




*Virtual Design Master*

## CHALLENGE 2

AN OUNCE OF PREVENTION

ADAM POST

ADAMJPOST@GMAIL.COM

@SEMI\_TECHNICAL 

## Table of Contents

Security Briefing .....	3
Overview .....	3
Intended Audience .....	3
Event Summary .....	3
Office Location .....	3
HumanityLink Distributed Modeling .....	3
HumanityLink Fleet Management.....	3
Root-Cause Analysis .....	4
Key Findings .....	5
Malware.....	5
Configuration .....	5
Design.....	5
Proposed Remediation .....	6
Network.....	6
Storage.....	6
Host .....	6
Management.....	6
Process .....	6
Scope and Requirements.....	7
Project Scope.....	7
Project Parameters .....	7
Requirements .....	7
Constraints .....	7
Assumptions.....	7
Risks .....	7
Technical Requirements.....	8
Security.....	8
Recoverability.....	8
HumanityLink Distributed Modeling .....	9
Security Remediation .....	9
Summary .....	9
Conceptual Diagram .....	10
Design Decisions.....	11
Configuration Adjustments .....	12

Physical Diagram.....	14
Disaster Recovery.....	15
Summary .....	15
Physical Configuration .....	15
Disaster Recovery Approach .....	16
Recovery Plan.....	17
HumanityLink Fleet Management.....	18
Security Remediation .....	18
Summary .....	18
Conceptual Diagram .....	19
Design Decisions.....	20
Configuration Adjustments .....	21
Physical Diagram.....	23
Disaster Recovery.....	24
Summary .....	24
Physical Configuration .....	24
Disaster Recovery Approach .....	25
Recovery Plan.....	26
Outbreak Recovery Plans .....	28
HumanityLink Distributed Modeling .....	28
HumanityLink Fleet Management.....	29
Process Improvements .....	30
Recurring Configuration Review .....	30
Review Security Groups.....	30
Review Network ACLs.....	31
Review Bucket Permissions .....	32
Review AMI Installed Packages.....	32
Change Control.....	33
Assess.....	33
Plan .....	33
Authorize.....	33
Implement.....	33
References.....	34

# Security Briefing

## Overview

Directly following rollout of the new HumanityLink 2.0 application and infrastructure architecture, a severe outage was experienced impacting production operations of Distributed Modeling and Fleet Management applications. This impact resulted in complete work stoppage across excavation sites waiting for work packages to be produced, setting back the excavation initiative and causing concerns regarding the infrastructure design. Throughout this period, maintenance and administrative functions were also unavailable.

As a result of these outages and their associated impacts, HumanityLink executive staff have commissioned an urgent follow-on project to address items found during event root-cause analysis. These items include missing security policy in several key areas, as well as a lack of change control and ongoing review. Absence of these measures allowed what might have been an isolated infection to become much broader in scope and effect. This effort will be focused on preventing recurrence.

## Intended Audience

This document is intended to educate HumanityLink business executives and technical staff on the topic of infection root cause, security design of the infrastructure, and recommended remediation steps.

## Event Summary

A high-level sequence of events related to the spread of infection and creation of the service outage is listed below. Due to matching security conditions, degradation proceeded in parallel across both environments, resulting in appearance of identical events.

### *Office Location*

- A member of the HumanityLink technical staff downloads an unauthorized package to their corporate workstation
- This workstation becomes infected and the malware in question begins searching the network for exploitable resources
- Communication to production is permitted via VPN. DR VPN connections are not online.

### *HumanityLink Distributed Modeling*

- HLDM nodes are attacked directly from the office location using an SMB exploit
- Application files on HLDM nodes are encrypted, resulting in failure of the provided service
- Infection spreads laterally through the app tier leveraging unrestricted communication
- Object storage buckets are accessed from infected servers, encrypting the contents and preventing immediate use of this dataset in DR.

### *HumanityLink Fleet Management*

- HLFM application servers are attacked directly from the office location using an SMB exploit
- Application files on HLFM nodes are encrypted, resulting in failure of the provided service
- Object storage buckets are accessed by infected servers and content is encrypted, preventing immediate cutover to this dataset in DR

## Root-Cause Analysis

Following analysis of samples provided to security researchers, the malware at the root of the outage event was found to be a derivative of the ransomware EatBrains. This code establishes itself on a user's machine, typically through web browsing, opening of an email attachment, or installation of a suspicious executable.

The malicious code then leverages permissions of the logged in user and existing network connectivity to infect other machines on the network via exploit of the Server Message Block protocol. The source of this infection has been identified as a workstation located at a branch office with VPN access to the production AWS environments. No policy restricting access rights of this workstation to these networks was found. **(R02)**

Because server packages that provide SMB services were not included within the original design, this type of vector was not initially suspected. However, following a review of installed packages across the instance fleet and source AMI images, a SAMBA server was found to be deployed on all HLDM and HLFM application-tier systems.

This EatBrains variant was found to possess functional exploits applicable to this open-source SMB implementation, in addition to those utilized by aging versions of Windows. The presence of SMB services on HLDM and HLFM application servers, combined with this enhanced exploit capability, resulted in failure of those hosted services. **(R02)**

In observing the malware in action, it was discovered that data alteration has been extended beyond file shares to include object storage, as well. This is accomplished by scanning local DNS cache and looking for recent communication targets. Records corresponding to resources hosted in S3 (\*.s3.amazonaws.com) are probed to assess level of access. If sufficient rights are discovered, objects are downloaded, deleted, encrypted and re-uploaded, causing inaccessibility of the data in question. Security policy on S3 buckets granting full access to HLDM and HLFM IAM roles allowed this to take place without interference. **(R02)**

Network device infection usually associated with EatBrains has not been fully seen in this outbreak. The traditional hardware firewall located at the impacted branch was compromised, yet network configuration of the AWS VPC's themselves shows no trace of alteration. This finding suggests the malware is not yet capable of obtaining root AWS privileges, emulating the AWS VPC router, or altering routing tables.

In a fortunate development, tunnels to DR environments were not active at the time of infection, so instances located there were not exposed to the attack. Versioning on S3 buckets was also enabled, and it was possible to delete all newly created (encrypted) objects, disable production environment access and bring operations online in DR. This process went as outlined in the original design and was a success, despite the adverse conditions.

Because this vector and extension of capabilities has not previously been documented, HumanityLink will be working closely with security researchers in providing infected samples, as well as following up regarding the effectiveness of planned remediation steps. In this way, HumanityLink is making the best of this unfortunate situation.

## Key Findings

### *Malware*

- 1) The observed EatBrains variant was found to have new, evolved capabilities
  - a. Exploitation of open-source implementations of SMB
  - b. Discovery and modification of object storage resources (S3)
- 2) Phone home functions were non-operational due to a lack of internet gateway and routing in app tiers
- 3) Impact was seen to the physical network firewall, but VPC network resources were unaffected

### *Configuration*

- 4) SAMBA components were mistakenly deployed as part of the HLDM/HLFM AMI image
  - a. All HLFM and HLDM systems were subsequently deployed with this package present
  - b. Actual impact limited to HLFM application and HLDM processing tiers
  - c. HLFM web tier was not impacted by the attack due to absence of the SAMBA package and presence of correctly configured security groups
  - d. HLFM RDS instances are not vulnerable to these attacks due to lack of low-level access, and correct security group configuration was also present
- 5) Application-tier IAM roles were granted inappropriate access to S3 resources
  - a. HLDM and HLFM roles granted full access to all S3 buckets
  - b. This access allowed deletion of legitimate data and replacement with encrypted data
  - c. This change was found to have been made for testing purposes without authorization following initial implementation
- 6) Application-tier security groups were not correctly configured
  - a. Open access between application servers allowed the infection to spread laterally
  - b. This change was found to have been made for testing purposes without authorization following initial implementation
- 7) Network ACL's were not implemented governing VPN access to AWS subnets
  - a. Unrestricted access to VPC subnets was permitted from the infected location
  - b. Lack of NACL appears to be an omission during the implementation phase

### *Design*

- 8) API calls to AWS services, specifically those to S3, were not being logged
  - a. A mechanism for tracking API calls was not included in the solution design
  - b. No initial evidence of object tampering was seen aside from updated modification dates and inaccessible data
- 9) Centralized logging was not being performed at the host level
  - a. Hosts were configured to log to local destinations
  - b. As a result of infection, logs became encrypted and unusable

In summary, the combination of evolved malware capabilities, unauthorized changes and implementation oversights resulted in complete impact to the production HLFM and HLDM sites until DR could be invoked. Because intrusion detection and logging tools were not previously implemented, actively tracking the spread of the infection was not possible. Focus should be placed on correcting this visibility issue **(R02)**.

## Proposed Remediation

The remediation steps below are direct outputs from a review of the infection root-cause analysis. These corrections directly address the gaps found in configuration, design and process that led to service disruption. It is recommended to implement these measures as soon as possible. **(R01)**

### *Network*

- Replace impacted physical firewall, patch all others globally to remove vulnerability to EatBrains
- Implement AWS network ACL's, limiting access to instances over VPN connection
- Address misconfigured security groups to enhance instance-to-instance security
  - HLDM
    - Restrict traffic between Application nodes to ports required
  - HLFM
    - Restrict traffic between Application nodes to ports required

### *Storage*

- Using the principle of least-privilege, reconfigure all S3 bucket policies in alignment with original design
  - HLDM
    - Design uploads to Model input bucket
    - Scout uploads to Geo data bucket
    - HLDM reads from Model input and Geo buckets
    - HLDM uploads to Delta model bucket
    - Excavation supervisor reads from Delta model bucket
  - HLFM
    - Excavation supervisor uploads to Health input bucket
    - Repair supervisor downloads from Repair output bucket
    - HLFM node read from Health input bucket
    - HLDM node write to Repair output bucket

### *Host*

- Audit all images used to deploy servers in EC2 and remove all non-essential packages
- Discover and remove vulnerable packages from all surviving instances not impacted by this attack
- Deploy an IDS system to assist with detecting abnormal system behavior and data modifications

### *Management*

- Utilize AWS CloudTrail to track all API calls to S3 along with the requesting security principal **(R05)**
- Configure CloudWatch Logs and use to aggregate system logs from all deployed instances **(R05)**

### *Process*

- Dedicate a human resource to audits and reviews, make accountable for configuration state **(A02)**
- Institute a recurring audit of installed packages and services for deployed instances and AML's
- Define in detail a recovery plan for both HLDM and HLFM using existing DR environments **(R04)**
- Conduct daily review of instance logs and platform events within CloudWatch and CloudTrail **(R05)**
- Institute Change Control and make all network and security changes subject to this process **(R03)**

# Scope and Requirements

## Project Scope

This project aims to remediate deficiencies in configuration, design and process that resulted in the recent infection of the HumanityLink platform. Once technical corrections are made, organizational focus should be placed on integrating recommended process improvements (**R03, A03**).

These include institution of change control and recurring environment audits to avoid unauthorized changes and configuration drift. Long term, the most benefit will be derived from these actions.

Aside from noted configuration adjustments, there will be no modification of existing primary and disaster recovery site architecture, as those elements have performed as intended (**R06, C01**).

Project assumes the operations team will re-deploy impacted instances using a clean AMI in parallel to this project (**A01**). Any item not listed in the previous proposed remediation section is out-of-scope for this project.

## Project Parameters

### Requirements

<b>R01</b>	Remediate gaps in security found as a result of the attack root-cause analysis
<b>R02</b>	Determine extent of the attack, data loss and systems impacted
<b>R03</b>	Use a process-oriented approach instead of relying exclusively on technical solutions
<b>R04</b>	Produce a disaster recovery plan for the HLDM and HLFM environments
<b>R05</b>	Utilized managed or platform-integrated solutions, wherever possible
<b>R06</b>	Preserve simplicity and architecture of the original design

### Constraints

<b>C01</b>	Overall infrastructure design should remain intact without significant modification

### Assumptions

<b>A01</b>	Damage to the production HLDM and HLFM environments will be repaired via re-deployment of the application tiers using a clean image
<b>A02</b>	Management will dedicate financial and human resources as required to support this initiative
<b>A03</b>	Proposed process improvements will be integrated with existing IT governance processes

### Risks

<b>r01</b>	Failure of a human to perform a required manual task presents a security risk
<b>r02</b>	Automating a task that is not understood does not increase security awareness
<b>r03</b>	Not adopting proper change control practices will likely result in similar future events



## Technical Requirements

The technical objectives of this project include improvements in recoverability and security. As a result, other infrastructure qualities are not shown here.

### *Security*

<b>RS01</b>	Correct Security Group configuration and restrict traffic between components / tiers
<b>RS02</b>	Correct S3 bucket policy permissions for security principals used in the solution
<b>RS03</b>	Implement centralized logging to increase visibility into OS-level activities
<b>RS04</b>	Improve environment capabilities in the area of intrusion detection
<b>RS05</b>	Correctly implement Network ACL's restricting access over VPN to SSH only
<b>RS06</b>	Enforce change control for any modifications to security configuration
<b>RS07</b>	Institute ongoing configuration audits and log reviews for cloud platform and instances

### *Recoverability*

<b>RR01</b>	Provide a disaster recovery process for HumanityLink Distributed Modeling that leverages existing DR environment and design
<b>RR02</b>	Provide a disaster recovery process for HumanityLink Fleet Management that leverages existing DR environment and design
<b>RR03</b>	Provide an infected site recovery plan for the production HLDM environment
<b>RR04</b>	Provide an infected site recovery plan for the production HLFM environment

# HumanityLink Distributed Modeling

## Security Remediation

### *Summary*

To directly address the impact of technical and operational issues found, a number of improvements will be made to HumanityLink Distributed Modeling, including tightened Security Groups, audited System Images, deployment of IDS and configuration of centralized logging. Regular, recurring reviews of system configuration will also be instituted, and dedicated resources will be assigned to this task. Any changes to this configuration once finalized will require authorization via Change Control board.

### *Security Groups*

The previous configuration permitted all communication between HLDM nodes, which allowed the infection to spread across the tier. As part of this remediation effort, communication between HLDM nodes will be restricted to ports required.

### *Network Access Lists*

The source of the infection was found to be a remote workstation with unrestricted access to subnets in the VPC. Access lists will be configured to block non-SSH access to all subnets within the environment.

### *System Images*

Because infection rendered all active, production HLDM nodes inoperable, this tier must be re-deployed. Before this occurs, system images (AMI's) will be audited and all un-needed software will be removed. IDS and logging components will be integrated with these images to ensure they are present in all future instance deployments.

### *Intrusion Detection*

In order to meet a requirement defined by management to provide increased intrusion detection capabilities, a host-based IDS platform (OSSEC) will be deployed across all HLDM nodes. This solution will detect modifications to content and attributes of files present on the system.

OSSEC will be configured to log events locally on the instance. This will allow the logging solution to aggregate these logs and make them available in the centralized console. It is expected that assigned resources will tune this configuration as time goes on to reduce false alarms and improve solution effectiveness. Implementation alone will not suffice.

### *Centralized Logging*

This solution will leverage AWS CloudWatch as an aggregation point for system logs using the Logs agent. Each instance will be configured to deposit logs to this location, resulting in increased visibility into instance-level service and security events. CloudTrail will be leveraged to track API access to S3.

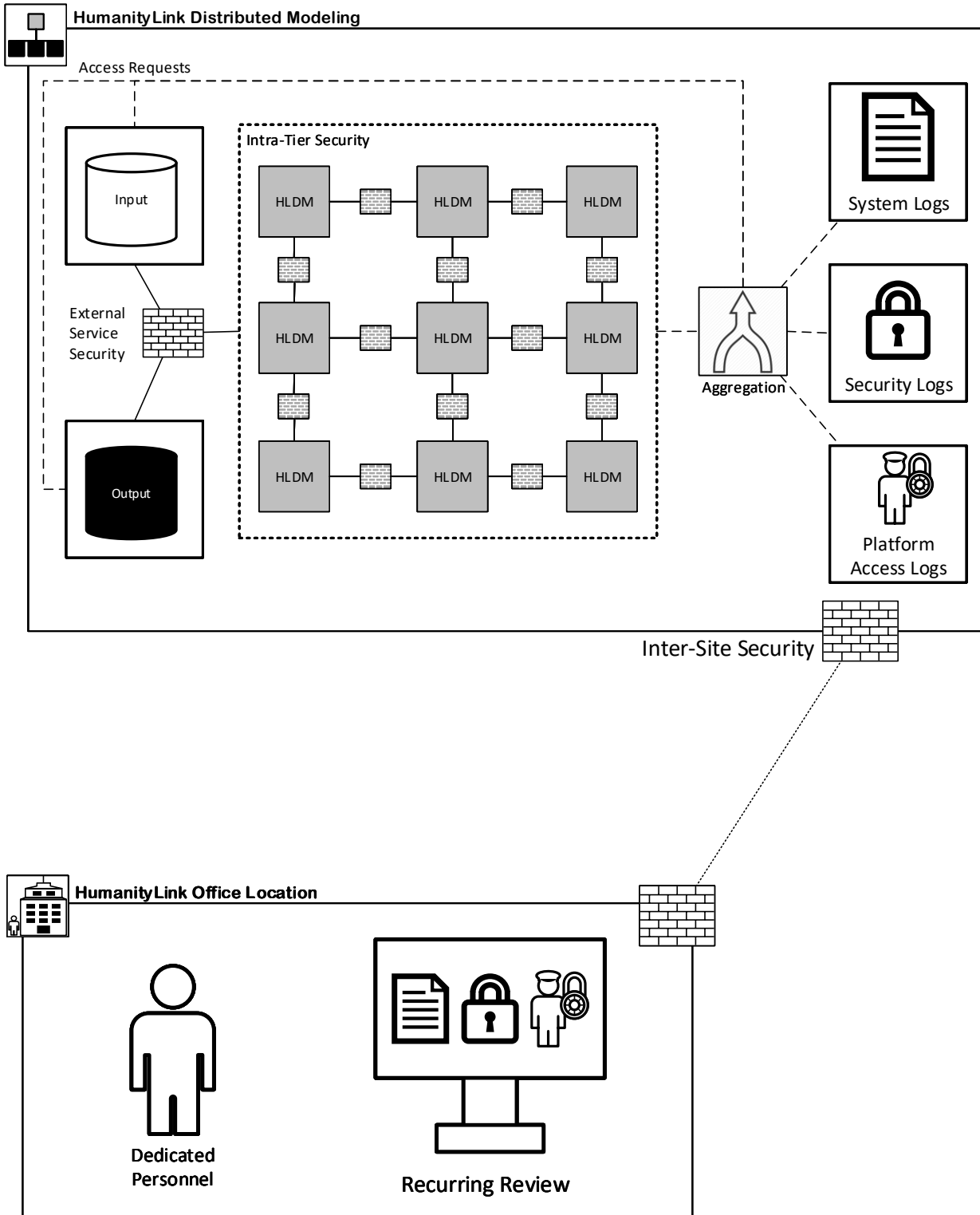
### *Recurring Review*

Resources will be assigned to conduct regular, recurring review of logs stored in CloudWatch and CloudTrail. This will promote general security awareness and ownership of environment security.

### *Change Control*

All changes to security or network configuration will be subject to change control, encouraging improved impact awareness and accountability for administrative actions taken in the environment.

## Conceptual Diagram



*HumanityLink Distributed Modeling points of security, review process and log aggregation.*

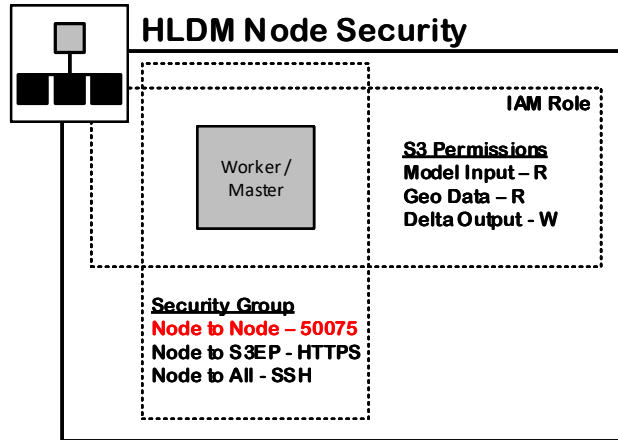
### Design Decisions

Area	Decision	Meets Requirement	Justification
Instance-Level Security	Security Group Corrections	<b>RS01</b>	Correcting security group configuration and performing ongoing audits will prevent unauthorized communication and malware spread
S3 Bucket Security	Bucket Policy Corrections	<b>RS02</b>	Properly restricting access to rights required prevents data alteration in alignment with security requirements.
S3 Logging	AWS CloudTrail	<b>R05 RS03</b>	CloudTrail will be configured to track access requests to S3 storage. This meets the defined logging requirement for S3
Instance Logging	AWS CloudWatch + Agent	<b>R05 RS03</b>	In conjunction with the OS-level agent, CloudWatch will serve as the system log aggregation and review point, meeting visibility requirements.
Host-Based Intrusion Detection (HIDS)	OSSEC	<b>R06 RS04 C01</b>	OSSEC meets the requirement of increased event visibility and aligns with the overall log aggregation approach. Tripwire does not generate real-time alerts.
HIDS Deployment Model	Manager	<b>R06 RS04 C01</b>	Manager installation will allow for local deposit of logs and relay of events to CloudWatch. Agent does not fit use-case
Network-Based Intrusion Detection	None	<b>RS04 C01</b>	Traditional Network IDS approaches are not available in AWS due to network abstraction. The visibility requirement can be met with HIDS.
VPN Security	NACL Addition	<b>RS05</b>	Network ACL's will be added for all subnets, limiting communication over VPN to SSH only. This will prevent non-administrative access.
IDS Logging Method	Local	<b>RS03 RS04</b>	OSSEC will be configured to generate logs and alerts locally. This content will be sent to CloudWatch via that agent and aggregated for team review.
Configuration Ownership	Dedicated resource(s) + recurring review	<b>RS07 A03 r01 r02</b>	Dedicating resources and assigning ownership of configuration and log review tasks improves security long-term
Unauthorized Configuration Avoidance	Change Control	<b>R03 RS06 A03</b>	All changes to network and security configuration will be subject to change control. This will increase accountability and prevent gaps from re-appearing.

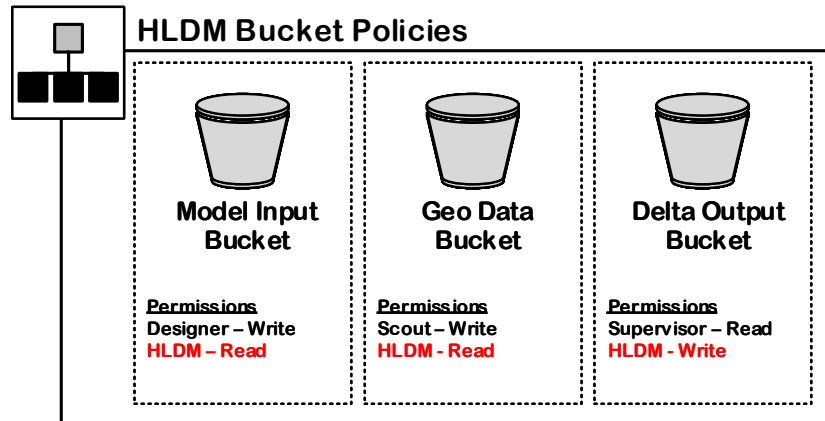
### Configuration Adjustments

The configurations below are the result of system audit and subsequent adjustment to bring the environment in alignment with stated recommendations.

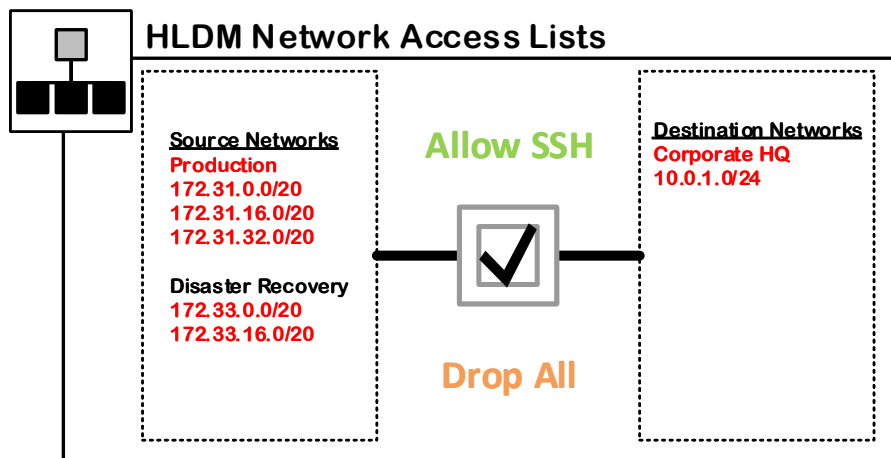
#### Adjusted Security Groups



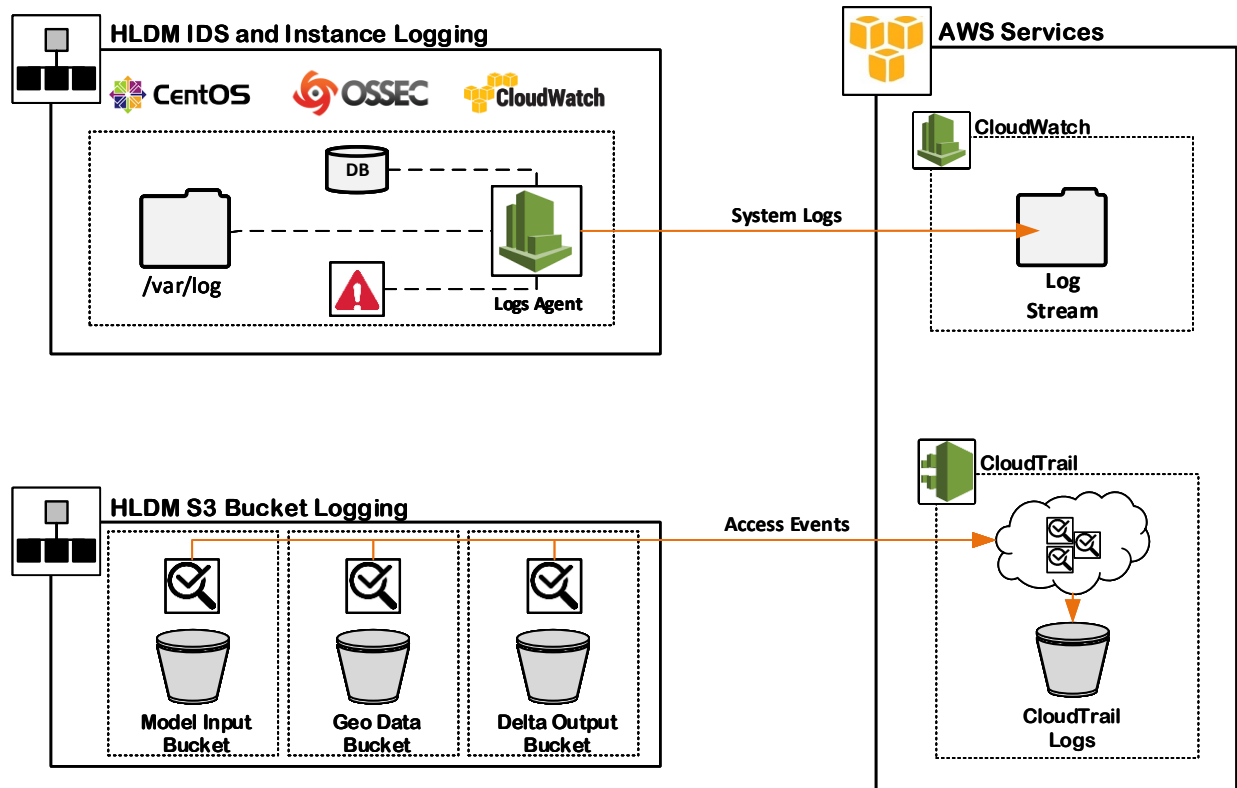
#### Adjusted Bucket Policies



#### New Access Lists

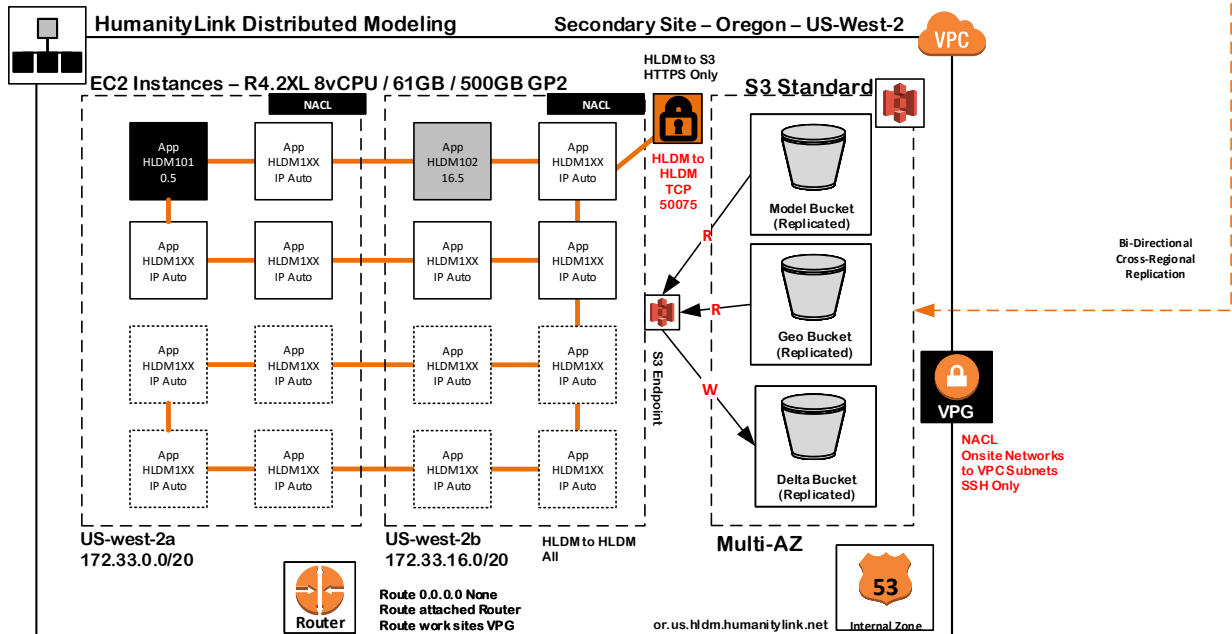
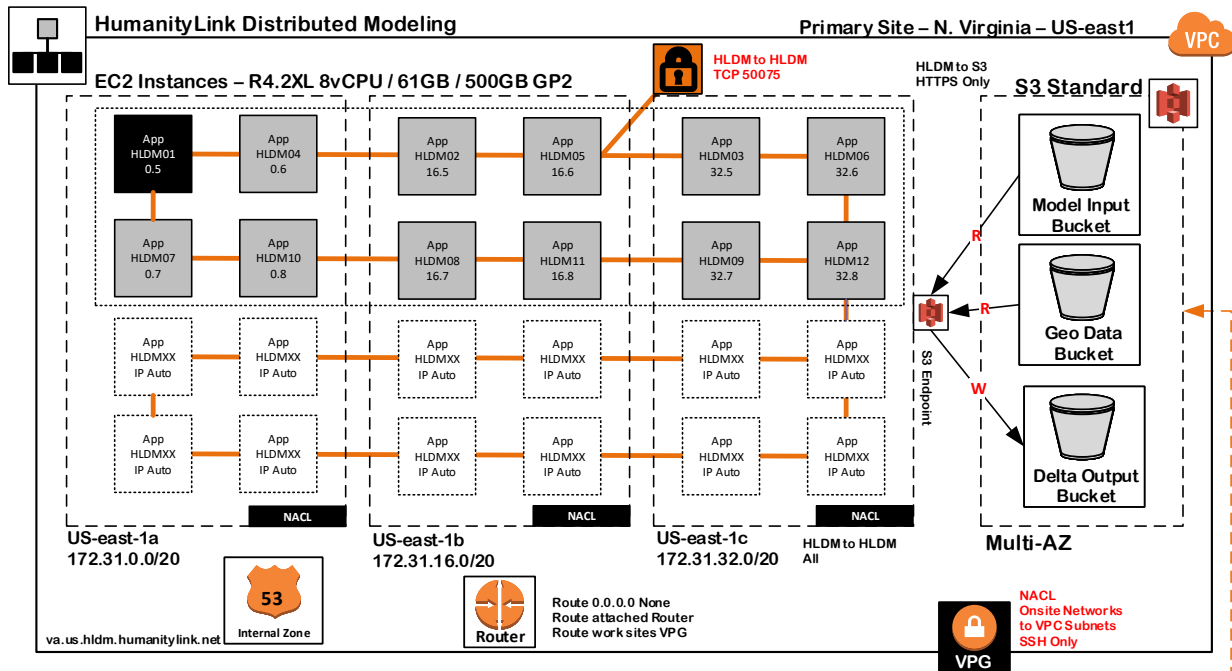


*Logging and Intrusion Detection Additions*



*Addition of OSSEC, CloudWatch Logs Agent and CloudTrail as supplemental security measures in HLDM*

## Physical Diagram



Updated physical diagram showing **addition** of recommended security groups, access lists, and correction of S3 permissions.

## Disaster Recovery

### *Summary*

In alignment with the original infrastructure design requirement, a fully-functional DR environment had previously been implemented. Although cutover to this environment was eventually successful during the outbreak, lack of clarity around specific procedure resulted in not meeting the defined recovery time objective. The sections below provide a summary of available resources, overall DR approach and specific procedure for cutover. Consult the updated physical diagram or [previous design document](#) for further details on layout and configuration of this environment.

### *Physical Configuration*

#### *Solution Capacity*

Area	Capacity (Min/Max)
vCPU	16 / 128
RAM	122GB / 976GB
Instance Storage Capacity	1TB / 8TB
Steady-State IOPS	3,000/24,000
Network Throughput	10Gbit Node-Node
Object Storage	Unlimited

#### *DR Instances*

Name	IP	Type	CPU	Memory	Disk 1	Disk 2	OS	Apps
HLDM101	172.33.0.5	R4.2XL	8	61	60	440	CentOS7	HLDM v1.0
HLDM102	172.33.16.5	R4.2XL	8	61	60	440	CentOS7	HLDM v1.0
HLDM103	172.33.0.6	R4.2XL	8	61	60	440	CentOS7	HLDM v1.0
HLDM104	172.33.16.6	R4.2XL	8	61	60	440	CentOS7	HLDM v1.0
HLDM105	172.33.0.7	R4.2XL	8	61	60	440	CentOS7	HLDM v1.0
HLDM106	172.33.16.7	R4.2XL	8	61	60	440	CentOS7	HLDM v1.0
HLDM107	172.33.0.8	R4.2XL	8	61	60	440	CentOS7	HLDM v1.0
HLDM108	172.33.16.8	R4.2XL	8	61	60	440	CentOS7	HLDM v1.0

#### *DR Networks*

Name	Network	Usable IP's
us-west-2a	172.33.0.0/20	4091
us-west-2b	172.33.16.0/20	4091



### *Disaster Recovery Approach*

The [original disaster recovery design](#) used the following approach to ensure production-equivalent capabilities were present in DR.

### *Architecture and Capacity*

The HLDM disaster recovery environment was created using the same architecture as the primary, with multiple compute nodes distributed across several availability zones for resiliency purposes.

Capacity was sized to exceed 50% of the primary environment when fully scaled. Should capacity needs be higher than these values, auto scaling parameters can easily be edited without any change to the architecture.

### *Network Connectivity*

An administrative VPN tunnel must be live in order for the disaster recovery process to be initiated. Because of this, the configuration has been pre-staged so this access is ready to utilize when needed. The HLDM DR master node need only be accessed from a physical location with this configuration present.

### *S3 Object Storage*

Core input and output functions performed by HLDM are facilitated through S3, and presence of this data is critical if operations are to resume in DR. As shown in the physical diagram, content of all buckets is replicated cross-region in bi-directional fashion to matching buckets in the DR environment.

If an item changes in the primary environment, that content will replicate to DR. If that item is changed during DR, it will be replicated once more and made available to the primary environment. In this way, seamless failover and failback is supported.

### *Notifications*

All components that subscribe to S3 object availability notifications receive updates for both primary and DR buckets. This ensures these downstream components always have a correct, functional location to retrieve their Excavation or Repair instructions from. This arrangement also supports seamless failover and failback.

### *HLDM Nodes*

A number of HLDM nodes remain running and on standby in the disaster recovery environment, but with processing services in a disabled state. Disaster recovery can quickly be invoked by verifying the primary environment is down, verifying presence of DR data, then starting master HLDM services via SSH connection. At this point, processing will resume using data in DR S3 buckets, downstream components will be made aware of output availability, and operations will be restored.

### *Recovery Plan*

Because the HLDM DR environment exists in a warm state with all required data being replicated continuously, the process needed to perform cutover is brief. This process is shown below, for reference. **(R04, RR01)**

#### *Bring VPN Connection Online*

- 1) Start a continuous ping from an administrative workstation to the HLDM master node in DR
  - a. Ping -t 172.33.0.5
- 2) Wait for a response from the master node
- 3) Initiate an SSH connection to the master node to verify connectivity
- 4) The VPN tunnel is now online

#### *Validate Object Storage*

- 1) Launch a web browser session to the AWS Management Console
  - a. <https://aws.amazon.com/console/>
- 2) Change the active region to US West (Oregon)
- 3) Navigate to S3 by selecting Storage > S3
- 4) Perform the following procedure for all buckets present
  - a. Click on the bucket name to view contents
  - b. Sort objects by Last Modified Date
- 5) Confirm that DR buckets contain recently replicated data
- 6) Object Storage validation is complete

#### *Test Notifications*

- 1) From the AWS Management Console, select Storage > S3
- 2) Select the **Delta model** output bucket
- 3) Upload a test object that aligns with the normal object naming convention
- 4) Monitor SNS notifications and ensure an event is triggered upon upload
- 5) Notification testing is complete

#### *Recover Application*

- 1) From within the AWS console, navigate to Compute > EC2
- 2) Verify all expected instances are present and running
  - a. Consult physical diagram or configuration table for details
- 3) Start an administrative SSH session to the master HLDM node
  - a. SSH 172.33.0.5
- 4) Start HumanityLink Distributed Modeling services
- 5) Application operations have been restored

# HumanityLink Fleet Management

## Security Remediation

### *Summary*

Several critical improvements will be made to HumanityLink Fleet Management as part of this effort. These include tightened Security Groups, audited System Images, deployment of IDS and configuration of centralized logging. Regular, recurring reviews of system configuration will also be performed, and dedicated resources will be assigned to this task. Any changes to this configuration once finalized will require authorization via Change Control board.

### *Security Groups*

The previous configuration permitted all communication between HLFM application servers, which allowed the infection to spread from node to node. Configuration following remediation will allow only ports required.

### *Network Access Lists*

The source of the infection was found to be a remote workstation. Access lists will be added to allow only SSH traffic from remote networks to bastion hosts.

### *System Images*

Because infection rendered all production HLFM application servers inoperable, this tier must be re-deployed. In preparation, AMI's must be audited and all un-needed software must be removed. IDS and logging components will be integrated with these images to ensure they are present in all future instance deployments.

### *Intrusion Detection*

A host-based IDS platform (OSSEC) will be deployed across all instances to provide increased intrusion detection capabilities. This solution will detect modifications to content and attributes of files present on the system. CloudWatch Logs can then be configured to generate alarms based on IDS activity.

### *Centralized Logging*

This solution will leverage AWS CloudWatch as an aggregation point for system logs using the Logs agent. Each instance will be configured to deposit logs to this location, resulting in increased visibility into instance-level service and security events.

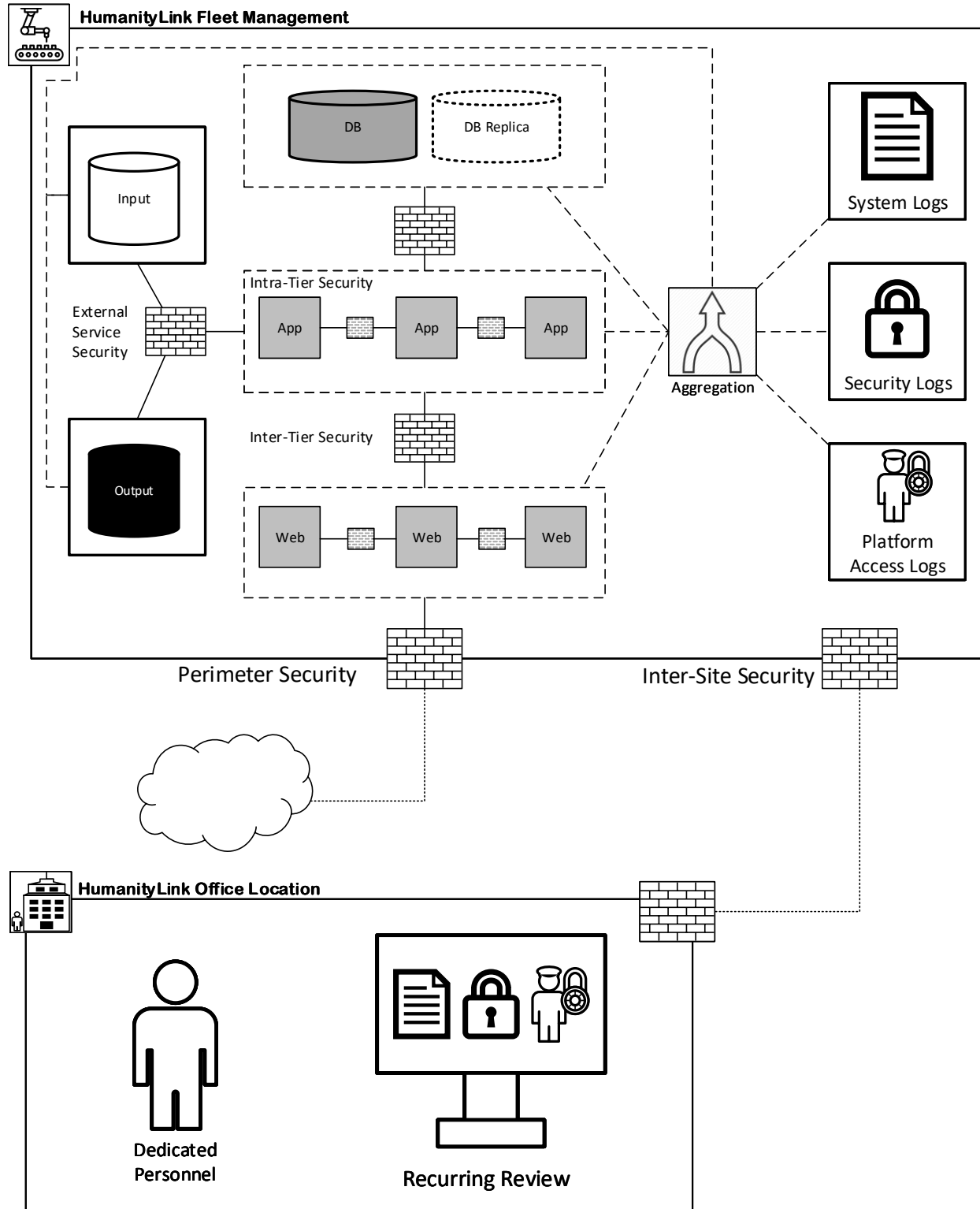
### *Recurring Review*

Resources will be assigned to conduct regular, recurring review of logs stored in CloudWatch.

### *Change Control*

All changes to security or network configuration will be subject to change control. This will maintain the integrity of the configuration and prevent creation of further issues.

### Conceptual Diagram



*HumanityLink Fleet Management points of security, review process and log aggregation.*

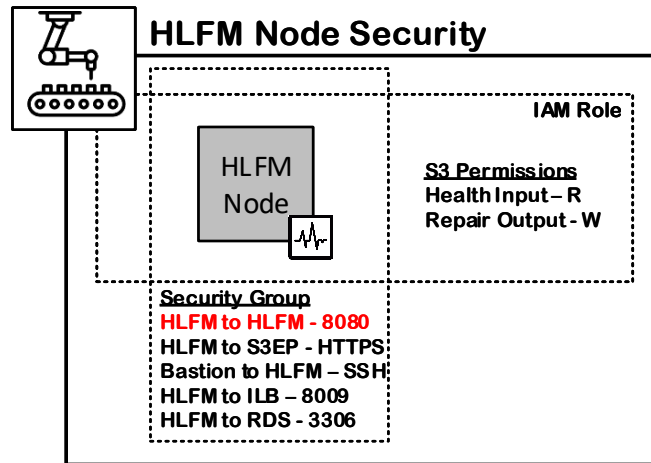
### Design Decisions

Area	Decision	Meets Requirement	Justification
Instance-Level Security	Security Group Corrections	<b>RS01</b>	Security groups will be corrected to restrict access between nodes in the application tier.
S3 Bucket Security	Bucket Policy Corrections	<b>RS02</b>	Restricting access rights to S3 will prevent unauthorized retrieval or deletion of HumanityLink data
S3 Logging	AWS CloudTrail	<b>R05 RS03</b>	CloudTrail will be configured to track access requests to S3 storage. This meets the defined logging requirement for S3
Instance Logging	AWS CloudWatch + Agent	<b>R05 RS03</b>	In conjunction with the OS-level agent, CloudWatch will serve as the system log aggregation and review point, meeting visibility requirements.
Host-Based Intrusion Detection (HIDS)	OSSEC	<b>R06 RS04 C01</b>	OSSEC meets the requirement of increased event visibility and aligns with the overall log aggregation approach. Tripwire does not generate real-time alerts.
HIDS Deployment Model	Manager	<b>R06 RS04 C01</b>	Manager installation will allow for local deposit of logs and relay of events to CloudWatch. Use of client/server model does not fit this use-case.
Network-Based Intrusion Detection	None	<b>RS04 C01</b>	Traditional Network IDS approaches are not available in AWS due to network abstraction. The visibility requirement can be met with HIDS.
VPN Security	NACL Addition	<b>RS05</b>	Network ACL's will be added for all subnets, limiting communication to bastions via SSH only. This will prevent non-administrative access.
IDS Logging Method	Local	<b>RS03 RS04</b>	OSSEC will be configured to generate logs and alerts locally. This content will be sent to CloudWatch via that agent and aggregated for team review.
Configuration Ownership	Dedicated resource(s) + recurring review	<b>RS07 A03 r01 r02</b>	Dedicating resources and assigning ownership of configuration and log review tasks improves security long-term
Unauthorized Configuration Avoidance	Change Control	<b>R03 RS06 A03</b>	All changes to network and security configuration will be subject to change control. This will increase accountability and prevent gaps from re-appearing.

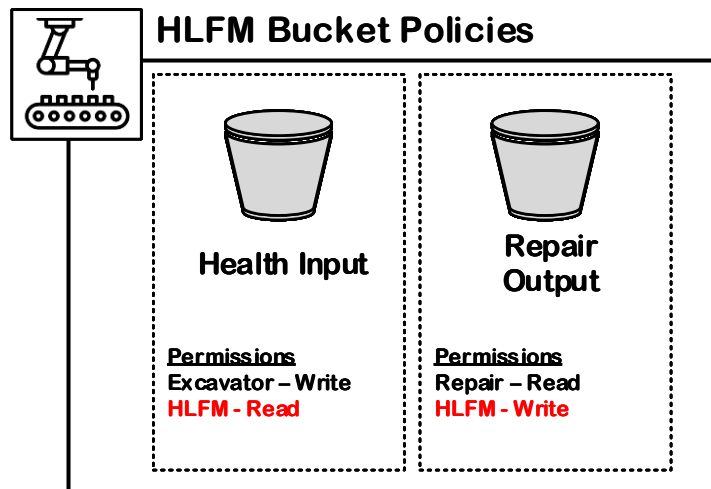
### Configuration Adjustments

The configurations below are the result of system audit and subsequent adjustment to bring the environment in alignment with stated recommendations.

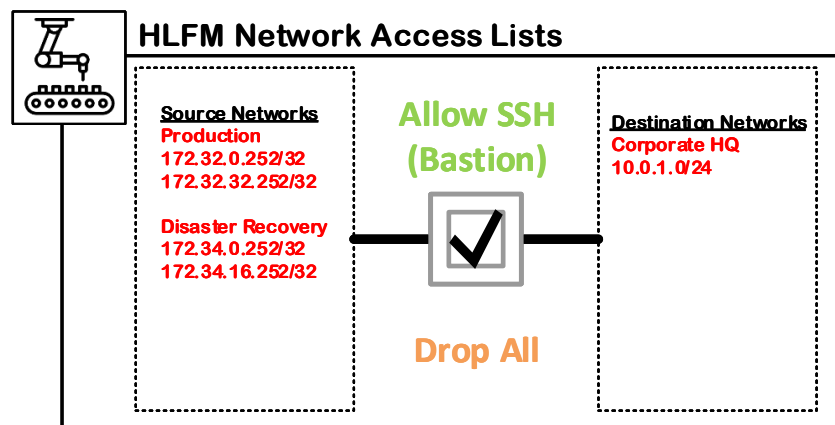
#### Adjusted Security Groups



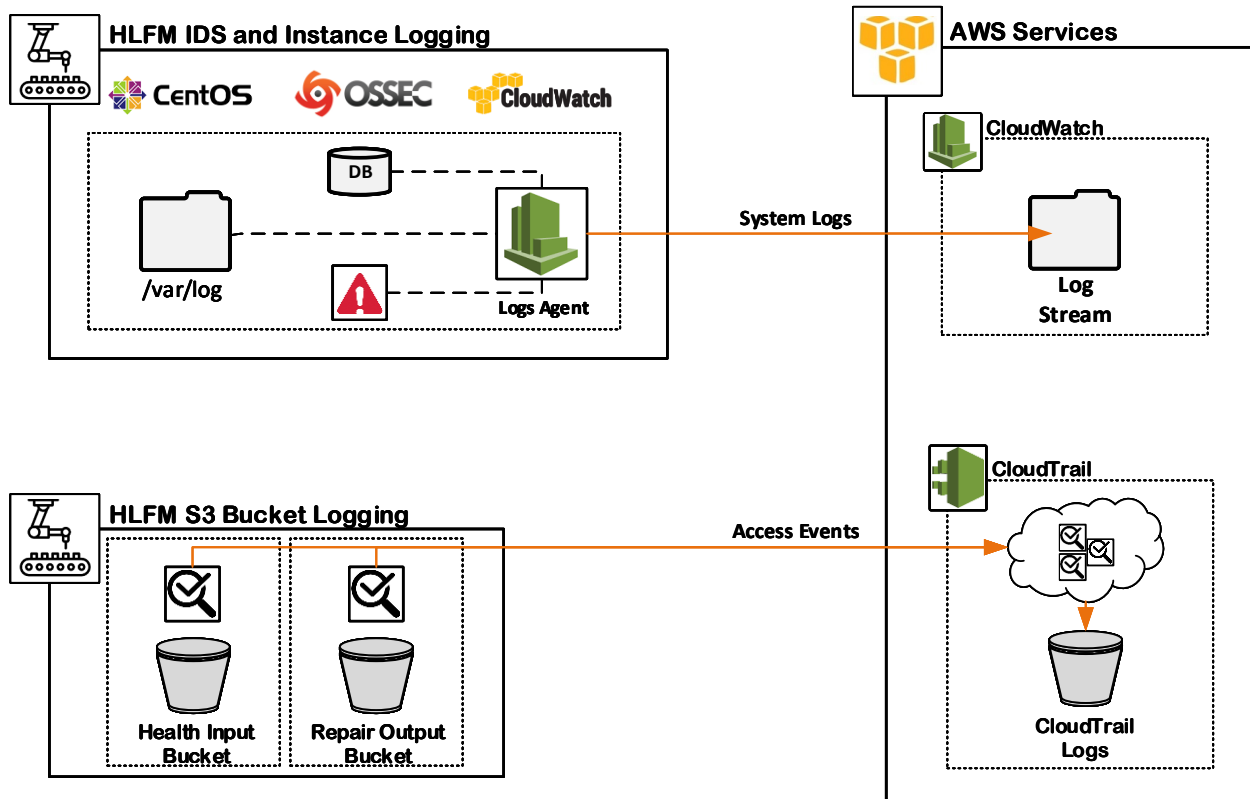
#### Adjusted Bucket Policies



#### New Access Lists

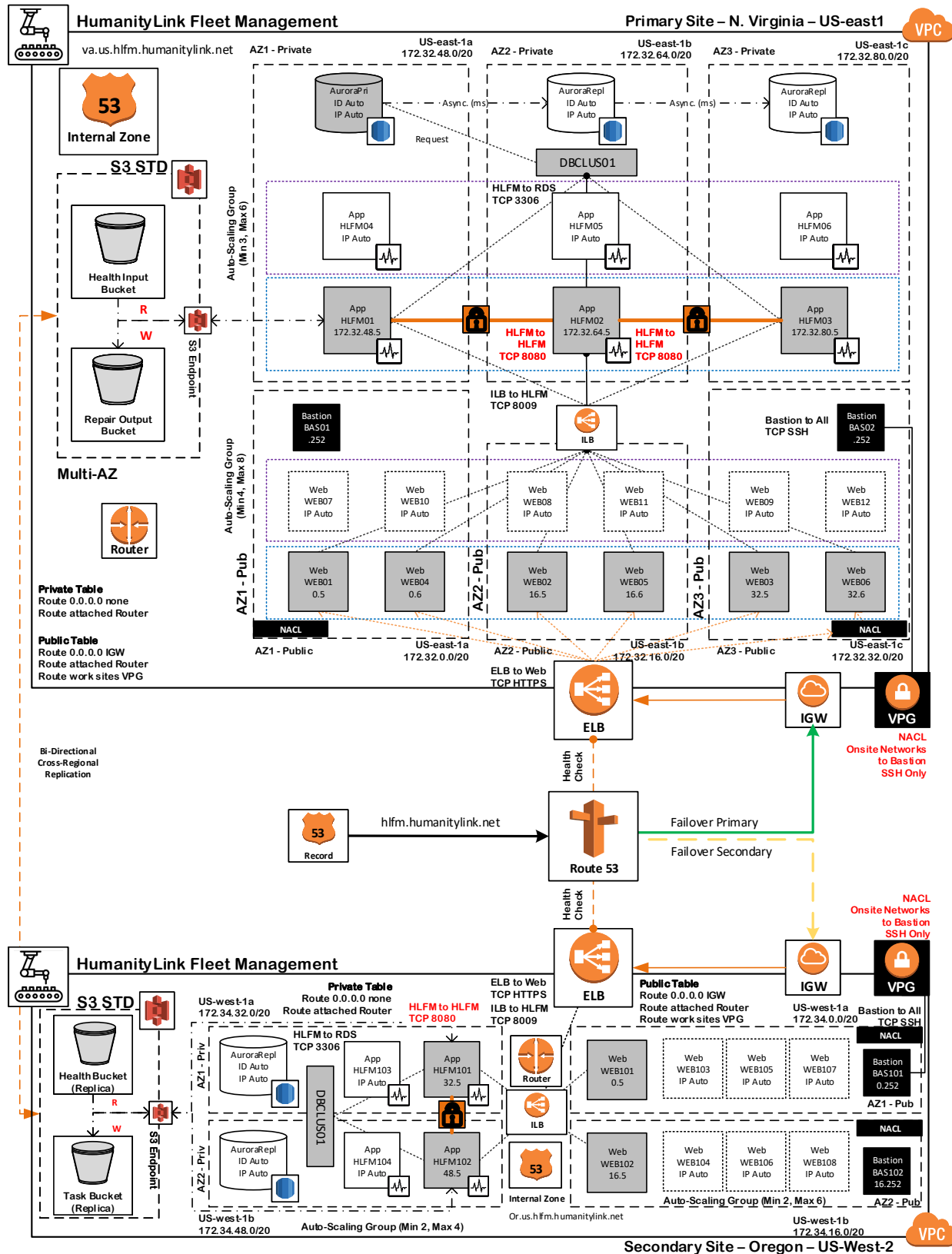


*Logging and Intrusion Detection Additions*



*Addition of OSSEC, CloudWatch Logs Agent and CloudTrail as supplemental security measures in HLFM*

## Physical Diagram



Updated physical diagram showing **addition** of recommended security groups and access lists



## Disaster Recovery

### *Summary*

As part of the [original HumanityLink 2.0 infrastructure design](#), a fully-functional disaster recovery environment was provided. During the outbreak, this environment allowed for restoration of HumanityLink Fleet Management services. However, the recovery time objective was not met and a defined process will be needed to prevent this situation from recurring.

The sections below provide a summary of the overall DR approach and specific procedure for cutover. Consult the updated physical diagram, configuration tables or previous design document for further details on layout and configuration of this environment.

### *Physical Configuration*

#### *DR Instances*

Name	IP	Type	CPU	Memory	Disk 1	OS	Apps
BAS101	172.34.0.252	T2.M	2	24	60	CentOS7	Lynx, SSH
BAS102	172.34.16.252	T2.M	2	24	60	CentOS7	Lynx, SSH
WEB101	172.34.0.5	M4.L	2	8	60	CentOS7	NGINX 1.13
WEB102	172.34.16.5	M4.L	2	8	60	CentOS7	NGINX 1.13
HLFM101	172.34.32.5	C4.2XL	8	15	100	CentOS7	HLFM 1.0
HLFM102	172.34.48.5	C4.2XL	8	15	100	CentOS7	HLFM 1.0
AuroraRepl3	Auto	DB.R3.2XL	8	61	1000	Managed	Aurora RDS
AuroraRepl4	Auto	DB.R3.2XL	8	61	1000	Managed	Aurora RDS

#### *DR Networks*

Name	Network	Usable IP's
pub us-west-2a	172.34.0.0/20	4091
pub us-west-2b	172.34.16.0/20	4091

Private Network	Network	Usable IP's
priv us-west-2a	172.34.32.0/20	4091
priv us-west-2b	172.34.48.0/20	4091

### *Disaster Recovery Approach*

The original disaster recovery design used the following approach to ensure production-equivalent capabilities were present in DR.

### *Architecture and Capacity*

The HLFM disaster recovery environment was created using the same architecture as the primary. This consists of a web tier, application tier, and database tier with load balancing being performed between them.

Object storage is used to provide a medium for input and output transmission, with SNS being used for notifications between components. External DNS hosting is handled using Route 53.

Capacity has been sized to exceed 50% of the primary once scaled. Should capacity needs be higher than these values, auto scaling parameters can easily be edited without any change to the architecture.

### *Network Connectivity*

An administrative VPN tunnel must be live in order for bastion hosts to be accessed from outside locations. These hosts serve as an access point where tiers can be validated in sequence before manual cutover. All authorized locations have this configuration pre-staged in case VPN access is needed.

### *S3 Object Storage*

Core input and output functions performed by HLFM are facilitated through S3, and presence of this data is critical if operations are to resume in DR. Content of all buckets is replicated cross-region in bi-directional fashion to matching buckets in the DR environment, supporting failover and failback.

### *Aurora Database Instances*

Application servers are backed by a pair of Aurora RDS replicas, which contain configuration information and historical fleet data. These replicas are always up to date and one can be promoted to primary, when needed. In full site failure this is automatic, and a manual process has been provided below.

### *HLFM Application Servers*

Several application servers reside in two private subnets and perform Fleet Management functions, including health analysis and actions supporting the web portal. An update of configuration files and test of database connectivity is all that is required to make this tier active in DR.

### *Web Servers*

Multiple web servers are present in two public subnets to provide a frontend for HLFM. These are dependent on the HLFM tier and internal/external load balancers. Because configuration management handles the state of these servers, no action is required during failover aside from basic testing.

### *Notifications*

All components that subscribe to S3 object availability notifications receive updates for both primary and DR buckets. This ensures these downstream components always have a correct, functional location to retrieve their instructions from. This arrangement also supports seamless failover and failback.

### *DNS*

External DNS is provided by Route 53 with records associated with each site arranged in a failover configuration. Full site failure results in automatic failover, but a manual process has been provided should this need to be invoked in another situation.

### *Recovery Plan*

Because the HLFM DR environment exists in a warm state with all required data being replicated continuously, the process needed to perform manual cutover is not extensive.

In the event of full region failure which impacts all tiers of the application, cutover to the DR environment will be handled automatically using a combination of SNS and Route 53.

The manual process, which can be invoked in other situations, is shown below for reference. **(R04, RR02)**

### *Bring VPN Connection Online*

- 1) Start a continuous ping from an administrative workstation to the primary bastion in DR
  - a. Ping -t 172.32.0.252
- 2) Wait for a response from the bastion host
- 3) Initiate an SSH connection to the bastion host to verify connectivity
- 4) The VPN tunnel is now online

### *Validate Object Storage*

- 1) Launch a web browser session to the AWS Management Console
  - a. <https://aws.amazon.com/console/>
- 2) Change the active region to US West (Oregon)
- 3) Navigate to S3 by selecting Storage > S3
- 4) Perform the following procedure for all buckets present
  - a. Click on the bucket name to view contents
  - b. Sort objects by Last Modified Date
- 5) Confirm that DR buckets contain recently replicated data
- 6) Object Storage validation is complete

### *Test Notifications*

- 1) From the AWS Management Console, select Storage > S3
- 2) Select the **Repair output** bucket
- 3) Upload a test object that aligns with the normal object naming convention
- 4) Monitor SNS notifications and ensure an event is triggered upon upload
- 5) Notification testing is complete

#### *Validate and Activate Aurora Replica*

- 1) Check replication status and lag
  - a. From the AWS Management Console, navigate to CloudWatch
  - b. Locate the Aurora Database Cluster
    - i. Expand system metrics and locate the ReplicaLag metric
    - ii. If this value is under a few minutes, proceed with cutover
- 2) Promote a DR Aurora replica to primary
  - a. From the AWS CLI, run the following commands
    - i. Run “describe-db-instances”
    - ii. Locate desired replica to promote
    - iii. Run “promote-read-replica --db-instance-identifier <value>”
- 3) Verify status of promoted Aurora replica
  - a. Run “describe-db-instances --db-instance-identifier <value>”
  - b. Verify instance state shows master
  - c. Run “aws rds describe-db-clusters --region us-west-2a”
  - d. Verify Status is Active and that the appropriate instance is set to “IsClusterWriter”
- 4) Attempt an administrative connection to the database using normal MySQL tools
- 5) Database cutover is complete

#### *Check HLFM Application Services*

- 1) Initiate a connection to both DR HLFM servers
  - a. 172.34.32.5
  - b. 172.34.48.5
- 2) Check running services and ensure HLFM components are online
- 3) Restore latest version of HLFM configuration files from production backups in S3
- 4) Attempt a database connection to the Aurora RDS instance
- 5) Application services are now online

#### *Test Web Servers*

- 1) Using Lynx on a bastion host, browse to the URL of one of the DR web servers
  - a. 172.34.0.5
  - b. 172.34.16.5
- 2) Verify site responds and presents expected information
- 3) Web services are now online

#### *Cutover DNS and external access*

- 1) Cut over External DNS records associated with HLFM
  - a. Log in to the AWS Management Portal
  - b. Select Networking > Route 53
  - c. Select hosted zone **hlfm.humanitylink.net**
  - d. Reverse record weights ensuring **or.us.hlfm.humanitylink.net** is now primary
  - e. External access to Humanity Link has been cut over

# Outbreak Recovery Plans

At the time of writing, all operations associated with HumanityLink 2.0 are being serviced from disaster recovery locations. Because of the matching architecture and correct sizing of those environments, no issues have been encountered since failover.

However, it is the desire of the business to cut operations back over to production sites in N. Virginia as soon as is feasible. The sections below outline tasks needed to resolve the outbreak at those locations and make those sites active once more.

## HumanityLink Distributed Modeling

Recovery plan corresponds to **RR03**. Remediation steps address **R01**.

Reference **Recovery Plan** in the DR section of this document for more detail on specific tasks needed to cut over HLDM.

- Schedule an outage window for performance of environment cutover
- Validate information in configuration management systems is valid and current for HLDM nodes
- Audit HLDM AMI and ensure recommended corrections have been made to the image
- Adjust HLDM inter-node security group to restrict traffic to TCP 50075
- Adjust bucket policies, restricting access according to remediation plan
- Configure CloudTrail to monitor Model, Geo and Delta buckets
- Configure CloudWatch and prepare to accept instance logs
- Implement network ACL for all subnets restricting outside access to SSH only
- Terminate (12) EC2 instances within the Production HLDM VPC corresponding to these nodes
  - HLDM01 - HLDM12
- Re-deploy (12) HLDM nodes from the new, audited HLDM AMI and validate configuration (**A01**)
- Configure OSSEC for local alerting and CloudWatch Logs Agent to report to CloudWatch
- Verify receipt of logs within the CloudWatch console
- Connect to the HLDM master node within the DR environment and stop HLDM services
- Validate that S3 data has replicated back to the corresponding primary buckets
- Connect to the HLDM master node within the production environment and start HLDM services
- Perform an object upload to the Model input bucket and verify a processing attempt is initiated
- Audit non-impacted instances in the DR environment to ensure absence of vulnerable packages
- Duplicate security group, bucket permissions and ACL adjustments in DR the environment

Recovery of HumanityLink Distributed Modeling within the previously infected primary site is complete.

## HumanityLink Fleet Management

Recovery plan corresponds to **RR04**. Remediation steps address **R01**.

Reference **Recovery Plan** in the DR section of this document for more detail on specific tasks needed to cut over HLFM.

- Schedule an outage window for performance of environment cutover
- Validate information in configuration management systems is valid and current for HLFM servers
- Audit HLFM AMI and ensure recommended corrections have been made to the image
- Audit non-impacted instances in the Web tier to ensure absence of vulnerable packages
- Adjust HLFM inter-node security group to restrict traffic to TCP 8080
- Adjust bucket policies, restricting access according to remediation plan
- Configure CloudTrail to monitor Health Input and Repair Output buckets
- Configure CloudWatch and prepare to accept instance logs
- Remove private routing table entry directing traffic to remote sites through the VPG
- Implement network ACL for all public subnets, restricting access to Bastion only via SSH
- Terminate (3) EC2 instances within the production HLFM VPC corresponding to these nodes
  - HLFM01 – HLFM03
- Re-deploy (3) HLFM nodes from the new, audited HLFM AMI and validate configuration (**A01**)
- Configure OSSEC for local alerting and CloudWatch Logs Agent to report to CloudWatch
- Verify receipt of logs within the CloudWatch console
- Connect to HLFM nodes in the DR environment and stop HLFM services
- Validate that S3 data has replicated back to the corresponding primary buckets
- Use procedure outlined in HLFM DR plan to make the Aurora instance in AZ1 active
- Check replication and overall Aurora cluster status within CloudWatch and AWS CLI
- Test administrative access to the database instance using MySQL tools
- Connect to each of the (3) HLFM nodes via SSH and synchronize application configuration files from S3 backups
- Ensure all HLFM services are started and in a healthy state on HLFM application servers
- Test database connection from HLFM to Aurora
- Browse to a web server in the local site, verify site is responsive and content is valid
- Make record **va.us.hlfm.humanitylink.net** primary and move Oregon to secondary
- Allow time for TTL expiry and test connectivity to HumanityLink Fleet Management externally.
- Audit non-impacted instances in the DR environment to ensure absence of vulnerable packages
- Duplicate security group, bucket permissions and ACL adjustments in the DR environment

Recovery of HumanityLink Fleet Management within the previously infected primary site is complete.

# Process Improvements

## Recurring Configuration Review

The outline below defines recommended high-level steps to assign to a technical resource for recurring execution (**R03, A02**). Once overall security consciousness has been raised with respect to the AWS infrastructure, some of these tasks may become candidates for automation (**r01, r02**). Until then, it is recommended that these tasks be executed manually.

### *Review Security Groups*

- Install [AWS CLI tools](#) on an administrative workstation
- Gather AMI credentials with privileges sufficient to read security group configurations
- Configure AWS CLI to use assigned access keys using **aws configure**
- Run the following command against each region with infrastructure present
  - **aws ec2 describe-security-groups**
- Analyze output and ensure rules are properly scoped from a port and IP perspective

```
"Description": "BastionToHLFM",
"Tags": [
  {
    "Value": "BastionToHLFM",
    "Key": "Name"
  }
],
"IpPermissions": [
  {
    "PrefixListIds": [],
    "FromPort": 22,
    "IpRanges": [
      {
        "CidrIp": "172.32.0.252/32"
      },
      {
        "CidrIp": "172.32.32.252/32"
      }
    ],
    "ToPort": 22,
    "IpProtocol": "tcp",
    "UserIdGroupPairs": [],
    "Ipv6Ranges": []
  }
],
"GroupName": "BastionToHLFM",
"VpcId": "vpc-928a08eb",
"OwnerId": "385431055526",
"GroupId": "sg-86a8c5f7"
```

*Output showing SSH allowed inbound as part of security group BastionToHLFM*

- Complete a review of the remainder of the defined security groups
- Save results to file, review and prepare for presentation to management

### Review Network ACLs

- Using the preconfigured AWS CLI session, run the following command
  - **aws ec2 describe-network-acls**
- Review output and ensure ACL's remain scoped according to the remediation plan

```
"NetworkAclId": "acl-0d013d74",
"VpcId": "vpc-928a08eb",
"Tags": [
  {
    "Value": "HLFMtoRemoteSites",
    "Key": "Name"
  }
],
"Entries": [
  {
    "RuleNumber": 101,
    "Protocol": "6",
    "PortRange": {
      "To": 22,
      "From": 22
    },
    "Egress": true,
    "RuleAction": "allow",
    "CidrBlock": "10.0.1.0/24"
  },
  {
    "RuleNumber": 32767,
    "Protocol": "-1",
    "Egress": true,
    "CidrBlock": "0.0.0.0/0",
    "RuleAction": "deny"
  },
  {
    "RuleNumber": 100,
    "Protocol": "6",
    "PortRange": {
      "To": 22,
      "From": 22
    },
    "Egress": false,
    "RuleAction": "allow",
    "CidrBlock": "10.0.1.0/24"
  },
  {
    "RuleNumber": 32767,
    "Protocol": "-1",
    "Egress": false,
    "CidrBlock": "0.0.0.0/0",
    "RuleAction": "deny"
  }
]
```

*Output showing configuration of NACL's allowing only SSH from remote locations to HLFM*

- Complete against each region with infrastructure present
- Save results to file, review and prepare for presentation to management



### Review S3 Bucket Permissions

- Using the preconfigured AWS CLI session, run the following command
  - **aws s3api get-bucket-acl --bucket BUCKETNAME**

```
C:\Program Files\Amazon\AWSCLI>aws s3api get-bucket-acl --bucket healthinput
{
  "Owner": {
    "DisplayName": "adam.post",
    "ID": "b66f90e0c6f6cf78f2693c6b3075ec63ebba755ff448529293b48647bdb47a81"
  },
  "Grants": [
    {
      "Grantee": {
        "Type": "CanonicalUser",
        "DisplayName": "adam.post",
        "ID": "b66f90e0c6f6cf78f2693c6b3075ec63ebba755ff448529293b48647bdb47a81"
      },
      "Permission": "FULL_CONTROL"
    }
  ]
}
```

*Access list for bucket Health Input showing full control for security principal adam.post*

- Repeat permissions gathering process for all other buckets present
- Compare results against S3 permissions design provided in remediation plan
- Save results and prepare for presentation to management

### Review AMI Installed Packages

Repeat the process below for each actively used AMI present in the region:

- Launch a new instance into a private subnet using the CentOS AMI in question
- Initiate an SSH connection to the new instance
- Run **yum list installed** within the remote session to view a list of installed packages

```
[centos@ip-172-32-2-43 ~]$ hostname
HLFM01.va.us.hlfm.humanitylink.net
[centos@ip-172-32-2-43 ~]$ yum list installed | grep samba
samba.x86_64                               4.4.4-14.el7_3                @updates
samba-client-libs.x86_64                  4.4.4-14.el7_3                @updates
samba-common.noarch                       4.4.4-14.el7_3                @updates
samba-common-libs.x86_64                  4.4.4-14.el7_3                @updates
samba-common-tools.x86_64                 4.4.4-14.el7_3                @updates
samba-libs.x86_64                         4.4.4-14.el7_3                @updates
```

*Identification of vulnerable SAMBA packages on HLFM01 using the outlined method*

- Review listed packages and ensure no unneeded or vulnerable items are present
- Save output to file, review and prepare for presentation to management

## Change Control

Of all recommended remediation steps, future use of Change Control is among the most impactful. Had configuration remained in its intended state, the infection likely would not have caused a severe impact.

It is highly recommended that institution of this process be treated as a high priority, as no technical solution can counter unauthorized misconfiguration (**R03, A03, r03**). A high-level overview of this process is provided below.

### *Assess*

If a configuration item in the environment requires adjustment, begin by defining the following areas:

- Overall scope of the change
- Identification of systems involved
- Scope of potential adverse impact
- Downtime requirements
- Potential timeframes for execution
- Technical contacts needed for execution or escalation purposes
- Business contacts that must be made aware of the change

By defining these areas, an engineer becomes more aware of the depth of work required and the potential scope of impact, should the procedure not proceed as expected.

### *Plan*

Work required to implement the change should be assembled into a formal plan, including all areas from the assessment phase and the items below:

- Clear identification of goals to be achieved as the result of the change
- Definition of phases to guide the assigned resource through execution
- Detailed procedural breakdown within each phase, including all steps and sub-steps
- **Important:** Detailed back-out plan that would result in restoration of the original configuration

Identifying clearly goals for the change, then researching and creating a detailed work plan results in superior, less error-prone work. Risk is also significantly mitigated by having all back-out step pre-defined, should reversal be required.

### *Authorize*

Once assessment and planning phases are complete, formal approval of the work plan should be obtained:

- Consult with senior technical resources and obtain approval on approach and procedure
- Submit the formalized plan for review and approval by a business decision-maker

### *Implement*

Proceed to implement the change as planned once authorization is obtained:

- Perform implementation according to the work plan without deviation or improvisation

Finally, review the outcome with management post-implementation to identify potential improvements to be made going forward.

## References

The following documentation was referenced in the creation of this solution design:

- [VDM S5 Challenge 1 – Adam Post – DR Designs](#)
- [AWS Marketplace - Intrusion Detection Systems for EC2 Instances](#)
- [AWS Blog – HIDS Monitoring with CloudWatch](#)
- [OSSEC Project – OSSEC Manual](#)
- [Tripwire Open Source vs OSSEC : Which Is Right For You?](#)
- [AWS Command Line Interface Documentation](#)
- [AWS VPC Documentation](#)
- [AWS S3 Documentation](#)
- [AWS RDS Documentation](#)
- [AWS Route 53 Documentation](#)