

AWS와 함께하는 부하테스트 입문

from end to start

2024.02.26 @ AWSKRUG #Beginner

발표자 : 문성혁

Server Developer

모요

AUSG



(preview time)

AWS Solutions Library

Distributed Load Testing on AWS

잠재적 성능 문제를 식별할 수 있도록 대규모 기반의 로드가 있는 상태에서 소프트웨어 애플리케이션 테스트를 자동화

[구현 가이드 보기](#)

개요

Distributed Load Testing on AWS는 로드가 있는 상태에서 소프트웨어 애플리케이션 테스트를 대규모로 자동화하는 솔루션으로, 애플리케이션 릴리스 전에 잠재적 성능 문제를 식별하는 데 도움이 됩니다. 이 AWS 솔루션을 배포하면 서버를 프로비저닝하지 않고도 일관된 속도로 트랜잭션 레코드를 생성하여 수천 명의 연결된 사용자를 구축하고 시뮬레이션할 수 있습니다. 이 솔루션을 사용하여 여러 AWS 리전에서 테스트를 실행할 수도 있습니다.

이점

컨테이너를 사용하여 로드 기능 테스트

AWS Fargate 컨테이너에서 독립된 Amazon Elastic Container Service(Amazon ECS)를 사용하여 소프트웨어의 로드 기능을 테스트합니다.

애플리케이션 테스트 사용자 지정

사용자 지정 JMeter 스크립트를 사용하여 애플리케이션 테스트를 사용자 지정합니다.

로드 테스트 자동화

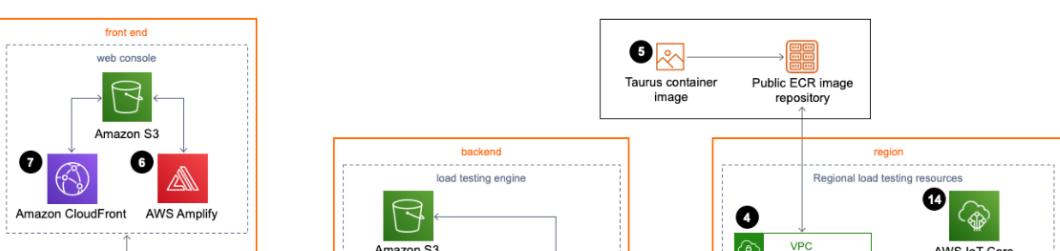
지정된 날짜 또는 반복 날짜에 자동으로 시작하도록 로드 테스트를 예약합니다.

라이브 테스트 데이터 보기

이 솔루션의 웹 콘솔을 사용하여 실행 중인 테스트의 라이브 데이터를 봅니다.

기술 세부 정보

[구현 가이드](#) 및 함께 제공되는 [AWS 리전](#) 또는 [AWS 중국 리전](#)용 AWS CloudFormation 템플릿을 사용하여 이 아키텍처를 자동으로 배포할 수 있습니다.



1단계

[Amazon API Gateway](#) API를 사용하여 이 솔루션의 마이크로서비스([AWS Lambda](#) 함수)를 간접적으로 호출합니다.

본 AWS 솔루션의 사용 사례

애플리케이션 현대화
애플리케이션 테스트
컨테이너 오케스트레이션
워크로드 및 애플리케이션

이 배포에 대한 정보

[구현 가이드 보기](#)

[구현 가이드 다운로드](#) ↗
[소스 코드](#) ↗
[CloudFormation 템플릿](#) ↗
[RSS 피드 구독](#) ↗

배포 옵션

시작할 준비가 되셨나요?

AWS Console에서 이 솔루션을 시작하여 배포

[AWS Console에서 시작](#)

Amazon Web Services 중국 리전에서 이 솔루션을 사용하고 싶으신가요? [여기클릭하세요](#).

도움이 필요하신가요? 파트너를 통해 배포하세요.

이 배포를 지원할 수 있는 AWS Certified 서드 파티 전문가를 찾아보세요.

[AWS IoT 기밀화된 환경 배포](#)

장점들

- AWS 리소스를 복합적으로 사용
- 수평 확장에 용이
- 실시간 조회 가능
- 업데이트가 꾸준히 이루어지고 있음



Search in this guide

Revisions

PDF

Date	Change
November 2019	Initial release
September 2020	Release version 1.1.0: Replaced Amazon SQS with AWS Step Functions support for JMeter scripts; for more information, refer to the CHANGELOG.md file in the GitHub repository.
December 2020	Release version 1.2.0: Added Amazon ECR checker to AWS Lambda function; for more information, refer to the CHANGELOG.md file in the GitHub repository.
April 2021	Release version 1.3.0: Added support for running concurrent tests; increased task limit to 1,000 tasks; removed concurrent test limit for AWS Lambda function.
September 2021	Release version 2.0.0: Added support to view complete test results in the AWS CloudWatch Metrics dashboard; removed requirement to manage metrics in AWS CloudWatch Metrics dashboard to show maximum data points; added support for running tests across multiple AWS Regions and availability zones. For more information, refer to the CHANGELOG.md file in the GitHub repository.
December 2021	Release version 2.0.1: Updated AWS SDK version in development environment to 2.16.100; fixed <code>ValidationException</code> error with DynamoDB. For more information, refer to the CHANGELOG.md file in the GitHub repository.
August 2022	Release version 3.0.0: Updated to AWS CDK V2, added multi-region support, and added support for AWS Lambda function bug fixes. For more information, refer to the CHANGELOG.md file in the GitHub repository.
November 2022	Release version 3.1.0: Added AppRegistry support for the AWS Lambda function.
November 2022	Release version 3.1.1: Bug fix. For more information, refer to the CHANGELOG.md file in the GitHub repository.
March 2023	Release version 3.2.0: Updated task limiting to be based on the number of concurrent tests.

Did this page help you?

Yes

No

Next topic: Notices

Previous topic: Contributors

Provide feedback

Need help?



Code

Issues 25

Pull requests 4

Discussions

Projects 1

Releases / v3.2.5

v3.2.5 Latest

kamyarz-aws released this Jan 19 <> v3.2.5 -o 78caed5 ✓

[3.2.5] - 2024-01-11

Changed

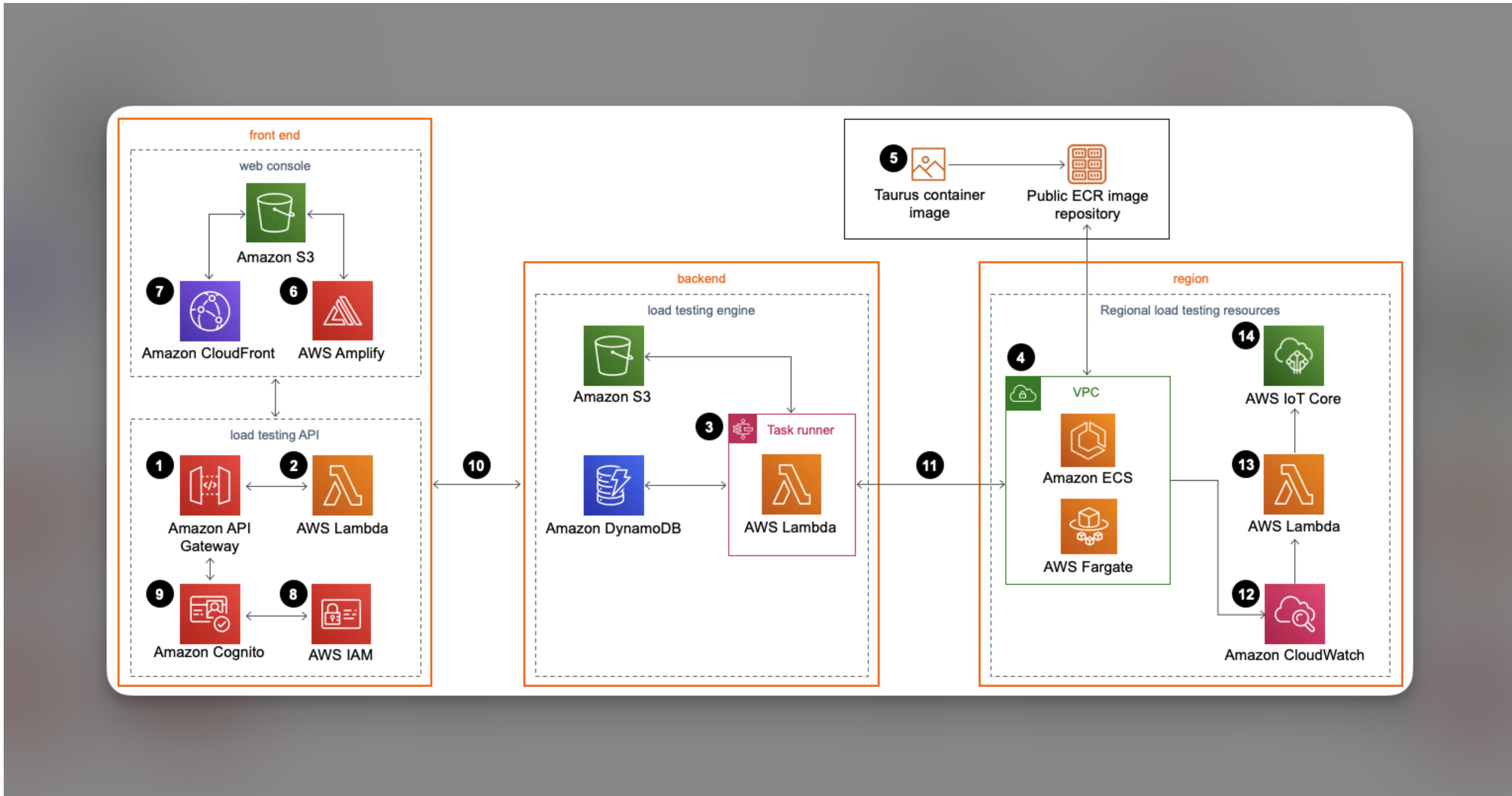
- Updated version of taurus image to 1.16.27
- Updated Jmeter dependencies and taurus dependencis within the docker image to enhance the performance.
- Updated version of "@aws-solutions-constructs/aws-cloudfront-s3" from 2.46.0 to 2.48.0
- Updated version of "@aws-cdk/aws-servicecatalogappregistry-alpha" from 2.108.0-alpha.0 to 2.110.0-alpha.0

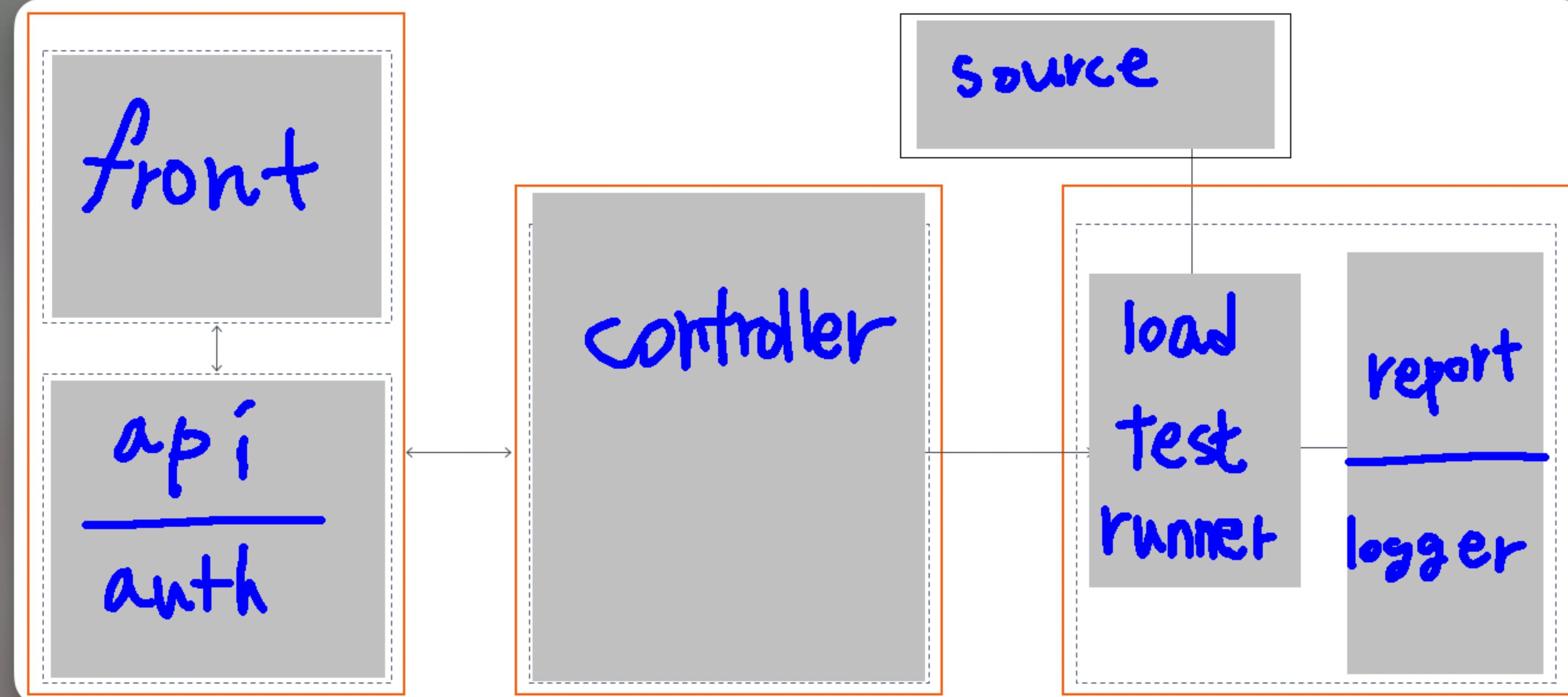
Fixed

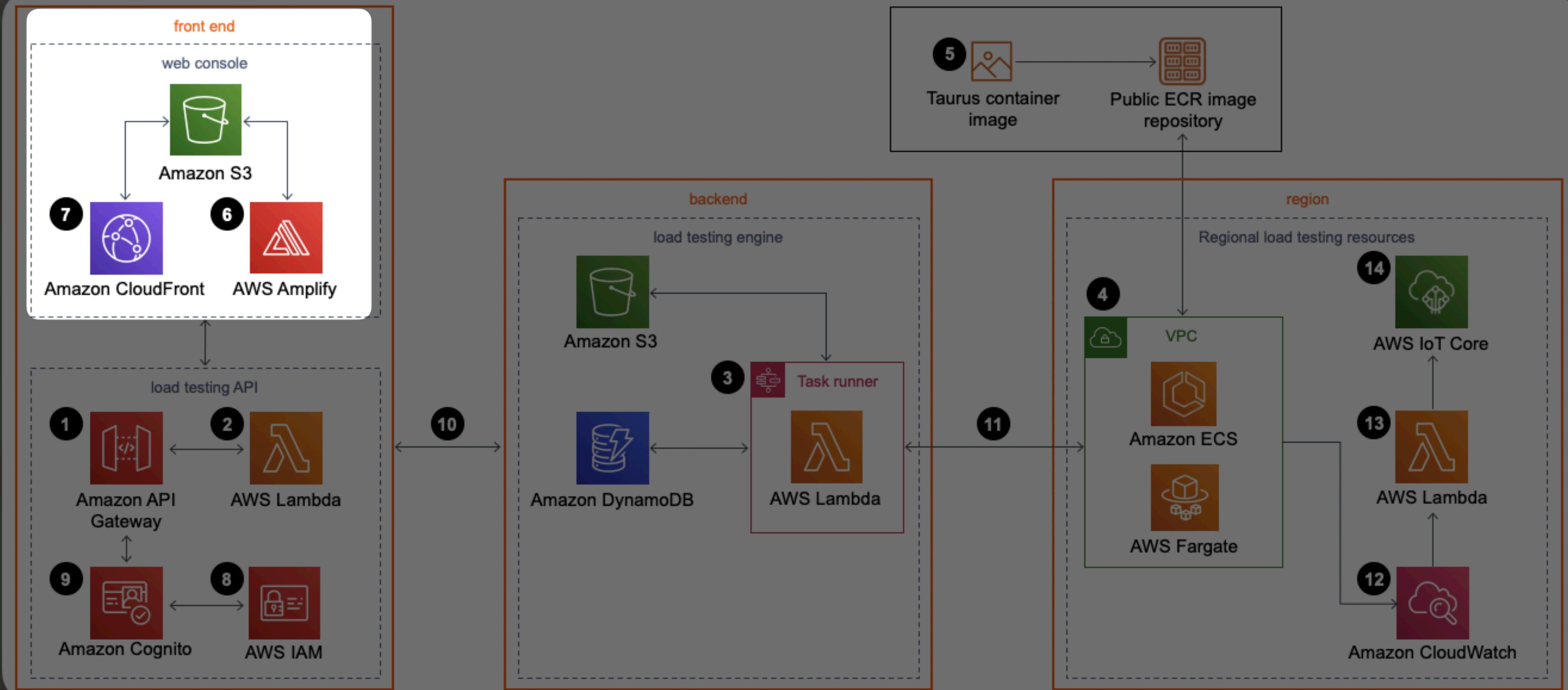
- Bug fix to resolve issues with automatic plugins installation [#152](#)

DLT를 차근차근 이해해봅시다

(Distributed Load Testing)







distributed-load-testing-on-aws / source / console / src / App.js

Code Blame 171 lines (158 loc) · 5.36 KB

```
4 import React from "react";
5 import { BrowserRouter as Router, Route, Switch, Link } from "react-router-dom";
6 import { Collapse, Navbar, NavbarToggler, NavbarBrand, Nav, NavItem } from "reactstrap";
7
8 //Amplify
9 import { Amplify, Auth } from "aws-amplify";
10 import { withAuthenticator } from "@aws-amplify/ui-react";
11 import { PubSub, AWSIoTProvider } from "@aws-amplify/pubsub";
12 import AWS from "aws-sdk";
13
14 //Components
15 import Dashboard from "./Components/Dashboard/Dashboard.js";
16 import Create from "./Components/Create/Create.js";
17 import Details from "./Components/Details/Details.js";
18 import RegionalModal from "./Components/RegionalModal/RegionalModal.js";
19
20 declare var awsConfig;
21 Amplify.addPluggable(
22   new AWSIoTProvider({
23     aws_pubsub_region: awsConfig.aws_project_region,
24     aws_pubsub_endpoint: "wss://" + awsConfig.aws_iot_endpoint + "/mqtt",
25   })
26 );
27 PubSub.configure(awsConfig);
28 Amplify.configure(awsConfig);
29
30 <div>App</div>
31 <div>constructor(props)</div>
32   super(props);
33   this.noMatch = this.noMatch.bind(this);
34   this.signOut = this.signOut.bind(this);
35   this.toggleNavbar = this.toggleNavbar.bind(this);
```

Files

main

+ Q

Go to file t

- > .github
- ✓ deployment
- > ecr/distributed-load-testing-on...

build-s3-dist.sh

run-unit-tests.sh

- ✓ source
 - > api-services
 - ✓ console
 - > public
 - ✓ src
 - > Components

App.js

index.css

index.js

manifest.json

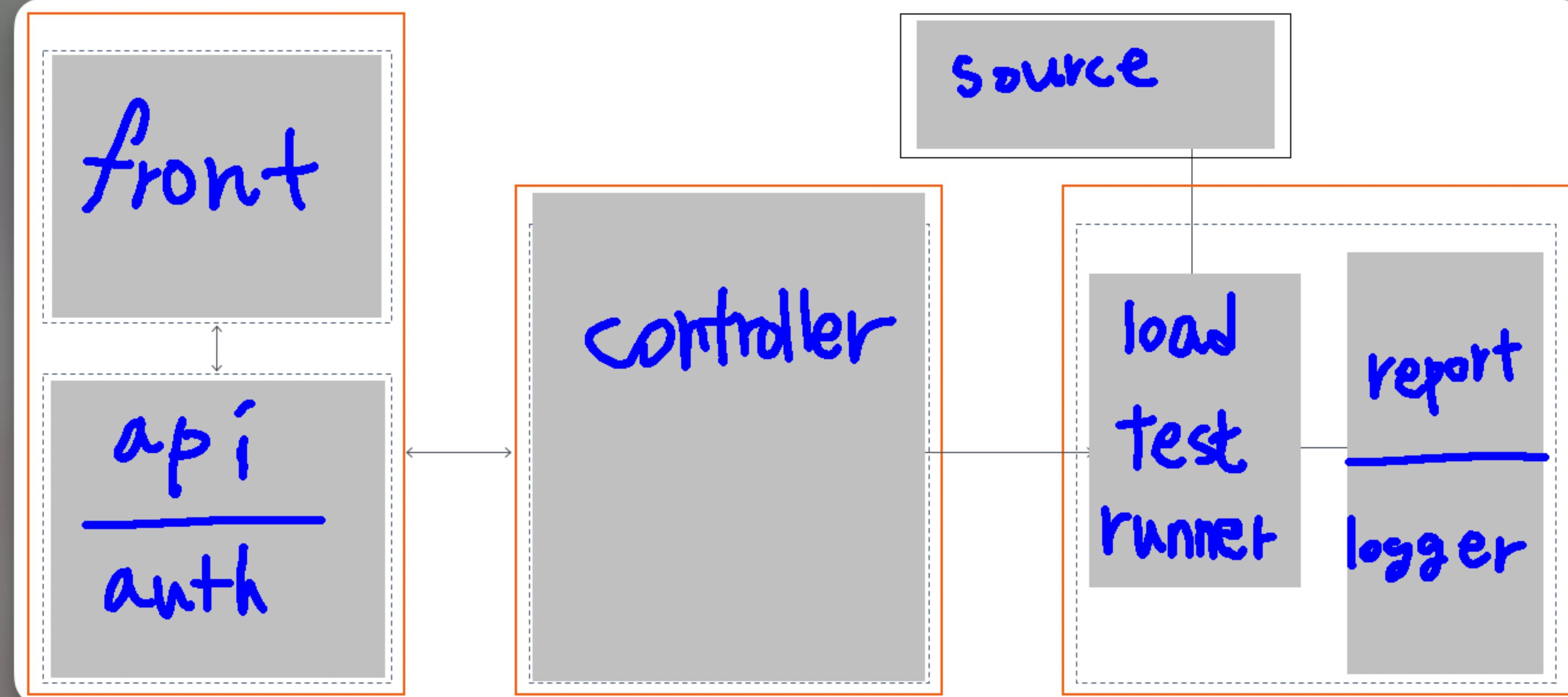
serviceWorker.js

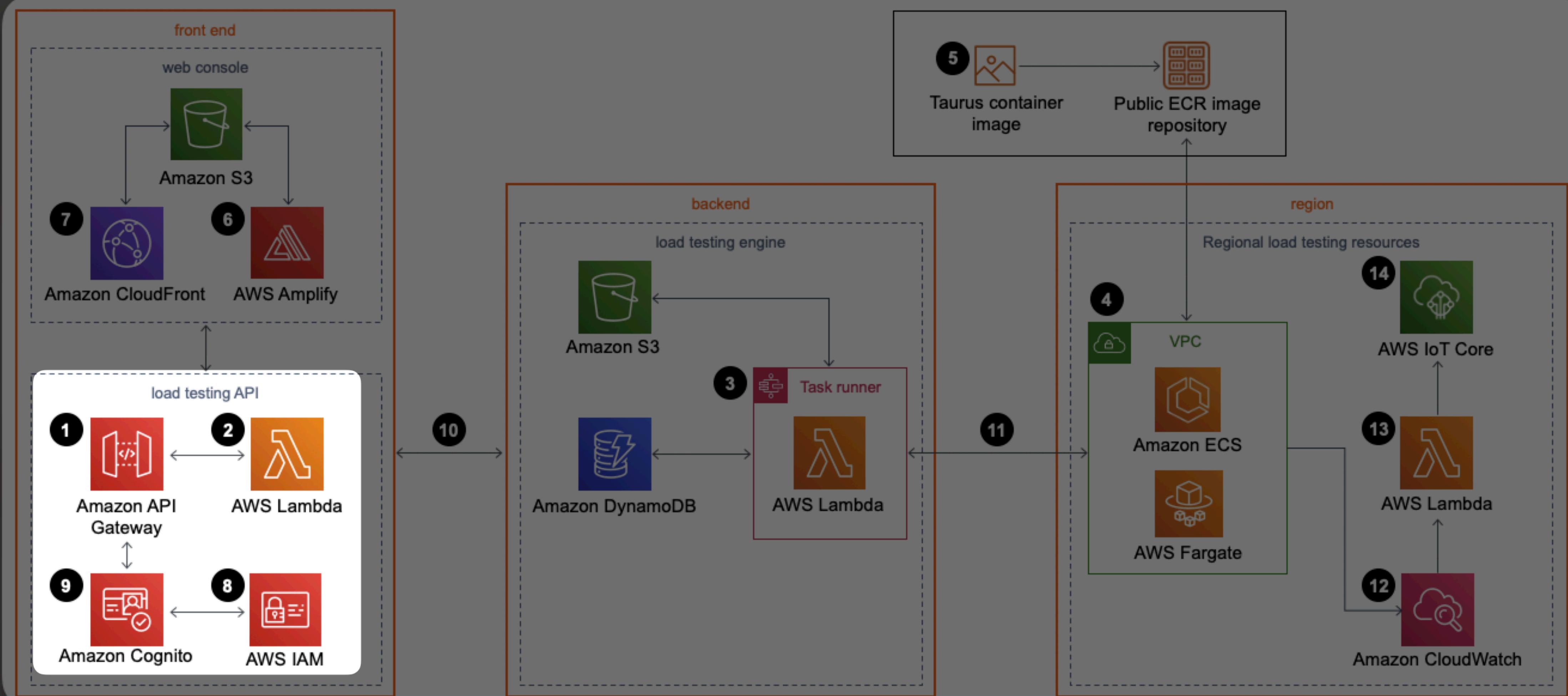
package.json

distributed-load-testing-on-aws / deployment / build-s3-dist.sh**Code****Blame**

Executable File · 175 lines (156 loc) · 6.89 KB

```
146 # Downloading jetty 9.4.53.v20231009
147 curl -O https://repo1.maven.org/maven2/org/eclipse/jetty/jetty-alpn-client/9
148 curl -O https://repo1.maven.org/maven2/org/eclipse/jetty/jetty-alpn-openjdk8
149 curl -O https://repo1.maven.org/maven2/org/eclipse/jetty/jetty-client/9.4.53
150 curl -O https://repo1.maven.org/maven2/org/eclipse/jetty/jetty-http/9.4.53.v2
151 curl -O https://repo1.maven.org/maven2/org/eclipse/jetty/jetty-io/9.4.53.v20
152 curl -O https://repo1.maven.org/maven2/org/eclipse/jetty/jetty-util/9.4.53.v
153
154 echo "-----"
155 echo "Building console"
156 echo "-----"
157 cd ${source_dir}/console
158 [ -e build ] && rm -r build
159 [ -e node_modules ] && rm -rf node_modules
160 npm install
161 npm run build
162 mkdir ${build_dist_dir}/console
163 cp -r ./build/* ${build_dist_dir}/console/ S3 directory
164
165 echo "-----"
166 echo "Generate console manifest file"
167 echo "-----"
168 cd ${build_dist_dir}/console
169 manifest=`find * -type f ! -iname ".DS_Store"`
170 manifest_json=$(IFS=,;printf "%s" "${manifest[*]}")
171 echo "[\"$manifest_json\"]" | sed 's/,/"/g' > ./console-manifest.json
172
173 echo "-----"
174 echo "Build S3 Packaging Complete"
175 echo "-----"
```





The screenshot shows the AWS API Gateway console. The top navigation bar includes the AWS logo, a services menu, a search bar with placeholder text "[Option+S]", and a breadcrumb trail "API Gateway > APIs". The left sidebar, titled "API Gateway", contains links for APIs, Custom domain names, VPC links, Usage plans, API keys, Client certificates, and Settings. The main content area, titled "APIs (1/2)", displays a table with one row. The table has columns for Name, Description, and ID. The single row shows "DLTApi" in the Name column, "Distributed Load Testing API - version v3.2.5" in the Description column, and "xkkpnpqedc" in the ID column. A search bar at the top of the main content area is set to "DLT".

Name	Description	ID
DLTApi	Distributed Load Testing API - version v3.2.5	xkkpnpqedc

The screenshot shows the AWS Amazon Cognito User Pools interface. The top navigation bar includes the AWS logo, a 'Services' dropdown, a search bar, and a keyboard shortcut '[Option+S]'. The main header 'Amazon Cognito' is followed by a breadcrumb trail 'Amazon Cognito > User pools'. On the left, a sidebar has 'User pools' selected (highlighted in blue) and 'Identity pools'. The main content area displays a callout for 'Integrate Amazon Cognito with Amazon Verified Permissions', explaining its purpose as a fine-grained authorization service. Below this, a section titled 'User pools (1) Info' shows one pool: 'roeniss-dlt-test-user-pool' (User pool name) and 'us-east-1_2WVfdAiyv' (User pool ID). A search bar at the bottom allows searching by name or ID.

User pool name	User pool ID
roeniss-dlt-test-user-pool	us-east-1_2WVfdAiyv



Services

Search

[Option+S]

Amazon Cognito



User pools

Identity pools

Amazon Cognito > Identity pools > DLTAuthIdentityPoolE110578F_BkrbAd76WahU

DLTAuthIdentityPoolE110578F_BkrbAd76WahU [Info](#)

Identity pool overview

Identity pool name

DLTAuthIdentityPoolE110578F_BkrbAd76WahU

ARN

Identity pool ID

User access

Authenticated

► Get started with your Amazon Cognito identity pool

[User statistics](#)[Identity browser](#)[User access](#)[Identity pool properties](#)[Other properties](#)

Authenticated access [Info](#)

Configure access to identity pool credentials for users who were authenticated by an identity provider.

Status

 Active

Delete all identity providers to deactivate authenticated access.

Authenticated role

roeniss-dlt-test-DLTAuthIdentityPoolE110578F_BkrbAd76WahU

Identity providers (1) [Info](#)

Manage the trusted sources of authenticated identities for your identity pool. You can issue identities to authenticated claims from social, OIDC, SAML, and custom developer authenticated sources.

 Search identity providers by name

Identity provider

[us-east-1_2WVfdAiyy](#)

▲ | Identity provider type

Amazon Cognito user pool

aws Services Search [Option+S]

Identity and Access Management (IAM) X

IAM > Roles > roeniss-dlt-test-DLTAuthDLTCognitoAuthorized-opNUAJnAvXYX

roeniss-dlt-test-DLTAuthDLTCognitoAuthorized-opNUAJnAvXYX [Info](#)

roeniss-dlt-test Identity Pool authenticated role

Summary

Creation date
February 25, 2024, 22:31 (UTC+09:00)

Last activity
 20 minutes ago

[Permissions](#) [Trust relationships](#) [Tags \(2\)](#) [Access Advisor](#) [Revoke sessions](#)

Permissions policies (2) [Info](#)

You can attach up to 10 managed policies.

Search

Policy name	Type
InvokeApiPolicy	Customer inline
IoTPolicy	Customer inline

▶ Permissions boundary (not set)

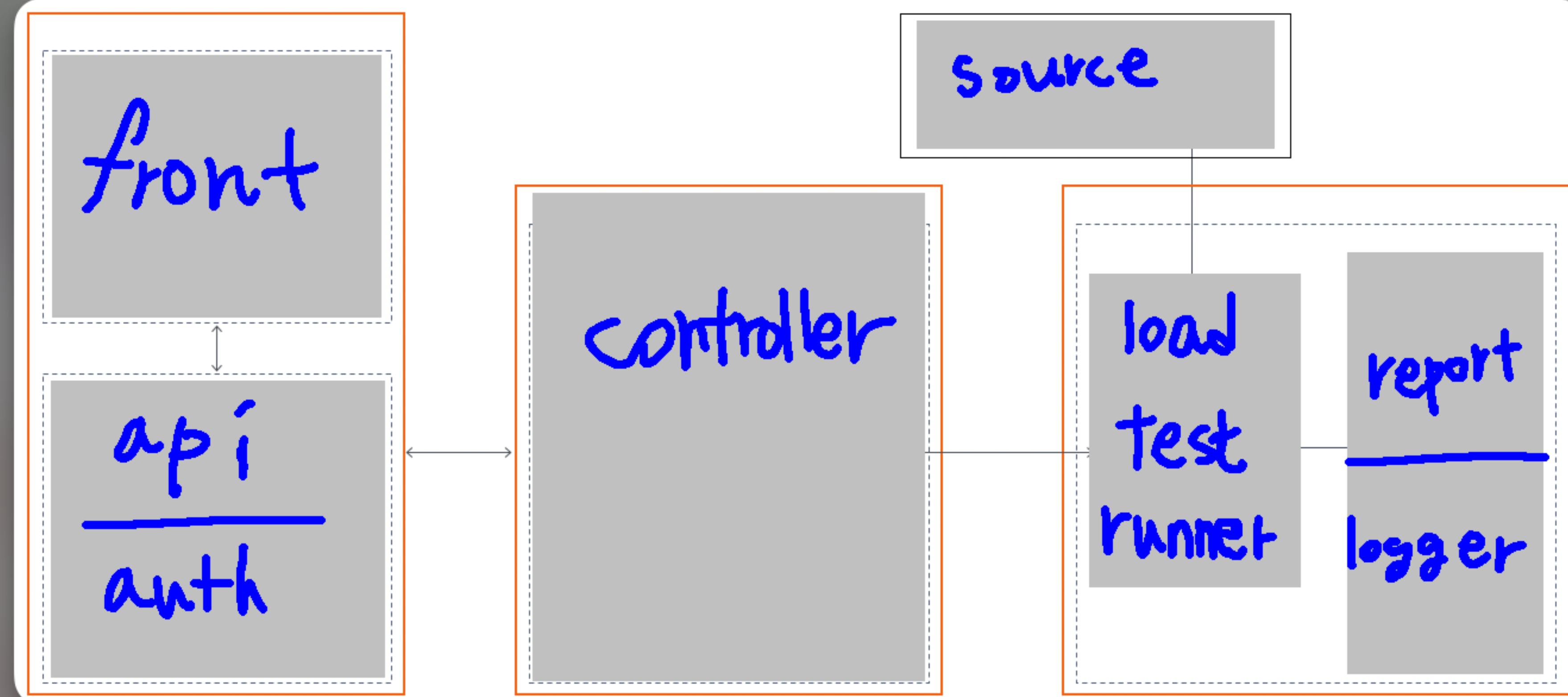
► Generate policy based on CloudTrail events

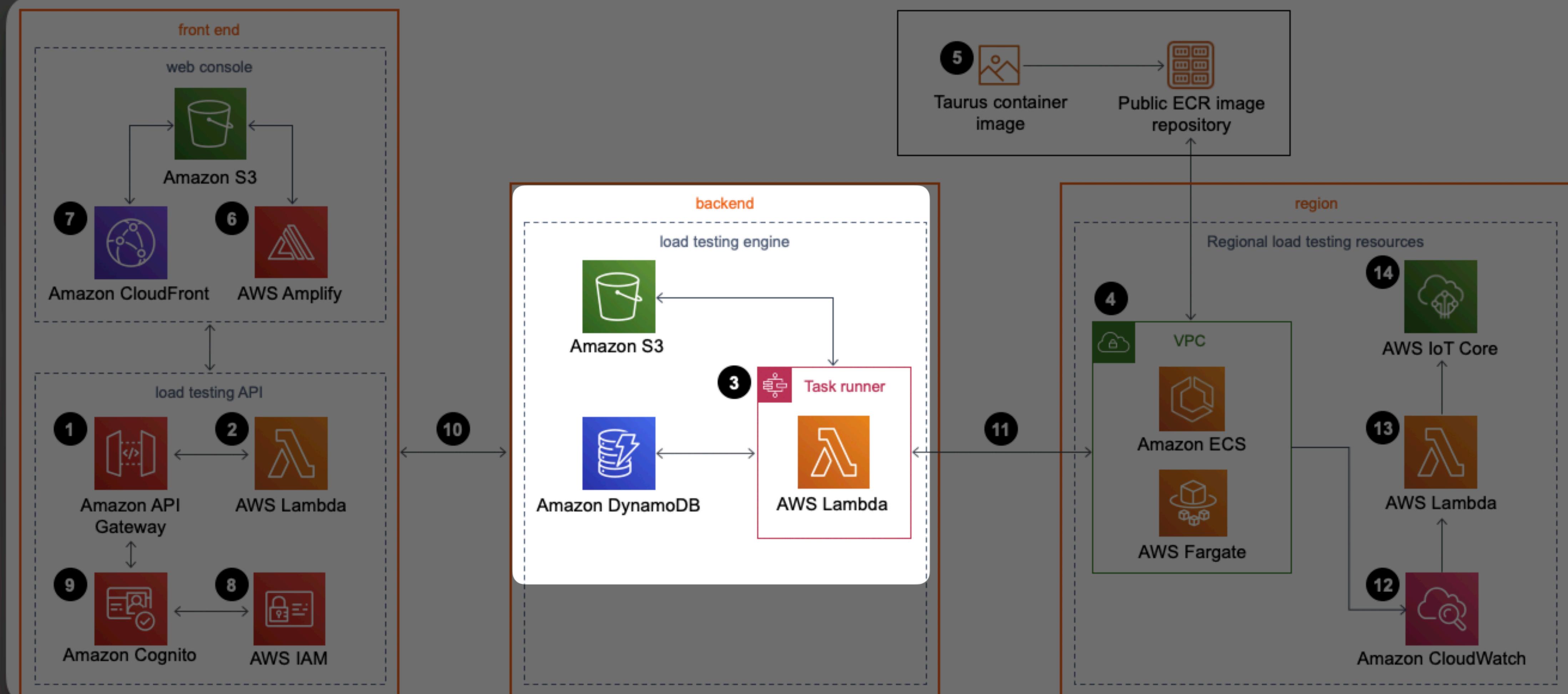
AWS Lambda < Lambda > Functions

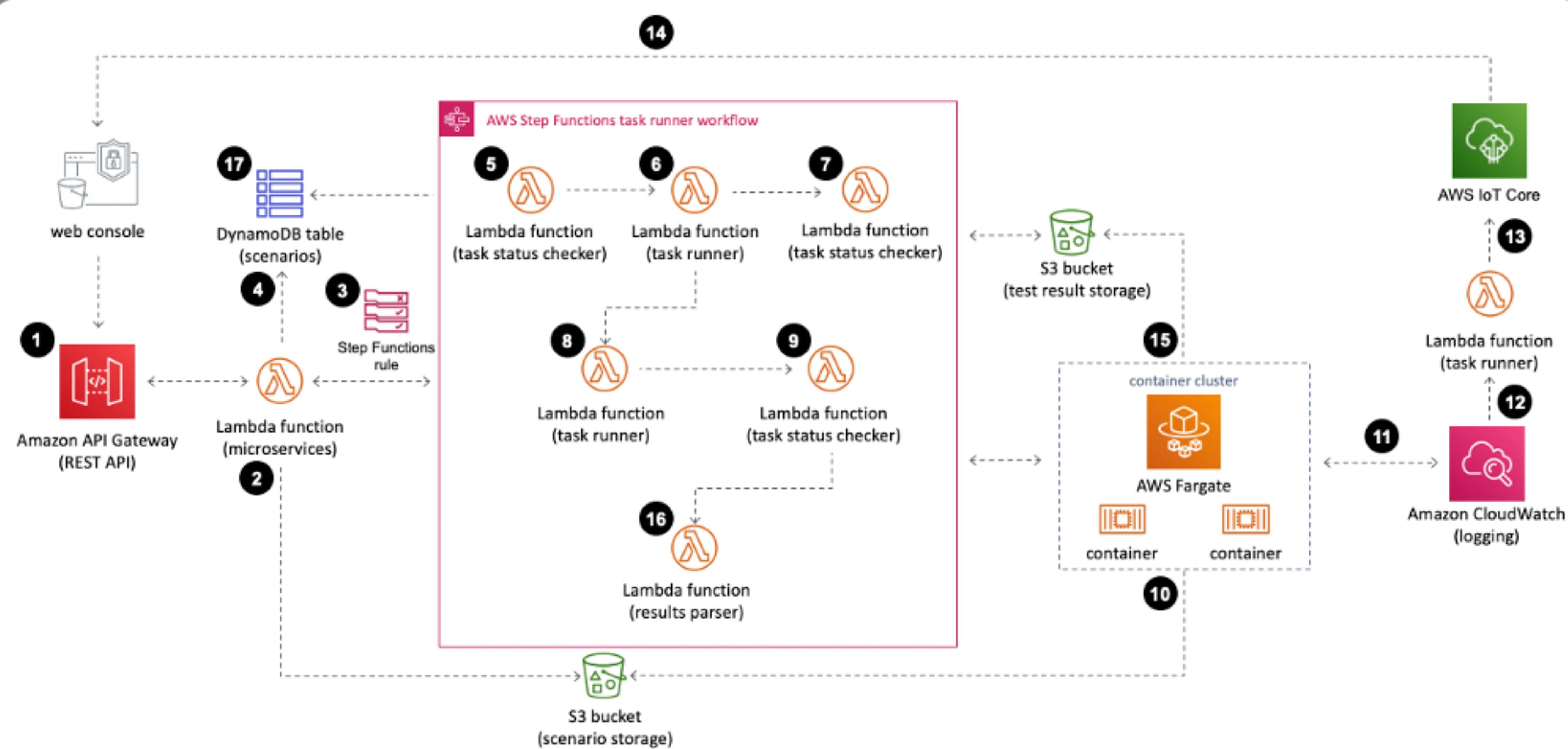
Filter by tags and attributes or search by keyword roeniss-dlt Clear filters

Matches: 7

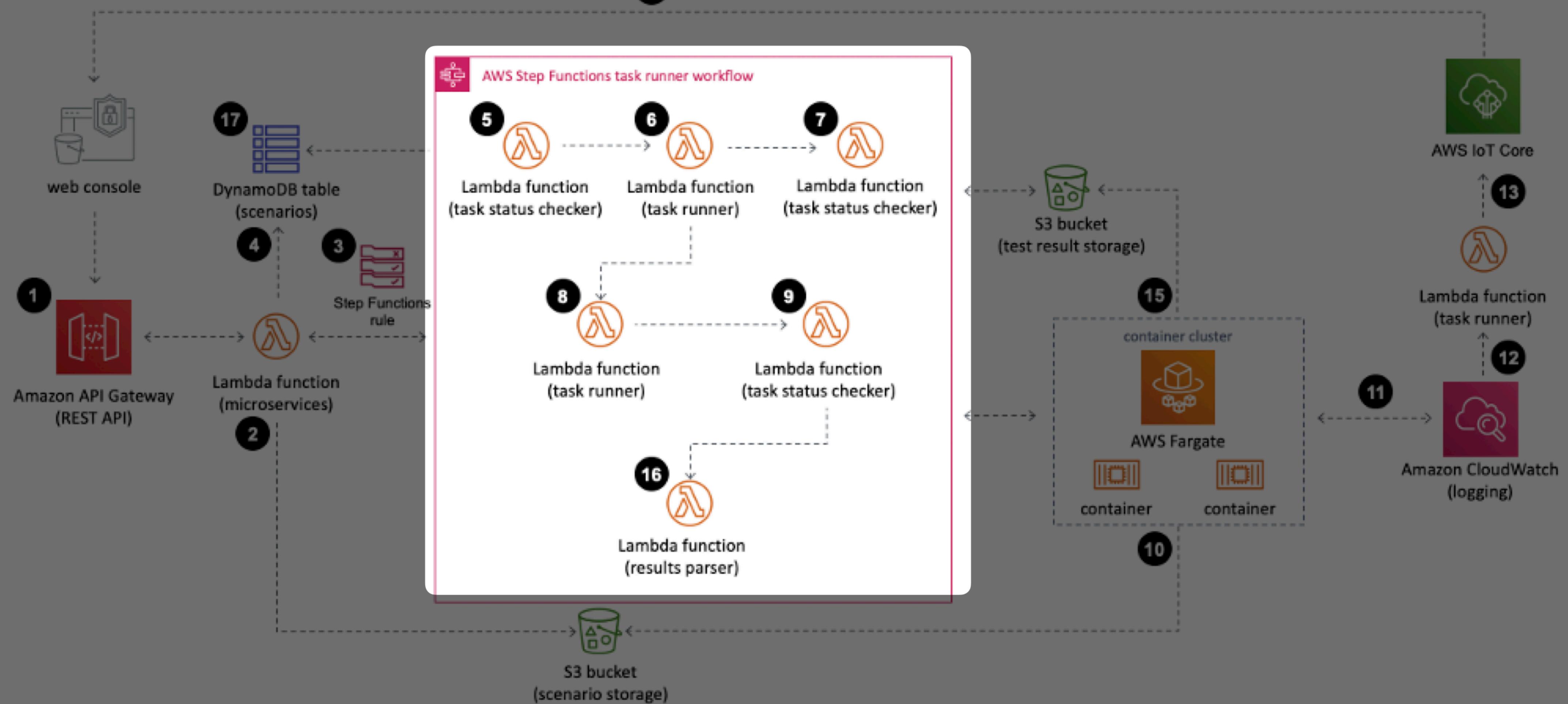
Function name	Description
roeniss-dlt-test-DLTApiDLTAPIServicesLambda9D76BA5-FjrXI8BB2FtS	API microservices for creating, updating, listing and deleting test scer
roeniss-dlt-test-DLTLambdaFunctionTaskRunnerAAD91-0k9T9YtIgAHq	Task runner for ECS task definitions
roeniss-dlt-test-DLTLambdaFunctionResultsParserFF5-fVs3UhMtElbo	Result parser for indexing xml test results to DynamoDB
roeniss-dlt-test-DLTLambdaFunctionTaskStatusChecke-TkF85XaKuVDd	Task status checker
roeniss-dlt-test-DLTLambdaFunctionTaskCanceler4E12-AGZqnfdCHhsI	Stops ECS task
roeniss-dlt-test-RealTimeDataRealTimeDataPublisher-XAXdIJAA5II43	Real time data publisher
roeniss-dlt-test-DLTCustomResourceInfraCustomResou-hWPOhqlUIh9h	CFN Lambda backed custom resource to deploy assets to s3







14



AWS Step Functions interface showing the execution history of a state machine.

Step Functions > **State machines** > State machine: DLTStepFunctionTaskRunnerStepFunctionsC295A535-f0Kp2DT7zpony

DLTStepFunctionTaskRunnerStepFunctionsC295A535-f0Kp2DT7zpony

Details

Arn: [REDACTED]
IAM role ARN: [REDACTED]

Executions (13)

Filter executions by property or value | Filter by status ▾ | Last 15 months

Name	Status	Started	End Time
f4fa9b84-a9f9-4acb-88b5-20dbdd833563	Succeeded	Feb 26, 2024, 05:13:57 (UTC+09:00)	Feb 26, 2024, 05:31:12 (UTC+09:00)
4b7d2a2d-86ff-45b3-a68c-6ed6e93202ec	Succeeded	Feb 26, 2024, 05:12:05 (UTC+09:00)	Feb 26, 2024, 05:16:22 (UTC+09:00)
df93af5e-f42a-4cb0-abe7-7df827d9eb24	Succeeded	Feb 26, 2024, 03:26:33 (UTC+09:00)	Feb 26, 2024, 03:39:53 (UTC+09:00)
6f7bc3f0-8823-4cf8-b4a7-73e43345c3f6	Succeeded	Feb 26, 2024, 02:24:09 (UTC+09:00)	Feb 26, 2024, 02:35:31 (UTC+09:00)
1d95c2a3-535e-4c88-a617-a54ca1ec04b6	Succeeded	Feb 26, 2024, 02:13:21 (UTC+09:00)	Feb 26, 2024, 02:23:38 (UTC+09:00)
c34b9c8d-e721-4c47-8eb4-498b9a16bcb	Succeeded	Feb 26, 2024, 02:03:33 (UTC+09:00)	Feb 26, 2024, 02:10:53 (UTC+09:00)
80c71b1c-7e0e-4f47-af02-615ff8de8766	Succeeded	Feb 26, 2024, 02:02:35 (UTC+09:00)	Feb 26, 2024, 02:04:48 (UTC+09:00)
aa395fb4-4ae1-4f07-9188-6b1433dd0b4d	Succeeded	Feb 26, 2024, 01:24:13 (UTC+09:00)	Feb 26, 2024, 01:44:36 (UTC+09:00)
9257863c-0e80-4e42-9b7a-b6198b80a073	Succeeded	Feb 26, 2024, 00:57:34 (UTC+09:00)	Feb 26, 2024, 01:14:00 (UTC+09:00)
f052dc88-c89e-44dd-824f-425aebdb7c06	Succeeded	Feb 26, 2024, 00:07:52 (UTC+09:00)	Feb 26, 2024, 00:17:18 (UTC+09:00)
89518936-d126-4dbe-96c7-f11351cddee8	Succeeded	Feb 26, 2024, 00:01:14 (UTC+09:00)	Feb 26, 2024, 00:04:37 (UTC+09:00)
24afd4d8-7063-415a-8f51-73011b7fe402	Failed	Feb 25, 2024, 23:55:19 (UTC+09:00)	Feb 25, 2024, 23:55:22 (UTC+09:00)
a832d94b-4324-4766-9aa7-e4d9e6d4c707	Failed	Feb 25, 2024, 23:50:05 (UTC+09:00)	Feb 25, 2024, 23:50:10 (UTC+09:00)

aws Services Search [Option+S]

Step Functions

State machines Activities Developer resources Online learning workshop Local Development Data flow simulator Feature spotlight Documentation Join our feedback panel

Execution: f4fa9b84-a9f9-4acb-88b5-20dbdd833563

Details Execution input and output Definition

Execution status: Succeeded

Execution type: Standard

Execution ARN: [REDACTED]

IAM role ARN: [REDACTED]

State transitions: 19

X-Ray trace map: 1-65db9f84-30d2835203f0527724b450ef

Execution Logs: Learn more CloudWatch Logs

Graph view Table view

Graph view

Actions ▾

```
graph TD; Start((Start)) --> Regions[Regions for testing Iteration #0]; Regions --> CheckRunning[Check running tests]; CheckRunning --> NoRunning{No running tests}; NoRunning --> RunWorkers[Run workers]; RunWorkers --> TestStillRunning[Test is still running]; TestStillRunning --> RequiresLeader{Requires leader?}; RequiresLeader --> Wait1MinWorker[Wait 1 minute - worker status]; Wait1MinWorker --> CheckWorkerStatus[Check worker status]; CheckWorkerStatus --> AreAllWorkersRunning{Are all workers running?}; AreAllWorkersRunning --> RunLeaderTask[Run leader task]; RunLeaderTask --> WaitSpecifiedDuration[Wait specified test duration]; WaitSpecifiedDuration --> CheckTaskStatus[Check task status]; CheckTaskStatus --> AreAllTasksDone{Are all tasks done?}; AreAllTasksDone --> MapEnd[Map End]; AreAllTasksDone --> Wait1MinTask[Wait 1 minute - task status]; Wait1MinTask --> ParseResult[Parse result]; ParseResult --> Done[Done];
```

In progress Failed Caught error Canceled Succeeded

Step Functions

State machines Activities

Developer resources

- Online learning workshop
- Local Development
- Data flow simulator
- Feature spotlight
- Documentation

Join our feedback panel

Execution: f4fa9b84-a9f9-4acb-88b5-20dbdd833563

Details Execution input and output Definition

Execution status: Succeeded

Execution type: Standard

Execution ARN: [REDACTED]

IAM role ARN: [REDACTED]

State transitions: 19

X-Ray trace map: 1-65db9f84-30d2835203f0527724b450ef

Execution Logs: CloudWatch Logs

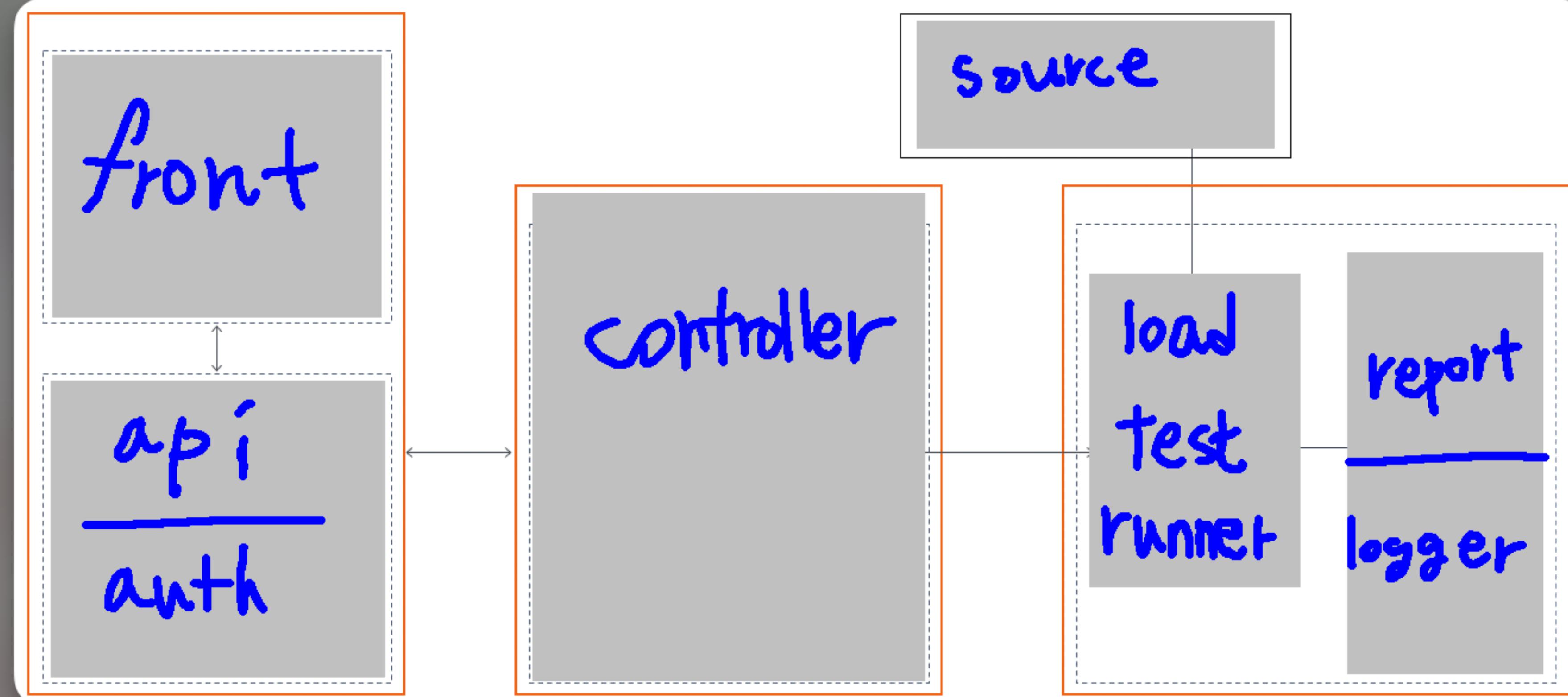
Graph view Table view

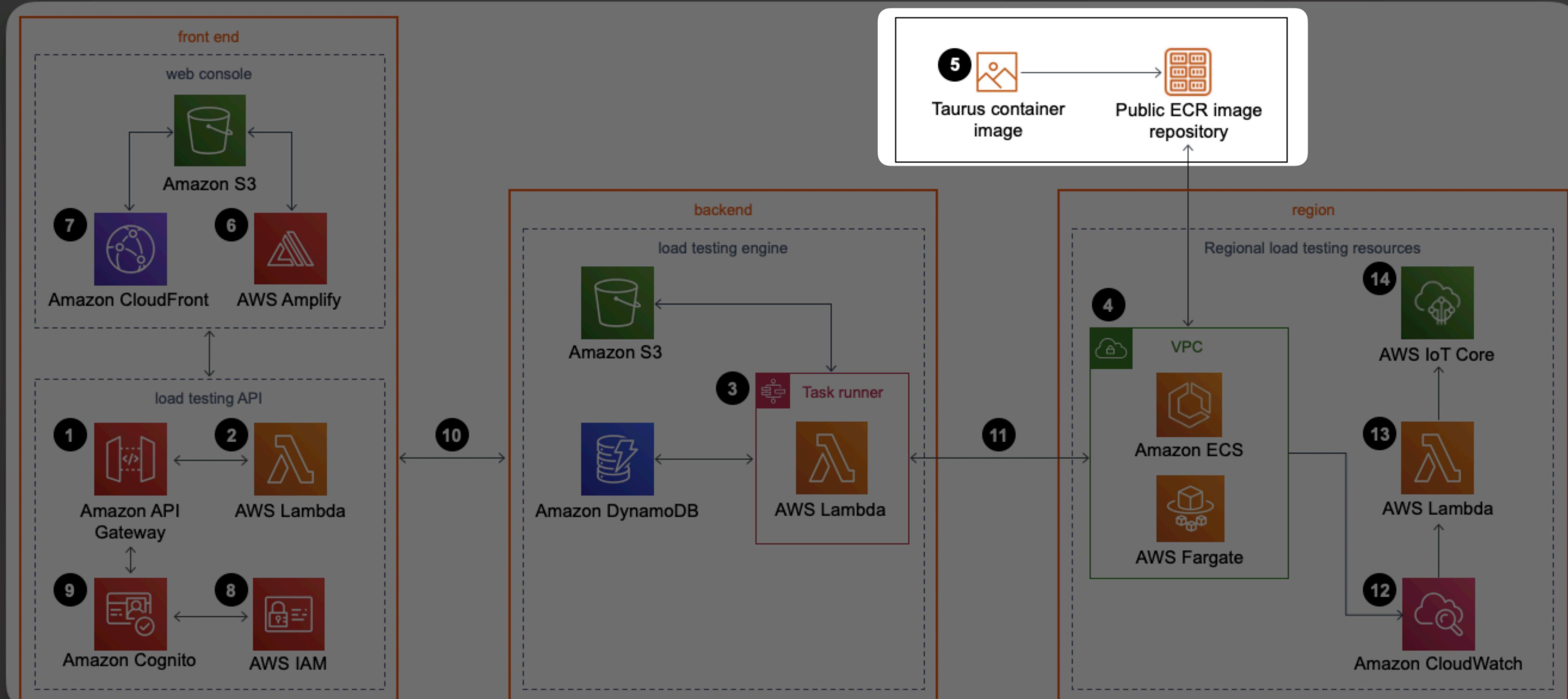
Graph view

Actions ▾

In progress Failed Caught error Canceled Succeeded







google.com 메모리 사용량: 348MB odename Taurus Automation-friendly framework for Continuous Testing

pypi v1.16.27 codecov 90% github stars 1.9k

Installation Learning Taurus User Manual Keyword Index Support Forum Changelog Search site...

You run tests more than once

It makes sense to automate anything you repeat 10+ times.
Taurus improves experience of JMeter, Selenium and others.

The screenshot displays the Taurus dashboard with several key sections:

- Top Left:** A histogram titled "Taurus" showing response times. Below it, a bar chart indicates 47 hits and 32 fails.
- Top Right:** Performance metrics and response codes. Average Times: Elapsed: 0.111, Connect: 0.000, Latency: 0.000. Percentiles: 0.0%: 0.050, 50.0%: 0.132, 90.0%: 0.147, 95.0%: 0.151, 99.0%: 0.158, 99.9%: 0.158, 100.0%: 0.158. Response Codes: 200: 31.91% (15), N/A: 68.09% (32), All: 100.00% (47).
- Bottom Left:** A histogram showing cumulative statistics for 00:01:01. Average Times: Elapsed: 0.445, Connect: 0.041, Latency: 0.055. Percentiles: 0.0%: 0.044, 50.0%: 0.134, 90.0%: 0.604, 95.0%: 1.361, 99.0%: 5.792, 99.9%: 6.988, 100.0%: 7.024. Response Codes: 200: 38.66% (893), 404: 3.90% (90), N/A: 57.45% (1327), All: 100.00% (2310).
- Bottom Right:** Test logs and system metrics. Logs show analysis steps like "Waiting for finish...", "Changed data analysis delay to 3s", etc. System metrics include local memory usage (mem: 82.300), disk operations (disk-write: 157,672, disk-read: 258,743), bytes transferred (bytes-recv: 10,688,077, bytes-sent: 520,708), disk space (disk-space: 27.900), engine loop time (engine-loop: 0.074), and CPU usage (cpu: 24.500).

Labels:

Label	Hits	Failures	Avg Time
http://blazemeter.com	773	0.00%	0.138
https://blazemeter.com	110	95.45%	5.380
https://blazemeter.com Redirect 1	667	99.25%	0.167
https://blazemeter.com/not_found	95	94.74%	1.230
https://blazemeter.com/not_found Redirect	665	100.00%	0.153

Errors:

- 1322 of: Invalid redirect to the same request
- 105 of: The operation lasted too long
- 90 of: Not Found
- 5 of: status.find.in(200,201,202,203,204,205,206,207,208,209,304), fo

다음에서 보기:



About

- [Overview](#)
- [License](#)

[Index](#)[Next >](#)

Download

- [Download Releases](#)
- [Release Notes](#)

Documentation

- [Get Started](#)
- [User Manual](#)
- [Best Practices](#)
- [Component Reference](#)
- [Functions Reference](#)
- [Properties Reference](#)
- [Change History](#)
- [Javadocs](#)
- [JMeter Wiki](#)
- [FAQ \(Wiki\)](#)

Tutorials

- [Distributed Testing](#)
- [Recording Tests](#)
- [JUnit Sampler](#)
- [Access Log Sampler](#)
- [Extending JMeter](#)

Community

- [Issue Tracking](#)
- [Security](#)
- [Mailing Lists](#)
- [Source Repositories](#)
- [Building and Contributing](#)
- [Project info at Apache](#)
- [Contributors](#)

Foundation

- [The Apache Software Foundation \(ASF\)](#)
- [Get Involved in the ASF](#)
- [Privacy Policy](#)
- [Sponsorship](#)
- [Thanks](#)

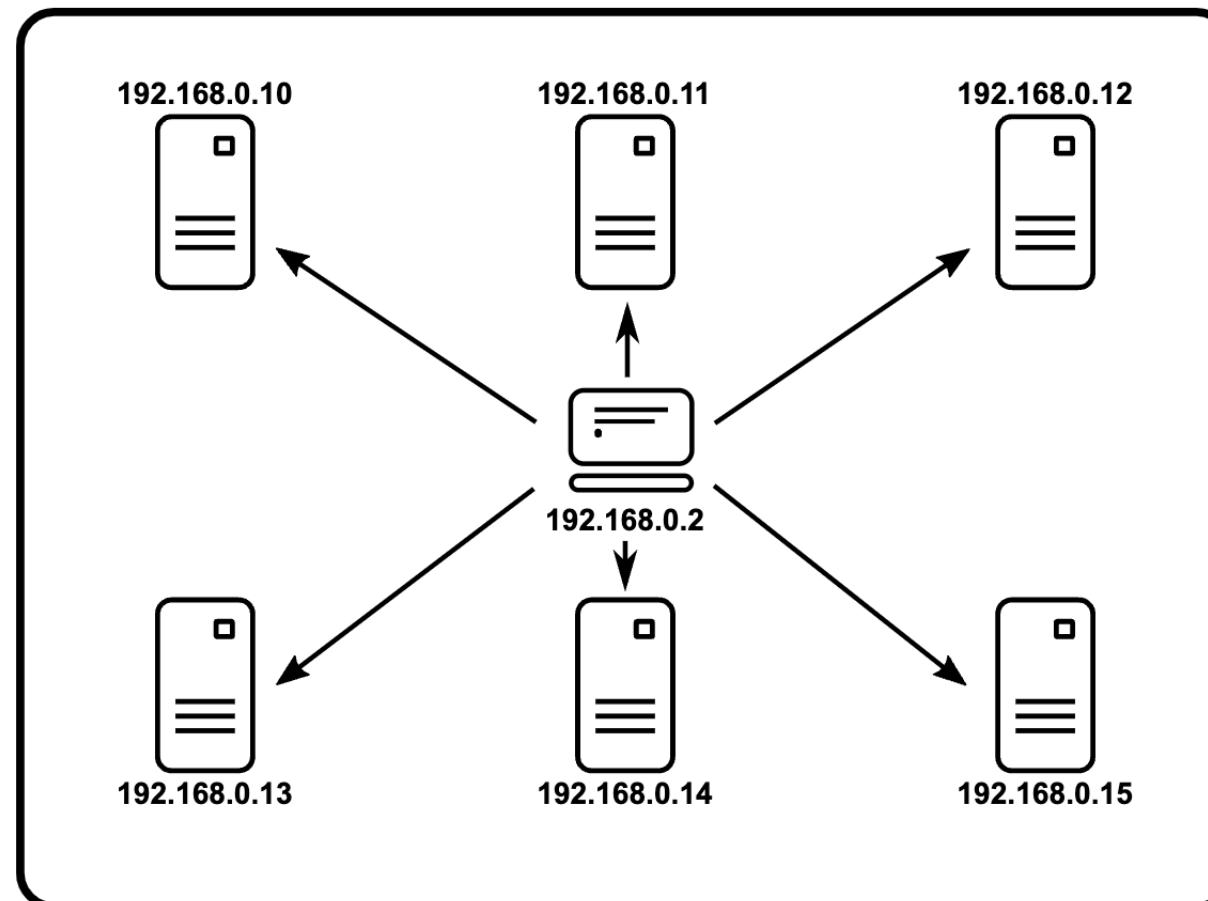
25. Apache JMeter Distributed Testing Step-by-step

This short tutorial explains how to use multiple systems to perform stress testing. Before we start, there are a couple of things to check.

- the firewalls on the systems are turned off or correct ports are opened.
- all the clients are on the same subnet.
- the server is in the same subnet, if **192.x.x.x** or **10.x.x.x** IP addresses are used. If the server doesn't use **192.xx** or **10.xx** IP address, there shouldn't be any problems.
- Make sure JMeter can access the server.
- Make sure you use the same version of JMeter and Java on all the systems. Mixing versions will not work correctly.
- You have [setup SSL for RMI](#) or disabled it.

Once you've made sure the systems are ready, it's time to setup remote testing. The tutorial assumes you already have JMeter installed on all the systems. The way JMeter works is one controller node initiates the test on multiple worker nodes.

In this tutorial we use GUI Mode just for demonstration. In real life you should use CLI mode (NON GUI) to start your load test



One controller node with multiple worker nodes



About
• Overview
• License

[Index](#)

[Next >](#)

[Twitter](#) [github](#)

Download
• [Download Releases](#)
• [Release Notes](#)

Documentation
• [Get Started](#)
• [User Manual](#)
• [Best Practices](#)
• [Component Reference](#)
• [Functions Reference](#)
• [Properties Reference](#)
• [Change History](#)
• [Javadocs](#)
• [JMeter Wiki](#)
• [FAQ \(Wiki\)](#)

Tutorials
• [Distributed Testing](#)
• [Recording Tests](#)
• [JUnit Sampler](#)
• [Access Log Sampler](#)
• [Extending JMeter](#)

Community
• [Issue Tracking](#)
• [Security](#)
• [Mailing Lists](#)
• [Source Repositories](#)
• [Building and Contributing](#)
• [Project info at Apache](#)
• [Contributors](#)

Foundation
• [The Apache Software Foundation \(ASF\)](#)
• [Get Involved in the ASF](#)
• [Privacy Policy](#)
• [Sponsorship](#)
• [Thanks](#)

25. Apache JMeter Distributed Testing Step-by-step

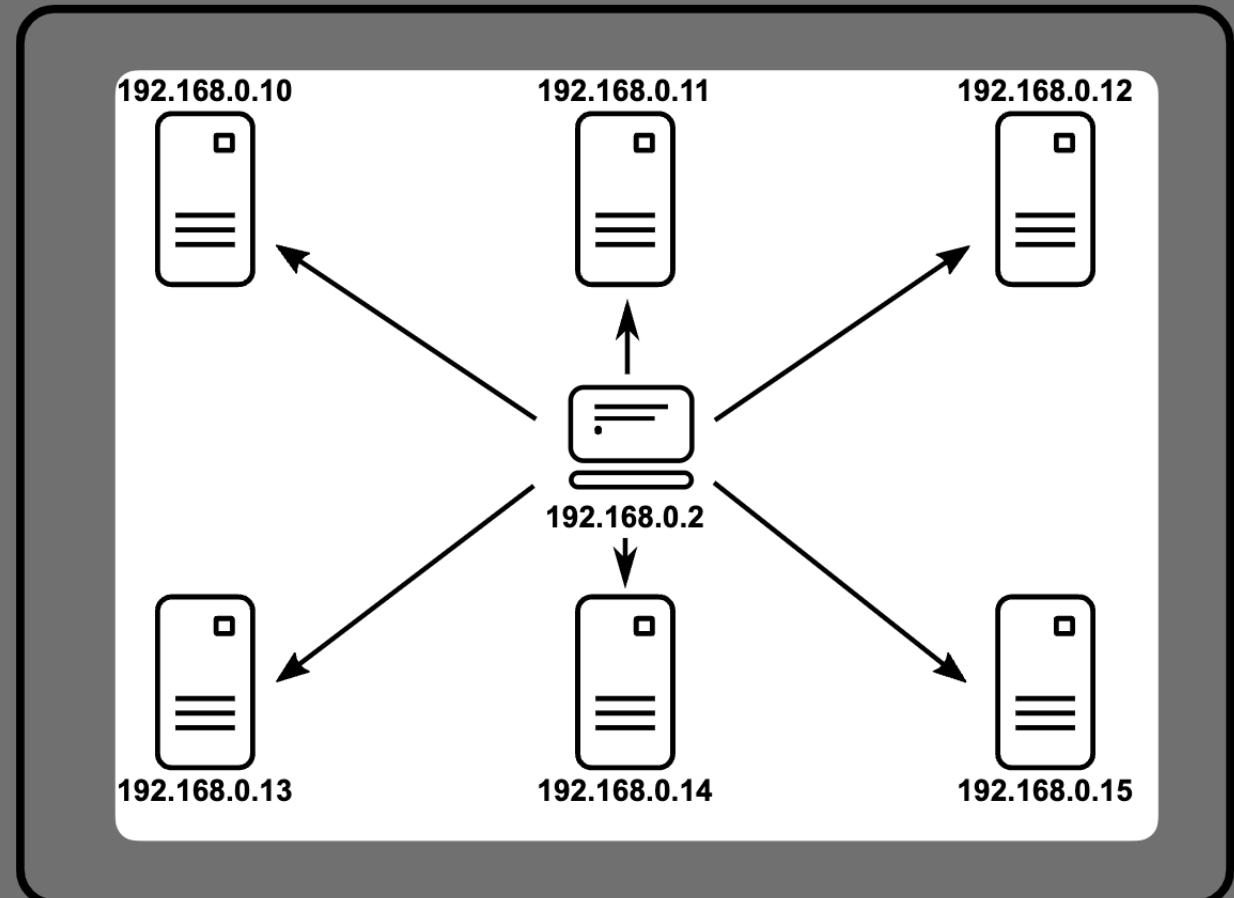
This short tutorial explains how to use multiple systems to perform stress testing. Before we start, there are a couple of things to check.

- the firewalls on the systems are turned off or correct ports are opened.
- all the clients are on the same subnet.
- the server is in the same subnet, if **192.x.x.x** or **10.x.x.x** IP addresses are used. If the server doesn't use **192.xx** or **10.xx** IP address, there shouldn't be any problems.
- Make sure JMeter can access the server.
- Make sure you use the same version of JMeter and Java on all the systems. Mixing versions will not work correctly.
- You have [setup SSL for RMI](#) or disabled it.

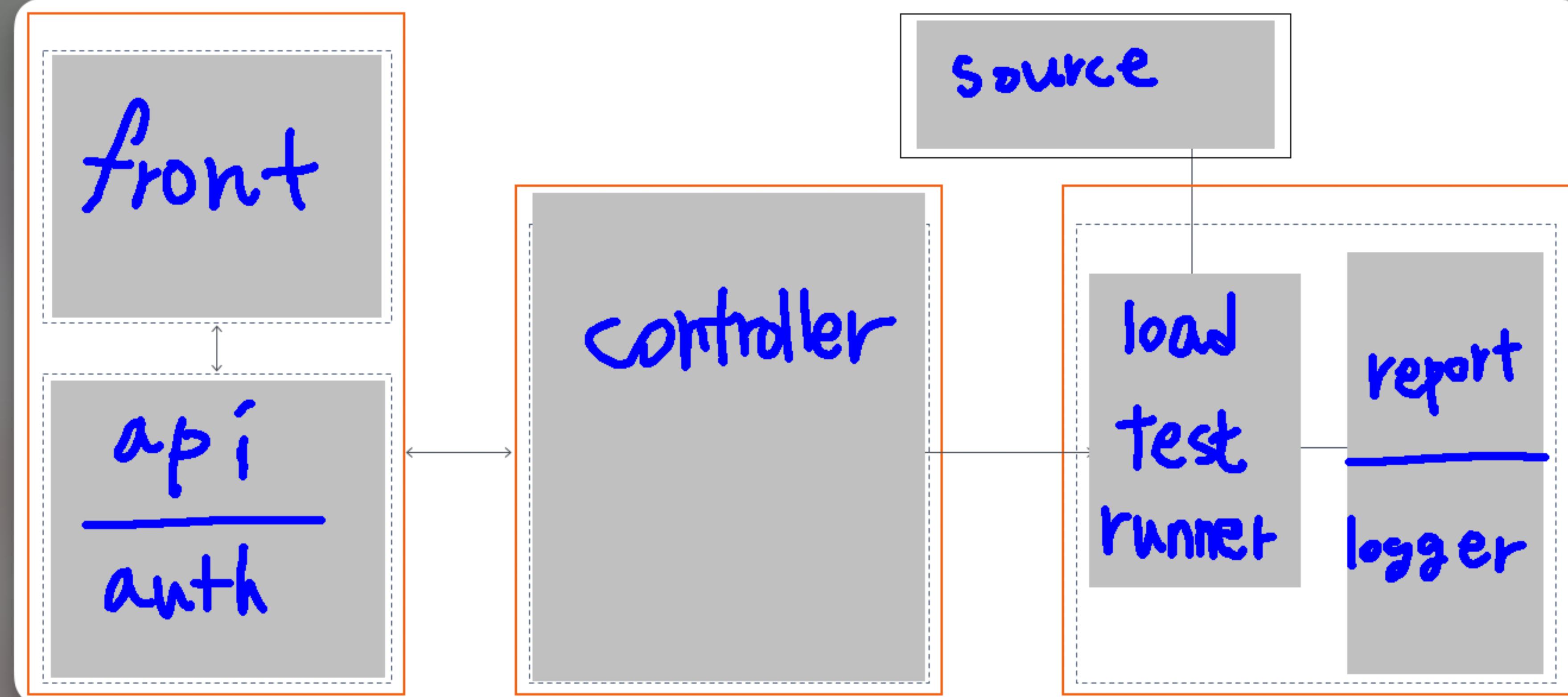
Once you've made sure the systems are ready, it's time to setup remote testing. The tutorial assumes you already have JMeter installed on all the systems. The way JMeter works is one controller node initiates the test on multiple worker nodes.

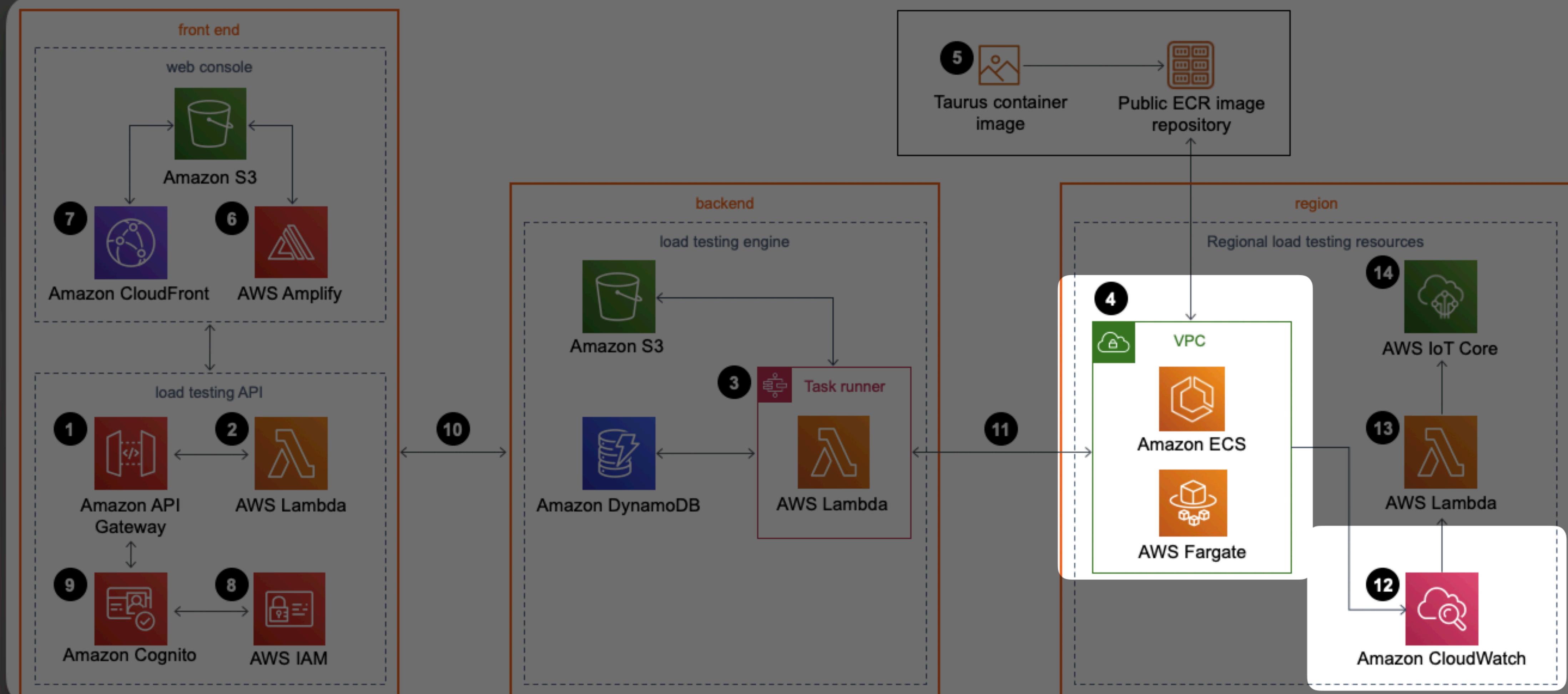
In this tutorial we use GUI Mode just for demonstration. In real life you should use CLI mode (NON GUI) to start your load test

taurus: 😊



One controller node with multiple worker nodes





warn Services Search [Option+S]

Tell us what you think X

Amazon Elastic Container Service > Task definitions > roeniss-dlt-test-DLTEcsDLTTaskDefinition6BFC2400-19v52JP14plb > Revision 1 > JSON

roeniss-dlt-test-DLTEcsDLTTaskDefinition6BFC2400-19v52JP14plb:1

Overview Info

ARN	Status	Time created
[REDACTED]	ACTIVE	2024-02-25T13:32:17.003Z
Task role	Task execution role	Operating system/Architecture
roeniss-dlt-test-DLTEcsDLTTaskExecutionRoleDE668717-pZ4Al0GuPUTV	roeniss-dlt-test-DLTEcsDLTTaskExecutionRoleDE668717-pZ4Al0GuPUTV	-

Containers JSON Task placement Volumes (0) Requires attributes Tags

JSON

```
1 {  
2     "taskDefinitionArn": "task-definition/roeniss-dlt-test-DLTEcsDLTTaskDefinition6BFC2400-19v52JP14plb:1",  
3     "containerDefinitions": [  
4         {  
5             "name": "roeniss-dlt-test-load-tester",  
6             "image": "public.ecr.aws/aws-solutions/distributed-load-testing-on-aws-load-tester:v3.2.5",  
7             "cpu": 0,  
8             "memory": 4096,  
9             "links": [],  
10            "portMappings": [],  
11            "essential": true,  
12            "entryPoint": [],  
13            "command": [],  
14            "environment": [],  
15            "environmentFiles": [],  
16            "mountPoints": [],  
17            "volumesFrom": [],  
18            "secrets": [],  
19            "dnsServers": [],  
20            "dnsSearchDomains": [],  
21            "extraHosts": [],  
22            "dockerSecurityOptions": [],  
23            "dockerLabels": {},  
24            "ulimits": [],  
25            "logConfiguration": {  
26                "logDriver": "awslogs",  
27                "options": {  
28                    "awslogs-group": "roeniss-dlt-test-DLTEcsDLTCloudWatchLogsGroupFE9EC144-10zKgM6cvhTV",  
29                    "awslogs-region": "us-east-1",  
30                    "awslogs-stream-prefix": "load-testing"  
31                },  
32            },  
33        }  
34    ]  
35}
```

AWS Services Search [Option+S]

AWS Services Search [Option+S]

Tell us what you think X

Amazon Elastic Container Service > Task definitions > roeniss-dlt-test-DLTEcsDLTTaskDefinition6BFC2400-19v52JP14plb > Revision 1 > JSON

roeniss-dlt-test-DLTEcsDLTTaskDefinition6BFC2400-19v52JP14plb:1

Overview Info

ARN: [REDACTED] Status: ACTIVE

Task role: roeniss-dlt-test-DLTEcsDLTTaskExecutionRoleDE668717-pZ4Al0GuPUTV []

Task execution role: roeniss-dlt-test-DLTEcsDLTTaskExecutionRoleDE668717-pZ4Al0GuPUTV []

Containers JSON Task placement Volumes (0) Requires attributes Tags

JSON

```
1 {  
2   "taskDefinitionArn": "arn:aws:ecs:us-east-1:145607721954:task-definition/roeniss-dlt-test-DLTEcsDLTTaskDefinition6BFC2400-19v52JP14plb:1",  
3   "containerDefinitions": [  
4     {  
5       "name": "roeniss-dlt-test-load-tester",  
6       "image": "public.ecr.aws/aws-solutions/distributed-load-testing-on-aws-load-tester:v3.2.5",  
7       "cpu": 0,  
8       "memory": 4096,  
9       "links": [],  
10      "portMappings": [],  
11      "essential": true,  
12      "entryPoint": [],  
13      "command": [],  
14      "environment": [],  
15      "environmentFiles": [],  
16      "mountPoints": [],  
17      "volumesFrom": [],  
18      "secrets": [],  
19      "dnsServers": [],  
20      "dnsSearchDomains": [],  
21      "extraHosts": [],  
22      "dockerSecurityOptions": [],  
23      "dockerLabels": {},  
24      "ulimits": [],  
25      "logConfiguration": {  
26        "logDriver": "awslogs",  
27        "options": {  
28          "awslogs-group": "roeniss-dlt-test-DLTEcsDLTCloudWatchLogsGroupFE9EC144-10zKgM6cvhTV",  
29          "awslogs-region": "us-east-1",  
30          "awslogs-stream-prefix": "load-testing"  
31        },  
32      },  
33    },  
34  ],  
35  "requiresAttributes": [  
36    {  
37      "name": "com.amazonaws.ecs.capability.logging-driver",  
38    },  
39    {  
40      "name": "ecs.capability.execution-role-awslogs",  
41    },  
42    {  
43      "name": "com.amazonaws.ecs.capability.docker-remote-api",  
44    },  
45    {  
46      "name": "com.amazonaws.ecs.capability.docker-remote-auth",  
47    },  
48    {  
49      "name": "com.amazonaws.ecs.capability.task-iam-role",  
50    },  
51    {  
52      "name": "com.amazonaws.ecs.capability.docker-remot",  
53    },  
54    {  
55      "name": "com.amazonaws.ecs.capability.docker-remot",  
56    },  
57    {  
58      "name": "com.amazonaws.ecs.capability.task-iam-rol",  
59    },  
60    {  
61      "name": "com.amazonaws.ecs.capability.docker-remot",  
62    },  
63    {  
64      "name": "com.amazonaws.ecs.capability.docker-remot",  
65    },  
66    {  
67      "name": "ecs.capability.task-eni",  
68    },  
69    {  
70      "placementConstraints": [],  
71      "compatibilities": [  
72        "EC2",  
73        "FARGATE"  
74      ],  
75      "requiresCompatibilities": [  
76        "FARGATE"  
77      ],  
78      "cpu": "2048",  
79      "memory": "4096",  
80      "registeredAt": "2024-02-25T13:32:17.003Z",  
81      "registeredBy": "arn:aws:iam::145607721954:user/sh.moon",  
82      "tags": [  
83        {  
84          "key": "SolutionId",  
85          "value": "S00062"  
86        },  
87        {  
88          "key": "purpose",  
89          "value": "test"  
90        }  
91      }  
92    }  
93  ]  
94 }
```

Tell us what you think X

Amazon Elastic Container Service

Clusters
Namespaces
Task definitions
Account settings

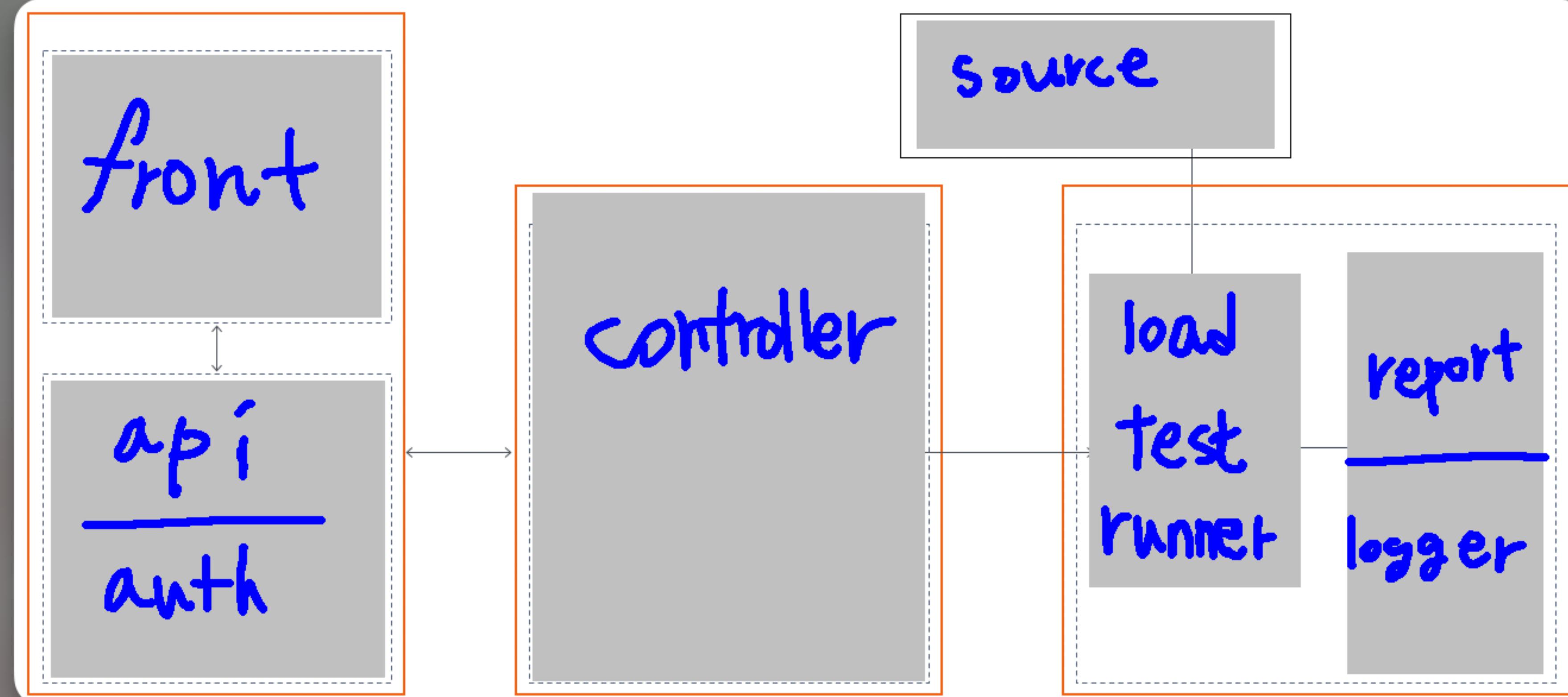
Install AWS Copilot

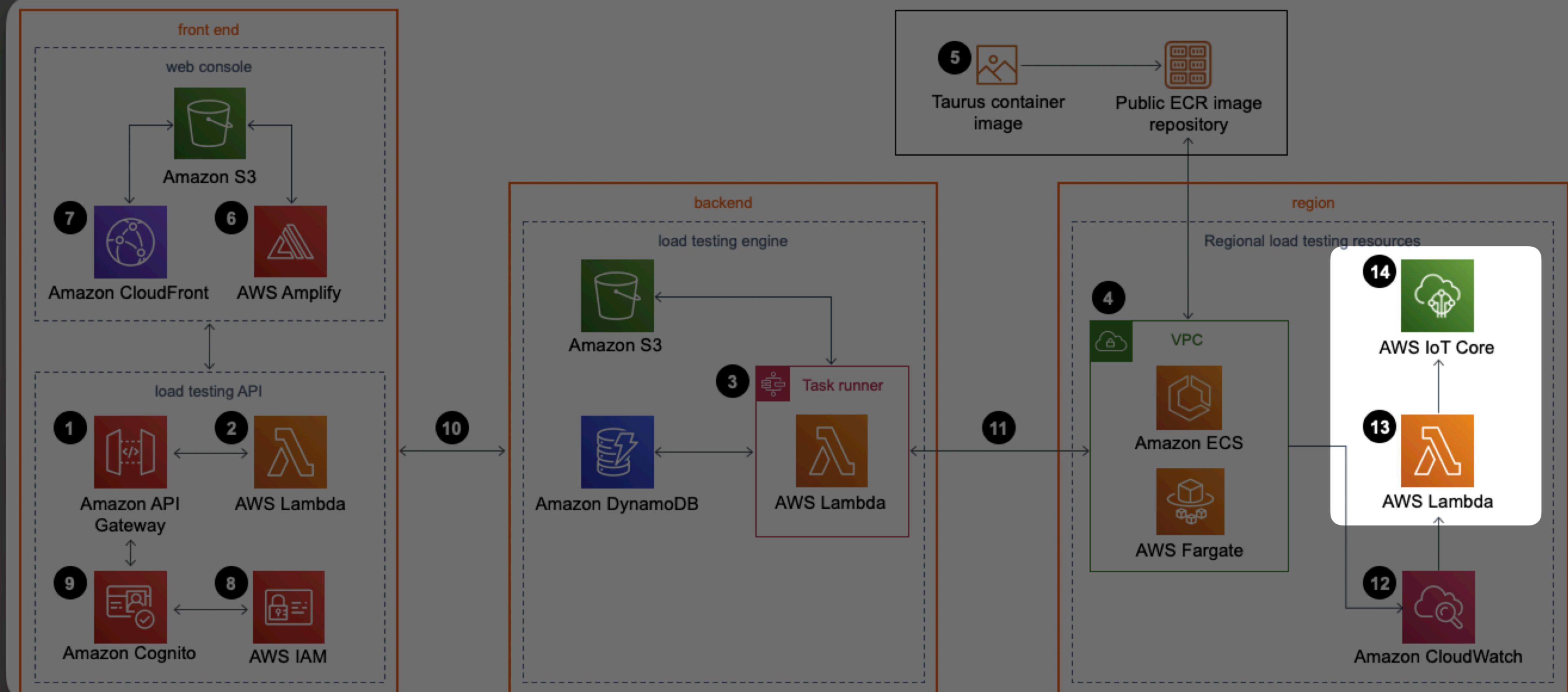
Amazon ECR
Repositories

AWS Batch

Documentation
Discover products
Subscriptions

```
35     "credentialSpecs": []
36   }
37 ],
38 "family": "roeniss-dlt-test-DLTEcsDLTTaskDefinition6BFC2400-19v52JP14plb",
39 "taskRoleArn": "arn:aws:iam::145607721954:role/roeniss-dlt-test-DLTEcsDLTTaskExecutionRoleDE668717-pZ4Al0GuPUTV",
40 "executionRoleArn": "arn:aws:iam::145607721954:role/roeniss-dlt-test-DLTEcsDLTTaskExecutionRoleDE668717-pZ4Al0GuPUTV",
41 "networkMode": "awsvpc",
42 "revision": 1,
43 "volumes": [],
44 "status": "ACTIVE",
45 "requiresAttributes": [
46   {
47     "name": "com.amazonaws.ecs.capability.logging-driver.awslogs"
48   },
49   {
50     "name": "ecs.capability.execution-role-awslogs"
51   },
52   {
53     "name": "com.amazonaws.ecs.capability.docker-remote-api.1.19"
54   },
55   {
56     "name": "com.amazonaws.ecs.capability.docker-remote-api.1.17"
57   },
58   {
59     "name": "com.amazonaws.ecs.capability.task-iam-role"
60   },
61   {
62     "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
63   },
64   {
65     "name": "ecs.capability.task-eni"
66   }
67 ],
68 "placementConstraints": [],
69 "compatibilities": [
70   "EC2",
71   "FARGATE"
72 ],
73 "requiresCompatibilities": [
74   "FARGATE"
75 ],
76 "cpu": "2048",
77 "memory": "4096",
78 "registeredAt": "2024-02-25T13:32:17.003Z",
79 "registeredBy": "arn:aws:iam::145607721954:user/sh.moon",
80 "tags": [
81   {
82     "key": "SolutionId",
83     "value": "S00062"
84   },
85   {
86     "key": "purpose",
87     "value": "test"
88   }
89 ]
90 }
```





AWS IoT

X

Monitor

Connect

- Connect one device
- ▶ Connect many devices

Test

- ▶ Device Advisor
- MQTT test client
- Device Location New

Manage

- ▶ All devices
- ▶ Greengrass devices
- ▶ LPWAN devices
- Software packages New
- ▶ Remote actions
- ▶ Message routing
- Retained messages
- ▶ Security
- ▶ Fleet Hub

Device software

Billing groups

Settings

Feature spotlight

Documentation ↗

[Tell us what you think](#)

AWS IoT > Monitor

Monitor Info

IoT metrics Job execution metrics

1h 3h 12h 1d 3d 1w Custom (6h) UTC timezone ▾ G ▾ Add to dashboard :

Successful connections

No unit

27

13.5

0

16:00 17:00 18:00 19:00 20:00 21:00 22:00

■ Connect.Success

Messages published

No unit

2.64k

1.32k

0

16:00 17:00 18:00 19:00 20:00 21:00 22:00

■ Inbound ■ Outbound

Rules executed

No unit

1

No data available.
Try adjusting the dashboard time range.

0.5

0

16:00 17:00 18:00 19:00 20:00 21:00 22:00

■ RulesExecuted

Protocol rate(%)

MQTT HTTP

Metric type rate(%)

Publish Connect Ping Subscribe

Publish metric message rate(%)

Inbound Outbound

Get thing shadow success rate(%)

Success rate (%) Failure rate (%)

Update thing shadow success rate(%)

Success rate (%) Failure rate (%)

Delete thing shadow success rate(%)

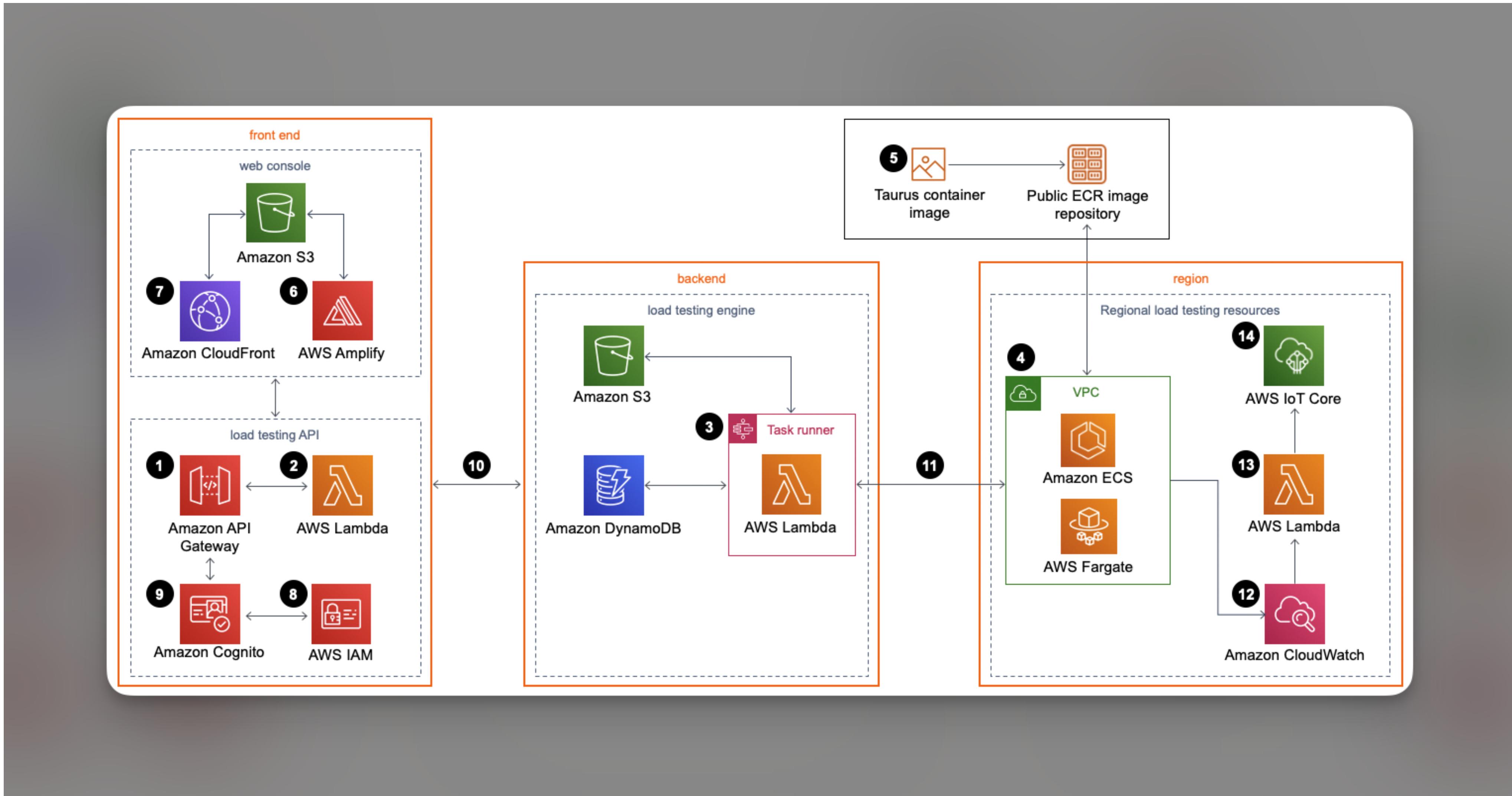
Success rate (%) Failure rate (%)

github.com/aws-solutions/distributed-load-testing-on-aws/blob/78caed5a07cd7920a70c7db5626c785fba63814a/source/real-time-data-publisher/index.js#L106

distributed-load-testing-on-aws / source / real-time-data-publisher / index.js

Code Blame 111 lines (97 loc) · 3.75 KB

```
34     exports.handler = async function (event) {  
35         extractedData[keys[index]] = value;  
36     }  
37     //get testId if not already received  
38     testId = testId || extractedData.testId;  
39     //add timestamp and push individual line data to aggregate  
40     extractedData.timestamp = Math.round(logItem.timestamp / 1000);  
41     aggregatedTestResultData[process.env.AWS_REGION].push(extractedData);  
42 }  
43 } catch (error) {  
44     console.error("Error decompressing payload: ", error);  
45     throw error;  
46 }  
47 //publish to testId topic using endpoint in main region  
48 const params = {  
49     topic: `dlt/${testId}`,  
50     payload: JSON.stringify(aggregatedTestResultData),  
51 };  
52 try {  
53     await iot.publish(params).promise();  
54     console.log(`Successfully sent data to topic dlt/${testId}`);  
55 } catch (error) {  
56     console.error("Error publishing to IoT Topic: ", error);  
57     throw error;  
58 }  
59 };  
60 };  
61 };  
62 };  
63 };  
64 };  
65 };  
66 };  
67 };  
68 };  
69 };  
70 };  
71 };  
72 };  
73 };  
74 };  
75 };  
76 };  
77 };  
78 };  
79 };  
80 };  
81 };  
82 };  
83 };  
84 };  
85 };  
86 };  
87 };  
88 };  
89 };  
90 };  
91 };  
92 };  
93 };  
94 };  
95 };  
96 };  
97 };  
98 };  
99 };  
100 };  
101 };  
102 };  
103 };  
104 };  
105 };  
106 };  
107 };  
108 };  
109 };  
110 };  
111 };
```



Welcome to Distributed Load Testing

Inbox x



no-reply@verificationemail.com

10:31PM (35 minutes ago)



to me ▾

Please use the credentials below to login to the Distributed Load Testing console.

Username:  

Password:    

Console: <https://dwx2vere6xv6e.cloudfront.net/>

 Reply

 Forward



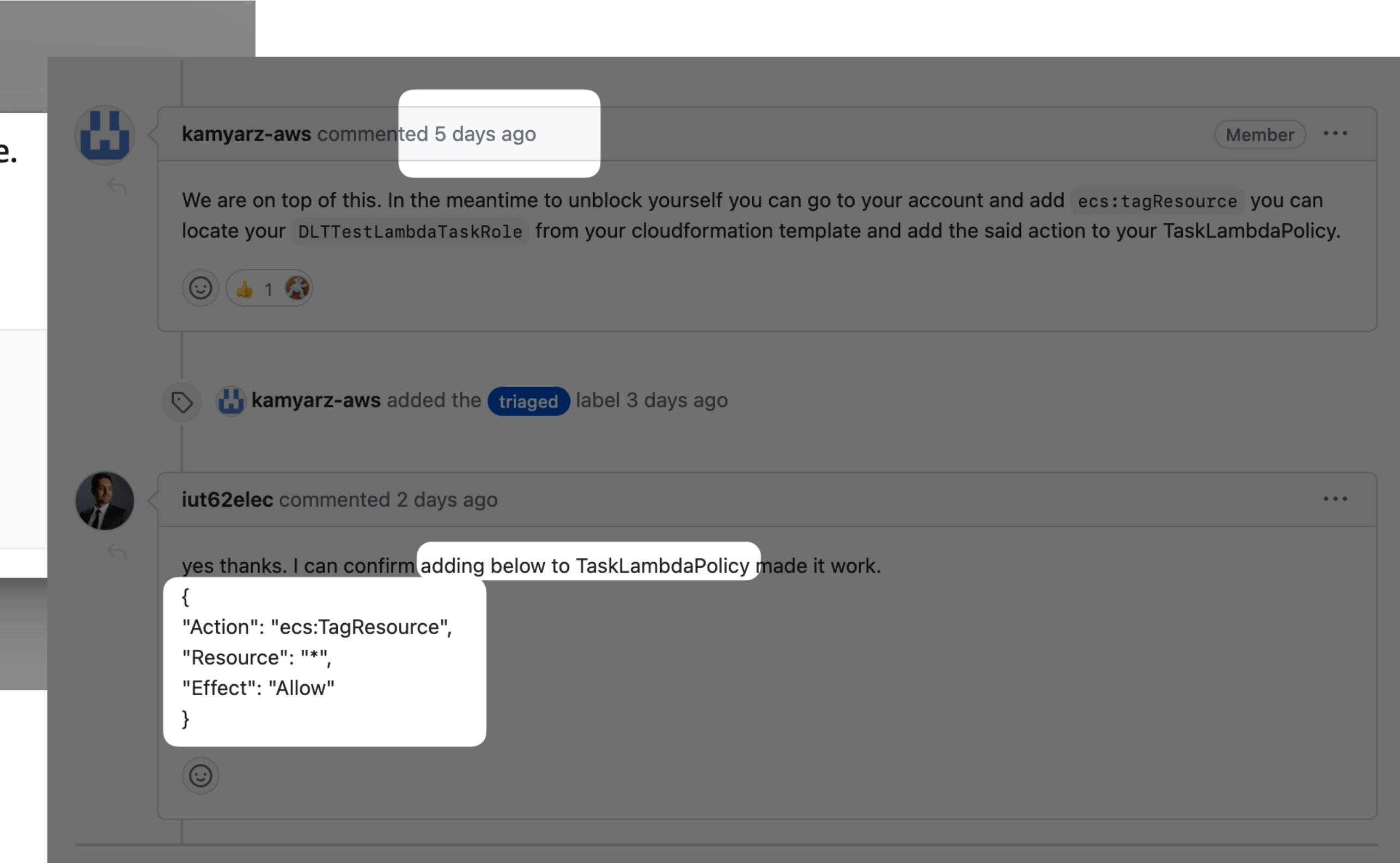
사소하..지 않은 주의사항들

1. Download the [AWS CloudFormation template](#) to your local hard drive.
2. Open the AWS CloudFormation template with a text editor.
3. Modify the AWS CloudFormation template mapping section from:

Solution:

Config:

SendAnonymousData: "Yes"



<https://github.com/aws-solutions/distributed-load-testing-on-aws/issues/162>

자투리1: 타겟 서버는 어디?

```
1 package main
2
3 import (
4     "fmt"
5     "log"
6     "net/http"
7 )
8
9 var i int64 3 usages ↗ 문성혁
10
11 func serve(w http.ResponseWriter, r *http.Request) { 1 usage ↗ 문성혁
12     i++
13     _, _ = fmt.Fprintf(w, format: "Hi there! This is %vth visit.", i)
14 }
15
16 ▶ func main() { ↗ 문성혁 +1
17     i = 0
18     http.HandleFunc("/", serve)
19     err := http.ListenAndServe(":80", nil)
20     if err != nil {
21         log.Fatalln(err)
22     }
23 }
24
```

AWS Services Search [Option+S]

App Runner > Create service

Step 1 Source and deployment

Step 2 Configure service

Step 3 Review and create

Source and deployment Info

Choose the source for your App Runner service and the way it's deployed.

Source and deployment

Source

Repository type

Container registry
Deploy your service using a container image stored in a container registry.

Source code repository
Deploy your service using the code hosted in a source repository.

Provider

Amazon ECR
Container images stored in the private ECR.

Amazon ECR Public
Container images publicly shared by vendors, open source projects, and community developers.

Container image URI

Enter a URI to an image you can access, or browse images in your Amazon ECR account.

111111111111.dkr.ecr.us-east-1.amazonaws.com/myfirstrepo:latest

Browse

Deployment settings

Deployment trigger

Manual
Start each deployment yourself using the App Runner console or AWS CLI.

Automatic
App Runner monitors your registry and deploys a new version of your service for each image push.

ECR access role Info

This role gives App Runner permission to access ECR. To create a custom role, go to the [IAM console](#)

Create new service role

Use existing service role

Service role name

The name of an IAM role that App Runner creates in your account with an attached managed policy for ECR access.

AppRunnerECRAccessRole

Cancel Next

AWS Services Search [Option+S]

App Runner > Create service

Step 1 Source and deployment

Step 2 Configure service

Step 3 Review and create

Configure service Info

Configure service

Service settings

Service name

Virtual CPU Virtual memory

1 vCPU ▾ 2 GB ▾

Runtime environment variables - *optional* Info

Add environment variables in plain text or reference them from [Secrets Manager](#) and [SSM Parameter Store](#). Update IAM Policies using the IAM Policy template given below to securely reference secrets and configurations as environment variables.

No environment variables have been configured.

Add environment variable

You can add up to 50 more items.

► IAM policy templates for secrets

Port
Your service uses this TCP port.

► Additional configuration

aws | Services Search [Option+S]

using the IAM Policy template given below to securely reference secrets and configurations as environment variables.
No environment variables have been configured.

Add environment variable
You can add up to 50 more items.

► IAM policy templates for secrets

Port
Your service uses this TCP port.
8080

► Additional configuration

▼ Auto scaling [Info](#)
Configure automatic scaling behavior.

Auto scaling configurations Create ▾

Existing configurations
DefaultConfiguration ▾ v1 ▾ G

Concurrency
100 requests

Minimum size
1 instance(s)

Maximum size
25 instances

This screenshot shows the AWS Lambda function configuration interface. The main panel displays settings for IAM policy templates and environment variables. A modal window is open over the configuration, specifically the 'Auto scaling' section. The modal shows existing configurations ('DefaultConfiguration' and 'v1'), concurrency settings (100 requests), and minimum/maximum instance sizes (1 and 25 instances respectively). The 'Create' button is visible in the top right corner of the modal.

aws | Services Search [Option+S]

Concurrency
100 requests

Minimum size
1 instance(s)

Maximum size
25 instances

▼ Health check Info
Configure load balancer health checks.

Protocol
The IP protocol that App Runner uses to perform health checks for your service.
▼

Path
The URL that App Runner sends HTTP health check requests to. Applicable only to HTTP checks.

Custom path example: /health or /health-check

Timeout
Amount of time the load balancer waits for a health check response.
 seconds

Interval
Amount of time between health checks of an individual instance.
 seconds

Unhealthy threshold
The number of consecutive health check failures that determine an instance is unhealthy.
 requests

Health threshold
The number of consecutive successful health checks that determine an instance is healthy.
 requests

ECS 대역폭 측정 벤치마크 (비공식)

<https://github.com/sjakthol/aws-network-benchmark>

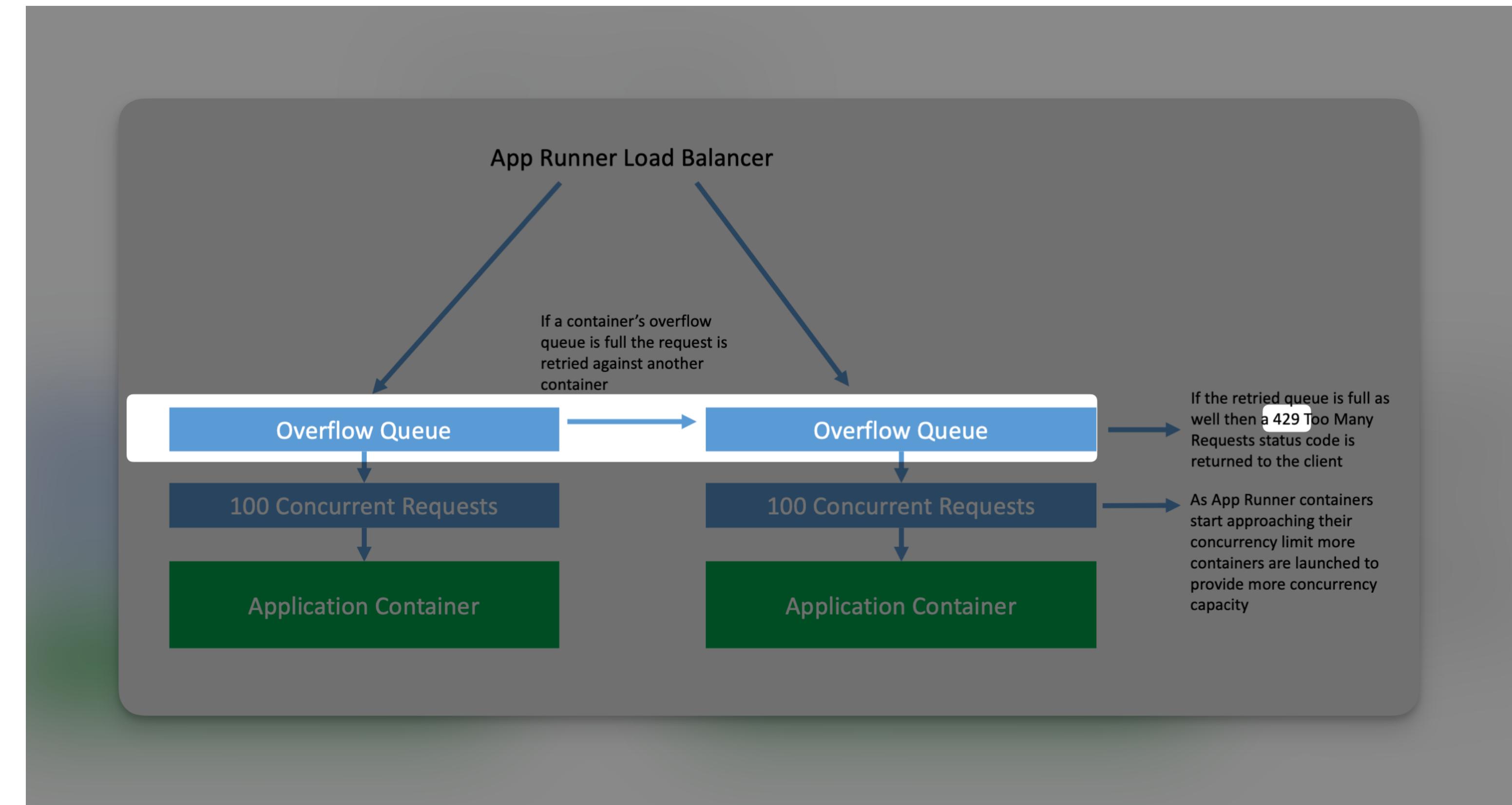
Results (Fargate)

Here are some results from benchmarks performed on eu-west-1 and eu-north-1 regions on 2020-05-17. See [analysis/results.ipynb](#) for more detailed graphs.

eu-west-1

vCPU	Memory (MB)	Baseline (Gbps)	Burst (Gbps)	Burst Duration (seconds)
0.25	512	0.254	0.850	80
0.25	1024	0.254	0.849	80
0.25	2048	0.254	0.841	80
0.5	1024	0.620	0.620	-
0.5	2048	0.620	0.620	-
0.5	4096	1.240	1.241	-
1	2048	0.620	0.620	-
1	4096	1.240	1.240	-
1	8192	0.745	10.093	270
2	4096	1.240	1.240	-
2	8192	0.745	10.093	290
2	16384	0.694	0.695	-
4	8192	0.744	0.744	-
4	16384	0.694	0.695	-
4	30720	0.694	0.695	-

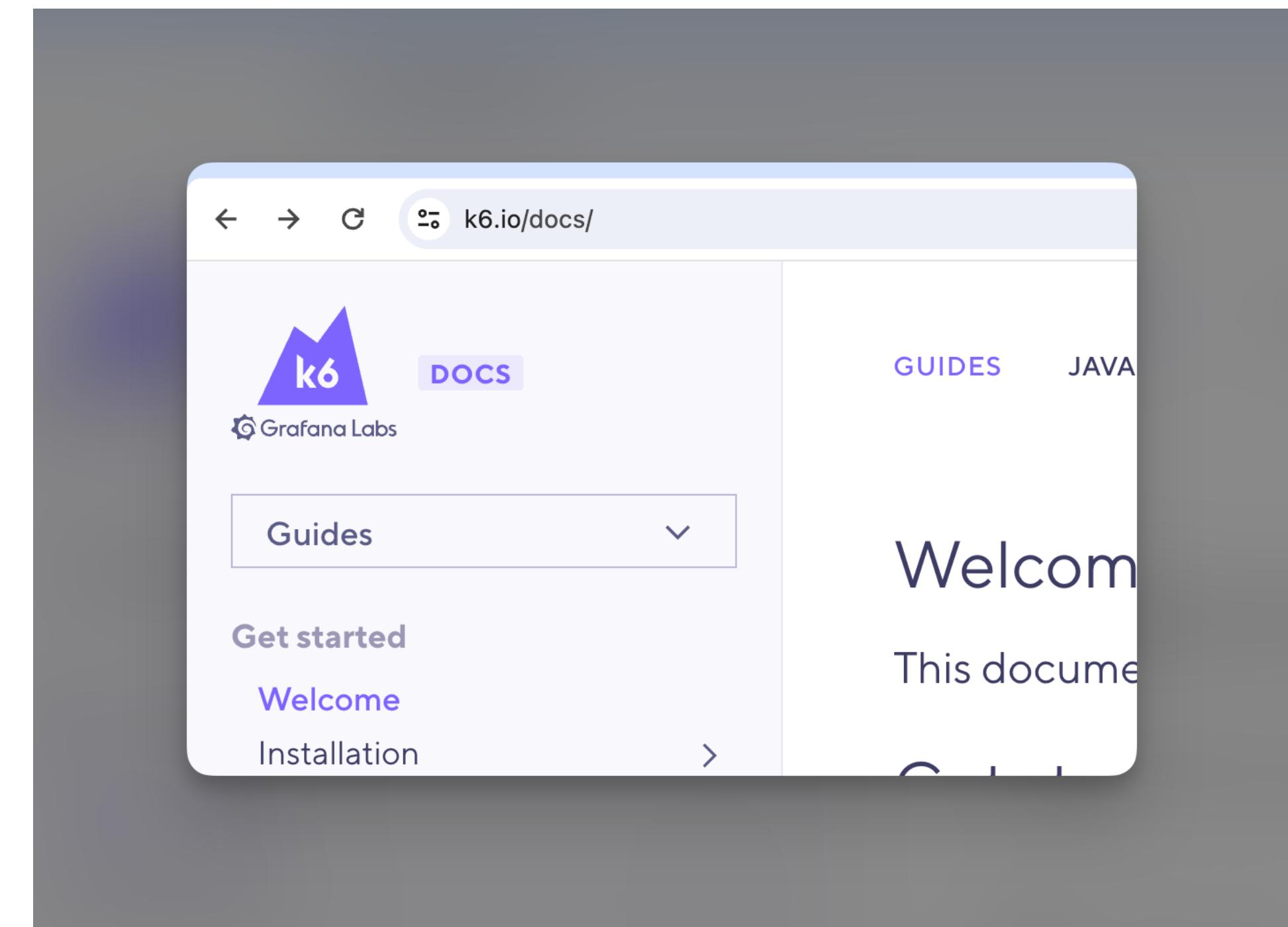
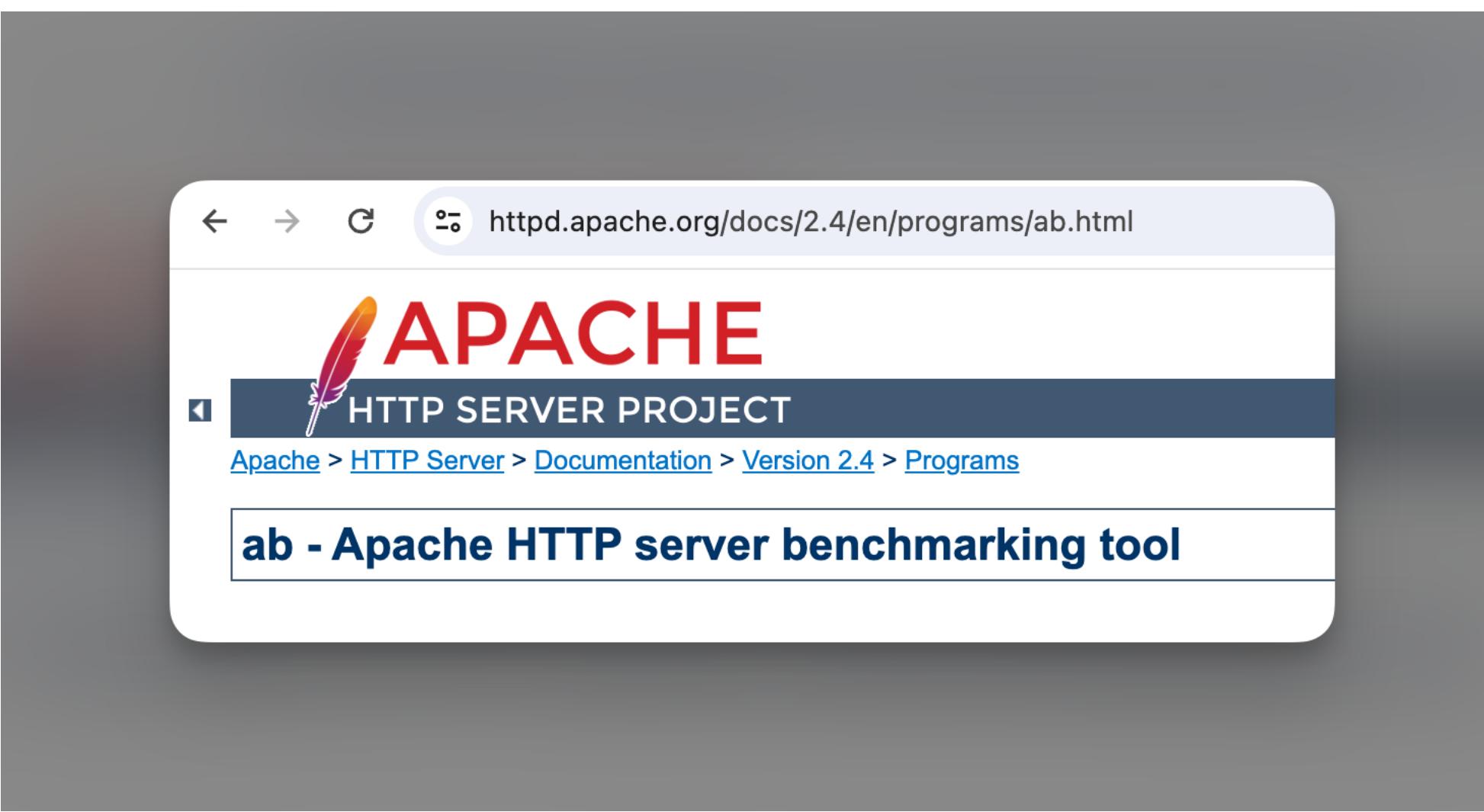
429 는 어떻게 나오는걸까?



자투리2: 부하 테스트는 뭐부터?

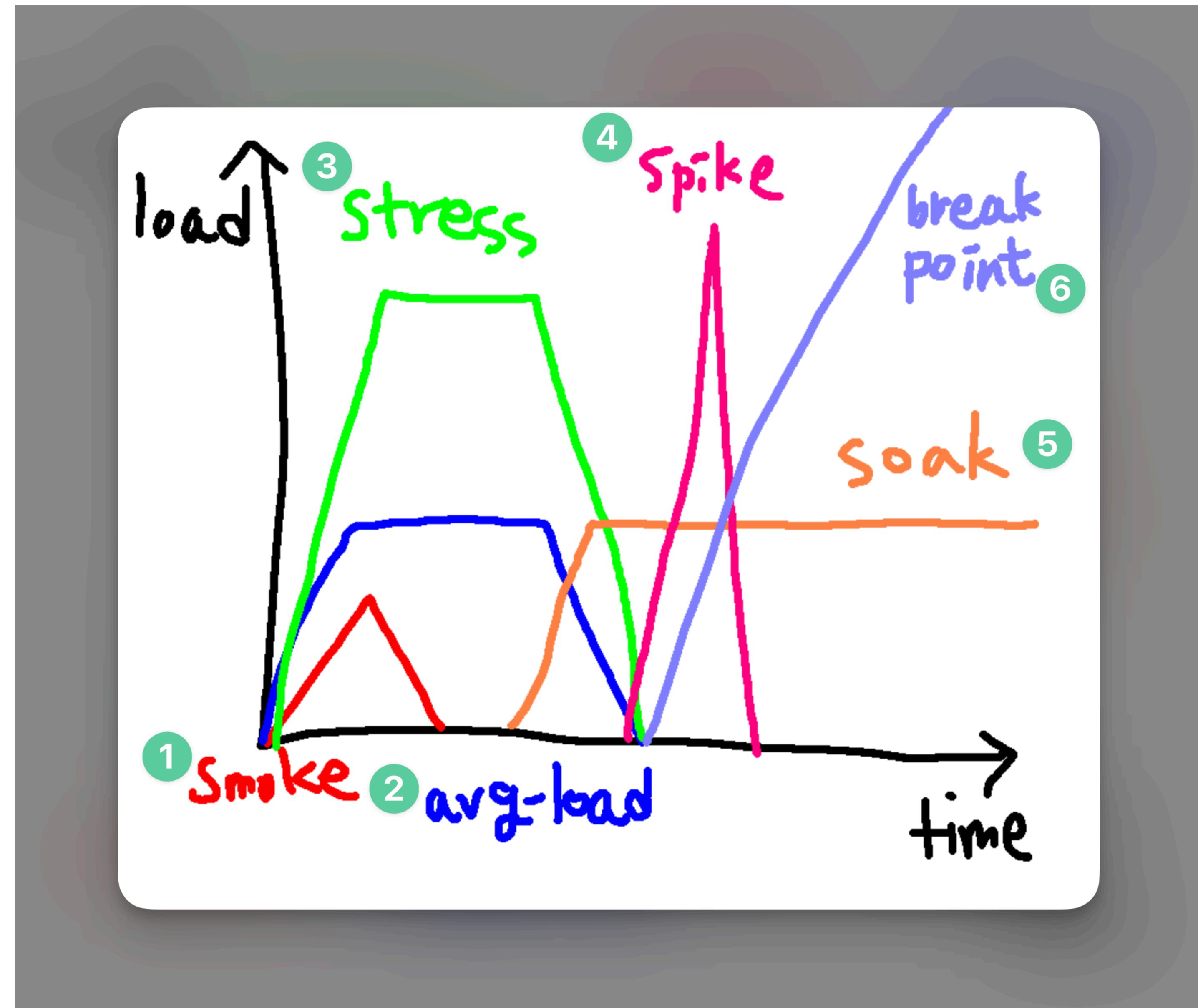
완전 처음이라면

- ab, k6
- 명령어 하나, 파일 하나 정도로 구동



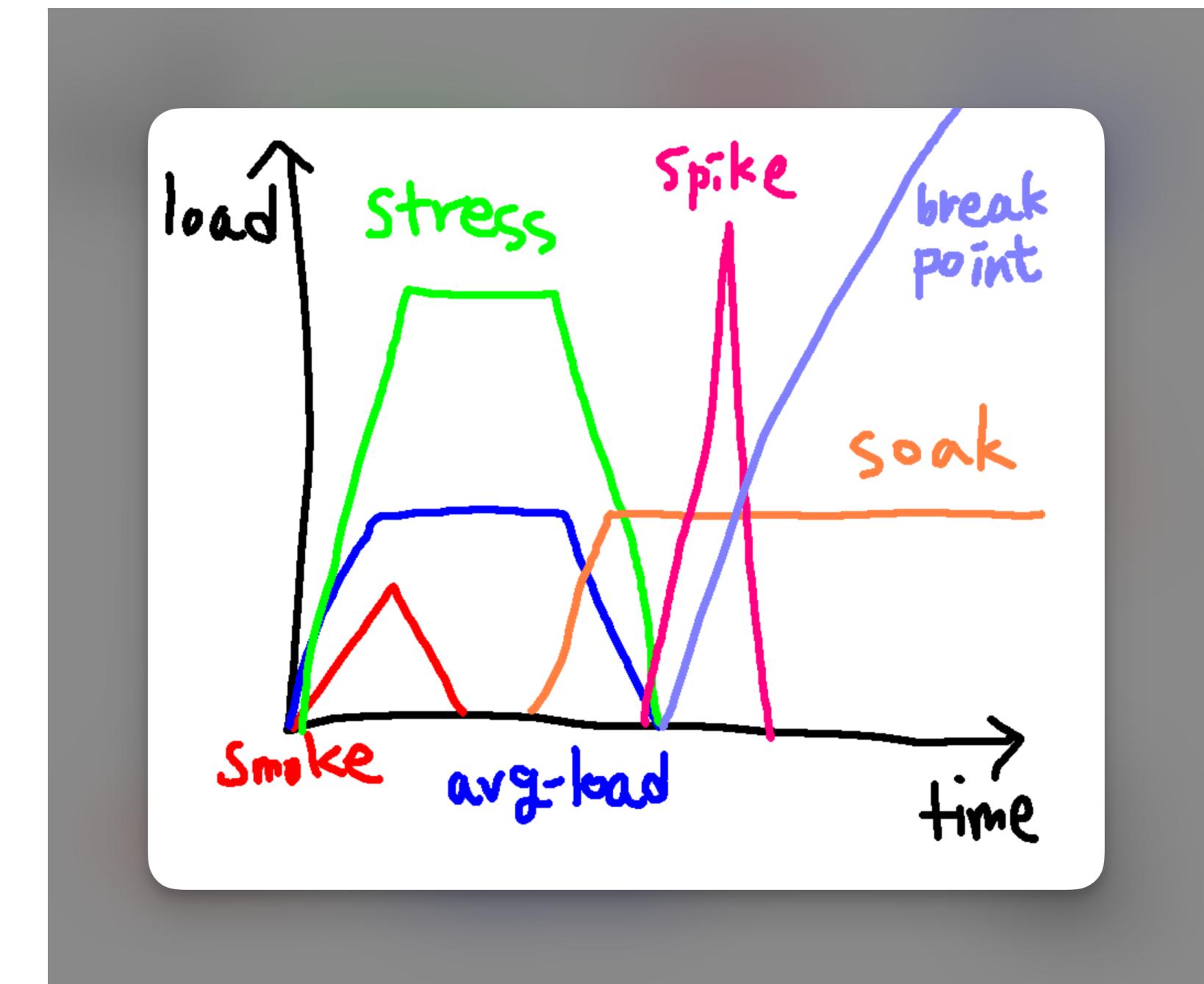
접근 순서는

접근 순서는



접근 순서는

- "작게, 자주" (Smoke Test) → 다양한 타입으로 확장
- Breakpoint test : 내 시스템의 한계치 체크



tl;dr:

1. StepFunctions, IoT (MQTT), Taurus
Let's go

tl;dr:

1. StepFunctions, Taurus, IoT (MQTT), Let's go
2. 부하테스트 시작은 Smoke Test 부터

(3. 항상 모든 리소스에 태그달기)