

## Managing Complexity in Dev/Test Infrastructure with Machine Learning

By: Ajay Gulati

Businesses are increasingly developing software solutions that can drive innovation and digital transformation. As the number of developers increases in the organization, the ratio of developers to IT also increases over time. The net result: more dynamic and complex development and testing (dev/test) infrastructure requirements and slower support from IT.



Developers and testers often cannot get their hands-on infrastructure environments in a timely manner to accelerate delivery of software applications to their customers. They have to deal with lack of timely support from IT and long wait times for hardware resources and for provisioning and configuring of environments.

How does engineering IT manage this complexity and deliver more agility to their engineering stakeholders?

## Provide Self-Service Creation of Dev/Test Sandbox Clouds

Engineering IT teams can increase the velocity of development and testing if they enable self-service ability for dev teams to quickly provision their own lab infrastructure and application stacks with the click of a button.

Self-service, API-driven infrastructure is a fundamental requirement for enabling developers to write code — which can be done using their favorite programming language — and make RESTful API calls to the underlying programmable infrastructure. This allows them to dynamically and automatically manage initial deployments and configurations, as well as manage ongoing, automated dynamic provisioning of infrastructure, autoscaling, monitoring, and alerting.



All this automation removes the confusion and error-prone manual steps for the entire application delivery process, including development, testing, staging, and production deployments. This, in turn, accelerates software delivery and increases quality.

Additionally, if self-service is available as a SaaS-based delivery model, it becomes easier to add

more features and workflows quickly without forcing IT to handle a major upgrade.

Breaking these silos improves collaboration between teams, accelerates velocity of software development, improves infrastructure utilization, increases overall operational efficiency, and reduces costs.

## **Eliminating Silos in the Dev/Test Infrastructure**

Silos are common in companies where various specialist teams (storage, networking, security) form fiefdoms around their respective functional areas. Silos impede velocity because they lead to complexity of operations, lack of consistency in the environment, and lack of automation.

Automating across silos turns into to an exercise of custom scripts and lot of “glue and duct tape,” which makes maintenance and change management complex, slow, and error-prone.

Breaking these silos improves collaboration between teams, accelerates velocity of software development, improves infrastructure utilization, increases overall operational efficiency, and reduces costs.

A hyper-converged cloud design with a software-centric, scale-out architecture tightly integrates compute, storage, networking and virtualization resources and other technologies from scratch in a commodity hardware box supported by a single vendor. Also, companies can keep costs under control by leveraging scale-out cloud designs that make it easy to start small, grow based on demand, and stay close to the right size and customer demands.

Additionally, development teams should have the ability to quickly deploy/clone/share complex multi-tiered application stacks between various teams and between development and testing, breaking the silos within the development organization

## **Automating Operations, Monitoring and Patching**

Engineering IT teams need to have complete visibility and control over their entire stack from the infrastructure up to applications. They need intelligent software to monitor the hardware and software stack, and to manage large-scale clusters, as well as automatically handle routine but time-consuming and complex operations such as failure handling, patching, security updates, and software upgrades.

Running a service with traditional sysadmin teams who execute the above activities manually becomes expensive, especially if they operate in server, storage, networking, and security silos. The costs only increase as dev/test environments become more dynamic and the demand for more environments and projects grows.

However, it is possible to use an intelligent private cloud platform that leverages hyper-converged scale-out designs, machine learning software, and a SaaS-based operational console to reduce complexity and increase the agility of dev/test teams.

Cloud-based monitoring and advanced analytics dramatically reduce the need for experts in different parts of the infrastructure, scale linearly as the size of the operation increases, and cut operational complexity by 90 percent.

## **Manage Resource Management Using Machine Learning**

Applying machine learning to infrastructure management means intelligent software could learn about operational patterns, anticipate capacity needs, raise alerts about security anomalies, self-monitor and self-heal environments in the face of failures. It becomes possible to intelligently apply security patches and automatically upgrade hardware and software systems without any downtime. The next generation of infrastructure management will be driven by advances in machine learning and artificial intelligence, where the infrastructure is able to basically “drive itself” with minimal user intervention.

This will help engineering IT teams optimize resource usage and capacity based on current and future dev/test demand, as well as better handle the availability and performance they can deliver to their engineering teams. A lot of efficient resource management comes down to capacity planning, utilization monitoring, right-sizing of workloads, demand forecasting, and detecting zombie virtual machines and unused resources.

Demand forecasting and capacity planning can be viewed as ensuring that there is sufficient capacity and redundancy to serve projected future demand with the required availability. Capacity planning should consider organic growth, which stems from natural service adoption and usage by dev/test teams. Having intelligent predictive analytics and machine learning can greatly help with accurate forecasting, alerting, and providing lead time for acquiring additional capacity.

Better insights into how the infrastructure is performing can also help in fine-tuning performance of end user workloads. For example, an intelligent system that is monitoring a workload for storage performance might recommend using solid state drives (SSDs) instead of spindles to increase the IOPS and improve workload responsiveness.

An intelligent private cloud platform that uses hyper-converged scale-out designs, machine learning software, and a SaaS-based operational console can reduce complexity and increase the agility of dev/test teams.