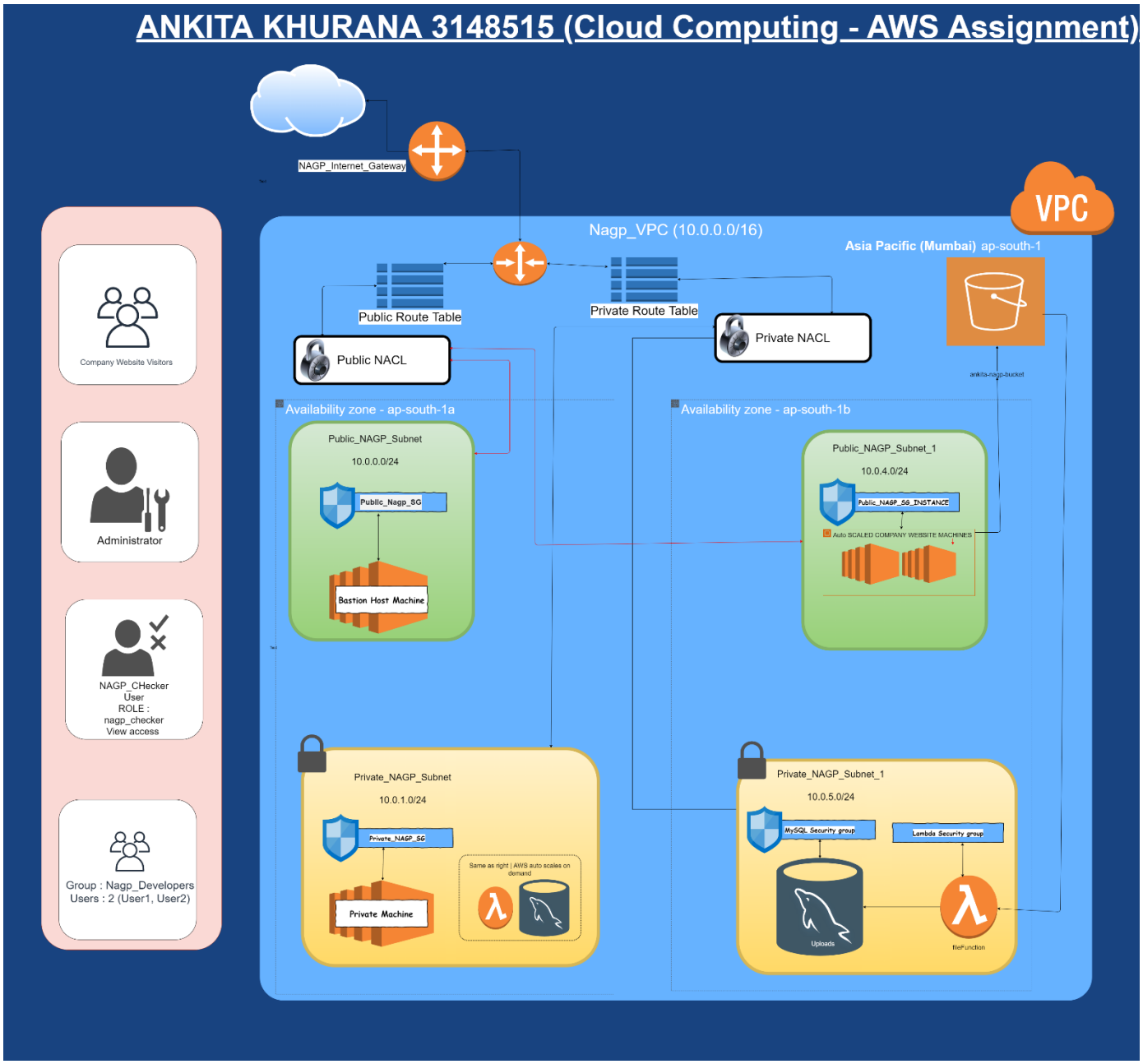
CLOUD COMPUTING ASSIGNMENT 2

- Ankita Khurana
- 3148515

Note:

1. **As mentioned in the group (No NAT Gateway) is added to the architecture due to non-free service.**

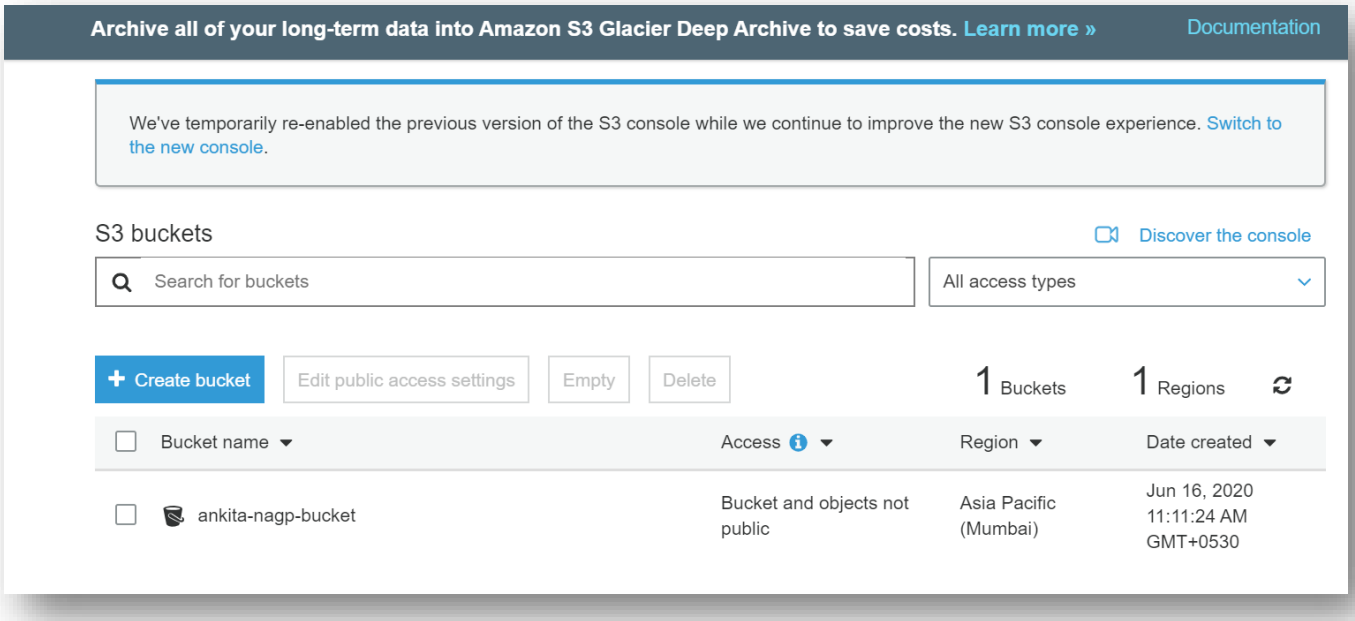


ARCHITECTURE

RESOURCES AND SERVICES used till date for XYZ project

1. **S3 Bucket: ankita-nagp-bucket**

- S3 bucket created in Asia Pacific region to store files uploaded via the company website.
- S3 bucket data is extracted in the company website to view data.
- Any upload of file in the bucket Triggers Lambda function (fileFunction)



2. **EC2 Machines :** (Bastion host on Public subnet + Private Ec2 on private subnet + Auto Scale company website hosted on machines on Public subnet 1)

- One EC2 machine acts as a Bastion host to connect to the private ec2 of company
- It also acts as an SSH layer to connect the admin to the RDS-MYSQL inside the private subnet
- The AUTO SCALING Machines which hosts the Company Website are present on the public network and load balanced via target group

a. Logging into Public Bastion Host

```
ankitakhurana@Ankita3148515 MINGW64 ~/Desktop
$ ssh -i "nagppublic.pem" ec2-user@13.234.136.193
Last login: Tue Jun 16 03:29:09 2020 from 106.215.82.111

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

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
```

b. Logging into Private Machine from Bastion Host

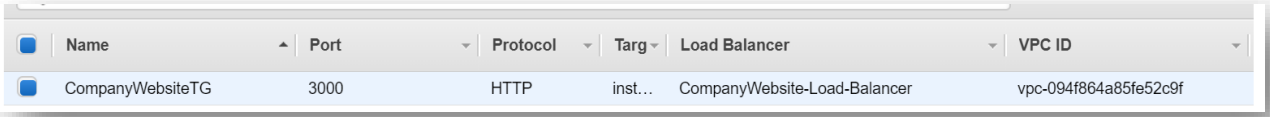
```
[ec2-user@ip-10-0-0-139 ~]$ chmod 400 key.pem
[ec2-user@ip-10-0-0-139 ~]$ ssh -i "key.pem" ec2-user@10.0.1.53

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

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
[ec2-user@ip-10-0-1-53 ~]$
```

 # Auto Scaling – user data

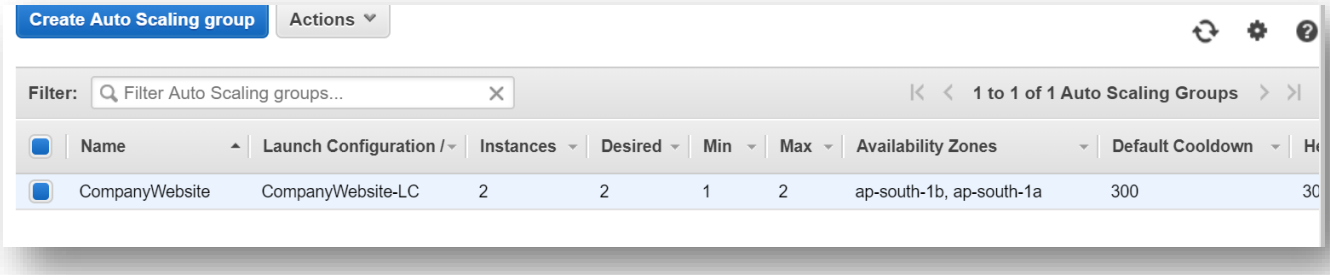
```
#!/bin/bash
sudo yum update -y
sudo yum install git -y
cd /home/ec2-user
git clone https://github.com/AnkitaKhurana/fileuploader.git
sudo chmod a+rwx fileuploader
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash
sudo chmod a+rwx /.nvm
. /.nvm/nvm.sh
nvm install node
cd fileuploader
npm install
npm start
```

3. **Target Group and Load Balancer || Launch Configuration and Auto Scaling group :**

- Target group listens to the port 3000 and load balancer loads between the EC2 machines instances created via autoscaling group and ports to 80.
- The auto scaling group SCALES between (min: 1 , max : 2) machines and scale when the "number of bytes in> 10000".



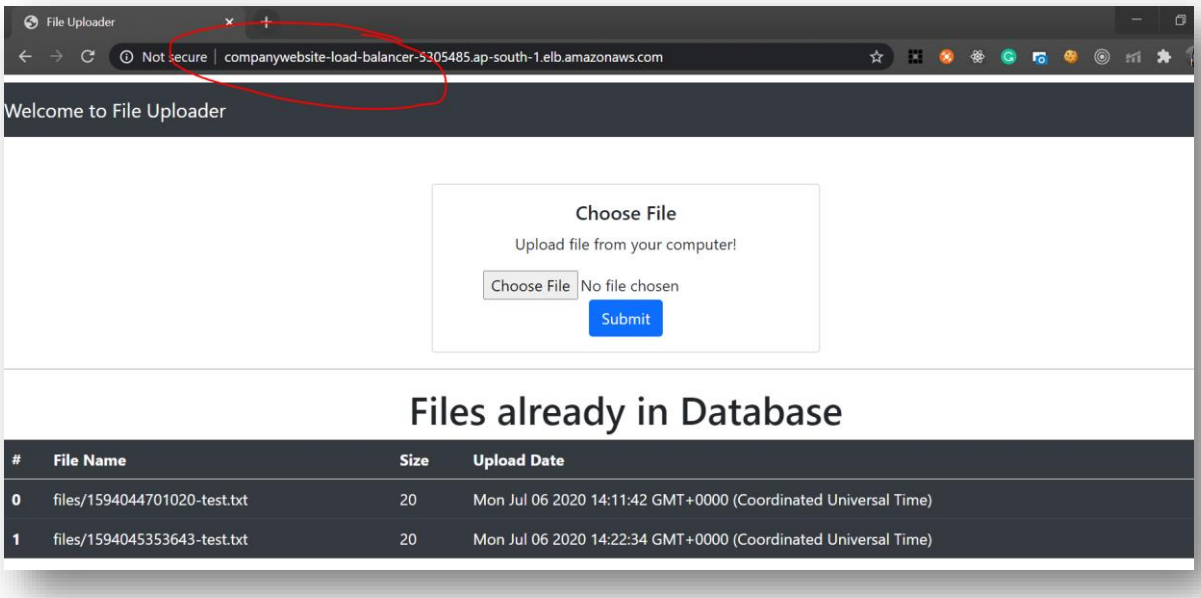
Name	Port	Protocol	Targ	Load Balancer	VPC ID
CompanyWebsiteTG	3000	HTTP	inst…	CompanyWebsite-Load-Balancer	vpc-094f864a85fe52c9f



Create Auto Scaling group | Actions

Filter: Filter Auto Scaling groups... | 1 to 1 of 1 Auto Scaling Groups

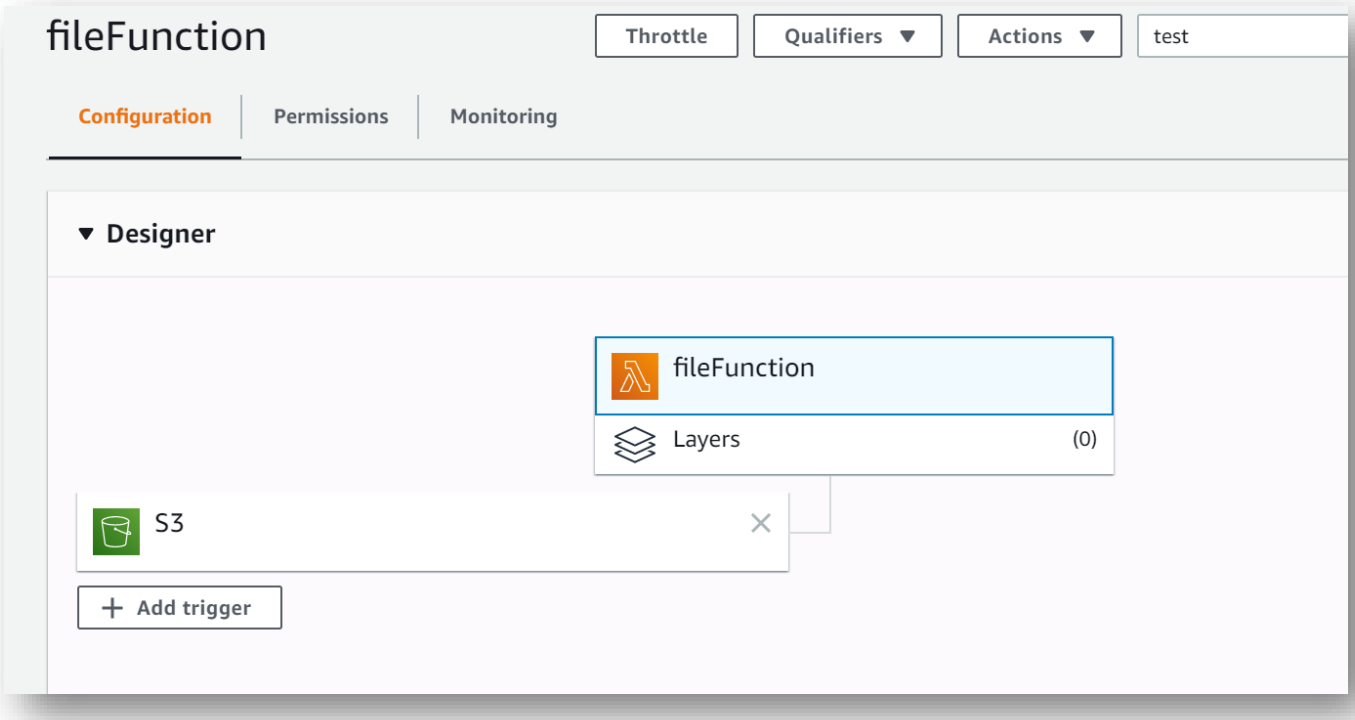
Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	He
CompanyWebsite	CompanyWebsite-LC	2	2	1	2	ap-south-1b, ap-south-1a	300	30



Code in : https://github.com/AnkitaKhurana/fileuploader.git

4. Lambda Function : fileFunction

- Any upload of file in the bucket (ankita-nagp-bucket)Triggers Lambda function (fileFunction)
- Responsible for adding file size and entry of the uploaded file in the private RDS

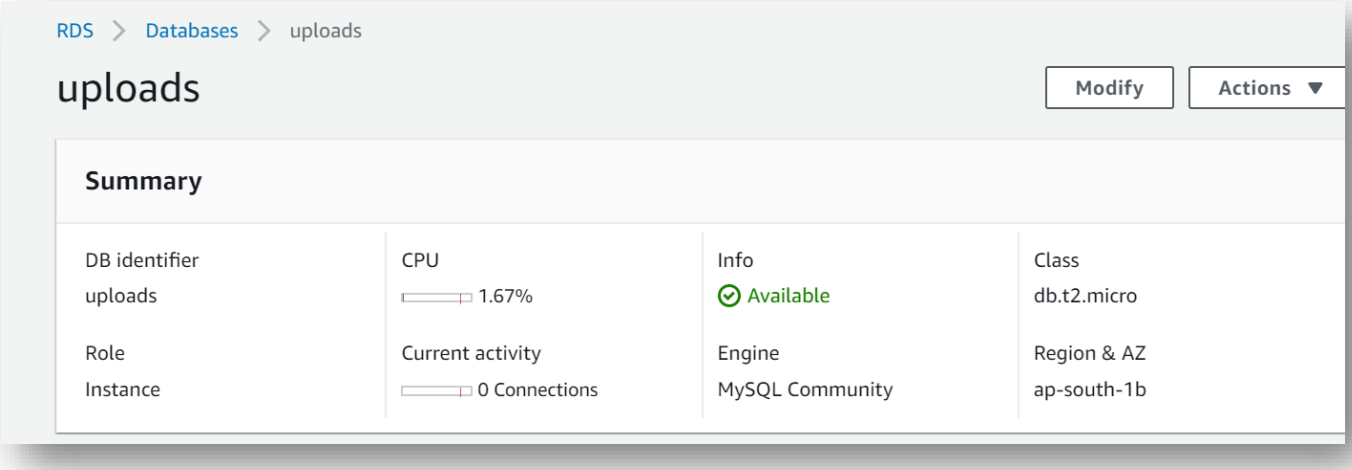


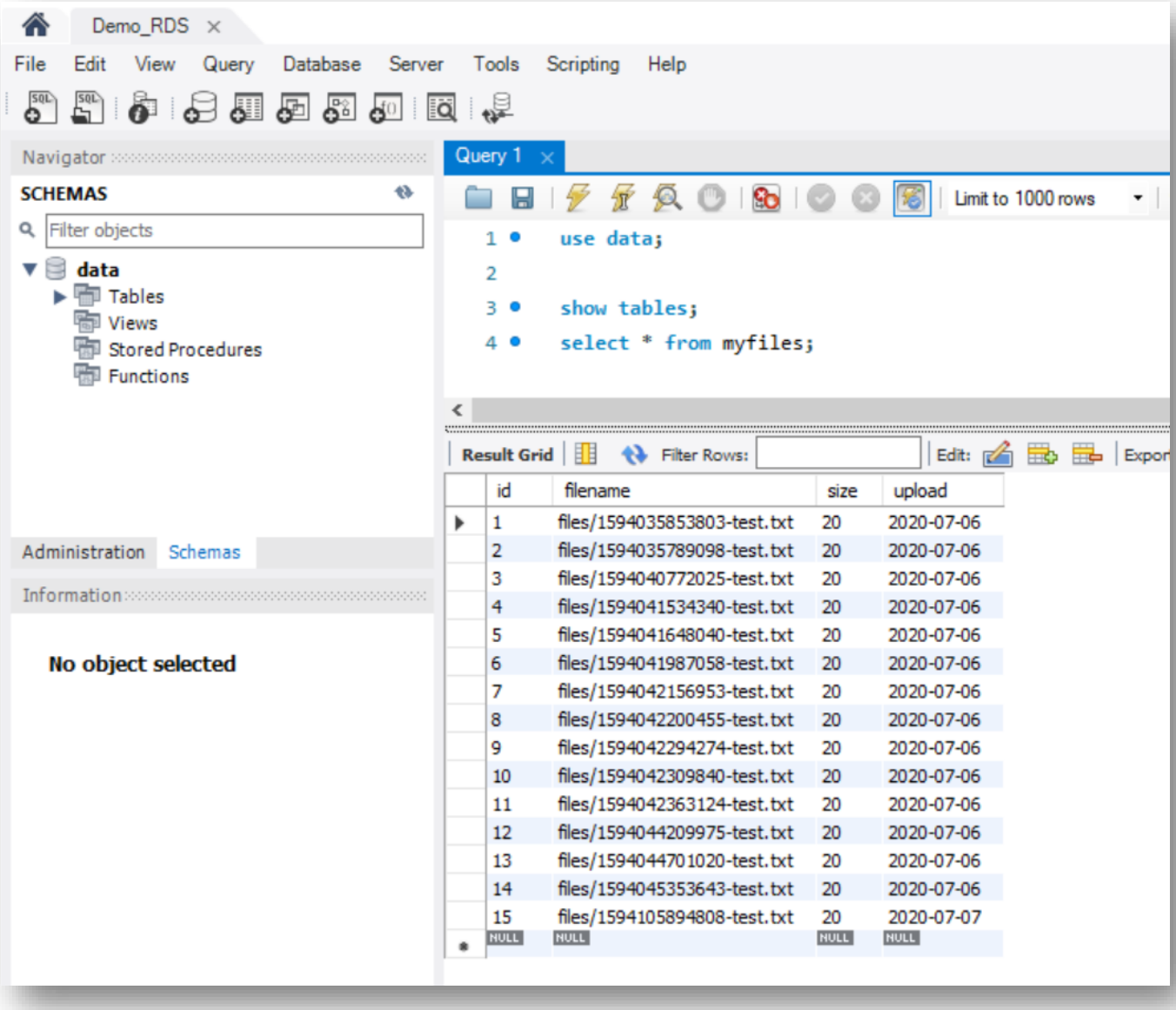
Code in Lambda function:

```
var AWSXRay = require('aws-xray-sdk-core')
var captureMySQL = require('aws-xray-sdk-mysql')
var mysql = captureMySQL(require('mysql2'))
const username = process.env.databaseUser
const password = process.env.databasePassword
const host = process.env.databaseHost
const AWS = require('aws-sdk')
var s3 = new AWS.S3();

exports.handler = async(event, context) => {
	var bucket = event.Records[0].s3.bucket.name;
	var key = event.Records[0].s3.object.key;
	var size = event.Records[0].s3.object.size;
	var connection = mysql.createConnection({
		host: host,
		user: username,
		password: password,
		database: 'data'
	});
	connection.connect();
	let date = new Date().toISOString().slice(0, 19).replace('T', ' ');
	var result;

	let query = "INSERT INTO myfiles(filename,size, upload) values('" + key +"','"+size+ ",'" + date + "');";
	connection.query(query, function (error, results, fields) {
		if (error) throw error;
			console.log("Ran query: " + query);
			for (result in results)
				console.log(results[result])
	});

	return new Promise((resolve, reject) => {
		connection.end(err => {
			if (err)
				return reject(err)

			const response = {
				statusCode: 200,
				body: "Saved to RDS"
			}
			resolve(response)
		})
	});

}
```

5. **RDS – MySQL : uploads** :

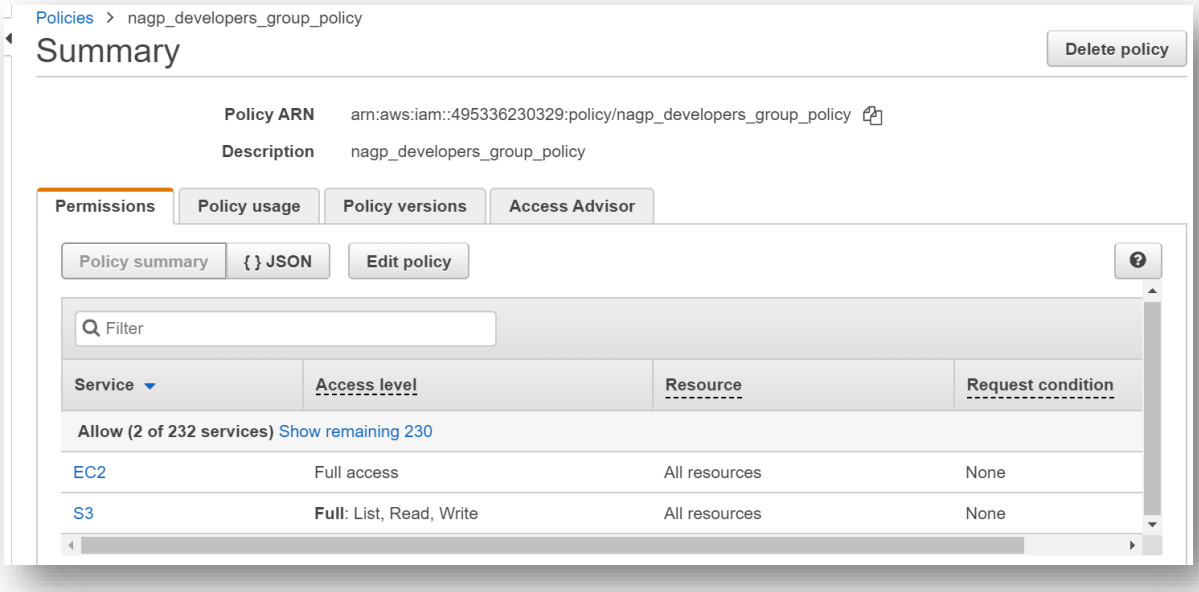
- Contains company sensitive data (here which text file was uploaded in the S3 Bucket and the size)



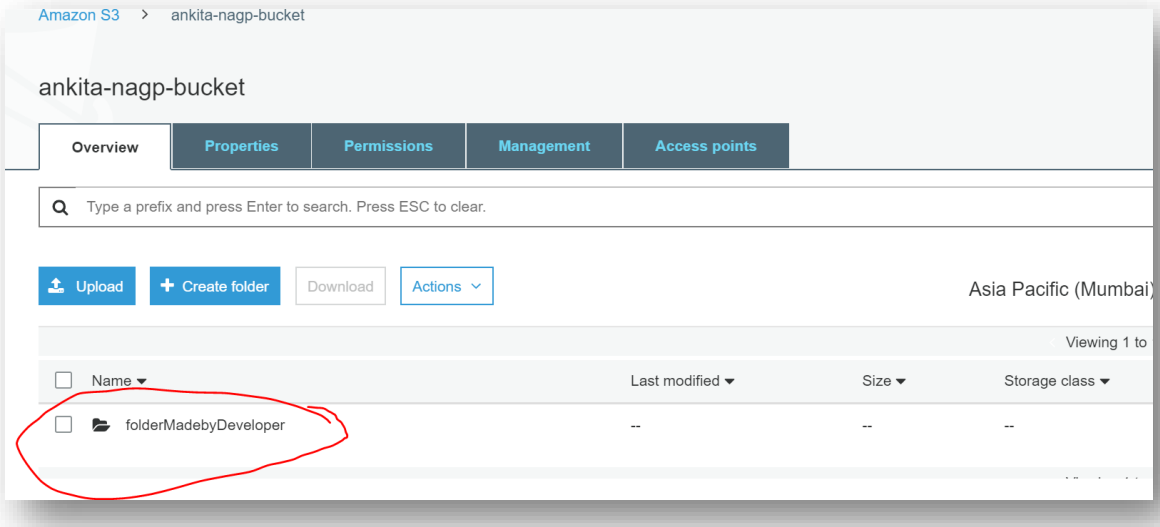


POLICIES

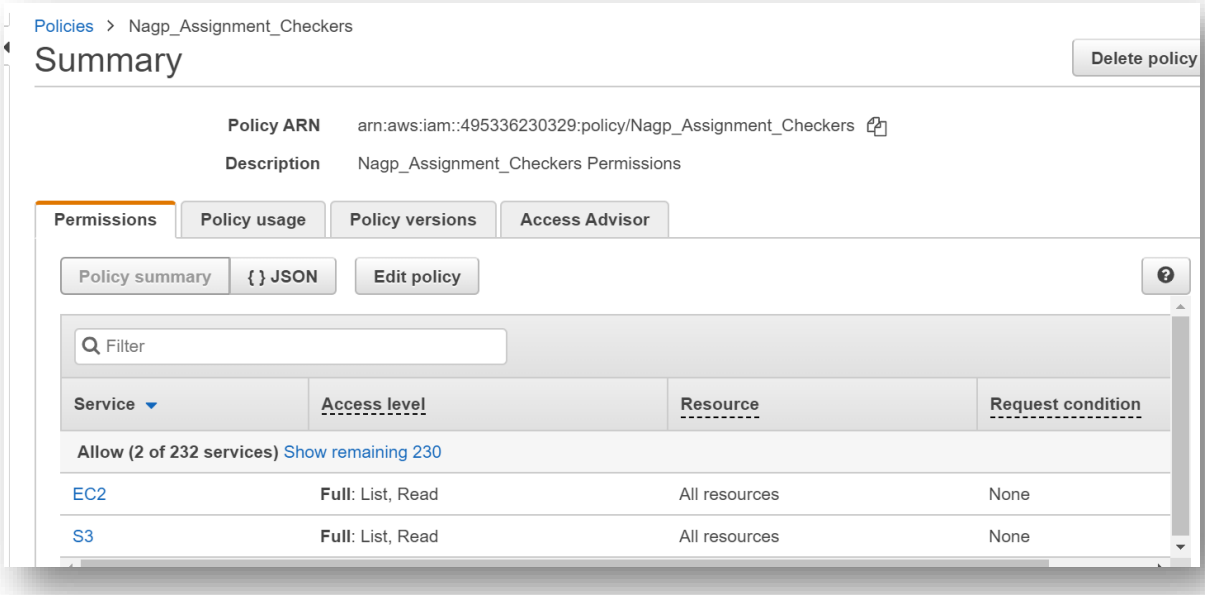
1. Policy for DEVELOPERS (R/W on S3 Buckets + Full access (Create etc) on EC2)



Example: On logging from a user in developer group (Writing in Bucket)

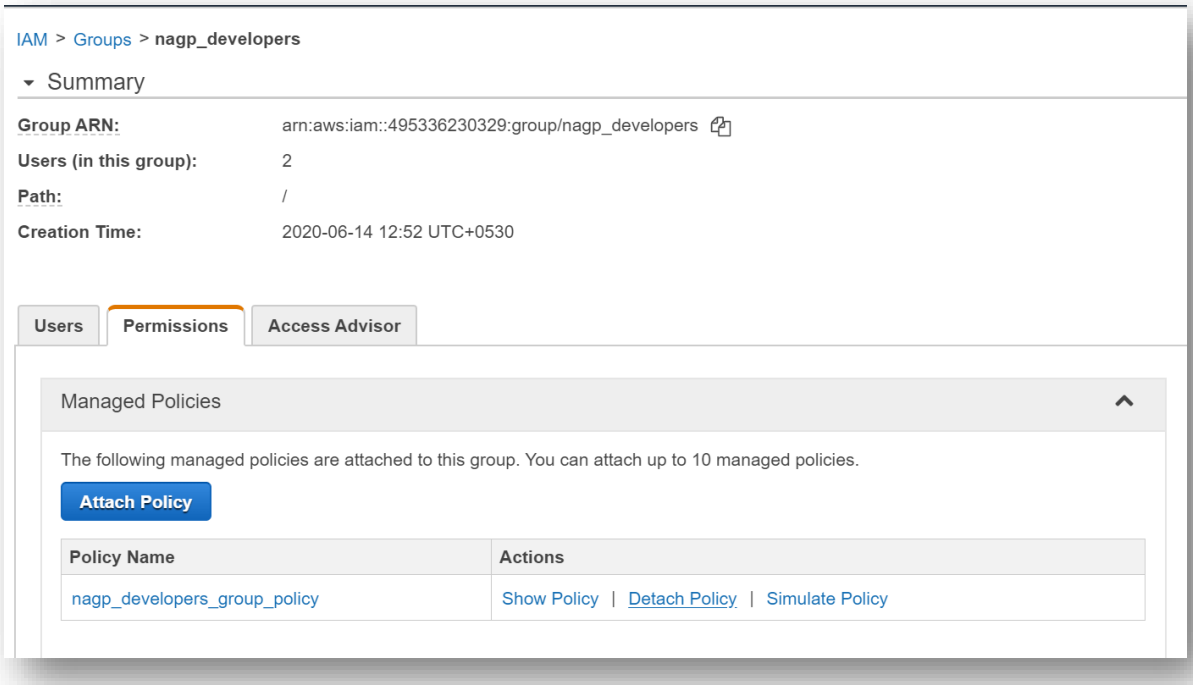


2. Policy for roles/users who wants to have min permissions (Assignment checkers)



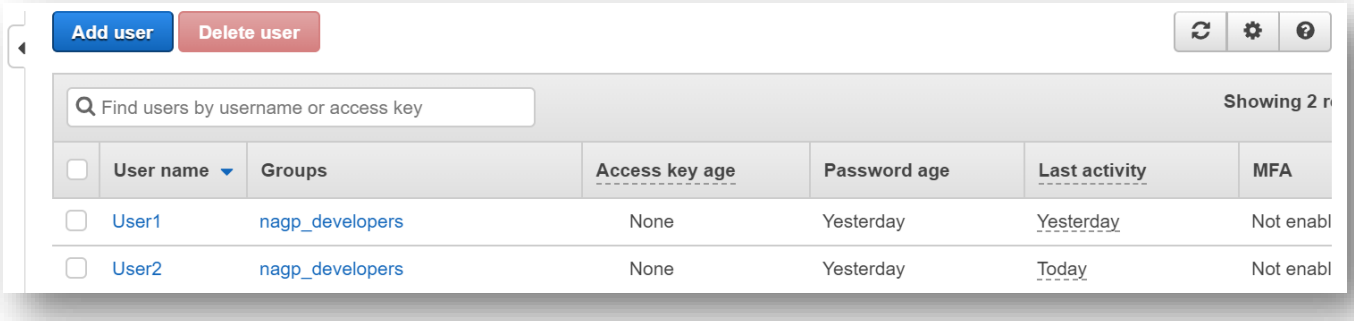
GROUPS

1.Developer group (For giving access to Services for development, two users made under this group)

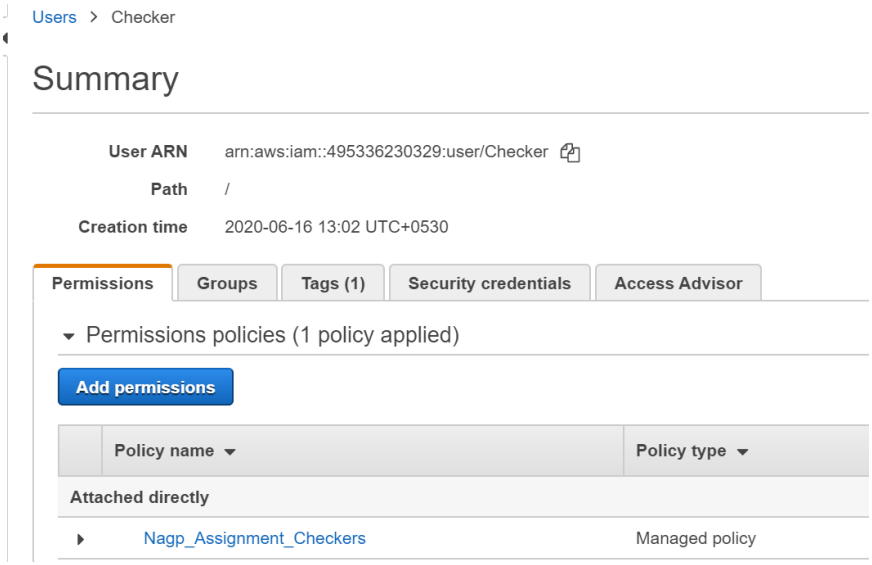


USERS

1. Developers user accounts made for the development team

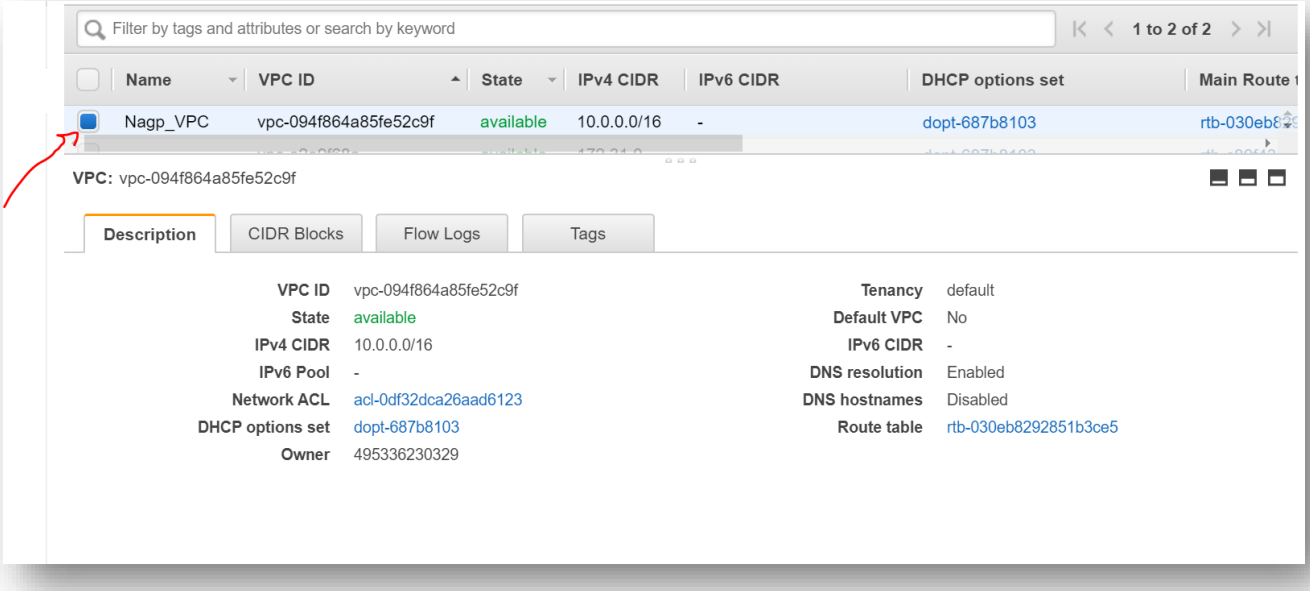


1. Single checker user account made for temp access

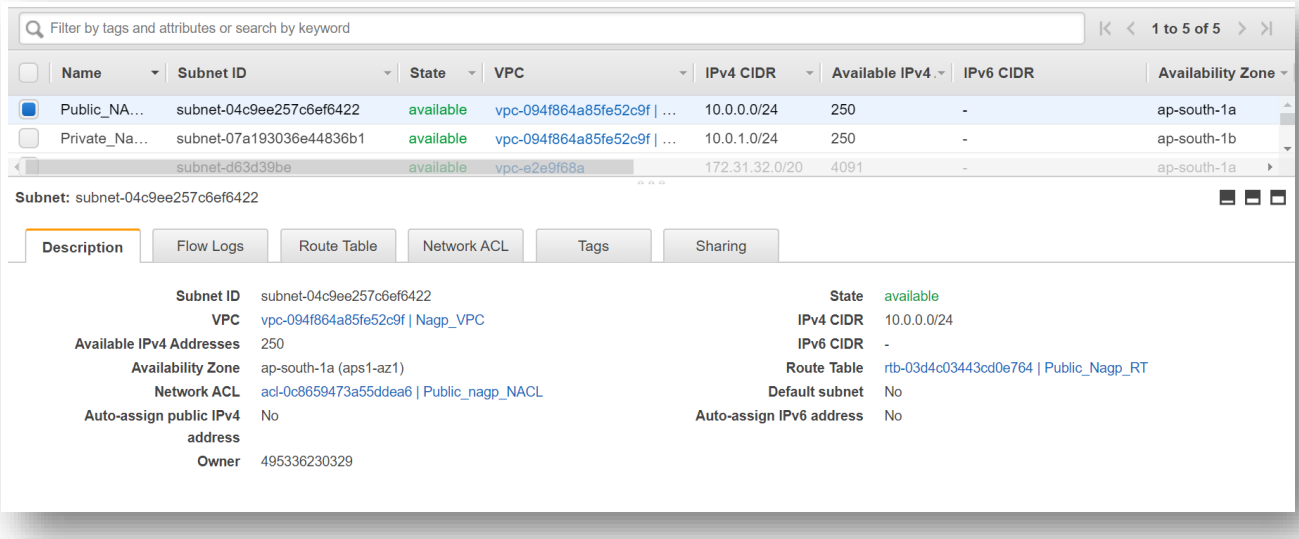


VPC COMPONENTS

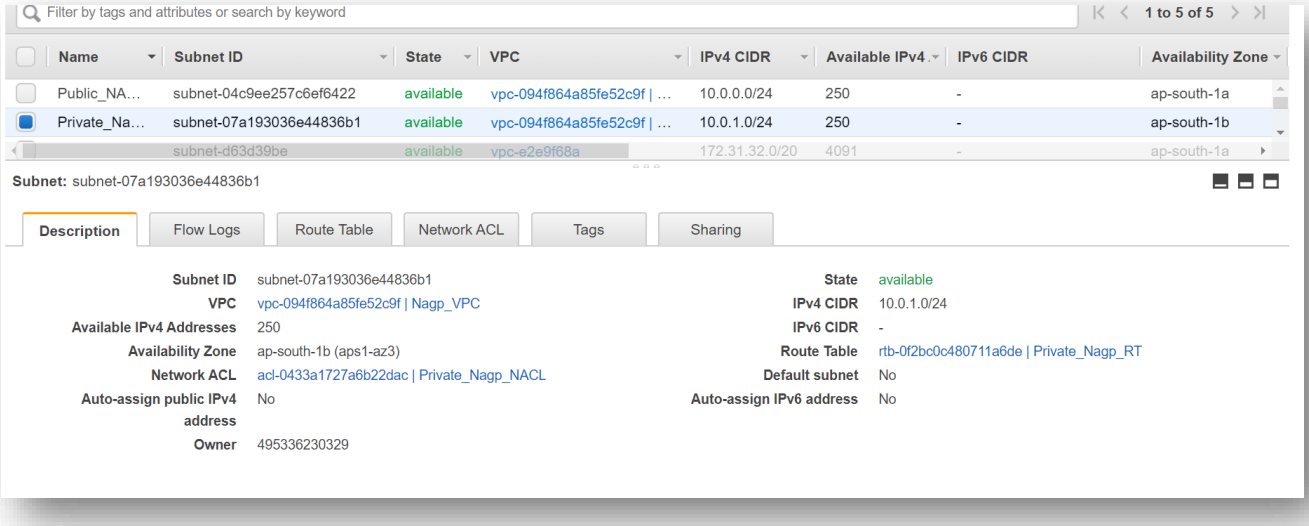
1. Nagp_VPC : 10.0.0.0/16



2. Public_Nagp_Subnet : 10.0.0.0/24



3. Private_Nagp_Subnet: 10.0.1.0/24



4. Public_Nagp_Subnet_1 : 10.0.4.0/24

Name	Subnet ID	State	VPC
Public_NAGP_Subnet_1	subnet-0978ce23778b118ca	available	vpc-094f864a85fe52c9f | …
Private_NAGP_Subnet_1	subnet-0ea9a5554289cbfb4	available	vpc-094f864a85fe52c9f | …

Subnet: subnet-0978ce23778b118ca

Description | Flow Logs | Route Table | Network ACL | Tags | Sharing

Subnet ID	subnet-0978ce23778b118ca	**State**	available
VPC	vpc-094f864a85fe52c9f | Nagp_VPC	**IPv4 CIDR**	10.0.4.0/24
Available IPv4 Addresses	248	**IPv6 CIDR**	-
Availability Zone	ap-south-1b (aps1-az3)	**Route Table**	rtb-03d4c03443cd0e764 | Public_Nagp_RT
Network ACL	acl-0c8659473a55ddea6 | Public_nagp_NACL	**Default subnet**	No
Auto-assign public IPv4 address	No	**Auto-assign IPv6 address**	No
Outpost ID	-	**Owner**	495336230329

5. Private_Nagp_Subnet_1 : 10.0.5.0/24

Private_NAGP_Subnet_1	subnet-0ea9a5554289cbfb4	available	vpc-094f864a85fe52c9f | …	10.0.5.0/24
	subnet-a62055ea	available	vpc-e2e9f68a	172.31.0.0/2

Subnet: subnet-0ea9a5554289cbfb4

Description | Flow Logs | Route Table | Network ACL | Tags | Sharing

Subnet ID	subnet-0ea9a5554289cbfb4	**State**	available
VPC	vpc-094f864a85fe52c9f | Nagp_VPC	**IPv4 CIDR**	10.0.5.0/24
Available IPv4 Addresses	250	**IPv6 CIDR**	-
Availability Zone	ap-south-1a (aps1-az1)	**Route Table**	rtb-0f2bc0c480711a6de | Private_Nagp_RT
Network ACL	acl-0433a1727a6b22dac | Private_Nagp_NACL	**Default subnet**	No
Auto-assign public IPv4 address	No	**Auto-assign IPv6 address**	No
Outpost ID	-	**Owner**	495336230329

6. Public_nagp_RT (Route table)

Filter by tags and attributes or search by keyword — 1 to 4 of 4

Name	Route Table ID	Explicit subnet association	Edge associations	Main
Public_Nagp_RT	rtb-03d4c03443cd0e764	subnet-04c9ee257c6ef6422	-	No
Private_Nagp_RT	rtb-0f2bc0c480711a6de	subnet-07a193036e44836b1	-	No

Route Table: rtb-03d4c03443cd0e764

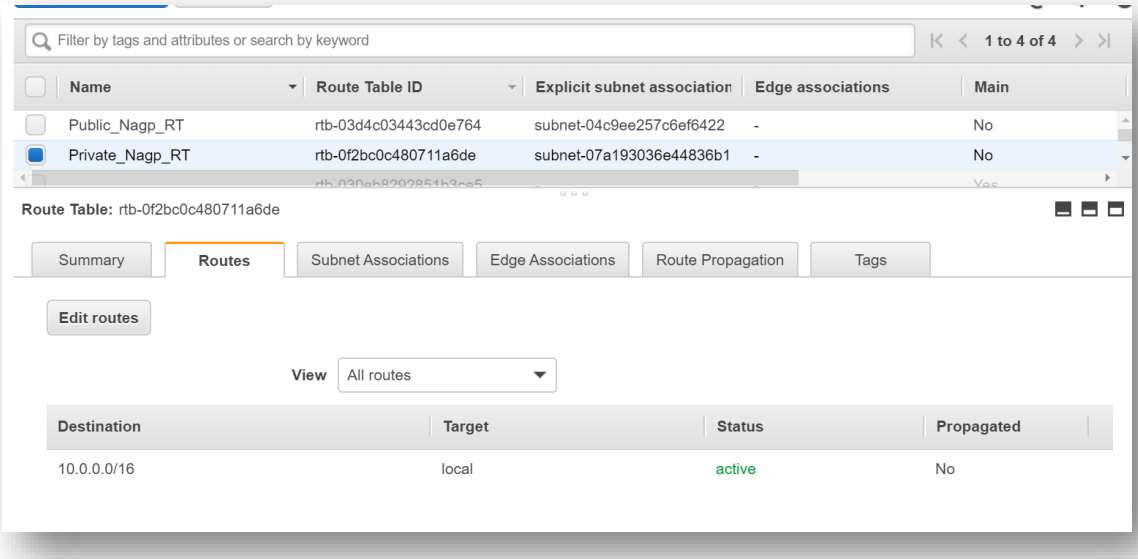
Summary | Routes | Subnet Associations | Edge Associations | Route Propagation | Tags

Edit routes

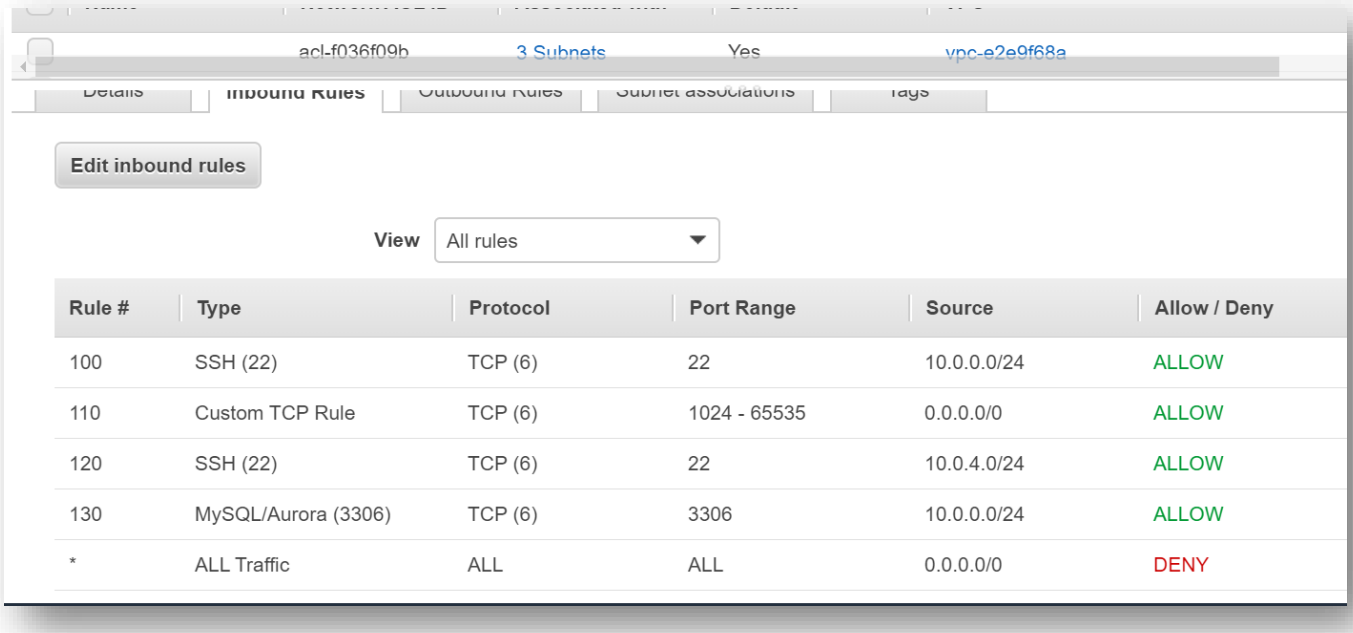
View: All routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	igw-0432df096eae556b2	active	No

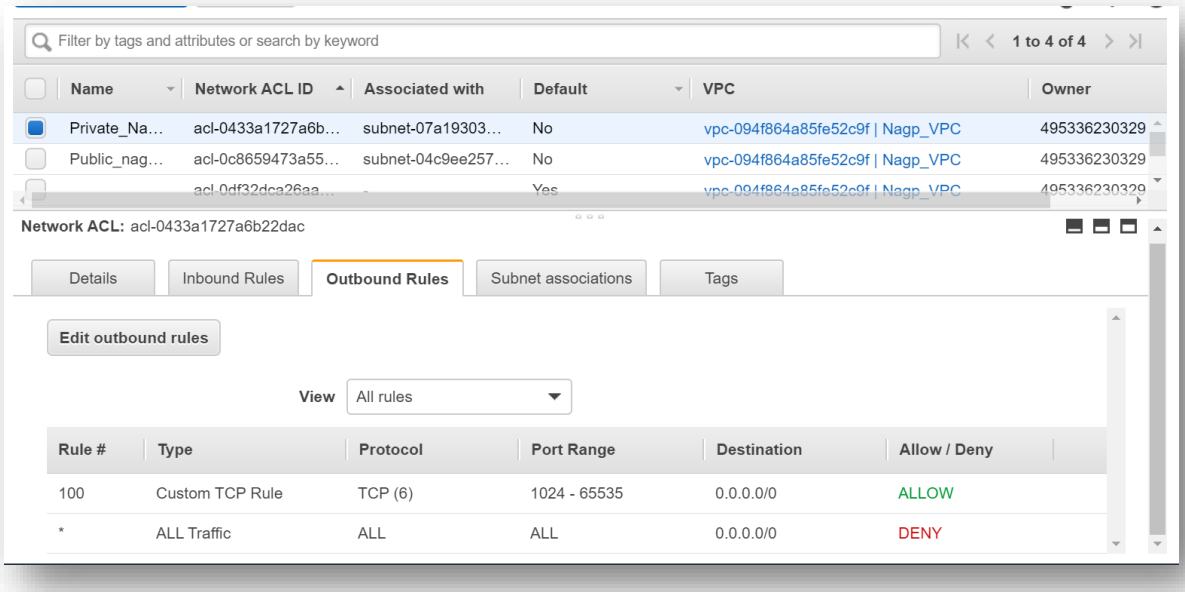
7. Private_nagp_RT (Route table)



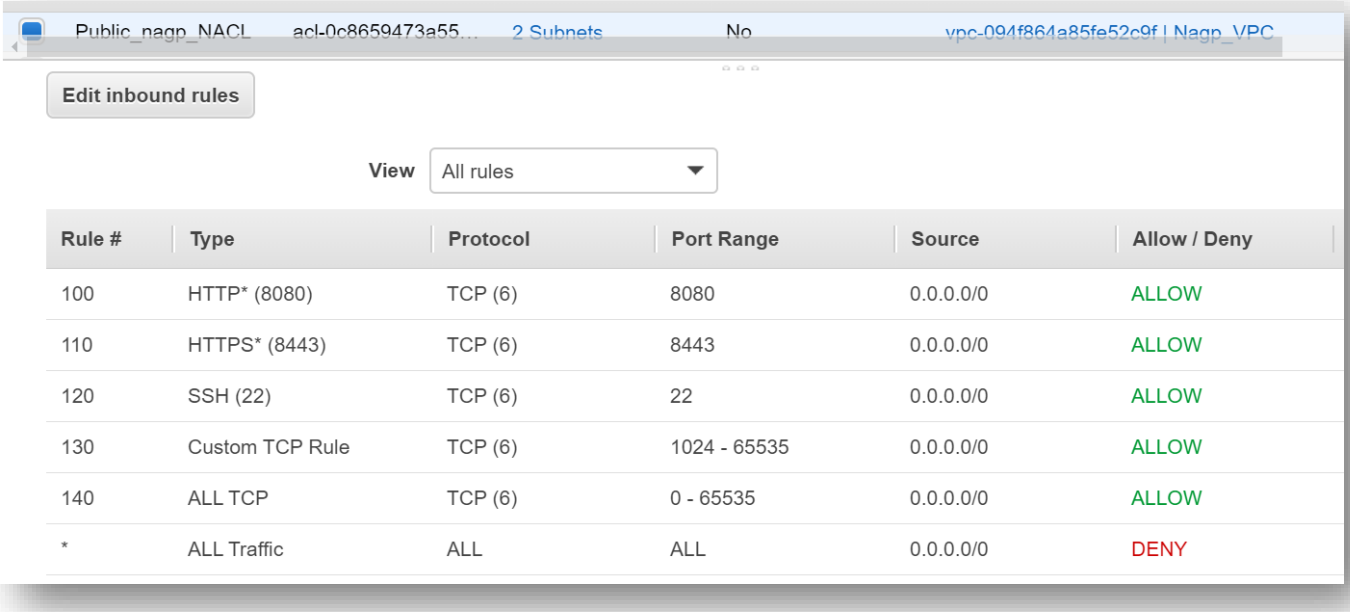
8. Private_nagp_NACL : inbound rules



9. Private_nagp_NACL : outbound rules

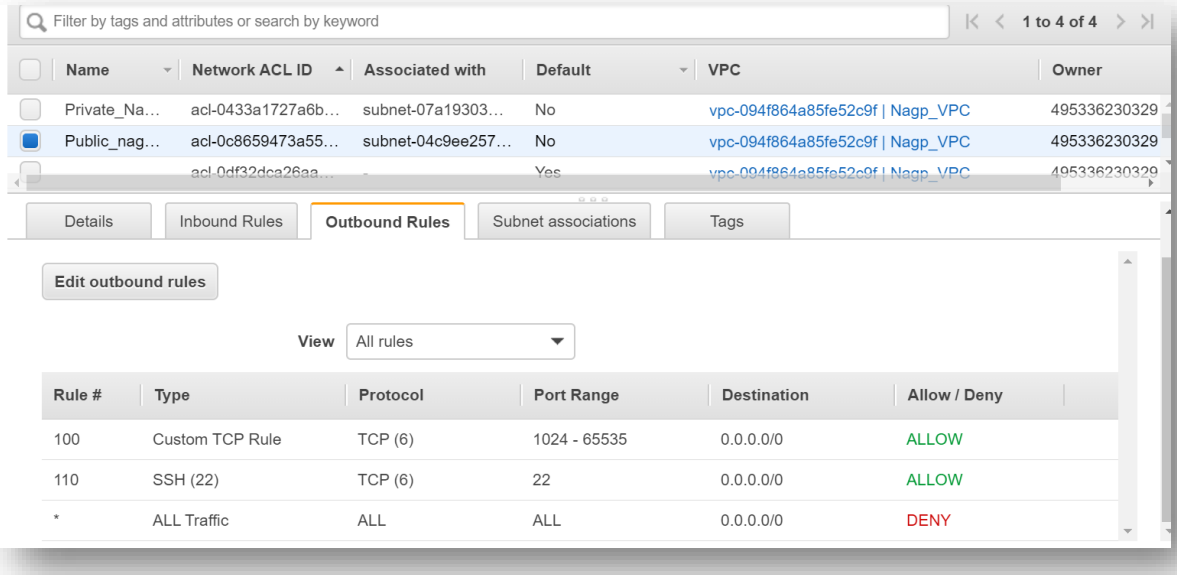


10. Public_nagp_NACL : inbound rules



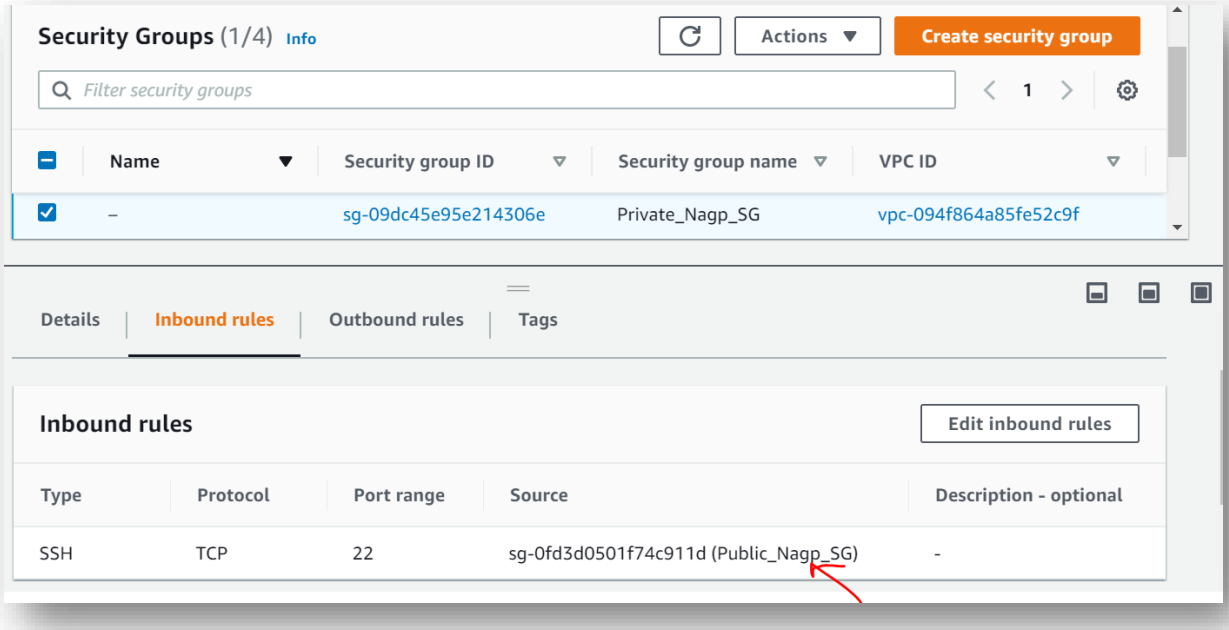
Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	HTTP* (8080)	TCP (6)	8080	0.0.0.0/0	ALLOW
110	HTTPS* (8443)	TCP (6)	8443	0.0.0.0/0	ALLOW
120	SSH (22)	TCP (6)	22	0.0.0.0/0	ALLOW
130	Custom TCP Rule	TCP (6)	1024 - 65535	0.0.0.0/0	ALLOW
140	ALL TCP	TCP (6)	0 - 65535	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

11. Public_nagp_NACL : outbound rules



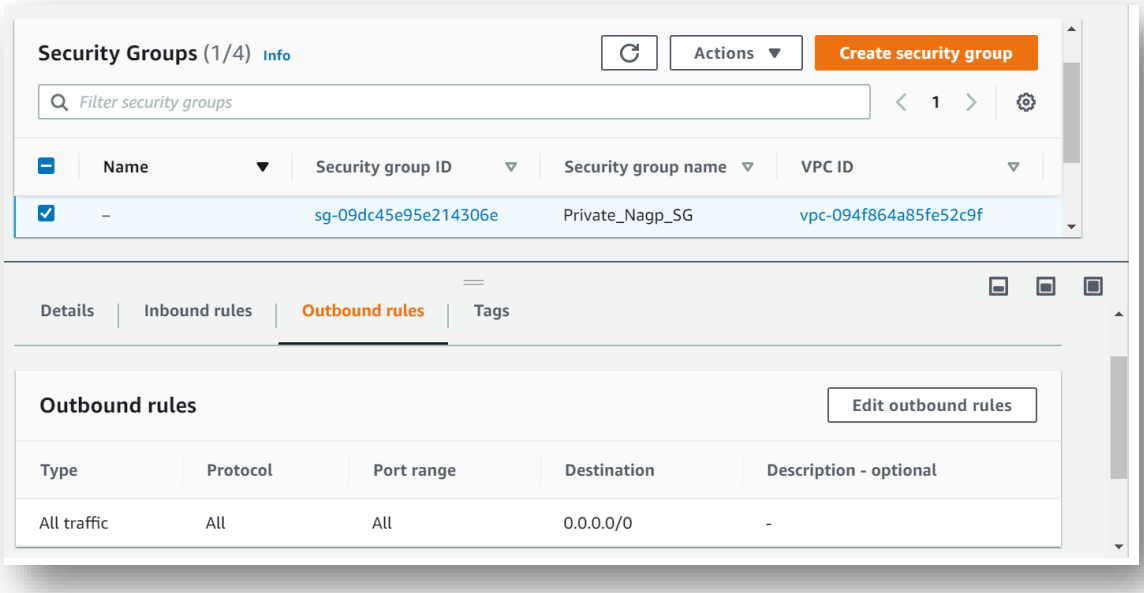
Rule #	Type	Protocol	Port Range	Destination	Allow / Deny
100	Custom TCP Rule	TCP (6)	1024 - 65535	0.0.0.0/0	ALLOW
110	SSH (22)	TCP (6)	22	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

12. Private_Nagp_SG (Security Group) : inbound rules

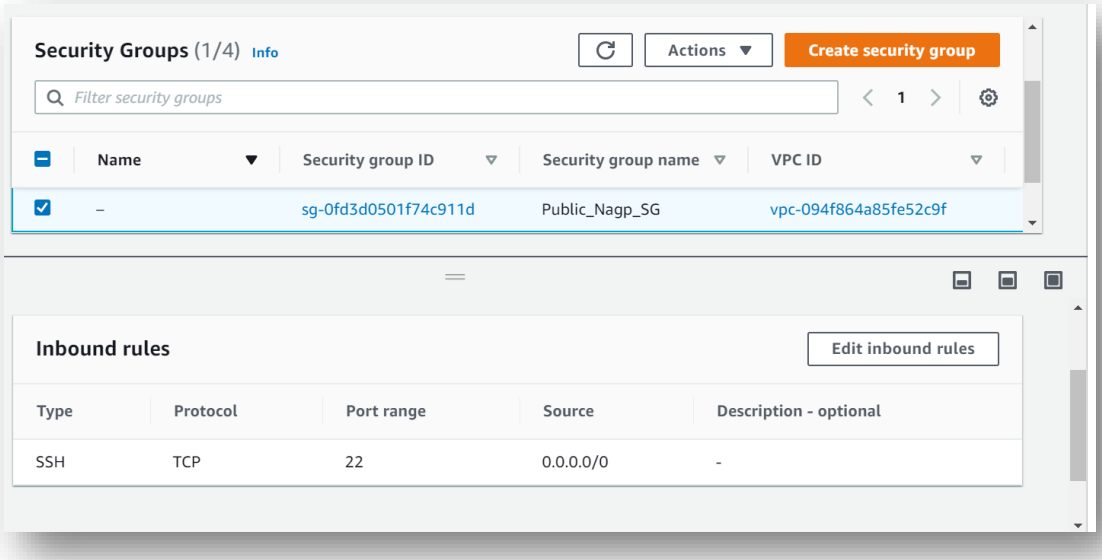


Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	sg-0fd3d0501f74c911d (Public_Nagp_SG)	-

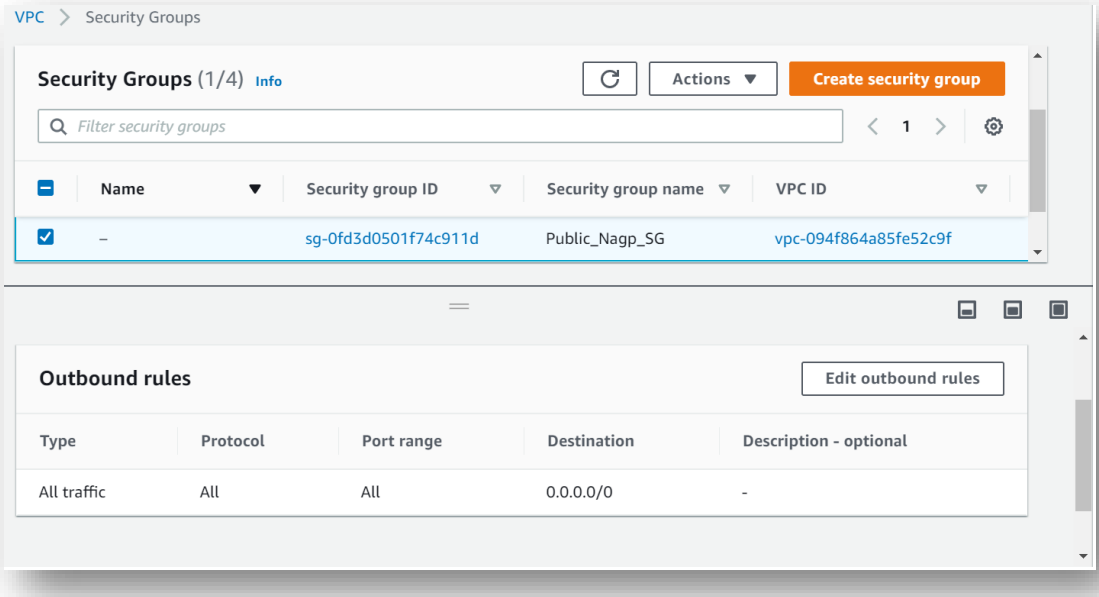
13. Private_Nagp_SG (Security Group) : outbound rules



14. Public_Nagp_SG (Security Group) : inbound rules



15. Public_Nagp_SG (Security Group) : outbound rules



16. MySQL Security Group

Inbound rules | Outbound rules | Tags

Inbound rules Edit inbo

Type	Protocol	Port range	Source	Descripti
All traffic	All	All	sg-02efa84dc66668392 (default)	-
MYSQL/Aurora	TCP	3306	sg-0c35a11c46687dc69 (LambdaSecurityGroup)	-
MYSQL/Aurora	TCP	3306	sg-0fd3d0501f74c911d (Public_Nagp_SG)	-

17. Auto Scaling EC2 Security Group

Inbound rules | Outbound rules | Tags

Inbound rules Edit inbound rules

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	0.0.0.0/0	-
SSH	TCP	22	0.0.0.0/0	-
Custom TCP	TCP	3000	0.0.0.0/0	-
HTTPS	TCP	443	0.0.0.0/0	-

18. Internet Gateway

VPC > Internet gateways

Internet gateways (1/2) Info

Actions | Create internet gateway

Filter internet gateways

1

Name	Internet gateway ID	State	VPC ID
NAGP_Internet_Gat...	igw-0432df096eae556b2	Attached	vpc-094f864a85fe52c9f | Na

Key	Value
Name	NAGP_Internet_Gateway