



# Automating Your Lab Infrastructure to Drive Down Costs

How to Build the Infrastructure for a Lab as a Service

# Introduction

Test Automation has become a commonly-used term in Application and Cloud companies. Software as a Service (SaaS) and Cloud-based applications dominate the product development landscape in the B2B and B2C markets. Test Automation in the form of continuous regression of these applications has become a widely-used practice in AGILE and Container-based Dev Ops environments.

This paper will focus on a different area of Test Automation: how to build the test lab infrastructure for a Lab as a Service (LaaSS), or Lab Consolidation. Specifically, we will discuss how to automate a test lab infrastructure, and ensure the proper “plumbing” is in place to maximize the value of a Lab as a Service (“LaaSS”) or Lab Consolidation project.

## What is a LaaS?

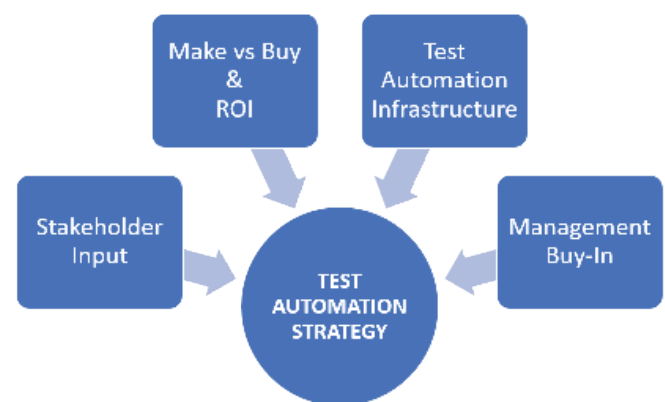
A LaaS is a centralized test lab initiative that acts as a test service available to the entire company. Lab consolidations and Labs as a Service represent a huge advance in the areas of testing efficiency, scale testing, and lab security. By utilizing a LaaS model, companies can eliminate the need for many small distributed test labs by replacing them with a single, or a small number of larger more efficient labs. These new labs would be constructed in areas where power, cooling, and real estate are optimal, and they would offer 24 x 7 availability to test users across the company, regardless of physical location.

Building test lab infrastructure correctly from the start will enable scale and remote sharing of lab resources in order to achieve a high level of functionality and to transform the Pre-Production, Network Equipment Lab, or QA lab into a central testing service for the entire company. This approach provides substantial benefits, including:

- Significant CAPEX and OPEX savings by sharing labs and lab equipment
- Increased quality by testing at scale
- Reduced time to market by optimizing test equipment and resources
- Increased test efficiency by having 24x7 testing and scheduling of lab resources
- Continuous security validation
- High quality analytics and KPIs to manage the test process
- Interoperability testing
- Incorporate machine learning and AI
- Performance testing

It is important to note that there are many facets to consider when developing a lab automation strategy. This process requires careful planning, beginning with a well-architected solution based on inputs from all stakeholders. The initiative must also gain the support of the senior management team, therefore a strong, well-reasoned ROI strategy must be in place.

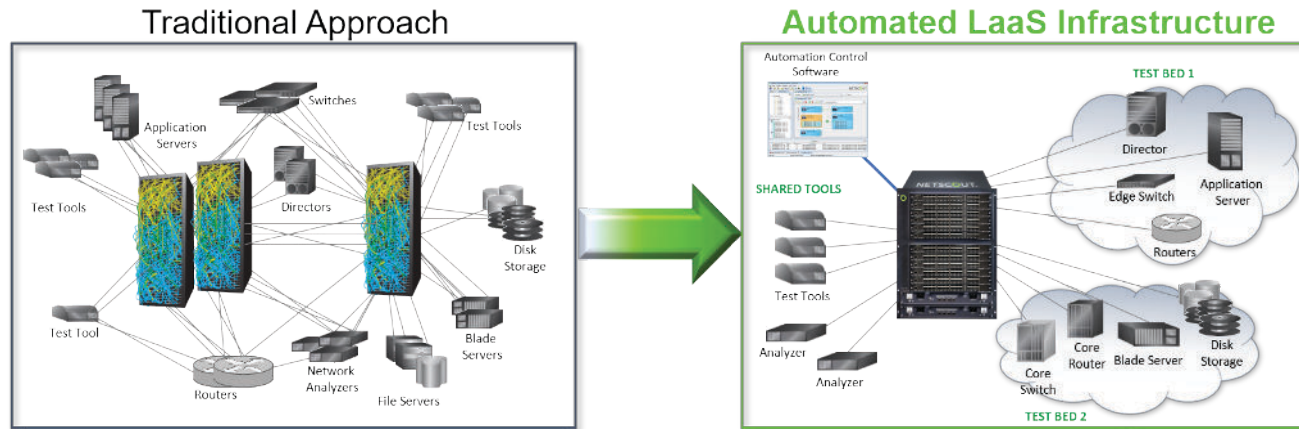
The automation strategy must consist of a well-thought-out automation infrastructure layer. There will also be a multitude of make versus buy versus open source decisions that will need to be made when architecting a test automation strategy.



## What are the Key Ingredients of an Automated Test Infrastructure?

From a testing value standpoint, a LaaS approach offers all the benefits described above. At the same time, this approach also optimizes OPEX and CAPEX for the business. The figure below illustrates how a lab can be transformed into a LaaS by automating test infrastructure. The components for an automated LaaS infrastructure include two main building blocks:

1. A centralized Layer 1 Switch array
2. Lab Automation Infrastructure Management Software



The traditional approach is often plagued with:

- Poor utilization of tools
- Wasted time spent on complex configurations
- A requirement for continuous expenditure on test tools

An automated LaaS infrastructure approach:

- Enables a dramatic increase in tool utilization
- Speeds the time to set up and complete tests
- Can provide substantial savings in both CAPEX and OPEX

## Why Layer 1 Switches versus Layer 2 Switches?

Layer 1 switches are unique in that they act like an automated patch panel, which enables instantaneous, lights-out reconfiguration of your test resources. Layer 1 switches enable test users to reserve and configure test equipment, DUTs, Traffic Generators, etc. by simply logging in remotely, authenticating and then creating and scheduling their topology and test job. The Layer 1 switches are essentially invisible and act like a virtual wire, enabling test users to rapidly create, schedule, and apply test topologies without even needing to know where in the world the physical lab assets are deployed.

Sometimes lab architects will consider using Layer 2 switches instead of Layer 1 switches to automate their test infrastructure. This may be viable in certain environments, but it is important to understand the ramifications involved. The table below summarizes some of the key differences between Layer 1 and Layer 2 Test Automation Infrastructure.

It is vital that the Test Automation Infrastructure does not induce such things as high or variable latency into the test lab. It must simply act like a "wire" so it does not introduce this variability, which could cause invalid test results. Additionally, since the purpose of testing is to expose and raise errors to the surface, it is important that the Lab Automation Infrastructure switches do not delete or modify packets containing errors. Layer 2 switches can drop and/or modify bad packets by design, however, Layer 1 switches do not. Layer 1 switches provide dedicated circuit-based connections from test device A to device B. They provide guaranteed delivery, unlike Layer 2 switches that do not guarantee packet delivery. Layer 2 switches can also insert and delete primitive characters, thus changing the timing of data flows. Layer 1 switches do not introduce these special characters. Finally, Layer 1 switches provide an ultra-secure connection that cannot be mis-routed to the wrong destination device.

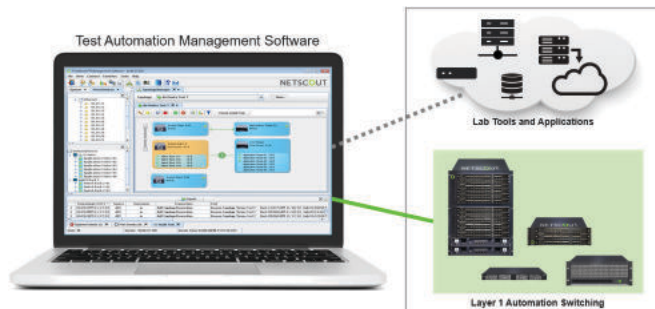
For these reasons, the Layer 1 approach is highly recommended over Layer 2 switches.

	Layer 1 Test Infrastructure	Layer 2 Test Infrastructure
<b>Latency</b>	5 nanoseconds – 100 nanoseconds	3 microsecond – 100s of milliseconds
<b>Error Pass Through</b>	Yes	No
<b>Dedicated Connection</b>	Yes – circuit-based connection, Guaranteed delivery	No – Packet switched, Non-guaranteed delivery
<b>Variable Timing</b>	No need to insert or delete primitive characters	Will change the timing via add/delete of primitive characters
<b>Security</b>	Mis-routing and CPU packet Data access not possible	Packets can get mis-routed or vectored to the CPU



## Lab Automation Infrastructure Management Software

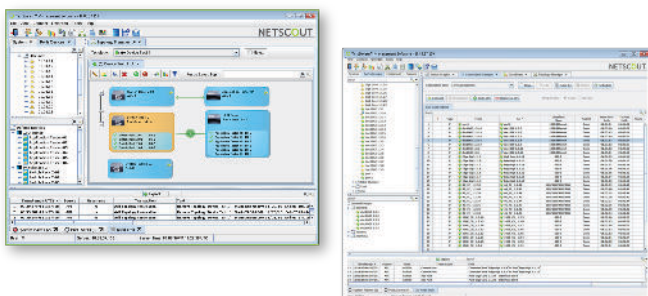
In conjunction with Layer 1 Switching, the second critical element in deploying a LaaS approach is the Lab Automation Infrastructure Management Software itself. This will provide the user with the necessary management of the Layer 1 switch array, topologies, reservations, security, reports and analytics, APIs, inventory, and partitioning. There are many different flavors of this type of software. When architecting the LaaS, there will be many choices ranging from commercial software, to open source, to custom or homegrown. Ideally, an integrated solution, where the Layer 1 switches and the Lab Automation Management Software are acquired as a complete solution, is desirable to maximize capability and minimize costs and integration time.



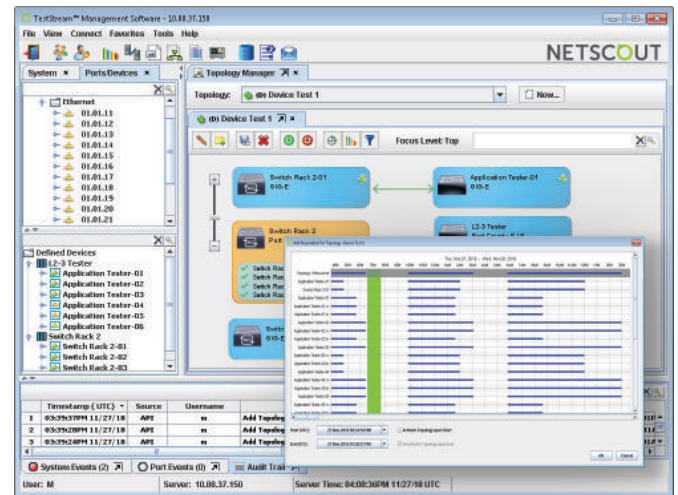
The Lab Automation Infrastructure Management Software acts as the central hub that the test user operates from to set up, schedule, and deploy their test topologies. This software should be a centralized resource management server capable of supporting hundreds of test users and thousands of test devices, all operating simultaneously. This software can be graphical in nature or it can be managed by some other 3rd party orchestration software through an API, discussed below.

Here are the critical functions required for Lab Automation Infrastructure Management Software:

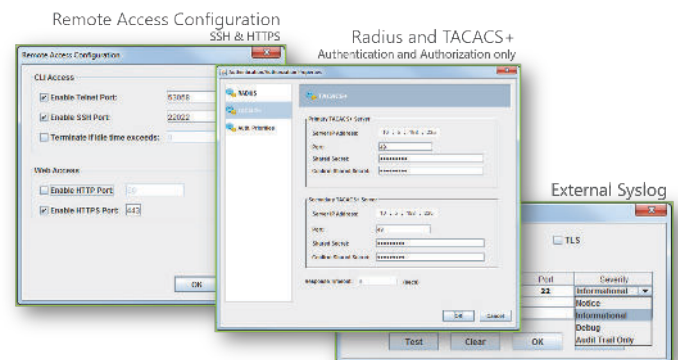
**Test Topology Editor** – An intuitive drag and drop graphical editor that can be used to develop test topologies by the test users. The topology editor function should house a database of all available test devices. The test user should be capable of defining bi-directional, one-way, multi-cast, and tap connections between the test devices. The device tree should be expandable so that it will scale to very large test topologies. It should also have a tabular device connectivity summary view to augment the graphical view. The Editor should allow the test users to customize names of devices. The Topology Editor will interconnect the test devices using the Layer 1 switch(es) at the moment in time that a given topology is scheduled. All the test user sees is that he or she connected Test Device A to Test Device B.



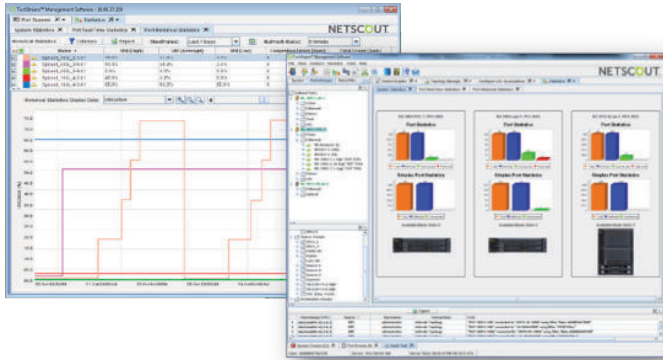
**Test Resource Reservation** – The ability to schedule topologies is critical. For example, let's assume you need to run a test for some application that requires 7 Servers, 5 Layer 2 Switches, and a Core Router along with ten 100G Ethernet Traffic Generators. The test user would draw the desired topology, interconnecting the devices required for the test. The test user would then invoke the scheduler function, which provides a graphical view of the timeslots that exist where each of the required devices are available. Ideally, the reservation tool will also help the test user to resolve conflicts with any devices at the desired timeslot.



**Security and Authentication** – The ability to authenticate a given test user by Radius/TACACS or some other means of validation is also important. The software should allow good granularity for the administrator to manage which devices and functions a given test user should have access to. These permissions would be configured by the Admin user.



**Reports and Analytics** – It has been said, "if you can't see it, you can't manage it." How will I know when my test resources are nearly exhausted, and I need to do some capacity expansion? How will I know if I have some test resources that are under-utilized or have been reserved for a period of time but are not in use? Having graphical reports on utilization, errors, and alerts is imperative to managing the LaaS. These reports should be flexible and customizable. The reports should also be exportable to Excel files or other common formats.



**API** – EachaaS deployment will have unique properties and functions. Inevitably, there will be proprietary lab management utilities that require integration with the Lab Automation Infrastructure Management Software Automation API. A straightforward common format such as REST or Python generally works well in test labs.

**Configuration** – There will be many test lab devices that need to be individually configured each time a new test topology is deployed. This process includes going out to each network element (IP address) involved in a particular test topology, and setting up the proper software revision, drivers, configuration of all ports and system attributes, and deploying VMs, to name a few. Often Test Administrators will choose to use a production tool for this rather than re-inventing the wheel. The production tool can be easily integrated with the Lab Automation Infrastructure Management Software.

**Inventory** – The Lab Automation Infrastructure Management Software should be able to maintain a database of all test devices. Ideally, this module should have the ability to auto-detect when devices are added or removed from theaaS, but some lab architects prefer to make this a manual process.

**Partitioning** – The Lab Automation Infrastructure Management Software should ideally provide the ability to divide your automated test lab and the equipment within it into secure logical domains, to fence off resources between groups and users.

**Reset** – A seemingly simple feature, but often overlooked is the ability to reset a given test topology and return everything to a known state. The Lab Automation Infrastructure Management Software must provide the ability to return all network elements involved in a particular test topology to a neutral state. What this means may vary depending on the environment.

**Advanced Test Automation Infrastructure Services** – Extended services such as Aggregation, Filtering, Impairment, and Rate Conversion are examples of advanced Lab Automation Infrastructure Management features that can help testing. For example, if 100G traffic generation is required but not available, an Aggregation function can be used to amplify and replicate generated traffic from a 10G traffic generator to achieve 100G line rate. As a second example, imagine the need to characterize how an application operates across a WAN link. A feature set that provides the test user the ability to configure a packet loss rate of 10%, or a delay behavior of 100 milliseconds, or even network jitter is quite helpful in testing how the application performs in a typical loss environment.

## Conclusions

Implementing a Test Automation solution has become very popular for testing both application and network products. Many businesses are exploring, as a next step, how they can take their test automation strategy to the next level. Building larger test labs that can be shared as a Lab as a Service and consolidating smaller labs into larger centralized lab(s) is a logical next step for many organizations. The purpose of this paper was to discuss how specifically to automate your test infrastructure so that the benefits of larger shared labs can be realized. The benefits are significant in terms of building better products and reducing time to market, while also reducing your overall OPEX and CAPEX.

NETSCOUT® Test Automation family of products delivers the gold standard in Lab Automation Infrastructure Management. They offer the only combined software and hardware solution in the industry, thus enabling customers one-stop shopping when it comes to building a centralized Lab as a Service. Please contact NETSCOUT today to learn how you can save money and significantly increase the quality of your testing.

# NETSCOUT®

## Corporate Headquarters

NETSCOUT Systems, Inc.  
Westford, MA 01886-4105  
Phone: +1 978-614-4000  
[www.netscout.com](http://www.netscout.com)

## Sales Information

Toll Free US: 800-309-4804  
(International numbers below)

## Product Support

Toll Free US: 888-357-7667  
(International numbers below)

NETSCOUT offers sales, support, and services in over 32 countries. Global addresses, and international numbers are listed on the NETSCOUT website at: [www.netscout.com/company/contact-us](http://www.netscout.com/company/contact-us)