

Effortless OS Upgrade with Anuta ATOM

OS upgrade is an activity that can quickly gather complications, confusion, and delays as it rolls forward. Most problems do not even pop on the network manager's radar until they become too unwieldy.

The current OS upgrade process has the following limitations:

- Lack of a simple and speedy upgrade process
- A lot of manual tasks, which are usually unaudited
- Dependencies on external and internal tools like ticketing systems and image servers
- Human errors like a wrong image upload lead to critical loss of network maintenance windows which are hard to get approved
- Every engineer must synchronously operate numerous command-line interfaces, track errors and alerts, perform rollbacks, and restart failed flows in the current process

The challenges that emerge during an upgrade can be broadly classified into two categories.

Device-Specific challenges:

- No two vendors are the same. Each vendor has a different set of procedures to upgrade one's devices
- Also, each device platform for a given vendor can differ on upgrade procedures
- Access to devices can differ, such as – SSH, Telnet, API, and others
- Various OS Image transfer protocols can add complexity to this area – Like FTP, TFTP, SCP, SFTP, USB
- On some devices, the management station comes into play instead of a device-level execution
- Multi-stage upgrade – For certain vendors, an upgrade between 2 different versions might require a multi-stage upgrade. Such upgrades can entail intermediate version upgrades.
- Some features that might cover command syntax might change. Additional configuration could also be needed.
- Network outages might trigger a rollback of the upgrade. In some cases, the rollback procedures are different from the upgrade procedures.

Organization-Specific:

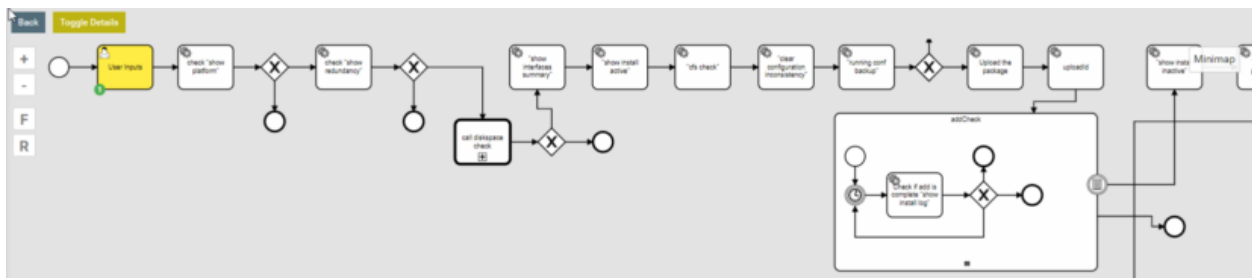
- Every organization has its own set of best practices or MoPs that consist of
 - Pre-checks, Post-checks, Rollbacks on error
 - Open tickets in the ticketing systems – this can manifest strongly in case of errors
 - User intervention at critical stages – such as review before device & reload in case of errors
- Device role (Core, Distribution, Security Gateway, Server Farm, etc.) and their impact to business in case of outage impact the MoPs.
- Redundant Pairs of devices do not get upgraded simultaneously.

- MoPs for critical devices with high business impact might be broken down into low-impact versus high-impact activities. For example, Image upload, which is a low-impact activity, can be done before a high-impact activity, such as device reload.
- Devices can be upgraded in parallel, primarily if organizations work on tight timelines.

Network OS Upgrades with Anuta ATOM

ATOM's workflow automation capabilities enable network administrators to automate manual processes and methods of procedures. These tasks often include pre-checks, post-checks, approval chains, integrations to OSS/BSS, ticketing/billing, and ITSM solutions.

Large networks typically have complex workflows consisting of hundreds of commands, integration to tens of network elements, and complex business logic to bind the various elements together. ATOM workflow automation has been designed not only to facilitate simple use cases but also to design and execute the most complicated use cases at scale.



Anuta ATOM has out of box support for Software Upgrade for the following platforms:

Vendor	Supported Platforms	
	Basic OS Upgrade	Advanced OS Upgrade
Juniper	Junos, MX, vMX, PTX, ACX, QFX, EX	Junos, MX
Cisco	IOS, NX-OS, IOS-XR, IOS-XE	
Fortinet	60E	

Single-Click Execution of OS Upgrades:

The out of box workflows can be executed on-demand with a single click or be scheduled for later. The network operator can monitor the workflow progress, the path it took, and the current status. Any inputs or approvals required will be initiated either in ATOM itself or triggered in the ITSM solution like ServiceNow. Just sit back and relax as ATOM runs an upgrade on all your devices.

Monitor and Audit OS Upgrades:

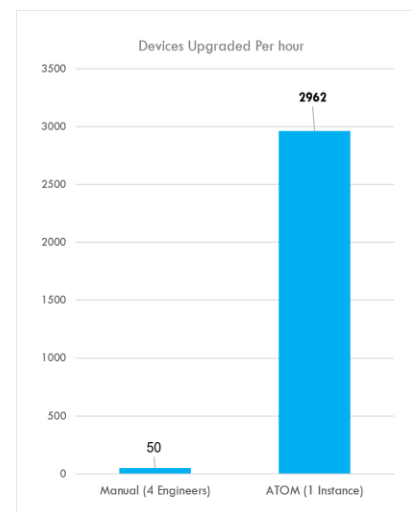
Anuta ATOM highlights every stage and helps operators understand its status as the workflow moves through the steps. A small icon at the top indicates what kind of a task ATOM is performing – whether it's a system task, user task, or something waiting for a timeout.

ATOM audits every action taken during the workflow – whether executing commands on the devices, executing REST APIs on integrated network elements, or handling user inputs provided during the workflow. ATOM also has a comprehensive workflow logging and monitoring functionality that helps operators pinpoint and debug any workflow. All workflows and sub-workflows are listed in a hierarchical format for easy viewing and traceability.

ATOM creates enough segregation with its RBAC feature, allowing network architects to create workflows and ease NetOps teams to execute them.

Massive Scale:

ATOM has been validated to upgrade 20,000 Juniper MX devices in a simulated environment in approximately 6 hours and 45 minutes - 2962 workflows/hour. The software upgrade consists of 49 prechecks, 49 post-checks, and 49 rollback procedures. Earlier, the manual process could upgrade only 50 devices in 1 hour by four engineers.



OSS/BSS Integration:

ATOM workflow can also integrate with any Ticketing/Billing/ITSM solutions such as ServiceNow, Jira, BMC remedy, etc. ATOM seamlessly integrates with all platforms such that the Open APIs can provide the input values required by the workflow.

Summary

ATOM allows network architects to create an upgrade workflow per the organization's demands and needs. There is no need for manually entering CLI commands and configurations. Operators receive customized alerts and know the issue instantly without wasting precious time.

The ATOM platform offers out-of-box software upgrade workflows for different vendors such that multi-vendor software upgrades can be combined into a single workflow. Post-Upgrade, ATOM can run the configuration compliance checks and remediate violations.

No matter how complex any step of the OS upgrade can get, ATOM has the power and clarity to make it simple. It makes an OS upgrade what it should ideally be – A quick switch where you plug in and play.

Additional Resources:

Effortless single click OS upgrade with Anuta ATOM Platform

<https://www.anutanetworks.com/effortless-single-click-os-upgrade-with-anuta-atom-platform/>

Massive Scale Workflow Automation with Anuta ATOM

<https://www.anutanetworks.com/massive-scale-workflow-automation/>

Network OS Upgrade – A quick and easy way to achieve it

<https://www.anutanetworks.com/network-os-upgrades-a-quick-and-easy-way-to-achieve-it/>

Software Upgrade on Junos MX Single RE using ATOM

1. SMU-junos user inputs
User has to provide the following inputs during Single RE Software Upgrade.
 - a. device
 - b. upgrade-OS-version
 - c. protocol
 - d. image-server-location
 - e. image-file-path
 - f. image-file-name
 - g. destination-path
 - h. username
 - i. password
2. Check if device is online
3. Check "*device type*"
4. Check "*show version*"
Command: *show version | match Junos:*
5. Verify image exists on device.
6. Check the disk space on device.
7. Upload the image on device
Command: *file copy http://"+serverlocation+"/"+imagepath+"/"+imagefile+"/"+destination+"/";*
8. Running conf backup on device locally
Command: *show configuration | save /var/tmp/Backupconfig.txt*
9. Run basic Pre-checks on device.
 - a. *show bgp neighbor*
 - b. *show arp no-resolve*
 - c. *show route protocol bgp-static*
 - d. *show route summary*
10. Rollback uncommitted changes on device.
11. Check "*request system or vmhost software add*"
Command: *"request "+type+" software add /"+destination+"/"+imagefile;*
12. Reboot Approval Form $\${device}$
13. Check "Reboot Device"
14. Check if device is online
15. Check "*show version after upgrade*"
 - a. *show version | match Junos:*
16. Post-checks
 - a. *show bgp neighbor*
 - b. *show arp no-resolve*
 - c. *show route protocol bgp-static*
 - d. *show route summary*
17. Upgrade Status

Software Upgrade on Junos MX Dual RE using ATOM

1. Junos MX SWIM User Inputs

- | | |
|-------------------------------|----------------|
| a. deviceId | f. hostname |
| b. junos_image_names | g. path |
| c. junos_rollback_image_names | h. file_system |
| d. peer_list | i. user |
| e. required_diskspace | J. password |

2. Check is device online or not

3. Check Master and Backup REs before upgrade

Command: *show chassis routing-engine*

4. Disk Space Check

Command: *show system storage invoke all-routing-engines | match*

If no space user needs to clean with this command: *request system storage cleanup re1/re0 no-confirm*

5. Verify the current version on REs

Command: *show version invoke-on all-routing-engines | no-more*

6. Upload Images - Upgrade and Rollback:

Command: *file copy scp://" + user + "@" + hostname + path + filename + " + reName + ":/var/tmp/" + filename + "!" + pwd*

7. Trigger config pull before image upgrade

8. Log copy of current configuration on the device before upgrade

Command: *show configuration | display set | no-more*

9. Execute Pre-Checks

- | | |
|--------------------------------|------------------------------|
| a. <i>show system alarms</i> | f. <i>show isis database</i> |
| b. <i>show interface terse</i> | g. <i>show mpls lsp</i> |
| c. <i>show rsvp session</i> | h. <i>show bgp summary</i> |
| d. <i>show bfd session</i> | i. <i>show route summary</i> |
| e. <i>show isis adjacency</i> | |

10. Approval Form to Upgrade \${deviceId}

11. Verify current version on REs after pre-checks approval

Command: *show version invoke-on all-routing-engines | no-more*

12. Rollback uncommitted changes

13. Disable GRES and NSR

14. Perform upgrade on first RE and checks

Command: *request "+isDeviceVMHost+" software add "+reName+"
/var/tmp/"+junos_image_names+" reboot
isDeviceVMHost either system or vmhost
reName either re0 or re1*

15. Perform upgrade on second RE and checks

16. Reactivate GRES and NSR

17. Check Master and Backup REs after upgrade

Command: *show chassis routing-engine*

18. Approval Form to Switch Mastership

19. Switch RE Membership

Command: *request chassis routing-engine master switch*

20: Clear device connection

21: Trigger config pull after image upgrade

22. Execute Post Checks

a. *show system alarms*

b. *show interface terse*

c. *show rsvp session*

d. *show bfd session*

e. *show isis adjacency*

f. *show isis database*

g. *show mpls lsp*

h. *show bgp summary*

i. *show route summary*

23. Log copy of current configuration on device after upgrade

Command: *show configuration | display set | no-more*

24. Get pre/post diff

25. Show diff for all protocols

26. Approval Form to Proceed

Approval form to proceed --> *approval_to_proceed == "false"*

Approval Form to retry or rollback or terminate --> *terminate == true"*

Software Upgrade on Cisco IOS-XE using ATOM

1. SMU-IOSEX user inputs
 - a. deviceid
 - b. Protocol - SFTP/TFTP/SCP
 - c. Image Server
 - d. Image Path
 - e. Config_Backup_file
 - f. Image File
 - g. Checksum

2. check for sufficient disk space


```
dir bootflash: | in free
```

Proceed if sufficient disk space is available

3. Pre-Checks
 - a. *show version*
 - b. *show platform*
 - c. *dir bootflash:*

4. Get base image


```
dir bootflash:Image/ | i .bin
```

5. Setting the Router to boot


```
config terminal
  config-register 0x2
  exit
!
```

```
request platform software package expand file bootflash:Image/<base-image>
request platform software package expand file stby-bootflash:Image/<base-image>
!
```

```
configure terminal
  boot system bootflash:Image/packages.conf
  exit
copy running-config startup-config
```

6. Backup running configuration


```
copy running-config tftp://<server>/<configfile>
```

7. Copy Image to Bootflash


```
copy tftp://<server>/<upgradefile> bootflash:Image/
```

8. Preparing for installation


```
config terminal
  redundancy
  mode sso
end
```

9. Get redundancy state


```
show redundancy
```


10. Save running configuration
copy running-config startup-config
11. Install the image
request platform software package install node file
bootflash:Image/<image_file> interface-module-delay 160
12. Check for the device to come online
13. Post-Checks
 - a. *show version*
 - b. *show platform*
 - c. *dir bootflash:*