black hat
BRIEFINGS
AUGUST 6-7, 2025
MANDALAY BAY / LAS VEGAS

ECScape

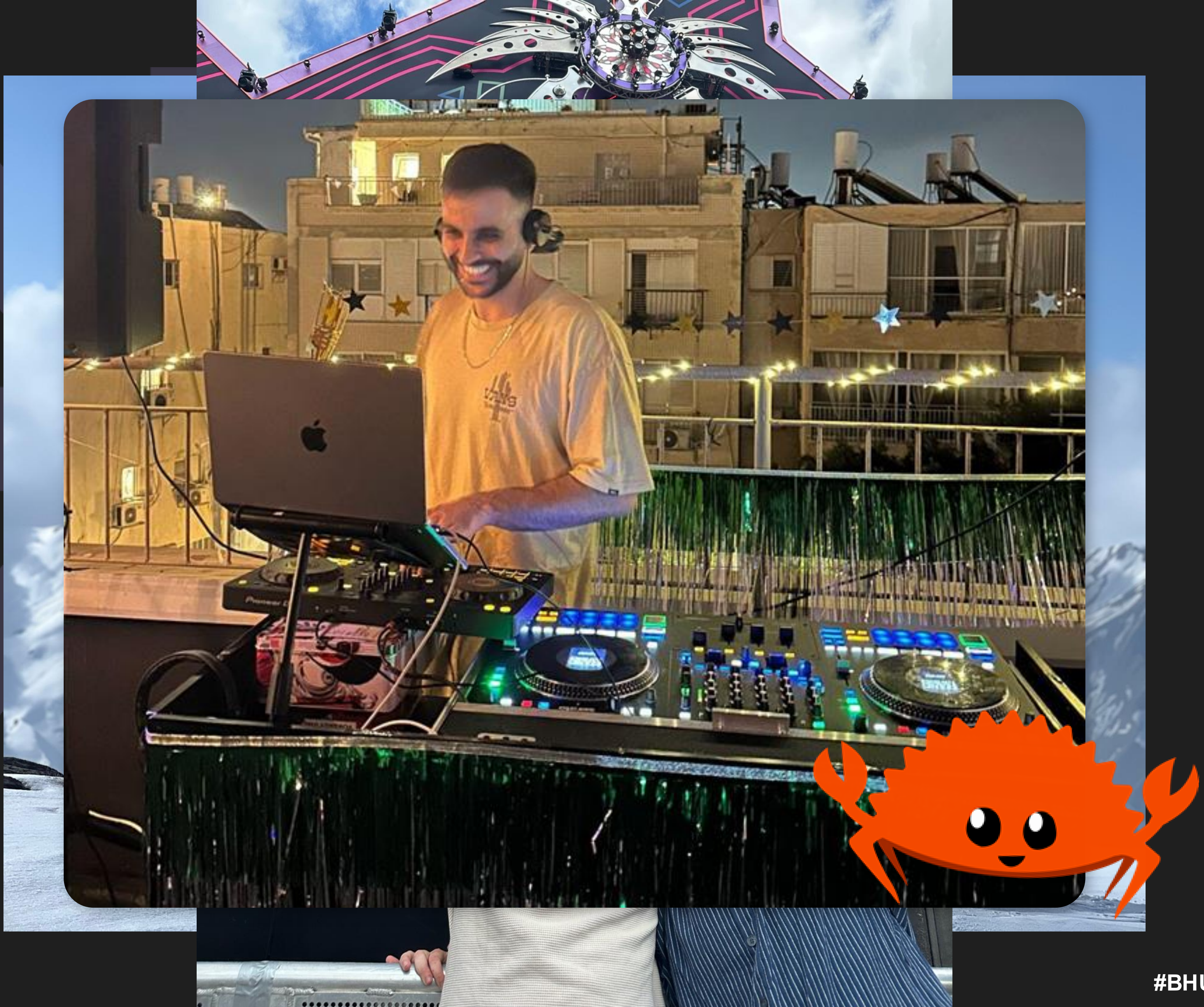
Expanding Privileges in the Cloud:

Exploring Security Boundaries in Amazon ECS

$whoami

Naor Haziz

- Israel IL
- Software Developer
- Security Researcher
- Sweet Security



Agenda

01 Technical Background

02 Story Time

03 ECScape

04 Impact

05 Demo

06 Mitigation

33%

Of Developers Using Orchestration Technologies rely on Amazon ECS

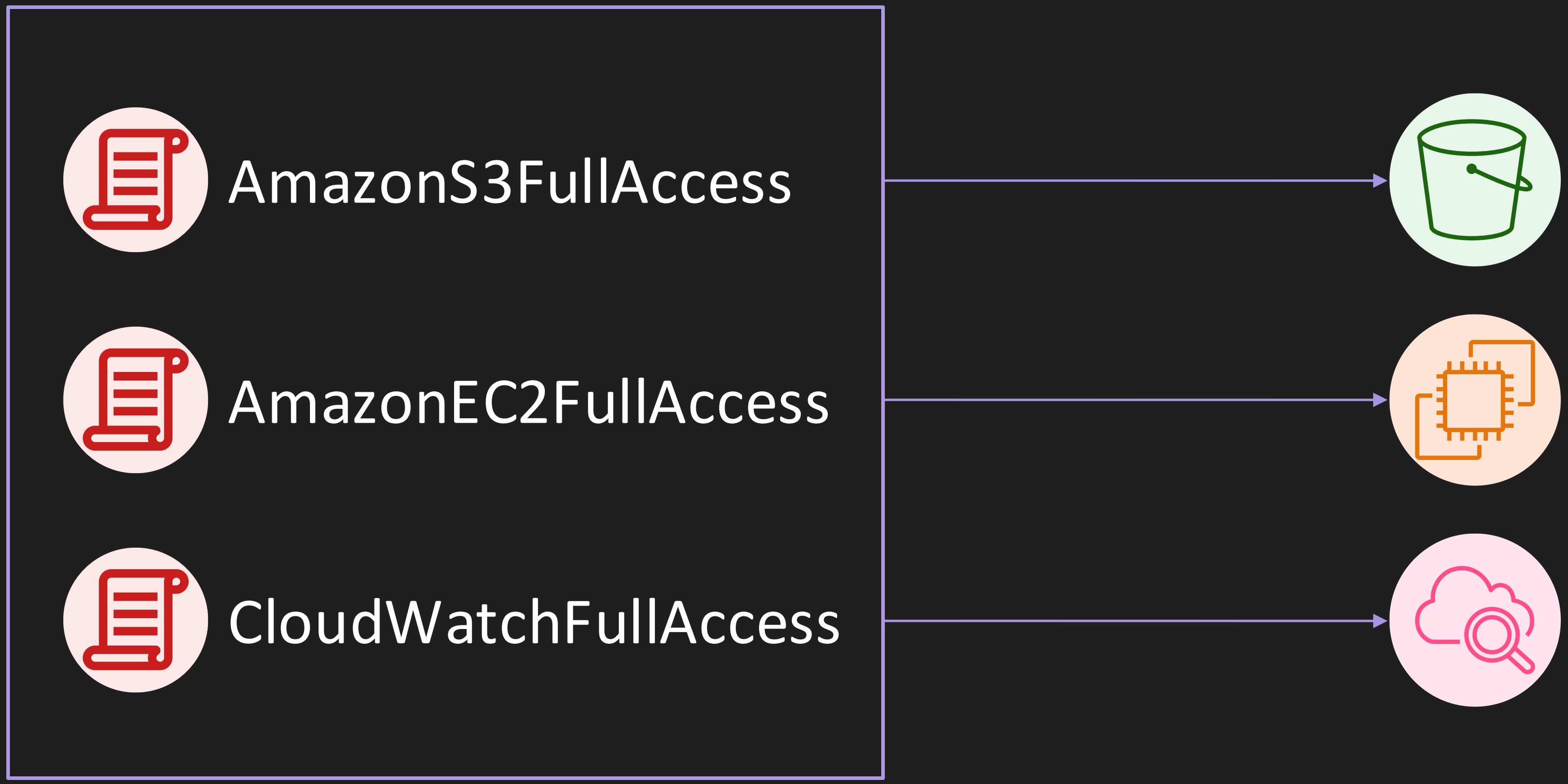
CNCF Annual Survey 2021

01
Technical Background

What is IAM?

Role

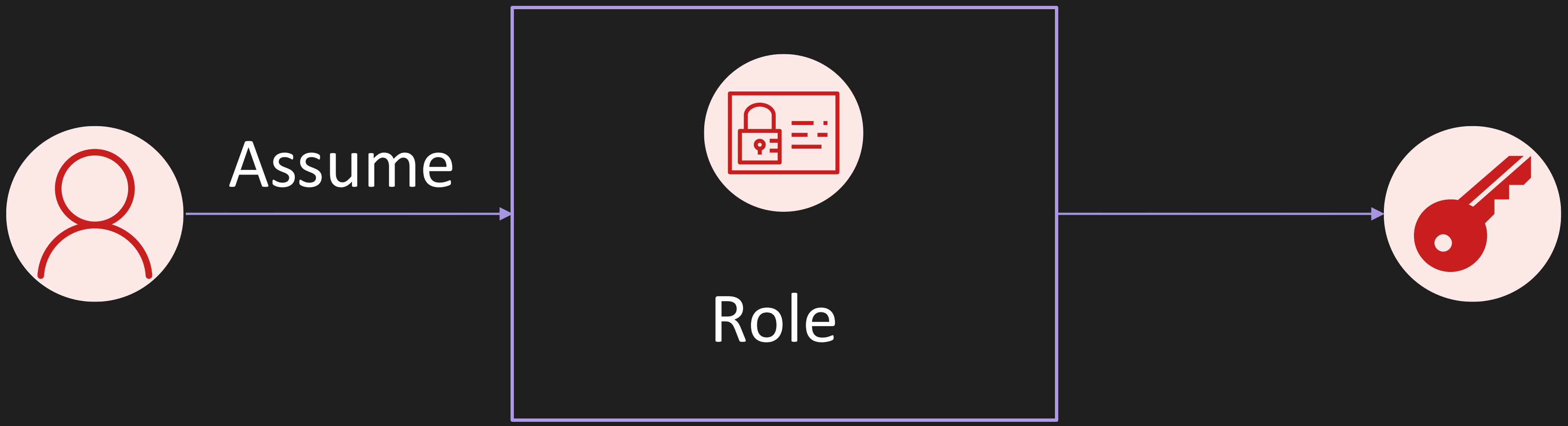
Policy

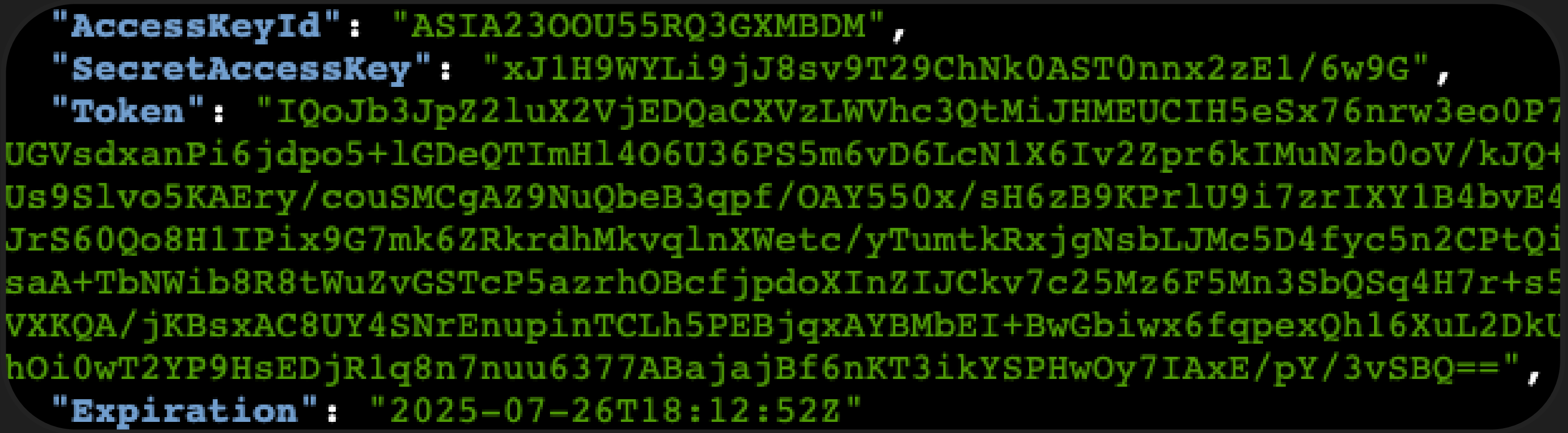


AmazonS3FullAccess

AmazonEC2FullAccess

CloudWatchFullAccess

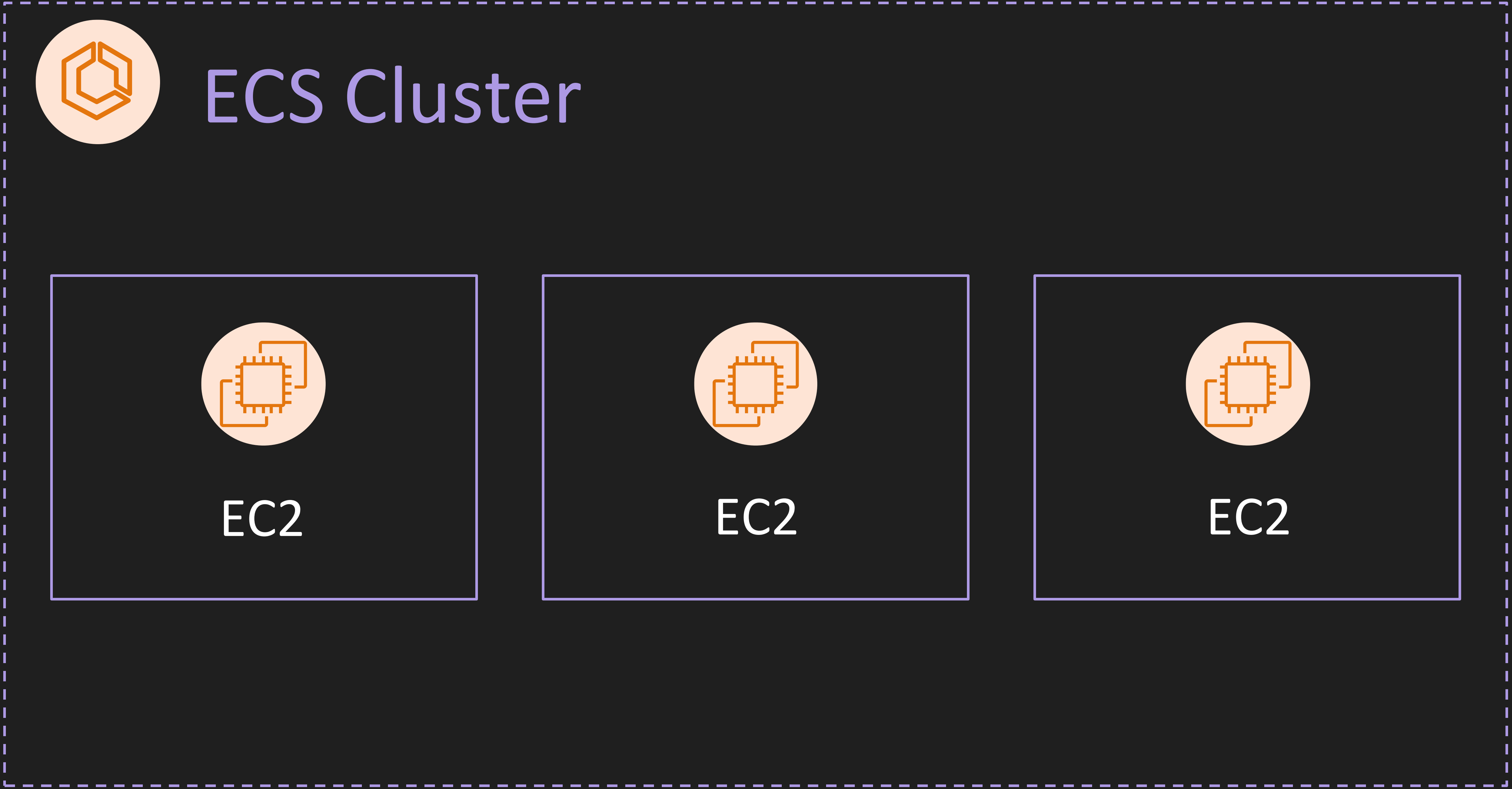


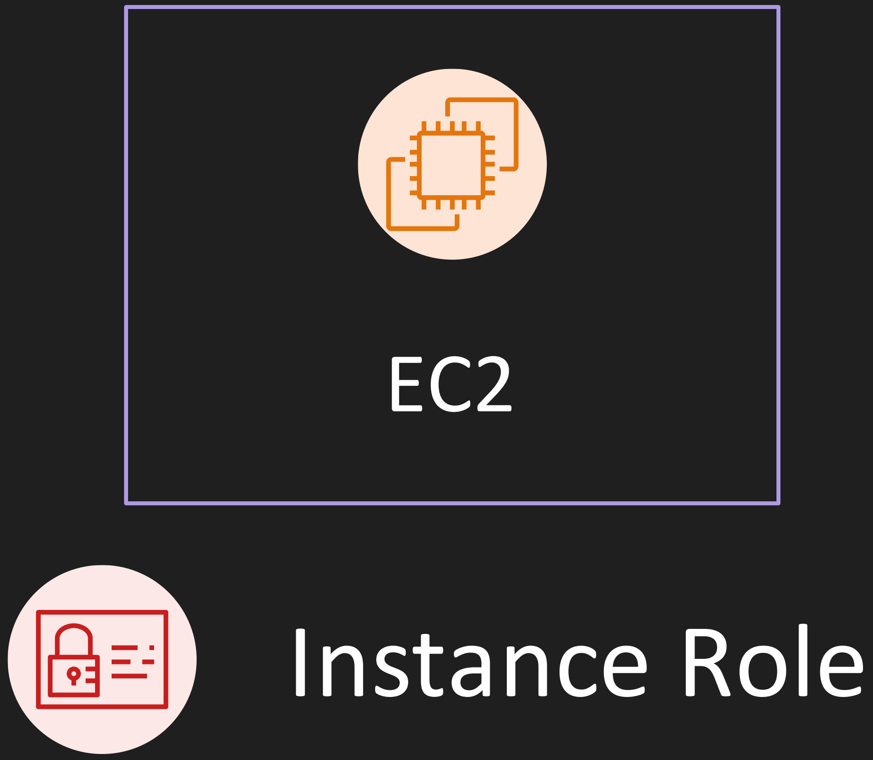


```
  "AccessKeyId": "ASIA23OOU55RQ3GXMBDM",
  "SecretAccessKey": "xJ1H9WYLi9jJ8sv9T29ChNk0AST0nnx2zE1/6w9G",
  "Token": "IQoJb3JpZ2luX2VjEDQaCXVzLWVhc3QtMiJHMEUCIH5eSx76nrw3eo0P7
UGVsdxanPi6jdpo5+lGDeQTImHl4O6U36PS5m6vD6LcN1X6Iv2Zpr6kIMuNzb0oV/kJQ+
Us9Slvo5KAEry/couSMCgAZ9NuQbeB3qpf/OAY550x/sH6zB9KPrlU9i7zrIXY1B4bvE4
JrS60Qo8H1IPix9G7mk6ZRkrdhMkvqlnXWetc/yTumtkRxjgNsbLJMc5D4fyc5n2CPtQi
saA+TbNWib8R8tWuZvGSTcP5azrhOBcfjpdoXInZIJCkv7c25Mz6F5Mn3SbQSq4H7r+s5
VXKQA/jKBsxAC8UY4SNrEnupinTCLh5PEBjqxAYBMbEI+BwGbiwx6fqpexQh16XuL2DkU
hOi0wT2YP9HsEDjR1q8n7nuu6377ABajajBf6nKT3ikYSPHwOy7IAxE/pY/3vSBQ==",
  "Expiration": "2025-07-26T18:12:52Z"
```

What Is Amazon ECS?

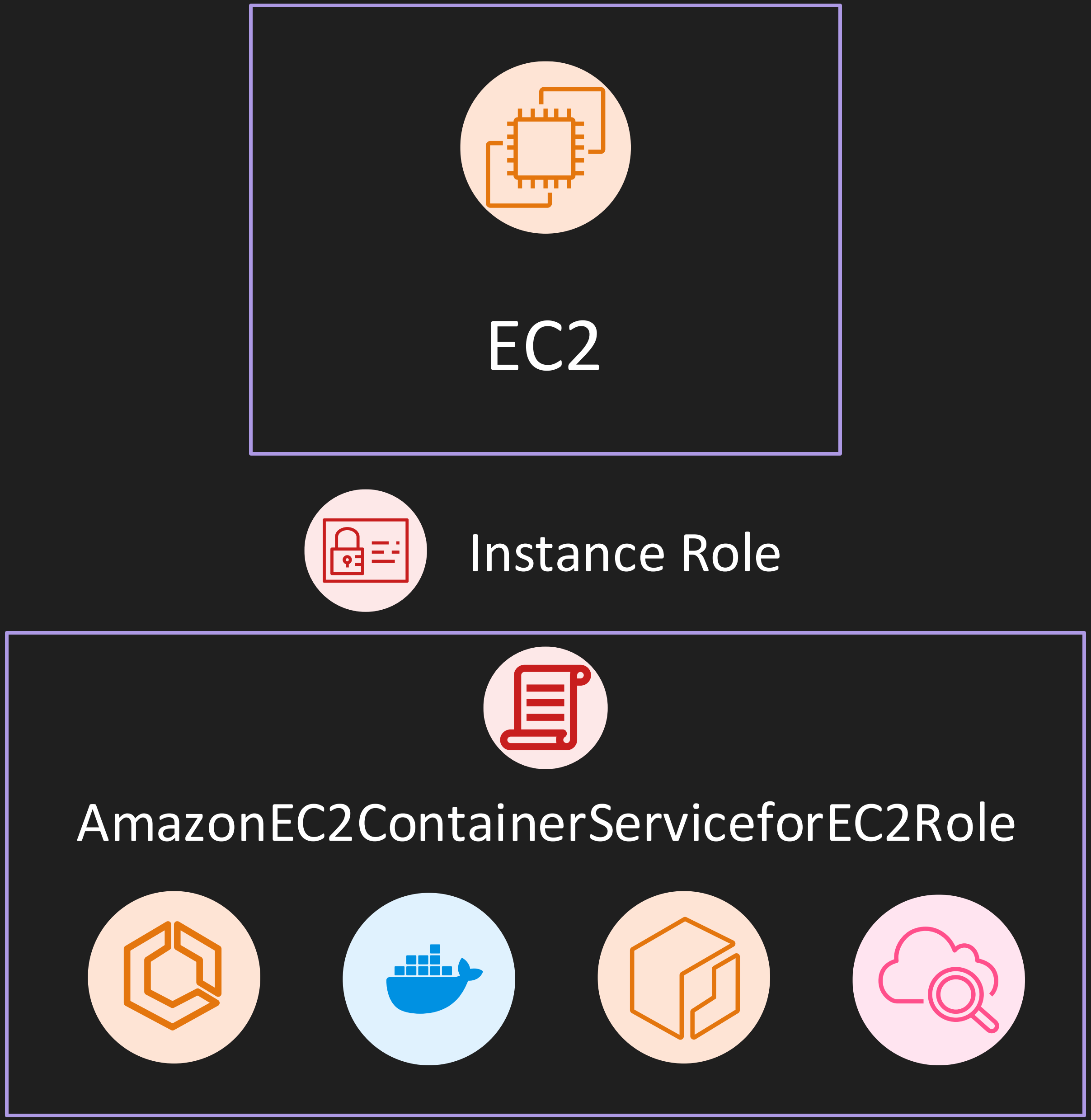
ECS ≈ **K8S**


ECS Cluster
EC2
EC2
EC2



EC2

Instance Role

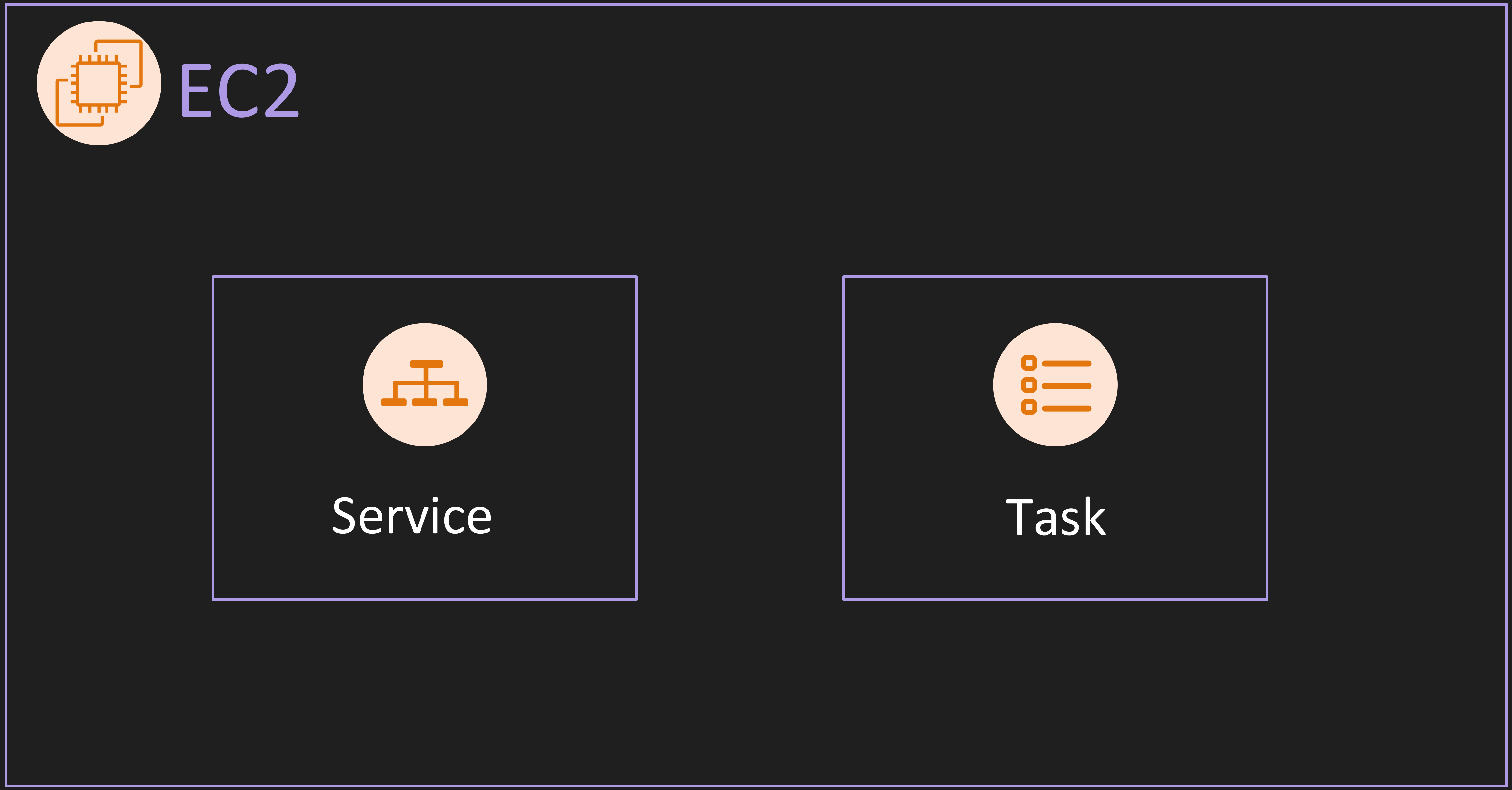


EC2

Instance Role

AmazonEC2ContainerServiceforEC2Role

EC2

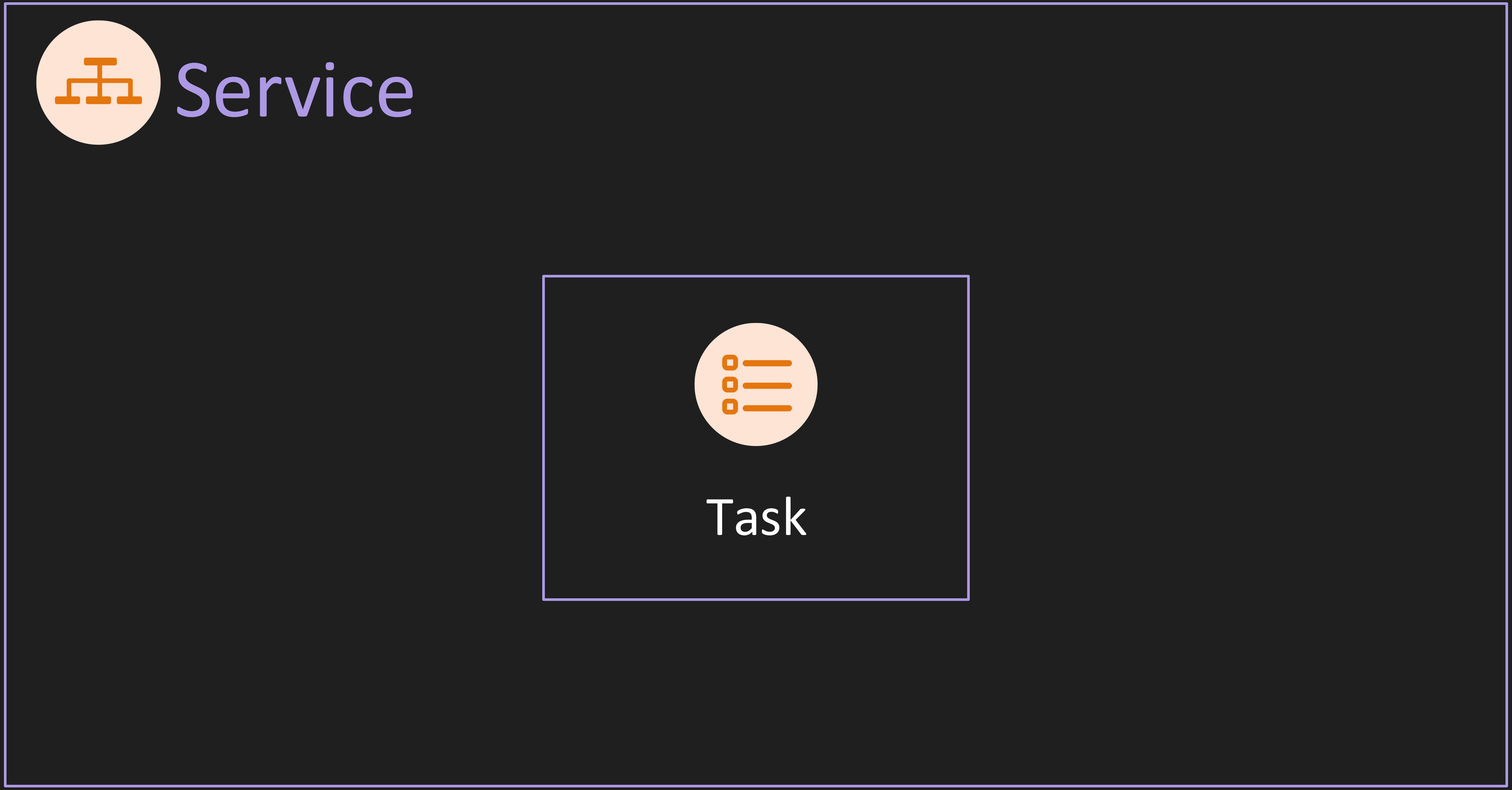


EC2

Service

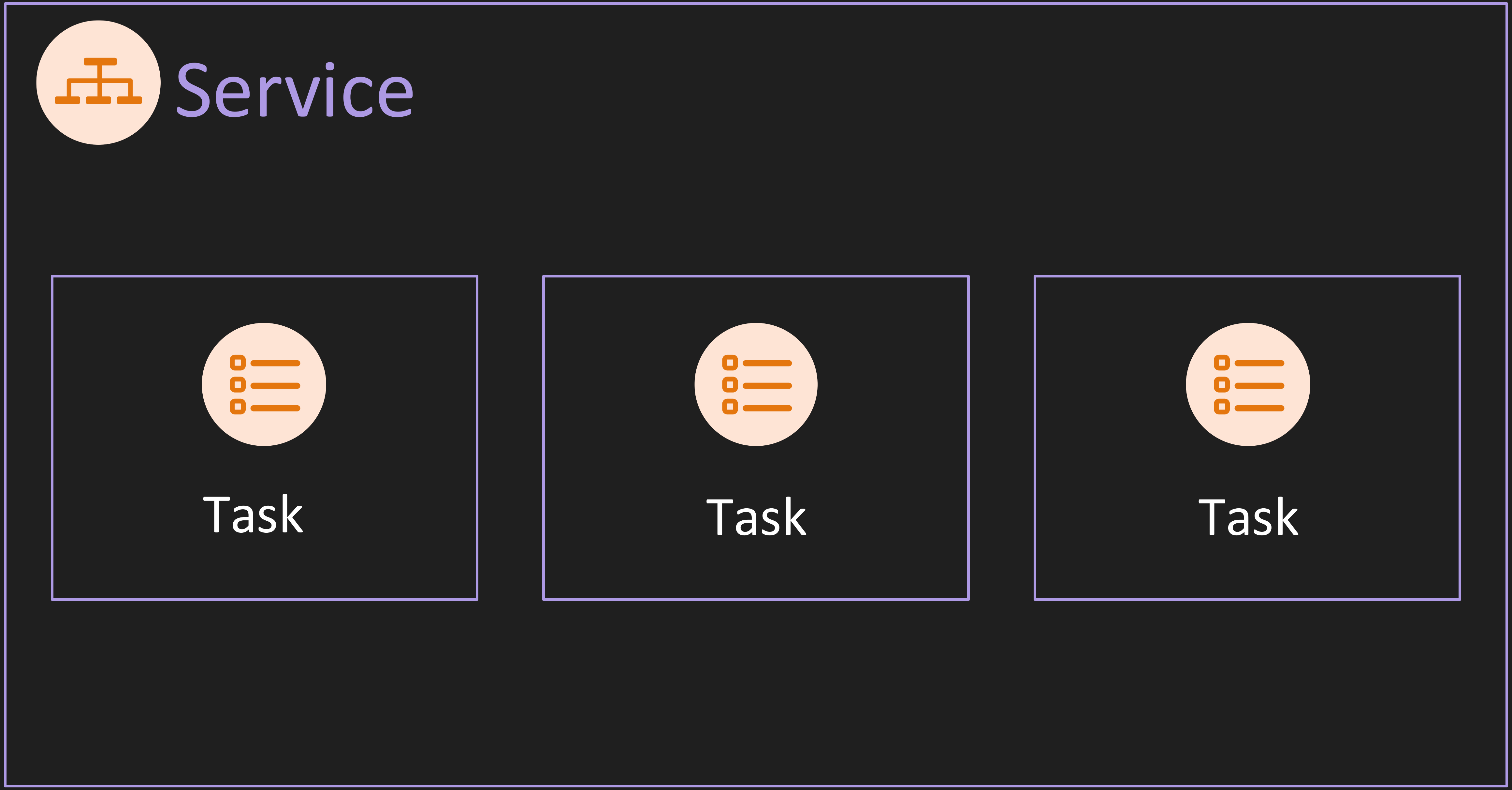
Task

Service

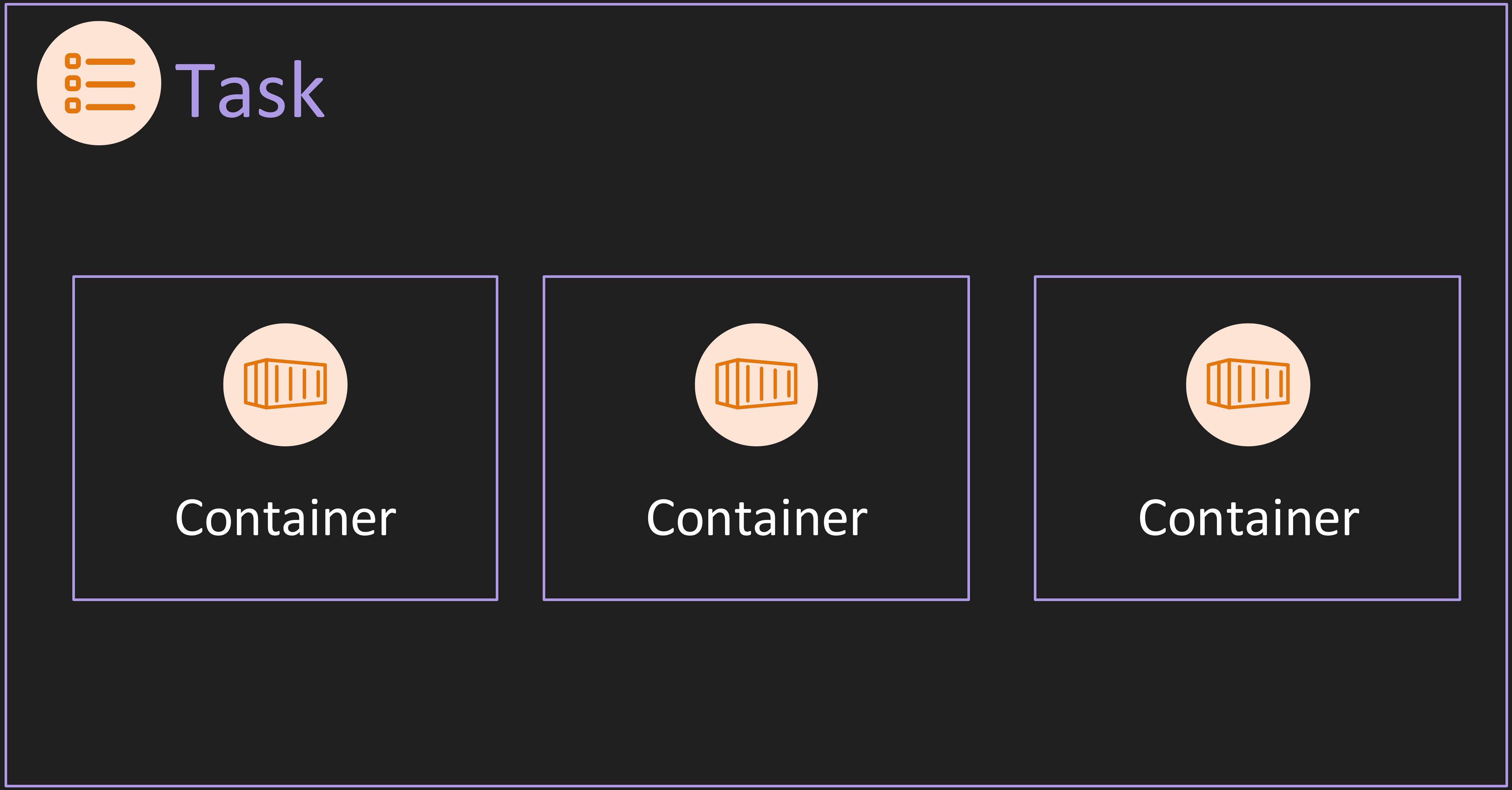


Service

Task


Service
Task
Task
Task

Task

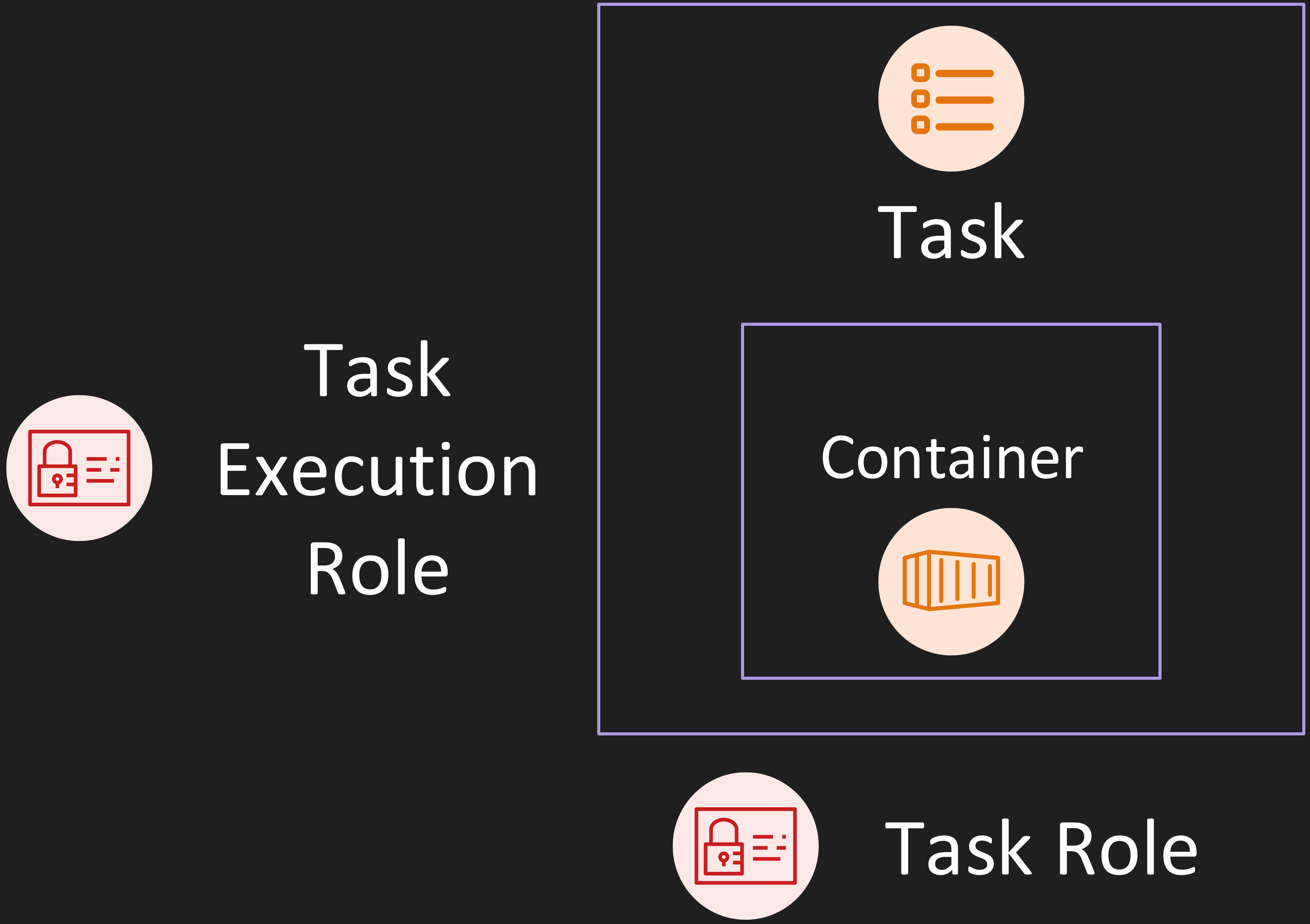


Task

Container

Container

Container



Task

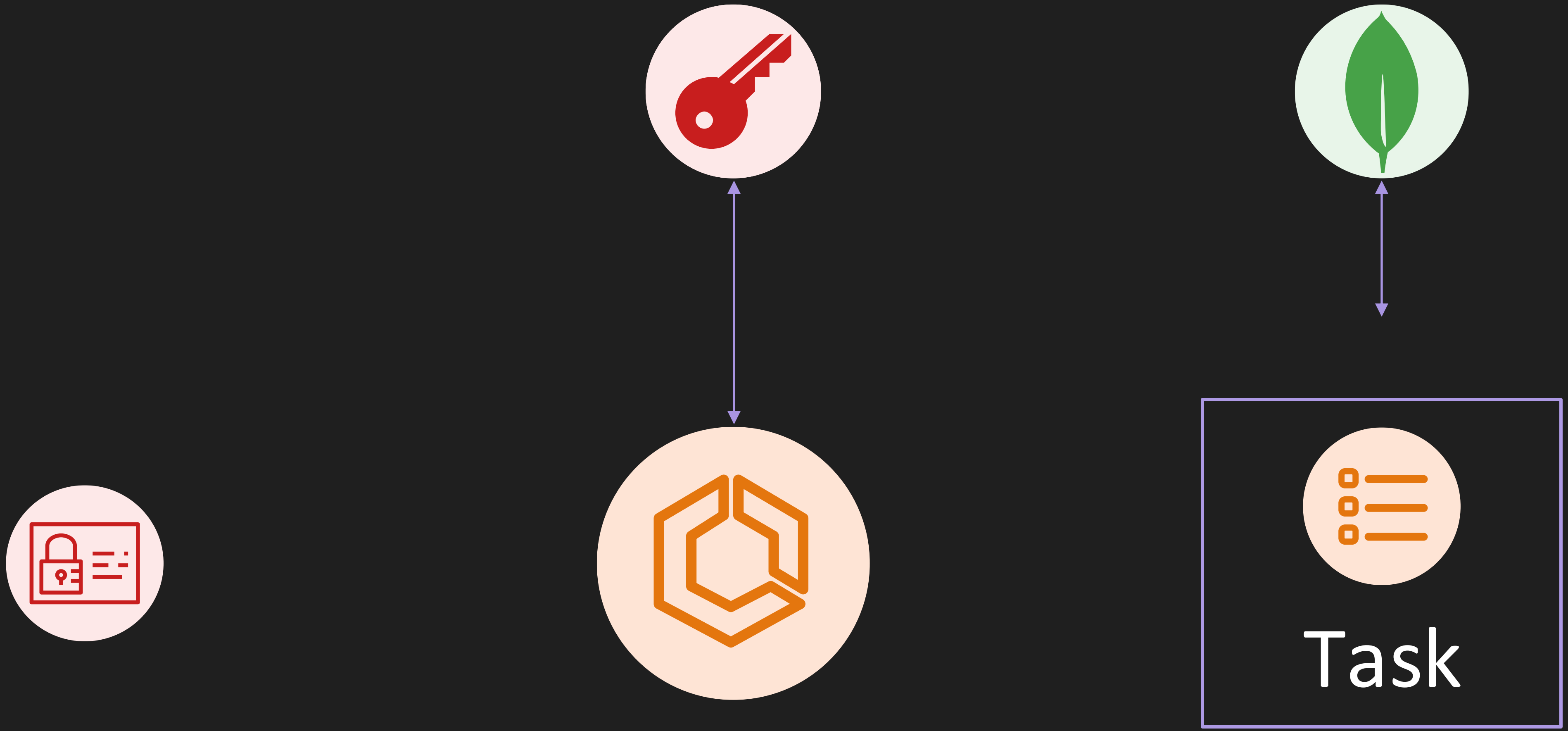
Container

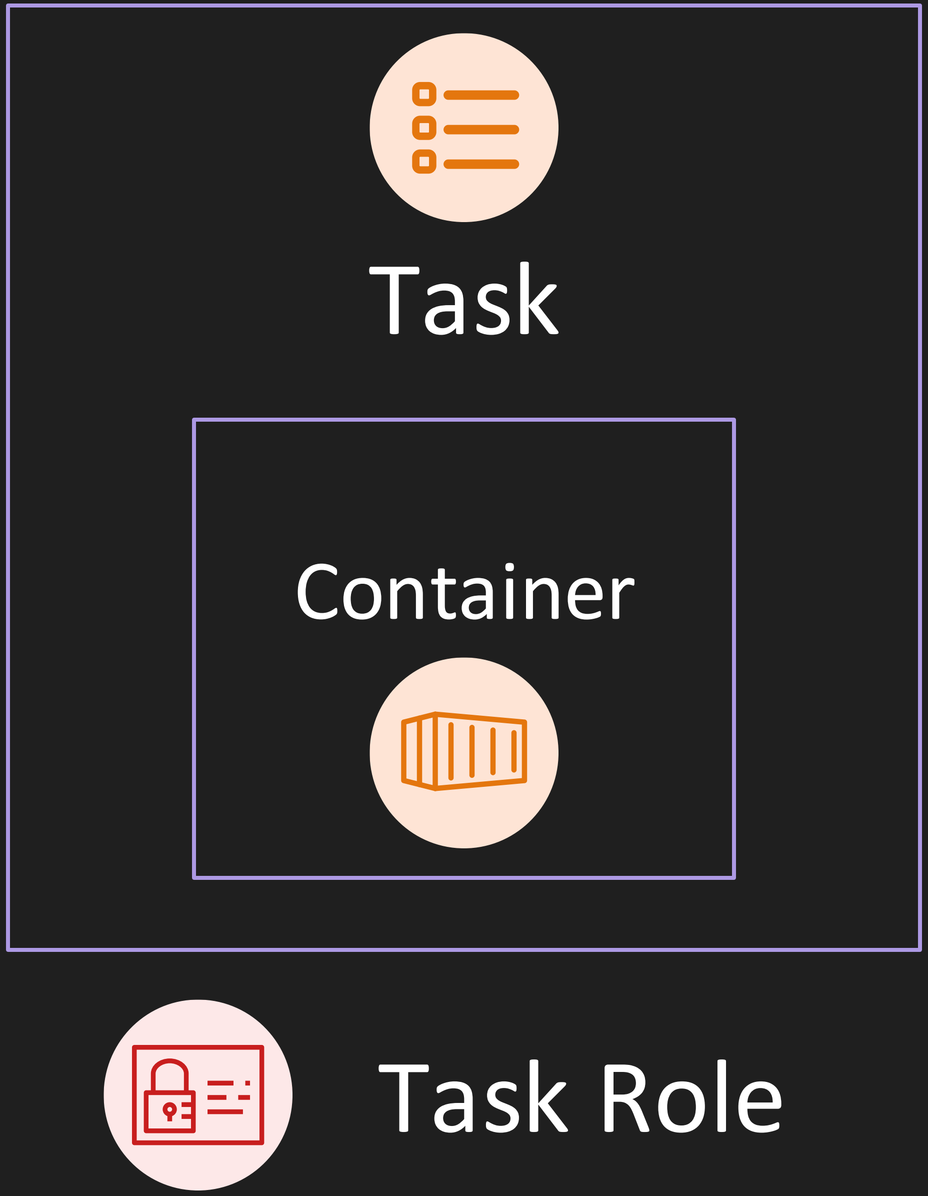
Task Execution Role

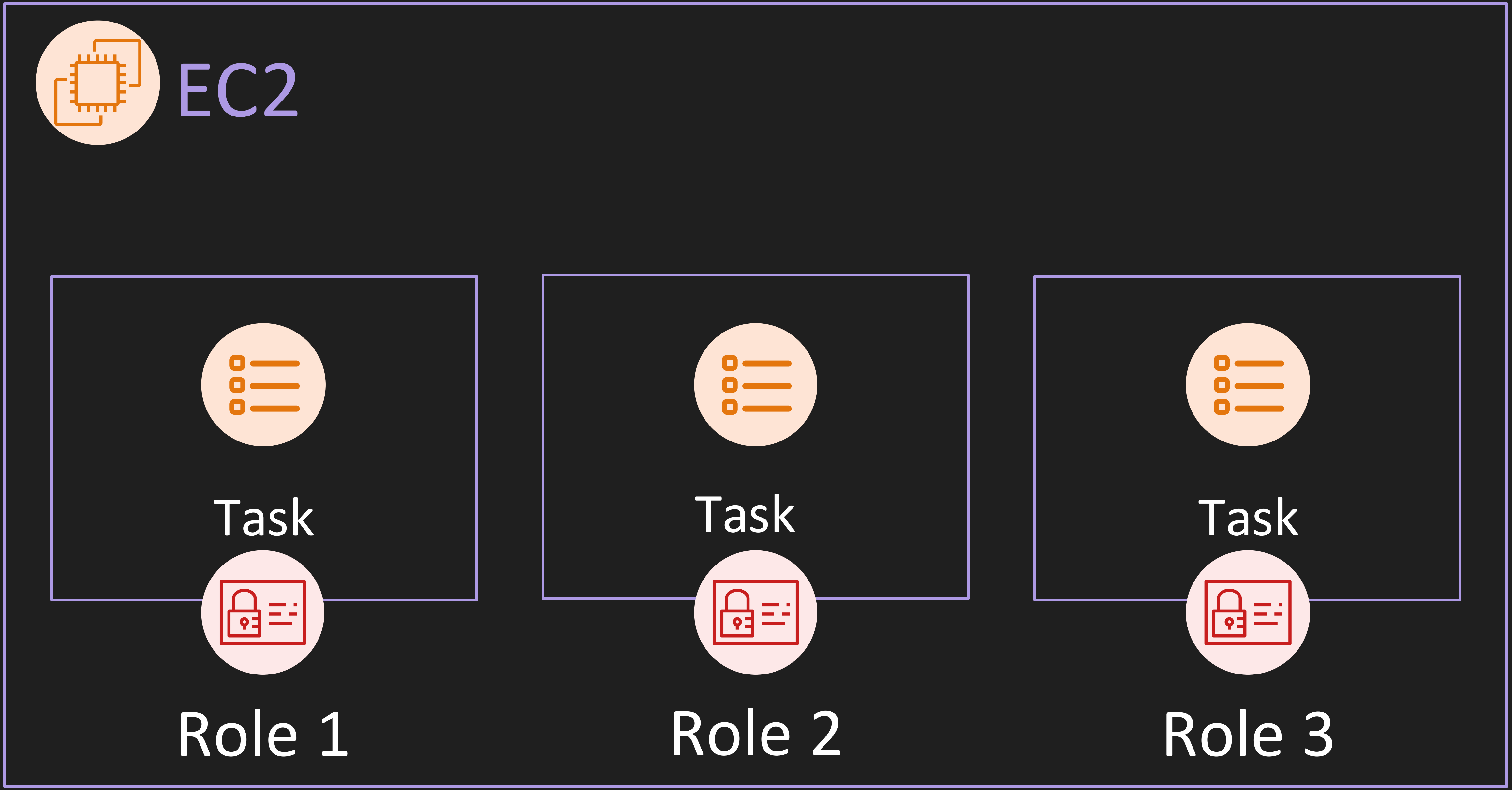
Task Role

Task Execution Role

Task


Task


Task
Container
Task Role


EC2
Task
Task
Task
Role 1
Role 2
Role 3

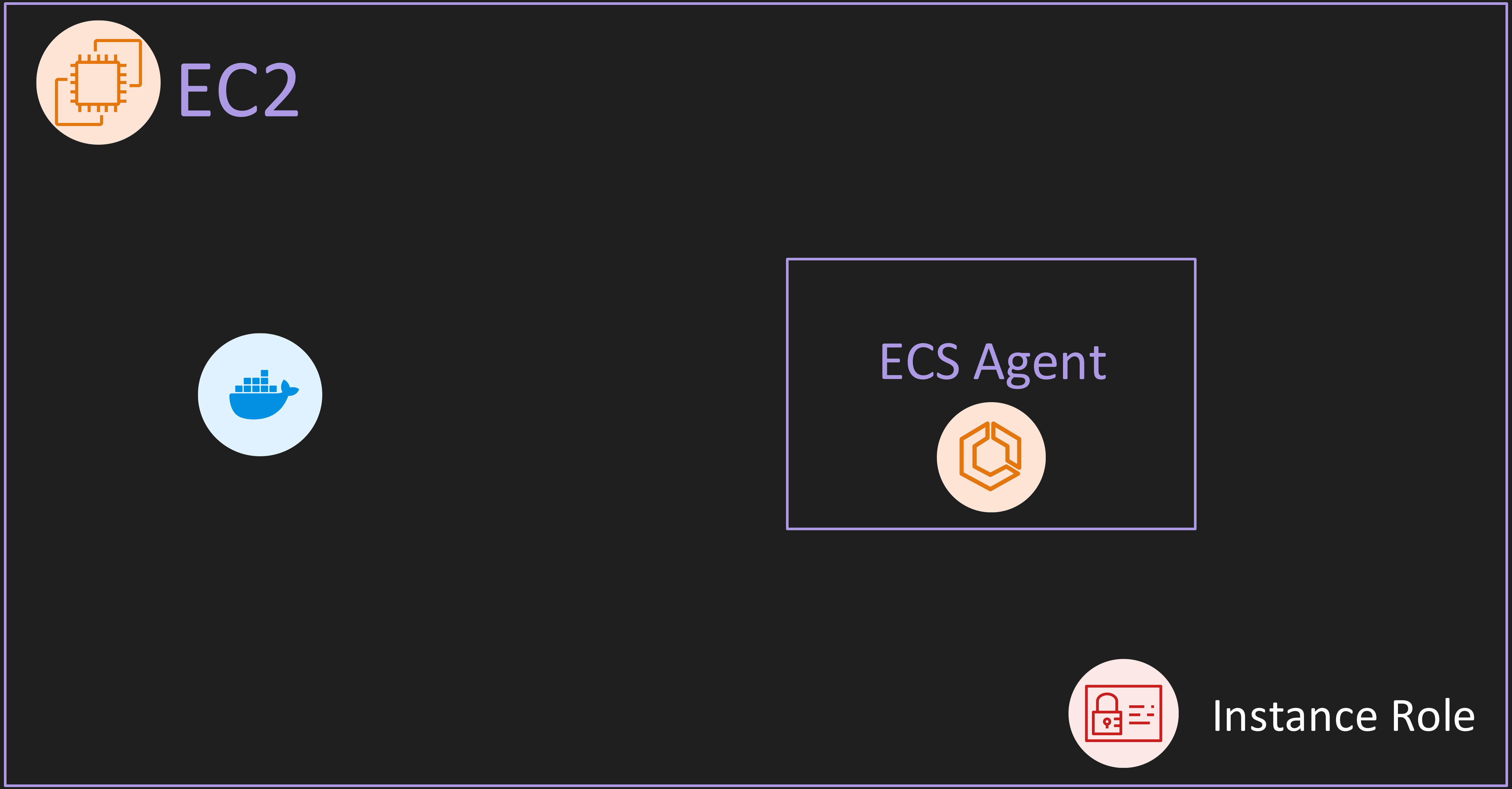
ECS Launch Modes

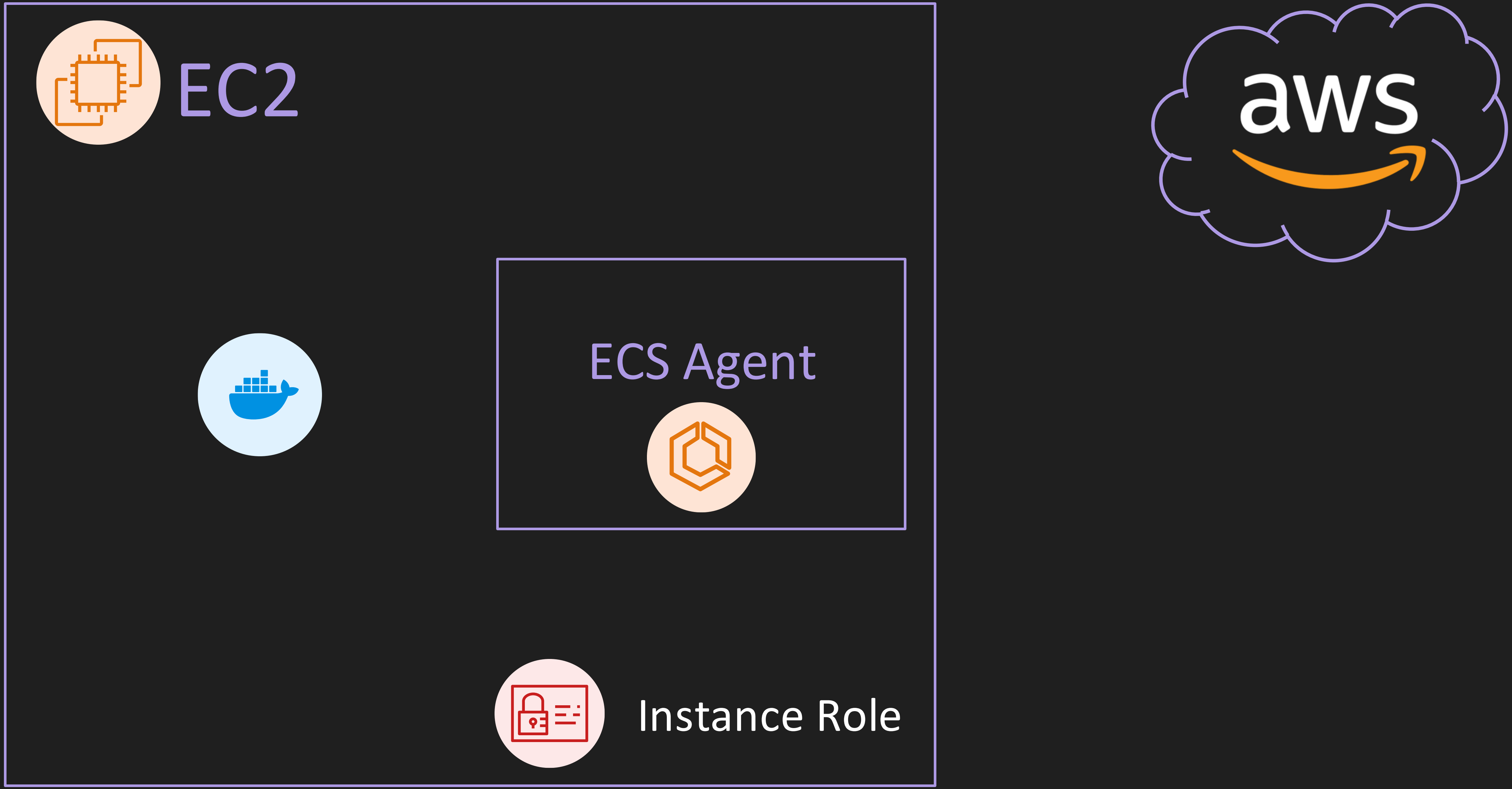
Fargate

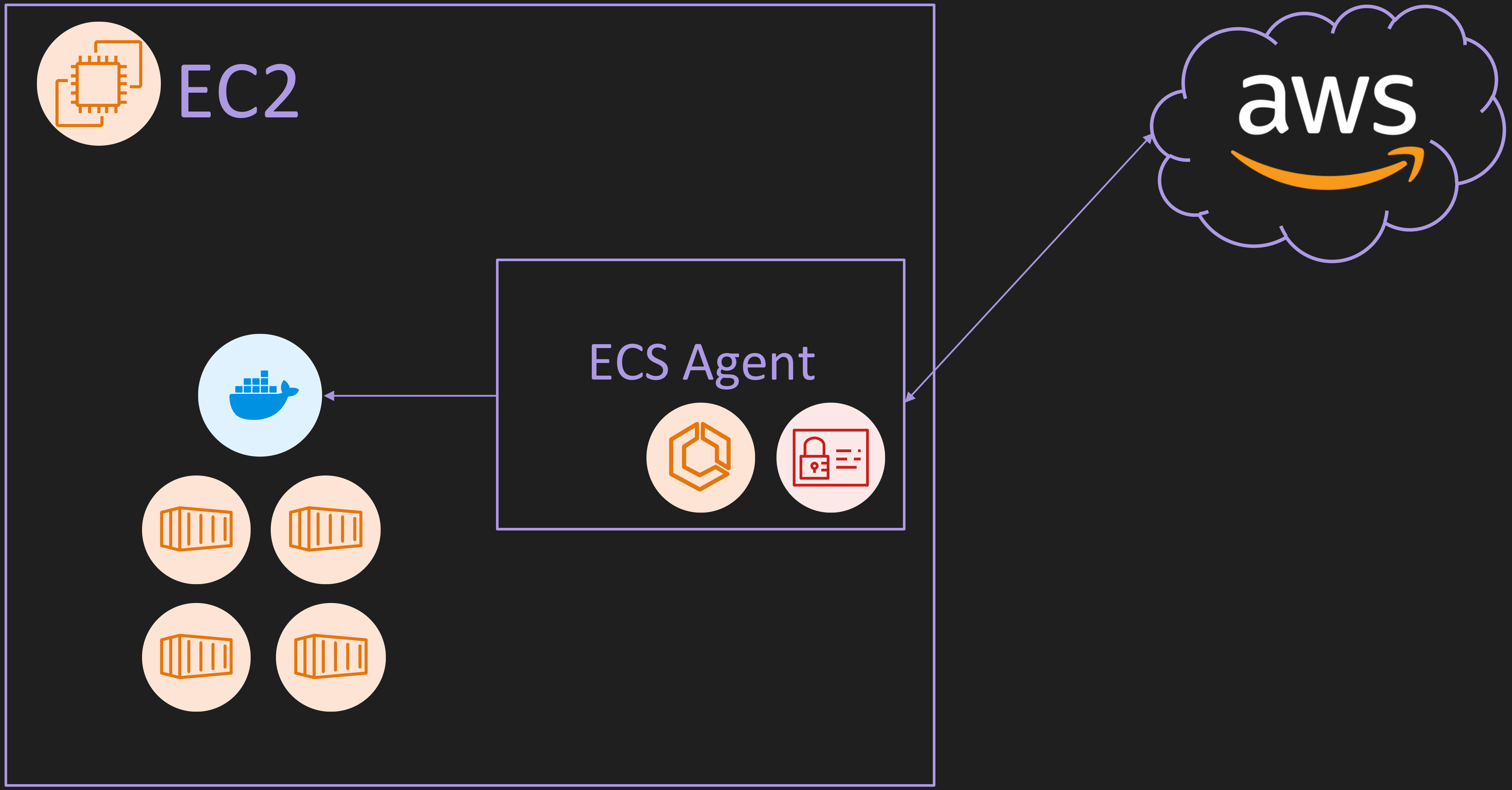
EC2

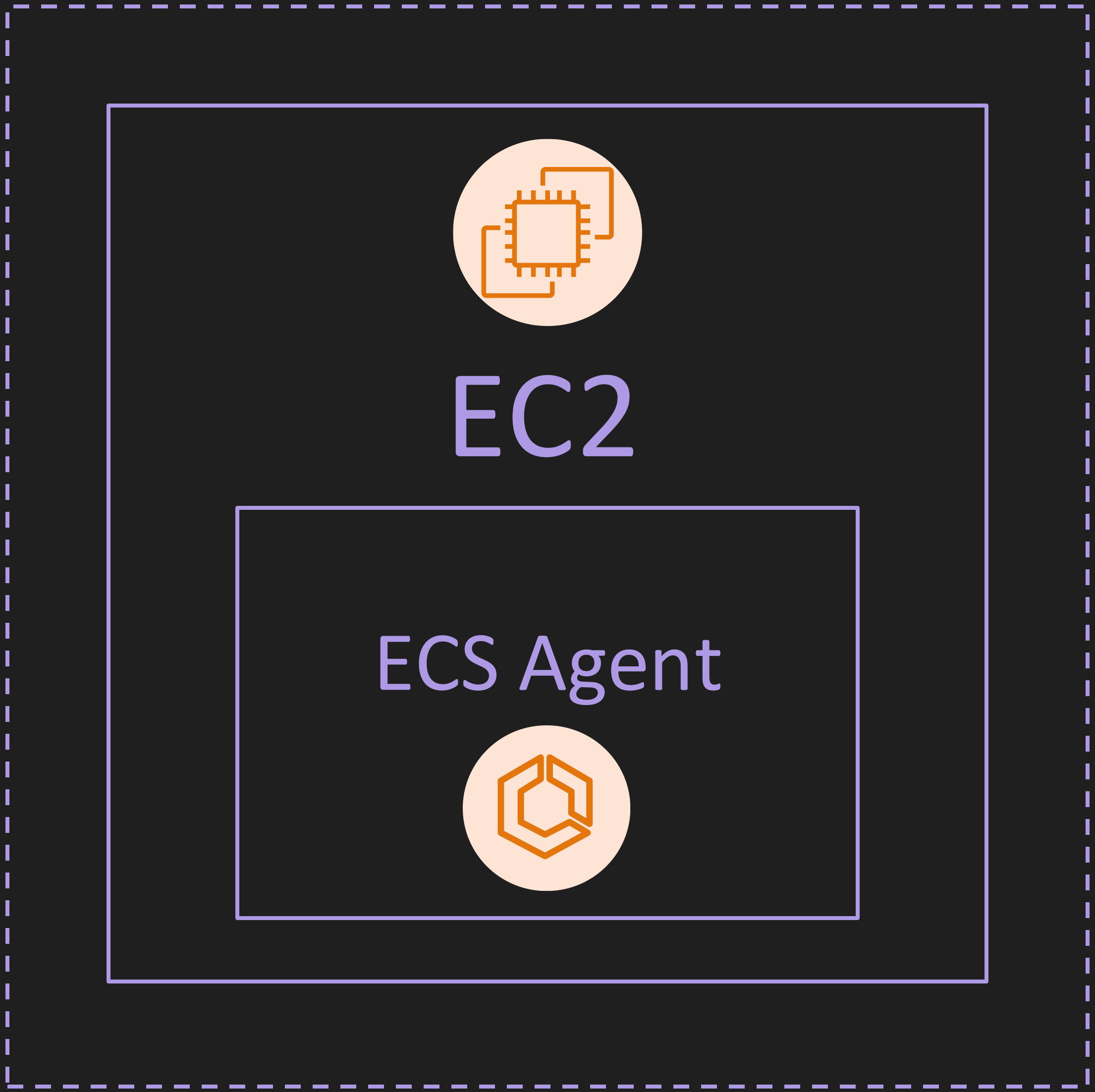
ECS Launch Modes

EC2

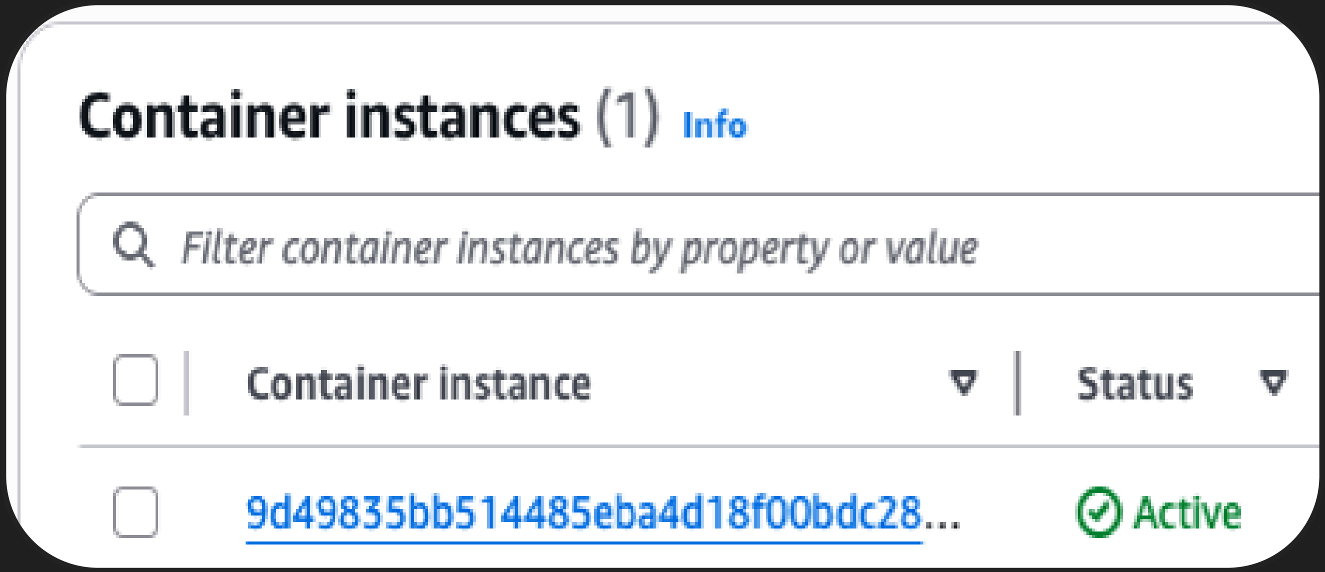

EC2
ECS Agent
Instance Role


EC2
ECS Agent
Instance Role
aws


EC2
ECS Agent
aws



Container Instance



Container instances (1) Info

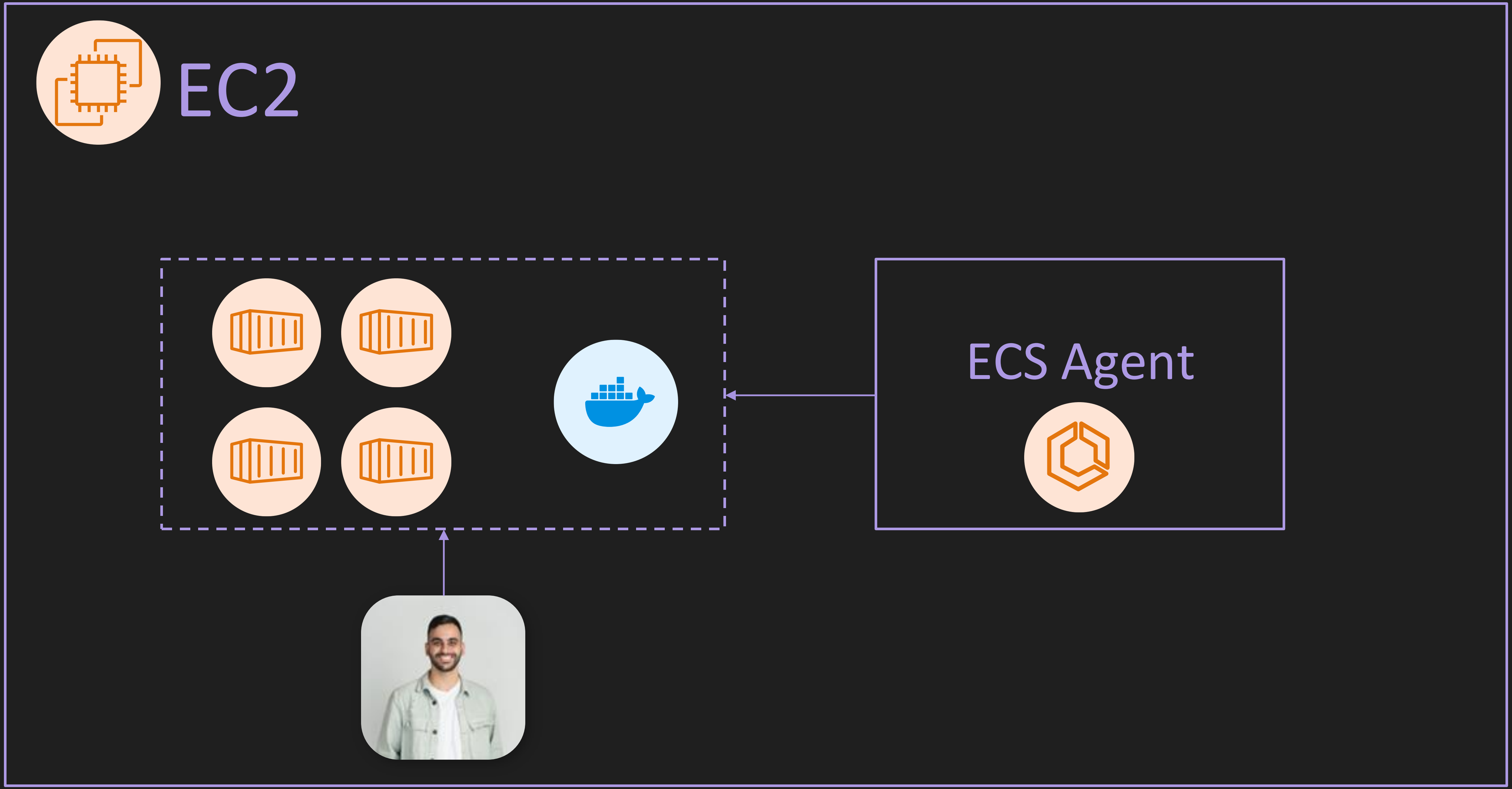
Filter container instances by property or value

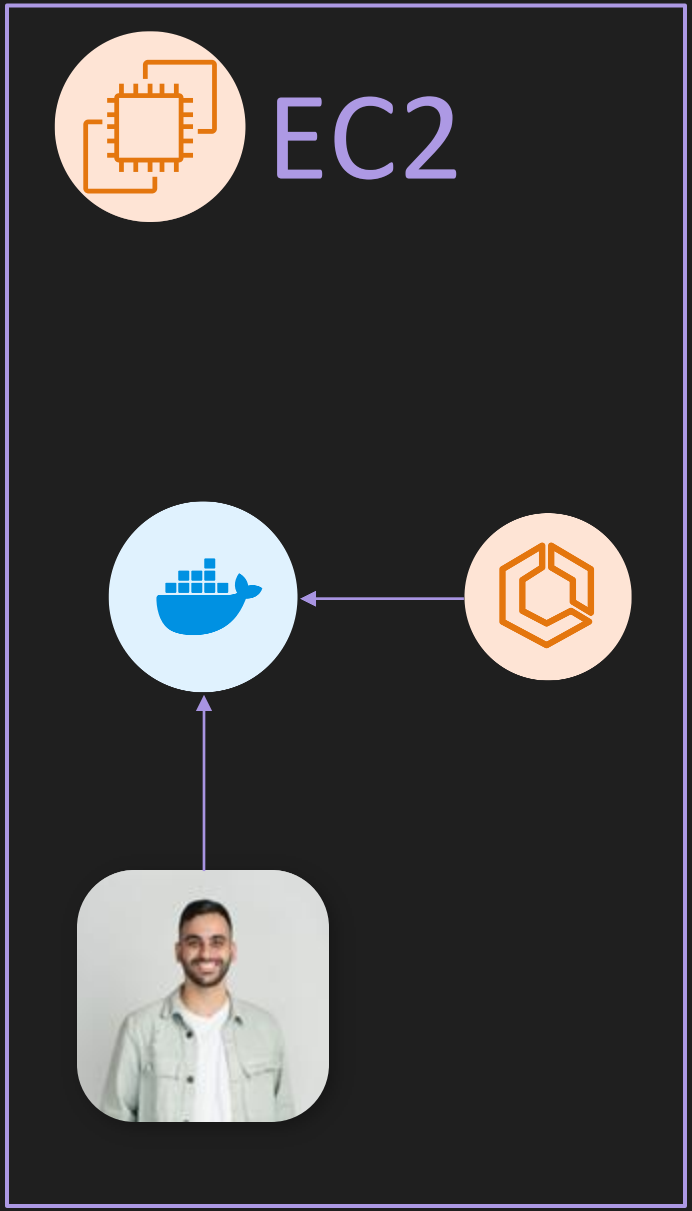
□	Container instance	Status
□	9d49835bb514485eba4d18f00bdc28...	✓ Active

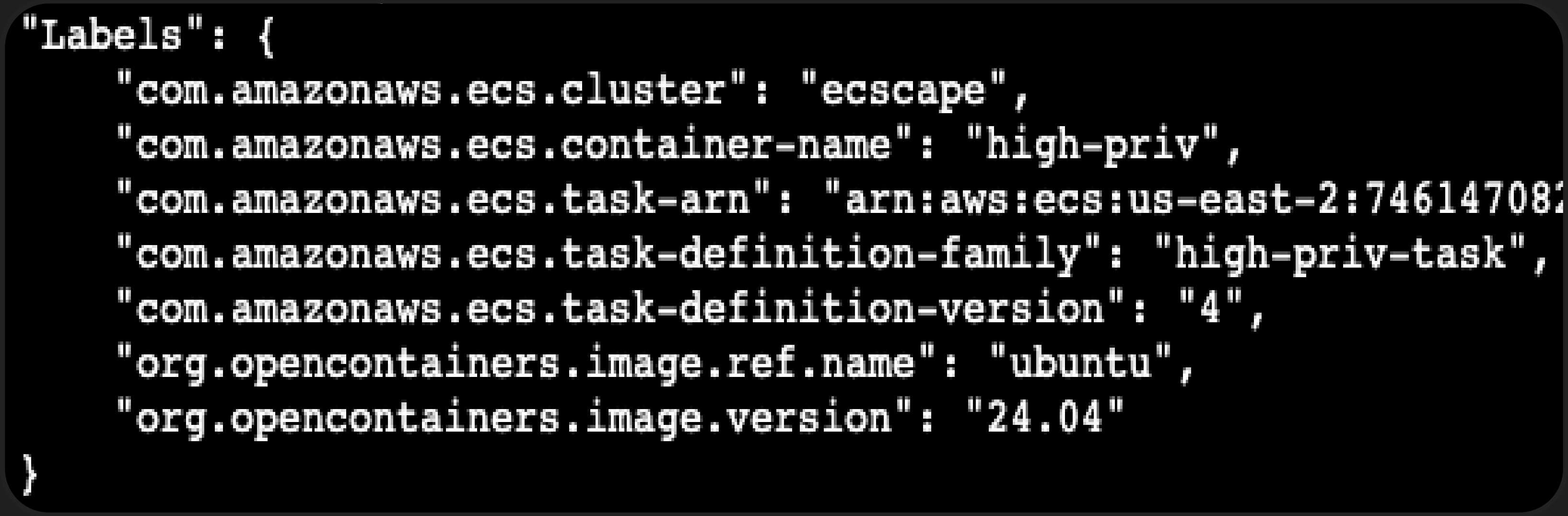
02
Story Time

Can you monitor ECS tasks?

Sure, let's try


EC2
ECS Agent

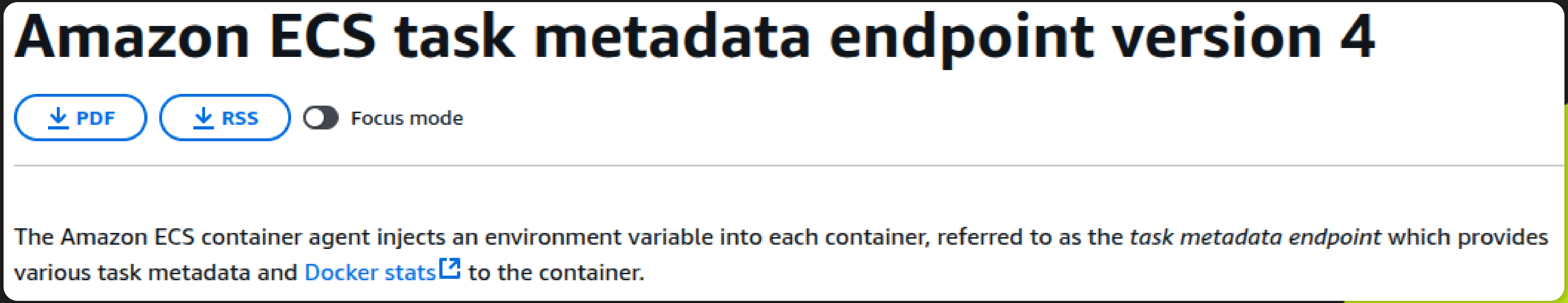




```
"Labels": {
    "com.amazonaws.ecs.cluster": "ecscape",
    "com.amazonaws.ecs.container-name": "high-priv",
    "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-east-2:74614708
    "com.amazonaws.ecs.task-definition-family": "high-priv-task",
    "com.amazonaws.ecs.task-definition-version": "4",
    "org.opencontainers.image.ref.name": "ubuntu",
    "org.opencontainers.image.version": "24.04"
}
```

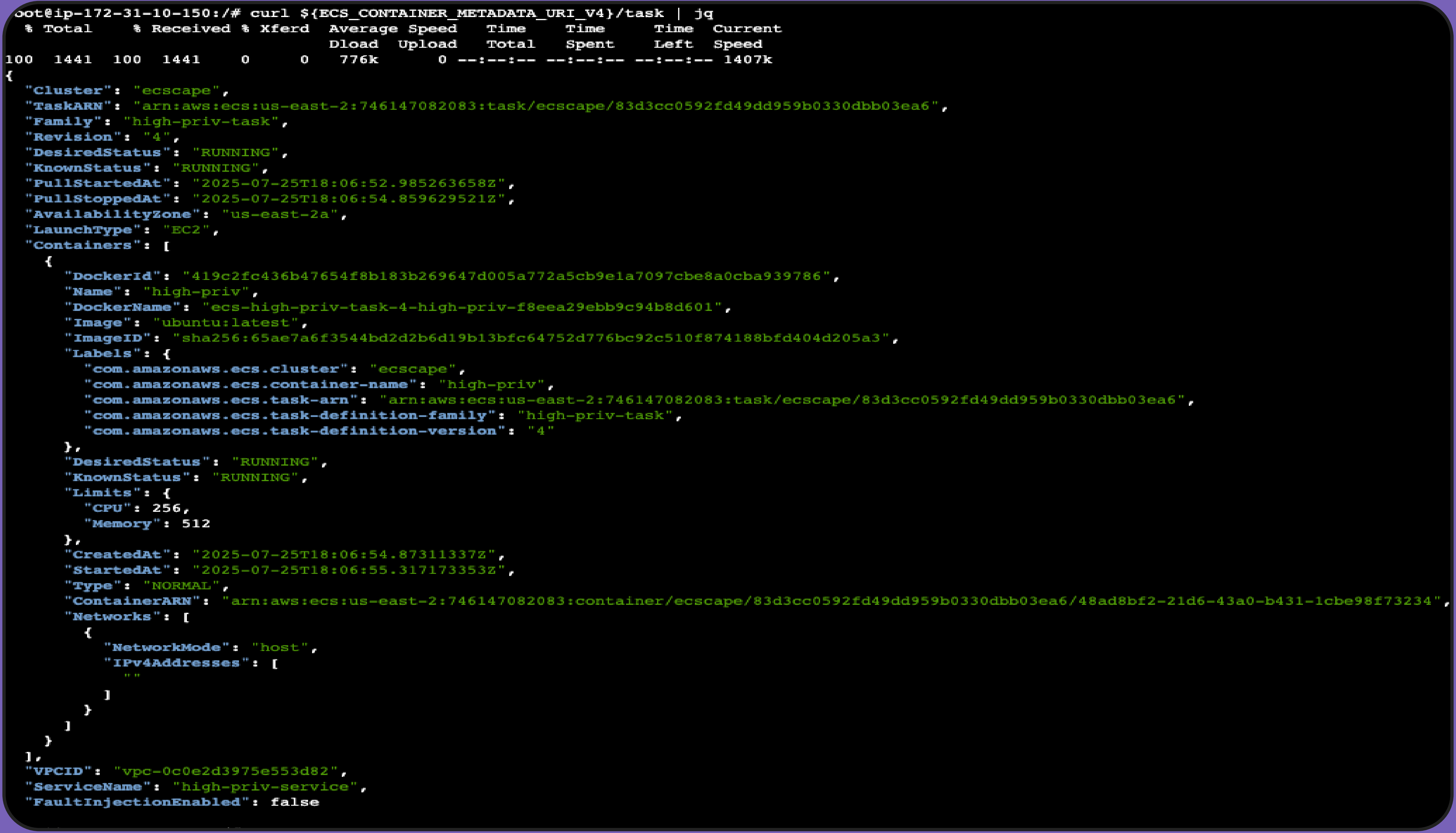
Exce
The
Nan

Amazon ECS task metadata endpoint

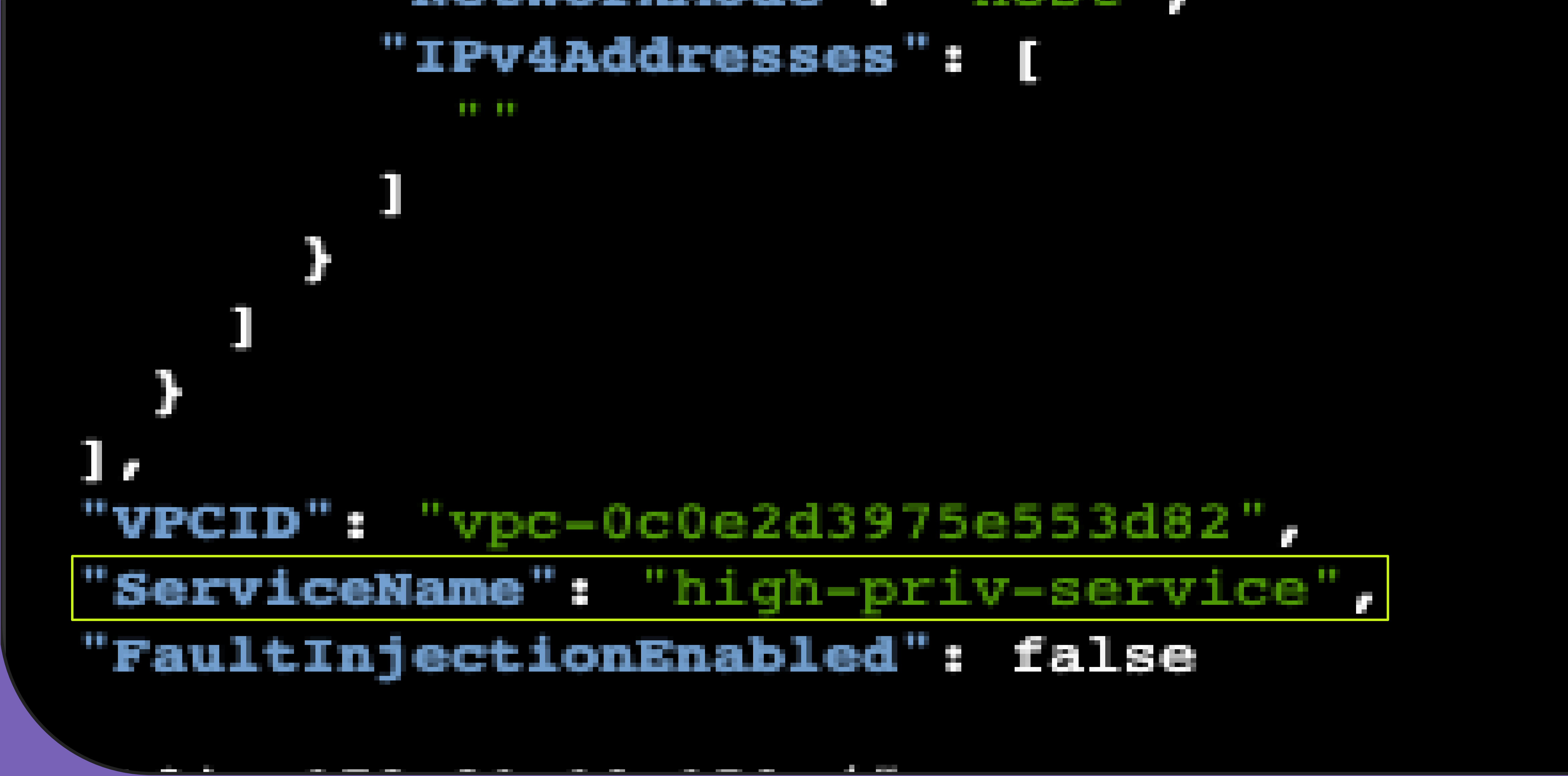


Amazon ECS task metadata endpoint version 4

The Amazon ECS container agent injects an environment variable into each container, referred to as the *task metadata endpoint* which provides various task metadata and Docker stats to the container.

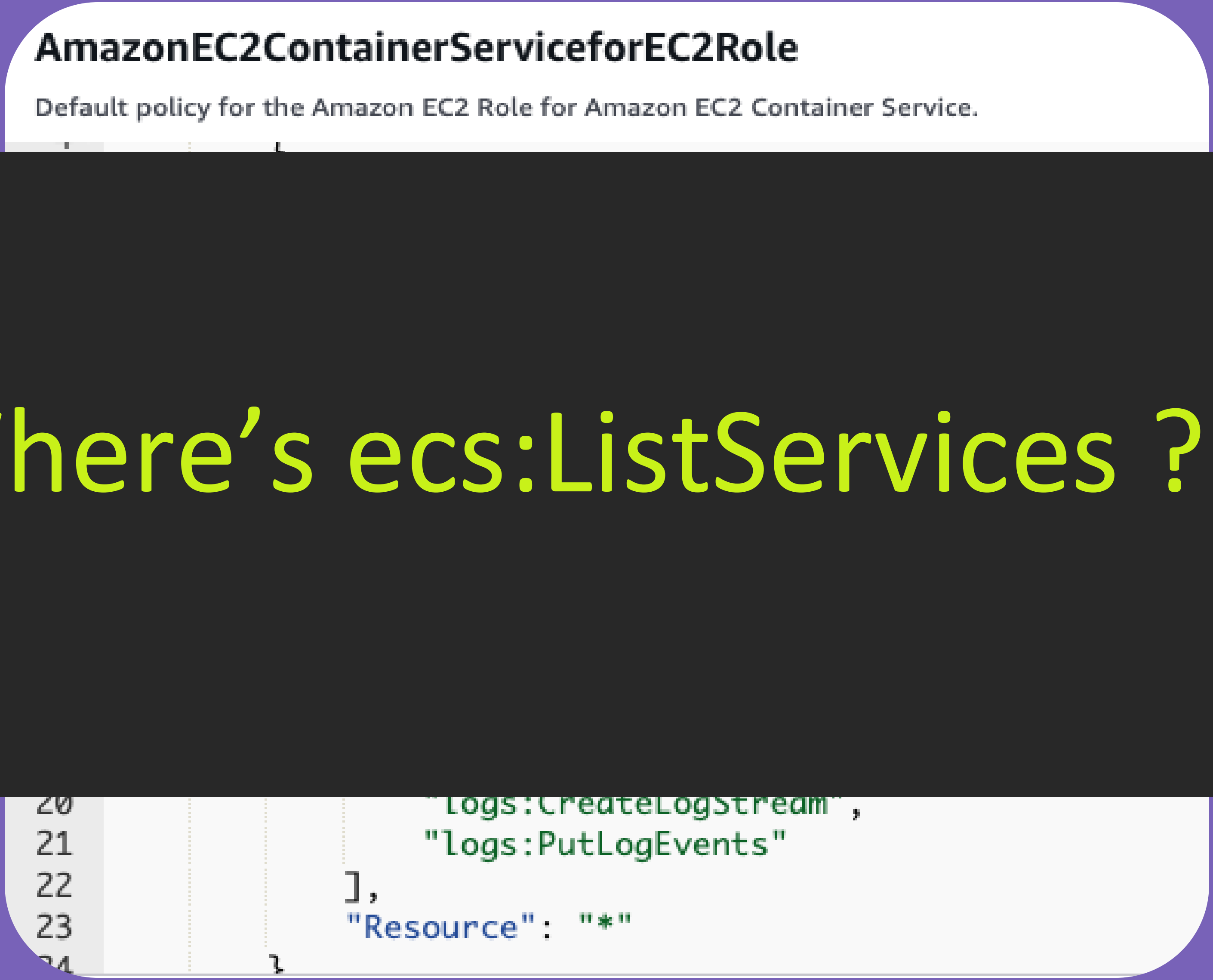


```
oot@ip-172-31-10-150:/# curl ${ECS_CONTAINER_METADATA_URI_V4}/task | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1441  100  1441    0     0   776k      0 --:--:-- --:--:-- --:--:-- 1407k
{
  "Cluster": "ecscape",
  "TaskARN": "arn:aws:ecs:us-east-2:746147082083:task/ecscape/83d3cc0592fd49dd959b0330dbb03ea6",
  "Family": "high-priv-task",
  "Revision": "4",
  "DesiredStatus": "RUNNING",
  "KnownStatus": "RUNNING",
  "PullStartedAt": "2025-07-25T18:06:52.985263658Z",
  "PullStoppedAt": "2025-07-25T18:06:54.859629521Z",
  "AvailabilityZone": "us-east-2a",
  "LaunchType": "EC2",
  "Containers": [
    {
      "DockerId": "419c2fc436b47654f8b183b269647d005a772a5cb9e1a7097cbe8a0cba939786",
      "Name": "high-priv",
      "DockerName": "ecs-high-priv-task-4-high-priv-f8eea29ebb9c94b8d601",
      "Image": "ubuntu:latest",
      "ImageID": "sha256:65ae7a6f3544bd2d2b6d19b13bfc64752d776bc92c510f874188bfd404d205a3",
      "Labels": {
        "com.amazonaws.ecs.cluster": "ecscape",
        "com.amazonaws.ecs.container-name": "high-priv",
        "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-east-2:746147082083:task/ecscape/83d3cc0592fd49dd959b0330dbb03ea6",
        "com.amazonaws.ecs.task-definition-family": "high-priv-task",
        "com.amazonaws.ecs.task-definition-version": "4"
      },
      "DesiredStatus": "RUNNING",
      "KnownStatus": "RUNNING",
      "Limits": {
        "CPU": 256,
        "Memory": 512
      },
      "CreatedAt": "2025-07-25T18:06:54.87311337Z",
      "StartedAt": "2025-07-25T18:06:55.317173353Z",
      "Type": "NORMAL",
      "ContainerARN": "arn:aws:ecs:us-east-2:746147082083:container/ecscape/83d3cc0592fd49dd959b0330dbb03ea6/48ad8bf2-21d6-43a0-b431-1cbe98f73234",
      "Networks": [
        {
          "NetworkMode": "host",
          "IPv4Addresses": [
            ""
          ]
        }
      ]
    }
  ],
  "VPCID": "vpc-0c0e2d3975e553d82",
  "ServiceName": "high-priv-service",
  "FaultInjectionEnabled": false
```



```
        "IPv4Addresses": [
          ""
        ]
      }
    ]
  }
],
"VPCID": "vpc-0c0e2d3975e553d82",
"ServiceName": "high-priv-service",
"FaultInjectionEnabled": false
```

It all started with a service name



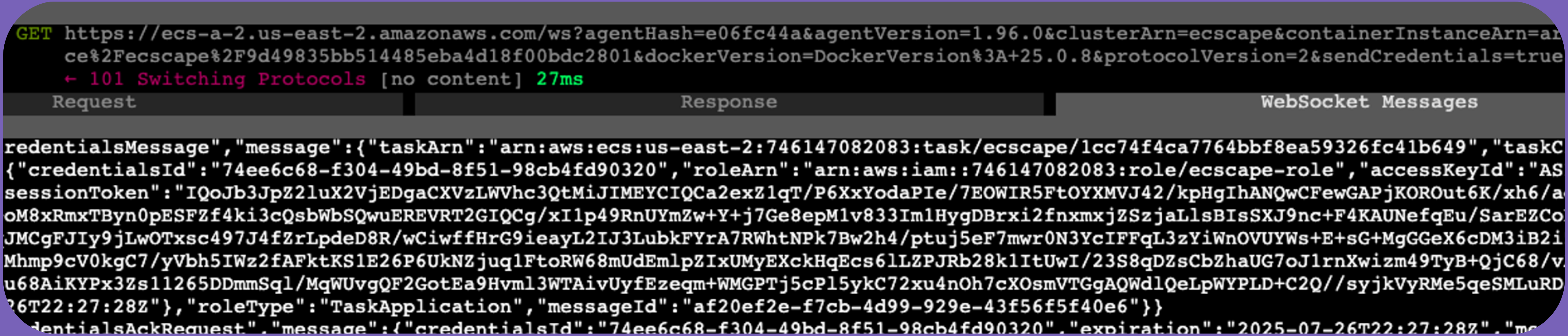
AmazonEC2ContainerServiceforEC2Role

Default policy for the Amazon EC2 Role for Amazon EC2 Container Service.

```
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        }
```

Where’s ecs:ListServices ???

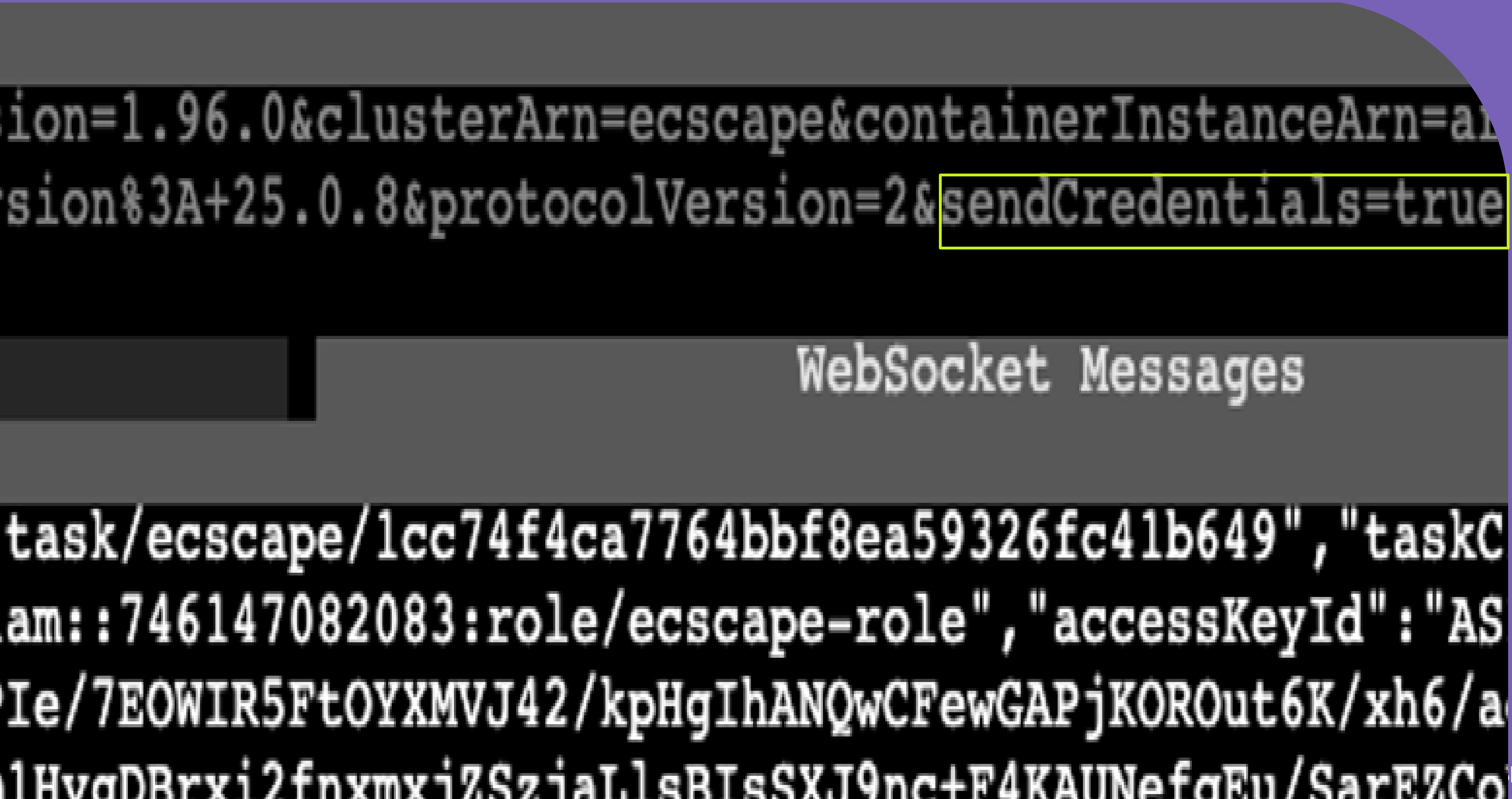
Proxy



```
GET https://ecs-a-2.us-east-2.amazonaws.com/ws?agentHash=e06fc44a&agentVersion=1.96.0&clusterArn=ecscape&containerInstanceArn=ar
    ce%2Fecscape%2F9d49835bb514485eba4d18f00bdc2801&dockerVersion=DockerVersion%3A+25.0.8&protocolVersion=2&sendCredentials=true
    ← 101 Switching Protocols [no content] 27ms
```

```
redentialsMessage","message":{"taskArn":"arn:aws:ecs:us-east-2:746147082083:task/ecscape/1cc74f4ca7764bbf8ea59326fc41b649","taskC
{"credentialsId":"74ee6c68-f304-49bd-8f51-98cb4fd90320","roleArn":"arn:aws:iam::746147082083:role/ecscape-role","accessKeyId":"AS
sessionToken":"IQoJb3JpZ2luX2VjEDgaCXVzLWVhc3QtMiJIMEYCIQCa2exZ1qT/P6XxYodaPIe/7EOWIR5FtOYXMVJ42/kpHgIhANQwCFewGAPjKOROut6K/xh6/a
oM8xRmxTByn0pESFZf4ki3cQsbWbSQwuEREVRT2GIQCg/xI1p49RnUYmZw+Y+j7Ge8epM1v833Im1HygDBrxi2fnxmxjZSzjaLlsBIsSXJ9nc+F4KAUNefqEu/SarEZCo
JMCgFJIy9jLwOTxsc497J4fZrLpdeD8R/wCiwffHrG9ieayL2IJ3LubkFYrA7RWhtNPk7Bw2h4/ptuj5eF7mwr0N3YcIFFqL3zYiWnOVUYWs+E+sG+MgGGeX6cDM3iB2i
Mhmp9cV0kgC7/yVbh5IWz2fAFktKS1E26P6UkNZjuq1FtoRW68mUdEmlpZIxUMyEXckHqEcs61LZPJRb28k1ItUwI/23S8qDZsCbZhaUG7oJ1rnXwizm49TyB+QjC68/v
u68AiKYPx3Zs11265DDmmSql/MqWUvgQF2GotEa9Hvml3WTAivUyfEzeqm+WMGPTj5cPl5ykC72xu4nOh7cXOsmVTGgAQWdlQeLpWYPLD+C2Q//syjkVyRMe5qeSMLuRD
26T22:27:28Z"},"roleType":"TaskApplication","messageId":"af20ef2e-f7cb-4d99-929e-43f56f5f40e6"}}
dentialsAckRequest","message":{"credentialsId":"74ee6c68-f304-49bd-8f51-98cb4fd90320","expiration":"2025-07-26T22:27:28Z","m
```

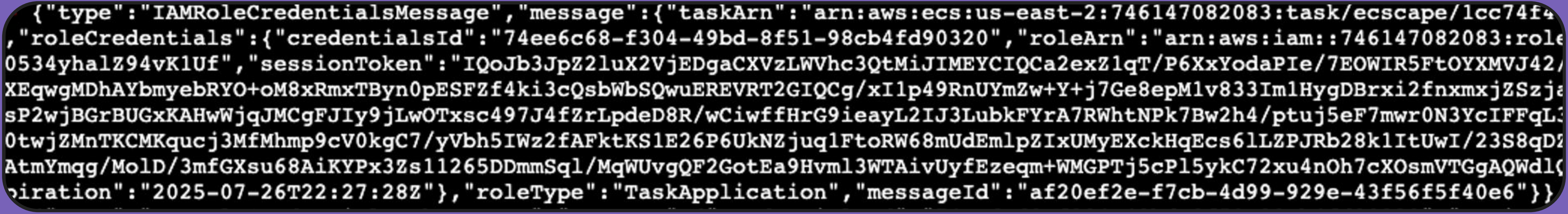
Proxy



```
ion=1.96.0&clusterArn=ecscape&containerInstanceArn=ai
rsion%3A+25.0.8&protocolVersion=2&sendCredentials=true
```

```
task/ecscape/1cc74f4ca7764bbf8ea59326fc41b649","taskC
am::746147082083:role/ecscape-role","accessKeyId":"AS
?Ie/7EOWIR5FtOYXMVJ42/kpHgIhANQwCFewGAPjKOROut6K/xh6/a
lHyqDBrxi2fnxmxjZSzjaLlsBIsSXJ9nc+F4KAUNefqEu/SarEZCo
```

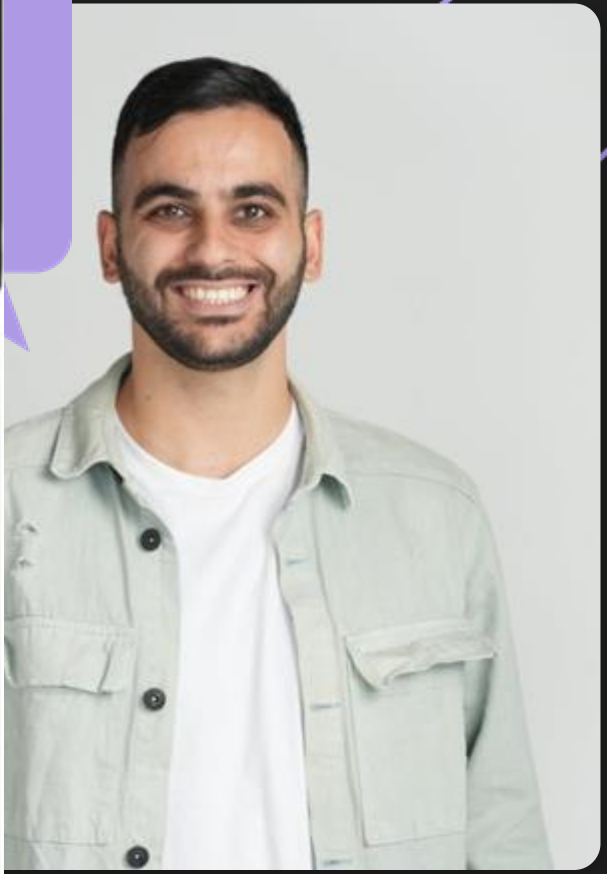
Proxy



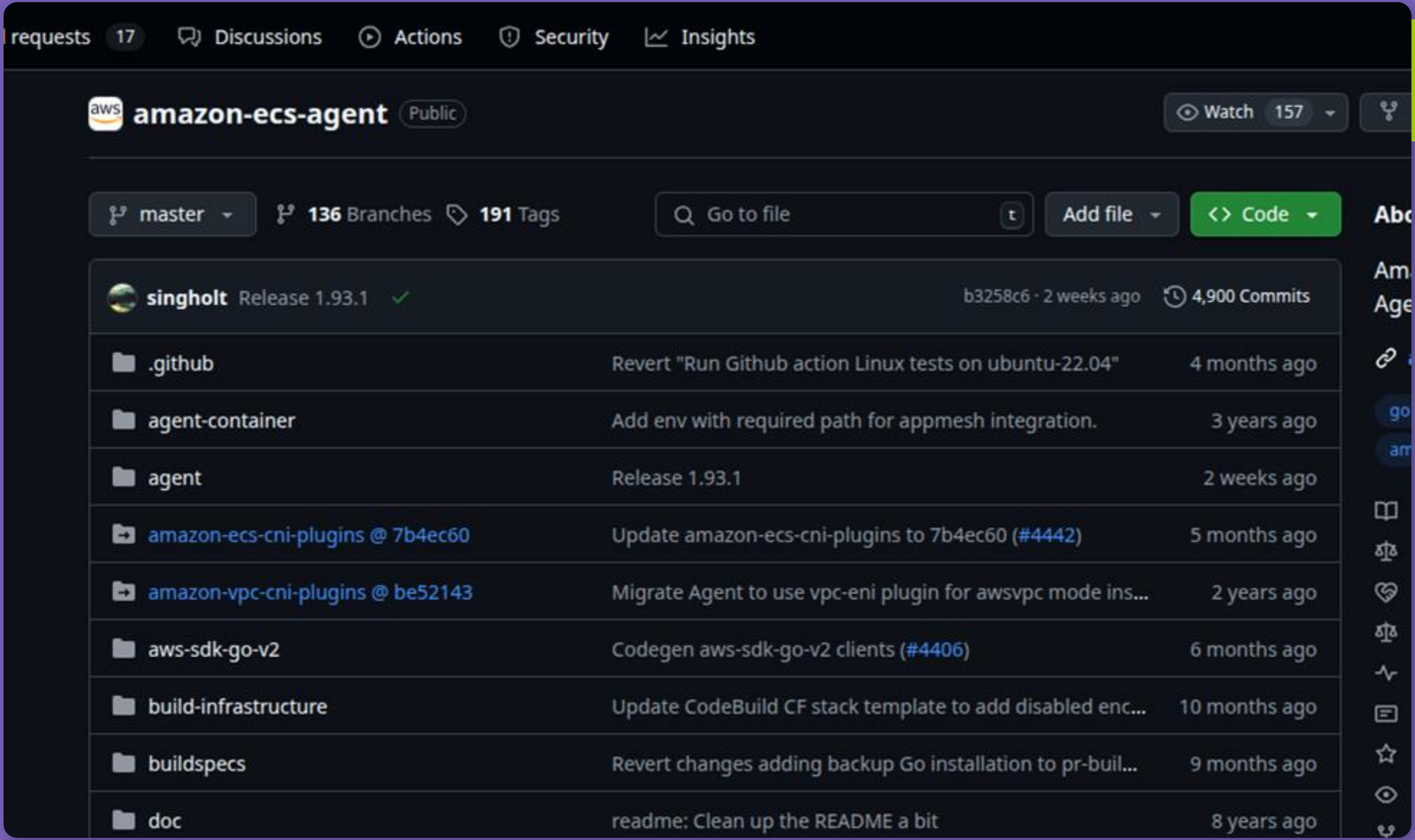
```
{"type":"IAMRoleCredentialsMessage","message":{"taskArn":"arn:aws:ecs:us-east-2:746147082083:task/ecscape/1cc74f4
,"roleCredentials":{"credentialsId":"74ee6c68-f304-49bd-8f51-98cb4fd90320","roleArn":"arn:aws:iam::746147082083:role
0534yhalZ94vK1Uf","sessionToken":"IQoJb3JpZ2luX2VjEDgaCXVzLWVhc3QtMiJIMEYCIQCa2exZ1qT/P6XxYodaPIe/7EOWIR5FtOYXMVJ42/
XEqwgMDhAYbmyebRYO+oM8xRmxTByn0pESFZf4ki3cQsbWbSQwuEREVRT2GIQCg/xI1p49RnUYmZw+Y+j7Ge8epM1v833Im1HygDBrxi2fnxmxjZSzja
sP2wjBGrBUGxKAHwWjqJMCgFJIy9jLwOTxsc497J4fZrLpdeD8R/wCiwffHrG9ieayL2IJ3LubkFYrA7RWhtNPk7Bw2h4/ptuj5eF7mwr0N3YcIFFqL3
0twjZMnTKCMKqucj3MfMhmp9cV0kgC7/yVbh5IWz2fAFktKS1E26P6UkNZjuq1FtoRW68mUdEmlpZIxUMyEXckHqEcs61LZPJRb28k1ItUwI/23S8qDZ
AtmYmqg/MolD/3mfGXsu68AiKYPx3Zs11265DDmmSql/MqWUvgQF2GotEa9Hvml3WTAivUyfEzeqm+WMGPTj5cPl5ykC72xu4nOh7cXOsmVTGgAQWdl
piration":"2025-07-26T22:27:28Z"},"roleType":"TaskApplication","messageId":"af20ef2e-f7cb-4d99-929e-43f56f5f40e6"}}
```

Can I impersonate the ECS agent?

Exciting research
Respect the roadmap
ME



03
ECScape



amazon-ecs-agent Public

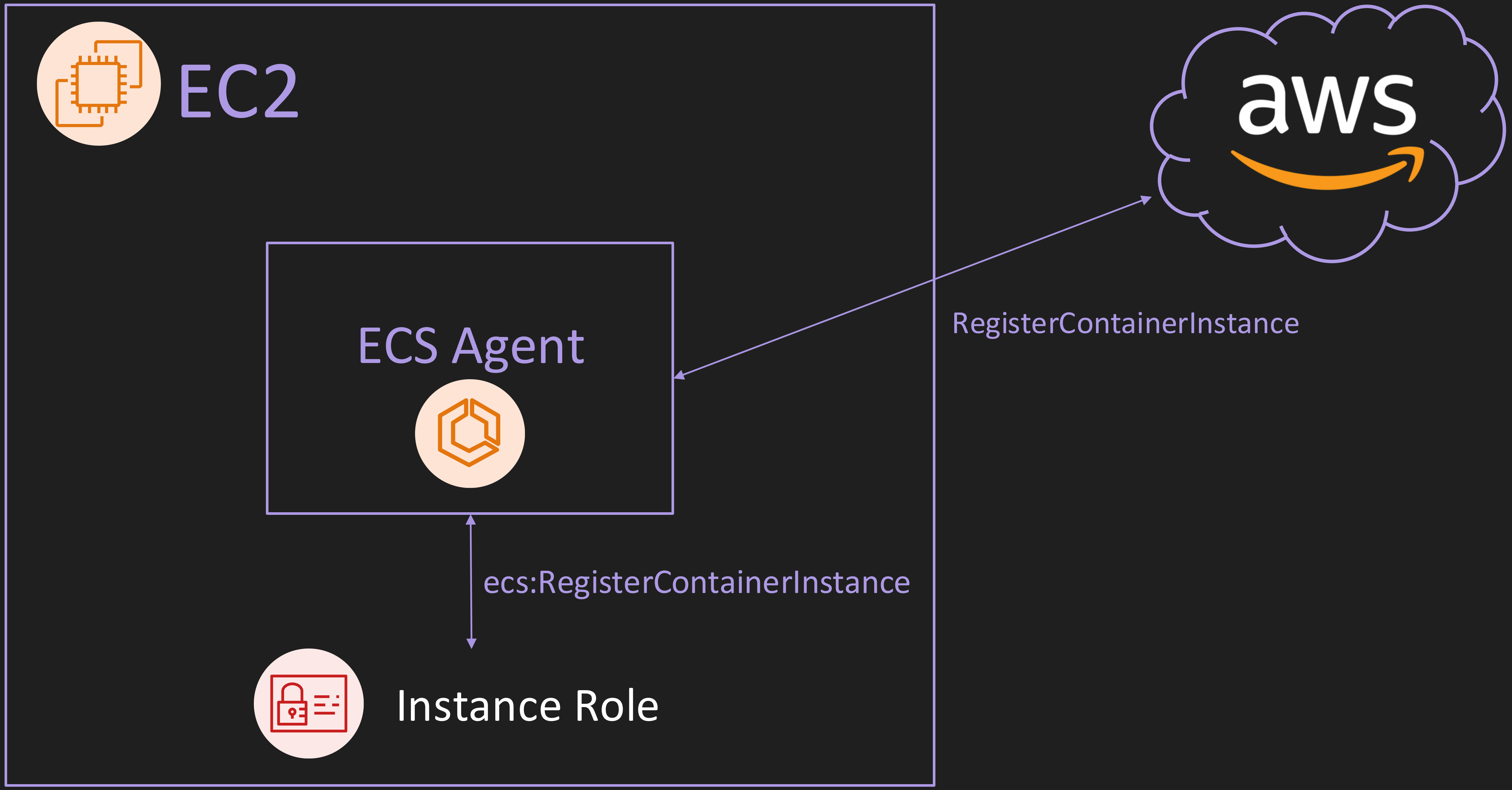
singholt Release 1.93.1 ✓ b3258c6 · 2 weeks ago 4,900 Commits

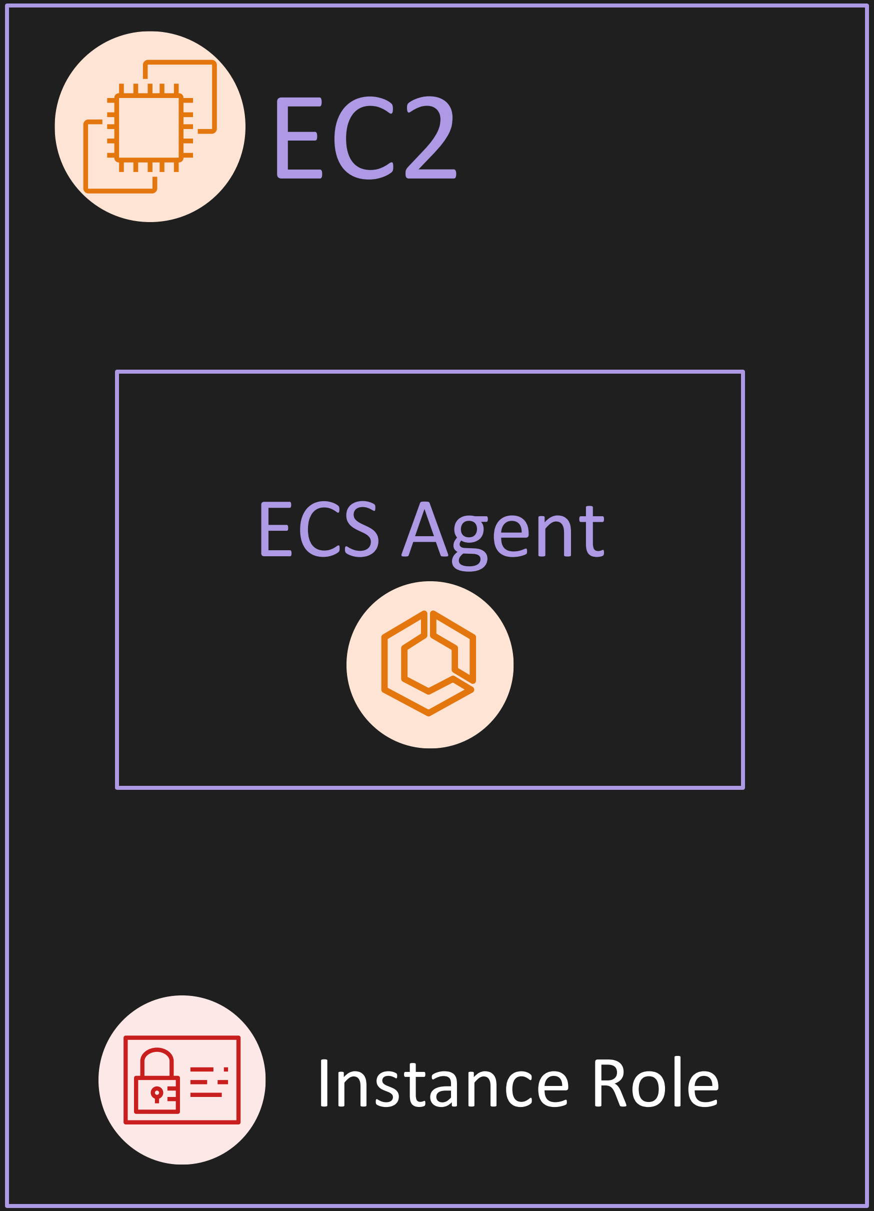
.github	Revert "Run Github action Linux tests on ubuntu-22.04"	4 months ago
agent-container	Add env with required path for appmesh integration.	3 years ago
agent	Release 1.93.1	2 weeks ago
amazon-ecs-cni-plugins @ 7b4ec60	Update amazon-ecs-cni-plugins to 7b4ec60 (#4442)	5 months ago
amazon-vpc-cni-plugins @ be52143	Migrate Agent to use vpc-eni plugin for awsvpc mode ins...	2 years ago
aws-sdk-go-v2	Codegen aws-sdk-go-v2 clients (#4406)	6 months ago
build-infrastructure	Update CodeBuild CF stack template to add disabled enc...	10 months ago
buildspecs	Revert changes adding backup Go installation to pr-buil...	9 months ago
doc	readme: Clean up the README a bit	8 years ago

Instance Role

AmazonEC2ContainerServiceforEC2Role

ecs:RegisterContainerInstance
ecs:DeregisterContainerInstance
ecs:DiscoverPollEndpoint
ecs:Poll

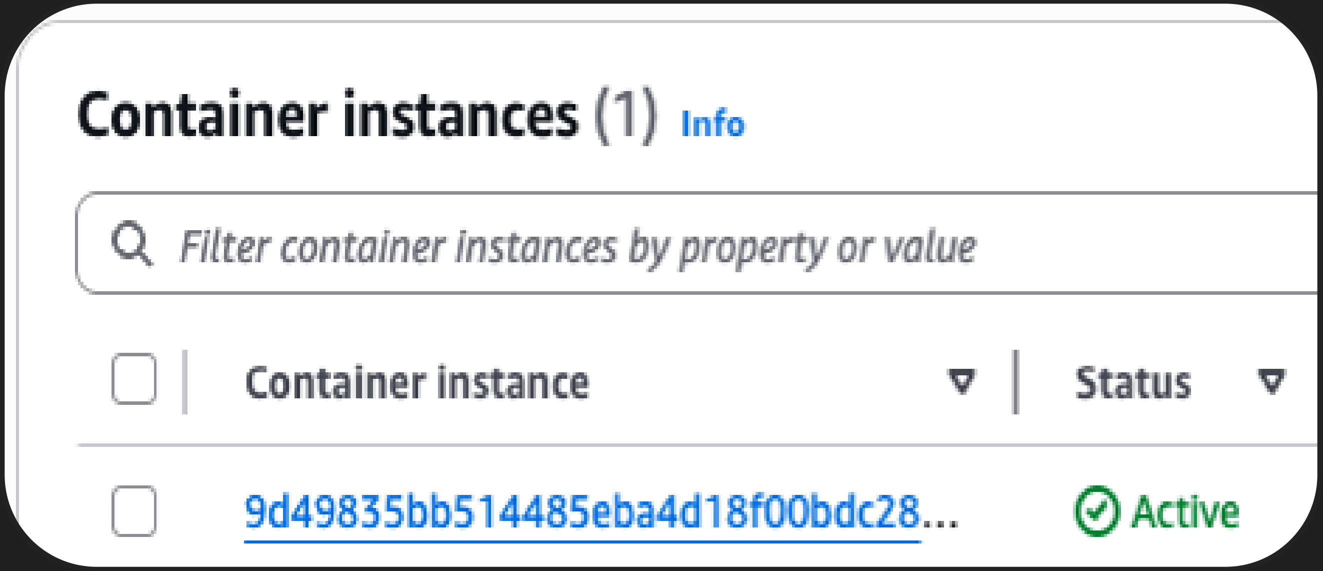

EC2
ECS Agent
ecs:RegisterContainerInstance
Instance Role
RegisterContainerInstance
aws



EC2

ECS Agent

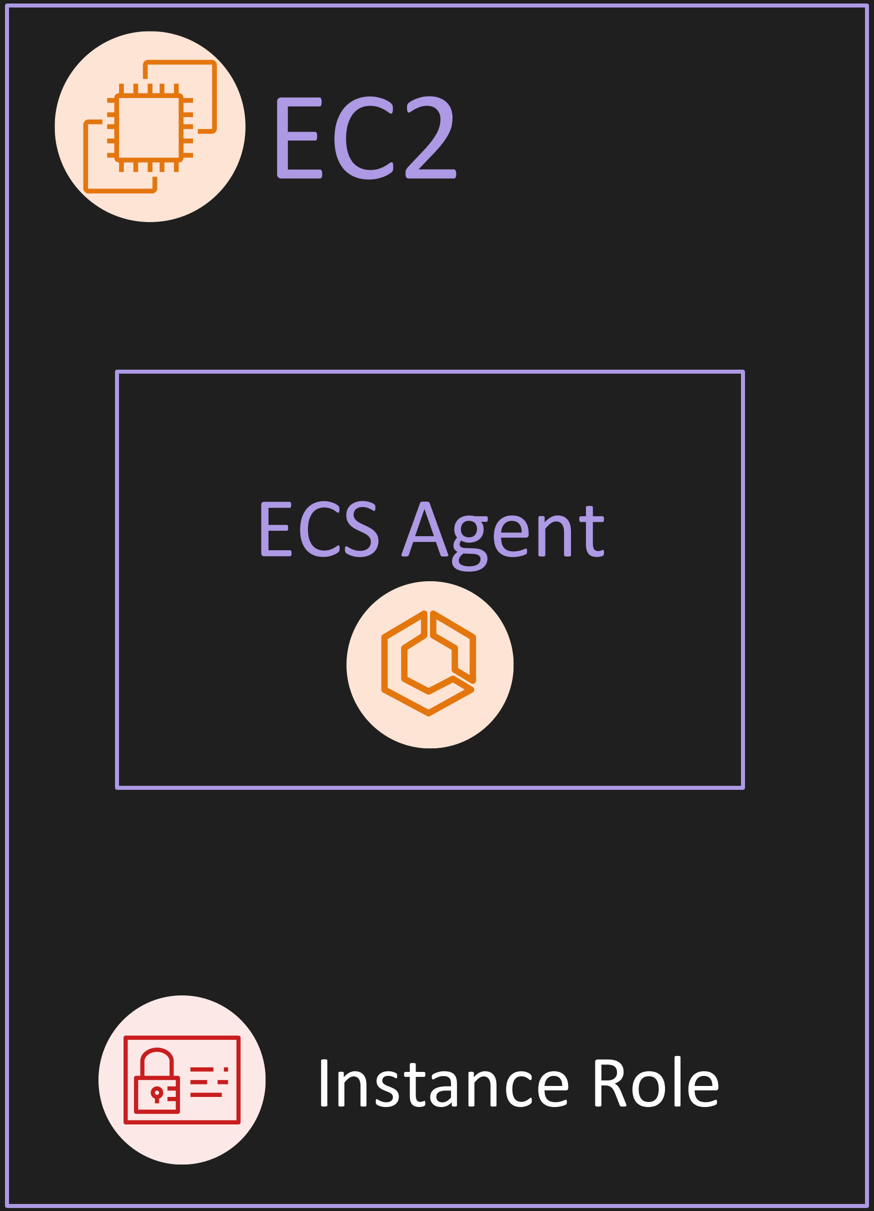
Instance Role



Container instances (1) Info

Filter container instances by property or value

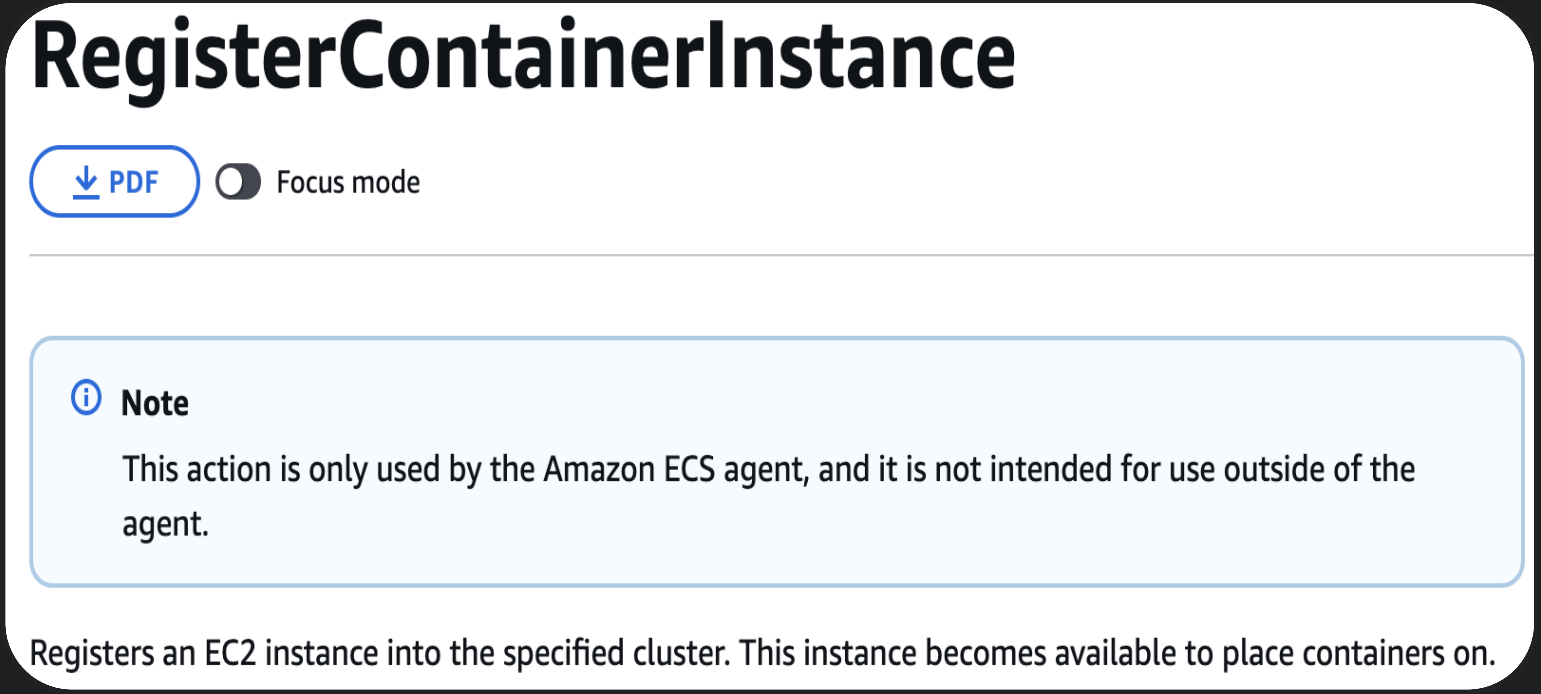
□	Container instance	Status
□	9d49835bb514485eba4d18f00bdc28...	✓ Active



EC2

ECS Agent

Instance Role

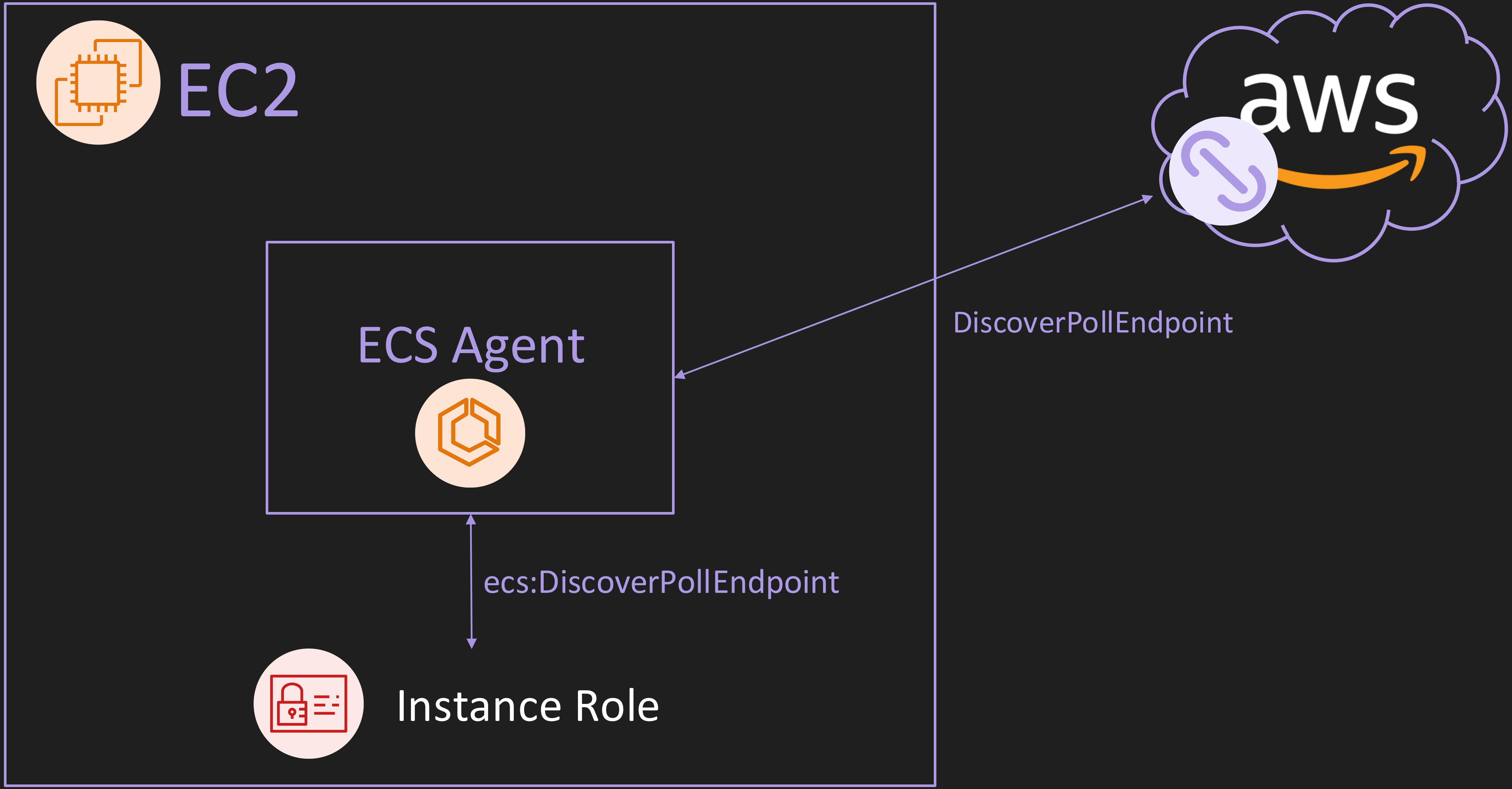


RegisterContainerInstance

Note

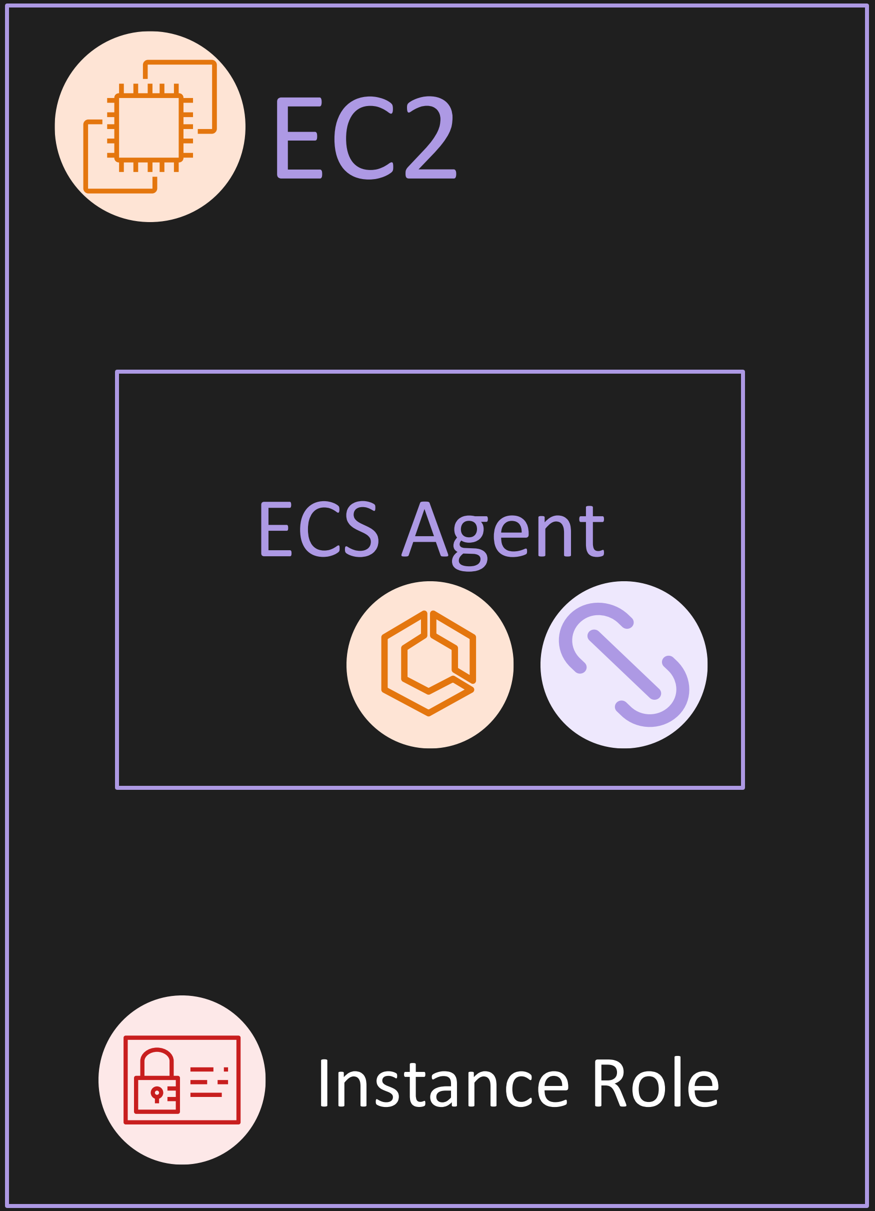
This action is only used by the Amazon ECS agent, and it is not intended for use outside of the agent.

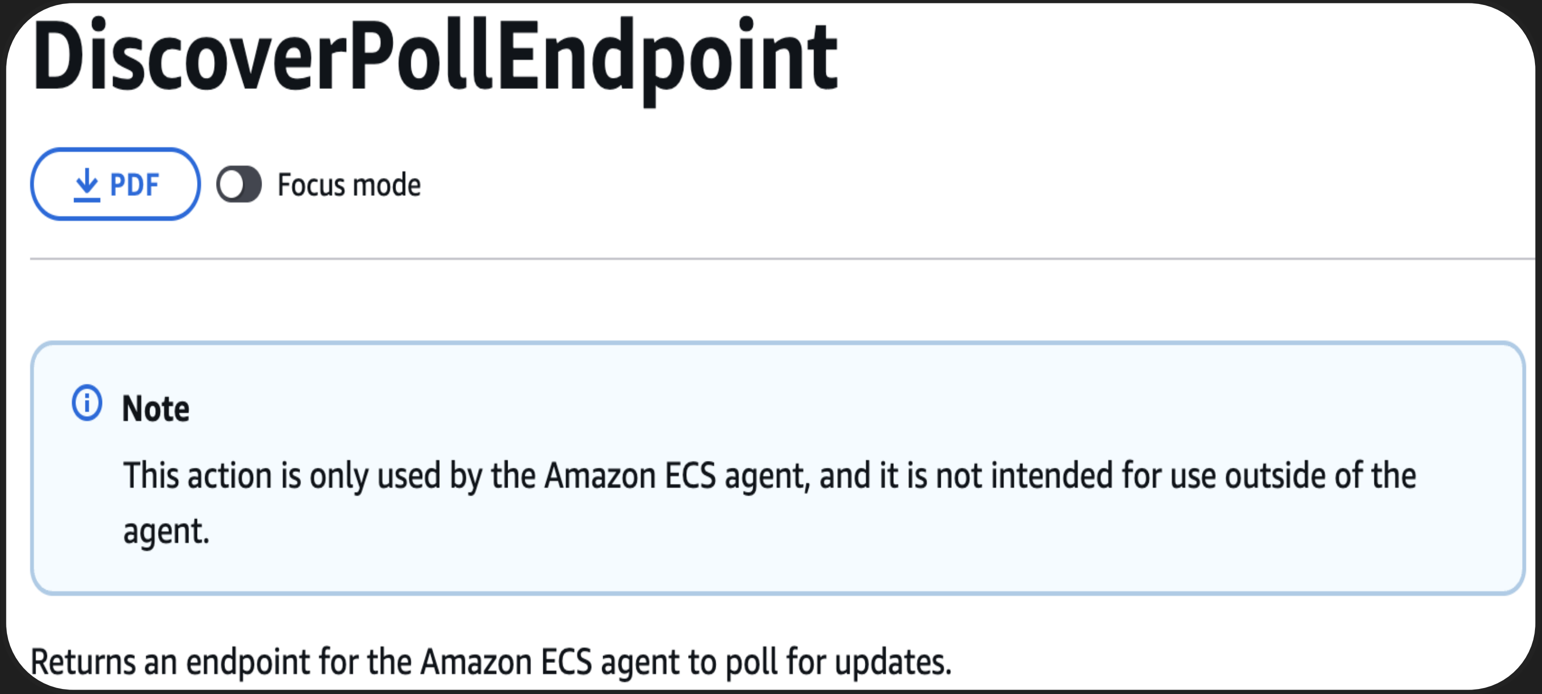
Registers an EC2 instance into the specified cluster. This instance becomes available to place containers on.


EC2
aws
ECS Agent
DiscoverPollEndpoint
ecs:DiscoverPollEndpoint
Instance Role

Poll Endpoint URL

https://ecs-a-1.us-east-2.amazonaws.com



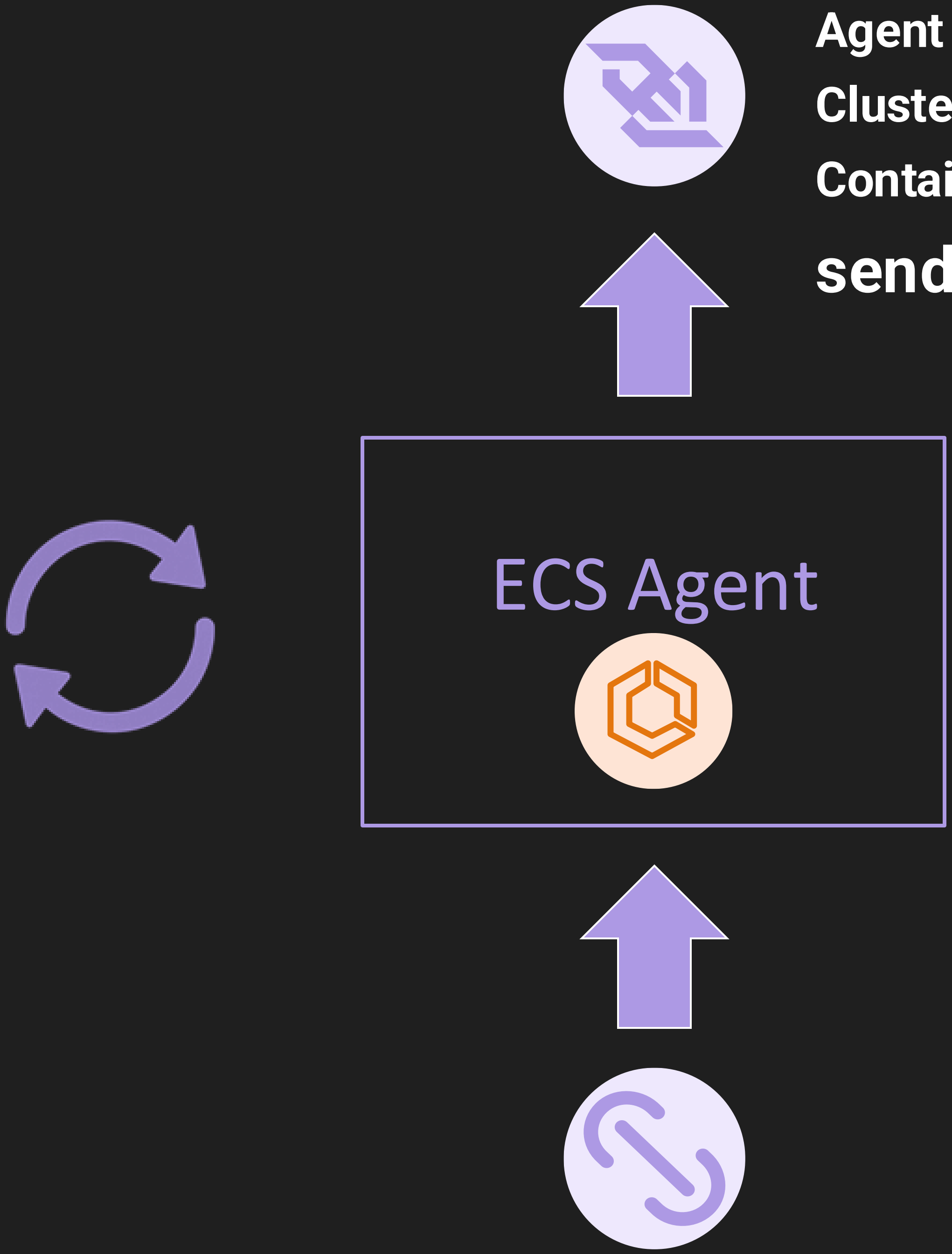


DiscoverPollEndpoint

Note

This action is only used by the Amazon ECS agent, and it is not intended for use outside of the agent.

Returns an endpoint for the Amazon ECS agent to poll for updates.

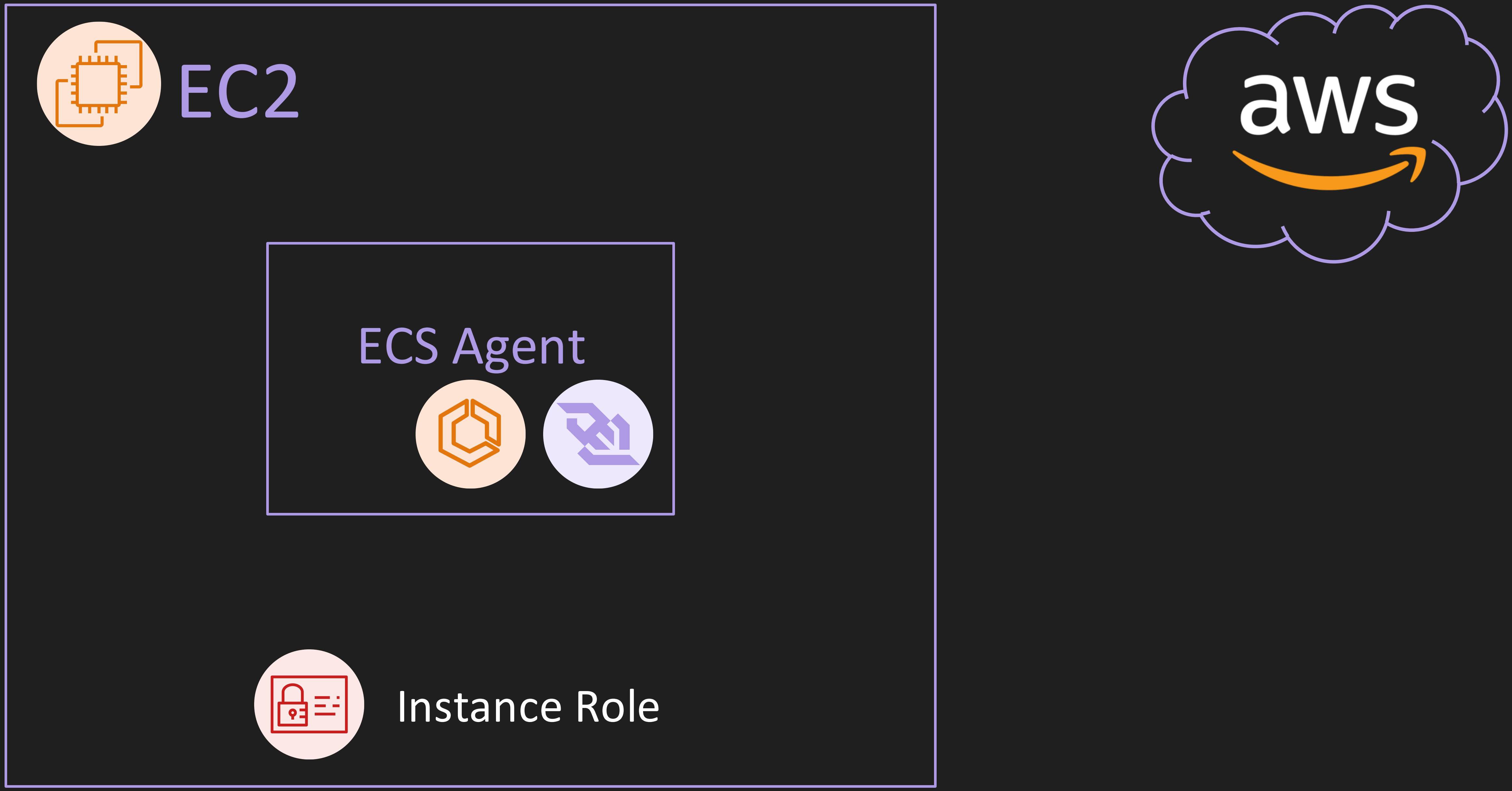


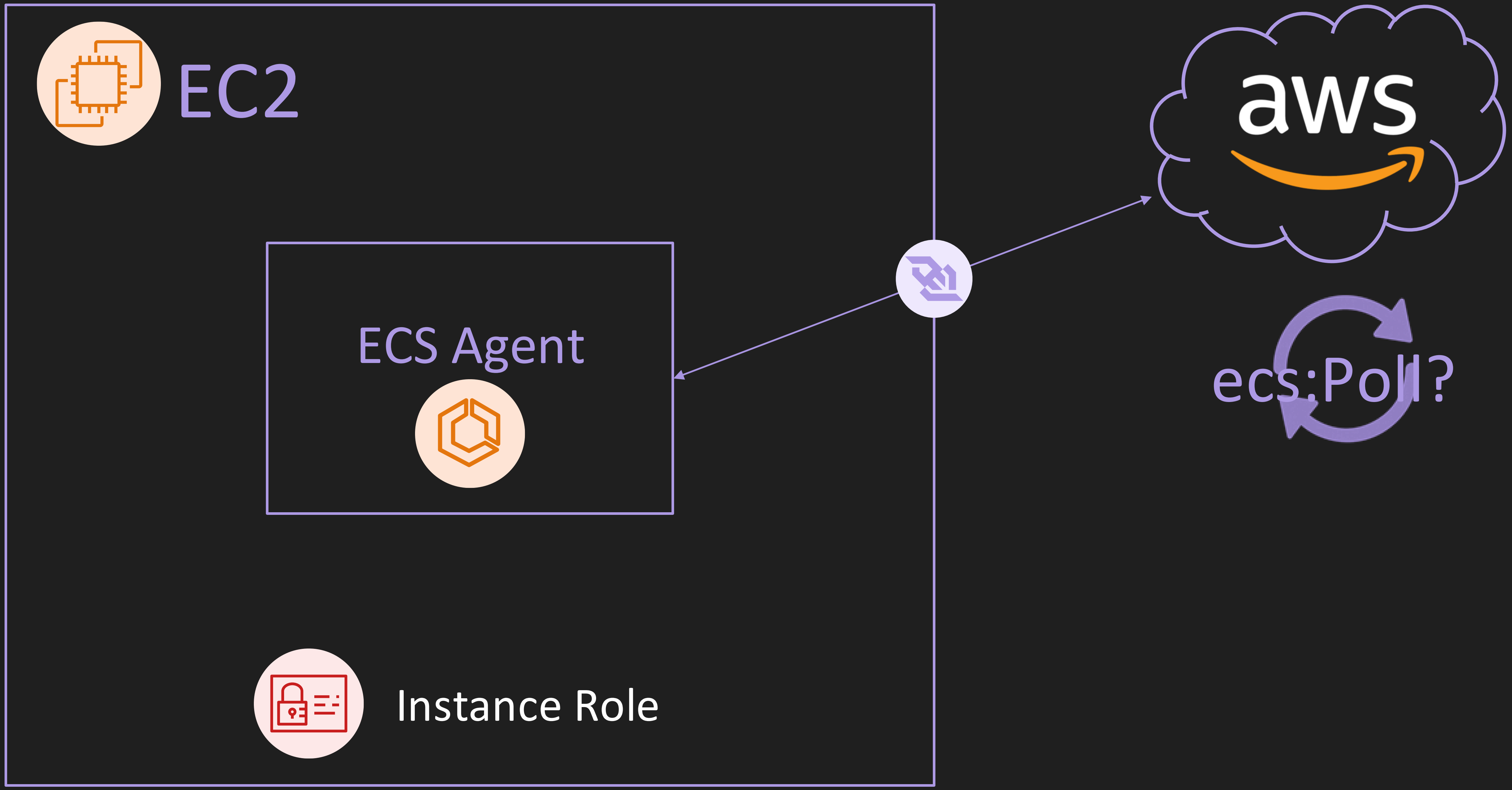
Agent Version

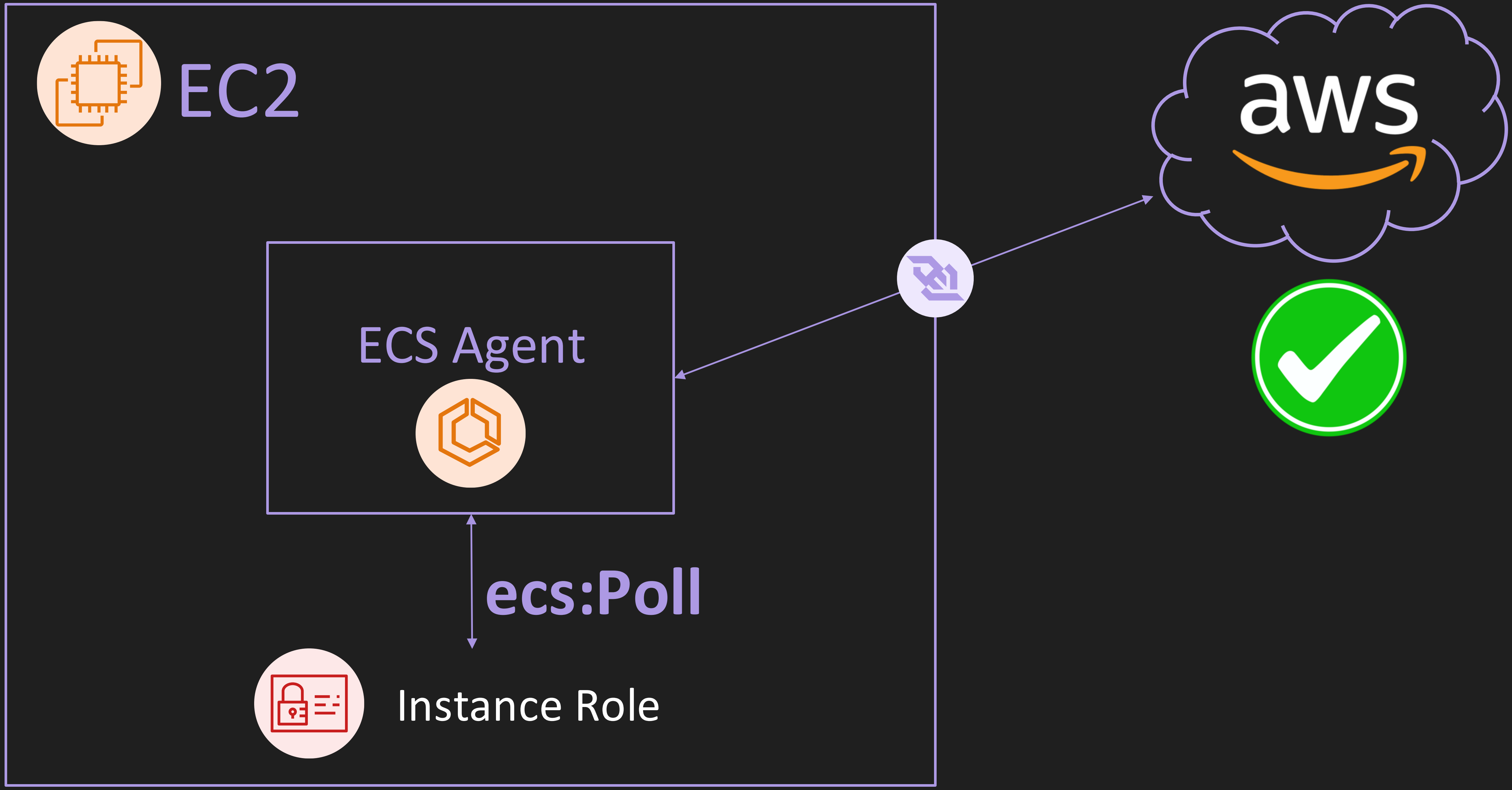
Cluster ARN

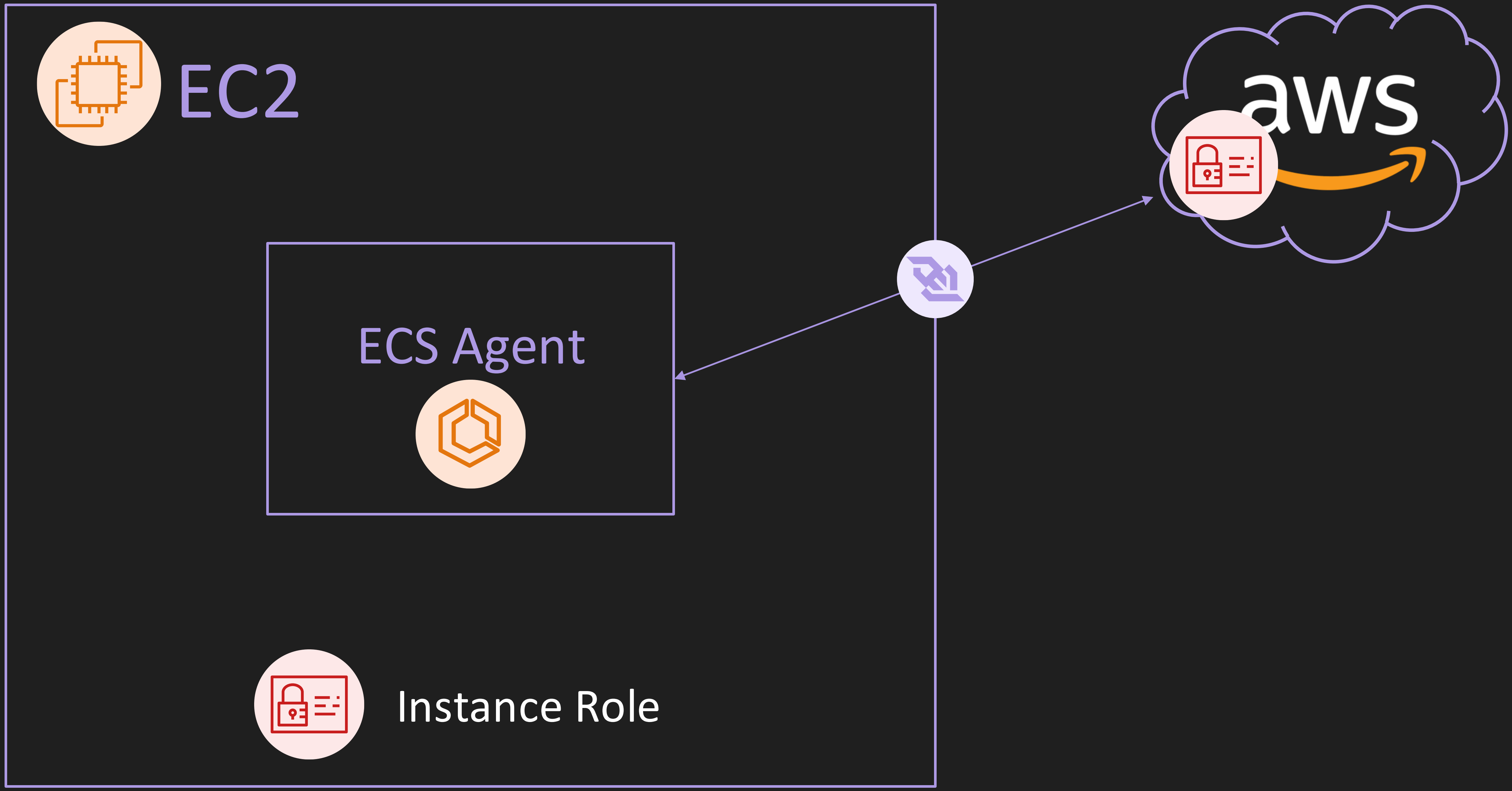
Container Instance ARN ...

sendCredentials=true


EC2
ECS Agent
Instance Role
aws


EC2
ECS Agent
Instance Role
aws
ecs:Poll?


EC2
ECS Agent
ecs:Poll
Instance Role
aws


EC2
ECS Agent
Instance Role
aws

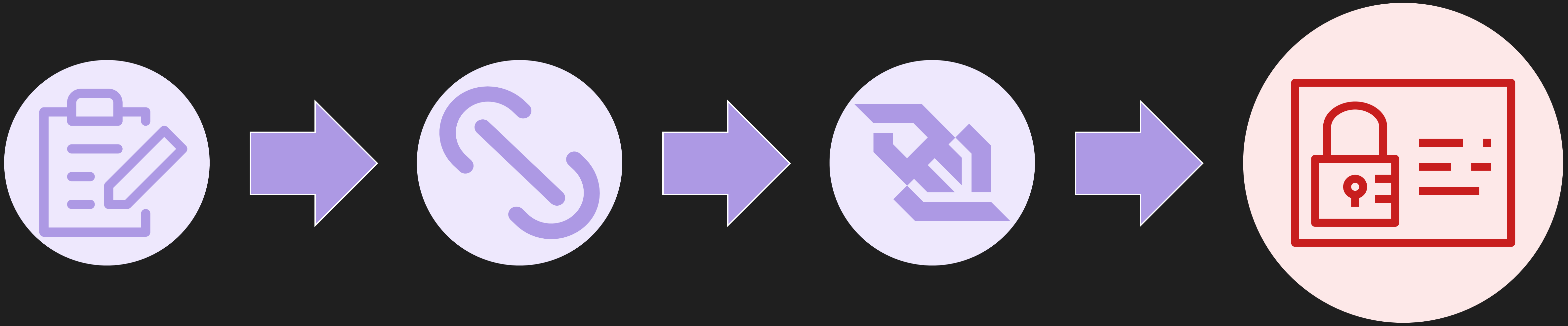
ACS – Agent Communication Service

Task Metadata

Agent-Level Directives

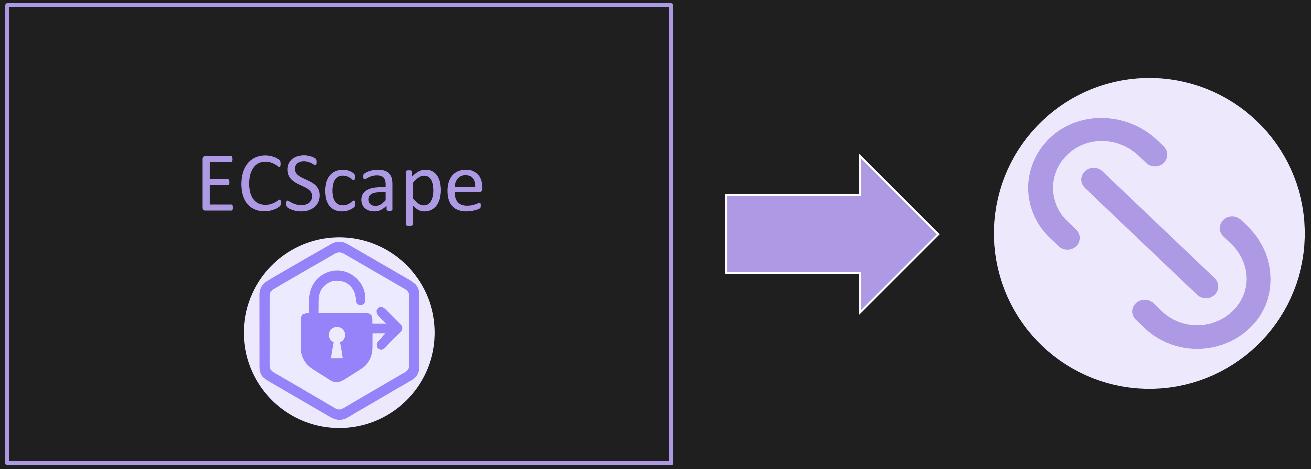
IAM Credentials

ECS Agent – Authentication Flow



Can a Task Impersonate the ECS Agent?

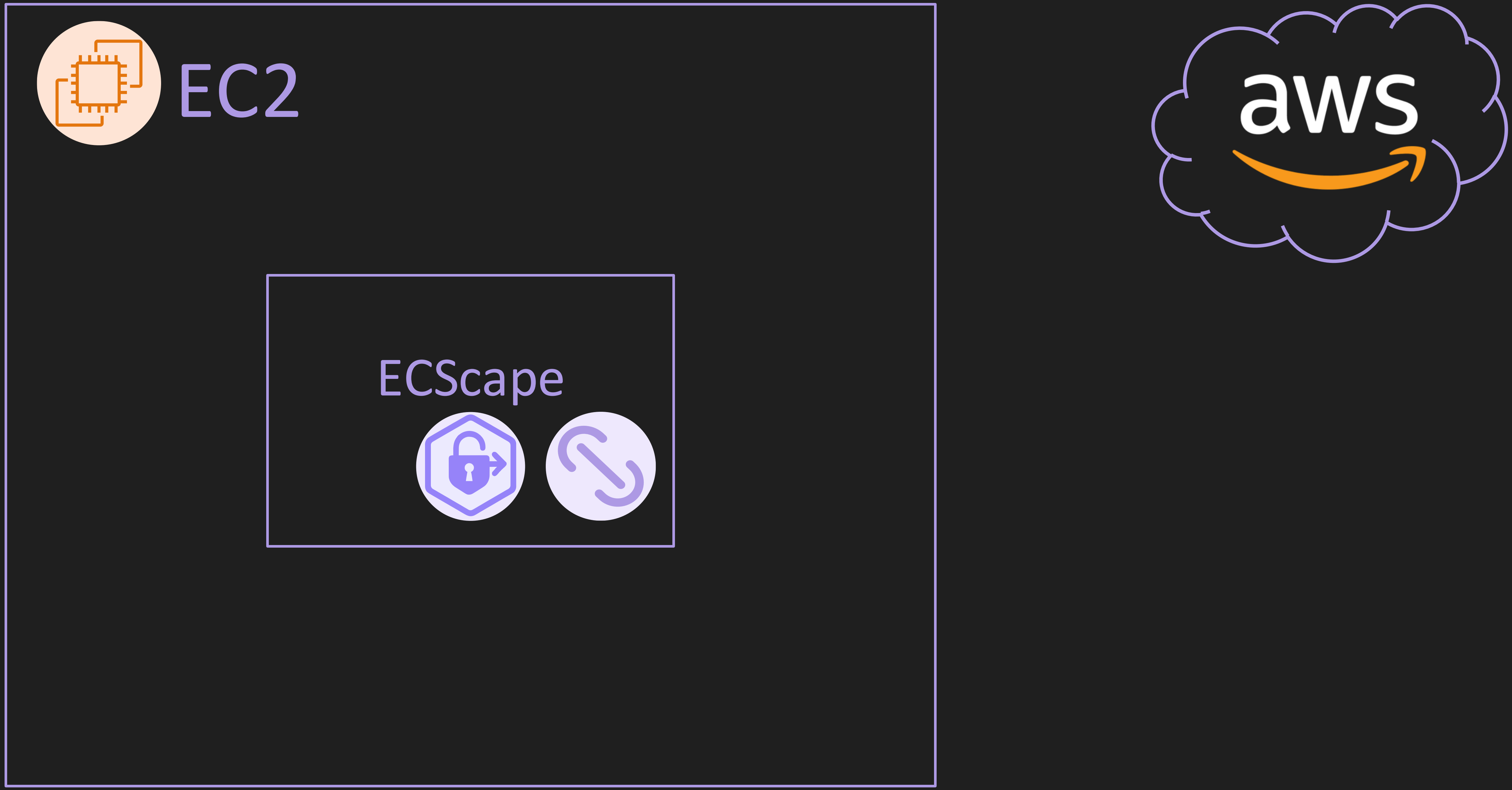
ecs:DiscoverPollEndpoint?

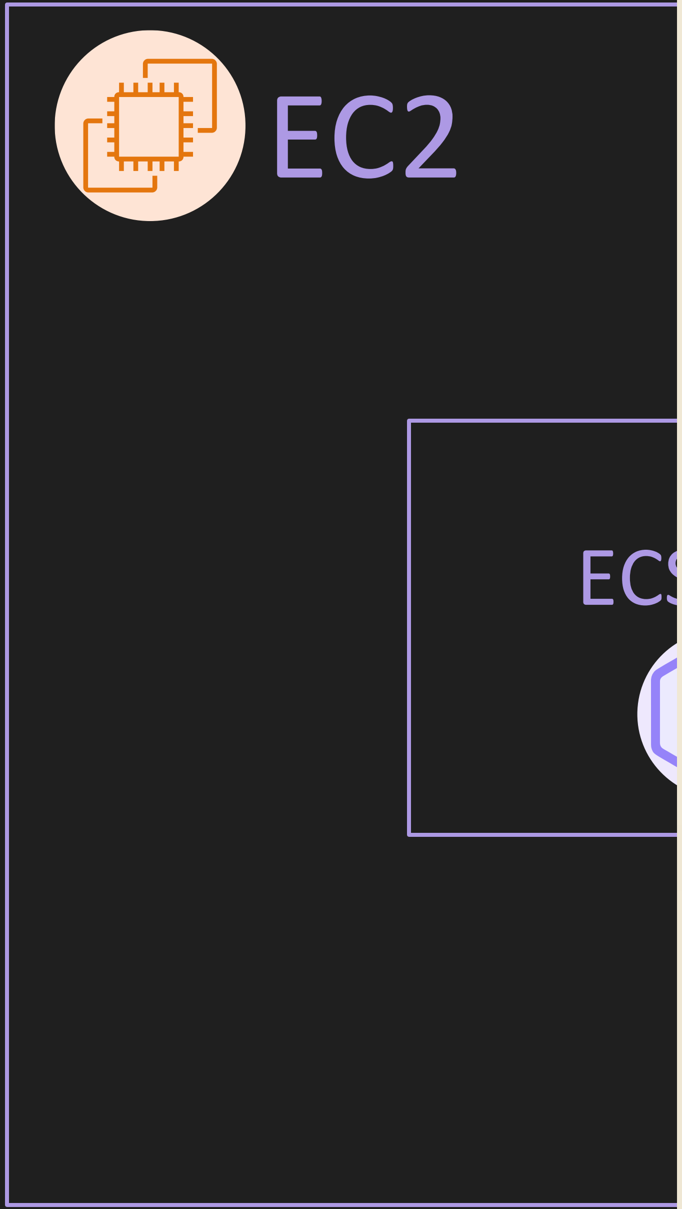


Brute Force

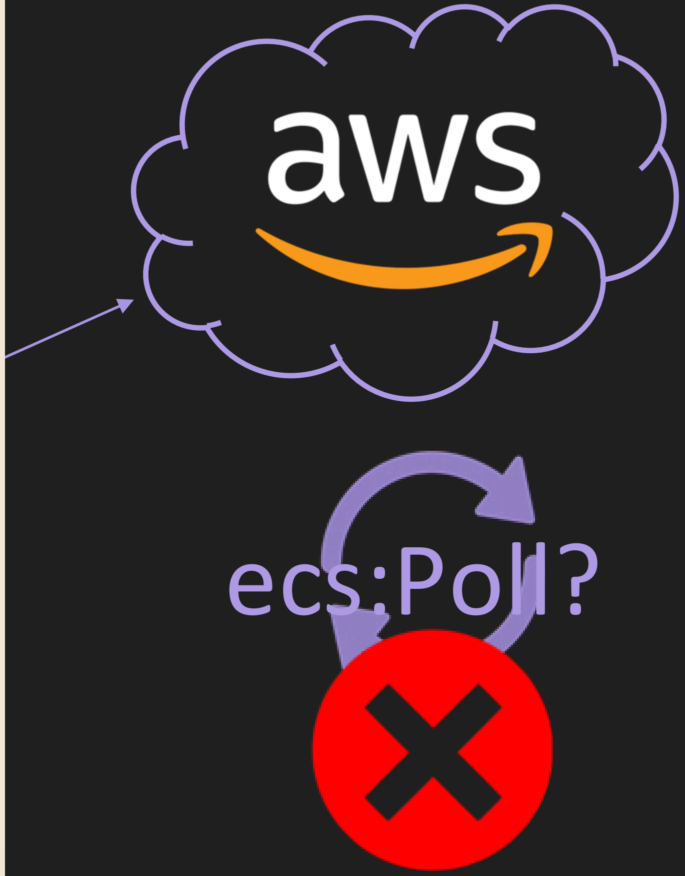
Poll Endpoint URL

https://ecs-a-**1**.<REGION>.amazonaws.com

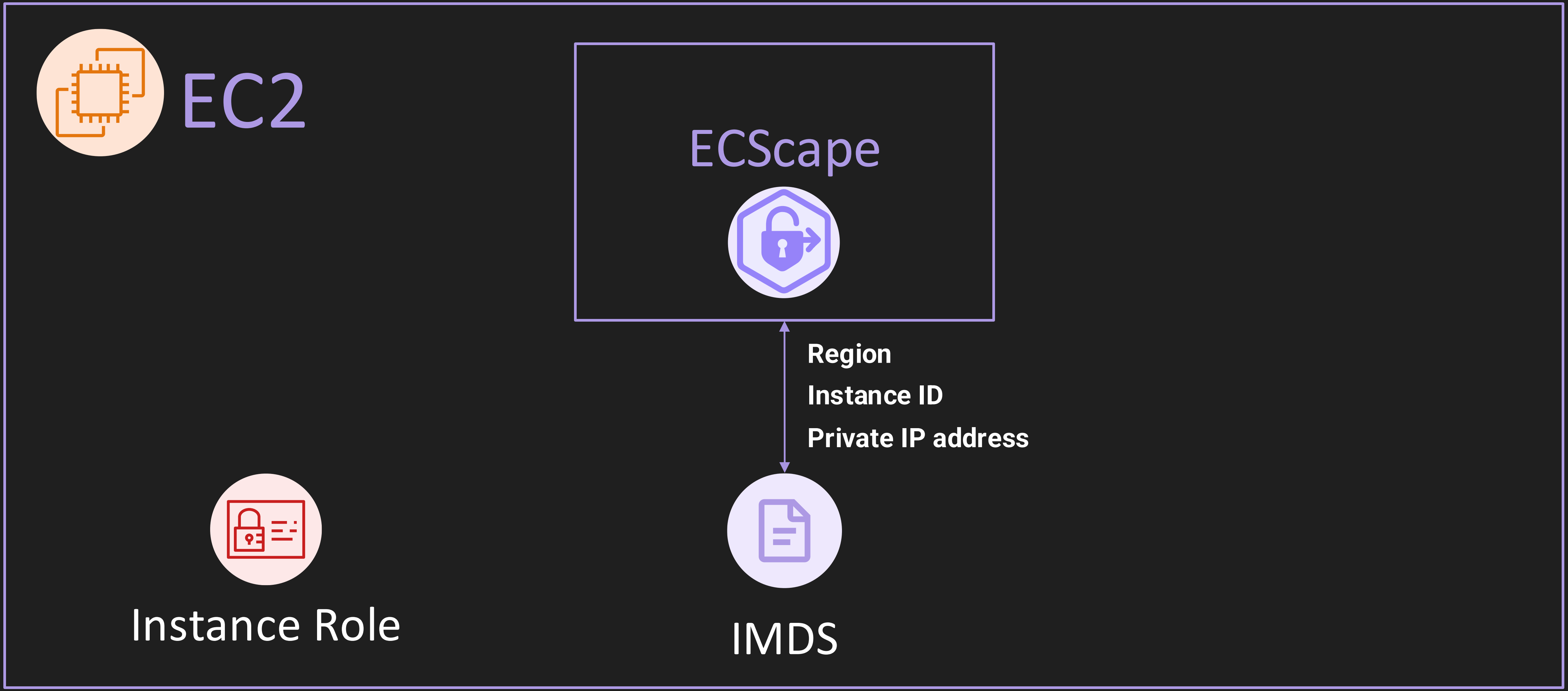

EC2
ECScape
aws

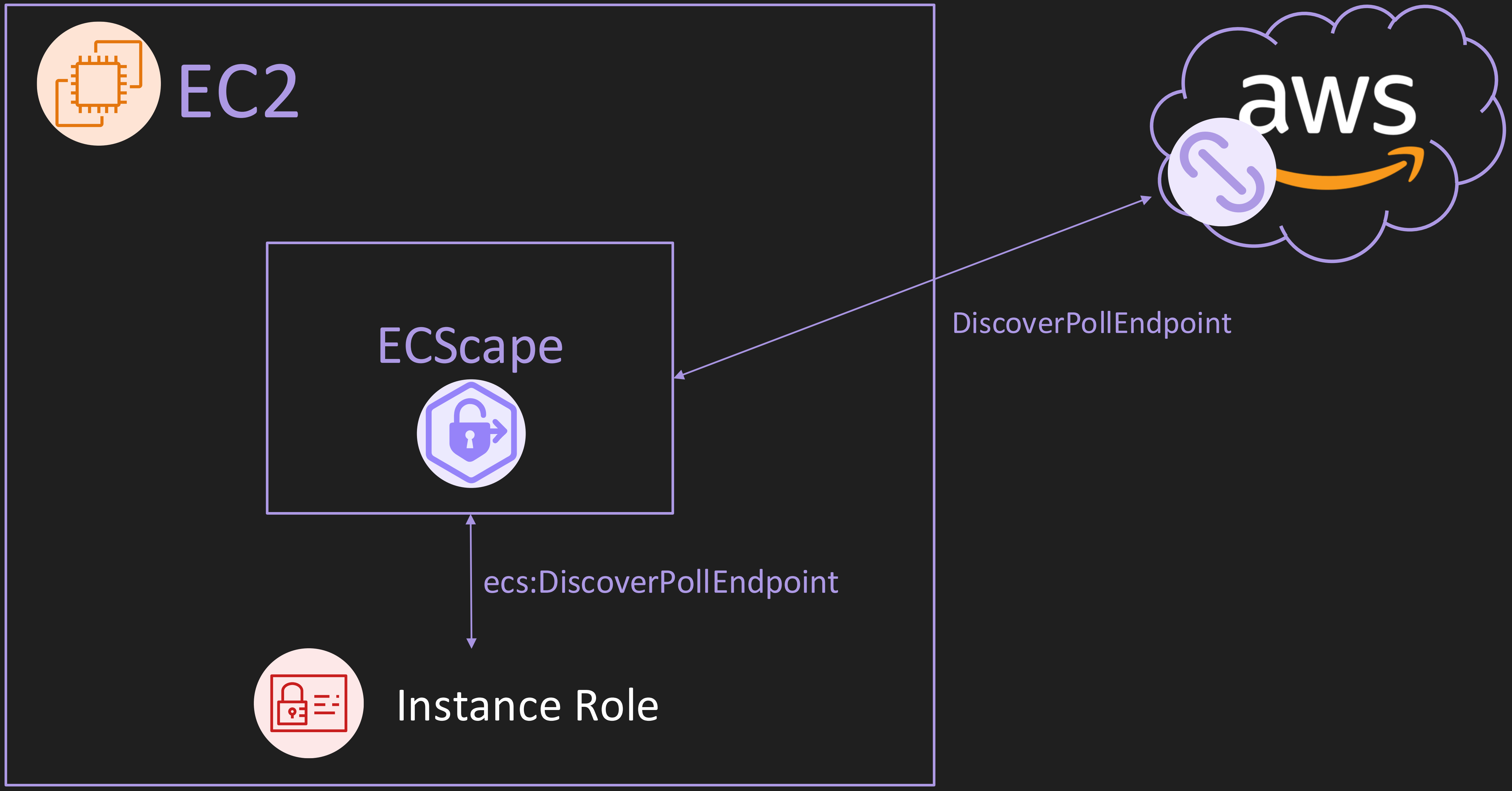

EC2
ECS

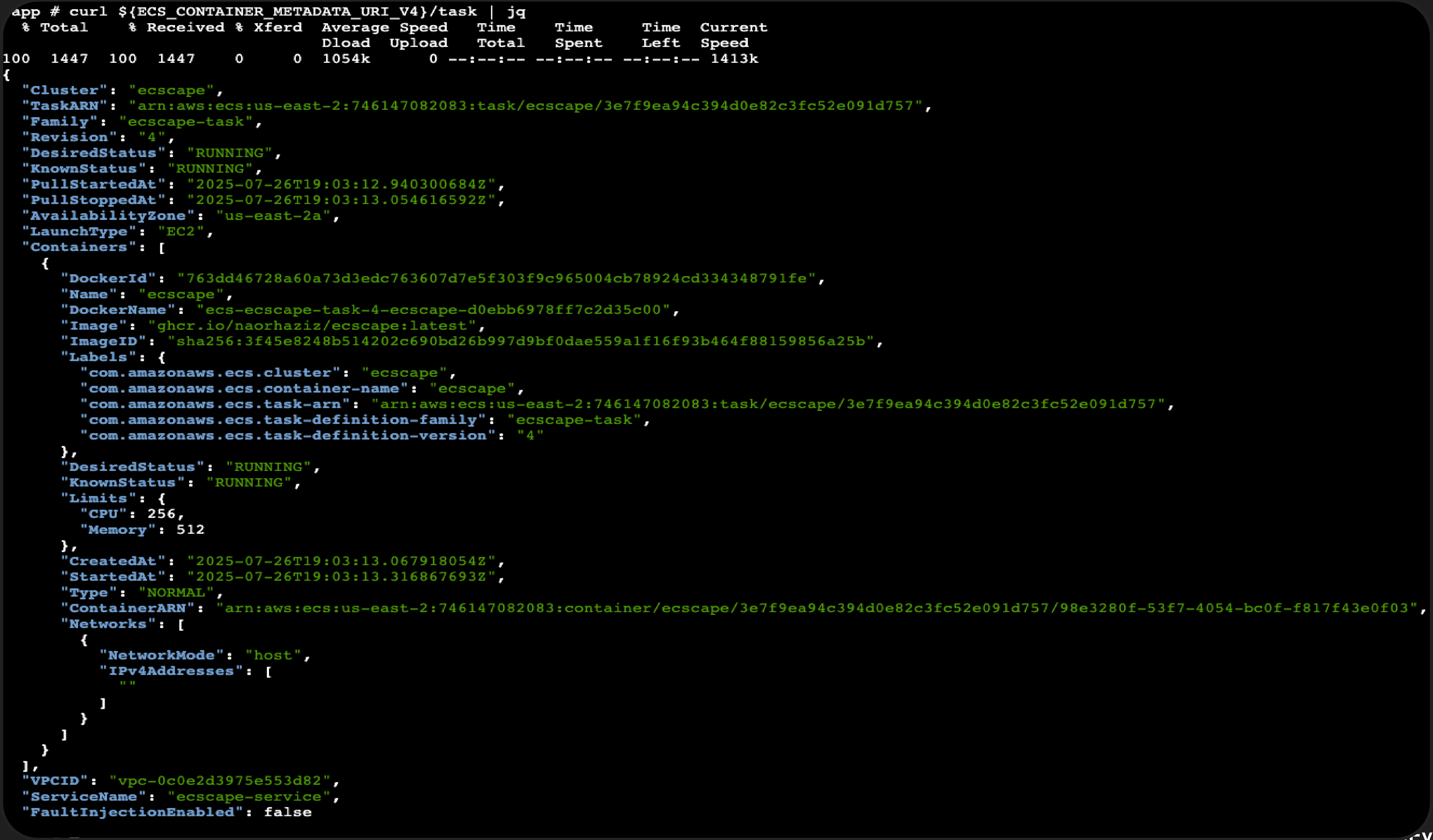
idea


aws
ecs:Poll?

IMDS – Instance Metadata Service



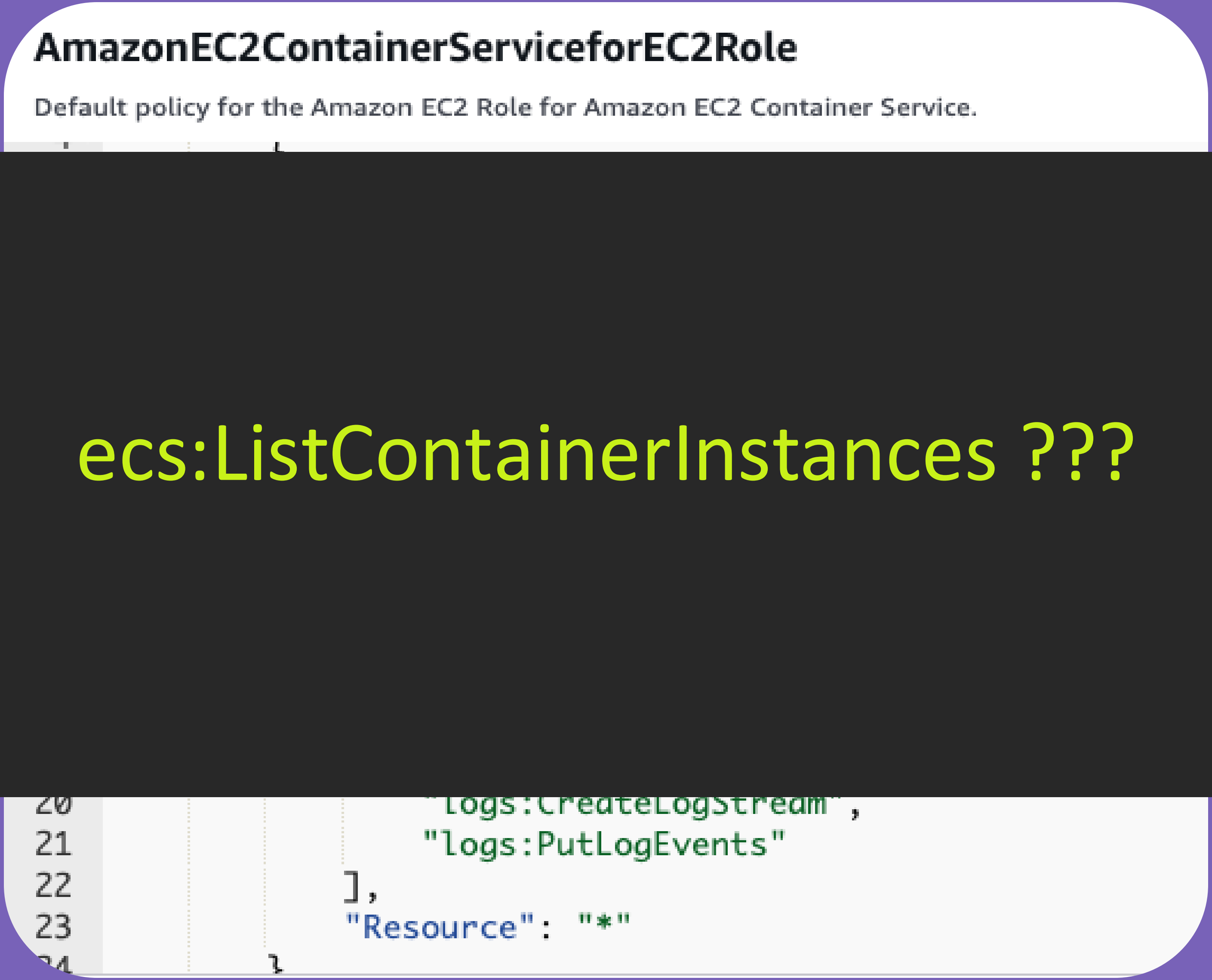

EC2
aws
ECScape
DiscoverPollEndpoint
ecs:DiscoverPollEndpoint
Instance Role



```
app # curl ${ECS_CONTAINER_METADATA_URI_V4}/task | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1447  100  1447    0     0  1054k      0 --:--:-- --:--:-- --:--:-- 1413k
{
  "Cluster": "ecscape",
  "TaskARN": "arn:aws:ecs:us-east-2:746147082083:task/ecscape/3e7f9ea94c394d0e82c3fc52e091d757",
  "Family": "ecscape-task",
  "Revision": "4",
  "DesiredStatus": "RUNNING",
  "KnownStatus": "RUNNING",
  "PullStartedAt": "2025-07-26T19:03:12.940300684Z",
  "PullStoppedAt": "2025-07-26T19:03:13.054616592Z",
  "AvailabilityZone": "us-east-2a",
  "LaunchType": "EC2",
  "Containers": [
    {
      "DockerId": "763dd46728a60a73d3edc763607d7e5f303f9c965004cb78924cd334348791fe",
      "Name": "ecscape",
      "DockerName": "ecs-ecscape-task-4-ecscape-d0ebb6978ff7c2d35c00",
      "Image": "ghcr.io/naorhaziz/ecscape:latest",
      "ImageID": "sha256:3f45e8248b514202c690bd26b997d9bf0dae559a1f16f93b464f88159856a25b",
      "Labels": {
        "com.amazonaws.ecs.cluster": "ecscape",
        "com.amazonaws.ecs.container-name": "ecscape",
        "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-east-2:746147082083:task/ecscape/3e7f9ea94c394d0e82c3fc52e091d757",
        "com.amazonaws.ecs.task-definition-family": "ecscape-task",
        "com.amazonaws.ecs.task-definition-version": "4"
      },
      "DesiredStatus": "RUNNING",
      "KnownStatus": "RUNNING",
      "Limits": {
        "CPU": 256,
        "Memory": 512
      },
      "CreatedAt": "2025-07-26T19:03:13.067918054Z",
      "StartedAt": "2025-07-26T19:03:13.316867693Z",
      "Type": "NORMAL",
      "ContainerARN": "arn:aws:ecs:us-east-2:746147082083:container/ecscape/3e7f9ea94c394d0e82c3fc52e091d757/98e3280f-53f7-4054-bc0f-f817f43e0f03",
      "Networks": [
        {
          "NetworkMode": "host",
          "IPv4Addresses": [
            ""
          ]
        }
      ]
    }
  ],
  "VPCID": "vpc-0c0e2d3975e553d82",
  "ServiceName": "ecscape-service",
  "FaultInjectionEnabled": false
```

Exce
Con
Inst

Container Instance ARN?



AmazonEC2ContainerServiceforEC2Role

Default policy for the Amazon EC2 Role for Amazon EC2 Container Service.

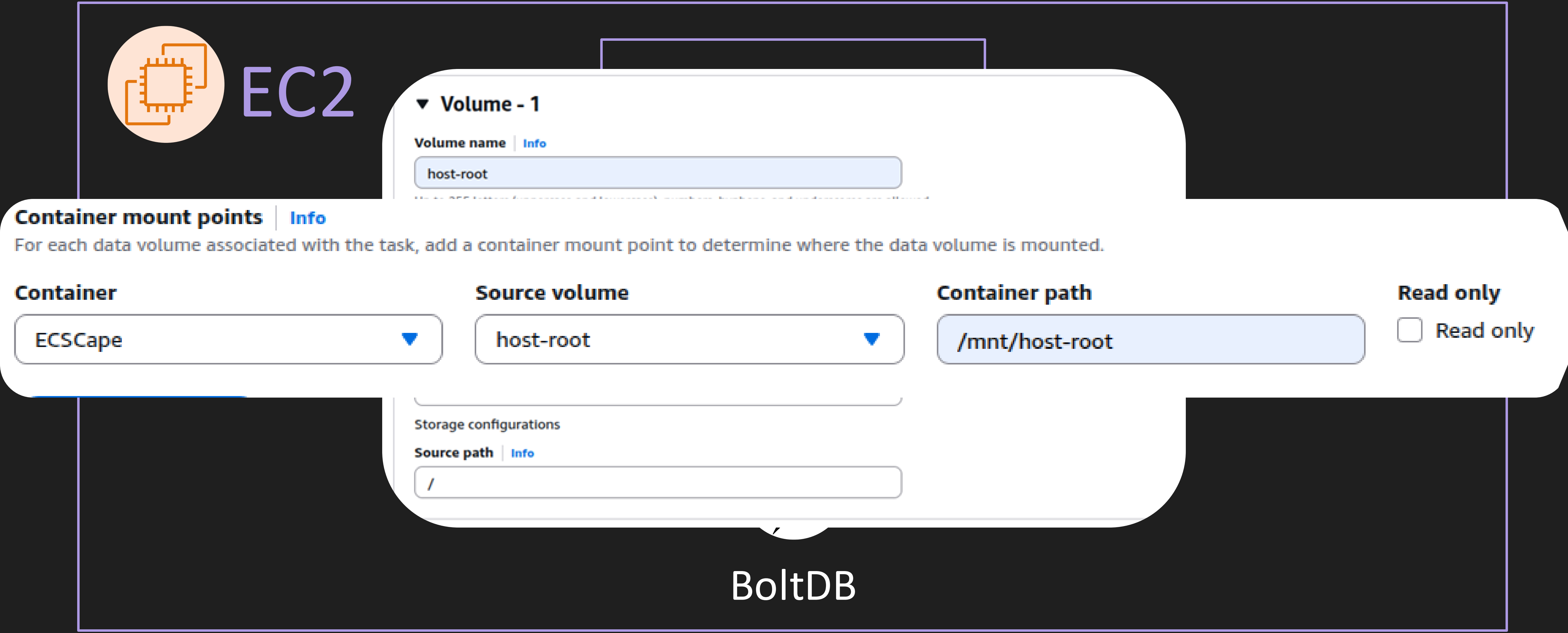
```
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        }
```

ecs:ListContainerInstances ???

EC2

stance

Container Instance ARN?



▼ Volume - 1

Volume name | Info

host-root

Storage configurations

Source path | Info

/

Container mount points | Info

For each data volume associated with the task, add a container mount point to determine where the data volume is mounted.

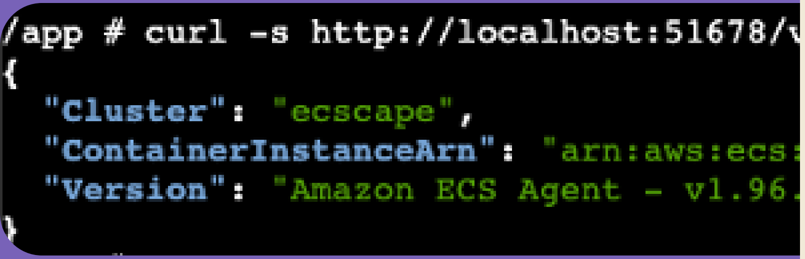
Container	Source volume	Container path	Read only
ECSCape ▼	host-root ▼	/mnt/host-root	□ Read only

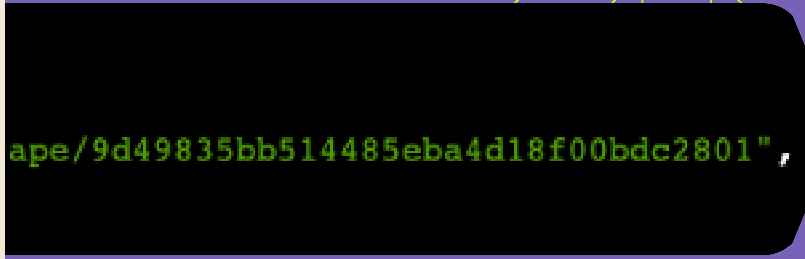
Containe

EC2

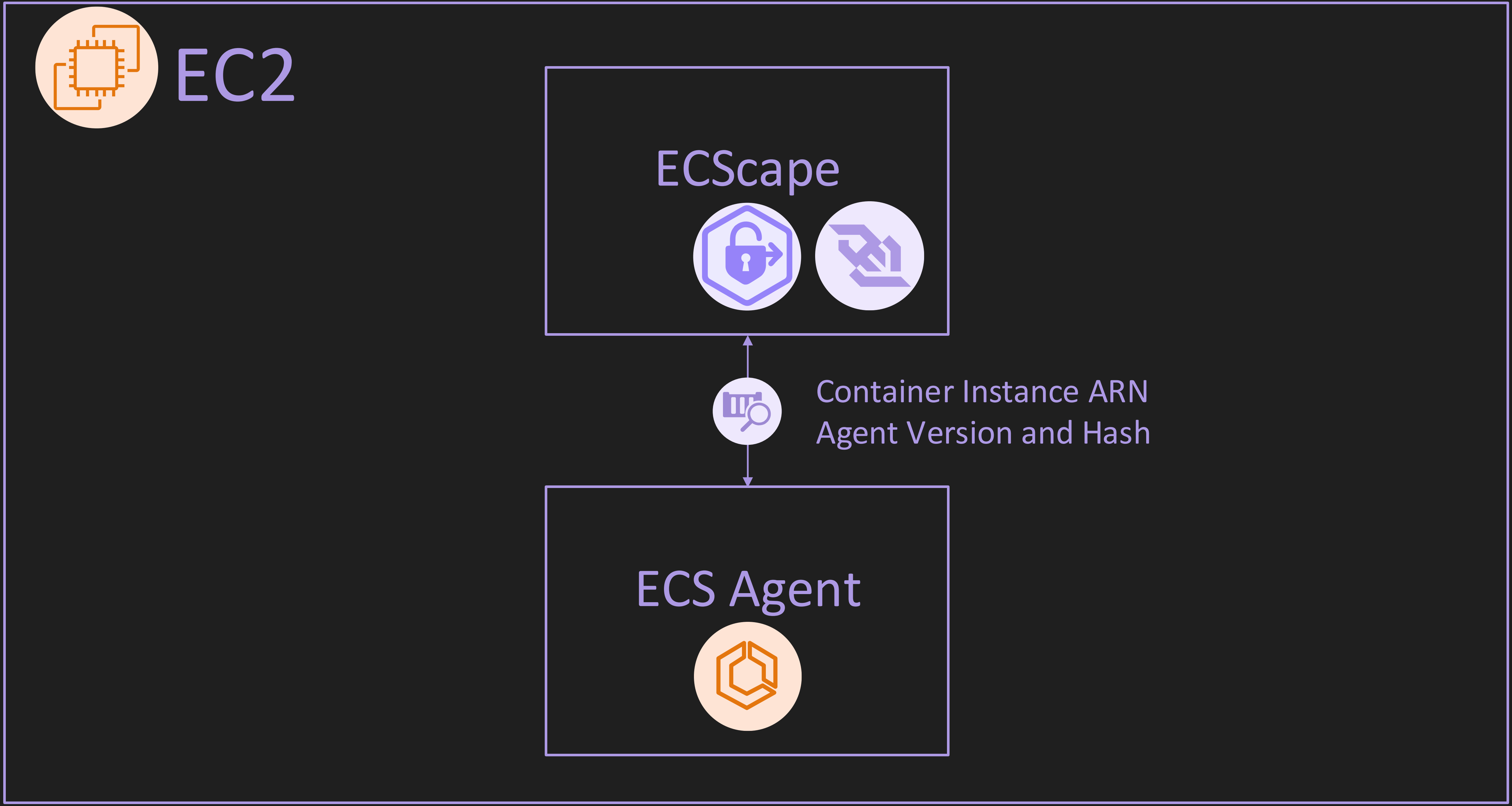
t

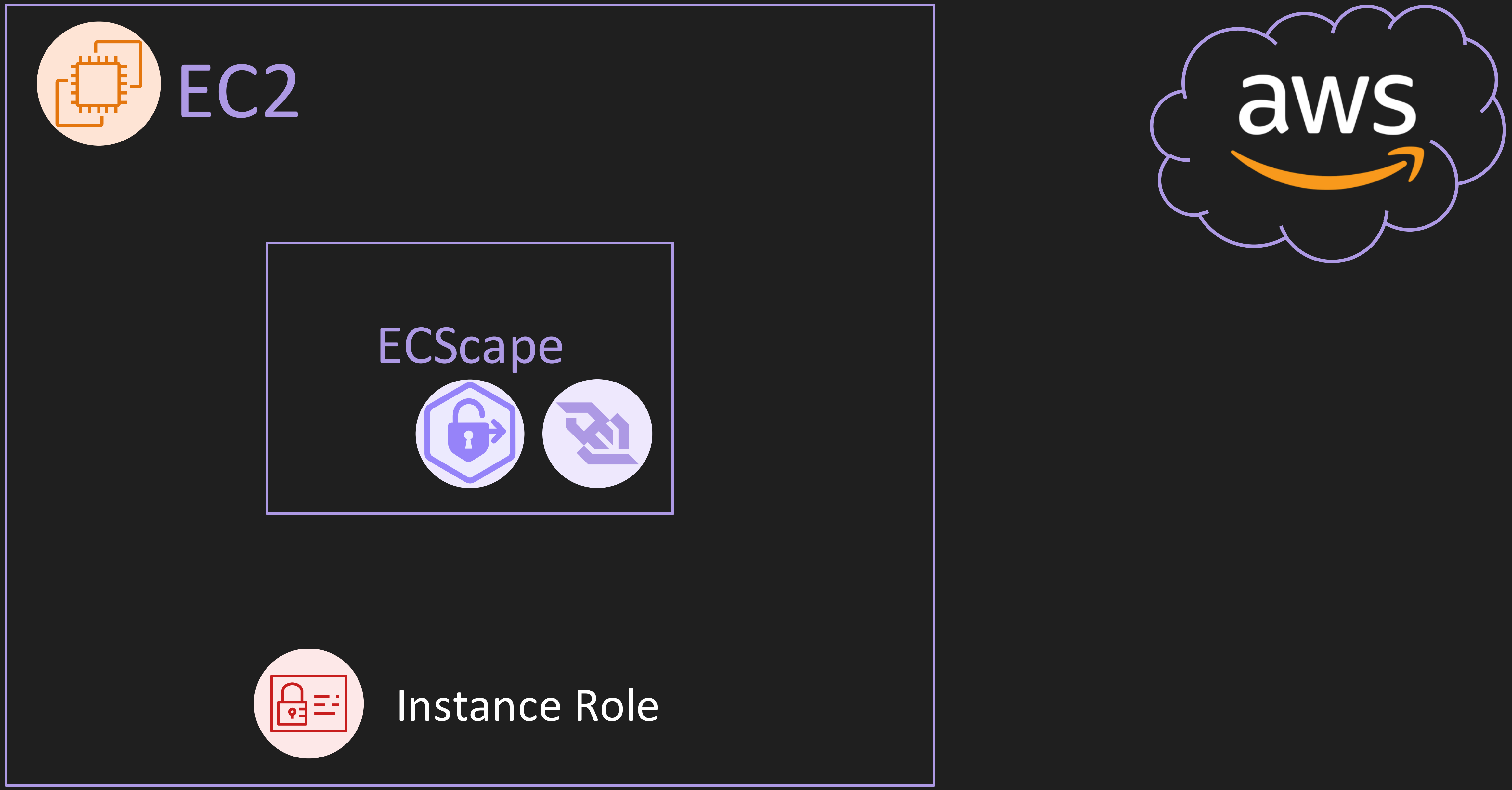
Amazon ECS ection





```
/app # curl -s http://localhost:51678/v
{
  "Cluster": "ecscape",
  "ContainerInstanceArn": "arn:aws:ecs:                    ape/9d49835bb514485eba4d18f00bdc2801",
  "Version": "Amazon ECS Agent - v1.96.
}
```

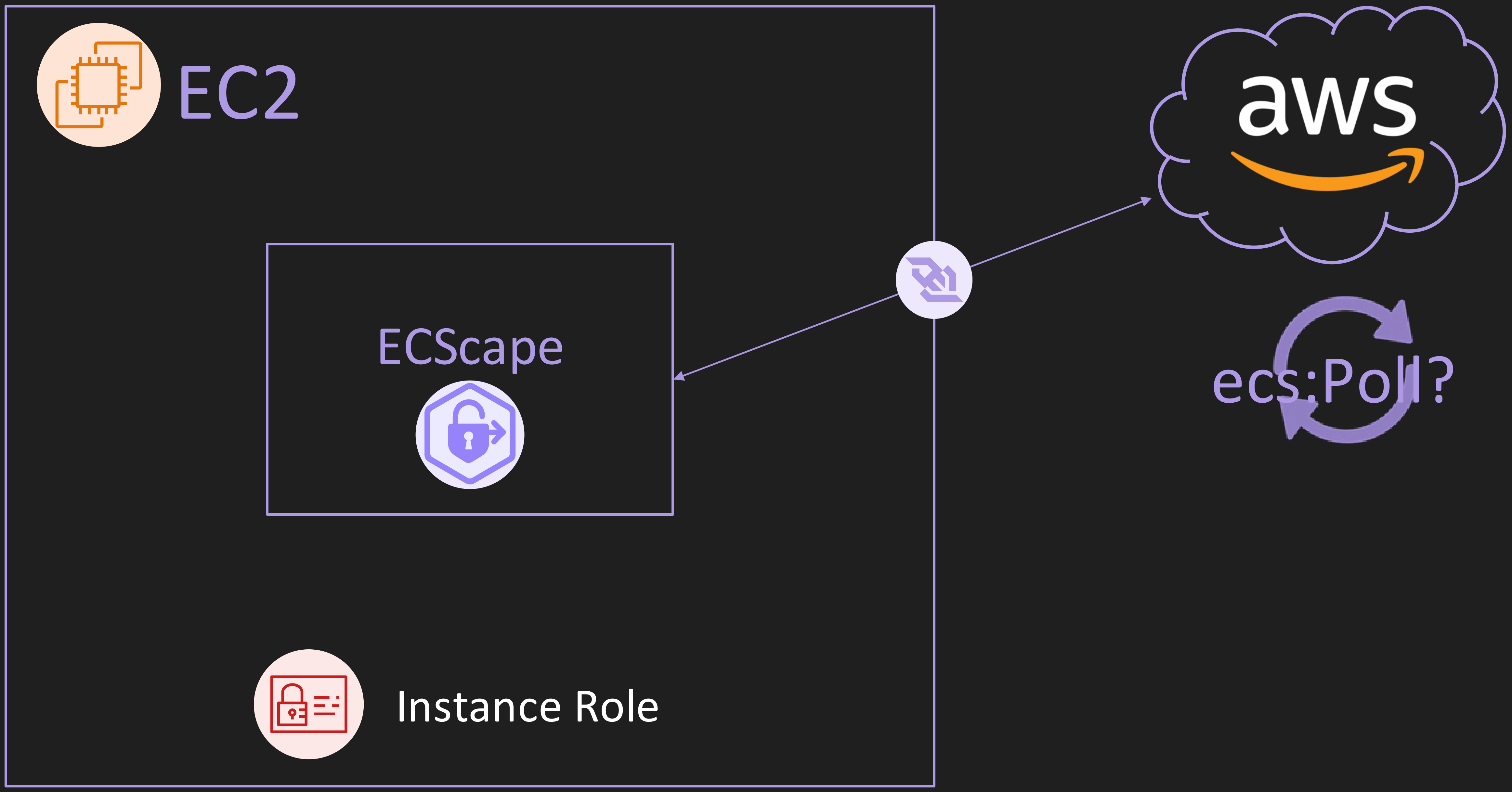

EC2
ECScape
Container Instance ARN
Agent Version and Hash
ECS Agent

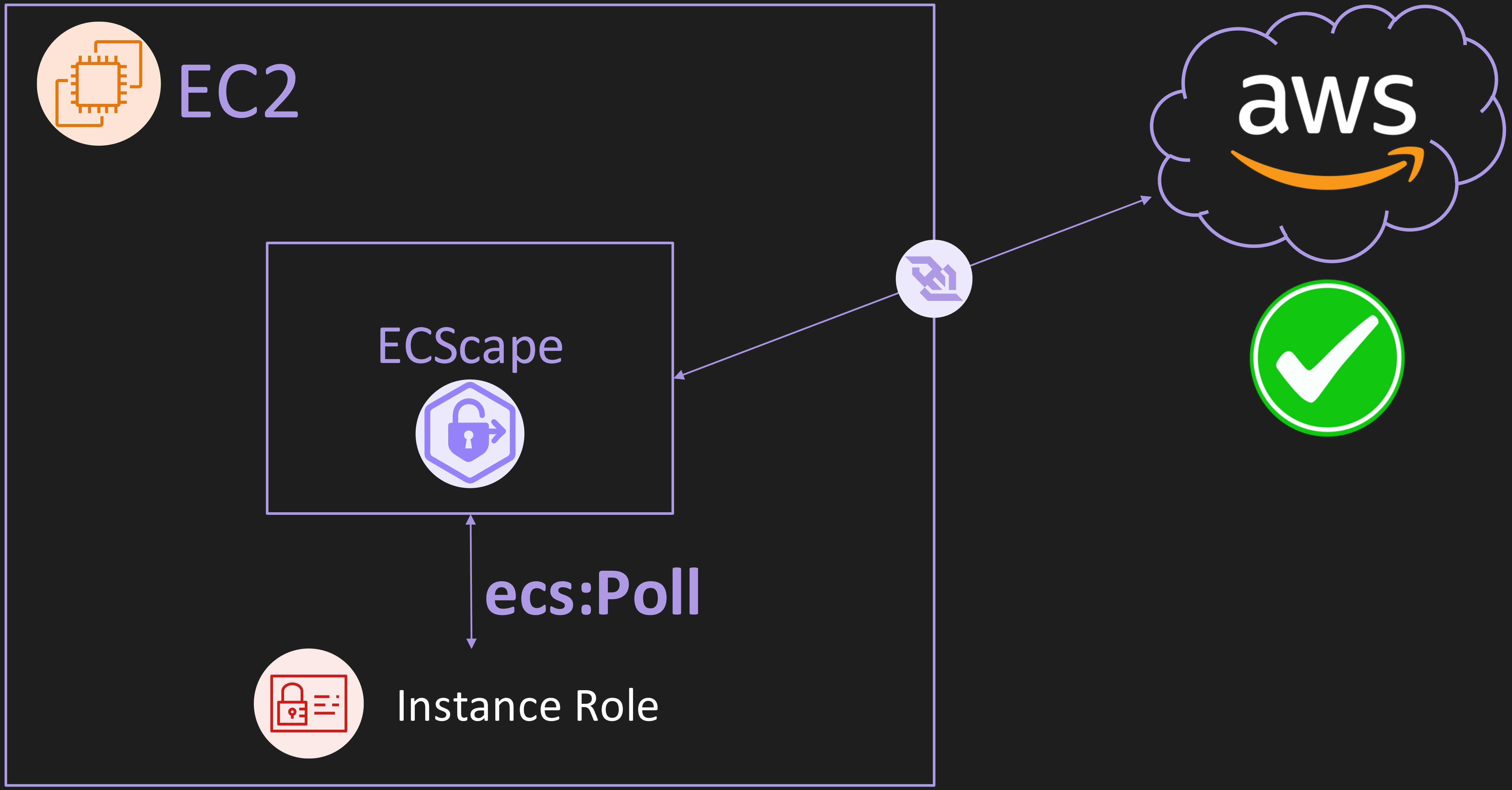


EC2

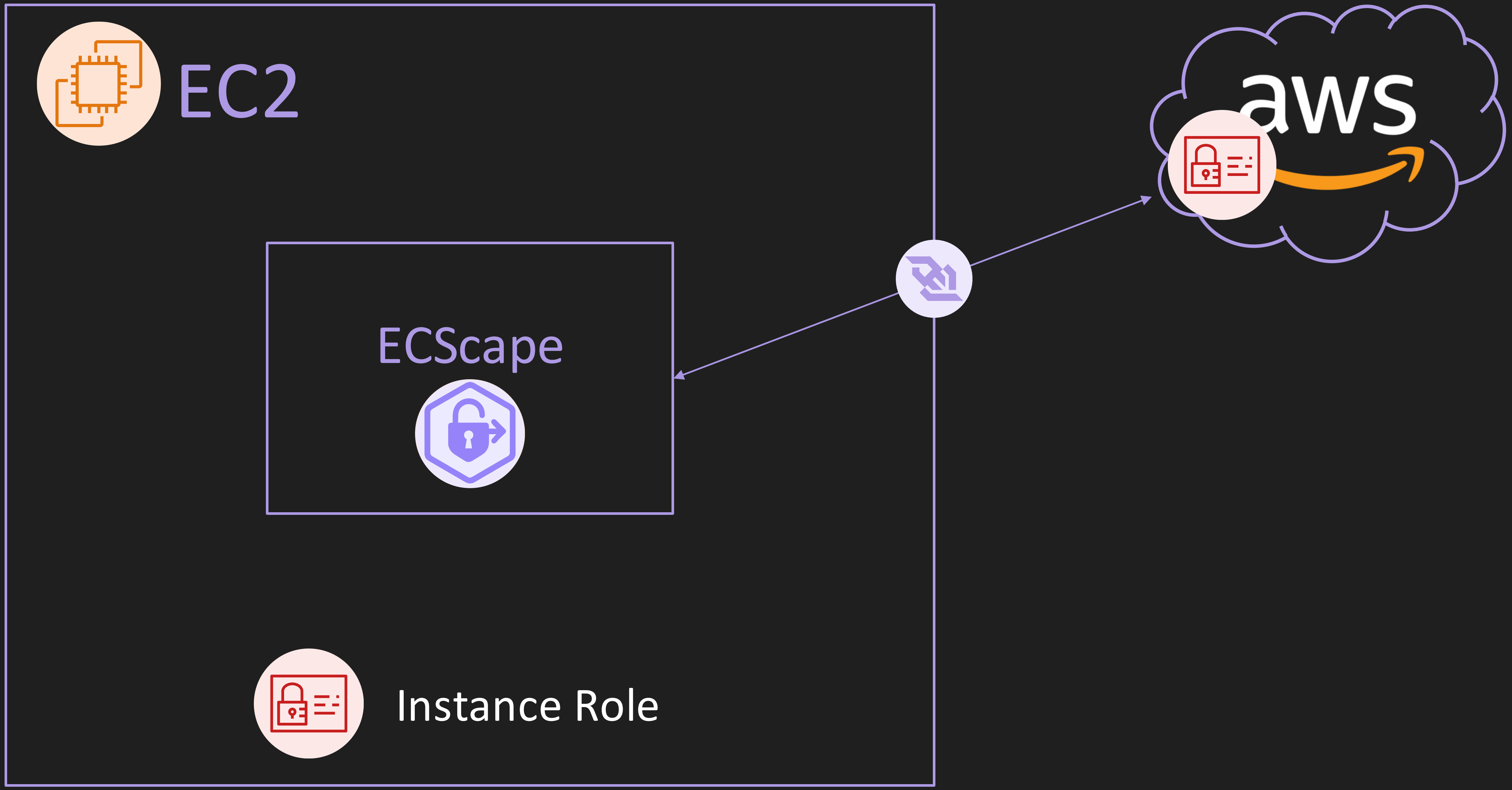
ECScape

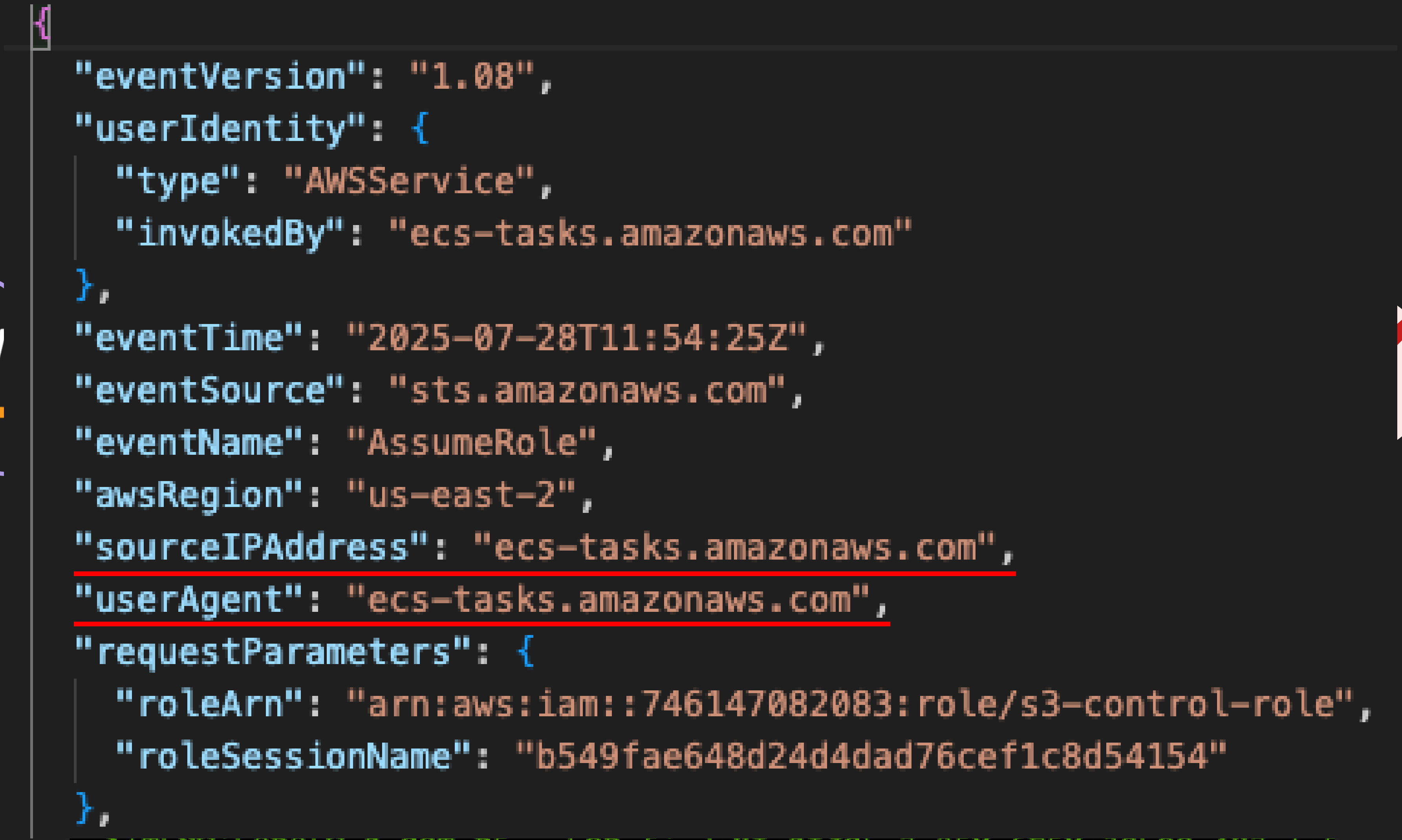
Instance Role


EC2
ECScape
Instance Role
aws
ecs:Poll?

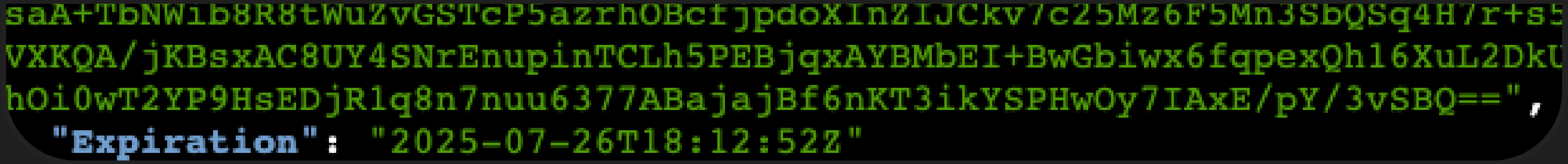


EC2
ECScape
ecs:Poll
Instance Role
aws


EC2
ECScape
Instance Role
aws

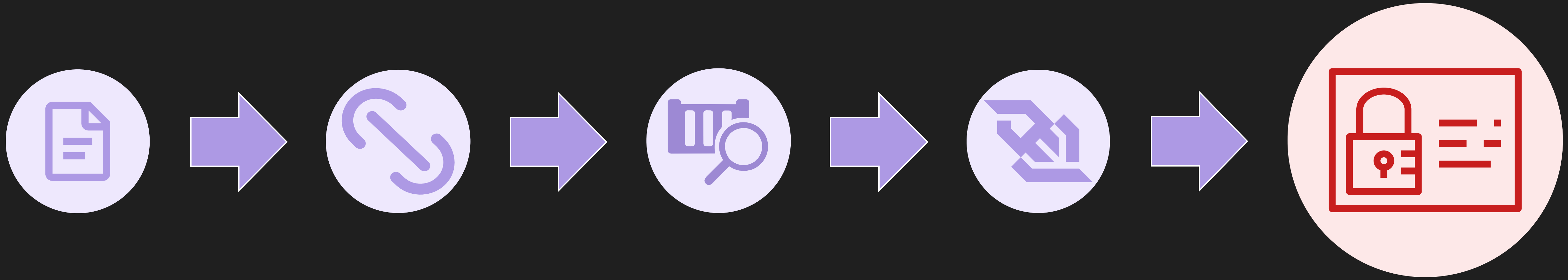


```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AWSService",
    "invokedBy": "ecs-tasks.amazonaws.com"
  },
  "eventTime": "2025-07-28T11:54:25Z",
  "eventSource": "sts.amazonaws.com",
  "eventName": "AssumeRole",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "ecs-tasks.amazonaws.com",
  "userAgent": "ecs-tasks.amazonaws.com",
  "requestParameters": {
    "roleArn": "arn:aws:iam::746147082083:role/s3-control-role",
    "roleSessionName": "b549fae648d24d4dad76cef1c8d54154"
  },
```



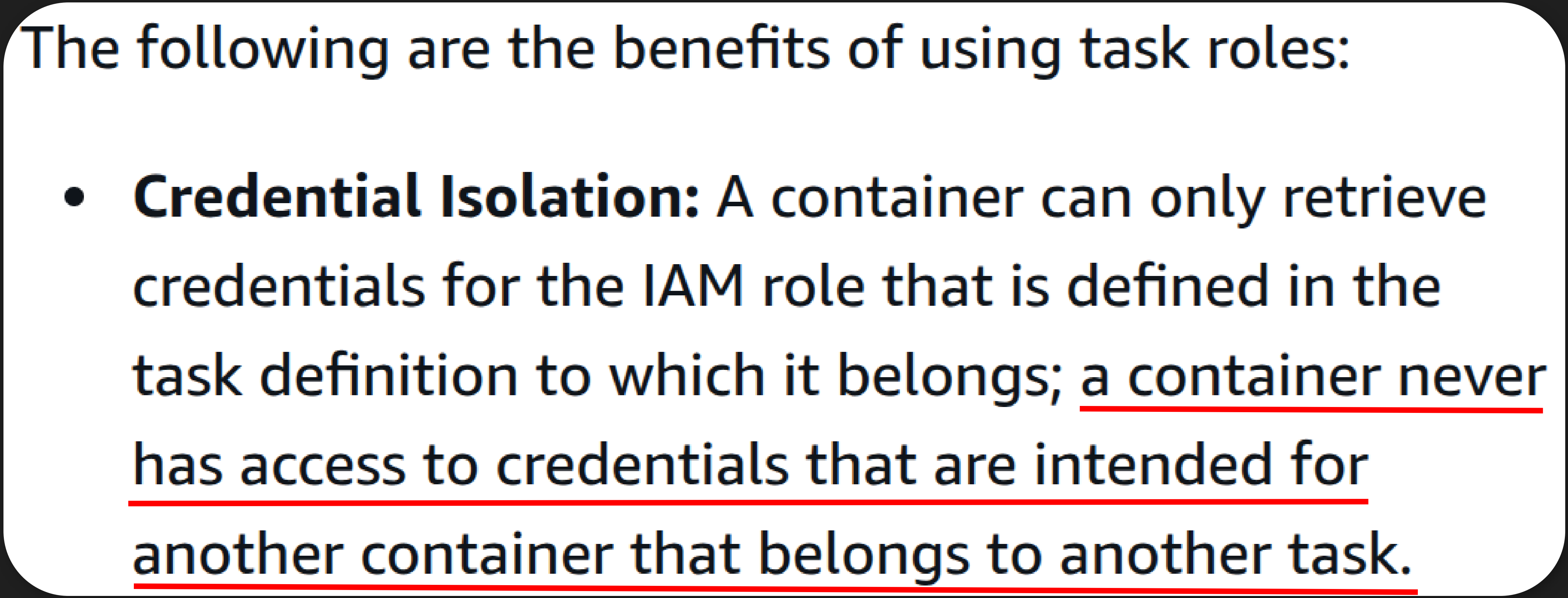
```
saA+TbNWib8R8tWuZvGSTcP5azrhOBcfjpdoXInZIJCkv7c25Mz6F5Mn3SbQSq4H7r+s5
VXKQA/jKBsxAC8UY4SNrEnupinTCLh5PEBjqxAYBMbEI+BwGbiwx6fqpexQh16XuL2DkU
hOi0wT2YP9HsEDjR1q8n7nuu6377ABajajBf6nKT3ikYSPHwOy7IAxE/pY/3vSBQ==",
  "Expiration": "2025-07-26T18:12:52Z"
```

ECScape - Final Flow



ECScape - Final Flow

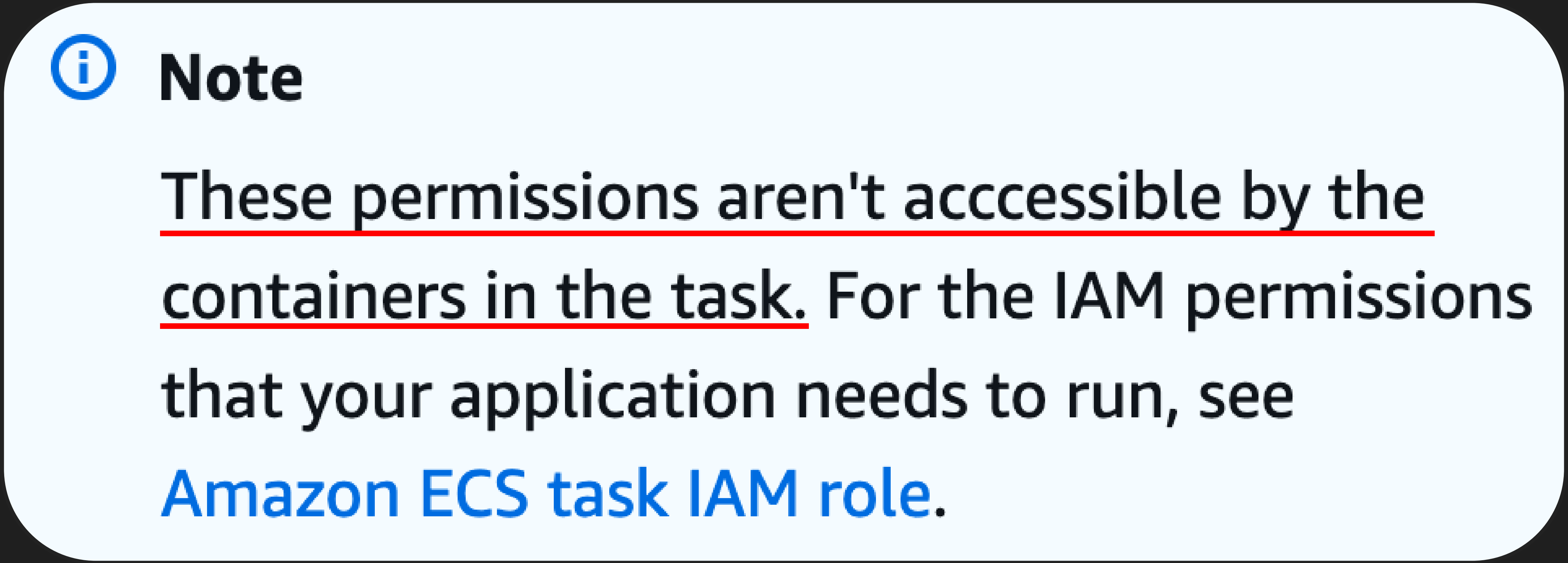
AWS Documentation



The following are the benefits of using task roles:

- **Credential Isolation:** A container can only retrieve credentials for the IAM role that is defined in the task definition to which it belongs; a container never has access to credentials that are intended for another container that belongs to another task.

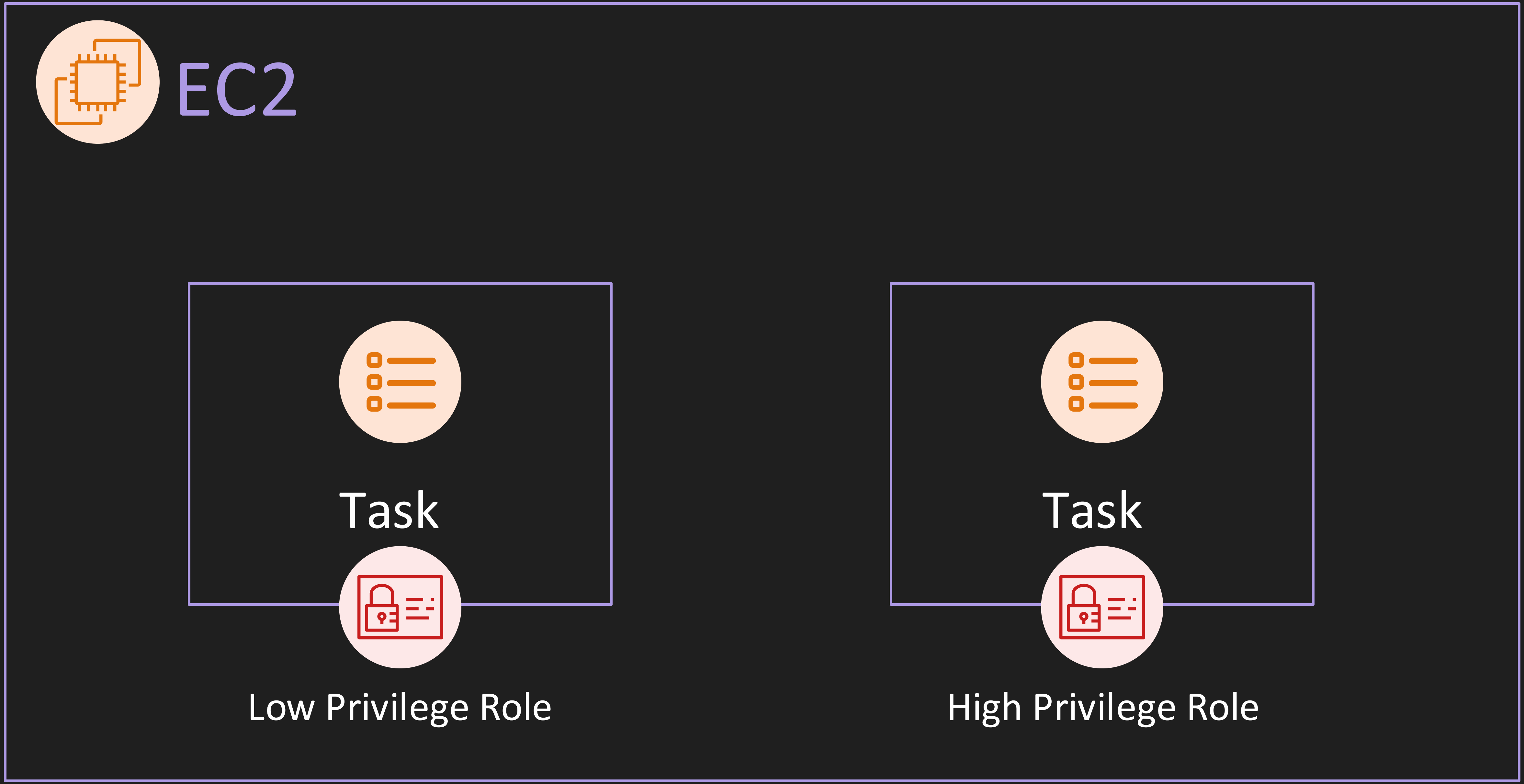
AWS Documentation

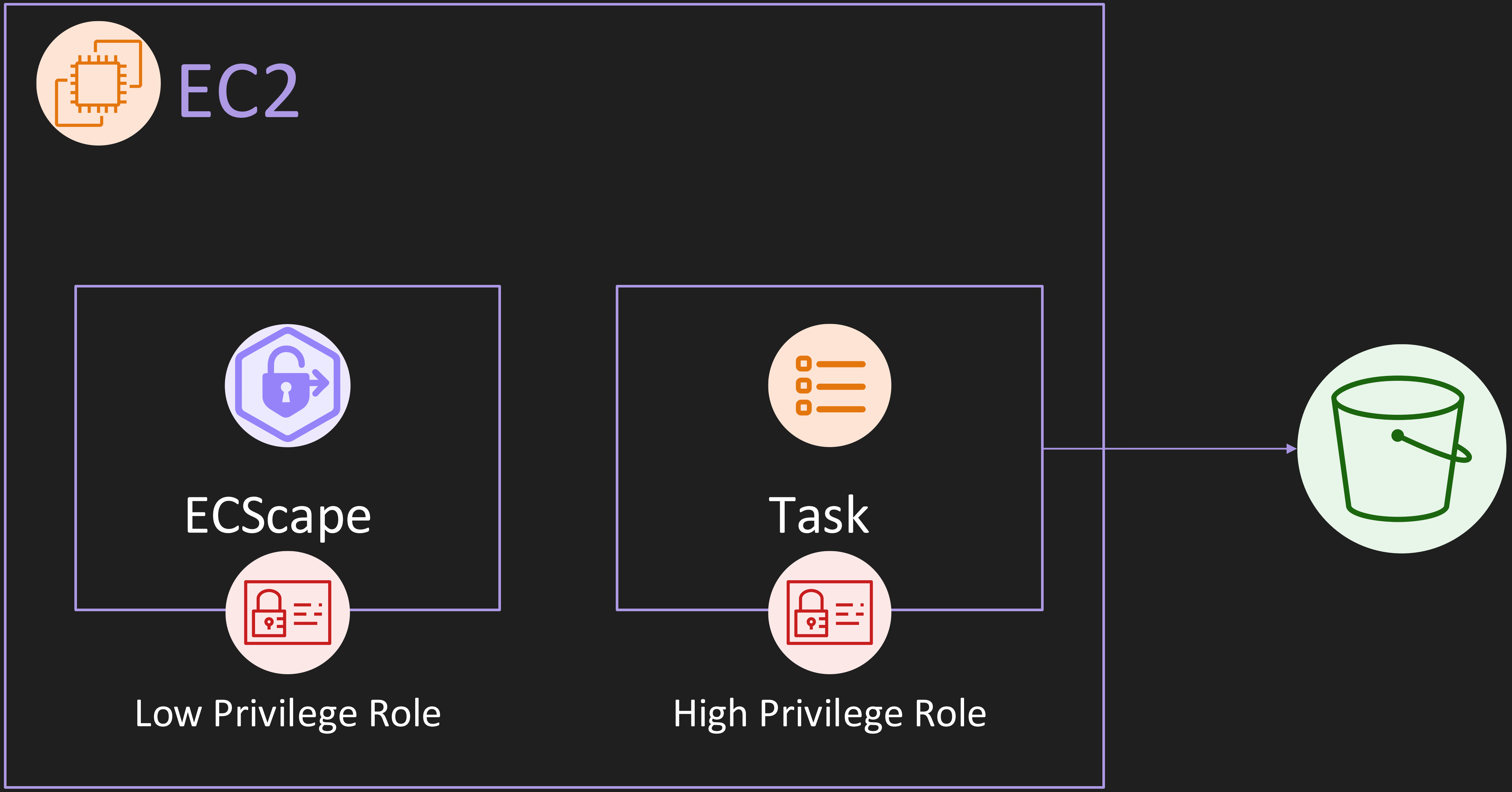


ⓘ **Note**

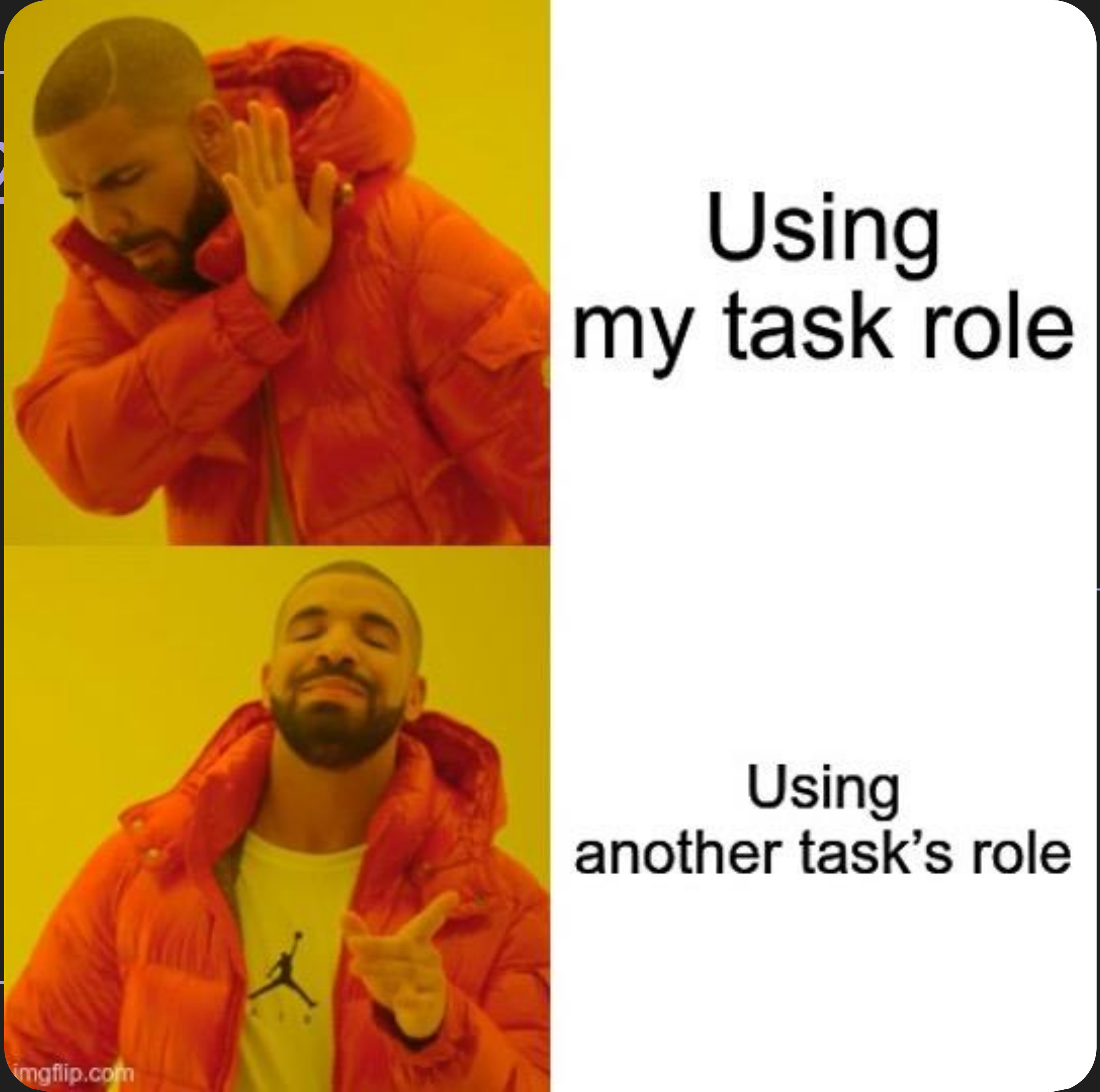
These permissions aren't acccessible by the containers in the task. For the IAM permissions that your application needs to run, see Amazon ECS task IAM role.

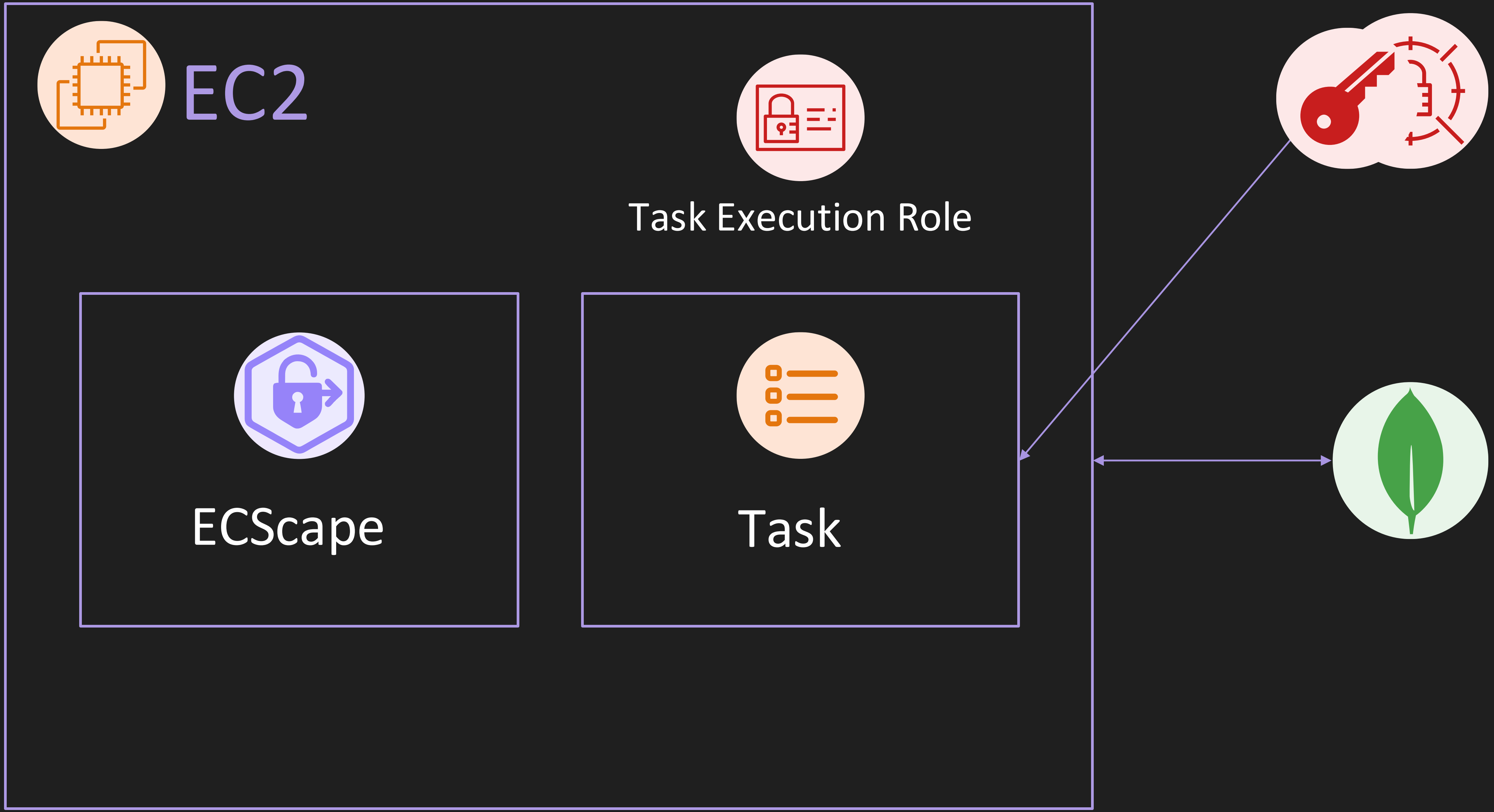
04
Impact


EC2
Task
Low Privilege Role
Task
High Privilege Role


EC2
ECScape
Task
Low Privilege Role
High Privilege Role

EC2



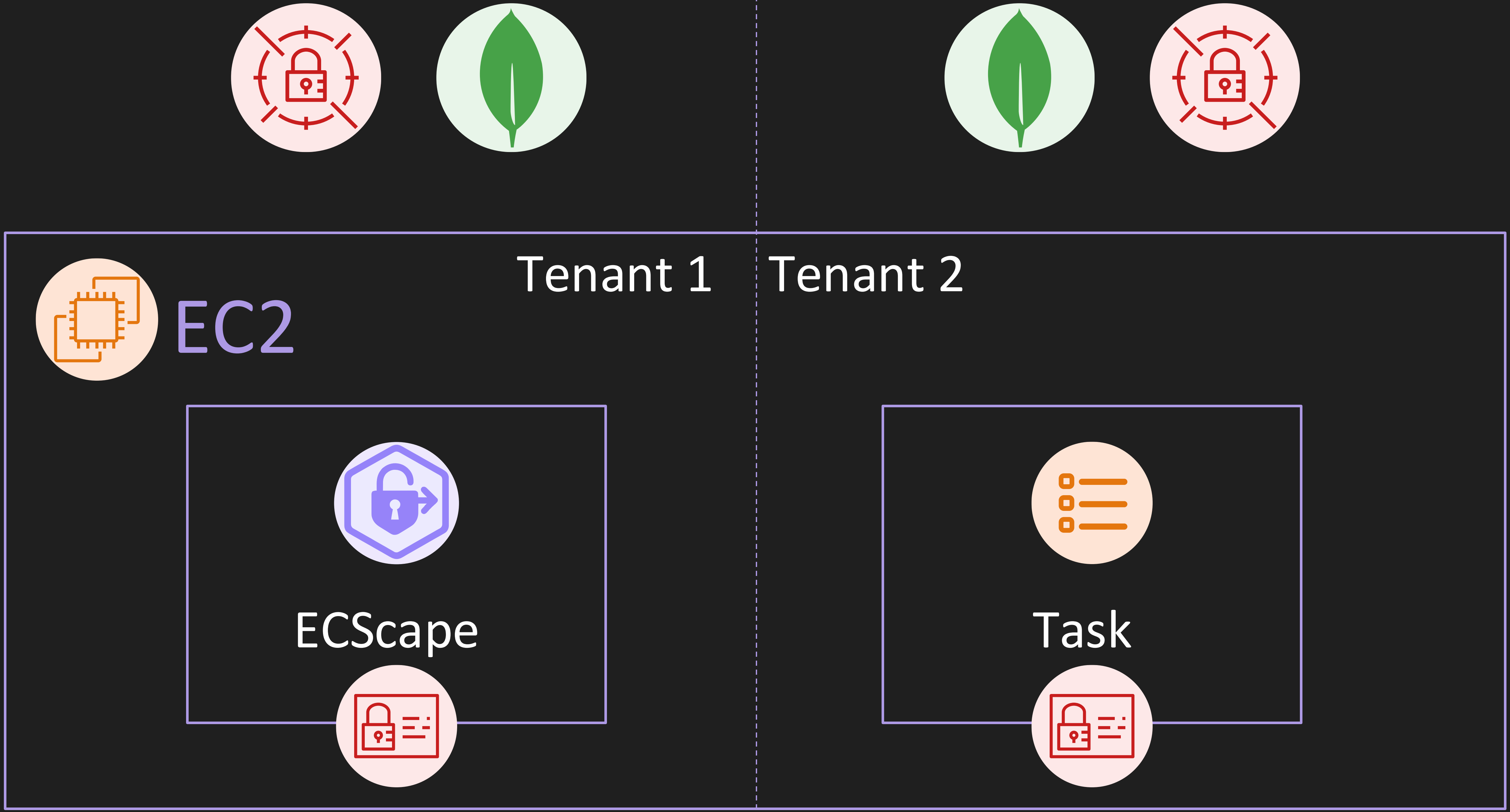

EC2
Task Execution Role
ECScape
Task

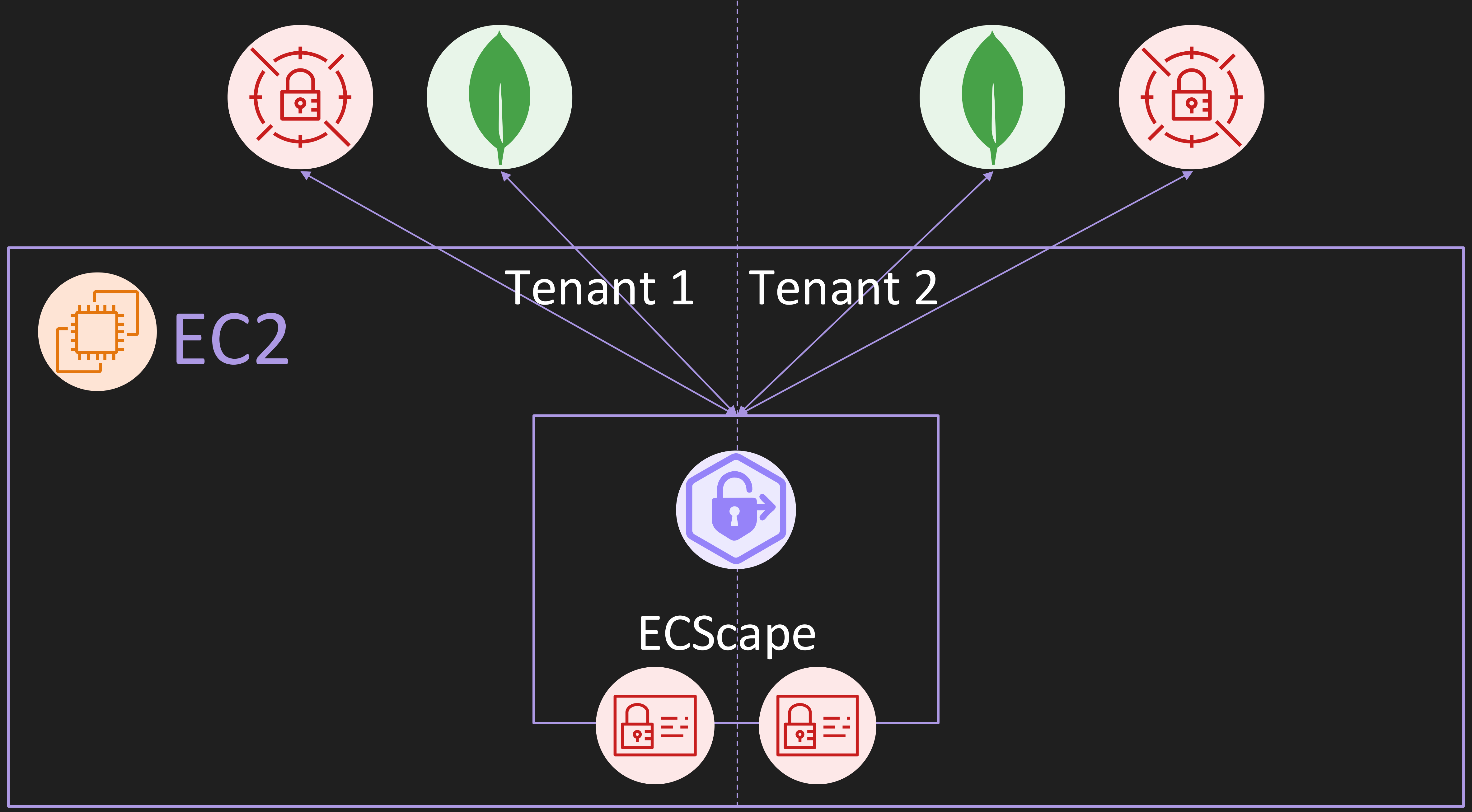
EC2

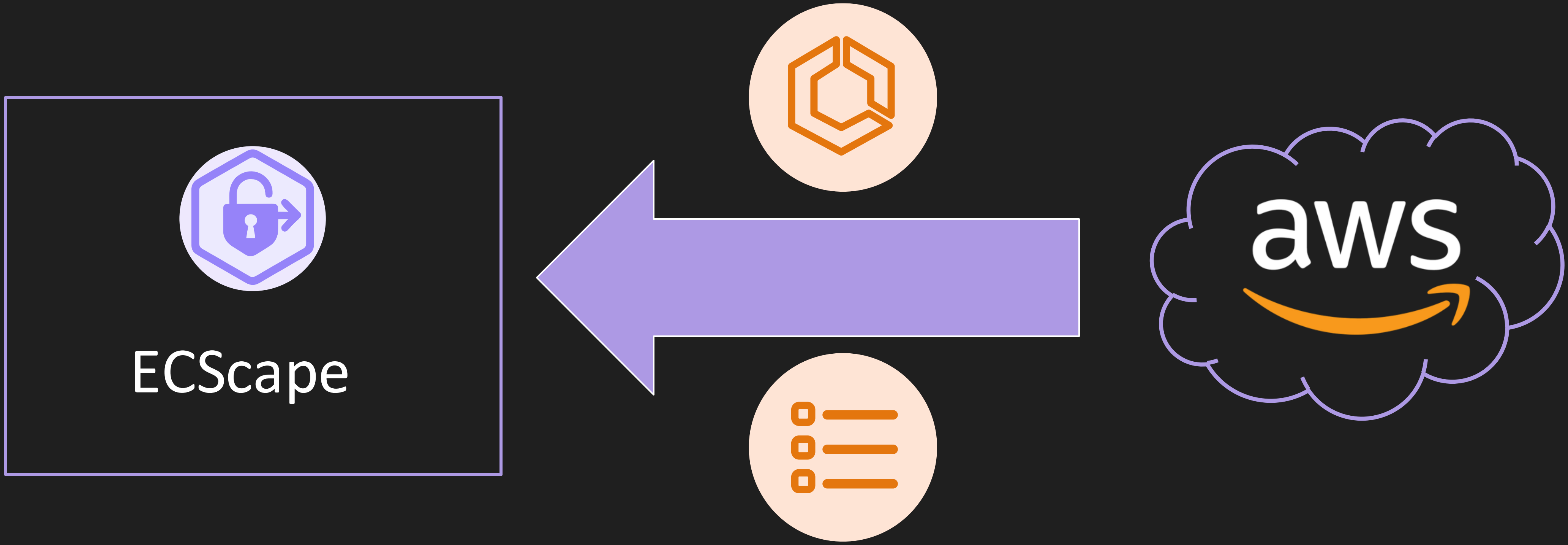
MY TASK ROLE CREDENTIALS
ANOTHER TASK'S ROLE CREDENTIALS
MY TASK EXECUTION ROLE CREDENTIALS
ANOTHER TASK'S EXECUTION ROLE CREDENTIALS

E

Task


Tenant 1
Tenant 2
EC2
ECScape
Task


Tenant 1
Tenant 2
EC2
ECScape

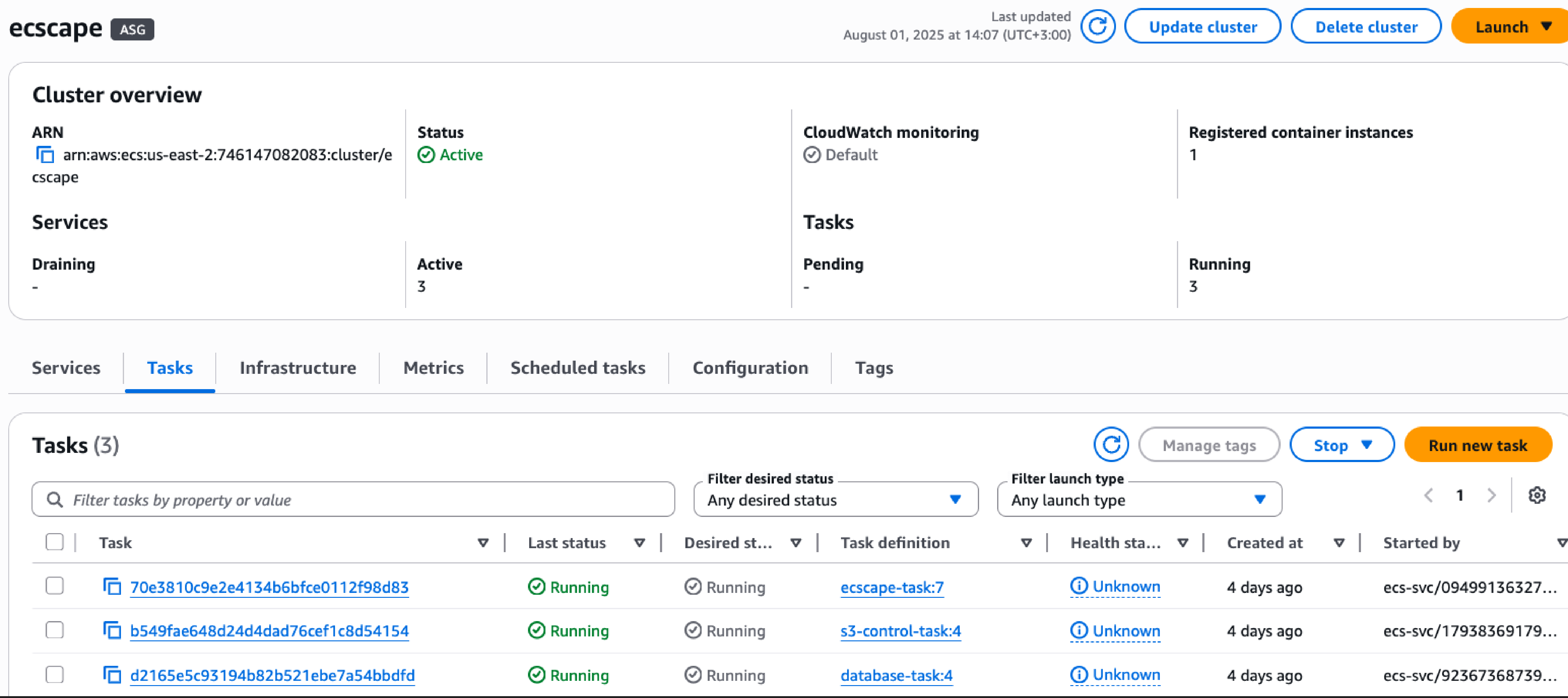

ECScape
aws

Impact

- Cross-Task IAM Role Hijacking
- Abuse of Task Execution Role
- Access to ECS Internals
- No Misconfiguration Needed - IMDS & Instance Role

05
Demo

Demo



ecscape ASG

Last updated
August 01, 2025 at 14:07 (UTC+3:00)

Cluster overview

ARN
arn:aws:ecs:us-east-2:746147082083:cluster/ecscape

Status
✓ Active

CloudWatch monitoring
✓ Default

Registered container instances
1

Services

Draining
-

Active
3

Tasks

Pending
-

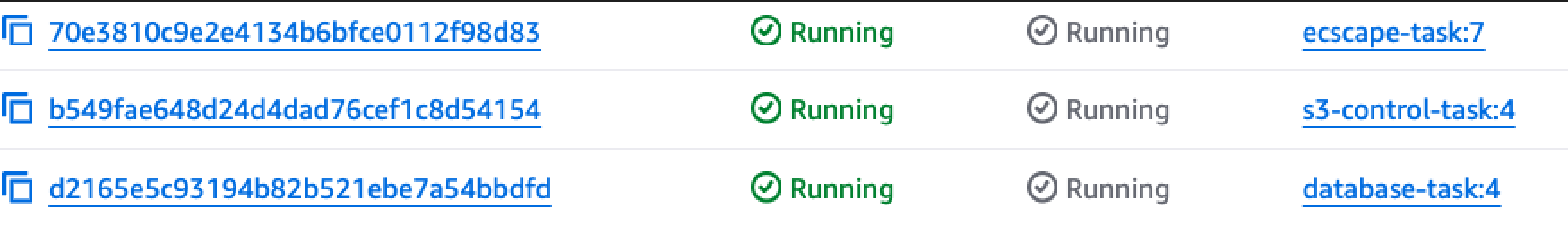
Running
3

Tasks (3)

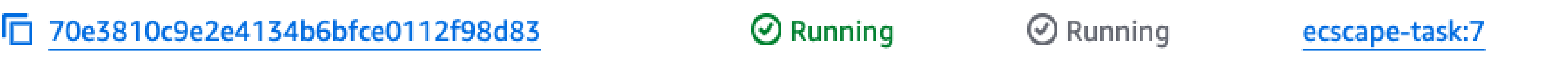
Filter desired status: Any desired status

Filter launch type: Any launch type

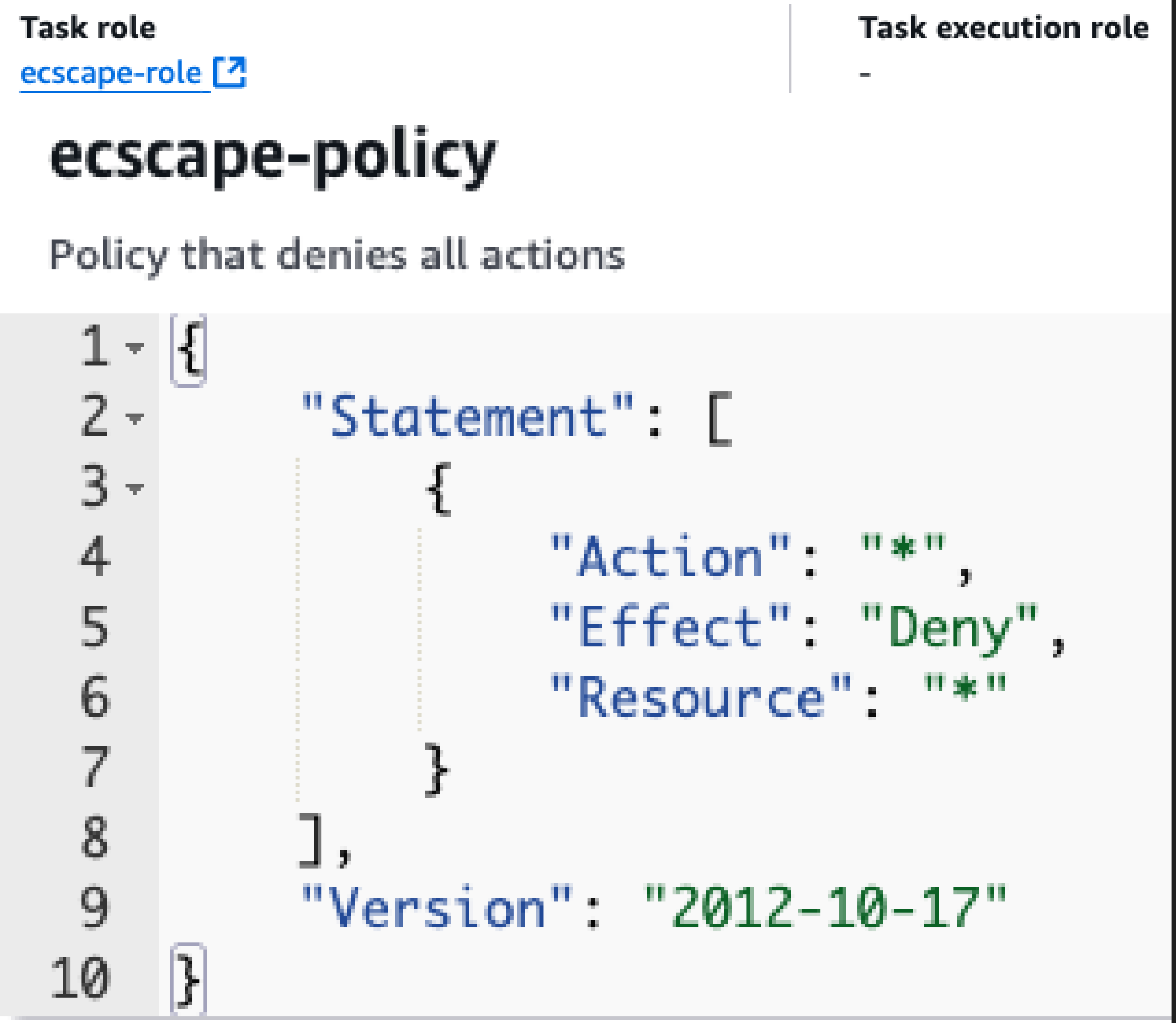
□	Task	Last status	Desired st...	Task definition	Health sta...	Created at	Started by
□	70e3810c9e2e4134b6bfce0112f98d83	✓ Running	✓ Running	ecscape-task:7	ⓘ Unknown	4 days ago	ecs-svc/09499136327...
□	b549fae648d24d4dad76cef1c8d54154	✓ Running	✓ Running	s3-control-task:4	ⓘ Unknown	4 days ago	ecs-svc/17938369179...
□	d2165e5c93194b82b521ebe7a54bbdfd	✓ Running	✓ Running	database-task:4	ⓘ Unknown	4 days ago	ecs-svc/92367368739...



70e3810c9e2e4134b6bfce0112f98d83	✓ Running	✓ Running	ecscape-task:7
b549fae648d24d4dad76cef1c8d54154	✓ Running	✓ Running	s3-control-task:4
d2165e5c93194b82b521ebe7a54bbdfd	✓ Running	✓ Running	database-task:4



70e3810c9e2e4134b6bfce0112f98d83 Running Running ecscape-task:7



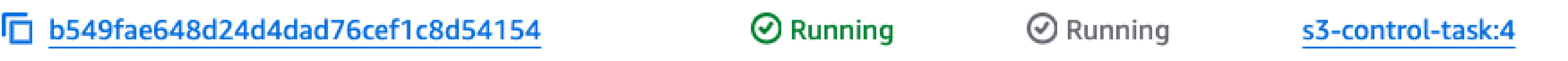
Task role
ecscape-role

Task execution role
-

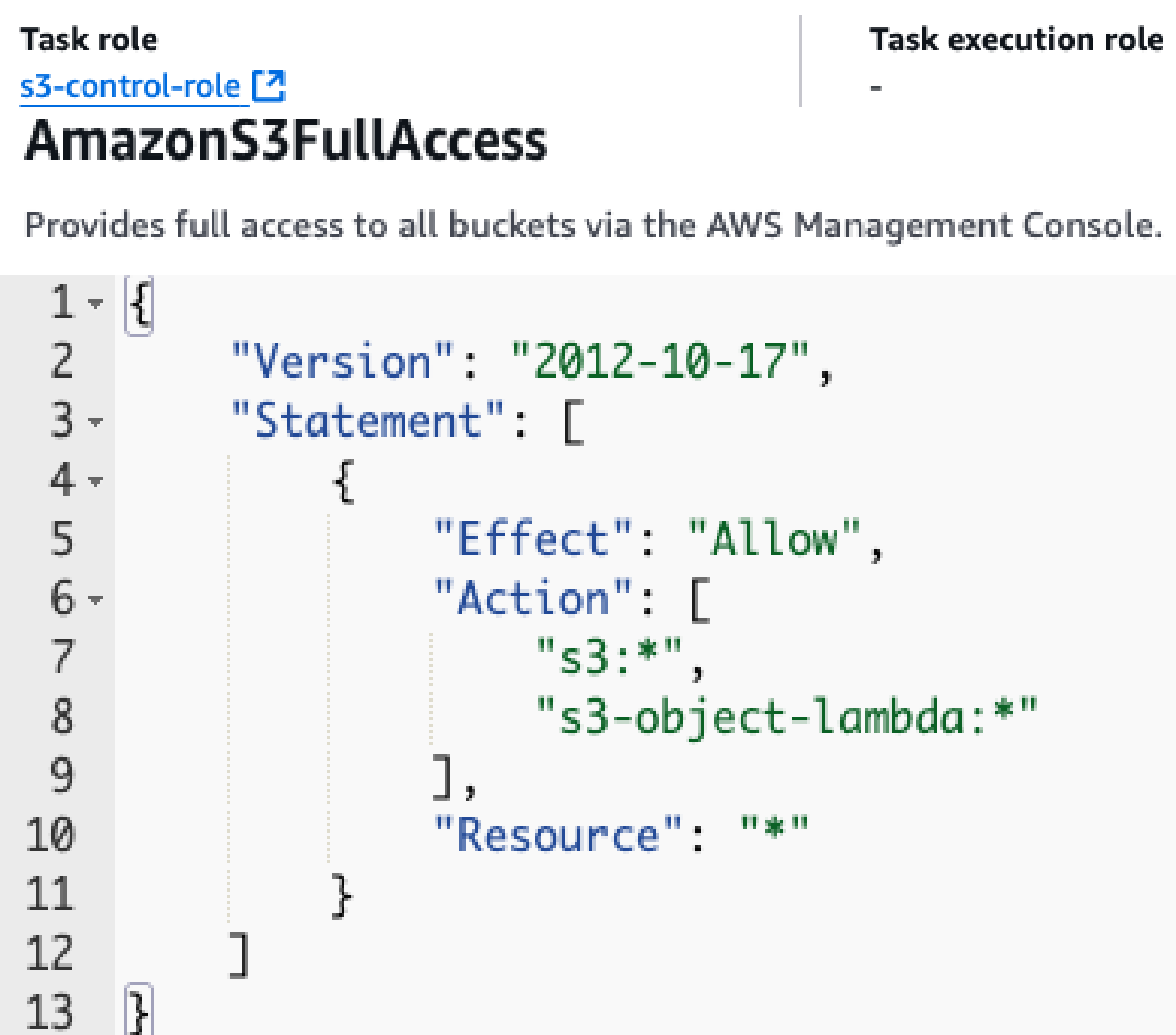
ecscape-policy

Policy that denies all actions

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Deny",
            "Resource": "*"
        }
    ],
    "Version": "2012-10-17"
}
```



b549fae648d24d4dad76cef1c8d54154 Running Running s3-control-task:4



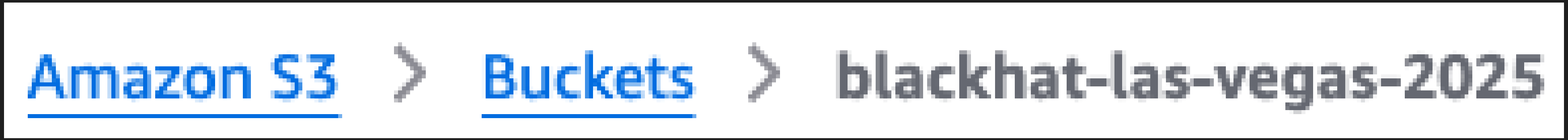
Task role
s3-control-role

Task execution role
-

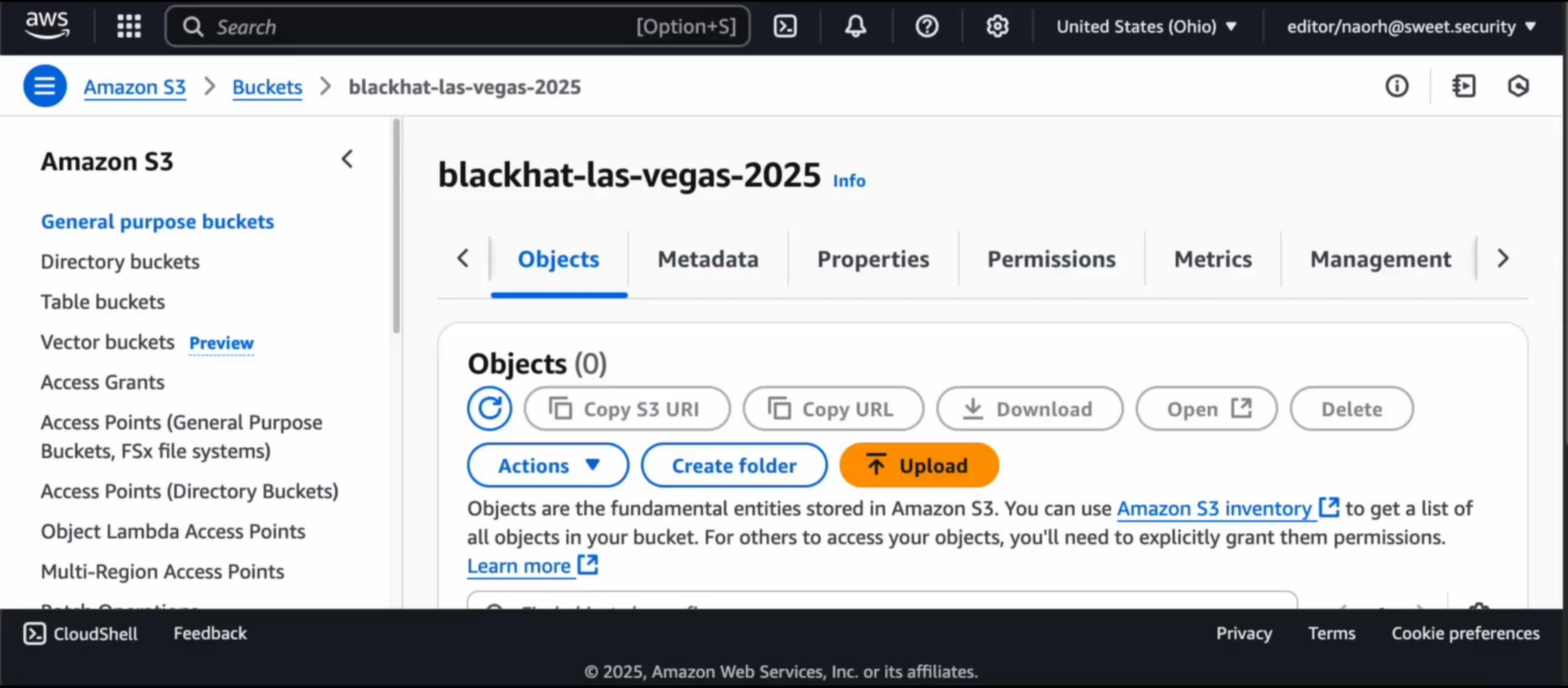
AmazonS3FullAccess

Provides full access to all buckets via the AWS Management Console.

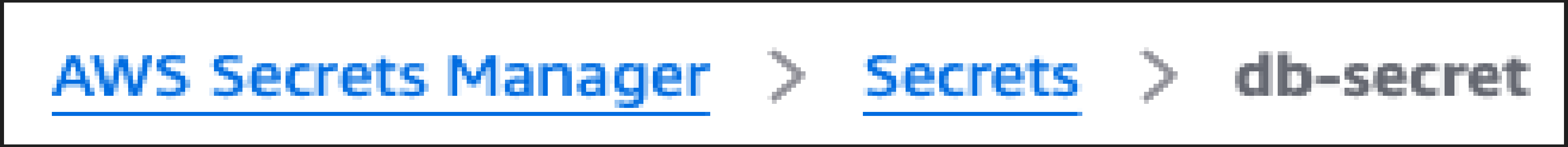
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:*",
                "s3-object-lambda:*"
            ],
            "Resource": "*"
        }
    ]
}
```



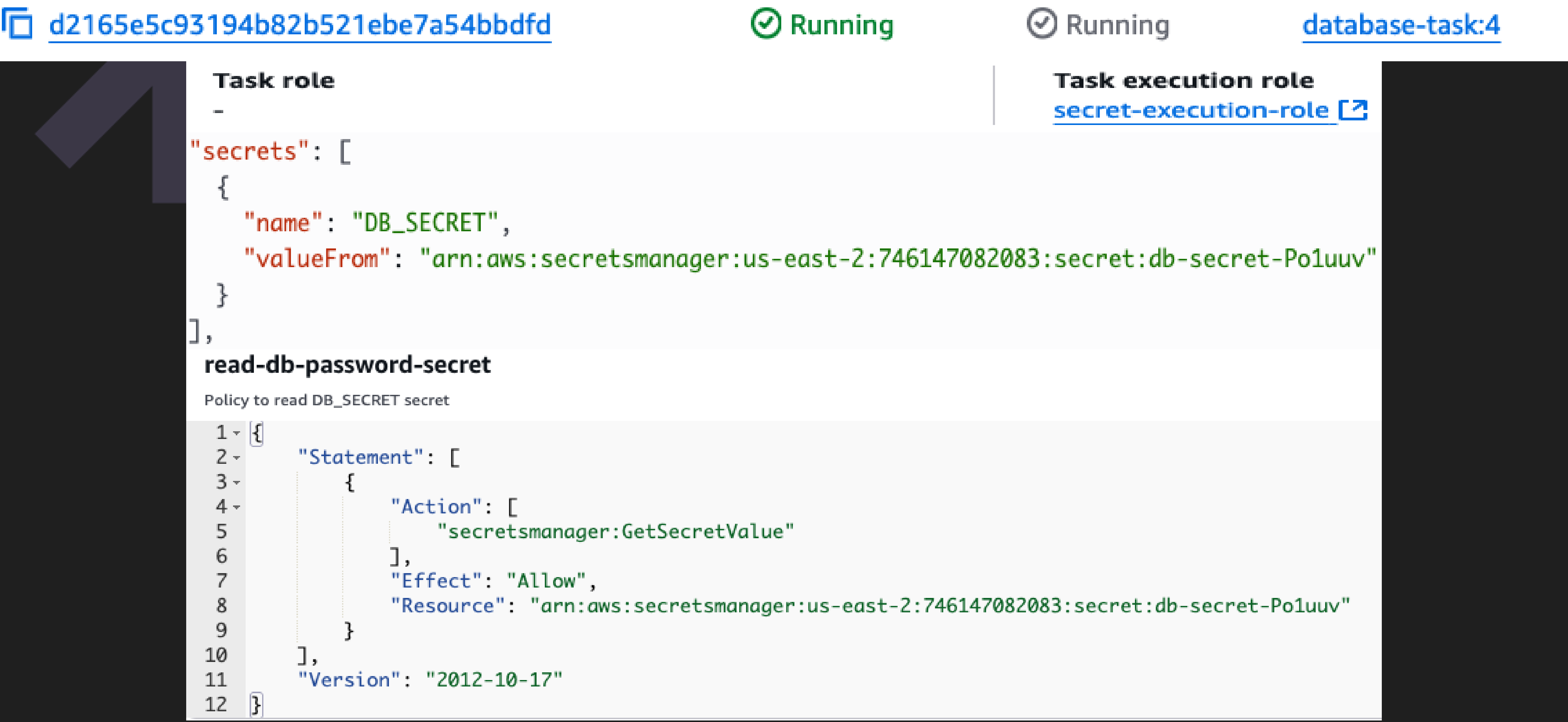
Amazon S3 > Buckets > blackhat-las-vegas-2025



Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more



AWS Secrets Manager > Secrets > db-secret



d2165e5c93194b82b521ebe7a54bbdfd ✓ Running ✓ Running database-task:4

Task role

-

Task execution role

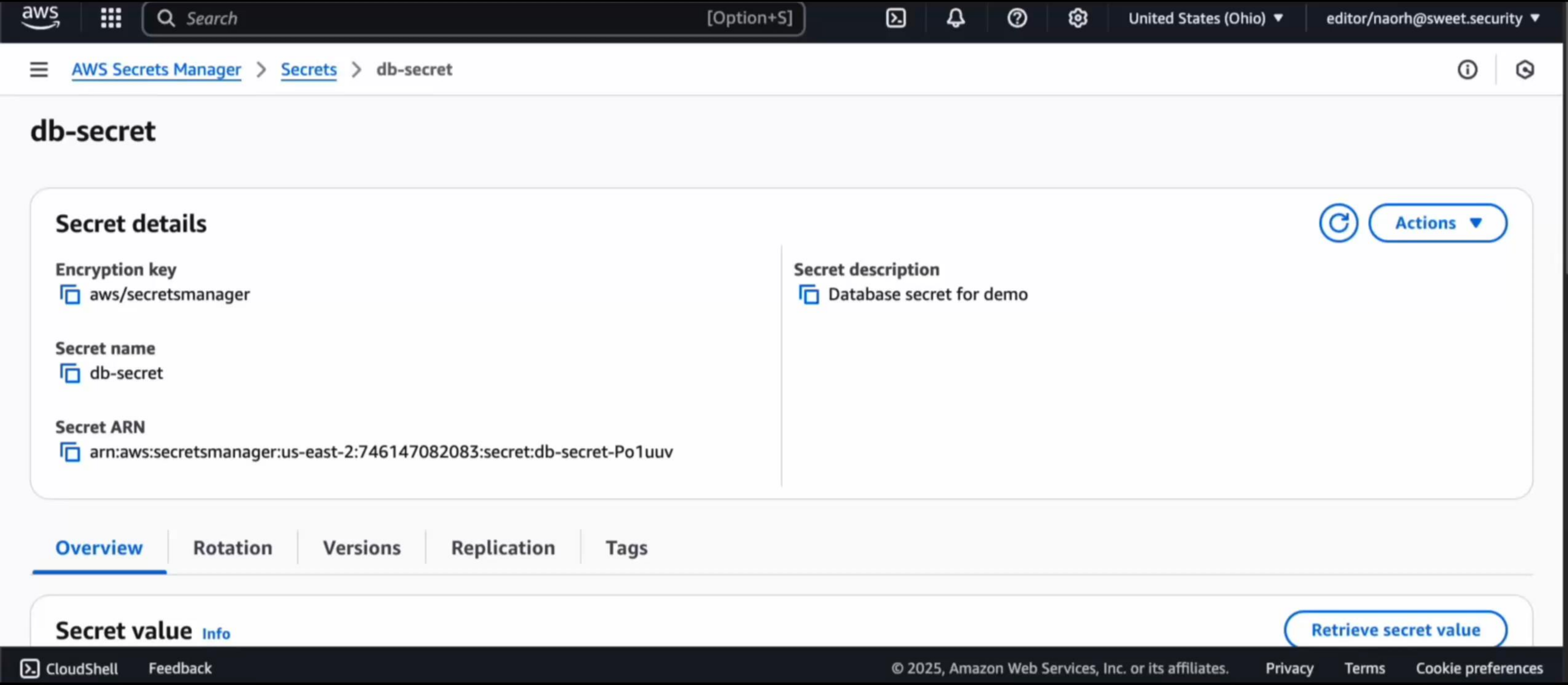
secret-execution-role

```
"secrets": [
  {
    "name": "DB_SECRET",
    "valueFrom": "arn:aws:secretsmanager:us-east-2:746147082083:secret:db-secret-Po1uuv"
  }
],
```

read-db-password-secret

Policy to read DB_SECRET secret

```
{
    "Statement": [
        {
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:secretsmanager:us-east-2:746147082083:secret:db-secret-Po1uuv"
        }
    ],
    "Version": "2012-10-17"
}
```



db-secret

Secret details

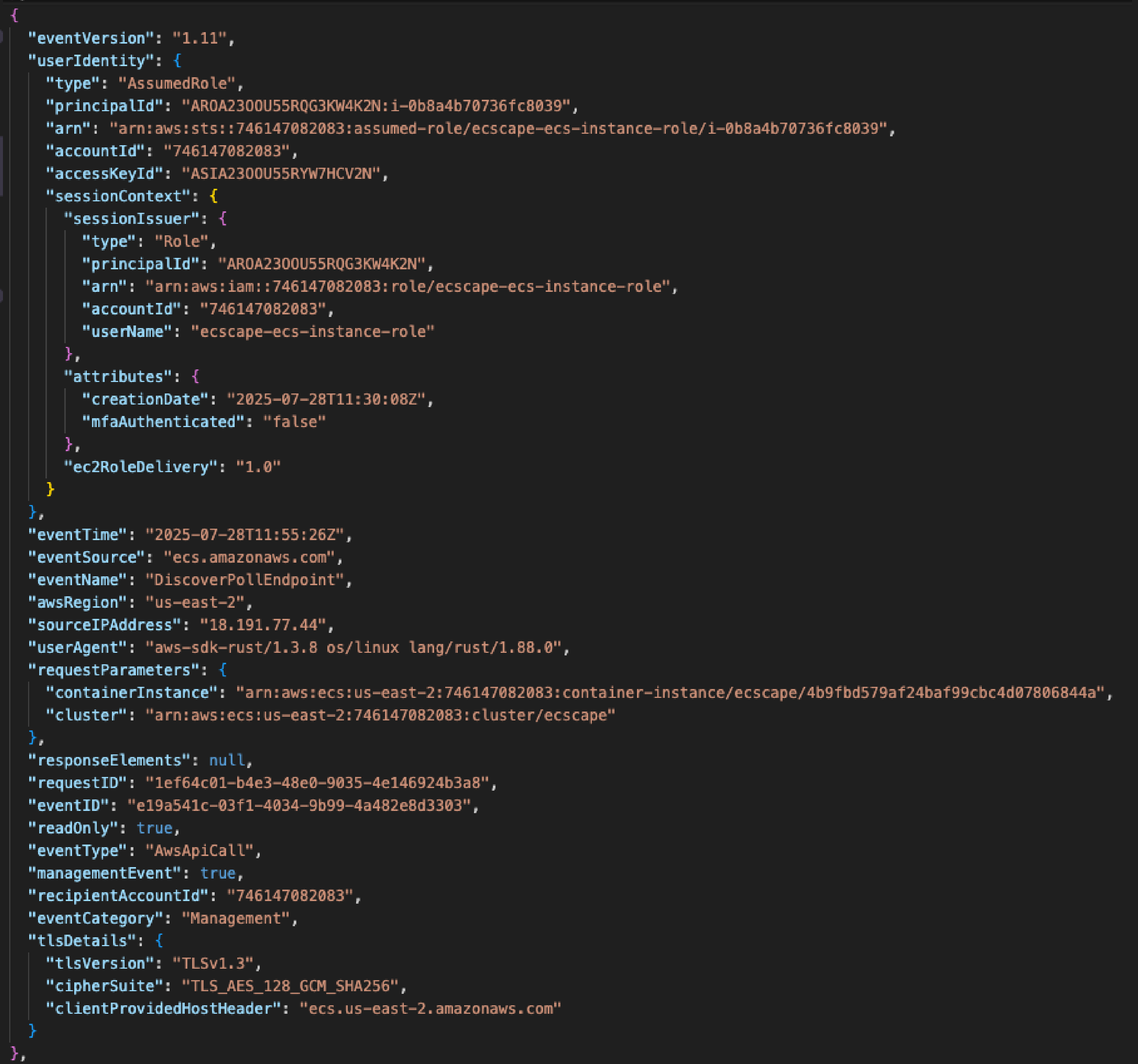
Encryption key
aws/secretsmanager

Secret name
db-secret

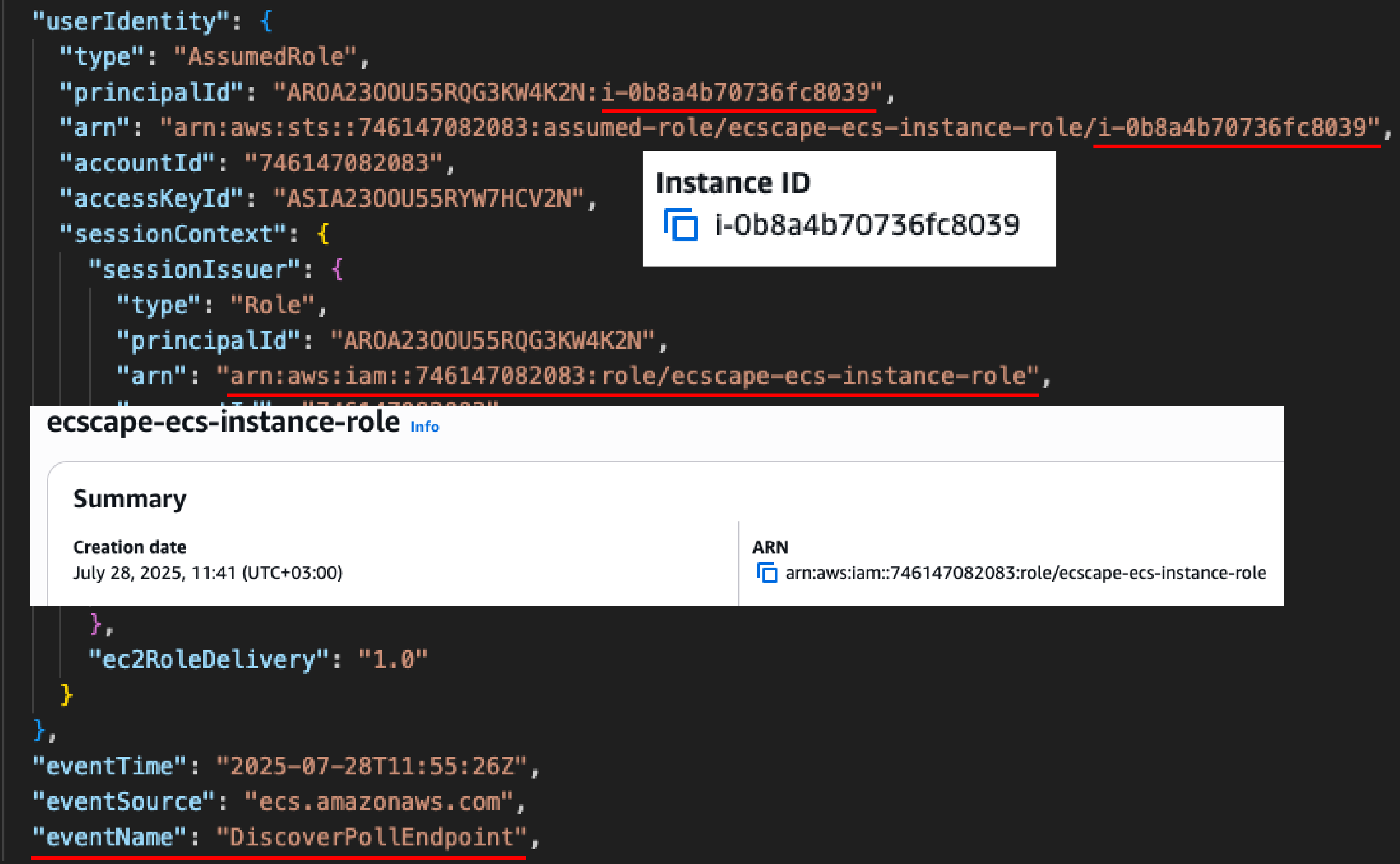
Secret ARN
arn:aws:secretsmanager:us-east-2:746147082083:secret:db-secret-Po1uuv

Secret description
Database secret for demo

Secret value



```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROA23OOU55RQG3KW4K2N:i-0b8a4b70736fc8039",
    "arn": "arn:aws:sts::746147082083:assumed-role/ecscape-ecs-instance-role/i-0b8a4b70736fc8039",
    "accountId": "746147082083",
    "accessKeyId": "ASIA23OOU55RYW7HCV2N",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROA23OOU55RQG3KW4K2N",
        "arn": "arn:aws:iam::746147082083:role/ecscape-ecs-instance-role",
        "accountId": "746147082083",
        "userName": "ecscape-ecs-instance-role"
      },
      "attributes": {
        "creationDate": "2025-07-28T11:30:08Z",
        "mfaAuthenticated": "false"
      },
      "ec2RoleDelivery": "1.0"
    }
  },
  "eventTime": "2025-07-28T11:55:26Z",
  "eventSource": "ecs.amazonaws.com",
  "eventName": "DiscoverPollEndpoint",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "18.191.77.44",
  "userAgent": "aws-sdk-rust/1.3.8 os/linux lang/rust/1.88.0",
  "requestParameters": {
    "containerInstance": "arn:aws:ecs:us-east-2:746147082083:container-instance/ecscape/4b9fbd579af24baf99cbc4d07806844a",
    "cluster": "arn:aws:ecs:us-east-2:746147082083:cluster/ecscape"
  },
  "responseElements": null,
  "requestID": "1ef64c01-b4e3-48e0-9035-4e146924b3a8",
  "eventID": "e19a541c-03f1-4034-9b99-4a482e8d3303",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "746147082083",
  "eventCategory": "Management",
  "tlsDetails": {
    "tlsVersion": "TLSv1.3",
    "cipherSuite": "TLS_AES_128_GCM_SHA256",
    "clientProvidedHostHeader": "ecs.us-east-2.amazonaws.com"
  }
},
```



```
"userIdentity": {
  "type": "AssumedRole",
  "principalId": "AROA23OOU55RQG3KW4K2N:i-0b8a4b70736fc8039",
  "arn": "arn:aws:sts::746147082083:assumed-role/ecscape-ecs-instance-role/i-0b8a4b70736fc8039",
  "accountId": "746147082083",
  "accessKeyId": "ASIA23OOU55RYW7HCV2N",
  "sessionContext": {
    "sessionIssuer": {
      "type": "Role",
      "principalId": "AROA23OOU55RQG3KW4K2N",
      "arn": "arn:aws:iam::746147082083:role/ecscape-ecs-instance-role",
```

Instance ID

i-0b8a4b70736fc8039

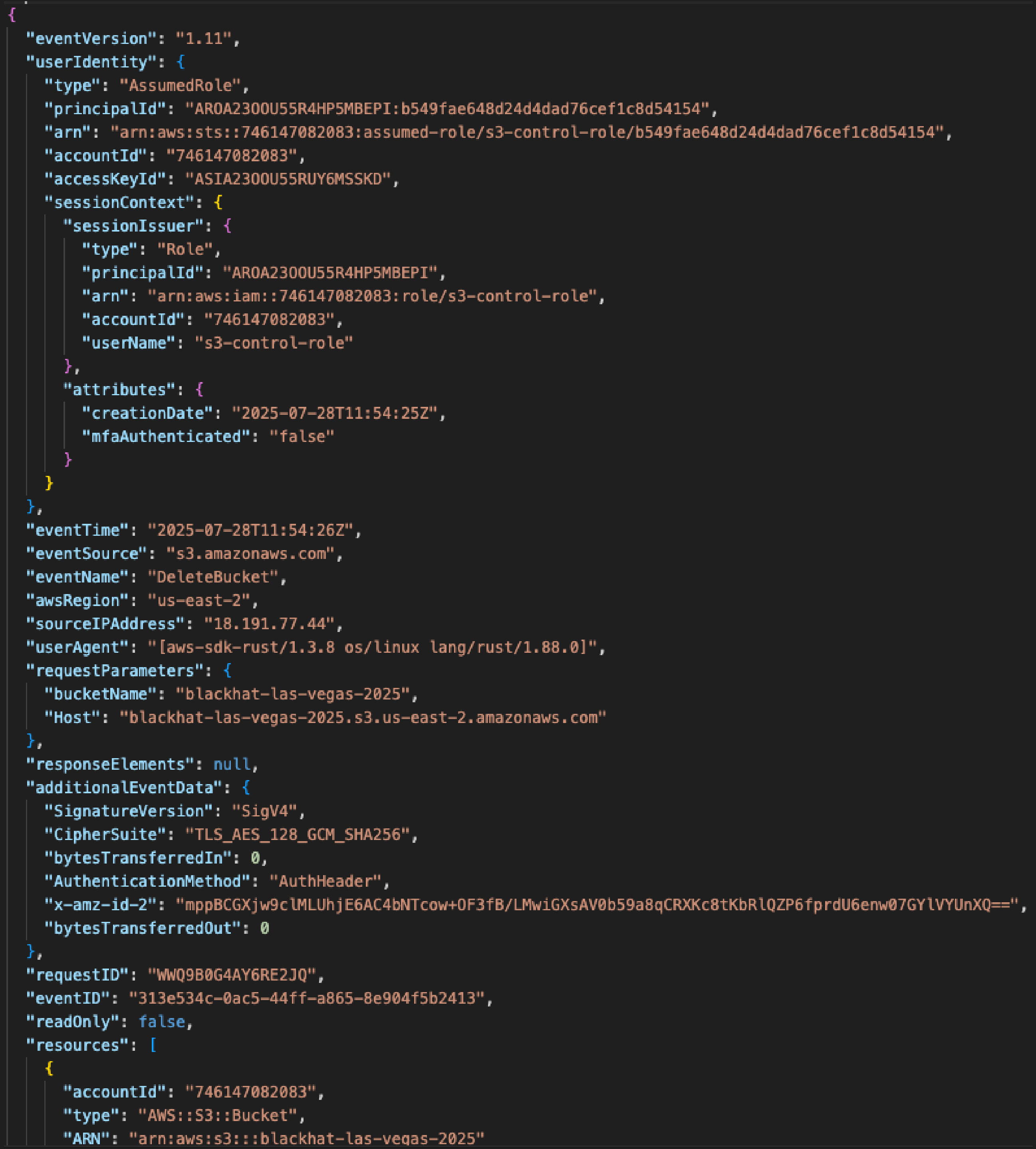
ecscape-ecs-instance-role Info

Summary

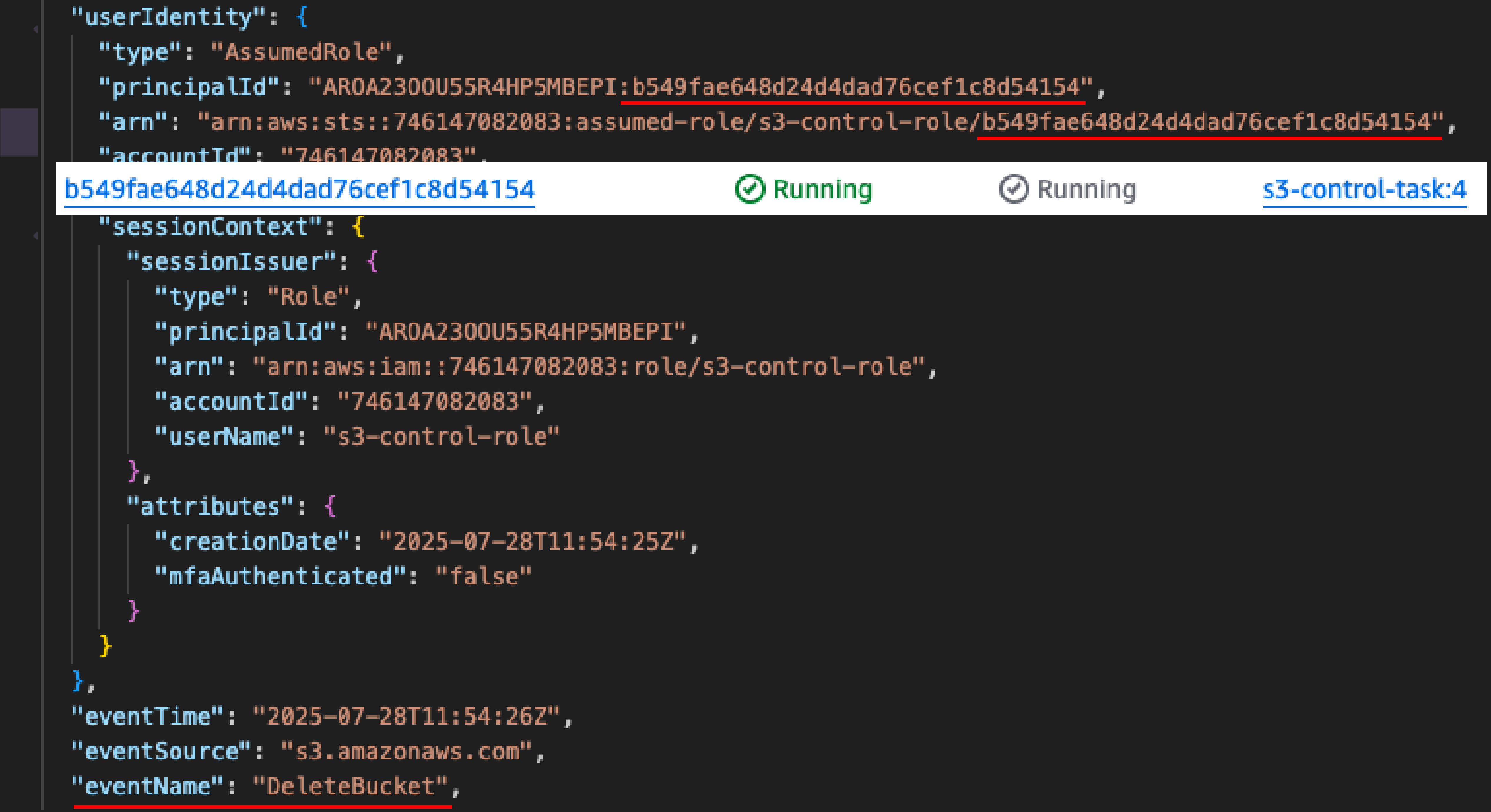
Creation date
July 28, 2025, 11:41 (UTC+03:00)

ARN
arn:aws:iam::746147082083:role/ecscape-ecs-instance-role

```
    },
    "ec2RoleDelivery": "1.0"
  }
},
"eventTime": "2025-07-28T11:55:26Z",
"eventSource": "ecs.amazonaws.com",
"eventName": "DiscoverPollEndpoint",
```



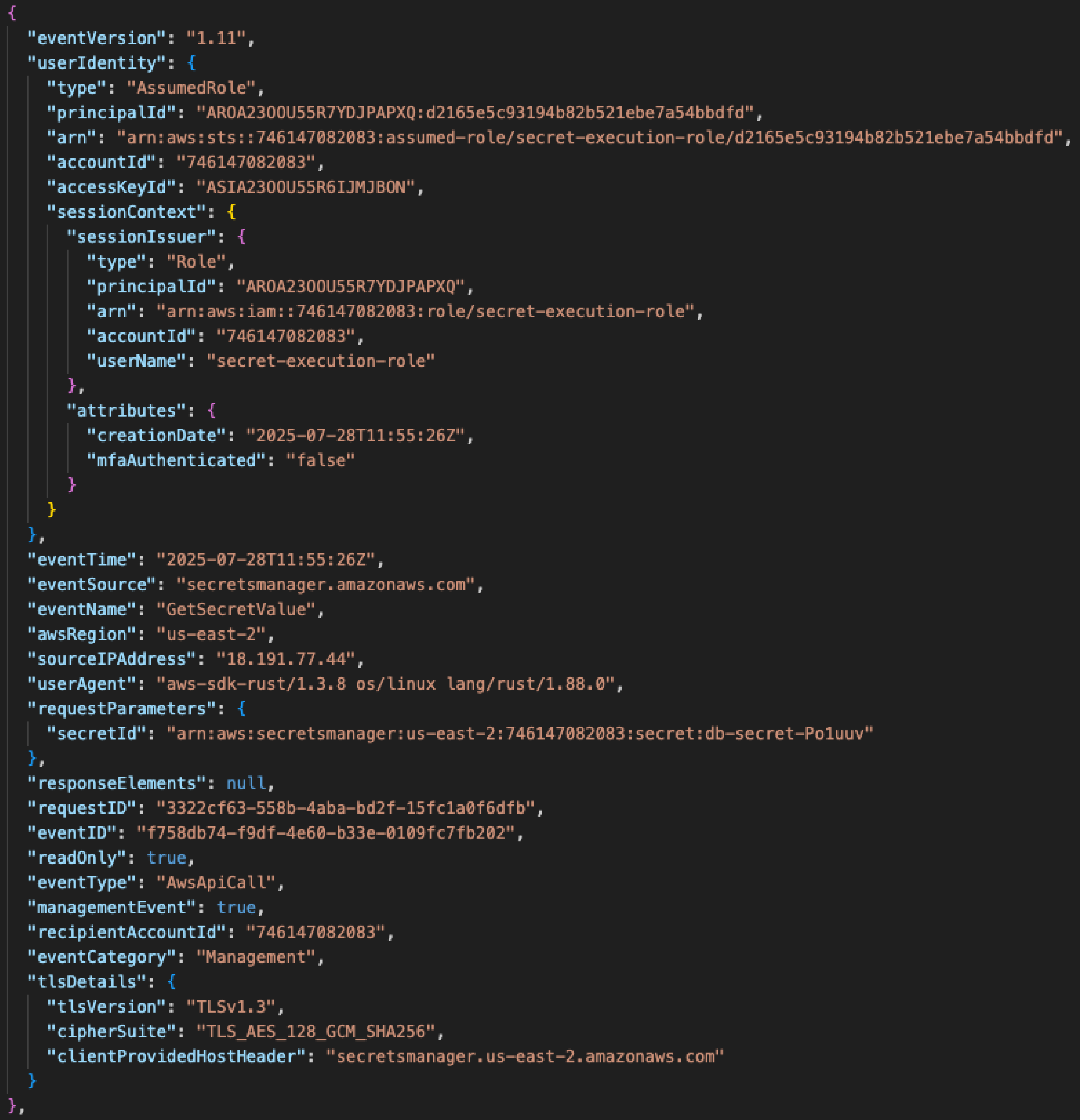
```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROA23OOU55R4HP5MBEPI:b549fae648d24d4dad76cef1c8d54154",
    "arn": "arn:aws:sts::746147082083:assumed-role/s3-control-role/b549fae648d24d4dad76cef1c8d54154",
    "accountId": "746147082083",
    "accessKeyId": "ASIA23OOU55RUY6MSSKD",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROA23OOU55R4HP5MBEPI",
        "arn": "arn:aws:iam::746147082083:role/s3-control-role",
        "accountId": "746147082083",
        "userName": "s3-control-role"
      },
      "attributes": {
        "creationDate": "2025-07-28T11:54:25Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2025-07-28T11:54:26Z",
  "eventSource": "s3.amazonaws.com",
  "eventName": "DeleteBucket",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "18.191.77.44",
  "userAgent": "[aws-sdk-rust/1.3.8 os/linux lang/rust/1.88.0]",
  "requestParameters": {
    "bucketName": "blackhat-las-vegas-2025",
    "Host": "blackhat-las-vegas-2025.s3.us-east-2.amazonaws.com"
  },
  "responseElements": null,
  "additionalEventData": {
    "SignatureVersion": "SigV4",
    "CipherSuite": "TLS_AES_128_GCM_SHA256",
    "bytesTransferredIn": 0,
    "AuthenticationMethod": "AuthHeader",
    "x-amz-id-2": "mppBCGXjw9clMLUhjE6AC4bNTcow+OF3fB/LMwiGXsAV0b59a8qCRXKc8tKbRlQZP6fprdU6enw07GYlVYUnXQ==",
    "bytesTransferredOut": 0
  },
  "requestID": "WWQ9B0G4AY6RE2JQ",
  "eventID": "313e534c-0ac5-44ff-a865-8e904f5b2413",
  "readOnly": false,
  "resources": [
    {
      "accountId": "746147082083",
      "type": "AWS::S3::Bucket",
      "ARN": "arn:aws:s3:::blackhat-las-vegas-2025"
```



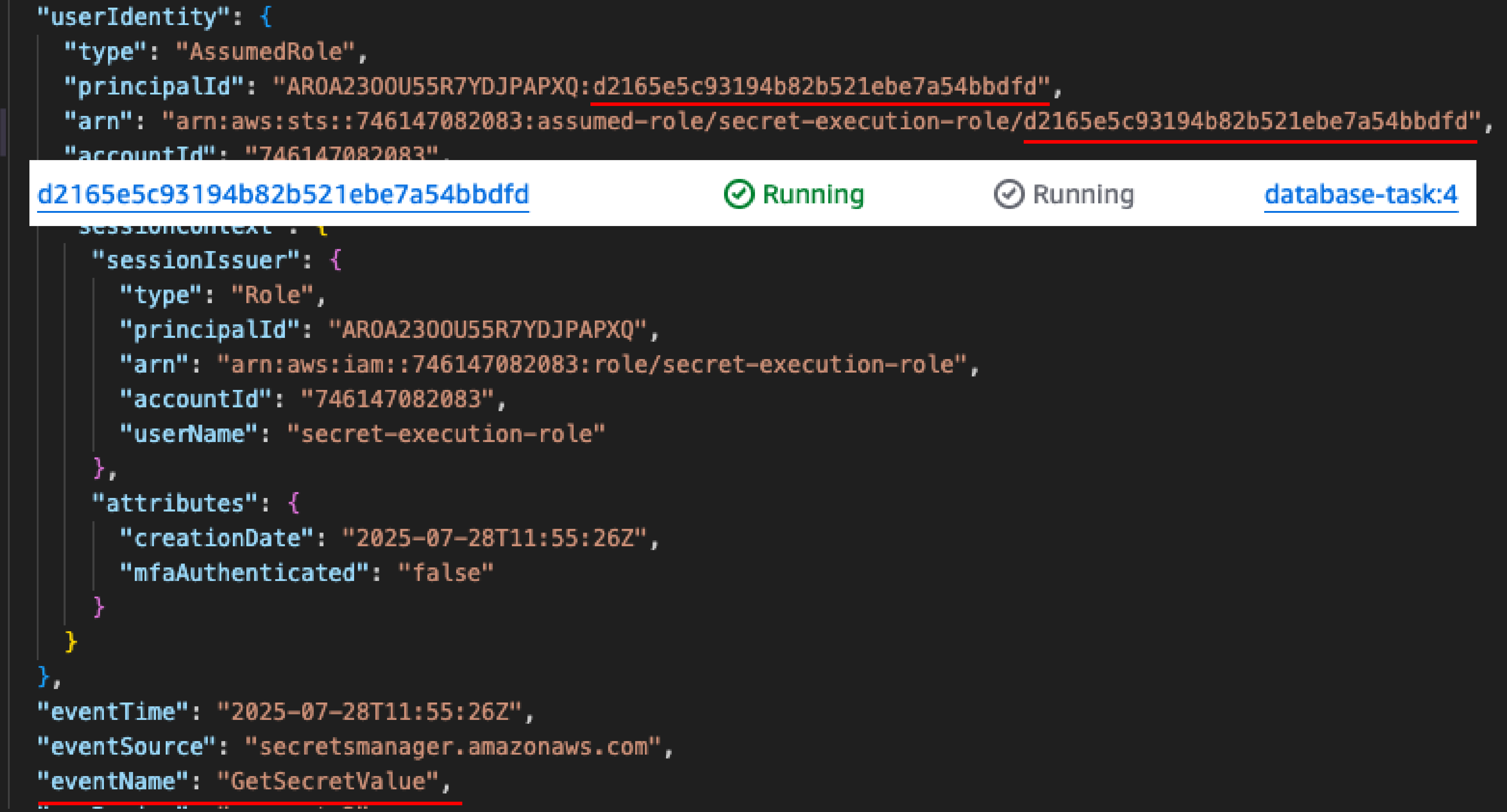
```
"userIdentity": {
  "type": "AssumedRole",
  "principalId": "AROA23OOU55R4HP5MBEPI:b549fae648d24d4dad76cef1c8d54154",
  "arn": "arn:aws:sts::746147082083:assumed-role/s3-control-role/b549fae648d24d4dad76cef1c8d54154",
  "accountId": "746147082083",
```

b549fae648d24d4dad76cef1c8d54154 ✓ Running ✓ Running s3-control-task:4

```
  "sessionContext": {
    "sessionIssuer": {
      "type": "Role",
      "principalId": "AROA23OOU55R4HP5MBEPI",
      "arn": "arn:aws:iam::746147082083:role/s3-control-role",
      "accountId": "746147082083",
      "userName": "s3-control-role"
    },
    "attributes": {
      "creationDate": "2025-07-28T11:54:25Z",
      "mfaAuthenticated": "false"
    }
  }
},
"eventTime": "2025-07-28T11:54:26Z",
"eventSource": "s3.amazonaws.com",
"eventName": "DeleteBucket",
```



```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROA23OOU55R7YDJPAPXQ:d2165e5c93194b82b521ebe7a54bbdfd",
    "arn": "arn:aws:sts::746147082083:assumed-role/secret-execution-role/d2165e5c93194b82b521ebe7a54bbdfd",
    "accountId": "746147082083",
    "accessKeyId": "ASIA23OOU55R6IJMJBON",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROA23OOU55R7YDJPAPXQ",
        "arn": "arn:aws:iam::746147082083:role/secret-execution-role",
        "accountId": "746147082083",
        "userName": "secret-execution-role"
      },
      "attributes": {
        "creationDate": "2025-07-28T11:55:26Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2025-07-28T11:55:26Z",
  "eventSource": "secretsmanager.amazonaws.com",
  "eventName": "GetSecretValue",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "18.191.77.44",
  "userAgent": "aws-sdk-rust/1.3.8 os/linux lang/rust/1.88.0",
  "requestParameters": {
    "secretId": "arn:aws:secretsmanager:us-east-2:746147082083:secret:db-secret-Po1uuv"
  },
  "responseElements": null,
  "requestID": "3322cf63-558b-4aba-bd2f-15fc1a0f6dfb",
  "eventID": "f758db74-f9df-4e60-b33e-0109fc7fb202",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "746147082083",
  "eventCategory": "Management",
  "tlsDetails": {
    "tlsVersion": "TLSv1.3",
    "cipherSuite": "TLS_AES_128_GCM_SHA256",
    "clientProvidedHostHeader": "secretsmanager.us-east-2.amazonaws.com"
  }
},
```



```
"userIdentity": {
  "type": "AssumedRole",
  "principalId": "AROA23OOU55R7YDJPAPXQ:d2165e5c93194b82b521ebe7a54bbdfd",
  "arn": "arn:aws:sts::746147082083:assumed-role/secret-execution-role/d2165e5c93194b82b521ebe7a54bbdfd",
  "accountId": "746147082083",
```

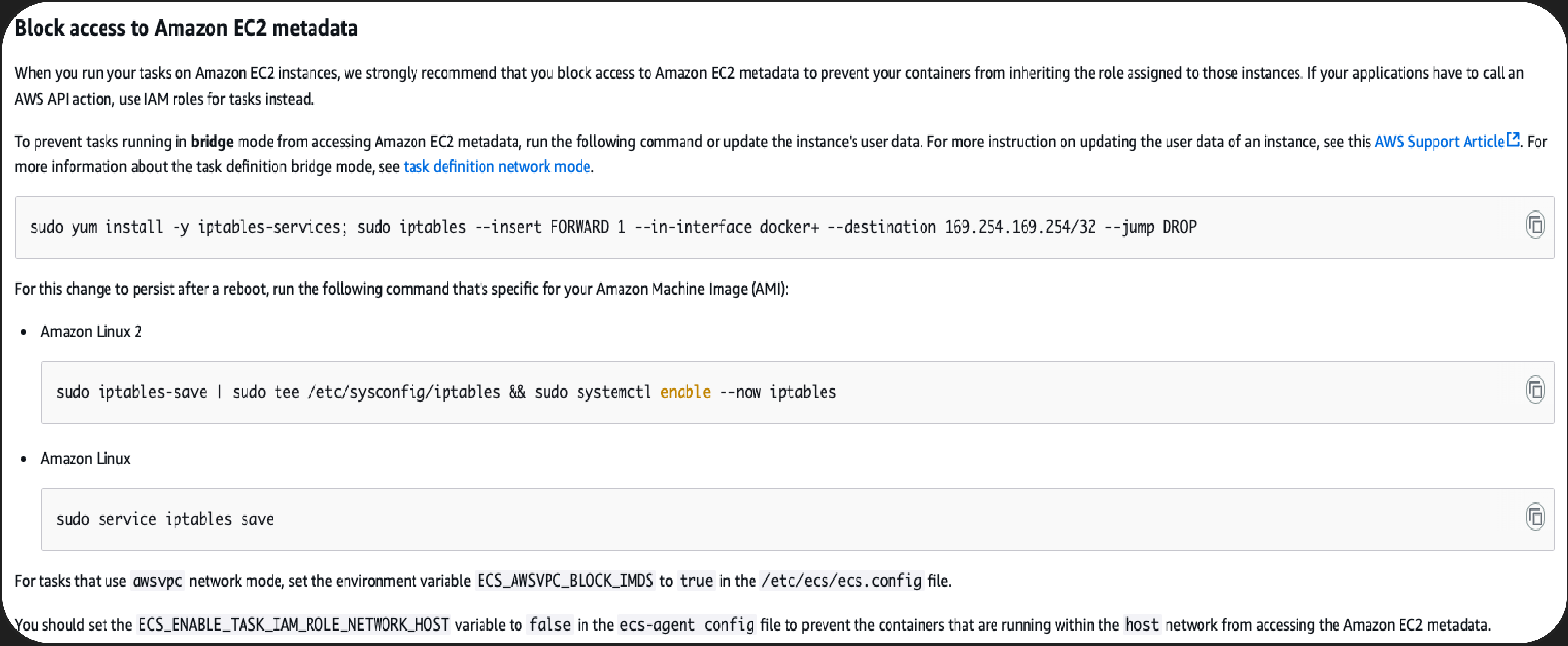
d2165e5c93194b82b521ebe7a54bbdfd ✓ Running ✓ Running database-task:4

```
   "sessionContext": {
    "sessionIssuer": {
      "type": "Role",
      "principalId": "AROA23OOU55R7YDJPAPXQ",
      "arn": "arn:aws:iam::746147082083:role/secret-execution-role",
      "accountId": "746147082083",
      "userName": "secret-execution-role"
    },
    "attributes": {
      "creationDate": "2025-07-28T11:55:26Z",
      "mfaAuthenticated": "false"
    }
  }
},
"eventTime": "2025-07-28T11:55:26Z",
"eventSource": "secretsmanager.amazonaws.com",
"eventName": "GetSecretValue",
```

ECScape POC GitHub:

06 Mitigation

Disable Tasks IMDS Access



Block access to Amazon EC2 metadata

When you run your tasks on Amazon EC2 instances, we strongly recommend that you block access to Amazon EC2 metadata to prevent your containers from inheriting the role assigned to those instances. If your applications have to call an AWS API action, use IAM roles for tasks instead.

To prevent tasks running in **bridge** mode from accessing Amazon EC2 metadata, run the following command or update the instance's user data. For more instruction on updating the user data of an instance, see this AWS Support Article. For more information about the task definition bridge mode, see task definition network mode.

```
sudo yum install -y iptables-services; sudo iptables --insert FORWARD 1 --in-interface docker+ --destination 169.254.169.254/32 --jump DROP
```

For this change to persist after a reboot, run the following command that's specific for your Amazon Machine Image (AMI):

- Amazon Linux 2

  ```
  sudo iptables-save | sudo tee /etc/sysconfig/iptables && sudo systemctl enable --now iptables
  ```

- Amazon Linux

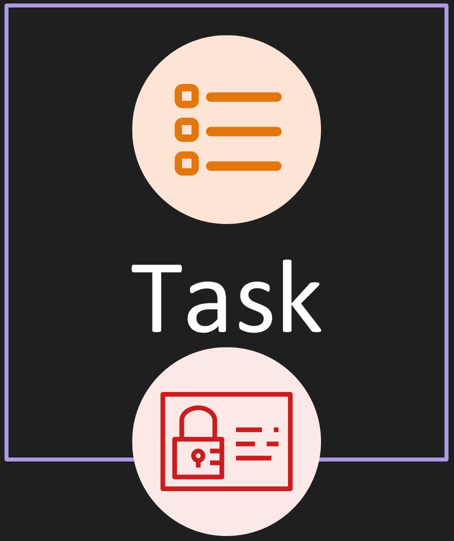
  ```
  sudo service iptables save
  ```

For tasks that use `awsvpc` network mode, set the environment variable `ECS_AWSVPC_BLOCK_IMDS` to `true` in the `/etc/ecs/ecs.config` file.

You should set the `ECS_ENABLE_TASK_IAM_ROLE_NETWORK_HOST` variable to `false` in the `ecs-agent config` file to prevent the containers that are running within the `host` network from accessing the Amazon EC2 metadata.

Task Role

```
"ecs:Disco          Endpoint",
"ecs:Poll"
```



Task

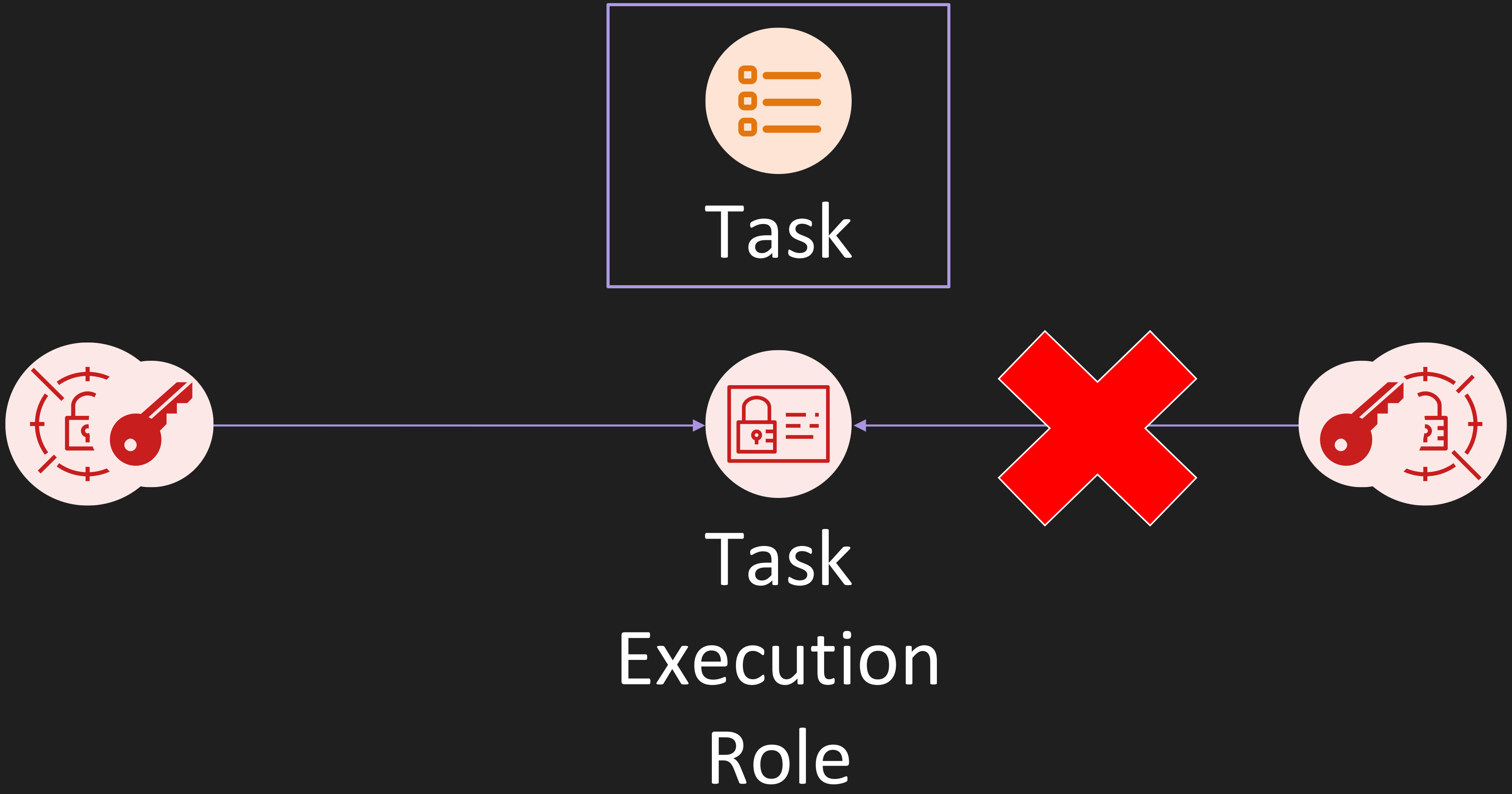
Task Role

```
{
                                 ",
                       "Allow",
                       "*",
                        "*"
}
```

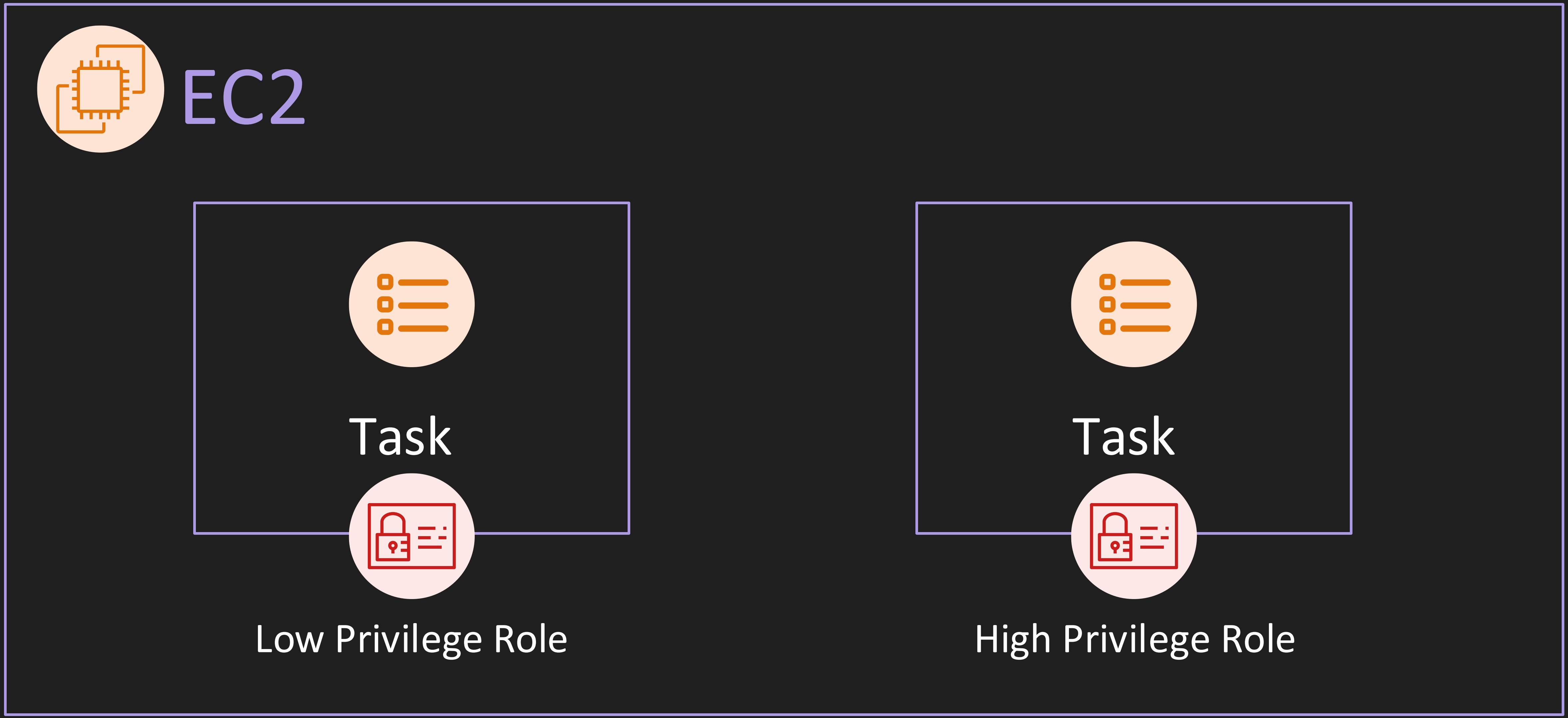
```
        ct"       ow",

      ]
                 *"
}
```

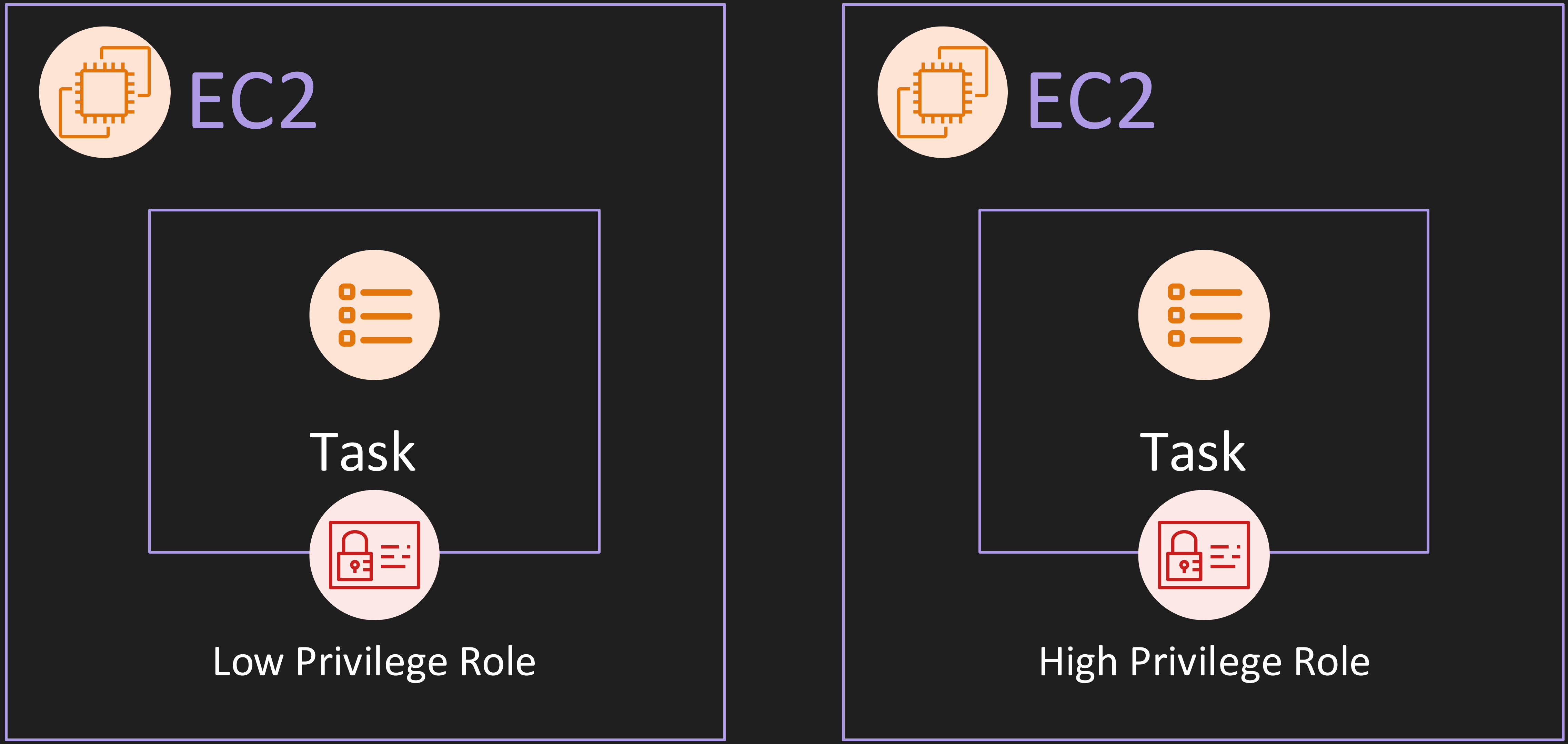
Minimize Task Execution Role Permissions



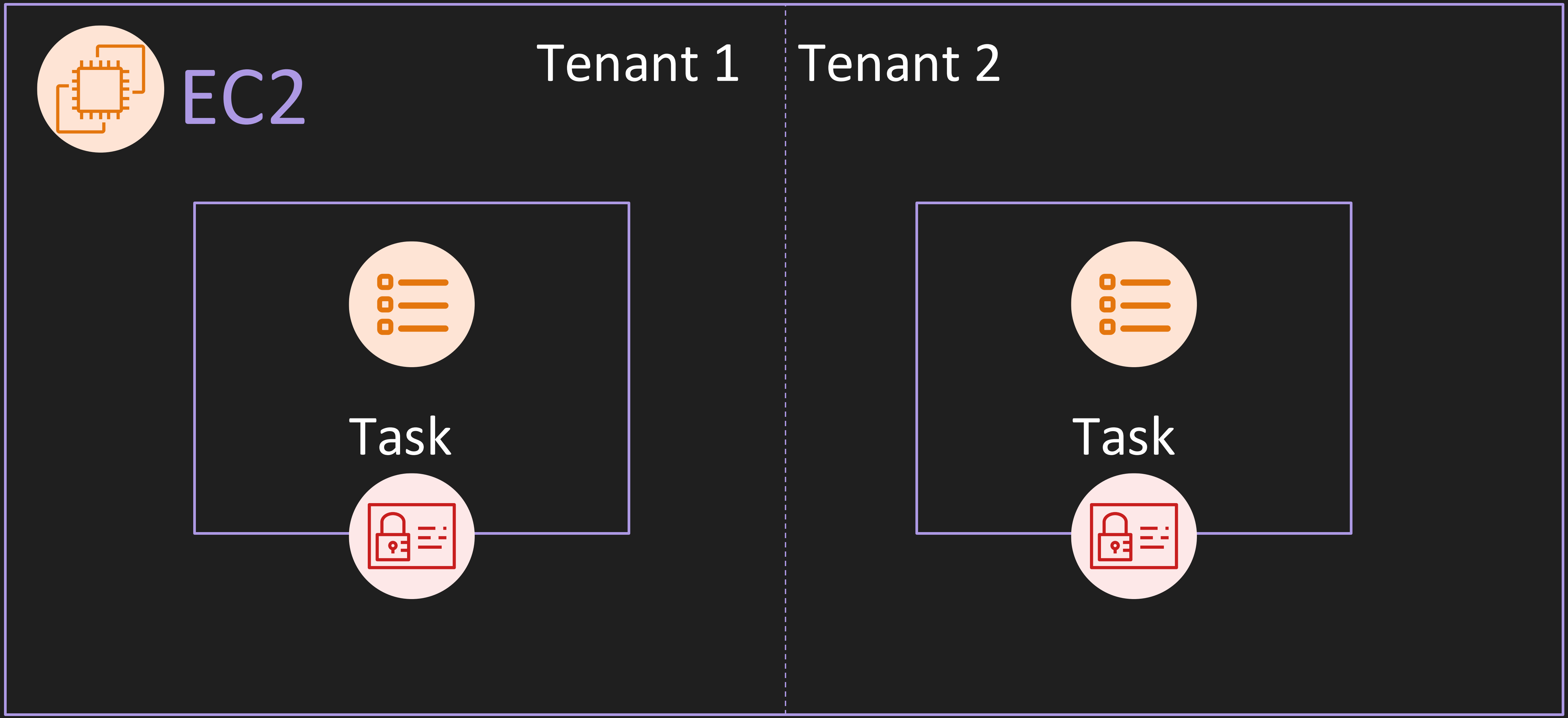
Separate high-privilege and low-privilege workloads



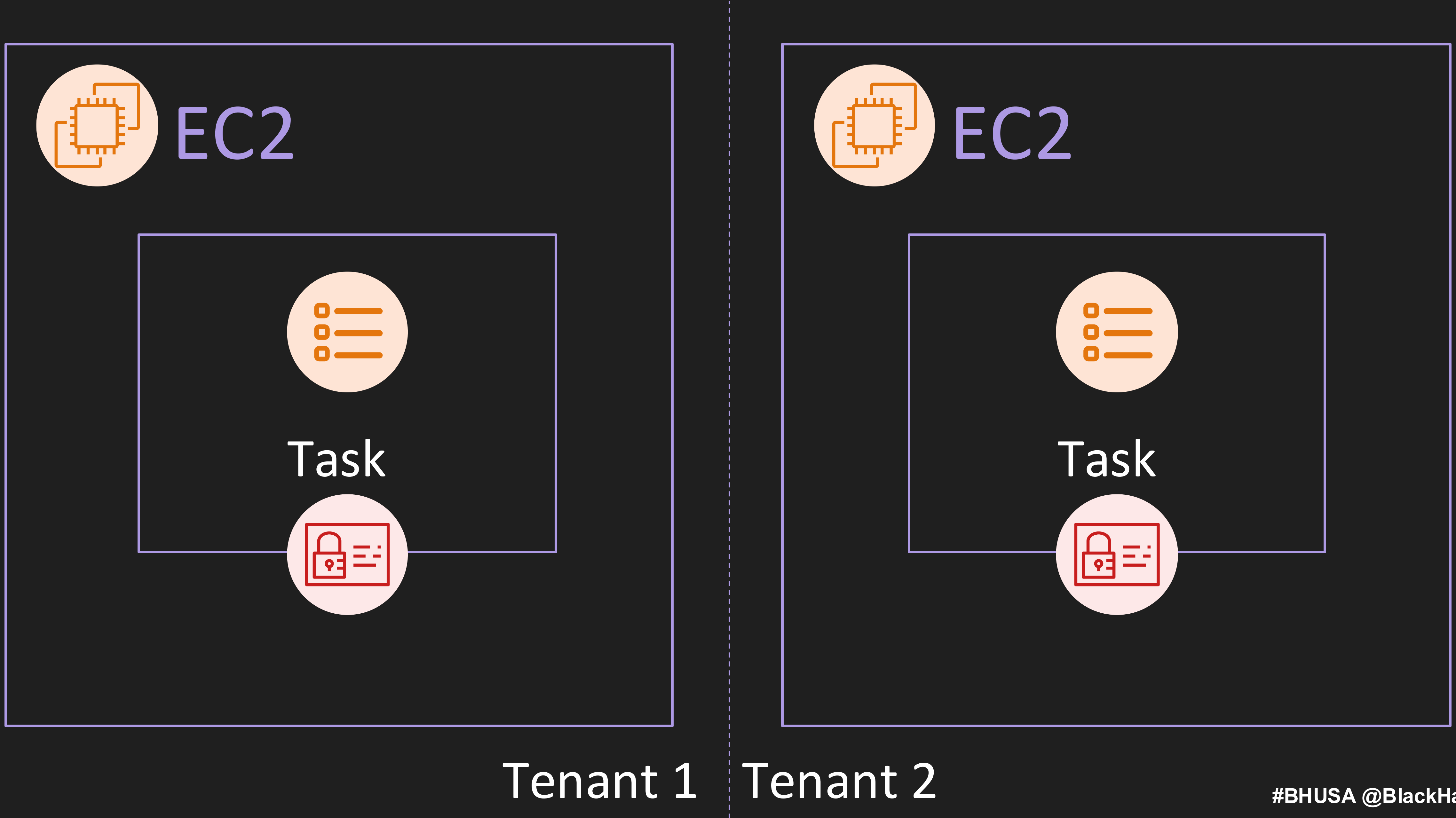
Separate high-privilege and low-privilege workloads



Isolate Tenants In Multi-Tenant Systems



Isolate Tenants In Multi-Tenant Systems

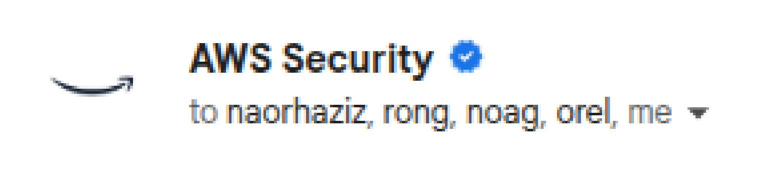


Best Practices

- Separate High and Low Privileged Workloads
- Isolate Tenants in Multi-Tenant Systems
- Minimize Task / Task Execution Role Permissions

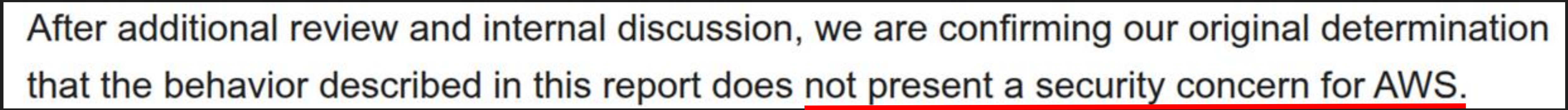
07 Summary

Vendor Response

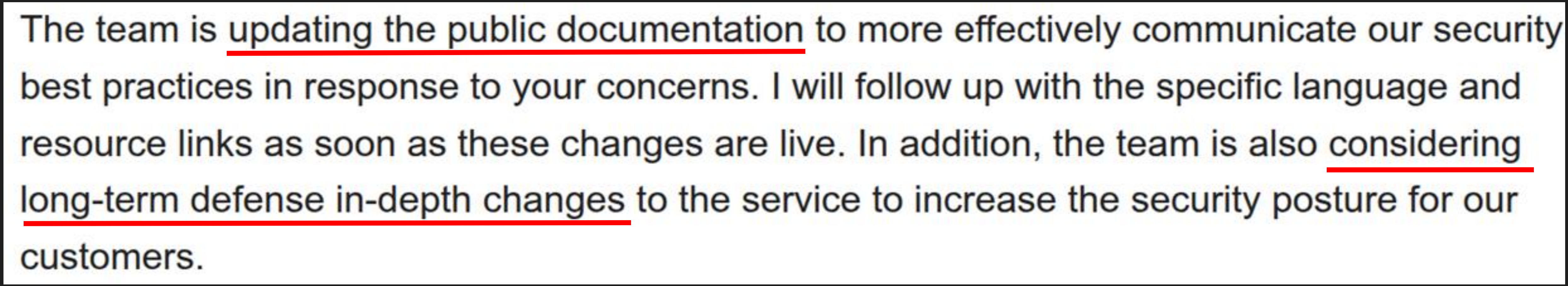


AWS Security

to naorhaziz, rong, noag, orel, me

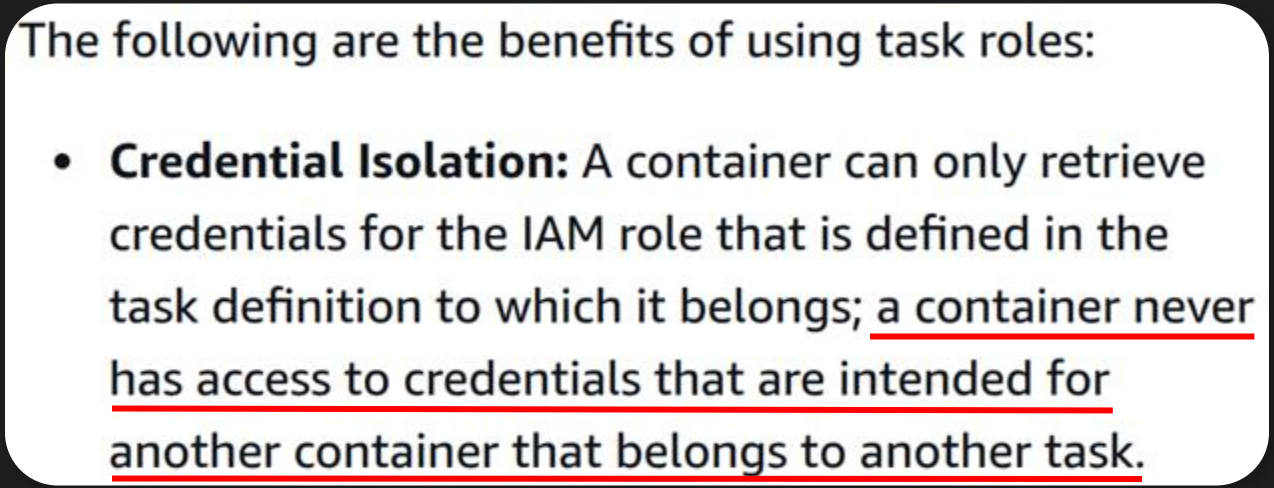


After additional review and internal discussion, we are confirming our original determination that the behavior described in this report does not present a security concern for AWS.



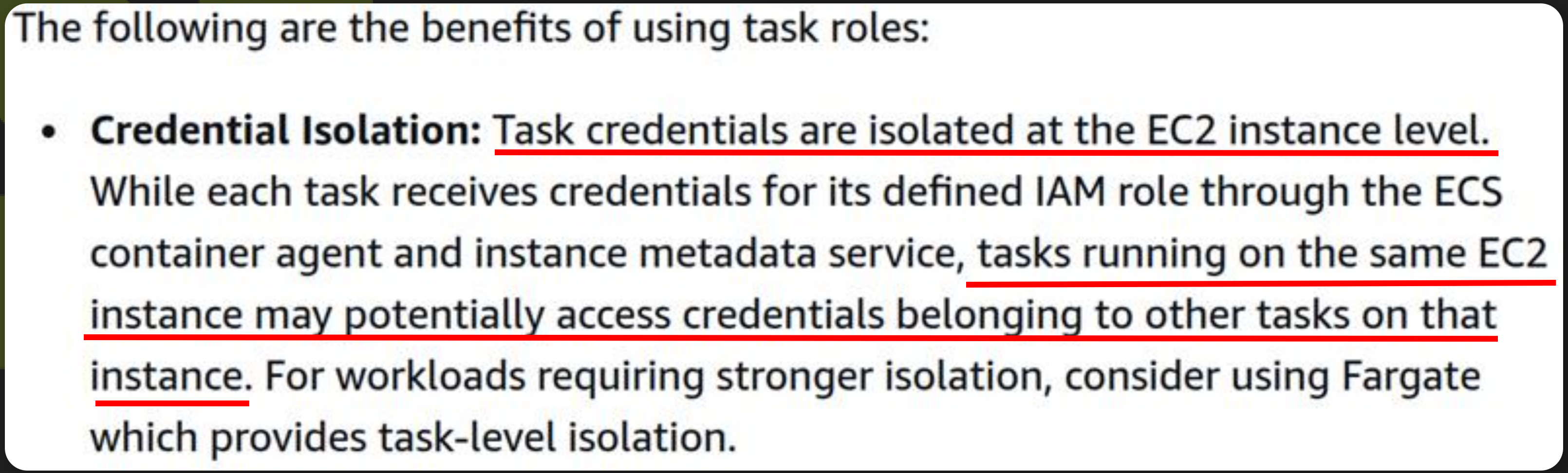
The team is updating the public documentation to more effectively communicate our security best practices in response to your concerns. I will follow up with the specific language and resource links as soon as these changes are live. In addition, the team is also considering long-term defense in-depth changes to the service to increase the security posture for our customers.

Documentation Change



The following are the benefits of using task roles:

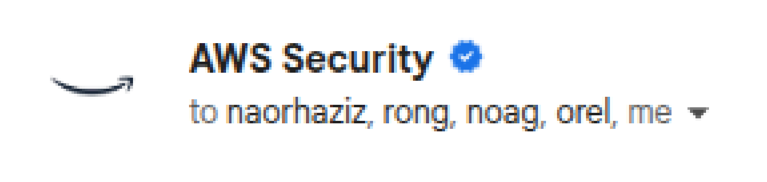
- **Credential Isolation:** A container can only retrieve credentials for the IAM role that is defined in the task definition to which it belongs; a container never has access to credentials that are intended for another container that belongs to another task.



The following are the benefits of using task roles:

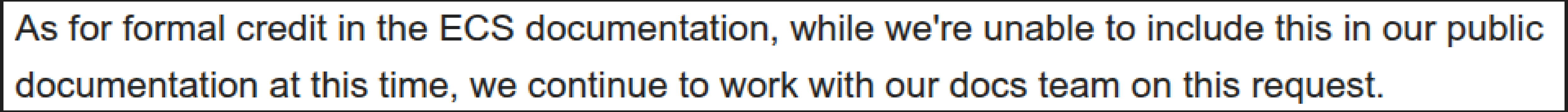
- **Credential Isolation:** Task credentials are isolated at the EC2 instance level. While each task receives credentials for its defined IAM role through the ECS container agent and instance metadata service, tasks running on the same EC2 instance may potentially access credentials belonging to other tasks on that instance. For workloads requiring stronger isolation, consider using Fargate which provides task-level isolation.

AWS Acknowledgements

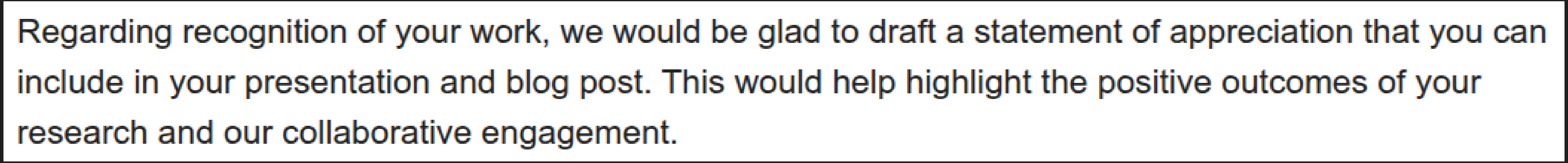


AWS Security

to naorhaziz, rong, noag, orel, me

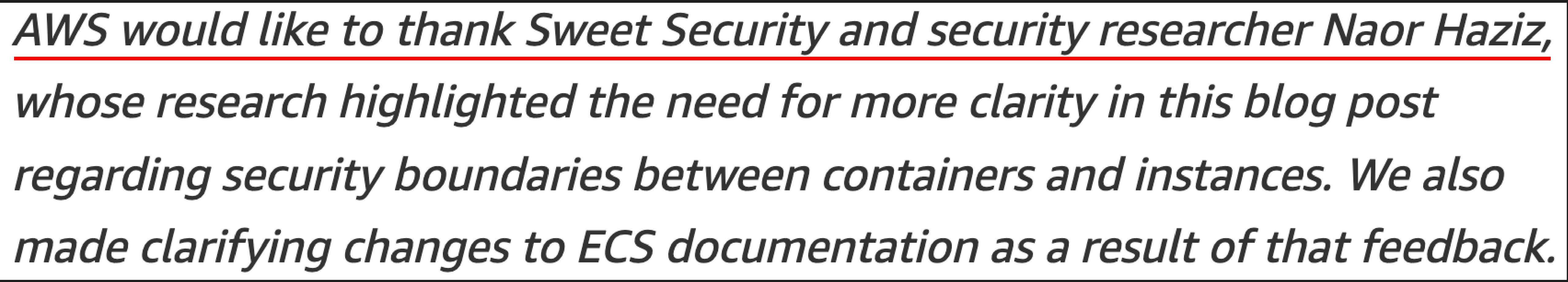


As for formal credit in the ECS documentation, while we're unable to include this in our public documentation at this time, we continue to work with our docs team on this request.



Regarding recognition of your work, we would be glad to draft a statement of appreciation that you can include in your presentation and blog post. This would help highlight the positive outcomes of your research and our collaborative engagement.

AWS Acknowledgements



AWS would like to thank Sweet Security and security researcher Naor Haziz, whose research highlighted the need for more clarity in this blog post regarding security boundaries between containers and instances. We also made clarifying changes to ECS documentation as a result of that feedback.

AWS Official Statement

The issues raised by Sweet Security are very important and instructive regarding basic elements of the AWS shared responsibility model [1]. While AWS often provides agents to run on customer-controlled EC2 instances [2] to provide service functionality (*e.g.*, ECS agent, CloudWatch agent, Systems Manager agent, EMR on EC2 agent, etc.), in all cases these agents run within the customer's security boundary, and any and all associated AWS roles (and their credentials and permissions) are understood and designed to be fully accessible to customers. The same is true of the open source components [3] running on customer EC2 instances used by the EKS service. Our threat model also assumes that such roles used by agents may be directly used and potentially abused by customers, and the services are designed to protect themselves from such possible abuse.

In the case of ECS, at the time of the original service launch the only roles/credentials/permissions made automatically available to tasks and containers were those of the underlying instance. Many customers continue to use ECS's IAM integration in that manner. Later, for customer convenience, a different set of roles (and associated credentials and permissions) was made available at the task level [4] [5] to separate and simplify management of the permissions granted directly to customer code running inside containers in ECS tasks. At that time, we added documentation [6] for iptables-based techniques whereby customers could deny network access to the underlying instance credentials from hosted containers. That configuration remains an option and is not the default behavior. AWS continuously reviews default configurations in our services and as over time a decreasing number customers use instance credentials in ECS/EC2, changes to this default behavior are under consideration. However, even if the default networking behavior was changed, containers were and are never considered a security boundary in AWS [7]. Thus, even such a networking change could make it more complicated for containers to access the privileges available to the ECS agent, not make it impossible. Moreover, in EC2-based deployments of ECS, the customer is in full control of both the underlying instance as well as the associated tasks and containers that run inside it. Thus, customers are responsible for guarding against all security issues within a container seeking to access code and data in the underlying instance or other containers hosted on it. In sum, whatever IAM privileges exist in the underlying instance / operating system are assumed to be and are directly or indirectly available to customers, and to the code that they deploy, one way or another.

AWS would like to thank Sweet Security for their interesting and valuable research, which resulted in modifications and clarifications in our documentation and an existing blog post to make more explicit the security boundaries and the implicit threat model of the ECS service (as well as, by implication, analogous scenarios involving AWS-supplied agents running on customer-controlled EC2 instances). Our customers have benefited from this research and collaboration.

[1] https://aws.amazon.com/compliance/shared-responsibility-model/
[2] https://docs.aws.amazon.com/AmazonECS/latest/developerguide/ecs-agent-config.html
[3] https://docs.aws.amazon.com/eks/latest/userguide/eks-compute.html
[4] https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task-iam-roles.html
[5] https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task_execution_IAM_role.html
[6] https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task-metadata-endpoint-v2.html
[7] https://aws.amazon.com/blogs/security/security-considerations-for-running-containers-on-amazon-ecs

Summary

- On EC2, tasks and the ECS agent share one trust boundary
- A task can impersonate the ECS agent
- Task-level hardening is essential

Thanks!

✉ naorhaziz@gmail.com

in Naor Haziz

https://github.com/naorhaziz

I'm so glad each task has its own IAM role
Then how's it getting other task creds on that instance?