

OPENSTACK & CLOUD COMPUTING



*an eBook primer
presented by **Mirantis***

Featuring an excerpt by John Rhoton



Contents

About this Book	III
Foreword: Why OpenStack Cloud, and Why Now?	IV
Preface	1
Analyze	5
Chapter 1 Study the Cloud	6
Core Attributes	6
Service Models (SaaS, PaaS, IaaS)	8
Deployment Models (Public, Private, Community, Hybrid)	11
Value Proposition	15
<i>Practical Recommendations</i>	19
Chapter 2 Gauge your Maturity	20
Enterprise Cloud Adoption Path	20
Software-Defined Data Center	21
Flexible Cloud Sourcing	23
<i>Practical Recommendations</i>	28
Assess	29
Chapter 3 Explore the Landscape	30
Software Services	30
Platform Services	31
Infrastructure Services	34
<i>Practical Recommendations</i>	40
Chapter 4 Make the Selection	41
Start with the Infrastructure	41
Leverage Private Assets	41
Maximize Flexibility with Open Source	42
Find the Best Fit	43
<i>Practical Recommendations</i>	46
OpenStack Cloud Computing Overview	47
Acronyms	49
About the authors	54
Also by John Rhonet	55
About Mirantis	58

About this book

[OpenStack Cloud Computing](#) by John Rhoton is an introduction to building a cloud based on OpenStack technologies. OpenStack's modular and extensible components enable enterprises and service providers to improve the efficiency, agility, security, quality and usability of their data center operations. [Mirantis](#) is the pure-play OpenStack company, delivering all the software, services, training and support needed for running OpenStack. Mirantis is sponsoring this eBook to guide you in integrating distributed, heterogeneous infrastructure components into a single, open cloud framework. To try OpenStack for yourself, download Mirantis OpenStack, or check out Mirantis Managed Services.

FOREWORD BY MIRANTIS

Why OpenStack Cloud, and Why Now?

With open source well established as an engine for innovation, owing to its ability to deliver rapid development outside the confines of a single company's IT team, it's frequently shortlisted by companies looking for a competitive edge with their IT. And unless you spend each morning deciding which of your "COBOL FOREVER" t-shirts you are going to wear to the office, you know cloud computing has emerged as the go-to platform for forward thinking IT organizations.

It should come as no surprise, then, that OpenStack comes up more and more in discussions of cloud infrastructure roadmaps, not least as a viable alternative for building out Infrastructure as a Service (IaaS). OpenStack, as the open cloud computing operating system, is the fastest-growing open source cloud project. Hundreds of organizations and thousands of individuals contribute code to continually improve and extend it.

OpenStack has been described as the fabric that glues together data center infrastructure. OpenStack software enables enterprises to tie together distributed, heterogeneous infrastructure components into a single, open cloud fabric. It allows companies that have long dealt with disparate infrastructure components that weren't designed to work in concert to instead weave them together under a common, industry-standard, well-specified set of application programming interfaces (APIs). And by standardizing on those common APIs, it allows you to take advantage of vendor innovations in your infrastructure core, tied together by the open fabric at the edge.

By reading this Ebook by cloud computing expert John Rhoton, you should get a good introduction to what cloud computing entails, as well as design considerations and implementation options for OpenStack clouds. And with that understanding in hand, you will be ready to deploy OpenStack to solve your business needs.

Cloud Computing Agility: The Basics

Before moving forward with OpenStack, it's essential to understand what cloud computing is and why it's important. While it may seem obvious, it's worth stating clearly that OpenStack is a platform for cloud computing, which it enables by delivering infrastructure as a service. IaaS takes infrastructure resources—such as a virtual-machine [disk image](#), raw [block storage](#), and file or [object storage](#), firewalls, load balancers, IP addresses, [virtual local area networks](#), and software packages¹—and exposes them to cloud users and operators in one, consistent RESTful² interface. Sounds cool, all right; but how do you know that this is what you need?

Here's what you need to know to understand cloud computing before you get going.

In its most basic sense, the cloud is a network—the global Internet is the prime example. Organizations using the public cloud use computing resources that are hosted remotely and delivered through the Internet. Some organizations opt for private clouds, where they take advantage of cloud computing technology and maintain the infrastructure and control the security of it themselves. Public and private clouds have become so popular because cloud computing enables efficient, on-demand access to a shared pool of configurable resources—including servers, storage, networking, applications, and services that can be rapidly provisioned, automated, and managed more easily.

In a public cloud environment, customers share resources; in a private cloud environment departments or cost centers share resources. This sharing or pooling of resources enables networks to scale, and allows for better

1 http://en.wikipedia.org/wiki/Cloud_computing#Infrastructure_as_a_service_.28IaaS.29

2 <http://www.ibm.com/developerworks/library/ws-restful/>

allocation and utilization of the shared resources, which translates to cost savings. Exposing them through a single, standardized, well-specified set of interfaces makes them easier to share and to automate.

But the greatest allure of the cloud may be that by letting cloud providers do what they do best—build infrastructure and development environments that scale—you can leverage their expertise and investment and focus on your core strengths, like writing applications and services for your customers, or more efficiently serving your employees. Cloud takes the cost of infrastructure out of the picture. Why rebuild what Google, Amazon, IBM, and others have built?

“Companies are using the cloud rather than buying new computers and software for their IT projects,” [Business Insider reported](#). *“They spent \$131 billion in 2013 on the cloud”,* [Gartner says](#). *“They will [spend] \$174.2 billion in 2014 and \$235 billion by 2017”,* [predicts market research firm IHS](#).

This may be bad news for manufacturers of servers and enterprise software, but it shows many companies are moving to the cloud to bypass fixed costs associated with building and maintaining their own data centers.

Achieving Flexibility and Cost Savings Via The Cloud

The cloud is defined by a few elements: virtualization, delivery “as a service”, elasticity, flexible billing, universal accessibility, simplified management, affordable resources, multi-tenancy (sharing of resources among organizations), and service-level management. It’s one thing to understand these features; it’s another to consider what they do for you.

Virtual: Virtualization is a layer of abstraction that lets you extract more from physical resources to reduce costs, enhance agility, and boost resource recoverability.

“As-a-service” model: Cloud computing delivers resources—infrastructure, platform, and software—“as a service.” In a public cloud environment, the service model lets you rely on a provider for the infrastructure plumbing, so you can focus on enabling applications and workloads specific to your needs. It also simplifies licensing, as you don’t need to acquire or directly pay for perpetual software licenses.

Elastic: The cloud is scalable, and services can be scaled up or down rapidly as needed. Your company or department gets billed like a utility. You pay for what you use.

Flexible billing: You can bill or be billed on a subscription basis, or by usage. This usually leads to substantial cost savings and efficiency.

Universally accessible: Cloud resources are available to anyone on any device authorized to use them, so customers or users can be anywhere and always connected.

Simplified management: Cloud environments offer automated provisioning and configuration management. They are self-service, so you can expedite resource allocation required for your business processes. And programming interfaces let you tie them into existing management frameworks.

Affordable: In a public cloud, resources cost less because you don’t need to purchase and maintain them. Public cloud computing changes operations spending from capital expenses to operating expenses. And service providers operate on a scale that lets them optimize their costs in a way that smaller companies aren’t able to. In a private cloud environment, you also get similar cost savings from virtualization, as the cloud provides resources that are actually used or likely to be used.

Multi-tenant: In public clouds, companies often share some of the same resources, which leads to cost savings. Each tenant’s data and activity is isolated from others.

Service-level management: In many public cloud environments, you enter an agreement with the cloud service provider for an expected level of service availability; the onus is on the provider to guarantee availability and uptime.

Taken together, these virtues make for a better fit between your applications and workloads and the infrastructure

they run—not just once, but all the time, making it easier to change them more quickly than with conventional infrastructures. Increasingly, it's those applications and workloads that connect you to your customers, markets, and users. The faster you can change them, the more you can add functionality that differentiates you and your value proposition from the competition, making it more likely that you will outpace your competition.

The Innovation Fast Track: Single Vendor Vs. OpenStack

Cloud computing is a competitive space with hundreds of vendors continuously bringing new innovations to market across compute, storage, and networking. Quickly adopting these innovations to solve critical business problems is crucial for an enterprise to stay competitive in today's information-driven economy.

However, depending on the way you approach cloud computing, you may eventually find yourself having to support a variety of different cloud infrastructure components that weren't necessarily designed to work together or talk to each other. Proprietary solutions from infrastructure providers create silos that remain underutilized, are costly to maintain, and most importantly, stifle the agility of your organization as a whole. In the past, the solution to this problem was to standardize infrastructure by leveraging solutions from a single vendor, which often led to vendor lock-in, and kept you from taking advantage of innovations happening across the broader infrastructure ecosystem.

Moreover, if you hand over control of your cloud infrastructure to a single vendor, you're betting that all the innovation you'll ever need will come from that single source, at a price that is better than any of the alternatives. And you're also assuming that this single vendor is in the best position to judge what best fits your interests.

Contrast that with the market merits of an open approach like OpenStack. Because multiple vendors compete to implement the standard interfaces, you are in a position to choose which vendor's infrastructure technology is best suited to the business problems you turned to cloud for in the first place. Moreover, you don't sacrifice the opportunity to seek an unfair advantage for the applications and workloads you run just because they are locked in to a particular vendor's platform. In economic terms, OpenStack lowers your switching costs, availing you of credible substitutes that you choose from, rather than having to commit to a single vendor in advance.

OpenStack is the glue that ties cloud computing infrastructure components together. Pure-play OpenStack distributions don't rely on a particular operating system, hypervisor, storage, or networking fabric. And because of the open source model, innovation accelerates as the community contributes ideas and code.

Moving Forward With OpenStack

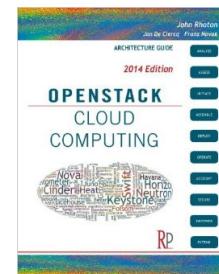
While IaaS can in fact support many kinds of applications and workloads, where it really shines is with workloads and applications designed for the cloud. Native cloud applications are flexible, resilient, scalable, and remotely accessible. In fact, the most important question you can ask in preparing to move to OpenStack is not “What can OpenStack do for me?”, but “What can OpenStack do for my applications and workloads?”

This EBook, a complete excerpt of Rhoton’s [OpenStack Cloud Computing: Architecture Guide](#), explains how the core attributes of cloud computing—virtualization, delivery “as-a-service,” elasticity, flexible billing, universal accessibility, simplified management, affordability, multi-tenancy, and service-level management—can benefit your business. From there, Rhoton makes practical recommendations on how to identify cloud opportunities and evaluate them, including building business cases, performing risk analyses, and developing technical designs—to prepare you for your OpenStack journey. With this perspective in hand, we believe you’ll be in a better position to take advantage of the transformative value of OpenStack as the platform for your cloud applications and services.

OpenStack Cloud Computing

By John Rhoton

*with contributions from:
Jan De Clercq and Franz Novak*



An Excerpt from **OpenStack Cloud Computnig**

RP Recursive Press

OpenStack Cloud Computing

By John Rhoton

Copyright © 2014 Recursive Limited. All rights reserved.

Recursive Press is an imprint of Recursive Limited.

The RP logo is a trademark of Recursive Limited.

Published simultaneously in the United States and the United Kingdom.

ISBN-10: 0-9563556-8-4

ISBN-13: 978-0-9563556-8-3

British Library Cataloguing-in-Publication Data

Application submitted.

All product and company names mentioned herein are property of their respective owners. Where those designations appear in this book, and Recursive Limited was aware of the claim, the designations have been printed in all capital letters or initial capital letters. Neither the publisher nor the authors or reviewers of this book have any affiliation with the trademark owners. The trademark owners do not endorse or approve the contents of this book. Readers should contact the trademark owners for more information regarding the trademarks.

The author and publisher have taken great care in the preparation of this book, but do not imply warranty of any kind. No warranty can be created or extended through any promotional activity. The publisher and author assume no responsibility for errors or omissions, nor do they assume liability for damages resulting from the use of the information contained in this book. Any recommendations implied in the book may not be applicable in every situation. The book is sold with the understanding that the author and publisher do not render legal or accounting services and that the reader should seek professional advice prior to engaging in any activity based on the content described in the book.

All rights reserved. No part of this book may be reproduced, in any form or by any means, stored in a retrieval system, or transmitted across any media without explicit written permission from the publisher.

Revision: 20150108131004

Preface

This book examines the deployment of cloud-based architectures using OpenStack technologies. Our overriding objective is to provide a comprehensive picture of the primary design considerations and implementation options. Our focus is not a high-level overview of cloud computing, so we do not elaborate on the business benefits of the technology. There are numerous books on the market that cover these topics, including Cloud Computing Explained and others listed in the bibliography.

This is also not a book about OpenStack technologies. Rather it is about implementing a cloud architecture based on OpenStack. The distinction may seem subtle, but it does have a significant impact on the scope of the material. There are some areas of cloud computing that OpenStack does not directly address. And there may be functions that it does cover but not as well as complementary products and services.

We have tried to put as much substance to the concepts as possible by including a number of references to vendors and service providers that are active in cloud computing today. This is not an endorsement of any particular solutions nor do we attempt to weigh strengths and weaknesses in the products. We may omit important information or characterize the services in different terms than the vendors. Therefore, we would encourage you to perform your own research before deciding on a particular service or eliminating it from your list of options.

Obviously, this text is not an exhaustive survey of all the available tools that might be used to supplement OpenStack. Given the volatile nature of the cloud landscape, we cannot even imply a guarantee that companies mentioned in this book will still be in business when you read about them or that they will offer the same functionality.

Nonetheless, we believe that you will get a better picture of what is happening in cloud computing with some actual examples and rough descriptions of the services currently on offer in the market place – if for no other reason than to give you a starting point for your own analysis.

AUDIENCE

You can look at the deployment of any IT system from many different perspectives, which depend on the operational role of the reader and the technical charter of the organization implementing the technology.

This material caters primarily to consultants, architects, technologists and strategists who are involved with the planning and implementation of information technology in large enterprises. Our background is heavily biased toward international corporations and the challenges they face in implementing new technologies.

However, most of the contents of this book will apply to the full spectrum from service providers to small and medium businesses. Indeed, one of the effects of cloud computing is to remove some of the artificial segmentation barriers that differentiate larger and smaller organizations.

There are many stakeholders who are involved in implementing new projects and who might be affected by a completely overhauled service delivery model. The chief executive officers, and others on the executive board, may be concerned about the strategic impact of cloud computing. The IT managers must plan the portfolio. The architects need some background to design the end-to-end system. The technologists require a starting point for a deep technical analysis that will support them as they implement and support the actual infrastructure.

In addition to the organizational role of the reader, it is vital to recognize that each topic will take on its

own flavor depending on the vantage point of the practitioner, whether it is the cloud consumer, cloud provider, application developer or regulator. In each case, it is worthwhile for the stakeholders to appreciate the challenges of the others. We have written this book primarily from the standpoint of the business customer, but have endeavored to include the most important elements of the other viewpoints too.

Each perspective is unique but, nonetheless, critical to the success of the overall objectives. We provide as much insight as we can for each viewpoint. This may mean that some sections will be less interesting for some readers. But we hope there is some value for everyone. To keep the discussion simple, we have placed the enterprise at the center of all our explanations. This means that when we mention customers and suppliers without any further qualifications, we mean enterprise customers and suppliers.

ORGANIZATION AND STRUCTURE

This book is structured as twenty-five chapters divided into ten parts:

Analyze - Any analysis begins with a systematic assessment. This section examines the notion of cloud computing and its delivery layers (SaaS, PaaS, IaaS) as well as common delivery models (private, public). We then proceed to look at how cloud is typically adopted in the enterprise starting with virtualization and automation to flexibility in sourcing its services.

Assess – Before implementing OpenStack, it is worthwhile to look at the alternatives. There are many commercial offerings for each type of cloud from IaaS to SaaS. The choice of OpenStack depends to a large extent on the value proposition of an open-source infrastructure service that caters to both private and public service providers. But there are also other open-source frameworks that are almost directly comparable, so the selection process should also consider them.

Initiate – The first step in getting started is to construct a clear picture of how the system should work. This means getting the system working in a pilot scenario with a minimum set of standard components. But you also need to make sure that you will eventually be able to address your requirements and integrate with your legacy environment. You might need more complex topologies or you may need to create linkages to additional components or ecosystems.

Assemble – The design of an OpenStack-based solution begins with the OpenStack services themselves. While it is possible to replace the individual modules, it is generally a good idea to start with the base solution and see to what extent it meets the business requirements. In particular, the core components of an infrastructure service include compute, storage and networking.

Deploy – After the initial design and implementation work is complete, you may have demonstrated the feasibility of the technology but that is a far cry from ensuring it will work in production, particularly for highly scalable workloads. The first task is to roll out the OpenStack software itself onto the bare machines in the data center. The second is to design the orchestration of the workloads so that they are able to launch easily and automatically.

Operate – Once deployed, the administration chores begin. On the one hand, there are proactive tasks to set policies, re-allocate resources and tailor the configuration of standard services based on user needs. On the other hand, it is also important to detect any unforeseen events. We must also keep an eye on trends in order to detect and resolve issues as they occur and to project where future problems may arise in order to prevent them.

Account – Financial governance is a top concern of almost every business. It relies on ensuring visibility of what activities generate expenses and what trends these cost drivers are projecting. Whether the charges are invoiced to external parties, cross-charged to internal departments or merely reported to

show value to the business, the numbers are critical in sustaining a compelling business case.

Secure – OpenStack itself is neither particularly secure nor insecure. Security is a discipline that requires systematic application. This means the first task of a risk analysis is simply to make sure all the components are implemented securely. After verifying that the configuration adheres to best practices, it is important to be vigilant of any newly found exploits and supplement the bare infrastructure with further layers of security. Other than the base infrastructure, a key component of the overall security model is identity and access management and the enforcement of consistent policies governing user activity.

Empower – One intent of cloud computing is to create an environment that maximizes the benefits of economy of scale. At some point, it may reach a size where failures are inevitable. The most effective solutions will not attempt to prevent them at any cost but rather ensure that the infrastructure and applications are able to withstand these through their high level of redundancy and automated self-healing. A parallelized architecture also enables auto-scaling which reduces the human effort required when load changes. Finally, autonomous operation requires the reduction of dependencies on other vendors or technologies and products.

Extend – Getting the software deployed and working efficiently in production is not the end of the journey. Technology and markets are in constant evolution making it necessary to perpetually adapt. But beyond these externally imposed changes, it is always possible to improve business value by building and extending the infrastructure. Moving up the cloud stack into platforms will drive increased efficiencies for new workloads. Analytics allows IT to generate more business value. And any improvements in the underlying software will help to support new business initiatives and give additional impetus to the community that is building it.

The first two parts are included in this excerpt. The remainder follow the same pattern and can be found in the [complete book](#).

Most chapters begin with a general overview of the challenges that any cloud deployment faces. Some readers with a strong background in cloud computing will find the topics familiar and may want to skim these sections. Nonetheless, we found it useful to include them because many readers will not previously have looked at the topics in a systematic fashion.

Using this baseline, we then show which functions OpenStack fulfills and how a typical deployment may implement them. In some cases, we complete the picture by elaborating on how to supplement the technology with other tools and processes.

FEEDBACK

A direct consequence of the print-on-demand model we have used for this book, and actually one of its primary benefits, is the flexibility it gives the author and publisher to incorporate incremental changes throughout the publication lifecycle. We would like to leverage that advantage by drawing on the collective experience and insights of our readers.

You may find errors and omissions. Or you may actually find some parts of the book very useful and interesting. Regardless of your feedback, if you have something to say then we'd love to hear from you. Please feel free to send us a message at:

john.rhodon@gmail.com, jan.declercq@hp.com or franz.novak@hp.com.

We can't guarantee that we will reply to every message but we will do our best to acknowledge your input!

ACKNOWLEDGEMENTS

You can find a large part of the information contained in this book on the public Internet. The OpenStack documentation covers many details of OpenStack and there are numerous blogs and other valuable resources that supplement it with practical advice. We have filled some of the gaps and tied all the pieces together, but we have tried to recognize the original sources where possible, both to give them credit and to make it easier for you to dig deeper should you wish.

We have received considerable help from Gill Shaw, who provided excellent proofreading and copy editing assistance. We would also like to acknowledge Elisabeth Rinaldin, who contributed to the design of the cover and layout.

A number of subject-matter experts and reviewers have provided valuable technical input, including: Patrick Joubert, David Fishman, Nicholas Chase, Kirill Ishanov, Jay Chaudhury, Sanjay Mishra, and Nick van der Zweep. We applaud the many sources listed at the end of the book, which have helped us immensely as we have dived into the details of the topics we have presented. Last but not least, we would like to point out that there would be no content to describe without the vision and creative talent of engineers at Rackspace, Red Hat, IBM, HP, Mirantis and other contributors to OpenStack.

We would also like to point out that some of the sections in this book were first published by IBM developer Works³.

3

<http://www.ibm.com/developerWorks/>

ANALYZE

Any analysis begins with a systematic assessment. This section examines the notion of cloud computing and its delivery layers (SaaS, PaaS, IaaS) as well as common delivery models (private, public). We then proceed to look at how cloud is typically adopted in the enterprise starting with virtualization and automation to flexibility in sourcing its services.

CHAPTER 1

Study the Cloud

OpenStack is currently the most popular consortium-led Infrastructure as a Service (IaaS) software stack. It was initiated by Rackspace Cloud and NASA. Since its founding, it has seen wide industry endorsement and now numbers over one hundred supporters, including many of the industry's largest organizations, such as IBM, AT&T, Canonical, HP, Rackspace, Red Hat and SUSE.

This book is centered on OpenStack. But before we dig into what OpenStack is, why it is important and how it works, let's take a step back and spend a couple of chapters making sure we are on the same page in our understanding of how it fits into the IT landscape.

There has been a lot of buzz about cloud computing the last few years. IT vendors have embraced the newest hype, which now dominates the computing landscape. Customers have found the concept fascinating and devoted significant resources to assessing its benefits and challenges.

Unfortunately, many have been deterred by what they have found. Even though its potential is largely undisputed, there are many obstacles that complicate an immediate deployment. According to most surveys, security concerns rank top of these lists.

By now, you probably have a good idea what the term "cloud computing" means – and perhaps even how it works. But, if you are responsible for protecting systems or applications, then a theoretical analysis of a new trend is only of limited use. Instead, you will want to understand the technical options and their potential impact on your business.

We have deliberately avoided spending much space on a foundational overview of cloud computing. If you are interested in a business-oriented perspective of the advantages and challenges to an enterprise implementation, then you may want to start with *Cloud Computing Explained* or one of the other introductory books listed in the bibliography. We will assume that you already have a good grasp of the basics and are ready to proceed to the next level.

That said, we did want to make sure that we are working from the same conceptual foundation. Your notion of cloud computing may be perfectly valid and yet significantly different from ours. To minimize any confusion and ensure a common framework and terminology, we will indulge in a brief characterization of what cloud computing means to us and describe the main services and delivery models.

Core Attributes

In the simplest sense, a cloud represents a network and, more specifically, the global Internet. Cloud computing, by inference, is the use of computational resources that are hosted remotely and delivered through the Internet. That is the basic idea underlying the term. It may be sufficient for your non-technical friends and colleagues, but shouldn't be adequate for anyone reading this book.

If you have ever tried to isolate the core meaning of "Cloud Computing" by looking for an authoritative definition, you will have quickly discovered that the term entails many different notions. There is some disagreement among the experts as to what constitutes the essence of this fundamental shift in technology. Some are able to articulate their perspectives more elegantly than others, but that doesn't mean they are

accepted any more universally.

The most commonly recognized definition in use today was articulated by the National Institute of Standards and Technology (NIST) (2011):

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Unfortunately, neither the NIST formulation, nor any interpretation of what it means, is universally accepted. A pragmatic approach to distilling the essence of the term is to examine the assortment of attributes of typical cloud solutions. This doesn't imply that every cloud attribute is essential to cloud computing, or that any combination qualifies a given approach as fitting the cloud paradigm. On their own, they are neither necessary nor sufficient prerequisites to the notion of cloud computing. However, the more of these attributes apply to a given implementation, the more likely others will accept it as a cloud solution.

If there is one element of cloud computing that can be considered a core concept, it is that of resource pooling. Generally, resources are shared across customers in a public environment and across departments or cost centers in a private implementation. The increased scale allows for better allocation and utilization, which contribute to additional benefits.

An informal survey of blogs and tweets, as well as published literature, on the subject reveals some other key components, including:

Off-premise: The service is hosted and delivered from a location that belongs to a service provider. This usually has two implications: The service is delivered over the public Internet, and the processing occurs outside the company firewall. In other words, the service must cross both physical and security boundaries.

Elasticity: One main benefit of cloud computing is the inherent scalability of the service provider, which is made available to the end-user. The model goes much further in providing an elastic provisioning mechanism so that resources can be scaled both up and down very rapidly as they are required. Since utility billing is also common, elasticity can equate to direct cost savings.

Flexible billing: Fine-grained metering of resource usage, combined with on-demand service provisioning, facilitate a number of options for charging customers. Fees can be levied on a subscription basis, or can be tied to actual consumption, or reservation, of resources. Monetization can take the form of placed advertising or can rely on simple credit card charges, in addition to elaborate contracts and central billing.

Virtualization: Cloud services are usually offered through an abstracted infrastructure. They leverage various virtualization mechanisms and achieve cost optimization through multi-tenancy.

Service delivery: Cloud functionality is often available as a service of some form. While there is great variance in the nature of these services, typically the services offer programmatic interfaces in addition to user interfaces.

Universal access: Resource democratization means that pooled resources are available to anyone authorized to utilize them. At the same time, location independence and high levels of resilience allow for an always-connected user experience.

Simplified management: Administration is simplified through automatic provisioning to meet scalability requirements, user self-service to expedite business processes, and programmatically accessible resources that facilitate integration into enterprise management frameworks.

Affordable resources: The cost of resources is dramatically reduced for two reasons. There is no requirement for capital expenditures on fixed purchases. Also, the economy of scale of the service providers allows them to optimize their cost structure with commodity hardware and fine-tuned operational procedures that are not easily matched by most companies.

Multi-tenancy: The cloud is used by many organizations (tenants) and includes mechanisms to protect and isolate each tenant from all others. Pooling resources across customers is an important factor in achieving scalability and cost savings.

Service-level management: Cloud services typically offer a service-level definition that sets the expectation to the customer as to how robust that service will be. Some services may come with only minimal (or non-existent) commitments. They can still be considered cloud services, but typically will not be “trusted” for mission-critical applications to the extent that others (which are governed by more precise commitments) might.

Service Models (SaaS, PaaS, IaaS)

One salient aspect of cloud computing is a strong focus toward service orientation. It is quite common to hear mention of it in conjunction with expressions like “anything, or everything, as a service” (XaaS). In other words, the cloud is not a single offering but instead an often fragmented amalgamation of heterogeneous services.

Rather than offering only packaged solutions that are installed monolithically on desktops and servers, or investing in single-purpose appliances, you need to decompose all the functionality that users require into primitives, which can be assembled as needed.

Unfortunately, it is difficult to aggregate the functionality in an optimal manner unless you can get a clear picture of all the services that are available. This is a lot easier if you can provide some structure and a model that illustrates the interrelationships between services.

The most common classification uses the so-called SPI (Software, Platform and Infrastructure as a Service) model (NIST, 2011). Amazon Elastic Compute Cloud (EC2) is a classic example of IaaS (Infrastructure as a Service). Google App Engine is generally considered to be a PaaS (Platform as a Service). And Salesforce.com represents one of the best known examples of SaaS (Software as a Service).

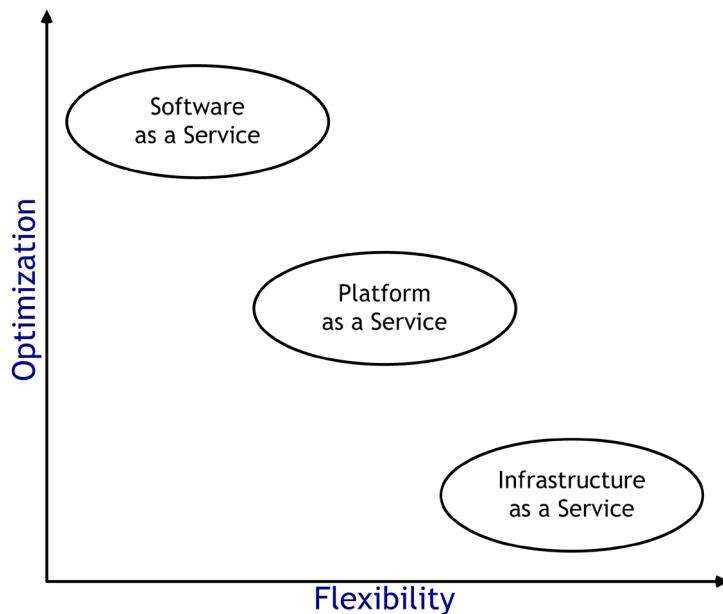


Figure 11: Software, Platform and Infrastructure Services

The three approaches differ in the extent of sharing that they provide to their consumers. Infrastructure services share the physical hardware. Platform services also allow tenants to share the same operating system and application frameworks. Software services generally share the entire software stack. As shown in Figure 11, these three approaches represent different tradeoffs in a balance between optimization, which leverages multi-tenancy and massive scalability, on the one hand, and flexibility to accommodate individual constraints and custom functionality, on the other.

The SPI model is a simple taxonomy that helps to present a first glimpse of the primary cloud-related services. However, as is often the case with classification systems, the lines are not nearly as clear in reality as they may appear on a diagram. There are many services that do not fit neatly into one category or the other. Over time, services may also drift between service types. For example, Amazon is constantly enhancing its AWS (Amazon Web Services) offering in an effort to increase differentiation and add value. As the product matures, some may begin to question if it wouldn't be more accurate to consider it a platform service.

SOFTWARE AS A SERVICE

Software as a Service is arguably the archetypical delivery model for cloud computing, attaining most of the claimed benefits. It benefits from the easiest enterprise implementation and is perhaps the most mature model. For those organizations wanting to focus on their core competencies it is the first place to look.

SaaS provides the full stack of cloud services, and ideally presents these to the end-user in a fashion that is not radically different from how users expect to use their applications. There may be some user interface changes that ripple through to the users, but the main difference is the deployment, licensing and billing model, which should be invisible to corporate end-users.

Consistent with the basic notion of cloud computing, SaaS is a model whereby the customer licenses applications and provisions them to users on demand. The services run on the provider's infrastructure and are accessed through a public network connection. Applications may be made available through the Internet as browser applications, or they may be downloaded and synchronized with user devices.

Some of the characteristics of SaaS services are that they are centrally managed and updated. Typically,

they are highly standardized although they may vary in their configurability as well as their efficiency and scalability. The most common pricing model is based on the number of users, but there may be additional fees based on bandwidth, storage and usage.

There are many similarities between SaaS and the services offered a few years ago by application service providers (ASPs). However, there are also stark differences in the approaches to multi-tenancy, the pay-as-you-go model and the ability to provision on demand.

SaaS offers several compelling benefits. It simplifies licensing. In fact, the customer doesn't need to acquire (or directly pay for) a software license at all. This is a task of the provider. There is also no need to calculate maximum capacity. It outsources the tedious task of application maintenance and upgrades and ties customer costs to usage, which lowers fixed costs and capital investment.

However, it does so at the price of restricting customer flexibility in terms of configuration options and update schedules. It also entails a significant commitment to the provider since it isn't trivial to switch from one SaaS vendor to another (or back on-site). There may be APIs for extraction and loading but there are no standards on the semantics of these interfaces, so it requires significant effort to automate a migration process.

PLATFORM AS A SERVICE

Cloud platforms act as run-time environments that support a set of (compiled or interpreted) programming languages. They may offer additional services such as reusable components and libraries that are available as objects and application programming interfaces. Ideally, the platform will offer plug-ins into common development environments, such as Eclipse, to facilitate development, testing and deployment.

There has been a marked increase in the number of web hosting services that support a variety of active server-side components, ranging from Microsoft ASP.NET and Java to scripts such as PHP, Python and Ruby on Rails. Compared to infrastructure services, these platforms reduce the storage requirements of each application and simplify deployment. Rather than moving virtual machines with entire operating systems, the application only requires the code written by the developer. An additional benefit is the increased ease for the service provider to sandbox each application by only providing functions that cannot disrupt other tenants on the same system and network.

Platforms may also offer further functions to support the developers, for example:

Integrated Development Environment to develop, test, host and maintain applications

Integration services for marshalling, database integration, security, storage persistence and state management

Scalability services for concurrency management and failover

Instrumentation to track activity and value to the customer

Workflow facilities for application design, development, testing, deployment and hosting

User Interface support for HTML, JavaScript, Flex, Flash, AIR

Visualization tools that show patterns of end-user interactions

Collaboration services to support distributed development and facilitate developer community

Source code services for version control, dynamic multiple-user testing, rollback, auditing and change-tracking

INFRASTRUCTURE AS A SERVICE

Infrastructure as a Service (IaaS) is the simplest of cloud offerings and the one most relevant to initial deployments of OpenStack. It is an evolution of virtual private server offerings and merely provides a mechanism to take advantage of hardware and other physical resources without any capital investment or physical administrative requirements. The benefit of services at this level is that there are very few limitations on the consumer. There may be challenges including (or interfacing with) dedicated hardware but almost any software application can run in an IaaS context.

We can divide IaaS services into three categories: Servers, Storage and Connectivity. Providers may offer virtual server instances on which the customer can install and run a custom image. Persistent storage is a separate service which the customer can purchase. And finally there are several offerings for extending connectivity options.

Usually all three are combined as part of a more complete infrastructure service.

Deployment Models (Public, Private, Community, Hybrid)

The previous section classified services according to the type of content that they offered. It can also be useful to examine the types of providers that are offering the services. In an ideal world, designed according to a service-oriented architecture, this distinction would not be meaningful. A service description should cover all relevant details of the service, so the consumer would be independent of the provider and therefore have no reason to prefer one over another.

Sadly, however, this is not the case. There are many implications in the choice of provider relating to security, governance, invoicing and settlement. It is therefore still very relevant to consider if the provider should be internal or external and the delivery should include an outsourcing partner, a community (such as the government) or a public cloud service.

PUBLIC CLOUD

In the earliest definitions of cloud computing, the term referred only to solutions where resources are dynamically provisioned over the Internet from an off-site third-party provider who shares resources and bills on a fine-grained utility computing basis. This computing model carries many inherent advantages in terms of cost and flexibility, but it also has some drawbacks in the areas of governance and security.

Many enterprises have looked at ways that they can leverage at least some of the benefits of resource pooling while minimizing the drawbacks, by only making use of some aspects of cloud computing. These efforts have led to a restricted model of cloud computing, which is often designated as a Private Cloud. In contrast, the fuller model is often labeled the Public Cloud. NIST (2011) have expanded on these two deployment options with the notion of Community Cloud and Hybrid Cloud.

Most experts would still consider Public Cloud as the quintessential paradigm for cloud computing. Nonetheless, the other options will be the focus of this book. They are not only rising in importance throughout the industry but are also a core element of the value proposition of OpenStack.

PRIVATE CLOUD

The term Private Cloud is disputed in some circles as many would argue that anything less than a full cloud model is not cloud computing at all but rather a simple extension of the current enterprise data center. Nonetheless, the term has become widespread, and it is useful to also examine enterprise options

that also fall into this category.

In simple theoretical terms, a private cloud is one that only leverages some of the aspects of cloud computing (Table 11). It is typically hosted on-premise, pools resources across departments, scales “only” into the hundreds or perhaps thousands of nodes, and is connected to the using organization through private network links. Since all applications and servers are shared within the corporation, the notion of multi-tenancy is minimized.

From a business perspective, you typically also find that the applications primarily support the business but do not directly drive additional revenue. So the solutions are usually financial cost centers rather than revenue or profit centers.

	PRIVATE	PUBLIC
Location	On-premise	Off-premise
Connection	Connected to private network	Internet-based delivery
Scale direction	Scale out (applications)	Scale up (users)
Maximum scale	100-1000 nodes	10 000 nodes
Sharing	Single tenant	Multi tenant
Pricing	Capacity pricing	Utility pricing
Financial center	Cost center	Revenue/Profit center

Table 11: Private and Public Clouds

Given the disparity in descriptions between private and public clouds on topics that seem core to the notion of cloud computing, it is valid to question whether there is actually any commonality at all. The most obvious area of intersection is around resource pooling. As mentioned earlier, resources are shared across customers in a public environment and across departments or cost centers in a private implementation. The increased scale allows for better allocation and utilization, which contributes to additional benefits.

Virtualization can also play a central role in both scenarios. By enabling higher degrees of automation and standardization, it is a pivotal technology for many cloud implementations. Enterprises can certainly leverage many of its benefits without necessarily outsourcing their entire infrastructure or running it over the Internet.

Depending on the size of the organization, as well as its internal structure and financial reporting, there may also be other aspects of cloud computing that become relevant even in a deployment that is confined to a single company. A central IT department can just as easily provide services on-demand and cross-charge businesses on a utility basis as could any external provider. The model would then be very similar to a public cloud with the business acting as the consumer and IT as the provider. At the same time, the security of the data may be easier to enforce and the controls would be internal.

A black-and-white distinction between private and public cloud computing may therefore not be realistic in all cases. In addition to the ambiguity in sourcing options mentioned above, other criteria are not binary. For example, there can be many different levels of multi-tenancy, depending on the scope of shared resources and the security controls in place.

There are also many different options an enterprise can choose for security administration, channel mar-

keting, integration, completion and billing. Some of these may share more similarity with conventional public cloud models while others may reflect a continuation of historic enterprise architectures.

What is important is that enterprises must select a combination that not only meets their current requirements in an optimal way but also offers a flexible path forward with the ability to tailor the options as their requirements and the underlying technologies change over time. In the short term, many corporations will want to adopt a course that minimizes their risk and only barely departs from an internal infrastructure. However, as cloud computing matures, they will want the ability to leverage increasing benefits without redesigning their solutions.

Regardless of whether the cloud is hosted internally or externally, it needs to leverage a great deal more than virtualization in order to achieve maximum value. There are a host of other improvements related to cloud computing ranging from fine-grained metering for usage-based cost allocation to rigorous service management, service-oriented architecture and federated access controls. An organization that implements these systematically has the most flexibility in selecting from private and public offerings or combining them for their business processes.

PARTNER DELIVERY

The distinction between internal and external delivery of cloud computing is not always clear. Depending on whether these delivery modes are determined on the basis of physical location, asset ownership or operational control, there may be three different perspectives on the source of a cloud service.

For the sake of completeness, it is also important to mention that there are more hosting options than internal/private versus external/public. It is not imperative that a private cloud be operated and hosted by the consuming organization itself. Other possibilities include co-location of servers in an external data center with, or without, managed hosting services.

A similar solution to a private cloud might be to enlist the services of an outsourcing partner, such as IBM, HP, or UNISYS. Since outsourcing is primarily an enterprise offering, the solutions provide a high degree of isolation and privacy. They also accommodate stringent service levels and an allowance for customization that is typically not available in the public cloud.

They will typically use some of the software and services of a private cloud. In addition, they often leverage a broad set of management tools and considerable experience in consolidation, standardization and automation. Their differentiation in these areas, along with a significant economy of scale, allows them to create a compelling value proposition to enterprises. They may be much more expensive than public cloud offerings due to the enhanced service levels and customization options they offer. Nonetheless, they represent significant potential for costs savings for many enterprises.

The implications of using an outsourcing partner for developing a cloud architecture need not be great. The same products and technical infrastructure could underpin a private or partner implementation. However, it is very important to ensure the contracts and service agreements are compatible with the preferred choice of development tools and run-time services. If the partner excludes these tools, for example in an effort to maximize standardization, then they are no longer viable. On the other hand, it is quite possible that a large outsourcing partner would be able to support a larger array of platform options than many customers could provide on their own.

In some ways, you can consider these “partner” clouds as another point on the continuum between private and public clouds. Large outsourcers are able to pass on some of their benefits of economy of scale, standardization, specialization and their point in the experience curve. And yet they offer a degree of protection and data isolation that is not common in public clouds.

COMMUNITY CLOUD

Another delivery model that is likely to receive increased attention in the future is a community cloud. It caters to a group of organizations with a common set of requirements or objectives. The most prominent examples are government clouds that are open to federal and municipal agencies. Similarly, major industries may have an incentive to work together to leverage common resources.

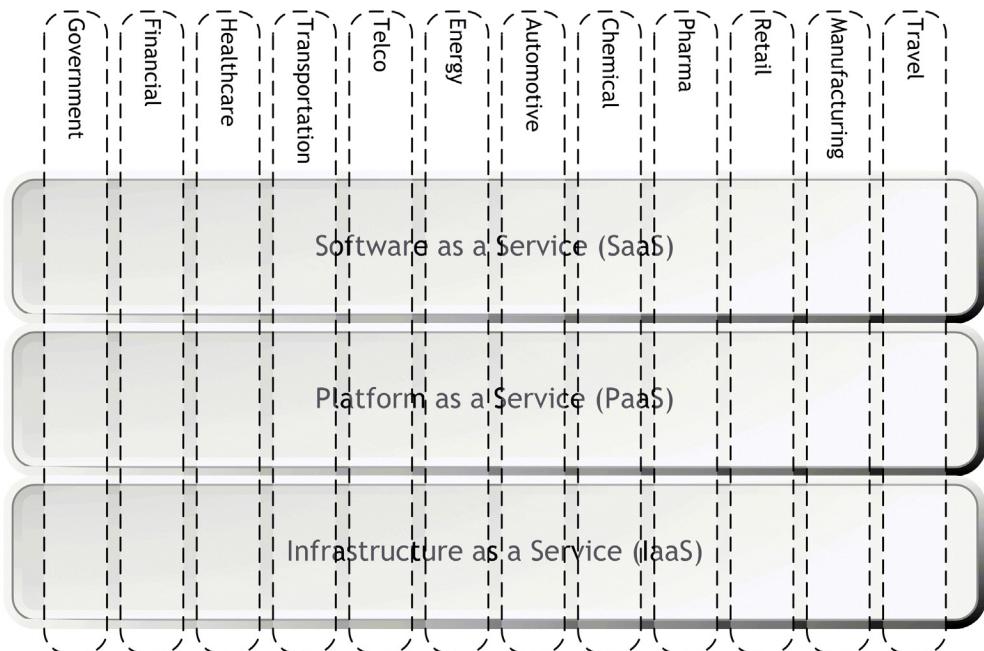


Figure 12: Community Clouds

The value proposition of a vertically-optimized cloud is initially based on the similarity of their requirements. Companies operating in the same industry are generally subject to the same regulations and very often share customers and suppliers who may impose additional standards and interfaces. A provider that caters to these specific demands can offer platforms and infrastructure with default service levels that meet all participants' obligations at a reasonable price.

However, the real benefits begin to accrue when a critical mass of industry players build a co-located ecosystem. When providers and consumers of software and information services are protected behind common security boundaries, and are connected with low-latency network links, the potential to share resources and data is greatly improved. The synergy that develops in this kind of cloud can spawn a virtuous cycle that breeds both additional functionality and efficiencies, and thereby allows the industry to advance in concert.

HYBRID CLOUD

The categorization of cloud providers in the previous section into private, public and community deployments is a great simplification. Not only is there no clear boundary between the three delivery models, but it is very likely that customers will not confine themselves to any given approach. Instead, you can expect to see a wide variety of inter-cloud constellations (Figure 13).

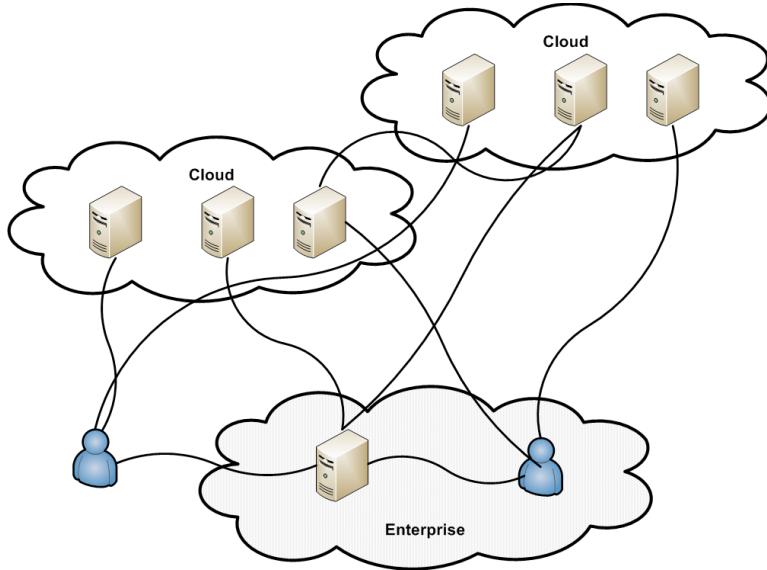


Figure 13 : Hybrid Delivery Model

Hybrid models can implement sourcing on the basis of at least four criteria.

Organizational: The simplest distinction would be that some business units use one source and other parts of the organization use another. This might be the case after a merger or acquisition, for instance.

Application: Another point of segregation would be the application. CRM, Email, ERP and Accounting may run from different delivery points for all applicable users in the organization.

Service: It is also possible that some services, such as Identity Management or a monitoring tool, are not immediately visible to the users but are transparently sourced from disparate cloud providers.

Resource: Virtual private clouds offer a means of extending the perimeter of the organization's internal network into the cloud to take advantage of resources with more elastic capacity than the internal systems. This extension is also invisible to end users.

Multi-sourced delivery models are inherently complex and therefore require careful planning. A framework such as eSCM (eSourcing Capability Model), developed by ITSqc, can be useful to ensure the design is systematic. It defines a set of sourcing life-cycle phases, practices, capability areas and capability levels as well as their interrelationships and suggests best practices both for the service providers and the customers who consume the services.

Value Proposition

A detailed analysis of the business case for cloud computing is outside the scope of this book. Nonetheless, it is vital to understand the basic value proposition of the technology in order to gauge the impact of its risks. Some of the primary benefits of cloud computing derive from improvements in cost, risk, security, flexibility, quality and focus. Let's look at each of them briefly.

COST

The most apparent advantages of cloud computing are around cost. In order to quantify the return, you will

need to perform a complete financial analysis of both a cloud model and any alternative options. There can be a significant reduction in up-front investment since there is no need to purchase extensive hardware infrastructure or software licenses. Instead you can align your costs to actual usage. This means that you can allocate costs to the contributing revenue much more easily and accurately.

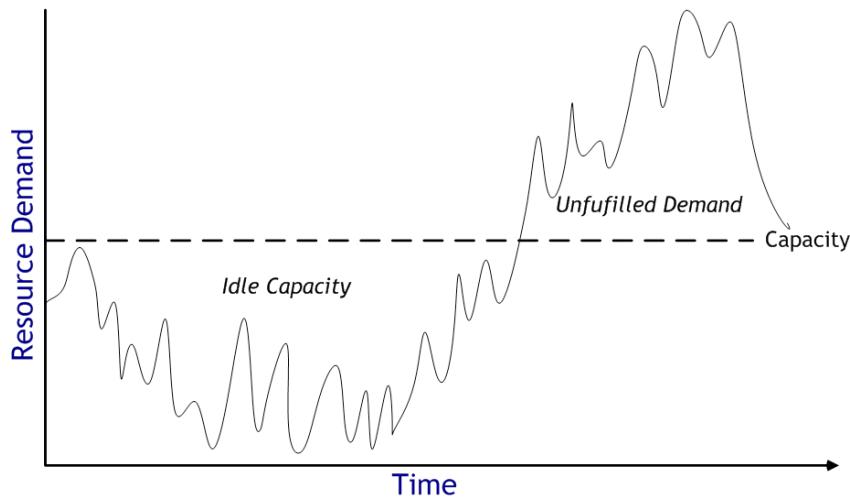


Figure 14: Fixed Capacity Utilization Curve

You also no longer need to over-provision resources in order to meet spikes in demand (Figure 14). High-end industry server utilization rates currently run at 15-20%. In the cloud, you do not pay for idle capacity which further reduces costs.

And finally, some benefits that the providers have acquired in terms of economies of scale and their place on the experience curve will translate into cost savings for the customer. Certainly, the providers will try to retain most of their advantage as profit; but, in a competitive environment with other efficient providers, you can also expect some savings to be passed on to customers.

RISK

Cloud computing can offload some risks from the customer to the service provider. By contractually stipulating data protection and disaster recovery provisions, and attaching them to indemnities in the case of failures, the company can mitigate its own risks.

It also reduces the likelihood of under-provisioning. Since it is not possible to accurately predict customer demand, there is always the possibility that there will be sudden unanticipated spikes of resource utilization. If the company owns its own resources, then there are limits to the amount of idle capacity that they will procure on the off-chance of a sudden increase in activity. On the other hand, the elastic capacity of a cloud provider should not often be exceeded.

It would be hard to over-emphasize this point. Scalability disasters can cause both direct and indirect costs. Lost revenues through unplanned downtime cost enterprises an average of over a hundred thousand dollars an hour and can exceed a million dollars an hour⁴ (Forrester 2004). In addition, there are numerous other consequences. The company may lose potential customers who are irked by the unpleasant experience of losing a transaction. Employees cannot work which increases their hourly costs. There may be compensatory payments. The brand damage can hurt relations with customers, suppliers, financial markets, banks, business partners and investors.

4

Estimates in U.S. dollars.

There may even be an impact on financial performance through interruptions in billing or investment activities. Revenue and cash flow recognition may be delayed and distort the financial picture and there are risks of lost discounts from accounts payable, which can also damage the credit rating. If that isn't enough, then consider the contractual payment obligations to temporary employees, schedules for equipment renewal, overtime costs, shipping costs and travel expenses, which can all be adversely impacted.

Rogue clouds represent another potential risk that an authorized cloud service can mitigate. Historically, when a technology is not deployed in an organization the likelihood of an unauthorized deployment increases. A stark example was that of WLANs (Wireless Local Area Networks). Companies that prohibited wireless technologies often found that employees were adding personal access points to the corporate network creating a huge attack surface. By implementing authorized WLANs, many organizations removed the incentive for unauthorized WLANs and were thereby able to control the risks more effectively.

Similarly, there is an incentive for many users or departments to leverage cloud-based services for personal and group use. It is extremely easy for them to access these on their own since many of the providers offer free functionality or credit-card-based payment. Their rogue use may jeopardize sensitive company information or expose the business to severe sanctions for non-compliance with industry regulations.

It is impossible to completely remove the threat of departmental cloud use. However, if the functionality is available on an authorized and supported basis, then the incentive for unauthorized and unmonitored usage declines.

SECURITY

Security is usually portrayed as a challenge for cloud computing, and rightfully so. Nonetheless there are several benefits that cloud computing may offer with respect to security. That is not to say that these benefits are necessarily exclusive to cloud computing, merely that they align very well with its implementation.

Cloud providers typically undergo very strict security audits. An enterprise may also institute the same audits but, on average, many businesses do not enforce the same level of rigor as a cloud provider. On a similar line, the cloud providers have access to the best-of-breed security solutions and procedures. They have been forced to inculcate a deep sense of security concern in their administrative staff. Again, this is not typically matched by smaller organizations.

The cloud also offers a platform for many security functions ranging from disaster recovery to monitoring, forensic readiness and password assurance or security testing. Its location makes it ideal for centralized monitoring. It is easier to isolate customer and employee data if they are managed in different environments. It might therefore increase security to segregate the data such that customer information is housed in the cloud while employee information is processed on the internal network.

Virtualization carries with it the inherent advantage that it is much easier to deploy preconfigured builds. It is possible to pre-harden these by locking down all traffic, eliminating unnecessary applications and features and applying the latest security patches.

There is some arguable advantage to the fact that the cloud obfuscates the physical infrastructure. Since virtual images may be brought up anywhere in the cloud and tend to move frequently, it makes it much more difficult for a hacker to launch a topology-based attack.

Finally, the constant presence of cloud services has an advantage in pervasive enforcement. Especially end-user devices tend to rely on connectivity to a central server for updates that address the most recent threats. Cloud resources are exposed on the Internet and therefore easily reachable from a wide variety of networks. Cloud services can also draw on the scale of security providers who gather threat intelligence

globally and share the newest threats and protection mechanisms in real time.

FLEXIBILITY

A cloud infrastructure adds considerable flexibility and agility to an enterprise architecture. It makes it much easier to roll out new services as they become necessary and to retire old applications when they are no longer needed. There is no need to procure hardware for the former or to cost-effectively dispose of the equipment in the case of the latter.

Similarly, a particular service can scale up and down as needed. There are cases where resource demand has spiked ten-fold overnight only to fall back to its original level shortly afterward⁵. The elasticity of a cloud allows the enterprise to exactly match the resources to the demand without overpaying for excess capacity or losing an opportunity to address market demand.

The flexibility also facilitates a faster time to market. When resources can be provisioned on demand, the usual lead time for procuring necessary equipment can be compressed to a few minutes. Ultimately, the speed and reduced commitment also lower barriers to innovation, which can encourage a more agile organizational culture.

A globally replicated cloud facilitates access from any place using any device at any time and therefore contributes to user flexibility and productivity. This advantage becomes even more visible when there is a need to integrate with business processes with suppliers, partners and customers. The absence of a firewall makes it easier to authorize fine-grained access to services and data without compromising or exposing other organizational assets.

QUALITY

Quality of service in all dimensions is a major concern around cloud computing. But in many cases, it is actually a benefit. Cloud service providers have great economy of scale and specialization. They have developed rigorous processes and procedures to maximize uptime and optimize performance. They run best-in-breed software to monitor and manage the infrastructure; and they employ some of the most skilled practitioners to oversee the management tools.

An on-demand model also differentiates itself from purchased and installed software in that the service provider can distribute new functionality and apply patches without any IT intervention. As a result, users can benefit from more frequent updates and newer functionality.

Cloud services also have the potential to deliver high availability since the provider's scale can offer multiple levels of redundancy with replication from the physical devices to the entire data center across large geographical distances.

FOCUS

The fact that some of the IT services are outsourced to a cloud provider reduces the effort and administration that is required by the corporate IT department. These responsibilities extend from user provisioning and support to application management and troubleshooting. Once service evolution is automated, experts can refocus on activities and opportunities that help to solidify the core competencies of the firm.

5

Animoto is a popular example: They scaled from 50 Amazon Servers to 3500 Servers in three days (16-19 April 2008)

Practical Recommendations

Cloud computing and, by implication, OpenStack come in many different flavors. They can be built privately or sourced publicly. They can deliver basic infrastructure or serve as the foundation for sophisticated platforms and applications. They can offer an exciting array of opportunities for information technology to demonstrate value to the business. However, they also present substantial obstacles, so their adoption requires careful planning.

In order to achieve long-term success, it is critical that customers identify opportunities for cloud computing as early as possible and submit them to extensive evaluation. They will need to build business cases, perform risk analyses and develop comprehensive technical designs, which all take considerable effort.

In many cases, the ultimate goal is not within immediate reach for financial, security or technical reasons. It is therefore necessary to plan an evolution through private and hybrid cloud computing before achieving a fully-fledged public cloud solution.

CHAPTER 2

Gauge your Maturity

Almost everything related to IT has been rebranded as “Cloud Computing” in the past few years. Many experts argue that the labels are merely an example of leveraging marketing hype to sell old technology. Without diving too deep into this very heated topic, let’s just say that adoption can come at varying speeds.

Cloud has a number of benefits, but it is also very disruptive. Large organizations cannot simply rip out their existing infrastructure and replace it with something entirely new, no matter how much better it might be. Instead, we need to look at the emerging trend as a journey more than a destination. Each step should take the customer closer to achieving the full benefits of cloud computing.

Enterprise Cloud Adoption Path

Service providers often have a highly standardized and homogenous data center. Rolling out cloud services and infrastructure is a logical extension of their existing operational model. This doesn’t mean it is easy, but it is feasible to implement broad upgrades and roll-outs as long as there is a compelling value proposition. Businesses, on the other hand, face a different set of challenges based on the complexity of their legacy environment.

While a hybrid model is the most likely end-point for many enterprises, a realistic look at the industry today reveals that we still have a way to go before we achieve it. It is not uncommon to find small startups today that are fully committed to cloud computing for all their service requirements. Large organizations, on the other hand, have been very cautious, even if they recognize the value that cloud computing can bring to them.

Corporate reluctance comes as no surprise to anyone who has followed the adoption path of emerging technologies over the past few years. Legacy applications, infrastructural investment, regulatory concerns and rigid business processes represent tremendous obstacles to change. Even if there are obvious early opportunities, the transition is likely to take time.

However, this doesn’t mean that enterprises are completely stationary. In their own way, most of them began the journey to a private cloud years ago and they are gradually evolving in the direction of a public cloud. We can break down this path by identifying three steps, which are each associated with an increasing level of efficiency.

Resource efficiencies are usually the first objective of a private cloud implementation. Standardization of components sets the scene for data-center consolidation and optimization. Each level of resource abstraction, from server virtualization to full multi-tenancy, increases the opportunity to share physical capacity, and thereby reduces the overall infrastructural needs.

Operational efficiencies target human labor, one of the highest cost factors related to information technology. Ideally, all systems are self-healing and self-managing. This implies a high degree of automation and end-user self-service. In addition to a reduction of administration costs, these optimizations also enable rapid deployment of new services and functionality.

Sourcing efficiencies are the final step and represent the flexibility to provision services, and allocate resources, from multiple internal and external providers without modifying the enterprise architecture.

This agility can only be attained if all systems adhere to rigorous principles of service-orientation and service management. They must also include fine-grained metering for cost control and a granular role-based authorization scheme that can guarantee confidentiality and integrity of data. On the plus side, the benefit of reaching this level of efficiency is that applications can enjoy near infinite elasticity of resources, and costs can be reduced to the minimum that the market has to offer.

Once the businesses have full sourcing independence, they are flexible in terms of where they procure their services. They can continue to obtain them from IT or they may switch to an external provider that is more efficient and reliable. However, this independence can work in both directions (Figure 21).

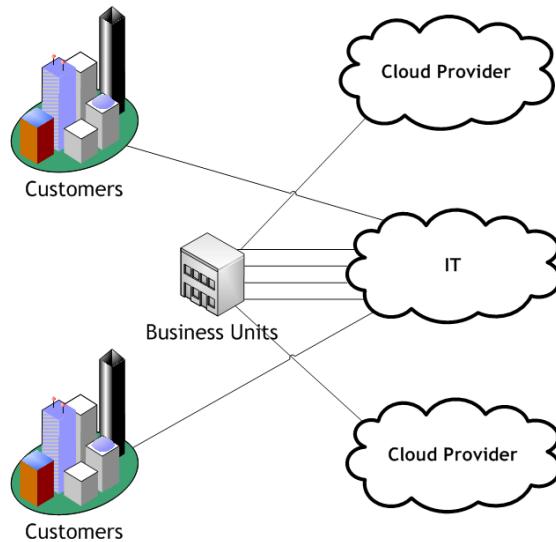


Figure 21 : Multi-source and Multi-target Services

If IT develops its services in a generic and modular form, then the organization also has the flexibility to offer parts of the functionality on the external market, and therefore monetize the investment in ways that were not possible before.

Software-Defined Data Center

As mentioned above, the SPI stack presents a simplified picture of how cloud services relate to each other. It facilitates the discussion by providing a common reference model. However, in reality, the lines are blurred, as many services (such as Identity Management) span multiple categories, and a complete solution involves additional components.

For example, in addition to the software and applications that run in the SPI model, and support a cloud application in its core functions, both the enterprise and service provider need to address core challenges, such as Implementation, Operation and Control, in order to successfully keep the solution going (Figure 22).

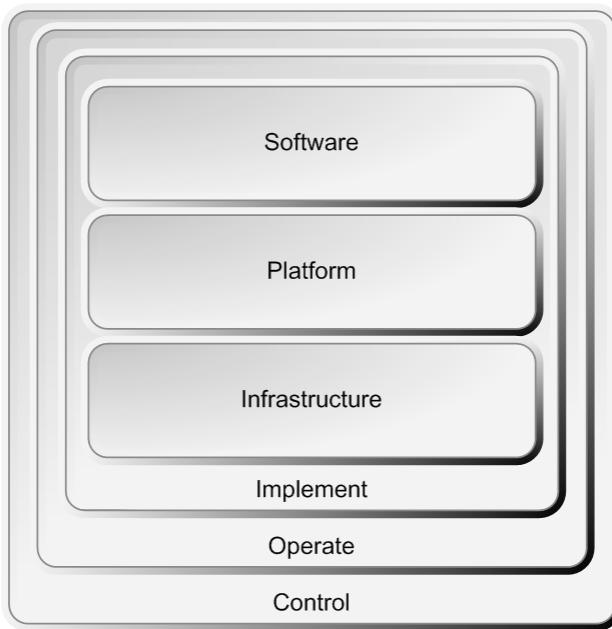


Figure 22: Implementation, Operation and Control

Implement: It is necessary to select and integrate all the components into a functioning solution. There are a large, and ever increasing, number of cloud-based services and solutions on the market. It is no simple task to categorize and compare them. And once that is done, it would be naïve to expect them all to work together seamlessly. The integration effort involves a careful selection of interfaces and configuration settings and may require additional connectors or custom software.

Operate: Once the solution has been brought online, it is necessary to keep it running. This means that you need to monitor it, troubleshoot it and support it. Since the service is unlikely to be completely static, you need to also have processes in place to provision new users, decommission old users, plan for capacity changes, track incidents and implement changes in the service.

Control: The operation of a complex set of services can be a difficult challenge. Some of the challenge may be reduced by working with solution providers and outsourcing organizations who take over the operative responsibilities. However, this doesn't completely obviate the need for overseeing the task. It is still necessary to ensure that service expectations are well defined and that they are validated on a continuous basis.

Fortunately, many of these capabilities are also in the process of being automated and delivered as services. The trend is to abstract the control plane (i.e. the administrative tools) from the service plane, where the workloads execute. This abstraction makes it easier to automate operational functions, such as provisioning, configuration and policy enforcement.

It is commonly referred to as the Software-Defined Data Center (SDDC) since it makes it possible to automate and flexibly deploy not only the workloads themselves but also the entire infrastructure that supports them. As with many emerging technologies, not everyone uses the SDDC label in the same way, but separation of control and service plane is usually a primary element in its definition. And this distinction is very relevant to the way we look at the data center.

In fact, we shall see that in many ways OpenStack is a means of implementing a software-defined data center and some of its core components, including Software-Defined Networking (SDN), Software-De-

fined Storage (SDS) and Software-Defined Compute (SDC).

Flexible Cloud Sourcing

One of the benefits of standardized cloud services is that the customer can easily and quickly change providers to optimize costs and functionality as the market and user requirements change. The broad industry adoption of OpenStack is a major advantage in this respect.

However, even with OpenStack, the current status is that not all implementations are interchangeable. Unfortunately, this incompatibility between providers makes the goal of seamless re-sourcing unrealistic without significant investment or complementary solutions. Nonetheless it is a big step in the right direction of reducing switching costs and we can expect the flexibility of moving from one provider to another to improve over time, either through increased standardization or the commoditization of brokering solutions.

HYBRID CLOUD DESIGN

Once an organization decides to embark on the journey of hybrid cloud, the task is not only to selectively source an increasing number of services from public providers. The bigger challenge is to integrate them dynamically with other services running internally and externally.

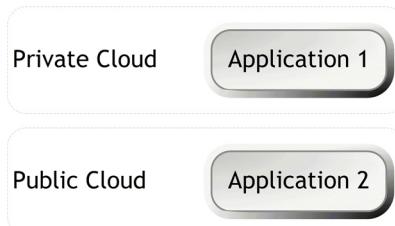


Figure 23: Hybrid Silos

To illustrate, consider some of the integration options for hybrid cloud computing. The first step of a hybrid cloud is for the services to run in independent silos without any interaction (Figure 23). For example, an organization might run Microsoft Exchange internally and use Salesforce.com as their publicly procured CRM service. If the two do not interact, this is not a difficult achievement.

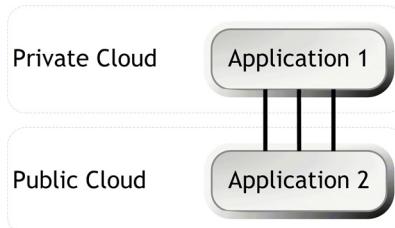


Figure 24: Static Hybrid Integration

The next step would then be to integrate them where possible (Figure 24). There might be a connection from Salesforce.com to the internal Active Directory for single sign-on or the service might leverage Microsoft Exchange to deliver email notifications and schedule tasks. This integration needs careful planning to ensure compatibility and safeguard any sensitive data.

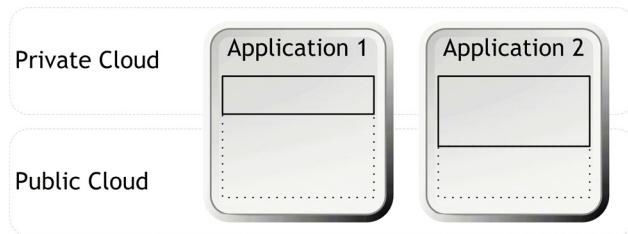


Figure 25: Dynamic Hybrid Integration

The last step is support for dynamic workload distribution (Figure 25). This means that there must be equivalent internal and external services and they need interface compatibility. One reason for this approach would be to establish a disaster recovery facility. In the event of catastrophic failure of the internal data center, the company could shift the workload to the cloud and restart the service there.

An even more ambitious goal is cloudbursting, which can be used to optimize costs and flexibility. If the organization is able to shift workloads in real-time, then it is possible to run services internally as a standard practice. However, if there are spikes in activity or the service grows faster than anticipated, the company can off-load any processing that exceeds its internal capacity.

Let's look at these two notions in more detail.

DISASTER RECOVERY

In Chapter 20, we will cover short-term resilience, which is the first line of defense for business continuity. If the environment is resilient, then there will be no need for an expensive and disruptive recovery. A large part of resilience relies on redundancy. If information, systems, processes, infrastructure and personnel are fully redundant, then the likelihood of an outage will be minimal.

In practice, it is usually not cost-effective, and sometimes not technically feasible, to ensure absolute redundancy.

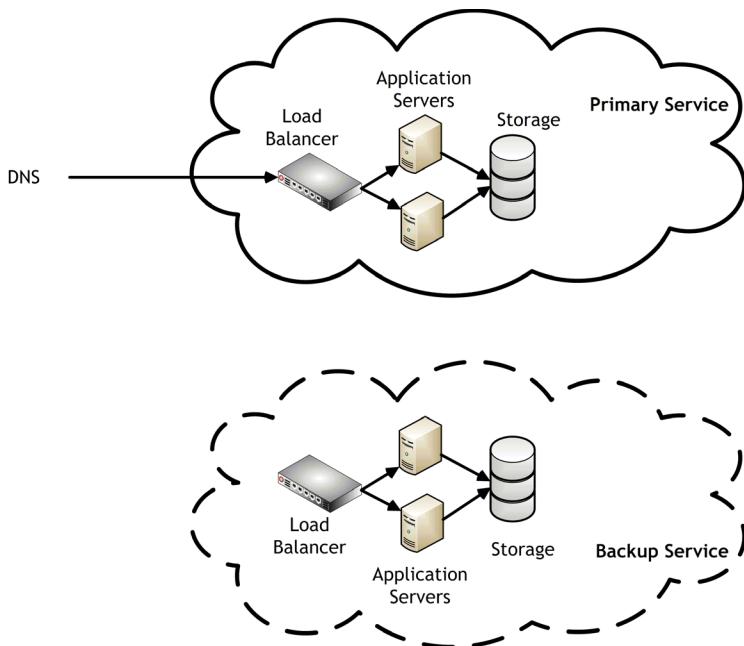


Figure 26: Cold Disaster Recovery

The simplest recovery scenario involves what is called a cold site (Figure 26). The entire environment is replicated to another location, which may be a cloud service. The service including both computational and storage instances is fully configured but it is not actually running. In the event of a disaster, there is a need to load the storage with current data and then activate the components.

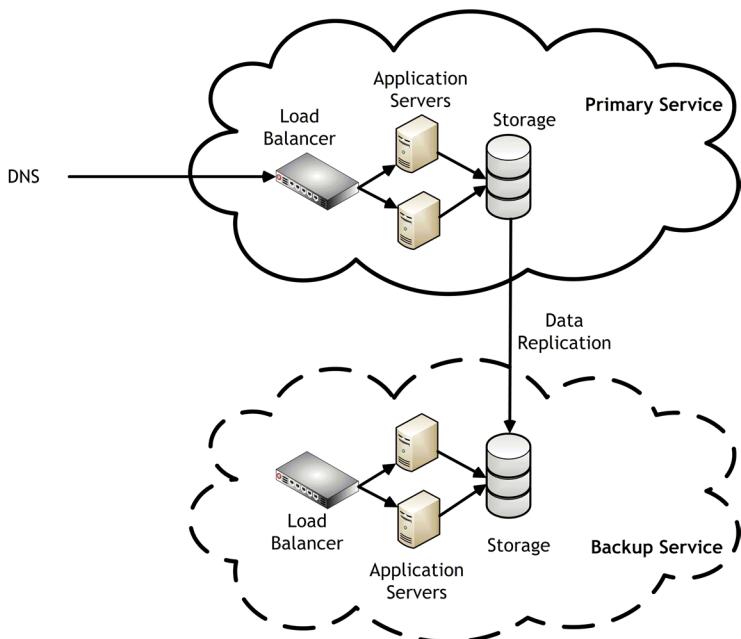


Figure 27: Warm/Hot Single-active Disaster Recovery

A more sophisticated option is a warm site. Its main technical difference is that the data is continuously replicated to the backup storage (Figure 27). This reduces the time it takes to launch the service when it is needed. In the case of a hot disaster recovery, the full service is also always operational, even if it is

receiving no traffic. When a disaster strikes, someone just needs to redirect the DNS entry to point to the backup solution and it will take over without any delays.

The disadvantage of this design, where only one service is active at any given time, is that it wastes valuable resources and can be very costly to implement.

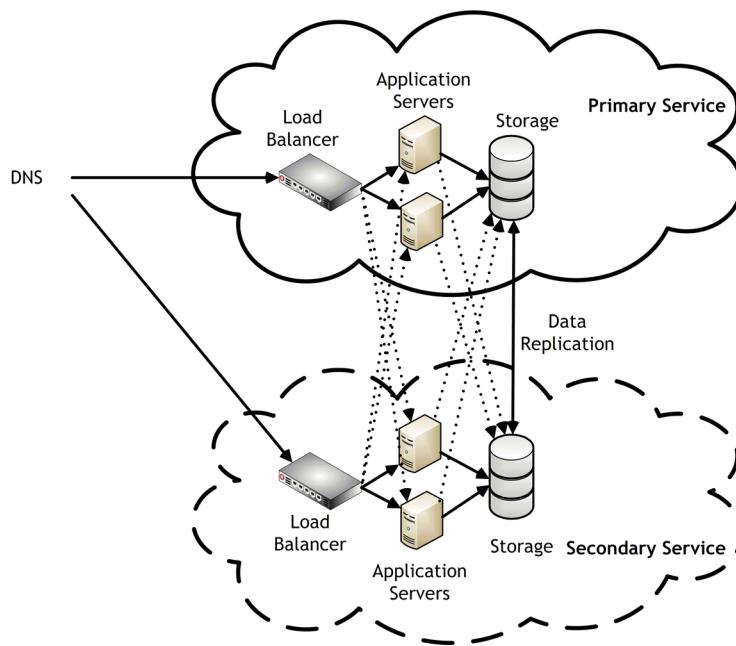


Figure 28: Dual-Active Redundant Services

In order to achieve greater efficiency, some organizations implement a dual-active scenario (Figure 28). In this setup, both services are fully operational and distribute the load between them. All end-user requests are shared, either through a global load balancer or DNS round robin. It is also vital that the storage be synchronized between the services to ensure consistency in the data.

In some cases, the entire architecture may be integrated so that the load balancers and application servers are pooled between data centers. The level of interconnection will vary depending on the solution and the infrastructure being used. It is particularly important to consider the cost and security ramifications of any network connections that need to traverse the public Internet.

After all, the service should ideally be replicated to a geographically remote location so that it is unlikely to be affected by the same disaster. The more distant the two locations are, the better the chances that the second will survive a disaster of the first. A different continent is therefore ideal, but that level of precaution may not be necessary in every case.

An additional level of independence can be achieved if a backup is hosted by a second provider. In this case, an incident (e.g. large-scale intrusion or malware attack) cannot easily affect both the backup and the primary system. Note that redundant systems are only effective if they can failover. Once they are set up, they should be tested during a "dry run" to ensure that they will switch over properly during an outage.

CLOUDBURSTING

An even more ambitious goal is cloudbursting, which can be used to optimize costs and flexibility as long as the organization is able to shift workloads in real-time. It is usually cheapest to run the services

internally as a standard practice. However, if there are spikes in activity or the service grows faster than anticipated, the company can off-load any processing that exceeds its internal capacity. Let's look at this concept in more detail.

When faced with static workloads, some large organizations can match and beat the costs of cloud services since they can achieve similar efficiencies through their own economy of scale. In these cases, the financial analysis requires a deeper study.

The calculation is fundamentally different for periodic workloads. If the resources are only used part of the time, then the customer bears the full cost with a fixed investment but only pays a fraction for a variably priced service.

Before we let the pendulum swing too far in the other direction, however, we need to consider the case of a private cloud. The fact that a particular service has an irregular usage pattern doesn't immediately lead to the need for a public service. If workloads are complementary, many organizations can achieve similar benefits through virtualization.

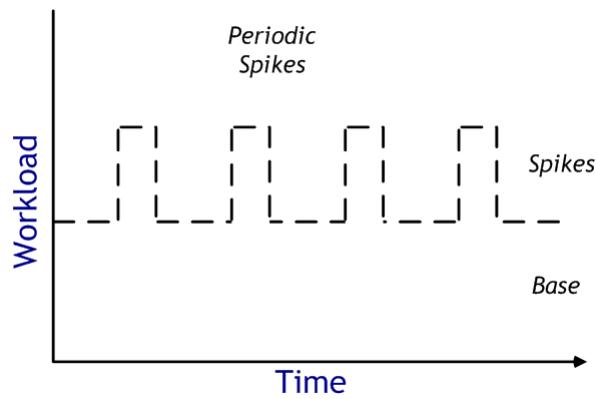


Figure 29: Periodic Incremental Workload

The picture in Figure 29 is typical for some workloads. Furthermore, when application consumption is aggregated through virtualization, almost all private clouds will manifest an uneven usage pattern over time. For organizations that are able to run a static workload more efficiently internally, the combined pattern presents a new opportunity.

Conventional wisdom would recommend an approach of “owning the ‘base’ and renting the ‘spikes’” (Weinman & Lapinski, 2009). For example, if the workload represents the sum of the two previous examples, then we would invest in an internal implementation of the minimum workload and source the additional periodic requirements from the public cloud.

Cloudbursting is the capability of the platform to call on other resource providers (processing, storage or networking resources) when internal resources run short and additional resources are available from other internal or external cloud platforms.

Enterprises may be able to satisfy some of their needs on premise, but still be interested in tapping into the cloud to handle peak loads for internal applications. This approach, often called ‘cloudbursting’ (Perry, 2008), involves extending an existing enterprise application, written on a private platform, to be able to leverage external services once internal capacity has been exhausted.

The most ambitious form of resource allocation exploits all available private assets, but combines these with the elasticity of public resources (Figure 210). When it is well designed and orchestrated, cloudbursting can minimize costs and maximize agility. However, while the approach sounds elegant in theory, it is

difficult to achieve in practice.

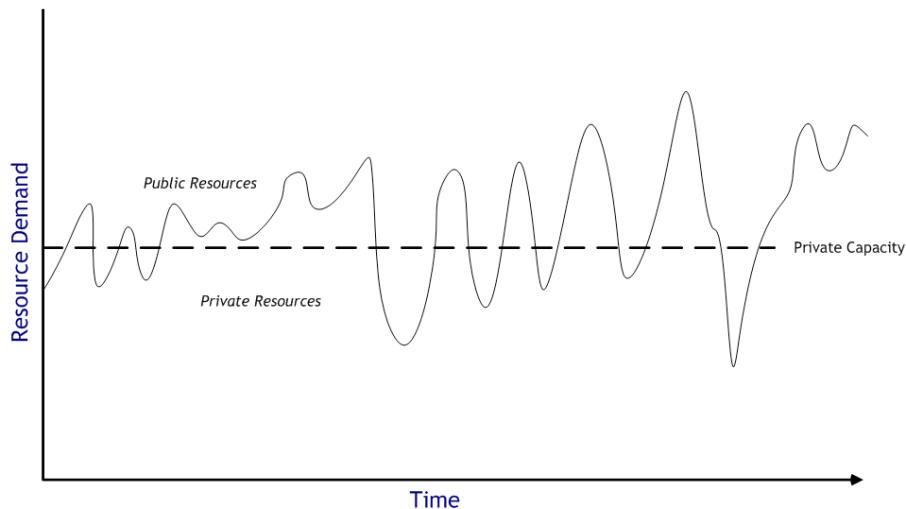


Figure 210: Cloudbursting

In order to shift load effectively, three prerequisites are needed: a trigger mechanism, capacity to launch public instances, and capacity to shift associated data. We will return to this topic in Chapter 21 to see how it might apply to OpenStack.

Practical Recommendations

The first steps toward an enterprise cloud rely on standardization, consolidation and virtualization. Beyond these, it is possible to achieve operational efficiencies and optimized sourcing options through a highly abstracted software-defined data center and well-defined service-oriented architecture.

OpenStack can be very helpful in implementing these phases with its extensible framework and standardized functions that facilitate hybrid operations, such as disaster recovery and cloudbursting.

ASSESS

Before implementing OpenStack, it is worthwhile to look at the alternatives. There are many commercial offerings for each type of cloud from IaaS to SaaS. The choice of OpenStack depends to a large extent on the value proposition of an open-source infrastructure service that caters to both private and public service providers. But there are also other open-source frameworks that are almost directly comparable, so the selection process should also consider them.

CHAPTER 3

Explore the Landscape

In Chapter 1, we gave a theoretical explanation of what cloud computing is and why it is so popular. But the concept is so broad and there is so much variety in the actual options that it only comes to life when we look at a few of the widely adopted services and technologies.

If you are interested in pursuing a cloud-based solution, you might ask yourself what delivery model you need. Should it be SaaS, PaaS or IaaS? You will also need to make a decision on a deployment model. Do you need a private, public or hybrid cloud?

OpenStack is best positioned as an infrastructure service, which can be consumed from either private or public sources. But it is not the only technology that you could use to build out your infrastructure. Before you build out your own service, you will be well advised to consider what is on the market. It may fully satisfy your requirements. And even if it doesn't, it will give you a baseline with which to compare your service and some good ideas on how to design your architecture.

This chapter will look at a cross-section of the commercially available offerings to consider as you make your selection. We will then look at the tradeoffs in the next chapter as well as what tools, including OpenStack, you might want to consider in creating a private or hybrid infrastructure service.

Software Services

One of the most popular and most publicized areas of Software as a Service is Customer Relationship Management (CRM) and in particular, sales-force automation. It includes functions such as account management, opportunity tracking and marketing campaign administration.

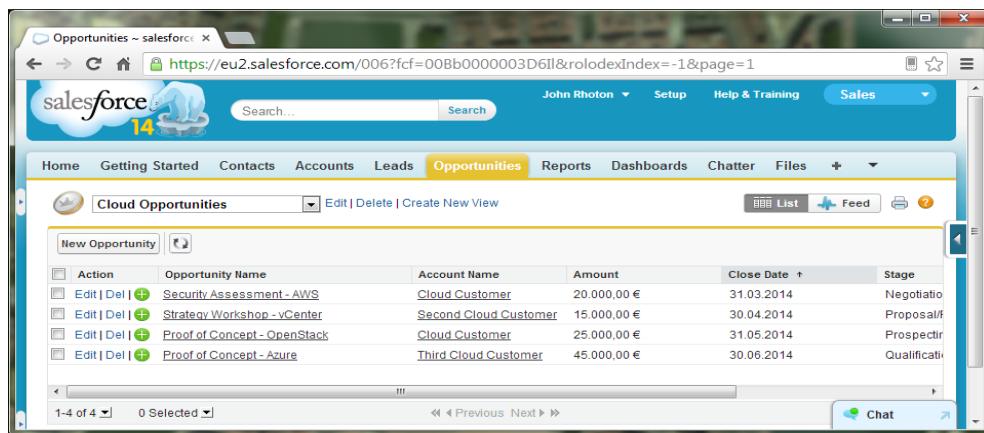


Figure 31: Salesforce.com

Arguably, the best known SaaS offering comes from Salesforce.com which provides a CRM solution consisting of several modules: Sales, Service & Support, Partner Relationship Management, Marketing, Content, Ideas and Analytics. It is available in over 20 languages and can be accessed from almost any Internet device including mobile platforms such as Android, iPhone and Windows Mobile.

NetSuite is another popular CRM package. Its base service is called NetSuite 2007.0 while NetSuite and NetSuite CRM+ are the two primary product options. Other options and services include NetSuite Global CRM, Dedicated Server Options, OpenAir and Payroll.

Human Resources (HR), or Human Capital Management (HCM), includes administration processes to support personnel functions such as recruiting, developing, retaining and motivating employees. Service providers include Workday and Taleo.

Workday, Netsuite, Intuit and others offer a variety of financial applications on-demand, ranging from accounting to procurement and inventory management.

Since collaboration involves establishing connectivity between people it is natural to also use a technology that is built on networking and utilizes a common infrastructure. There are a growing number of Web 2.0 services that are almost exclusively delivered over the Internet. But even some of the more traditional applications such as desktop productivity (e.g. from Google Apps, Microsoft Online Services and Zoho) and conferencing (Cisco Webex, Citrix GoToMeeting) can benefit from cloud computing.

Platform Services

Platform services lend themselves well for building new applications based on cloud-optimized infrastructure. In some cases, the boundary to infrastructure services is blurred, but platforms typically offer more programmer-oriented services, such as code libraries and development environments.

GOOGLE APP ENGINE

Google App Engine is one of the best known platform services. In addition to a basic run-time environment, it eliminates many of the system administration and development challenges involved in building applications that can scale to millions of users. It includes facilities to deploy code to a cluster as well as monitoring, failover, automatic scaling and load balancing.

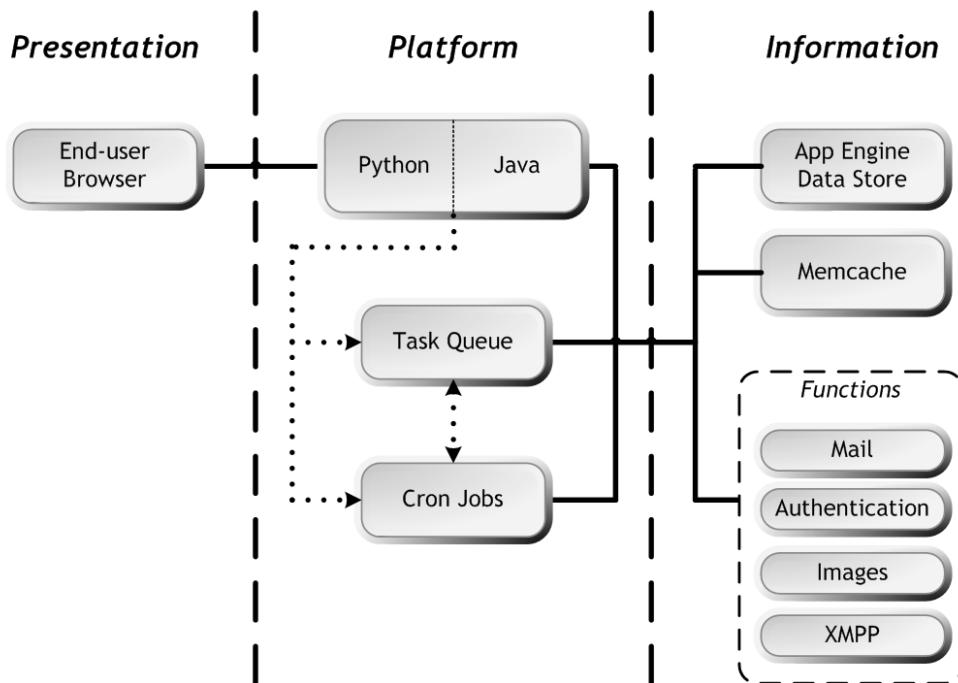


Figure 32: Google App Engine Architecture

Figure 32 offers a simple view of the Google App Engine Architecture. Users access the application through a browser, which connects to a hosted application written either in Python or Java. In addition to the runtime environment, batch jobs may run in the background either through a task queue or as scheduled “cron” jobs. The compute instances can access a persistent data store as well as a high-speed distributed cache.

The App Engine Datastore supports queries, sorting and transactions using optimistic concurrency control. It is a strongly consistent distributed database built on top of the lower-level BigTable with some added functionality. Unfortunately for legacy code, the App Engine Datastore is not like a traditional relational database. In particular, the datastore entities are schemaless. Two entities of the same kind are not required to possess the same properties, nor do they need to use the same value types if they do share properties. Instead, the application is responsible for ensuring that entities conform to any schema required by the business logic. To assist, the Python SDK includes a data modeling library that helps enforce consistency.

Google App Engine’s query language (called GQL) is similar to SQL in its SELECT statements however with some significant limitations. GQL intentionally does not support the Join statement and can therefore only accommodate single table queries. The rationale behind the restriction is the inefficiency that queries spanning more than one machine might introduce. However, Google does provide a workaround in the form of a ReferenceProperty class that can indicate one-to-many and many-to-many relationships.

MICROSOFT AZURE

Windows Azure is Microsoft’s Platform as a Service. Similar in concept to Google App Engine, it allows applications based on Microsoft technologies to be hosted and run from Microsoft data centers. Its fabric controller automatically manages resources, balances load, replicates for resilience and manages the application lifecycle.

The Windows Azure platform is built as a distributed service hosted in Microsoft data centers and built on a special-purpose operating system called Windows Azure. It is implemented as three components: Compute, Storage and a Fabric to manage the platform.

The **Compute** instances are exposed to the customer as role types that specify tailored configurations for typical purposes. The Web Role instances generally interact with the end user. They may host web sites and other front-end code. On the other hand, Worker Role instances cater to background tasks similar to Google App Engine cron jobs.

While Web and Worker role types are the most popular, Windows Azure provides additional templates for specific needs. For example, the CGI web role supports the FastCGI protocol and thereby enables other programming languages including PHP, Ruby, Python and Java. The WCF (Windows Communications Foundation) service is a web role that facilitates support of WCF services.

Azure **Storage** provides services that host three kinds of data:

- Blobs
- Tables
- Queue

A blob is simply a stream of unstructured (or at least opaque) data. It can be a picture, a file or anything else the application needs. There is a four-level hierarchy of blob storage. At the highest level is a storage account, which is the root of the namespace for the blobs. Each account can hold multiple containers,

which provide groupings (e.g. similar permissions or metadata). Within a container can be many blobs. Each can be uniquely identified and may hold up to 50GB. In order to optimize uploads and downloads (especially for very large files) it is possible to break a blob into blocks. Each transfer request therefore refers to a smaller portion of data (e.g. 4MB) making the transaction less vulnerable to transient network errors.

Tables are used for structured data. As you might expect from the name, they typically hold a set of homogenous rows (called entities) that are defined by a set of columns (called properties). Each property is defined by a name and type (e.g. String, Binary, Int). Despite the conceptual similarity there are important distinctions to make between Windows Azure storage tables and relational tables. Azure does not enforce a schema nor does it support SQL as a query language. While this may lead to portability challenges for many legacy applications, Microsoft's strategy is similar to that of Google and reflects the importance of ensuring the new technology is optimized for the scalability requirements of cloud-based services.

Queues provide a mechanism for applications to communicate and coordinate asynchronously. This is an important requirement when applications are geographically distributed over high-latency links. Synchronous communication can severely degrade performance and introduces stability risks that must be minimized. Like Blobs and Tables, Queues are associated with a Storage Account. They hold a linear set of XML messages. There is no limit to the number of messages per queue but they typically will be removed from the queue if they are not processed within seven days (or earlier if requested).

Similar to Amazon S3, it is possible to specify a region for any storage requirements in order to ensure compliance with local data privacy laws and minimize latency between applications and users. If performance is critical, the Azure CDN (Content Delivery Network) also provides distributed delivery of static content from Azure Storage from Microsoft's worldwide network of data centers.

The **Fabric**, in Azure terminology, refers to a set of machines running the Azure operating system that are collectively managed and generally co-located in the same region. The Fabric Controller is the layer of code that provisions all the user instances (web and worker roles) and performs any necessary upgrades. It also monitors the applications, re-provisioning and reallocating resources as needed to ensure that all services remain healthy.

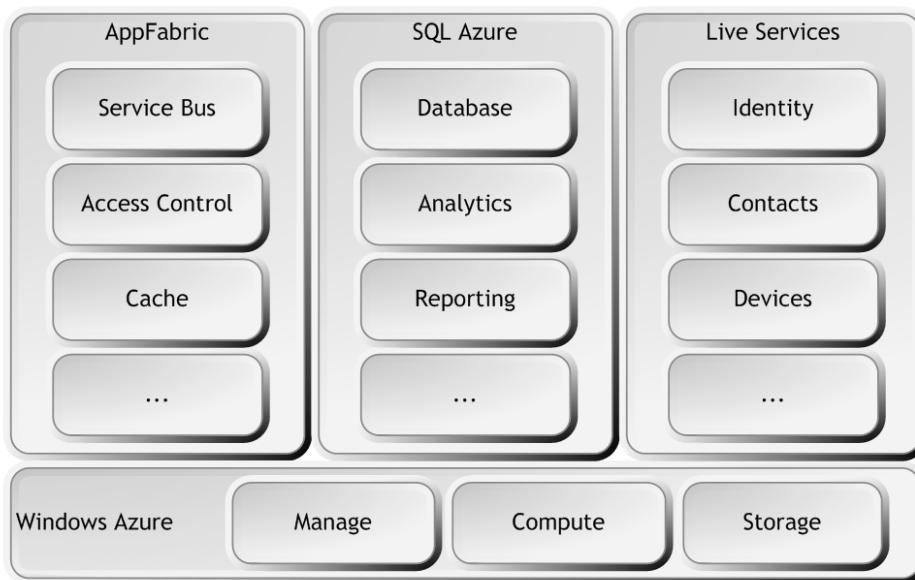


Figure 33: Azure Services

Azure also provides a set of services that can be consumed both from the Internet (including the Azure

platform itself) and on-premise applications. These include HDInsight, Backup, Cache, Messaging, Notification, Active Directory, and Multi-Factor Authentication.

OTHER PLATFORM SERVICES

Salesforce.com also delivers a Platform-as-a-Service which is called Force.com. It is very different from both Google's and Microsoft's offerings in this space. It does also offer hosting services based on its technology with the usual features of redundancy, security and scalability. But Force.com is much more data-oriented than code-oriented.

Engine Yard Cloud provides a Ruby on Rails technology stack, including web, application and database servers, monitoring and process management. The Linux distribution is optimized for Rails and includes in-memory cache.

Salesforce.com's Heroku is another cloud application platform for Ruby. It is a managed multi-tenant platform and hosting environment. Each service consists of one or more dynos, or web processes running code and responding to HTTP requests.

A key differentiator of Facebook applications is that they revolve around a "social graph", which connects people with other people and their interests. One way to conceptualize the potential of the Facebook Platform is by thinking of it in terms of a three-tier model: Presentation, Application Logic and Data.

Intuit offers a platform service, called the Intuit Partner Platform (IPP), which focuses on the development, sale and distribution of multitenant SaaS applications. Initially it consisted of an Adobe Flex-based development environment with functions to target the large installed base of QuickBooks users. It has since expanded into a set of APIs that can be leveraged from any platform, notably including Microsoft Azure, a strong partner of Intuit.

Pivotal One is an enterprise PaaS built on Cloud Foundry. It is an open-source platform that can also run on OpenStack, which is a topic we will cover again in Chapter 23.

Infrastructure Services

One distinguishing factor of IaaS, as opposed to PaaS or SaaS, is that there are many more private and hybrid options available in addition to the public services on the market.

AMAZON WEB SERVICES

The de facto standard for infrastructure services is Amazon. While they are not unique in their offerings, virtually all IaaS services are either complements to Amazon Web Services or else considered competitors to them. We therefore find it useful to structure the analysis of IaaS along the lines of what Amazon's offerings.

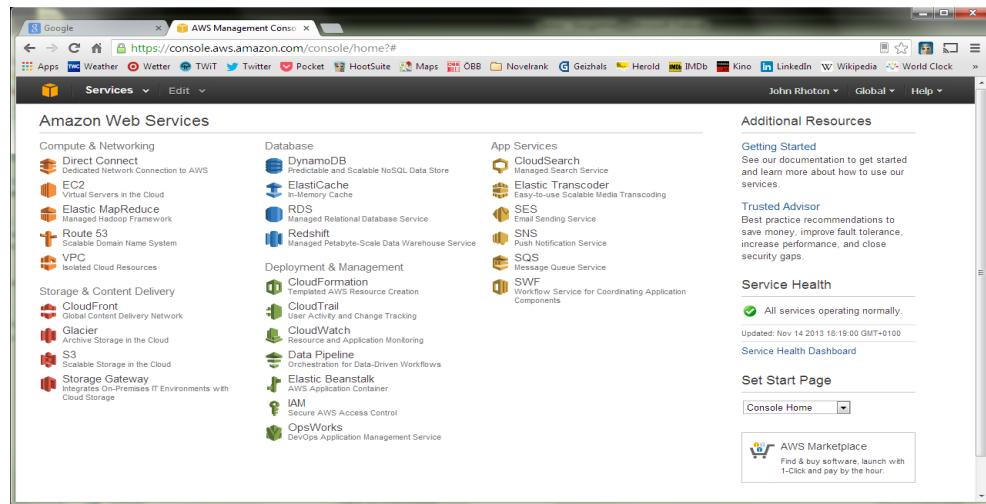


Figure 34: Amazon Web Services

Amazon EC2 (Elastic Compute) is the core service, which falls into the category of Shared Virtual machines, each based on an Amazon Machine Image (AMI). The customer can use pre-packaged AMIs from Amazon and 3rd parties, or they can build their own. They vary in resources (RAM, compute units, local disk size), operating systems (several Windows versions and many Linux distributions) and the application frameworks that are installed on them (e.g. JBoss, MySQL, Oracle).

Figure 35 gives an overview of some of the main services that are available from AWS. As you can see, the offering is very broad. There are many options for computation, storage, integration and scalability, not to mention functions for management and billing that are not even represented on the diagram.

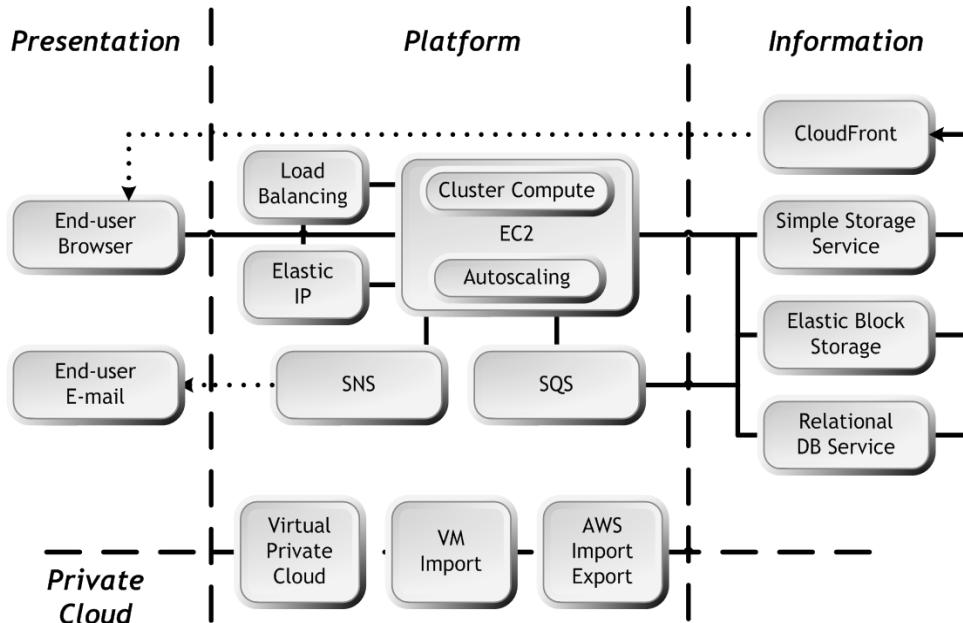


Figure 35: AWS Architecture

To help Amazon Web Services novices, we have tried to assemble the elements into one conceptual framework. However, although they are portrayed here in unified form, keep in mind that it is possible to consume almost every Amazon service independently of the others.

MICROSOFT AZURE VIRTUAL MACHINES

Microsoft Azure initially focused on platform services as described above. However, they have since introduced a service called Microsoft Azure Virtual Machines, which fits more closely to the characteristics of an infrastructure service comparable to Amazon EC2. It supports both Windows and Linux Virtual machines.

Additionally, the Windows Azure Virtual Network enables logical isolation in Windows Azure coupled with secure connections to on-premise datacenters. And the Windows Azure Drive allows applications to mount a blob which they can use to move VMs between private and public clouds.

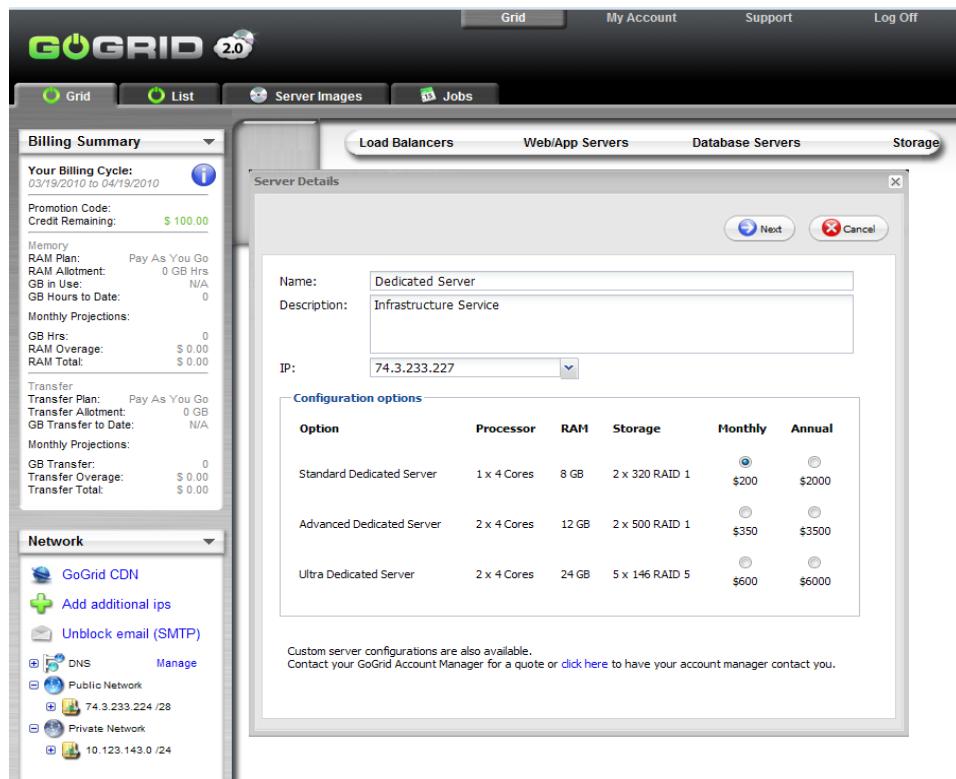
It is worth noting that Microsoft also has a strong private cloud offering based on System Center and Windows Server 2012. Since these products are based on the same underlying code as Windows Azure, it is relatively easy for customer to build hybrid environments that span both their internal data center as well as the Windows Azure public cloud.

GOOGLE COMPUTE ENGINE

Similarly, Google also extended its App Engine with compute capabilities in the form of Google Compute Engine, an infrastructure service that lets large-scale computing workloads run on Linux virtual machines hosted on Google's infrastructure.

As would be expected, there are also load balancing services, object storage accessible through a REST API, the capability to reserve static IP addresses, and customizable firewall rules defining access to external traffic.

GOGRID



The screenshot shows the GoGrid 2.0 control panel. At the top, there are tabs for Grid, List, Server Images, and Jobs. Below these are tabs for Load Balancers, Web/App Servers, Database Servers, and Storage. A 'Billing Summary' sidebar on the left shows a billing cycle from 03/19/2010 to 04/19/2010 with a promotion code and \$100.00 credit remaining. The main area is titled 'Server Details' and contains fields for Name (Dedicated Server), Description (Infrastructure Service), and IP (74.3.233.227). A 'Configuration options' table lists three server types:

Option	Processor	RAM	Storage	Monthly	Annual
Standard Dedicated Server	1 x 4 Cores	8 GB	2 x 320 RAID 1	\$200	\$2000
Advanced Dedicated Server	2 x 4 Cores	12 GB	2 x 500 RAID 1	\$350	\$3500
Ultra Dedicated Server	2 x 4 Cores	24 GB	5 x 146 RAID 5	\$600	\$6000

At the bottom of the dialog, it says: "Custom server configurations are also available. Contact your GoGrid Account Manager for a quote or [click here](#) to have your account manager contact you."

Figure 36: GoGrid Dedicated Server

GoGrid offers dedicated server configurations (Figure 36) as well as preinstalled images of Windows and Linux with Apache, IIS, MySQL and several other applications. It also provides free hardware-based load balancing to optimize the performance of customer instances.

FLEXISCALE

Flexiscale is an infrastructure service comparable to Amazon EC2 or Rackspace Cloud (discussed later). Similar to its competition, it supports Linux and Windows operating systems and facilitates self-provisioning of Cloud servers via the Control Panel or API. Customers can start, stop and delete instances and change memory, CPU, Storage and Network addresses of cloud servers.

They offer extensive firewall rules based on IP addresses and protocols, and each customer has their own dedicated VLAN. Unlike some IaaS providers, their virtual machines offer persistent storage, which is based on a virtualized high-end Storage Area Network.

CSC

CSC's cloud services consist of a multi-tiered approach in which CSC manages the complete ecosystems of cloud service providers, including platform-as-a-service, infrastructure-as-a-service and software-as-a-service. Orchestration helps clients manage data and collaborate across public and private networks.

These are complemented with a private cloud solution, called BizCloud, which runs on customers' premises behind their firewall, and BizCloud VPE, which is hosted in CSC data centers, but runs on isolated networks using dedicated resources.

VERIZON TERREMARK

Verizon became a major cloud player with its acquisition of Terremark, a global provider of IT infrastructure services. In addition to collocation, they offer VMware-based infrastructure services, such as utility hosting and enterprise-cloud virtual data centers. Terremark's data centers are built to stringent security standards with one location specialized for serving U.S. federal government needs.

The Enterprise Cloud from Terremark is an enterprise-class, Internet-optimized computing platform. A managed platform gives customers the ability to configure and deploy computing resources for mission-critical applications on-demand. The Enterprise Cloud gives control over a pool of processing, storage and memory resources to deploy server capacity. It's built around Terremark's Infinitructure utility computing platform, top-tier datacenters and access to global connectivity.

SAVVIS

Savvis is an outsourcing provider of managed computing and network infrastructure for IT applications. Its services include managed hosting, co-location and network connectivity, which are supported by the company's global datacenter and network infrastructure. Savvis offers enterprise customers three primary variants of its Symphony services: Dedicated, Open and Virtual Private Data Center (VPDC).

Savvis Symphony Dedicated is a fully dedicated virtualized compute environment that is hosted and managed in Savvis data centers. The solution can be partitioned into multiple self-contained virtual machines (powered by VMware), each capable of running its own operating system and set of applications. Once deployed, customers can add instances automatically through the SavvisStation Portal.

Savvis Symphony Open is built on a scalable, multi-tenant infrastructure and delivers a secure, enterprise-class cloud environment with built-in high availability and automated resource balancing. It uses a

purchase-by-the-instance cost model with flexible month-to-month terms for each instance.

Savvis Symphony VPDC introduces an enterprise-grade virtual private data center solution. Data center provisioning is facilitated through a self-service, web-based, drag-and-drop topology designer or through an application programming interface (API). The VPDC supports enterprise-grade security, platform redundancy, and high-performance information lifecycle management (ILM) storage as well as multi-tiered QoS levels with policy enforcement. A VPDC can contain a complete set of enterprise data center services, including compute instances of varying sizes, multiple tiers of storage, redundant bandwidth and load balancing.

VMWARE

VMware's cloud-oriented activity comes largely under the umbrella of its vCloud initiative. It represents a set of enabling technologies including vSphere, the vCloud API and the vCloud service provider ecosystem.

The **vSphere** platform, VMware's flagship product, is a virtualization framework that is capable of managing large pools of infrastructure, including software and hardware both from internal and external networks.

The **vCloud API** is a REST interface for providing and consuming virtual resources in the cloud. It enables deployment and management of virtualized workloads in private, public and hybrid clouds. The API enables the upload, download, instantiation, deployment and operation of virtual appliances (vApps), networks and "virtual datacenters". The two major components are the User API, focused on vApp provisioning, and the Admin API, focused on platform/tenant administration.

The **vCloud service provider ecosystem** is a common set of cloud computing services for businesses and service providers – with support for any application or OS and the ability to choose where applications live, on or off premise. It includes a set of applications available as virtual Appliances and is delivered by service providers, such as Savvis, T-Systems, AT&T, and Dell Cloud.

Beyond these infrastructure-oriented offerings, VMware vFabric offers a viable private platform service. It combines the Spring Java development framework with a set of integrated services including an application server, data management, cloud-ready messaging, load balancing and performance management.

JOYENT

Joyent offers infrastructure services for both public and private cloud deployments as well as their own web application development platform. For their infrastructure services, they use the term 'Accelerators' to refer to their persistent virtual machines. They run OpenSolaris with Apache, Nginx, MySQL, PHP, Ruby on Rails and Java pre-installed and the ability to add other packages. A feature called "automatic CPU bursting" provides reactive elasticity.

Joyent also offers a private version of their framework called SmartDataCenter for enterprise data centers. SmartDataCenter is software that runs on top of your existing hardware, or on new dedicated machines. It manages physical networking equipment and virtualized compute instances that are hosted within traditional physical servers and storage servers.

RED HAT

For those who resist the lock-in of a converged architecture such as Oracle Exadata, VCE Vblock, Dell vStart, HP CloudSystem Matrix and IBM PureSystems, the first port of call is open source. Red Hat, in addition to

being a major contributor to OpenStack, provides its own stack, which is not backed by any consortium but has the advantage of being open.

Red Hat Hybrid IaaS Solution removes the complexity of building a hybrid cloud with open source tools. All the software to enable cloud computing is included in one product. The solution lets customers create a hybrid cloud spanning both private and public cloud resources. Since the design is modular, customers can use Red Hat's included infrastructure-management tools or another vendor's management tools.

An IT-governed, self-service portal facilitates application deployment and policy-based usage gives control over which applications can be run, where they can run, who can use them, and how they should be optimized.

For those who need to build their own applications, Red Hat OpenShift Enterprise Platform-as-a-Service (PaaS) is an enterprise cloud application platform providing a development and execution environment for enterprise applications. OpenShift provides a preconfigured, auto-scaling, self-managing application platform in the cloud that lets developers develop, deploy, and run their applications.

RACKSPACE

Rackspace is another very well-known IaaS provider with many managed hosting options, in addition to virtual server offerings covering a number of Linux distributions (such as Ubuntu, Fedora, Centos and RedHat Enterprise Linux). It has a large pool of dedicated IP addresses and offers persistent storage on all instances.

Rackspace was originally called Mosso, which is a label still occasionally seen in reference to its services. As mentioned earlier, they are one of the co-founders of OpenStack and one of the biggest contributors. In fact the OpenStack Object Storage service (Swift) is derived from a Rackspace offering called Cloud Files.

In addition to their public infrastructure service, Rackspace also offers an on-premise version of their technology called Rackspace Private Cloud, which is similarly based on OpenStack.

HP CLOUD

HP is similar to Rackspace in offering both public and private versions of OpenStack technology. The HP public cloud is available at www.hpccloud.com and it includes most of the OpenStack projects. Additionally, HP has also aligned its converged infrastructure product suite with their OpenStack strategy. HP Cloud OS provides the foundation for a common architecture across private, public, and hybrid cloud delivery. It facilitates enterprise-grade OpenStack with optimized workload portability, enhanced service lifecycle management and simplified installation and upgrades.

IBM SOFTLAYER

IBM SoftLayer is a hosting environment running both bare metal and virtual servers on demand. It is one of the largest hosters offering services to small, medium and large businesses for Big Data, Disaster Recovery and web applications. Although it is currently based on proprietary technology, IBM has announced that SoftLayer will eventually transform into an OpenStack operation.

Practical Recommendations

OpenStack has its benefits, but it is neither the easiest nor the quickest cloud service to deploy. Any organization should begin its technology selection process by looking at what is available on the market and has already been successfully implemented in the industry.

There are countless software, platform and infrastructure services that cater to a variety of needs. Even if you choose to continue with OpenStack, you will have a better understanding of what is architecturally possible in the cloud after exploring these options.

CHAPTER 4

Make the Selection

The choice of OpenStack as the underlying technology for a cloud deployment relies on several decisions that enterprises (and their service providers) must make.

In a nutshell, these are:

- Keep it simple: start with the infrastructure
- Play it safe: minimize disruption and risk profile changes
- Maximize long-term flexibility: look for standard solutions
- Find the best fit: match external offerings to internal capabilities and requirements

Let's look at these in sequence.

Start with the Infrastructure

The first decision in adopting a cloud solution is what kind of service to select. As we saw in Chapter 1, most offerings can be classified as SaaS, PaaS or IaaS. The choice of which depends on the need of the business.

Figure 11 illustrated the primary tradeoffs between flexibility and efficiency. If we look at the three service models as a stack, then we can say that the higher we go, the more efficient we become. Since efficiency is one of the main benefits of cloud computing, it is usually desirable to go as high up the stack as we can.

Infrastructure services are relatively inefficient. Certainly they waste fewer resources than dedicated physical machines, but they do not allow tenants to share much more than the physical resources. Software services, on the other hand, have the advantage that they allow tenants to share not only the hardware but also the operating system and even the application. In the long term, we can expect most services to evolve towards the top of the stack where they can achieve the greatest efficiency.

However, flexibility is also an important factor. In this context, we don't mean flexibility in the sense of agility to scale up and down quickly. Instead it is the latitude to use the service for a variety of applications. In spite of their benefits in efficiency, an effective SaaS model is not trivial to implement particularly for a set of customers with greatly differing requirements. And software services usually only offer a very narrow spectrum of functionality with little, if any, ability for the customer to extend the feature set.

The advantage of infrastructure services is that they are much easier to deploy. Most applications that were designed for a x86 architecture will run in a virtual machine without any modifications. This makes IaaS a logical first step for organizations with significant investment in legacy software.

Leverage Private Assets

A second important consideration is whether to build or buy. Large organizations can develop and deploy

their own cloud services. But this doesn't mean that it is necessarily the best, or even the cheapest solution.

Service providers are typically in the business of implementing their own infrastructure. This is generally a core competence and their scale makes them attractive for highly scalable resource pools. If usage is volatile or unpredictable, it can be worthwhile for enterprises to consume the services externally.

The biggest drawback of a private cloud is that you don't automatically gain any benefits related to cloud computing. You can replicate many of the advantages internally, but it requires significant effort as well as a level of investment that may only be realistic for large enterprises.

The main financial considerations will be the scale of operation and the patterns of utilization. Typically large-scale deployments have a smaller unit cost which may be competitive with the services of an external provider. If the applications running on the infrastructure have a constant load, or the peaks and valleys tend to balance each other out across the portfolio, then the pay-as-you-go pricing model is not as compelling.

In addition to the costs, customers need to consider security and risk. External services rely on another organization, which may increase risk or insecurity or unavailability. However, the same risks also apply for an internal implementation. The main tradeoff is visibility and direct control versus trust and reliance on contractual agreements.

The choice to host your applications in a private data center gives the customer most control and flexibility. For example, after they install Apache or Microsoft IIS on the hardware of their choice with any necessary web frameworks, they can upload applications developed in their own environment.

They can choose any programming language and have complete freedom to implement any interfaces that may be necessary to connect to legacy or partner systems. It is only a matter of installing PHP, Python/Django, Ruby/Rails or a complete set of Java tools.

For those that do decide to implement a private cloud, there are tools, product and services that can be instrumental to attaining benefits in performance, utilization and automation. OpenStack is one of those options, but not the only one.

Maximize Flexibility with Open Source

For internal solutions, another critical question is whether to rely on their own code, purchase licenses of commercial software or leverage open source solutions. They may even want to combine elements of two or all three of these models to build the solution.

In the past, the case for open source was often considered to revolve around licensing costs. The implied guideline was that if you could afford it, commercial software was your best bet but, if you were on a tight budget, then open source might be an option.

However, software licensing is usually only a small portion of the full cost of a business service. In addition to the hardware and other infrastructure, you need to consider the human effort to deploy, manage and support the solution as well as the financial impact any application downtime may cause.

Poorly coded applications reduce user productivity and acceptance, which lead to higher indirect costs. At the same time, direct costs can explode through additional hotline calls and troubleshooting efforts. As such, they are likely to lead to much higher total cost of ownership in a production environment than more polished equivalents.

Fortunately for open source, licensing is not the only benefit it offers. The very nature of the easily access-

sible and publicly scrutinized code can also increase reliability and security. The pivotal factor is really the size and dedication of the team creating and refining the software.

Open source projects can often leverage a global community that far exceeds the scale any individual software vendor can dedicate to development. If they are willing to invest a significant effort to fixing the code and expanding its functionality, the results of crowdsourcing can be unbeatable.

A large development effort can mean faster roll-out of new functionality. The very fact that the source is publicly visible also exposes it to in-depth scrutiny. With enough experts involved, it is easier to find any significant vulnerability and fix it immediately rather than trying to keep it secret. Similarly, open-source software goes through a wider peer-review than proprietary software, which adds to its maturity and lays the foundation for high reliability.

Furthermore, open source yields benefits in terms of flexibility for the end customers. If the software doesn't fulfill all the requirements out-of-the-box, they can customize it or extend the foundation in order to plug in other services or write their own extensions. In many cases, they will also make these modifications available to the community thereby again increasing the pace of development.

This doesn't mean that open source is without any drawbacks. The lack of accountability in a community-driven program constitutes a significant risk. It can be difficult to find anybody who will accept responsibility if something doesn't work and support can hinge on the whim and other priorities of good-natured developers. It is possible to reduce these concerns by entering into a business arrangement with a distributor or other open-source specialist. However, open source is really a mindset. Unless an organization is comfortable with the general approach, they may be better off with the well-defined agreements common to commercial software.

This book mainly targets readers who gravitate toward open-source, private implementations of cloud computing at the infrastructure layer for some of the reasons outlined above. But even this choice doesn't lead unambiguously to OpenStack. There are still other options that merit consideration.

Find the Best Fit

We will look at the components and architecture of OpenStack in more detail in the coming chapters, but for now let's just say that it is not that dissimilar to Eucalyptus, CloudStack or OpenNebula. With so many options available, making the selection is not an easy task. In some cases, a feature analysis will lead to a clear favorite. But in many situations, there will be no clear winner.

Most organizations could probably implement any one of those three to satisfy their current requirements. OpenStack's clearest differentiation is its long-term trajectory. It is an excellent example of a project carried forward by a very large open-source community. With over a hundred members including technology leaders like IBM and HP and cloud services pioneers, such as Rackspace, the consortium is unparalleled as an open-source cloud service.

As such, OpenStack can potentially enjoy many of the benefits of open source, such as improved reliability, security and speed of development. Compared to historical infrastructure services that were not much more than a hypervisor, OpenStack is a relatively complete offering including storage, provisioning, identity management and self-service administration.

The fact that it is widely endorsed and adopted also helps to address one of the biggest concerns around cloud computing: vendor lock-in. Cloud providers, such as Rackspace, HP Cloud and many others, offer standard interfaces so that customers can freely move their deployments between providers as well as

between internal implementations and services.

GAPS

This is not to say that OpenStack is entirely complete and addresses every conceivable need. Probably the biggest concern is its maturity. It is a relatively new effort that has undertaken a huge challenge.

As such, it is not quite as polished as some commercial offerings, like Amazon Web Services, VMware vCloud or Microsoft Azure. Some would claim that it also trails behind CloudStack in this respect. It takes more effort to deploy and manage than these other services. And one could argue that the overall reliability is still work in progress.

At the same time, there is also the fact that this is an infrastructure service and many organizations are looking for a platform or software service. This doesn't automatically exclude OpenStack but does mean that there would be a need to extend it with additional software.

For example, it is possible to build an open-source PaaS by layering Stackato on top of OpenStack, or more generally by creating a run-time environment with a web server and open-source tools supporting the programming languages of choice.

SaaS is much more specific to the actual services that are required. Someone who needs a CRM solution might add SugarCRM for sales-force automation, marketing campaigns and customer support as well as collaboration and reporting functionality. For a Content Management System they might choose WordPress, Joomla or Drupal. There are also open-source ERP (OpenERP, Openbravo, ADempiere) and HRM (SimpleHRM, Orange HRM, Waypoint HR) solutions, all of which can run in an OpenStack environment.

In summary, OpenStack is unlikely to ever yield a complete solution. But it can be a component of many solutions to any common problem.

USE CASE FOR OPENSTACK

Given that the hurdles for OpenStack adoption are substantial, it won't be an automatic choice for many businesses. There are three factors that influence how well it will match an organization's requirements.

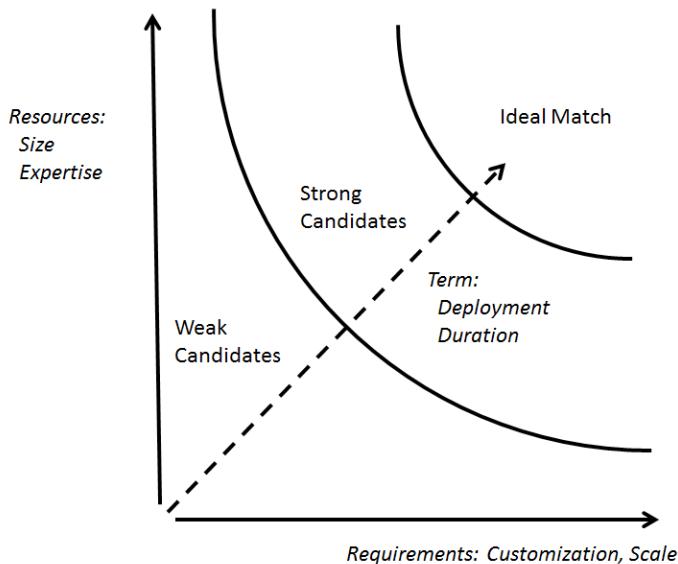


Figure 46: OpenStack Match Criteria

The first consideration is the capability of the team responsible for designing and deploying the cloud infrastructure. This is both a function of size (e.g. budget and number of employees) and expertise (historic focus of this team and the skills of its members). A large group that has considerable experience with infrastructure management as well as the underlying technologies (e.g. Linux, Python, Django) will find it much easier to adopt OpenStack and resolve any issues they have in deploying, operating or extending the environment.

Another consideration is the set of requirements. Someone with relatively simple and modest needs can probably find a commercial solution that fits the bill. On the other hand, an organization with advanced requirements will almost invariably need to add some functionality themselves. Open-source is extensible almost by definition. By providing the results back to the community, users may even be able to share some of the development and support effort.

The scale of the implementation is also an important requirement, particularly from a financial perspective. The larger the deployment, the more significant the variable costs become with respect to the fixed costs. As the installation grows, the licensing fees are likely to become a major element of the marginal costs. At the same time, the fixed costs associated with designing and bootstrapping the new technology will carry less relative weight.

The third axis to consider is time. OpenStack is rapidly maturing and its industry acceptance seems to be growing. What this means is that for projects with a short time frame, both in terms of how soon they need to be deployed and how long they are likely to remain in production, OpenStack might not necessarily be the best fit. On the other hand, solutions that won't launch for years and may need to run indefinitely will benefit from both the increased stability and broader adoption of the future. Over time, benefits will increase while the costs decrease.

Taking these factors into consideration, it is natural to find that most early adoption is from top-tier service providers and some large enterprises. Some software vendors may also consider it as a means of offering a packaged infrastructure solution. These organizations have the resources to overcome any short-term obstacles in deploying OpenStack and they have the large-scale workloads to justify the investment.

There is a good match for service providers at all tiers once the software matures to a point where they can deploy and manage it without investing heavily in specialized expertise. Over time, we can expect the value proposition to become more compelling for smaller and less sophisticated customers.

Enterprises may trail in internal implementation initially, but as they adopt services from OpenStack cloud service providers, they will become more familiar with the interfaces and its administration. They will also find it attractive that by deploying a private cloud based on the same technology, they will be able to reduce provider lock-in and possibly evolve to a hybrid cloud that facilitates cloudbursting and other promising forms of tapping the potential of standardized services. Organizations that are looking for a scalable, long-term strategy may find that OpenStack is their best choice.

Practical Recommendations

The value proposition for an open-source private infrastructure service in the enterprise is rapidly evolving. It is most compelling for organizations with complex requirements and sophisticated capabilities, but as the technologies mature, we can expect smaller businesses to take advantage of the same tools.

There is no riveting technical difference between OpenStack and similar projects, such as Eucalyptus, CloudStack or OpenNebula. Each deserves individual, but not necessarily equal, consideration. OpenStack currently has the most industry attention and is evolving most quickly making its future trajectory very appealing. It would be foolish to ignore it in defining any long-term technical strategy.

OpenStack Cloud Computing Overview

This concludes the excerpt from OpenStack Cloud Computing. The following provides an overview of the complete book.

Initiate

The first step in getting started is to construct a clear picture of how the system should work. This means getting the system working in a pilot scenario with a minimum set of standard components. But you also need to make sure that you will eventually be able to address your requirements and integrate with your legacy environment. You might need more complex topologies or you may need to create linkages to additional components or ecosystems.

Assemble

The design of an OpenStack-based solution begins with the OpenStack services themselves. While it is possible to replace the individual modules, it is generally a good idea to start with the base solution and see to what extent it meets the business requirements. In particular, the core components of an infrastructure service include compute, storage and networking.

Deploy

After the initial design and implementation work is complete, you may have demonstrated the feasibility of the technology but that is a far cry from ensuring it will work in production, particularly for highly scalable workloads. The first task is to roll out the OpenStack software itself onto the bare machines in the data center. The second is to design the orchestration of the workloads so that they are able to launch easily and automatically.

Operate

Once deployed, the administration chores begin. On the one hand, there are proactive tasks to set policies, re-allocate resources and tailor the configuration of standard services based on user needs. On the other hand, it is also important to detect any unforeseen events. We must also keep an eye on trends in order to detect and resolve issues as they occur and to project where future problems may arise in order to prevent them.

Account

Financial governance is a top concern of almost every business. It relies on ensuring visibility of what activities generate expenses and what trends these cost drivers are projecting. Whether the charges are

invoiced to external parties, cross-charged to internal departments or merely reported to show value to the business, the numbers are critical in sustaining a compelling business case.

Secure

OpenStack itself is neither particularly secure nor insecure. Security is a discipline that requires systematic application. This means the first task of a risk analysis is simply to make sure all the components are implemented securely. After verifying that the configuration adheres to best practices, it is important to be vigilant of any newly found exploits and supplement the bare infrastructure with further layers of security. In addition to the base infrastructure, a key component of the overall security model is identity and access management and the enforcement of consistent policies governing user activity.

Empower

One intent of cloud computing is to create an environment that maximizes the benefits of economy of scale. At some point, it may reach a size where failures are inevitable. The most effective solutions will not attempt to prevent them at any cost but rather ensure that the infrastructure and applications are able to withstand these through their high level of redundancy and automated self-healing. A parallelized architecture also enables auto-scaling which reduces the human effort required when load changes. Finally, autonomous operation requires the reduction of dependencies on other vendors or technologies and products.

Extend

Getting the software deployed and working efficiently in production is not the end of the journey. Technology and markets are in constant evolution making it necessary to perpetually adapt. But beyond these externally imposed changes, it is always possible to improve business value by building and extending the infrastructure. Moving up the cloud stack into platforms will drive increased efficiencies for new workloads. Analytics allow IT to generate more business value. And any improvements in the underlying software will help to support new business initiatives and give additional impetus to the community that is building it.

Acronyms

AAA	Authentication, Authorization and Accounting
ACL	Access Control List
AD	Active Directory
AMI	Amazon Machine Image
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
ASIC	Application-Specific Integrated Circuit
ASP	Application service provider
AWS	Amazon Web Services
BCP	Business Continuity Plan
BIOS	Basic Input/Output System
BPEL	Business Process Execution Language
CA	Certificate Authority
CDN	Content Delivery Network
CIDR	Classless Inter-Domain Routing
CIFS	Common Internet File System
CLI	Command-Line Interface
CPU	Central Processing Unit
CRM	Customer Relationship Management
CSA	Cloud Security Alliance
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DoS	Denial of Service
EBS	Amazon Elastic Block Store
EC2	Amazon Elastic Compute Cloud
EDI	Electronic Data Interchange
ERP	Enterprise Resource Planning
eSCM	eSourcing Capability Model

FAQ	Frequently Asked Questions
FCoE	FibreChannel over Ethernet
FPS	Flexible Payments Service
GPU	Graphics Processing Unit
GRE	Generic Routing Encapsulation
GUI	Graphical User Interface
HDD	Hard Disk Drive
HDFS	Hadoop Distributed File System
HIPAA	Health Insurance Portability and Accountability Act
HPC	High Performance Computing
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
I2RS	Interface to the Routing System
IaaS	Infrastructure as a Service
IAM	Identity and Access Management
IETS	Internet Engineering Taskforce
IIS	Internet Information Services
IOPS	Input/Output Operations Per Second
IP	Internet Protocol
IPMI	Intelligent Platform Management Interface
IRC	Internet Relay Chat
iSCSI	Internet SCSI
ITIL	Information Technology Infrastructure Library
JSON	JavaScript Object Notation
KVM	Kernel-based Virtual Machine
LDAP	Lightweight Directory Access Protocol
LVM	Logical Volume Manager
LXC	Linux Container
MAC	Medium Access Control
MIB	Management Information Base

MPI	Message Passing Interface
MTBF	Mean Time Between Failures
MTTR	Mean Time To Repair
NAS	Network Attached Storage
NAT	Network Address Translation
NFS	Network File System
NFV	Network Function Virtualization
NIC	Network Interface Card
NRPE	Nagios Remote Plugin Executor
OASIS	Organization for the Advancement of Structured Information Standards
OLAP	On-Line Analytical Processing
OLTP	On-Line Transaction Processing
OSI	Open Systems Interconnection
PaaS	Platform as a Service
PAM	Pluggable Authentication Modules
PCEP	Path Computation Element Protocol
PCI	Payment Card Industry
PCI-DSS	Payment Card Industry Data Security Standard
PEM	Privacy Enhanced Mail
PHP	PHP: Hypertext Preprocessor
PUM	Privileged User Management
PXE	Preboot Execution Environment
QoS	Quality of Service
RAID	Redundant Array of Independent Disks
RAM	Random Access Memory
RBAC	Role-Based Access Control
REST	REpresentational State Transfer
RHEL	Red Hat Enterprise Linux
RPC	Remote Procedure Call
RPO	Recovery Point Objective
RTO	Recovery Time Objective

SaaS	Software as a Service
SAML	Security Assertion Markup Language
SAN	Storage Area Network
SATA	Serial Advanced Technology Attachment
SCM	Source Control Management
SCSI	Small Computer Systems Interface
SDC	Software-Defined Compute
SDDC	Software-Defined Data Center
SDN	Software-Defined Networking
SDS	Software-Defined Storage
SI	Système International d'unités)
SLA	Service Level Agreement
SMS	Short Message Service
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SPML	Service Provisioning Markup Language
SPOF	Single Point of Failure
SQL	Structured Query Language
SSD	Solid-State Drive
SSH	Secure Shell
SSL	Secure Sockets Layer
SSSD	System Security Services Daemon
STP	Spanning Tree Protocol
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TOSCA	Topology and Orchestration Specification for Cloud Applications
TPM	Trusted Platform Module
UDP	User Datagram Protocol
UEFI	Unified Extensible Firmware Interface
UUID	Universally Unique IDentifier
VHD	Virtual Hard Disk

VIP	Virtual IP address
VLAN	Virtual Local Area Network
VM	Virtual Machine
VPC	Virtual Private Cloud
VPN	Virtual Private Network
WAN	Wide Area Network
XML	Extensible Markup Language
YAML	YAML Ain't Markup Language

About the Authors

John Rhoton is a Strategy Consultant who specializes in defining and driving the adoption of emerging technologies in international corporations. He provides workshops, training and consulting in business strategy and emerging technology around the world.



John has over 25 years of industry experience working for Digital Equipment Corporation, Compaq Computer Corporation, Hewlett-Packard and Symantec where he has led technical communities and driven the services strategies around a wide variety of initiatives including cloud computing, mobility, next-generation-networking and virtualization.

Feel free to follow him on Twitter (@johnrhton), connect with him on LinkedIn ([linkedin.com/in/rhton](https://www.linkedin.com/in/rhton)) or find out more about his work (about.me/rhton).



Jan De Clercq is a member of HP's Technology Services Consulting IT Assurance & Security Portfolio team where he develops new services and delivers consultancy to HP accounts worldwide.

He focuses on cloud security, software-defined security, networking security, mobility security, and identity and access management.

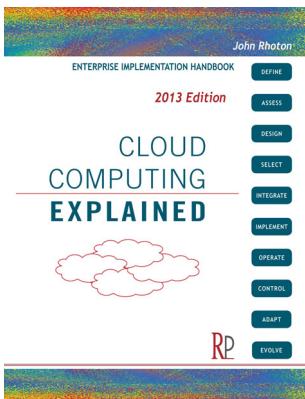
You can reach him at jan.declercq@hp.com.



Franz Novak is a Presales Consultant and Solution Architect in the HP Enterprise Group.

Franz has over 25 years of experience in the computing industry working for Silicon Graphics, Sun Microsystems and Hewlett-Packard. His areas of expertise include Enterprise Architecture, IT Transformation and SOA.

For the past five years he has worked in the area of cloud computing and solution design. He has an M.S. in Computer Science from the Technical University of Vienna.



Also by John Rhoton

Cloud Computing Explained *Enterprise Implementation Handbook* *2013 Edition*

Paperback: 472 pages
Publisher: Recursive Press
ISBN-10: 0956355609
ISBN-13: 978-0956355607
Product Dimensions: 24.6 x 18.8 x 2.5 cm

Cloud Computing Explained provides an overview of Cloud Computing in an enterprise environment. There is a tremendous amount of enthusiasm around cloud-based solutions and services as well as the cost-savings and flexibility that they can provide. It is imperative that all senior technologists have a solid understanding of the ramifications of cloud computing since its impact is likely to permeate the entire IT landscape. However, it is not trivial to introduce a fundamentally different service-delivery paradigm into an existing enterprise architecture.

This book describes the benefits and challenges of Cloud Computing and then leads the reader through the process of assessing the suitability of a cloud-based approach for a given situation, calculating and justifying the investment that is required to transform the process or application, and then developing a solid design that considers the implementation as well as the ongoing operations and governance required to maintain the solution in a partially outsourced delivery model.



Cloud Computing Architected

Solution Design Handbook

2013 Edition

Paperback: 384 pages

Publisher: Recursive Press

ISBN-10: 0956355617

ISBN-13: 978-0956355614

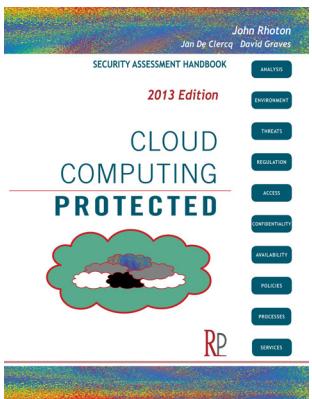
Product Dimensions: 24.6 x 18.9 x 2 cm

Cloud Computing Architected describes the essential components of a cloud-based application and presents the architectural options that are available to create large-scale, distributed applications spanning administrative domains.

The requirements of cloud computing have far-reaching implications for software engineering. Applications must be built to provide flexible and elastic services, and designed to consume functionality delivered remotely across of spectrum of reliable, and unreliable, sources. Architects need to consider the impact of scalability and multi-tenancy in terms of:

- New development tools
- Internet-based delivery and mobile devices
- Identity federation
- Fragmented services and providers
- Exploding information volume
- Availability and elasticity techniques
- New business models and monetization strategies
- Revised software development cycle
- Increased operational automation

This book looks at these and other areas where the advent of cloud computing has the opportunity to influence the architecture of software applications.



Cloud Computing Protected

Security Assessment Handbook

2013 Edition

Paperback: 412 pages

Publisher: Recursive Press

ISBN-10: 0956355625

ISBN-13: 978-0956355621

Product Dimensions: 24.6 x 18.9 x 2.1 cm

Cloud Computing Protected describes the most important security challenges that organizations face as they seek to adopt public cloud services and implement their own cloud-based infrastructure.

There is no question that these emerging technologies introduce new risks:

- Virtualization hinders monitoring and can lead to server sprawl.
- Multi-tenancy exposes risks of data leakage to co-tenants.
- Outsourcing reduces both control and visibility over services and data.
- Internet service delivery increases the exposure of valuable information assets.
- Ambiguity in jurisdiction and national regulations complicates regulatory compliance.
- Lack of standardization can lead to a lock-in binding customers to their providers.

Fortunately, there are also many security benefits that customers can enjoy as they implement cloud services:

- Highly specialized providers have the economy of scale to invest in best-in-class tools and expertise.
- Contractual terms can clearly define the function and scope of critical services.
- Public services receive unprecedented scrutiny from the collective worldwide community.
- It is possible to achieve unlimited levels of redundancy by subscribing to multiple providers.
- The global reach of the Internet and security specialists facilitates early alerts and drives consistent policy enforcement.

This book looks at these and other areas where the advent of cloud computing has the opportunity to influence security risks, safeguards and processes.

About Mirantis

Mirantis OpenStack lets you create production deployments on premise with enterprise-class service levels, but without compromising flexible access to open source innovation. You get a range of powerful tools for deployment and management, and proven configurations, all backed by commercial support, as well as engagement with the upstream OpenStack community. For the fastest and easiest way to get an OpenStack cloud, Mirantis Managed Services On Demand offers your own on-demand private cloud powered by OpenStack. Deploy, develop, or go to production with the cloud – in under an hour, all in one cloud-based service you run right from your web browser.