Infrastructure-as-a-Service Builder's Guide

v1.0.2 – Q4 2009

For a discussion of considerations specific to the network, please download
Infrastructure-as-a-Service Builder's Guide
Network Edition: The Case for Network Virtualization
v1.0.4 - Q4 2010

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Cloud computing offers a truly revolutionary paradigm shift in the model for how IT services are consumed and managed. Much more than a set of technologies or even an outsourcing model, cloud computing provides a new way for IT to "do business." This whitepaper is focused on helping those who are cloud providers and want to build Infrastructure-as-a-Service (IaaS) clouds. It explains the opportunity at a high level and then proceeds to explain how IaaS clouds can be built. We provide a broad look at the way that an infrastructure cloud can be deployed in the public domain by service providers to generate revenue and also how these same clouds can be deployed inside a large enterprise to enable greater business agility.

CLOUD VISION: SELF-SERVICE

At Cloudscaling, we believe the cloud has one fundamental value proposition: *lowering the friction of IT services consumption to zero or nearly zero*. What this means is that the consumer of IT services is becoming increasingly empowered. This differs from historical IT service delivery models, while simultaneously being an evolution from them. Cloud computing becomes a process of enabling the consumer of the service on the terms that consumer dictates. This is what we call the "self-service" delivery model.

ABOUT CLOUDSCALING

Cloudscaling is the leading cloud computing consulting firm specializing in strategy, execution, and support services for enterprise and service provider clients interested in building clouds. The company was founded by pioneers of the cloud computing world with deep experience building some of the largest public and private clouds in service today.

Visit our website at cloudscaling.com, follow us on Twitter (@cloudscaling), email us at info@cloudscaling.com or call (877) 636-8589 to learn more.

The notion of self-service delivery encompasses more than simply "on-demand." The on-demand model says that a service can be consumed when the consumer of the service requests it. The self-service model encompasses more than the "when", it also encompasses the "how" and the "what." Self-service delivery models mean that the friction points in "how", "when", "what", and "why" are systematically removed or mitigated. These friction points differ depending on who the service consumer is.

For example, if the cloud consumer is a large enterprise business or government entity, a credit card only transaction is **not** a desirable way to consume services. This introduces more friction into the process of consuming a cloud service. On the other hand, for small and medium businesses, credit card transactions are ideal!

For all businesses, the ability to have metered billing based on usage with no long term contracts is important; however, for larger businesses, monitoring your usage, just like you would any infrastructure (e.g. electrical power usage), is also critical. Without both metering and visibility into the metering it is impossible to service yourself.

The self-service model is the new dominant paradigm for IT service delivery, epitomized by Amazon and Google, and is key to understanding how to be successful in building an infrastructure cloud. Put another way:

Cloud *is about removing friction for your customers*

INFRASTRUCTURE-AS-A-SERVICE (IAAS)

There are four fundamental components to the cloud economy: infrastructure, platforms, applications, and business process (Figure 1 to right). The infrastructure component is an ideal starting place for businesses looking to drive new revenue opportunities, whether by reselling or by enabling greater agility within the business.

Data infrastructure is moving towards a utility model, much like electricity, roads, and the telephone system. Can you build a self-service utility that empowers your customers? Will the ability to order servers, storage, and networking on-demand at any time with no limits create new opportunities or drive new revenue? For most, the answer is yes.

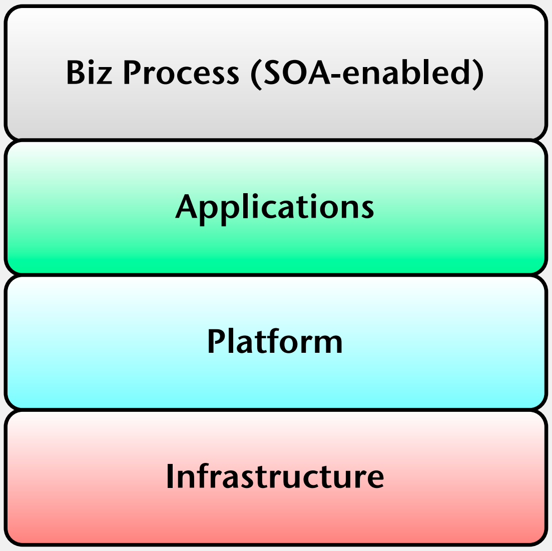

Figure 1. The cloud economy

In fact, on-demand self-service infrastructure is a key enabler of the other components of the cloud economy. A properly built, scalable, secure, and robust Infrastructure-as-a-Service (IaaS) system can be used to further deliver business value to the layer above: platforms and developers. There are many examples of how scalable platform and technology 'stacks' are delivered on-demand to allow rapid deployment of applications. Examples of this in the public domain include solutions like RightScale, EngineYard, Appistry, Gigaspaces, and SpringSource.

Platform or technology stacks delivered on top of an IaaS offering are capable of providing rapid provisioning of applications, which allows for greater agility and reduced time to market for software developers. For a public cloud, this means enabling technology startups and mid-tier businesses. For a private cloud, this means reducing wait times for internal development projects, increasing developer and product throughput.

Regardless of your motivation, IaaS is a foundational component to any well conceived cloud strategy.

CHALLENGES TO CLOUD ADOPTION

Whether a public or private cloud, there are many challenges to cloud adoption. We have heard anecdotally of empowered enterprise users who had to be trained on how to "self-service." For others, the challenges lay in differentiating and increasing adoption rates. When building a new infrastructure cloud, you should treat your endeavor as if it were a startup and consider the following carefully:

- Who are my cloud consumers?
- What are their use cases and how are they using this cloud?
- How do I reach them and enable them to self-service using my cloud?
- What is the best way to move quickly, show value, and gather momentum?

Make sure, as a cloud provider, you are clear about your objectives and your target cloud consumers before starting.

HOW TO DESIGN A CLOUD

Building an infrastructure cloud can be challenging at scale; however, there are many lessons to be learned from the early cloud pioneers, like the folks on the Cloudscaling team. The following sections provide a framework for discussing infrastructure clouds, examples of tradeoffs in architecture, and other areas of consideration.

Cloud Workloads

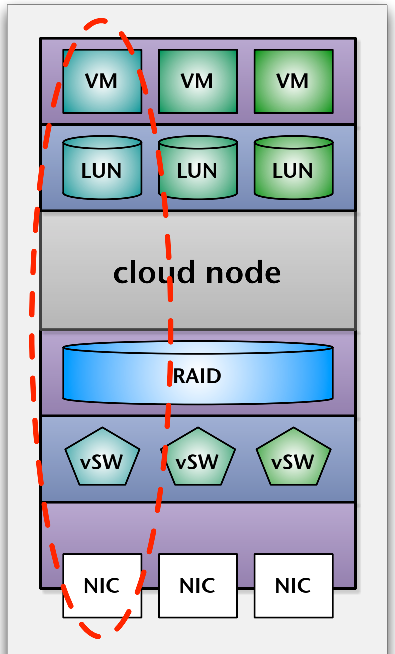

Figure 2. Virtualized workload

The fundamental building block of an infrastructure is a 'workload.'[1] Workloads can be thought of as the amount of work that a single server or 'application container' can provide given the amount of resources allocated to it. Those resources encompass processing (CPU & RAM), data (disk latency & throughput), and networking (latency & throughput). Frequently, but not always, cloud workloads are delivered in virtual servers. Figure 2 (right) shows how a single workload (circled in red) might be delivered using a single virtual server spanning a variety of physical resources including compute, storage, and networking. A workload is an application or part of an application. Examples of workloads include:

- Transactional Database
- Fileserver
- Application Server
- Web Server
- Batch Data Processing (e.g. running Monte Carlo simulations)

This means that a web application might have three distinct workload types: database, application business logic, and web serving. What you'll notice about these three workloads is that they have differing requirements in terms of computation, storage, and networking. A database may require large amounts of CPU & RAM, fast storage, and low latency networking, while an application server might require large amounts of CPU & RAM only. Web servers need very little resources other than networking.

When someone says "It depends on the workload" this is what they are referring to. Understanding workloads, designing your cloud for certain workload types, and the requirements those workloads may put on your underlying infrastructure is critical to success. This is why cloud providers must ask themselves: "Who is my customer and how can I make them successful?"

Infrastructure Cloud Architecture

Since cloud workloads map, generally one-to-one, to a physical or virtual server, creating a large-scale cloud becomes an exercise of putting these workloads together as efficiently as possible. Architectural decisions directly impact this efficiency. For example, Amazon's Elastic Compute Cloud (EC2) does not provide a network per customer. Instead, every server is on their own small (individual) network. This allows EC2 to get around an obvious scaling constraint, like the size of a single Ethernet network. Instead, for EC2, all server-to-server traffic is routed. The downside is that many kinds of network traffic (e.g. broadcast packets, multi-cast, and shared IP addresses) that require layer-2 networking are not possible[2]. In Amazon's case they decided to tradeoff the impact on network usage to gain greater scalability.

When considering your cloud architecture, understand that there will be limitations on the number of workloads you can put together based on the tradeoffs you choose. For example, if you chose to provide layer-2 networking, unlike Amazon (or in line with how a VMware-based cloud would look), you would soon discover that many of the protocols for layer-2 network-

[1] The notion of a 'workload' as applied to cloud computing is still in flux. There is not a commonly agreed definition. Later revisions of this paper will take this into account.

[2] Of course, to some degree you can use Amazon's new Virtual Private Cloud (VPC) to make your servers look like they are on the same network, but this is a recent development and not the default configuration on EC2.

ing were not designed with the cloud in mind. Using 802.1q VLAN tagging to isolate one customer from another limits you to 4096 networks. This limitation is inherent in the 802.1q specification. Although there are ways around this problem from the novel (e.g. Q-in-Q aka 'double-tagging') to the proprietary (e.g. Cisco VN-LINK, aka VNtag), they also have tradeoffs in complexity and vendor lock-in.

Understanding Pods & Cloud Control Systems

You will find many limitations similar to the networking example above throughout any cloud architecture. Large clouds have been working around these problems for a long time using a technique calling 'podding' or 'sharding.' Much like database sharding, the technique consists of understanding that there is a fundamental size limitation at some point and says: "let's just create a group of machines at the maximum size, then create copies of that group and spread the load across them." This is similar to load-balancing in spirit, but is not exactly the same. Usually, a customer or application can only fit in one pod[3].

For example, I'm sure you have seen times where a large cloud provider such as Gmail had an outage and only some users are affected. That might mean that users with the last names starting with A and B were in the pod that was affected. Of course, Google may not distribute users between pods using such a crude mechanism as last name, but the example holds true. Every pod is a complete version of Gmail, but only serves a subset of the total Gmail users. A distribution system knows when you login to place you on the pod you were assigned.

In practice, you want the largest size of pods possible and you want to be able to aggregate your pods into a larger pool. For an infrastructure cloud, this aggregation is done at the level of the cloud control system (CCS). The CCS allows you to manage a large number of pods. Figure 3 illustrates this (below).

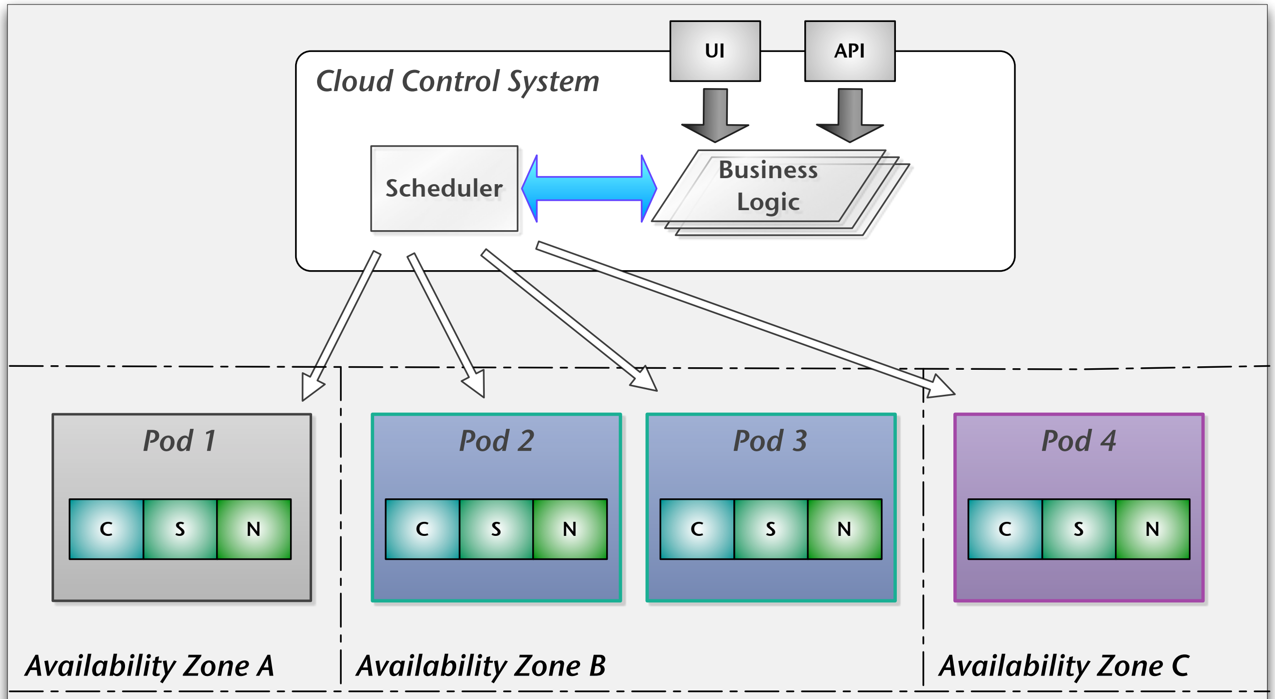

Figure 3. Infrastructure cloud using pod architecture

[3] This is a simplification; some advanced techniques will allow spanning multiple pods.

Pod Sizing

Just like Amazon or Google, you need to aggregate as many workloads as possible in each pod. Google, the leader in datacenter efficiency, builds a pod for 10,000 physical servers[4]. You won't need this scale. Many or most of your customer workloads can be virtualized. At a ratio of 30 virtual servers per physical server (30:1), Google could support 300,000 virtual servers per pod![5] This far exceeds the requirements of a majority of applications being developed today. In addition, Moore's Law and technology advances mean that ratios of 60:1 or even 100:1 are likely in the future.

What's a good size to plan for? The answer is that it varies. If you are building a public cloud, the size may be driven by a variety of factors including oversubscription models, technology partners, and network limitations. For private clouds, it will be driven by support for VMware, bare metal clouds, and even open source architecture limitations.

The best answer is to prototype and test your initial pods to determine their scalability & capacity. We find many of our clients use us to help them make a first best guess.

Aggregating Workloads & Pods

Once you have a cloud control system (CCS) and pods, you'll want to aggregate and deploy them to create your infrastructure cloud. Today, everyone follows a model that Amazon set with their Elastic Compute Cloud (EC2): *Availability Zones* (Figure 3 above). The availability zone model is a natural way to isolate one set of pods from another. It is implied, or even written contractually, that each availability zone has isolation and hence redundancy in network, power, and facilities. If power is lost in a given availability zone, a customer's servers in another availability zone are not impacted. Customers are then encouraged to design their mission critical applications to run in multiple availability zones. Non-mission critical apps can run in just one. The following diagram (Figure 4) shows how workloads, pods, and availability zones fit together.

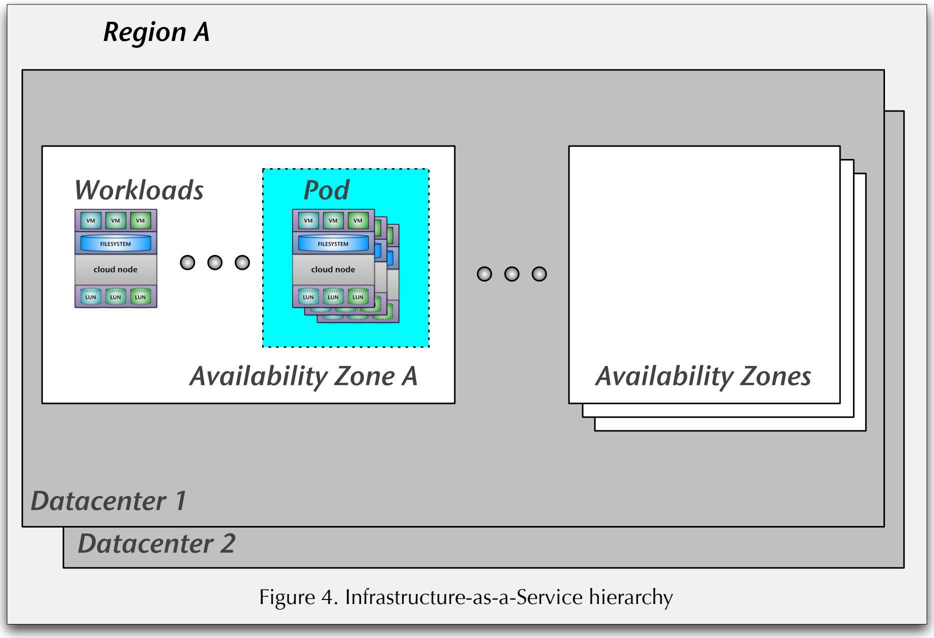

Figure 4. Infrastructure-as-a-Service hierarchy

[4] This constraint is created by their 10,000 port homegrown Ethernet switches. GOOG would like to achieve pods of 100,000 servers and is actively working towards this objective.

[5] Of course, Google does not currently use virtualization in their infrastructure today.

Each availability zone resides in a single datacenter facility, although it is theoretically possible to have multiple availability zones in a single datacenter, provided there is isolation in power and networking. The key is that each availability zone should be relatively independent. Datacenters are then aggregated into a region and regions further into your global cloud.

Pod Architectures

In-depth guidance on pod design is outside the scope of this document. There are many architectural decisions to be made, each of which has its own tradeoffs. As a guideline, design your pods based on workload needs, business requirements, and scale desirability. Find a partner with a strong track record to assist you if this is your first time. There are many dimensions to consider, but let's pick one to illustrate further in this section such as your storage architecture.

There are a many ways to build a storage architecture for virtualization, but we will constrain the discussion to two options: Direct-Attached Storage (DAS) and Storage Area Networks (SAN). Network-Attached Storage (NAS), like NFS, would also be a fine choice for your storage architecture, but is left out of this paper for brevity's sake[6].

Another reason to limit our selves to DAS and SAN is that they are the dominant storage architectures in most public clouds. The first public clouds such as Amazon's EC2, Rackspace, and GoGrid use a combination of the open source Xen hypervisor along with DAS. The latest entrants, such as Savvis, Terremark, and AT&T's Synaptic Services use the VMware hypervisor along with SAN. Their choices are driven largely by cost and architecture. In the case of the early entrants, they positioned their clouds as consumer clouds using Xen and DAS for their lower price point ('free' and cheap, respectively). The latest public clouds chose to position themselves as 'business' or enterprise clouds using the VMware ESX hypervisor. VMware's recommended deployment model uses centralized SAN storage, which allows for a number of features, such as live migration (aka 'vMotion' in VMware parlance) where you can move a virtual server from one physical server to another. The enterprise-class cloud choices are strategic in nature. They hope that businesses will pay a premium for 'advanced' features or brand names.

Regardless, a Xen pod can be built with SAN and a VMware pod can be built with DAS. The choices made by the current crop of providers are reflective of where they sourced their architectural models: Amazon EC2 or VMware's best practices.

Let's take a closer look.

[6] We actually have a number of cloud deployments on NFS in both lab and production environments and it has been quite robust, particularly when using the right NAS appliance.

Pod Architecture (DAS)

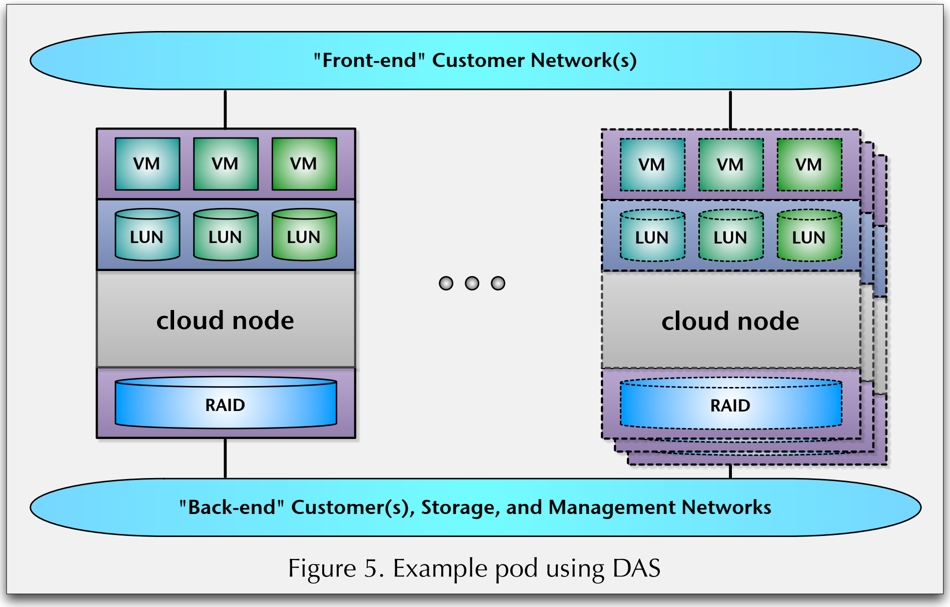
"Front-end" Customer Network(s)
VM
VM
VM
LUN
LUN
LUN
cloud node
RAID
"Back-end" Customer(s), Storage, and Management Networks

Figure 5. Example pod using DAS

The DAS model dictates that every physical server ("cloud node") will have its own local storage system (Figure 5 to right). This means that from a storage perspective a pod can be quite large as each node added to the pod also adds storage capacity.

As a downside, the DAS model also means that since there is no common storage system across all servers some features like live migration are extremely difficult or impossible to implement.

DAS forces your cloud operations or IT team into managing a large amount of decentralized and distributed storage. This can be a challenge at scale, which is part of why centralized Storage Area Networks (SAN) became quite popular. Imagine that every node has 8 disk drives and you have 1,000 nodes. That is 8,000 disk drives spread over 60 racks. Each node may also have a local RAID controller. Your team will have to tightly manage the firmware of disk drives and RAID controllers both in order to reduce hard to diagnose failure conditions. Guaranteeing homogeneity of firmware and chip versions across so many servers is difficult. This can be a significant management challenge if not planned for.

Pod Architecture (SAN)

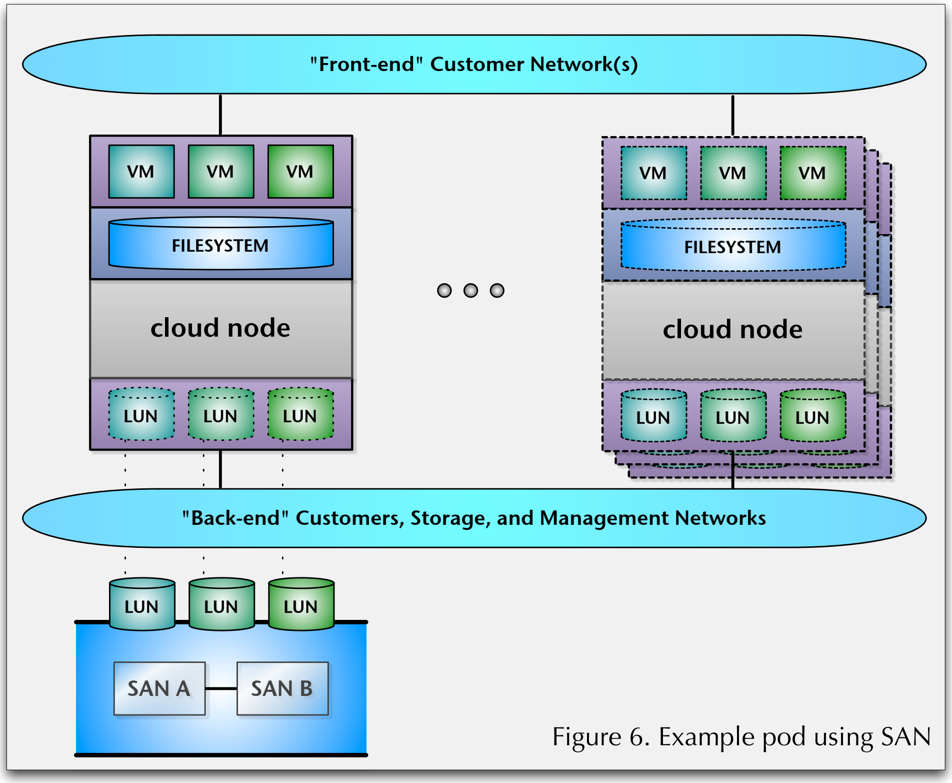
"Front-end" Customer Network(s)
VM
VM
VM
FILESYSTEM
cloud node
LUN
LUN
LUN
"Back-end" Customers, Storage, and Management Networks
SAN A
SAN B

Figure 6. Example pod using SAN

In contrast to DAS, SAN embraces centralization, which brings its own positives and negatives. On the positive side, live migration and similar technologies are now possible, lowering the operational overhead associated with running a large scale cloud. Other capabilities like backups and high availability in the case of a node's failure are also quite easy.

The negative is that pods must be much smaller. The reason is that any given SAN will have some kind of scaling limitation. It can only be so big and can only serve so many cloud nodes. You **could** deploy more than one SAN per pod, but you then break the pod architectural model. Obviously, the larger a SAN can become, the greater the expense. Be prepared to size your pod according to the size of the largest SAN you are willing to purchase. This will also likely determine the performance characteristics of your pod and its VMs.

For example, if you design a pod that is for high transaction web applications, you may be willing to purchase a very large and expensive SAN, which will allow the pod to be much bigger. On the other hand, for a pod designed for dev and test usage, a smaller, inexpensive SAN would do.

Pods as Building Blocks

Remember, with the proper cloud control system (CCS) it is possible to design your cloud to support multiple pod types created for different workloads. Examples include:

- High performance web applications
- Low cost, mid-tier performance applications
- High performance GPGPU computing on bare metal

Design your pods and cloud according to customer requirements and you can't go wrong.

PLANNING YOUR CLOUD

Given the number of different dimensions of consideration when planning your cloud, you should have a quick checklist you can use when planning.

Following are two sections, for technology and business, and both private and public clouds that give a rough cut of areas you should particularly focus on.

Technology Concerns

- Storage scalability: How much storage per pod?
- Storage performance: IOPS vs. raw throughput
- Network scalability: Can I span a network across pods?
- Network performance: 10GE vs. 1GE
- Network architecture limitations (e.g. L2 vs. L3)
- Oversubscription rates & capacity planning
- What kind of hardware will I use? (e.g. Does hyper-threading impact oversubscription?)
- Is there a happy medium between DAS and SAN like NAS or Distributed Filesystems (DFS)?

Business Concerns

- Cost-basis requirements (do you need a certain margin?)
- Licensing issues (e.g. Hyper-V & Windows licensing model)
- Can I grow my cloud organically through revenues or customer adoption?

CLOUD DEPLOYMENT & MANAGEMENT

Once designed, deploying and managing your Infrastructure-as-a-Service (IaaS) cloud becomes your next primary goal. Like any other large scale system, there are several key concerns, including:

- Integration & Extensibility
- Monitoring & Capacity Management
- Datacenter & Resource Optimization

It is beyond the scope of this document to address these items in full, but let's briefly touch on each.

INTEGRATION & EXTENSIBILITY

A key feature of your Cloud Control System (CCS) should be its ability to integrate with external systems. The extensibility of your cloud dictates your ability to use third party tools of your choice for authentication, monitoring, and legacy applications. The best CCS software should allow you to integrate legacy systems and software with relative ease.

MONITORING & CAPACITY MANAGEMENT

It is your choice whether you will provide monitoring for customer servers and services; however, you will certainly need to monitor your cloud nodes, network gear, and control systems. Monitoring comes in three basic flavors:

- Reactive
- Proactive
- Forecasting

Reactive monitoring is 'break/fix' monitoring where systems and nodes are monitored for availability and when one fails, someone is notified to rectify the situation. Proactive monitoring is the art of collecting metrics data and surfacing it as historical trend data such as RRD graphs and similar. Forecasting is where business analysts shine, using proactive metric data to perform capacity planning.

Capacity planning is underrated as a science. For public clouds, it will be a critical business capability to optimize your capital usage.

DATACENTER & RESOURCE OPTIMIZATION

Directly related to capacity planning is datacenter and resource optimization. Whether a public or private cloud, a strategy for designing your cloud would involve designing your pods for various workloads, such as high performance I/O on bare metal, production web applications, and test applications for functional (not performance) testing. Robust scheduling systems that are part of a Cloud Control System will allow for optimizing placement of your workloads on different pods & servers. The following diagram (Figure 7) shows an example.

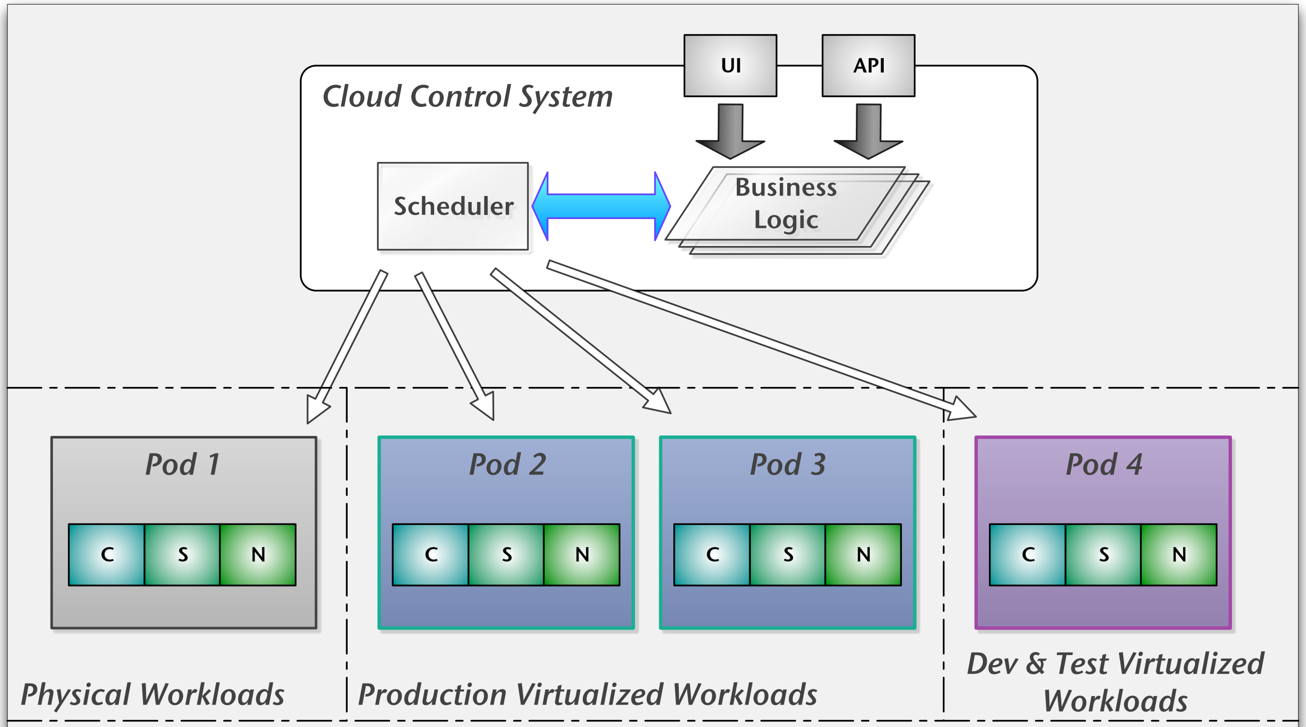

Figure 7. Using a CCS to bridge different pod workloads

Here we can see three different kinds of pods designed for three distinct workloads. A strong cloud scheduler will allow for this model and also help you to optimize resource usage across the various pods and servers. Combined with a good capacity planning discipline it's possible to drive extremely high utilization rates on either a public or private cloud.

INFRASTRUCTURE-AS-A-SERVICE ADOPTION STRATEGIES

This section is specific to internal private cloud deployments at enterprises, although it may have some applicability to public clouds that were historically hosting providers. Let's talk briefly about what it means to have a clear IaaS adoption strategy inside your business.

Because cloud computing is a fundamental paradigm shift in how IT services are delivered, it will cause significant disruption inside of IT organizations. Embracing this change will be key to success. While the final impacts are hard to measure, it's clear that a self-service model is widely expected by IT consumers and can turn both internal (IT) and external cloud providers into strong business partners. Cloudscaling offers a simple recipe to encourage success internally.

Here are our DOs:

1. Find an internal customer who can act as your champion with a strong use case for a trial
2. Standup a simple IaaS trial using open source, a consulting partner, or commercial software
3. Show the value of self-service for your internal customer so the business can measure success
4. Work with your customer to measure the ROI based on cost of the cloud service delivery combined with the business value their application generates
5. Reuse internal equipment and resources if at all possible

6. Plan to expand the ‘trial’ organically after showing initial success; that means pick your initial CCS carefully
7. Find a CCS vendor who will work with you

Here are our DON’Ts:

1. Don’t build something complex for the trial
2. Avoid spending large amounts of capital expenses on the trial initially; go small
3. Don’t get hung up on specific vendors or solutions; that’s not what’s important

Remember, you are proving you can drive top line revenue through increasing agility and enabling internal customers. Speed and delivery are of the essence in proving that you can help your business partners and deliver true self-service.

We would summarize your basic adoption strategy as: *start small, get quick wins, show success, iterate, and expand*. Obviously, although we are biased, it is highly recommended that you find a trusted neutral third partner, a consulting business or otherwise, to help you get your initial infrastructure cloud up quickly and with as little risk as possible.

CLOUDSCALING RECOMMENDATIONS

We are a vendor agnostic cloud consultancy; however, we do evaluate various vendors for our clients and have very good experience with some of them. Following is a list of vendors and their software (in alphabetical order) that we have successfully deployed with a short description including strengths and weaknesses. Please note that for the purposes of brevity we only include software suitable for a cloud control system (CCS), although this may change in the future. Recommendations for storage, networking, and compute are provided separately.

If a particular vendor is not in this list, we simply have not had the time or resources to evaluate their technology. We will update this list over time and, of course, can provide a set of tailored recommendations if you choose to engage us to help you with your cloud strategy and architecture.

CCS Options

DynamicOps Virtual Resource Manager (VRM)

One of our team members was responsible for building the DynamicOps product (VRM) initially, so we have very good visibility into its capabilities. It is probably the most mature IaaS CCS out there today; however, it does not support bare metal clouds and it is backed by a startup, so you’ll have to decide if either of those are a major concern.

Also, depending on how your preferences, VRM runs on Windows, which may make integration easier or harder for your team.

One major upside of the VRM product is that it was designed in a large enterprise for their particular needs and has strong resource and user management features. You can dictate what kinds of resources a given user or department can access without authorization and combined with an approval workflow for additional resources. There are a number of related convenience features similar to this one that make managing a large cloud consumer population much easier.

OpenNebula (ONE)

The only open source product listed here, the Open Nebula Engine (ONE) has proven to be a very well designed CCS. There are several downsides however. There are few commercial support options (Cloudscaling and the Universidad Complutense de Madrid) and there is no built in user interface (UI). It was also written in a foreign country, so government agencies who are sensitive to such things may be concerned. Of course, it is 100% open source, which means the code is open for inspection at any time.

On the plus side, ONE is free to download and experiment with. It is also well written and can be easily extended.

Some thought should be given to using ONE as a CCS for a single pod and aggregating the control system through another CCS (e.g. Platform ISF). This allow you to create a VMware Virtual Center equivalent for open source pods.

Platform ISF

Platform Computing has a long history in grid and high-performance computing (HPC). They have an impressive enterprise client list and a strong track record in this space that they bring to cloud computing. Platform's advanced scheduling capabilities may make it desirable for businesses trying to extract maximum efficiency out of their clouds. It is also the only CCS with support for bare metal clouds. This makes it ideal for environments that require different kinds of workloads, including those that should not be virtualized. Because it was designed for a multi-workload pod environment, it will easily bridge VMware vSphere, Citrix XenServer, bare metal, and open source pods.

A major downside of Platform in the near term is a lack of API support in the GA product, although we understand that it is forthcoming in the next point release. As cloud engineers, we are actively providing input to the Platform team.

VMware Lab Manager

VMware's vCloud product, code-named 'Redwood', is not yet released. As the market leader in virtualization, it's not surprising that they have technologies in the marketplace that are similar to a cloud control system (CCS). One such product is VMware's Lab Manager, which could be used with VMware's vSphere to create a multi-tenant on-demand cloud. Unfortunately, you will have to roll your own API and other features to make it feature complete.

Regardless of whether you use Lab Manager, VMware is unlikely to design a CCS that manages bare metal clouds. If you have that need or desire a heterogeneous environment we don't recommend it.

SUMMARY

Building an infrastructure-as-a-service (IaaS) cloud will be a fundamental piece of any comprehensive cloud strategy, whether you are building for profit or 'fun'. There are a number of important considerations you must make when designing your architecture and planning to grow into a larger scale deployment. *You must remember that your job is to empower the consumer of your cloud by enabling them to 'self-service.'* Allowing self-service, you increase agility and reduce friction in how others build value on top of your cloud. Focus on the big picture objectives and lean on trusted partners to help you with strategy and tactical decisions. Measure success by how well you enabled your cloud consumers to self-service.

CLOUDSCALING WORKSHOPS

We provide detailed one and two day workshops for both executives and technologists who want more detail on either the cloud marketplace or cloud architectures. Please contact as at info@cloudscaling.com.