

# Cisco Nexus (NX-OS) High Availability

---

# Contents

Introduction ..... 3

System Manager ..... 3

Message & Transaction service (MTS) ..... 4

Persistent Storage Service (PSS) ..... 4

Software Maintenance Update (SMU) ..... 8

Nexus SMU Install Commands ..... 9

Graceful Insertion and Removal (GIR) - Maintenance mode ..... 9

In-Service Software Upgrade (ISSU)..... 11

BIOS Version Compatibility ..... 12

FPGA/EPLD Version Compatibility ..... 12

Enhanced In-Service Software Upgrade (EISSU) ..... 14

Standard ISSU vs EISSU vs Reload Down Times ..... 15

Conclusion..... 16

Glossary..... 16

---

## Introduction

As network environments continue to grow and evolve at a rapid pace, the surge in connected devices and increasing bandwidth demands present serious challenges. Any system failure can lead to significant disruptions or complete outages. To minimize downtime and ensure uninterrupted network operations, achieving 99.999% service availability is essential. Meeting this standard requires continuous network upgrades and staying current with the latest software enhancements to maintain optimal performance and resilience.

In today's digital economy, enterprises especially those in critical sectors like banking face stringent Service Level Agreements (SLAs) for system uptime and service availability. For many financial institutions, the acceptable downtime is measured in mere minutes per year; for example, an SLA of "five nines" (99.999% availability) translates to less than 5 minutes of downtime annually. This level of reliability is not just a competitive differentiator but a regulatory and business necessity, as even brief outages can result in significant financial losses, regulatory penalties, and reputational harm. Cisco provides high availability in data center switches to maintain operational continuity and prevent downtime. High Availability (HA) ensures the network remains operational with minimal interruption, even in the event of component failures, thereby supporting business continuity.

To achieve such demanding SLAs, technologies like Cisco NX-OS High Availability (HA) are essential. NX-OS HA architectures are engineered to minimize service disruption, ensuring that network control and data-plane operations continue uninterrupted even during software upgrades, hardware failures, or other maintenance events. Features such as stateful process restart, in-service software upgrades (ISSU), and resilient active/standby container models allow organizations to perform critical maintenance and upgrades without affecting end-user connectivity or application performance.

By minimizing downtime, HA ensures critical applications and services remain accessible, reducing operational costs, preventing revenue loss, and enhancing productivity. It also improves user experience by providing a reliable and consistent infrastructure. Additionally, HA, combined with disaster recovery, helps organizations prepare for and recover from major IT incidents.

HA solutions leverage fault tolerance mechanisms, such as redundant components and failover capabilities, to ensure uninterrupted operations. Features like dual supervisors in Cisco Nexus switches enable seamless switchover during failures, while isolating critical processes to prevent localized issues from affecting the entire system.

By combining robust HA mechanisms with enterprise-grade SLAs, Cisco NX-OS empowers organizations to confidently meet the strictest uptime requirements. This not only supports business continuity and customer trust but also enables rapid innovation and adaptability in fast-paced, high-stakes environments like banking. High availability is not just a technical feature; it is a fundamental enabler of modern enterprise operations and digital trust.

NX-OS HA Architecture and three important pieces of NX-OS that contribute to HA are,

## System Manager

Central coordinator of system processes and services. Oversees the initialization, monitoring, and lifecycle management of all services running on the device. Detects failures in services and triggers appropriate recovery mechanisms (such as automatic restarts or failover procedures). Works closely with the HA infrastructure to ensure orderly service operation during upgrades, reloads, or switchover scenarios.

---

Enforces dependencies and orchestrates service startup sequences, helping maintain system consistency and integrity.

By continuously monitoring and controlling services, the System Manager ensures that the system remains operational and stable even during transitions or partial failures.

## Message & Transaction service (MTS)

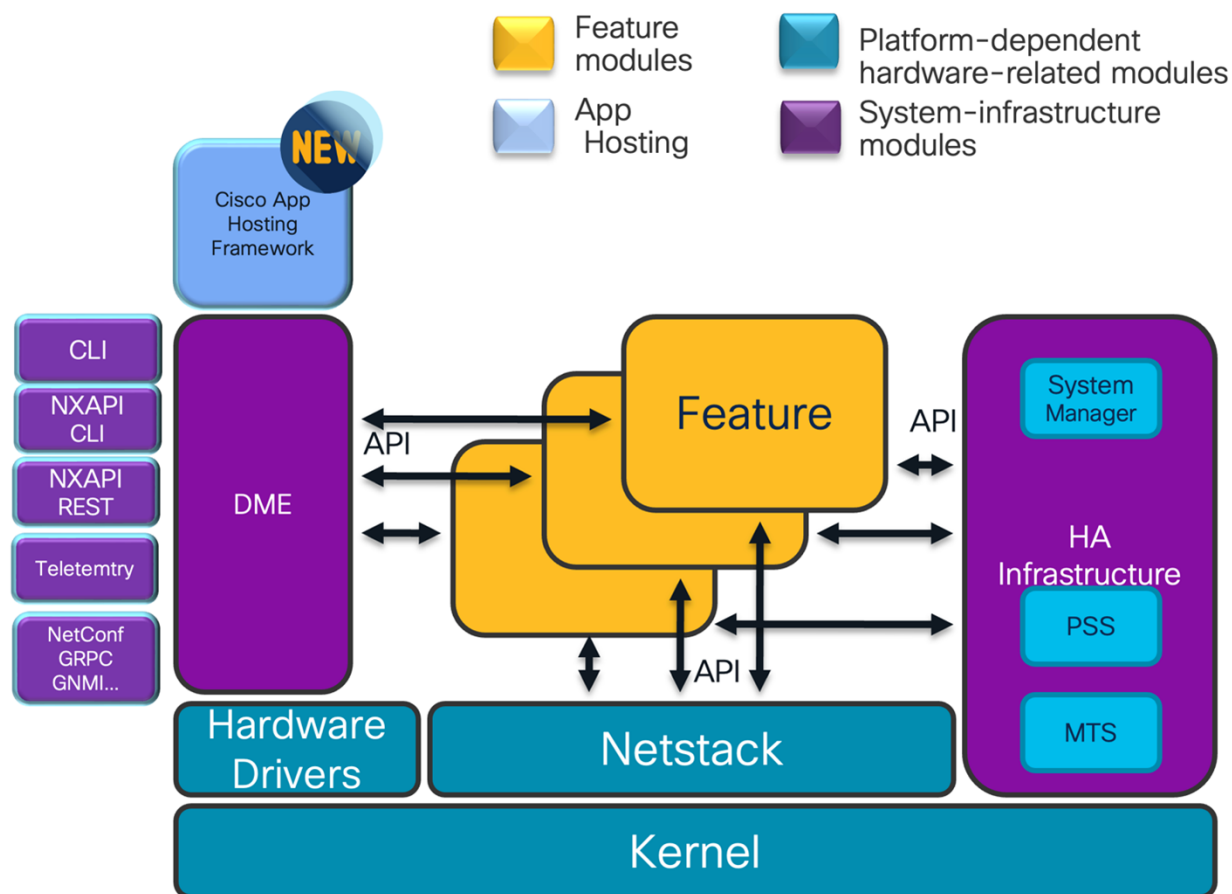
Manages all internal messaging and transactions between distributed system modules while ensuring persistent communication. Provides a reliable and durable communication framework for various functional modules, enabling them to exchange state, commands, and status efficiently. Supports transactional semantics to ensure consistent and recoverable state changes across modules, even in the event of failures.

By incorporating persistence, the system retains critical messaging and state information, allowing for seamless recovery and continuity in HA environments. This persistence enhances modularity by enabling loosely coupled services to operate independently while staying synchronized. With MTS managing robust, transactional, and persistent communication, system components can recover independently while ensuring overall system consistency and reliability—key requirements for HA systems.

## Persistent Storage Service (PSS)

Maintains critical runtime data across service restarts and system transitions. Stores essential service states, configurations, and session information. Allows services to resume from the exact point of interruption, rather than restarting from scratch. Ensures that temporary events such as process crashes or upgrades do not lead to data loss or service disruption.

By preserving state persistently, PSS enables quick recovery of services without impacting the user experience or requiring full system reboots—ensuring seamless high availability.



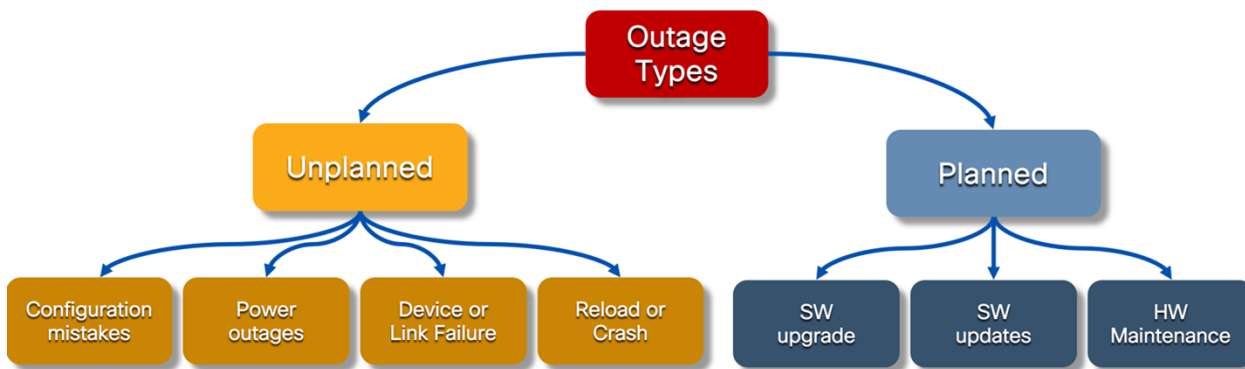
**Figure 1.**  
NX-OS Architecture

Key architectural principles of HA are:

- Fully Modular System Design
  - Enables independent module management and failure isolation.
  - Each feature is like a self-contained piece that can run independently.
  - Supports better scalability and easier maintenance.
- Separation of Control Plane and Data Plane
  - Ensures data forwarding continues even during control plane failures or restarts.
  - Separates control (decision-making) from data (actual traffic).
- Service Restart-ability.
  - Individual processes or features can restart without affecting the entire system or disrupting traffic.
  - Can restart individual services without restarting the whole device.
- Non-Disruptive Upgrades (ISSU/SSO)
  - Support for In-Service Software Upgrades (ISSU) and Stateful Switchover (SSO) ensures minimal downtime during software updates.

- Dual-supervisor switches support SSO (stateful switchover), ensuring no downtime during switchovers.
- Model-Driven Architecture
  - Enables programmability and automation through open APIs like REST, gRPC, and GNMI.
- Persistent Storage Service (PSS)
  - Maintains the state of services and enables rapid recovery after process restarts or upgrades.
- Robust Inter-Process Messaging (MTS)
  - Ensures reliable communication between modular components.
- System Manager for Orchestration
  - Coordinates the lifecycle and health of system processes for efficient management.
- Platform-Aware Hardware Abstraction
  - Enables HA features to interact efficiently with hardware without being tightly coupled to specific platforms.
- Support for Application Hosting
  - Embeds flexibility for running custom apps and telemetry directly on the switch.
  - Capable of app hosting and using modern APIs like REST, gRPC, etc.

To maintain HA of network, it is very important to understand the types of outages that we may be getting into in a live network with hundreds if not thousands of switches. The following covers a range of outages.



**Figure 2.**  
Network Switch Outage Types

Unplanned outages are unexpected disruptions that occur without prior scheduling or notice. They often result in service degradation, downtime, or operational instability. These events are reactive in nature and require immediate attention to restore normal function.






Unplanned outages are costly in terms of time, customer experience, and operational efficiency. Proactive monitoring, redundancy, and failover mechanisms are essential to minimize their impact. Building a resilient architecture is key to mitigating unplanned disruptions.

**Table 1.** Outage Types & Mitigation Strategies

Outage Type	Root Cause	Typical Impact	Mitigation Strategies
<b>Crashes</b>	Bugs, memory leaks, process failure	Traffic loss, control plane downtime	<ul style="list-style-type: none"> <li>• SMUs, ISSU/SSO, logging, proactive monitoring.</li> <li>• Regular software updates with SMUs or patches.</li> <li>• Enable core dumps and logging to diagnose issues post-crash.</li> <li>• Use features like Stateful Switch Over (SSO) and Non-Stop Forwarding (NSF) for redundancy.</li> <li>• Monitor CPU/memory usage using telemetry.</li> </ul>
<b>Power outages</b>	Electrical failure, PSU issues	Device shutdown, full outage	<ul style="list-style-type: none"> <li>• Deploy dual power supplies with separate circuits.</li> <li>• Use Redundant Power Systems (RPS).</li> <li>• Ensure all configurations are written (copy run start).</li> <li>• Integrate power monitoring and alerting systems.</li> </ul>
<b>Configuration errors</b>	Human error, bad scripts, stale scripts	Loss of traffic, security risks	<ul style="list-style-type: none"> <li>• Use configuration rollback/checkpointing features in NX-OS.</li> <li>• Implement GIR (Graceful Insertion and Removal) before changes.</li> <li>• Validate changes in a staging/lab environment.</li> <li>• Apply Role-Based Access Control (RBAC) to limit critical commands.</li> <li>• Leverage automation platforms like Cisco NDFC for controlled deployments.</li> </ul>
<b>Device or link failures</b>	Module, line card failure, Fiber optic issues, bad transceivers (copper, fiber)	Loss of traffic	<ul style="list-style-type: none"> <li>• Redundant Links (vPC, Port-Channels).</li> <li>• Redundant Devices.</li> <li>• Link State Tracking / Monitoring.</li> <li>• Auto-Recovery Features.</li> <li>• Telemetry &amp; Alerting.</li> </ul>

In addition to above mitigation strategies, Cisco incorporates following well established and proven network operations to minimize unplanned outages and downtime.

**Table 2.** Operational Strategies

Operational Strategy	Description
 <b>Proactive Monitoring</b>	Use tools like telemetry, SNMP, and syslog to detect anomalies early.
 <b>Redundancy</b>	Deploy dual supervisors, PSUs, and network paths for failover scenarios.
 <b>Validation Tools</b>	Use cisco modelling labs (CML), simulation tools, lab testing, and pre-deployment validations.
 <b>Automated Recovery</b>	Script failover and service restoration procedures for rapid response.
 <b>Resilient Architecture</b>	Design high availability topologies (for example, vPC, fabric spine-leaf).

Planned outages are intentional, scheduled disruptions to network services for the purpose of performing upgrades, maintenance, or infrastructure changes. These events are communicated in advance, allowing for preparation to minimize risk and impact.

Designing a highly redundant network involves implementing various strategies to ensure uninterrupted operation and minimize downtime, with the goal of achieving zero downtime. Key approaches include multiple network paths between critical points, load balancing, and dual-homed devices and switches. Cisco supported hitless upgrades for data centers built with Cisco Nexus switches in both Cisco Application Centric Infrastructure (ACI) and NX-OS operating models. The following explores the hitless upgrade options available in Cisco NX-OS and best practices recommended.

Planned outages are broadly categorized into following types. they are software upgrades, hardware upgrades and maintenance mode.

Software upgrades involve updating the NX-OS image, patching vulnerabilities, or applying bug fixes (SMUs).

Hardware upgrades include physical interventions such as replacing hardware, adding/removing modules, or relocating equipment.

Software Maintenance Update (SMU)

SMU is a quick patch type of software updates designed to address specific critical issues and security vulnerabilities in a software system. These updates are released typically to ensure the continued reliability, security, and performance of the software system.

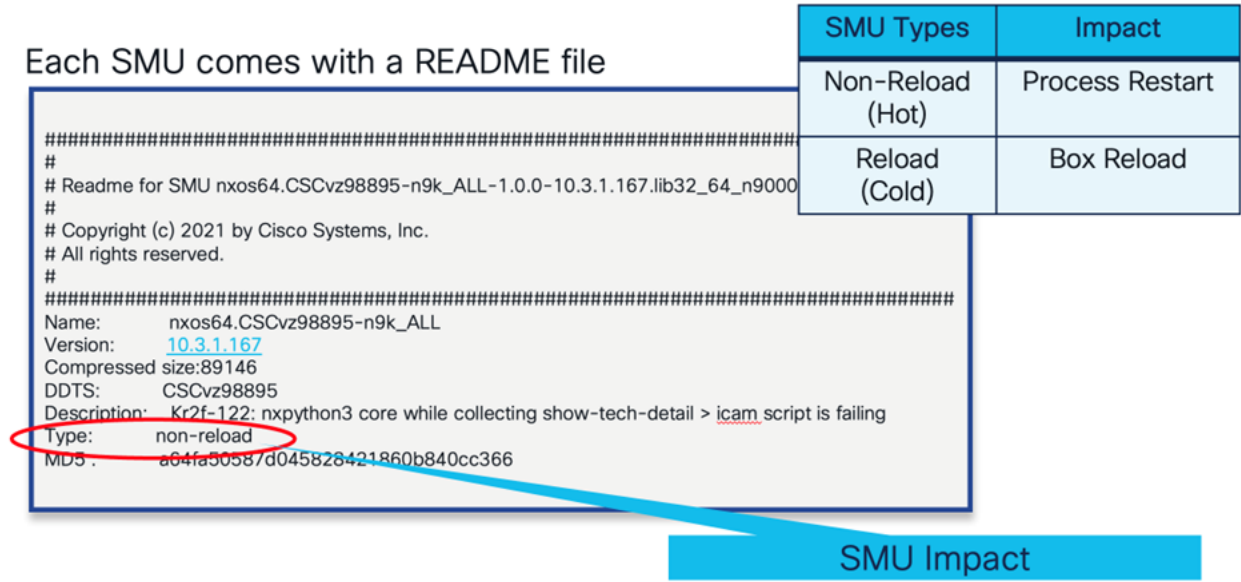


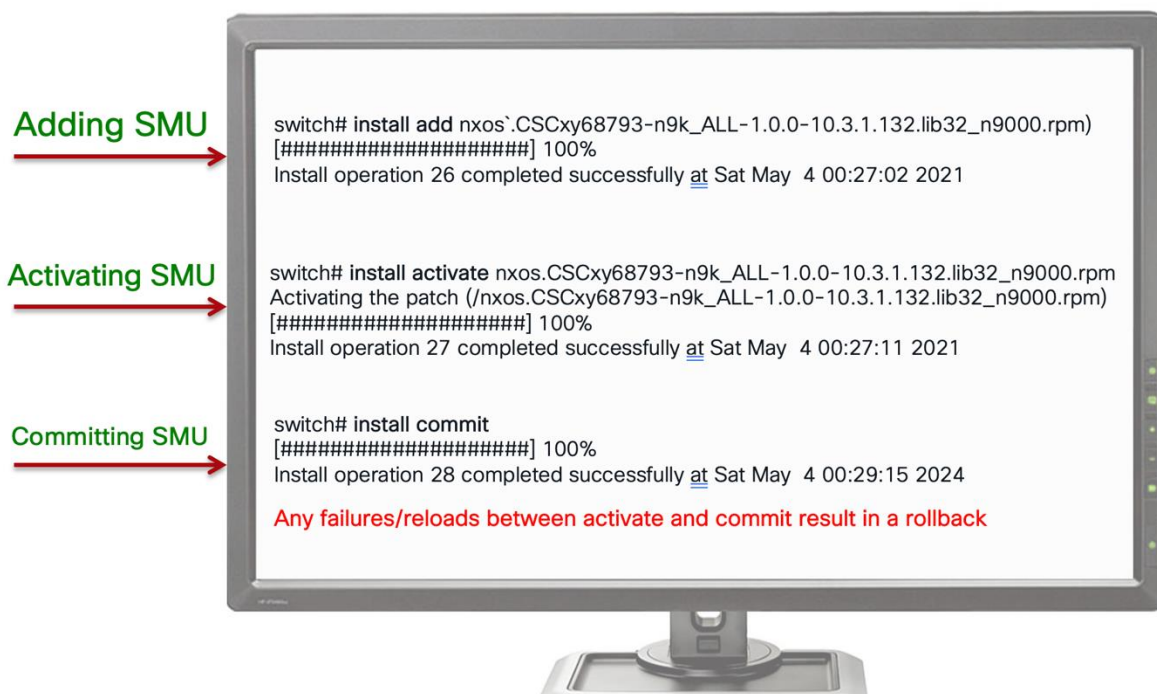
Figure 3. NX-OS SMU

SMU will come in two modes HOT (Non-reload) and COLD (reload). Libraries that are used across processes and needs a fix to address a vulnerability of system or impacting functional issue will be patched with full reload of the system, but most of the SMU updates that involves process binaries will come in HOT mode and NXOS services will take care of installing the patch without reloading the system and by simply



reloading respective process or service in the system. Internally, the service checkpoints its state in PSS, stops via job scheduler, updates the file system with SMU files, and restarts, ensuring no impact on the data plane, with full state recovery and a total recovery time of ~10ms.

## Nexus SMU Install Commands



**Figure 4.**  
NX-OS SMU Install Commands

## Graceful Insertion and Removal (GIR) - Maintenance mode

GIR is a powerful feature in NX-OS that enables you to safely perform maintenance task. This mode allows certain hardware and software processes to be disabled or isolated so that maintenance tasks, such as software upgrades, hardware replacement, and troubleshooting, can be performed without affecting the normal operation of the rest of the network. GIR uses redundant paths in the network to gracefully remove a device from an active network, place it out of service, and insert it back into service when the maintenance is complete.

As modular chassis does not support Non-Disruptive In Service Software Upgrade (ND-ISSU), Graceful Insertion and Removal model can be used in CloudScale Edge of Rack (CS EOR) to achieve non-disruptive upgrades. Following NX-OS CLIs help with maintenance mode.

**System mode maintenance command with optional parameters:**

```
switch(config)# system mode maintenance [always-use-custom-profile | dont-generate-profile | on-reload reset-reason reason | shutdown | timeout value]
```

**Maintenance mode with all protocols isolated:**

```
switch(config)# system mode maintenance
```

**Enter maintenance mode shutting down all protocols and interfaces:**

```
switch(config)# system mode maintenance shutdown
```

**Check current system mode and timer:**

```
switch# show system mode
```

**Create a maintenance mode profile to specify protocol shutdown/isolation commands:**

```
switch(config)# configure profile maintenance-mode type admin
```

**To automatically boot into maintenance mode on reload due to fatal error:**

```
switch(config)# system mode maintenance on-reload reset-reason
```

```
fatal_error
```

**To apply a custom maintenance mode profile always:**

```
switch(config)# system mode maintenance always-use-custom-profile
```

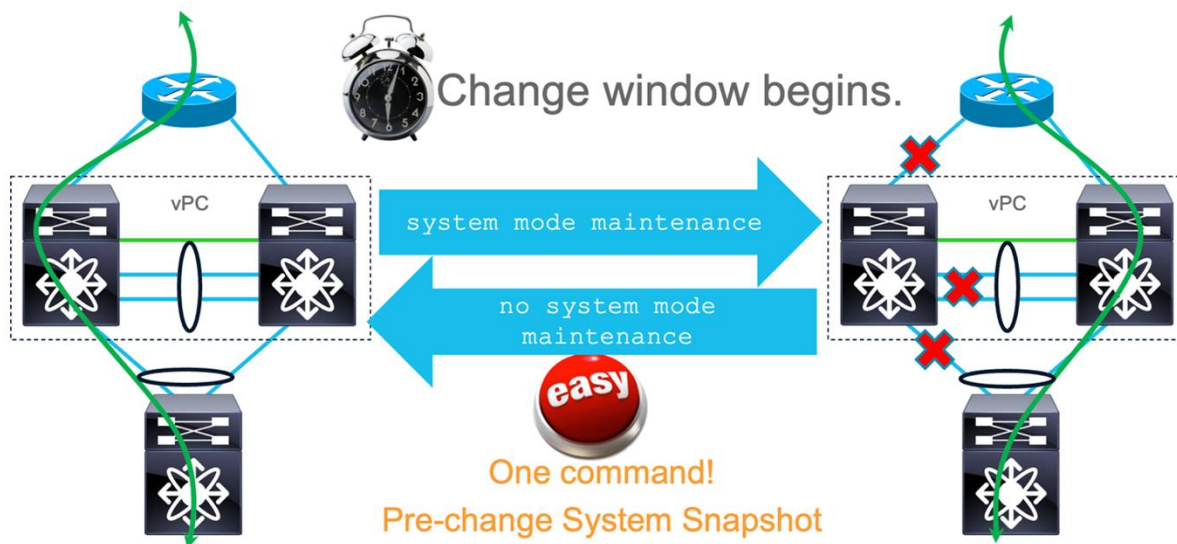
**To enter maintenance mode non-interactively (no prompts):**

```
switch(config)# system mode maintenance non-interactive
```

**Exit maintenance mode**

```
switch(config)# system mode normal
```

GIR can be performed on all vPC pairs one by one or few vPC pairs based on the need.



**Figure 5.**  
GIR Maintenance mode

Specific to GIR, some vendors only support a subset of protocols, such as Border Gateway Protocol (BGP) and Multi Chassis Link Aggregation Group (MLAG) for maintenance modes of operation. NX-OS isolates devices from the network with support for all Layer-3 protocols, including:

- Border Gateway Protocol (BGP)
- Enhanced Interior Gateway Routing Protocol (EIGRP)
- Intermediate System-to-Intermediate System (IS-IS)
- Open Shortest Path First (OSPF)
- Protocol Independent Multicast (PIM)
- Routing Information Protocol (RIP)








- Multi-Chassis Link Aggregation (MLAG)

## In-Service Software Upgrade (ISSU)

In Cisco Nexus fixed chassis (CloudScale Top of Rack - CS TOR) switches, ISSU (In-Service Software Upgrade) enables software upgrades to be performed without interrupting the ongoing network services. A major advantage of ISSU is that it ensures zero packet loss, meaning the data plane continues to forward traffic seamlessly throughout the upgrade process. However, it is important to note that ISSU does introduce a brief control plane disruption, typically lasting between 90 to 120 seconds. During this time, Layer 3 protocol exchanges with neighboring devices are temporarily paused, but they are automatically re-established once the upgrade completes. Because the data plane remains operational and unaffected during this control plane downtime, data center workloads and applications continue to function normally. This makes ISSU a critical feature for environments that demand high availability, such as enterprise networks and data centers, where maintaining uninterrupted network connectivity is essential.

One of the most critical reasons for using ISSU in Cisco Nexus switches is to maintain continuous service availability during software upgrades. In modern data centers and enterprise networks, even brief outages can lead to significant disruptions, lost revenue, or degraded customer experience. ISSU enables network administrators to apply updates without interrupting active data traffic, ensuring that applications and services remain online. This capability is essential in environments that require high up time, such as financial services, healthcare, or cloud service providers.

**Table 3.** Cookbook

Good practice cookbook	
 <b>Release Notes</b>	Review ISSU support in the release notes if planning a non-disruptive upgrade.
 <b>Pre-checks</b>	Run <b>show install all impact</b> command to, check compatibility and validate image.
 <b>Pre-stage images</b>	Upload NX-OS and SMUs during off-hours, outside install window.
 <b>Lab test upgrades</b>	Rehearse in a staging environment.
 <b>Define maintenance window</b>	Ensure window is long enough for rollback if needed.
 <b>Backup configurations</b>	Use copy run start and export configurations and logs.
 <b>Use High Availability</b>	vPC, SSO, NSF, GIR to maintain traffic and minimize impact.

Run manual check on the impact of upgrade to a target image. Running this pre-check prevents upgrade failures due to hardware or feature incompatibility, ensuring your system will boot correctly after the upgrade. It also confirms high-availability and ISSU options for a non-disruptive upgrade, allowing you to plan accordingly. Additionally, it provides time to address issues such as missing space or incompatible modules before your scheduled maintenance window, reducing the risk of downtime or unexpected complications during the upgrade process.

```
switch# show install all impact nxos bootflash:nxos64.10.6.1.F.bin
Installer will perform impact only check. Please wait.

Verifying image bootflash:/nxos64.10.3.1.F.bin for boot variable "nxos".
[#####] 100% -- SUCCESS

Verifying image type.
[#####] 100% -- SUCCESS

Preparing "nxos" version info using image bootflash:/nxos64.10.6.1.F.bin.
[#####] 100% -- SUCCESS

Preparing "bios" version info using image bootflash:/nxos64.10.6.1.F.bin.
[#####] 100% -- SUCCESS

Notifying services about system upgrade.
[#####] 100% -- SUCCESS
```

**Figure 6.**  
NX-OS Version impact

Impact check performs a dry run for,

- Compatibility with hardware (line cards, supervisors, fabric modules).
- Available space in/bootflash.
- Required kickstart/system image match (in older versions).
- Impact on high-availability and ISSU (In-Service Software Upgrade) support.
- Compatibility with current features, packages, and licenses.
- Check Spanning-tree impact on ISSU using the **show spanning-tree issu-impact** command.
- Perform compatibility checks before ISSU to ensure BIOS and FPGA/EPLD versions are supported.
- Upgrade BIOS and EPLD firmware as needed before or during ISSU to avoid disruptive reloads.

## BIOS Version Compatibility

The BIOS version on standby and active supervisors should match to avoid issues. For example, when upgrading a Nexus 9800 modular switch, the standby supervisor BIOS must be upgraded to match the active supervisor BIOS version. The upgrade process includes a compatibility check before installation, and a reload is required for disruptive upgrades. BIOS upgrades are necessary to support the new NXOS image and to avoid PCI error issues during ISSU. Some platforms require manual BIOS upgrades before ISSU to avoid disruptive processes.

## FPGA/EPLD Version Compatibility

FPGA and EPLD firmware versions must be compatible with the NXOS version. If FPGA or EPLD versions are too far out of date, interfaces may fail to come up after upgrade. The running and expected versions can be verified using commands such as 'show install impact epld bootflash:<nxos-image>' or specific firmware queries. EPLD upgrades are often required alongside NXOS upgrades to fix issues like FPGA CRC errors that can cause system reloads. NXOS requires that EPLD firmware version is the same or earlier than the NXOS version. Upgrading NXOS alone without EPLD upgrade is insufficient.

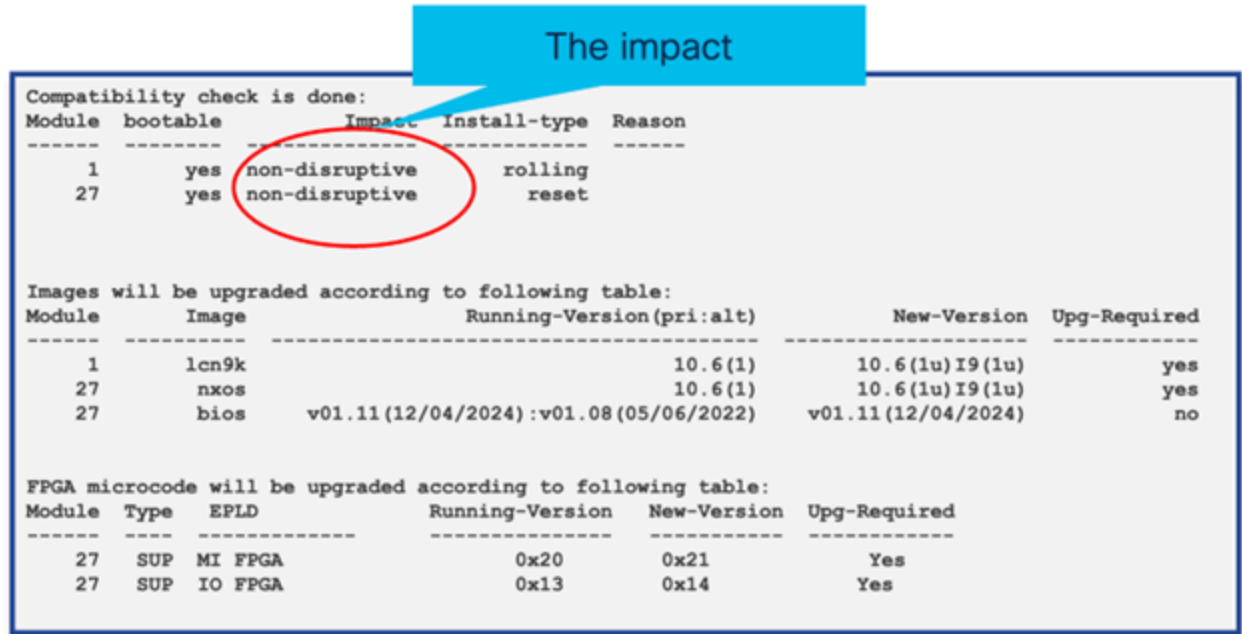


Figure 7.  
NX-OS ISSU Impact

Following install command help upgrading to target image.

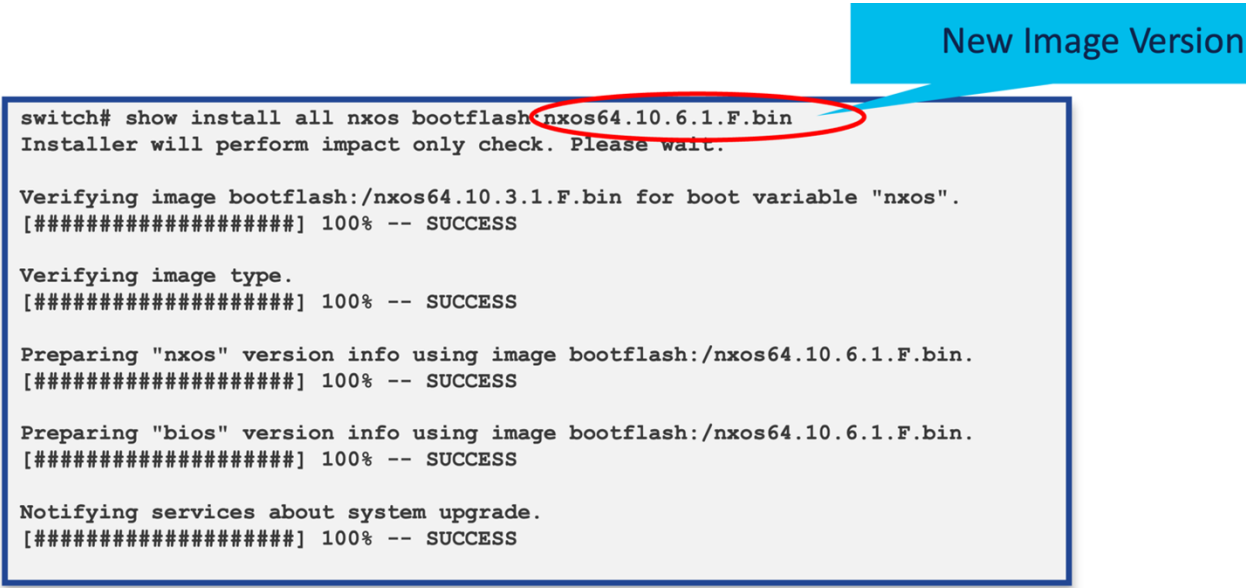
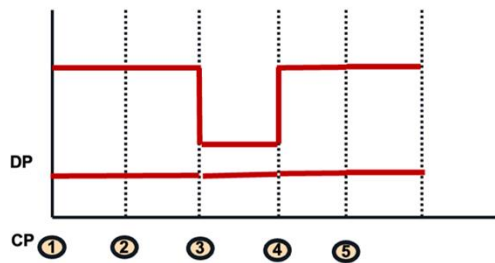


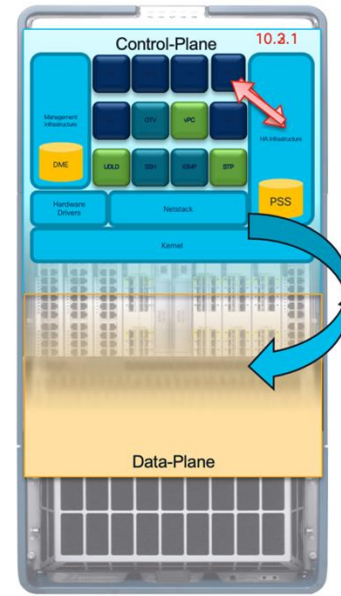
Figure 8.  
NX-OS ISSU Install

During the pre-upgrade check, the configuration is locked, and processes save their states to Persistent Storage Service (PSS). The NX-OS kernel then creates a container with a new image and restores the saved state into this new container. The control plane will be unavailable for 90-120 seconds during this process, but there will be no impact on the data plane.





1. Pre Upgrade Check
  - Config Locked
  - Stable Network
2. Save State
3. kexec to new Image
4. Restore control plane from saved State
5. Reconcile with Data Plane



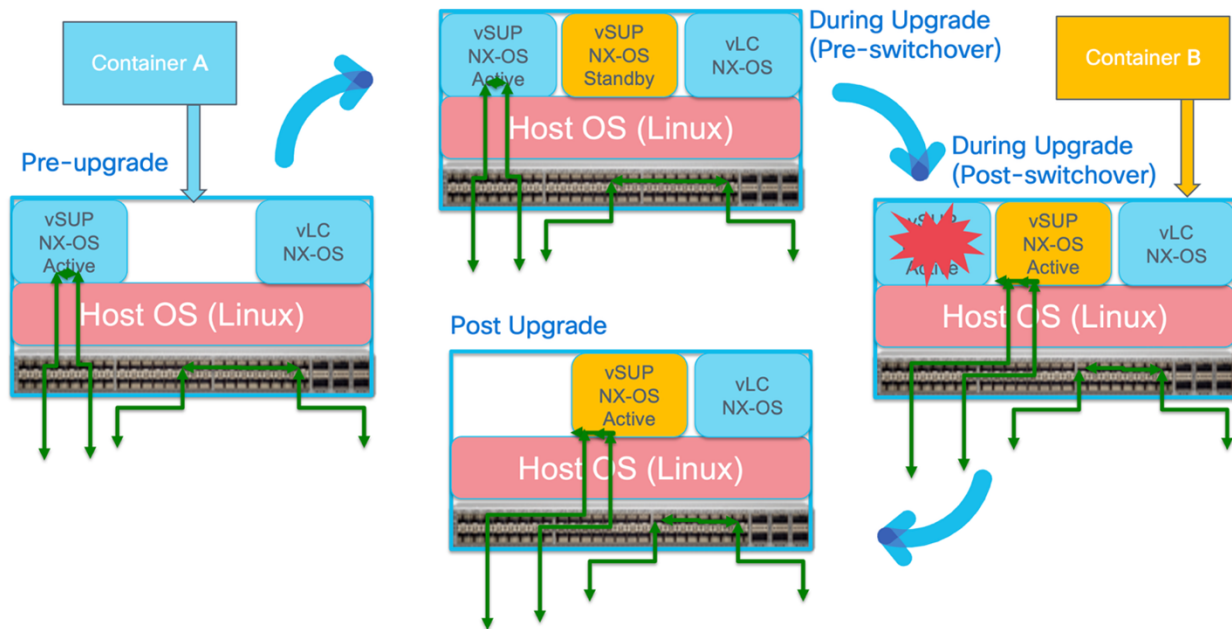
**Figure 9.**  
NX-OS ISSU Transition

## Enhanced In-Service Software Upgrade (EISSU)

Enhanced In-Service Software Upgrade (EISSU) is a next-generation evolution of ISSU that leverages container technology integrated within NX-OS. Designed to meet the demands of highly available and complex network environments, EISSU extends the benefits of traditional ISSU by delivering even greater efficiency and reliability during software upgrades. Instead of directly replacing the active supervisor software, EISSU creates a secondary virtual supervisor engine in a container, loads the new image into it, and then seamlessly switches it with the current image. This innovative method ensures there is no impact on the data plane, maintaining zero packet loss throughout the upgrade process. Furthermore, it significantly shortens the control plane downtime from a minute in standard ISSU to just 3-6 seconds, making EISSU an ideal solution for mission-critical infrastructure where minimizing disruption is paramount.

From a Layer 3 standpoint, both ISSU and EISSU ensure that all routing protocols continue operating smoothly during the upgrade process by supporting graceful restart, also referred to as Nonstop Forwarding (NSF). This mechanism allows the control plane to restart without disrupting the forwarding of packets. On the Layer 2 side, key protocols such as Spanning Tree Protocol (STP) and Virtual Port Channel (vPC) are fully supported during ISSU and EISSU operations. vPC enables two physical Nexus switches to function as a single logical switch to any connected Layer 2 device, ensuring redundancy and load balancing. Meanwhile, STP continues to perform its critical role in loop prevention by maintaining a loop-free topology in networks where multiple redundant paths exist. Together, these protocol capabilities allow ISSU and EISSU to deliver seamless upgrades without compromising Layer 2 or Layer 3 network stability.

As shown in the diagram below, the upgrade process begins with installing the NX-OS version2.bin software on the existing container, referred to as Container A. Once the new image is in place, a new container referred as Container B is spawned and booted up with the NX-OS (Version2) software version. After successful initialization and validation, Container B assumes an active role in the environment. At this stage, the system transitions to the new version of NX-OS, resulting in an estimated brief impact of 3-6 seconds to control plane traffic. Following this successful transition and once operational stability is confirmed, the original Container A is safely destroyed, completing the upgrade process.



**Figure 10.**  
NX-OS EISSU Transition

One might wonder what happens if the kernel itself needs to be patched—does that require a full system reload? Interestingly, with NX-OS version 10.2(2)F and later, EISSU has a built-in fallback mechanism to manage such cases. If a kernel patch is necessary, EISSU will automatically switch to a standard ISSU process and proceed with the upgrade using Zero Packet Loss (ZPL) methodology. The key distinction in this scenario is that the control plane will experience a longer downtime compared to an EISSU upgrade, but the upgrade will still be completed seamlessly and without impacting data plane traffic.

Cisco Nexus 9300 Series switches from the GX2A and GX2B families come with Enhanced In-Service Software Upgrade (EISSU) enabled right out of the box. Additionally, EISSU is also turned on by default for Nexus 9300 Series GX and FX3 models running NX-OS version 10.3(3)F or later. However, for earlier generations such as the FX and FX2 models, enabling EISSU requires an extra configuration step. This involves manually entering a specific command and then performing a system reload to activate the feature. As shown below, there is no down time for data path traffic except for control plane.

### Standard ISSU vs EISSU vs Reload Down Times

Following table shows the upgrade type and down time of control plane (CP) and data plane (DP) respectively.

**Table 4.** CP/DP Down Times

Upgrade Type	CP Down Time	DP Down Time
Standard-ISSU	90-120 Seconds	0 Seconds
Enhanced-ISSU	3-6 Seconds	0 Seconds
Switch Reload	5 minutes (scale dependent)	Same as control Plane (CP)

---

## Conclusion

High Availability (HA) in Cisco NX-OS is essential for maintaining continuous network operations in modern data centers. With features like supervisor redundancy, vPC, ISSU, modular process architecture, and graceful restart protocols, NX-OS provides a resilient platform that minimizes downtime and ensures service continuity. By leveraging these HA capabilities and following best practices, organizations can build highly reliable and fault-tolerant network infrastructures that meet the demands of mission-critical applications.

## Glossary

ACI	Application Centric Infrastructure
BGP	Border Gateway Protocol
CML	Cisco Modelling Labs
CS	Cloud Scale
CP	Control Plane
DP	Data Plane
EIGRP	Enhanced Interior Gateway Routing Protocol
EISSU	Enhanced In Service Software Upgrade
EOR	Edge of the Rack
EPLD	Electronic Programmable Logic Device
GIR	Graceful Insertion and Removal
HA	High Availability
IS-IS	Intermediate System-to-Intermediate System
ISSU	In Service Software Upgrade
MLAG	Multi-Chassis Link Aggregation
MTS	Message & Transaction Service
NSF	Non-Stop Forwarding
NX-OS	Cisco Nexus Operating System
OSPF	Open Shortest Path First
PIM	Protocol Independent Multicast
PSS	Persistent Storage Service
RBAC	Role-Based Access Control
RIP	Routing Information Protocol
RPS	Redundant Power Systems
SLA	Service Level Agreements



---

SMU	Software Management Update
SSO	Stateful Switchover
STP	Spanning Tree Protocol
TOR	Top of the Rack
vPC	Virtual Private Channel
ZPL	Zero Packet Loss

---

**Americas Headquarters**

Cisco Systems, Inc.  
San Jose, CA

**Asia Pacific Headquarters**

Cisco Systems (USA) Pte. Ltd.  
Singapore

**Europe Headquarters**

Cisco Systems International BV Amsterdam,  
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at <https://www.cisco.com/go/offices>.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Printed in USA

Cxx-xxxxxx-xx 01/22