AutoCon 6 Workshop: Building Network Automations with BackBox Syllabus

Nov 15, 2026 · @Sandra Topalovic

WORKSHOP DETAILS

Date: **Monday, November 16, 2026**

Time: **2:00 PM – 6:00 PM (MST)**

Room: **Indigo**

Venue: **The Westin La Paloma Resort & Spa, Tucson, AZ**

Workshop overview

In this four-hour hands-on workshop (2:00–6:00 PM), each attendee builds a working network automation on a live Cisco Nexus device. The automation backs up configs, checks them against a compliance policy, remediates drift, verifies its own changes, and rolls back on failure.

Who it's for: network engineers comfortable with the CLI, config files, and common protocols. No BackBox experience is needed.

Format: guided labs in a dedicated lab environment. Each student has their own Nexus device and site, so nothing can be harmed. Proctors are on hand; raise your hand early and often.

Learning objectives

By the end of the workshop, you will be able to:

- Build a Kilter automation from scratch and with AI-assisted task creation, and spot where the AI gets it wrong
- Use variables and basic conditional logic to make tasks reusable
- Schedule configuration backups and define compliance policies with IntelliChecks
- Detect configuration drift and trigger automated remediation
- Run automations across multiple devices, on a schedule or in response to an event
- Add pre- and post-change verification and rollback so production changes can be trusted
- Diagnose common automation failures and plan for scale

Schedule At A Glance

Time	Module	Focus
2:00 PM	Welcome and Orientation	Goals, lab login, platform architecture
2:20 PM	Building Your First Automation	Manual and AI-assisted tasks, variables, conditions
3:00 PM	Config Backup and Compliance	Scheduled backups, IntelliChecks, drift remediation
3:45 PM	Break (30 min)	
4:15 PM	Advanced Workflows	Multi-device runs, conditional logic, triggers
4:55 PM	Change Verification and Rollback	Pre/post checks, Nexus upgrade with rollback
5:30 PM	Troubleshooting and Scaling	Failure modes, fleet-scale best practices
5:50 PM	Wrap-Up and Resources	Recap, documentation, next steps

Lab Environment and Logistics

- **Lab URL:** (removed, ask in Slack) (open in your browser)
- **Credentials:** sent by private Slack message this morning to everyone signed into the AutoCon Slack workspace. If you didn't get yours, ask a proctor.
- **Your resources:** one Cisco Nexus device and one dedicated site number, shown in the top right of the UI
- **Prerequisites:** a laptop with a browser; CLI (included in the platform), config-file (included), and common-protocol familiarity
- **Help:** proctors circulate throughout

Module details

1. Welcome and Orientation (2:00 PM)

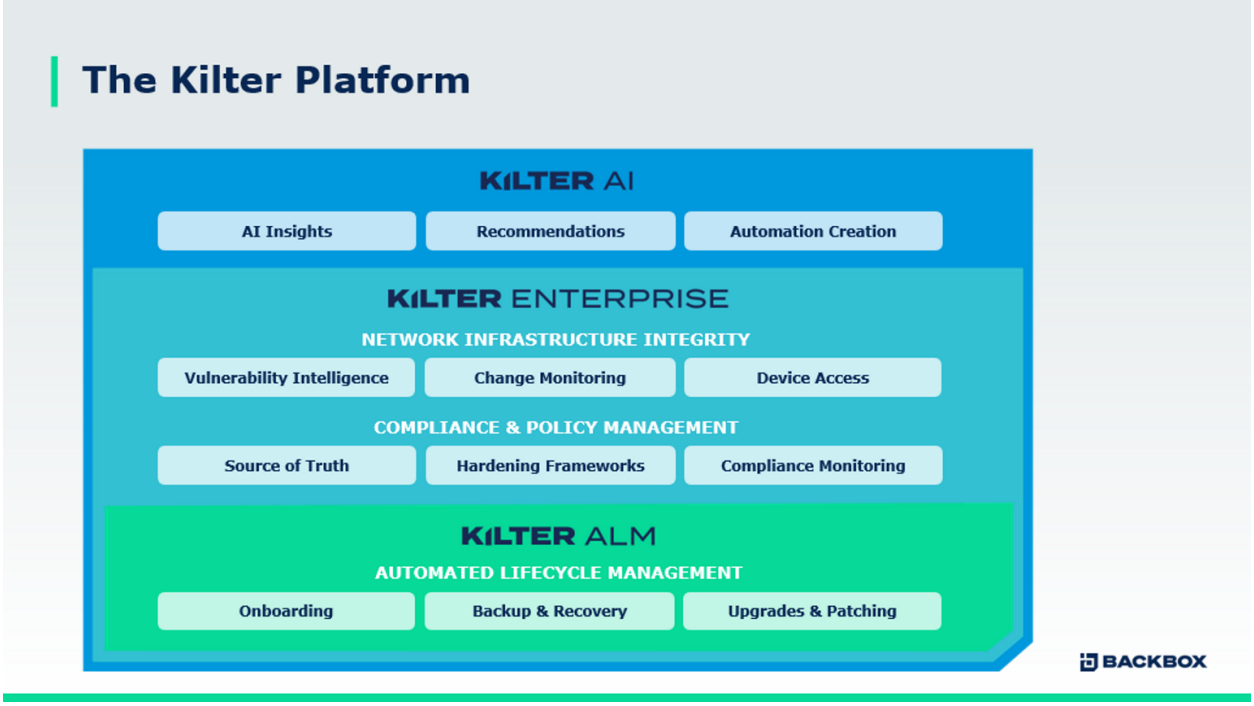
- Introduction
- Workshop goals and the path from first automation to production-ready safety nets
- Lab login
- Tour the BackBox architecture and automation engine

BackBox Platform Overview

Kilter by BackBox is built in three layers:

1. **Kilter ALM** (automated lifecycle management: onboarding, backup and recovery, upgrades and patching)

2. **Kilter Enterprise** (network infrastructure integrity, compliance and policy management)
3. **Kilter AI** (insights, recommendations, and automation creation)



BackBox Architecture and Communication Flow

https://support.backbox.com/s/article/Port-Requirements-and-Communication-Flow

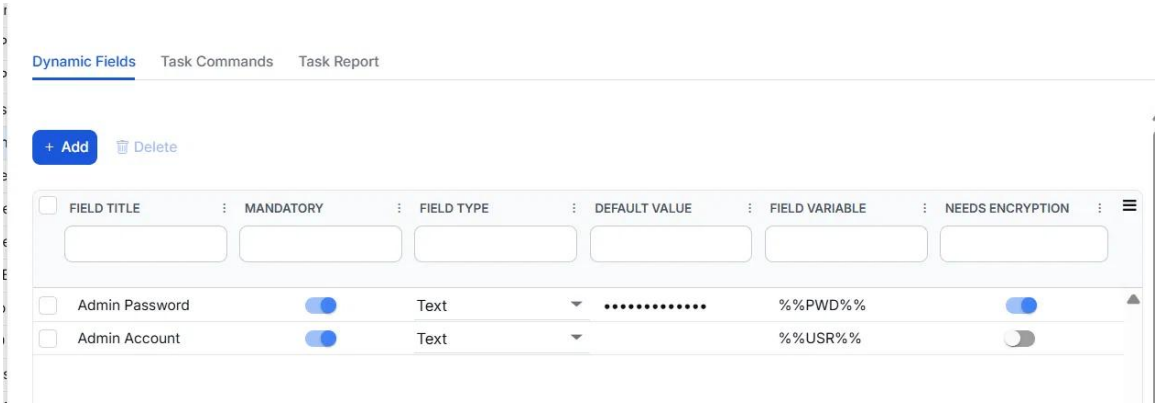
2. Building Your First Automation (2:20 PM)

Lab 1: build a task manually (about 20 minutes)

Goal: a Cisco Nexus task that adds an admin user, with the username and password supplied as variables.

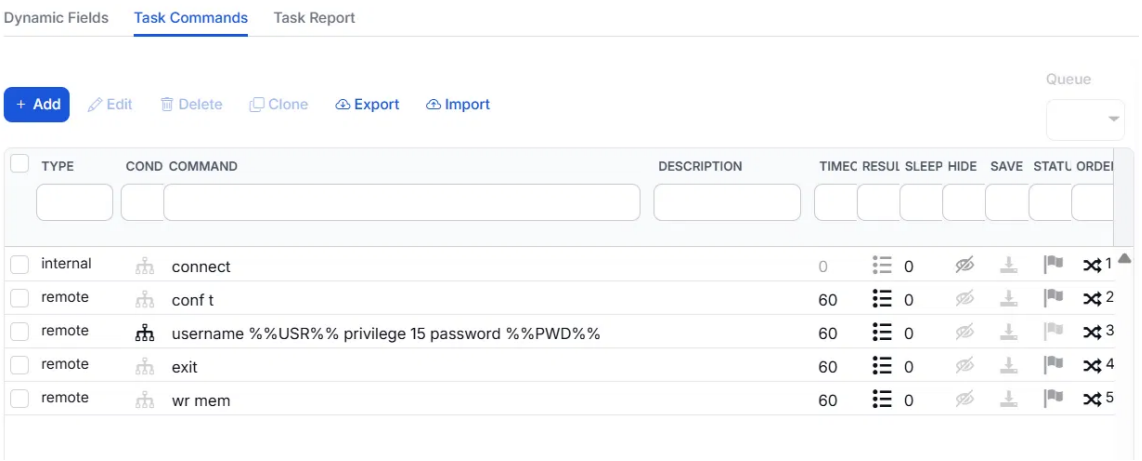
1. Go to **Automations** → **Tasks** and click **Add**. Name the task (for example, *Nexus - Add Admin User*).
2. Open the **Dynamic Fields** tab and click **Add** to create two fields:
    - **Admin Account:** Mandatory on, Field Type Text, Field Variable `%%USR%%`, Needs Encryption off

- **Admin Password:** Mandatory on, Field Type Text, Field Variable `%%PWD%%`, Needs Encryption on (its default value is masked)



Step 2: the Dynamic Fields tab with both fields defined.

3. Open the **Task Commands** tab and click **Add** once per command, in this order:
   - Order 1: `connect`, Type internal, Timeout 0. Opens the session to the device
   - Order 2: `conf t`, Type remote, Timeout 60. Enters global configuration mode
   - Order 3: `username %%USR%% privilege 15 password %%PWD%%`, Type remote, Timeout 60. Creates a local user with privilege level 15 (full access)
   - Order 4: `exit`, Type remote, Timeout 60. Leaves config mode
   - Order 5: `wr mem`, Type remote, Timeout 60. Saves the running config to startup so it survives a reboot
4. Check the COND column: only the `username` command has its condition icon highlighted. *The instructor will demo the condition during the walkthrough (details to be confirmed).*



Steps 3 and 4: the Task Commands tab. Note the highlighted condition icon on the `username` command.

5. Save the task

At run time, BackBox replaces `%%USR%%` and `%%PWD%%` with the values entered in the Dynamic Fields, so nothing is hard-coded in the command.

Self-check: Task Commands lists five commands numbered 1 to 5 in the order above, and Dynamic Fields lists two fields.

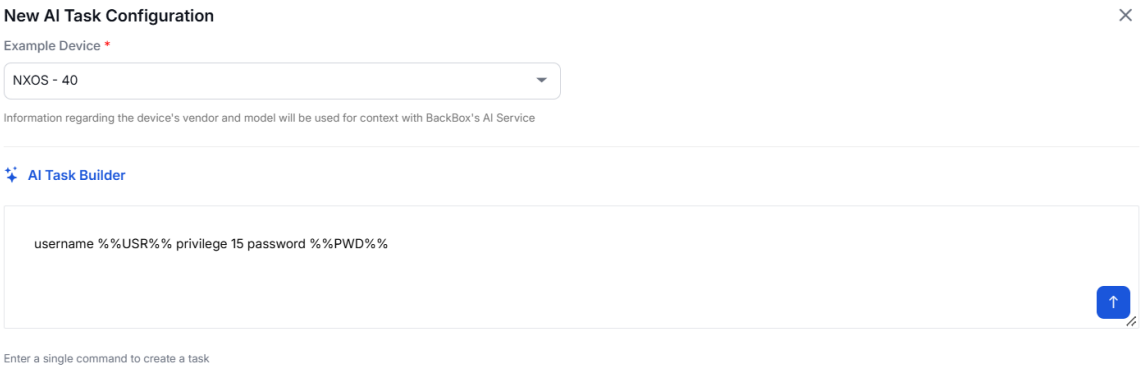
If something looks wrong:

- A variable must match exactly in the command and in Dynamic Fields, including both percent signs on each side (`%%USR%%`, not `%USR%`)
- Save is greyed out or a field is outlined red: fill in the required field
- Commands in the wrong order: check the Order column and re-add or reorder until it reads 1 to 5

Lab 2: create the same task with AI (about 15 minutes)

Goal: see how Kilter AI builds a task from a single command, and judge whether to trust it.

1. Go to **Automations** → **Tasks** and click **New AI Task**.
2. For **Example Device**, select your Nexus (named for your site ID, for example *NXOS - 40*). Its vendor and model give the AI context.
3. In the AI Task Builder box, enter one command: `username %%USR%% privilege 15 password %%PWD%%`
4. Click the blue arrow button to generate the task



Steps 2 to 4: the New AI Task dialog. Your device name will show your own site ID.

5. Open the generated task and review its Task Commands and Dynamic Fields tabs

Compare it to your Lab 1 task. AI can make mistakes. Check each of these and note any difference:

Check	What to look for
Commands and order	Same five commands, in the same order? Anything missing or added?
Type and timeout	`connect` is internal; the others are remote with a 60-second timeout?
Condition	Is the condition icon set on the `username` command only?
Dynamic fields	Are `%%USR%%` and `%%PWD%%` both defined, Mandatory, as Text?
Password handling	Is Needs Encryption on for the password field?

Self-check: you can name at least one way the AI version matches your manual task and, if any, one way it differs.

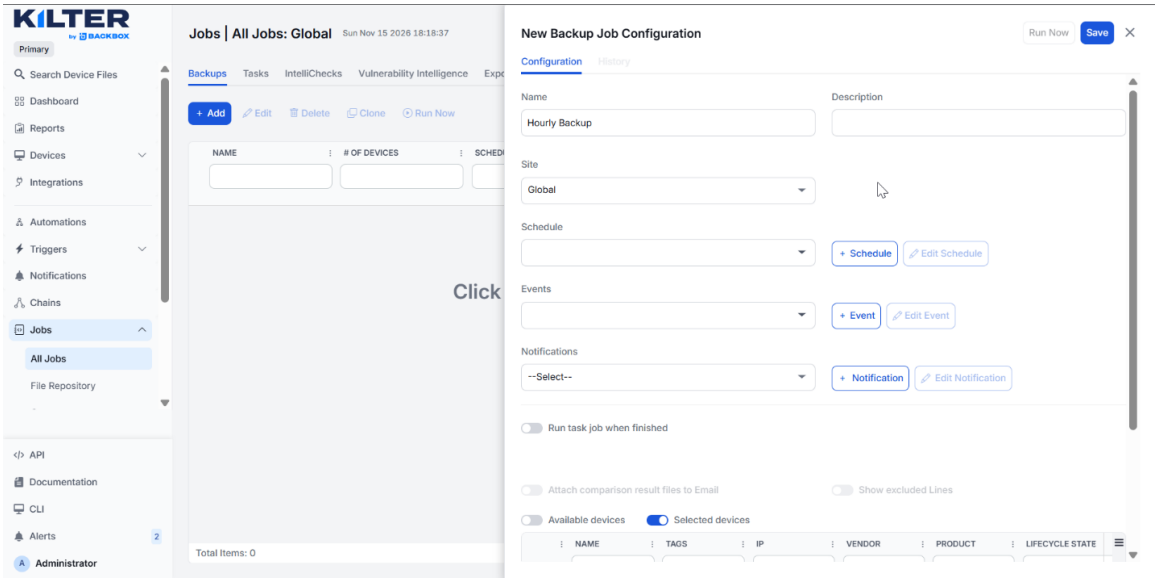
If something looks wrong: Make sure an Example Device is selected (it is required). Enter only one command in the builder, since it creates a task from a single command.

3. Config Backup and Compliance Automation (3:00 PM)

Lab 3: schedule recurring backups (about 15 minutes)

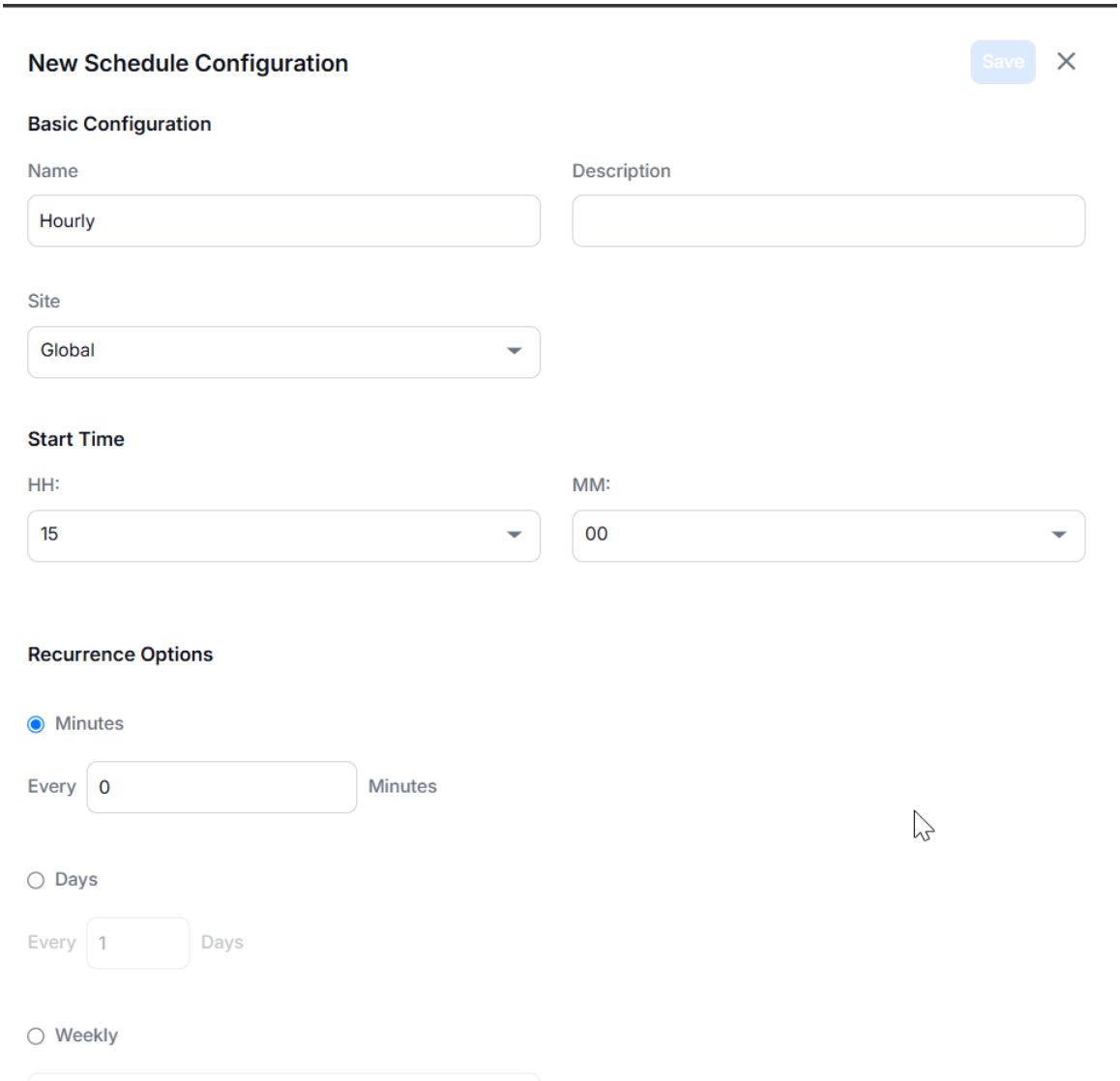
Goal: a job that backs up your Nexus config on a schedule. Daily is typical in production; we use hourly so you can see runs before the end of class.

1. Go to **Jobs** → **All Jobs**, open the **Backups** tab, and click **Add**
2. Set **Name** to `Hourly Backups`
3. Set **Site** to your site ID (not Global)



Steps 1 to 3: the backup job panel. Set Site to your site ID instead of Global, and name it Hourly Backups.

4. Next to the **Schedule** dropdown, click **+ Schedule**. In New Schedule Configuration:
    - Name: `Hourly`
    - Site: your site ID
    - Start Time: choose an hour (HH) and minute (MM) to begin
    - Recurrence Options: select **Minutes** and set Every to `60` (60 minutes is hourly)
    - Click **Save**



Step 4: the schedule panel. Choose Minutes under Recurrence Options and enter 60.

5. Back on the backup job, select the `Hourly` schedule in the Schedule dropdown
6. Turn on the **Available devices** toggle and check your Nexus in the device list
7. Click **Save**

We return to this job around 5:00 PM to review the runs and any errors. To check sooner, reopen the job and look at its History tab.

Self-check: *Hourly Backups* appears in the Backups list with 1 device.

If something looks wrong:

- No devices in the list: Reopen the job and confirm Site is your site ID

- Schedule dropdown is empty: save the new schedule first, then reselect it in the job
- Save is greyed out: check that Name and Site are filled in

Lab 4: define a compliance policy, detect drift, and remediate (about 35 minutes)

Goal: check that your Nexus has a message-of-the-day (MOTD) banner, watch the check fail, then have BackBox fix it automatically.

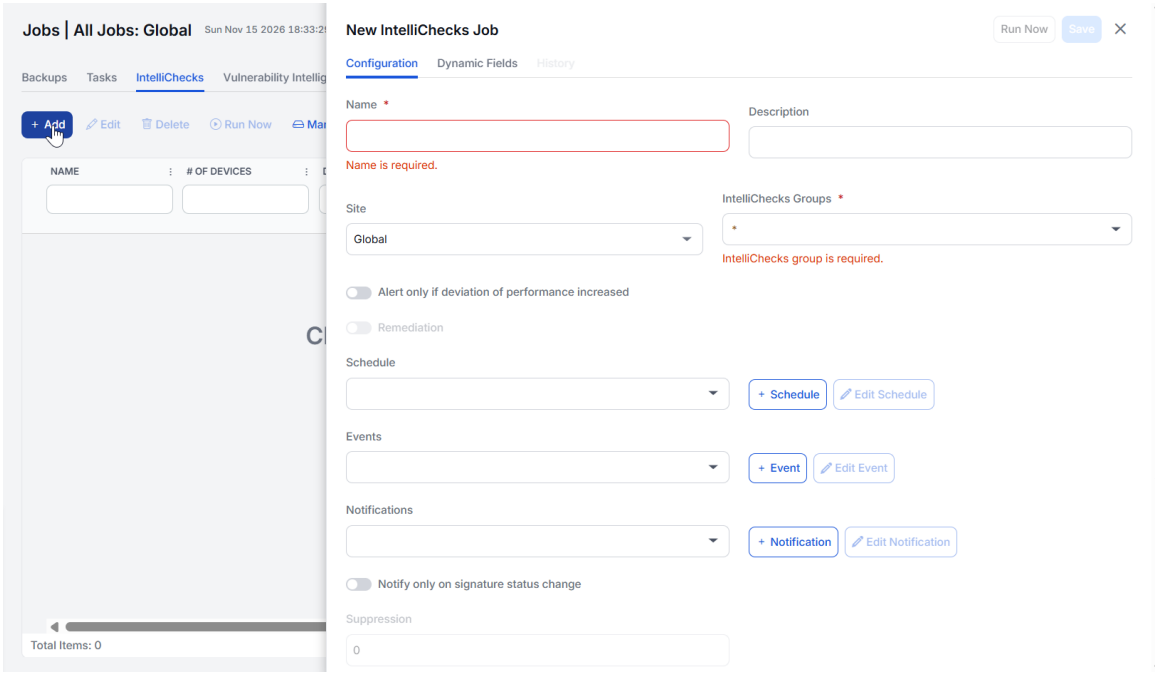
Part A: Clone the signature and build a group

1. Go to **Automations** → **IntelliChecks**
2. Check the box next to *1.7.1 Configure an MOTD (Message of the day) Banner* and click **Clone**
3. Name the clone `Cisco Nexus configure MOTD`, set Site to your site ID, and save
4. Create an IntelliChecks **group**. Only groups can run in jobs, even for a single signature. Name it `Cisco Nexus MOTD`, add your cloned signature to it, and save

Self-check A: your group *Cisco Nexus MOTD* exists and contains one signature, *Cisco Nexus configure MOTD*.

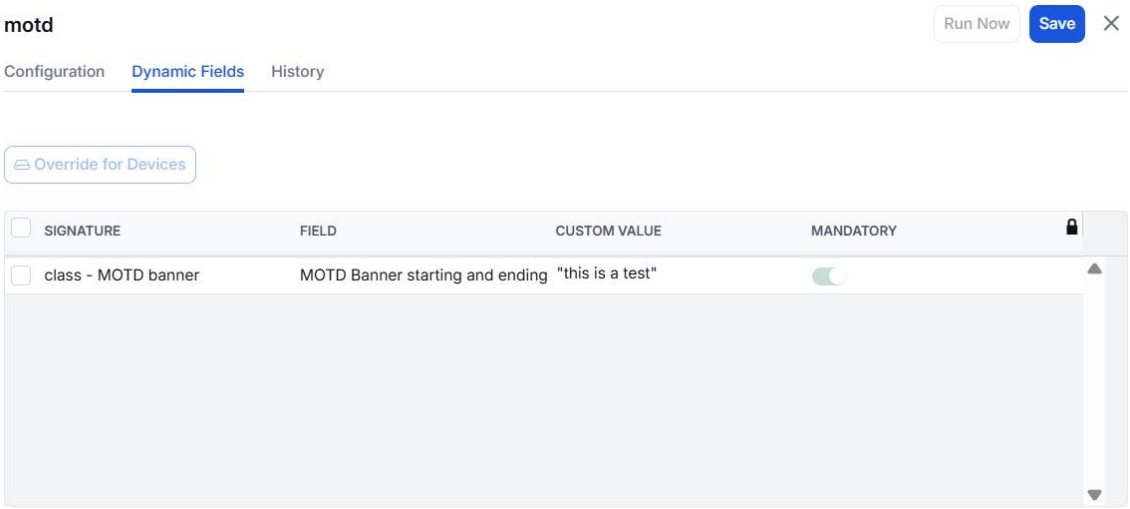
Part B: Create the compliance job

1. Go to **Jobs** → **All Jobs**, open the **IntelliChecks** tab, and click **Add**
2. Fill in the required fields (marked with a red asterisk): **Name** (for example `MOTD Check`) and **IntelliChecks Groups** (select `Cisco Nexus MOTD`). Set Site to your site ID



Steps 1 and 2: the New IntelliChecks Job panel. The red messages show the required fields.

3. Leave **Remediation** off for now
4. Open the **Dynamic Fields** tab. Click the **Mandatory** column header to sort so the MOTD signature is at the top
5. In the **Custom Value** column, enter a message of the day **in double quotes**, for example `"I love this workshop!"`. Without the quotes it will not parse correctly



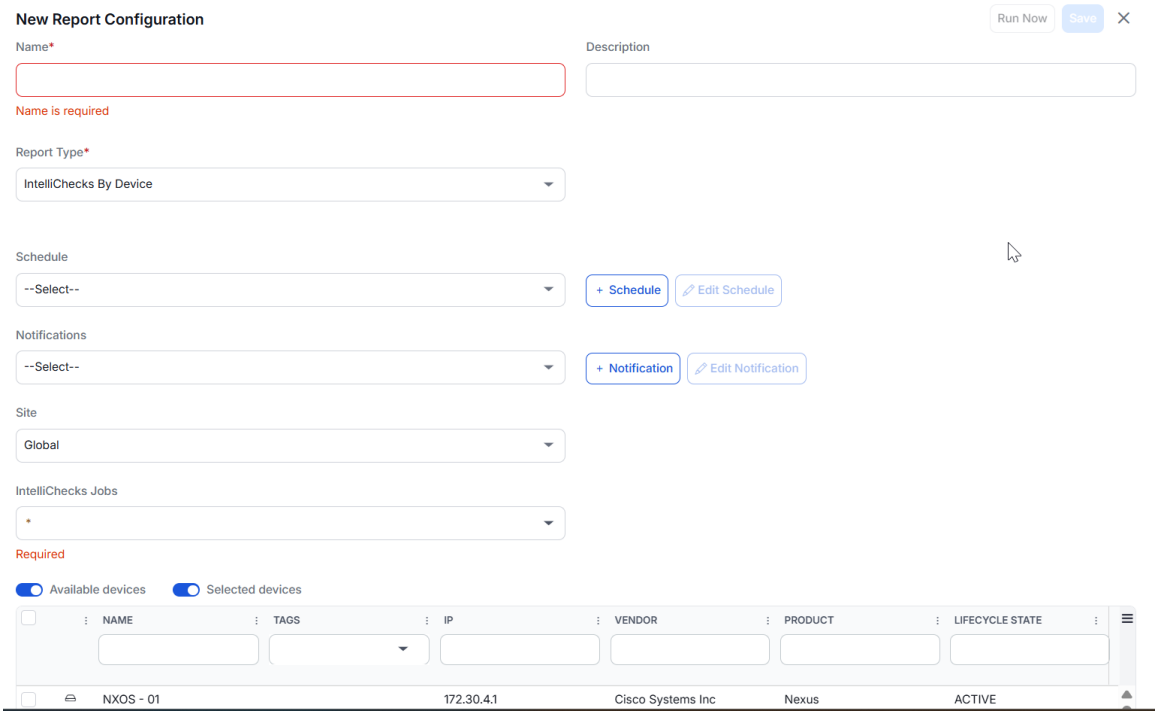
Steps 4 and 5: the Dynamic Fields tab with the quoted MOTD message. Your signature and job names will match what you created previously.

6. Click **Save**

Self-check B: the job appears in the IntelliChecks tab, and its Dynamic Fields tab shows your quoted message.

Part C: run a report and observe drift

1. Go to **Reports** and click **Add**
2. Enter a Name (required), set Report Type to `IntelliChecks By Device`, and set Site to your site ID
3. In **IntelliChecks Jobs**, select the job from Part B (required)
4. Turn on **Available devices** and check your Nexus

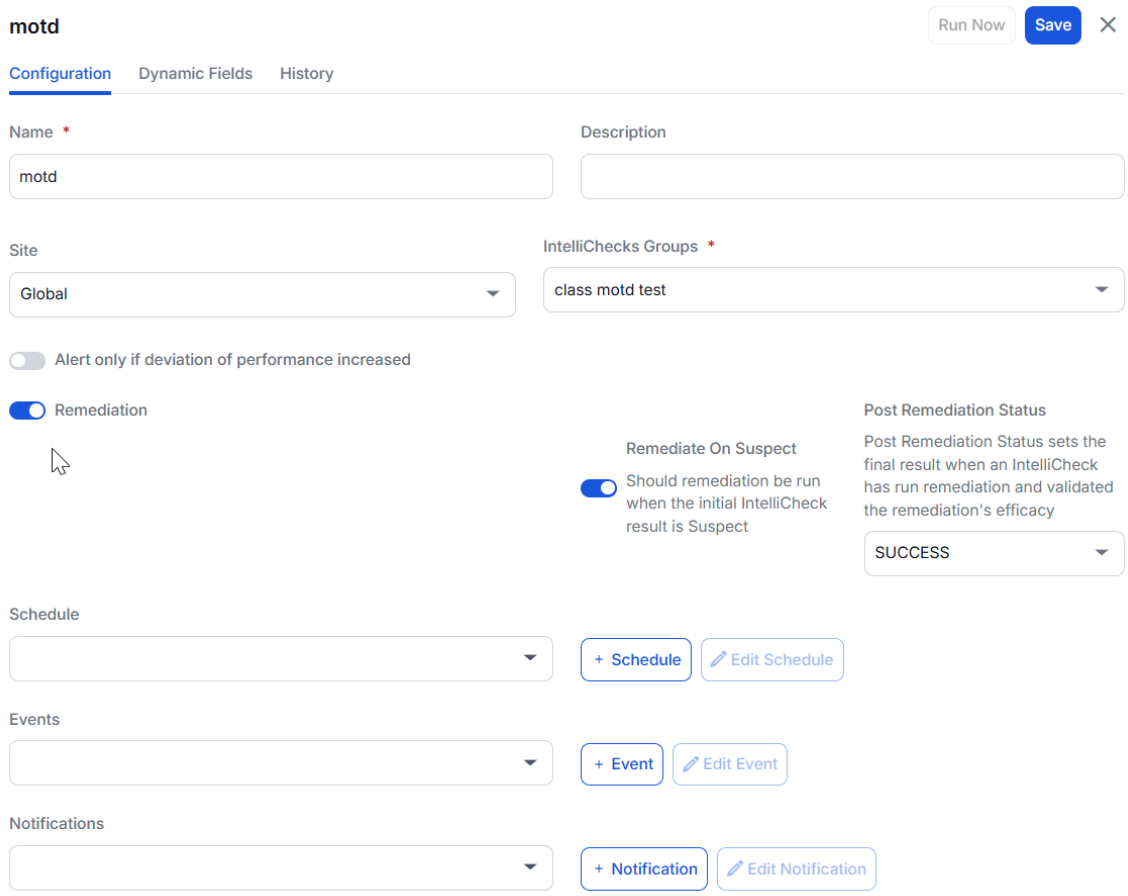


Steps 2 to 4: the report panel with Report Type set to IntelliChecks By Device and the device list at the bottom.

5. Click **Save**, then run the report
6. Open the results. The MOTD check **fails**, because no MOTD was set on the device. That is configuration drift

Part D: turn on automated remediation

1. Go back to your IntelliChecks job (Jobs → All Jobs → IntelliChecks) and open its **Configuration** tab
2. Turn **Remediation** on. Leave *Remediate On Suspect* on
3. Set **Post Remediation Status** to `SUCCESS`. This is the final result recorded when remediation runs and validates that the fix worked



Steps 2 and 3: Remediation on, Post Remediation Status set to SUCCESS.

4. Click **Save**, then **Run Now**
5. Run your report again. The MOTD check should now pass

Self-check D: the second report run shows the MOTD check passing for your device.

If something looks wrong:

- Name is required or IntelliChecks group is required (red text): fill in those fields. Save and Run Now stay greyed out until you do
- The group is not in the dropdown: only groups appear there, not individual signatures. Revisit Part A step 4
- Check still fails after remediation: confirm Remediation is on, you clicked Save before Run Now, and the custom value is in double quotes

4. Break (3:45 PM)

30 minutes

5. Advanced Workflows (4:15 PM)

Extend your automation to a multi-device environment, and trigger it on a schedule or an event.

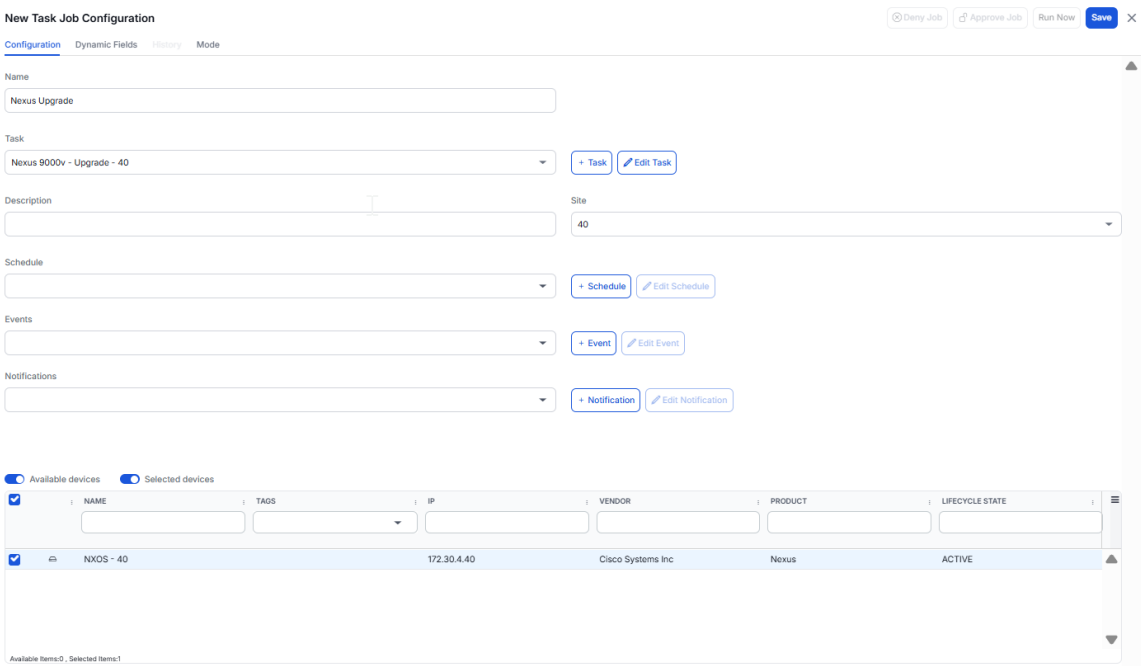
Discussion only due to limited devices in lab

6. Change Verification and Rollback Safety Nets (4:55 PM)

Lab 5: upgrade your Nexus from 9.3(9) to 9.3(16) (about 25 minutes)

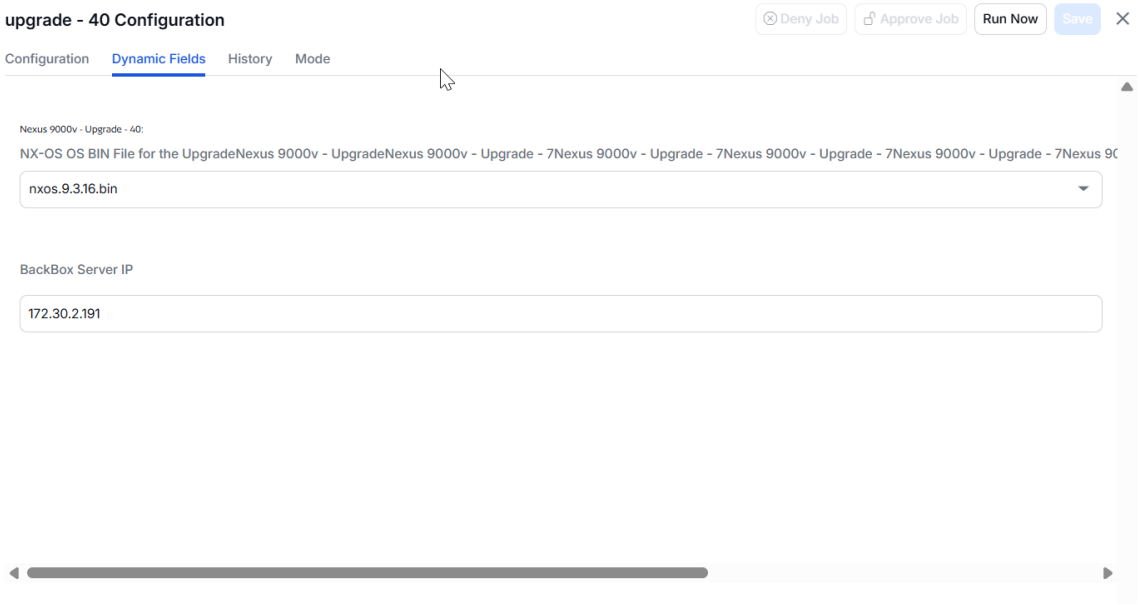
Goal: run a real change as a job, then verify the result. Upgrades are where pre-change checks, post-change checks, and rollback matter most. *Steps for adding pre/post checks and rollback logic to this automation are to be added; the source draft only covers building and running the upgrade job.*

1. Go to **Jobs** → **All Jobs**, open the **Tasks** tab, and click **Add**
2. Set **Name** to `Nexus Upgrade`
3. Set **Task** to `Nexus 9000v - Upgrade - <your site ID>` (for example, *Nexus 9000v - Upgrade - 40*)
4. Set **Site** to your site ID
5. In the device list, turn on **Available devices** and check your Nexus. The footer should read *Selected Items: 1*



Steps 2 to 5: the job configuration with your site, task, and device selected. Yours will show your own site ID, not 40.

6. Open the **Dynamic Fields** tab and fill in:
    - **NX-OS BIN File for the Upgrade:** select `nxos.9.3.16.bin`
    - **BackBox Server IP:** `172.30.2.191`



Step 6: the Dynamic Fields tab with the upgrade file and server IP.

7. Click **Save**, then **Run Now**
8. Open the job's **History** tab to follow the run.

Self-check: the job runs against your device and the History tab shows its result. After the upgrade completes, your Nexus should report version 9.3(16).

If something looks wrong:

- Run Now is greyed out: save the job first
- The BIN file dropdown is empty or the wrong file is selected: confirm you chose the task for your own site ID in step 3
- Selected Items shows 0: turn on Available devices and check your device
- Typo in the server IP: it must be exactly `172.30.2.191`

7. Troubleshooting and Scaling Best Practices (5:30 PM)

Review common automation failure modes and how to diagnose them, plus best practices for scaling from a few lab devices to thousands in production.

Where to look when something fails (use this all session, not just now):

- **Required-field errors:** red text under a field means a required value is missing; Save stays greyed out until it is fixed.
- **Job History tab:** open any job and check its History tab for each run and its status.
- **Task Report tab:** open a task and check its Task Report tab for per-command results.
- **Alerts:** the Alerts item in the left menu shows a count of open alerts.
- **Values you typed:** check variable names (`%%USR%%`, `%%PWD%%`), double quotes around the MOTD value, your site ID, and the server IP.

Revisit your hourly backups (about 5:00 PM). Go to Jobs → All Jobs → Backups, open *Hourly Backups*, and review its History tab. Note which runs succeeded and read the message on any that failed.

8. Wrap-Up and Resources (5:50 PM)

Recap what you built and get pointed to documentation for continued learning.

Wrap-up and resources

You leave with a working automation that backs up configs, checks compliance, remediates drift, verifies its own changes, and rolls back on failure. Proctors will point you to BackBox documentation and learning resources for continued practice. *Links to be added.*