Temario



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**.

Algunos indicadores





El mercado global de computación en nube tiene un valor de

\$180b en ingresos para los proveedores por SaaS/PaaS/laaS y el mercado continúa creciendo un 24% anual. Las Empresas que se benefician de los servicios de la nube que aumentan la productividad crecen

19.6%

más rápidamente que sus competidores.



Se pronostica que surjan



\$7,42b

de nuevas oportunidades en el mercado latinoamericano de laaS, para 2022. Para 2020, el 50%

de las principales 2000 empresas
del mundo verán que la mayoría de
sus negocios dependen de su
capacidad para crear productos,
servicios y experiencias mejorados
digitalmente.



Fuentes: Businesscomputingworld.co.uk, CenturyLink, Cloudcouncil.org, Cloudcomputing-news.net, Enterprisesproject.com, Centro Europeo para la Promoción de Importaciones, Gemalto.com, Harveynash.com, Intel.com, Skyhighnetworks.com, Statista, Rightscale.com; IDC Latinamerica Investment Trends; IDC Black Book 3rd.Platform Edition: Frost&Sullivan Latin American IaaS Market. Forecast.

Motivos para adopción de Cloud





Fuentes: Businesscomputingworld.co.uk, CenturyLink, Cloudcouncil.org, Cloudcomputing-news.net, Enterprisesproject.com, Centro Europeo para la Promoción de Importaciones, Gemalto.com, Harveynash.com, Intel.com, Skyhighnetworks.com, Statista, Rightscale.com; IDC Latinamerica Investment Trends; IDC Black Book 3rd.Platform Edition; Frost&Sullivan Latin American IaaS Market, Forecast.

Cloud computing ... La nube

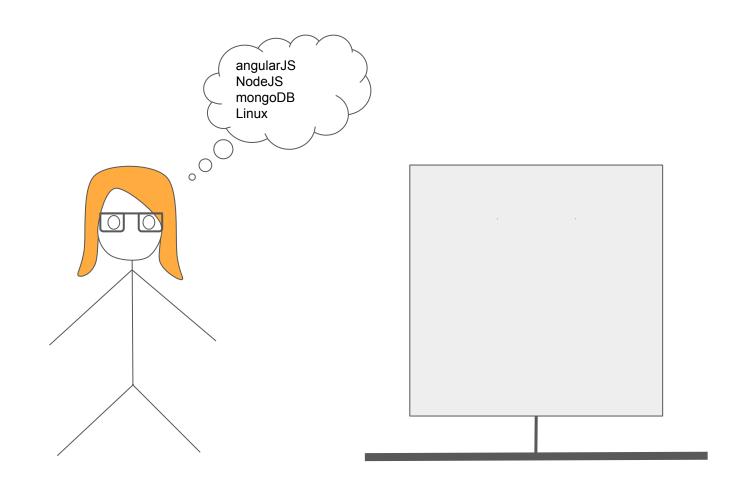
¿ lo qué?



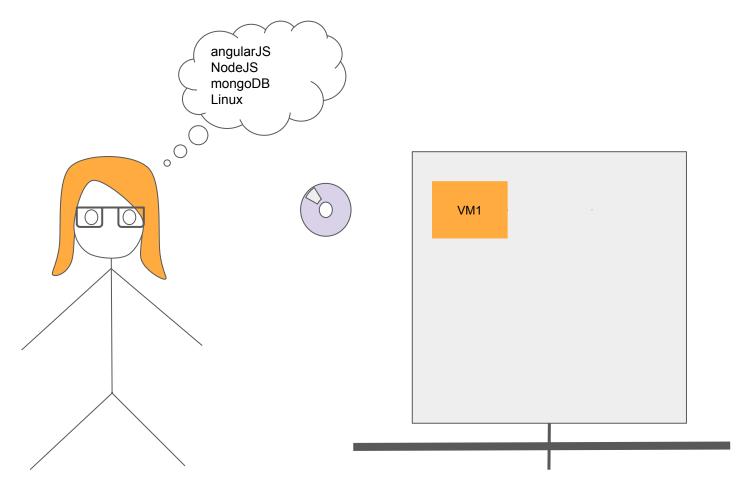




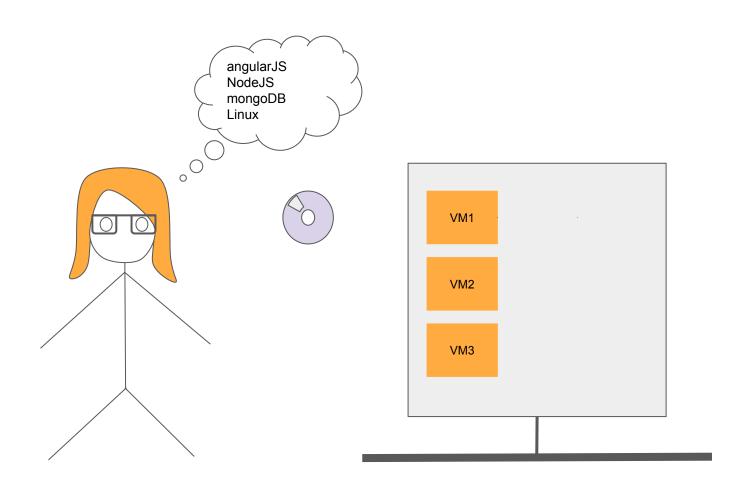




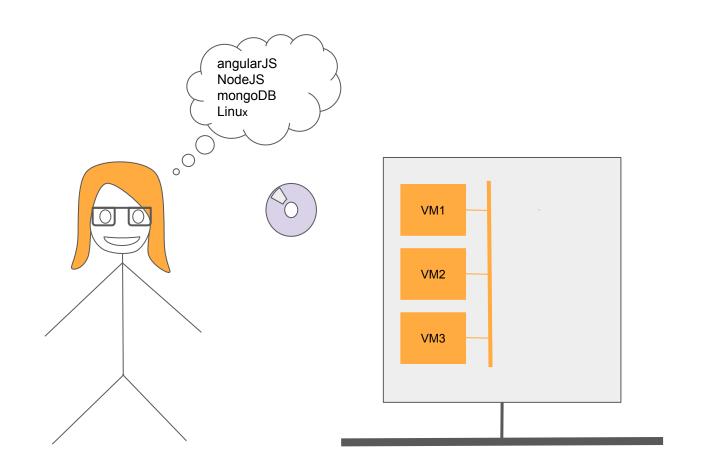




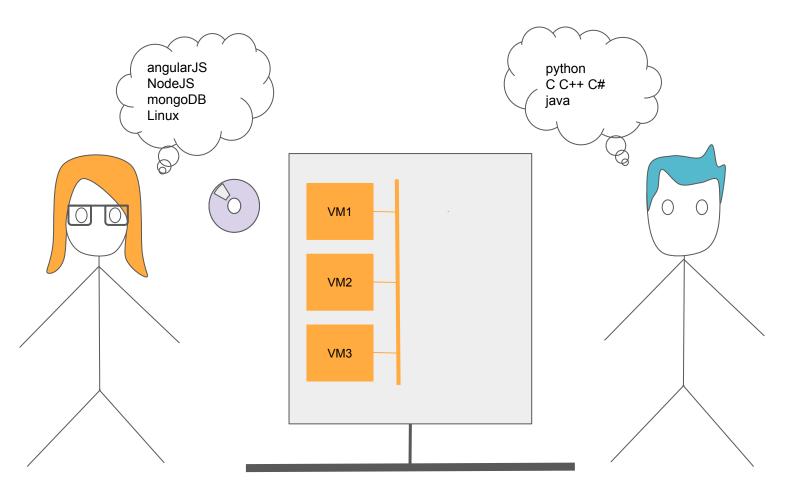




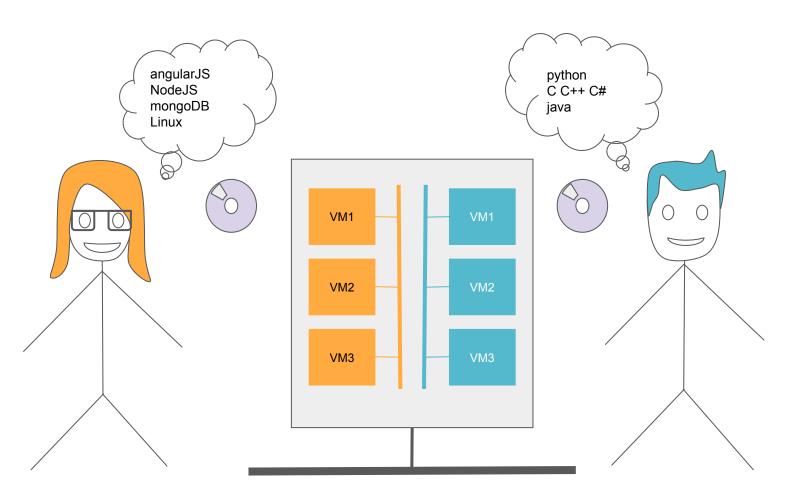




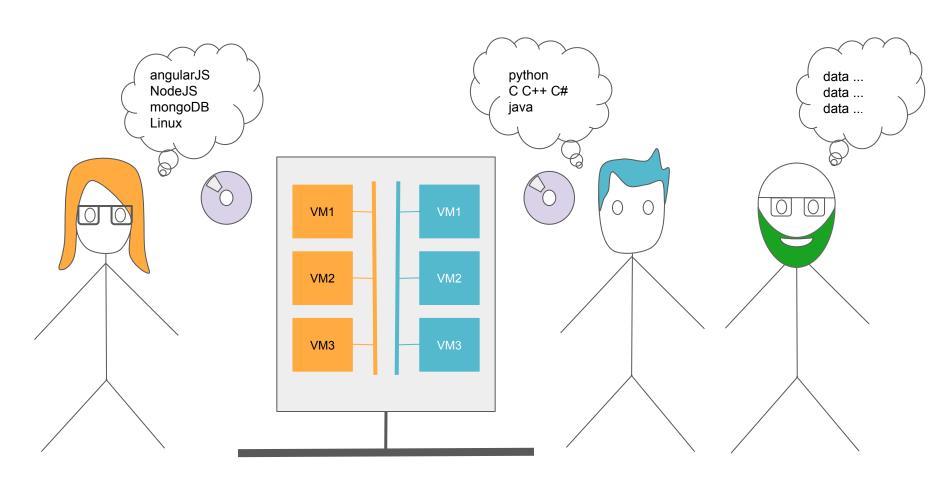




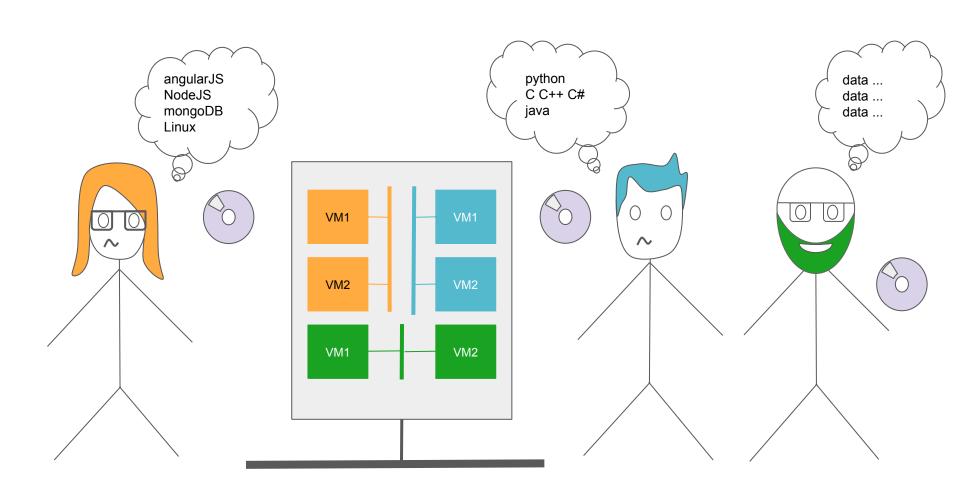




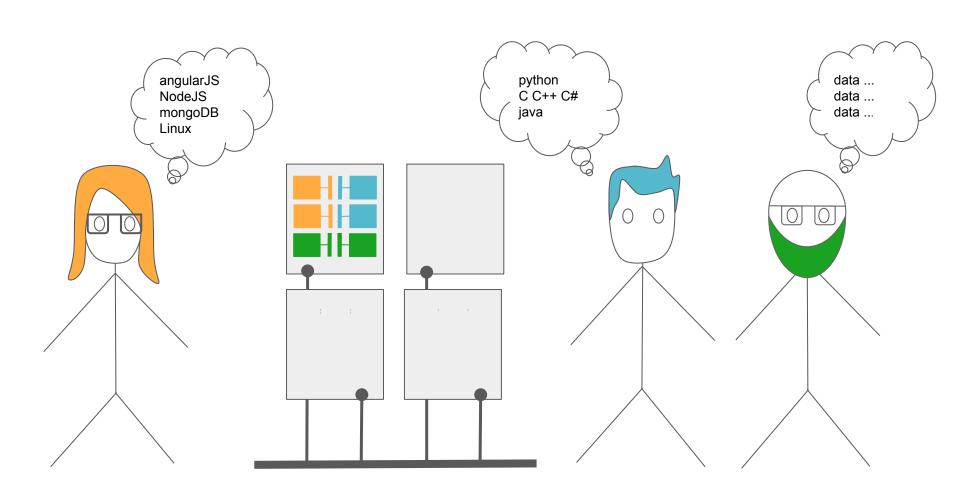




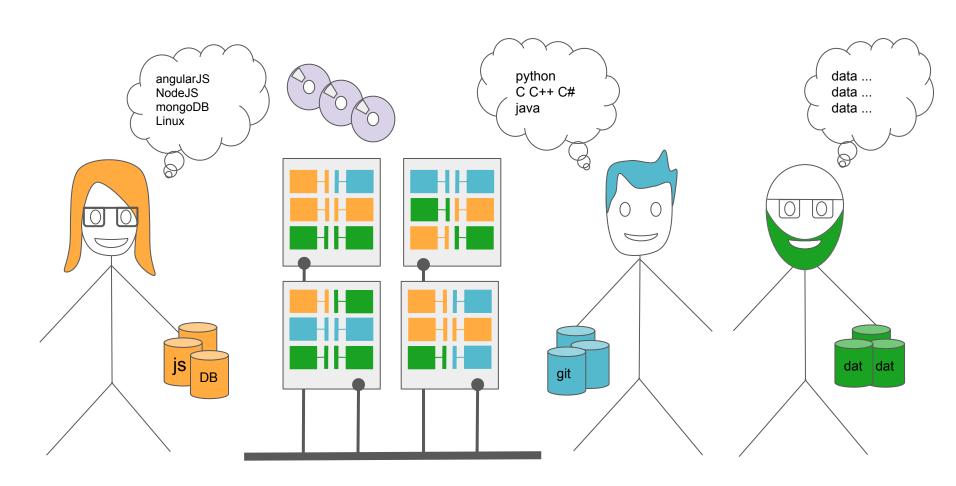


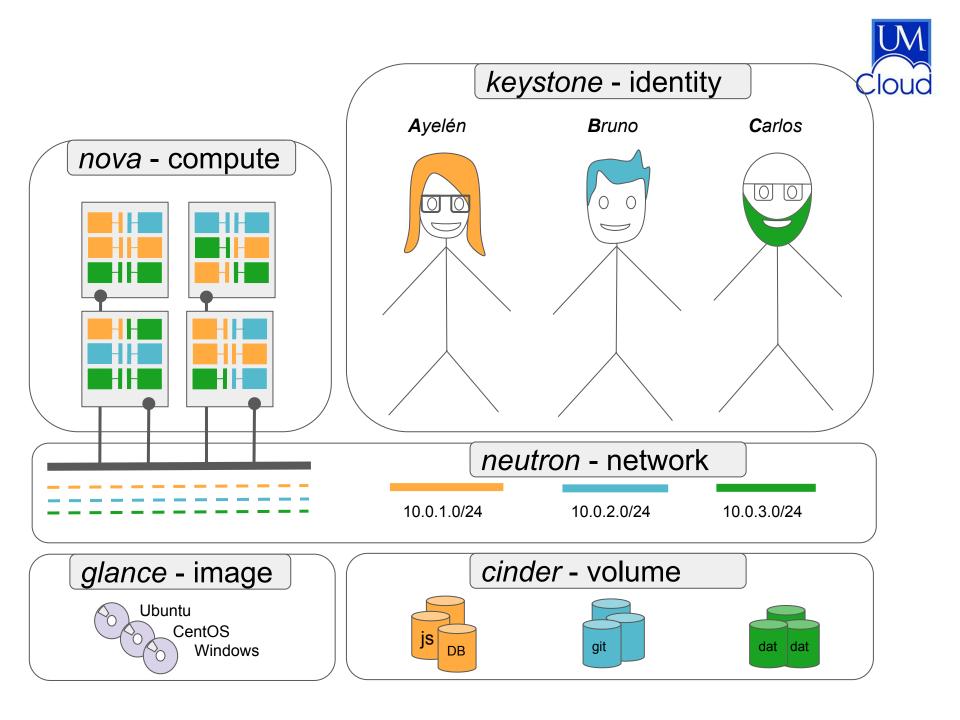












Cloud Computing: Definición



"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 r**ecursos de computación** (redes, servidores, almacenamiento, aplicaciones y servicios) que pueden ser provisionados facilmente y rapidamente 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
- laaS Infra aaS

Modelos de Despliegue

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

^{1 &}lt;a href="http://en.wikipedia.org/wiki/Cloud_computing">http://en.wikipedia.org/wiki/Cloud_computing

² The NIST Definition of Cloud Computing

Cloud Computing = Cloud Classic "para nos"



	On Premises	laaS	PaaS	SaaS	Cloud
	Aplicación	Aplicación	Aplicación	Aplicación	
	Datos	Datos	Datos	Datos	
	Tiempo de Ejecución	Tiempo de Ejecución	Tiempo de Ejecución	Tiempo de Ejecución	
	Sistema Operativo	Sistema Operativo	Sistema Operativo	Sistema Operativo	
	Servidores	Servidores	Servidores	Servidores	
	Almacenamiento	Almacenamiento	Almacenamiento	Almacenamiento	
	Red	Red	Red	Red	
mouting					

Cloud Computing

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



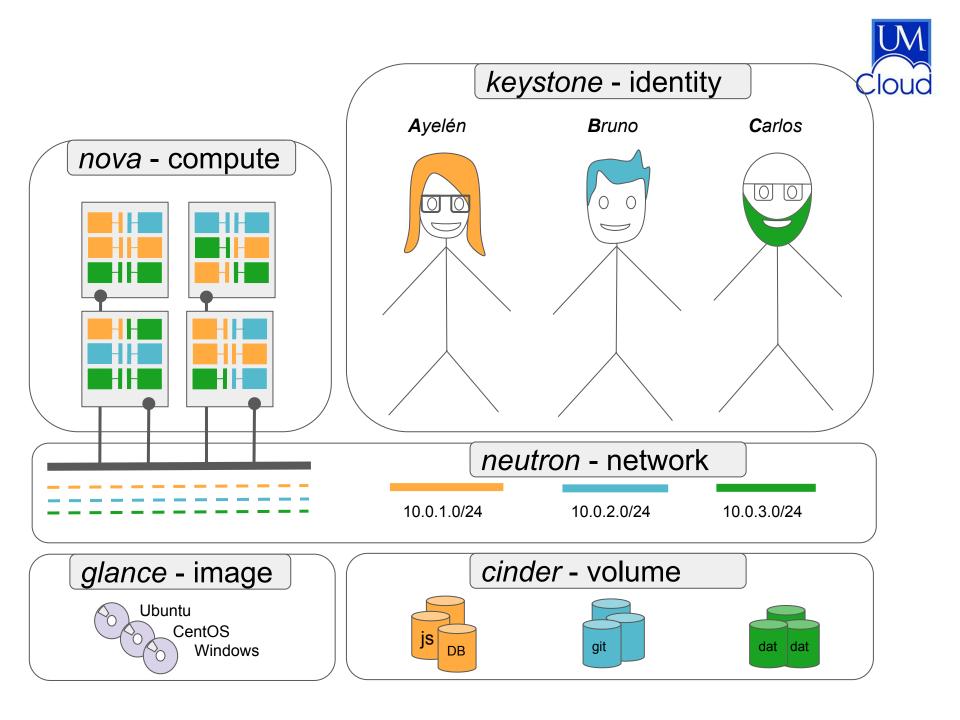
Administración del Proveedor



Paradigma que permite ofrecer recursos computacionales mediante Internet

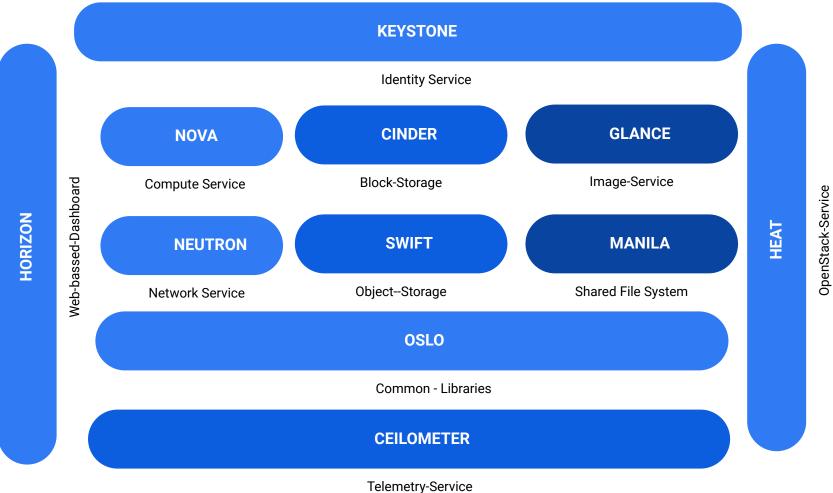


Recursos virtuales agrupados para diseñar y gestionar nubes Publicas o privadas



Cloud Computing





UMCloud: Onboarding



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



- 2. Oauth ["apellido um.edu.ar"]
 - a. @um.edu.ar
 - b. @alumno.um.edu.ar



UMCloud: Onboarding

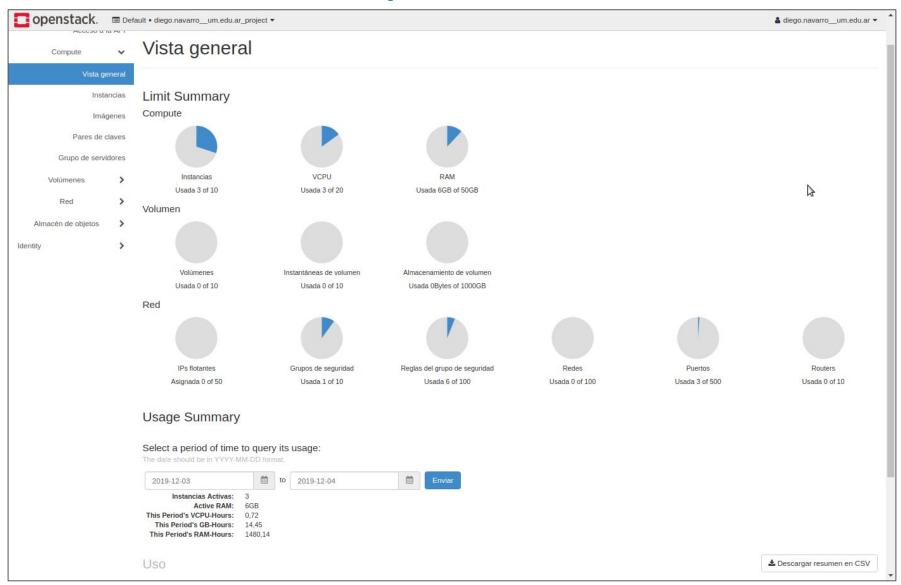


3. https://my.cloud.um.edu.ar



UMCloud: Onboarding

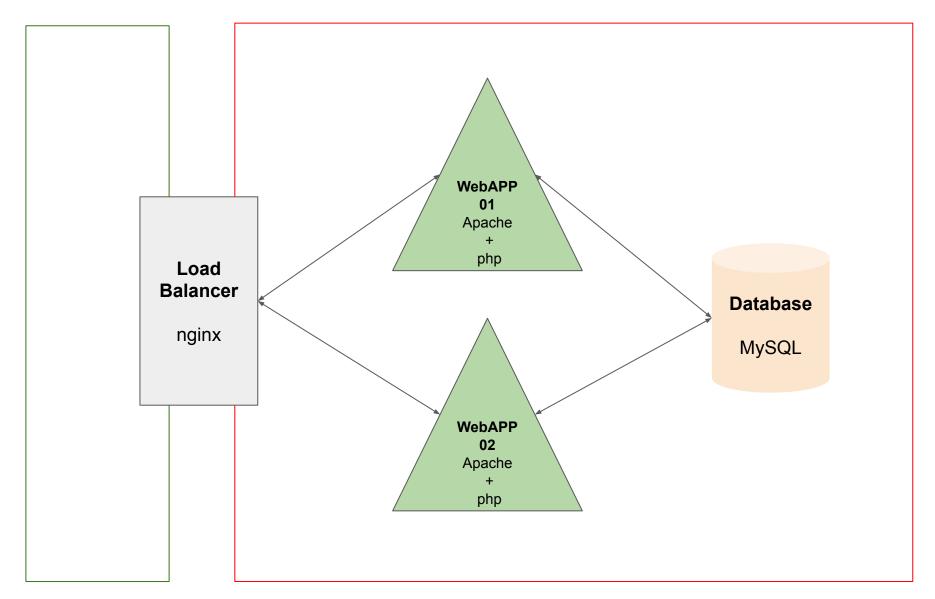
4. Horizon Dashboard https://console.cloud.um.edu.ar





Cloud Classic Caso Práctico







Database

MySQL

10.201.0.35

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: 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
#Poner Base de datos disponible en la RED
sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf
     #bind-address
                              = 127.0.0.1
     hind-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';
```

Necesitamos desplegar Wordpress y configurarlo para que acceda a la db remedio

- 1) Creamos instancia ubuntu_minimal_1804, llamarla: wp-webapp-01
- Nos logueamos: ssh ubuntu@<IP_instancia> sudo apt-get update sudo mount -o remount,nobarrier,commit=120 / sudo apt-get install -y wordpress php libapache2-mod-php php-mysql

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

WebAPP

Apache Php

10.201.0.33



#Apuntar navegador a http://10.201.0.33/blog/



```
Creamos instancia ubuntu_minimal_1804 , llamarla: wp-fe-01
1)
```

Load Balancer

nginx

```
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 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://10.201.0.33; # my backend WP
sudo service nginx restart
# Should show "backend" (WP) content
curl http://localhost/
# Ver logs de acceso en el frontend proxy
```

```
tail -f /var/log/nginx/access.log
```



Cloud Classic Tema Floating IP



```
# Con la Floating IP, hago un symlink para apuntar al mismo archivo

cd /etc/wordpress
sudo ln -s config-10.201.0.XX.php config-192.168.3.YY.php
sudo ln -s config-10.201.0.XX.php config-TUNOMBRE.tele.cloud.um.edu.ar.php

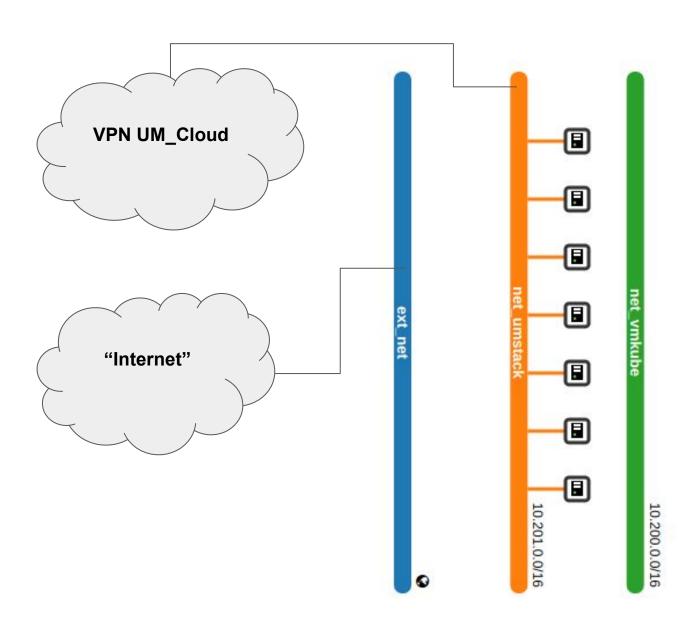
# En mysql, para reconstruir el sitio desde cero con el FQDN -^
DROP DATABASE wordpress;
CREATE DATABASE wordpress;
```



Cloud Classic Caso Práctico ++

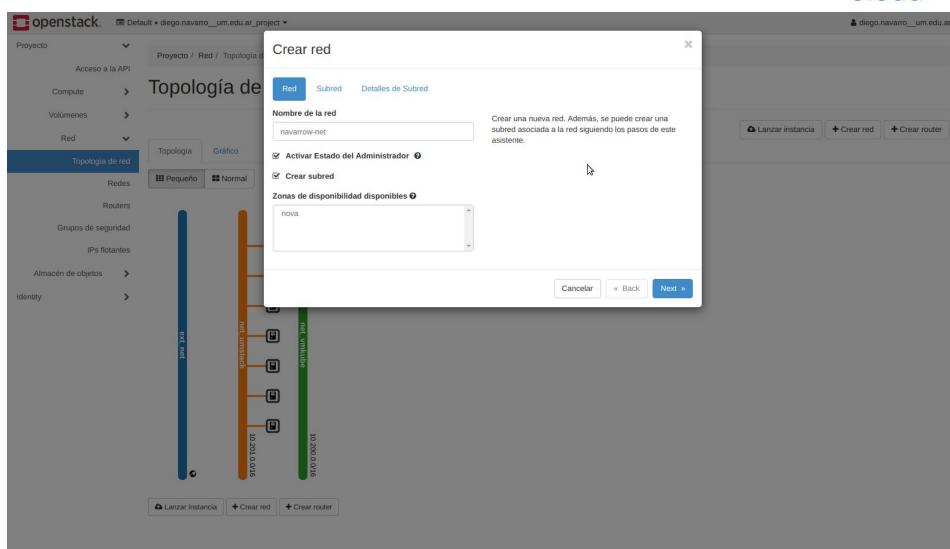
Cloud Networking: Crear Red



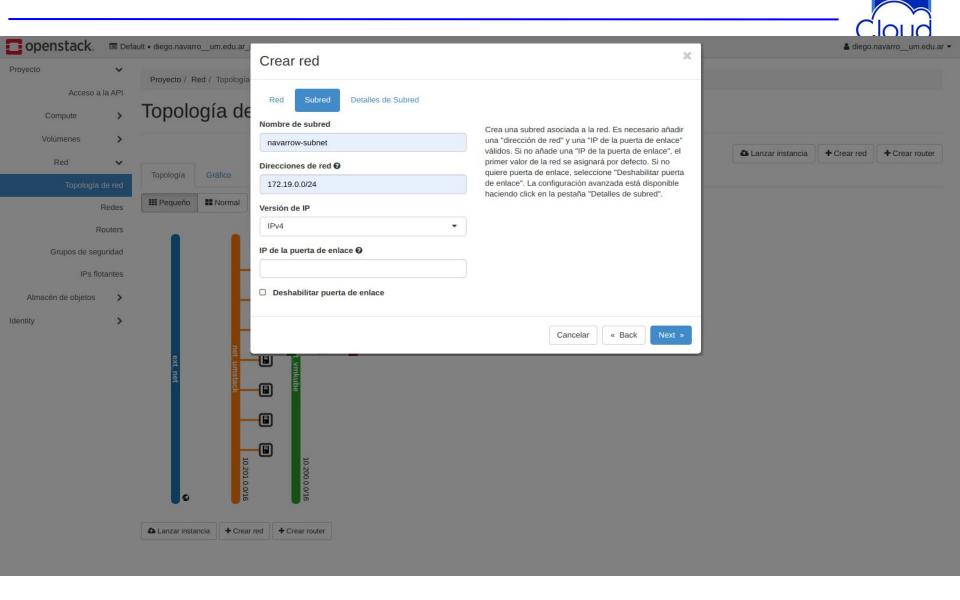


Cloud Networking: Crear Red



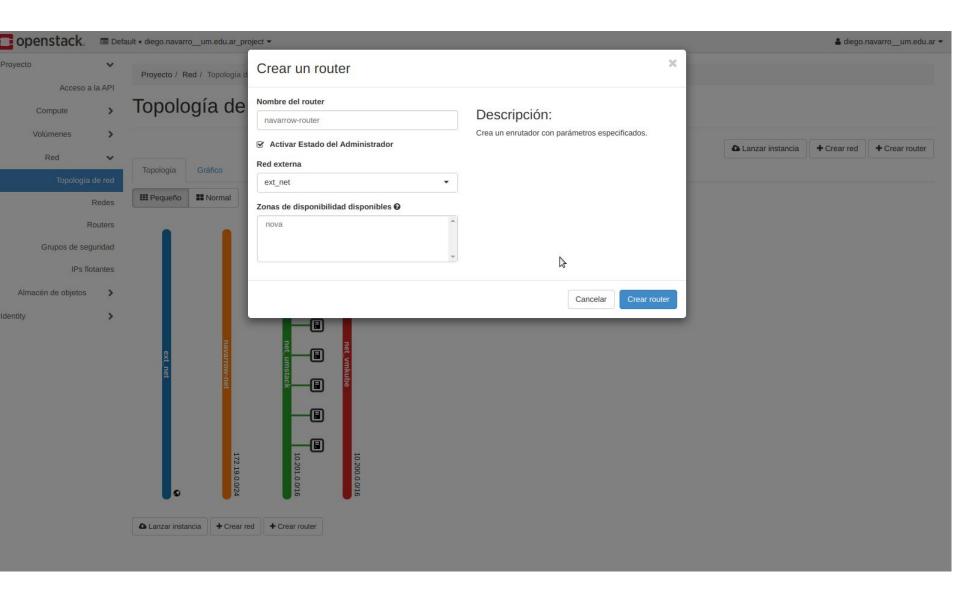


Cloud Networking: Crear Red



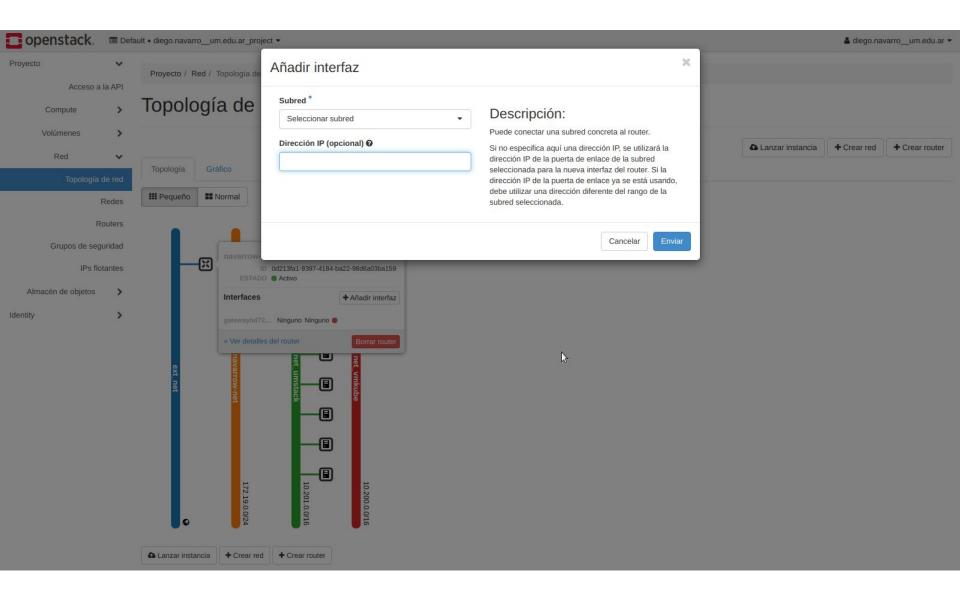
Cloud Networking: Crear Router





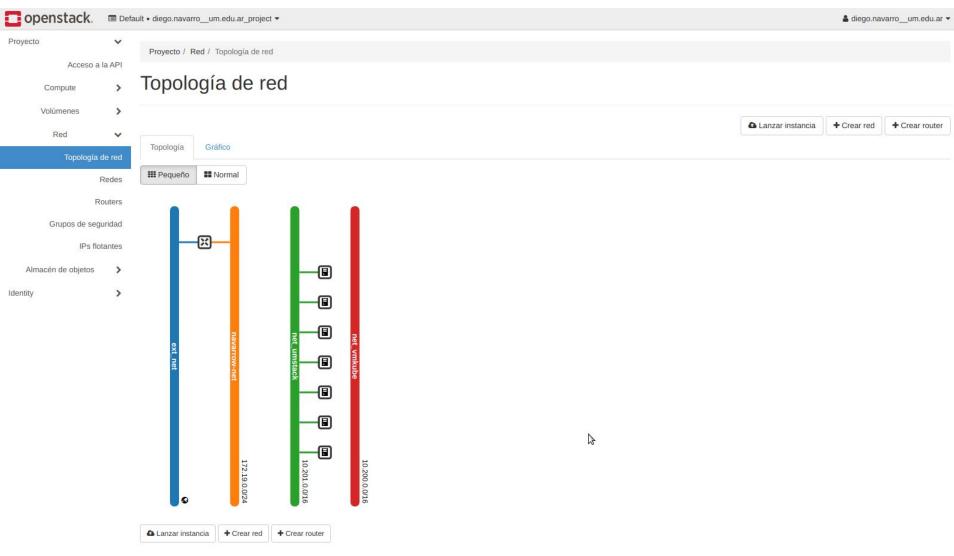
Cloud Networking: Crear Router, Añadir Interfaz





Cloud Networking





Instancias

Cloud

▣

▣

▣

Creamos nuevamente las instancias sobre la nueva red Privada y Asignamos la

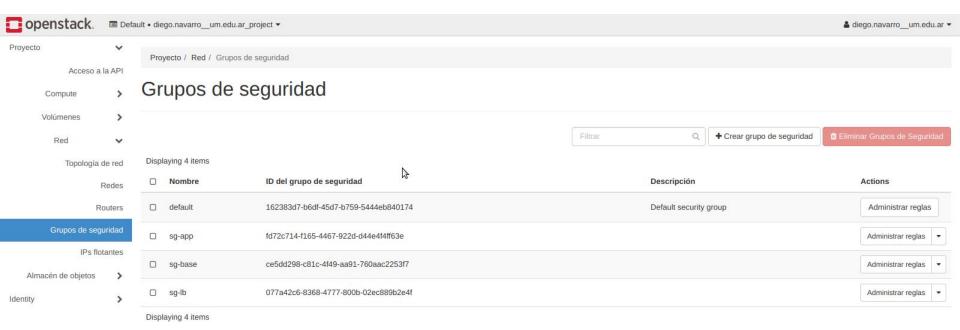
IP Elástica a una [bastión] [remember -> ssh -A]

0	Nombre de la instancia	Nombre de la imagen	Dirección IP	Sabor	Par de claves	Estado		Zona de Disponibilida
	db-priv	ubuntu_minimal_1804	172.19.0.10	m1.c1m1d20	navarrow	Activo	m ⁿ	workers-az
0	wp-priv	ubuntu_minimal_1804	172.19.0.8	m1.c1m1d20	navarrow	Activo	1	workers-az
0	lb-priv	ubuntu_minimal_1804	172.19.0.6 IPs flotantes: 192.168.3.85	m1.c1m1d20	navarrow	Activo	m ²	workers-az

Rehacemos la configuración de la clase anterior con el nuevo direccionamiento

Grupos de Seguridad





Grupos de Seguridad: sg-base [a todos]

Displaying 3 items





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





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





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







Cloud native

"Cloud Native" según CNCF.io



- CNCF == Cloud Native Computing Foundation
- CNCF Cloud Native Definition v1.0 [1]

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

Cloud Native - containers son cool, pero ...

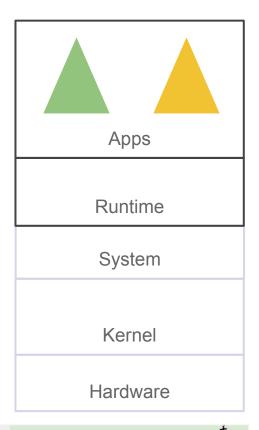




Cloud Native - un poco de arqueología ...



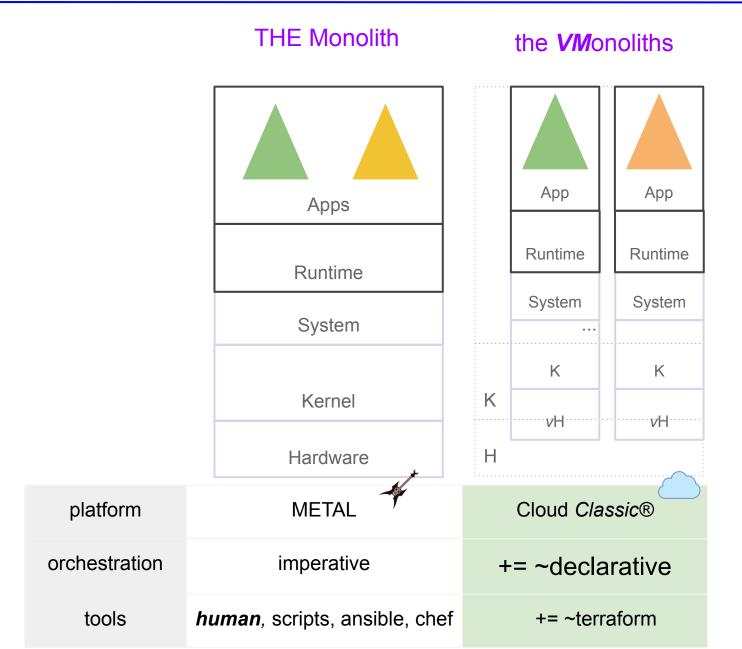
THE Monolith



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

Cloud Native - más cerca en el espacio-tiempo

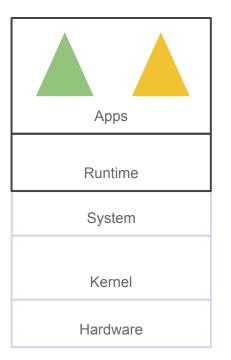




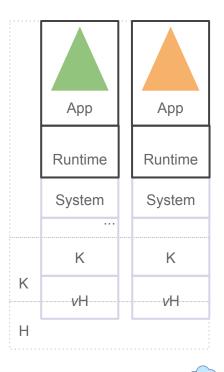
Cloud Native - desengrasándonos del sistema



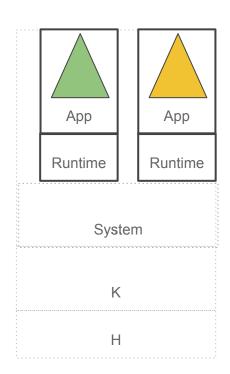
THE Monolith



The VMonoliths



Containers



n	latform
Ρ	alioiiii

orchestratio

tools

METAL*

imperative

human, scripts,

ansible, chef

Cloud Classic®

+= ~declarative

+= ~terraform

Cloud Native

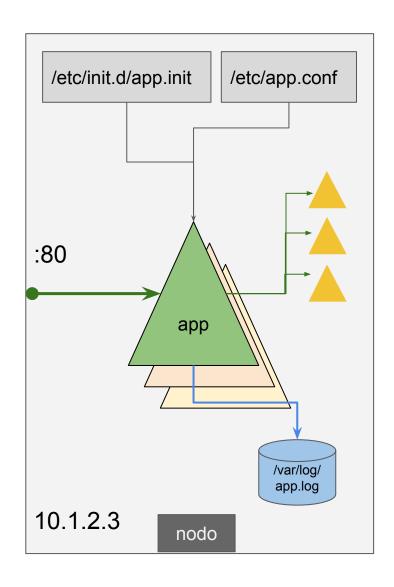
declarative

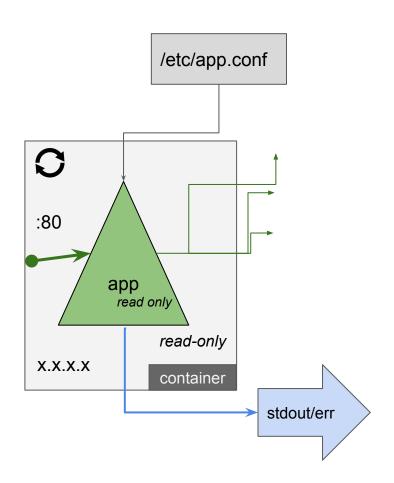
Kubernetes, Docker SWarm

Cloud Native - del monolito al nativo

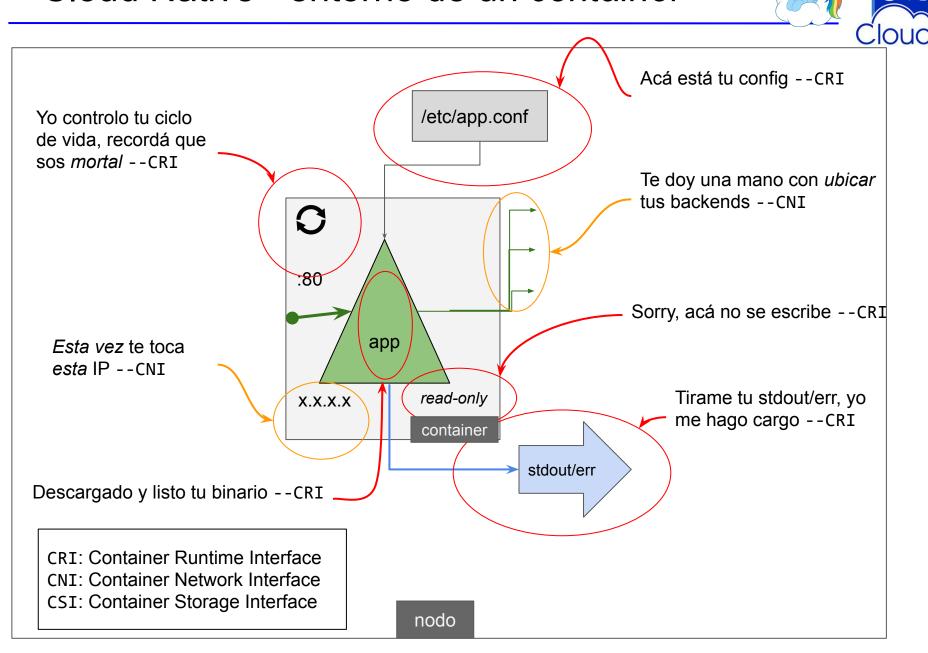








Cloud Native - entorno de un container



Cloud Native



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]
- Gestion[ables]
- Observables

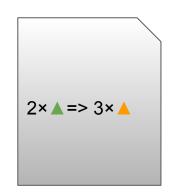
Combinadas con automatización robusta permite a los ingenieros hacer cambios de alto impacto con frecuencia y mínimo esfuerzo.

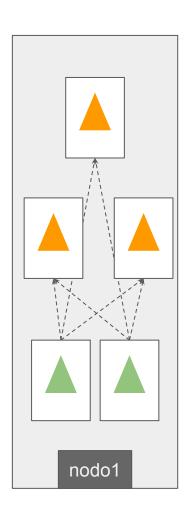
¿ Cómo sería ? ...

Beyond 12FactorApps

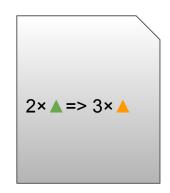
- 1. One codebase, one application
- 2. API first
- 3. Dependency management
- 4. Design, build, release, and run
- 5. Configuration, credentials, and code
- 6. Logs
- 7. Disposability
- 8. Backing services
- 9. Environment parity
- 10. Administrative processes
- 11. Port binding
- 12. Stateless processes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

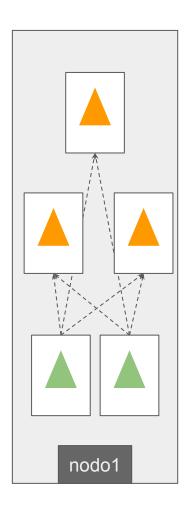


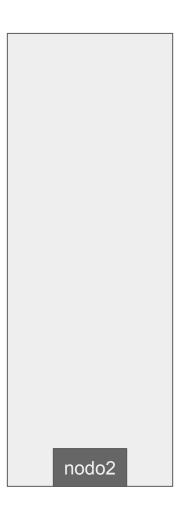


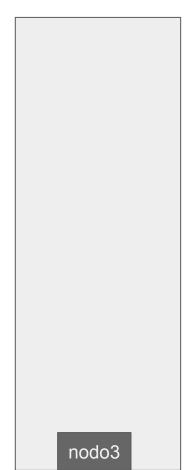




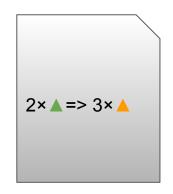


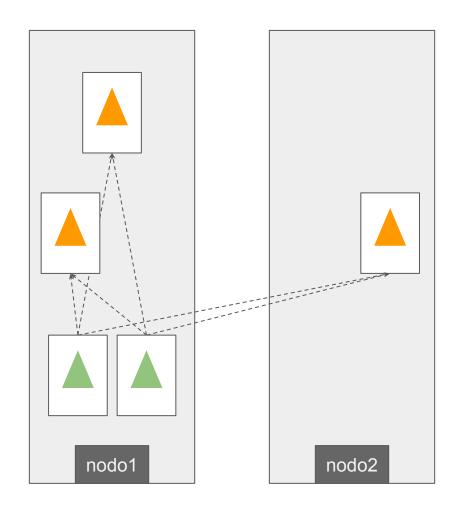


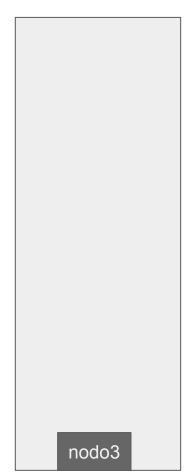




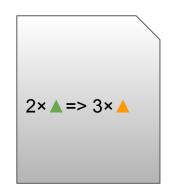


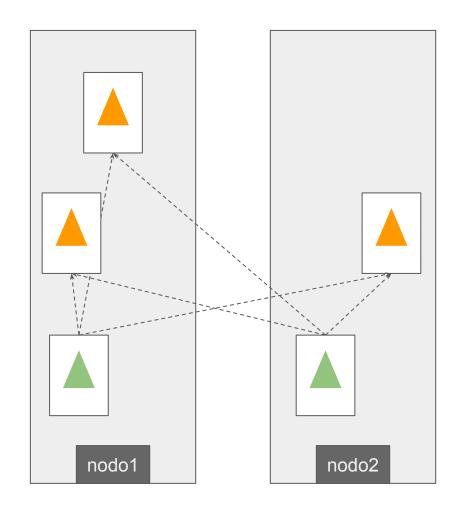


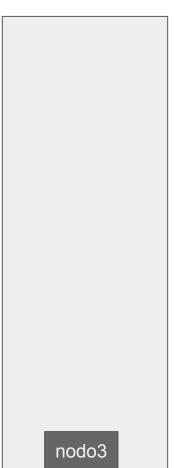




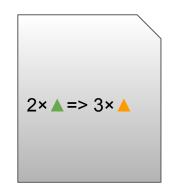


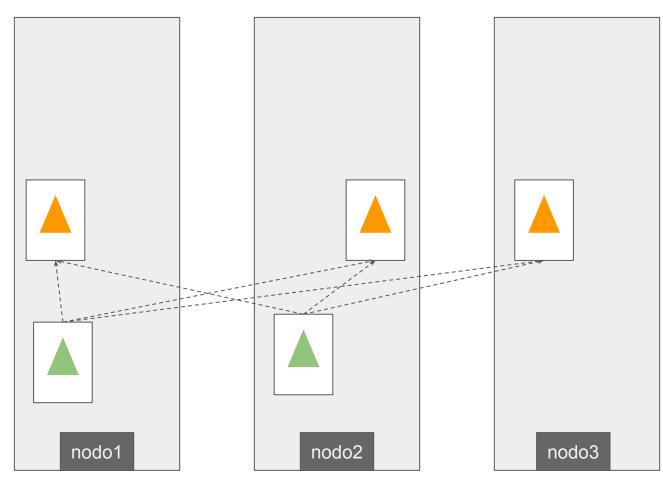




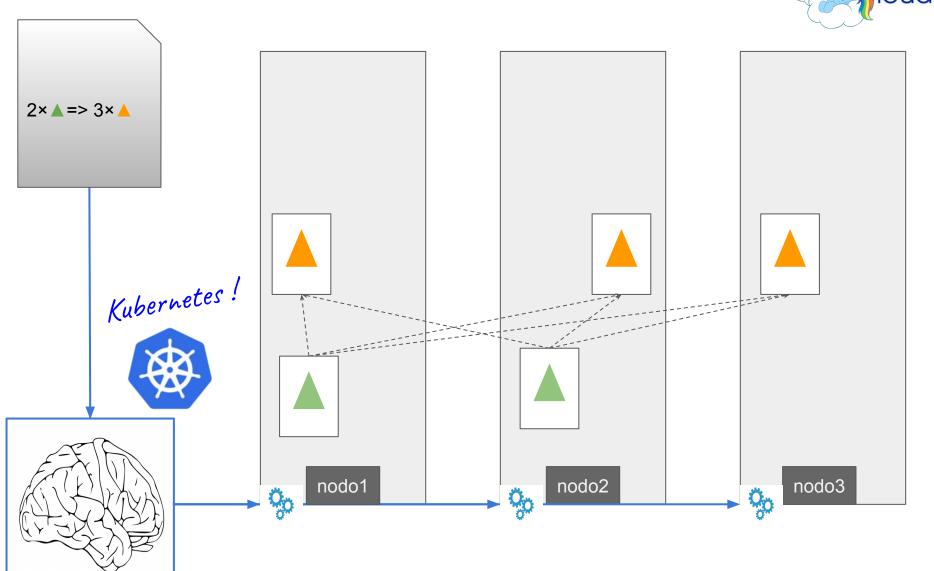












Kubernetes para todes





- "Containers cluster orchestrator":
 - Declarative Spec
 - 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
 - o CSI: provisión y *lifecycle* de almacenamiento persistente
 - o Multiplataforma:
 - **GKE**: Google Kubernetes Engine (Google)
 - EKS: Elastic Kubernetes Service (AWS)
 - AKS: Azure Kubernetes Service (Microsoft)
 - DIY: <u>kubespray</u>, <u>kubeadm</u>, <u>minikube</u> (p/dev) o <u>kubeadm-dind-cluster</u>, ...
 - ... desde Raspberry-PI hasta 5000 nodos
- Lingua franca! para Cloud Native Workloads
 - YAML, JSON
 - code + spec!
- Complejo para comenzar a aprender
 - "Kubernetes dificulta las cosas simples, y hace posible

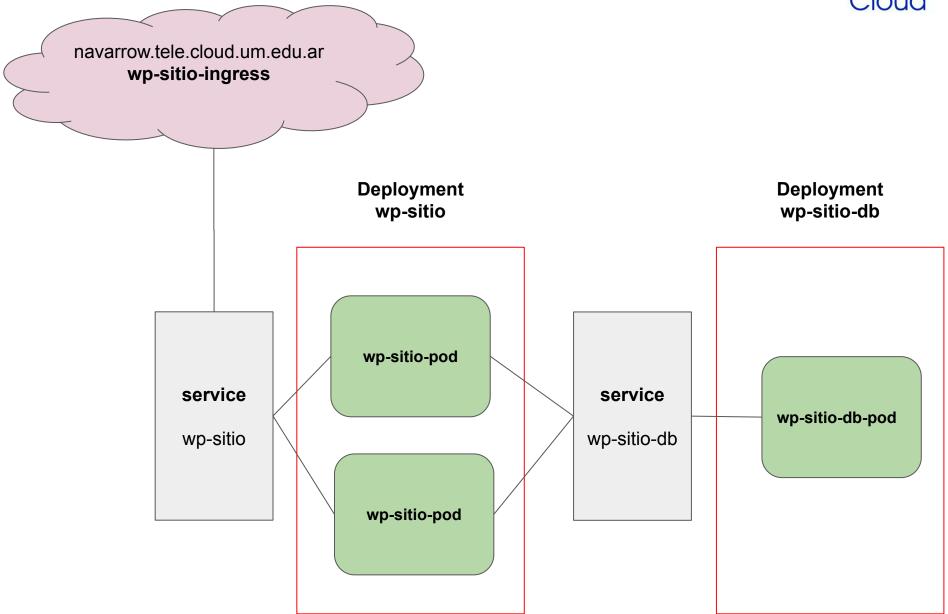




Cloud Native Caso Práctico

Wordpress sobre Kubernetes [stateless]





Wordpress sobre Kubernetes [stateless]



kubect1	- n	tele	get	denlov	,svc,ingress
I G D C C L L		~~~	200	исртоу	, , , , , , , , , , , , , , , ,

kubectl -n tele get deploy,svc,ingress								01000	
NAME	HOSTS			ADDRESS		PORTS	AGE		
<pre>ingress.extensions/wp-sit</pre>	navarrow.tele.cloud.um.edu.ar			172.16.16.208		80, 443	6m41s		
NAME	TYPE	CLUST	ER-IP	EXT	ERNAL-IP	PORT(S)	AGE		
<pre>service/wp-sitio</pre>	ClusterIP	10.233.31.4		<no< td=""><td>ne></td><td colspan="2">80/TCP 6m43s</td><td></td><td></td></no<>	ne>	80/TCP 6m43s			
<pre>service/wp-sitio-db</pre>	ClusterIP	10.233.51.183		<no< td=""><td>ne></td><td>3306/TCP</td><td>5m39s</td><td></td><td></td></no<>	ne>	3306/TCP	5m39s		
NAME		READY	UP-TO-DAT	Έ	AVAILABLE	AGE			
<pre>deployment.extensions/wp-</pre>	2/2	2		2	6m44s				
<pre>deployment.extensions/wp-</pre>	1/1	1		1	5m38s				

NAME	READY	STATUS	RESTARTS	AGE
<pre>pod/wp-sitio-65ddcccdbc-b9wgd</pre>	1/1	Running	0	2m28s
<pre>pod/wp-sitio-65ddcccdbc-hvn6h</pre>	1/1	Running	0	6m43s
<pre>pod/wp-sitio-db-7586549f8c-zbtvh</pre>	1/1	Running	0	5m38s

Wordpress sobre Kubernetes [yaml's]

volumes:

secret:

- name: wp-sitio-sec

secretName: wp-sitio-sec



```
apiVersion: apps/v1
                                                                                                                 apiVersion: apps/v1
apiVersion: extensions/v1beta1
                                                     kind: Deployment
                                                                                                                 kind: Deployment
kind: Ingress
                                                     metadata:
                                                                                                                 metadata:
metadata:
                                                       name: wp-sitio
                                                                                                                   name: wp-sitio-db
  annotations:
                                                       labels:
                                                                                                                   labels:
    cert-manager.io/cluster-issuer: letsencrypt-prod
                                                         run: wp-sitio
                                                                                                                     run: wp-sitio-db
    nginx.ingress.kubernetes.io/rewrite-target: /
                                                     spec:
  name: wp-sitio-ing
                                                                                                                 spec:
                                                       replicas: 1
                                                                                                                   replicas: 1
spec:
                                                       selector:
 rules:
                                                                                                                   selector:
                                                         matchLabels:
  - host: navarrow.tele.cloud.um.edu.ar
                                                                                                                     matchLabels:
                                                           run: wp-sitio
   http:
                                                       template:
                                                                                                                        run: wp-sitio-db
     paths:
                                                         metadata:
                                                                                                                   template:
     - backend:
                                                           labels:
                                                                                                                     metadata:
          serviceName: wp-sitio
                                                             run: wp-sitio
                                                                                                                       labels:
          servicePort: 80
                                                         spec:
                                                                                                                          run: wp-sitio-db
  tls:
                                                           restartPolicy: Always
                                                                                                                     spec:
    - hosts:
                                                           containers:
                                                                                                                        restartPolicy: Always
     - navarrow.tele.cloud.um.edu.ar
                                                             - name: wp-sitio
                                                                                                                       containers:
     secretName: wp-sitio-tls
                                                               image: wordpress
                                                                                                                          - name: wp-sitio-db
                                                               imagePullPolicy: Always
                                                                                                                            image: mysql:5.7
                                                                                                                            imagePullPolicy: Always
                                                                 - name: WORDPRESS DB HOST
                                                                   valueFrom:
                                                                                               apiVersion: v1
           apiVersion: v1
                                                                     secretKevRef:

    name: MYSQL ROOT PASSWORD

                                                                                               kind: Service
           kind: Service
                                                                       name: wp-sitio-sec
                                                                                                                                valueFrom:
                                                                                               metadata:
           metadata:
                                                                       key: WORDPRESS DB HOST
                                                                                                                                   secretKeyRef:
                                                                                                  name: wp-sitio-db
             name: wp-sitio
                                                                 - name: WORDPRESS DB USER
                                                                                                                                     name: wp-sitio-db-sec
                                                                                                  labels:
             labels:
                                                                   valueFrom:
                                                                                                                                     key: MYSQL ROOT PASSWORD
               run: wp-sitio
                                                                                                    run: wp-sitio-db
                                                                     secretKevRef:
                                                                                                                              - name: MYSQL DATABASE
            spec:
                                                                                               spec:
                                                                       name: wp-sitio-sec
                                                                                                                                valueFrom:
             ports:
                                                                                                  ports:
                                                                       key: WORDPRESS DB USER
                                                                                                                                   secretKeyRef:
             - port: 80
                                                                                                  - port: 3306
                                                                 - name: WORDPRESS DB PASSWORD
               name: wp-sitio
                                                                                                                                     name: wp-sitio-db-sec
                                                                                                    name: wp-sitio-db
                                                                   valueFrom:
             selector:
                                                                                                                                     key: MYSQL DATABASE
                                                                                                  selector:
                                                                     secretKeyRef:
               run: wp-sitio
                                                                                                                            ports:
                                                                       name: wp-sitio-sec
                                                                                                    run: wp-sitio-db
                                                                                                                               name: db-mysql
                                                                       key: WORDPRESS DB PASSWORD
                                                                                                                                 containerPort: 3306
                                                                 - name: WORDPRESS DB NAME
                                                                   valueFrom:
                                                                                                                       volumes:
                                                                     secretKeyRef:
                                                                                                                          name: secret-volume
                                                                       name: wp-sitio-sec
                                                                                                                            secret:
                                                                       key: WORDPRESS DB NAME
                                                                                                                              secretName: wp-sitio-db-sec
                                                               ports:
                                                                 - name: wp-sitio
                                                                   containerPort: 80
```

Wordpress sobre Kubernetes [yaml's]



```
apiVersion: v1
kind: Secret
data:
 WORDPRESS DB HOST: d3Atc210aW8tZGI=
 WORDPRESS DB NAME: d3Atc210aW8=
 WORDPRESS DB PASSWORD: YjBkOWI2MDljOWE5ZDQ2YmEyNzNkYjA4MGU2ZTAwNmY=
 WORDPRESS DB USER: cm9vdA==
metadata:
  creationTimestamp: null
  name: wp-sitio-sec
apiVersion: v1
kind: Secret
data:
 MYSQL DATABASE: d3Atc210aW8=
 MYSQL ROOT PASSWORD: YjBkOWI2MDljOWE5ZDQ2YmEyNzNkYjA4MGU2ZTAwNmY=
metadata:
  creationTimestamp: null
  name: wp-sitio-db-sec
  namespace: tele
```



Docker

Docker: Conceptos

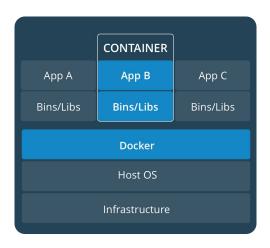


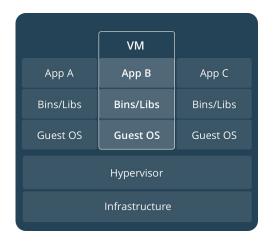
Una plataforma para:

"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.





Ref: https://docs.docker.com/get-started/

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
```

Ref: https://docs.docker.com/get-started/

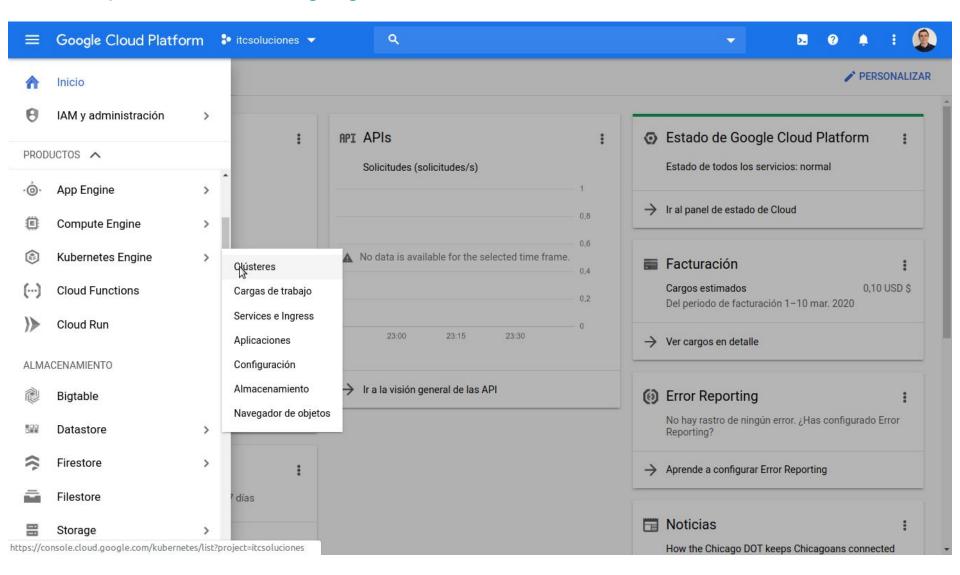


Cloud Native Como puedo seguir ...

Google Cloud [Kubernetes Engine]

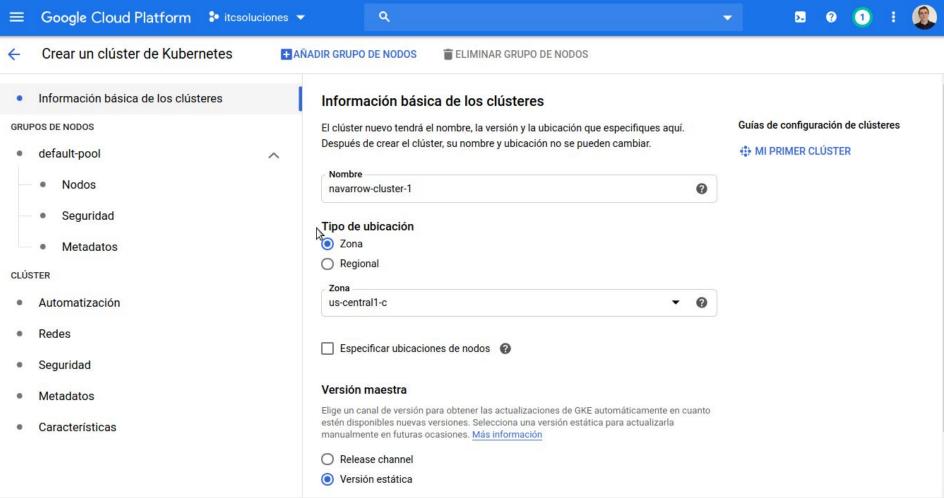


https://console.cloud.google.com



Google Cloud: Creamos el cluster





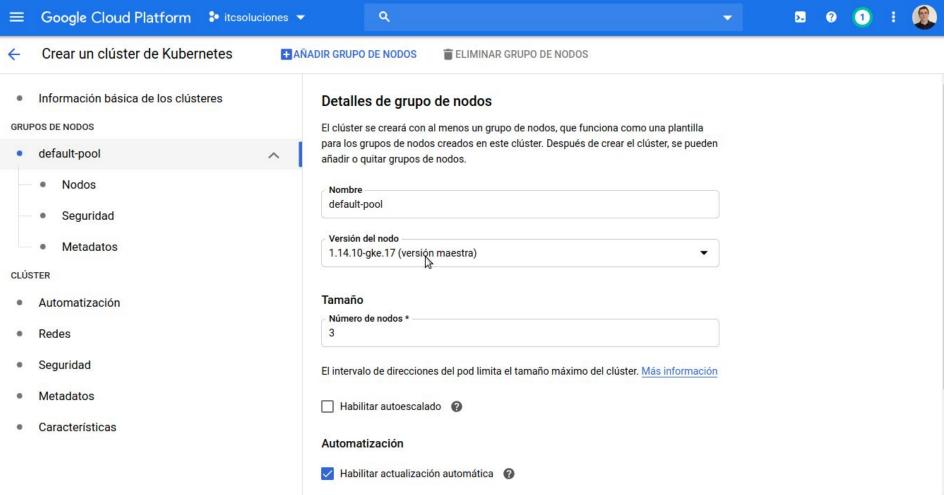
CANCELAR

CREAR

REST o línea de comando equivalente

Google Cloud: Creamos el cluster





CANCELAR

CREAR

REST o línea de comando equivalente

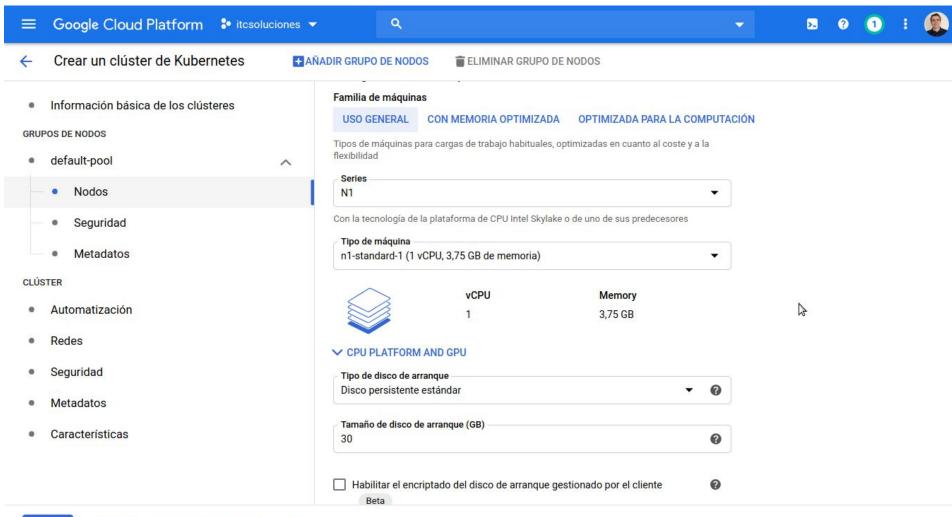
Google Cloud: Creamos el cluster

CREAR

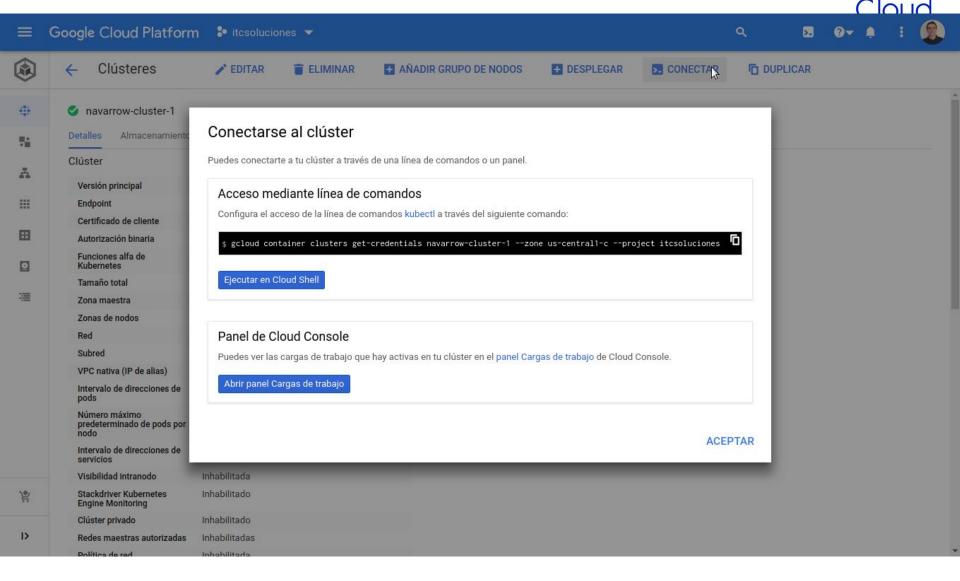
CANCELAR

REST o línea de comando equivalente

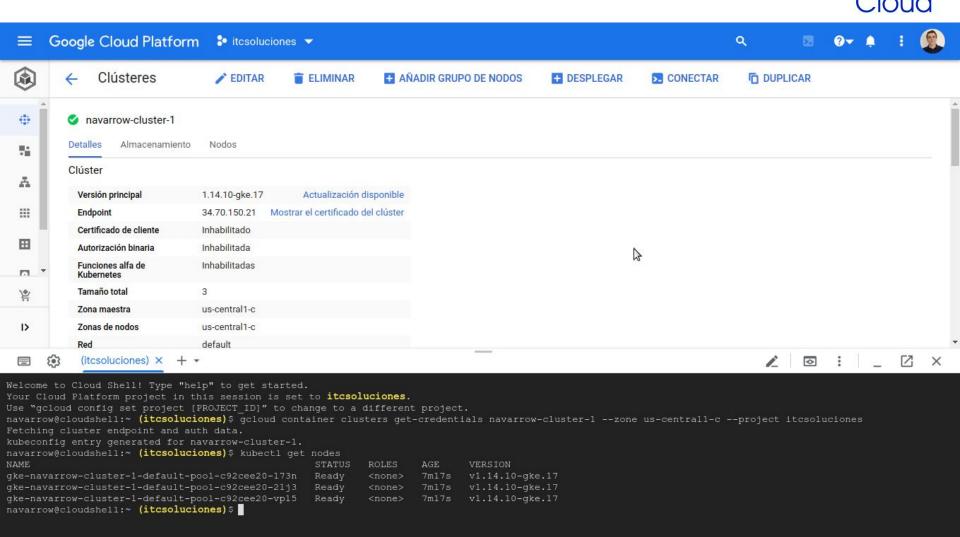




Google Cloud: Conectamos al cluster con cloud sh



Google Cloud: Conectamos al cluster con cloud sh



pero ... cuidado con los Cloud Native ninjas!



