

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**GRADO EN INGENIERÍA EN TECNOLOGÍAS Y
SERVICIOS DE TELECOMUNICACIÓN**

TRABAJO FIN DE GRADO

**DESPLIEGUE AUTOMATIZADO DE
ESCENARIOS DE RED VIRTUALIZADOS**

SAMUEL GARCÍA SÁNCHEZ

2022

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**GRADO EN INGENIERÍA EN TECNOLOGÍAS Y
SERVICIOS DE TELECOMUNICACIÓN**

TRABAJO FIN DE GRADO

**DESPLIEGUE AUTOMATIZADO DE
ESCENARIOS DE RED VIRTUALIZADOS**

**Autor
SAMUEL GARCÍA SÁNCHEZ**

**Tutor
MARIO SANZ RODRIGO**

2022

Resumen

Ejemplo resumen **Nombre:** Tecnológica Ecosistemas SAU [Accenture](#) Rebus sic stantibus¹

¹os la estoy metiendo doblada

Abstract

Ejemplo de código en bash:

```
1  #!/bin/bash
2  #Enumeracion de los puertos abiertos de una maquina
3  for port in $(seq 1 65535); do
4      timeout 1 bash -c "echo > /dev/tcp/10.10.10.52/$port" > /dev/null
5      2>&1 && echo "$port/tcp" &
6  done; wait
```

Código 1: Bash example

Referencia a la figura 2.1 de la página 3

Índice general

Resumen	I
Abstract	II
Índice de figuras	IV
1. Introducción	1
2. Estado del arte	2
2.1. Tecnologías de virtualización	2
2.1.1. Virtualización mediante hipervisor	2
2.1.2. Virtualización en contenedores	6
2.2. Tecnologías de aprovisionamiento	6
2.2.1. Aprovisionamiento estático	6
2.2.2. Aprovisionamiento dinámico	6
2.3. Tecnologías de orquestación	6
Bibliografía	7

Índice de figuras

2.1. Tipos de hipervisor	3
2.2. Logo de KVM	4
2.3. Logo de VirtualBox	4
2.4. Logo de VMware	5

Capítulo 1

Introducción

El código empleado en este trabajo está disponible en el siguiente repositorio de GitHub:

Dirección URL

<https://github.com/samugs13/DAERV>

Capítulo 2

Estado del arte

2.1. Tecnologías de virtualización

Se podría decir que la virtualización es ya uno de los pilares fundamentales del mundo IT debido a las grandes ventajas que proporciona. Previo al desarrollo de las tecnologías y tipos de virtualización disponibles, es conveniente explicar en qué consiste la virtualización, que no es más que una representación mediante software de un entorno físico o recurso tecnológico, como pueden ser aplicaciones, servidores o almacenamiento. [1]

Gracias a esta tecnología, es posible contar con varios ordenadores virtuales en el mismo hardware, donde cada uno de ellos puede interactuar de forma independiente y ejecutar sistemas operativos o aplicaciones diferentes mientras comparten los recursos de una sola máquina host. Al crear varios recursos a partir de un único equipo o servidor, la virtualización mejora la escalabilidad y las cargas de trabajo, al tiempo que permite usar menos servidores y reducir el consumo de energía, los costos de infraestructura y el mantenimiento.

En función del sistema a simular, podemos encontrar diferentes categorías [2], un ejemplo es la virtualización de red, que consiste en crear redes virtuales sobre redes físicas o reproducir completamente redes físicas en software. Otro ejemplo sería la virtualización de almacenamiento, que combina varios dispositivos de almacenamiento en red, con la apariencia de una única unidad o dispositivo de almacenamiento, accesible por varios usuarios. Podríamos enumerar más tipos de virtualización, pero en lo que a este trabajo respecta vamos a centrarnos en la virtualización de software, que separa las aplicaciones del hardware y el sistema operativo, y en la que distinguimos dos subtipos: virtualización mediante hipervisor y virtualización en contenedores.

2.1.1. Virtualización mediante hipervisor

Una máquina virtual es un software que ejecuta programas o procesos como si fuera la máquina física. Es decir, se abstrae el hardware y se representa con una capa de software que proporciona una interfaz igual que el hardware, de forma que sobre ella podemos instalar uno o varios sistemas operativos invitados o guests distintos. Esta capa de software



también se encarga de repartir y aislar los recursos del host entre las VM¹, de manera que el host queda protegido si falla una VM, y las VM estén protegidas entre ellas. Pues bien, cuando hablamos de esta capa de software estamos hablando de lo que se conoce como hipervisor.

Como ya se ha mencionado, un hipervisor es una capa intermedia de software que permite al ordenador anfitrión prestar soporte a varias máquinas virtuales mediante el uso compartido de sus recursos. Cuando se ejecuta una instrucción en el SO² invitado, el hipervisor la coje y la ejecuta en el SO anfitrión. En este proceso, el SO no diferencia entre ejecutar procesos en la máquina virtual o en la física, lo que representa plenamente el concepto de virtualización.

Dentro de los hipervisores [3], podemos distinguir dos tipos. El primero es el Tipo 1, conocido también como hipervisor nativo o bare-metal. Este hipervisor se ejecuta directamente sobre el hardware en lugar de un SO clásico. Todos los hipervisores necesitan algunos elementos del sistema operativo (por ejemplo, el administrador de memoria, el programador de procesos, la pila de entrada o salida [E/S], los controladores de dispositivos, entre otros) para ejecutar las máquinas virtuales. Por tanto, este hipervisor es equivalente a un SO con un poco de información adicional que le permite gestionar los SO invitados. Es muy común encontrarlos en centros de datos, por la eficiencia que supone el ahorrar una capa de software.

Los hipervisores de Tipo 2 se ejecutan sobre el SO anfitrión como una capa de software o aplicación. Están orientados a usuarios individuales que buscan ejecutar varios SO en el mismo ordenador. La ejecución de una VM sobre un hipervisor de este tipo es más lenta que en un hipervisor de Tipo 1.

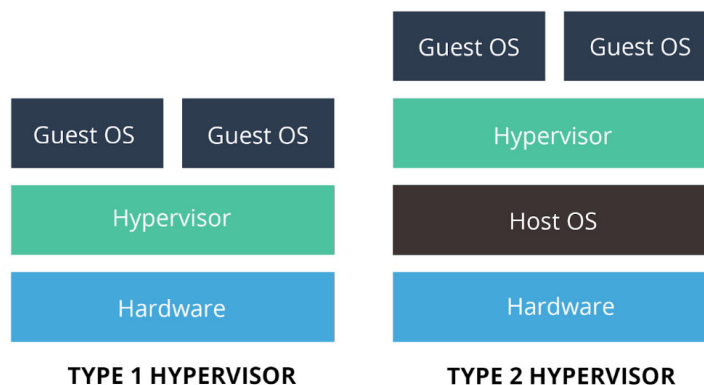


Figura 2.1: Tipos de hipervisor

KVM

KVM (Kernel Virtual Machine) [4] es una tecnología de virtualización open source que convierte el kernel de Linux en un hipervisor de Tipo 1 que se puede usar para la virtualización. Las KVM tienen todos los elementos necesarios de un SO porque forman parte del kernel de Linux. Cada máquina virtual se implementa como un proceso habitual de Linux. Al ser un hipervisor de Tipo 1 ofrece un mejor rendimiento.

¹Virtual Machine

²Sistema Operativo



Figura 2.2: Logo de KVM

La configuración de la máquina virtual creada se almacena internamente en un fichero XML, el cual es posible editar manualmente a posteriori si se quiere hacer algún cambio. KVM nos permite disfrutar de las ventajas del software open source: no habrá restricciones en cuanto a integración, como sí puede haberlas si se usa un software propietario como VMware; y es independiente de proveedores. Es posible instalar una GUI³ como virt-manager, que se apoya en la biblioteca libvirt (API de virtualización estándar de Linux), para facilitar su uso.

VirtualBox

VirtualBox [5] es desarrollado por Oracle, aunque es gratuito y open source al igual que KVM. Es un hipervisor de Tipo 2, por lo que ofrece un rendimiento inferior comparado con un Tipo 1. VirtualBox permite crear y cargar máquinas virtuales de una forma muy sencilla, lo que hace que sea la alternativa elegida por muchos usuarios. Su asistente ofrece algunos valores sugeridos para tipos específicos de máquinas virtuales durante la creación de estas, pero su gestión final se produce en una configuración posterior.



Figura 2.3: Logo de VirtualBox

Una ventaja de VirtualBox son las instantáneas, que permiten tomar una imagen de la máquina virtual en un momento dado. La imagen conserva la máquina virtual, lo que permite volver a ese momento específico.

VirtualBox ofrece un soporte muy completo. Tiene versiones para Windows, Linux, Macintosh y Solaris, y puede ejecutar un amplísimo número de sistemas operativos invitados, incluidos Windows, macOS, Linux, DOS, Solaris u OpenBSD.

³Graphical User Interface

VMWare

VMware [6] es un software comercial, lo que significa que si queremos aprovechar al máximo todas sus herramientas y configuraciones, debemos pagar por la licencia de uso. Aquí vamos a hablar de VMware Workstation Player, que es el producto gratuito de VMware para virtualización de máquinas, orientado para uso personal, doméstico y sistema educativo.



Figura 2.4: Logo de VMware

En comparación con VirtualBox, VMware Workstation Player es una experiencia más fluida y ágil, y ofrece mejor soporte y estabilidad para una amplia gama de hardware. Está disponible para Windows y Linux y también admite todo tipo de sistemas invitados. [7]

Además, VMware Workstation Player sí permite personalizar toda la configuración durante el proceso de creación de la máquina virtual. La diferencia no es mucha, pero significa que la máquina virtual está lista para ejecutarse después de finalizar el asistente, en lugar de tener que realizar más configuraciones una vez que se completa.

Por el contrario, no admite instantáneas o puntos de control. Puede suspender temporalmente el sistema operativo invitado para reanudar desde un punto específico, pero no es tan completo como la creación de un historial de imágenes para la máquina virtual.



2.1.2. Virtualización en contenedores

LXC

Docker

2.2. Tecnologías de aprovisionamiento

2.2.1. Aprovisionamiento estático

Docker

Vagrant

2.2.2. Aprovisionamiento dinámico

Chef

Ansible

2.3. Tecnologías de orquestación

Docker Compose

Kubernetes

Terraform

A

Bibliografía

- [1] VMWare. *¿En qué consiste la virtualización?* [Online]. URL: <https://www.vmware.com/es/solutions/virtualization.html>.
- [2] Microsoft. *¿Qué es virtualización?* [Online]. URL: <https://azure.microsoft.com/es-es/overview/what-is-virtualization/>.
- [3] Red Hat. *¿Qué es un hipervisor?* [Online]. URL: <https://www.redhat.com/es/topics/virtualization/what-is-a-hypervisor>.
- [4] Linux KVM. *Documentación Oficial*. [Online]. URL: <https://www.linux-kvm.org/>.
- [5] VirtualBox. *Documentación Oficial*. [Online]. URL: <https://www.virtualbox.org/>.
- [6] VMware. *Documentación Oficial*. [Online]. URL: <https://www.vmware.com/>.
- [7] MuyComputer. *¿Cuáles son las diferencias entre VirtualBox, VMWare e Hyper-V?* [Online]. URL: <https://www.muycomputer.com/2020/03/27/hipervisor-virtualbox-vmware-hyperv/>.