Internship Report

On

DEPLOYE A WINDOWS VIRTUAL MACHINE ON AWS

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR, ANANTHAPURAMU

In Partial Fulfillment of the Requirements for the Award of the Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & TECHNOLOGY

Submitted By

G LIKHITHA - 21691A2876

Under the Guidance of

Ms. Gopika Venu

Assistant Professor

Department of Computer Science & Technology

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE

(UGC – AUTONOMOUS)

(Affiliated to JNTUA, Ananthapuramu)

Accredited by NBA, Approved by AICTE, New Delhi)

AN ISO 9001:2015 Certified Institution

P. B. No: 14, Angallu, Madanapalle, Annamayya – 517325

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE

(UGC-AUTONOMOUS INSTITUTION)
Affiliated to JNTUA, Ananthapuramu &Approved by AICTE, New Delhi
NAAC Accredited with A+ Grade

NBA Accredited -B.Tech. (CIVIL, CSE, ECE, EEE, MECH), MBA&MCA

DEPARTMENT OF COMPUTER SCIENCE & TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that the **SUMMER INTERNSHIP-1 (20CST702)** entitled **"DEPLOYE A WINDOWS VIRTUAL MACHINE ON AWS"** is a bonafide work carried out by

G LIKHITHA - 21691A2876

Submitted in partial fulfillment of the requirements for the award of degree **Bachelor of Technology** in the stream of **Computer Science & Technology** in **Madanapalle Institute of Technology & Science, Madanapalle,** affiliated to **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** during the academic year 2023-2024

Guide
Ms. Gopika Venu
Assistant Professor,
Department of CST

Internship Coordinator/CST
Mr.V.Naveen
Assistant Professor,
Department of CST

Head of the Department
Dr.M. Sreedevi
Professor and Head
Department of CST

INTERNSHIP CERTIFICATE

Cert. No:.SDC/23-24/AWSINTERN/00268 PIN: ...21691A2876.....

ANDHRA PRADESH STATE
SKILL DEVELOPMENT CORPORATION

Skill AP APSSDC

CERTIFICATE FOR INTERNSHIP

This is to certify that

Mr/Ms. ...Likhitha G.....

from ....Madanapalle Institute of Technology & Science.....

has completed Internship

on

.....Amazon Web Services-Cloud Computing [Online].....

in APSSDC

from

...22-05-2023..... **to** ...29-07-2023.....

Dr. Ravi K Gujjula
Chief General Manager,
APSSDC

Sri K. Dinesh Kumar
Executive Director
APSSDC

Dr. Vinod Kumar.V, I.A.S.
Managing Director
APSSDC

DECLARATION

I hereby declare that results embodied in this **SUMMER INTERNSHIP-1(20CST702) "DEPLOYE A WINDOWS VIRTUAL MACHINE ON AWS"** by me under the guidance of **Ms. Gopika Venu**, **Assistant Professor, Dept. of CST** in partial fulfillment of the award of **Bachelor of Technology** in **Computer Science & Technology** from **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** and I have not submitted the sameto any other University/institutefor award of any other degree.

Date :

Place :

PROJECT MEMBER

G.Likhitha (21691A2876)

I certify that above statement made by the student is correct to the best of my knowledge.

Date : **Guide**

Ms. Gopika Venu

TABLE OF CONTENT

ABSTRACT

The abstract of creating and deploying a virtual machine (VM) on Amazon Web Services (AWS) encapsulates the core essence of leveraging cloud-based infrastructure for efficient computing. It involves the seamless setup, configuration, and utilization of VM instances within AWS's ecosystem. This abstract outlines the process of creating a flexible, scalable, and cost-effective computing environment by customizing VMs to match specific project requirements. The deployment involves considerations such as resource optimization, cost management through AWS's flexible payment models, and robust security measures ensuring data integrity and compliance. Additionally, deploying VMs across multiple availability zones bolsters reliability, minimizing potential downtime. The abstract underscores the accessibility and global reach offered by AWS, facilitating enhanced user experiences. Moreover, the utilization of AWS's management tools and automation capabilities streamlines deployment proce`dures, fostering agile development and testing of applications. In essence, the abstract of creating and deploying a VM on AWS encapsulates the journey toward a dynamic, secure, and accessible cloud-based infrastructure for optimized computing solutions.

List of Figure

LIST OF ABBREVIATIONS

AWS	Amazon Web Services
DNS	Domain Name System
CCC	Cloud Control Center
CDN	Content Delivery Network
S3	Simple Storage Service
URL	Uniform Resource Locator
AMP	Access &Mobility Management Function
API	Application Programming Interface
BBS	Business Support System
CDK	Cloud Development System
AUSF	Authentication Support Function
CLI	Command Line Interface

CHAPTER 1

INTRODUCTION

1.1 About Industry or Organization Details

APSSDC is a unique organization formed as a Public Private Partnership (PPP) corporation to promote skill development & entrepreneurship in the state of Andhra Pradesh. The Corporation is incorporated as a Section-8 company (not-for-profit) with a private equity component of 51% and 49% by Govt. of AP. The Corporation serves as the 'Executive Agency' for the Department of Skill Development, Entrepreneurship and Innovation, Govt. of AP. APSSDC will also serve the important task of providing high quality skilled manpower as part of the 'Knowledge and Skills Mission' of GoAP, which is one of the cross-cutting missions among the seven missions formed by the Govt. to facilitate double digit growth of the state. In this backdrop, APSSDC, in partnership with NortheasternUniversity (NU), Boston, Massachusetts, USA, has established International Institute of Entrepreneurship Development (i2E) to rapidly build a vibrant ecosystem consisting of world-class entrepreneurship development centers, supported by maker spaces, mentor/investor networks, entrepreneurship clubs (e-clubs) and a host of other catalytic incentives, to:

- Promote a culture of innovation-based entrepreneurship
- Build competencies through practical, hands-on experiential learning
- Help empower entrepreneurs to launch successful ventures.

1.2 My Personal Benefits

Through the AWS Cloud Computing internship, I've gained practical experience and skills. This includes foundational knowledge of key AWS services, hands-on experience with infrastructure automation using AWS CloudFormation, and the ability to design scalable architectures with AWS Auto Scaling. I've also developed proficiency in data management using services like S3 and RDS, along with knowledge of security practices such as IAM and VPC. Networking skills were honed through the setup and configuration of virtual networks with AmazonVPC. Additionally, I gained experience in monitoring and logging using AWS CloudWatch and CloudTrail and learned about cost optimization through tools like AWS Budgets and Trusted Advisor. These skills collectively enhance my technicalcapabilities and prepare me for future rolesin AWS and cloud technology.

1.3 Objective of the Project

The primary objective lies in leveraging cloud-based infrastructure to optimize computing resources. This entails creating a flexible, scalable, and cost-efficient environment tailored to the project's needs. By deploying vms on AWS, the project aims to achieve agility in resource allocation, allowremains a key focus, utilizing aws's pay-as-you-go model to manage expenses by starting, stopping, or terminating instances as necessary. Security measures, encompassing network isolation, encryption, and robust IAM, ensure data protection and compliance adherence. The project emphasizes high availability and reliability by deploying vms across multiple availability zonesing seamless adjustments in CPU, memory, and storage to meet varying workload demands. Cost-effectiveness, mitigating downtime risks.

1.4 Limitations of Project

1. Learning Curve: Utilizing AWS services requires a learning curve, especially for beginners, due to the complexity of various services, configurations, and networking principles.

2. Cost Management: AWS's pay-as-you-go model can result in unexpected costs if instances are not properly managed or if resources are over-provisioned. Monitoring and optimizing costs demand continuous attention.

3. Security Concerns: Misconfigurations in security settings or inadequate access controls might expose vulnerabilities. Ensuring proper configurations and adherence to security best practices is essential.

4. Vendor Lock-in: Using AWS-specific services could lead to vendor lock-in, making it challenging to migrate to another cloud provider without significant architectural changes.

5. Performance Variability: As a shared infrastructure, instances might occasionally experience performance fluctuations due to resource contention, affecting consistent performance.

6. Reliance on Internet Connectivity: AWS services heavily depend on internet connectivity, making them vulnerable to disruptions or connectivity issues.

7. Complexity in Management: As the number of instances grows, managing and monitoring the infrastructure becomes more complex, requiring additional tools and resources.

CHAPTER 2
SYSTEM ANALYSIS

2.1 INTRODUCTION

2.1.1 CLOUD DEPLOYING MODEL

The integration of software-as-a-service (SaaS), platform-as-a-service (PaaS), and infrastructure-as-a-service (IaaS) into the cloud, to act as a solution that allows users to access datais known as cloud deployment.

Types of deployment models

Most cloud hubs have tens of thousands of servers and storage devices to enable fast loading. It is often possible to choose a geographic area to put the data "closer" to users. Thus, deployment models for cloud computing are categorized based on their location. To know which model would best fit the requirements of your organization.

Public Cloud

The name says it all. It is accessible by the public. Public deployment models in the cloud are perfectfor organizations with growing and fluctuating demands. It also makes a great choice for companieswith low-security concerns. Thus, you pay a cloud service provider for networking services, compute virtualization & storage available on the public internet. This is also a great delivery modelfor the teams with development and testing. Its configuration and deployment are quick and easy, making it an ideal choice for test environments.

Benefits of Public Cloud

- Minimal Investment
- No Hardware Set-up
- No Infrastructure Management
- Limitations of Public Cloud
- Data Security and Privacy
- Reliability Issues

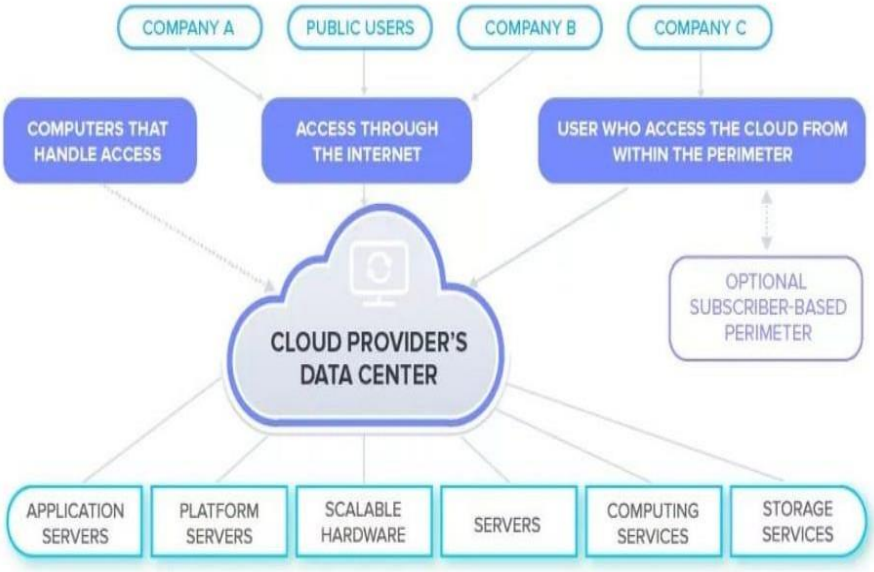

FIG : 2.1.1 Cloud providers

2.2 Existing System:

Ensure you have an AWS account. If not, sign up for one.Access the AWS Management Console. Create a new S3 bucket, providing a unique name.Set up the necessary configurations foryour S3 bucket, including versioning, logging,and permissions.Upload your static website files (HTML, CSS, JavaScript, images, etc.) to the S3bucket using the Navigate to the S3 service in the AWS Management Console.AWS Management Console, AWS CLI, or SDKs.In the S3 bucket **AWS** properties, go to the "Static website hosting" pane.Enable static website hosting and specify the default index and error documents.Adjust bucket policies and permissions to ensure proper access to your static websiteresources. If you have a custom domain, consider using AWS Route 53or another DNS providerto configure the necessary settings for your domain to point to the S3 staticwebsite endpoint.Test your static website by accessing the provided S3 website endpoint or customdomain if configured. Set up load balancers using Elastic Load Balancing (ELB) to distribute traffic and ensure high availability. Configure auto-scaling to handle traffic fluctuations.**i**mplement monitoring using AWS CloudWatch for resource performance, logs, and alarms.

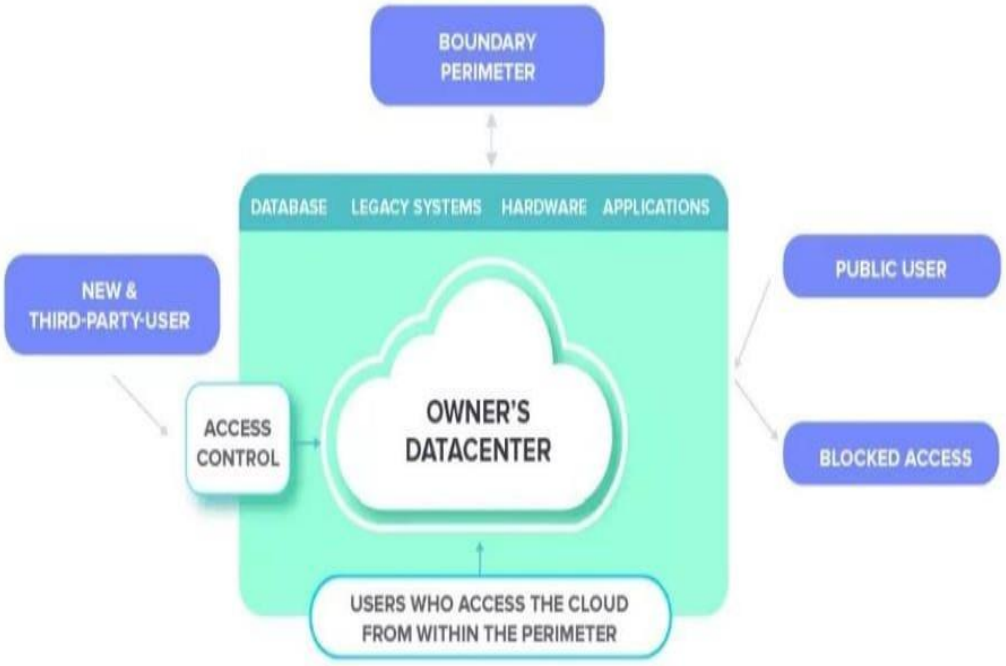

FIG : 2.2.1 Owner Datacenter

2.3 Disadvantages of Existing System

The disadvantages of an existing system, which you might want to address orimprove upon by migrating to AWS, could include:

Scalability Limitations

The current system might struggle to handle sudden spikes in traffic or scaling demands. AWSoffers scalable infrastructure and services that can dynamically adjust to loads.Complexity for Dynamic Content.

Limited Availability and Reliability

If the existing system lacks redundancy or high availability measures, it could be prone to downtime. AWS provides fault-tolerant services and multiple availability zones to enhancereliability**.**

Performance Issues

The performance of the current system might not meet desired standards, leading to slow response times or poor user experiences. AWS offers various performance-optimized solutionsand CDN services like CloudFront for faster content delivery**.**

Security

Security vulnerabilities or insufficient security measures in the existing system could pose risks. AWS provides robust security features, including encryption, IAM, security groups, and compliance certifications.

Limited Disaster and Recovery

Inadequate backup strategies or lack of disaster recovery options in the current setup could result in data loss or extended downtime during disruptions. AWS provides backup services,snapshots, and disaster recovery solutions for data resilience.

Technological

The existing system might be built on outdated technologies or software versions, leading to compatibility issues or limited capabilities. AWS offers modern, up-to-date services and supports various programming languages and frameworks.

FIG :2.3.1 Traditional Methodology

2.4 Proposed System

In AWS we have lot of services on pay as you go basis And We can host easily with the help of those services and the cost will also reduced.If you are considering a proposed system for hosting a static website on AWS S3, hereare steps you might take.

The proposed system aims to streamline this process by implementing AWS's Infrastructure as Code (IaC) services, specifically AWS CloudFormation. The company plans to create templates using CloudFormation that define the desired infrastructure and configurations for various development environments such as testing, staging, and production.

Instead of manually setting up VM instances each time, developers will utilize these CloudFormation templates. For instance, a single CloudFormation template will describe the necessary resources, including EC2 instances, networking, security groups, and associated services, required for a particular development environment.

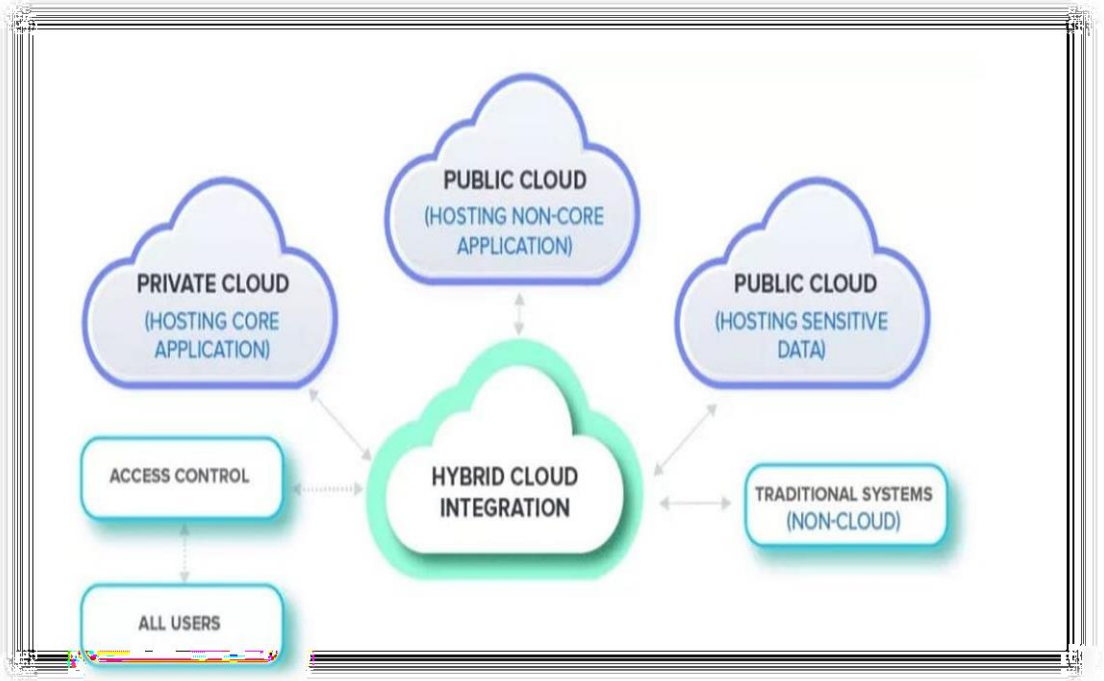

FIG :2.4.1 : Hybride cloud Integration

2.5 Advantages Over Existing System

Hosting a static website on AWS S3 offers several advantages, including:

Scalability:

AWS S3 is highly scalable, allowing your website to handle varying levels of trafficeasily.

Cost-Effective:

S3 can be cost-effective for hosting static content, especially for low to moderate levelsof traffic, as you only pay for the storage and data transfer you use.

Reliability:

AWS provides a reliable infrastructure with redundancy and backups, ensuring highavailability for your static website.

Global Content Delivery:

Use of AWS CloudFront in conjunction with S3 allows for global content delivery,reducing latency by distributing content across multiple edge locations.

Security:

AWS offers robust security features, including access controls and encryption options,ensuring the confidentiality and integrity of your static website data.

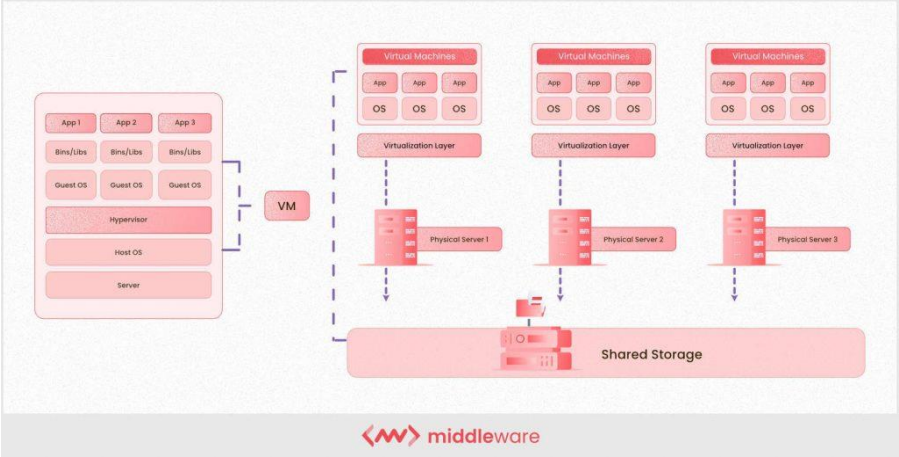

FIG : 2.5.1Advantages Over Existing System

2.6 MANAGING THE RISKS OF PUBLIC CLOUDS

The same hardware platform. A shared environment also presents resource competition problems whenever one of the customers uses most of the resources due either to need or to being exposed to targeted attacks, such as DDoS (distributed denial of service).

Security Measures and Access Control: Utilize AWS Identity and Access Management (IAM) to manage user access, permissions, and roles, ensuring the principle of least privilege. Implement robust authentication mechanisms and encryption to protect data both in transit and at rest. Regularly monitor and audit access to identify and mitigate security risks.

Data Protection and Compliance: Understand data residency and compliance requirements. Employ AWS services like AWS Key Management Service (KMS) for encryption, AWS CloudHSM for key storage, and AWS Config for compliance monitoring. Regularly back up data and ensure it complies with relevant regulatory standards (GDPR, HIPAA, etc.).

Resilience and Disaster Recovery: Leverage AWS's global infrastructure for high availability and disaster recovery. Use AWS services such as Amazon S3 for data backup, AWS CloudWatch for monitoring, and AWS Elastic Load Balancing for fault tolerance to ensure resilience against failures.

Risk Assessment and Monitoring: Continuously assess and monitor risks using AWS-native tools like Amazon Inspector for vulnerability scanning and AWS GuardDuty for threat detection. Implement logging and monitoring services like AWS CloudTrail and Amazon CloudWatch to track activities and respond promptly to security incidents.

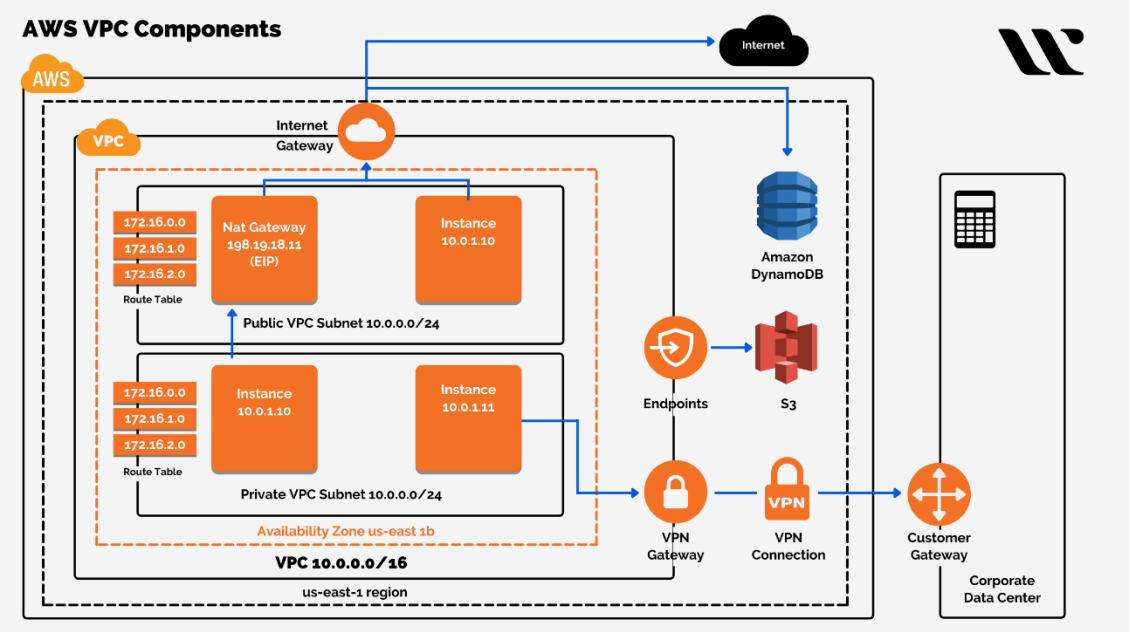

FIG : 2.6.1 MANAGING THE RISKS OF PUBLIC CLOUDS

2.7 AMAZON WEB SERVICES

Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud platform, offering over 200 fully featured services from data centers globally. Millions of customers-including the fastest-growing startups, largest enter prises, and leading government agencies- are using AWS to lower costs, become moragile, and innovate faster.

Most Functionality

- Largest community of consumers and partners
- Most Secure
- Fastest pace of Innovation
- Most proven operational expertise

Amazon Relational Database Service (Amazon RDS) is a web service that makesit easier to set up, operate, and scale a relational database in the AWS Cloud. It provide cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

Amazon Elastic Compute Cloud (amazon ec2)**provides scalable computing** capacity in the AmazonWeb Services (AWS) Cloud. Using Amazon EC2 eliminatesyour need to invest in hardware up front, So you can develop and deploy applicationsfaster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, Reducing your need to forecast traffic**.**

FIG : 2.1.7 Amazon EC2

Amazon Route 53 is a highly available and scalable Domain Name System (DNS)web service. You can use Route 53 to perform three main functions in any combination: domain registration, DNS routing, and health checking

CHAPTER 3

SYSTEM SPECIFICATON

3.1 Hardware Requirement Specification

- Device name : No-One
- Processor : Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz 2.50 GHz
- Installed RAM : 16.0 GB (15.8 GB usable)
- System type : 64-bit operating system, x64-based processor

3.2 Software Requirement Specification

- Operating System : windows
- Edition : Windows 11 Home Single Language
- Version : 22H2
- OS build : 22621.2283

3.3 Other Requirements

- AWS account

Whether you are moving to Amazon Web Services or are already running cloud- native, Ubuntu is the platform of choice for AWS. Canonical continuously tracks and delivers updates to Ubuntu images. Lean, stable, fast and powerful, Ubuntu Server deliversservices reliably, predictably and economically. It is the perfect base on which to build yourinstances. Ubuntu is very secure, built-in from the moment your machines and containers launch. Also, it is free and will always be, and you have the option to get support and Landscape from Canonical.Visual Studio Code is a lightweight but powerful code editor redefined and optimized for building and debugging modern web and cloud applications runs on your desktop and it is available for MacOS, Windows and Linux.

CHAPTER 4

SYSTEM DESIGN

4.1 System Architecture

User Interface Layer: This layer represents the user-facing components, such as web or mobile applications, through which users interact with the system. It includes the client-side interfaces and applications used to access the services deployed on AWS.

Application Layer: The application layer consists of the backend services, logic, and processing components that handle business logic, data processing, and application functionalities. This layer might include AWS services like AWS Lambda for serverless computing, Amazon EC2 for virtual servers, or AWS Elastic Beanstalk for application deployment and management.

Data Storage Layer: This layer involves various data storage components where information is stored and managed. It may include Amazon S3 for object storage, Amazon RDS or Amazon DynamoDB for databases, or Amazon Redshift for data warehousing, based on the specific requirements of the application.

Networking and Security Layer: This layer includes components responsible for network connectivity, security, and access control within the AWS environment. It involves services like Amazon VPC for network isolation, AWS IAM for access management, AWS WAF or AWS Shield for protection against web attacks, and AWS Security Hub for centralized security monitoring.

Integration Layer: The integration layer facilitates communication and interaction between different components or services within the system. It may involve AWS API Gateway for creating and managing APIs, AWS Event Bridge for event-driven architectures, and AWS Step Functions for orchestrating workflows.

Management and Monitoring Layer: This layer includes tools and services for managing, monitoring, and optimizing the AWS environment. It encompasses AWS CloudWatch for monitoring metrics and logs, AWS CloudTrail for auditing and tracking API activity, and AWS Trusted Advisor for cost optimization and best practice recommendations.

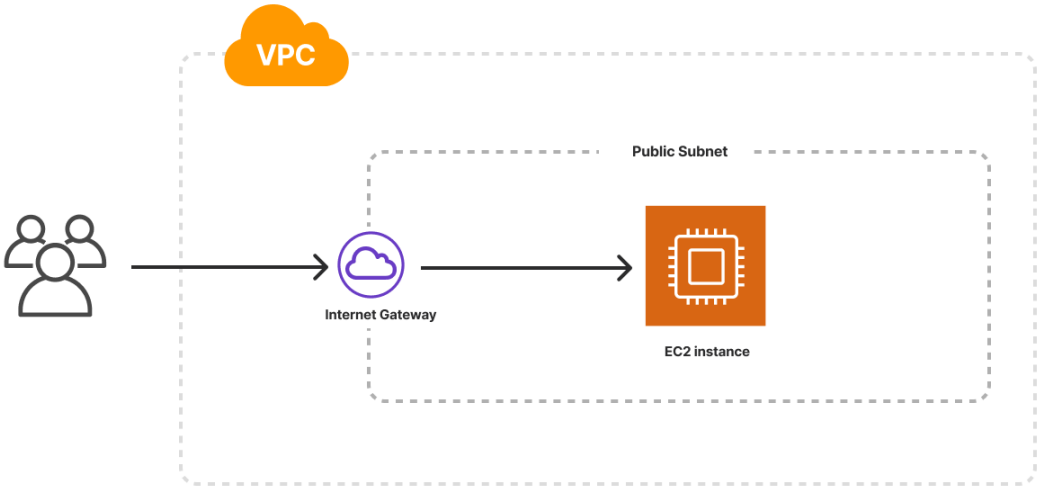

FIG:4.1.1 AWS Architecture

The Virtual Machine template creates an infrastructure as code project in your favorite language that deploys a virtual machine to AWS. You can then use the virtual machine to build your own web application, backend service, or database. The architecture includes Amazon EC2 for the virtual machine and Amazon VPC for the virtual network. The template generates a complete Pulumi program, including a simple HTTP server, to give you a working project out of the box that you can customize easily and extend to suit your needs.

4.2 Model Flow Diagram

There are eight Modules

1. Create an AWS account
2. Launch AWS virtual machine
3. Choose AMI
4. Choose and configure instance type
5. Add storage and tags
6. Configure security
7. Review and launch your AWS virtual machine
8. Connect to an instance

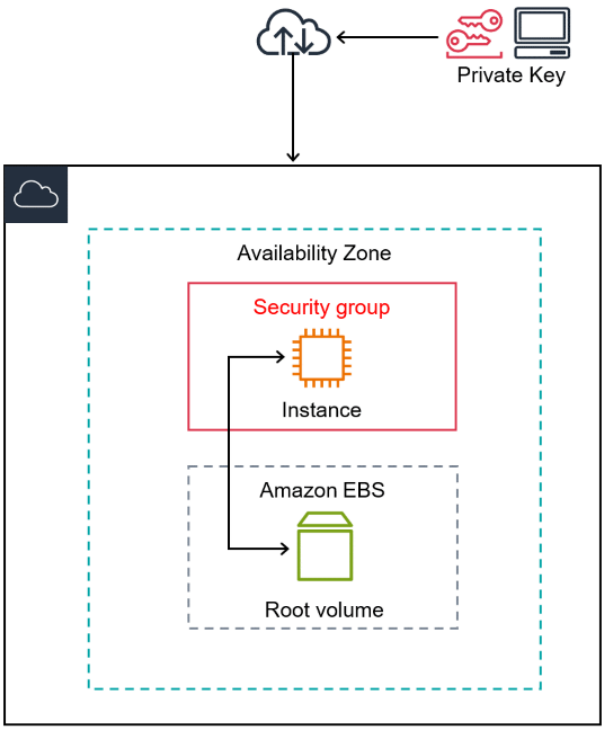

FIG:4.1.2 AWS Flow Diagram

The AWS Management Console or AWS SDK/API can be used to execute these steps programmatically, allowing for automation and efficient deployment of virtual machines on the AWS cloud.

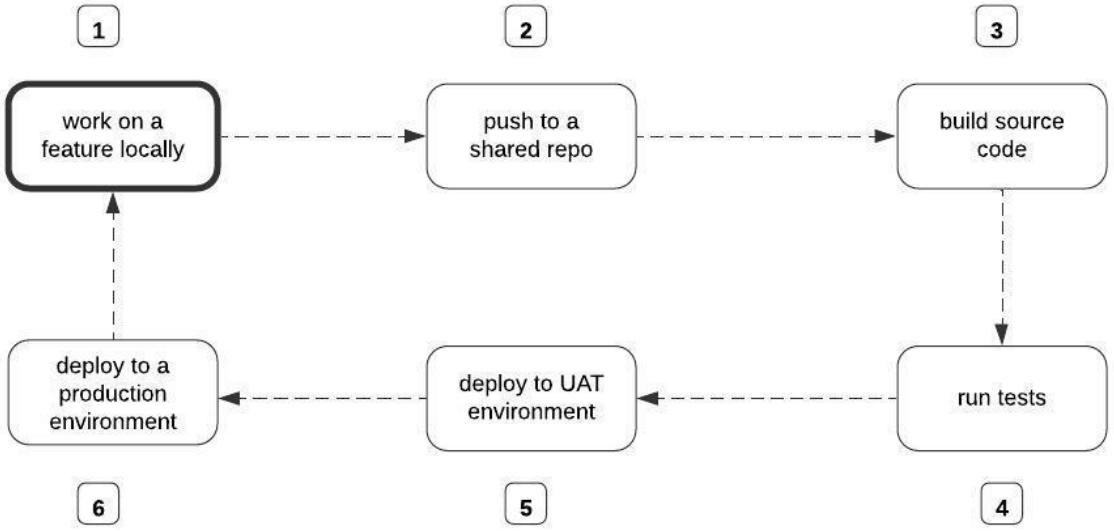

Fig4.1.3 module flow diagram

Module Interactions:

AWS Management Console or AWS CLI/API: These are interfaces used to interact with AWS services.

EC2 Service: The core service responsible for managing virtual servers (EC2 instances).

Amazon Machine Image (AMI): An AMI serves as a template for the root volume of an EC2 instance. It can be a base OS image or a pre-configured image with specific software.

Instance Types: Different instance types offer varying combinations of CPU, memory, storage, and networking capacity**.**

Virtual Private Cloud (VPC): Provides isolated network environments within AWS.

Subnets: Segments of IP address ranges in the VPC. Instances reside within subnets.

Security Groups: Acts as a virtual firewall to control inbound and outbound traffic to instances.

Elastic Block Store (EBS): Provides block-level storage volumes for EC2 instances. It's used for persistent storage.

Key Pairs: Used for securely connecting to instances via SSH (Secure Shell) or RDP (Remote Desktop Protocol).

Selection and Configuration:

User accesses the AWS Management Console or uses AWS CLI/API to specify requirements for the EC2 instance.

They choose the region, instance type, AMI, networking configuration, storage options, security settings, etc.

Instance Provisioning:

The EC2 service provisions an instance based on the selected specifications and the chosen AMI.

It attaches necessary EBS volumes according to the configuration.

Networking Configuration:

VPC allocates IP addresses and assigns them to the instance within specified subnets.

Security Groups enforce inbound and outbound traffic rules as per the defined settings.

Key Pair Usage:

During instance creation, the user can select or create a key pair, which grants secure access to the instance.

Instance Start-Up:

The instance boots up using the specified AMI and gets associated with the configured networking components.

Monitoring and Management:

The user can monitor the instance status, performance metrics, and manage the instance lifecycle through the console or API.

Instance Interaction:

Once the instance is running, users can interact with it via SSH, RDP, or other protocols depending on the configured setting.

Database Interaction:

Databases interact with the deployment and creation of virtual machines in AWS because they store, manage, and provide access to data crucial for applications and services running on those VMs, forming an integral part of the interconnected infrastructure.

Response to User:

They play a critical role in storing, managing, and providing access to data that applications hosted on those virtual machines rely upon for their functionality and operation.

4.3 Setting Up The AWS

Create an AWS account

You can easily create an AWS account on the AWS Console. You can easily create an AWS account on the AWS Console. All new sign-ups get a free-tier offer.

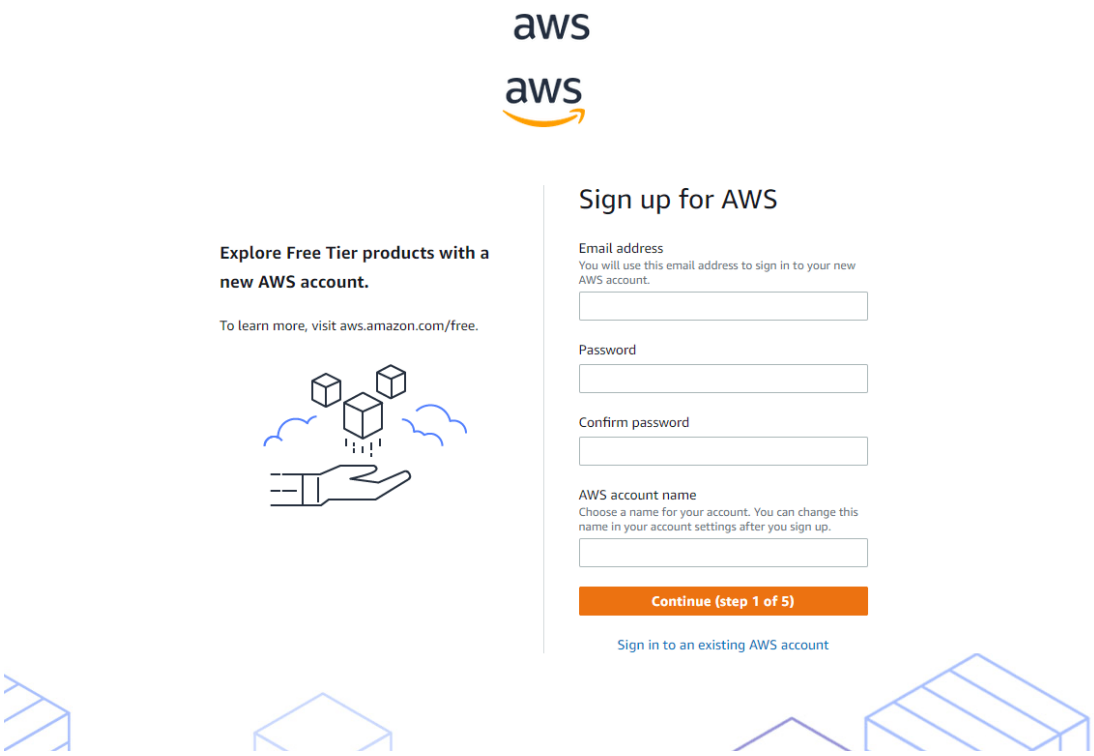

Fig 4.3.1 sign up for AWS

Launch AWS virtual machine

Once you finish setting up your account, you can click on the AWS logo on the top left corner or search "console" on the search bar. You'll find a number of options in the AWS console. Select "Launch a virtual machine" to get started with VMs. If you're a new user, it can take up to 24 hours for your account to activate.

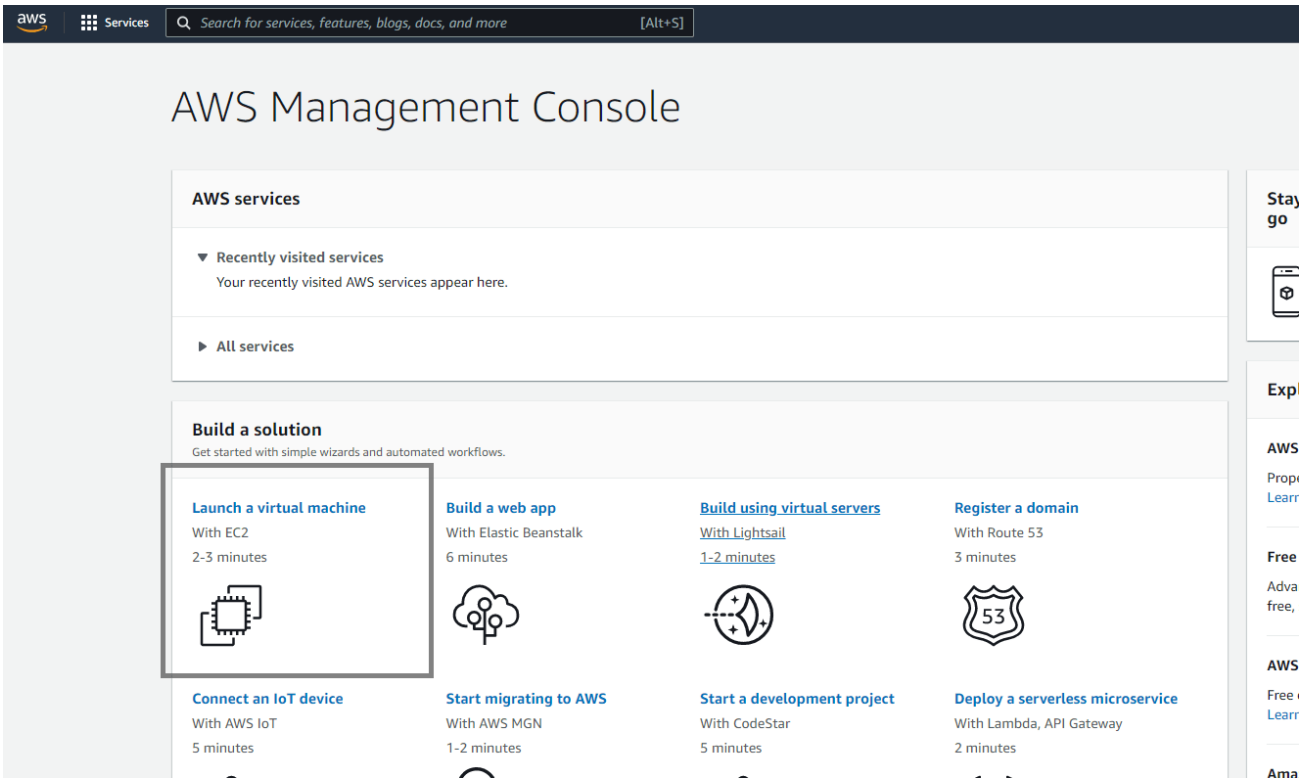

FIG 4.3.2 AWS MANAGEMENT CONSOLE

Choose AMI

Amazon Machine Image (AMI) highlights the software setup (OS, application server, and apps). You can select Mac, Linux, or Windows OS. We'll look at the setup for Windows virtual machines here.

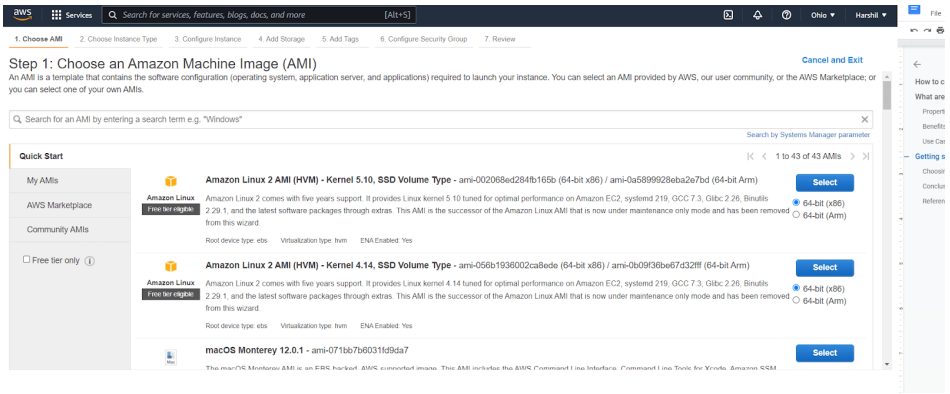

FIG 4.3.3 CHOOSE an AMI

Choose and configure instance type

After choosing your operating system, you need to pick an instance type. Amazon EC2 offers many instance types tailored to specific use cases. An instance is a virtual server or virtual machine. They come in a variety of CPU, memory, storage, networking, and a lot more.

Add storage and tags

Once you configure an instance type, you can add or update storage info. AWS allows you to add more EBS volumes and instance store volumes, as well as change the root volume's parameters. Amazon Elastic Block Store (EBS) provides block-level storage volumes for use with EC2 instances. It behaves like raw, unformatted block devices. You can mount these volumes as devices on your instances.

The next step is adding tags. A tag is a label applied to an AWS resource. Each tag has a key and an optional value, which users define.

Configure security

A security group is a set of firewall rules that control data entering and exiting your instance. You may either recreate it or pick an existing security group.

Security is a major concern when working on public clouds like AWS and Google Cloud Platform (GCP). Attackers can launch different attacks on public cloud deployments for lack of provider security. These attacks include DOS, DDOS, website defacement, and brute-force.

Public clouds have poor security, but with the right set of rules, they can be improved.

Public clouds offer limited customization. Clients can choose the operating system and size of the virtual machine.

Cloud data breaches are frequently caused by misconfigured cloud security settings. Many companies' cloud security posture management solutions are insufficient for safeguarding their cloud-based infrastructure.

Review and launch your AWS virtual machine

The final step in creating an AWS virtual machine is to go through your instance details. Make sure every detail is correct, then click “Launch”.

When you click “Launch,” you need to provide a key. To create a new key, select “Create a new key pair” from the drop-down menu and set a key name, for example, keytask, keytest1, and so on. Make sure you download “key pair” before launching your instance.

A key pair is made up of a public key stored by AWS and your private key file. They work together to allow you to connect to your instance safely.

Connect to an instance

After starting the instance, you can check the status using “Dashboard>Instances”. Select your instance in the instance dashboard and click “Connect”.

Select “RDP client,” click “Get password,” then upload the key pair downloaded when the instance launched (in step 7). After uploading the file, click “decrypt password” and download the remote desktop file.

Open the downloaded file and enter your password.

You should now see a screen similar to the one below, indicating that your AWS Windows virtual machine successfully launched!

Optimize your resources with a virtual machine.

Software developers are always looking to increase software efficiency for better usage and management. As explained above, virtual machines can be a great asset to both businesses and individuals, helping them get rid of the hassle of physical server management. All they need is a deep understanding of what VMs are and how they’re created!

CHAPTER 5

IMPLEMENTATION AND RESULTS

5.1 Introduction

When introducing the implementation and results for deploying end to end website onAWS S3,consider outlining the following points:

1. Project Overview:

The project aims to establish standardized, automated procedures for deploying virtual machines on AWS, focusing on security, efficiency, and scalability.

2. AWS EC2 Setup:

- "Configuring an AWS EC2 instance involves selecting specifications, setting up storage, defining networking, configuring security, and launching the instance for cloud-based operations."

3. File Structure and Content:

- Discuss the organization of remote desktop connection within the EC2 instance.

4. Deployment Process:

- The deployment process for an EC2 instance involves selecting an operating system, specifying hardware configuration, setting up networking, storage, security rules, and launching the instance in the AWS cloud platform.

5. Performance Considerations:

- Discuss any optimizations made for performance, such as content compression,caching strategies, or content delivery network (CDN) integration.

6. Security Measures:

- Touch upon security configurations implemented, like restricting public access,setting proper permissions, and enabling logging for the ec2 instance.

7. Results and Metrics:

- The results and metrics for deploying an EC2 instance on AWS include monitoring instance performance, ensuring security compliance, optimizing resource usage, and tracking associated costs to maintain operational efficiency and meet business requirements.

8. Cost Analysis:

- Provide insights into the cost implications of deploying virtual machine on AWS
- Highlight any cost-saving measures or considerations.

5.2 Implementation of Key Functions

The implementation of key functions within a project involves executing and bringing to life the essential features or core components that define the project's purpose or objectives. Here are steps for implementing key functions within a project:

1. Requirement Analysis:

Understand and document the specific requirements and functionalities that the key functions need to fulfill based on project goals and user needs.

2. Design Phase:

Create a detailed plan or design that outlines how the key functions will be developed and integrated into the project. This phase might involve architectural design, data flow diagrams, and outlining algorithms or logic.

3. Development:

Coding: Write code to develop the key functions, following the design specifications and best practices.

Testing: Perform unit tests and validation to ensure the implemented functions work as intended, catching and fixing any issues early in the development process.

4. Integration and Quality Assurance:

Integrate the implemented functions into the larger project or system.

Conduct system testing and quality assurance to verify that the integrated functions interact correctly with other components and meet the overall project requirements.

5. Documentation:

Create documentation detailing the implemented functions, including usage guidelines, APIs, or any relevant technical information for future reference or for other team members.

6. User Acceptance Testing (UAT):

Involve stakeholders or end-users to perform UAT, allowing them to validate and confirm that the implemented functions meet their needs and expectations.

7. Deployment:

Once the implemented functions pass testing and validation, deploy them into the production environment or the intended platform for operational use.

8. Monitoring and Maintenance:

Continuously monitor the performance and behavior of the implemented functions in the live environment.

Provide ongoing maintenance and support to address any issues, optimize performance, and accommodate potential updates or improvements.

Throughout the implementation process, effective communication, collaboration among team members, adherence to best practices, and agile methodologies can enhance the efficiency and success of implementing key project functions. Adjustments or iterations might be necessary based on feedback and changing requirements to ensure that the implemented functions align with the project's objectives.

5.3 Output Screens and Result Analysis

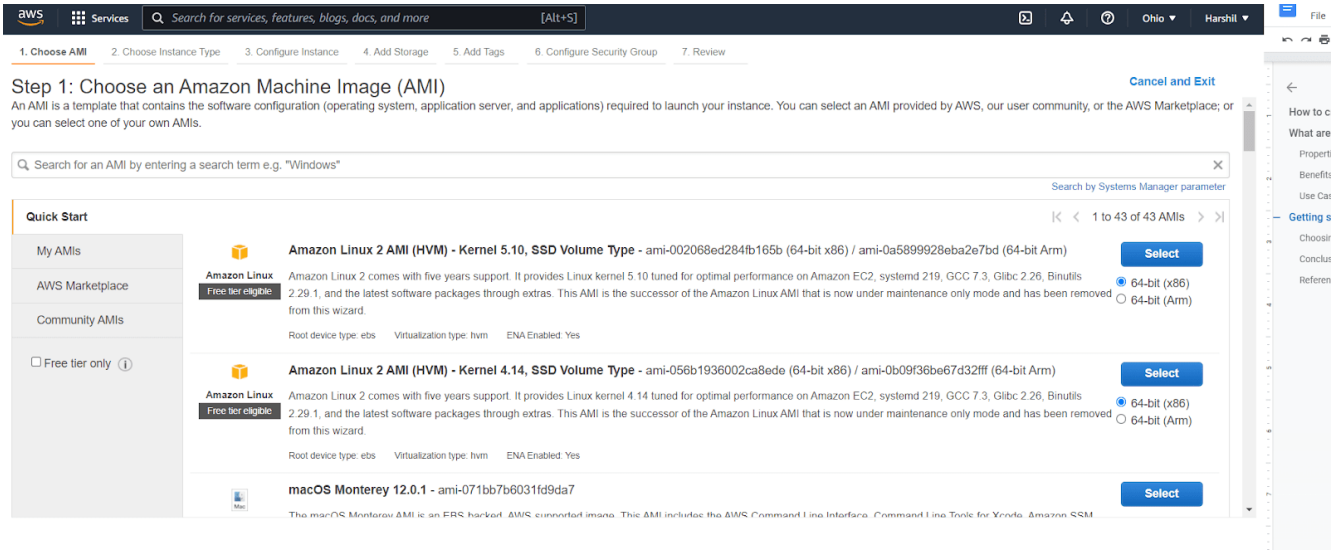

FIG 5.3.1 Step 1

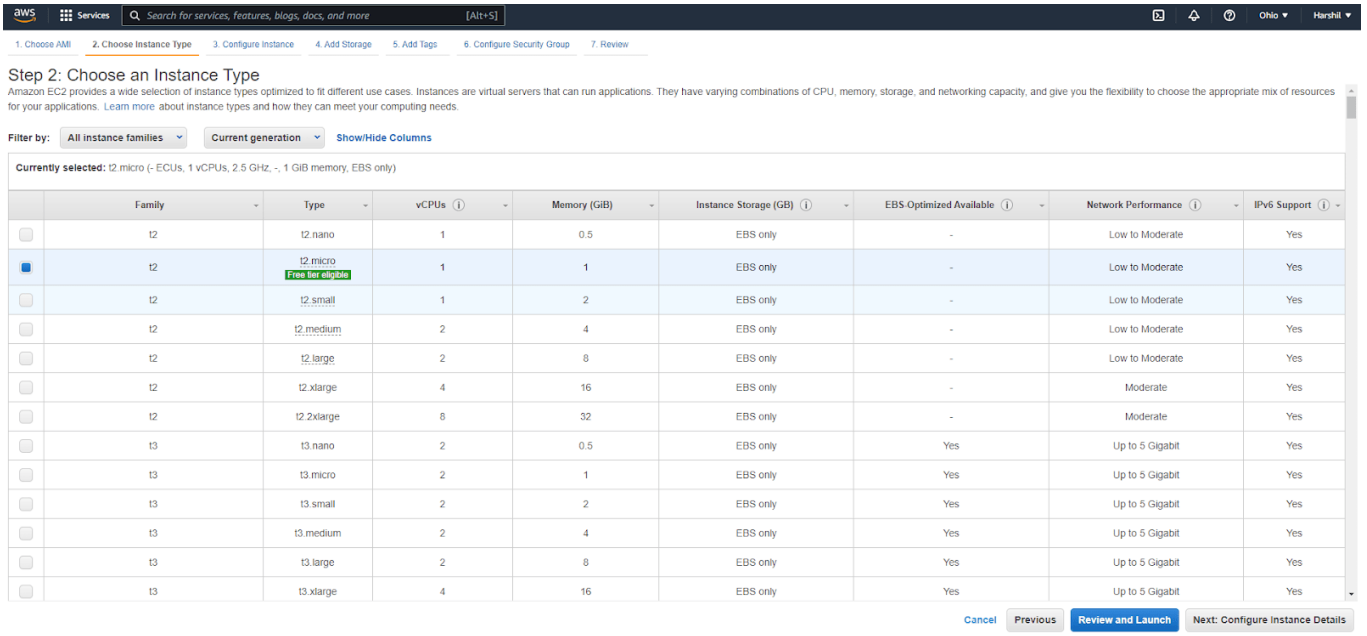

FIG 5.3.2 STEP 2

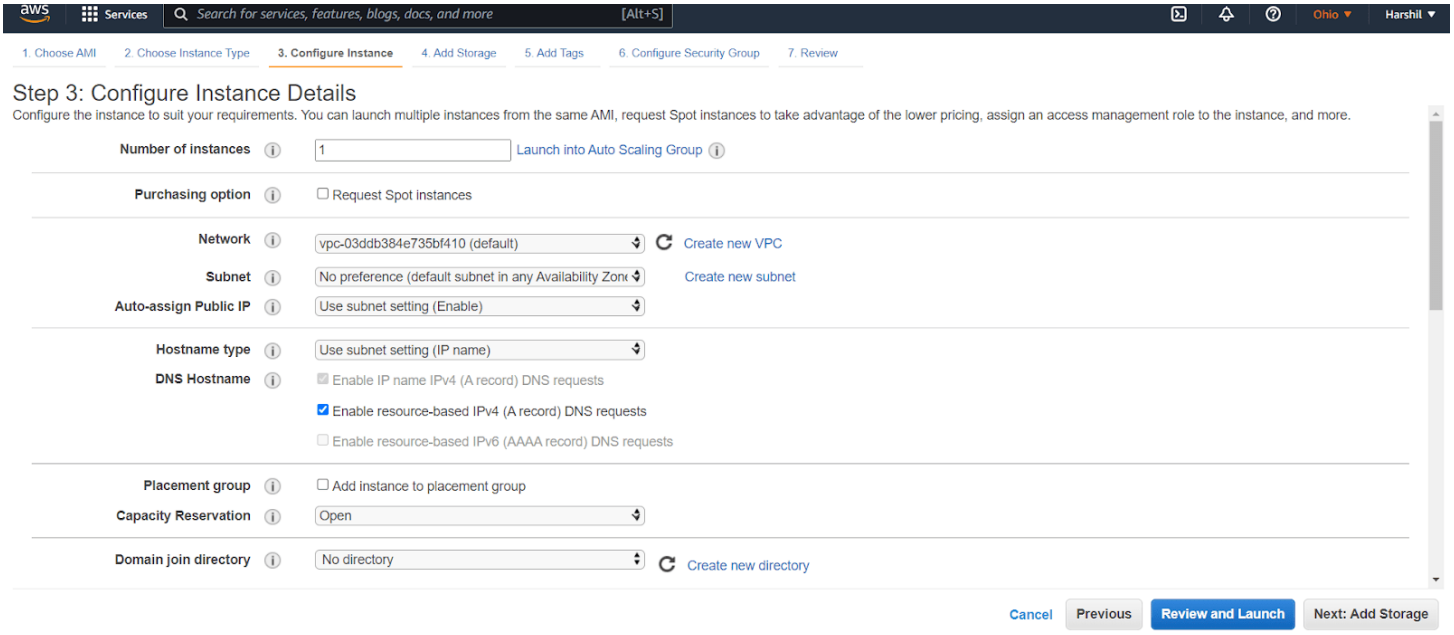

FIG 5.3.3 STEP 3

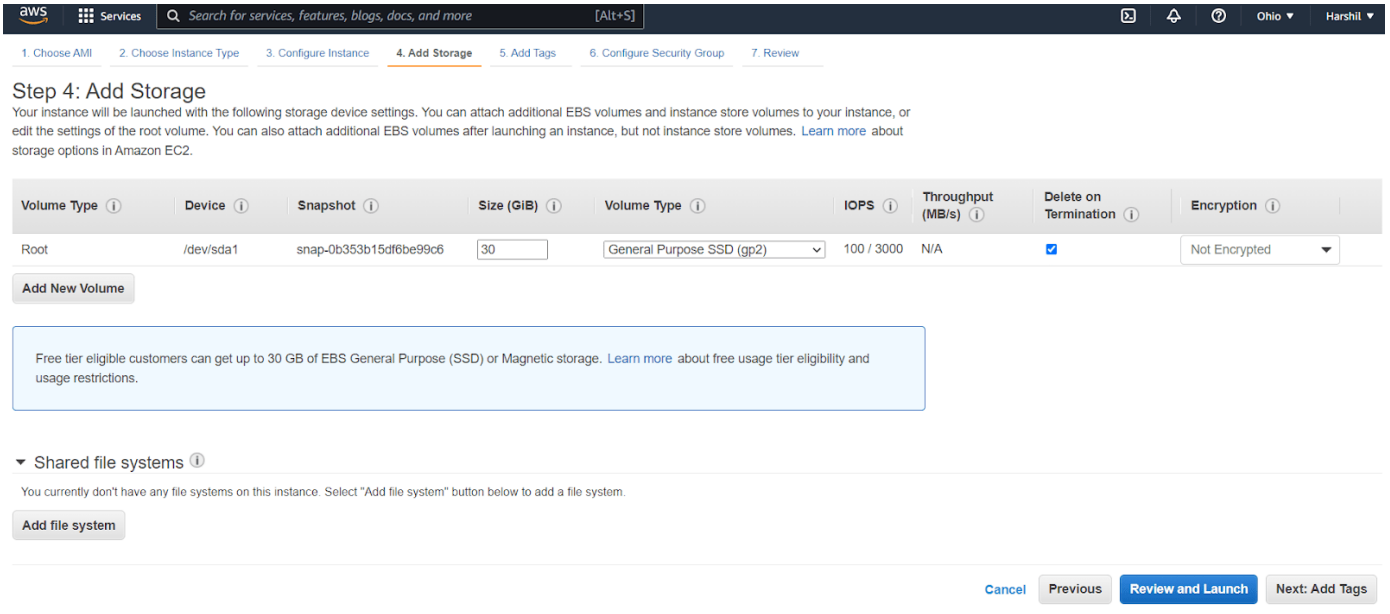

FIG 5.3.4 STEP 4

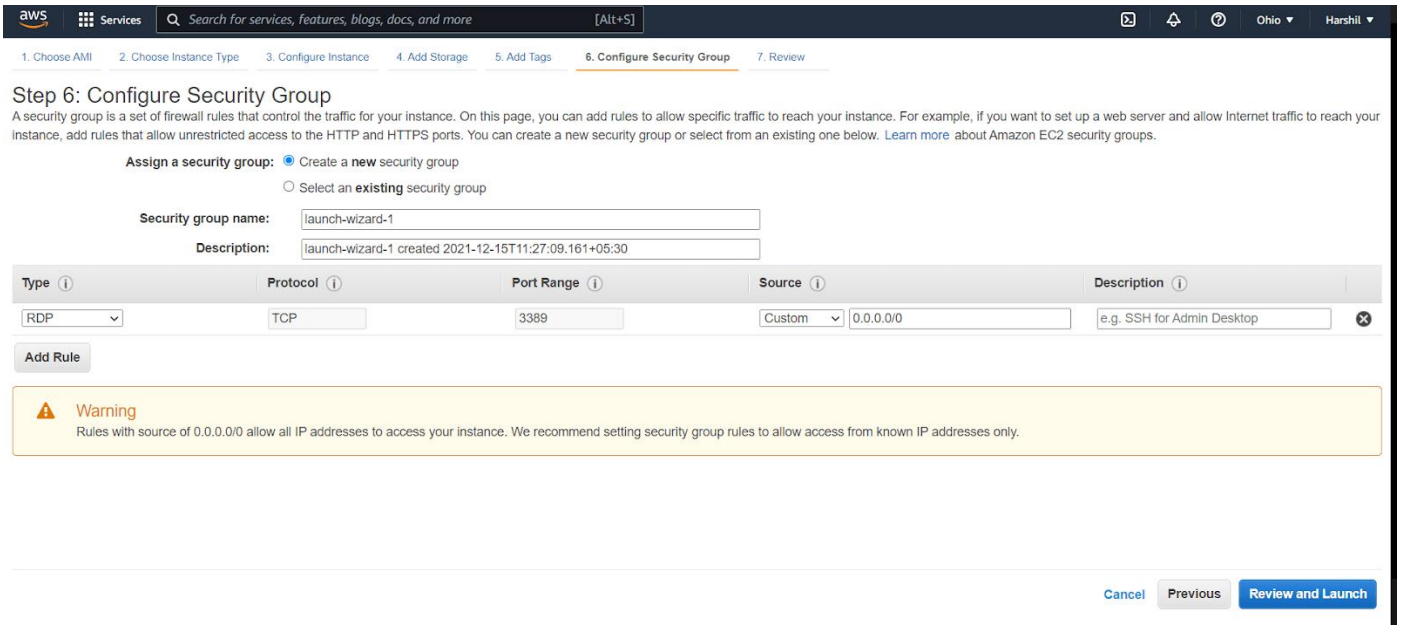

FIG 5.3.5 STEP 5

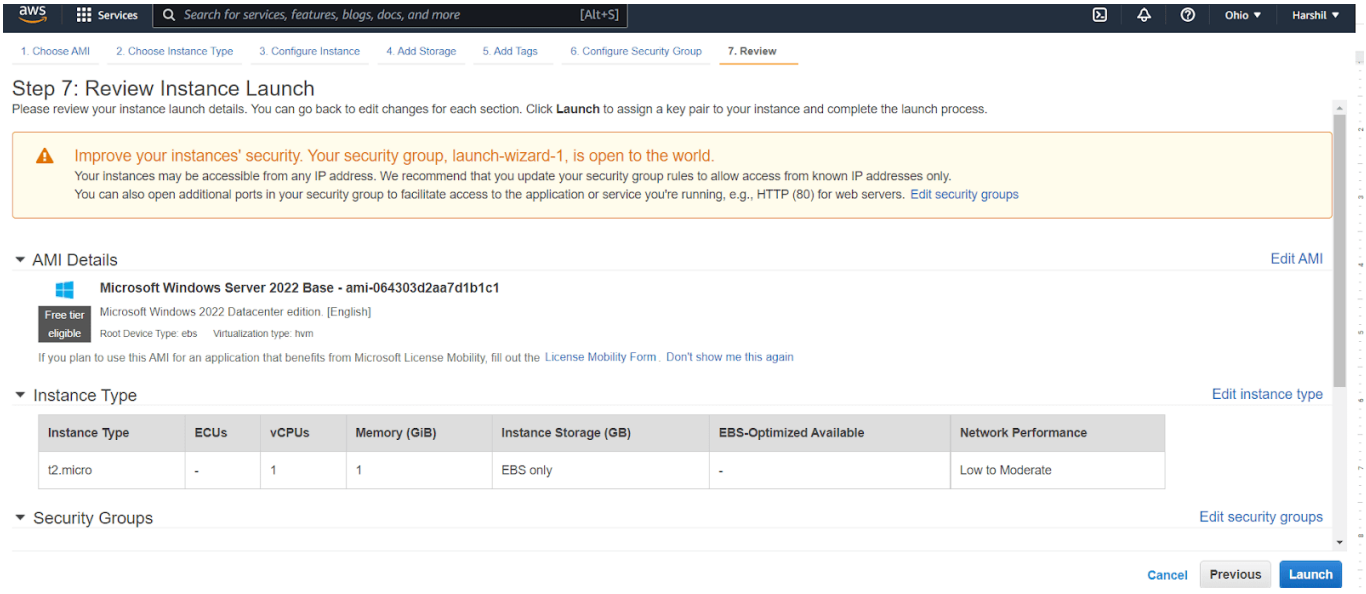

FIG 5.3.6 STEP 6

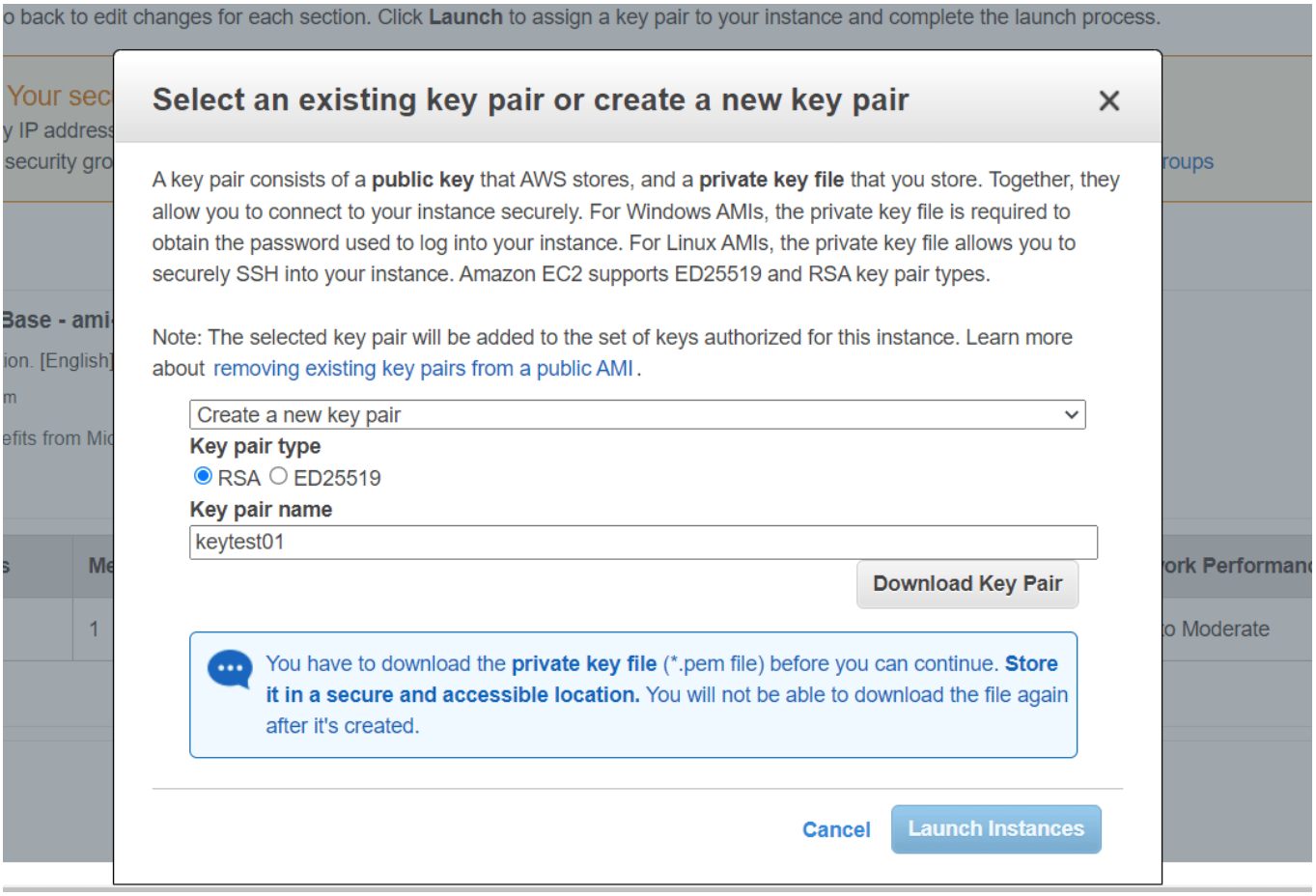

FIG 5.3.7 SELECT KEY PAIR

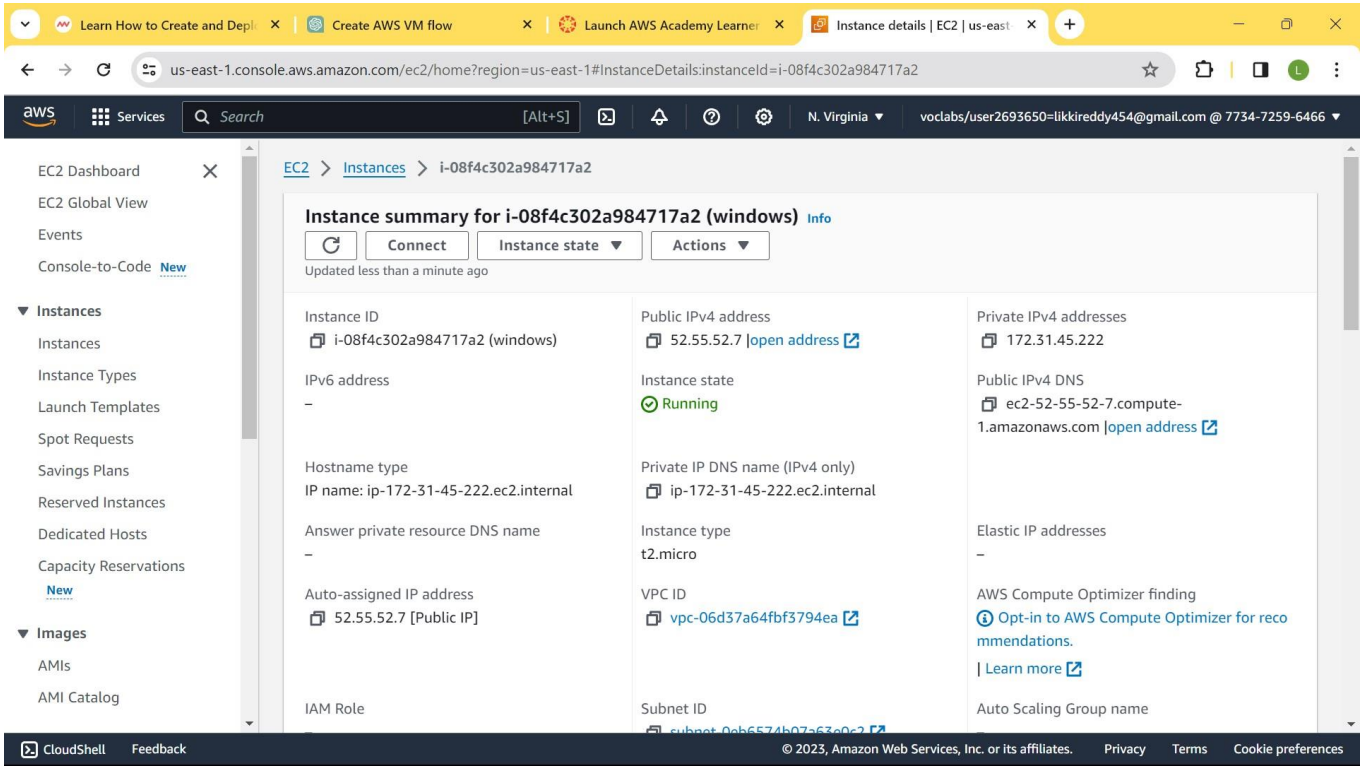

FIG 5.3.8 CHECKING INSTANCE

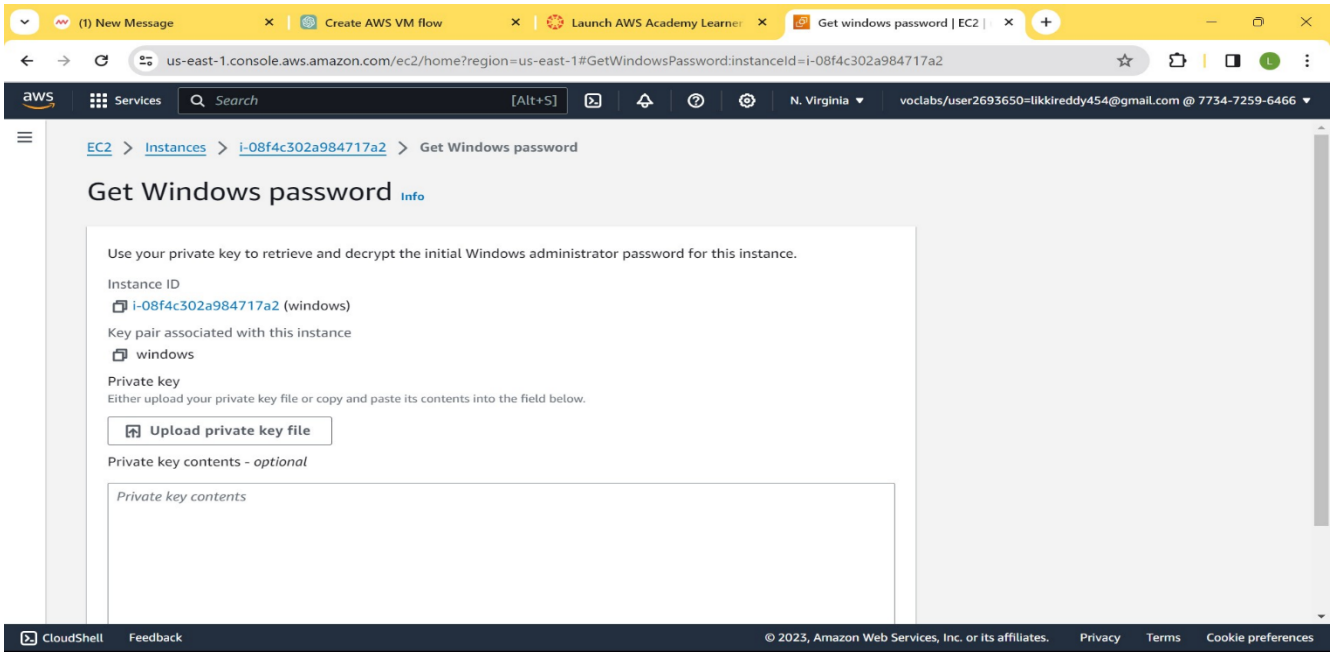

FIG 5.3.9 CREATING PASSWORD

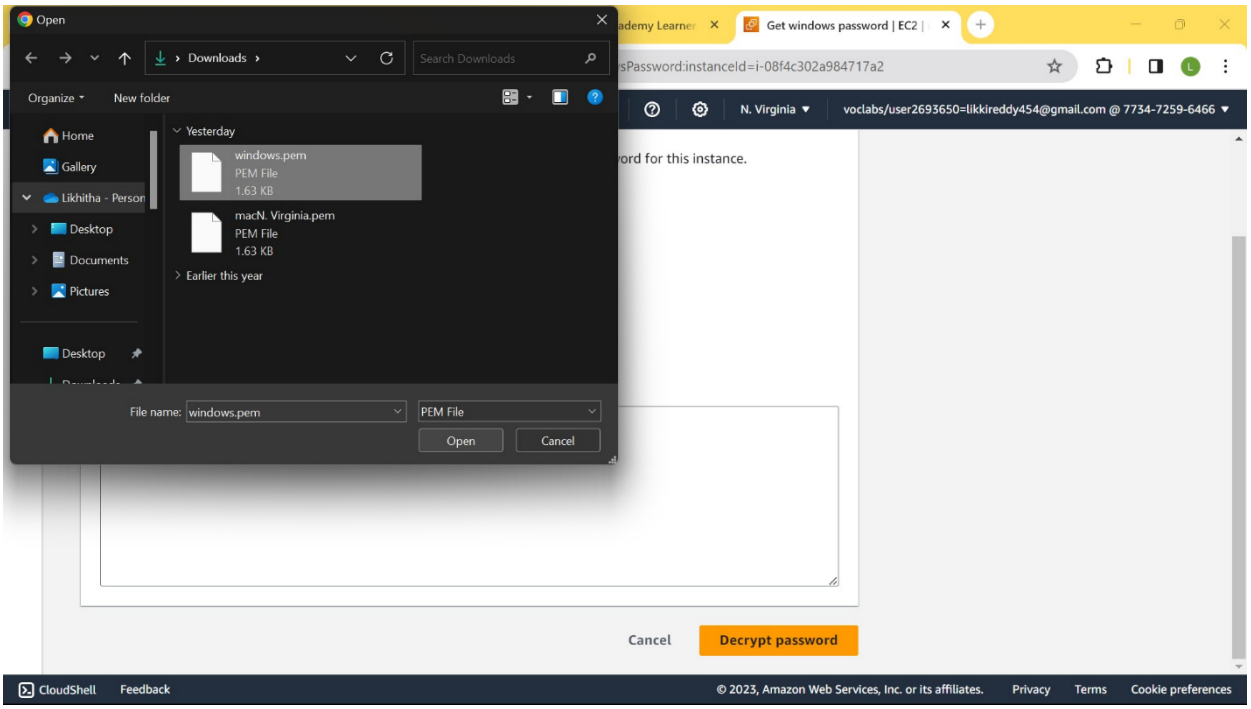

FIG 5.3.10 UPLOAD KEYPAIR FILE

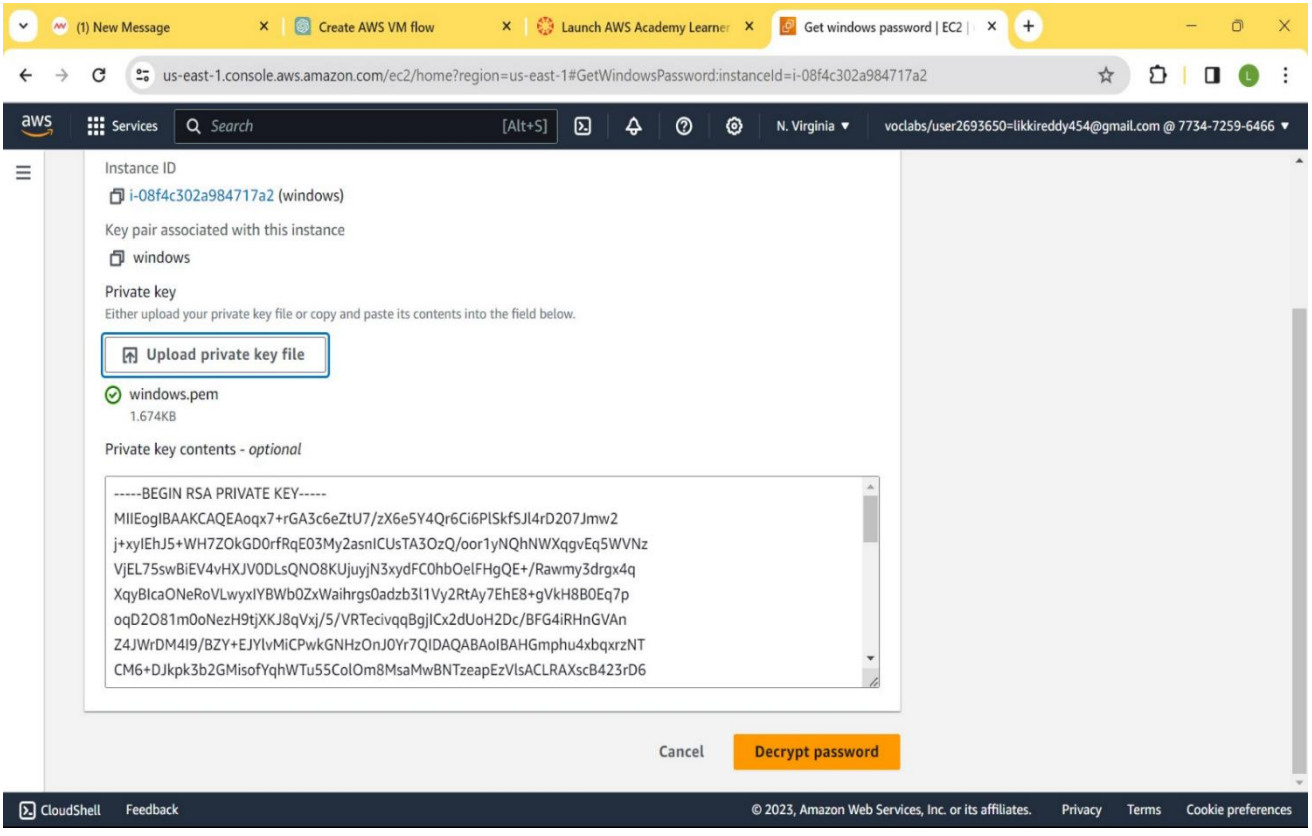

FIG 5.3.11 DECRYPT THE PASSWORD

Output Result

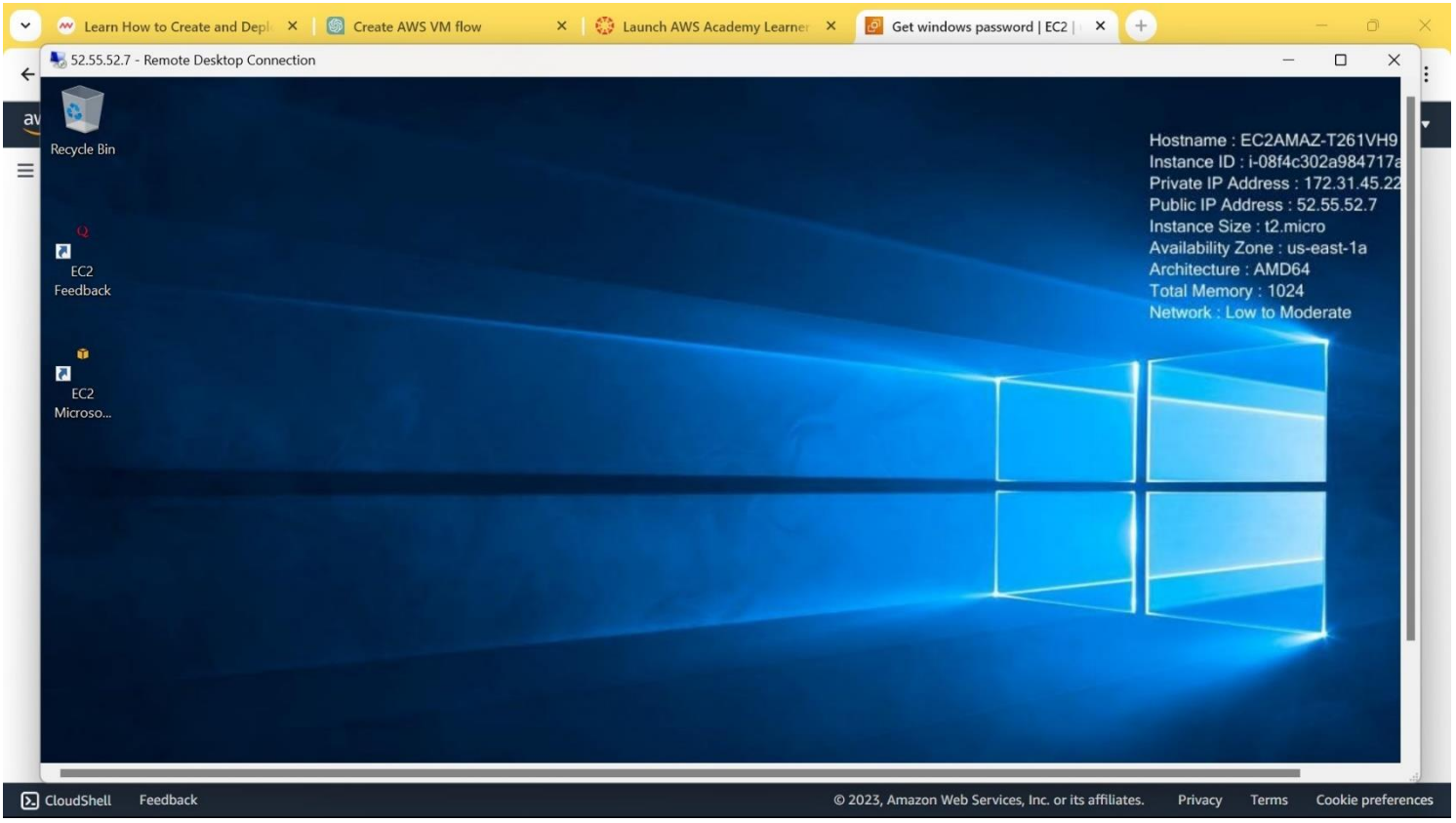

FIG 5.3.12 OUPUT SCREEN

5.2 Conclusion

In conclusion, the process of deploying and creating virtual machines (VMs) in AWS represents a structured and critical pathway towards establishing scalable and versatile computing resources. This process involves a series of deliberate steps, from selecting appropriate instance types and configuring networking settings to defining storage options and implementing stringent security measures. Through this methodical approach, organizations harness the power of AWS EC2 instances, gaining the agility to adjust computing capacity as needed while optimizing costs. The deployment process serves as the foundation for hosting applications, managing data, and accommodating diverse workloads within the AWS ecosystem. Continuous monitoring, optimization, and adherence to best practices in this process contribute to the creation of resilient, secure, and scalable cloud-based solutions, empowering businesses to innovate, adapt, and thrive in the dynamic landscape of cloud computing.

CHAPTER 6
TESTING AND VALIDATION

6.1 Introduction

Testing and validation are fundamental processes within any development or implementation cycle, serving as critical checkpoints to ensure the quality, functionality, and reliability of software, systems, or products. Testing involves systematically examining and evaluating various components, functionalities, or aspects of a product to identify and rectify defects, errors, or inconsistencies. On the other hand, validation confirms whether the end product meets the intended requirements, aligns with user expectations, and functions effectively within its environment. These processes encompass diverse methodologies, techniques, and tools aimed at uncovering issues, verifying functionalities, and validating performance, ultimately contributing to enhancing the overall quality and usability of the final deliverable. Effective testing and validation procedures not only improve product reliability but also instill confidence, mitigate risks, and ensure that the end product satisfies the desired specifications and end-user needs.

Purpose of Testing and Validation:

The purpose of testing and validation within any development or implementation process is multi-faceted, aiming to achieve several key objectives:

Ensure Quality: Testing and validation are crucial for ensuring that the developed software, system, or product meets predefined quality standards. It aims to identify and rectify defects, bugs, or issues that could affect performance, reliability, or user experience.

Verify Functionality: The primary purpose is to validate that the software or system performs its intended functions as per the specified requirements. It ensures that all features and functionalities work as expected.

Mitigate Risks: By systematically identifying and addressing issues, testing and validation help mitigate potential risks associated with deploying faulty or unreliable software. This includes risks related to security vulnerabilities, performance bottlenecks, or incorrect functionality.

6.2 Design Test Cases and Scenarios

Designing test cases and scenarios involves creating detailed plans outlining specific conditions, actions, and expected outcomes to validate the functionality, reliability, and performance of software or systems. Here's how to design test cases and scenarios:

Understand Requirements: Gain a comprehensive understanding of the software/system requirements to ensure test cases align with intended functionalities.

Identify Test Scenarios: Define various scenarios that represent different user interactions, system behaviors, or use cases. These scenarios should cover both normal and edge-case scenarios.

Create Test Cases: Break down each test scenario into individual test cases. Each test case should be specific, detailed, and cover a particular aspect or feature of the software/system.

Outline Test Steps: For each test case, list the steps required to execute the test, including preconditions, actions, inputs, and expected outcomes.

Incorporate Test Data: Include relevant test data or parameters required to execute the test effectively. This may involve varying inputs to validate different outcomes.

Consider Boundary Cases: Include boundary conditions and exceptional scenarios to ensure the system behaves correctly under various conditions, including both expected and unexpected scenarios.

Prioritize and Organize: Arrange test cases based on priority, grouping related cases and organizing them logically for efficient execution.

Review and Validation: Review test cases to ensure they cover all functionalities, are clear and executable, and align with the project requirements. Validate test cases with stakeholders or domain experts if needed.

6.3 Validation

Validation, in the context of software or system development, refers to the process of confirming that the end product meets the intended requirements and specifications set forth by stakeholders and users. It involves verifying that the developed software or system fulfills its intended purpose and functions correctly within its intended environment.

Key aspects of validation include:

Requirement Adherence: Ensuring that the software/system meets the specified requirements outlined by stakeholders, users, or business needs.

User Expectation Alignment:

Confirming that the software/system aligns with user expectations, preferences, and usability standards.

Functionality Confirmation:

Verifying that the software/system performs its intended functions accurately and effectively.

Performance Evaluation:

Assessing the software/system's performance metrics, including speed, responsiveness, and scalability, to ensure they meet predefined benchmarks.

Regulatory Compliance:

Validating that the software/system complies with relevant industry standards, regulations, and security protocols.

Conclusion

In conclusion, deploying and creating AWS Virtual Machines (VMs) is a fundamental process within the cloud computing realm, representing the initial step in establishing scalable, reliable, and versatile computing resources. This deployment involves a structured sequence of actions within the AWS Management Console or through automation tools, encompassing critical decisions such as selecting the appropriate instance type, configuring networking settings, defining storage options, and implementing stringent security measures. By leveraging AWS EC2 instances, organizations gain the flexibility to adjust computing capacity based on demand, fostering agility and cost-efficiency. The deployment process signifies not only the creation of virtual computing resources but also marks the foundation for hosting applications, managing data, and supporting diverse workloads, underpinning the infrastructure that powers modern digital operations. Continuous monitoring, optimization, and adherence to best practices in deployment contribute significantly to robust, secure, and scalable cloud-based solutions that drive innovation and support business objectives within the AWS ecosystem.

CHAPTER 7
CONCLUSION

7.1 Conclusion

In wrapping up our exploration of hosting a static website on AWS through Amazon S3, we've achieved a remarkable feat by seamlessly blending the power of cloud infrastructure with the elegance of static web content. Our project showcases the efficiency and versatilityof AWS services in facilitating web hosting while emphasizing the importanceof continuous learning and expansion. Throughout this journey, we've harnessed the capabilities of AWS tocreate a robust and costeffective hosting solution. Our static website, composed of HTML, CSS, and JavaScript, effortlessly resides within an S3 bucket. This process illuminated the straightforward yet impactful approach AWS offers for deploying web content. The benefits are numerous: from global scalability to reliable availability, we'veseen how AWS enables us to present our digital creations to the world with confidence. Reflecting on our experiences, we've gained practical insights into the steps involved in configuring S3 for website hosting. From organizing files to defining index documents and permissions, we'venavigated the intricacies of cloud-based web deployment. Witnessing thetangible result of our efforts—our website accessible via its designated S3 endpoint— reinforces the achievement and deepens our appreciation for this technology. As we conclude, we're motivated to encourage further exploration. AWS provides an expansive landscape for web development and beyond. Beyond hosting, AWS offers tools for enhancing website performance through content delivery networks (CDNs) and scaling with serverless architecture. Venturing into the world of databases, authentication, and dynamic functionalitywith AWS services like Amazon DynamoDB and Amazon Cognito can elevateyour projectsto new heights. By completing this project, you've taken a significant stride toward mastering AWS's capabilities for web hosting and demonstrated an enthusiasm for innovation. As youcontinue to learn and apply AWS services, you're poised to create future projects that harnessthe full potential of cloud-based solutions. We hope this journey serves as a stepping stone to your ongoing success in the dynamic realm of web development with AWS

REFERENCES

1. Amazon Management Console. http://aws.amazon.com/codedeploy/
2. ^ "Cloud computing with AWS". aws.amazon.com. March 1, 2021. Archived from the original on May 26, 2022. Retrieved March 3, 2021.
3. Bergmayr, A., J. Troya, P. Neubauer, M. Wimmer, and G. Kappel: UML-based Cloud Application Modeling with Libraries, Profiles, and Templates. Proceedings of the 2nd International Workshop on Model-Driven Engineering on and for the Cloud (CloudMDE 2014). http://ceur-ws.org/ Vol. 242/paper7.pd.
4. Cloud Standards Customer Council (2017): Practical Guide to Cloud Computing, V3.0. https://www.omg.org/cloud/deliverables/practical-guide-tocloud-computing.
5. ^ Mosca, David (April 14, 2021). "Jersey City's ElectrifAi a leader in artificial intelligence software for business". nj. Archived from the original on April 30, 2022. Retrieved April 21, 2021.
6. ^ "Infographic: Amazon Leads $100 Billion Cloud Market". Statista Infographics. Archived from the original on January 4, 2021. Retrieved October 12, 2020.
7. Azure, Google, IBM". ChannelE2E. Archived from the original on January 10, 2021. Retrieved October 12, 2020.
8. ^ "Top 10 AWS Services according to popularity". medium.com. August 31, 2019. Archived from the original on October 12, 2020. Retrieved October 5, 2020.
9. Windriver.https://events.windriver.com/wrcd01/wrcm/2016/08/WPcybersecurity-and-secure-deployments.pdf.
10. "What is Cloud Computing by Amazon Web Services | AWS". Archived from the original on December 25, 2018. Retrieved July 17, 2013.