

Escenic Cloud
Introduction
2.0-SNAPSHOT

Table of Contents

1 About Escenic Cloud	3
1.1 Philosophy	4
1.2 Limitations	4
2 Architecture	6
3 Cloud Configurations	7

1 About Escenic Cloud

Escenic Cloud is a set of tools that supports the configuration, installation and maintenance of complete Escenic Content Management Systems (CMS) in virtual machines hosted in a cloud.

The heart of an Escenic installation is the Escenic Content Engine, Escenic Widget Framework and Escenic plug-ins provided by Escenic. There are, however, many other vital components of an Escenic system: operating system, Java VM, database, web servers, application servers, caches and so on, all of which need to be installed, maintained and configured to work together across a network of multiple machines. Installation and maintenance of such a system is complex and expensive.

Escenic Cloud is Escenic's response to this challenge. Escenic Cloud allows an entire Escenic installation including all of its components to be described in a small set of configuration files. The description encompasses everything that goes to make up a working Escenic installation:

- "Hardware" (i.e virtual machines)
- Supporting SW (operating systems, databases, web servers and so on)
- The Content Engine, Widget Framework and plug-ins
- The customer's own publications

Once these configuration files have been made, the entire network can be generated from them in a highly automated, repeatable way.

Escenic Cloud also incorporates support for development best practices based on Escenic's long in-house experience. In addition to a ready-configured working installation, Escenic Cloud also provides a ready-to-use development environment and build/deployment pipeline.

Escenic Cloud tools can be used to manage installations either in the Amazon EC2 public cloud or in private clouds run on your own hardware (**bare metal** installations). Escenic Cloud 2.0, however, as the first official release of the toolset only supports bare metal installations.

Escenic Cloud 2.0 offers major advantages over a traditional "hand-installed" Escenic system:

Greater stability

Your Escenic system is effectively pre-configured by the experts.

Improved scalability

Given the available hardware resources, adding or removing servers is simply a matter of editing text files.

Reduced costs

Almost all aspects of installation, maintenance and deployment are greatly simplified and therefore cheaper.

Easier upgrades

Just edit the necessary version numbers and reinstall.

Easier deployment

A built-in build server makes deployment of your apps straightforward and safe.

1.1 Philosophy

The fundamental idea underlying Escenic Cloud is the idea of the **zero-value machine**.

In a traditional hardware-based installation of a complex system, the installation and its configuration embodies a significant investment. The investment we're talking about here isn't the value of the hardware, or the value of the software installed on it, or the value of any data stored in it. It is the value of the effort required to bring all those components together as a functioning system.

For a complex system such as an Escenic CMS, this effort usually represents a great deal of value and therefore a major risk. Purchasing the right hardware, installing the correct versions of all the required software components and then configuring them to work correctly with each other and with the software installed on other machines in the network requires skilled staff, the maintenance of large amounts of institutional knowledge and often a considerable amount of time.

A zero-value machine is a virtual machine (which by definition has no hardware value) that can be brought to a fully-functional state with effectively no expenditure of effort.

Escenic Cloud tries to achieve the goal of zero-value machines by:

- Encapsulating all the knowledge required to produce a particular Escenic installation in a configuration file called a **service design package**. A service design package contains all the information needed to:
 1. Create and configure a set of virtual machines
 2. Download and install all the software components required on each machine
 3. Configure all the installed software components so that they work together
- Providing a toolset that, given a service design package, can generate and start up the described virtual machines in a matter of minutes.

In this way, Escenic Cloud greatly reduces the costs and risks involved in:

- Initial installation of Escenic systems
- Reconfiguration and scaling of existing Escenic installations
- Upgrading Escenic installations
- Recovery from system failures

1.2 Limitations

At version 2.0, Escenic Cloud has not yet completely achieved its "zero-value" goal since two of the VM types potentially required in an Escenic Cloud still require manual installation:

- Git VM
- Builder VM

These are, however, relatively minor limitations since:

- Many installations will not include a Git VM since the customer prefers to use a pre-existing Git repository or a cloud-based Git hosting service such as GitHub or Stash.

- The Builder VM is in most cases a "one-time" installation that is unlikely to require much in the way of upgrades, reconfiguration or other maintenance. It is, nevertheless, an objective to remove the need for manual installation of the Builder in a future version of Escenic Cloud.

2 Architecture

An Escenic Cloud is a network of virtual machines that interoperate to provide a complete, production-ready Content Management System (CMS) based on Escenic technology. It incorporates not only Escenic-supplied components such as the Escenic Content Engine, Escenic Widget Framework and various Escenic plug-ins, but all the other supporting components required to make a complete system (operating systems, Java VM, database, web servers, application servers, caches and so on).

The VMs in an Escenic Cloud can be configured to fill specific roles that determine the SW components installed on them. Some typical roles include:

Git

A Git VM hosts a central Git repository used to store Escenic applications.

Builder

A builder VM hosts everything required to build Escenic applications. It is configured to pull code from the central Git repository, build applications and make them available for deployment on other machines in the cloud.

Database

A database VM hosts the MySQL database used to store Escenic publication content.

Escenic all-in-one

An Escenic all-in-one VM hosts all Escenic components plus a database, thus providing a complete Escenic system that can both support back-end editorial access and also provide the front-end publishing function of a CMS. All-in-one VMs are typically used for development purposes.

Escenic editorial

An Escenic editorial VM hosts an Escenic installation that is specialized to support back-end editorial access and does not include presentation-related components.

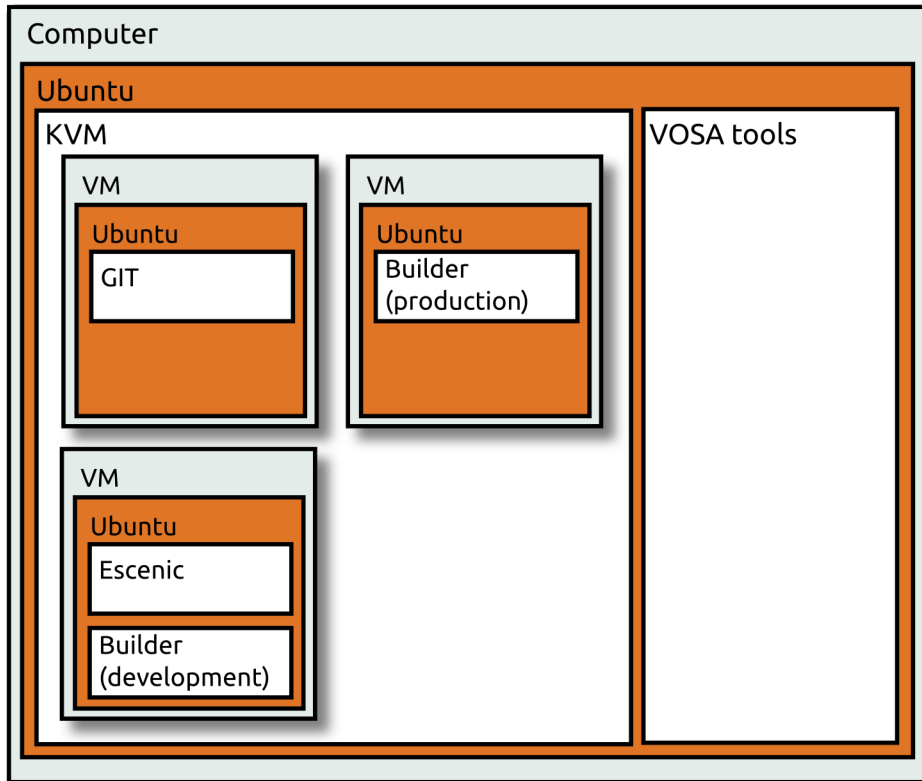
Escenic presentation

An Escenic presentation VM hosts an Escenic installation that is specialized to provide the front-end publishing function of a CMS and does not contain editorial components.

In other words, an Escenic Cloud can be configured in many different ways in order to meet different requirements. In addition, the configuration may change over time in order to meet changing needs.

3 Cloud Configurations

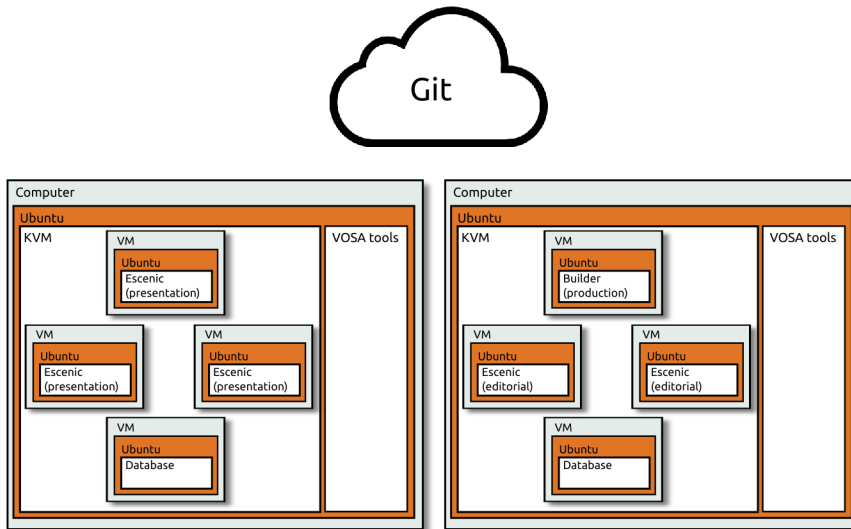
The [Escenic Cloud Hardware Installation Guide](#) provides step-by-step installation instructions for creating the very simplest kind of Escenic Cloud: an all-in-one cloud that can be installed on a single computer for demo or development purposes:



It consists of three VMs: a Git VM, a Builder VM and an Escenic all-in-one VM.

An all-in-one cloud would never be used for production purposes. Typically, a production installation still has a single Builder instance, but the components installed together in the Escenic all-in-one VM are divided between a number of special-purpose VMs. Many organizations prefer to use a pre-existing Git repository or to make use of an external Git hosting service. In order to help ensure a high level of

availability a production cloud is also usually spread across at least two servers, and configured so that service is maintained even if one server fails. For example:



The exact number of Escenic Editorial VMs and Escenic Presentation VMs included in a cloud varies according to requirements, and can also be increased and decreased to meet peaks and troughs in demand. In general, however, a production cloud is likely to include more presentation VMs (the front-end VMs that serve web pages to the Internet) than editorial VMs (the back-end VMs that provide the editorial functionality used by newsroom staff).