**Coming to Cloud using AWS**

**In AWS Services like**

**EC2 elastic compute cloud**

**CloudFront**

**CloudWatch**

**Route 53**

**S3 simple storage service**

**S3 Glacier**

**IAM**

**Data Transfer**

**ElastiCache**

**Key management service**

**Saving plans for AWS computer usage**

**Simple notification services**

**Simple storage**

**Support developer**

**Virtual private cloud**

**CloudFront**-

* Is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment.
* Offers the most advanced security capabilities, including field level encryption and HTTPS support

**CloudWatch-**

* CloudWatch is a monitoring and observability service
* CloudWatch provides you with data and actionable insights to monitor your applications, respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health.
* CloudWatch collects monitoring and operational data in the form of logs, metrics, and events, providing you with a unified view of AWS resources, applications, and services that run on AWS and on-premises servers.
* You can use CloudWatch to detect anomalous behavior in your environments, set alarms, visualize logs and metrics side by side, take automated actions, troubleshoot issues, and discover insights to keep your applications running smoothly.

**What is DNS ?**

DNS, or the Domain Name System, translates human readable domain names to machine readable IP addresses

For example - [www.tcs.com](http://www.tcs.com) to 23.58.220

**Types of DNS service**

**ADNS and RDNS**

**ADNS (**Authoritative DNS)

* An authoritative DNS service provides an update mechanism that developers use to manage their public DNS names

**Route 53**

* **Route 53** to perform **three main functions** in any combination:
* domain registration, DNS **routing**, and health checking.
* Your website needs a name, such as www.example.com. **Route 53** lets you register a name for your website or web application, known as a domain name.
* You can create records that have a type of addresses.

IPv4 =A

IPv6= AAAA, addresses.

**S3**

* Simple Storage Service
* a simple web service interface that allows huge amount of data storage and retrieval from anywhere from the internet
* Its provides developers highly scalable, reliable, fast and low cost data storage infrastructure.

**Maximum size of S3**

* you can store unlimited volume of data .
* size of range 0 bytes to 5 terabytes
* You can put an object of around 5 GB but you must have to enable Multipart Upload capability.

**S3 Glacier**

* is a secure, durable, and extremely low-cost
* **Amazon S3** storage class for data archiving and long-term backup
* With **S3 Glacier**, customers can store their data cost effectively for months, years, or even decades
* the underlying data model, the operations it supports, and the AWS SDKs that you can use to interact with the service.

**IAM** I know how to give IAM roles access but it is used by manager

* Identity and Access Management
* is a web service that helps you securely control access to AWS resources. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

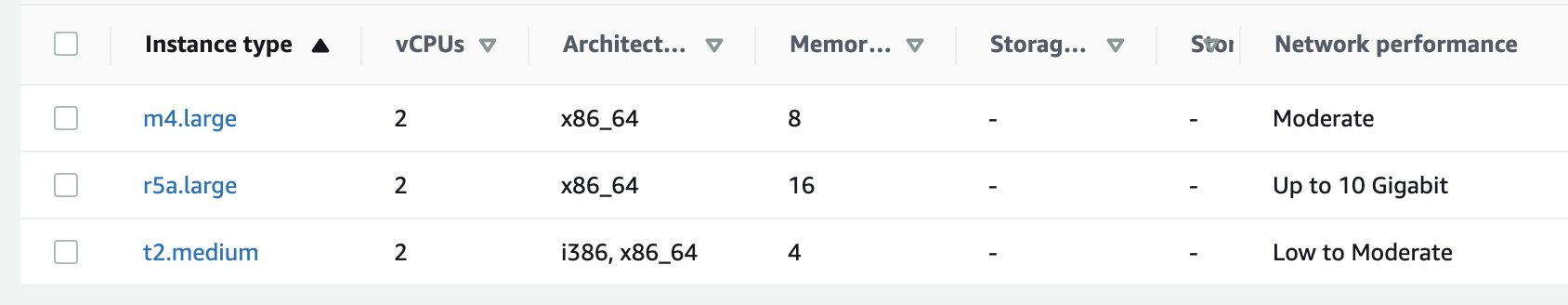
**Data transfer EFS**

* Charges are incurred when **data** is **transferred** out from **AWS** services to the Internet, or between **AWS** regions or Availability Zones.

**ElastiCache**

* The service improves the performance of web applications by retrieving information from managed in-memory caches, instead of relying entirely on slower disk-based databases.
* INSTANCES

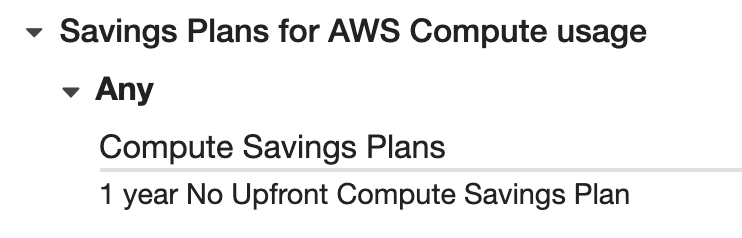
**Instances** -----like r5a.2xlarge,t2.medium,m4.xlarge



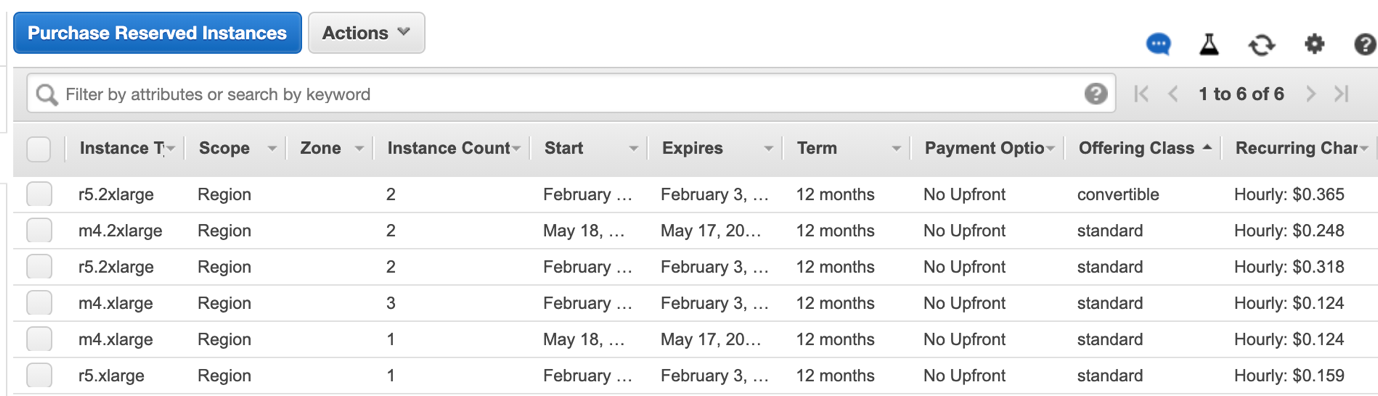
Launch Templates – not using

Sport request ------ not using

**Savings plans**



**Reserved Instances** ----- it depends

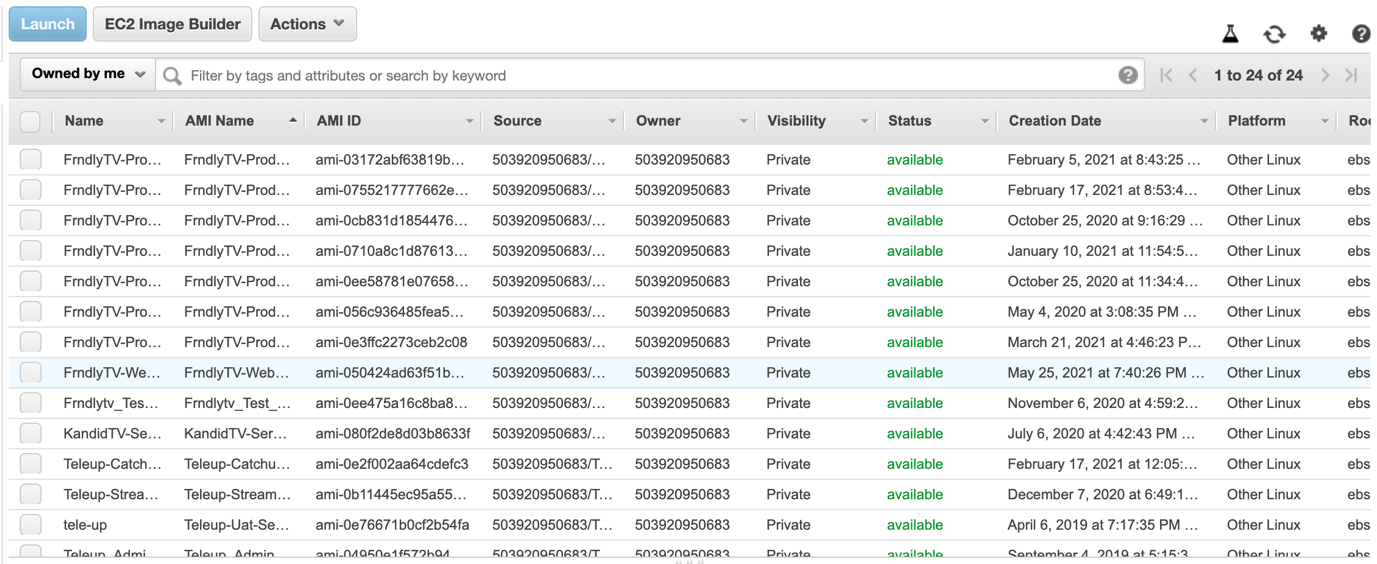


Dedicated Hosts ---- not using

Scheduled instances --- not using

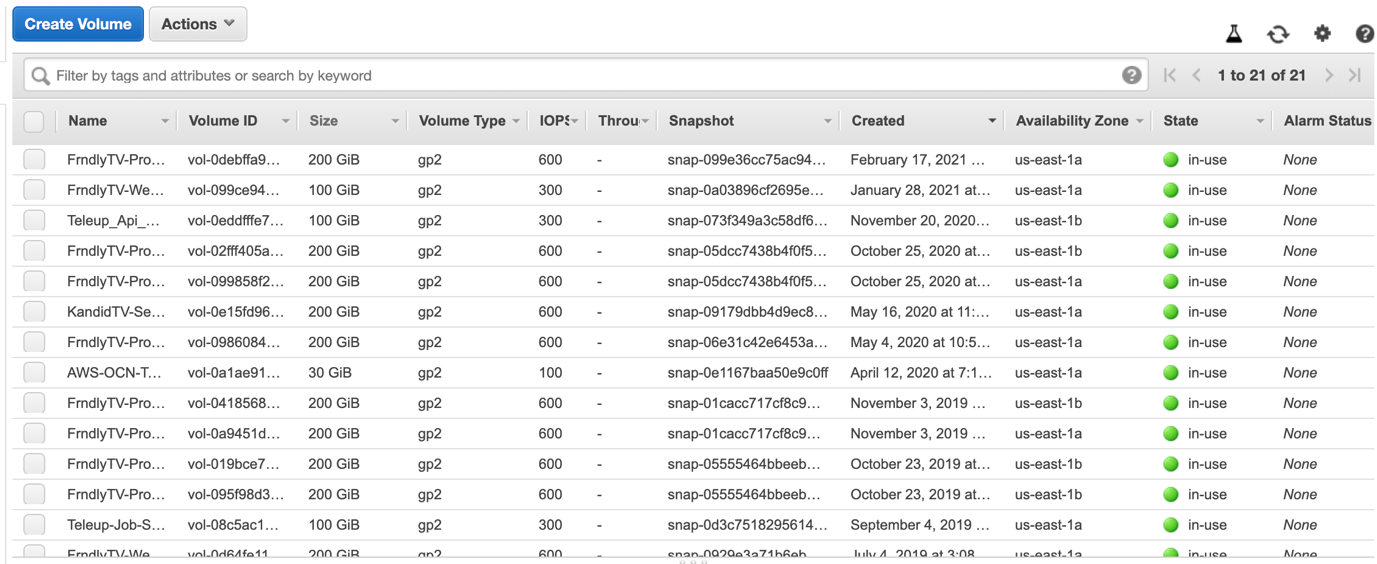
* IMAGES **–**

AMIs

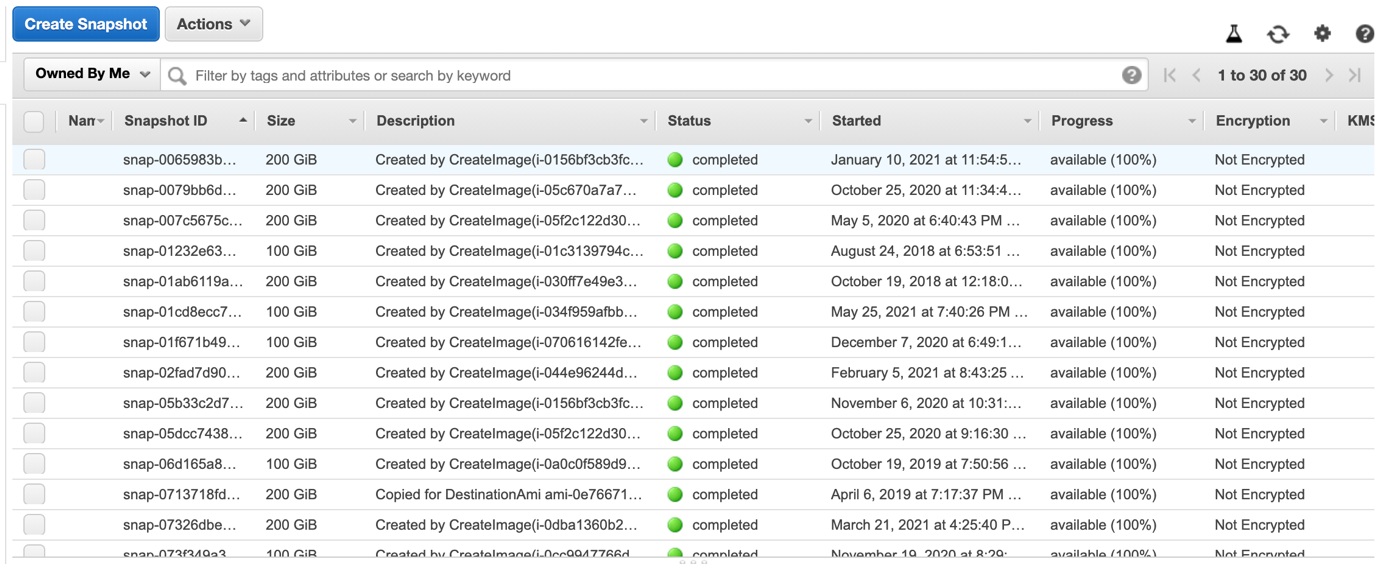


* ELASTIC BLOCK STORE in elastic block –

**Volumes typ**e gp2 200gb, 100gb



**Snapshots**

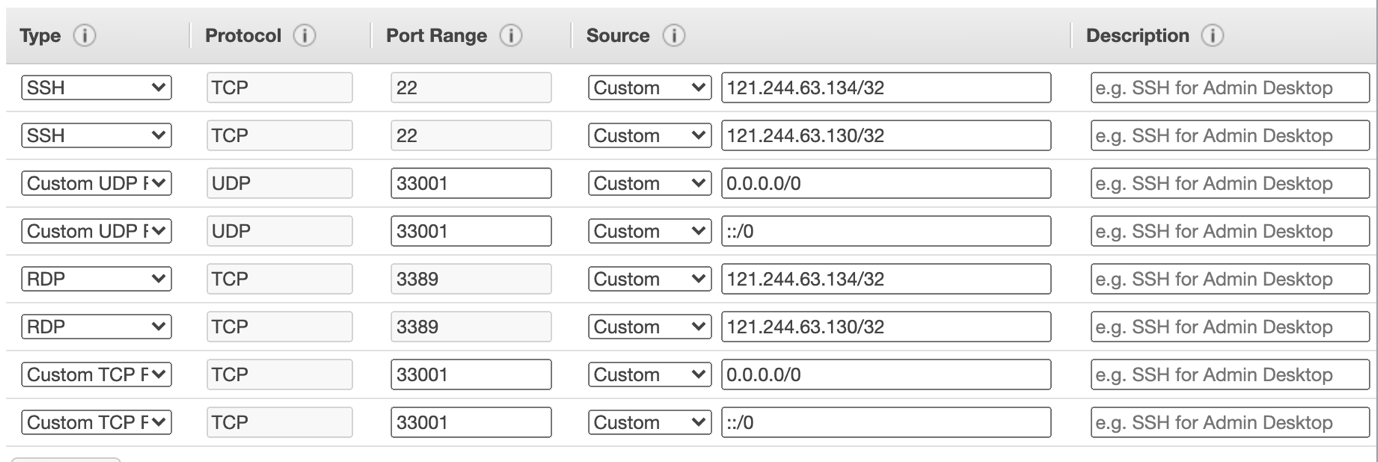


**Lifecycle Manager** ----- Not using

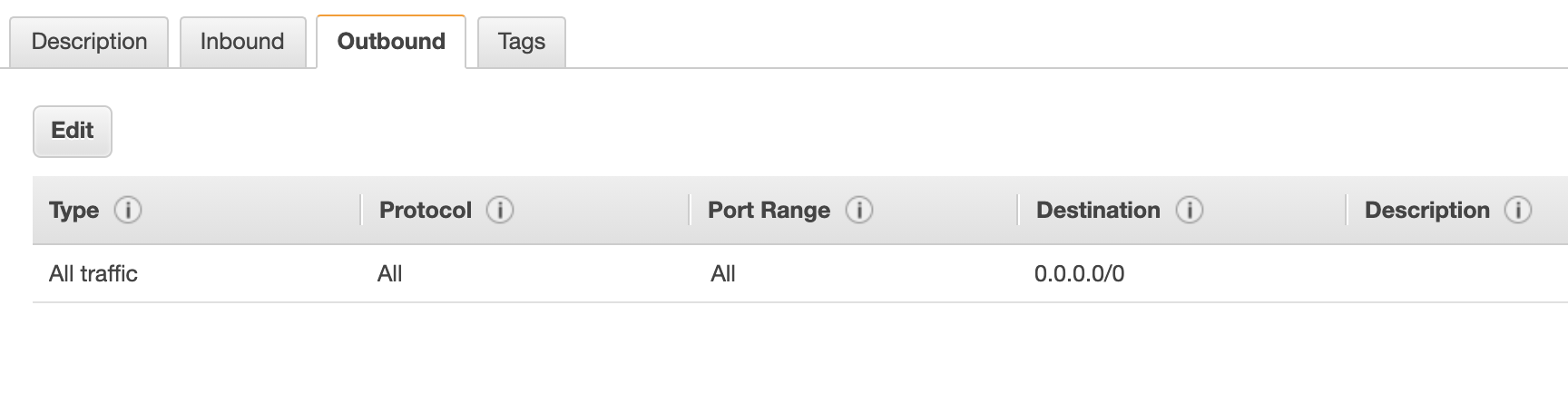
* NETWORK & SECURITY
* **Security Groups**

----in security groups

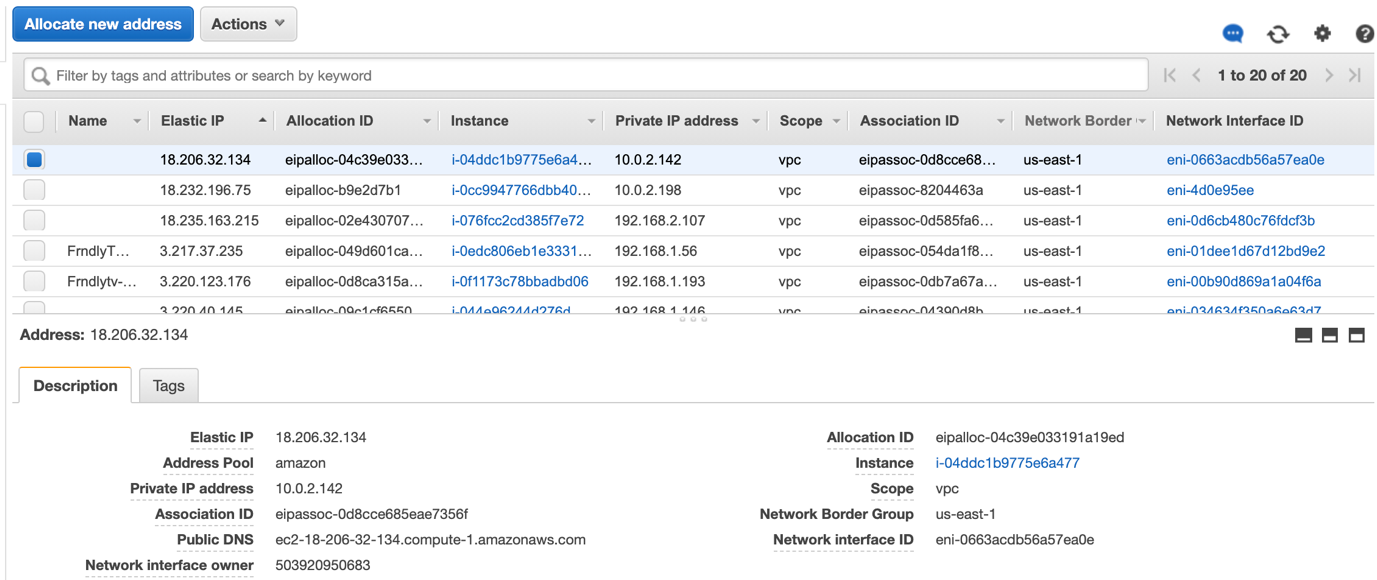
**Inbound Outbound**

**Inbound adding rule **

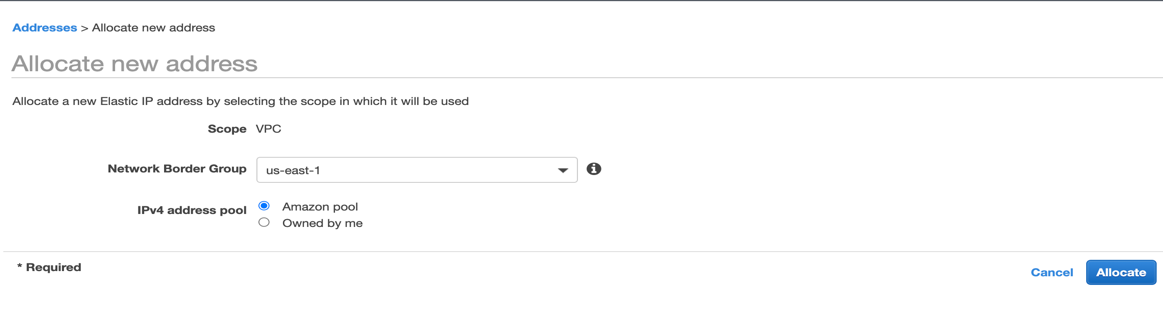
**Outbound**

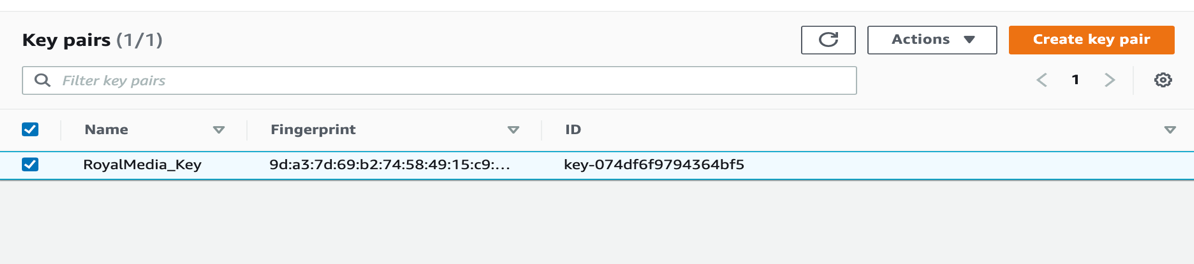
****

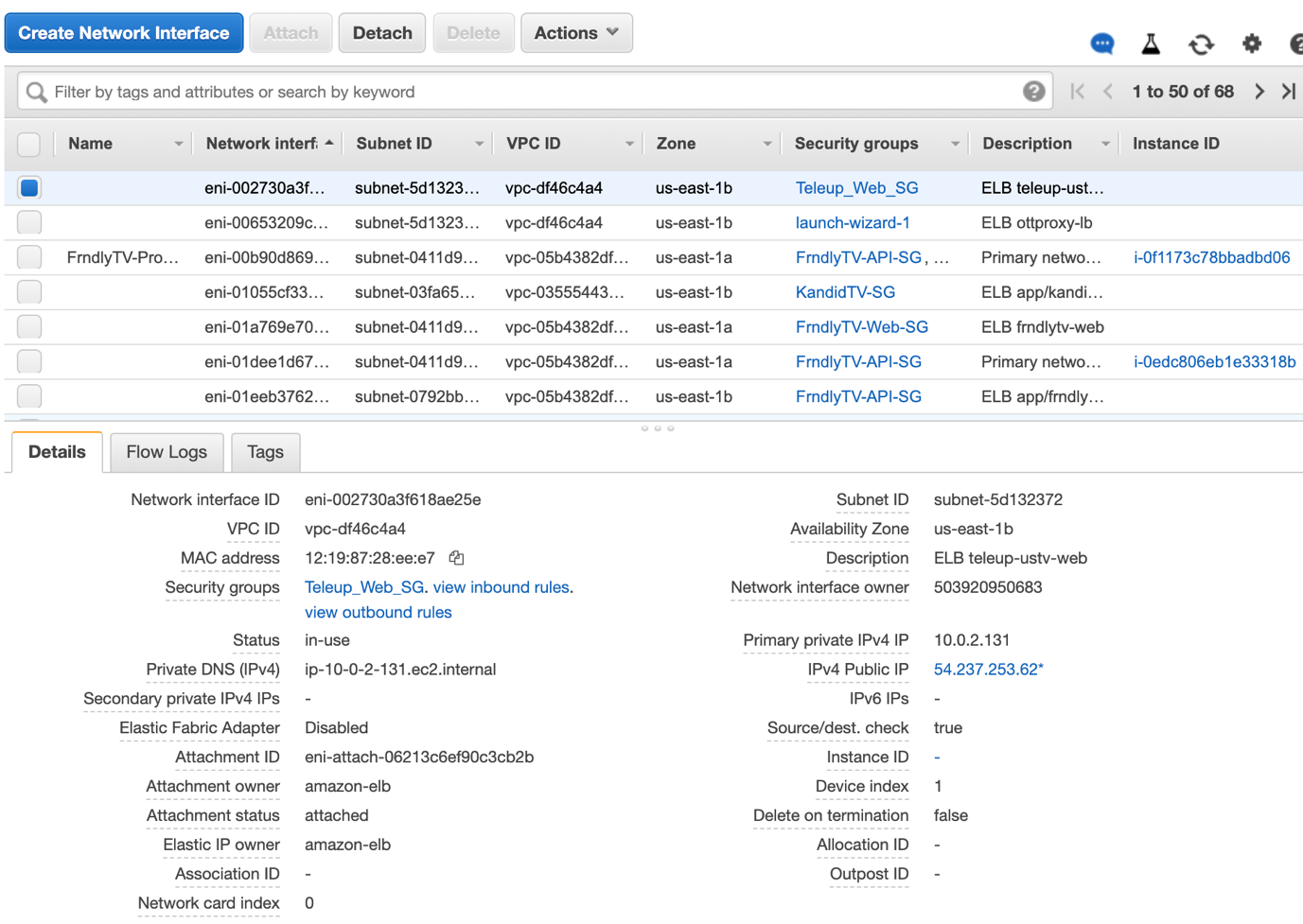
* **Elastic IPs**

****

**For new ips click on allocate new**

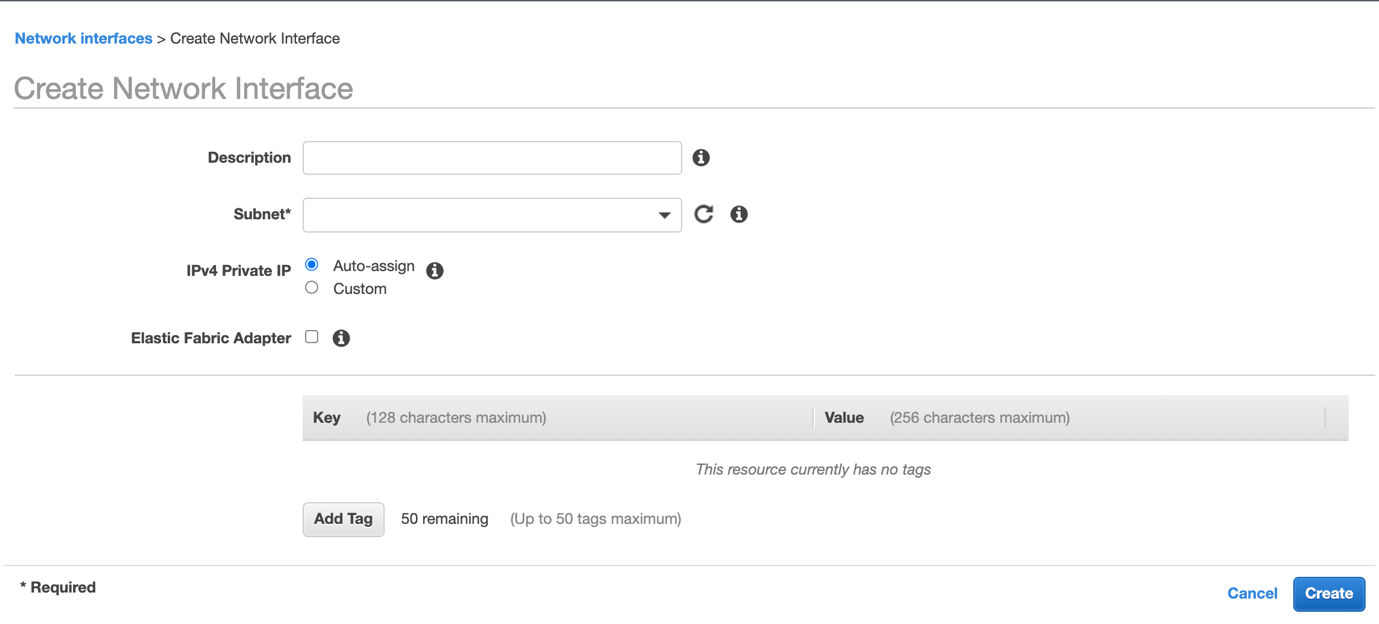
****

* **Placement Groups – Not using**
* **Key Pairs**
* **Network Interfaces**

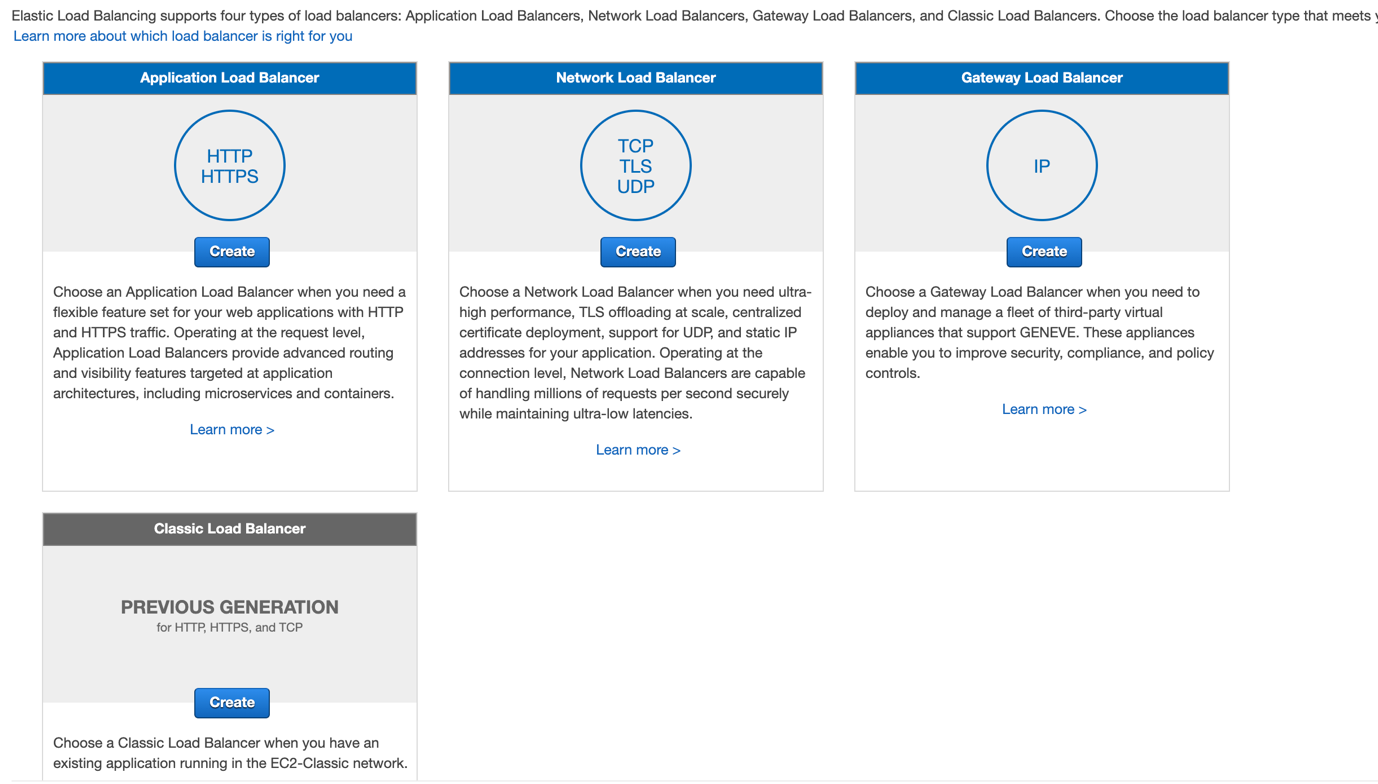
****

**To create new network interface**

**Select subnet**

****

* **LOAD BALANCING**

****

## **Application Load Balancer**

An Application Load Balancer makes routing decisions at the application layer (HTTP/HTTPS), supports path-based routing, and can route requests to one or more ports on each container instance in your cluster. Application Load Balancers support dynamic host port mapping. For example, if your task's container definition specifies port 80 for an NGINX container port, and port 0 for the host port, then the host port is dynamically chosen from the ephemeral port range of the container instance (such as 32768 to 61000 on the latest Amazon ECS-optimized AMI). When the task is launched, the NGINX container is registered with the Application Load Balancer as an instance ID and port combination, and traffic is distributed to the instance ID and port corresponding to that container. This dynamic mapping allows you to have multiple tasks from a single service on the same container instance. For more information, see the [User Guide for Application Load Balancers](https://docs.aws.amazon.com/elasticloadbalancing/latest/application/).


                        Application Load Balancer
                    

## **Network Load Balancer**

A Network Load Balancer makes routing decisions at the transport layer (TCP/SSL). It can handle millions of requests per second. After the load balancer receives a connection, it selects a target from the target group for the default rule using a flow hash routing algorithm. It attempts to open a TCP connection to the selected target on the port specified in the listener configuration. It forwards the request without modifying the headers. Network Load Balancers support dynamic host port mapping. For example, if your task's container definition specifies port 80 for an NGINX container port, and port 0 for the host port, then the host port is dynamically chosen from the ephemeral port range of the container instance (such as 32768 to 61000 on the latest Amazon ECS-optimized AMI). When the task is launched, the NGINX container is registered with the Network Load Balancer as an instance ID and port combination, and traffic is distributed to the instance ID and port corresponding to that container. This dynamic mapping allows you to have multiple tasks from a single service on the same container instance. For more information, see the [User Guide for Network Load Balancers](https://docs.aws.amazon.com/elasticloadbalancing/latest/network/).

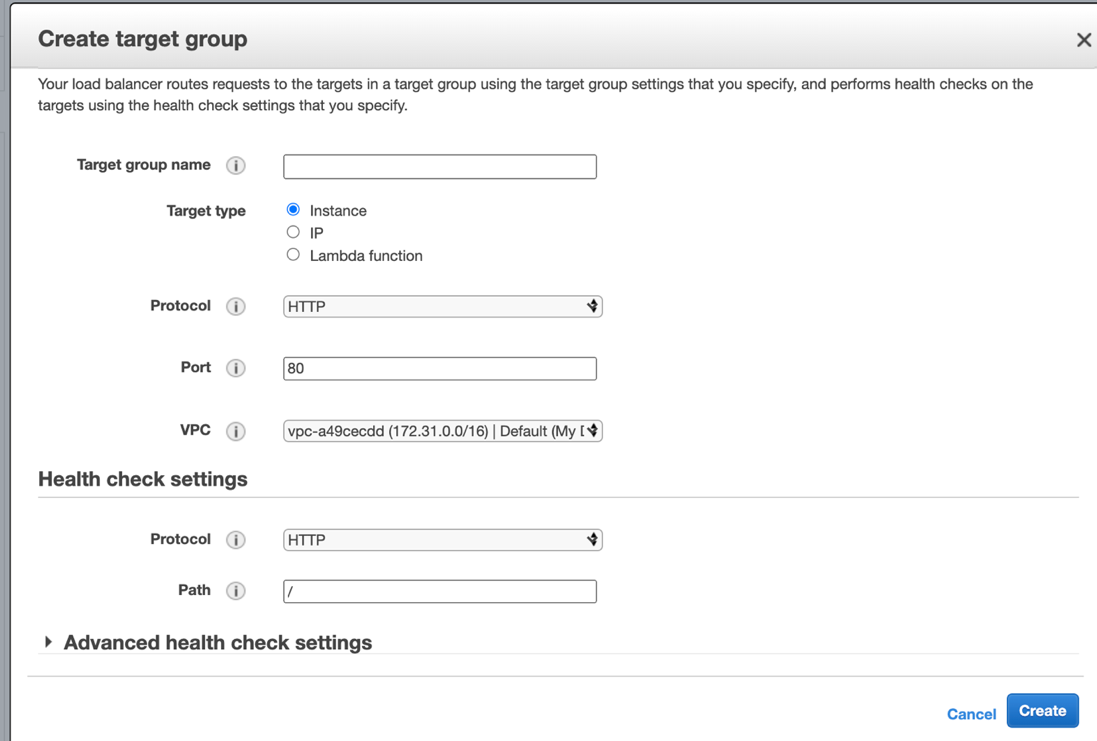

                        Network Load Balancer
                    

## **Classic Load Balancer**

A Classic Load Balancer makes routing decisions at either the transport layer (TCP/SSL) or the application layer (HTTP/HTTPS). Classic Load Balancers currently require a fixed relationship between the load balancer port and the container instance port. For example, it is possible to map the load balancer port 80 to the container instance port 3030 and the load balancer port 4040 to the container instance port 4040. However, it is not possible to map the load balancer port 80 to port 3030 on one container instance and port 4040 on another container instance. This static mapping requires that your cluster has at least as many container instances as the desired count of a single service that uses a Classic Load Balancer. For more information, see the [User Guide for Classic Load Balancers](https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/).


                        Classic Load Balancer
                    

**Target Groups**

****

* **AUTO SCALING -not using**

### **What is auto-scaling?**

[Auto-scaling](https://www.simplilearn.com/tutorials/aws-tutorial/aws-auto-scaling) is a function that allows you to provision and launch new instances whenever there is a demand. It allows you to automatically increase or decrease resource capacity in relation to the demand.

### **How do you upgrade or downgrade a system with near-zero downtime?**

### You can upgrade or downgrade a system with near-zero downtime using the following steps of migration:

* Open EC2 console
* Choose Operating System AMI
* Launch an instance with the new instance type
* Install all the updates
* Install applications
* Test the instance to see if it’s working
* If working, deploy the new instance and replace the older instance
* Once it’s deployed, you can upgrade or downgrade the system with near-zero downtime.

Elastic Block Store (EBS Instance Storage)

* EBS –(block storage) – we can increase until max 16TB but we cannot decrees volume
* EBS is a persistent storage
* We have different types of storages in ebs (SSD & HDD)
* While launching instance there will be EBS only cannot use instance store

vs

Instance store – 1x50 (SSD) with EBS

* + Instance storage is fixed . No Extra cost for instance store.
  + IS is a temporary storage. If you stop and start (not reboot) the data in the drive is lost
  + Instance storage is fixed depending on the more and it cannot be changed.

M3.medium instance

EFS

FSX

Simple storage services (s3)

Glaceir

Storage Gateway