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

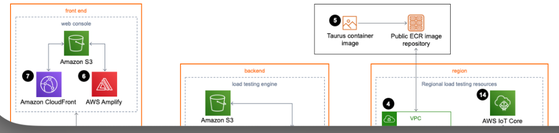
지정된 날짜 또는 반복 날짜에 자동으로 시작하도록 로드 테스트를 예약합니다.

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 리전용 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

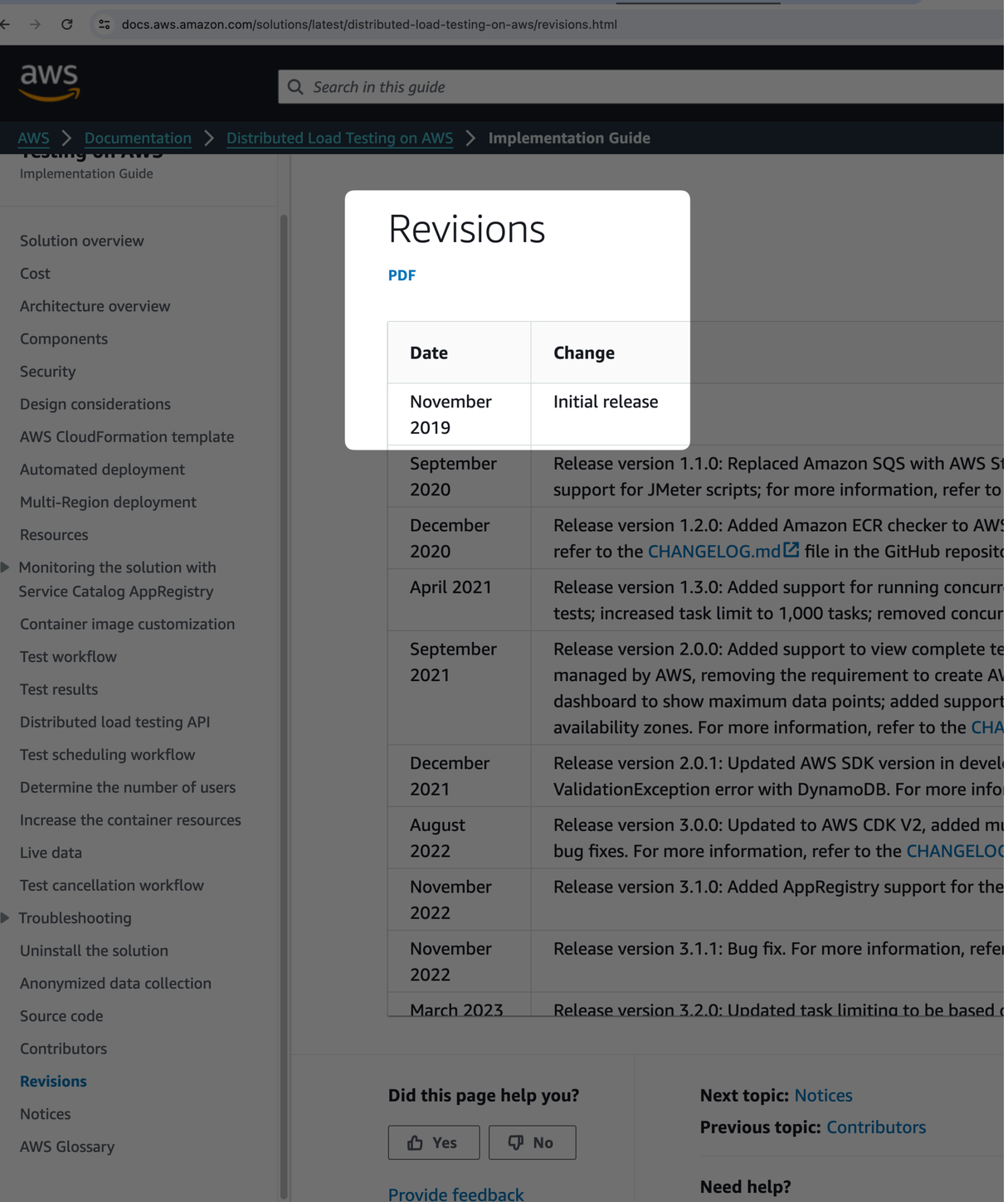
Amazon Web Services 중국 리전에서 이 솔루션을 사용하고 싶으신가요? 여기를 클릭하세요.

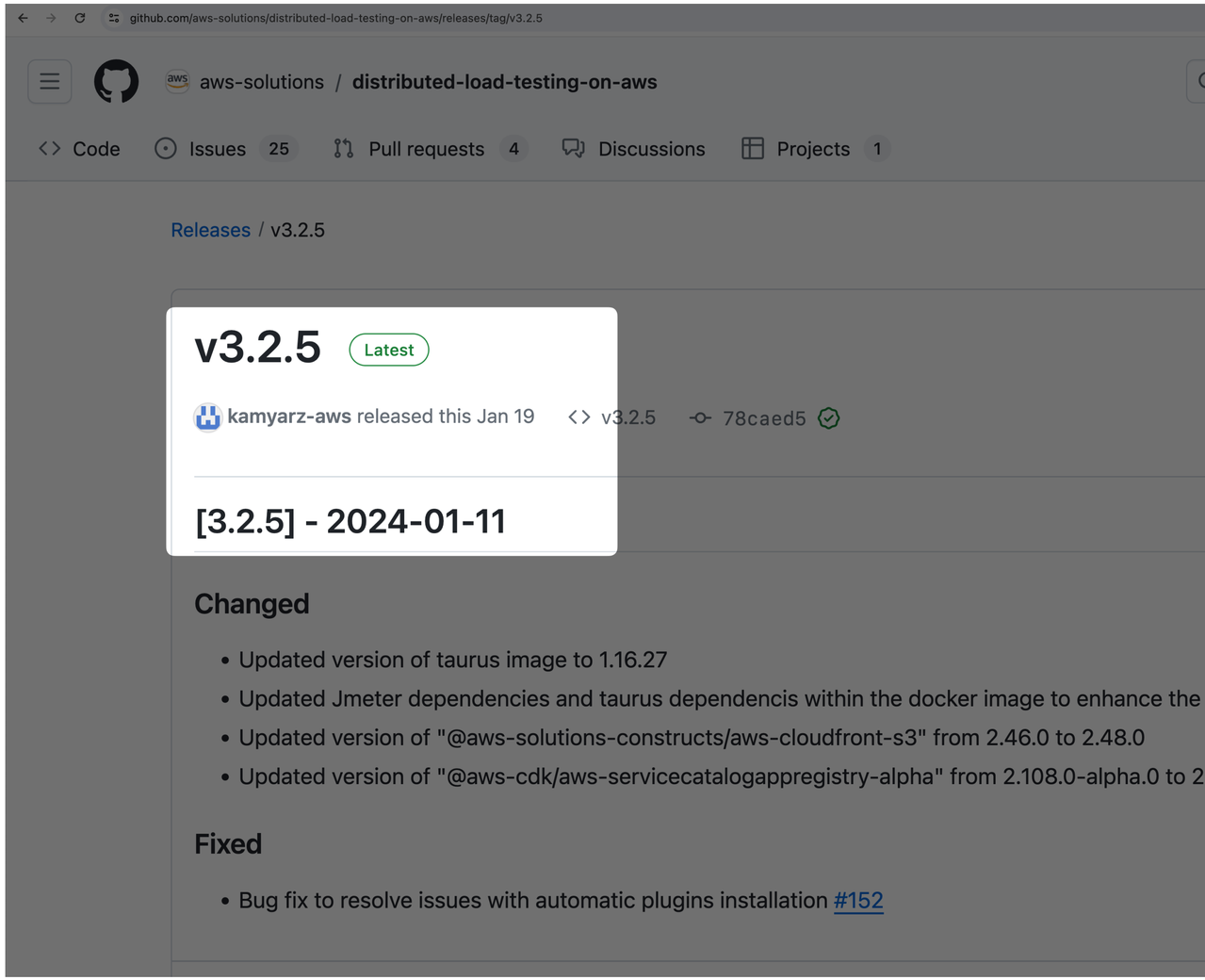
도움이 필요하신가요? 파트너를 통해 배포하세요.

이 배포를 지원할 수 있는 AWS Certified 서드파티 전문가를 찾아보세요.

장점들

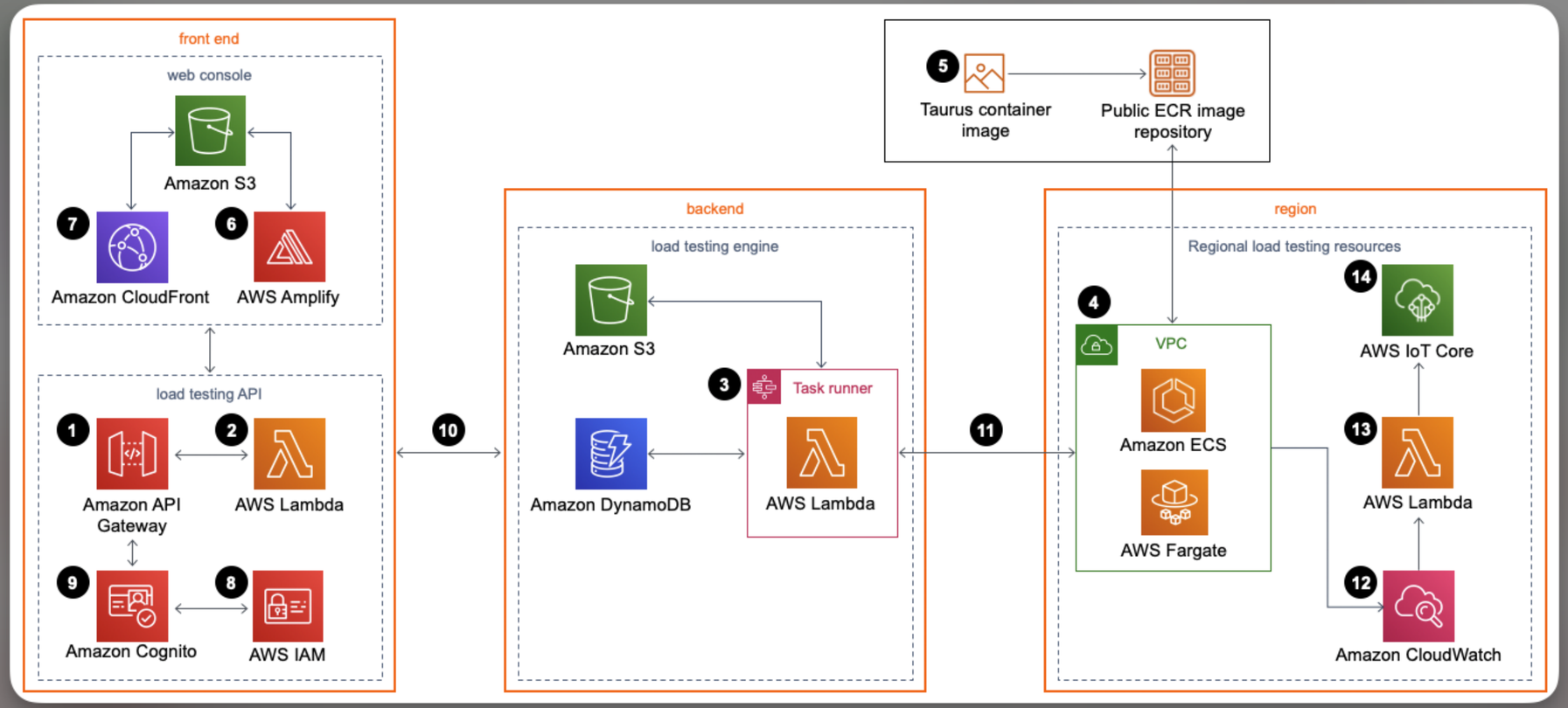
- AWS 리소스를 복합적으로 사용
- 수평 확장에 용이
- 실시간 조회 가능
- 업데이트가 꾸준히 이루어지고 있음

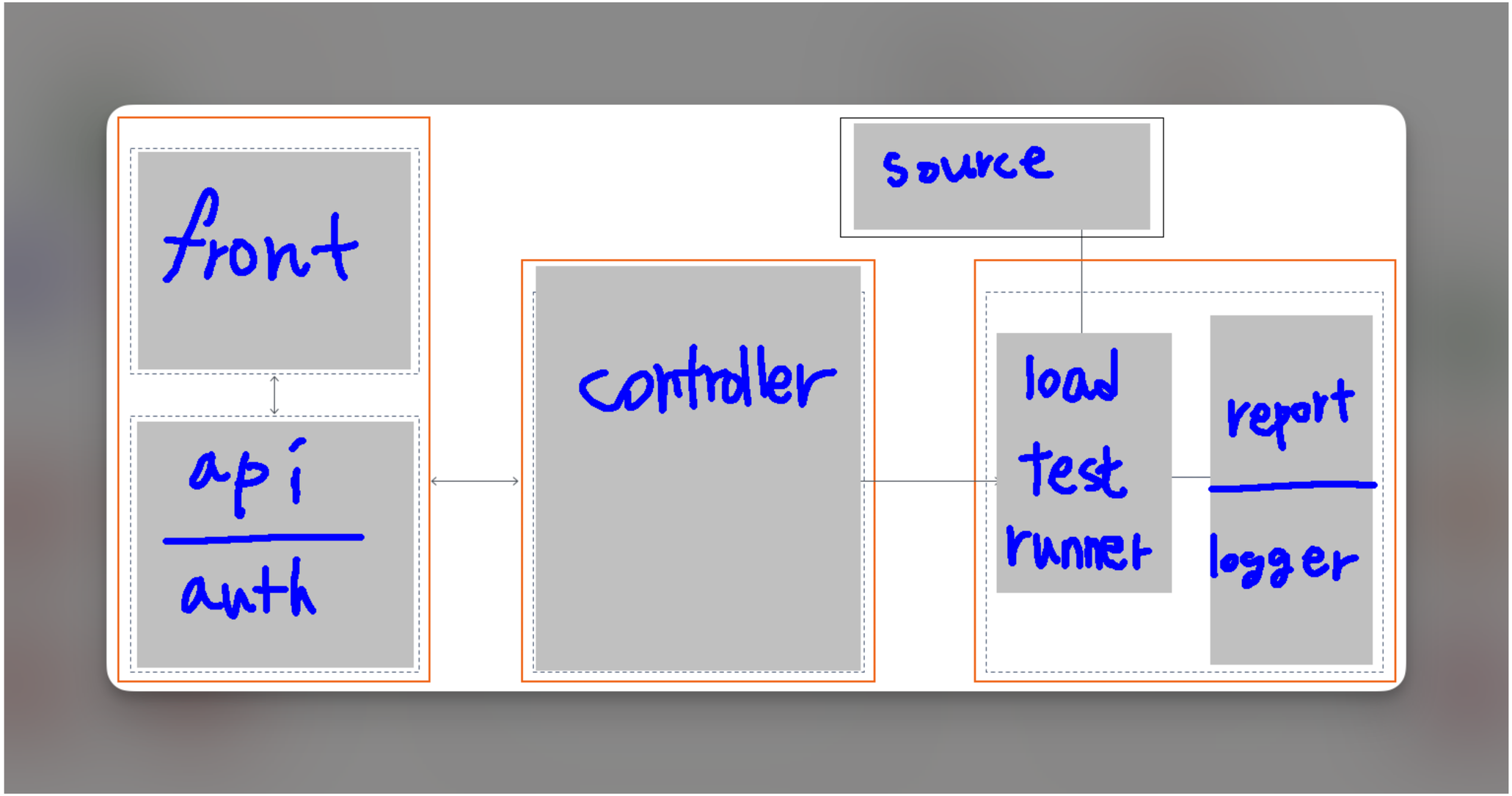

docs.aws.amazon.com/solutions/latest/distributed-load-testing-on-aws/revisions.html
aws
Search in this guide
AWS > Documentation > Distributed Load Testing on AWS > Implementation Guide
Implementation Guide
Solution overview
Cost
Architecture overview
Components
Security
Design considerations
AWS CloudFormation template
Automated deployment
Multi-Region deployment
Resources
Monitoring the solution with Service Catalog AppRegistry
Container image customization
Test workflow
Test results
Distributed load testing API
Test scheduling workflow
Determine the number of users
Increase the container resources
Live data
Test cancellation workflow
Troubleshooting
Uninstall the solution
Anonymized data collection
Source code
Contributors
Revisions
Notices
AWS Glossary
Revisions
PDF
Date
Change
November 2019
Initial release
September 2020
December 2020
April 2021
September 2021
December 2021
August 2022
November 2022
November 2022
March 2023
Did this page help you?
Yes
No
Provide feedback
Next topic: Notices
Previous topic: Contributors
Need help?

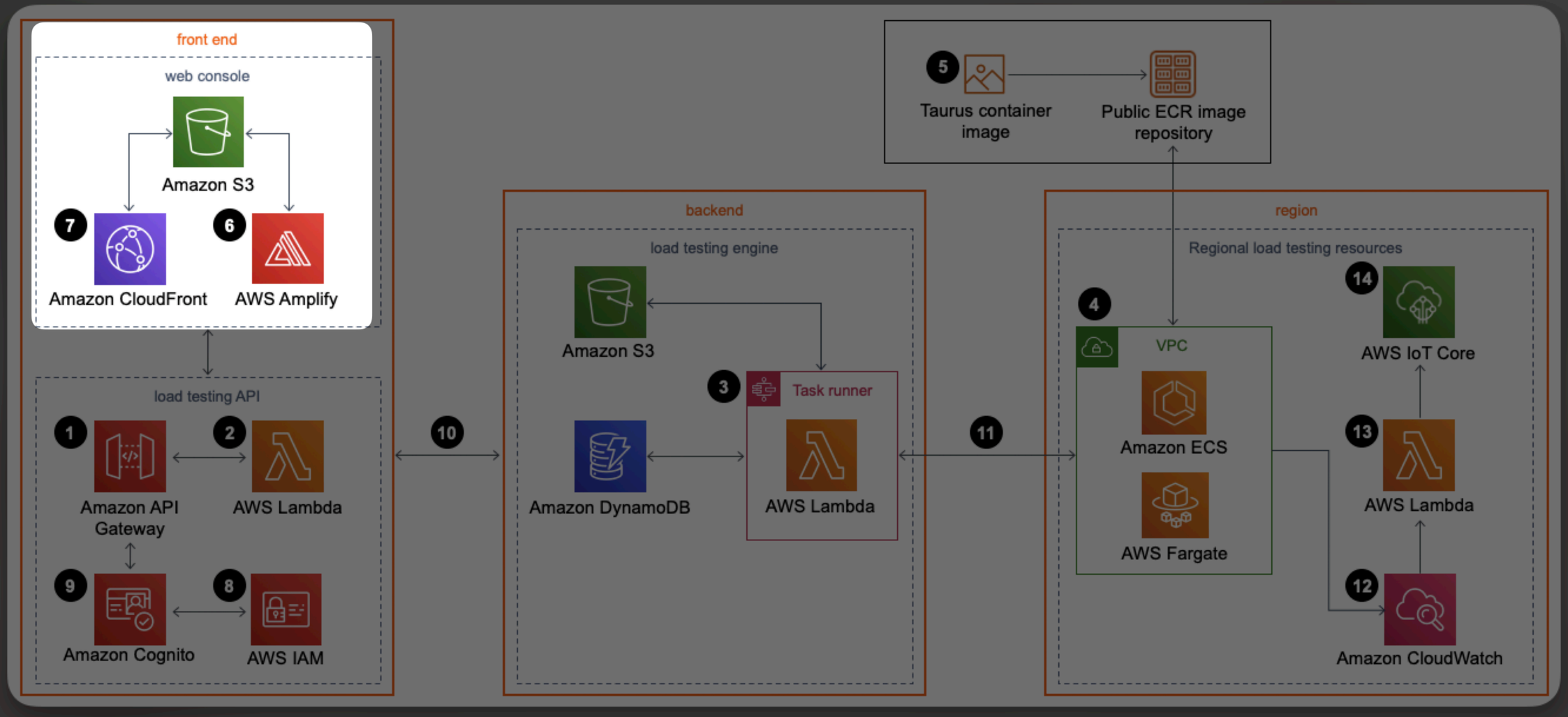

github.com/aws-solutions/distributed-load-testing-on-aws/releases/tag/v3.2.5
aws-solutions / distributed-load-testing-on-aws
Code
Issues 25
Pull requests 4
Discussions
Projects 1
Releases / v3.2.5
v3.2.5
Latest
kamyarz-aws released this Jan 19
v3.2.5
78caed5
[3.2.5] - 2024-01-11
Changed
Updated version of taurus image to 1.16.27
Updated Jmeter dependencies and taurus dependencis within the docker image to enhance the
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
Fixed
Bug fix to resolve issues with automatic plugins installation #152

DLT를 차근차근 이해해봅시다

(Distributed Load Testing)


front end
web console
Amazon S3
7
Amazon CloudFront
6
AWS Amplify
load testing API
1
Amazon API Gateway
2
AWS Lambda
9
Amazon Cognito
8
AWS IAM
10
backend
load testing engine
Amazon S3
3
Task runner
Amazon DynamoDB
AWS Lambda
11
5
Taurus container image
Public ECR image repository
region
Regional load testing resources
4
VPC
Amazon ECS
AWS Fargate
14
AWS IoT Core
13
AWS Lambda
12
Amazon CloudWatch


front
api
auth
controller
source
load
test
runner
report
logger


front end
web console
Amazon S3
7
Amazon CloudFront
6
AWS Amplify
load testing API
1
Amazon API Gateway
2
AWS Lambda
9
Amazon Cognito
8
AWS IAM
10
backend
load testing engine
Amazon S3
3
Task runner
Amazon DynamoDB
AWS Lambda
11
5
Taurus container image
Public ECR image repository
region
Regional load testing resources
4
VPC
Amazon ECS
AWS Fargate
14
AWS IoT Core
13
AWS Lambda
12
Amazon CloudWatch

github.com/aws-solutions/distributed-load-testing-on-aws/blob/main/source/console/src/App.js

Files

main

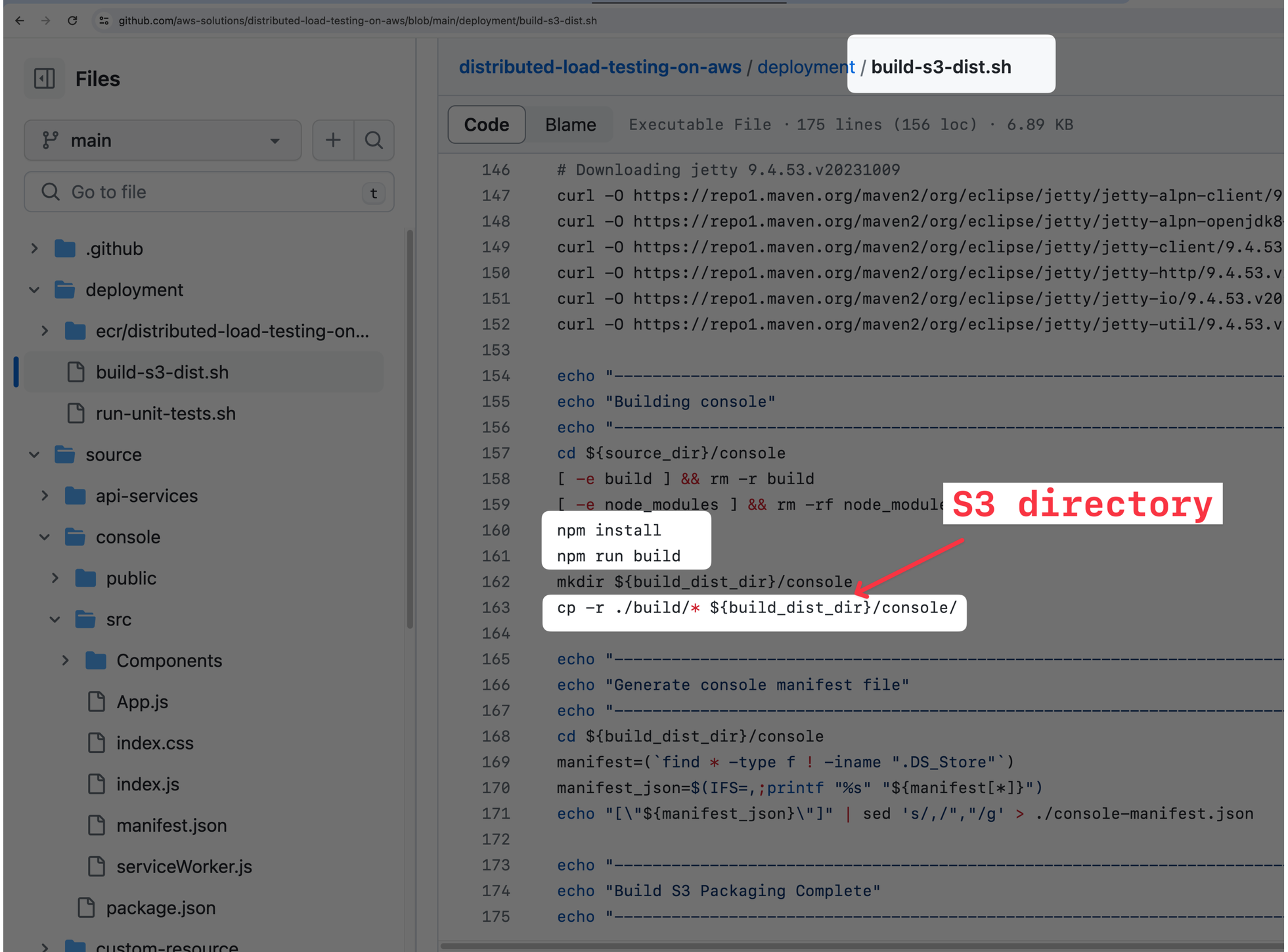
Go to file t

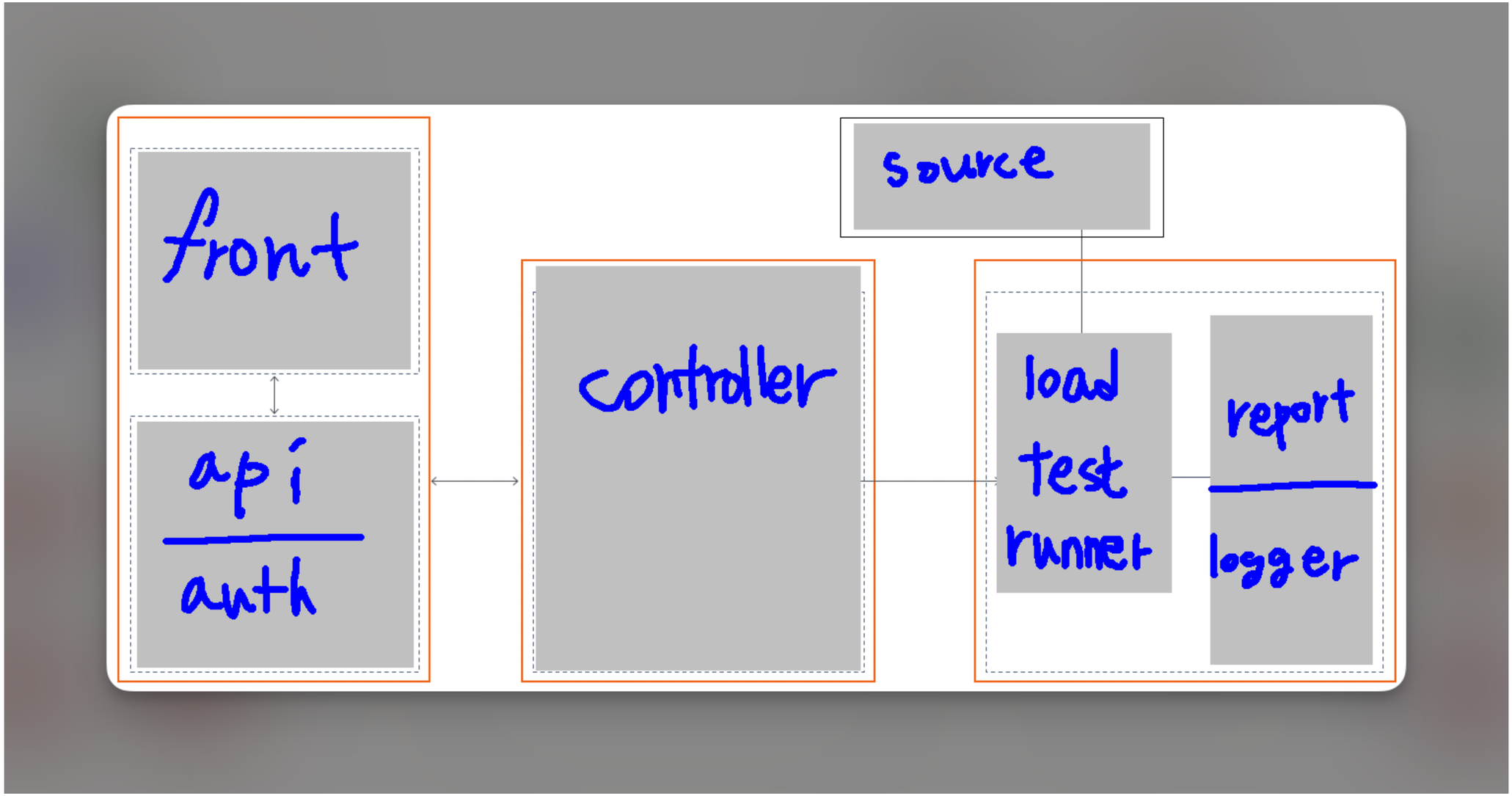
- .github
- deployment
  - ecr/distributed-load-testing-on...
  - build-s3-dist.sh
  - run-unit-tests.sh
- source
  - api-services
  - console
    - public
    - src
      - Components
      - App.js
      - index.css
      - index.js
      - manifest.json
      - serviceWorker.js
    - package.json
  - custom-resource

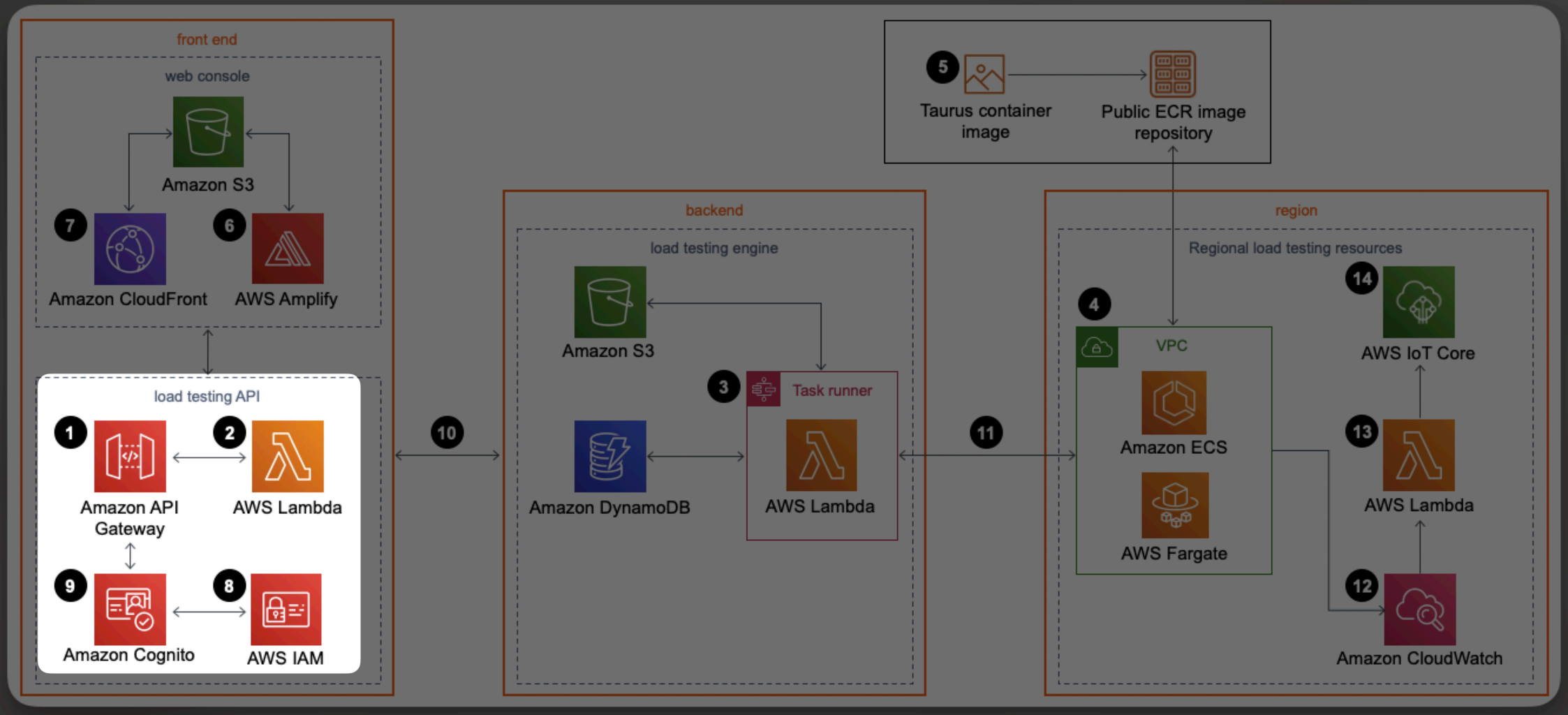
distributed-load-testing-on-aws / source / console / src / App.js

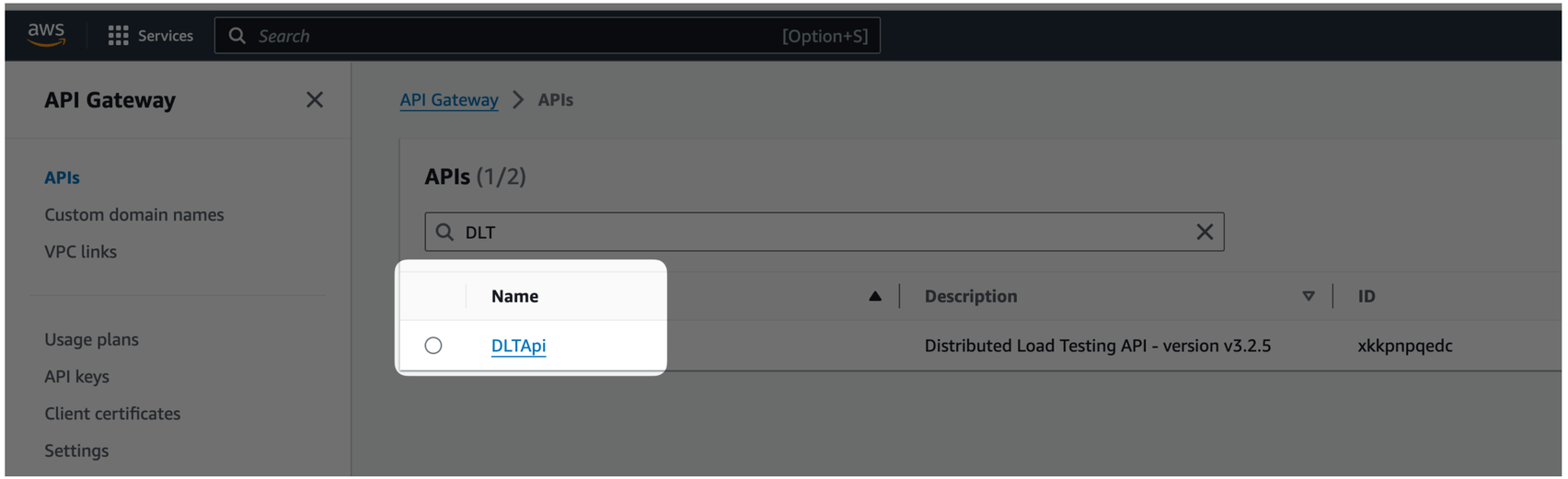
Code Blame 171 lines (158 loc) · 5.36 KB

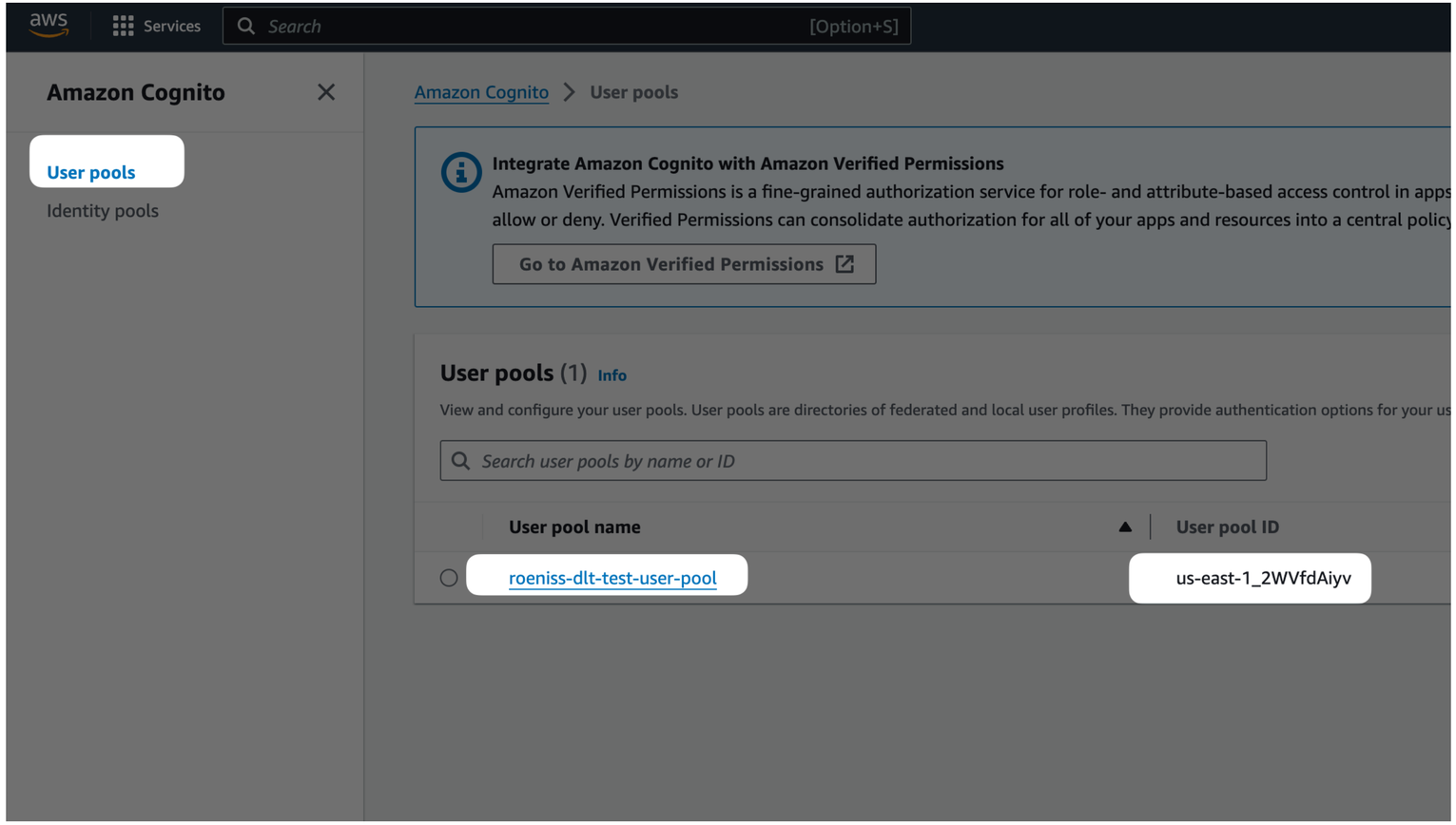
```
4  import React from "react";
5  import { BrowserRouter as Router, Route, Switch, Link } from "react-router
6  import { Collapse, Navbar, NavbarToggler, NavbarBrand, Nav, NavItem } from
7
8  //Amplify
9  import { Amplify, Auth } from "aws-amplify";
10 import { withAuthenticator } from "@aws-amplify/ui-react";
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 class App extends React.Component {
31   constructor(props) {
32     super(props);
33     this.noMatch = this.noMatch.bind(this);
34     this.signOut = this.signOut.bind(this);
35     this.toggleNavbar = this.toggleNavbar.bind(this);
```

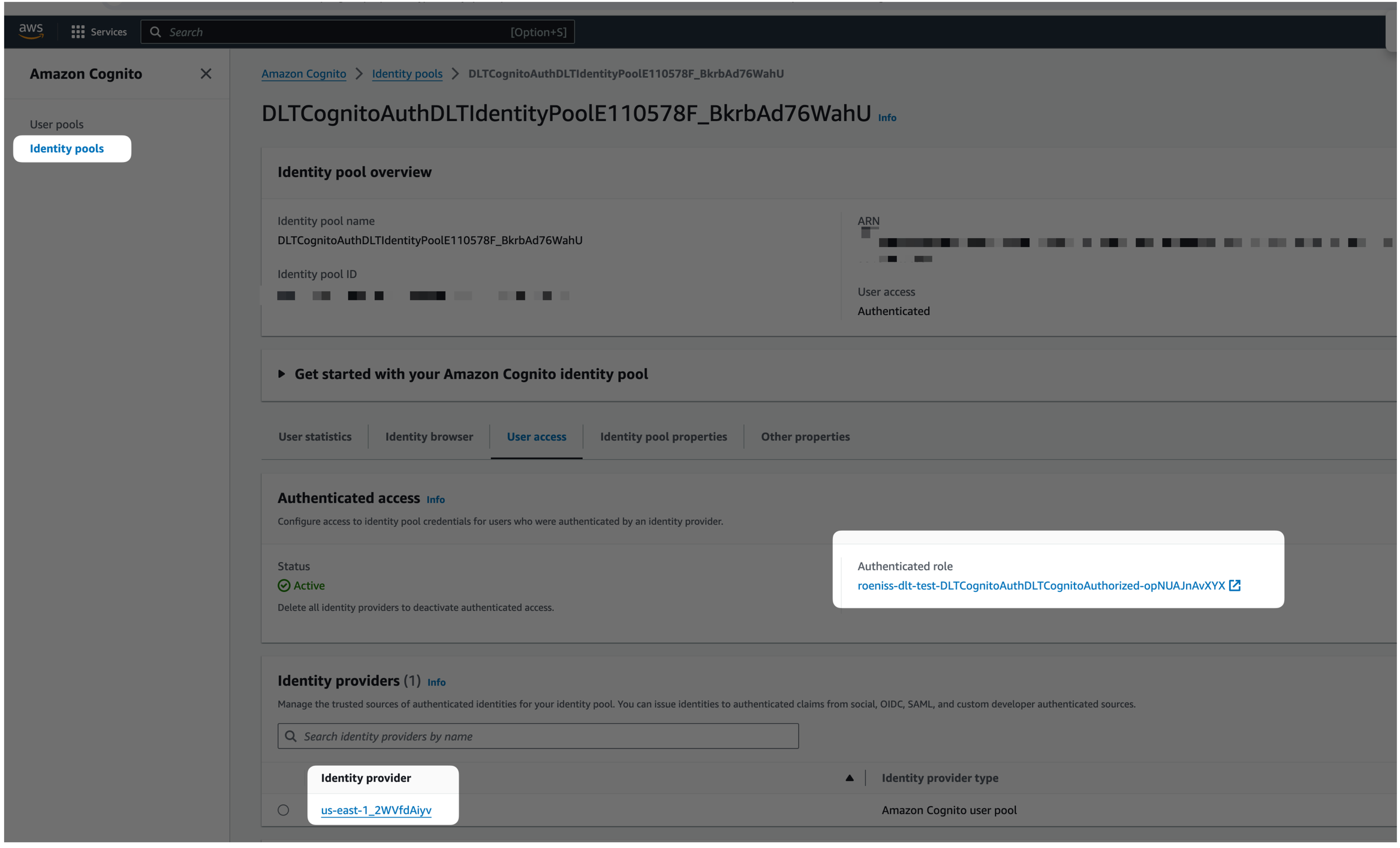

github.com/aws-solutions/distributed-load-testing-on-aws/blob/main/deployment/build-s3-dist.sh
Files
main
Go to file
t
.github
deployment
ecr/distributed-load-testing-on...
build-s3-dist.sh
run-unit-tests.sh
source
api-services
console
public
src
Components
App.js
index.css
index.js
manifest.json
serviceWorker.js
package.json
distributed-load-testing-on-aws / deployment / build-s3-dist.sh
Code
Blame
Executable File · 175 lines (156 loc) · 6.89 KB
Downloading jetty 9.4.53.v20231009
curl -O https://repo1.maven.org/maven2/org/eclipse/jetty/jetty-alpn-client/9
curl -O https://repo1.maven.org/maven2/org/eclipse/jetty/jetty-alpn-openjdk8
curl -O https://repo1.maven.org/maven2/org/eclipse/jetty/jetty-client/9.4.53
curl -O https://repo1.maven.org/maven2/org/eclipse/jetty/jetty-http/9.4.53.v
curl -O https://repo1.maven.org/maven2/org/eclipse/jetty/jetty-io/9.4.53.v20
curl -O https://repo1.maven.org/maven2/org/eclipse/jetty/jetty-util/9.4.53.v
echo "--
echo "Building console"
echo "--
cd ${source_dir}/console
[-e build] && rm -r build
[-e node_modules] && rm -rf node_module
npm install
npm run build
mkdir ${build_dist_dir}/console
cp -r ./build/* ${build_dist_dir}/console/
S3 directory
echo "--
echo "Generate console manifest file"
echo "--
cd ${build_dist_dir}/console
manifest=(`find * -type f ! -iname ".DS_Store"`)
manifest_json=$(IFS=,;printf "%s" "${manifest[*]}")
echo "[\"${manifest_json}\"]" | sed 's/,/","/g' > ./console-manifest.json
echo "--
echo "Build S3 Packaging Complete"
echo "--

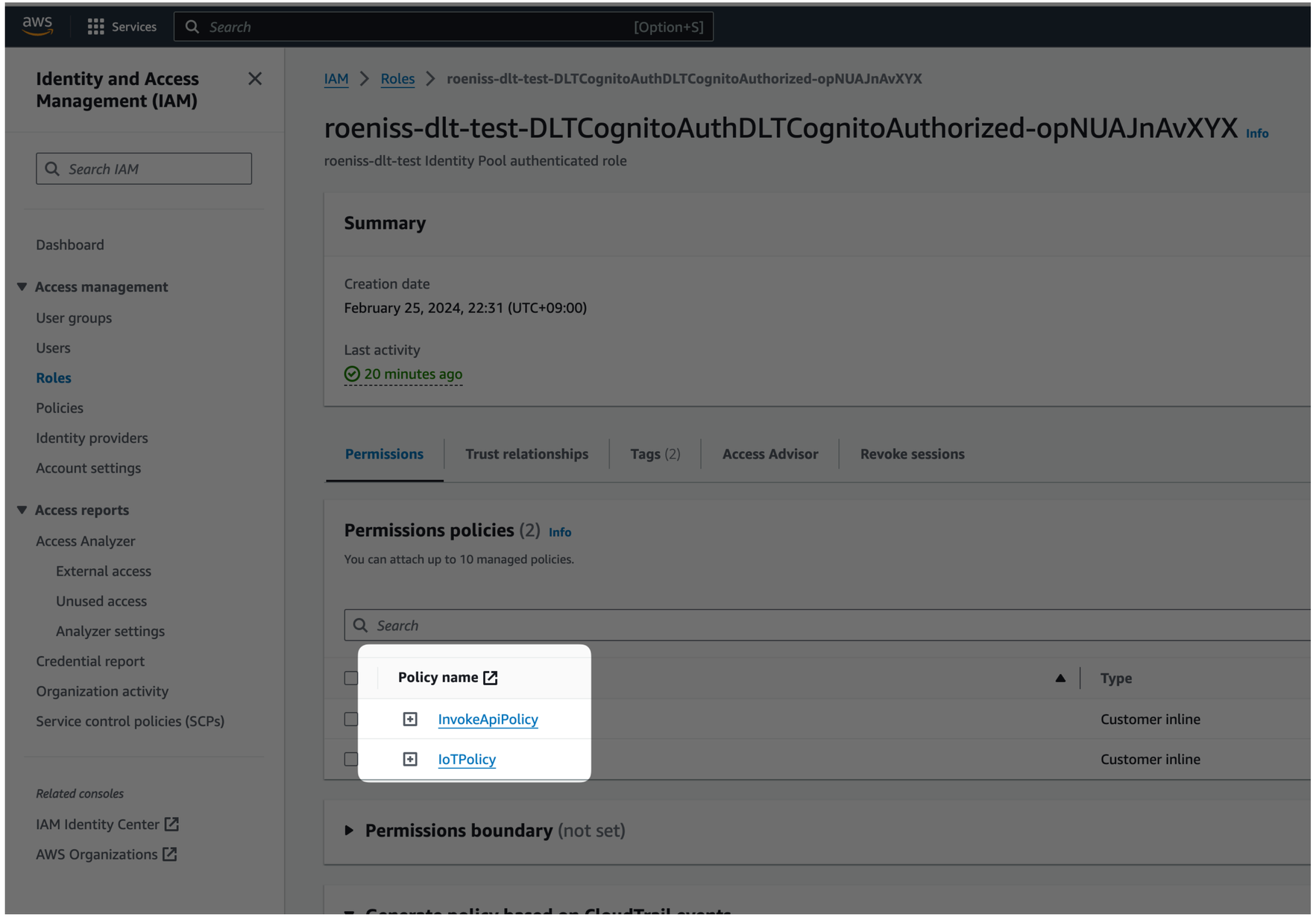

front
api
auth
controller
source
load
test
runner
report
logger

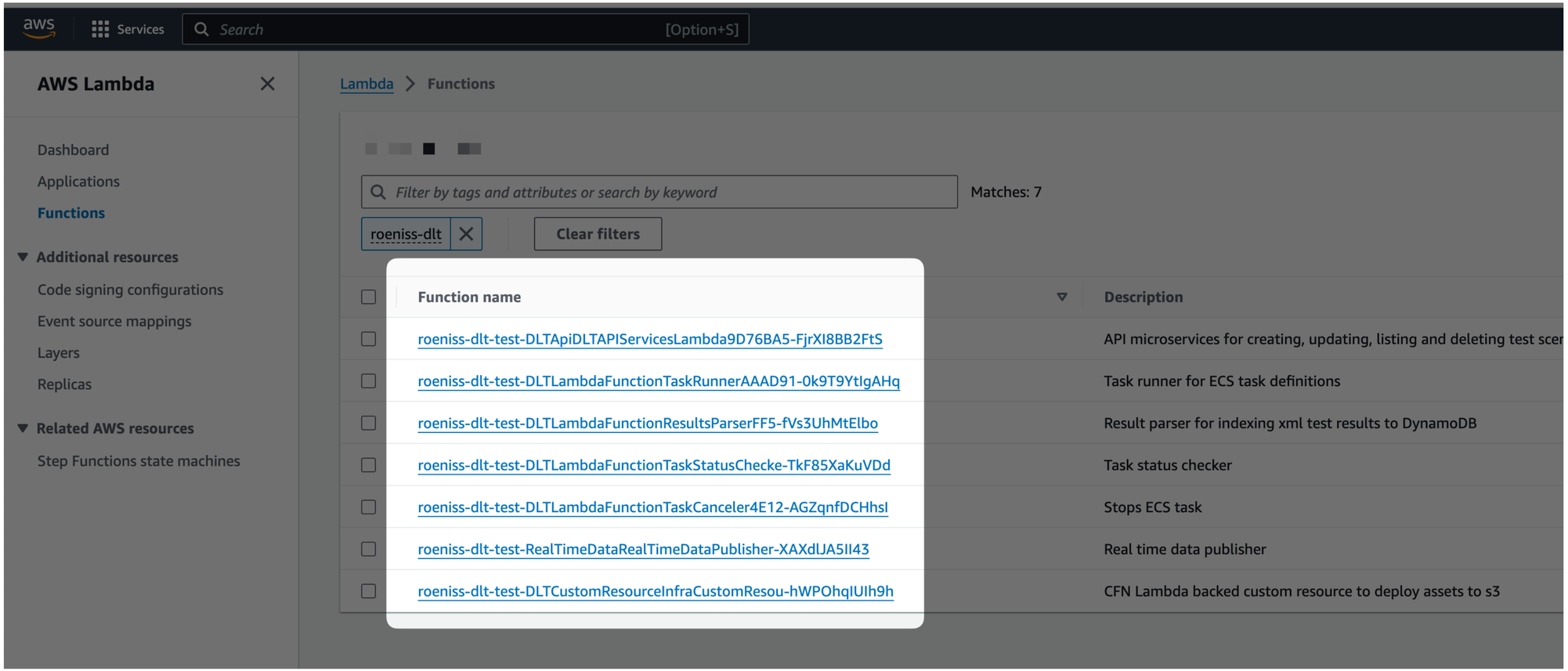

front end
web console
Amazon S3
7
Amazon CloudFront
6
AWS Amplify
load testing API
1
Amazon API Gateway
2
AWS Lambda
9
Amazon Cognito
8
AWS IAM
10
backend
load testing engine
Amazon S3
3
Task runner
Amazon DynamoDB
AWS Lambda
11
5
Taurus container image
Public ECR image repository
region
Regional load testing resources
4
VPC
Amazon ECS
AWS Fargate
14
AWS IoT Core
13
AWS Lambda
12
Amazon CloudWatch

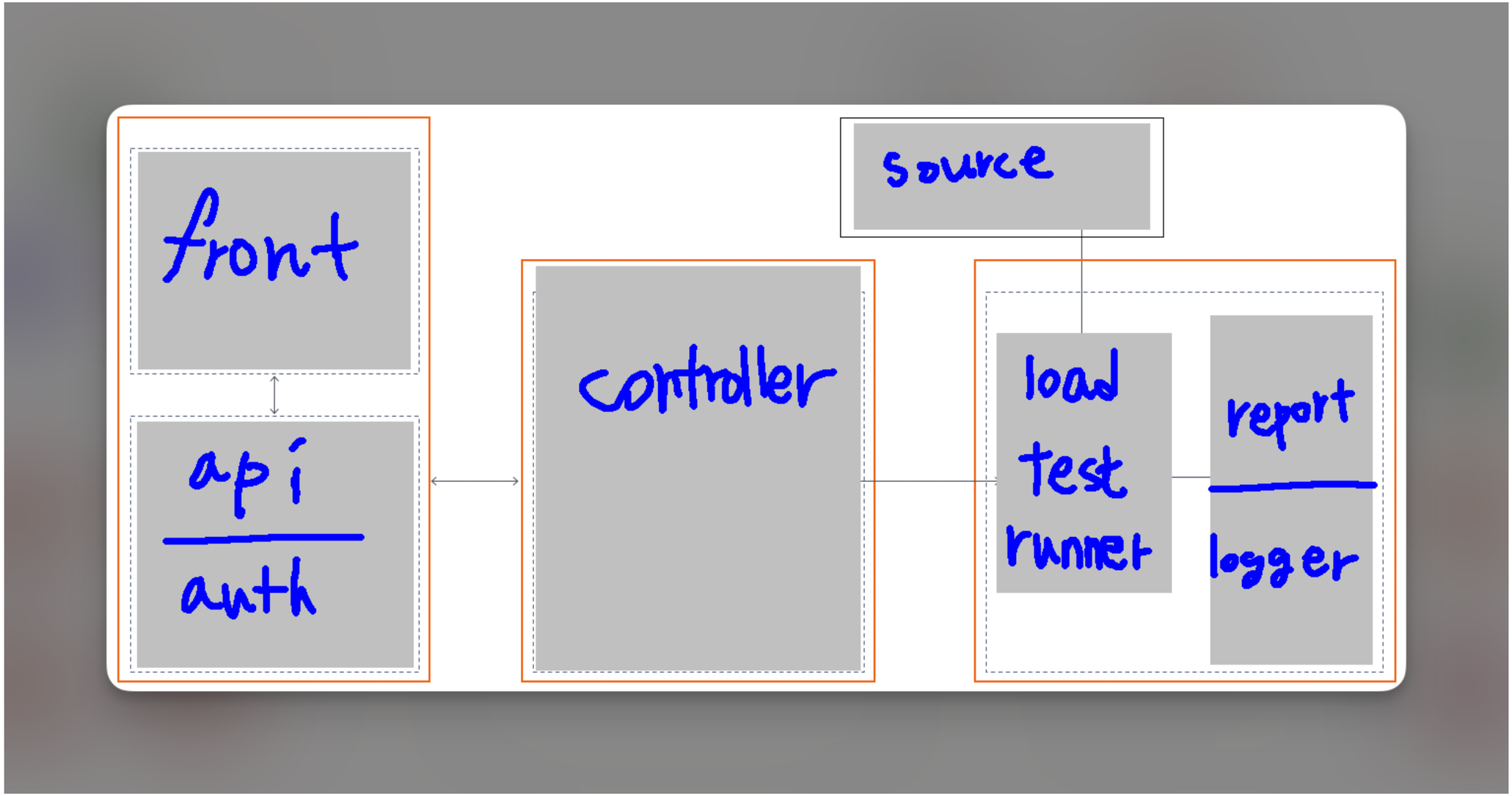

aws
Services
Search
[Option+S]
API Gateway
APIs
Custom domain names
VPC links
Usage plans
API keys
Client certificates
Settings
API Gateway
APIs
APIs (1/2)
DLT
Name
Description
ID
DLTApi
Distributed Load Testing API - version v3.2.5
xkkpnpqedc

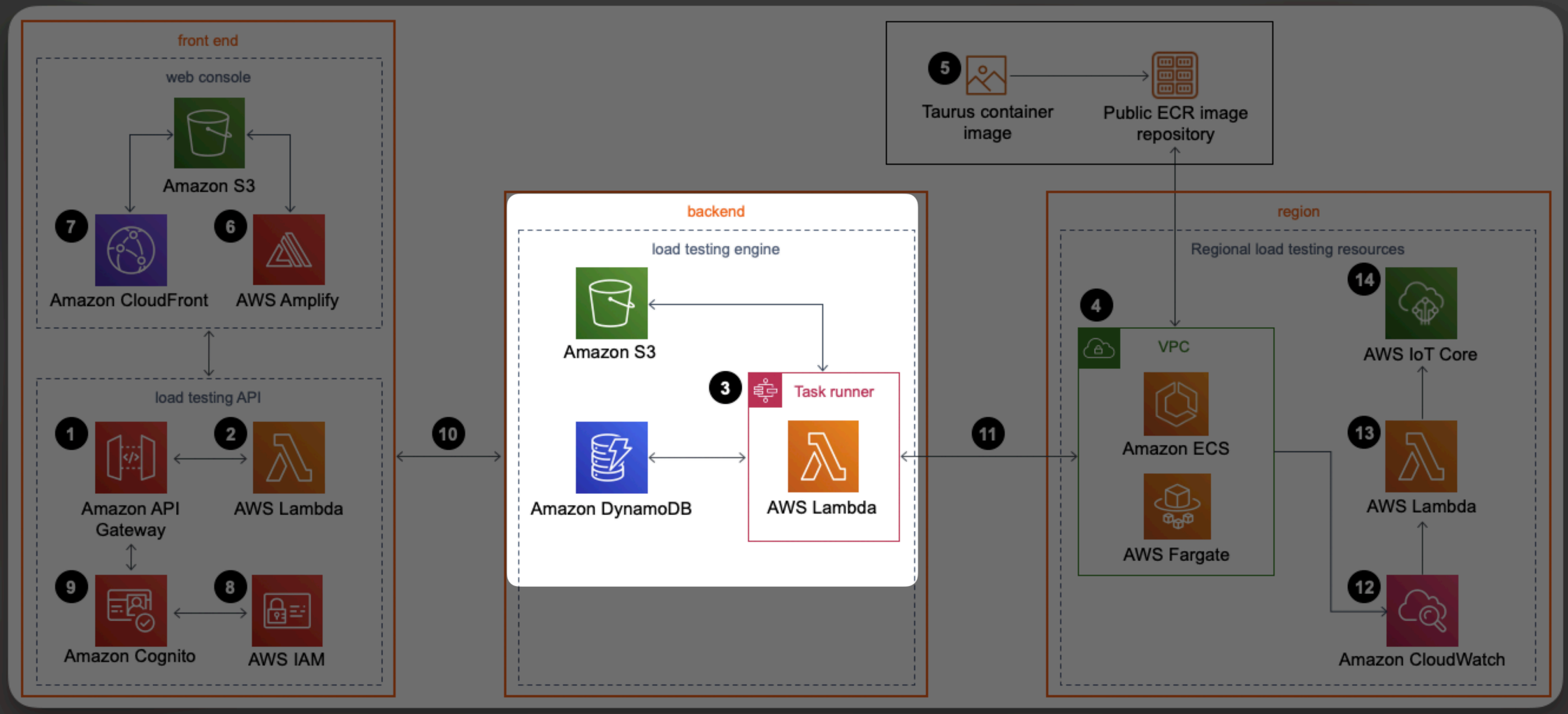

aws
Services
Search
[Option+S]
Amazon Cognito
User pools
Identity pools
Amazon Cognito > User pools
Integrate Amazon Cognito with Amazon Verified Permissions
Amazon Verified Permissions is a fine-grained authorization service for role- and attribute-based access control in apps
allow or deny. Verified Permissions can consolidate authorization for all of your apps and resources into a central policy
Go to Amazon Verified Permissions
User pools (1) Info
View and configure your user pools. User pools are directories of federated and local user profiles. They provide authentication options for your us
Search user pools by name or ID
User pool name
User pool ID
roeniss-dlt-test-user-pool
us-east-1_2WVfdAiyv

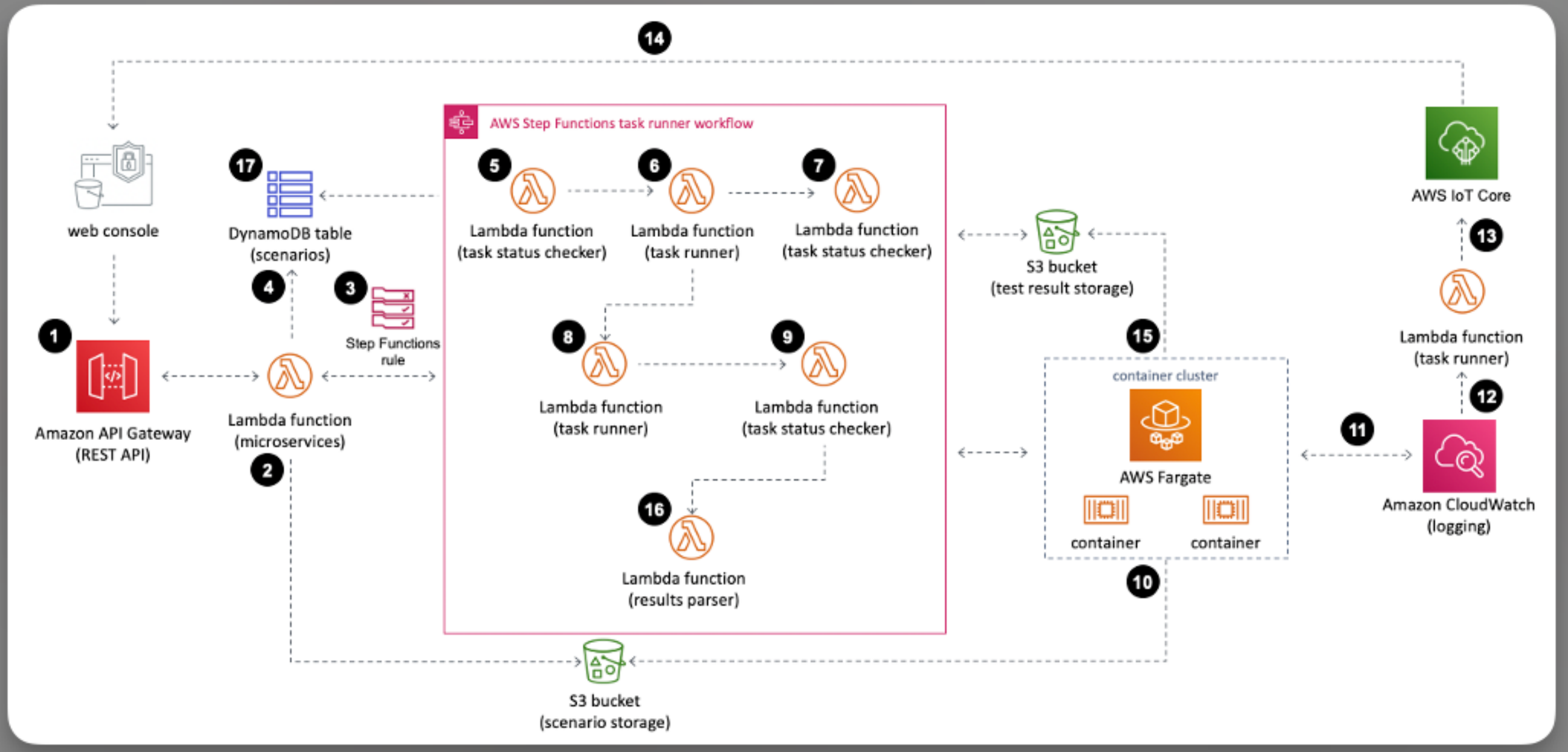

aws
Services
Search
[Option+S]
Amazon Cognito
User pools
Identity pools
Amazon Cognito > Identity pools > DLTCognitoAuthDLTIdentityPoolE110578F_BkrbAd76WahU
DLTCognitoAuthDLTIdentityPoolE110578F_BkrbAd76WahU Info
Identity pool overview
Identity pool name
DLTCognitoAuthDLTIdentityPoolE110578F_BkrbAd76WahU
Identity pool ID
ARN
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-DLTCognitoAuthDLTCognitoAuthorized-opNUAJnAvXYX
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
Identity provider type
us-east-1_2WVfdAiyv
Amazon Cognito user pool

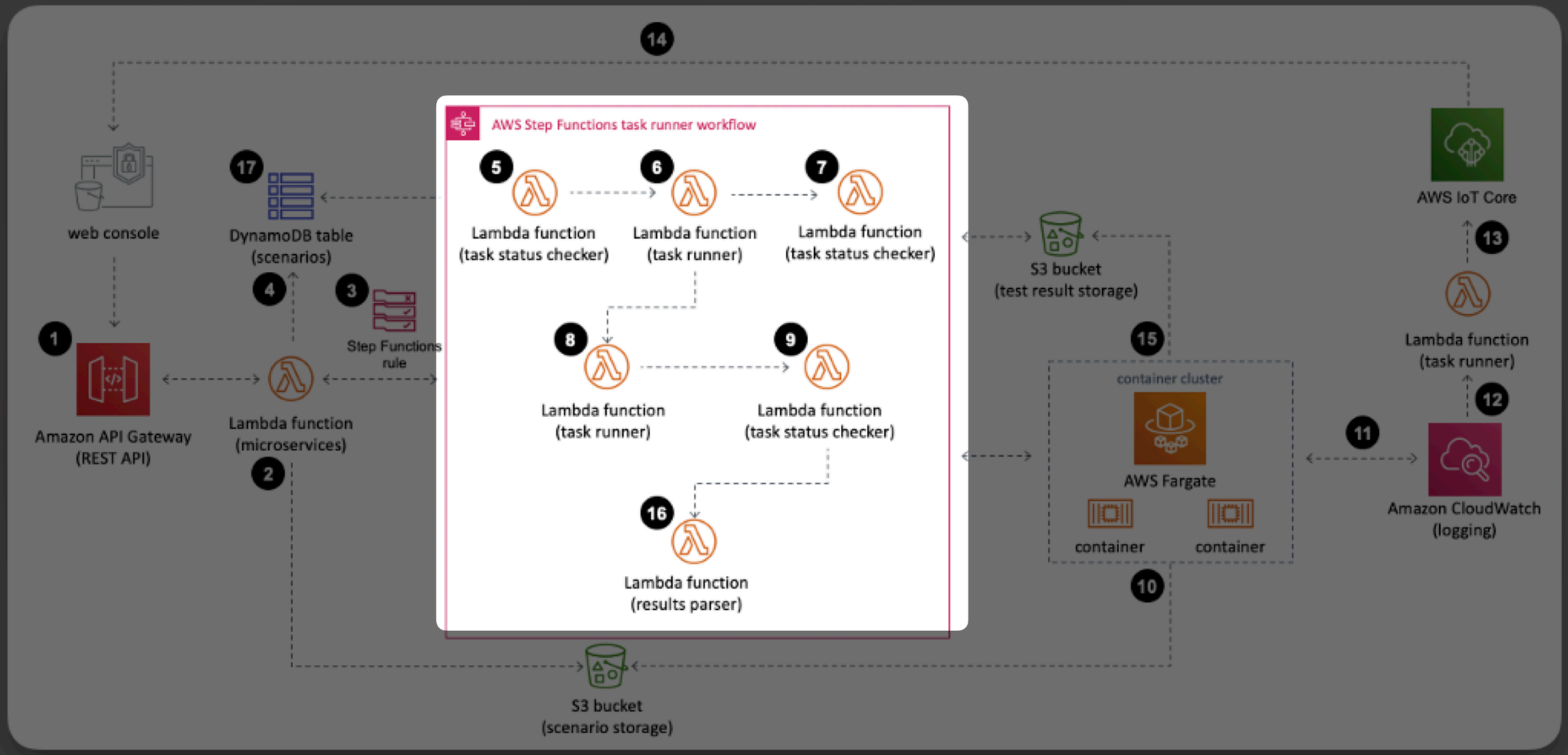

aws
Services
Search
[Option+S]
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access Analyzer
External access
Unused access
Analyzer settings
Credential report
Organization activity
Service control policies (SCPs)
Related consoles
IAM Identity Center
AWS Organizations
IAM > Roles > roeniss-dlt-test-DLTCognitoAuthDLTCognitoAuthorized-opNUAJnAvXYX
roeniss-dlt-test-DLTCognitoAuthDLTCognitoAuthorized-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)

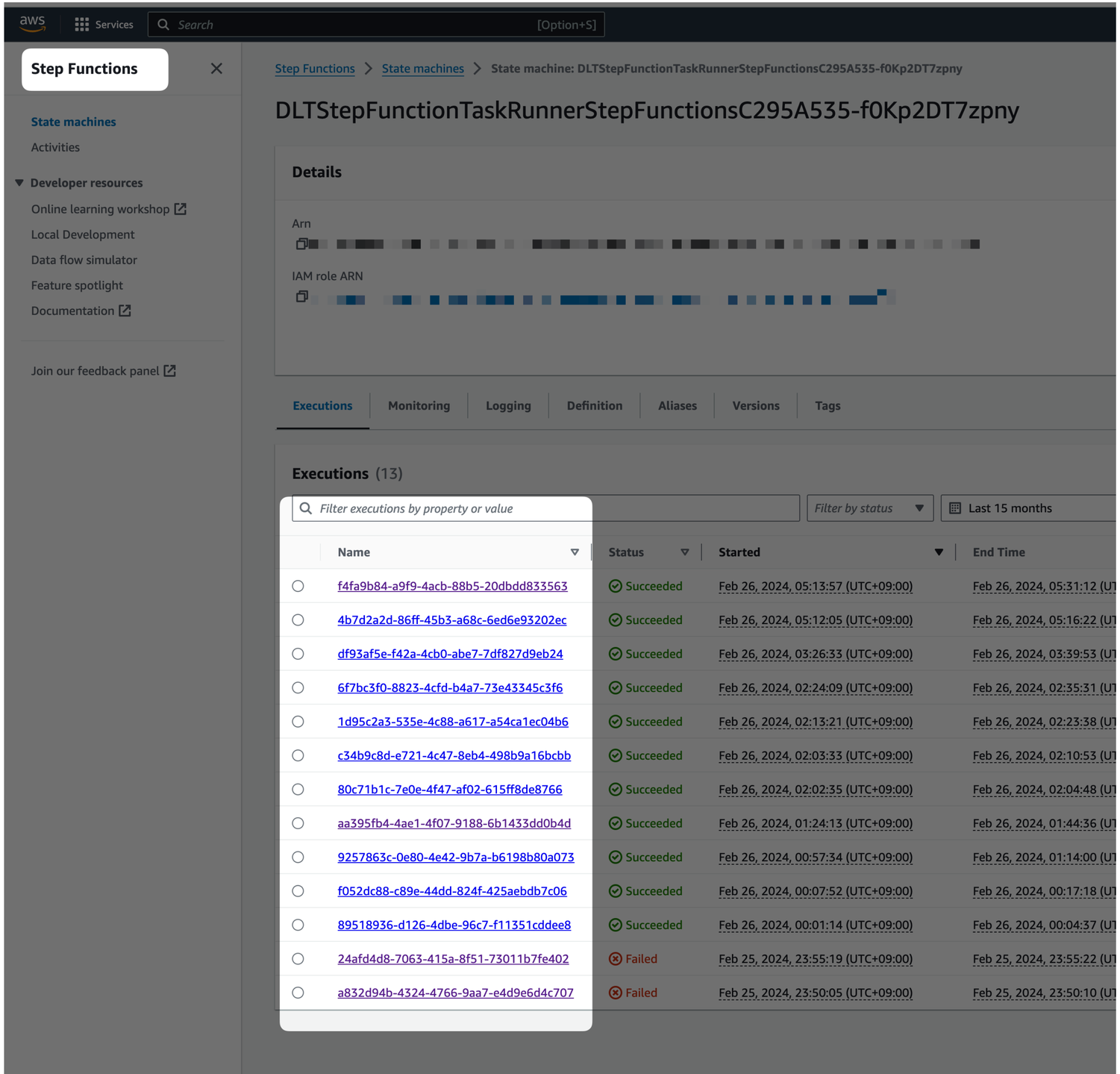

aws
Services
Search
[Option+S]
AWS Lambda
Dashboard
Applications
Functions
Additional resources
Code signing configurations
Event source mappings
Layers
Replicas
Related AWS resources
Step Functions state machines
Lambda
Functions
Filter by tags and attributes or search by keyword
Matches: 7
roeniss-dlt
Clear filters
Function name
Description
roeniss-dlt-test-DLTApiDLTAPIServicesLambda9D76BA5-FjrXI8BB2FtS
API microservices for creating, updating, listing and deleting test scer
roeniss-dlt-test-DLTLambdaFunctionTaskRunnerAAAD91-0k9T9YtIgAHq
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-XAXdlJA5II43
Real time data publisher
roeniss-dlt-test-DLTCustomResourceInfraCustomResou-hWPOhqIUIh9h
CFN Lambda backed custom resource to deploy assets to s3


front
api
auth
controller
source
load
test
runner
report
logger


front end
web console
Amazon S3
7
Amazon CloudFront
6
AWS Amplify
load testing API
1
Amazon API Gateway
2
AWS Lambda
9
Amazon Cognito
8
AWS IAM
10
backend
load testing engine
Amazon S3
3
Task runner
Amazon DynamoDB
AWS Lambda
11
5
Taurus container image
Public ECR image repository
region
Regional load testing resources
4
VPC
Amazon ECS
AWS Fargate
14
AWS IoT Core
13
AWS Lambda
12
Amazon CloudWatch


14
AWS Step Functions task runner workflow
web console
17
DynamoDB table
(scenarios)
5
Lambda function
(task status checker)
6
Lambda function
(task runner)
7
Lambda function
(task status checker)
AWS IoT Core
S3 bucket
(test result storage)
13
Lambda function
(task runner)
1
4
3
Step Functions
rule
8
Lambda function
(task runner)
9
Lambda function
(task status checker)
15
container cluster
12
11
Amazon API Gateway
(REST API)
Lambda function
(microservices)
2
AWS Fargate
container
container
Amazon CloudWatch
(logging)
16
Lambda function
(results parser)
10
S3 bucket
(scenario storage)


14
AWS Step Functions task runner workflow
web console
17
DynamoDB table
(scenarios)
5
Lambda function
(task status checker)
6
Lambda function
(task runner)
7
Lambda function
(task status checker)
AWS IoT Core
S3 bucket
(test result storage)
13
Lambda function
(task runner)
1
4
3
Step Functions
rule
8
Lambda function
(task runner)
9
Lambda function
(task status checker)
15
container cluster
12
11
Amazon API Gateway
(REST API)
Lambda function
(microservices)
2
AWS Fargate
container
container
Amazon CloudWatch
(logging)
16
Lambda function
(results parser)
10
S3 bucket
(scenario storage)


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
Step Functions > State machines > State machine: DLTStepFunctionTaskRunnerStepFunctionsC295A535-f0Kp2DT7zpny
DLTStepFunctionTaskRunnerStepFunctionsC295A535-f0Kp2DT7zpny
Details
Arn
IAM role ARN
Executions
Monitoring
Logging
Definition
Aliases
Versions
Tags
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 (UT
4b7d2a2d-86ff-45b3-a68c-6ed6e93202ec Succeeded Feb 26, 2024, 05:12:05 (UTC+09:00) Feb 26, 2024, 05:16:22 (UT
df93af5e-f42a-4cb0-abe7-7df827d9eb24 Succeeded Feb 26, 2024, 03:26:33 (UTC+09:00) Feb 26, 2024, 03:39:53 (UT
6f7bc3f0-8823-4cfd-b4a7-73e43345c3f6 Succeeded Feb 26, 2024, 02:24:09 (UTC+09:00) Feb 26, 2024, 02:35:31 (UT
1d95c2a3-535e-4c88-a617-a54ca1ec04b6 Succeeded Feb 26, 2024, 02:13:21 (UTC+09:00) Feb 26, 2024, 02:23:38 (UT
c34b9c8d-e721-4c47-8eb4-498b9a16bcbb Succeeded Feb 26, 2024, 02:03:33 (UTC+09:00) Feb 26, 2024, 02:10:53 (UT
80c71b1c-7e0e-4f47-af02-615ff8de8766 Succeeded Feb 26, 2024, 02:02:35 (UTC+09:00) Feb 26, 2024, 02:04:48 (UT
aa395fb4-4ae1-4f07-9188-6b1433dd0b4d Succeeded Feb 26, 2024, 01:24:13 (UTC+09:00) Feb 26, 2024, 01:44:36 (UT
9257863c-0e80-4e42-9b7a-b6198b80a073 Succeeded Feb 26, 2024, 00:57:34 (UTC+09:00) Feb 26, 2024, 01:14:00 (UT
f052dc88-c89e-44dd-824f-425aebdb7c06 Succeeded Feb 26, 2024, 00:07:52 (UTC+09:00) Feb 26, 2024, 00:17:18 (UT
89518936-d126-4dbe-96c7-f11351cddee8 Succeeded Feb 26, 2024, 00:01:14 (UTC+09:00) Feb 26, 2024, 00:04:37 (UT
24afd4d8-7063-415a-8f51-73011b7fe402 Failed Feb 25, 2024, 23:55:19 (UTC+09:00) Feb 25, 2024, 23:55:22 (UT
a832d94b-4324-4766-9aa7-e4d9e6d4c707 Failed Feb 25, 2024, 23:50:05 (UTC+09:00) Feb 25, 2024, 23:50:10 (UT

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 Learn more
19

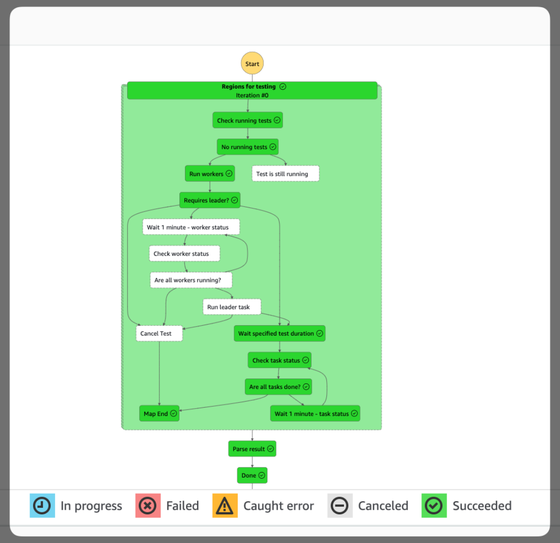
X-Ray trace map Learn more
1-65db9f84-30d2835203f0527724b450ef

Execution Logs Learn more
CloudWatch Logs

Graph view | Table view

Graph view

Actions



us-east-1.console.aws.amazon.com/states/home?region=us-east-1#/v2/executions/details/arn:aws:states:us-east-1:145607721954:execution:DLTStepFunctionTaskRunne

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

Step Functions > State machines > DLTStepFunctionTaskRunnerStepFunctionsC295A535-f0Kp2DT7zpny > Execution: f4fa9b84-a9f9-4acb-88

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 Learn more
19

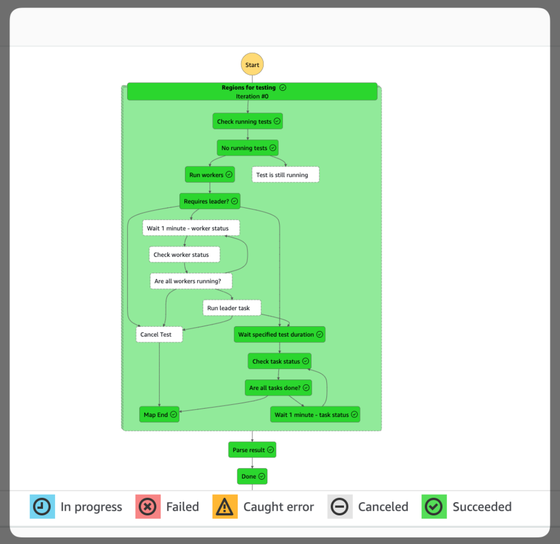
X-Ray trace map Learn more
1-65db9f84-30d2835203f0527724b450ef

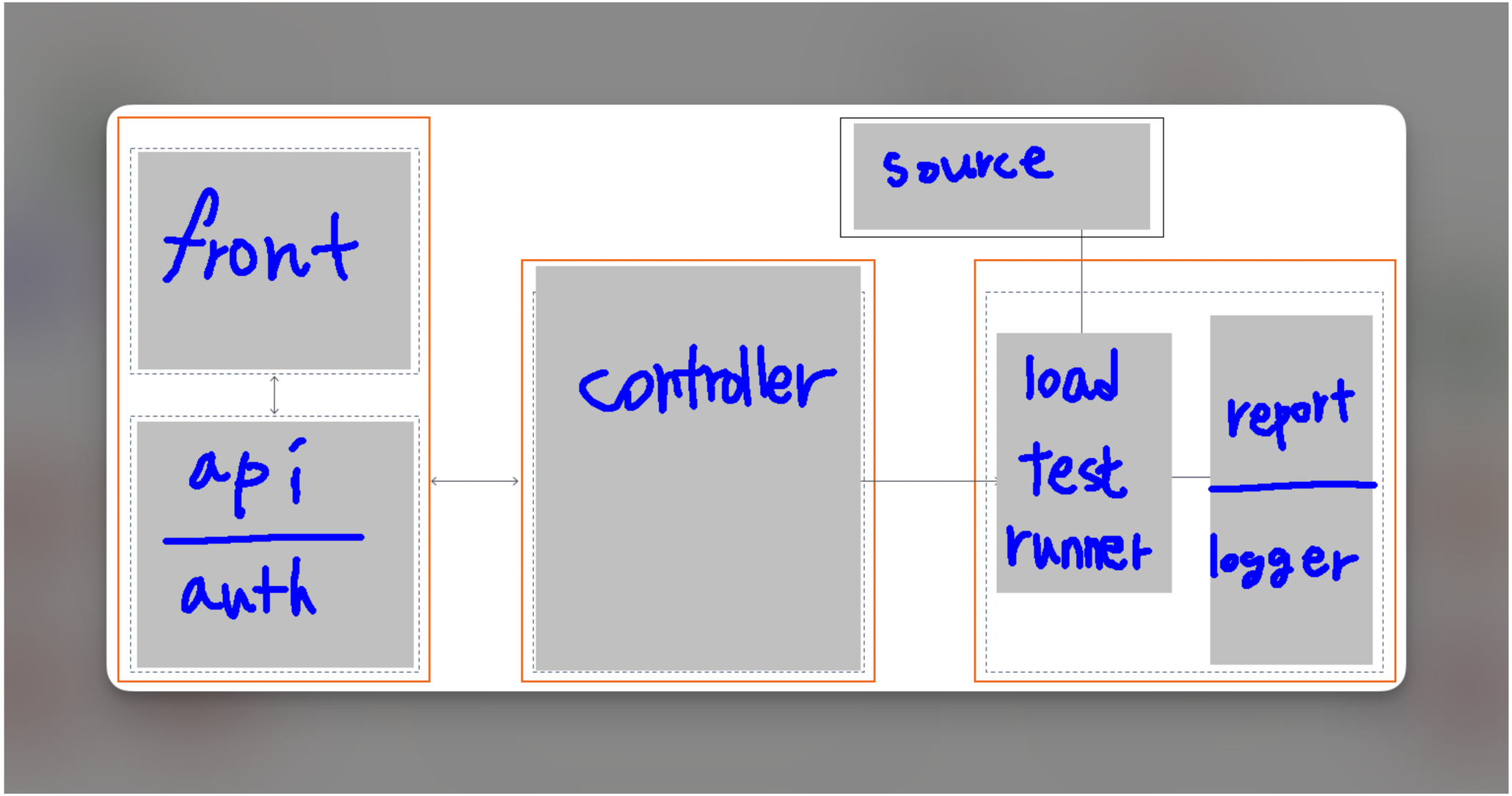
Execution Logs Learn more
CloudWatch Logs

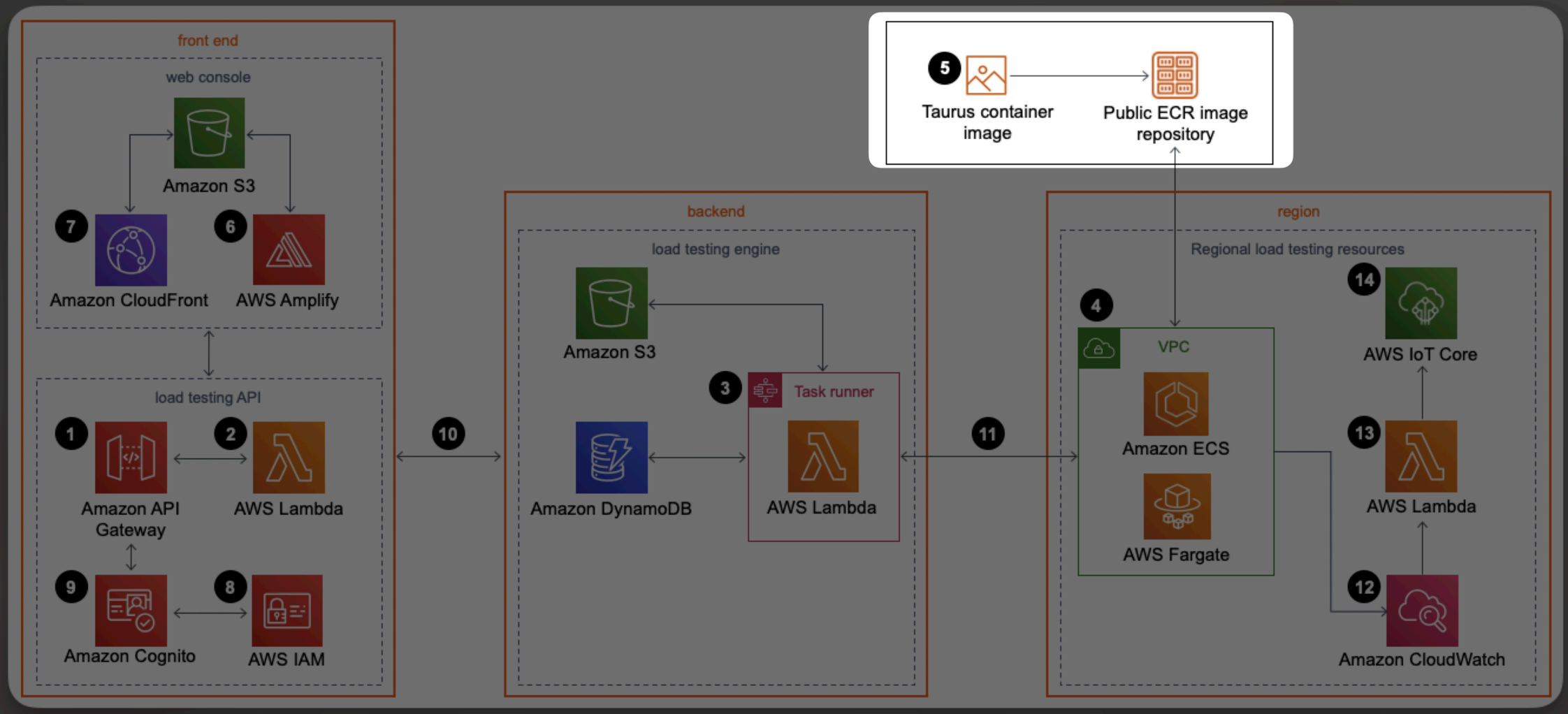
Graph view | Table view

Graph view

Actions




front
api
auth
controller
source
load
test
runner
report
logger


front end
web console
Amazon S3
7
Amazon CloudFront
6
AWS Amplify
load testing API
1
Amazon API Gateway
2
AWS Lambda
9
Amazon Cognito
8
AWS IAM
10
backend
load testing engine
Amazon S3
3
Task runner
Amazon DynamoDB
AWS Lambda
11
5
Taurus container image
Public ECR image repository
region
Regional load testing resources
4
VPC
Amazon ECS
AWS Fargate
14
AWS IoT Core
13
AWS Lambda
12
Amazon CloudWatch

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.

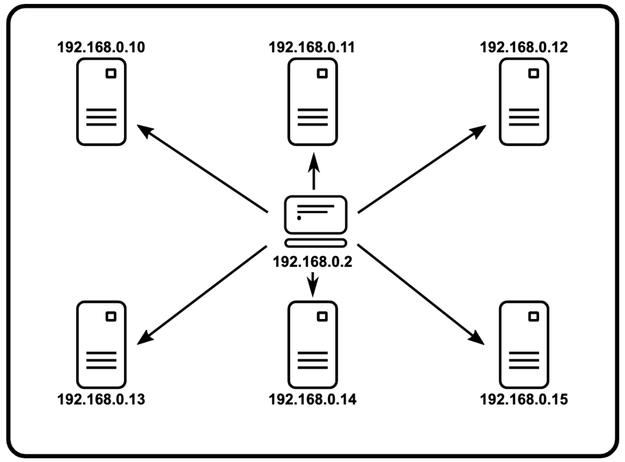
25. Apache JMeter Distributed Testing Step-by-step

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

25. Apache JMeter Distributed Testing Step-by-step

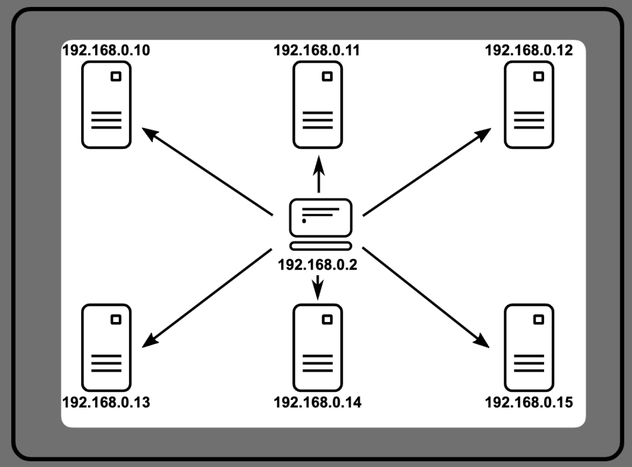
This short tutorial explains how to use multiple systems to perform stress testing. Before we start, there are a couple of things to check.

- 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.

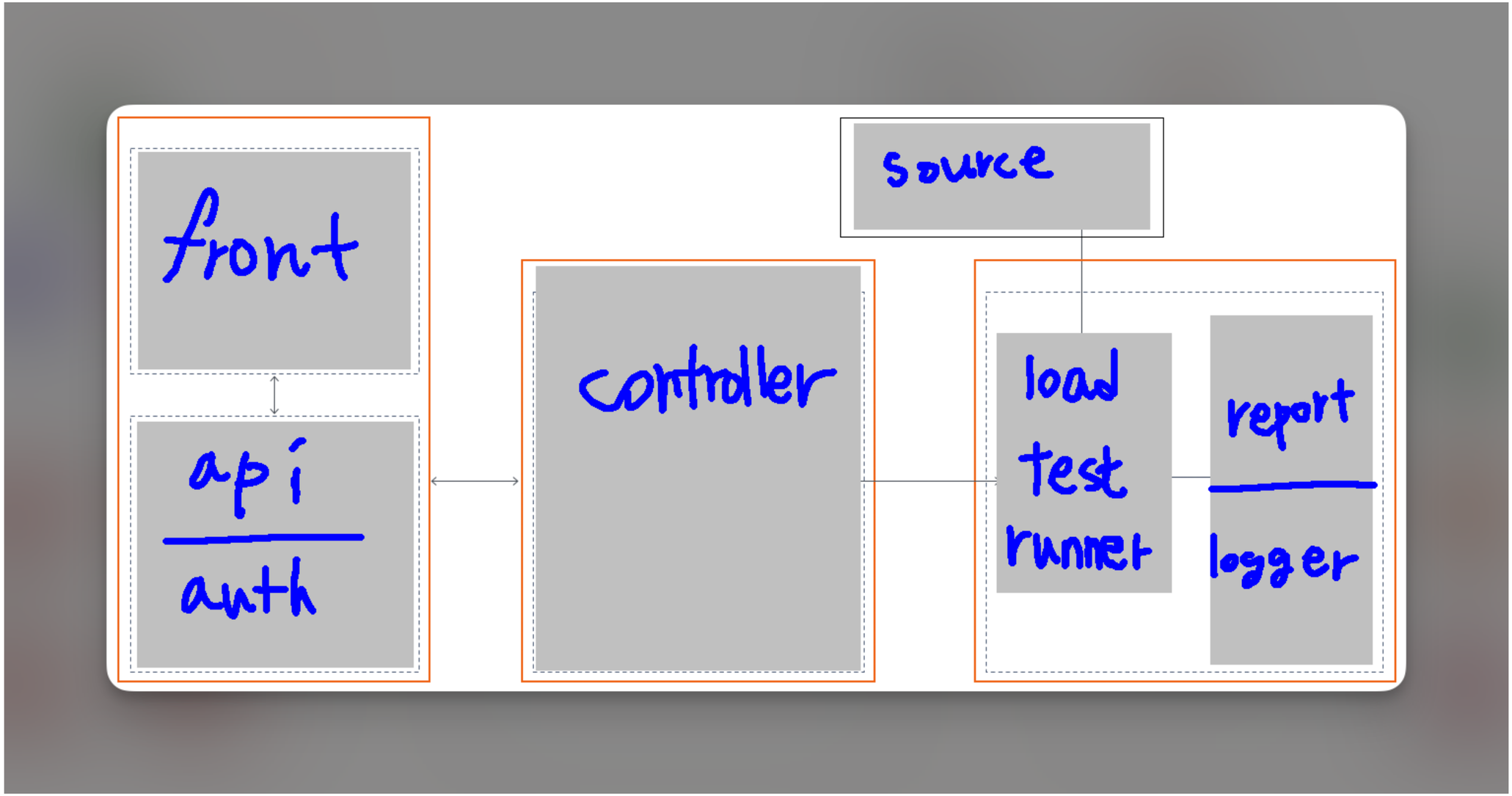
Once you've made sure the systems are ready, it's time to setup remote testing. The tutorial assumes you already have JMeter installed on all the systems. The way JMeter works is one controller node initiates the test on multiple worker nodes.

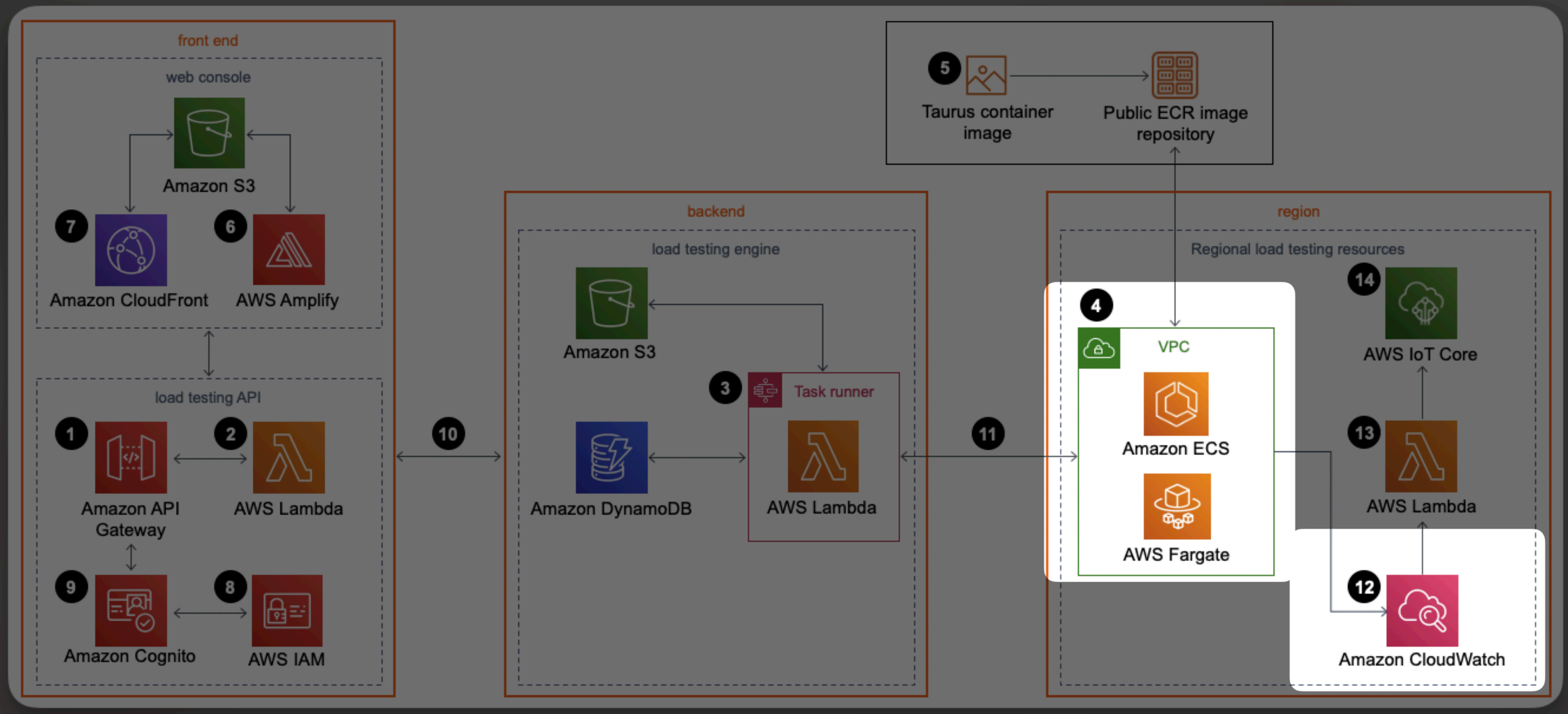
In this tutorial we use GUI Mode just for demonstration. In real life you should use CLI mode (NON GUI) to start your load test

taurus: 😉



One controller node with multiple worker nodes


front
api
auth
controller
source
load
test
runner
report
logger


front end
web console
Amazon S3
7
Amazon CloudFront
6
AWS Amplify
load testing API
1
Amazon API Gateway
2
AWS Lambda
9
Amazon Cognito
8
AWS IAM
10
backend
load testing engine
Amazon S3
3
Task runner
Amazon DynamoDB
AWS Lambda
11
5
Taurus container image
Public ECR image repository
region
Regional load testing resources
4
VPC
Amazon ECS
AWS Fargate
14
AWS IoT Core
13
AWS Lambda
12
Amazon CloudWatch

Tell us what you think

Amazon Elastic Container Service

Clusters
Namespaces
Task definitions
Account settings

Install AWS Copilot

Amazon ECR
Repositories

AWS Batch

Documentation
Discover products
Subscriptions

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


```
{
    "taskDefinitionArn": [redacted]task-definition/roeniss-dlt-test-DLTEcsDLTTaskDefinition6BFC2400-19v52JP14plb:1",
    "containerDefinitions": [
        {
            "name": "roeniss-dlt-test-load-tester",
            "image": "public.ecr.aws/aws-solutions/distributed-load-testing-on-aws-load-tester:v3.2.5",
            "cpu": 0,
            "memory": 4096,
            "links": [],
            "portMappings": [],
            "essential": true,
            "entryPoint": [],
            "command": [],
            "environment": [],
            "environmentFiles": [],
            "mountPoints": [],
            "volumesFrom": [],
            "secrets": [],
            "dnsServers": [],
            "dnsSearchDomains": [],
            "extraHosts": [],
            "dockerSecurityOptions": [],
            "dockerLabels": {},
            "ulimits": [],
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                    "awslogs-group": "roeniss-dlt-test-DLTEcsDLTCloudWatchLogsGroupFE9EC144-l0zKgM6cvhTV",
                    "awslogs-region": "us-east-1",
                    "awslogs-stream-prefix": "load-testing"
                },
```

Tell us what you think

Amazon Elastic Container Service

Clusters
Namespaces
Task definitions
Account settings

Install AWS Copilot

Amazon ECR
Repositories

AWS Batch

Documentation
Discover products
Subscriptions

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


```
{
    "taskDefinitionArn": [redacted]task-definition/roeniss-dlt-test-DLTEcsDLTTaskDefini
    "containerDefinitions": [
        {
            "name": "roeniss-dlt-test-load-tester",
            "image": "public.ecr.aws/aws-solutions/distributed-load-testing-on-aws-load-tester:v3.2.5",
            "cpu": 0,
            "memory": 4096,
            "links": [],
            "portMappings": [],
            "essential": true,
            "entryPoint": [],
            "command": [],
            "environment": [],
            "environmentFiles": [],
            "mountPoints": [],
            "volumesFrom": [],
            "secrets": [],
            "dnsServers": [],
            "dnsSearchDomains": [],
            "extraHosts": [],
            "dockerSecurityOptions": [],
            "dockerLabels": {},
            "ulimits": [],
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                    "awslogs-group": "roeniss-dlt-test-DLTEcsDLTCloudWatchLogsGroupFE9EC144-l0zKgM6cvhTV",
                    "awslogs-region": "us-east-1",
                    "awslogs-stream-prefix": "load-testing"
                },
```


aws Services Search [Option+S]

Tell us what you think

Amazon Elastic Container Service

Clusters
Namespaces
Task definitions
Account settings

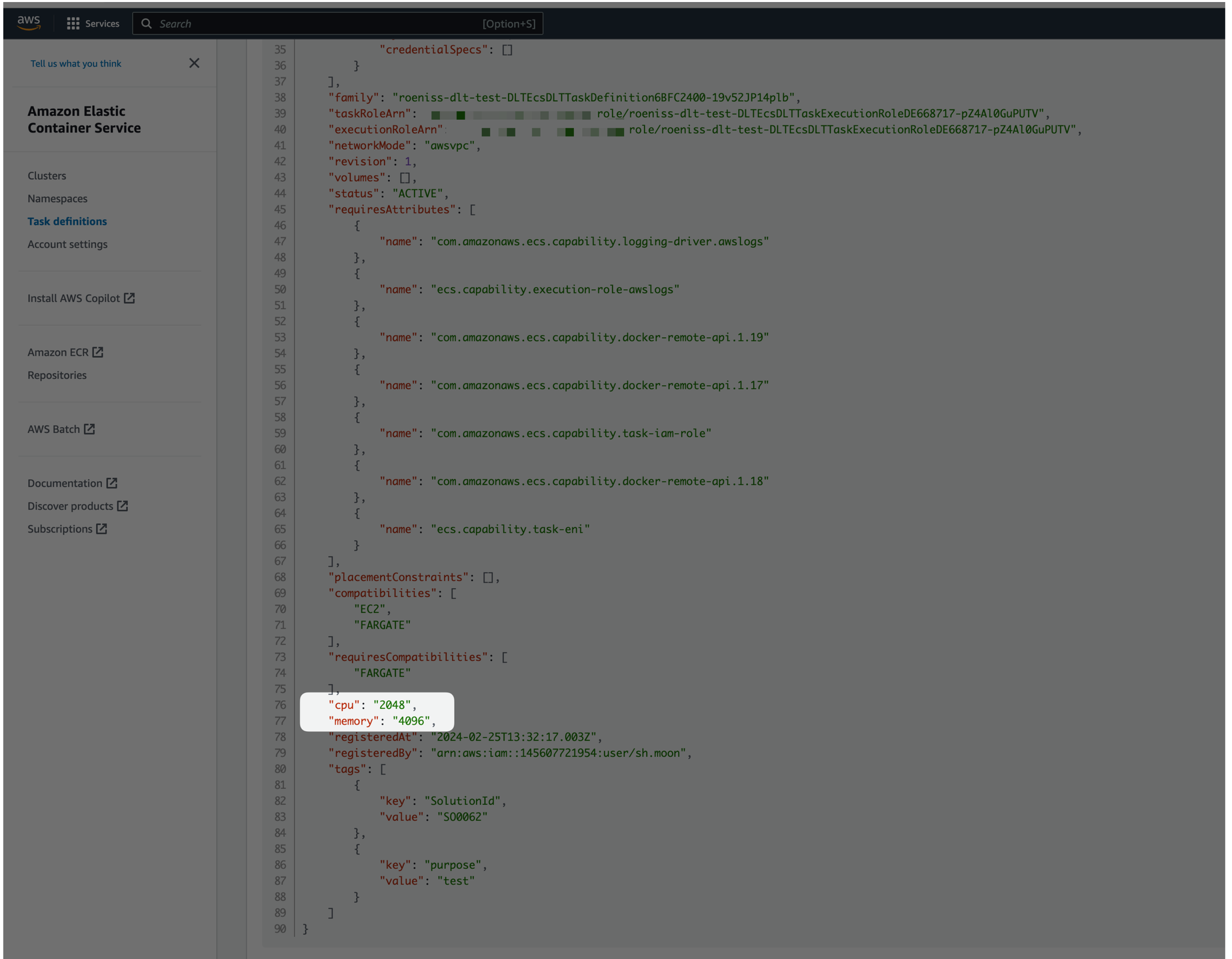
Install AWS Copilot

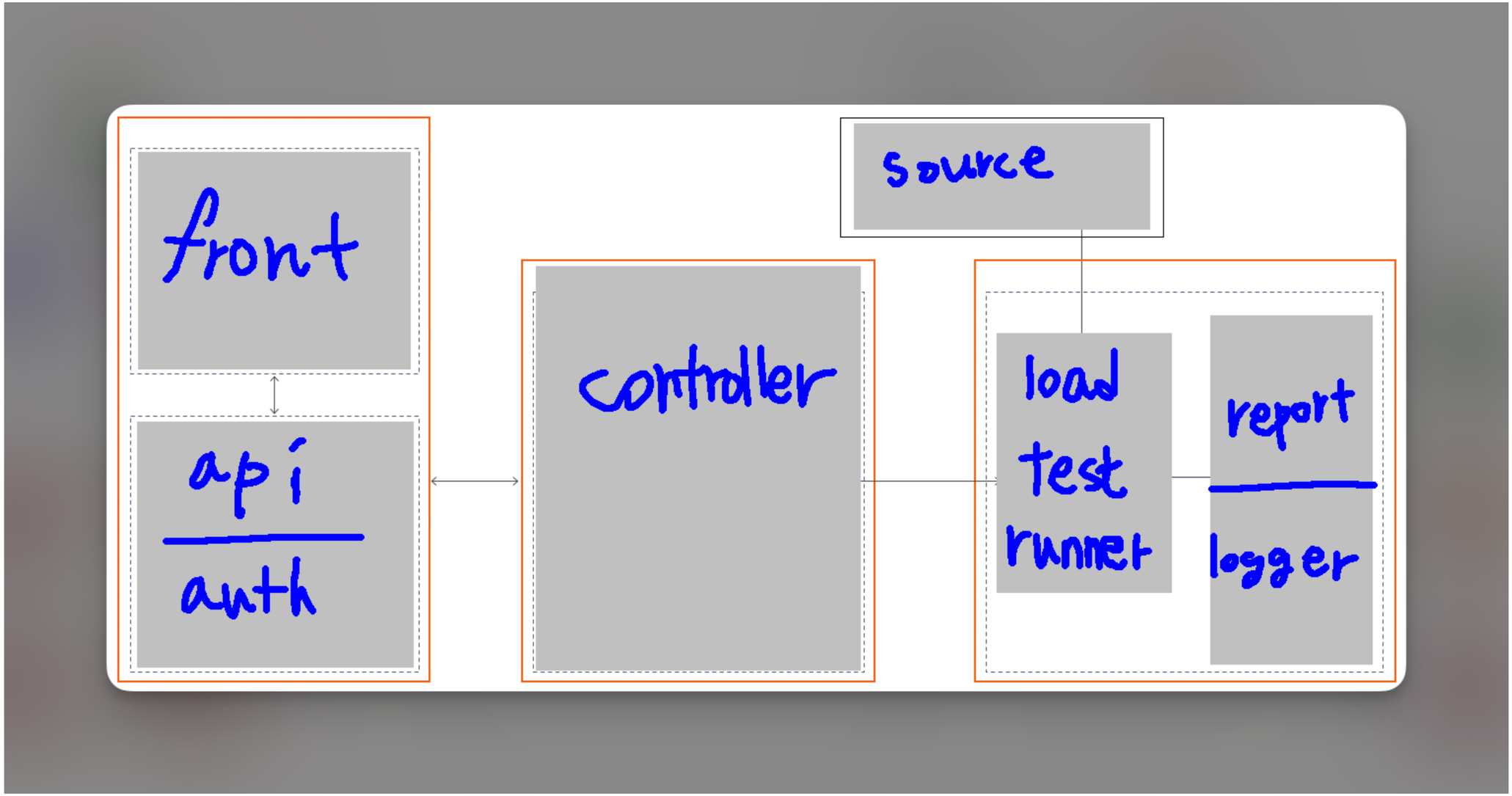
Amazon ECR
Repositories

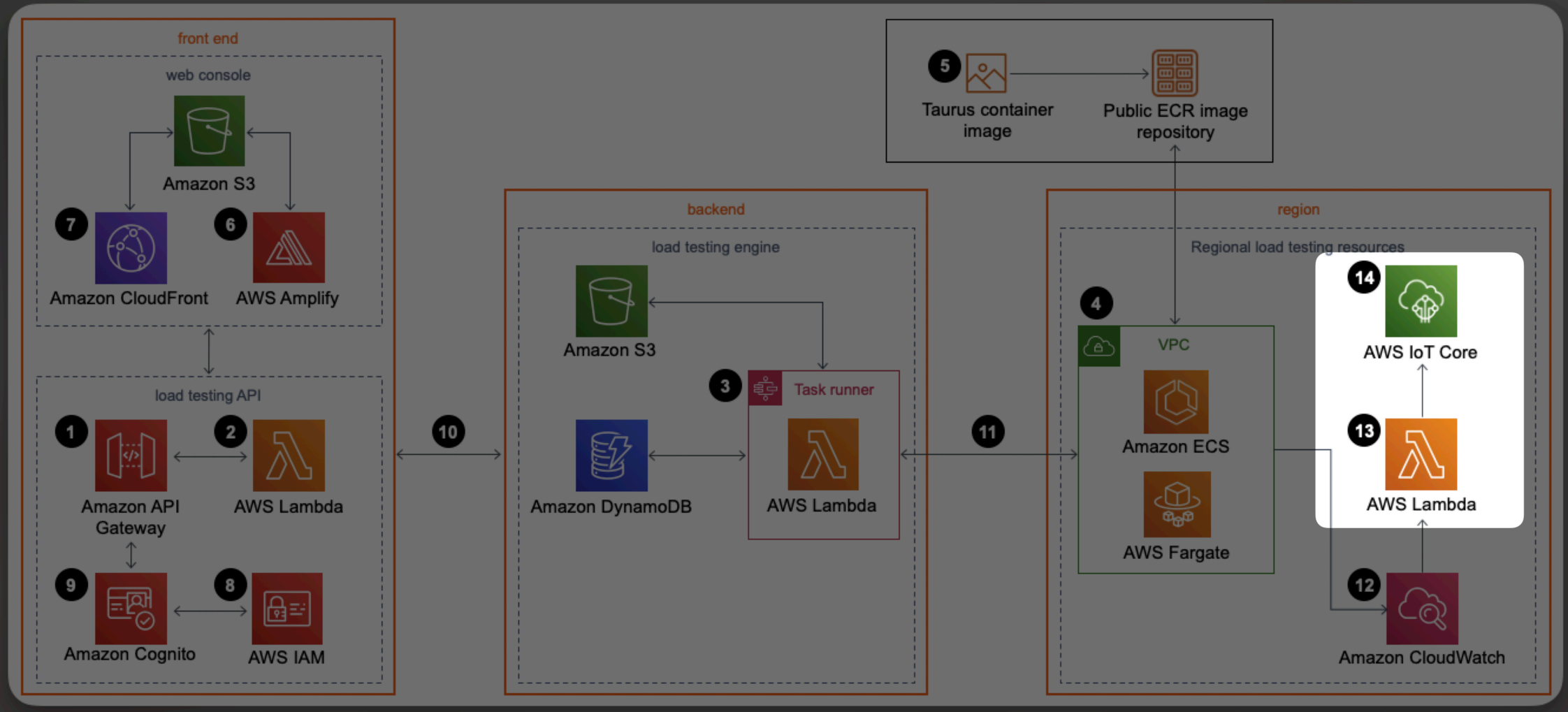
AWS Batch

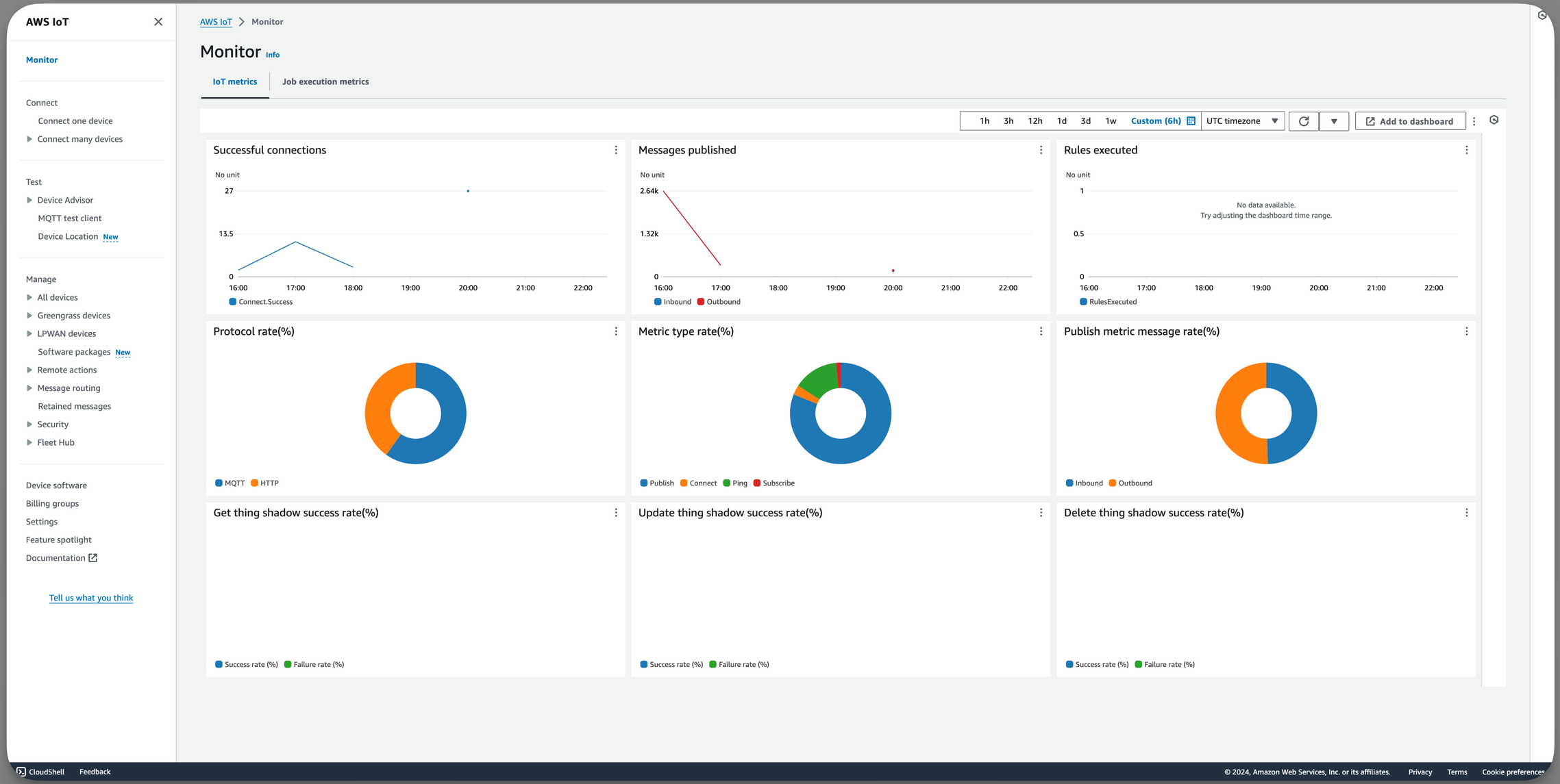
Documentation
Discover products
Subscriptions

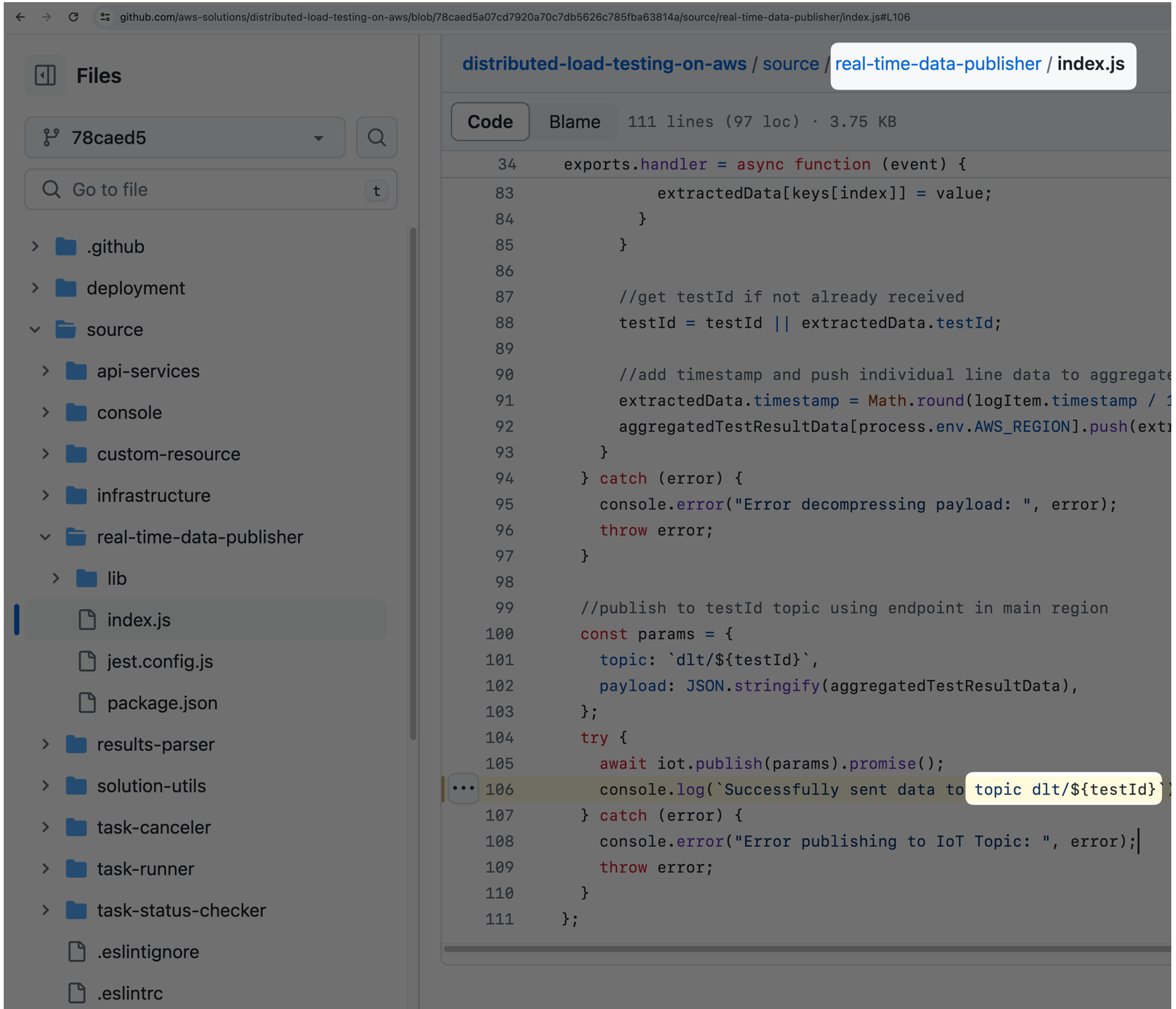

```
            "credentialSpecs": []
        }
    ],
    "family": "roeniss-dlt-test-DLTEcsDLTTaskDefinition6BFC240
    "taskRoleArn": [redacted]role/roeniss-dlt
    "executionRoleArn": [redacted]role/roenis
    "networkMode": "awsvpc",
    "revision": 1,
    "volumes": [],
    "status": "ACTIVE",
    "requiresAttributes": [
        {
            "name": "com.amazonaws.ecs.capability.logging-driv
        },
        {
            "name": "ecs.capability.execution-role-awslogs"
        },
        {
            "name": "com.amazonaws.ecs.capability.docker-remot
        },
        {
            "name": "com.amazonaws.ecs.capability.docker-remot
        },
        {
            "name": "com.amazonaws.ecs.capability.task-iam-rol
        },
        {
            "name": "com.amazonaws.ecs.capability.docker-remot
        },
        {
            "name": "ecs.capability.task-eni"
        }
    ],
    "placementConstraints": [],
    "compatibilities": [
        "EC2",
        "FARGATE"
    ],
    "requiresCompatibilities": [
        "FARGATE"
    ],
    "cpu": "2048",
    "memory": "4096",
    "registeredAt": "2024-02-25T13:32:17.003Z",
    "registeredBy": "arn:aws:iam::145607721954:user/sh.moon",
    "tags": [
        {
            "key": "SolutionId",
            "value": "SO0062"
        },
        {
            "key": "purpose",
            "value": "test"
        }
    ]
}
```

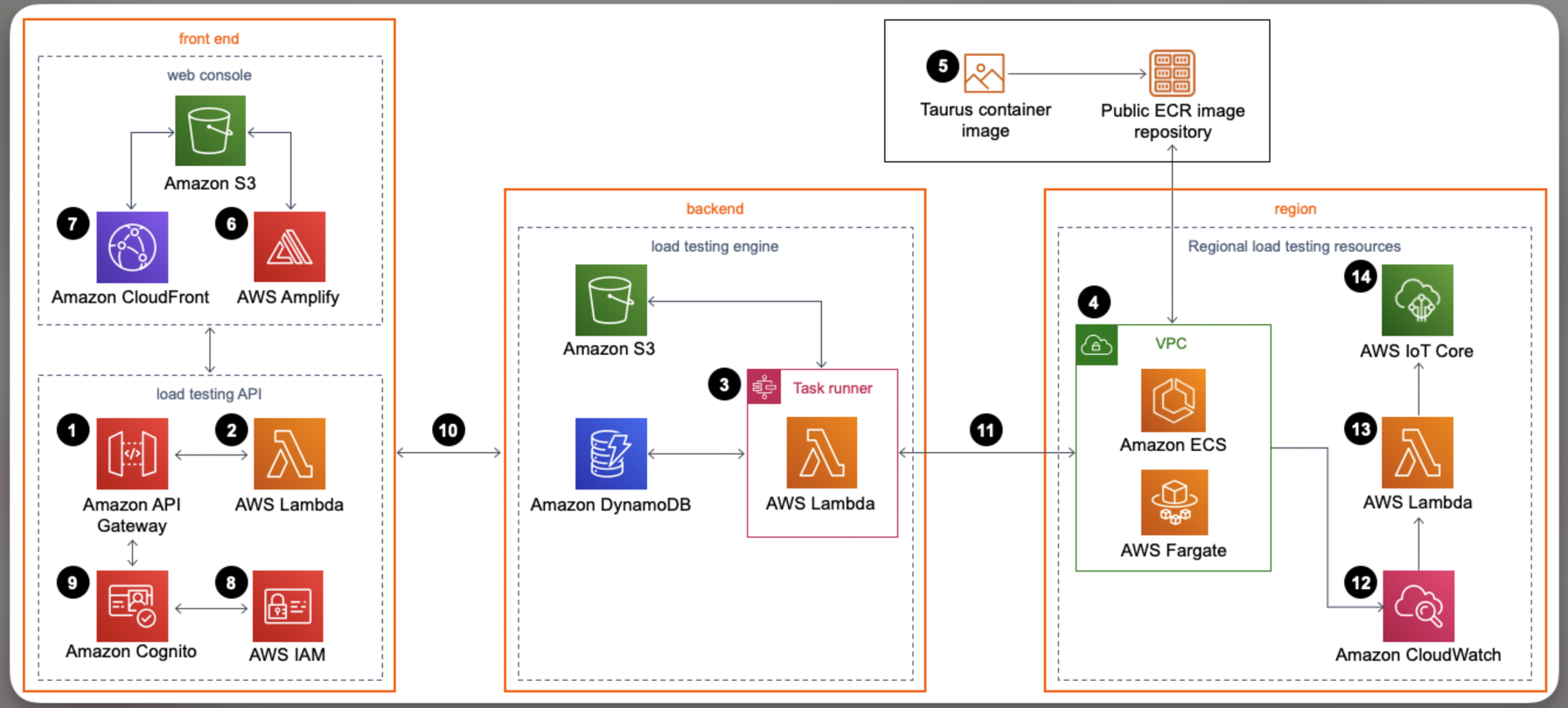

aws
Services
Search
[Option+S]
Tell us what you think
Amazon Elastic Container Service
Clusters
Namespaces
Task definitions
Account settings
Install AWS Copilot
Amazon ECR
Repositories
AWS Batch
Documentation
Discover products
Subscriptions
"credentialSpecs": []
}
],
"family": "roeniss-dlt-test-DLTEcsDLTTaskDefinition6BFC2400-19v52JP14plb",
"taskRoleArn": role/roeniss-dlt-test-DLTEcsDLTTaskExecutionRoleDE668717-pZ4Al0GuPUTV",
"executionRoleArn" role/roeniss-dlt-test-DLTEcsDLTTaskExecutionRoleDE668717-pZ4Al0GuPUTV",
"networkMode": "awsvpc",
"revision": 1,
"volumes": [],
"status": "ACTIVE",
"requiresAttributes": [
{
"name": "com.amazonaws.ecs.capability.logging-driver.awslogs"
},
{
"name": "ecs.capability.execution-role-awslogs"
},
{
"name": "com.amazonaws.ecs.capability.docker-remote-api.1.19"
},
{
"name": "com.amazonaws.ecs.capability.docker-remote-api.1.17"
},
{
"name": "com.amazonaws.ecs.capability.task-iam-role"
},
{
"name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
},
{
"name": "ecs.capability.task-eni"
}
],
"placementConstraints": [],
"compatibilities": [
"EC2",
"FARGATE"
],
"requiresCompatibilities": [
"FARGATE"
],
"cpu": "2048",
"memory": "4096",
"registeredAt": "2024-02-25T13:32:17.003Z",
"registeredBy": "arn:aws:iam::145607721954:user/sh.moon",
"tags": [
{
"key": "SolutionId",
"value": "SO0062"
},
{
"key": "purpose",
"value": "test"
}
]
}

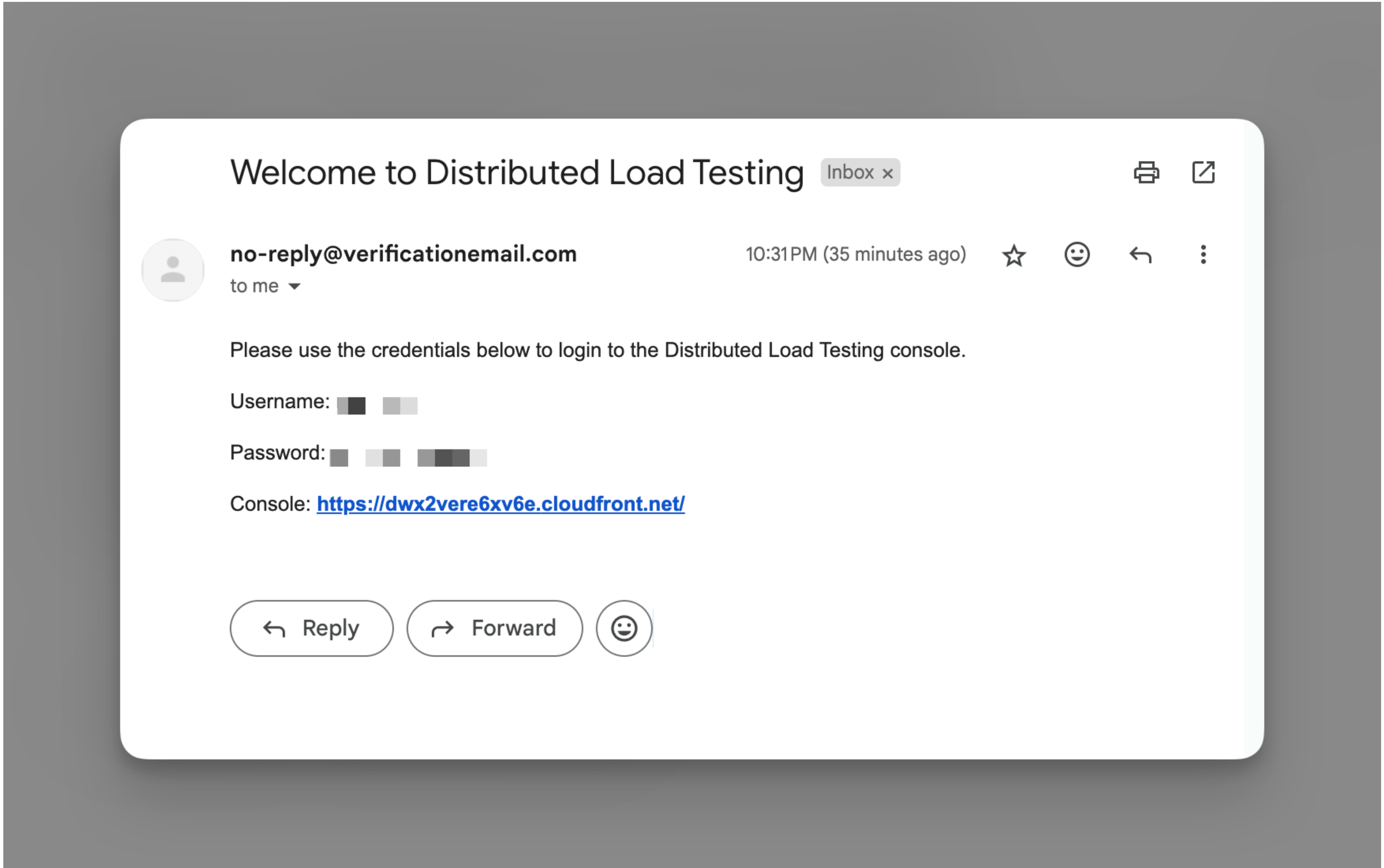

front
api
auth
controller
source
load
test
runner
report
logger


front end
web console
Amazon S3
7
Amazon CloudFront
6
AWS Amplify
load testing API
1
Amazon API Gateway
2
AWS Lambda
9
Amazon Cognito
8
AWS IAM
10
backend
load testing engine
Amazon S3
3
Task runner
Amazon DynamoDB
AWS Lambda
11
5
Taurus container image
Public ECR image repository
region
Regional load testing resources
4
VPC
Amazon ECS
AWS Fargate
14
AWS IoT Core
13
AWS Lambda
12
Amazon CloudWatch


AWS IoT
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
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
0.5
0
No data available.
Try adjusting the dashboard time range.
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 (%)
CloudShell Feedback
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences


github.com/aws-solutions/distributed-load-testing-on-aws/blob/78caed5a07cd7920a70c7db5626c785fba63814a/source/real-time-data-publisher/index.js#L106
Files
78caed5
Go to file
t
.github
deployment
source
api-services
console
custom-resource
infrastructure
real-time-data-publisher
lib
index.js
jest.config.js
package.json
results-parser
solution-utils
task-canceler
task-runner
task-status-checker
.eslintignore
.eslintrc
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) {
83 extractedData[keys[index]] = value;
84 }
85 }
86
87 //get testId if not already received
88 testId = testId || extractedData.testId;
89
90 //add timestamp and push individual line data to aggregate
91 extractedData.timestamp = Math.round(logItem.timestamp / 1
92 aggregatedTestResultData[process.env.AWS_REGION].push(extr
93 }
94 } catch (error) {
95 console.error("Error decompressing payload: ", error);
96 throw error;
97 }
98
99 //publish to testId topic using endpoint in main region
100 const params = {
101 topic: `dlt/${testId}`,
102 payload: JSON.stringify(aggregatedTestResultData),
103 };
104 try {
105 await iot.publish(params).promise();
106 console.log(`Successfully sent data to topic dlt/${testId}`)
107 } catch (error) {
108 console.error("Error publishing to IoT Topic: ", error);
109 throw error;
110 }
111 };

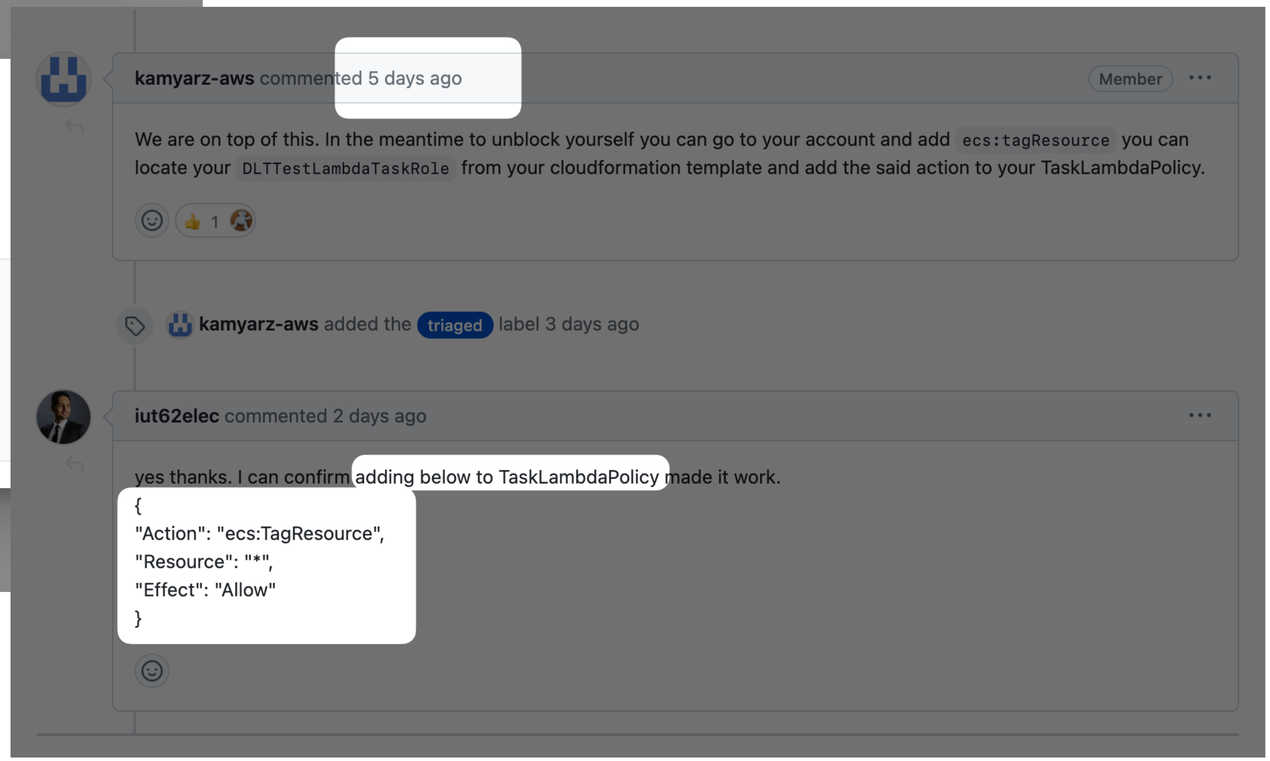

front end
web console
Amazon S3
7
Amazon CloudFront
6
AWS Amplify
load testing API
1
Amazon API Gateway
2
AWS Lambda
9
Amazon Cognito
8
AWS IAM
10
backend
load testing engine
Amazon S3
3
Task runner
Amazon DynamoDB
AWS Lambda
11
5
Taurus container image
Public ECR image repository
region
Regional load testing resources
4
VPC
Amazon ECS
AWS Fargate
14
AWS IoT Core
13
AWS Lambda
12
Amazon CloudWatch


Welcome to Distributed Load Testing
Inbox ×
no-reply@verificationemail.com
10:31 PM (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

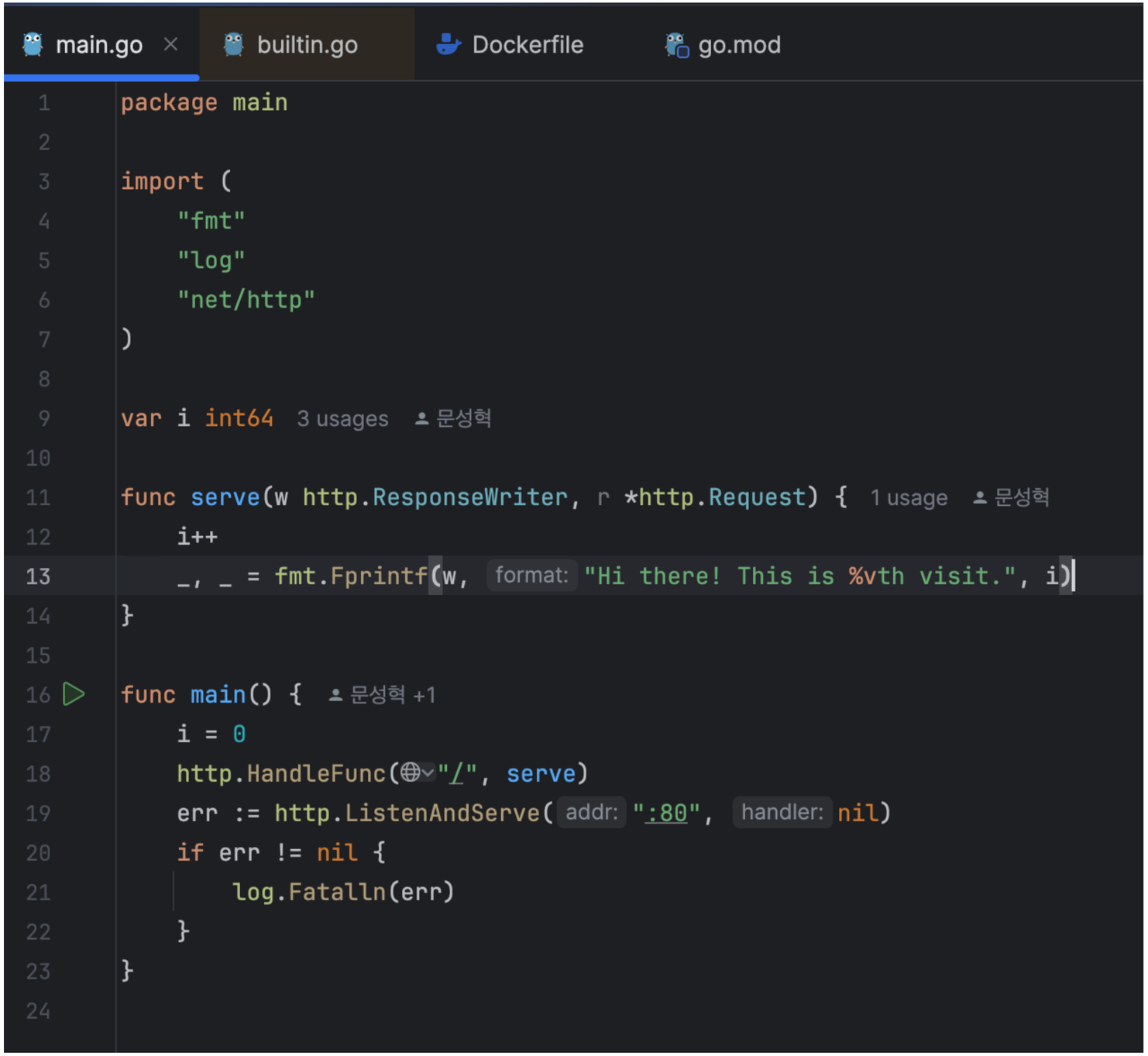
1. Download the AWS CloudFormation template to your local hard drive.
2. Open the AWS CloudFormation template with a text editor.
3. Modify the AWS CloudFormation template mapping section from:

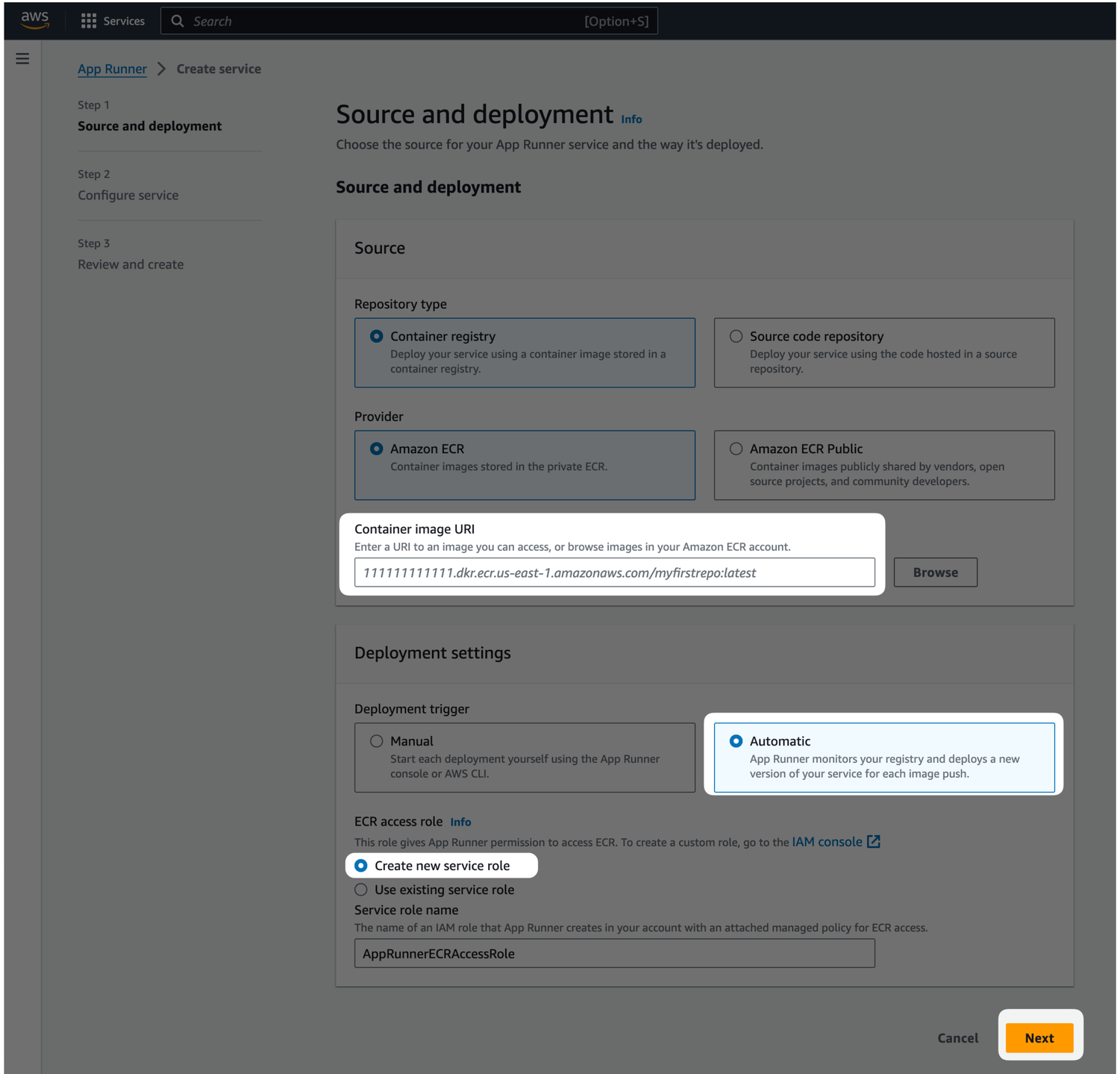
```
Solution:
  Config:
    SendAnonymousData: "Yes"
```

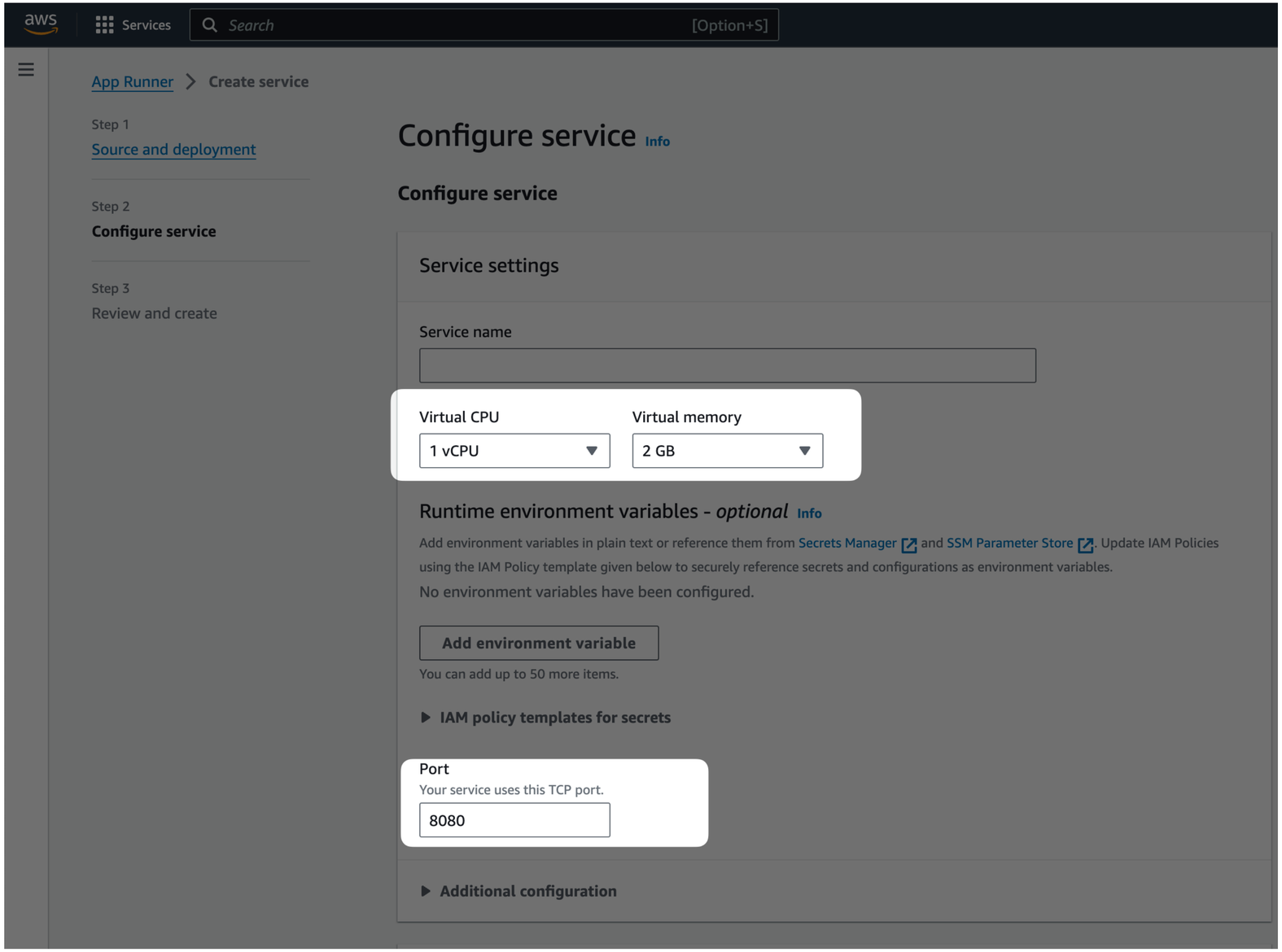


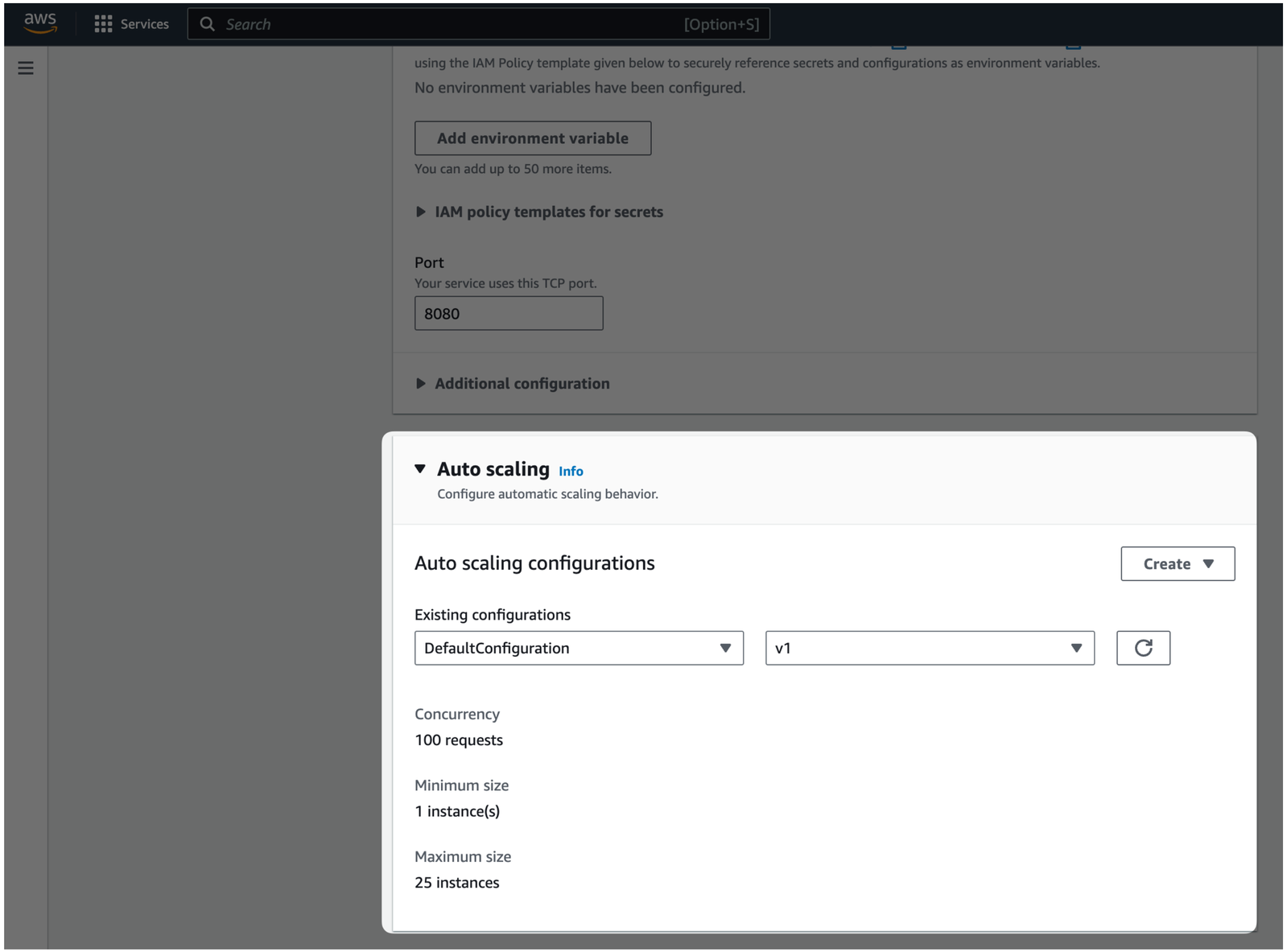
https://github.com/aws-solutions/distributed-load-testing-on-aws/issues/162

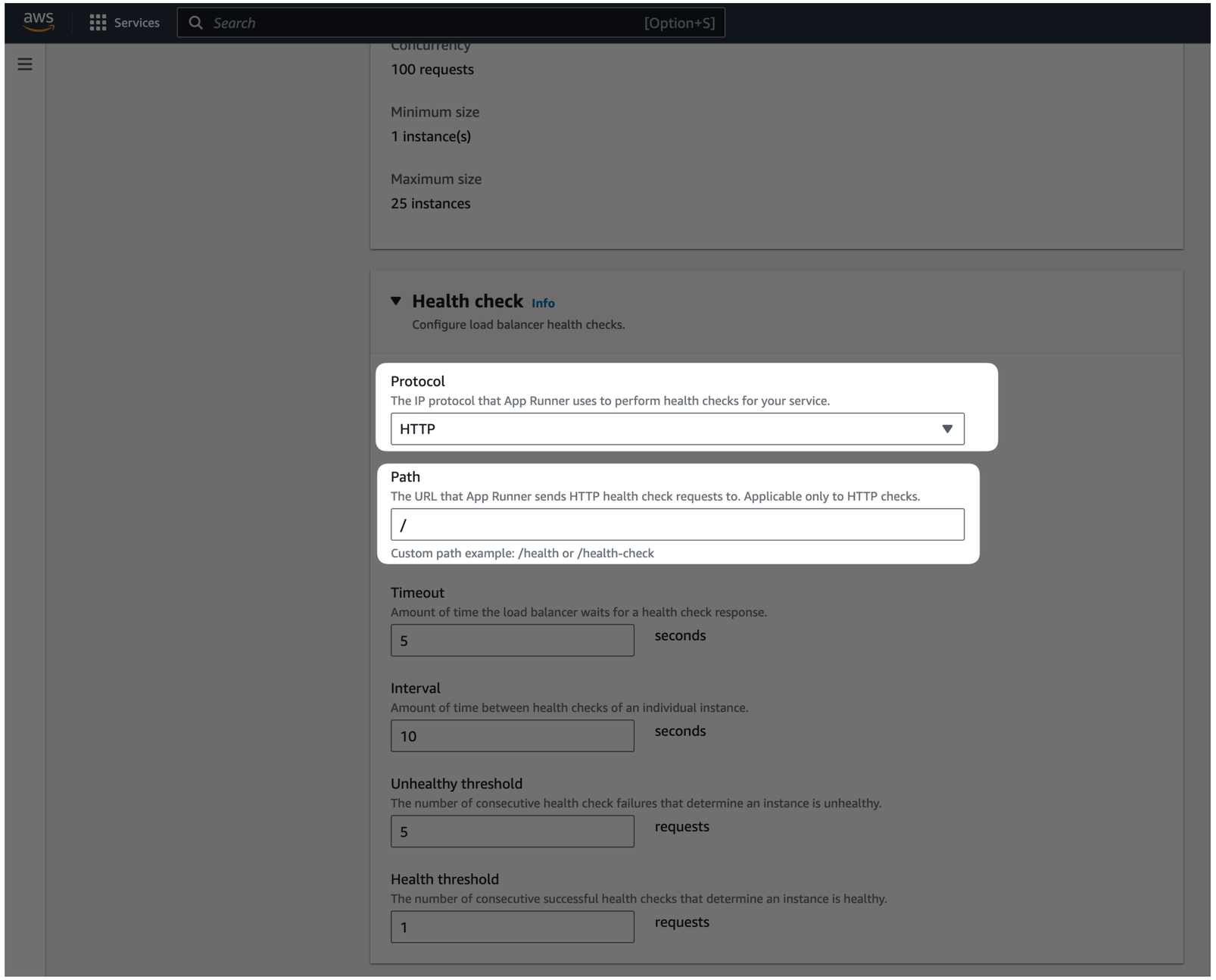
자투리1: 타겟 서버는 어디?


main.go
builtin.go
Dockerfile
go.mod
package main
import (
"fmt"
"log"
"net/http"
)
var i int64 3 usages 문성혁
func serve(w http.ResponseWriter, r *http.Request) { 1 usage 문성혁
i++
_, _ = fmt.Fprintf(w, format: "Hi there! This is %vth visit.", i)
}
func main() { 문성혁 +1
i = 0
http.HandleFunc("/", serve)
err := http.ListenAndServe(addr: ":80", handler: nil)
if err != nil {
log.Fatalln(err)
}
}


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
1 vCPU
Virtual memory
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.
8080
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
Concurrency
100 requests
Minimum size
1 instance(s)
Maximum size
25 instances


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.
HTTP
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.
5
seconds
Interval
Amount of time between health checks of an individual instance.
10
seconds
Unhealthy threshold
The number of consecutive health check failures that determine an instance is unhealthy.
5
requests
Health threshold
The number of consecutive successful health checks that determine an instance is healthy.
1
requests

ECS 대역폭 측정 벤치마크 (비공식)

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

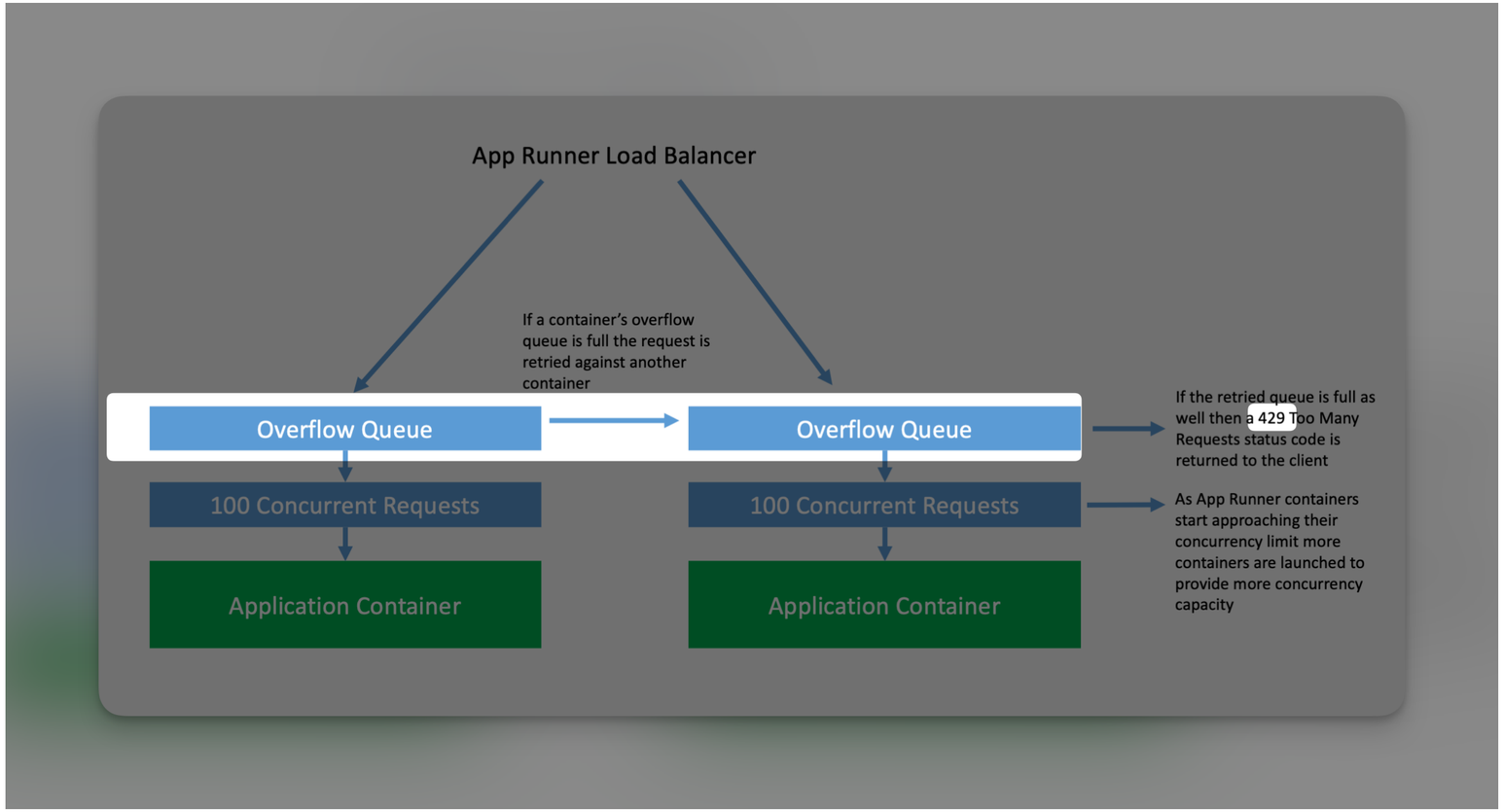
Results (Fargate)

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	-

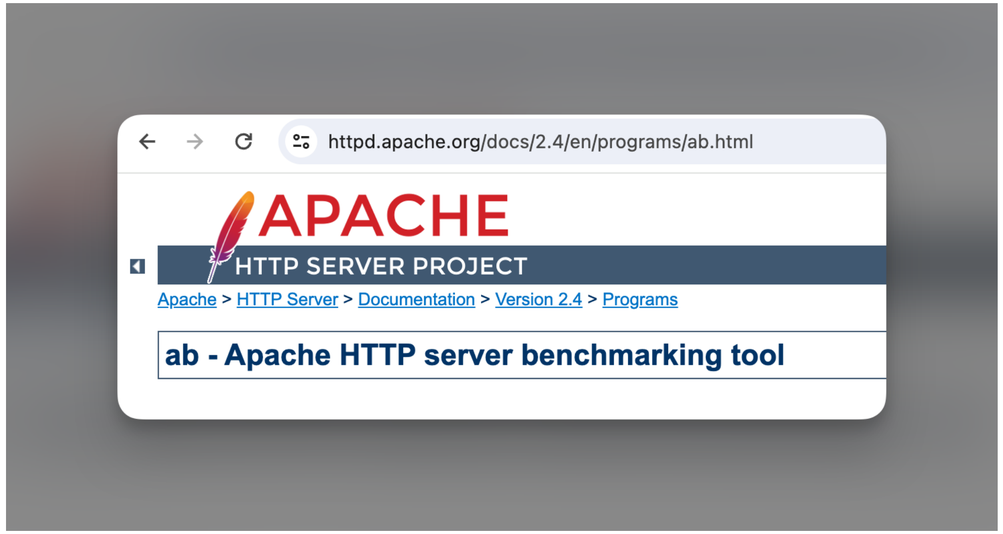
429 는 어떻게 나오는걸까?

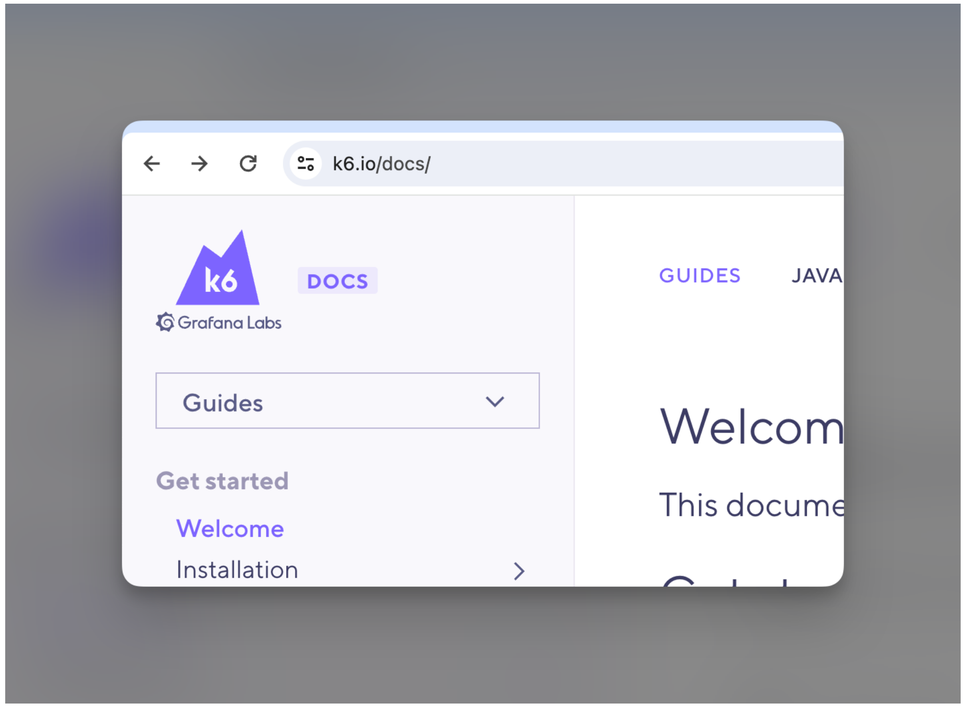


자투리2: 부하 테스트는 뭐부터?

완전 처음이라면

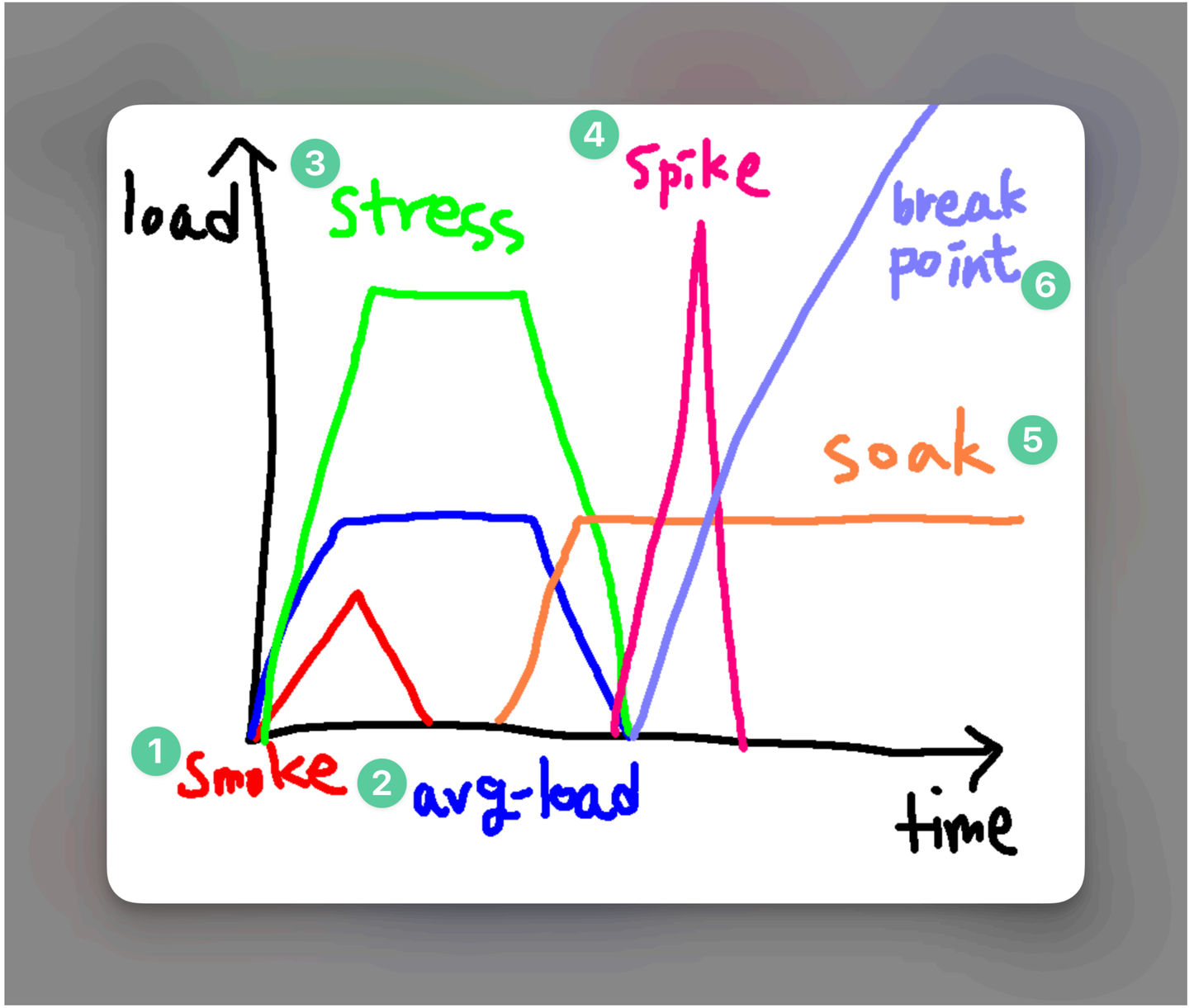
- ab, k6
- 명령어 하나, 파일 하나 정도로 구동





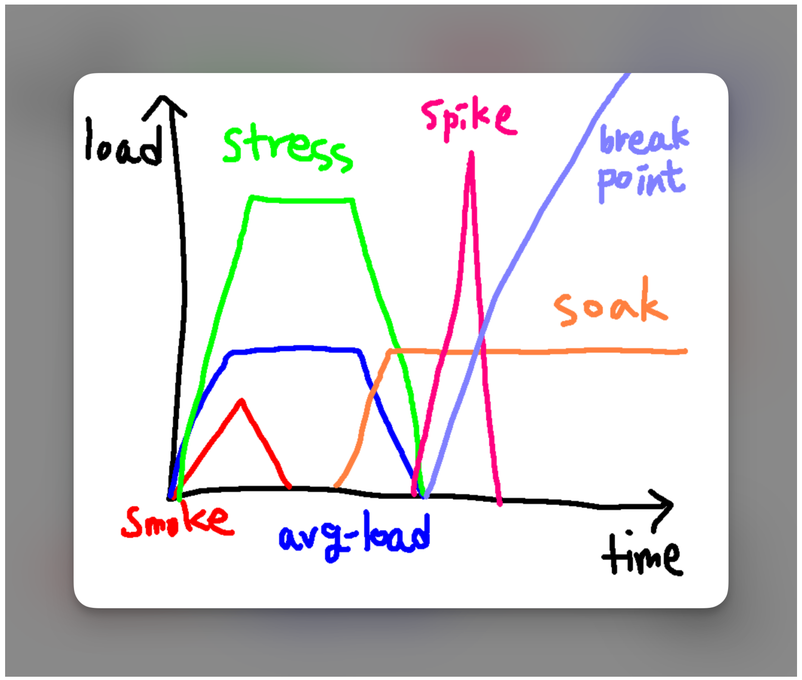
접근 순서는

접근 순서는


load
3
Stress
4
Spike
break
point
6
soak
5
1
Smoke
2
avg-load
time

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)