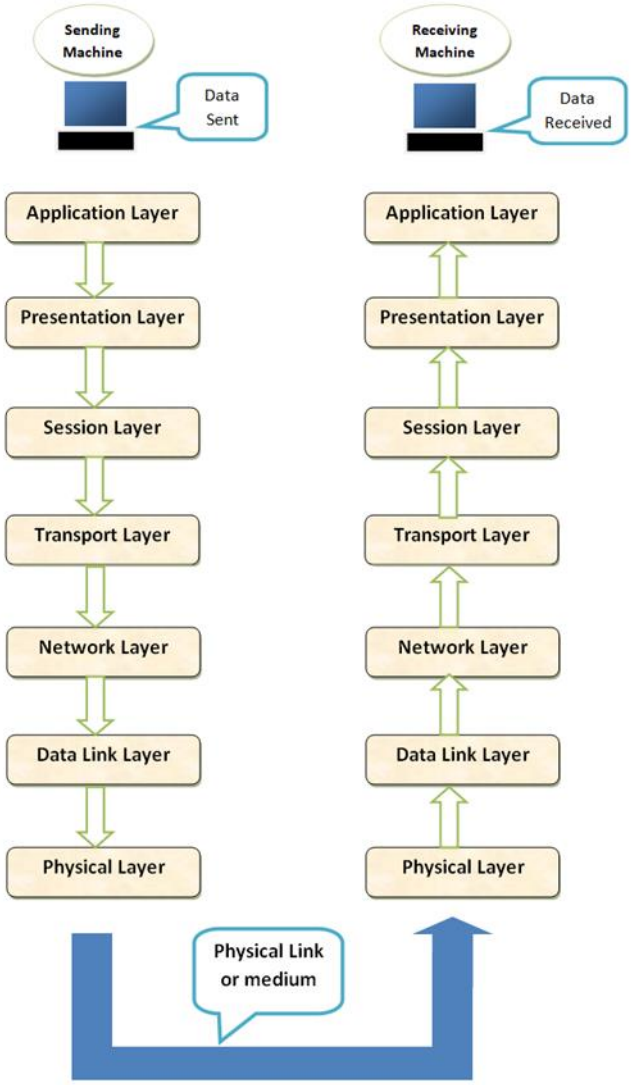
AWS ELB & ASG???

OSI (Open System Interconnection) layers of communication	Responsibility of 7 layers
	**1.Physical layer:** Physical transmission of data through cables, wires, or wireless signals. **2.Data link layer:** Accurate and reliable transmission of data (error correction bits). . **3. Network layer:** Routing data from one device to another on a network (**Route Table** is used). **4. Transport layer:** Reliable connection and transport of data in correct order. **5. Session layer:** Managing the communication Opening and closing connections between the devices. **6. Presentation layer:** Data formatting, compression, and encryption – data understood by the application layer. **7.Application Layer:** It provides network services to applications where user can access the resources.

Note:

Protocol	Description	Connection type	Use cases
HTTP	Hypertext Transfer Protocol	connectionless	Used to transfer hypertext documents, such as web pages, between a client (web browser) and a server.
HTTPS	Hypertext Transfer Protocol Secure	connection	A secure version of HTTP that uses Transport Layer Security (TLS) to encrypt data sent between a client and a server.
TCP	Transmission Control Protocol	connection	A reliable, connection-oriented protocol that guarantees the delivery of data.
UDP	User Datagram Protocol	connectionless	A connectionless, unreliable protocol that is not guaranteed to deliver data.

1. **Types of ELB??**

1. Application Load Balancer (ALB): (Path based)

- ALB is a layer 7 load balancer
- Good choice:
    - Traffics based on url paths, host name, (eg: path based traffic to containers, microservices)
    - Content based requests (inspects the inside content),

2. Network Load Balancer (ALB): (Port based)

- NLB is a Layer 4 load balancer
- TCP based traffic on the destination port
- provides ultra-low latency and high throughput
  (*throughput- how many units of information a system can process in a given amount of time)
- NLB is capable of handling millions of requests per second while maintaining high availability

3.Classic load balancer (ALB): (Port based)

- CLB is a Layer 4 load balancer
- Distributing traffic based on network information like IP addresses and ports.
- TCP or SSL protocol

Load Balancer	**ALB**	**NLB**	**CLB**
Throughput	High	Very High	Moderate to High
Latency	Low to Moderate	Very Low	Moderate
Supported Protocols	HTTP, HTTPS	TCP, UDP	TCP, SSL
performance	Very good performance, especially for HTTP and HTTPS traffic, with advanced routing capabilities.	Excellent performance, ultra-low latency, and high scalability for TCP and UDP-based protocols.	Good performance, but less scalable compared to ALB and NLB.

2. **CLB Vs ALB?**

	CLB (Port based)	**ALB (Path based)**
1	Operates at Layer 4 of the OSI model	Operates at Layer 7 of the OSI model
2	It operates at the transport layer (TCP/SSL)	Operates at the application layer (HTTP/HTTPS) of the OSI model (Traffic based on the content of the HTTP/HTTPS requests)
3	**Basic load balancing features,** 1. Traffic across multiple EC2 instances in different availability zones to improve availability and fault tolerance.	**Advanced Load Balancing Features** • Path-based routing, • Host-based routing, • Content-based routing, • Support for **websockets** (chat apps) • integrates with AWS Lambda - provides advanced routing and load balancing capabilities for modern applications
CLB is the older load balancer in AWS and primarily focuses on distributing traffic at the transport layer. **ALB** is a more feature-rich load balancer designed for more granular routing and load balancing decisions (**modern application architectures**)		

3. **ALB Vs NLB?**

	ALB (path based)	**NLB (Port based)**
1	layer 7 load balancer	Layer 4 load balancer
2	operates at the application layer (HTTP/HTTPS) of the OSI model	operates at the transport layer (TCP/UDP) of the OSI model
3	ALB is designed for modern application architectures	NLB is designed to handle high volumes of traffic and provides ultra-low latency and high throughput
4	**Uniqueness:** Suitable for applications with complex routing requirements (Traffic based on the content of the HTTP/HTTPS requests)	**Uniqueness:** Handle millions of requests per second (high-performance load balancing)
5.	ALB supports features such as path-based routing, host-based routing, content-based routing, and integration with AWS services like AWS Lambda.	NLB supports static IP addresses for maintaining a consistent endpoint (provides integration with Elastic IP addresses)
NLB is ideal for scenarios that require high throughput and low latency, ALB is suitable for modern applications that require flexible routing and advanced features		

4. **What is GLB (Gateway Load Balancer)?**
   - **GLB is needed:** To set up and run a group of network virtual appliances (NVAs) from third parties that support GENEVE. You can improve security, compliance, and policy rules with the help of these appliances.
   - It aims for high performance and high availability.

- Gateway Load Balancer is a cloud service that enables the deployment and management of network virtual appliances (NVAs). It works at the network layer (3rd layer) of the OSI model, monitoring all IP packets across all ports. It routes traffic to the appropriate target group based on the listener rule.

s.no.	Network Virtual Appliance (NVAs)	Functions
1.	Firewall Appliances	Advanced network security, traffic inspection, threat prevention
2.	Intrusion Detection and Prevention Systems (IDPS)	Monitoring network traffic, detecting and preventing intrusions, malicious activity prevention
3.	Web Application Firewalls (WAF)	Protecting web applications, mitigating common vulnerabilities, filtering and blocking malicious traffic
4.	Load Balancers	Traffic distribution across backend servers, scalability and high availability
5.	VPN Appliances	Secure remote access connections, site-to-site VPN connections
6.	Network Monitoring Appliances	Network performance monitoring, traffic analysis, troubleshooting network issues
7.	DDoS Protection System	Detecting and mitigating Distributed Denial of Service (DDoS) attacks, protecting network resources and availability

5. What is Stickiness in LB?

Session stickiness, also known as session affinity, is a feature in load balancers that ensures that when you visit a website or use an application, your requests are consistently routed to the same backend server throughout your session. This is important because it allows the server to remember information specific to you, like your login credentials or the items in your shopping cart.

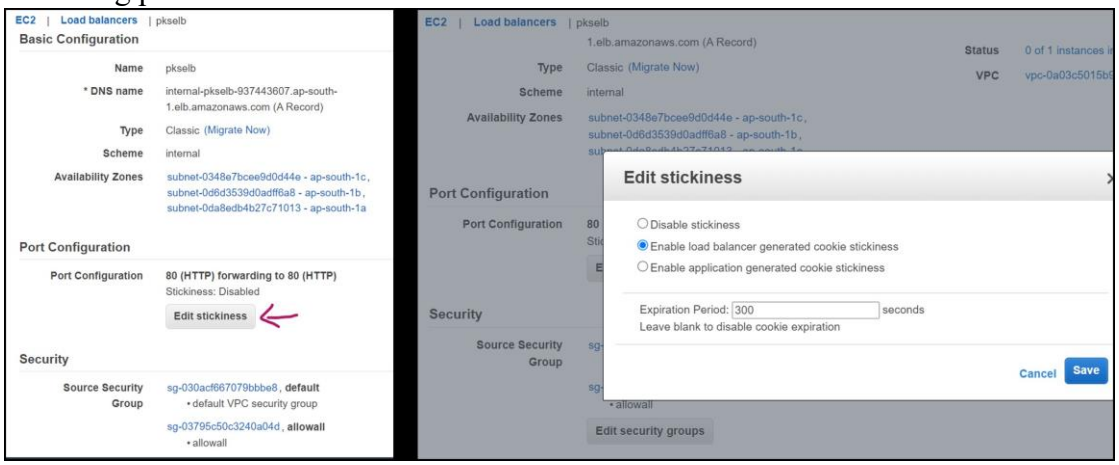
When session stickiness enabled, the load balancer can identify you as a unique client and direct your requests to the same server each time. This helps maintain your personalized data and context.

Some use-cases:

E-commerce websites: Maintaining consistent shopping carts throughout the user's session.

Authentication and authorization: Preserving user authentication state across multiple requests.

Stateful applications: Ensuring uninterrupted collaboration and real-time updates in applications.
Gaming platforms: Keeping players connected to the same game server for synchronized gameplay.
Personalization and user preferences: Retaining user-specific settings and preferences during a session.
Streaming services: Preserving playback state and preferences in video or audio streaming platforms.



6. **Differentiate between Launch configuration & Launch template in AWS**?

Feature	**Launch Configuration**	**Launch Template**
Definition	A launch configuration is a JSON or YAML file that defines the configuration of an EC2 instance.	A launch template is a JSON or YAML file that defines the configuration of an EC2 instance. It also supports **versioning.**
Use	Launch configurations are used to create Auto Scaling groups.	Launch templates can be used to create Auto Scaling groups, launch EC2 instances directly, or be used as a reference in other AWS services.
Versioning	Launch configurations **cannot** be versioned.	Launch templates **support** versioning, which allows you to track changes to the template over time.
Recommendation	AWS recommends using launch templates instead of launch configurations.	AWS recommends using launch templates instead of launch configurations.
Support for new features	Launch configurations do not support all new features that are released for EC2.	Launch templates support all new features that are released for EC2. (refer next table for more details:)

Feature	Launch Configuration	Launch Template
EC2 instance types	Supports only instance types that were released before December 31, 2022.	Supports all instance types, including those that are released in the future.
Purchase options	Supports only On-Demand Instances.	Supports both On-Demand Instances and Spot Instances.
Dedicated Hosts	Does not support Dedicated Hosts.	Supports Dedicated Hosts.
EBS encryption	Supports only EBS encryption with default keys.	Supports EBS encryption with default keys and customer-managed keys.
IAM roles	Supports only IAM roles that were created before December 31, 2022.	Supports all IAM roles, including those that are created in the future.
Other features	May not support other new features that are released for EC2.	Supports all new features that are released for EC2.

7. **What is Grouping Size & Target Scaling?**

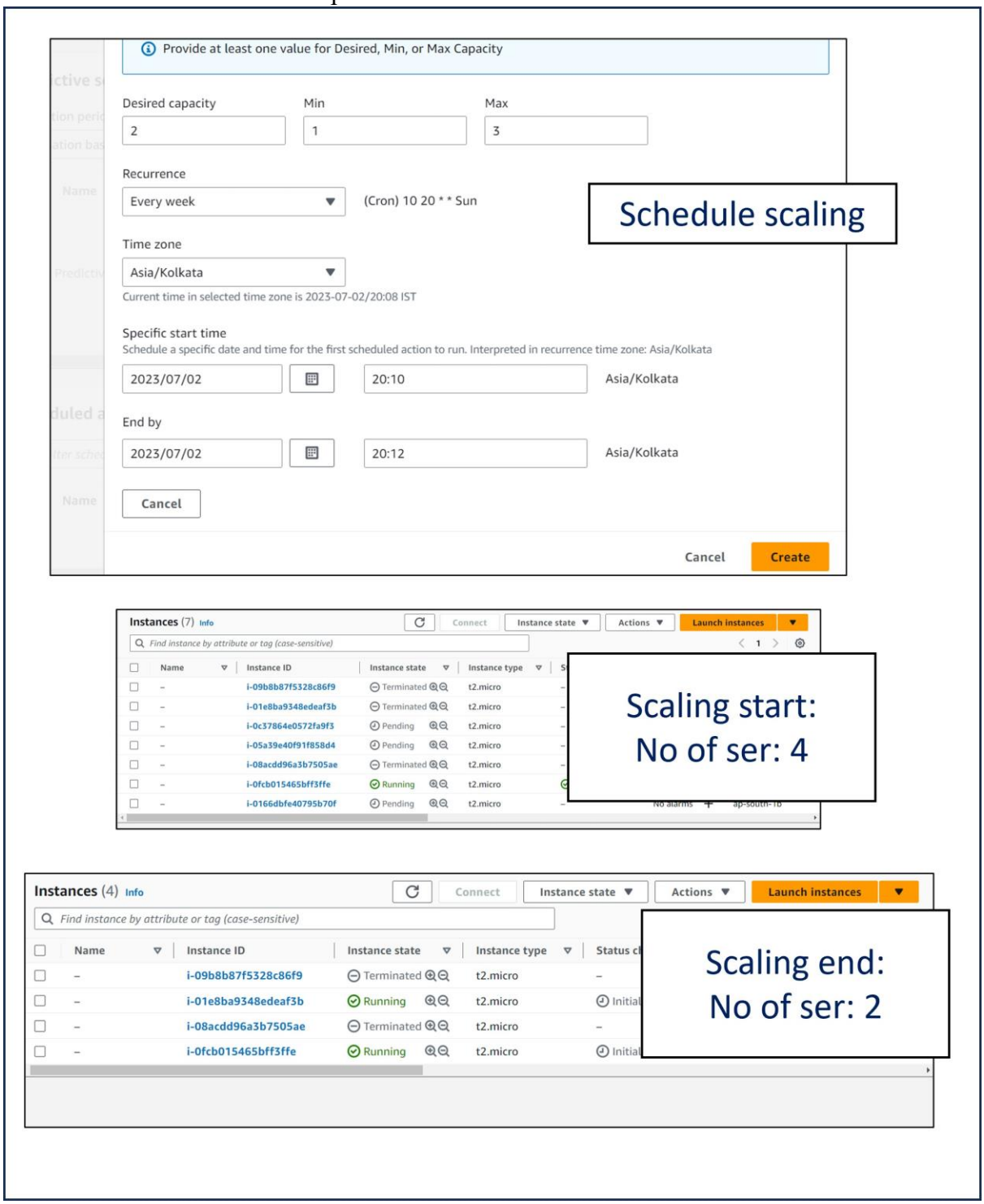
   Grouping size refers to the number of instances that are in an Auto Scaling group. The desired capacity is the initial size of the group, and you can adjust it up or down as needed. The minimum capacity and maximum capacity are the limits on how small or large the group can be.

   Target scaling refers to the way that Auto Scaling adjusts the size of the group based on a metric. There are two types of target scaling: **step scaling and target tracking scaling.**

   - **Step scaling** adjusts the size of the group in steps. For example, you could configure Auto Scaling to add one instance when the CPU utilization reaches 80%, and then add another instance when the CPU utilization reaches 90%.
   - **Target tracking scaling** adjusts the size of the group to maintain a target value for a metric. For example, you could configure Auto Scaling to maintain a target CPU utilization of 50%. If the CPU utilization goes above 50%, Auto Scaling will add instances. If the CPU utilization drops below 50%, Auto Scaling will remove instances.

8. **How to configure ASG at 9AM MONDAY ?.**

Time I chose is 2.07.2023 8.10 pm



9. What is Scale-in & Scale-Out?

	Scale In	Scale Out
Definition	Reducing the size or capacity of a system	Increasing the size or capacity of a system
Purpose	Optimize costs, improve efficiency	Accommodate growth, handle increased workloads
Resource Change	Remove resources (servers, storage, network)	Add resources (servers, storage, network)
Goal	Reduce infrastructure and operational costs	Improve system performance and handle increased demand
Main Focus	Cost reduction and efficiency	Scalability and performance improvement
Suitability	Decreased user traffic, lower workload, cost optimization	Increased user traffic, higher workload, handle growth
Potential Benefits	Cost savings, simplified management	Better scalability, improved performance, increased capacity
Potential Challenges	Limited system capacity	Increased complexity, higher costs, management challenges

10. Differentiate Horizontal scaling and Vertical scaling?

Horizontal scaling	Vertical scaling
Adds more machines or nodes to a system.	Adds more power (CPU, RAM, storage, etc.) to an existing machine.
Also known as "scaling out".	Also known as "scaling up".
Ideal for handling increasing amounts of traffic or workload.	Ideal for **handling resource-intensive tasks or applications that require more** processing power.
More scalable, as there is no upper limit to the number of machines that can be added.	**Less scalable**, as there is an upper limit to the amount of power that can be added to a single machine.
Easier to implement and manage.	More difficult to implement and manage, as changes need to be made to the underlying software.
Typically less expensive, as it does not require the purchase of new hardware.	Typically more expensive, as it requires the purchase of new hardware.

11.Is it possible to achieve vertical scaling in AWS?

Vertical scaling in AWS involves increasing or decreasing the resources of a single instance, and it can be done using services like Amazon EC2 and Amazon RDS. In EC2,

you can modify the instance type to scale up or down, while in RDS, you can adjust the instance class to allocate more or fewer resources.

12. How will you monitor your ELB?

Ways of monitoring Elastic Load Balancer (ELB) in AWS:

CloudWatch Metrics: Use CloudWatch to access pre-configured metrics for ELB, such as request count, latency, error rates, and backend instance metrics

CloudWatch Alarms: Set up threshold-based alarms in CloudWatch to receive notifications to identify and respond to performance issues.

Access Logs: Enable ELB access logging to generate detailed logs for each request, providing insights into traffic patterns, errors, and troubleshooting opportunities. Store logs in an Amazon S3 bucket.

AWS CloudTrail: Enable CloudTrail to monitor and record API activity related to your ELB, including configuration changes and load balancer requests, for audit and security purposes.

By monitoring, you can effectively track the performance, availability, and overall health of your Elastic Load Balancer, ensuring smooth operation and timely response to any issues.

(Note: CloudWatch is a monitoring service that provides metrics and alerts for AWS resources, while CloudTrail is a logging service that records API activity and actions taken within an AWS account.)

==========

VPC???

1.No of VPCs in a Region?

The default limit for the maximum number of VPCs in a region is 5. This limit is made up of the primary CIDR block plus 4 secondary CIDR blocks.

2. No of Subnets in a VPC?

Maximum number of subnets can be created for a region is 200

Maximum number of subnets can be created for an account is 500.

3. No of SG in VPC?

Maximum no of security groups in a VPC:500

Maximum no of security groups in a region:2500

4. NO of NACL in VPC?

Each VPC can have upto 200 NACLs

5. How many inbound rules & Outbound rules in SG & NACL?

The maximum number of inbound and outbound rules: 60 / security group

The maximum number of inbound and outbound rules: 20 per NACL

6. Use Jump Server/bastion host to connect your Private ec2?

Vpc with three subnets,1 IGW, 3 route tables, 3 security groups, 3 servers(1 baston server, 1 private httpd server, 1 private mysql server)

Aim :1. To login to baston server with public ip

2. Connecting to httpd private server and install httpd (with natgateway routed to RT-2)

3. Connecting to mysql private server and install mysql (with natgateway routed to RT-3)

After creating vpc infrastructure:

login as: ec2-user

[ec2-user@ip-10-0-1-123~]$ sudo -i

[root@ip-10-0-1-123 ~]# vi test.pem

[root@ip-10-0-1-123~]# chmod 400 test.pem

[root@ip-10-0-1-123~]# ssh -i test.pem ec2-user@10.0.3.221.

From baston server connected to httpd server(2)

```
[root@ip-10-0-1-123 ~]# ssh -i test.pem ec2-user@10.0.3.221
Last login: Tue Jul  4 05:04:21 2023 from 10.0.1.123

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-3-221 ~]$ sudo -i
```

[ec2-user@ip-10-0-03-221 ~]$ sudo -i

[ec2-user@ip-10-0-03-221 ~]$ vi test.pem

[ec2-user@ip-10-0-03-221 ~]$ chmod 400 test.pem

Natgateway: net connected in httpd server

```
[root@ip-10-0-3-221 ~]# ping google.com
PING google.com (142.250.192.14) 56(84) bytes of data.
64 bytes from bom12s14-in-f14.1e100.net (142.250.192.14): icmp_seq=1 ttl=50 time
=3.24 ms
64 bytes from bom12s14-in-f14.1e100.net (142.250.192.14): icmp_seq=2 ttl=50 time
=2.61 ms
64 bytes from bom12s14-in-f14.1e100.net (142.250.192.14): icmp_seq=3 ttl=50 time
=2.58 ms
64 bytes from bom12s14-in-f14.1e100.net (142.250.192.14): icmp_seq=4 ttl=50 time
=2.58 ms
64 bytes from bom12s14-in-f14.1e100.net (142.250.192.14): icmp_seq=5 ttl=50 time
=2.60 ms
64 bytes from bom12s14-in-f14.1e100.net (142.250.192.14): icmp_seq=6 ttl=50 time
=2.61 ms
^C
--- google.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 2.584/2.708/3.242/0.240 ms
[root@ip-10-0-3-221 ~]#
```

With Natgateway: httpd also installed:

```
Installed:
  httpd.x86_64 0:2.4.57-1.amzn2

Dependency Installed:
  apr.x86_64 0:1.7.2-1.amzn2
  apr-util.x86_64 0:1.6.3-1.amzn2.0.1
  apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1
  generic-logos-httpd.noarch 0:18.0.0-4.amzn2
  httpd-filesystem.noarch 0:2.4.57-1.amzn2
  httpd-tools.x86_64 0:2.4.57-1.amzn2
  mailcap.noarch 0:2.1.41-2.amzn2
  mod_http2.x86_64 0:1.15.19-1.amzn2.0.1

Complete!
[root@ip-10-0-3-221 ~]#
```

<u>Connected to sql server (3)from httpd server:</u>

[root@ip-10-0-3-221 ~]# ls

test.pem

[root@ip-10-0-3-221 ~]# ssh -i test.pem ec2-user@10.0.5.5

```
[root@ip-10-0-3-221 ~]# ssh -i test.pem ec2-user@10.0.5.5
The authenticity of host '10.0.5.5 (10.0.5.5)' can't be established.
ECDSA key fingerprint is SHA256:IwSDilqknKEvTgnCG7ZjioNV7YXzJ9s/jeJ/5XUV15o.
ECDSA key fingerprint is MD5:69:1b:eb:49:86:0e:29:4d:63:00:f9:de:f8:ca:9a:ae.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.5.5' (ECDSA) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-5-5 ~]$
```

With natgateway: Connected internet in the mysql server

```
[root@ip-10-0-5-5 ~]# ping google.com
PING google.com (142.250.183.46) 56(84) bytes of data.
64 bytes from bom12s11-in-f14.1e100.net (142.250.183.46): icmp_seq=1 ttl=50 time=2.45 ms
64 bytes from bom12s11-in-f14.1e100.net (142.250.183.46): icmp_seq=2 ttl=50 time=2.23 ms
64 bytes from bom12s11-in-f14.1e100.net (142.250.183.46): icmp_seq=3 ttl=50 time=2.18 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 2.182/2.290/2.455/0.124 ms
[root@ip-10-0-5-5 ~]#
```

With Natgateway: httpd also installed:

```
Total download size: 8.8 M
Installed size: 49 M
Downloading packages:
mariadb-5.5.68-1.amzn2.0.1.x86_64.rpm
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 1:mariadb-5.5.68-1.amzn2.0.1.x86_64
  Verifying  : 1:mariadb-5.5.68-1.amzn2.0.1.x86_64

Installed:
  mariadb.x86_64 1:5.5.68-1.amzn2.0.1

Complete!
```

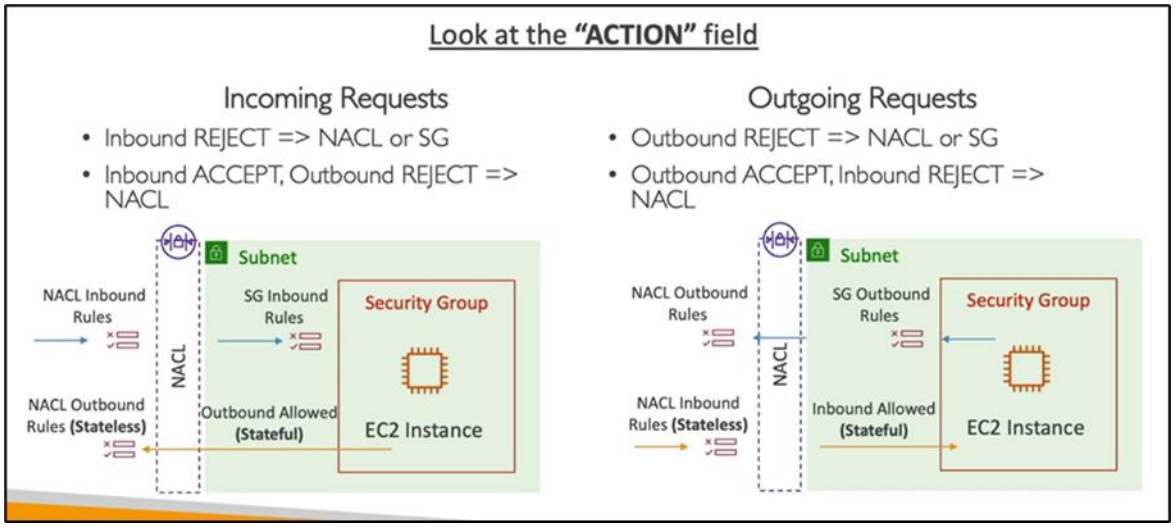
7.How to enable VPC Flow logs & Export the logs to AWS S3?

VPC Flow Logs

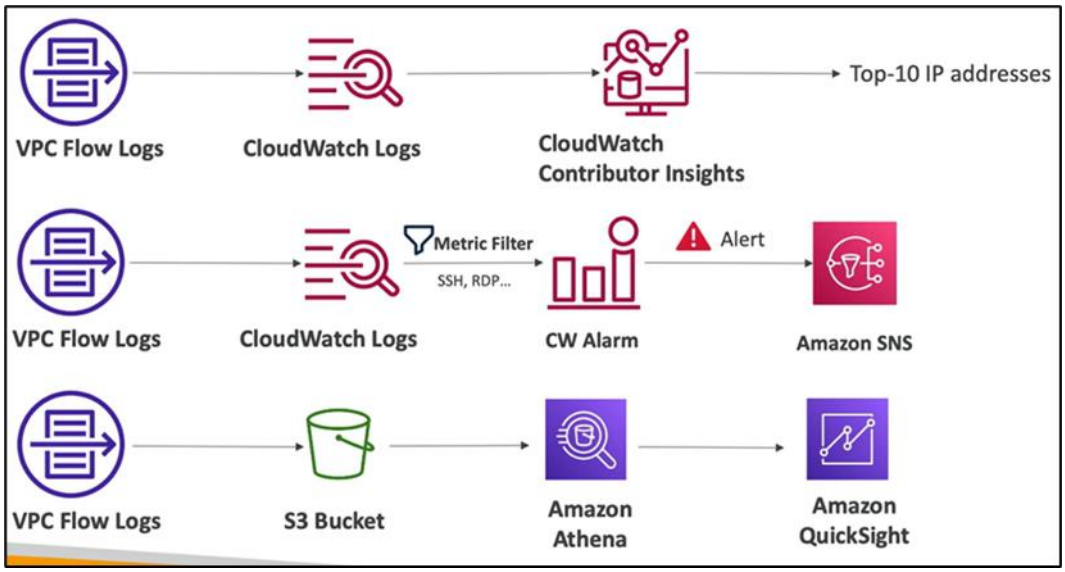
- Capture information about IP traffic going into your interfaces:
  1. VPC Flow Logs
  2. Subnet Flow Logs
  3. Elastic Network Interface (ENI) Flow Logs
- Helps to monitor & troubleshoot connectivity issues
- Flow logs data can go to S3 / CloudWatch Logs
- Captures network information from AWS managed interfaces too: ELB, RDS, ElastiCache, Redshift, WorkSpaces, NATGW, Transit Gateway.

- **srcaddr & dstaddr** — help identify problematic IP
- **srcport & dstport** — help identity problematic ports
- **Action** — success or failure of the request due to Security Group / NACL
- Flow logs Can be used for analytics on usage patterns, or malicious behavior.
- **Query** VPC flow logs using Athena on S3 or CloudWatch Logs Insights

- **Flow Logs examples:** https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs-records-examples.html



VPC flow log architecture:



Hands on -steps:

Step 1: VPC creation with 2 subnets, bastion server, private ec2, 2 RTs, 2 SGs, 1 IGW, 1 NAT GW

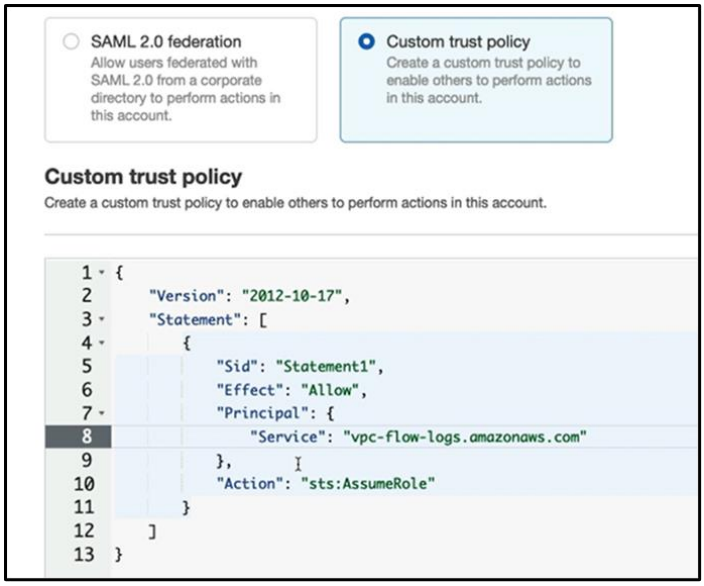
Step 2:

1.flow log creation with CloudWatch flowlogs

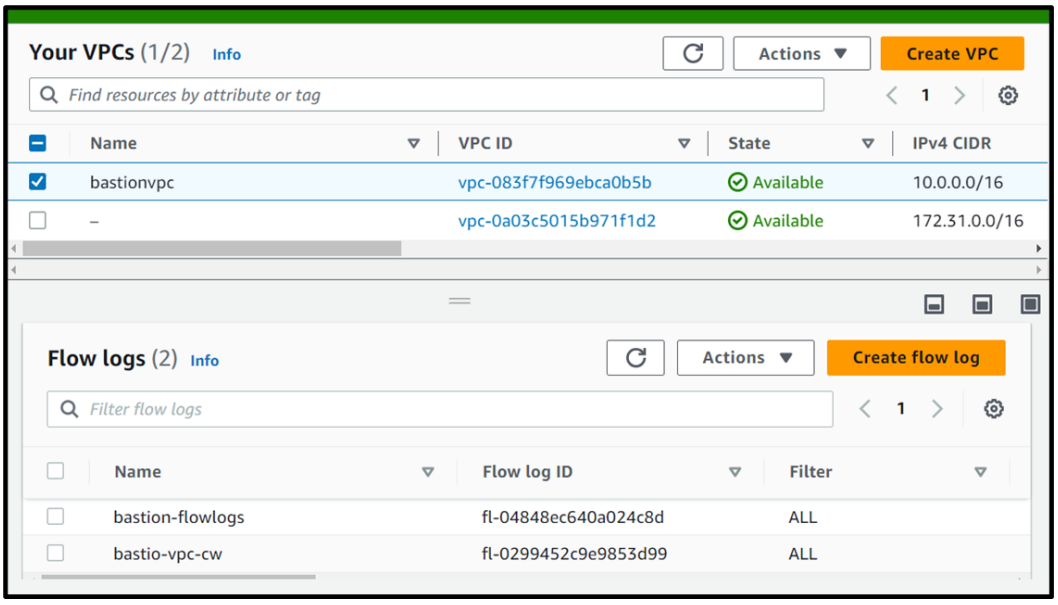
2. flow log creation with amozon S3 and anthena editor

1.VPC flow log creation with CloudWatch flowlogs

a) Log group creation using cloud watch logs
b) Role creation with custom policy: “Service”: “vpc-flow-logs.amazonaws.com”, with **“cloudwatchrolefullaccess”**

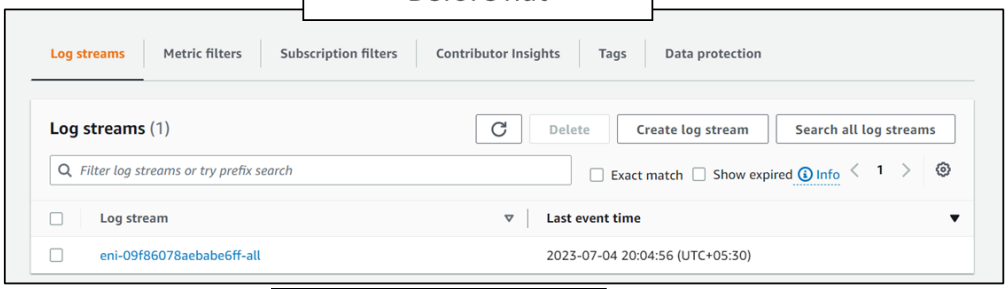


c) Now include above details in the VPC flow log creation step
d) After creating flowlogs : If we refresh VPC flow logs, entries got updated

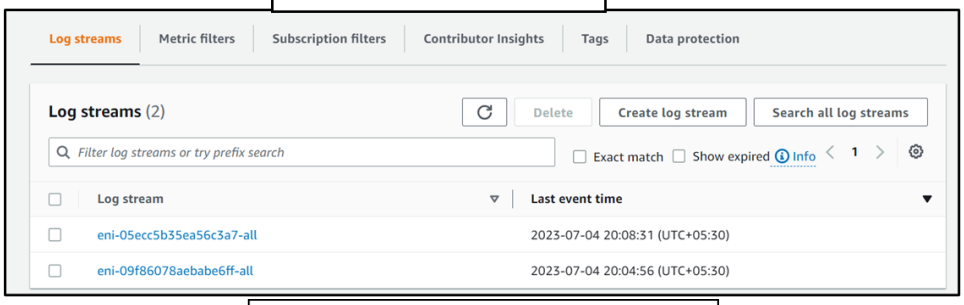


e) Go to flowlog groups in the cloud watch page. Refresh and fetch the details of log streams

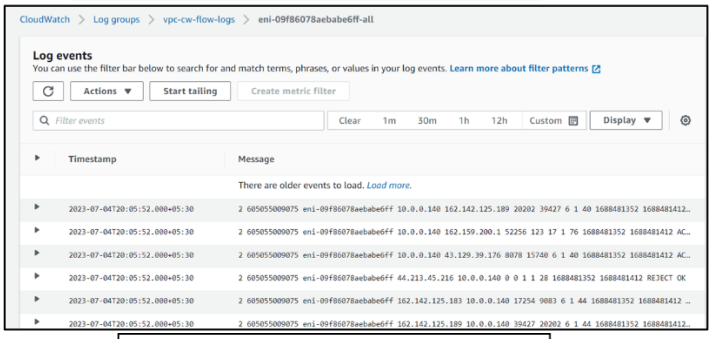
Before nat



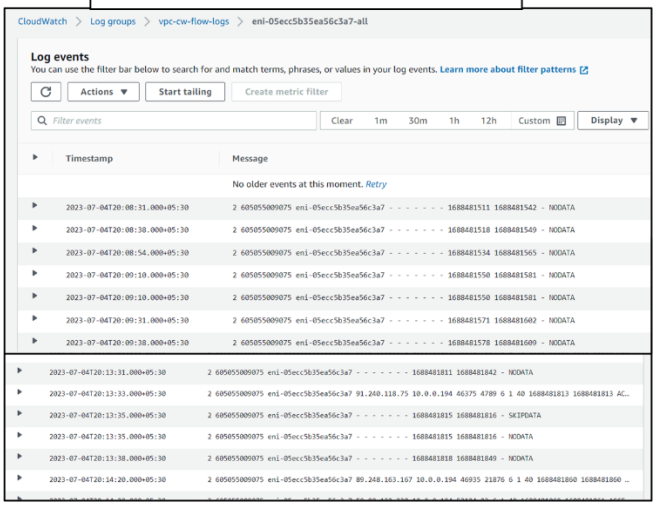
After nat- connection



Cw flow-logs: bastion server



Cw flow-logs: Private ec2

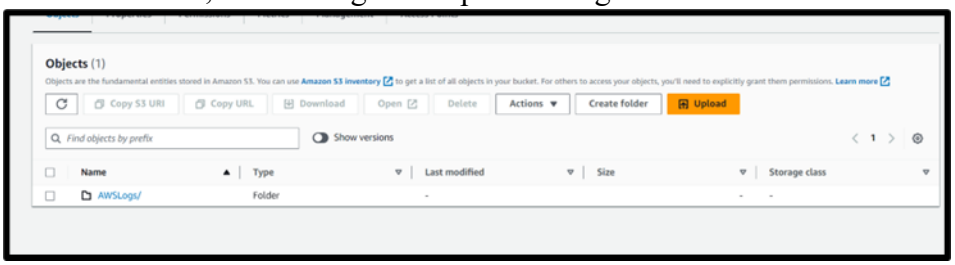


2. Flow log creation with amozon S3 and anthena editor

Part 1:

a) Flow log creation with s3 bucket creation (create a bucket and give the arn address in box)

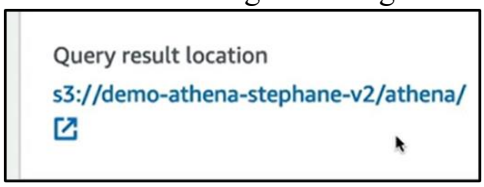
b) Goto s3 bucket , refresh to get the uploaded logs.



Part 2:

a) S3 bucket creation for anthena results storing location

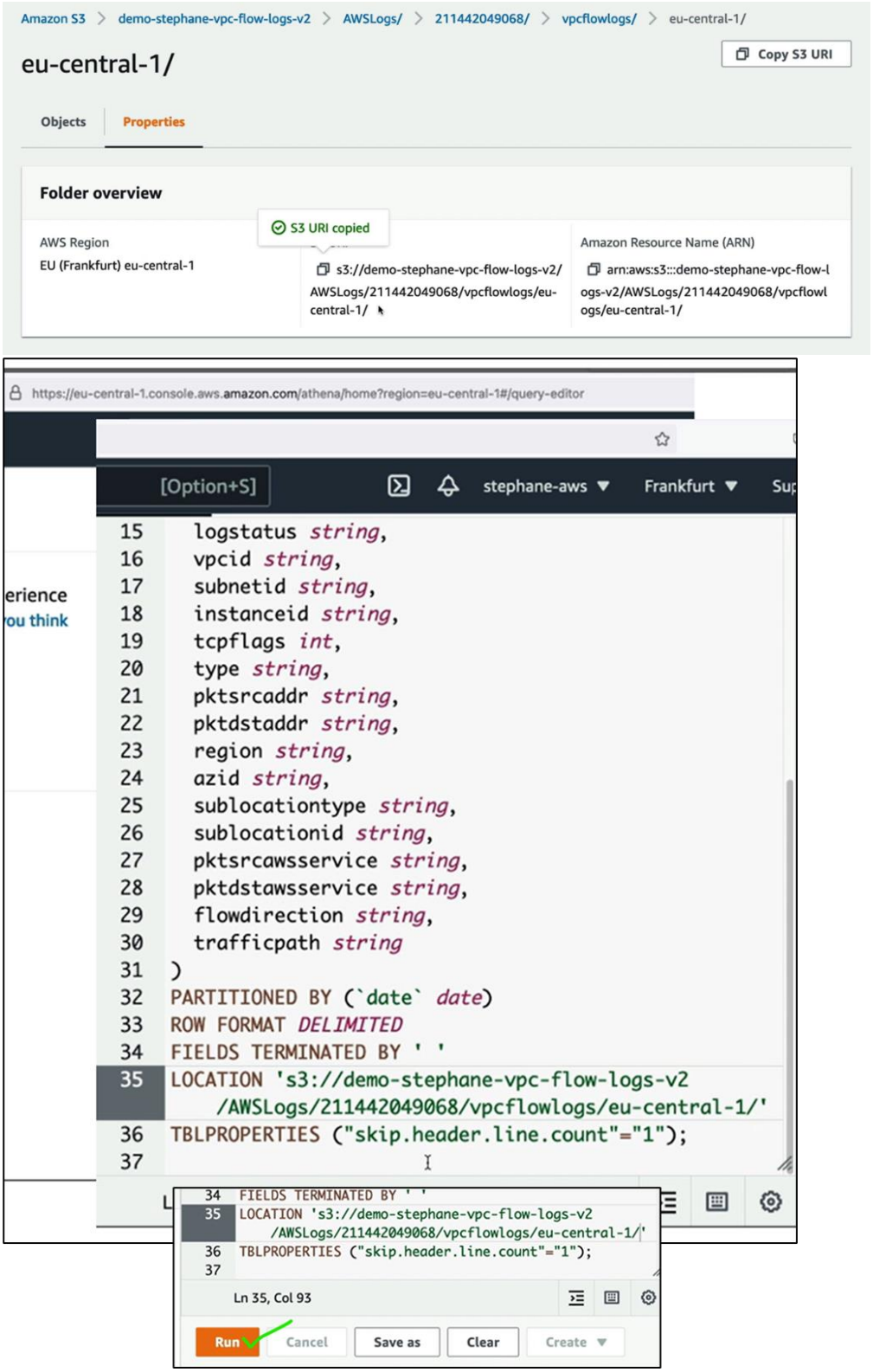
b) Go to anthena and goto settings and mention above created bucket location



c) Save and goto editor where program statements for table creation is required to get results.

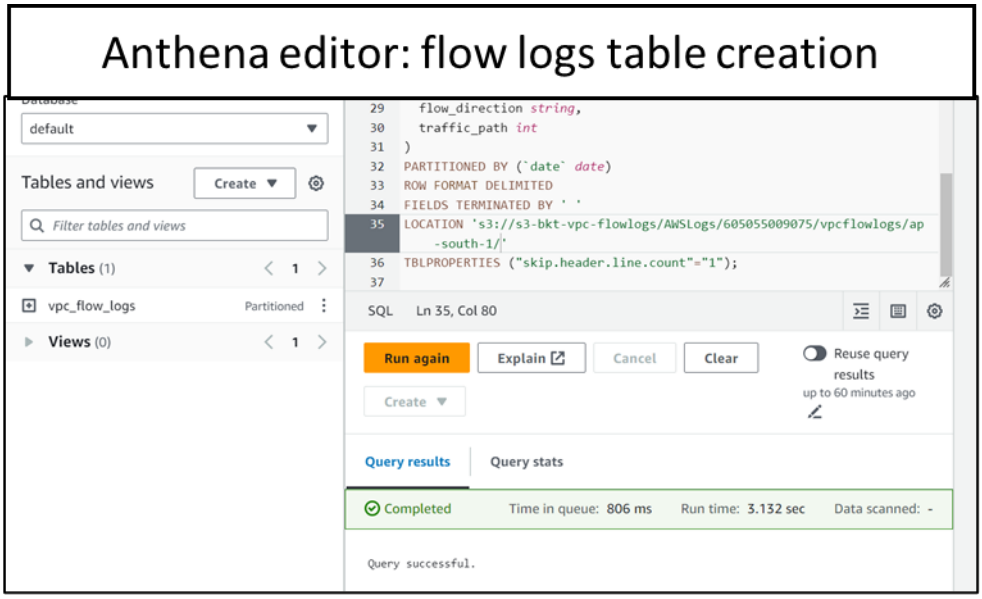
d) Goto this link: https://docs.aws.amazon.com/athena/latest/ug/vpc-flow-logs.html

e) Copy the first program statement, edit location, paste and run in the anthena editor --- flow logs will be created

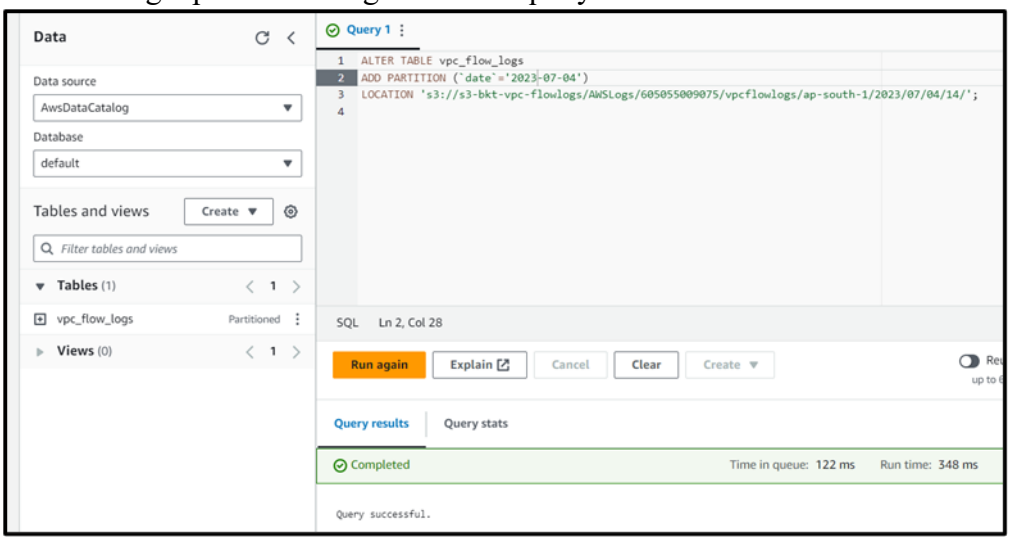


Copying and modifying the s3 bucket location

f) After editing the location, run the statement



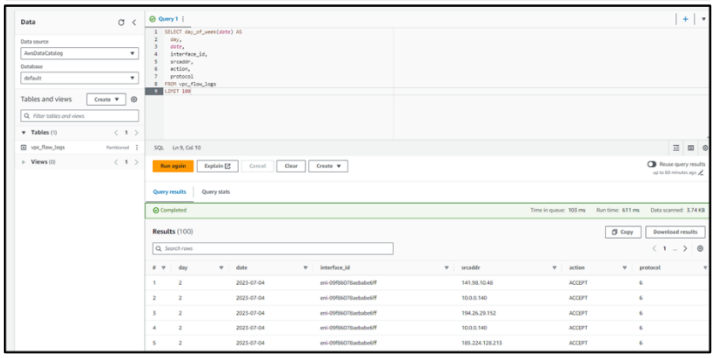
g) create a single partition using the below query



h) This query lists all of the all TCP connections

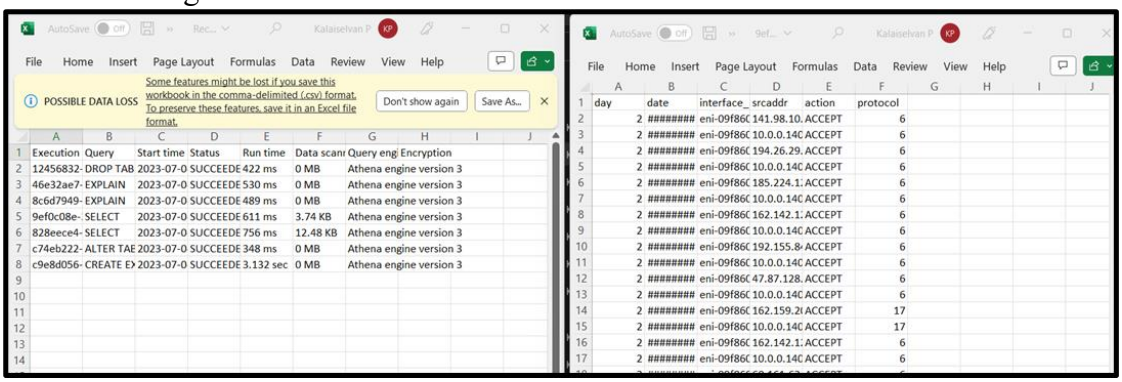
```
SELECT day_of_week(date) AS
  day,
  date,
  interface_id,
  srcaddr,
  action,
  protocol
FROM vpc_flow_logs
LIMIT 100
```

Anthena editor: This query lists all of the all TCP connections

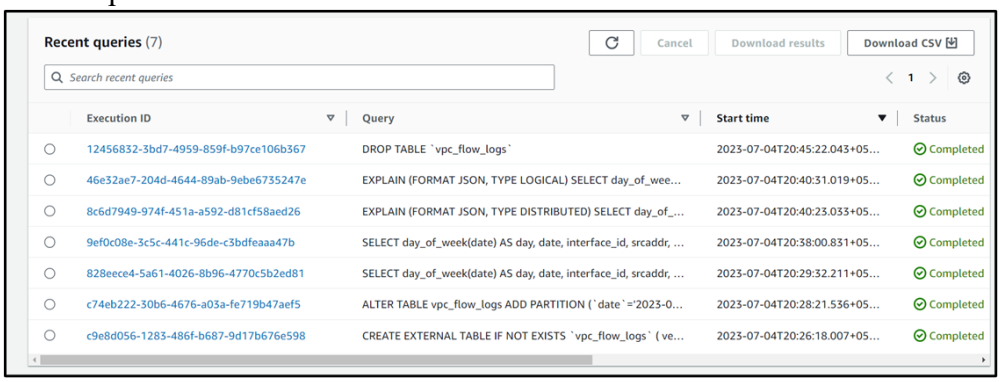


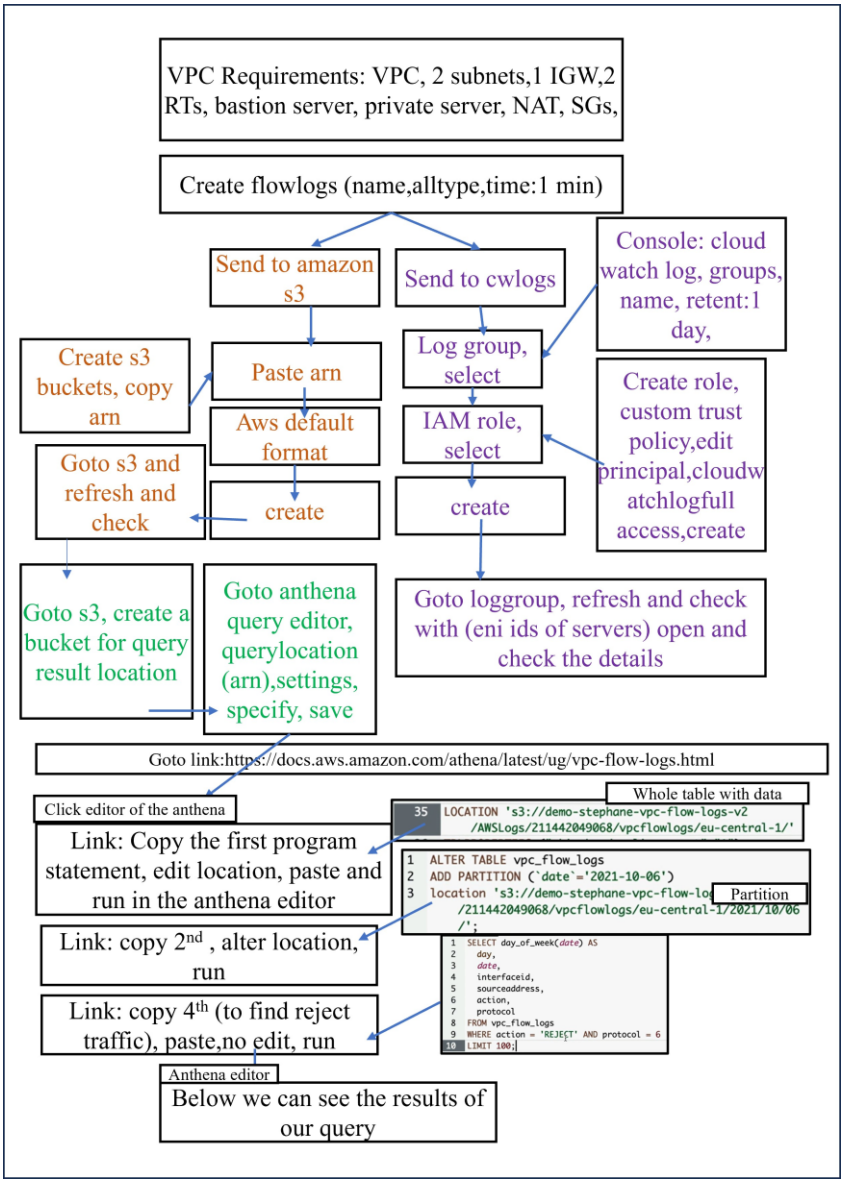
i) Query stats:

j) Csv files of logs:



k) Recent queries





8.Is it possible to edit VPC CIDR & Subnet CIDR?

It's not possible to change or modify the IP address range of an existing virtual private cloud (VPC) or subnet. However, we can do one of the following:

- Add an additional IPv4 CIDR block as a secondary CIDR to your VPC.
- Create a new VPC with your preferred CIDR block and then migrate the resources from your old VPC to the new VPC (if applicable)1.

Note : cidr values and respective ip,s:

https://www.freecodecamp.org/news/subnet-cheat-sheet-24-subnet-mask-30-26-27-29-and-other-ip-address-cidr-network-references