I used the Fargate option to run the container.

1. Artillery for testing

```yaml
config:
  target: 'http://my-app-alb-903367878.eu-north-1.elb.amazonaws.com'
  phases:

    - duration: 60
      arrivalRate: 200


    - duration: 400
      arrivalRate: 500
      rampTo: 1000
```

2. Scaling

Scaling activities

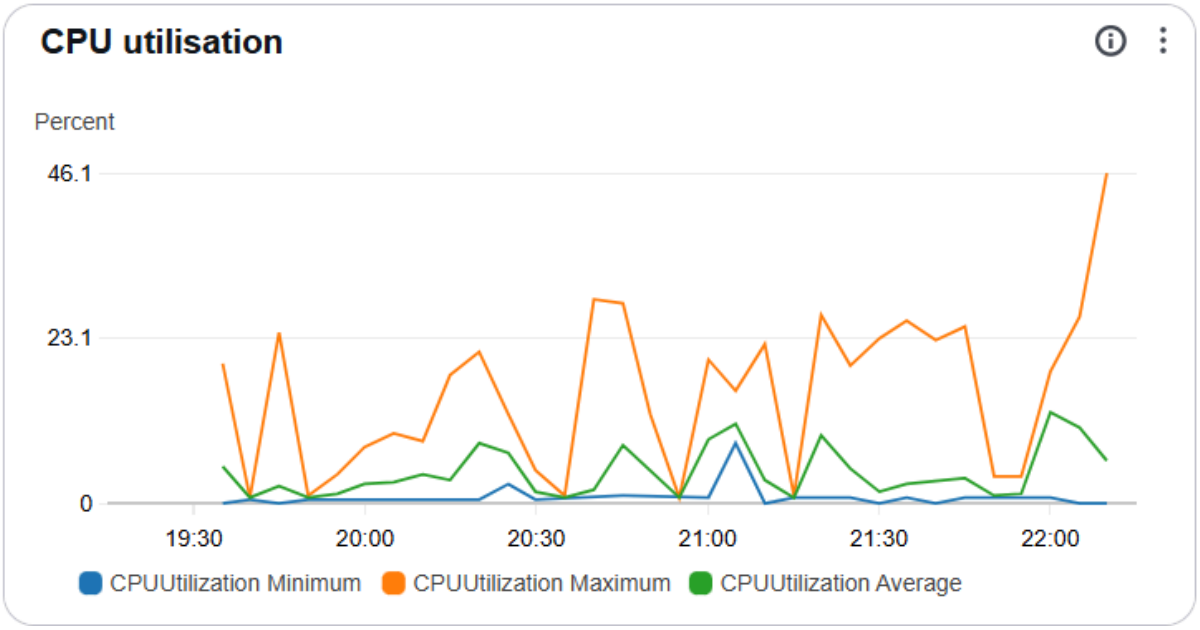
Find scaling activity

1

Resource ID	Scalable dimension	Status	Status message	Start time
service/MyTestCLICluster...	ecs:service:DesiredCount	Successful	Successfully set desired count to 4. Change successfully fulfilled by ecs.	15 May 2025, 00:07
service/MyTestCLICluster...	ecs:service:DesiredCount	Overridden	Successfully set desired count to 3. Found it was later changed to 4.	15 May 2025, 00:06

Cloudwatch

CPU using



Scale down

Resource ID	Scalable dimension	Status	Status message	Start time
service/MyTestCLICluster...	ecs:service:DesiredCount	Successful	Successfully set desired count to 3. Change successfully fulfilled by ecs.	15 May 2025, 00:33
service/MyTestCLICluster...	ecs:service:DesiredCount	Successful	Successfully set desired count to 4. Change successfully fulfilled by ecs.	15 May 2025, 00:07

Comparative Analysis

Fargate vs. EC2 Auto Scaling

Let's talk about the two ways we could have run our app on AWS: using AWS Fargate (which we did) or using EC2 Auto Scaling Groups (the more traditional way). We'll compare them on how easy they are to set up, how much they might cost, and how well they can grow if our app gets more users.

4.1. AWS Fargate

Think of AWS Fargate as a helper that runs your app's containers (which hold your app code) without you needing to manage the actual servers (EC2 instances) yourself. AWS takes care of the servers for you.

- **Getting Started (Ease of Setup):**
    - **Good Points:** The best part is you don't have to worry about setting up servers, updating their software (OS patching), or installing Docker on them. If your team already knows Docker, it can be quicker to get your app running. Fargate also works well with other AWS tools like ECS (for managing your containers), ECR (for storing your app's images), and Application Load Balancers (for handling website traffic).

    - **Things to Think About:** You'll need to learn some new terms and ideas for ECS and Fargate, like "Task Definitions" and "Services." Setting up the network parts (like VPCs, subnets, security groups, and making sure Fargate can get your app's image from ECR, maybe using a NAT Gateway) can also be a bit tricky at first. Plus, you have a little less direct control over how things run compared to using regular EC2 servers.
- **What It Means for Cost:**
    - **Good Points:** You only pay for the computer power (CPU) and memory your app actually uses, and AWS charges you by the second (after the first minute). This means you don't pay for empty server space if your app isn't busy or if it scales down to zero running copies. Auto-scaling also helps you save money by only using the resources you really need. If your app can handle small interruptions, Fargate Spot can save you a lot of money (up to 70%).
    - **Things to Think About:** If your app is extremely busy all the time, Fargate might sometimes cost a bit more per hour than if you used specially set-up EC2 servers (like Reserved Instances or EC2 Spot). Also, if you use a NAT Gateway for your app to send traffic out to the internet, that can add to your bill.
- **Scalability:**
    - **Good Points:** Fargate can start new copies (tasks) of your app pretty quickly, usually faster than starting up a whole new EC2 server. ECS auto-scaling is very flexible; it can add or remove copies of your app based on how busy it is – looking at things like CPU use, memory, how many requests it's getting, or even custom signals you set up. This is great for modern apps made of small, independent parts (microservices) or for apps where the amount of traffic changes a lot. *In our own tests, we saw our app successfully grow from one to four copies when the average CPU use went over our 5% target during the load test.*
    - **Things to Think About:** AWS has some limits on how many Fargate tasks you can run, but these limits are usually very high. Also, while starting new tasks is fast, it doesn't happen in a flash, which might matter for very rare apps that need to scale in less than a second.

4.2. EC2 Auto Scaling Groups

This is the older way of doing things. You put your app's containers on normal EC2 servers, and "Auto Scaling Groups" help manage how many of these servers are running.

- **Getting Started (Ease of Setup):**
    - **Good Points:** You have complete control over the server – its operating system, any software on it, and the exact type of server you use. It's an older and well-known way of doing things, so it might feel more familiar if your team has a lot of experience with EC2 and managing servers.

    - **Things to Think About:** It means more extra work for you. You have to take care of server updates, security, installing and setting up Docker, and managing server images or writing startup scripts. Setting up how new servers are created (using Launch Templates), the Auto Scaling Groups themselves, and making them work correctly with load balancers and your containers is also more complicated.
- **What It Means for Cost:**
    - **Good Points:** You might pay less per hour if your app has very steady, high traffic all the time, especially if you use EC2 Reserved Instances, Savings Plans, or EC2 Spot Instances (which offer big discounts). There are also many types of EC2 servers to choose from, so you can try to pick one that perfectly fits your needs and budget.
    - **Things to Think About:** There's a risk you might pay for more server space than you actually need if your auto-scaling isn't set up just right. Or, you might not have enough servers if traffic suddenly spikes. You also pay for the whole EC2 server when it's running, even if your app is only using a small part of its power.
- **Scalability:**
    - **Good Points:** This method can also scale a lot – you can have thousands of servers if your app needs them. Auto Scaling Groups can also react to many different signals from CloudWatch (an AWS monitoring service) to decide when to add or remove servers.
    - **Things to Think About:** Starting new EC2 servers (which includes booting up the operating system, running any setup scripts, and then starting your container) usually takes longer than starting Fargate tasks. This can make it slower to react to sudden jumps in website traffic.