

VirtShell

Framework para aprovisionamiento de soluciones virtuales

Carlos Alberto Llano R.

Escuela de Ingeniería de Sistemas y Computación
Maestría en Ingeniería con énfasis en Ingeniería de Sistemas y Computación

Director:

John Alexander Sanabria

August 25, 2016



Contents

- ① Propósitos alcanzados
- ② Flujo de aprovisionamiento
- ③ Experiencia y Evaluación
- ④ Conclusiones
- ⑤ Recomendaciones

Objetivos

General

Diseñar un framework web, que permita el aprovisionamiento de software automático, para ambientes virtualizados.

Objetivos

General

Diseñar un framework web, que permita el aprovisionamiento de software automático, para ambientes virtualizados.

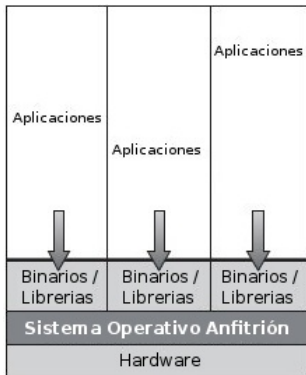
Específicos

- 1 Evaluar diferentes técnicas y soluciones de aprovisionamiento que se utilizan en la actualidad.
- 2 Evaluar diferentes mecanismos de aprovisionamiento.
- 3 Realizar una ejemplificación del framework.



Técnicas y soluciones de aprovisionamiento actuales (Objetivo #1)

Técnicas de Virtualización trabajadas

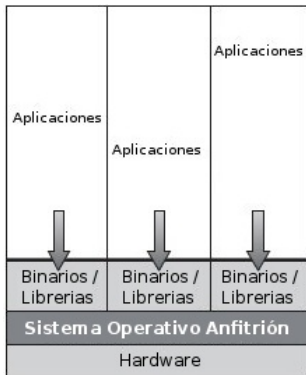


Contenedores

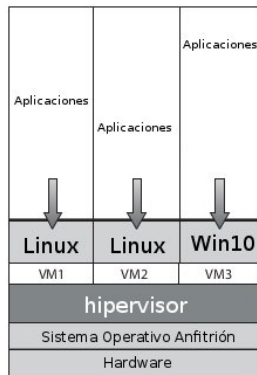


Técnicas y soluciones de aprovisionamiento actuales (Objetivo #1)

Técnicas de Virtualización trabajadas



Contenedores



Máquinas Virtuales



Soluciones de aprovisionamiento evaluadas

- Fabric
- Chef
- Puppet
- Juju
- CFEngine
- Bcfg2
- Ansible
- Cobbler
- SmartFrog
- Amazon EC2
- Docker composer
- SaltStack
- Vagrant



Técnicas y soluciones de aprovisionamiento actuales (Objetivo #1)

Soluciones de aprovisionamiento evaluadas

Solución	Sop.Nubes	Curv.Apren	Crea	Aprov	API REST	Mult.SO
Chef	Todas	Alta	✓	✓	✓	✓
Juju	OpenStack MAAS	Baja	✓	✓	✗	✗
Puppet	La mayoría	Media	✗	✓	✗	✓
Ansible	Amazon OpenStack	Baja	✗	✓	✗	✓
Amazon EC2	Amazon	Media	✓	✓	✓	✓
Docker	✗	Media	✓	✓	✓	✓
Vagrant	✗	Baja	✓	✓	✗	✓



Evaluar diferentes mecanismos de aprovisionamiento. (Objetivo #2)

Selección del mecanismo de aprovisionamiento

Se evaluaron diferentes formas de escritura de scripts de aprovisionamiento

- json
- xml
- yaml
- archivos texto
- lenguajes de programación (ruby, python, etc.)
- bash



Evaluar diferentes mecanismos de aprovisionamiento. (Objetivo #2)

Selección del mecanismo de aprovisionamiento





Evaluar diferentes mecanismos de aprovisionamiento. (Objetivo #2)

Selección del mecanismo de aprovisionamiento



```
#!/bin/bash
```



Evaluar diferentes mecanismos de aprovisionamiento. (Objetivo #2)

Selección del mecanismo de aprovisionamiento



Bash coming to Windows



Evaluar diferentes mecanismos de aprovisionamiento. (Objetivo #2)

Selección del mecanismo de aprovisionamiento

Se evaluaron diferentes formas de guardar los scripts en el sistema

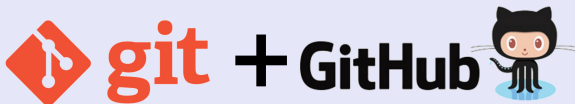
- repositorio central propio?
- enviarlo en el momento del aprovisionamiento?



Evaluar diferentes mecanismos de aprovisionamiento. (Objetivo #2)

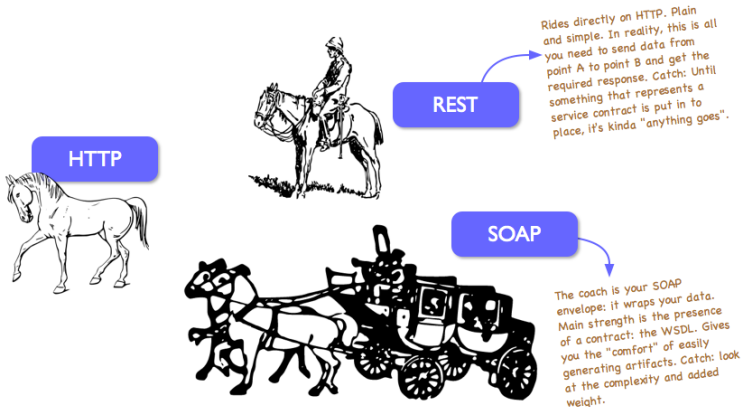
Selección del mecanismo de aprovisionamiento

Solución



Evaluar diferentes mecanismos de aprovisionamiento. (Objetivo #2)

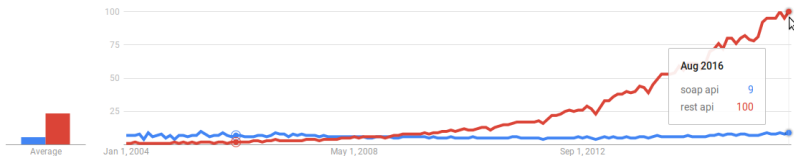
Evaluación y selección de un estilo arquitectural





Evaluar diferentes mecanismos de aprovisionamiento. (Objetivo #2)

Evaluación y selección de un estilo arquitectural

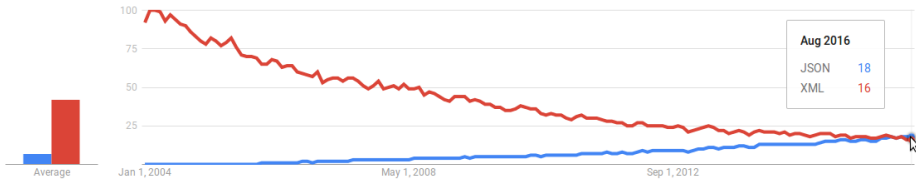


Fuente: Google Trends



Evaluar diferentes mecanismos de aprovisionamiento. (Objetivo #2)

Evaluación y selección de un estilo arquitectural

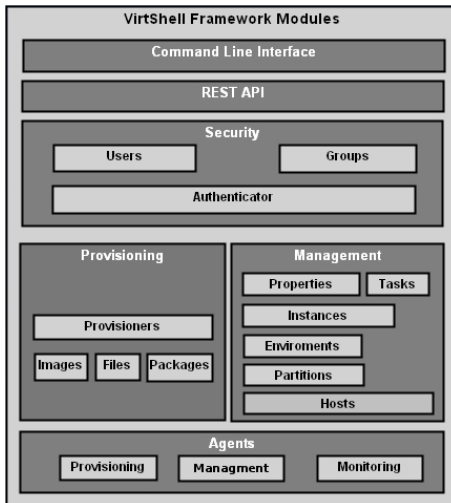


Fuente: Google Trends



Realizar una ejemplificación del framework. (Objetivo #3)

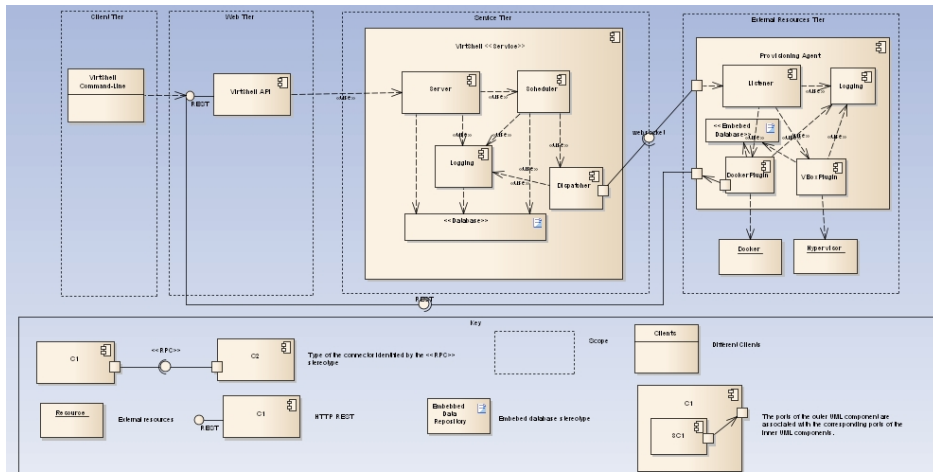
Framework





Realizar una ejemplificación del framework. (Objetivo #3)

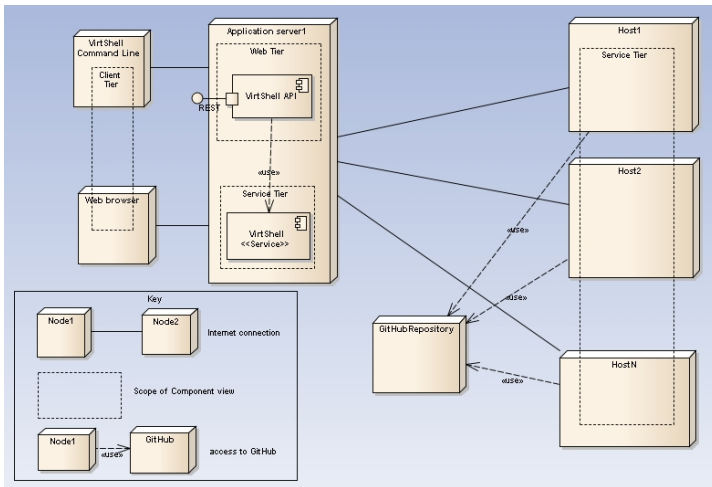
Vista de componentes





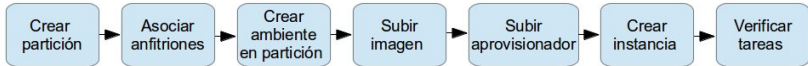
Realizar una ejemplificación del framework. (Objetivo #3)

Vista de deployment



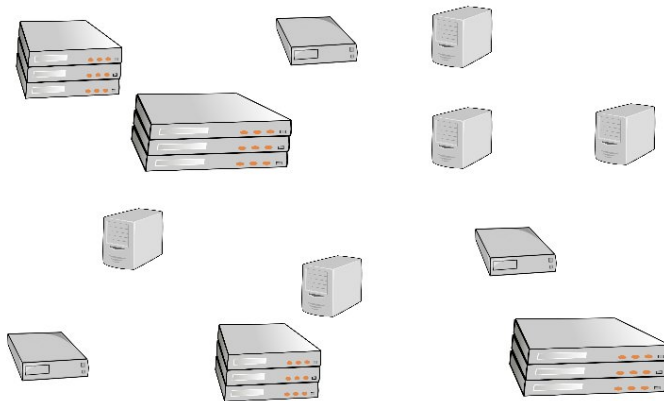
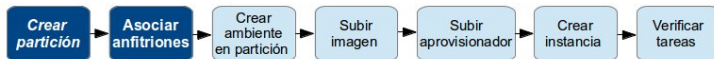
ooo
ooooooooo
ooo

Flujo de aprovisionamiento



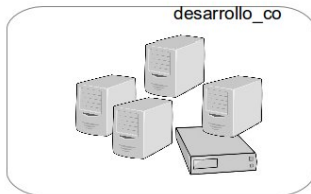
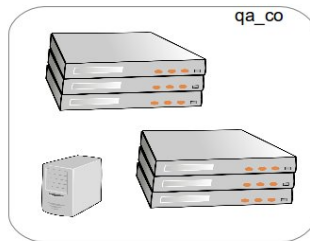
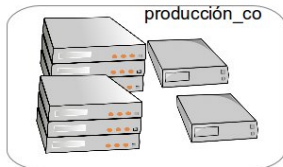
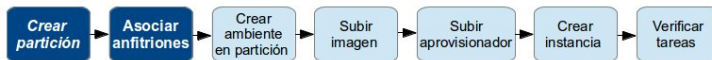
ooo
ooooooooo
ooo

Particiones y Anfitriones



ooo
ooooooooo
ooo

Particiones y Anfitriones

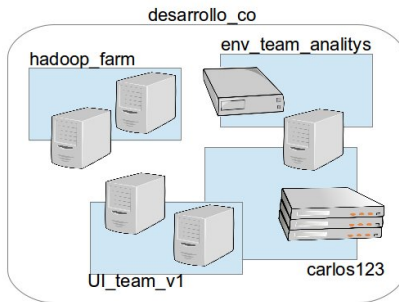
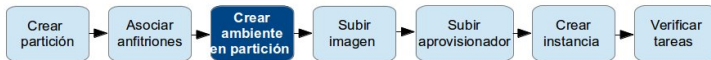


Particiones y Anfitriones (Ejemplo)

```
1 curl -X POST http://virtshellsrv:80/partitions/  
   -d "{\"name\":\"development_co\",  
3     \"description\":\"Collection of servers oriented to  
       development team in Colombia.\"}"  
   -H "accept:application/json" | jq .
```


ooo
ooooooooo
ooo

Ambientes de trabajo



ooo
ooooooooo
ooo

Ambientes de trabajo (Ejemplo)

```
curl -X POST http://virtshellsrv:80/enviroments/  
-d "{\"name\": \"development\",  
    \"description\": \"Development enviroment\",  
    \"partition\": \"development_co\",  
    \"users\": [{\"login\": \"development_user\",  
                {\"login\": \"guest\"}]}"  
-H "accept: application/json" | jq .
```

ooo
ooooooooo
ooo

Imagenes

ooo
ooooooooo
ooo

Aprovisionadores

ooo
ooooooooo
ooo

Creación de instancias

ooo
oooooooo
ooo

Chequeo de tareas

ooo
ooooooooo
ooo

Demo



Experiencia y Evaluación

En las pruebas realizadas VirtShell demostró que parece ser una herramienta útil para aprovisionar software de manera sencilla y fiable.

La experiencia adquirida con la primera versión es la siguiente:

- *VirtShell funciona.*
- *Aprovisionar ambientes virtuales via web usando scripts escritos en el lenguaje que prefiera si es posible.*
- *El aprovisionamiento de máquinas virtuales o contenedores es prácticamente el mismo.*

Experiencia y Evaluación

La primera versión de VirtShell fue desarrollada en el lenguaje Python (versión 3)

Se encuentra alojada en el repositorio git:

<https://github.com/janutechnology/VirtShell>.

Esta versión inicial aun no esta terminada y se encuentra en continuo desarrollo para lograr tener todas las funcionalidades funcionando.

ooo
oooooooo
ooo

Conclusiones

ooo
ooooooooo
ooo

Recomendaciones

ooo
oooooooo
ooo

Preguntas?

