

[illegible]

<b>Name</b>	GuardDuty : IAM Findings
<b>URL</b>	<a href="https://attackdefense.com/challengedetails?cid=2478">https://attackdefense.com/challengedetails?cid=2478</a>
<b>Type</b>	AWS Cloud Security : Defense

**Important Note:** This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

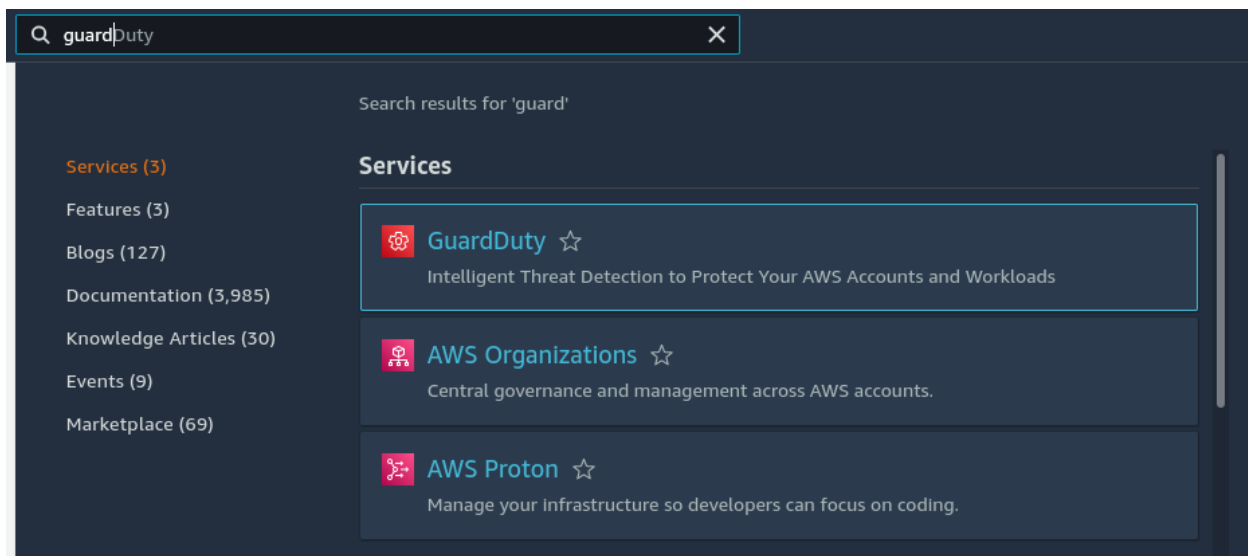
### Solution:

**Step 1:** Click the lab link button to get access credentials.

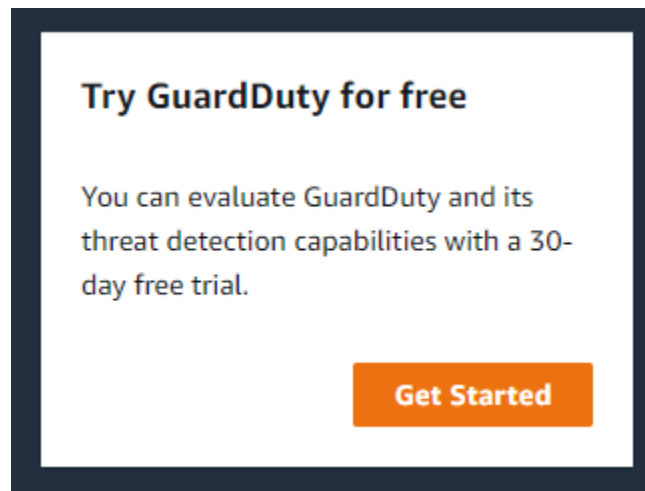
## Access Credentials to your AWS lab Account

<b>Login URL</b>	<a href="https://029911053369.signin.aws.amazon.com/console">https://029911053369.signin.aws.amazon.com/console</a>
<b>Region</b>	US East (N. Virginia) us-east-1
<b>Username</b>	student
<b>Password</b>	Ad0jwjpEsZaLFsKx
<b>Access Key ID</b>	AKIAQN5WWOQ42ZAKYJV6
<b>Secret Access Key</b>	ZOmZa+CJfpO8QAWHFHrjHk3EU1wbsjJd9dJZ4Q2CE

**Step 2:** Enable GuardDuty from the console. Search for GuardDuty in the search bar and navigate to the GuardDuty dashboard.



**Step 3:** Click on Get Started.



**Step 4:** Click on Enable GuardDuty.

data, events, or logs available to you. You can configure

in a 30 day [GuardDuty free trial](#). [Learn more](#)

**Enable GuardDuty**

There will not be any findings at first.



Resource



**You don't have any findings.**

GuardDuty continuously monitors your AWS environment and reports findings on this page.

[Learn more](#)

**Step 5:** Configure AWS CLI with given credentials.

**Command:** aws configure

```
(kali㉿kali)-[~/lab]
$ aws configure
AWS Access Key ID [*****QSPA]: AKIAQN5WWOQ42ZAKYJV6
AWS Secret Access Key [*****i9tf]: ZOmZa+CJfp08QAWHFrjHk3EU1wbsjJd9dJJZ4Q2CE
Default region name [us-east-1]:
Default output format [None]:
```

**Step 6:** Interact with AWS account using various commands. Change the AWS account password policy to a weak policy.

**Command:** aws iam update-account-password-policy --no-require-symbols  
--no-require-numbers --no-require-uppercase-characters --no-require-lowercase-characters

```
(kali㉿kali)-[~/lab]
$ aws iam update-account-password-policy --no-require-symbols
--no-require-numbers --no-require-uppercase-characters --no-re
quire-lowercase-characters
```

Create an access key for this account.

**Command:** aws iam create-access-key --user-name student

```
(kali㉿kali)-[~/lab]
$ aws iam create-access-key --user-name student
{
  "AccessKey": {
    "UserName": "student",
    "AccessKeyId": "AKIAQN5WWOQ4Y5GUBER7",
    "Status": "Active",
    "SecretAccessKey": "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",
    "CreateDate": "2022-08-29T17:45:53+00:00"
  }
}
```

Update the username of this account to the name "Bob".

**Command:** aws iam update-user --user-name student --new-user-name Bob

```
(kali㉿kali)-[~/lab]
$ aws iam update-user --user-name student --new-user-name Bob
```

Revert back the name “Bob” to “student”.

**Command:** aws iam update-user --user-name Bob --new-user-name student

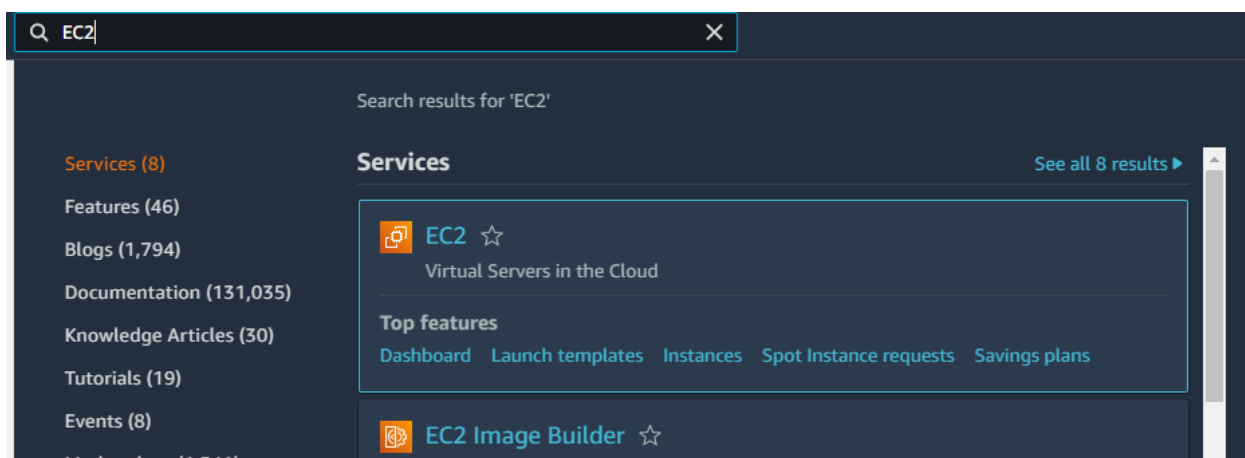
```
(kali㉿kali)-[~/lab]
$ aws iam update-user --user-name Bob --new-user-name student
```

Try to list the access keys of this account.

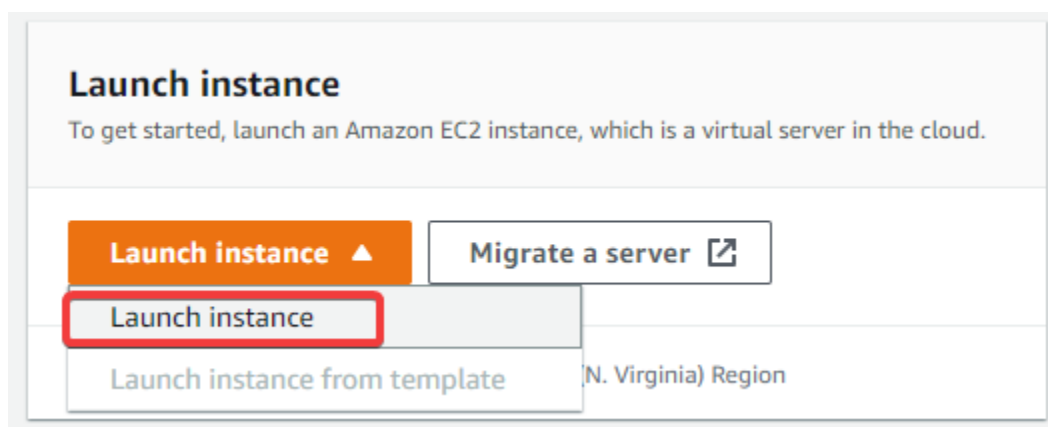
**Command:** aws iam list-access-keys --user-name student

```
(kali㉿kali)-[~/lab]
$ aws iam list-access-keys --user-name student
{
  "AccessKeyMetadata": [
    {
      "UserName": "student",
      "AccessKeyId": "AKIAQN5WWOQ42ZAKYJV6",
      "Status": "Active",
      "CreateDate": "2022-08-17T23:17:42+00:00"
    },
    {
      "UserName": "student",
      "AccessKeyId": "AKIAQN5WWOQ4Y5GUBER7",
      "Status": "Active",
      "CreateDate": "2022-08-29T17:45:53+00:00"
    }
  ]
}
```

**Step 7:** Get back to the AWS console and search for “EC2” and navigate to EC2 dashboard.



**Step 8:** Create a new instance with an instance profile. Launch a new instance by clicking “Launch instance”.



**Step 9:** Enter “lab-instance” as name and “Amazon Linux” as AMI.

The Amazon Linux AMI is a supported and maintained Linux image provided by Amazon Web Services for use on Amazon Elastic Compute Cloud (Amazon EC2).



Name and tags

Info

Name

lab-instance

Add additional tags

▼ Application and OS Images (Amazon Machine Image)

Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Q Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

S

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) – Kernel 5.10, SSD Volume Type

ami-05fa00d4c63e32376 (64-bit (x86)) / ami-05f3141013eebdc12 (64-bit (Arm))

Virtualization: hvm    ENA enabled: true    Root device type: ebs

Free tier eligible

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20220805.0 x86\_64 HVM gp2

Architecture

AMI ID

64-bit (x86)

ami-05fa00d4c63e32376

Verified provider

**Step 10:** Select “Proceed without a key pair” and select “Create security group”. By default these settings will allow port 22 so that we can connect with the instance.



### ▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Proceed without a key pair (Not recommended)

Default value ▼

 [Create new key pair](#)

### ▼ Network settings [Get guidance](#)

[Edit](#)

Network [Info](#)

vpc-091769248e7e4ffb0

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group

☐ Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:

☒ Allow SSH traffic from  
Helps you connect to your instance

Anywhere  
0.0.0.0/0 ▼

**Step 11:** Click on “Advanced details”.

[Add new volume](#)

0 x File systems

► **Advanced details** [Info](#)

**Step 12:** Enable the Metadata for the instance. Choose “V1 and V2(token optional)” as the Metadata version.

Instance Metadata Service (IMDS) provides data about your instance that you can use to configure or manage the running instance. Instance metadata is divided into categories, for example, host name, events, and security groups. The temporary access credentials can be retrieved using the metadata.



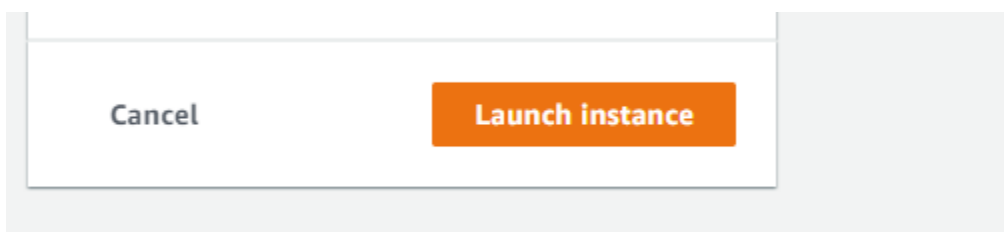
Metadata accessible [Info](#)

Enabled ▼

Metadata version [Info](#)

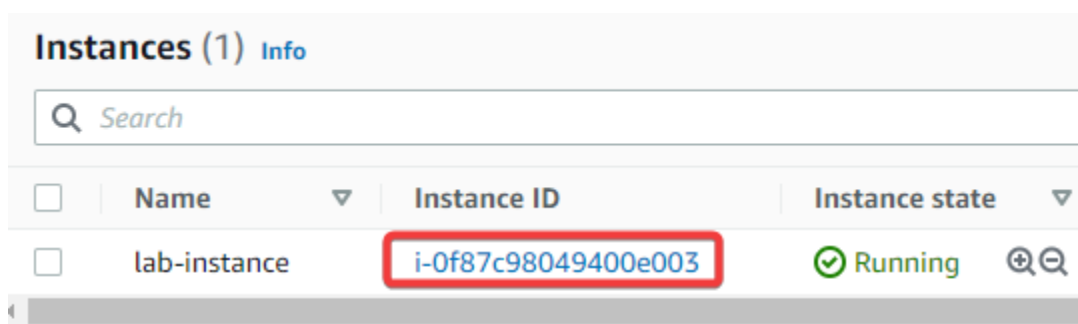
V1 and V2 (token optional) ▼

Click on “Launch instance”.



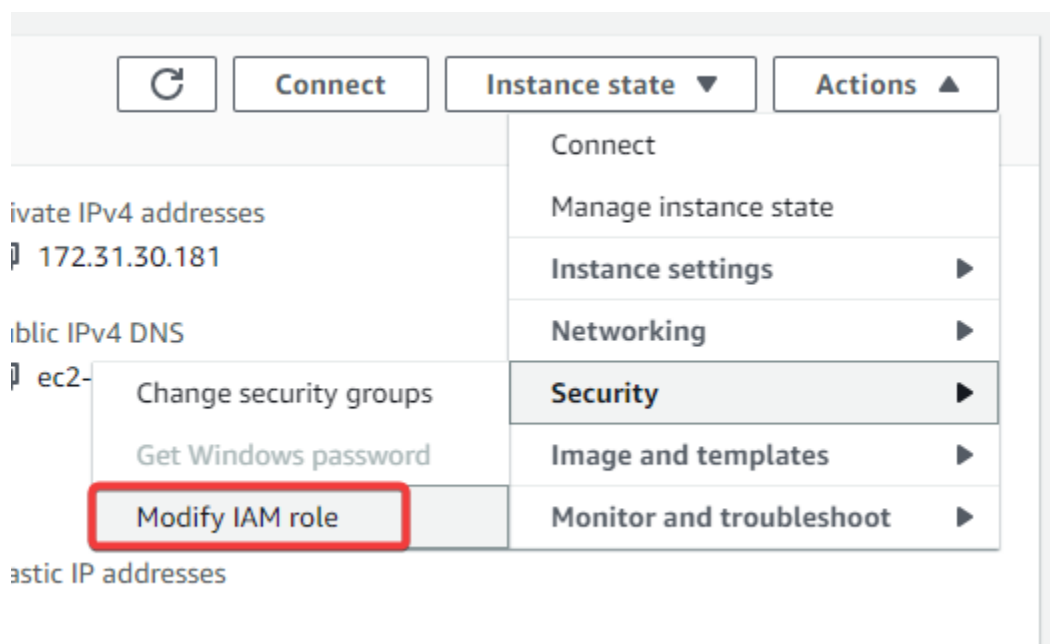
Cancel Launch instance

**Step 13:** Wait until the instance state changes to “Running”. After that click on the instance id.

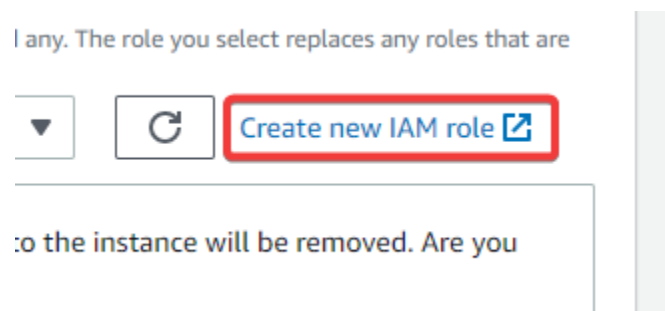


Instances (1) <a href="#">Info</a>			
<input type="text" value="Search"/>			
<input type="checkbox"/>	Name ▼	Instance ID	Instance state ▼
<input type="checkbox"/>	lab-instance	i-0f87c98049400e003	✓ Running 🔍

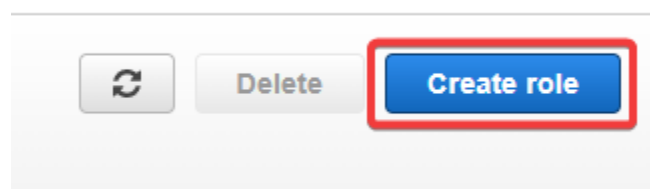
**Step 14:** Click on “Modify IAM role”.



**Step 15:** Click on “Create new IAM role”.



**Step 16:** Click on “Create role”.



**Step 17:** Select Trusted entity type as “AWS service” and use case as “EC2”.

☒ **AWS service**

Allow AWS services like EC2, Lambda, or others to perform actions in this account.

☐ **AWS account**

Allow entities in or a 3rd party to

☐ **SAML 2.0 federation**

Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

☐ **Custom trust**

Create a custom trust relationship to allow actions in this account.

### Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

#### 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.

**Step 18:** Search for “s3full” and select “AmazonS3FullAccess” policy and click “Next”.

**Permissions policies (766)**  
Choose one or more policies to attach to your new role.

"s3full" X

Clear filters

<input type="checkbox"/>	Policy name	Type	Description
<input type="checkbox"/>	AmazonS3FullAccess	AWS m...	Provides full acc...

**Step 19:** Set role name as “instance\_role” and click on “Create role”.

## Role details

### Role name

Enter a meaningful name to identify this role.

instance\_role

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

### Description

Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

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

**Step 20:** Navigate back to the previous tab and attach a role with the EC2 instance. Click on the refresh button to load the new role.

You haven't created any. The role you select replaces any role currently attached to the instance.

 [Create new IAM role](#)


Any role currently attached to the instance will be removed. Any role currently attached to the instance will be removed.

Now, click on "Update IAM role".

Attach an IAM role to your instance.


Instance ID  
i-0f87c98049400e003 (lab-instance)

IAM role  
Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

instance\_role ▼  [Create new IAM role](#)

Cancel **Update IAM role**

**Step 21:** From the EC2 instance listing page , Select EC2 instance and click on “Connect”.

 **Connect** Instance state ▼

Status	Availability Zone ▼	Public IPv4 DNS
ns +	us-east-1a	ec2-54-160-156

Use EC2 instance connect and click on “Connect”.

EC2 > Instances > i-0f87c98049400e003 > Connect to instance

### Connect to instance [Info](#)

Connect to your instance i-0f87c98049400e003 (lab-instance) using any of these options

**EC2 Instance Connect**

Session Manager

SSH client

EC2 serial console

Instance ID

 i-0f87c98049400e003 (lab-instance)


Public IP address

 54.160.156.29

User name

ec2-user

Connect using a custom user name, or use the default user name ec2-user for the AMI used to launch the instance.

 **Note:** In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel

Connect

**Step 22:** Retrieve temporary credentials using metadata services version 1.

**Command:** `curl http://169.254.169.254/latest/meta-data/iam/security-credentials/instance_role`

```
[ec2-user@ip-172-31-30-181 ~]$ curl http://169.254.169.254/latest/meta-data/iam/security-credentials/instance_role
{
  "Code" : "Success",
  "LastUpdated" : "2022-08-29T17:55:04Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "ASIAQN5WWOQ42ET7C27U",
  "SecretAccessKey" : "Q/QBd9EoGVGTWr2zzQPvYjjDYmziW0liTNpze6+4",
  "Token" : "IQoJb3JpZ2luX2VjEKr////////wEaCXVzLWVhc3QtMSJHMEUCIQDK0xfzobwL2g17Jb22uf645ESNxm6JGMI6mCpNsc4YgIgMwW0NexmxI
5MTewNTMzNjkiDNWZmgGeTkzyAc2M3yqvBOuVelCyev9nASyi97ZS+SVdxWTRe/g8DXtQ6S6prKKL6+c+jCURooVBhoJ8nR3RrLl667xMbAKVMo3pzGW7Oekonv
ijYDe7Uis3eMkOwyNbBAC++hz0ShJtqMAIZKFKenvLi6MuQlmuF+a5Unn7DhBxty7+18KTczxWHEWVMrL9ORF/qWt5E+3AHukP30dgdSmVmQbpJLpS4WlbqMHZ
n7ckWRMzrYhTutRW7rF28yDtbjKsBuIXyI3UGpU1tb23BdbCARBgFHSNI6UaV7mcGWHaj4cPq9NDIrdXrVE00zLc1Af1sxK1xf1i8cBV06ICiM5oV/lnPVoBy5y
3eId5cPjNdWp0mEBTQCC78Z+uzoORLlDiUGjiY0aTMuYJTnyjgaDV7NfoabORhJ1YkBu5BrPNeOGcvbNlM9KZp5Dn0oWOCpH3pI2bd5b+aZydI2xbqXXvdTcPC
ZONen4FTVmgXNKdg+UvNMeIi8tHPeaflEE/XAYUHmyrNG8PsINHqaV2/R8ksjfsOUC6dRll7qsEroModBlk4LOCl+Qvgw4PqzmAY6qQEpkj7cbkDWK/3qnT84n
1PPxRRtFRUCUOMssJCwXMDTPx8qy66Dd0ra851YjNc7FAMlbdgLC50c3fuaobVqIQDPgY9dUX/jIf58AWrujhyHOMaHUva9SpurlTOR9M+B/pg42MHPth9KfvH3
"Expiration" : "2022-08-30T00:29:40Z"
}
[ec2-user@ip-172-31-30-181 ~]$
```



**Step 23:** Set the required environment variable to allow AWS CLI to use the temporary access credentials. AWS CLI prioritizes the environment variable over the stored credentials.

**Command:**

```
export AWS_ACCESS_KEY_ID=ASIAQN5WWOQ42ET7C27U
export AWS_SECRET_ACCESS_KEY=Q/QBd9EoGVGTWr2zZQPvYjjDYmzIW01iTNpze6+4
export
AWS_SESSION_TOKEN=IQoJb3JpZ2luX2VjEKR////////wEaCXVzLWVhc3QtMSJHMEUCIQDK0xfzobwL2g17Jb22uf6
45ESNxm6JGMI6mCpNsc4YgIgMwW0NexmxEQ3SS0hrFtP+H9+nKhgj/yeSn0kLhOBblgq0gQIMxAAGgwwMjk5MT
EwNTMzNjkiDNWZmgGeTkzyAc2M3yqvBOuVelCyeV9nASyi97ZS+SVdxWTRe/g8DXtQ6S6prKKL6+c+jCURooVBho
J8nR3RrLI667xMbAKVMo3pzGW7OekonwHxWRjyUkPcPI7HcxiQrzQaghl6l0WsPG1u1aGvJla/DbN3b7fijYDe7UiS3e
MkOwyNbBAC++hz0ShJtqMAIZKFkEnvLi6MuQlmuF+a5Unn7DhBXty7+18KTczxWHEWVMrL9ORF/qWt5E+3AHukP
3OdqdSmVmQbpJLpS4WlbqMHZRMrGP/KBlvR9bo5b+9B8p5NGPYF2twJh6XXmav8BD1oJDwyEEHn7ckWRMzrYh
TutRW7rF28yDtjbKsBulXyl3UGpU1tb23BdbCARBgFHSNI6UaV7mcGWHaj4cPq9NDIrdXrVEO0zLc1Af1sxK1xf1i8cB
VO6ICiM5oV/lnPVoBy5yKSTiqSYxWz3jjDZfy/8TKSKbH2q9WVGKmANKjwADU8C0+Yz+w3eld5cPjNdWp0mEBTQC
C78Z+uzoORLIDiuGjiY0aTMuYJTnyjgaDV7NfoabORhJ1YkBu5BrPNeOGcvbNIM9KZp5Dn0oWOCpH3pl2bd5b+aZy
dl2xbqXXvdTcPC13DsYrZIJDKNk11KvUMPZNNplo5Ai626oue29FwCSw6s098HI7ZoNEn4FTVmgXNKdq+UvNMeli8t
HPeafIEE/XAYUHmyrNG8PsINHqaV2/R8ksjfsOUC6dRll7qsEroModBlk4LOCl+Qvgw4PqzmAY6qQEpkj7cbkDWK/3
qnT84mCRUC9sg/IO0w3Xy0AUSAJmayoLAgA5FYeYhSQYzu5S6OxmK0/1PPxRRtFRUCUOMssJCwXMDTPx8qy66
Dd0ra85lYjNc7FAMLbdgLC5Oc3fuaobVqlQDPgY9dUX/jlf58AWruijyHOMaHUva9SpurITOR9M+B/pg42MHPth9KfvH
36ODA++sNXD74xj92fZ5PfPWD/WMTVFZeYf
```

```
(kali㉿kali)-[~/lab]
$ export AWS_ACCESS_KEY_ID=ASIAQN5WWOQ42ET7C27U
export AWS_SECRET_ACCESS_KEY=Q/QBd9EoGVGTWr2zZQPvYjjDYmzIW01iTNpze6+4
export AWS_SESSION_TOKEN=IQoJb3JpZ2luX2VjEKR////////wEaCXVzLWVhc3QtMSJHMEUCIQDK0xfzobwL2g17Jb22uf6
45ESNxm6JGMI6mCpNsc4YgIgMwW0NexmxEQ3SS0hrFtP+H9+nKhgj/yeSn0kLhOBblgq0gQIMxAAGgwwMjk5MT
EwNTMzNjkiDNWZmgGeTkzyAc2M3yqvBOuVelCyeV9nASyi97ZS+SVdxWTRe/g8DXtQ6S6prKKL6+c+jCURooVBho
J8nR3RrLI667xMbAKVMo3pzGW7OekonwHxWRjyUkPcPI7HcxiQrzQaghl6l0WsPG1u1aGvJla/DbN3b7fijYDe7UiS3e
MkOwyNbBAC++hz0ShJtqMAIZKFkEnvLi6MuQlmuF+a5Unn7DhBXty7+18KTczxWHEWVMrL9ORF/qWt5E+3AHukP
3OdqdSmVmQbpJLpS4WlbqMHZRMrGP/KBlvR9bo5b+9B8p5NGPYF2twJh6XXmav8BD1oJDwyEEHn7ckWRMzrYh
TutRW7rF28yDtjbKsBulXyl3UGpU1tb23BdbCARBgFHSNI6UaV7mcGWHaj4cPq9NDIrdXrVEO0zLc1Af1sxK1xf1i8cB
VO6ICiM5oV/lnPVoBy5yKSTiqSYxWz3jjDZfy/8TKSKbH2q9WVGKmANKjwADU8C0+Yz+w3eld5cPjNdWp0mEBTQC
C78Z+uzoORLIDiuGjiY0aTMuYJTnyjgaDV7NfoabORhJ1YkBu5BrPNeOGcvbNIM9KZp5Dn0oWOCpH3pl2bd5b+aZy
dl2xbqXXvdTcPC13DsYrZIJDKNk11KvUMPZNNplo5Ai626oue29FwCSw6s098HI7ZoNEn4FTVmgXNKdq+UvNMeli8t
HPeafIEE/XAYUHmyrNG8PsINHqaV2/R8ksjfsOUC6dRll7qsEroModBlk4LOCl+Qvgw4PqzmAY6qQEpkj7cbkDWK/3
qnT84mCRUC9sg/IO0w3Xy0AUSAJmayoLAgA5FYeYhSQYzu5S6OxmK0/1PPxRRtFRUCUOMssJCwXMDTPx8qy66
Dd0ra85lYjNc7FAMLbdgLC5Oc3fuaobVqlQDPgY9dUX/jlf58AWruijyHOMaHUva9SpurITOR9M+B/pg42MHPth9KfvH
36ODA++sNXD74xj92fZ5PfPWD/WMTVFZeYf
```

**Step 24:** Check the caller identity.

**Command:** aws sts get-caller-identity

```
(kali㉿kali)-[~/lab]
$ aws sts get-caller-identity
{
  "UserId": "AROAQN5WWOQ4UDCIOSWRD:i-0f87c98049400e003",
  "Account": "029911053369",
  "Arn": "arn:aws:sts::029911053369:assumed-role/instance_role/i-0f87c98049400e003"
}
```

**Step 25:** Create an S3 bucket. Append account id with the name to make it unique.

**Command:**

```
aws s3api create-bucket \
  --bucket lab-bucket-029911053369 \
  --region us-east-1
```

```
(kali㉿kali)-[~/lab]
$ aws s3api create-bucket \
  --bucket lab-bucket-029911053369 \
  --region us-east-1
{
  "Location": "/lab-bucket-029911053369"
}
```

**Step 26:** Fetch your public ip address.

**Command:** echo \$(curl -s https://api.ipify.org)

```
(kali㉿kali)-[~/lab]
$ echo $(curl -s https://api.ipify.org)
137.97.83.9
```

**Step 27:** Write this IP address to a plain text file and save it as "ip\_list.txt" .

**Command:** echo "137.97.83.9" > ip\_list.txt

```
(kali㉿kali)-[~/lab]
$ echo "137.97.83.9" > ip_list.txt
```

**Step 28:** Upload the file “ip\_list.txt” to the created bucket.

**Command:** `aws s3 cp ip_list.txt s3://lab-bucket-029911053369/`

```
(kali㉿kali)-[~/lab]
$ aws s3 cp ip_list.txt s3://lab-bucket-029911053369/
upload: ./ip_list.txt to s3://lab-bucket-029911053369/ip_list.txt
```

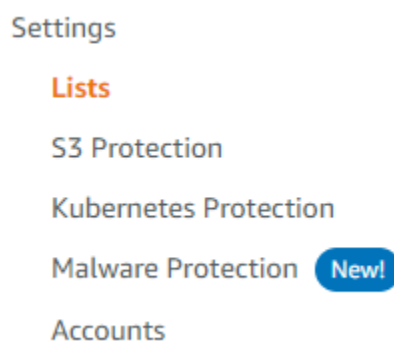
We got an S3 URI as an output. Convert the URI to an object URL using this syntax.

**Syntax:** `https://<bucket-name>.s3.amazonaws.com/<object or key name>`

So finally the object URL will be similar to the following one.

**Object URL:** `https://lab-bucket-029911053369.s3.amazonaws.com/ip_list.txt`

**Step 29:** Navigate back to GuardDuty dashboard and click on lists.



**Step 30:** Click on “Add a threat IP list”.

Threat lists consist of known malicious IP addresses. These lists can be supplied by third party threat intelligence or created specifically for your organization. GuardDuty generates findings based on threat lists. You can include a maximum of 250,000 IP addresses and CIDR ranges in a single threat list. GuardDuty only generates findings based on activity that involves IP addresses and CIDR ranges in your threat lists, findings will not be generated based on domain names. At any given time, you can have up to six uploaded threat lists per AWS account per Region.

### Threat IP lists

Threat IP lists consist of known malicious IP addresses. GuardDuty gei

Add a threat IP list

List name	List file URL
<div><div></div><div><div>Threat IP lists</div><div>Threat IP lists consist of known malicious IP addresses. Gua</div><div><a href="#">Learn more</a></div></div></div>	

Enter the “List name” as “ip\_list”. Paste object URL in the “Location” field and choose “Plaintext” as format

### Add a threat IP list



GuardDuty generates findings based on threat IP lists. [Learn more](#)

List name

ip\_list

Location

https://lab-bucket-029911053369.s3.amazonaws.com/ip\_list

Format

Plaintext

Use TXT for files that contain simple IP lists.

By adding this list, you accept and agree to the GuardDuty service terms, including those related to third-party threat intelligence, and direct GuardDuty to read data from this resource.

☒ I agree

Cancel

Add list

**Step 31:** Make the created threat IP active.

for IP addresses that are included in threat IP lists.

	Format	Active	
/ip_lis...	TXT	<input checked="" type="checkbox"/>	

**Step 32:** Interact with the resources so that it could make a finding. List S3 bucket.

**Command:** aws s3api list-objects --bucket lab-bucket-029911053369 --query 'Contents[].{Key: Key, Size: Size}'

```
(kali㉿kali)-[~/lab]
$ aws s3api list-objects --bucket lab-bucket-029911053369 --query 'Contents[].{Key: Key, Size: Size}'
[
  {
    "Key": "ip_list.txt",
    "Size": 12
  }
]
```

**Step 33:** Download “ip\_list.txt” file.

**Command:** aws s3api get-object --bucket lab-bucket-029911053369 --key ip\_list.txt iplist.txt

```
(kali㉿kali)-[~/lab]
$ aws s3api get-object --bucket lab-bucket-029911053369 --key ip_list.txt iplist.txt
{
  "AcceptRanges": "bytes",
  "LastModified": "2022-08-29T18:01:20+00:00",
  "ContentLength": 12,
  "ETag": "\"61cdc372de5606a98e372d2b4cf3d865\"",
  "ContentType": "text/plain",
  "Metadata": {}
}
```

Navigate back to the GuardDuty dashboard and refresh the page. You will find findings similar to this.

**Note:** Listing all the findings might take some time.



▼	Finding type	▼	Resource
🔍	UnauthorizedAccess:IAMUser/MaliciousIPCaller.Custom		student: <a href="#">ASIAQN5WWOQ4TTTVR4CD</a>
🔍	Recon:IAMUser/MaliciousIPCaller.Custom		student: <a href="#">ASIAQN5WWOQ4TTTVR4CD</a>
⚠️	UnauthorizedAccess:IAMUser/InstanceCredentialExfiltration.OutsideAWS		S3 Bucket: <a href="#">lab-bucket-029911053369</a>
🔍	PenTest:IAMUser/KaliLinux		instance_role: <a href="#">ASIAQN5WWOQ42ET7C27U</a>
⚠️	UnauthorizedAccess:IAMUser/InstanceCredentialExfiltration.OutsideAWS		instance_role: <a href="#">ASIAQN5WWOQ42ET7C27U</a>
🔍	PenTest:IAMUser/KaliLinux		student: <a href="#">AKIAQN5WWOQ42ZAKYJV6</a>

**Step 34:** Click on “UnauthorizedAccess:IAMUser/MaliciousIPCaller.Custom”.

This finding informs you that an API operation was invoked from an IP address that is included on a threat list that you uploaded. In , a threat list consists of known malicious IP addresses. This can indicate unauthorized access to AWS resources within your environment.



## UnauthorizedAccess:IAMUser/MaliciousIPCaller.Custom 🔍

Finding ID: 14c175690195ce7eaa130c1913639300

[Feedback](#)

**Medium** API GetFindingsStatistics was invoked from an IP address 137.97.83.9 on the custom threat list ip\_list. [Info](#)

[Investigate with Detective](#)

### Overview

Severity	MEDIUM	🔍
Region	us-east-1	
Count	5	
Account ID	029911053369	🔍
Resource ID	No information available	
Created at	08-29-2022 23:46:48 (7 minutes ago)	
Updated at	08-29-2022 23:46:48 (7 minutes ago)	

### Resource affected

Resource role	TARGET	🔍
Resource type	AccessKey	🔍
Access key ID	ASIAQN5WWOQ4TTTVR4CD	🔍
Principal ID	AIDAQN5WWOQ4Y2Y3YYT17	🔍
User type	IAMUser	🔍
User name	student	🔍

### Affected resources

### Action

Action type	AWS_API_CALL	🔍
API	GetFindingsStatistics	🔍
Service name	guardduty.amazonaws.com	🔍
First seen	08-29-2022 23:37:01 (17 minutes ago)	
Last seen	08-29-2022 23:37:39 (16 minutes ago)	

### Actor

Caller type	Remote IP	🔍
IP address	137.97.83.9	🔍

**Step 35:** Click on “Recon:IAMUser/MaliciousIPCaller.Custom”.

This finding informs you that an API operation that can list or describe AWS resources in an account within your environment was invoked from an IP address that is included on a custom threat list. The threat list used will be listed in the finding's details. An attacker might use stolen credentials to perform this type of reconnaissance of your AWS resources in order to find more valuable credentials or determine the capabilities of the credentials they already have.

## Recon:IAMUser/MaliciousIPCaller.Custom 🔍

Finding ID: 6ec1756901b23a8785c93023ae41dbfa

[Feedback](#)

**Medium**

API ListPublishingDestinations, commonly used in reconnaissance attacks, was invoked from an IP address 137.97.83.9 on the custom threat list ip\_list. Unauthorized actors perform such activity to gather information and discover resources like databases, S3 buckets etc., in order to further tailor the attack. [Info](#)

[Investigate with Detective](#)

### Overview

Severity	MEDIUM	🔍
Region	us-east-1	
Count	5	
Account ID	029911053369	🔍
Resource ID	No information available	
Created at	08-29-2022 23:46:48 (7 minutes ago)	
Updated at	08-29-2022 23:46:48 (7 minutes ago)	

### Resource affected

Resource role	TARGET	🔍
Resource type	AccessKey	🔍
Access key ID	ASIAQN5WWOQ4TTTVR4CD	🔍
Principal ID	AIDAQN5WWOQ4Y2Y3YYTI7	🔍
User type	IAMUser	🔍
User name	student	🔍

Affected resources

### Action

Action type	AWS_API_CALL	🔍
API	ListPublishingDestinations	🔍
Service name	guardduty.amazonaws.com	🔍
First seen	08-29-2022 23:31:56 (22 minutes ago)	
Last seen	08-29-2022 23:32:04 (22 minutes ago)	

### Actor

Caller type	Remote IP	🔍
IP address	137.97.83.9	🔍

Location

**Step 36:** Click on “UnauthorizedAccess:IAMUser/InstanceCredentialExfiltration.OutsideAWS” of resource “S3 Bucket”.

This finding informs you that a host outside of AWS has attempted to run AWS API operations using temporary AWS credentials that were created on an EC2 instance in your AWS environment. The listed EC2 instance might be compromised, and the temporary credentials from this instance might have been exfiltrated to a remote host outside of AWS. AWS does not recommend redistributing temporary credentials outside of the entity that created them (for example, AWS applications, EC2, or Lambda). However, authorized users can export credentials from their EC2 instances to make legitimate API calls. To rule out a potential attack and verify the legitimacy of the activity, validate if the use of instance credentials from the remote IP in the finding is expected.

#### UnauthorizedAccess:IAMUser/InstanceCredentialExfiltration.OutsideAWS 🔍 🔍

Finding ID: 72c175669bbb4894b3f01ff91d3f6c5c

[Feedback](#)

**High** Credentials created exclusively for an EC2 instance using instance role instance\_role have been used from external IP address 137.97.83.9. [Learn More](#) 📄

[Investigate with Detective](#)

Overview			
Severity	HIGH		🔍 🔍
Region	us-east-1		
Count	1		
Account ID	029911053369		🔍 🔍
Resource ID	<a href="#">lab-bucket-029911053369</a> 📄		
Created at	08-29-2022 23:41:33 (13 minutes ago)		
Updated at	08-29-2022 23:41:33 (13 minutes ago)		
Resource affected			
Resource role	TARGET		🔍 🔍
Resource type	S3Bucket		🔍 🔍
Access key ID	ASIAQN5WWOQ42ET7C27U		🔍 🔍
Principal ID	AROAN5WWOQ4UDCIOSWRD:i-0f87c98049400e003		🔍 🔍
User type	AssumedRole		🔍 🔍
User name	instance_role		🔍 🔍
Instance details			
Instance ID	<a href="#">i-0f87c98049400e003</a> 📄		🔍 🔍

Affected resources		
S3 buckets		
Destination: lab-bucket-029911053369		
Name	lab-bucket-029911053369 <a href="#">🔗</a>	🔍 🔍
Type	Destination	🔍 🔍
ARN	arn:aws:s3:::lab-bucket-029911053369	
Effective permission	NOT_PUBLIC	🔍 🔍
Created at	08-29-2022 17:59:03 UTC	
Owner		
ID	b7776fcc401da4227c5559f478e673fa7509e4b12c1f2ae4b5ee8181c391c747	
Action		
Action type	AWS_API_CALL	🔍 🔍
API	PutObject	🔍 🔍
Service name	s3.amazonaws.com	🔍 🔍
First seen	08-29-2022 23:31:19 (23 minutes ago)	
Last seen	08-29-2022 23:31:19 (23 minutes ago)	
Actor		
Caller type	Remote IP	🔍 🔍
IP address	137.97.83.9	🔍 🔍

**Step 37:** Click on “PenTest:IAMUser/KaliLinux” of resource “instance\_role”.

This finding informs you that a machine running Kali Linux is making API calls using credentials that belong to the listed AWS account in your environment. Here the affected resource is an instance role and the access credentials are set on the Kali Linux machine and invoked an API to create a new bucket.

## PenTest:IAMUser/KaliLinux 🔍

Finding ID: **3ec17563731649274fd3678fa42fd4ad**

[Feedback](#)

**Medium** API CreateBucket was invoked from a remote host with IP address 137.97.83.9 that is potentially running the Kali Linux penetration testing tool. [Info](#)

[Investigate with Detective](#)

### Overview

Severity	MEDIUM	🔍
Region	us-east-1	
Count	1	
Account ID	029911053369	🔍
Resource ID	<a href="#">i-Of87c98049400e003</a>	
Created at	08-29-2022 23:34:39 (22 minutes ago)	
Updated at	08-29-2022 23:34:39 (22 minutes ago)	

### Resource affected

Resource role	TARGET	🔍
Resource type	AccessKey	🔍
Access key ID	ASIAQN5WWOQ42ET7C27U	🔍
Principal ID	AROAN5WWOQ4UDCIOSWRD:i-Of87c98049400e003	🔍
User type	AssumedRole	🔍
User name	instance_role	🔍
Instance details		
Instance ID	<a href="#">i-Of87c98049400e003</a>	🔍

### Action

Action type	AWS_API_CALL	🔍
API	CreateBucket	🔍
Service name	s3.amazonaws.com	🔍
First seen	08-29-2022 23:29:02 (28 minutes ago)	
Last seen	08-29-2022 23:29:02 (28 minutes ago)	

### Actor

Caller type	Remote IP	🔍
IP address	137.97.83.9	🔍

**Step 38:** Click on “UnauthorizedAccess:IAMUser/InstanceCredentialExfiltration.OutsideAWS” of resource “instance\_role”.

This finding informs you that a host outside of AWS has attempted to run AWS API operations using temporary AWS credentials that were created on an EC2 instance in your AWS environment. The instance role credentials are set outside of the AWS environment and invoked an API to create a new bucket.

#### UnauthorizedAccess:IAMUser/InstanceCredentialExfiltration.OutsideAWS 🔍 🔍

Finding ID: 84c17566ae4fac54be6161cd7333ee84

[Feedback](#)

**High** Credentials created exclusively for an EC2 instance using instance role instance\_role have been used from external IP address 137.97.83.9. [Learn More](#)

[Investigate with Detective](#)

##### Overview

Severity	HIGH	🔍 🔍
Region	us-east-1	
Count	1	
Account ID	029911053369	🔍 🔍
Resource ID	<a href="#">i-0f87c98049400e003</a>	
Created at	08-29-2022 23:41:43 (17 minutes ago)	
Updated at	08-29-2022 23:41:43 (17 minutes ago)	

##### Resource affected

Resource role	TARGET	🔍 🔍
Resource type	AccessKey	🔍 🔍
Access key ID	ASIAQN5WWOQ42ET7C27U	🔍 🔍
Principal ID	AROAN5WWOQ4UDCIOSWRD:i-0f87c98049400e003	🔍 🔍
User type	AssumedRole	🔍 🔍
User name	instance_role	🔍 🔍

##### Instance details

Instance ID	<a href="#">i-0f87c98049400e003</a>	🔍 🔍
-------------	-------------------------------------	-----

##### Affected resources



Action			
	Action type	AWS_API_CALL	🔍🔍
	API	CreateBucket	🔍🔍
	Service name	s3.amazonaws.com	🔍🔍
	First seen	08-29-2022 23:29:02 (30 minutes ago)	
	Last seen	08-29-2022 23:29:02 (30 minutes ago)	

Actor			
	Caller type	Remote IP	🔍🔍
	IP address	137.97.83.9	🔍🔍

**Step 39:** Click on “PenTest:IAMUser/KaliLinux” of resource “student”.

This finding informs you that a machine running Kali Linux is making API calls using credentials that belong to the listed AWS account in your environment. Here the affected resource is an instance role and the access credentials are set on the Kali Linux machine and invoked an API to Update the user data.

**PenTest:IAMUser/KaliLinux** 🔍🔍
 Feedback

Finding ID: a8c1755db9639f354b20b2a15c0a5d32

**Medium** API UpdateUser was invoked from a remote host with IP address 137.97.83.9 that is potentially running the Kali Linux penetration testing tool. [Info](#)

[Investigate with Detective](#)

Overview			
	Severity	MEDIUM	🔍🔍
	Region	us-east-1	
	Count	6	
	Account ID	029911053369	🔍🔍
	Resource ID	No information available	
	Created at	08-29-2022 23:22:09 (37 minutes ago)	
	Updated at	08-29-2022 23:24:19 (35 minutes ago)	

Resource affected			
	Resource role	TARGET	🔍🔍
	Resource type	AccessKey	🔍🔍
	Access key ID	AKIAQN5WWOQ42ZAKYJV6	🔍🔍
	Principal ID	AIDAQN5WWOQ4Y2Y3YYTI7	🔍🔍
	User type	IAMUser	🔍🔍
	User name	student	🔍🔍

Affected resources



Action			
	Action type	AWS_API_CALL	🔍 🔍
	API	UpdateUser	🔍 🔍
	Service name	iam.amazonaws.com	🔍 🔍
	First seen	08-29-2022 23:14:57 (44 minutes ago)	
	Last seen	08-29-2022 23:17:59 (41 minutes ago)	

Actor			
	Caller type	Remote IP	🔍 🔍
	IP address	137.97.83.9	🔍 🔍

Thus GuardDuty generates a finding whenever it detects unexpected and potentially malicious activity in IAM operations.

## References:

1. Amazon GuardDuty  
([https://docs.aws.amazon.com/guardduty/latest/ug/guardduty\\_setup.html](https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_setup.html))