

Infraestructura tecnológica virtual con automatización y orquestación.


Arese, Juan Pablo - Diers, Werner Christian

Facultad de Ciencias Exactas, Físicas y Naturales - UNC

Marzo 2017



Organización de la Presentación


- ▶ Introducción
 - ▶ Arquitectura
 - ▶ Desarrollo del sistema
 - ▶ Herramienta de virtualización
 - ▶ Herramienta de aprovisionamiento
 - ▶ Herramienta de orquestación
 - ▶ Interfaz web
 - ▶ Demostración
 - ▶ Conclusión
- 

Introducción



Introducción

Una infraestructura moderna implica:

- ▶ Costos
 - ▶ Virtualización
 - ▶ Rendimiento computacional
 - ▶ Aplicar políticas
 - ▶ Configuraciones establecidas por cada entidad
 - ▶ Estandarización de los parámetros utilizados
 - ▶ Agilidad
- 

Introducción


¿Qué es **virtualización**?

- ▶ Software ejecutándose
- ▶ Concurrencia
- ▶ Aislamiento



Introducción

¿Qué es aprovisionamiento?

- ▶ Proveer o hacer que algo esté disponible
 - ▶ Conjunto de acciones para preparar una máquina virtual
 - ▶ Disco
 - ▶ Memoria RAM
 - ▶ CPU
 - ▶ Sistema operativo
 - ▶ Servicios
 - ▶ Configuración
- 

Introducción

¿Qué es orquestación?

- ▶ Automatizar procesos y flujos de trabajo
- ▶ Integrar servicios
- ▶ Proveer información de forma síncrona o asíncrona



Objetivo



Objetivo

Integrar diferentes herramientas con el fin de implementar técnicas de orquestación, virtualización, instalación y configuración automática para facilitar la gestión de servidores virtuales y sus servicios asociados.



Arquitectura



Arquitectura

La arquitectura implementada es la de cliente - servidor.

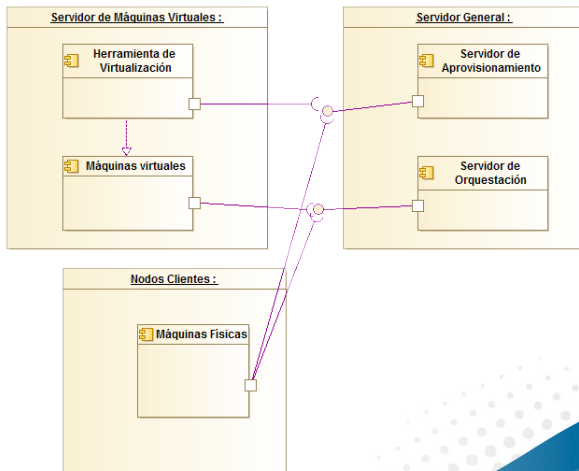
Las tareas del servidor son las siguientes:

- ▶ Crear las máquinas virtuales
- ▶ Asignar direcciones IP por medio del protocolo DHCP
- ▶ Proveer a la máquina con el sistema operativo deseado y los parámetros de configuración establecidos
- ▶ Orquestrar las políticas definidas para una máquina o un conjunto de máquinas



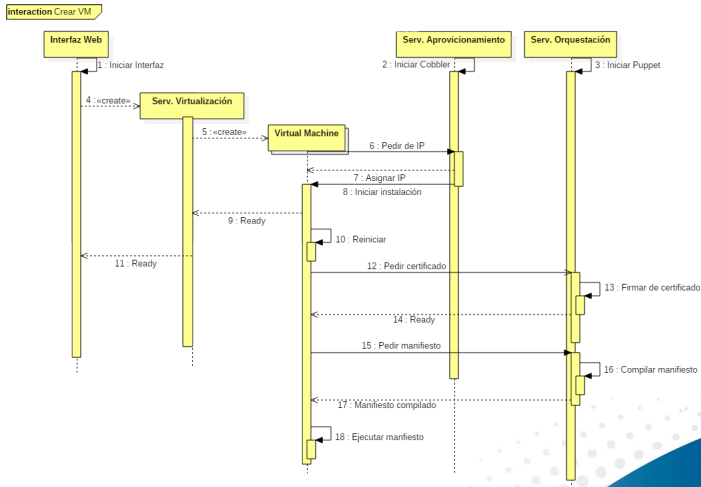
Arquitectura

Arquitectura de desarrollo



Arquitectura

Diagrama de secuencia



Desarrollo



Desarrollo - Interfaz Web

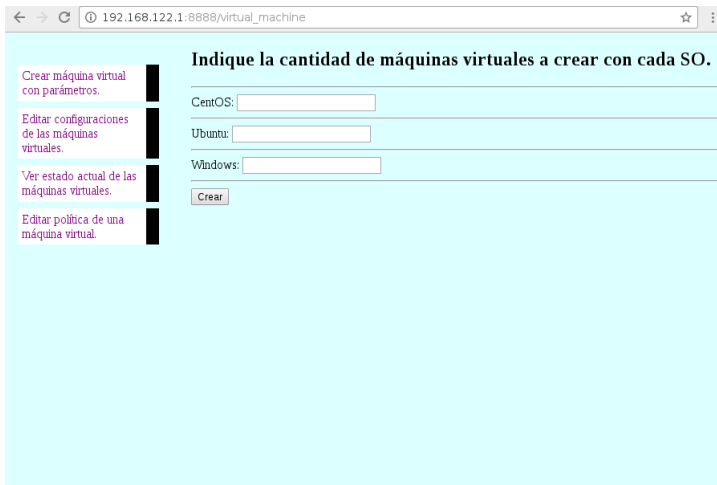
La herramienta utilizada para crear la interfaz web fue Python Bottle. Utiliza una arquitectura cliente - servidor.

- ▶ Es un WSGI (Web Server Gateway Interface) rápido, sencillo y ligero
- ▶ Distribuido como un módulo único
- ▶ Su única dependencia es la Librería Estándar de Python
- ▶ Puede ejecutarse como un servidor web autónomo
- ▶ Plugins para bases de datos populares



Desarrollo - Interfaz Web

Crear múltiples máquinas virtuales



A screenshot of a web browser window displaying a web application for creating virtual machines. The browser's address bar shows the URL "192.168.122.1:8888/virtual_machine". The page has a light blue background. On the left side, there is a vertical menu with four items, each in a light pink box with a black icon to its right: "Crear máquina virtual con parámetros.", "Editar configuraciones de las máquinas virtuales.", "Ver estado actual de las máquinas virtuales.", and "Editar política de una máquina virtual.". The main content area on the right has the heading "Indique la cantidad de máquinas virtuales a crear con cada SO." in bold. Below this heading are three horizontal input fields labeled "CentOS:", "Ubuntu:", and "Windows:". At the bottom of this section is a "Crear" button.

← → ↻ 192.168.122.1:8888/virtual_machine ☆ ⋮

Crear máquina virtual con parámetros.

Editar configuraciones de las máquinas virtuales.

Ver estado actual de las máquinas virtuales.

Editar política de una máquina virtual.

Indique la cantidad de máquinas virtuales a crear con cada SO.

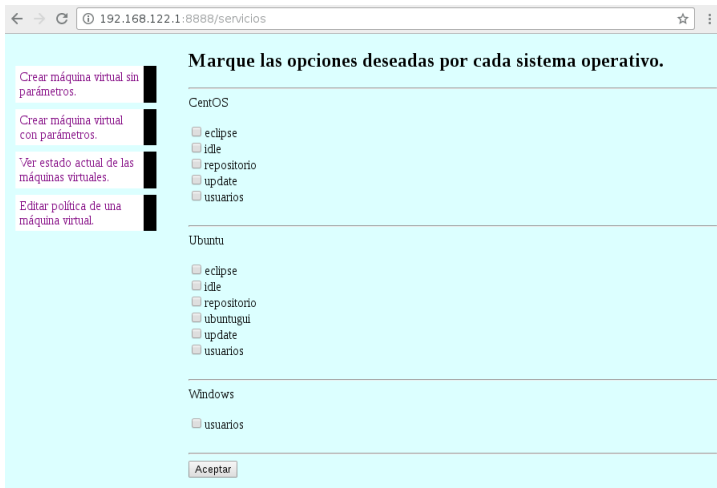
CentOS:

Ubuntu:

Windows:

Desarrollo - Interfaz Web

Aplicar políticas por sistema operativo



← → ↻ ⓘ 192.168.122.1:8888/servicios ☆ ⋮

Crear máquina virtual sin parámetros.

Crear máquina virtual con parámetros.

Ver estado actual de las máquinas virtuales.

Editar política de una máquina virtual.

Marque las opciones deseadas por cada sistema operativo.

CentOS

- ☐ eclipse
- ☐ idle
- ☐ repositorio
- ☐ update
- ☐ usuarios

Ubuntu

- ☐ eclipse
- ☐ idle
- ☐ repositorio
- ☐ ubuntuui
- ☐ update
- ☐ usuarios

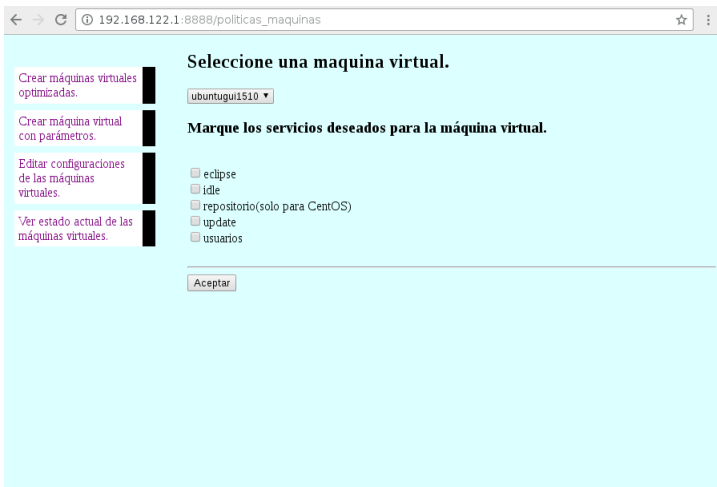
Windows

- ☐ usuarios

Aceptar

Desarrollo - Interfaz Web

Aplicar políticas por máquina específica



The screenshot shows a web browser window with the address bar displaying "192.168.122.1:8888/politicas_maquinas". The page has a light blue background and a sidebar on the left with four menu items: "Crear máquinas virtuales optimizadas.", "Crear máquina virtual con parámetros.", "Editar configuraciones de las máquinas virtuales.", and "Ver estado actual de las máquinas virtuales.". The main content area contains the heading "Seleccione una maquina virtual." followed by a dropdown menu showing "ubuntugui1510". Below this is the heading "Marque los servicios deseados para la máquina virtual." followed by a list of services with checkboxes: "eclipse", "idle", "repositorio(solo para CentOS)", "update", and "usuarios". At the bottom of the main content area is a button labeled "Aceptar".

← → ↻ ⓘ 192.168.122.1:8888/politicas_maquinas ☆ ⋮

Crear máquinas virtuales optimizadas.

Crear máquina virtual con parámetros.

Editar configuraciones de las máquinas virtuales.

Ver estado actual de las máquinas virtuales.

Seleccione una maquina virtual.

ubuntugui1510 ▼

Marque los servicios deseados para la máquina virtual.

- ☐ eclipse
- ☐ idle
- ☐ repositorio(solo para CentOS)
- ☐ update
- ☐ usuarios

Aceptar

Desarrollo - Interfaz Web

Ver estado de las máquinas

← → ↻

192.168.122.1:8888/estados

☆ ⋮

Crear máquinas virtuales optimizadas.

Crear máquina virtual con parámetros.

Editar configuraciones de las máquinas virtuales.

Editar política de una máquina virtual.

Nombre de la VM	Estado	Acción	
ubuntugui1510	ejecutando	<input type="checkbox"/> Encender	<input type="checkbox"/> Apagar
centos366	apagado	<input type="checkbox"/> Encender	<input type="checkbox"/> Apagar
windows1652	apagado	<input type="checkbox"/> Encender	<input type="checkbox"/> Apagar

Ejecutar acción

Desarrollo - Virtualización

La herramienta utilizada para virtualizar fue KVM/Qemu.

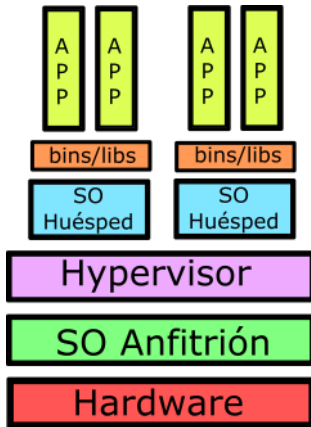
KVM utiliza virtualización completa:

- ▶ El sistema operativo huésped desconoce que está en un entorno virtual
- ▶ El hardware se encuentra virtualizado por el sistema operativo anfitrión
- ▶ La capa de virtualización, el hypervisor, media entre los sistemas huéspedes y el anfitrión



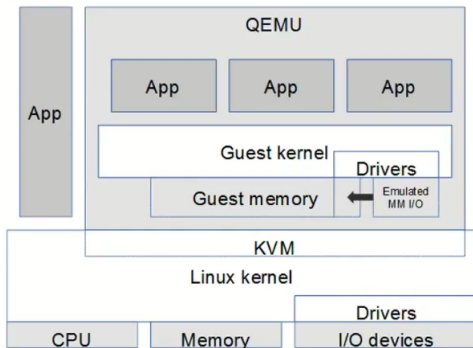
Desarrollo - Virtualización

Esquema de virtualización completa



Desarrollo - Virtualización


Arquitectura de KVM



Desarrollo - Virtualización

Ejemplo de creación de una máquina virtual

```
# virt-install \  
    --connect qemu:///system \  
    --name=centos-vm \  
    --disk path=/var/lib/libvirt/images/centos-vm.qcow2,size=25 \  
    --graphics spice \  
    --vcpus=2 --ram=3072 \  
    --network network=puppet, mac="52:54:00:d5:a1:76" --pxe \  
    --os-type=linux \  
    --os-variant=centos7
```



Desarrollo - Aprovisionamiento

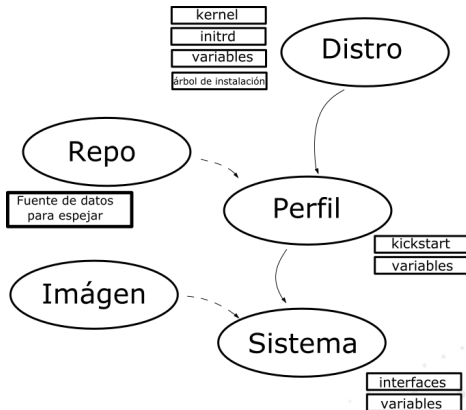
La herramienta utilizada para el aprovisionamiento fue Cobbler.
Utiliza una arquitectura cliente - servidor.

- ▶ El servidor debe ejecutarse en un sistema basado en Unix
- ▶ Centraliza y simplifica el control de servicios incluyendo PXE, DHCP, TFTP y DNS con propósito de realizar instalaciones basadas en red de sistemas operativos
- ▶ Cobbler utiliza objetos para definir la configuración de aprovisionamiento:



Desarrollo - Aprovisionamiento

Modelado de Cobbler



Desarrollo - Aprovisionamiento

- ▶ **Distro:** Distribución que se desea instalar
- ▶ **Repo:** Repositorio, sitio centralizado donde se almacena y mantiene información digital
- ▶ **Perfil:** Asocia una distribución a opciones especializadas adicionales, como puede ser un archivo de configuración
- ▶ **Imágen:** Copia del estado de un sistema computacional, guardado en un archivo o disco
- ▶ **Sistema:** Mapea una pieza de hardware (o una máquina virtual) con el perfil asignado a correr en ella



Desarrollo - Orquestación

La herramienta utilizada para orquestar fue Puppet.

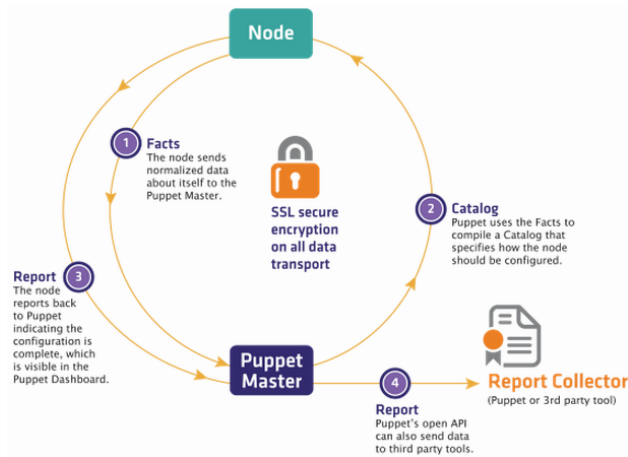
Utiliza una arquitectura cliente - servidor.

- ▶ El servidor (nodo maestro) debe ejecutarse en un sistema basado en Unix
- ▶ Los clientes (agentes) soportan múltiples plataformas
- ▶ Posee su propio DSL (Domain Specific Language)
- ▶ El usuario describe los recursos del sistema y sus estados utilizando un lenguaje declarativo
- ▶ La comunicación entre los agentes y el nodo maestro se realiza bajo el protocolo HTTPS



Desarrollo - Orquestación

Ciclo de Puppet



Desarrollo - Orquestación

Estructura de los módulos de Puppet

```
[root@puppet Modulos Puppet]# tree -C
.
├── eclipse
│   ├── files
│   │   ├── eclipse.desktop.centos
│   │   └── eclipse.desktop.ubuntu
│   └── manifests
│       └── init.pp
├── httpd
│   └── manifests
│       ├── init.pp
│       └── install_httpd.pp
├── idle
│   └── manifests
│       └── init.pp
├── MySQL
│   └── manifests
│       ├── init.pp
│       └── mysql.pp
├── nfs
│   └── manifests
│       ├── ClaseMyNFS.pp
│       ├── Clientenfs.pp
│       ├── client.pp
│       ├── init.pp
│       └── server.pp
├── refrezca_hosts
│   └── manifests
│       └── init.pp
├── repositorio
│   ├── files
│   │   └── proyectointegrador.repo
│   └── manifests
│       └── init.pp
├── ubuntuGUI
│   └── manifests
│       └── init.pp
├── update
│   └── manifests
│       └── init.pp
├── usuarios
│   └── manifests
│       └── init.pp
├── windows_eclipse
│   └── files
│       └── descomprime_eclipse.bat
```

Desarrollo - Orquestación

Ejemplo lenguaje declarativo de Puppet

```
class mysql()
{
    package { 'package_mysql':
        ensure => installed,
        name => "mysql",
    }

    package { 'package_mysql-server':
        ensure => installed,
        name => "mysql-community-server",
    }

    exec{'activo_mysql' :
        command => "chkconfig mysqld on",
        cwd => "/",
        require => Package['package_mysql-server'],
        require => Package['package_mysql'],
    }

    service {'servicio_mysql':
        ensure => running,
        name => "mysqld",
        require => Exec['activo_mysql'],
    }

    exec{'setteo_clave_root_mysql' :
        command => "mysqladmin -u root password 'qwerty'",
        cwd => "/",
        require => Service['servicio_mysql'],
    }

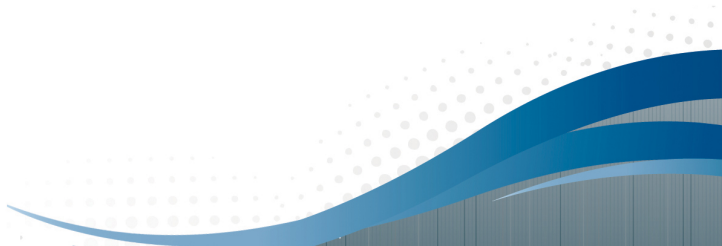
    exec{'creadb_mysql' :
        command => "mysqladmin -uroot -pqwerty create basededatospuppet",
        cwd => "/",
        require => Exec['setteo_clave_root_mysql'],
    }
}
```

Video demostración



Video demostración


- ▶ Creación de las máquinas virtuales
- ▶ Aprovisionamiento de las máquinas con el sistema operativo deseado
- ▶ Orquestrar las políticas definidas para una máquina o un conjunto de máquinas



Conclusiones



Conclusiones

- ▶ Elección del entorno realizado utilizando factores de decisión ponderados
 - ▶ Solución modularizada
 - ▶ Permite la escalabilidad necesaria
 - ▶ Soluciones de código abierto no siempre permiten estar en la "cresta de la ola"
 - ▶ El sistema final cumple con los requerimientos
- 

Trabajos Futuros



Trabajos Futuros

- ▶ Protección:
 - ▶ Modificar el sistema para que funcione con firewall y SELinux
 - ▶ Incluir autenticación por usuario en la interfaz web
 - ▶ Incluir un log de cambios al sistema que permita saber quién y qué cambio realizó
- ▶ Migración: Poder realizar el traslado de hosts virtuales entre las diferentes máquinas físicas



Preguntas



Muchas Gracias!

