

## Accelerating Deployment of Cloud Services Using Open Source Software

With the use of OpenStack\*, in combination with our own internal code and existing enterprise software, Intel IT deployed a cloud infrastructure that serves as a foundation to transform data center solutions into quickly obtainable, consumable services.

### Executive Overview

To improve the speed and availability of our cloud deployment, Intel IT recently augmented our private cloud with a hybrid solution including OpenStack\* software, a scalable open source cloud operating environment. Using OpenStack, in combination with our own internal code and existing enterprise software, Intel IT deployed a cloud infrastructure that serves as the foundation to transform data center solutions into quickly obtainable, consumable services.

Intel IT implemented Intel's first private cloud solution using existing enterprise manageability tools and solutions in 2010. A year later, seeking to better support our end users and reduce costs, we began to look for solutions to improve our on-demand self-service, reduce service provisioning time, increase the number of services we expose, and increase automated management and consumption of cloud services.

By integrating OpenStack with our implementation of automated monitoring and configuration tools, which increase our automation of resource management, we are now able to deliver:

- Support for an active/active application design that through greater redundancy and automation enables us to make faster progress toward our goal of zero downtime for cloud architected applications
- Shorter software upgrade cycles enabling faster implementation of the latest improvements, features and functionality, leading to more state-of-the-art, consumable cloud services
- Improved service levels and compliance

In the future, we plan to increase the interoperability between our private and public clouds as we move toward our goal of a federated, interoperable, and open cloud as our standard way of providing services.

- Rapid provisioning of new capacity for application developers

**Stephen Anderson**

Cloud Foundation Engineer, Intel IT

**James Chamings**

Cloud Foundation Engineer, Intel IT

**Winson Chan**

Cloud Automation Engineer, Intel IT

**Joel Cooklin**

Cloud Automation Engineer, Intel IT

**Das Kamhout**

Principal Engineer/Intel IT Cloud Lead, Intel IT

## Contents

Executive Overview.....	1
Background.....	2
Our Cloud Strategy.....	2
Solution.....	3
The Advantages of Using Open Source Software.....	5
Intel IT Cloud Platform Solution Stack.....	6
Intel IT Cloud Operating Environment.....	6
Manageability.....	7
Improving Availability and Utilization.....	9
Improving Service Levels and Compliance.....	10
Results.....	10
Conclusion.....	11
For More Information.....	11
Acronyms.....	11

## IT@INTEL

The IT@Intel program connects IT professionals around the world with their peers inside our organization – sharing lessons learned, methods and strategies. Our goal is simple: Share Intel IT best practices that create business value and make IT a competitive advantage. Visit us today at [www.intel.com/IT](http://www.intel.com/IT) or contact your local Intel representative if you'd like to learn more.

## BACKGROUND

**Intel IT operates a massive, worldwide computing environment that supports about 91,500 Intel employees and includes approximately 75,000 servers in 69 data centers. Employees connect to our services through more than 138,000 devices, the majority of which are mobile business PCs and handheld devices.**

Our cloud operations are vital to our business. They enable us to deliver a highly available computing environment where secure services and data are delivered on-demand to authenticated devices and users through a shared, elastic infrastructure that concurrently supports multiple tenants.

We implemented our first private cloud in 2010 through existing enterprise manageability tools and solutions, coupled with integration software and databases. This implementation enabled us to provide a cohesive solution that combined compute, storage, and network resources to expose compute infrastructure as a service (IaaS) to our end users.

On-demand self-service is a critical aspect of our cloud environment. To develop a true enterprise private cloud that provides a self-service portal that is end-user focused, we originally developed a hosting automation framework that included entitlement, quotas, transparent measured services, and data-driven business logic. This framework enabled us to provide capacity to our business users when they needed it, which removed IT from the critical launch path for business services and created a more agile enterprise infrastructure to support our dynamic and ever-changing set of business requirements.

Through our private cloud, we also established a federated capacity, which through the use of multi-tenancy and pooling of assets has increased resource sharing, leading to over USD 9 million in savings over the last three years. We also delivered to the operations team numerous improvements in automation and data transparency, helping us reduce infrastructure operating costs.

Our private cloud continues to be successful. Through our on-demand service inside our cloud, we have reduced the provisioning of new capacity for application developers from 90 days to acquire a server to less than three hours for a virtual machine (VM) with a service-level agreement. In many cases, it can be done in less than 45 minutes. This fits well with our objective to be able to go from innovative idea to production in less than a day.

## OUR CLOUD STRATEGY

**The Intel IT Open Cloud is an important step in our progression to a federated, interoperable, and open cloud.**

When we began to implement our private cloud in 2010, our initial goal was to supply a simple compute IaaS to make it easier and faster to acquire a server. This allowed us to introduce automation through the environment and empower our application developers and application owners to quickly provision infrastructure for their application environments.

In 2011 (see Figure 1), we partnered with our software developers to architect cloud-aware applications that work in a distributed fashion and, when node or environment failures occur, can keep the service running for the end users. This required thinking of an application as both a provider and a consumer of web services. Our goal was to partition each selected application and its capabilities into more granular components that could be implemented, tested, and scaled separately, as well as handle failures throughout the environment.

In 2012 we began moving to a full private cloud, which involved the following considerations:

- A shift to all data center solutions being exposed as consumable web services, advancing our cloud beyond compute IaaS, to include storage and network consumable web services.

- Providing platform as a service (PaaS), supplying end users with a hosting platform consisting of a web and data fabric exposed as web services. This facilitates the development and deployment of applications without end users having to deal with the complexity of the underlying infrastructure such as the OS, networking, and platform application configurations.
- An investigation and implementation of greenfield solutions, as detailed in this paper, to run cloud-architected applications in a full private IaaS environment.

In 2013 we plan to shift to more hybrid solutions—with early production work in 2012—so that a percentage of capacity is running in both our public and our private clouds. We even plan to implement modules and capabilities from the hybrid cloud model into our enterprise private cloud on a three-to-six-month cadence.

Use of an external cloud provider will allow us to abstract even further the deployment of resources, giving us the flexibility to decide, based on cost, location, and regulatory requirements, where we source our compute and storage demands. Our goal is to create

an open cloud and treat the public cloud environment as an extension of our data center services. This includes monitoring costs and making this environment look and work like an extension of our infrastructure, allowing us to provide applications and data as a seamless user experience. Whether it's internal or external, our end users need easy access to their applications and data in a secure fashion, anywhere and anytime from any device.

When everything becomes a service—compute, network, storage, and software—APIs become the building blocks and leverage points of cloud applications. Deconstructing an application into separate web service APIs enables the re-use of an application's capabilities for integration into other cloud applications, while also enabling appropriate scaling for each individual service. Well-designed APIs also improve interoperability, shielding an application from underlying technology implementation and supplier-specific implementations.

Our goal is to have a federated, interoperable, and open cloud implemented by 2014. The actual implementation depends on the industry's movement toward open solutions and open standards. Interoperability

between web services, data services, and identity services is important to help us enable federation and seamless interoperability. Through the use of common APIs or an abstraction layer, we want to make it easier for our end users to consume compute, storage, and networking from different providers.

## SOLUTION

**To increase agility, velocity, and efficiency, we performed an analysis of our current private cloud in 2011, looking to further improve its performance, self-service provisioning, and automated management.**

The primary motivation for this effort came from the new expectations we see for enterprise IT. The consumerization of IT, often associated with the policy of bring your own device, is spreading across enterprises. IT end users accustomed to services delivered on the Internet expect to acquire applications, storage, and connectivity easily, and receive updates rapidly.

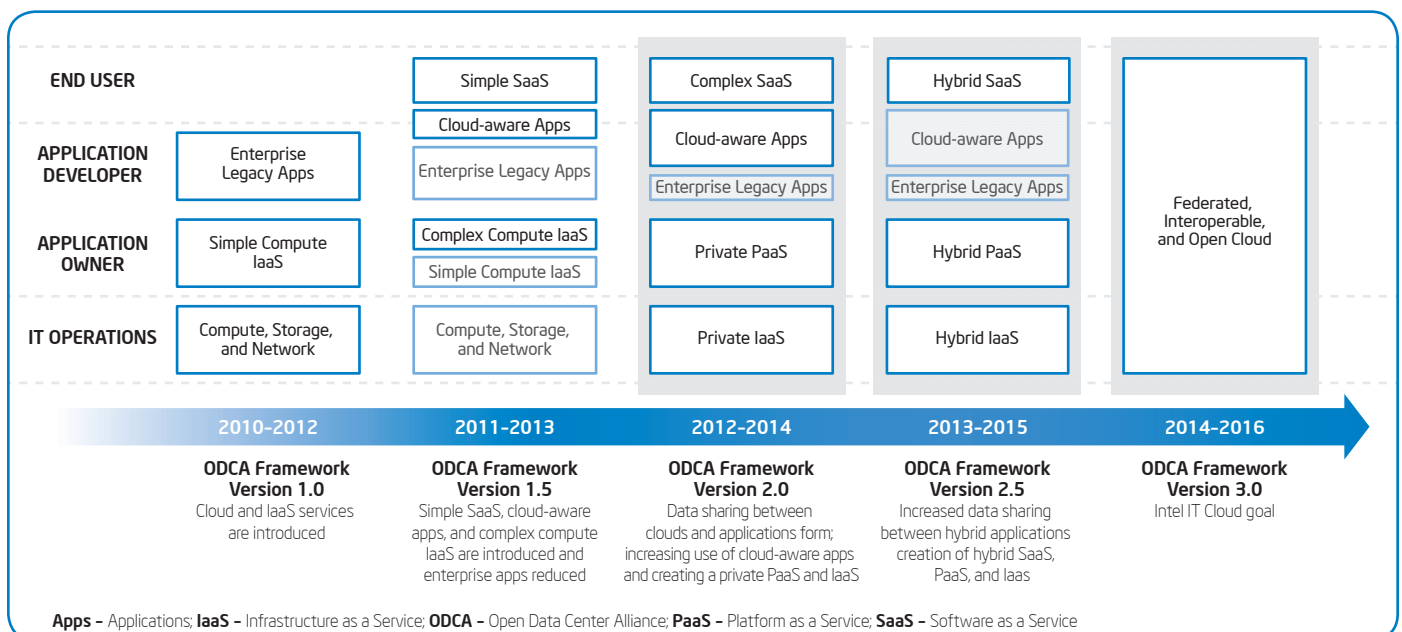


Figure 1. Intel and the enterprise IT industry are on a multi-year path to maturing the cloud. En route to the ultimate goal of a federated, interoperable, and open cloud, hybrid applications spanning private and public clouds will increasingly become the norm. This will help optimize costs and enable seamless data sharing between services for greater end-user productivity.

Planning a greenfield implementation expanded our options and enabled us to think beyond our current solution and look for the most flexible, manageable, and efficient solution to meet our needs. We needed to quickly switch to exposing all of our data center solutions as consumable services, which is a large undertaking for most enterprise IT organizations that run infrastructure that is not built with APIs. For us this meant laying a foundation for the solution: 10 Gigabit Ethernet for our network fabric, all new blade servers, and high density, two-rack unit servers for storage nodes.

We're currently launching servers based on the latest Intel® Xeon® processor E5-2600 product family. Nearly everything is redundant for resiliency and high availability, but not overdone because the software needs to assume some level of failure to contain infrastructure costs.

To meet the expectations for consumable services and enable automated management for all data center components, we compared a variety of commercial off-the-shelf, open source, and public cloud solutions, reviewing several large-scale cloud solutions from multiple

suppliers. Ultimately we decided the best solution for our environment was to augment our current environment with OpenStack®. OpenStack is an open source software stack designed to support highly scalable infrastructure. It provides an open, extensible framework for managing various resources in an IaaS cloud, including compute, network, and storage. We based our decision on the strength of the OpenStack developer community, the quality of the developer and administrator documentation, and the pace of code evolution.

## Our Cloud Runs on the Intel® Xeon® Processor E5-2600 Product Family

To meet Intel IT's needs for more processing power and energy efficiency, and better security, we are moving to the Intel® Xeon® processor E5-2600 product family for our mainstream two-socket platform.

Designed to be at the center of modern data centers and clouds, the Intel Xeon E5-2600 processor family offers the best combination of performance, capabilities, and cost effectiveness, giving us the ability to run more virtual machines (VMs) per server and consolidate VMs onto a smaller number of servers. These processors increase performance by up to 55 percent compared to the previous generation Intel® Xeon® processor 5600 series.<sup>1</sup> They also provide 50 percent more performance per watt compared to the previous generation.<sup>2</sup>

The Intel Xeon processor E5-2600 product family features Intel® Integrated I/O. This integration moves the I/O controller from a separate chip on the motherboard directly onto the processor die while also supporting the latest generation PCI Express® (PCIe) 3.0 standard and Intel® Data Direct I/O. This can help reduce I/O latency up to 30 percent over previous architectures.<sup>3</sup> Support for the PCIe 3.0 specification can improve I/O bandwidth by up to 2x.<sup>4</sup>

The Intel Xeon processor E5-2600 product family continues to build on key foundational security technologies for enterprise-wide data protection such as Intel® Advanced Encryption Standard New Instructions, which helps systems to quickly encrypt and decrypt data running over a range of applications and transactions, and Intel® Trusted Execution Technology, which helps provide hardened protection for virtual and cloud environments by creating a trusted foundation to reduce the infrastructure exposure to malicious attacks.

Optimized platform solutions are available featuring the Intel® C600 series chipset and Intel's 10 Gigabit Ethernet solutions to help ensure high performance across diverse workloads and advanced functionality, such as integrated serial-attached SCSI and Fibre Channel over Ethernet. These can help simplify, consolidate, and accelerate storage and network connectivity in today's virtual and cloud environments.

<sup>1</sup> See the Intel IT brief, "Increasing Design Throughput with Workstations Based on New Intel® Xeon® Processor E5-2600 Product Family."

<sup>2</sup> Performance comparison using SPECint\*\_rate\_base2006 benchmark result divided by the processor TDP. Baseline score of 416 on prior generation two-socket Intel® Xeon® processor X5690 (130W TDP) based on best-published score to [www.spec.org](http://www.spec.org) as of December 5, 2011. For more details, see: [www.spec.org/cpu2006/results/res2011q4/cpu2006-20111121-19037.html](http://www.spec.org/cpu2006/results/res2011q4/cpu2006-20111121-19037.html). New score of 659 based on Intel internal measured estimates using an Intel® Rose City platform with two Intel® Xeon® processor E5-2690 (135W TDP), Intel® Turbo Boost Technology enabled, enhanced Intel SpeedStep® technology enabled, Intel® Hyper-Threading Technology enabled, 128 GB RAM, Intel® Compiler 12.1, THP enabled, Red Hat Enterprise Linux Server 6.1 beta for x86\_64.

<sup>3</sup> Intel internal measurements of average time for an I/O device read to local system memory under idle conditions comparing the Intel® Xeon® processor E5-2600 product family (230 ns) versus the Intel® Xeon® processor 5500 series (340 ns). Baseline configuration: Green City system with two Intel® Xeon® processor E5520 (2.26 GHz, 4 core), 12 GB memory at 1333, c-states disabled, Intel® Turbo Boost Technology disabled, SMT disabled. New configuration: Meridian system with two Intel® Xeon® processor E5-2665 (2.4 GHz, 8 core), 32 GB memory at 1600 MHz, c-states enabled, Intel® Turbo Boost Technology enabled. The measurements were taken with a LeCroy PCIe\* protocol analyzer using Intel internal Rubicon (PCIe 2.0) and Florin (PCIe 3.0) test cards running under Windows® 2008 R2 with SP1.

<sup>4</sup> 8 GT/s and 128b/130b encoding in PCIe\* 3.0 specification enables double the interconnect bandwidth over the PCIe 2.0 specification. Source: [www.pcisig.com/news\\_room/November\\_18\\_2010\\_Press\\_Release](http://www.pcisig.com/news_room/November_18_2010_Press_Release)

## The Advantages of Using Open Source Software

IT organizations may be concerned that open source software could carry certain risks, including the following:

- A potential lack of technical support for open source applications, compared to off-the-shelf products, which often have dedicated support teams for technical issues
- The possibility that the community developing the open source solution may eventually head in a different direction than is in the best interest of the company, or the open source solution may be acquired by a large company
- The accrual of implementation time, and training and redevelopment costs that eventually exceed that of an off-the-shelf solution
- The concern that open source is unregulated so anyone can compromise the code

Our Enterprise IT team shared many of these concerns. However, Intel IT has a long history of using open source software for our Design Grid and have found the following advantages:

- A strong, vibrant community with excellent support options through third-party companies. A large number of companies of our size or larger implementing or running OpenStack, supplemented by their own code. The widespread use of OpenStack ensures that features and functionality will continue to reflect our requirements.
- Strong documentation, including administration guides, API documentation, and developer documentation to ensure a quick learning curve.

- Full access to open source code repositories, including rigorous peer review, as well as acceptance and regression testing before any code is committed to the project.
- Frequent update cycles—a major release every six months—to allow fast implementation of the latest features and capabilities, including performance and efficiency improvements.

In our evaluation of OpenStack as open source software, we found that OpenStack was not only a safe choice, but also matched well with our other large-scale computing requirements. In addition, we saw compelling advantages in moving from proprietary solutions to open source, including the following:

- **The ability to be interoperable, minimize vendor lock-in, and help work toward our goal of a hybrid cloud.** Our development teams can focus on higher value cloud capability areas and work with the community to quickly mature the open cloud industry.
- **Move faster and at a lower cost.** In contrast to having to implement the basics for our private cloud environment, with open source software much of the work has already been done in the open source community. Because others are also using OpenStack, IT organizations can write the necessary core code together instead of individually and then share the code through the community.
- **Spend less time providing compute IaaS and more time providing higher level services areas.** This exposes the more advanced services our application developers need to drive higher end-user productivity.



### OpenStack\*

OpenStack is a cloud OS that controls large pools of compute, storage, and networking resources throughout a data center, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface.

As an open source solution, OpenStack is developed and supported by a global collaboration of developers and cloud computing technologists. The project seeks to deliver solutions for all types of clouds by being simple to implement, scalable, and feature rich. The technology consists of a series of interrelated projects delivering various components for a cloud infrastructure solution. All OpenStack source code is available under the Apache 2.0 license.

Intel IT Cloud Platform Solution Stack

As shown in Figure 2, our new cloud solution consists of the following:

- An interface layer exposing a graphical user interface (GUI), a command line interface (CLI), and an API that exposes all key data center components as consumable services. Both the GUI and CLI are developed on top of the API layer.
- A manageability layer with a configuration management database (configuration and state management), Watcher (event monitoring), Decider (automated rule-based decision maker), Actor (configuration and state enforcement), and Collector (operational database for runtime configuration and inventory) components.
- An OpenStack cloud operating environment with its foundational components.
- A physical infrastructure including compute, storage, and network components.

To minimize the impact of new infrastructure and software version updates, we designed our cloud platform for rolling upgrades, which

helps us better manage the rapid changes in IT infrastructure and software requirements for employee devices. For example, a new release of OpenStack comes out every six months, and we expect to integrate the next version—Folsom—within three months of its release. Rolling upgrades of the core services and object storage with no end-user downtime are a key feature of the continuous integration and evolution philosophy at the heart of the open cloud operating model.

Intel IT Cloud Operating Environment

The cloud operating environment consists of various open source services that provide the underlying capabilities consumable by SaaS and end users through the API or GUI. Each of these components is part of OpenStack 2012.2—the Essex release.

NOVA COMPUTE SERVICE

The primary component of the cloud operating environment is the Nova compute service. Nova schedules deployment of VM images to an appropriate host and manages the VM through its operational life span. We use locally

connected, non-shared iSCSI resources to host all VMs and a deployment model based on a base image plus differencing disk for the most efficient use of disk resources.

Nova is designed to operate as much as possible as hypervisor-agnostic. It works with open source libraries such as libvirt, an open source API and management tool that can be used to manage most major kernel-based VMs that are supported. We chose a kernel-based VM as our hypervisor for its speed, reliability, and excellent CPU and memory resource management.

GLANCE

The Glance component is our image storage repository and currently resides on the only shared storage platform available to IaaS. Glance also stores VM snapshots, which in the context of the Essex release of OpenStack are singular crash-consistent copies of running VMs. We plan to use Glance as an object storage back-end later in 2012. This will provide a secure, robust VM image and snapshot storage solution to support a large number of VMs based on both Windows\* and Linux\*.

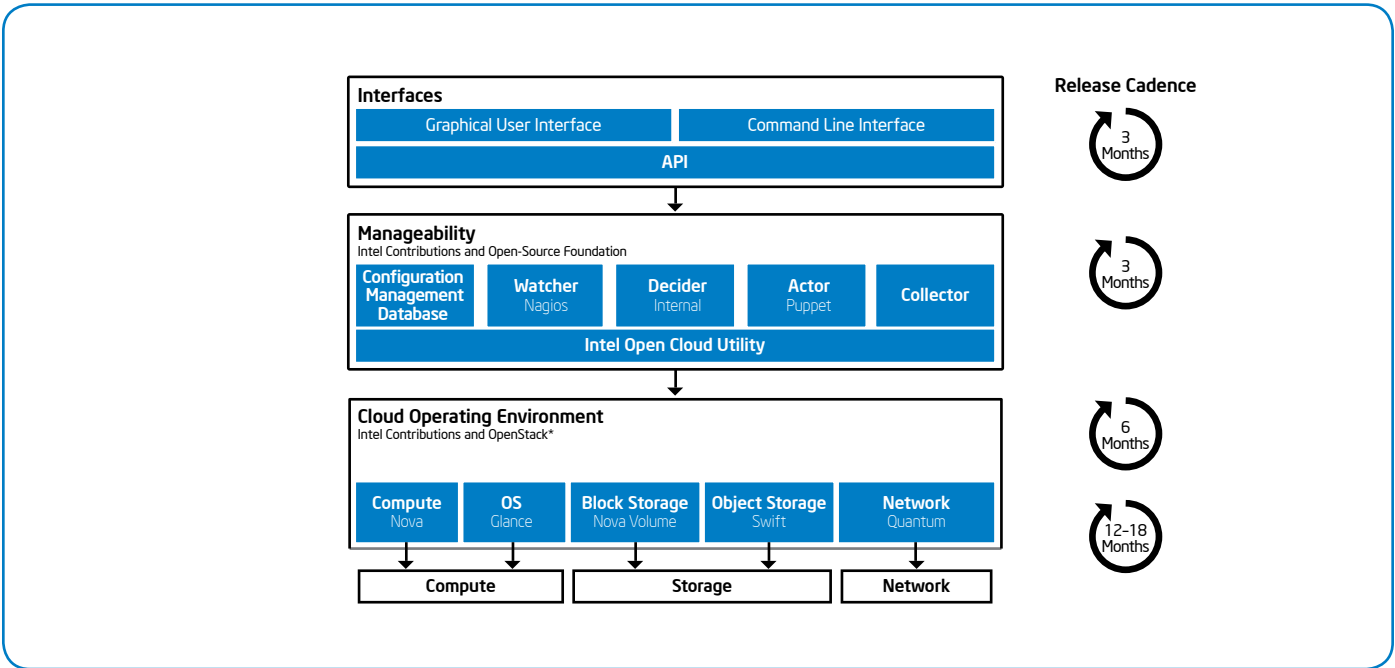


Figure 2. Diagram of the Intel IT Cloud Platform Solution Stack showing open source and other components, and their associated refresh cycles.



## NOVA VOLUME

Nova Volume (or its successor, Cinder, when available) is a service we plan to deploy in the fourth quarter of 2012. It will serve as a robust, high availability (HA) persistent block storage solution for VM use, enhancing the overall platform capabilities.

## SWIFT

Swift is one of the supported object storage solutions available to use within OpenStack. It employs a distributed storage methodology that uses commodity hardware and by design is HA. We plan to use Swift as a backing for Glance VM image and snapshot storage starting in the fourth quarter of 2012, and within the same time frame we plan to expand this to provide an open and API-compatible version of Amazon Simple Storage Service for an end-user object storage solution.

## QUANTUM

Quantum is an OpenStack project that provides network connectivity as a service between interface devices, such as virtual NICs, managed by Nova and other OpenStack services. As Quantum becomes more mature in OpenStack and in solutions provided by OEMs, we expect our networking architecture will be modified to use it more extensively for complex segmentation and real-time definition of the network configuration.

## USE OF ZONES

We have found that a few components of the virtual environment are structural, such as the core identity and the directory services and patch repositories that our guest OSs use. To ensure reliability for this small amount of virtual infrastructure, we divided our compute and storage resources into two zones—stateful and stateless—both addressable by the same set of cloud APIs.

VMs that are meant to be essentially permanent reside in the stateful zone of the deployment. Stateful resources enjoy more robust, redundant storage and some measure of isolation for both performance and security concerns. The stateless zone of the environment is where all cloud-aware and automated resources are deployed. The storage and compute solutions here are tuned for maximum available capacity as the primary design goal. In the future, we plan to add more stateless zones, such as a low-cost development zone.

## SECURITY

Another key aspect of our OpenStack implementation is the use of Nova security groups to allow for automated logical segmentation among tenants in the environment, as well as between VM roles inside of a single tenant. Such separation is important for enabling secure multi-tenancy. The utilization of Nova security groups enables abstraction and rapid configuration of iptables, the built-in Linux kernel-resident firewall system. This significantly lowers the effort expended in ensuring reasonable segmentation in a multi-tenant resource pool, avoiding the need to coordinate automation with an external firewall service.

## Manageability

The use of OpenStack as the IaaS framework presented a number of challenges. OpenStack is at the core of the cloud operating environment, but the core is not currently pre-integrated with the type of manageability features that Intel IT needs to manage cloud resources in a scalable way. We needed a layer of automation to:

- Monitor the nodes in the infrastructure, as well as the application servers, which we refer to as the *Watcher*.

- Determine the appropriate response to relevant events, which we refer to as the *Decider*.
- Act on the decision, which we refer to as the *Actor*.
- Maintain an audit trail for compliance, which we refer to as the *Collector*.

To perform these functions, we designed the Intel Open Cloud Utility incorporating five manageability components: Configuration Management Database (CMDB), Watcher, Decider, Actor, and Collector.

## INTEL OPEN CLOUD UTILITY

At the core of our automation and manageability solutions is the Intel Open Cloud Utility. Exposed through a REpresentational State Transfer (REST) API, the utility manages configurations, templates, relationships, state, and integration across multiple element managers. Some of the most commonly leveraged element managers include Nova for OpenStack compute, Nagios for monitoring, and Puppet for VM customization and application deployment. It also includes public cloud compute and hardware load balancers.

When a user requests a new VM instance or a collection of VM instances, the API records the request and provisions the virtual infrastructure through the appropriate element managers. Puppet then customizes the VMs and applications and deploys them, pulling its manifests directly from our Open Cloud Utility. If an actionable event is received, the utility immediately takes action while Puppet ensures and enforces consistency. Actions can range from taking an individual VM out of a load balancer pool, promoting a slave to become a master, or taking an entire auto-scaling group (ASG) offline, enabling its matching ASG—presumably running in our other Open Cloud Utility data center—to respond to

application requests. Auto-scaling is an open source solution for automatically increasing or decreasing capacity based on demand to ensure performance is maintained and costs minimized.

## CONFIGURATION MANAGEMENT DATABASE

At the core of our automation and manageability solutions is the CMDB that keeps track of the VMs that are deployed into the environment. This domain model records desired settings, configurations, relationships, and states. Back-end orchestration—the automated arrangement, coordination, and management of our computer systems, middleware, and services—and automation solutions use this model to enforce states in the environment. For example, when an end user requests a new collection of VMs, the API records the request into the model and triggers automation to start provisioning. Once a VM is provisioned, the Actor reads the desired configuration and state information from the model and enforces that on the VM. Our current implementation of CMDB is based on an open source Web 2.0 application framework, and we expose it through a RESTful API, a web service implemented using HTTP.

## WATCHER

Our Watcher is based on Nagios, an open source monitoring tool. The Watcher takes advantage of Nagios's extensive list of pre-built monitors, support of multiple OSs, and ability to monitor many resources beyond the OS, including load balancers, firewalls, and network switches. As a Watcher, Nagios provides enterprise-class open source monitoring of systems, applications, services, and business processes. If a failure occurs, Nagios runs health checks, identifies issues, sends actionable alerts,

and enables remediation processes that can resolve problems before they affect business processes, customers, and end users. Our plan is to expose monitoring as a service and empower our end users with the ability to enable and disable from a selection of core monitors in their environment or to define their own custom monitors. Nagios is deployed through Puppet, which is described below.

## DECIDER

Our Decider component is a real-time configuration system designed to make state-based decisions. Our current implementation of the Decider leverages both custom and default event handlers. The Decider subscribes to alerts published by the Watcher. On receipt of an alert, an action, generally a stop, is called against a specific VM instance. The most powerful extension point for a tenant is the VM sub-classing, through which a tenant can respond to events with custom business logic. As described earlier, depending on the configuration and current state, the decision may be to shut down an application server and remove it from the load balancer pool, or, if no automated remediation has been identified, it may simply auto generate an incident in our service management system for the operations team. Actions may also be called against ASGs and, similar to the VM sub-classing, they may be sub-classed to provide custom logic.

## ACTOR

To execute and ensure configuration actions, we use Puppet, an open source IT automation software tool, as our Actor. Puppet is designed to help IT organizations manage infrastructure throughout its lifecycle, from provisioning and configuration

to patch management and compliance. Puppet enables the automation of repetitive tasks, quickly deploys critical applications, and can proactively manage events that require scaling from 10s to 1000s of servers, on-premise or in the public cloud. Puppet pulls its manifest directly from our Open Cloud Utility, which helps us ensure that the current state within the utility is reflected on the server. For instance, if an alert is sent just prior to a server unexpectedly rebooting, the Watcher component could possibly have acted on the troubled server, saving its new state and attempting to enforce the state immediately. However, since the server was offline, the desired action could not be taken. Nevertheless, because Puppet is configured to run at startup, Puppet will ensure the system conforms to its currently configured state in the Open Cloud Utility.

## COLLECTOR

Like the Actor, the role of Collector is at least partially played by Puppet through the Facter feature. Facter is a cross-platform program for retrieving basic server information about the hardware and OS. It acts as our runtime CMDB. We also add several custom facts to every VM we manage that help relate the VM to its Open Cloud Utility VM template, tenant, ASG, support group, and more. This information is provided to the service management platform to enable problem and incident management.

## CLOUD RESOURCE MANAGEMENT IN ACTION

Figure 3 shows how the different components in our automated cloud resource management work together. The Watcher sends alerts onto the message bus that the Decider subscribes to. The message bus is a combination of a common data model, a common command set, and a



messaging infrastructure that allows different systems to communicate through a shared set of interfaces. The Decider then makes decisions based on real-time configuration data of the infrastructure and application layout, and if necessary, instructs the Actor to enforce changes through the CMDB.

For example, if the Watcher detects a specific node having issues inside of a given application scale unit (combination of server instances for scaling), the Decider disables that node and directs the Actor to provision a replacement. If a more catastrophic failure happens that affects the entire data center, the Decider may disable a scale unit inside the data center or even remove the data center completely from the global load balancer list of DNS end points.<sup>5</sup>

## Improving Availability and Utilization

As one of our many initiatives, Intel IT is working toward achieving higher availability. Our availability goal is 99.99 percent for key business services, which means no more than 52 minutes of planned or unplanned downtime a year. Such HA requires significant automated remediation to achieve. Currently, our availability level ranges from 99.7 to 99.95 percent depending on the application.

One way we're working toward this goal is through an active/active application design. As shown in Figure 4, in an active/active implementation, two or more data centers are simultaneously in an active state, each participating in running common applications on different virtual servers. The data centers are completely symmetric. Any transaction can be routed within the application network

to any data center, which then reads or updates any set of data items.

An active/active approach provides the most flexibility and maximizes system investment as requests are load-balanced across all available processing capacity. If a data center fails, users at the other data center(s) are unaffected. Furthermore, the users at the failed data center are quickly switched to surviving data centers, restoring their services quickly. By using multiple instances in an active/active mode, we achieve better performance, availability, utilization, and flexibility.

In the first phase of this project, we focused on three levels of automated remediation, which enable a cloud service to operate with high reliability and availability across multiple data centers.

- Destroy and re-create the node.
- Remove the scale unit from the load balancer, which in some situations means removing large numbers of servers from the load balancer.
- Remove data center from the global load balancer pool.

We plan to add more complex use cases in later phases. We see the need to evolve the Decider with a correlation engine that is capable of performing root cause analysis against a dynamic model. For example, the Watcher and Decider may suddenly receive a large number of alerts for a group of application servers. The culprit is an underlying network or storage failure, but the Decider is not aware of that and proceeds to perform appropriate actions on this group of application servers. By evolving the Decider with a more capable correlation and analysis engine that can understand the context of the messages it receives, we can enable it to make more intelligent decisions.

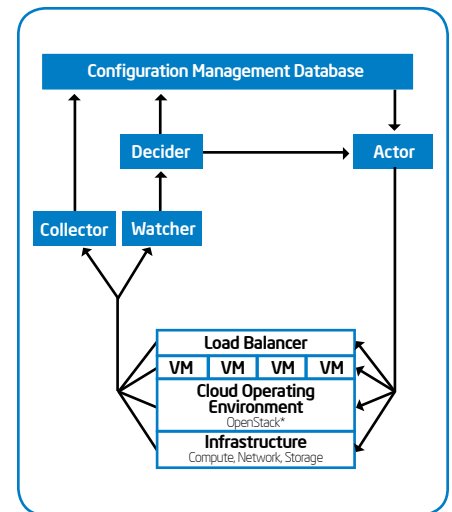


Figure 3. To automate the management of cloud resources, Intel IT added a layer on top of OpenStack\* that includes the configuration management database, Watcher, Decider, Actor, and Collector functions.

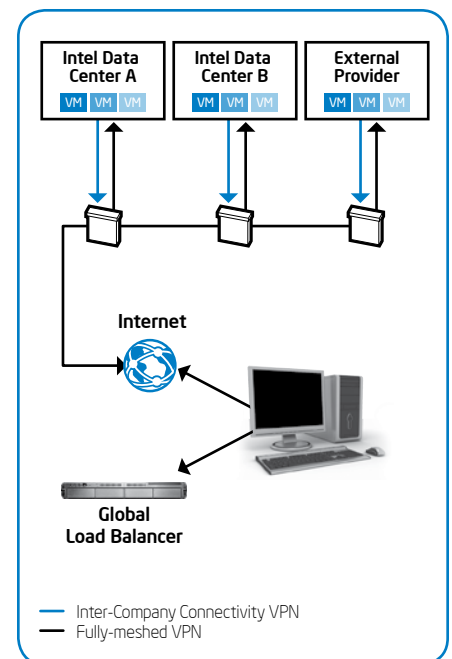


Figure 4. In an active/active application design, two or more data centers are simultaneously in an active state running common applications on different virtual servers to ensure high availability.

<sup>5</sup> To learn more about how to design to recover quickly from a failure, see the Open Data Center Alliance's white paper [Developing Cloud-Capable Applications](#).

## Using Open Source Software for Platform as a Service

Intel IT is actively implementing platform as a service (PaaS) as the next logical step for our enterprise private cloud, to accelerate custom application deployment and promote cloud-aware application design principles. This PaaS environment will build on our already successful infrastructure-as-a-service efforts. We will use open source software to provide an environment featuring self-service, on-demand tools, resources, automation, and a hosted platform runtime container. We anticipate that PaaS will facilitate the creation of cloud-aware applications through the use of templates, resource sharing, reusable Web services, and large-scale multi-tenancy. Having already conducted a successful proof of concept, we are in the process of offering an early adopter pilot, which we expect to lead quickly to full production deployment. By increasing programmer productivity, we anticipate our PaaS will enable us to extend the value of our private cloud to more groups and usages, thereby supporting our technology roadmap for using hybrid (private-public) clouds to further increase scalability and cost efficiency.

## Improving Service Levels and Compliance

Like many IT organizations, Intel IT has investments in numerous enterprise technologies—from service management tools to authentication and entitlement tools. A primary goal for our open source infrastructure was to see how well we could integrate it with existing solutions, such as our service management system, that run our enterprise. Integration into our service management system is important in particular because we are in the process of transforming into a complete Information Technology Information Library environment.

Our architecture and design goals require that our systems provide the necessary data for tracking service levels and compliance. The utilization of the configuration management system, coupled with the monitoring system and correlation engine, allow for correlation of resources at provision time. This information is then fed onto the message bus and imported into the service management tool.

The Watcher is also fed information at provision time to ensure that the resources are immediately monitored and that alerts on those resources can be easily ingested into the service management tool again through the use of the message bus. This allows fast, automated remediation to occur in a self-contained fashion, with only exceptions requiring an operator to receive an incident ticket for problem management.

By utilizing a message bus model with publish and subscribe methods, the design allows for a very flexible approach to identifying and recording what causes alerts, what causes auto-remediation, and what generates a ticket for operator analysis.

## RESULTS

**Through greater automation of resource management and other optimizations, the new Intel IT Cloud Platform Solution Stack is a significant step toward our goal of a federated, interoperable, and open cloud. It enables us to spend less time on engineering the core IaaS solution and more time on higher level service areas that allow us to expose the more advanced services our application developers need to build applications that increase end-user productivity.**

Our new solution stack has dramatically reduced the amount of time it takes to provision services and automatically resolve many resource issues. We can now deploy a VM to just five to 10 minutes. In addition to providing faster self service to our customers, our solution stack is providing a more reliable infrastructure with rolling upgrades that will keep our infrastructure current while reducing staff involvement.

We also anticipate that our implementation of an active/active application design will drive significant results. We expect to see improved utilization and availability, as well as a faster mean time to recovery.

## CONCLUSION

**As we move toward our goal of a federated, interoperable, and open cloud, open source projects such as OpenStack enable us to deploy a cloud infrastructure that serves as the foundation to transform data center solutions into quickly obtainable, consumable services.**

While we will continue to use commercial software when it provides the best solution and fit for our cloud evolution, OpenStack provides a versatile tool for the building of an open, extensible framework for managing and exposing the various resources, such as compute, network, and storage in our cloud.

In the short term, we expect to continue the improvement of our cloud platform at every level. Our next areas of focus include orchestration, block storage, auto-scaling policies, live migrations, and complex application deployment. We also plan to continue our evolution to hybrid solutions, increasing our ability to easily consume public cloud services.

## FOR MORE INFORMATION

**Visit [www.intel.com/it](http://www.intel.com/it) to find white papers on related topics:**

- "An Enterprise Private Cloud Architecture and Implementation Roadmap"
- "Best Practices for Building an Enterprise Public Cloud"
- "Extending Intel's Enterprise Private Cloud with Platform as a Service"
- "Implementing On-Demand Services Inside the Intel IT Private Cloud"

**For more information on Intel IT best practices, visit [www.intel.com/it](http://www.intel.com/it).**

## ACRONYMS

ASG	auto-scaling group
CLI	command line interface
CMDB	configuration management database
GUI	graphical user interface
HA	high availability
IaaS	infrastructure as a service
PaaS	platform as a service
REST	Representational State Transfer
SaaS	software as a service
VM	virtual machine

---

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, reference [www.intel.com/performance/resources/benchmark\\_limitations.htm](http://www.intel.com/performance/resources/benchmark_limitations.htm) or call (U.S.) 1-800-628-8686 or 1-916-356-3104.

This paper is for informational purposes only. THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. Intel disclaims all liability, including liability for infringement of any patent, copyright, or other intellectual property rights, relating to use of information in this specification. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein.

Intel, the Intel logo, Intel SpeedStep, and Xeon are trademarks of Intel Corporation in the U.S. and other countries.

\* Other names and brands may be claimed as the property of others.

