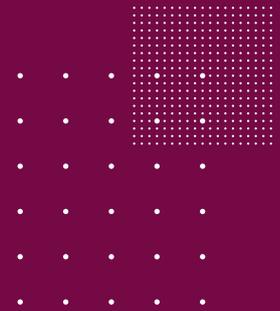


CANONICAL



WHITE PAPER

Creating the open cloud

May 2012

Contents

INTRODUCTION	03
WHY THE OPEN CLOUD?	03
USING A PUBLIC CLOUD	04
Cutting complexity with cloud-init	04
Ensure your guest is optimised	05
BUILDING A PRIVATE CLOUD	06
The importance of OpenStack	06
Bringing a public cloud workload in-house	06
TOWARDS THE HYBRID CLOUD	07
Managing multiple clouds	07
ACCELERATED DEVELOPMENT	08
Developers need micro-clouds	08
How service orchestration saves time	08
A little charm goes a long way	08
An end to image sprawl	09
PRIVATE CLOUD DEPLOYMENT	10
Turn your bare metal into a service	10
The hyperscale era is here	10
WHERE TO FIND OUT MORE	11



Introduction

Amid all the hype surrounding the cloud, there has been a growing need for open and proven components that enable any business to benefit from elastic computing. With public cloud services now widely available and OpenStack established as the de facto infrastructure platform, these components are now ready to deploy.

This white paper explains how OpenStack and other open cloud tools can be accessed via Ubuntu Server 12.04 LTS – and the technical reasons why we think this represents the best route to the cloud.

Why open cloud?

Today's organisations have a common vision: accelerating the move to utility-style cloud computing. They want on-demand access to the fundamental commodities of IT: network, compute and storage. They want to be able to scale that pool of resources from departmental pilot to massive infrastructure. Most importantly, they want to cut out the fat, the complexity and the vendor lock-in that can hamper their ability to deliver fast, efficient service to their users.

Their challenge is to tame the complexity of cloud design and construction. Clouds combine many diverse infrastructure components, from hardware, networks and storage, to operating systems and applications. Those have all been developed independently, to different standards, with different configuration systems and specialist skills needed to connect them. And they were all designed to be installed one piece at a time. But to make a cloud, those parts have to become dynamic. They need to offer true computing elasticity, automated provisioning and deprovisioning, open APIs and pay-as-you-go metering and billing. Above all, they need to go beyond open standards and be fully open source. That is the open cloud.

One of the biggest single decisions made in the formulation of any cloud strategy is whether to procure resources from one of the public cloud services or to build private cloud infrastructure. While a third way is emerging – the option to combine both in a hybrid cloud – it is first worth considering the two established options in turn.



Using a public cloud

When choosing a public cloud, it is important to make the right choice of guest operating system. Most public clouds offer both proprietary and free options. In the virtual world of the cloud, where you need the freedom to spin up server instances as demand dictates, the licensing restrictions associated with proprietary software can make such a choice financially impractical. What is required is a guest OS that's free from licensing constraints, yet for which enterprise-grade support is accessible for workloads that demand it.

Ubuntu Cloud Guests are wildly popular on the leading public clouds for precisely these reasons. Ubuntu 12.04 LTS will receive free security and maintenance updates for the next five years, guaranteed, so you can be confident that your workloads will be supported in the long term. And commercial support is available from Canonical, the company that backs Ubuntu.

As an open source OS and the leading cloud guest, Ubuntu receives contributions from businesses and independent developers alike. This gives it the widest selection of cloud tools. One example is cloud-init, a popular open source tool, pioneered by Canonical.

CUTTING COMPLEXITY WITH CLOUD-INIT

Cloud-init is used to customise standard images to specific needs, dynamically on startup, reducing the complexity of managing large numbers of custom images. Today, it is widely recognised as the best tool for boot-time workload customisation.

Instead of maintaining one image per workload (or worse, one image per node), it is service descriptions that are maintained. A standard image is then customised at start-up to suit that service description and security requirements. Early adopters of the cloud complained of image sprawl; with cloud-init, the problem can be avoided entirely.

Correct and complete support for cloud-init is a requirement for public clouds that publish Ubuntu images. This ensures that deployments will work well across all Ubuntu-certified clouds.



ENSURE YOUR GUEST IS OPTIMISED

Every public cloud is different. For example, each makes its own decisions about kernel and virtualisation. That's why it's vital that any chosen guest OS is optimised for that cloud.

Canonical optimises Ubuntu Cloud Guest for every public cloud it certifies. It offers a commercial service to all public cloud providers, to help them provide a quality guest OS that can be consumed easily and freely. Public cloud providers can therefore ensure that their cloud meets the standards set on AWS, Rackspace and elsewhere, offering complete compatibility with Ubuntu when used on any other certified cloud.

Canonical also designs and implements optimal update mechanisms. So on AWS, for example, it is now possible to get updates at very low cost from S3. This infrastructure is unique to Ubuntu on AWS, however all certified clouds are engaged with, to provide a first class update experience.

Canonical invests in the Ubuntu experience on public clouds to guarantee the most complete combination of performance, update handling, compliance and reliability. It also extends its certification, management, audit compliance and IP assurance services to paying customers using Ubuntu on certified clouds. Whatever guest OS you choose to run in the public cloud, it is vital that you insist on this level of support.



Building a private cloud

For many reasons, not least security of data and control over applications, many organisations now choose to build their own private cloud infrastructure. Fortunately, there is an established open source infrastructure platform: OpenStack.

THE IMPORTANCE OF OPENSTACK

An open source project initiated by Rackspace and NASA, OpenStack has quickly become the standard for open cloud infrastructure. All OpenStack code is available freely under the Apache 2.0 licence – so anyone can run it, build on it, or contribute to the project. This development model has fostered a vibrant community, with the largest ecosystem of tools, solutions and service providers.

OpenStack is built into Ubuntu Server and Ubuntu is the reference operating system for Openstack deployments. An enormous amount of work has gone into the integration of OpenStack and Ubuntu – work that’s been done so users needn’t do it themselves.

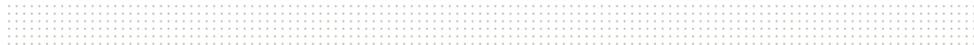
The relationship between Ubuntu and OpenStack is significant. Both brands are focused on high quality governance, platforms for mass computing, predictable schedules and robust releases. Ubuntu has included cloud infrastructure since 2009 and, since 2011, the latest version of OpenStack has been in every Ubuntu release. The release schedules of the two projects are synchronised, ensuring that OpenStack updates and releases are immediately available on widely deployed releases of Ubuntu.

BRINGING A PUBLIC CLOUD WORKLOAD IN-HOUSE

As a cloud strategy proves itself in a production environment, it is common for workloads that were initially run on a public cloud to be brought in-house, onto newly-built private cloud infrastructure.

In these situations, the ability to use the same components and the same tools as the major public cloud vendors offers a convenient assurance of practical – rather than theoretical – compatibility between private infrastructure and the public clouds.

Like Ubuntu Cloud Infrastructure, the public clouds from Rackspace, AT&T and HP are all based on OpenStack, making migration between them and private OpenStack infrastructure a straightforward process. For migrations from Amazon’s EC2 and S3, Canonical’s Awsome component provides a solution. Awsome is a separate server component that acts as a proxy between AWS and OpenStack, enabling it to take AWS requests on the front end and translate them to OpenStack’s native protocols on the back end. This ensures all tools can work with both AWS and OpenStack-based clouds, private or public, simplifying migration significantly.



Towards the hybrid cloud

Many organisations want the best of both worlds – the private and the public cloud. Private cloud provides control, regulatory compliance, cost management and security. Public cloud offers economic efficiency, burst capacity and disaster recovery. But they are not exclusive to one another. With careful planning and the right tools, it is possible to scale private clouds into the public cloud and stay in control.

MANAGING MULTIPLE CLOUDS

Canonical's support customers have access to Cloud Deck, a hybrid cloud dashboard, compliance and control system. It provides a simple graphical interface for managing multiple clouds that can span public and private infrastructure.

Cloud Deck is a solution to the key problem facing IT managers around the world – how to enable self-service public cloud adoption while retaining the ability to audit, manage and ensure compliance of that use. Public clouds enable teams to use the corporate credit card to bypass central provisioning bottlenecks. Cloud Deck lets administrators give teams the freedom they have come to expect, while reclaiming the management and audit control necessary for regulatory compliance.

Cloud Deck facilitates the use of multiple clouds supporting the EC2 API (OpenStack, Eucalyptus, AWS EC2) through a single, uniform API and interface. It makes it possible to synthesise multiple clouds – like Amazon, AT&T, HP and your private OpenStack infrastructure – providing access to them all as if they were one cloud. Administrators can set access controls and policies to govern the cloud regions in which staff are allowed to create workloads, along with the constraints or monitoring requirements they need to meet.

Cloud Deck provides a single API, a single credential and a single configuration they can use to deploy workloads in any approved cloud region. To add relationships with new public cloud providers, there is no need to create new credentials or provide new configuration information. The new regions simply show up in the unified hybrid cloud.



Accelerated development

The speed with which capacity can be provisioned in the cloud shifts the efficiency bottleneck from the provisioning stage to deployment. For devops teams, this means the speed with which they can develop, test and deploy new application iterations becomes a critical measure of success.

DEVELOPERS NEED MICRO-CLOUDS

Ubuntu includes a micro-cloud capability on every developer workstation. When building complex multi-tier applications, developers can use it to create microcosms of the production test and deployment systems, creating nodes in their micro-clouds for every equivalent node in the production deployment. In combination with Juju, Ubuntu's service orchestration tool, this means any changes in the code or the required configuration can be communicated instantly to operations staff for testing or production deployment.

HOW SERVICE ORCHESTRATION SAVES TIME

Juju is a service orchestration tool that works on both public and private clouds, on physical deployments and on developer workstations. In combination with the developer micro-cloud in Ubuntu, it dramatically reduces friction in the handovers between development, testing and deployment.

Juju works by encapsulating service deployment expertise in files called charms. These charms can then be used to spin up workloads across all the deployment environments – development, test and production. When developers make a change in the deployment process as part of the development cycle, that change is passed on to system administrators when the code is handed over.

Using a single framework for development, test, staging and production saves time in the development cycle. That suits the move to cloud computing perfectly, because it matches the reduction in friction from automated and virtualised provisioning.

A LITTLE CHARM GOES A LONG WAY

A Juju charm is a collection of instructions that deploys, updates and scales a particular cloud service. When defining a new workload or service, a charm is created for it using whatever system works best. It can be a shell script, it can use puppet, or it can use any other framework you like. This makes it easy to re-use existing tools or expertise that may be present in-house, wrapping it up in a way that will work on the cloud.

Most services can be charmed in an hour or two, at least for initial testing. And investments in a charm pay off every time it is re-used. Charms encapsulate everything a service needs to know about itself, or tell other services about itself, so it's very easy to re-use them in a different team or environment.



Canonical maintains a collection of public charms that are developed in the open, under the same transparent governance that has made Ubuntu the leading cloud OS. Each charm distills best practice from the leading devops for that particular service, worldwide. Juju puts them all at devops teams' fingertips.

Those charms continue to improve and evolve, so cloud deployments become smarter, more efficient and more reliable every time they are updated. In a recent example, work done to reduce the cost per day of a very high-traffic cloud-hosted website was shared immediately with other websites using the same cloud stack. In an enterprise setting, an improvement to the charm for a component in many cloud stacks brings benefit to all users.

The collection of Juju charms includes all the common components of typical cloud deployments – popular databases, web application servers, load balancing systems, computational frameworks; everything from game servers to finite element analysis is ready for off-the-shelf deployment in the cloud.

AN END TO IMAGE SPRAWL

Cloud computing creates new challenges for IT governance. Workloads can be spun up on public services, without management's knowledge and at an enormous scale. To make things worse, customised images in public clouds become unmaintainable very quickly, leaving IT managers with the mess. Early adopters of the cloud struggled with this snowballing accumulation of images. But with cloud-init and Juju, which together eliminate the need for custom images in favour of dynamic specialisation on startup, image sprawl becomes a thing of the past.



Private cloud deployment

There are many technology options when it comes to building private cloud infrastructure. But before any kind of software infrastructure can be deployed, the thorny issue of hardware provisioning must be addressed.

TURN YOUR BARE METAL INTO A SERVICE

Metal as a Service (MAAS) is a new bare metal provisioning system in Ubuntu 12.04 LTS.

Among its many applications, MAAS can quickly provision a cluster of servers, prior to deploying an OpenStack cloud – and it does so by treating that cluster of metal just like virtual machines in a cloud. Together with Juju, it can then radically speed up deployment of the cloud itself.

Once MAAS has been informed of the servers it needs to control, it performs a hardware health-check on each one. Juju then sends a request to MAAS for the resources it needs, before MAAS installs Ubuntu. Finally, Juju completes the deployment of OpenStack.

Rather than worrying about individual machines, MAAS and Juju focus on the service to be deployed. Managing clusters of servers like this is a cloud-like way of thinking about provisioning. To put it another way, MAAS brings cloud semantics to the physical world.

THE HYPERSCALE ERA IS HERE

Servers used to be big, expensive and precious. But those days are numbered.

With computing capacity now available as a service, data centres are shifting to a model in which more power is gained by adding more nodes to their clusters, rather than via bigger machines. Just as RAID changed the storage game, this scale-out philosophy, pioneered by Google, is changing the server landscape.

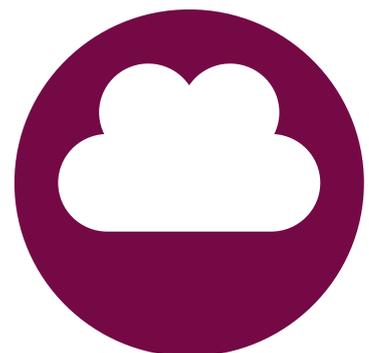
This trend culminates in the hyperscale era. Instead of managing tens of servers, businesses can end up with tens of thousands. It will soon be normal for organisations to have more nodes in their cloud or cluster than they have employees.

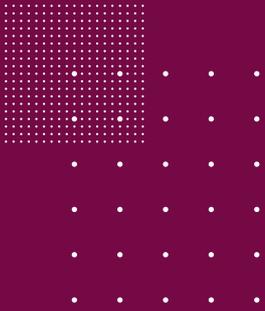
For this hyperscale model to work, however, leaner operating systems are required, that are free from licensing restrictions. Leaner provisioning processes are necessary, via deployment tools that can cope with the new physical reality. Ubuntu – when used with its associated technologies like OpenStack, MAAS and Juju – meets all these requirements. As we enter an era in which ATOM is as important as XEON, open source software like Ubuntu will be essential to the development of the cloud.

To find out more about cloud computing with Ubuntu, go to www.ubuntu.com/I-cloud

To learn about Ubuntu Advantage, the Canonical support programme that includes Cloud Deck, go to www.ubuntu.com/I-cloud-services

To speak directly to a member of the Canonical team, please call **+44 (0)20 763 2471**





CANONICAL

© Canonical Limited 2012. Ubuntu, Kubuntu, Canonical and their associated logos are the registered trademarks of Canonical Limited. All other trademarks are the properties of their respective owners. Any information referred to in this document may change without notice and Canonical will not be held responsible for any such changes.

Canonical Limited, Registered in England and Wales, Company number 110334C
Registered Office: One Circular Road, Douglas, Isle of Man IM1 1SB VAT Registration: GB 003 2322 47