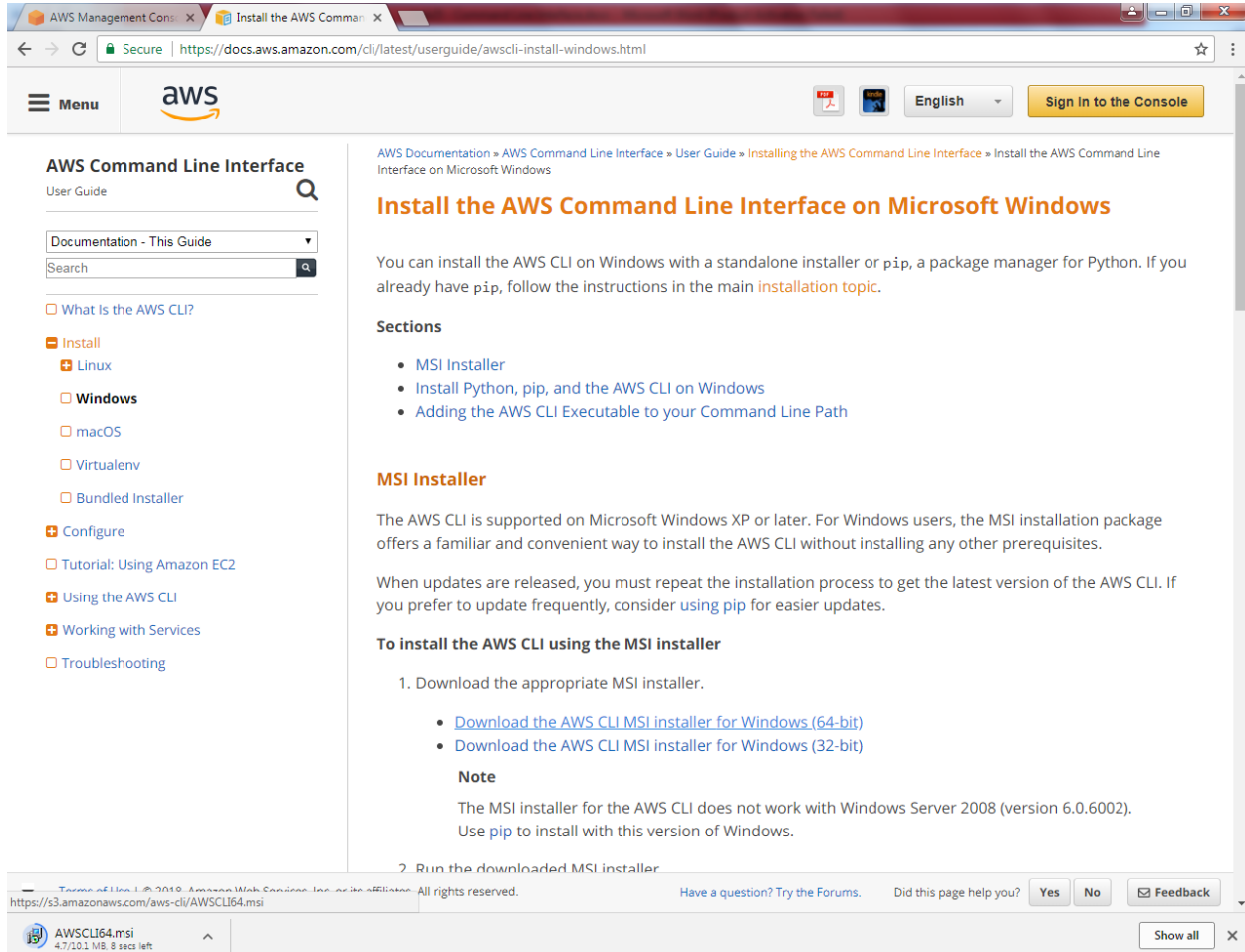


Lab25

AWS - Command Line Interface


Use the below URL to download CLI for Windows

<https://docs.aws.amazon.com/cli/latest/userguide/awscli-install-windows.html>



AWS Management Console x Install the AWS Command Line Interface x

Secure | <https://docs.aws.amazon.com/cli/latest/userguide/awscli-install-windows.html>

Menu  English Sign In to the Console

AWS Command Line Interface

User Guide

Documentation - This Guide

Search

- What is the AWS CLI?
- Install
 - Linux
 - Windows**
 - macOS
 - Virtualenv
 - Bundled Installer
- Configure
- Tutorial: Using Amazon EC2
- Using the AWS CLI
- Working with Services
- Troubleshooting

AWS Documentation » AWS Command Line Interface » User Guide » Installing the AWS Command Line Interface » Install the AWS Command Line Interface on Microsoft Windows

Install the AWS Command Line Interface on Microsoft Windows

You can install the AWS CLI on Windows with a standalone installer or `pip`, a package manager for Python. If you already have `pip`, follow the instructions in the main [installation topic](#).

Sections

- MSI Installer
- Install Python, `pip`, and the AWS CLI on Windows
- Adding the AWS CLI Executable to your Command Line Path

MSI Installer

The AWS CLI is supported on Microsoft Windows XP or later. For Windows users, the MSI installation package offers a familiar and convenient way to install the AWS CLI without installing any other prerequisites.

When updates are released, you must repeat the installation process to get the latest version of the AWS CLI. If you prefer to update frequently, consider [using `pip`](#) for easier updates.

To install the AWS CLI using the MSI installer


- Download the appropriate MSI installer.
 - [Download the AWS CLI MSI installer for Windows \(64-bit\)](#)
 - [Download the AWS CLI MSI installer for Windows \(32-bit\)](#)

Note

The MSI installer for the AWS CLI does not work with Windows Server 2008 (version 6.0.6002). Use `pip` to install with this version of Windows.

- Run the downloaded MSI installer

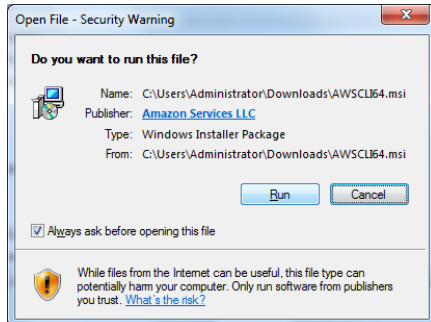
Terms of Use | © 2018 Amazon Web Services, Inc. or its affiliates. All rights reserved. [Have a question? Try the Forums.](#) Did this page help you?

 AWSCLI64.msi 4.7/10.1 MB, 8 secs left

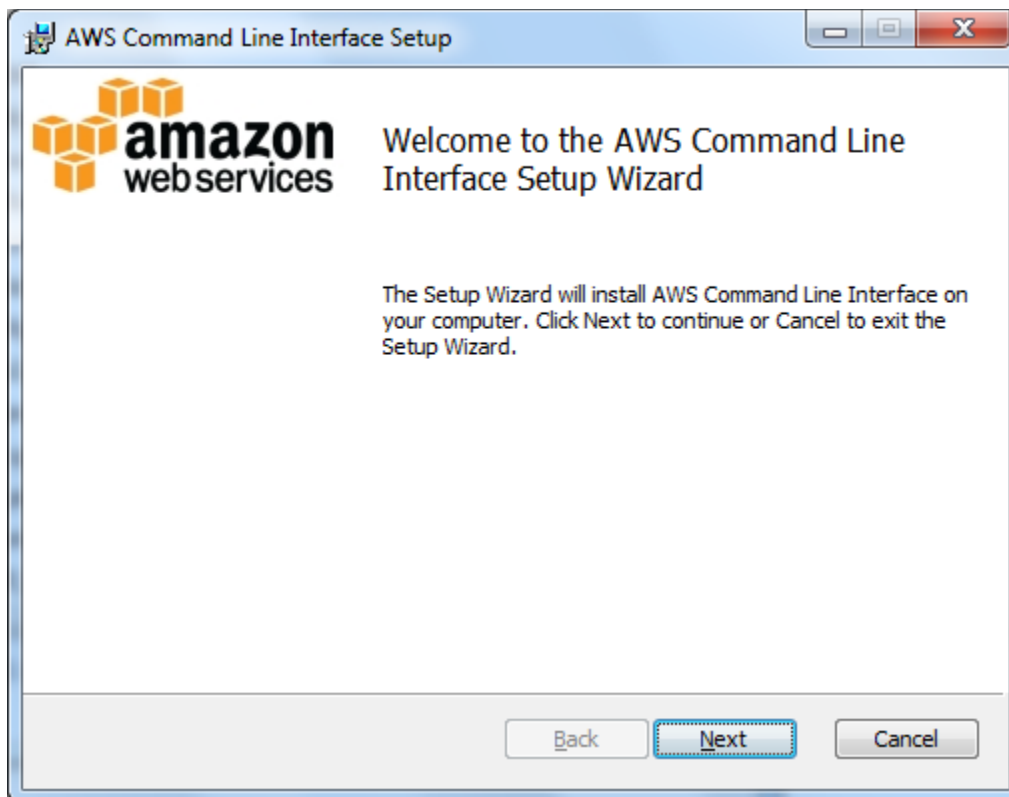
File is getting download.

We need to install in our local machine.

Run AWSCLI64.msi file.



Click "Run".

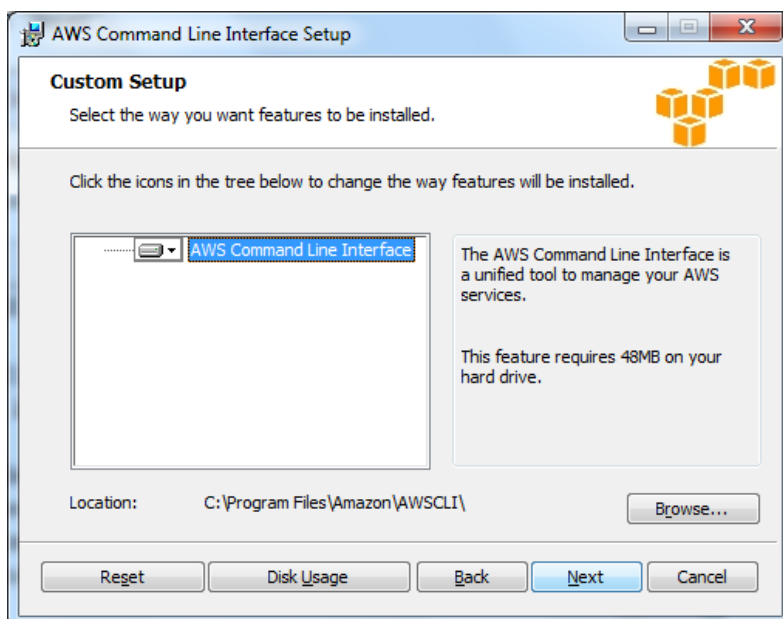


Click "Next".

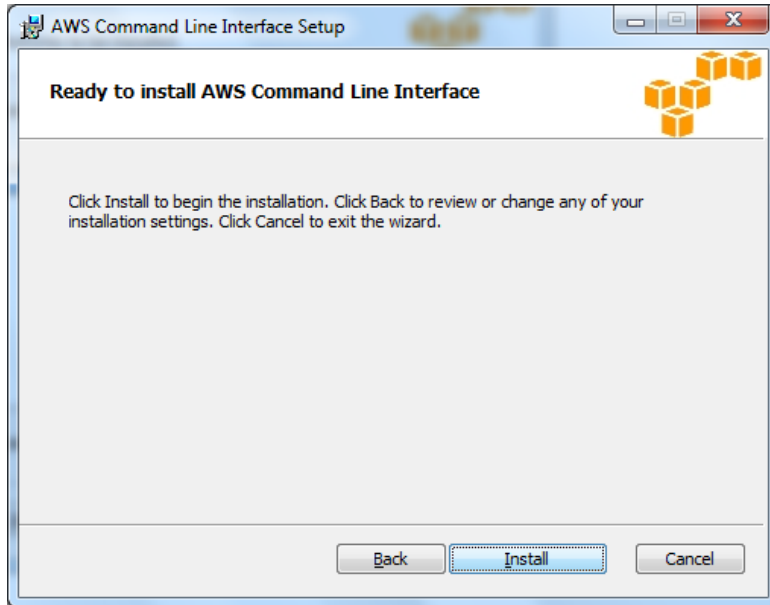
Click I accept and click “Next”.



Click “Next”.

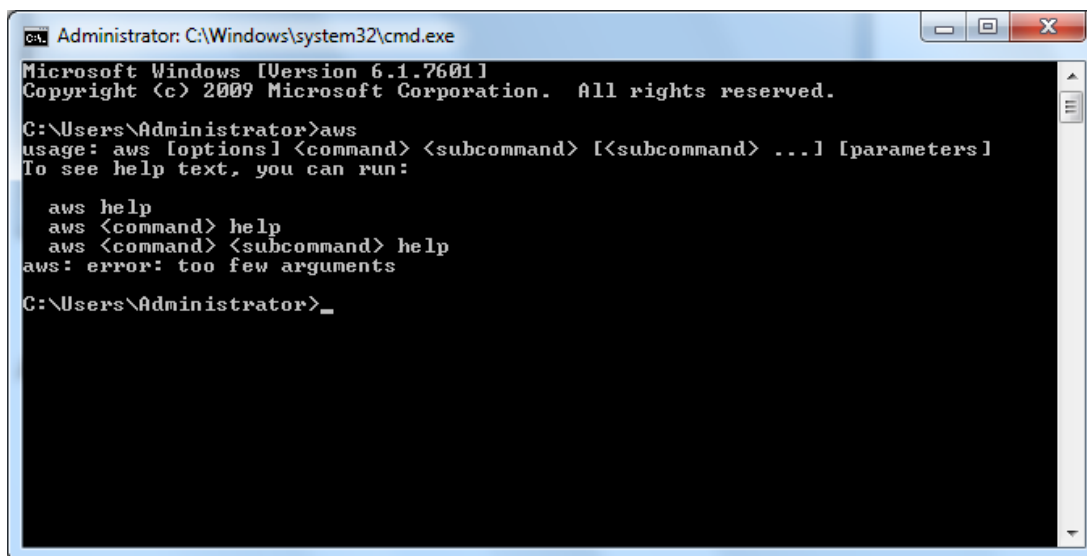


Click "Install".



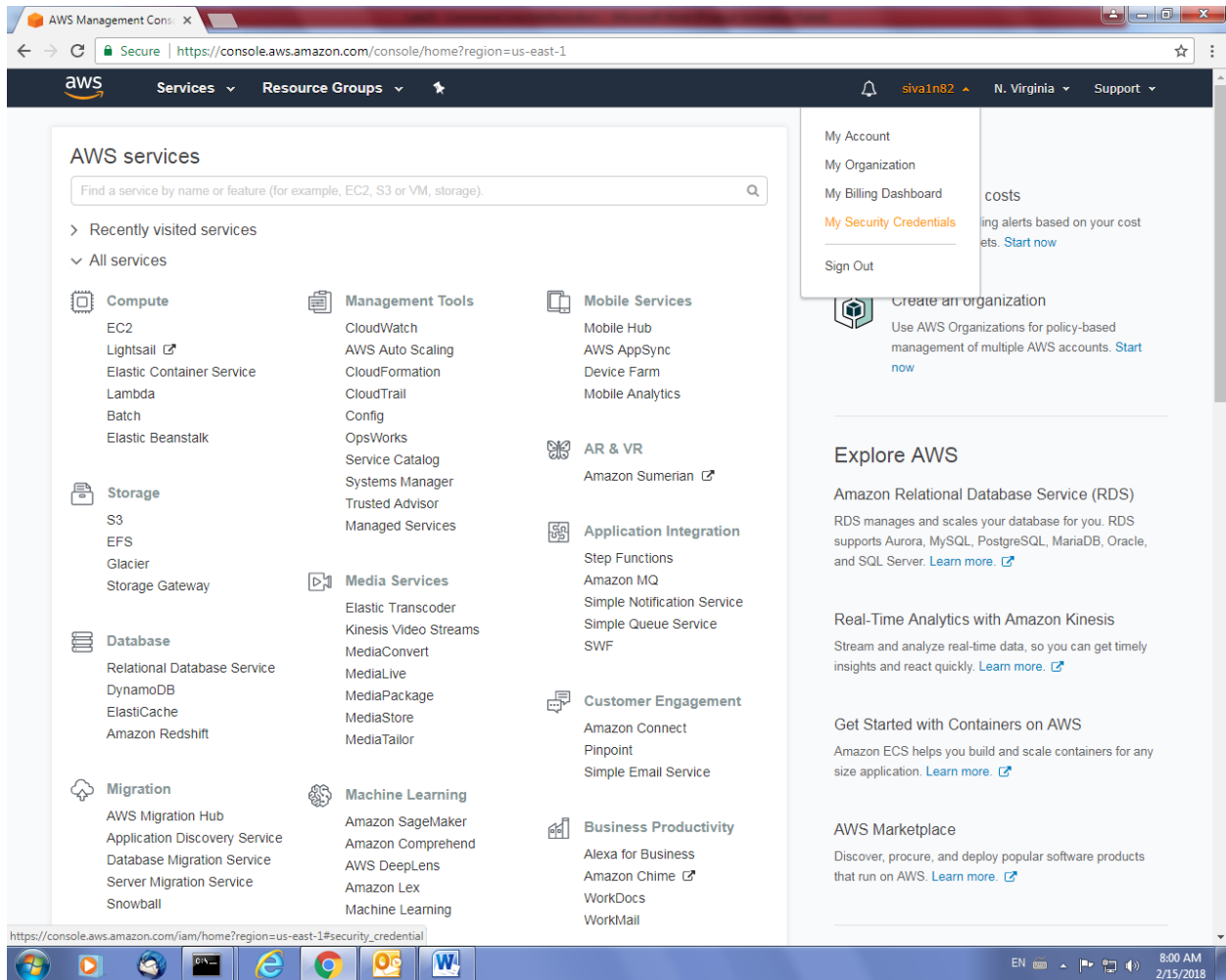
Application installation will be successfully completed.

Type aws and then press enter. You can able to see the commands in command prompt.

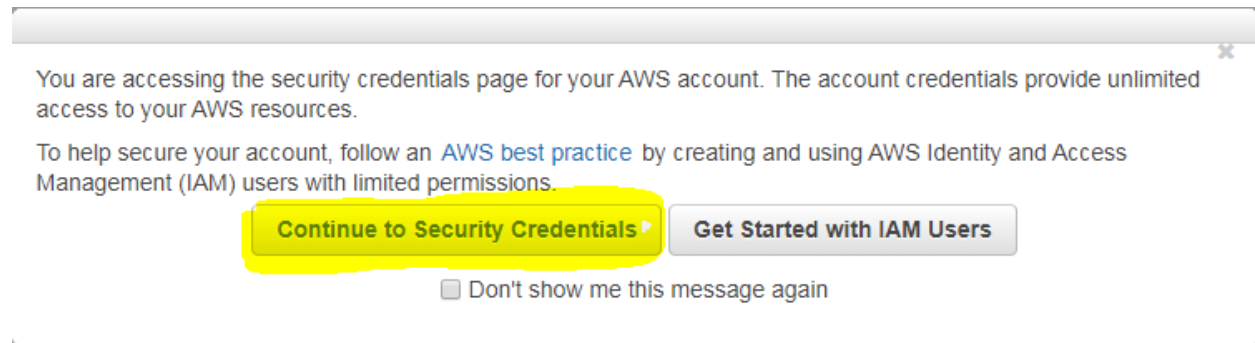


Before Login to CLI, we have required root keys for my account to login to CLI interface.

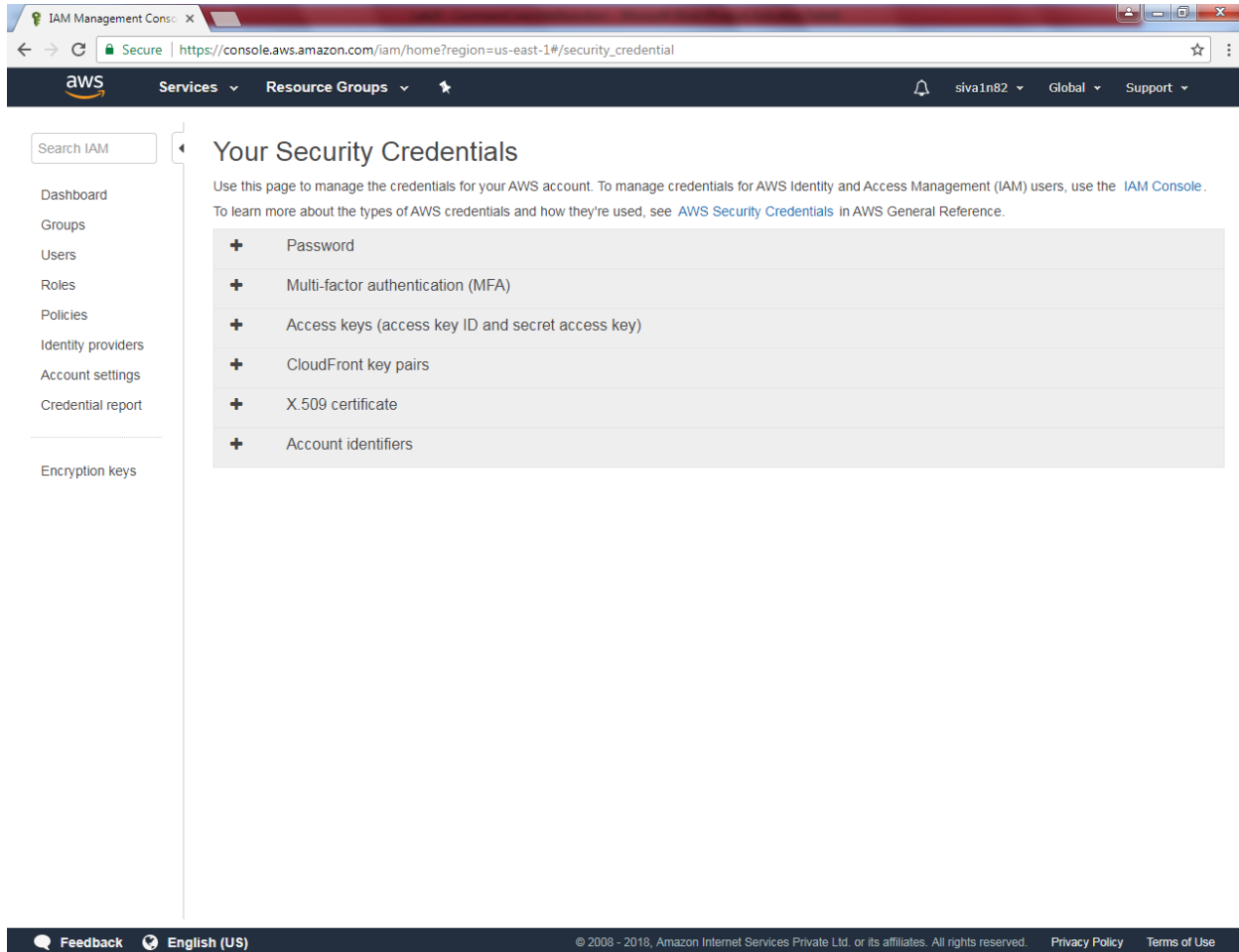
Click “My Security Credentials”.



Click “continue to security credentials”.



Press “+” key in **Access keys** to expand it.

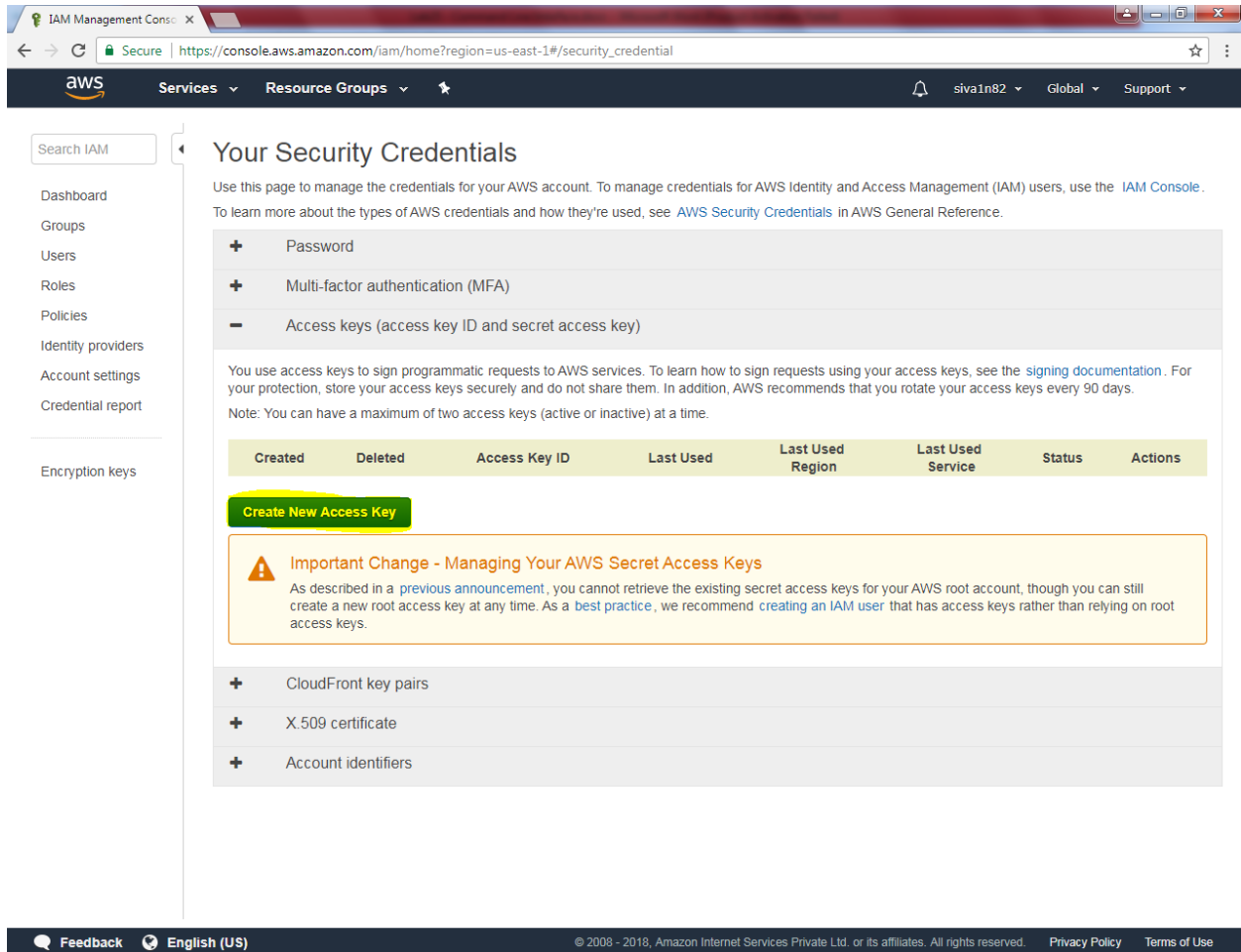


The screenshot shows the AWS IAM console interface. The browser address bar displays the URL `https://console.aws.amazon.com/iam/home?region=us-east-1#/security_credential`. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information 'siva1n82'. A left-hand sidebar contains a search bar labeled 'Search IAM' and a list of navigation links: Dashboard, Groups, Users, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The main content area is titled 'Your Security Credentials' and includes a brief instruction: '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.' Below this instruction is a table listing six types of credentials, each with a plus icon to its left:

+	Password
+	Multi-factor authentication (MFA)
+	Access keys (access key ID and secret access key)
+	CloudFront key pairs
+	X.509 certificate
+	Account identifiers

The footer of the console contains a 'Feedback' link, a language selector set to 'English (US)', and copyright information: '© 2008 - 2018, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.' It also includes links for 'Privacy Policy' and 'Terms of Use'.

Click “Create New Access Key”.



Search IAM

- Dashboard
- Groups
- Users
- Roles
- Policies
- Identity providers
- Account settings
- Credential report
- Encryption keys


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)

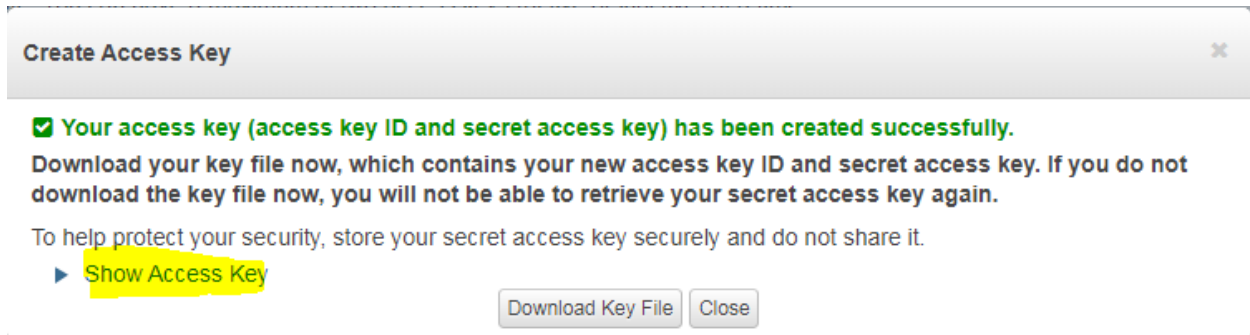
You use access keys to sign programmatic requests to AWS services. To learn how to sign requests using your access keys, see the [signing documentation](#). For your protection, store your access keys securely and do not share them. In addition, AWS recommends that you rotate your access keys every 90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

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

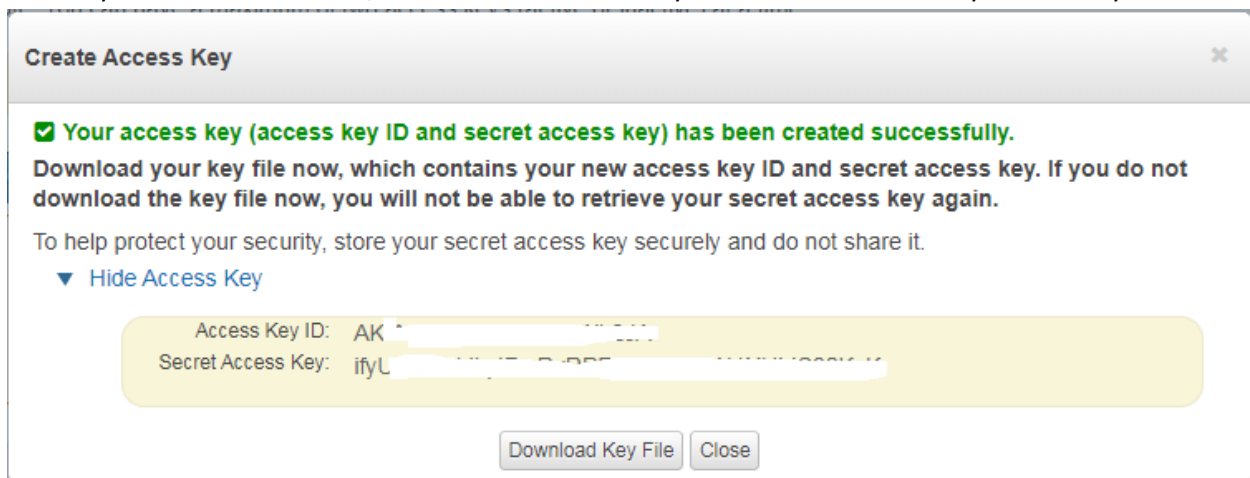
Created	Deleted	Access Key ID	Last Used	Last Used Region	Last Used Service	Status	Actions
Create New Access Key							
<div> Important Change - Managing Your AWS Secret Access Keys As described in a previous announcement, you cannot retrieve the existing secret access keys for your AWS root account, 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.</div>							
+ CloudFront key pairs							
+ X.509 certificate							
+ Account identifiers							

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

Click “Show access key” then copy the key into notepad. Because you would not be able to get the password key after this mode / you skip copy from this mode.



Root keys will be like as below, I have masked Access key ID and Secret access key for security reasons.

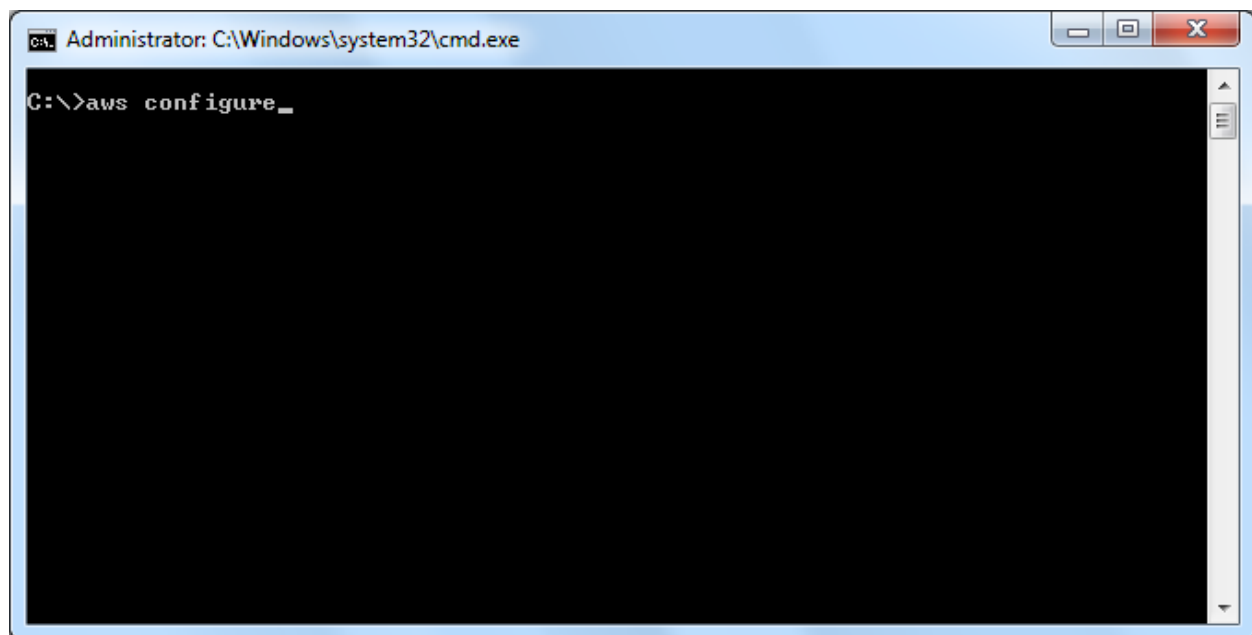


Click “Download Key File” and click “close”.

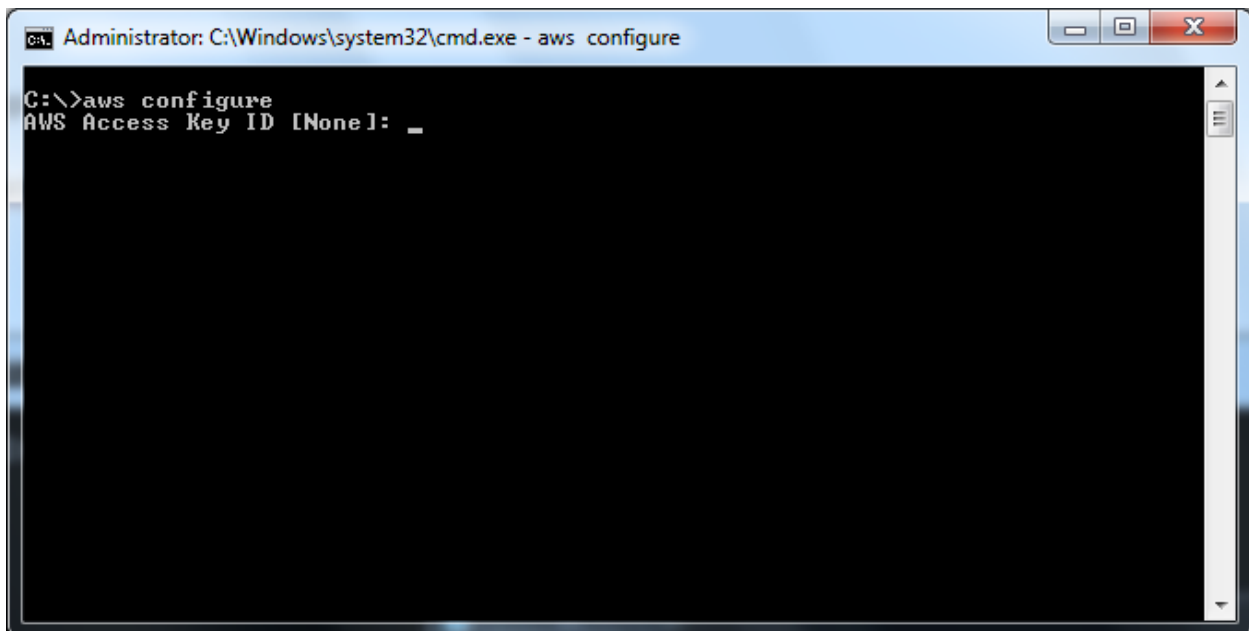
Now we need to login to command prompt by using Root keys.

Type

aws configure



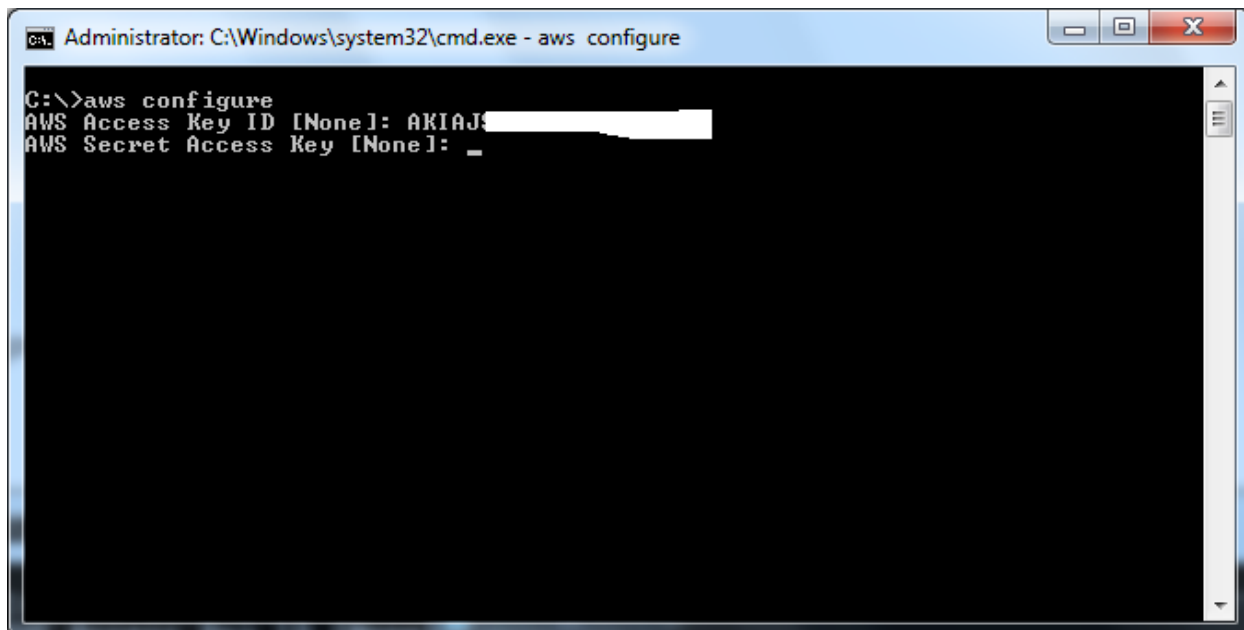
It prompts user id,



```
Administrator: C:\Windows\system32\cmd.exe - aws configure

C:\>aws configure
AWS Access Key ID [None]: _
```

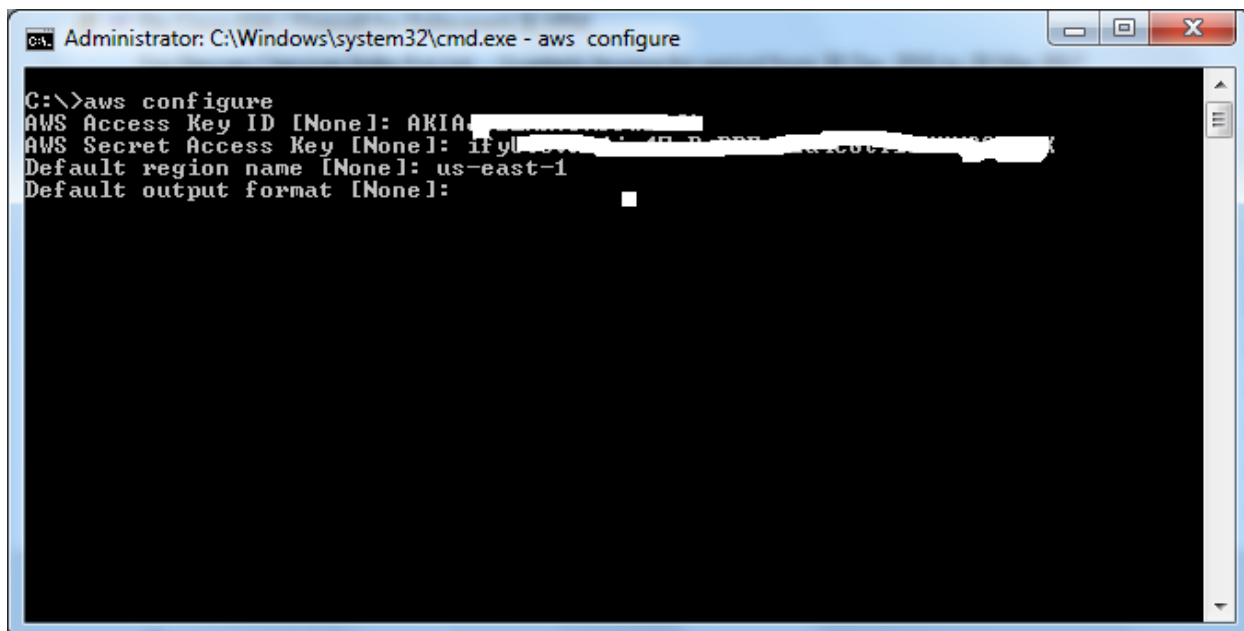
It prompts password, type secret access key



```
Administrator: C:\Windows\system32\cmd.exe - aws configure

C:\>aws configure
AWS Access Key ID [None]: AKIAJ[REDACTED]
AWS Secret Access Key [None]: [REDACTED]
```

Type region name as us-east-1/where you have connected and type output format json

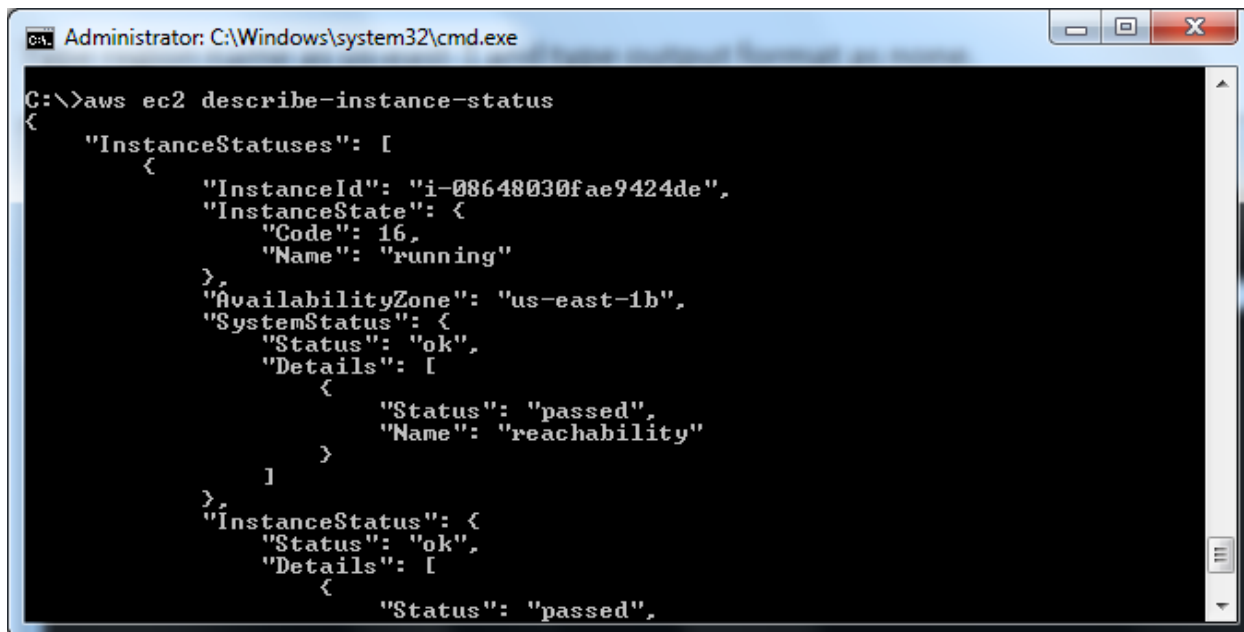


```
Administrator: C:\Windows\system32\cmd.exe - aws configure

C:\>aws configure
AWS Access Key ID [None]: AKIAJ[REDACTED]
AWS Secret Access Key [None]: ify[REDACTED]
Default region name [None]: us-east-1
Default output format [None]: [REDACTED]
```

Type

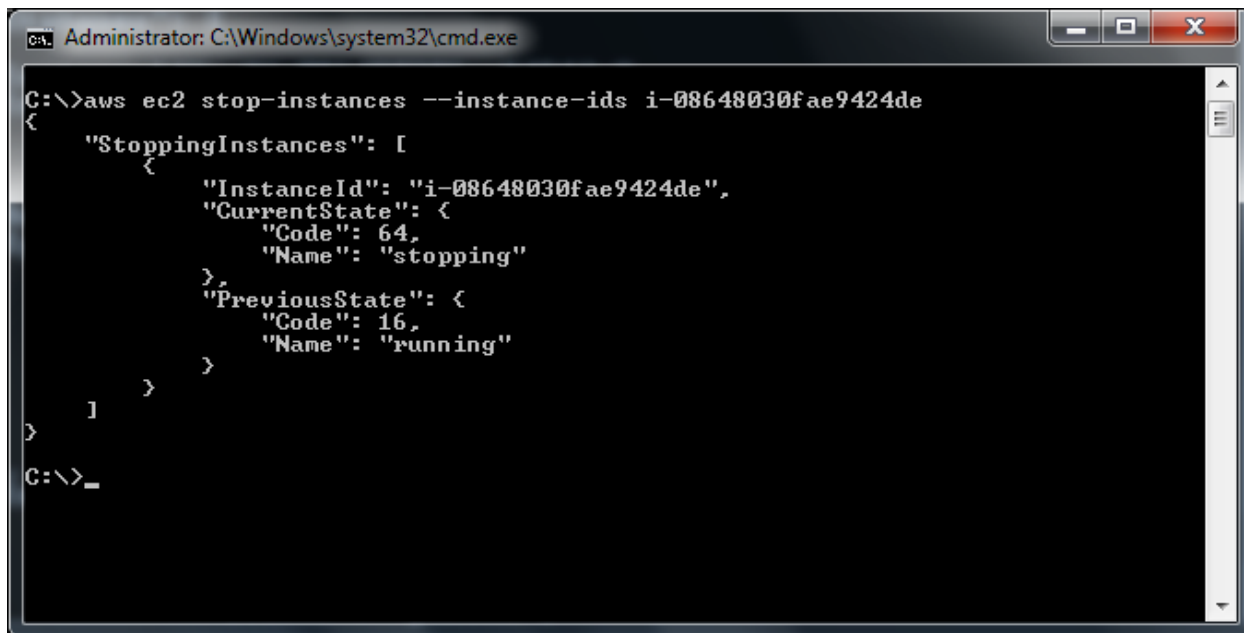
Aws ec2 describe-instance-status



```
Administrator: C:\Windows\system32\cmd.exe
C:\>aws ec2 describe-instance-status
{
  "InstanceStatuses": [
    {
      "InstanceId": "i-08648030fae9424de",
      "InstanceState": {
        "Code": 16,
        "Name": "running"
      },
      "AvailabilityZone": "us-east-1b",
      "SystemStatus": {
        "Status": "ok",
        "Details": [
          {
            "Status": "passed",
            "Name": "reachability"
          }
        ]
      }
    }
  ],
  "InstanceState": {
    "Status": "ok",
    "Details": [
      {
        "Status": "passed",
```

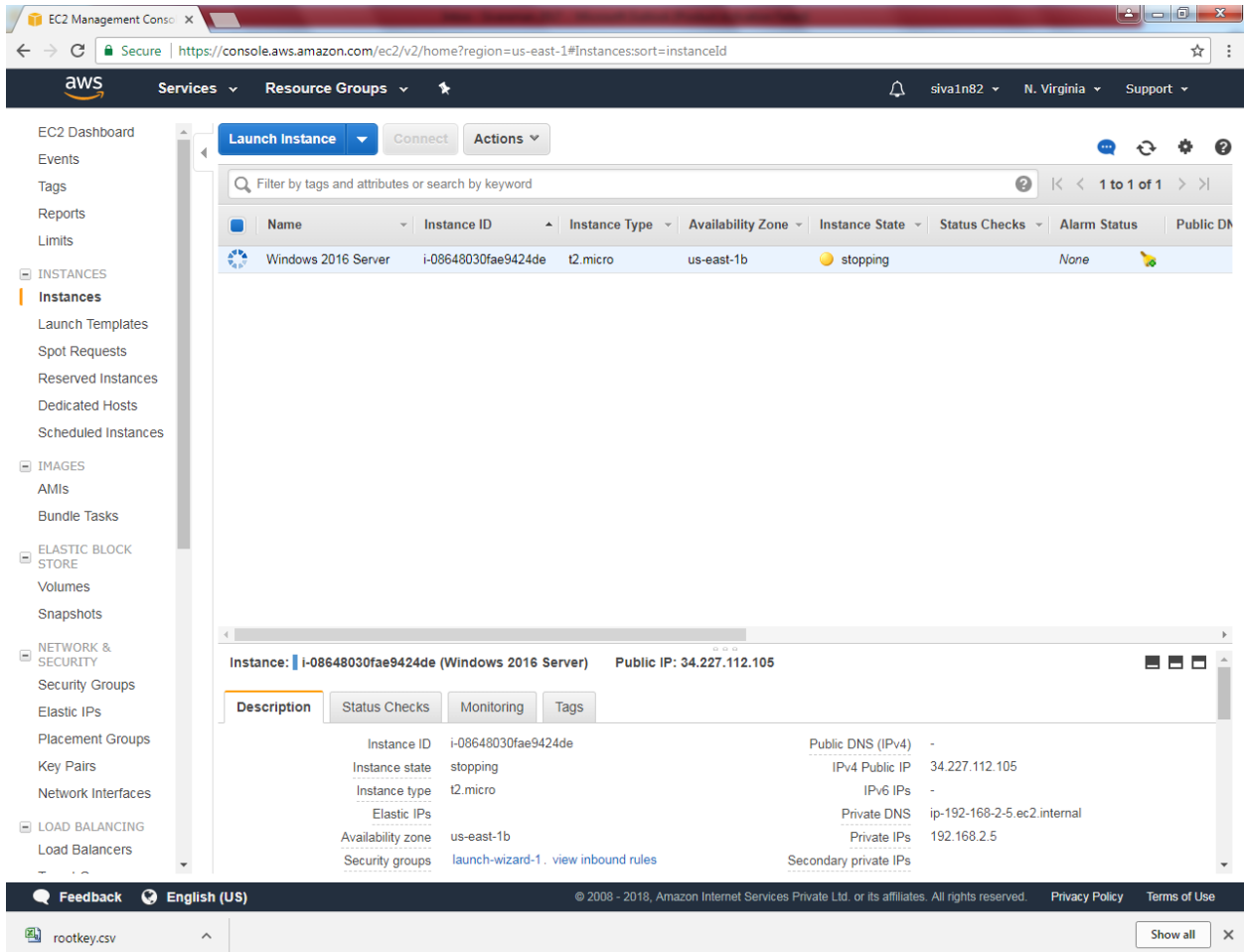
Type

Aws ec2 stop-instances --instance-ids <instance id>



```
Administrator: C:\Windows\system32\cmd.exe
C:\>aws ec2 stop-instances --instance-ids i-08648030fae9424de
{
  "StoppingInstances": [
    {
      "InstanceId": "i-08648030fae9424de",
      "CurrentState": {
        "Code": 64,
        "Name": "stopping"
      },
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
  ]
}
```

In output you can be able to see that instance is getting stop.



The screenshot displays the AWS Management Console interface for the EC2 service. The left-hand navigation pane lists various AWS services and resources, including EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES, IMAGES, ELASTIC BLOCK STORE, NETWORK & SECURITY, and LOAD BALANCING. The main content area shows a list of EC2 instances. A single instance, 'Windows 2016 Server', is listed with the following details:

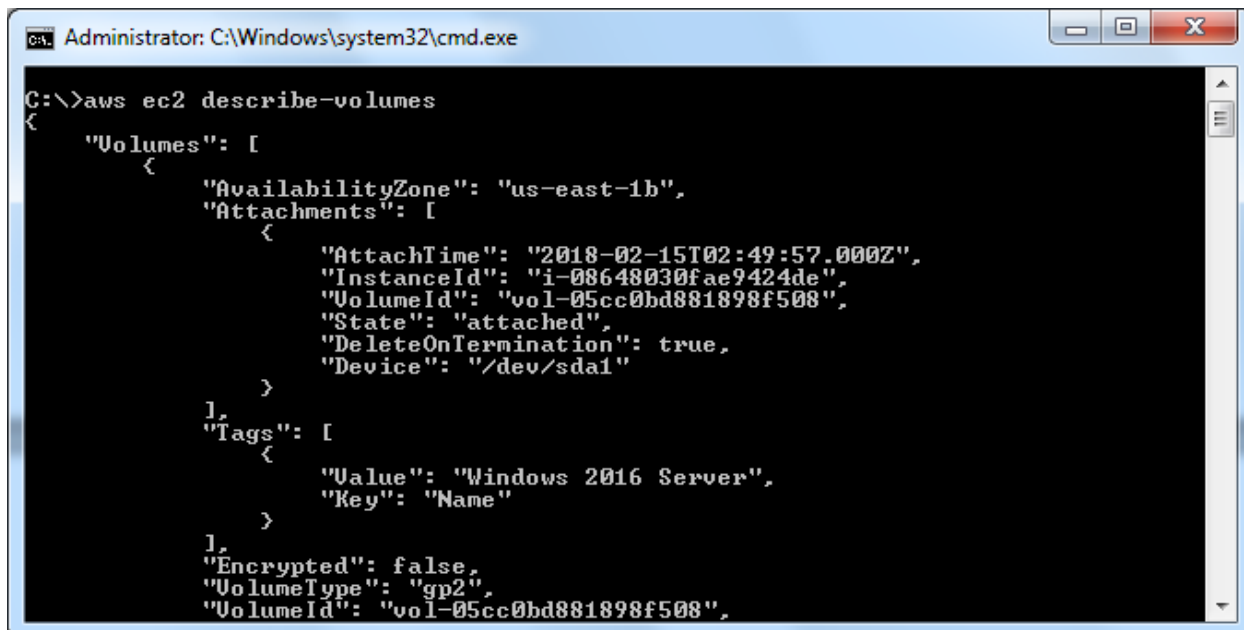
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
Windows 2016 Server	i-08648030fae9424de	t2.micro	us-east-1b	stopping		None	

Below the instance list, the details for the selected instance 'i-08648030fae9424de (Windows 2016 Server)' are shown. The 'Description' tab is active, displaying the following information:

Property	Value
Instance ID	i-08648030fae9424de
Instance state	stopping
Instance type	t2.micro
Elastic IPs	
Availability zone	us-east-1b
Security groups	launch-wizard-1, view inbound rules
Public DNS (IPv4)	-
IPv4 Public IP	34.227.112.105
IPv6 IPs	-
Private DNS	ip-192-168-2-5.ec2.internal
Private IPs	192.168.2.5
Secondary private IPs	

Type

Aws ec2 describe-volumes

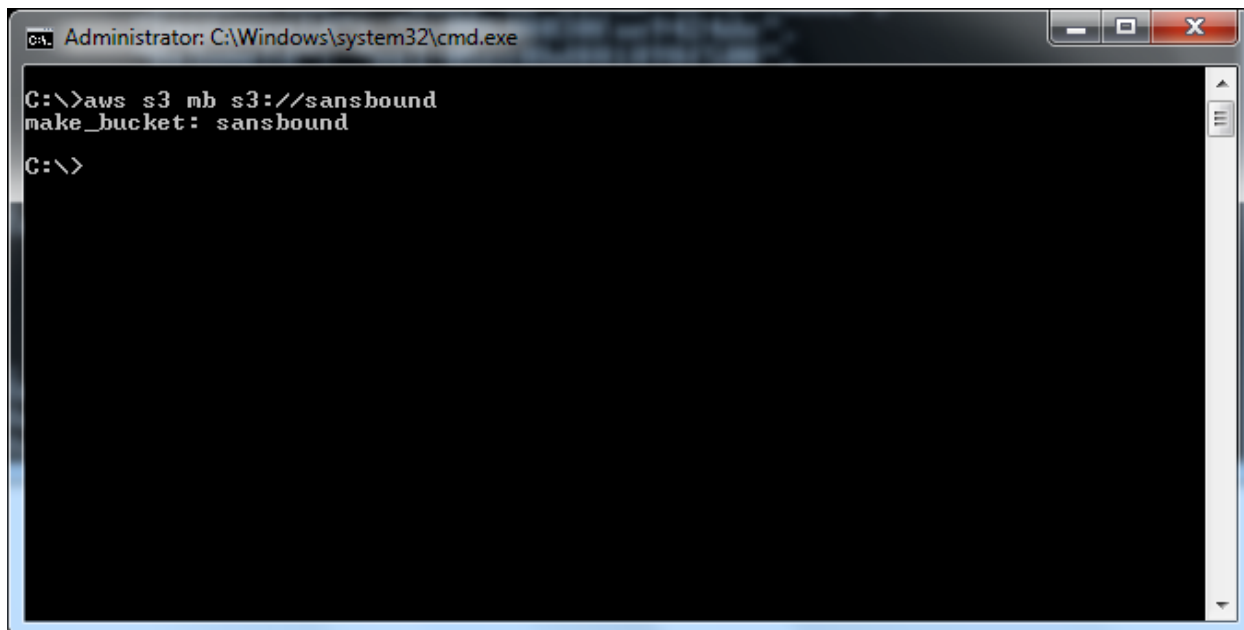


```
Administrator: C:\Windows\system32\cmd.exe

C:\>aws ec2 describe-volumes
{
  "Volumes": [
    {
      "AvailabilityZone": "us-east-1b",
      "Attachments": [
        {
          "AttachTime": "2018-02-15T02:49:57.000Z",
          "InstanceId": "i-08648030fae9424de",
          "VolumeId": "vol-05cc0bd881898f508",
          "State": "attached",
          "DeleteOnTermination": true,
          "Device": "/dev/sda1"
        }
      ],
      "Tags": [
        {
          "Value": "Windows 2016 Server",
          "Key": "Name"
        }
      ],
      "Encrypted": false,
      "VolumeType": "gp2",
      "VolumeId": "vol-05cc0bd881898f508",
```

Type

Aws s3 mb s3://sansbound

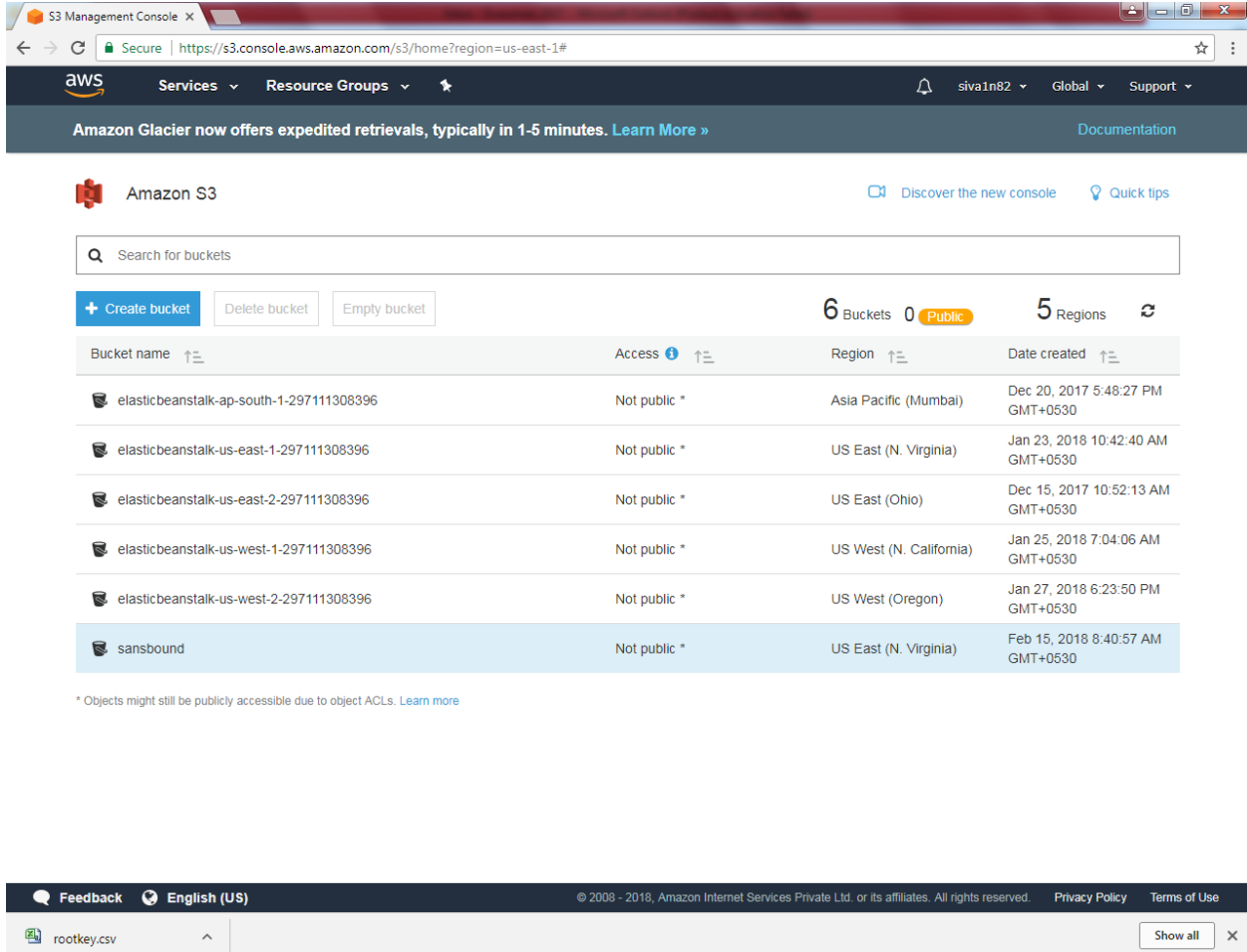


```
Administrator: C:\Windows\system32\cmd.exe

C:\>aws s3 mb s3://sansbound
make_bucket: sansbound

C:\>
```

Go to S3 and able to see that bucket has been created



The screenshot shows the AWS S3 Management Console interface. At the top, there's a navigation bar with the AWS logo, 'Services', 'Resource Groups', and a user profile 'siva1n82'. Below this is a banner for Amazon Glacier. The main heading is 'Amazon S3' with links to 'Discover the new console' and 'Quick tips'. A search bar is present with the text 'Search for buckets'. Below the search bar are three buttons: 'Create bucket', 'Delete bucket', and 'Empty bucket'. To the right of these buttons, it shows '6 Buckets', '0 Public', and '5 Regions'. The main content is a table of buckets:

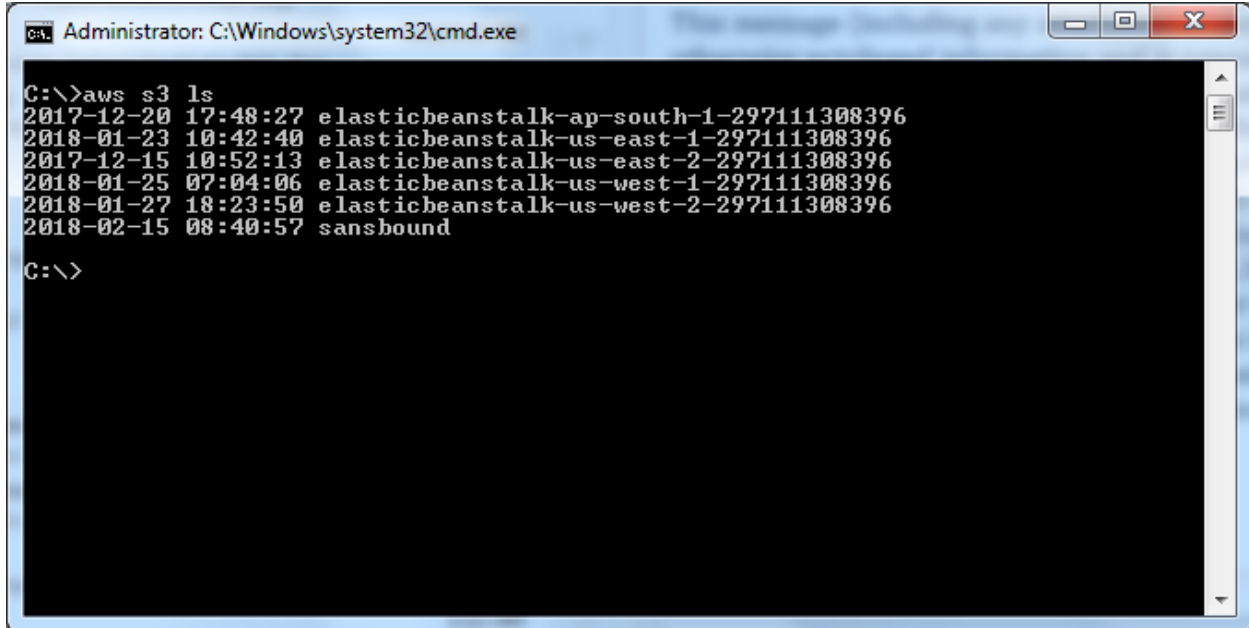
Bucket name	Access	Region	Date created
elasticbeanstalk-ap-south-1-297111308396	Not public *	Asia Pacific (Mumbai)	Dec 20, 2017 5:48:27 PM GMT+0530
elasticbeanstalk-us-east-1-297111308396	Not public *	US East (N. Virginia)	Jan 23, 2018 10:42:40 AM GMT+0530
elasticbeanstalk-us-east-2-297111308396	Not public *	US East (Ohio)	Dec 15, 2017 10:52:13 AM GMT+0530
elasticbeanstalk-us-west-1-297111308396	Not public *	US West (N. California)	Jan 25, 2018 7:04:06 AM GMT+0530
elasticbeanstalk-us-west-2-297111308396	Not public *	US West (Oregon)	Jan 27, 2018 6:23:50 PM GMT+0530
sansbound	Not public *	US East (N. Virginia)	Feb 15, 2018 8:40:57 AM GMT+0530

Below the table, a note states: '* Objects might still be publicly accessible due to object ACLs. [Learn more](#)'

At the bottom, there's a footer with 'Feedback', 'English (US)', copyright information '© 2008 - 2018, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.', 'Privacy Policy', and 'Terms of Use'. A file named 'rootkey.csv' is shown at the bottom with a 'Show all' button.

Type

Aws s3 ls



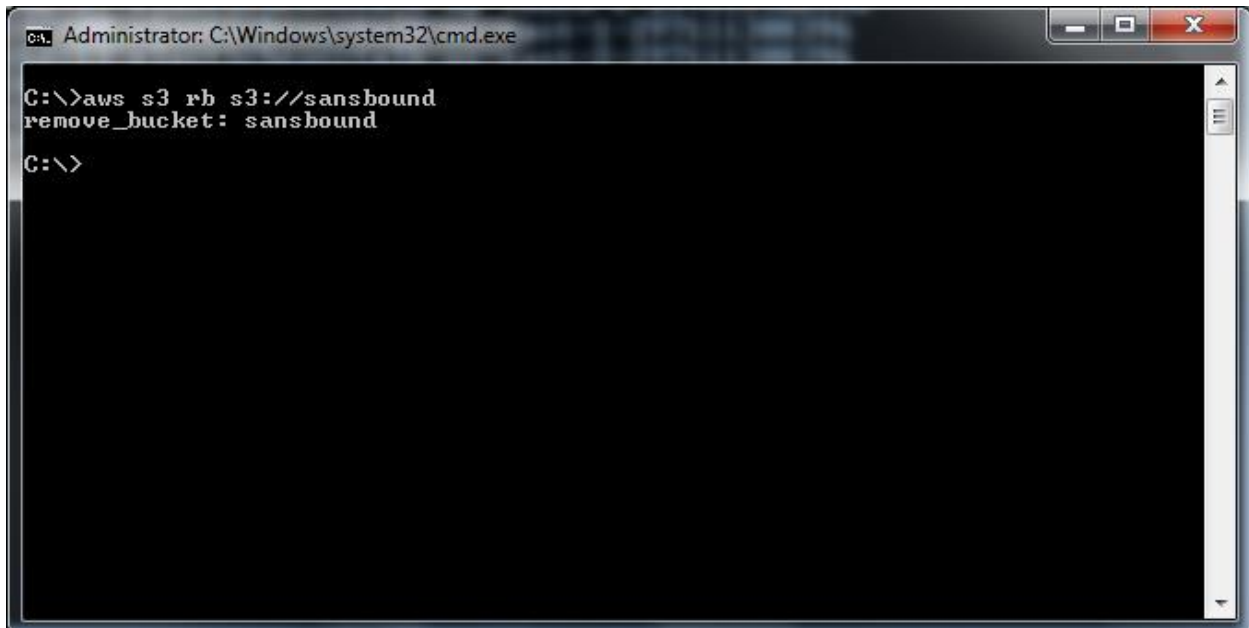
```
Administrator: C:\Windows\system32\cmd.exe

C:\>aws s3 ls
2017-12-20 17:48:27 elasticbeanstalk-ap-south-1-297111308396
2018-01-23 10:42:40 elasticbeanstalk-us-east-1-297111308396
2017-12-15 10:52:13 elasticbeanstalk-us-east-2-297111308396
2018-01-25 07:04:06 elasticbeanstalk-us-west-1-297111308396
2018-01-27 18:23:50 elasticbeanstalk-us-west-2-297111308396
2018-02-15 08:40:57 sansbound

C:\>
```

Type

Aws s3 rb s3://sansbound



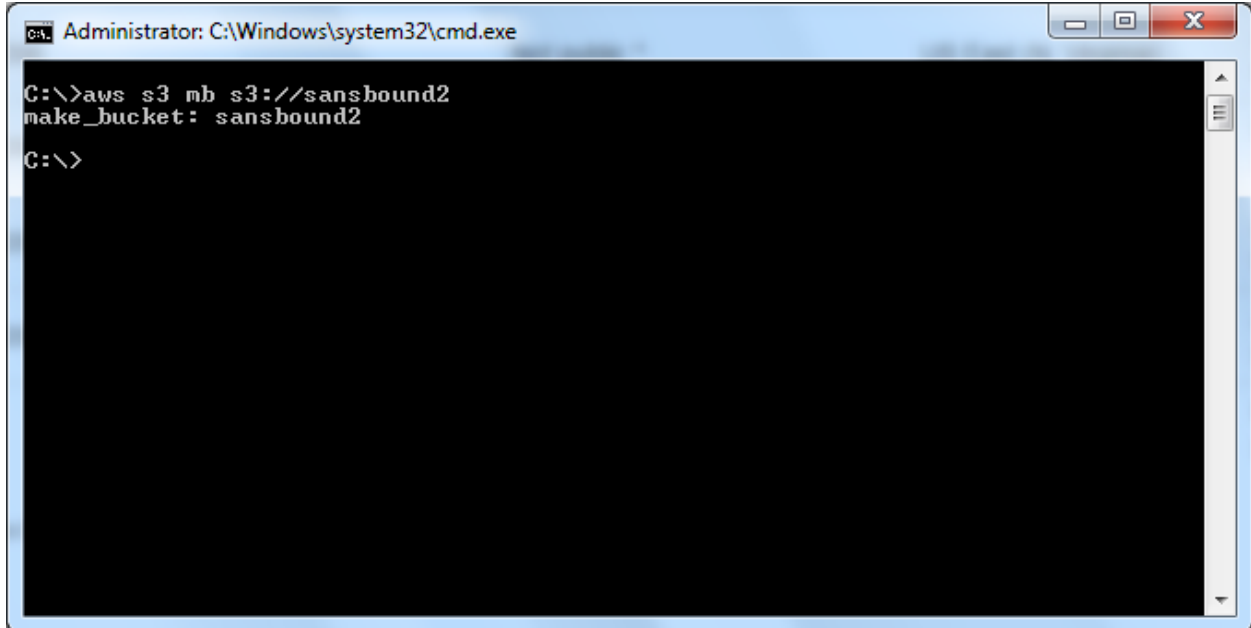
```
Administrator: C:\Windows\system32\cmd.exe

C:\>aws s3 rb s3://sansbound
remove_bucket: sansbound

C:\>
```

Type

Aws s3 mb s3://sansbound2



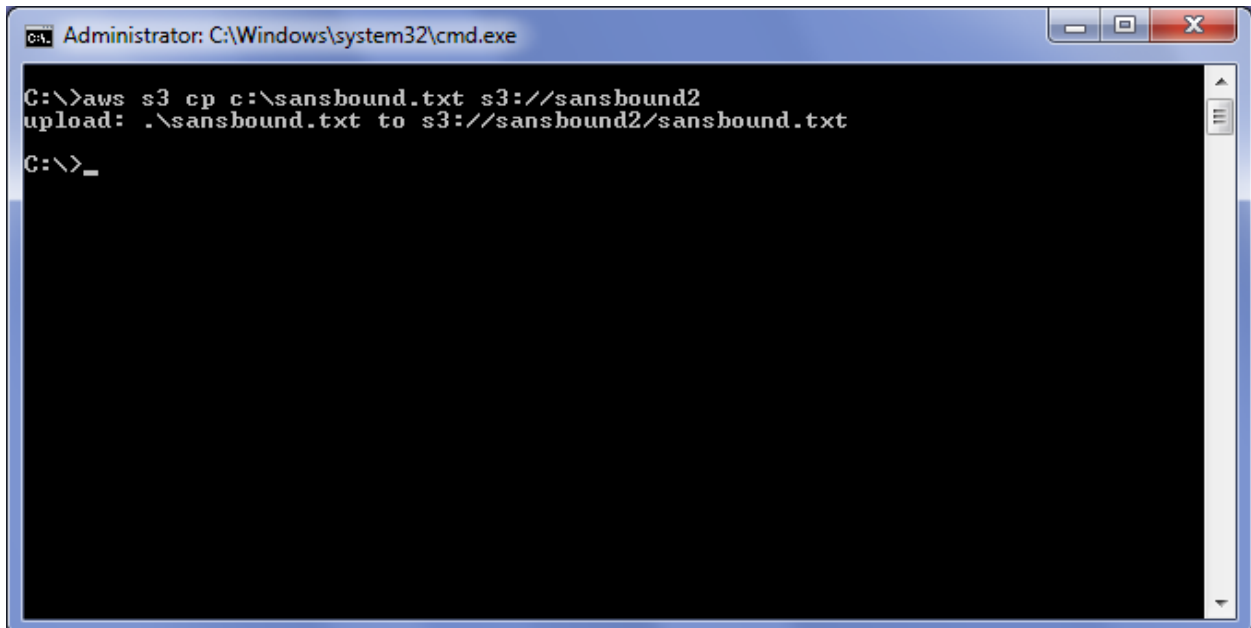
```
Administrator: C:\Windows\system32\cmd.exe

C:\>aws s3 mb s3://sansbound2
make_bucket: sansbound2

C:\>
```

Type

Aws s3 cp c:\sansbound.txt s3://sansbound2



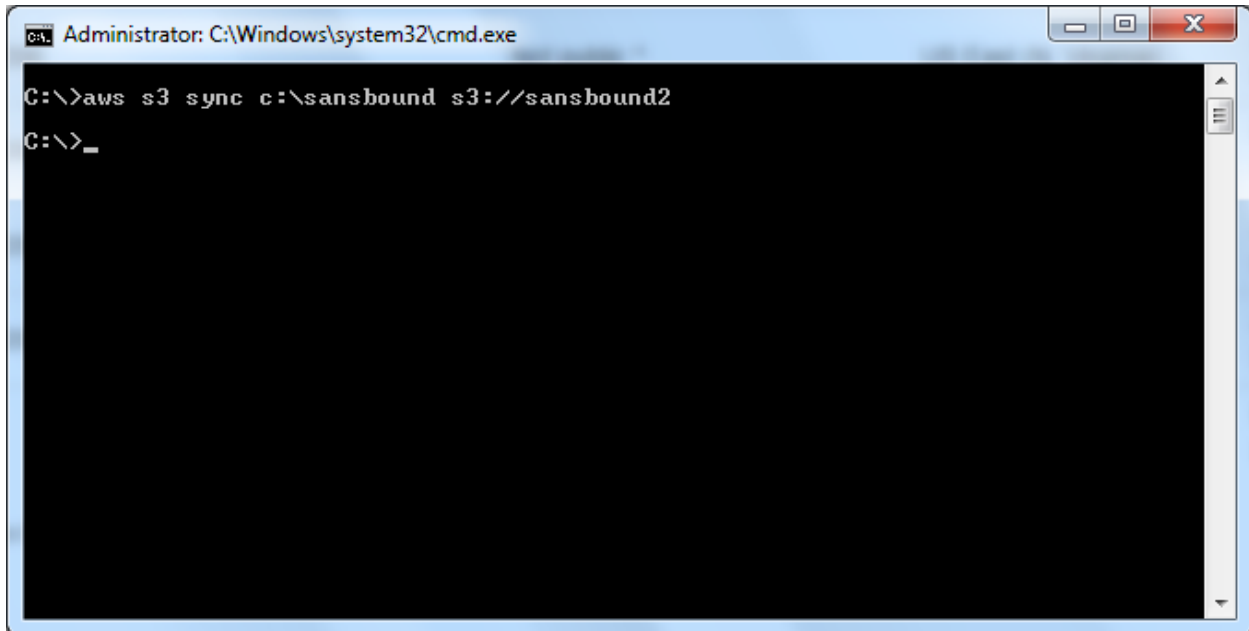
```
Administrator: C:\Windows\system32\cmd.exe

C:\>aws s3 cp c:\sansbound.txt s3://sansbound2
upload: .\sansbound.txt to s3://sansbound2/sansbound.txt

C:\>_
```

Type

Aws s3 sync c:\sansbound s3://sansbound2

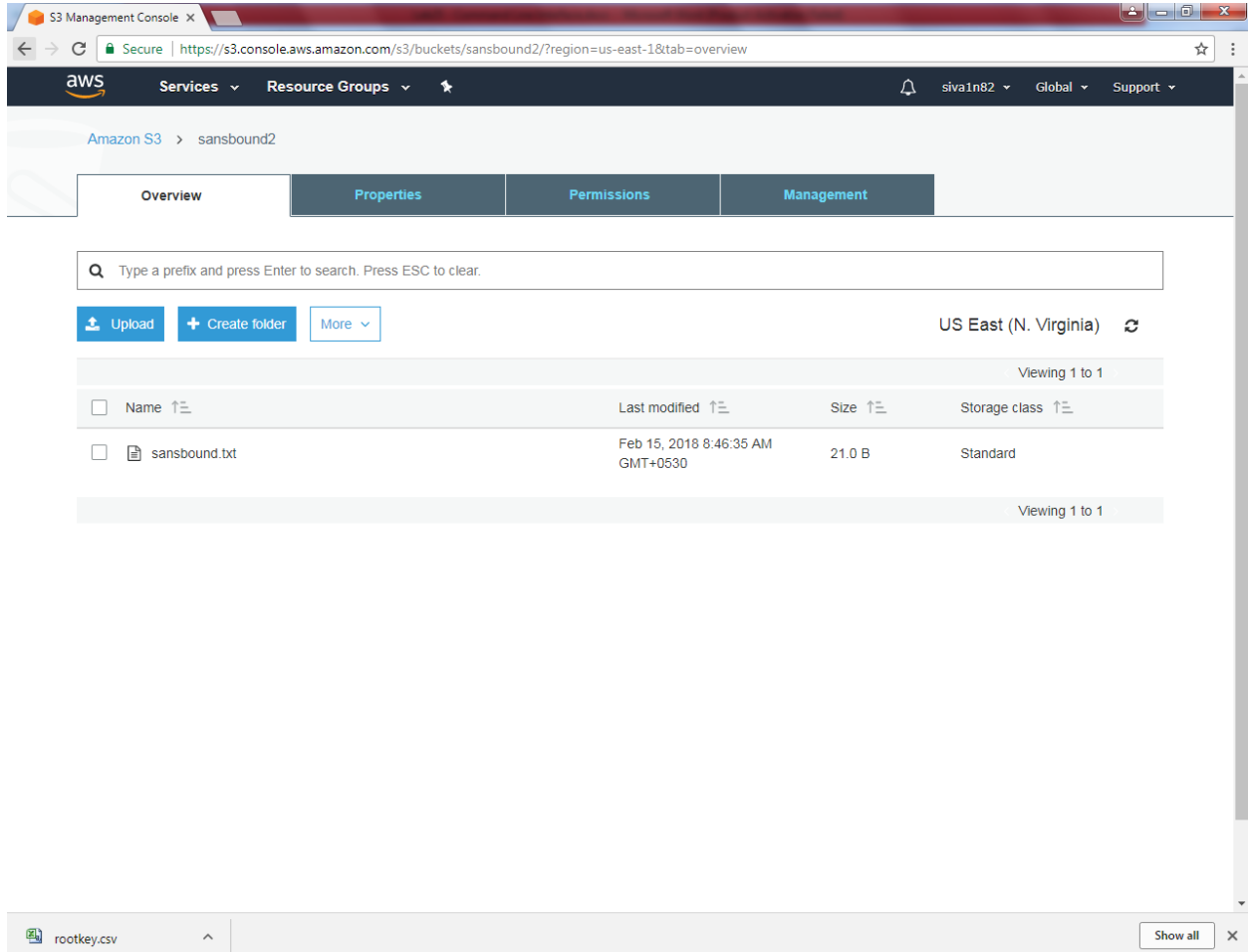


```
C:\>Administrator: C:\Windows\system32\cmd.exe

C:\>aws s3 sync c:\sansbound s3://sansbound2

C:\>_
```

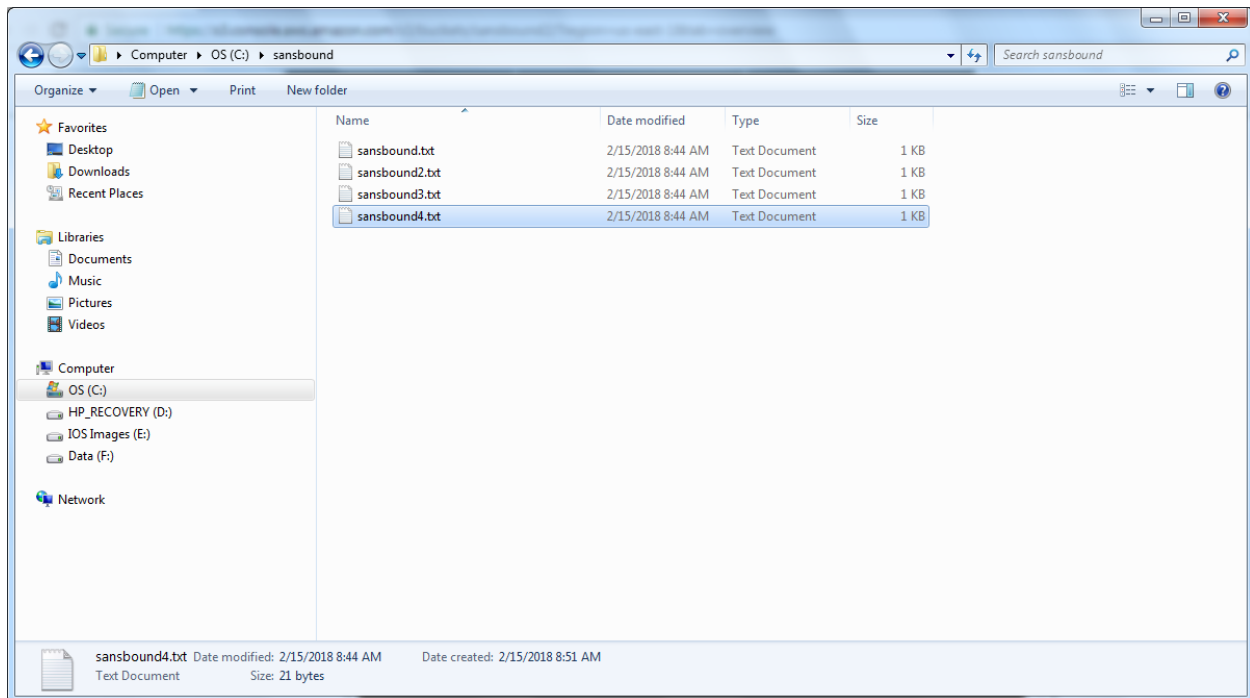
You can able to see the file in sanbound2 bucket.



The screenshot shows the AWS S3 Management Console interface. The breadcrumb navigation indicates the path is **Amazon S3** > **sansbound2**. The **Overview** tab is selected, with other tabs for **Properties**, **Permissions**, and **Management**. A search bar is present with the placeholder text "Type a prefix and press Enter to search. Press ESC to clear." Below the search bar are buttons for **Upload**, **Create folder**, and a **More** dropdown menu. The region is set to **US East (N. Virginia)**. A table displays the contents of the bucket, showing one object: **sansbound.txt**. The table headers are **Name**, **Last modified**, **Size**, and **Storage class**. The object **sansbound.txt** was last modified on **Feb 15, 2018 8:46:35 AM GMT+0530** and has a size of **21.0 B** with a **Standard** storage class. At the bottom of the console, a task list shows a task named **rootkey.csv** with a **Show all** button.

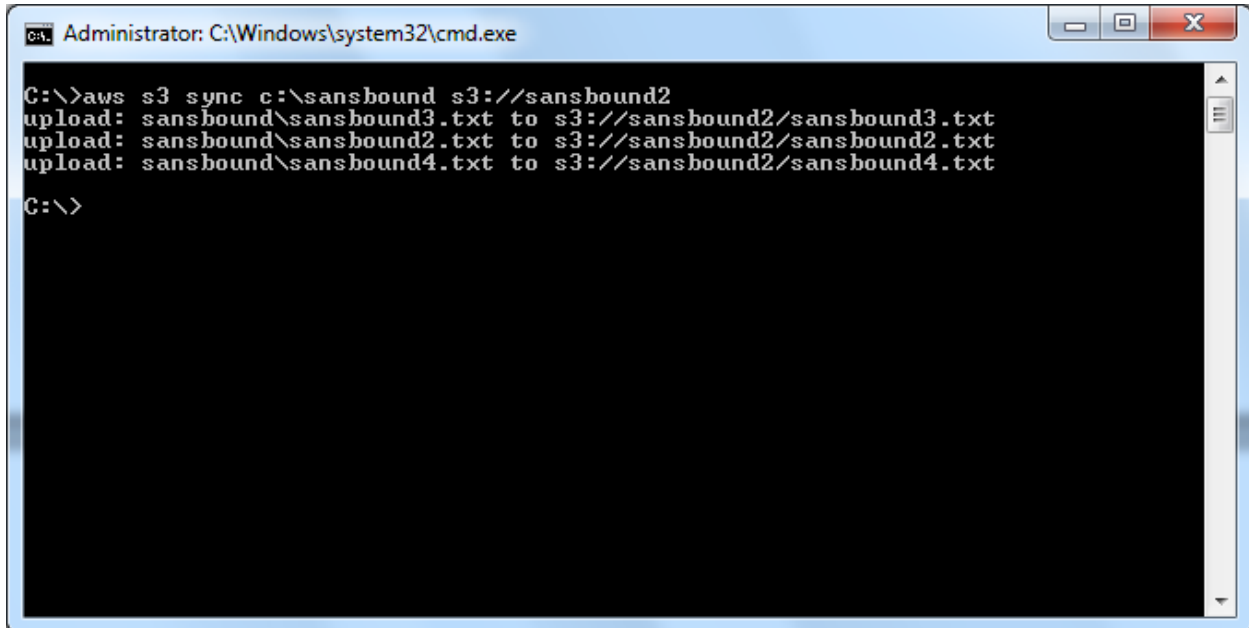
Name	Last modified	Size	Storage class
sansbound.txt	Feb 15, 2018 8:46:35 AM GMT+0530	21.0 B	Standard

Now I will copy the files into sansbound2 bucket.



Type

```
s3 sync c:\sansbound s3://sansbound2
```



```
Administrator: C:\Windows\system32\cmd.exe

C:\>aws s3 sync c:\sansbound s3://sansbound2
upload: sansbound\sansbound3.txt to s3://sansbound2/sansbound3.txt
upload: sansbound\sansbound2.txt to s3://sansbound2/sansbound2.txt
upload: sansbound\sansbound4.txt to s3://sansbound2/sansbound4.txt
C:\>
```