REPORT ON

FIFTEEN WEEKS OF INTERNSHIP

Carried out at

AWS CLOUD COMPUTING FROM MICRO DEGREE

Submitted to

NMAM INSTITUTE OF TECHNOLOGY, NITTE

(An Autonomous Institution affiliated to VTU, Belagavi)

In partial fulfillment of the requirements for the award of the

Degree of Bachelor of Engineering
in
Electronics and Communication Engineering

by

JEEVAN H R

4NM22EC404

Under the guidance of

Mr. Shreedhar M.N

Senior Cloud Architect

(MicroDgree)

NITTE EDUCATION TRUST | N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India

NITTE EDUCATION TRUST | N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India

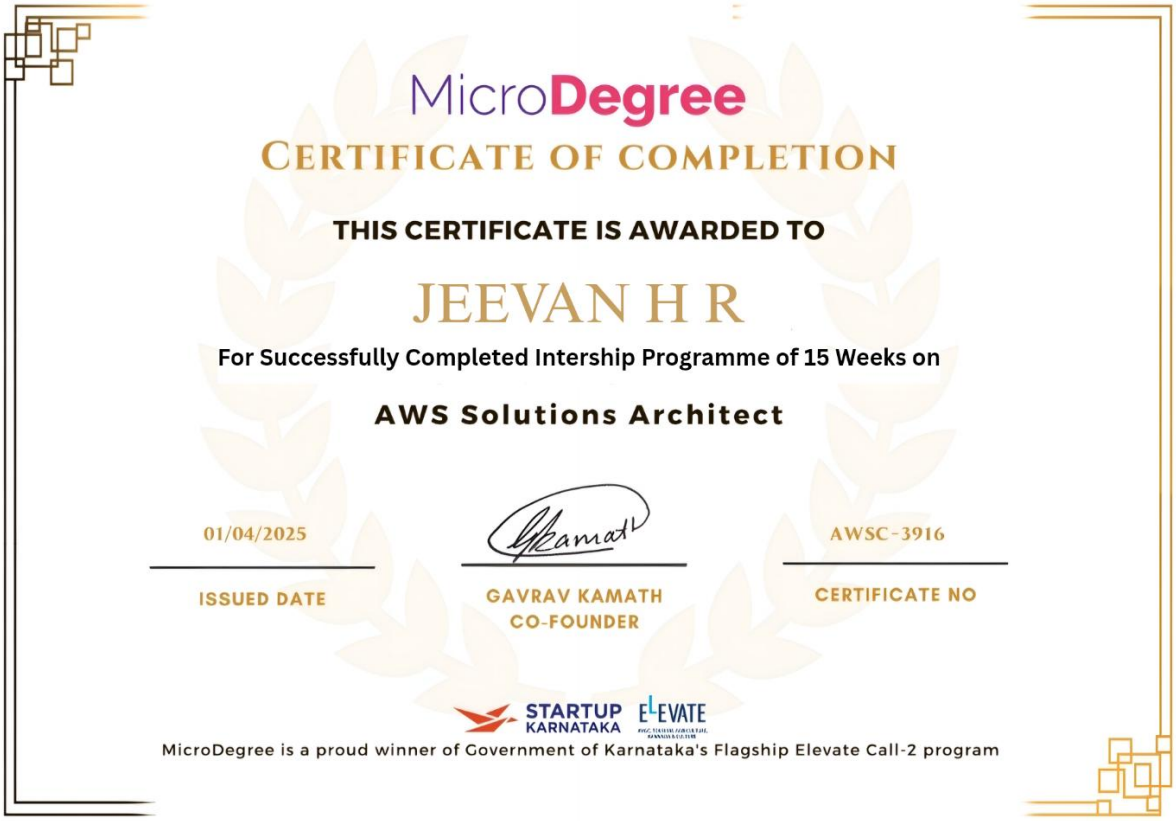
CERTIFICATE

This is to certify that the "Internship report" submitted by Mr. JEEVAN H R bearing USN 4NM22EC404 *of* 8^{th} *semester B.E., a bonafide student of NMAM Institute of Technology, Nitte, has undergone fifteen weeks of internship at AWS Cloud Computing from Micro Degree from December 2024 to April 2025 fulfilling the partial requirements for the award of degree of Bachelor of* ***Engineering in Electronics and Communication Engineering*** *at NMAM Institute of Technology, Nitte.*

Name and Signature of Mentor

Signature of HOD

INDUSTRY CERTIFICATE



Acknowledgement

I would like to express my sincere gratitude to Mr. Shreedhar M.N, Senior Cloud Architect of MicroDegree for providing me the opportunity to complete the AWS Cloud Internship program. This internship has greatly enhanced my understanding of cloud computing concepts, AWS services, and their practical applications

Throughout this internship, I was able to apply theoretical knowledge in real-world scenarios, work on practical projects, and gain a solid understanding of AWS services and cloud computing. The structured guidance, mentorship, and continuous support provided by the MicroDegree team were instrumental in my learning journey. This hands-on exposure has been invaluable in enhancing my technical skills and confidence in cloud technologies.

Furthermore, I extend my heartfelt appreciation to Dr. Niranjan N. Chiplunkar, Principal of N. M. A. M Institute of Technology, for providing us with an excellent academic environment and the necessary resources to successfully complete this endeavor.

My sincere thanks to Dr. K. V. S. S. S. S. Sairam, Professor and Head of the Department of Electronics and Communication Engineering, N. M. A. M Institute of Technology, for his continuous motivation, support, and valuable inputs that have significantly contributed to my progress.

I am deeply grateful to my guide, Dr. Durga Prasad, Professor and Head of the Department of Advanced Communication Technology, N. M. A. M Institute of Technology, for his constant mentorship, technical guidance, and encouragement throughout the course of this project. His expertise and constructive feedback have played a crucial role in refining my work.

Finally, I extend my gratitude to all faculty members, colleagues, and everyone who directly or indirectly contributed to the completion of this internship experience.

Table of Contents

List of Figures

Abstract

The AWS Cloud Computing Internship provides hands-on experience in working with Amazon Web Services (AWS), one of the leading cloud computing platforms globally. Interns are introduced to the fundamental concepts of cloud infrastructure, including computing, storage, networking, and database services, using AWS technologies. Throughout the internship, participants will work on real-world projects, helping companies migrate their systems and applications to the cloud, optimize existing workloads, and implement best practices for scalability, security, and performance.

Interns will gain expertise in AWS services like EC2 (Elastic Compute Cloud), S3 (Simple Storage Service), RDS (Relational Database Service), Lambda, VPC (Virtual Private Cloud), and CloudFormation. They will also learn how to deploy and manage cloud resources, monitor performance, ensure security compliance, and apply cost optimization strategies using AWS tools.

The internship provides an opportunity to learn about the architecture of cloud-native applications and how to effectively integrate and automate cloud solutions. Participants will gain valuable insights into the cloud computing industry, working with industry professionals, and understanding the key role of cloud infrastructure in supporting modern business applications.

By the end of the internship, participants will have developed practical skills in cloud architecture, problem-solving, and cloud services management, positioning them for future roles in the rapidly expanding field of cloud computing.

Chapter 1

INRODUCTION TO THE INDUSTRY

I am completed my internship on Cloud Computing in MicroDegree. This internship designed to introduce and aspiring professionals to the exciting world of cloud technology. Cloud computing has revolutionized the way businesses and individuals access technology, offering flexible, scalable, and cost-effective solutions across industries. The Cloud Computing Internship Microdegree is crafted to provide participants with a deep understanding of cloud fundamentals, including service models like Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Through a combination of theoretical knowledge and hands-on experience with platforms such as AWS, Microsoft Azure, and Google Cloud, this internship ensures that learners are equipped with the practical skills necessary to design, manage, and deploy cloud-based solutions.

This internship not only enhances technical proficiency but also prepares participants for real-world challenges in the cloud domain. By working on live projects, case studies, and industry scenarios, interns will bridge the gap between academic learning and professional expectations. Whether you are a student, a recent graduate, or a professional looking to upskill, this microDegree offers a strong foundation to kickstart a successful career in cloud computing, one of the fastest-growing sectors in the tech industry.

CHAPTER 2

DETAILS OF THE TRAINING UNDERGONE

2.1 Introduction to Cloud Computing:

In today's digital world, cloud computing has become the backbone of how individuals and businesses use technology. It allows users to access computing resources such as servers, storage, databases, networking, software, and analytics over the internet, without the need to own or manage physical hardware.

Cloud computing provides on-demand access to a wide range of services, offering flexibility, scalability, and cost-efficiency. Instead of making large investments in data centers and servers, users can simply rent computing power and storage from cloud providers like Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP).

Fig.2.1 Cloud Service providers.

2.2 Introduction Amazon Web Services (AWS) and its Services:

Amazon Web Services (AWS) is the world's most comprehensive and widely adopted cloud platform, offering over 200 fully featured services from data centers globally. Launched by Amazon in 2006, AWS provides on-demand cloud computing resources such as computing power, storage, databases, networking, machine learning, analytics, and much more. It enables businesses of all sizes to move faster, lower IT costs, and scale applications with ease. Today, millions of customers including startups, enterprises, and government organizations trust AWS to power their infrastructure and drive innovation.

Benefits of AWS Compared to Other Tools

- Scalability: Easily scale resources up or down based on demand.
- Cost-Effective: Pay only for what you use with no upfront investments.
- High Security: Advanced security measures to protect data and applications.
- Global Reach: Access AWS services from data centers around the world.

AWS Services

Here are some of the most widely used AWS services:

- **Amazon EC2 (Elastic Compute Cloud)**: Virtual servers in the cloud to run applications.
- **Amazon S3 (Simple Storage Service)**: Secure, scalable object storage for data, backups, and applications.
- **AWS Lambda**: Run code without provisioning or managing servers (serverless computing).
- **Amazon RDS (Relational Database Service)**: Managed database service for MySQL, PostgreSQL, Oracle, and others.
- **AWS CloudFront**: Content delivery network (CDN) for faster distribution of data globally.
- **Amazon VPC (Virtual Private Cloud)**: Securely isolate and manage network environments.
- **AWS IAM (Identity and Access Management)**: Manage user permissions and security policies.

2.2.1 Creating AWS Account:

There are few steps to involved to create AWS account are mentioned below,

---

1. Go to the AWS website:

- Visit https://aws.amazon.com/.

2. Click "Create an AWS Account":

- It's usually at the top right corner.

3. Enter your Email and Password:

- Provide a working email address.
- Create a strong password.
- Set your AWS account name (this can be anything, like "MyCompany" or "MyName").

4. Provide Contact Information:

- Choose whether the account is for Personal or Business use.
- Enter your name, address, and phone number.

5. Payment Information:

- Add a credit card or debit card (they require this even if you're just using the free tier).
- Don't worry they won't charge you immediately. You'll get access to AWS Free Tier, which gives you free resources for 12 months (up to certain limits).

6. Identity Verification:

- AWS will ask to verify your phone number.
- They'll send you a code via text or call.

7. Choose a Support Plan:

- Select the free "Basic" support plan unless you know you need premium support.

8. Sign in to the AWS Console:

- After everything is set up, you can log into the AWS Management Console and start using AWS services.

2.2.2 Amazon EC2 (Elastic Compute Cloud):

Fig. 2.2 Icon of Amazon EC2

By using AWS EC2 helps users to avoid the investment in hardware up front, so the user can deploy and develop applications easier. It is used to launch many virtual servers, configure networking and security, and managing storage.

Task: Launch Your Amazon EC2 Instance

In this task, you will launch an Amazon EC2 instance with *termination protection*. Termination protection prevents you from accidentally terminating an EC2 instance. You will deploy your instance with a User Data script that will allow you to deploy a simple web server.

Step 1: Choose an Amazon Machine Image (AMI)

An Amazon Machine Image (AMI) provides the information required to launch an instance, which is a virtual server in the cloud. An AMI includes:

A template for the root volume for the instance (for example, an operating system or an application server with applications)

The Quick Start list contains the most commonly-used AMIs. You can also create your own AMI or select an AMI from the AWS Marketplace, an online store where you can sell or buy software that runs on AWS.

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of *instance types* optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. Each instance type includes one or more *instance sizes*, allowing you to scale your resources to the requirements of your target workload.

Step 3: Configure Instance Details

The Network indicates which Virtual Private Cloud (VPC) you wish to launch the instance into. You can have multiple networks, such as different ones for development, testing and production.

The Lab VPC was created using an AWS CloudFormation template during the setup process of your lab. This VPC includes two public subnets in two different Availability Zones.

Scroll down, then expand **Advanced Details**.

A field for **User data** will appear.

When you launch an instance, you can pass *user data* to the instance that can be used to perform common automated configuration tasks and even run scripts after the instance starts.

Your instance is running Amazon Linux, so you will provide a *shell script* that will run when the instance starts.

Copy the following commands and paste them into the **User data** field:

yum -y install httpd

systemctl enable httpd

systemctl start httpd

echo '<html><h1>Hello From Your Web Server!</h1></html>' > /var/www/html/index.html

The script will:

- Install an Apache web server (httpd)
- Configure the web server to automatically start on boot
- Activate the Web server
- Create a simple web page

Step 4: Add Storage

Amazon EC2 stores data on a network-attached virtual disk called *Elastic Block Store*.

You will launch the Amazon EC2 instance using a default 8 GiB disk volume. This will be your root volume (also known as a 'boot' volume).

Step 5: Configure Security Group

A *security group* acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You add *rules* to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances that are associated with the security group.

On **Step 5: Configure Security Group**, configure:

- **Security group name:** Web Server security group
- **Description:** Security group for my web server

In this lab, you will not log into your instance using SSH. Removing SSH access will improve the security of the instance.

Delete the existing SSH rule.

Step 6: Review Instance Launch

The Review page displays the configuration for the instance you are about to launch.

Choose Launch

A **Select an existing key pair or create a new key pair** window will appear.

Amazon EC2 uses public–key cryptography to encrypt and decrypt login information. To log in to your instance, you must create a key pair, specify the name of the key pair when you launch the instance, and provide the private key when you connect to the instance.

Wait for your instance to display the following:

- **Instance State:** running
- **Status Checks:** 2/2 checks passed

We have successfully launched your first Amazon EC2 instance.

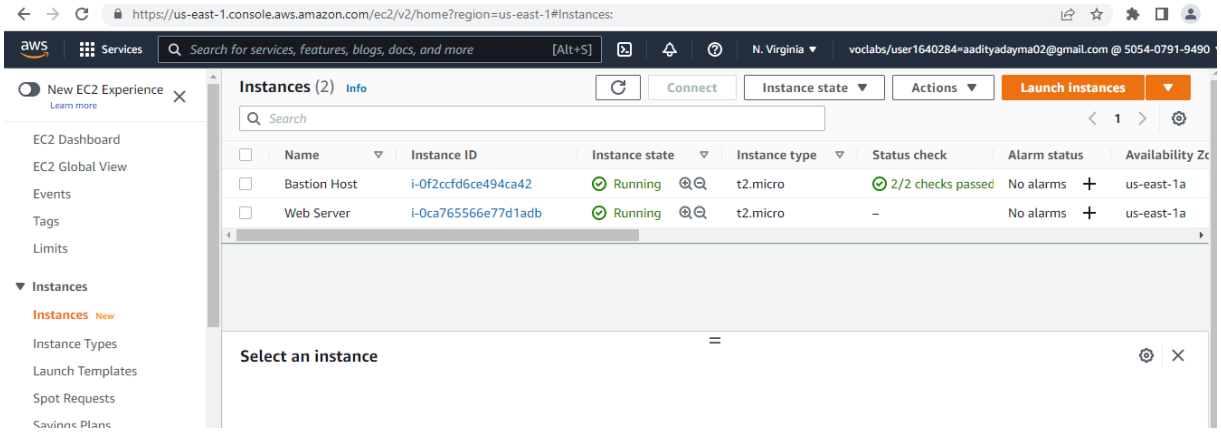


Fig 2.3 Overview of running Instances

2.2.3 Amazon S3 (Simple Storage Service):

Fig :2.4 Icon of Amazon S3

Amazon Simple Storage Service (Amazon S3) is a cloud-based object storage service offered by Amazon Web Services (AWS). It provides scalable, secure, and durable storage for a wide range of data types and use cases, including backups, data lakes, websites, mobile applications, and big data analytics

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. Customers of all sizes and industries can use Amazon S3 to store and protect any amount of data for a range of use cases, such as data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics. Amazon S3 provides management features so that you can optimize, organize, and configure access to your data to meet your specific business, organizational, and compliance requirements.

Key Benefits of Amazon S3:

- **Scalability**: You can store unlimited data without worrying about running out of space.
- **Durability**: Data stored in S3 is designed for 99.99% (11 nines!) durability.
- **Availability**: Very high uptime — your data is almost always accessible.
- **Security**: Built-in encryption, access control (IAM, bucket policies), and compliance support.
- **Cost-effective**: You only pay for what you use, with different storage classes for hot/cold/archive data.
- **Performance**: Fast uploads/downloads, especially when integrated with other

AWS services (like CloudFront for CDN).

- **Backup and Restore**: Easy to use for backup solutions, disaster recovery, and archiving.
- **Data Management**: Features like versioning, lifecycle rules, cross-region replication, and object locking.
- **Integration**: Works smoothly with analytics (like AWS Athena), machine learning, serverless apps, and more.

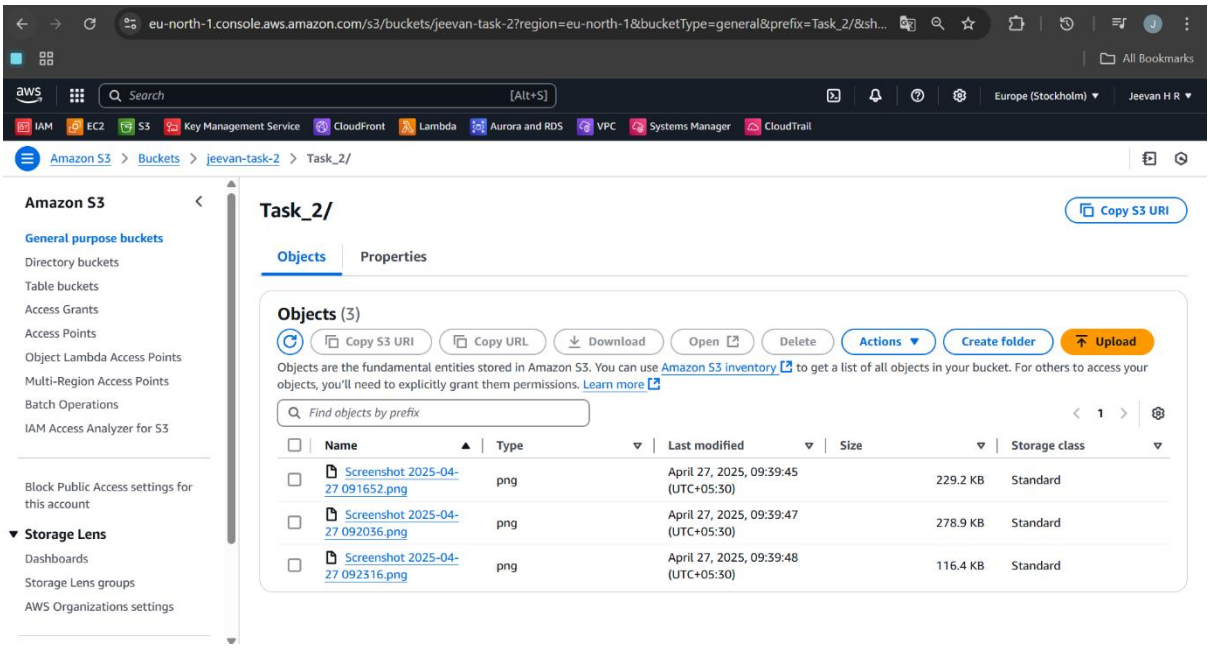


Fig.2.5 Overview of created S3 Bucket

2.2.4 AWS Lambda

Fig.2.6 Icon of AWS Lamda

AWS Lambda is a serverless compute service that automatically runs your code in response to events and triggers, such as changes to data in Amazon S3 buckets or updates to a database. With AWS Lambda, you can run code without provisioning or managing servers. Lambda executes code only when needed, and scales automatically by running code in parallel based on incoming requests or events.

AWS Lambda follows a serverless, event-driven methodology that allows developers to run code without managing the underlying infrastructure. Lambda functions are triggered by events from various AWS services such as S3, DynamoDB, API Gateway, and CloudWatch, or external sources. This approach is ideal for use cases that require lightweight, stateless computing tasks. Lambda functions are executed in response to these events and can scale automatically to handle varying workloads without manual intervention. The serverless nature of Lambda means you only pay for the compute time your code consumes, making it highly cost-effective. It supports a microservices architecture where each function can perform a specific task and integrate with other services through events. Lambda also ensures high availability and fault tolerance by automatically managing the infrastructure across multiple Availability Zones. With its focus on short-duration tasks (up to 15 minutes per invocation), AWS Lambda is particularly suited for real-time data processing, automation, and backend services for web applications.

2.2.5 Amazon RDS (Relational Database Service):

Fig. 2.7 Icon of Amazon RDS

Amazon Relational Database Service (RDS) is a fully managed relational database service provided by AWS that simplifies the setup, operation, and scaling of relational databases in the cloud. It supports several popular database engines, including MySQL, PostgreSQL, Oracle, SQL Server, and Amazon Aurora. RDS automates administrative tasks such as hardware provisioning, database setup, patching, and backups, allowing developers to focus on building and deploying applications without managing the complexities of traditional database administration. RDS also provides high availability, automatic failover, and robust security features, making it an ideal choice for production-level applications that require reliable and scalable relational database solutions.

2.2.6 AWS CloudFront:

Fig. 2.8 Icon of AWS CloudFront

Amazon CloudFront is a content delivery network (CDN) service provided by AWS that accelerates the delivery of websites, APIs, video content, and other web assets

to users worldwide. CloudFront integrates with other AWS services such as S3, EC2, and Elastic Load Balancing to distribute content with low latency and high transfer speeds. By caching content at edge locations close to the users, CloudFront reduces the load on the origin server and minimizes the latency experienced by end users when accessing the content. Whether it's delivering static content like images and videos or dynamic content like API responses, CloudFront is designed to optimize the performance and availability of web applications, making it an essential service for delivering fast and secure content to a global audience.

Content Delivery Network (CDN) :

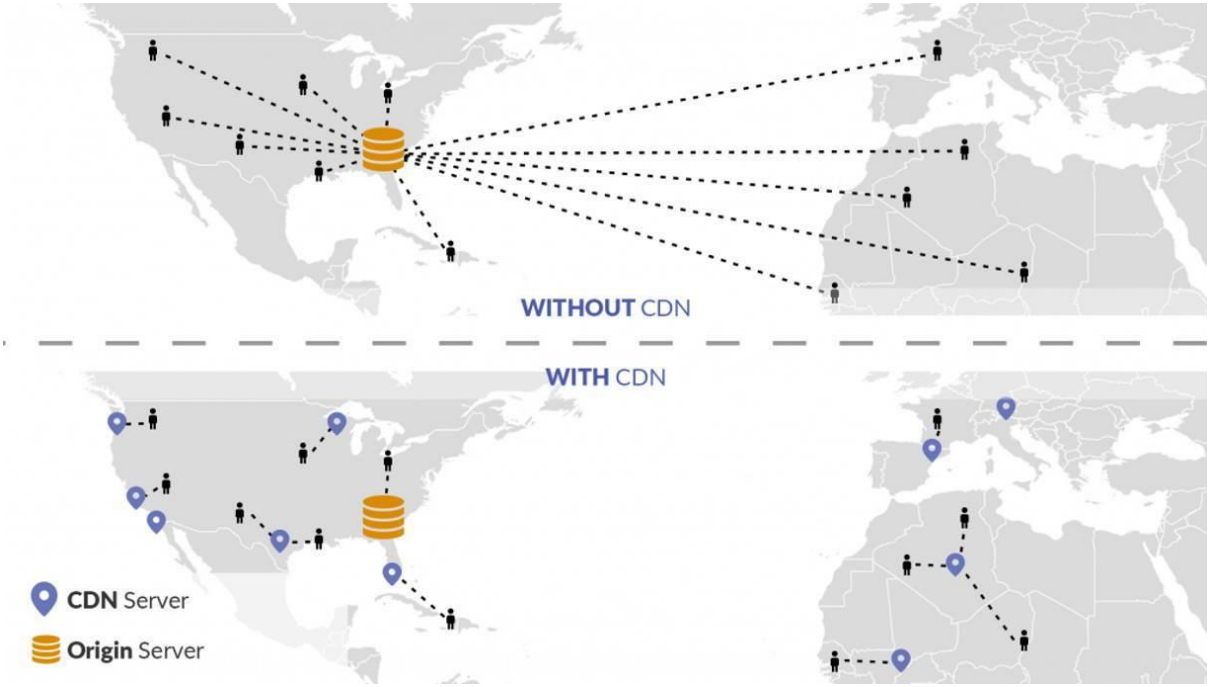


Fig.2.9 Working of Content Delivery Network (CDN)

AWS CloudFront is a powerful CDN service designed to accelerate the delivery of both static and dynamic content, ensuring fast, reliable, and secure access for users globally. Whether for websites, APIs, or media content, CloudFront provides high performance and scalability at a cost-effective price.

2.2.7 Amazon VPC (Virtual Private Cloud):

Fig.2.10 Icon of Amazon VPC

Amazon Virtual Private Cloud (Amazon VPC) is a service that allows users to create a logically isolated network within the AWS Cloud. This private network allows users to launch AWS resources such as EC2 instances, databases, and applications in a secure, customizable environment. A VPC offers complete control over the network configuration, including IP address range, subnets, route tables, and network gateways, giving users the flexibility to create and manage a network that fits their specific needs.

With Amazon VPC, you can set up a private network that closely resembles a traditional on-premises data center, but with the benefits of scalability, flexibility, and security offered by AWS. VPC enables users to extend their on-premises network into the cloud, providing secure and reliable communication between AWS resources and on-premises systems. It also allows integration with other AWS services, providing a secure foundation for running your cloud applications

Components of Amazon VPC

1. **VPC**: A virtual private network where your resources (EC2 instances, databases, etc.) are placed.
2. **Subnets**: Segments of the VPC that divide the network into smaller, manageable portions. Subnets can be public (accessible from the internet) or private (not accessible from the internet).
3. **Route Tables**: Define how traffic is routed between subnets, the internet, and other networks. Custom route tables allow for fine-grained control of traffic flow.

4. **Internet Gateway**: A gateway that allows communication between resources in a VPC and the internet. It is used to provide internet access to instances in public subnets.
5. **Virtual Private Gateway**: A gateway used to connect your VPC to your on-premises network via VPN, enabling secure communication between your cloud resources and on-premises systems.
6. **NAT Gateway**: Used to allow instances in private subnets to access the internet while preventing inbound internet traffic to those instances.
7. **Elastic IP Address**: A static IP address designed for dynamic cloud computing. It is associated with your AWS account and can be moved between instances as needed.
8. **Security Groups**: Virtual firewalls for EC2 instances that control inbound and outbound traffic at the instance level.

2.2.8 AWS IAM (Identity and Access Management):

Fig.2.11 Icon of AWS IAM

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS services and resources for your users. IAM allows you to manage who can access your resources in AWS, what actions they can perform on those resources, and from where they can perform those actions. It is a critical service for enforcing security best practices and ensuring that only authorized users and applications have the appropriate level of access to your AWS environment. With IAM, you can create and manage AWS users, groups, and roles, as well as configure permissions for these identities. IAM provides a centralized way to manage access to your AWS resources, making it easier to ensure that your cloud environment is secure, organized, and compliant.

AWS IAM (Identity and Access Management) policies are used to define permissions for actions on AWS resources. A policy is a JSON document that specifies which actions are allowed or denied on which resources. IAM policies help AWS administrators define who can access resources and what actions they can perform. These policies are attached to IAM users, groups, or roles and govern the level of access granted.

Types of IAM Policies

1. **Managed Policies**:
    - **AWS Managed Policies**: Predefined policies provided by AWS to simplify access control. These policies are maintained and updated by AWS.
    - **Customer Managed Policies**: Custom policies created by you to fine-tune access control based on your specific needs.
2. **Inline Policies**: These policies are embedded directly into a specific user, group, or role. Inline policies are tightly coupled with the identity they are attached to, unlike managed policies that can be reused across different identities.

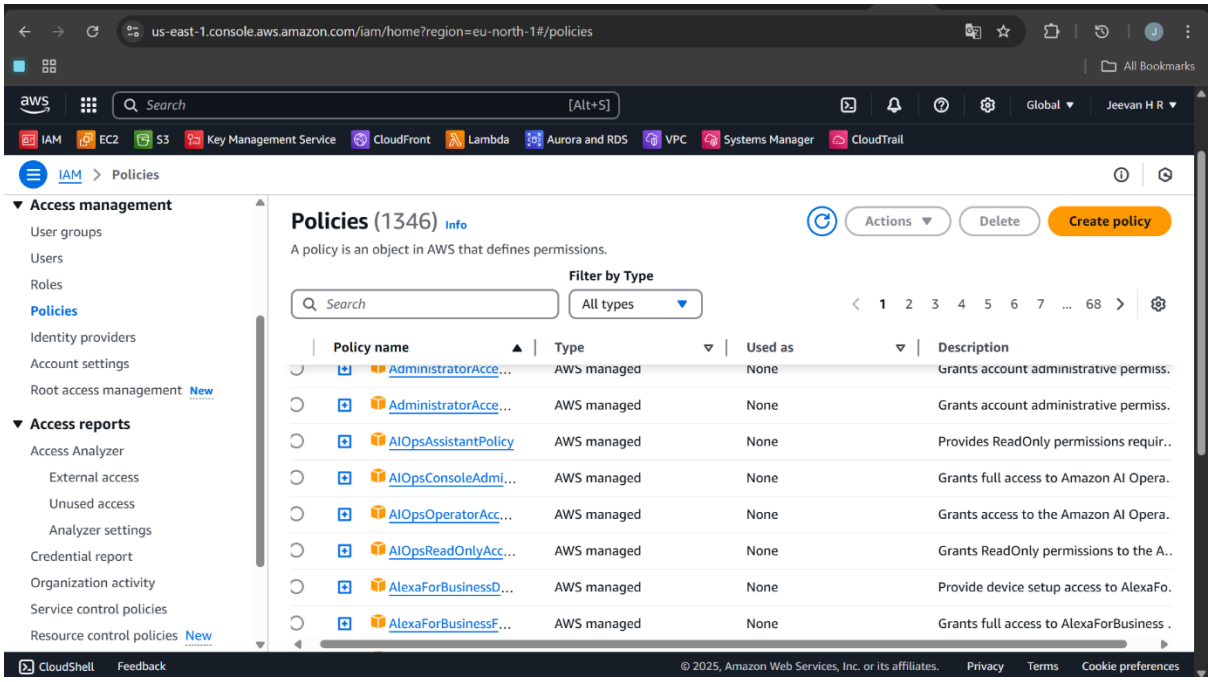


Fig.2.12 Overview of AWS IAM Policies

2.2.9 About Projects

Statement 1:

Create a VPC with 2 subnets one public and the other private in different Availability Zones(AZ's) and attach Internet gateway.

Solution:

To create a VPC with 2 subnets (one public, one private) across different Availability Zones (AZs), first create a VPC with a CIDR block like 10.0.0.0/16. Then, create two subnets: one public subnet (e.g., 10.0.1.0/24) in AZ-1 and one private subnet (e.g., 10.0.2.0/24) in AZ-2. Create an Internet Gateway (IGW) and attach it to the VPC. For the public subnet, create a route table that routes 0.0.0.0/0 traffic to the IGW and associate the public subnet with this route table.

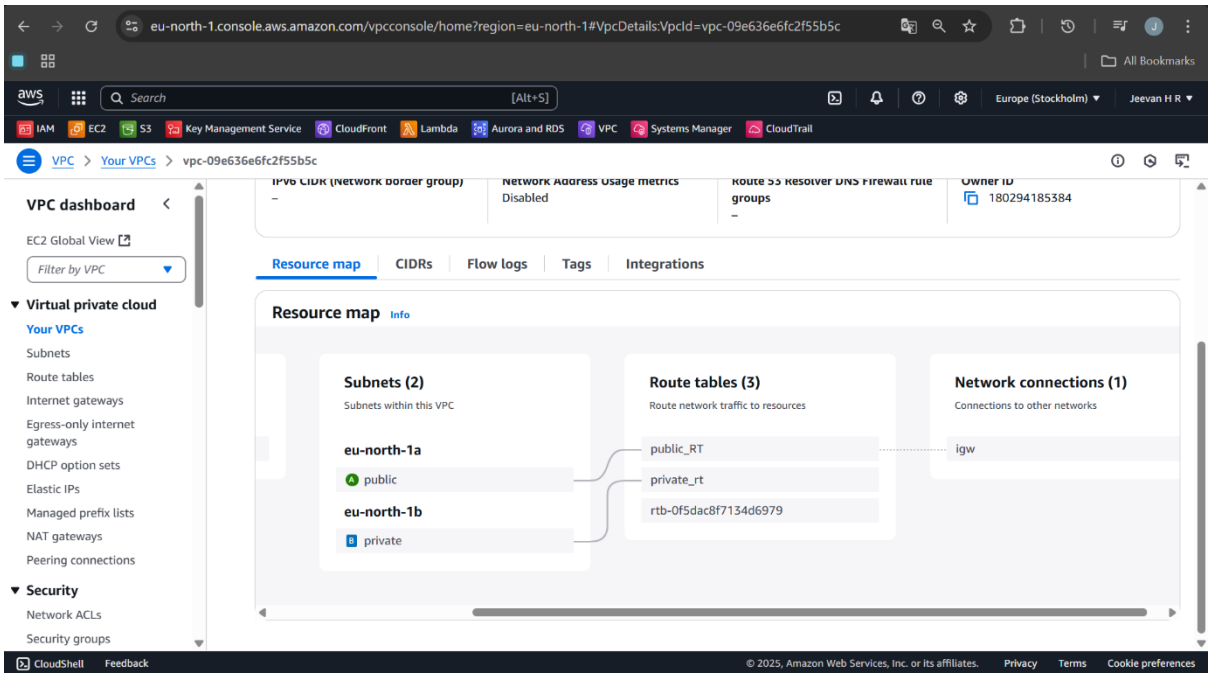


Fig.2.13 Configuration of VPC

The private subnet will use the default route table or a private route table (no direct Internet access). Optionally, you can add a NAT Gateway in the public subnet if the private subnet needs outbound internet access.

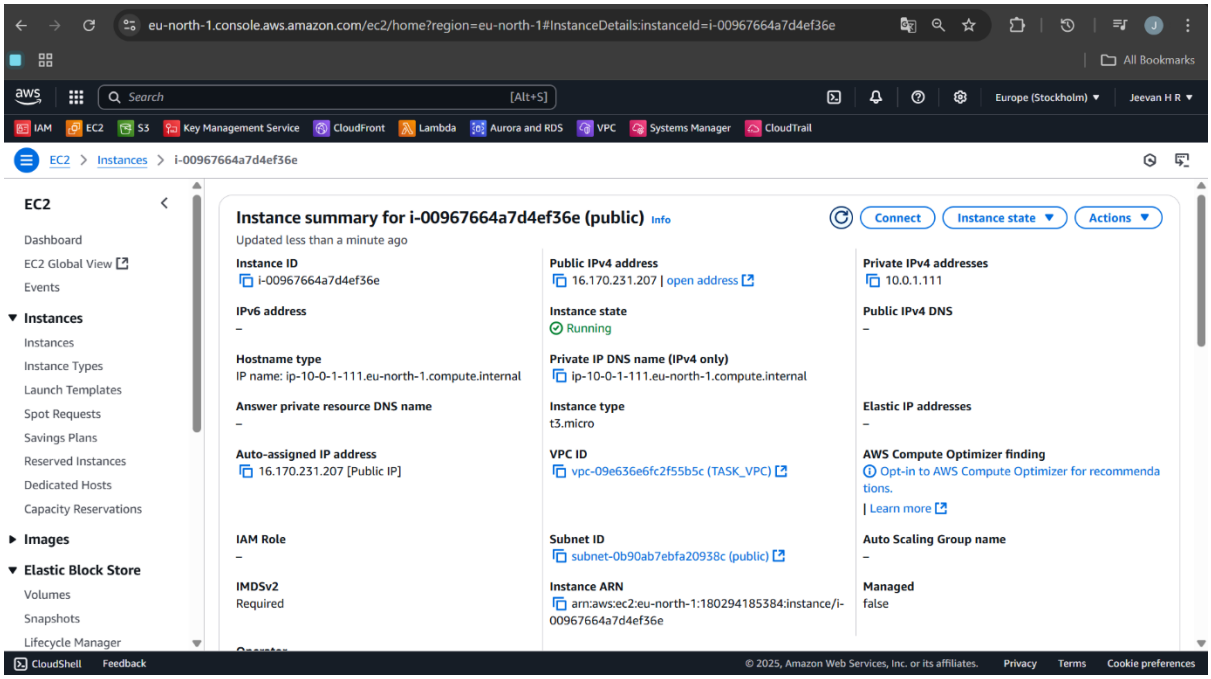


Fig.2.14 Launching Instance in the VPC bound

Statement 2:

Create a basic web-based calculator (Add, Subtract, Multiply, Divide) hosted completely in the cloud.

Solution:

In this project, we will create a basic web-based calculator that can perform simple mathematical operations such as Addition, Subtraction, Multiplication, and Division. The calculator will be built using HTML, CSS, and JavaScript for the frontend, and it will be hosted entirely in the cloud using Amazon Web Services (AWS). We will launch an EC2 instance (a virtual server) to host the website, making it accessible from anywhere via the Internet. This project demonstrates how easy it is to deploy a simple application on the cloud, while also introducing key AWS services like EC2, Security Groups, and Elastic IPs.

By the end of this project, you will have a functional calculator app running live in the cloud, giving you practical experience with cloud hosting, website deployment, and basic cloud security.

Fig. 2.15 Output window of simple calculator

I mentioned only few project in report for the documentation. I have hands-on experience with the all core AWS services.

2.3 Outcomes form the Internship Program

- Learn core cloud computing concepts and understand key AWS services.
- Deploy, configure, and manage services like EC2 (servers), S3 (storage), VPC (networking), and Lambda (serverless computing).
- Apply AWS security best practices using IAM roles, security groups, encryption, and access policies.
- Build, host, and manage real-world applications fully in the cloud environment.
- Monitor AWS resources and optimize usage to control and reduce cloud costs.
- Gain the skills needed to prepare for AWS certifications and start entry-level cloud computing roles.

Chapter 3

Conclusion

Completing my AWS Cloud Computing Internship at MicroDegree as an AWS Cloud Architect has been an incredibly valuable experience. Throughout this journey, I gained strong hands-on knowledge of designing, deploying, and managing scalable, secure, and cost-efficient cloud solutions using AWS services. I worked extensively with services like EC2, S3, RDS, Lambda, VPC, and CloudFront, and developed a solid understanding of real-world cloud architectures and best practices.

This internship not only helped me strengthen my technical skills but also enhanced my problem-solving abilities and understanding of cloud security, automation, and cost optimization strategies. It gave me confidence in building solutions that meet business needs while following AWS Well-Architected principles. I am excited to carry forward the skills, experience, and professional insights I gained and continue growing in the field of cloud computing.

Reafference

1. https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html
2. Amazon Web Services (AWS) Official Documentation
   URL: https://docs.aws.amazon.com/
   Description: Official technical documentation and user guides for all AWS
3. https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html'
4. https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-identity-pools.html
5. https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/kms.html [9] https://medium.com/petabytz/amazon-web-services-wordpress-configuration-on-ecs-using-ecr images-9c1e97d05aaa