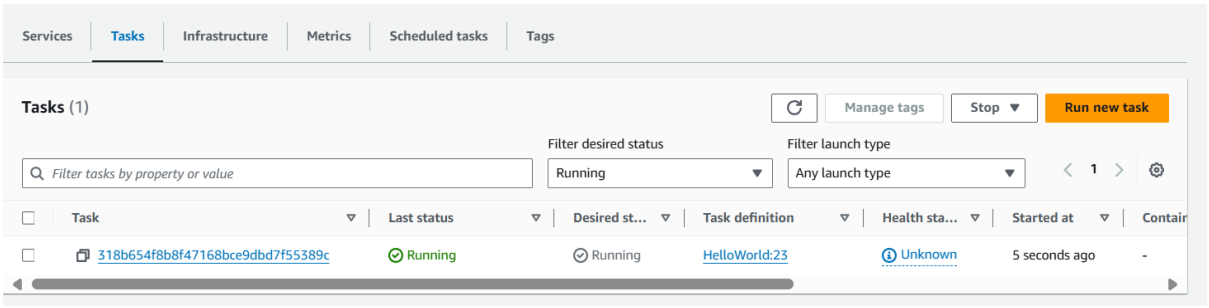
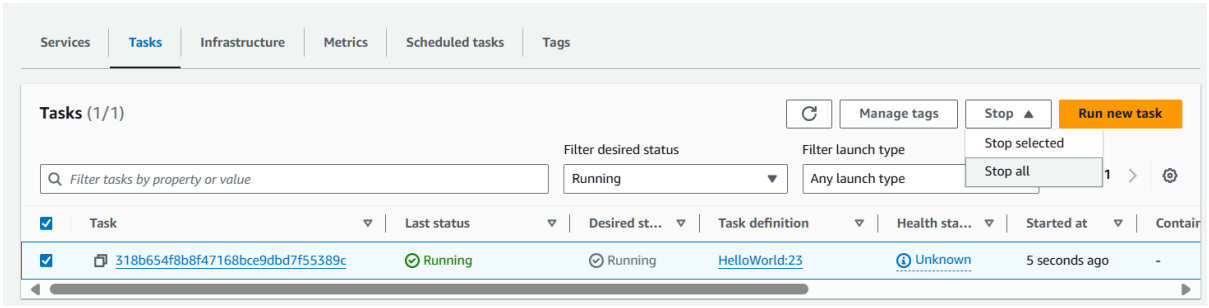
1. Any task that is running on Amazon ECS Spot capacity provider is subject to interruption. There are two possibilities for task non-completion.
   a. The running Amazon ECS Fargate task is interrupted. Whenever the task is interrupted there will 120 seconds post which task will be killed.
   b. There is possibility that Amazon ECS Fargate task is not available for invocation.

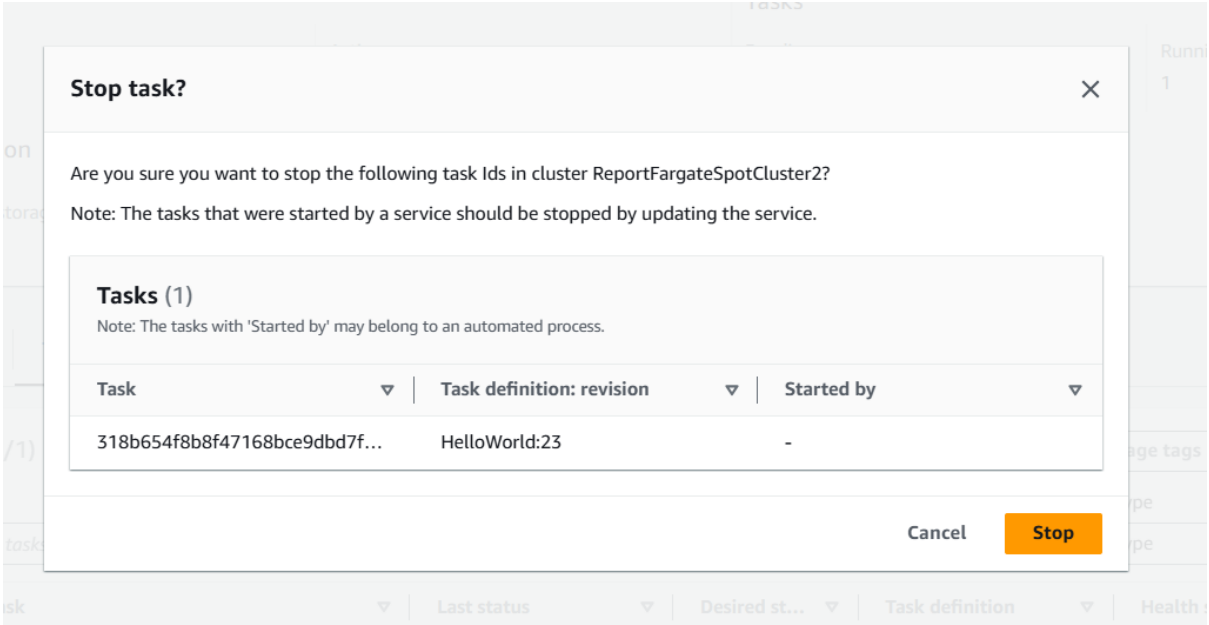
   In the write up we shall detail how to simulate both the situation in testing environment and how to remediate.

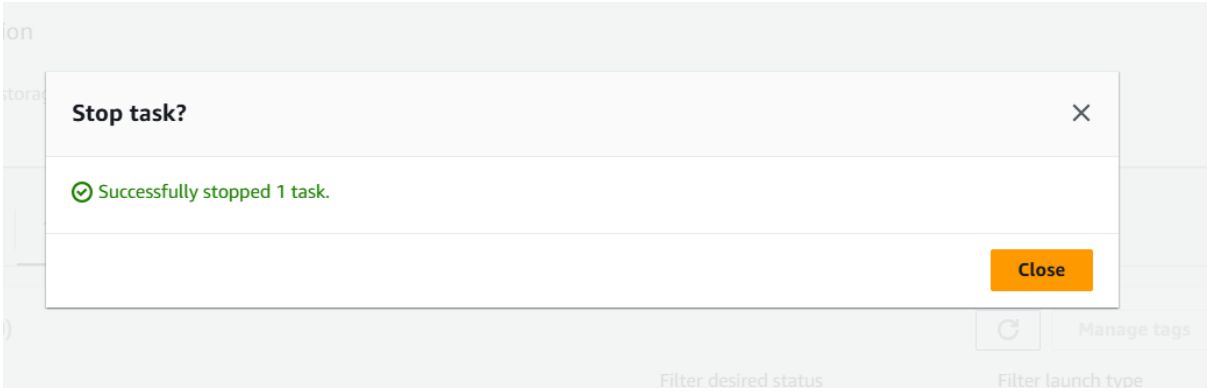
2. How to simulate Amazon ECS Fargate task interruption while the task is running.
   When any Amazon ECS Fargate task is interrupted, the SIGTERM state change event is sent from task kernel to process. The SIGTERM event has to be captured in the code and do needful. Here we are simulating the process by killing the task manually.
   a. Run the task manually from Amazon ECS cluster



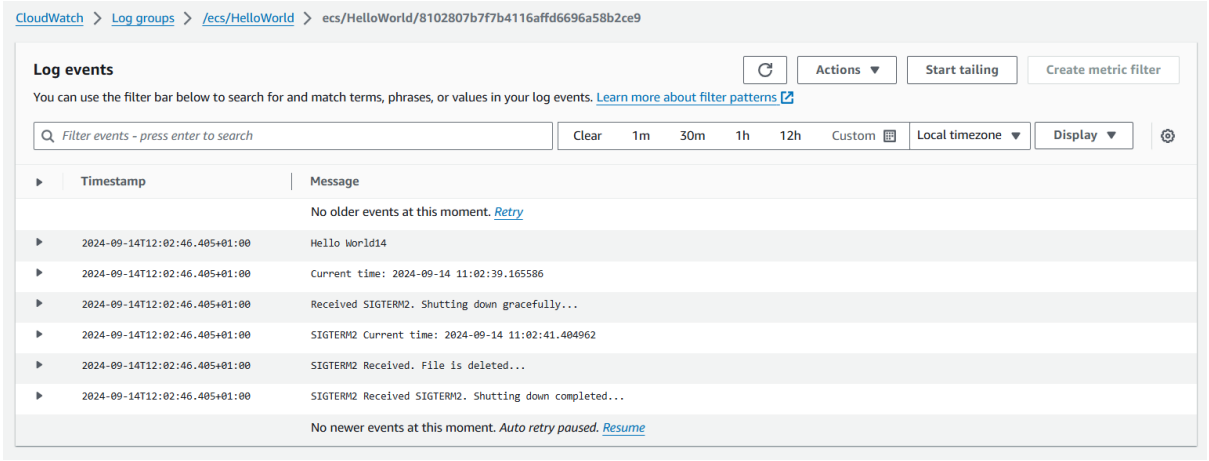

   b. While the task is on running condition, select the task and manually stop the task






c. Validate in Amazon CloudWatch Loggroup that task has been interrupted.




3. Code for Gracious Exit
   We have introduced the function which will receive the SIGTERM when Amazon ECS Fargate task is interrupted.

```
10
11  def sigterm_handler(signum, frame):
12      print("Received SIGTERM2. Shutting down gracefully...")
13      current_time = datetime.datetime.fromtimestamp(time.time())
14      print(f"SIGTERM2 Current time: {current_time}")
15      time.sleep(5)
16      current_time = datetime.datetime.fromtimestamp(time.time())
17      s3 = boto3.client('s3')
18      s3.delete_object(Bucket=bucket_name, Key=file_name)
19      print("SIGTERM2 Received. File is deleted...")
20      print("SIGTERM2 Received SIGTERM2. Shutting down completed...")
21      # Perform cleanup operations here
22      sys.exit(0)
23
```

4. The Amazon ECS Fargate task definition for timeout
   When a Amazon ECS Fargate Task is interrupted, the process gets 120 seconds to graciously exit the process. In the function, all necessary work can be undone so that freshly task can be started.
   To simulate this we have introduced some delays in the process.

a. Manually introduce delay in code
We have introduced a delay in the code so that process runs for longer time when task can be interrupted by manually stopping the task.

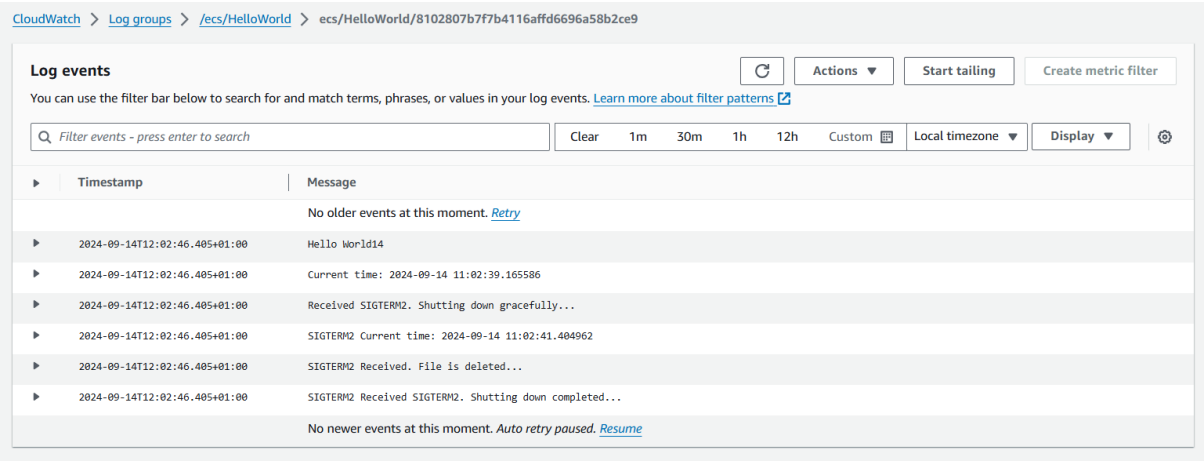
```
110
111   print("Hello World14")
112   current_time = datetime.datetime.fromtimestamp(time.time())
113   print(f"Current time: {current_time}")
114   time.sleep(30)
115   current_time = datetime.datetime.fromtimestamp(time.time())
116   print(f"Current time+: {current_time}")
117   print(sum)
```

b. Increate Amazon ECS Fargate task timeout
Any Amazon ECS Fargate Task by default gets 30 second to complete before stopping. This is a configurable and we have configured it in task definition. This gives the time to SIGTERM handler to complete the gracious exit. When the timeout is configured with less second the SIGTERM handler couldn't finish the execution.

```
21              "startTimeout": 5,
22              "stopTimeout": 30,
23              "logConfiguration": {
```

c. Add delay in in SIGTERM code
Finally we have added the delay in SIGTERM handler to validate the SIGTERM handler gets enough time to complete the gracious exit.

```
11  def sigterm_handler(signum, frame):
12      print("Received SIGTERM2. Shutting down gracefully...")
13      current_time = datetime.datetime.fromtimestamp(time.time())
14      print(f"SIGTERM2 Current time: {current_time}")
15      time.sleep(5)
16      current_time = datetime.datetime.fromtimestamp(time.time())
```

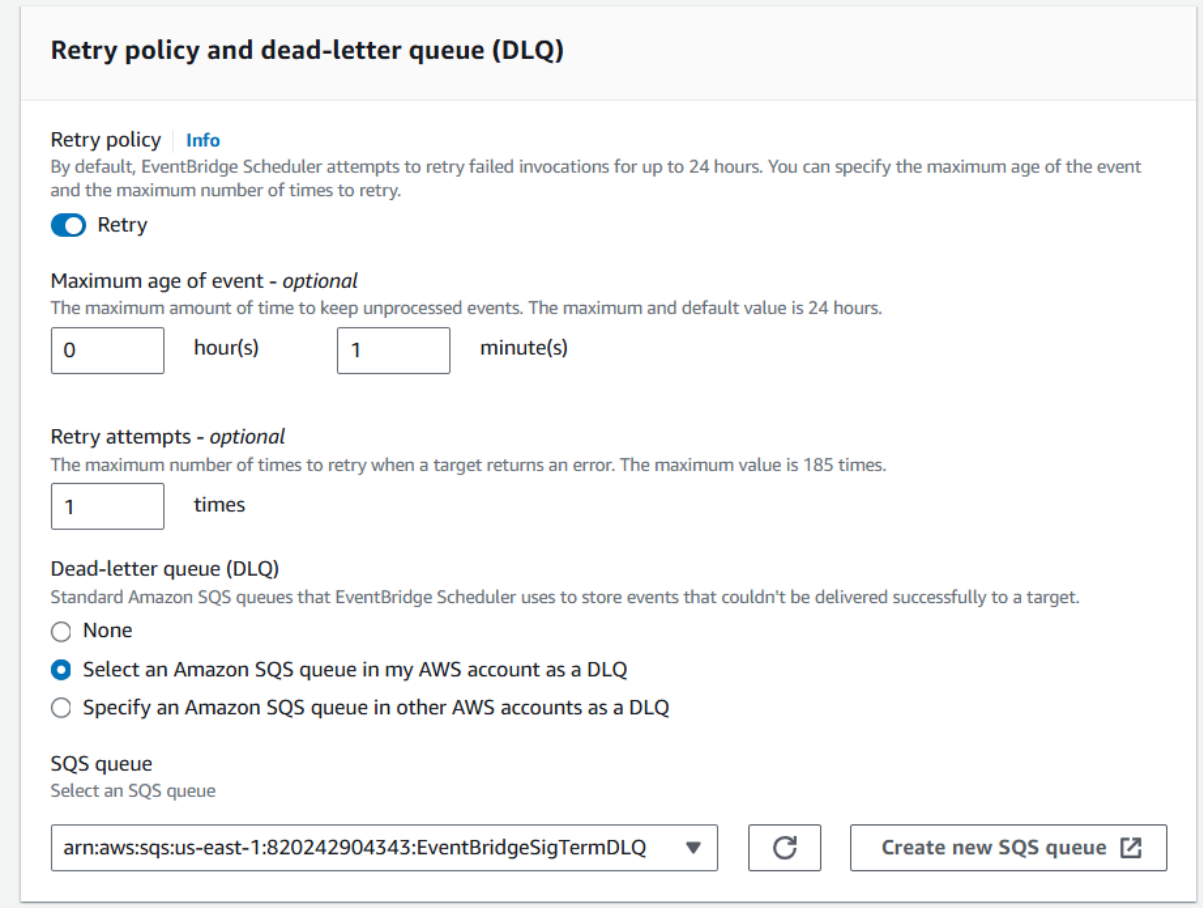


5. How to simulate when the Amazon ECS Fargate task(spot) not able to invoke for capacity non-availability.

   In our solution, the docker container in Amazon ECS Fargate Spot is invoked through Amazon EventBridge scheduler. In earlier section, we have seen how explained how task interruption can be alerted and remediated.

   In the following section, we shall explain how to get notification when Amazon EventBridge Scheduler not able to invoke the Amazon ECS Fargate Task on spot for non availability.

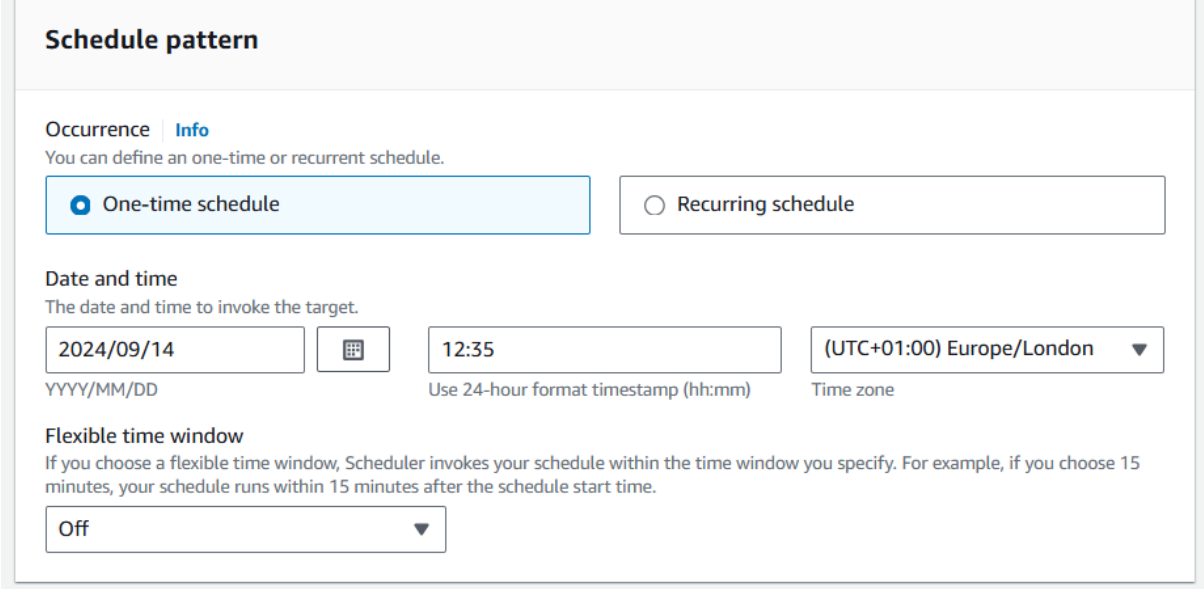
   a. Amazon Schedule redrive if task not available. If the task fails after retry it will send to Amazon SQS DLQ which will send mail through Amazon SNS notification.



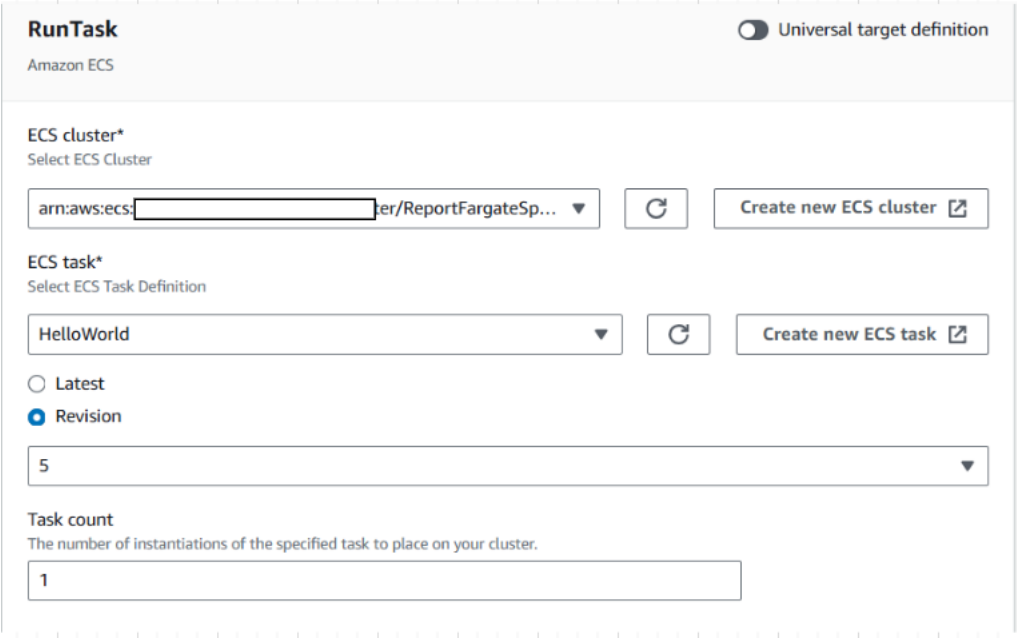

   We shall define redrive, SNS notification in case the task is not able to place.

   Further we shall simulate the task non availability.

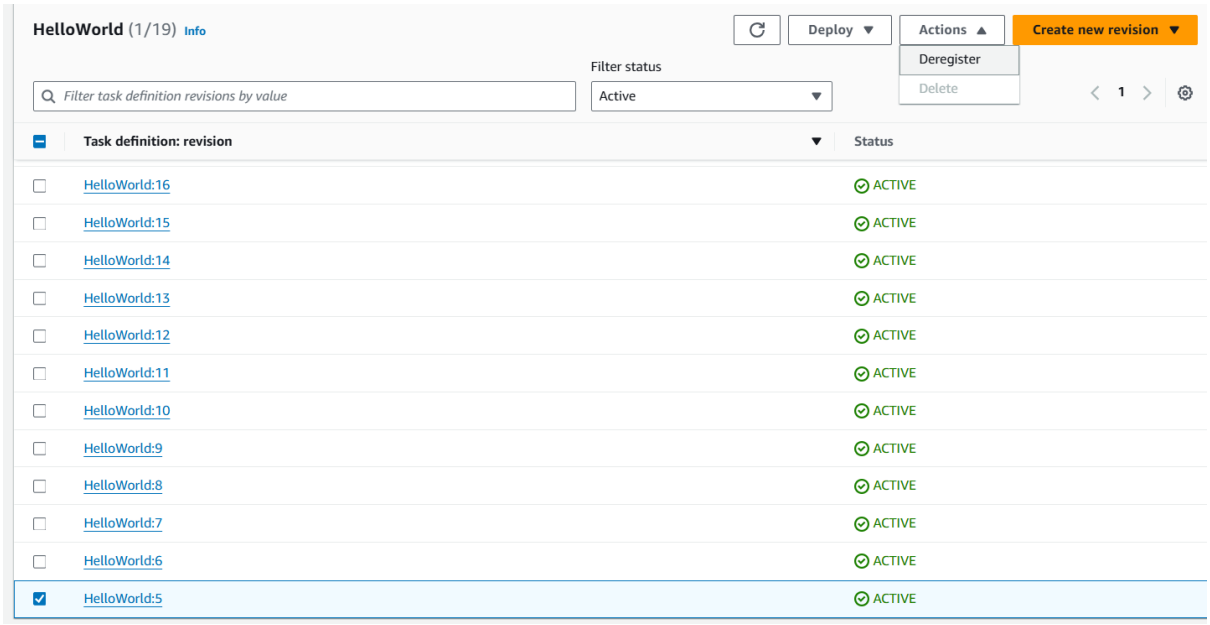
6. Simulation of Amazon ECS task on spot non available.
   We shall create a schedule 5 minutes in advance with Amazon ECS Fargate task definition version. Once the schedule is active and ready to be launched, we shall deregister the task definition version. As a result the task will fail to get invoked and message will be passed to Amazon SQS DLQ which can trigger notification.



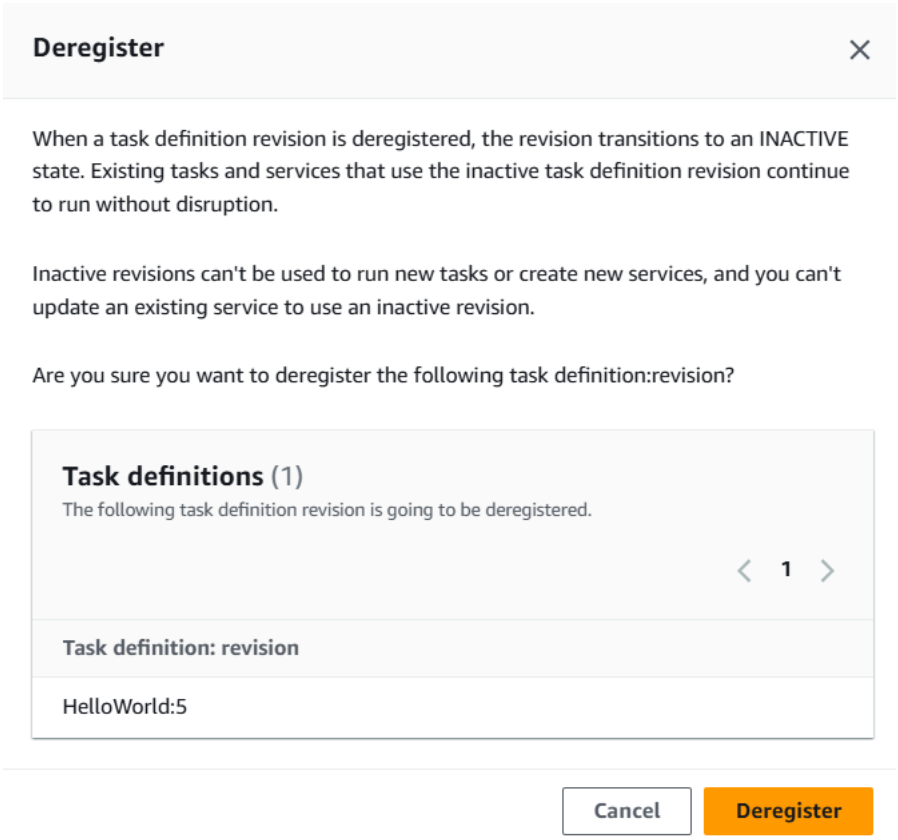

The Amazon EventBridge schedule is set at 12:35

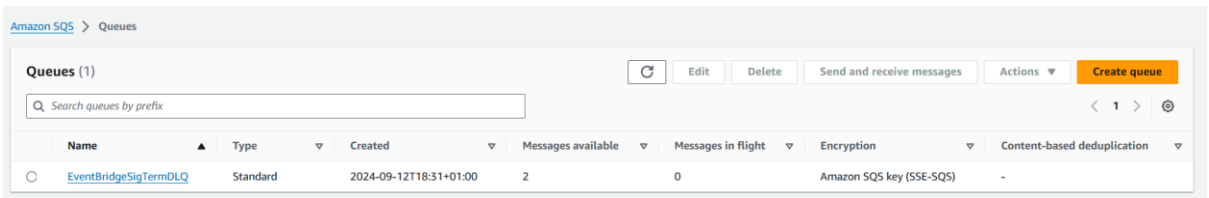



The version is set at 5.




The selected version is deregistered.



SQS Message available after Amazon EventBridge failed to invoke the Amazon ECS Fargate task
There is 2 messages available as redrive is implemented after failure.

7. Clean Up