Continuous Cloud Validation
Cope with the Uncertainty of Software-Defined Clouds
this
page
intentionally
left
blank
TABLE OF CONTENTS

1. SOFTWARE-DEFINED CLOUD UNCERTAINTY. .................................................. 1
   1.1. Uncertainty Factors ................................................................. 1
   1.2. Validation Dilemma ............................................................... 2
       1.2.1. Lure of Freeware ......................................................... 4
   1.3. Cloud Stakeholders ............................................................... 5
       1.3.1. Stakeholder Barriers ..................................................... 6

2. COPE WITH UNCERTAINTY – CONTINUOUS CLOUD VALIDATION. ............... 7
   2.1. Cloud Lifecycle Checkpoints .................................................. 7
   2.2. Multi-Dimension Automation ................................................ 8
       2.2.1. Orchestration Dimension ............................................ 8
       2.2.2. Emulation Dimension .................................................. 9
   2.3. Validation Across Boundaries .............................................. 9
   2.4. Continuous Cloud Validation Use Cases ................................. 10

3. CONCLUSION .................................................................................. 12
1. Software-Defined Cloud Uncertainty

1.1. Uncertainty Factors

Data centers in the past were rigid with predictable 1-to-1 mapping of services or applications to infrastructure. Today's cloud data centers are built for multi-tenant or multi-service workloads whereby the infrastructure is designed to be shared by tenants in the case of public clouds or shared by many applications or services in private clouds used in IT organizations.

![Figure 1. Software-Defined Cloud Uncertainty](image)

The business advantages to employing cloud capabilities is documented ad nauseam. Centralized investment and management allows operators to reduce costs while cloud consumers gain elasticity and agility in delivering new applications and services. The agility has driven the industry to move towards defining all layers of the cloud stack in a software-defined fashion. This creates conditions for on-demand provisioning of resources by tenants or consumers of the cloud in an elastic fashion. The terms orchestration and automation are synonymous with software-defined and are being used to define the control processes that allow for these flexible provisioning and deployment models.

The agility and abstraction that a software-defined cloud requires comes with cost, which is highlighted in Figure 1 with an exponential rise in data center operating uncertainty when cloud flexibility is increased. The uncertainty and validation burden is placed on those responsible for delivering the cloud.

The new software-defined cloud poses several key technical challenges to cloud stakeholders illustrated in Figure 2 including:

- Integration of the many layers that comprise a software-defined infrastructure
- Orchestration that continually changes an infrastructure that was once static to support multi-tenant/service agility
- Increasingly unpredictable application workloads that can be chaotic to the infrastructure; especially when ‘noisy’ tenants wreak havoc on the shared infrastructure
The term cloud stack is used in this paper to denote the major layers (compute, virtual, storage, network, etc.) of a software-defined cloud that can be controlled independently. Cloud stakeholders are faced with the challenge of integrating these many layers and maintaining a high level of complexity within the data center, which creates a tremendous amount of operating uncertainty. Performance and security must be sustained reliability over the chaotic lifetime of services that come and go.

The lifecycle of the cloud involves many checkpoints that span from the development of cloud stack layers with technology through to operators selecting and deploying stacks to DevOps consuming the infrastructure to deliver the applications and services.

1.2. Validation Dilemma

The task of validating older 1-to-1 data center models of apps to infrastructure was easy compared to the burden stakeholders face today. The diverse combinations of cloud stack layers that can be assembled, the unpredictability of multi-tenant workloads and real-time orchestration demands on infrastructure create a large multi-variable equation to validate against. The permutations that test engineers need to cover is daunting. Manually testing these permutations is nearly impossible in a cost- and time-efficient manner. A non-exhaustive list of steps to validate one permutation manually include:

1. Acquiring hardware and software (compute, storage, network, security)
2. Interconnecting the network between the layers
3. Installing the virtual infrastructure on top of the compute
4. Deploying virtual machines and tuning compute resource assignment
5. Debugging and tuning the configurations of the components/layers on top of the virtual infrastructure
6. Validation time that includes planning, configuring, executing, interpreting and reporting using test solutions
These steps are not only required by cloud service providers but also by the technology vendors. Each vendor must be able to setup complex lab environments to integrate multi-vendor solutions together and build cloud stack ecosystem combinations. These lab deployments are required to determine how vendor components will function alongside other stack layers. It is difficult to arrive at any operating uncertainty conclusions by building stack variations and validating manually.

How can stakeholders cope with the uncertainty that amorphous software-defined clouds present? The solution requires the ability to automate with orchestration and emulation in all key dimensions of the cloud stack layers. Otherwise, the problem is too large and all cloud stakeholders don’t invest enough in validation. This in turn often leads to cloud outages, security breaches and resource over-provisioning.

The validation dilemma that software-defined clouds create requires fresh approaches and new solutions from test and measurement vendors. To solve the problem a validation solution must be capable of:

- Automating the orchestration and emulation of cloud layers to cover all key cloud personas
- Orchestrating mixtures of real infrastructure and emulated infrastructure through suitable test interfaces without costly lab build-outs for each iteration
- Controlling emulation in all dimensions within the cloud stack layers including:
  - **Infrastructures**: Compute, Virtual, Network Function Virtualization (NFV)/Software-Defined Networking (SDN) Storage, Security
  - **Services/Applications**: Application workload modeling, API emulation, Denial-of-Service/ Malware emulation
- Spanning lab and live testing scenarios whereby live cases are used for validating public cloud services where setting up private lab environments is prohibitive
- Facilitating agile development approaches with continuous validation to incrementally expose issues
- Allowing stakeholders the ability to ‘role-play’ by emulating components they aren’t responsible for example, operator simulating application workloads DevOps deploys
- Testing from the application down for system validation and isolating by emulating surrounding layers to pinpoint issues
1.2.1. Lure of Freeware

The use of test freeware can be particularly enticing to cloud stakeholders. DevOps teams can be lured into software development that leverages freeware, as they are capable of quickly assimilating simple test harnesses using abundant cloud resources at their disposal. Momentum for freeware initiatives can build at the outset of new projects because early success can be achieved. However, these freeware hacks quickly draw in more and more resources over time to deal with a myriad of deficiencies in such approaches that are outlined below.

• Non-existent result repeatability will frustrate those trying to characterize performance
• Developing business logic to control test end-points and harvest results draws ever increasing investment in resources and time
• Poor instrumentation for realistic emulation fails to expose system flaws in the cloud stack
• Hidden costs become apparent later in development cycles associated with equipment costs and operating costs in managing virtual layers, sustaining test harness platform capabilities and supporting the harness across the organization

But perhaps the most relevant argument countering freeware for software-defined cloud validation is its inability to operate in multiple dimensions to emulate the rich set of cloud stack layers. Organizations that lack a massive commitment to freeware hacks usually do realize that they cannot be used effectively in the long-term to solve the software-defined cloud validation dilemma.
1.3. Cloud Stakeholders

The movement towards centralized computing has created new divisions between stakeholders involved in delivering cloud services and applications. Cloud service providers can broadly be divided into infrastructure operators and a new breed of skillset classified as DevOps. While the distinction between the two will vary depending on organization, the split falls somewhere between services/applications and the virtualized compute infrastructure. There is further specialization within these two categories but abstracted here for simplicity.

![Cloud Stakeholder Roles](image)

Figure 3. Cloud Stakeholder Roles

Figure 3 shows a simplified diagram of the division of roles between Infrastructure Operators and DevOps. Ownership of the emerging orchestration functions can be divided in different ways depending on the type of cloud service being delivered. Private clouds run by IT operations can also be considered cloud service providers with infrastructure operators and DevOps job functions. They will typically have the same operating challenges and validation issues for the purposes of this paper.

DevOps is an emerging role and the skill set can be characterized as a third system administrator, a third software developer and a third cloud computing expert. This blend of skill set is required to provide the glue, which stitches shared infrastructure(s) to the delivery of services and applications. DevOps teams are responsible for delivering applications and services with a high quality of experience (QoE) on top of shared infrastructures they do not own, which poses accountability issues. Testing different public and private infrastructures for design and selection purposes is time consuming. They often need live ‘over the Internet’ test capability in order to validate pre-production applications and services, test for geographical data center placement and harden applications against current cyber threats/vulnerabilities. DevOps use Continuous Integration (CI) development approaches to help cope with these challenges and test solutions must support CI processes in order to be useful to them.
Infrastructure operators are those who are responsible for delivering the physical hardware including networking, compute, security and storage and are usually responsible for the virtual tier. The industry uses the terms Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) to characterize an increasing level of abstraction provided to cloud consumers. This paper does not attempt to use these terms. Instead, it simply uses the labels Operators and DevOps as the two key stakeholders driving the entire cloud stack from the applications down to the network. Deployment and integration of layers for operators can be lengthy without flexible emulation and integrating multi-vendor solutions can be costly. Operators require test solutions that are able to emulate diverse infrastructure layer dimensions to isolate and pinpoint bottlenecks. Furthermore, test solutions must be able to generate realistic, multi-tenant, application workloads to allow operators to ‘role-play’ to reduce the sometimes painful handoff with DevOps.

Technology vendors provide the building blocks to operators that assemble the cloud stack layers. They will typically face the same challenges operators face in validation because their solutions must fit within the cloud stack ecosystem. However, vendors require more depth and sophistication in test emulations.

1.3.1. Stakeholder Barriers

The evolving boundaries between stack layers and job functions pose new barriers to the proliferation of software-defined data centers. How can system validation be performed if no one party has ownership of the cloud stack top to bottom? How can stakeholders ‘role-play’ to ensure the hand-off between responsibilities is performed efficiently?

Stakeholders need automated solutions that allow them to emulate cloud layers in multiple dimensions and perform validations across layer ownership. The goal is sustained performance, reliability and security for cloud providers and consumers. This requires a continuous validation approach by all stakeholders at key integration points with multi-dimension automation.
2. Cope with Uncertainty – Continuous Cloud Validation

All cloud stakeholders can cope with the uncertainty of software-defined clouds with Spirent’s Continuous Cloud Validation (CCV). It facilities performant, reliable and secure service delivery over software-defined clouds by empowering stakeholders to validate cost-effectively within short time-scales. The new validation process is being used to converge on optimal cloud infrastructures and services.

The recent unveiling of Spirent’s Velocity and Virtual Private Test Cloud (VPTC) solutions orchestrate the breadth of Spirent’s portfolio and combined, deliver continuous cloud validation. The crux of the strategy is to continuously orchestrate and validate in order to automate and eliminate uncertainty.

2.1. Cloud Lifecycle Checkpoints

Cloud stakeholders are increasingly using Continuous Integration (CI) to support incremental development. Continuous cloud validation allows stakeholders to validate at many integration points within CI in an iterative manner to expose bottlenecks early and collaborate with fellow stakeholders more effectively. Architects and designers working for technology vendors can employ it to improve designs answering ‘what-if’ scalability questions that would otherwise be unwieldy to answer with conventional methods. Test engineers can plan more effectively and be more productive during testing phases by executing many more permutations in shorter intervals to improve product quality.

Vendors and operators delivering piecemeal components of cloud infrastructure can characterize how their solutions will interact with other open-source and commercial cloud layer solutions. Cloud providers can chip away at the uncertainty problem by performing validations more frequently during cloud lifecycle check points empowering them to:

- Improve architecture during design phases
- Make more informed vendor selections during tight decision windows
- Improve performance and scaling by exposing bottlenecks and tuning stacks
- Reduce deployment times performing integrations more effectively
- Perform penetration assessments on infrastructure security provisions continuously to mitigate potential attacks in shared cloud
- Harden applications to counter the rise of security breaches and denial of service attacks

Stakeholders can role-play emulating portions of the cloud stack they lack ownership of. This provides a common validation language for stakeholders to collaborate in more meaningful ways allowing them to deliver on the promises of software-defined clouds.
2.2. Multi-Dimension Automation

Continuous cloud validation requires multi-dimension automation to tame the software-defined cloud with software-defined control of test and cloud infrastructure. Spirent Velocity allows virtual test emulation interfaces and real virtual infrastructure to be orchestrated to validate many cloud stack topologies in an automated fashion. Spirent's new VPTC solution is a ‘sandbox’ that allows real infrastructure to be orchestrated alongside virtual infrastructure under test. Users can use pre-provisioned and/or upload virtual appliances and virtual network functions (VNF) to form various NFV service chains that create the virtual infrastructures under test.

2.2.1. Orchestration Dimension

Deploying different combinations of cloud stack layers is a time consuming proposition. It limits the number of permutations test engineers can validate within fixed testing cycles. The Spirent Velocity and VPTC solutions support the orchestration of real and test infrastructure boosting the productivity of test engineers by allowing them to cover a much wider swath of combinations. Spirent Velocity can be used with VPTC or separately for maximum productivity gains. The orchestration planes include:

- **Real Infrastructure**: Spirent VPTC is a lab blueprint that allows real infrastructure such as computing power, hypervisor variants, and virtual machine resources to be quickly provisioned in a cloud environment. Place virtual components or VNF’s within real infrastructure without costly lab build-out and ongoing operating expenses.

- **Emulated Infrastructure**: Spirent Velocity orchestrates virtual test emulation interfaces in an overall cloud topology or NFV service chain.
2.2.2. Emulation Dimension

Spirent’s holistic product set uniquely positions them to emulate all key cloud stack layers in various planes. This includes

- **Workload Plane**: emulate realistic mixtures of application client, client to server or service to service API interfaces that are capable of generating massive loads with highly controlled instrumentation.

- **Storage Plane**: emulate storage file and block clients/imitators to find the scalability and reliability limits of storage infrastructure.

- **L2-3 Data-Plane**: precise and deterministic L2-3 host emulation allows network switching fabrics and virtual/physical network boundaries to be plumbed for scalability.

- **Threat/Attack Plane**: emulate attack interfaces to launch malware, application exploits and denial of service attacks to harden network security and application countermeasures against the latest vulnerabilities.

- **Routing Control-Plane**: carrier-scale routing topologies can be emulated behind physical and virtual interfaces across interior and exterior routing protocols.

- **Data Center Bridging Control-Plane**: emulate data center switching topologies that support multipath and high availability protocols.

- **Software Defined Network (SDN) Control-Plane**: emulate SDN controllers and switches to isolate SDN elements and find performance bottlenecks.

- **Virtual Orchestration-Plane**: emulate infrastructure orchestration layers that add churn to the virtual layer including performing live VM migrations.

These extensive Spirent orchestration and emulation capabilities enable the automated orchestration/validation cycle that continuous cloud validation promises. All cloud stakeholders can yield tangible results with an approach that is feasible for software-defined clouds.

2.3. Validation Across Boundaries

The Spirent continuous cloud validation solution breaks barriers between cloud stakeholder interactions, cloud layer integrations and lab/live validations. Cloud stakeholders can achieve an unprecedented level of collaborative synergy with a common validation sandbox to deliver performant, reliable and secure clouds over time. DevOps and operators can take more onus to validate scenarios outside of the role-based silos being created with software-defined clouds, which is illustrated conceptually in Figure 4.

![Figure 4. Breaking barriers with validation across boundaries](image)
Spirent supports seamless emulation across virtual and physical boundaries to uncover subtle issues between cloud stack layers. Cloud layers can be isolated to pinpoint issues by orchestrating real/test infrastructure quickly to surround it.

Spirent solutions also span lab and live validation scenarios. DevOps staff can use Spirent’s Blitz solution to test services and applications running live on public clouds using pre-deployed cloud interfaces. Combinations of Spirent Blitz and Avalanche (Application/Web Service workload emulation) are being used today to validate application business logic by modelling complex transaction workflows.

### 2.4. Continuous Cloud Validation Use Cases

Figure 5 illustrates a use case that an infrastructure operator or platform cloud stack vendor might employ to characterize the performance of an infrastructure under test. Spirent continuous cloud validation can be used to orchestrate test emulation interfaces in a virtualized topology that load the infrastructure with realistic application workloads end-to-end using client/server emulation. Spirent can simultaneously orchestrate and emulate an SDN controller within the infrastructure under test to validate network virtualization switching functions. This entire use case can then be automated repeating it using various hypervisor or cloud stack platforms within Spirent VPTC solution to characterize performance deltas between variations.

![Figure 5. Infrastructure under Test Use-Case](image-url)
Figure 6 illustrates a typical use case for DevOps. They want to validate service and applications running on different real infrastructure profiles. The infrastructure profiles may be tethered to different hypervisors such as KVM or VMware’s ESXi in private cloud environments or might be a live scenario whereby public clouds are being assessed for future deployments. Real infrastructure with KVM and ESXi can be orchestrated and provisioned using Spirent’s VPTC solution for DevOps to run “what-if” design experiments. Performance characterizations, in either case, are made using Spirent Avalanche or Blitz client-side only emulation to directly load real applications or services. Blitz can be used for live cases on public clouds where lab setup is not possible. Avalanche is effective at generating Internet-scale loads within the VPTC solution or other lab environments. DevOps can continuously iterate on infrastructure profiles and application workloads to converge on optimal configurations, topologies, application partitions and infrastructures.

Figure 6. Service/Apps under Test Use-Case
3. Conclusion

Software-defined clouds inherently have an abundance of operating uncertainty due to the agility demands placed on them. Many cloud service providers will not adopt the technology without this uncertainty being tempered first. Taming the software-defined cloud will require the continuous cloud validation approach outlined in this paper with all key stakeholders using it during cloud lifecycle checkpoints. Spirent’s multi-dimension automation is critical in the strategy to be able to cover all the required permutations and key test scenarios across all the cloud stack layers. Freeware that tests in one dimension cannot yield meaningful results. The ability to orchestrate and validate iteratively will be required to cope with the growing multi-variable validation complexity software-defined clouds pose. Spirent’s comprehensive solutions enable validation across boundaries to break barriers between cloud stakeholders, cloud technology layers and lab/live scenarios.

The benefits of continuous cloud validation are many with some listed here to close:

- Unlocks productivity increasing testing efficiency for all stakeholders. Software-defined clouds can be validated without infinite cost and time requirements that usually lead to inaction.
- Vastly decreases capital and operating expenditures for cloud service providers. Providers attempting to migrate to commercial off-the-shelf hardware coupled with open-source can especially benefit. These technologies are rarely validated as a whole and suffer from reliability issues without the required testing. Providers can use continuous cloud validation to converge on optimal design choices and improve resource utilization avoiding typical over provisioning measures. DevOps and infrastructure operators can work together more synergistically with the ability to emulate flexibly to troubleshoot during cloud layer integration phases.
- Delivers services/applications quickly with performance, reliability and security. The software-defined cloud is designed to meet this goal. However, it can have the opposite effect if continuous cloud validation approaches aren’t used to automate away the uncertainty.