

PopKachiu

High Avaliable Online Games with AWS

資訊碩二 11177035 林彥輝

Agenda

Keywords: Auto Scaling, Load Balance, Online Games
Container, Fargate



Introduction



System Architecture



Cloud Properties



Demo Video



Conclusion

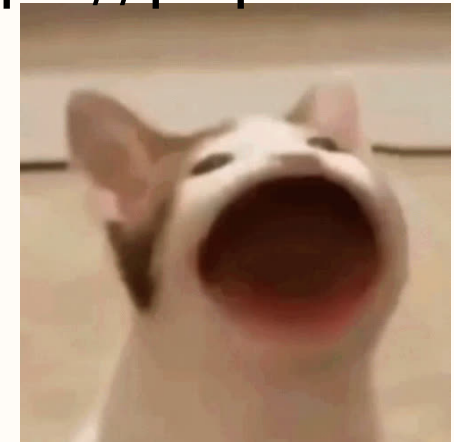


PopCats

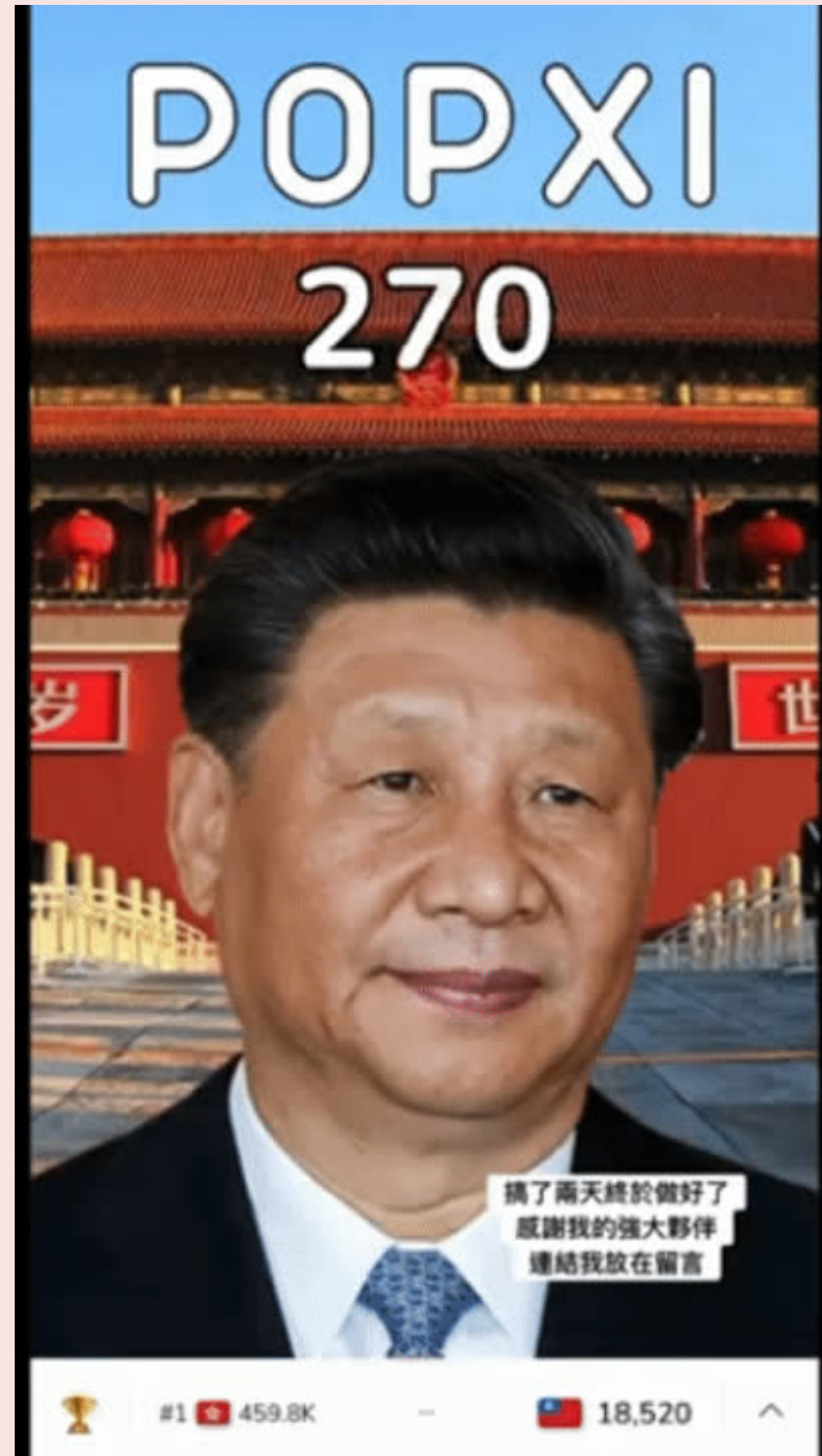
Pop Cat

- 是2021年由三名英國學生創建的網頁遊戲。
 - Department of Computer Science, University of Sheffield, UK
 - 約書亞・奧蘇利文 (Joshua O'Sullivan)
 - 愛德華・哈利斯 (Edward Halis)
 - 弗萊迪・哈佩爾 (Freddy Heppell)
- 背景
 - 來自Twitter的一則影片 (2020/10/11)
 - 原：Xavier的貓(Oatmeal)正在對蟲子鳴叫
 - 改：Xavier將此動作加工成gif圖
 - 後：被該朋友上傳到Reddit論壇上，並被網友加上Pop音效並爆紅。
- 遊戲進行方式
 - 玩家只要透過任何方式點擊圖片或按下螢幕、鍵盤等等，就會讓Pop Cat的嘴巴快速開合以獲得分數。
- 遊戲經歷
 - 2021年5月上線
 - 排名前列國家：東北歐國家 (芬蘭)
 - 2021年8月1日
 - 台灣爆紅，超越芬蘭為第一
 - 泰國也加入遊戲，其原因為反政府抗議活動，使世界了解到他們的處境

<https://popcat.click/>



Pop Xi



上線時間

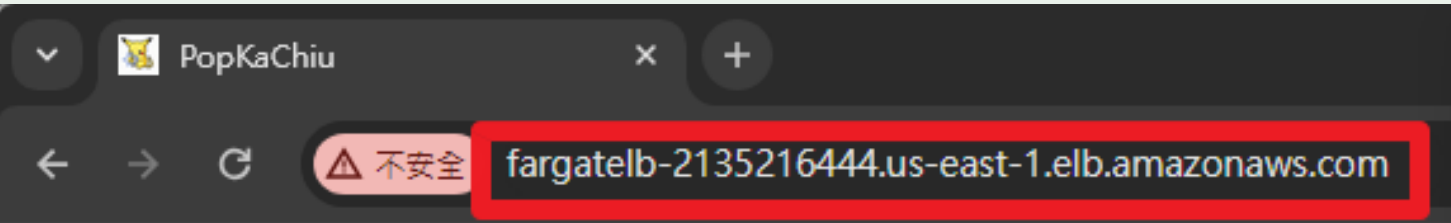
最早追朔：2021年8月（Retired）



遊戲方式

玩家只要透過任何方式點擊圖片或按下螢幕、鍵盤等等，就會讓Pop Xi的表情快速變換，並聽見「我們懷念他」。

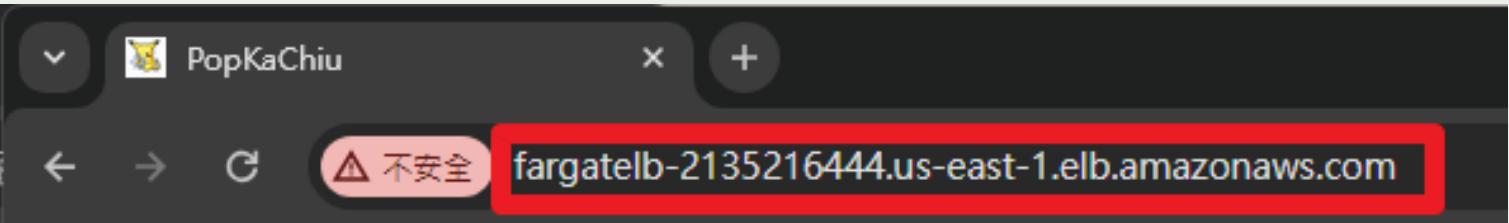
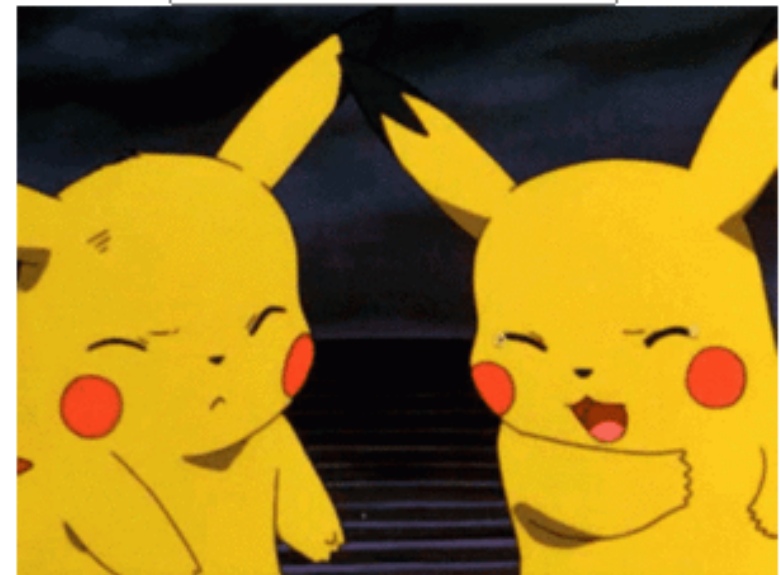
Pop KaChiu



11177035 AWS Final - PopKaChiu

VM IP: 172.31.35.206

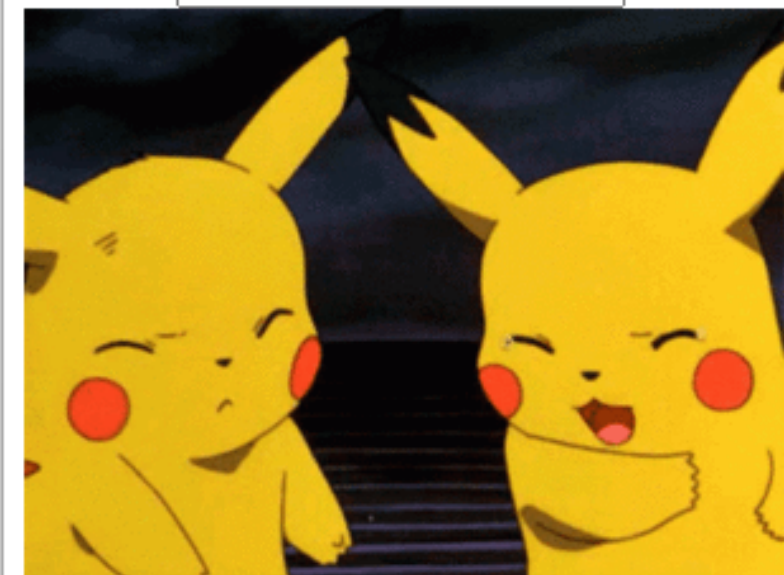
User ID:



11177035 AWS Final - PopKaChiu

VM IP: 172.31.76.164

User ID:

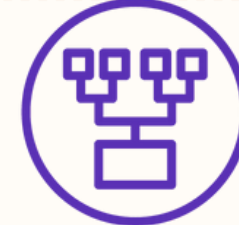
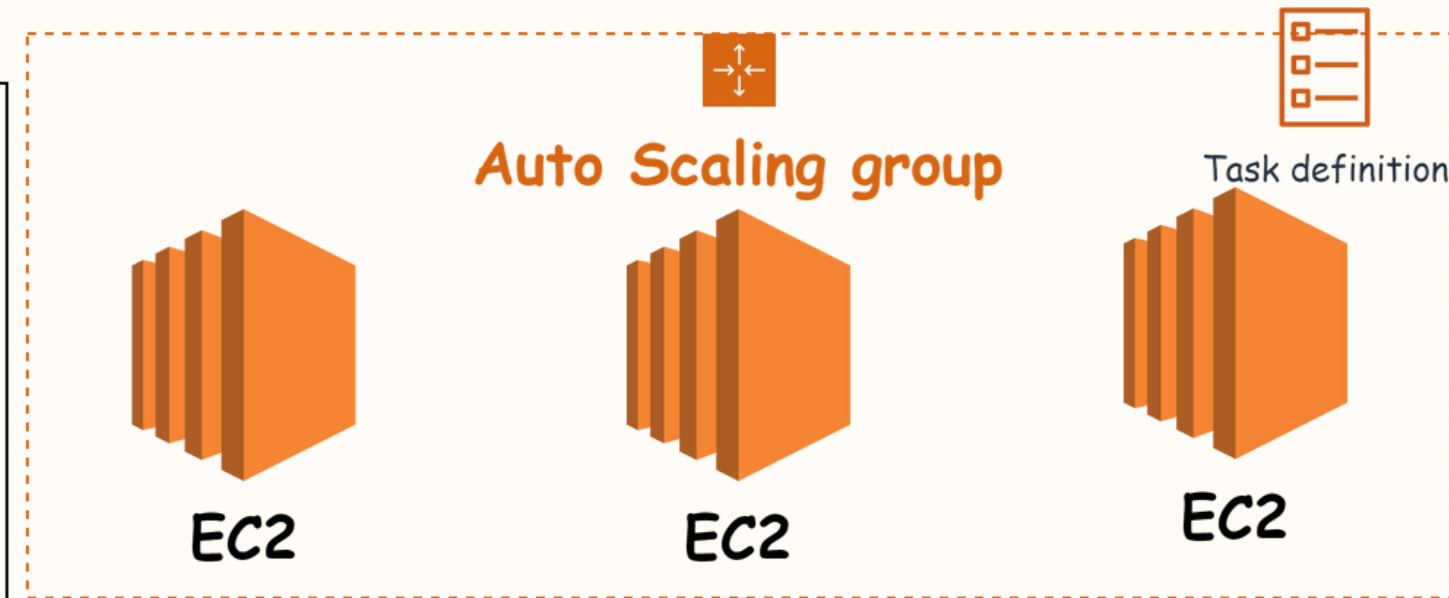


ID	Sum
test-72	41
test-489	7
test-24	25
test-164	7
test-400	7
test-255	7
test-347	7
test-151	7
test-214	7
test-239	7
test-450	7
test-167	7
test-386	7

System Architecture

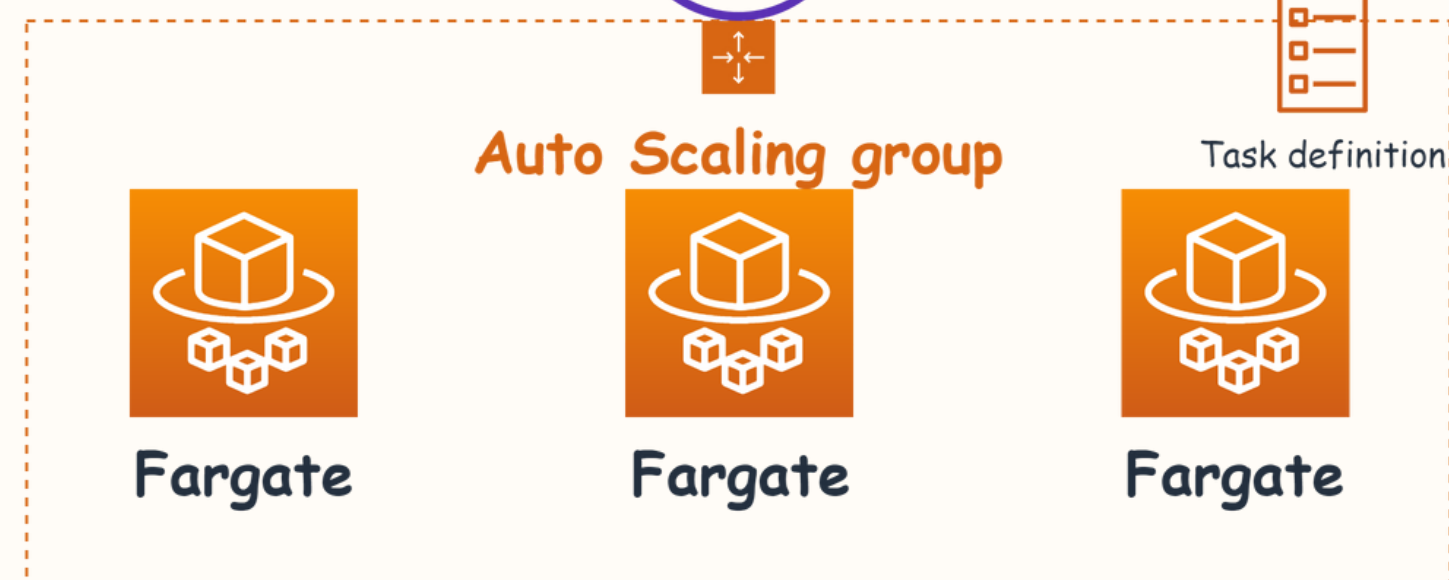


```
#!/bin/bash
sudo yum update -y
sudo amazon-linux-extras install docker -y
sudo service docker start
sudo docker pull phil890310/awsfinal
sudo docker run -d \
  -p 80:80 \
  --cpus=1 \
  --memory=1000m \
  phil890310/awsfinal
```



FinalELB-1316462922.us-east-1.elb.amazonaws.com

fargatelb-2135216444.us-east-1.elb.amazonaws.com



Lambda

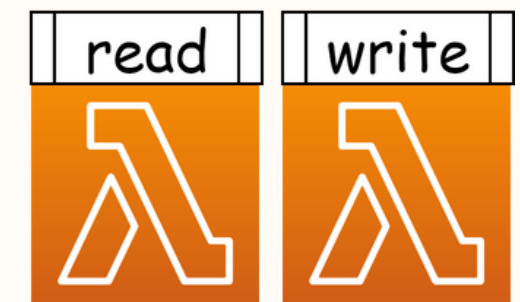
StressTest



ECR



S3 Bucket

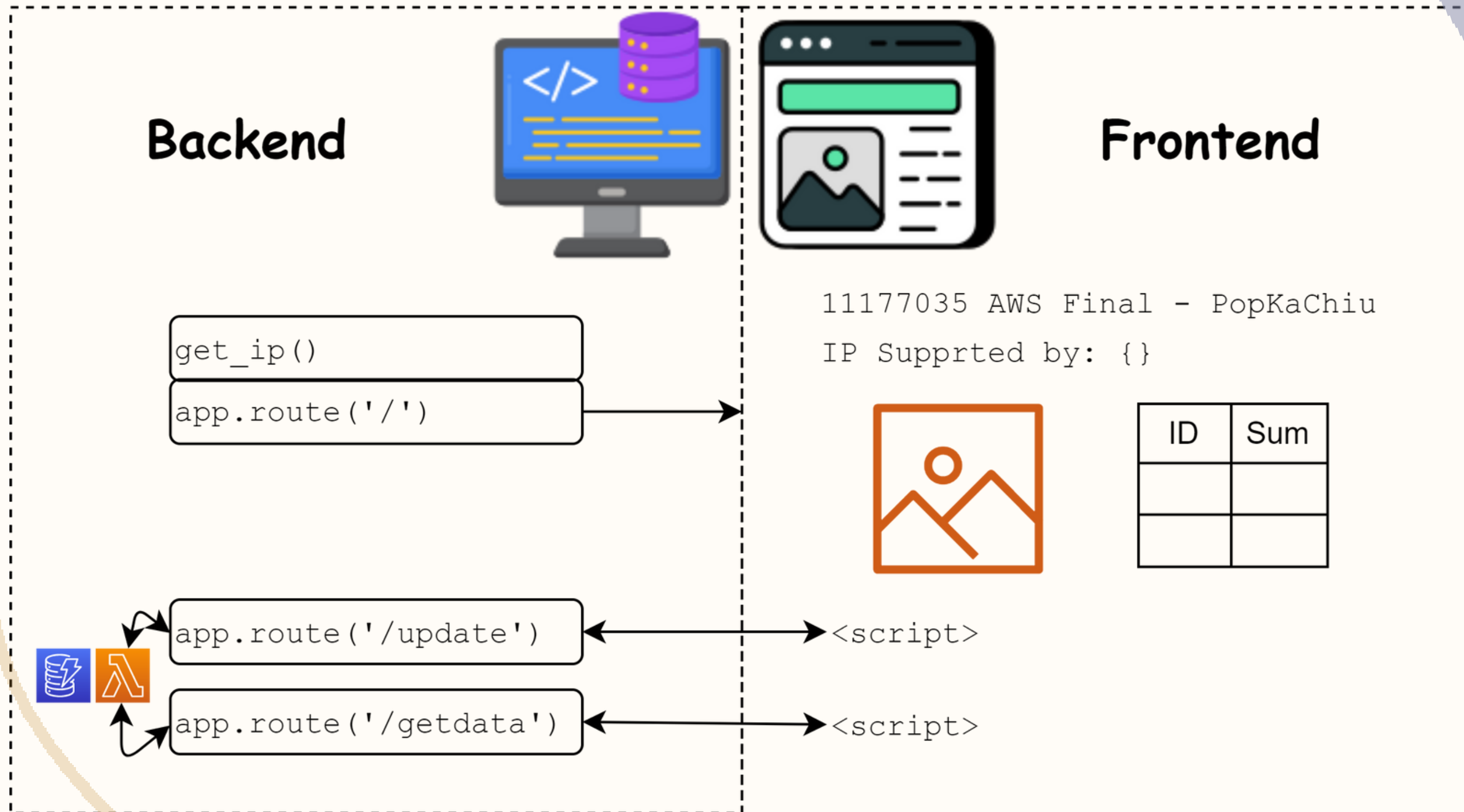


Lambda Lambda



DynamoDB

Web



Auto Scaling

部署 (1) 資訊

Q 篩選部署

< 1 > ⚙

開始日期	狀態	失敗的任務	任務	版本	任務定義
2023年12月10日 16:02 (UTC+8:00)	✔ 主要 <div><div></div> 67%</div>	0	2 執行中 1 待處理 3 所需的	1.4.0	fargate-de

事件 (7)

Q 依值篩選事件

< 1 > ⚙

開始於	訊息	事件
2023年12月10日 16:19 (UTC+8:00)	service fargate-srv registered 1 targets in target-group fargate-targetgroup ↗	7cc0a7
2023年12月10日 16:18 (UTC+8:00)	service fargate-srv has started 2 tasks: task 8ced700902e74db2af616f23be14de77 task b6ee48ef516f4f4cbd532563ec3a11d0 .	1a784b
2023年12月10日 16:18 (UTC+8:00)	訊息：Successfully set desired count to 3. Waiting for change to be fulfilled by ecs. 原因：monitor alarm TargetTracking-service/fargate-cluster/fargate-srv-AlarmHigh-64d0e5f1-fd91-47f0-a939-23e0db0ca0d1 in state ALARM triggered policy scalingfargate	4846a7
2023年12月10日 16:03 (UTC+8:00)	service fargate-srv has reached a steady state.	46bfeb
2023年12月10日 16:03 (UTC+8:00)	service fargate-srv deployment ecs-svc/7577812015846542049 deployment completed.	8404c7
2023年12月10日 16:03 (UTC+8:00)	service fargate-srv registered 1 targets in target-group fargate-targetgroup ↗	8a9b77
2023年12月10日 16:02 (UTC+8:00)	service fargate-srv has started 1 tasks: task aac421f0d0264316ac8327cbb0be6955 .	68e055

Stress Test

Response

```
{
  "statusCode": 200,
  "body": "\"Stress test completed successfully!\""
}
```

Function Logs

START RequestId: 242109da-6d14-4dd1-94a2-efe1e28fc2f7 Version: \$LATEST

Average Response Time: 0.3850364923477173 seconds

HTTP Status Code Statistics:

- 200: 100 times

END RequestId: 242109da-6d14-4dd1-94a2-efe1e28fc2f7

REPORT RequestId: 242109da-6d14-4dd1-94a2-efe1e28fc2f7 Duration: 38506.83 ms Billed Duration: 38507 ms Memory Size: 10240 MB Max Memory Used: 44 MB

```
import json
import time
import urllib.request
import urllib.parse

from statistics import mean

def lambda_handler(event, context):
    # url = "http://finalelb-1316462922.us-east-1.elb.amazonaws.com/update" # ?ID=1&UpdateCount=1
    url = "http://fargatelb-2135216444.us-east-1.elb.amazonaws.com//update" # ?ID=1&UpdateCount=1

    num_requests = 500
    response_times = []
    http_codes = {}

    for i in range(num_requests):
        user_id = f"test-{str(i).zfill(2)}"
        payload = {
            "ID": user_id,
            "UpdateCount": 1
        }

        url_get = url + '?ID=' + str(payload['ID']) + '&UpdateCount=' + str(payload['UpdateCount'])
        start_time = time.time()
        try:
            request = urllib.request.Request(url_get, method='GET')
            with urllib.request.urlopen(request) as response:
                response_code = response.getcode()
        except Exception as e:
            print(f"Error: {e}")
            response_code = 500

        end_time = time.time()
        response_time = end_time - start_time

        response_times.append(response_time)
        http_codes[response_code] = http_codes.get(response_code, 0) + 1

    avg_response_time = mean(response_times)

    print(f"Average Response Time: {avg_response_time} seconds")
    print("HTTP Status Code Statistics:")
    for code, count in http_codes.items():
        print(f" - {code}: {count} times")

    return {
        'statusCode': 200,
        'body': json.dumps('Stress test completed successfully!')
    }
```

Additional Configure:

- Function URL
- Timeout: 5 min
- Memory: 10240 MB

ApacheBench

```
$ sudo apt-get update  
$ sudo apt-get install -y apache2-utils
```



設定 ApacheBench

ApacheBench 可讓您設定要傳送的請求數、逾時限制和請求標頭。ab 將發送請求，等待回應（直到使用者指定的逾時），並將統計資訊輸出為報告。

您可以使用以下格式執行 ApacheBench 命令：

```
ab <OPTIONS> <WEB_SERVER_ADDRESS> /<PATH>
```

唯一必要的參數是 Web 伺服器的位址，後面跟著 /（沒有尾隨的 URL / 將導致 ab 傳回錯誤）和可選的 URL 路徑。如果您不指定任何選項，ApacheBench 將發送單一請求。ab 的選項可讓您調整請求量，以及（對於特殊情況）它們的標頭和請求正文。一些常用的選項包括：

- n：發送的請求數
- t：持續時間（以秒為單位），之後 ab 將停止發送請求
- c：並發請求數

如果您同時使用 -t 和 -n 標誌，請注意，-t 應始終放在前面，否則 ApacheBench 將覆蓋的值 -n 並分配預設 -n 值 50000 或 50,000 個請求。此外，ab 在單一執行緒上執行 - 該 -c 值指示 ab 一次為 TCP 連線分配多少個檔案描述符，而不是同時傳送多少個 HTTP 請求。該 -c 標誌確實允許 ab 在更短的時間內完成測試，並模擬更多數量的並發連接。例如，下面的兩個時間序列圖顯示了該命令的並發連接數以及每秒請求數：

ab -n 30000 -c 1000 http://fargatelb-2135216444.us-east-1.elb.amazonaws.com/update?ID=StressTester&UpdateCount=1

```
(base) ubuntu@ip-172-31-84-30:~$ 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 fargatelb-2135216444.us-east-1.elb.amazonaws.com (be patient)
Completed 3000 requests
Completed 6000 requests
Completed 9000 requests
Completed 12000 requests
Completed 15000 requests
Completed 18000 requests
Completed 21000 requests
Completed 24000 requests
Completed 27000 requests
Completed 30000 requests
Finished 30000 requests
```

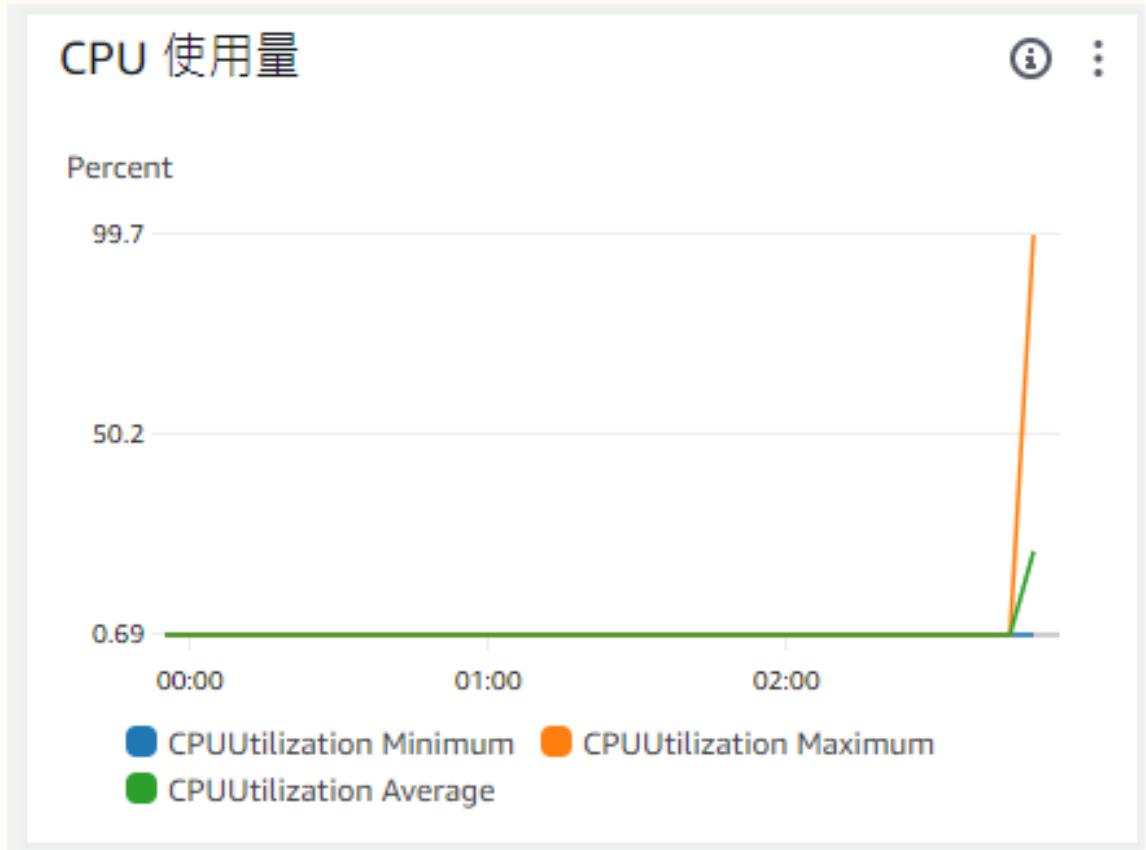
```
Server Software: Werkzeug/2.0.2
Server Hostname: fargatelb-2135216444.us-east-1.elb.amazonaws.com
Server Port: 80
```

```
Document Path: /update?ID=StressTester
Document Length: 15581 bytes
```

```
Concurrency Level: 1000
Time taken for tests: 80.488 seconds
Complete requests: 30000
Failed requests: 28214
 (Connect: 0, Receive: 0, Length: 28214, Exceptions: 0)
Non-2xx responses: 30000
Total transferred: 35852812 bytes
HTML transferred: 31273934 bytes
Requests per second: 372.72 [#/sec] (mean)
Time per request: 2682.943 [ms] (mean)
Time per request: 2.683 [ms] (mean, across all concurrent requests)
Transfer rate: 435.00 [Kbytes/sec] received
```

```
Connection Times (ms)
      min      mean[+/-sd] median    max
Connect:    0    24 116.8      8   1049
Processing:  1 1965 8914.0     14  61680
Waiting:    1 1963 8913.8     14  61680
Total:      2 1989 8913.7     24  61704
```

```
Percentage of the requests served within a certain time (ms)
 50%    24
 66%    31
 75%    38
 80%    48
 90%   3983
 95%   7812
 98%  59612
 99%  59638
100% 61704 (longest request)
```



任務 (1/3)							🔄	停止 ▼
🔍 依屬性或值篩選任務								
篩選所需狀態			篩選啟動類型				< 1 > ⚙️	
執行中 ▼			任何啟動類型 ▼					
	任務 ▼	上次狀態 ▼	所需的狀態 ▼	任務定義 ▼	運作狀態 ▼	開始於		
○	📄 a2356...	✅ 執行中	✅ 執行中	fargate-def:3	🔍 未知	18 秒前		
○	📄 b5de8...	✅ 執行中	✅ 執行中	fargate-def:3	🔍 未知	17 秒前		
●	📄 c78a1...	✅ 執行中	✅ 執行中	fargate-def:3	🔍 未知	2 分鐘前		

Size: 0.5 vCPU, 3 GB vRAM

fargate-srv 資訊

↺

更新服務

刪除服務

運作狀態和指標

任務

日誌

部署

事件

Configuration and networking

標籤

任務 (1/10)

↺

停止 ▼

🔍 依屬性或值篩選任務

篩選所需狀態

執行中 ▼

篩選啟動類型

任何啟動類型 ▼

< 1 > ⚙️

	任務 ▼	上次狀態 ▼	所需的狀態 ▼	任務定義 ▼	運作狀態 ▼	開始於
●	📄 08f1c...	⌛ 正在啟用	✔ 執行中	fargate-def:3	🔴 未知	-
○	📄 0961f...	✔ 執行中	✔ 執行中	fargate-def:3	🔴 未知	1 秒前
○	📄 0c82b...	✔ 執行中	✔ 執行中	fargate-def:3	🔴 未知	1 秒前
○	📄 125c5...	✔ 執行中	✔ 執行中	fargate-def:3	🔴 未知	10 秒前
○	📄 51bb7...	✔ 執行中	✔ 執行中	fargate-def:3	🔴 未知	8 秒前
○	📄 55cc6...	✔ 執行中	✔ 執行中	fargate-def:3	🔴 未知	11 秒前
○	📄 61d6d...	✔ 執行中	✔ 執行中	fargate-def:3	🔴 未知	8 秒前
○	📄 ad161...	✔ 執行中	✔ 執行中	fargate-def:3	🔴 未知	11 秒前
○	📄 dcc3fd...	✔ 執行中	✔ 執行中	fargate-def:3	🔴 未知	13 秒前
○	📄 f108e...	✔ 執行中	✔ 執行中	fargate-def:3	🔴 未知	11 秒前

▼ Service Auto Scaling - 選用

在指定的範圍內，自動向上和向下調整服務所需的計數，以回應 CloudWatch 警報。您可以隨時修改 Service Auto Scaling 組態，以符合應用程式的需求。

☒ 使用 Service Auto Scaling

設定 Service Auto Scaling 以調整服務所需的計數

任務數量下限

Service Auto Scaling 可調整的服務所需計數下限。

5

任務數量上限

Service Auto Scaling 可調整的服務所需計數上限。

10

擴展政策

移除

[Amazon Elastic Container Service](#) > [叢集](#) > [fargate-cluster](#) > [服務](#) > [fargate-srv](#) > 任務

fargate-srv 資訊

運作狀態和指標 任務 日誌 部署 事件 Configuration and networking 標籤

任務 (1/9)

↺

停止 ▼

🔍 依屬性或值篩選任務

篩選所需狀態

執行中 ▼

篩選啟動類型

任何啟動類型 ▼

< 1 > ⚙️


```
(base) ubuntu@ip-172-31-84-30:~$ ab -n 30000 -c 1000 http://fargatelb-2135216444.us-east-1.elb.amazonaws.com/update?ID=StressTester&UpdateCount=1
[1] 1724
(base) ubuntu@ip-172-31-84-30:~$ 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 fargatelb-2135216444.us-east-1.elb.amazonaws.com (be patient)
Completed 3000 requests
Completed 6000 requests
Completed 9000 requests
Completed 12000 requests
Completed 15000 requests
Completed 18000 requests
Completed 21000 requests
Completed 24000 requests
Completed 27000 requests
Completed 30000 requests
Finished 30000 requests

Server Software: Werkzeug/2.0.2
Server Hostname: fargatelb-2135216444.us-east-1.elb.amazonaws.com
Server Port: 80

Document Path: /update?ID=StressTester
Document Length: 15581 bytes

Concurrency Level: 1000
Time taken for tests: 46.813 seconds
Complete requests: 30000
Failed requests: 1811
    (Connect: 0, Receive: 0, Length: 1811, Exceptions: 0)
Non-2xx responses: 30000
Total transferred: 445152125 bytes
HTML transferred: 439437777 bytes
Requests per second: 640.84 [#/sec] (mean)
Time per request: 1560.445 [ms] (mean)
Time per request: 1.560 [ms] (mean, across all concurrent requests)
Transfer rate: 9286.21 [Kbytes/sec] received

Connection Times (ms)
      min      mean[+/-sd] median   max
Connect:    1       7   6.1      8    44
Processing:  1 1112 2170.8    388  33716
Waiting:    1 1107 2170.5    380  33714
Total:      2 1119 2170.5    396  33724

Percentage of the requests served within a certain time (ms)
 50%    396
 66%    824
 75%   1165
 80%   1497
 90%   2830
 95%   5132
 98%  10002
 99%  10012
100% 33724 (longest request)
```

任務 (1/10)





















🔄 停止 ▼

🔍 依屬性或值篩選任務

篩選所需狀態 篩選啟動類型

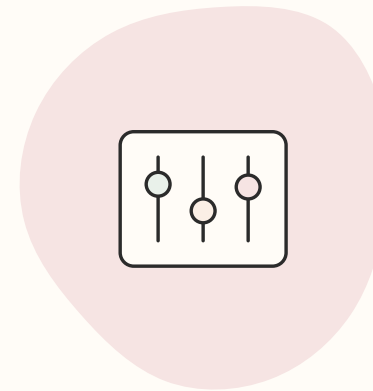
執行中 任何啟動類型

< 1 > ⚙

	任務 ▼	上次狀態 ▼	所需的狀態 ▼	任務定義 ▼	運作狀態 ▼	開始於
○	 07bb1...	🔄 正在佈建	🟢 執行中	fargate-def:5	 未知	-
○	 1fe8e...	🔄 正在佈建	🟢 執行中	fargate-def:5	 未知	-
○	 3f592...	🔄 正在佈建	🟢 執行中	fargate-def:5	 未知	-
○	 4e4ab...	🔄 正在佈建	🟢 執行中	fargate-def:5	 未知	-
○	 74029...	🔄 正在佈建	🟢 執行中	fargate-def:5	 未知	-
○	 7bfc7...	🔄 正在佈建	🟢 執行中	fargate-def:5	 未知	-
○	 a0c1d...	🔄 正在佈建	🟢 執行中	fargate-def:5	 未知	-
●	 ad088...	🟢 執行中	🟢 執行中	fargate-def:5	 未知	1 分鐘前
○	 d2175...	🔄 正在佈建	🟢 執行中	fargate-def:5	 未知	-
○	 e1246...	🟢 執行中	🟢 執行中	fargate-def:5	 未知	1 分鐘前

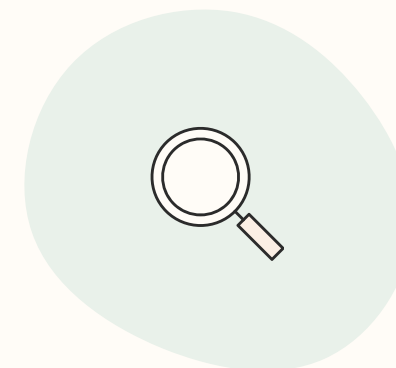
Size: 2 vCPU, 4GB vRAM

Cloud Properties



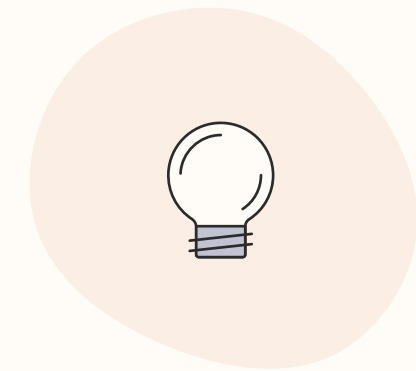
Scalability &
Optimization

ECS Auto Scaling &
Load Balancing



Manageability

Monitoring in ECS



Availability

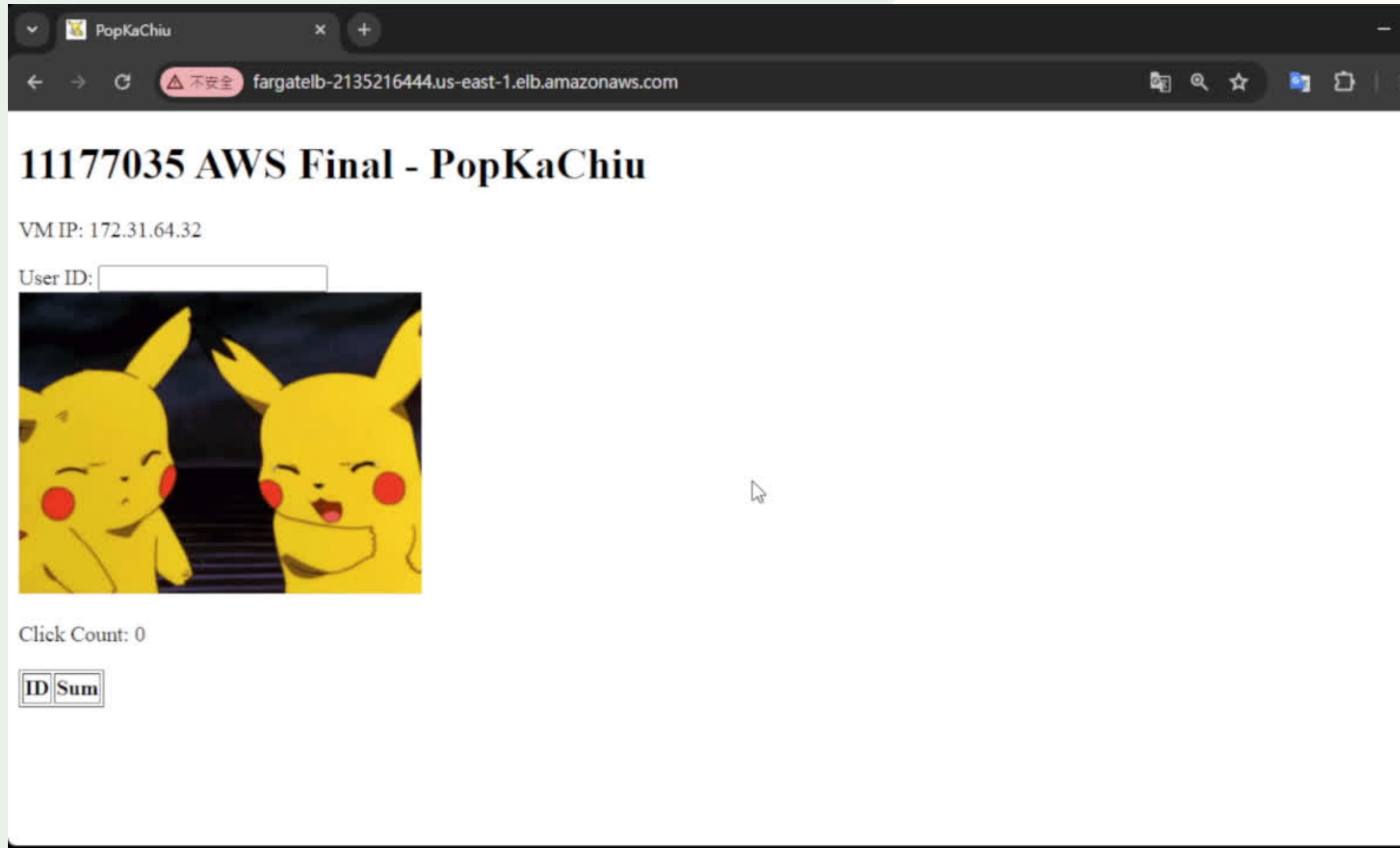
Cloud Basic



Portability

Thin Client

Demo



Other Implementation Methods?



ECS with
EC2 AutoScaling



Beanstalk



EKS

Conclusion



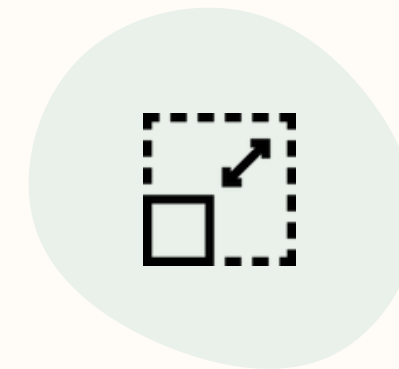
Online Games

PopKaChiu



ECS Fargate

Container, Fargate



Auto Scaling &
Load Balancing

horizontal

Thanks

