Our Journey at Australia Post – Part 1

Nitin Sharma
Engineering Manager
Platform Engineering
Acknowledgment

I begin today by acknowledging the Traditional Custodians of the land on which I present this today, and I pay my respects to their Elders past and present. I extend that respect to all Aboriginal and Torres Strait Islander peoples.
Australia Post

• Existed for over 212 years

• Self-funded government business enterprise which receives $0 tax funding.

• In the past decade, we’ve paid over $1.5 billion in dividends to the Australian Government.

• Diverse and inclusive workforce of more than 64000
Agenda

• Present: Who, What, How

• Past: Landscape, Lessons

• Future: Change[s]
Who we are

• A centralised enablement team called the Systems team

• With a vision and a mission

• Supports a business unit comprising of ~300 people and beyond*
Where does the Systems Team fit
Where does the Systems Team fit
What we do

Provide Platform[s], Patterns and Consultation

**Patterns**
- Prescribed
  - Fargate
  - Docker (on EC2)
- Blueprints
  - KMS
  - RDS
  - S3
  - ElastiCache
  - DynamoDB
  - Lambda
- Auxiliary
  - EKS
  - Nginx Reverse Proxy
  - Squid Proxy
  - Jumphost

**Platforms**
- GitLab
- SonarQube
- Nexus

**Artefacts**
- Amazon Linux 2 AMI
- Docker Images
If I was trying to hire you

We enable our teams by providing capabilities to make great products.

**Skills:**
- Strong Linux skills
- Strong AWS exposure in production (cloudformation, cli, *infrastructure deployment frameworks*)
- Shell/Python/[Ruby]
- Production level exposure to Continuous Integration and Continuous delivery pipelines (Gitlab, Bamboo)
- Container ecosystem (Docker, Strong Foundational Kubernetes knowledge, AWS EKS)
- Configuration management (Ansible)
- Monitoring and Logging tools (New Relic, Sumo Logic)
- Serverless, FaaS (AWS Lambdas)
- Strong awareness of DevOps practices, Agile and Lean methodologies
- Exposure to working with PCI and ISM environments will be beneficial

**Attitude:**
- open to feedback, resilient, life-long learner, outcome driven
- data driven but also have the enthusiasm to archive great results.
- big believer in collaboration and pair programming.
- good at architecture and a stickler for having the big picture in front.
Product[ify]

- Asset / Product lifecycle is important.
- Define phases all the way to EOL.
- Have definitions for each phase.

**Product Lifecycle**

<table>
<thead>
<tr>
<th>Asset</th>
<th>Full Support</th>
<th>Maintenance Support Phase 1</th>
<th>Maintenance Support Phase 2</th>
<th>End of Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bitbucket</td>
<td></td>
<td>No new features or projects. Please migrate to GitLab</td>
<td></td>
<td>April 01, 2022</td>
</tr>
<tr>
<td>Bamboo</td>
<td></td>
<td>No new feature or projects. Please migrate to GitLab</td>
<td></td>
<td>April 01, 2022</td>
</tr>
</tbody>
</table>
## Asset Lifecycle Management Phases

<table>
<thead>
<tr>
<th>Full Support</th>
<th>Maintenance Support Phase 1</th>
<th>Maintenance Support Phase 2</th>
<th>End of Support</th>
</tr>
</thead>
</table>
| • New features  
• New projects onboarding  
• Quarterly patching  
• Full troubleshooting support  
• Bug fixes | • No new features  
• No new project onboarding  
• Limited Service Integration  
• Quarterly Patching  
• Full troubleshooting support  
• Bug Fixes | • No new features  
• No new project onboarding/integration  
• Limited troubleshooting support  
• Only Critical Security patching  
• No bug fixes | • Solution availability on best-effort basis.  
• No patching  
• No support to the users |

---

*Note: The table above outlines the different phases during the asset lifecycle management. Full Support includes all new features, projects, and support. Maintenance Support Phase 1 and Phase 2 gradually reduce the support offered, culminating in End of Support, where the solution is only made available on a best-effort basis.*
What we don’t do

• AWS account vending

• Most things windows

• SRE function for public facing applications
How we do it

• Agile Methodology
• Ceremonies
• Types of work
• Initiatives / Pseudo Projects
• Projects (Business Cases)
How we do it

Ceremonies

- Daily stand-up
- Fortnightly planning session
- Fortnightly retro
- Monthly planning session
- Story grooming session
- Inception session

Types of work

- Goal keeping
- Firefighting
- Planned engagement
- Feature requests
- Patching
- Team Initiatives
Achievements

• Decommissioning of our old tooling

• Successful business cases

• Culture Shift

• Organisation wide impact
Challenges

• Larger context

• Scaling our capabilities beyond our business unit

• Introducing better metrics
# Challenges

<table>
<thead>
<tr>
<th>Product</th>
<th>Support</th>
<th>Significance (number of assets x number of requests)</th>
<th>Who knows</th>
<th>How much (out of 5)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Docker Deployer</td>
<td>Current</td>
<td>50</td>
<td>person_a</td>
<td>3</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>person_b</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>person_c</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>person_d</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>person_e</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Fargate Deployer</td>
<td>Current</td>
<td>100</td>
<td>person_a</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>person_b</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>person_c</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>person_d</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>person_e</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Gitlab Runners</td>
<td>Current</td>
<td>120</td>
<td>person_a</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>person_b</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>person_c</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>person_d</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>person_e</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

---

**Product Competency Diagram**

- Docker Deployer: 12
- Fargate Deployer: 11
- Gitlab Runners: 12

![Product Competency Chart](chart.png)
Past

• Team setup

• Technology sprawl

• Processes
Past Technology Landscape
Past Technology Landscape

AWS Cloud Account Types

**Legacy**
- Legacy Public Cloud:
  - Developed 2014-15
  - Consists of 3 accounts only
  - Is being decommissioned

- Trusted Services Cloud (TSC):
  - Developed 2015-16
  - Follows older multi-account pattern which has major shortcomings
  - Planned for decommissioning

**Current**
- Public Cloud (multi-account):
  - Developed 2020
  - Multi-account model is Architecture Governance Forum (AGF) approved.
  - Multiple accounts per project allowing for better operational, security and billing segregation.

- Compliant Cloud Environment (CCE):
  - Developed 2020-21
  - All the goodness of Public Cloud (multi-account) along with additional Compliance (ISMP/PCI) controls and processes.
Future

• Evolving the Systems Craft

• Cross-team collaboration

• Technology
Lesson’s learnt

• Stable foundation is necessary.

• Work with the system.

• Things take time
Thank you

Nitin Sharma
Engineering Manager
nitin.sharma@auspost.com.au