HybridOps: a contract runtime for reproducible infrastructure operations

A governed network-emulation case across cloud and institutional infrastructure

Jeleel Muibi
Version 1.0 | Institutional and technical review edition
First published June 2026 · revised 8 August 2026

Abstract

Network-emulation environments depend on more than the emulator itself. Infrastructure readiness, dependency order, private access, health, cost, state preservation and teardown remain operational concerns even when suitable compute capacity already exists.

HybridOps is an open-source contract runtime for that lifecycle. It separates declared intent and environment policy from provider and workload implementation, resolves ordered dependencies, evaluates readiness, verifies resulting capability and records lifecycle state and operational evidence.

This paper examines the runtime through a governed network-emulation case using cloud and on-premises infrastructure. EVE-NG and GNS3 are treated as replaceable workloads rather than as the HybridOps platform itself. The exercised lifecycle covers provider readiness, workload preparation, private access, guest connectivity, health verification, cost visibility, portable lab-definition persistence, writable node-state preservation, archive and restore, rebuild and teardown.

The paper reports the implemented behaviour, identifies its limits and asks whether the resulting operating contract provides a useful boundary for institutional and individual practical environments.

Keywords: infrastructure contracts; network emulation; lifecycle governance; operational evidence; reproducibility; virtual laboratories.

1. The operating gap

Structured simulators, institution-managed laboratories and learner-managed infrastructure solve different problems. The gap considered here appears when an exercise requires complete operating systems, multiple network platforms, Linux hosts, automation or security tooling, but the environment itself is not the intended subject of the exercise.

Available compute does not close that gap. Cloud capacity or institutional virtualisation still requires identity, networking, approved images, private access, readiness checks, verification, state handling and controlled teardown before it becomes a usable practical environment.

The appropriate abstraction therefore depends on the learning objective. A cloud or platform-engineering module may deliberately expose provider configuration and deployment. A routing, automation or security exercise may instead require a reproducible baseline so that infrastructure setup does not consume most of the available practical work.

Prior research establishes the value of virtual laboratories, simulation and student-managed virtualisation while also showing that no single delivery model fits every course. Structured simulation supports guided networking practice at scale, while open virtual environments support work involving complete systems, automation and failure investigation (Mikroyannidis et al., 2018; Kear et al., 2019; Donelan, Smith and Wong, 2018; Vykopal et al., 2021).

The question for this paper is narrower: whether the infrastructure lifecycle beneath an open network-emulation environment can be made reproducible, inspectable and portable without completing the learner's networking or cybersecurity work.

2. Runtime boundary

HybridOps is a contract runtime for infrastructure operations. A module contract defines inputs, requirements, execution binding, outputs and required checks. A blueprint composes modules into a dependency graph under environment policy.

The runtime validates those contracts, resolves implementation, orders dependent stages and evaluates readiness before execution proceeds. Health checks verify the resulting capability, while environment-scoped records retain the resolved operation and its result.

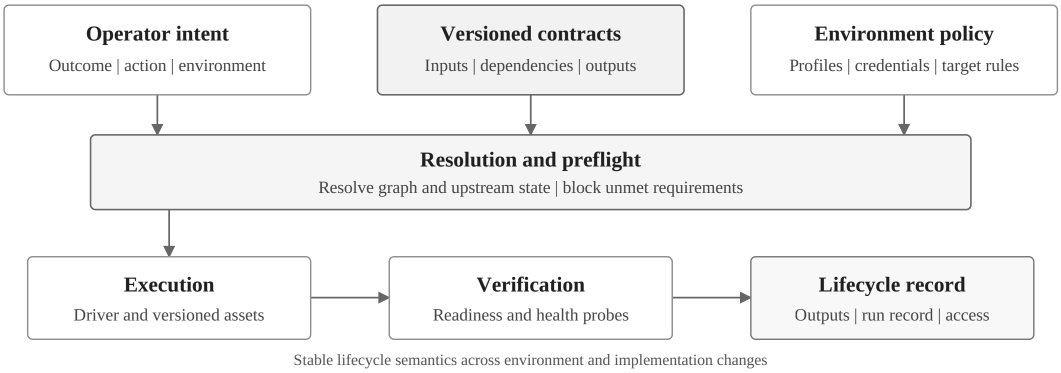


Figure 1. Intent and environment policy resolve through an ordered HybridOps lifecycle before verified state is recorded.

The operator lifecycle can be summarised as:

$$\text{readiness} \rightarrow \text{ordered execution} \rightarrow \text{verification} \rightarrow \text{access and state} \rightarrow \text{teardown}$$

A provider or workload can change below this boundary without requiring a different higher-level operating sequence. This property is the principal technical claim exercised by the network-emulation case.

3. Technical contribution

The contribution considered in this paper is the runtime contract formed by five related properties.

1. **Separated intent, policy and implementation.** The requested capability and environment policy are resolved independently from provider and workload procedure.
2. **Ordered and gated execution.** Dependency order, required credentials, upstream state and readiness conditions are evaluated before dependent work proceeds.
3. **Replaceable providers and workloads.** Infrastructure and network-emulation implementations can change behind the same higher-level lifecycle.
4. **Operational evidence and continuity.** Resolved inputs, non-secret state, verification results and continuity artifacts remain associated with the environment and run that produced them.
5. **Explicit closure.** Rebuild, recovery and teardown are part of the lifecycle contract rather than separate cleanup procedures.

No individual mechanism is presented as a new primitive. The claim under review is whether their composition creates a coherent operating boundary across the implemented provider and workload substitutions.

4. Position relative to existing laboratory platforms

Remote laboratories, simulators and code-defined cyber ranges are established practice. Structured simulators provide guided networking environments; institutional platforms add scheduling, isolation and managed access; cyber ranges demonstrate reusable virtual networks and automated local or cloud delivery.

The relevant distinction is therefore not whether virtual laboratories, automation, remote access or persistent lab state already exist. They do.

Platform	Established primary boundary	Boundary relevant to this review
Packet Tracer	Guided network simulation, activity authoring and collaboration	Tests learning within a supported simulation model
Cisco Modeling Labs	Network simulation with authorised images, persistent labs and APIs	Manages the simulation lifecycle inside a CML deployment
NETLAB+ VE	Institution-managed scheduling, isolated pods, equipment and virtual-machine access	Provides course and user delivery around a managed lab estate
KYPO Cyber Range Platform	Code-defined cyber ranges, training activities and learning analytics	Establishes automated local or cloud cyber ranges as prior work
EVE-NG and GNS3	Network emulation, topology design, consoles and workload state	Operate as replaceable workloads in the HybridOps case
HybridOps	Contract resolution, environment policy, ordered execution, readiness, access, continuity and teardown	Governs the infrastructure lifecycle surrounding the selected workload

HybridOps does not replace the emulator or learning platform. EVE-NG and GNS3 own network-emulation behaviour. Scheduling, grading and learning analytics remain outside the HybridOps runtime.

The claim under review is narrower: whether one infrastructure operating contract can govern readiness, execution, private access, verification, continuity and closure while the provider or emulation workload changes.

5. Origin of the worked case

The network-emulation case grew from a final-year Computer Science project at the University of East London, *Network Programmability and Automation: A Pythonic Approach*.

The project examined software-driven operation of a multi-site network, including automated configuration, validation, monitoring and rollback. A substantial part of the available project time was nevertheless spent preparing the virtual environment required to perform that investigation.

That experience produced the engineering question behind this paper: can the hosting lifecycle become reproducible while remaining visible and inspectable, so that infrastructure setup does not unintentionally become the dominant part of a networking or security exercise?

HybridOps applies its wider runtime contract to that problem. It prepares and verifies the environment while leaving the topology, configuration, diagnosis and security outcome to the learner or operator.

6. Governed network emulation as a worked case

The current case combines two infrastructure routes with two network-emulation workloads. The purpose is not to compare emulator features. It is to test whether changes below the contract require a different operator lifecycle.

6.1 Provider routes

The cloud route creates an isolated network, a private host with nested virtualisation and managed guest egress. Provider readiness covers project access, authentication, billing and secure host access before dependent execution proceeds. Management access is established through a private workstation boundary rather than by requiring a publicly addressed lab host.

The institutional route uses existing Proxmox capacity. Where the required base template is absent, the runtime can build and verify it before creating the lab workload; where it already exists, the verified template can be reused. Workload teardown removes the lab while reusable base capacity remains under an explicit lifecycle boundary.

6.2 Replaceable network-emulation workloads

EVE-NG and GNS3 implement the workload side of the same higher-level contract. Both routes support application readiness, private management access, guest connectivity and workload health without redefining the provider lifecycle.

The emulator remains responsible for topology and node behaviour. HybridOps owns the surrounding infrastructure lifecycle: readiness, ordered execution, access boundary, continuity, verification and closure.

6.3 State preservation and recovery

The continuity model separates a portable lab or project definition from the writable state of virtual nodes. This allows an operator to retain topology and working device state independently from the lifetime of the cost-bearing or capacity-bearing host.

The exercised paths export the portable definition and selected writable node state, verify the resulting archives and restore them after the workload has been recreated and passed health checks. Base device images remain separately managed inputs rather than being redistributed as part of the portable archive. Existing runtime disks are protected from replacement unless restoration is explicit.

6.4 Common lifecycle and evidence

Across the provider and workload substitutions, the case exercises the same operating sequence:

1. evaluate provider and environment readiness;
2. resolve and execute the ordered blueprint;
3. verify workload and guest connectivity;
4. establish private operator access;
5. expose cost posture where provider billing data is available;
6. preserve portable definition and writable node state when required;
7. rebuild or restore through the declared lifecycle; and
8. complete guarded teardown with retained operational evidence.

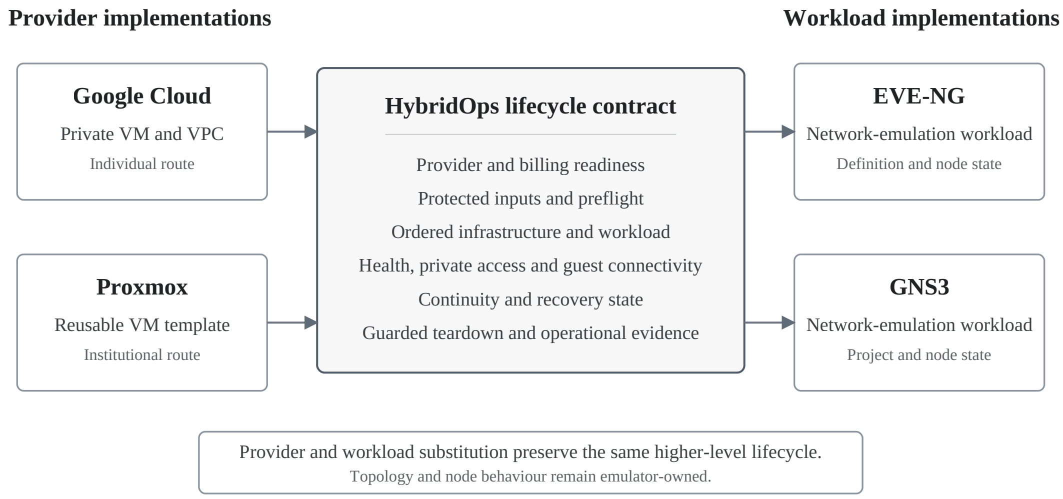


Figure 2. Google Cloud and Proxmox provide infrastructure for replaceable network-emulation workloads through one HybridOps lifecycle contract.

7. Teaching and institutional boundary

The runtime prepares and governs the environment. It does not supply the learner's networking or cybersecurity result.

Suitable activities include:

- network automation with validation, idempotence and rollback;
- routing and fault investigation using protocol state and packet captures;
- controlled connectivity, firewall, segmentation and failure exercises; and
- isolated security exercises in which targets and external access remain explicitly bounded.

The abstraction can therefore change with the learning objective. A networking or security exercise can begin from a verified environment, while a cloud or platform module can deliberately expose the provider and deployment stages.

8. Governance, cost and closure

Institutions and tutors remain responsible for lawful image use, approved training targets and the exercise boundary. HybridOps does not distribute a proprietary network-device image library. Device images remain declared inputs under the operator's licensing and policy controls.

Management access is private by default in the exercised cloud path. Browser or client sessions terminate on the operator workstation, and the cloud host does not require a public address. Readiness gates, health checks and confirmation of destructive operations remain part of the lifecycle.

For the cloud route, the access flow reports billing readiness, resource age, estimated fixed hourly cost and estimated session cost from the deployed VM and disk. Closing the private access path does not stop billable infrastructure, so teardown remains an explicit operation. An estimated ongoing blueprint cost of zero is reported only when managed stages have reached terminal state; otherwise the operator is directed to inspect what remains.

Portable lab definitions and selected writable node state can be preserved before the blueprint-managed host, disk and network are removed. This keeps continuity separate from infrastructure lifetime and makes closure reviewable.

9. Engineering validation and limits

Engineering validation covered the principal lifecycle paths used in this case.

Validation claim	Implemented evidence
Runtime portability	The same contract lifecycle was exercised from Linux, macOS and Windows 11 workstations
Provider substitution	Google Cloud and Proxmox routes reached private, healthy environments through the same higher-level blueprint lifecycle
Workload substitution	EVE-NG and GNS3 completed deployment, private access, guest-connectivity and health paths without redefining the provider lifecycle
Lifecycle continuity	Portable definitions and writable node state were verified, preserved across teardown and restored after redeployment
Governance and closure	Provider readiness gated execution; private access, cost visibility where applicable and confirmed teardown completed through the declared lifecycle

These results establish implemented behaviour, not educational effectiveness or institutional adoption. The case does not establish improved learning outcomes, reduced tutor workload or large-cohort scalability.

The demonstrated provider boundary is limited to Google Cloud and Proxmox, and the workload substitution considered here is limited to EVE-NG and GNS3. Network-device images remain subject to vendor licensing and institutional policy. Cost estimates depend on the provider inputs available at the time of the session and should not be read as a general affordability claim.

Further evaluation should test independent operation, failure recovery, concurrent use and larger institutional cohorts rather than infer those results from the present case.

10. Request for academic and technical review

This paper seeks critical assessment from academics and experienced practitioners in networking, cybersecurity, cloud and computing education. Review can focus on four questions:

1. Is the boundary between HybridOps and the selected network-emulation workload technically coherent?
2. Do the implemented provider and workload substitutions support the claim of one higher-level lifecycle contract?
3. Are the readiness, access, continuity, cost and teardown boundaries strong enough for institutional review and recovery from partial failure?
4. Which existing platform, counterexample or additional evidence most strongly challenges the need for this runtime boundary?

Specific criticism, implementation-level observations and identified limits are welcome.

References

Research and comparison literature

- Donelan, H., Smith, A. and Wong, P. (2018). Virtualization for computer networking skills development in a distance learning environment. *Computer Applications in Engineering Education*, *26*(4), 872–883.
- Kear, K. et al. (2019). An Open Networking Lab for practical open education. OER19, National University of Ireland, Galway.
- Mikroyannidis, A. et al. (2018). PT Anywhere: a mobile environment for practical learning of network engineering. *Interactive Learning Environments*, *28*(4), 482–496.
- Vykopal, J., Čeleda, P., Seda, P., Švábenský, V. and Tovarňák, D. (2021). Scalable Learning Environments for Teaching Cybersecurity Hands-on. *51st IEEE Frontiers in Education Conference*.

Platforms and implementation sources

- HybridOps.Tech (2026). HybridOps Core source, blueprint catalogue and module catalogue.
- HybridOps.Tech (2026). Network-emulation implementations for EVE-NG and GNS3.
- Cisco Systems (2026). Cisco Modeling Labs (accessed 28 July 2026).
- Cisco Networking Academy (2016). Cisco Packet Tracer data sheet (accessed 1 August 2026).
- Network Development Group (2026). NETLAB+ VE software documentation (accessed 28 July 2026).
- GNS3 (2026). GNS3 architecture and server options (accessed 28 July 2026).
- EVE-NG (2026). EVE-NG platform features (accessed 28 July 2026).
- Muibi, J. (2024). *Network Programmability and Automation: A Pythonic Approach*.

Recorded demonstration

Network-emulation lifecycle demonstration

https://www.youtube.com/watch?v=tD7Z9QVK25M

Technical appendix: worked network-emulation routes

The table summarises the provider routes exercised by the case. It records the operational boundary rather than emulator feature differences.

Layer	Google Cloud route	Proxmox route
Workstations	Linux, macOS and Windows 11	Linux, macOS and Windows 11
Infrastructure	Isolated VPC, private VM, managed guest egress and IAP	Existing bridge and DHCP, cloned VM
Workloads	Replaceable network-emulation workload	Same workload contract
Images	Declared or checksum-verified workload image profiles	Same workload contract
Access	Private loopback through IAP and SSH	Private loopback through SSH
Guest networking	Managed DHCP and internet egress	Existing upstream DHCP or managed workload NAT
Health	Workload-specific service and virtualisation checks	Same workload health contract
Continuity	Verified definition and writable-node-state export and restore	Same continuity contract
Cost visibility	Billing readiness and estimated VM, disk and session cost	Provider billing not modelled in this case
Closure	Full blueprint-managed cloud teardown	Workload teardown with reusable base template retained until explicit removal

Supporting figures

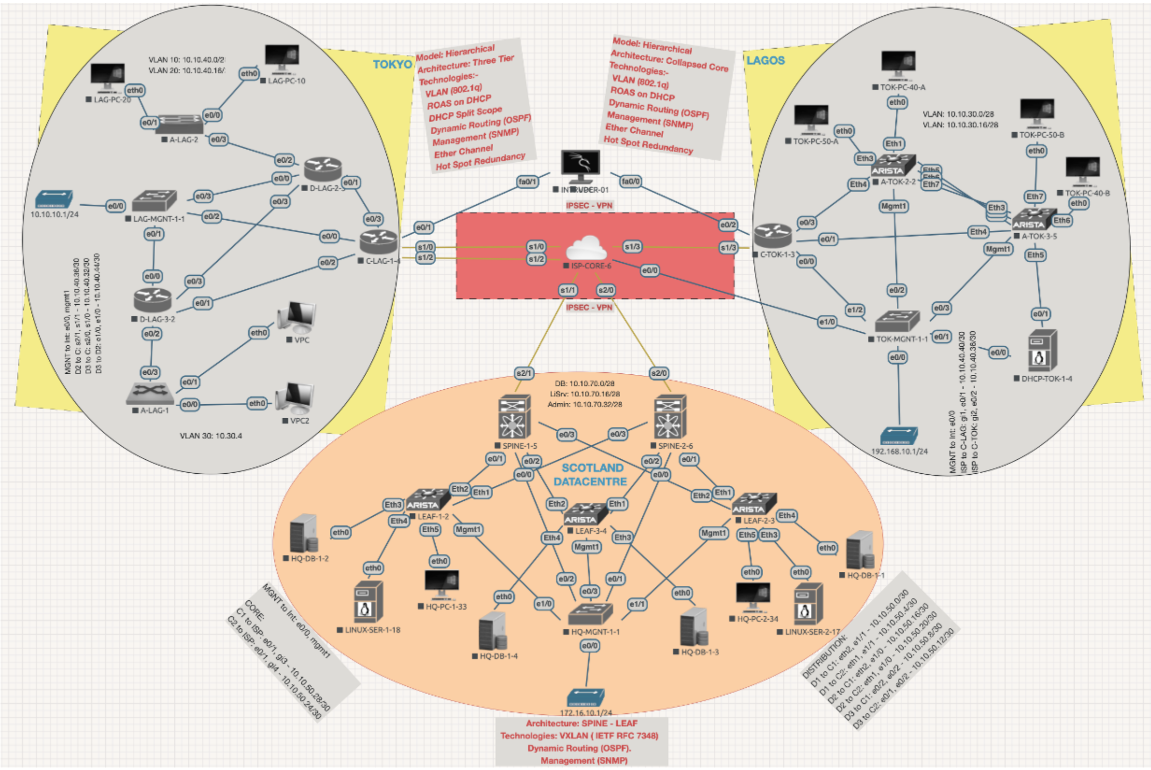


Figure A1. Multi-site network from the 2024 final-year project that motivated the reproducible-environment question.

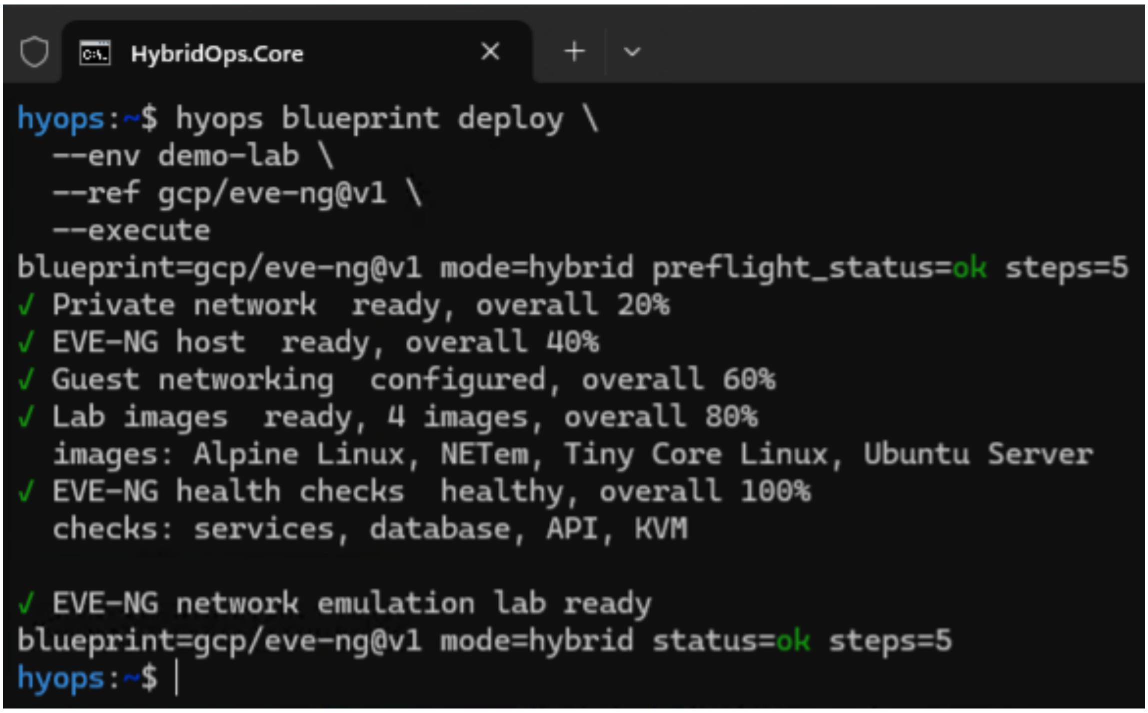


Figure A2. The governed teaching environment completed its ordered deployment and health checks.

Supporting figures (continued)

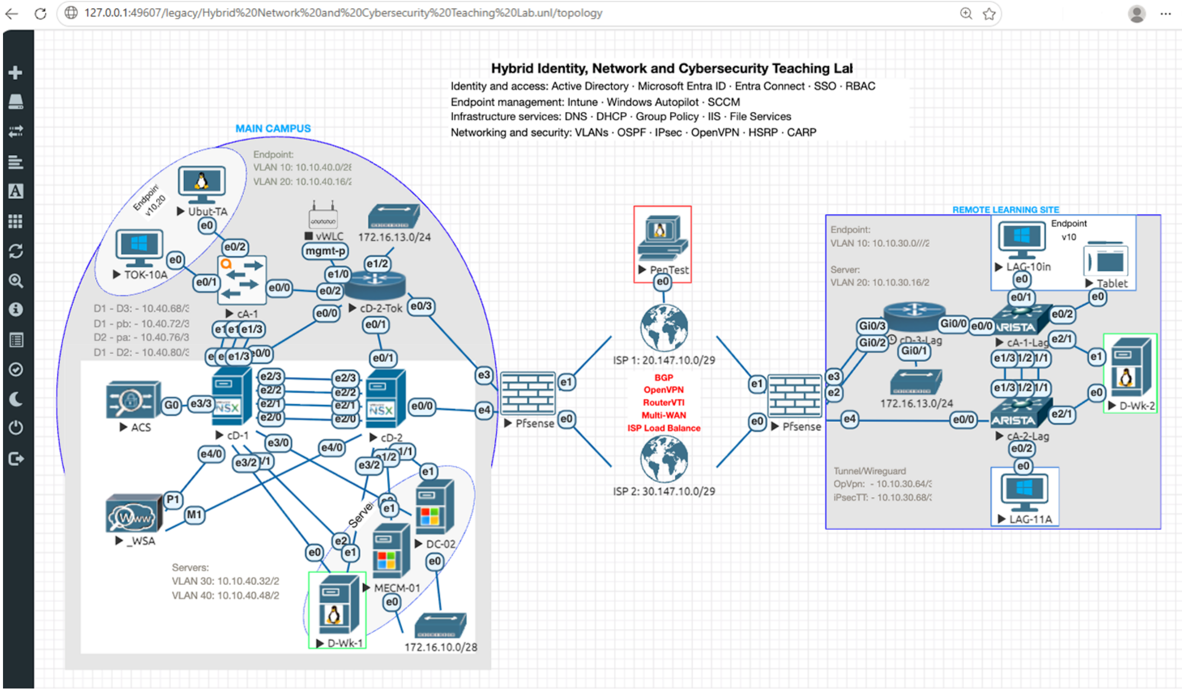
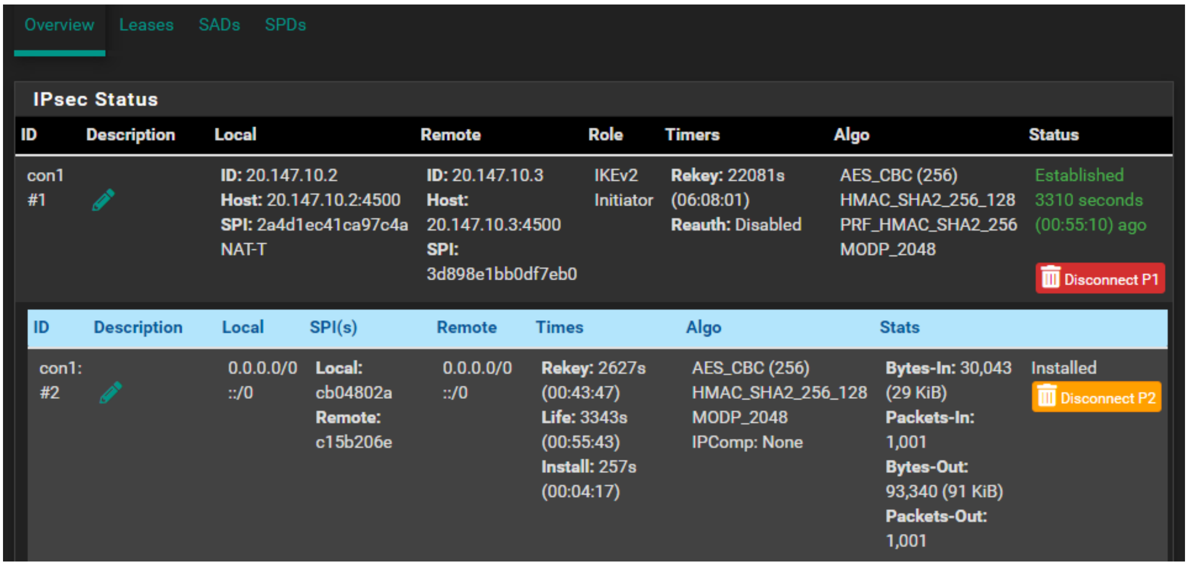


Figure A3. A multi-site network-emulation topology running inside the managed environment.



Overview | Leases | SADs | SPDs

IPsec Status

ID	Description	Local	Remote	Role	Timers	Algo	Status
con1 #1		**ID:** 20.147.10.2 **Host:** 20.147.10.2:4500 **SPI:** 2a4d1ec41ca97c4a NAT-T	**ID:** 20.147.10.3 **Host:** 20.147.10.3:4500 **SPI:** 3d898e1bb0df7eb0	IKEv2 Initiator	**Rekey:** 22081s (06:08:01) **Reauth:** Disabled	AES_CBC (256) HMAC_SHA2_256_128 PRF_HMAC_SHA2_256 MODP_2048	Established 3310 seconds (00:55:10) ago Disconnect P1

ID	Description	Local	SPI(s)	Remote	Times	Algo	Stats	
con1: #2		0.0.0.0/0 ::/0	**Local:** cb04802a **Remote:** c15b206e	0.0.0.0/0 ::/0	**Rekey:** 2627s (00:43:47) **Life:** 3343s (00:55:43) **Install:** 257s (00:04:17)	AES_CBC (256) HMAC_SHA2_256_128 MODP_2048 IPComp: None	**Bytes-In:** 30,043 (29 KiB) **Packets-In:** 1,001 **Bytes-Out:** 93,340 (91 KiB) **Packets-Out:** 1,001	Installed Disconnect P2

Figure A4. Encrypted cross-site connectivity under test, with active security associations and packet counters confirming protected delivery.