/serverless/ DAYS
Bengaluru 2024

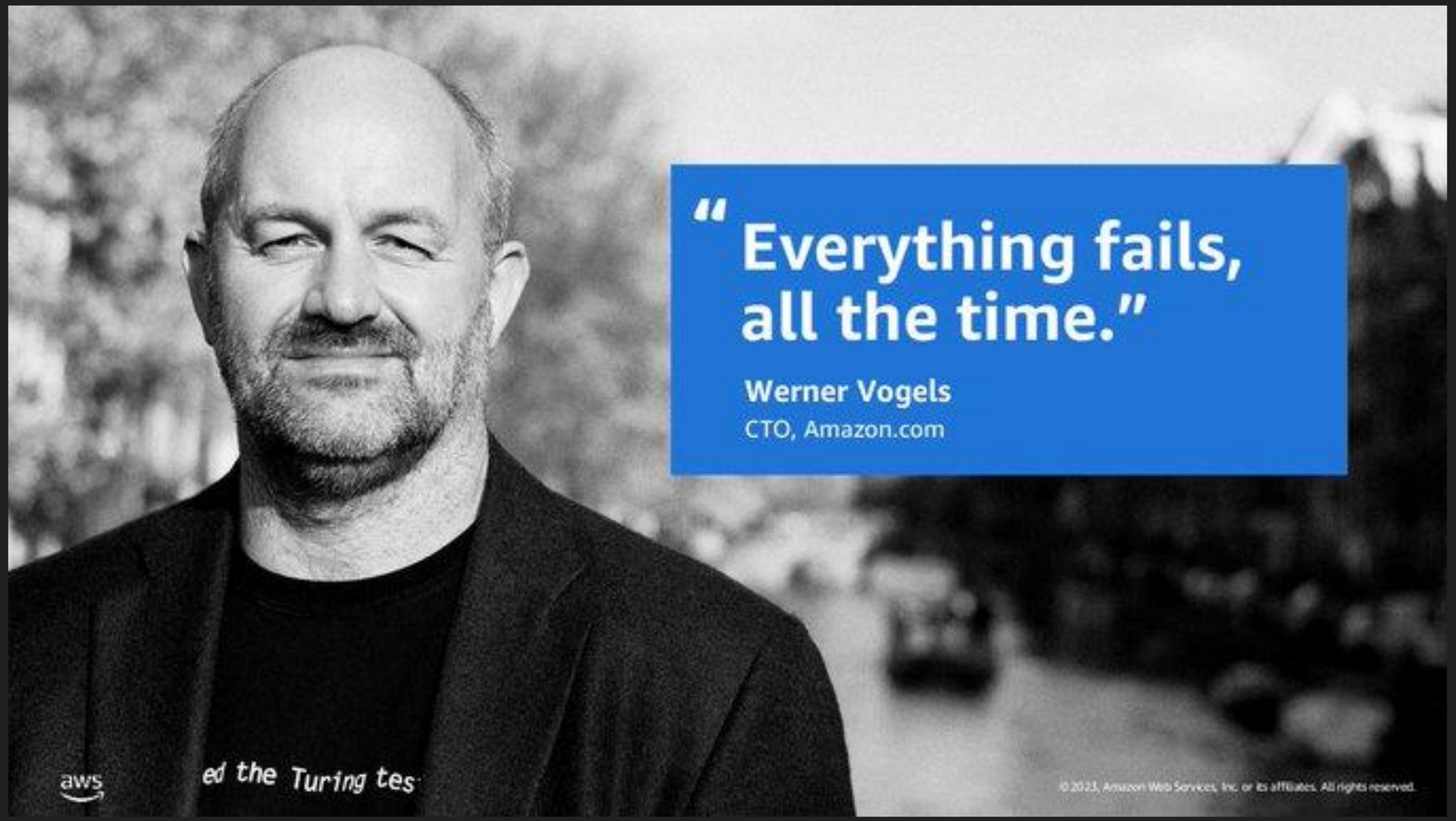
Serverless Sherlock

Unveiling the Mysteries of Fargate Debugging

Avinash Shashikant Dalvi
Senior Software Engineer @Eagleview
AWS Community Builder

/serverless/ DAYS
Bengaluru 2024

The Detective Journey


“ Everything fails, all the time.”
Werner Vogels
CTO, Amazon.com
aws
ed the Turing tes
© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The Mystery Unfolds

Symptoms

- Average response times jumped from 200ms to 1200ms within 30 minutes
- Error rates spiked by 25%
- Unexplained task restarts

With each passing minute, user complaints were piling up, and our on-call team was scrambling for answers.

/serverless/ DAYS
Bengaluru 2024

Without direct server access, where do you start solving this mystery?

Without direct server access, where do you start solving this mystery?

/serverless/ DAYS Bengaluru 2024

Why Fargate Failure is mysterious

In the world of serverless, we've traded control for convenience. But when things go wrong, we find ourselves in a fog of abstraction.

Key challenges

- Limited visibility into the underlying infrastructure
- Ephemeral nature of Fargate task
- Lack of direct access to the host
- Difficulty in local reproduction

“ Obstacles are not roadblocks, they are stepping stones.”

To solve this case, we need to gather clues using specialized tools designed for Fargate. Let’s dive into the investigation.

/serverless/ DAYS
Bengaluru 2024

The Detective's Toolkit

Essential Debugging Techniques

- Enhanced Logging
- Distributed Tracing
- CloudWatch Container Insights
- ECS Exec*
- The Status Page Technique (Hidden tool)

The Magnifying Glass

Image by Freepik

/serverless/ DAYS
Bengaluru 2024

Enhanced Logging

In Fargate, logs are our eyes and ears

Unstructured Log

```
2024-08-10 12:45:03 [INFO] User Avinash logged in from IP 192.168.1.10
2024-08-10 12:45:10 [ERROR] Failed to connect to database. Retrying...
2024-08-10 12:45:12 [INFO] Connection successful to database.
2024-08-10 12:45:15 [WARN] Response time of /api/v1/data exceeded 2 seconds.
2024-08-10 12:45:20 [INFO] User Avinash requested resource /api/v1/data
```

Structured Log


```
{
    "timestamp": "2024-08-10T12:45:03Z",
    "level": "INFO",
    "message": "User logged in",
    "user_id": "Avinash",
    "ip_address": "192.168.1.10"
}
{
    "timestamp": "2024-08-10T12:45:10Z",
    "level": "ERROR",
    "message": "Failed to connect to database",
    "retry_attempts": 1
}
{
    "timestamp": "2024-08-10T12:45:15Z",
    "level": "WARN",
    "message": "Response time exceeded threshold",
    "endpoint": "/api/v1/data",
    "response_time_ms": 2005
}
```


awslogs

Task Defination

```
{
   "logDriver":"awslogs",
   "options":{
      "awslogs-group":"awslogs-wordpress",
      "awslogs-region":"us-west-2",
      "awslogs-stream-prefix":"awslogs-example"
   }
}
```

If you check on your task's definition you'll see an

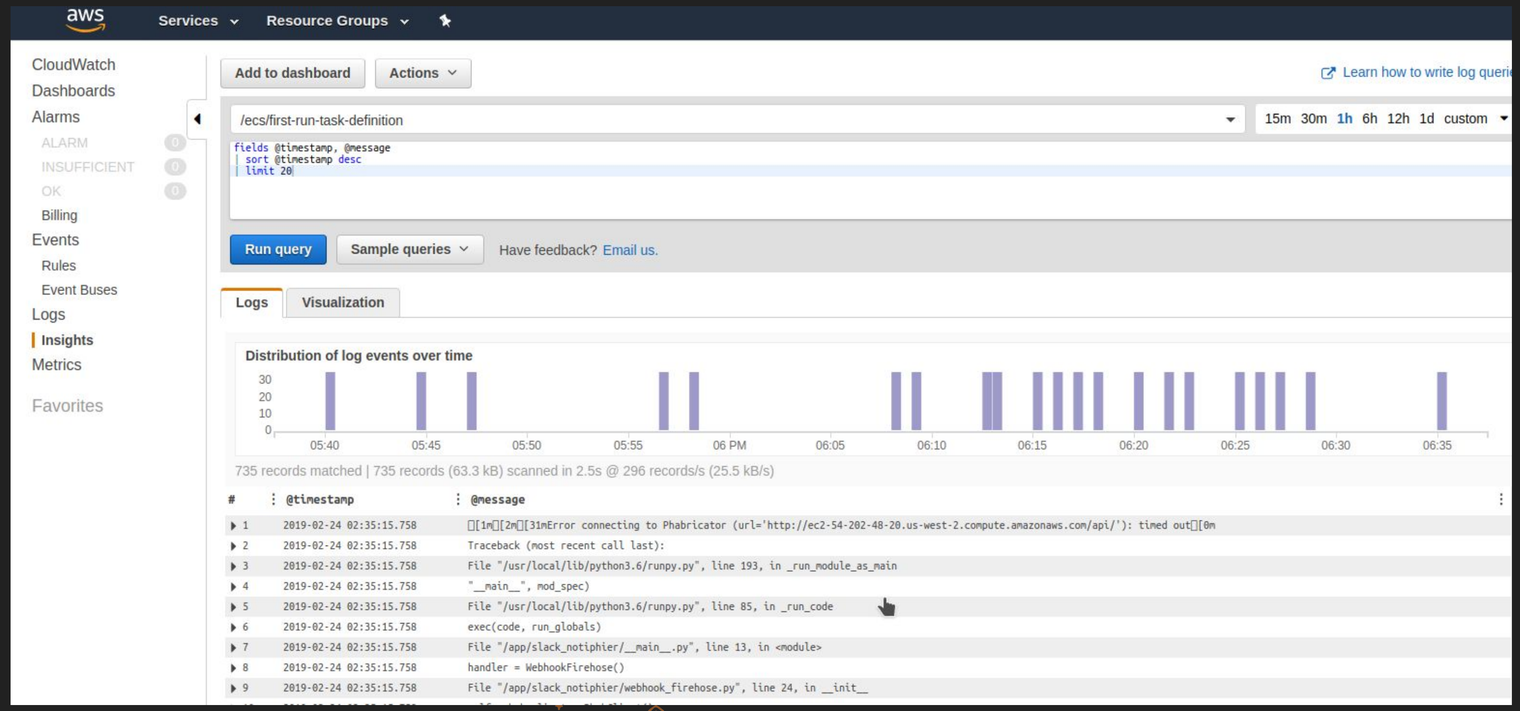
"awslogs-group".

Then take that string to the "CloudWatch -> Logs ->

Insights" and click on "Run Query"

/serverless/ DAYS
Bengaluru 2024

CloudWatch Logging

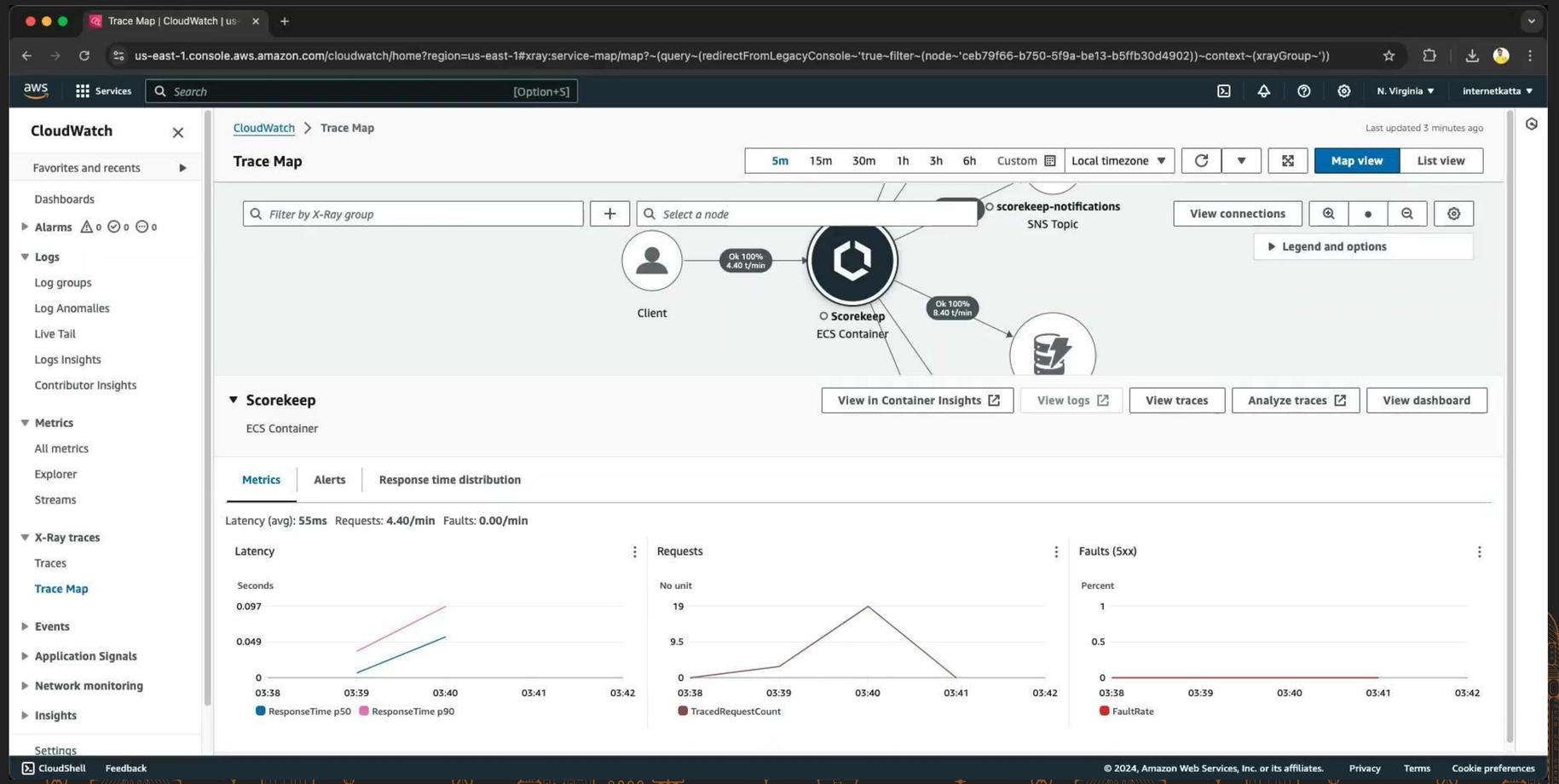


Structured Logs

- Application-Level Logging: Use logging frameworks (e.g., Winston, Logback) to output structured JSON logs.
- AWS FireLens: Route and transform logs using Fluent Bit/Fluentd; send structured logs to CloudWatch, S3, or third-party services.
- Direct to CloudWatch: Configure ECS task definition with awslogs driver to send JSON-formatted logs to CloudWatch.
- Advanced Processing: Use Lambda for custom log processing or ship logs to Amazon OpenSearch for querying and visualization.

The Chalk Lines

Distributed Tracing

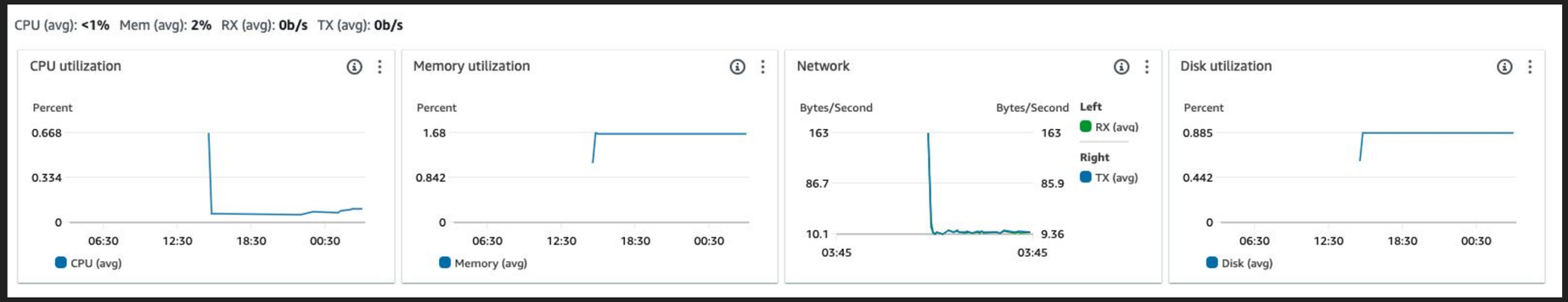


/serverless/ DAYS
Bengaluru 2024

The Surveillance Camera

CloudWatch Container Insights

- CPU utilization
- Memory usage
- Network metrics



The Close Examination

"Sometimes, we need to examine the scene up close. That's where ECS Exec comes in."

Image by Freepik

ECS Exec

```
aws ecs execute-command --cluster sample-cluster \
    --task 1a1466028dd24a97b021111c1e3da4d3 \
    --container sample-container \
    --interactive \
    --command "/bin/sh"
```

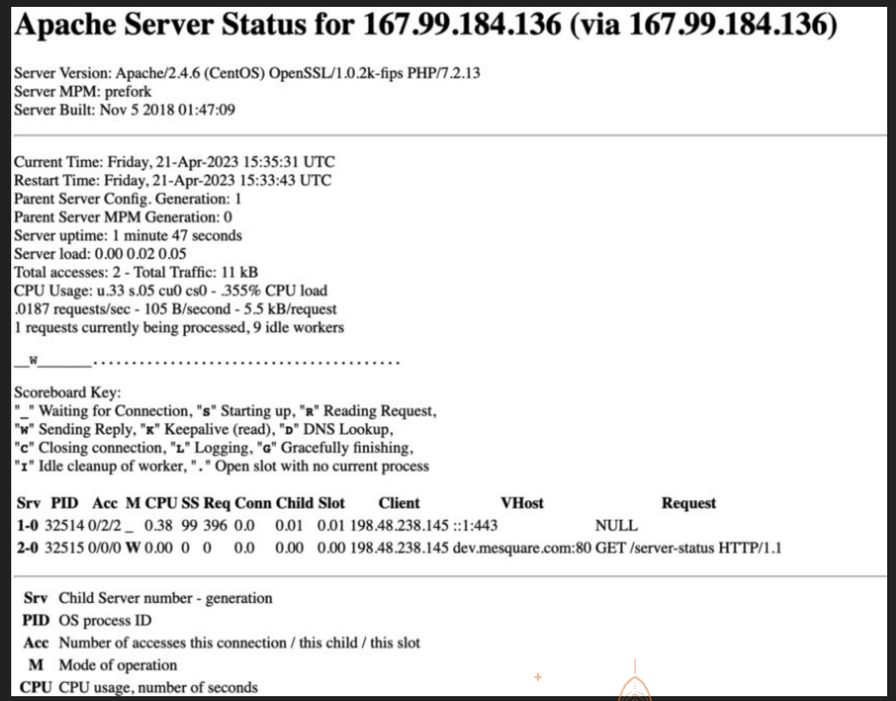
- Inspect the file system
- Check running processes
- Analyze network connections

/serverless/ DAYS
Bengaluru 2024

Pinpointing Resource-Hungry Requests

/serverless/ DAYS
Bengaluru 2024

Apache Status Page



Apache Server Status for 167.99.184.136 (via 167.99.184.136)

Server Version: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.2.13
Server MPM: prefork
Server Built: Nov 5 2018 01:47:09

Current Time: Friday, 21-Apr-2023 15:35:31 UTC
Restart Time: Friday, 21-Apr-2023 15:33:43 UTC
Parent Server Config. Generation: 1
Parent Server MPM Generation: 0
Server uptime: 1 minute 47 seconds
Server load: 0.00 0.02 0.05
Total accesses: 2 - Total Traffic: 11 kB
CPU Usage: u.33 s.05 cu0 cs0 - .355% CPU load
.0187 requests/sec - 105 B/second - 5.5 kB/request
1 requests currently being processed, 9 idle workers

_W_____.....

Scoreboard Key:
"_" Waiting for Connection, "S" Starting up, "R" Reading Request,
"W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,
"C" Closing connection, "L" Logging, "G" Gracefully finishing,
"I" Idle cleanup of worker, "." Open slot with no current process

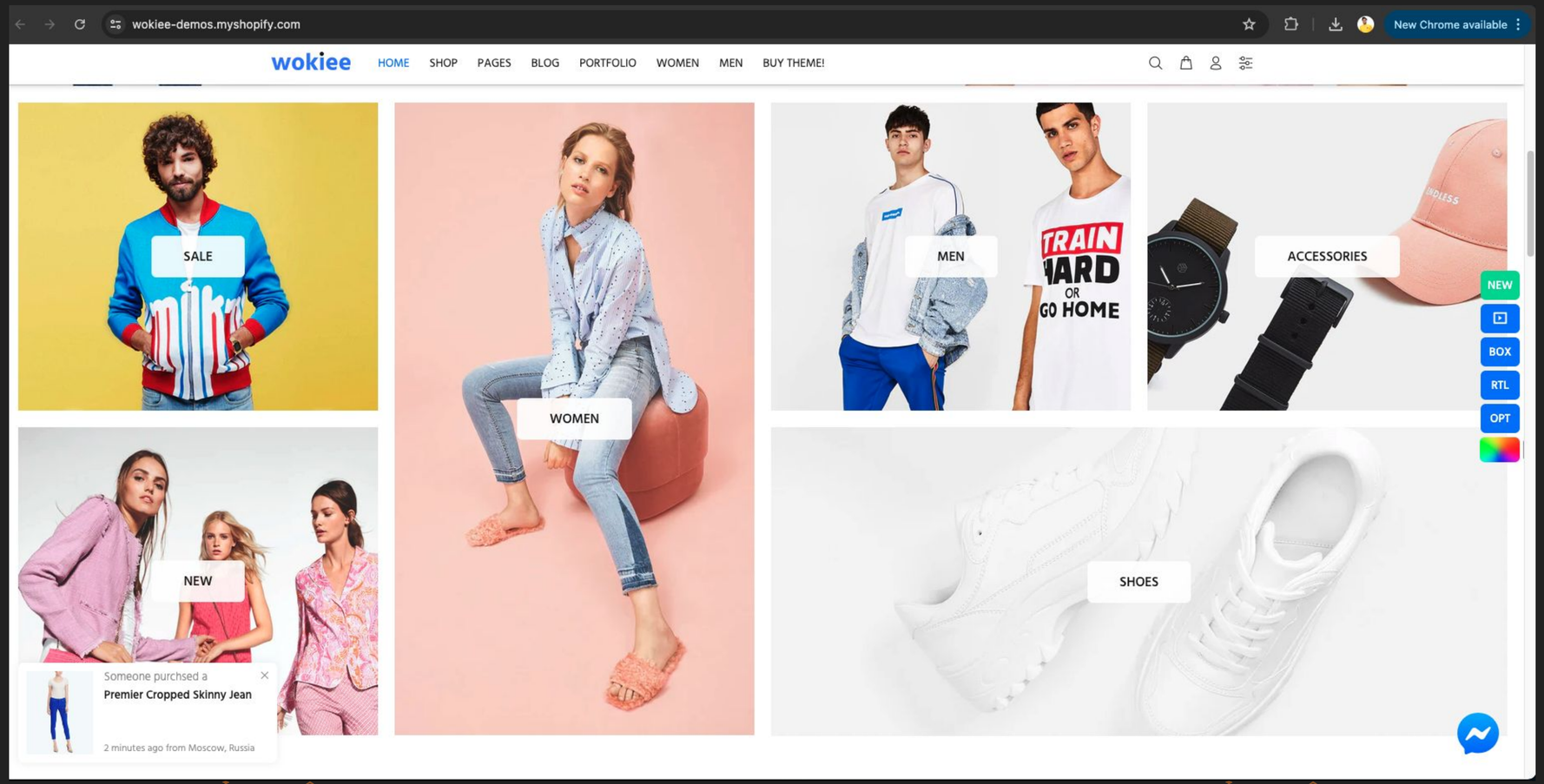
Srv	PID	Acc	M	CPU	SS	Req	Conn	Child	Slot	Client	VHost	Request
1-0	32514	0/2/2	_	0.38	99	396	0.0	0.01	0.01	198.48.238.145	::1:443	NULL
2-0	32515	0/0/0	W	0.00	0	0	0.0	0.00	0.00	198.48.238.145	dev.mesquare.com:80	GET /server-status HTTP/1.1

Srv	Child Server number - generation
PID	OS process ID
Acc	Number of accesses this connection / this child / this slot
M	Mode of operation
CPU	CPU usage, number of seconds

- Current number of requests being processed
- CPU and memory usage per request
- Request processing times
- Number of requests processed in the last minute/hour

/serverless/ DAYS
Bengaluru 2024

Solving a Real Fargate Mystery


wokiee-demos.myshopify.com
New Chrome available
wokiee
HOME
SHOP
PAGES
BLOG
PORTFOLIO
WOMEN
MEN
BUY THEME!
SALE
WOMEN
MEN
TRAIN
HARD
OR
GO HOME
ACCESSORIES
NEW
BOX
RTL
OPT
NEW
SHOES
Someone purchsed a
Premier Cropped Skinny Jean
2 minutes ago from Moscow, Russia

High latency during peak hours

Users complain about slow checkouts

Losing sales

“As any good detective knows, preventing crime is as important as solving it”

/serverless/ DAYS
Bengaluru 2024

Our investigation process

- CloudWatch Logs Insights: We identify error patterns and suspicious events.
- X-Ray traces: We follow the path of slow requests through our system.
- CloudWatch Container Insights: We notice CPU usage spikes correlating with slow periods.
- Status Page Technique: We pinpoint a specific API endpoint consuming excessive CPU time.
- ECS Exec: We examine the running tasks and discover an inefficient query causing database contention.

/serverless/ DAYS
Bengaluru 2024

Implementing Proactive Measures

/serverless/ DAYS
Bengaluru 2024

Key strategies

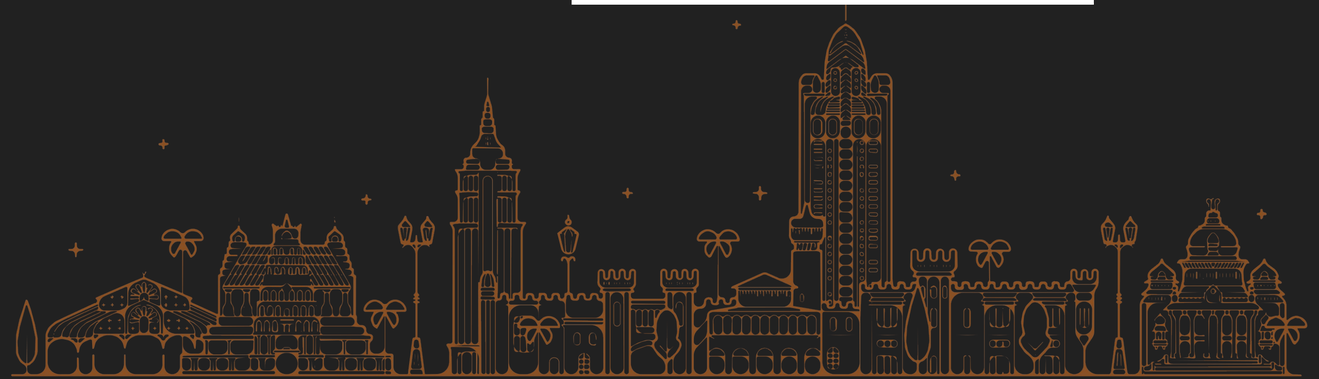
- Implement comprehensive logging, tracing, and status monitoring from the start.
- Set up proactive alerting using CloudWatch Alarms.
- Use AWS Fargate Platform Version 1.4.0 or later for enhanced debugging capabilities.
- Regularly review and optimize your task definitions.
- Embrace Infrastructure as Code for consistent, reproducible deployments.
- Implement and regularly check your application's status page for early warning signs.

“In the end, it’s not just about solving the mystery—it’s about preventing the next one.”

Thank You!

To Connect with me scan this code

@LearnwithAvinashDalvi



/serverless/ DAYS
Bengaluru 2024