

PROFESSIONAL TRAINING REPORT
AT
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of Bachelor
of Technology Degree in Information Technology

By

CHARULATHA K
REG. NO. 40120020



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU

APRIL 2023



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

(www.sathyabama.com)

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of CHARULATHA K (Reg. No: 40120020) who carried out the project entitled "GRAPHICAL WEB INTERFACE FOR ACCESSIBILITY OF APPLICATION URL ALERTS" under my supervision from January 2023 to April 2023.

Internal Guide

Dr. G. Mathivanan, M.E., PhD.,

Head of the Department

Dr. R. Subhashini, M.E., Ph.D.,

Submitted for Viva voce Examination held on 28/04/2023

Internal Examiner
External Examiner

DECLARATION

I, **CHARULATHA K** hereby declare that the project report entitled "**GRAPHICAL WEB INTERFACE FOR ACCESSIBILITY OF APPLICATION URL ALERTS**" done by me under the guidance of **Dr. G.Mathivanan, M.E., Ph.D.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Information Technology.

DATE: 28/04/2023

PLACE: CHENNAI

Charulatha.K

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to express my sincere thanks to the **Board of Management** of **Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D., Dean, School of Computing** and **Dr. R. Subhashini M.E., Ph.D., Head of the Department of Information Technology** for providing me necessary support and details during the progress reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.G.Mathivanan, M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE

Ref.No.: 51 /FSPCC/BC001/04/2023



इलेक्ट्रॉनिक्स एवं
सूचना प्रौद्योगिकी मंत्रालय
MINISTRY OF
ELECTRONICS AND
INFORMATION TECHNOLOGY
सत्यमेव जयते



COURSE COMPLETION LETTER

It is certified that

Charulatha K

Sathyabama Institute of Science and Technology

has successfully undergone the Bridge Course Training Program on

Cloud Computing Technology

Under "FutureSkills PRIME" Programme

commenced from 11 Mar, 2023

Shri. V. Sridharan
Chief Investigator
Cloud Computing Technology
Lead Resource Centre

L R PRAKASH Digitally signed by L R PRAKASH
Date: 2023.04.26 01:46:23
+05'30'

Cdr.(Retd.) L.R. Prakash
Senior Director
Centre Head
C-DAC, Chennai

ABSTRACT

This document introduces on how to give an alert notification to the admin of the website when the server goes down or about to crash. This project is based on Amazon Web Service which is called as AWS. By providing on-demand cloud computing at scale, AWS has revolutionized enterprise computing. With codeless architecture, businesses can rapidly build and effectively manage sophisticated, enterprise-grade software in an entirely visual user interface (UI). We often use websites to get information that are required and it has become mandatory in our day to day life. Sometimes a server may not work properly or malfunction due to a load of user who are accessing the server at the same time, so this project enables to give a notification to the admin who's in-charge of the website or the server, so that the admin can easily verify and solve the error without interruptions which will make the users to use the website comfortably.

TABLE OF CONTENTS

Chapter No	TITLE	Page No
1	INTRODUCTION	1
	1.1 Cloud Computing	1
	1.2 Amazon web service	1
	1.2.1 Benefits of cloud computing	2
2	AIM AND SCOPE OF AWS	3
	2.1 Aim	3
	2.2 Scope	3
3	SYSTEM REQUIREMENTS AND METHODS USED	4
	3.1 System requirements	4
	3.1.1 Windows	4
	3.1.2 Mac	4
	3.1.3 Recommended Hardware Requirements	5
	3.1.4 Minimum Hardware Requirements	5
	3.2 Process to create an Instance	6
	3.3 Configure amazon SNS notification for AWS manager	12
4	RESULTS AND PERFORMANCE ANALYSIS	20
	4.1 Working of VM	21

	4.2	Benefits of using AWS VM	22
	4.3	Connecting to windows instance using fleet manager	23
	4.4	Connect to a windows instance using IPV6 address	25
	4.5	Encrypting and decrypting Data keys	26
	4.6	Decrypting the Data keys on AWS	28
	4.7	Accessing Cloud Watch	29
	4.8	Implementation of solution to send in email notification	31
	4.9	Output of EC2 instance with notification alert on email	34
5		FUTURE WORK AND CONCLUSION	36
	5.1	Discussion	36
	5.2	Conclusion	37
6		REFERENCES	38

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
3.2.1	Connecting to Instance	9
3.2.2	Connection Request	10
3.2.3	Permission for access	11
3.3.1	IAM code generation	14
3.3.2	Configuring user access	18
4.1.1	Instance generation	21
4.1.2	Accessing the instance	21
4.3.1	Connecting to RDP client	24
4.4.1	Credential generation	26
4.5.1	Password decryption	27
4.7.1	Instance processing	31
4.9.1	EC2 instance desktop view	34
4.9.2	Email notification	34

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
HTML	HYPERTEXT MARKUP LANGUAGE
UI	USER INTERFACE
OS	OPERATING SYSTEM
IDE	INTEGRATED DEVELOPMENT ENVIRONMENT
AOT	AHEAD OF TIME
JIT	JUST IN TIME
RAM	RANDOM ACCESS MEMORY
VM	VIRTUAL MACHINE
EC2	AMAZON ELASTIC COMPUTE CLOUD
SNS	SIMPLE NOTIFICATION SERVICE
AWS	AMAZON WEB SERVICES
RDP	REMOTE DESKTOP PROTOCOL

CHAPTER – 1

INTRODUCTION

1.1 CLOUD COMPUTING

Cloud computing is a computing technique in which all the data is stored in the cloud. All the SDKs and databases are stored in the cloud only. A Cloud is primarily a platform that allow execution in various forms across multiple resources and potentially across enterprise boundaries. A Cloud is an infrastructure that enables execution of codes (applications, services, etc.) in a managed and elastic fashion, here the meaning of managed is reliability according to pre-defined quality parameters which are automatically ensured and elastic here means that the resources are used according to the current actual requirements. Cloud fits the client-server model, and so far the typical cloud client is the same as the typical enterprise client (i.e. single desktop or laptop computer), some observers have a tendency to stop at this level of analysis. But of course, the real action in the cloud happens on the server side and that's where things get interesting.

1.2 AMAZON WEB SERVICE

The leading cloud provider in the marketplace is Amazon Web Services. It provides over 170 AWS services to the developers so they can access them from anywhere at the time of need. AWS has customers in over 190 countries worldwide including 5000 ed-tech institutions and 2000 government organizations. Many companies ESPN, Adobe, Twitter, Netflix, Facebook, BBC etc., use AWS services. For example, Adobe creates and updates software without depending upon the IT teams. It uses its services by offering multi-terabyte operating environments for its clients. By deploying its services with Amazon services, Adobe integrated and operated its software in a simple manner.

1.2.1 BENEFITS OF CLOUD COMPUTING

Cloud computing has been around for approximately two decades and despite the data pointing to the business efficiencies, cost-benefits, and competitive advantages it holds, a large portion of the business community continues to operate without it. According to a study by the International Data Group, 69% of businesses are already using cloud technology in one capacity or another, and 18% say they plan to implement cloud-computing solutions at some point. At the same time, Dell reports that 11 companies that invest in big data, cloud, mobility, and security enjoy up to 53% faster revenue growth than their competitors. As this data shows, an increasing number of tech-savvy businesses and industry leaders are recognizing the many benefits of the cloud-computing trend. But more than that, they are using this technology to more efficiently run their organizations, better serve their customers, and dramatically increase their overall profit margins.

- Cost Savings
- Security
- Flexibility
- Mobility
- Insight
- Increased Collaboration
- Disaster Recovery
- Loss Prevention
- Automatic Software Updates
- Competitive Edge
- Sustainability

CHAPTER – 2

AIM AND SCOPE OF AWS

2.1 AIM

The aim of the project is to create a Virtual Machine (VM) on an Cloud Based system called AWS (Amazon Web Service). It allows using cloud storage so that it enables us to create a server and initialize an alert while the server of the particular instance goes down. Instances are useful for running memory-intensive workloads such as real-time data ingestion, distributed in-memory caches, big data analytics, memory-intensive enterprise applications, and high-performance databases.

2.2 SCOPE

Amazon EC2 provides the broadest and deepest instance choice to match your workload's needs. General purpose, compute optimized, memory optimized, storage optimized, and accelerated computing instance types are available that provide the optimal compute, memory, storage, and networking balance for your workloads. Processors from Intel, AMD, NVIDIA and AWS power these instance types and provide additional performance and cost optimizations. Local storage and enhanced networking options available with instance types further help optimize performance for workloads that are disk or network I/O bound. Many instance types also offer bare metal instances that provide your applications with direct access to the processor and memory of the underlying server for running in non-virtualized environments or for applications where you want to use your own hypervisor.

CHAPTER – 3

SYSTEM REQUIREMENTS AND METHODS USED

3.1. SYSTEM REQUIREMENTS

The recommended AWS EC2 instance size and software specifications for servers that are required when implementing CyberArk's Privileged Access Manager - Self-Hosted solution. These specifications are based on the entry level industry standard for small-mid range servers. For other implementation sizes, requirements should be customized according to customer needs.

3.1.1 Windows:

- Microsoft® Windows® 7/8/10 (64-bit)
- 16 GB RAM minimum, 8 GB RAM recommended
- 40 GB of available disk space minimum,
- 30 GB Recommended (15 GB for Virtual Machine + 15 GB for Alert Notification, SNS, EC2)
- 1280 x 800 minimum screen resolution
-

3.1.2 Mac:

- Mac® OS X® 10.10 (Yosemite) or higher, up to 10.14 (macOS Mojave)
- 8 GB RAM minimum, 4 GB RAM recommended
- 30 GB of available disk space minimum,
- 30 GB Recommended (15GB for Virtual Machine + 15 GB for Alert Notification,SNS, EC2)
- 1280 x 800 minimum screen resolution

3.1.3 Recommended Hardware Requirements:

The resources that you have available impact your performance. For encoders, the resources determine the speed for encoding assets and the number of streams, bitrate, and type of encoding that's possible. We recommend the following hardware specifications for optimum performance.

AWS Elemental Conductor Live, AWS Elemental Conductor File, AWS Elemental Statmux

- RAM: 16 GB
- Disk space: 500 GB
- CPU cores: 24
- Processor speed: 2.3 GHz or more (Comparable to an Intel® Xeon processor E5-2630)

AWS Elemental Server and AWS Elemental Live

- RAM: 16 GB
- Disk space: 500 GB
- CPU cores: 32
- Processor speed: 2.0 GHz or more (Comparable to an Intel® Xeon processor E5-2650)

AWS Elemental Delta

- RAM: 128 GB
- Disk space: 500 GB
- CPU cores: 24
- Processor speed: 2.3 GHz or more (Comparable to an Intel® Xeon processor E5-2630)

3.1.4 Minimum Hardware Requirements:

You can use host hardware with these minimum resources to run AWS Elemental products for functional testing or for integrating with the AWS Elemental software API. These resource levels are not for performance testing.

All products except AWS Elemental Delta

- RAM: 12 GB
- Disk space: 400 GB
- CPU cores: 8
- RAM: 16 GB
- Disk space: 40 GB
- CPU cores: 8

3.2 PROCESS TO CREATE AN INSTANCE

STEP 1: LAUNCH AN INSTANCE:

You can launch a Windows instance using the AWS Management Console as described in the following procedure. This tutorial is intended to help you quickly launch your first instance, so it doesn't cover all possible options. For information about advanced options, see [Launch an instance using the new launch instance wizard](#). For information about other ways to launch your instance, see [Launch your instance](#).

To launch an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the EC2 console dashboard, in the **Launch instance** box, choose **Launch instance**, and then choose **Launch instance** from the options that appear.
3. Under **Name and tags**, for **Name**, enter a descriptive name for your instance.
4. Under **Application and OS Images (Amazon Machine Image)**, do the following:
 - a. Choose **Quick Start**, and then choose Windows. This is the operating system (OS) for your instance.
 - b. From **Amazon Machine Image (AMI)**, select the AMI for Windows Server 2016 Base or later.. Notice that these AMIs are marked **Free tier eligible**. An *Amazon Machine Image (AMI)* is a basic configuration that serves as a template for your instance.

5. Under **Instance type**, from the **Instance type** list, you can select the hardware configuration for your instance. Choose the t2.micro instance type, which is selected by default. The t2.micro instance type is eligible for the free tier. In Regions where t2.micro is unavailable, you can use a t3.micro instance under the free tier.
6. Under **Key pair (login)**, for **Key pair name**, choose the key pair that you created when getting set up. Note that you must select an **RSA** key. **ED25519** keys are not supported for Windows instances.
7. Next to **Network settings**, choose **Edit**. For **Security group name**, you'll see that the wizard created and selected a security group for you. You can use this security group, or alternatively you can select the security group that you created when getting set up using the following steps:
 - a. Choose **Select existing security group**.
 - b. From **Common security groups**, choose your security group from the list of existing security groups.
8. Keep the default selections for the other configuration settings for your instance.
9. Review a summary of your instance configuration in the **Summary** panel, and when you're ready, choose **Launch instance**.
10. A confirmation page lets you know that your instance is launching. Choose **View all instances** to close the confirmation page and return to the console.
11. On the **Instances** screen, you can view the status of the launch. It takes a short time for an instance to launch. When you launch an instance, its initial state is pending. After the instance starts, its state changes to running and it receives a public DNS name. If the **Public IPv4 DNS** column is hidden, choose the settings icon () in the top-right corner, toggle on **Public IPv4 DNS**, and choose **Confirm**.
12. It can take a few minutes for the instance to be ready for you to connect to it. Check that your instance has passed its status checks; you can view this information in the **Status check** column.

STEP 2: CONNECT TO YOUR INSTANCE

To connect to a Windows instance, you must retrieve the initial administrator password and then enter this password when you connect to your instance using Remote Desktop. It takes a few minutes after instance launch before this password is available.

The name of the administrator account depends on the language of the operating system. For example, for English, it's Administrator, for French it's Administrateur. For more information, see Localized Names for Administrator Account in Windows in the Microsoft TechNet Wiki.

If you've joined your instance to a domain, you can connect to your instance using domain credentials you've defined in AWS Directory Service. On the Remote Desktop login screen, instead of using the local computer name and the generated password, use the fully-qualified user name for the administrator (for example, **corp.example.com\Admin**), and the password for this account.

If you receive an error while attempting to connect to your instance, see Remote Desktop can't connect to the remote computer.

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, select **Instances**. Select the instance and then choose **Connect**.
3. On the **Connect to instance** page, choose the **RDP client** tab, and then choose **Get password**.

Connect to instance [Info](#)
Connect to your instance i- (Source-NET-Webserver) using any of these options

Session Manager | **RDP client** | **EC2 serial console**

Instance ID
i- (Source-NET-Webserver)

Connection Type

☒ **Connect using RDP client**
Download a file to use with your RDP client and retrieve your password.

☐ **Connect using Fleet Manager**
Connect to your instance using Fleet Manager Remote Desktop.

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- .us-west-2.compute.amazonaws.com

User name
Administrator

Password [Get password](#)

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

[Cancel](#)

Fig 3.2.1 Connecting to instance

4. Choose **Browse** and navigate to the private key (.pem) file you created when you launched the instance. Select the file and choose **Open** to copy the entire contents of the file to this window.
5. Choose **Decrypt Password**. The console displays the default administrator password for the instance under **Password**, replacing the **Get password** link shown previously. Save the password in a safe place. This password is required to connect to the instance.
1. Choose **Download remote desktop file**. Your browser prompts you to either open or save the RDP shortcut file. When you have finished downloading the file, choose **Cancel** to return to the **Instances** page.
 - If you opened the RDP file, you'll see the **Remote Desktop Connection** dialog box.
 - If you saved the RDP file, navigate to your downloads directory, and open the RDP file to display the dialog box.

2. You may get a warning that the publisher of the remote connection is unknown. Choose **Connect** to continue to connect to your instance.

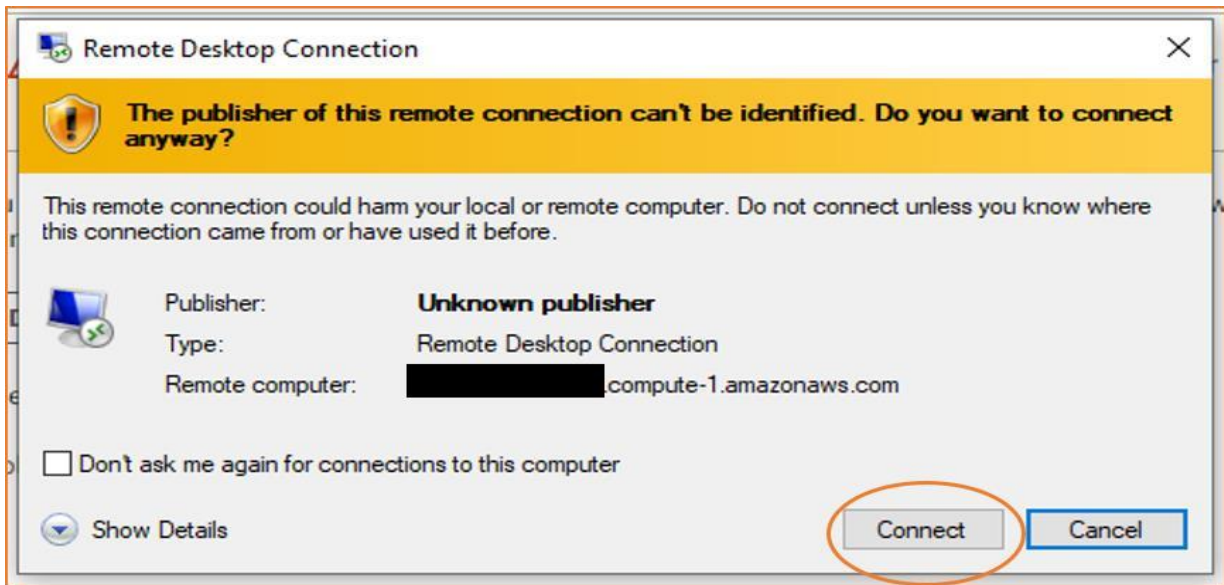


Fig 3.2.2 Connection request

3. The administrator account is chosen by default. Copy and paste the password that you saved previously.

Tip

If you receive a "Password Failed" error, try entering the password manually. Copying and pasting content can corrupt it.

4. Due to the nature of self-signed certificates, you may get a warning that the security certificate could not be authenticated. Use the following steps to verify the identity of the remote computer, or simply choose **Yes** (Windows) or **Continue** (Mac OS X) if you trust the certificate.

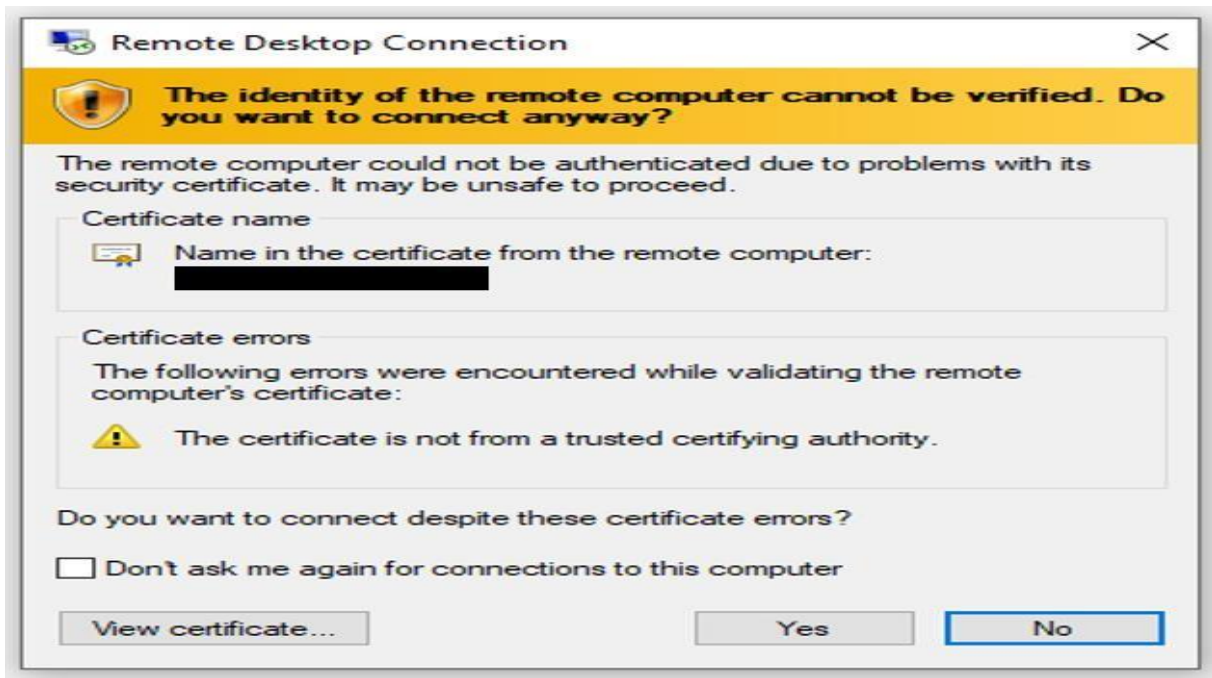


Fig 3.2.3 Permission for access

- If you are using **Remote Desktop Connection** on a Windows computer, choose **View certificate**. If you are using **Microsoft Remote Desktop** on a Mac, choose **Show Certificate**.
- Choose the **Details** tab, and scroll down to **Thumbprint** (Windows) or **SHA1 Fingerprints** (Mac OS X). This is the unique identifier for the remote computer's security certificate.
- In the Amazon EC2 console, select the instance, choose **Actions, Monitor and troubleshoot, Get system log**.
- In the system log output, look for RDPCERTIFICATE-THUMBPRINT. If this value matches the thumbprint or fingerprint of the certificate, you have verified the identity of the remote computer.
- If you are using **Remote Desktop Connection** on a Windows computer, return to the **Certificate** dialog box and choose **OK**. If you are using **Microsoft Remote Desktop** on a Mac, return to the **Verify Certificate** and choose **Continue**.

- [Windows] Choose **Yes** in the **Remote Desktop Connection** window to connect to your instance.

[Mac OS X] Log in as prompted, using the default administrator account and the default administrator password that you recorded or copied previously. Note that you might need to switch spaces to see the login screen.

STEP 3: CLEAN UP YOUR INSTANCE

After you've finished with the instance that you created for this tutorial, you should clean up by terminating the instance. If you want to do more with this instance before you clean up. If you launched an instance that is not within the AWS Free Tier, you'll stop incurring charges for that instance as soon as the instance status changes to shutting down or terminated. To keep your instance for later, but not incur charges, you can stop the instance now and then start it again later. For more information, see [Stop and start your instance](#).

STEP 4: TO TERMINATE THE INSTANCE

To terminate your instance

1. In the navigation pane, choose **Instances**. In the list of instances, select the instance.
2. Choose **Instance state, Terminate instance**.
3. Choose **Terminate** when prompted for confirmation.

Amazon EC2 shuts down and terminates your instance. After your instance is terminated, it remains visible on the console for a short while, and then the entry is automatically deleted. You cannot remove the terminated instance from the console display yourself.

3.3 CONFIGURE AMAZON SNS NOTIFICATION FOR AWS SYSTEM MANAGER AND VM

STEP 1: CREATE AND SUBSCRIBE TO AMAZON SNS TOPIC FROM AWS

An Amazon SNS topic is a communication channel that Run Command and Run Command tasks that are registered to a maintenance window use to send notifications

about the status of your commands. Amazon SNS supports different communication protocols, including HTTP/S, email, and other AWS services like Amazon Simple Queue Service (Amazon SQS). To get started, we recommend that you start with the email protocol. For information about how to create a topic, see *Creating an Amazon SNS topic* in the *Amazon Simple Notification Service Developer Guide*.

After you create the topic, subscribe to it by specifying an Endpoint. If you chose the Email protocol, the endpoint is the email address where you want to receive notifications. For more information about how to subscribe to a topic, see *Subscribing to an Amazon SNS topic* in the *Amazon Simple Notification Service Developer Guide*.

Amazon SNS sends a confirmation email from *AWS Notifications* to the email address that you specify. Open the email and choose the Confirm subscription link.

You will receive an acknowledgement message from AWS. Amazon SNS is now configured to receive notifications and send the notification as an email to the email address that you specified.

STEP 2: CREATE AN IAM POLICY FOR AMAZON SNS NOTIFICATION

Use the following procedure to create a custom AWS Identity and Access Management (IAM) policy that provides permissions for initiating Amazon SNS notifications.

To create a custom IAM policy for Amazon SNS notifications

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose Policies, and then choose Create Policy. (If a Get Started button is shown, choose it, and then choose Create Policy.)
3. Choose the JSON tab.
4. Replace the default content with the following.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sns:Publish"
      ],
      "Resource": "arn:aws:sns:region:account-id:sns-topic-name"
    }
  ]
}

```

Fig 3.3.1 IAM code generation

1. *region* represents the identifier for an AWS Region supported by AWS Systems Manager, such as *us-east-2* for the US East (Ohio) Region. For a list of supported *region* values, see the **Region** column in Systems Manager service endpoints in the *Amazon Web Services General Reference*.

account-id represents the 12-digit identifier for your AWS account, in the format 123456789012.

sns-topic-name represents the name of the Amazon SNS topic you want to use for publishing notifications.

2. Choose **Next: Tags**.
3. (Optional) Add one or more tag-key value pairs to organize, track, or control access for this policy.
4. Choose **Next: Review**.
5. On the **Review policy** page, for **Name**, enter a name for the inline policy. For example: **my-sns-publish-permissions**.
6. (Optional) For **Description**, enter a description for the policy.
7. Choose **Create policy**.

STEP 3: CREATE AN IAM ROLE FOR AMAZON SNS NOTIFICATION

Use the following procedure to create an IAM role for Amazon SNS notifications. This service role is used by Systems Manager to initiate Amazon SNS notifications. In all subsequent procedures, this role is referred to as the Amazon SNS IAM role.

To create an IAM service role for Amazon SNS notifications

1. Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane of the IAM console, choose **Roles**, and then choose **Create role**.
3. Choose the **AWS service** role type, and then choose Systems Manager.
4. Choose the Systems Manager use case. Then, choose **Next**.
5. On the **Attach permissions policies** page, select the box to the left of the name of the custom policy you created in Task 2. For example: **my-sns-publish-permissions**.
6. (Optional) Set a permissions boundary. This is an advanced feature that is available for service roles, but not service-linked roles.

Expand the **Permissions boundary** section and choose **Use a permissions boundary to control the maximum role permissions**. IAM includes a list of the AWS managed and customer managed policies in your account. Select the policy to use for the permissions boundary or choose **Create policy** to open a new browser tab and create a new policy from scratch. For more information, see *Creating IAM policies* in the *IAM User Guide*. After you create the policy, close that tab and return to your original tab to select the policy to use for the permissions boundary.

7. Choose **Next**.
8. If possible, enter a role name or role name suffix to help you identify the purpose of this role. Role names must be unique within your AWS account. They are not distinguished by case. For example, you cannot create roles named both **PRODROLE** and **prodrole**. Because various entities might reference the role, you cannot edit the name of the role after it has been created.

9. (Optional) For **Description**, enter a description for the new role.
10. Choose **Edit** in the **Step 1: Select trusted entities** or **Step 2: Select permissions** sections to edit the use cases and permissions for the role.
11. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM resources in the *IAM User Guide*.
12. Review the role and then choose **Create role**.
13. Choose the name of the role, and then copy or make a note of the **Role ARN** value. This Amazon Resource Name (ARN) for the role is used when you send a command that is configured to return Amazon SNS notifications.
14. Keep the **Summary** page open.

STEP 4: CONFIGURE USER ACCESS

If an IAM entity (user, role, or group) is assigned administrator permissions, then the user or role has access to Run Command and Maintenance Windows, capabilities of AWS Systems Manager.

For entities without administrator permissions, an administrator must grant the following permissions to the IAM entity:

- The Amazon SSM Full Access managed policy, or a policy that provides comparable permissions.

To provide access, add permissions to your users, groups, or roles:

- Users and groups in AWS IAM Identity Center (successor to AWS Single Sign-On):
Create a permission set. Follow the instructions in Create a permission set in the *AWS IAM Identity Center (successor to AWS Single Sign-On) User Guide*.
- Users managed in IAM through an identity provider:
Create a role for identity federation. Follow the instructions in Creating a role for a third-party identity provider (federation) in the *IAM User Guide*.

- IAM users:
 - Create a role that your user can assume. Follow the instructions in *Creating a role for an IAM user in the IAM User Guide*.
 - (Not recommended) Attach a policy directly to a user or add a user to a user group. Follow the instructions in *Adding permissions to a user (console)* in the *IAM User Guide*.

To configure user access and attach the iam:PassRole policy to a user account

1. In the IAM navigation pane, choose **Users**, and then choose the user account that you want to configure.
2. On the **Permissions** tab, in the policies list, verify that either the **Amazon SSM Full Access** policy is listed or that there is a comparable policy that gives the account permissions to access Systems Manager.
3. Choose **Add inline policy**.
4. On the **Create policy** page, choose the **Visual editor** tab.
5. Choose a service, and then choose **IAM**.
6. For **Actions**, in the **Filter actions** text box, enter **PassRole**, and then select the check box next to **PassRole**.
7. For **Resources**, verify that **Specific** is selected, and then choose **Add ARN**.
8. In the **Specify ARN for role** field, paste the Amazon SNS IAM role ARN that you copied at the end of Task 3. The system automatically populates the **Account** and **Role name with path** fields.
9. Choose **Add**.
10. Choose **Review policy**.
11. On the **Review Policy** page, enter a name and then choose **Create policy**.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::account-id:role/sns-role-name"
    }
  ]
}

```

Fig 3.3.2 Configuring user access

STEP 5: ATTACH THE IAM: PASSROLE POLICY TO THE MAINTENANCE WINDOW ROLE

When you register a Run Command task with a maintenance window, you specify a service role Amazon Resource Name (ARN). This service role is used by Systems Manager to run tasks registered to the maintenance window. To configure Amazon SNS notifications for a registered Run Command task, attach an iam:PassRole policy to the maintenance window service role specified. If you don't intend to configure the registered task for Amazon SNS notifications, then you can skip this task.

The iam:PassRole policy allows the Maintenance Windows service role to pass the Amazon SNS IAM role created in Task 3 to the Amazon SNS service. The following procedure shows how to attach the iam:PassRole policy to the Maintenance Windows service role.

To attach the iam:PassRole policy to your Maintenance Windows role

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Roles** and select the Amazon SNS IAM role created in Task 3.

3. Copy or make a note of the **Role ARN** and return to the **Roles** section of the IAM console.
4. Select the custom Maintenance Windows service role you created from the **Role name** list.
5. On the **Permissions** tab, verify that either the Amazon SSM Maintenance Window Role policy is listed or there is a comparable policy that gives maintenance windows permission to the Systems Manager API. If it is not, choose **Add permissions, Attach policies** to attach it.
6. Choose **Add permissions, Create inline policy**.
7. Choose the **Visual editor** tab.
8. For **Service**, choose **IAM**.
9. For **Actions**, in the **Filter actions** text box, enter **PassRole**, and then select the check box next to **PassRole**.
10. For **Resources**, choose **Specific**, and then choose **Add ARN**.
11. In the **Specify ARN for role** box, paste the ARN of the Amazon SNS IAM role created in Task 3, and then choose **Add**.
12. Choose **Review policy**.

CHAPTER – 4

RESULTS AND PERFORMANCE ANALYSIS

4.1 WORKING OF VM

A **virtual machine** (VM) is a software demonstration and development model that simulates an operating system and its applications. VMs create a software simulation environment for testing software and application programs without physical hardware investment.

An Amazon Web Services (AWS) Virtual Machine (EC2) is a type of service offered by AWS that allows users to rent virtualized computing resources, such as virtual CPUs and memory, by the hour. These resources can be used to run a wide range of applications, including web servers, databases, and big data processing jobs.

Users have the ability to configure and customize the operating system, storage, and networking resources of their EC2 instances to meet the specific requirements of their applications. EC2 is one of the core services of AWS and is a part of the Elastic Compute Cloud (EC2) service.

To get a better understanding of this booming resource, let's explore what an AWS virtual machine is, how it's created and managed, and why it's so common in information technology (IT).

A virtual machine mimics the different aspects of an operating system or application program with guest operating systems hosted on it. These guest systems can be different architectures (e.g., x86 host and ARM guest) or different versions (e.g., 32-bit Windows XP, hosted within 64-bit Windows 7).

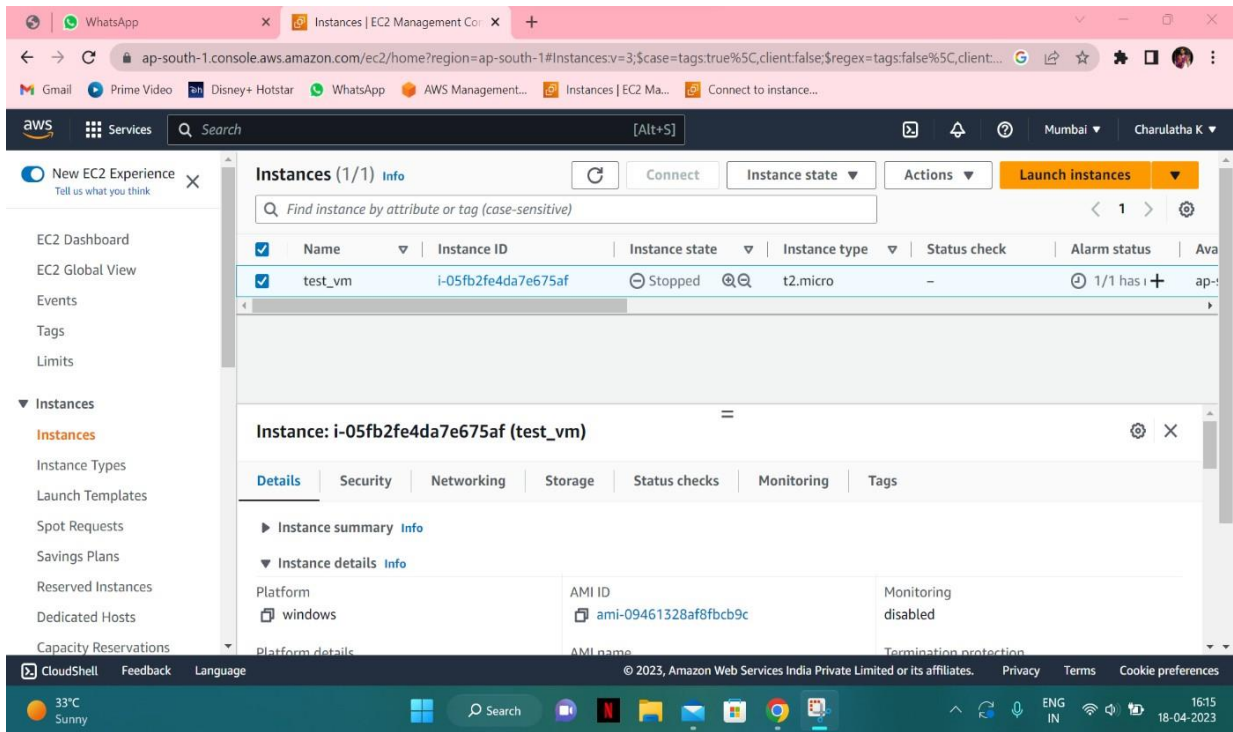


Fig 4.1.1 Instance generation

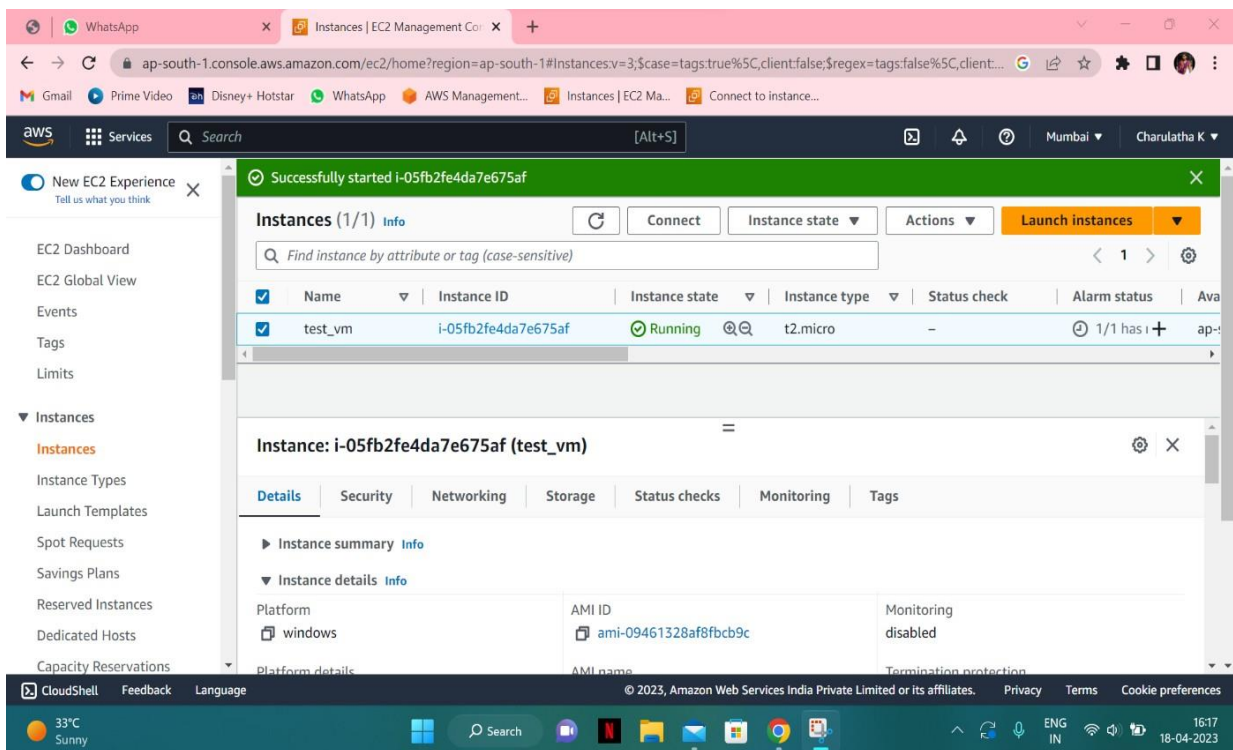


Fig 4.1.2 Accessing the instance

4.2 BENEFITS OF USING AWS VM

A virtual machine is like any other computer that operates on a central processor and runs guest operating systems. It provides resources beyond a single physical server, such as thousands of servers in terms of CPU power and network bandwidth. VMs are used in cloud computing applications because they allow easier scaling.

- Isolation: Each VM runs in its own isolated environment, which can improve security and prevent conflicts between different applications and operating systems.
- Data safety and security: VMs run in a secure environment and reduce data loss risk. The data is isolated from the host operating system, so it can be copied to another location without spreading across the network.
- Simplified maintenance: A virtualization software enables a system administrator to configure a VM's operating system, allocate memory and storage, install applications, and connect to networks.
- Remote access: An administrator can access guest operating systems from a central location using graphical management tools and standard network protocols, such as HTTP/HTTPS or RDP/VNC. You can also connect to any VM remotely via a secure shell (SSH) connection. This remote connection provides an easy way for remote support.
- Flexible resource allocation: A VM can be configured for optimal resource utilization like CPU power and storage space by creating multiple virtual machines within a single physical server or server cluster.
- Portability: VMs can be easily moved or copied to different hosts, which can make it easier to scale up or down as needed.

Virtual machines are widely used in many enterprise environments, cloud infrastructure, software development and testing, and other areas where IT resources need to be shared and managed in a flexible and efficient way.

VMs are virtualized computing resources' fundamental components that play a key role in creating applications, tools, or environments, both online and on-premises. Here are some use cases for VMs:

- **Create environments for development and testing:** Virtual machines (VMs) create isolated environments for testing and development, with full functionality and without affecting the surrounding infrastructure.
- **Consolidate servers:** Organizations use virtual machines to host additional VMs to reduce sprawl by consolidating more resources into a single physical system.
- **Create a hybrid configuration:** Virtual machines (VMs) serve as the foundation for building a cloud environment with an on-premises one, enabling greater flexibility without sacrificing old systems.
- **Enhance business continuity and incident recovery:** Using virtual machines to replicate systems in cloud settings creates additional protection and trust. Cloud environments can also be updated regularly with VMs.

Many cloud service platforms offer VM trials, but few provide them free of cost. AWS (Amazon Web Service) is among the popular cloud solutions as it allows hands-on user experience with free cloud virtualization platforms, services, and solutions. It also provides users access to over 60 features to develop their own cloud VM.

4.3 CONNECTING TO WINDOWS INSTANCE USING FLEET MANAGER

You can use Fleet Manager, a capability of AWS Systems Manager, to connect to Windows instances using the Remote Desktop Protocol (RDP) and display up to four Windows instances on the same page in the AWS Management Console. You can connect to the first instance in the Fleet Manager Remote Desktop directly from the **Instances** page in the Amazon EC2 console as follows.

To connect to instances using RDP with Fleet Manager (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation pane, choose **Instances**.
3. Select the instance and then choose **Connect**.
4. On the **Connect to instance** page, choose the option to **Connect using Fleet Manager**, then choose **Fleet Manager Remote Desktop**. This opens the **Fleet Manager Remote Desktop** page in the AWS Systems Manager console.

Decrypts ciphertext and then reencrypts it entirely within AWS KMS. You can use this operation to change the KMS key under which data is encrypted, such as when you manually rotate a KMS key or change the KMS key that protects a ciphertext. You can also use it to reencrypt ciphertext under the same KMS key, such as to change the encryption context of a ciphertext.

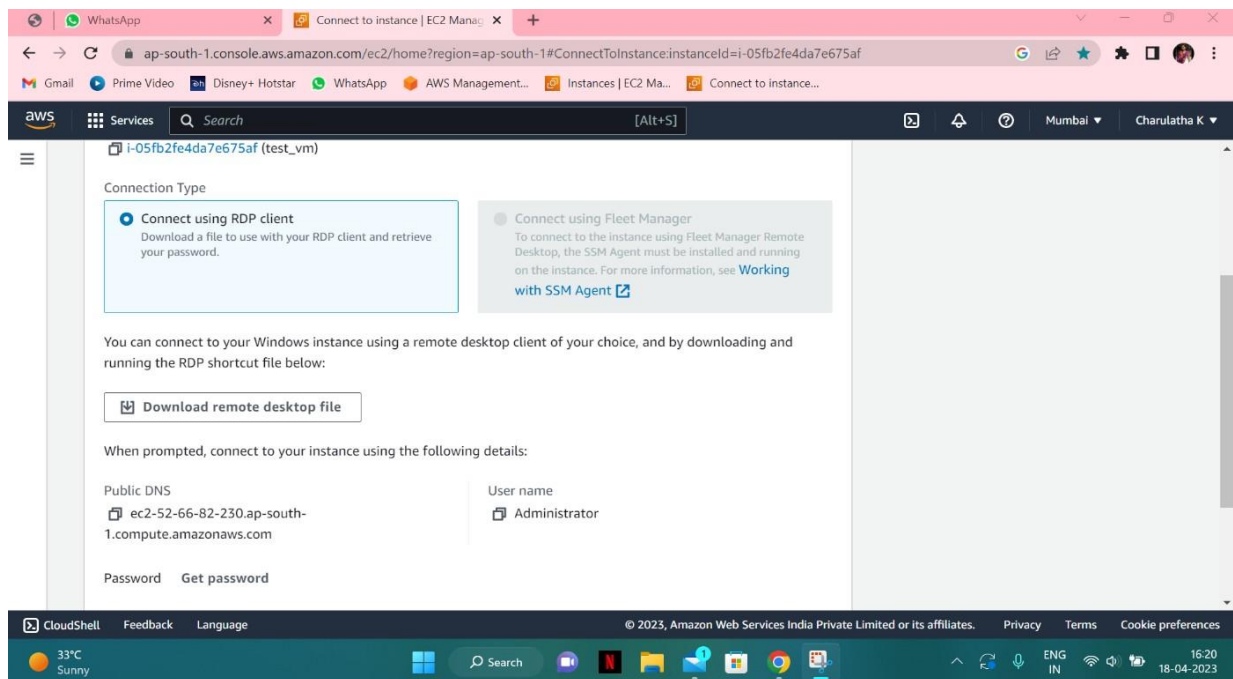


Fig 4.3.1 Connecting to RDP client

4.4 CONNECT TO A WINDOWS INSTANCE USING IPV6 ADDRESS

If you've enabled your VPC for IPv6 and assigned an IPv6 address to your Windows instance, you can use an RDP client to connect to your instance using its IPv6 address (for example, 2001:db8:1234:1a00:9691:9503:25ad:1761) instead of using its public IPv4 address or public DNS hostname.

To connect to your Windows instance using its IPv6 address

1. Get the initial administrator password for your instance, as described in [Connect to your Windows instance using RDP](#). This password is required to connect to your instance.
2. [Windows] Open the RDP client on your Windows computer, choose **Show Options**, and do the following:
 - For **Computer**, enter the IPv6 address of your Windows instance.
 - For **User name**, enter **Administrator**.
 - Choose **Connect**.
 - When prompted, enter the password that you saved previously.

[Mac OS X] Open the RDP client on your computer and do the following:

- Choose **New**.
- For **PC Name**, enter the IPv6 address of your Windows instance.
- For **User name**, enter **Administrator**.
- Close the dialog box. Under **My Desktops**, select the connection, and choose **Start**.
- When prompted, enter the password that you saved previously.

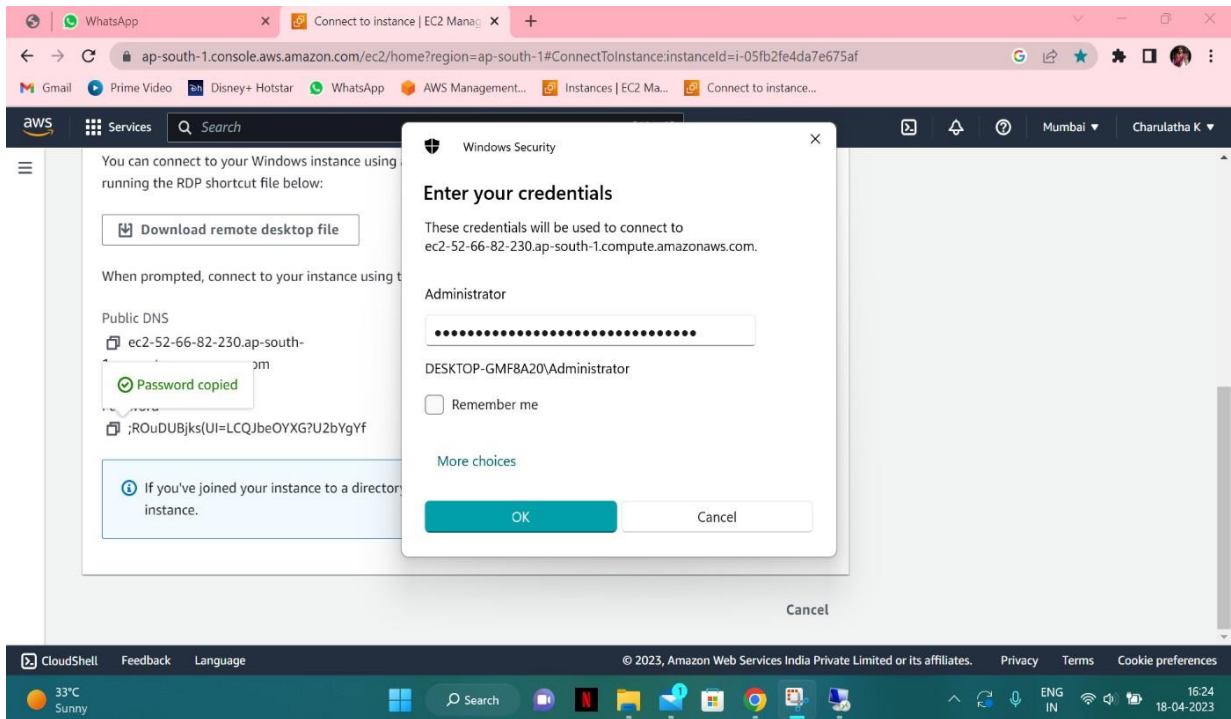


Fig 4.4.1 Credential generation

Due to the nature of self-signed certificates, you may get a warning that the security certificate could not be authenticated. If you trust the certificate, you can choose **Yes** or **Continue**. Otherwise, you can verify the identity of the remote computer, as described in [Connect to your Windows instance using RDP](#).

4.5 ENCRYPTING AND DECRYPTING DATA KEYS

The examples in this topic use the `Encrypt`, `Decrypt`, and `ReEncrypt` operations in the AWS KMS API.

These operations are designed to encrypt and decrypt data keys. They use an AWS KMS key in the encryption operations and they cannot accept more than 4 KB (4096 bytes) of data. Although you might use them to encrypt small amounts of data, such as a password or RSA key, they are not designed to encrypt application data.

To encrypt application data, use the server-side encryption features of an AWS service, or a client-side encryption library, such as the AWS Encryption SDK or the Amazon S3 encryption client.

The Encrypt operation is designed to encrypt data keys, but it is not frequently used. The GenerateDataKey and GenerateDataKeyWithoutPlaintext operations return encrypted data keys. You might use this method when you are moving encrypted data to a different Region and want to encrypt its data key with a KMS key in the new Region.

In languages that require a client object, these examples use the AWS KMS client object that you created in Creating a client.

To decrypt a data key, use the Decrypt operation.

The ciphertextBlob that you specify must be the value of the CiphertextBlob field from a GenerateDataKey, GenerateDataKeyWithoutPlaintext, or Encrypt response, or the PrivateKeyCiphertextBlob field from a GenerateDataKeyPair or GenerateDataKeyPairWithoutPlaintext response. You can also use the Decrypt operation to decrypt data encrypted outside of AWS KMS by the public key in an asymmetric KMS key.

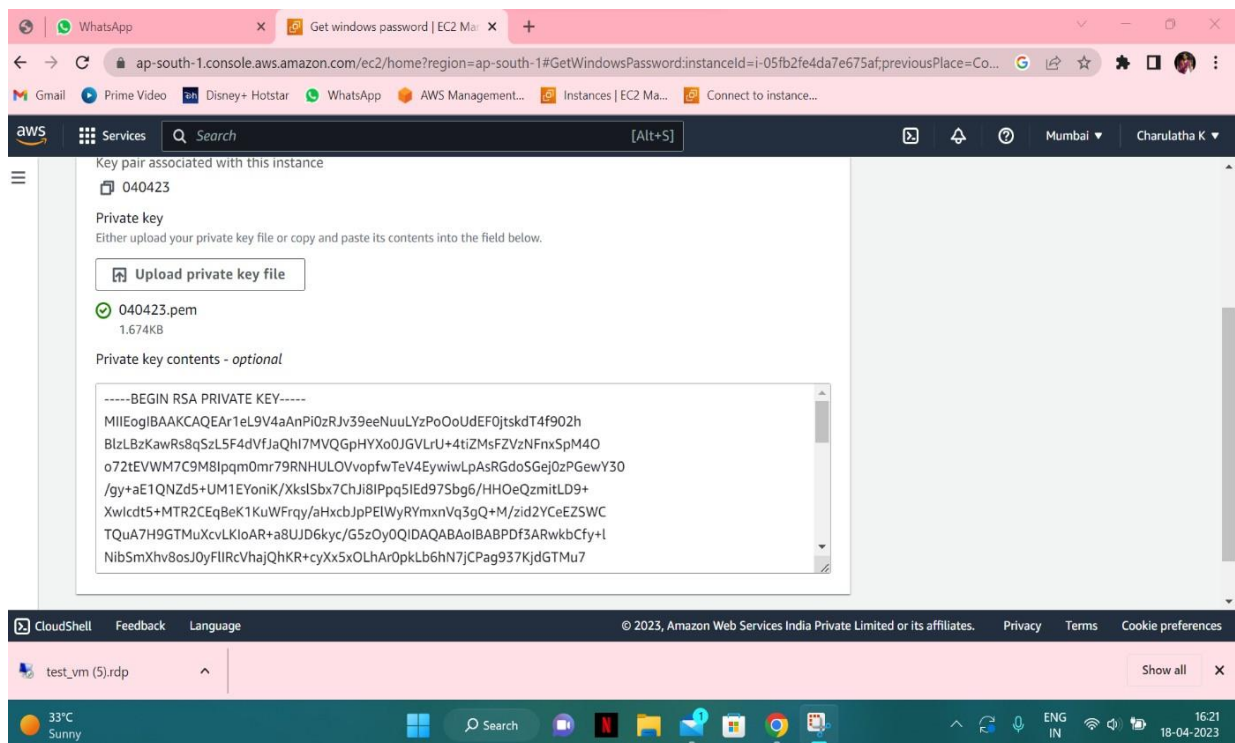


Fig 4.5.1 Password decryption

The `KeyId` parameter is not required when decrypting with symmetric encryption KMS keys. AWS KMS can get the KMS key that was used to encrypt the data from the metadata in the ciphertext blob. But it's always a best practice to specify the KMS key you are using. This practice ensures that you use the intended KMS key, and prevents you from inadvertently decrypting a ciphertext using a KMS key you do not trust.

In languages that require a client object, these examples use the AWS KMS client object that you created in [Creating a client](#).

4.6 RECRYPTING THE DATA KEY ON AWS

To decrypt an encrypted data key, and then immediately re-encrypt the data key under a different AWS KMS key, use the `ReEncrypt` operation. The operations are performed entirely on the server side within AWS KMS, so they never expose your plaintext outside of AWS KMS.

The `ciphertextBlob` that you specify must be the value of the `CiphertextBlob` field from a `GenerateDataKey`, `GenerateDataKeyWithoutPlaintext`, or `Encrypt` response, or the `Private Key Ciphertext Blob` field from a `Generate Data Key Pair` or `Generate DataKey Pair Without Plaintext` response. You can also use the `ReEncrypt` operation to re-encrypt data encrypted outside of AWS KMS by the public key in an asymmetric KMS key.

The `SourceKeyId` parameter is not required when re-encrypting with symmetric encryption KMS keys. AWS KMS can get the KMS key that was used to encrypt the data from the metadata in the ciphertext blob. But it's always a best practice to specify the KMS key you are using. This practice ensures that you use the intended KMS key, and prevents you from inadvertently decrypting a ciphertext using a KMS key you do not trust.

In languages that require a client object, these examples use the AWS KMS client object that you created in [Creating a client](#).

4.7 ACCESSING CLOUD WATCH

CloudWatch is a monitoring service offered by Amazon Web Services (AWS). It provides real-time monitoring and visibility into resources such as EC2 instances, EBS volumes, and RDS databases, as well as custom metrics and logs. In this guide, we will walk you through the steps to access CloudWatch on AWS.

Step 1: Sign in to AWS Console

The first step is to sign in to the AWS Management Console. If you don't have an account, you can sign up for a free tier account on the AWS website.

Step 2: Navigate to CloudWatch

Once you have signed in to the AWS Management Console, navigate to the CloudWatch service. You can either search for it in the search bar or find it under the 'Services' dropdown menu.

Step 3: Create a new dashboard (optional)

If you want to create a new dashboard, click on the 'Create Dashboard' button. Give it a name and select the type of widget you want to add to it, such as a line graph or a table.

Step 4: Add widgets to the dashboard (optional)

Once you have created a new dashboard, you can add widgets to it. Widgets are visual representations of metrics that you want to monitor. To add a widget, click on the 'Add widget' button, select the type of widget you want to add, and configure its settings.

Step 5: View metrics

To view metrics for a particular resource, click on the 'Metrics' option in the left-hand menu. You can select the resource you want to monitor from the dropdown menu and select the metrics you want to monitor.

Step 6: View logs

If you want to view logs, click on the 'Logs' option in the left-hand menu. You can select the log group you want to view logs for, and then select the log stream you want to view logs for.

Step 7: Set up alarms

CloudWatch allows you to set up alarms that notify you when a metric breaches a threshold. To set up an alarm, click on the 'Alarms' option in the left-hand menu, select the metric you want to monitor, and configure the alarm settings.

Step 8: Use CloudWatch CLI

You can also use the CloudWatch Command Line Interface (CLI) to access CloudWatch. The CLI allows you to access CloudWatch from the command line and automate monitoring tasks. To use the CLI, you will need to install it on your local machine and configure it with your AWS credentials.

Step 9: Use CloudWatch API

Finally, you can use the CloudWatch API to access CloudWatch programmatically. The API allows you to integrate CloudWatch into your own applications and automate monitoring tasks. To use the API, you will need to have programming skills and knowledge of AWS services.

In conclusion, accessing CloudWatch on AWS is essential for monitoring and maintaining the health of your resources. By following the above steps, you can access CloudWatch, create dashboards, view metrics and logs, set up alarms, and even use the CloudWatch CLI and API.

4.8 IMPLEMENTATION OF A SOLUTION TO SEND IN EMAIL NOTIFICATION

This implementation is based on Python and involves several AWS services, including Amazon SNS, Amazon CloudWatch, and Amazon EC2.

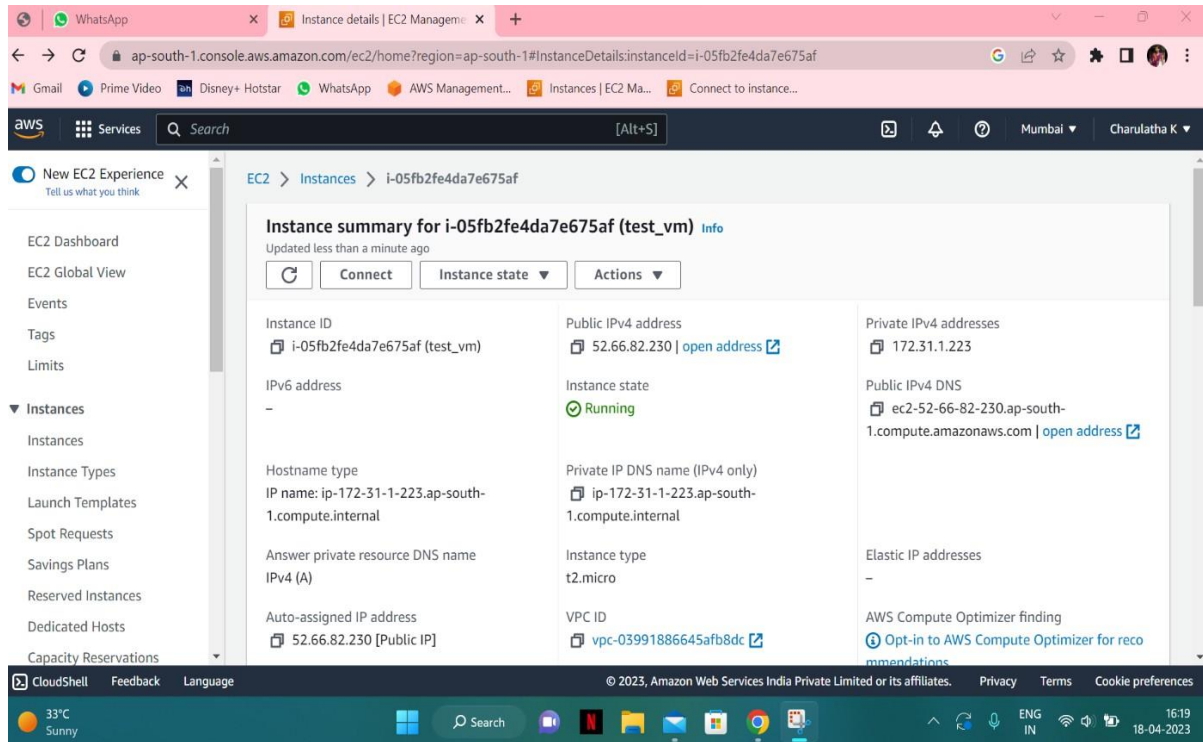


Fig 4.7.1 Instance processing

Prerequisites:

- An AWS account with permissions to create and manage EC2 instances, SNS topics, and CloudWatch alarms.
- A running EC2 instance to monitor.

Step 1: Create an SNS topic

Amazon SNS (Simple Notification Service) is a messaging service that allows you to send notifications to subscribers or endpoints. We will use SNS to send email notifications when the EC2 instance goes down.

1.1. Go to the AWS Management Console and navigate to the SNS service.

- 1.2. Click on the "Create topic" button and enter a name for the topic, such as "EC2-Down-Alert".
- 1.3. Once the topic is created, click on the "Create subscription" button and select "Email" as the protocol.
- 1.4. Enter the email address where you want to receive the notifications and click on the "Create subscription" button.
- 1.5. AWS will send a confirmation email to the email address you provided. Follow the instructions in the email to confirm the subscription.

Step 2: Create a CloudWatch alarm

Amazon CloudWatch is a monitoring service that allows you to monitor your applications, resources, and services on AWS. We will use CloudWatch to monitor the status of the EC2 instance and trigger an SNS notification when it goes down.

- 2.1. Go to the AWS Management Console and navigate to the CloudWatch service.
- 2.2. Click on the "Alarms" option in the left-hand menu and then click on the "Create alarm" button.
- 2.3. In the "Create Alarm" wizard, select the "EC2" namespace and choose the "Per-Instance Metrics" option.
- 2.4. Select the EC2 instance you want to monitor and choose the "StatusCheckFailed_Instance" metric.
- 2.5. Set the threshold for the alarm to "1" and choose "Whenever this alarm state is ALARM".
- 2.6. In the "Actions" section, choose "Add notification" and select the SNS topic you created in Step 1.
- 2.7. Complete the wizard and save the alarm.

Step 3: Install and configure the CloudWatch agent

To monitor the EC2 instance with CloudWatch, we need to install and configure the CloudWatch agent on the EC2 instance.

- 3.1. SSH into the EC2 instance you want to monitor.
- 3.2. Download and install the CloudWatch agent by following the instructions [CloudWatch-Agent-on-EC2-Instance.html](#)
- 3.3. Configure the CloudWatch agent by creating a configuration file in the `"/opt/aws/amazon-cloudwatch-agent/etc"` directory. Here's an example configuration file:

```
{
  "metrics": {
    "append_dimensions": {
      "InstanceId": "${aws:InstanceId}"
    },
    "metrics_collected": {
      "mem": {
        "measurement": [
          "mem_used_percent"
        ]
      },
      "swap": {
        "measurement": [
          "swap_used_percent"
        ]
      },
      "disk": {
        "
```

To permit reencryption from or to a KMS key, include the `"kms:ReEncrypt*"` permission in your key policy. This permission is automatically included in the key policy when you use the console to create a KMS key. But you must include it manually when you create a KMS key programmatically or when you use the `PutKeyPolicy` operation to set a key policy.

4.9 OUTPUT OF EC2 INSTANCE WITH NOTIFICATION ALERT ON EMAIL

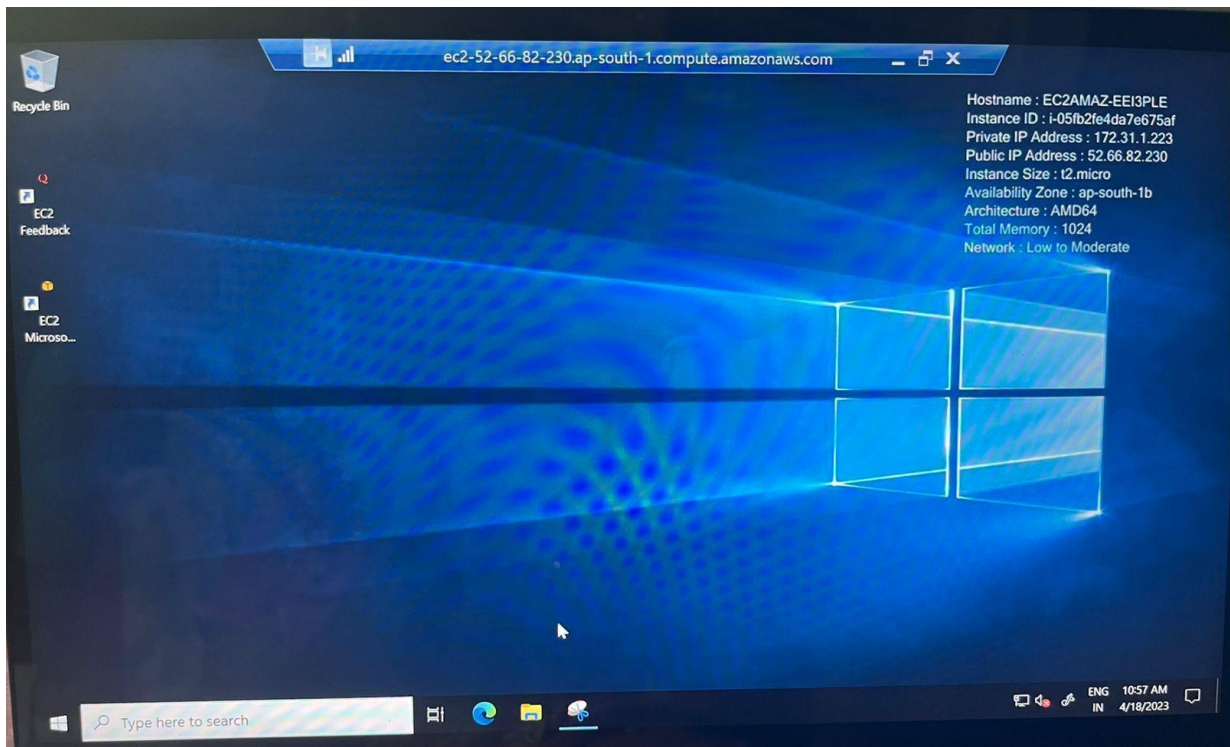


Fig 4.9.1 EC2 instance desktop view

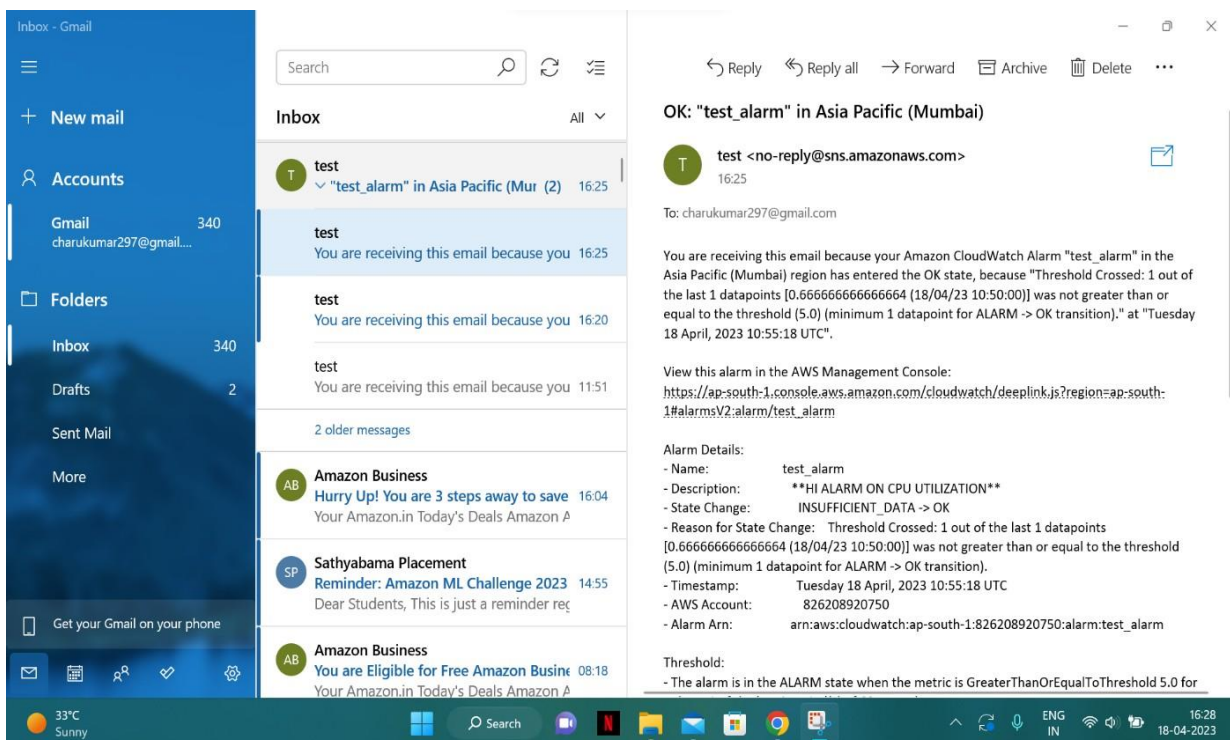


Fig 4.9.2 Email notification

The server pushes the notification on every 5 minutes of interval to the authorized admin. Push notification services, such as APNs and FCM, maintain a connection with each app and associated mobile device registered to use their service. When an app and mobile device register, the push notification service returns a device token. Amazon SNS uses the device token to create a mobile endpoint, to which it can send direct push notification messages. In order for Amazon SNS to communicate with the different push notification.

CHAPTER – 5

FUTURE WORK AND CONCLUSION

5.1 FUTURE WORK

Amazon Web Services (AWS) provides a variety of services to help businesses manage their cloud infrastructure. One such service is the ability to send notifications via email from an Elastic Compute Cloud (EC2) instance. This can be useful for a variety of purposes, such as sending alerts when a certain threshold is exceeded, or notifying administrators when a critical process has completed.

There are a few best practices to keep in mind when setting up email notifications from an EC2 instance. First, it's important to ensure that your SES account is properly secured. This means using strong passwords and enabling multi-factor authentication (MFA) to prevent unauthorized access. Additionally, you should use SSL/TLS encryption when sending emails to ensure that they are not intercepted or tampered with.

Another best practice is to limit the number of emails that are sent from your EC2 instance. This can help prevent your SES account from being flagged as a spammer, which can result in your emails being blocked by recipient email servers. To limit the number of emails sent, you can use a rate limiter or a queueing system to ensure that emails are sent at a controlled pace.

Overall, setting up email notifications from an EC2 instance using AWS SES is a straightforward process that can be extremely useful for a variety of purposes. By following best practices and ensuring that your SES account is properly secured, you can ensure that your notifications are delivered reliably and securely.

5.2 CONCLUSION

In conclusion, AWS push notifications of email from EC2 provide numerous benefits to businesses and individuals using the AWS platform. These notifications offer real-time alerts and notifications, making it easy to identify and address issues quickly, which can help to minimize downtime and improve overall reliability. The notifications are easy to configure and highly customizable, allowing users to tailor them to their specific needs and preferences. Push notifications are also cost-effective, meaning that businesses of all sizes can implement and use them without having to worry about high costs. Furthermore, push notifications can improve productivity by reducing the time and effort required to troubleshoot and resolve issues. Overall, AWS push notifications of email from EC2 are a powerful tool that can help businesses stay informed and responsive, improving their overall performance and success in the cloud.

CHAPTER – 6

REFERENCES

- 1) AWS Simple Notification Service (SNS) Documentation:
<https://docs.aws.amazon.com/sns/index.html>
- 2) Amazon EC2 User Guide for Linux Instances:
<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>
- 3) AWS Lambda Documentation:
<https://docs.aws.amazon.com/lambda/index.html>
- 4) Amazon S3 Developer Guide:
<https://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html>
- 5) Amazon CloudWatch Documentation:
<https://docs.aws.amazon.com/cloudwatch/index.html>
- 6) AWS Identity and Access Management (IAM) Documentation:
<https://docs.aws.amazon.com/iam/index.html>
- 7) Amazon Route 53 Developer Guide:
<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/Welcome.html>
- 8) Amazon RDS Documentation:
<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html>
- 9) AWS Elastic Beanstalk Documentation:
<https://docs.aws.amazon.com/elasticbeanstalk/index.html>

- 10) Amazon DynamoDB Developer Guide:
<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html>
- 11) AWS CloudFormation Documentation:
<https://docs.aws.amazon.com/cloudformation/index.html>
- 12) Amazon SQS Developer Guide:
<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/Welcome.html>
- 13) AWS SDK for Java Documentation: <https://docs.aws.amazon.com/sdk-for-java/index.html>
- 14) Amazon SNS Developer Guide:
<https://docs.aws.amazon.com/sns/latest/dg/Welcome.html>
- 15) AWS SDK for JavaScript Documentation: <https://docs.aws.amazon.com/sdk-for-javascript/index.html>
- 16) Amazon API Gateway Developer Guide: <https://docs.aws.amazon.com/ap>
- 17) AWS architecture: https://aws.amazon.com/architecture/?cards-all.sort-by=item.additionalFields.sortDate&cards-all.sort-order=desc&awsf.content-type=*all&awsf.methodology=*all&awsf.tech-category=*all&awsf.industries=*all&awsf.business-category=*all
- 18) <https://aws.amazon.com/ec2/faqs/>
- 19) [https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.htm](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html)
I