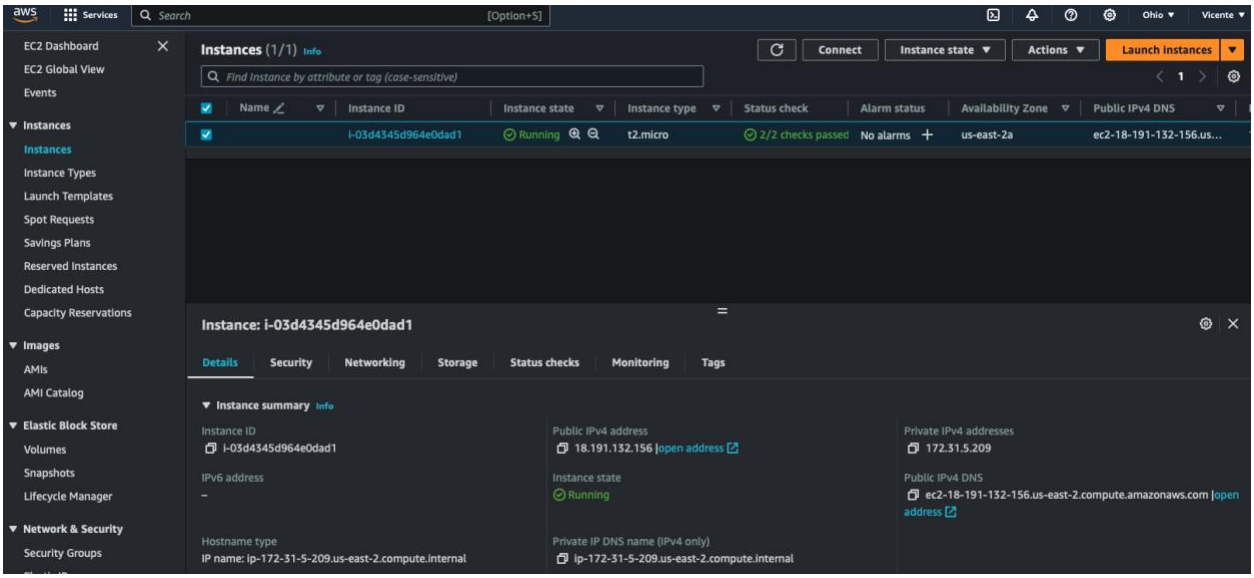
AWS CLI EC2 instances: https://www.techtarget.com/searchcloudcomputing/tutorial/Use-the-AWS-CLI-to-create-an-EC2-instance

Free tier instance: https://aws.amazon.com/about-aws/whats-new/2017/01/amazon-elasticsearch-service-free-tier-now-available-on-t2-small-elasticsearch-instances/

Free tier instance: https://aws.amazon.com/free/compute/

10. Connect to the EC2 instance using SSH and perform basic system configurations.

I need my public IP address of EC2 instance: 18.191.132.156 (located in EC2 Dashboard):




```
(base) deleonv@Vicentes-MacBook-Air ~ % cd ~/Desktop
(base) deleonv@Vicentes-MacBook-Air Desktop % chmod 400 step9-hw6.pem
(base) deleonv@Vicentes-MacBook-Air Desktop % ssh -i step9-hw6.pem ec2-user@18.191.132.156

The authenticity of host '18.191.132.156 (18.191.132.156)' can't be established.
ED25519 key fingerprint is SHA256:PGt3KYl/5mDDMfJ3yR3xPkc8B7uo52oYZnr1ZCARKZ0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '18.191.132.156' (ED25519) to the list of known hosts.
       #_
       ~\_  ####_        Amazon Linux 2023
      ~~  \_#####\
      ~~     \###|
      ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
       ~~       V~' '->
        ~~~         /
          ~~._.   _/
             _/ _/
           _/m/'
[ec2-user@ip-172-31-5-209 ~]$
```

```
[ec2-user@ip-172-31-5-209 ~]$ sudo dnf update -y
Last metadata expiration check: 1:03:07 ago on Fri Oct 27 21:27:56 2023.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-5-209 ~]$
```

```
[ec2-user@ip-172-31-5-209 ~]$ sudo dnf install -y httpd php php-mysqli mariadb105
Last metadata expiration check: 1:08:33 ago on Fri Oct 27 21:27:56 2023.
Dependencies resolved.
================================================================================================================================
 Package                                  Architecture            Version                                  Repository          Size
================================================================================================================================
Installing:
 httpd                                    x86_64                  2.4.56-1.amzn2023                        amazonlinux          48 k
 mariadb105                               x86_64                  3:10.5.20-1.amzn2023.0.1                 amazonlinux         1.6 M
 php8.2                                   x86_64                  8.2.9-1.amzn2023.0.3                     amazonlinux          13 k
 php8.2-mysqlnd                           x86_64                  8.2.9-1.amzn2023.0.3                     amazonlinux         150 k
Installing dependencies:
 apr                                      x86_64                  1.7.2-2.amzn2023.0.2                     amazonlinux         129 k
 apr-util                                 x86_64                  1.6.3-1.amzn2023.0.1                     amazonlinux          98 k
 generic-logos-httpd                      noarch                  18.0.0-12.amzn2023.0.3                   amazonlinux          19 k
 httpd-core                               x86_64                  2.4.56-1.amzn2023                        amazonlinux         1.4 M
 httpd-filesystem                         noarch                  2.4.56-1.amzn2023                        amazonlinux          15 k
 httpd-tools                              x86_64                  2.4.56-1.amzn2023                        amazonlinux          82 k
 libbrotli                                x86_64                  1.0.9-4.amzn2023.0.2                     amazonlinux         315 k
 libsodium                                x86_64                  1.0.18-13.amzn2023.0.1                   amazonlinux         166 k
 libxslt                                  x86_64                  1.1.34-5.amzn2023.0.2                    amazonlinux         241 k
```

```
Installed:
  apr-1.7.2-2.amzn2023.0.2.x86_64                       apr-util-1.6.3-1.amzn2023.0.1.x86_64                  apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
  generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch     httpd-2.4.56-1.amzn2023.x86_64                        httpd-core-2.4.56-1.amzn2023.x86_64
  httpd-filesystem-2.4.56-1.amzn2023.noarch             httpd-tools-2.4.56-1.amzn2023.x86_64                  libbrotli-1.0.9-4.amzn2023.0.2.x86_64
  libsodium-1.0.18-13.amzn2023.0.1.x86_64               libxslt-1.1.34-5.amzn2023.0.2.x86_64                  mailcap-2.1.49-3.amzn2023.0.3.noarch
  mariadb-connector-c-3.1.13-1.amzn2023.0.3.x86_64      mariadb-connector-c-config-3.1.13-1.amzn2023.0.3.noarch  mariadb105-3:10.5.20-1.amzn2023.0.1.x86_64
  mariadb105-common-3:10.5.20-1.amzn2023.0.1.x86_64     mod_http2-2.0.11-2.amzn2023.x86_64                    mod_lua-2.4.56-1.amzn2023.x86_64
  nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch       perl-Sys-Hostname-1.23-477.amzn2023.0.5.x86_64        php8.2-8.2.9-1.amzn2023.0.3.x86_64
  php8.2-cli-8.2.9-1.amzn2023.0.3.x86_64                php8.2-common-8.2.9-1.amzn2023.0.3.x86_64             php8.2-fpm-8.2.9-1.amzn2023.0.3.x86_64
  php8.2-mbstring-8.2.9-1.amzn2023.0.3.x86_64           php8.2-mysqlnd-8.2.9-1.amzn2023.0.3.x86_64            php8.2-opcache-8.2.9-1.amzn2023.0.3.x86_64
  php8.2-pdo-8.2.9-1.amzn2023.0.3.x86_64                php8.2-process-8.2.9-1.amzn2023.0.3.x86_64            php8.2-sodium-8.2.9-1.amzn2023.0.3.x86_64
  php8.2-xml-8.2.9-1.amzn2023.0.3.x86_64

Complete!
```

So, for the above installation I have the following: Linux, Apache, MySQL and PHP (Apache Web Server, PHP, MySQL, MariaDB). I will use Apache Web Server to test step 11.

Pem connection: https://repost.aws/questions/QUsZqMJGVtQmemkEMVwxUiIw/why-should-i-change-the-permissions-on-the-ssh-pem-file

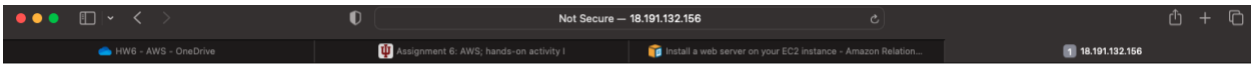
Chmod 400 pem connection: https://99robots.com/how-to-fix-permission-error-ssh-amazon-ec2-instance/

Connect using SSH: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/connect-linux-inst-ssh.html

Updating EC2 software and installing packages: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateWebServer.html

11. Test EC2 connectivity be accessing web server running form my local machine.

```
Complete!
[[ec2-user@ip-172-31-5-209 ~]$ sudo systemctl start httpd
[[ec2-user@ip-172-31-5-209 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-5-209 ~]$
```



It works!

It looks like it successfully worked!

Updating EC2 software and installing packages (including starting Web Server): https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateWebServer.html