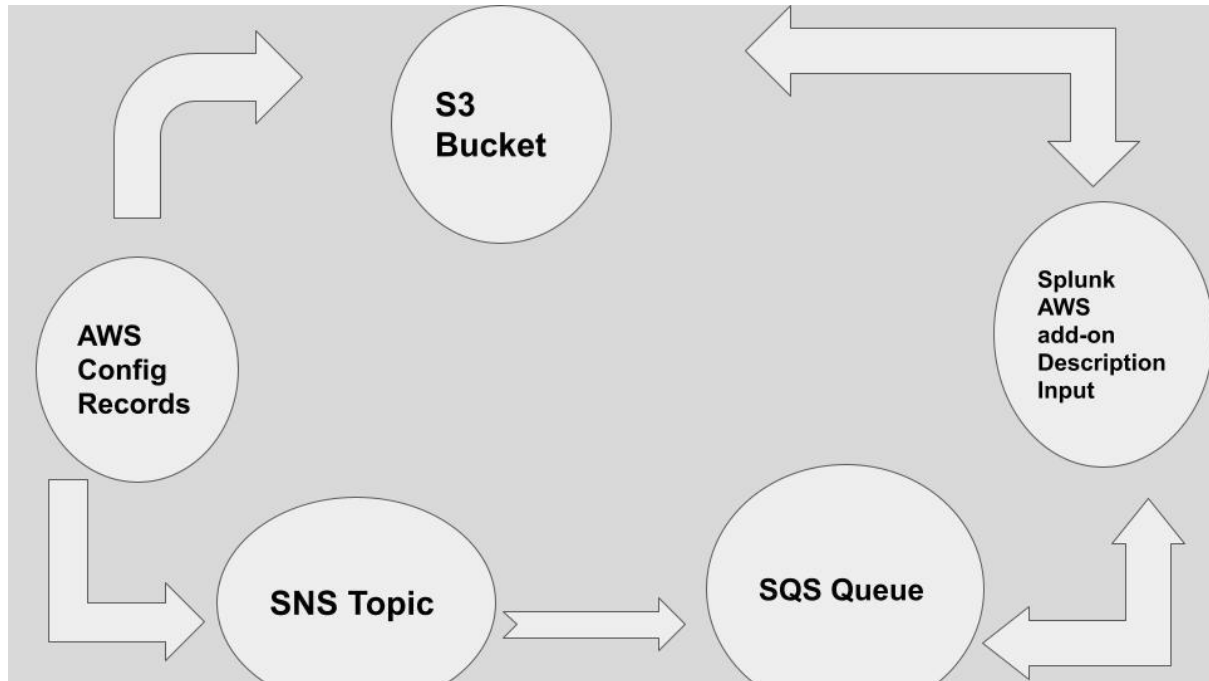


### ### AWS Config-Splunk ###

#### Architecture Flow of AWS Config send records to Splunk AWS add-on Description Input:



#### 1.AWS Config:

- 1.Open the AWS Config Console
- 2.In navigation pane,choose settings
3. I got below error when turn on the configuration recorder so first i will setup delivery channel



#### Error

There is no delivery channel available to record configurations.

## 2.Delivery Channel Setup:

### Purpose:

As AWS Config continually records the changes that occur to your AWS resources, it sends notifications and updated configuration states through the delivery channel. You can manage the delivery channel to control where AWS Config sends configuration updates.

#### 2.(a).1)Create the Amazon S3 bucket:

	Name ▲	AWS Region ▼	Access ▼	Creation date ▼
<input type="radio"/>	course-dhana	US East (Ohio) us-east-2	Bucket and objects not public	March 2, 2021, 11:21:17 (UTC+05:30)

2) In S3 buckets, click the S3 bucket just I created

3) Choose permissions,choose the bucket policy

Objects	Properties	Permissions	Metrics	Management	Access Points
---------	------------	-------------	---------	------------	---------------

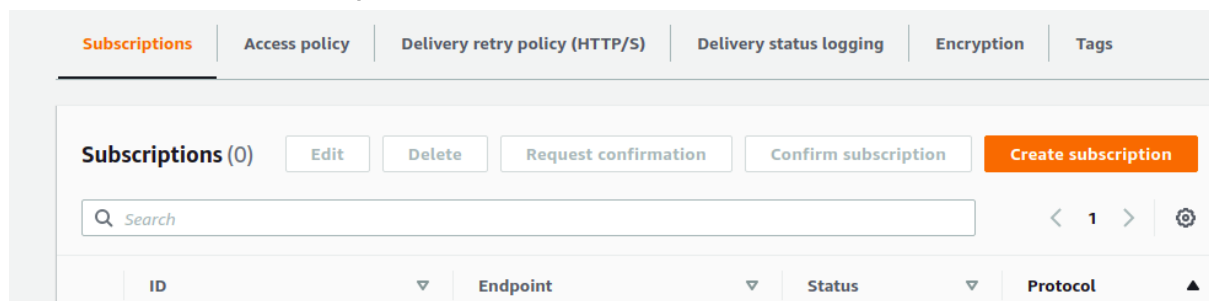
4) Copy and paste the following bucket policy, and then save the policy

## Policy

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Sid": "AWSConfigBucketPermissionsCheck",  
6       "Effect": "Allow",  
7       "Principal": {  
8         "Service": [  
9           "config.amazonaws.com"  
10        ]  
11      },  
12      "Action": "s3:GetBucketAcl",  
13      "Resource": "arn:aws:s3:::course-dhana"  
14    },  
15    {  
16      "Sid": "AWSConfigBucketExistenceCheck",  
17      "Effect": "Allow",  
18      "Principal": {  
19        "Service": [  
20          "config.amazonaws.com"  
21        ]  
22      },  
23      "Action": "s3:ListBucket",  
24      "Resource": "arn:aws:s3:::course-dhana"  
25    },  
26    {  
27      "Sid": "AWSConfigBucketDelivery",  
28      "Effect": "Allow",  
29    }  
30  ]  
31 }
```

## 2.(b)Create the SNS topic:

1. Open the Amazon SNS console in the same Region as your AWS Config service, and then click Topics.
2. Click Create topic.
3. enter a name for SNS topic, and then click Create topic.
4. Click Create subscription.



5. In Protocol, click Email and enter the email-address and click the create subscription

## Create subscription

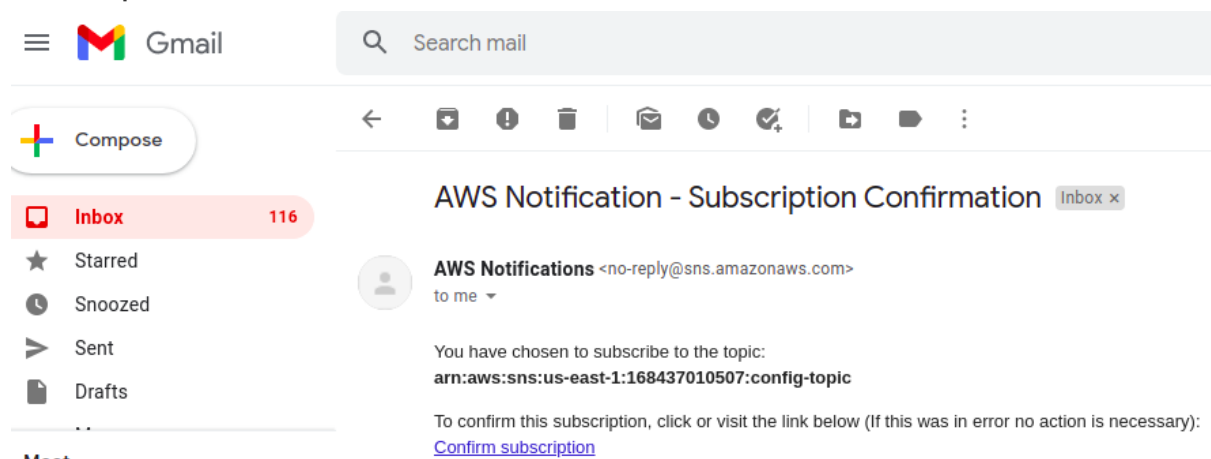
**Details**

**Topic ARN**

**Protocol**  
The type of endpoint to subscribe

**Endpoint**  
An email address that can receive notifications from Amazon SNS.

6. Check email for the subscription confirmation, and then click Confirm subscription.



7. receive the message Subscription confirmed!



### Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:

arn:aws:sns:us-east-1:168437010507:sns-topic:41dfa58e-2298-4d4c-a619-3d3653ca8068

If it was not your intention to subscribe, [click here to unsubscribe](#).

## 2.(c)Configure IAM Role:

### Purpose:

An IAM role is an IAM entity that defines a set of permissions for making AWS service requests. IAM roles are not associated with a specific user or group. Instead, trusted entities assume roles, such as IAM users, applications, or AWS services such as EC2.

### 1.Open the IAM console

### 2.Choose Roles, and then choose Create role

#### Identity and Access Management (IAM)

##### Dashboard

##### ▼ Access management

Groups

Users

**Roles**

Policies

Identity providers

Account settings

IAM roles issue keys that are valid for short duration

#### Additional resources:

- [IAM Roles FAQ](#)
- [IAM Roles Documentation](#)
- [Tutorial: Setting Up Cross Account Access](#)
- [Common Scenarios for Roles](#)

Create role

Delete role

Search





Role name ▼

### 3. Select type of trusted entity, Choose AWS Service:

Create role



Select type of trusted entity

 <b>AWS service</b> EC2, Lambda and others	 <b>Another AWS account</b> Belonging to you or 3rd party	 <b>Web identity</b> Cognito or any OpenID provider	 <b>SAML 2.0 federation</b> Your corporate directory
--	---	---	--

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose a use case

### 4. Choose a **Config**

## Choose a use case

---

### Common use cases

#### EC2

Allows EC2 instances to call AWS services on your behalf.

---

#### Lambda

Allows Lambda functions to call AWS services on your behalf.

---

### Or select a service to view its use cases

<a href="#">API Gateway</a>	<a href="#">CloudWatch Events</a>	<a href="#">EKS</a>
<a href="#">AWS Backup</a>	<a href="#">CodeBuild</a>	<a href="#">EMR</a>
<a href="#">AWS Chatbot</a>	<a href="#">CodeDeploy</a>	<a href="#">ElastiCache</a>
<a href="#">AWS Marketplace</a>	<a href="#">CodeGuru</a>	<a href="#">Elastic Beanstalk</a>
<a href="#">AWS Support</a>	<a href="#">CodeStar Notifications</a>	<a href="#">Elastic Container Registry</a>
<a href="#">Amplify</a>	<a href="#">Comprehend</a>	<a href="#">Elastic Container Service</a>
<a href="#">AppStream 2.0</a>	<a href="#">Config</a>	<a href="#">Elastic Transcoder</a>

5.Choose-Customizable

## Select your use case

### Config

Allows Config to call AWS services and collect resource configurations on your behalf.

### Config - Conformance Packs.

Allows Config to create and manage conformance packs on your behalf.

### Config - Customizable

Allows Config to call AWS services and collect resource configurations on your behalf.

### Config - Organizations

Allows Config to access Organizations resources on your behalf.

6. choose Next: Permissions Next: Tags, and then Next: Review

## 7. Enter Role name and Choose create Role

Create role



Review

Provide the required information below and review this role before you create it.

Role name\*

aws-config

Use alphanumeric and '+,=,.,@,-,\_' characters. Maximum 64 characters.

Role description

Allows Config to call AWS services and collect resource configurations on your behalf.

Maximum 1000 characters. Use alphanumeric and '+,=,.,@,-,\_' characters.

Trusted entities

AWS service: config.amazonaws.com

Policies

 AWSConfigRole 

Permissions boundary Permissions boundary is not set

\* Required

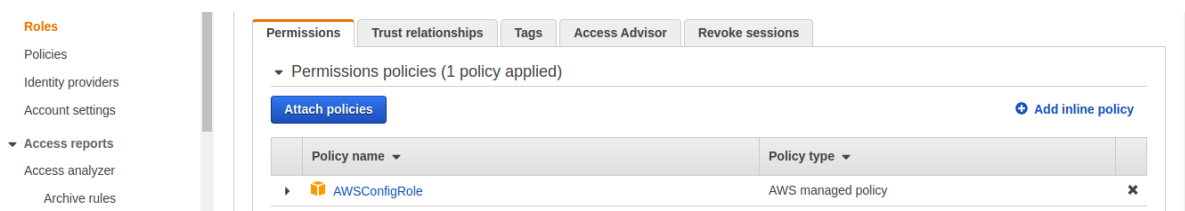
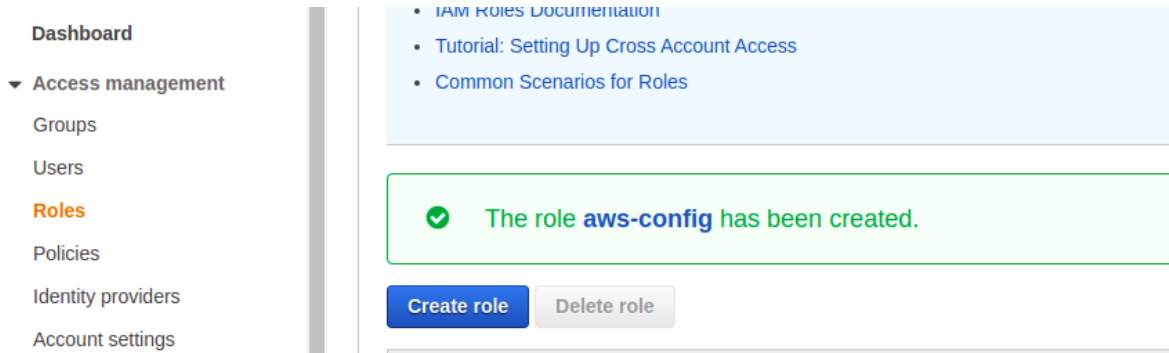
Cancel

Previous

Create role



7. Click the role that was created, click Add inline policy, and then click the JSON tab.



8. Copy and paste the following policy:

```
1 {
2     "Version": "2012-10-17",
3     "Statement": [
4         {
5             "Effect": "Allow",
6             "Action": [
7                 "s3:PutObject",
8                 "s3:PutObjectAcl"
9             ],
10            "Resource": [
11                "arn:aws:s3:::arn:aws:s3:::course-dhana/AWSLogs/168437010507/*"
12            ]
13        }
14    ]
15 }
```

## 2.(d) Create the delivery channel:

## 1. Install aws Command Line Interface on Linux:

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip"  
-o "awscliv2.zip"
```

```
dhana@dhana-Ubuntu:~$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
			Dload	Upload	Total	Spent	Left
							Speed
100	35.1M	100	35.1M	0	0	5154k	0
					0:00:06	0:00:06	--:--:-- 5389k

## 2.Unzip the Zip file:

```
unzip awscliv2.zip
```

```
dhana@dhana-Ubuntu:~$ unzip awscliv2.zip
Archive:  awscliv2.zip
replace aws/THIRD_PARTY_LICENSES? [y]es, [n]o, [A]ll, [N]one, [r]ename: A
  inflating: aws/THIRD_PARTY_LICENSES
  inflating: aws/install
  inflating: aws/README.md
  inflating: aws/dist/_sha512.cpython-38-x86_64-linux-gnu.so
  inflating: aws/dist/_asyncio.cpython-38-x86_64-linux-gnu.so
  inflating: aws/dist/_blake2.cpython-38-x86_64-linux-gnu.so
  inflating: aws/dist/_codecs_cn.cpython-38-x86_64-linux-gnu.so
```

### 3.Run the install program

```
sudo ./aws/install
```

```
dhana@dhana-Ubuntu:~$ sudo ./aws/install
[sudo] password for dhana:
```

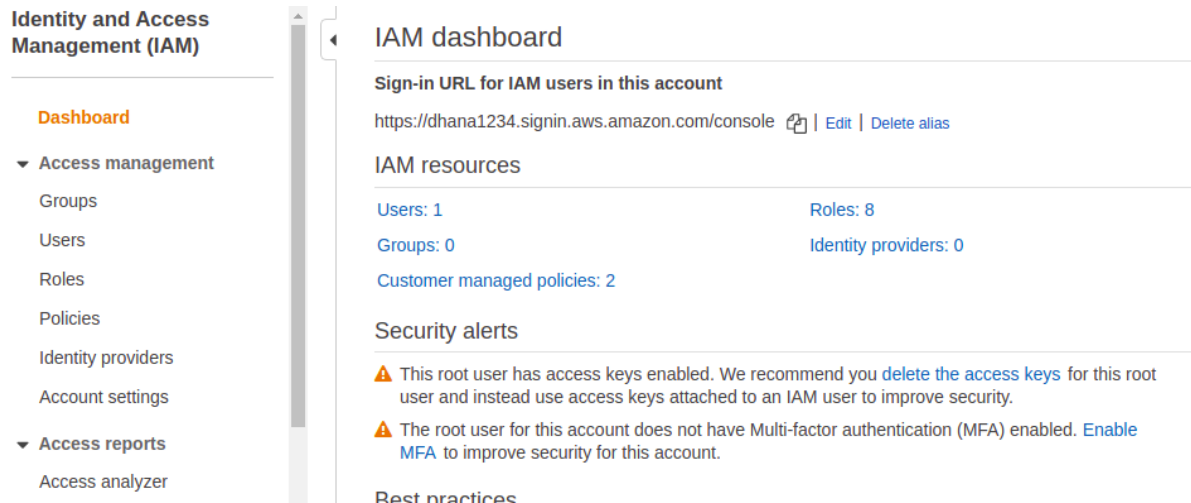
#### 4. Confirm the installation:

```
aws --version
```

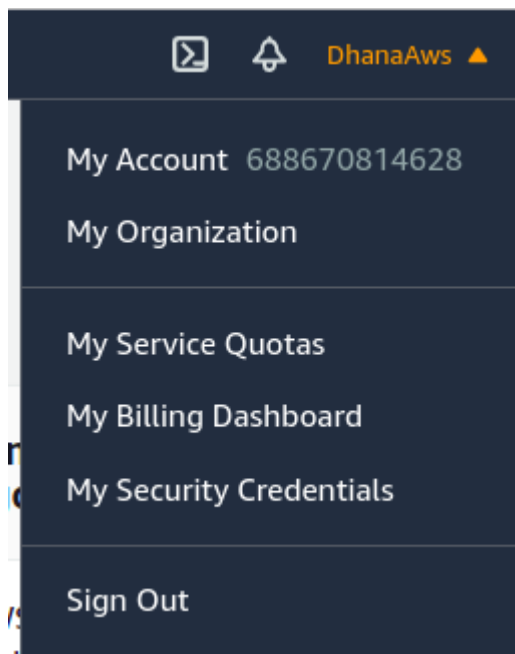
```
dhana@dhana-Ubuntu:~$ aws --version
aws-cli/2.1.28 Python/3.8.8 Linux/5.8.0-44-generic exe/x86_64.ubuntu.20 prompt/off
```

## 5.Configured AWS CLI:

### 1)Go to the IAM Dashboard console



### 2)click username Icon and blow image will appear



3. Click **My Security Credentials** and below image will appear:

## Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS IAM, see [AWS IAM User Guide](#).

To learn more about the types of AWS credentials and how they're used, see [AWS Security Credentials](#).

▼ Password

You use an email address and password to sign in to secure pages on AWS, such as the AWS Management Console. For your protection, create a password that contains many characters, including numbers and symbols, and change it periodically.

[Click here](#) to change the password, name, or email address for your root AWS account.

▲ Multi-factor authentication (MFA)

▲ Access keys (access key ID and secret access key)

▲ CloudFront key pairs

4) Click **Access keys (access key ID and secret access key)**

And Click **create access key** then below image will appear and you can download the **Download.csv** file:

Create access key ×

✓ Success

This is the **only** time that the secret access keys can be viewed or downloaded. You cannot recover them later. However, you can create new access keys at any time.

Download .csv file

Access key ID	Secret access key
AKIASON5A2BF5V432MXI	***** <a href="#">Show</a>

Close

5) Then Enter the **aws configure** in command line

```
dhana@dhana-Ubuntu:~$ aws configure
AWS Access Key ID [*****4JNH]: AKIASON5A2BFSBFYERSV
AWS Secret Access Key [*****EMmz]: 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
Default region name [us-east-2]: us-east-1
Default output format [json]: json
```

6) Create new json file In local System using name in deliveryChannel.json

```
dhana@dhana-Ubuntu:~$ vi deliveryChannel.json
```

Enter following format Policy in deliveryChannel.json:

```
{
  "name": "default",
  "s3BucketName": "targetBucketName",
  "snsTopicARN": "arn:aws:sns:region:account_number:targetTopicName",
  "configSnapshotDeliveryProperties": {
    "deliveryFrequency": "Twelve_Hours"
  }
}
```

**Replace:**

**s3BucketName:** course-dhana (we have already created, please the delivery channel setup)

**snsTopicARN:** "arn:aws:sns:us-east-1:168437010507:config-topic" (we have already created)

```
{
  "name": "default",
  "s3BucketName": "course-dhana",
  "snsTopicARN": "arn:aws:sns:us-east-1:168437010507:config-topic",
  "configSnapshotDeliveryProperties": {
    "deliveryFrequency": "Twelve_Hours"
  }
}
```

7) Run the following aws cli command:

```
aws configservice put-delivery-channel --delivery-channel
file://deliveryChannel.json
```

```
dhana@dhana-Ubuntu:~$ aws configservice put-delivery-channel --delivery-channel
file://deliveryChannel.json
dhana@dhana-Ubuntu:~$
```

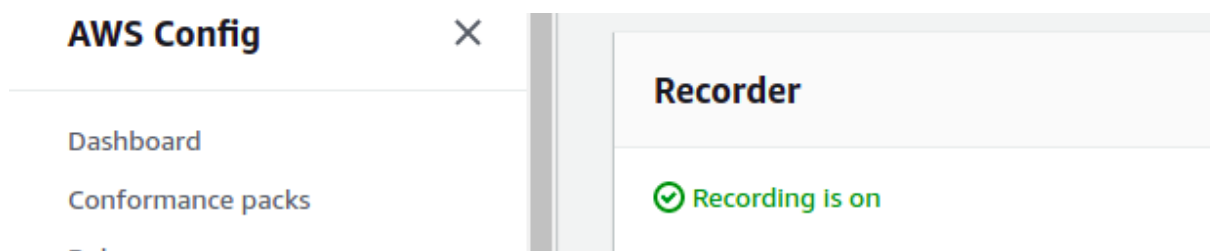
8). Run the following AWS CLI command to confirm that the Delivery Channel created:

**aws configservice describe-delivery-channels**

```
dhana@dhana-Ubuntu:~$ aws configservice describe-delivery-channels
{
  "DeliveryChannels": [
    {
      "name": "default",
      "s3BucketName": "course-dhana",
      "snsTopicARN": "arn:aws:sns:us-east-1:168437010507:config-topic",
      "configSnapshotDeliveryProperties": {
        "deliveryFrequency": "Twelve_Hours"
      }
    }
  ]
}
```

### 3.Start the configuration recorder:

1. Open the AWS Config Console:
2. In the navigation pane, choose Settings.
3. In Recording is off, click Turn on, and then choose Continue.



### 4.Configure Simple Queue service:

1. Open SQS Console
2. Click Create queue Then Enter Queue name and Click Create queue

**Details**

**Type**  
Choose the queue type for your application or cloud infrastructure.

You can't change the queue type after you create a queue.

☒ **Standard** [Info](#)  
At-least-once delivery, message ordering isn't preserved

- At-least once delivery
- Best-effort ordering

☐ **FIFO** [Info](#)  
First-in-first-out delivery, message ordering is preserved

- First-in-first-out delivery
- Exactly-once processing

**Name**

A queue name is case-sensitive and can have up to 80 characters. You can use alphanumeric characters, hyphens (-), and underscores (\_).

### 3. Click **Subscribe to Amazon SNS topic**

test-queue [Edit](#) [Delete](#) [Purge](#) [Send and receive messages](#)

**Details** [Info](#)

Name	Type	ARN
test-queue	Standard	arn:aws:sqs:us-east-1:168437010507:test-queue
Encryption	URL	Dead-letter queue
-	https://sqs.us-east-1.amazonaws.com/168437010507/test-queue	-

[More](#)

[SNS subscriptions](#) [Lambda triggers](#) [Dead-letter queue](#) [Monitoring](#) [Tagging](#) [Access policy](#) [Encryption](#)

**SNS subscriptions (0)** [Info](#) [Refresh](#) [View in SNS](#) [Delete](#) [Subscribe to Amazon SNS topic](#)

### 4. Click Specific Amazon topic and save it

### Subscribe to Amazon SNS topic [Info](#)

**Amazon SNS topic**  
To allow your queue to receive messages from an Amazon SNS topic, subscribe it to an Amazon SNS topic.

Specify an Amazon SNS topic available for this queue.

Choose a topic

Enter Amazon SNS topic ARN

Use existing resource

arn:aws:sns:us-east-2:168437010507:aws-cloudtrail-logs

arn:aws:sns:us-east-2:168437010507:config-topic

[Cancel](#) [Save](#)

### 5. Click queues in navigation pane and get below image:

**Queues (1)** [Refresh](#) [Edit](#) [Delete](#) [Send and receive messages](#) [Actions](#) [Create queue](#)

	Name	Type	Created	Messages available	Messages in flight	Encryption	Content-based deduplication
<input type="radio"/>	test-queue	Standard	02/03/2021, 16:30:06 GMT+5:30	0	0	-	-



## 5.Splunk Installation:

1.I have Launched 6 EC2 Instance in Linux Platform(Ubuntu 20.04)

( Storage Minimum = 15 GB,

(\*) Heavy Forwarder Instance type ---> t2.medium

(\*) Search Head Instance type -----> t2.small

(\*) All others can be ----> t2.micro

<input type="checkbox"/>	Name ▼	Instance ID	Instance state ▼	Instance type
<input type="checkbox"/>	Heavy-Forwar...	i-0ce71fe2353fca174	✓ Running 🔍🔍	t2.medium
<input type="checkbox"/>	peer-node-1	i-05210d22dea7dbbf3	✓ Running 🔍🔍	t2.micro
<input type="checkbox"/>	peer-node-2	i-00675573af38330d9	✓ Running 🔍🔍	t2.micro
<input type="checkbox"/>	master-node	i-0a5f43e0d9609ca84	✓ Running 🔍🔍	t2.micro
<input type="checkbox"/>	Search-Head	i-06a04941f91e52cee	✓ Running 🔍🔍	t2.small
<input type="checkbox"/>	peer-node-3	i-0bf11b1b73406e921	✓ Running 🔍🔍	t2.micro

## 2.Install splunk enterprise software

1. Using SSH protocol to Connect EC2 Instance.

```
dhana@dhana-Ubuntu:~$ ssh ubuntu@54.157.183.227 -i Downloads/splunk.pem
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-1038-aws x86_64)
```

2.Go to the /opt Directory:

```
ubuntu@ip-172-31-63-211:~$ cd /opt/
ubuntu@ip-172-31-63-211:/opt$
```

3.Download splunk Software using wget Command and Press enter:

```
ubuntu@ip-172-31-63-211:/opt$ sudo su
root@ip-172-31-63-211:/opt# wget -O splunk-8.1.2-545206cc9f70-Linux-x86_64.tgz
'https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86_64&
platform=linux&version=8.1.2&product=splunk&filename=splunk-8.1.2-545206cc9f70-
Linux-x86_64.tgz&wget=true'
```

4.Extract the archived file using tar command:

```
root@ip-172-31-63-211:/opt# tar -xf splunk-8.1.2-545206cc9f70-Linux-x86_64.tgz
root@ip-172-31-63-211:/opt#
```

5. Change Directory to splunk/bin then Start and accept the license:  
(At the time prompt the username and password you can enter it)

For example:

username=admin

Password = development

```
root@ip-172-31-63-211:/opt# cd splunk/bin/
root@ip-172-31-63-211:/opt/splunk/bin# ./splunk start --accept-license
```

6. Splunk Essential Port setup in Security group :

Management port = 8089

Web port = 8000

Forwarding or receiving port = 9997

Replication port = 8080

SSH Port = 22

Inbound rules

Outbound rules

Tags

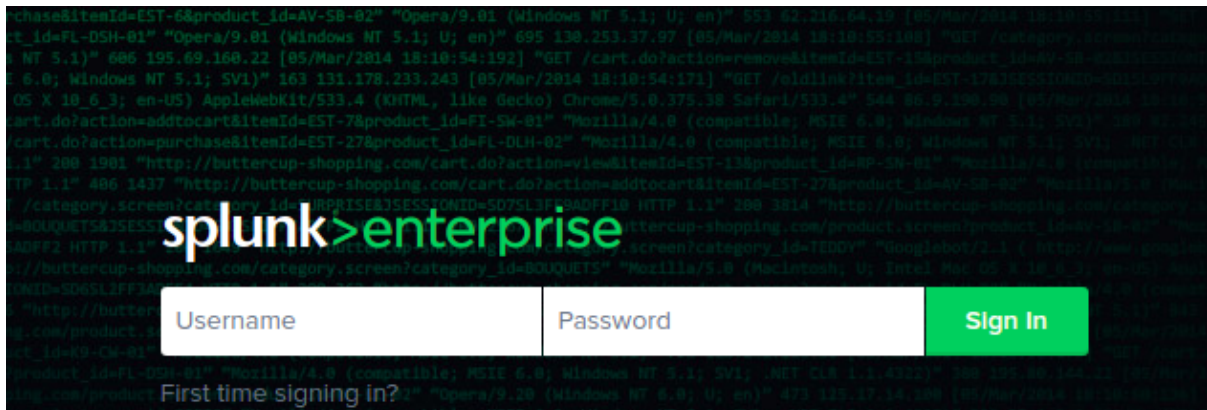
Inbound rules (5)

Edit inbound rules

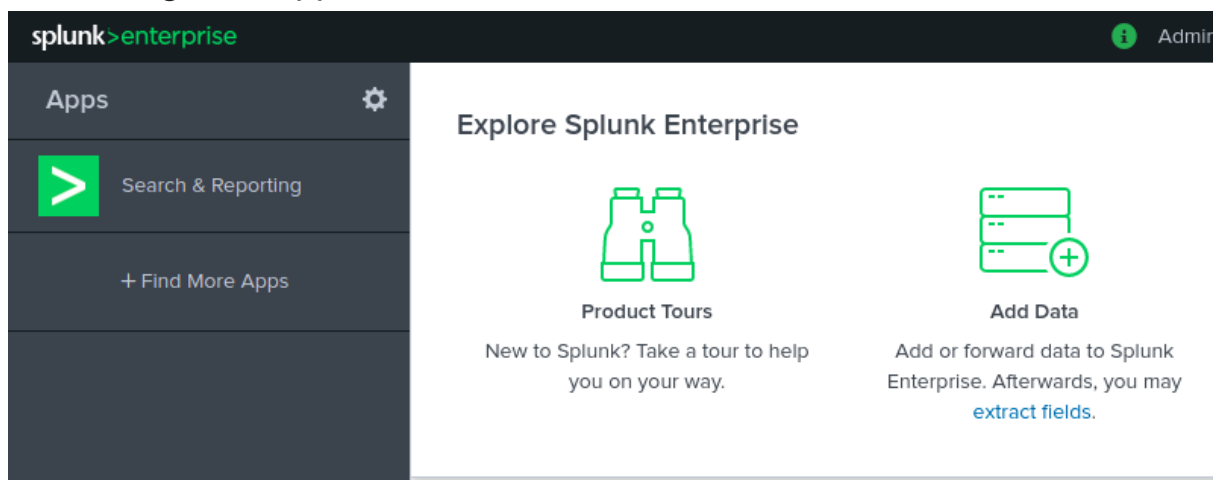
Type	Protocol	Port range	Source	Description - optional
Custom TCP	TCP	8080	0.0.0.0/0	–
Custom TCP	TCP	8000	0.0.0.0/0	–
SSH	TCP	22	103.5.112.131/32	–
Custom TCP	TCP	8089	0.0.0.0/0	–
Custom TCP	TCP	9997	0.0.0.0/0	–

## 6. Indexer Clustering Setup :

- . Enter Public IP and port 8000 in web browser tab and get below image

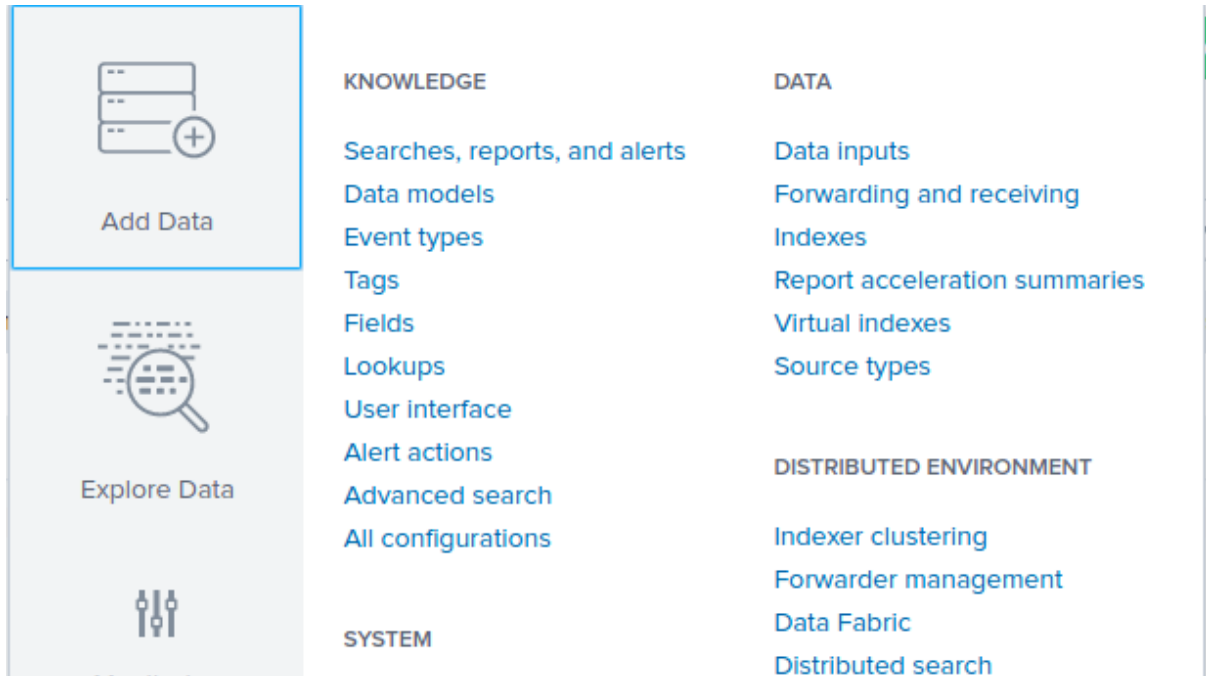


- .Enter user name and password then Sign to console and below image will appear:




- . **Configuring Master Node :**

Then we should go to settings - Indexer clustering -  
Enable Indexer clustering - (select)  
Master node - (click on) Next.



● .

## Indexer Clustering

Indexer Clusters are groups of Splunk Indexers configured to keep multiple copies of data. This increases data availability, data fidelity, data redundancy, and search performance. Indexer clustering is a complex feature, we recommend reading the documentation before enabling indexer clustering. [Learn More](#) 

[Enable Indexer clustering](#)

## Enable Clustering

☒ Master node

The master node coordinates the activities of the peer nodes. It does not store or replicate data (aside from its own internal data).

☐ Peer node

Peer nodes receive and index incoming data. They also replicate data from other nodes in the cluster.

☐ Search head node

The search head manages searches across one or more clusters.

Next

Cancel

- **After clicking on next it will ask you to set Replication Factor, Search Factor, Security Key, Cluster Label.**

## Master Node Configuration

Replication Factor

3

The number of copies of raw data that you want the cluster to maintain. A higher replication factor protects against loss of data if peer nodes fail.

Search Factor

2

The number of searchable copies of data the cluster maintains. A higher search factor speeds up the time to recover lost data at the cost of disk space. Must be less than or equal to Replication Factor.

Security Key

Optional

This key authenticates communication between the master and the peers and search heads.

Cluster Label

Optional

Name your cluster using this field. This label is also used to identify this cluster in the Monitoring Console.

Back

Enable Master Node

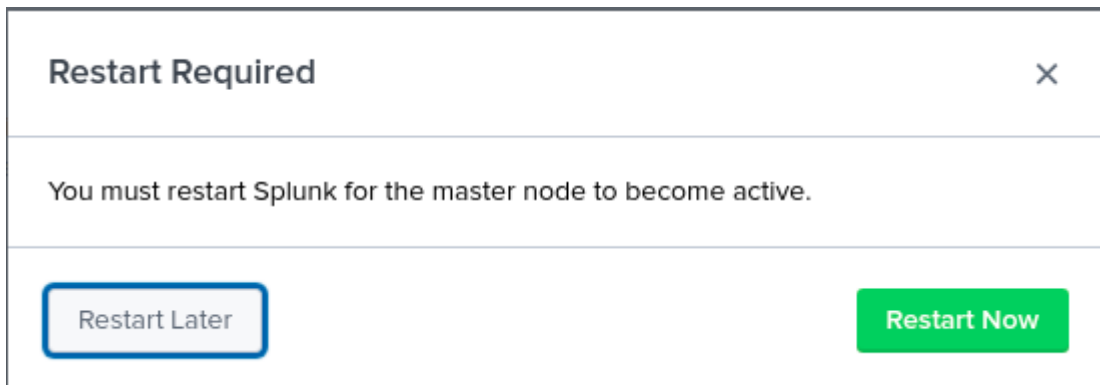
- **Replication Factor - Keep the Replication Factor as 3 which is by default**

**NOTE : Indexers (peer nodes) should not be less than Replication Factor value.**

- **Search Factor - Keep the Search Factor as 2 which is by default.**

- **NOTE : The password which you enter should be same for setting up the peer nodes.**

- **After clicking on Enable Master Node the Splunk has to be restarted.**



- **Now Connect to Terminal using SSH and login as root user in terminal.**

```
[root@ip-172-31-39-12 local]# cd /opt/splunk/etc/master-apps/_cluster/local
[root@ip-172-31-39-12 local]# vi indexes.conf
```

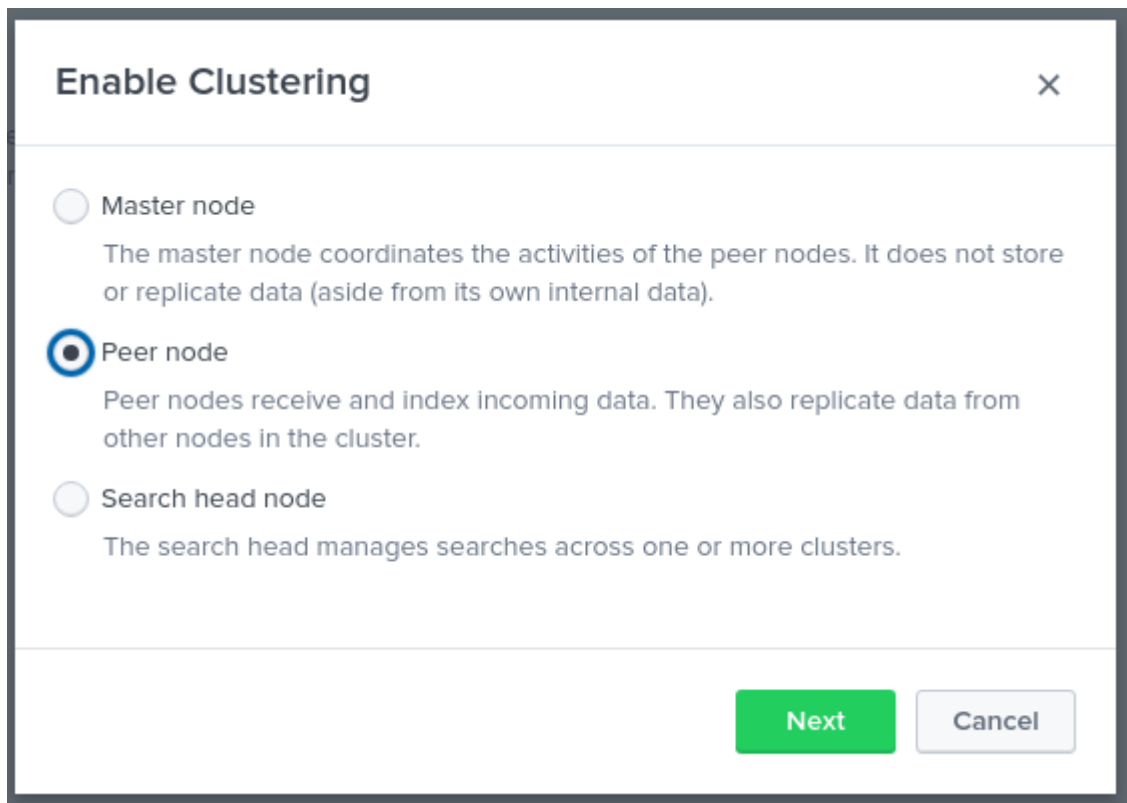
- Now insert the below text in indexes.conf file

```
[aws_index]
repFactor = auto
homePath  = $SPLUNK_DB/aws_index/db
coldPath  = $SPLUNK_DB/aws_index/colddb
thawedPath = $SPLUNK_DB/aws_index/thaweddb
```

- After pasting the above text SAVE the file and Restart the server.

## Connecting Peer Nodes To Cluster:

- Now for connecting peer nodes we should login to the Indexer Instance.
- Then we should go to settings - Indexer clustering - Enable Indexer clustering - (select) Peer node -(click on) Next.



The screenshot shows a dialog box titled "Enable Clustering" with a close button (X) in the top right corner. Inside the dialog, there are three radio button options, each with a description:

- ☐ Master node  
The master node coordinates the activities of the peer nodes. It does not store or replicate data (aside from its own internal data).
- ☒ Peer node  
Peer nodes receive and index incoming data. They also replicate data from other nodes in the cluster.
- ☐ Search head node  
The search head manages searches across one or more clusters.

At the bottom right of the dialog, there are two buttons: a green "Next" button and a grey "Cancel" button.

- After clicking on next it will ask you to set Master URI, Peer replication port, Security key.



Peer node configuration

Master URI

https://18.223.134.4:8089

E.g. https://10.152.31.202:8089

Peer replication port

8080

The port peer nodes use to stream data to each other (Eg: 8080).

Security key

.....|

This key authenticates communication between the master and the peers and search heads.

Back

Enable peer node

- NOTE : Use same Security Key which we used for configuring Master Node.
- After clicking on Enable Peer Node the Splunk has to be restarted.

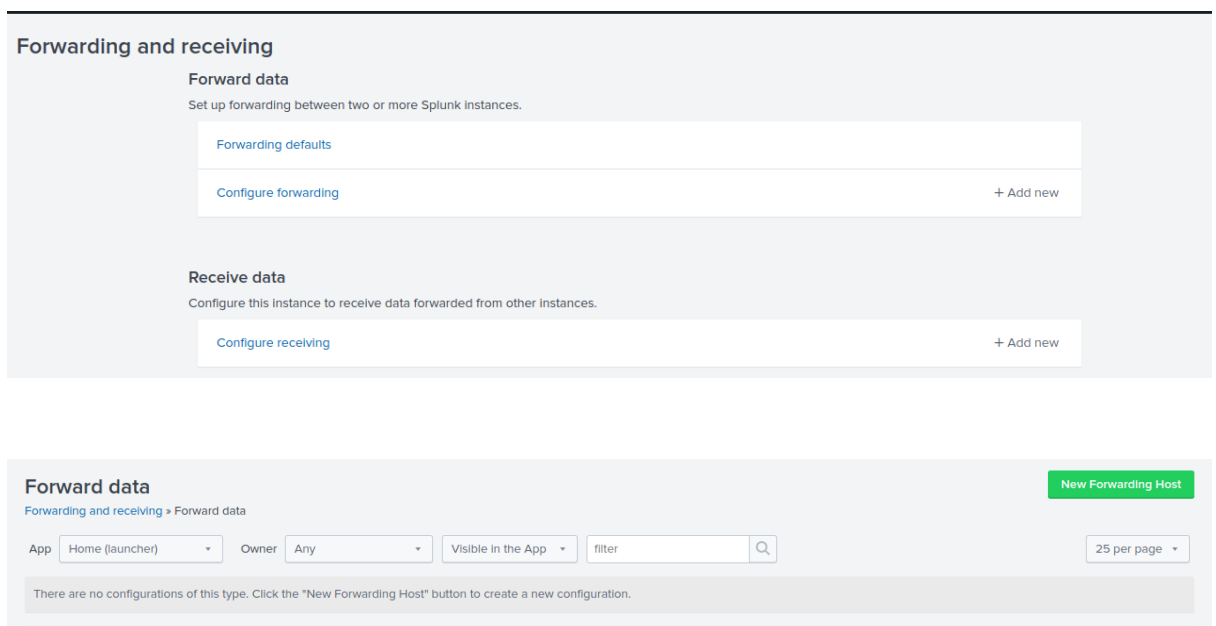
Restart Later

Restart Now

- Likewise connect other 2 (Indexer's) peer nodes to cluster.

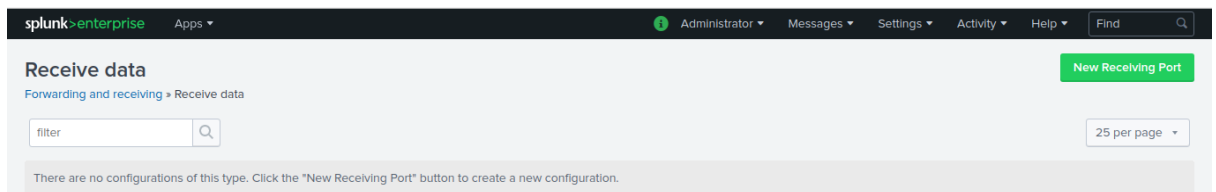
# Heavy Forwarder Configuration

- Now for connecting peer nodes we should login to the Heavy Forwarder Instance.
- Then we should go to settings - Forwarding and Receiving - Configure forwarding - New Forwarding Host.



- After clicking on New Forwarding Host it will ask you to set Host. Host = Host IP:9997
- NOTE : Host IP is nothing but Indexer IP

- After entering Host IP:9997 click on Save and Restart the Splunk.
  - Likewise connect host for other 2 Indexers.
  - Now Connect to Terminal using and login as root user in terminal
- 
- Now come to Peer Nodes GUI and go to settings - Forwarding and Receiving - Configure Receiving - New Receiving Port.



- Likewise set port for other 2 Indexers.

## Connecting Search Head to Cluster:

Then we should go to settings - Indexer clustering - Enable Indexer clustering - (select) Search Head node - (click on) Next.

### Enable Clustering

Master node

The master node coordinates the activities of the peer nodes. It does not store or replicate data (aside from its own internal data).

Peer node

Peer nodes receive and index incoming data. They also replicate data from other nodes in the cluster.

☒

Search head node

The search head manages searches across one or more clusters.

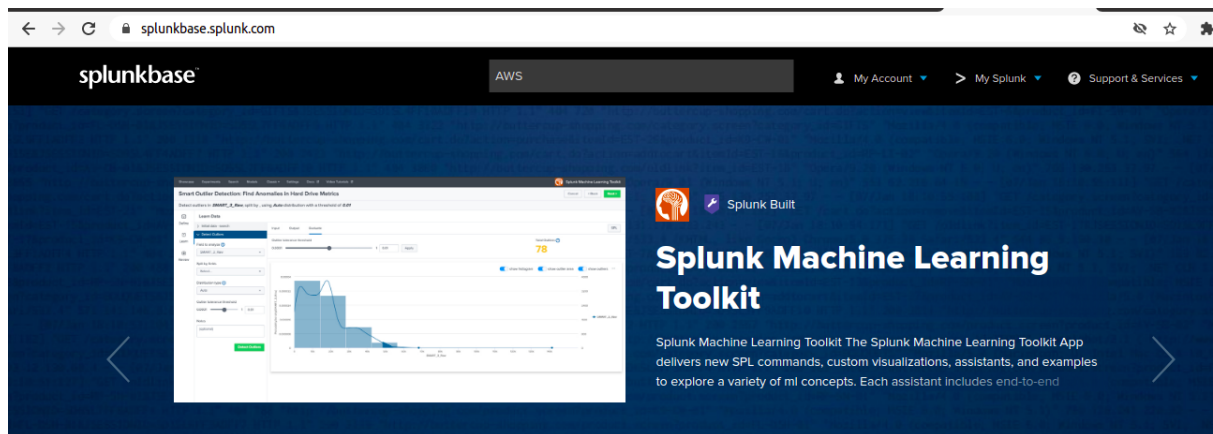
Next

Cancel

- Master IP = Master Node IP
- Security Key : Use same Security Key which we used for configuring Master Node.
- After clicking on Enable Search Head Node the Splunk has to be restarted.

## 7.AWS add-on and AWS app Download:

(\*) Download AWS add-on and AWS app in **splunkbase.com**



(\*) Search AWS add-on in search bar and download it



### **Splunk Add-on for AWS Network**

14 Installs



### **Event Push by Deductiv**

28 Installs



### **Splunk Add-on for Amazon Web**

12784 Installs



### **Splunk Add-on for Amazon Kinesis**

7982 Installs




(\*) Search AWS app in search bar and download it:

# App Search Results


PRODUCTS & SOLUTIONS	>
CATEGORIES	>
TECHNOLOGIES	>
APP TYPE	>
APP CONTENTS	>
SPLUNK VERSION	>

Search: aws app X

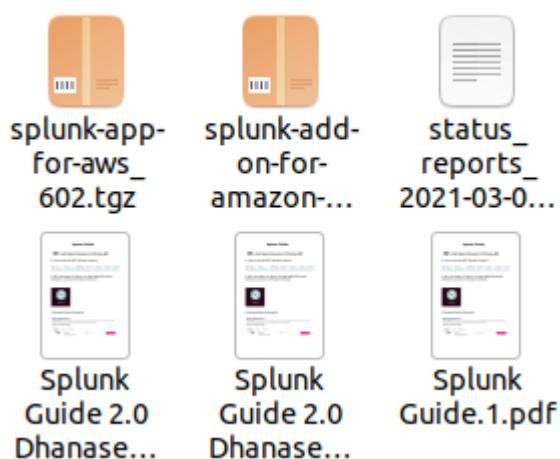
Showing 1-20 of 58 results



**Splunk App for AWS**

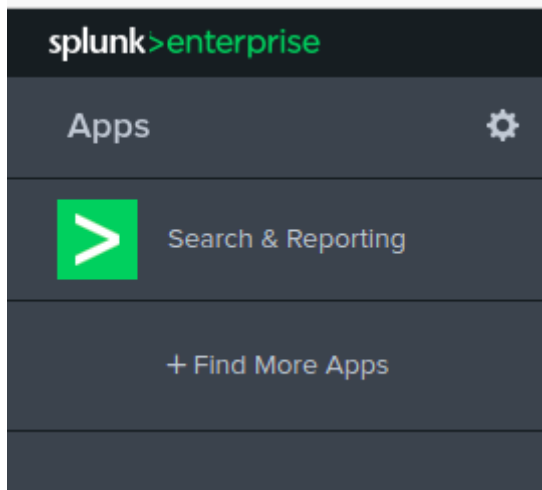
7516 Installs 

(\*)Now Downloaded in AWS add-on and AWS app:



## 7.Install the AWS Add-on In Heavy Forwarder :

1)Click Manage Apps Icon and get below image

A screenshot of the Splunk Apps management page. The top navigation bar includes 'splunk>enterprise', 'Apps', and various user and system links. The main section is titled 'Apps' and shows a list of installed and available apps. A table lists the following apps: SplunkForwarder, SplunkLightForwarder, Log Event Alert Action, and Webhook Alert Action. Each row includes details like folder name, version, update checking status, visibility, sharing, and status, along with links to edit properties or view objects.

Name	Folder name	Version	Update checking	Visible	Sharing	Status	Actions
SplunkForwarder	SplunkForwarder		Yes	No	App   <a href="#">Permissions</a>	Disabled   <a href="#">Enable</a>	
SplunkLightForwarder	SplunkLightForwarder		Yes	No	App   <a href="#">Permissions</a>	Disabled   <a href="#">Enable</a>	
Log Event Alert Action	alert_logevent	8.1.2	Yes	No	App   <a href="#">Permissions</a>	Enabled   <a href="#">Disable</a>	<a href="#">Edit properties</a>   <a href="#">View objects</a>
Webhook Alert Action	alert_webhook	8.1.2	Yes	No	App   <a href="#">Permissions</a>	Enabled   <a href="#">Disable</a>	<a href="#">Edit properties</a>   <a href="#">View objects</a>

(2)

### Upload an app

If you have a .spl or .tar.gz app file to install, you can upload it using this form.

You can replace an existing app via the Splunk CLI. [Learn more.](#)

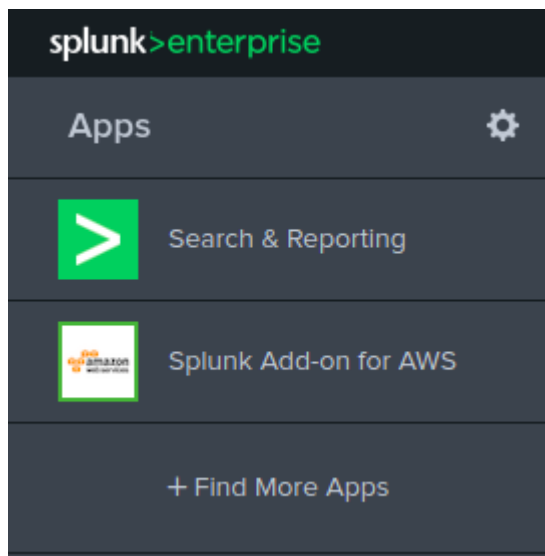
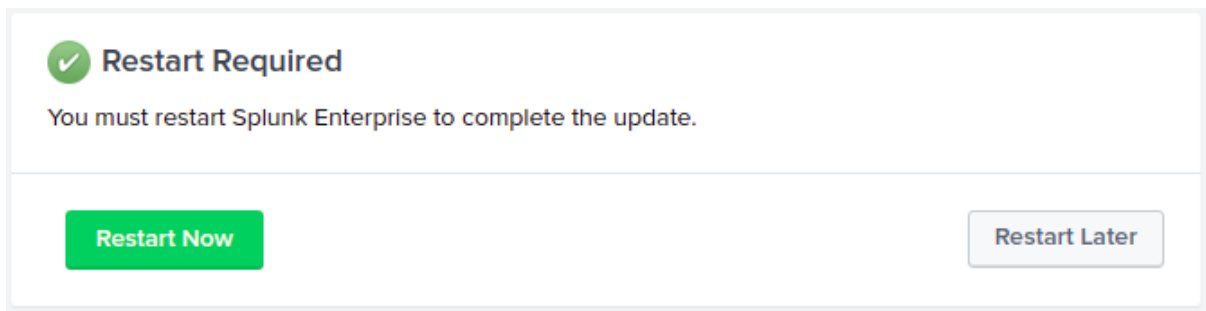
File

splunk-add-on-f...rvices\_503.tgz

☐ Upgrade app. Checking this will overwrite the app if it already exists.

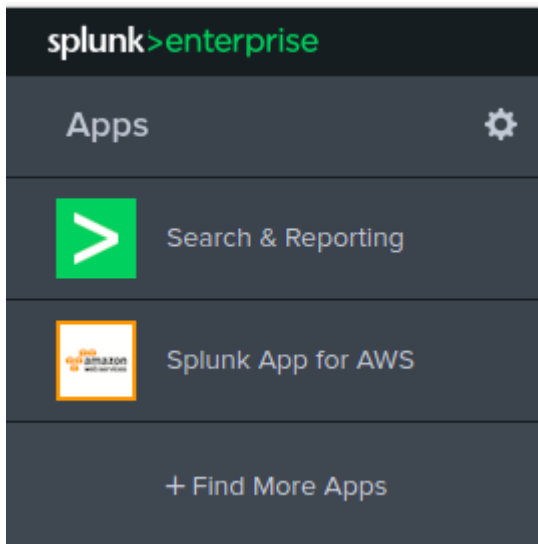


(3)Then click restart icon and below image will appear



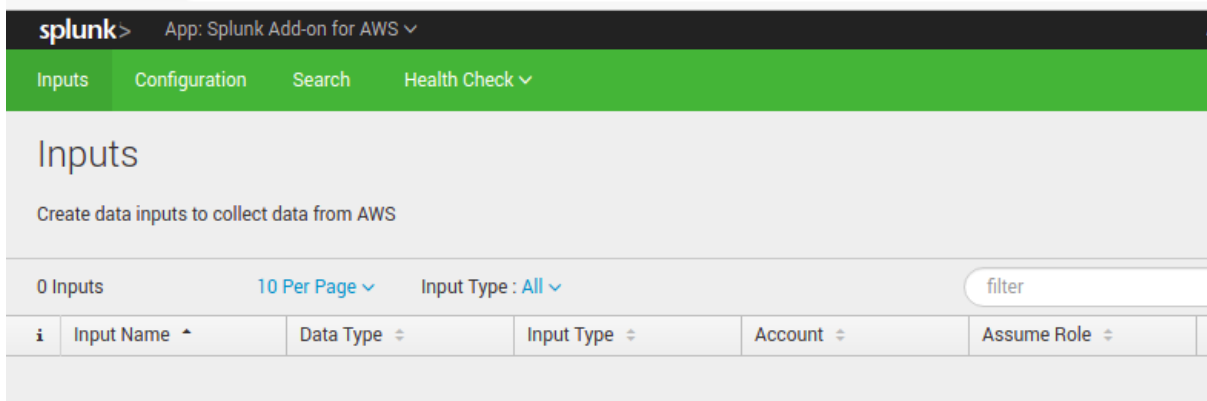
## 8.Install **AWS** app in **Search Head** :

1)Click Manage Apps Icon and upload the AWS app zip file  
Then click restart icon and below image will appear



## 9. Configure AWS add-on :

1) click splunk add-on for aws and below image will appear



(2) first click **configuration** then Click **Add Icon**

Update Account

Name

user1

Key ID

AKIAIVFA6XBLFHAUIHSQ

Secret Key

.....

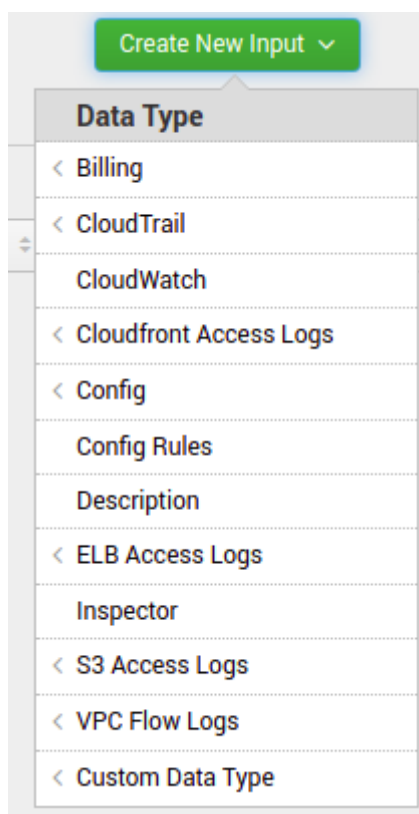
Region Category

Global

Cancel

Update

3)Click **Inputs** then Click **Create new input Icon** and **Click Description**



4)Enter **AWS Input Configuration** and Save it

## AWS Input Configuration [Learn more](#)

Name

AWS Account

Assume Role

AWS Regions

APIs/Interval (seconds)	API	Interval (in seconds)
<input checked="" type="checkbox"/>	ec2_volumes	<input type="text" value="3600"/>
<input checked="" type="checkbox"/>	ec2_instances	<input type="text" value="3600"/>
<input checked="" type="checkbox"/>	ec2_reserved_instances	<input type="text" value="3600"/>
<input checked="" type="checkbox"/>	ebs_snapshots	<input type="text" value="3600"/>
<input checked="" type="checkbox"/>	classic_load_balancers	<input type="text" value="3600"/>
<input checked="" type="checkbox"/>	application load balancers	<input type="text" value="3600"/>

☒ iam\_users

## Splunk-related Configuration

Source Type

Index

Cancel

Save

## 5.Finally Following input will give following input:

i	Input Name ^	Data Type ^	Input Type ^	Account ^
>	aws	Description	Description	aws-config
>	cloud-watch	CloudWatch	CloudWatch	aws-config
>	cloudtrail	CloudTrail	CloudTrail	aws-config
>	config	Config	Config	aws-config

## 10. Configuration Bundle: Validate and Push

- Now login to Mastor Node web interface and Edit Configuration Bundle Actions to Click Validate and Check Restart and push

### Configuration Bundle Actions

Click Push to distribute the configuration bundle to the set of peers. Optionally, validate the bundle and check if peer restart is required without distributing the bundle, or rollback to the previous bundle. [Learn More](#)

[Back to Master Node](#)

[Validate and Check Restart](#) [Push](#) [Rollback](#)

[Documentation](#)

Last Push: ✓ Successful

Updated Time ..... 22/01/2021, 10:49:17

Active Bundle ID ? ..... 2B1C0DEDB72F09C4A6ED02BC5026D13B

Latest Bundle ID ? ..... 2B1C0DEDB72F09C4A6ED02BC5026D13B

Previous Bundle ID ? ..... 50C325AD39E94AC324538EE3544F8D97

10 per page ▼

i	Peer ↕	Site	Status	Action Status
>	ip-172-31-1-161.us-east-2.compute.internal	default	Up	None
>	ip-172-31-30-129.us-east-2.compute.internal	default	Up	None
>	ip-172-31-42-182.us-east-2.compute.internal	default	Up	None

Then automatically creates **aws\_index** on PEER Nodes.See the below Image

<a href="#">_thefishbucket</a>	<a href="#">Edit</a>	<a href="#">Delete</a>	<a href="#">Disable</a>	<a href="#">Events</a>	<a href="#">_cluster</a>	1 MB	488.28 GB	0
<a href="#">aws_index</a>	<a href="#">Edit</a>	<a href="#">Delete</a>	<a href="#">Disable</a>	<a href="#">Events</a>	<a href="#">_cluster</a>	11 MB	488.28 GB	65.8K

Now go to one of the Search head GUI and search for index="aws\_index".  
You get information similar to below image

## New Search

index=aws\_index

✓ **79,274 events** (14/03/2021 17:00:00.000 to 15/03/2021 17:31:46.000) No Event Sampling ▼

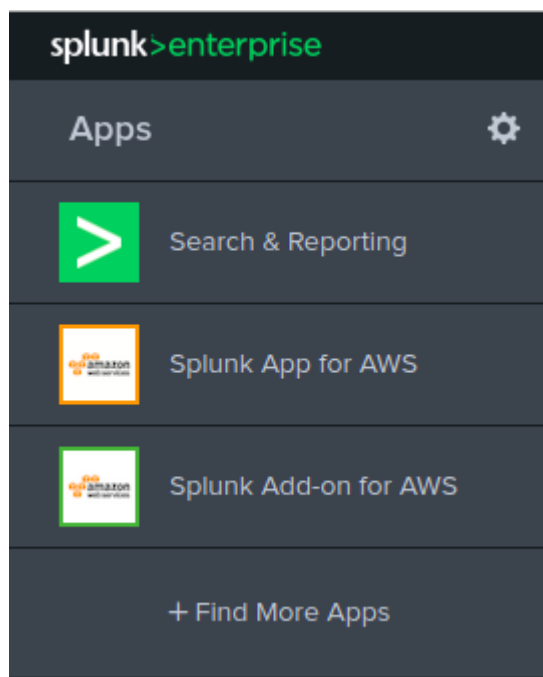
Events (79,274) Patterns Statistics Visualization

Format Timeline ▼ — Zoom Out + Zoom to Selection × Deselect


List ▼ / Format 50 Per Page ▼

< Hide Fields		≡ All Fields		i	Time	Event
SELECTED FIELDS				>	15/03/2021 17:00:00.000	03/15/2021 17:00:00 +0000.000, info_search_time=pe=aws_description, input host = ip-172-31-37-123   s
a host 2						
a source 11						
a sourcetyp 4						

### 11.Install **AWS** add-on in **Search Head** :



## 12. Using SSH Protocol to Connect search Head Instance and Change directory

cd /opt/splunk/etc/apps/splunk\_app\_aws/local/

```
root@ip-172-31-21-17:/home/ubuntu# cd /opt/splunk/etc/apps/splunk_app_aws/local
root@ip-172-31-21-17:/opt/splunk/etc/apps/splunk_app_aws/local#
```

### Create Outputs.conf

```
root@ip-172-31-21-17:/opt/splunk/etc/apps/splunk_app_aws/local# vi outputs.conf
```

### Enter the following Content :

```
[indexAndForward]
index = false # Turn off indexing on the search head
[tcput]
defaultGroup = my_search_peers # Name of the search peer group
forwardedindex.filter.disable = true
indexAndForward = false
[tcput:my_search_peers]
server=3.138.142.225:9997,3.23.95.41:9997,3.142.184.66:9997 #
list of peers
```

```
[indexAndForward]
index = false # Turn off indexing on the search head
[tcput]
defaultGroup = my_search_peers # Name of the search peer group
forwardedindex.filter.disable = true
indexAndForward = false
[tcput:my_search_peers]
server=3.138.142.225:9997,3.23.95.41:9997,3.142.184.66:9997 # list of peers
~
~
~
```

(List of peers means Peer nodes public IP )

### Restart the Splunk server

```
root@ip-172-31-21-17:~# cd /opt/splunk/bin/
root@ip-172-31-21-17:/opt/splunk/bin# ./splunk restart
```

## 5. Click **AWS app** and **Click Overview**

Now get below image will appear

