

```

1  ---
2  # =====
3  # Playbook   : os_version_upgrade.yaml
4  # Description: Multi-vendor OS upgrade for devices flagged by Flow Weaver
5  #             as non-compliant. Supports Arista EOS, Cisco IOS XE, and
6  #             Juniper Junos. Target List is built dynamically by Workflow 2
7  #             from open ServiceNow tickets – no static inventory required.
8  #
9  # Extra vars (passed by Flow Weaver or manually):
10 #   target_devices : comma-separated hostnames, e.g. "ATL-LAB-DIST-CISCO-01,SIN-HUB-SPINE-AR-01"
11 #                 If not passed, the playbook runs against the full inventory group
12 # [upgrade_targets]
13 #   image_store    : base URL/path where OS images are staged
14 #                 e.g. "http://192.168.1.100/images" or "flash://"
15 #   upgrade_timeout : seconds to wait for reload (default: 600)
16 #
17 # Inventory group expected: [upgrade_targets]
18 # Each host should define:
19 #   ansible_network_os      : eos | ios | junos
20 #   ansible_host            : management IP
21 #   ansible_user / password : pulled from vault by Flow Weaver
22 #   desired_os_version      : target version (matches NetBox custom_fields.os_version)
23 #   os_image_filename       : image file to install (optional, can default per platform)
24 #
25 # Usage examples:
26 #   ansible-playbook os_version_upgrade.yaml -i inventory/hosts
27 #   ansible-playbook os_version_upgrade.yaml -i inventory/dynamic_hosts.ini \
28 #     -e "target_devices=ATL-LAB-DIST-CISCO-01,SEA-OFF-EDGE-CISCO-01"
29 # =====
30 - name: OS Version Upgrade – Pre-checks and staging
31   hosts: "{{ target_devices | default('upgrade_targets') }}"
32   gather_facts: false
33   any_errors_fatal: false
34
35   vars:
36     upgrade_timeout: "{{ upgrade_timeout | default(600) }}"
37     image_store: "{{ image_store | default('http://fileserv.networkingdev.local/images') }}"
38     upgrade_report: []
39
40   tasks:
41
42     # — PLATFORM DETECTION —————
43
44     - name: Detect network platform
45       ansible.builtin.set_fact:
46         platform: "{{ ansible_network_os | default('unknown') }}"
47
48     - name: Abort if platform not supported
49       ansible.builtin.fail:
50         msg: "Platform '{{ platform }}' not supported by this playbook. Supported: eos, ios, junos."
51       when: platform not in ['eos', 'ios', 'junos']
52
53     # — PRE-UPGRADE VERSION CHECK —————
54
55     - name: "[EOS] Get current version"
56       arista.eos.eos_command:
57         commands: "show version | include Software image version"
58       register: eos_version_raw
59       when: platform == 'eos'
60
61     - name: "[IOS] Get current version"
62       cisco.ios.ios_command:
63         commands: "show version | include Cisco IOS XE Software"

```

```

64     register: ios_version_raw
65     when: platform == 'ios'
66
67 - name: "[Junos] Get current version"
68   junipernetworks.junos.junos_command:
69     commands: "show version | match Junos:"
70   register: junos_version_raw
71   when: platform == 'junos'
72
73 - name: Parse current version - EOS
74   ansible.builtin.set_fact:
75     current_version: "{{ eos_version_raw.stdout[0] |
76       regex_search('[0-9]+\.[0-9]+\.[0-9]+[A-Z]') }}"
77   when: platform == 'eos'
78
79 - name: Parse current version - IOS XE
80   ansible.builtin.set_fact:
81     current_version: "{{ ios_version_raw.stdout[0] |
82       regex_search('[0-9]+\.[0-9]+\.[0-9]+') }}"
83   when: platform == 'ios'
84
85 - name: Parse current version - Junos
86   ansible.builtin.set_fact:
87     current_version: "{{ junos_version_raw.stdout[0] |
88       regex_search('[0-9]+\.[0-9]+[A-Z][0-9]+(-D[0-9]+)?') }}"
89   when: platform == 'junos'
90
91 - name: Log pre-upgrade version
92   ansible.builtin.debug:
93     msg: "{{ inventory_hostname }} | Platform: {{ platform }} | Current: {{ current_version
94       }} | Target: {{ desired_os_version }}"
95
96 - name: Skip if already on target version
97   ansible.builtin.meta: end_host
98   when: current_version == desired_os_version
99
100 # — IMAGE TRANSFER —————
101
102 - name: "[EOS] Copy image to flash"
103   arista.eos.eos_command:
104     commands:
105       - "copy {{ image_store }}/{{ os_image_filename }} flash:"
106   vars:
107     ansible_command_timeout: "{{ upgrade_timeout }}"
108   when:
109     - platform == 'eos'
110     - os_image_filename is defined
111
112 - name: "[IOS] Copy image to flash via HTTP"
113   cisco.ios.ios_command:
114     commands:
115       - "copy {{ image_store }}/{{ os_image_filename }} flash:"
116   vars:
117     ansible_command_timeout: "{{ upgrade_timeout }}"
118   when:
119     - platform == 'ios'
120     - os_image_filename is defined
121
122 - name: "[Junos] Stage image package"
123   junipernetworks.junos.junos_package:
124     src: "{{ image_store }}/{{ os_image_filename }}"
125     no_copy: false
126     cleanfs: false
127     validate: true

```

```

125     vars:
126         ansible_command_timeout: "{{ upgrade_timeout }}"
127     when:
128         - platform == 'junos'
129         - os_image_filename is defined
130
131
132 # — BOOT CONFIGURATION —————
133
134 - name: "[EOS] Set boot image"
135   arista.eos.eos_command:
136     commands:
137       - "boot system flash:{{ os_image_filename }}"
138   when:
139       - platform == 'eos'
140       - os_image_filename is defined
141
142 - name: "[IOS] Set boot statement"
143   cisco.ios.ios_config:
144     lines:
145       - "boot system flash:{{ os_image_filename }}"
146     save_when: always
147   when:
148       - platform == 'ios'
149       - os_image_filename is defined
150
151
152 # — RELOAD / UPGRADE —————
153
154 - name: "[EOS] Reload device"
155   arista.eos.eos_command:
156     commands:
157       - command: "reload now"
158         prompt: '\[confirm\]'
159         answer: 'y'
160   when: platform == 'eos'
161   ignore_errors: true # connection drops on reload – expected
162
163 - name: "[IOS] Reload device"
164   cisco.ios.ios_command:
165     commands:
166       - command: "reload"
167         prompt: 'Proceed with reload\?'
168         answer: 'y'
169   when: platform == 'ios'
170   ignore_errors: true
171
172 - name: "[Junos] Request system software add and reboot"
173   junipernetworks.junos.junos_package:
174     src: "{{ image_store }}/{{ os_image_filename }}"
175     reboot: true
176     no_copy: true # already staged above
177   vars:
178     ansible_command_timeout: "{{ upgrade_timeout }}"
179   when:
180       - platform == 'junos'
181       - os_image_filename is defined
182   ignore_errors: true
183
184
185 # — WAIT FOR DEVICE TO COME BACK —————
186
187 - name: Wait for device to reload
188   ansible.builtin.wait_for:
189     host: "{{ ansible_host }}"

```

```

190     port: 22
191     delay: 60
192     timeout: "{{ upgrade_timeout }}"
193     state: started
194     delegate_to: localhost
195
196
197 # — POST-UPGRADE VERSION VERIFICATION —————
198
199 - name: "[EOS] Get version after upgrade"
200   arista.eos.eos_command:
201     commands: "show version | include Software image version"
202   register: eos_post_raw
203   when: platform == 'eos'
204
205 - name: "[IOS] Get version after upgrade"
206   cisco.ios.ios_command:
207     commands: "show version | include Cisco IOS XE Software"
208   register: ios_post_raw
209   when: platform == 'ios'
210
211 - name: "[Junos] Get version after upgrade"
212   junipernetworks.junos.junos_command:
213     commands: "show version | match Junos:"
214   register: junos_post_raw
215   when: platform == 'junos'
216
217 - name: Parse post-upgrade version – EOS
218   ansible.builtin.set_fact:
219     post_version: "{{ eos_post_raw.stdout[0] | regex_search('[0-9]+\.[0-9]+\.[0-9]+[A-Z]') }}"
220   when: platform == 'eos'
221
222 - name: Parse post-upgrade version – IOS XE
223   ansible.builtin.set_fact:
224     post_version: "{{ ios_post_raw.stdout[0] | regex_search('[0-9]+\.[0-9]+\.[0-9]+' ) }}"
225   when: platform == 'ios'
226
227 - name: Parse post-upgrade version – Junos
228   ansible.builtin.set_fact:
229     post_version: "{{ junos_post_raw.stdout[0] |
230     regex_search('[0-9]+\.[0-9]+[A-Z][0-9]+(-D[0-9]+)?') }}"
231   when: platform == 'junos'
232
233 - name: Evaluate upgrade result
234   ansible.builtin.set_fact:
235     upgrade_status: "{{ 'SUCCESS' if post_version == desired_os_version else 'FAILED' }}"
236
237 - name: Log post-upgrade result
238   ansible.builtin.debug:
239     msg: >
240       {{ inventory_hostname }} | {{ upgrade_status }}
241       | Before: {{ current_version }}
242       | After: {{ post_version }}
243       | Expected: {{ desired_os_version }}
244
245 - name: Fail task if version does not match expected
246   ansible.builtin.fail:
247     msg: >
248       {{ inventory_hostname }}: upgrade did not reach target version.
249       Got {{ post_version }}, expected {{ desired_os_version }}.
250   when: upgrade_status == 'FAILED'
251
252 # =====

```

```

253 # REPORT GENERATION
254 # Runs after all hosts finish (success or failure). Produces the summary
255 # that Flow Weaver picks up and emails to info@networkingdev.com
256 # =====
257
258 - name: Generate upgrade report
259   hosts: localhost
260   gather_facts: false
261
262   tasks:
263
264     - name: Build per-device result summary
265       ansible.builtin.set_fact:
266         device_results: >-
267           {{
268             groups['upgrade_targets'] | default([]) | map('extract', hostvars) |
269             list | map(attribute='upgrade_status', default='NOT_RUN') | list
270           }}
271
272     - name: Write CSV report
273       ansible.builtin.copy:
274         dest: "/tmp/os_upgrade_report_{{ lookup('pipe', 'date +%Y%m%d_%H%M%S') }}.csv"
275         content: |
276           device_name,platform,version_before,version_after,desired_version,status
277           {% for host in groups['upgrade_targets'] | default([]) %}
278           {% set h = hostvars[host] %}
279           {{ host }},{{ h.platform | default('unknown') }},{{ h.current_version | default('N/A')
280           }},{{ h.post_version | default('N/A') }},{{ h.desired_os_version | default('N/A')
281           }},{{ h.upgrade_status | default('NOT_RUN') }}
282           {% endfor %}
283         mode: '0644'
284
285     - name: Print final summary
286       ansible.builtin.debug:
287         msg: >
288           =====
289           OS Upgrade Run Complete
290           =====
291           {% for host in groups['upgrade_targets'] | default([]) %}
292           {% set h = hostvars[host] %}
293           {{ '%-35s' | format(host) }} {{ h.upgrade_status | default('NOT_RUN') }} ({{
294           h.current_version | default('?') }} → {{ h.post_version | default('?') }})

```