Reproducible network labs without dedicated learner hardware

An open-source lifecycle across cloud and on-premises capacity

Jeleel Muibi · HybridOps.Tech
Technical paper · Version 1.0
First published February 2026 · revised 8 August 2026

Abstract

Many learners can follow a networking exercise once the lab exists but cannot afford hardware powerful enough to run a realistic multi-system environment. Trial or education credit can supply cloud compute, and a department may already operate virtualisation capacity through Proxmox, but converting either resource into a safe lab requires infrastructure work before networking practice can begin.

HybridOps governs the infrastructure lifecycle around that gap. Before a lab is exposed, the runtime resolves target policy, dependencies, capacity, content, private connectivity and required verification. Declared open or operator-licensed images, managed node egress, application health and browser or desktop access remain part of the same operation. The emulator supplies the lab experience; it does not govern the external lifecycle.

The Google Cloud path derives estimated fixed hourly and session costs from the deployed VM and disk. Portable lab definitions and writable QEMU node-disk state can persist between sessions while guarded teardown removes the managed infrastructure. Once every managed stage reaches a terminal teardown state, the runtime reports zero estimated fixed ongoing cost for that environment. The Proxmox path applies the same lab lifecycle to existing institutional or trainer capacity.

The practical test is whether a learner can reproduce, inspect, change and resume the demonstrated lab outside the original session.

1. The missing handoff, not another emulator

Virtual laboratories, remote access and persistent simulations are established capabilities. Packet Tracer supports guided network simulation and authored activities. Cisco Modeling Labs provides authorised reference platforms, persistent simulations and an API. NETLAB+ VE manages users, schedules and isolated lab access for institutions. EVE-NG and GNS3 provide flexible network emulation.

HybridOps does not replace those systems or claim their functions as new. It operates at a different boundary: the infrastructure and workload lifecycle around the lab.

Platform	Primary role	Boundary in this paper
Packet Tracer	Guided network simulation and activity authoring	Learning within a supported simulation model
Cisco Modeling Labs	Network simulation with licensed images, persistent labs and APIs	Simulation lifecycle within a CML deployment
NETLAB+ VE	Institution-managed users, schedules, pods and remote access	Course delivery around a managed lab estate
EVE-NG and GNS3	Network emulation, topology design and consoles	Replaceable lab workloads
HybridOps	Declared control of target policy, dependencies, connectivity, readiness, state, cost posture, continuity and teardown	One external lifecycle across providers, sites and workloads

The remaining difficulty is the handoff to a learner who does not receive a managed institutional lab. The problem becomes more visible as the exercise grows:

- virtual appliances must be prepared in the expected form;
- nested virtualisation and emulator services must be available;
- nodes need addressing, DNS and controlled internet egress;
- browser and native consoles must reach a private host;
- cloud resources must be removed when the session ends; and
- the learner's work must survive that teardown when continuity is required.

These are not the networking lesson, yet any one of them can prevent it from starting. HybridOps tests whether a versioned environment contract can keep those operations consistent as the provider or emulator changes. The implementation exercises Google Cloud, Proxmox, EVE-NG and GNS3.

2. A lab is more than a topology file

The required capabilities, dependencies, inputs, outputs and readiness conditions are declared before execution. Environment policy selects cloud or on-premises capacity without changing the intended learning outcome.

Stage	Outcome
Demonstrate	Establish the concept and working topology
Publish	Record the lab contract, images and target policy
Reproduce	Create the private environment and verify its starting state
Practise	Configure, observe and diagnose the lab
Continue	Archive, remove and later restore the lab

The same resolved operation governs readiness, deployment, access, rebuild and teardown. This matters because reproducibility includes the end of a session, not only its beginning.

3. Implemented and verified paths

EVE-NG and GNS3 are the first worked delivery paths within the wider operating model. Each follows the same declared lifecycle while retaining its own image, access and health requirements.

The EVE-NG path delivers images and a lab from local, repository or remote sources. Its health gate tests the service, database, API and hardware virtualisation path; browser and native-console traffic crosses a private workstation tunnel.

The GNS3 path exposes its loopback API through a private desktop tunnel, checksum-verifies QEMU images before registration and runs a disposable project to prove node execution.

The following boundaries have been exercised in the current implementation:

Tested boundary	Observed behaviour
Delivery	Private Google Cloud and Proxmox targets, operator-controlled images and topology delivery
Verification	Private access plus service, API, virtualisation, guest-network and node-execution checks
Continuity	Verified persistence of portable definitions and writable QEMU node-disk state across teardown and redeployment

4. Scale through separation and composition

Scale is not only a larger emulator host. The current implementation can place separate environments behind site, team or security boundaries and connect them through declared network and state relationships. Its wider paths include multi-VM cloud and on-premises targets, governed addressing and inventory, and routed cloud-to-site extension. This is a composable route to larger scenarios, with cohort concurrency and maximum topology size remaining the next scale evaluation.

Content is independent of compute. Images can come from declared remote sources, local archives or shared libraries, while lab definitions can come from local or versioned remote sources. The GNS3 path verifies declared QEMU images and retains project definitions and writable node disks across teardown and redeployment.

The emulators remain replaceable execution environments. The same boundary supports enterprise scenarios in which simulated sites connect to physical gateways, telemetry services or other external systems.

5. One environment, several levels of practice

A reproducible lab can support a sequence of lessons rather than one isolated exercise. A learner may begin with addressing and reachability, then introduce VLANs, inter-VLAN routing, dynamic routing, first-hop redundancy, firewall policy, NAT, IPsec and packet capture. Linux systems can provide services or authorised security-testing endpoints. Later work can apply Python, APIs or configuration automation to the same network.

Physical equipment remains valuable where device handling and hardware behaviour are learning outcomes. A reproducible virtual lab extends the lesson beyond the live session: learners can repeat configuration, compare behaviour and prepare before returning to the equipment.

6. The infrastructure is disposable; the lab is not

Cloud access creates an opportunity for independent practice and a duty to control resources that may continue to incur cost. HybridOps verifies billing readiness before deployment and reports the active environment, resource age, estimated fixed hourly cost and accumulated access-session estimate. Closing the private tunnel does not imply that the VM and disk have stopped incurring cost, so the access flow states that risk before offering guarded teardown.

HybridOps supports two persistence layers. Definition persistence retains the portable lab or project structure. Full-state persistence also retains the writable QEMU node disks that hold saved configurations, installed software and disk-level changes. The operator can retain the portable definition alone or carry the working disks forward as well. The learner's work is therefore separated from the lifetime of the host.

For QEMU-based EVE-NG nodes, the saved state includes the writable QCOW2 overlays that hold device disk changes, not only the topology definition. HybridOps stops active nodes, excludes the operator-supplied base-image library, validates every selected overlay and produces a checksummed companion archive. After redeployment, it requires the matching base images, protects existing runtime disks and validates the restored overlays. A learner can tear down the billable host and later resume from the saved device state without packaging or redistributing vendor images.

For GNS3, the archive carries the project definition, metadata and writable node disks after the controller has been stopped for a consistent snapshot. The archive is checksum-verified, restored through a guarded path and followed by an application check. Base images remain separately managed by default.

After every managed stage reaches a terminal teardown state, HybridOps reports an estimated ongoing cost of zero for the removed environment. The host is temporary; the topology and writable QEMU node-disk state are durable.

7. Why the open implementation matters

The runtime, environment declarations and operating guidance are public. An instructor can inspect how a lab is assembled and which checks define readiness. A learner can trace the relationship between declared intent, runtime state, configuration and observed outcome.

That surrounding system can itself become teaching material. Network automation students can examine dependency resolution and state handoff. Cybersecurity students can inspect private access boundaries, secret handling and the difference between an available service and a verified workload.

A corrected topology, additional health check or improved image declaration can be reviewed and versioned rather than circulated as an undocumented local setup.

8. What a reviewer can test

The accompanying demonstration shows one representative run. A practical review can establish whether:

1. required stages pass their readiness checks;
2. the private lab reaches a verified starting state;
3. declared images and topology content are available;
4. guest nodes receive addressing and reach the intended network;
5. one meaningful routing or security result is observed; and
6. the lab is archived before the infrastructure is removed.

The strongest test is not whether the command is short. It is whether another person can reach the same starting state and continue the exercise.

9. Request for technical review

1. Where in an existing network or cybersecurity training series would this continuity model add the most practical value?
2. Which part of a physical or streamed demonstration should a learner be able to reproduce independently?
3. Which lesson or enterprise integration would provide the fairest practical test of the model?

References

1. HybridOps Core.
2. HybridOps blueprint catalogue.
3. HybridOps platform runbooks.
4. Network and cybersecurity teaching-lab demonstration.
5. Cisco Packet Tracer FAQ.
6. Cisco Modeling Labs.
7. NETLAB+ software and virtualisation requirements.
8. EVE-NG documentation.
9. GNS3 documentation.