

# AWS Multi-Factor Authentication (MFA)

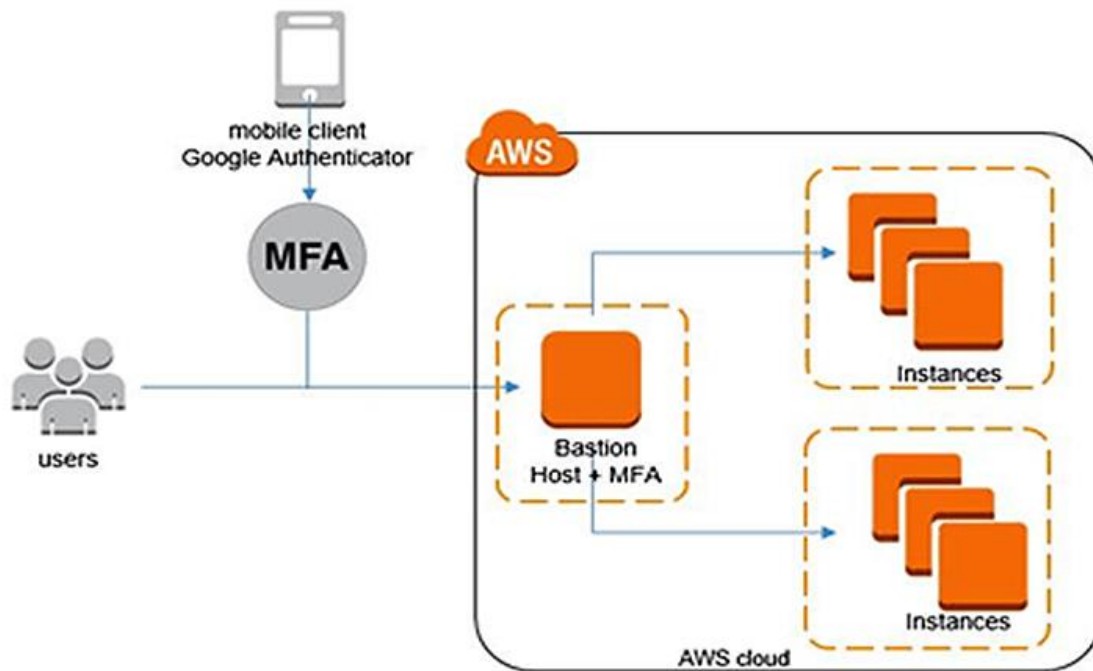
## Overview of AWS MFA

**AWS Multi-Factor Authentication (MFA)** is a simple best practice that adds an extra layer of protection on top of your username and password. With MFA enabled, when a user signs in to an [AWS Management Console](#), they will be prompted for their user name and password (the first factor is what they know), as well as for an authentication code from their AWS MFA device (the second factor is what they have). Taken together, these multiple factors provide increased security for your AWS account settings and resources.



## Why AWS MFA is Required

- Users have access to your account and can possibly change configurations and delete resources in your AWS account, so to overcome this it is required
- If you want to protect your root accounts and IAM user.
- Even if the password is stolen or hacked, the account is not compromised.
- When you enable this authentication for the root user, it affects only the root user credentials. IAM users in the account are distinct identities with their own credentials, and each identity has its own MFA configuration.



## MFA Device Options In AWS

The following are the MFA device options in AWS:

- **Virtual MFA Device:** Support for multiple tokens on a single device e.g **Google Authenticator** (Phone Only) **Authy** (Multi-Device)
- **Universal 2nd Factor (U2F) Security Key:** Supports multiple root and IAM users using a single security key. e.g **Yubikey** by Yubico (Third Party)
- **Hardware Key Fob MFA Device:** Provided by Gemalto (Third Party)
- **Hardware Key Fob MFA Device AWS GovCloud (US):** Provided by SurePassID (Third Party)

## AWS EC2 Instance | Steps to Connect Windows EC2 Instance

## What Is Amazon EC2?



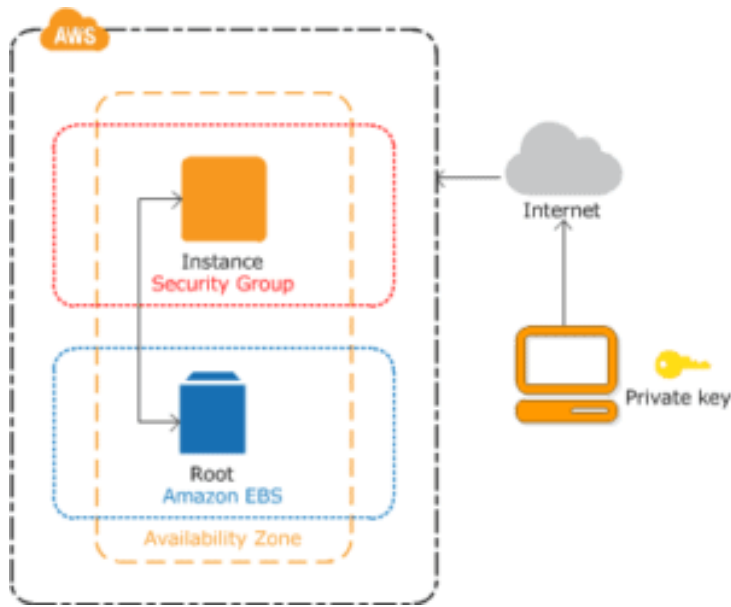
Amazon EC2 is a web service that provides **secure** and **resizable** compute capacity in the cloud to users. The EC2 allows us to configure the capacity, and it also provides complete control of all your computing resources. An AWS user can increase or decrease instance capacity by using the EC2 web interface or an application programming interface (API). A developer can automatically code an application to scale up and down instances with **AWS Auto Scaling**. To manage multiple instances, a user can also define an autoscaling policy and group.

## Why We Use Amazon Elastic Compute Cloud?

Organizations have to invest a considerable amount in buying hardware components, and managing them is more hectic than buying, so the **EC2 Amazon eliminates the investment in hardware upfront** so that you can develop and deploy applications faster. Using EC2, you can launch as many or as few virtual servers as your need, manage storage, and configure networking and security. EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

### How does AWS EC2 Work?

It's pretty simple to get up and running with EC2. You have a choice of **pre-configured, templated Amazon Machine Images (AMI)** to use for a quick launch. Or, if you prefer, you can create your own AMI that contains all of your libraries, data, applications, and relevant configuration settings. EC2 allows you to customize settings by configuring security and network access. Because you can instantly scale your VM environment to address usage spikes or drops, you have the power to control the number of resources being used at any point in time. The service's **elasticity** supports the minimized costs of a "**pay-for-what-you-use**" billing model.



### Features of Amazon Elastic Compute Cloud

- **Reliable:** Amazon EC2 offers a highly reliable environment where the replacement of instances is rapidly possible.
- **Secure:** **Amazon Virtual Private Cloud** where EC2 works and provides a secured and robust network to resources.
- **Flexible tools:** EC2 Amazon provides the tools for developers and system administrators to build failure applications and isolate themselves from common failure situations.
- **Inexpensive:** EC2 Amazon wants us to pay only for the resources that we use on an hourly basis. It includes multiple purchase plans such as On-Demand Instances, Reserved Instances, Spot Instances, etc. which we can choose as per our needs.

### Use Cases of AWS EC2 Instances

- **Hosting environments:** Primarily EC2 uses for hosting a website and application on the cloud. The best part of EC2 Amazon is that it provides a **Dynamic** and **Scalable** environment where its compute capacity can grow as per the need of the application. Companies like Netflix and Reddit are the best proof of EC2 hosting success.
- **Backup and disaster recovery:** Companies hold EC2 as a medium for performing disaster recovery for both active and passive environments. EC2 turns quickly in case of an emergency, which means businesses have access to the **fastest failover with minimal downtime** of their application.
- **High-performance computing:** The EC2 Amazon provides **virtualized servers** with high-performance networking and high compute power. Organizations like NASA and Pfizer employ high-performance computing using the EC2 Amazon Instance.
- **Development and test environment:** EC2 Amazon is scalable, so now the organization can create and deploy **large-scale testing and development** environment with unprecedented ease. EC2 eliminates the investments in hardware and also provides a **scalable** solution.
- **Banking and financial sector:** These sectors demand maximum scalability and security, and EC2 provides both. Amazon cloud builds trust by providing **highly secure** services. Retail and commercial banks, insurance providers, and significant payment platforms rely on AWS.

## Amazon Machine Images (AMIs)

Amazon Machine Images or AMIs are **pre-configured virtual machine images** that you can use to launch EC2 instances. You can create and **customize your own AMIs** to save time and effort in setting up new instances. AMIs provide a fast and reliable way to launch instances with your desired operating system, applications, and security settings.

1. **Amazon Linux 2 / 2023 AMI:** This is a pre-configured image of the Amazon Linux operating system that includes commonly used software packages and tools. It comes with AWS CLI built in. It is optimized for running on AWS and can be used for various types of workloads.
2. **Windows Server AMIs:** These are images of the Microsoft Windows Server operating system that can be used to launch instances for running Windows-based applications.
3. **Database AMIs:** These are images of popular databases such as Amazon RDS, MySQL, Oracle, PostgreSQL, and MongoDB. These AMIs can be used to launch instances with a pre-configured database environment.
4. **Application AMIs:** These are images of popular applications such as WordPress, Magento, Drupal, Joomla, and many others. These AMIs can be used to launch instances with a pre-configured application environment.
5. **Machine Learning AMIs:** These are images of Amazon Machine Learning environments such as Amazon SageMaker, TensorFlow, Apache MXNet, and others. These AMIs can be used to launch instances with pre-installed machine learning libraries and tools.
6. **Security AMIs:** These are images of security-focused software such as firewalls, intrusion detection systems, and vulnerability scanners. These AMIs can be used to launch instances with a pre-configured security environment.

## Instance Types

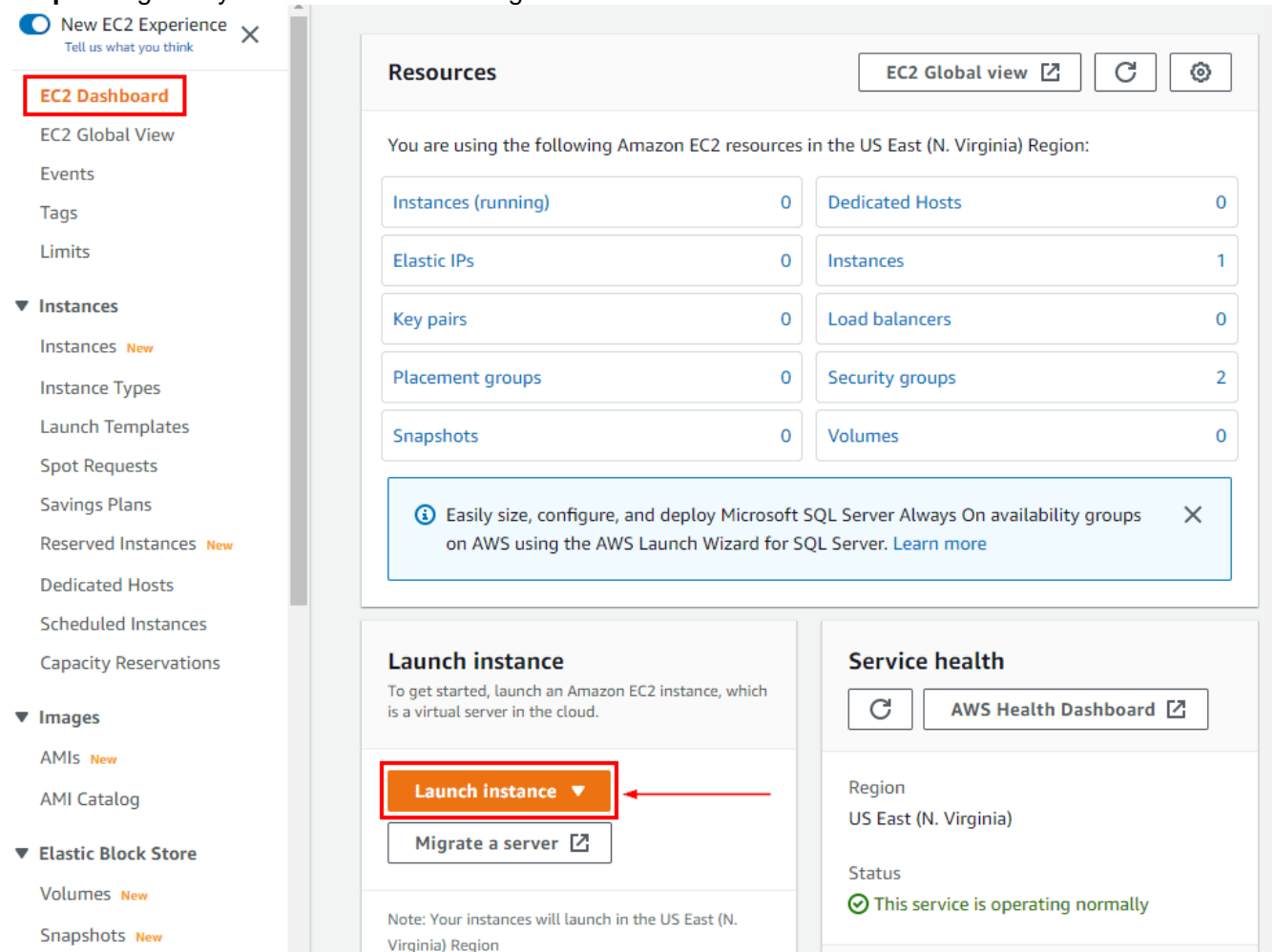
AWS EC2 offers a wide range of instance types to choose from, each designed to meet specific computing needs. These instance types differ in terms of their **CPU, memory, storage, and networking capacity**. By understanding the different instance types and their capabilities, you can choose the right one to optimize your workload performance and cost-effectiveness.

1. **General Purpose Instances:** These instances are suitable for a wide range of workloads, such as small to medium-sized databases, web servers, and development environments. For example, a t2.micro instance can be used for running a small-scale WordPress website, while an m5.large instance can be used for running a small-scale e-commerce website. Examples of general purpose instances include t2, m5, and m6g.
2. **Compute Optimized Instances:** These instances are ideal for compute-intensive workloads, such as high-performance computing (HPC), scientific modeling, and data analytics. Examples of compute optimized instances include c5, c6g, and c6gd.
3. **Memory Optimized Instances:** These instances are designed for workloads that require large amounts of memory, such as in-memory databases, big data processing, and analytics. Examples of memory optimized instances include r5, r6g, and x1e.
4. **Storage Optimized Instances:** These instances are suitable for storage-intensive workloads that require high disk throughput and low latency, such as data warehousing and log processing. Examples of storage optimized instances include i3, i3en, and d3.
5. **GPU Instances:** These instances are designed for workloads that require high-performance GPU resources, such as machine learning, deep learning, and graphics rendering. Examples of GPU instances include p3, g4, and inf1.
6. **High Performance Computing Instances:** These instances are designed for high-performance computing (HPC) workloads that require high network performance and low latency, such as scientific modeling, financial simulations, and engineering simulations. Examples of HPC instances include hpc6id, and Hpc6a.

## 8 Steps To Create AWS Windows EC2 Instance

**Note:** If you don't have an AWS account, check our blog on how to create [AWS Free Tier Account](#)

**Step 1:** Log in to your **AWS account** and go to the **EC2 dashboard** to **launch** a new instance.



**EC2 Dashboard**

EC2 Global View

Events

Tags

Limits

▼ **Instances**

Instances *New*

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances *New*

Dedicated Hosts

Scheduled Instances

Capacity Reservations

▼ **Images**

AMIs *New*

AMI Catalog

▼ **Elastic Block Store**

Volumes *New*

Snapshots *New*

**Resources** EC2 Global view Refresh Settings

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

Instances (running)	0	Dedicated Hosts	0
Elastic IPs	0	Instances	1
Key pairs	0	Load balancers	0
Placement groups	0	Security groups	2
Snapshots	0	Volumes	0

*Info* Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. [Learn more](#) Close

**Launch instance**

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

**Launch instance** ▼

[Migrate a server](#)

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

**Service health** Refresh AWS Health Dashboard

Region

US East (N. Virginia)

Status

✓ This service is operating normally

**Step 2:** In the Name and Tags step you can add tags to an instance, here tags help you to enable categorizing AWS resources in different ways, for example, by owner, environment, or purpose. For example, you could define a set of tags for your account's EC2 instances that help you track each instance's owner and stack level.

EC2 > Instances > Launch an instance

## Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

### Name and tags

Name

K21-Windows-Server

[Add additional tags](#)

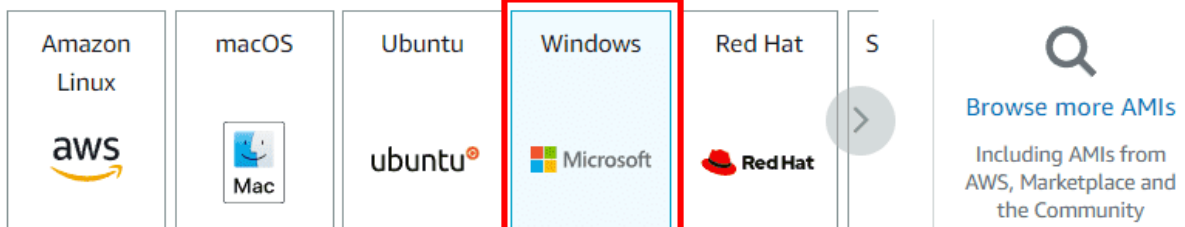
**Step 3:** Select **Windows** under QuickStart and Select **Microsoft Windows Server 2022 Base** AMI. You can also select other AMI as per your need but here we launching a Windows Server, so we have to select the Windows Server 2022 Base AMI.

## ▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

 Search our full catalog including 1000s of application and OS images

### Quick Start



Amazon Machine Image (AMI)

Microsoft Windows Server 2022 Base

ami-03cf1a25c0360a382 (64-bit (x86))

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▼

### Description

Microsoft Windows Server 2022 Full Locale English AMI provided by Amazon

Architecture

AMI ID

64-bit (x86)

ami-03cf1a25c0360a382

Verified provider

**Step 3:** Select the **t2.micro** instance type, if you want you may select another instance type but they are chargeable so we choose the t2.micro instance type which is eligible for the free tier and limited resources.



▼ Instance type [Info](#)

Instance type

t2.micro Free tier eligible [Compare instance types](#)

Family: t2 1 vCPU 1 GiB Memory

Q |

t1.micro Free tier eligible

Family: t1 1 vCPU 0.612 GiB Memory

t2.nano

Family: t2 1 vCPU 0.5 GiB Memory

t2.micro Free tier eligible

Family: t2 1 vCPU 1 GiB Memory

t2.small

Family: t2 1 vCPU 2 GiB Memory

t2.medium

Family: t2 2 vCPU 4 GiB Memory

t2.large

Family: t2 2 vCPU 8 GiB Memory

t2.xlarge

Family: t2 4 vCPU 16 GiB Memory

t2.2xlarge

Family: t2 8 vCPU 32 GiB Memory

t3.nano

Family: t3 2 vCPU 0.5 GiB Memory

t3.micro

selected key pair before you launch

[Create new key pair](#)

decrypted password to connect to

[Edit](#)

**Step 4:** Select an existing key pair or create a new one, we will Create a new one, enter the name of the Key-pair as Windows-Key and Create the Key Pair.

▼ **Key pair (login)** [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Select ▼



**Create new key pair**

For Windows instances, you use a key pair to decrypt the administrator password. You then use the decrypted password to connect to your instance.

**Create key pair**



Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more](#)

Key pair name

Windows-Key

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Private key file format



.pem

For use with OpenSSH



.ppk

For use with PuTTY

Cancel

**Create key pair**



**Also Read:** Our Previous Blog On [SDLC Automation](#)

**Step 5:** Now, keep everything default and click on **Launch Instance**.

▼ Configure storage [Info](#)

Advanced

1x  GiB  Root volume (Not encrypted)

*Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage*

Add new volume

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

0 x File systems Edit

► Advanced details [Info](#)

▼ Summary

Number of instances [Info](#)

Software Image (AMI)  
Microsoft Windows Server 2022 ...[read more](#)  
ami-0bd3f43f107376d6b

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
New security group

Storage (volumes)  
1 volume(s) - 30 GiB

Cancel → Launch instance

**Step 6:** Now Click on **View all Instances**.

[EC2](#) > [Instances](#) > Launch an instance

✓ Success

Successfully initiated launch of instance (i-01bac39bf43f9790d)

► Launch log

Next Steps

Get notified of estimated charges  
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier)

How to connect to your instance  
Your instance is launching and it might be a few minutes until it is in the running state, when it will be ready for you to use  
Click View Instances to monitor your instance's status. Once your instance is in the 'running' state, you can connect to it from the Instances screen. Find out [how to connect to your instance](#)  
View more resources to get you started

→ View all Instances

**Step 7:** Here, you shall see your instance is launching and the Status check is Initializing, wait for some time.

Instances (1/1) <a href="#">Info</a>									
<input type="text" value="Search"/> <span>◀ 1 ▶</span>									
✓	Name	Instance ID	Instance state	Inst...	Status check	Alarm status	Availa...		
✓	-	i-01bac39bf43f9790d	Running	t2.micro	Initializing	No alarms	us-east-1c	e...	

**Step 8:** Refresh and you shall see your instance is Up and Running, and the Status check has changed to 2/2 checks.

Instances (1/1) Info

Search

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Inst...	Status check	Alarm status	Availa...
<input checked="" type="checkbox"/>	-	i-01bac39bf43f9790d	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c

Instance: i-01bac39bf43f9790d

Select an instance above

Details Security Networking Storage Status checks Monitoring Tags

Instance summary Info

Instance ID i-01bac39bf43f9790d	Public IPv4 address 3.93.46.181   <a href="#">open address</a>	Private IPv4 addresses 172.31.19.165
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-3-93-46-181.compute-1.amazonaws.com   <a href="#">open address</a>

## 10 Steps To Connect AWS Windows EC2 Instance

**Step 1:** Firstly we have to select the Windows instance **From the Running Instance** of the EC2 dashboard and click on **Connect**.

Instances (1/1) Info

Search

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Inst...	Status check	Alarm status	Availa...
<input checked="" type="checkbox"/>	-	i-01bac39bf43f9790d	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c

Instance: i-01bac39bf43f9790d

Select an instance above

Details Security Networking Storage Status checks Monitoring Tags

Instance summary Info

**Step 2:** Here we have to select the **RDP (Remote desktop protocol) Client** and then **Download** the RDP File and save it somewhere safe then, we need a password to access the RDP file, so click on **Get Password**.

## Connect to instance [Info](#)

Connect to your instance i-01bac39bf43f9790d using any of these options

Session Manager

**RDP client**

EC2 Serial Console


You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:

**Download remote desktop file**


1

When prompted, connect to your instance using the following details:

Public DNS

 ec2-3-93-46-181.compute-1.amazonaws.com

User name

 Administrator

Password

**Get password**

2

If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.

Cancel

**Step 3:** At this step of launching, we have to upload the Key-pair (the key which we have created in the earlier step). Click on **Browse** and then select the key and click on **Decrypt Password**. This provides us with a usable password.

## Get Windows password [Info](#)

Retrieve and decrypt the initial Windows administrator password for this instance.

To decrypt the password, you will need your key pair for this instance.



Key pair associated with this instance

Windows-Key

Browse to your key pair:



Browse



Windows-Key.pem

1.674KB

Or copy and paste the contents of the key pair below:

-----BEGIN RSA PRIVATE KEY-----

```
MIIIEogIBAAKCAQEApkdhmbkYHF9S6/SnTrtE6Ocg7ROUtGZmtyejnlBjygBGaaS
YFbx1WFOkVsheHH8O8O4UgO5eW2Q96MO+B6s+9kh0FwfSr0B03de7Dq6H4A8t5m6
eKYXakzS48OHojK72Va0mGwq2TU3XyssPR5boSEnZNwUcSd/vtXqsoJGMvC7FSed
6CWHYHYIbVGu/IZDpr27agJnjAESnlhW8prPX6cqcfcZRaRfsG5bH62naiD2t0H0
w7SFrTYC/vyxh9v761XKe7JEzuAEkjuDgZKj2VC9k+x03E/SfiCk13+5oGm6JZvS
UyJK0jVULFv+yjnUg6yf7wKAlOKRgmbgmAtdywIDAQABAoIBAElA608FvItsODAU
+rCWULY7bLboazB6JhzAaLtF6PT0/xa3dNN1aNIjmFAa/zmQrGTgClmtVIGPLbKr
```

Cancel

Decrypt password

**Step 4:** After submitting the Key-pair here the **Password is Generated**, copy and save it somewhere safe.

✔ **Password Decryption Successful**  
The password for instance i-01bac39bf43f9790d was successfully decrypted.

EC2 > Instances > i-01bac39bf43f9790d > Connect to instance

### Connect to instance [Info](#)

Connect to your instance i-01bac39bf43f9790d using any of these options

[Session Manager](#) | **RDP client** | [EC2 Serial Console](#)

You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:

[Download remote desktop file](#)

When prompted, connect to your instance using the following details:

Public DNS

ec2-3-93-46-181.compute-1.amazonaws.com

User name

Administrator



Password

-z@sLVWmm88lNy-GXw5sGlZ;?hVZ4sTl

If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.

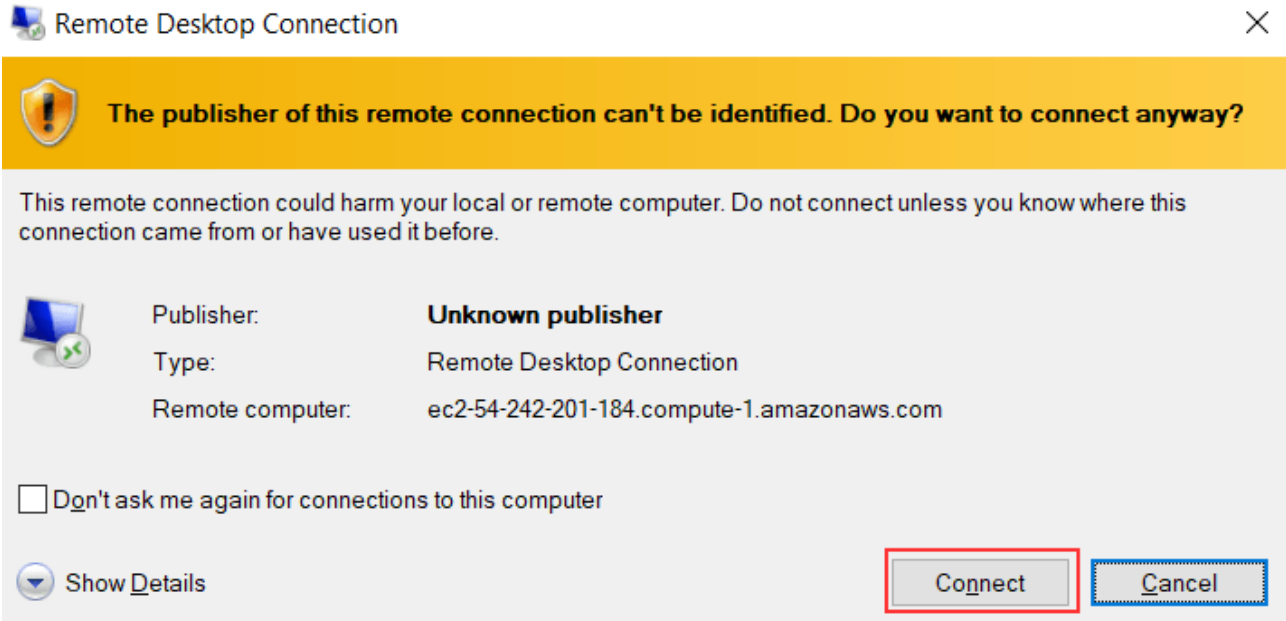
[Cancel](#)

**Step 5:** Now open the **Remote Desktop File** from downloads for launching the Windows instance. If your local computer is a Mac, you will need to download "Microsoft Remote Desktop" from the App Store to be able to open your RDP file.

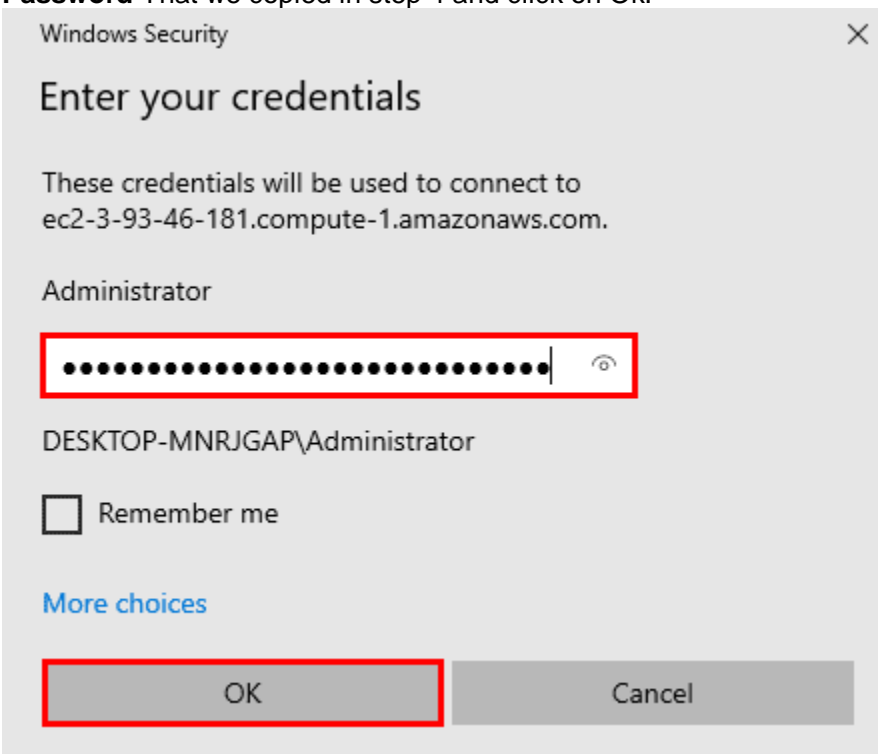
Name	Date modified	Type	Size
▼ Today (3)			
 ec2-3-93-46-181.compute-1.amazonaws....	24-04-2022 12:40	Remote Desktop ...	1 KB
 Windows-Key.pem	24-04-2022 12:18	PEM File	2 KB

**Step 6:** After opening the RDP file click on **Connect** to launch the Window instance.

**Note:** Windows has the **Remote Desktop Connection Application** pre-installed; so for other OS like mac you need to download the **Microsoft Remote Desktop** app from the **Mac App Store**.

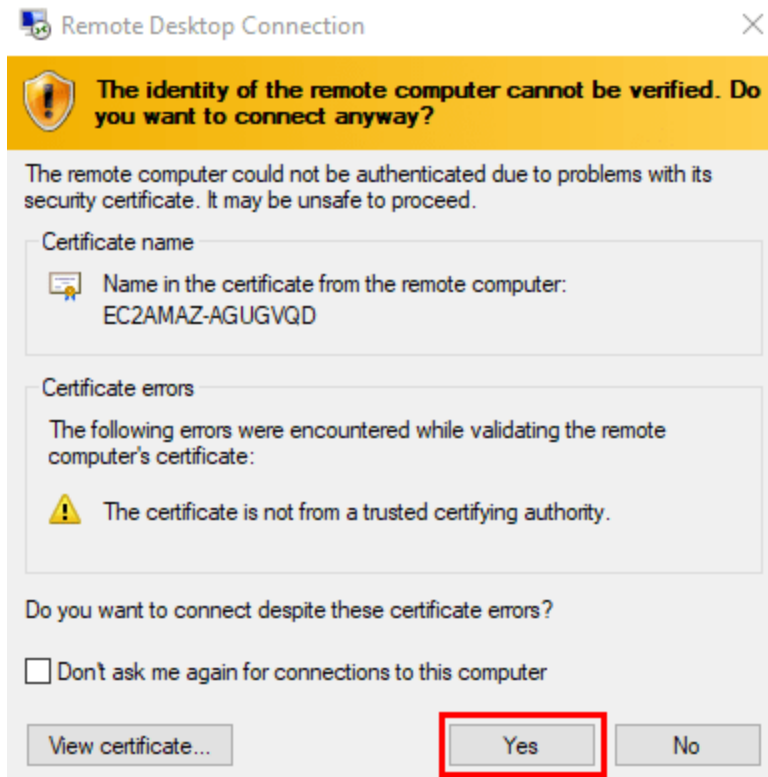


**Step 7:** Here we have to provide the credentials for accessing the Instance so we have to **Enter the Password** That we copied in step 4 and click on Ok.



**Step 9:** Click on Yes.





**Step 10:** So we have successfully connected to an **Amazon Windows Instance**, here we perform all our operations and task that we are performing on the normal windows operating system.

