Project Documentation

Services Used on the Application

- **Amazon S3 (Simple Storage Service)** Amazon S3 is a scalable cloud storage service used to store and retrieve data over the internet. Main features:
    - **object storage**: Files (objects) are stored in containers called "buckets".
    - **Durability and usability:** It backs up your data in multiple data centers (availability zones).
    - **Security:** Offers access controls, encryption options and auditing capabilities

- **Amazon EC2 (Elastic Compute Cloud)** Amazon EC2 provides scalable virtual servers (instances) in the cloud. Main features:
    - **Flexibility**: Quickly expand computing capacity when you need it. you can scale.
    - **Various instance types:** It comes with different performance, memory and storage options.
    - **Auto Scaling ve Load Balancing**: Used to manage traffic and ensure high availability.

- **Amazon RDS (Relational Database Service)** Amazon RDS is a service for running, scaling, and managing relational databases in the cloud. Main features:
    - **Supported Databases:** Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle ve Microsoft SQL Server.
    - **Ease of Management:** Backups, software patches, monitoring and scaling are automatically managed.
    - **High Availability:** Multi-AZ deployments and automatic backups

- **Amazon EFS (Elastic File System)** Amazon EFS provides a fully managed NFS (Network File System) file system in the cloud. Main features:
    - **Scalability:** Storage capacity scales automatically.
    - **Ease of Use:** Simultaneous access is provided via multiple EC2 instances.
    - **Security**: Data access is secured with VPC, IAM and KMS integrations.

- **Amazon VPC (Virtual Private Cloud)** Amazon VPC provides a logically isolated network in the AWS cloud. Main features:
    - **Network Control:** You can specify IP address ranges, subnets, and routing tables.
    - **Security:** Security groups and network ACLs You can control traffic with .
    - **Connection Options:** You can connect to your on-premises networks with VPN connections and AWS Direct Connect.

- **Amazon Load Balancer** Send incoming application traffic to multiple destinations (for example, EC2 instances) distributes it among. Types:
    - **Elastic Load Balancing (ELB):** Distributes incoming traffic among application, network, and gateway load balancers.
    - **Application Load Balancer (ALB):** Layer 7 load balancing for HTTP and HTTPS traffic.
    - **Network Load Balancer (NLB):** Layer 4 load balancing for TCP, UDP and TLS traffic.
    - **Elastic Load Balancing (ELB):** Distributes incoming traffic among application, network, and gateway load balancers.
    - **Application Load Balancer (ALB):** Layer 7 load balancing for HTTP and HTTPS traffic.
    - **Network Load Balancer (NLB):** Layer 4 load balancing for TCP, UDP and TLS traffic.

- **Amazon Security Group** in VPC instance'lar It acts as a virtual firewall for Main features:
    - **Inbound and Outbound Rules:** Which traffic? instance'a Controls whether you can enter or exit.
    - **Stateful Filtering:** It automatically allows return traffic.
    - **Easy Management:** Rules can be easily added and removed.

- **AWS Lambda** It is a serverless compute service that lets you run code in response to events. Main features:
    - **Serverless:** It does not require infrastructure management.
    - **Event Based:** It responds to events from different AWS services.
    - **Scalability:** It automatically scales according to demand.

- **Amazon Route 53** DNS (Domain Name System) is a web service. Main features:

    - **Domain Name Management:** You can register and manage domain names.
    - **High Availability and Scalability:** It forwards DNS requests quickly and reliably.
    - **Route Routing:** It can direct users in different geographical regions to the most suitable server.

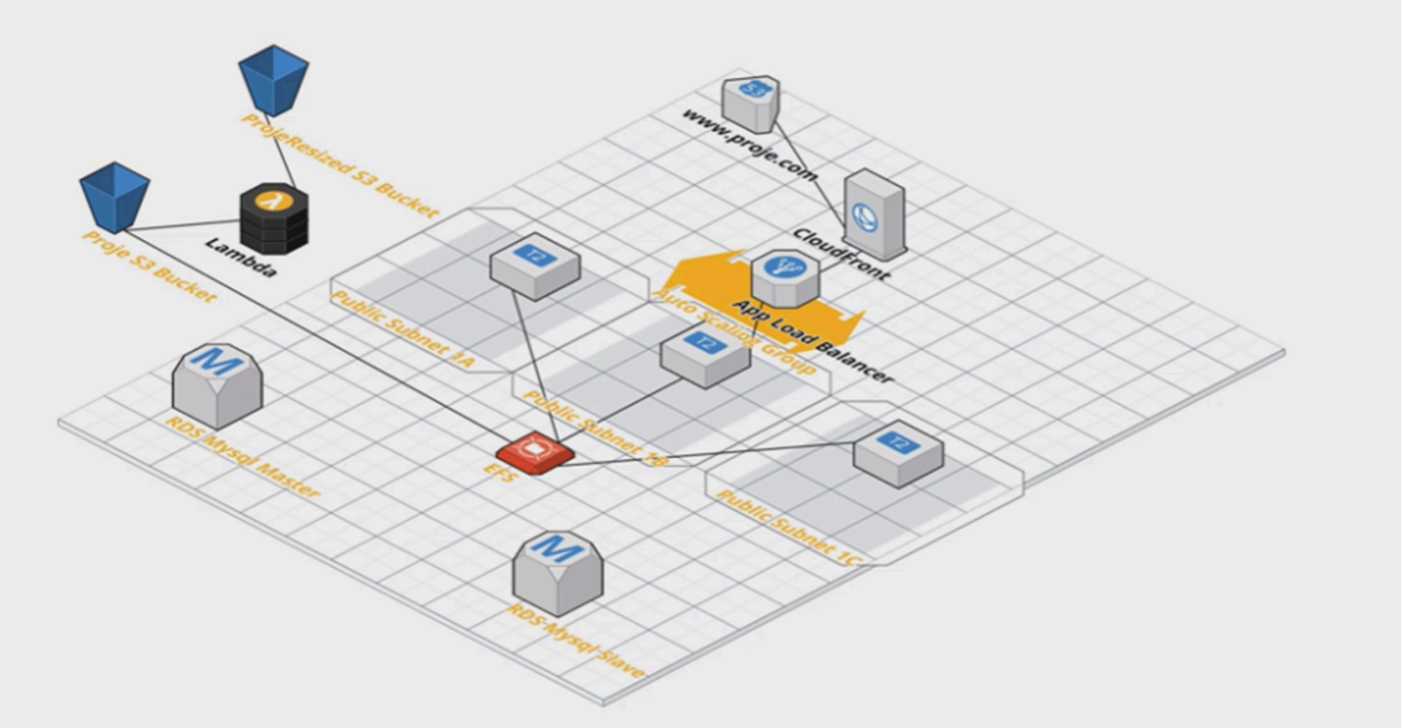
- **Amazon Auto Scaling** Amazon Auto Scaling is a service that automatically scales the resources used in the cloud environment. The main features and advantages of this service are:

    - **Dynamic Scaling:** Automatically allocate resources according to user demand adds or removes.
    - **Planned Scaling:** Resources automatically at certain times scales.
    - **Predictive Scaling:** It predicts future demands and preempts resources by analyzing historical data. scales.
    - **High Availability:** It prevents service interruptions and improves performance by increasing or decreasing resources.
    - **Cost Optimization:** Optimizes costs by adding and removing resources as needed.

- **Amazon CloudFront** AWS'nin It is a content delivery network (CDN) service. It delivers web content (such as HTML, CSS, JavaScript, images and videos) to users quickly and securely through data centers in various regions of the world. Main features:

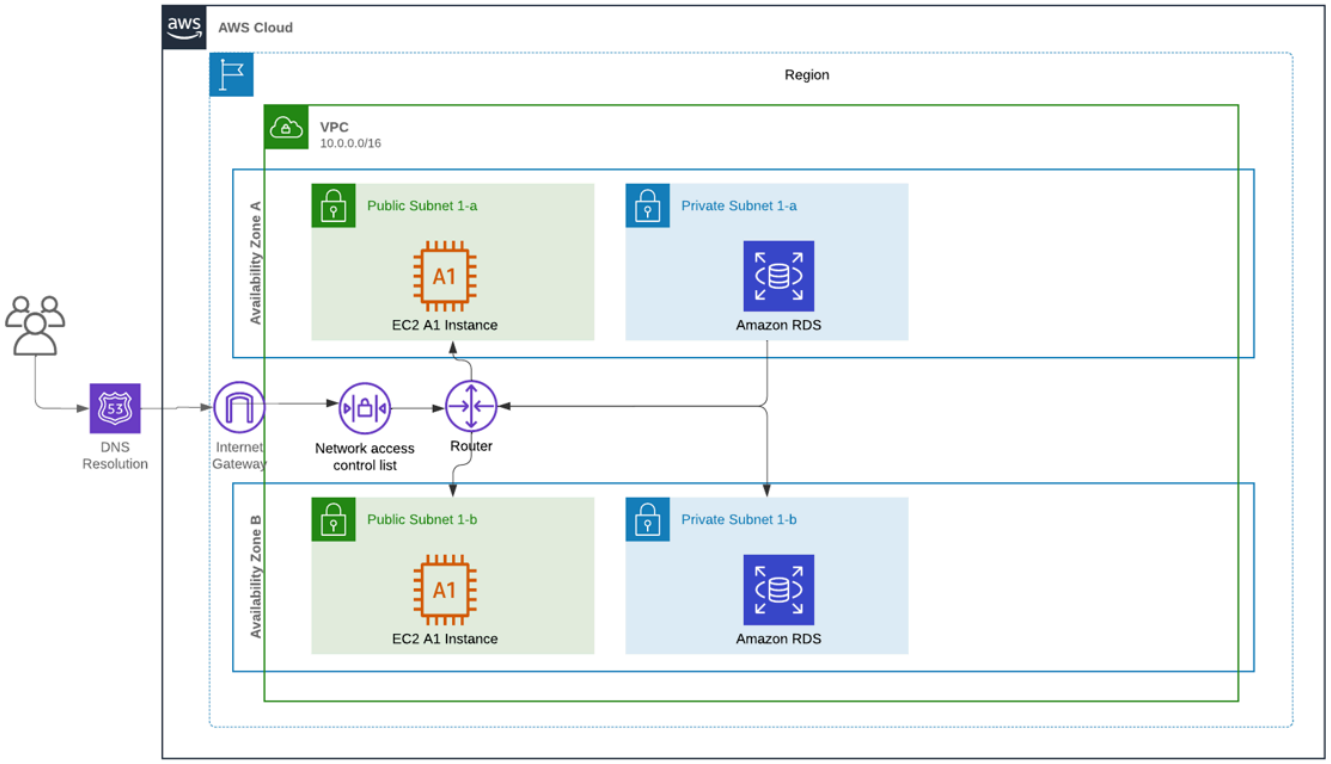
    - **Low Latency and High Speed:** It improves performance by delivering content from the server closest to the user.
    - **Global Reach:** Thanks to distribution points around the world, content becomes accessible from anywhere.
    - **Security:** AWS Shield provides high security with AWS WAF and SSL/TLS support.
    - **Integration:** It integrates seamlessly with S3, EC2, Lambda and other AWS services.

Project Architecture



- This application will run on EC2 virtual machines that we will create in the Auto Scaling group. Auto Scaling group, each public in our subnet will create at least three virtual machines, with at least one If these virtual machines consume more than 90% CPU usage, increase their number to up to five virtual machines. will increase. All these virtual machines will be created by a Load Balancer behind by taking load distribution will be provided. Later, it will be quickly accessible from all over the world with a CloudFront distribution we will create, and this web-based application will be accessed from a URL such as www.proje.com using the Route 53 DNS service.

- Our application will write users' images to the EFS drive shared by EC2 machines; Name, e-mail and phone information will be recorded in the RDS database. Users can then save their uploaded images to S3 bucket'ta will store it and send it to another S3 using the Lambda function in the same way. bucket'ta They will get backup.



AWS'of Its infrastructure provides a hierarchical structure for organizing and managing various services and resources. This structure **Region**, **Availability Zone** ve **VPC (Virtual Private Cloud)** It consists of components such as. Here is the explanation of this hierarchy:

1. AWS Cloud

AWS Cloud, Amazon Web Services'in is the cloud computing platform it offers. It allows users to run their applications and infrastructures in the cloud by taking advantage of various services.

2. Region

Region: AWS offers many clusters of data centers, called geographies, around the world. Each region contains one or more Availability Zones.
Independent Regions: Each region is independent from other regions and is designed to provide data transfer, low latency, and high availability. Regions are often located in large geographic areas (e.g., North America, Europe, Asia-Pacific).
Example: us-east-1 (Northern Virginia), eu-west-1 (Ireland).

3. Availability Zone

Availability Zone (AZ): Each region contains one or more AZs with isolated and independent electrical, network, and cooling infrastructure.
Physical Privilege: AZ'ler located in physically separate locations but low latency with their duration are connected to each other. This ensures high availability and fault tolerance.
Örnek: us-east-1a, us-east-1b, us-east-1c.

4. Virtual Private Cloud (VPC)

VPC: It is a virtual private network environment.
Network Configuration: You can specify IP address ranges, subnets, routing tables and gateways within the VPC.
Security: VPCs, security groups, and network access control lists (ACL's) is secured by.

5. Subnets

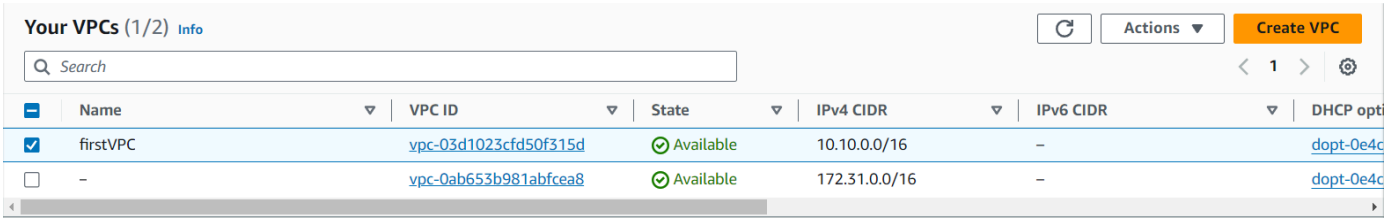
You can create two types of subnets within VPC: **public subnet** ve **private subnet**.

Public Subnet: These are subnets that are accessible over the Internet. **of EC2** web servers, front-end applications and other components are public subnet is placed.

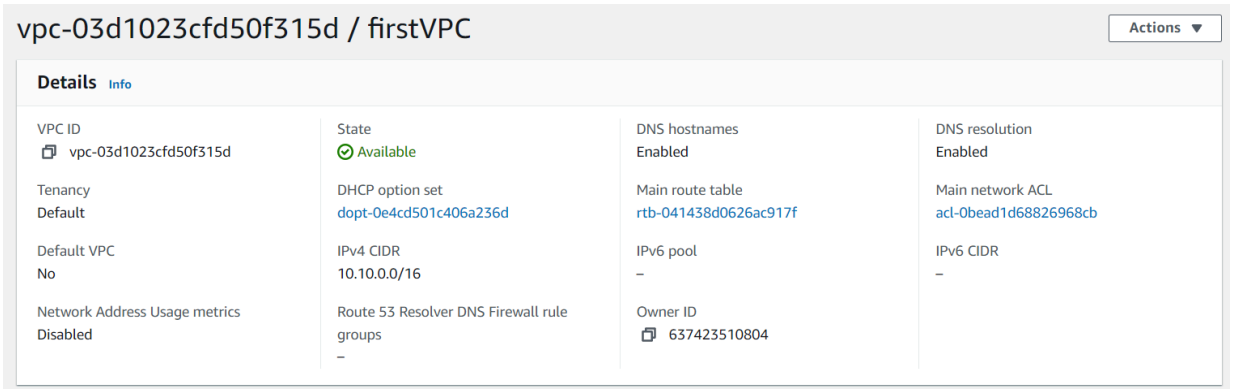
Private Subnet: These are subnets that do not have internet access and only communicate with other resources within the VPC. **Amazon RDS** private for privacy and security subnet takes place. **of EC2** a backend transaction server or data analytics server is located here.

Project steps

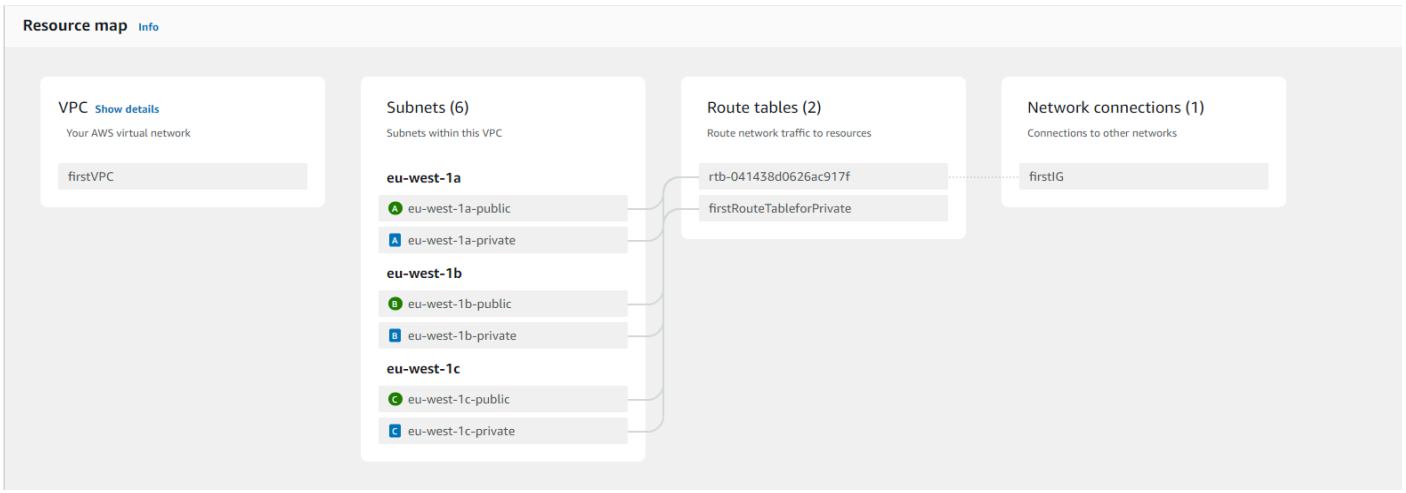
- **VPC (Virtual Private Cloud) oluşturma**
    - First, we created a VPC on AWS and created and set the necessary configurations such as subnet, route table, and internet gateway.



- We can easily create a vpc when we click on the button that says Create VPC. However, the point we need to pay attention to here is the region in which our AWS service is located. We chose the Ireland region in our project. Let's look at the VPC called firstVPC that we created for our project.



- We can see the details of the VPC we created here. We can see settings such as IPv4 CIDR and IPv6 CIDR settings, DNS configurations, Firewall, and configure them according to our needs.
- Let's look at the Subnet and route table settings we use in the project.

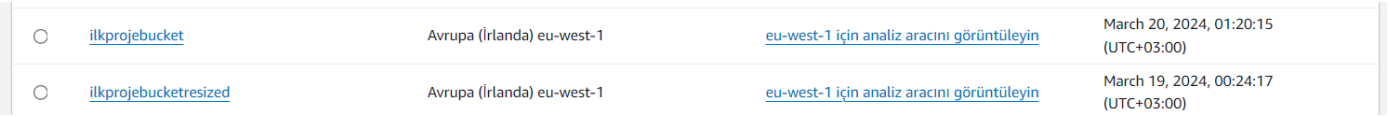


- I created separate route tables for public and private subnets. While the route table I connected to the public subnets will enable my public

subnets to be opened to the internet and control the communication traffic with the internet, the route table I created for the private subnets will only control the communication traffic between the subnets on the vpc we created.

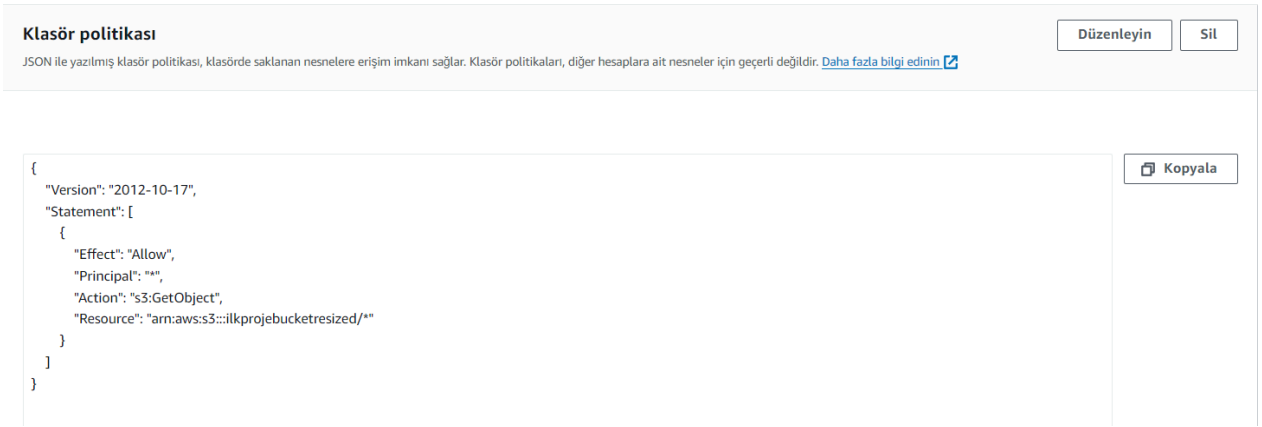
- I connected an internet gateway to the route table to which I connected my public subnets and ensured that requests within the desired IP range could log into the VPC, thus increasing application security.

- **Creating an S3 Bucket**
  - In the application, the images uploaded by the users were stored in the S3 bucket, and at the same time, a second S3 bucket was created to create a backup system and to create and store resized images.
  - AWS Lambda function was used to upload the uploaded images in the project to S3 bucket and store the resized version, this process is mentioned in the following stages.



○	ilkprojebucket	Avrupa (İrlanda) eu-west-1	eu-west-1 için analiz aracını görüntüleyin	March 20, 2024, 01:20:15 (UTC+03:00)
○	ilkprojebucketresized	Avrupa (İrlanda) eu-west-1	eu-west-1 için analiz aracını görüntüleyin	March 19, 2024, 00:24:17 (UTC+03:00)

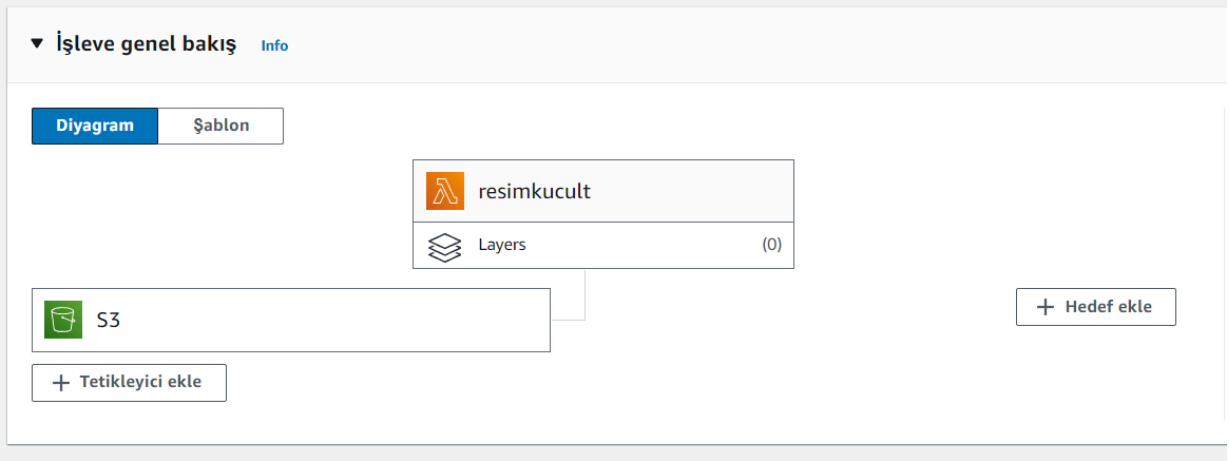
  - There are some points we need to pay attention to when creating an S3 bucket. Setting the "Manage public access control lists (ACLs) for this bucket" and "Manage public bucket policies for this bucket" options in the S3 bucket "Configure options" section and the bucket policy configuration required for public access.



  - Thanks to the Bucket policy we wrote here, the bucket accepts all http requests.

- **Using AWS Lambda Functions**
  - We created a backup system by copying images from one S3 bucket to another S3 bucket with the Lambda function.

- We added a trigger to the Lambada function, and every time an image is added to the S3 bucket, this trigger will run and copy the image to the other S3 bucket.



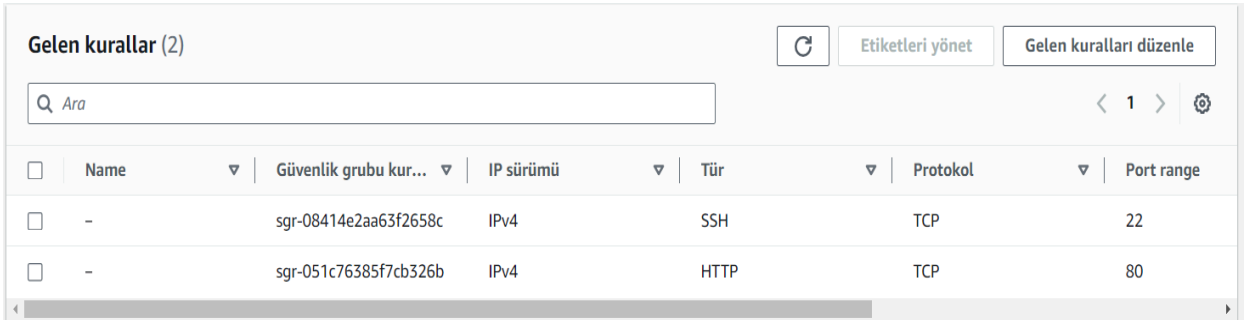
- lambda function

- **Creating a Security Group**

- EC2 of instances to ensure security and for HTTPS traffic, SSH connection ProjectSecGroup named Security Group'u we created.

- Details and Incoming Rules section:

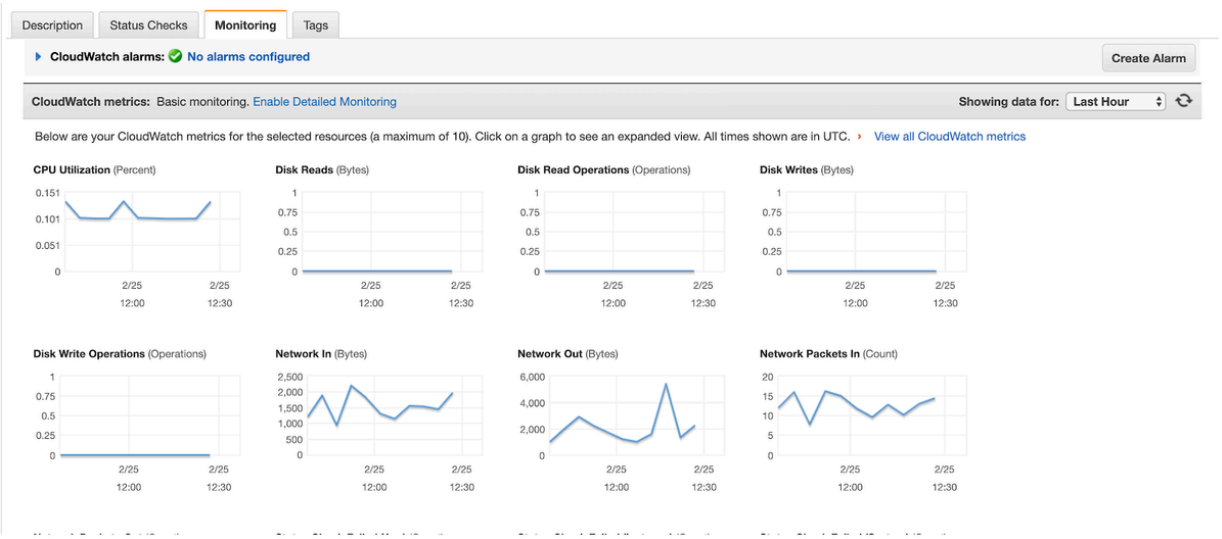
- IP version, type, and protocol information in the incoming rules:



- HTTP ve SSH our connections The IP version is IPv4. The protocol used is TCP protocol. port 22 SSH'aIf port 80 is HTTP'ye has been appointed.

- **AWS EC2 (Elastic Compute Cloud)**
  - We created our EC2 machine to work as a Server in the application. This machine is a virtual machine running on AWS. While creating this machine, we can manually adjust all the settings. We can make all hardware-related configs such as ram, processor, storage space, ssd hdd, etc. At the same time, we can use pre-created images in the AWS market as the operating system, we can create an image ourselves and we can also use the ready-made AWS images offered by AWS.
  - What are these images? In fact, we can say that it is all the software required for our application. This could be a database. We carried out these operations in our project.
  - We created the EC2 using T2 micro and Amazon Linux 2 Ami.

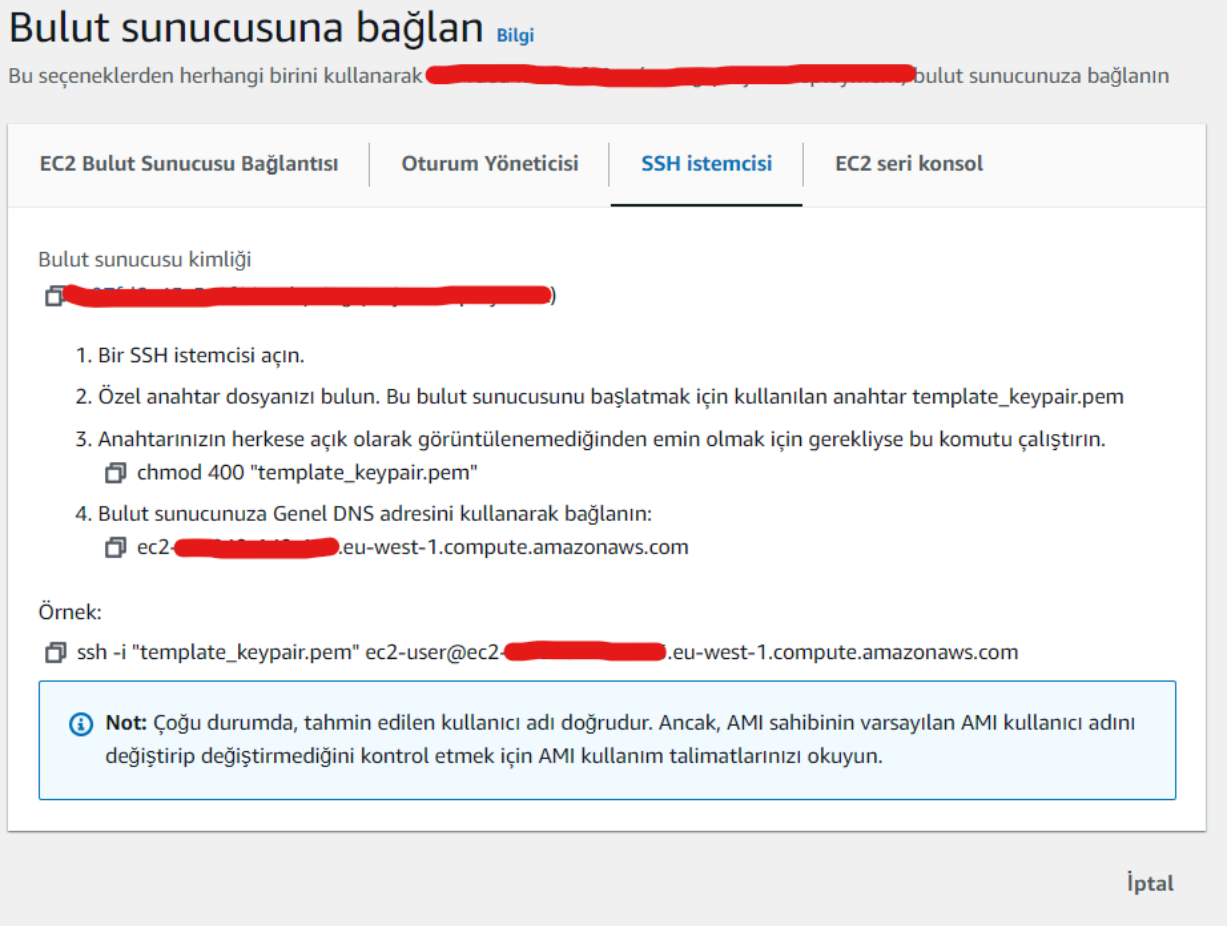
- We have the opportunity to monitor the EC2 status in real time on the AWS management console, and we can also create an alarm mechanism and notify us via SMS or e-mail.



- When creating EC2, we need to choose a security group. We can give this setting as default or create a custom security group ourselves. We used a custom security group in our project. Here, we need to give the correct permission and port of the ssh connection in the security group, because EC2 can connect to the machine remotely via ssh. so we can connect.

- **EC2 Connection with SSH**

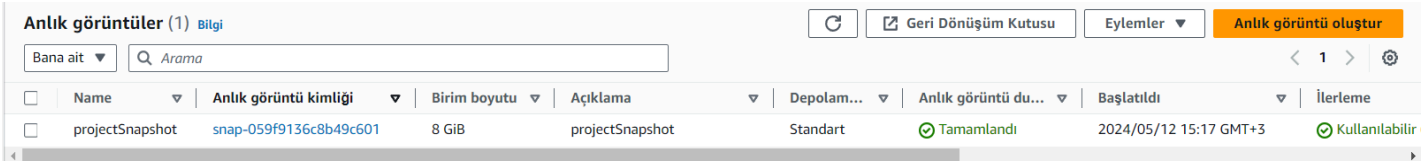
  - To connect to EC2 via ssh, AWS needs to provide a key pair when creating EC2. This file is a private key encrypted with RSA-256.



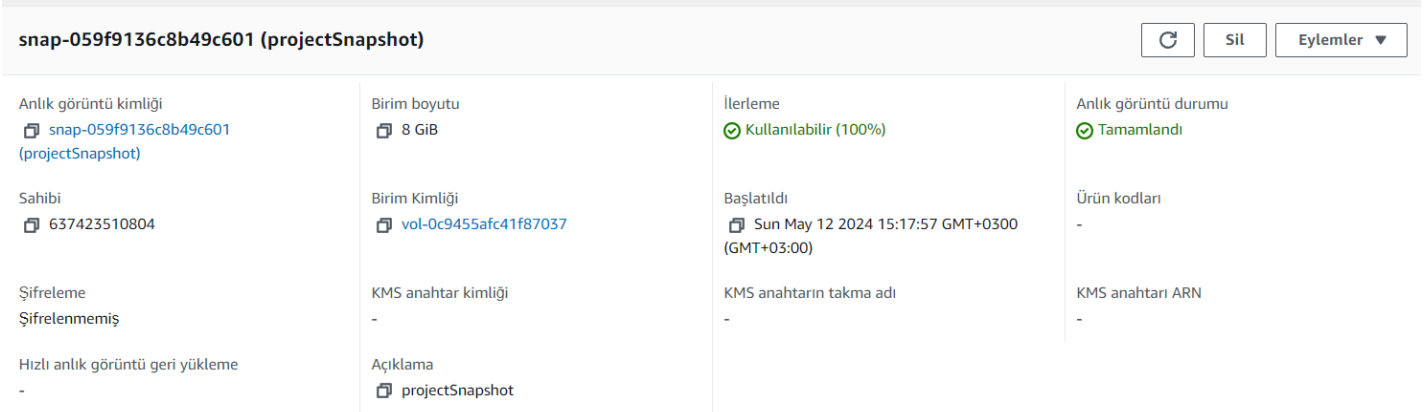
- ssh -i "template_keypair. ec2-user@ec2.xxxxxxxxx5.eu-west-1.compute.amazonaws.com
- We can connect from the terminal with the code. Another point to note is that we need to give full access to the key pair file.
- After making the connection, you can do whatever you want within the machine. Now you have a virtual computer, you can install the necessary packages and files and even deploy your applications.

```
C:\Users\omer faruk kan>ssh -i "C:\awsKeypair\template_keypair.pem" ec2-user@ec2-34-249-149-175.eu-west-1.compute.amazonaws.com
The authenticity of host 'ec2-34-249-149-175.eu-west-1.compute.amazonaws.com (34.249.149.175)' can't be established.
ECDSA key fingerprint is SHA256:Dq6I6gbNGQYVcmpmoFttEU73LtISOEANLPHXFtCNum8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-249-149-175.eu-west-1.compute.amazonaws.com,34.249.149.175' (ECDSA) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Sun May 19 11:25:57 2024 from 176.30.243.214
[ec2-user@ip-172-31-30-114 ~]$ sudo su
[root@ip-172-31-30-114 ec2-user]#
```

- **Template instance'den SnapShot alma**
    - Snapshot template instantly creates new EC2 instance images with the same EC2 settings to use as an EC2 machine.
    - After taking the snapshot, we need to create the AMI, this is explained in the next step.

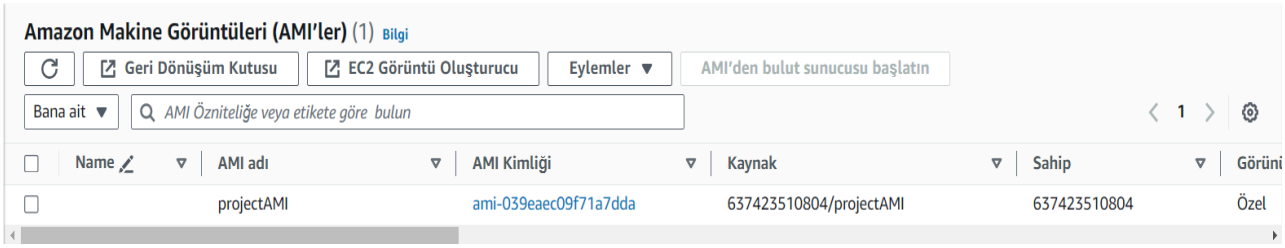


- We can take snapshots from all EC2s we create.

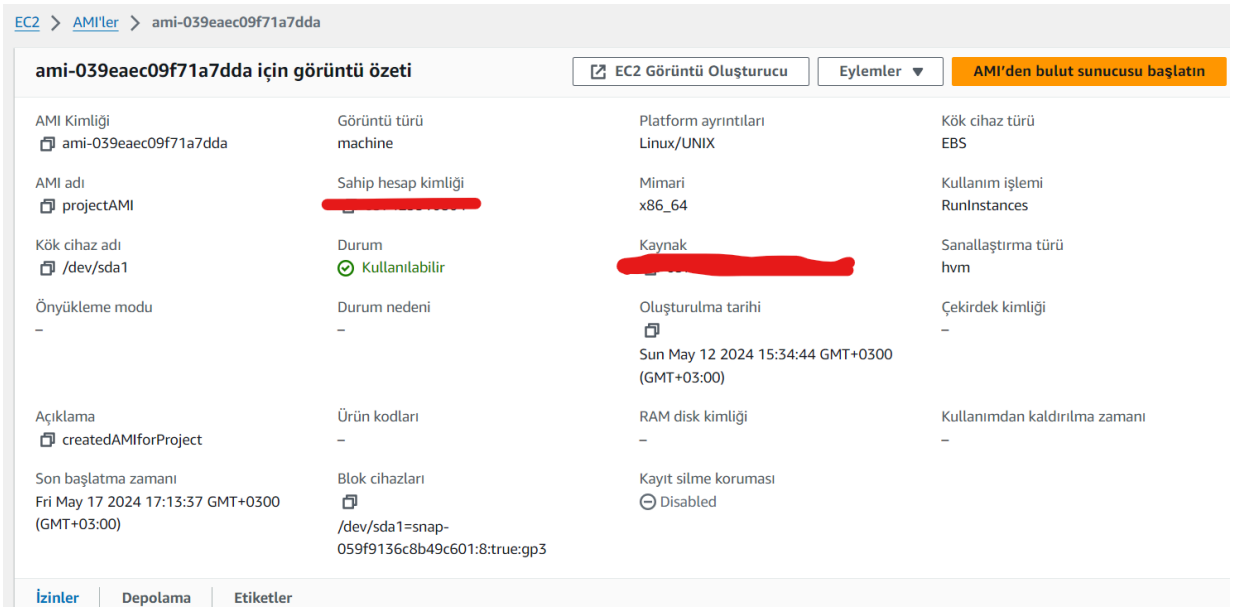


- We can see the snapshot details this way and create our new instances based on this snapshot. In practice we will use it under an auto scaling group.

- **Create an instance AMI**
    - First we got SnapShot'tan We created an instance AMI. We will create new machines using the auto scaling group with the instance AMI.

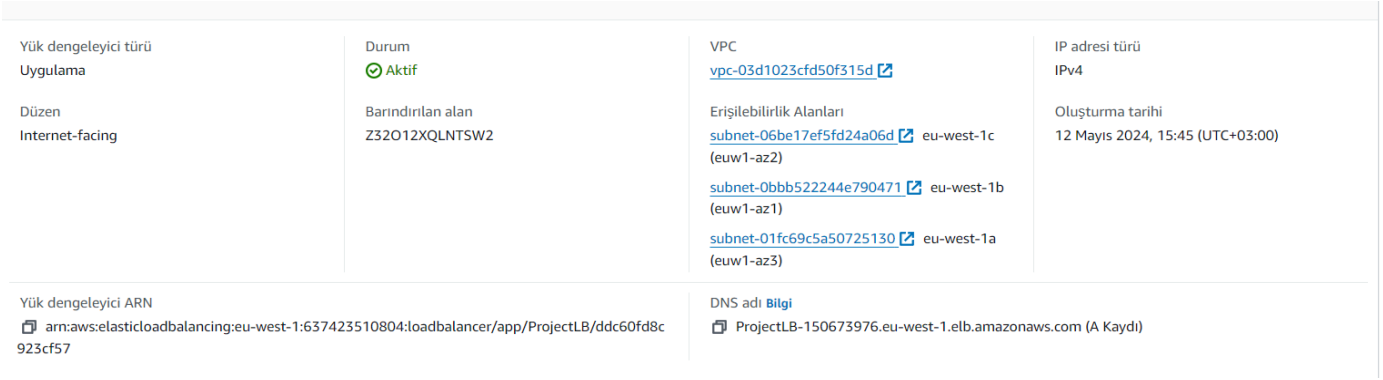


    - AMI ID details



- **Load Balancer**
    - A load balancer is a device or service that improves the performance, reliability, and availability of applications and services by distributing incoming network traffic across multiple servers. AWS'deVarious types of load balancers are offered under the name Amazon Elastic Load Balancing (ELB).
    - In the project, the Load balancer ensures that the requests coming to the application are distributed on the servers and thus continuous access is provided. Continuous access is of great importance in the cloud world

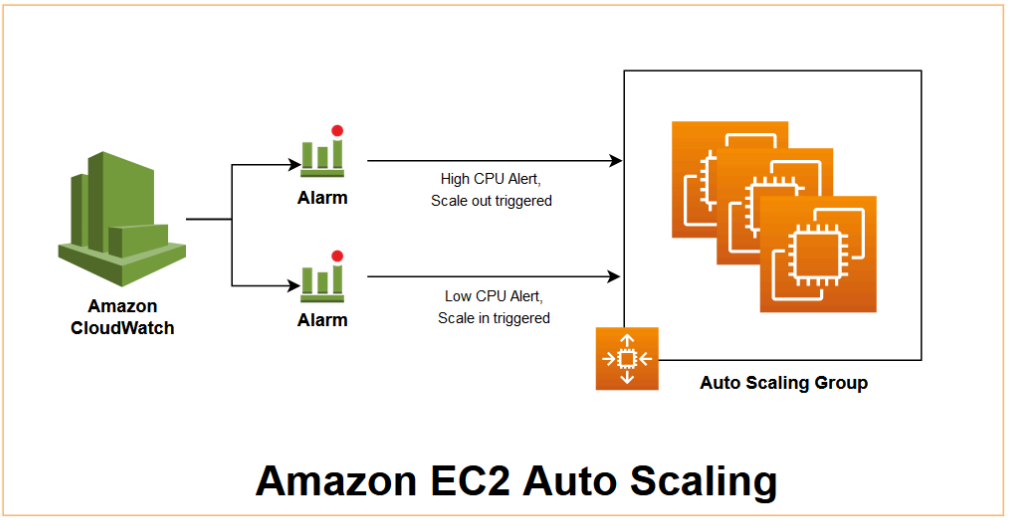
- Since AWS provides a service, continuous access is of great importance. Businesses' servers and applications must be both scalable and manageable.
- In our project, we gave importance to this situation and created a system with high availability and used Load Balancer in load distribution.



- If you pay attention, we only placed it on public subnets. This is because we want to ensure load distribution against requests from the internet.

- **Auto Scaling Group**

  - AWS Auto Scaling Group (ASG) is a service that allows you to automatically scale on Amazon Web Services (AWS). ABO is EC2 based on the criteria you specify. instances automatically starts or terminates, allowing your application to dynamically adapt to changes in demand. This ensures that your application always runs at optimum performance and costs are kept under control.

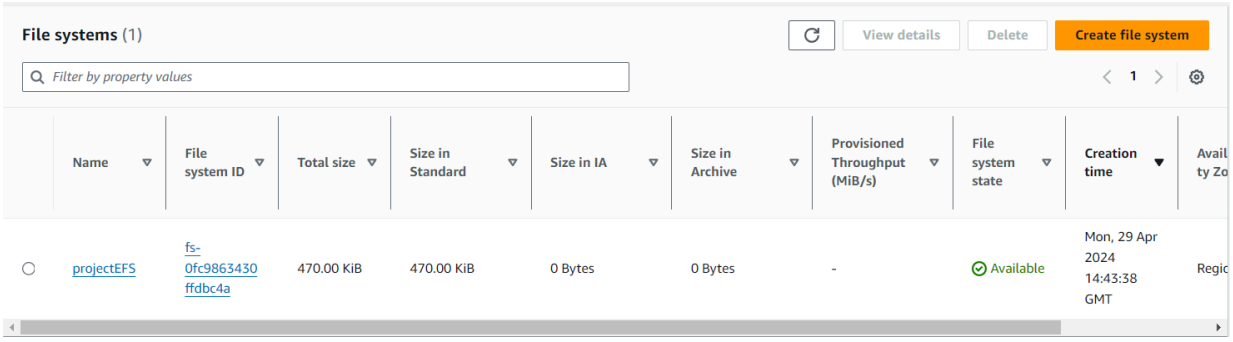


- In order to ensure continuous access to our application with the auto scaling group in the project, if a problem is encountered on any EC2 machine, a new one is created immediately. We configured this process as creating a new one when the CPU usage of the EC2 machine is above 90%.

.

- **Amazon EFS (Elastic File System)**

  - With EFS Elastic File System, we can both backup and distribute our application files within the system. we provided. TCreating a fully managed and scalable file storage system we aimed.



  - To do this, we need to mount the EFS system we created in our application server to our EC2 machine.
  - The mounting process can be done simply with a series of Linux codes. First, you need to download the AWS EFS package according to the operating system you use.

```
sudo yum install -y amazon-efs-utils
```

aws for linux

```
sudo apt-get update
sudo apt-get install -y nfs-common
```

ubuntu for

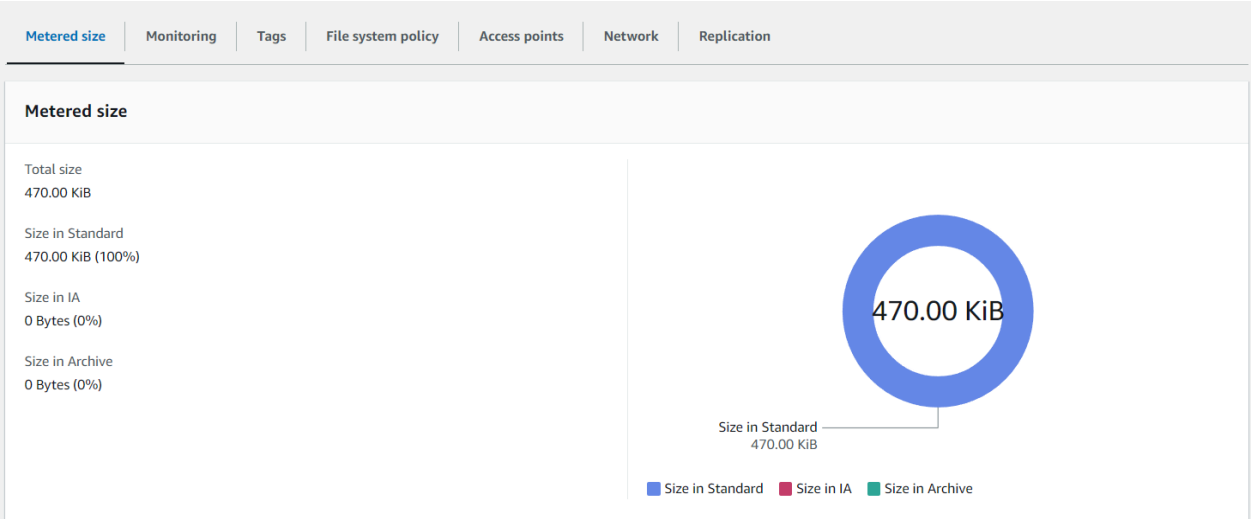
  - We create the file directory to be mounted

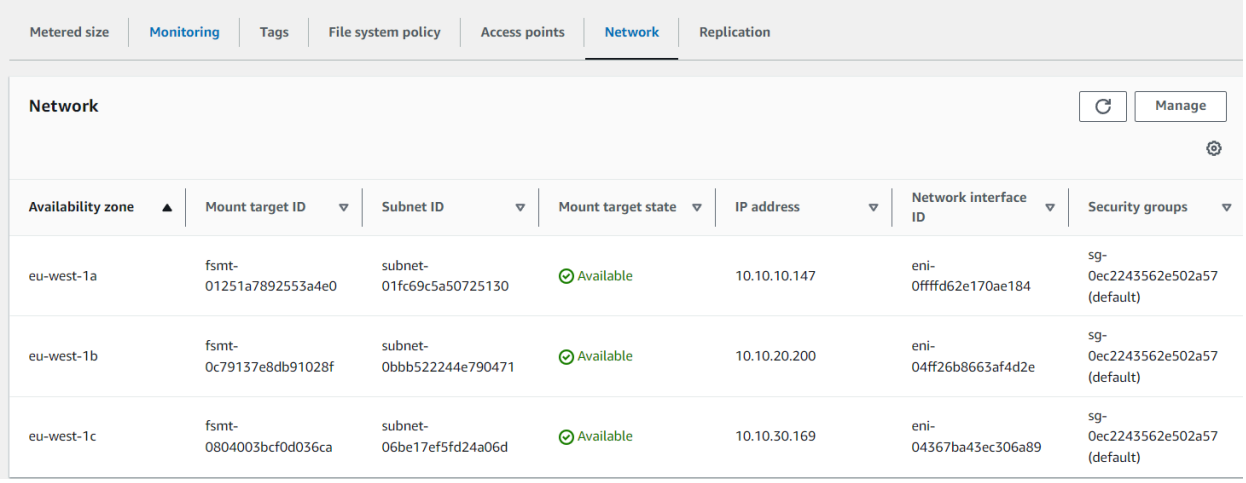
```
sudo mkdir /mnt/efs
```

  - We write the mount code

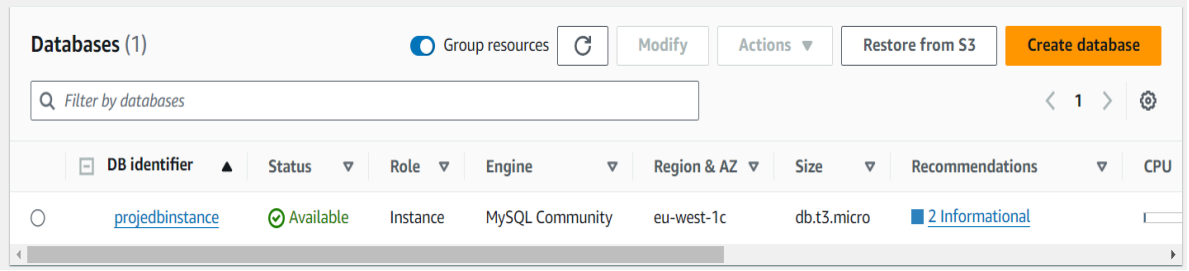
```
sudo mount -t efs <file-system-id>:/ /mnt/efs
```

  - The mounting process can be easily done this way. We can examine and monitor all logs within the EFS system.

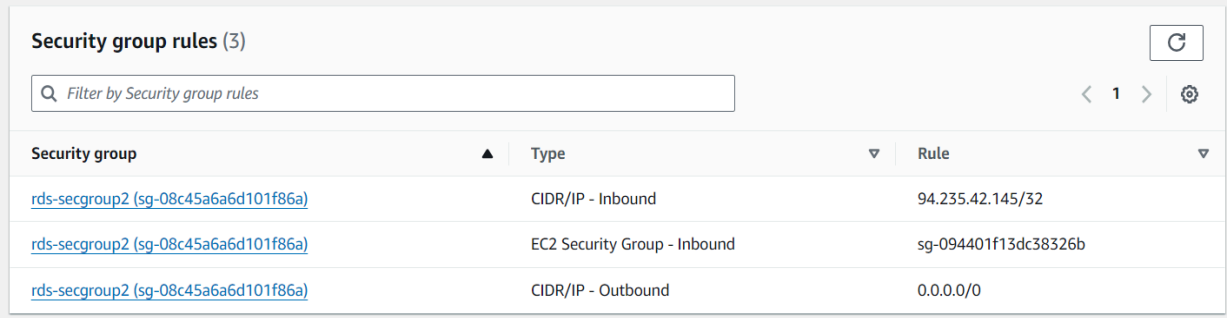


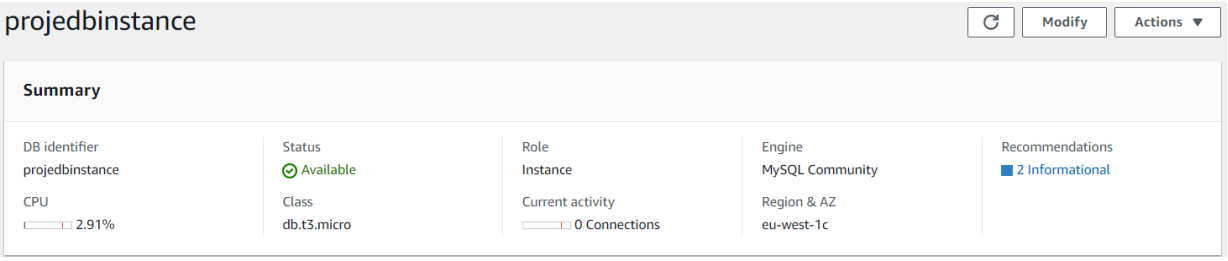


- **Amazon RDS (Relational Database Service)**
    - Amazon Web Service's It is a relational database service. In our project, users' data is kept here. For example, name, surname, e-mail, telephone, etc.
    - Amazon RDS supports various database engines such as MySQL, PostgreSQL, MariaDB, Oracle and Microsoft SQL Server.
    - We used MySQL database in our project.

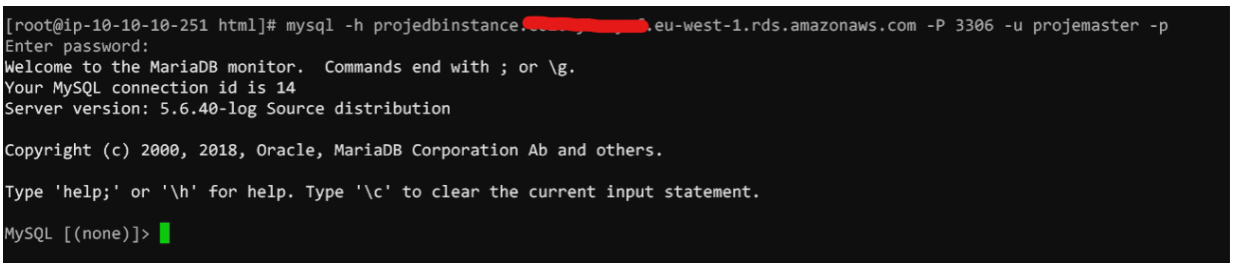


- **Amazon RDS** private for privacy and security subnet takes place. There is also a eu-west-1c private subnet in our project.
- What we want in our relational database IP those in the range We defined a security group so that others can access it and others cannot access it.





- Let's show MySQL Connection on EC2 machine.



- After making the MySQL connection, let's see our tables in the MySQL shell.