March 26-29, 2018
Los Angeles, CA

NORTH AMERICA
OPEN NETWORKING //
Integrate, Automate, Accelerate
Cost Management in the Cloud for Enterprises

Faisal Mushtaq and Tushar Agarwal
GlobalLogic Inc.
Who we are

- Faisal Mushtaq
- Tushar Agarwal
Agenda

- Why is cost management important?
- Cost Management Challenges
- Approach to Cost Management
- A Cost Management Framework
- Automation Tools Landscape
- Case Study
- Q&A
Why is cost management important in the cloud?

- Cloud sprawl
- Bill shocks
- Unmanaged costs leading enterprises to abandon cloud journey
- Cloud costs manifest differently than traditional IT (CAPEX vs OPEX)
- Dev/Test can end up costing a lot more than production
Questions enterprises (should) ask about costs

**Planning**
- What cloud resources do I need?
- Who should be able to create them?
- How much should it cost?
- Who owns the cloud budget for the app?

**Operations**
- How much am I spending?
- Is it in line with my expected budget?
- Are there any surprises?
Questions enterprises (should) ask about costs

• Optimization
  • Am I using my resources in the most efficient manner?
  • Am I utilizing cloud features to reduce cost?
  • Can I control costs as my apps evolve?
Cost Management Challenges
Key Cloud Cost Management Challenges

- **Ownership**
  - Who owns the cloud resources?
  - Who pays for them?

- **Budgets and TCO**
  - Has sizing and TCO been done?
  - Have Pre-Go-Live costs been considered?
  - CSP provided credits are a double edged sword
Key Cloud Cost Management Challenges

• **Visibility**
  • No visibility or insights for stakeholders
  • Lack of continuous cost monitoring
  • Cost projections not available → No proactive response

• **Governance**
  • Governance systems not in place, or not enforced
  • Who gets to create what resources?
Key Cloud Cost Management Challenges

**Automation**
- Governance not enforced through tools and automation
- Non-existent or manual cost and compliance reporting

**Optimization**
- Apps not utilizing cloud features to optimize cost (e.g. autoscaling)
- Cloud platform discounts not utilized
- Periodic reviews not in place for validating evolving application architectures
Approach to Cost Management
Approach

• Think about costs from **Day 1**
• Enable **behavioural change** for costs in cloud
• Think about **cost impact of all design decisions**
• Implement, enforce and automate governance mechanisms
A Cost Management Framework
A Cost Management Framework

Automation

Initial Planning (Sizing and Provisioning) → Operational Visibility and Forecasting → Cost Optimization

Governance
Governance
Cloud Resource Ownership

• Transform provisioning practices for the cloud
  • De-centralized – App teams can provision resources, but IT team tracks usage
  • Chargeback/Showback – IT teams charge app teams for usage
  • Build reference architectures and templates (EA responsibility)
  • Automate provisioning of reference architectures and templates
  • Use a cloud aware CMDB tool (e.g. ServiceNow)
Cloud Resource Provisioning

- Governance
  - IT teams can define the governance practices
  - Application teams implement practices for their environment
    - Labeling
    - Resource lifecycle
  - IT teams track conformance to governance practices
Tagging and Labeling

• Rule of thumb is to use tags for resource management and labels for resource identification, grouping, searches & billing.

• Define list of labels and tags to be applied.
Tagging and Labeling

- Identification/Classification Tags
  - BU/Cost Center
  - Application
  - Owner-email – application owner/group
  - Environment – Prod/Dev/Test/QA/Perf
  - Environment-Name – Prod1a, Dev4 etc.
  - Chargeback/Showback ID
  - created-by - User who created resource.
  - role = <db, appserver, proxy, etc.> - Classify by application role
Tagging and Labeling

• Operations/Automation Tags
  • schedule-* - Used to drive instance scheduling
  • can-delete = <true/false> -
    • Can be added by app teams once resources are ready to be removed.
    • Can also be added by automation scripts, after untagged resources have been reported and no action taken.
    • Subsequently, a delete script will read this label and clean up this resource).
  • image-type - App type for baseline images, e.g. Apache,
  • image-version - Adds version ID of all the images of a certain app.
  • Reservation-expiry - Used to alert and renew reservations
Inventory Management

• Build or use a lightweight inventory management system
  • Continuously track cloud sprawl
  • Report current inventory, new resources, projected costs
  • Do periodic gap analysis
Initial Planning
Initial Planning (Sizing and Provisioning)

• TCO and Budgeting
  • Use max CPU/RAM for budgeting
  • Do initial right-sizing based on on-prem monitoring data
  • For Dev/Test, consider uptime hours (9x5 as opposed to 24x7) for TCO calculations
  • Enforce budgets from Day 1
Initial Planning (Sizing and Provisioning)

- Service Catalogs and Provisioning
  - IAM policies to restrict creation of only allowed services
  - Build IT certified base images and templates for reference architectures
  - Publish and enable self-provisioning through tools like ServiceNow
  - Integrate with approval processes
Operational Visibility and Forecasting
Reporting Approach

• Daily Reporting including cost, utilization, non-conformant resources
• Analytics based recommendations
• Monthly billing reports to required stakeholders
Reporting and Automation Architecture

- Reporting
  - Recommendations
  - Automation
  - Exceptions
- Billing Optimization
- Inventory
- Tagging
- Resource Optimization
- Billing Data
- Audit Data
- API
- Utilization Data
- Cloud Accounts
Data Points to Report

- **Cost (by app/environment)**
  - Daily, MTD and Projected Monthly Spend
  - Budgeted vs Actual, and Overrun Projection
  - Alerts on any change in usage pattern and/or budget overruns

- **Utilization**
  - Show unused resources
    - Unattached disks
    - Orphaned snapshots
    - Unallocated IPs
    - Unused/unaccessed Storage (recommend archival)
  - Show underutilized resources
    - Show instances and environments with no utilization
Data Points to Report

• Inventory
  • Current Inventory
  • New resources created + corresponding cost
  • New projected monthly spend based on new resources

• Conformance
  • List of resources without tags and labels
  • List of resources not confirming to naming conventions
  • List of instances based on older versions of baseline images
Recommendations (with savings projections)

- Rightsizing
- Eligible Reservation Planning/Committed Use
  - *Results in up to 24%-57% potential savings*
- Instance Scheduling
- Eligible Spot/Pre-emptible Instances
  - *Results in up to 60-80% potential savings*
- Reserved/Committed Instance Renewal Alerts (for instances with approaching expiry dates)
Cost Optimization Approach

• Effective Cost Optimization is a combination of two approaches:
  • In-depth operational visibility for action by app teams
  • Automatic resource cleanup (as a fall-back)

• Approach
  • Stakeholders given opportunities to take action, or register exceptions
  • No action from stakeholders → recommendations are automatically actioned
  • The two approaches, used in combination will lead to better awareness and accountability
Automation Approach (Tools vs In-House)

• If automation not available in tools, build in-house.
• Start small and grow the automation catalog.
• No one tool will solve all cost management problems. Build and integrate tools as services (see Reporting and Automation Architecture)
Infrastructure Optimization

• **Instance Rightsizing**
  • Right-size individual instances based on utilization

• **Cleanup of unused resources**
  • As per utilization reports

• **Cleanup of underutilized resources**
  • Minimal or No CPU utilization
  • Minimal or No Disk activity
  • Minimal or No IO activity
Infrastructure Optimization

• **Instance scheduling**
  - Based on spikes in usages patterns
  - Instance scheduling for dev/test servers that don't need to be run 24/7

• **Instance modernization**
  - Based on new instance families released by the cloud providers (often cheaper)

• **Cleanup of other Cloud Services**
  - Based on Cloud based metrics
Billing Optimization

• Production - Use reserved/committed use discounts
  • Start with – 30%-40% servers (before stabilization = immediate cost savings)
  • End with – 100% servers (after app stabilizes in cloud)
  • Use Cloud Services to get TCO of reference architectures before deployment
Billing Optimization

• Dev/Test - Enable committed use & spot/preemptible instances
  • Start with – 10% servers (Eg., Toolchains – Gerrit/Jenkins etc.)
  • End with – 100% (after app stabilizes in cloud)
  • Spot/Preemptible instances for environments that can be torn down and recreated.
  • Integrate use of spot/preemptible instances with DevOps build processes.
DevOps Optimization

• Restrict number of Dev/Test environments
  • Review DevOps pipelines
  • Evaluate alternate and efficient ways for developers to unit test their code
  • Right-size Dev/Test environments

• For Consultants/Vendors
  • Customer doesn’t mind spending ← NOT AN EXCUSE
  • Think about cost impacts for all design decisions
Other Automation Opportunities

• Enforcing Labels
  • Automatically add certain missing tags such as created-by (*use to track creators of orphaned resources*).
  • Create and maintain virtual tags for Cloud services that don't support tags yet, in the inventory management system.
<table>
<thead>
<tr>
<th>Concern</th>
<th>AWS</th>
<th>Azure</th>
<th>GCP</th>
<th>Third Party/Custom</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Sizing</td>
<td>• AWS Cost calculator</td>
<td>• Azure Pricing Calculator</td>
<td>• GCP Pricing Calculator</td>
<td>• GL’s custom tools</td>
</tr>
<tr>
<td></td>
<td>• AWS TCO Calculator</td>
<td>• Azure TCO Calculator</td>
<td>• GCP TCO Calculator</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• AWS Application discovery service</td>
<td>• Azure Application discovery service</td>
<td>• GCP Application discovery service</td>
<td></td>
</tr>
<tr>
<td>Operational Visibility</td>
<td>• Trusted Advisor</td>
<td>• Azure Advisor</td>
<td>• GCP Labels</td>
<td>• Cloudability</td>
</tr>
<tr>
<td>and Forecasting</td>
<td>• AWS Organizations</td>
<td>• Azure Cost Management</td>
<td>• GCP Projects</td>
<td>• Cmpute.io</td>
</tr>
<tr>
<td></td>
<td>• Tags</td>
<td>• Azure Cost Management</td>
<td>• GCP Data Studio</td>
<td>• Cloudcheckr</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Azure Automation</td>
<td></td>
<td>• Cloudyn</td>
</tr>
<tr>
<td>Cost Optimization</td>
<td>• Trusted Advisor</td>
<td>• Azure Automation</td>
<td>• Google Cloud Functions</td>
<td>• Botmetric</td>
</tr>
<tr>
<td></td>
<td>• AWS Lambda</td>
<td>• Azure Cost Management</td>
<td></td>
<td>• RightScale</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Google Cloud Functions</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Monitoring tools (Nagios, DataDog, Sensu)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Skeddly, Ylastic, etc.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Case Study
A Retail Customer Migrating to GCP
Current State

• Migrating their entire landscape to GCP over a 2 year period
• About 40% applications already migrated to GCP
• Initial sizing and planning done
• Manual Cost Reporting with automation in progress
Challenges

- No upfront cost management strategy and governance framework
  - Some governance mechanisms defined but not enforced
  - Other mechanisms being implemented on a 3-6 month timeline
- Cost management looked at after the fact
  - Actual cloud spends substantially larger than budgets
    - Monthly spends – Actual 5x times the budgeted spent
    - GCP discounts not utilized (committed and pre-emptible instance discounts)
    - No Chargeback - App teams create resources, IT team pays for them
Challenges

• No granular visibility into spends
• Dev/Test spends substantially larger than Prod
  • ~40/50 Dev environments for some apps
Recommendations Given

• Immediately start using a Cost Management Tool
  • Build reporting and recommendations
  • Build inventory management
• Enforce labelling
• Enforce clean-up of non-conformant resource
• Use cloud discounts
Thank you!
Additional Slides (if needed)