Project Name- User Remediation

A. We write the Linux script and find the resources in which have not define tags.

1.install awscli

2.configure user

- aws configure

3.Fire this command- aws resourcegroupstaggingapi get-resources --output Json

- This command retrieves a list of AWS resources along with their tags in JSON format using the AWS CLI.

4.Write script in- vi list-untags-resources.sh

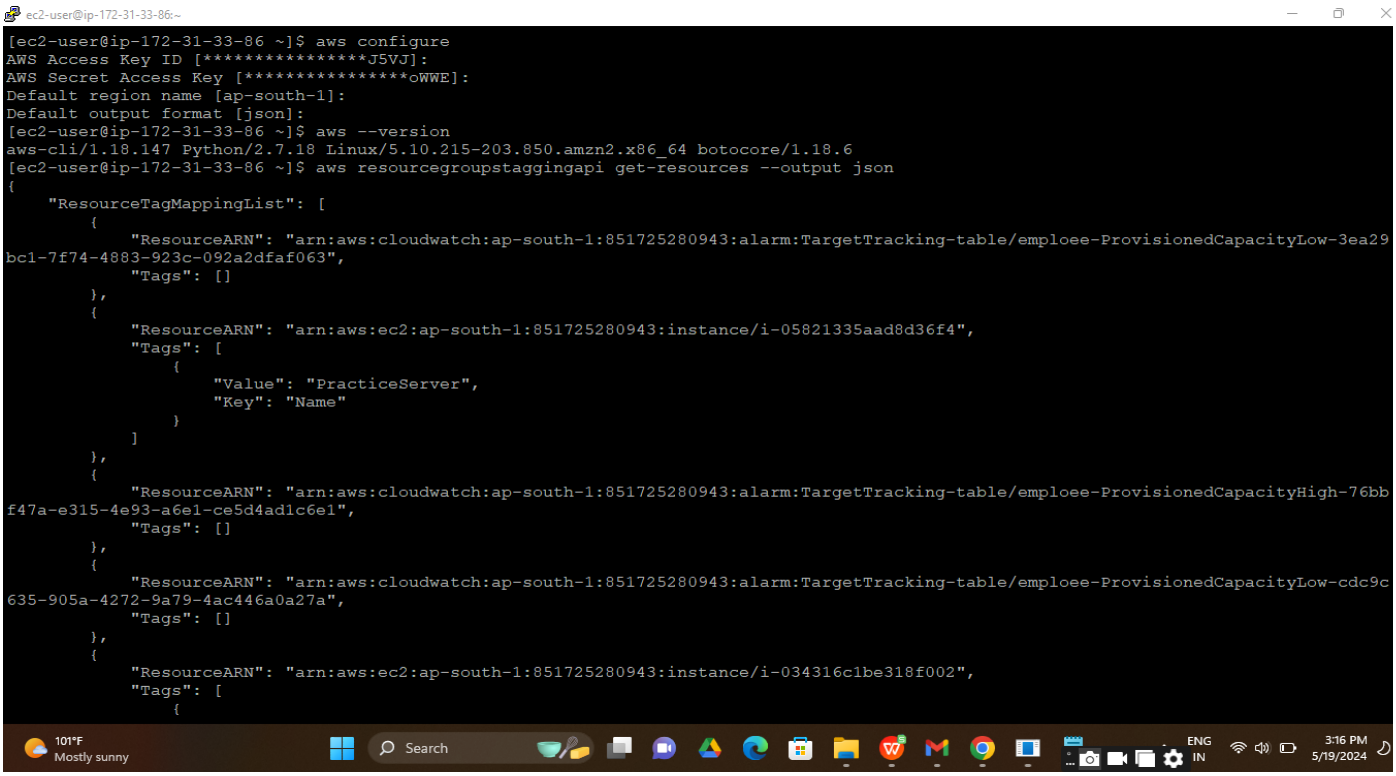
- aws resourcesgroupstaggingapi get-resources --output json | jq -r '.ResourceTagMappingList[] | select(.Tags | length==0) | .ResourceARN'

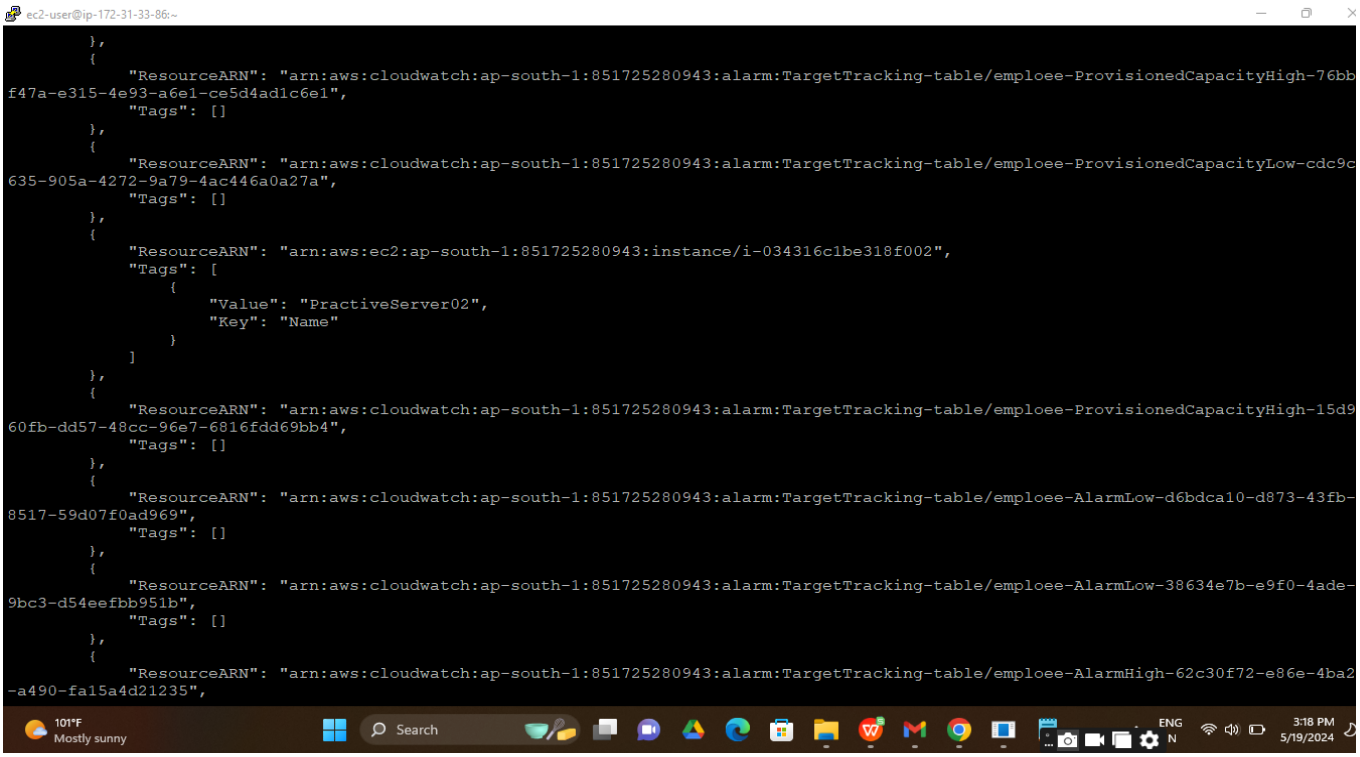
5.Retrieves a list of AWS resources along with their tags, filters out the resources that do not have any tags, and outputs the ARNs of those untagged resources.

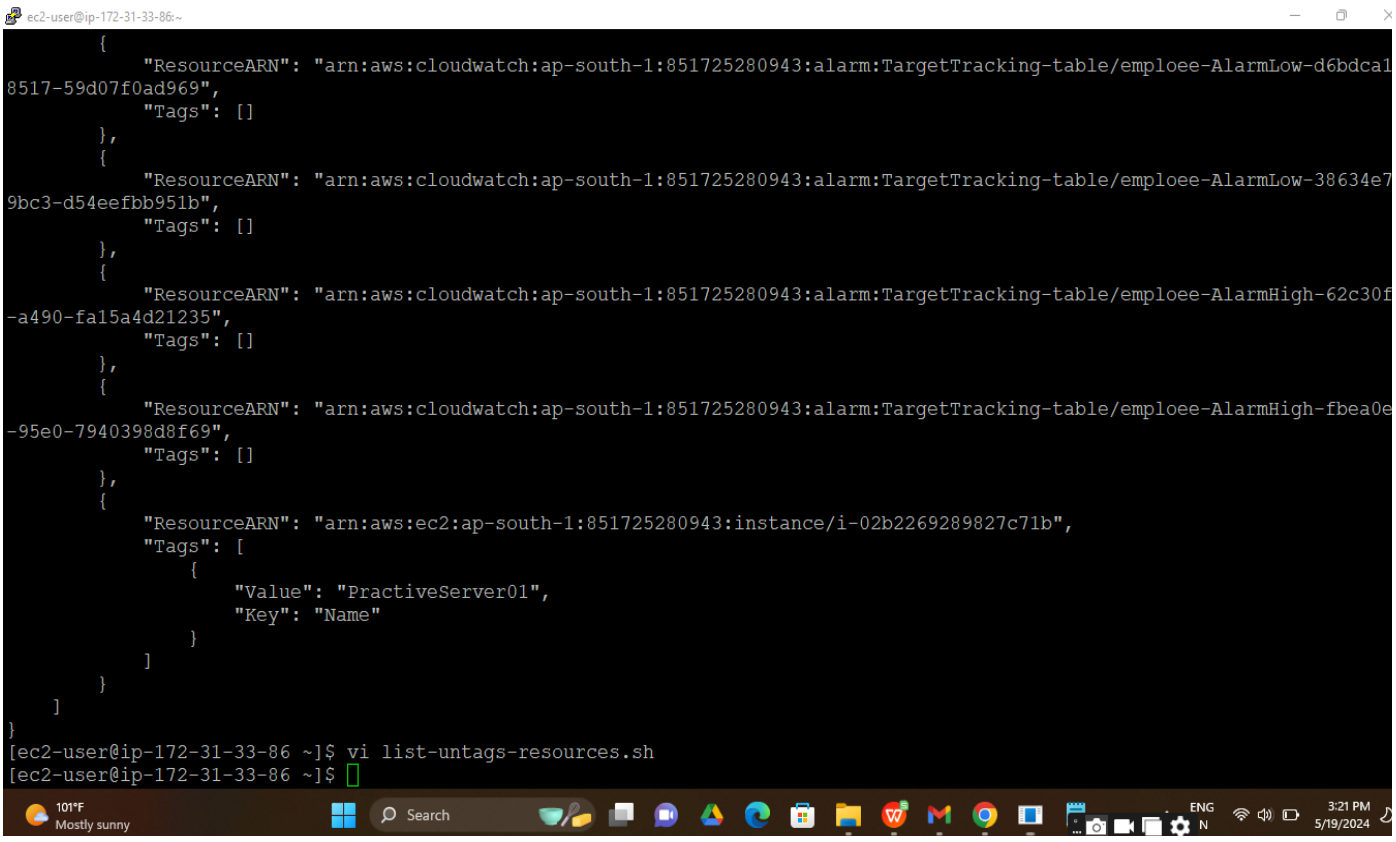
6. Give the permission to execute- chmod +x list-untags-resources.sh

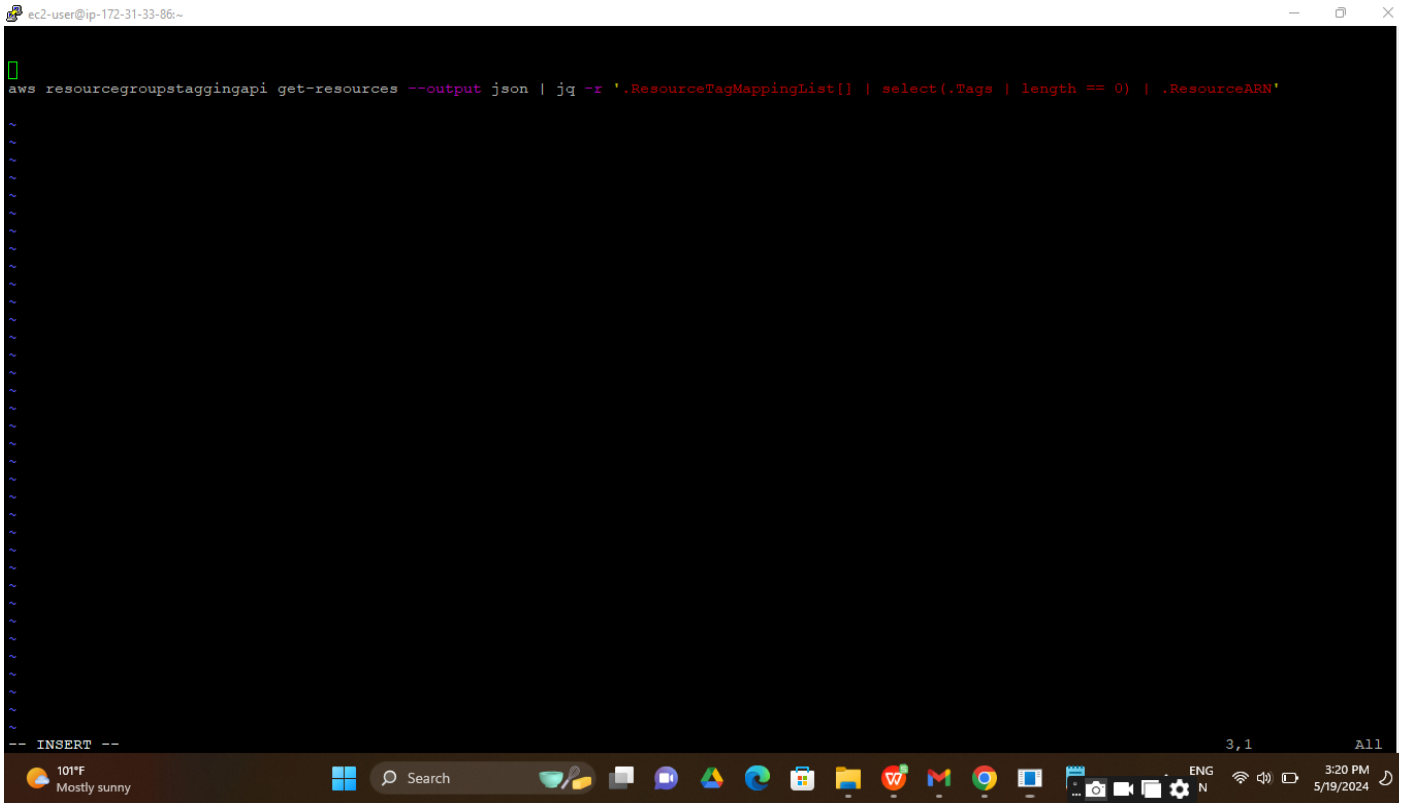
- ./list-untags-resources.sh

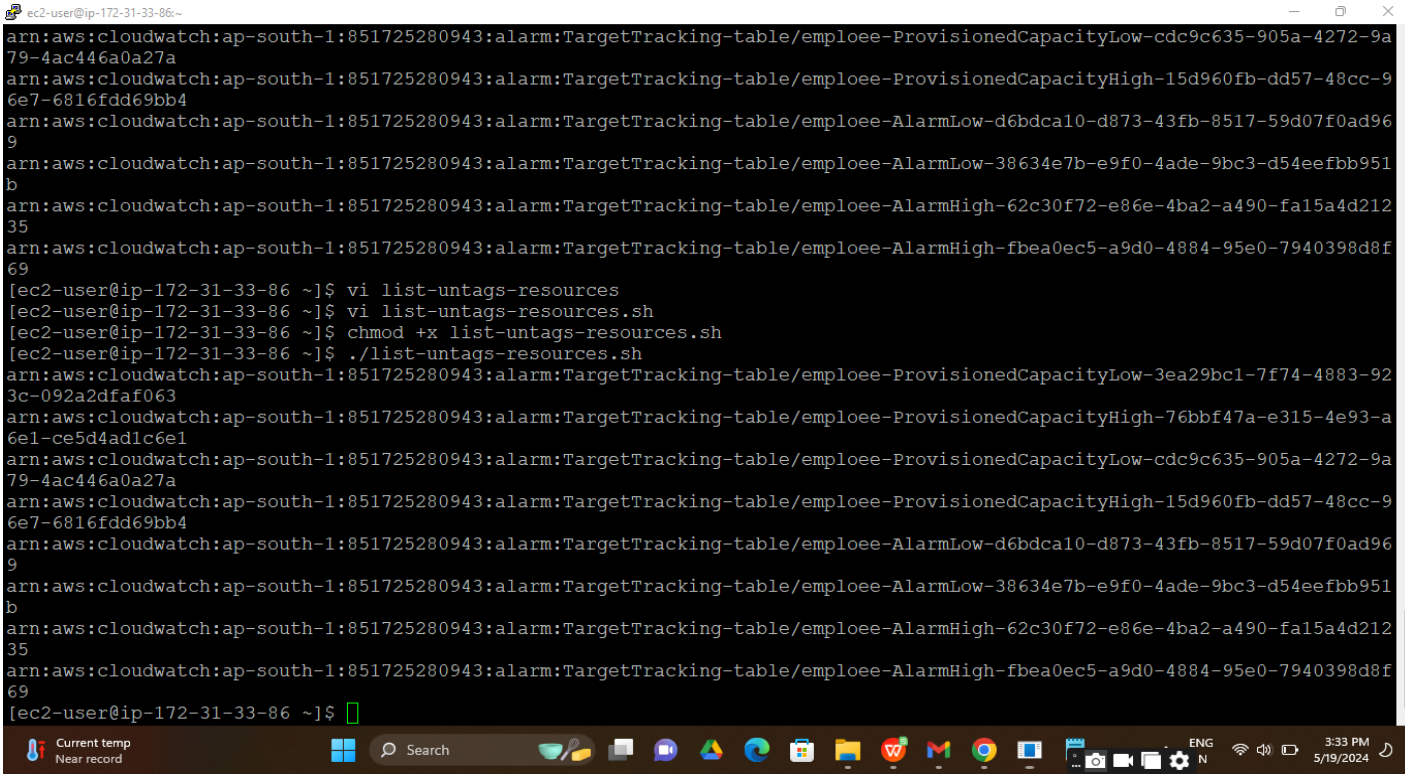
7. Here the untaggs resources.




ec2-user@ip-172-31-33-86:~
},
{
"ResourceARN": "arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/employee-ProvisionedCapacityHigh-76bb
f47a-e315-4e93-a6e1-ce5d4ad1c6e1",
"Tags": []
},
{
"ResourceARN": "arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/employee-ProvisionedCapacityLow-cdc9c
635-905a-4272-9a79-4ac446a0a27a",
"Tags": []
},
{
"ResourceARN": "arn:aws:ec2:ap-south-1:851725280943:instance/i-034316c1be318f002",
"Tags": [
{
"Value": "PractiveServer02",
"Key": "Name"
}
]
},
{
"ResourceARN": "arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/employee-ProvisionedCapacityHigh-15d9
60fb-dd57-48cc-96e7-6816fdd69bb4",
"Tags": []
},
{
"ResourceARN": "arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/employee-AlarmLow-d6bdca10-d873-43fb-
8517-59d07f0ad969",
"Tags": []
},
{
"ResourceARN": "arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/employee-AlarmLow-38634e7b-e9f0-4ade-
9bc3-d54eefbb951b",
"Tags": []
},
{
"ResourceARN": "arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/employee-AlarmHigh-62c30f72-e86e-4ba2
-a490-fa15a4d21235",
101°F
Mostly sunny
Search
ENG
N
3:18 PM
5/19/2024


ec2-user@ip-172-31-33-86:~
{
"ResourceARN": "arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/employee-AlarmLow-d6bdca1
8517-59d07f0ad969",
"Tags": []
},
{
"ResourceARN": "arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/employee-AlarmLow-38634e7
9bc3-d54eefbb951b",
"Tags": []
},
{
"ResourceARN": "arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/employee-AlarmHigh-62c30f
-a490-fa15a4d21235",
"Tags": []
},
{
"ResourceARN": "arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/employee-AlarmHigh-fbea0e
-95e0-7940398d8f69",
"Tags": []
},
{
"ResourceARN": "arn:aws:ec2:ap-south-1:851725280943:instance/i-02b2269289827c71b",
"Tags": [
{
"Value": "PractiveServer01",
"Key": "Name"
}
]
}
]
}
[ec2-user@ip-172-31-33-86 ~]$ vi list-untags-resources.sh
[ec2-user@ip-172-31-33-86 ~]$
101°F
Mostly sunny
Search
ENG
N
3:21 PM
5/19/2024


ec2-user@ip-172-31-33-86:~
aws resourcegroupstaggingapi get-resources --output json | jq -r '.ResourceTagMappingList[] | select(.Tags | length == 0) | .ResourceARN'
-- INSERT --
3,1
All
101°F
Mostly sunny
Search
ENG
N
3:20 PM
5/19/2024


ec2-user@ip-172-31-33-86:~
arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/emploee-ProvisionedCapacityLow-cdc9c635-905a-4272-9a79-4ac446a0a27a
arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/emploee-ProvisionedCapacityHigh-15d960fb-dd57-48cc-96e7-6816fdd69bb4
arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/emploee-AlarmLow-d6bdca10-d873-43fb-8517-59d07f0ad969
arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/emploee-AlarmLow-38634e7b-e9f0-4ade-9bc3-d54eefbb951b
arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/emploee-AlarmHigh-62c30f72-e86e-4ba2-a490-fa15a4d21235
arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/emploee-AlarmHigh-fbea0ec5-a9d0-4884-95e0-7940398d8f69
[ec2-user@ip-172-31-33-86 ~]$ vi list-untags-resources
[ec2-user@ip-172-31-33-86 ~]$ vi list-untags-resources.sh
[ec2-user@ip-172-31-33-86 ~]$ chmod +x list-untags-resources.sh
[ec2-user@ip-172-31-33-86 ~]$ ./list-untags-resources.sh
arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/emploee-ProvisionedCapacityLow-3ea29bc1-7f74-4883-923c-092a2dfaf063
arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/emploee-ProvisionedCapacityHigh-76bbf47a-e315-4e93-a6e1-ce5d4ad1c6e1
arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/emploee-ProvisionedCapacityLow-cdc9c635-905a-4272-9a79-4ac446a0a27a
arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/emploee-ProvisionedCapacityHigh-15d960fb-dd57-48cc-96e7-6816fdd69bb4
arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/emploee-AlarmLow-d6bdca10-d873-43fb-8517-59d07f0ad969
arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/emploee-AlarmLow-38634e7b-e9f0-4ade-9bc3-d54eefbb951b
arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/emploee-AlarmHigh-62c30f72-e86e-4ba2-a490-fa15a4d21235
arn:aws:cloudwatch:ap-south-1:851725280943:alarm:TargetTracking-table/emploee-AlarmHigh-fbea0ec5-a9d0-4884-95e0-7940398d8f69
[ec2-user@ip-172-31-33-86 ~]$
Current temp
Near record
Search
ENG
N
3:33 PM
5/19/2024

B.IAM users whose last activity days are greater than 30 so I want those users to mark as de activate.

1. install awscli

2. aws configure

3. fire this command- aws iam list-users

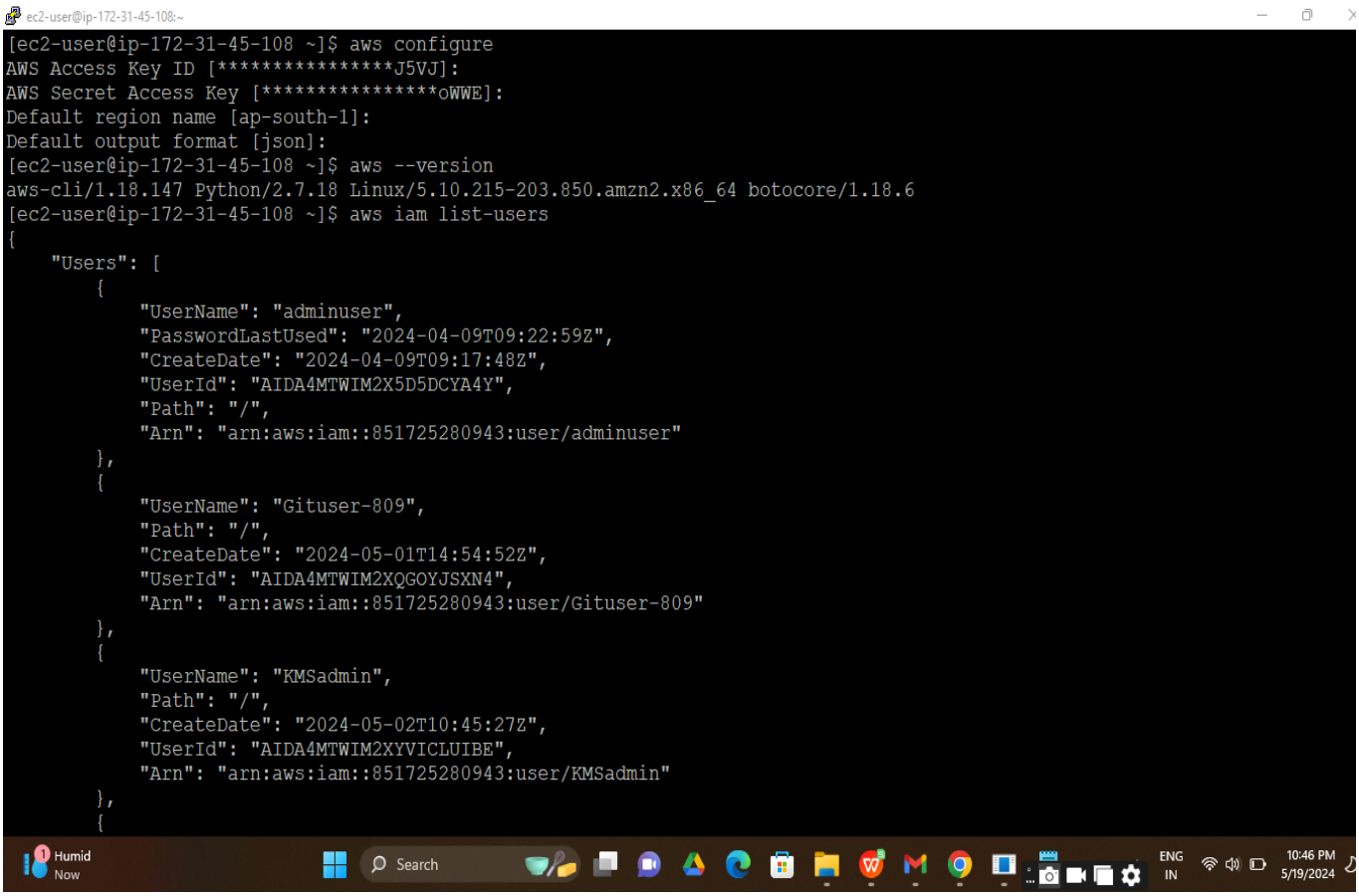
   - The command aws iam list-users -lists all IAM users in our AWS account using the AWS CLI.

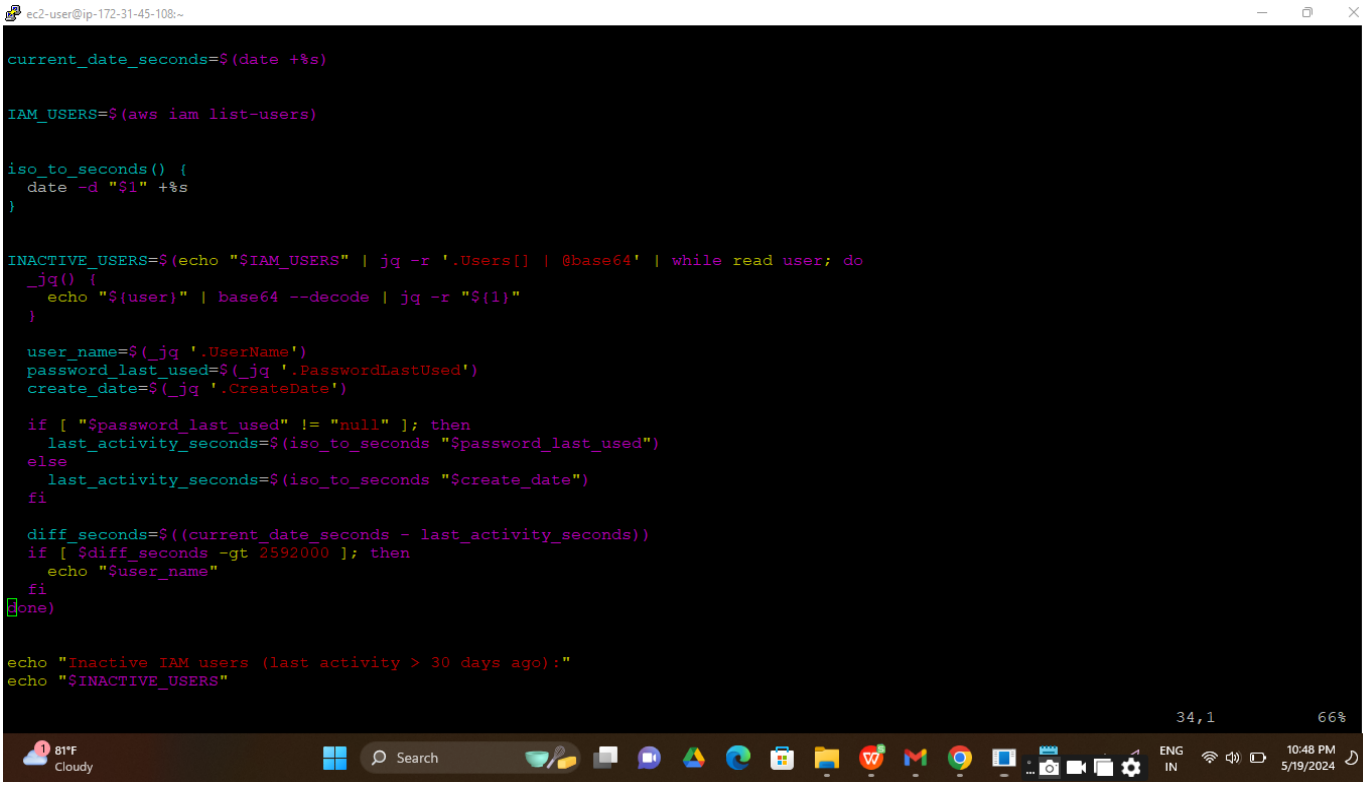
4.Write script in- vi list-of-inactive-users.sh

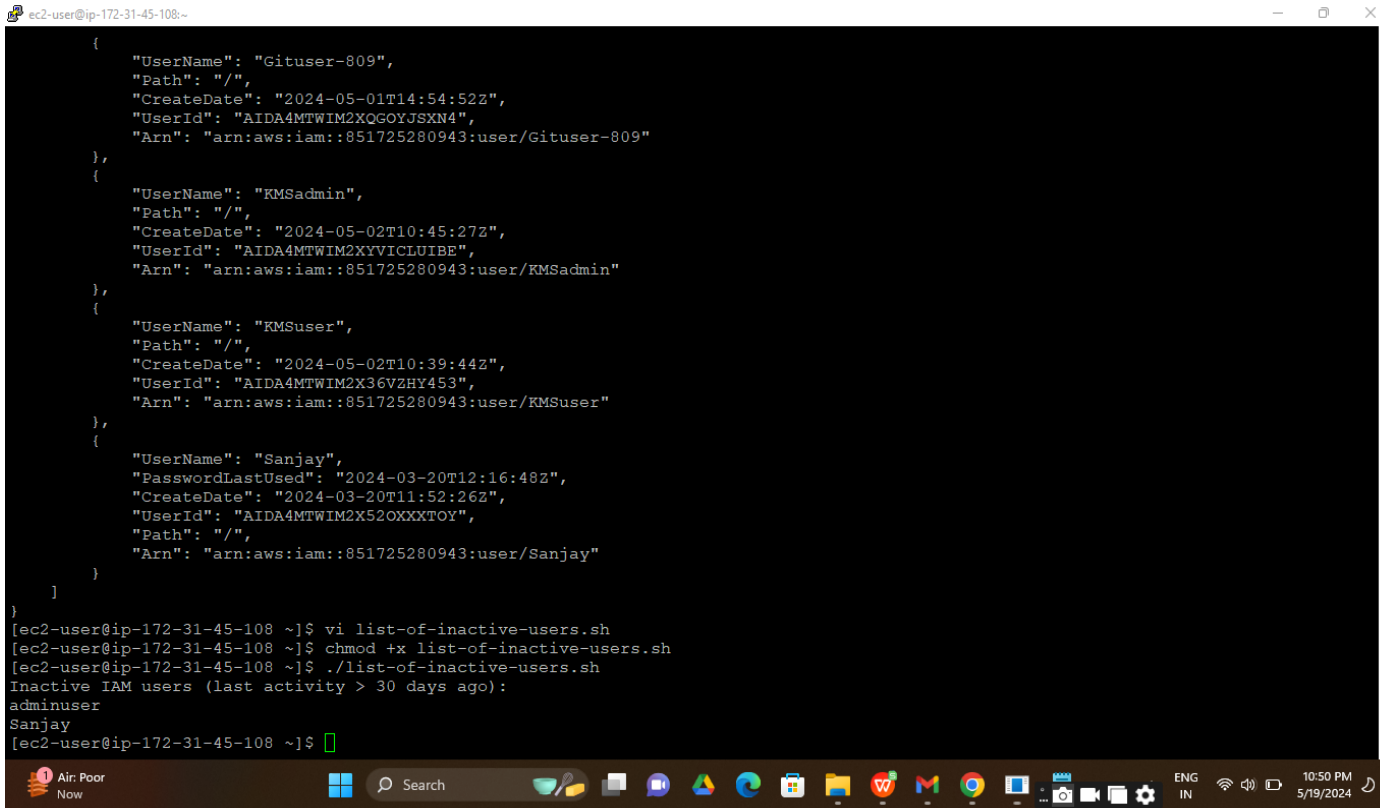
5.script executable- chmod +x list-of-inactive-users.sh

6.- ./list-of-inactive-users.sh

7. Inactive iam users (last activity > 30 days ago):




ec2-user@ip-172-31-45-108:~
current_date_seconds=$(date +%s)
IAM_USERS=$(aws iam list-users)
iso_to_seconds() {
date -d "$1" +%s
}
INACTIVE_USERS=$(echo "$IAM_USERS" | jq -r '.Users[] | @base64' | while read user; do
_jq() {
echo "${user}" | base64 --decode | jq -r "${1}"
}
user_name=$(_jq '.UserName')
password_last_used=$(_jq '.PasswordLastUsed')
create_date=$(_jq '.CreateDate')
if ["$password_last_used" != "null"]; then
last_activity_seconds=$(iso_to_seconds "$password_last_used")
else
last_activity_seconds=$(iso_to_seconds "$create_date")
fi
diff_seconds=$((current_date_seconds - last_activity_seconds))
if [$diff_seconds -gt 2592000]; then
echo "$user_name"
fi
done)
echo "Inactive IAM users (last activity > 30 days ago):"
echo "$INACTIVE_USERS"
34,1 66%
81°F Cloudy
Search
ENG IN
10:48 PM 5/19/2024


ec2-user@ip-172-31-45-108:~
{
"UserName": "Gituser-809",
"Path": "/",
"CreateDate": "2024-05-01T14:54:52Z",
"UserId": "AIDA4MTWIM2XQGOYJSXN4",
"Arn": "arn:aws:iam::851725280943:user/Gituser-809"
},
{
"UserName": "KMSadmin",
"Path": "/",
"CreateDate": "2024-05-02T10:45:27Z",
"UserId": "AIDA4MTWIM2XYVICLUIBE",
"Arn": "arn:aws:iam::851725280943:user/KMSadmin"
},
{
"UserName": "KMSuser",
"Path": "/",
"CreateDate": "2024-05-02T10:39:44Z",
"UserId": "AIDA4MTWIM2X36VZHY453",
"Arn": "arn:aws:iam::851725280943:user/KMSuser"
},
{
"UserName": "Sanjay",
"PasswordLastUsed": "2024-03-20T12:16:48Z",
"CreateDate": "2024-03-20T11:52:26Z",
"UserId": "AIDA4MTWIM2X52OXXXTOY",
"Path": "/",
"Arn": "arn:aws:iam::851725280943:user/Sanjay"
}
]
}
[ec2-user@ip-172-31-45-108 ~]$ vi list-of-inactive-users.sh
[ec2-user@ip-172-31-45-108 ~]$ chmod +x list-of-inactive-users.sh
[ec2-user@ip-172-31-45-108 ~]$ ./list-of-inactive-users.sh
Inactive IAM users (last activity > 30 days ago):
adminuser
Sanjay
[ec2-user@ip-172-31-45-108 ~]$
Air: Poor Now
Search
ENG IN
10:50 PM 5/19/2024

C. I have find that in my VPC user has checked flow logs.

1. Run- sudo yum update apply all updates.

2. configure user- aws configure

3.- aws ec2 describe-vpcs this command in Linux lists detailed information about all the Virtual Private Clouds (VPCs) in our AWS account.

4. In aws account we create 1 VPC means total 2 VPCs default and manual create VPC.

5. In default VPC we not attached flow logs.

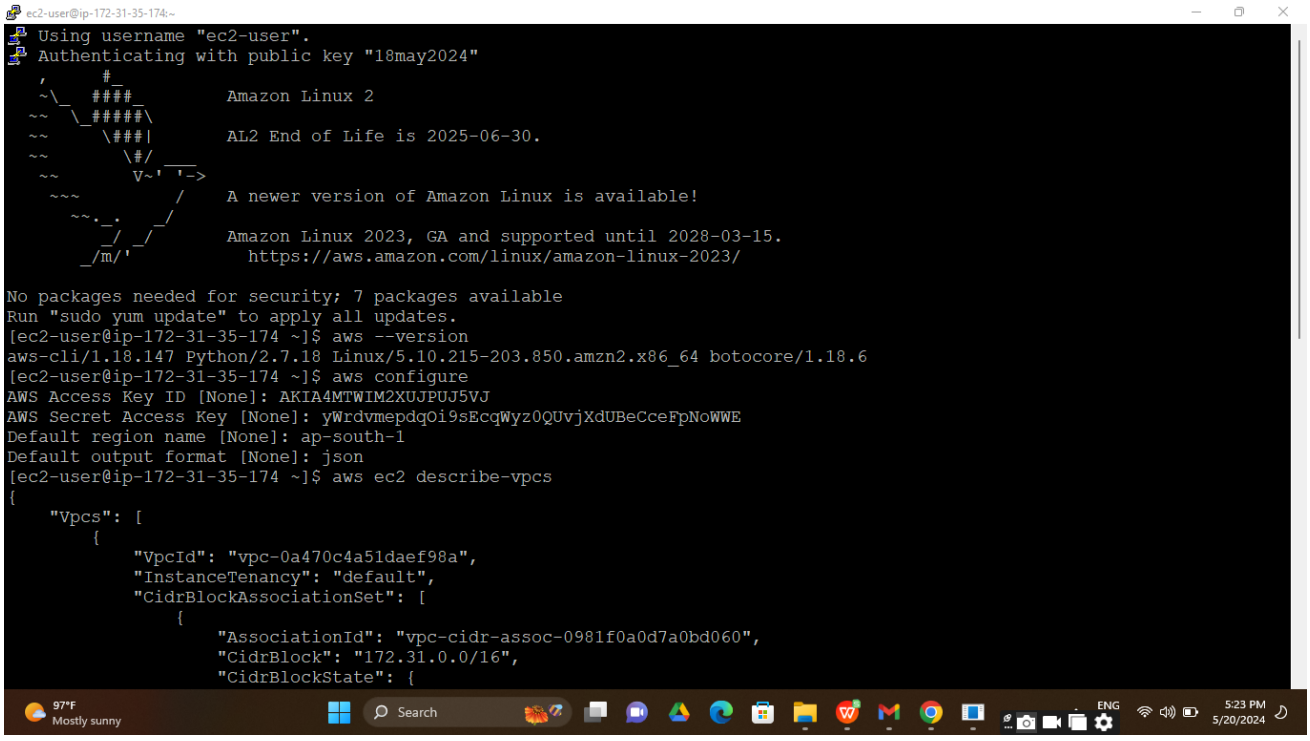
6. Manual create VPC we attached flow logs manually.

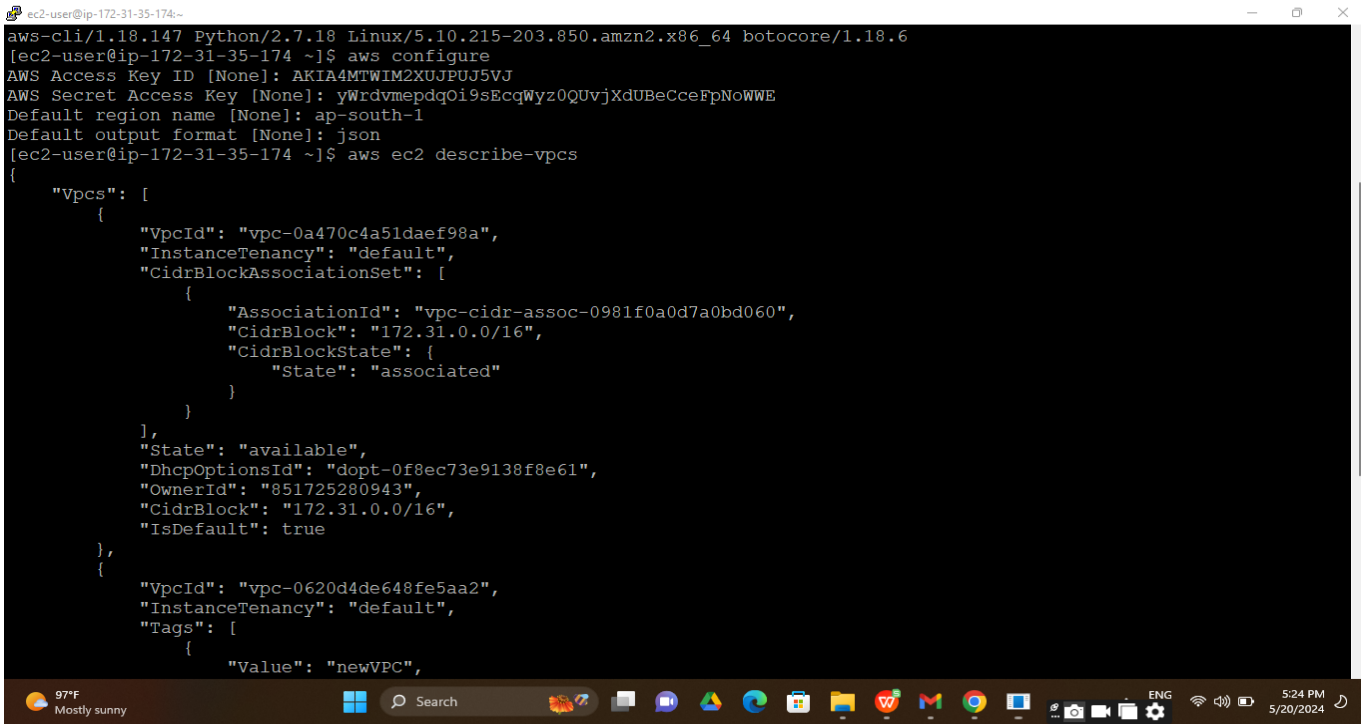
7. We write the script and find that in my VPC user has checked flow logs and see the script will run properly or not let's see.

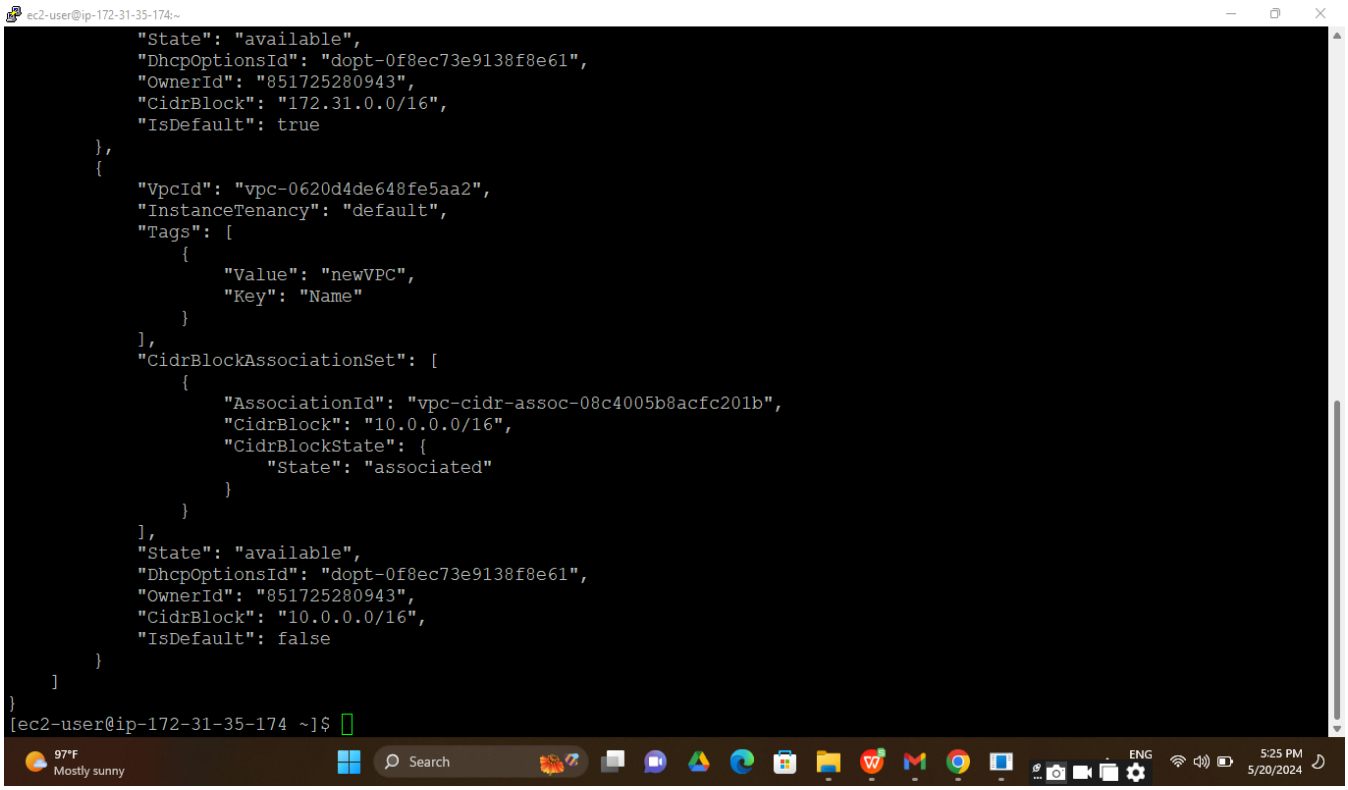
8. We write script in- check-vpc-flow-logs.sh

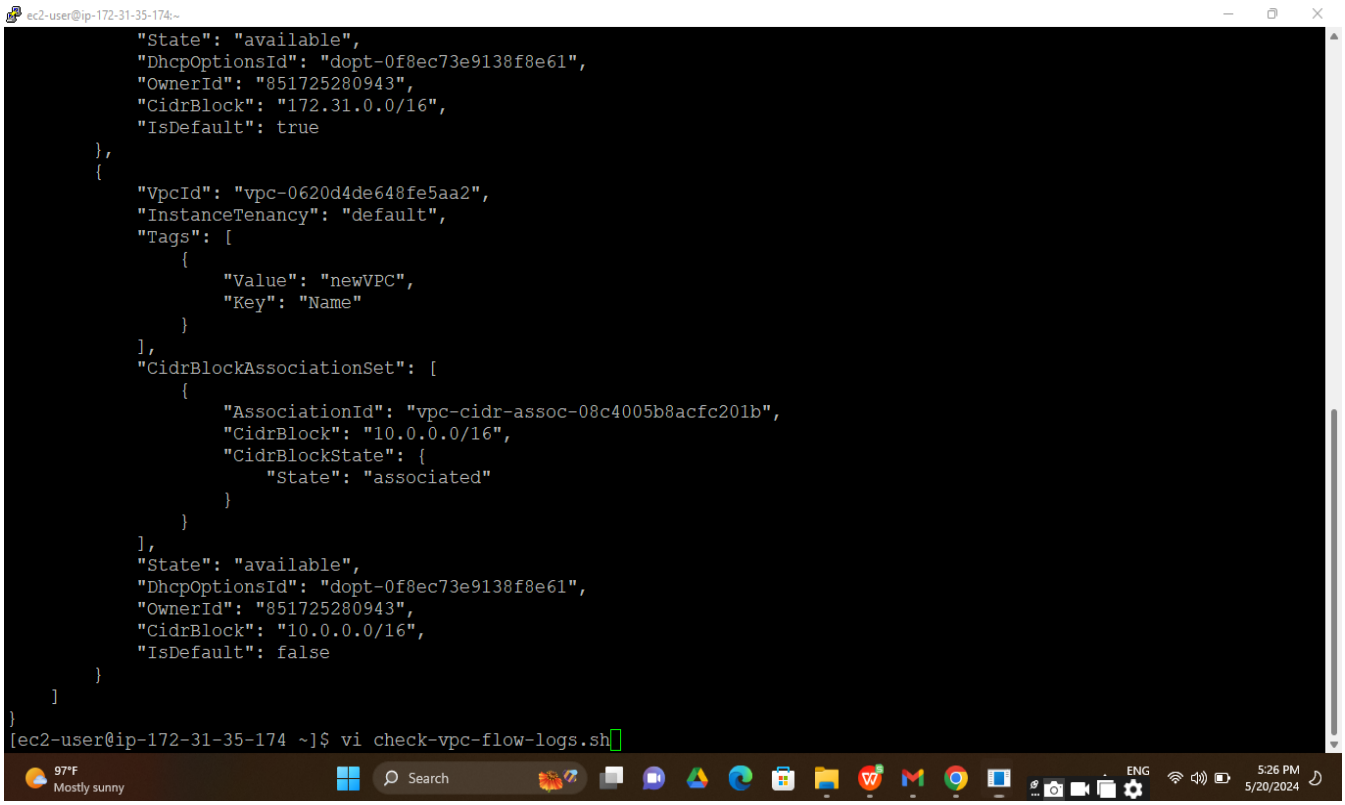
   - chomp +x check-vpc-flow-logs.sh
   - ./check-vpc-flow-logs.sh vpc-0a470c4a51daef98a (Give the id of default VPC for check flow logs attached or not)
   - No flow logs are attached in VPC vpc-0a470c4a51daef98a
   - ./check-vpc-flow-logs.sh vpc-0620d4de648fe5aa2 (Give the id of manual create VPC for check flow logs attached or not)
   - Flow logs are attached in VPC vpc-0620d4de648fe5aa2

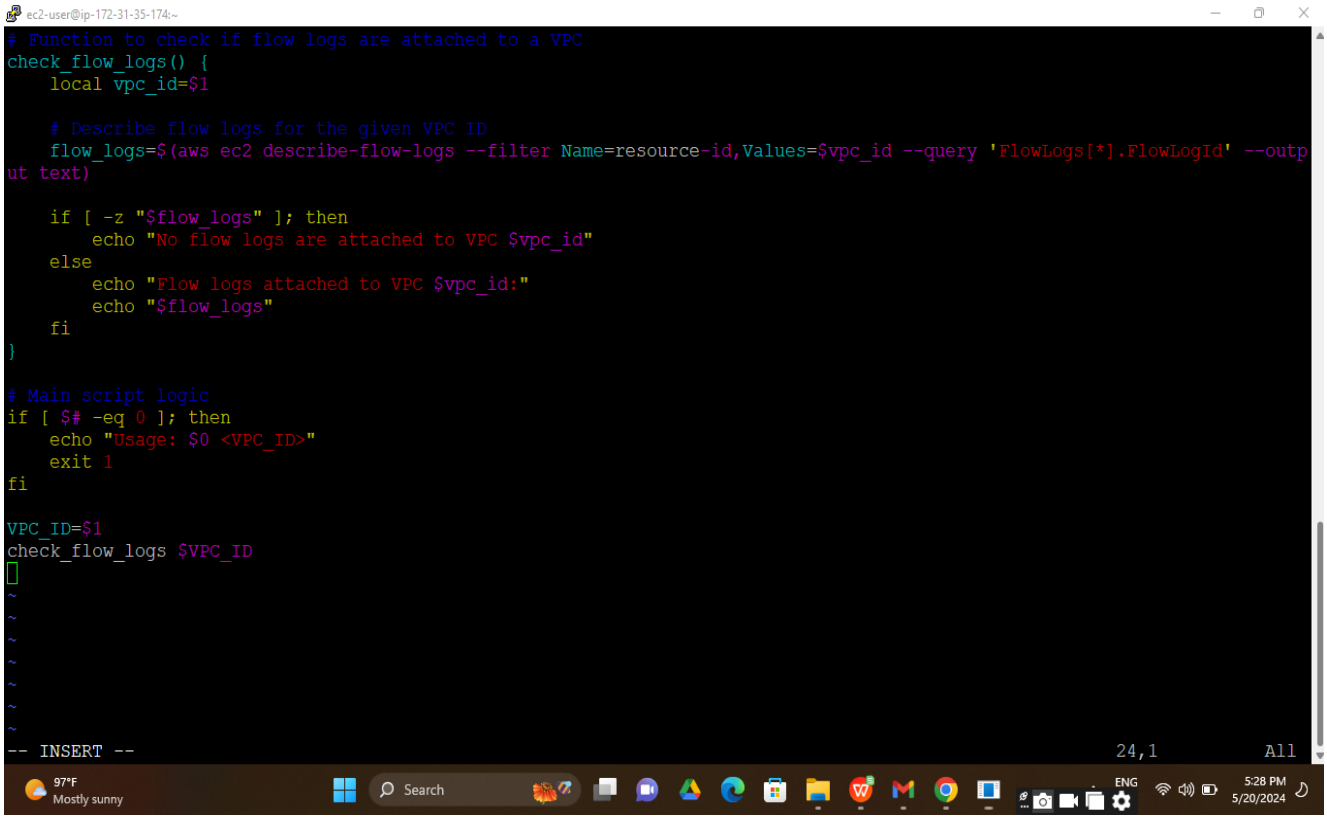
9. Script is run properly and we find in my VPC users has checked flow logs.

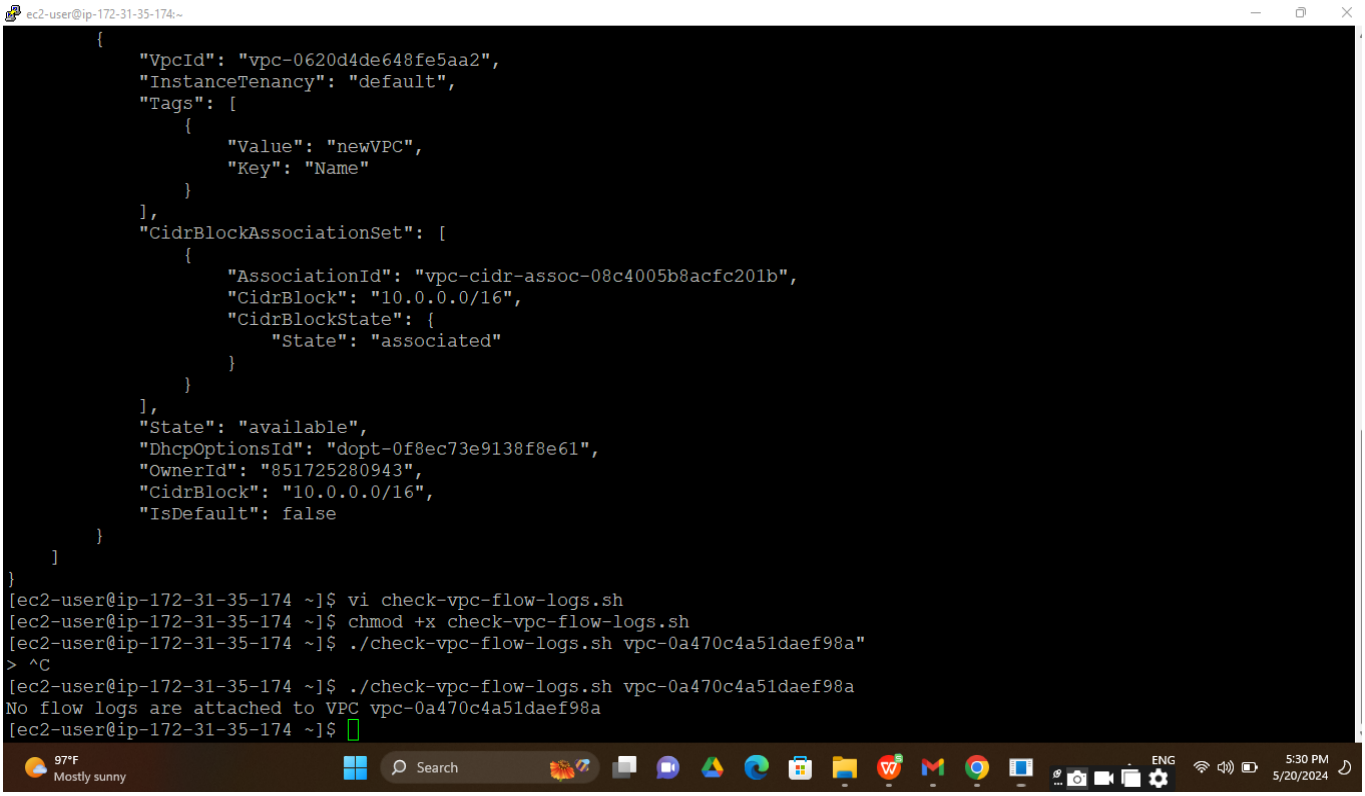


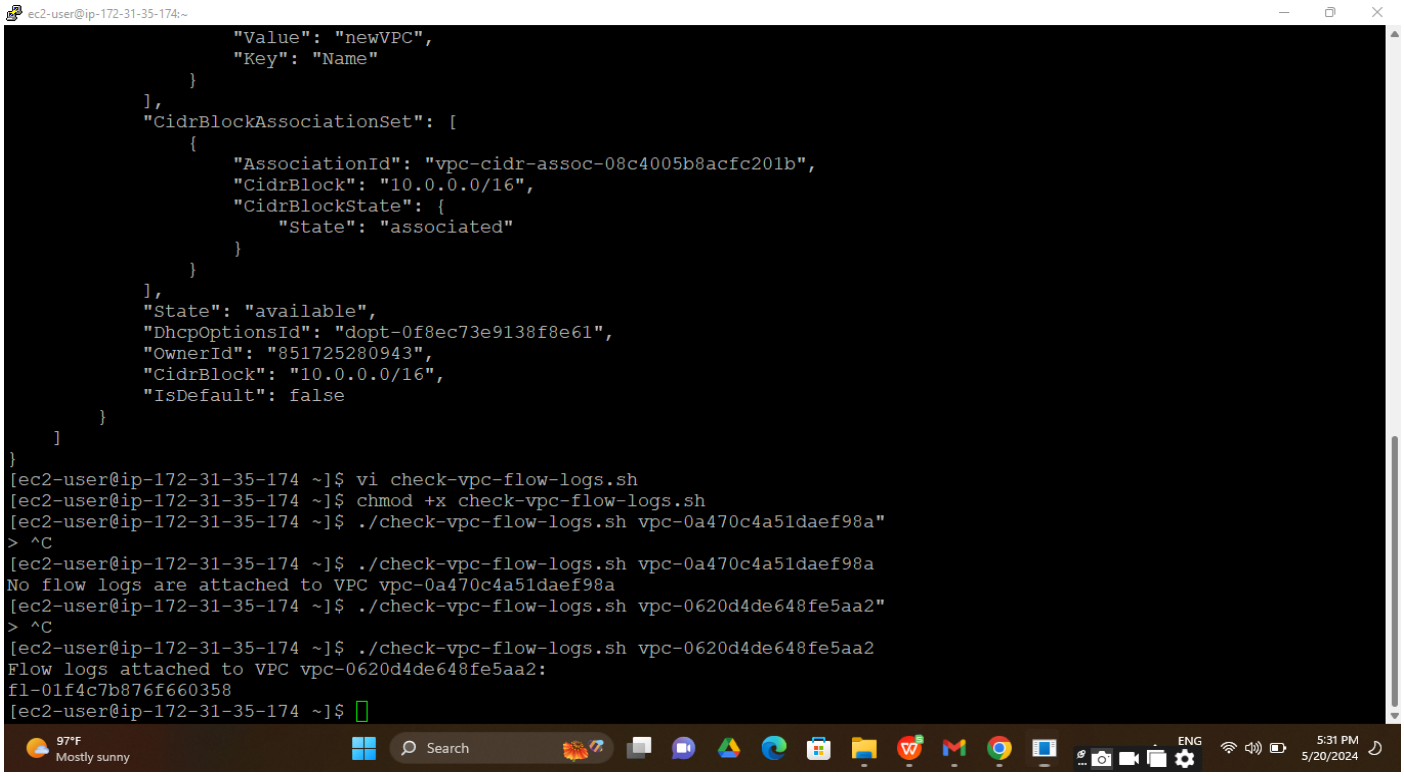

ec2-user@ip-172-31-35-174:~
aws-cli/1.18.147 Python/2.7.18 Linux/5.10.215-203.850.amzn2.x86_64 botocore/1.18.6
[ec2-user@ip-172-31-35-174 ~]$ aws configure
AWS Access Key ID [None]: AKIA4MTWIM2XUJPUJ5VJ
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]: json
[ec2-user@ip-172-31-35-174 ~]$ aws ec2 describe-vpcs
{
"Vpcs": [
{
"VpcId": "vpc-0a470c4a51daef98a",
"InstanceTenancy": "default",
"CidrBlockAssociationSet": [
{
"AssociationId": "vpc-cidr-assoc-0981f0a0d7a0bd060",
"CidrBlock": "172.31.0.0/16",
"CidrBlockState": {
"State": "associated"
}
}
],
"State": "available",
"DhcpOptionsId": "dopt-0f8ec73e9138f8e61",
"OwnerId": "851725280943",
"CidrBlock": "172.31.0.0/16",
"IsDefault": true
},
{
"VpcId": "vpc-0620d4de648fe5aa2",
"InstanceTenancy": "default",
"Tags": [
{
"Value": "newVPC",
97°F
Mostly sunny
Search
ENG
5:24 PM
5/20/2024


ec2-user@ip-172-31-35-174:~
"State": "available",
"DhcpOptionsId": "dopt-0f8ec73e9138f8e61",
"OwnerId": "851725280943",
"CidrBlock": "172.31.0.0/16",
"IsDefault": true
},
{
"VpcId": "vpc-0620d4de648fe5aa2",
"InstanceTenancy": "default",
"Tags": [
{
"Value": "newVPC",
"Key": "Name"
}
],
"CidrBlockAssociationSet": [
{
"AssociationId": "vpc-cidr-assoc-08c4005b8acfc201b",
"CidrBlock": "10.0.0.0/16",
"CidrBlockState": {
"State": "associated"
}
}
],
"State": "available",
"DhcpOptionsId": "dopt-0f8ec73e9138f8e61",
"OwnerId": "851725280943",
"CidrBlock": "10.0.0.0/16",
"IsDefault": false
}
]
}
[ec2-user@ip-172-31-35-174 ~]$
97°F
Mostly sunny
Search
ENG
5:25 PM
5/20/2024


ec2-user@ip-172-31-35-174:~
"State": "available",
"DhcpOptionsId": "dopt-0f8ec73e9138f8e61",
"OwnerId": "851725280943",
"CidrBlock": "172.31.0.0/16",
"IsDefault": true
},
{
"VpcId": "vpc-0620d4de648fe5aa2",
"InstanceTenancy": "default",
"Tags": [
{
"Value": "newVPC",
"Key": "Name"
}
],
"CidrBlockAssociationSet": [
{
"AssociationId": "vpc-cidr-assoc-08c4005b8acfc201b",
"CidrBlock": "10.0.0.0/16",
"CidrBlockState": {
"State": "associated"
}
}
],
"State": "available",
"DhcpOptionsId": "dopt-0f8ec73e9138f8e61",
"OwnerId": "851725280943",
"CidrBlock": "10.0.0.0/16",
"IsDefault": false
}
]
}
[ec2-user@ip-172-31-35-174 ~]$ vi check-vpc-flow-logs.sh
97°F
Mostly sunny
Search
ENG
5:26 PM
5/20/2024


ec2-user@ip-172-31-35-174:~
# Function to check if flow logs are attached to a VPC
check_flow_logs() {
local vpc_id=$1
# Describe flow logs for the given VPC ID
flow_logs=$(aws ec2 describe-flow-logs --filter Name=resource-id,Values=$vpc_id --query 'FlowLogs[*].FlowLogId' --outp
ut text)
if [-z "$flow_logs"]; then
echo "No flow logs are attached to VPC $vpc_id"
else
echo "Flow logs attached to VPC $vpc_id:"
echo "$flow_logs"
fi
}
# Main script logic
if [$# -eq 0]; then
echo "Usage: $0 <VPC_ID>"
exit 1
fi
VPC_ID=$1
check_flow_logs $VPC_ID
-- INSERT --
24,1
All
97°F
Mostly sunny
Search
ENG
5:28 PM
5/20/2024


ec2-user@ip-172-31-35-174:~
{
"VpcId": "vpc-0620d4de648fe5aa2",
"InstanceTenancy": "default",
"Tags": [
{
"Value": "newVPC",
"Key": "Name"
}
],
"CidrBlockAssociationSet": [
{
"AssociationId": "vpc-cidr-assoc-08c4005b8acfc201b",
"CidrBlock": "10.0.0.0/16",
"CidrBlockState": {
"State": "associated"
}
}
],
"State": "available",
"DhcpOptionsId": "dopt-0f8ec73e9138f8e61",
"OwnerId": "851725280943",
"CidrBlock": "10.0.0.0/16",
"IsDefault": false
}
]
}
[ec2-user@ip-172-31-35-174 ~]$ vi check-vpc-flow-logs.sh
[ec2-user@ip-172-31-35-174 ~]$ chmod +x check-vpc-flow-logs.sh
[ec2-user@ip-172-31-35-174 ~]$ ./check-vpc-flow-logs.sh vpc-0a470c4a51daef98a"
> ^C
[ec2-user@ip-172-31-35-174 ~]$ ./check-vpc-flow-logs.sh vpc-0a470c4a51daef98a
No flow logs are attached to VPC vpc-0a470c4a51daef98a
[ec2-user@ip-172-31-35-174 ~]$
97°F
Mostly sunny
Search
ENG
5:30 PM
5/20/2024


ec2-user@ip-172-31-35-174:~
"Value": "newVPC",
"Key": "Name"
}
],
"CidrBlockAssociationSet": [
{
"AssociationId": "vpc-cidr-assoc-08c4005b8acfc201b",
"CidrBlock": "10.0.0.0/16",
"CidrBlockState": {
"State": "associated"
}
}
],
"State": "available",
"DhcpOptionsId": "dopt-0f8ec73e9138f8e61",
"OwnerId": "851725280943",
"CidrBlock": "10.0.0.0/16",
"IsDefault": false
}
]
}
[ec2-user@ip-172-31-35-174 ~]$ vi check-vpc-flow-logs.sh
[ec2-user@ip-172-31-35-174 ~]$ chmod +x check-vpc-flow-logs.sh
[ec2-user@ip-172-31-35-174 ~]$ ./check-vpc-flow-logs.sh vpc-0a470c4a51daef98a"
> ^C
[ec2-user@ip-172-31-35-174 ~]$ ./check-vpc-flow-logs.sh vpc-0a470c4a51daef98a
No flow logs are attached to VPC vpc-0a470c4a51daef98a
[ec2-user@ip-172-31-35-174 ~]$ ./check-vpc-flow-logs.sh vpc-0620d4de648fe5aa2"
> ^C
[ec2-user@ip-172-31-35-174 ~]$ ./check-vpc-flow-logs.sh vpc-0620d4de648fe5aa2
Flow logs attached to VPC vpc-0620d4de648fe5aa2:
fl-01f4c7b876f660358
[ec2-user@ip-172-31-35-174 ~]$
97°F
Mostly sunny
Search
ENG
5:31 PM
5/20/2024