SNIA
Cloud Storage Initiative
www.snia-europe.org

The new data center is "software defined"

By Alex McDonald, Cloud Storage Initiative Chair, NetApp.

AT THE END OF 2015, the Distributed Management Task Force (DMTF) produced a whitepaper that defined the idea of a "software defined data center" or SDDC. It states:

"The Software Defined Data Center (SDDC) is an evolutionary result of virtualization and cloud computing technologies… [An SDDC is] a programmatic abstraction of logical compute, network, storage, and other resources, represented as software. These resources are dynamically discovered, provisioned, and configured based on workload requirements."

Which sounds well and fine – but what problems is an SDDC meant to fix, and why is enabling storage, compute and network to be managed by software so important? SDDC is another way of looking at storage, compute and network that are covered by, I'm afraid, yet more acronyms (or more precisely in this case, a family of related acronyms) that describe how we as end users wish to consume these technologies at scale. These are the SDx family, or "Software Defined [X]"; the XaaS family or "[X] as a Service"; and DAL, a Data-center Abstraction Layer.

Something as a service

To start, I'm going to look at two XaaS use cases covered in the DMTF paper; IaaS (Infrastructure) and SaaS (Software as a Service).

IaaS (Infrastructure as a Service)
A delivery model for IT infrastructure (compute, storage, and network) resources are provided as a service via network protocols.

SaaS (Software as a Service)
A delivery model whereby software applications are provided as a service via network protocols.

SDC, SDN and SDS (Software Defined Compute, Network and Storage)
SDC or server virtualization releases CPU and memory from the limitations of the underlying physical hardware. As a standard infrastructure technology, server virtualization is the basis of the SDDC, which extends the same principles to all infrastructure services. In a Software Defined Network (SDN), the network control plane is moved from the switch to the software running on a server. This improves programmability, efficiency, and extensibility. Software Defined Storage (SDS) dynamically leverages heterogeneity of pooled storage to respond to changing workload demands.

DAL (Data-centre Abstraction Layer)
An automation layer that interprets the requirements for the deployed software and configures the resources appropriately to meet those requirements

In IaaS, an end user chooses to execute a workload and uses the data center or service provider to host the infrastructure. After the infrastructure is available, the user installs the necessary software and content/data, then executes the workload. This is the most basic of the XaaS; there's no more than a bare set of resources. With SaaS there are different resources being provided; software & applications. Examples include databases, email and so on; and unlike IaaS, the underlying infrastructure generally isn't visible to the consumer. (There are other XaaS such as DaaS; data storage that is abstracted away from the underlying hardware. They all have similar definitions.) Each of these services must be managed; that is, we have to have not only a way to address the resources or software over the network, but also a need to be able to provision it, to monitor it and to account for it. SDDC comprises a set of features that include:

- The resources themselves; logical compute, network, storage, and other resources that are needed to provide services
- The ability to discover the capability of the resources; for instance, the type and performance of storage
- The automated provisioning of logical resources based on workload requirements; for example, provisioning of block based storage as required by a database application
- Measurement and management of resources consumed so that they can be billed (if not a chargeback, at least a "showback" of the costs of providing the service)
- And policy-driven orchestration of resources to meet service requirements of the workloads; services and their resource needs should be definable by generic templates rather than by requesting specific named resources.

Software defined storage

Let's take a closer look at the storage component. Software Defined Storage (SDS) is an emerging ecosystem of products that make visible to an orchestration system all the physical and virtual resources that make up a managed storage pool.

Is any of this new? Not really; in many

respects, SDS is more about packaging and how IT users think about and design data centers (whether they're private, public or hybrid). Storage has been "software defined" for a decade or more and the vast majority of storage features that users regularly employ have been designed and delivered as software components.

Because many storage offerings today have already been abstracted and virtualized, we know pretty well what SDS should include so that it can act as part of an SDDC:

- Automation – Simplified management that reduces the cost of maintaining the storage infrastructure.
- Standard Interfaces – APIs for the management, provisioning, and maintenance of storage devices and services.
- Virtualized Data Path – Block, File, and Object interfaces that support applications written to these interfaces.
- Scalability – Seamless ability to scale the storage infrastructure without disruption to availability or performance.

Key to this is the separation of the control plane (the process by which the storage is measured and managed) from the data plane (the process and path by which the storage is consumed).

SDS allows management of data by the storage itself, and without the need for intervention from storage administrators. There should be no explicit provisioning operations; data services should be deployed dynamically, and policies should be used to drive service level requirements and match the requirements with capabilities. SDS enables the publishing of storage service catalogs and enables resources to be provisioned on-demand and consumed according to policy.

With this separation, applications – the consumers of the storage – should be unaware of any of these control plane processes as they use it.

A cohesive whole; Towards an SDDC and the role of a DAL

We now have a set of services (the XaaS) that use SDC, SDN and SDS (respectively: software defined compute, storage and networking) as a cohesive whole; and that makes up our SDDC. This programmatic abstraction of logical compute, network, storage, and other resources represents the entire data center as a set of software interfaces.

So where does virtualization fit in? Does it play a part? Virtualization is central to the SDDC, but it's not sufficient to make a data center an SDDC. For that, we need a set of (preferably standard) APIs that make up something called a Data-center Abstraction Layer or DAL. The DAL is the total set of interfaces that sit over the top of the virtualized resources. Today, these requirements are dealt with by involving an administrator, who composes or orchestrates the resources largely manually. The automation brought by a DAL and its associated APIs allows us to interpret the requirements (based on profiles and policy) and automate what the administrator would otherwise need to do manually.

For instance, we might ask an administrator to provide a "Large Compute Intensive SQL Database" SaaS service. This might be defined in a template as a specified number of cores and memory, along with high speed Flash based storage and network to support a low latency environment; along with an instance of an SQL database. Traditionally, this would have required an administrator (or several) in implementing this service. An SDDC could provide a self-service portal to instantiate these instead, at much reduced cost and time.

SDDC solutions

As we've seen, and as the DMTF white paper points out, "[an SDDC] needs to have certain characteristics, such as multi-tenancy, rapid resource provisioning, elastic scaling, policy-driven resource management, shared infrastructure, instrumentation, and self-service, accounting, and auditing. This ultimately entails a programmable infrastructure that enables resources to be automatically catalogued, commissioned, decommissioned, repurposed, and repositioned."

It's a tall order to implement all that in a consistent fashion, but there are solutions available that are beginning to meet those requirements. it's a tall order to implement all that in a consistent fashion, but there are solutions available that are beginning to meet those requirements.

For instance, Amazon, Microsoft, HPE, IBM, Rackspace and others provide cloud services, mainly public or hybrid, that operate much like SDDCs; although full control is normally only available to the vendor of the solution and the APIs may be proprietary and incompatible. Some provide IaaS, others SaaS and more XaaS services.

> "(an SDDC) needs to have certain characteristics, such as multitenancy, rapid resource provisioning, elastic scaling, policy-driven resource management,shared infrastructure, instrumentation, self-service, accounting, and auditing."

There are also open source solutions such as OpenStack and Cloudstack; and there are a growing number of companies implementing private and hybrid solutions that are based on these technologies.

What will become increasingly important as the data center migrates from a set of manually administered and managed set of resources to a fully automated and software defined entity is the requirement for the APIs used to manage to be consistent and interoperable. The DMTF white paper covers existing and important standards for which there are already software products (both proprietary and open source), and identifies some of the areas not covered by standards that will need to be if we're to provide a full coverage SDDC.

For more information about SNIA, visit **www.snia.org.** For more information on SDDC, download the DMTF white paper.