

AWS
re:Invent

ENT214

Cloud migration in the face of data center eviction

Matt Dimich

Head of Solutions Architecture
Thomson Reuters

Jamie Greco

Sr. Product Manager Technical
Amazon Web Services

Agenda

Introductions

Problem statement

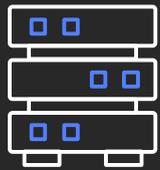
People

Process

Technology

Lessons learned

Thomson Reuters story to this point



Building new
project cloud native



re:Invent 2016 and
2017: multi account
Landing Zone
strategy



Company split



re:Invent 2018:
Landing Zone
design: What to do
when your company
splits in half?



What do we do with
our data centers?



the answer company™

THOMSON REUTERS®

Thomson Reuters is the most trusted provider of essential news, information, and tools for professionals in the legal, tax and accounting, compliance, government, and media markets.

Our customers rely on us to deliver the intelligence, technology, and human expertise they need to find trusted answers that inform their most important decisions.

We have operated in more than 100 countries for more than 100 years and have 25,800 employees around the world. Some of our primary locations include Toronto, Minneapolis/St. Paul, London, Dallas, Ann Arbor, and New York.

Thomson Reuters structure (post split)



Platforms

Infrastructure, hosting, and networking
(Cloud Center of Excellence and Data Center teams unite)

Security



2-year deadline



300+ services and applications
and 10,000+ servers



Merging our Data Center Technology and Cloud Center of Excellence teams provided flexible resourcing to adapt to our migration needs

Cloud strategy acceleration required

What we needed:



Accelerate our migration



To empower our teams



To operate in the cloud without consolidating 170 existing processes



Operational security and compliance to support their business



Visibility; not black box to processes and tools



To leverage best practices



Focus on innovation and cloud-native strategy



Reduce security and compliance risk

“How do we migrate quickly, stay security neutral or better, and not take some of our legacy proprietary technology with us to the cloud?”

—DC Exit Steering Committee

AWS Managed Services (AMS)



Provisioning/Onboarding



Patch and backup



Security and compliance



Change management



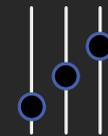
Monitoring/Alerting



Incident management



Insights and cost optimization

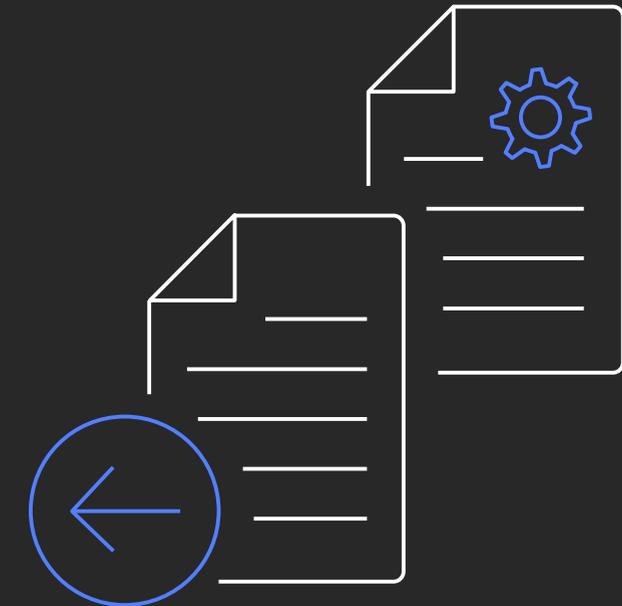
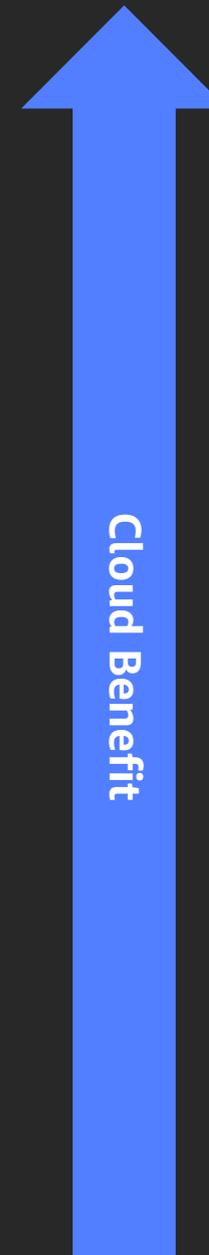


Service integration

Cloud migration strategy

Portfolio assessment

Rebuild	<ul style="list-style-type: none">• Fully cloud native• Cost-optimized architecture• Infrastructure as code• Immutable infrastructure• Increased developer efficiency	Cloud native
Revise	<ul style="list-style-type: none">• Application is optimized for cloud• Elastic IaaS + some PaaS• Infrastructure as code• Immutable infrastructure	
Refactor	<ul style="list-style-type: none">• Mutable infrastructure• Leverage some AWS managed services (i.e., Amazon Relational Database Service (Amazon RDS))• Needs OS in place patching and/or upgrades	Cloud ready
Rehost	<ul style="list-style-type: none">• Mutable infrastructure• Leverage some AWS managed services (i.e., Amazon RDS)• No OS version updates• Needs OS in place patching• Bit for bit copy	
Replace	Application is replaced with commercial SaaS/aPaaS	
Retire	Application is decommissioned and shutdown	



The decision between refactor and rehost is your migration approach

Acceleration approach

To acceleration our journey to the cloud in the face of data center eviction we had to make significant changes in three major areas:



People



Process



Technology



Enabling people to succeed

Organizational changes

Centralization and consolidation of Cloud Center of Excellence and Data Center Teams

- Facilitated frictionless collaboration
- Enabled resource flexibility to rally on the highest priority
- Accelerated opportunities for resources to upskill

Program change

Structure distributed direct responsibilities to the portfolio owners vs. centralized migration factory that worked through the application teams

Supporting process, resources, and training

Steering committee with members from all business units

Steering committee focused on prioritization and enabling portfolio owners to take ownership and drive the migration for their portfolio

Motivation—timeline, ownership, implications for not making it (heavy financial impacts)

Communication transparency to all the information means they did not have to convince people it was important, it was obvious

Get the right people involved

People with the knowledge need to be owners



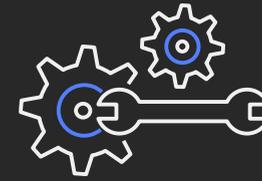
Processes to migrate and operate



Discovery



Pattern development



Operational



Migration

1

2

3

4

Get started with MVP Process and iterate; don't wait until process is fully baked, it will slow you down!

A defined escalation path for each process is essential to keeping the program on track



Process: Discovery

Eviction changed discovery from cloud native to cloud ready

Less dramatic change of architectures

Focus the most efficient way to migrate with the fewest changes

- But include MV refactoring as project allows

Discovery is critical, and anything we miss slows down the migration teams later

Discovery is never perfect, plan for wrong information in your migration process



Iterate



Perfect

Process: Patterns

Often central team implemented the pattern
(Oracle, Windows file shares ...)

The more patterns the more the central team was a potential bottleneck

More patterns mean more operational costs

Patterns were a way to work efficiently
and implement as few “new” things as possible

Increased the efficiency of discovery over time

Transparency is key when developing patterns
that are dependencies of application migrations



Process: Operations

Gameday sessions are key

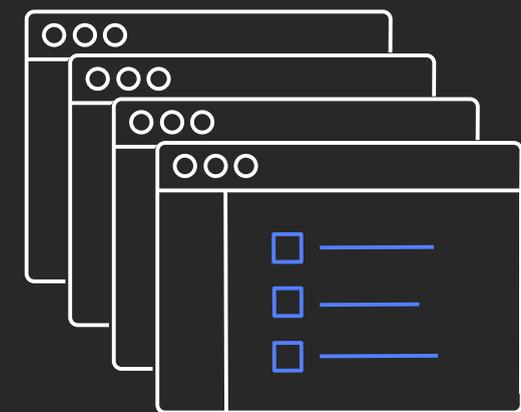
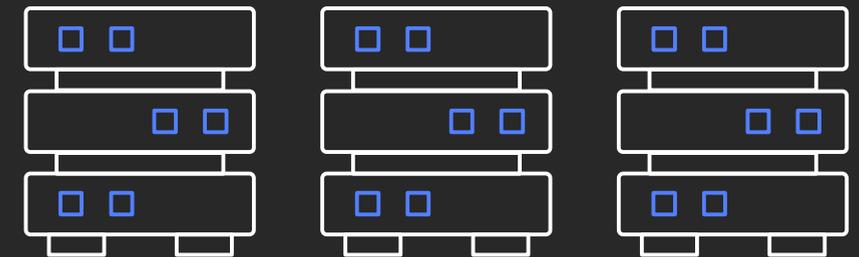
Operations processes need to be defined early

- They tend to be overlooked until it's too late and an incident is already happening
-

Changing from on-premises to cloud-ready involves a mindset change with your configuration items

- Cis on-premises are often servers, disk devices, network switches, etc. and you use these as anchors for change
- Cis for cloud ready are the application name and your ITSM procedures need to be updated to reflect that

Training sessions prior to gamedays are essential or gamedays become training



Process: Migration

Cohort

Trying to accomplish migrations BAU took several times longer than when people got together in a room

For us the cost of flying people across the globe to work for a week was easily made up by the financial penalties we avoided by delivering the program successfully

Having the infrastructure created **before** the cohort, enabled time to adapt to unknowns from discovery

Have all the people you need in the room and as few remote lifelines as possible (network is a good example)

Discovery → Migration

Common format that's consumable by machine enables more efficient infrastructure creation

Automating infrastructure creation enabled easy iteration before cohorts

Automating RFCs became critical for efficiency as well

There is no substitute for getting people in the same room to accomplish a goal



Acceleration through automation

Infrastructure creation would take **weeks**

Iterating on infrastructure creation
(i.e., load balancer configurations) was hard

Developers were frustrated

The program was not on track



Acceleration through partnership

1. AMS took Landing Zone creation down: 2 weeks to days (44 Landing Zones)
2. AMS created features that further accelerated migration and enabled automation (i.e. Developer Mode)

Automation is a key enabler but people and process are equally as important. Have a partner that can adapt and evolve quickly to meet your needs.

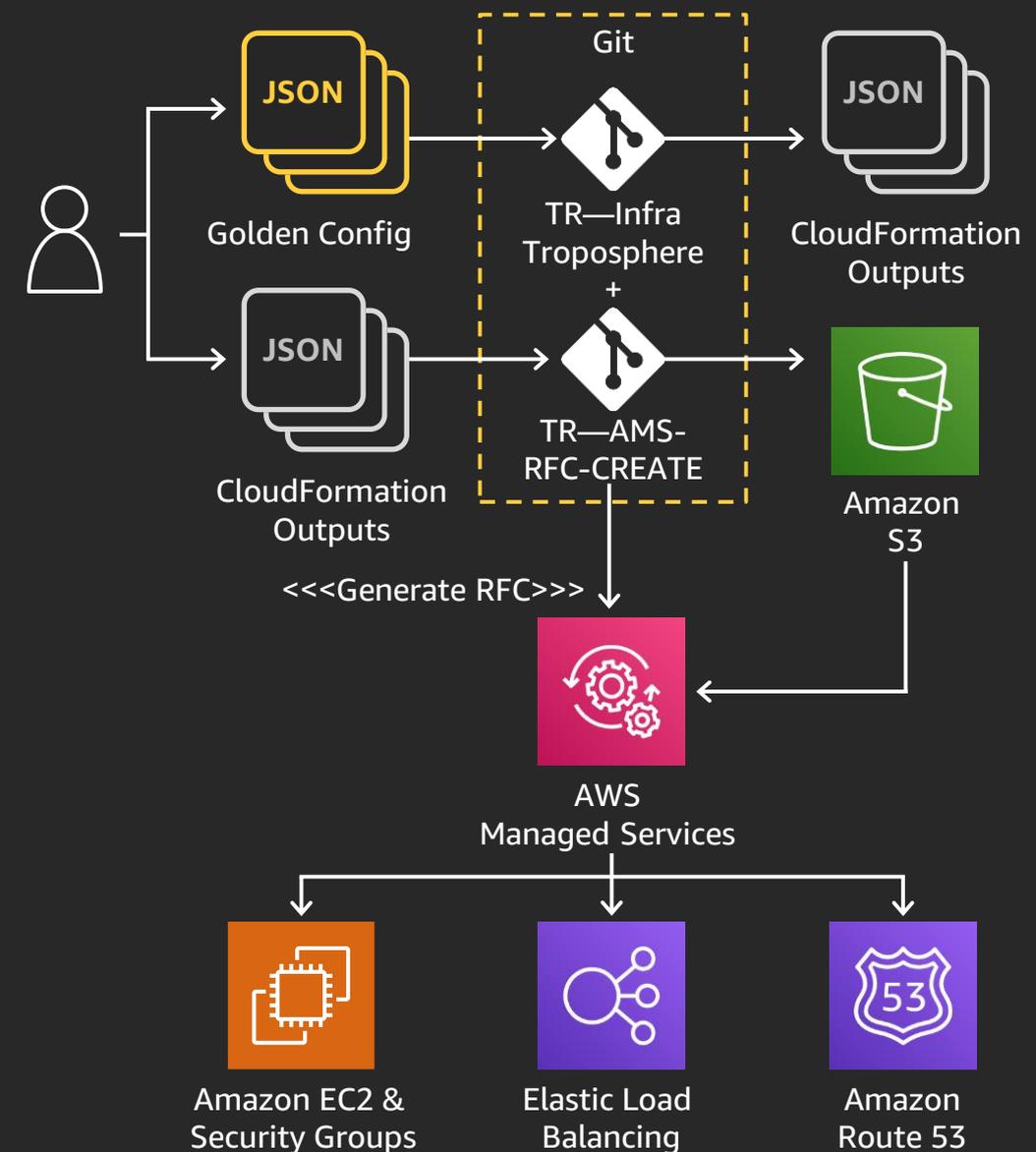


Automation: Infrastructure creation

Discovery and building infrastructure

1. Troposphere
2. Goal: Minimize change to input file
3. Broken down into multiple AWS CloudFormation chunks
4. Tool to automatically create Requests for Changes (RFCs)

Infrastructure deployment using TR Troposphere framework and TR AMS-RFC-CREATE



Demo

```
{  
  "version":1.1,  
  "tr:application-asset-insight-id": "1234",  
  "tr:environment-type": "PRODUCTION",  
  "tr:financial-identifier": "123456789",  
  "tr:resource-owner": "Support@thomson.com",  
  "tr:project-name": "DEMO",  
  "aws_account": "104007887976",  
  "region": "eu-west-1",  
  "short_name": "",  
  "security_group_load_balance": [],  
  "security_group_ec2": [],  
  "ec2": [],  
  "version2_lb": [],  
  "route53_health": [],  
  "route53": []  
}
```

Automation: Application deployment

Deploying the application

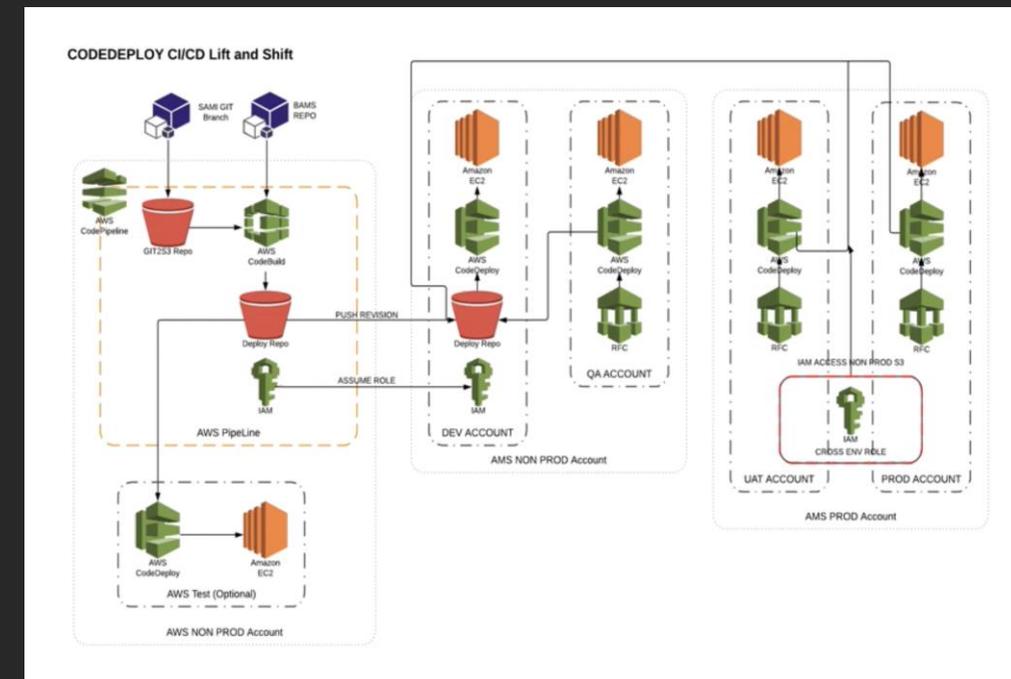
1. Common deployment pattern enabled innovation
2. Supports multiple deployment types
3. Automated change requests
4. Simple input and apply to many applications

Application Deployment

TR Troposphere Framework

Amazon Managed Services

Amazon Web Services



```
file.yml
1  version: 0.2
2  env:
3    variables:
4      #BAMS Artifact info
5      #buildsourcepath is the path in BAMS where the application artifact resides.
6      #Remove the https://bams-aws.refinitiv.com/artifactory/default.generic.global/
7      #art of the string.
8      #buildsourcename is the file anme of the application artifact in BAMS
9      buildsourcepath: "Lynx/DataSync/builddistribution/LynxDataSyncTools/LynxDataSyncTools.4.2/"
10     buildsourcename: "LynxDataSyncTools.4.2.4201.0e8dfc8.zip"
11     #Asset ID for the applciation
12     assetid: "234683"
13     #Options for if do not have own AWS account then select remote hosted to y and ente the asset of
14     #the aws account
15     remote_hosted: "y"
16     host_assetid: "4567325"
17     #AMS Account info
18     #i.e. - tr-ams-news-nonprod-eu-west-1
19     #ams_acc_id - account ID of the non-prod AMS account the application being installed into
20     # this is a list of the dev envs
21     #i.e. - 217158195230 list looking
22     #ams_acc_region - region that host sthe remote ams Account this is a list of the dev envs
23     #i.e. - us-east-1
24     # Both of the above lists need to be in the same order and the primary dev details must be first in t
25     # as this is where the buckets are held to deploy to all other prod and dev accounts
```

Summary

Our centralized teams got cut in size because of the divestiture, so to execute on DC Exit we needed to do everything we were doing + a cloud migration

AMS helped us become operational in the cloud to support legacy applications so we can focus on our value add more quickly:

Modernizing our applications

Lessons learned: Recap

1. Merging our Data Center Technology and Cloud Center of Excellence teams provided flexible resourcing to adapt to our migration needs
2. The decision between Refactor and Rehost is your migration approach
3. Get the right people involved
4. People with the knowledge need to be owners
5. Get started with MVP Process and iterate. Don't wait until process is fully baked, it will slow you down!
6. A defined escalation path for each process is essential to keeping the program on track
7. Discovery is really important, and anything we miss slows down the migration teams later on
8. Discovery is never perfect, plan for wrong information in your migration process
9. Transparency is key when developing patterns that are dependencies of application migrations
10. Training sessions prior to gamedays are essential or gamedays become training
11. There is no substitute for getting people in the same room to accomplish a goal
12. Automation is a key enabler but people and process are equally as important. Have a partner that can adapt and evolve quickly to meet your needs.

Related breakouts



ENT320

Cloud Ops Engineer: A day in the life

ENT215

Five steps AMS leverages to accelerate cloud adoption

ENT214

Cloud migration In the face of data center eviction

ENT229L

Leadership session: The enterprise transformation journey with AWS

MGT203

Simple path to AWS Managed Services (AMS), 2nd Watch

Thank you!

Matt Dimich
matt.dimich@thomsonreuters.com

Jamie Greco
grecjami@amazon.com



Please complete the session survey in the mobile app.