UT 6: Despliegue Aplicaciones WAMP/LAMP

2ºDAW – Despliegue Aplicaciones Web

WAMP

- ✓ Sistema Operativo : **Windows**
- ✓ Servidor Web: **Apache**
- ✓ Sistema Gestor Base Datos: **MySQL**
- ✓ Lenguaje Programación: **PHP, Perl** o **Python**

LAMP

- ✓ Sistema Operativo : **Linux**
- ✓ Servidor Web: **Apache**
- ✓ Sistema Gestor Base Datos: **MySQL**
- ✓ Lenguaje Programación: **PHP, Perl** o **Python**

Arquitectura

Instalación

- ✓ Instalar y configurar cada paquete por separado
- ✓ Instalar paquetes que integran todos los componentes:
  - XAMPP
  - WAMP Server
  - Appserv
  - EasyPHP
  - Uniform Server

Entornos

- ✓ Entorno Desarrollo
- ✓ Entorno Integración
- ✓ Entorno Pre-Producción
- ✓ Entorno Producción
- ✓ Entorno Pruebas

Despliegues

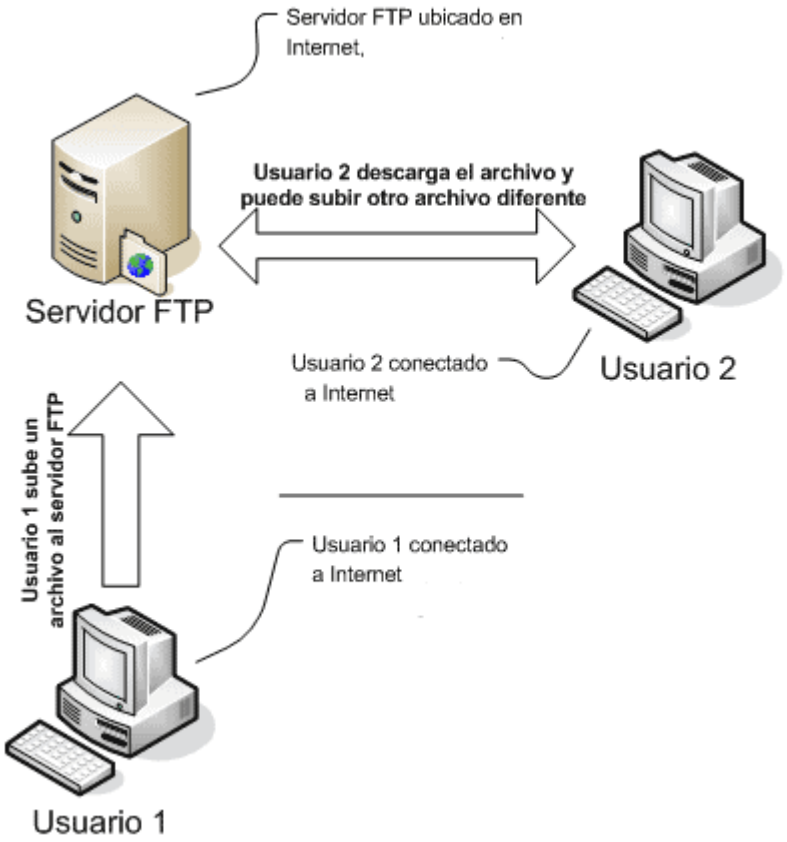
- ✓ Utilizar un instalador repositorios (ej. **PHPMyAdmin**)
- ✓ Instalar desde repositorios de código (ej, **git**, **subversion**, **cvs**)
- ✓ Procedimiento general:

  1) Obtener el código fuente de la aplicación (normalmente comprimido).
  2) Descomprimir el archivo en un directorio accesible por el servidor web.
  3) Dependiendo de donde se coloque la aplicación configurar el servidor web para que pueda servir los archivos.
  4) Configurar el sistema gestor de bases de datos para que permita conexiones de la aplicación.
  5) Crear la base de datos de la aplicación , ejecutar un script o acceder a una página de la aplicación que cree la base de datos.

Servicio FTP

Redes TCP/IP permiten la transferencia información entre equipos. Uno servicios es FTP (File Transfer Protocol).

Un servidor FTP es un programa especial que se ejecuta en un servidor conectado normalmente en Internet. La función del mismo es permitir el desplazamiento de datos entre diferentes servidores / ordenadores.



Servicio FTP

Protocolo FTP nivel aplicación:

- Acceso a sistemas remotos ficheros
- Transferir ficheros a/desde sistemas remotos ficheros
- Crear estructuras directorio en sistemas remotos ficheros

Sigue modelo Cliente/Servidor:

- **Clientes FTP**: establecen conexiones con los servidores FTP

- **Servidores FTP**: acceden al sistema ficheros y controlan conexiones con los clientes en función privilegios del usuario que se conecta para la descarga/subida archivos

- **Protocolo FTP**: conjunto normas que rigen el diálogo entre clientes y servidores FTP

Servicio FTP

Servidor FTP proporciona servicio a través de dos puertos:

- El puerto 20, para la transferencia de datos.
- El puerto 21, para la transferencia de órdenes (control).

El cliente se conecta al servidor:

- Usando un puerto local (origen) mayor que 1024

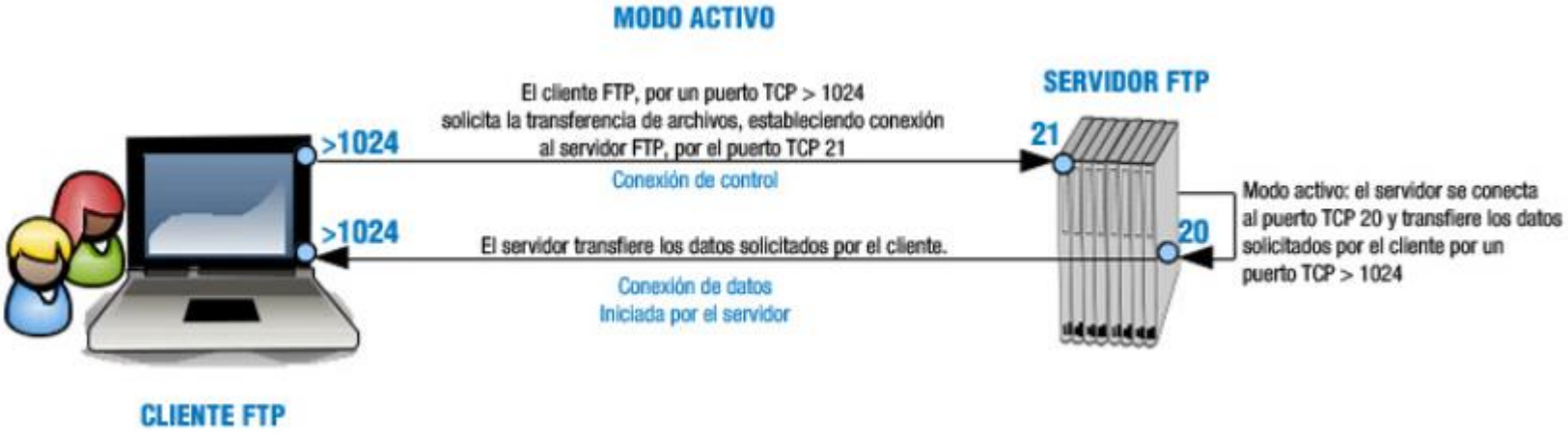
- Usando como puerto destino el puerto 21 del servidor. Una vez se ha establecido la conexión ya se puede hacer uso de las órdenes específicas FTP de manejo de archivos y directorios remotos.

Servicio FTP - Modos de Conexión

Servicio FTP soporta dos modos de conexión

Modo Activo

1. Cliente FTP comienza la conexión desde un puerto de control P(>1024) con destino al puerto 21 del servidor.
2. Servidor responde desde el puerto de control.
3. Servidor inicia entonces la conexión del canal de datos: puerto 20 para el servidor y puerto P+1 para el cliente.
4. Cliente responde desde el puerto de datos estableciendo así la conexión.



Problema: Servidor inicia conexión datos en cliente → cortafuegos en el lado del cliente normalmente rechazan las conexiones entrantes desde servidores FTP en modo activo

Servicio FTP - Modos de Conexión

Servicio FTP soporta dos modos de conexión

Modo Pasivo

1. Cliente FTP comienza la conexión desde un puerto de control P(>1024) con destino al puerto 21 del servidor.
2. Servidor responde desde el puerto de control.
3. El cliente inicia la conexión del canal de datos dese el puerto P+1 hacia el puerto Q del servidor.
4. El servidor responde desde el puerto de datos estableciendo así la conexión.

Servicio SSH

SSH (Secure SHell) protocolo que permite a los usuarios conectarse a un host remotamente, utilizando una arquitectura cliente/servidor y permitiendo conexiones seguras.

A diferencia de otros protocolos de comunicación remota tales como FTP o Telnet, SSH encripta la sesión de conexión.

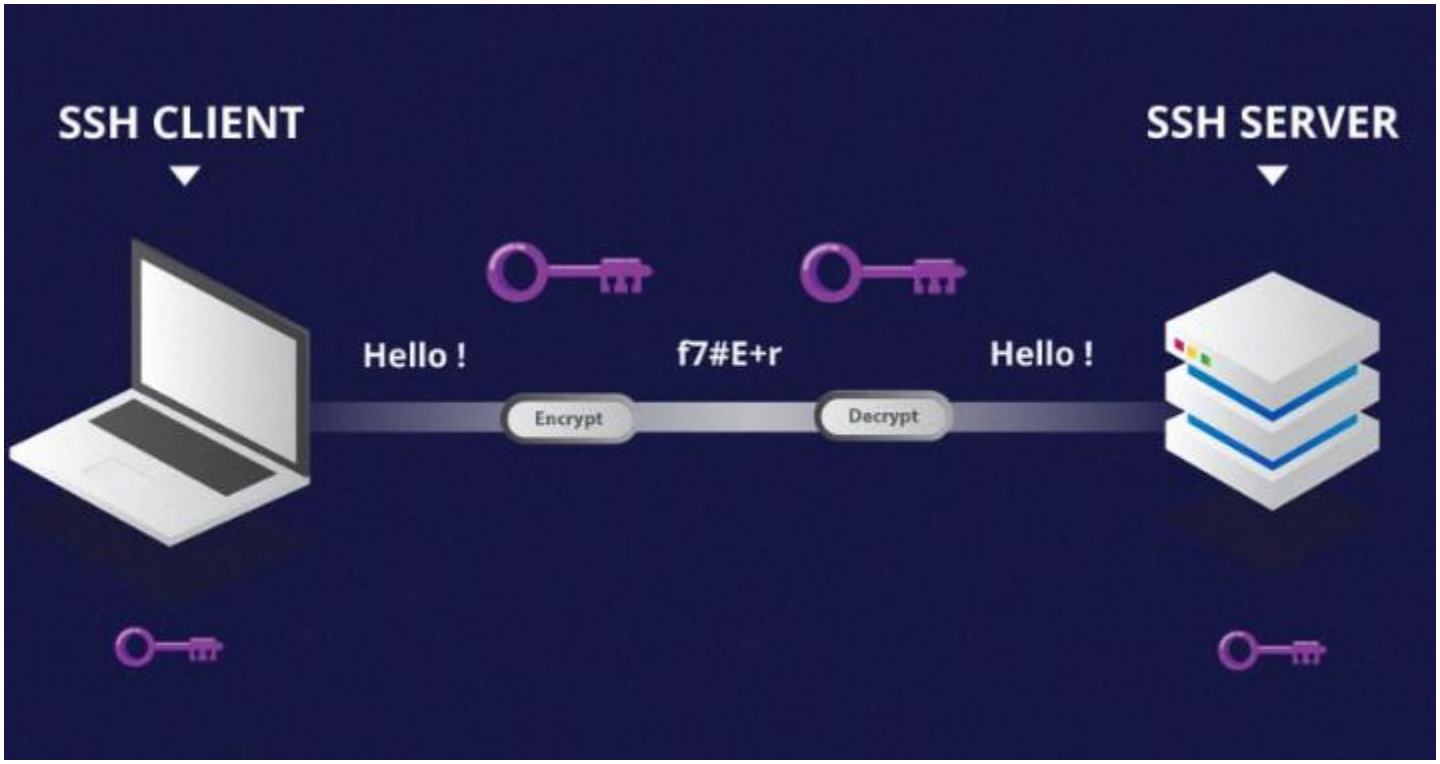
Todos los datos enviados y recibidos durante la conexión, así como la informacion de autenticación del cliente, se transfieren por medio de encriptación de 128 bits, lo cual los hacen extremamente difícil de descifrar y leer.

Muchas aplicaciones SSH cliente están disponibles para casi todos los principales sistemas operativos en uso hoy día.

En Windows, se utiliza herramienta Putty para realizar conexiones SSH.

Por otro lado en Linux y Unix normalmente ya viene instalado la aplicación SSH.

Servicio SSH



Servicio SSH

SSH utiliza algoritmos encriptación para garantizar:

- **Confidencialidad**: nadie que no sea legítimo destinatario puede acceder a la información.
- **Integridad**: información no puede ser alterada tránsito del Emisor al Receptor.
- **Autenticación**: tanto Emisor como Receptor pueden confirmar la identidad del otro.
- **No Rechazo**: el emisor de la información no puede negar que es al autor

SSH utiliza:

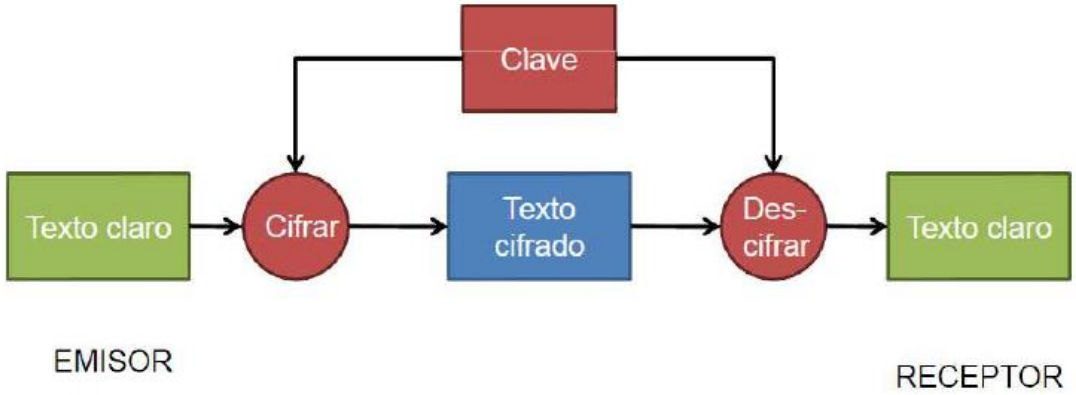
- **Algortimos encriptación Simétricos** (o de clave compartida) para transferencia de datos al ser más rápidos.

- **Algortimos encriptación Asimétricos** (o de clave pública/privada) para establecer conexión con máquina remota.

SSH – Algoritmos de Clave Simétrica

Algoritmos de clave simétrica

Se usa la misma clave para cifrar y para descifrar la información. La seguridad está en la clave no en el algoritmo. Este cifrado solo ofrece **confidencialidad**



Los principales problemas de los sistemas de cifrado son:

- El **intercambio de claves**: una vez que el remitente y el destinatario hayan intercambiado las claves pueden usarlas para comunicarse con seguridad (no hay nada menos secreto que un secreto compartido).

- El **número de claves** que se necesitan: si tenemos n personas que necesitan comunicarse entre sí, se necesitan n/2 claves diferentes. Esto puede funcionar en un grupo reducido, pero no en grupos grandes

SSH – Algoritmos de Clave Simétrica

Las **ventajas** que ofrece:

- Eficiente (los algoritmos utilizados son muy rápidos).

Ejemplos de algoritmos simétricos: DES, Triple DES (3DES), IDEA, AES, BLOWFISH, RC4, RC5

Usos principales (aplicaciones):

- Transmisión de datos sobre un canal inseguro (emails, …).
- Almacenamiento de datos (ficheros, particiones, bases de datos).

Método de ataque:

- Fuerza bruta.
- Para que el ataque sea computacionalmente irrealizable se recomienda una longitud mínima de 128 bits de clave.

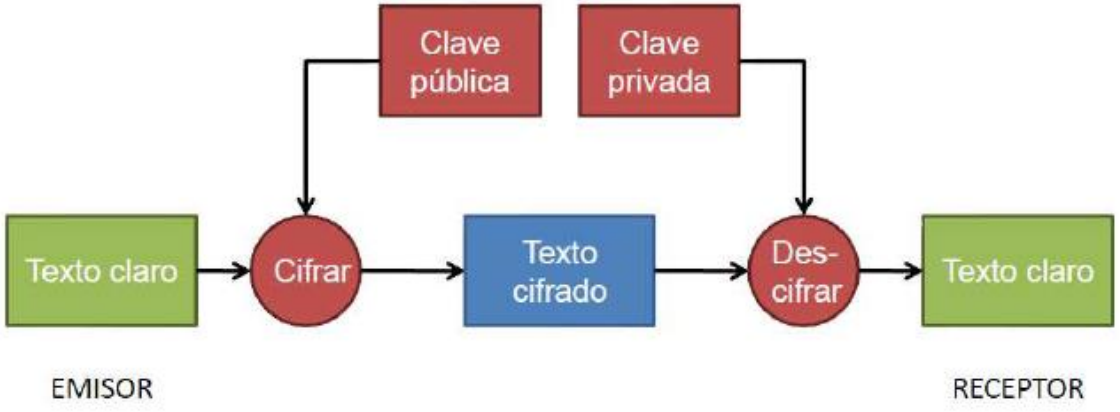
Sistemas de cifrado con clave simétrica:
http://www.criptored.upm.es/intypedia/video.php?id=criptografia-simetrica&lang=es

SSH – Algoritmos de Clave Asimétrica

Algoritmos de clave asimétrica
Se basan en el uso de dos claves: una pública y otra privada.

Cada emisor/receptor tiene dos claves:

- La **clave privada** sólo la conoce el dueño de la clave, es decir, no se publica (no se envía por la red).
- La **clave pública** es conocida por otros.



Se generan al mismo tiempo dando lugar a pares biunívocos, de tal forma que la combinación pública-privada es única.

Lo que se cifra con la clave privada solo se puede descifrar con la pública. Lo que se cifra con la clave pública solo se puede descifrar con la privada.

SSH – Algoritmos de Clave Asimétrica

Los principales problemas de los sistemas de cifrado son:

- Para una misma longitud de clave y mensaje se necesita mayor tiempo de proceso.
- Las claves deben ser de mayor tamaño que las simétricas.
- El mensaje cifrado ocupa más espacio que el original.

Este cifrado ofrece confidencialidad, autentificación y no repudio

Las ventajas que ofrece: clave privada no se transmite y es suficiente que cada usuario tenga su clave doble pública-privada.

Usos principales (aplicaciones):

- Distribución de claves secretas (SSL/TLS, SSH, ...).
- Firma digital.

Sistemas de cifrado con clave asimétrica:
http://www.criptored.upm.es/intypedia/video.php?id=criptografia-asimetrica&lang=es

Servicio SSH

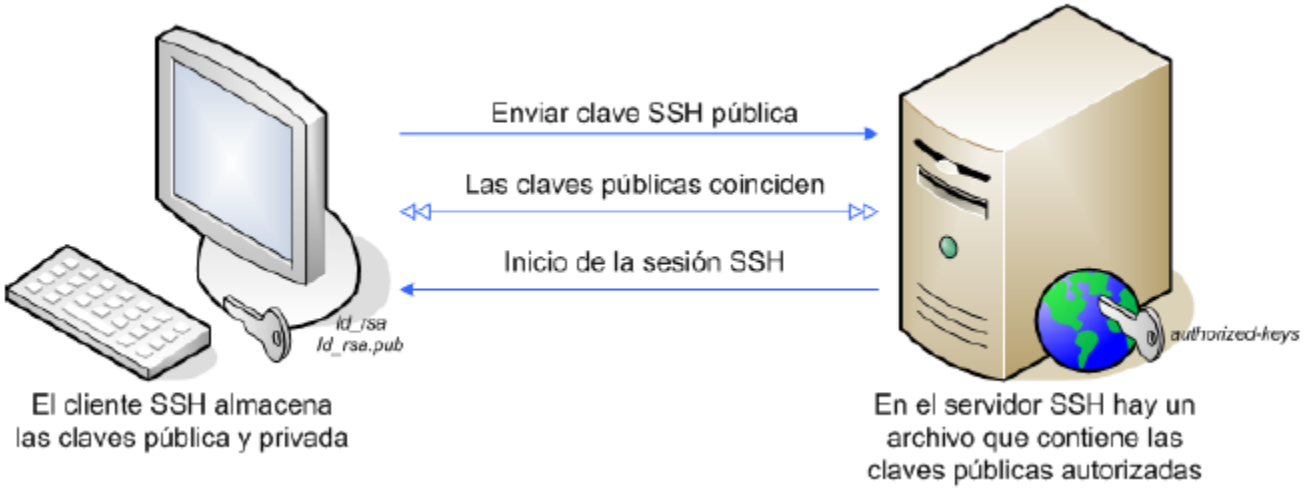
Algortimos encriptación Simétricos (o de clave compartida):

- Clave compartida entre emisor y receptor
- Ventaja: algortimos muy rápidos por tanto muy eficiente
- Desventaja: ambas partes conocen clave
- DES (Data Encryption Standar)
- IDEA (International Data Encryption Algorithm)

Algortimos encriptación Asimétricos (o de clave pública/privada):

- Se basan uso de dos claves: clave pública y clave privada
- Clave pública cifra y privada descifra
- Clave privada sólo conocida por el propietario, no se transmite por la red
- Clave pública conocida por todos, se transmite por la red
- RSA (Rivest-Shamir-Adleman)
- DSA (Digital Signature Algorithm)

Servicio SSH

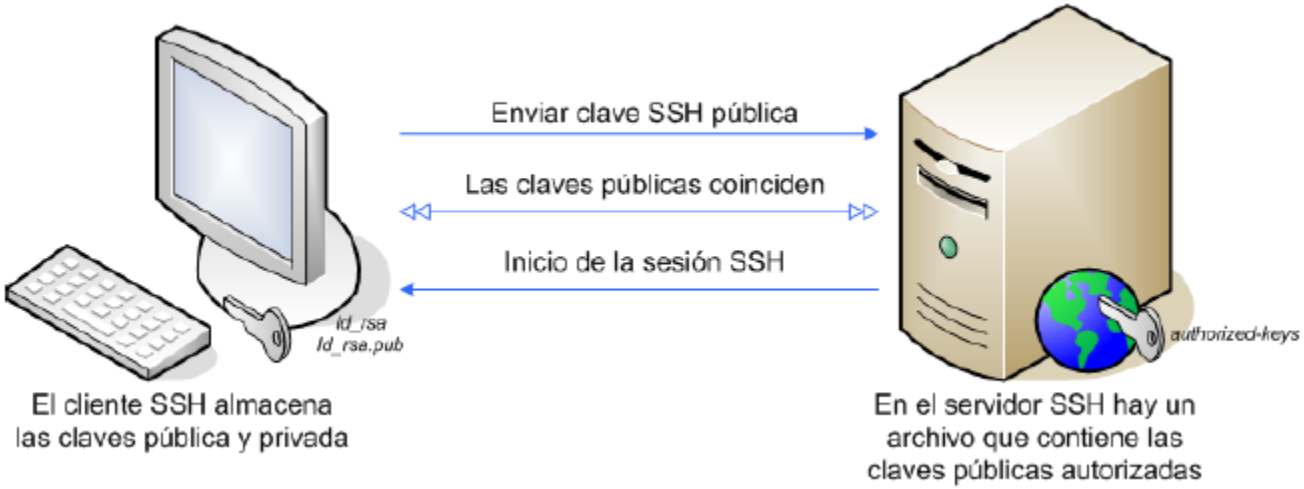


1. El cliente inicia una conexión TCP con puerto 22 del servicio.

2. El cliente y el servidor se ponen de acuerdo en la versión del protocolo a utilizar, así como el algoritmo de cifrado utilizado para el intercambio de la información.

3. El servidor, que tiene en su poder dos claves (una privada y una pública), manda su clave pública al cliente.

Servicio SSH

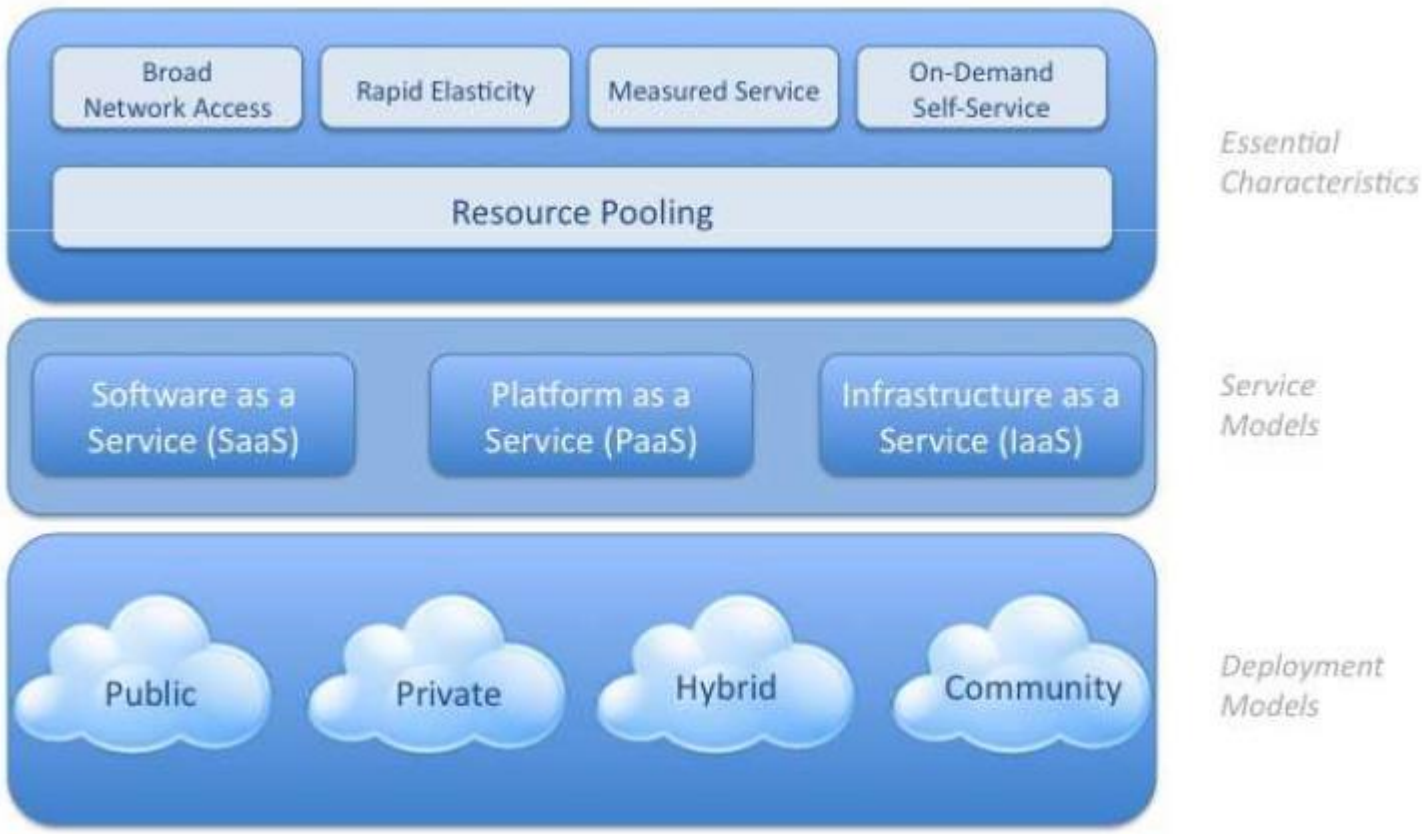


4. Cuando el cliente recibe la clave enviada por el servidor, la compara con la que tiene almacenada (si no la tuviera el protocolo SSH exige que el cliente la confirme la primera vez) para verificar su autenticidad**.**

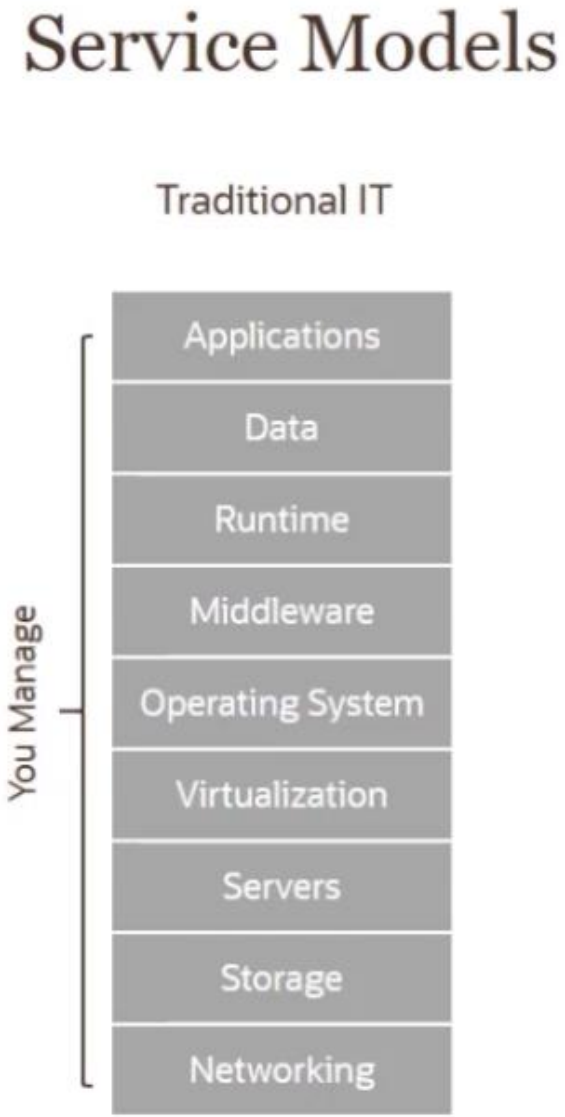
5. Con la clave pública del servidor en su poder, **el cliente genera una clave de sesión aleatoria,** creando un mensaje que contiene esa clave y el algoritmo seleccionado para la encriptación de la información. Toda esa información es enviada al servidor haciendo uso de la clave pública que envió en un paso anterior de forma cifrada.

6. Si todo es correcto, el cliente queda autenticado, iniciando la sesión para comunicarse con el servidor.

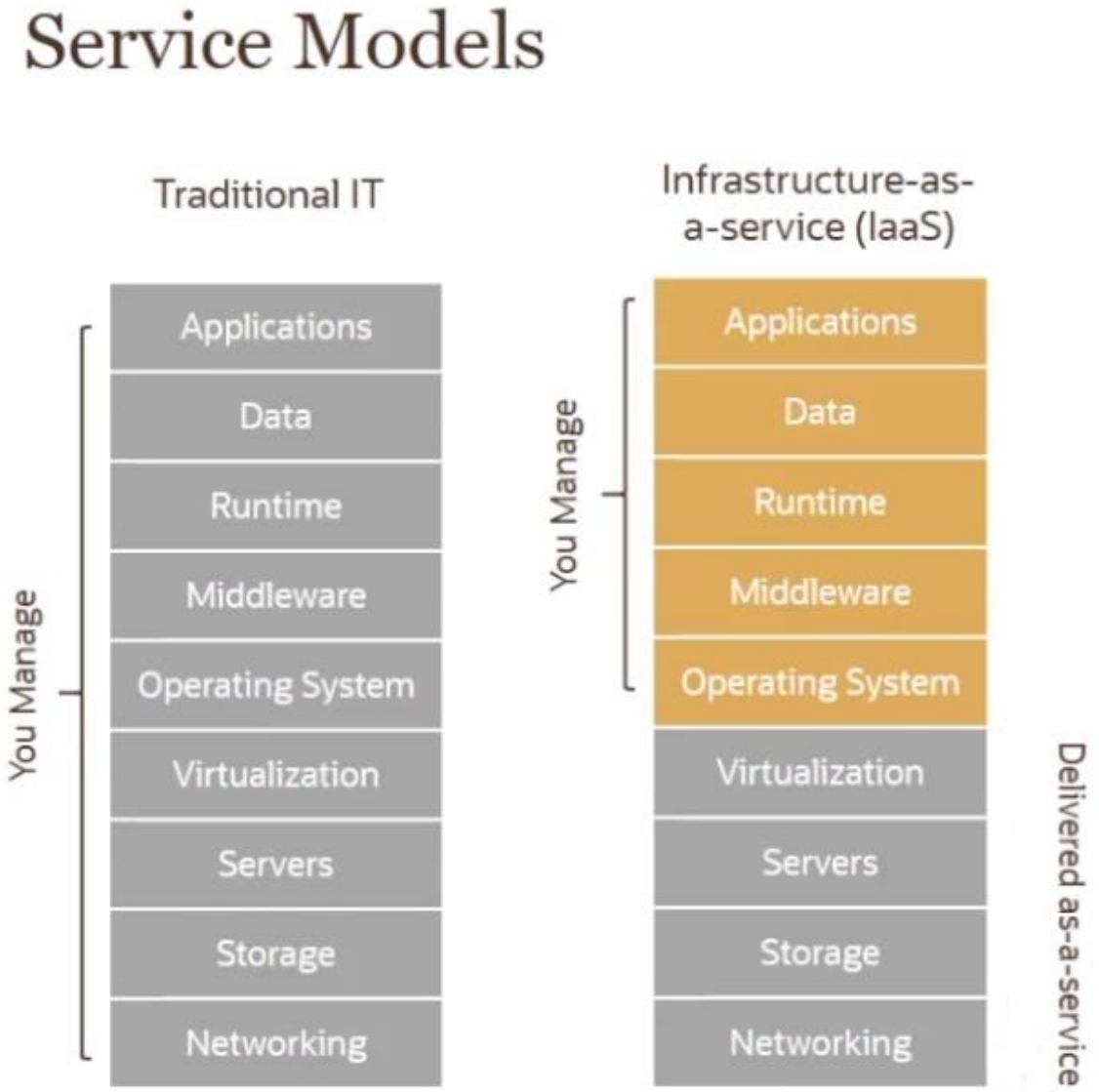
Cloud Computing



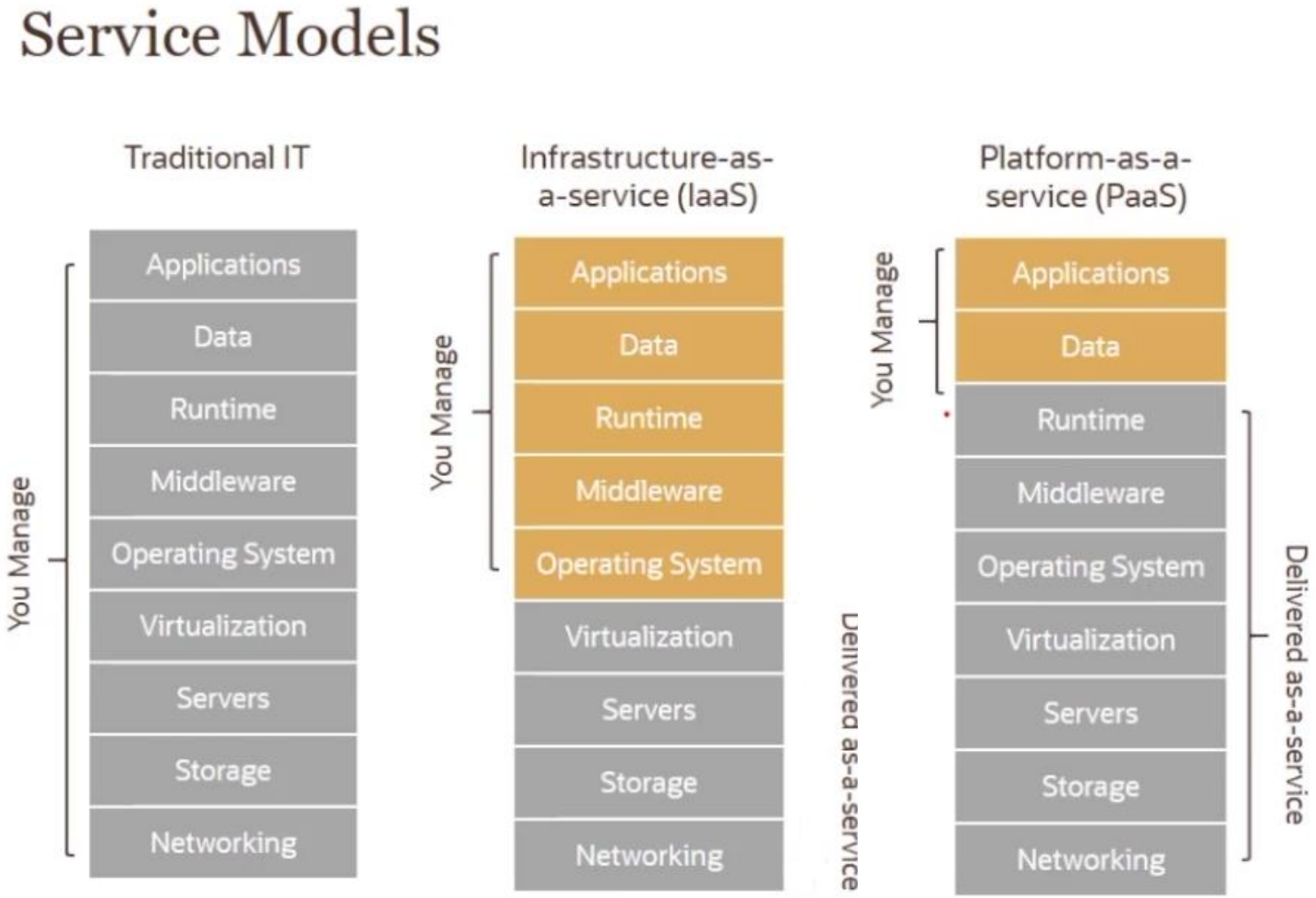
Cloud Computing



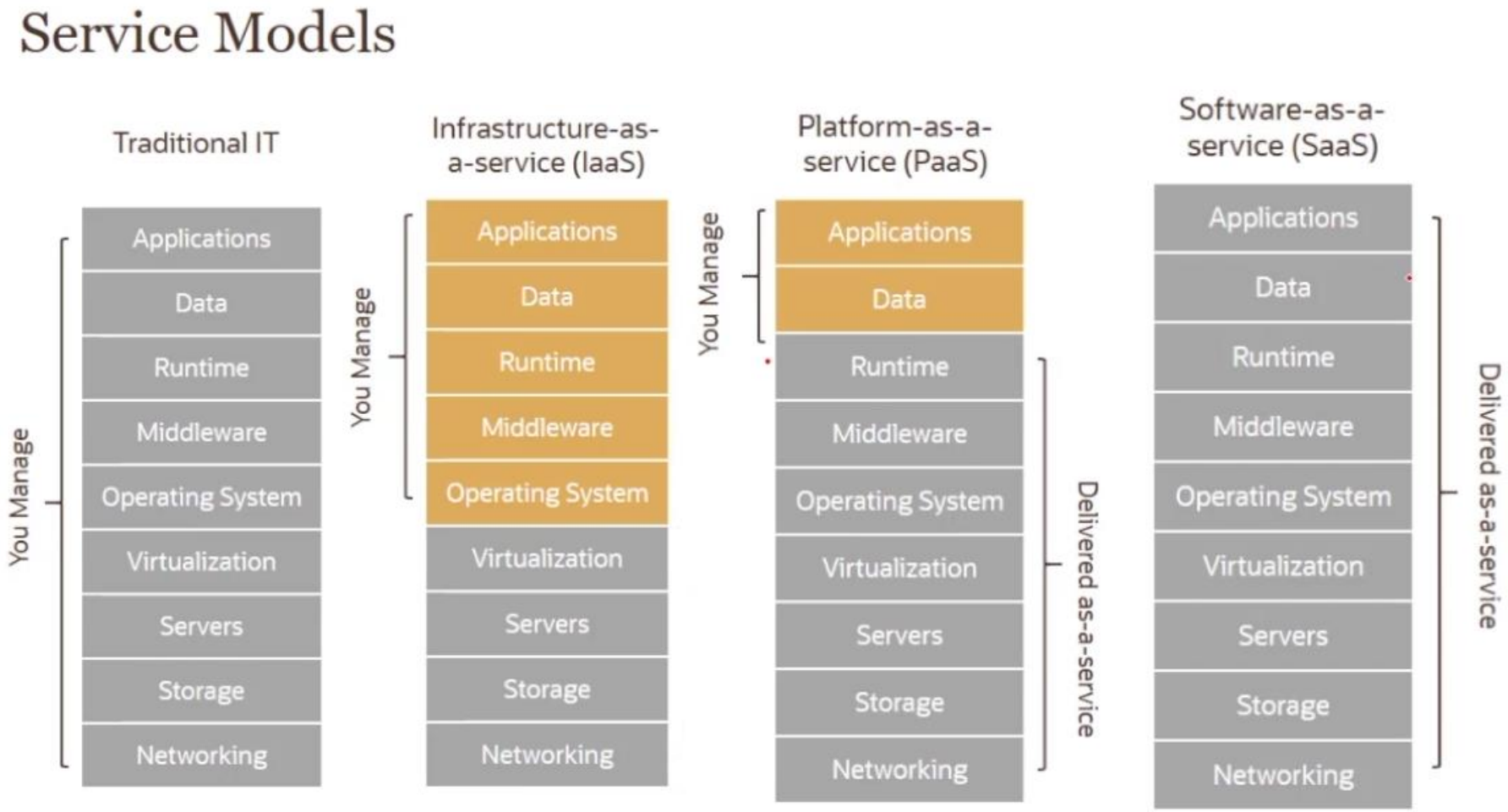
Cloud Computing - IaaS – Infraestructure As Service



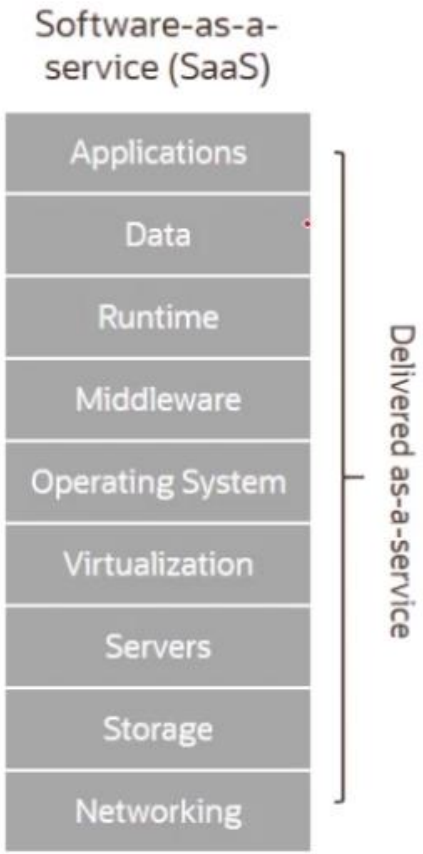
Cloud Computing - PaaS – Platform As Service



Cloud Computing - SaaS – Software As Service



Modelo de Servicio SaaS – Software As Service



Aplicaciones ejecutando sobre una infraestructura *cloud* son ofrecidas como servicio.

Orientado a **usuarios.**

Los usuarios pagan por el uso, no por poseer el software (ni siquiera licencias)

Ejemplos

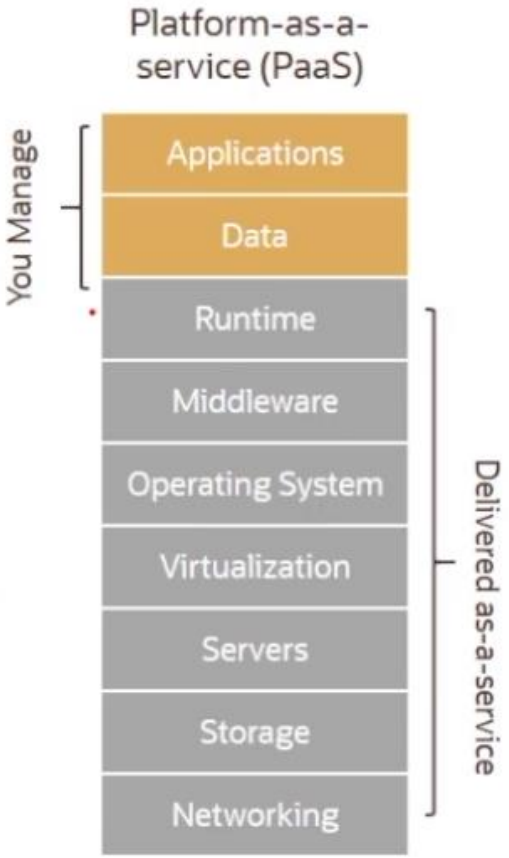
- *Dropbox*
- *Google Drive*
- *Evernote*
- *Google Apps*
- *Office Web Apps*
- ...

Dropbox

Google Apps

Office Web Apps

Cloud Computing - PaaS – Platform As Service



Plataforma de software y entornos de desarrollo y pruebas ofrecidos como servicio.

- Empaqueta el entorno de desarrollo y ofrece un API
- Abstrae hardware y servicios.

Permite el despliegue y ejecución de aplicaciones.

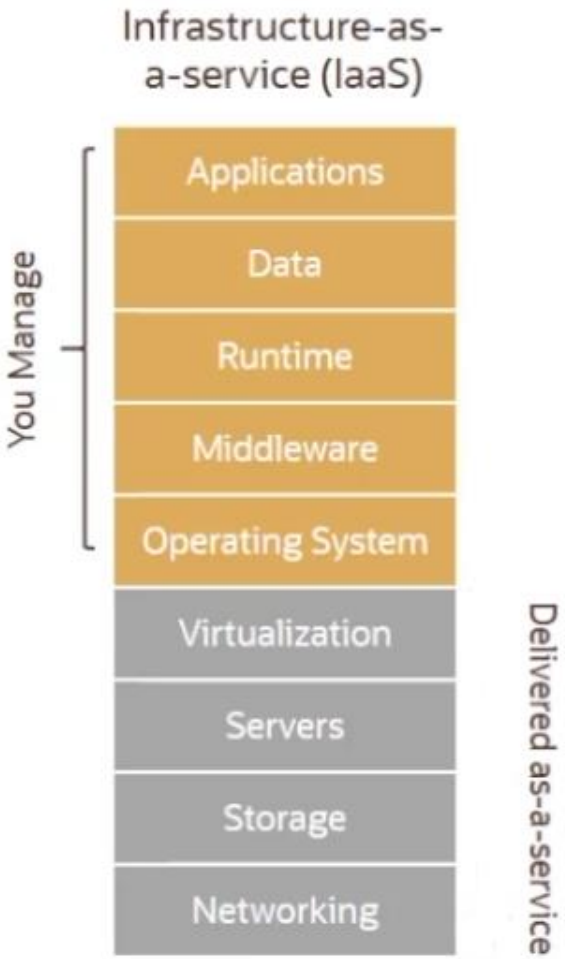
Servicio para todas las fases de desarrollo y pruebas de software.

Orientado a **desarrolladores**.

Proveedores de PaaS

- *OpenShitt*
- *Heroku*
- *Google App Engine*
- *Microsoft Azure*
- *Amazon Web Services*
- *Force.com*
- *Jelastic*
- *Pivoltal*
- ...

Cloud Computing - IaaS – Infraestructure As Service



Despliegue de Infraestructura completo MVs. Orientado a arquitectos software

Proveedores de Iaas

- *Amazon Web Services*
- *RackSpace*
- *Google Compute Engine*
- *Azure Service Plataform*
- *HP Cloud*
- *GoGrid*
- *...*

Bibliografía

- ✓ Servicios de Red e Internet. Álvaro García Sánchez, Luis Enamorado Sarmiento, Javier Sanz Rodríguez. Editorial Garceta.