



Universidad
De La Salle®
Bajío

Facultad de
**Tecnologías
de Información**

MTIE 501

Modelos de arquitecturas orientadas a los servicios

Proyecto Final - SOA
Solución Tecnológica BI
UDLSB

Presenta:
Marco Antonio Ramos Herrera

Profesor:
José Luis Rosas Peimbert

León, Gto. Mayo de 2021

INDICE

INDICE	2
Tabla de Ilustraciones	2
INTRODUCCIÓN.....	3
OBJETIVO.....	4
DESARROLLO	4
Infraestructura a utilizar	4
Servidores Hyperconvergentes	4
Máquina Virtual	5
Contenedores	5
Herramientas de apoyo	7
Bases de datos	7
Diagrama de Arquitectura	8
Maapeo de los componentes en los dominios de TI	10
Procedimiento de despliegue	11
Dashboards Kibana	16
CONCLUSIONES.....	17
REFERENCIAS.....	18

Tabla de Ilustraciones

ILUSTRACIÓN 1 DESPLIEGUE DE IAC CON TERRAFORM.....	3
ILUSTRACIÓN 2 CLUSTER SERVIDORES HYPERCONVERGENTES CON 10 NODOS.....	4
ILUSTRACIÓN 3 RECURSOS DE LA VM RANCHEROS	5
ILUSTRACIÓN 4 CONEXIÓN VÍA SSH A VM RANCHEROSPROYFINAL.....	5
ILUSTRACIÓN 5 DIAGRAMA DE ARQUITECTURA	9
ILUSTRACIÓN 6 DOMINIOS DE TECNOLOGÍAS DE INFORMACIÓN EN LAS EMPRESAS.....	10
ILUSTRACIÓN 7 IMPORTACIÓN DE OBJETOS EN KIBANA	15
ILUSTRACIÓN 8 DASHBOARD DE ALUMNOS UDLSB.....	16
ILUSTRACIÓN 9 DASHBOARD DE DOCENTES UDLSB.....	16
ILUSTRACIÓN 10 DASHBOARD DE ESTADÍSTICAS TEAMS.....	17

INTRODUCCIÓN

Si bien las máquinas virtuales (VM) tradicionales permiten la virtualización de la infraestructura de computación, los contenedores habilitan la de las aplicaciones de software. A diferencia de las máquinas virtuales, los contenedores utilizan el sistema operativo (SO) de su host en lugar de proporcionar uno propio.

Dado que no incluyen sistemas operativos completos, los contenedores requieren recursos informáticos mínimos y son rápidos y fáciles de instalar. Esta eficiencia permite que se implementen en clústeres, con contenedores individuales que encapsulan componentes únicos de aplicaciones complejas. Separar los componentes de la aplicación en diferentes contenedores permite a los desarrolladores actualizar componentes individuales sin necesidad de rehacer la aplicación completa.

Lo que se esconde detrás del software de contenedores es prácticamente lo mismo. En vez de hacer un envío de un sistema operativo completo y su software, simplemente se hace un paquete de código con todas sus dependencias dentro de un contenedor que se puede ejecutar en cualquier lugar. Además, como por norma general los contenedores suelen ocupar muy poco espacio, es posible almacenar muchos paquetes de contenedores dentro de un solo ordenador.

La infraestructura como código (IaC) es la automatización de la infraestructura que utiliza principios y prácticas de desarrollo de software. La idea es que trate su infraestructura como software y luego escriba, pruebe y ejecute código para definir, implementar, actualizar y destruir su infraestructura. Escribe código para administrar sus servidores, bases de datos, redes, registros, implementación y configuración de aplicaciones. Cuando desee realizar cambios en su infraestructura, realice cambios en el código, lo pruebe y luego lo aplique a sus sistemas.

Permite controlar más eficientemente los cambios y las configuraciones en un entorno de nube. La idea detrás de la IaC es llevar las prácticas encontrada en DevOps que permitan a los desarrolladores y a las operaciones trabajar más estrechamente en la implementación de máquinas virtuales, las redes virtuales que se construyen alrededor de ellas, y la aplicación en las mismas.

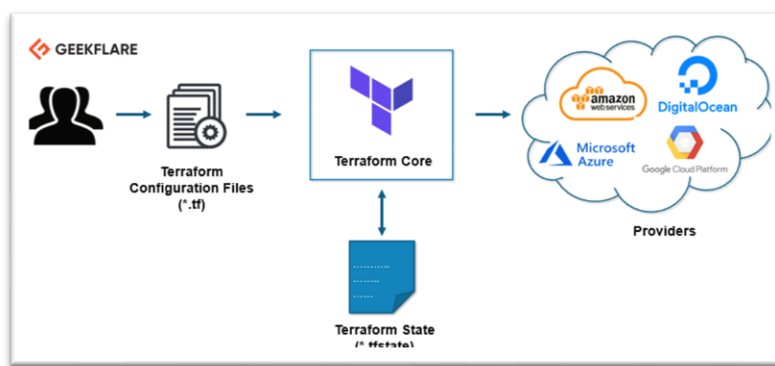


Ilustración 1 Despliegue de IaC con Terraform

OBJETIVO

El presente proyecto tiene como finalidad principal desarrollar una solución tecnológica de inteligencia de negocios (BI), para que pueda ayudar al análisis, estadísticas y conocimiento de indicadores en la Universidad de la Salle Bajío; dicha solución tecnológica se realiza mediante el uso de diferentes herramientas que permiten un despliegue automático de toda la infraestructura necesaria. Con la ventaja de realizar dicho despliegue en cuestión de minutos; con dicha solución se están abarcando todos los dominios de TI en base el enfoque de TOGAF.

DESARROLLO

Infraestructura a utilizar

Servidores Hyperconvergentes

Para la implementación de la solución, se está haciendo uso de un Cluster de 7 Servidores Hyperconvergentes (nodos) en los cuáles se cuenta con el hardware integrado en dicho cluster como memoria RAM, procesadores, y el almacenamiento necesario para en caso de que se presente alguna falla en uno de los nodos, las máquinas virtuales alojadas en el cluster no presenten ninguna afectación. La capacidad de dicho Cluster es de 280 procesadores, 2.6 TB de Memoria RAM, y 26 TB de espacio de almacenamiento.

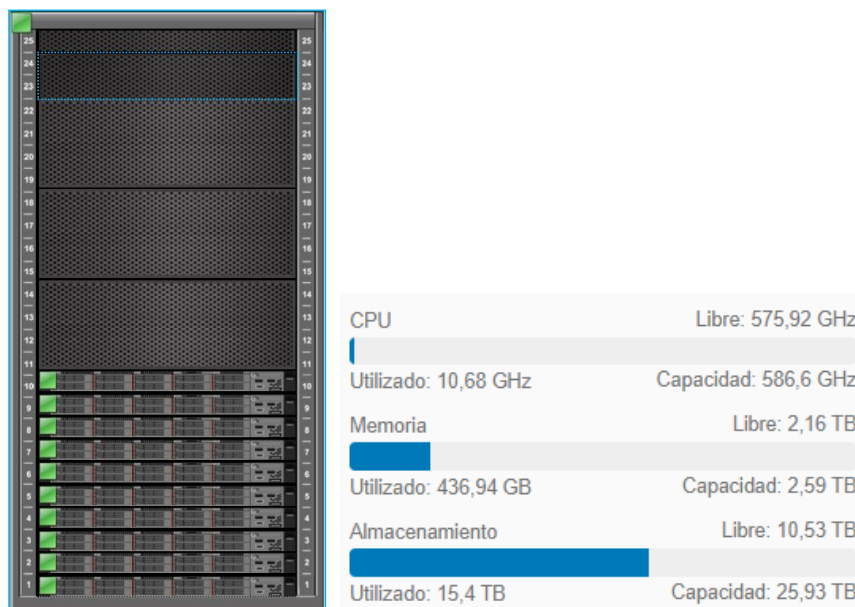


Ilustración 2 Cluster Servidores Hyperconvergentes con 10 nodos

Esos servidores hyperconvergentes tienen instalado Lenovo XClarity Pro para la administración y gestión del hardware VMWare ESXi para la gestión de virtualización, y se cuenta con licenciamiento de VCenter para así mediante el cliente VSphere administrar las VM del cluster.

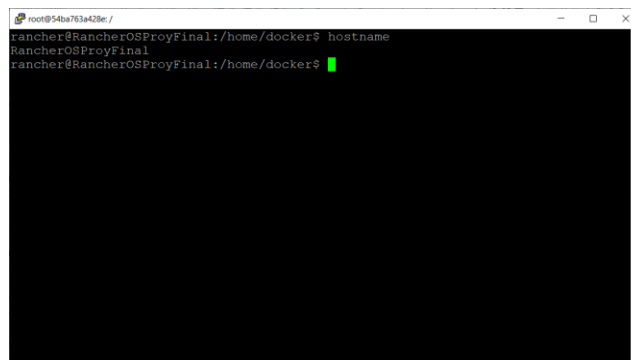
Máquina Virtual

Dentro de ese cluster de servidores hyperconvergentes, se va a crear una máquina virtual mediante una plantilla con el sistema operativo RancherOS, la cual se desplegará automáticamente cada vez que se requiera. Dicha VM cuenta con una capacidad de 8GB en RAM, 4 procesadores, y 20GB de espacio en disco duro, y su adaptador de red correspondiente. A la plantilla se le llamó “RancherOS_Plantilla”, y tiene instalado el sistema operativo mediante un procedimiento donde se modificó el archivo “cloud-config.yml”, en el cuál se generó una llave privada para la conexión al SO. Además las credenciales que quedaron configuradas en el mismo archivo y son las que se utilizarán para ingresar son “rancher/P4sswOrd”.

Hardware de máquina virtual	
> CPU	4 CPU
> Memoria	7.94 GB, 1.27 GB memoria activa
> Disco duro 1	20 GB
> Adaptador de red 1	Servidores95. (conectado)
> Tarjeta de video	4 MB
Dispositivo VMCI	Dispositivo del bus PCI de la máquina virtual que brinda compatibilidad con la interfaz de comunicación de la máquina virtual.
> Otro	Hardware adicional
Compatibilidad	ESXi 6.7 y posterior (máquina virtual versión 14)

Ilustración 3 Recursos de la VM RancherOS

Estos recursos son suficientes para la carga que tendrá la VM, ya que en dicho servidor es donde se alojarán todos los contenedores que proveerán los servicios; todos estos contenedores se crearán utilizando la herramienta Docker, la cual (Docker engine) ya se encuentra preinstalada en el sistema operativo RancherOS. El nombre de la VM de RancherOS es **RancherOSProyFinal**, el cual al terminar el despliegue automático se puede ingresar mediante SSH para la configuración correspondiente.

A terminal window titled 'root@54ba763a42be: /' showing a successful SSH connection to a host named 'RancherOSProyFinal'. The prompt is 'rancher@RancherOSProyFinal: /home/docker\$' and the user has entered the command 'hostname', which returns 'RancherOSProyFinal'.

```
root@54ba763a42be: /
rancher@RancherOSProyFinal: /home/docker$ hostname
RancherOSProyFinal
rancher@RancherOSProyFinal: /home/docker$
```

Ilustración 4 Conexión vía SSH a VM RancherOSProyFinal

Contenedores

Los contenedores son un paquete de elementos que permiten ejecutar una aplicación determinada en cualquier sistema operativo, en nuestro caso será RancherOS, y se utilizan para garantizar que una determinada aplicación se ejecute correctamente cuando cambie su entorno, sin dar fallos de ningún tipo. Los contenedores se generarán automáticamente mediante un despliegue automatizado utilizando el archivo docker-compose.yml. Los contenedores que tendrá esta solución son los siguientes:

Nombre	Imagen	Función	Virtual Host
Reverse-Proxy	nginx-proxy:alpine	Se encarga de recibir todas las peticiones para dirigirlas a los contenedores o aplicaciones que corresponda, incluso se puede utilizar para balanceo de cargas.	--
MTIE-Portal	nginx	Servidor web que en esta solución se tiene para cuestión de pruebas del servicio.	portal.mtiemarco.net
MTIE-Inicio	nginx	Servidor web que se tiene como “index” o página principal de la solución tecnológica, de ahí se pueden redirigir a los demás sitios del sistema.	inicio.mtiemarco.net
MTIE-MySQL	mysql:8	Contenedor para gestión de base de datos SQL, en donde se aloja la tabla de “alumnos”.	--
MTIE-MongoDB	mongo:4.0	Contenedor para gestión de base de datos no relacionales, en donde están alojadas las colecciones “docentes” y “teams”.	--
MTIE-Logstash	Dockerfile-logstash	Contenedor que se encargará de sincronizar los registros de la base de datos de MySQL con la aplicación de Elasticsearch para poderlos visualizar en los tableros de Kibana	--
MTIE-Monstache	rwynn/monstache:rel6	Contenedor que se encargará de sincronizar los registros de las colecciones de la base de datos no relacional Mongo para poderlos almacenar en Elasticsearch para visualizar en los tableros de Kibana	--
MTIE-ElasticSearch	docker.elastic.co/elasticsearch/elasticsearch:7.9.2	Contenedor de búsqueda y analítica que se encarga de almacenar los datos de ambas bases de datos para visualizar los datos en los tableros de Kibana	--
MTIE-Kibana	docker.elastic.co/kibana/kibana:7.9.2	Contenedor de panel de visualización de datos de Elasticsearch en el cual se encontrarán los tableros de la solución de BI.	kibana.mtiemarco.net
MONITOR	netdata/netdata	Contenedor para visualizar y monitorear métricas en tiempo real como el uso de cpu, disco duro, red y otros parámetros de la VM y contenedores.	monitor.mtiemarco.net

Herramientas de apoyo

Para el despliegue de la solución tecnológica se utilizarán métodos de IaC (Infraestructura como código) con el que se va a automatizar la creación de toda la infraestructura mencionada anteriormente, para ello se utilizará Terraform.

- **Terraform.**- Con esta herramienta se realiza la creación de la VM en Vsphere gracias a la plantilla generada en un inicio del RancherOS; tomando en cuenta todas las especificaciones mencionadas en el archivo main.tf; al terminar de generar la VM, en el mismo código se están ejecutando scripts que realizan una conexión a la misma VM, para comenzar a descargar repositorios, ajustar las carpetas en las ubicaciones necesarias, para terminar ejecutando el docker-compose para el despliegue de todos los contenedores. Con la ejecución de este despliegue se tiene aproximadamente un 80% de toda la infraestructura completa, y en un tiempo aproximado de 5 minutos.
- **GIT.**- Se utiliza la herramienta **git** para generar un repositorio para todos los scripts que se utilizarán en Terraform para el despliegue de la infraestructura, se estuvieron trabajando con diferentes versiones a lo largo del desarrollo de la solución, para lo cuál el uso de esta herramienta ayudó bastante para ir gestionando las diferentes versiones; se creó un repositorio final directamente en github, solo que se debe de tener en cuenta que dicho repositorio queda publicado en internet de manera pública. El repositorio usado en este proyecto es: <https://github.com/tonydumatrix/MTIE-Proy3>

Bases de datos

Para el desarrollo del presente sistema se está utilizando una base de datos tipo SQL gestionada por MySQL, y otra base de datos no relacional por medio de MongoDB.

Base de datos SQL.

En la base de datos SQL se tiene una tabla almacenan los registros de los alumnos actuales de la universidad que contiene las siguientes columnas:

```
CREATE TABLE `alumnos` (  
  `matricula` varchar(15) NOT NULL,  
  `nombre` varchar(200) NOT NULL,  
  `paterno` varchar(50) DEFAULT NULL,  
  `materno` varchar(50) DEFAULT NULL,  
  `correo` varchar(50) DEFAULT NULL,  
  `nivel` varchar(50) DEFAULT NULL,  
  `plantel` varchar(50) DEFAULT NULL,  
  `facultad` varchar(100) DEFAULT NULL,  
  `programa` varchar(200) DEFAULT NULL,  
  `grupo` varchar(20) DEFAULT NULL,  
  `date` date DEFAULT NULL,  
  PRIMARY KEY (`matricula`)
```

Y los datos de la base de datos son los siguientes:

- **MYSQL_DATABASE:** alumnos
- **MYSQL_TABLE:** alumnos

- **MYSQL_USER:** student_user
- **MYSQL_PASSWORD:** i_love_alumnos

Base de datos no relacional MongoDB.

En la base de datos no relacional MongoDB se tienen 2 colecciones, una de ellas es donde se encuentran los datos de los docentes actuales de la Universidad, y otra colección donde se almacenan por cada semana el uso de la plataforma de Microsoft Teams tanto de alumnos como docentes. Los datos de cada colección son los siguientes:

collection docentes

```
"campus.string(), facultad.string(), programa.string(), nivel.string(), folio.string(), paterno.string(), materno.string(), nombre.string(), correo.string(), grupo.string(), clavemat.string(), materia.string(), hrssemana.string(), dia.string(), corte.date(2006-01-02)"
```

collection teams

```
"nombre.string(), correo.string(), mensajeschat.int32(), reunionesorganizadas.int32(), reunionesparticipadas.int32(), llamadas.int32(), audiotime.int32(), videotime.int32(), screentime.int32(), corte.date(2006-01-02)"
```

Y los datos de la base de datos dbdata son:

- **MONGO_INITDB_ROOT_USERNAME:** root
- **MONGO_INITDB_ROOT_PASSWORD:** mongoadmin
- **MONGO_DATABASE:** dbdata
- **User en dbdata:** dbo-operator
- **Password:** operadoradmin

Diagrama de Arquitectura

La solución tecnológica se estructuró mediante el siguiente diagrama de la arquitectura, que incluyen todos los componentes mencionados en la infraestructura a utilizar. Donde principalmente se identifican 3 componentes que son:

- Cluster de servidores hyperconvergentes.
- VM con RancherOS donde se alojan todos los contenedores.
- Los contenedores en las que se tienen las diferentes aplicaciones y servicios.

Es importante mencionar que el servicio web de toda la solución tecnológica está publicada en la dirección IP pública 207.249.157.246, no se cuenta con un dns donde estén los registros a utilizar por lo que es necesario agregar en el archivo host los registros:

207.249.157.246	kibana.mtiemarco.net
207.249.157.246	monitor.mtiemarco.net
207.249.157.246	portal.mtiemarco.net
207.249.157.246	inicio.mtiemarco.net

207.249.157.246



INTERNET

Cluster Servers Hyperconvergentes



VM Rancher OS

CONTENEDORES



MTIE-MONGODB



MTIE-MYSQL



MTIE-MONSTACHE



MTIE-LOGSTASH



MTIE-ELASTICSEARCH



REVERSE-PROXY



MTIE-INICIO



MTIE-KIBANA



MONITOR



MTIE-PORTAL

Ilustración 5 Diagrama de Arquitectura

Mapeo de los componentes en los dominios de TI.

Para una completa arquitectura orientada a servicios (SOA) de Tecnologías de Información en las empresas, se puede tomar como referencia el enfoque de TOGAF el cual nos marca que se tienen 7 dominios los cuales tienen que ir alineados con las necesidades del negocio/empresa.

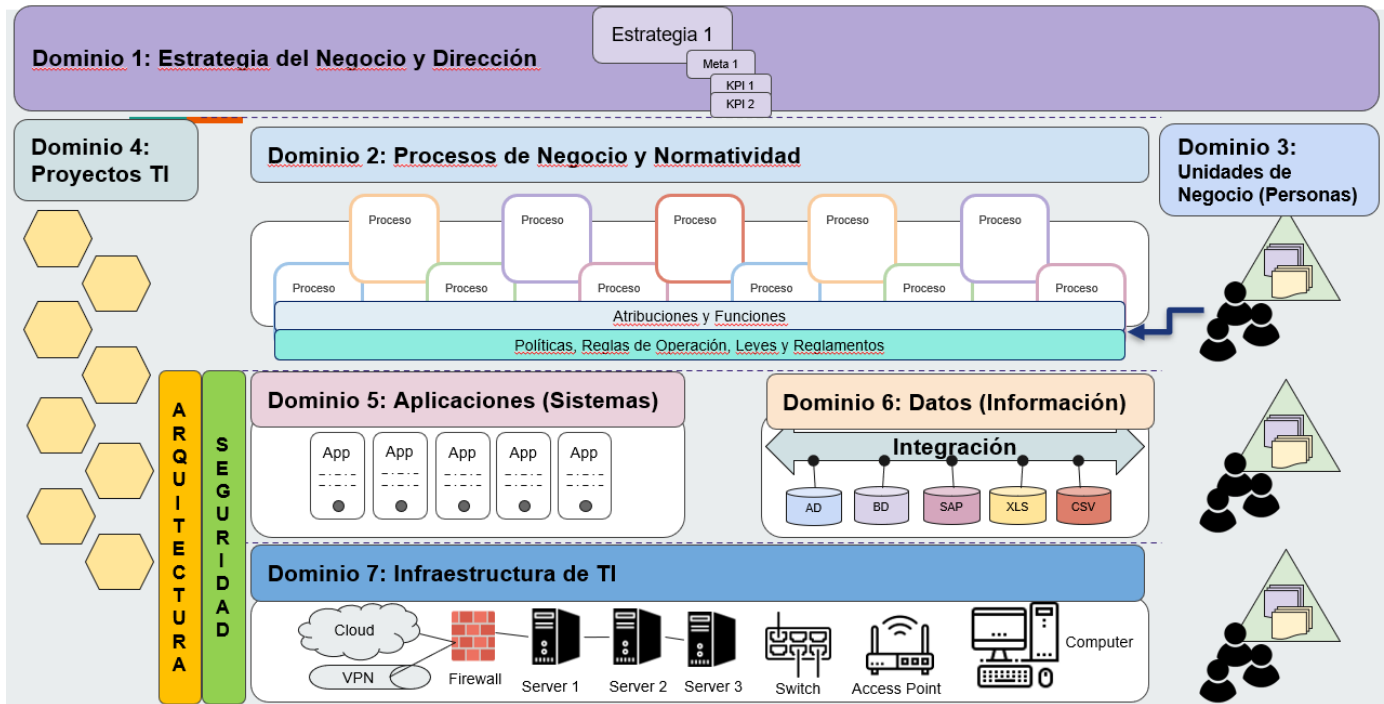


Ilustración 6 Dominios de Tecnologías de Información en las Empresas

Para la presente solución tecnológica se pueden identificar todos los componentes dentro de cada uno de los dominios como a continuación se describen:

- **Dominio 7. Infraestructura de TI.**- En este dominio encontramos a todo lo que se utilizó para montar dicha solución, tales como los servidores hyperconvergentes, la máquina virtual generada con RancherOS, así como cada uno de los contenedores desplegados.
- **Dominio 6. Datos (Información).**- En este dominio podemos encontrar a las 2 bases de datos que se tienen actualmente tanto SQL como no relacional, donde se tiene la información de los alumnos, docentes y el uso de Microsoft teams.
- **Dominio 5. Aplicaciones (Sistemas).**- En este dominio podemos mapear a las aplicaciones web como lo son los portales de inicio, el de pruebas, el monitoreo de la infraestructura, así como la aplicación de Kibana, que es donde se desplegaron finalmente los tableros para BI.
- **Dominio 4. Proyectos TI.**- En este dominio se engloba la totalidad de la solución, debido a que se integra toda la infraestructura, aplicaciones y datos, pero también se tiene una interacción con la alta dirección y todas las áreas involucradas para poder llegar al objetivo principal de contar con una solución integral.
- **Dominio 3. Unidades de negocio.**- En este dominio identificamos a todas las “personas” o unidades operativas que intervienen en mantener la información necesaria para los datos, dichas personas son las que nos van a estar ayudando a la identificación de todos los KPIs y resultados, en este caso

podemos identificar a las áreas Imagen y Comunicación, Servicios Escolares, así como todas las direcciones de las diferentes facultades.

- **Dominio 2.** Procesos de Negocio y Normatividad.- En este dominio se pueden identificar todos los procesos que son necesarios para la continuidad del negocio, así como generar e identificar la cadena de valor, es por ello que la solución tecnológica esta basada en los principales procesos que generan valor y utilidad a la Universidad, que son los alumnos y el desempeño de los mismos, así como de los docentes.
- **Dominio 1.** Estrategia del Negocio y Dirección. Por último y no menos importante identificamos que todos los dominios antes mencionados deben de ir alineados a la estrategia de lo que la alta dirección o en nuestro caso Rectoría y Vicerrectoría nos dicten como prioridades; y mediante esta pequeña solución tecnológica de inteligencia de negocio, se puede dar valor a toda la organización y cumplir con los objetivos planteados en una planeación estratégica que se revisa continuamente y que implica el trabajo en conjunto de todas las direcciones de la Universidad.

Procedimiento de despliegue

1. Despliegue de la infraestructura con Terraform

Como ya se ha mencionado, el despliegue se genera automáticamente utilizando la herramienta de Terraform la cuál se encarga de generar la VM en base a la plantilla preconfigurada en Vsphere.

Al terminar de generar dicha VM, se customiza colocándole el nombre del host, se le cambia la dirección IP definitiva, al terminar de realizar esos cambios, se conecta con la nueva IP para continuar con el despliegue de los contenedores.

Al conectarse por SSH lo primero es descargar el repositorio cargado en github y realiza todo lo necesario de crear y mover carpetas necesarias, para al final ejecutar el Docker-compose y así desplegar todos los contenedores.

Lo que ejecutaremos en Terraform será:

```
terraform init
terraform fmt
terraform validate
terraform plan
terraform apply -auto-approve
```

El script de Terraform utilizado es el siguiente:

```
#####
# Define Variable
#####
variable "vsphere_user" {}
variable "vsphere_password" {}
variable "vsphere_server" {}

#####
# Provider section
#####
provider "vsphere" {
```

```

user          = var.vsphere_user
password      = var.vsphere_password
vsphere_server = var.vsphere_server

# If you have a self-signed cert
allow_unverified_ssl = true
}

#####
#Capturing the data from vsphere
#####
data "vsphere_datacenter" "dc" {
  name = "DC Campestre"
}

data "vsphere_datastore" "datastore" {
  name          = "vsanDatastore"
  datacenter_id = data.vsphere_datacenter.dc.id
}

#####
data "vsphere_resource_pool" "pool" {
  name          = "GPO_REC"
  datacenter_id = data.vsphere_datacenter.dc.id
}

#####
data "vsphere_compute_cluster" "cluster" {
  name          = "CLUSTAVCENTER"
  datacenter_id = data.vsphere_datacenter.dc.id
}

data "vsphere_network" "network" {
  name          = "Servidores95"
  datacenter_id = data.vsphere_datacenter.dc.id
}

#####
# sourcing template
#####
data "vsphere_virtual_machine" "template" {
  name          = "RancherOS_Plantilla"
  datacenter_id = data.vsphere_datacenter.dc.id
}

#####
# Resource
#####
resource "vsphere_virtual_machine" "vm" {
  name          = "RancherOS-MTIE-ProyFinal"
  folder        = "MTIE_MARCO"
  resource_pool_id = data.vsphere_resource_pool.pool.id
  datastore_id   = data.vsphere_datastore.datastore.id

  num_cpus = 4
  memory   = 8128
  guest_id = "otherLinux64Guest"

  network_interface {
    network_id = data.vsphere_network.network.id
  }
}

```

```

disk {
  label = "disk1"
  size  = 20
}

#####
# Initiate the clone
#####
clone {
  template_uuid = data.vsphere_virtual_machine.template.id
}

provisioner "remote-exec" {
  inline = [
    "cd /home/docker",
    "sudo ros config set rancher.network.interfaces.eth0.address 172.17.94.126/23",
    "sudo ros config set hostname RancherOSProyFinal",
    "sudo curl -L https://github.com/docker/compose/releases/download/1.27.4/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose",
    "sudo chmod +x /usr/local/bin/docker-compose",
    "alias git='docker run -ti --rm -v $(pwd):/git bwits/docker-git-alpine'",
    "git help",
    "sudo system-docker restart network",
  ]
  connection {
    host      = "172.17.94.121"
    type      = "ssh"
    user      = "rancher"
    password  = "P4ssw0rd"
  }
}

provisioner "remote-exec" {
  inline = [
    "cd /home/docker",
    "alias git='docker run -ti --rm -v $(pwd):/git bwits/docker-git-alpine'",
    "git help",
    "sudo mkdir -p volumes/mongodb/data/db",
    "sudo mkdir -p volumes/elk-stack/elasticsearch",
    "sudo mkdir -p monstache",
    "cd volumes/elk-stack/",
    "sudo chmod 777 elasticsearch/",
    "cd /home/docker",
    "git clone https://github.com/tonydumatrix/MTIE-Proy3.git && cd MTIE-Proy3/sync-elasticsearch-mysql",
    "sudo mv Dockerfile-logstash /home/docker/",
    "sudo mv data /home/docker/",
    "cd volumes",
    "sudo mv logstash/ /home/docker/volumes/",
    "cd /home/docker/MTIE-Proy3",
    "sudo mv docker-compose.yml ../",
    "sudo mv monstache.config.toml /home/docker/monstache",
    "sudo cp sysctl.conf /etc/sysctl.conf",
    "cd /home/docker/MTIE-Proy3",
    "sudo mv startbootstrap-heroic-features ../",
    "sudo mv startbootstrap-one-page-wonder ../",
    "sudo docker-compose up -d",
  ]
  connection {
    host      = "172.17.94.126"
    type      = "ssh"
  }
}

```

```

    user      = "rancher"
    password  = "P4ssw0rd"

  }
}
}
#####
# Output
#####

```

2. Conexión al RancherOS para preparar MongoDB

Nos conectamos al contenedor de MongoDB para inicializar el replicaset y generar la base de datos **dbdata**, crear el usuario y asignar permisos al mismo, instalamos algunos paquetes necesarios para copiar y editar archivos dentro de Mongo, se copian los archivos csv al contenedor, para terminar importando dichos registros de los archivos csv a cada una de las colecciones. Para ello ejecutamos lo siguiente:

```

** Vamos a la ruta de docker
cd /home/docker

** Entramos al contenedor de docker y copiamos el archivo de la bd al mongo
docker exec -it MTIE-MongoDB bash

** Instalamos paquetes necesarios para copiar y editar archivos dentro de contenedor Mongo
apt-get -y update
apt-get -y install vim
apt-get -y install openssh-client

** Copiamos el archivo con la bd docentes y teams al contenedor Mongo
scp -r rancher@172.19.0.1:/home/docker/MTIE-Proy3/bases/docentestotal2.csv /usr/src/docentestotal2.csv
scp -r rancher@172.19.0.1:/home/docker/MTIE-Proy3/bases/teams.csv /usr/src/teams.csv

** Entramos a Mongo para iniciar replicaset y verificamos status,
** el cluster se llama rs0 como lo hicimos en docker-compose

mongo -u root -p
** el password es mongoadmin ***

rs.initiate();
rs.status();

** generamos la nueva BD llamada dbdata y creamos usuario dbo-operator y
** se asignan los roles de dicho usuario
use dbdata
db.createUser({
  user: 'dbo-operator',
  pwd: 'operadoradmin',
  roles: [
    { role: 'readWrite', db: 'dbdata' },
    { role: 'read', db: 'local' },
    { role: "dbAdmin", db: "dbdata" },
    { role: 'root', db: 'admin' }
  ]
});

** se puede corroborar el usuario recién creado****

```

```

db.getUsers();

** salimos del motor de la base de datos, y salimos del contenedor de mongo**
exit

** importamos los registros de archivos csv a las colecciones
mongoimport -u root -p mongoadmin --authenticationDatabase admin --db dbdata --collection docentes --type csv
--columnsHaveTypes --fields
"campus.string(), facultad.string(), programa.string(), nivel.string(), folio.string(), paterno.string(), materno.s
tring(), nombre.string(), correo.string(), grupo.string(), clavemat.string(), materia.string(), hrssemana.string(),
dia.string(), corte.date(2006-01-02)" --file /usr/src/docentestotal2.csv
mongoimport -u root -p mongoadmin --authenticationDatabase admin --db dbdata --collection teams --type csv --
columnsHaveTypes --fields
"nombre.string(), correo.string(), mensajeschat.int32(), reunionesorganizadas.int32(), reunionesparticipadas.int3
2(), llamadas.int32(), audiotime.int32(), videotime.int32(), screentime.int32(), corte.date(2006-01-02)" --file
/usr/src/teams.csv

** salimos del contenedor y podemos corroborar que todos los contenedores estén arriba
exit
docker ps -a

```

3. Importar los objetos a Kibana

Ya que se tiene toda la infraestructura desplegada, lo último en realizar es importar todos los objetos pre-creados en Kibana, como los son índices, visualizaciones así como los dashboards para con ello tener toda la solución tecnológica completa.

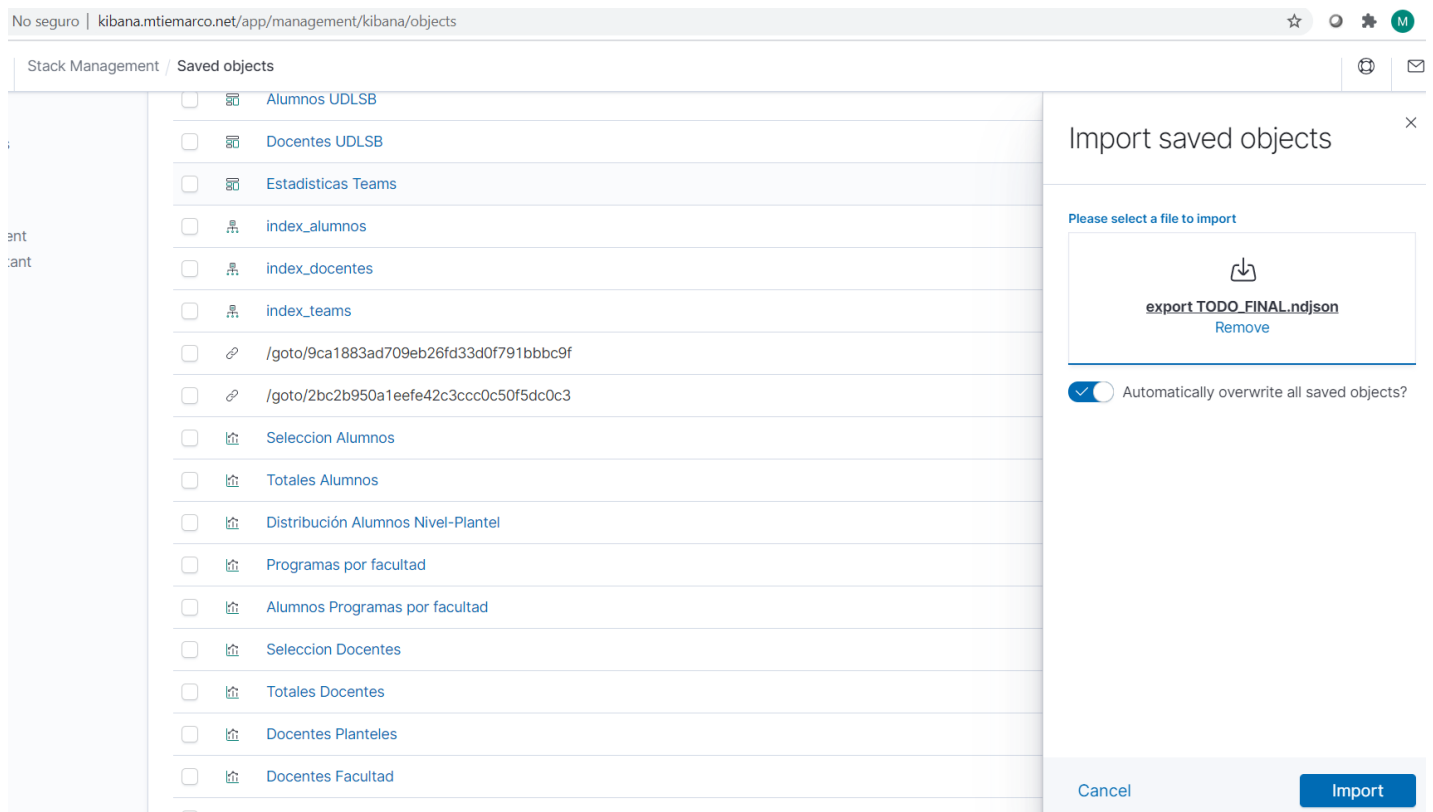


Ilustración 7 Importación de objetos en Kibana

Dashboards Kibana

Al finalizar todo el despliegue y configuración de toda la solución tecnológica, obtenemos como resultado y tablero para el análisis de los datos en Kibana, en el cuál encontramos 3 diferentes dashboards: Alumnos UDLSB, Docentes UDLSB y Estadísticas Teams.

- **Alumnos UDLSB.**– Cantidad de alumnos vigentes de la Universidad de la Salle Bajío, clasificados por plantel, facultad, programa, nivel académico.

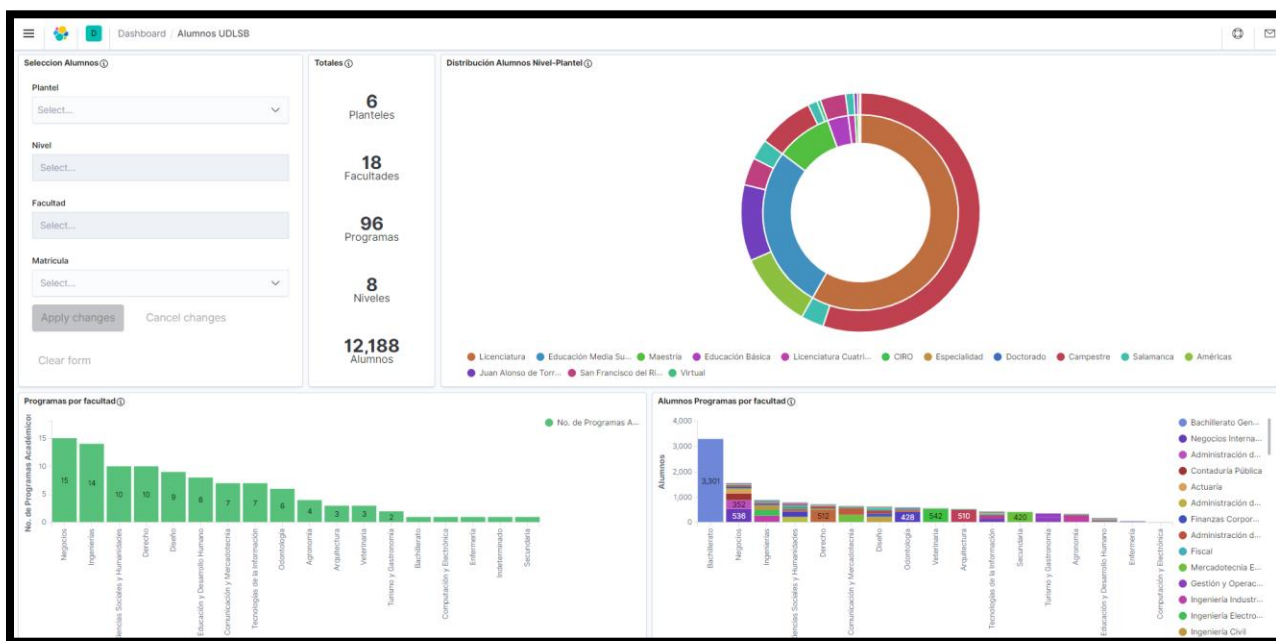


Ilustración 8 Dashboard de alumnos UDLSB

- **Docentes UDLSB.**– Cantidad de docentes vigentes de la Universidad de la Salle Bajío, clasificados por plantel, facultad, programa, nivel académico, cantidad de materias y grupos.

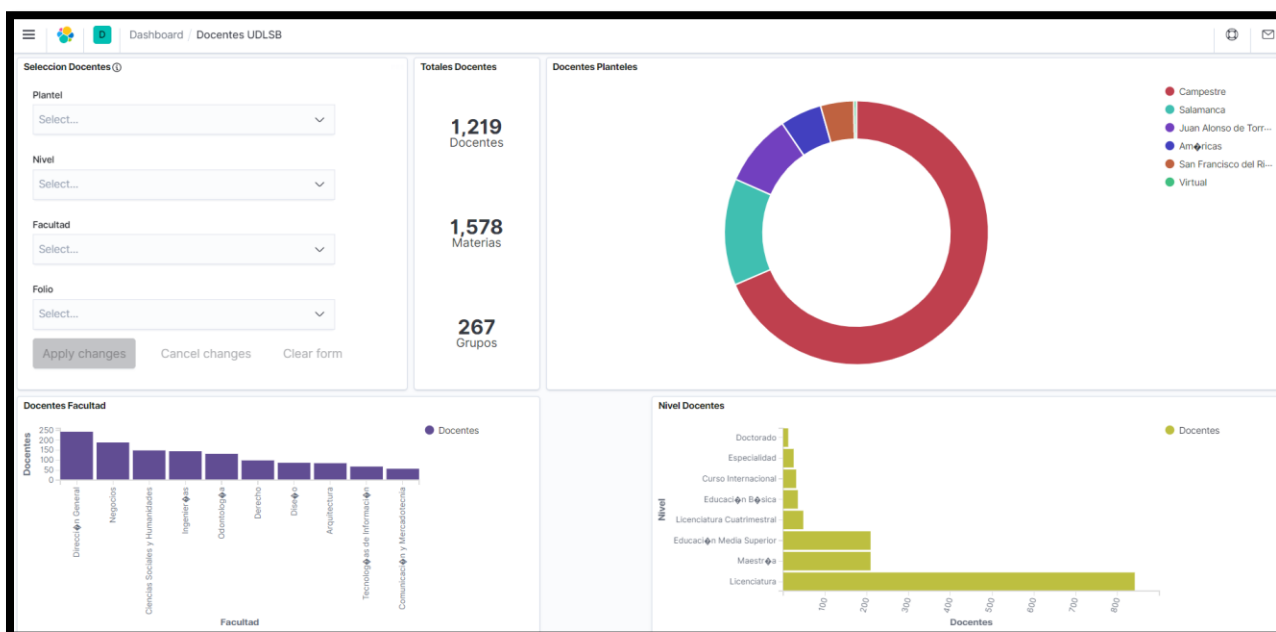


Ilustración 9 Dashboard de docentes UDLSB

- **Estadísticas Teams.**– Corresponde a las estadísticas e indicadores del uso de la plataforma de Teams por semana, incluyendo minutos de audio, video, pantalla compartida, cantidad de reuniones, chat, que corresponden tanto a alumnos como docentes de la Universidad.



Ilustración 10 Dashboard de estadísticas Teams

CONCLUSIONES

A lo largo de todo el cuatrimestre se estuvieron conociendo y aprendiendo diferentes temas y herramientas que como finalidad principal es la de crear una Arquitectura Orientada a Servicios; en este proyecto final es lo que se llevó a cabo, se generó una solución tecnológica integral donde se implementaron diferentes herramientas, se generaron diferentes scripts con la finalidad de eficientar el despliegue total de dicha solución; si bien es cierto que preparar todo ese despliegue puede llevarse demasiadas horas-días en prepararlo, se cumplió con la finalidad de que, en un tiempo aproximado de **6 minutos** se genere desde cero toda la infraestructura automatizada, y al final tener el resultado de la solución tecnológica completamente desplegada; esto brinda muchísimas bondades a los ambientes productivos, ya que en un tiempo mínimo se puede generar lo que manualmente se puede hacer en días. Este proyecto fue un buen reto y a manera personal lo traté de llevar con la infraestructura y medios que actualmente cuento en mi lugar de trabajo, pensando en desde ahora comenzar a aprovechar todo lo que se trabajó y aprendió y llevarlo a producción.

Si bien el ritmo de la materia fue algo acelerado, eso sirvió para poder abarcar mayor cantidad de temas, y aun así hubo algunos temas de los que no se pudo trabajar en clase, pero todo ello ya nos toca a cada uno de nosotros seguir indagando sobre dichos temas. En general todo lo visto en la materia y aprendido del excelente maestro, puede darnos mayor valor curricular, y poder seguir creciendo profesionalmente. Te agradezco de manera personal Peimbert el que compartas tus conocimientos con nosotros. Hasta la próxima.

REFERENCIAS

<https://docs.mongodb.com/v4.0/reference/program/mongoimport/#example-csv-import-types>

<https://registry.terraform.io/modules/Terraform-VMWare-Modules/vm/vsphere/latest>

<https://steemit.com/rancheros/@ynoot/rancheros-installation>

<https://forums.rancher.com/>

<https://registry.terraform.io/providers/hashicorp/vsphere/latest/docs>

<https://github.com/rancher/os/releases/>

<https://www.elastic.co/es/elastic-stack>

https://linuxhint.com/setup_docker_machine_vmware/

<https://sdbrett.com/BrettsITBlog/2017/01/rancheros-installing-to-hard-disk/>

<https://rancher.com/docs/os/v1.x/en/about/custom-partition-layout/>

<https://www.youtube.com/watch?v=aTeE58rQd9A>

<https://rancher.com/docs/rancher/v1.6/en/cli/commands/>

<https://rancher.com/docs/os/v1.x/en/configuration/adding-kernel-parameters/>