

## Part 1 – Setting Up a Flask App Image

### Build Flask App Image Using Dockerfile

Create a dockerfile with instructions on how to set up the Flask application:

```
Cadens-MacBook-Pro:ecs-practice cadenhong$ cat dockerfile
FROM python:latest

RUN apt update && apt install git -y

WORKDIR /flask-app

RUN git clone https://github.com/cadenhong/kl_wk14_deployment3.git

WORKDIR ./kl_wk14_deployment3

RUN pip install -r requirements.txt

EXPOSE 5000

ENTRYPOINT FLASK_APP=application flask run --host=0.0.0.0
```

Build an image using the dockerfile – image is called flaskapp:v1

```
Cadens-MacBook-Pro:ecs-practice cadenhong$ docker build -t flaskapp:v1 .
[+] Building 24.0s (10/10) FINISHED
=> [internal] load build definition from Dockerfile                                0.1s
=> => transferring dockerfile: 329B                                              0.0s
=> [internal] load .dockerignore                                                  0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/python:latest                 0.0s
=> CACHED [1/6] FROM docker.io/library/python:latest                           0.0s
=> [2/6] RUN apt update && apt install git -y                                    6.8s
=> [3/6] WORKDIR /flask-app                                                      0.0s
=> [4/6] RUN git clone https://github.com/cadenhong/kl_wk14_deployment3.git      2.8s
=> [5/6] WORKDIR ./kl_wk14_deployment3                                           0.1s
=> [6/6] RUN pip install -r requirements.txt                                    13.2s
=> exporting to image                                                            0.8s
=> => exporting layers                                                            0.8s
=> => writing image sha256:cd045b85584a45ecba5df14d14e821b73ce914a058e7bb8f43338bf019a38188 0.0s
=> => naming to docker.io/library/flaskapp:v1                                   0.0s
```

Run image on a container and map host port 8321 to container port 5000:

```
Cadens-MacBook-Pro:ecs-practice cadenhong$ docker run -p 8321:5000 flaskapp:v1
* Serving Flask app 'application'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.2:5000
Press CTRL+C to quit
172.17.0.1 - - [07/Nov/2022 20:12:14] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [07/Nov/2022 20:12:14] "GET /static/bootstrap.min.js HTTP/1.1" 200 -
172.17.0.1 - - [07/Nov/2022 20:12:14] "GET /static/bootstrap.min.css HTTP/1.1" 200 -
172.17.0.1 - - [07/Nov/2022 20:12:14] "GET /static/jquery-3.3.1.slim.min.js HTTP/1.1" 200 -
172.17.0.1 - - [07/Nov/2022 20:12:14] "GET /static/popper.min.js HTTP/1.1" 200 -
```

Check by navigating to <http://localhost:8321>:

URL Shortener API [New URL](#)

### Website

Short Name

Website URL

[Shorten](#)

### File

Short Name

Website URL

[Browse...](#) No file selected.

[Shorten](#)

**Codes you have created:**

- [goog](#)

## Push the Flask App Image to Docker Remote Repository

Tag the flaskapp image first using **docker tag** command:

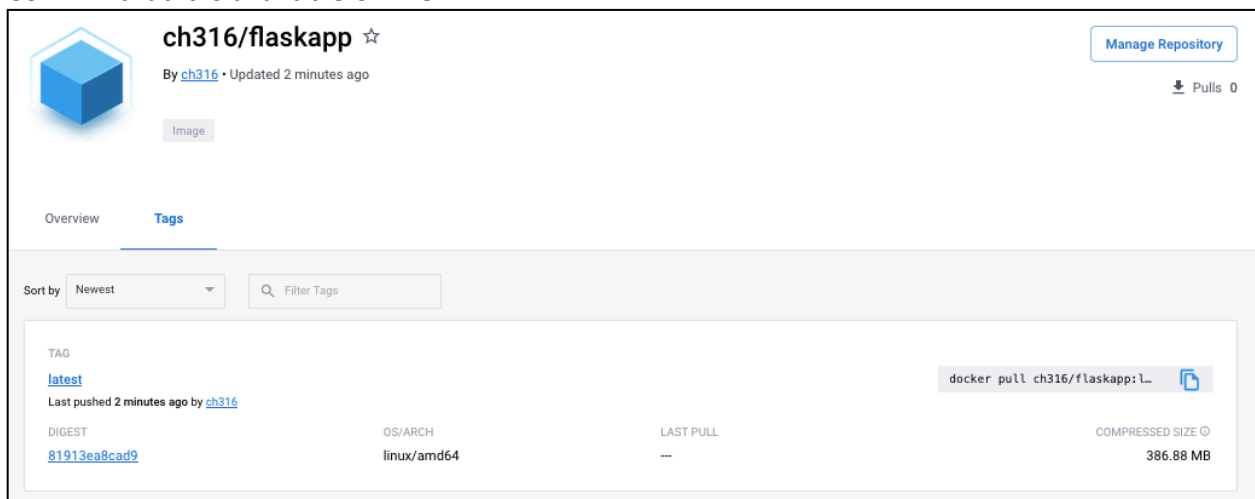
```
[Cadens-MacBook-Pro:ecs-practice cadenhong$ docker tag flaskapp:v1 ch316/flaskapp:latest
[Cadens-MacBook-Pro:ecs-practice cadenhong$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ch316/flaskapp	latest	cd045b85584a	7 minutes ago	1GB
flaskapp	v1	cd045b85584a	7 minutes ago	1GB

Push the newly created image to remote repository with **docker push** command:

```
Cadens-MacBook-Pro:ecs-practice cadenhong$ docker push ch316/flaskapp
Using default tag: latest
The push refers to repository [docker.io/ch316/flaskapp]
3df8674cd1f9: Pushed
5f70bf18a086: Pushed
e75cfad7ba91: Pushed
cb749602d04d: Pushed
0c1aa7e0f4e9: Pushed
6f6e69c2c592: Mounted from ch316/pyappdockerized
53b8bfee7a0a: Mounted from ch316/pyappdockerized
5b3f1ed98915: Mounted from ch316/pyappdockerized
6b183c62e3d7: Mounted from ch316/pyappdockerized
882fd36bfd35: Mounted from ch316/pyappdockerized
d1dec9917839: Mounted from ch316/pyappdockerized
d38adf39e1dd: Mounted from ch316/pyappdockerized
4ed121b04368: Mounted from ch316/pyappdockerized
d9d07d703dd5: Mounted from ch316/pyappdockerized
latest: digest: sha256:81913ea8cad9200beaaafe51d238776273fa2757170eb40120824f6596a249b7 size: 3266
```

Confirm that it is available online:



ch316/flaskapp ☆

By [ch316](#) · Updated 2 minutes ago

Manage Repository

Pulls 0

Overview **Tags**

Sort by Newest Filter Tags

TAG

[latest](#)

Last pushed 2 minutes ago by [ch316](#)

docker pull ch316/flaskapp:latest

DIGEST

[81913ea8cad9](#)

OS/ARCH

linux/amd64

LAST PULL

---

COMPRESSED SIZE

386.88 MB

## Part 2 – Setting Up AWS ECS

### IAM – ECS Role Creation

Select *AWS Services* for Trusted Entity Type and *Elastic Container Service (Elastic Container Service Task)* for Use Case:

IAM > Roles > Create role

Step 1  
**Select trusted entity**

Step 2  
Add permissions

Step 3  
Name, review, and create

### Select trusted entity [Info](#)

**Trusted entity type**

- ☒ **AWS service**  
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- ☐ **AWS account**  
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- ☐ **Web identity**  
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- ☐ **SAML 2.0 federation**  
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- ☐ **Custom trust policy**  
Create a custom trust policy to enable others to perform actions in this account.

**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](#)

Select *AmazonECSTaskExecutionRolePolicy* for permissions:

IAM > Roles > Create role

Step 1  
Select trusted entity

Step 2  
**Add permissions**

Step 3  
Name, review, and create

## Add permissions [Info](#)

**Permissions policies** (Selected 1/779) [Info](#)

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

Filter policies by property or policy name and press enter. 1 match < 1 > ⚙

"ecstask" X Clear filters

<input checked="" type="checkbox"/>	Policy name <a href="#">↗</a>	Type	Description
<input checked="" type="checkbox"/>	<a href="#">+</a> AmazonECSTaskExecutionRolePolicy	AWS m...	Provides access to other AWS service res...

► **Set permissions boundary - optional** [Info](#)

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

Cancel Previous **Next**

Enter Role Details and Create Role:

IAM > Roles > Create role

Step 1  
Select trusted entity

Step 2  
Add permissions

Step 3  
**Name, review, and create**

## Name, review, and create

### Role details

**Role name**  
Enter a meaningful name to identify this role.

ecsTaskEx

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

**Description**  
Add a short explanation for this role.

Allows ECS tasks to call AWS services on your behalf.

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

## ECS Cluster

Navigate to ECS > Cluster > Create Cluster and select Networking Only to use AWS Fargate:

### Create Cluster

**Step 1: Select cluster template**  
Step 2: Configure cluster

#### Select cluster template

The following cluster templates are available to simplify cluster creation. Additional configuration and integrations can be added later.

**Networking only ⓘ**  
Resources to be created:  
Cluster  
VPC (optional)  
Subnets (optional)  
  
 ⓘ For use with either AWS Fargate (Windows/Linux) or with External instance capacity.

**EC2 Linux + Networking**  
Resources to be created:  
Cluster  
VPC  
Subnets  
Auto Scaling group with Linux AMI

**EC2 Windows + Networking**  
Resources to be created:  
Cluster  
VPC  
Subnets  
Auto Scaling group with Windows AMI

\*Required

[Cancel](#) [Next step](#)


Specify Cluster Name; since I will be using the custom VPC created from previous weeks, I will leave Create as unchecked:

## Create Cluster

[Step 1: Select cluster template](#)

**Step 2: Configure cluster**

### Configure cluster

**Cluster name\***  

#### Networking

Create a new VPC for your cluster to use. A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Fargate tasks.

**Create VPC** ☐ Create a new VPC for this cluster

#### Tags

Key	Value
<input data-bbox="505 705 774 739" type="text" value="Add key"/>	<input data-bbox="812 705 1317 739" type="text" value="Add value"/>

#### CloudWatch Container Insights

CloudWatch Container Insights is a monitoring and troubleshooting solution for containerized applications and microservices. It collects, aggregates, and summarizes compute utilization such as CPU, memory, disk, and network; and diagnostic information such as container restart failures to help you isolate issues with your clusters and resolve them quickly. [Learn more](#)

**CloudWatch Container Insights** ☐ Enable Container Insights

**\*Required** [Cancel](#) [Previous](#) [Create](#)


## Create Cluster:

### Launch status

Your container instances are launching, and it may take a few minutes until they are in the running state and ready to access. Usage hours on your new container instances start immediately and continue to accrue until you stop or terminate them.

[Back](#) [View Cluster](#)

ECS status - 1 of 1 complete **wk18-flaskcluster**



**ECS cluster**  
ECS Cluster wk18-flaskcluster successfully created

## ECS Task Definition


Navigate to ECS > Task Definition > Create Task Definition and select *Fargate*:

### 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 other AWS services used  
Self-managed on-premise infrastructure with ECS  
Anywhere

8



Set task definition name, select the created IAM role for Task Role, and select *Linux* for OS:

### 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\*** FARGATE

**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.

**Operating system family**  ⓘ

**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 can create one for you.

**Task execution role**  ⓘ

Set Task memory and CPU size:

### Task size

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 (GB)**  ⓘ

The valid memory range for 0.5 vCPU is: 1GB - 4GB, in 1GB increments.

**Task CPU (vCPU)**  ⓘ

The valid CPU range for 1GB memory is: 0.25 vCPU - 0.5 vCPU.

**Task memory maximum allocation for container memory reservation**

0 1024 shared of 1024 MiB



**Task CPU maximum allocation for containers**

0 512 shared of 512 CPU units

Click Add Container:

Container definitions

Add container

Containe...	Image	Hard/Sof...	CPU Units	GPU	Inference...	Essential	
	url-sh...	ch316/flas...	--/128			true	

Enter name of container, name of image created from Part 1 under Image, default memory limit to 128, and map container port 5000:

Add container

Standard

Container name\*

url-shortener

Image\*

ch316/flaskapp:latest

Private repository authentication\*

☐

Memory Limits (MiB)

Soft limit

128

+ Add Hard 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 definitions.  
ECS recommends 300-500 MiB as a starting point for web applications.

Port mappings

Container port

5000

Protocol

tcp

+ Add port mapping

Port mappings allow containers to access ports on the host container instance to send or receive traffic.

Keep all other configurations as default and create task definition:

### Launch Status

Task definition status - 2 of 2 completed

Create Task Definition: wk18-flaskapp

wk18-flaskapp succeeded

Create CloudWatch Log Group

✔

CloudWatch Log Group created

CloudWatch Log Group [/ecs/wk18-flaskapp](#)

Back

View task definition

## ECS Task (i.e. Container)

Navigate to the cluster created from earlier, then click on Tasks tab and select Run New Task:

**Clusters** > wk18-flaskcluster

### Cluster : wk18-flaskcluster

**Update Cluster****Delete Cluster**

Get a detailed view of the resources on your cluster.

**Cluster ARN**arn:aws:ecs:us-east-1:108026381256:cluster/wk18-flaskcluster

**Status**ACTIVE

**Registered container instances**0

**Pending tasks count**0 Fargate, 0 EC2, 0 External

**Running tasks count**0 Fargate, 0 EC2, 0 External

**Active service count**0 Fargate, 0 EC2, 0 External

**Draining service count**0 Fargate, 0 EC2, 0 External

ServicesTasksECS InstancesMetricsScheduled TasksTagsCapacity Providers

**Run new Task**StopStop AllActions

Last updated on November 7, 2022 3:38:37 PM (0m ago)

Desired task status:RunningStopped

Filter in this pageLaunch typeALL

	Task	Task defi...	Containe...	Last stat...	Desired ...	Started at	Started By	Group	Launch t...	Platform ...
No results										

Select Fargate as launch type and Linux as OS Family:

## Run Task

Select the cluster to run your task definition on and the number of copies of that task to run. To apply container overrides or target particular container instances, click Advanced Options.

**Launch type** ☒ FARGATE ☐ EC2 ☐ EXTERNAL ⓘ

AWS Fargate is migrating service quotas from the current Amazon ECS task count-based quotas to vCPU-based quotas. [To learn more, refer to the AWS Fargate FAQs.](#)

[Switch to capacity provider strategy](#) ⓘ

**Operating system family** Linux ▼

**Task Definition** Family  
wk18-flaskapp ▼

Revision  
1 (latest) ▼

**Platform version** LATEST ▼ ⓘ

**Cluster** wk18-flaskcluster ▼

**Number of tasks** 1

**Task Group** ⓘ

Select a VPC and a subnet to use – make sure to set security group correctly with right ports:

### VPC and security groups

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

Cluster VPC\*

vpc-088f820674463f327 (172.25.0.0/16)...

Subnets\*

subnet-0c76d5c8ca9df0aa6  
(172.25.0.0/18) | PUBLIC-REVIEW-A - us-e  
ast-1a  
assign ipv6 on creation: Disabled

Security groups\*

sg-01f7f2b6b641a48af

Edit

Auto-assign public  
IP

ENABLED

For security group, select existing security group that has port 5000 open:

### Configure security groups ✕

A security group is a set of firewall rules that control the traffic for your task. On this page, you can add rules to allow specific traffic to reach your task, or you can choose to use an existing security group. [Learn more.](#)

**Assigned security groups**
☐ Create new security group  
☒ Select existing security group

#### Existing security groups

All existing security groups for the VPC of this cluster are listed below.

1 selected

< 0-0 >

<input type="checkbox"/>	Security group ID	Name	Description	Actions
<input type="checkbox"/>	sg-0e90e26cad7a95105	SSH/PING	launch-wizard-7 creat...	<a href="#">Copy to new</a>
<input type="checkbox"/>	sg-0f93842ed9b854f83	deployment3	ports 22 and 5000	<a href="#">Copy to new</a>
<input type="checkbox"/>	sg-02c9a94accae4de17	wk18-p-6480	2022-11-05T18:50:43....	<a href="#">Copy to new</a>
<input type="checkbox"/>	sg-036ca978336ca0e24	pri-test	launch-wizard-1 creat...	<a href="#">Copy to new</a>
<input type="checkbox"/>	sg-0867560432ae98c46	default	default VPC security g...	<a href="#">Copy to new</a>
<input type="checkbox"/>	sg-0dba7c7fe746f40b4	ALB-Flask	Port 80 open	<a href="#">Copy to new</a>
<input checked="" type="checkbox"/>	sg-01f7f2b6b641a48af	Flask-App	For the Flask App	<a href="#">Copy to new</a>

#### Inbound rules for selected security groups

Security group ID	Type	Protocol	Port range	Source
sg-01f7f2b6b641a4...	SSH	TCP	22	0.0.0.0/0
sg-01f7f2b6b641a4...	customtcp	TCP	5000	0.0.0.0/0
sg-01f7f2b6b641a4...	All ICMP Rule - IPV4	ICMP	all	0.0.0.0/0

Confirm creation of task:

✓

**Created tasks successfully**

Task ids : ["wk18-flaskcluster/e001c1b8597142e8a7773128e96536f8"]

✕

Clusters > wk18-flaskcluster > Task: e001c1b8597142e8a7773128e96536f8

## Task : e001c1b8597142e8a7773128e96536f8

Run more like this Stop

Details Tags Logs

**Cluster** wk18-flaskcluster

**Launch type** FARGATE

**Platform version** 1.4.0

**Task definition** wk18-flaskapp:1

**Group** family:wk18-flaskapp

**Task role** ecsTaskEx

**Last status** RUNNING

**Desired status** RUNNING

**Created at** 2022-11-07 15:42:57 -0500

**Started at** 2022-11-07 15:43:38 -0500

### Network

**Network mode** awsvpc

**ENI Id** eni-04de4078fc134b6f1

**Subnet Id** subnet-0c76d5c8ca9df0aa6

**Private IP** 172.25.38.63

**Public IP** 3.82.240.254

**Mac address** 12:b6:85:cc:58:df

### Containers

Last updated on November 7, 2022 3:43:55 PM (0m ago) ↻ ⓘ

Name	Container Runtime ID	Status	Image	Image Di...	CPU ...	Hard...	Esse...	Reso...
url-shortener	e001c1b8597142e8a7...	RUN...	ch316/flaskapp:latest		--	--/128	true	e77fd...

Navigate to <http://3.82.240.254:5000/> to check that the container is running:

← → ↺ 3.82.240.254:5000 ☆ 📌 📄 📱 📧 📧 📧 📧 📧 📧

## URL Shortener

API [New URL](#)

### Website

Short Name

Website URL

Shorten

### File

Short Name

Website URL

[Browse...](#) No file selected.

Shorten

### Codes you have created:

- kura



Check logs as well:

[Clusters](#) > [wk18-flaskcluster](#) > Task: e001c1b8597142e8a7773128e96536f8

Task : e001c1b8597142e8a7773128e96536f8

Run more like this

Stop

Details

Tags

Logs

Last updated on November 7, 2022 3:43:55 PM (1m ago)

Filter logs

×

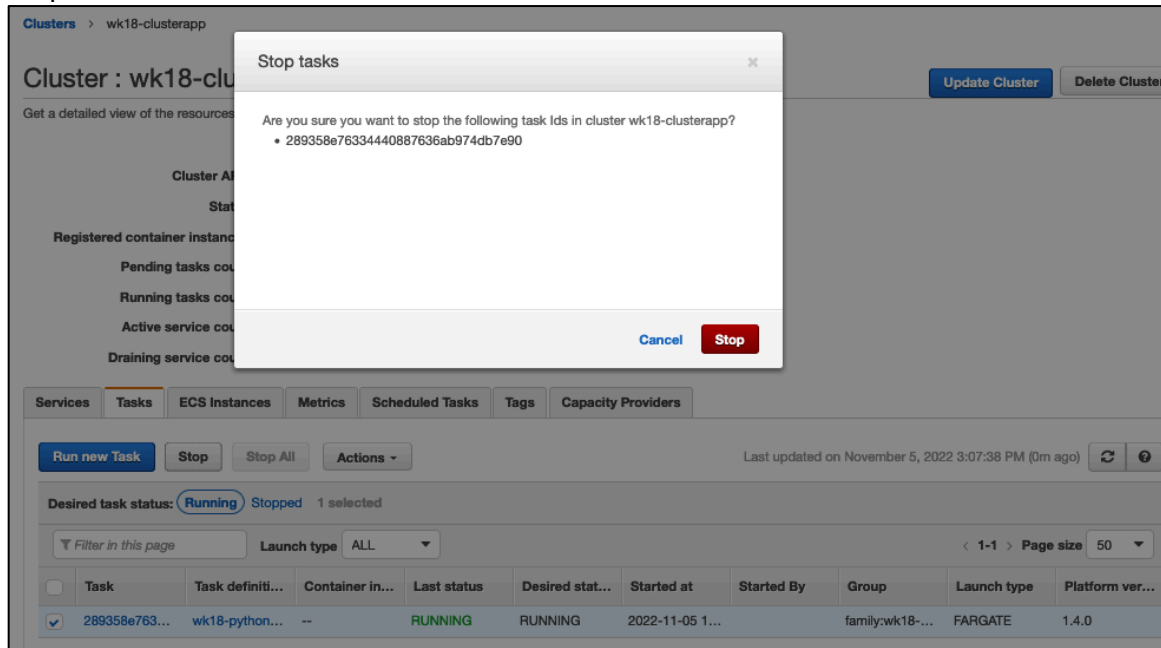
All30s5m1h6h1d< 1-7 >

1w

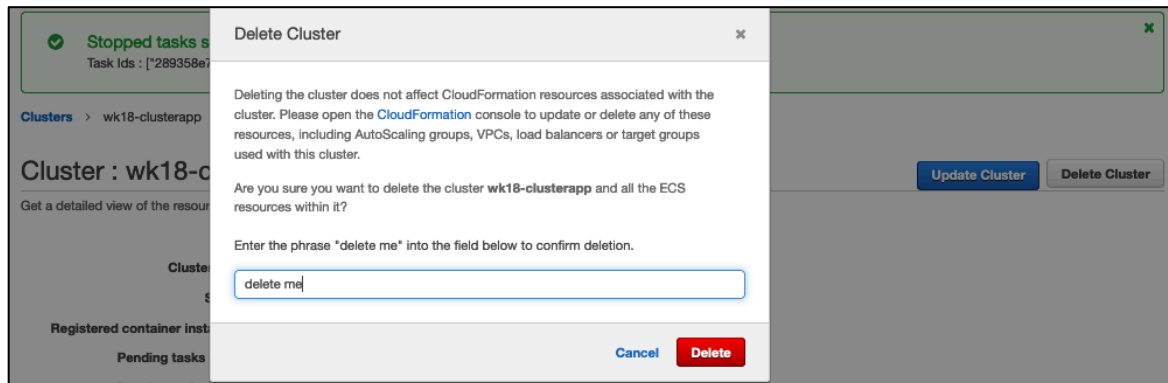
Timestamp (UTC+00:00) ...	Message
▶ 2022-11-07 15:43:39	[31m[1mWARNING: This is a development server. Do not use it in a production deployment. Use a production WS...
▶ 2022-11-07 15:43:39	* Running on all addresses (0.0.0.0)
▶ 2022-11-07 15:43:39	* Running on http://127.0.0.1:5000
▶ 2022-11-07 15:43:39	* Running on http://172.25.38.63:5000
▶ 2022-11-07 15:43:39	[33mPress CTRL+C to quit[0m
▶ 2022-11-07 15:43:39	* Serving Flask app 'application'
▶ 2022-11-07 15:43:39	* Debug mode: off

## Tearing Down Resources in ECS

### Stop the Task:



### Delete Cluster:



### Deregister Task Definition:

