使用fluentbit收集日志，通过firehose打入S3

参考文档https://aws.amazon.com/cn/blogs/china/easy-aws-fargate-container-log-processing-with-aws-firelens/

https://github.com/aws-samples/amazon-ecs-firelens-examples

包含如下步骤

1. 创建app docker -- 使用firelens输出日志
2. 创建fluentbit docker -- 收集app docker中的指定目录下的日志，输出至firehose
3. 创建ecs task definition
4. 创建firehose datastream，打入S3

1. 创建app docker

使用如下docker配置，上传ecr构建app镜像

Dockerfile:

```
FROM alpine
COPY testlog.sh /bin/
RUN chmod +x /bin/testlog.sh
ENTRYPOINT ["/bin/testlog.sh"]
```

testlog.sh

```
#!/bin/sh

while :
do
   echo "{\"server_date\":\"2020-01-19\",\"hostname\":\"ip-172-31-43-24.cn-northwest-1.compute.internal\",\"pid\":5404,\"method\":\"POST\",\"clientIP\":\"10.11.12.13\",\"countryCode\":\"ID\",\"url\":\"/v1/mail/list\",\"status\":\"200\",\"latency\":7,\"length\":24,\"userId\":9536605,\"code\":20001}" >> /data/access.log
   echo "{\"server_date\":\"2020-01-19\",\"hostname\":\"ip-172-31-43-24.cn-northwest-1.compute.internal\",\"pid\":1000,\"method\":\"GET\",\"clientIP\":\"20.21.22.23\",\"countryCode\":\"ID\",\"url\":\"/v1/mail/list\",\"status\":\"500\",\"latency\":10,\"length\":12,\"userId\":1010001,\"code\":10001}" >> /data/error.log
sleep 10
done
```

2. 创建fluentbit docker

使用如下docker配置，上传ecr构建fluentbit镜像。因为需要使用自定义的fluentbit config文件，所以需要自定义fluentbit镜像，其中包含本地配置文件extra.conf。后面需要告诉fluentbit镜像使用该本地file作为配置文件，用于修改fluentbit使用的缺省配置文件（/fluent-bit/etc/fluent-bit.conf）

Dockerfile:

```
FROM amazon/aws-for-fluent-bit:latest
COPY extra.conf /extra.conf
```

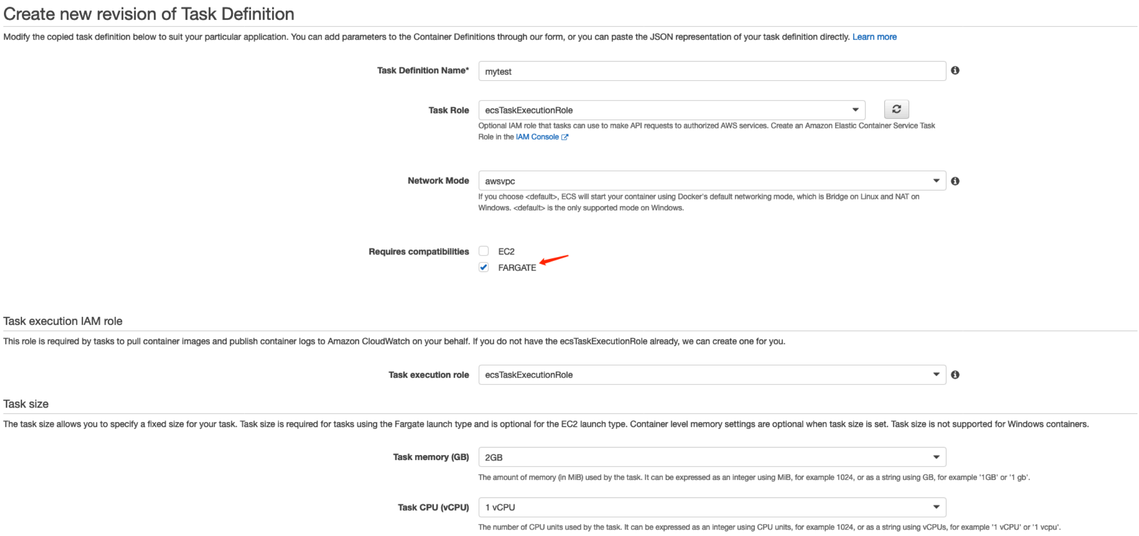
Extra.conf

```
[SERVICE]
    Parsers_File /fluent-bit/parsers/parsers.conf
    Flush 1
    Grace 30

[INPUT]
    Name tail
    Path /data/access.log
    Tag access

[FILTER]
    Name parser
    Match *
    Key_Name log
    Parser json
    Reserve_Data True

[OUTPUT]
    Name firehose
    Match access
    region cn-northwest-1
    delivery_stream fluentbit-access
```

Input 为使用tail的方式收集/data/access.log内容，输出到名为fluentbit-access的firehose datasream中

3. 创建ecs task definition

- 根据目前使用模式，选择fargate



Create new revision of Task Definition

Modify the copied task definition below to suit your particular application. You can add parameters to the Container Definitions through our form, or you can paste the JSON representation of your task definition directly. Learn more

Task Definition Name* mytest

Task Role ecsTaskExecutionRole

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 awsvpc

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

Requires compatibilities □ EC2 ☑ FARGATE

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 ecsTaskExecutionRole

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 launch type. Container level memory settings are optional when task size is set. Task size is not supported for Windows containers.

Task memory (GB) 2GB

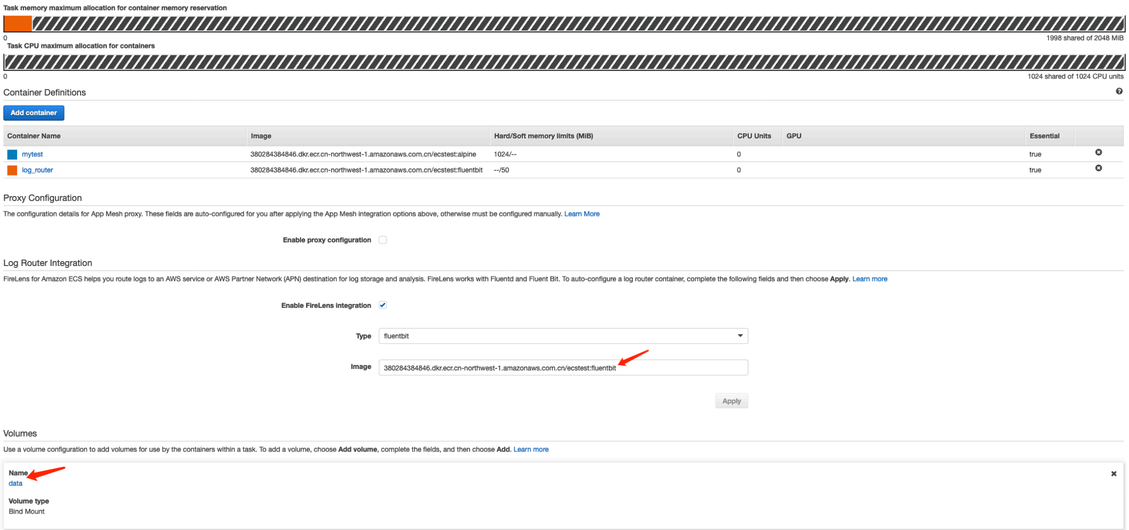
The amount of memory (in MiB) used by the task. It can be expressed as an integer using MiB, for example 1024, or as a string using GB, for example '1GB' or '1 gb'.

Task CPU (vCPU) 1 vCPU

The number of CPU units used by the task. It can be expressed as an integer using CPU units, for example 1024, or as a string using vCPUs, for example '1 vCPU' or '1 vcpu'.

- 选择enable firelens integration，image使用上面创建的fluentbit 镜像

  创建名字为data的volume，用于两个container共享，app container将log输出到该volume，fluentbit从该volume读取日志发至firehose



Task memory maximum allocation for container memory reservation

0 1998 shared of 2048 MiB

Task CPU maximum allocation for containers

0 1024 shared of 1024 CPU units

Container Definitions

Container Name	Image	Hard/Soft memory limits (MiB)	CPU Units	GPU	Essential
mytest	380284384846.dkr.ecr.cn-northwest-1.amazonaws.com.cn/ecstest:alpine	1024/--	0		true
log_router	380284384846.dkr.ecr.cn-northwest-1.amazonaws.com.cn/ecstest:fluentbit	--/50	0		true

Proxy Configuration

The configuration details for App Mesh proxy. These fields are auto-configured for you after applying the App Mesh integration options above, otherwise must be configured manually. Learn More

Enable proxy configuration □

Log Router Integration

FireLens for Amazon ECS helps you route logs to an AWS service or AWS Partner Network (APN) destination for log storage and analysis. FireLens works with Fluentd and Fluent Bit. To auto-configure a log router container, complete the following fields and then choose **Apply**. Learn more

Enable FireLens integration ☑

Type fluentbit

Image 380284384846.dkr.ecr.cn-northwest-1.amazonaws.com.cn/ecstest:fluentbit

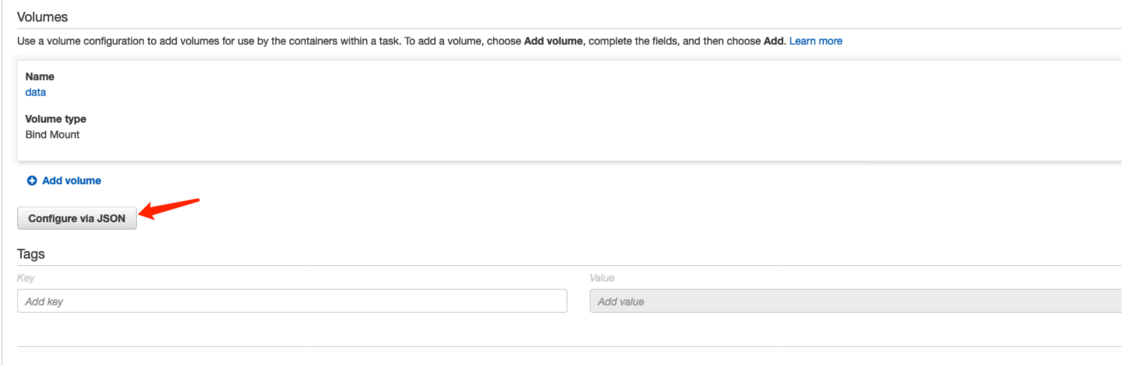
Volumes

Use a volume configuration to add volumes for use by the containers within a task. To add a volume, choose **Add volume**, complete the fields, and then choose **Add**. Learn more

Name
data

Volume type
Bind Mount

- 使用json格式，告知fluentbit使用extra.conf作为配置文件：



Volumes

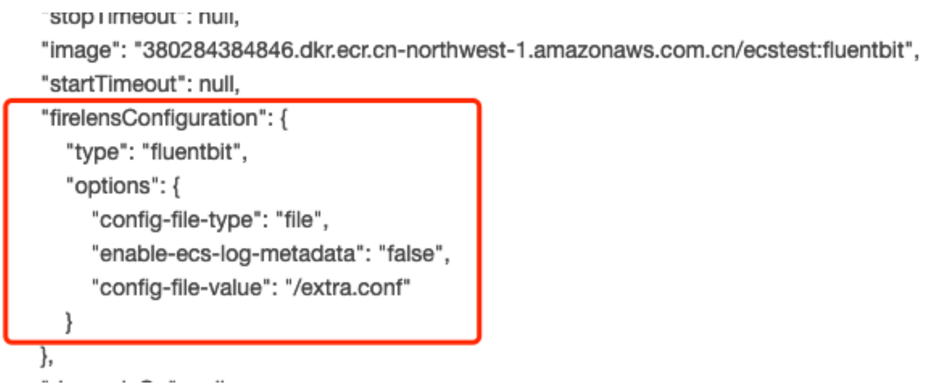
Use a volume configuration to add volumes for use by the containers within a task. To add a volume, choose **Add volume**, complete the fields, and then choose **Add**. Learn more

Name
data

Volume type
Bind Mount

Tags

Key Value

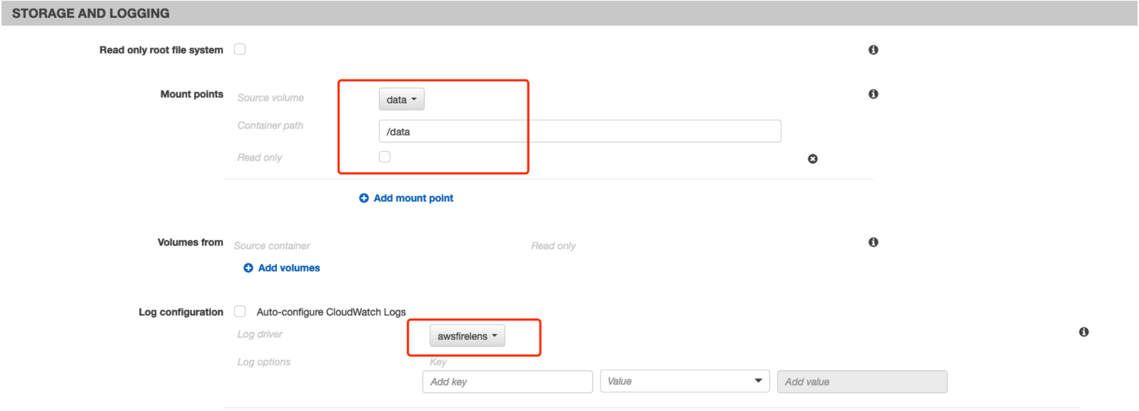


```
"stopTimeout": null,
"image": "380284384846.dkr.ecr.cn-northwest-1.amazonaws.com.cn/ecstest:fluentbit",
"startTimeout": null,
"firelensConfiguration": {
   "type": "fluentbit",
   "options": {
      "config-file-type": "file",
      "enable-ecs-log-metadata": "false",
      "config-file-value": "/extra.conf"
   }
},
```

```
"firelensConfiguration": {
                "type": "fluentbit",
                "options": {
                    "config-file-type": "file",  ##可以为s3或file，对
fargate只能用file
                    "enable-ecs-log-metadata": "false",
                    "config-file-value": "/extra.conf"
                }
            },
```

- 分别配置app container和fluentbit container的日志输出和文件挂载

  app container使用firelens的配置进行输出



STORAGE AND LOGGING

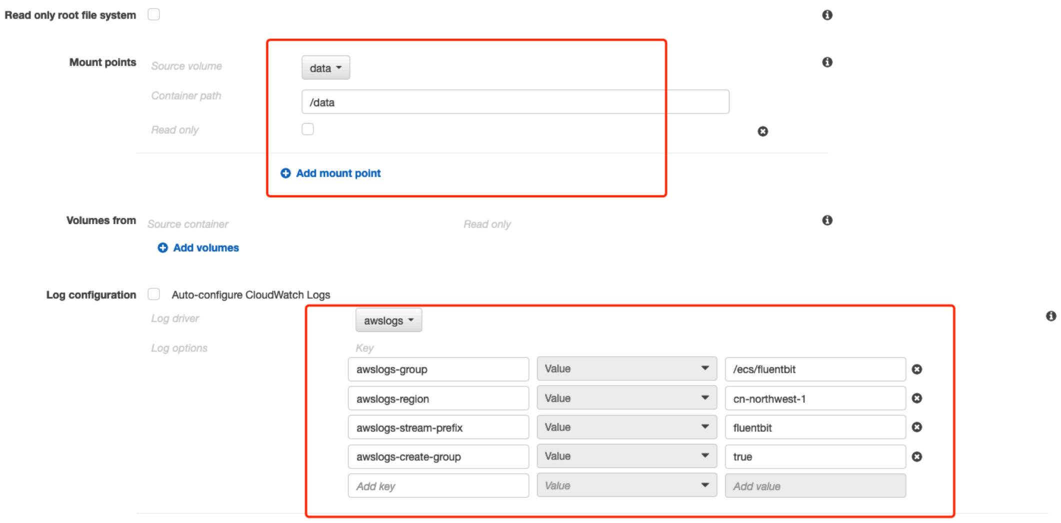
Read only root file system □

Mount points
Source volume: data
Container path: /data
Read only: □

Volumes from
Source container
Read only

Log configuration □ Auto-configure CloudWatch Logs
Log driver: awsfirelens
Log options
Key: Add key | Value | Add value

fluentbit container使用cloudwatch log进行日志输出，即fluentbit的运行情况从cloudwatch log进行观察：



Read only root file system □

Mount points
Source volume: data
Container path: /data
Read only: □

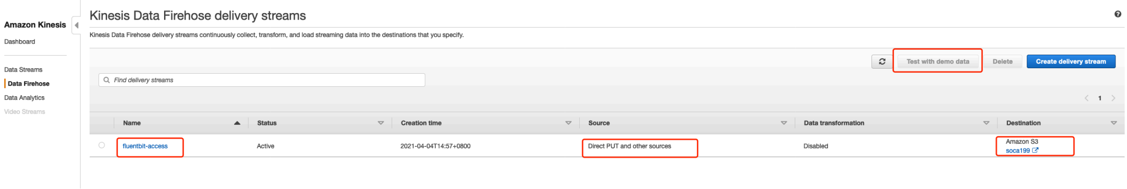
Volumes from
Source container
Read only

Log configuration □ Auto-configure CloudWatch Logs
Log driver: awslogs
Log options:

Key		
awslogs-group	Value	/ecs/fluentbit
awslogs-region	Value	cn-northwest-1
awslogs-stream-prefix	Value	fluentbit
awslogs-create-group	Value	true
Add key	Value	Add value

4. 创建firehose datastream

- 创建firehose datastream



Kinesis Data Firehose delivery streams

Kinesis Data Firehose delivery streams continuously collect, transform, and load streaming data into the destinations that you specify.

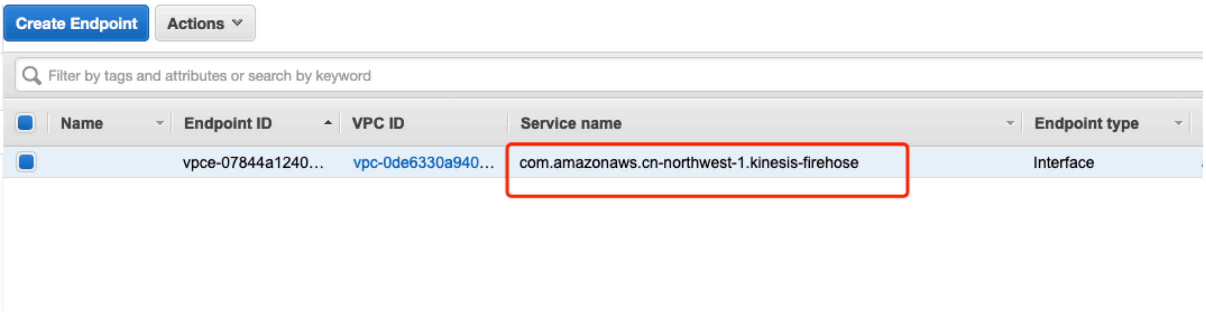
	Name	Status	Creation time	Source	Data transformation	Destination
○	fluentbit-access	Active	2021-04-04T14:57+0800	Direct PUT and other sources	Disabled	Amazon S3 soca199

注意delivery stream应该使用fluentbit output中的delivery_stream相同的名称。

为防止权限有关的问题，使用自带的测试数据功能，先保证firehose可以将数据输出到s3

- 创建firehose endpoint

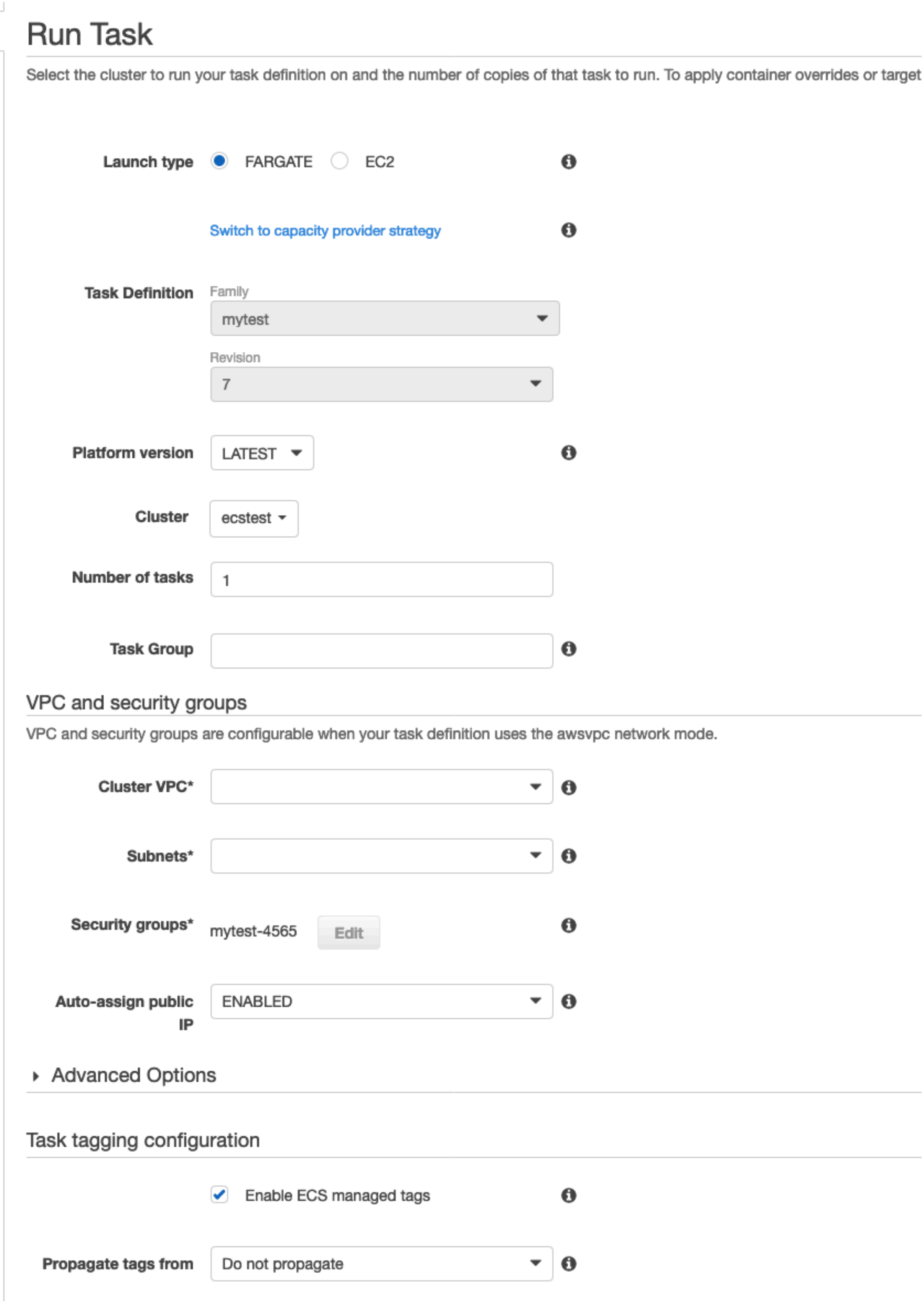
对firehose创建vpc endpoint，保证vpc内启动的fluentbit container可以访问到firehose服务。否则fluentbit container访问firehose时可能报访问firehose url 443端口timeout的错误



	Name	Endpoint ID	VPC ID	Service name	Endpoint type
■		vpce-07844a1240…	vpc-0de6330a940…	com.amazonaws.cn-northwest-1.kinesis-firehose	Interface

5.启动task

选择指定的vpc，subnet和sg。



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

Launch type ● FARGATE ○ EC2

Switch to capacity provider strategy

Task Definition Family: mytest

Revision: 7

Platform version: LATEST

Cluster: ecstest

Number of tasks: 1

Task Group:

VPC and security groups

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

Cluster VPC*:

Subnets*:

Security groups*: mytest-4565 Edit

Auto-assign public IP: ENABLED

Advanced Options

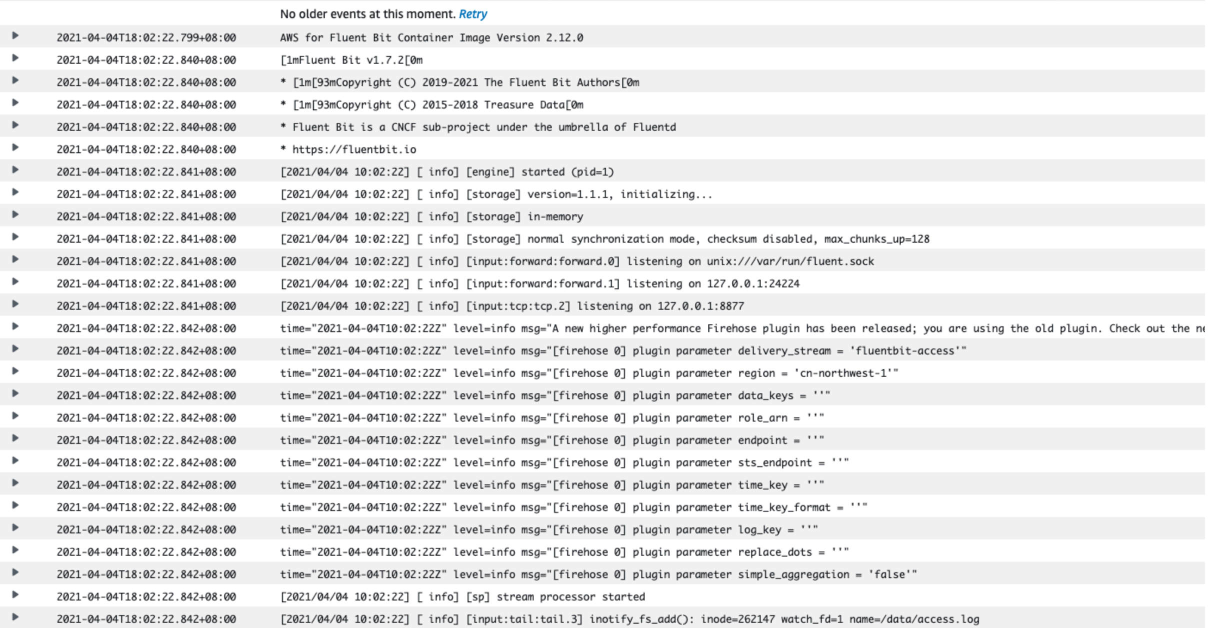
Task tagging configuration

☑ Enable ECS managed tags

Propagate tags from: Do not propagate

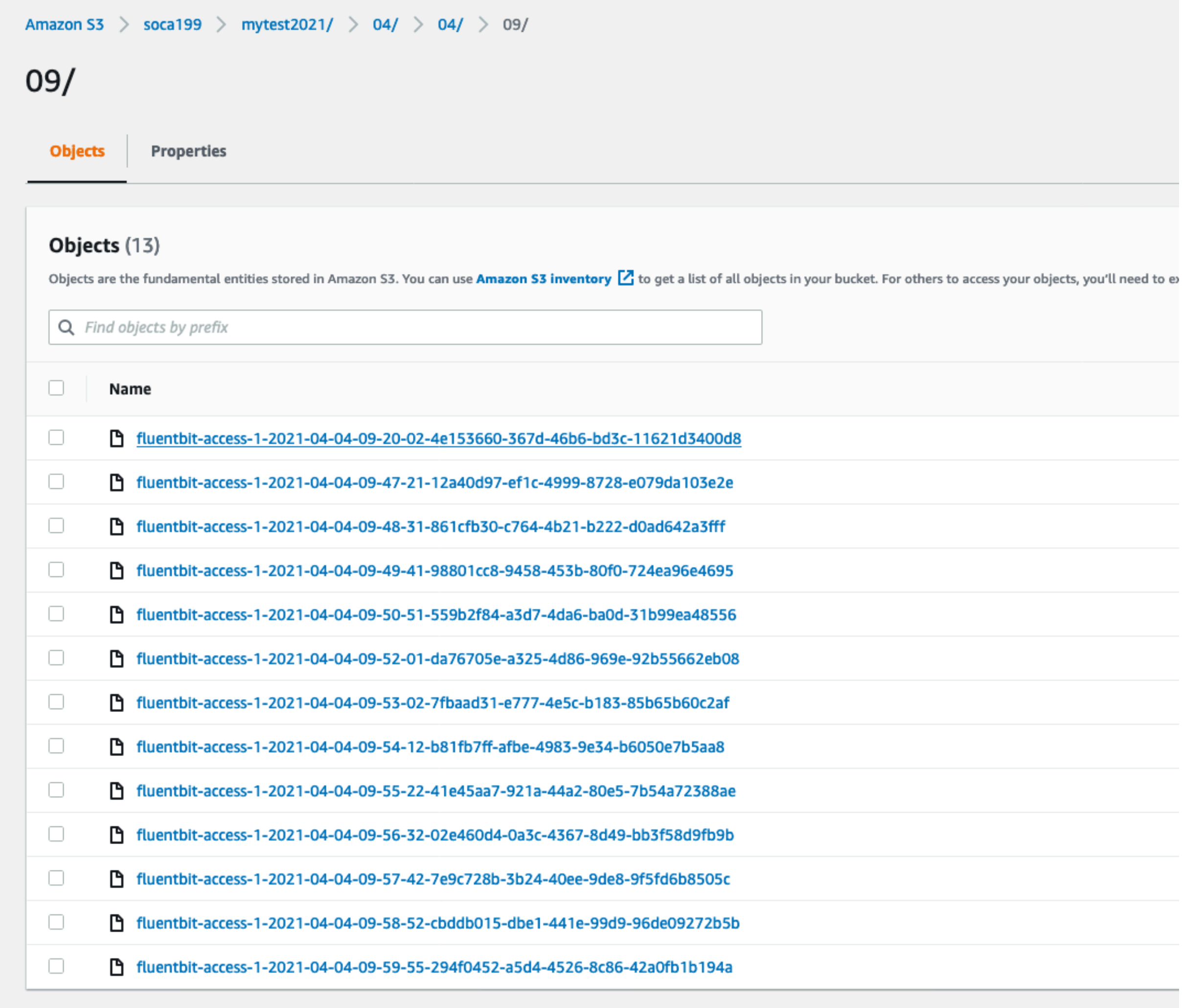
6. 检查

- fluentbit container在cloudwatch log中输出的日志：



```
                                       No older events at this moment. Retry
▶  2021-04-04T18:02:22.799+08:00   AWS for Fluent Bit Container Image Version 2.12.0
▶  2021-04-04T18:02:22.840+08:00   [1mFluent Bit v1.7.2[0m
▶  2021-04-04T18:02:22.840+08:00   * [1m[93mCopyright (C) 2019-2021 The Fluent Bit Authors[0m
▶  2021-04-04T18:02:22.840+08:00   * [1m[93mCopyright (C) 2015-2018 Treasure Data[0m
▶  2021-04-04T18:02:22.840+08:00   * Fluent Bit is a CNCF sub-project under the umbrella of Fluentd
▶  2021-04-04T18:02:22.840+08:00   * https://fluentbit.io
▶  2021-04-04T18:02:22.841+08:00   [2021/04/04 10:02:22] [ info] [engine] started (pid=1)
▶  2021-04-04T18:02:22.841+08:00   [2021/04/04 10:02:22] [ info] [storage] version=1.1.1, initializing...
▶  2021-04-04T18:02:22.841+08:00   [2021/04/04 10:02:22] [ info] [storage] in-memory
▶  2021-04-04T18:02:22.841+08:00   [2021/04/04 10:02:22] [ info] [storage] normal synchronization mode, checksum disabled, max_chunks_up=128
▶  2021-04-04T18:02:22.841+08:00   [2021/04/04 10:02:22] [ info] [input:forward:forward.0] listening on unix:///var/run/fluent.sock
▶  2021-04-04T18:02:22.841+08:00   [2021/04/04 10:02:22] [ info] [input:forward:forward.1] listening on 127.0.0.1:24224
▶  2021-04-04T18:02:22.841+08:00   [2021/04/04 10:02:22] [ info] [input:tcp:tcp.2] listening on 127.0.0.1:8877
▶  2021-04-04T18:02:22.842+08:00   time="2021-04-04T10:02:22Z" level=info msg="A new higher performance Firehose plugin has been released; you are using the old plugin. Check out the ne
▶  2021-04-04T18:02:22.842+08:00   time="2021-04-04T10:02:22Z" level=info msg="[firehose 0] plugin parameter delivery_stream = 'fluentbit-access'"
▶  2021-04-04T18:02:22.842+08:00   time="2021-04-04T10:02:22Z" level=info msg="[firehose 0] plugin parameter region = 'cn-northwest-1'"
▶  2021-04-04T18:02:22.842+08:00   time="2021-04-04T10:02:22Z" level=info msg="[firehose 0] plugin parameter data_keys = ''"
▶  2021-04-04T18:02:22.842+08:00   time="2021-04-04T10:02:22Z" level=info msg="[firehose 0] plugin parameter role_arn = ''"
▶  2021-04-04T18:02:22.842+08:00   time="2021-04-04T10:02:22Z" level=info msg="[firehose 0] plugin parameter endpoint = ''"
▶  2021-04-04T18:02:22.842+08:00   time="2021-04-04T10:02:22Z" level=info msg="[firehose 0] plugin parameter sts_endpoint = ''"
▶  2021-04-04T18:02:22.842+08:00   time="2021-04-04T10:02:22Z" level=info msg="[firehose 0] plugin parameter time_key = ''"
▶  2021-04-04T18:02:22.842+08:00   time="2021-04-04T10:02:22Z" level=info msg="[firehose 0] plugin parameter time_key_format = ''"
▶  2021-04-04T18:02:22.842+08:00   time="2021-04-04T10:02:22Z" level=info msg="[firehose 0] plugin parameter log_key = ''"
▶  2021-04-04T18:02:22.842+08:00   time="2021-04-04T10:02:22Z" level=info msg="[firehose 0] plugin parameter replace_dots = ''"
▶  2021-04-04T18:02:22.842+08:00   time="2021-04-04T10:02:22Z" level=info msg="[firehose 0] plugin parameter simple_aggregation = 'false'"
▶  2021-04-04T18:02:22.842+08:00   [2021/04/04 10:02:22] [ info] [sp] stream processor started
▶  2021-04-04T18:02:22.842+08:00   [2021/04/04 10:02:22] [ info] [input:tail:tail.3] inotify_fs_add(): inode=262147 watch_fd=1 name=/data/access.log
```

- app container输出的日志



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 ex

☐	Name
☐	fluentbit-access-1-2021-04-04-09-20-02-4e153660-367d-46b6-bd3c-11621d3400d8
☐	fluentbit-access-1-2021-04-04-09-47-21-12a40d97-ef1c-4999-8728-e079da103e2e
☐	fluentbit-access-1-2021-04-04-09-48-31-861cfb30-c764-4b21-b222-d0ad642a3fff
☐	fluentbit-access-1-2021-04-04-09-49-41-98801cc8-9458-453b-80f0-724ea96e4695
☐	fluentbit-access-1-2021-04-04-09-50-51-559b2f84-a3d7-4da6-ba0d-31b99ea48556
☐	fluentbit-access-1-2021-04-04-09-52-01-da76705e-a325-4d86-969e-92b55662eb08
☐	fluentbit-access-1-2021-04-04-09-53-02-7fbaad31-e777-4e5c-b183-85b65b60c2af
☐	fluentbit-access-1-2021-04-04-09-54-12-b81fb7ff-afbe-4983-9e34-b6050e7b5aa8
☐	fluentbit-access-1-2021-04-04-09-55-22-41e45aa7-921a-44a2-80e5-7b54a72388ae
☐	fluentbit-access-1-2021-04-04-09-56-32-02e460d4-0a3c-4367-8d49-bb3f58d9fb9b
☐	fluentbit-access-1-2021-04-04-09-57-42-7e9c728b-3b24-40ee-9de8-9f5fd6b8505c
☐	fluentbit-access-1-2021-04-04-09-58-52-cbddb015-dbe1-441e-99d9-96de09272b5b
☐	fluentbit-access-1-2021-04-04-09-59-55-294f0452-a5d4-4526-8c86-42a0fb1b194a