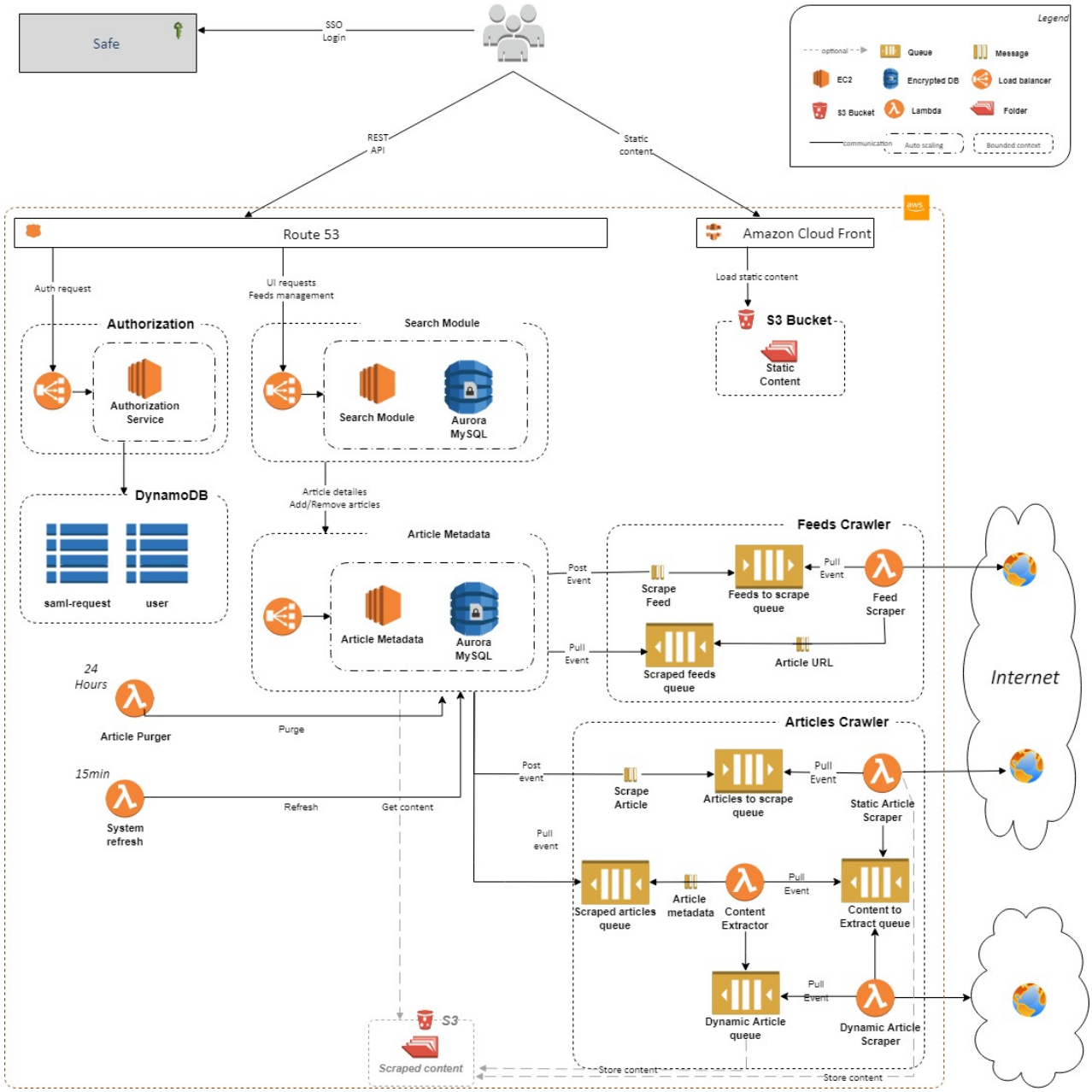
SM Architecture

Jenga

Exported on 11/22/2019

Table of Contents

1 High-level Architecture diagram



2 Architectural Overview recording

With Nick Maddock-Lyons from Architecture

https://thomsonreuters.webex.com/thomsonreuters/lsr.php?RCID=fd6793af8f774be9b94b88403219389e

3 Components description

Name	Search Module Service
Description	Supports Frontend mainly to get Articles for particular Search Modules and add/update/ delete Search Modules
Technology Stack	EC2, Java, Spring
Functional responsibilities	Support Frontend for: • Get Articles • Add Search Module • Update Search Module • Delete Search Module
Dependencies	Used by Frontend Depends on Article Metadata Service

Name	Article Metadata Service
Description	This is the main worker engine of Source Monitoring. It is responsible for polling RSS feeds, refreshing system state and updating Search modules views in DB
Technology Stack	EC2, Java, Spring
Functional responsibilities	• Poll data from RSS feeds • Prepare data for Search modules • Doing system refresh • Support refresh logic by tracking last_update timestamps, etc.
Dependencies	Used by Search Module

Name	Authorization Service
Description	Responsible of supporting login/auth flow for the users
Technology Stack	EC2, Java, Spring

Functional responsibilities	• Support user login through SAFE • Validate user identity based on SAML response • Generate, manage and validate JWT tokens
Dependencies	Used by Frontend Integrated with SAFE

Name	Feed Scraper Lambda
Description	Responsible purely for fetching/scrapping Feed Page by URL
Technology Stack	Phyton, Lambda
Functional responsibilities	Scrape Feeds by URL
Dependencies	It uses "Feed to scrape Queue" for pulling Events previously added by Article Metadata Service.

Name	Static Article Scraper Lambda
Description	Responsible purely for fetching/scrapping articles by URL
Technology Stack	Phyton, Lambda
Functional responsibilities	• Pull Events from “Article to Scrape Queue” • Send http request on Internet for scraping article by URL • Fetch actual Article text from **Static** HTML page • Write Events into “Content to Extract Queue”
Dependencies	It uses"Article to scrape Queue" for pulling Events previously added by Article Metadata Service.

Name	Content Extractor Lambda
Description	Responsible purely for extracting article content from web page source
Technology Stack	Phyton, Lambda
Functional responsibilities	Extract Content from a web page source. • Pull Events from “Content to Extract Queue” • Write Event in “Scraped Articles Queue” or in “Dynamic Article Queue”

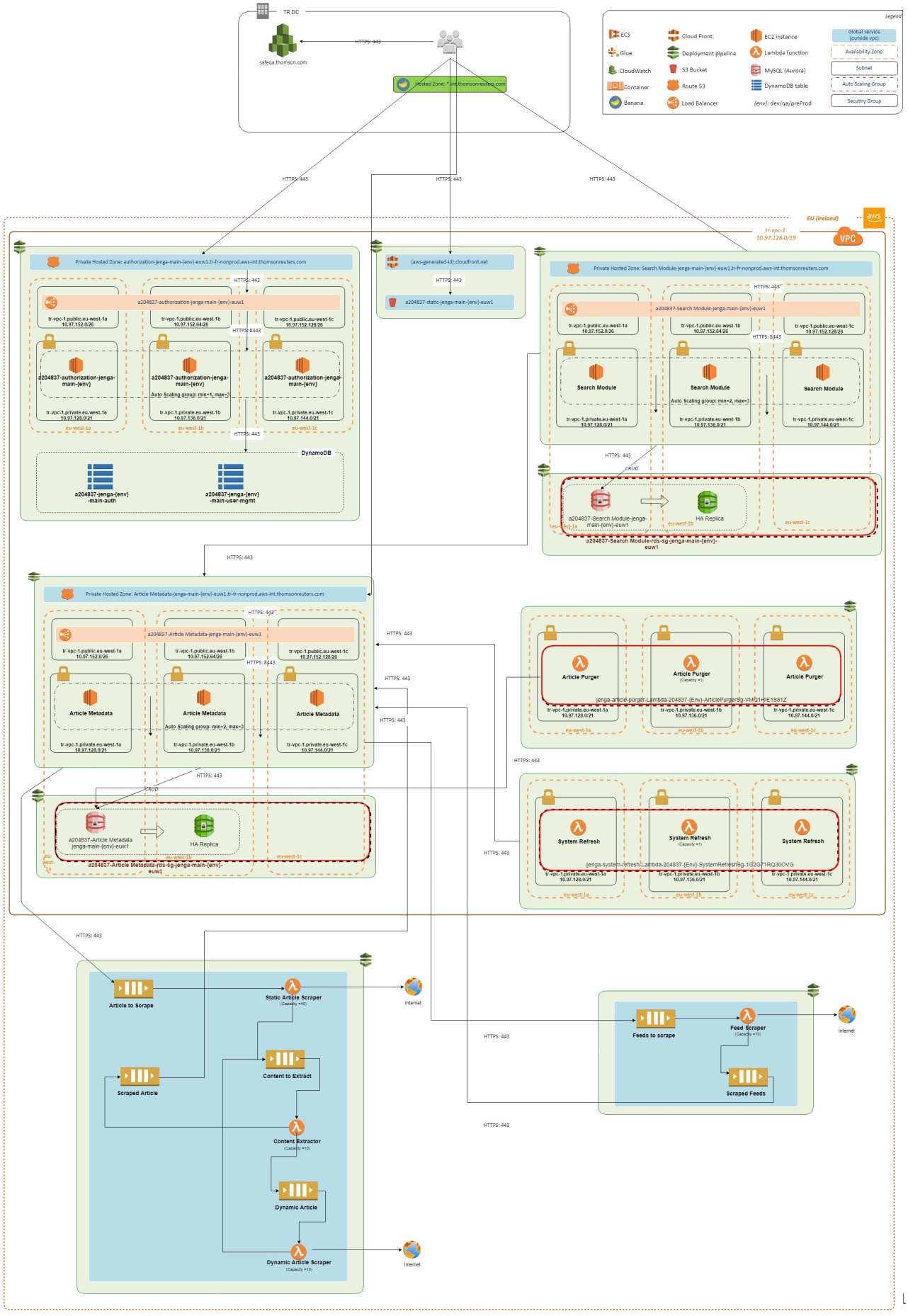
Dependencies	It uses “Content to Extract Queue" for pulling Events

Name	Dynamic Article Scraper Lambda
Description	Responsible purely for fetching/scrapping Dynamic Page by URL
Technology Stack	Phyton, Lambda
Functional responsibilities	• Pull Events from “Dynamic Article Queue” • Send http request on Internet for scraping article by URL • Fetch actual Article text from **Dynamic** HTML page • Write Events into “Content to Extract Queue”
Dependencies	It uses “Dynamic Article Queue” for pulling Events

Name	System Refresh Lambda
Description	Lambda function which is called by trigger to initiate system refresh once in 15 minutes
Technology Stack	Java, Lambda
Functional responsibilities	- Initiate system refresh by schedule
Dependencies	Calls Article Metadata Service

Name	Article Purger Lambda
Description	Responsible of purging articles that are 2 weeks old to be compliant with legal REQs
Technology Stack	Phyton, Lambda
Functional responsibilities	- Purging Articles from DB that are 2 weeks older
Dependencies	Directly connected to DB

4 SM Deployment view


TR DC
safeqa.thomson.com
HTTPS: 443
Hosted Zone: *.int.thomsonreuters.com
Legend
ECS
Glue
CloudWatch
Container
Banana
Cloud Front
Deployment pipeline
S3 Bucket
Route 53
Load Balancer
EC2 instance
Lambda function
MySQL (Aurora)
DynamoDB table
{env}: dev/qa/preProd
Global service (outside vpc)
Availability Zone
Subnet
Auto Scaling Group
Security Group
EU (Ireland)
tr-vpc-1
10.97.128.0/19
VPC
Private Hosted Zone: authorization-jenga-main-{env}-euw1.tr-fr-nonprod.aws-int.thomsonreuters.com
a204837-authorization-jenga-main-{env}-euw1
a204837-authorization-jenga-main-{env}
Auto Scaling group: min=1, max=3
DynamoDB
a204837-jenga-{env}-main-auth
a204837-jenga-{env}-main-user-mgmt
{aws-generated-id}.cloudfront.net
a204837-static-jenga-main-{env}-euw1
Private Hosted Zone: Search Module-jenga-main-{env}-euw1.tr-fr-nonprod.aws-int.thomsonreuters.com
a204837-Search Module-jenga-main-{env}-euw1
Search Module
Auto Scaling group: min=2, max=3
CRUD
a204837-Search Module-jenga-main-{env}-euw1
HA Replica
a204837-Search Module-rds-sg-jenga-main-{env}-euw1
Private Hosted Zone: Article Metadata-jenga-main-{env}-euw1.tr-fr-nonprod.aws-int.thomsonreuters.com
a204837-Article Metadata-jenga-main-{env}-euw1
Article Metadata
Auto Scaling group: min=2, max=3
a204837-Article Metadata-jenga-main-{env}-euw1
HA Replica
a204837-Article Metadata-rds-sg-jenga-main-{env}-euw1
Article Purger
jenga-article-purger-Lambda-204837-{Env}-ArticlePurgerSg-VM01HE1S81Z
System Refresh
(jenga-system-refresh-Lambda-204837-{Env}-SystemRefreshSg-1G2G71RQ300VG
Article to Scrape
Static Article Scraper
Scraped Article
Content to Extract
Content Extractor
Dynamic Article
Dynamic Article Scraper
Feeds to scrape
Feed Scraper
Scraped Feeds
Internet

5 SM Disaster recovery view

5.1 Requirements

Acceptable downtime in case of disaster: *8h.*

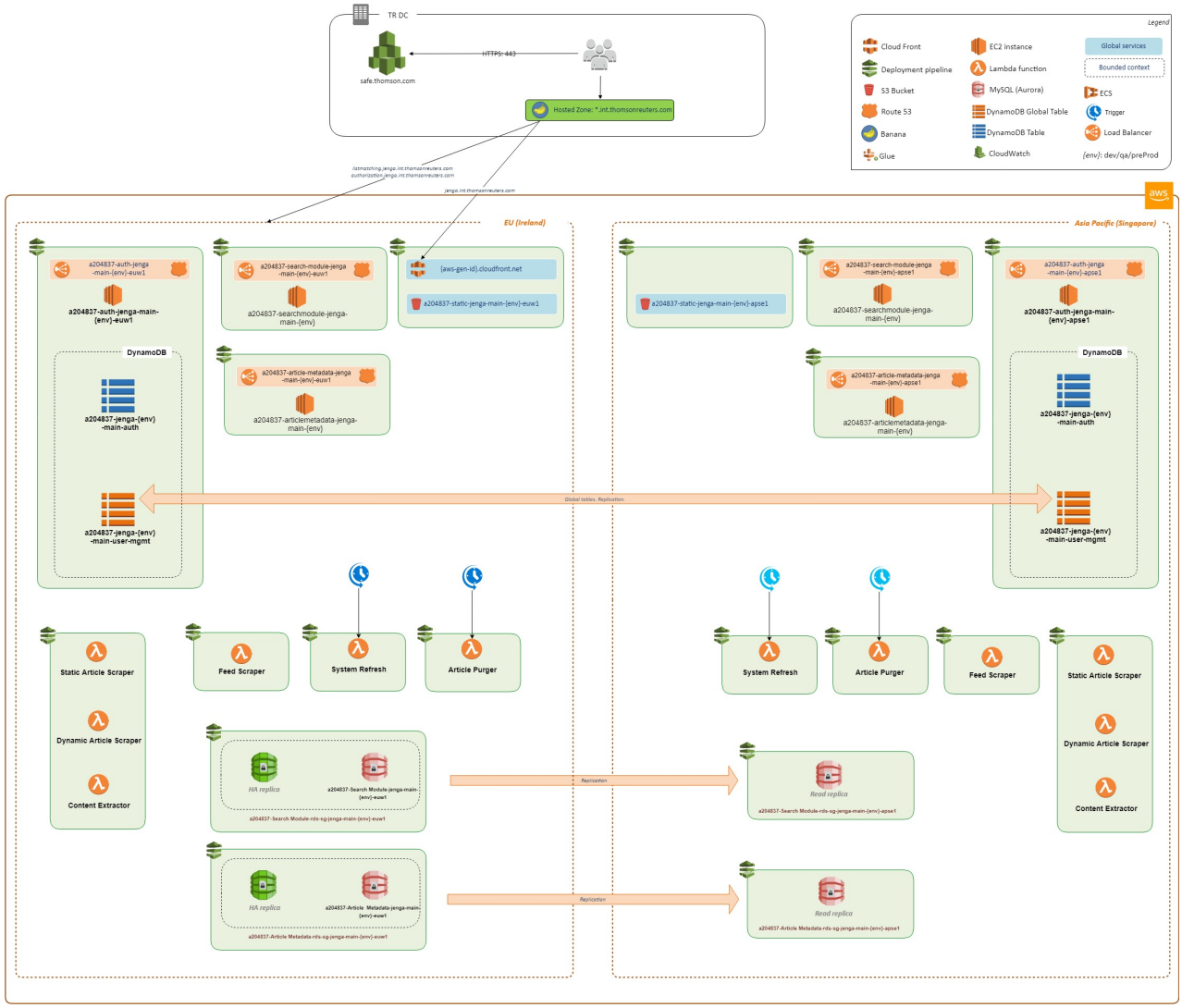
5.2 Recovery strategy

To satisfy non-functional requirements it was decided to use *warm standby* strategy. A DR scenario in which a scaled-down version of a fully functional environment is always running in the cloud. In a disaster, the system is scaled up quickly to handle the production load.

DOC-1104841. Warm Standby[1]

1 https://thehub.thomsonreuters.com/docs/DOC-1104841#jive_content_id_Warm_Standby

5.3 Disaster recovery



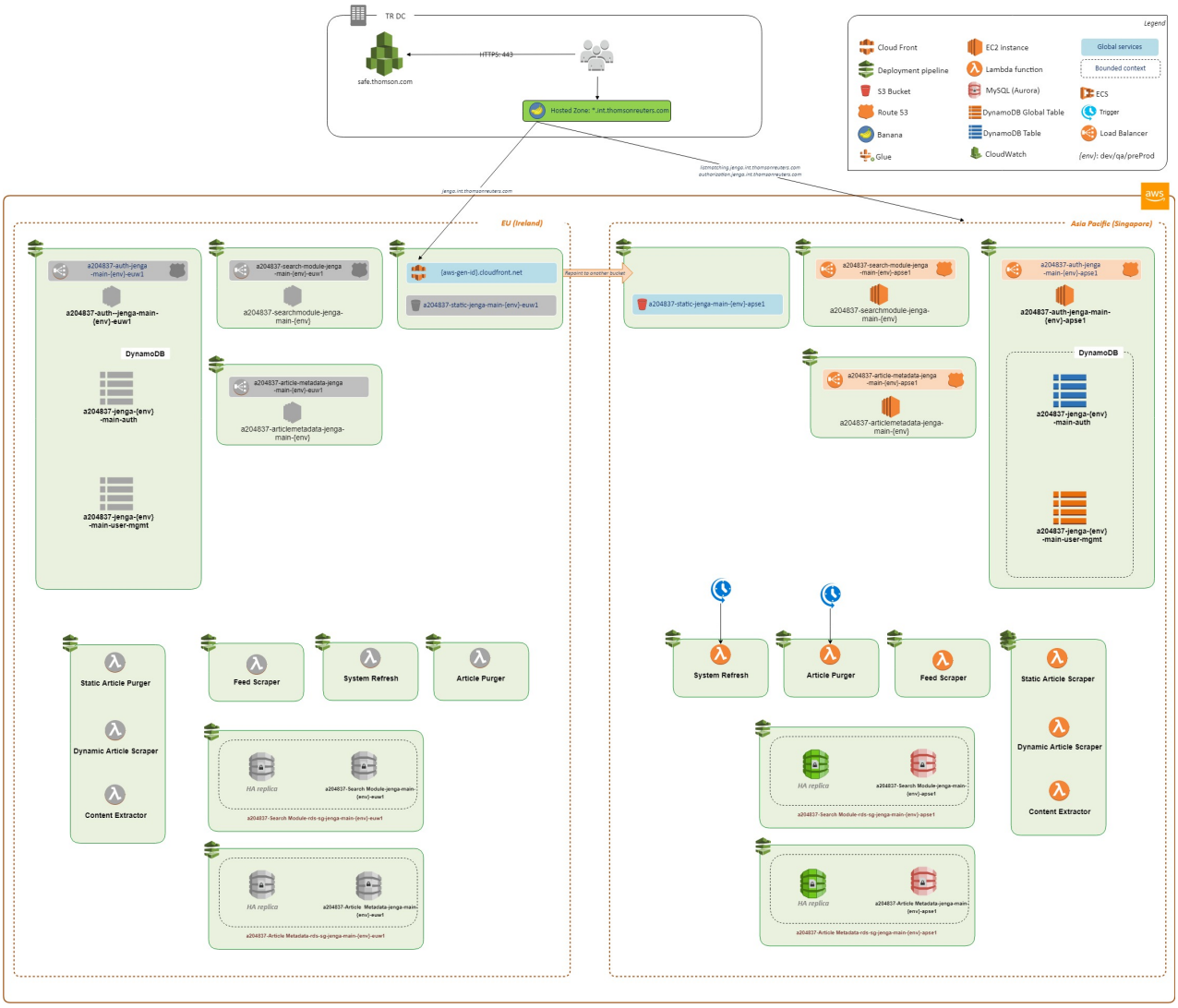
5.3.1 Pre-requisites

- F&R code pipelines are created in both regions;
- All components are created in both regions;
- DR region works with minimum reserved capacity;
- Read-replica in DR region is up and running for SearchModule and ArticleMetadata databases

5.3.2 Procedure

- Promote rds clusters to Master(SM/AM instances);
- Create Read Replica in Singapore(SM/AM instances);
- Re-point CloudFront to use DR bucket(skip in case already done by LM);

- Re-point prod-environment services (*{SM,AM,Auth}*.jenga.int.thomsonreuters.com[2]) to the DR region;
- Enable System Refresh/Article Purger Triggers in Singapore Region



5.3.3 Switching back to primary region

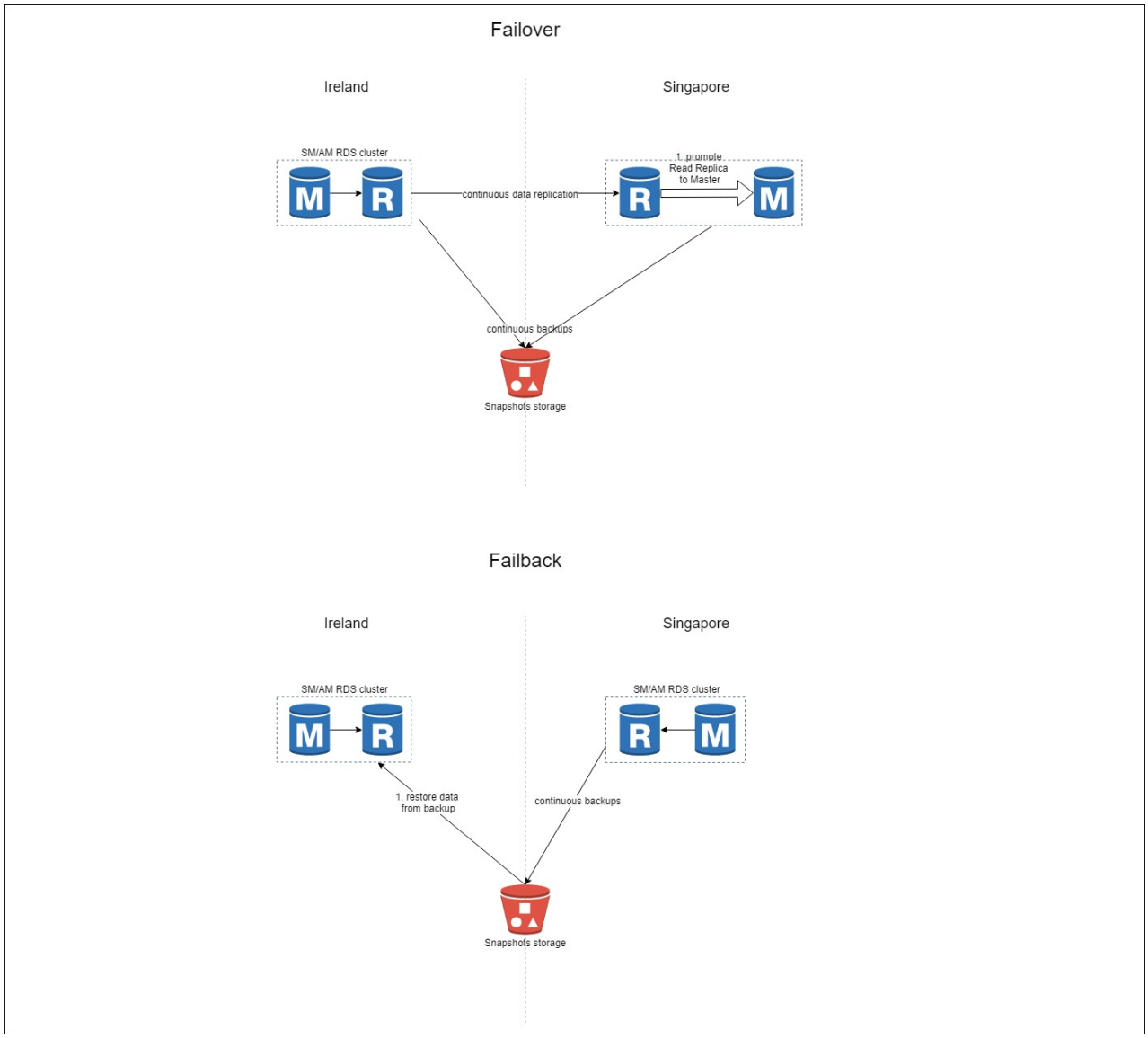
This process should have no interference with the normal operation of the users, however the system will be considered in a "Maintenance mode" and for some time will not allow users interact with the system.

Here are the high level steps:

- Check that all components in a primary region are up and running;
- Restore Search Module/Article Metadata databases data :
  - Create a RDS Snapshot of SM/AM databases from Singapore Region;
  - Restore SM/AM databases from Snapshot into Ireland region;
- Re-point CloudFront to use primary region bucket(skip in case already done by LM);
- Re-point prod-environment url's (*.int.thomsonreuters.com[3]) to the primaty region;

2 http://jenga.int.thomsonreuters.com/
3 http://int.thomsonreuters.com/

- Disable Lambda Triggers in Singapore Region and ensure that are enabled in Ireland Region;
- From Ireland Region, create a Read Replica for Aurora SM/AM databases in Singapore Region;



6 Login and Authentication flow

Current Solution

6.1 SAFE Security Background

Max Session duration is 18 Hours, then system forces a Logout

Max time of in-activity is 20 minutes, then system forces a Logout

The following chart helps explain the differences between the three different Security Levels SAFE has.

	SAFE LEVEL 5	**SAFE LEVEL 7**	**SAFE LEVEL 10**
Inactivity Timeout	9 hours	1 hour	20 minutes
Type of Data being protected	Generally available internal company, department, or team/ project information	Critical Company and/or employee-related information	Private and/or Personally confidential information
Sign-on Experience	Seamless (when requirements are met)	Sign-on always required (unless coming from a Level 7 or Level 10 app)	Sign-on always required
Application Examples	theHub	MyExpenses	Workday, Benefits
Sign-on Experience	Seamless (when requirements are met)	Sign-on always required (unless coming from a Level 7 or Level 10 app)	Sign-on always required
Type of Data being protected	Generally available internal company, department, or team/ project information	Critical Company and/or employee-related information	Private and/or Personally confidential information

Jenga project is classified as **SAFE Level 10**

6.2 Functional Requirements Description

In order to be complaint with SAFE Level 10 policies has been introduced in this section new requirements based on Max User Session duration and Max User Session In-active time

In more detail, Max User Session Duration is of 18 hours and after that User is automatically logged out. Max User Session In-active time is of 20 minutes and after that User (if not requested for extension) then is automatically logged out at Session Expiration time.

In both cases, if user needs an further access to Jenga then a new login to SAFE will be required.

Please refer to the followinfg scenario for more details

6.2.1 Scenario #1 - Warning about expired session

Given the User is Jenga User
And the User is logged to Jenga Application
And the User is active (has open Jenga App in browser and TAB)
When user is active during 15 minutes (5 minutes left till 20 minutes)
Then Warning message should be displayed to the User
And User is able to extend session or let token to expire
*

1a **If** the User selects Resume session option
Then System should extend User`s Session

6.2.2 Scenario #2 - Autologout due to inactivity

Given the User is Jenga User
And the User is logged to Jenga Application
When the User is in-active for more than 20 consecutive minutes
Then the User should be logout immediatly

Scenario #3 - User Logout Request Application

Given the User is Jenga User
And the User is logged to Jenga Application
When the User select Logout
Then the User should be logout immediatly

6.2.3 Scenario #4 - Autologout when reaching Max Session Duration (18 hours)

Given the User is Jenga User
And the User is logged to Jenga Application
When the User is active 18 hours
Then the User should be logout when his token expires

6.3 Security Requirement

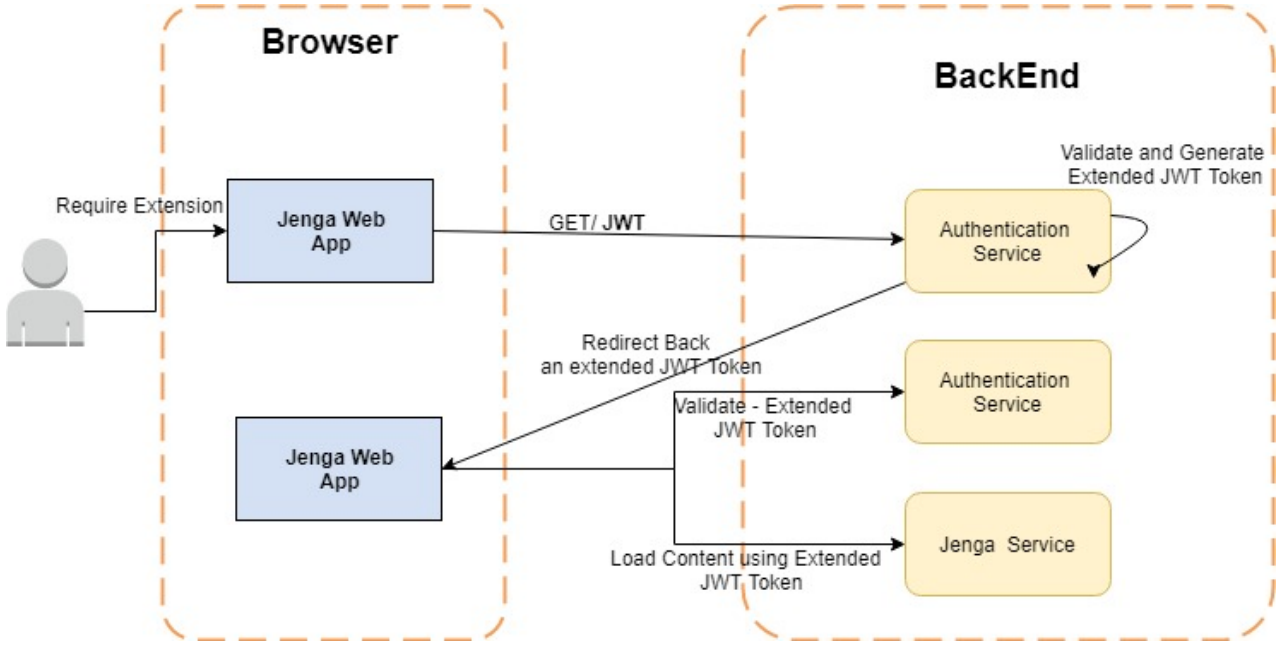
6.3.1 Refresh JWT Token every 15 minutes in Background

As Security Architect i want to increase security on JWT Token having a periodic refresh every 15 minutes in background so this reduce the risk associate to a eventual stolen token

Note: this is trasparent to the user, will not have any impact on User Experience

6.4 Proposed Solution

User Session Extension Solution



The Proposed Solution is designed for covering the listed requirements above. The Graph shows the flow when user requests a session extension.

Key Implementation part of the solution is a additional End-Point for re-authorization within the actual Back-end Authentication Service.

6.4.1 Technical considerations:

At high level such end-point will be invoked for extending exsting Token

An inactive session longer than 20 minutes triggers a Token Cancellation

A Max Session Duration of 18 hours triggers a Token Cancellation and an addition Login may be required

Authorization Service will receive User's JWT Token and will validate Token data (email, SAFE ID) against DynamoDB Data and If Validation Pass, then a new Token is created and previous one removed. If validation doesn't pass then Token is removed and no new Token released.

User will receive the extended token in background every 15 minutes

7 SM AWS Resources Inventory

Authorization

Environment	Instance type	Ireland Number of nodes min (max)	Singapore Number of nodes min (max)
Dev	t3.medium	1(1)	1(1)
QA	t3.medium	1(3)	1(3)
PPE	t3.medium	2(6)	2(6)
PROD	t3.medium	2(6)	2(6)

Search Module

Environment	Instance type	Ireland Number of nodes	Singapore Number of nodes
Dev	t3.large	1(1)	1(1)
QA	t3.large	1(3)	1(3)
PPE	t3.large	2(6)	2(6)
PROD	t3.large	2(6)	2(6)

Article Metadata

Environment	Instance type	Ireland Number of nodes	Singapore Number of nodes
Dev	t3.large	1(1)	1(1)
QA	t3.large	1(3)	1(3)
PPE	t3.large	2(6)	2(6)
PROD	t3.large	2(6)	2(6)

Search Module RDS

Environment	Instance type	Ireland Number of nodes	Singapore Number of nodes
Dev	r4.large	1 + 0	0
QA	r4.large	1 + 1	0
PPE	r4.large	1 + 1	0
PROD	r4.large	1 + 1	0

Search Module RDS DR Replica

Environment	Instance type	Ireland Number of nodes	Singapore Number of nodes
Dev	r4.large	0	1 + 0
QA	r4.large	0	1 + 0
PPE	r4.large	0	1 + 0
PROD	r4.large	0	1 + 0

Article Metadata RDS

Environment	Instance type	Ireland Number of nodes	Singapore Number of nodes
Dev	r4.large	1 + 0	0
QA	r4.large	1 + 1	0
PPE	r4.large	1 + 1	0
PROD	r4.large	1 + 1	0

Article Metadata RDS DR Replica

Environment	Instance type	Ireland Number of nodes	Singapore Number of nodes
Dev	r4.large	0	1 + 0
QA	r4.large	0	1 + 0
PPE	r4.large	0	1 + 0
PROD	r4.large	0	1 + 0

8 SM Monitoring view

8.1 Architecture design

8.2 Beacon black-box monitoring

The Hub. Beacon[4]

8.2.1 Purpose

Provide view on system health in general and health of particular components/services of the system from Users/ Clients perspective by running bunch of acceptance tests against system, that provides an insight in system operational readiness and correctness.

These tests should be:

- **Fast** - to provide quick feedback on health of the system. Whole suite should run <0.5min, cause it will be scheduled to run each 1min
- **Reliable** - to not provide false positives, cause Beacon dashboard will be proactively monitored by Service Ops and each failure will raise an alert to be actioned upon
- **Stateless** - to not generate test data on Production and to not leave trace/footprint after monitoring calls

It will be enough to have at least 1 test per Microservice, which will check health of this Microservice by calling any it's endpoint.

8.2.2 Test scenarios to monitor

Test scenarios might be found by link: Test Scenarios for Black-box System Health checks[5]

2. List Matching Service. **GET** https://listmatching.jenga-prod.int.thomsonreuters.com/health

Invocation of /health endpoint includes checks:

- availability of List Matching load balancer
- availability of List Matching service
- availability of S3-bucket (*content* bucket)
- availability of DynamoDB (*BatchInfo* table)

8.3 White-box monitoring

8.3.1 Purpose

Provide near real-time monitoring of AWS resources used by system that are critical for it functioning. The whitebox monitoring examines component health and should report an error in the case if component is unavailable.

4 https://thehub.thomsonreuters.com/groups/beacon
5 https://confluence.refinitiv.com/display/Jenga/Test+Scenarios+for+Black-box+System+Health+checks

Whitebox monitoring results are visible in the DataDog monitors. DataDog has an integration with CAM alerting tool to setup proper Alerts based on defined thresholds.

All metrics might be classified:

- ***System metrics.*** *Standard metrics that are available in CloudWatch, doesn't require code changes;*
- ***Custom metrics.*** *Published via log-abstraction tier, requires explicit invocation from code;*

8.3.2 System metrics, collected by DataDog from AWS

*For **EC2** the following metrics should be made available:*

- Cpu utilization
- Memory utilization
- Status check failed system
- Status check failed instance
- Host OK

https://docs.datadoghq.com/integrations/amazon_ec2/

*For **ALB** the following metrics should be made available:*

- Healthy host count
- Target response time p95
- Active connection count

https://docs.datadoghq.com/integrations/amazon_elb/

*For **Lambda** functions the following metrics should be made available:*

- Duration
- Invocations
- Invocation errors
- Throttled functions

https://www.datadoghq.com/blog/monitoring-lambda-functions-datadog/

*For **DynamoDb** the following metrics should be made available:*

- Throttled reads per minutes
- Throttled writes per minutes
- Failed Requests
- Consumed Reads Capacity
- Consumed Writes Capacity
- Table Size
- System Error
- User Error

https://docs.datadoghq.com/integrations/amazon_dynamodb/

*For **RDS** the following metrics should be made available:*

- CPU utilization
- Database connections
- Freeable memory
- Free storage space
- Replica lag
- Active transactions
- Blocked transactions

https://docs.datadoghq.com/integrations/amazon_rds/

*For **SQS** the following metrics should be made available:*

- Messages Received
- Messages Sent
- Messages Visible
- Message age

https://docs.datadoghq.com/integrations/amazon_sqs/

8.3.3 CAM integration

https://thehub.thomsonreuters.com/docs/DOC-851597

8.4 Thresholds and alarms

8.4.1 System state (Green, Yellow, Red)

- System state monitoring is implemented as several datadog monitors.
- Each monitor aggregates an event stream received from the backend service and calculates a metric for a single part.
- The monitor can be in one of the following defined states: OK, WARNING, ALERT.
- The monitors can be composed into larger parts with boolean operators which allow the monitoring of the whole system's current via one monitor if required.

8.4.2 Alert rules for SM services(please refer to the Run Book)