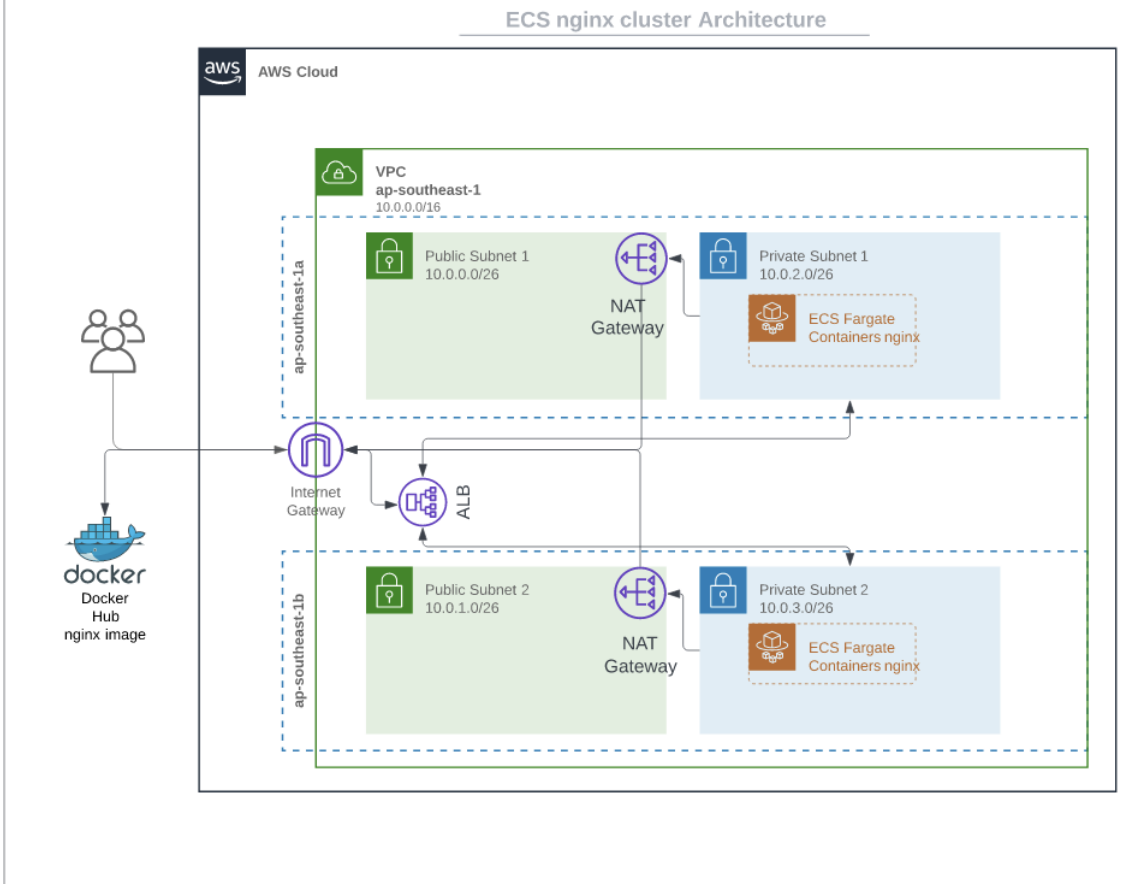
13519165
Kadek Surya Mahardika

Source Code

https://github.com/kadeksuryam/AWS-Nginx-ECS-Cluster

Architecture



How I Implement The Assignment

- First I read the documentation about ECS Cluster at https://docs.aws.amazon.com/AmazonECS/latest/developerguide/clusters.html
- Then, I tried to find terraform resources related to ECS cluster at https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/ecs_account_setting_default
- Finally, I implement all the required AWS components. For me, the simplest to implement was AWS networking, then ECS related stuff.

Difficulties encountered

The difficulties mainly come when I integrate ECS components with the networking components. Turns out I missed a couple configuration in networking components, initially ECS cannot pull nginx images from Docker Hub, after I fixed NAT and Subnet configurations, the service successfully pulled the image.

How to test the cluster is working

By doing HTTP Request to load balancer hostname:
Example Request:

Request
curl http://tf-lb-20221208053814856300000005-1397205297.ap-southeast-1.elb.amazonaws.com/
Response
<!DOCTYPE html> <html> <head> <title>Welcome to nginx!</title> <style> html { color-scheme: light dark; } body { width: 35em; margin: 0 auto; font-family: Tahoma, Verdana, Arial, sans-serif; } </style> </head> <body> Welcome to nginx!   If you see this page, the nginx web server is successfully installed and working. Further configuration is required.   For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.   Thank you for using nginx. </body> </html>

What I have learned

I learned a lot, especially about AWS Networking stuff such as VPC, Subnet, Load Balancer. I also learned about ECS and how to integrate it with AWS Networking components so ECS Task is able to reach the internet.

BONUS

How do I Implement the autoscaling

- First I create a role with policy for auto scaling service so the service will have authorization to adjust the desired count of ecs tasks
- Then, I setup the policy, in this policy I defined the metrics for ALBRequestCountPerTarget that's 10 request i.e. load balancer requests per target count

How do I test it

I use a tools named Apache Benchmark

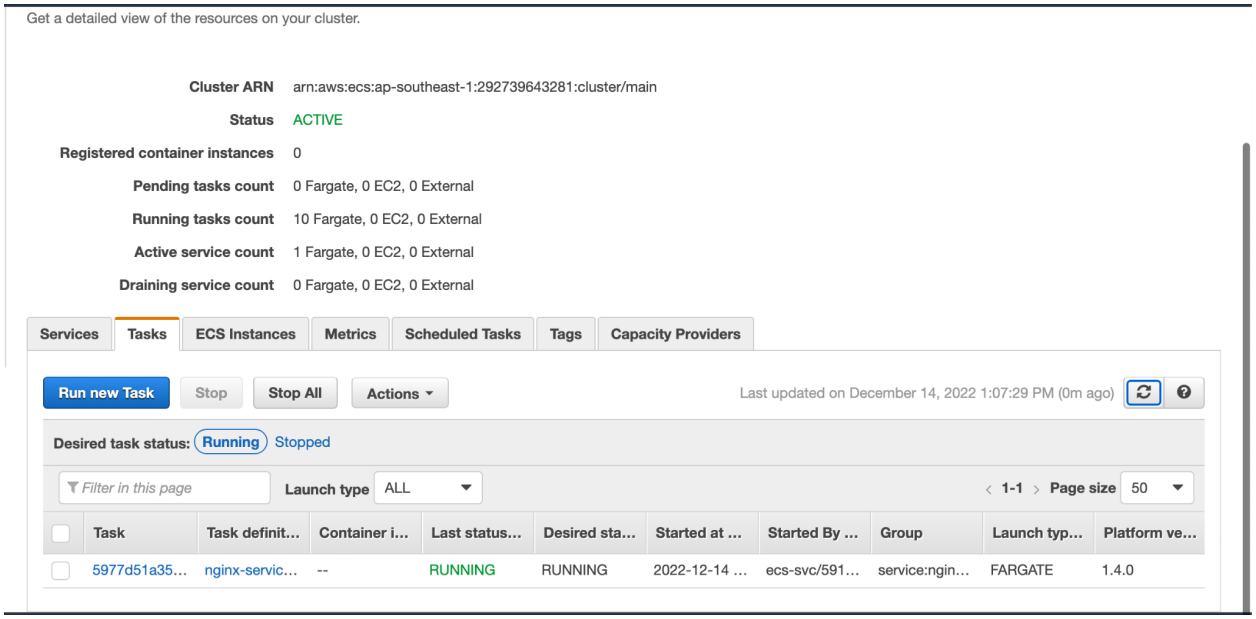
```
→  pat git:(master) ✗ ab -n 100000 -c 50 http://ecs-alb-1785620585.ap-southeast-1.elb.amazonaws.com/
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking ecs-alb-1785620585.ap-southeast-1.elb.amazonaws.com (be patient)
apr_socket_recv: Operation timed out (60)
Total of 2950 requests completed
```

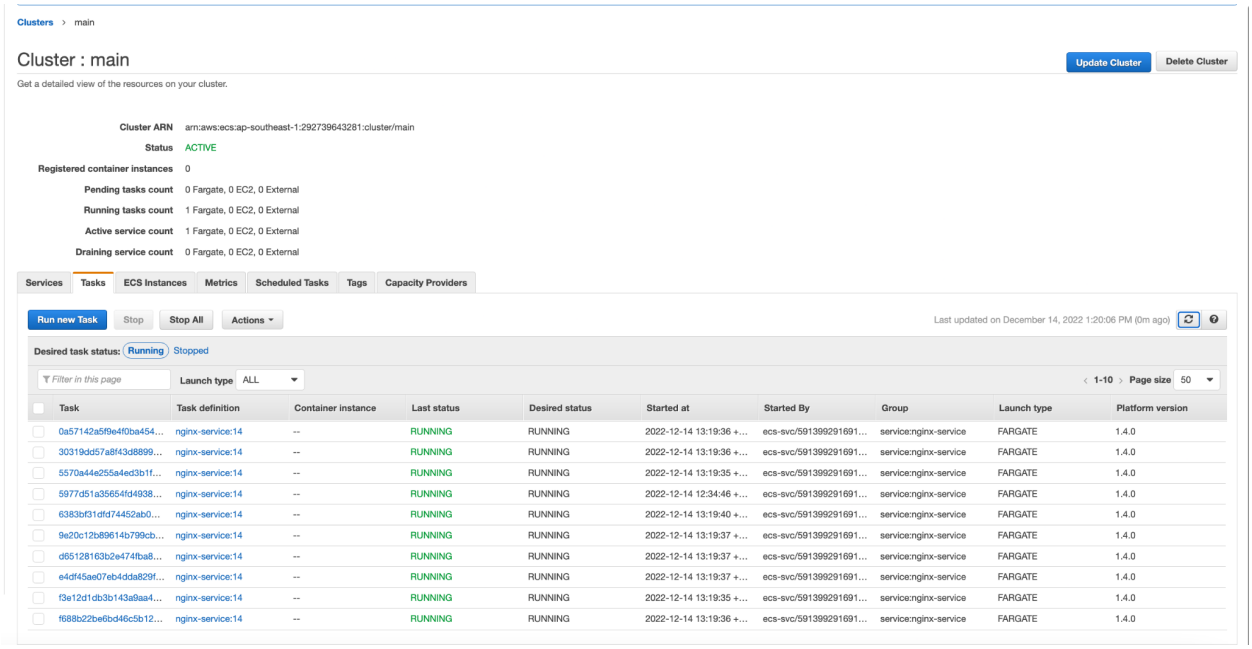
I sent 100k requests with 50 concurrent requests.

After that, you can see the number of ECS tasks has been adjusted to 10. Initially the number of tasks is 1.

Initial state



After applying benchmark (Wait several minutes for the alarms to be triggered)


Clusters › main
Cluster : main
Update Cluster
Delete Cluster
Get a detailed view of the resources on your cluster.
Cluster ARN arn:aws:ecs:ap-southeast-1:292739643281:cluster/main
Status ACTIVE
Registered container instances 0
Pending tasks count 0 Fargate, 0 EC2, 0 External
Running tasks count 1 Fargate, 0 EC2, 0 External
Active service count 1 Fargate, 0 EC2, 0 External
Draining service count 0 Fargate, 0 EC2, 0 External
Services Tasks ECS Instances Metrics Scheduled Tasks Tags Capacity Providers
Run new Task Stop Stop All Actions
Last updated on December 14, 2022 1:20:06 PM (0m ago)
Desired task status: Running Stopped
Filter in this page
Launch type ALL
1-10 Page size 50
Task Task definition Container instance Last status Desired status Started at Started By Group Launch type Platform version
0a57142a5f9e4f0ba454... nginx-service:14 -- RUNNING RUNNING 2022-12-14 13:19:36 +... ecs-svc/591399291691... service:nginx-service FARGATE 1.4.0
30319dd57a8f43d8899... nginx-service:14 -- RUNNING RUNNING 2022-12-14 13:19:36 +... ecs-svc/591399291691... service:nginx-service FARGATE 1.4.0
5570a44e255a4ed3b1f... nginx-service:14 -- RUNNING RUNNING 2022-12-14 13:19:35 +... ecs-svc/591399291691... service:nginx-service FARGATE 1.4.0
5977d51a35654fd4938... nginx-service:14 -- RUNNING RUNNING 2022-12-14 12:34:46 +... ecs-svc/591399291691... service:nginx-service FARGATE 1.4.0
6383bf31dfd74452ab0... nginx-service:14 -- RUNNING RUNNING 2022-12-14 13:19:40 +... ecs-svc/591399291691... service:nginx-service FARGATE 1.4.0
9e20c12b89614b799cb... nginx-service:14 -- RUNNING RUNNING 2022-12-14 13:19:37 +... ecs-svc/591399291691... service:nginx-service FARGATE 1.4.0
d65128163b2e474fba8... nginx-service:14 -- RUNNING RUNNING 2022-12-14 13:19:37 +... ecs-svc/591399291691... service:nginx-service FARGATE 1.4.0
e4df45ae07eb4dda829f... nginx-service:14 -- RUNNING RUNNING 2022-12-14 13:19:37 +... ecs-svc/591399291691... service:nginx-service FARGATE 1.4.0
f3e12d1db3b143a9aa4... nginx-service:14 -- RUNNING RUNNING 2022-12-14 13:19:35 +... ecs-svc/591399291691... service:nginx-service FARGATE 1.4.0
f688b22be6bd46c5b12... nginx-service:14 -- RUNNING RUNNING 2022-12-14 13:19:36 +... ecs-svc/591399291691... service:nginx-service FARGATE 1.4.0