

Alta disponibilidad con sistemas de  
ficheros distribuidos

ANSIBLE



Álvaro Vaca Ferreras  
IES Gonzalo Nazareno

# Alta disponibilidad

Consiste en tener un sistema redundante que permita a uno o varios servicios seguir ejecutándose con normalidad en caso de ocurrir algún tipo de fallo.

Para conseguirlo, tratamos de eliminar todos los **SPOF** mediante redundancia a todos los niveles posibles.

- *Cluster*
- *Recurso*
- *Heartbeat*
- *Split brain*
- *Quorum*
- *Stonith*

# Ceph (I)

En ocasiones, las aplicaciones requieren almacenamiento compartido entre los nodos que la ejecutan, de manera que acudimos a los sistemas de ficheros distribuidos.

- **Basado en objetos binarios**
- **Escalabilidad**
- **Abstracción**
- **Redundancia de datos**
- **Journaling**
- **Necesidad de una red grande**
- **Relativamente complejo**
- **Única compatibilidad con Linux**

# Ceph (II)

El sistema de almacenamiento consta de una red de demonios que se ejecutan entre los diferentes nodos constituyentes del clúster:

- **Monitor (MONs)**: Gestionan el estado de los nodos del clúster.
- **Manager (MGRs)**: Gestionan la utilización del espacio, la carga del sistema y la utilización de los nodos.
- **Object Storage Devices (OSDs)**: Gestionan los archivos: son responsables del almacenamiento, el duplicado y la restauración de los datos.
- **Metadata (MDs)**: Almacenan metadatos por motivos de rendimiento: rutas de almacenamiento, sellos de tiempo, nombres de archivos...

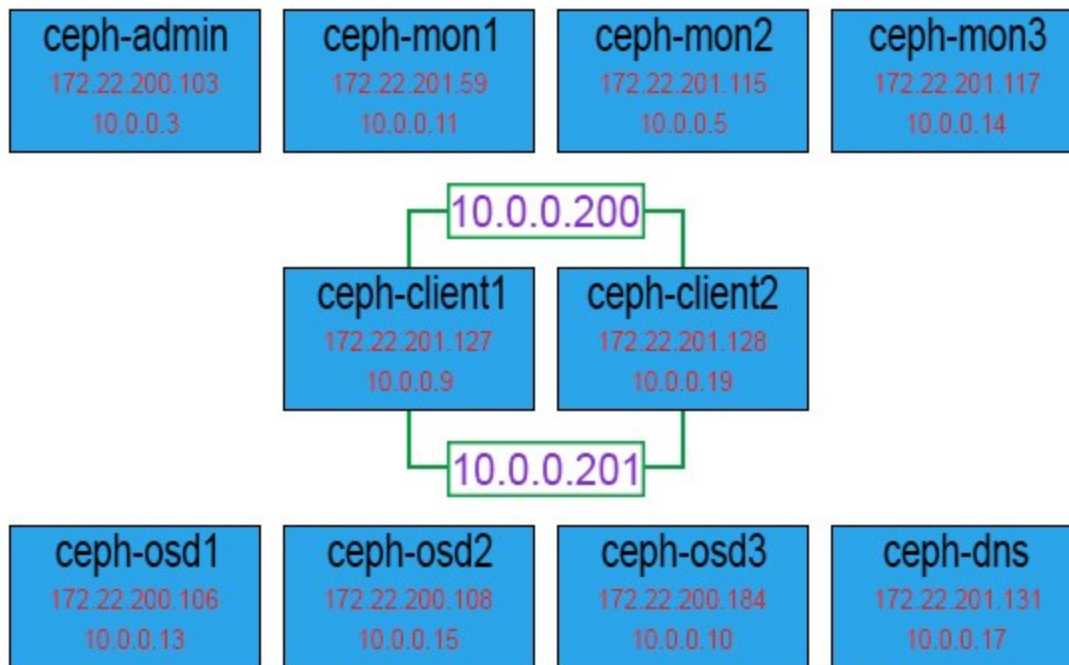
# Ansible

Configurar a mano las 10 máquinas del escenario sería una tarea larga y tediosa, así que haremos uso de **Ansible** para hacerlo más ameno.

Los proyectos suelen organizarse en forma de roles en los que se definen una serie de tareas ordenadas a realizar (**play**). Por último, se genera un **playbook** en el que se indicará la correspondencia entre los roles y las máquinas o grupos de máquinas del inventario a los que aplicar dichas tareas.

- **Gestión de servidores**
- **Gestión de configuraciones**
- **Gestión de aplicaciones**
- **Robustez**
- **Paralelismo**
- **Idempotencia**

# Escenario



# Configuración del escenario (I)

- **all:**
  - Ensure apt does not use debian replica
  - Ensure apt uses cica replica
  - Ensure system is updated
  - Set timezone to Europe/Madrid
  - Ensure hosts file is not managed by cloud\_init
  - Ensure hosts file does not resolve hostname
  - Add DNS server to resolvconf configuration
  - Add search pattern to resolvconf configuration
  - restart resolvconf
- **dns:**
  - Ensure bind9 is installed
  - Copy named.conf.options file
  - Copy named.conf.local file
  - Copy db.example.com zone using template
  - restart bind9
- **admin, osd, mon, client:**
  - Create cephuser user
  - Allow cephuser to execute sudo commands without password
  - Create .ssh structure
  - Copy ansible\_rsa private key
  - Copy ansible\_rsa public key
  - Copy authorized\_keys file
  - Check if known\_hosts file exists
  - Scan and add SSH keys of the machines to known\_hosts file
  - Change known\_hosts file owner
  - Add Ceph apt key
  - Add Ceph repository
  - Ensure needed packages are installed

# Configuración del escenario (II)

- `mkdir cephcluster && cd cephcluster/`
- `ceph-deploy new ceph-mon{1..3}`
- `echo "osd pool default size = 2" >> ceph.conf`
- `ceph-deploy install ceph-admin ceph-mon{1..3} ceph-osd{1..3} ceph-client{1..2}`
- `ceph-deploy mon create-initial`
- `ceph-deploy admin ceph-admin ceph-mon{1..3} ceph-osd{1..3} ceph-client{1..2}`
- `ceph-deploy mgr create ceph-mon{1..3}`
- `ceph-deploy osd create --data /dev/vdb ceph-osd1`
- `ceph-deploy osd create --data /dev/vdb ceph-osd2`
- `ceph-deploy osd create --data /dev/vdb ceph-osd3`
- `ceph-deploy mds create ceph-osd{1..3}`
- `ceph osd pool create cephfs_data 64`
- `ceph osd pool create cephfs_metadata 64`
- `ceph fs new cephfs cephfs_metadata cephfs_data`



# Configuración del escenario (III)

- **client:**

- Ensure needed packages are installed
- disable apache2
- disable mariadb
- Obtain secret key and save as a variable
- Store the secret key in admin.secret file
- Mount CephFS on /ceph and add to fstab
- Create the necessary structure on /ceph
- Check if /ceph/prestashop folder is empty before proceeding
- Download and unzip prestashop\_1.7.7.0.zip package
- Unzip prestashop.zip package
- Delete unnecessary PrestaShop files
- Copy prestashop.conf VirtualHost file
- Create prestashop.conf VirtualHost symlink
- Delete 000-default.conf VirtualHost symlink
- Create rewrite module symlink
- Change MariaDB datadir to /ceph/sql
- Change MariaDB bind-address to 0.0.0.0
- Check if /ceph/sql folder is empty before proceeding
- Install DB prerequisites
- Set hacluster user password
- Destroy default cluster
- Add hosts to the new cluster
- Create, start and enable the new cluster
- Disable STONITH property
- Create VirtualIP resource for Apache2
- Create Apache2 resource
- Create Apache2 and VirtualIP colocation
- Create VirtualIP resource for MariaDB
- Create MariaDB resource
- Create MariaDB and VirtualIP colocation
- Create constraint to preferably run all resources on the first node
- Create a new MariaDB database
- Create a new user with privileges in the previous database