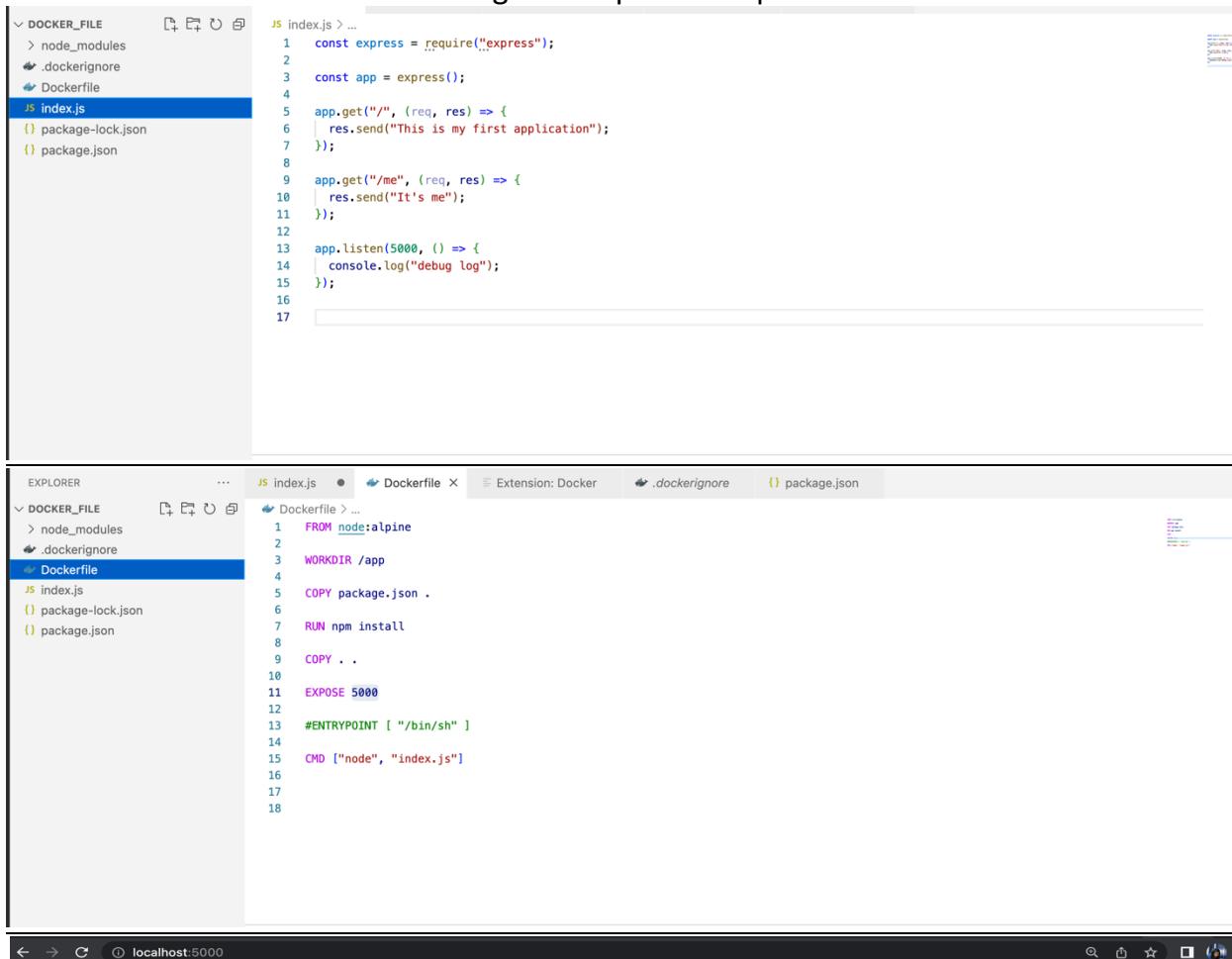


Deploy a Docker App to AWS ECS

STEP 1: Create an application and run it on local system to make sure that there would be no errors it's working fine as per the expected behavior.



The screenshot shows a code editor with two files open. The top file is `index.js` containing the following Node.js code:

```
1  const express = require("express");
2
3  const app = express();
4
5  app.get("/", (req, res) => {
6    | res.send("This is my first application");
7  });
8
9  app.get("/me", (req, res) => {
10   | res.send("It's me");
11 });
12
13 app.listen(5000, () => {
14   | console.log("debug log");
15 });
16
17
```

The bottom file is `Dockerfile` containing the following Dockerfile code:

```
1  FROM node:alpine
2
3  WORKDIR /app
4
5  COPY package.json .
6
7  RUN npm install
8
9  COPY .
10
11 EXPOSE 5000
12
13 #ENTRYPOINT [ "/bin/sh" ]
14
15 CMD [ "node", "index.js" ]
16
17
18
```

This is my first application



STEP 2: Create docker image for this application

STEP 3: Use docker run command to create a container from disc image

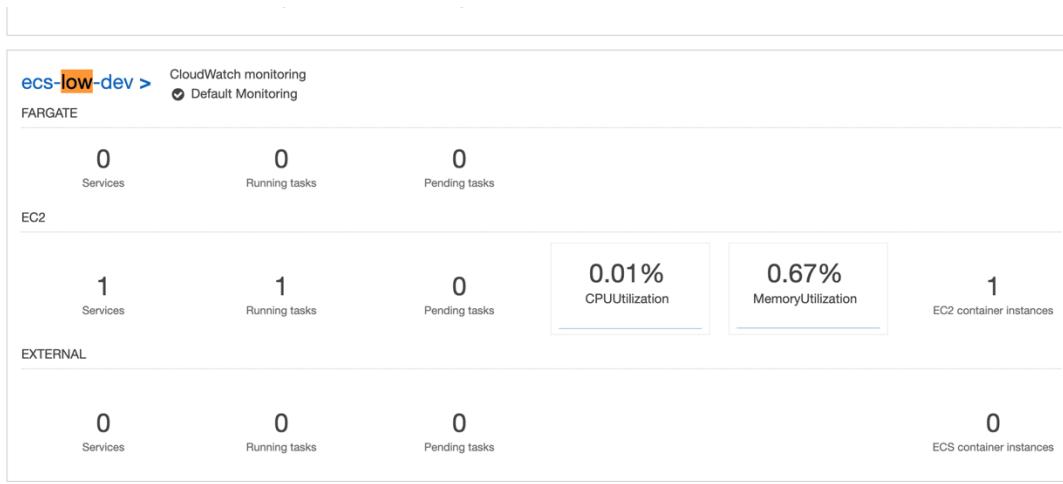
```
asingh4-26017s:Docker_file asingh4$ lsof -i :5000
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
node 15392 asingh4 22u IPv6 0xa92845a0ae8bee8b 0t0 TCP *:commplex-main (LISTEN)
asingh4-26017s:Docker_file asingh4$ kill -9 15392
asingh4-26017s:Docker_file asingh4$
asingh4-26017s:Docker_file asingh4$
asingh4-26017s:Docker_file asingh4$ docker build -t ankita-docker .
[+] Building 16.1s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 43B
=> [internal] load .dockerignore
=> => transferring context: 35B
=> [internal] load metadata for docker.io/library/node:alpine
0.0s
0.0s
0.0s
0.0s
1.3s
```

STEP 4: Try to hit the url

```
asingh4-26017s:Docker_file asingh4$ docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
b18b0283623b cef88eecb3c1 "docker-entrypoint.s..." 5 hours ago Up 5 hours 0.0.0.0:8080->5000/tcp tender_jemison
asingh4-26017s:Docker_file asingh4$ docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
ankita-docker latest 88869ba1443f 2 minutes ago 360MB
express-app latest a08a63c4a74e 33 minutes ago 180MB
<none> <none> cef88eecb3c1 30 hours ago 180MB
<none> <none> fad6c98f259f 30 hours ago 180MB
<none> <none> 20a2f3ac8b48 30 hours ago 180MB
<none> <none> 2556948ed004 30 hours ago 180MB
ubuntu latest 27941809078c 4 weeks ago 77.8MB
docker/getting-started latest cb90f98fd791 2 months ago 28.8MB
docker/dev-environments-default stable-1 7c85b0303242 11 months ago 607MB
docker/desktop-git-helper 5a4fcfa126aacd3f6cc3a011aa991de982ae7000 efe2d67c403b 11 months ago 44.2MB
asingh4-26017s:Docker_file asingh4$ docker run -it -p 5000:5000 88869ba1443f
```

```
asingh4-26017s:Docker_file asingh4$ docker run -it -p 5000:5000 first-dockerapp
debug log
```

STEP 5: Go to the below cluster



Step 6: To create a cluster before that first create “Task Definition”

Adding one more container to the task definition which was created

Amazon EKS

Clusters

Amazon ECR

Repositories

AWS Marketplace

Discover software

Subscriptions ↗

Builder JSON Tags

Task definition name express_app

Task role None

Optional IAM role that tasks can use to make API requests to authorized AWS services. Create an Amazon Elastic Container Service Task Role in the [IAM Console](#) ↗

Network mode <default>

If you choose <default>, ECS will start your container using Docker's default networking mode, which is Bridge on Linux and NAT on Windows. Windows tasks support the <default> and awsvpc network modes.

Compatibilities EXTERNAL, EC2

Requires compatibilities EC2

Task execution IAM role

This role is required by tasks to pull container images and publish container logs to Amazon CloudWatch on your behalf. If you do not have the `ecsTaskExecutionRole` already, we

The task size allows you to specify a fixed size for your task. Task size is required for tasks using the Fargate launch type and is optional for the EC2 or External launch type. Container level memory settings are optional when task size is set. Task size is not supported for Windows containers.

Task memory (MiB) --

Task CPU (unit) --

Task memory maximum allocation for container memory reservation



0 128 shared of 128 MiB total

Task Placement

Constraint No constraints

Container definitions

Container Name	Image	CPU Units	GPU	Inference Accelerators	Hard/Soft memory limits (MiB)	Essential
dockerapp...	public.ecr.aws/f...	0			128/--	true

Volumes

Requires attributes

Name	Value

[AWS Copilot CLI guide:](#)

[\(https://aws.github.io/copilot-cli/docs/getting-started/first-app-tutorial/\)](https://aws.github.io/copilot-cli/docs/getting-started/first-app-tutorial/)

Error encountered

```
/usr/local/bin/aws
asingh4-26017s:~ asingh4$ aws ecr-public get-login-password --region us-east-1 | docker login --username AWS --password-stdin public.ecr.aws/f3f7a1w6
The SSO session associated with this profile has expired or is otherwise invalid. To refresh this SSO session run aws sso login with the corresponding profile.
Error: Cannot perform an interactive login from a non TTY device
```

Step 7: Make “Dev” profile your default profile

```

asingh4-26017s:~ asingh4$ vim ~/.aws/config
asingh4-26017s:~ asingh4$ vim ~/.aws/credentials
asingh4-26017s:~ asingh4$ cat ~/.aws/config
[profile nrel-aws-dev-developers]
region = us-west-2
output = json
sso_start_url = https://nrel-ace.awsapps.com/start#
sso_account_id = 246460460343
sso_role_name = developers
sso_region = us-west-2

[profile nrel-aws-ace-cyber-read-only-access]
region = us-west-2
output = json
sso_start_url = https://nrel-ace.awsapps.com/start#
sso_account_id = 991404956194
sso_role_name = cyber-read-only-access
sso_region = us-west-2

[default]
region = us-west-2
output = json
sso_start_url = https://nrel-ace.awsapps.com/start#
sso_account_id = 246460460343
sso_role_name = developers
sso_region = us-west-2

asingh4-26017s:~ asingh4$ cat ~/.aws/credentials

```

Step 8: To get the image that we created into AWS we will have to Create a private repository in ECS service

Clusters

An Amazon ECS cluster is a regional grouping of one or more container instances on which you can run task requests. Each account receives a default cluster the first time you use the Amazon ECS service. Clusters may contain more than one Amazon EC2 instance type.

For more information, see the [ECS documentation](#).

[Create Cluster](#) [Get Started](#)

View [list](#) [card](#) [GO to repository to create one](#) [view all](#) [refresh](#)

computeenvionment_noeltest31_Batch_ba965bff-4947-3e67-9b70-6293d13e3ed8 > CloudWatch monitoring Default Monitoring

FARGATE

0 0 0

General settings

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.

246460460343.dkr.ecr.us-west-2.amazonaws.com/

14 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.

Tag immutability [Info](#)
Enable tag immutability to prevent image tags from being overwritten by subsequent image pushes using the same tag. Disable tag immutability to allow image tags to be overwritten.

Disabled

Once a repository is created, the visibility setting of the repository can't be changed.

Image scan settings

To Push our image into this repository we need to go to the repo and click on “View commands “ it will guide us through pushing the image into the repo.

Push commands for repo-firstapp

X

macOS / Linux

Windows

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  
246460460343.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 repo-firstapp .
```

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

```
docker tag repo-firstapp:latest 246460460343.dkr.ecr.us-west-2.amazonaws.com/repo-firstapp:latest
```

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

```
docker push 246460460343.dkr.ecr.us-west-2.amazonaws.com/repo-firstapp:latest
```

Copied

Step 9: Perform SSO Log in (for this we need to have AWSCLI version 2 which I updated by uninstalling the current version)

```
asingh4-26017s:~ asingh4$ cat ~/.aws/credentials  
[246460460343_developers]  
aws_access_key_id=ASIASTSQRQFU3ZM6PFUZT  
aws_secret_access_key=APuN+G+mCdCwZALCjGetoHNd6InSPLGCBAh6MO  
aws_session_token=IQoJb3JpZ2IuX2VjENT//////////wEaXVzLXd1c3QtMjJHMEUCIQD1GdTQy1800IbdRR7bYxzK7DtpVR1Ks0vZwShJW0e9rwIgZdaR+98ei+qcc3/b21chUFhz042yLky5YR6IQ  
Xdde10qgQ1/f//////////ARADGwyNDY0NjA0NjA2nDm1DHQd2Y-5hfc4Tf73yRA65H91gjpJWxtn2w9FhkUNB6uayEB0DgNTE9WnPQ68aySdGAtyxw+taYon1mtBsukGC1cxDFT1TQ7cKv6xmrv  
cZzvwoG6fV99whUhoBKv540uI968wJTF+4DV64p7N+vb/uuAb71YFSzb6wf6Kz6j1ey7zRxDtu60IO+pTxeaxLwts4Ugw9USKNVC3iifLONA4ab4PPF4RfcCsuo+KaD230xcxjrs8OKNs55DnFuNRI0  
3drH8CtH41spXUZTfrJayg+byJ+2ek1CHM5fU1adraAvtx4TY/R525x0MyCaSw0w41xdwz1FnPZG+upaoId0ibkyMfynx0f7yZM5/wSqGrWPcmfR6X1vV18B3E5H0FByj6fVRyCb0QHaKSwxEZaqz  
3Ftqak6Gdf4QW6Z7srDV9w7t0vj0Md4qSMUehBRwEkY5MDHP01UYq70c8jUhiRnYrorBo+ZCawCgCmtyKNmVwPSCpYQd208a0c2b8yi5TSTUStgyHvibk/v53H0P01x/nQRASNL.peYeP+SnElffB6  
nrqtM/2+tgB03fDMuwtkUkw0VOJMRKw91ALK7cEcjihs6nK96zbDnDfP0yKsSnPc12a11vJdrgKSWBjqmAYuGh/fuIM9c0RRakNgUZ+iKyTFEHNm1gkZQw/cjK84eHByJ+j8D22VLkjwC8Z9gMtrvumdT  
2pbVh041+bwS8eWnzgh7tmCOA1mdJGhsol1jcr2Fhlobne1Dg8hodu181+oIzs/1U/o8FDUKWpFJfW2rzcLJgskMiW7eaHXJ0NY22U2Q9AxnnhNV+yAH2gQXEdjxJGhyzob5BtMkpULYxQ8=
```

```
asingh4-26017s:~ asingh4$ vi ~/.aws/credentials  
asingh4-26017s:~ asingh4$ aws ecr-public get-login-password --region us-east-1 | docker login --username AWS --password-stdin public.ecr.aws/f3f7a1w6  
The SSO session associated with this profile has expired or is otherwise invalid. To refresh this SSO session run aws sso login with the corresponding profile.  
Error: Cannot perform an interactive login from a non TTY device  
asingh4-26017s:~ asingh4$ aws sso login  
Attempting to automatically open the SSO authorization page in your default browser.  
If the browser does not open or you wish to use a different device to authorize this request, open the following URL:  
https://device.sso.us-west-2.amazonaws.com/  
Then enter the code:  
QJQG-SPDF  
Successfully logged into Start URL: https://nrel-ace.awssapps.com/start/  
asingh4-26017s:~ asingh4$ aws ecr-public get-login-password --region us-east-1 | docker login --username AWS --password-stdin public.ecr.aws/f3f7a1w6  
Login Succeeded  
Logging in with your password grants your terminal complete access to your account.  
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/  
asingh4-26017s:~ asingh4$ aws ecr-public get-login-password --region us-east-1 | docker login --username AWS --password-stdin public.ecr.aws/f3f7a1w6  
Login Succeeded
```

Step 9: build the image

```
asingh4-26017s:Docker_file asingh4$ aws ecr get-login-password --region us-west-2 | docker login --username AWS --password-stdin 246460460343.dkr.ecr.us-west-2.amazonaws.com
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/
[asingh4-26017s:Docker_file asingh4$ docker build -t repo-firstapp .
[+] Building 1.3s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> transferring dockerfile: 43B
=> [internal] load .dockerignore
=> => transferring context: 35B
=> [internal] load metadata for docker.io/library/node:alpine
=> [auth] library/node:pull token for registry-1.docker.io
=> [1/5] FROM docker.io/library/node:alpine@sha256:e479d86de1ef8403adda6800733a7f8d18df4f3c1c2e68ba3e2bc05fdea9de20
=> => resolve docker.io/library/node:alpine@sha256:e479d86de1ef8403adda6800733a7f8d18df4f3c1c2e68ba3e2bc05fdea9de20
=> [internal] load build context
=> => transferring context: 100B
=> CACHED [2/5] WORKDIR /app
=> CACHED [3/5] COPY package.json .
=> CACHED [4/5] RUN npm install
=> CACHED [5/5] COPY . .
=> exporting to image
=> => exporting layers
=> => writing image sha256:ad960c44c1b024ccf42b7c6dbd56bb9186a63053fc17dc84a245cd910838593
=> => naming to docker.io/library/repo-firstapp
```

Step 10: Pushing docker image on ECR by tagging it

```
asingh4-26017s:Docker_file asingh4$ docker tag repo-firstapp:latest 246460460343.dkr.ecr.us-west-2.amazonaws.com/repo-firstapp:latest
asingh4-26017s:Docker_file asingh4$ docker push 246460460343.dkr.ecr.us-west-2.amazonaws.com/repo-firstapp:latest
The push refers to repository [246460460343.dkr.ecr.us-west-2.amazonaws.com/repo-firstapp]          Pushing the image
d38741d8076c: Pushed
f9cac0f9b4c1: Pushed
dee2bc3c5aea: Pushed
347104bc0386: Pushed
9b31eee258f6: Pushed
55a7fed1743e: Pushed
f916969e36f7: Pushed
24302eb7d908: Pushed
latest: digest: sha256:3cf83385bafa54901477377b6dbd7e6c3276debd7765744feecb4e6bc363bea3 size: 1992
asingh4-26017s:Docker_file asingh4$
```

tagging the image

Pushing the image

Till now I have just pushed my image into the AWS repo it is still not running any application so in order to do so I will have to set up a networking in ECS service

The screenshot shows the AWS ECR console with the following details:

- Repository:** repo-firstapp
- Images (1):**
 - Image tag:** latest
 - Artifact type:** Image
 - Pushed at:** July 13, 2022, 12:00:38 (UTC-05)
 - Size (MB):** 135.19
 - Image URI:** sha256:3cf83385bafa549...
 - Digest:** sha256:3cf83385bafa549...
 - Vulnerabilities:** See findings

Step 11: Adding & Editing permissions

The screenshot shows the AWS ECR Permissions 'Edit permissions' page. The 'Add statement' button is highlighted with a red box. The JSON editor contains the following policy statement:

```
{ "Effect": "Allow", "Principal": "246460460343", "Action": "ecr:DescribeImages", "Condition": { "StringEquals": { "aws:username": "johndoe" } } }
```

Step 12: go to the “Edit Jason” option and edit the commands

Amazon Elastic Container Registry

Private registry

Public registry

Repositories

Summary

Images

Permissions

Lifecycle Policy

Tags

Getting started

Documentation

Public gallery

Amazon ECR > Repositories > repo-firstapp > Permissions > Edit permissions

Edit JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "repo-access",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::246460460343:root",
          "arn:aws:iam::246460460343:role/asingh4-genericECSDeveloperRole"
        ]
      },
      "NotAction": [
        "ecr:DescribeRepositories"
      ]
    }
  ]
}
```

Reset Close Save

AWS account IDs - optional
The AWS accounts to apply the statement to. All users under the AWS account will be affected.

246460460343

Edit JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "repo-access",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::246460460343:root",
          "arn:aws:iam::246460460343:role/asingh4-genericECSDeveloperRole"
        ]
      },
      "Action": "ecr:)"
    }
  ]
}
```

Reset Close Save

```
denied: requested access to the resource is denied
asingh4-26017s:Docker_file asingh4$ ecrurl=246460460343.dkr.ecr.us-west-2.amazonaws.com
asingh4-26017s:Docker_file asingh4$ asingh4-26017s:Docker_file asingh4$ asingh4-26017s:Docker_file asingh4$ dockerurl="$ecrurl/repo-firstapp"
asingh4-26017s:Docker_file asingh4$ echo $dockerurl
246460460343.dkr.ecr.us-west-2.amazonaws.com/repo-firstapp
asingh4-26017s:Docker_file asingh4$ asingh4-26017s:Docker_file asingh4$ docker tag repo-firstapp:latest $ecrurl:latest
asingh4-26017s:Docker_file asingh4$ asingh4-26017s:Docker_file asingh4$ docker tag repo-firstapp:latest $dockerurl:latest
asingh4-26017s:Docker_file asingh4$ asingh4-26017s:Docker_file asingh4$ docker push $dockerurl:latest
[The push refers to repository [246460460343.dkr.ecr.us-west-2.amazonaws.com/repo-firstapp]
56741d0070c: Layer already exists
f9cac0f9b4c1: Layer already exists
dee2bc3c5aea: Layer already exists
347104bc0386: Layer already exists
9b31eee258f6: Layer already exists
55a7fed1743e: Layer already exists
f916969e36f7: Layer already exists
24302eb7d908: Layer already exists
latest: digest: sha256:3cf83385bafa54901477377b6dbd7e6c3276debd7765744feecb4e6bc363bea3 size: 1992
asingh4-26017s:Docker_file asingh4$
```

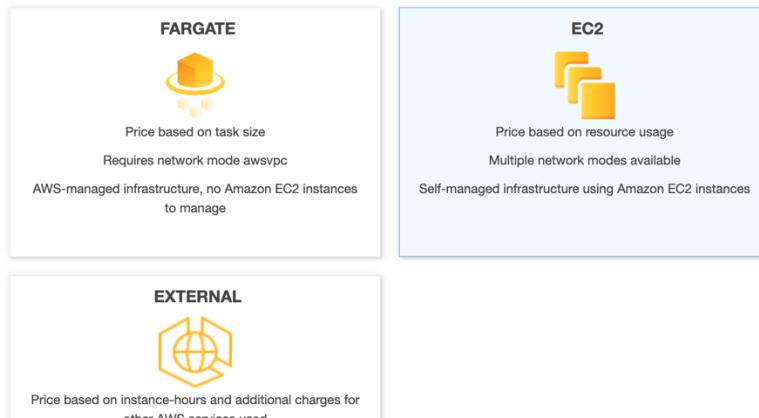
Create new Task Definition

Step 1: Select launch type compatibility

Step 2: Configure task and container definitions

Select launch type compatibility

Select which launch type you want your task definition to be compatible with based on where you want to launch your task.



Create new Task Definition

Step 1: Select launch type compatibility

Step 2: Configure task and container definitions

Configure task and container definitions

A task definition specifies which containers are included in your task and how they interact with each other. You can also specify data volumes for your containers to use. [Learn more](#)

Task definition name*

Requires compatibilities* EC2

Task role  
 Optional IAM role that tasks can use to make API requests to authorized AWS services. Create an Amazon Elastic Container Service Task Role in the [IAM Console](#).

Network mode  
 If you choose <default>, ECS will start your container using Docker's default networking mode, which is Bridge on Linux and NAT on Windows. Windows tasks support the <default> and awsvpc network modes.

us-west-2.console.aws.amazon.com/ecs/home?region=us-west-2#/taskDefinitions/create

Add container

for the EC2 Windows container

Standard

Container name*

Image*

Private repository authentication* so for private repo, we need to create the roles which will have the permission to pull from the repository.

Secrets Manager ARN or name

The task execution role is required to use this feature. If your secret uses a custom encryption key, then the task execution role requires KMS key decrypt permissions ("kms:Decrypt"). Otherwise, image authentication fails, preventing the container image pull. [Learn more](#)

Memory Limits (MiB)*  

Add Soft limit

Define hard and/or soft memory limits in MiB for your container. Hard and soft limits correspond to the 'memory' and 'memoryReservation' parameters, respectively, in task definition. [Learn more](#)

* Required

Cancel **Add**

AWS Services Search for services, features, blogs, docs, and more [Option+S] Global developers/asinh4@nrel.gov

Identity and Access Management (IAM)

Introducing the new IAM roles experience We've redesigned the IAM roles experience to make it easier to use. [Let us know what you think.](#)

New! Securely access AWS services from your data center with IAM Roles Anywhere. [Learn more](#)

Unable to load search Dashboard

Access management

- User groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings

Access reports

- Access analyzer
- Archive rules
- Analyzers
- Settings
- Credential report
- Organization activity
- Service control policies (SCPs)

Search for services, features, blogs, docs, and more [Option+S] Global developers/asinh4@nrel.gov

IAM > Roles

Roles (389) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
AccessKeyRotatedFunction-role-y4dy7a7	AWS Service: lambda	-
AccessKeyRotateFunction-role-aw07cl70	AWS Service: lambda	-
airflow-jgu2-env-execution-role	AWS Service: airflow-env, and 1 more. Edit	-
AmazonEC2SpotFleetRole	AWS Service: spotfleet	-
AmazonMWAAP-MyAirflowEnvironment-OBb6rz	AWS Service: airflow-env, and 1 more. Edit	-
AmazonSageMaker-ExecutionRole-20171214T094065	AWS Service: sagemaker	-
AmazonSageMaker-ExecutionRole-20200803T165820	AWS Service: sagemaker	-

IAM > Roles > Create role

Step 1 Select trusted entity

Step 2 Add permissions

Step 3 Name, review, and create

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.

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:

perform actions in this account. 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:

Elastic Container Service

- Elastic Container Service Allows ECS to create and manage AWS resources on your behalf.
- Elastic Container Service Autoscale Allows Auto Scaling to access and update ECS services.
- Elastic Container Service Task Allows ECS tasks to call AWS services on your behalf.
- EC2 Role for Elastic Container Service Allows EC2 instances in an ECS cluster to access ECS.

Cancel Next

Identity and Access Management (IAM)

- Unable to load search
- Dashboard
- Access management
 - User groups

Introducing the new IAM roles experience
We've redesigned the IAM roles experience to make it easier to use. [Let us know what you think.](#)

Role FirstApp_pultaskRole created [View role](#)

New! Securely access AWS services from your data center with IAM Roles Anywhere. [Learn more](#)

Identity and Access Management (IAM)

- Unable to load search
- Dashboard
- Access management
 - User groups
 - Users
 - Roles**
 - Policies
 - Identity providers
 - Account settings
- Access reports
 - Access analyzer
 - Archive rules
 - Analyzers
 - Settings
- Credential report
- Organization activity
- Service control policies (SCPs)

IAM > Roles > FirstApp_pultaskRole

FirstApp_pultaskRole

Allows ECS to create and manage AWS resources on your behalf.

Summary

Creation date	ARN
July 18, 2022, 17:15 (UTC-05:00)	arn:aws:iam::246460460343:role/FirstApp_pultaskRole
Last activity	Maximum session duration
None	1 hour

Permissions **Trust relationships** **Tags** **Access Advisor** **Revoke sessions**

Permissions policies (1)
You can attach up to 10 managed policies.

Policy name	Type	Description
AmazonEC2ContainerServiceRole	AWS managed	Default policy for Amazon ECS service role.

Identity and Access Management (IAM)

- Unable to load search
- Dashboard
- Access management
 - User groups
 - Users
 - Roles**
 - Policies
 - Identity providers
 - Account settings
- Access reports
 - Access analyzer
 - Archive rules
 - Analyzers
 - Settings
- Credential report
- Organization activity
- Service control policies (SCPs)

Identity and Access Management (IAM)

- Unable to load search
- Dashboard
- Access management
 - User groups
 - Users
 - Roles**
 - Policies
 - Identity providers
 - Account settings
- Access reports
 - Access analyzer
 - Archive rules
 - Analyzers
 - Settings
- Credential report
- Organization activity
- Service control policies (SCPs)

Identity and Access Management (IAM)

- Unable to load search
- Dashboard
- Access management
 - User groups
 - Users
 - Roles**
 - Policies
 - Identity providers
 - Account settings
- Access reports
 - Access analyzer
 - Archive rules
 - Analyzers
 - Settings
- Credential report
- Organization activity
- Service control policies (SCPs)

We've redesigned the IAM roles experience to make it easier to use. [Let us know what you think.](#)

IAM > Roles > FirstApp_pultaskRole

FirstApp_pultaskRole

Allows ECS to create and manage AWS resources on your behalf.

Summary

Creation date	ARN
July 18, 2022, 17:15 (UTC-05:00)	arn:aws:iam::246460460343:role/FirstApp_pultaskRole
Last activity	Maximum session duration
None	1 hour

Permissions **Trust relationships** **Tags** **Access Advisor** **Revoke sessions**

Permissions policies (1)
You can attach up to 10 managed policies.

Policy name	Type	Description
AmazonEC2ContainerServiceRole	AWS managed	Default policy for Amazon ECS service role.

Permissions boundary - (not set)
Set a permissions boundary to control the maximum permissions this role can have. This is

[Add permissions](#) [Attach policies](#) [Create inline policy](#)

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. [Learn more](#)

Visual editor **JSON** **Import managed policy**

[Expand all](#) [Collapse all](#)

Elastic Container Registry (All actions) 1 warning **Clone** **Remove**

Service Elastic Container Registry

Actions Specify the actions allowed in Elastic Container Registry [Switch to deny permissions](#)

Manual actions (add actions)

All Elastic Container Registry actions (ecr:*)

Access level

List (3 selected)

Read (15 selected)

Tagging (2 selected)

Write (19 selected)

Permissions management (4 selected)

Resources Specify repository resource ARN for the BatchGetImage and 28 more actions. [Learn more](#)

Resources Specific All resources

repository [Specify repository resource ARN for the BatchGetImage and 28 more actions.](#) [Add ARN to restrict access](#)

Elastic Container Registry (All actions) **Clone** **Remove**

Service

Actions

Resources Specific All resources

repository [Specify repository resource ARN for the BatchGetImage and 28 more actions.](#) Any in this account [Add ARN to restrict access](#)

Add ARN(s)

Amazon Resource Names (ARNs) uniquely identify AWS resources. Resources are unique to each service. [Learn more](#)

Specify ARN for repository [List ARNs manually](#)

arn:aws:ecr:*:246460460343:repository/

Region * Any

Account * 246460460343 Any

Repository name * repo-firstapp Any

repo-firstapp [Ank](#) [1](#) [Cancel](#) [Add](#) [+ Add additional permissions](#)

Character count: 39 of 10,240. The current character count includes character for all inline policies in the role: FirstApp_pulTaskRole. [Cancel](#) [Review policy](#)

Create policy 1 2

Review policy

Before you create this policy, provide the required information and review this policy.

Name* Policy_FirstAppRegister

Maximum 128 characters. Use alphanumeric and '+,-,_,@-' characters.

Summary

[Filter](#)

Service	Access level	Resource	Request condition
Allow (1 of 330 services) Show remaining 329	Full access	Multiple	None

Elastic Container Registry

* Required

[Cancel](#)

[Previous](#)

[Create policy](#)

We created a role which will have the permission to deploy the container and now we will check the permissions on the container and also will allow the role to pull

Identity and Access Management (IAM)

- Unable to load search
- Dashboard
- Access management**
 - User groups
 - Users
 - Roles**
 - Policies
 - Identity providers
 - Account settings
- Access reports**
 - Access analyzer
 - Archive rules
 - Analyzers
 - Settings
- Credential report
- Organization activity
- Service control policies (SCPs)

FirstApp_pulldtaskRole

Allows ECS to create and manage AWS resources on your behalf.

Summary

Permissions

Permissions policies (2)

Policy name	Type	Description
AmazonEC2ContainerServiceRole	AWS managed	Default policy for Amazon EC2 Container Service
Policy_FirstAppRegister	Customer inline	

Amazon Elastic Container Registry

- Private registry
- Public registry
- Repositories
- Summary
- Images**
- Permissions
- Lifecycle Policy
- Tags
- Getting started
- Documentation
- Public gallery

express-app

Images (1)

Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Vulnerabilities
latest	Image	July 12, 2022, 13:36:15 (UTC-05)	135.19	Copy URI	sha256:3cf83385bafa549...	See findings

Amazon Elastic Container Registry

- Private registry
- Public registry
- Repositories
- Summary
- Images
- Permissions**
- Lifecycle Policy
- Tags
- Getting started
- Documentation
- Public gallery

Permissions

ankita-access

Effect	Principal
Allow	-
Service principals	AWS Account IDs 246460460343

IAM Entities (2)

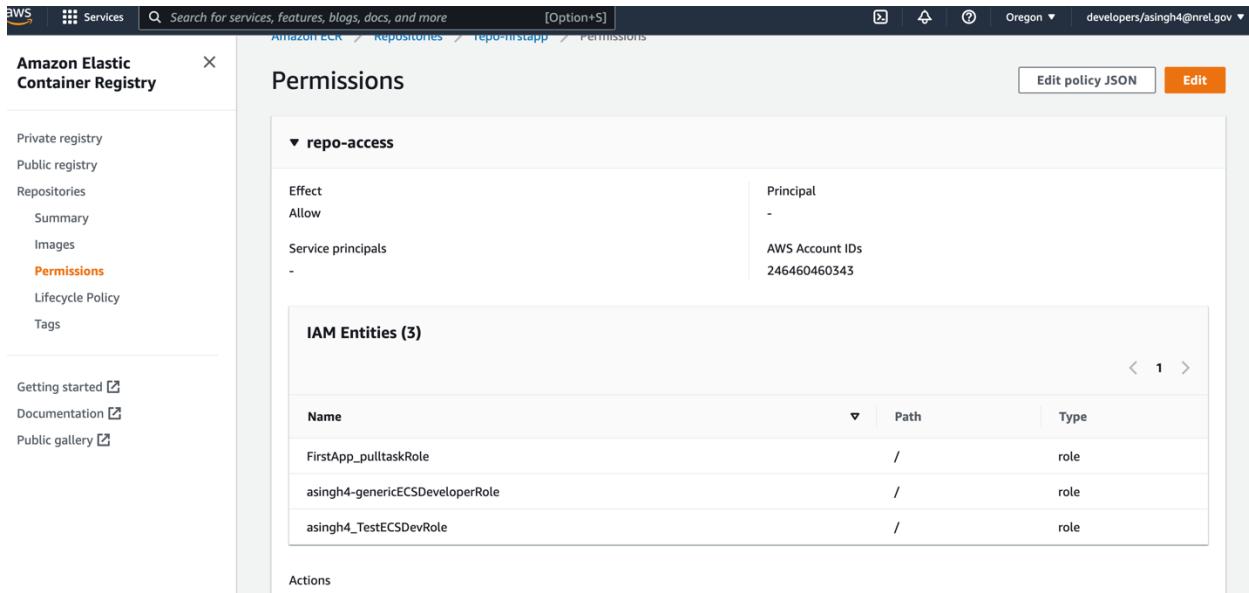
Name	Path	Type
ecs-ankita-ecr-pull-task-role	/	role
asingh4-genericECSDeveloperRole	/	role

Actions

ecr:*

<https://docs.aws.amazon.com/AmazonECR/latest/userguide/repository-policy-examples.html>

NOTE: once the “FirstApp_pulltaskRole” is created you need to make sure that the role is added in to the Elastic register’s permission section because that’s why the role has been separately created (just to access the private registry)



Amazon Elastic Container Registry

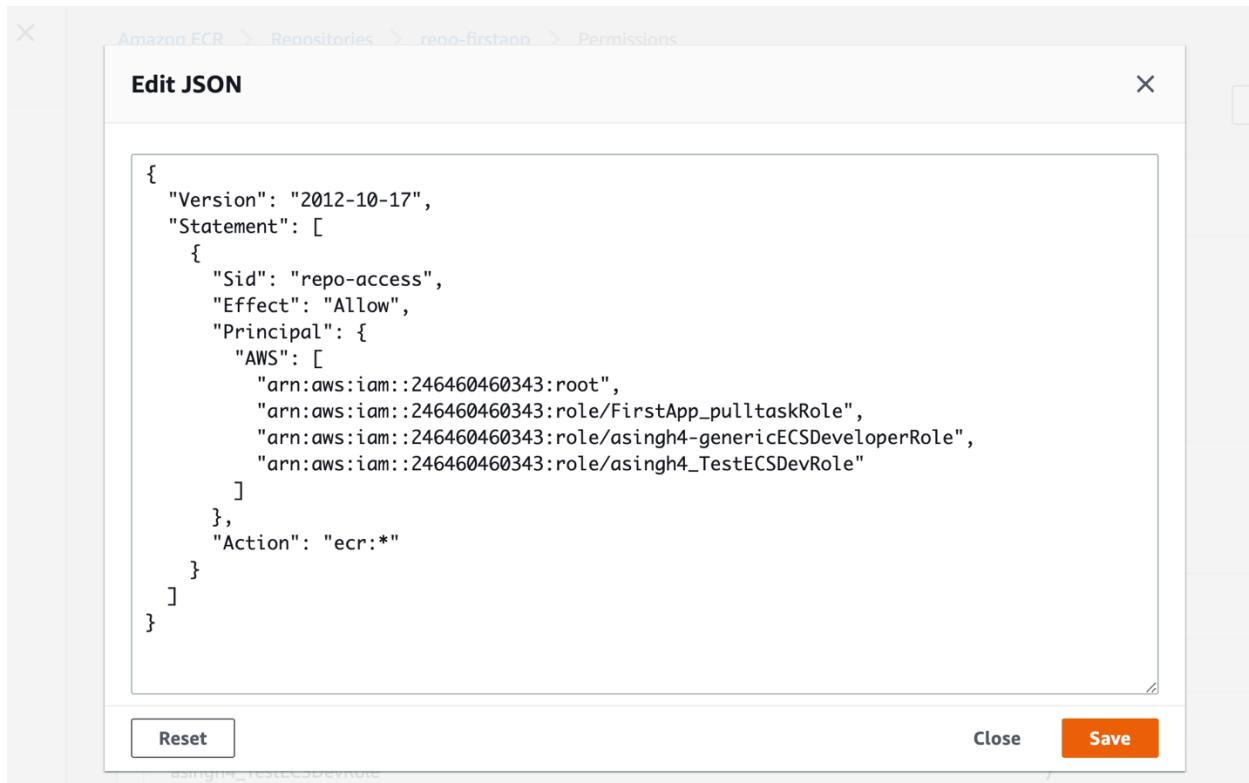
Permissions

repo-access

Effect: Allow, Principal: AWS Account IDs (246460460343)

IAM Entities (3)

Name	Path	Type
FirstApp_pulltaskRole	/	role
asingh4-genericECSDeveloperRole	/	role
asingh4_TestECSDevRole	/	role



```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "repo-access",  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": [  
          "arn:aws:iam::246460460343:root",  
          "arn:aws:iam::246460460343:role/FirstApp_pulltaskRole",  
          "arn:aws:iam::246460460343:role/asingh4-genericECSDeveloperRole",  
          "arn:aws:iam::246460460343:role/asingh4_TestECSDevRole"  
        ]  
      },  
      "Action": "ecr:*"  
    }  
  ]  
}
```

Reset Close Save

Now, we will create a Task definition and add the role which we created.

Here, we can see that the role which we created is not coming up in the drop down list so, for that we need to cross check our created role’s “trust relationships” section.

Create new Task Definition

Step 1: Select launch type compatibility

Step 2: Configure task and container definitions

Select launch type compatibility

Select which launch type you want your task definition to be compatible with based on where you want to launch your task.

FARGATE



Price based on task size

Requires network mode awsvpc

AWS-managed infrastructure, no Amazon EC2 instances to manage

EC2



Price based on resource usage

Multiple network modes available

Self-managed infrastructure using Amazon EC2 instances

EXTERNAL



Price based on instance-hours and additional charges for

Step 1: Select launch type compatibility

Step 2: Configure task and container definitions

Configure task and container definitions

A task definition specifies which containers are included in your task and how they interact with each other. You can also specify data volumes for your containers to use. [Learn more](#)

Task definition name*

Requires compatibilities* EC2

Task role

Select a role...

- None
 - ecs_task_role_project_ressto...
 - buildstock-batch-test1
 - ecs_task_role_datamoon
 - ecs_task_role_project_ressto...
 - ecs-ankita-ecr-pull-task-role
 - ecs_task_role_project_ressto...
- Windows tasks support the <default> and awsvpc network modes.

Task execution IAM role

So updating the policy's :trust relationship" JSON

IAM > Roles > FirstApp_pulldtaskRole > Edit trust policy

Edit trust policy

```

1  {
2    "Version": "2008-10-17",
3    "Statement": [
4      {
5        "Sid": "",
6        "Effect": "Allow",
7        "Principal": {
8          "Service": "ecs-tasks.amazonaws.com"
9        },
10       "Action": "sts:AssumeRole"
11     }
12   ]
13 }
```

Insert here :::

Identity and Access Management (IAM)

- Unable to load search
- Dashboard
- Access management**
 - User groups
 - Users
 - Roles**
 - Policies
 - Identity providers
 - Account settings
- Access reports**
 - Access analyzer
 - Archive rules
 - Analyzers
 - Settings
 - Credential report
 - Organization activity
 - Service control policies (SCPs)

FirstApp_pulitaskRole

Allows ECS to create and manage AWS resources on your behalf.

Summary

ARN: arn:aws:iam::246460460343:role/FirstApp_pulitaskRole

Creation date: July 18, 2022, 17:15 (UTC-05:00)

Last activity: None

Maximum session duration: 1 hour

Permissions | **Trust relationships** (highlighted with a red box) | **Tags** | **Access Advisor** | **Revoke sessions**

Trusted entities

Entities that can assume this role under specified conditions.

Edit trust policy

```

1 ~ {
2   "Version": "2008-10-17",
3   "Statement": [
4     {
5       "Sid": "",
6       "Effect": "Allow",
7       "Principal": {
8         "Service": "ecs.amazonaws.com"
9       },
10      "Action": "sts:AssumeRole"
11    }
12  ]
13 }

```

Editing it to ecs-tasks.amazonaws.com

Introducing the new IAM roles experience

We've redesigned the IAM roles experience to make it easier to use. [Let us know what you think.](#)

IAM > Roles > FirstApp_pulitaskRole > Edit trust policy

Edit trust policy

```

1 ~ {
2   "Version": "2008-10-17",
3   "Statement": [
4     {
5       "Sid": "",
6       "Effect": "Allow",
7       "Principal": {
8         "Service": "ecs-tasks.amazonaws.com"
9       },
10      "Action": "sts:AssumeRole"
11    }
12  ]
13 }

```

Identity and Access Management (IAM)

- Unable to load search
- Dashboard
- Access management**
 - User groups
 - Users
 - Roles**
 - Policies
 - Identity providers
 - Account settings
- Access reports**
 - Access analyzer
 - Archive rules
 - Analyzers
 - Settings
 - Credential report
 - Organization activity
 - Service control policies (SCPs)

FirstApp_pulitaskRole

Allows ECS to create and manage AWS resources on your behalf.

Summary

ARN: arn:aws:iam::246460460343:role/FirstApp_pulitaskRole

Creation date: July 18, 2022, 17:15 (UTC-05:00)

Last activity: None

Maximum session duration: 1 hour

Permissions | **Trust relationships** (highlighted with a red box) | **Tags** | **Access Advisor** | **Revoke sessions**

Trusted entities

Entities that can assume this role under specified conditions.

Edit trust policy

Trust policy updated

Step 1: Select launch type compatibility

Step 2: Configure task and container definitions

Configure task and container definitions

A task definition specifies which containers are included in your task and how they interact with each other. You can also specify data volumes for your containers to use. [Learn more](#)

Task definition name* FirstApp_TD

Requires compatibilities* EC2

Task role Select a role...

ecs_task_role_project_ressto...
ecs_task_role_project_ressto...
nats-dev-task-exec-role
ecs_task_role_noeltest31

Now the role is visible in DDL

Network mode

FirstApp_pulldtaskRole
ecs_task_rol FirstApp_pulldtaskRole
ecs_task_role_project_ressto...

Windows tasks support the <default> and awsvpc network modes.

Task execution IAM role

Chrome File Edit View History Bookmarks Profiles Tab Window Help Fri Jul 22 5:07 PM

Amazon Web Services (AWS) | Amazon ECR | Amazon ECS | Target groups | EC2 Management | Load balancers | EC2 Management

us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#TargetGroup:targetGroupArn=arn:aws:elasticloadbalancing:us-west-2:246460460343:targetgro... | +

Services Search for services, features, blogs, docs, and more [Option+S]

aws Oregon developers/asinh4@nrel.gov

New EC2 Experience Tell us what you think

EC2 Dashboard EC2 Global View Events Tags Limits

Instances Instances New Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances New Dedicated Hosts Scheduled Instances Capacity Reservations

Images AMIs New AMI Catalog

EC2 > Target groups > FirstApp-TG

FirstApp-TG

Actions

Details

arn:aws:elasticloadbalancing:us-west-2:246460460343:targetgroup/FirstApp-TG/d52f2d74421a8747

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-010ef166
IP address type	Load balancer	None associated	
IPv4			
Total targets	Healthy	Unhealthy	Unused
0	0	0	0
Initial			Draining
0			0

Targets Monitoring Health checks Attributes Tags

Registered targets (0)

Filter resources by property or value

Deregister Register targets

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Chrome File Edit View History Bookmarks Profiles Tab Window Help Fri Jul 22 5:18 PM

Amazon Web Services (AWS) | Target groups | EC2 Management | Amazon ECS | Create a Network Load Balancer | New Tab

us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#TargetGroup:targetGroupArn=arn:aws:elasticloadbalancing:us-west-2:246460460343:targetgro... | +

Services Search for services, features, blogs, docs, and more [Option+S]

AMIs New AMI Catalog

Elastic Block Store Volumes New Snapshots New Lifecycle Manager New

Network & Security Security Groups Elastic IPs Placement Groups Key Pairs Network Interfaces

Load Balancing Load Balancers Target Groups New

Auto Scaling Launch Configurations Auto Scaling Groups

EC2 > Target groups > FirstApp-TG

FirstApp-TG

arn:aws:elasticloadbalancing:us-west-2:246460460343:targetgroup/FirstApp-TG/d52f2d74421a8747

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-010ef166
IP address type	Load balancer	None associated	
IPv4			
Total targets	Healthy	Unhealthy	Unused
1	0	0	1
Initial			Draining
0			0

Targets Monitoring Health checks Attributes Tags

Registered targets (1)

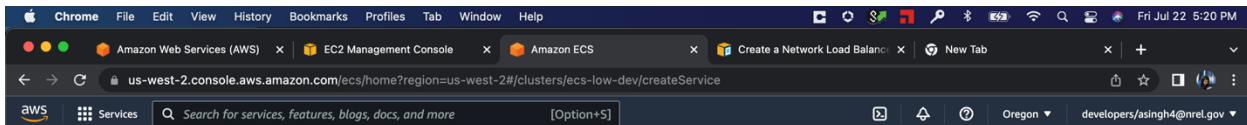
Filter resources by property or value

Deregister Register targets

Instance ID	Name	Port	Zone	Health status	Health status details
i-009b7b5539cb07798	testExpressApp-deleteme	80	us-west-2b	unused	Target group is not configured to receive traffic from the load balancer

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



Create Service

Step 1: Configure service

- Step 2: Configure network
- Step 3: Set Auto Scaling (optional)
- Step 4: Review

Configure service

A service lets you specify how many copies of your task definition to run and maintain in a cluster. You can optionally use an Elastic Load Balancing load balancer to distribute incoming traffic to containers in your service. Amazon ECS maintains the number of tasks and coordinates task scheduling with the load balancer. You can also optionally use Service Auto Scaling to adjust the number of tasks in your service.

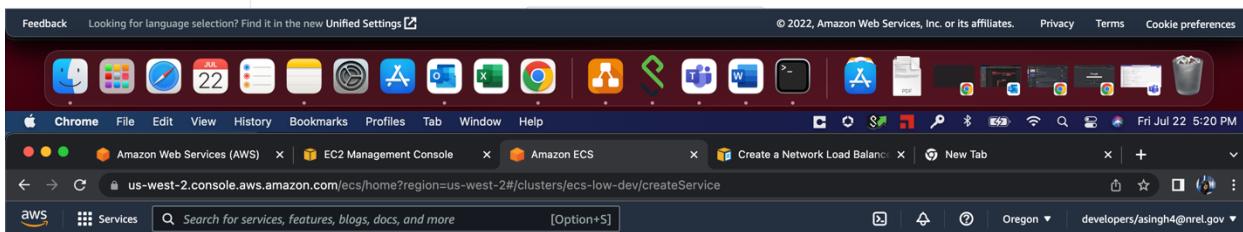
Launch type FARGATE EC2 EXTERNAL

[Switch to capacity provider strategy](#)

Task Definition Family

Revision

Cluster



Deployments

Choose a deployment option for the service.

Deployment type* Rolling update [i](#)

Blue/green deployment (powered by AWS CodeDeploy) [i](#)

This sets AWS CodeDeploy as the deployment controller for the service. A



Chrome File Edit View History Bookmarks Profiles Tab Window Help

Amazon Web Services (AWS) | EC2 Management Console | Amazon ECS | Create a Network Load Balancer | New Tab

us-west-2.console.aws.amazon.com/ecs/home?region=us-west-2#/clusters/ecs-low-dev/createTime

Services Search for services, features, blogs, docs, and more [Option+S]

Fri Jul 22 5:20 PM

Lets you customize how tasks are placed on instances within your cluster. Different placement strategies are available to optimize for availability and efficiency.

Placement Templates AZ Balanced Spread

This template will spread tasks across availability zones and within the availability zone spread tasks across instances. [Learn more](#)

Strategy: spread(attribute:ecs.availability-zone), spread(instanceId)

Task tagging configuration

Enable ECS managed tags

Propagate tags from Service

Tags

Key	Value
billingId	210001
Add key	Add value

*Required



Feedback Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Amazon Web Services (AWS) | EC2 Management Console | Amazon ECS | Create a Network Load Balancer | New Tab

us-west-2.console.aws.amazon.com/ecs/home?region=us-west-2#/clusters/ecs-low-dev/createTime

Services Search for services, features, blogs, docs, and more [Option+S]

Fri Jul 22 5:21 PM

Step 1: Configure service

Step 2: Configure network

Step 3: Set Auto Scaling (optional)

Step 4: Review

Configure network

VPC and security groups

VPC and security groups are configurable when your task definition uses the awsvpc network mode.

Health check grace period

If your service's tasks take a while to start and respond to ELB health checks, you can specify a health check grace period of up to 2,147,483,647 seconds during which the ECS service scheduler will ignore ELB health check status. This grace period can prevent the ECS service scheduler from marking tasks as unhealthy and stopping them before they have time to come up. This is only valid if your service is configured to use a load balancer.

Health check grace period

Load balancing

An Elastic Load Balancing load balancer distributes incoming traffic across the tasks running in your service. Choose an existing load balancer, or create a new one in the [Amazon EC2 console](#).

Load balancer type*

None

Your service will not use a load balancer.

Application Load Balancer

Allows containers to use dynamic host port mapping (multiple tasks allowed per container instance). Multiple services can use the same listener port on a single load balancer with rule-

Feedback Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Amazon Web Services (AWS) | EC2 Management Console | Amazon ECS | Create a Network Load Balancer | New Tab

us-west-2.console.aws.amazon.com/ecs/home?region=us-west-2#/clusters/ecs-low-dev/createTime

Services Search for services, features, blogs, docs, and more [Option+S]

Fri Jul 22 5:21 PM

Chrome File Edit View History Bookmarks Profiles Tab Window Help Fri Jul 22 5:21 PM

Amazon Web Services (AWS) | EC2 Management Console | Amazon ECS | Create a Network Load Balancer | New Tab

us-west-2.console.aws.amazon.com/ecs/home?region=us-west-2#/clusters/ecs-low-dev/createService

Services Search for services, features, blogs, docs, and more [Option+S]

Application Load Balancer

Allows containers to use dynamic host port mapping (multiple tasks allowed per container instance). Multiple services can use the same listener port on a single load balancer with rule-based routing and paths.

Network Load Balancer

A Network Load Balancer functions at the fourth layer of the Open Systems Interconnection (OSI) model. After the load balancer receives a request, it selects a target from the target group for the default rule using a flow hash routing algorithm.

Classic Load Balancer

Requires static host port mappings (only one task allowed per container instance); rule-based routing and paths are not supported.

Service IAM role AWSServiceRoleForECS

Load balancer name FirstAppLoadBalancer

Container to load balance

container_FirstApp : 5000 Remove

Production listener port* create new Enter a listener port

Production listener protocol* HTTP

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms

Chrome File Edit View History Bookmarks Profiles Tab Window Help Fri Jul 22 5:22 PM

Amazon Web Services (AWS) | EC2 Management Console | Amazon ECS | Create a Network Load Balancer | New Tab

us-west-2.console.aws.amazon.com/ecs/home?region=us-west-2#/clusters/ecs-low-dev/createService

Services Search for services, features, blogs, docs, and more [Option+S]

Container to load balance

container_FirstApp : 5000 Remove

Production listener port* 80:HTTP

Production listener protocol* HTTP

Target group name FirstApp-TG

Target group protocol HTTP

Target type instance

Path pattern / Evaluation order default

Health check path /

Additional health check options can be configured in the ELB console after you create your service.

App Mesh

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Chrome File Edit View History Bookmarks Profiles Tab Window Help Fri Jul 22 5:22 PM

Amazon Web Services (AWS) | EC2 Management Console | Amazon ECS | Create a Network Load Balancer | New Tab

us-west-2.console.aws.amazon.com/ecs/home?region=us-west-2#/clusters/ecs-low-dev/createService

Services Search for services, features, blogs, docs, and more [Option+S]

Container to load balance

container_FirstApp : 5000 Remove

Production listener port* 80:HTTP

Production listener protocol* HTTP

Target group name FirstApp-TG

Target group protocol HTTP

Target type instance

Path pattern / Evaluation order default

Health check path /

Additional health check options can be configured in the ELB console after you create your service.

App Mesh

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Amazon Web Services (AWS) | EC2 Management Console | Amazon ECS | Create a Network Load Balancer | New Tab

us-west-2.console.aws.amazon.com/ecs/home?region=us-west-2#/clusters/ecs-low-dev/createService

Services | Search for services, features, blogs, docs, and more | [Option+S]

Fri Jul 22 5:23 PM | Oregon | developers/asingh4@nrel.gov

Service IAM role ⓘ

Load balancer name ⓘ

Container to load balance

container_FirstApp : 5000

Remove ⓘ

Production listener port* ⓘ ⓘ

Production listener protocol* ⓘ

Target group name ⓘ ⓘ

Target group protocol ⓘ

Target type ⓘ

Path pattern

Path pattern: The first path pattern for a listener is the default path (/), which accepts all traffic that

Feedback | Looking for language selection? Find it in the new Unified Settings ⓘ

© 2022, Amazon Web Services, Inc. or its affiliates. | Privacy | Terms | Cookie preferences

Chrome | File | Edit | View | History | Bookmarks | Profiles | Tab | Window | Help

Amazon Web Services (AWS) | EC2 Management Console | Amazon ECS | Create a Network Load Balancer | New Tab

us-west-2.console.aws.amazon.com/ecs/home?region=us-west-2#/clusters/ecs-low-dev/createService

Services | Search for services, features, blogs, docs, and more | [Option+S]

Fri Jul 22 5:24 PM | Oregon | developers/asingh4@nrel.gov

Production listener protocol*

Target group name ⓘ

Target group protocol ⓘ

Target type ⓘ

Path pattern Evaluation order

Health check path

Additional health check options can be configured in the ELB console after you create your service.

App Mesh

- To use your service with App Mesh, you must
- Ensure your task definition is configured properly. Edit your task definition if you haven't done this.
 - Set up your service to use Service Discovery.

Service discovery (optional)

Service discovery uses Amazon Route 53 to create a namespace for your service, which allows it to be discoverable via DNS.

Create Service

Step 1: Configure service

Step 2: Configure network

Step 3: Set Auto Scaling (optional)

Step 4: Review

Review	
Cluster	ecs-low-dev
Launch type	EC2
Task Definition	FirstApp_TD:1
Service name	Service-FirstApp
Service type	REPLICAS
Number of tasks	1
Minimum healthy percent	100
Maximum percent	200
Deployment circuit breaker	Disabled

Configure network

Container Name: container_FirstApp

Container Port: 5000

ELB Name: FirstAppLoadBalancer

Target Group: FirstApp-TG

Health Check Path: /

Listener Port: 80

Path-pattern: /

Service Role: AWSServiceRoleForECS

Set Auto Scaling (optional)

not configured

Cancel **Previous** **Create Service**

Feedback Looking for language selection? Find it in the new Unified Settings 

© 2022, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Ter](#)



Amazon Web Services (AWS) | EC2 Management Console | Amazon ECS | Create a Network Load Balancer | New Tab

Search for services, features, blogs, docs, and more [Option+S]

aws Services Oregon developers/asinh4@nrel.gov

Launch Status

ECS Service status - 1 of 1 completed

Create Load Balancer

Create Service

Create service: Service-FirstApp



Service created
Service created. Tasks will start momentarily. View: [Service-FirstApp](#)

Additional integrations you can connect to your ECS service

Code Pipeline

Setup a CI/CD process from your service. You can build from source or have an ECR repository as the source for your deployment.

[Create a pipeline](#)

Back [View Service](#)

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms

Amazon Web Services (AWS) | EC2 Management Console | Amazon ECS | Create a Network Load Balancer | New Tab

us-west-2.console.aws.amazon.com/ecs/home?region=us-west-2#/clusters/ecs-low-dev/services/Service-FirstApp/tasks

aws Services Oregon developers/asinh4@nrel.gov

Service : Service-FirstApp

Clusters

Task Definitions

Account Settings

Amazon EKS

Clusters

Amazon ECR

Repositories

AWS Marketplace

Discover software

Subscriptions

Cluster: ecs-low-dev

Status: ACTIVE

Task definition: FirstApp_TD:1

Service type: REPLICA

Launch type: EC2

Service role: AWSServiceRoleForECS

Created By: arn:aws:iam::246460460343:role/aws-reserved/sso.amazonaws.com/us-west-2/AWSReservedSSO_developers_41783ae43c8789c9

Desired count: 1

Pending count: 0

Running count: 1

Update Delete

Details Tasks Events Auto Scaling Deployments Metrics Tags

Last updated on July 22, 2022 5:26:52 PM (0m ago)

Task status: Running Stopped

Filter in this page Page size 50

Task	Task Definition	Last status	Desired status	Group	Launch type
367823427da14453a534...	FirstApp_TD:1	RUNNING	RUNNING	service:Service-FirstApp	EC2

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Amazon Web Services (AWS) | EC2 Management Console | Amazon ECS | Create a Network Load Balancer | New Tab

us-west-2.console.aws.amazon.com/ecs/home?region=us-west-2#/clusters/ecs-low-dev/services/Service-FirstApp/tasks

Task	Task Definition	Last status	Desired status	Group	Launch type
367823427da14453a534...	FirstApp_TD:1	RUNNING	RUNNING	service:Service-FirstApp	EC2

OutputKey: OutputValue

Output

```
bash-3.2$ curl -H: "Host: Service-FirstApp.ace.nrel.gov" internal-dev-low-alb-821729687.us-west-2.elb.amazonaws.com
curl: (3) URL using bad/illegal format or missing URL
This is my first applicationbash-3.2$
```

