

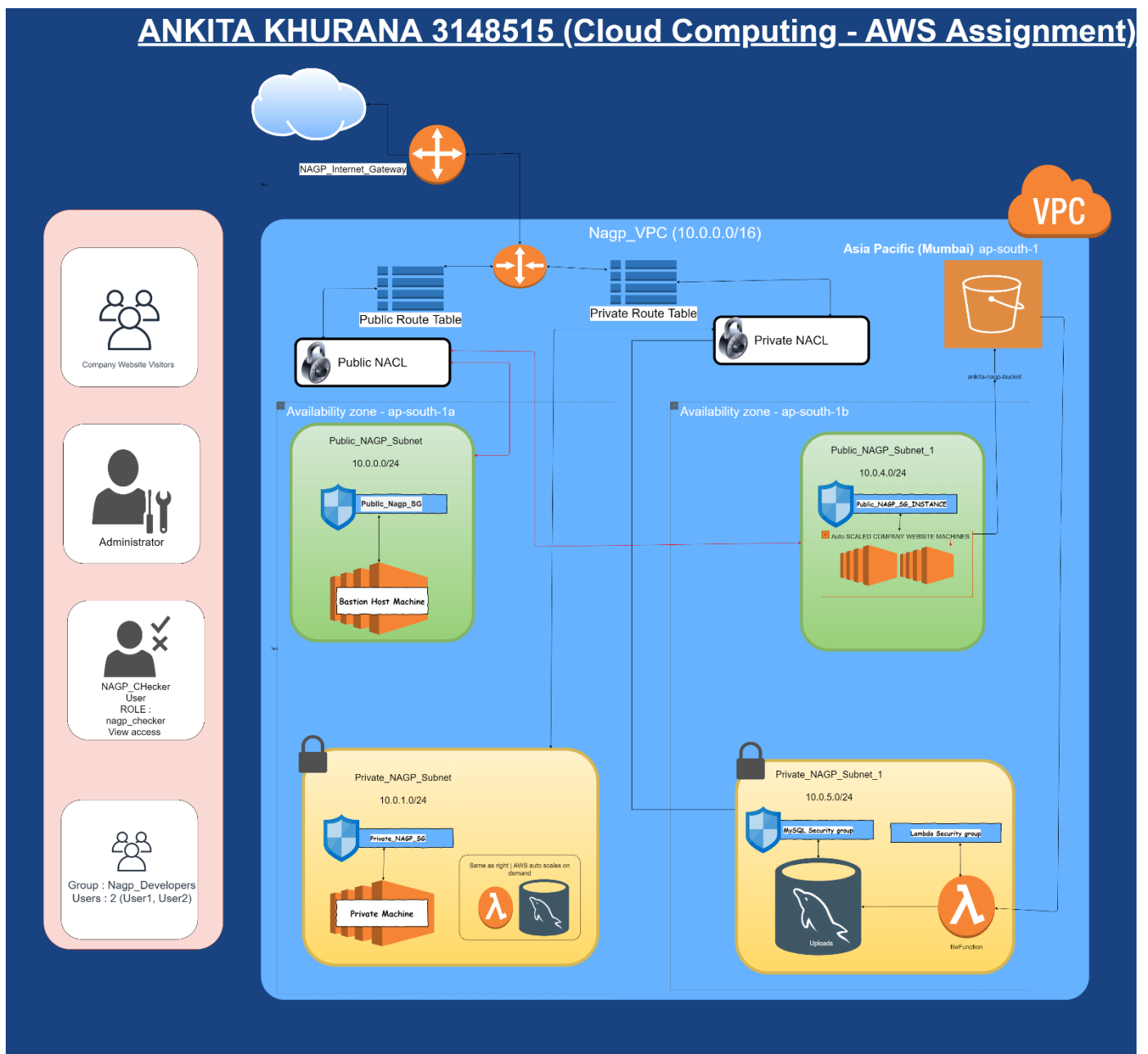
CLOUD COMPUTING ASSIGNMENT 2

- Ankita Khurana

- 3148515

Note:

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

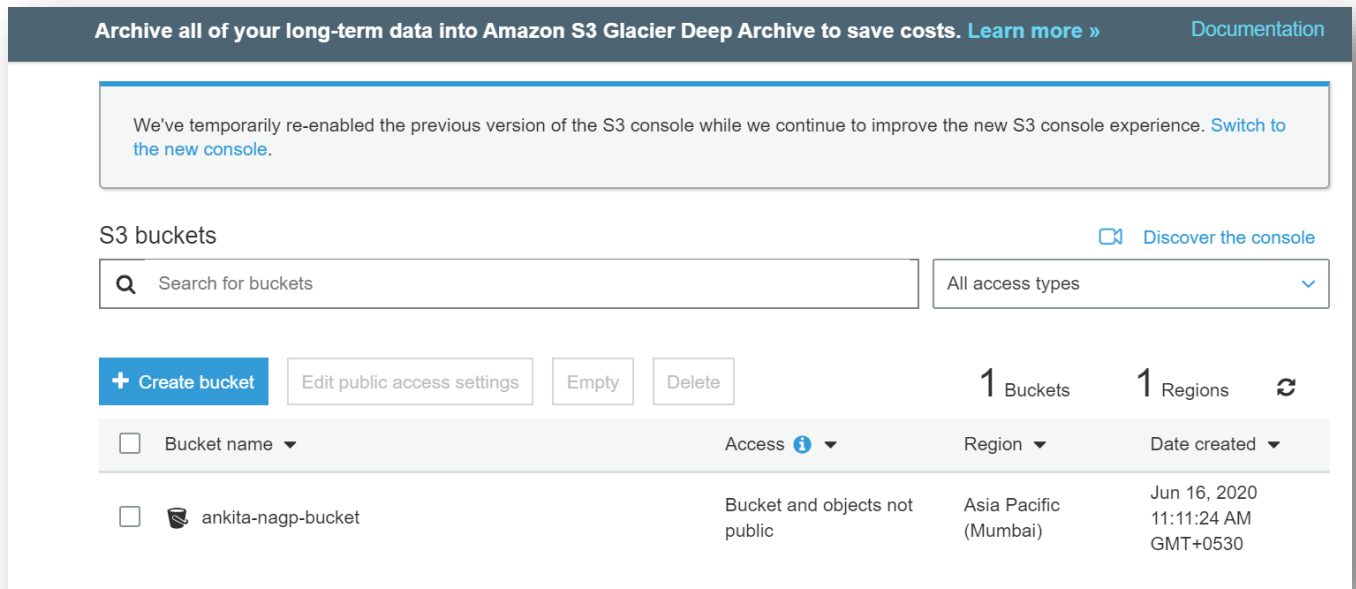


ARCHITECTURE

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

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

File Uploader

Not secure | companywebsite-load-balancer-5305485.ap-south-1.elb.amazonaws.com

Welcome to File Uploader

Choose File

Upload file from your computer!

Choose File No file chosen

Submit

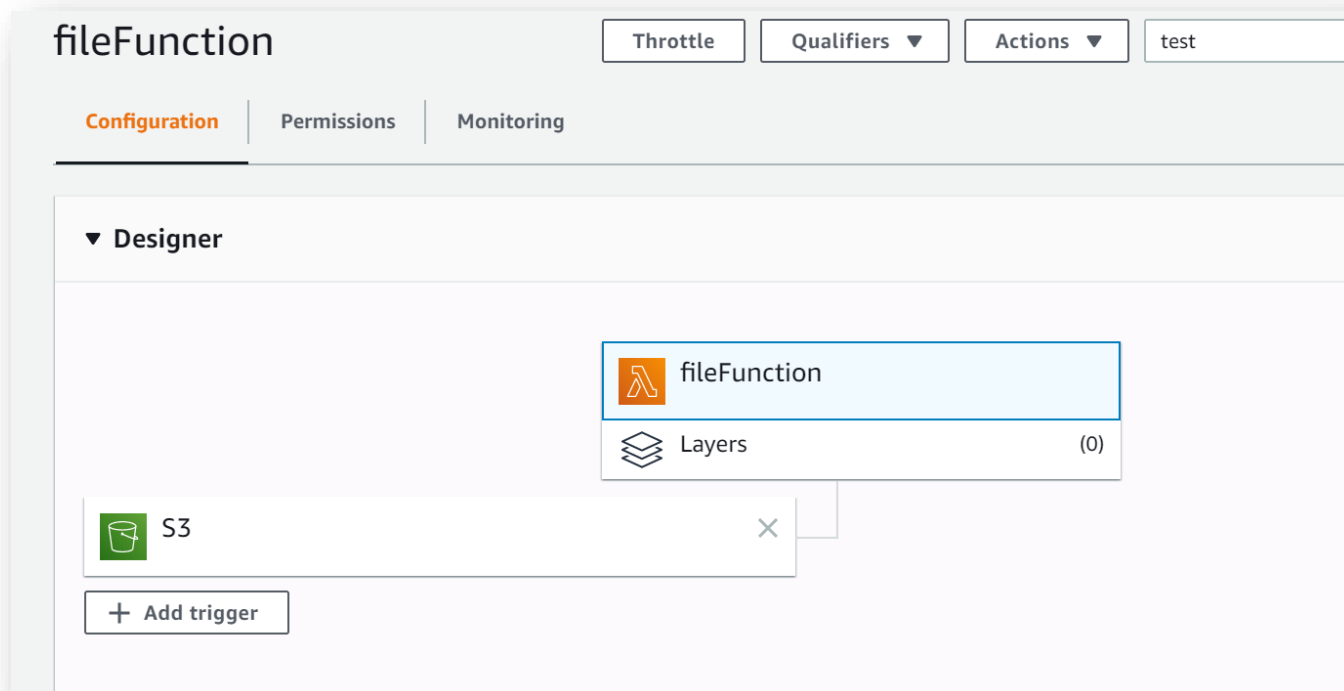
Files already in Database

#	File Name	Size	Upload Date
0	files/1594044701020-test.txt	20	Mon Jul 06 2020 14:11:42 GMT+0000 (Coordinated Universal Time)
1	files/1594045353643-test.txt	20	Mon Jul 06 2020 14:22:34 GMT+0000 (Coordinated Universal Time)

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)

RDS > Databases > uploads

uploads

Modify Actions

Summary

DB identifier uploads	CPU <div><div></div></div> 1.67%	Info Available	Class db.t2.micro
Role Instance	Current activity <div><div></div></div> 0 Connections	Engine MySQL Community	Region & AZ ap-south-1b

Demo_RDS x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

data

- Tables
- Views
- Stored Procedures
- Functions

Administration Schemas Information

No object selected

Query 1 x

Limit to 1000 rows

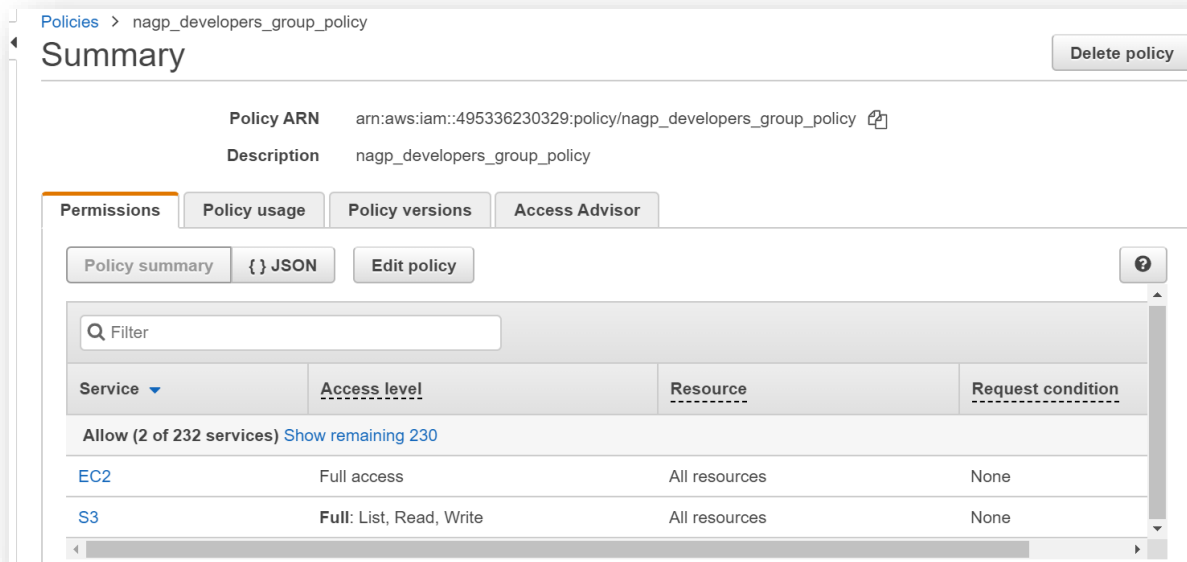
```
1 • use data;
2
3 • show tables;
4 • select * from myfiles;
```

Result Grid

	id	filename	size	upload
▶	1	files/1594035853803-test.txt	20	2020-07-06
	2	files/1594035789098-test.txt	20	2020-07-06
	3	files/1594040772025-test.txt	20	2020-07-06
	4	files/1594041534340-test.txt	20	2020-07-06
	5	files/1594041648040-test.txt	20	2020-07-06
	6	files/1594041987058-test.txt	20	2020-07-06
	7	files/1594042156953-test.txt	20	2020-07-06
	8	files/1594042200455-test.txt	20	2020-07-06
	9	files/1594042294274-test.txt	20	2020-07-06
	10	files/1594042309840-test.txt	20	2020-07-06
	11	files/1594042363124-test.txt	20	2020-07-06
	12	files/1594044209975-test.txt	20	2020-07-06
	13	files/1594044701020-test.txt	20	2020-07-06
	14	files/1594045353643-test.txt	20	2020-07-06
	15	files/1594105894808-test.txt	20	2020-07-07
*	NULL	NULL	NULL	NULL

POLICIES

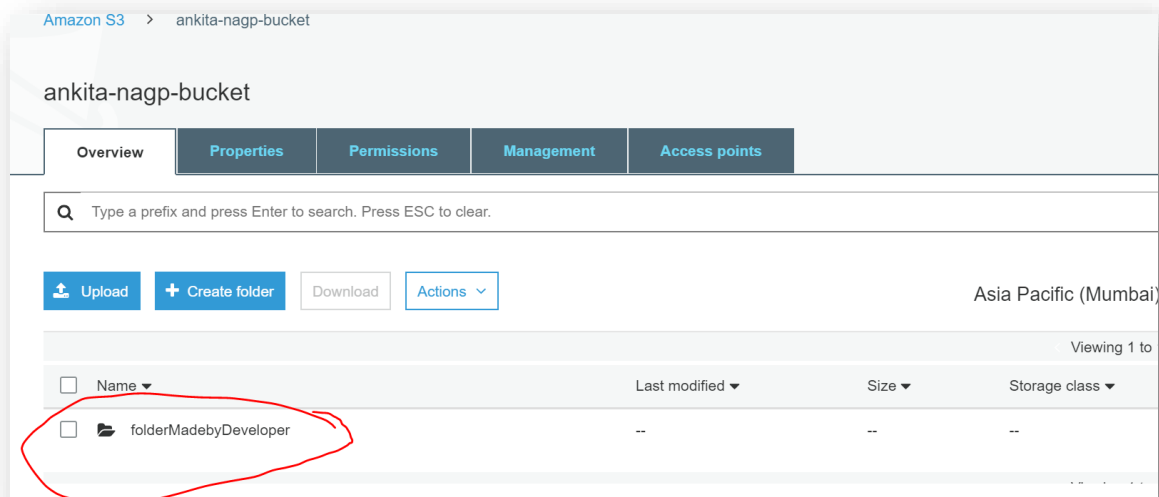
1. Policy for **DEVELOPERS** (R/W on S3 Buckets + Full access (Create etc) on EC2)



The screenshot shows the AWS IAM Policy console for the policy `nagp_developers_group_policy`. The **Summary** tab is active, displaying the Policy ARN `arn:aws:iam::495336230329:policy/nagp_developers_group_policy` and the Description `nagp_developers_group_policy`. Below the summary, there are tabs for **Permissions**, **Policy usage**, **Policy versions**, and **Access Advisor**. The **Permissions** tab is selected, showing a table of permissions. The table has columns for **Service**, **Access level**, **Resource**, and **Request condition**. It lists two permissions: **EC2** with **Full access** on **All resources**, and **S3** with **Full: List, Read, Write** on **All resources**. A red circle highlights the **S3** row.

Service	Access level	Resource	Request condition
EC2	Full access	All resources	None
S3	Full: List, Read, Write	All resources	None

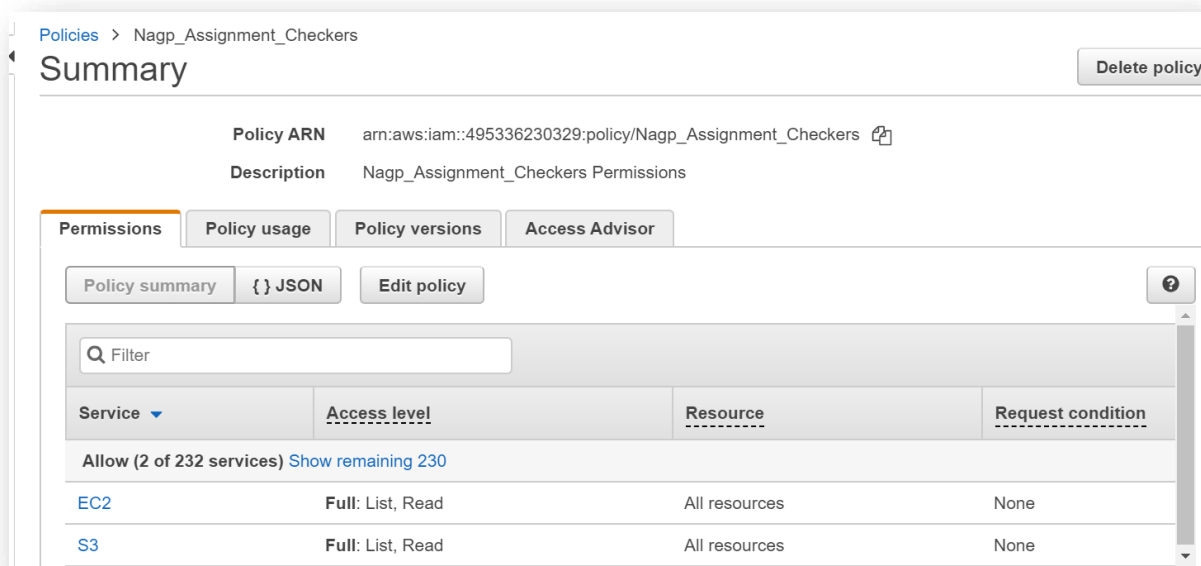
Example: On logging from a user in developer group (Writing in Bucket)



The screenshot shows the Amazon S3 console for the bucket `ankita-nagp-bucket`. The **Overview** tab is active, displaying the bucket name and the region `Asia Pacific (Mumbai)`. Below the overview, there are tabs for **Overview**, **Properties**, **Permissions**, **Management**, and **Access points**. The **Overview** tab is selected, showing a table of objects. The table has columns for **Name**, **Last modified**, **Size**, and **Storage class**. It lists one object: `folderMadebyDeveloper`. A red circle highlights the `folderMadebyDeveloper` row.

Name	Last modified	Size	Storage class
folderMadebyDeveloper	--	--	--

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



The screenshot shows the AWS IAM Policy console for the policy `Nagp_Assignment_Checkers`. The **Summary** tab is active, displaying the Policy ARN `arn:aws:iam::495336230329:policy/Nagp_Assignment_Checkers` and the Description `Nagp_Assignment_Checkers Permissions`. Below the summary, there are tabs for **Permissions**, **Policy usage**, **Policy versions**, and **Access Advisor**. The **Permissions** tab is selected, showing a table of permissions. The table has columns for **Service**, **Access level**, **Resource**, and **Request condition**. It lists two permissions: **EC2** with **Full: List, Read** on **All resources**, and **S3** with **Full: List, Read** on **All resources**. A red circle highlights the **S3** row.

Service	Access level	Resource	Request condition
EC2	Full: List, Read	All resources	None
S3	Full: List, Read	All resources	None

GROUPS

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

IAM > Groups > nagp_developers

▼ Summary

Group ARN: arn:aws:iam::495336230329:group/nagp_developers

Users (in this group): 2

Path: /

Creation Time: 2020-06-14 12:52 UTC+0530

Users | **Permissions** | Access Advisor

Managed Policies

The following managed policies are attached to this group. You can attach up to 10 managed policies.

[Attach Policy](#)

Policy Name	Actions
nagp_developers_group_policy	Show Policy Detach Policy Simulate Policy

USERS

1. Developers user accounts made for the development team

[Add user](#) [Delete user](#)

Find users by username or access key Showing 2 results

<input type="checkbox"/>	User name ▼	Groups	Access key age	Password age	Last activity	MFA
<input type="checkbox"/>	User1	nagp_developers	None	Yesterday	Yesterday	Not enabled
<input type="checkbox"/>	User2	nagp_developers	None	Yesterday	Today	Not enabled

1. Single checker user account made for temp access

Users > Checker

Summary

User ARN arn:aws:iam::495336230329:user/Checker

Path /

Creation time 2020-06-16 13:02 UTC+0530

Permissions | Groups | Tags (1) | Security credentials | Access Advisor

▼ Permissions policies (1 policy applied)

[Add permissions](#)

Policy name ▼	Policy type ▼
Attached directly	
▶ Nagp_Assignment_Checkers	Managed policy

VPC COMPONENTS

1. Nagp_VPC : 10.0.0.0/16

The screenshot shows the AWS VPC console interface. At the top, there is a search bar and a table of VPCs. The table has columns: Name, VPC ID, State, IPv4 CIDR, IPv6 CIDR, DHCP options set, and Main Route table. The VPC 'Nagp_VPC' with ID 'vpc-094f864a85fe52c9f' is selected and highlighted in blue. A red arrow points to the selection checkbox. Below the table, the details for 'Nagp_VPC' are displayed. The details are organized into two columns. The left column includes: VPC ID (vpc-094f864a85fe52c9f), State (available), IPv4 CIDR (10.0.0.0/16), IPv6 Pool (-), Network ACL (acl-0df32dca26aad6123), DHCP options set (dopt-687b8103), and Owner (495336230329). The right column includes: Tenancy (default), Default VPC (No), IPv6 CIDR (-), DNS resolution (Enabled), DNS hostnames (Disabled), and Route table (rtb-030eb8292851b3ce5).

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Main Route table
Nagp_VPC	vpc-094f864a85fe52c9f	available	10.0.0.0/16	-	dopt-687b8103	rtb-030eb8292851b3ce5

VPC: vpc-094f864a85fe52c9f

Property	Value
VPC ID	vpc-094f864a85fe52c9f
State	available
IPv4 CIDR	10.0.0.0/16
IPv6 Pool	-
Network ACL	acl-0df32dca26aad6123
DHCP options set	dopt-687b8103
Owner	495336230329
Tenancy	default
Default VPC	No
IPv6 CIDR	-
DNS resolution	Enabled
DNS hostnames	Disabled
Route table	rtb-030eb8292851b3ce5

2. Public_Nagp_Subnet : 10.0.0.0/24

The screenshot shows the AWS VPC console interface. At the top, there is a search bar and a table of subnets. The table has columns: Name, Subnet ID, State, VPC, IPv4 CIDR, Available IPv4, IPv6 CIDR, and Availability Zone. The subnet 'Public_Nagp_Subnet' with ID 'subnet-04c9ee257c6ef6422' is selected and highlighted in blue. Below the table, the details for 'Public_Nagp_Subnet' are displayed. The details are organized into two columns. The left column includes: Subnet ID (subnet-04c9ee257c6ef6422), VPC (vpc-094f864a85fe52c9f | Nagp_VPC), Available IPv4 Addresses (250), Availability Zone (ap-south-1a (aps1-az1)), Network ACL (acl-0c8659473a55ddea6 | Public_nagp_NACL), Auto-assign public IPv4 address (No), and Owner (495336230329). The right column includes: State (available), IPv4 CIDR (10.0.0.0/24), IPv6 CIDR (-), Route Table (rtb-03d4c03443cd0e764 | Public_Nagp_RT), Default subnet (No), and Auto-assign IPv6 address (No).

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone
Public_Nagp_Subnet	subnet-04c9ee257c6ef6422	available	vpc-094f864a85fe52c9f ...	10.0.0.0/24	250	-	ap-south-1a
Private_Nagp_Subnet	subnet-07a193036e44836b1	available	vpc-094f864a85fe52c9f ...	10.0.1.0/24	250	-	ap-south-1b

Subnet: subnet-04c9ee257c6ef6422

Property	Value
Subnet ID	subnet-04c9ee257c6ef6422
VPC	vpc-094f864a85fe52c9f Nagp_VPC
Available IPv4 Addresses	250
Availability Zone	ap-south-1a (aps1-az1)
Network ACL	acl-0c8659473a55ddea6 Public_nagp_NACL
Auto-assign public IPv4 address	No
Owner	495336230329
State	available
IPv4 CIDR	10.0.0.0/24
IPv6 CIDR	-
Route Table	rtb-03d4c03443cd0e764 Public_Nagp_RT
Default subnet	No
Auto-assign IPv6 address	No

3. Private_Nagp_Subnet: 10.0.1.0/24

The screenshot shows the AWS VPC console interface. At the top, there is a search bar and a table of subnets. The table has columns: Name, Subnet ID, State, VPC, IPv4 CIDR, Available IPv4, IPv6 CIDR, and Availability Zone. The subnet 'Private_Nagp_Subnet' with ID 'subnet-07a193036e44836b1' is selected and highlighted in blue. Below the table, the details for 'Private_Nagp_Subnet' are displayed. The details are organized into two columns. The left column includes: Subnet ID (subnet-07a193036e44836b1), VPC (vpc-094f864a85fe52c9f | Nagp_VPC), Available IPv4 Addresses (250), Availability Zone (ap-south-1b (aps1-az3)), Network ACL (acl-0433a1727a6b22dac | Private_Nagp_NACL), Auto-assign public IPv4 address (No), and Owner (495336230329). The right column includes: State (available), IPv4 CIDR (10.0.1.0/24), IPv6 CIDR (-), Route Table (rtb-0f2bc0c480711a6de | Private_Nagp_RT), Default subnet (No), and Auto-assign IPv6 address (No).

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone
Public_Nagp_Subnet	subnet-04c9ee257c6ef6422	available	vpc-094f864a85fe52c9f ...	10.0.0.0/24	250	-	ap-south-1a
Private_Nagp_Subnet	subnet-07a193036e44836b1	available	vpc-094f864a85fe52c9f ...	10.0.1.0/24	250	-	ap-south-1b

Subnet: subnet-07a193036e44836b1

Property	Value
Subnet ID	subnet-07a193036e44836b1
VPC	vpc-094f864a85fe52c9f Nagp_VPC
Available IPv4 Addresses	250
Availability Zone	ap-south-1b (aps1-az3)
Network ACL	acl-0433a1727a6b22dac Private_Nagp_NACL
Auto-assign public IPv4 address	No
Owner	495336230329
State	available
IPv4 CIDR	10.0.1.0/24
IPv6 CIDR	-
Route Table	rtb-0f2bc0c480711a6de Private_Nagp_RT
Default subnet	No
Auto-assign IPv6 address	No

4. Public_Nagp_Subnet_1 : 10.0.4.0/24

The screenshot shows the AWS console details for the subnet **Public_NAGP_Subnet_1** (subnet-0978ce23778b118ca). The subnet is in an **available** state and is associated with the VPC **vpc-094f864a85fe52c9f**. It has an IPv4 CIDR of 10.0.4.0/24 and is located in the **ap-south-1b** availability zone. The subnet is associated with the route table **rtb-03d4c03443cd0e764** (Public_Nagp_RT) and the network ACL **acl-0c8659473a55ddea6** (Public_nagp_NACL). The owner is 495336230329.

Property	Value
Subnet ID	subnet-0978ce23778b118ca
State	available
VPC	vpc-094f864a85fe52c9f Nagp_VPC
Available IPv4 Addresses	248
IPv4 CIDR	10.0.4.0/24
Availability Zone	ap-south-1b (aps1-az3)
IPv6 CIDR	-
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

The screenshot shows the AWS console details for the subnet **Private_NAGP_Subnet_1** (subnet-0ea9a554289cbfb4). The subnet is in an **available** state and is associated with the VPC **vpc-094f864a85fe52c9f**. It has an IPv4 CIDR of 10.0.5.0/24 and is located in the **ap-south-1a** availability zone. The subnet is associated with the route table **rtb-0f2bc0c480711a6de** (Private_Nagp_RT) and the network ACL **acl-0433a1727a6b22dac** (Private_Nagp_NACL). The owner is 495336230329.

Property	Value
Subnet ID	subnet-0ea9a554289cbfb4
State	available
VPC	vpc-094f864a85fe52c9f Nagp_VPC
Available IPv4 Addresses	250
IPv4 CIDR	10.0.5.0/24
Availability Zone	ap-south-1a (aps1-az1)
IPv6 CIDR	-
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)

The screenshot shows the AWS console details for the route table **Public_Nagp_RT** (rtb-03d4c03443cd0e764). The route table is associated with the subnet **subnet-04c9ee257c6ef6422**. It has two routes: one for destination 10.0.0.0/16 with target **local**, and another for destination 0.0.0.0/0 with target **igw-0432df096eae556b2**. Both routes are in an **active** status and have not been propagated.

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

7. Private_nagp_RT (Route table)

The screenshot shows the AWS Management Console interface for the **Private_Nagp_RT** route table. The table is associated with the subnet `subnet-07a193036e44836b1` and is not the main route table for the VPC. The **Routes** tab is selected, showing a single route with the destination `10.0.0.0/16`, target `local`, and status `active`.

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-0f2bc0c480711a6de

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

8. Private_nagp_NACL : inbound rules

The screenshot shows the AWS Management Console interface for the **Private_nagp_NACL** Network ACL. The **Inbound Rules** tab is selected, showing a list of rules. The rules are numbered 100 through 130, with a final `*` rule for all traffic. The rules allow SSH (22) and MySQL/Aurora (3306) traffic from specific source IP ranges, and allow all other traffic.

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	SSH (22)	TCP (6)	22	10.0.0.0/24	ALLOW
110	Custom TCP Rule	TCP (6)	1024 - 65535	0.0.0.0/0	ALLOW
120	SSH (22)	TCP (6)	22	10.0.4.0/24	ALLOW
130	MySQL/Aurora (3306)	TCP (6)	3306	10.0.0.0/24	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

9. Private_nagp_NACL : outbound rules

The screenshot shows the AWS Management Console interface for the **Private_nagp_NACL** Network ACL. The **Outbound Rules** tab is selected, showing a list of rules. The rules are numbered 100 through 130, with a final `*` rule for all traffic. The rules allow all other traffic to the destination `0.0.0.0/0`.

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

10. Public_nagp_NACL : inbound rules

Public_nagp_NACL acl-0c8659473a55... 2 Subnets No vpc-094f864a85fe52c9f | Nagp_VPC

Edit inbound rules

View All 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

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

Name	Network ACL ID	Associated with	Default	VPC	Owner
Private_Na...	acl-0433a1727a6b...	subnet-07a19303...	No	vpc-094f864a85fe52c9f Nagp_VPC	495336230329
Public_nag...	acl-0c8659473a55...	subnet-04c9ee257...	No	vpc-094f864a85fe52c9f Nagp_VPC	495336230329
	acl-0df32dca26aa...		Yes	vpc-094f864a85fe52c9f Nagp_VPC	495336230329

Details Inbound Rules Outbound Rules Subnet associations Tags

Edit outbound rules

View All 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

Security Groups (1/4) Info

Filter security groups

Name	Security group ID	Security group name	VPC ID
-	sg-09dc45e95e214306e	Private_Nagp_SG	vpc-094f864a85fe52c9f

Details Inbound rules Outbound rules Tags

Inbound rules Edit 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

The screenshot shows the AWS Management Console interface for the 'Security Groups' page. The 'Outbound rules' tab is selected. The table below shows the outbound rules for the 'Private_Nagp_SG' security group.

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	0.0.0.0/0	-

14. Public_Nagp_SG (Security Group) : inbound rules

The screenshot shows the AWS Management Console interface for the 'Security Groups' page. The 'Inbound rules' tab is selected. The table below shows the inbound rules for the 'Public_Nagp_SG' security group.

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	0.0.0.0/0	-

15. Public_Nagp_SG (Security Group) : outbound rules

The screenshot shows the AWS Management Console interface for the 'Security Groups' page. The 'Outbound rules' tab is selected. The table below shows the outbound rules for the 'Public_Nagp_SG' security group.

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	0.0.0.0/0	-

16. MySQL Security Group

Inbound rules					Edit inbound rules
Type	Protocol	Port range	Source	Description	
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					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

>

⚙

<div>☐</div>	Name ▾	Internet gateway ID ▾	State ▾	VPC ID
<div>☑</div>	NAGP_Internet_Gat...	igw-0432df096eae556b2	✔ Attached	vpc-094f864a85fe52c9f Na

Key	Value
Name	NAGP_Internet_Gateway