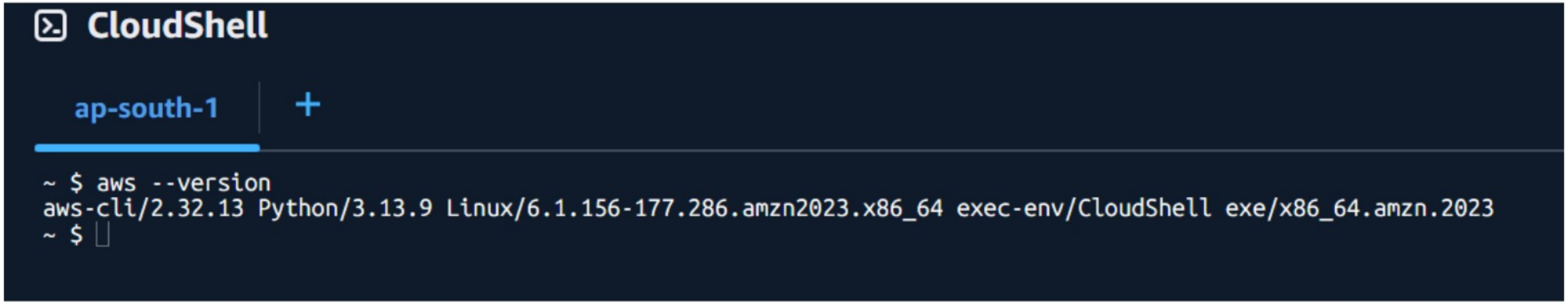
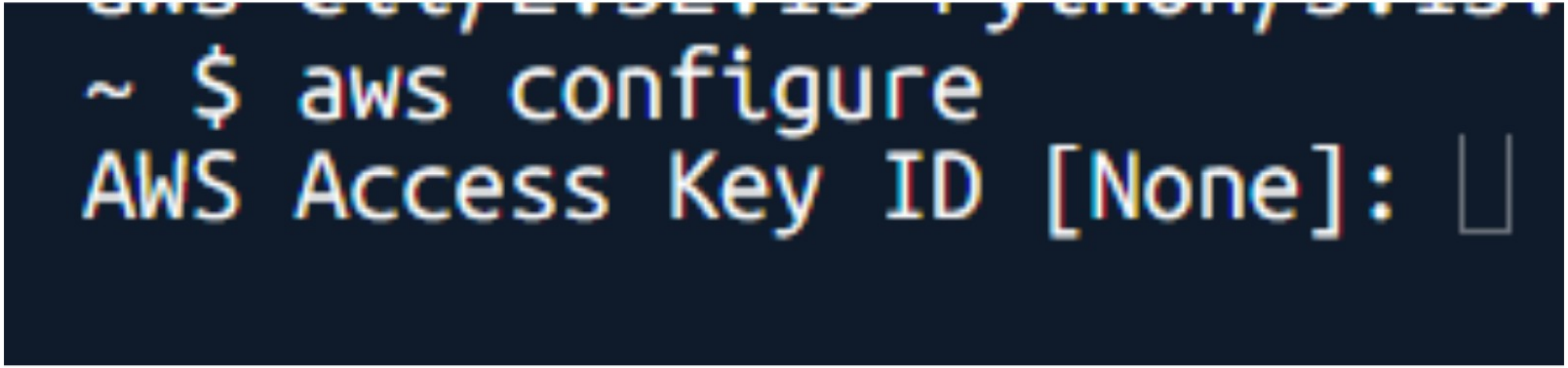
# 2. Store Docker images in ECR

# 1. Check the version of aws cli



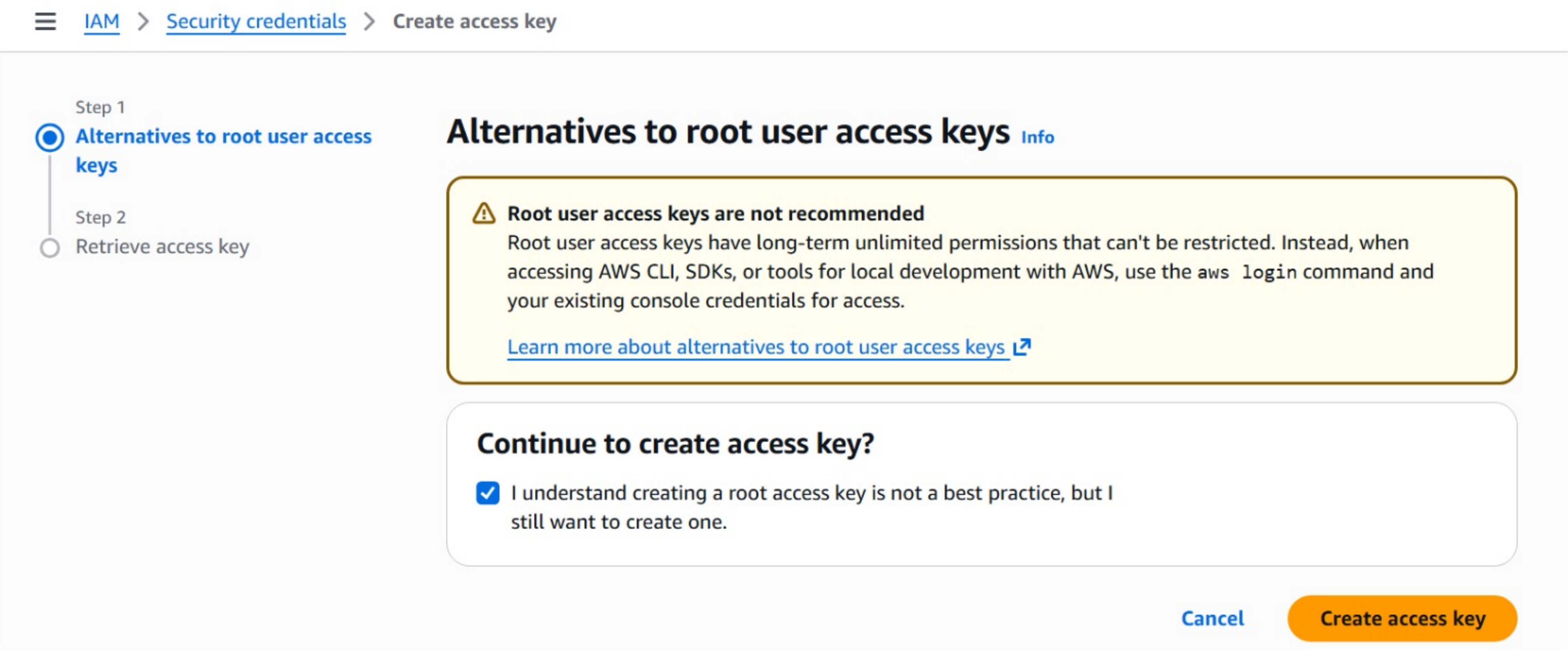
```
~ $ aws --version
aws-cli/2.32.13 Python/3.13.9 Linux/6.1.156-177.286.amzn2023.x86_64 exec-env/CloudShell exe/x86_64.amzn.2023
~ $
```

# 2. Run amazon configure command



```
~ $ aws configure
AWS Access Key ID [None]: 
```

# 3. Create access key



☰ IAM > Security credentials > Create access key

Step 1
● Alternatives to root user access keys
Step 2
○ Retrieve access key

## Alternatives to root user access keys

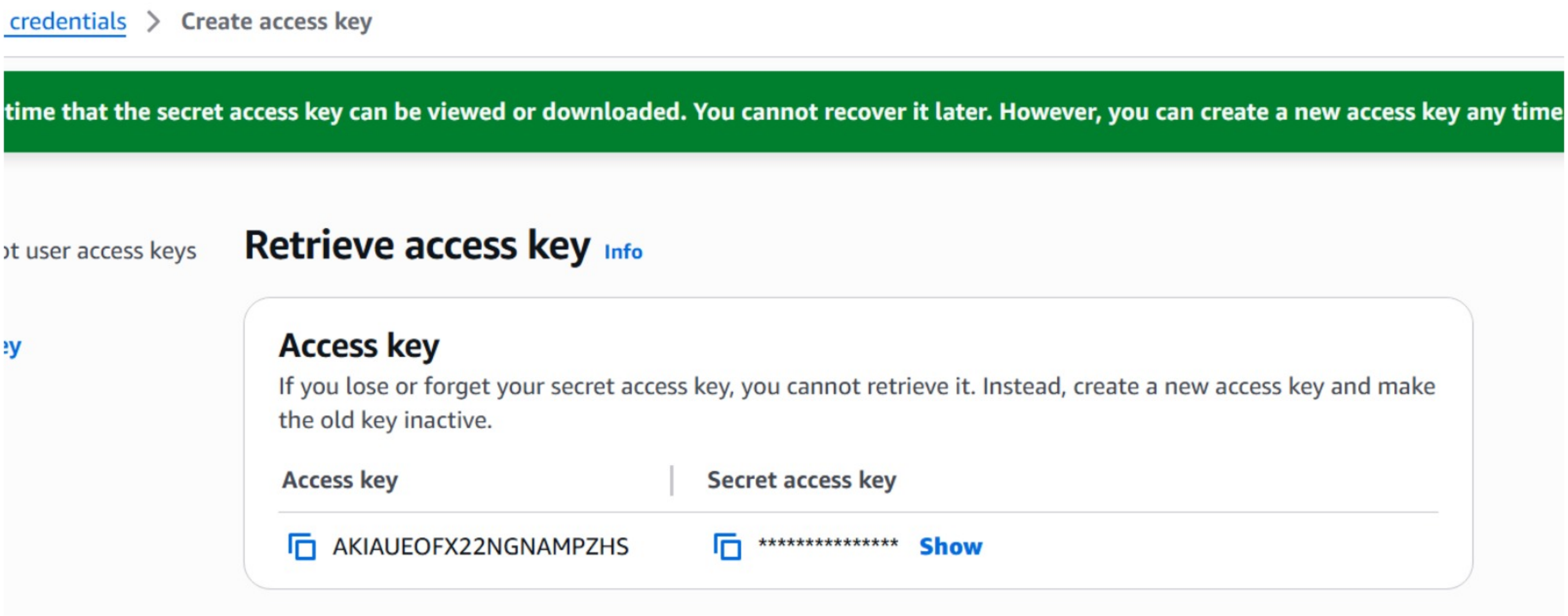
**Root user access keys are not recommended**
Root user access keys have long-term unlimited permissions that can't be restricted. Instead, when accessing AWS CLI, SDKs, or tools for local development with AWS, use the `aws login` command and your existing console credentials for access.

Learn more about alternatives to root user access keys

### Continue to create access key?

☑ I understand creating a root access key is not a best practice, but I still want to create one.

# 4. Copy acess key and secret



credentials > Create access key

time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time

t user access keys

ey

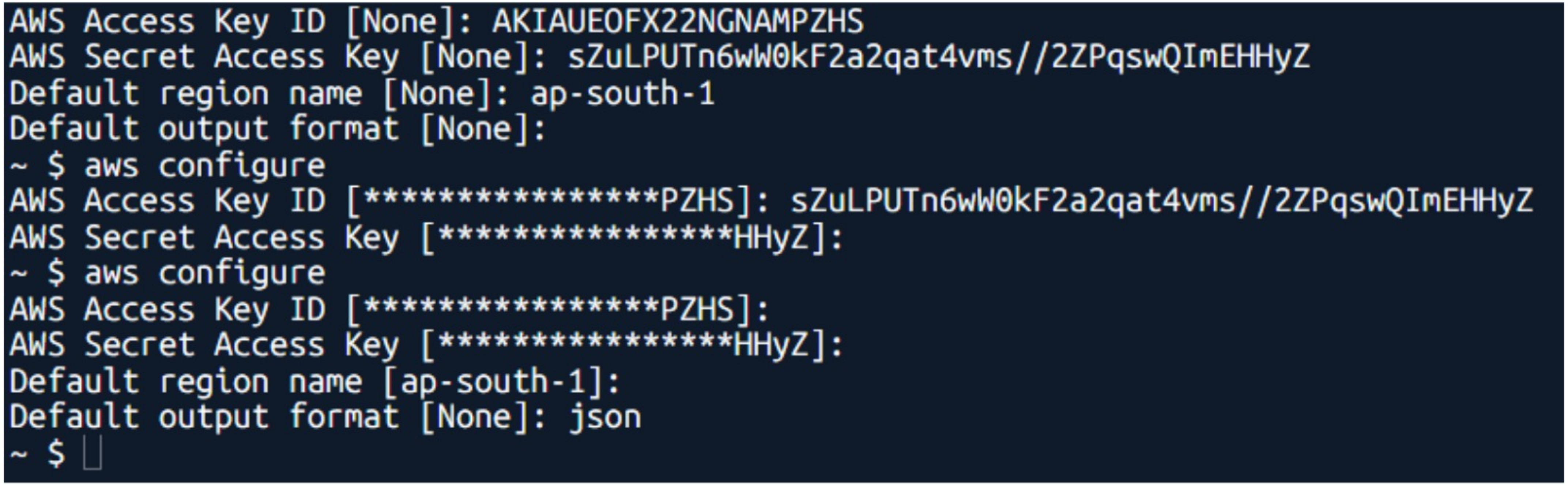
## Retrieve access key

### Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

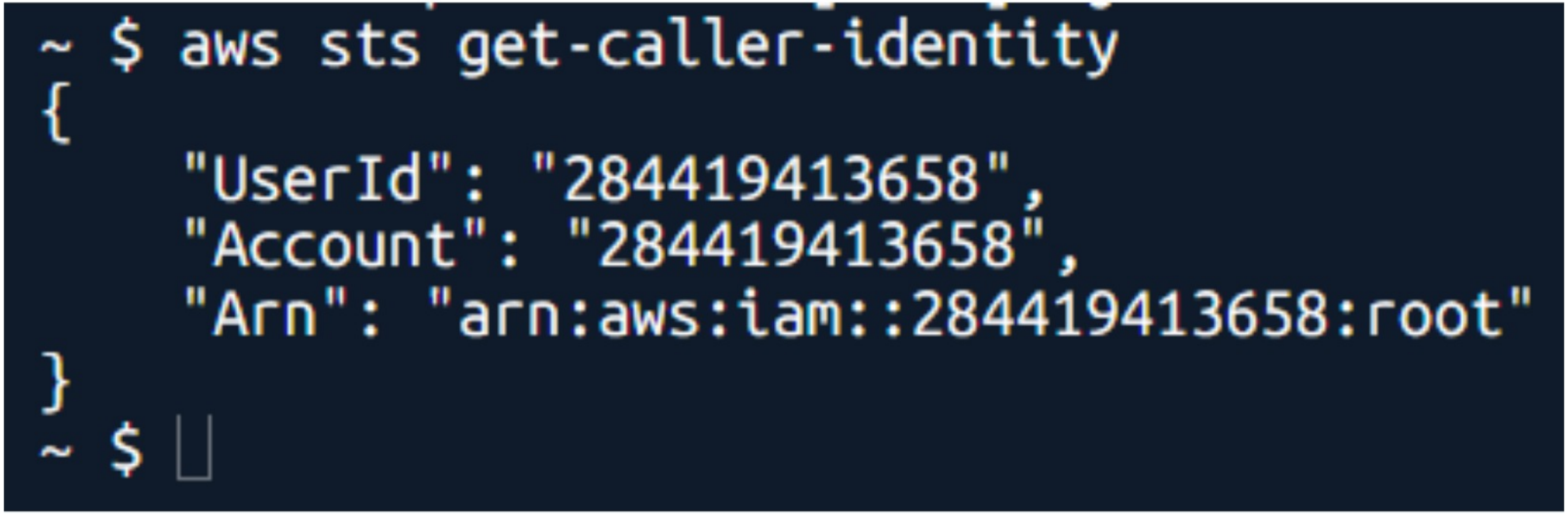
| Access key | Secret access key |
|---|---|
| AKIAUEOFX22NGNAMPZHS | *************** Show |

# 5. Paste it on



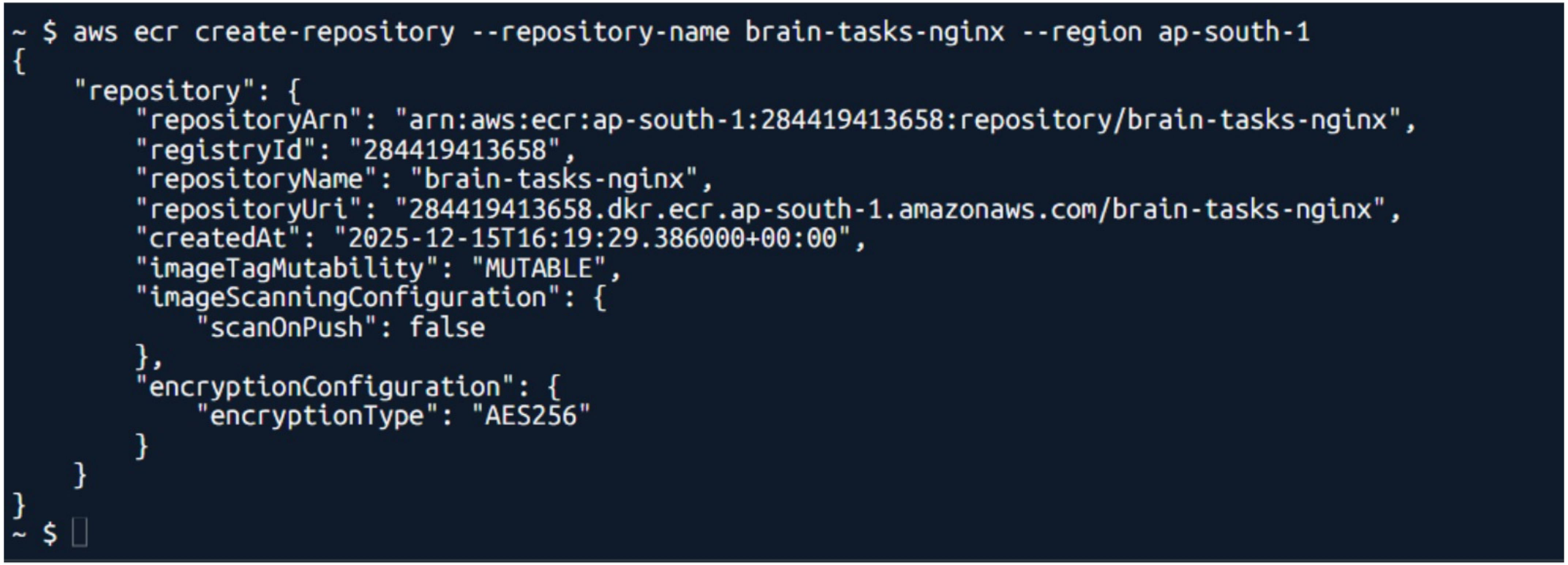
```
AWS Access Key ID [None]: AKIAUEOFX22NGNAMPZHS
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]:
~ $ aws configure
AWS Access Key ID [****************PZHS]: sZuLPUTn6wW0kF2a2qat4vms//2ZPqswQImEHHyZ
AWS Secret Access Key [****************HHyZ]:
~ $ aws configure
AWS Access Key ID [****************PZHS]:
AWS Secret Access Key [****************HHyZ]:
Default region name [ap-south-1]:
Default output format [None]: json
~ $
```

# 6. Verify credentials are working



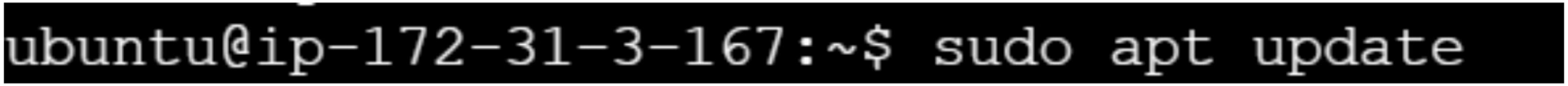
```
~ $ aws sts get-caller-identity
{
    "UserId": "284419413658",
    "Account": "284419413658",
    "Arn": "arn:aws:iam::284419413658:root"
}
~ $ 
```

# 7. Create ECR repository



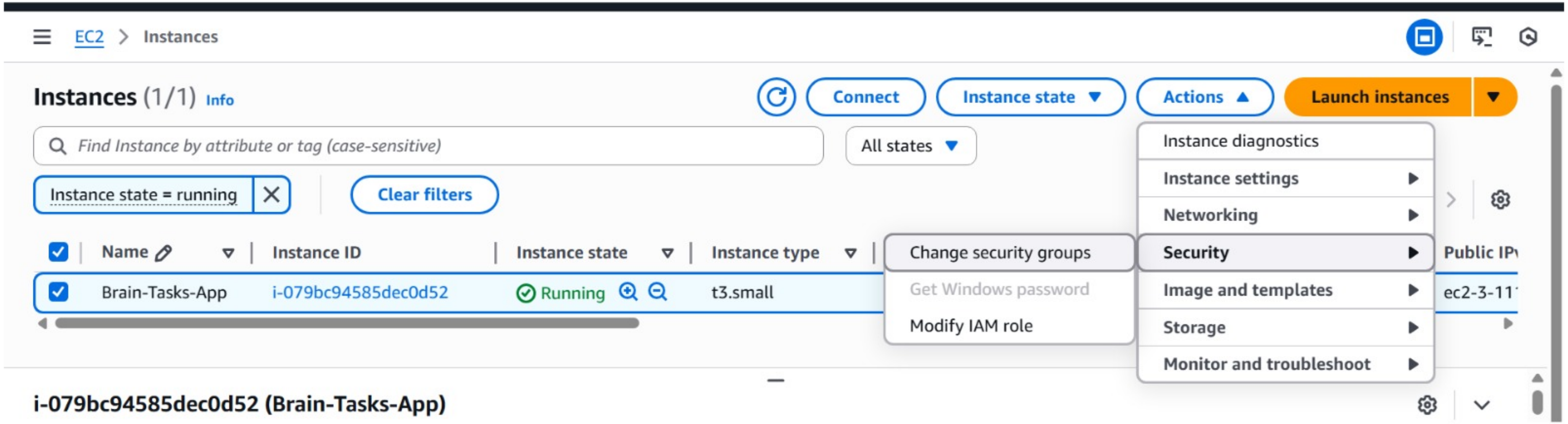
```
~ $ aws ecr create-repository --repository-name brain-tasks-nginx --region ap-south-1
{
    "repository": {
        "repositoryArn": "arn:aws:ecr:ap-south-1:284419413658:repository/brain-tasks-nginx",
        "registryId": "284419413658",
        "repositoryName": "brain-tasks-nginx",
        "repositoryUri": "284419413658.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-nginx",
        "createdAt": "2025-12-15T16:19:29.386000+00:00",
        "imageTagMutability": "MUTABLE",
        "imageScanningConfiguration": {
            "scanOnPush": false
        },
        "encryptionConfiguration": {
            "encryptionType": "AES256"
        }
    }
}
~ $ 
```

# 8. Update the packages



```
ubuntu@ip-172-31-3-167:~$ sudo apt update
```

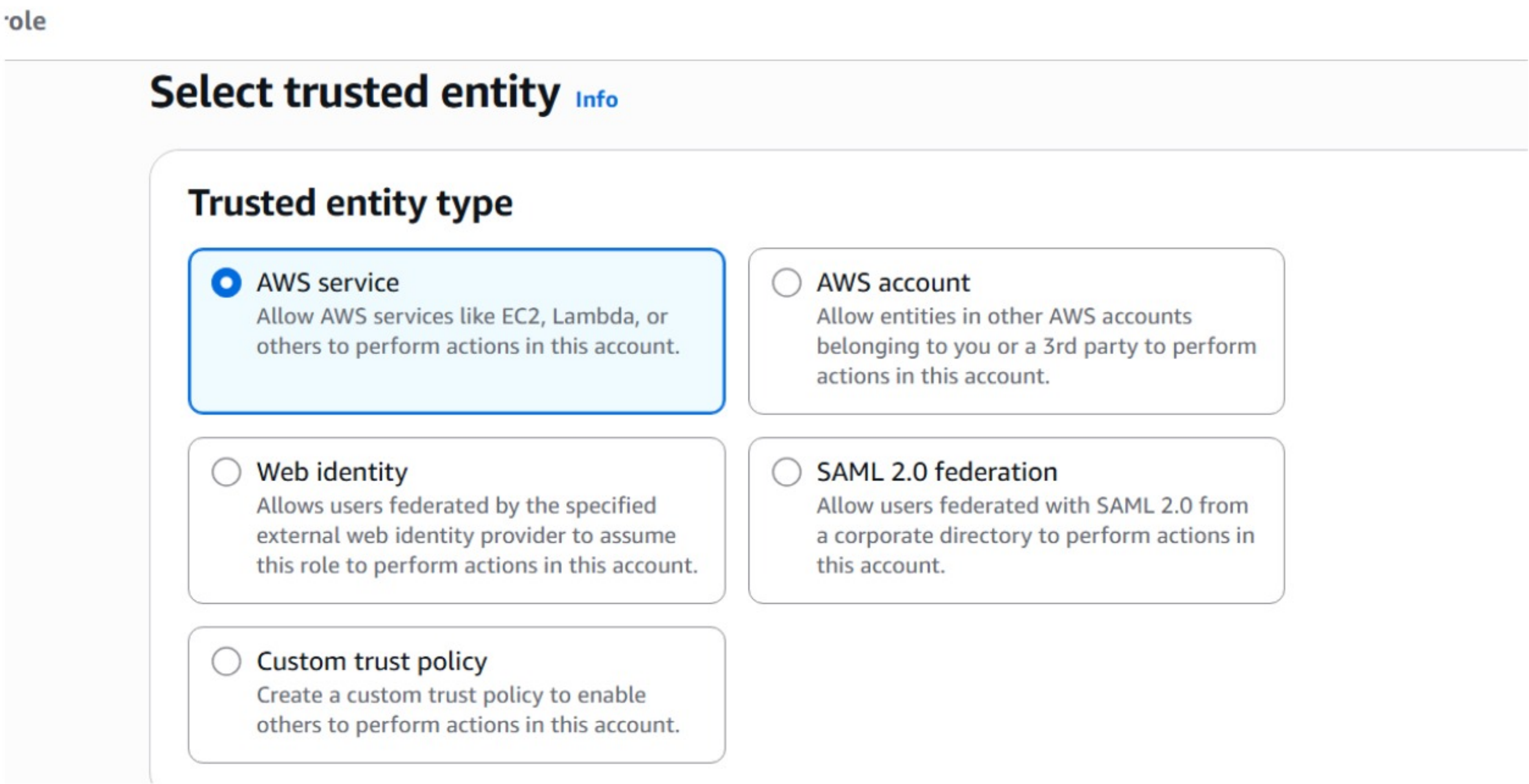
# 9. Select ec2 instance and go to modify iam role option



| ☑ | Name | Instance ID | Instance state | Instance type | Public IP |
|---|---|---|---|---|---|
| ☑ | Brain-Tasks-App | i-079bc94585dec0d52 | Running | t3.small | ec2-3-11 |

i-079bc94585dec0d52 (Brain-Tasks-App)

# 10. Create a role



ole

## Select trusted entity Info

### Trusted entity type

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

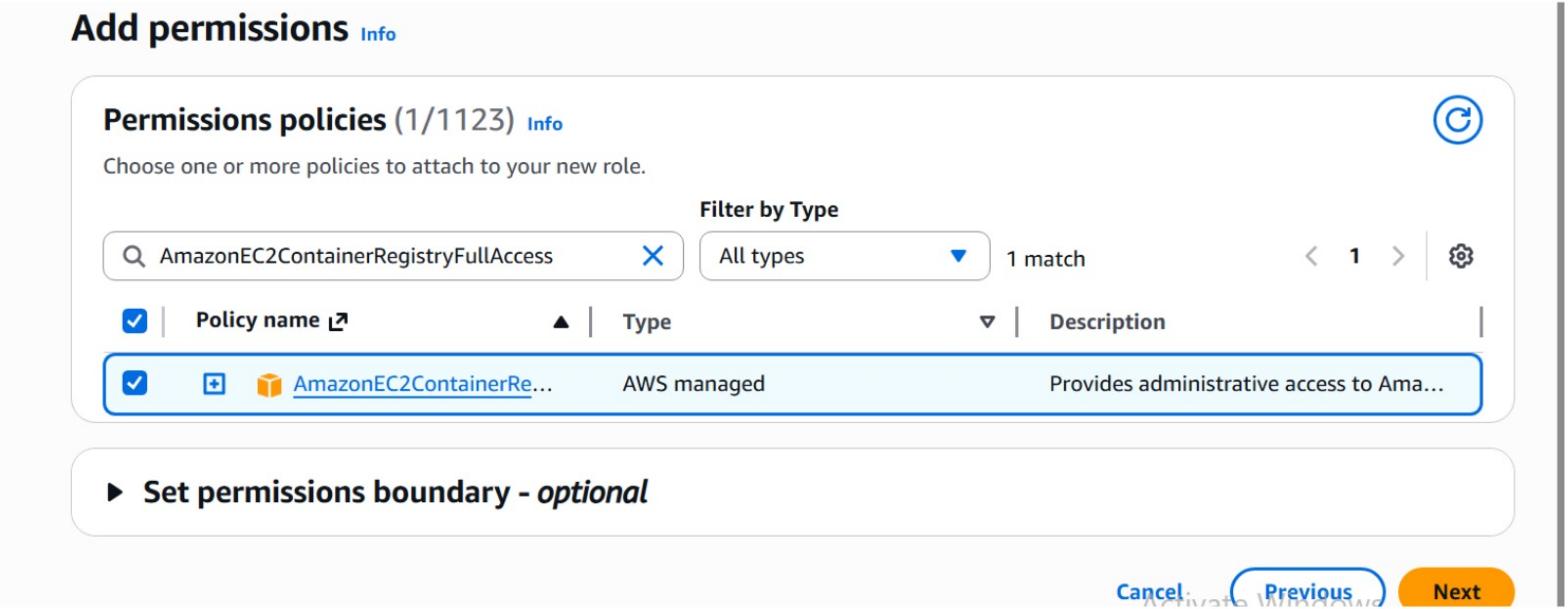
○ AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

○ 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 others to perform actions in this account.

# 11. Attach required permissions



## Add permissions Info

### Permissions policies (1/1123) Info

Choose one or more policies to attach to your new role.

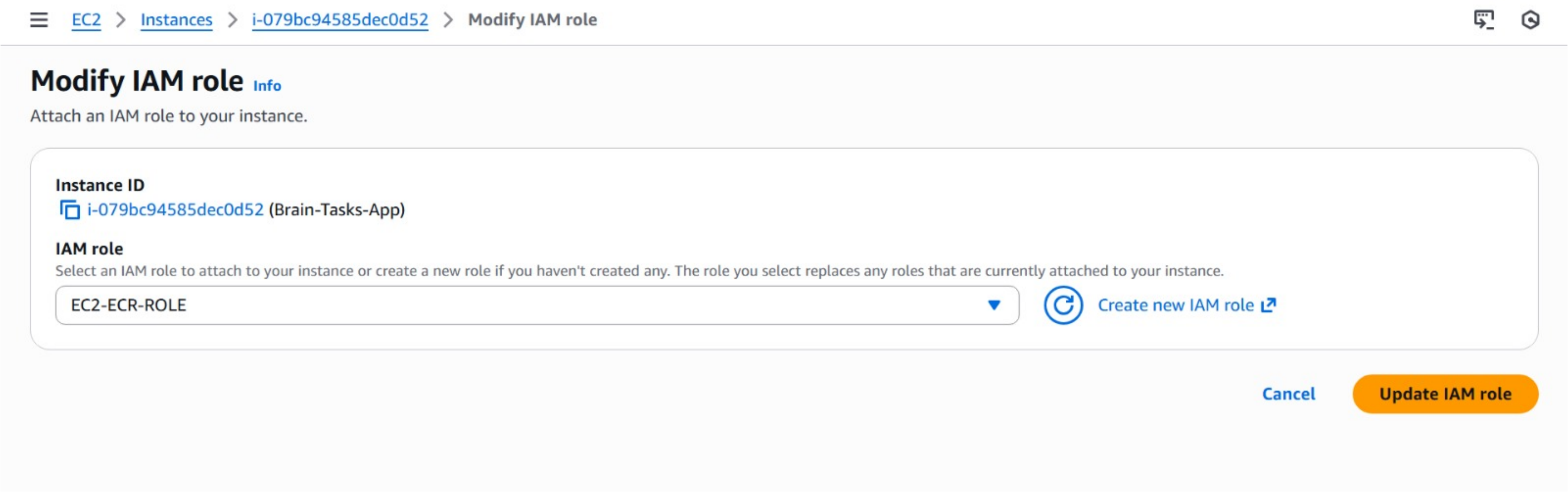
Filter by Type

AmazonEC2ContainerRegistryFullAccess ✕ | All types ▼ | 1 match

| ☑ | Policy name ▲ | Type | Description |
|---|---|---|---|
| ☑ | AmazonEC2ContainerRe... | AWS managed | Provides administrative access to Ama... |

▶ Set permissions boundary - *optional*

# 12. Add iam role to ec2 instance



Modify IAM role Info

Attach an IAM role to your instance.

Instance ID

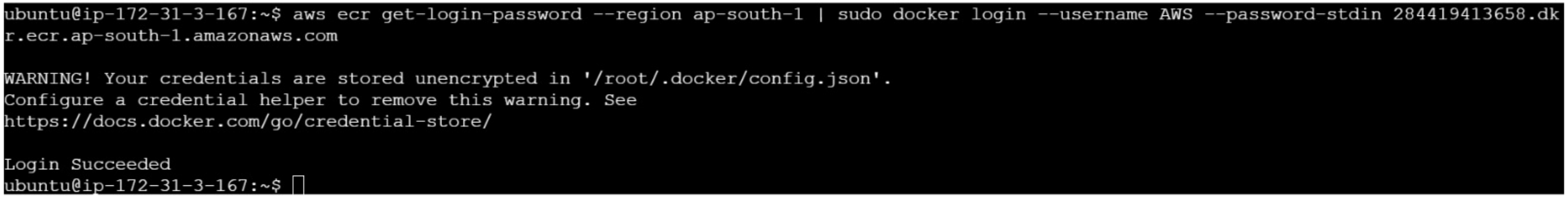
i-079bc94585dec0d52 (Brain-Tasks-App)

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.

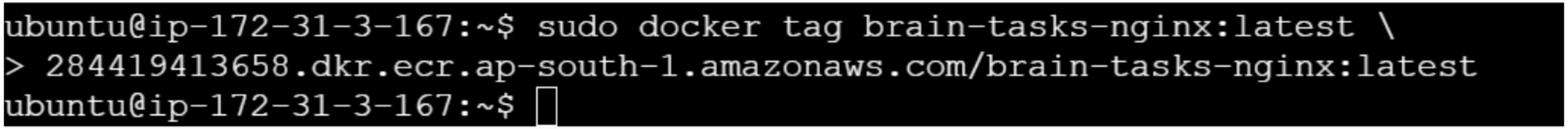
EC2-ECR-ROLE

# 13. In ec2 login to ECR



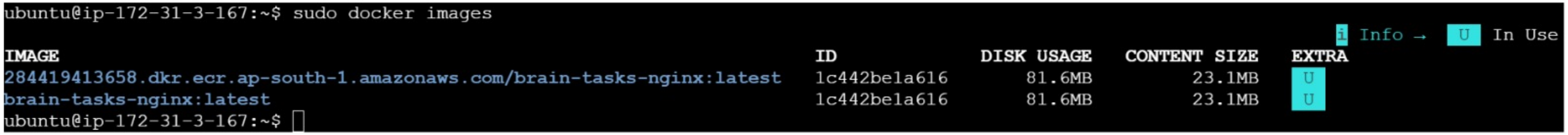
```
ubuntu@ip-172-31-3-167:~$ aws ecr get-login-password --region ap-south-1 | sudo docker login --username AWS --password-stdin 284419413658.dk
r.ecr.ap-south-1.amazonaws.com

WARNING! Your credentials are stored unencrypted in '/root/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
ubuntu@ip-172-31-3-167:~$ 
```

# 14. Tag the docker image



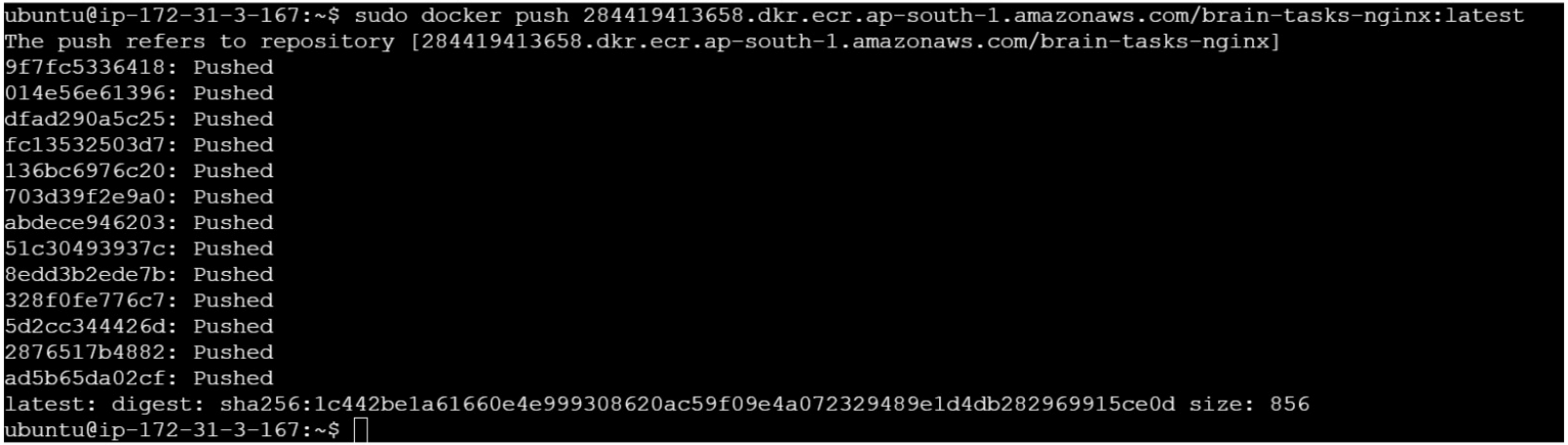
```
ubuntu@ip-172-31-3-167:~$ sudo docker tag brain-tasks-nginx:latest \
> 284419413658.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-nginx:latest
ubuntu@ip-172-31-3-167:~$ 
```

# 15. Check the status of images



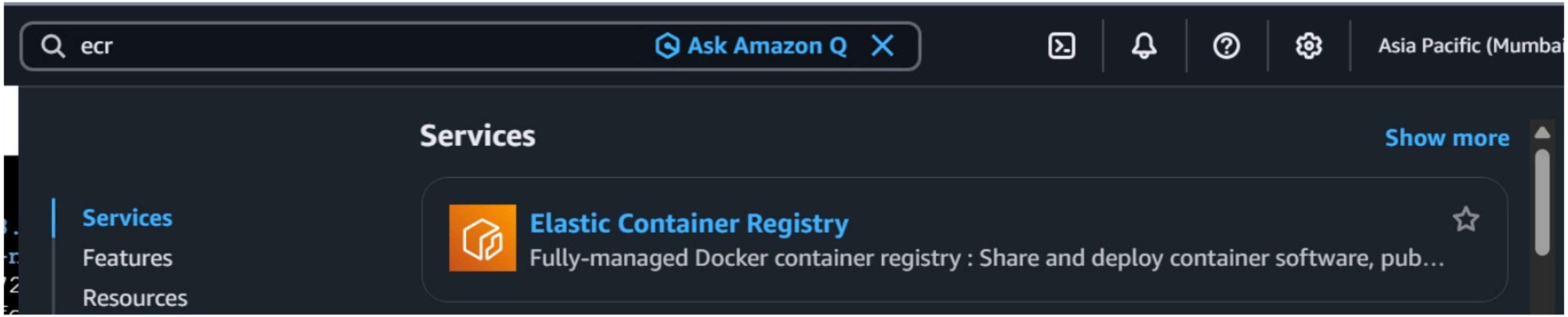
```
ubuntu@ip-172-31-3-167:~$ sudo docker images
                                                                                                                    i Info →   U  In Use
IMAGE                                                                    ID             DISK USAGE   CONTENT SIZE   EXTRA
284419413658.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-nginx:latest   1c442be1a616       81.6MB         23.1MB    U
brain-tasks-nginx:latest                                                 1c442be1a616       81.6MB         23.1MB    U
ubuntu@ip-172-31-3-167:~$
```

# 16. Push the image to ECR



```
ubuntu@ip-172-31-3-167:~$ sudo docker push 284419413658.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-nginx:latest
The push refers to repository [284419413658.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-nginx]
9f7fc5336418: Pushed
014e56e61396: Pushed
dfad290a5c25: Pushed
fc13532503d7: Pushed
136bc6976c20: Pushed
703d39f2e9a0: Pushed
abdece946203: Pushed
51c30493937c: Pushed
8edd3b2ede7b: Pushed
328f0fe776c7: Pushed
5d2cc344426d: Pushed
2876517b4882: Pushed
ad5b65da02cf: Pushed
latest: digest: sha256:1c442be1a61660e4e999308620ac59f09e4a072329489e1d4db282969915ce0d size: 856
ubuntu@ip-172-31-3-167:~$
```

# 17. Go to ECR page

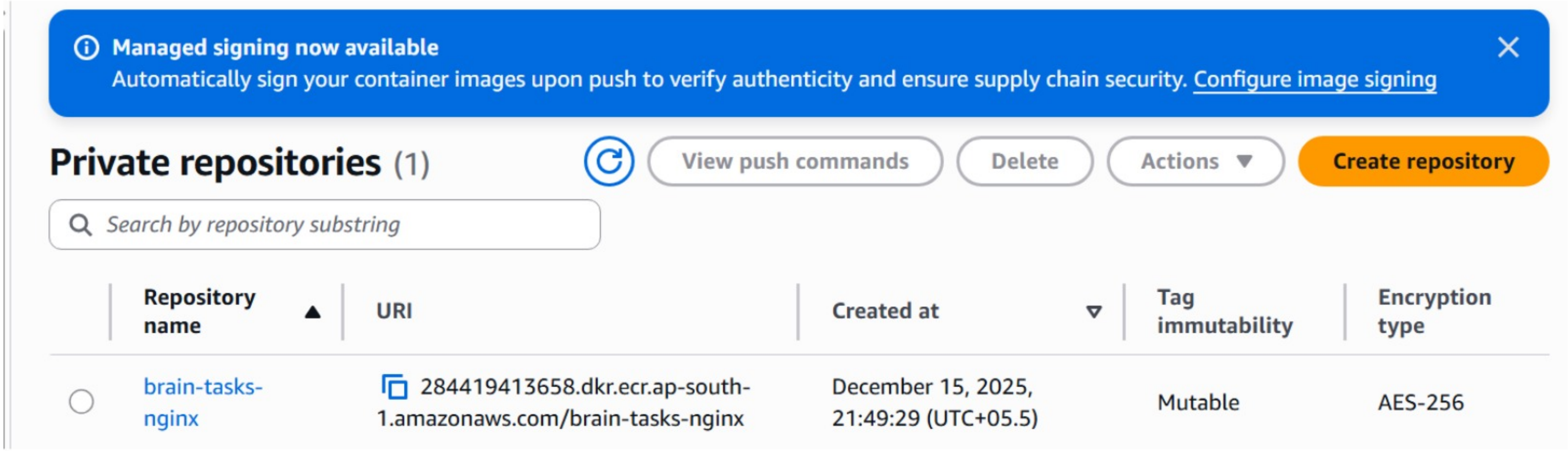


Services

Elastic Container Registry

Fully-managed Docker container registry : Share and deploy container software, pub…

# 18. Here ECR repository created



Automatically sign your container images upon push to verify authenticity and ensure supply chain security. Configure image signing

## Private repositories (1)

| | Repository name ▲ | URI | Created at ▽ | Tag immutability | Encryption type |
|---|---|---|---|---|---|
| ○ | brain-tasks-nginx | 284419413658.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-nginx | December 15, 2025, 21:49:29 (UTC+05.5) | Mutable | AES-256 |