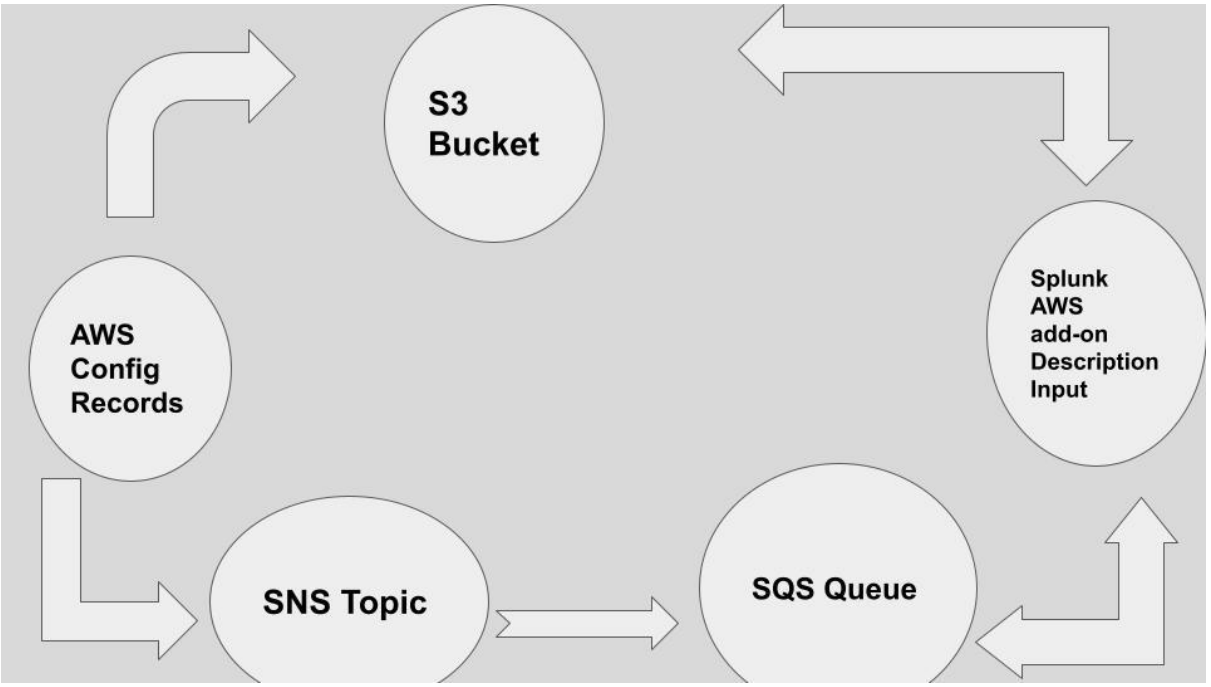
AWS Config-Splunk

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

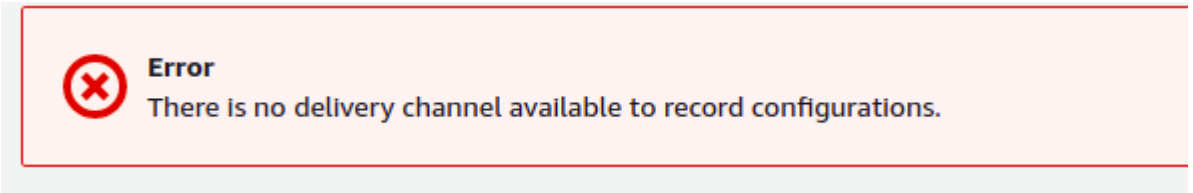


1.AWS Config:

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



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
○	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",
```

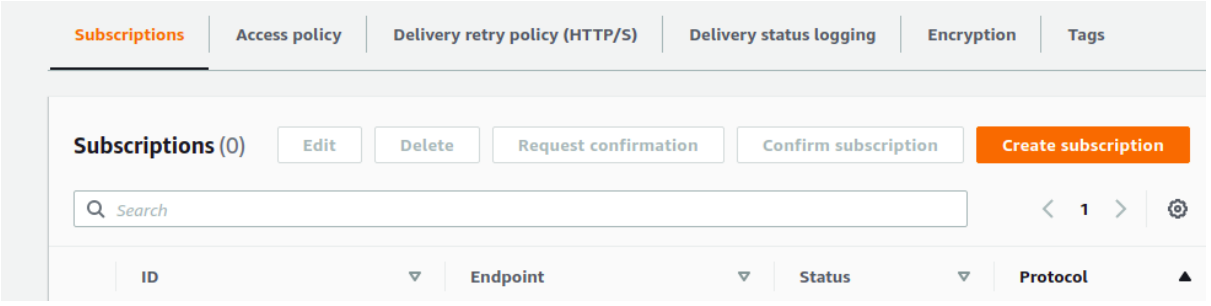
2.(b)Create the SNS topic:

1. Open the Amazon SNS console in the same Region as your AWS Config service, and then click Topics.

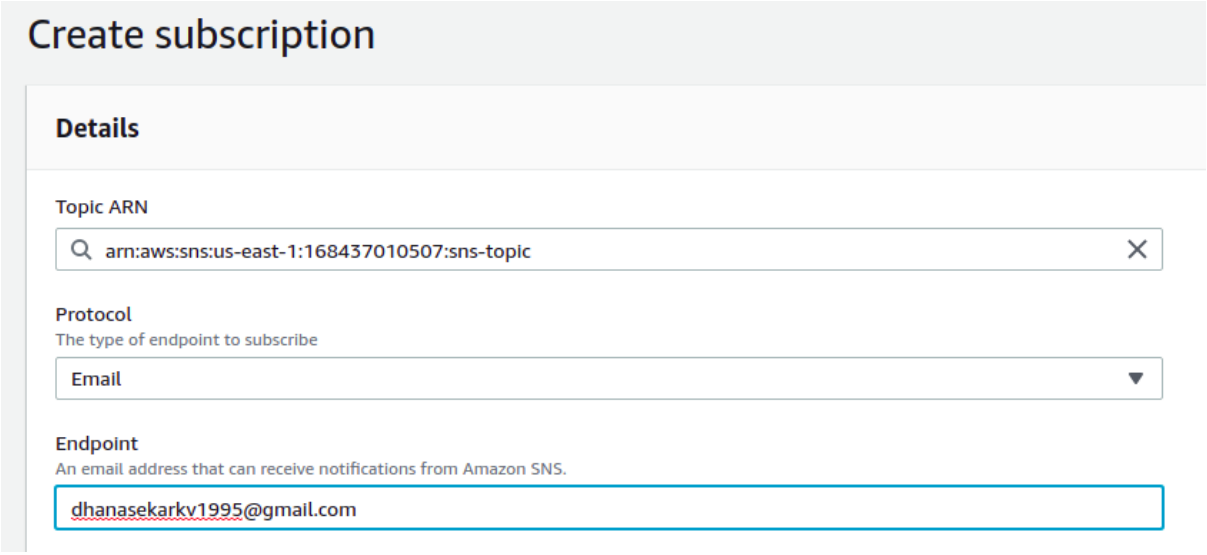
2. Click Create topic.

3. enter a name for SNS topic, and then click Create topic.

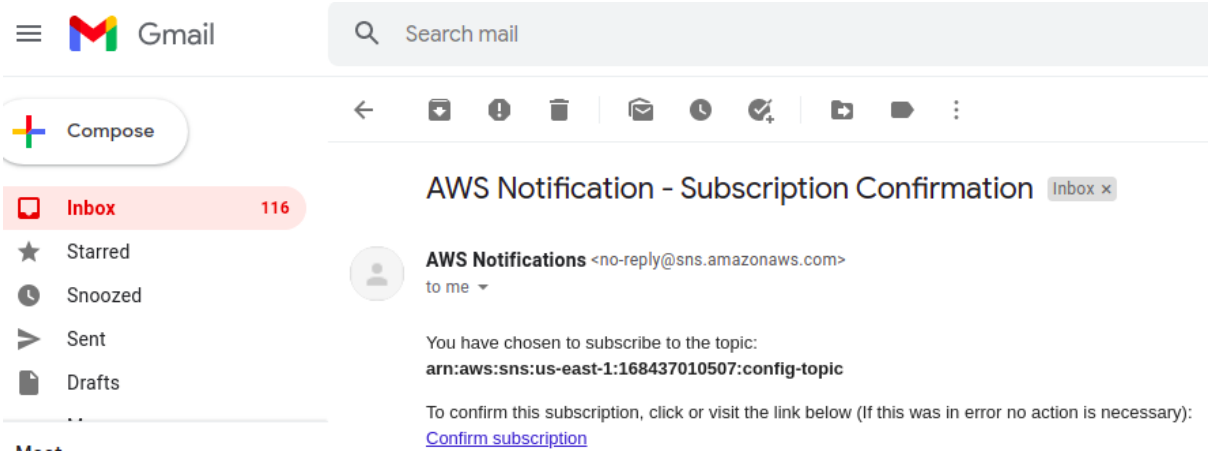
4.Click Create subscription.



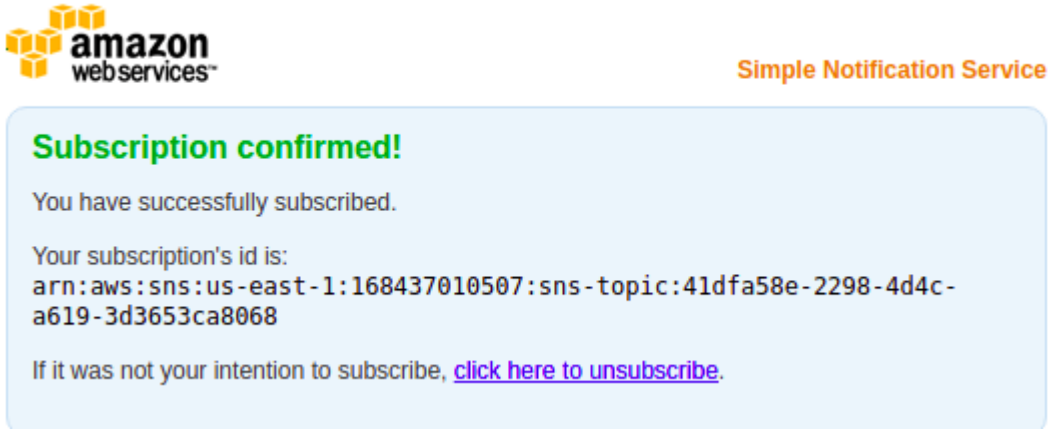
5. In Protocol, click Email and enter the email-address and click the create subscription



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



7.receive the message Subscription confirmed!



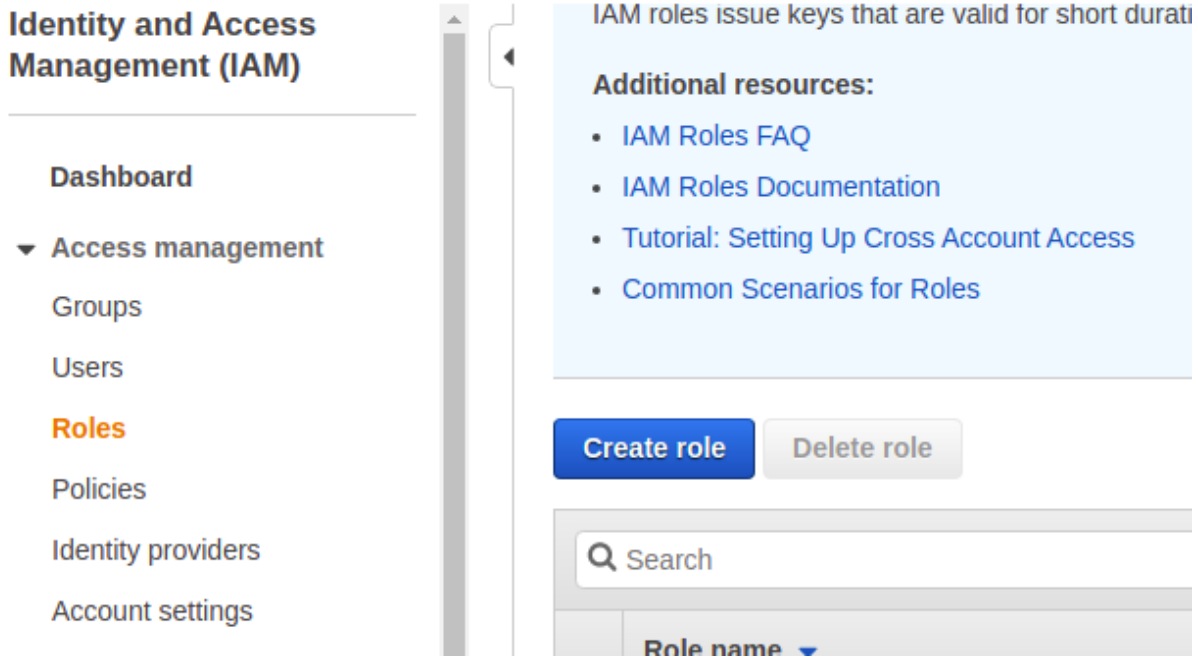
2.(c)Configure IAM Role:

Purpose:

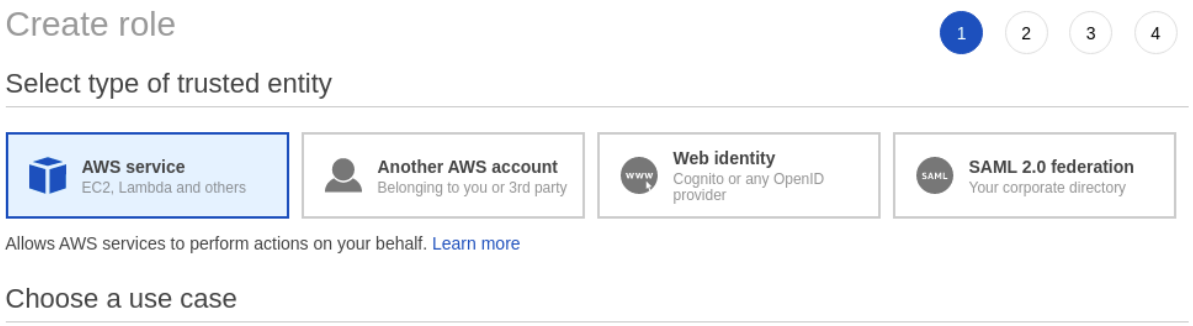
An IAM role is an IAM entity that defines a set of permissions for making AWS service requests. IAM roles are not associated with a specific user or group. Instead, trusted entities assume roles, such as IAM users, applications, or AWS services such as EC2.

1.Open the IAM console

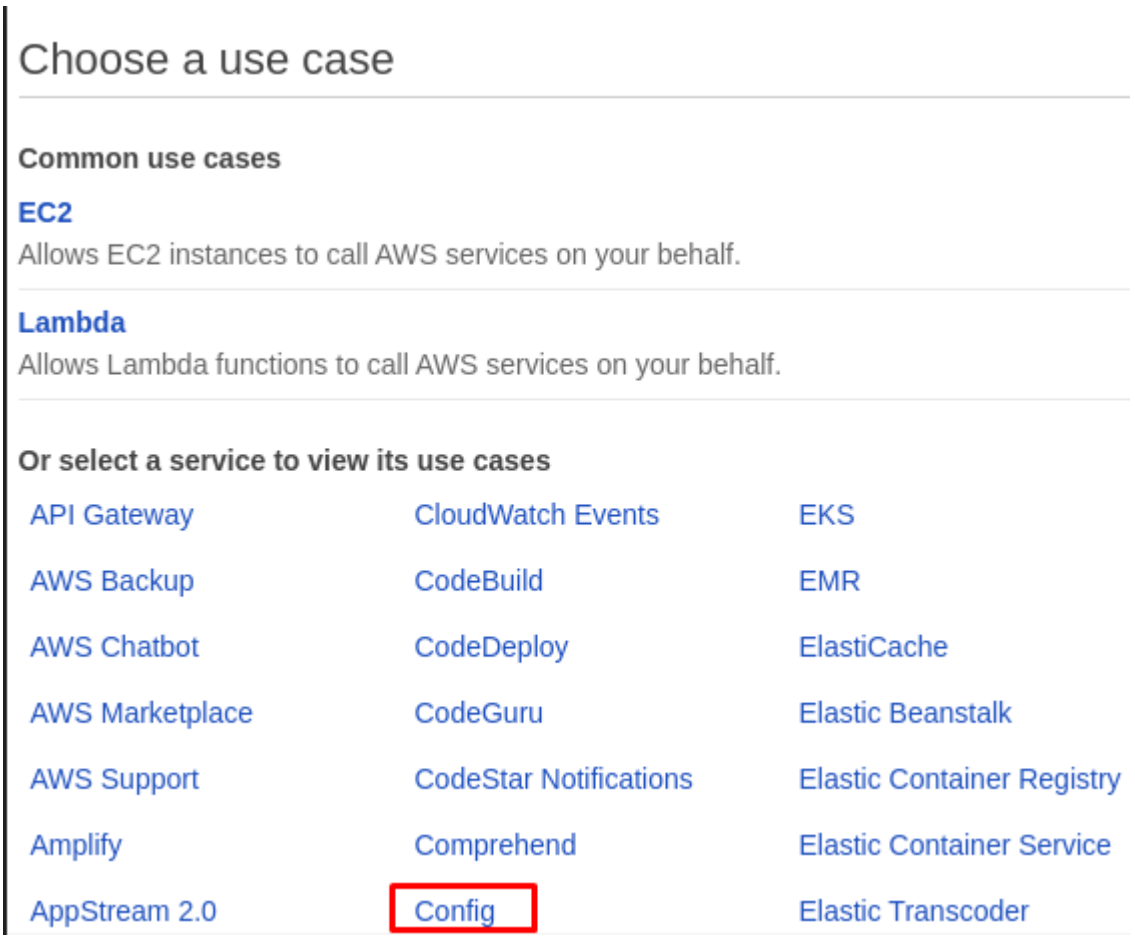
2.Choose Roles, and then choose Create role



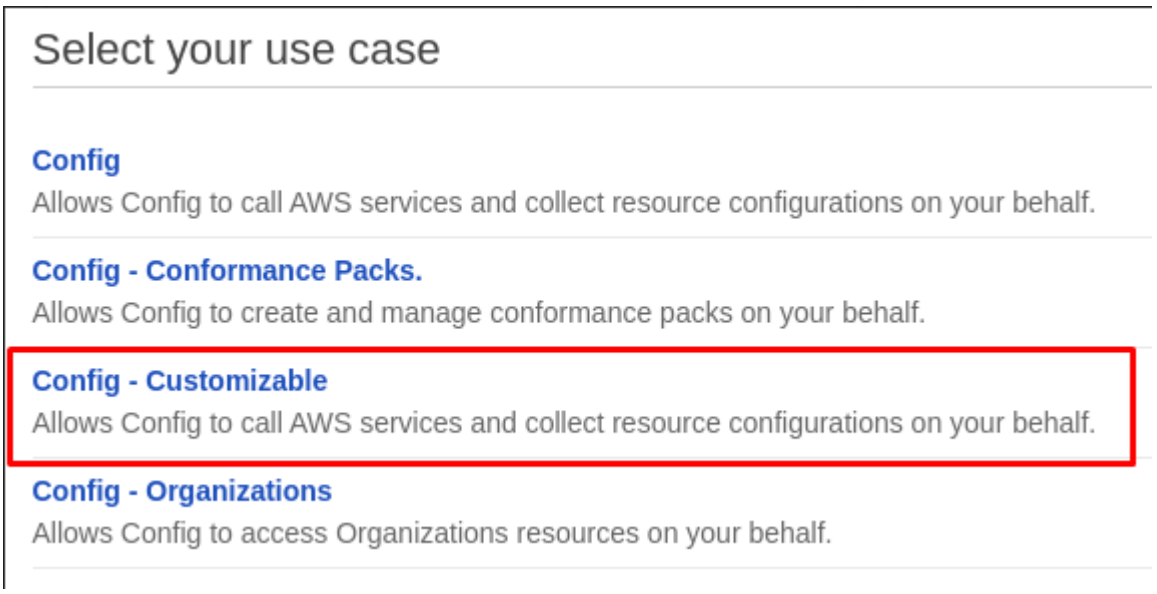
3.Select type of trusted entity,Choose AWS Service:



4.Choose a **Config**

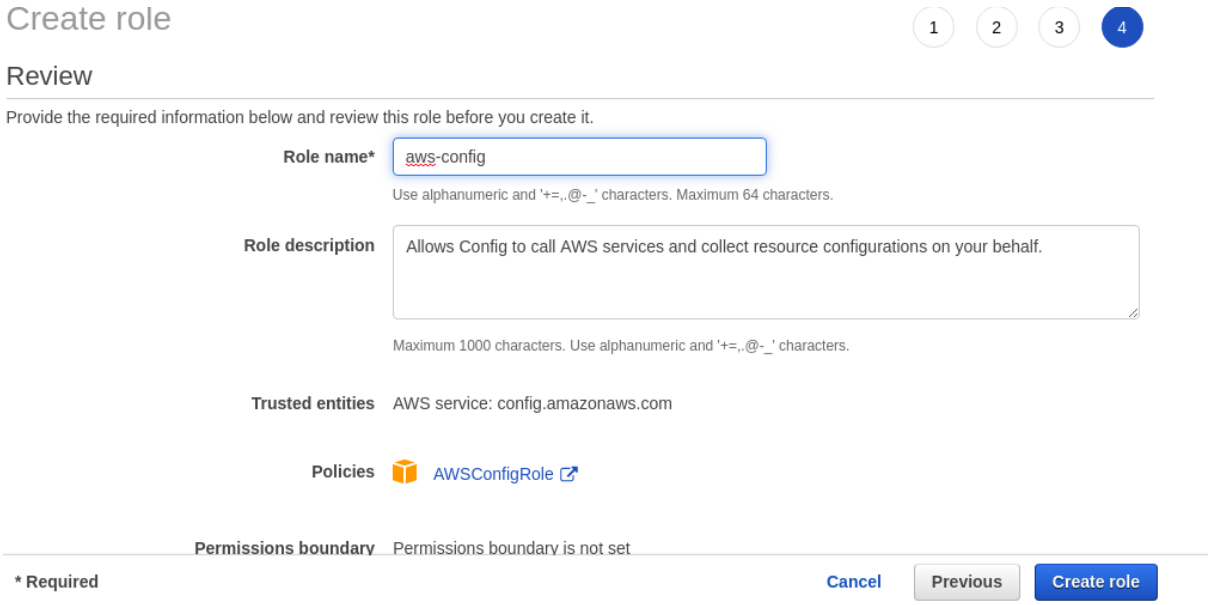


5.Choose-Customizable

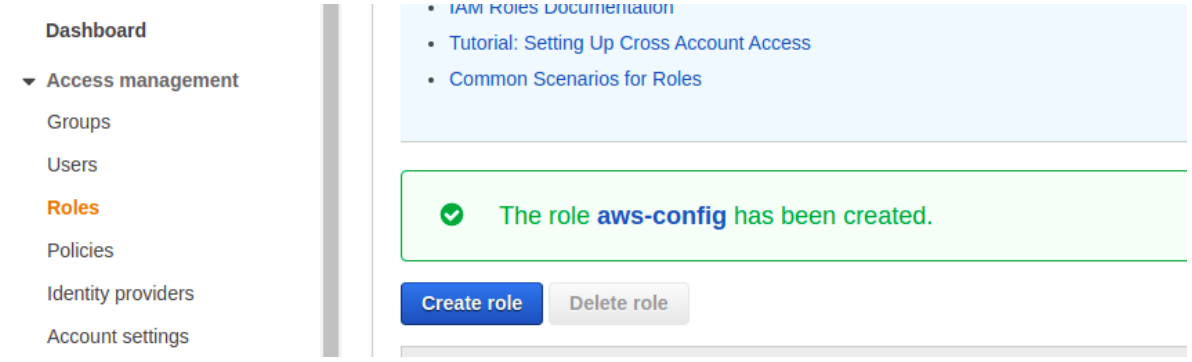


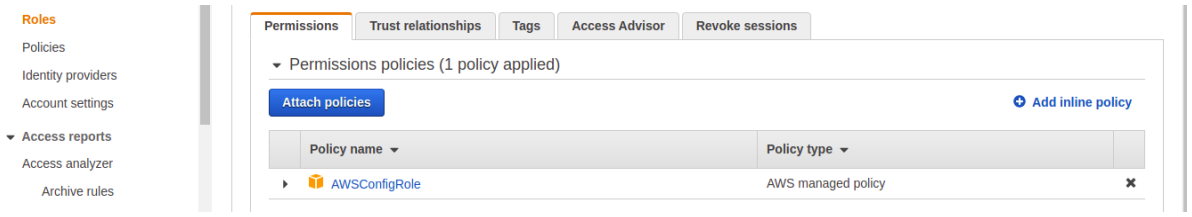
6.choose Next: Permissions Next: Tags, and then Next: Review

7.Enter Role name and Choose 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:

```
{
        "Version": "2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Action": [
              "s3:PutObject",
                  "s3:PutObjectAcl"
            ],
            "Resource": [
              "arn:aws:s3:::arn:aws:s3:::course-dhana/AWSLogs/168437010507/*"
            ],
```

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:
```

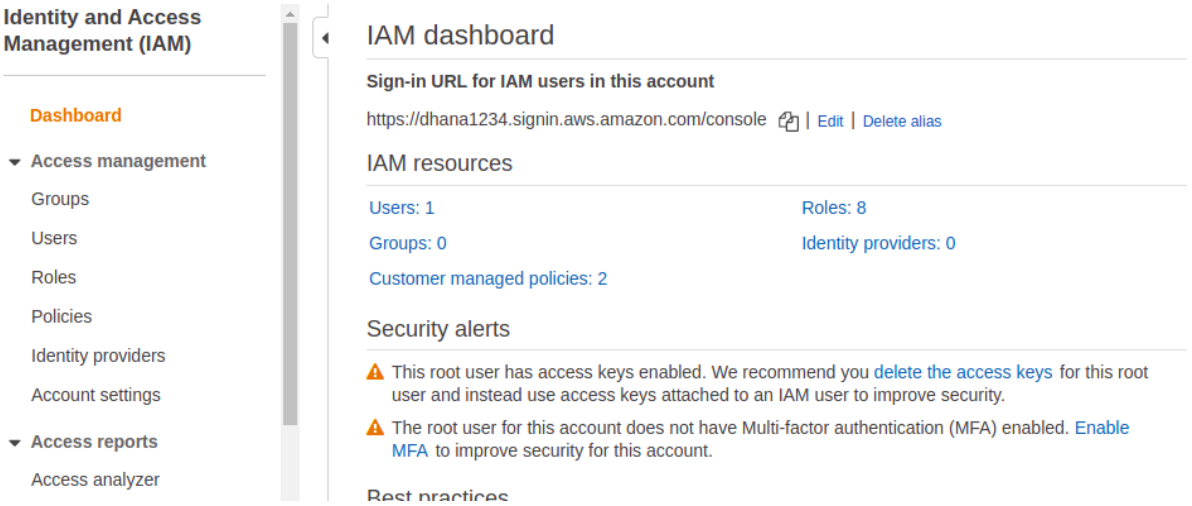
4.Confirm the installation:

aws --version

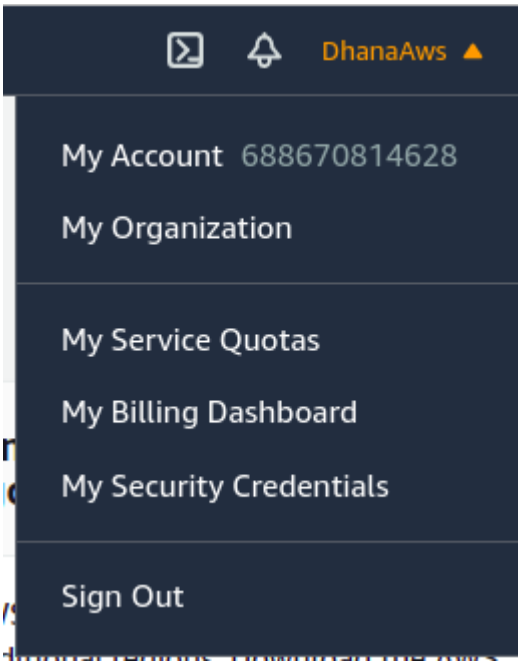
```
dhana@dhana-Ubuntu:~$ aws --version
aws-cli/2.1.28 Python/3.8.8 Linux/5.8.0-44-generic exe/x86_64.ubuntu.20 prompt/o
ff
```

5.Configured AWS CLI:

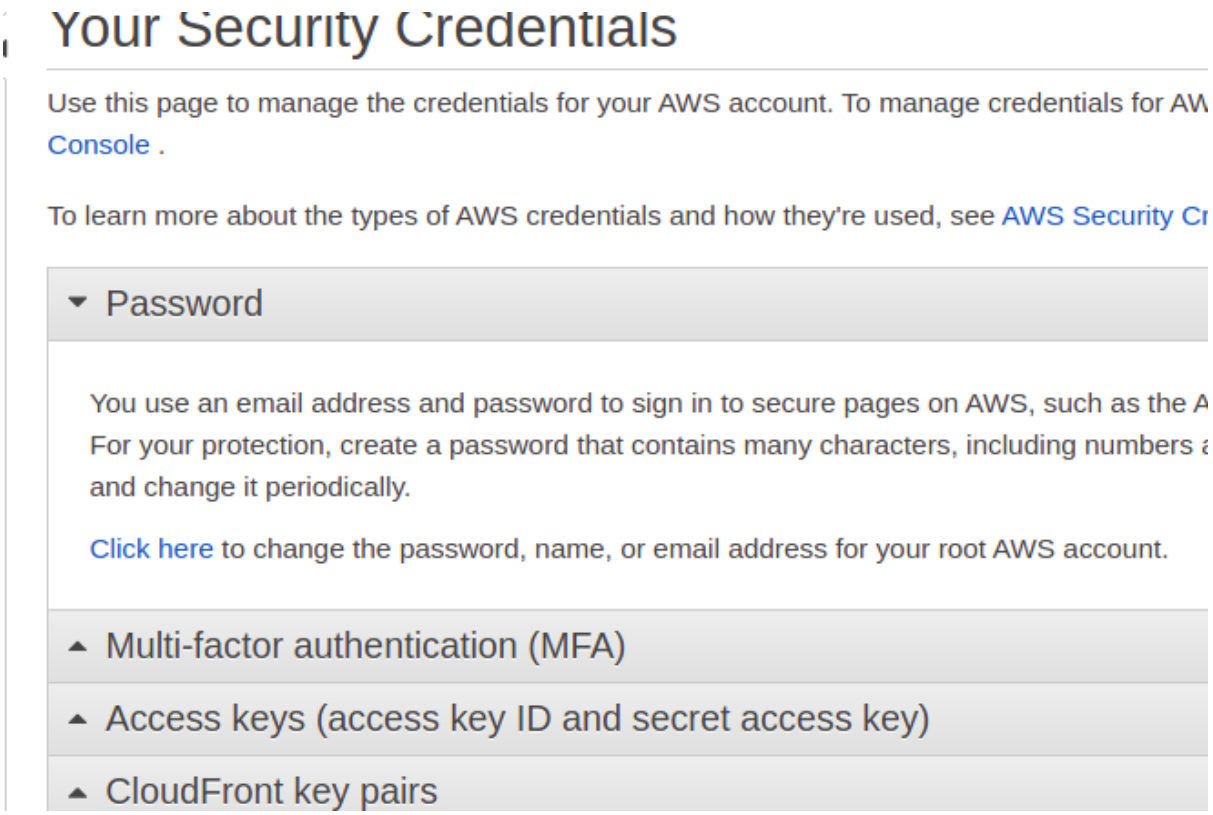
1)Go to the IAM Dashboard console



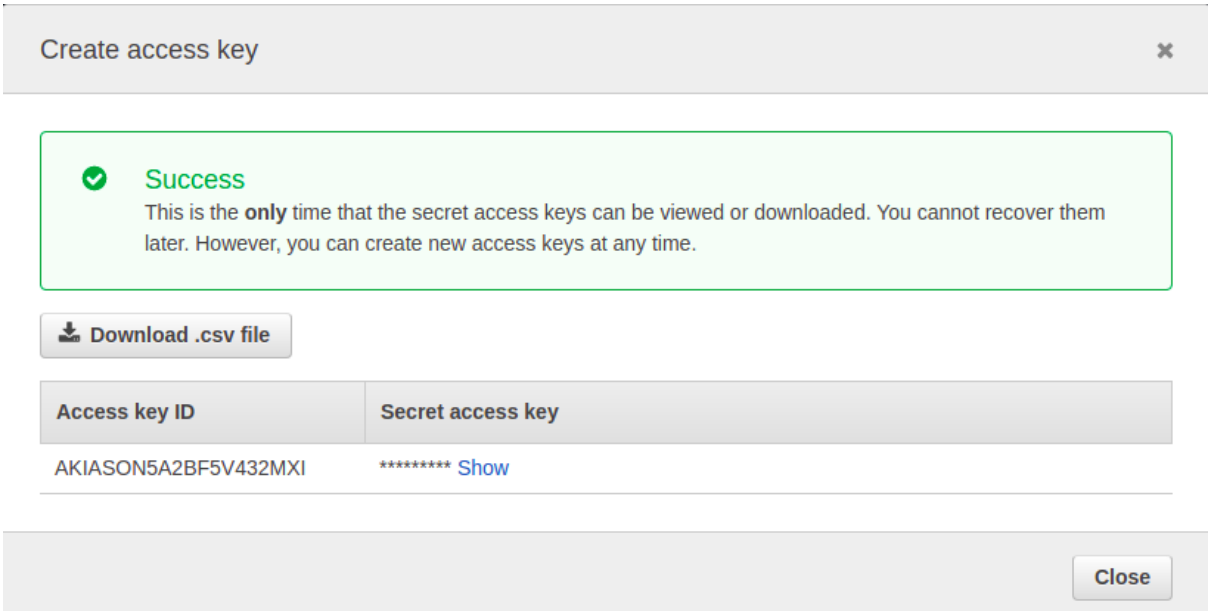
2)click username Icon and blow image will appear



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



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 **Dowload.csv file:**



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]: kqrdko7yoYHCxz3p1gEjB4zTlDIAQR6dLP
nck02B
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

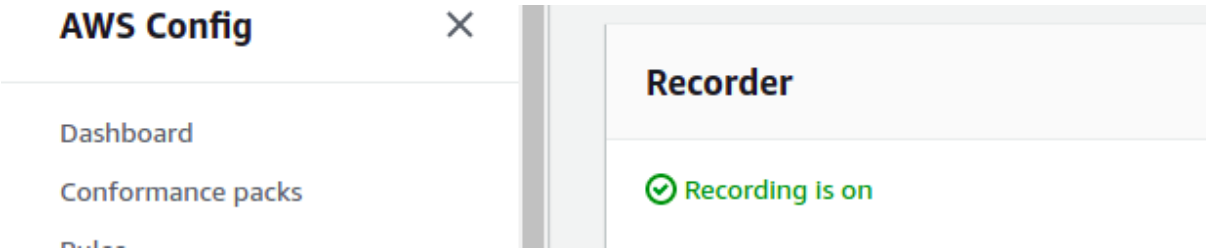
```
dhana@dhana-Ubuntu:~$ aws configservice describe-delivery-channels
{
    "DeliveryChannels": [
        {
            "name": "default",
            "s3BucketName": "course-dhana",
            "snsTopicARN": "arn:aws:sns:us-east-1:168437010507:config-topic",
            "configSnapshotDeliveryProperties":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

3.Start the configuration recorder:

1. Open the AWS Config Console:

2. In the navigation pane, choose Settings.

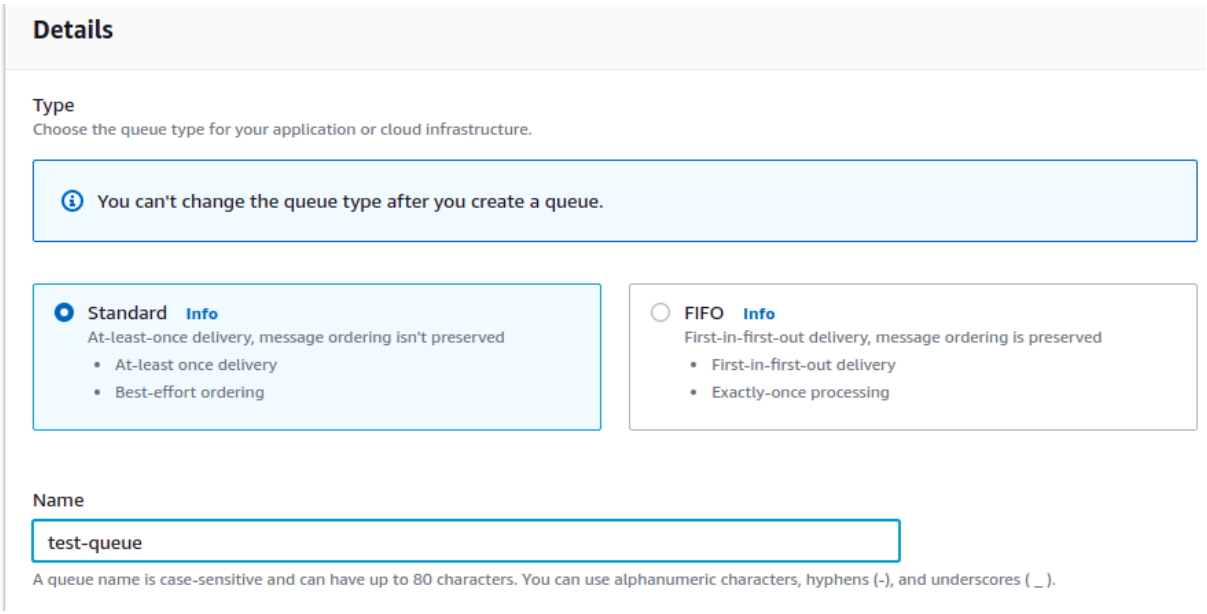
3. In Recording is off, click Turn on, and then choose Continue.



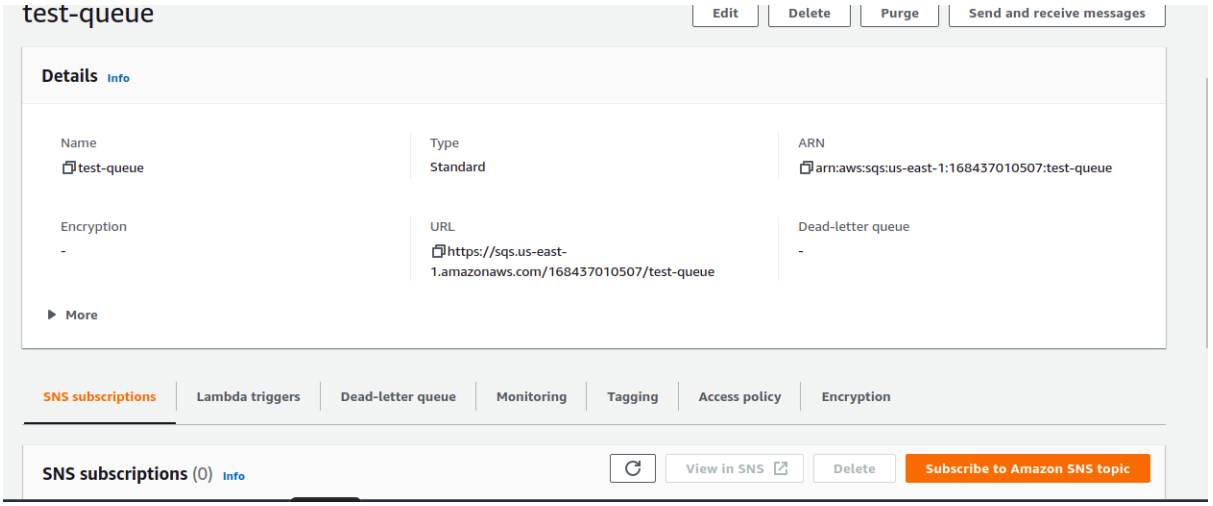
4.Configure Simple Queue service:

1. Open SQS Console

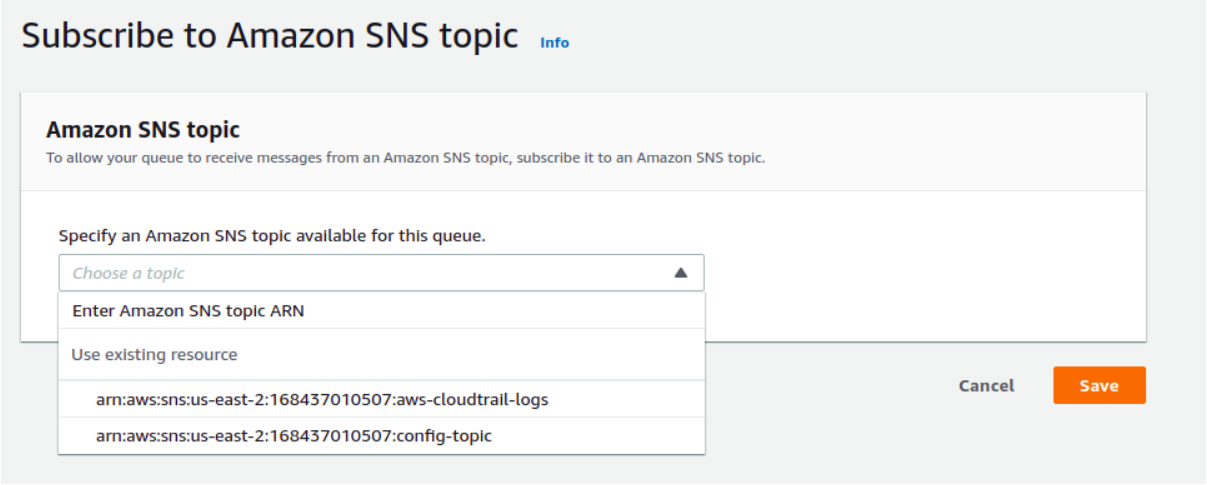
2.Click Create queue Then Enter Queue name and Click Create queue



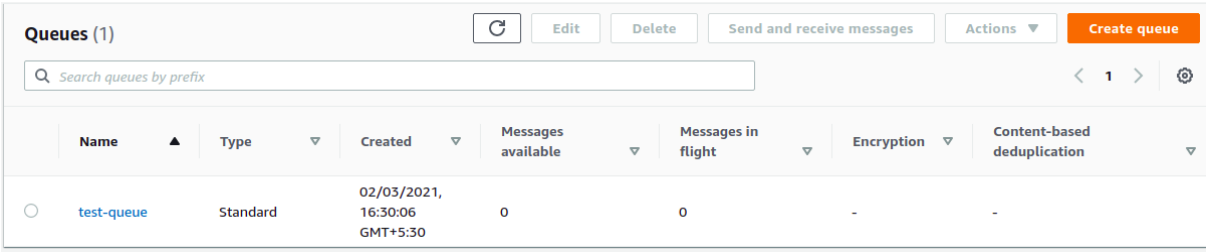
3.Click **Subscribe to Amazon SNS topic**



4.Click Specific Amazon topic and save it



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



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

Name	Instance ID	Instance state	Instance type
Heavy-Forwar...	i-0ce71fe2353fca174	Running	t2.medium
peer-node-1	i-05210d22dea7dbbf3	Running	t2.micro
peer-node-2	i-00675573af38330d9	Running	t2.micro
master-node	i-0a5f43e0d9609ca84	Running	t2.micro
Search-Head	i-06a04941f91e52cee	Running	t2.small
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

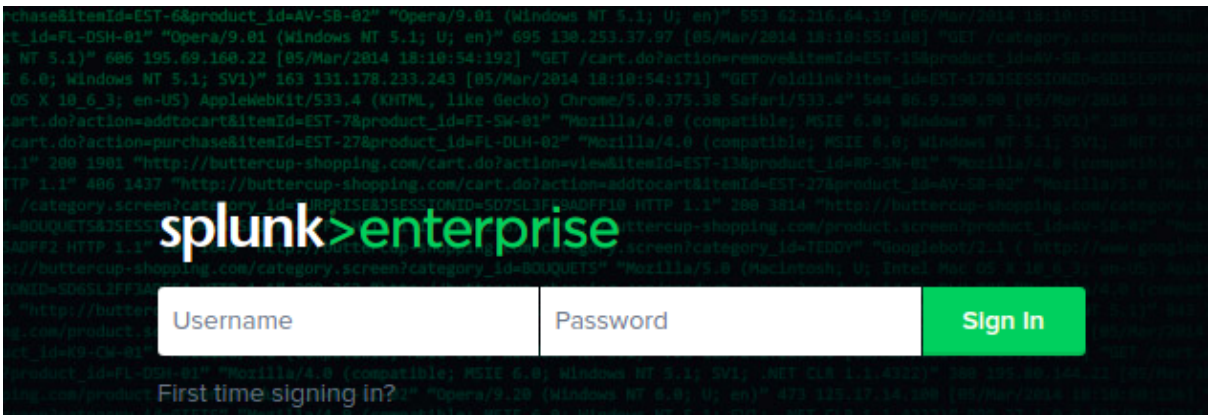
Inbound rules | Outbound rules | Tags

Inbound rules (5) Edit inbound rules

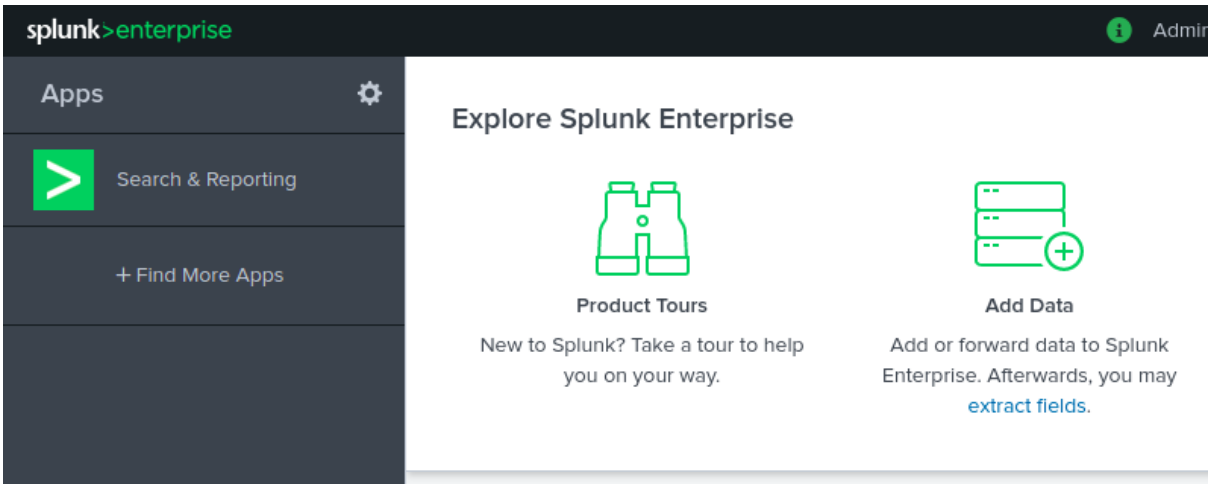
Type	Protocol	Port range	Source	Description - optional
Custom TCP	TCP	8080	0.0.0.0/0	–
Custom TCP	TCP	8000	0.0.0.0/0	–
SSH	TCP	22	103.5.112.131/32	–
Custom TCP	TCP	8089	0.0.0.0/0	–
Custom TCP	TCP	9997	0.0.0.0/0	–

6. Indexer Clustering Setup :

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

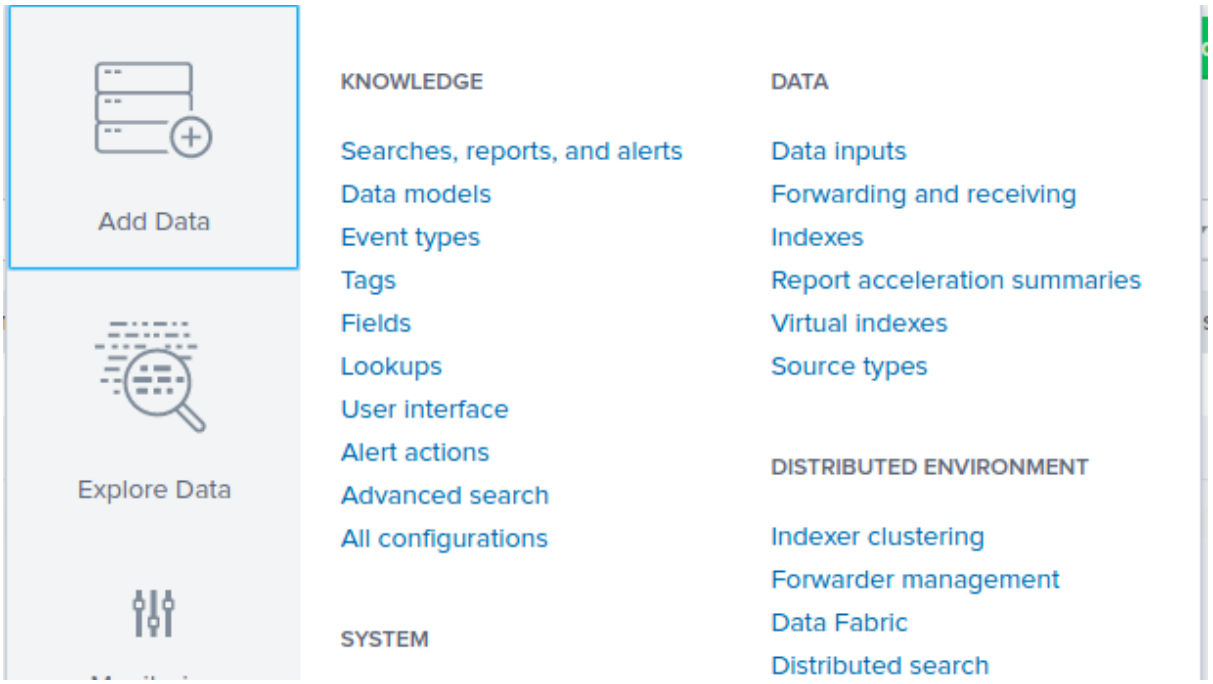


- .Enter user name and password then Sigin to console and below image will appear:

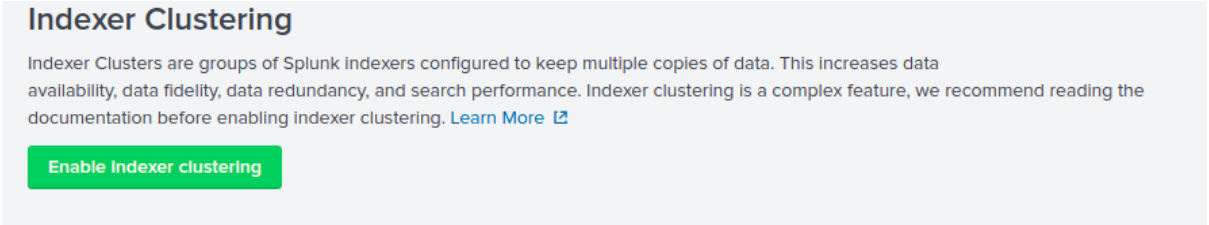


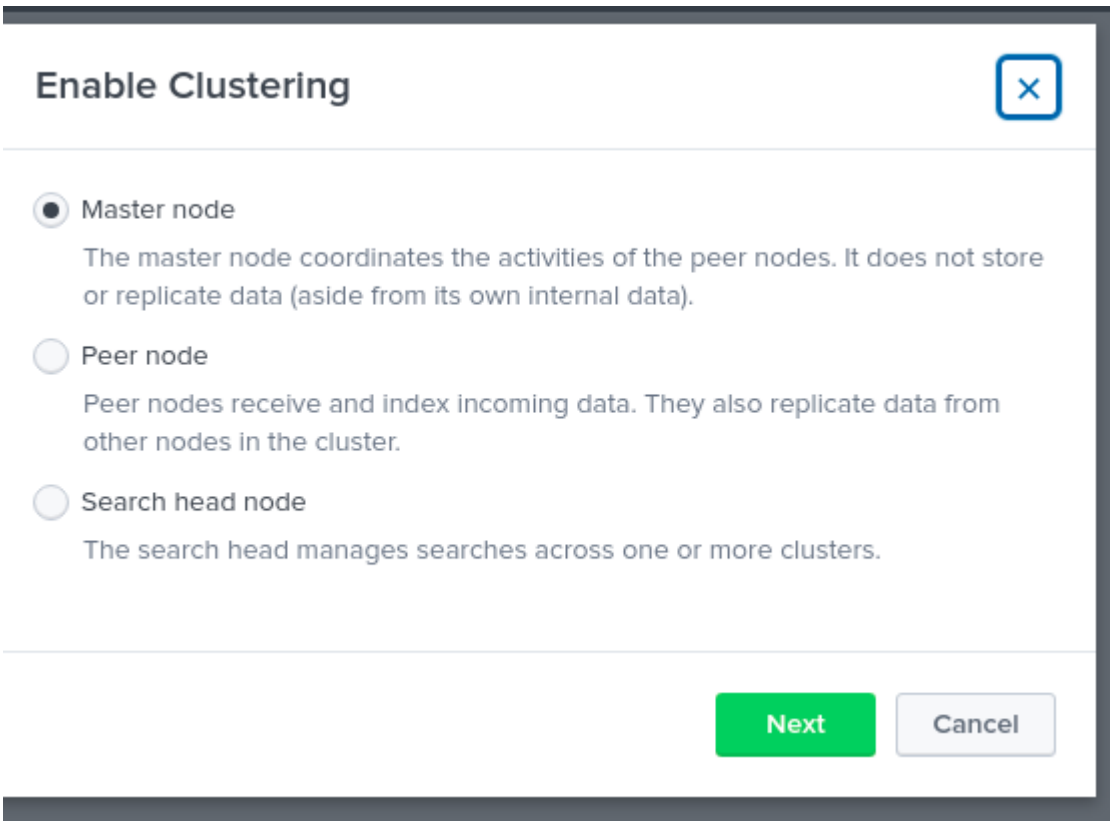
- . **Configuring Master Node :**

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

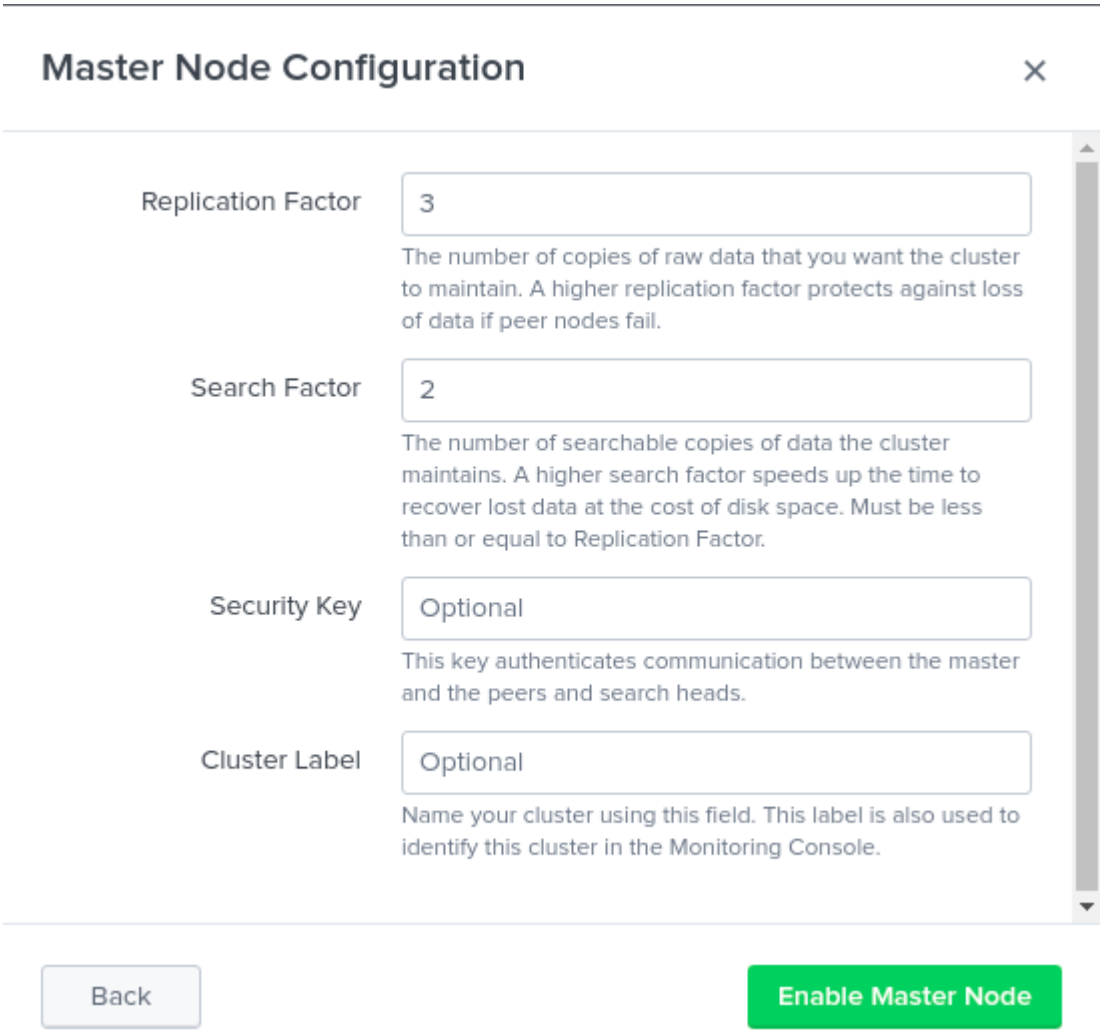


- .





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



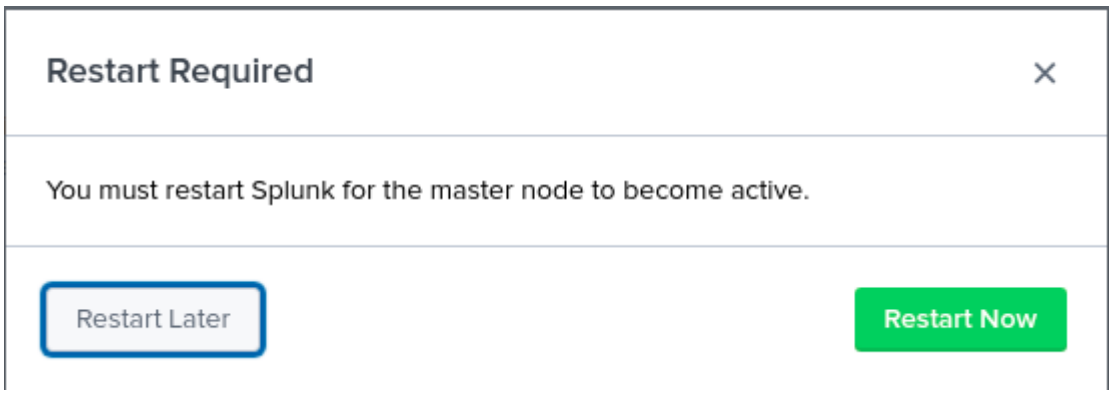
● Replication Factor - Keep the Replication Factor as 3 which is by default

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
```

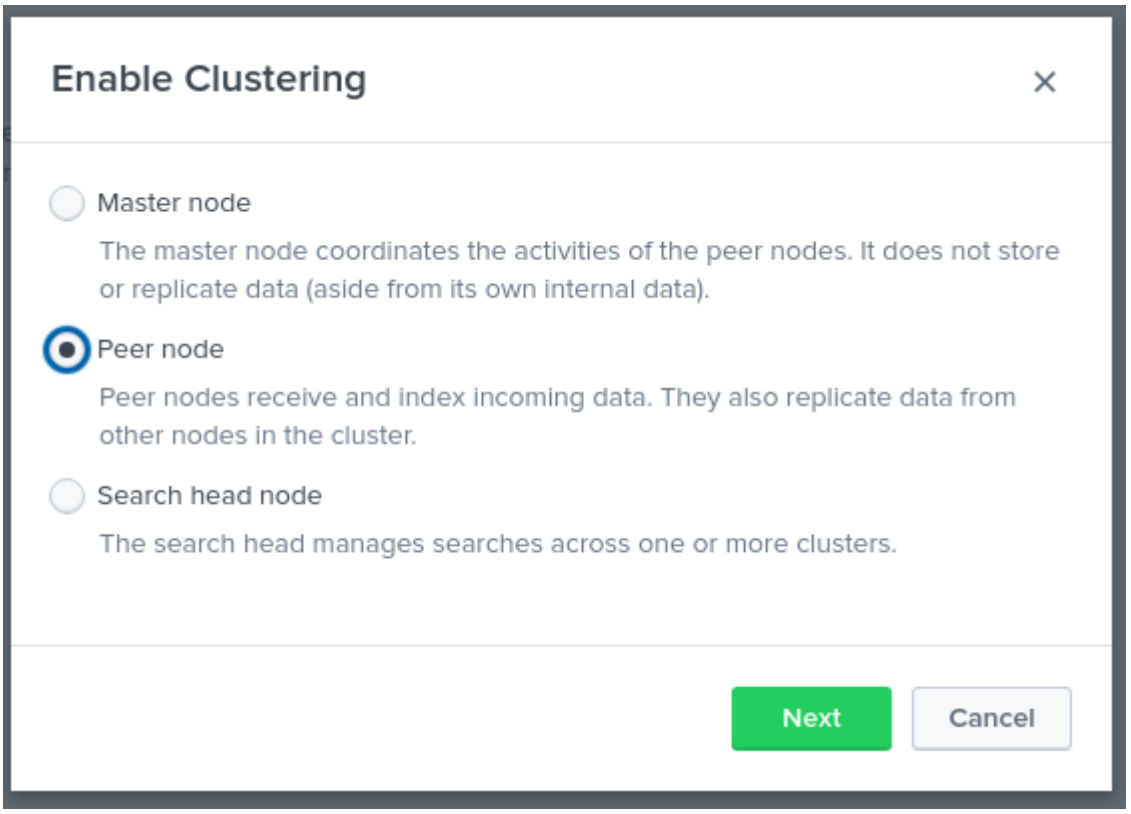
- Now insert the below text in indexes.conf file

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

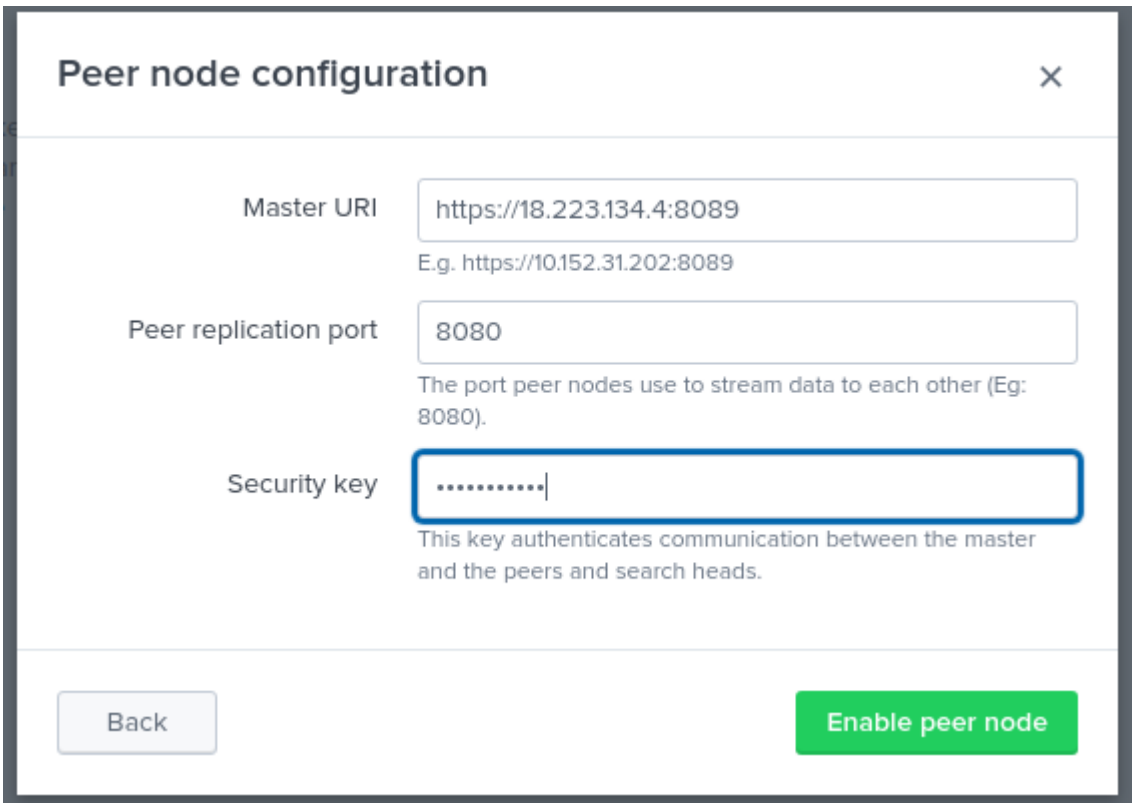
- After pasting the above text SAVE the file and Restart the server.

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.



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



- NOTE : Use same Security Key which we used for configuring Master Node.

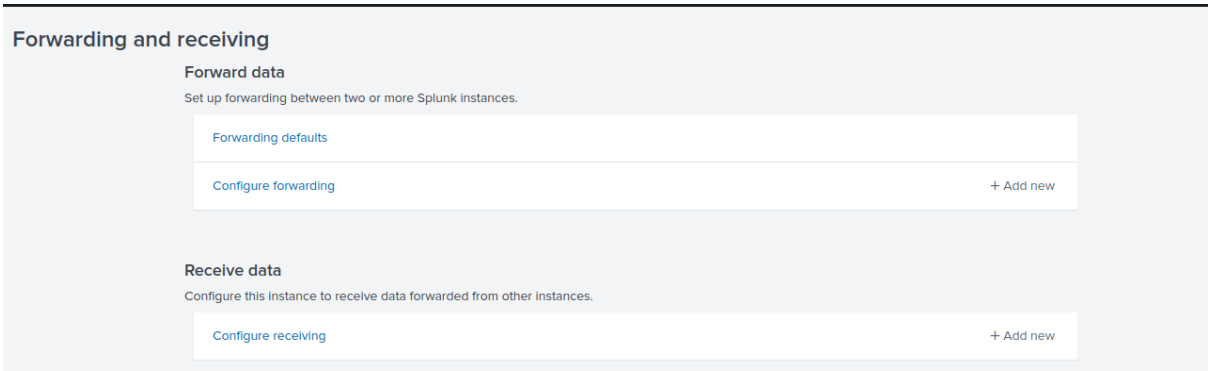
- After clicking on Enable Peer Node the Splunk has to be restarted.

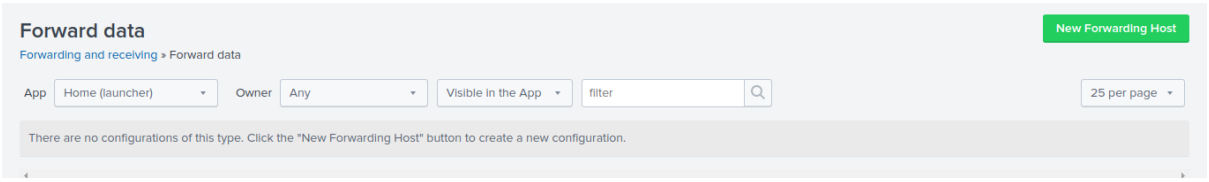
- Likewise connect other 2 (Indexer's) peer nodes to cluster.

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.



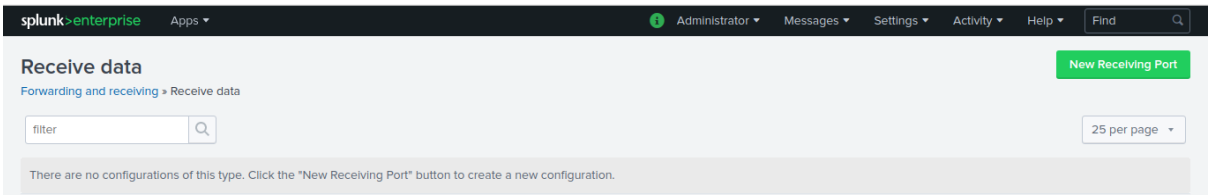


- After clicking on New Forwarding Host it will ask you to set Host. Host = Host IP:9997

- 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

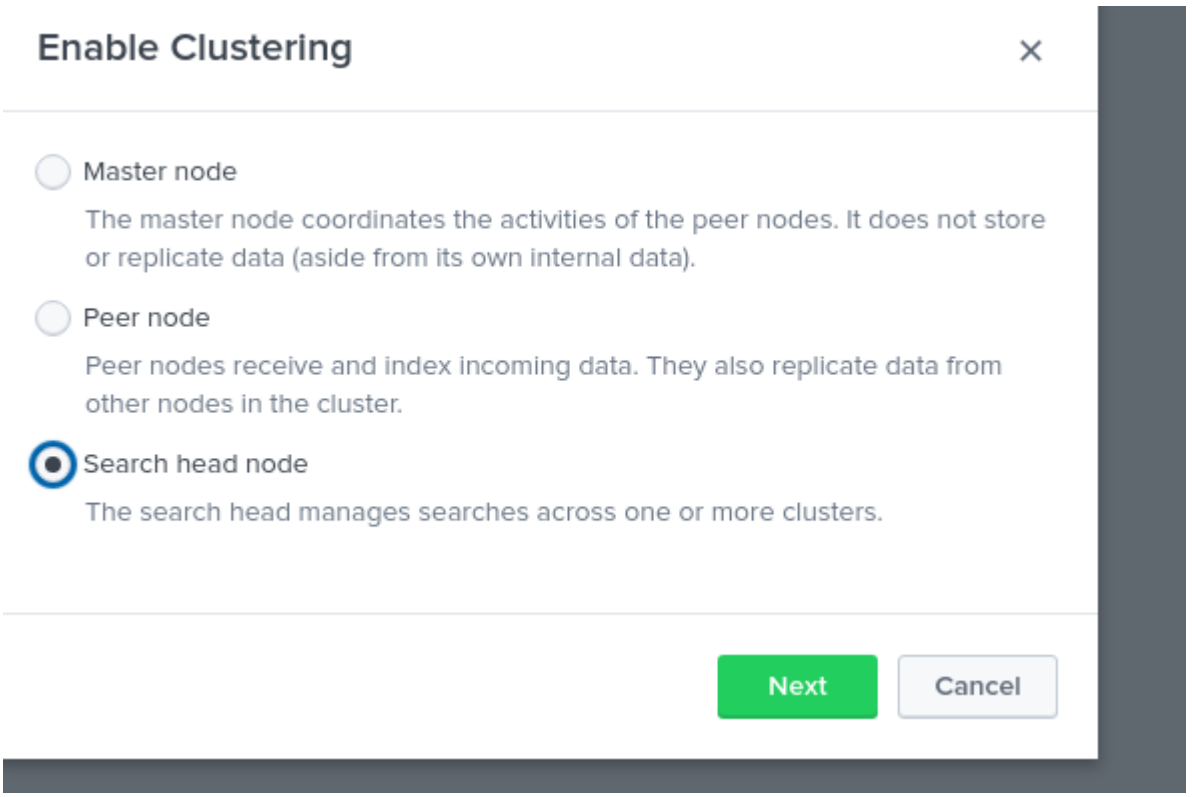
- Now come to Peer Nodes GUI and go to settings - Forwarding and Receiving - Configure Receiving - New Receiving Port.



- Likewise set port for other 2 Indexers.

Connecting Search Head to Cluster:

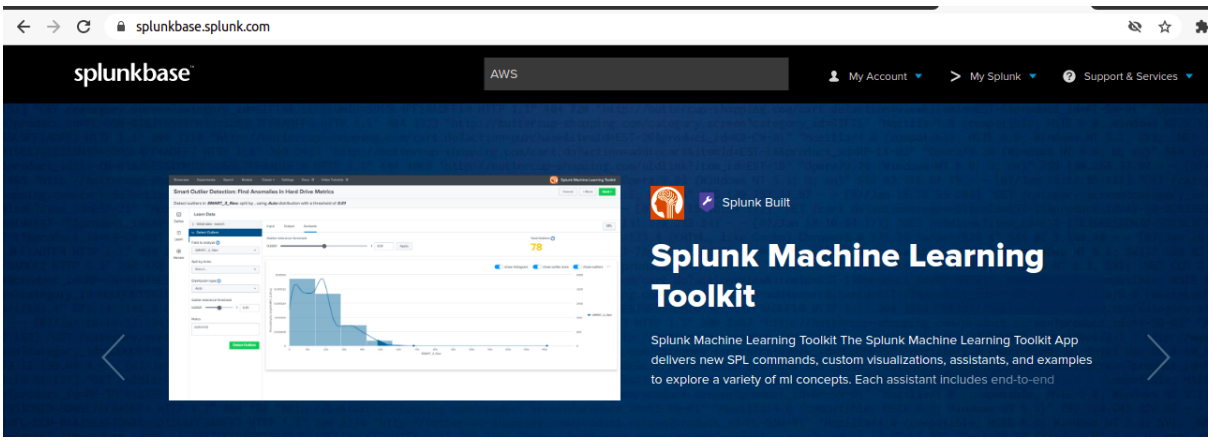
Then we should go to settings - Indexer clustering - Enable Indexer clustering - (select)
Search Head node - (click on) Next.



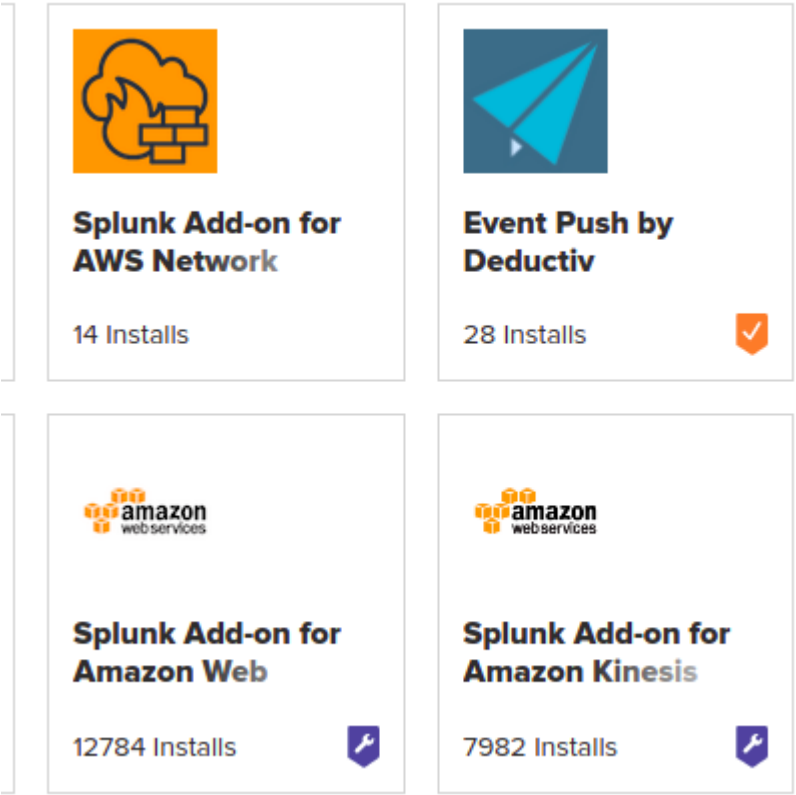
- Master IP = Master Node IP
- Security Key : Use same Security Key which we used for configuring Master Node.
- After clicking on Enable Search Head Node the Splunk has to be restarted.

7.AWS add-on and AWS app Download:

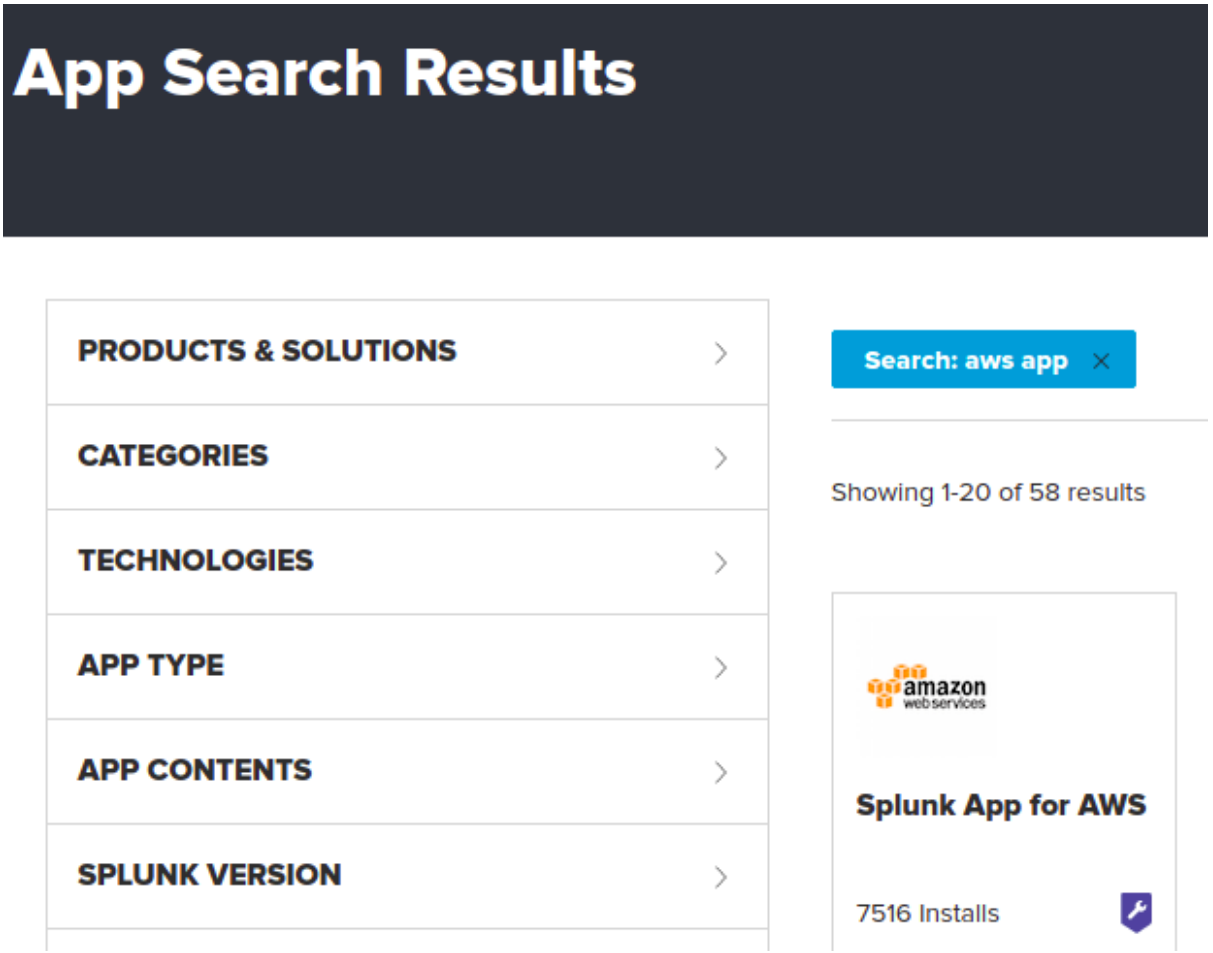
(*) Download AWS add-on and AWS app in **splunkbase.com**



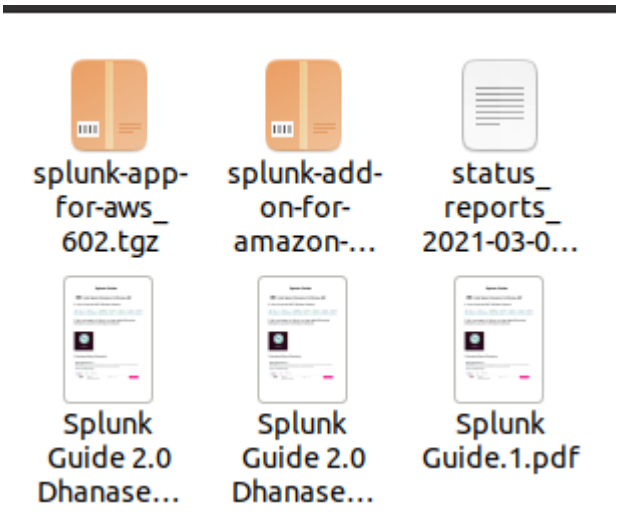
(*) Search AWS add-on in search bar and download it



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

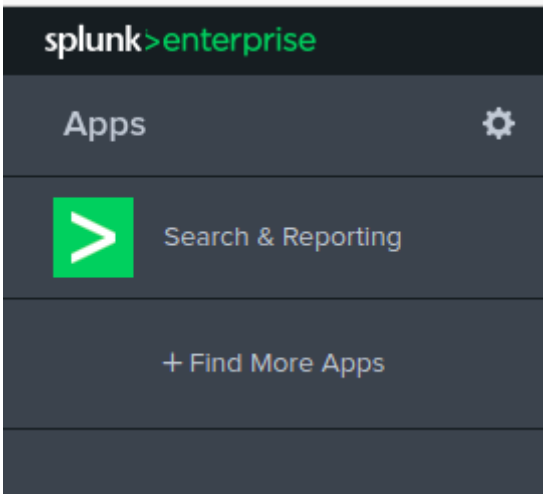


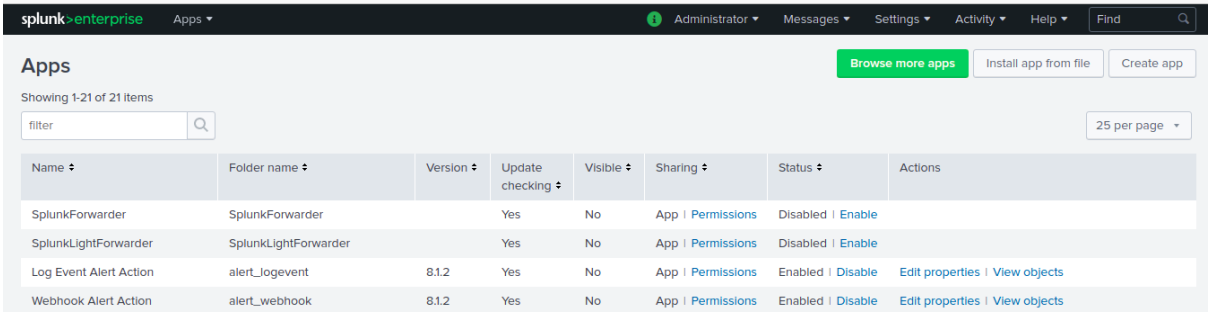
(*)Now Downloaded in AWS add-on and AWS app:



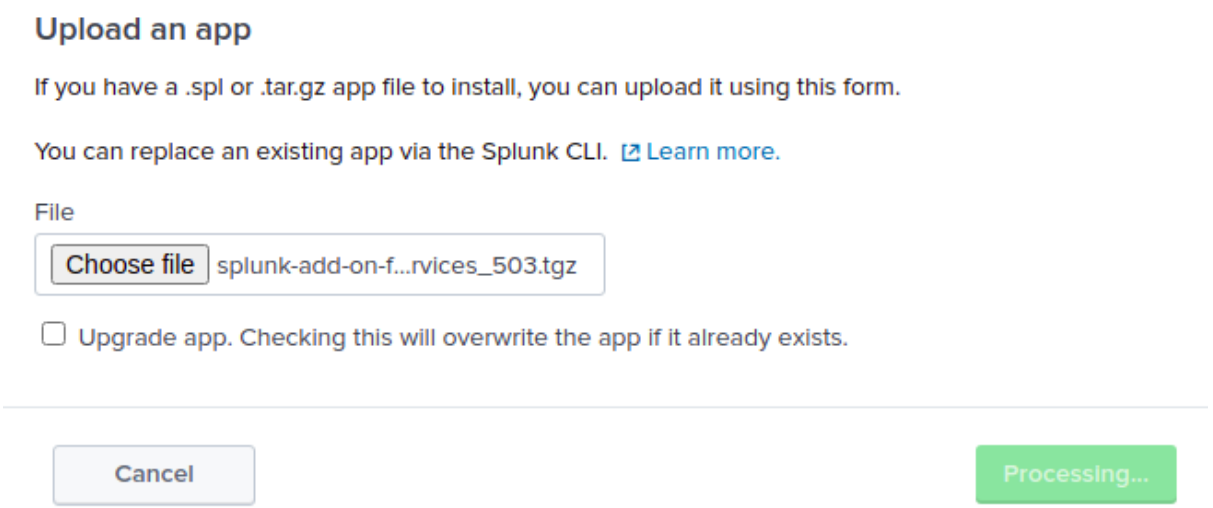
7.Install the AWS Add-on In Heavy Forwarder：

1)Click Manage Apps Icon and get below image

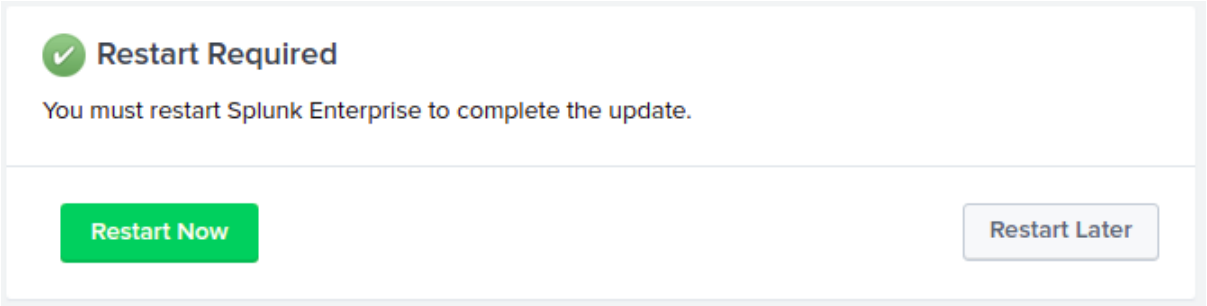


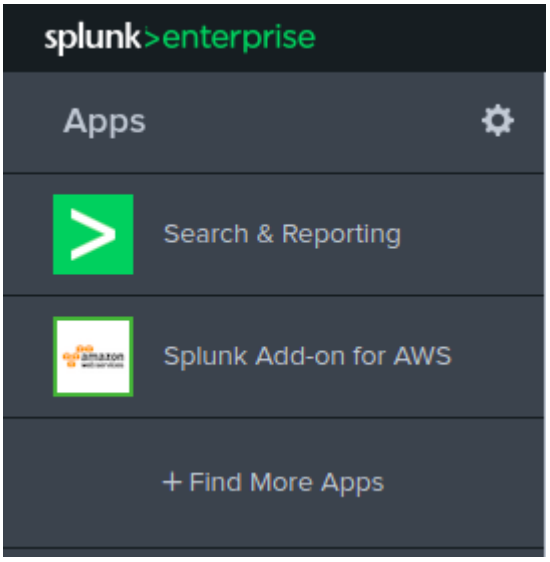


(2)



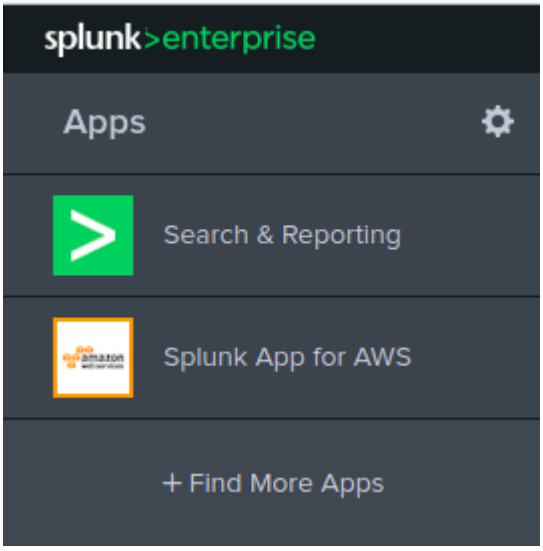
(3)Then click restart icon and below image will appear





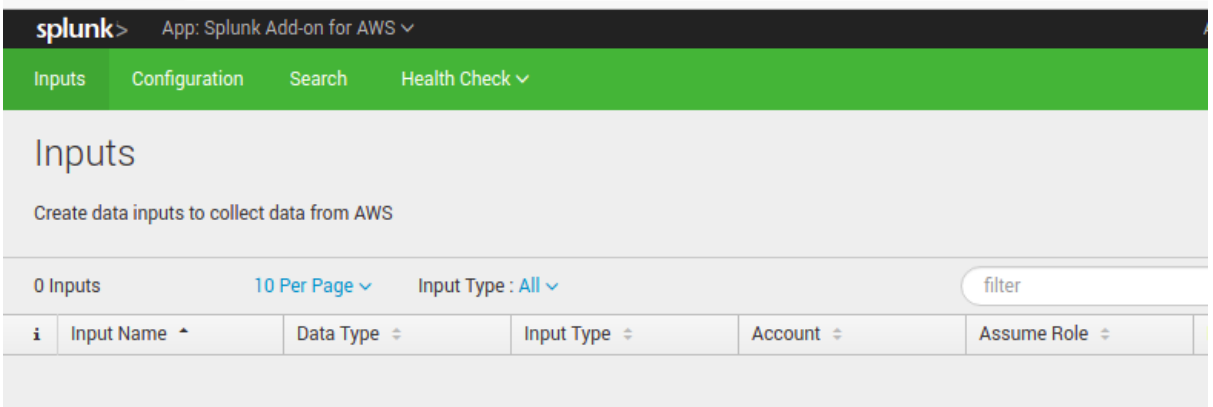
8.Install **AWS app in Search Head :**

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

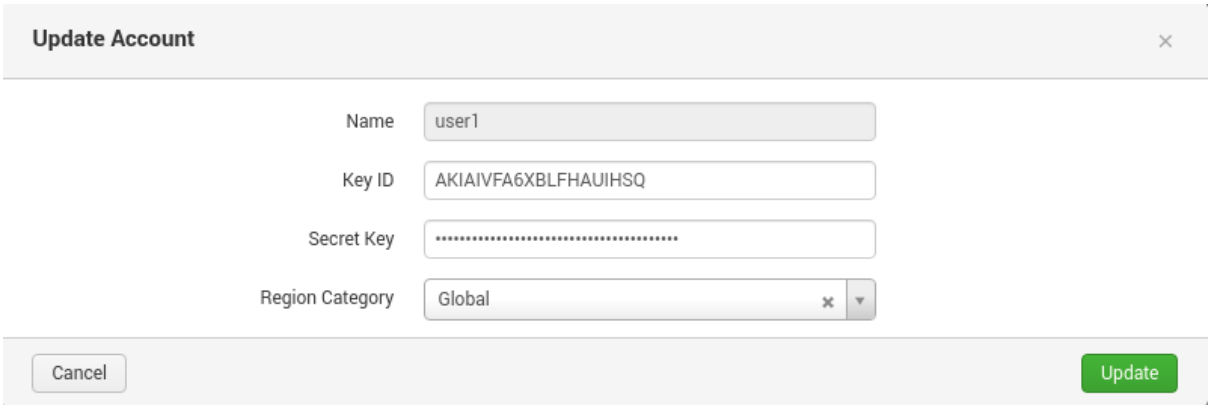


9.Configure AWS add-on :

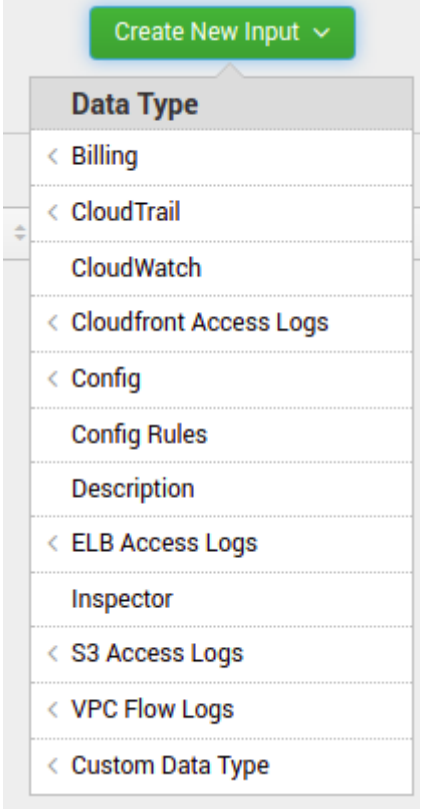
1)click splunk add-on for aws and below image will appear



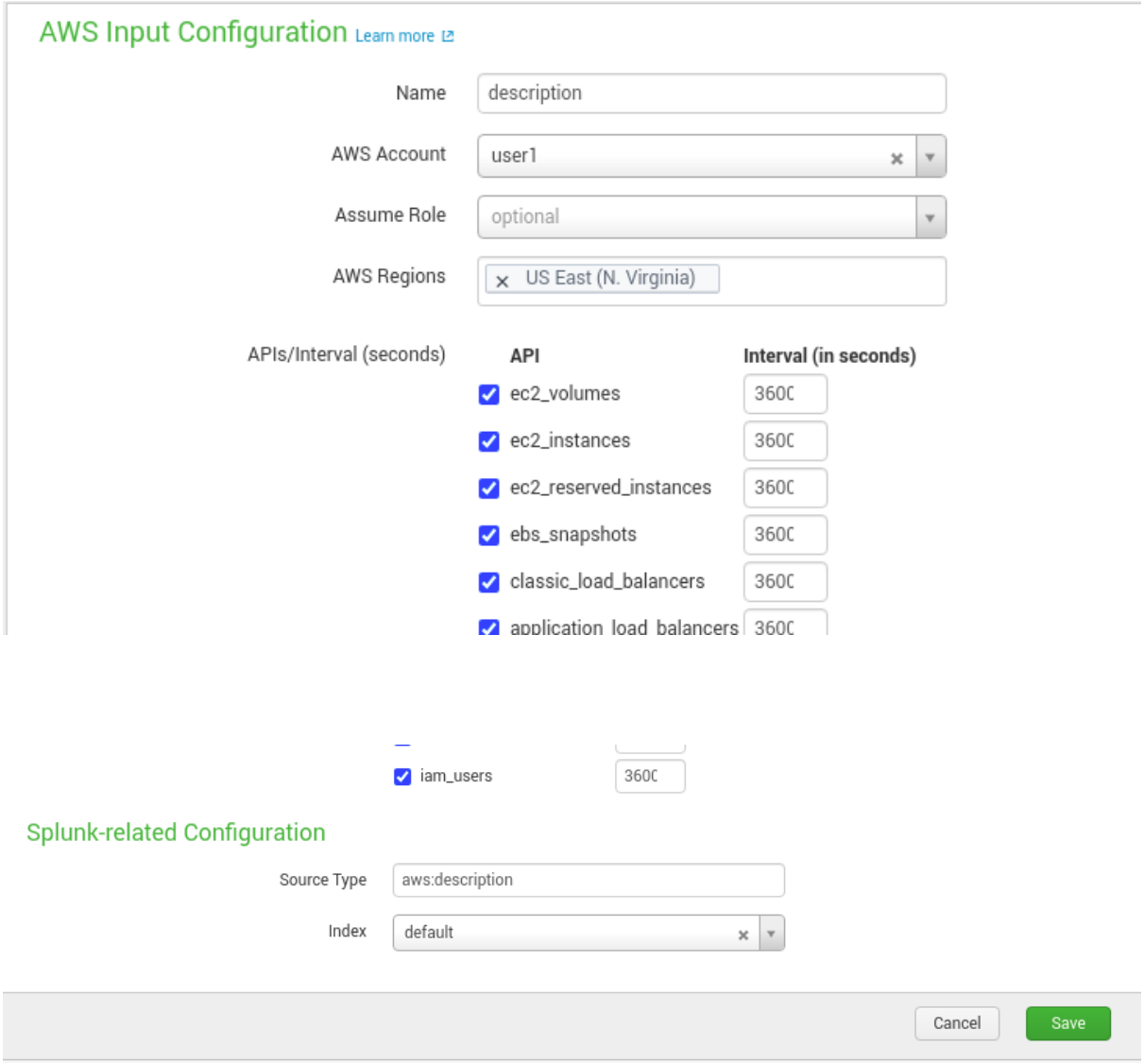
(2)first click **configuration** then Click **Add Icon**



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



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

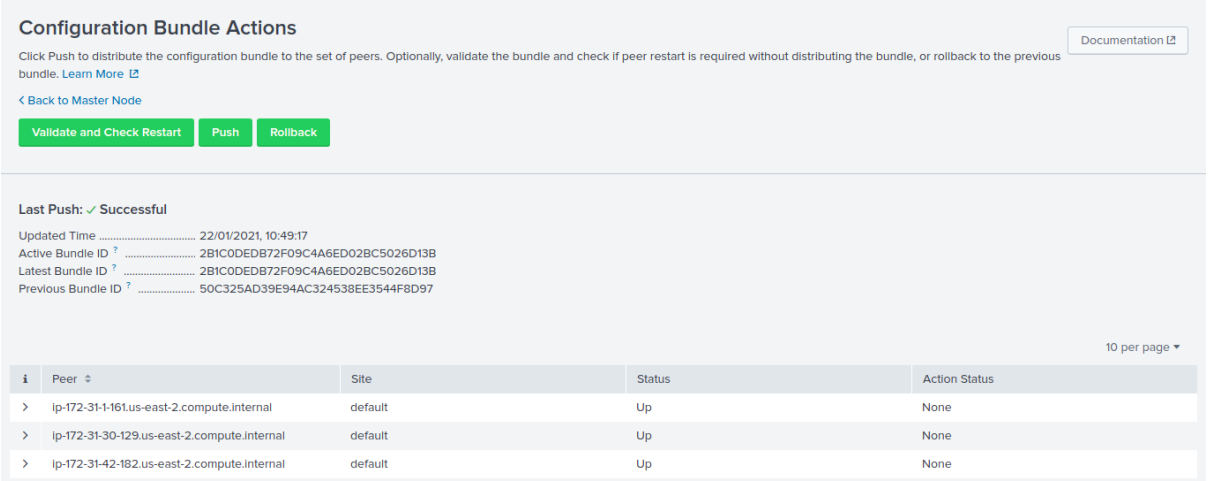


5.Finally Following input will give following input:

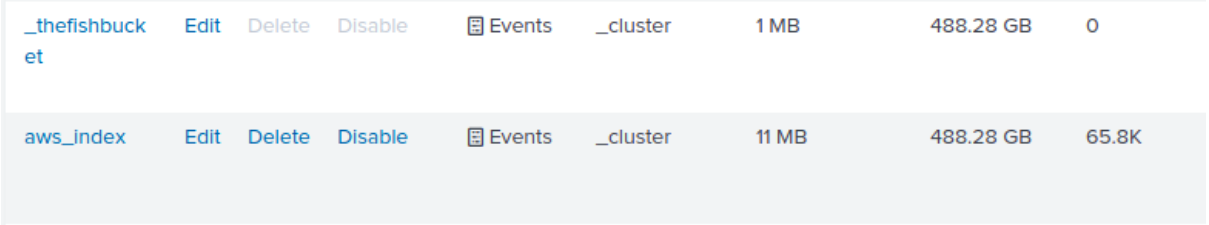
i	Input Name	Data Type	Input Type	Account
>	aws	Description	Description	aws-config
>	cloud-watch	CloudWatch	CloudWatch	aws-config
>	cloudtrail	CloudTrail	CloudTrail	aws-config
>	config	Config	Config	aws-config

10. Configuration Bundle: Validate and Push

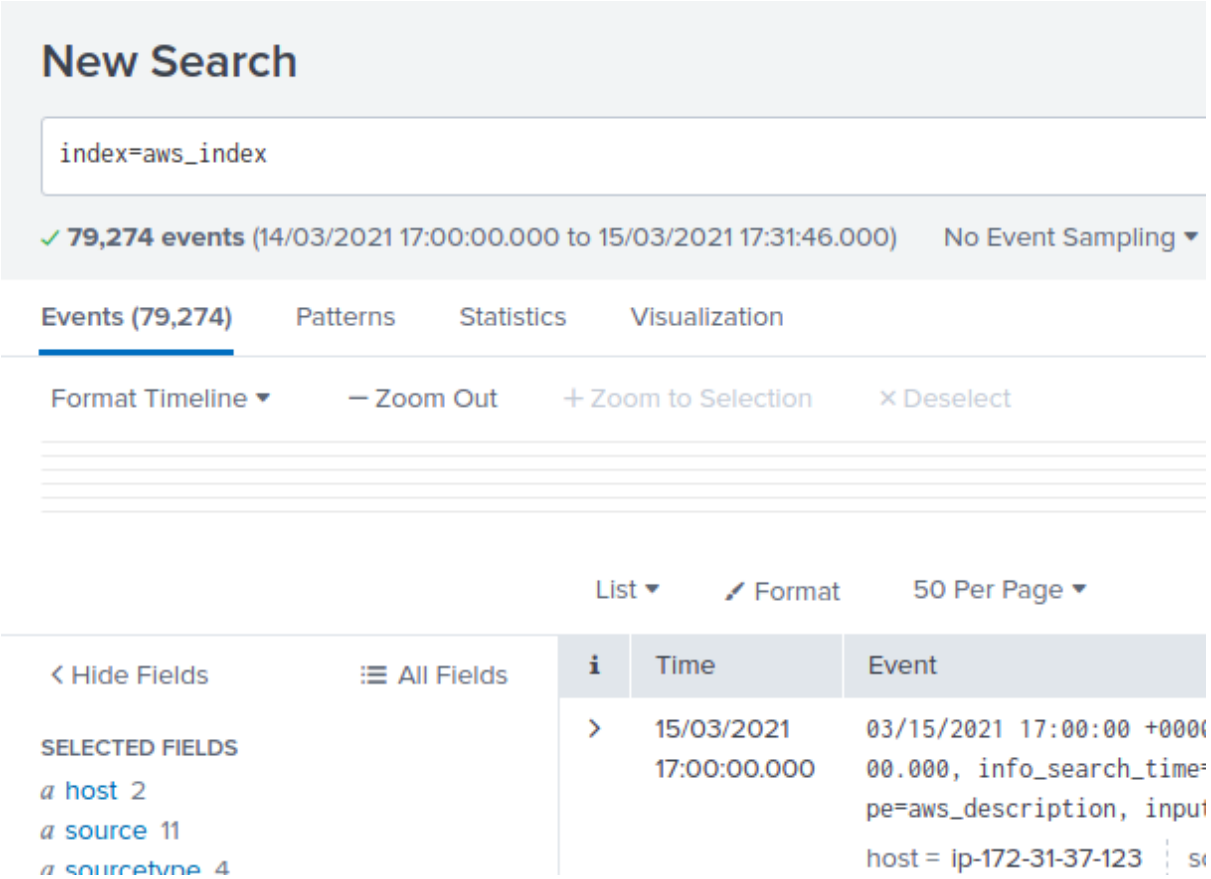
● Now login to Mastor Node web interface and Edit Configuration Bundle Actions to Click Validate and Check Restart and push



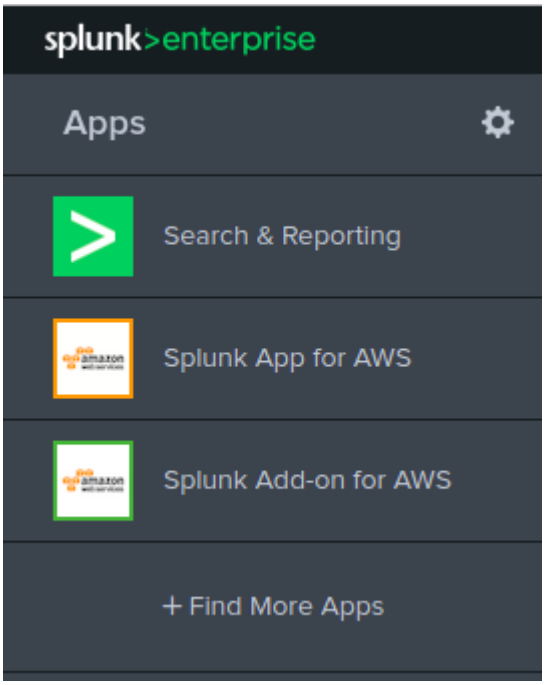
Then automatically creates **aws_index** on PEER Nodes.See the below Image



Now go to one of the Searc head GUI and search for index="aws_index".
You get information
similar to below image



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
[tcpout]
defaultGroup = my_search_peers  # Name of the search peer group
forwardedindex.filter.disable = true
indexAndForward = false
[tcpout: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
[tcpout]
defaultGroup = my_search_peers  # Name of the search peer group
forwardedindex.filter.disable = true
indexAndForward = false
[tcpout: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