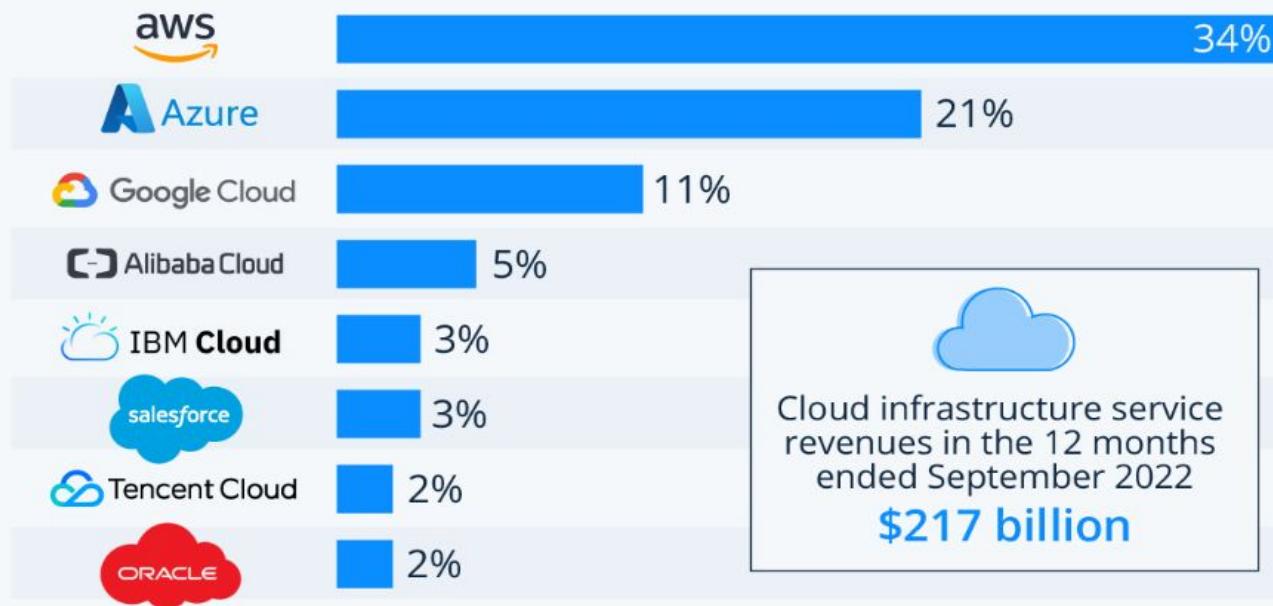


- **Cloud Computing**
 - Indicadores / Motivaciones
 - **Cloud Classic:**
 - Conceptos.
 - Modelos de Servicio y Despliegue.
 - Seguridad. Herramientas: **Openstack**.
 - Desarrollo de caso práctico
 - **Cloud Native:**
 - Conceptos.
 - Metodologías de Trabajo.
 - Seguridad. Herramientas: **Kubernetes**.

Amazon, Microsoft & Google Dominate Cloud Market

Worldwide market share of leading cloud infrastructure service providers in Q3 2022*



Cloud infrastructure service revenues in the 12 months ended September 2022
\$217 billion

* includes platform as a service (PaaS) and infrastructure as a service (IaaS) as well as hosted private cloud services

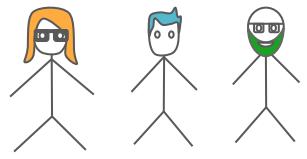
Source: Synergy Research Group

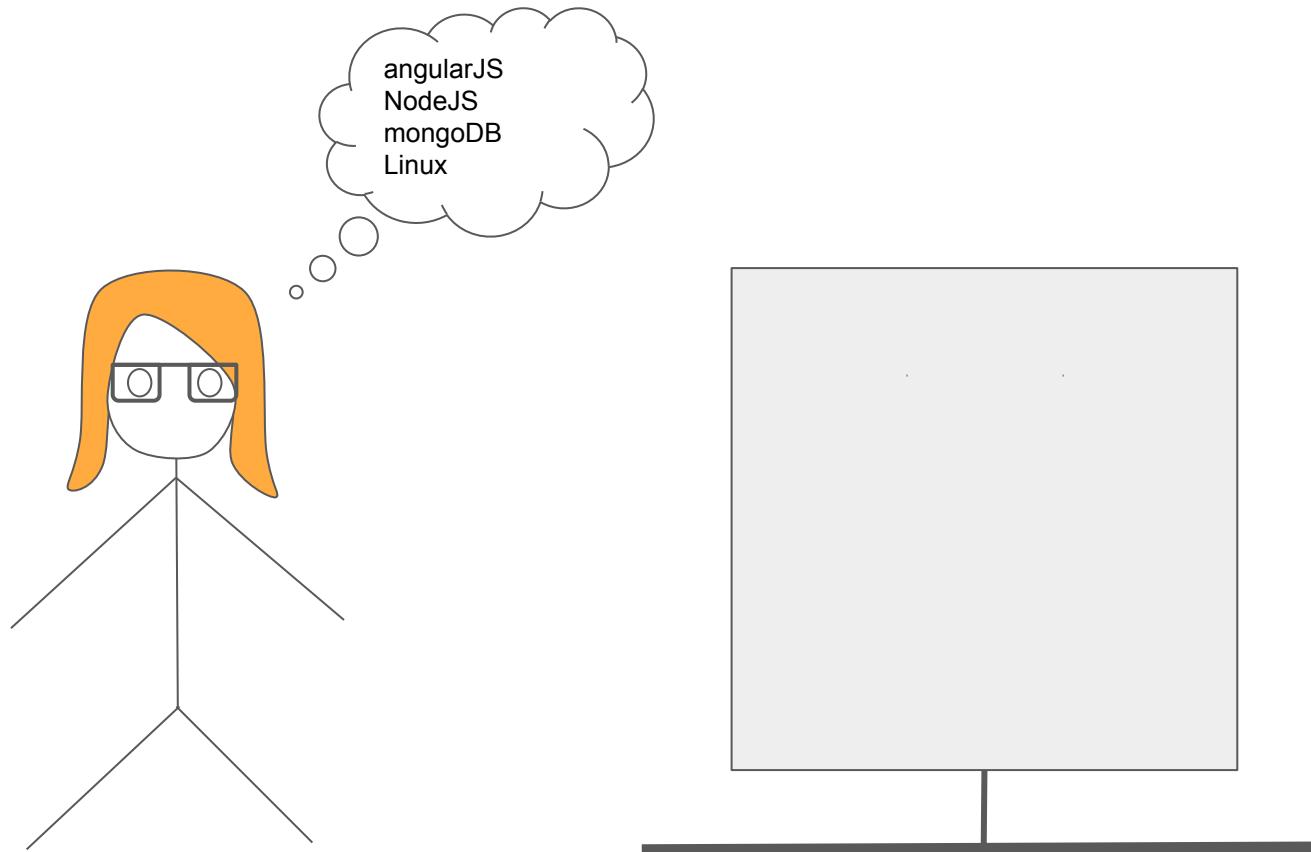
Motivos para adopción de Cloud

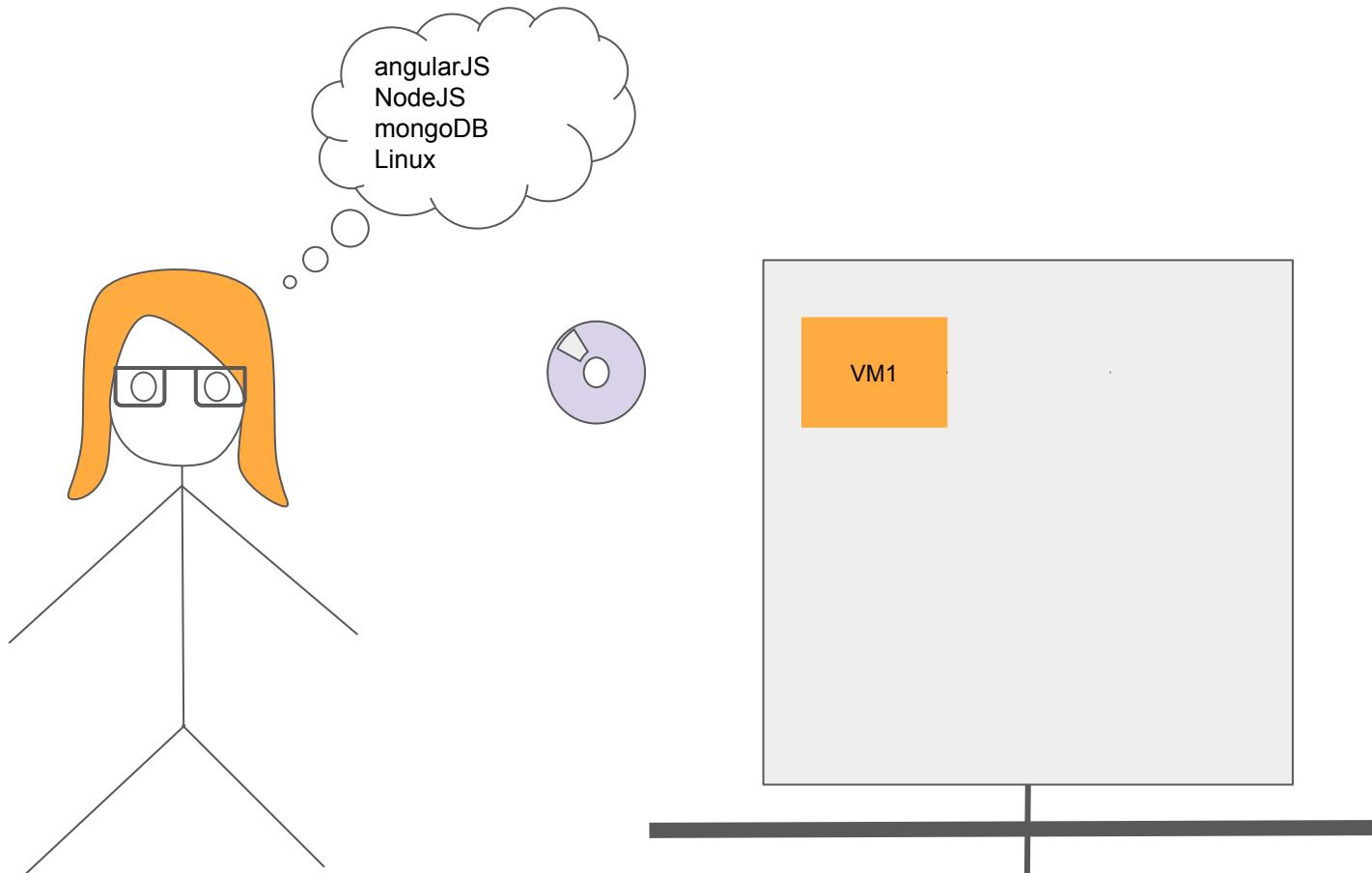


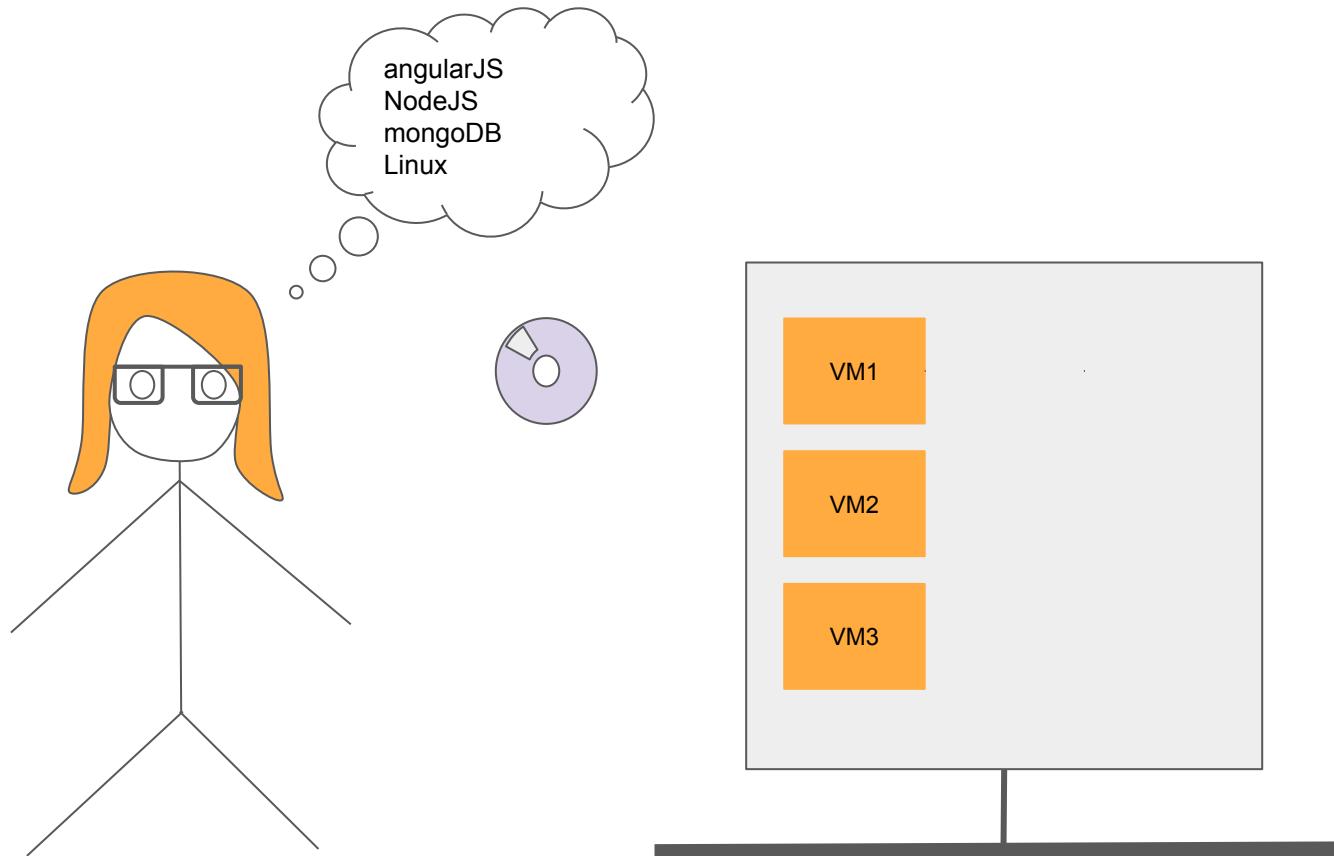
Cloud computing ... La nube

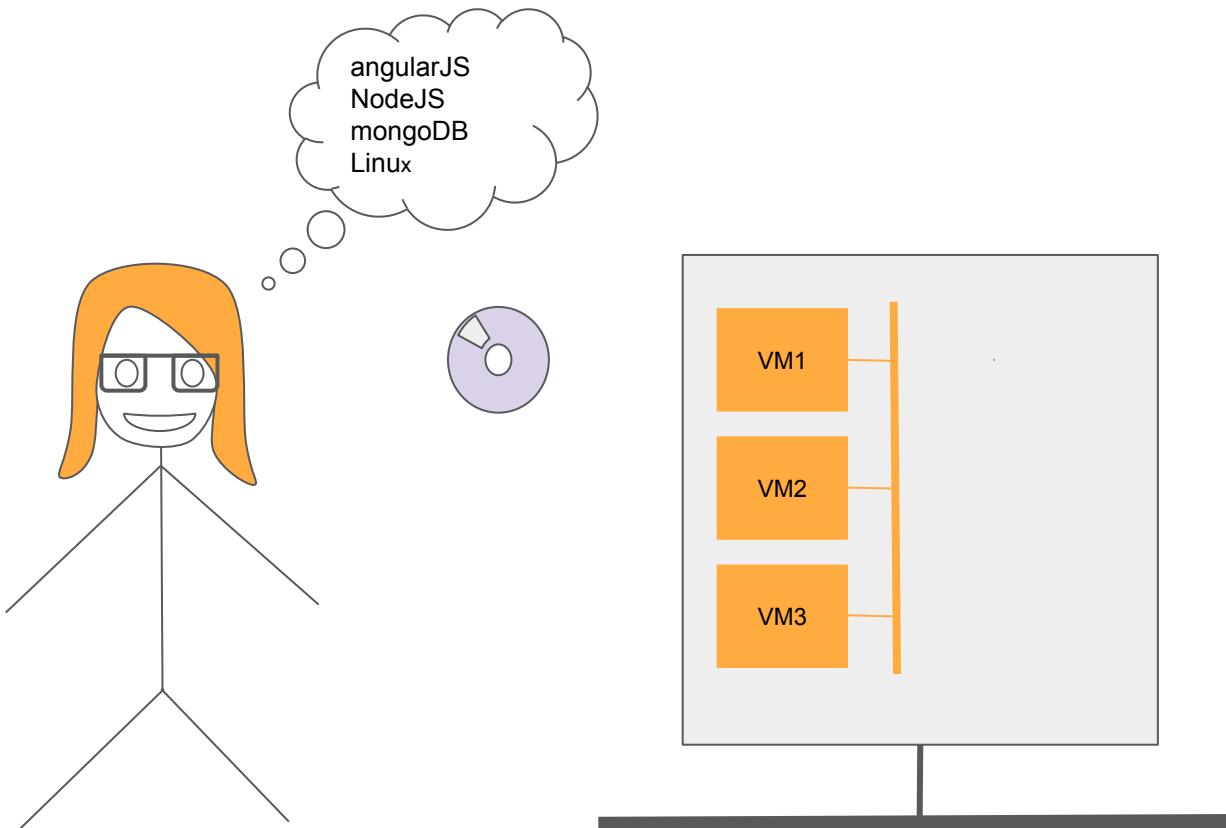
¿ lo qué ?

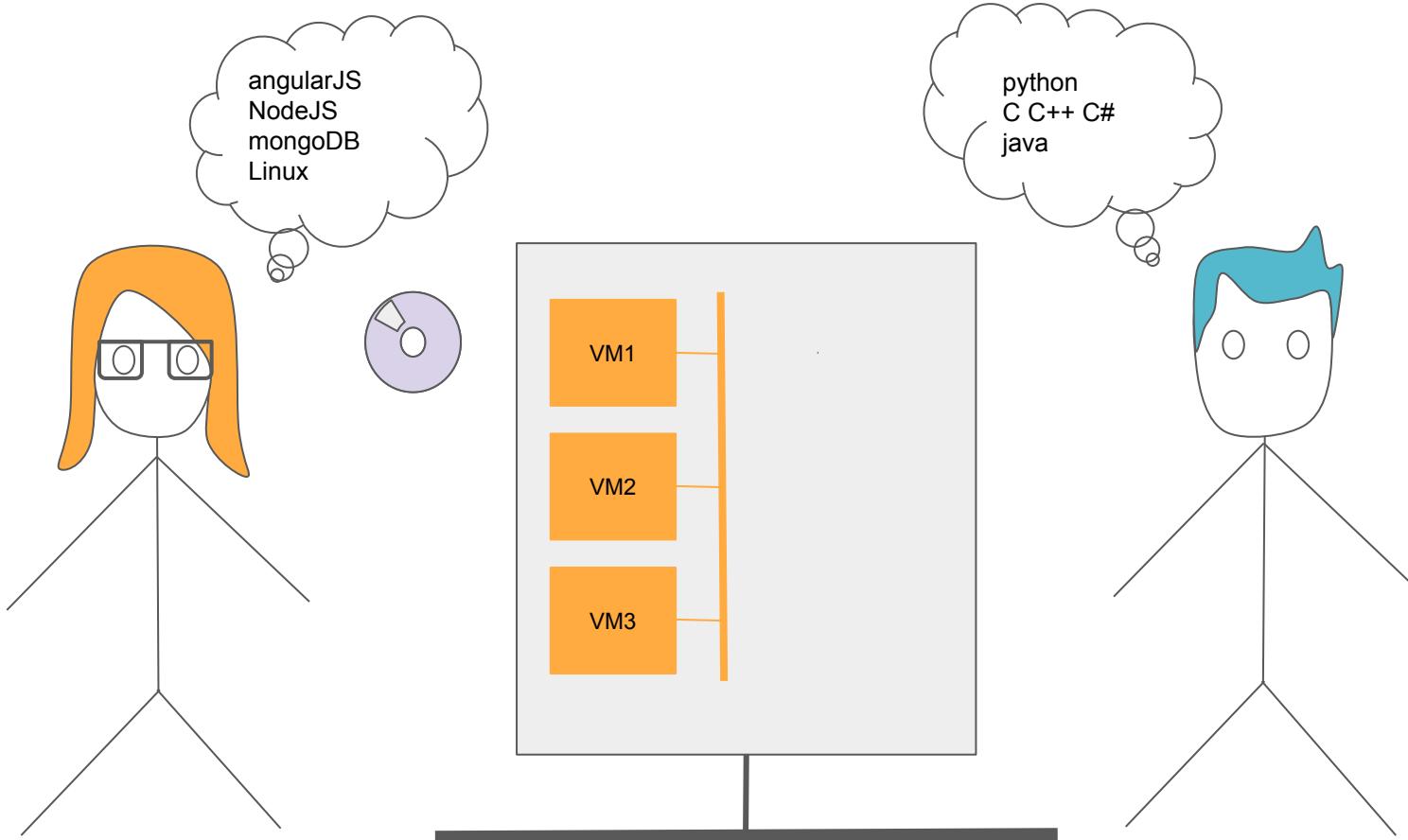


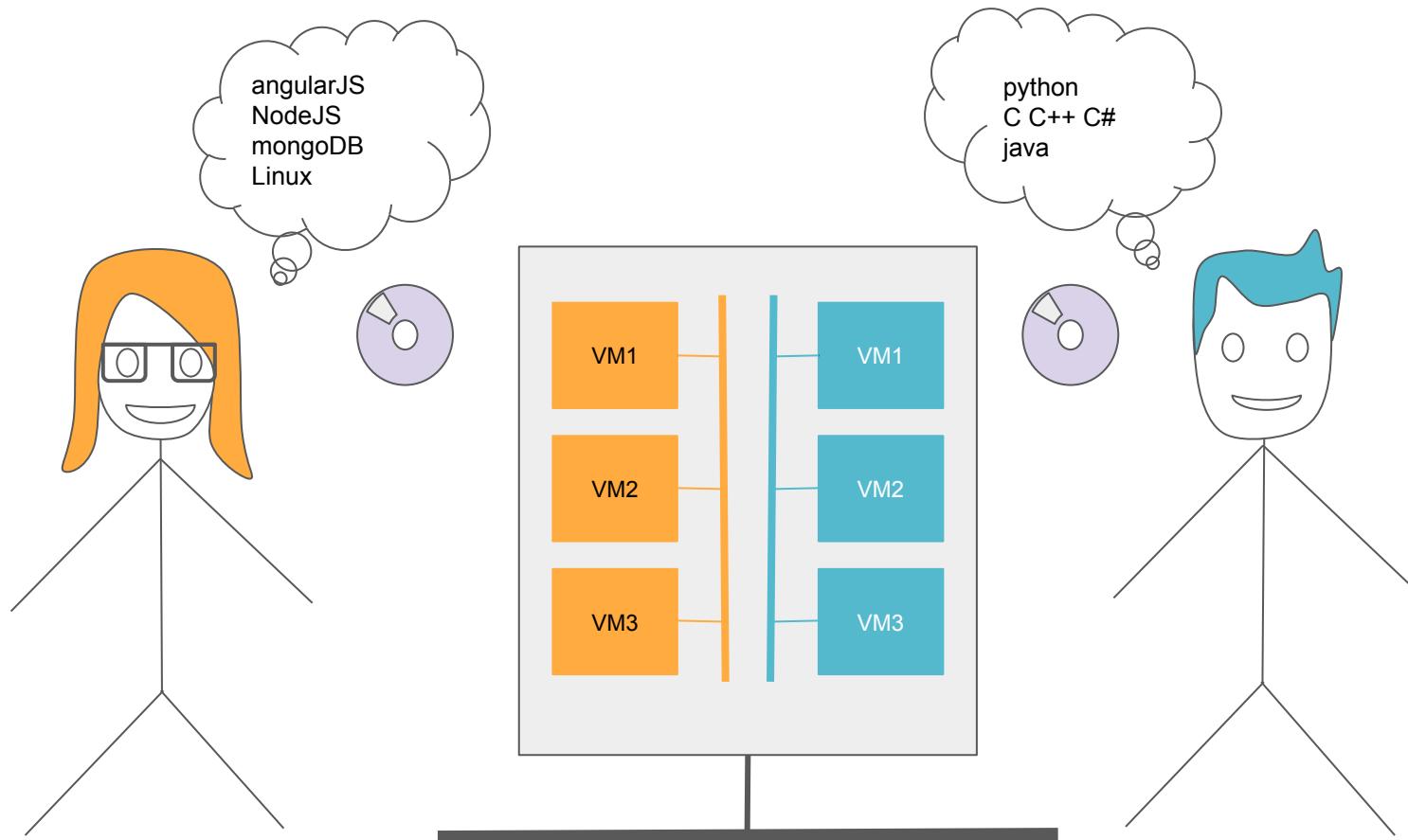


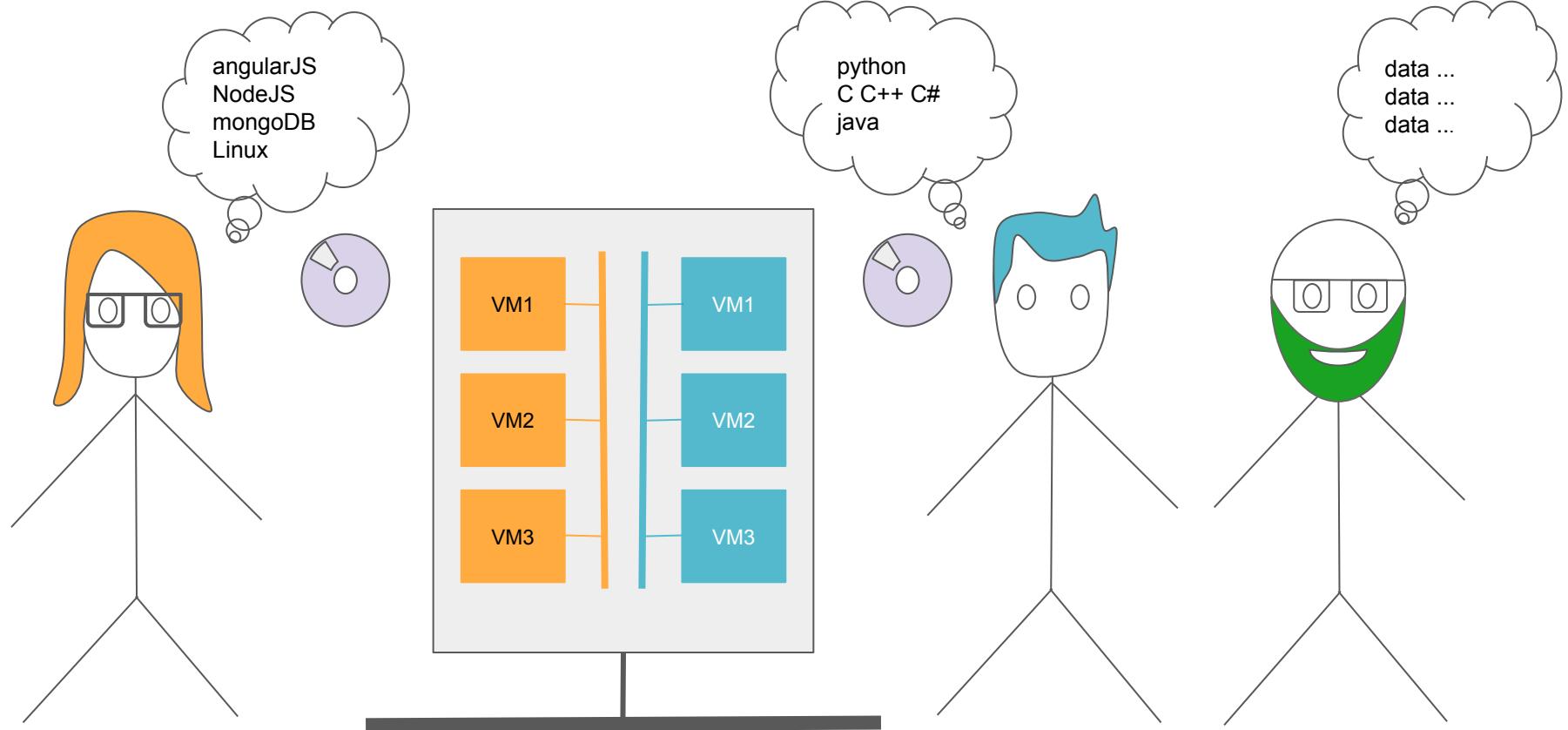


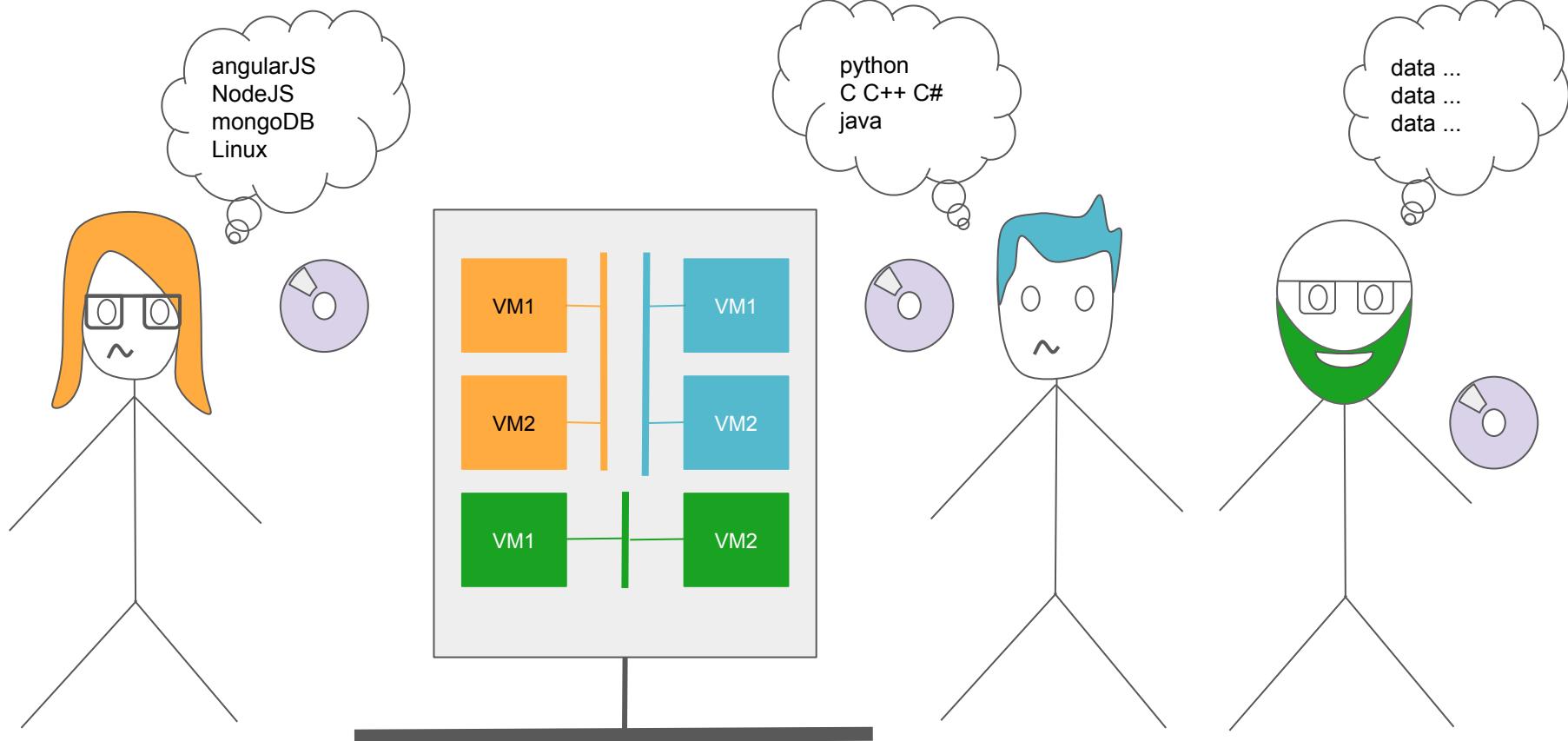


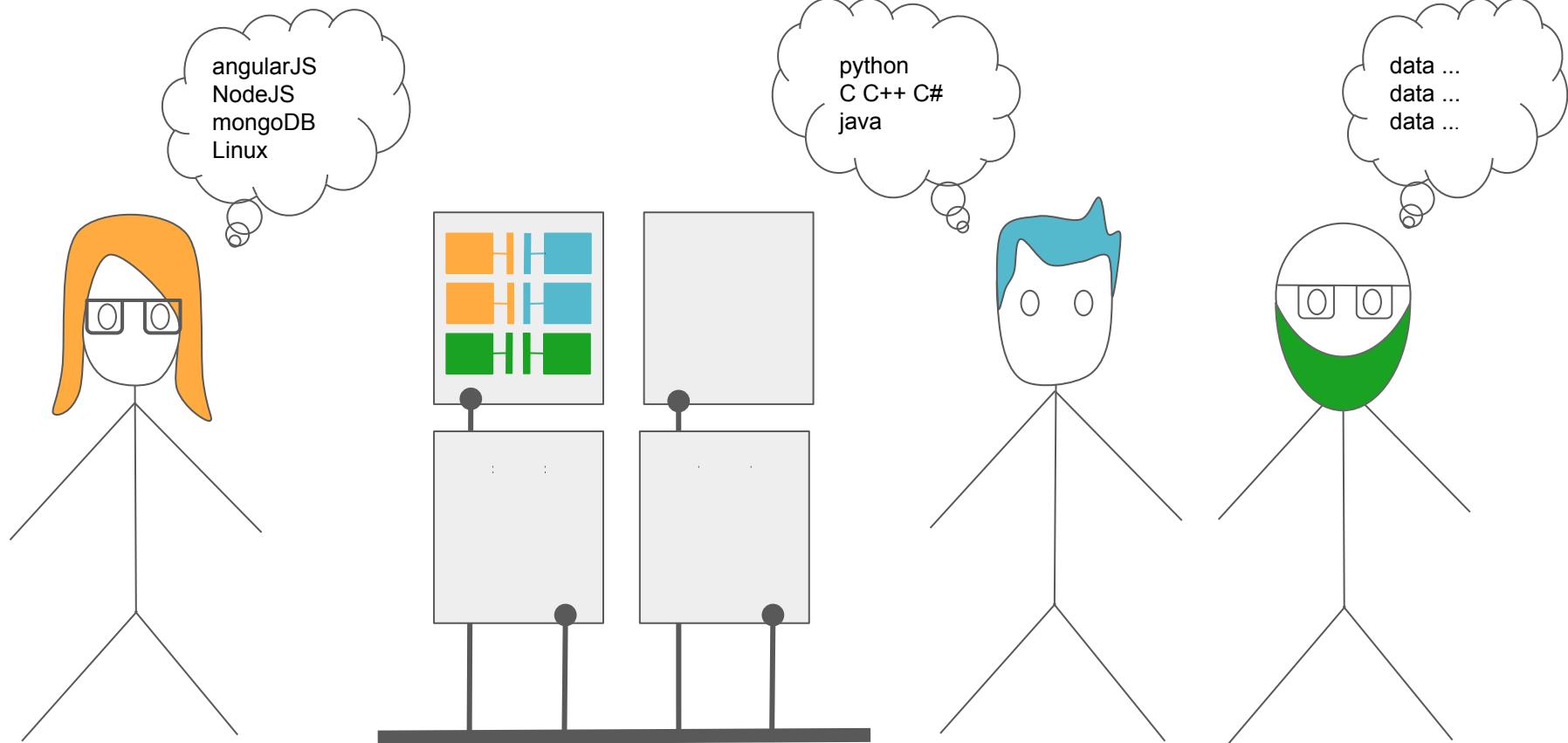


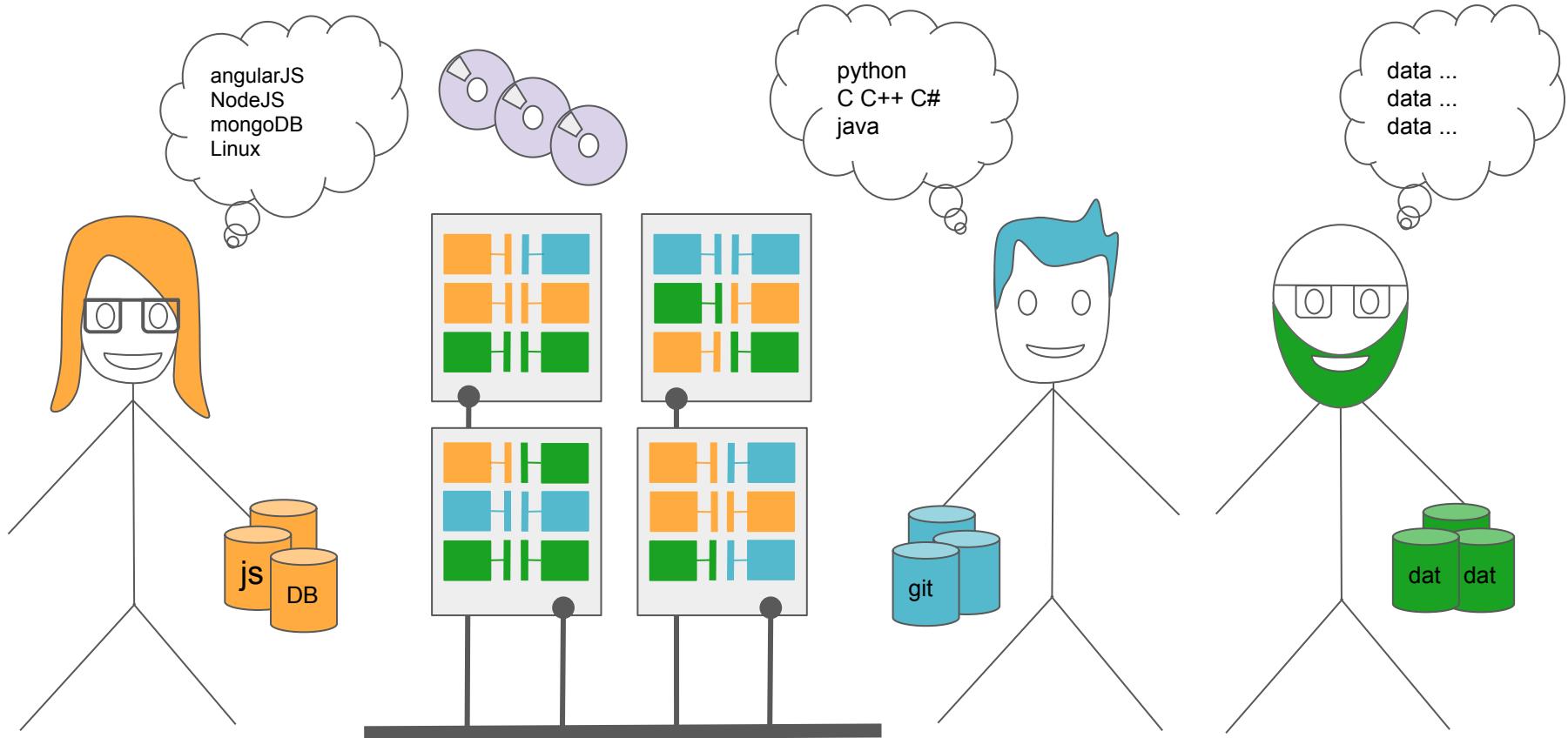






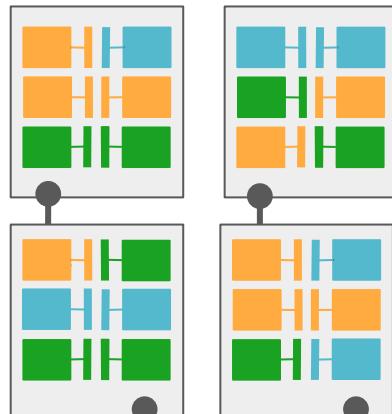




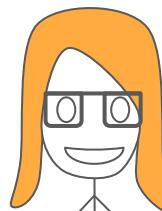


keystone - identity

nova - compute



Ayelén



Bruno



Carlos



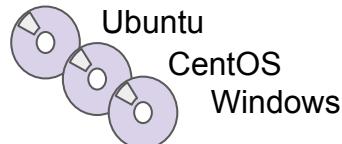
neutron - network

10.0.1.0/24

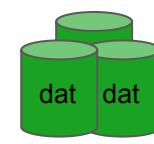
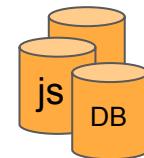
10.0.2.0/24

10.0.3.0/24

glance - image



cinder - volume



Cloud Computing: Definición

“**Uso de recursos de computación entregados sobre una red como un servicio**” (1)

“Un modelo para facilitar el acceso a **través de la red** de manera conveniente y ubicua a un conjunto compartido de **recursos de computación** (redes, servidores, almacenamiento, aplicaciones y servicios) que pueden ser provisionados fácilmente y rápidamente con un mínimo esfuerzo de gestión o interacción por parte del proveedor de servicios” (2)

Este modelo de cloud se compone de cinco características esenciales, tres modelos de servicio y cuatro de despliegue.

Características

- Auto servicio y bajo demanda
- Acceso por red
- Compartición de recursos [multi-inquilino]
- Rápida elasticidad [escalabilidad]
- Servicio Medido [Monitoreo]

Modelos de Servicio

- **SaaS** - Software aaS
- **PaaS** - Platform aaS
- **IaaS** - Infra aaS

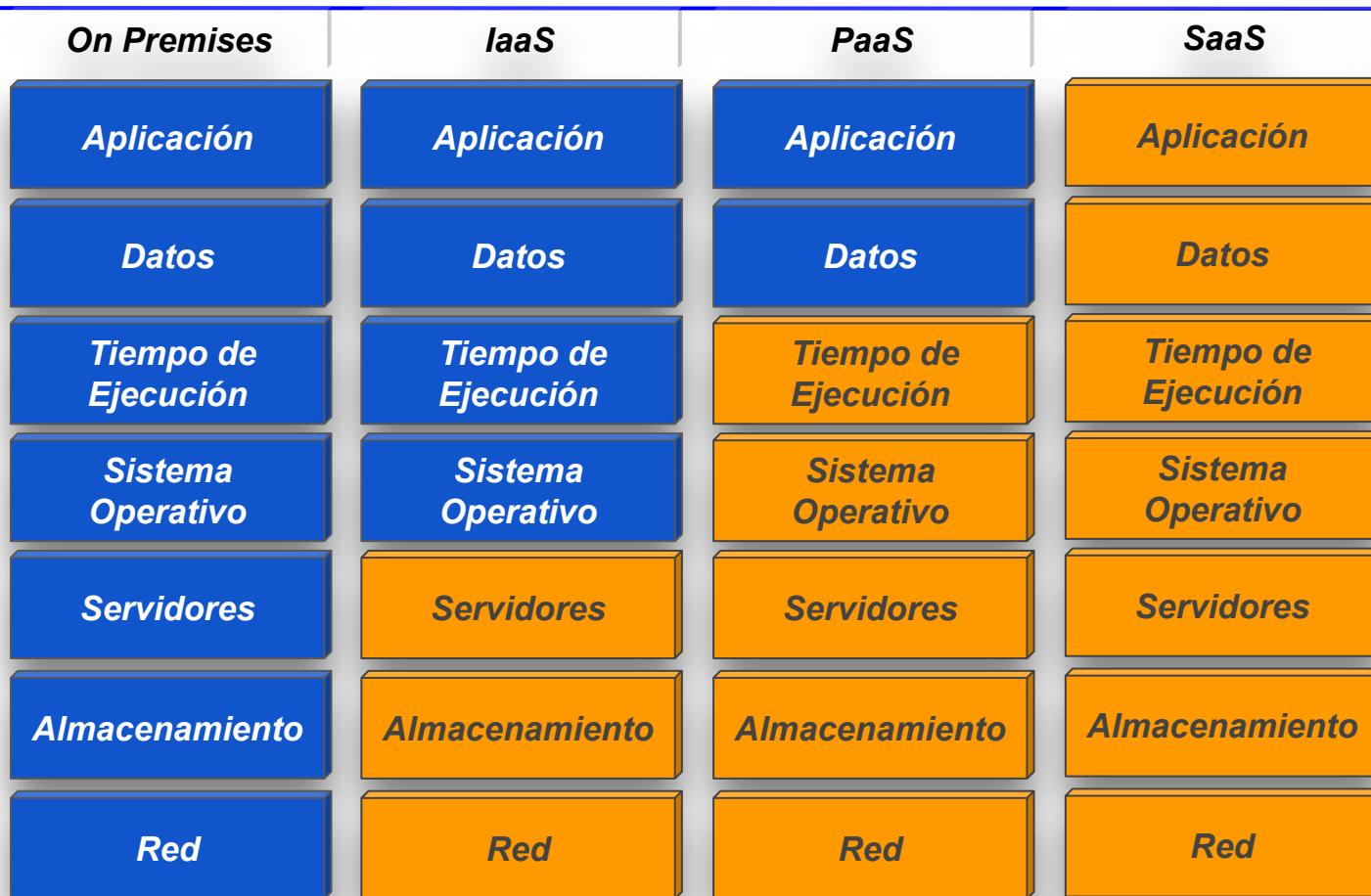
Modelos de Despliegue

- Nube privada
- Nube comunitaria
- Nube pública
- Nube híbrida

1 http://en.wikipedia.org/wiki/Cloud_computing

2 [The NIST Definition of Cloud Computing](http://www.nist.gov/itl/cloud/computing/nistclouddefinition.htm)

Cloud Computing = Cloud Classic “para nos”



Cloud Computing

- autoservicio
- bajo-demanda
- acceso por red
- recursos en común (multi-inquilino)
- rápida elasticidad (escalabilidad)
- servicio medido (monitoreo)



Administración del Cliente



Administración del Proveedor

Cloud Computing

**Paradigma que permite ofrecer
recursos computacionales
mediante Internet**

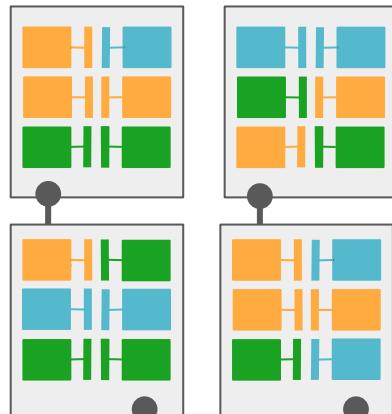


openstack.®

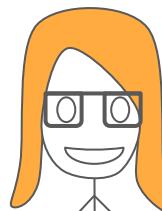
**Recursos virtuales agrupados
para diseñar y gestionar nubes
Publicas o privadas**

keystone - identity

nova - compute



Ayelén



Bruno



Carlos



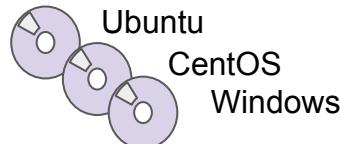
neutron - network

10.0.1.0/24

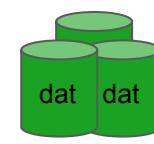
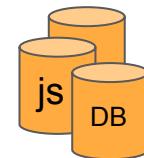
10.0.2.0/24

10.0.3.0/24

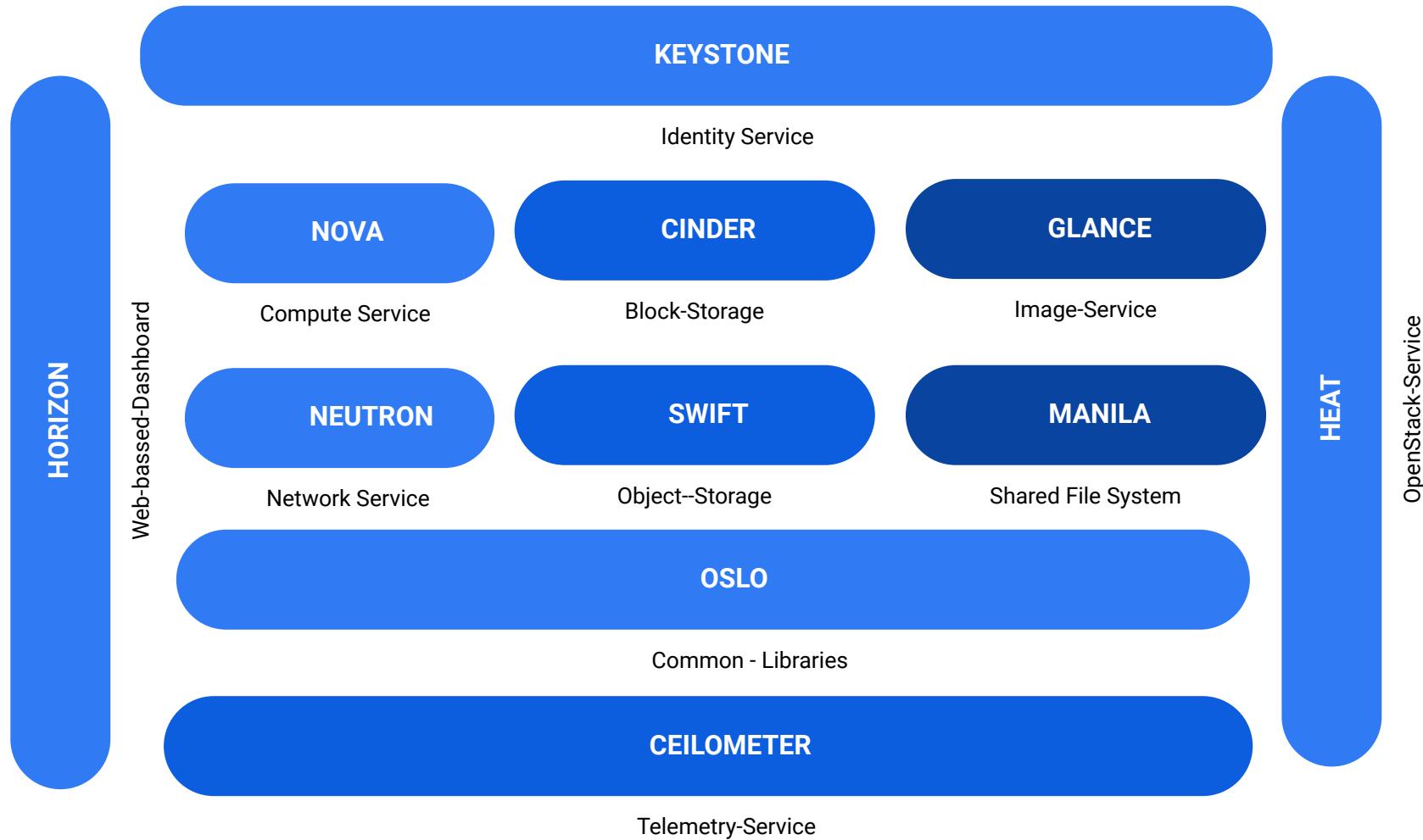
glance - image



cinder - volume



Cloud Computing



UMCloud: Onboarding



1. <https://cloud.um.edu.ar>

The screenshot shows the UM Cloud homepage with a large logo at the top right. Below the logo, there are several navigation links:

- Entrenamiento
 - Programa: Cloud Fundamentals
 - Programa: Cloud DevOps
- Presentaciones
 - UM a la Altura de la NUBE: Diapos
 - UM a la Altura de la NUBE: Video Parte 1
 - UM a la Altura de la NUBE: Video Parte 2
- Herramientas
 - My-UM-Cloud
- Repositorios

2. Oauth [“apellido um.edu.ar”]
- @um.edu.ar
 - @alumno.um.edu.ar

The screenshot shows the Google OAuth login page. It features the Google logo and the option to "Acceder con Google". Below this, there is a large button labeled "Acceder" and a link to "Ir a um.edu.ar". A text input field is provided for "Correo electrónico o teléfono". Below the input field, a link says "¿Olvidaste el correo electrónico?". At the bottom, there is explanatory text about Google sharing information with um.edu.ar, and two buttons: "Crear cuenta" and "Siguiente".

UMCloud: Onboarding



3. <https://my.cloud.um.edu.ar>

My-UM-Cloud [diego.navarro@um.edu.ar] ▾ Help

Bienvenido a My-UM-Cloud

A) Para poder hacer uso de OpenStack necesitas tus credenciales de acceso.
1) Haz click en Cloud_Credentials [Boton NARANJA]
2) Si es tu primera vez se crearan tus credenciales con un mensaje Credential creation in progress... vuelve a clickear en Cloud_Credentials
3) Apareceran tus credenciales: username xxxx / password yyyy y un enlace para acceder al Dashboard.

B) Para poder acceder a los recursos dentro de la cloud necesitas vincular tu equipo a la VPN Zerotier de la UM-Cloud.
1) Si es la primera vez ingresa a <https://zerotier.com/download/> y sigue los pasos para instalar en tu sistema operativo.
1.a) Una vez instalado y funcionando debes hacer un join a la red de la Cloud:

sudo zerotier-cli join a84ac5c10a1a8ff2
200 join OK

1.b) Posteriormente debes obtener tu address Zerotier

sudo zerotier-cli info
200 info 2725d6592c 1.2.12 ONLINE

1.c) Haz click en Zerotier_Config [Boton AZUL]

Completa en la caja de texto tu address Zerotier del paso 1.b y haz click en Create_ZT

1.d) Continua al proximo paso

2) Haz click en Zerotier_Config [Boton AZUL]

Si estas conectado deberias ver tu usuario/direccion zerotier/ip en la red y un boton VERDE

UMCloud: Onboarding



4. Horizon Dashboard <https://console.cloud.um.edu.ar>

openstack. Default • diego.navarro_um.edu.ar_project ▾ diego.navarro_um.edu.ar

Compute ▾ Vista general

Vista general

Instancias

Imágenes

Pares de claves

Grupo de servidores

Volúmenes >

Red >

Almacén de objetos >

Identity >

Vista general

Limit Summary

Compute

| Categoría | Usada | Total |
|-------------------------------|--------|--------|
| Instancias | 3 | 10 |
| VCPU | 3 | 20 |
| RAM | 6GB | 50GB |
| Volúmenes | 0 | 10 |
| Instantáneas de volumen | 0 | 10 |
| Almacenamiento de volumen | 0Bytes | 1000GB |
| Red | 0 | 100 |
| IPs flotantes | 0 | 50 |
| Grupos de seguridad | 1 | 10 |
| Reglas del grupo de seguridad | 6 | 100 |
| Redes | 0 | 500 |
| Puertos | 3 | 500 |
| Routers | 0 | 10 |

Usage Summary

Select a period of time to query its usage:
The date should be in YYYY-MM-DD format.

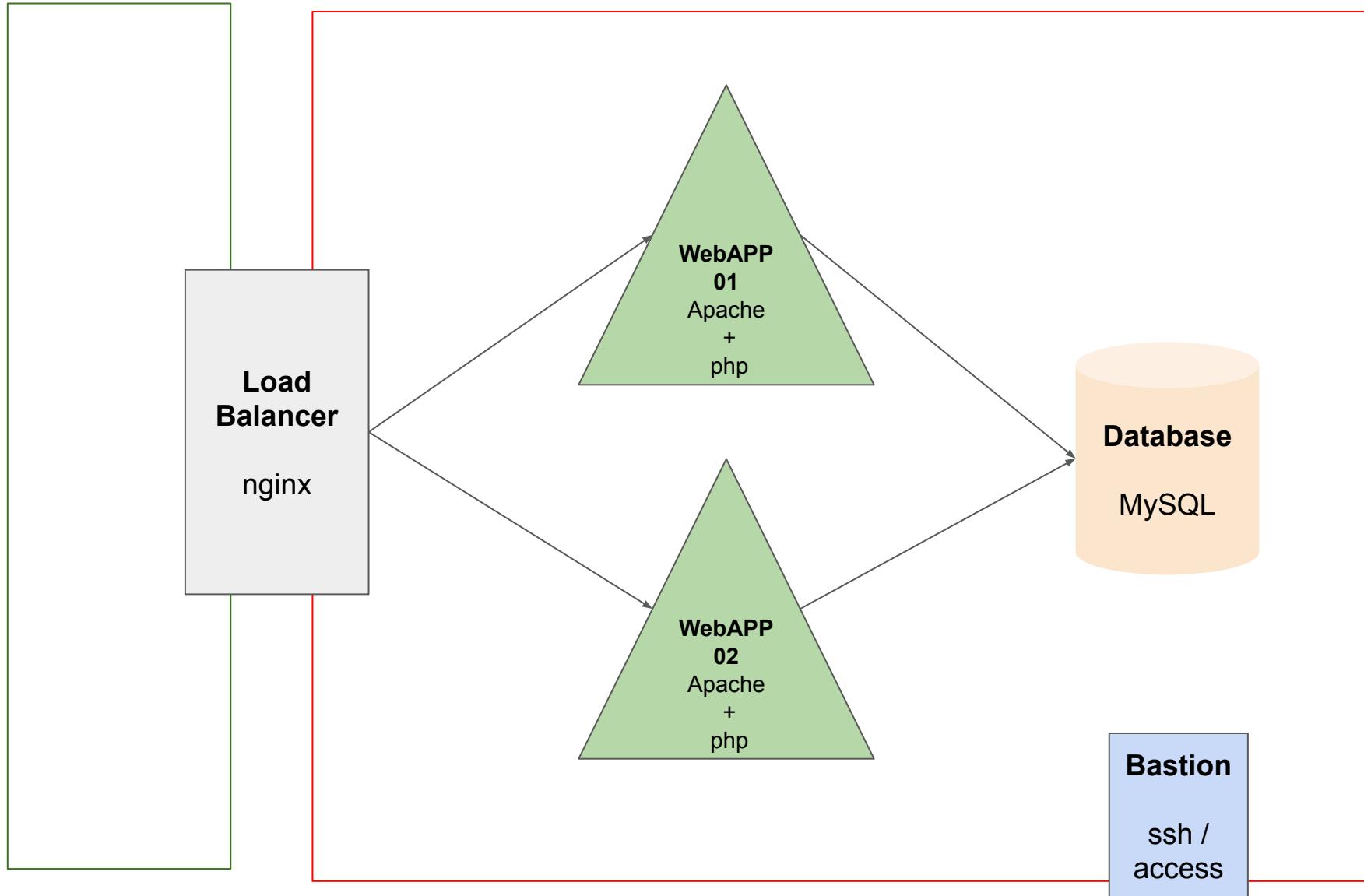
2019-12-03 to 2019-12-04 Enviar

Instancias Activas: 3
Active RAM: 6GB
This Period's VCPU-Hours: 0,72
This Period's GB-Hours: 14,45
This Period's RAM-Hours: 1480,14

Uso

Cloud Classic Caso Práctico

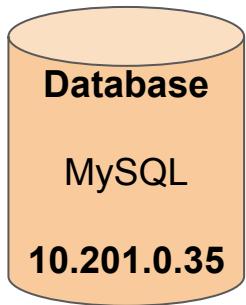
Aplicación Web con múltiples capas: Wordpress



Aplicación Web con múltiples capas: Wordpress

Necesitamos instalar un motor de base de datos, exponerlo a que sea

accesible por la red, crear la base de datos y un usuario con permisos para esa base de datos.



- 1) Creamos instancia `ubuntu_minimal_1804`, llamarla: `USUARIO-wp-db`
[zona de disponibilidad : workers-az]
- 2) Nos logueamos: `ssh ubuntu@<IP_instancia>`

```
sudo mount -o remount,nobarrier,commit=120 /
sudo apt update && sudo apt-get install -y mysql-server vim-tiny
```

```
#Poner Base de datos disponible en la RED
sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf
    #bind-address          = 127.0.0.1
    bind-address            = 0.0.0.0
sudo systemctl restart mysql
#Creamos DB y usuario para WP
sudo mysql
```

```
CREATE DATABASE wordpress;
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER
    ON wordpress.* 
    TO wordpress@'%'
    IDENTIFIED BY 'telewordpress';
```

```
sudo journalctl -u mysql # ver logs del servicio mysql
```

Aplicación Web con múltiples capas: Wordpress



Necesitamos desplegar Wordpress y configurarlo para que acceda a la db remota.

- 1) Creamos instancia ubuntu_minimal_1804, llamarla: USUARIO-wp-webapp-01
- 2) Nos logueamos: `ssh ubuntu@<IP_instancia>`

```
sudo mount -o remount,nobarrier,commit=120 /
sudo apt-get update && sudo apt-get install -y wordpress php libapache2-mod-php php-mysql vim-tiny
```

```
sudo vi /etc/apache2/sites-available/wordpress.conf
```

```
Alias /blog /usr/share/wordpress
<Directory /usr/share/wordpress>
    Options FollowSymLinks
    AllowOverride Limit Options FileInfo
    DirectoryIndex index.php
    Order allow,deny
    Allow from all
</Directory>
<Directory /usr/share/wordpress/wp-content>
    Options FollowSymLinks
    Order allow,deny
    Allow from all
</Directory>
```

```
#Activar Sitio wordpress y modulo de rewrite
sudo a2ensite wordpress && sudo a2enmod rewrite
#Reiniciar apache
sudo service apache2 restart
```

Aplicación Web con múltiples capas: Wordpress



WebAPP
Apache
Php

10.201.0.XX

```
#sudo vi /etc/wordpress/config-<IP_INSTANCIA_WEBAPP>.php

sudo vi /etc/wordpress/config-10.201.0.XX.php
<?php
define('DB_NAME', 'wordpress');
define('DB_USER', 'wordpress');
define('DB_PASSWORD', 'telewordpress'); #password DB
define('DB_HOST', '10.201.Y.Z'); #Reemplazar por la IP_INSTANCIA_DB
define('DB_COLLATE', 'utf8_general_ci');
define('WP_CONTENT_DIR', '/usr/share/wordpress/wp-content');

?>
```

#Apuntar navegador a <http://10.201.0.XX/blog/>

NOTA: necesario agregar al grupo de seguridad (SecurityGroup) una regla permitiendo **HTTP entrante** desde **0.0.0.0/0**

Para caso IP_FLOTANTE *en un futuro cercano...*

```
define( 'WP_HOME', 'http://<IP\_FLOTANTE>/blog' );
define( 'WP_SITEURL', 'http://<IP\_FLOTANTE>/blog' );
```

Aplicación Web con múltiples capas: Wordpress



Load
Balancer

nginx

```
1) Creamos instancia ubuntu_minimal_1804 , llamarla: wp-fe-01
2) Nos logueamos: ssh ubuntu@<IP_instancia>
3) Ejecutamos:

# Truquillo de optimización de disco:
sudo mount -o remount,nobarrier,commit=120 /

# Install nginx, and helper tools
sudo apt-get update && sudo apt-get install nginx vim-tiny curl

# Remove default nginx site
sudo rm /etc/nginx/sites-enabled/default
# Configure nginx as reverse proxy
sudo vi /etc/nginx/conf.d/lb.conf
server {
listen 80;
location / {
proxy_pass http://10.201.0.XX; # my backend WP
}
}

sudo service nginx restart
# Should show "backend" (WP) content
curl http://localhost/
# Ver logs de acceso en el frontend proxy
journalctl -u nginx
tail -f /var/log/nginx/access.log
# Chequear que puedo navegar:
http://<IP de wp-fe-01>
```

Cloud Classic

Tema Floating IP

```
# En la máquina de la DB
```

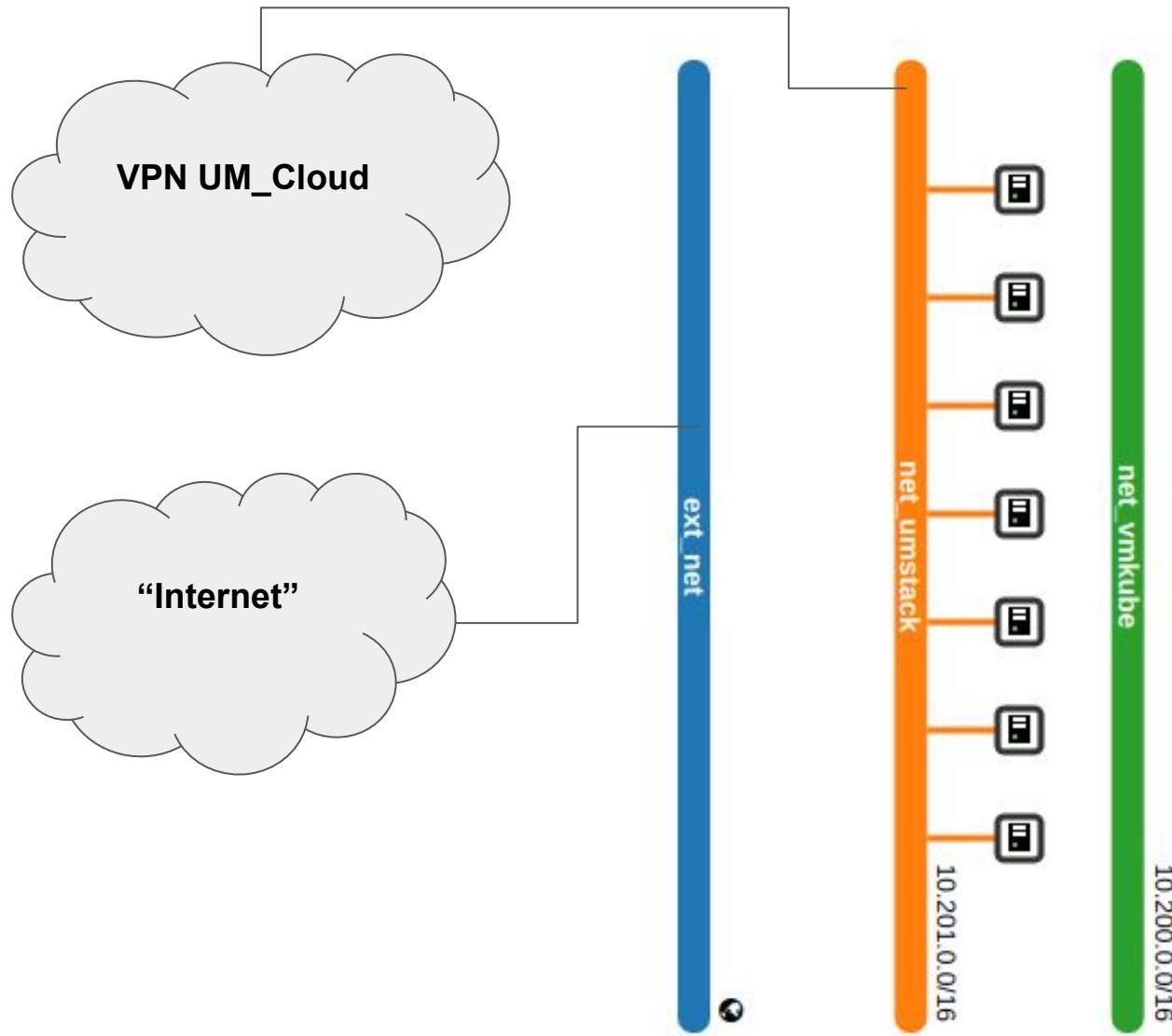
```
#Para actualizar la IP flotante en la config de wordpress
echo 'update wp_options set option_value = "http://<ip_flotante>/blog" where
option_name="siteurl"' | sudo mysql wordpress
```

```
echo 'update wp_options set option_value = "http://<ip_flotante>/blog" where
option_name="home"' | sudo mysql wordpress
```

Cloud Classic

Caso Práctico ++

Cloud Networking : Crear Red



Cloud Networking: Crear Red

openstack Default • diego.navarro_um.edu.ar_project ▾

Proyecto ▾

Acceso a la API

Compute ▾

Volúmenes ▾

Red ▾

Topología de red

Redes

Routers

Grupos de seguridad

IPs flotantes

Almacén de objetos ▾

Identity ▾

Topología de red

Gráfico

Pequeño Normal

Crear red

Red Subred Detalles de Subred

Nombre de la red

navarrow-net

Activar Estado del Administrador

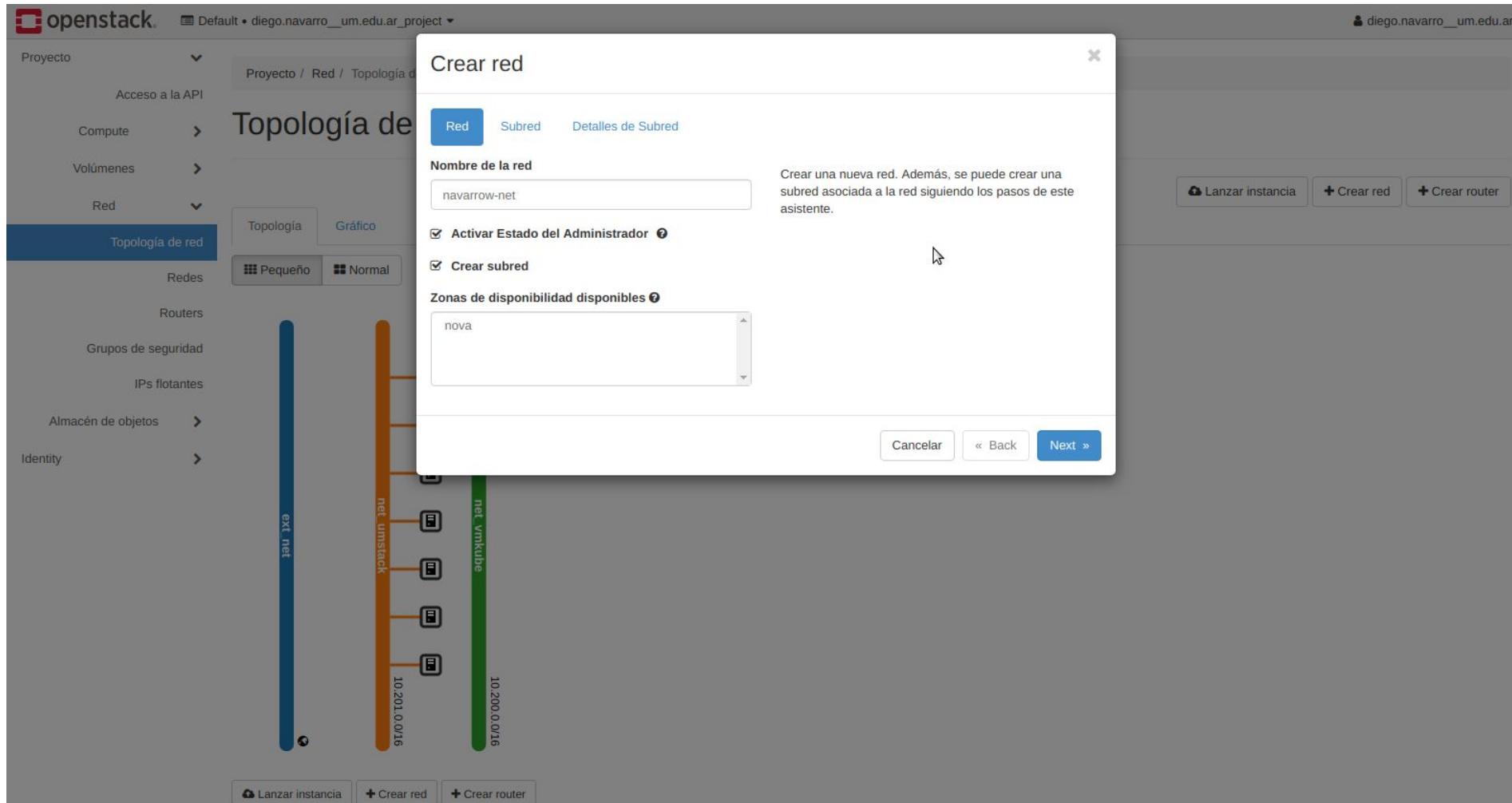
Crear subred

Zonas de disponibilidad disponibles

nova

Cancelar « Back Next »

Lanzar instancia + Crear red + Crear router



Nombre de la red:
usuario-red

Cloud Networking: Crear Red



openstack Default • diego.navarro_um.edu.ar_project ▾

Proyecto ▾

Acceso a la API

Compute ▾

Volúmenes

Red ▾

Topología de red

Redes

Routers

Grupos de seguridad

IPs flotantes

Almacén de objetos ▾

Identity ▾

Proyecto / Red / Topología de red

Crear red Subred Detalles de Subred

Nombre de subred navarrow-subnet

Direcciones de red 172.19.0.0/24

Versión de IP IPv4

IP de la puerta de enlace ▾

Deshabilitar puerta de enlace

Lanzar instancia + Crear red + Crear router

ext net

net umstack

ube

172.19.0.0/24

10.20.0.0/16

10.20.0.0/16

Cancelar « Back Next »

Nombre de subred:
usuario-subnet

Direcciones de red:
172.19.0.0/24

Nombre de subred:
usuario-subnet

Direcciones de red:
172.19.0.0/24

Cloud Networking: Crear Router

The screenshot shows the openstack interface with the 'openstack' logo at the top left. The top navigation bar includes 'Default • diego.navarro_um.edu.ar_project' and a user icon 'diego.navarro_um.edu.ar'. On the left, a sidebar menu lists 'Proyecto', 'Acceso a la API', 'Compute', 'Volúmenes', 'Red', 'Topología de red' (selected), 'Redes', 'Routers', 'Grupos de seguridad', 'IPs flotantes', 'Almacén de objetos', and 'Identity'. The main content area has a title 'Topología de red' and tabs 'Topología' and 'Gráfico'. Below these are buttons for 'Pequeño' and 'Normal'. A central modal window titled 'Crear un router' is open, containing fields for 'Nombre del router' (set to 'navarrow-router'), a checked checkbox for 'Activar Estado del Administrador', and a dropdown for 'Red externa' (set to 'ext_net'). A list of 'Zonas de disponibilidad disponibles' shows 'nova'. At the bottom of the modal are 'Cancelar' and 'Crear router' buttons. In the background, a network topology diagram is visible, featuring several vertical bars representing networks ('ext_net', 'navarrow-net', 'net_unstack', 'net_vmkube') and a green bar representing a subnet ('172.19.0.0/16'). IP addresses like '172.19.0.0/24' and '10.200.0.0/16' are also shown. At the very bottom of the screen are three buttons: 'Lanzar instancia', '+ Crear red', and '+ Crear router'.

Nombre de router:

Usuario-router

Red externa:

ext net

Cloud Networking: Crear Router, Añadir Interfaz



Screenshot of the OpenStack Horizon interface showing the creation of a router and adding an interface.

The main navigation bar shows "Default • diego.navarro_um.edu.ar_project". The left sidebar has sections: Proyecto, Acceso a la API, Compute, Volúmenes, Red (selected), Topología de red, Redes, Routers, Grupos de seguridad, IPs flotantes, Almacén de objetos, and Identity.

The "Topología de red" tab is selected under "Red". It displays a network diagram with nodes: "ext.net" (blue), "navarro" (orange), "net_umstack" (green), and "net_vmkube" (red). The "navarro" node has an interface "gatewaybd72..." connected to "ext.net". Below the diagram is a table of interfaces:

| Interface | MAC Address | IP Address |
|----------------|-------------|------------|
| gatewaybd72... | Ninguno | Ninguno |

Buttons at the bottom of the diagram area include "Lanzar instancia", "+ Crear red", and "+ Crear router".

A modal window titled "Añadir interfaz" is open. It contains fields for "Subred *": "Seleccionar subred" (dropdown menu), "Dirección IP (opcional)": an empty input field, and a "Descripción:" text area with the following content:

Puede conectar una subred concreta al router.
Si no especifica aquí una dirección IP, se utilizará la dirección IP de la puerta de enlace de la subred seleccionada para la nueva interfaz del router. Si la dirección IP de la puerta de enlace ya se está usando, debe utilizar una dirección diferente del rango de la subred seleccionada.

Buttons at the bottom of the modal are "Cancelar" and "Enviar".

Cloud Networking

openstack. Default • diego.navarro_um.edu.ar_project ▾ diego.navarro_um.edu.ar

Proyecto ▾

Acceso a la API

Compute

Volúmenes

Red ▾

Topología de red

Redes

Routers

Grupos de seguridad

IPs flotantes

Almacén de objetos

Identity

Proyecto / Red / Topología de red

Topología de red

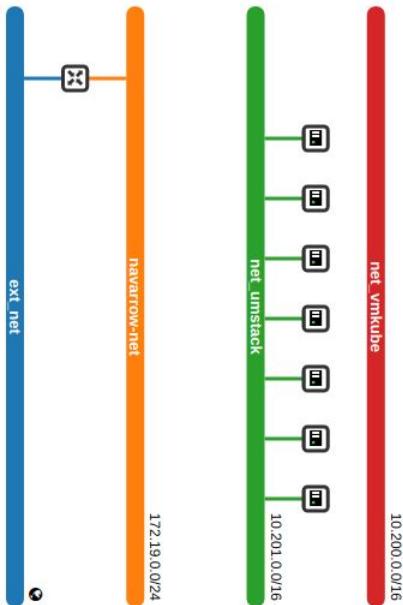
Topología Gráfico

Pequeño Normal

Lanzar instancia + Crear red + Crear router

ext_net navarro-net net_umstack net_vmkube

172.19.0.0/24 10.0.1.0/16 10.200.0.0/16



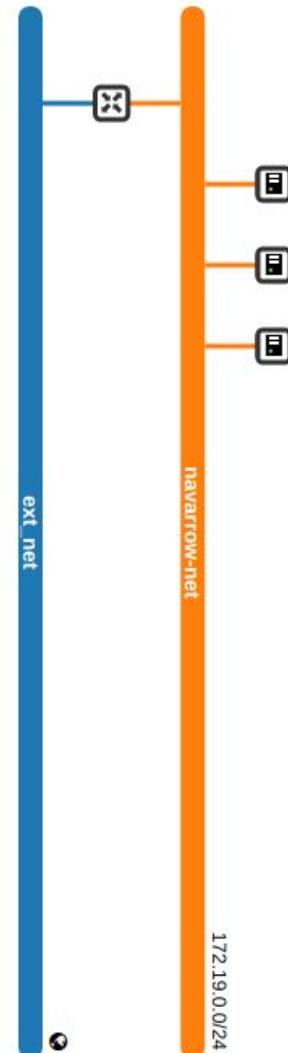
Lanzar instancia + Crear red + Crear router

Instancias

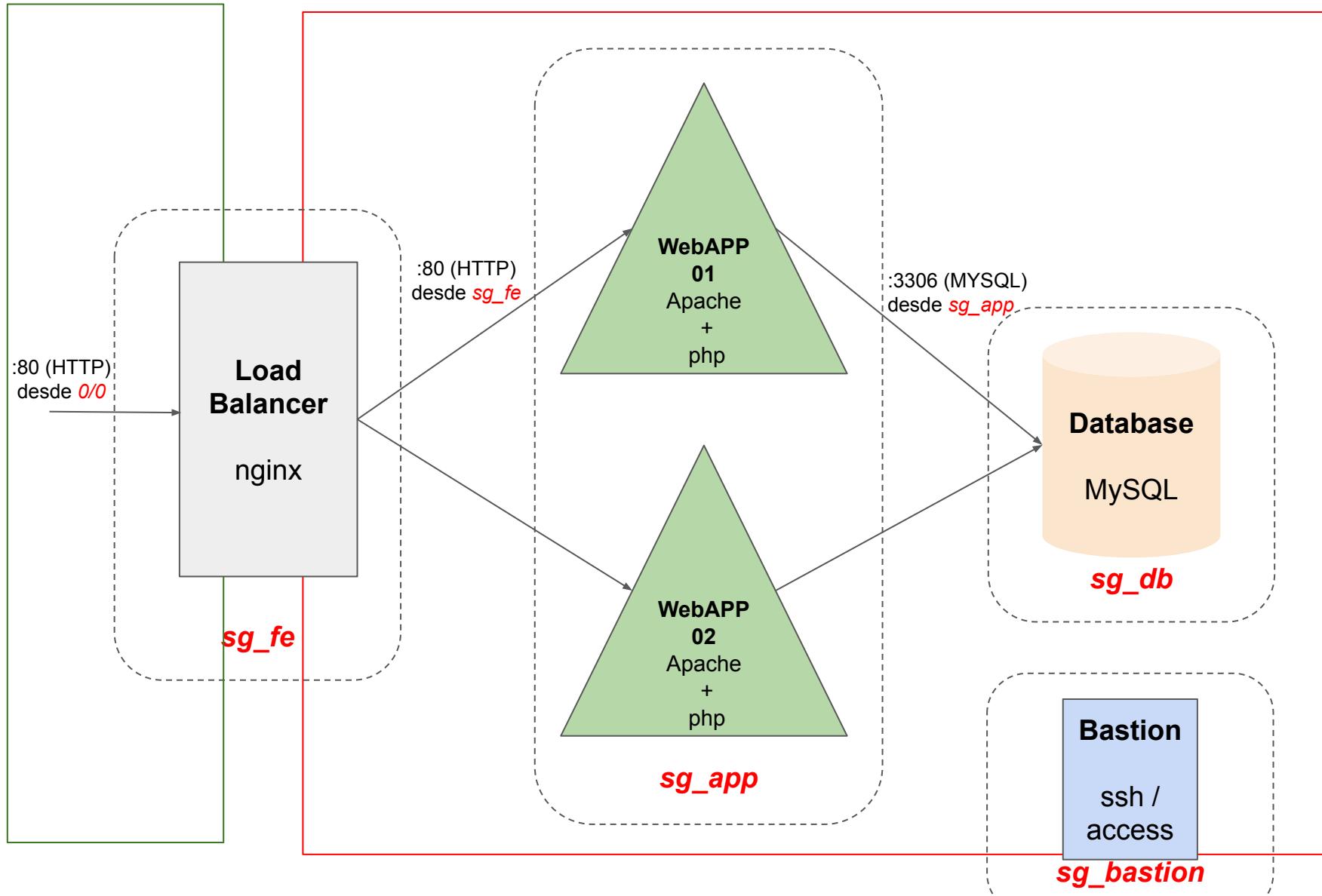
Creamos nuevamente las instancias sobre la nueva red Privada y Asignamos la IP Elástica a una [bastión] [remember -> ssh -A]

| | Nombre de la instancia | Nombre de la imagen | Dirección IP | Sabor | Par de claves | Estado | Zona de Disponibilidad |
|--------------------------|------------------------|---------------------|--|------------|---------------|--------|--|
| <input type="checkbox"/> | db-priv | ubuntu_minimal_1804 | 172.19.0.10 | m1.c1m1d20 | navarrow | Activo |  workers-az |
| <input type="checkbox"/> | wp-priv | ubuntu_minimal_1804 | 172.19.0.8 | m1.c1m1d20 | navarrow | Activo |  workers-az |
| <input type="checkbox"/> | lb-priv | ubuntu_minimal_1804 | 172.19.0.6 IPs flotantes: 192.168.3.85 | m1.c1m1d20 | navarrow | Activo |  workers-az |

Rehacemos la configuración de la clase anterior con el nuevo
direcccionamiento



Aplicación Web con múltiples capas: Wordpress



Grupos de Seguridad



openstack. Default • diego.navarro_um.edu.ar_project ▾

Proyecto ▾

Proyecto / Red / Grupos de seguridad

Acceso a la API

Compute

Volúmenes

Red ▾

Topología de red

Redes

Routers

Grupos de seguridad

IPs flotantes

Almacén de objetos

Identity

Grupos de seguridad

Displaying 4 items

| | Nombre | ID del grupo de seguridad | Descripción | Actions |
|--------------------------|---------|--------------------------------------|------------------------|-------------------------------------|
| <input type="checkbox"/> | default | 162383d7-b6df-45d7-b759-5444eb840174 | Default security group | <button>Administrar reglas</button> |
| <input type="checkbox"/> | sg-app | fd72c714-f165-4467-922d-d44e4f4ff63e | | <button>Administrar reglas</button> |
| <input type="checkbox"/> | sg-base | ce5dd298-c81c-4f49-aa91-760aac2253f7 | | <button>Administrar reglas</button> |
| <input type="checkbox"/> | sg-lb | 077a42c6-8368-4777-800b-02ec889b2e4f | | <button>Administrar reglas</button> |

Filtrar

+ Crear grupo de seguridad

Eliminar Grupos de Seguridad

Displaying 4 items

Grupos de Seguridad: sg-base [a todos]



openstack. Default • diego.navarro_um.edu.ar_project ▾

Proyecto ▾

Acceso a la API

Compute > Administrar Reglas de Grupo de Seguridad: sg-base (ce5dd298-c81c-4f49-aa91-760aac2253f7)

Volúmenes >

Red ▾

Topología de red

Redes

Routers

Grupos de seguridad

IPs flotantes

Almacén de objetos >

Identity >

Displaying 3 items

| <input type="checkbox"/> | Dirección | Tipo Ethernet | Protocolo IP | Rango de puertos | Prefijo de IP Remota | Grupo de Seguridad Remoto | Description | Actions |
|--------------------------|-----------|---------------|--------------|------------------|----------------------|---------------------------|-------------|---------------------------------|
| <input type="checkbox"/> | Saliente | IPv4 | Cualquier | Cualquier | 0.0.0.0/0 | - | - | <button>Eliminar Regla</button> |
| <input type="checkbox"/> | Saliente | IPv6 | Cualquier | Cualquier | ::/0 | - | - | <button>Eliminar Regla</button> |
| <input type="checkbox"/> | Entrante | IPv4 | ICMP | Cualquier | 0.0.0.0/0 | - | - | <button>Eliminar Regla</button> |

Displaying 3 items

Grupos de Seguridad: sg-lb [solo Load Balancer]



openstack. Default • diego.navarro_um.edu.ar_project ▾

Proyecto ▾

Acceso a la API

Compute

Volúmenes

Red ▾

Topología de red

Redes

Routers

Grupos de seguridad

IPs flotantes

Almacén de objetos ▾

Identity ▾

Default • diego.navarro_um.edu.ar_project ▾

diego.navarro_um.edu.ar ▾

Proyecto / Red / Grupos de seguridad / Administrar Reglas de Grup...

Administrador Reglas de Grupo de Seguridad: sg-lb (077a42c6-8368-4777-800b-02ec889b2e4f)

+ Agregar regla Eliminar Reglas

| Dirección | Tipo Ethernet | Protocolo IP | Rango de puertos | Prefijo de IP Remota | Grupo de Seguridad Remoto | Description | Actions |
|--------------------------|---------------|--------------|------------------|----------------------|---------------------------|-------------|---------------------------------|
| <input type="checkbox"/> | Saliente | IPv4 | Cualquier | Cualquier | 0.0.0.0/0 | - | <button>Eliminar Regla</button> |
| <input type="checkbox"/> | Saliente | IPv6 | Cualquier | Cualquier | ::/0 | - | <button>Eliminar Regla</button> |
| <input type="checkbox"/> | Entrante | IPv4 | TCP | 22 (SSH) | 0.0.0.0/0 | - | <button>Eliminar Regla</button> |
| <input type="checkbox"/> | Entrante | IPv4 | TCP | 80 (HTTP) | 0.0.0.0/0 | - | <button>Eliminar Regla</button> |
| <input type="checkbox"/> | Entrante | IPv4 | TCP | 443 (HTTPS) | 0.0.0.0/0 | - | <button>Eliminar Regla</button> |

Displaying 5 items

Displaying 5 items

Grupos de Seguridad: sg-app [solo a la WebApp]



openstack. Default • diego.navarro_um.edu.ar_project ▾

Proyecto ▾

Acceso a la API

Compute

Volúmenes

Red ▾

Topología de red

Redes

Routers

Grupos de seguridad

IPs flotantes

Almacén de objetos ▾

Identity ▾

Default • diego.navarro_um.edu.ar_project ▾

diego.navarro_um.edu.ar ▾

Proyecto / Red / Grupos de seguridad / Administrar Reglas de Grup...

Administrador Reglas de Grupo de Seguridad: sg-app (fd72c714-f165-4467-922d-d44e4f4ff63e)

Displaying 4 items

| <input type="checkbox"/> | Dirección | Tipo Ethernet | Protocolo IP | Rango de puertos | Prefijo de IP Remota | Grupo de Seguridad Remoto | Description | Actions |
|--------------------------|-----------|---------------|--------------|------------------|----------------------|---------------------------|-------------|---------------------------------|
| <input type="checkbox"/> | Saliente | IPv4 | Cualquier | Cualquier | 0.0.0.0/0 | - | - | <button>Eliminar Regla</button> |
| <input type="checkbox"/> | Saliente | IPv6 | Cualquier | Cualquier | ::/0 | - | - | <button>Eliminar Regla</button> |
| <input type="checkbox"/> | Entrante | IPv4 | TCP | 22 (SSH) | - | sg-lb | - | <button>Eliminar Regla</button> |
| <input type="checkbox"/> | Entrante | IPv4 | TCP | 80 (HTTP) | - | sg-lb | - | <button>Eliminar Regla</button> |

Displaying 4 items



Grupos de Seguridad: sg-db [solo a la db]



openstack. Default • diego.navarro_um.edu.ar_project ▾

Proyecto ▾

Acceso a la API

Compute

Volúmenes

Red ▾

Topología de red

Redes

Routers

Grupos de seguridad

IPs flotantes

Almacén de objetos ▾

Identity ▾

Default • diego.navarro_um.edu.ar_project ▾

diego.navarro_um.edu.ar ▾

Proyecto / Red / Grupos de seguridad / Administrar Reglas de Grup...

Administrador Reglas de Grupo de Seguridad: sg-db (24f6f635-0088-47d3-a513-dc32c46bd2db)

+ Agregar regla Eliminar Reglas

| | Dirección | Tipo Ethernet | Protocolo IP | Rango de puertos | Prefijo de IP Remota | Grupo de Seguridad Remoto | Description | Actions |
|--------------------------|-----------|---------------|--------------|------------------|----------------------|---------------------------|-------------|---------------------------------|
| <input type="checkbox"/> | Saliente | IPv4 | Cualquier | Cualquier | 0.0.0.0/0 | - | - | <button>Eliminar Regla</button> |
| <input type="checkbox"/> | Saliente | IPv6 | Cualquier | Cualquier | ::/0 | - | - | <button>Eliminar Regla</button> |
| <input type="checkbox"/> | Entrante | IPv4 | TCP | 22 (SSH) | - | sg-lb | - | <button>Eliminar Regla</button> |
| <input type="checkbox"/> | Entrante | IPv4 | TCP | 3306 (MYSQL) | - | sg-app | - | <button>Eliminar Regla</button> |

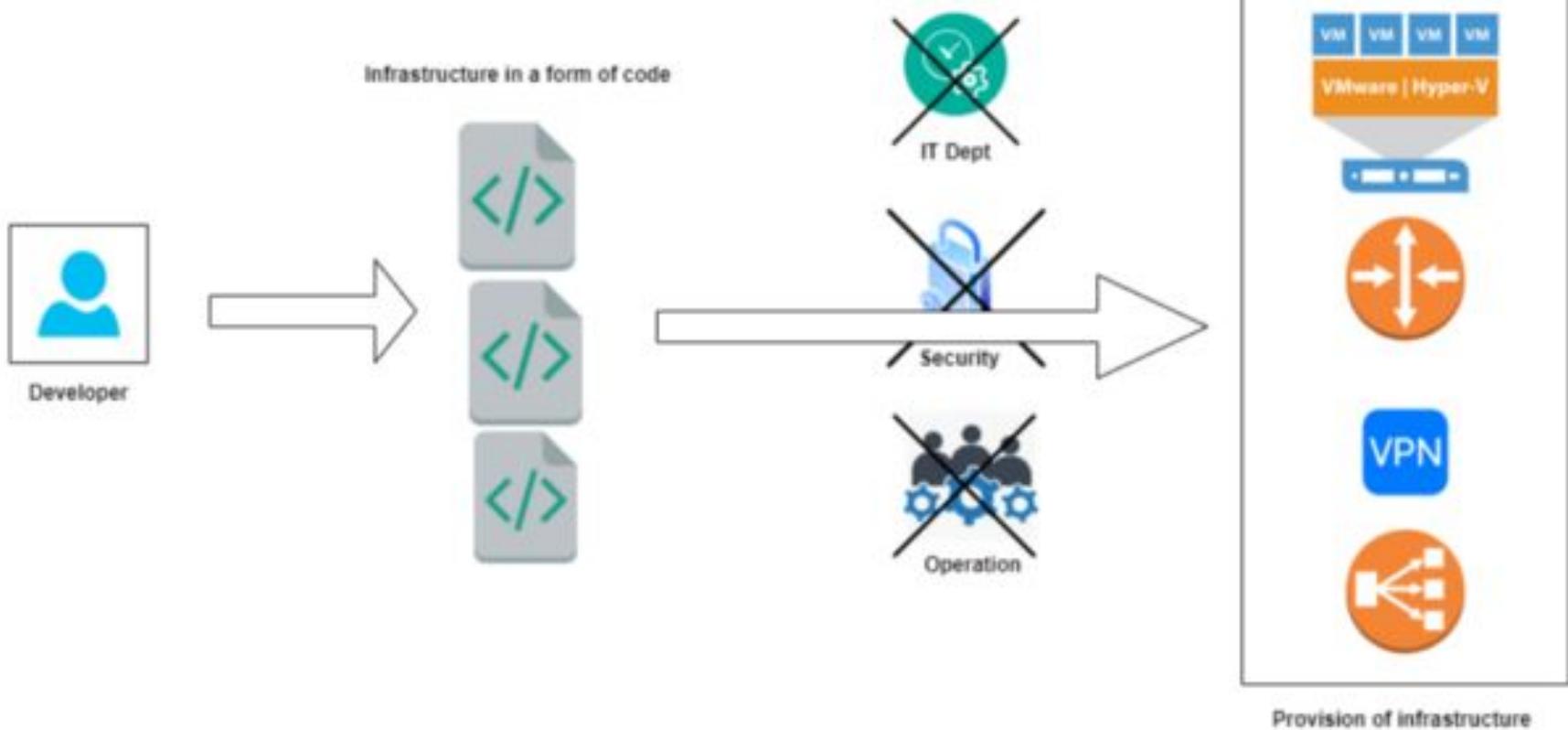
Displaying 4 items

Displaying 4 items

IaC

Infra as Code

Infraestructura como código: IaC



Cloud Native

5

Platform as Code

Provisioning of platform elements as Code;
Examples: Kubernetes, RedHat OpenShift, KubePlus

4

Container as a Service

Provisioning and management of Containers as a
Service; Examples: Kubernetes, GKE, EKS

3

Infrastructure as Code

Provisioning of infrastructure elements as Code;
Examples: AWS Cloud Formation, Terraform

2

Platform as a Service

Provisioning of platform elements as a Service;
Examples: Heroku, AWS Elastic Beanstalk, Google
App Engine

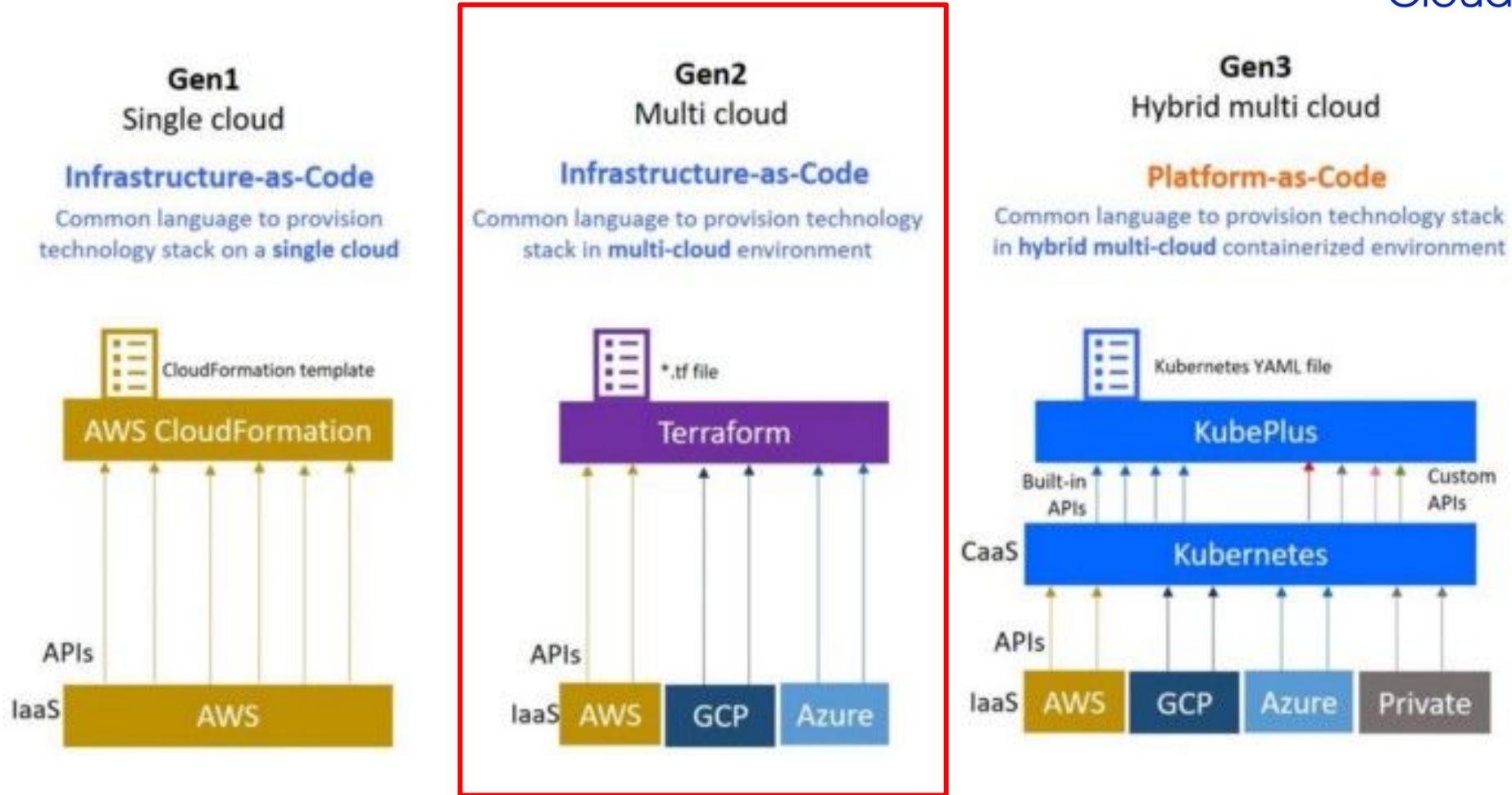
1

Infrastructure as a Service

Provisioning of infrastructure elements as a Service; Examples: AWS EC2, Google
Compute Engine (GCE)

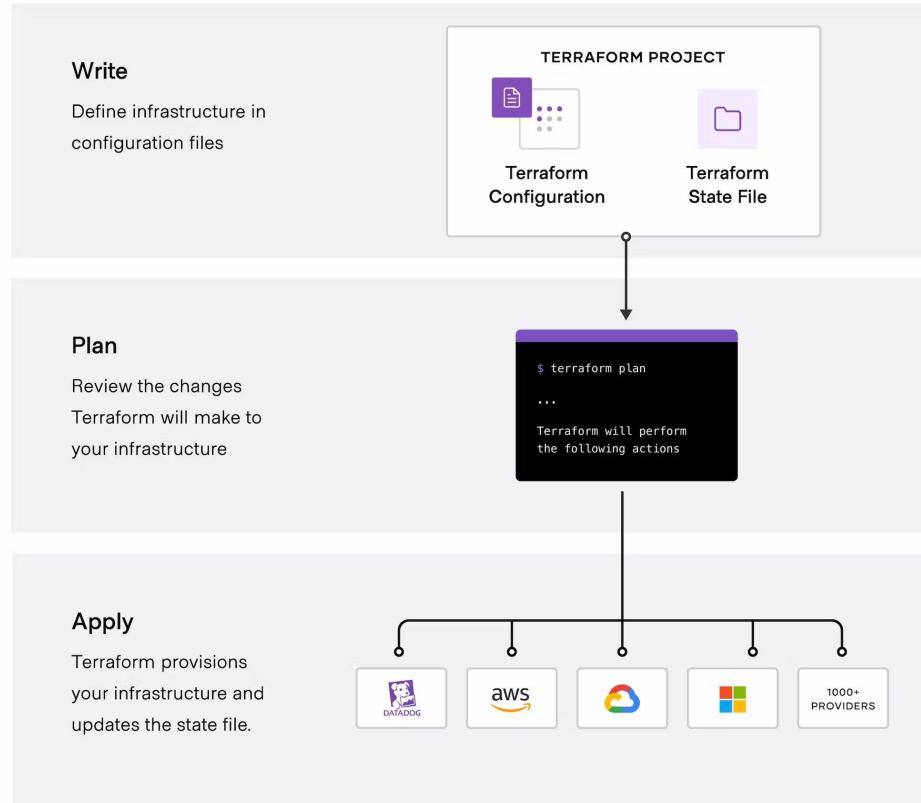
Cloud Classic

Plataforma como código



<https://medium.com/@cloudark/kubernetes-and-the-future-of-as-code-systems-b1b2de312742>

IaC: Terraform



Terraform: Setup

```
## Descargar Terraform
```

```
wget https://releases.hashicorp.com/terraform/1.4.6/terraform_1.4.6_linux_amd64.zip
```

```
## Descomprimimos
```

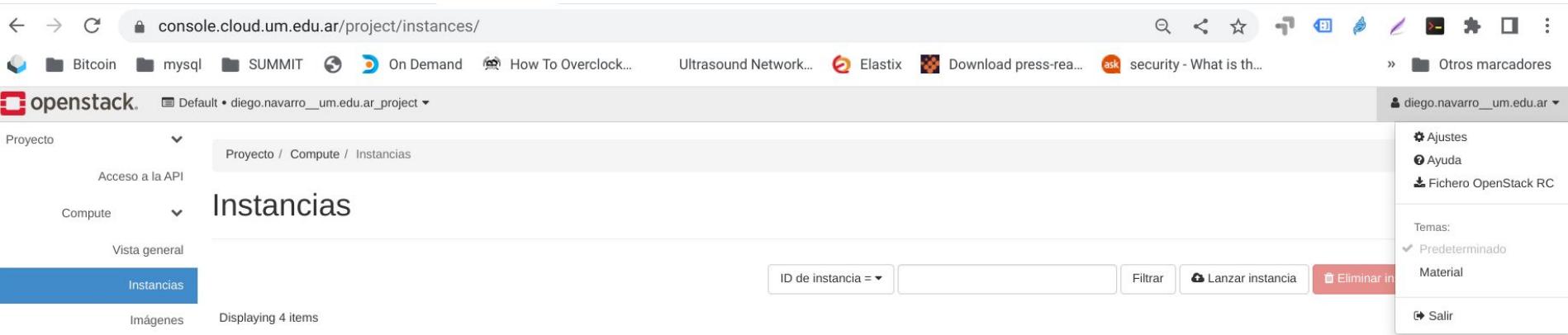
```
unzip terraform_1.4.6_linux_amd64.zip
```

```
#probamos
```

```
./terraform
```

```
#Descargamos fichero openrc (Credenciales API Openstack)
```

```
## ej: diego.navarro_um.edu.ar_project-openrc
```



The screenshot shows the OpenStack Compute (Nova) dashboard at console.cloud.um.edu.ar/project/instances/. The user is in the 'diego.navarro_um.edu.ar_project' project. The 'Instancias' (Instances) tab is selected under the 'Compute' menu. The page displays a table with 4 items, though only 1 instance is visible. The instance details show an IP address of 10.0.0.4, a status of 'Running', and a flavor of 'm1.small'. There are buttons for 'Lanzar instancia' (Launch instance) and 'Eliminar instancia' (Delete instance). The top navigation bar includes links for Bitcoin, mysql, SUMMIT, On Demand, How To Overclock..., Ultrasound Network..., Elastix, Download press-re... (ask), security - What is th..., and Otros marcadores. The bottom right corner shows a sidebar with Ajustes, Ayuda, Fichero OpenStack RC, Temas: Predeterminado, Material, and Salir.

Terraform: Setup

```
## Cargamos en el ambiente las credenciales
source diego.navarro_um.edu.ar_project-openrc.sh
## Por seguridad nos pide que coloquemos nuestro password de Horizon
```

```
##Seteamos el provider Openstack
vi versions.tf
```

```
terraform {
  required_providers {
    openstack = {
      source = "terraform-provider-openstack/openstack"
    }
  }
  required_version = ">= 0.13"
}
```

```
##Inicializamos Terraform
./terraform init
```

My-UM-Cloud

ZeroTier_Config Cloud_Credentials Help

username: diego.navarro_um.edu.ar

password:

[Click aqui para ir al Dashboard](#)

```
Initializing the backend...
Initializing provider plugins...
- Finding latest version of terraform-provider-openstack/openstack...
- Installing terraform-provider-openstack/openstack v1.51.1...
- Installed terraform-provider-openstack/openstack v1.51.1 (self-signed, key ID 4F80527A391BEFD2)

Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it here:
https://www.terraform.io/docs/cli/plugins/signing.html

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

Terraform: Nuestra Primer VM - IaC

vi vm.tf

```

data "openstack_images_image_v2" "ubuntu_2204" {
  name      = "ubuntu_2204"
  most_recent = true
}
data "openstack_compute_flavor_v2" "small" {
  vcpus = 1
  ram   = 2048
}
data "openstack_networking_network_v2" "net_umstack" {
  name = "net_umstack"
}
resource "openstack_networking_port_v2" "vm_port_umcloud" {
  name      = "vm_port_umcloud"
  network_id      = data.openstack_networking_network_v2.net_umstack.id
  admin_state_up = "true"
}
resource "openstack_compute_instance_v2" "terraform_vm" {
  lifecycle {
    ignore_changes = [
      image_id,
    ]
  }
  name          = "terraform_vm"
  image_id      = data.openstack_images_image_v2.ubuntu_2204.id
  flavor_id      = data.openstack_compute_flavor_v2.small.id
  key_pair      = "<REEMPLAZAR-POR-MIS-KEYS>"
  security_groups  = ["default"]
  availability_zone = "nodos-amd-2022"
  network {
    port = openstack_networking_port_v2.vm_port_umcloud.id
  }
  user_data = "${file("init.sh")}"
}

```

vi init.sh

```

#!/bin/bash
apt-get update -y
apt-get install -y nginx vim-tiny curl

```

Terraform: Nuestra Primer VM - IaC



##Aplicamos Terraform

./terraform apply

```
data.openstack_networking_network_v2.net_umstack: Reading...
data.openstack_compute_flavor_v2.small: Reading...
data.openstack_images_image_v2.ubuntu_2204: Reading...
data.openstack_images_image_v2.ubuntu_2204: Read complete after 0s [id=ece749b8-1736-40e7-ade5-e41114111b6a]
data.openstack_compute_flavor_v2.small: Read complete after 0s [id=2d357d3d-32c1-4af8-81dd-a71a7d7cf303]
data.openstack_networking_network_v2.net_umstack: Read complete after 0s [id=6f728afe-d289-4ccc-b108-87c3fbfc30c]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# openstack_compute_instance_v2.terraform_vm will be created
resource "openstack_compute_instance_v2" "terraform_vm" {
  access_ip_v4      = (known after apply)
  access_ip_v6      = (known after apply)
  all_metadata       = (known after apply)
  all_tags          = (known after apply)
  availability_zone = "nodos-and-2022"
  created           = (known after apply)
  flavor_id          = "2d357d3d-32c1-4af8-81dd-a71a7d7cf303"
  flavor_name        = (known after apply)
  force_delete       = false
  id                = (known after apply)
  image_id          = "ece749b8-1736-40e7-ade5-e41114111b6a"
  key_pair           = "navarrow"
  name              = "terraform_vm"
  power_state        = "active"
  region             = (known after apply)
  security_groups    = [
    + "default",
  ]
  stop_before_destroy = false
  updated            = (known after apply)
  user_data          = "cd983027ddbdaec321bf4ea1217a4812878dd2398"

  network {
    access_network = false
    fixed_ip_v4    = (known after apply)
    fixed_ip_v6    = (known after apply)
    floating_ip    = (known after apply)
    mac             = (known after apply)
    name            = (known after apply)
    port            = (known after apply)
    uuid            = (known after apply)
  }
}

# openstack_networking_port_v2.vm_port_umcloud will be created
resource "openstack_networking_port_v2" "vm_port_umcloud" {
  admin_state_up      = true
  all_fixed_ips       = (known after apply)
  all_security_group_ids = (known after apply)
  all_tags            = (known after apply)
  device_id           = (known after apply)
  device_owner         = (known after apply)
  dns_assignment       = (known after apply)
  dns_name             = (known after apply)
  id                  = (known after apply)
  mac_address          = (known after apply)
  name                = "vm_port_umcloud"
  network_id           = "6f728afe-d289-4ccc-b108-87c3fbfc30c"
  port_security_enabled = (known after apply)
  qos_policy_id        = (known after apply)
  region               = (known after apply)
  tenant_id            = (known after apply)
}

Plan: 2 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

Enter a value: █
```

Terraform: Nuestra Primer VM - IaC



##Verificamos Terraform

./terraform plan

```
navarrow@penguin:~/Proyectos/umcloud/terraform$ ./terraform plan
data.openstack_images_image_v2.ubuntu_2204: Reading...
data.openstack_networking_network_v2.net_umstack: Reading...
data.openstack_compute_flavor_v2.small: Reading...
data.openstack_images_image_v2.ubuntu_2204: Read complete after 1s [id=ece749b8-1736-40e7-ade5-e41114111b6a]
data.openstack_compute_flavor_v2.small: Read complete after 1s [id=2d357d3d-32c1-4af8-81dd-a71a7d7cf303]
data.openstack_networking_network_v2.net_umstack: Read complete after 1s [id=6f728afe-d289-4ccc-b108-87c3fbfc30c]
openstack_networking_port_v2.vm_port_umcloud: Refreshing state... [id=49a11c81-751c-40fb-b6d7-c80b58b3dd7a]
openstack_compute_instance_v2.terraform_vm: Refreshing state... [id=1cf8281a-2131-4b4e-a618-78199687db56]
```

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.

##Vemos en el Dashboard de Horizon que tenemos una nueva VM !!!

Instancias

| | ID de instancia = ▾ | | Filtrar | Lanzar instancia | Eliminar instancias | More Actions | | | | | | |
|--------------------------|---------------------|-------------|---|------------------|---------------------|-------------------|------|----------------|---------|-----------|-----------|------------------------------------|
| Instance Name | Image Name | IP Address | Flavor | Key Pair | Status | Availability Zone | Task | Power State | Age | Actions | | |
| <input type="checkbox"/> | terraform_vm | ubuntu_2204 | 10.201.4.57, 2002:c833:29b0:2:f816:3eff:feef:b76d | m1.small | navarrow | Activo | | nodos-amd-2022 | Ninguno | Corriendo | 2 minutos | <button>Crear instantánea</button> |

IaC Integrando Conceptos: Caso +++

Desarrollaremos todo el caso ahora de manera "programática"

Para ello:

- Verificar instalación Terraform accesible (Clase anterior)
 - wget https://releases.hashicorp.com/terraform/1.4.6/terraform_1.4.6_linux_amd64.zip
 - ## Descomprimimos
 - unzip terraform_1.4.6_linux_amd64.zip
 - #probamos
 - ./terraform
 - #Lo ponemos invocable desde cualquier lado.
 - sudo cp ./terraform /usr/local/bin
- Cargamos en el ambiente las credenciales
- source diego.navarro_um.edu.ar_project-openrc.sh
- Eliminar todo lo que tengamos creado:
 - VMs, Routers, Nets, SGs, Floating IP
- Clonar el repo del curso
 - git clone <https://github.com/umcloud/clases-devops>
- Ingresar a la clase 02
 - cd clases-devops/c02-terraform_fe_app_db/
- Iremos paso a paso

Repo GitHub

<https://github.com/umcloud>



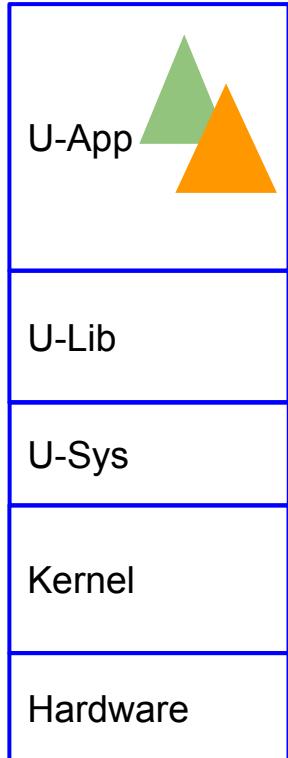
Mirar los Makefile 😊 ... mucha 🎉

Cloud native

Cloud Native - containers son cool, pero ...



El Monolito

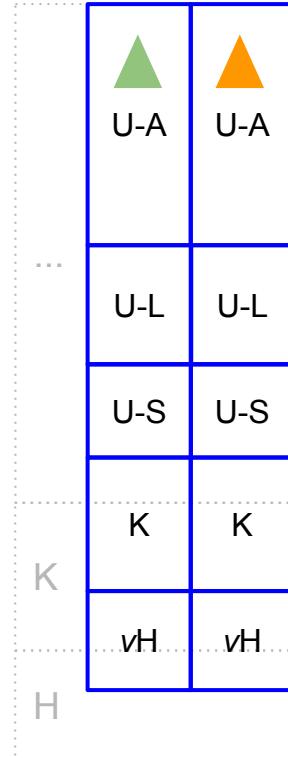


Cloud Native - “un poco de arqueología ...”

El Monolito



VM

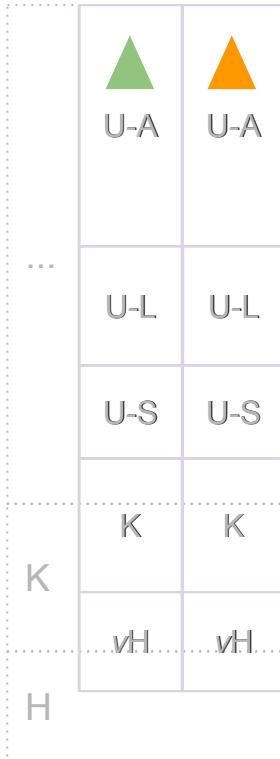


Cloud Native - “un poco de arqueología ...”

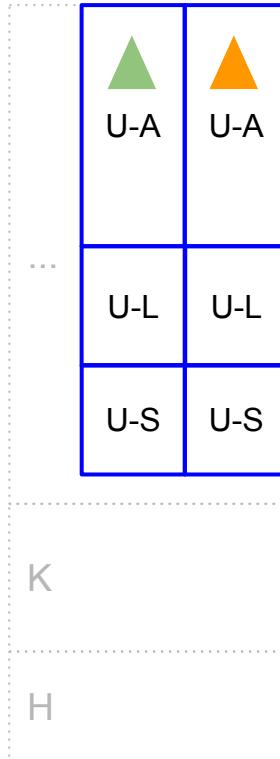
El Monolito



VM



Container
(LXC/LXD)

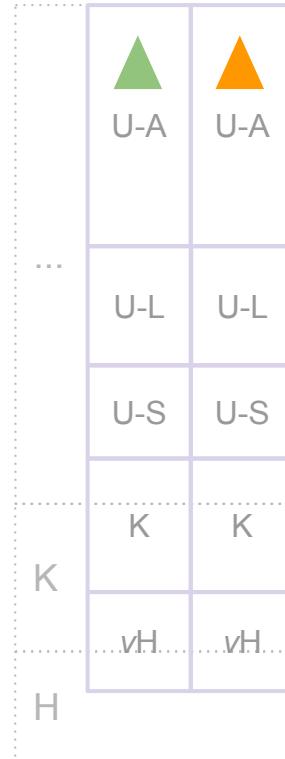


Cloud Native - “un poco de arqueología ...”

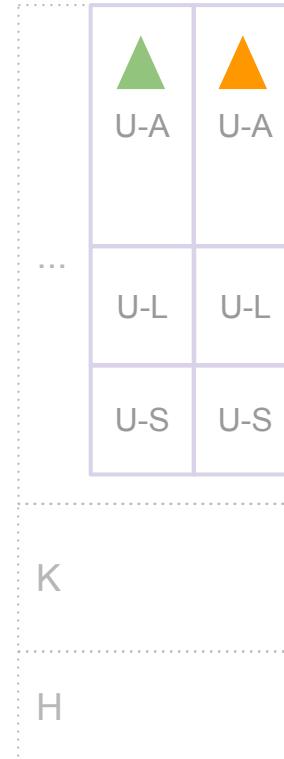
El Monolito



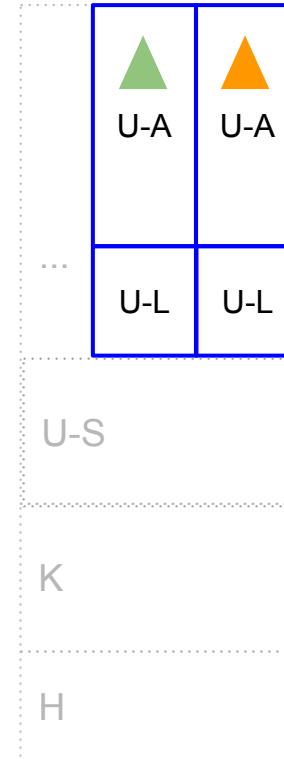
VM



Container
(LXC/LXD)



Container
(Docker)

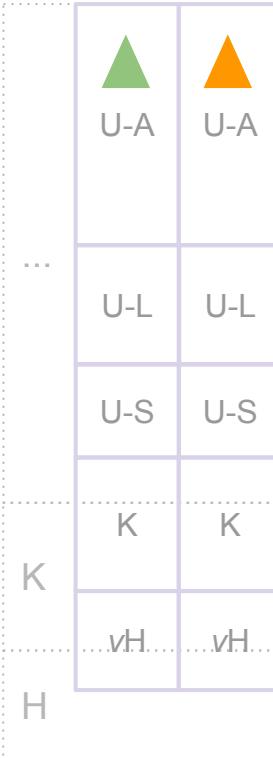


Cloud Native - “un poco de arqueología ...”

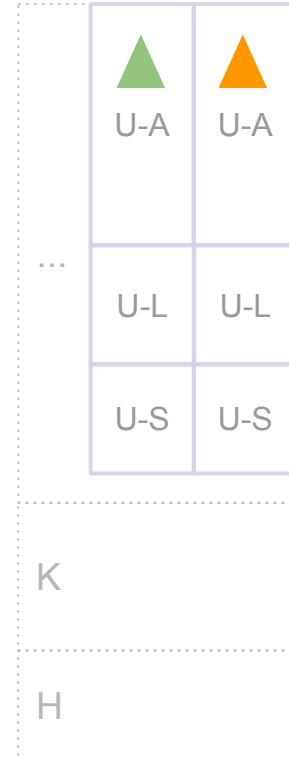
El Monolito



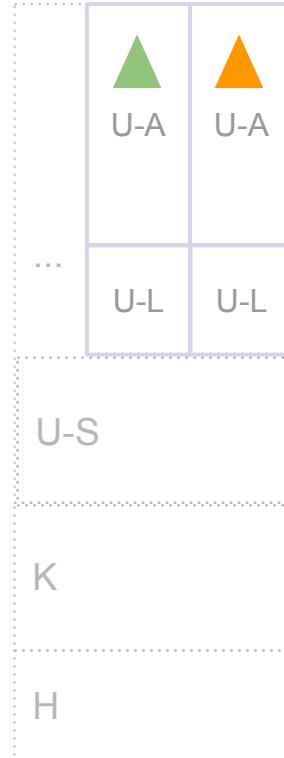
VM



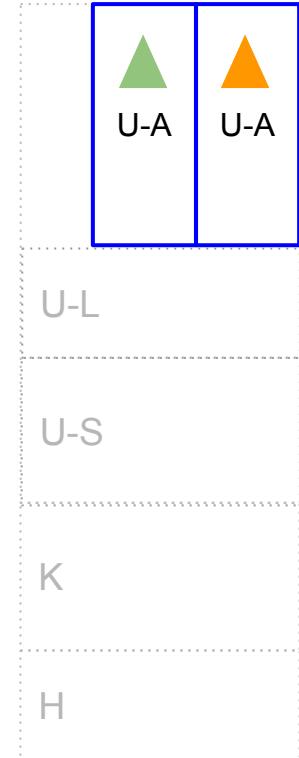
Container
(LXC/LXD)



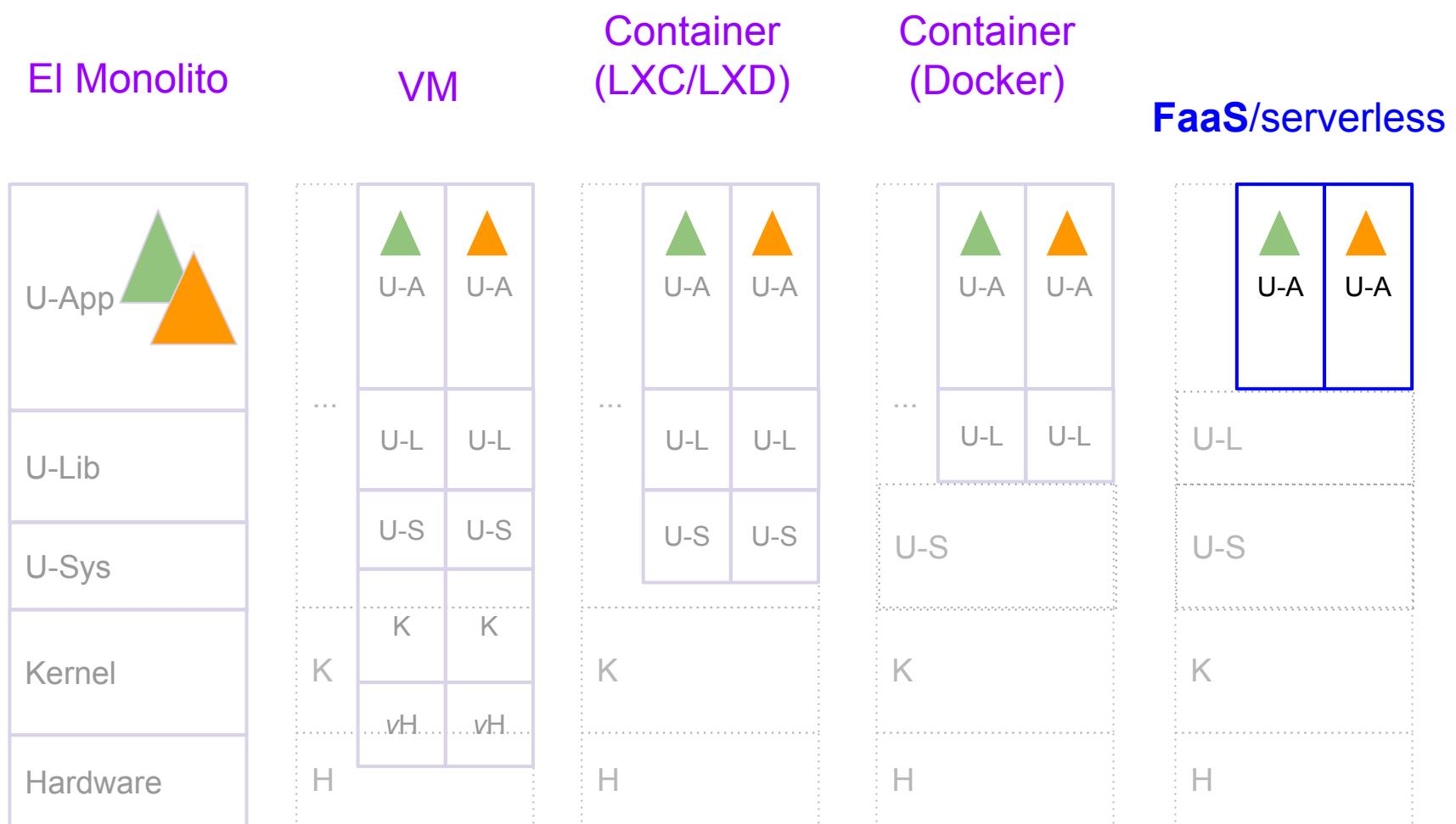
Container
(Docker)



?

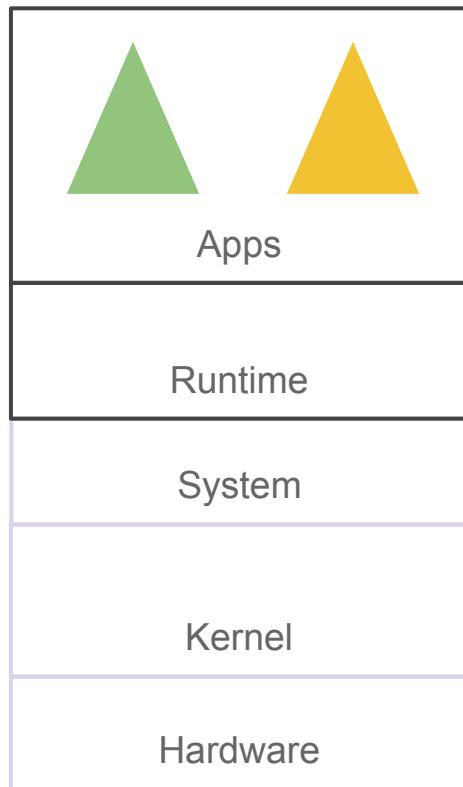


Cloud Native - “un poco de arqueología ...”



Cloud Native - un poco de arqueología ...

THE Monolith

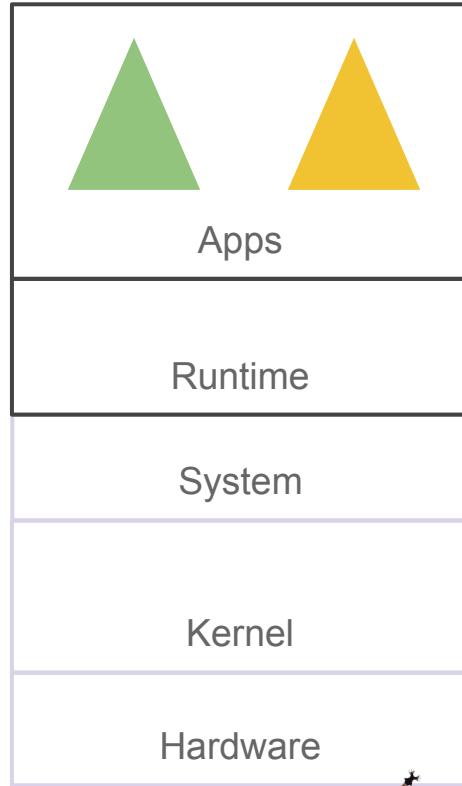


| | |
|---------------|---|
| platform | METAL  |
| orchestration | imperative |
| tools | <i>human</i> , scripts, ansible, chef |

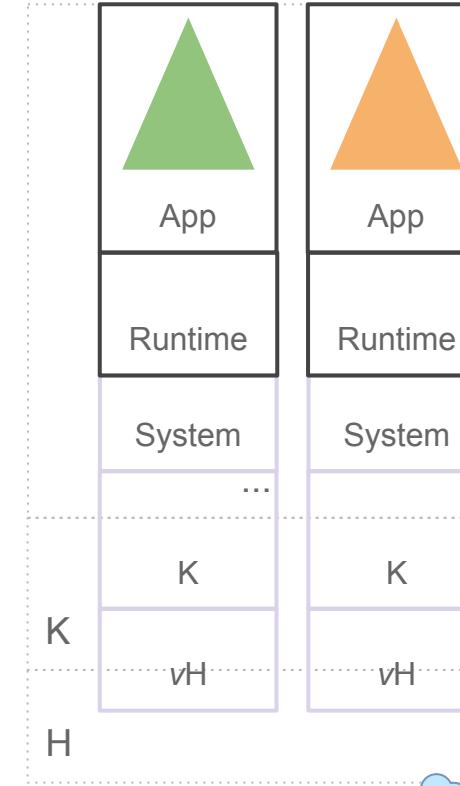
Cloud Native - más cerca en el espacio-tiempo



THE Monolith



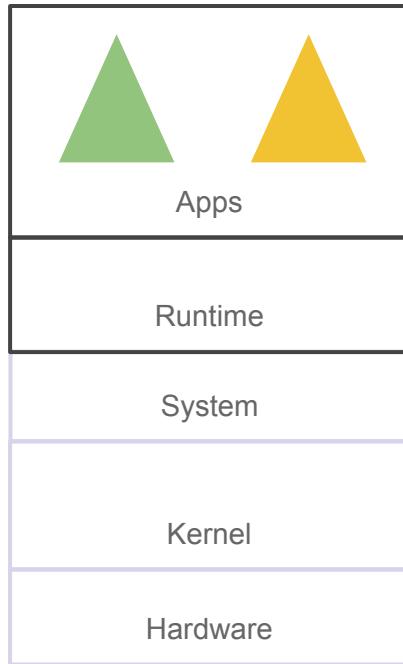
the *v*Monoliths



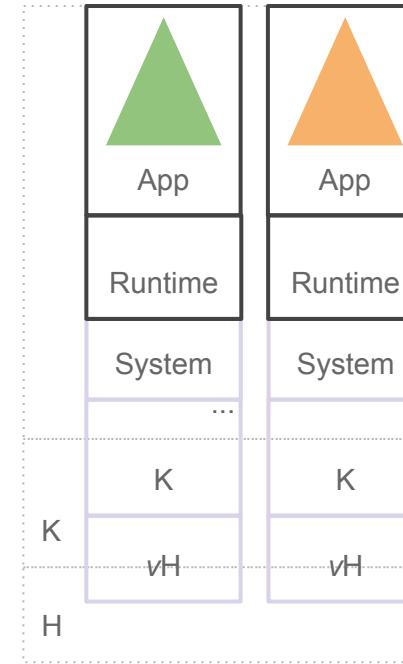
| | | | |
|---------------|---------------------------------------|---|------------------------|
| platform | METAL |  | Cloud <i>Classic</i> ® |
| orchestration | imperative | | + = ~declarative |
| tools | <i>human</i> , scripts, ansible, chef | | + = ~terraform |

Cloud Native - desengrasándonos del sistema

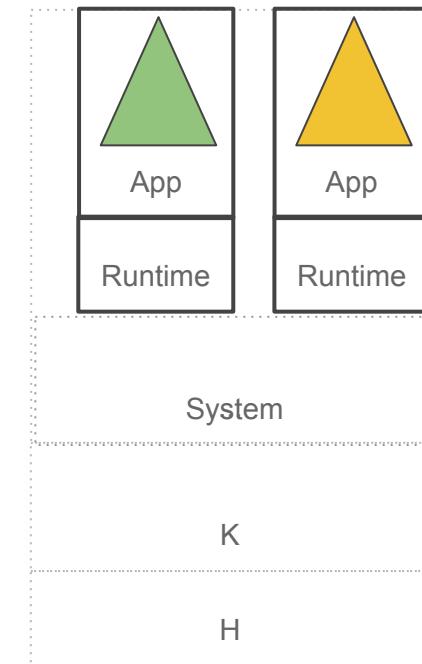
THE Monolith



The VMonoliths



Containers



platform

METAL 

Cloud Classic® 

Cloud Native 

orchestratio
n

imperative

+ = ~declarative

declarative

tools

human, scripts,
ansible, chef

+ =
~terraform

Kubernetes, Docker
SWarm

Tecnologías que empoderan a las empresas para construir aplicaciones escalables en ambientes dinámicos de cloud.

Estas **Tecnologías** permiten crear:

- Sistemas con bajo acomplamiento
- Resistentes[Resilientes]
- Gestión[ables]
- Observables

Combinadas con automatización robusta permite a los ingenieros hacer cambios de alto impacto con frecuencia y mínimo esfuerzo.

¿ Cómo sería ? ...

Beyond 12FactorApps

1. *One codebase, one application*
2. **API first**
3. *Dependency management*
4. *Design, build, release, and run*
5. *Configuration, credentials, and code*
6. *Logs*
7. *Disposability*
8. *Backing services*
9. *Environment parity*
10. *Administrative processes*
11. *Port binding*
12. *Stateless processes*
13. *Concurrency*
14. **Telemetry**
15. **Authentication and authorization**

Containers Docker

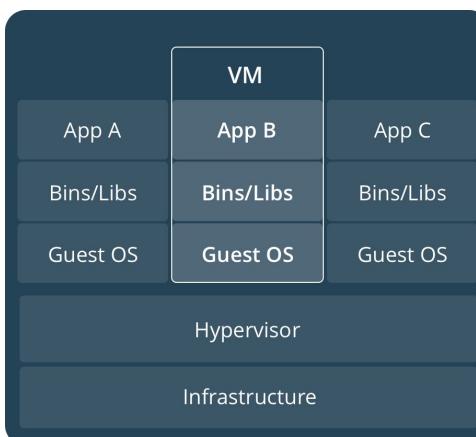
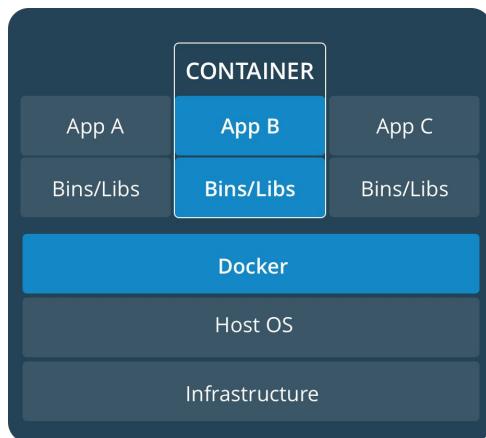
Docker: Conceptos

Una plataforma para:

“desarrollar, desplegar y ejecutar aplicaciones con containers”

Una “**imagen**” es un paquete ejecutable que contiene todo lo necesario para ejecutar una aplicación [código, librerías de ejecución, ambiente y configuración]

Un “**container**” es una instancia de ejecución de una imagen.



cURL un amigo ... y el peluquero jq

```
sudo apt install jq pv
```

client URL, transferencias de recursos especificados como URLs usando pro

MUY usado como CLI/LIB para APIs REST

[Otro muy usado con GUI: [Postman](#)]

jq: “sed” para JSON

Ejemplos:

```
curl -s https://www.creditos.com.ar/wp-json/wp/v2/posts
```



Despeinado

```
curl -s https://www.creditos.com.ar/wp-json/wp/v2/posts | jq . | less
```

Peinado

```
curl -s https://www.creditos.com.ar/wp-json/wp/v2/posts | jq .[]."link"|less
```

Pá la foto...

Perlitas..

```
curl -s http://wttr.in/MENDOZA
```

```
curl -s http://artscene.textfiles.com/vt100/movglobe.vt | pv -q -L 9600
```

```
curl -s http://artscene.textfiles.com/vt100/firework.vt | pv -q -L 4800
```

Mas -> <http://artscene.textfiles.com/vt100/>

Docker: CLI

```
## Ejecutamos una imagen de ejemplo
docker run hello-world
[--name asigna nombre (ps,kill,exec), -d Daemon mode]
docker run --name websrv -d nginx
## Ejecutamos un comando dentro de un container en ejecución
docker exec -it websrv bash
## Listamos las imágenes disponibles localmente
docker image ls
## Listamos los containers
docker container ls
## Listamos los containers en ejecución
docker ps
## Detenemos un container en ejecución
docker stop <container_id|name>
## Matamos un container en ejecución
docker kill <container_id|name>
## Eliminamos un container [apagado]
docker rm <container_id|name>
## Ejecutamos una imagen de la distro Alpine en modo interactivo
docker run -it alpine
```

Docker: Dockerfile

Un Dockerfile define que va dentro del ambiente de un container.

| | |
|--|--|
| FROM alpine:3.7 | FROM: Origen de la imagen Base |
| LABEL maintainer=" cloud@um.edu.ar " | LABEL: Adiciona Metadata |
| ENV TERM xterm | ENV: Setea variables de entorno |
| RUN apk update | RUN: Ejecuta en un shell en la imagen |
| RUN apk add python py2-pip uwsgi uwsgi-python | |
| RUN mkdir /opt | |
| RUN mkdir /opt/validate-email | |
| WORKDIR /opt/validate-email | WORKDIR: Setea el directorio por default |
| COPY src/ /opt/validate-email | COPY: Copia a la imagen |
| RUN RUN pip install --upgrade pip | |
| RUN pip install -r requirements.txt | |
| USER 1000:1000 | USER: Setea el user para RUN,CMD,ENTRYPOINT |
| CMD uwsgi --ini /opt/validate-email/validate-email.ini | CMD: Ejecución por defecto, solo 1 |
| EXPOSE 5000 | EXPOSE: Publica un puerto del container |

Docker: Construyendo nuestra app validate-email

```
## Actualizamos el repo
git clone https://github.com/umcloud/clases-devops.git
## Ingresamos a las apps en el repo, c03-docker, práctico 01 [p01]
cd clases-devops/c03-docker/p01/validate-email/
docker build -t validate-email:p01 .
## Ejecutamos la app
docker run -d --name validate-email-p01 -p 80:5000 -t validate-email:p01
## Probamos
## Modo validación sintaxis
curl http://56ca8f8738ecc03cc4b2859db342e243@localhost/diego.navarrow@um.edu.ar/0
{"validate": true}
## Modo validación sintaxis + dominio existente
curl http://56ca8f8738ecc03cc4b2859db342e243@localhost/diego.navarrow@um.edu.ar/1
{"validate": true}
## Modo validación sintaxis + dominio existente + casilla existente
curl http://56ca8f8738ecc03cc4b2859db342e243@localhost/diego.navarrow@um.edu.ar/2
{"validate": false}
## Matamos el container
docker kill validate-email-p01
```

#Tip solo para "habitantes de UM-Cloud" - Sirve para estudiar MSS
iptables -I FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --set-mss 800

Docker: Comunicándonos con el “Entorno”



```
## Ingresamos a la app en el repo, c03, práctico 02,  
cd c03-docker/p02/validate-email  
  
## Hacemos el build e intentamos ejecutarlas  
docker build -t validate-email:p02 .  
  
## Intentamos Ejecutar la app  
docker run -d --name validate-email-p02 -p 80:5000 -t validate-email:p02  
  
## ¿que paso?  
docker logs validate-email-p02  
docker rm validate-email-p02  
  
## Intentamos Ejecutar la app con Su variable de Entorno  
docker run -d --name validate-email-p02 -p 80:5000 --env API_KEY=1234 -t validate-email:p02  
  
## Probamos  
  
curl http://1234@localhost/diego.navarro@um.edu.ar/2  
{"validate": true}
```

Docker: Publicando en una registry

```
## Nos Registramos en docker hub https://hub.docker.com/signup/
```

```
## Creamos un repositorio para la aplicación
```

Create repository

Namespace
dnavarrow

Repository Name *
validate-email

```
## Tomamos las credenciales para loguearnos en la registry
```

```
docker login
```

```
## Taggeamos las imágenes y la publicamos
```

```
#Docker Hub
```

```
docker tag validate-email:p01 TU_USUARIO_DOCKERHUB/validate-email:p01
```

```
docker push TU_USUARIO_DOCKERHUB/validate-email:p01
```

 dnavarrow / validate-email

Description
Validate Email Microservice 

Last pushed: a minute ago

Docker commands

To push a new tag to this repository,

```
docker push dnavarrow/validate-email:tagname
```

[Public View](#)

Tags

This repository contains 2 tag(s).

| Tag | OS | Type | Pulled | Pushed |
|-----|---|-------|--------|---------------|
| p02 |  | Image | --- | 2 minutes ago |
| p01 |  | Image | --- | 3 minutes ago |

[See all](#) [Go to Advanced Image Management](#)

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more about automated builds](#).

[Upgrade](#)

Cloud Native Kubernetes

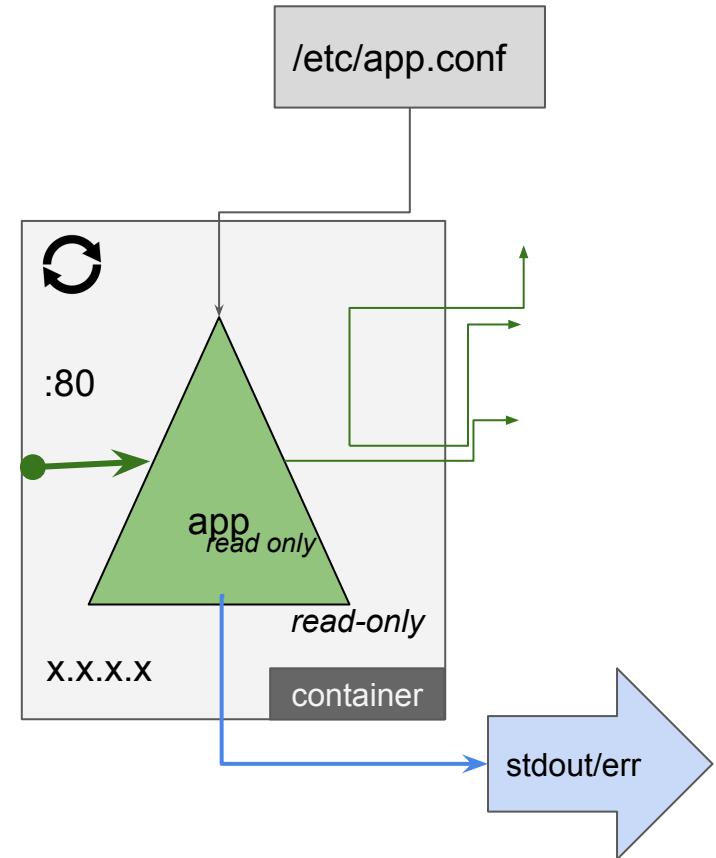
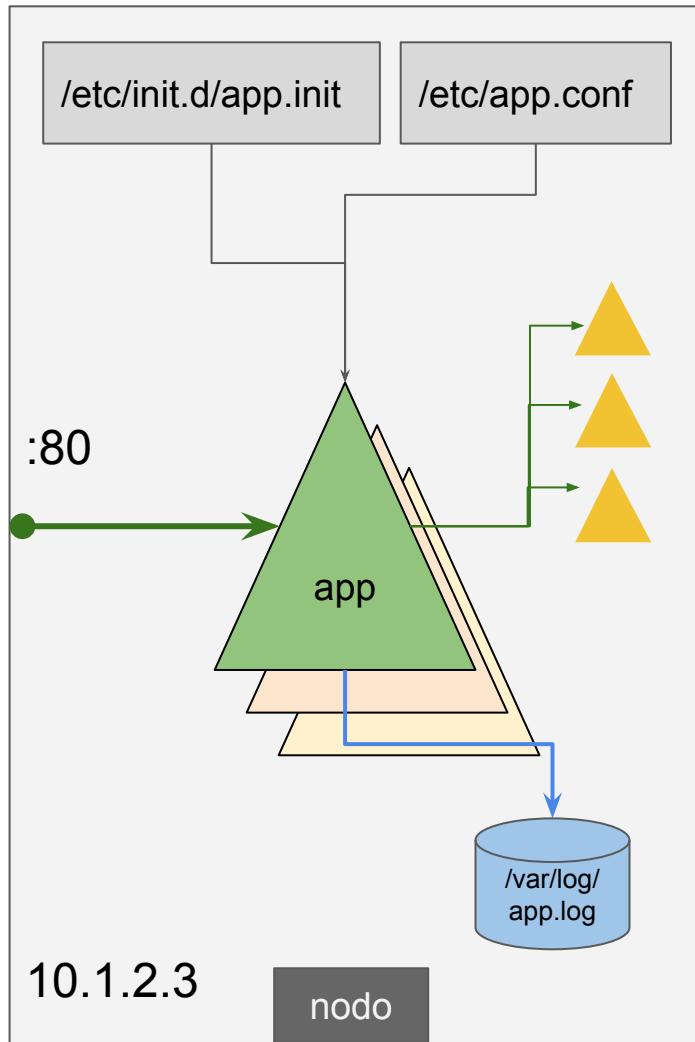
Infraestructura basada en containers



Un Infraestructura basada en containers requiere de una plataforma que provea:

- **Scheduling:**
 - Decida dónde deberían ejecutar los containers
- **Lifecycle and health:**
 - Mantenga los containers corriendo a pesar de los fallos
- **Scaling:**
 - que haga que un conjunto de containers se estire o contraiga
- **Naming and discovery:**
 - Encuentre dónde están los containers
- **Load Balancing:**
 - distribuya tráfico través de un conjunto de containers
- **Storage volumes:**
 - Provea datos a los containers (stateful vs stateless)
- **Logging and monitoring:**
 - Registre qué pasa con los containers
- **Debugging and introspection:**
 - Ingresar/Unirse a un container
- **Identity and authorization:**
 - Controlar quién puede hacer que

Cloud Native - del monolito al nativo



Cloud Native - entorno de un container



Yo controlo tu ciclo de vida, recordá que sos *mortal* --CRI

Esta vez te toca esta IP --CNI

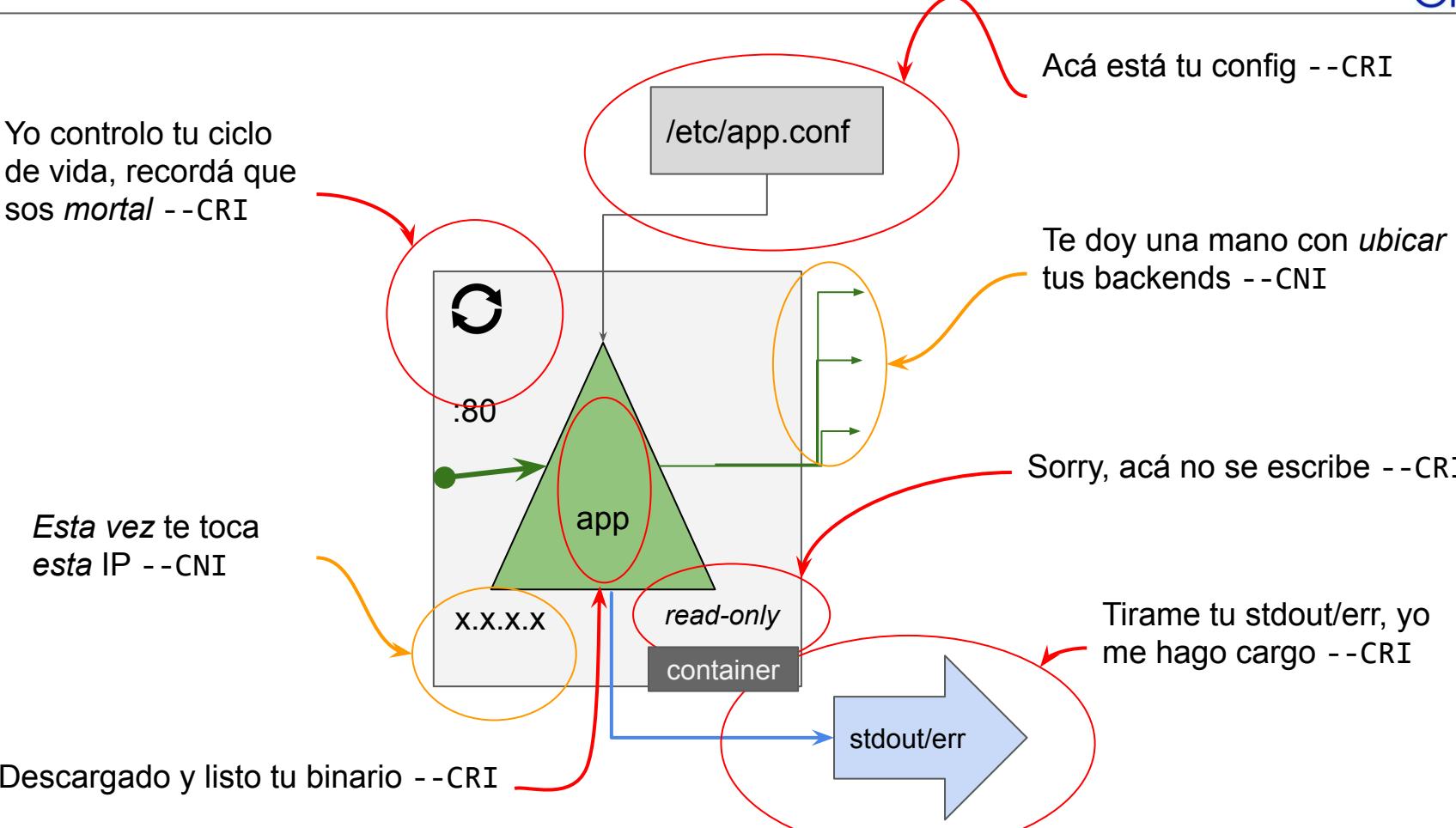
Descargado y listo tu binario --CRI

Acá está tu config --CRI

Te doy una mano con *ubicar* tus backends --CNI

Sorry, acá no se escribe --CRI

Tirame tu stdout/err, yo me hago cargo --CRI



CRI: Container Runtime Interface

CNI: Container Network Interface

CSI: Container Storage Interface

nodo

Kubernetes para tod@s 1de2



- <https://kubernetes.io/>

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.

- Término “Kubernetes”:
 - de la palabra Griega "Timonel":
 - también es la raíz de las palabras "gobernador" y "cibernética"
- API de workloads declarativo
- Plataforma para automatizar despliegue, escalado y operación
- 100% Open Source (Apache license), escrito en Golang
 - Basado en la experiencia de Google con sus cluster-managers internos (Borg, Omega)
 - Kubernetes v1.0: liberada el 21/julio/2015
- *In short:* Orquestador de containers

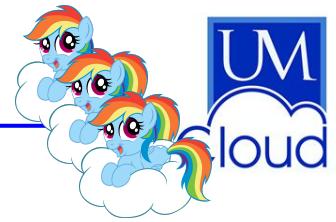
Kubernetes para todos 2de2



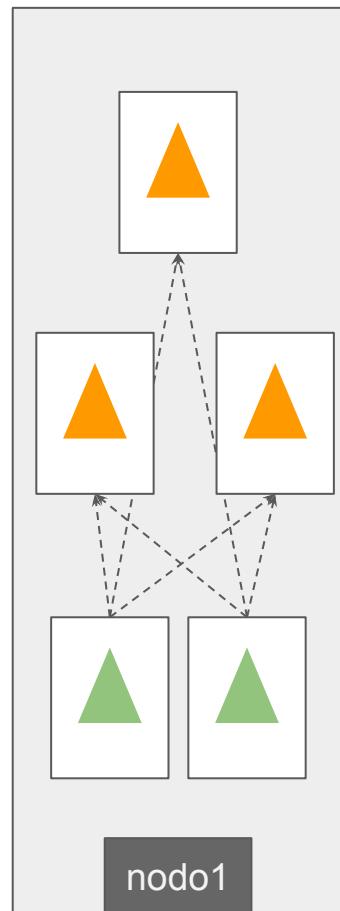
- "Containers cluster orchestrator":
 - Declarative Spec
 - Componentes:
 - CRI: manejo del container runtime en c/nodo
 - CNI: conectividad intra, extra cluster, descubrimiento de "endpoints" (frontend -> backend), "firewall" as a service, "virtualhost" as a service
 - CSI: provisión y *lifecycle* de almacenamiento persistente
 - Multiplataforma:
 - [GKE](#): Google Kubernetes Engine (Google)
 - [EKS](#): Elastic Kubernetes Service (AWS)
 - [AKS](#): Azure Kubernetes Service (Microsoft)
 - CCE: Cloud Container Engine (*Huawei Cloud*)
 - D/I/Y:
 - dev: [k3d](#), [kind](#), [minikube](#) (p/dev)
 - prod: [rke](#), [kubespray](#), [kubeadm](#)
 - ... desde Raspberry-PI hasta 5000 nodos
- Lingua franca! para *Cloud Native Workloads*
 - spec: YAML, JSON
 - **repo** (ejemplo):
 - ./src/app/
 - ./src/deploy/
- Complejo para comenzar a aprender
 - "Kubernetes dificulta las cosas simples, y hace posible las (muy) complejas"

2x => 3x
mi_stack.yaml

Cloud Native - a reproducirse !



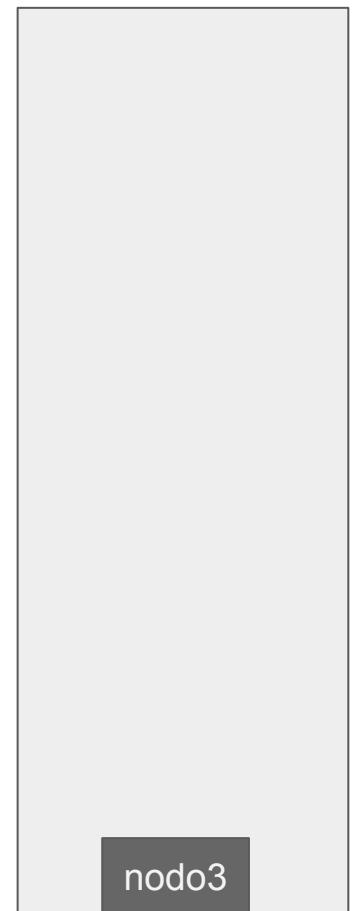
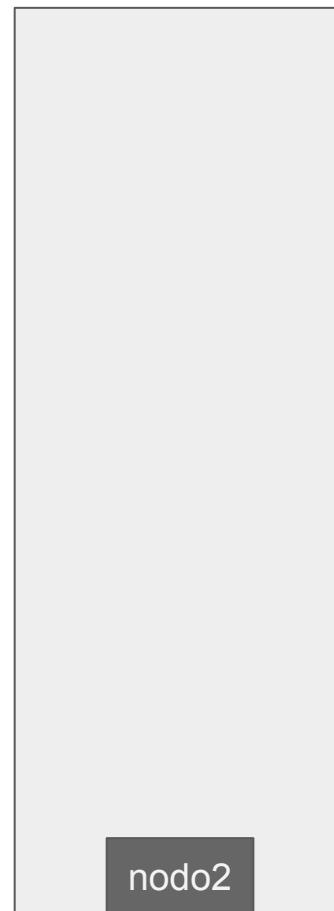
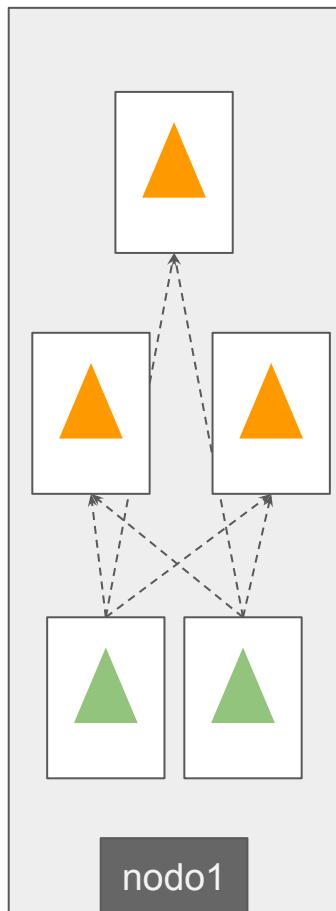
2x => 3x



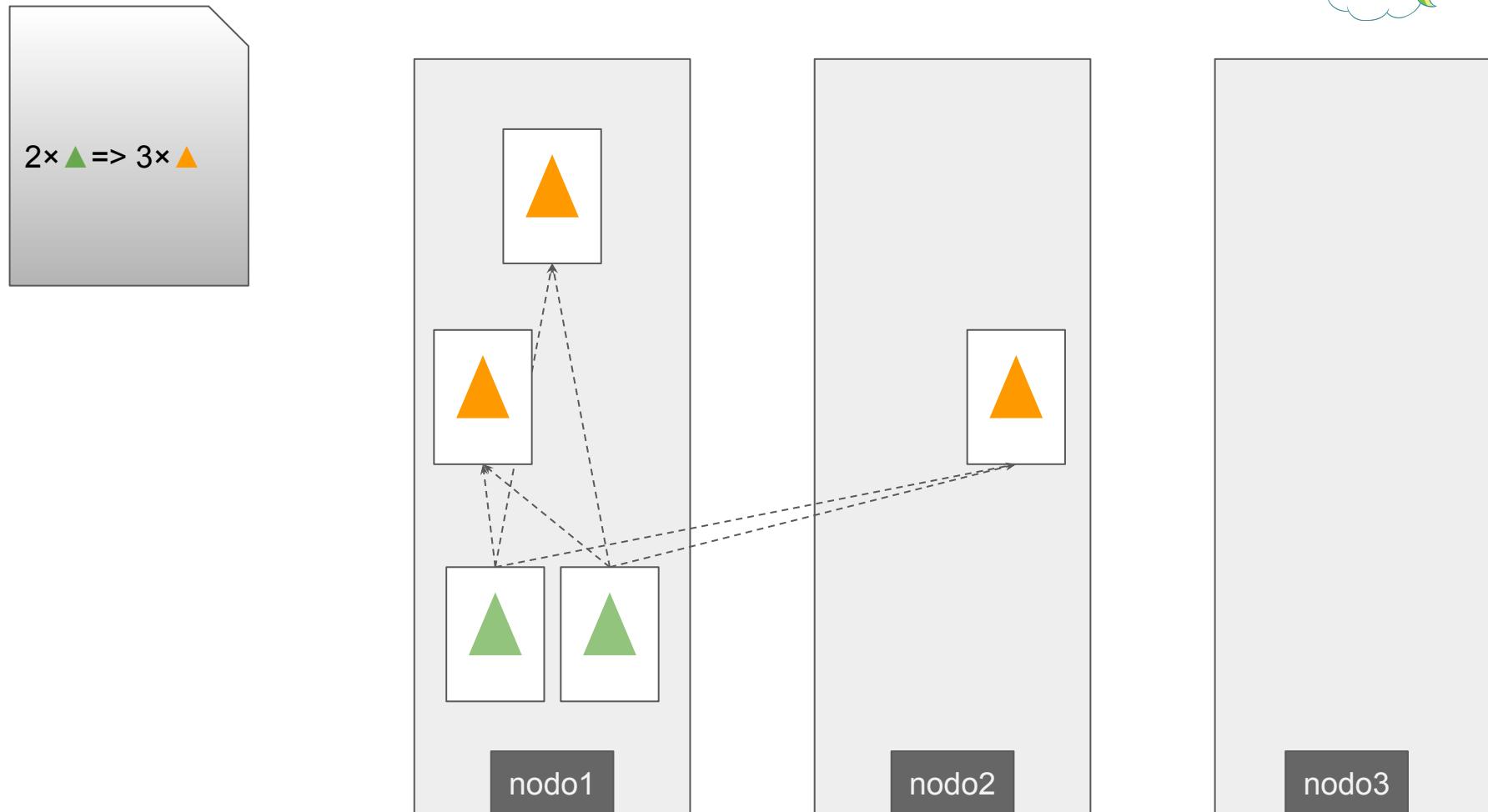
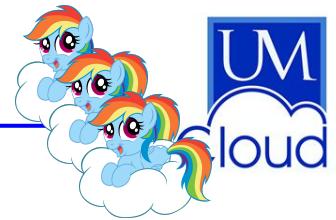
Cloud Native - a reproducirse !



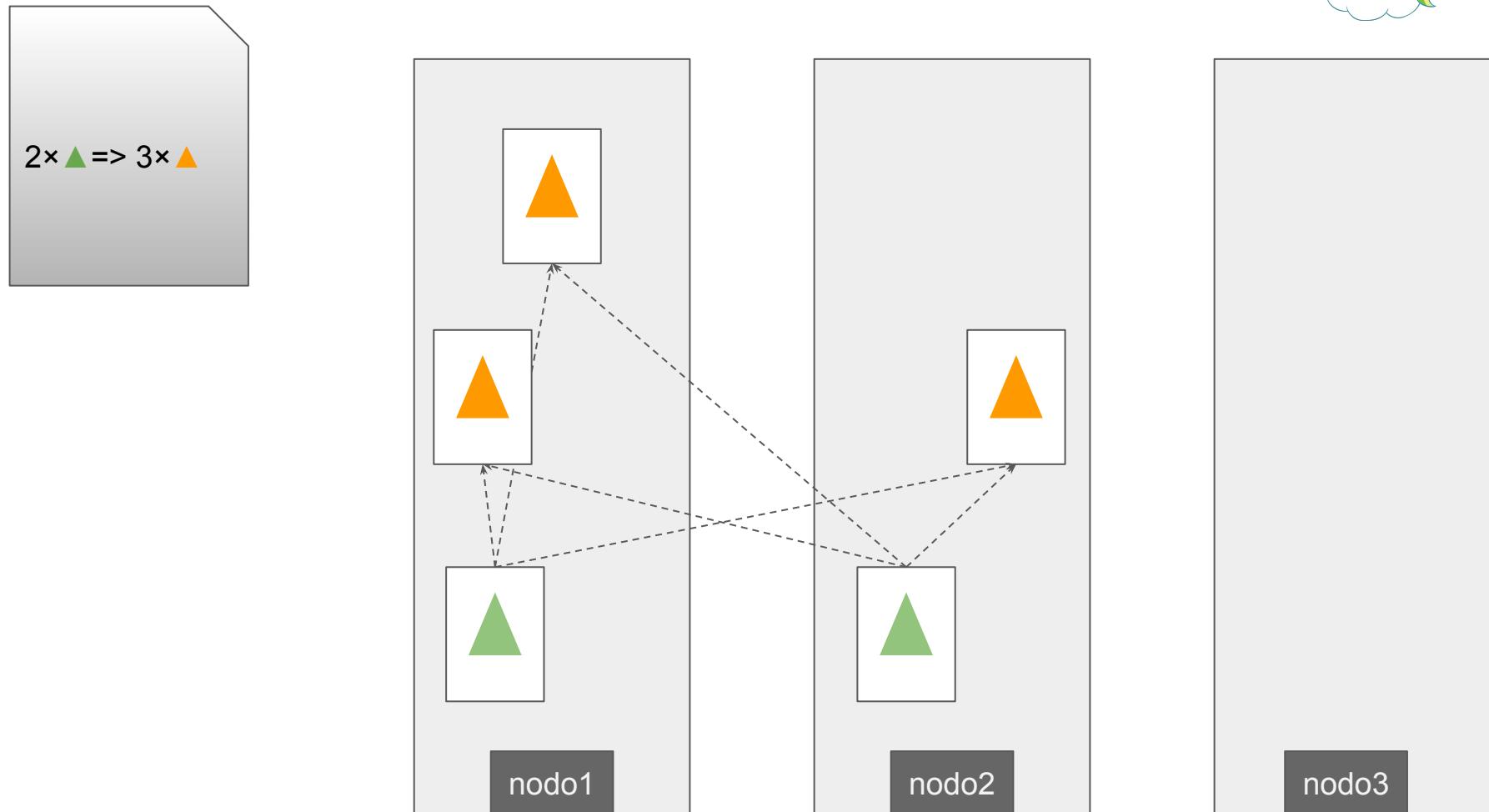
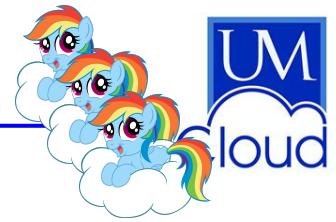
2× ▲ => 3× ▲



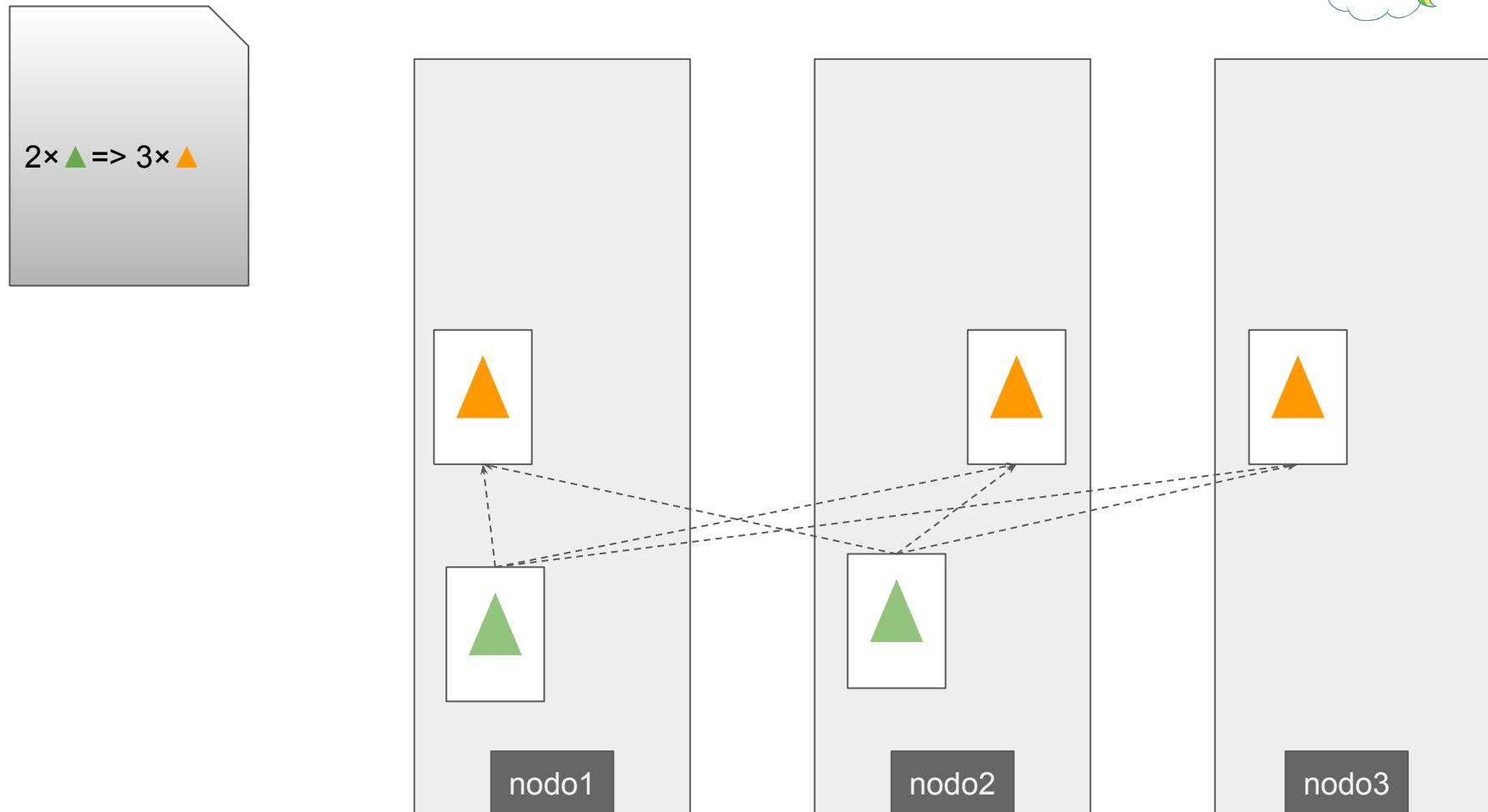
Cloud Native - a reproducirse !



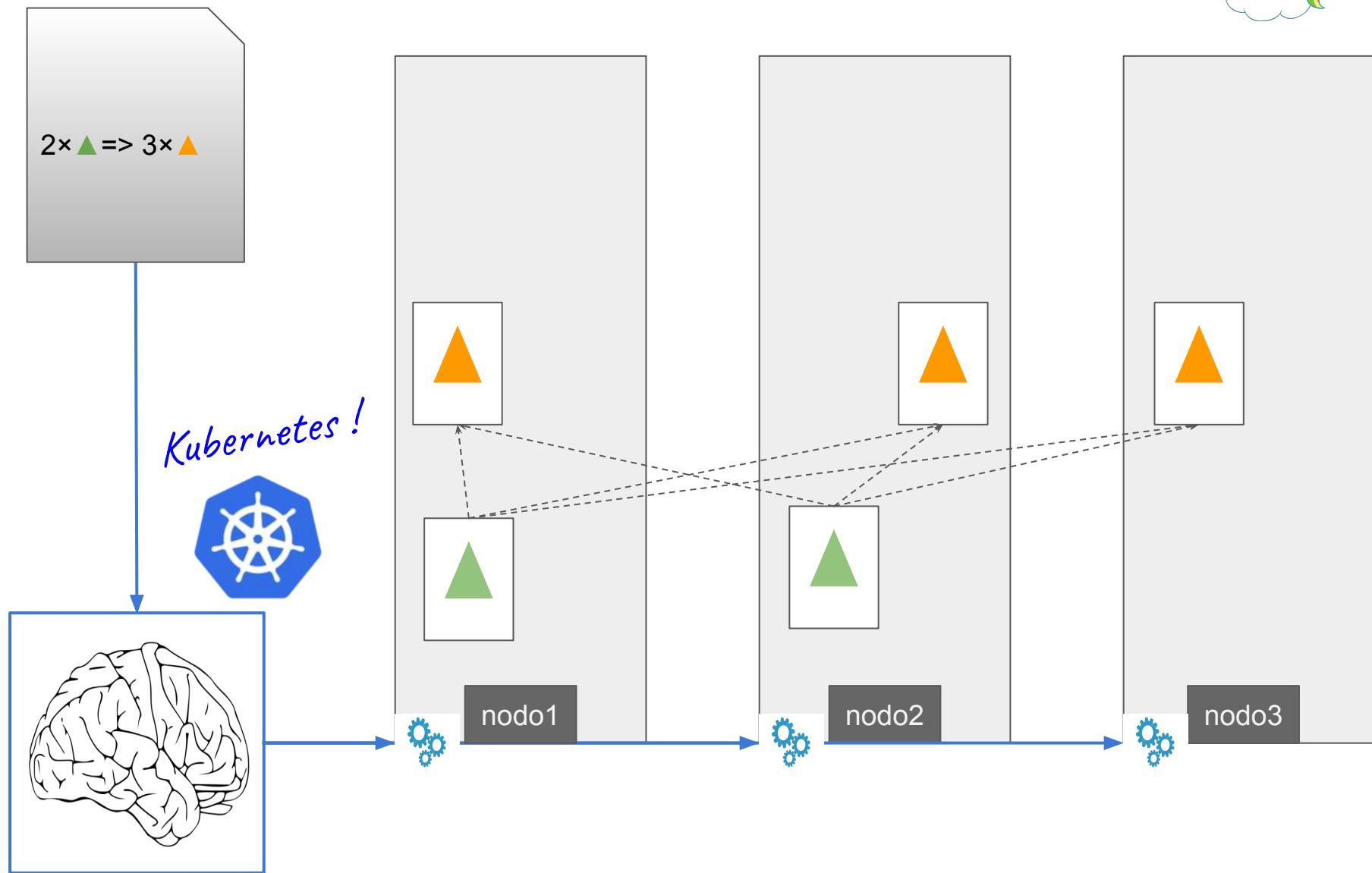
Cloud Native - a reproducirse !



Cloud Native - a reproducirse !

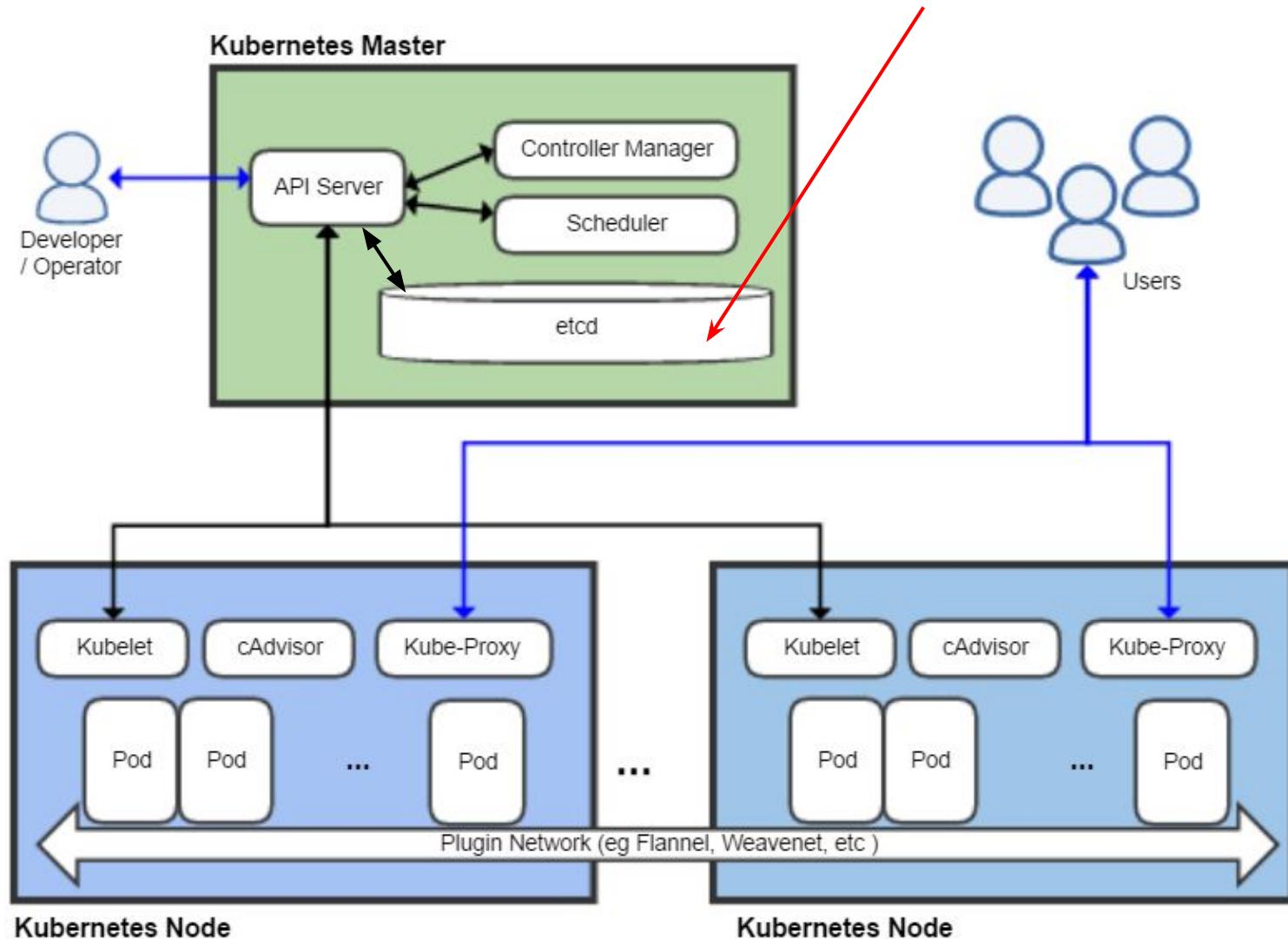


Cloud Native - a reproducirse !



Kubernetes - Arquitectura

Único componente stateful (*k-v storage*)



Kubernetes: primeros pasos -> kubectl

- Loggearse a <https://rancher.kube.um.edu.ar/dashboard/> usando tu cuenta de <https://github.com/> (como miembro de `umcloud-external/students-2023`)
 - Crear una API KEY desde <https://rancher.kube.um.edu.ar/dashboard/account/create-key> ->
 - Scope: **No scope**
 - Expire: **Never**
 - Guardar el `Bearer Token` que muestra “**token-xxxxx:abc...xyz**”
- Crear una VM usando la imagen: **um-kube-tools** (m1.small)
- Loggearse en la VM, ejecutar lo que muestra el comando:
um-help
 - **kube-setup.sh BEARER_TOKEN USUARIO** # USUARIO: por ejemplo tu nick en slack
Select a Project:3 # cluster-01 c-.....:p-xxxxx Default
 - **kube-create-ns.sh NAMESPACE** # NAMESPACE: *USUARIO-dev*

Explorando `kubectl`

```
kubectl version --short
```

```
kubectl get componentstatus
```

```
kubectl get nodes
```

```
kubectl auth can-i delete node
```

```
kubectl auth can-i list nodes
```

Kubernetes object: Pod

- Es el objeto elemental de *workload* de Kubernetes
 - unidad básica de "scheduling"
 - todos los otros objetos se construyen "on-top" de pods
- Puede estar formado por uno o más containers
 - típicamente uno solo
- Tiene una dirección IP única dentro del cluster (1:1)
- Son *mortales* por diseño (*Cloud-Native*)
- Pueden tener *attached* recursos tales como (*12factor*):
 - configmaps
 - secrets
 - volumes
- Probar:
kubectl explain pod

Kubernetes object: Pod - *hands-on!*

```
# Creamos un pod simple con image=nginx  
kubectl run --restart=Never --image=nginx mipod
```

```
# Ver estado, y detalles  
kubectl get pod  
kubectl describe pod mipod
```

```
# Ver estado++  
kubectl get pod -owide  
kubectl get pod -owide --show-labels
```

```
# Ver el universo ... (de pods corriendo)  
kubectl get pod --all-namespaces -owide
```

Kubernetes object: Pod - *hands-on++!*

Entrñas del pod:

```
kubectl get pod mipod -oyaml | less
```

↵ ¿ Cuáles campos especificamos nosotros ?, comparar con ↵
kubectl run --restart=Never --image=nginx mipod --dry-run=client
-oyaml

Entremos al pod:

```
kubectl exec -it mipod -- /bin/bash
```

root pod ! :P . Cambiemos su imagen:

```
kubectl set image pod/mipod mipod=bitnami/nginx
```

```
kubectl logs mipod
```

```
kubectl exec -it mipod -- /bin/bash
```

Chau pod:

```
kubectl delete pod mipod
```

Kubernetes object: namespace

- Permite separar "contextos", *think-of: "resource folders"*
- Facilita definir bordes de:
 - seguridad
 - planificación (schedule-ability)
 - valores x default, tope de recursos
- Casos de uso:
 - dev vs staging vs production
 - namespace por usuario (nuestro caso) o proyecto
 - namespaces "del sistema": kube-system ...
- Probar:

```
kubectl get ns
```

```
kube-create-ns USUARIO-prod
```

```
kubens USUARIO-dev # volver al previo NS
```

```
kubectl delete pod --namespace=kube-system --all --dry-run=server
```

- **Deployments**
 - arreando pods ...
- **Services**
 - cómo llego a mis pods ?
- **ConfigMaps**
 - cómo configuro mis containers (dentro de los pods) ?
- **Secrets**
 - cómo proveo configuración de secretos
- **StatefulSets, DaemonSets**
 - deployments + volúmenes persistentes
- **Ingress**
 - cómo llego a mis pods via *http virtualhost* ?

p01- Kubernetes object: Deployment

IMPORTANTE: Loggearse a la VM con ->

```
ssh -L 8080:localhost:8080 -L 8001:localhost:8081 -A ubuntu@IP
```

- Actualizamos el repo

```
git clone https://github.com/umcloud/clases-devops
cd clases-devops
git pull
cd c04-kube-deploy_svc_cm/p01
```



Deployment

ReplicaSet

- Creamos un deploy, modo declarativo via YAML

```
kubectl create deployment --image=bitnami/nginx:1.24.0 web --dry-run=client -oyaml > web.deploy.yaml
kubectl apply -f web.deploy.yaml
kubectl get deploy,rs,pod                                # ¿ Y esos nombres de pods, algún patrón ?
```

- Update: 1->3 replicas, y última versión de image de nginx

```
kubectl scale deployment web --replicas=3          # <-- ¿ Estaría bien hacer esto ?
sed -i -e /nginx:/s/1.24.0/1.25.0/ -e /replicas:/s/1/3/ web.deploy.yaml
kubectl apply -f web.deploy.yaml
kubectl rollout status -f web.deploy.yaml           # igual a: ... rollout status web
kubectl rollout history -f web.deploy.yaml         # igual a: ... rollout history web
kubectl get deploy,rs,pod                           # ¿ Qué cambió ?
```

- Rollback/rollforward via `kubectl rollout undo ...`

p01- Kubernetes object: Deployment

- Objeto de más "alto" nivel, controla *Pods* via *ReplicaSets*
- Tiene estrategia de rollout:

```
kubectl get deployment web -oyaml | grep -A3 strategy:
```

- maxSurge: cuánto se puede "inflar" (en pods) el deploy durante rollout
- maxUnavailable: cuántos pods pueden estar *down* durante rollout

- Objeto intermedio: *ReplicaSet*
 - observar el *output* del último `kubectl get deploy,rs,pod`
- Exploraremos un poco:

```
kubectl get pod  
kubectl get pod web-xxxxxxxxx-yyyyy  
kubectl get pod -l app=web  
kubectl logs web-xxxxxxxxx-yyyyy # kubectl logs deploy/web  
kubectl exec -it web-xxxxxxxxx-yyyyy -- /bin/bash # kubectl exec -it deploy/web -- /bin/bash
```

- Ahora ... ¿ Cómo lo accedemos ?

```
kubectl get pod  
kubectl port-forward deploy/web 8080:8080 &  
curl http://localhost:8080/ # o via browser  
kill %1 # mato el port-forward corriendo en background
```

- Todo ésto ^^^ no sirve para production, ¿ Por qué ?

p02- Kubernetes object: Service

- ¿ Y cómo exponemos el *Deployment*? via un *Service*:

```
cd ..\p02
```

```
kubectl expose deploy web --port=80 --target-port=8080 --dry-run=client -oyaml >web.svc.yaml
```

```
less web.svc.yaml # Notar: ports y selector
```

```
kubectl apply -f web.svc.yaml
```

- ¿ Qué ocurrió ?

```
kubectl get svc,ep,pod -owide
```

- notar las IPs en juego ...

- Accediendo el servicio desde dentro del cluster

```
kubectl run -it --rm --restart=Never --image=alpine alp-shell # pod efímero
```

```
/ # apk add curl
```

```
/ # curl -v http://web
```

```
/ # ping web      # criKcriK ...
```



p02- Kubernetes object: Service

- Son efectivamente *loadbalancers internos* de Kubernetes
 - provistos por la plataforma, via CNI
- Típicamente a cargo del [kube-proxy](#) que corre en cada nodo
- Hay varios tipos, de acuerdo a su *reachability*:
 - **ClusterIP** (default)
 - accesible desde **dentro** del cluster (nodos y pods)
 - Típicamente usado para conectar micro-services internos, o necesario para →
 - **NodePort**
 - ClusterIP + **ports dedicados en cada nodo worker**
 - Por ejemplo para apuntar un *loadbalancer* externo a todos los nodos, mismo port
 - **LoadBalancer**
 - NodePort + creación automática de LB provisto por el Cloud Provider

p03- Kubernetes object: ConfigMap

- Objetivo: Ponerle data al servicio web via configmap (alias cm), el cual contiene un conjunto de key=values, ej:

```
kubectl create cm app-cfg --from-literal=foo=uno --from-literal=bar=dos --dry-run=client -oyaml
```

- Construir un configmap con data de un directorio

```
cd ../p03  
mkdir -p assets  
echo "hola mundo desde $USER" > assets/index.html  
kubectl create cm --from-file=assets/ web-assets --dry-run=client -oyaml>web-assets.cm.yaml  
kubectl apply -f web-assets.cm.yaml  
kubectl get cm  
kubectl get cm web-assets -oyaml
```

- Web (re)deploy montando este cm, usando el yaml provisto ([web.deploy.yaml](#)):

```
diff -u ../p01/web.deploy.yaml web.deploy.yaml  
kubectl diff -f web.deploy.yaml  
kubectl apply -f web.deploy.yaml
```

- Accediendo al servicio desde tu laptop ([README.md](#))

```
kubectl proxy &           # Reemplazar YOUR_NAMESPACE debajo ->  
curl http://localhost:8001/api/v1/namespaces/YOUR_NAMESPACE/services/web/proxy/
```

p03- Kubernetes object: ConfigMap

- Es un objeto que contiene un conjunto de **key: value**, en su campo **data**
- Fundamentalmente usado para configurar los parámetros runtime de las aplicaciones
- Se puede "consumir" como:
 - **Archivos** mapeados en en container (key: filename, value: contenido)
 - **Environmental variables**
 - Command line **arguments** \$(ENV_VAR)

p04- Kubernetes object: Secret

- Objetivo: Arrancar un motor de base de datos [usamos https://hub.docker.com/_/mysql/]
cd ..\p04
kubectl apply -f 01-db-simple.deploy.yaml
- ¿Qué ocurrió? puufffff...

```
kubectl get pod -owide  
kubectl logs deploy/db-simple # NOTAR deploy/ en vez del pod
```

Un secreto es un objeto que contiene una pequeña cantidad de datos confidenciales, como una contraseña, un token o una clave. [12f++: no debe quedar información sensitiva dentro de las imágenes/repos] Solución: “volúmenes” tipo secreto.

```
kubectl create secret generic db-secret \  
  --from-literal=db-username=root \  
  --from-literal=db-password=secretisimo \  
  --from-literal=db-name=whois -o yaml \  
  --dry-run=client > 02-db-prod.secret.yaml
```

```
kubectl apply -f 02-db-prod.secret.yaml  
kubectl apply -f 03-db-prod.deploy.yaml
```

- ¿Qué ocurrió?
kubectl get deploy,secret,pod -owide
kubectl exec -it deploy/db-prod -- /bin/bash mysql -uroot -psecretisimo

p04- Kubernetes object: Secret & ConfigMap

- Objetivo: Arrancar un motor de base de datos con su SCHEMA ya creado.

Creamos un ConfigMap con el código sql para crear el SCHEMA de la DB y luego Deployamos

```
kubectl create cm db-sql --from-file=schema.sql -o yaml --dry-run=client >  
04-db-prod-schema.cm.yaml  
kubectl apply -f 04-db-prod-schema.cm.yaml  
kubectl apply -f 05-db-prod-schema.deploy.yaml
```

* Este deploy usa los secrets de 02-db-prod.secret.yaml

- ¿ Qué ocurrió ?

```
kubectl get deploy,secret,pod -owide  
kubectl exec -it deploy/db-prod-schema -- /bin/bash
```

- ¿ Le metemos data ?

```
cat data.sql | kubectl exec deploy/db-prod-schema -it -- mysql -u root -psecretisimo  
whois
```

- ¿ Hay data ?

```
echo 'select cidr,json_extract(whois,"$.network.name") from whois;' | \  
kubectl exec deploy/db-prod-schema -it -- \  
mysql -r -s -u root --password=secretisimo whois
```

- **Deployments**
 - arreando pods ...
- **Services**
 - cómo llego a mis pods ?
- **ConfigMaps**
 - cómo configuro mis containers (dentro de los pods) ?
- **Secrets**
 - cómo proveo configuración de secretos
- **StatefulSets**
 - deployments + volúmenes persistentes
- **Ingress**
 - cómo llego a mis pods via *http virtualhost* ?

p01- Kubernetes object: Deployment + PVC [PV]



PersistentVolume (PV): una porción de almacenamiento en el cluster aprovisionado a tal fin. [un “recurso” como un nodo]

PersistentVolumeClaim (PVC): es un requerimiento de almacenamiento hecho por un usuario del cluster.

StorageClasses (SC): Para ofrecer una variedad de tipos de PVs [tamaño, perf] sin exponer al usuario los detalles de implementación.

- Actualizamos el repo (<https://github.com/umcloud/clases-devops.git>)

```
cd clases-devops  
git pull  
cd c05-kube-pvc_sts_ingress/p01
```

- Objetivo: Arrancar un motor de base de datos con su SCHEMA ya creado y que persista la información

```
# Creamos el pvc  
kubectl apply -f 01-db-prod-pvc.yaml
```

```
# Adicionamos Secret MYSQL_DATABASE, MYSQL_ROOT_PASSWORD Env  
kubectl apply -f 02-db-prod.secret.yaml
```

```
# Usamos ConfigMap para schema+data  
kubectl apply -f 03-db-prod.configmap.yaml
```

```
# Deploy: mysql+initContainer+pvc+ConfigMap  
kubectl apply -f 04-db-prod.deploy.yaml
```

p01- Kubernetes object: Deployment + PVC[PV]



```
$ kubectl get deploy,pvc,pod,cm
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deploy        deployment.apps/db-prod   1/1         1           8m19s

NAME          STATUS    VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS
pvc           persistentvolumeclaim/mysql-data   Bound   pvc-0163cf76-6126-4b66-a6cc-037dfcb23a99   1Gi        RWO          longhorn

NAME          READY   STATUS    RESTARTS   AGE
pod           pod/db-prod-9fccb67cf-4m4qh   1/1       Running   0           8m21s

NAME          DATA   AGE
configmap/db-schema-data   2       8m23s
```

p02- Kubernetes object: StatefulSet (sts)



- **StatefulSet (STS):** gestiona el despliegue y escalado de un conjunto de Pods, provee garantía en el ordenamiento y unicidad de esos Pods, y acoplamiento "ordenado" con los PVCs creados.
- **Objetivo:** Arrancar un motor de base de datos con su SCHEMA ya creado, que persista la información y ofrecerlo como un servicio

```
cd ../p02  
kubectl delete -f ./p01  
kubectl get pvc          # b00M, gone, kaput
```

```
# Creamos Secret MYSQL_DATABASE, MYSQL_ROOT_PASSWORD Env  
kubectl apply -f 01-db-prod.secret.yaml  
# Usamos ConfigMap para schema+data  
kubectl apply -f 02-db-prod.configmap.yaml  
# StatefulSet: svc + mysql + initContainer + pvc + ConfigMap  
kubectl apply -f 03-db-prod.sts.yaml
```

| \$ kubectl get svc,sts,pvc,pod,cm | | | | | | |
|--|--|-----------|----------------|-------------|--|-------------------------------|
| | NAME | TYPE | CLUSTER-IP | EXTERNAL-IP | PORT(S) | AGE |
| SVC | service/db | ClusterIP | 10.43.159.134 | <none> | 3306/TCP | 17s |
| sts | statefulset.apps/db-prod | | | READY | AGE | |
| | | | | 0/1 | 17s | |
| pvc persistentvolumeclaim/mysql-data-db-prod-0 | | | | | | |
| | NAME | | | STATUS | VOLUME | |
| pvc | persistentvolumeclaim/mysql-data-db-prod-0 | | | Bound | pvc-e2b14aa3-e814-48c2-a060-f15958854fa6 | CAPACITY 1Gi ACCESS MODES RWO |
| pod | pod/db-prod-0 | READY 1/1 | STATUS Running | RESTARTS 0 | AGE 18s | |
| configmap/db-schema-data | | | | | | |
| | NAME | | | DATA | AGE | |
| | configmap/db-schema-data | | | 2 | 20s | |

p04- Kubernetes object: Service (cont)

- Recordemos: service es la manera en que Kubernetes expone un set de pods ("endpoints") detrás de *IP:PORT* estables, en forma de "load-balancers" internos
- De lo que hicimos en `../c04-kube-deploy_svc_cm/p01` ->

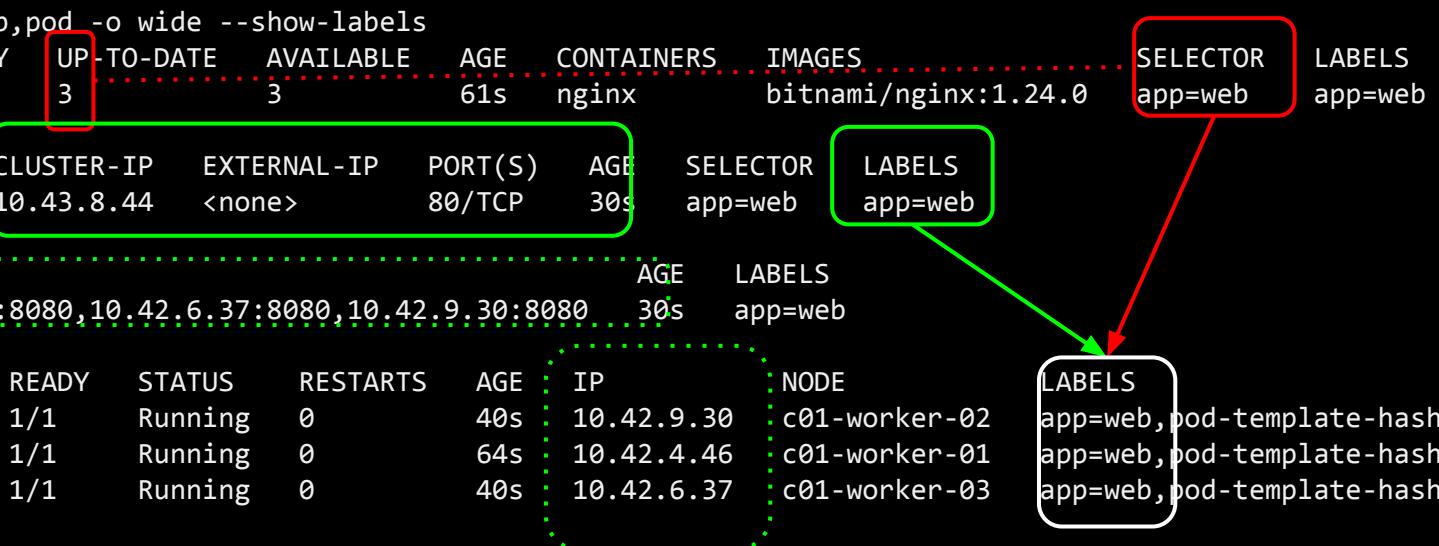
```
$ kubectl get deploy,svc,ep,pod -o wide --show-labels
```

| NAME | READY | UP-TO-DATE | AVAILABLE | AGE | CONTAINERS | IMAGE | SELECTOR | LABELS |
|---------------------|-------|------------|-----------|-----|------------|----------------------|----------|---------|
| deployment.apps/web | 3/3 | 3 | 3 | 61s | nginx | bitnami/nginx:1.24.0 | app=web | app=web |

| NAME | TYPE | CLUSTER-IP | EXTERNAL-IP | PORT(S) | AGE | SELECTOR | LABELS |
|-------------|-----------|------------|-------------|---------|-----|----------|---------|
| service/web | ClusterIP | 10.43.8.44 | <none> | 80/TCP | 30s | app=web | app=web |

| NAME | ENDPOINTS | AGE | LABELS |
|---------------|---|-----|---------|
| endpoints/web | 10.42.4.46:8080,10.42.6.37:8080,10.42.9.30:8080 | 30s | app=web |

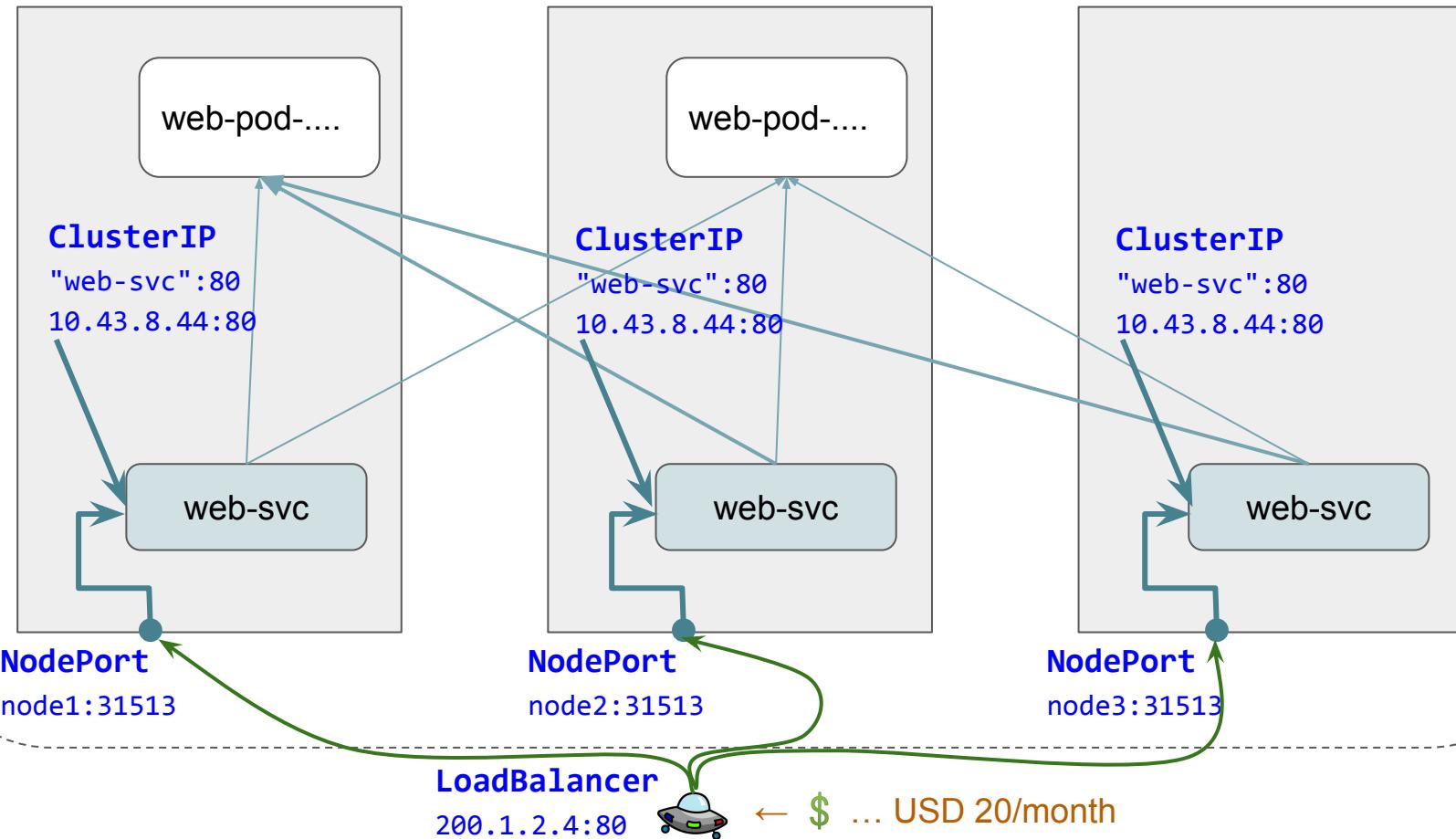
| NAME | READY | STATUS | RESTARTS | AGE | IP | NODE | LABELS |
|--------------------------|-------|---------|----------|-----|------------|---------------|---------------------------------------|
| pod/web-65c4fd5458-5jnbc | 1/1 | Running | 0 | 40s | 10.42.9.30 | c01-worker-02 | app=web, pod-template-hash=65c4fd5458 |
| pod/web-65c4fd5458-hvwzl | 1/1 | Running | 0 | 64s | 10.42.4.46 | c01-worker-01 | app=web, pod-template-hash=65c4fd5458 |
| pod/web-65c4fd5458-w944d | 1/1 | Running | 0 | 40s | 10.42.6.37 | c01-worker-03 | app=web, pod-template-hash=65c4fd5458 |



p04- Kubernetes object: Service types

Hay varios tipos, de acuerdo a su *reachability*:

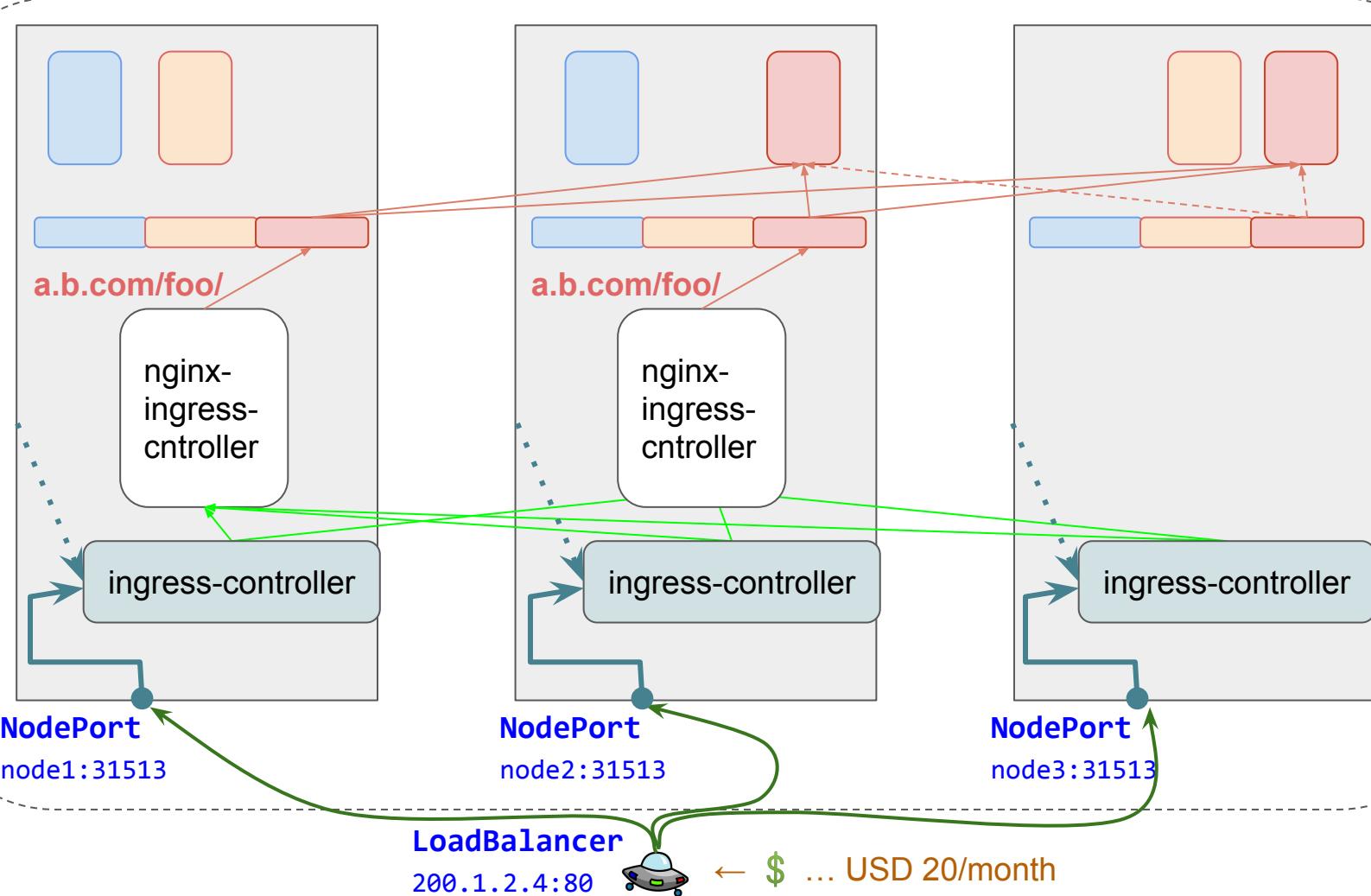
- **ClusterIP** (default): accesible desde dentro del cluster (nodos y pods)
- **NodePort**: ClusterIP + ports dedicados en cada nodo worker
- **LoadBalancer**: NodePort + LB provisto por el Cloud Provider



p04- Kubernetes object: Ingress

- **Ingress** es un recurso de L7:

- http/https reverse proxy
- basado en virtualhost/path



p04- Kubernetes object: Ingress



- **Ingress** es un recurso de L7:
 - http/https reverse proxy
 - basado en virtualhost/path

```
cd ../p04
```

```
vi web-ingress.yaml #modificar key host por url USER-web.my.kube.um.edu.ar  
kubectl apply -f web-ingress.yaml
```

dnavarro-web.my.kube.um.edu.ar

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Google Cloud [Kubernetes Engine]



<https://console.cloud.google.com>

A screenshot of the Google Cloud Console homepage. At the top, there's a navigation bar with the Google Cloud logo, a dropdown menu for the project 'um-cloud', a search bar, and various icons for notifications and help. The main header features a colorful graphic of colored dots and a triangle. Below the header, a welcome message says 'Te damos la bienvenida' (Welcome) with a small cloud icon. It shows the user is working on the 'um-cloud' project (Número de proyecto: 833592655843, ID del proyecto: um-cloud-221804). There are two tabs: 'Panel' (selected) and 'Recomendaciones'. At the bottom of the page, there are four buttons for quick actions: 'Crea una VM', 'Ejecuta una consulta en BigQuery', 'Crea un clúster de GKE', and 'Crea un bucket de almacenamiento'. The 'Acceso rápido' section contains four links: 'API APIs y servicios', 'IAM y administración', 'Facturación', and 'Compute Engine'. A footer at the bottom includes links for 'Política de Privacidad' and 'Condiciones del Servicio'.

Google Cloud: Creamos el cluster

Google Cloud um-cloud Buscar (/) recursos, documentos, productos y más Buscar ☰ 🔍 ⓘ

← Crea un clúster de Autopilot ⚙ CAMBIAR A CLÚSTER ESTÁNDAR APRENDE

1 Aspectos básicos del clúster
Configura los aspectos básicos de tu clúster

2 Redes
Define la comunicación de aplicaciones en el clúster

3 Configuración avanzada
Revisa las opciones adicionales

4 Revisar y crear
Revisa toda la configuración y crea tu clúster

Aspectos básicos del clúster

Especifica un nombre y una región para crear un clúster de Autopilot. Después de crear el clúster, puedes implementar tu carga de trabajo a través de Kubernetes y nosotros nos encargaremos del resto, incluidos los siguientes aspectos:

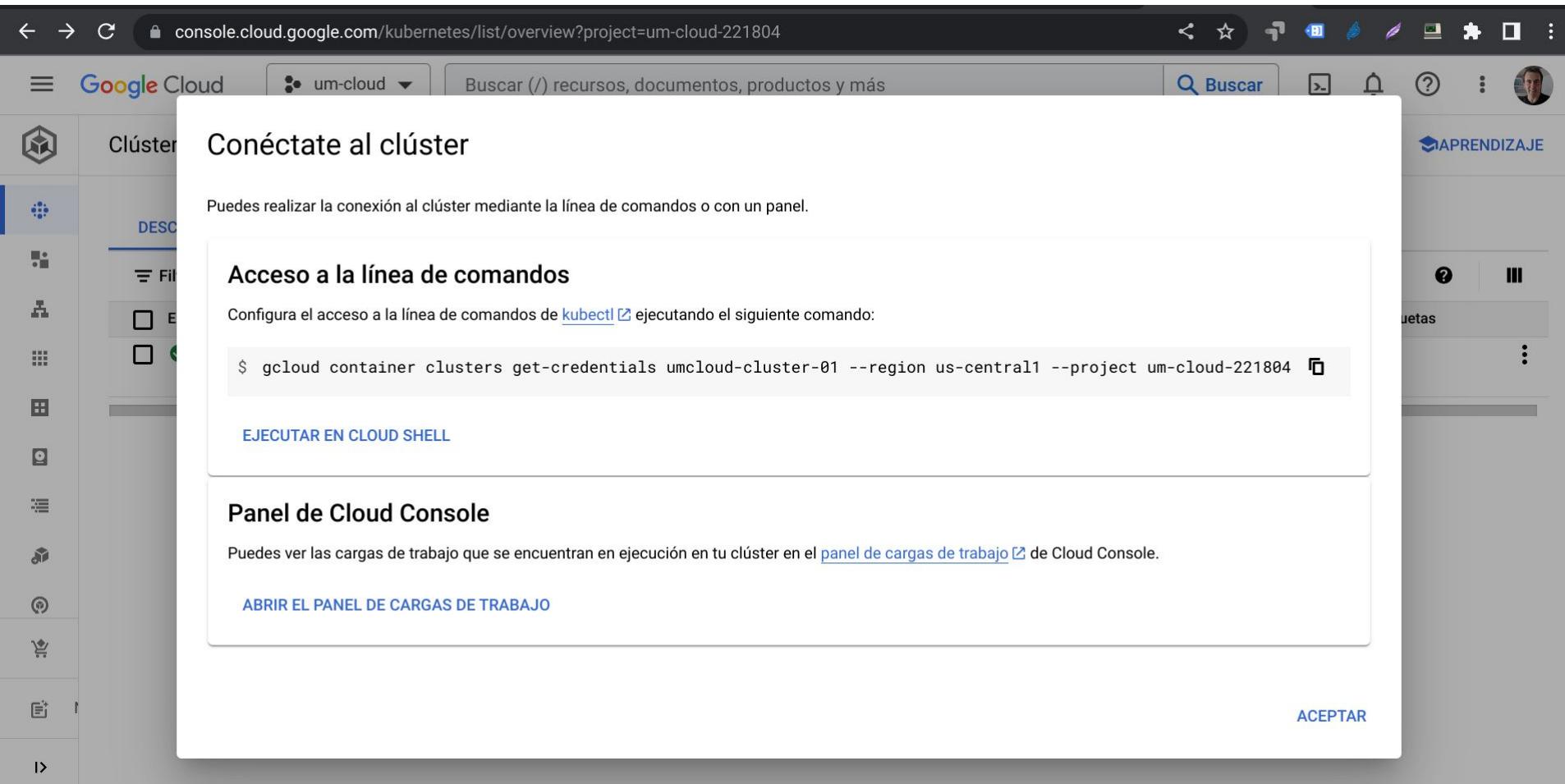
- ✓ **Nodos:** Escalamiento, mantenimiento y aprovisionamiento automático de nodos
- ✓ **Herramientas de redes:** Enrutamiento del tráfico nativo de la VPC para clústeres públicos o privados
- ✓ **Seguridad:** Nodos de GKE protegidos y Workload Identity
- ✓ **Telemetría:** Registro y supervisión de Cloud Operations

Nombre Los nombres de los clústeres deben comenzar con una letra minúscula seguida por un máximo de 39 letras minúsculas, números o guiones. No puede terminar con un guion. No puedes cambiar el nombre del clúster una vez creado.

Región La ubicación regional en la que se encuentran el plano de control y los nodos de tu clúster. No puedes cambiar la región del clúster una vez creada.

SIGUIENTE: REDES **CREAR** **CANCELAR REST o LÍNEA DE COMANDOS** **equivalente** **RESTABLECER CONFIGURACIÓN**

Google Cloud: Conectamos al cluster con cloud shell



The screenshot shows a Google Cloud Console interface with a central modal dialog. The URL in the browser bar is `console.cloud.google.com/kubernetes/list/overview?project=um-cloud-221804`. The modal title is "Conéctate al clúster". It contains instructions: "Puedes realizar la conexión al clúster mediante la línea de comandos o con un panel." Below this, under "Acceso a la línea de comandos", it says "Configura el acceso a la línea de comandos de [kubectl](#) ejecutando el siguiente comando:" followed by a command line snippet: `$ gcloud container clusters get-credentials umcloud-cluster-01 --region us-central1 --project um-cloud-221804`. A blue button labeled "EJECUTAR EN CLOUD SHELL" is present. Another section, "Panel de Cloud Console", provides a link to "panel de cargas de trabajo" and a blue button "ABRIR EL PANEL DE CARGAS DE TRABAJO". At the bottom right of the modal is a blue "ACEPTAR" button. The background shows the main Google Cloud navigation bar and some blurred cluster management options.

Google Cloud: Conectamos al cluster con cloud shell



Clústeres de Kubernetes

+ CREAR + IMPLEMENTAR C ACTUALIZAR

OPERATIONS APRENDIZAJE

DESCRIPCIÓN GENERAL OBSERVABILIDAD OPTIMIZACIÓN DE COSTOS

Filtro Ingresar el nombre o el valor de la propiedad

| <input type="checkbox"/> Estado | Nombre ▲ | Ubicación | Modo | Cantidad de nodos | CPU virtuales totales | Memoria total | Notificaciones | Etiquetas |
|---------------------------------|-----------------------------------|-------------|-----------|-------------------|-----------------------|---------------|----------------|-----------|
| <input type="checkbox"/> | ✓ umcloud-cluster-01 | us-central1 | Autopilot | 0 | 0 GB | - | - | - |

CLOUD SHELL

Terminal (um-cloud-221804) +

Abrir editor

Acciones

Conectar

Borrar

```
Welcome to Cloud Shell! Type "help" to get started.  
Your Cloud Platform project in this session is set to um-cloud-221804.  
Use "gcloud config set project [PROJECT_ID]" to change to a different project.  
navarrow@cloudshell:~ (um-cloud-221804)$ gcloud container clusters get-credentials umcloud-cluster-01 --region us-central1 --project um-cloud-221804  
Fetching cluster endpoint and auth data.  
kubeconfig entry generated for umcloud-cluster-01.  
navarrow@cloudshell:~ (um-cloud-221804)$ kubectl  
kubectl kubectx  
navarrow@cloudshell:~ (um-cloud-221804)$ kubectl get nodes  
NAME STATUS ROLES AGE VERSION  
gk3-umcloud-cluster-01-default-pool-639fe206-rhnw Ready <none> 3m v1.25.8-gke.500  
gk3-umcloud-cluster-01-default-pool-c09ea9ce-dtlg Ready <none> 3m1s v1.25.8-gke.500  
navarrow@cloudshell:~ (um-cloud-221804)$ 
```

Caso Práctico

- Queremos desplegar nuestra aplicación para gestión ágil KanBoard en nuestro flamante cluster k8s :)

The screenshot shows a KanBoard application interface titled "KB Demo Project". The top navigation bar includes "Menu", "Overview", "Board" (selected), "Calendar", "List", "Gantt", and a status filter "status:open". The main area displays a board with three columns: "Backlog", "Ready", and "Work in progress".

- Backlog:** Contains one task: "#52 - Update screenshots documentation". It is labeled "P0 <15m <15m".
- Ready:** Contains three tasks:
 - #53 - Fix API bug (edge case) under the "api" category. Status: "0/1h P3 <15m <15m".
 - #50 - Improve Markdown editor under the "markdown" and "user interface" categories. Status: "P0 <15m <15m".
 - #51 - Validate installation on Debian Jessie under the "sysadmin" and "os" categories. Status: "P0 <15m <15m".
- Work in progress:** Contains one task: "#49 - Improve the documentation documentation". It is labeled "P1 <15m <15m".

- Ref: <https://kanboard.org/>
- Ref: https://docs.kanboard.org/en/latest/admin_guide/docker.html

De Ref: https://docs.kanboard.org/en/latest/admin_guide/docker.html

Running the Container

Basic Usage

```
docker pull kanboard/kanboard:v1.2.9
```

```
docker run -d --name kanboard -p 80:80 -t kanboard/kanboard:v1.2.9
```

Si hay container entonces ... puede haber pod ... entonces puede haber deploy ...

Stateless

cd ../p05-caso

Deploy

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: kb-prod
  labels:
    run: kb-prod
spec:
  replicas: 1
  selector:
    matchLabels:
      run: kb-prod
  template:
    metadata:
      generateName: kb-prod
      labels:
        run: kb-prod
    spec:
      restartPolicy: Always
      containers:
        - name: kb-prod
          image: kanboard/kanboard:v1.2.9
          imagePullPolicy: Always
          ports:
            - name: kb-prod
              containerPort: 80
```

Service

```
apiVersion: v1
kind: Service
metadata:
  name: kb-prod
  labels:
    run: kb-prod
spec:
  ports:
    - port: 80
      name: kb-prod
      targetPort: 80
  selector:
    run: kb-prod
```

Ingress

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
    kubernetes.io/ingress.class: nginx
  name: kb-ing
spec:
  rules:
    - host: USER-kb.my.kube.um.edu.ar
      http:
        paths:
          - backend:
              service:
                name: kb-prod
                port:
                  number: 80
              path: /
              pathType: Prefix
```

Statefull ?

Sts ...

Service

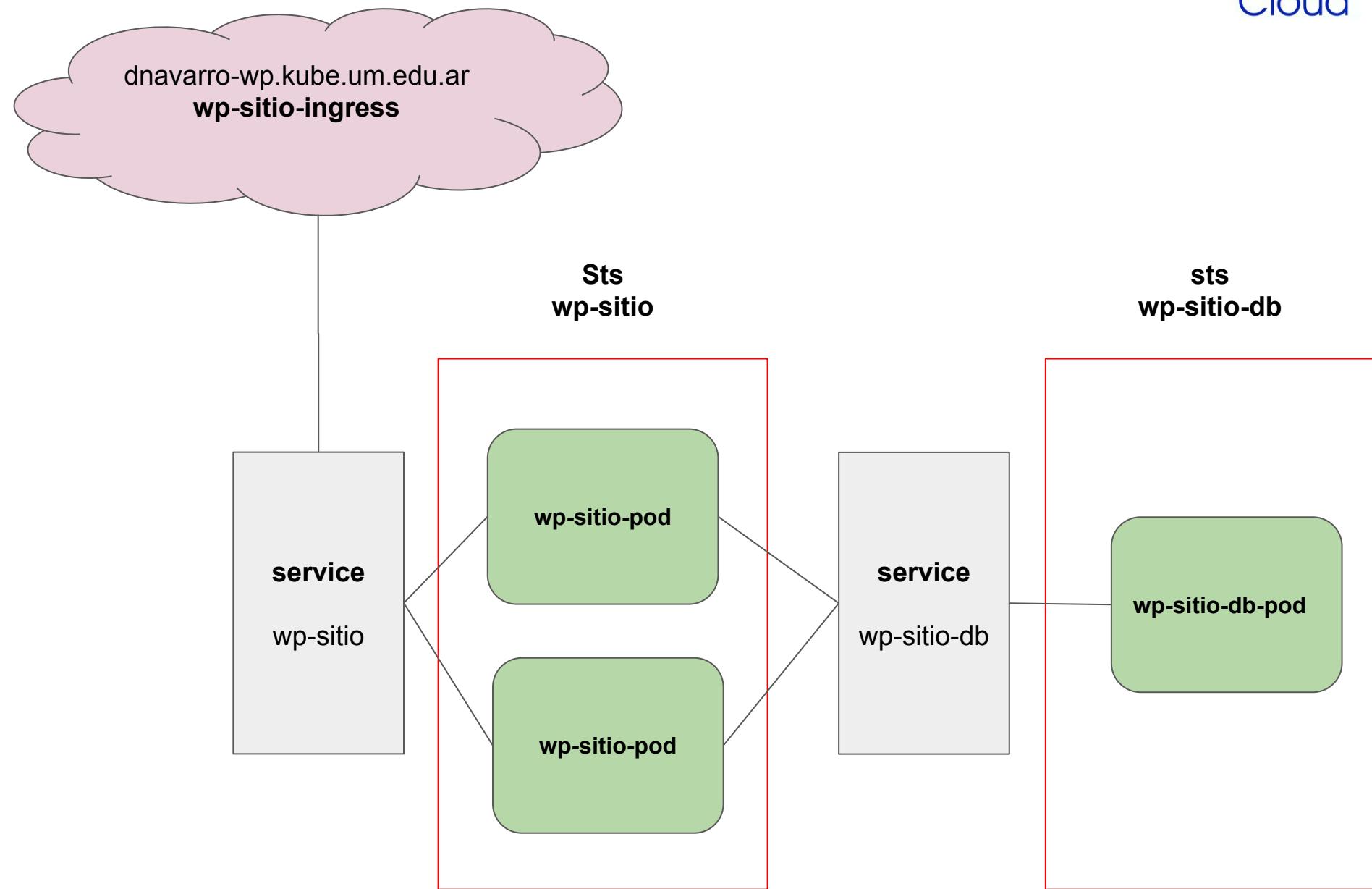
```
apiVersion: v1
kind: Service
metadata:
  name: kb-prod
  labels:
    run: kb-prod
spec:
  ports:
    - port: 80
      name: kb-prod
      targetPort: 80
  selector:
    run: kb-prod
```

Ingress

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
    kubernetes.io/ingress.class: nginx
  name: kb-ing
spec:
  rules:
    - host: USER-kb.my.kube.um.edu.ar
      http:
        paths:
          - backend:
              service:
                name: kb-prod
                port:
                  number: 80
            path: /
            pathType: Prefix
```

WorPress: Caso Cloud Native

Wordpress sobre Kubernetes [stateless]



Wordpress sobre Kubernetes [stateless]



```
kubectl -n tele get deploy,svc,ingress
```

| NAME | CLASS | HOSTS | ADDRESS | PORTS | AGE |
|--|--------|-------------------------------|--------------|-------|------|
| ingress.networking.k8s.io/wp-sitio-ing | <none> | navarrow.tele.cloud.um.edu.ar | 172.16.16.74 | 80 | 141m |

| NAME | TYPE | CLUSTER-IP | EXTERNAL-IP | PORT(S) | AGE |
|---------------------|-----------|---------------|-------------|----------|------|
| service/wp-sitio | ClusterIP | 10.43.234.31 | <none> | 80/TCP | 141m |
| service/wp-sitio-db | ClusterIP | 10.43.112.242 | <none> | 3306/TCP | 141m |

| NAME | READY | UP-TO-DATE | AVAILABLE | AGE |
|-----------------------------|-------|------------|-----------|------|
| deployment.apps/wp-sitio | 2/2 | 2 | 2 | 141m |
| deployment.apps/wp-sitio-db | 1/1 | 1 | 1 | 141m |

| NAME | READY | STATUS | RESTARTS | AGE |
|----------------------------------|-------|---------|----------|------|
| pod/wp-sitio-56d8b8b468-6fx6v | 1/1 | Running | 0 | 141m |
| pod/wp-sitio-56d8b8b468-stdtv | 1/1 | Running | 0 | 16s |
| pod/wp-sitio-db-69ddcd5dcd-pbbqz | 1/1 | Running | 0 | 141m |

Wordpress sobre Kubernetes [yaml's]



```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
annotations:
  nginx.ingress.kubernetes.io/rewrite-target: /
  kubernetes.io/ingress.class: nginx
  kubernetes.io/tls-acme: "false"
  name: wp-sitio-ing
spec:
rules:
- host: dnavarro-wp.my.kube.um.edu.ar
  http:
    paths:
      - backend:
          service:
            name: wp-sitio
            port:
              number: 80
        path: /
        pathType: Prefix

apiVersion: v1
kind: Service
metadata:
  name: wp-sitio
  labels:
    run: wp-sitio
spec:
  ports:
  - port: 80
    name: wp-sitio
  selector:
    run: wp-sitio

apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: wp-sitio
  labels:
    run: wp-sitio
spec:
  selector:
    matchLabels:
      run: wp-sitio
  serviceName: "wp-sitio"
  template:
    metadata:
      generateName: wp-sitio
      labels:
        run: wp-sitio
    spec:
      containers:
        - name: wp-sitio
          image: wordpress:php7.4-apache
          imagePullPolicy: IfNotPresent
          env:
            - name: WORDPRESS_DB_HOST
              valueFrom:
                secretKeyRef:
                  name: wp-sitio-sec
                  key: WORDPRESS_DB_HOST
            - name: WORDPRESS_DB_USER
              valueFrom:
                secretKeyRef:
                  name: wp-sitio-sec
                  key: WORDPRESS_DB_USER
            - name: WORDPRESS_DB_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: wp-sitio-sec
                  key: WORDPRESS_DB_PASSWORD
            - name: WORDPRESS_DB_NAME
              valueFrom:
                secretKeyRef:
                  name: wp-sitio-sec
                  key: WORDPRESS_DB_NAME
        ports:
          - name: wp-sitio
            containerPort: 80
  volumeMounts:
    - mountPath: "/var/www/html"
      name: wp-sitio-data

apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: wp-sitio-db
  labels:
    run: wp-sitio-db
spec:
  selector:
    matchLabels:
      run: wp-sitio-db
  serviceName: "wp-sitio-db"
  template:
    metadata:
      generateName: wp-sitio-db
      labels:
        run: wp-sitio-db
    spec:
      initContainers:
        - name: "remove-lost-found"
          image: "busybox:1.25.0"
          imagePullPolicy: "IfNotPresent"
          command: ["rm", "-fr", "/var/lib/mysql/lost+found"]
      volumeMounts:
        - name: mysql-data
          mountPath: /var/lib/mysql
      containers:
        - name: wp-sitio-db
          image: mysql:5.7
          imagePullPolicy: IfNotPresent
          env:
            - name: MYSQL_DATABASE
              valueFrom:
                secretKeyRef:
                  name: wp-sitio-db-sec
                  key: MYSQL_DATABASE
            - name: MYSQL_ROOT_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: wp-sitio-db-sec
                  key: MYSQL_ROOT_PASSWORD
        ports:
          - port: 3306
            name: wp-sitio-db
  selector:
    run: wp-sitio-db

ports:
  - name: wp-sitio-db
    containerPort: 3306
volumeMounts:
  - mountPath: "/var/lib/mysql"
```

Wordpress sobre Kubernetes [yaml's]



```
apiVersion: v1
kind: Secret
data:
  WORDPRESS_DB_HOST: d3Atc2l0aw8tZGI=
  WORDPRESS_DB_NAME: d3Atc2l0aw8=
  WORDPRESS_DB_PASSWORD: YjBkOWI2MD1j0WE5ZDQ2YmEyNzNkYjA4MGU2ZTAwNmY=
  WORDPRESS_DB_USER: cm9vdA==
metadata:
  creationTimestamp: null
  name: wp-sitio-sec
```

```
apiVersion: v1
kind: Secret
data:
  MYSQL_DATABASE: d3Atc2l0aw8=
  MYSQL_ROOT_PASSWORD: YjBkOWI2MD1j0WE5ZDQ2YmEyNzNkYjA4MGU2ZTAwNmY=
metadata:
  creationTimestamp: null
  name: wp-sitio-db-sec
```

Cloud Native

Como puedo seguir ...

pero ... cuidado con los Cloud Native ninjas !



Education

Recursos para aprender containers / Kubernetes

EdX: para iniciar

- <https://www.edx.org/course/introduction-to-cloud-infrastructure-technologies>
- <https://www.edx.org/course/introduction-to-kubernetes>

Katacoda: hands-on!

- <https://www.katacoda.com/courses/kubernetes/>

Cómo sigo ? / qué me gusta

- <https://roadmap.sh/>