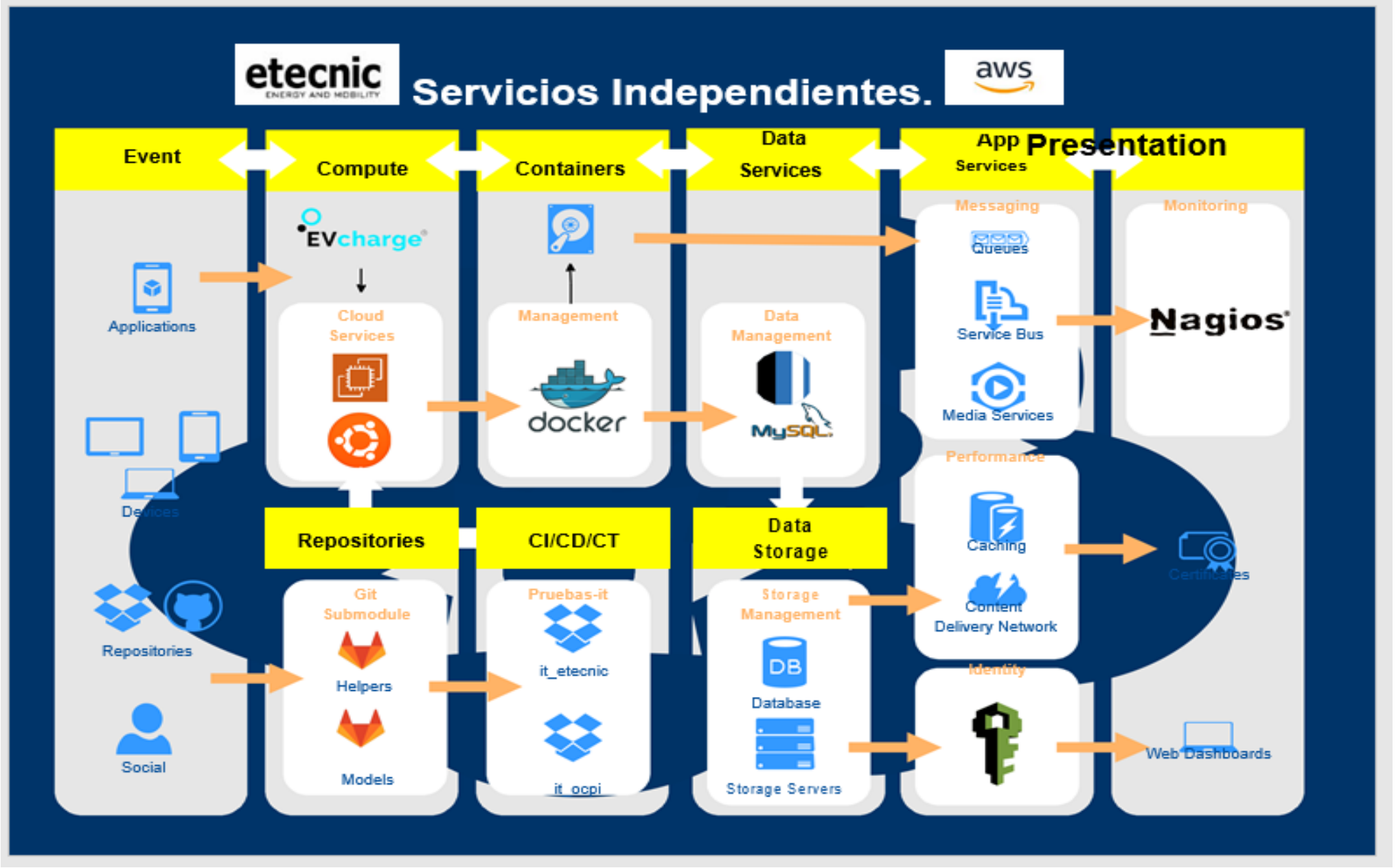
Monitoreo de servicios y conexiones.



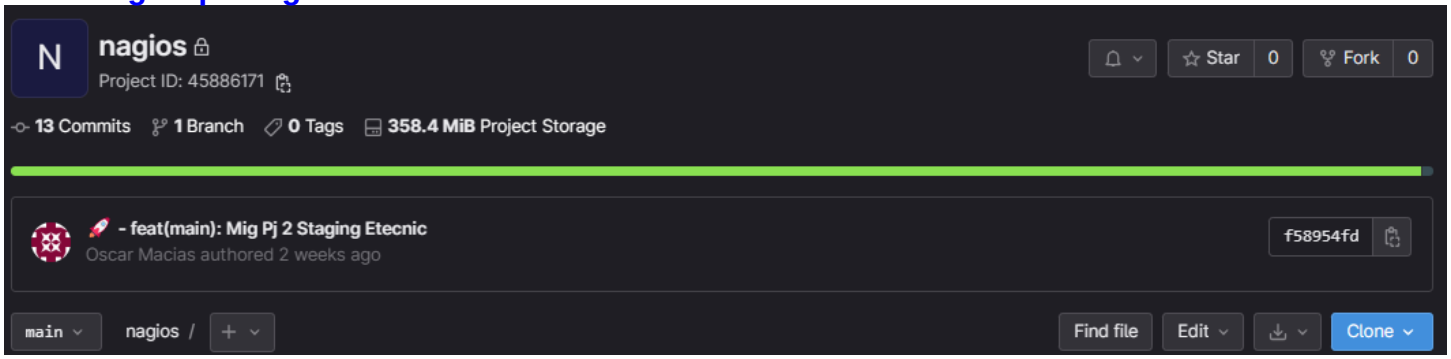
Realizado por Oscar Macias.
Fecha: 22/06/2023

Implementación de Docker.

1. Acceso del usuario a los repositorios locales.
   Requisitos:
   - Cuenta GitLab.
   - Configuración de acceso SSH por Token.
   - Acceso a lectura de repositorios por proyecto.

2. Lanzamientos de cambios sobre los repositorios git.
   - Acceso de escritura sobre repositorios nuevos por proyecto.

3. Acceso al repositorio global:
   etecnic-group / nagios · GitLab



4. Clonamos el proyecto que contiene el contenedor.

Requisitos:

- Instalar Docker en servidor Ubuntu.

```
$ for pkg in docker.io docker-doc docker-compose podman-docker containerd runc; do sudo apt-get remove $pkg; done
$ sudo apt-get update
$ sudo apt-get install ca-certificates curl gnupg
$ sudo install -m 0755 -d /etc/apt/keyrings
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
$ sudo chmod a+r /etc/apt/keyrings/docker.gpg


$ echo \
"deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu \
"$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null


$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
$ sudo docker run hello-world


$ git clone --recursive git@gitlab.com:etecnic-group/nagios.git
$ ll
```

5. Construimos el contenedor.

```
$ docker compose -f "./nagios/Docker/docker-compose.yml" up -d --build
```

docker-compose.yml

```
version: '3'

services:

  nagios:
    build:
      context: .
      dockerfile: ./dockerfile
    ports:
      - "8080:80"
    tty: true
```

dockerfile

```
FROM ubuntu:lunar-20230522

COPY . .

ENTRYPOINT bash ./install.sh && tail -f /dev/null
```

install.sh

```
#!/bin/sh

yes | apt update --fix-missing;
        yes | apt upgrade;
        yes | apt autoremove;
        yes | apt install autoconf \
                        bc \
                        gawk \
                        dc \
                        build-essential \
                        gcc \
                        libc6 \
                        make \
                        unzip \
                        libgd-dev \
                        libmcrypt-dev \
                        libssl-dev \
                        snmp \
                        libnet-snmp-perl \
                        gettext \
                        nano \
                        nmap \
                        traceroute \
                        net-tools \
                        apache2 \
                        wget \
                        sudo;

exit
```

6. Iniciamos el contenedor.

```
$ docker run --rm -it docker-nagios:latest

$ docker ps -a | grep nagios
```

- Instalar Nagios en Docker sobre el servidor Ubuntu.

7. Iniciamos el contenedor.

```
$ docker exec -it 1d0ck3rc0n741n3r sh
```

8. Instalación del paquete del servicio.

nagios.sh

```
#!/bin/bash/usr env

yes | apt upgrade;
    yes | apt update;
    yes | apt autoremove;
    yes | apt install php \
        libapache2-mod-php;

wget https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.6.tar.gz;
    tar -xf nagios-4.4.6.tar.gz;
    cd nagioscore-nagios-4.4.6/ || exit;
    ./configure --with-httpd-conf=/etc/apache2/sites-enabled;
    make all;
    make install-groups-users;
    usermod -a -G nagios www-data;
    make install;
    make install-daemoninit;
    make install-commandmode;
    make install-config;
    make install-webconf;
    a2enmod rewrite cgi;
    service apache2 restart;
    htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin;
```

- Configurar Nagios en Docker sobre el servidor Ubuntu.

9. Instalación de los plugins del servicio.

nagios.sh

```
yes | apt install monitoring-plugins nagios-nrpe-plugin;
```

10. Configuración del servicio nagios.

nagios.sh

```
mkdir -p /usr/local/nagios/etc/servers;
nano /usr/local/nagios/etc/nagios.cfg
cfg_file=/usr/local/nagios/etc/objects/linux.cfg
cfg_dir=/usr/local/nagios/etc/servers
```

11. Configuración de la ruta de los plugins de Nagios.

nagios.sh

```
nano /usr/local/nagios/etc/resource.cfg
$USER1$=/usr/lib/nagios/plugins
```

12. Configuración de contacto de reportes en Nagios.

nagios.sh

```
nano /usr/local/nagios/etc/objects/contacts.cfg
    email                   software@etenic.es; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS ******
```

13. Configuración de comandos para monitoreo de umbrales en Nagios.

nagios.sh

```
# Commands.
nano /usr/local/nagios/etc/objects/commands.cfg
define command{
    command_name check_nrpe
    command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}

define command {
    command_name    check_traceroute
    command_line    /usr/sbin/traceroute -n -T -p $ARG2$ $HOSTADDRESS$
}
```

14. Configuración de tiempo de chequeo para cada Template según servidores en Nagios.

nagios.sh

```
# Templates.
nano /usr/local/nagios/etc/objects/templates.cfg
        check_interval   1
```

15. Configuración de umbrales para monitoreo según servidores y puertos en Nagios.

nagios.sh

```
# Host Services.
nano /usr/local/nagios/etc/objects/linux.cfg
```

linux.cfg

```
define host {
        use                     linux-server
        host_name               Cargador_Etenic
        alias                   Cliente
        address                 DIRECCION.IP.PUBLICA.CARGADOR
}

define service {
        use                     generic-service
        host_name               Cargador_Etenic
        service_description     Puerto_Direccion_IP_Publica_Cargador_Cliente
        check_command           check_tcp!PUERTO
}
```

16. Administración del servicio de Nagios dentro de Docker.

nagios.sh

```
nano /root/.bashrc
alias nagioscheck='/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg'
alias nagiosreload='service nagios restart'

su
source /root/.bashrc
nagioscheck
exit

service nagios start
service apache2 restart

su
nagiosreload
exit

exit
```

Uso de Nagios.

17. Accedemos al portal de Nagios establecido para Staging Etecnic, con su respectivo puerto publicado.
http://168.119.156.192:8080