Tales from the Cloud: Real-World Experience Moving IdM to the Cloud at Illinois

IAM Online

Wednesday, September 11, 2019

Erik Coleman, University of Illinois at Urbana-Champaign

Keith Wessel, University of Illinois at Urbana-Champaign

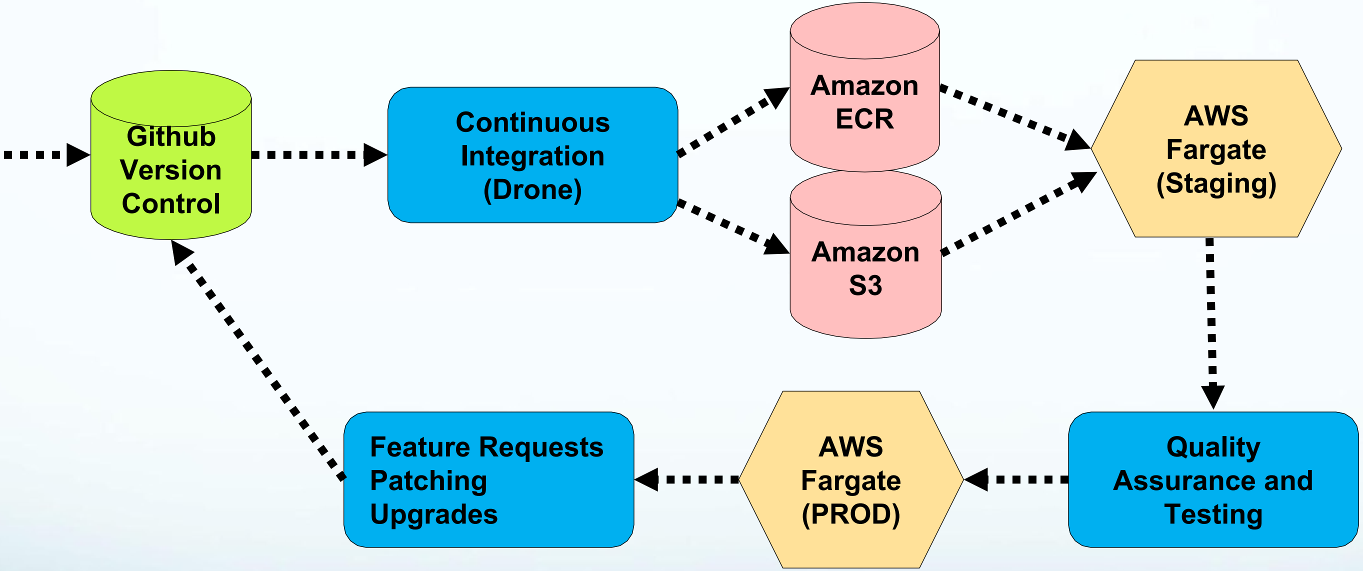
Illinois' Cloud First Strategy

- Organization-wide effort to move to cloud-hosted services
- AWS adopted first, Azure and GCP added later
- Developed an initial "DevOps" model for all central application services
- Preference for AWS EC2 (standalone instances) or AWS ECS Fargate (Docker containers)

Incommon Collaboration Success Program

- Illinois joined in 2017
- Key piece: InCommon Trusted Access Platform
- Project Proposal for Shibboleth and Grouper
- Collaboration with product dev teams
- "Greenfield" Grouper installation
- "Hybrid" Shibboleth installation

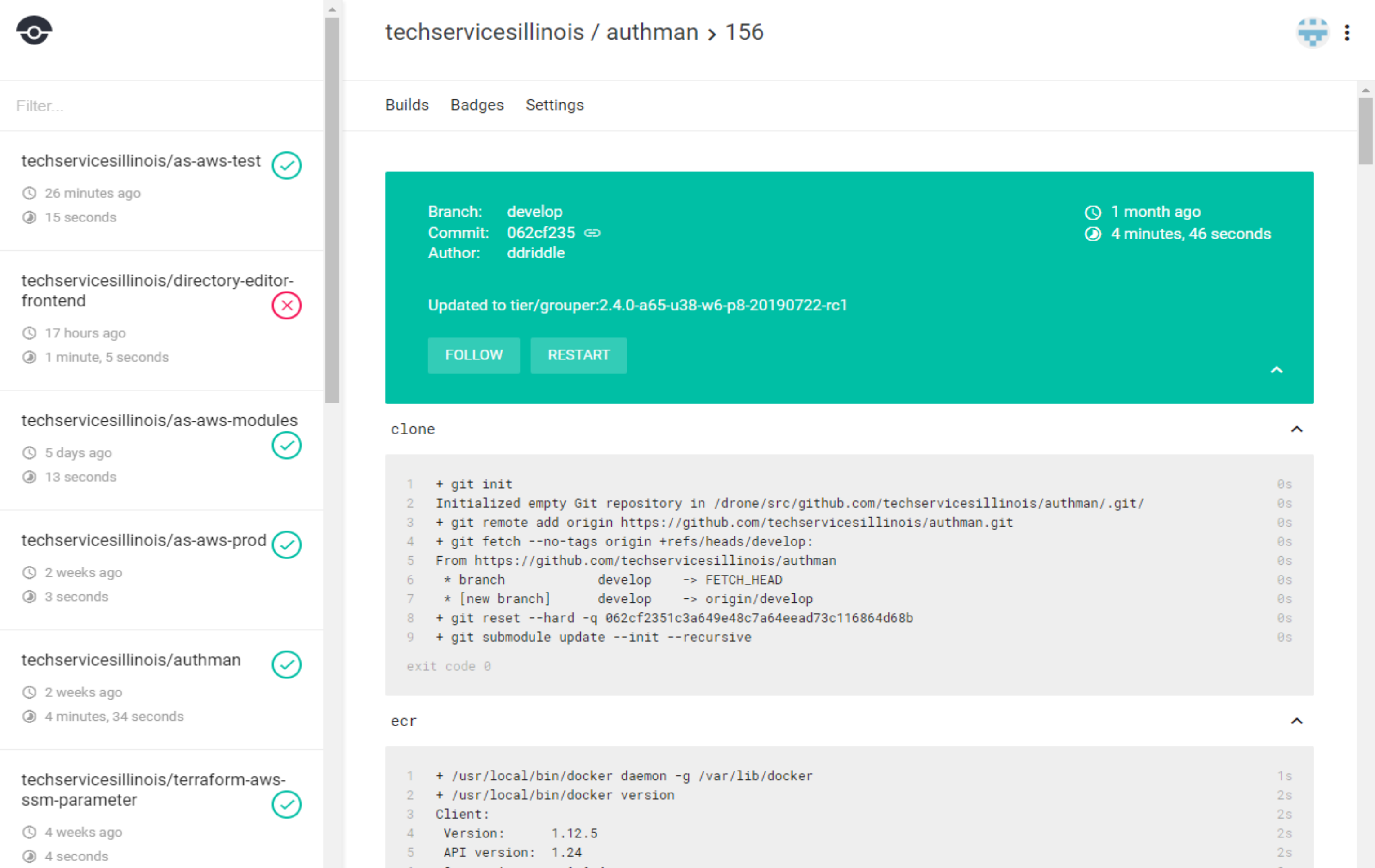
CI/CD Process for Docker Builds


Github Version Control
Continuous Integration (Drone)
Amazon ECR
Amazon S3
AWS Fargate (Staging)
Quality Assurance and Testing
AWS Fargate (PROD)
Feature Requests Patching Upgrades

Setting up AWS Infrastructure

- Terraform chosen for "infrastructure as code"
- A platform-agnostic declarative language
- Defines infrastructure elements
  - Networking subnets
  - Security groups
  - Load balancers
  - Container tasks, instance sizes
  - SSM parameter to environment variable mappings

Drone



techservicesillinois/as-aws-test ✓
26 minutes ago
15 seconds

techservicesillinois/directory-editor-frontend ✗
17 hours ago
1 minute, 5 seconds

techservicesillinois/as-aws-modules ✓
5 days ago
13 seconds

techservicesillinois/as-aws-prod ✓
2 weeks ago
3 seconds

techservicesillinois/authman ✓
2 weeks ago
4 minutes, 34 seconds

techservicesillinois/terraform-aws-ssm-parameter ✓
4 weeks ago
4 seconds

techservicesillinois / authman › 156

Branch: develop
Commit: 062cf235
Author: ddriddle

1 month ago
4 minutes, 46 seconds

Updated to tier/grouper:2.4.0-a65-u38-w6-p8-20190722-rc1

clone

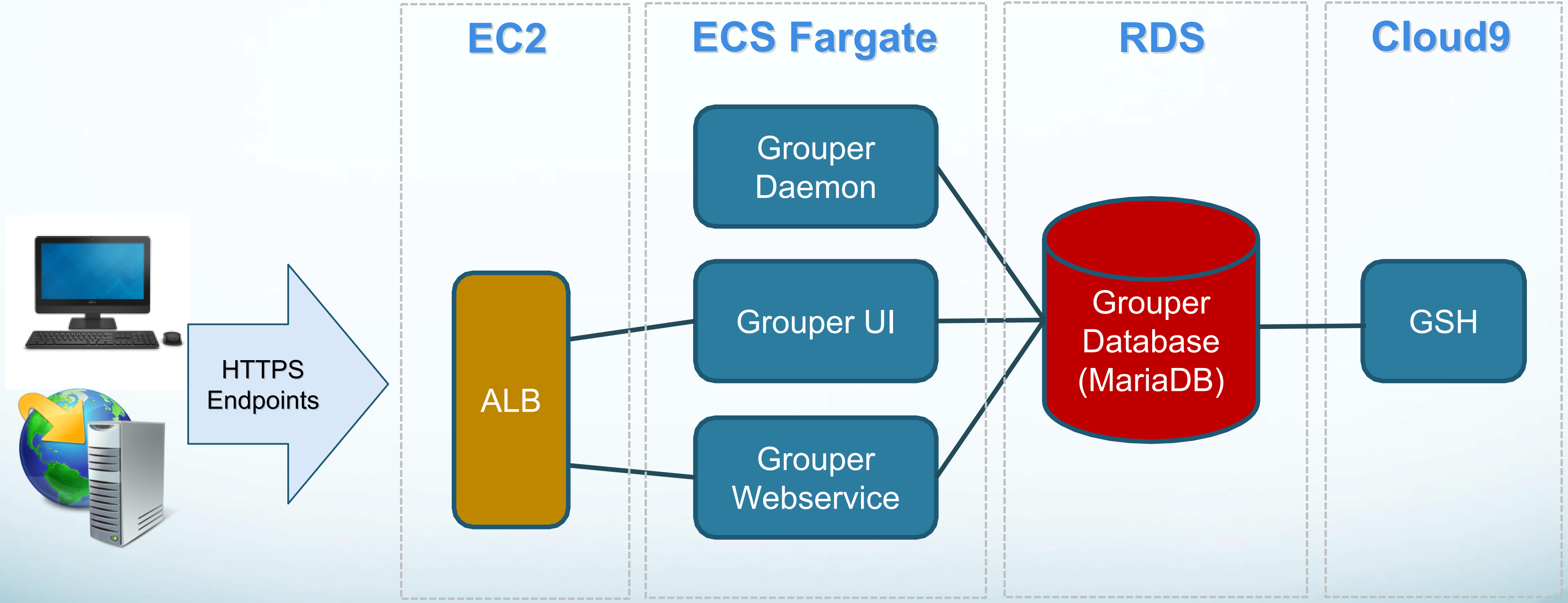
```
+ git init                                                                                  0s
Initialized empty Git repository in /drone/src/github.com/techservicesillinois/authman/.git/  0s
+ git remote add origin https://github.com/techservicesillinois/authman.git                  0s
+ git fetch --no-tags origin +refs/heads/develop:                                            0s
From https://github.com/techservicesillinois/authman                                         0s
 * branch            develop    -> FETCH_HEAD                                                0s
 * [new branch]      develop    -> origin/develop                                            0s
+ git reset --hard -q 062cf2351c3a649e48c7a64eead73c116864d68b                               0s
+ git submodule update --init --recursive                                                    0s

exit code 0
```

ecr

```
+ /usr/local/bin/docker daemon -g /var/lib/docker                                            1s
+ /usr/local/bin/docker version                                                              2s
Client:                                                                                      2s
 Version:      1.12.5                                                                        2s
 API version:  1.24                                                                          2s
```

Grouper Architecture



Layering the Grouper Image

- Baseline Trusted Access Platform Grouper Image
  - Customized configuration files
    - Remove HTTPS components
      - Import Intermediate Certificates for LDAP
- All Grouper nodes share same source image
- Entrypoint script determines node type with environment variable set by JSON

Grouper External Dependencies

- Active Directory DCs available in AWS VPC
  - Subject source
  - Provisioning target
- Shibboleth as authentication provider
- Amazon RDS (MariaDB) for backend database

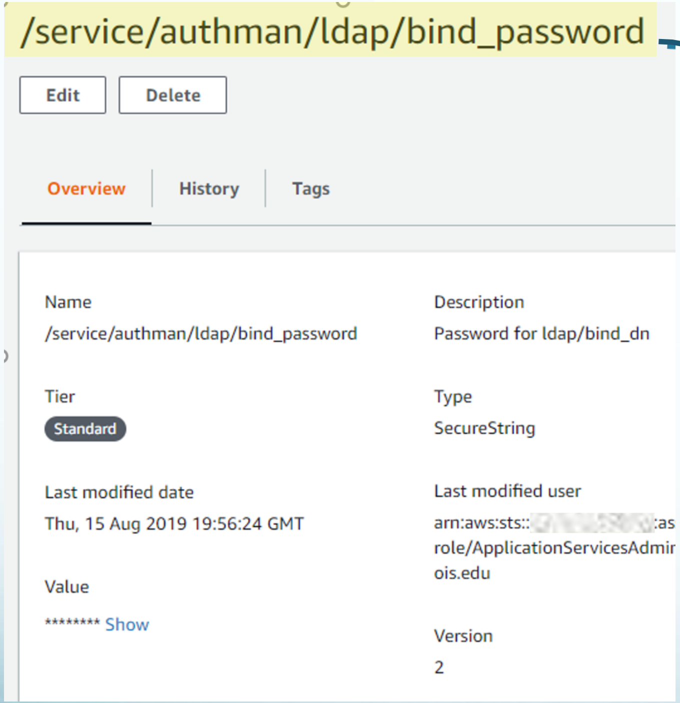
Some “Gotchas”

- Storage of Secrets
- Logging
- Back-end networking (VPCs)
- Where to declare the environment variables
- ECS container sizing
- Network security (AWS Security Groups)

Storage Considerations – Secrets

- Bad idea: Store passwords in Github
- Good idea: Store passwords in S3
- Great idea: Use AWS SSM Parameter Store
- Secrets can also be stored in Drone and built into the image

Storage Considerations – Secrets



/service/authman/ldap/bind_password

Name
/service/authman/ldap/bind_password

Description
Password for ldap/bind_dn

Tier
Standard

Type
SecureString

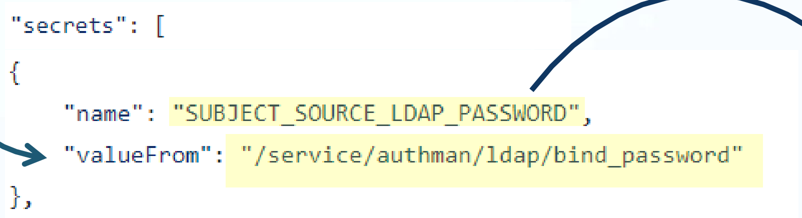
Last modified date
Thu, 15 Aug 2019 19:56:24 GMT

Last modified user
arn:aws:sts::[redacted]:as
role/ApplicationServicesAdmir
ois.edu

Value
******** Show

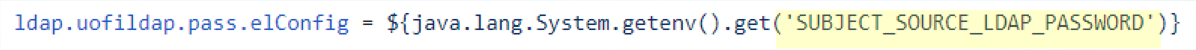
Version
2

containers.json:



```
"secrets": [
{
    "name": "SUBJECT_SOURCE_LDAP_PASSWORD",
    "valueFrom": "/service/authman/ldap/bind_password"
},
```

grouper-loader.properties:

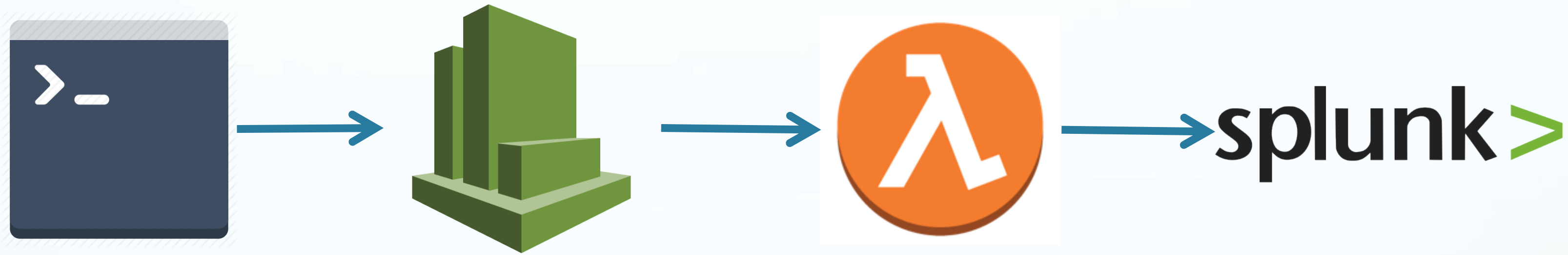


```
ldap.uofildap.pass.elConfig = ${java.lang.System.getenv().get('SUBJECT_SOURCE_LDAP_PASSWORD')}
```

Storage Considerations – Logs

- Problem: Application logs reside in virtual storage within ephemeral container, which will disappear
- Solution: Ship the logs out of the container
  - Logs piped to console output
  - Logs captured in AWS CloudWatch
  - Lambda Function imports Cloudwatch and pushes into Splunk

Storage Considerations – Logs



App logs piped to stdout

Cloudwatch Events Recorded

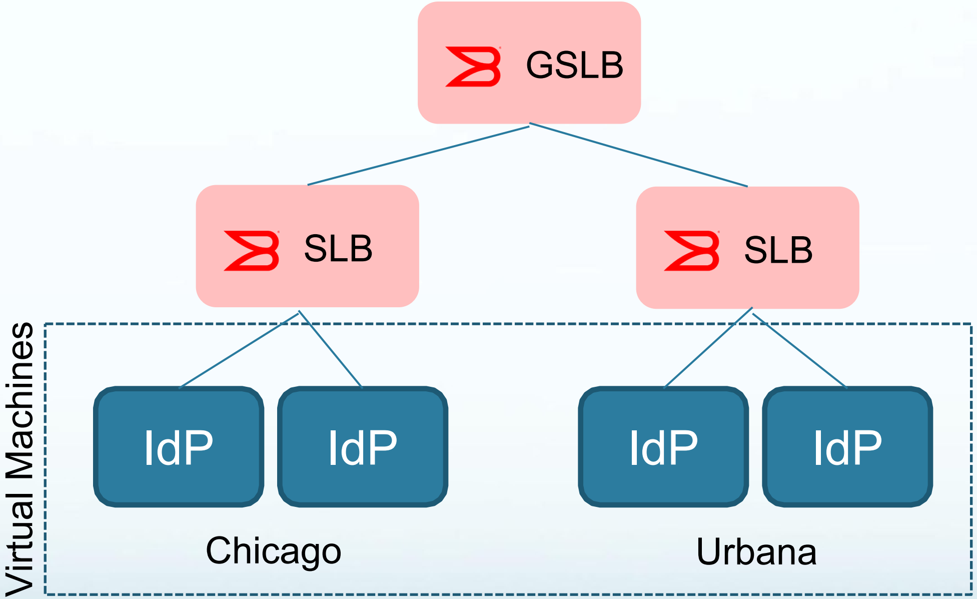
Lambda Function Processes and Forwards

HTTP Event Collector Ingests

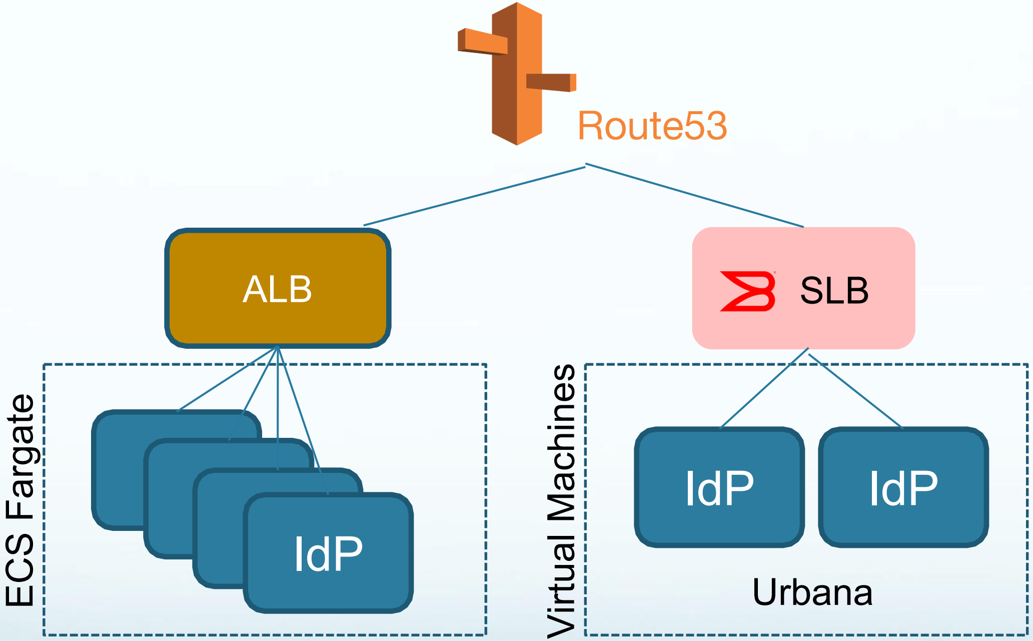
Shibboleth Current setup

- IdP nodes running on local virtual machines
- Global load balancing: primary cluster on campus, hot spare cluster remote
- MySQL cluster for consent storage
- Authentication and attribute stores from AD and IBM LDAP
- All dependencies globally load balanced

Shibboleth Architecture Diagram



Shibboleth Architecture Diagram



Layering the Shibboleth Image

- Baseline Trusted Access Platform IdP Image
  - Remove HTTPS Support
    - Download and Install Geant OIDC Extension
      - Import configuration
        - Download current federation metadata
          - Import UI templates

Shibboleth External Dependencies

- Kerberos keytabs are associated with specific IdP nodes as clients; still working on this one
    - Could consider LDAP authN, but not desired
- Amazon RDS (MySQL) for backend database
    - Consent storage data
    - Eliminating on-premise consent storage
- LDAP access dependent on node location

Shibboleth Metadata Considerations

- Metadata
  - Pulled in at IdP start-up…
  - But in case that fails, a local copy will be added to the image at build time
- Per-entity metadata (MDQ) solves this problem

Shibboleth Data Sealer Considerations

- "The biggest challenge"
- Timely rotation and copy to all running IdP nodes
- Future support for scripted key retrieval
  - Script the web service call to retrieve this from Secret Manager
  - Or used for any other storage of data sealer key that can be scripted
  - Key cached in memory as fallback

Shibboleth Data Sealer Considerations

- Until ready, alternatives include:
- Daily automated rolling IdP cluster restart
  - Retrieve the key on container start-up
  - During rolling restart, some nodes may be briefly out of sync
- A cron job inside each container that retrieves the key from Secret Manager
- Either of these solutions require building the .jceks file to store the current and previous keys

Shibboleth Scalability and redundancy

- We can do far better than our two active IdP nodes
- Smaller instances; one or two when usage is low
- ECS can dynamically spin up more at peak times or when load increases
- Application Load Balancer ties them together
- Load balanced clusters can span data centers for redundancy

Bonus Content: Shibboleth in a Box

- Goal: Build a "better" image of the SP
- Solution: Shibboleth as a micro-service
- Eliminates the need for:
  - Burning the SP into every image
  - Burning Apache as proxy
- Multi-node service containers talk to single SP node
  - Easier horizontal scaling
  - Reduces application container image size, lowers cost
  - Heavy authentication traffic? Add a 2nd Shib in a Box container

What's in the Box?

- Image contains the Shib SP, Apache, API
- Full login and logout capabilities supported
- Uses SP 3.X's session recovery functionality to allow for multiple cluster SP nodes
- Uses Secret Manager to rotate and distribute session recovery sealer key
- Illinois-built
- Available soon as open-source

For More Information

- Incommon Trusted Access Platform
  https://www.incommon.org/trusted-access/
- Incommon Collaboration Success Program
  https://www.incommon.org/academy/csp/
- AWS ECS Fargate
  https://aws.amazon.com/fargate/
- Terraform - https://www.terraform.io/
- Drone – https://drone.io/

2019 Technology Exchange

https://meetings.internet2.edu/2019-technology-exchange/

December 9-13, 2019
New Orleans, Louisiana

CAMP – Two days of campus case studies and key identity management issues

AdvanceCAMP – The premier forward-looking meeting with international IdM thought leaders

Join Us!

Upcoming Training

https://www.incommon.org/academy/library/

Shibboleth Installation Workshop

Denver, Colorado - October 22-23, 2019

Grouper School

November 12-13, 2019 – Philadelphia, Pennsylvania

COmanage Class

November 12-13, 2019 – Philadelphia, Pennsylvania

midPoint Basics

December 3-4, 2019 – Online

Please evaluate today’s session

https://www.surveymonkey.com/r/IAMOnline-Sep2019