

# VPC Task -1

## 1. Created VPC with 2 private and 2 public subnets.

The screenshot shows the AWS Management Console interface. On the left, the 'Virtual private cloud' section is expanded, showing 'Your VPCs', 'Subnets', 'Route tables', and 'Internet gateways'. The main area displays a table of VPCs with columns: Name, VPC ID, State, Encryption c..., Encryption control ..., Block Public..., and IPv... The table lists 'my-vpc' with VPC ID 'vpc-0277d18e91aa78b45' and State 'Available'. Below this, a table of subnets is shown with columns: Name, Subnet ID, State, VPC, Block Public..., and IPv4 CIDR. The subnets listed are 'public-subnet-2', 'public-subnet-1', 'private-subnet-2', and 'private-subnet-1', all with State 'Available' and associated with 'my-vpc'. The 'private-subnet-1' row is highlighted.

Select a subnet

## 2. Enabled DNS Hostname in VPC.

- Enabled the dns resolution and dns hostname.
- I went to vpc > actions > vpc settings > then enable DNS

The screenshot shows the 'Edit VPC settings' page in the AWS Management Console. The breadcrumb trail is 'VPC > Your VPCs > vpc-0277d18e91aa78b45 > Edit VPC settings'. The page has three sections: 'VPC details' showing VPC ID 'vpc-0277d18e91aa78b45' and Name 'my-vpc'; 'DHCP settings' showing 'DHCP option set' 'dopt-047e914323e06077e'; and 'DNS settings' with two checkboxes: 'Enable DNS resolution' (checked) and 'Enable DNS hostnames' (checked). Both checkboxes have an 'Info' link next to them.

## 3. Enable Auto Assign Public IP in 2 public subnets.

The screenshot shows the 'Edit subnet settings' page in the AWS Management Console. The breadcrumb trail is 'VPC > Subnets > subnet-08b26f68db3efb950 > Edit subnet settings'. The page has three sections: 'Subnet' showing Subnet ID 'subnet-08b26f68db3efb950' and Name 'Public-subnet-1'; 'Auto-assign IP settings' with the description 'Enable AWS to automatically assign a public IPv4 or IPv6 address to a new primary network interface for an instance in this subnet.' and two checkboxes: 'Enable auto-assign public IPv4 address' (checked) and 'Enable auto-assign customer-owned IPv4 address' (unchecked, with a note 'Option disabled because no customer owned pools found.'). Both checkboxes have an 'Info' link next to them; 'Enable resource name (RBN) settings' (unchecked) with the description 'Specify the hostname type for EC2 instances in this subnet and optional RBN DNS query settings.' and a note 'Option disabled because no customer owned pools found.'; and 'Resource-based name (RBN) settings' (unchecked) with the description 'Specify the hostname type for EC2 instances in this subnet and optional RBN DNS query settings.' and a note 'Option disabled because no customer owned pools found.'

VPC > Subnets > subnet-03ca7dc51a3996af4 > Edit subnet settings

### Edit subnet settings [Info](#)

Subnet

Subnet ID

subnet-03ca7dc51a3996af4

Name

public-subnet-2

Auto-assign IP settings [Info](#)

Enable AWS to automatically assign a public IPv4 or IPv6 address to a new primary network interface for an instance in this subnet.

☒ Enable auto-assign public IPv4 address [Info](#)

☐ Enable auto-assign customer-owned IPv4 address [Info](#)  
Option disabled because no customer owned pools found.

Resource-based name (RBN) settings [Info](#)

#### 4. Add 2 private subnets in private route table.

- I went to route table < create private root < subnet < edit subnet association < add private subnet.

Subnet associations

Explicit subnet associations (2) [Edit subnet associations](#)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
private-subnet-1	<a href="#">subnet-096d33cb6dba2313f</a>	198.168.3.0/24	-
private-subnet-2	<a href="#">subnet-004502991cfb4e6e1</a>	198.168.4.0/24	-

Subnets without explicit associations (2) [Edit subnet associations](#)

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
Public-subnet-1	<a href="#">subnet-08b26f68db3efb950</a>	198.168.1.0/24	-
public-subnet-2	<a href="#">subnet-03ca7dc51a3996af4</a>	198.168.2.0/24	-

#### 5. Add 2 public subnets in public route table.

- I went to route table < create public root < subnet < edit subnet association < add public subnet.

Route tables > rtb-0a8fb6ef8cd6b758b

You have successfully updated subnet associations for rtb-0a8fb6ef8cd6b758b / public-route.

Subnet associations

Explicit subnet associations (2) [Edit subnet associations](#)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
Public-subnet-1	<a href="#">subnet-08b26f68db3efb950</a>	198.168.1.0/24	-
public-subnet-2	<a href="#">subnet-03ca7dc51a3996af4</a>	198.168.2.0/24	-

Subnets without explicit associations (0) [Edit subnet associations](#)

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
------	-----------	-----------	-----------

No subnets without explicit associations

## 6. Public route table will have the routes to internet and local.

- Created new internetgateway for my vpc
- opened **Edit routes** for the public-route table.
- verified the **local route (198.168.0.0/16 → local)** which AWS added automatically.
- then **added a new route (0.0.0.0/0 → your Internet Gateway igw-0be596178cec47331)** to enable

VPC > Route tables > rtb-0a8fb6ef8cd6b758b > Edit routes

### Edit routes

Destination	Target	Status	Propagated	Route Origin
198.168.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	Internet Gateway	-	No	CreateRoute

igw-0be596178cec47331 (my-internet-gateway)

Use: "igw-0be596178cec47331"

igw-0be596178cec47331 (my-internet-gateway)

Add route

Cancel Preview Save changes

## 7. Create EC2 in public subnet with t2.micro and install PHP.

EC2 > Instances > i-028f17a283b09fa61

### Instance summary for i-028f17a283b09fa61 (Server\_with\_VPC)

Updated less than a minute ago

<b>Instance ID</b> i-028f17a283b09fa61	<b>Public IPv4 address</b> 44.202.223.145   open address	<b>Private IPv4 addresses</b> 198.168.1.17
<b>IPv6 address</b> -	<b>Instance state</b> Running	<b>Public DNS</b> ec2-44-202-223-145.compute-1.amazonaws.com   open address
<b>Hostname type</b> IP name: ip-198-168-1-17.ec2.internal	<b>Private IP DNS name (IPv4 only)</b> ip-198-168-1-17.ec2.internal	<b>Elastic IP addresses</b> -
<b>Answer private resource DNS name</b> -	<b>Instance type</b> t2.micro	<b>AWS Compute Optimizer finding</b> Opt-in to AWS Compute Optimizer for recommendations.   Learn more
<b>Auto-assigned IP address</b> 44.202.223.145 [Public IP]	<b>VPC ID</b> vpc-0277d18e91aa78b45 (my-vpc)	<b>Activate Windows</b> Go to Settings to activate Windows.

## PHP Version 8.4.14



System	Linux ip-198-168-1-17.ec2.internal 6.1.158-178.288.amzn2023.x86_64 #1 SMP PREEMPT_DYNAMIC Mon Nov 3 18:38:36 UTC 2025 x86_64
Build Date	Oct 21 2025 19:23:55
Build System	Linux
Build Provider	Amazon Linux
Compiler	gcc (GCC) 11.5.0 20240719 (Red Hat 11.5.0-5)
Architecture	x86_64
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/10-opcache.ini, /etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-dom.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-mbstring.ini, /etc/php.d/20-mysqli.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-posix.ini, /etc/php.d/20-shmop.ini, /etc/php.d/20-simplexml.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sodium.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-sysvmsg.ini, /etc/php.d/20-sysvsem.ini, /etc/php.d/20-sysvshm.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/20-xml.ini, /etc/php.d/20-xmlwriter.ini, /etc/php.d/20-xsl.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini, /etc/php.d/30-xmlreader.ini
PHP API	20240924
PHP Extension	20240924
Zend Extension	420240924
Zend Extension Build	API20240924.NTS
PHP Extension Build	API20240924.NTS
PHP Integer Size	64 bits
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled

Activate Window  
Go to Settings to activate

```
Verifying : php8.4-mysqlnd-8.4.14-1.amzn2023.0.1.x86_64
Verifying : php8.4-opcache-8.4.14-1.amzn2023.0.1.x86_64
Verifying : php8.4-pdo-8.4.14-1.amzn2023.0.1.x86_64
Verifying : php8.4-process-8.4.14-1.amzn2023.0.1.x86_64
Verifying : php8.4-sodium-8.4.14-1.amzn2023.0.1.x86_64
Verifying : php8.4-xml-8.4.14-1.amzn2023.0.1.x86_64

Installed:
apr-1.7.5-1.amzn2023.0.4.x86_64      apr-util-1.6.3-1.amzn2023.0.2.x86_64      apr-util-1mdb-1.6.3-1.amzn2023.0.2.x86_64      apr-util-openssl-1.6.3-1.amzn2023.0.2.x86_64      httpd-2.4.65-1.amzn2
PHP 8.4.14 (cli) (built: Oct 21 2025 19:23:55) (NTS gcc x86_64)
Copyright (c) The PHP Group
Built by Amazon Linux
Zend Engine v4.4.14, Copyright (c) Zend Technologies
with Zend OPcache v8.4.14, Copyright (c), by Zend Technologies
[ec2-user@ip-198-168-1-17 ~]$ sudo systemctl start php-fpm
sudo systemctl enable php-fpm
Created symlink /etc/systemd/system/multi-user.target.wants/php-fpm.service → /usr/lib/systemd/system/php-fpm.service.
[ec2-user@ip-198-168-1-17 ~]$ sudo systemctl status php-fpm
● php-fpm.service - The PHP FastCGI Process Manager
   Loaded: loaded (/usr/lib/systemd/system/php-fpm.service; enabled; preset: disabled)
   Active: active (running) since Fri 2025-11-21 12:59:21 UTC; 31s ago
     Main PID: 26597 (php-fpm)
    Status: "Processes active: 0, idle: 5, Requests: 0, slow: 0, Traffic: 0.00req/sec"
      Tasks: 6 (limit: 1106)
     Memory: 10.3M
        CPU: 50ms
    CGroup: /system.slice/php-fpm.service
            └─26597 "php-fpm: master process (/etc/php-fpm.conf)"
               └─26616 "php-fpm: pool www"
                  └─26617 "php-fpm: pool www"
                     └─26618 "php-fpm: pool www"
                        └─26619 "php-fpm: pool www"
                           └─26620 "php-fpm: pool www"

Nov 21 12:59:21 ip-198-168-1-17.ec2.internal systemd[1]: Starting php-fpm.service - The PHP FastCGI Process Manager...
Nov 21 12:59:21 ip-198-168-1-17.ec2.internal systemd[1]: Started php-fpm.service - The PHP FastCGI Process Manager.
[ec2-user@ip-198-168-1-17 ~]$ sudo nano /etc/nginx/nginx.conf
[ec2-user@ip-198-168-1-17 ~]$ sudo vi /etc/nginx/nginx.conf
[ec2-user@ip-198-168-1-17 ~]$ sudo systemctl restart nginx
[ec2-user@ip-198-168-1-17 ~]$ sudo systemctl php-fpm
Unknown command verb php-fpm.
[ec2-user@ip-198-168-1-17 ~]$ sudo systemctl start php-fpm
[ec2-user@ip-198-168-1-17 ~]$ echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/index.php
<?php phpinfo(); ?>
[ec2-user@ip-198-168-1-17 ~]$ sudo systemctl restart nginx
[ec2-user@ip-198-168-1-17 ~]$ sudo systemctl restart php-fpm
[ec2-user@ip-198-168-1-17 ~]$ sudo root /usr/share/nginx/html;
-bash: root: command not found
[ec2-user@ip-198-168-1-17 ~]$ sudo root /usr/share/nginx/html
-bash: root: command not found
[ec2-user@ip-198-168-1-17 ~]$ cd root /usr/share/nginx
-bash: cd: too many arguments
[ec2-user@ip-198-168-1-17 ~]$ cd root /usr/share/nginx/
-bash: cd: too many arguments
[ec2-user@ip-198-168-1-17 ~]$ cd /usr/share/nginx/
[ec2-user@ip-198-168-1-17 ~]$ cd /usr/share/nginx/
[ec2-user@ip-198-168-1-17 ~]$ ls
html modules
[ec2-user@ip-198-168-1-17 ~]$ nginx$ vi html
[ec2-user@ip-198-168-1-17 ~]$ nginx$ sudo mv /var/www/html/index.php /usr/share/nginx/html/index.php
[ec2-user@ip-198-168-1-17 ~]$ nginx$ sudo rm -f /usr/share/nginx/html/index.html
[ec2-user@ip-198-168-1-17 ~]$ nginx$ sudo systemctl restart php-fpm
[ec2-user@ip-198-168-1-17 ~]$ sudo systemctl restart nginx
[ec2-user@ip-198-168-1-17 ~]$ nginx$
```

## 8. Configure NAT gateway in public subnet and connect to private instance.

### **I created an Elastic IP**

You must allocate a new Elastic IP:

AWS Console > **VPC** > **Elastic IPs** → Allocate.

---

### **I created a NAT Gateway in the Public Subnet**

AWS Console > **VPC** > **NAT Gateways** > **Create NAT Gateway**

Choose:

- **Subnet** → Any *public* subnet
- **Elastic IP** → Select the one you allocated
- **Name** → nat-gw

Click **Create NAT Gateway**

---

### **I updated the Private Route Table**

Go to:

**VPC** → **Route Tables** → **Select the private route table**

Then:

Route → **0.0.0.0/0**

Target → **NAT Gateway (nat-gw-xxxx)**

Click **Save Changes**

This ensures ALL private instances route outbound traffic through the NAT Gateway.

---

### **I confirmed private subnets are associated with Private Route Table**

Under the same route table:

- Subnet associations →  
Both **private-subnet-1** and **private-subnet-2** must be selected.
- 

### **I launched an EC2 instance in Private Subnet**

You will now:

- Launch instance
  - Choose **private subnet**
  - Auto-assign public IP = **Disabled**
  - Security Group → allow SSH **only from the public EC2** (optional)
- 

### **I verified private instance can access Internet (via NAT Gateway)**

SSH to public EC2:

```
ssh -i key.pem ec2-user@<public-ec2-ip>
```

Then from public EC2, SSH to private EC2:

```
ssh ec2-user@<private-ec2-ip>
```

Test internet:

```
ping google.com
```

```
sudo yum install git -y
```

[illegible]

## 9. Install Apache Tomcat in private EC2 and deploy a sample app.

- **Connected to Private EC2 through Public EC2**
- SSH → public instance
- From there SSH → private instance (jump host method)

---

### 2. Installed Java and Set Up Tomcat

- Downloaded Tomcat, extracted to /opt/tomcat
- Created tomcat user
- Gave ownership:
- `sudo chown -R tomcat: /opt/tomcat`

---

### 3. Fixed the Permission Issue (bin Folder Not Opening)

- You were stuck entering the bin folder.

You fixed it using:

- `sudo chmod -R 755 /opt/tomcat/bin`
- Then successfully ran:
- `cd /opt/tomcat/bin`

---

### 4. Created Tomcat Systemd Service & Started Tomcat

- `sudo systemctl daemon-reload`
- `sudo systemctl start tomcat`
- `sudo systemctl enable tomcat`

---

### 5. Deployed a Simple Test App

- Created a simple JSP app:
- `echo "<h1>Hello from Tomcat!</h1>" | sudo tee /opt/tomcat/webapps/ROOT/test.jsp`

---

### 6. Verified Using curl from Public Instance

- You tested the app:
- `curl http://<private-ec2-ip>:8080/test.jsp`

```
50 sudo wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.112/bin/apache-tomcat-9.0.112.tar.gz
51 ls
52 rm tomcat
53 sudo rm tomcat
54 sudo tar -xvzf apache-tomcat-9.0.112.tar.gz
55 ls
56 mv apache-tomcat-9.0.112 tomcat9
57 sudo mv apache-tomcat-9.0.112 tomcat9
58 cd tomcat9
59 ls
60 sudo cd bin
61 ls
62 sudo bin/
63 sudo cd bin/
64 sudo cd /bin
65 cd bin
66 sudo cd bin
67 cd webapp
68 cd webapps
69 sudo cd webapps
70 sudo chown -R ec2-user:ec2-user /opt/tomcat9
71 cd /opt/tomcat9
72 ls -l
73 cd bin
74 sudo ./startup.sh
75 cd /opt/tomcat9/webapps
76 sudo wget https://tomcat.apache.org/tomcat-9.0-doc/appdev/sample/sample.war
77 cd /opt/tomcat9/bin
78 sudo ./shutdown.sh
79 sudo ./startup.sh
80 curl http://198.168.3.186:8080/sample/
```

Activate  
Go to Settings

```
awscli> aws ec2-user ec2-user 81 Nov 8 07:46 webapps
[ec2-user@ip-198-168-3-186 ~]$ cd bin
[ec2-user@ip-198-168-3-186 tomcat9]$ cd bin
[ec2-user@ip-198-168-3-186 bin]$ sudo ./startup.sh
Using CATALINA_BASE:   /opt/tomcat9
Using CATALINA_HOME:   /opt/tomcat9
Using CATALINA_TMPDIR: /opt/tomcat9/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/tomcat9/bin/bootstrap.jar:/opt/tomcat9/bin/tomcat-juli.jar
Tomcat started.
[ec2-user@ip-198-168-3-186 bin]$ cd /opt/tomcat9/webapps
[ec2-user@ip-198-168-3-186 webapps]$ sudo wget https://tomcat.apache.org/tomcat-9.0-doc/appdev/sample/sample.war
--2023-11-22 18:07:05-- https://tomcat.apache.org/tomcat-9.0-doc/appdev/sample/sample.war
Resolving tomcat.apache.org (tomcat.apache.org)... 151.101.2.132, 204.4642.104
Connecting to tomcat.apache.org (tomcat.apache.org)[151.101.2.132]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4606 (4.5K)
Saving to: 'sample.war'

sample.war                                     100%[=====] 4.50K  ---KB/s  in 0s

2023-11-22 18:07:05 (83.3 MB/s) - 'sample.war' saved [4606/4606]
[ec2-user@ip-198-168-3-186 webapps]$ cd /opt/tomcat9/bin
[ec2-user@ip-198-168-3-186 bin]$ sudo ./startup.sh
Using CATALINA_BASE:   /opt/tomcat9
Using CATALINA_HOME:   /opt/tomcat9
Using CATALINA_TMPDIR: /opt/tomcat9/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/tomcat9/bin/bootstrap.jar:/opt/tomcat9/bin/tomcat-juli.jar
Tomcat started.
[ec2-user@ip-198-168-3-186 bin]$ curl http://198.168.3.186:8080/sample/
ok:
<title>Sample "Hello, World" Application</title>
</head>
<body bgcolor=white>
<table border="0">
<tr>
<td>

</td>
<td>
<div>Sample "Hello, World" Application</div>
<div>This is the home page for a sample application used to illustrate the
source directory organization of a web application utilizing the principles
outlined in the Application Developer's Guide.
</div>
</td>
</tr>
</table>
<p>To prove that they work, you can execute either of the following links:<br>
<a href="hello.jsp">JSP page</a>.<br>
<a href="hello-servlet/>
</p>
</body>
</html>
[ec2-user@ip-198-168-3-186 bin]$
```

Activate Windows  
Go to Settings to activate Windows.

## 10. Configure VPC flow logs and store the logs in S3 and CloudWatch.

### Steps:

1. Go to VPC Console, select your VPC (project-vpc), and open the Flow Logs tab.
2. Click Create flow log and enter the name: vpc-flowlogs-cw.
3. Set Filter = All and Aggregation Interval = 1 or 10 minutes.
4. Choose Destination → CloudWatch Logs, then create a new log group:
5. /vpc/flowlogs/project-vpc
6. Under Service Access, click Set up permissions → let AWS create a new IAM role → select that role.
7. Click Create flow log and CloudWatch will start receiving VPC traffic logs.



#### Filter

The type of traffic to capture (accepted traffic only, rejected traffic only, or all traffic).

- ☐ Accept
- ☐ Reject
- ☒ All

#### Maximum aggregation interval [Info](#)

The maximum interval of time during which a flow of packets is captured and aggregated into a flow log record.

- ☒ 10 minutes
- ☐ 1 minute

#### Destination

The destination to which to publish the flow log data.

- ☒ Send to CloudWatch Logs
- ☐ Send to an Amazon S3 bucket
- ☐ Send to Amazon Data Firehose in the same account
- ☐ Send to Amazon Data Firehose in a different account

#### Destination log group [Info](#)

The name of an existing log group or the name of a new log group that will be created when you create this flow log. A new log stream is created for each monitored network interface.

#### Service access

VPC flow logs require permissions to create log groups and publish events in CloudWatch.

- ☐ Use an existing service role
- ☒ Create and use a new service role

#### Service role name [Info](#)

#### Log record format

Specify the fields to include in the flow log record.

- ☒ AWS default format
- ☐ Custom format

#### Additional metadata

Include additional metadata to AWS default log record format.

- ☐ Include Amazon ECS metadata

#### Format preview

```
$(version) ${account-id} ${interface-id} ${srcaddr} ${dstaddr} ${srcport} ${dstport} ${protocol} ${packets} ${bytes} ${start} ${end} ${action} ${log-status}
```

## Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS

#### Key

#### Value - optional

[Add tag](#)

You can add 49 more tags

▼ Log group details

Log class [Info](#)

Standard

ARN

arn:aws:logs:us-east-1:679625722057:log-group:/vpc/flowlogs/project-vpc:\*

Creation time

28 minutes ago

Retention

Never expire

Stored bytes

-

Metric filters

0

Subscription filters

0

Contributor Insights rules

-

KMS key ID

-

Data protection

-

Sensitive data count

-

Custom field indexes

[Configure](#)

Transformer

[Configure](#)

Anomaly detection

[Configure](#)

Log streams

Tags

Anomaly detection

Metric filters

Subscription filters

Contributor Insights

Data protection

Field indexes

Transformer

Log streams (3)

[Delete](#)

[Create log stream](#)

[Search all log streams](#)

By default, we only load the most recent log streams.

☐ Exact match
☐ Show expired [Info](#)

Filter log streams or try prefix search

< 1 > ⚙

Activate Windows

Go to Settings to activate Windows.

## VPC Flow Logs to S3 — Simple 7 Steps

1. Create an S3 bucket (example: mmyflowlog-s3)
2. Copy the bucket ARN from S3 → Properties (example: arn:aws:s3:::mmyflowlog-s3)
3. Go to VPC → Your VPCs → Flow Logs → Create flow log
4. Set Filter = All and Aggregation = 10 minutes
5. Choose Destination = Send to Amazon S3 bucket
6. Paste your S3 bucket ARN and keep default log format
7. Click Create flow log — DONE

## New Bucket created:

fl-0c89d5e5e649f9b2a / flowlog-s3

[Actions](#)

Details

Flow Log ID

fl-0c89d5e5e649f9b2a

Name

flowlog-s3

State

Active

Creation Time

Sunday, November 23, 2025 at 00:29:52 GMT+5:30

Destination Type

s3

Destination Name

[mmyflowlog-s3](#)

IAM Role

-

Cross Account IAM Role

-

Traffic Type

All

Max Aggregation Interval

10 minutes

Log Format

Default

File Format

Plain text

Hive Compatible Partitions

Not enabled

Partition Logs

Daily

Tags

Integrations

Tags

[Manage tags](#)

Search tags

< 1 > ⚙

Key	Value
Name	flowlog-s3

## Filter

The type of traffic to capture (accepted traffic only, rejected traffic only, or all traffic).

- ☐ Accept
- ☐ Reject
- ☒ All

## Maximum aggregation interval | [Info](#)

The maximum interval of time during which a flow of packets is captured and aggregated into a flow log record.

- ☒ 10 minutes
- ☐ 1 minute

## Destination

The destination to which to publish the flow log data.

- ☐ Send to CloudWatch Logs
- ☒ Send to an Amazon S3 bucket
- ☐ Send to Amazon Data Firehose in the same account
- ☐ Send to Amazon Data Firehose in a different account

## S3 bucket ARN

The ARN of the Amazon S3 bucket to which the flow log is published. You can specify a specific folder in the bucket using the arn:aws:s3:::[bucket-name]

*S3 bucket ARN...*

Must be in the format: arn:aws:s3:::[bucket-name]

**i** Please note, a resource-based policy will be created for you and attached to the target bucket.

## Log record format

Specify the fields to include in the flow log record.

- ☒ AWS default format
- ☐ Custom format

## Additional metadata

Include additional metadata to AWS default log record format.

- ☐ Include Amazon ECS metadata

## Format preview

`${version} ${account-id} ${interface-id} ${srcaddr} ${dstaddr} ${srcport} ${dstport} ${protocol} ${packets} ${bytes}`

### mmyflowlog-s3 [Info](#)

[Objects](#) | [Metadata](#) | [Properties](#) | [Permissions](#) | [Metrics](#) | [Management](#) | [Access Points](#)

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	<a href="#">AWSLogs/</a>	Folder	-	-	-