SCTE·ISBE CABLE-TEC EXPO 2018 ATLANTA, GA OCTOBER 22-25

2018 Fall Technical Forum SCTE·ISBE • NCTA • CABLELABS

Running a Multi-Tenant Hybrid Cloud for Large Scale Cable Applications

A Technical Paper prepared for SCTE•ISBE by

Neill A. Kipp
Distinguished Engineer and Cloud Software Architect
Comcast
1401 Wynkoop Street, Suite 300
Denver, CO 80202
m: 720.530.6917
Neill_Kipp@cable.comcast.com

Table of Contents

List of Figures

Abstract

The cloud—with its automation and virtualization of compute, network, and storage—has fundamentally changed the way software engineers design, develop, and deploy applications. Instead of by-hand configuration, software programs deploy virtual routers, firewalls, databases, and application servers. Software causes application servers to respond to load changes, change network topology, and scale deployments accordingly. As a result, applications can be effectively tested while they are running in production!

Before public cloud, cable companies invested in data centers and managed their own compute, storage, networks, and applications as their own private cloud. The public cloud alternative offers a compelling agility but often comes at an increased price. For the best of both worlds, cable companies can implement a hybrid cloud solution that leverages the existing capital investment in their private cloud infrastructure alongside the increased agility of the public cloud.

Running a hybrid cloud is challenging. The hybrid cloud must manage multiple tenants, each represented by multiple users. Users must be able to request cloud resources for their tenants in any private cloud region and from multiple public cloud vendors. All cloud deployments must secure video media, customer data, and application services.

The Comcast cloud team has built a hybrid cloud for large-scale cable applications. We publish hybrid cloud architectures and tools to help product owners upgrade their applications to be virtualized, containerized, and orchestrated. We provide user permissions, a network security framework, and automated controls that provide guardrails to streamline software development and minimize risks of security breaches. Our hybrid private cloud is running in eight regions and uses three public cloud providers. It hosts hundreds of tenants, thousands of users, and the software it hosts serves tens of millions of customers.

1. Dream of the Hybrid Cloud

Years ago, when server closets overflowed with gear, cable companies expanded into data centers to deploy their computing resources for television delivery and software applications. While a “data center” typically provides racks of servers, connectivity, power, cooling, and technical staff, a “cloud” extends these services to provide virtual servers, networks, and storage. It also includes software that lets application teams configure their own deployments.

Demand for computing has increased such that these on-premises data centers with virtualization (“private clouds”) are usually operating at maximum capacity. As an alternative to improving compute facilities in the private cloud with additional capital expenditures, companies are increasingly deploying their expanded operations into seemingly unbounded “public clouds” including those offered by Amazon Web Services, Microsoft Azure, and Google Cloud.

Private clouds retain certain perceived advantages, including physical security, operational control, capitalized infrastructure, and simple familiarity. As such, full migration to the public cloud may never be warranted. As a result, the natural evolution from private closet to public cloud has left the largest companies with an amalgamation of software deployments spread across networks, firewalls, hand-racked servers, and virtualized instances. Private infrastructure runs in well-known locations, is operated by esteemed colleagues, and the story of each data center contributes to a collective corporate history. Public infrastructure runs in undisclosed, regional locations (with nonspecific names like “US East”).

For the foreseeable future, it makes sense that companies will operate "hybrid cloud" environments—transient combinations of private and public virtualized services. Widely distributed, heterogeneous infrastructure will support a broad range of custom interconnected applications. Toward that mission, cloud operators are seeking to normalize the cloud technology substrate and realize their vision for the cloud: a homogeneous, distributed, secure, and fault-tolerant computing infrastructure, networking, and storage "fabric" for all application deployment needs (Figure 1).

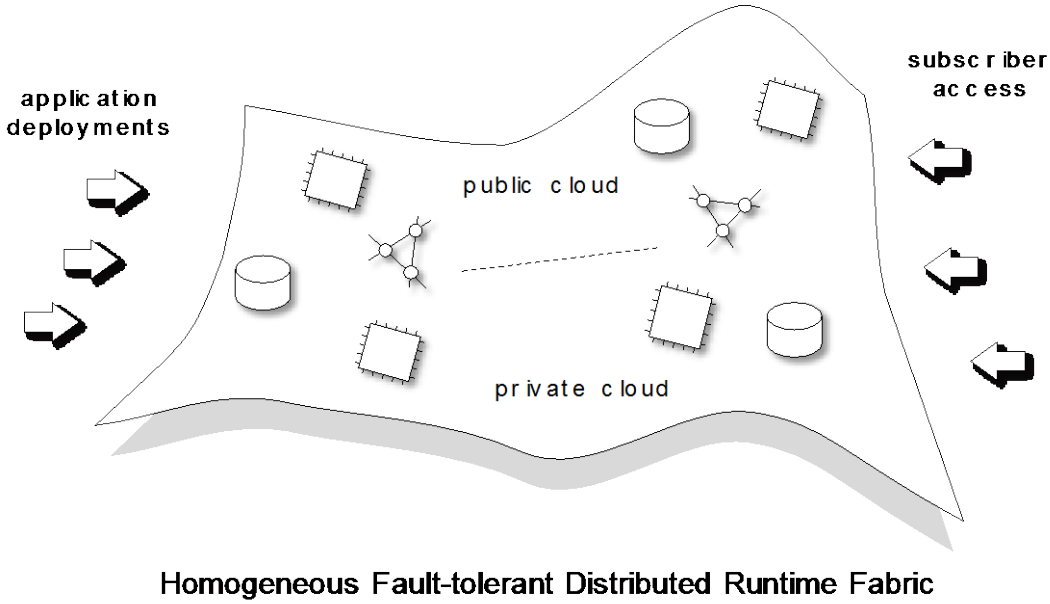

Figure 1 - The idealized "hybrid cloud" provides a runtime fabric of computing infrastructure, networking and storage.

2. Data Center Snowflakes

The private cloud, a key component of the hybrid cloud, comes with its own challenges. Data centers are not homogeneous; in practice, they are as unique as snowflakes. Application teams must manage their own resiliency and high availability. Quarters are usually cramped.

Comcast has deployed servers, network, and storage to more than 50 data centers. Historically, deployment teams reserve space in a specific data center, then purchase and ship servers, storage, and networking equipment to be racked there. Teams took latitude in their purchasing choices, and negotiated with individual operators in each data center to resolve specific conflicts with the local autonomy. As a result, each data center grew into its own operating "silo" of unique heterogeneous sprawl. This diversity has become challenging to operate, as the corporation must hire and retain operational expertise for each vendor and model of networking equipment, storage unit, and compute node.

A typical hardware deployment in a single facility requires months of lead time and an extensive approval process. The team orders hardware to be shipped and works with the operators to get the hardware racked, powered, and networked. The operators install the requested operating systems and provide specific network addresses per server. Before automation such as PXE boot and Foreman, servers were hand-configured by typing in values from spreadsheets. When the data center operations team completed its phase of work, application teams were left to verify and debug each hand-typed deployment. Inflexibly

numbered servers, by convention, became the “pets” of application teams, each server having a personality of its own.

Growth in these data centers sometimes required servers to be re-racked in rows close together so that network configuration could be normalized. Otherwise, routing became more complex and therefore subject to accidental misconfiguration. For example, an application deployment in the first year might require only one rack, but due to increased usage, may need additional racks in year two to satisfy demand. The operations team would choose between re-planning the footprint or reconfiguring its network routers. Sprawling router configuration is its own problem, where manual mistakes lead to long down times.

While some data centers are owned (and therefore operated) by the corporation, others are made available by vendors, and provide “remote hands” to do physical installations, maintenance, and upgrades. Service-level agreements vary by facility, and therefore increase the “snowflake” problem.

Software deployment automation similarly evolved. In the earliest days, an operator had to insert an installation disk into each node. On-site manual operations were quickly replaced by remote deployments such as secure shell (SSH) and secure copy (SCP), then automated by deployment tools such as Puppet or Chef. Even with automation, these agent-based deployments too often get “stuffed up” and fail to reach the desired state. Thus, the versions of deployed software in the tier of application servers could diverge and, as a result, the application service would be degraded.

Enter virtualization. With virtualization, each physical host server is configured with “hypervisor” hosting technology that allows multiple “virtual machine” guests to be deployed atop. A centralized console lets teams manage their virtual machines, networking, and storage attachments for each data center. Virtualization provides a clean division between hardware operations and software applications.

Ideally, virtualization consoles would span data center installations, and application teams could rely on a single console, command line, and programming interface to manage their national deployments. Virtualization consoles have only recently provided the feature to coalesce and manage multiple deployments as a homogeneous private cloud.

Even the virtualization technology substrate has diverged. Today, the Comcast private cloud supports virtualization technologies that include VMWare and OpenStack. Application teams must therefore pick a substrate technology or learn the console and programming interface for each virtualization system on which they wish to deploy.

One of the goals of application deployment is high availability. Single points of contact (SPOC) directly imply single points of failure (SPOF). Applications, servers, networking, power, air conditioning, must each be redundant. Multiple fiber optic connections must be provided for each rack, row, and data center—when a backhoe cuts one fiber connection, the redundant fiber must suffice.

For reliability, storage must also be redundant. Initially, redundant arrays of independent disks (RAID) technology provided this redundancy. Unfortunately, a rack, row, or whole data center might be partitioned from the rest of the network, rendering the entirety of compute and storage there unavailable. In today’s private cloud, individual application engineers must have the additional expertise to design for availability and reliability. Cloud storage is increasingly realized by “just a bunch of disks” (JBOD) and employ software that implements robust storage.

Ideally, cloud infrastructure configuration and subsequent software application deployments are developed “as code.” Infrastructure configuration tools such as Terraform can be launched using files that

are stored in familiar source code control systems such as Git. Applications can then be pushed to servers using tools such as Ansible. “Configuration as code” significantly reduces the complexity faced by application teams when deploying software, and as such has become a software engineering “best practice.”

Private clouds also suffer from being visibly finite and ultimately cramped.

If an application team was prescient enough to provision for growth, a majority of its physical resources would remain underutilized until demand met supply. If not, growth is crippled by delays in purchase, delivery, rack, and configuration.

With virtualization, teams are selfishly motivated to secure quota above current usage. Unmanaged “land grabs” constrains the ability for new teams to obtain resources. Worse yet, to accommodate requests, private cloud operators can allow over-commitment of virtualized resources (as much as 8:1 virtual CPU to CPU!) in the hopes that actual utilization will not simultaneously peak on each hypervisor. Unfortunately for cable companies, actual loads peak at the same times every day, every week, and during specific events like championship sports. When a virtual machine instance hogs the compute or network resources for the whole hypervisor, other instances must suffer the “noisy neighbor” problem. On the other hand, strict quota enforcement leads to stranded compute and underutilized bandwidth (Figure 2).

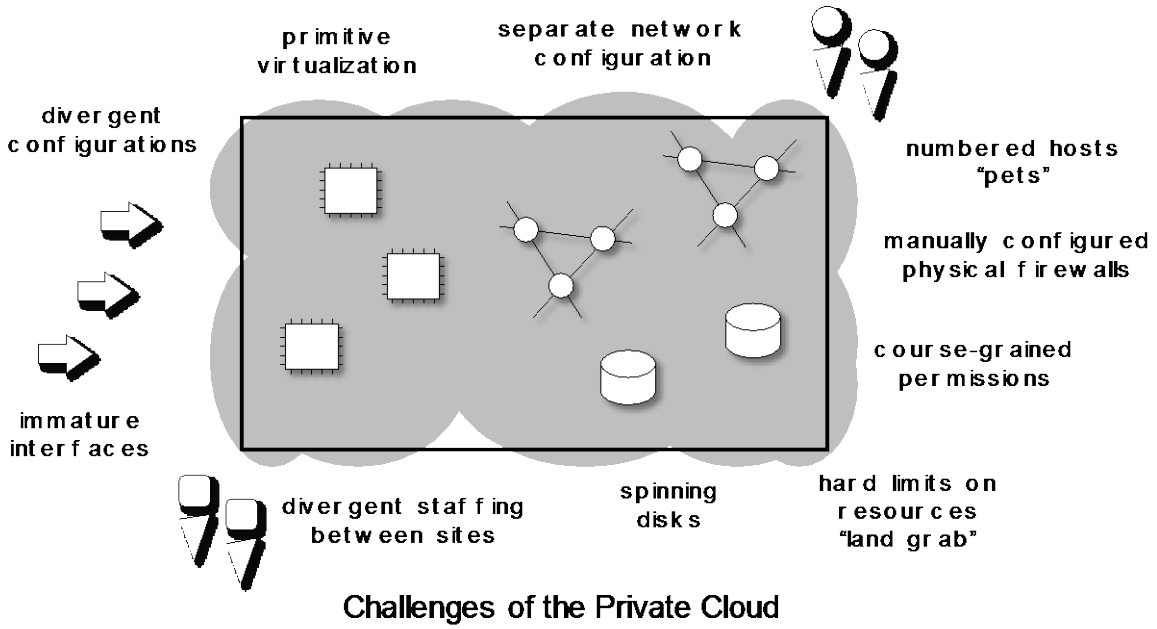

Figure 2 - Private clouds come with a diversity of management issues.

3. Use Case: Single Tenant Private Cloud

Some applications at Comcast are central to the business and continually serve nearly every subscriber, including IP Video on Demand (VOD), IP Linear Channel delivery, and Content Delivery Network (CDN). VOD, linear, and CDN each represent a single “tenant” in the private cloud, and because of their demand volume, each runs on dedicated infrastructure.

The IP VOD system runs on dedicated servers hosted in national data centers. End to end, the VOD library holds seventeen years of video content, with titles rotating continually. Network-attached video

storage is in the petabyte range. VOD server software for packaging and encryption run on virtual instances and are automatically deployed with Puppet.

Similarly, 13,000 linear channels are served from a combination of dedicated servers in national data centers and regional installations. The content delivery network protects these servers, and thus allows the software that packages and originates linear channels to run at a steady resource state.

Servers that implement our two tiers of content delivery network (CDN) run on hardware configured specifically for network and storage performance. For compute efficiency, CDN caching runs without a layer of virtualization. Comcast has deployed the second largest CDN in the country, delivering terabytes daily. We support CDN clusters in more than 50 data centers nationwide.

These full-stack deployments require dedicated operations teams to provision and maintain these mission-critical applications. The hybrid cloud must continue to support large, single-tenant deployments such as IP Linear Video both for Comcast and syndicated to other providers in the US and Canada (Figure 3).

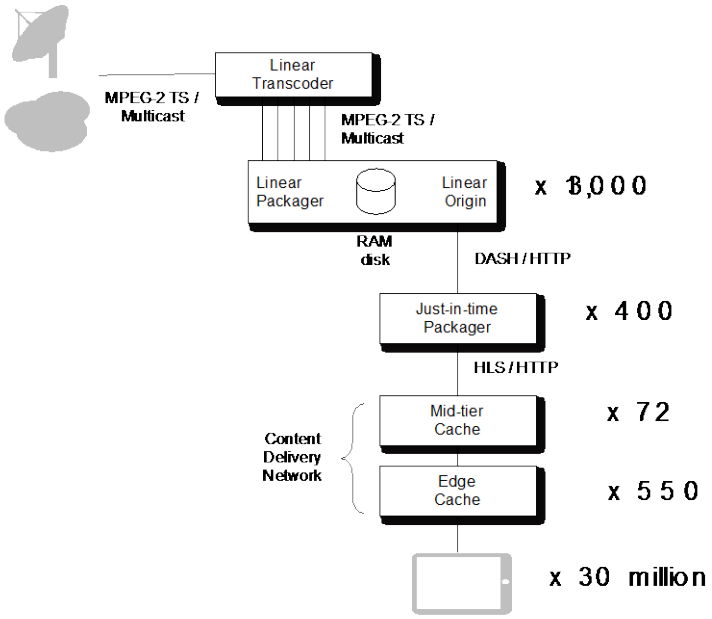

Figure 3 - The IP linear application stack deploys thousands of components into the private cloud.

4. Lure of the Public Cloud

Often cable applications teams can code and deploy applications in less time than it takes to purchase and rack the servers these applications would be hosted on. Procurement need not risk private cloud augmentation for applications that might not be embraced by the market. Developers can approach a homogeneous infrastructure that closely resembles the cloud vision of a generic compute fabric. These statements are true, when using a public cloud.

In the public cloud, all resource reservation and allocation is done using consoles and automation interfaces. Each public cloud is a homogeneous entity spread across multiple regions. Within each region is a set of “availability zones” that provide a singular failure domain. Within each availability zone,

application teams are unable to learn exactly where their virtual machines are deployed, and if restarted, if they have been migrated. Even virtual server internal network addresses should be expected to change.

In the public cloud, everything is "software defined" and all configuration is stored "as code." Thus, configuration can be automated by Terraform or similar proprietary public cloud software. Deployments that took months now can take hours and upgrades can happen in minutes.

Software defined compute means that configuration files ensure that enough virtual machines are running at all times; should one fail, it will be recycled and another will be online in minutes. Advanced deployments can combine load monitors with compute configurations—the public cloud can add virtual machines when loads spike upward and remove them as load reduces.

Software defined networking (SDN) means that routing is accomplished as follows: Software configures the domain name system (DNS), global server load balancing (GSLB), high availability load balancing (LB), virtual private clouds (VPCs), and peering between application stacks. And once a hardware router is installed between public and private clouds, software configures routes between public and private networks. Software configures intrusion detection systems and distributed denial of service mitigation systems.

Teams from many large companies, including cable companies, have begun to consider the public cloud, overcoming this often-repeated sentiment: "Our proprietary data should be stored within our premises." With that consideration, significant security challenges must be overcome, as follows:

"Our proprietary data should be stored within our premises."

Networks must be secured. Similar to the risks with storage, access to servers within the public cloud must be secured and controlled. Inadvertently making firewall changes such that protected servers become public can lead to disastrous consequences.

Data must be secured. Storage "buckets" in the public cloud can be converted from protected to public with one click. While malicious intent is possible, much more likely is that someone on the application team inadvertently assigned incorrect configurations. Furthermore, data stored in buckets should be encrypted at rest, and the keys maintained separately.

Permissions must be managed. As a key part of access control, companies should apply a blanket DENY permission to any resource or permissions changes (especially global ones!) and carefully add ALLOW permissions for specific team members or services assuming specific roles to make specific changes to specific resources. This implements the "principle of least privilege," where access is provided only to users (or pseudo-users) at the level of that entity's legitimate purpose in the larger system. A dangerous shortcut in the public cloud skirts researching and applying specific permissions for specific resource types, instead giving blanket authority to "do anything, just get it done." Furthermore, permissions should be managed "as code" as well, to ensure each account complies with human and automated audits.

Access must be controlled. Unless further restricted, permission to modify any resource within a cloud account is by username and password. Because teams share the account, it is far too easy to share the passwords to it. Advanced access protection includes the following:

- Requiring multi-factor access (MFA) on each login mitigates password sharing.
- Binding cloud account access to a corporate single sign-on (SSO) expires users when they change teams or leave the corporation.
- Monitoring and alerting on access, including times and locations of login, helps ensure a reasonable workplace context.
- Monitoring and alerting on permissions changes helps prevent breaches of data and network.
- Restricting root access to a select few trusted operators, ensuring passwords are extremely long and complex, rotating these passwords often, and storing these passwords in a "vault" that requires MFA and SSO to access significantly reduces the risk of accounts and data being accessed inappropriately.
- Implementing VPCs (micro-segmentation) and peering in the private cloud reduces the impact of any single breach and protects resources in the private cloud.

Costs must be kept within budget. With the public cloud, everything on the menu is available for a team to very quickly buy and deploy. Inefficient architectures or inadvertent deployments can lead to huge cost overruns. Public cloud costs should be continually monitored and tracked, and alerts sent before spending goes too far awry. For example, many subversive crypto-currency mining operations have been discovered and dismantled in the public cloud. In data ingress and egress can be free or very expensive. Extreme care must be taken to ensure the cost of video egress to subscribers is well managed (Figure 4).

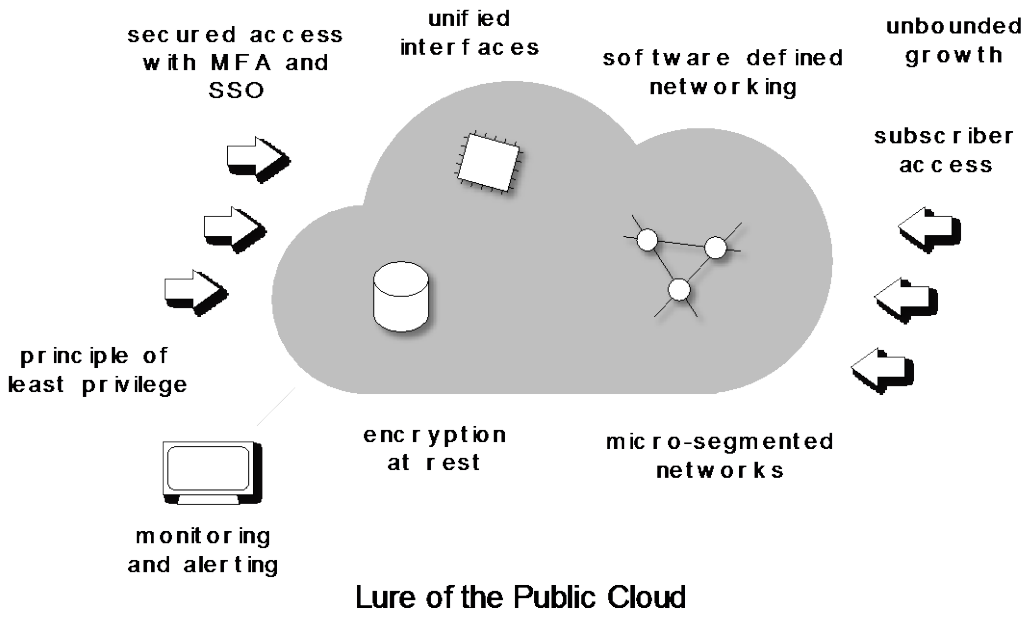

Figure 4 - Public clouds normalize runtime infrastructure.

5. Hybrid Cloud Management Platform

Every cloud needs a console. The Comcast cloud engineering team is developing a "OneCloud Portal" with the following specific features for hybrid cloud users:

Tenant management. In the OneCloud data model, each application team represents a "tenant." Each tenant has a name, funding entity identifier, set of tenant owners, set of tenant members, and a set of public and private cloud accounts for which the tenant is singly responsible. For example, a tenant could have separate public cloud accounts to distinguish their deployments into development, staging, and production. A tenant could additionally have separate accounts in a private cloud, one per regional deployment. OneCloud integrates corporate SSO to manage logins for the portal. Once logged in, however, any user can see how any other tenant is using the hybrid cloud. Such transparency is useful for knowledge sharing and reuse across the enterprise.

Costs. Information technology departments use "showback" reports that explain how much their cost centers will be charged for resources consumed. For each supported cloud provider, the portal provides monthly, quarterly, and annual costs attributed to the accounts and aggregated by tenant, specifically including operational overhead and volume discounts. For supported private clouds, we show operating expenses for each account using a total cost of ownership model, developed specifically for this project. Using showback reports, users can compare cloud provider costs, find opportunities to reduce spending, and budget for growth.

Permissions. Permissions in the hybrid cloud can be stored in a corporate user management system such as Microsoft Active Directory. The OneCloud Portal provides a convenient way for tenant owners to assign tenant members and pseudo-users specific permissions in specific clouds.

Roles. Role based access control (RBAC) simplifies user permission management by letting users bind to roles and letting roles bind to access policies. When the prefabricated roles are insufficient, users can author roles and ultimately bind policies in supported clouds.

Documentation. In addition to tenant and account management, we provide a convenient starting point for teams to learn how to use the hybrid cloud, how it is governed, and "best practices" architectural patterns for use by application developers (Figure 5).

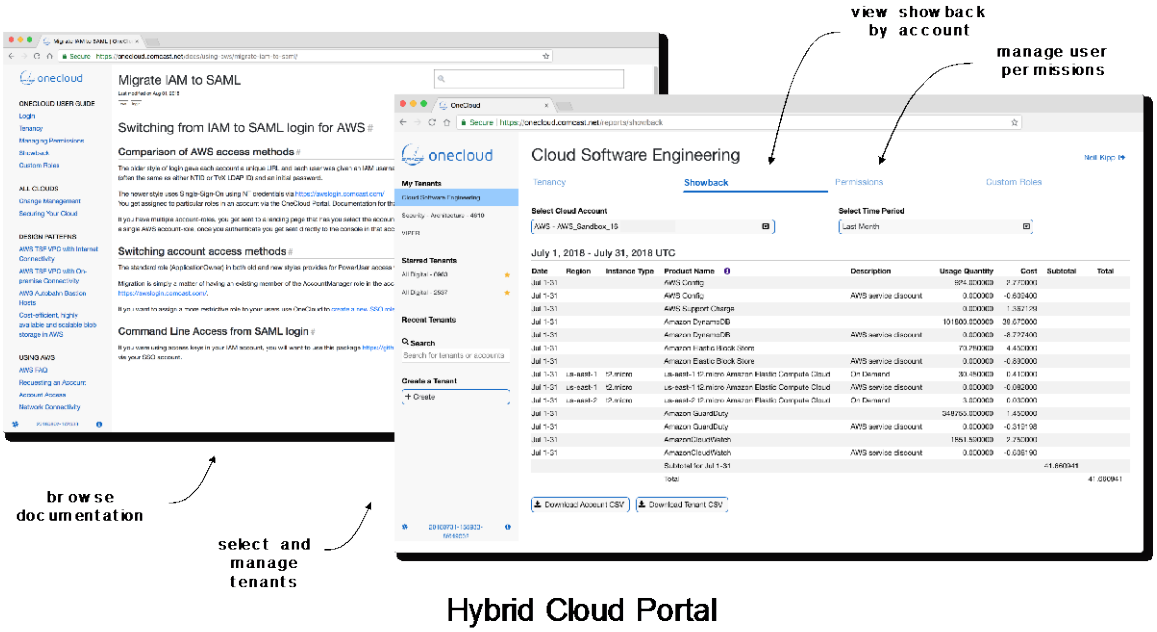

Figure 5 - The hybrid cloud portal lets users browse relevant documentation and manage tenancy, showback, and permissions.

6. Beyond Virtualization: Containers, Orchestration, Serverless

Cable companies can quickly take advantage of recent advances in compute resource virtualization including containerization, orchestration, and serverless computing, discussed below.

Containerization. A "virtual machine" is an emulation of a server running with the assistance of a hypervisor application. Hypervisors can provide a compute density of approximately 10 virtual machines per host. A "container" such as those provided by Docker or Rkt technologies is a single process or small group of processes running directly on the underlying operating system, without the overhead of a hypervisor. A single host can run as many as 100 containers. Thus, the efficiency gained by containerization can increase a server's compute density by an order of magnitude. Note that containerization does not improve network resource efficiency, and compute density might be constrained by the physical network capability of the underlying host (Figure 6).

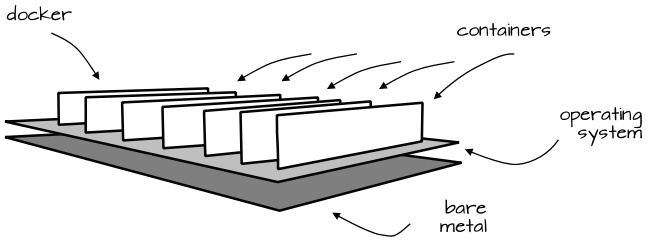

Figure 6 - Containers virtualize the operating system with increased compute density.

Orchestration. Like an operating system schedules processes, an orchestration system such as Kubernetes or Apache Mesos schedules containers across multiple servers. In this way, developers can create robust, long-lived "microservices" that scale independently alongside short-lived "jobs" useful for big data analytics. Note that orchestration systems themselves require installation and maintenance. Orchestration systems manage multiple tenants and container versioning, and as a result, can be used to deploy, install, test, and roll back updates *in production environments* prior to full application launch (Figure 7).

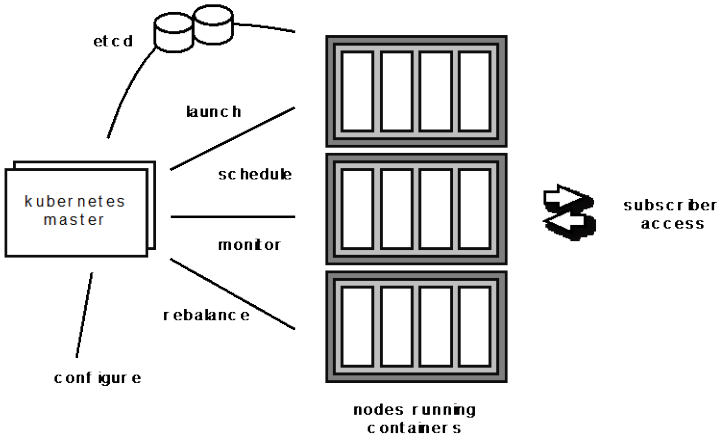

Figure 7 - Kubernetes attaches to a cluster for management, but does not broker communication from application clients.

Serverless. "Functions as a service" occurs when the runtime infrastructure has grown in sophistication such that the unit of application deployment is not a virtual machine with installed software, nor a container, but the application function that would run each time the service inside the container was called. Functions as a service are not ideal for every workload. Such functions typically have a wide deviation for startup latency, must operate within strict timeout periods, must work within random access memory (RAM) constraints, and must be written in specific languages such as Python or Node.js.

Applications must be reworked significantly to enable the move from virtual machine deployment to containers. Server applications that were once deployed on monolithic server such as those running Enterprise Java on a load-balanced Tomcat cluster can now be rewritten into containers as Go microservices and deployed on a Kubernetes cluster. Launch times are faster. Testing can be done directly in the production environment, particularly using strategies such as "blue/green" deployments and "canary" testing. Well-designed containerized applications can be updated seamlessly in production and do not require maintenance windows.

Even more so than containers, applications must be reworked to use functions as a service. However, by writing functions that integrate sophisticated cloud "software as a service" including document databases, messaging systems, and machine learning libraries, application time to market can be dramatically reduced. Today functions as a service are provided by the major public cloud providers, yet once written are not directly reusable between cloud providers.

7. Use Cases of the Hybrid Cloud

The hybrid cloud provides a substrate for innovative benefits to application developers and users. Those benefits are outlined here.

Fault tolerance. Private cloud resources can be lost during single points of failure. Adding redundant storage, network, and servers in the public cloud can make application services more robust.

Spot markets. Application teams with less time-sensitive workloads could delay processing until a public cloud provides variable pricing windows or auctions services outright. With the right tooling, application teams could take advantage of spot market pricing for resources.

Cloud bursting. For cable applications, compute and network load attributable to subscriber activity comes predictably every evening such as for prime time, every week such as for weekend football, and for special events such as the Olympics. Ostensibly, the bounded private cloud resources could handle the baseline application load and effectively unbounded public cloud resources could be used exclusively for additional, transient load. Bursting load onto public cloud requires additional engineering, but promises to be an effective way to manage variability in resource demand.

For these hybrid cloud use cases to become a reality, however, each development team must specifically engineer resource configurations, create unique scale triggers, extend application deployment scripts to run on hybrid infrastructure, while monitoring deployments in multiple clouds. Internally developed and third-party tools will lead the way to supporting these and other hybrid cloud use cases (Figure 8).

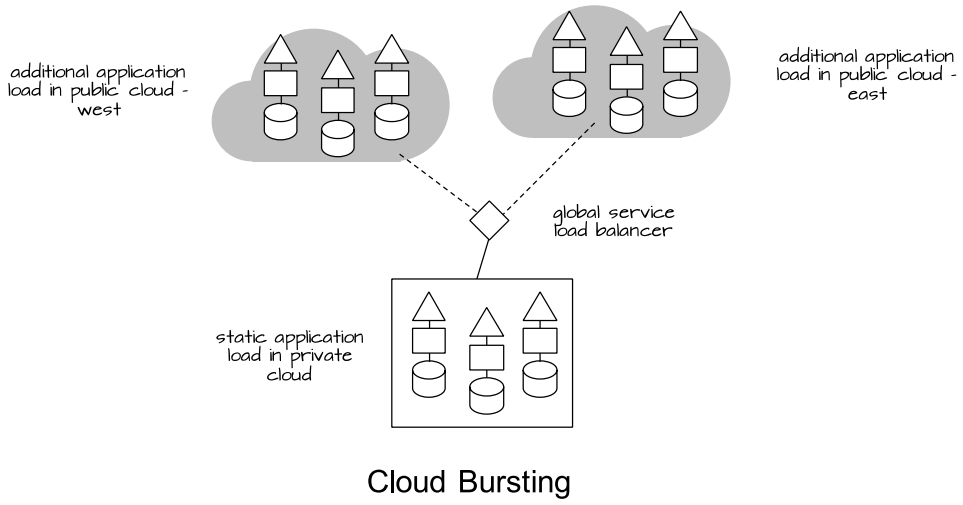

Figure 8 - During high usage, applications could auto-scale into the public cloud.

8. Hybrid Cloud Community

As with all popular technologies, conversations ensue and communities form. Here are some communication channels that we see building the communities that surround and support the hybrid cloud.

Documentation portal. Starting with the "home page" for the hybrid cloud, a documentation portal includes new feature announcements, governance deadlines, adoption instructions, case studies, and architectural design patterns. In the documentation portal, the entire community can read and contribute to the knowledge base for the hybrid cloud.

Email blasts. The cloud operations team regularly sends announcements to distribution lists concerning security governance, new feature announcements, and upcoming events.

Scheduled training. Not everyone is an expert in all aspects of the private cloud. Instructor-led and online training keeps engineers apprised of the newest and best things happening in the hybrid cloud. Training courses also inspire discussions and build bridges between teams.

Cloud Center of Excellence. Trained cloud architects, especially those with public cloud certifications, are the primary instructive resource for application teams throughout the enterprise.

Collaboration channels. Team-centric collaborative chat platforms such as Slack or Skype keep everyone up-to-date in the hybrid cloud discussion.

Cloud summit. Similar in form to an industry conference, a regularly scheduled corporate cloud summit provides a podium for leadership and technical content that coalesces discussion and provides a venue for far-flung, face-down teams to come together and share their experiences with the hybrid cloud. Catered lunches for attendees add a personal touch.

9. Summarizing the Hybrid Cloud

The hybrid cloud is a combination of private cloud and public cloud. A cable operator's private cloud has necessarily evolved and may have diverged between regions. Meanwhile the public cloud is purchased and used as a product that is largely consistent across regions. A cable-specific application stack that delivers IP linear video naturally deploys in the private cloud, provided enough physical infrastructure can be made available. For a cable company to embrace the public cloud, networks must be secured, permissions managed, access controlled, and costs must be kept within budget. Furthermore, companies will need a hybrid cloud software platform to manage tenants, accounts, showback, permissions, roles, and documentation.

Virtualization does not end with virtual machines. Teams are embracing new technologies for containerization, orchestration, and functions as a service, even though applications must be redesigned and rewritten for the cloud. Hybrid clouds provide innovative benefits to improve service or save money, especially using strategies including fault tolerance, spot markets, and cloud bursting. Hybrid clouds are fertile fields for communications and conversations, including documentation, news, training, and collaboration.

The Comcast hybrid cloud uses multiple private and public cloud technologies including OpenStack, VMWare, Amazon AWS, Microsoft Azure, and Google Cloud Platform. We run the multi-tenant portion of the private cloud in eight regions, and have single tenant deployments in more than 50 data centers. We have hundreds of tenants, thousands of users, and serve tens of millions of subscribers.

Abbreviations

CDN	Content delivery network
CPU	Central processing unit
DNS	Domain name system
GSLB	Global service load balancing
IP	Internet protocol
ISBE	International Society of Broadband Experts
JBOD	Just a bunch of disks
LB	Load balancer
MFA	Multi-factor authentication
RAID	Redundant array of independent disks
RAM	Random access memory
RBAC	Role based access control
SCP	Secure copy
SCTE	Society of Cable Telecommunications Engineers
SDN	Software defined networking
SSH	Secure shell
SSO	Single sign-on
SPOC	Single point of contact
SPOF	Single point of failure
vCPU	Virtual CPU
VOD	Video on demand
VPC	Virtual private cloud

Bibliography & References

Amazon Web Services, https://aws.amazon.com/

Ansible, https://www.ansible.com/

Apache Mesos, http://mesos.apache.org/

Docker, https://www.docker.com/

Git, https://git-scm.com/

Google Cloud, https://cloud.google.com/

Kipp, Neill, “A Highly Scalable Cloud Architecture for Delivering Linear IP Video,” SCTE Expo, Philadelphia, September 2016.

Kubernetes, https://kubernetes.io/

Microsoft Azure, https://azure.microsoft.com/en-us/

Node.js, https://nodejs.org/en/

OpenStack, https://www.openstack.org/

Puppet, https://puppet.com/

Python, https://www.python.org/

Rkt, https://coreos.com/rkt/

Terraform, https://www.terraform.io/

VMWare, https://www.vmware.com/

