EOSDIS
NASA'S EARTH OBSERVING SYSTEM
DATA AND INFORMATION SYSTEM

Cloud Governance at Scale

Summer ESIP 2019

Ben Williams
EED-2 Cloud Operations Lead
Benjamin.j.Williams@nasa.gov

This work was supported by NASA/GSFC under Raytheon Co. contract number NNG15HZ39C.

Elements of a Well Architected Cloud Governance Solution

End User Access
Methods of access to the cloud environment

Common Services
Infrastructure and Shared services accessible by cloud tenants

Networking
Enterprise networking strategy for intra-AWS Account communication and ingress/egress control

Security Services
Central log aggregation and security event analysis

Certification and Accreditation Strategy
Methodology to reach ATO fast with a repeatable process

Governance of Cloud Accounts
Tools for account management, budget enforcement, compliance automation + Access to CSP CLI, API, Console

EOSDIS

Agenda

GOALS

CLOUD-SPECIFIC CHALLENGES

METHODS

LOOKING FORWARD

EOSDIS

Goals

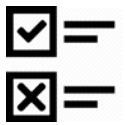
This is Sully. Sully wants toys. Sully hates water.

Image Source: Ben Williams

EOSDIS

Goals

Large multi-tenant ecosystem



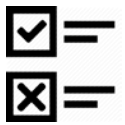
Earthdata Cloud (EDC):

- Cloud-based development environment
- Supports multiple missions and organizations
- Hosts a diverse set of applications and application architectures
  - Traditional Servers/VMs
  - Containers / Microservices
  - Serverless
  - Object Storage & Distribution
  - Network Management Apps
- Allows various application lifecycles
  - Quarterly releases
  - Daily releases
  - Continuous Delivery
- Differing developer expertise
  - Developer interns
  - Cloud experts
  - Managers and Executives

EOSDIS

Goals

Security assurances



EDC must assure a baseline of security across the entire ecosystem

- Authority to Operate (ATO)
- National Institute of Standards and Technology (NIST)
- Federal Risk and Authorization Management Program (FedRAMP)
- Agency-specific mandates
- Industry best-practices

Cloud-specific standards continue to mature as cloud adoption increases

- Many COTS and FOSS tools available for traditional Virtual Machine (VM) and Firewall model
- Few standard tools for Infrastructure-as-a-Service (IaaS) – especially:
  - Boundary protection in the cloud
  - Cloud-Native / Serverless

Goals

Budget assurances

EDC must assure expenditures do not exceed budget caps

- Antideficiency Act (ADA)
  "hard" limit

- Minimize "un-needed" expense
  "soft" limit

Traditional IT procurement supports budget assurance through up-front capital purchases

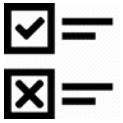
- Cost estimates
- Justification
- Approvals
- Procurement
- Inventory
- Disposition

Pay-as-you-go model in cloud requires new processes new controls

- Individual developers impact cost daily
- "Efficient" use reduces cost
  "Inefficient" use increases cost

EOSDIS

Goals



Tenant autonomy

EDC aims to provide Cloud-Users with as much development autonomy as possible

- Direct access to cloud console and application-program-interfaces (APIs)
- Direct control over cloud-resource provisioning and decommissioning
- Direct control over role and permission delegation

Tenant autonomy must be balanced with security assurance and budget assurance

EDC establishes “guard-rails” for key configurations

- Limited permissions for networking changes
- Publishing to Internet requires man-in-the-loop approval and implementation
- Role delegation within acceptable ”high watermark”

EOSDIS

Cloud specific challenges

Cumulonimbus – a very challenging cloud

Image source: Wikimedia (link) (Creative Commons)

EOSDIS

Cloud specific challenges

“Firewalls” for

Cost Overruns

Traditional firewalls prevent unwanted traffic from entering or exiting systems

Similar “firewall” type solutions are needed to ensure cloud costs can not overrun approved budgets

- *Freeze-Spend*: Removes permissions to launch any NEW resources
  > reduces daily cost rate
- *Circuit-Breaker*: Non-destructively suspends operations of EXISTING resources
  > reduces daily cost totals
- *Egress Controls*: Monitor and control data egress to avoid cost overruns

Intrusion Detection and Prevention Systems (IDPS) monitor for suspicious behavior and optionally take action to prevent activity

- *AWS GuardDuty*:
  - monitors for unusual AWS usage
    (ex: Bit-Coin Mining)
- *AWS Soft-Limits*:
  - limits the number of resources that can be created without man-in-the-loop approval

EOSDIS

Cloud specific challenges

Boundary Protections

On-Premises data centers typically have a small finite number of physical ingress/egress points.

Boundary protection tools can focus on these points and see all traffic.

In Cloud, the boundary definition is much less clear:

- **Virtual Private Cloud (VPC)**
  - *AWS EC2*: Elastic Compute Cloud
  - *AWS RDS*: Relational Database Service

  Protections similar to on-prem:
  - Firewalls / DNS / PCAP / etc.

- **Non-VPC**
  - *AWS S3*: Simple Storage Service
  - *AWS DynamoDB*: NO-SQL Datastore
  - *AWS Lambda*: Serverless Functions
  - etc. etc.

  Requires cloud ready alternatives:
  - AWS *CloudTrail*: API Usage Logs
  - AWS *CloudWatch*: Events and Metrics
  - *AWS S3 Access Logs*: S3 requests
  - *AWS S3 Bucket Policies*: S3 permissions

EOSDIS

Cloud specific challenges

Serverless

Basic on-prem security starts with the server operating system and attempts to prevent or identify compromise

- Firewall configurations
- Root permissions
- Malicious code
- Etc.

Serverless cloud resources have no OS to scan. Alternative methods are used to assess vulnerabilities:

- Static code analysis
- Invocations
  - Triggers / Permissions
  - Successes / Failures
  - Durations / Volume
- Output logs
- Optionally run within a VPC
  - to inherit network monitoring

EOSDIS

Methods

Image Source: PICRYL link (Creative Commons)

Image Source: PICRYL link (Creative Commons)

Image Source: PICRYL link (Creative Commons)

Image Source: PICRYL link (Creative Commons)

EOSDIS

Methods

Oversight

Various tools and methods are employed to monitor activity within the cloud:

- Authentication and Authorization
- Network traffic and flows
- API activity
- Server logs
- Resource utilization
- Inventory
- Compliance
- Health monitoring
- Metrics and Trends
- Errors
- “Unusual” behavior
- Etc. etc. …

EOSDIS

Methods

Automation

Operating at scale requires extensive automation

- Reduces human error
- Normalizes environments
- Accelerates updates
- Staff multiplier

Infrastructure-as-Code: *Terraform*

- Common modules and templates for:
  - application networking
  - permission management
  - security baselines
  - and more
- Continuous-Integration / Continuous-Delivery (CICD)
  - accelerate feedback to developers
  - complete and attributable history of updates to accounts

Man-in-the-loop processes reserved for true review and approval tasks

EOSDIS

Methods

Self-Service

Empowering Cloud-Users to self-service their needs allows our integrated DevOps team to keep-up with a growing user base

Cloud-User autonomy must be balanced with the need for security and budget assurances

Example: Delegating role management

- *AWS Permissions Boundaries* allows EDC to define the maximum allowable permissions
- Cloud-Users create and manage their own Identity and Access Management (IAM) Roles within the limits of the Permissions Boundary

EOSDIS

Methods

Community

Fostering and facilitating a supportive development community allows answers to tough questions to come from anyone

Multiple communication tools in-place to allow synchronous and asynchronous knowledge sharing

- EDC moderated knowledgebase and community forum
- User-guides and Getting-Started documentation
- Online document collaboration
- Secure document sharing
- End-user Wiki
- Operations Wiki
- How-To Videos
- “Office-Hours” with EDC engineer panel
- Announcement distribution lists
- Online chat for full community
- Ticket management system

EOSDIS

Methods

Customization

Special-cases come up frequently where exceptions to normal rules must be granted

- Priority customers / demos
- Special test events
- Rapid-Prototyping efforts

Permitting exceptions poses challenges to configuration management across the ecosystem

- Request and approval tracking
- Implementation estimates
- Potential for re-use
- Modular and versioned infrastructure

EOSDIS

Looking forward

More missions	More accounts
More developers	More applications

- Automate everything
- Further empower end-users to self-service
- Develop intelligent oversight tools and analytics
- Continue to build an open and supportive community

EOSDIS

This work was supported by NASA/GSFC under Raytheon Co. contract number NNG15HZ39C.

Raytheon

in partnership with

ati

Bellfore Consulting GROUP

Element 84

GENEX SYSTEMS

The HDF Group

INNOVIM The Power of Innovation

MARYLAND HAWK CORPORATION

OPeNDAP

STINGER GHAFFARIAN TECHNOLOGIES

Red Canyon Engineering & Software

VALADOR

EOSDIS