From Manual Toil to One-Click Recovery:

Safe Destructive Operations at Scale

Bloomberg Engineering

SREcon26 EMEA
October 14, 2026

Pranay B. Kolakkar, Senior Software Engineer
Timothy Izod, Engineering Team Lead

TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.

What is Bloomberg?

Platform & Tools

- Provides financial software tools, applications, and news to financial companies
- Enables analytics, data services, etc.

Global Scale

~400B–650B+

pieces of financial data daily

~2M+

news stories published daily

Infrastructure & Team

9,000+

engineers globally

- Runs and manages one of the world's largest private networks

TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Agenda

1. Standard Major OS Version Upgrade Mechanisms
2. Why Did We Build Another Upgrade Mechanism?
3. Our Approach
4. Architectural Overview
5. Challenges and Key Takeaways
6. Questions

TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Standard Major OS Version Upgrade Mechanisms



do-release-upgrade

Standard upgrade mechanism for Ubuntu systems



leapp

Standard upgrade tool for RHEL ecosystem

Common Grounds

- In-place upgrades using existing hard-disk layouts
- Replaces package repositories automatically
- Runs upgrades directly through underlying package managers
- Reboot the host into the new operating system version

TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Why Did We Build Another Upgrade Mechanism?

Requirements

- Need to repartition the hard drive layout
- A clean install provides consistent state without remnants of the previous OS
- Lack of human presence at some target sites
- Must back up and restore application state (certificates, configs, keys) post upgrade
- Need to support heterogeneous environment (physical/virtual, UEFI/non-UEFI)
- Should work on low network bandwidth targets

Key Challenges

- Lack of support for modifying existing disk layouts within current mechanisms
- Absence of out-of-band network availability on specific host systems
- Restricted network bandwidth at designated operational sites
- Determining backup staging locations during disk restructuring
- Need for early rollback mechanisms to avoid client outages

© 2026 Bloomberg Finance L.P. All rights reserved.

Our Approach: Scale & Constraints

DEPLOYMENT SCALE

12,000+ Hosts pending major OS upgrades across global infrastructure under strict maintenance windows

Operational & Site Challenges

- Unmanned target sites
- Sites with highly restricted network connectivity
- Standard remote provisioning is not permitted over restricted connectivity
- Strict client maintenance windows required rapid, zero-touch execution
- Support for a varied list of hardware models ranging from 2016 to today

Technical & Tooling Bottlenecks

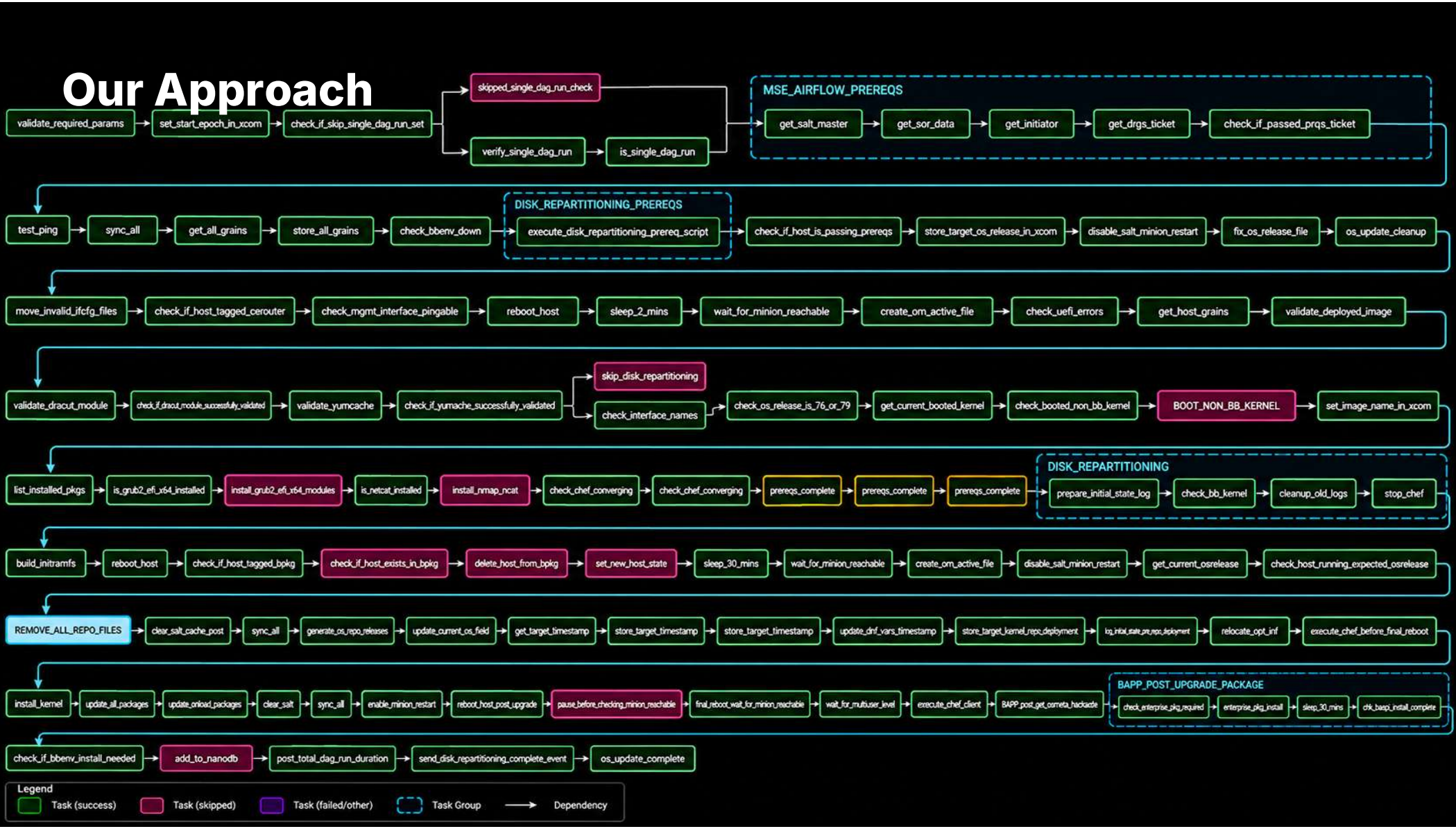
- Partition layout changes required full, clean OS re-installations
- Out-of-band remote installs remained heavily manual and risk-prone
- Existing mechanisms provided no built-in path for disk restructuring
- Minimal operational overhead to avoid human errors
- Detecting failure proactively to prevent client outages

© 2026 Bloomberg Finance L.P. All rights reserved.

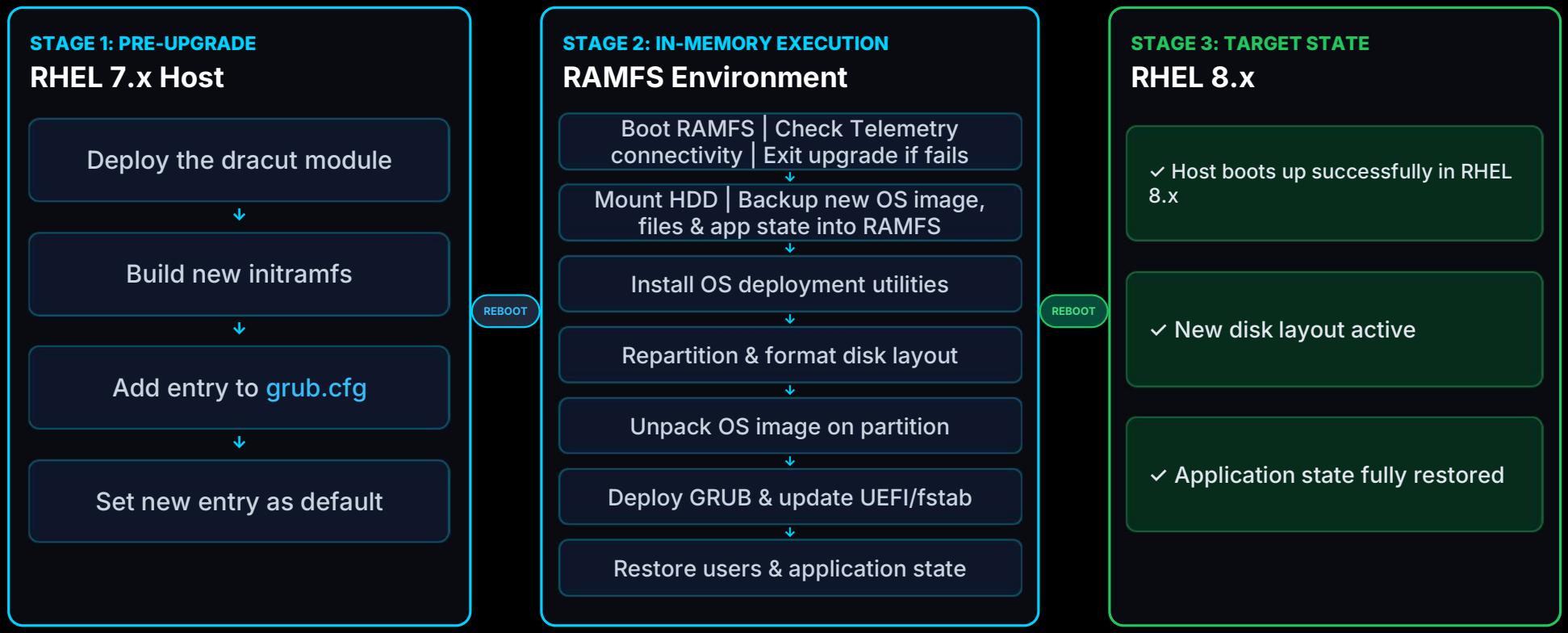
Bloomberg

Engineering

Our Approach



Architecture Overview (Simplified)



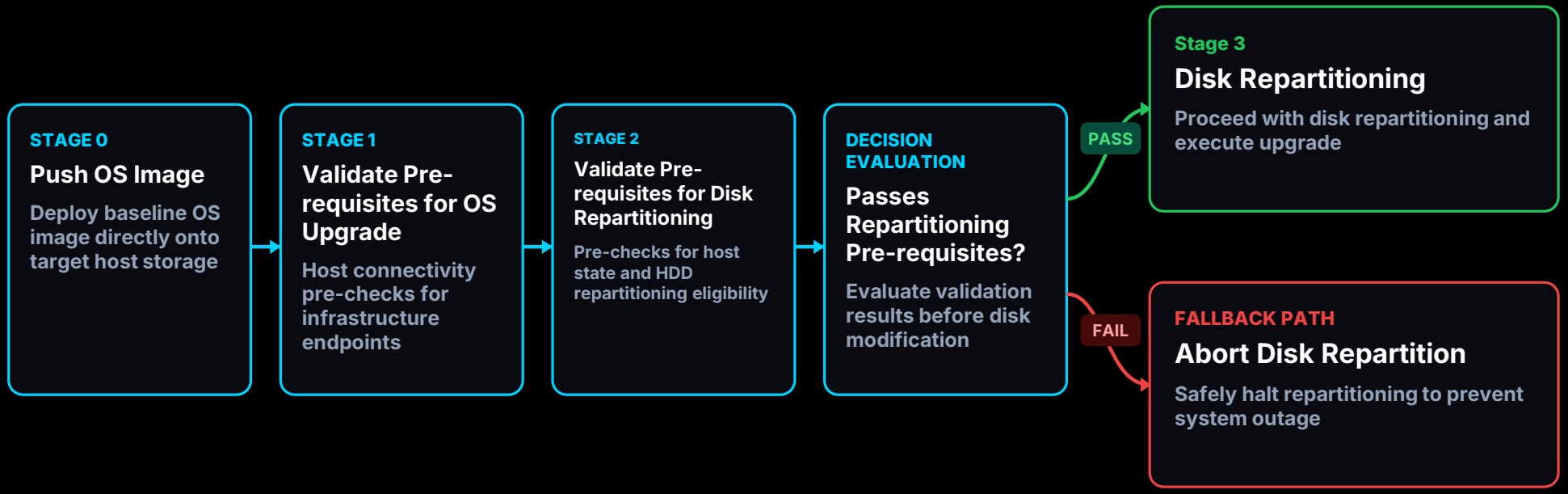
TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Workflow Overview



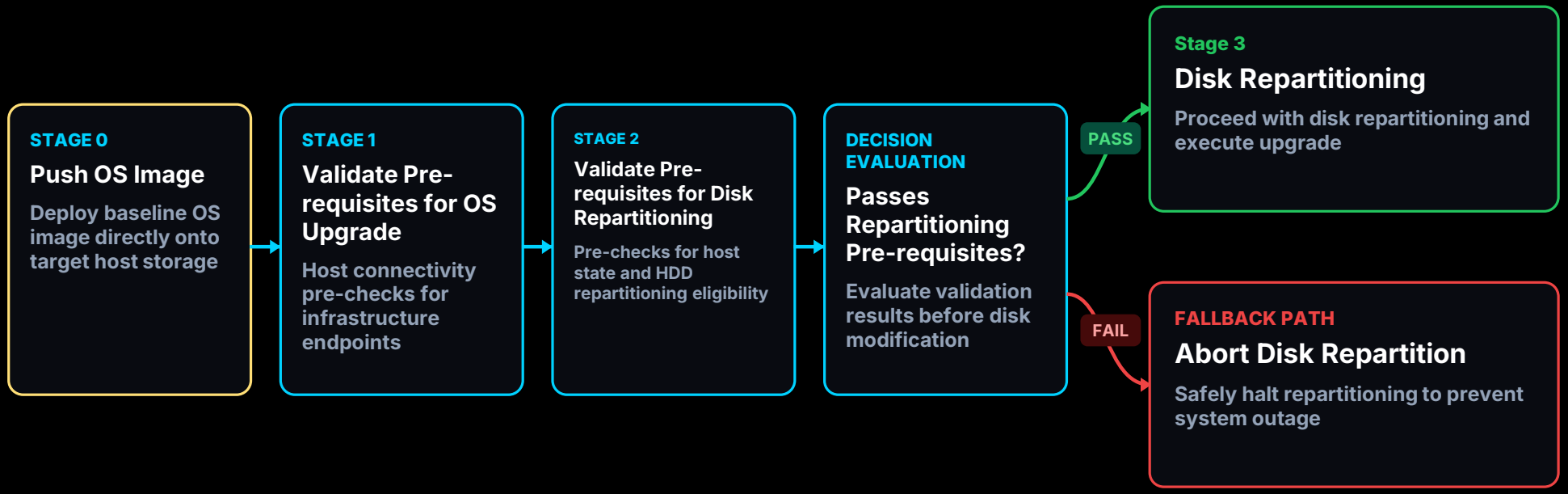
TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Workflow Overview

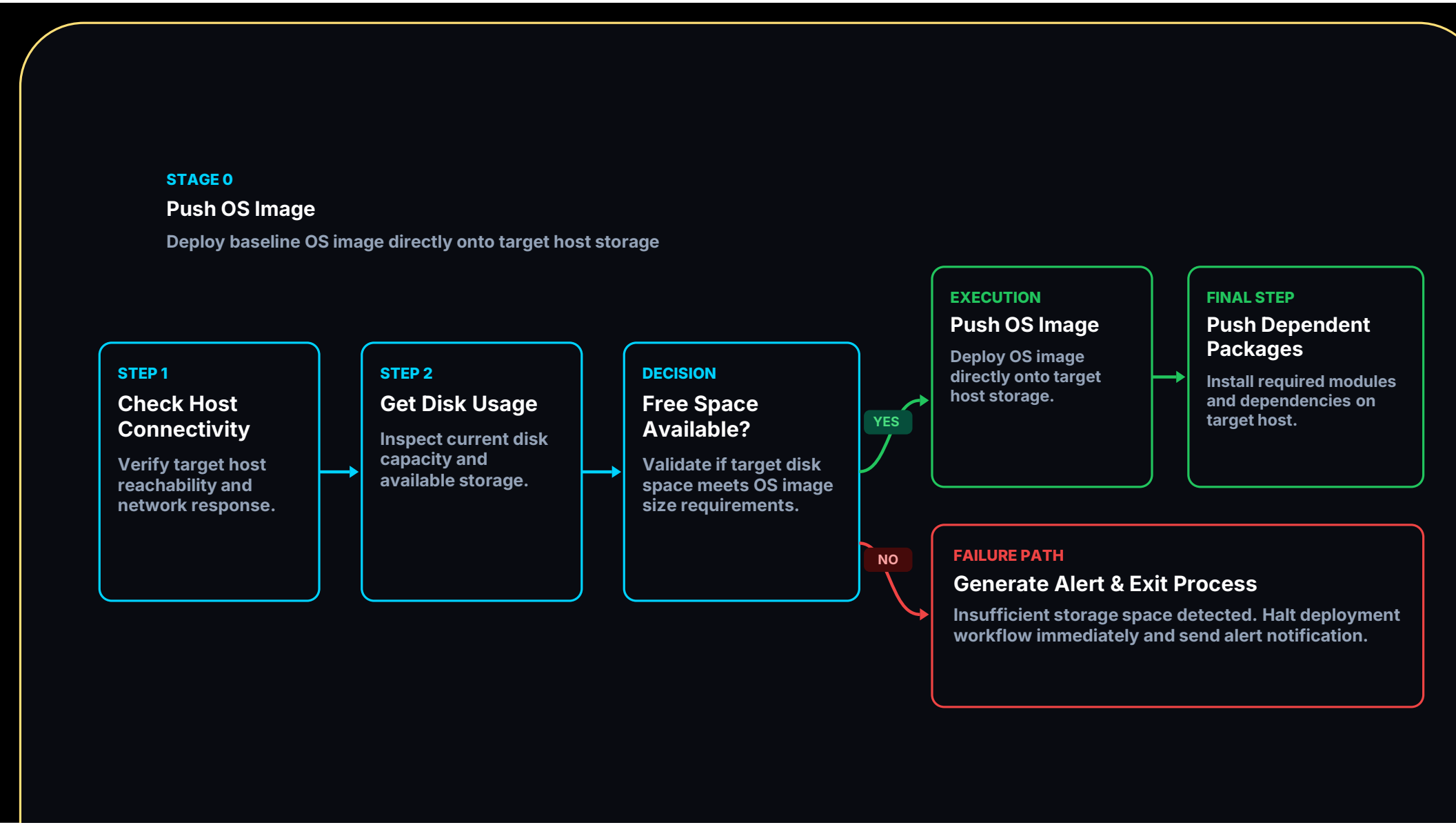


TechAtBloomberg.com

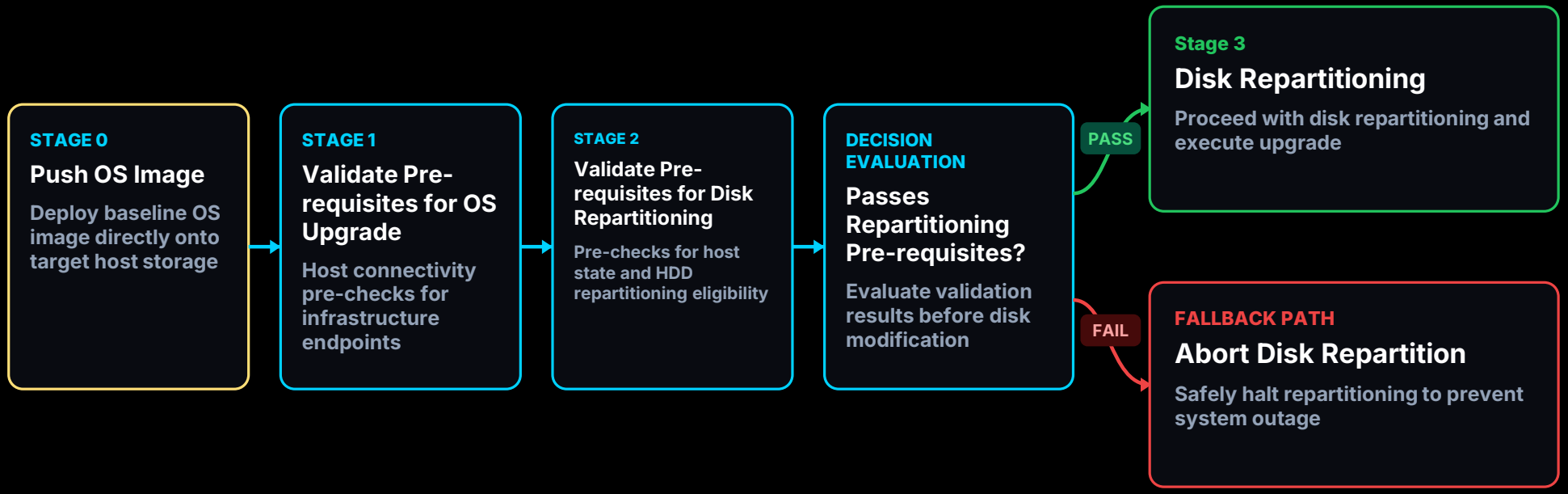
© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering


STAGE 0
Push OS Image
Deploy baseline OS image directly onto target host storage
STEP 1
Check Host Connectivity
Verify target host reachability and network response.
STEP 2
Get Disk Usage
Inspect current disk capacity and available storage.
DECISION
Free Space Available?
Validate if target disk space meets OS image size requirements.
YES
NO
EXECUTION
Push OS Image
Deploy OS image directly onto target host storage.
FINAL STEP
Push Dependent Packages
Install required modules and dependencies on target host.
FAILURE PATH
Generate Alert & Exit Process
Insufficient storage space detected. Halt deployment workflow immediately and send alert notification.

Workflow Overview



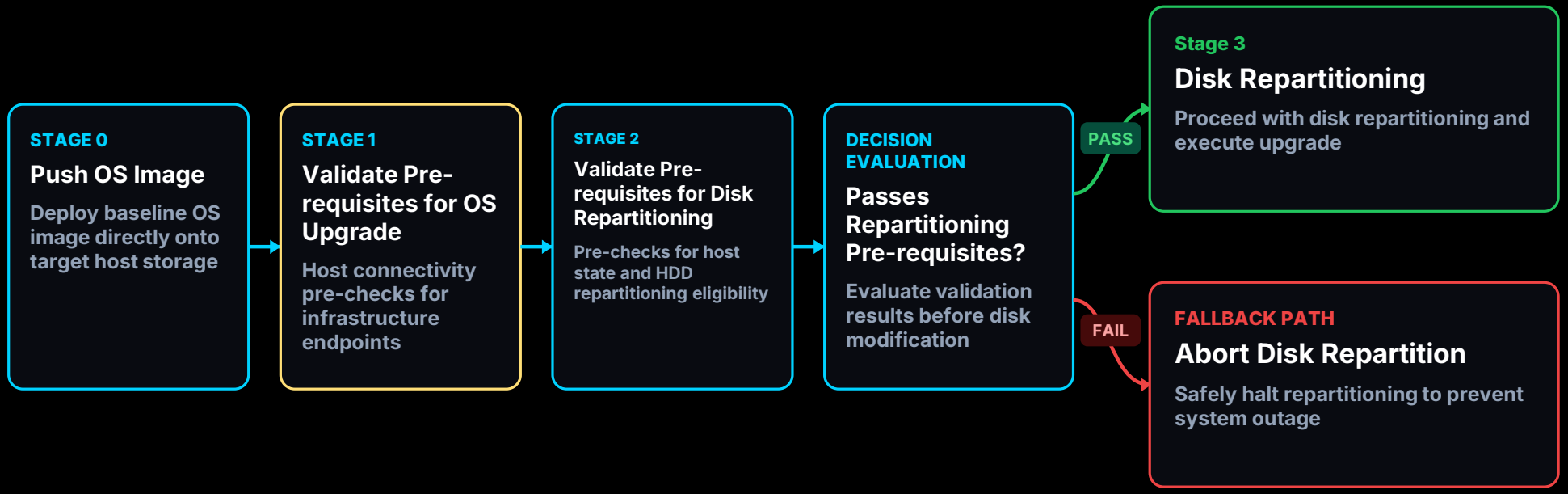
TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Workflow Overview

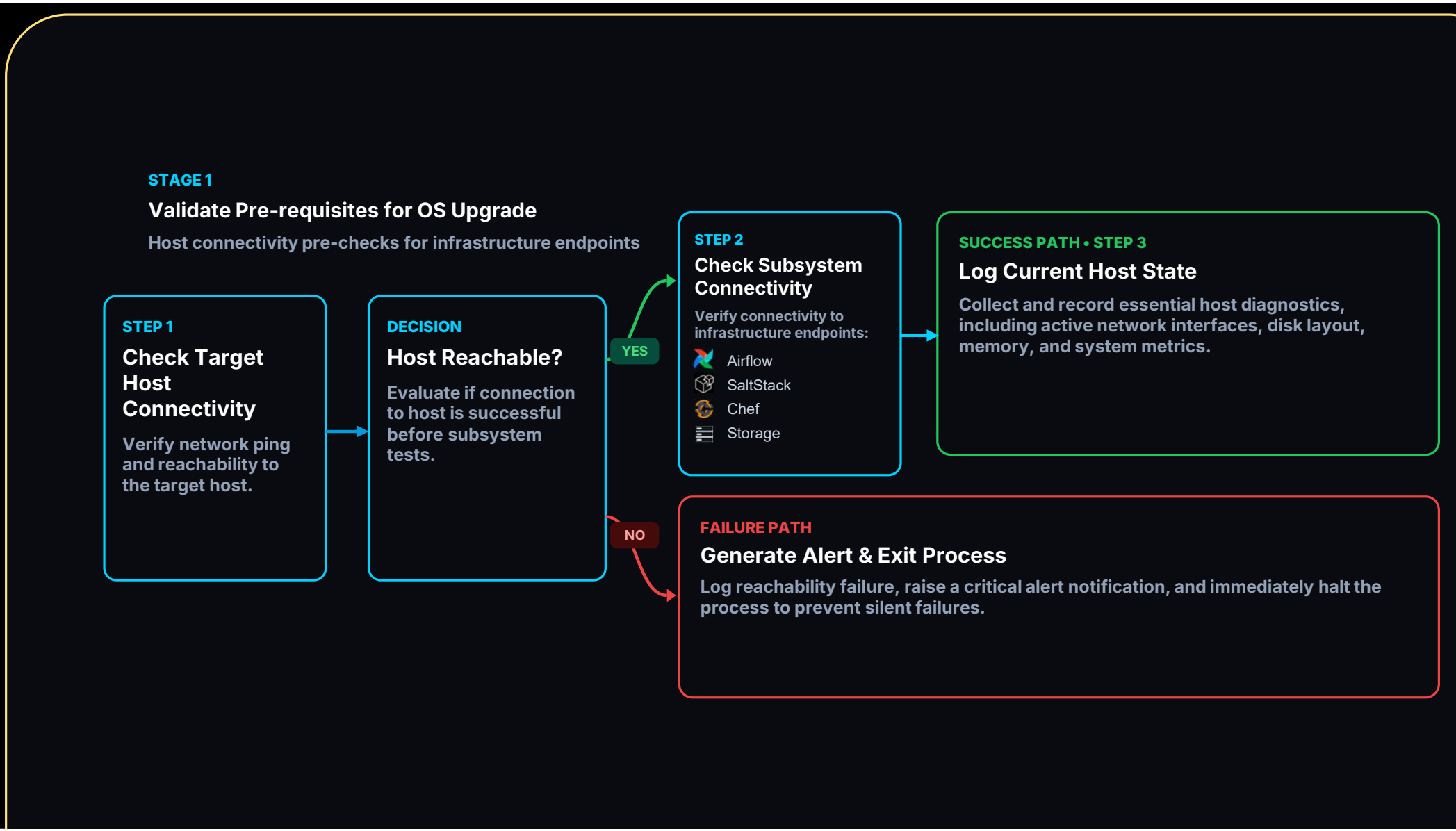


TechAtBloomberg.com

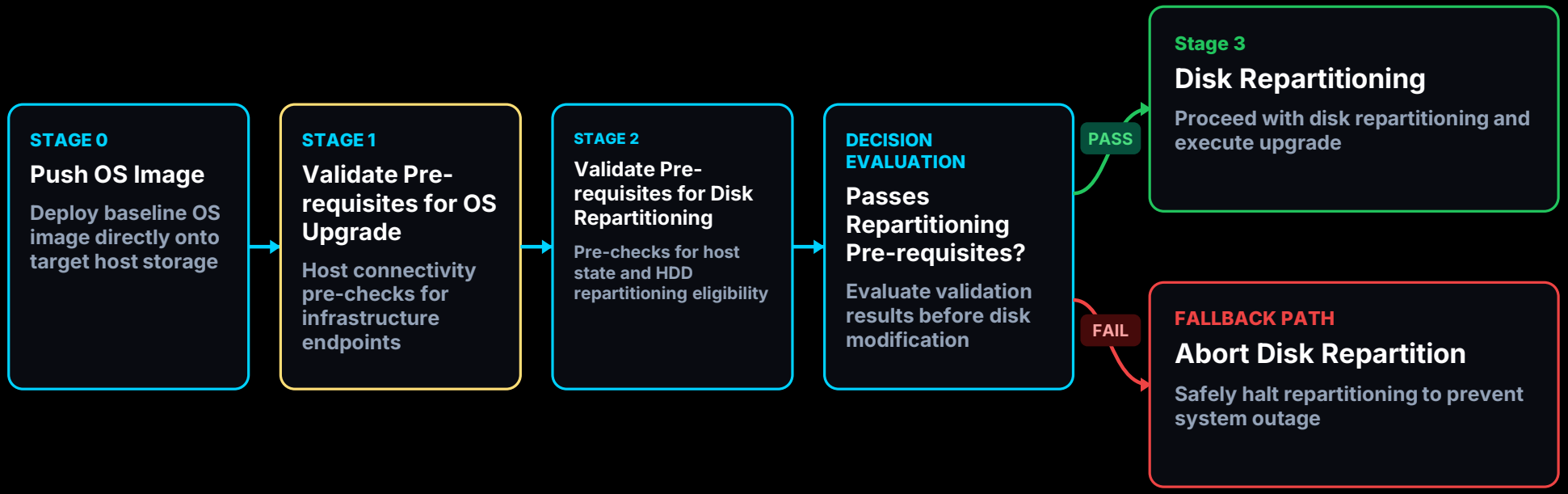
© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering


STAGE 1
Validate Pre-requisites for OS Upgrade
Host connectivity pre-checks for infrastructure endpoints
STEP 1
Check Target Host Connectivity
Verify network ping and reachability to the target host.
DECISION
Host Reachable?
Evaluate if connection to host is successful before subsystem tests.
YES
NO
STEP 2
Check Subsystem Connectivity
Verify connectivity to infrastructure endpoints:
Airflow
SaltStack
Chef
Storage
SUCCESS PATH • STEP 3
Log Current Host State
Collect and record essential host diagnostics, including active network interfaces, disk layout, memory, and system metrics.
FAILURE PATH
Generate Alert & Exit Process
Log reachability failure, raise a critical alert notification, and immediately halt the process to prevent silent failures.

Workflow Overview



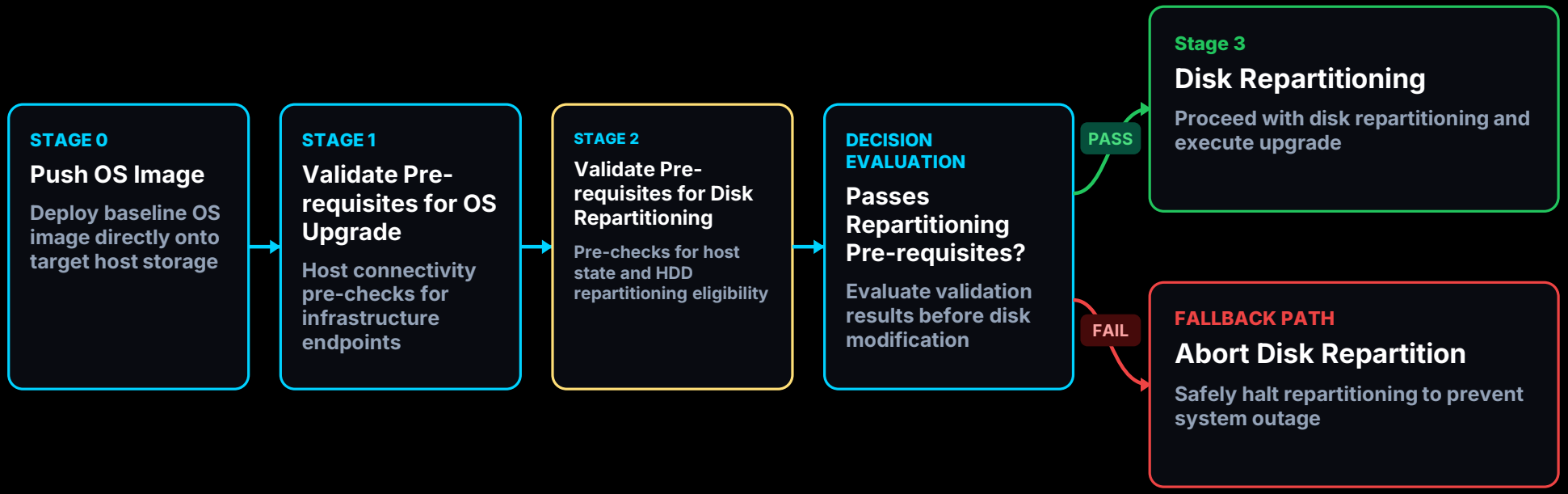
TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Workflow Overview

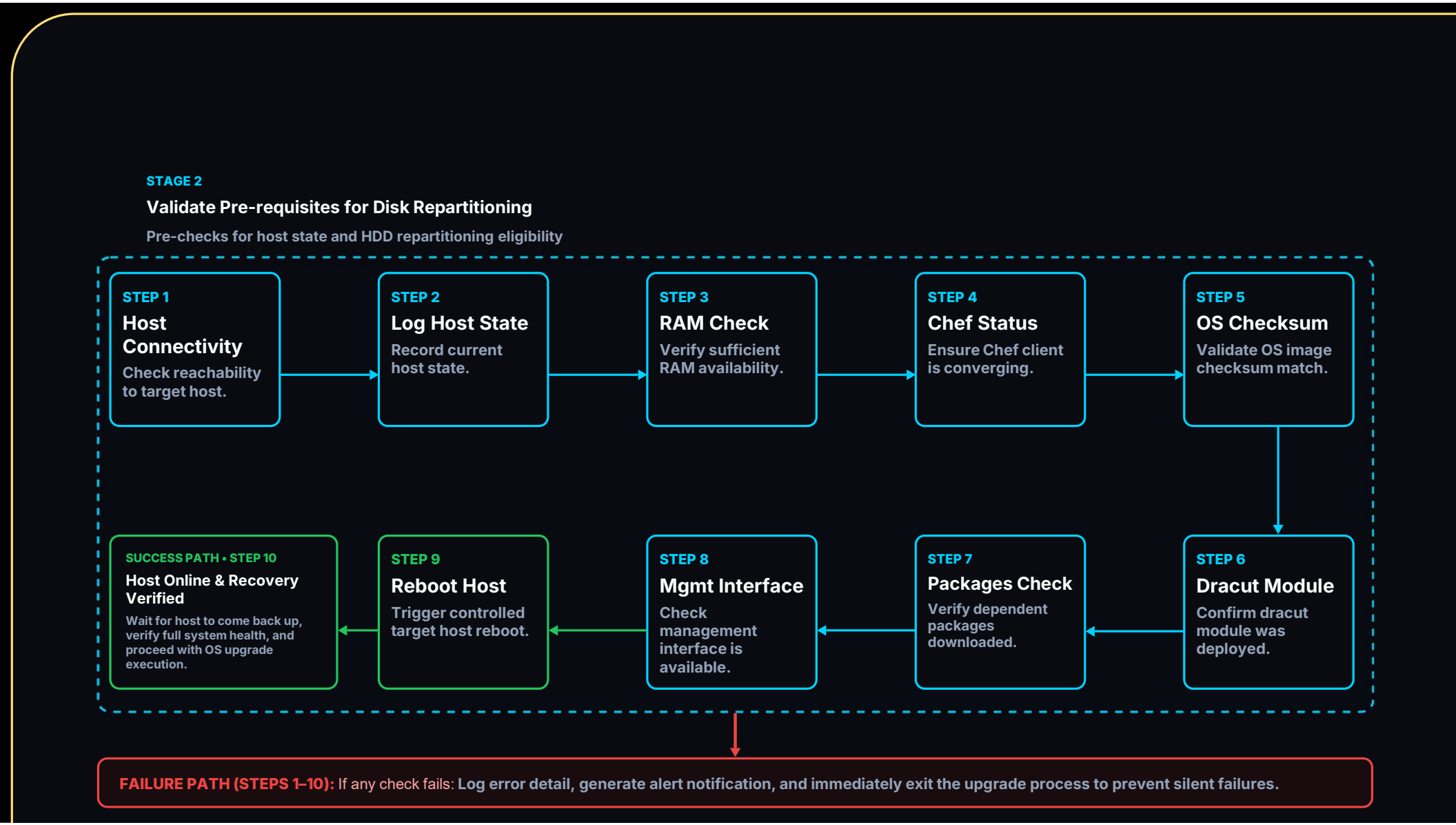


TechAtBloomberg.com

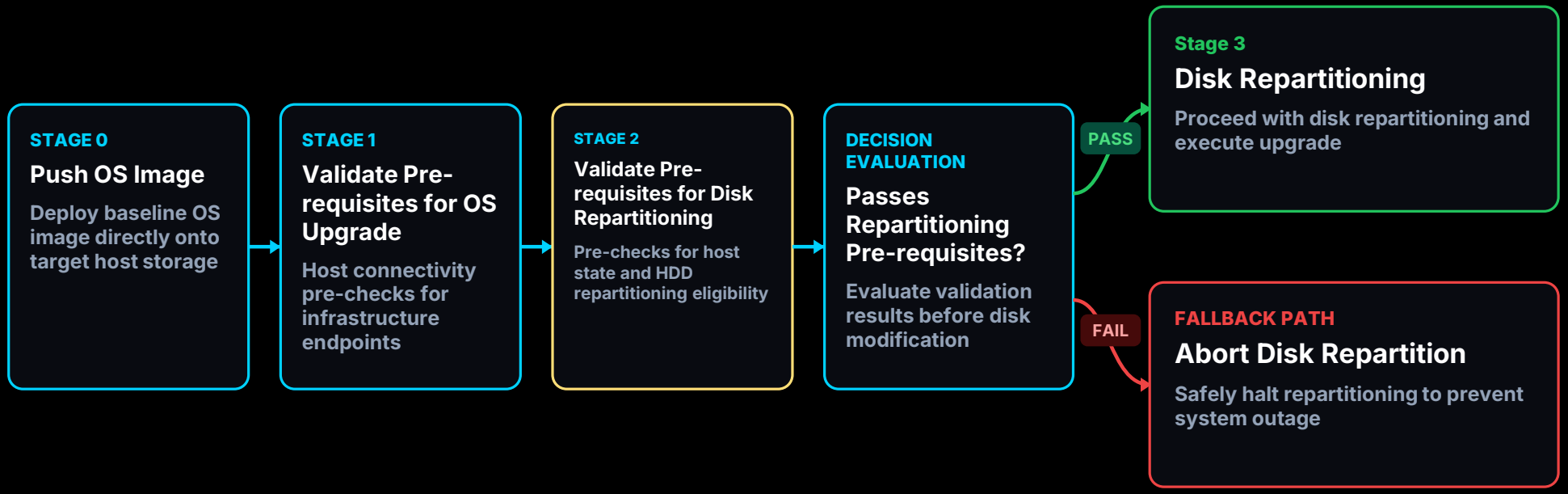
© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering


STAGE 2
Validate Pre-requisites for Disk Repartitioning
Pre-checks for host state and HDD repartitioning eligibility
STEP 1
Host Connectivity
Check reachability to target host.
STEP 2
Log Host State
Record current host state.
STEP 3
RAM Check
Verify sufficient RAM availability.
STEP 4
Chef Status
Ensure Chef client is converging.
STEP 5
OS Checksum
Validate OS image checksum match.
STEP 6
Dracut Module
Confirm dracut module was deployed.
STEP 7
Packages Check
Verify dependent packages downloaded.
STEP 8
Mgmt Interface
Check management interface is available.
STEP 9
Reboot Host
Trigger controlled target host reboot.
SUCCESS PATH • STEP 10
Host Online & Recovery Verified
Wait for host to come back up, verify full system health, and proceed with OS upgrade execution.
FAILURE PATH (STEPS 1–10): If any check fails: Log error detail, generate alert notification, and immediately exit the upgrade process to prevent silent failures.

Workflow Overview



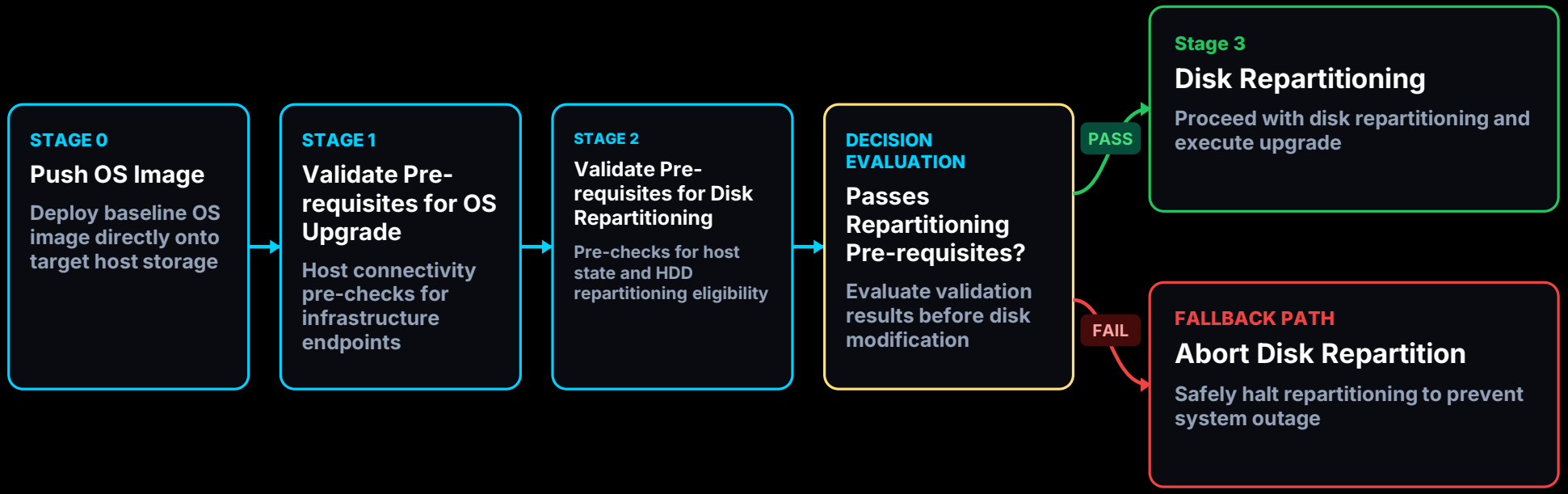
TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Workflow Overview



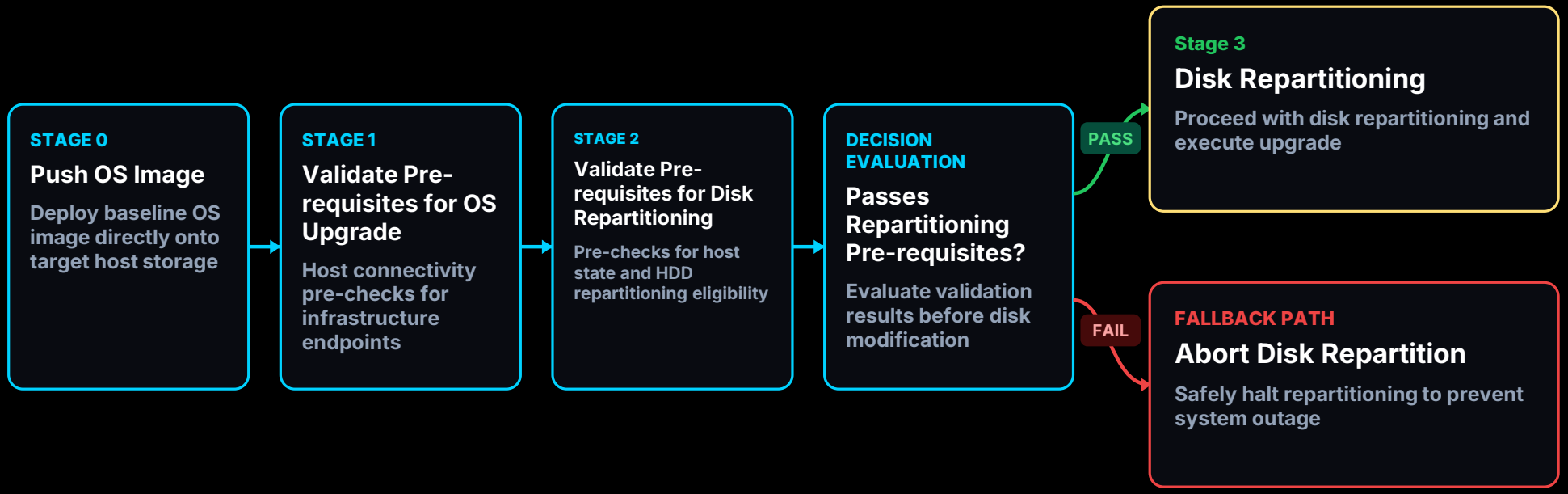
TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Workflow Overview



TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.

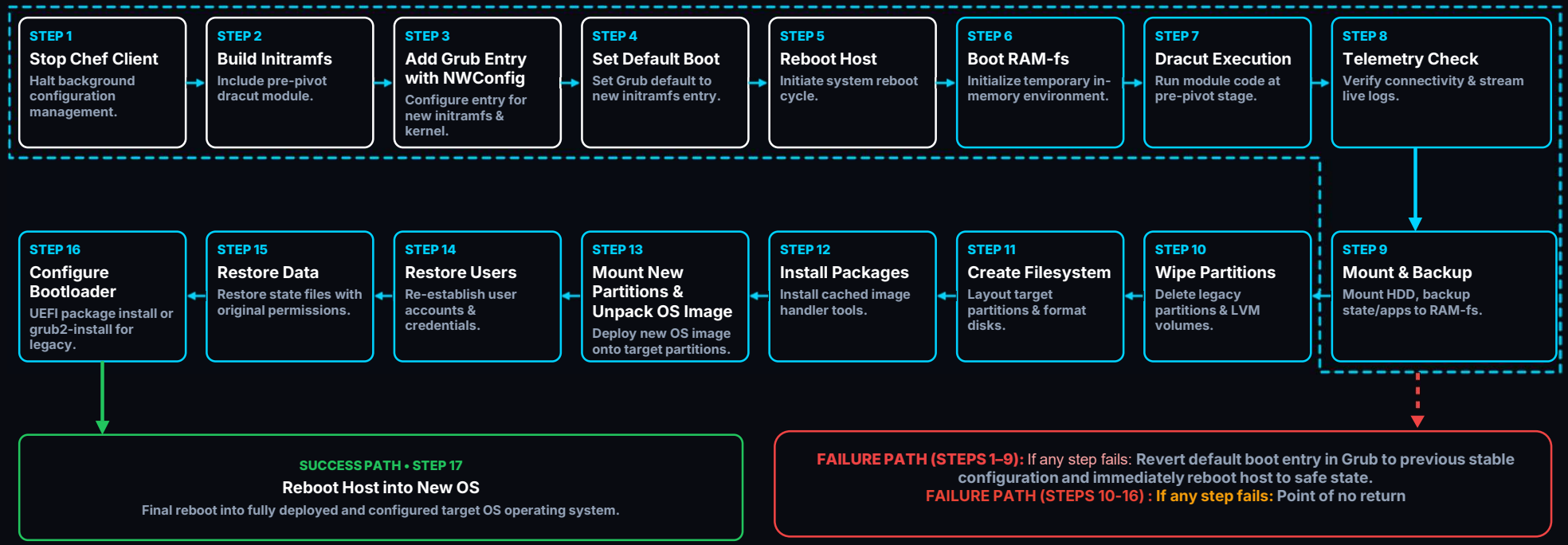
Bloomberg

Engineering

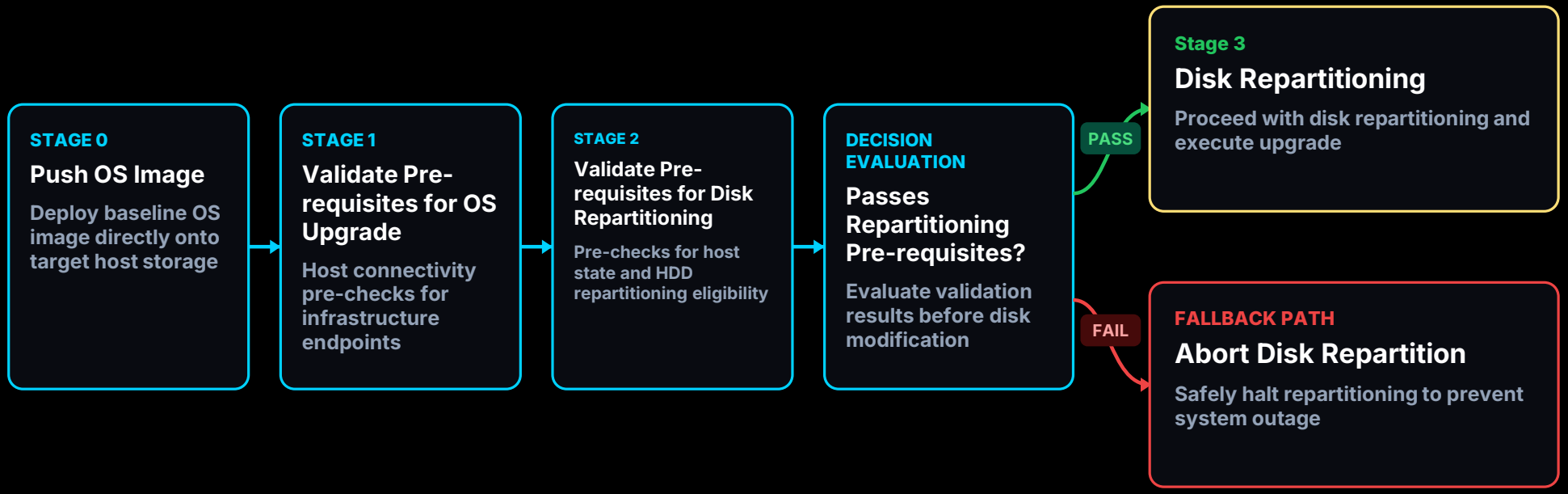
Stage 3

Disk Repartitioning

Proceed with disk repartitioning and execute upgrade



Workflow Overview



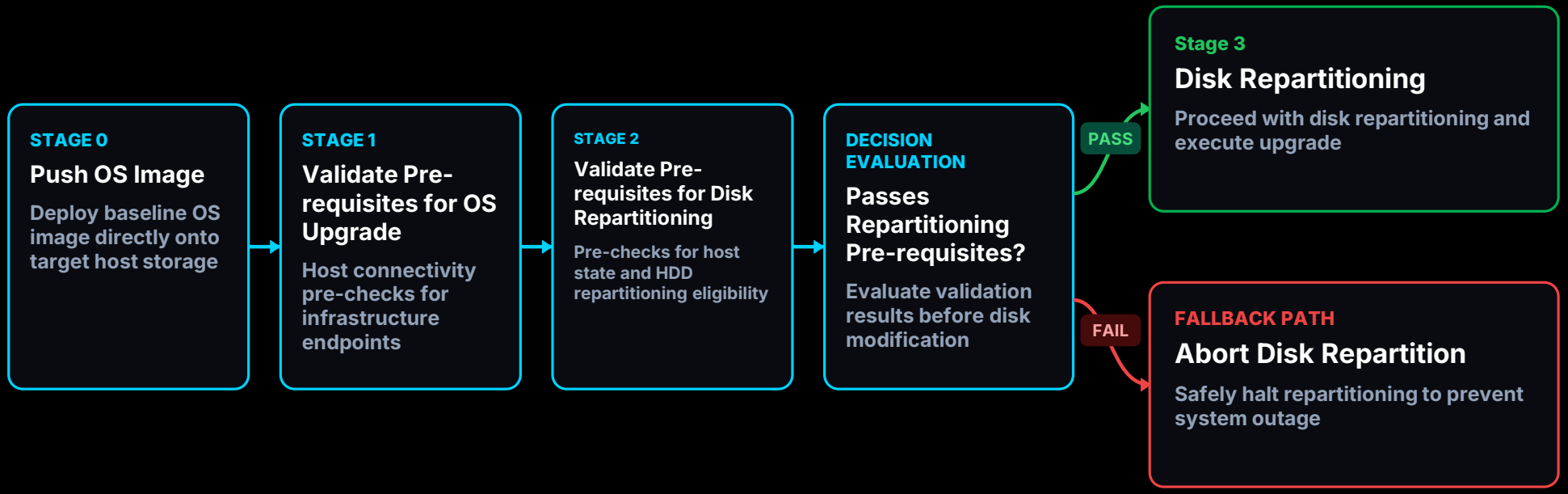
TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Workflow Overview



TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Challenges and Takeaways

AUTOMATION & RECOVERY

Automate Error-Prone Manual Steps

Manual intervention during failed upgrades was prone to error. Developed auto-detection to resume execution from exact failure step.

SYSTEM RESILIENCE

Guardrails for Distributed Systems

Distributed systems involve numerous moving components. Overcame false assumptions regarding infrastructure resilience by engineering dedicated guardrails.

NETWORK OPTIMIZATION

One-Shot Low-Bandwidth Upgrades

Target hosts operate in bandwidth-constrained sites. Built an optimized single-pass upgrade mechanism to execute seamlessly despite limitations.

HARDWARE PREREQUISITES

Strict Memory Eligibility Rules

Upgrades on resource-constrained hardware posed system stability risks. Enforced strict pre-checks to exclude any host with less than 10 GB of RAM.

TESTING & RELIABILITY

99.999% Overall Success Rate

Achieved 99.999% success rate in production. 0.001% failures during dev environment testing steered building guardrails for manual intervention. 12K+ hosts upgraded successfully resulting in cost savings.

EXECUTION EFFICIENCY

40-Minute End-to-End Process

Entire upgrade process took 40 minutes end-to-end to return the host safely back into production environment.

PREVENTATIVE PRECHECKS

Fail-Early Network Safeguards

0.001% failures helped build a plethora of prechecks to fail early in the process, avoiding losing host connectivity over network.

OBSERVABILITY & TRACING

Log Live Stream & Tracing

Entire process streams live back to Log Dashboard with trace availability, helping operators escalate outages early.

TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Challenges and Takeaways

AUTOMATION & RECOVERY

Automate Error-Prone Manual Steps

Manual intervention during failed upgrades was prone to error. Developed auto-detection to resume execution from exact failure step.

SYSTEM RESILIENCE

Guardrails for Distributed Systems

Distributed systems involve numerous moving components. Overcame false assumptions regarding infrastructure resilience by engineering dedicated guardrails.

NETWORK OPTIMIZATION

One-Shot Low-Bandwidth Upgrades

Target hosts operate in bandwidth-constrained sites. Built an optimized single-pass upgrade mechanism to execute seamlessly despite limitations.

HARDWARE PREREQUISITES

Strict Memory Eligibility Rules

Upgrades on resource-constrained hardware posed system stability risks. Enforced strict pre-checks to exclude any host with less than 10 GB of RAM.

TESTING & RELIABILITY

99.999% Overall Success Rate

Achieved 99.999% success rate in production. 0.001% failures during dev environment testing steered building guardrails for manual intervention. 12K+ hosts upgraded successfully resulting in cost savings.

EXECUTION EFFICIENCY

40-Minute End-to-End Process

Entire upgrade process took 40 minutes end-to-end to return the host safely back into production environment.

PREVENTATIVE PRECHECKS

Fail-Early Network Safeguards

0.001% failures helped build a plethora of prechecks to fail early in the process, avoiding losing host connectivity over network.

OBSERVABILITY & TRACING

Log Live Stream & Tracing

Entire process streams live back to Log Dashboard with trace availability, helping operators escalate outages early.

TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Key Takeaways - Automation & Recovery

AUTOMATION & RECOVERY

Automate Error-Prone Manual Steps

Manual intervention during failed upgrades was prone to error. Developed auto-detection to resume execution from exact failure step.

ORCHESTRATION LAYER

Open Source frameworks

We use **open source orchestration frameworks** as the orchestration layer for our upgrade workflows.

PREVIOUS PROCESS

Manual Process Restarts

Previously, if an upgrade failed at any stage, an engineer had to **manually restart the process from the appropriate failed task**.

OPERATIONAL RISK

Execution Errors

Manual recovery introduced operational risk: in some cases, the workflow was restarted from the **wrong task**, causing incorrect or unnecessary execution.

AUTOMATED RECOVERY

Smart Auto-Detection

We built a mechanism to **automatically determine the correct recovery point and restart the workflow from a predefined task** based on the failure location.

Challenges and Takeaways

AUTOMATION & RECOVERY

Automate Error-Prone Manual Steps

Manual intervention during failed upgrades was prone to error. Developed auto-detection to resume execution from exact failure step.

SYSTEM RESILIENCE

Guardrails for Distributed Systems

Distributed systems involve numerous moving components. Overcame false assumptions regarding infrastructure resilience by engineering dedicated guardrails.

NETWORK OPTIMIZATION

One-Shot Low-Bandwidth Upgrades

Target hosts operate in bandwidth-constrained sites. Built an optimized single-pass upgrade mechanism to execute seamlessly despite limitations.

HARDWARE PREREQUISITES

Strict Memory Eligibility Rules

Upgrades on resource-constrained hardware posed system stability risks. Enforced strict pre-checks to exclude any host with less than 10 GB of RAM.

TESTING & RELIABILITY

99.999% Overall Success Rate

Achieved 99.999% success rate in production. 0.001% failures during dev environment testing steered building guardrails for manual intervention. 12K+ hosts upgraded successfully resulting in cost savings.

EXECUTION EFFICIENCY

40-Minute End-to-End Process

Entire upgrade process took 40 minutes end-to-end to return the host safely back into production environment.

PREVENTATIVE PRECHECKS

Fail-Early Network Safeguards

0.001% failures helped build a plethora of prechecks to fail early in the process, avoiding losing host connectivity over network.

OBSERVABILITY & TRACING

Log Live Stream & Tracing

Entire process streams live back to Log Dashboard with trace availability, helping operators escalate outages early.

TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Challenges and Takeaways

AUTOMATION & RECOVERY

Automate Error-Prone Manual Steps

Manual intervention during failed upgrades was prone to error. Developed auto-detection to resume execution from exact failure step.

SYSTEM RESILIENCE

Guardrails for Distributed Systems

Distributed systems involve numerous moving components. Overcame false assumptions regarding infrastructure resilience by engineering dedicated guardrails.

NETWORK OPTIMIZATION

One-Shot Low-Bandwidth Upgrades

Target hosts operate in bandwidth-constrained sites. Built an optimized single-pass upgrade mechanism to execute seamlessly despite limitations.

HARDWARE PREREQUISITES

Strict Memory Eligibility Rules

Upgrades on resource-constrained hardware posed system stability risks. Enforced strict pre-checks to exclude any host with less than 10 GB of RAM.

TESTING & RELIABILITY

99.999% Overall Success Rate

Achieved 99.999% success rate in production. 0.001% failures during dev environment testing steered building guardrails for manual intervention. 12K+ hosts upgraded successfully resulting in cost savings.

EXECUTION EFFICIENCY

40-Minute End-to-End Process

Entire upgrade process took 40 minutes end-to-end to return the host safely back into production environment.

PREVENTATIVE PRECHECKS

Fail-Early Network Safeguards

0.001% failures helped build a plethora of prechecks to fail early in the process, avoiding losing host connectivity over network.

OBSERVABILITY & TRACING

Log Live Stream & Tracing

Entire process streams live back to Log Dashboard with trace availability, helping operators escalate outages early.

TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Key Takeaways - System Resilience

SYSTEM RESILIENCE

Guardrails for Distributed Systems

Distributed systems involve numerous moving components. Overcame false assumptions regarding infrastructure resilience by engineering dedicated guardrails.

SYSTEM COMPLEXITY

Multiple Failure Modes

In a distributed system, there are **many independent components and failure modes** operating simultaneously.

STACK-WIDE RISKS

Unpredictable Errors

Failures can occur **at any layer in the stack**, and designing proactively for all possibilities is inherently challenging.

CASCADING SCENARIOS

Cascading Outages

Examples include **exhausting connection pools**, PostgreSQL instance unavailability, or cascading failovers.

DEDICATED GUARDRAILS

Controlled Behavior

Introduced **guardrails and failure handling** to contain scenarios and make system behavior **predictable and deterministic**.

Challenges and Takeaways

AUTOMATION & RECOVERY

Automate Error-Prone Manual Steps

Manual intervention during failed upgrades was prone to error. Developed auto-detection to resume execution from exact failure step.

SYSTEM RESILIENCE

Guardrails for Distributed Systems

Distributed systems involve numerous moving components. Overcame false assumptions regarding infrastructure resilience by engineering dedicated guardrails.

NETWORK OPTIMIZATION

One-Shot Low-Bandwidth Upgrades

Target hosts operate in bandwidth-constrained sites. Built an optimized single-pass upgrade mechanism to execute seamlessly despite limitations.

HARDWARE PREREQUISITES

Strict Memory Eligibility Rules

Upgrades on resource-constrained hardware posed system stability risks. Enforced strict pre-checks to exclude any host with less than 10 GB of RAM.

TESTING & RELIABILITY

99.999% Overall Success Rate

Achieved 99.999% success rate in production. 0.001% failures during dev environment testing steered building guardrails for manual intervention. 12K+ hosts upgraded successfully resulting in cost savings.

EXECUTION EFFICIENCY

40-Minute End-to-End Process

Entire upgrade process took 40 minutes end-to-end to return the host safely back into production environment.

PREVENTATIVE PRECHECKS

Fail-Early Network Safeguards

0.001% failures helped build a plethora of prechecks to fail early in the process, avoiding losing host connectivity over network.

OBSERVABILITY & TRACING

Log Live Stream & Tracing

Entire process streams live back to Log Dashboard with trace availability, helping operators escalate outages early.

TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Challenges and Takeaways

AUTOMATION & RECOVERY

Automate Error-Prone Manual Steps

Manual intervention during failed upgrades was prone to error. Developed auto-detection to resume execution from exact failure step.

SYSTEM RESILIENCE

Guardrails for Distributed Systems

Distributed systems involve numerous moving components. Overcame false assumptions regarding infrastructure resilience by engineering dedicated guardrails.

NETWORK OPTIMIZATION

One-Shot Low-Bandwidth Upgrades

Target hosts operate in bandwidth-constrained sites. Built an optimized single-pass upgrade mechanism to execute seamlessly despite limitations.

HARDWARE PREREQUISITES

Strict Memory Eligibility Rules

Upgrades on resource-constrained hardware posed system stability risks. Enforced strict pre-checks to exclude any host with less than 10 GB of RAM.

TESTING & RELIABILITY

99.999% Overall Success Rate

Achieved 99.999% success rate in production. 0.001% failures during dev environment testing steered building guardrails for manual intervention. 12K+ hosts upgraded successfully resulting in cost savings.

EXECUTION EFFICIENCY

40-Minute End-to-End Process

Entire upgrade process took 40 minutes end-to-end to return the host safely back into production environment.

PREVENTATIVE PRECHECKS

Fail-Early Network Safeguards

0.001% failures helped build a plethora of prechecks to fail early in the process, avoiding losing host connectivity over network.

OBSERVABILITY & TRACING

Log Live Stream & Tracing

Entire process streams live back to Log Dashboard with trace availability, helping operators escalate outages early.

TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Key Takeaways - Preventative Prechecks

PREVENTATIVE PRECHECKS

Fail-Early Network Safeguards

0.001% failures helped build a plethora of prechecks to fail early in the process, avoiding losing host connectivity over network.

DIVERSE ENVIRONMENTS

Operational Constraints

Upgrades run across **diverse environments**—our data centers, customer sites, and co-located infrastructure—each with unique constraints.

HIGH-RISK FACTORS

System Environment Issues

Issues like **endpoint connectivity, bandwidth, memory, and missing artifacts** can derail upgrades midway and cause outages.

WORKFLOW IMPACT

Partial Failures

Impact depends on **where in the upgrade workflow failure occurs**, making partial failures difficult to recover from.

PRE-FLIGHT GUARDRAILS

Predictable Recovery

Introduced **pre-flight checks & guardrails** to validate prerequisites and turn unpredictable failures into **predictable recovery paths**.

TechAtBloomberg.com

Bloomberg

Engineering

© 2026 Bloomberg Finance L.P. All rights reserved.

© 2018 Bloomberg Finance L.P. All rights reserved.

Challenges and Takeaways

AUTOMATION & RECOVERY

Automate Error-Prone Manual Steps

Manual intervention during failed upgrades was prone to error. Developed auto-detection to resume execution from exact failure step.

SYSTEM RESILIENCE

Guardrails for Distributed Systems

Distributed systems involve numerous moving components. Overcame false assumptions regarding infrastructure resilience by engineering dedicated guardrails.

NETWORK OPTIMIZATION

One-Shot Low-Bandwidth Upgrades

Target hosts operate in bandwidth-constrained sites. Built an optimized single-pass upgrade mechanism to execute seamlessly despite limitations.

HARDWARE PREREQUISITES

Strict Memory Eligibility Rules

Upgrades on resource-constrained hardware posed system stability risks. Enforced strict pre-checks to exclude any host with less than 10 GB of RAM.

TESTING & RELIABILITY

99.999% Overall Success Rate

Achieved 99.999% success rate in production. 0.001% failures during dev environment testing steered building guardrails for manual intervention. 12K+ hosts upgraded successfully resulting in cost savings.

EXECUTION EFFICIENCY

40-Minute End-to-End Process

Entire upgrade process took 40 minutes end-to-end to return the host safely back into production environment.

PREVENTATIVE PRECHECKS

Fail-Early Network Safeguards

0.001% failures helped build a plethora of prechecks to fail early in the process, avoiding losing host connectivity over network.

OBSERVABILITY & TRACING

Log Live Stream & Tracing

Entire process streams live back to Log Dashboard with trace availability, helping operators escalate outages early.

TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Challenges and Takeaways

AUTOMATION & RECOVERY

Automate Error-Prone Manual Steps

Manual intervention during failed upgrades was prone to error. Developed auto-detection to resume execution from exact failure step.

SYSTEM RESILIENCE

Guardrails for Distributed Systems

Distributed systems involve numerous moving components. Overcame false assumptions regarding infrastructure resilience by engineering dedicated guardrails.

NETWORK OPTIMIZATION

One-Shot Low-Bandwidth Upgrades

Target hosts operate in bandwidth-constrained sites. Built an optimized single-pass upgrade mechanism to execute seamlessly despite limitations.

HARDWARE PREREQUISITES

Strict Memory Eligibility Rules

Upgrades on resource-constrained hardware posed system stability risks. Enforced strict pre-checks to exclude any host with less than 10 GB of RAM.

TESTING & RELIABILITY

99.999% Overall Success Rate

Achieved 99.999% success rate in production. 0.001% failures during dev environment testing steered building guardrails for manual intervention. 12K+ hosts upgraded successfully resulting in cost savings.

EXECUTION EFFICIENCY

40-Minute End-to-End Process

Entire upgrade process took 40 minutes end-to-end to return the host safely back into production environment.

PREVENTATIVE PRECHECKS

Fail-Early Network Safeguards

0.001% failures helped build a plethora of prechecks to fail early in the process, avoiding losing host connectivity over network.

OBSERVABILITY & TRACING

Log Live Stream & Tracing

Entire process streams live back to Log Dashboard with trace availability, helping operators escalate outages early.

TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Questions & Discussion

TECHNICAL DEEP DIVE

Infrastructure & Systems

Ask about disk repartitioning, dracut pre-pivot execution, automation, host reboots, or fallback mechanisms.

pkolakkar@bloomberg.net
tizod@bloomberg.net

CONNECT WITH US

Tech at Bloomberg

Explore our engineering culture, open-source projects, technical blogs, and career opportunities at TechAtBloomberg.com.

FEEDBACK & EXCHANGE

Share Your Insights

We welcome your feedback, questions, and shared experiences on executing zero-downtime OS upgrades at scale.

TechAtBloomberg.com

© 2026 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering