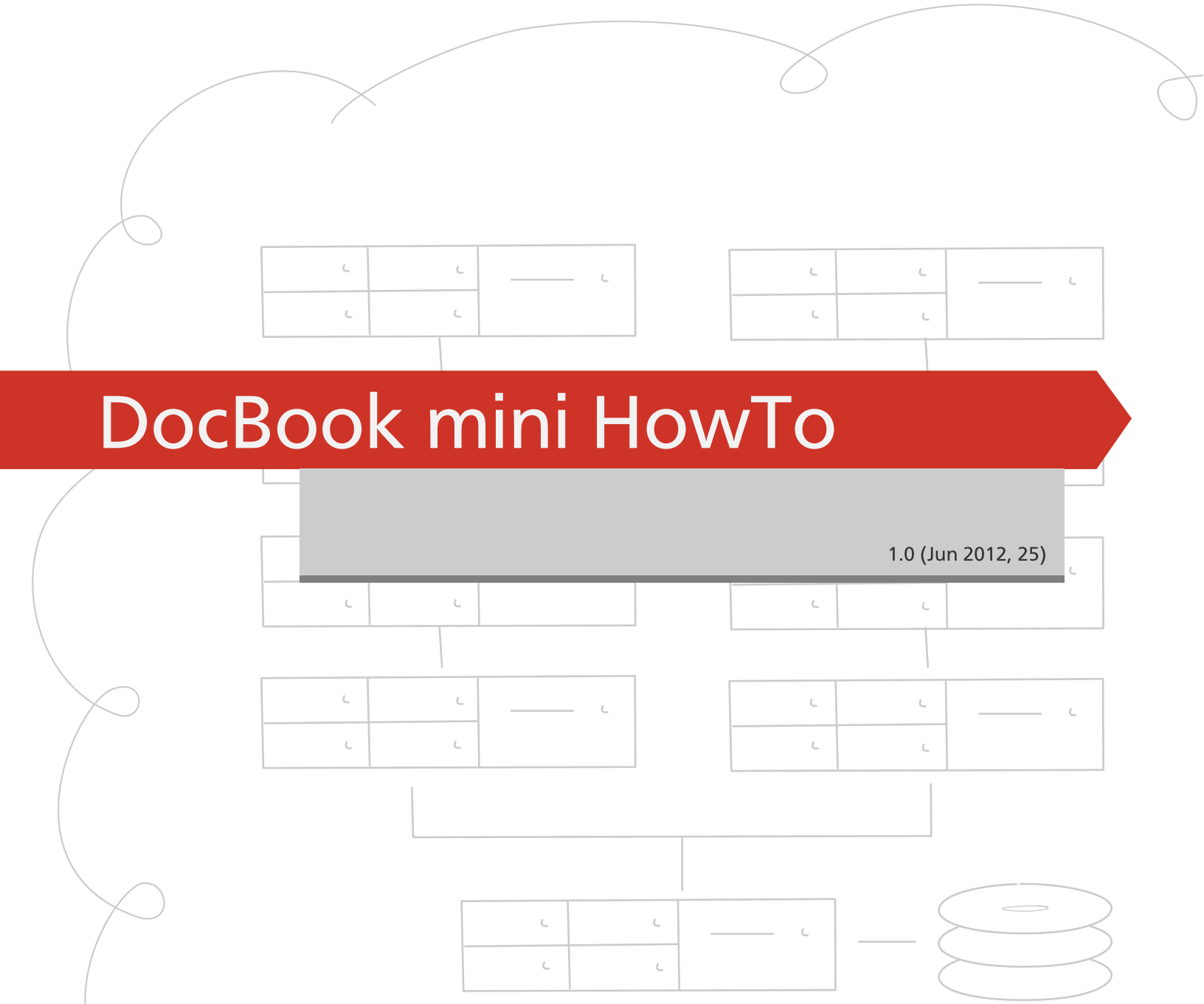


DocBook mini HowTo

1.0 (Jun 2012, 25)

docs.openstack.org

DocBook mini HowTo

Alejandro Roca Alhama

1.0 (25-06-2012)

Este documento trata de ser una pequeña guía de introducción a DocBook, utilizada principalmente en la creación de la documentación asociada al proyecto de innovación de Formación Profesional "Implantación y puesta a punto de la infraestructura de un cloud computing privado para el despliegue de servicios en la nube", desarrollado por profesores de Informática de los centros: IES Gonzalo Nazareno, IES Ingeniero de la Cierva (Murcia), IES "La Campiña" e IES "Los Albares".

Table of Contents

List of Figures

List of Tables

1. Prerequisitos

Para seguir esta guía se necesita:

- Muchas ganas de aprender, DocBook se hace muy duro al principio y solo se ven las increíbles ventajas cuando se empieza a tomar cierta soltura después de trabajar con él unas cuantas horas.
- Conocimientos básicos sobre XML.
- Una distribución GNU/Linux como Ubuntu (11.10 ó 12.04 LTS), Debian ó Fedora.
- Acceso a los repositorios GIT del proyecto OpenStack.

2. Trabajando con Docbook: emacs y nxml-mode

Párrafo de introducción del capítulo.

¿Por qué emacs?

Para alguien que empieza con emacs es complicado justificar su uso, hasta que poco a poco cada vez va gustando más. Entre las razones para lanzarse a emacs se podr#a destacar:

- Es Software Libre, y creado en su origen por el mismísimo Richard Stallman.
- Es un editor de texto, no un procesador, se puede llegar a ser muy productivo a la hora de escribir documentación si se usa junto con Latex ó con DocBook.
- Tiene modos específicos para una infinidad de lenguajes de programación: C/C++, Python, Bash...
- Uno de los mejores editores de marcas (XML, SGML, (X)HTML, y como no, DocBook.
- Muy configurable y fácilmente extensible a través de Lisp.
- Muchas de las combinaciones de teclas que se aplican en Emacs también se aplican en Bash.

Pero por desgracia, también tiene desventajas, la única remarcable sería la curva de aprendizaje, hay que hacerse un poco con la filosofía de emacs y sobretodo con sus atajos de teclado.

Instalación y configuración

Para el uso de Emacs+DocBook vamos a utilizar **nxml-mode**, nxml-mode es un plugin para GNU Emacs que lo convierte en un poderoso editor XML. Todos los comandos de emacs se pueden utilizar sin restricción alguna en este nuevo modo. El modo nXML permite asociar cualquier esquema al documento XML que se está editando, incluso sin que el documento XMl sea en todo momento ni válido ni bien formado.

nXML permite asociar un esquema al documento que se está editando, el esquema utilizado proporciona dos características básicas para la edición:

- Validación continua. nXML valida el documento conforme se va editando, resaltando en todo momento cualquier parte errónea.
- Completado automático. nXML puede ayudar a la hora de insertar etiquetas, nombres de elemento, nombres de atributos o valores según la información que obtiene del esquema y del contexto.

El modo nXML utiliza RelaxNG como lenguaje de definición de esquemas, RelaxNG utiliza dos sintaxis: una basada en XML y otra más legible y compacta pero que no está basada en XML. nXML soporta la sintaxis compacta. Aunque existen herramientas para convertir tanto DTDs copo otros lenguajes de esquema a la sintaxis de RelaxNG, no van a ser

necesarias ya que el paque de software de nXML incluye los esquemas para Docbook, XHTML, RDF y RelaxNG.

También se puede utiliza nXML sin utilizar ningún esquema, no están disponibles todas las características, pero por ejemplo, nXML sigue autocompletando cerrando de forma automática los tags.

nXML está escrito completamente en Lisp, por lo que está disponible para cualquier plataforma en la que Emacs esté disponible.

Para instalar tanto emacs como nXML basta seguir los siguientes pasos:

1. Instalamos los siguientes paquetes:

```
$ apt-get install emacs
```

   Nos descargamos el siguiente fichero de la página oficial del proyecto:

```
$ cd ~/Descargas
$ wget http://www.thaiopensource.com/download/nxml-mode-20041004.tar.gz
```

   Tenemos todas las versiones disponibles desde: `http://www.thaiopensource.com/download/`

2. Instalamos el plugin en el directorio de Emacs:

```
$ cd ~/.emacs.d
$ tar -xvzf ~/Descargas/nxml-mode-20041004.tar.gz
```

3. Indicamos a Emacs que utilice el plugin, para ello basta añadir el siguiente contenido al fichero `~/.emacs.d/init.el`:

```
;;--
;; Make sure nxml-mode can autoload
;;--
(load "~/.emacs.d/nxml-mode-20041004/rng-auto.el")

;;--
;; Load nxml-mode for files ending in .xml, .xsl, .rng, .xhtml
;;--
(setq auto-mode-alist
      (cons '("\\.\\(xml\\|xsl\\|rng\\|xhtml\\)\\'" . nxml-mode)
     auto-mode-alist))
```

Note

Si estos ficheros no existen es porque emacs nunca se ha ejecutado, basta iniciar y cerrar el programa.

4. Emacs necesita saber dónde encontrar los esquemas para poder validar el documento al mismo tiempo que se está escribiendo. Para ello basta con crear el fichero `schemas.xml` dentro del directorio `~/.schemas` con el siguiente contenido:

```
<locatingRules xmlns="http://thaiopensource.com/ns/locating-rules/1.0">
  <namespace ns="http://docbook.org/ns/docbook" uri="/usr/share/xml/docbook/
schema/rng/5.0/docbookxi.rnc"/>
</locatingRules>
```

Comandos nXML a utilizar

Los comandos más utilizados a la hora de escribir en Emacs con el modo nXML son:

Table 2.1. Comandos del modo nXML

Comando	Descripción
Tabulador	Identa la actual línea (o la región actual) al nivel de indentación del bloque en el que estemos.
C-ENTER	Completa el nombre de la etiqueta o el nombre del atributo que estemos escribiendo en ese momento, si hay varios mostra#a una lista de posibilidades.
C-c-C-f	Completa la línea cerrando el tag que corresponda. Si la utilizamos repetidamente irá cerrando todos los tags que estén abiertos.
C-c-C-i	Cuando se acaba de escribir el nombre de la etiqueta, pero sin llegar a cerrarla con >, incluye el signo > y el tag de cierre, poniendo el cursor entre las etiquetas facilitando la inserción de texto. Muy útil para la inseción de elementos inline (como *para*, *literal*, *filename*, etc.).
C-c-C-b	Igual que la anterior pero para elementos de bloque.

Note

En terminología Emacs, "C-c-C-i" significar pulsar la tecla Control + la tecla 'c' + la tecla 'i'.

Snippets de código

Siempre que se escriban documentos XML, es muy útil acumular un conjunto de plantillas pequeñas que faciliten la escritura. Una plantilla no será más que un pequeño fichero que contiene un esqueleto básico de DocBook ó de XML que permita tareas ritinarias como insertar una tabla en blanco, una figura, una sección, etc.

Estas plantillas se pueden almacenar en un directorio y se podrán insertar en cualquier momento a través del comando Emacs `insert-file` (C-x-i).

Para configurar estos snippets basta con:

1. Crear pequeños ficheros, uno para cada snippet, y poner un nombre al fichero identificativo del trozo de código. Por ejemplo, si queremos crear un snippet para poder insertar listas ordenadas, crearíamos el fichero `./.snippetsXML/orderedlist` con el siguiente contenido:

```
   <orderedlist>
     <listitem>
<para></para>
     </listitem>
   </orderedlist>
```

A la hora de insertar el snippet anterior bastaría con pulsar C-c-i más el principio de la palabra `orderedlist` completando si es necesario con la tecla tabulador. En el caso anterior no vendría mal incluir además un snippet denominado `listitem`.

2. Para mayor comodidad se puede asociar al comando `insert-file` otra combinación de teclas más cómoda, al mismo tiempo que la inserción del snippet incluya una identación automática. Para ello asociaremos la combinación de teclas **C-c-C-e nombre_snippet**) a la inserción de snippets de nuestro directorio configurado en el paso anterior. Editamos el fichero de configuración de Emacs incluyendo el siguiente fragmento:

```

```

3. Este es el paso 3 y dos comandos asociados (separados, pero se pueden asociar con un solo screen):

```
$ apt-get update
```

```
$ apt-get upgrade
```

Las plantillas más usuales son las siguientes, estas plantillas se pueden almacenar el directorio `./snippetsXML`:

section

Permitirá incluir de manera fácil

Ejemplo de sección

El capítulo tiene una sección, y este es su párrafo introductorio

Es una sección bastante corta antes de que empiece otra más interesante. Mientras podemos porner un enlace a Google: http://www.google.es

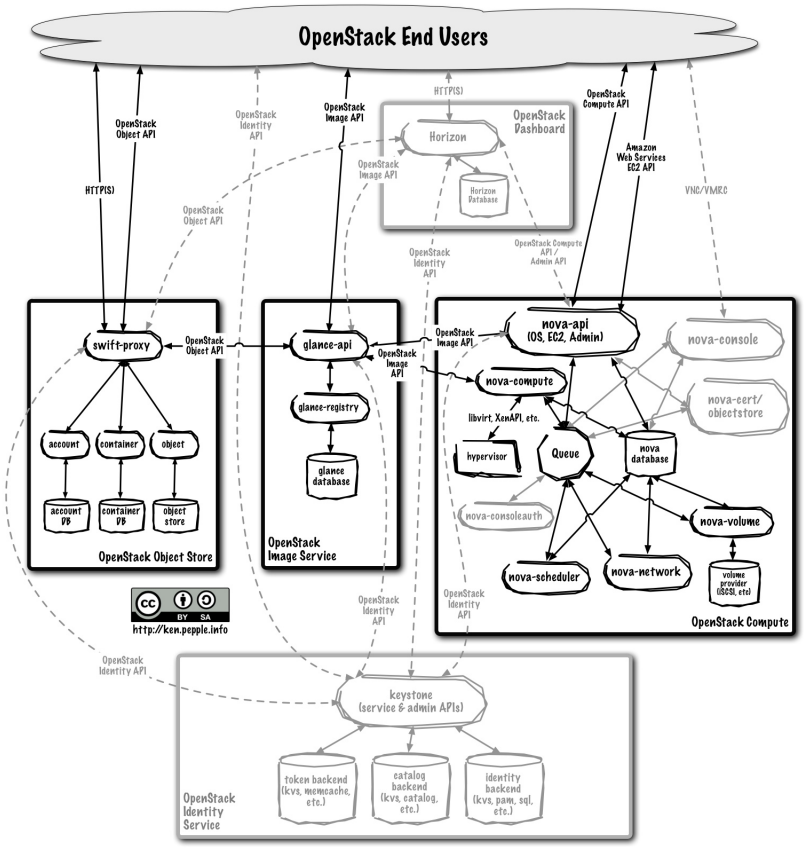
Elementos a insertar...

Al escribir se pueden introducir multitud de elementos como una lista de elementos:

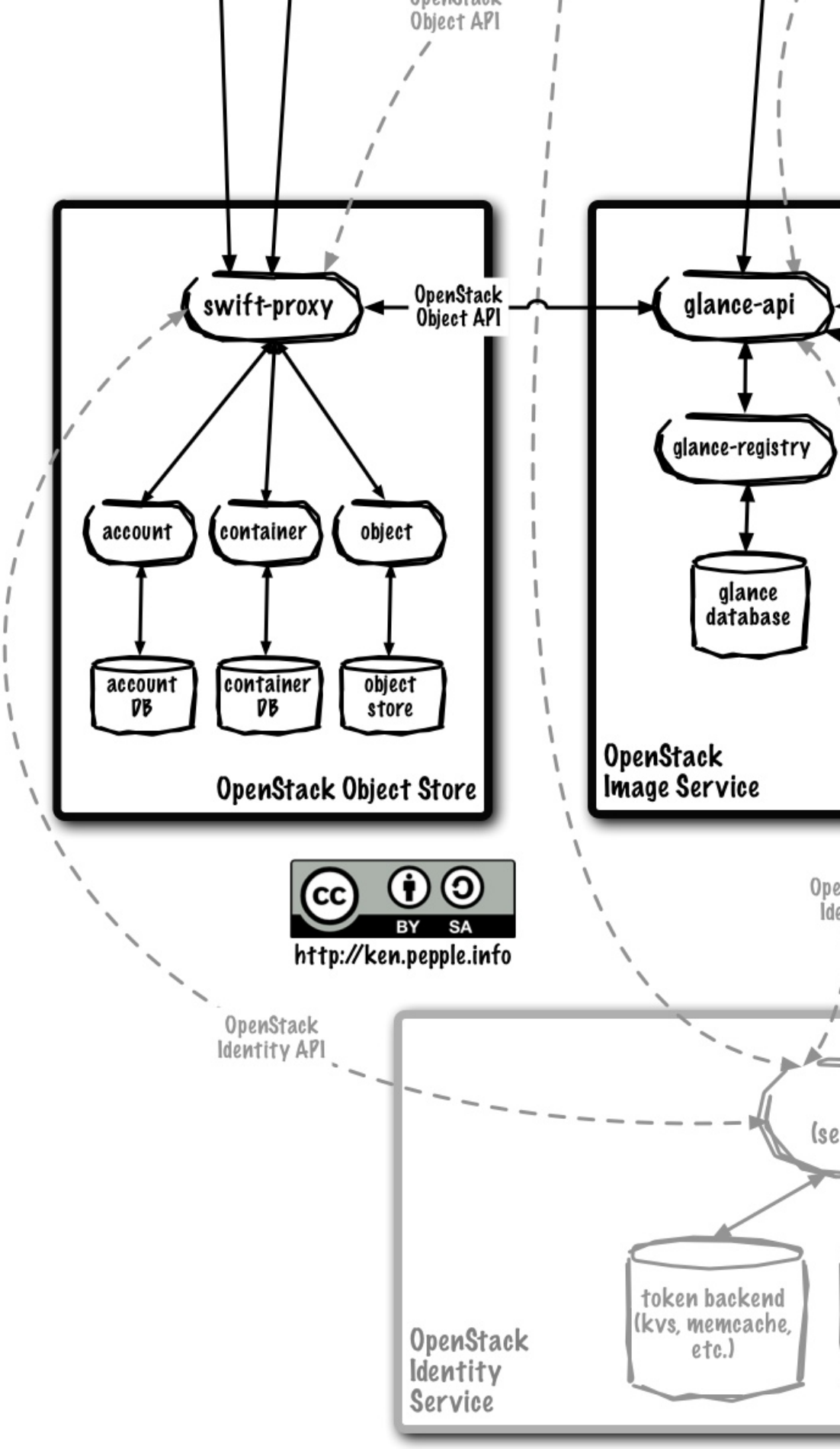
- Este es el elemento 1 de la lista.
- Este es el elemento 2 de la lista.

- Este es el elemento 3 de la lista.
- Y con esto terminamos nuestra lista.

Imágenes:



Pero también se pueden insertar figuras que aparezcan en el índice de figuras así:


Object API
swift-proxy
OpenStack
Object API
glance-api
glance-registry
glance
database
account
container
object
account
DB
container
DB
object
store
OpenStack Object Store
OpenStack
Image Service
BY SA
http://ken.pepple.info
OpenStack
Identity API
token backend
(kvs, memcache,
etc.)
OpenStack
Identity
Service

Una lista de definiciones poniendo en negrita el elemento a definir:

Concepto 1: Y aquí vendría su definición más o menos detallada.

Otro concepto: Y aquí vendría su definición más o menos detallada.

Último concepto: la última definición.

Una tabla (con su entrada en el índice de tablas):

Table 2.2. Tabla de ejemplo:

Encabezado 1	Encabezado 2	Encabezado 3
Fila 1 - columna 1	Una linea Otra linea (esto es optativo)	Fila 1 - columna 3
Fila 2 - columna 1	Fila 2 - columna 2	Fila 2 - columna 3

También podemos estar contando algo y necesitar de una nota:

Note

Pues esto es una nota aclarando algún concepto o algo que haya que aclarar

También es posible introducir un comando que teclearíamos en consola así:

```
$ apt-get install apache2
```

También se pueden utilizar listas ordenadas para explicar procedimientos o un conjunto de pasos para hacer algo:

1. Este es el paso 1
2. Este es el paso 2
3. Este es el paso 3 y dos comandos asociados (separados, pero se pueden asociar con un solo screen):

   ```
   $ apt-get update
   ```

   ```
   $ apt-get upgrade
   ```

Podemos insertar un fichero de código fuente o un fichero de configuración así:

```
#include <stdio.h>

int main(int argc, char **argv)
{
    printf("Hola caracola!!\n");

    return(0);
}
```

O incrustando el código:

```
#!/bin/bash

echo "Hola a todos"
```

También es posible escribir y marcar algo importante como una comando (como **apt-get install**), o como un nombre de fichero `/etc/passwd` o como un literal como `#` o como `int` o como un propietario como `alex.alex`y unos permisos como `0640`.

Cuando se escribe algo importante también se puede utilizar negrita **así**.