Clear as Clouds

New approaches with AWS

Apeksha Jain
AWS Senior Solutions Architect
jainapek@amazon.com
24th March 2021
Walk through of next 45 minutes

- re:Cap
- Migration, Modernisation and Transformation
- Different Migration Paths
- Patterns for Decomposition
- Technology and Use Cases
  - Serverless
  - Containers
  - Infrastructure as Code
- Research and Ronin
- Thinking Beyond
re:Cap from last time
Migration, Modernisation and Transformation

What’s the difference?

Link to a podcast on the topic
Development Transformation at Amazon.com

1994-2001

Monolithic architecture + hierarchical organization

2002+

Decoupled services + Two-pizza teams
Break up the work

People
Smaller teams that own products and services, and organize around customer outcomes

Components
Smaller releasable components that are easier to own and operate, with well-defined boundaries, and a smaller blast radius
Monolith

- UX
- Users
- Threads
- Posts
- UI layer
- Data access layer
- Shared database

Microservice

- UX
- User UI
- Threads UI
- Posts UI
- API
- Users
- Threads
- Posts
- DAL
- DAL
- DAL

© 2021, Amazon Web Services, Inc. or its Affiliates.
If time was not an issue...
## How to select the correct application

<table>
<thead>
<tr>
<th>Business Reasons</th>
<th>Technical Reasons</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOB, critical to business success</td>
<td>Old technology – no support</td>
</tr>
<tr>
<td>Customer facing</td>
<td>Performance issues, inability to scale</td>
</tr>
<tr>
<td>Significant impact to revenue</td>
<td>Lack of extensibility</td>
</tr>
<tr>
<td>Proprietary business logic (not HR, CRM)</td>
<td>Lack of skill sets, rewrite and build skills</td>
</tr>
<tr>
<td>Impacts a large # of customers</td>
<td>Tightly coupled, single codebase</td>
</tr>
<tr>
<td>Market differentiator</td>
<td>Difficult to integrate</td>
</tr>
<tr>
<td>Value exceeds cost</td>
<td>Expensive to run (move to open source)</td>
</tr>
<tr>
<td></td>
<td>Too many bugs, spaghetti code</td>
</tr>
</tbody>
</table>
Migration Paths

The 6 Rs
Know your application portfolio and understand your options

**Reduce** the size of your estate*

- Retire
- SaaS

**Move** to AWS

- Lift and shift

**Modernize** on AWS

- Refactor
- Re-platform
Application migration strategies

Link to the blog post
## Comparison of Migration Strategies

<table>
<thead>
<tr>
<th></th>
<th>Effort (time &amp; cost)</th>
<th>Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retire</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Retain</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Re-host</td>
<td>■</td>
<td>○</td>
</tr>
<tr>
<td>Re-purchase</td>
<td>■</td>
<td>○</td>
</tr>
<tr>
<td>Re-platform</td>
<td>■</td>
<td>●</td>
</tr>
<tr>
<td>Re-architect</td>
<td>■</td>
<td>●●●</td>
</tr>
</tbody>
</table>

Increasing complexity
Patterns for Decomposition
Strangler Pattern example

Monolith

Shopping Cart
Orders
Inventory
Shipping

Shipping Service

Shopping Cart
Orders
Inventory

Shipping Service
Inventory Service

Shopping Cart

Orders Service
Inventory Service
Shipping Service
Facade Pattern

Analogous to a facade in architecture, a facade is an object that serves as a front-facing interface masking more complex underlying or structural code.
Adapter Pattern

An adapter is a wrapper that allows an interface to communicate with another interface without modifying the source code.
Technologies to help
What is serverless?

- No infrastructure provisioning, no management
- Pay for value
- Automatic scaling
- Highly available and secure
Move up the stack = less work for you

Serverless
- Continuous scaling
- Fault tolerance built-in
- Pay for value
- Zero maintenance
- Focus on business value

Level of abstraction
- Physical machines
- Virtual machines
- Containerization
- AWS Lambda
- AWS Fargate

Focus on business logic
University of York Goes Serverless

“When we sat down to discuss, we saw major benefits from a serverless approach for our applications with low traffic. We knew that in terms of the number of things we’re running and the financial benefits, going serverless would be the right strategy for us.”

– David Thompson, Head of Software Development
Containers: Amazon EKS and Amazon ECS

ECS
Powerful simplicity

EKS
Open flexibility
Run your containers anywhere based on your workload needs

**Serverless**
- AWS Fargate

**EC2 options**
- Amazon EC2
  - Spot instance

**Edge and 5G**
- AWS Local Zones
- AWS Wavelength

**On-premises**
- AWS Outposts
- EKS Anywhere
  - ECS Anywhere
Infrastructure as Code

The code template describes the intended state of your resources.

CloudFormation translates the intention to API calls.
AWS CloudFormation/Terraform

The code template describes the intended state of your resources

CloudFormation translates the intention to API calls
AWS CloudFormation/Terraform

The code template describes the intended state of your resources

CloudFormation translates the intention to API calls

1. Code your template
2. Upload, test, and review changes
AWS CloudFormation/Terraform

The code template describes the intended state of your resources

CloudFormation translates the intention to API calls

1. Code your template
2. Upload, test, and review changes
3. A stack is created by executing the changes
AWS CloudFormation/Terraform

The code template describes the intended state of your resources

CloudFormation translates the intention to API calls

1. Code your template
2. Upload, test, review changes
3. A stack is created by executing the changes
4. Manage many stacks and stack sets over time
Why reusable templates?

AWS cloud – Prod account

AWS CloudFormation template
Why reusable templates?

AWS CloudFormation template
Why reusable templates?

Using the same template to build the same infrastructure ensures consistency across multiple environments.
“Once you’ve adopted infrastructure as code, your ability to spin up environments that use a new feature or test something, can now be done in minutes, whereas before it would take us weeks. With AWS, that is no longer a bounding factor.”

– Derek Masseth, Chief Technology Officer
Transforming Research
RONIN

PROJECT DASHBOARD

MARS PROJECT

DESCRIPTION
For several decades, scientists across the globe have dedicated countless years in pursuit of finding life on Mars, even going as far as planning for humans to migrate to the red planet. And their recent pursuits may just bear fruit soon, since NASA has already released a detailed plan of how they are going to send humans to Mars in the coming decades. The plan involves sending humans to Mars and have them permanently reside on the planet. According to NASA, “Unlike Apollo, we will be going to stay.”

TIMELINE
476 DAYS REMAINING

RUNNING MACHINES
3/4

RUNNING CLUSTER NODES
15/124

COST
$2.34/yr

BUDGET
$6400

REMAINING
$4953

FORECASTED
$2106

SPENT
$1447

TAGS
- DECADES
- SCIENTISTS
- GLOBE
- DEDICATED
- COUNTLESS
- Pursuit
- MARS
- MIGRATE
- RELENTLESS
- NASA
- DETAIL
- INVOLVES
- REUSE
- APOLLO
- PLANET

670 GB SSD STORAGE
$80.40/month

0 GB HOT HDD STORAGE
$0.00/month

0 GB COLD HDD STORAGE
$0.00/month

2475 GB MAGNETIC STORAGE
$0.00/month

© 2021, Amazon Web Services, Inc. or its Affiliates.
Thinking Beyond..
The Oxford X-Reality Hub Ed Tech project
Thank you!

We are here to help you on your cloud journey!

Apeksha Jain
Solutions Architect
jainapek@amazon.com

Courtney Waugh
Solutions Architect
cmwaugh@amazon.com

Matt Burridge
Account Manager
mburridg@amazon.com

© 2021, Amazon Web Services, Inc. or its Affiliates.