INCIDENT REPORT AWS MGN Agent Removal & Environment Cleanup

v1.0.0

Author:

Eldon Gabriel

March 4th, 2026

TABLE OF CONTENTS

REVISION HISTORY 2
0.0 VMware-to-AWS Migration Topology 3
0.1 Executive Summary 4
1.0: AWS MGN Agent Removal & Environment Cleanup 5
1.1 Incident Overview 5
1.2 Initial Symptoms 5
1.3 Impact Assessment 5
1.4 Technical Analysis: Multilayer Infrastructure Stack 6
1.4.1 Diagnostic & Remediation Steps 6
Phase 1: Windows Kernel Decommissioning 6
Phase 2: Network & Throughput Optimization 7
Phase 3: AWS Launch & Cloud Access Management 7
Phase 4: Out-of-Band (OOB) Troubleshooting 7
1.5 Root Cause 8
1.6 Resolution Implemented 8
1.7 Verification 8
1.8 Financial Monitoring & Cost Variance 8
1.9 Lessons Learned 10
1.10 Pattern Tracking 10

Disclaimer: This report documents my personal work completed during an independent lab exercise involving AWS Application Migration Service (MGN) and hybrid infrastructure troubleshooting. It reflects my own analysis, configuration, and remediation steps performed in a controlled lab environment. No proprietary training materials, instructor content, or restricted lab documentation have been reproduced or distributed. All technical explanations and findings are based on my independent work and follow the academic integrity and disclosure policies of the Mossé Cyber Security Institute (MCSI).

REVISION HISTORY

Version	Date	Author	Description of Changes
v1.0.0	03/04/2026	Eldon G.	Initial draft.

0.0 VMware-to-AWS Migration Topology

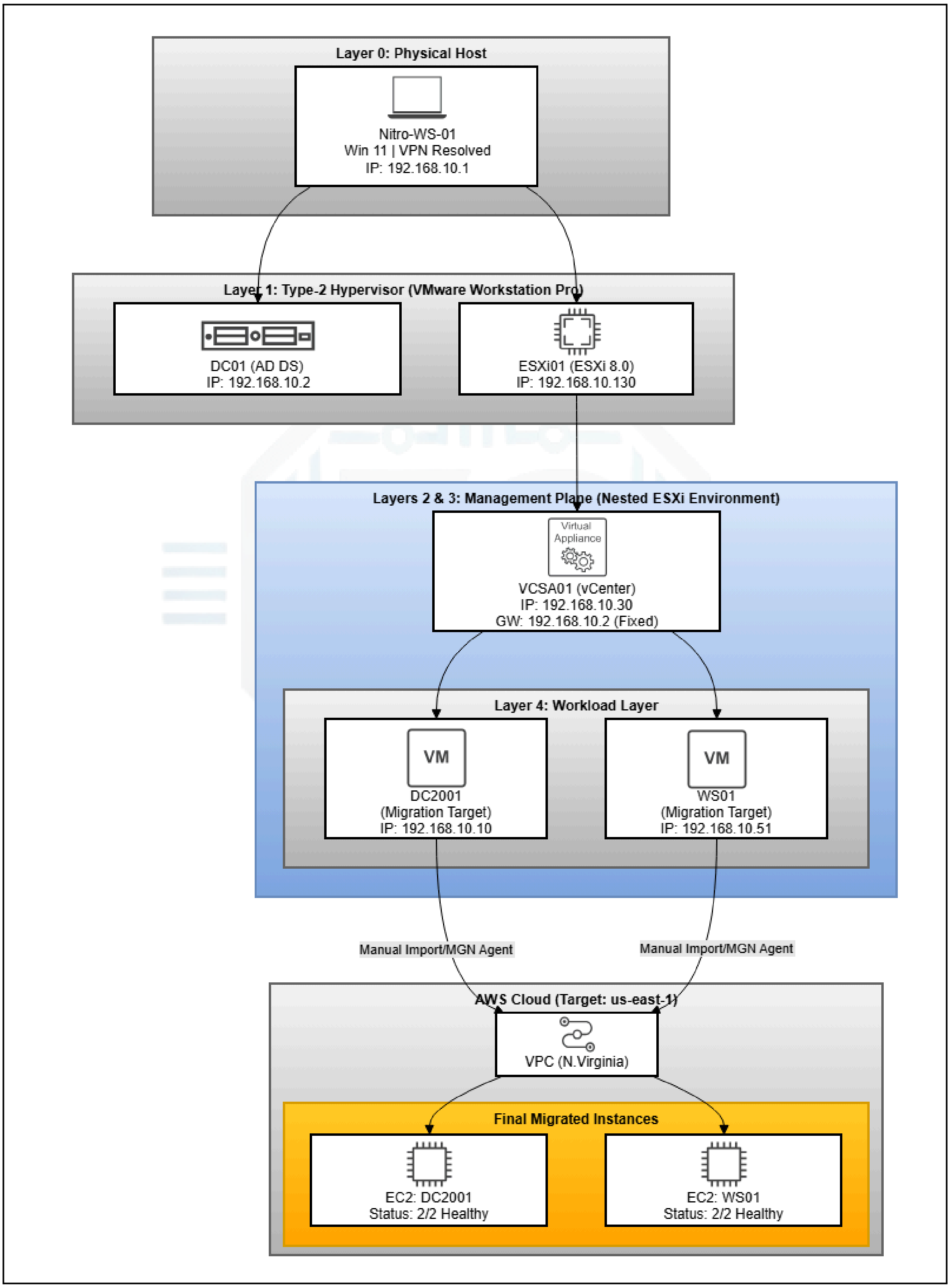


Figure 1: VMware-to-AWS Migration Topology. March 4th 2026. Eldon Gabriel

0.1 Executive Summary

This report explains how two VMware virtual machines, **Windows Server (DC2001)** and **Windows 11 Workstation (WS01)**, were moved to **Amazon EC2 using the AWS Application Migration Service (MGN**).

After the move, the AWS replication agent could not be removed from the original system. The uninstaller did not work because the driver was still running and blocking the files. Normal tools could not remove this driver. The problem was that the driver started with the system and stopped deleting the files.

Starting the system in Safe Mode prevented the driver from loading. This allowed the Amazon installation directory and registry entries to be eliminated.

There were more problems during the move. A VPN connection slowed down the data transfer and caused synchronization to take over 43 h.

Turning off the VPN and resetting the network fixed the speed issues.

There were also issues with the cloud launch due to licensing and configuration settings in Amazon Web Services. Changing the EC2 launch template and enabling the Default Host Management Configuration restored remote access to the moved systems.

During the review, the automated tools identified **347 unused cloud resources**. These resources were checked and removed to reduce costs.

After addressing these issues, both systems exhibited healthy replication and passed the EC2 status checks. The environment is now stable, and the process has been improved to prevent similar problems in the future.

1.0: AWS MGN Agent Removal & Environment Cleanup

1.1 Incident Overview

Incident ID: AWS-MGN-2026-03-04

Date: March 4, 2026

Environment: Hybrid Lab (VMware vSphere to AWS EC2)

1.2 Initial Symptoms

- Orphaned AWS Replication Agent processes identified in the Windows Task Manager.
- “Access Denied” when deleting `C:\Program Files (x86)\Amazon`
- The standard uninstaller failed to remove the kernel-mode driver (`.sys` file), indicating an active file lock.

1.3 Impact Assessment

- **Systems Affected:** Primary Migration Wave Windows Source Server (DC2001), Windows 11 Workstation (WS01)
- **Users Affected:** Lab Administrator
- **Severity Level:** High (Blocked cleanup and decommissioning of local resources)

1.4 Technical Analysis: Multilayer Infrastructure Stack

Layer Identification:

- **Layer 0 (Physical Host):** Nitro-WS-01. This is the foundation. **VPN interference** was the bottleneck for all upstream layers.
- **Layer 1 (Type-2 Hypervisor):** VMware Workstation Pro. Hosts the base infrastructure: **DC01** and **ESXi01**.
- **Layer 2 (Type-1 Hypervisor):** Nested ESXi 8.0. Provides compute resources for the management and workload layers.
- **Layer 3 (Management Plane):** VMware vCenter Server Appliance (**VCSA01**). The critical fix was correcting the **.254 gateway error** via the VAMI to allow Internet withdrawal for the nested cluster.
- **Layer 4 (Workload Layer): DC2001** and **WS01**. The final migration targets where **Kernel-Mode driver locks** were resolved in Safe Mode.

1.4.1 Diagnostic & Remediation Steps

Phase 1: Windows Kernel Decommissioning

Stage 1 (File Lock Identification): Identified kernel-mode locks preventing folder deletion during normal runtime.

Stage 2 (Permission Override): Attempted ownership reassignment using `takeown` and `icacls`.

Result: Failed because of active kernel handle ownership by SYSTEM.

Stage 3 (Safe Mode Isolation): Booted the source OS into Safe Mode to prevent the AWS Replication driver initialization.

Result: The Amazon directory and associated registry keys were successfully removed.

Phase 2: Network & Throughput Optimization

Stage 4 (VPN Contention): Observed extremely high synchronization ETAs (43+ h). Identified active VPN on the host OS.

Stage 5 (Gateway Realignment): Disabled host VPN. Performed a Winsock reset and restarted the AWS Replication Agent to force a new handshake via the direct ISP gateway.

Result: The throughput increased significantly, and synchronization resumed.

Phase 3: AWS Launch & Cloud Access Management

Stage 6 (Licensing & Tenancy Conflict): The initial launch failed because of a `ConflictException` (BYOL/Dedicated Host mismatch).
Reconfigured Launch Template to "AWS Provided" licensing and "Default" tenancy.

Stage 7 (Identity & Management Integration): Instances launched but appeared “Unmanaged” in the Fleet Manager. The Default Host Management Configuration (DHMC) was activated at the account level.

Result: Secure browser-based RDP/SSM access was enabled without local client restrictions.

Phase 4: Out-of-Band (OOB) Troubleshooting

Stage 8 (Serial Console Authorization): Standard RDP/SSM became unreachable during driver updates. EC2 Serial Console access was enabled at the account level.

Stage 9 (OOB Verification): Established serial session; black screen observed. Determined to be related to EMS/BCD configuration post-migration.

1.5 Root Cause

The AWS MGN agent uses a kernel-mode filter driver for block-level replication. The driver remains a resident in memory during active OS operations, preventing modification or deletion by standard administrative accounts.

1.6 Resolution Implemented

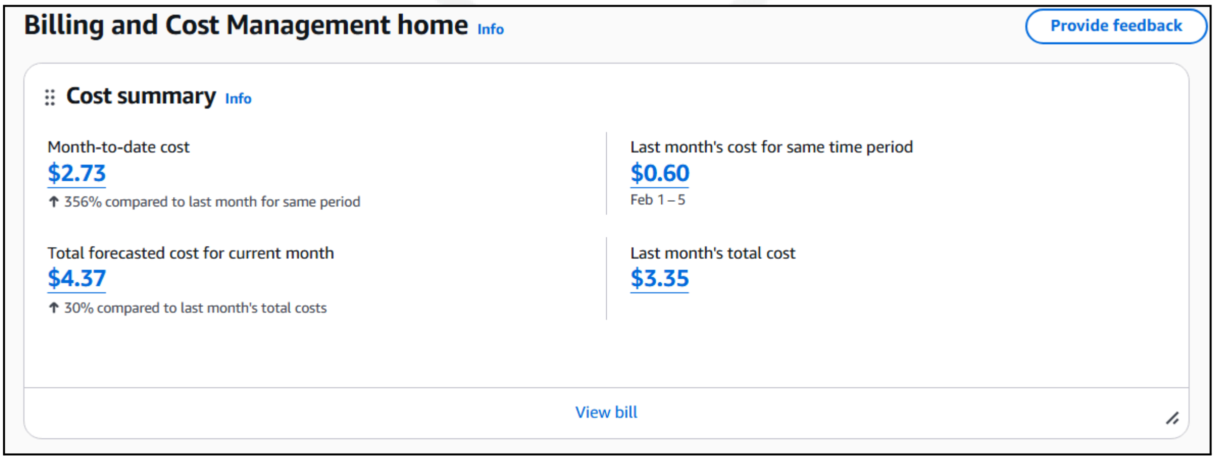
Windows: Booted into Safe Mode to isolate the driver. Performed manual folder and registry purge, followed by local `C:\` execution of the uninstaller.

Infrastructure: Initiated a global regional audit to decommission 347 host resources (VPCs, Subnets, and ENIs) automatically provisioned by AWS Transform.

1.7 Verification

- Rebooted DC2001 and WS01 into Normal mode and verified the absence of the Amazon directory.
- Confirmed that no AWS services were listed in `services.msc`.
- Cloud validation confirmed a healthy heartbeat in the AWS MGN Console and 2/2 EC2 status checks.

1.8 Financial Monitoring & Cost Variance



Billing and Cost Management home

Cost summary

Month-to-date cost
$2.73
↑ 356% compared to last month for same period

Last month's cost for same time period
$0.60
Feb 1 – 5

Total forecasted cost for current month
$4.37
↑ 30% compared to last month's total costs

Last month's total cost
$3.35

Figure 2: Cost Summary. March 4th 2026. www.aws.amazon.com.

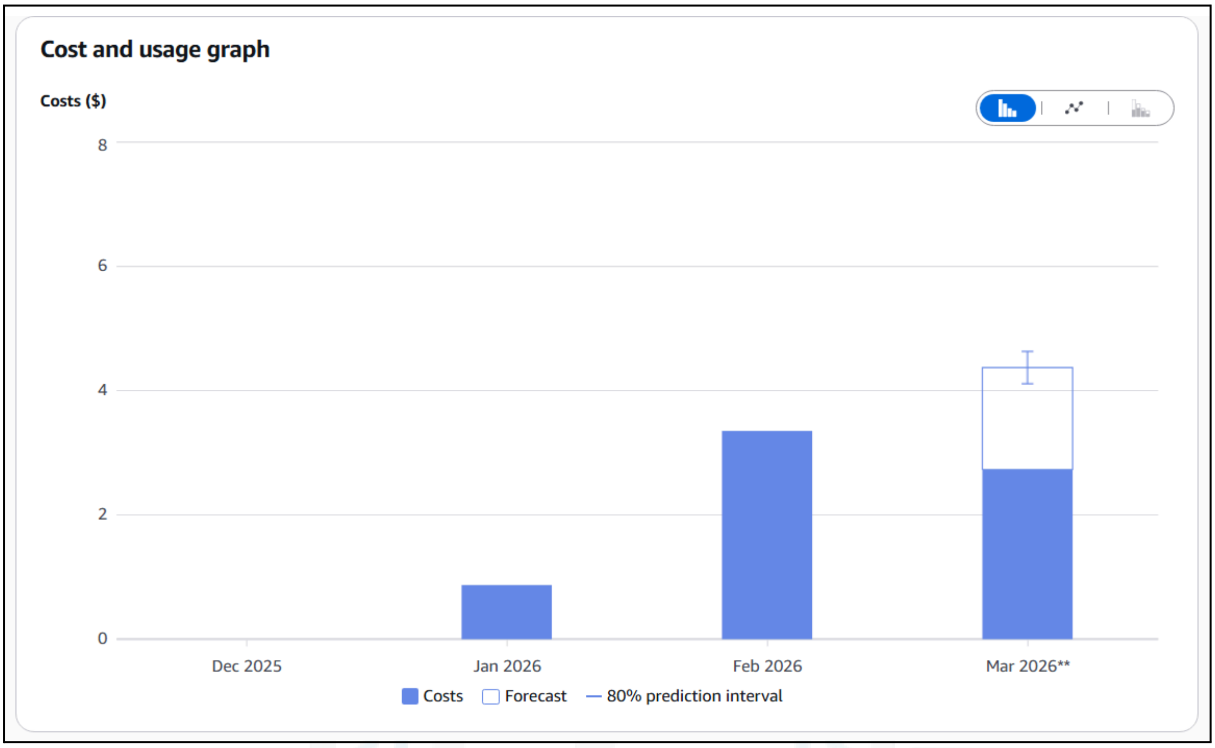


Figure 3: Cost and Usage Graph. March 4th 2026. www.aws.amazon.com.

- **MTD Cost:** $2.73 (356% increase compared to previous period)
- **Projected Cost:** $4.37 attributed to global resource sprawl from automated discovery
- **Mitigation:** Pivoted to manual migration and decommissioned 347 orphaned resources, thereby eliminating ongoing idle charges.

By transitioning to manual migration, the projected monthly burn was reduced by approximately 70%, aligning the project with a 'Minimalist Cloud' architecture.

1.9 Lessons Learned

Key Technical Takeaway: Kernel-level drivers require Safe Mode for manual removal because they are initialized before user-level access controls.

Project Note: vCenter Server excluded from the initial migration wave due to Photon OS kernel header mismatches and scheduled for a secondary agentless migration wave.

Preventive Measure: Uninstall the local agent before deleting the server record in the AWS Console to prevent the agent from being in an orphaned or locked state.

Although automation should be viewed as a tool for speed, manual validation remains the primary tool for accuracy and cost control.

1.10 Pattern Tracking

Recurring Issue: No

Correlation: Deleting cloud metadata before terminating the local agent results in a persistent file lock.