ATTACK DEFENSE

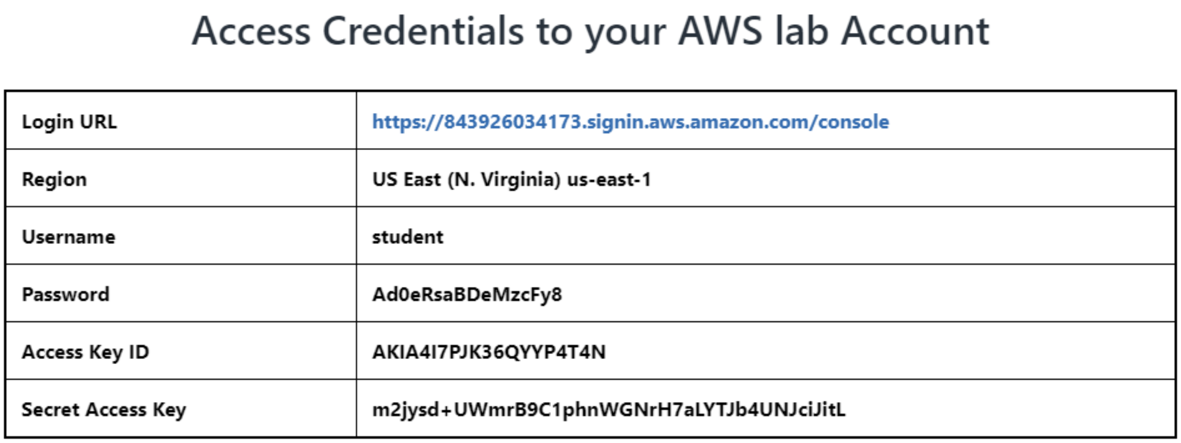
by PentesterAcademy

Name	Amazon Inspector
URL	https://attackdefense.com/challengedetails?cid=2486
Type	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:

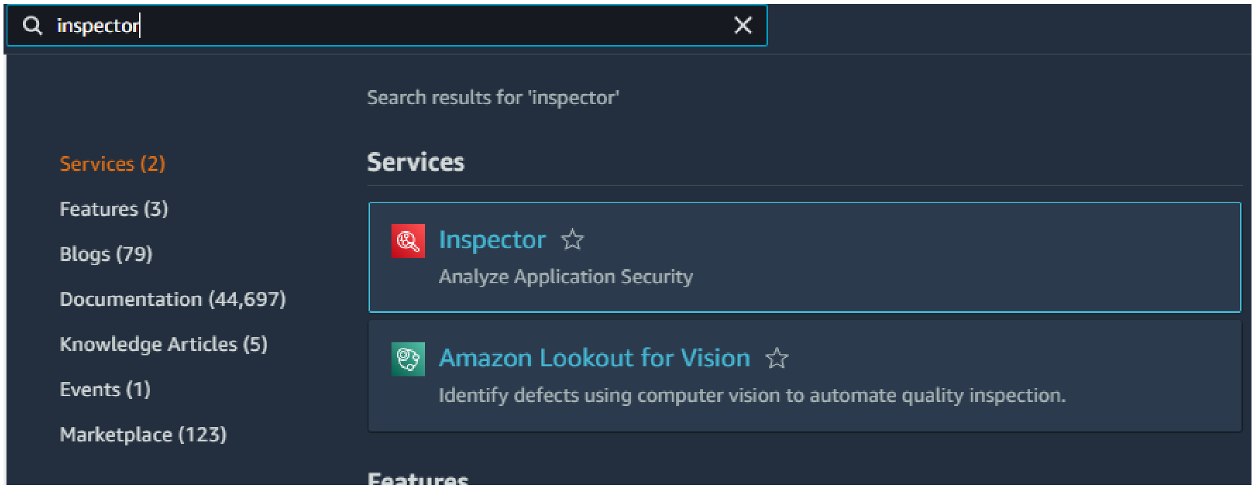
Step 1: Click the lab link button to get access credentials.



Access Credentials to your AWS lab Account

Login URL	https://843926034173.signin.aws.amazon.com/console
Region	US East (N. Virginia) us-east-1
Username	student
Password	Ad0eRsaBDeMzcFy8
Access Key ID	AKIA4I7PJK36QYYP4T4N
Secret Access Key	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

Step 2: Search for inspector in the search bar and navigate to the Inspector dashboard.



inspector

Search results for 'inspector'

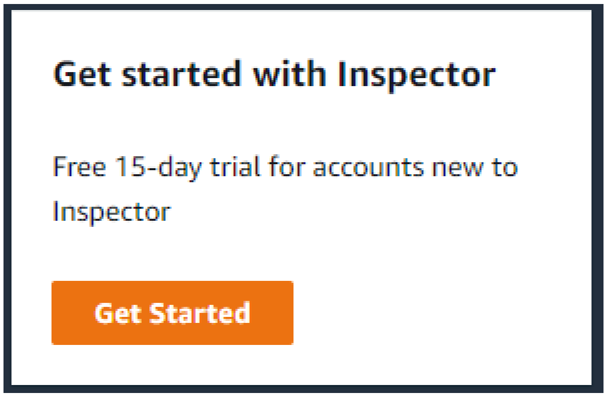
Services

Inspector ☆
Analyze Application Security

Amazon Lookout for Vision ☆
Identify defects using computer vision to automate quality inspection.

Features

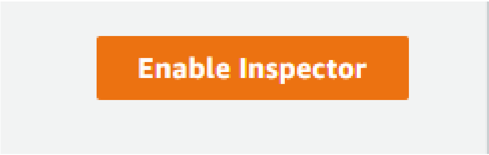
Step 3: Click on the “Get Started” button.



Get started with Inspector

Free 15-day trial for accounts new to Inspector

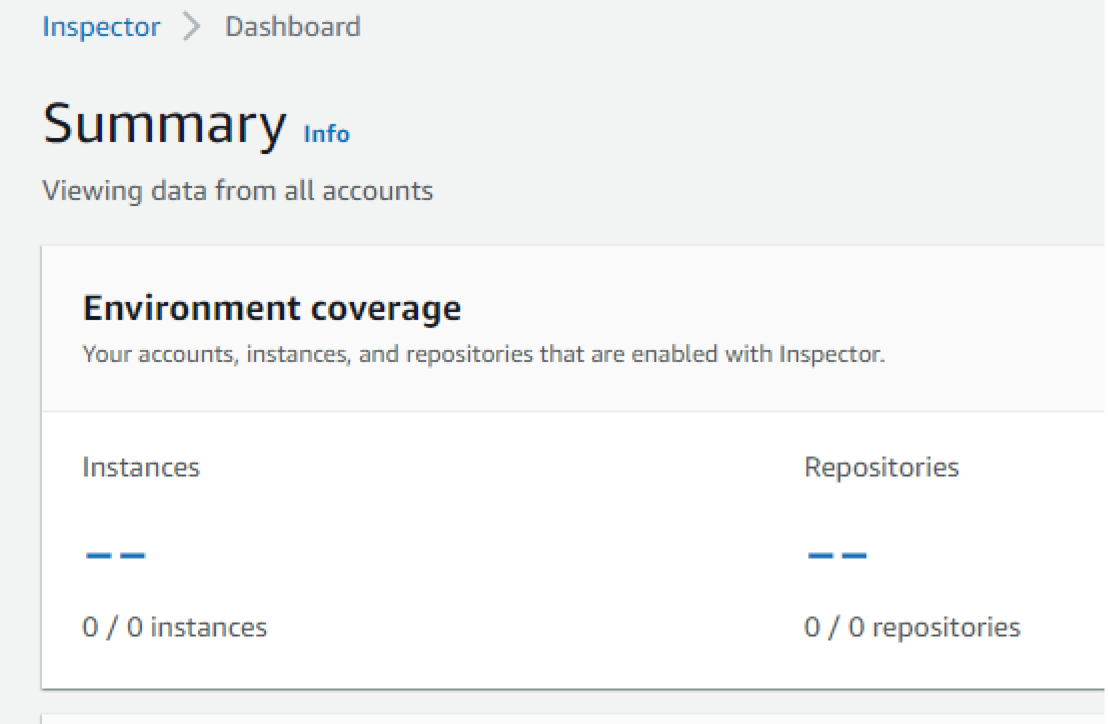
Step 4: Enable the inspector by clicking the “Enable Inspector” button.



The Amazon Inspector dashboard provides a snapshot of aggregated statistics for your Amazon resources. These statistics include key metrics for resource coverage and active vulnerabilities.

The dashboard also displays groups of aggregated findings data for your account, such as EC2 instances with most critical findings.

The Environment coverage section provides statistics about the resources scanned by Amazon Inspector. In this section, you can see the count and percentage of Amazon EC2 instances and Amazon ECR images scanned by Amazon Inspector.



Inspector > Dashboard

Summary Info

Viewing data from all accounts

Environment coverage

Your accounts, instances, and repositories that are enabled with Inspector.

Instances

--

0 / 0 instances

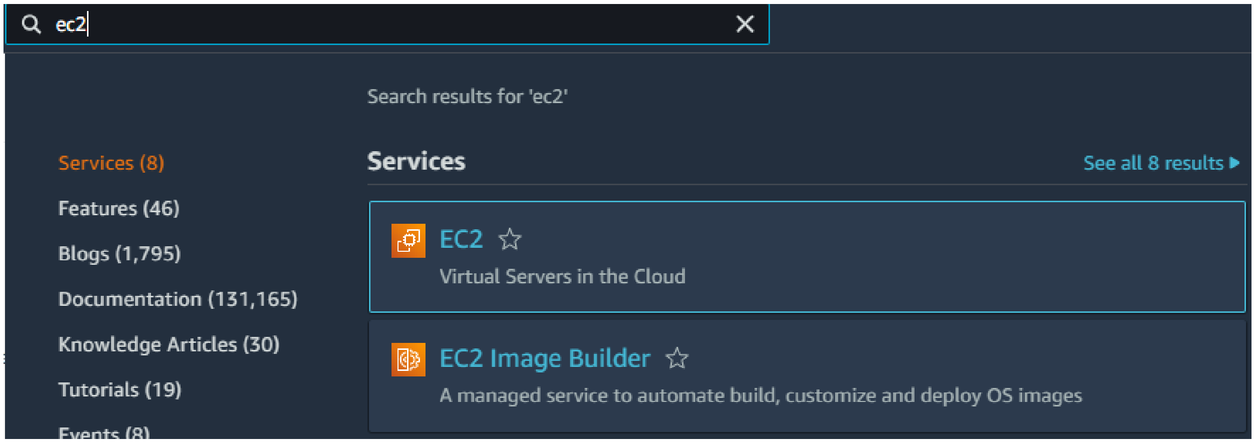
Repositories

--

0 / 0 repositories

Now create an EC2 instance and install a vulnerable package.

Step 5: Search for EC2 in the search bar and navigate to the EC2 dashboard.



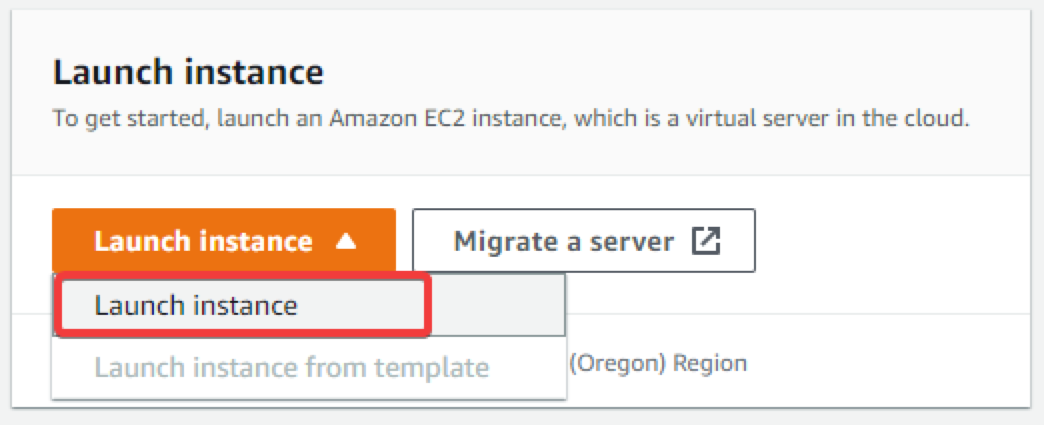
ec2

Search results for 'ec2'

EC2 ☆
Virtual Servers in the Cloud

EC2 Image Builder ☆
A managed service to automate build, customize and deploy OS images

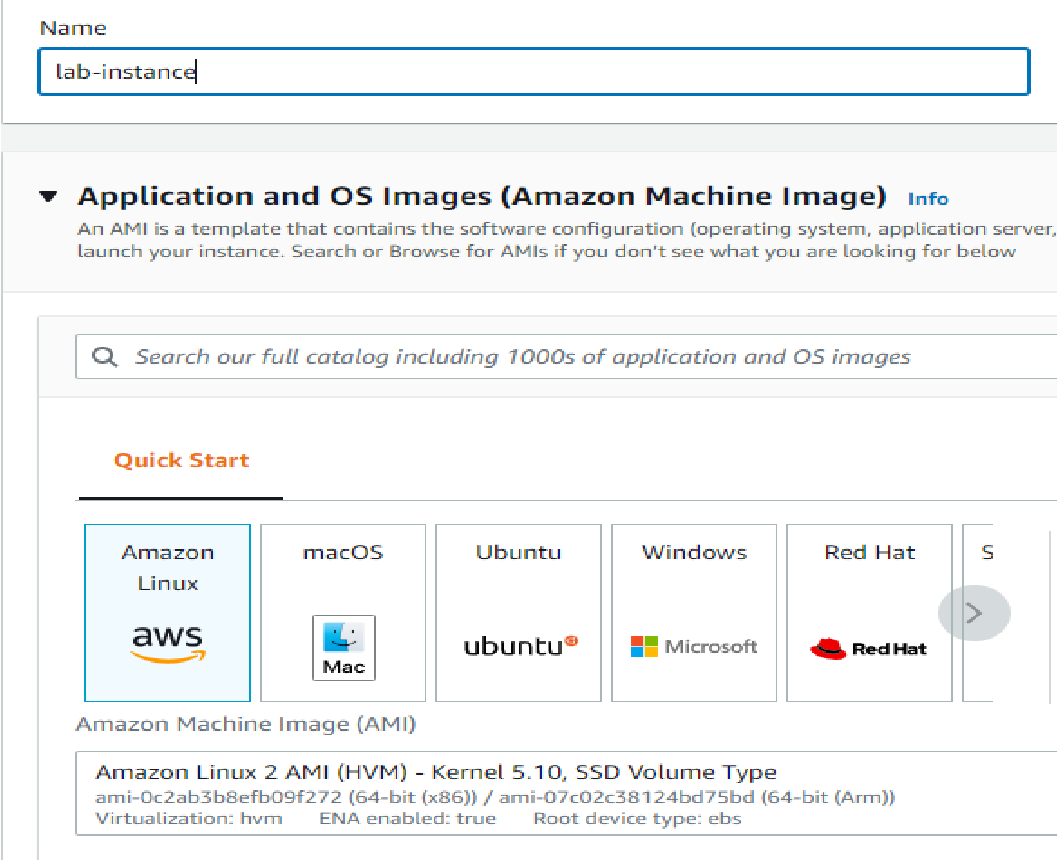
Step 6: Click on the “Launch instance” option.



Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Step 7: Set name as “lab-instance” and choose “Amazon Linux” from Quick Start.



Name

lab-instance

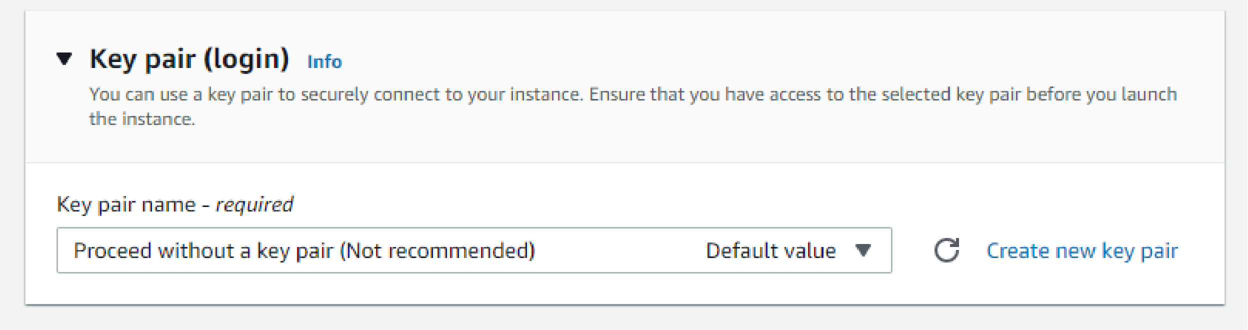
▼ Application and OS Images (Amazon Machine Image) Info

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

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type
ami-0c2ab3b8efb09f272 (64-bit (x86)) / ami-07c02c38124bd75bd (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Step 8: In the key pair section , choose the option to proceed without a key pair.



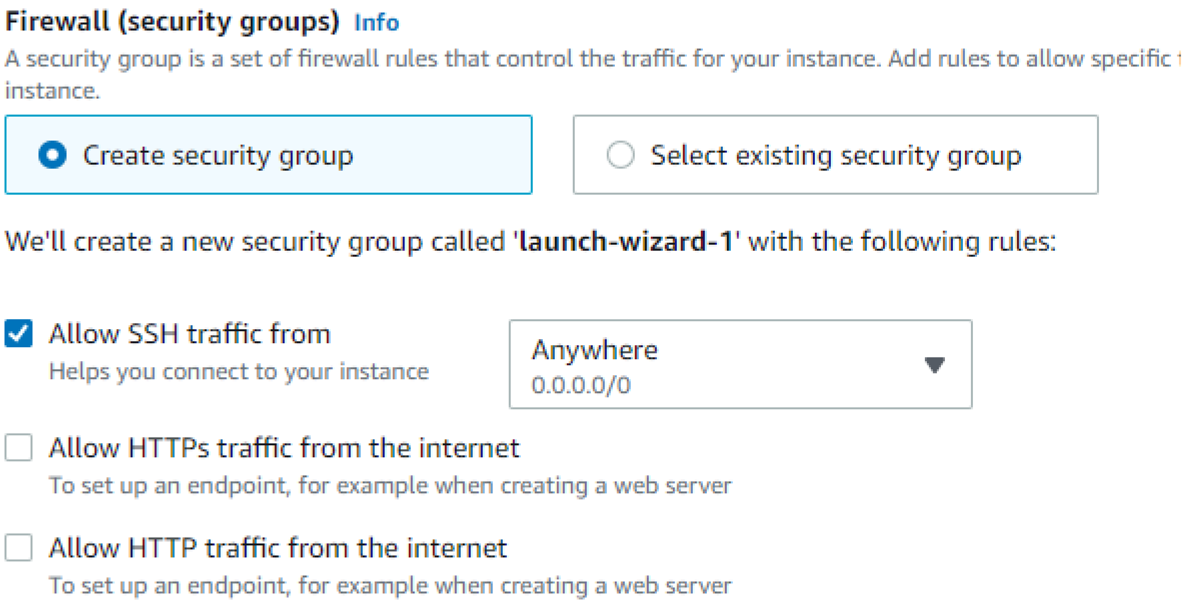
▼ Key pair (login) Info

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

Step 9: Now choose “Create security group” and allow SSH traffic.



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 instance.

● Create security group ○ Select existing security group

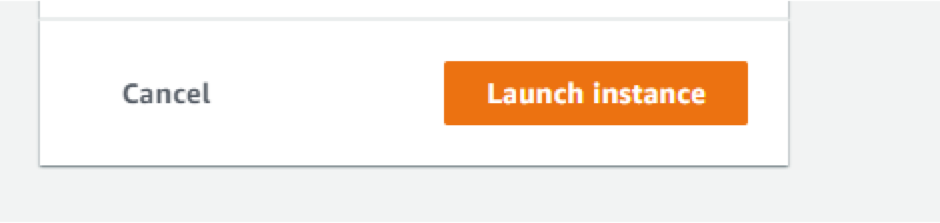
We'll create a new security group called '**launch-wizard-1**' with the following rules:

☑ Allow SSH traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0

□ Allow HTTPs traffic from the internet
To set up an endpoint, for example when creating a web server

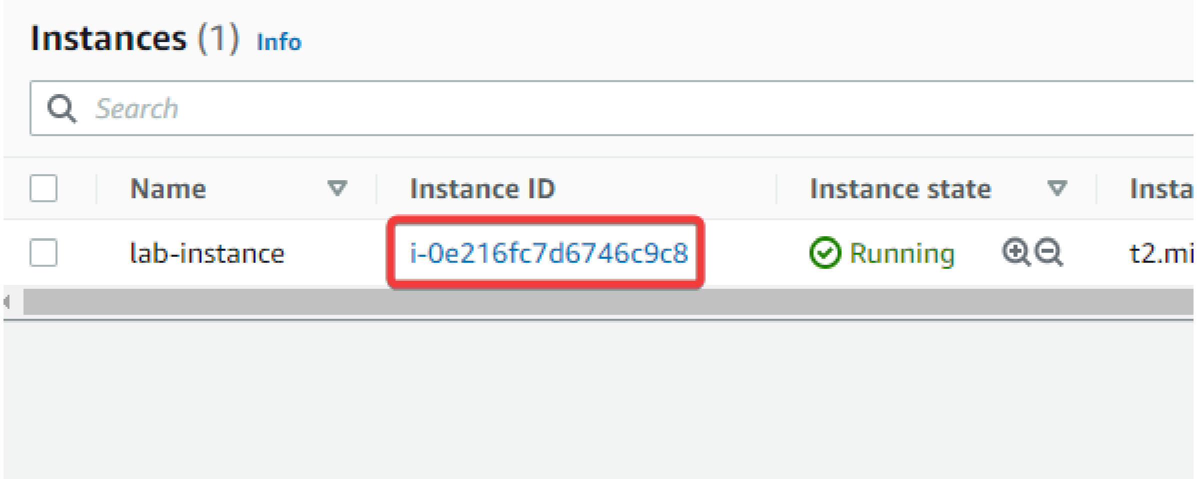
□ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

Step 10: Click on the “Launch instance” button.



In order for Amazon Inspector to detect software vulnerabilities for an EC2 instance, the instance must be a managed instance in Amazon EC2 Systems Manager (SSM). An SSM managed instance has the SSM Agent installed and running, and has an attached IAM instance profile that allows SSM to manage the instance.

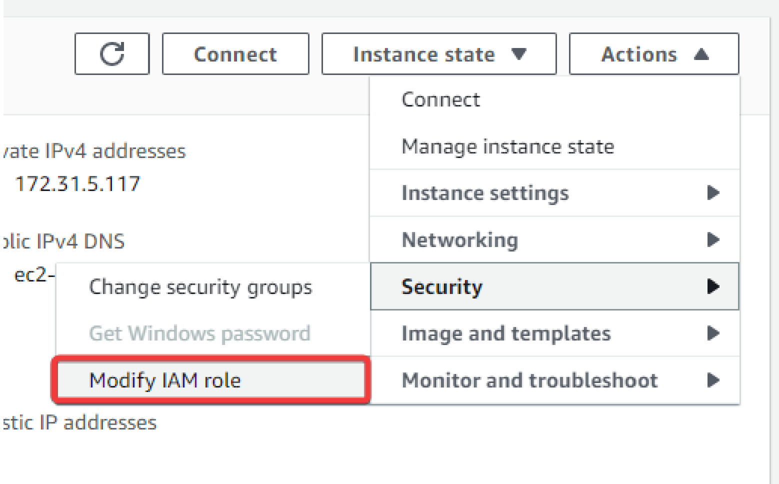
Step 11: Click the instance id after the state turns “Running”.


Instances (1) Info

Search

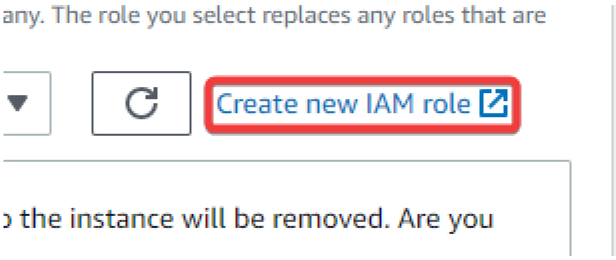
Name	Instance ID	Instance state	Insta
lab-instance	i-0e216fc7d6746c9c8	Running	t2.mi

Step 12: Select “Modify IAM role” from Security under the actions drop-down.



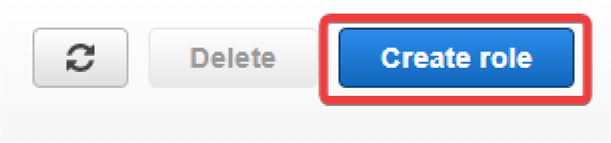
/ate IPv4 addresses
172.31.5.117
olic IPv4 DNS
ec2-
stic IP addresses

Step 13: Click on “Create new IAM role”.

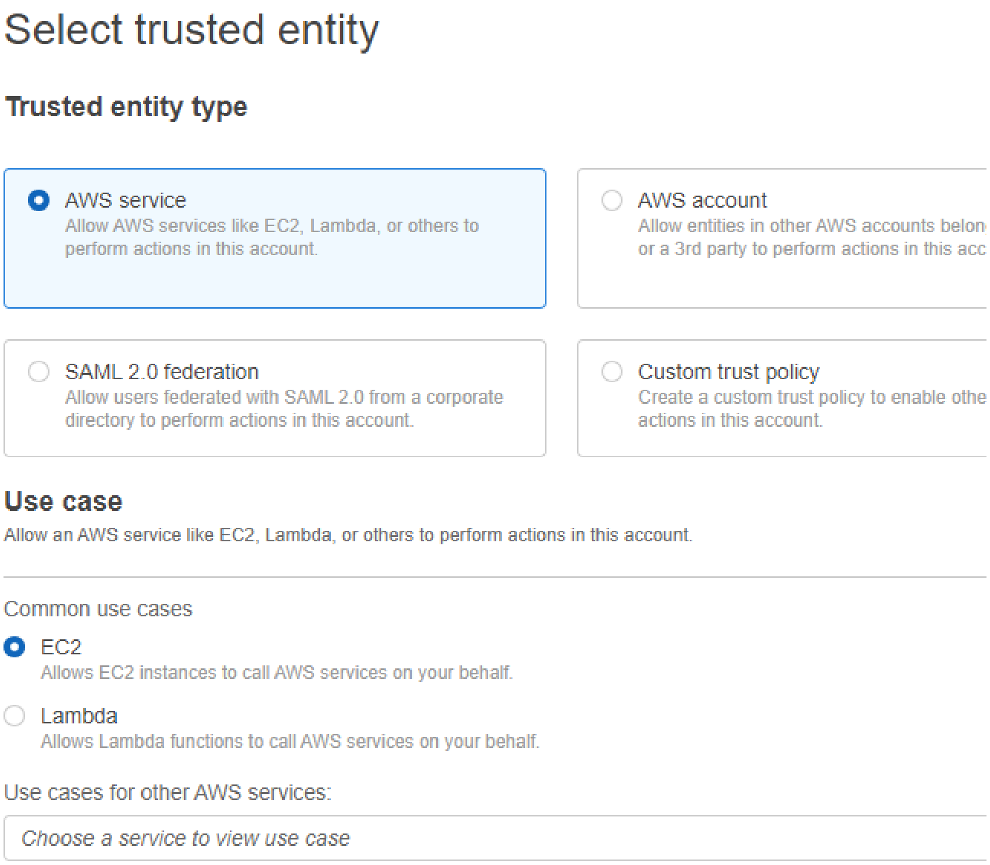

any. The role you select replaces any roles that are

› the instance will be removed. Are you

Step 14: Click on “Create role”.



Step 15: Choose trusted entity type as “AWS service” and use case as “EC2”.



Select trusted entity

Trusted entity type

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

○ AWS account
Allow entities in other AWS accounts belon
or a 3rd party to perform actions in this acc

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

○ Custom trust policy
Create a custom trust policy to enable othe
actions in this account.

Use case

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

Common use cases

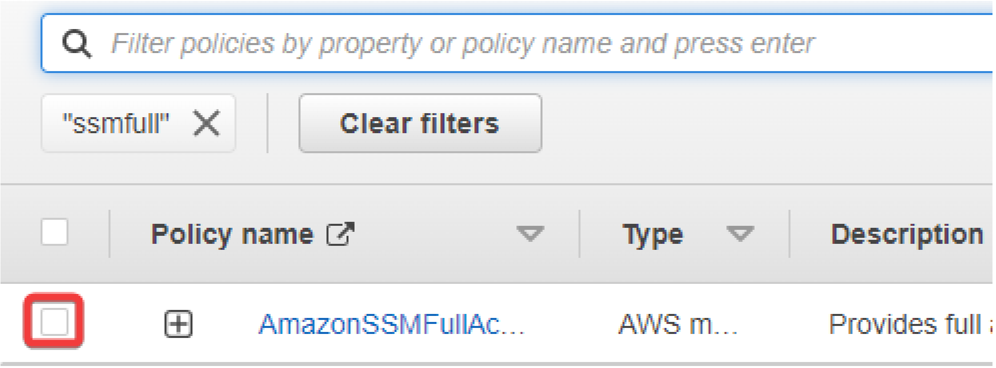
● EC2
Allows EC2 instances to call AWS services on your behalf.

○ Lambda
Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

Choose a service to view use case

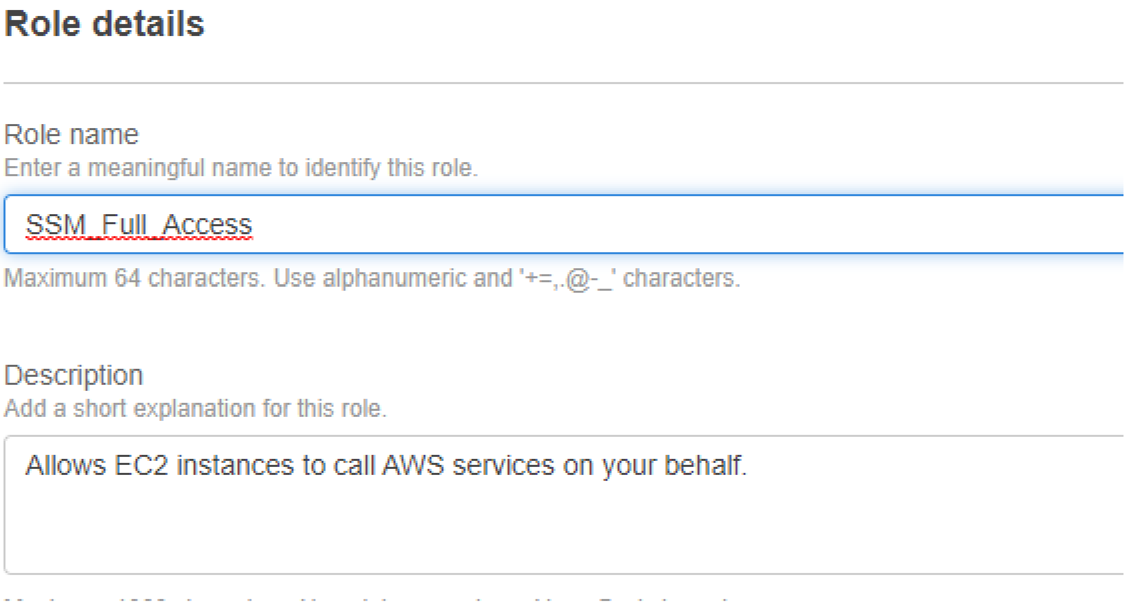
Step 16: Search “ssmfull” in policies search bar and select “AmazonSSMFullAccess” and click on “Next” button.


Filter policies by property or policy name and press enter

"ssmfull" ✕ Clear filters

□	Policy name	Type	Description
□	⊞ AmazonSSMFullAc...	AWS m...	Provides full

Step 17: Set role name as “SSM_Full_Access”.


Role details

Role name
Enter a meaningful name to identify this role.

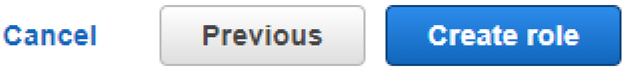
SSM_Full_Access

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

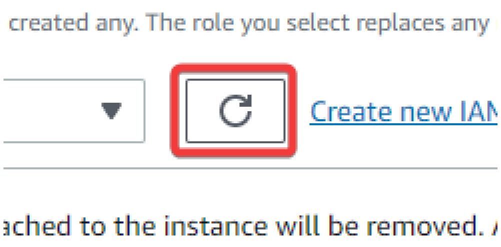
Description
Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

Step 18: Click on “Create role” .

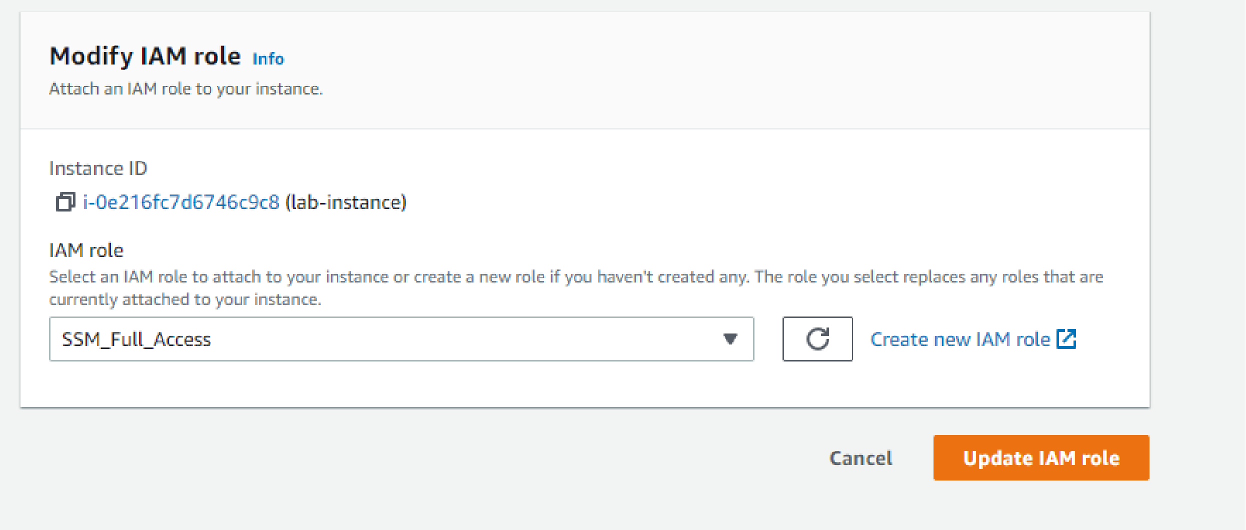


Step 19: Navigate back to the EC2 instance and attach a role with the instance. Click on the refresh button.


created any. The role you select replaces any

ached to the instance will be removed.

Step 20: Select “SSM_Full_Access” and click on the “Update IAM role” button.


Modify IAM role Info

Attach an IAM role to your instance.

Instance ID

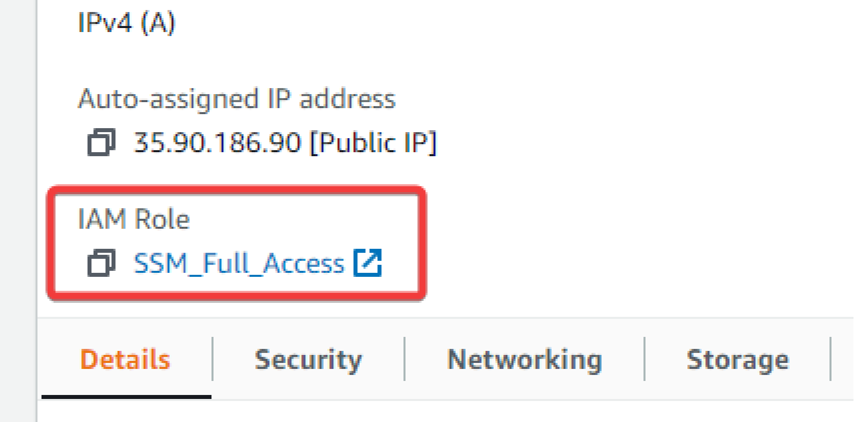
i-0e216fc7d6746c9c8 (lab-instance)

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.

SSM_Full_Access

Successfully attached an IAM role with the instance.


IPv4 (A)

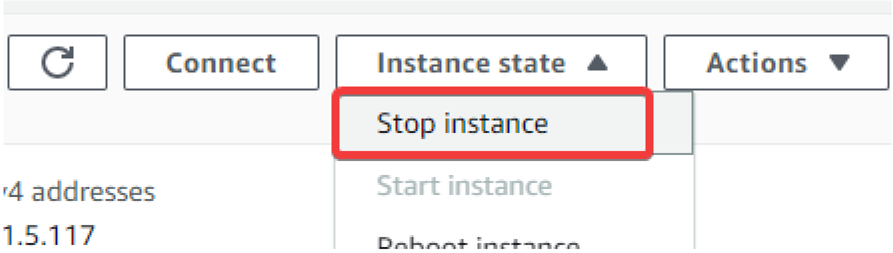
Auto-assigned IP address

35.90.186.90 [Public IP]

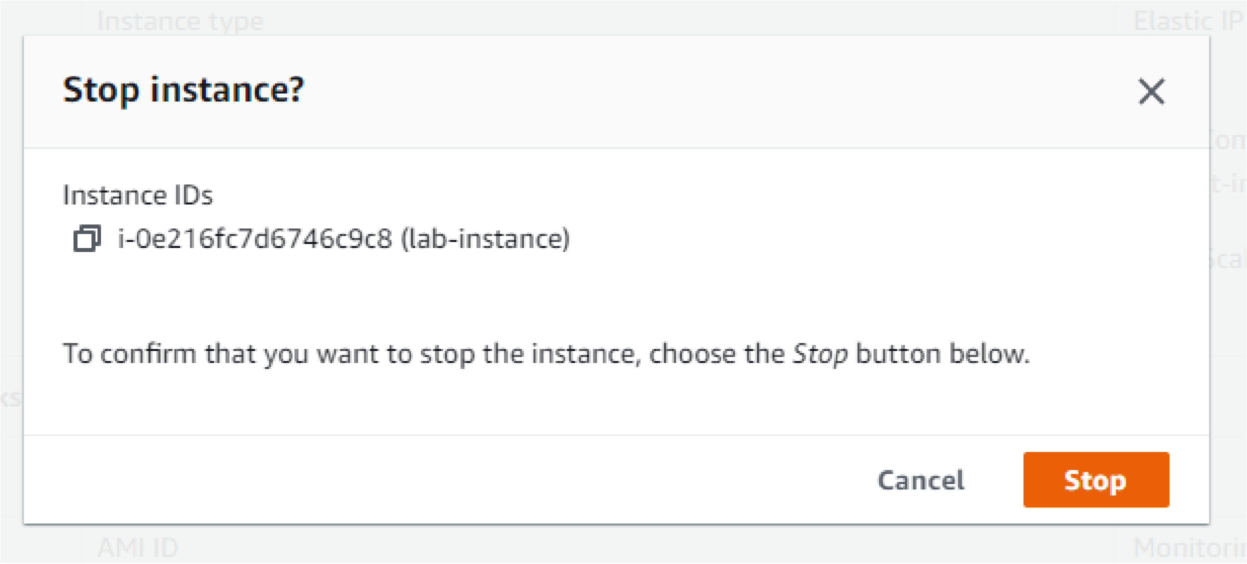
IAM Role

SSM_Full_Access

Step 21: Now stop and start the instance to make the configuration to take effect. Click on "Stop" under the "Instance state".



Click on "Stop" and confirm.



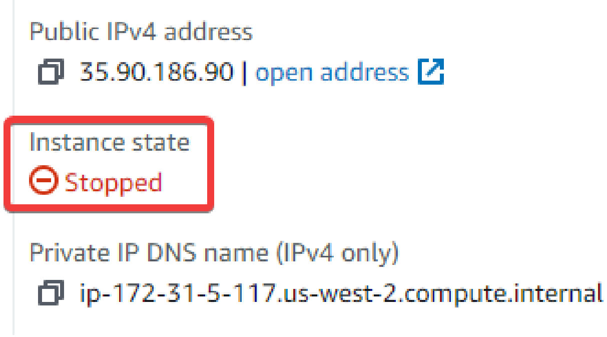
Stop instance?

Instance IDs

i-0e216fc7d6746c9c8 (lab-instance)

To confirm that you want to stop the instance, choose the *Stop* button below.

Successfully "Stopped" the instance.


Public IPv4 address

35.90.186.90 | open address

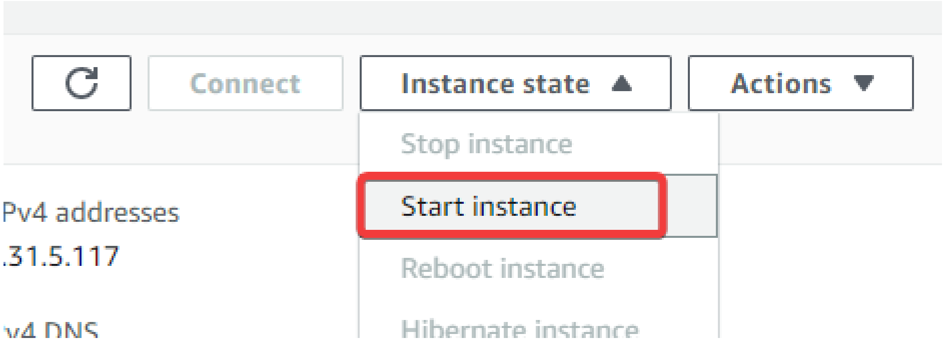
Instance state

Stopped

Private IP DNS name (IPv4 only)

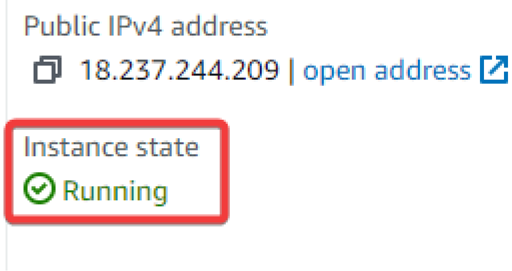
ip-172-31-5-117.us-west-2.compute.internal

Step 22: Click on “Start instance” under “Instance state”.



Pv4 addresses
.31.5.117
v4 DNS

Successfully started the instance.

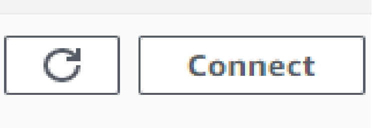

Public IPv4 address

18.237.244.209 | open address

Instance state

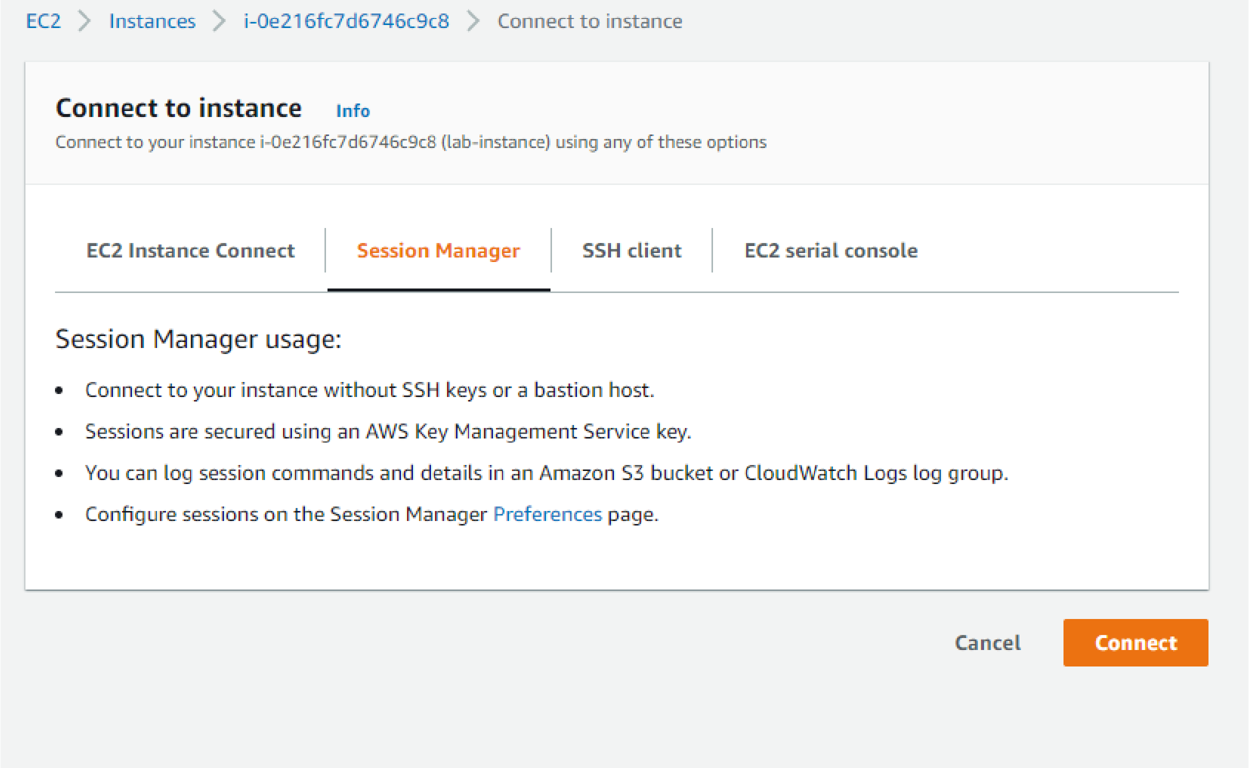
Running

Step 23: Now, click on “Connect”.



Step 24: Select “Session Manager” and click on the “Connect” button.

Note: If it shows any configuration issue, start and stop the instance again.



Connect to instance Info

Connect to your instance i-0e216fc7d6746c9c8 (lab-instance) using any of these options

Session Manager usage:

- Connect to your instance without SSH keys or a bastion host.
- Sessions are secured using an AWS Key Management Service key.
- You can log session commands and details in an Amazon S3 bucket or CloudWatch Logs log group.
- Configure sessions on the Session Manager Preferences page.

Step 25: Select bash shell and switch to root user and execute the following commands in the shell to install a vulnerable httpd package.

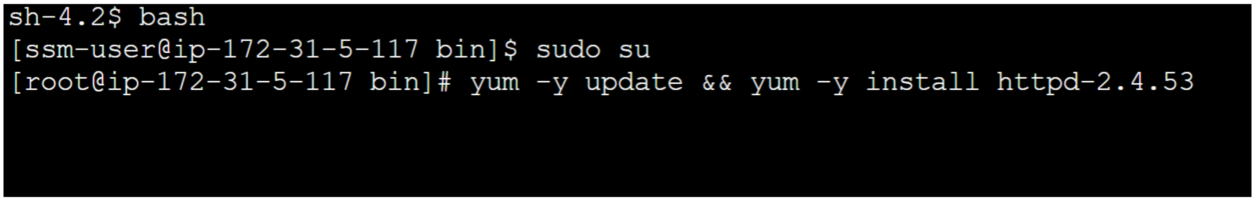
Apache HTTP Server 2.4.53 and earlier may not send the X-Forwarded-* headers to the origin server based on client side connection header hop-by-hop mechanism. An unauthenticated attacker with network access to the data plane may exploit this vulnerability to bypass IP-based authentication on the origin server or application (CVE-2022-31813)

Commands:

```
bash
```

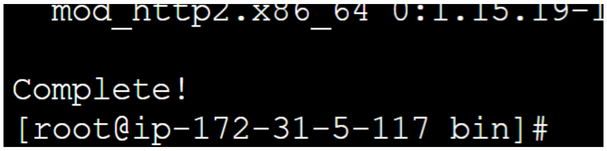
sudo su

yum -y update && yum -y install httpd-2.4.53



```
sh-4.2$ bash
[ssm-user@ip-172-31-5-117 bin]$ sudo su
[root@ip-172-31-5-117 bin]# yum -y update && yum -y install httpd-2.4.53
```

Successfully installed httpd package with version 2.4.53.

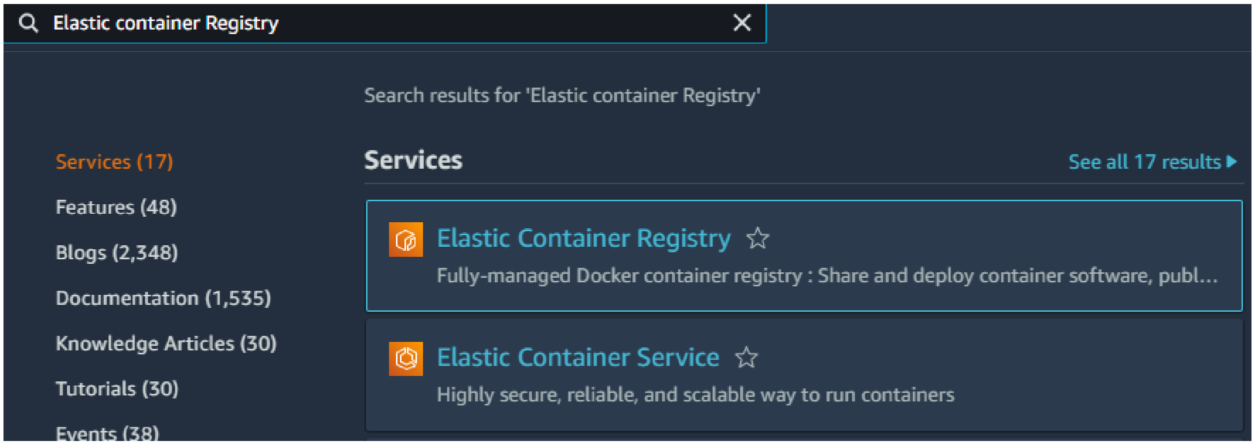


```
  mod_http2.x86_64 0:1.15.19-1

Complete!
[root@ip-172-31-5-117 bin]#
```

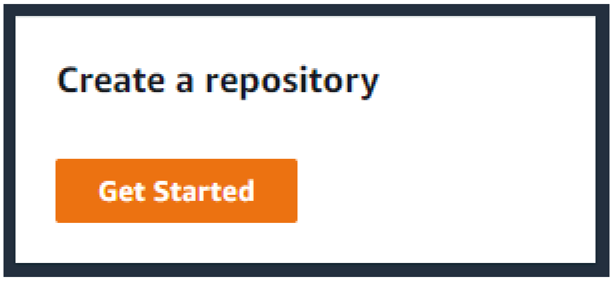
Now create an image repository and push a docker image.

Step 26: Search for “Elastic container registry” in the search bar and navigate to the ECR dashboard.

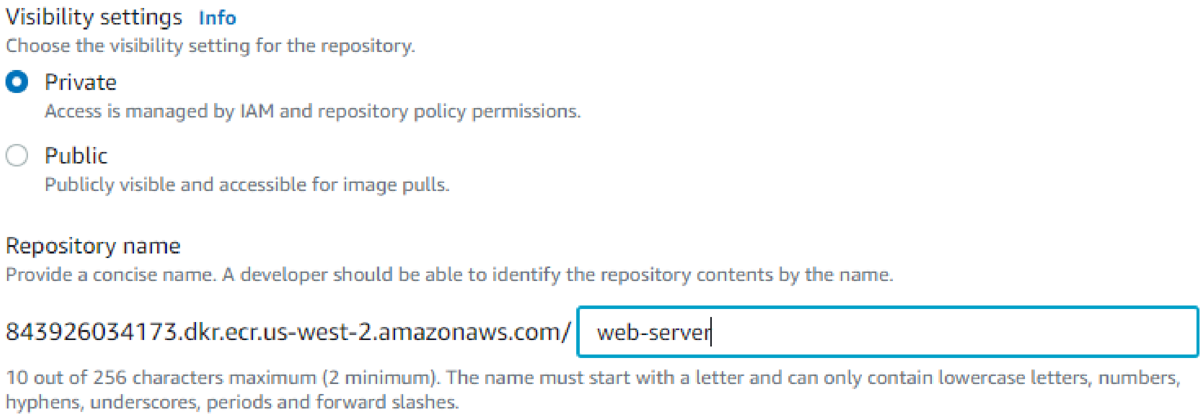


Elastic container Registry
Search results for 'Elastic container Registry'
Services (17)
Features (48)
Blogs (2,348)
Documentation (1,535)
Knowledge Articles (30)
Tutorials (30)
Events (38)
Services
See all 17 results ▶
Elastic Container Registry ☆
Fully-managed Docker container registry : Share and deploy container software, publ...
Elastic Container Service ☆
Highly secure, reliable, and scalable way to run containers

Step 27: Click on “Get Started”.



Step 28: Set visibility as “Private” and repository name as “web-server”.



Visibility settings Info
Choose the visibility setting for the repository.

● Private
Access is managed by IAM and repository policy permissions.

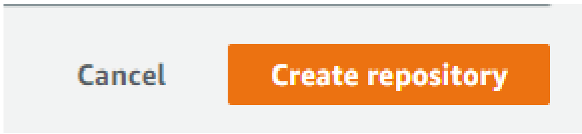
○ Public
Publicly visible and accessible for image pulls.

Repository name
Provide a concise name. A developer should be able to identify the repository contents by the name.

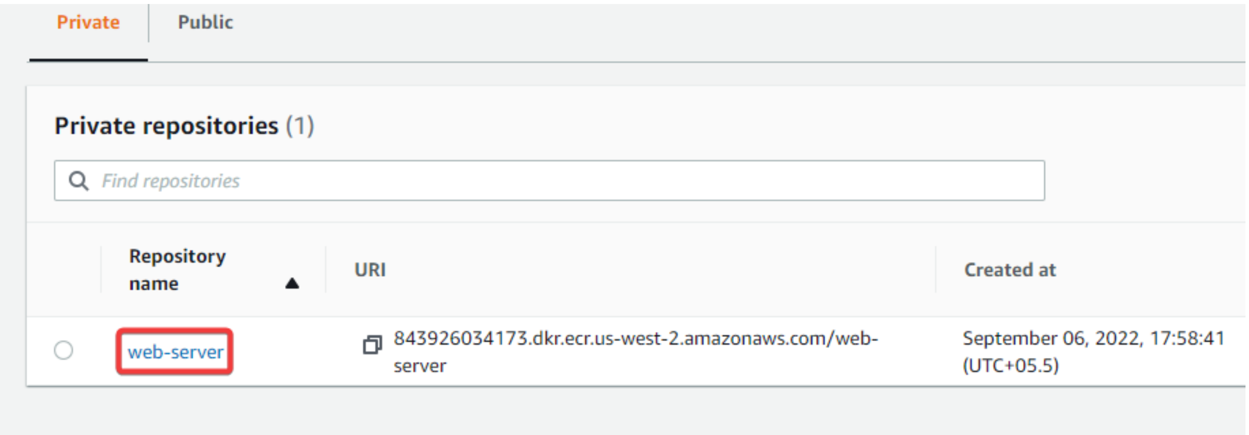
843926034173.dkr.ecr.us-west-2.amazonaws.com/ web-server

10 out of 256 characters maximum (2 minimum). The name must start with a letter and can only contain lowercase letters, numbers, hyphens, underscores, periods and forward slashes.

Step 29: Click on the “Create repository” button.



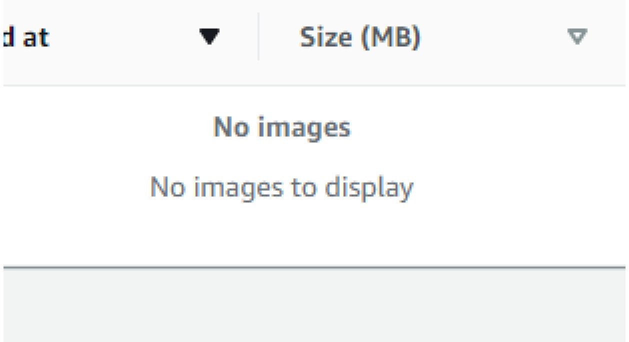
Step 30: Successfully created a private repository. Click on “web-server”.



Private repositories (1)

	Repository name	URI	Created at
○	web-server	843926034173.dkr.ecr.us-west-2.amazonaws.com/web-server	September 06, 2022, 17:58:41 (UTC+05.5)

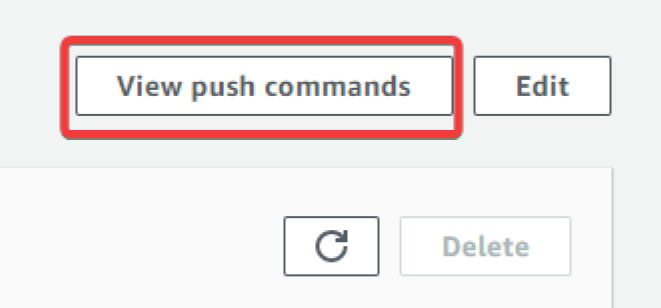
There are no images available in the repository.



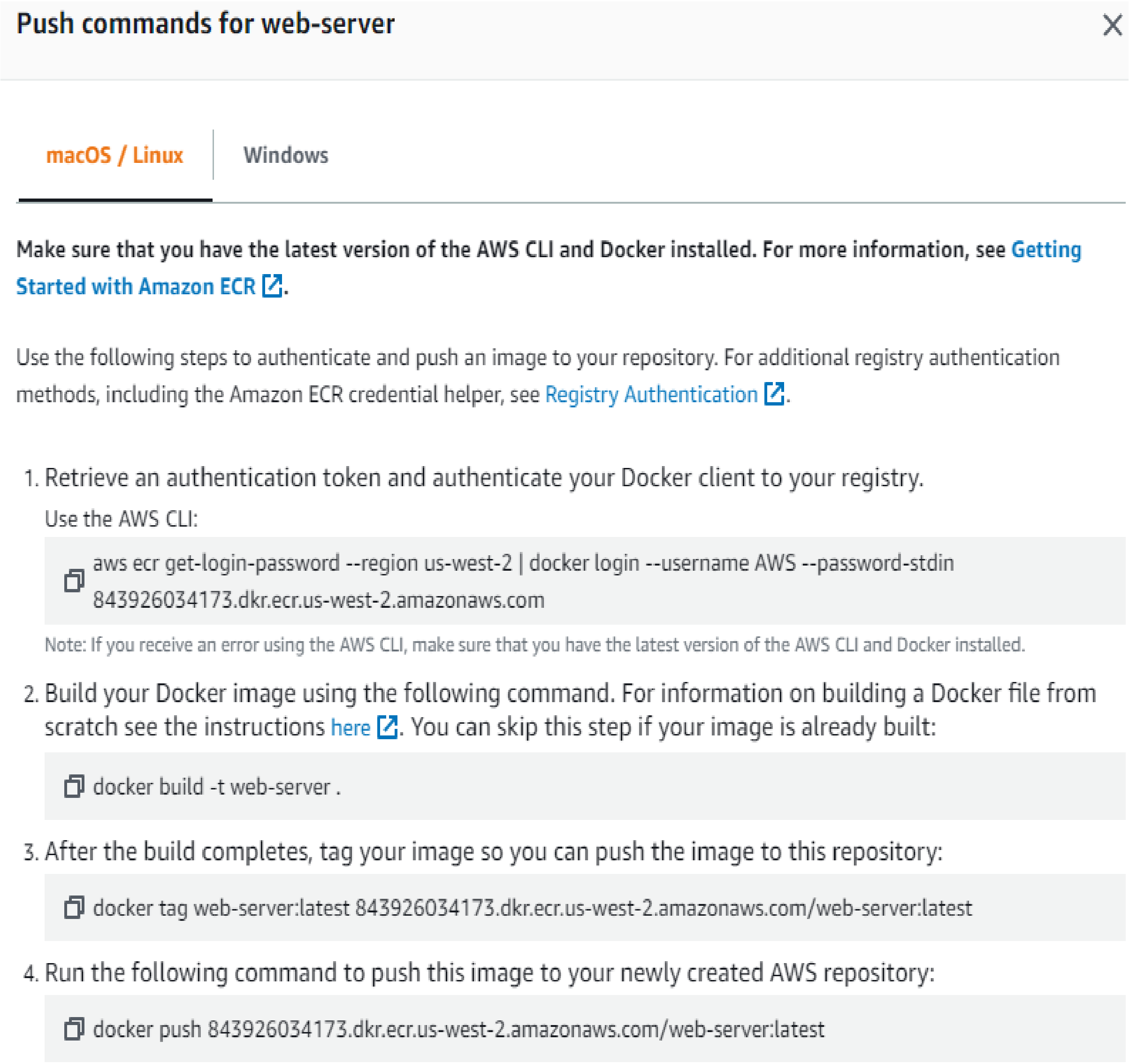
No images

No images to display

Step 31: Click on “View push commands”.



Follow these steps in the local machine to push the image to the created repository.



Push commands for web-server

Make sure that you have the latest version of the AWS CLI and Docker installed. For more information, see Getting Started with Amazon ECR.

Use the following steps to authenticate and push an image to your repository. For additional registry authentication methods, including the Amazon ECR credential helper, see Registry Authentication.

1. Retrieve an authentication token and authenticate your Docker client to your registry.

   Use the AWS CLI:

   ```
   aws ecr get-login-password --region us-west-2 | docker login --username AWS --password-stdin
   843926034173.dkr.ecr.us-west-2.amazonaws.com
   ```

   Note: If you receive an error using the AWS CLI, make sure that you have the latest version of the AWS CLI and Docker installed.

2. Build your Docker image using the following command. For information on building a Docker file from scratch see the instructions here. You can skip this step if your image is already built:

   ```
   docker build -t web-server .
   ```

3. After the build completes, tag your image so you can push the image to this repository:

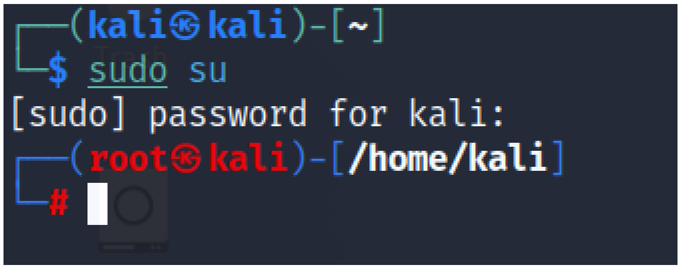
   ```
   docker tag web-server:latest 843926034173.dkr.ecr.us-west-2.amazonaws.com/web-server:latest
   ```

4. Run the following command to push this image to your newly created AWS repository:

   ```
   docker push 843926034173.dkr.ecr.us-west-2.amazonaws.com/web-server:latest
   ```

Step 32: Switch to root user.

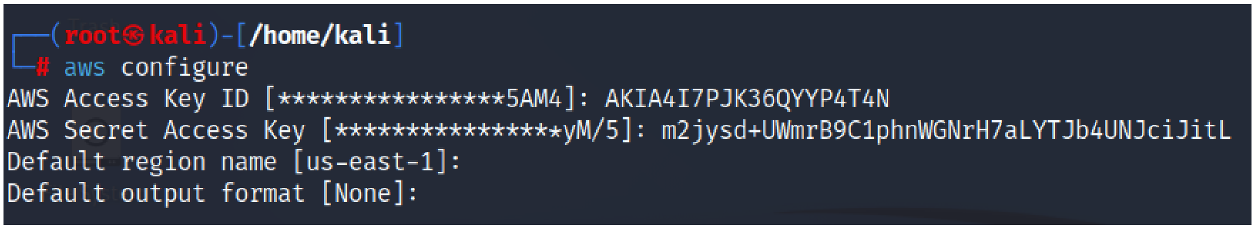
Command: sudo su



```
┌──(kali㉿kali)-[~]
└─$ sudo su
[sudo] password for kali:
┌──(root㉿kali)-[/home/kali]
└─#
```

Step 33: Configure AWS CLI using the provided credentials.

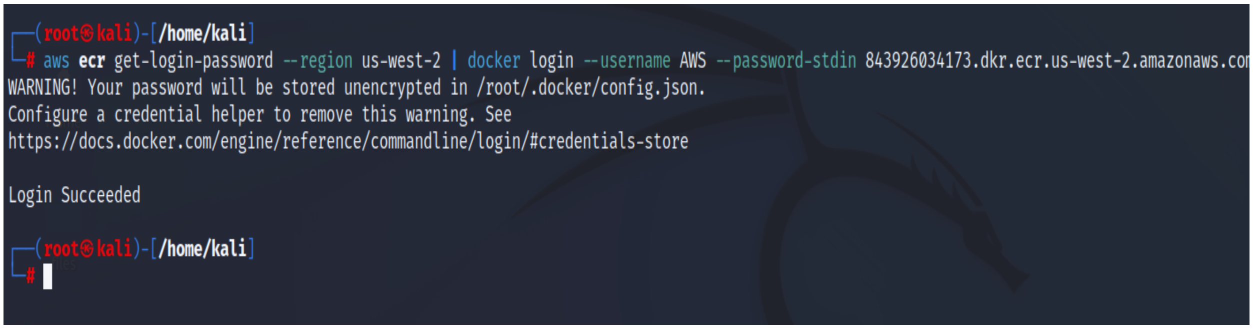
Command: aws configure



```
┌──(root㉿kali)-[/home/kali]
└─# aws configure
AWS Access Key ID [****************5AM4]: AKIA4I7PJK36QYYP4T4N
AWS Secret Access Key [****************yM/5]: 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
Default region name [us-east-1]:
Default output format [None]:
```

Step 34: Retrieve an authentication token and authenticate your Docker client to your registry.

Command: aws ecr get-login-password --region us-west-2 | docker login --username AWS --password-stdin 843926034173.dkr.ecr.us-west-2.amazonaws.com



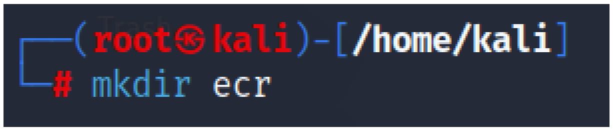
```
┌──(root㉿kali)-[/home/kali]
└─# aws ecr get-login-password --region us-west-2 | docker login --username AWS --password-stdin 843926034173.dkr.ecr.us-west-2.amazonaws.com
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded

┌──(root㉿kali)-[/home/kali]
└─#
```

Step 35: Create a new directory to setup a Dockerfile.

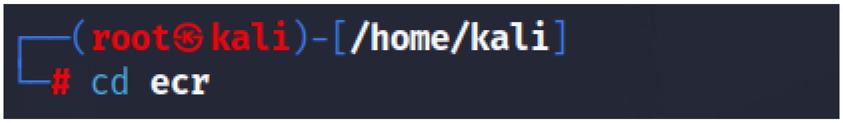
Command: mkdir ecr



```
┌──(root㉿kali)-[/home/kali]
└─# mkdir ecr
```

Step 36: Navigate to the “ecr” directory.

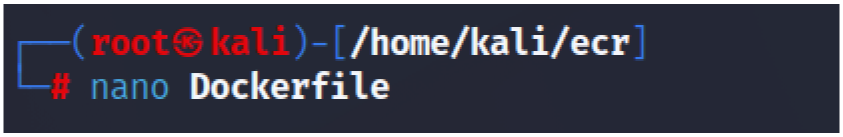
Command: cd ecr



```
┌──(root㉿kali)-[/home/kali]
└─# cd ecr
```

Step 37: Use nano to create a Dockerfile with the following code.

Command: nano Dockerfile

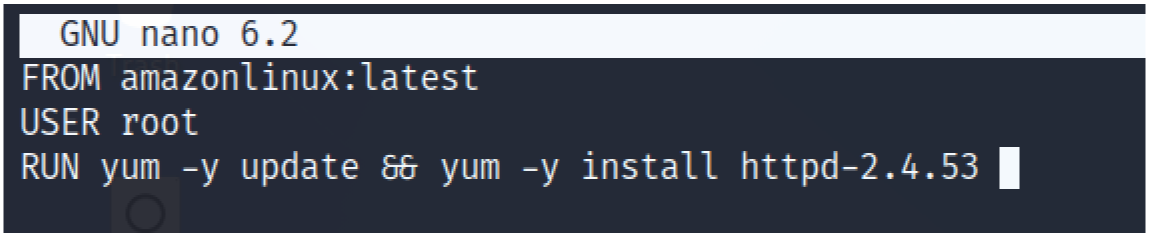


```
┌──(root㉿kali)-[/home/kali/ecr]
└─# nano Dockerfile
```

Paste the following code into the file. This code will pack the vulnerable httpd package into an image after build.

Dockerfile:

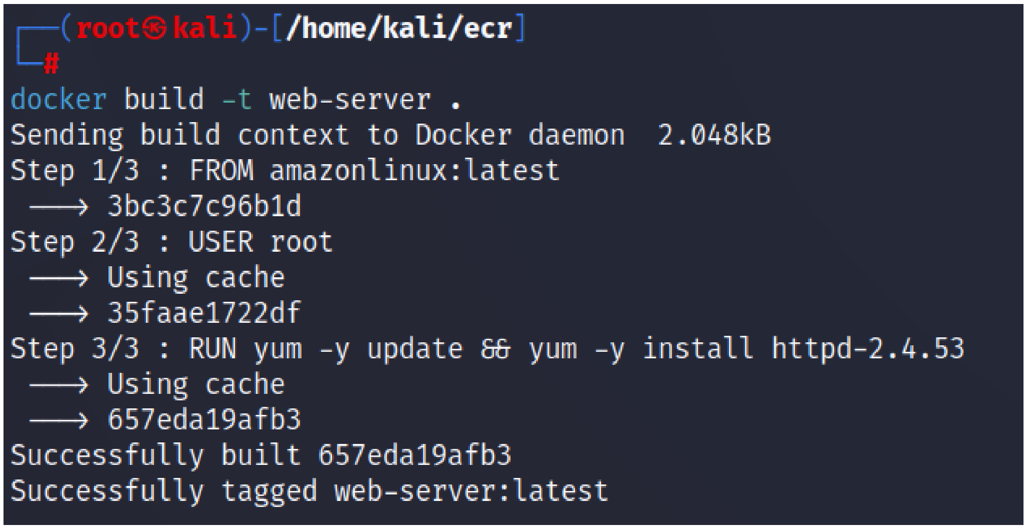
```
FROM amazonlinux:latest
USER root
RUN yum -y update && yum -y install httpd-2.4.53
```



```
  GNU nano 6.2
FROM amazonlinux:latest
USER root
RUN yum -y update && yum -y install httpd-2.4.53
```

Step 38: Build your Docker image using the following command.

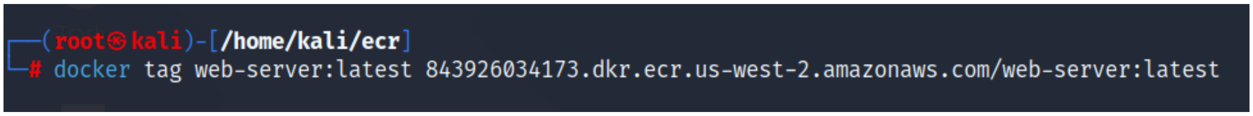
Command: docker build -t web-server .



```
┌──(root㉿kali)-[/home/kali/ecr]
└─#
docker build -t web-server .
Sending build context to Docker daemon  2.048kB
Step 1/3 : FROM amazonlinux:latest
 ---> 3bc3c7c96b1d
Step 2/3 : USER root
 ---> Using cache
 ---> 35faae1722df
Step 3/3 : RUN yum -y update && yum -y install httpd-2.4.53
 ---> Using cache
 ---> 657eda19afb3
Successfully built 657eda19afb3
Successfully tagged web-server:latest
```

Step 39: Tag the image to push the image to the created repository.

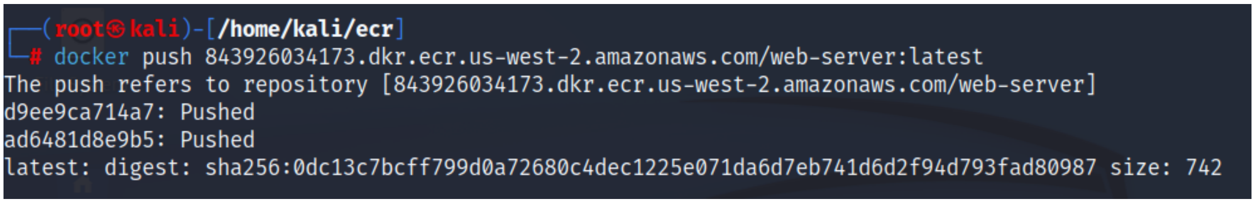
Command: docker tag web-server:latest 843926034173.dkr.ecr.us-west-2.amazonaws.com/web-server:latest



```
┌──(root㉿kali)-[/home/kali/ecr]
└─# docker tag web-server:latest 843926034173.dkr.ecr.us-west-2.amazonaws.com/web-server:latest
```

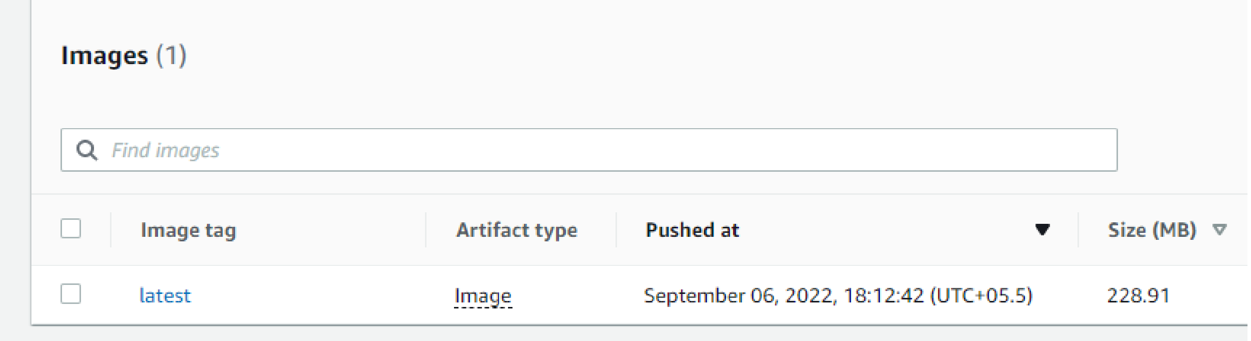
Step 40: Execute the following command to push this image to your newly created AWS repository.

Command: docker push 843926034173.dkr.ecr.us-west-2.amazonaws.com/web-server:latest



```
┌──(root㉿kali)-[/home/kali/ecr]
└─# docker push 843926034173.dkr.ecr.us-west-2.amazonaws.com/web-server:latest
The push refers to repository [843926034173.dkr.ecr.us-west-2.amazonaws.com/web-server]
d9ee9ca714a7: Pushed
ad6481d8e9b5: Pushed
latest: digest: sha256:0dc13c7bcff799d0a72680c4dec1225e071da6d7eb741d6d2f94d793fad80987 size: 742
```

Successfully pushed the created image.

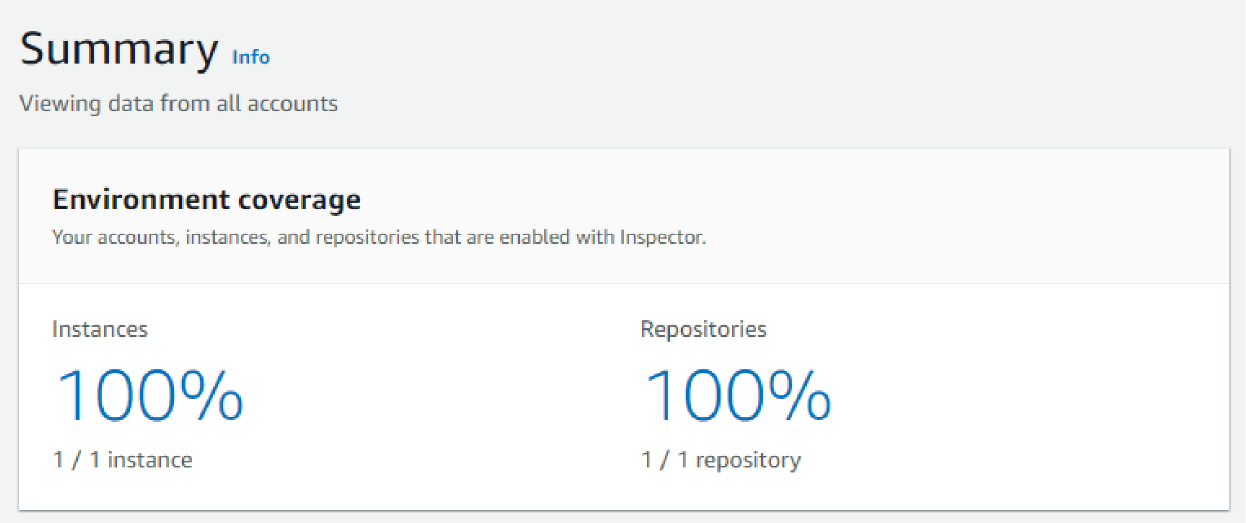


Images (1)

□	Image tag	Artifact type	Pushed at	Size (MB)
□	latest	Image	September 06, 2022, 18:12:42 (UTC+05.5)	228.91

Step 41: Navigate back to the Inspector dashboard and check out the environment coverage.

Now we have one instance and one repository with 100% coverage.



Summary Info

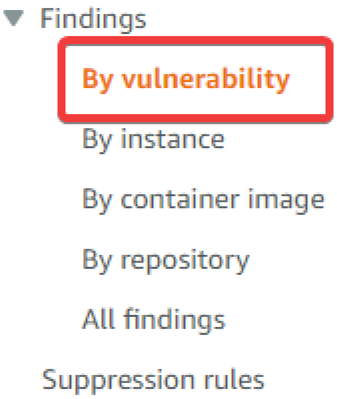
Viewing data from all accounts

Environment coverage

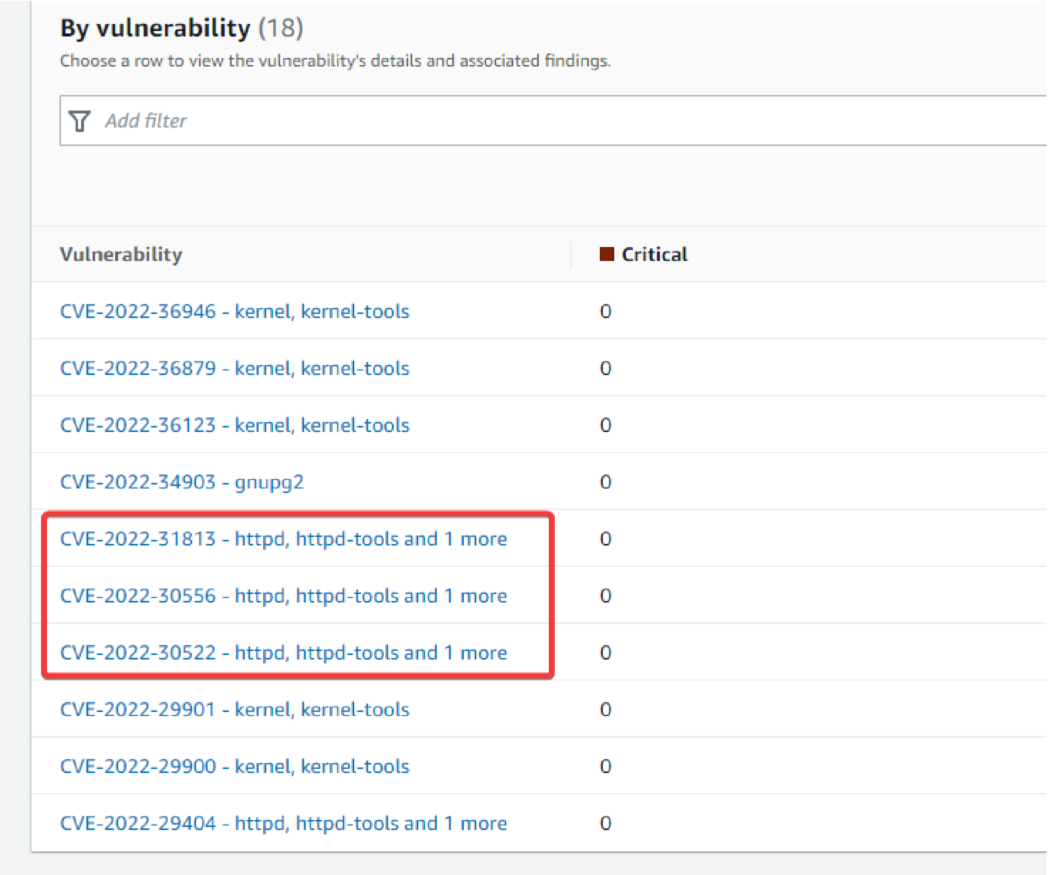
Your accounts, instances, and repositories that are enabled with Inspector.

Instances	Repositories
100%	100%
1 / 1 instance	1 / 1 repository

Step 42: Click on "By vulnerability" under findings in the navigation pane.



Notice the vulnerability detected by the inspector. The following vulnerabilities are related to the httpd package that we have installed in the instance and repository.

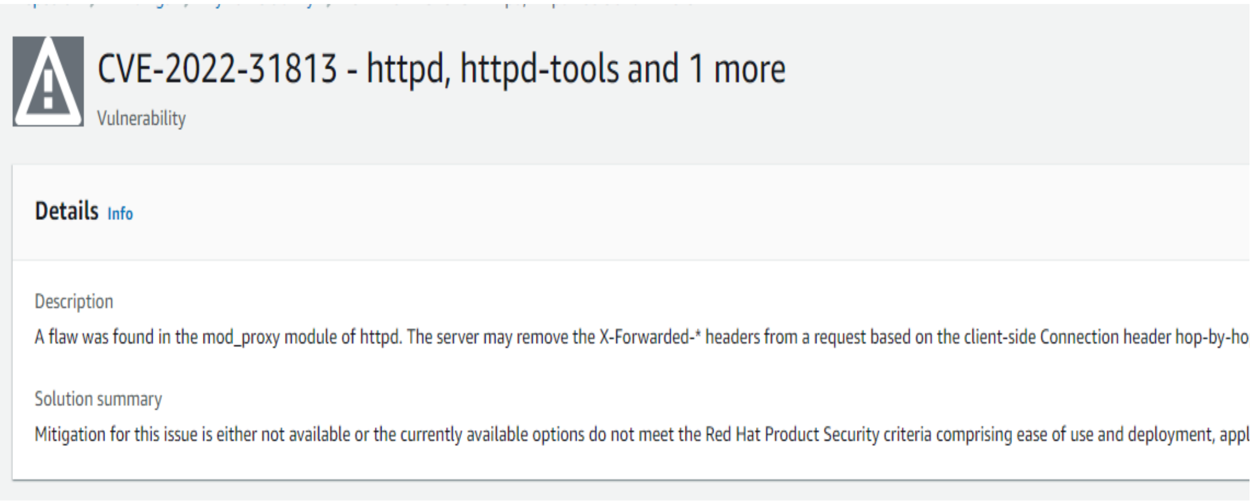


By vulnerability (18)

Choose a row to view the vulnerability's details and associated findings.

Vulnerability	■ Critical
CVE-2022-36946 - kernel, kernel-tools	0
CVE-2022-36879 - kernel, kernel-tools	0
CVE-2022-36123 - kernel, kernel-tools	0
CVE-2022-34903 - gnupg2	0
CVE-2022-31813 - httpd, httpd-tools and 1 more	0
CVE-2022-30556 - httpd, httpd-tools and 1 more	0
CVE-2022-30522 - httpd, httpd-tools and 1 more	0
CVE-2022-29901 - kernel, kernel-tools	0
CVE-2022-29900 - kernel, kernel-tools	0
CVE-2022-29404 - httpd, httpd-tools and 1 more	0

Step 43: Click on “CVE-2022-31813” to get more information about the detected vulnerability.



CVE-2022-31813 - httpd, httpd-tools and 1 more

Vulnerability

Details Info

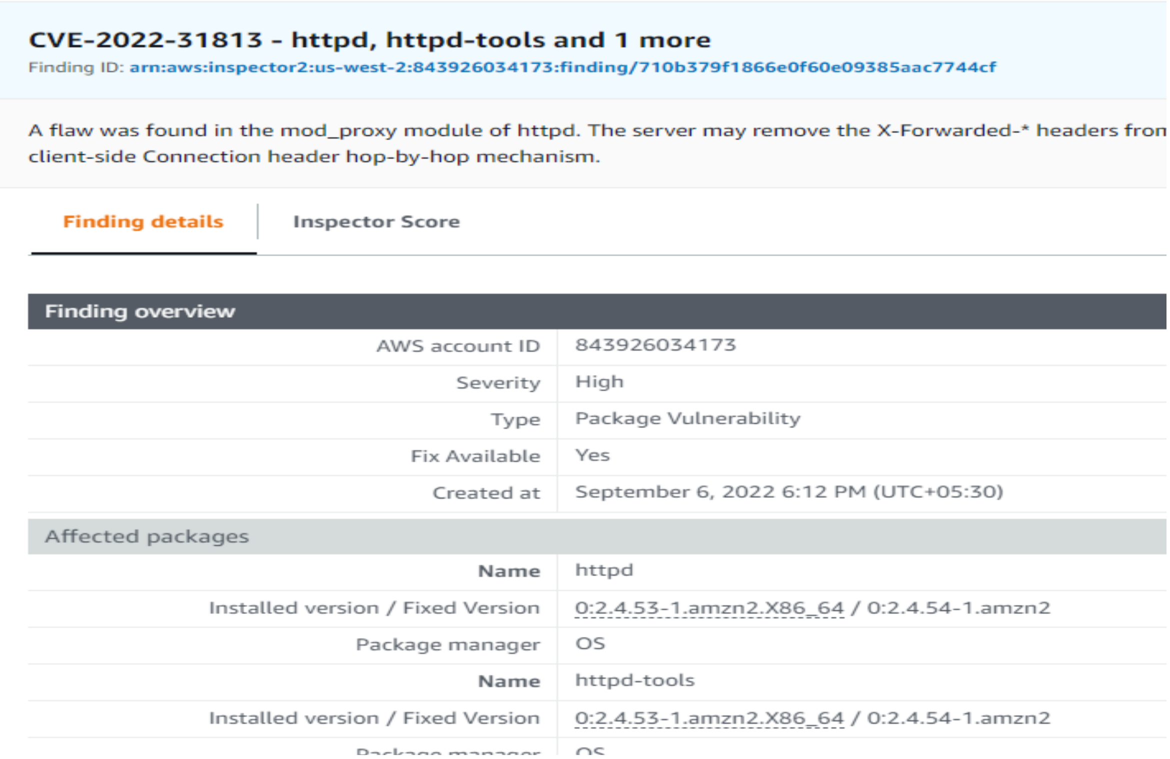
Description

A flaw was found in the mod_proxy module of httpd. The server may remove the X-Forwarded-* headers from a request based on the client-side Connection header hop-by-ho

Solution summary

Mitigation for this issue is either not available or the currently available options do not meet the Red Hat Product Security criteria comprising ease of use and deployment, appl

Click on the title and check the finding details.



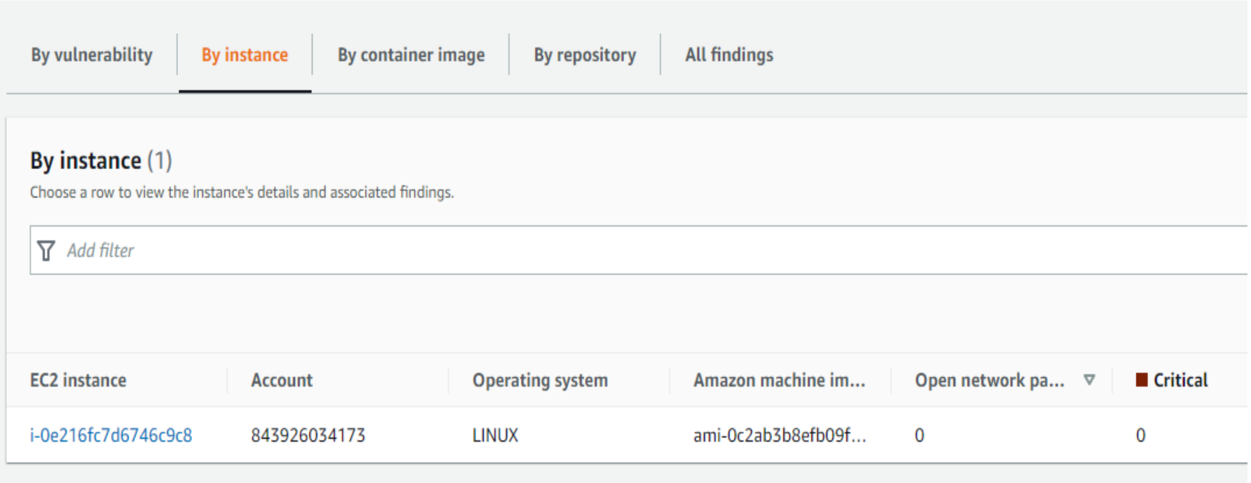
CVE-2022-31813 - httpd, httpd-tools and 1 more

Finding ID: arn:aws:inspector2:us-west-2:843926034173:finding/710b379f1866e0f60e09385aac7744cf

A flaw was found in the mod_proxy module of httpd. The server may remove the X-Forwarded-* headers fron client-side Connection header hop-by-hop mechanism.

Finding overview	
AWS account ID	843926034173
Severity	High
Type	Package Vulnerability
Fix Available	Yes
Created at	September 6, 2022 6:12 PM (UTC+05:30)
Affected packages	
Name	httpd
Installed version / Fixed Version	0:2.4.53-1.amzn2.X86_64 / 0:2.4.54-1.amzn2
Package manager	OS
Name	httpd-tools
Installed version / Fixed Version	0:2.4.53-1.amzn2.X86_64 / 0:2.4.54-1.amzn2
Package manager	OS

Click on “By instance” to get the vulnerability details from the instance.

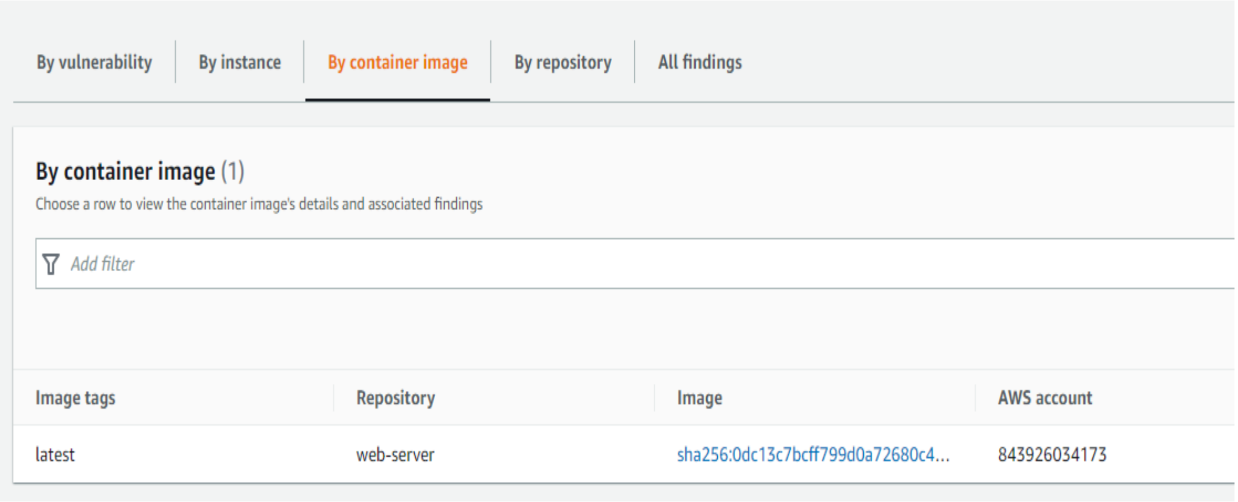


By instance (1)

Choose a row to view the instance's details and associated findings.

EC2 instance	Account	Operating system	Amazon machine im...	Open network pa... ▽	🟥 Critical
i-0e216fc7d6746c9c8	843926034173	LINUX	ami-0c2ab3b8efb09f...	0	0

Click on “By container image” to get the vulnerability details from the image.

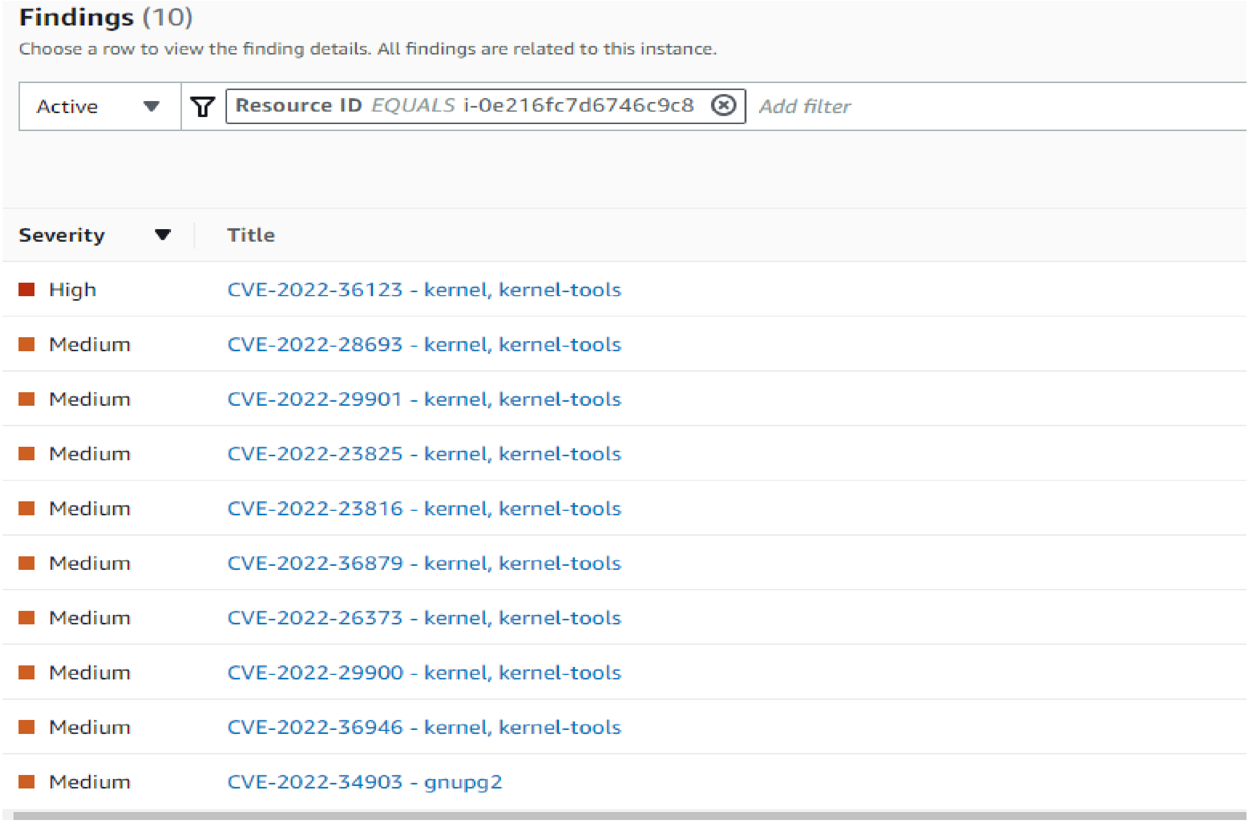


By container image (1)

Choose a row to view the container image's details and associated findings

Image tags	Repository	Image	AWS account
latest	web-server	sha256:0dc13c7bcff799d0a72680c4...	843926034173

Click on “All findings” to get all the vulnerability details.



Findings (10)

Choose a row to view the finding details. All findings are related to this instance.

Active ▼ | Resource ID *EQUALS* i-0e216fc7d6746c9c8 ⊗ | *Add filter*

Severity ▼	Title
🟥 High	CVE-2022-36123 - kernel, kernel-tools
🟧 Medium	CVE-2022-28693 - kernel, kernel-tools
🟧 Medium	CVE-2022-29901 - kernel, kernel-tools
🟧 Medium	CVE-2022-23825 - kernel, kernel-tools
🟧 Medium	CVE-2022-23816 - kernel, kernel-tools
🟧 Medium	CVE-2022-36879 - kernel, kernel-tools
🟧 Medium	CVE-2022-26373 - kernel, kernel-tools
🟧 Medium	CVE-2022-29900 - kernel, kernel-tools
🟧 Medium	CVE-2022-36946 - kernel, kernel-tools
🟧 Medium	CVE-2022-34903 - gnupg2

Successfully enabled Amazon Inspector and detected the vulnerabilities from the instance and container image.

References:

1. Amazon Inspector (https://docs.aws.amazon.com/inspector/latest/user/what-is-inspector.html)
2. CVE-2022-31813 (https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-31813)