

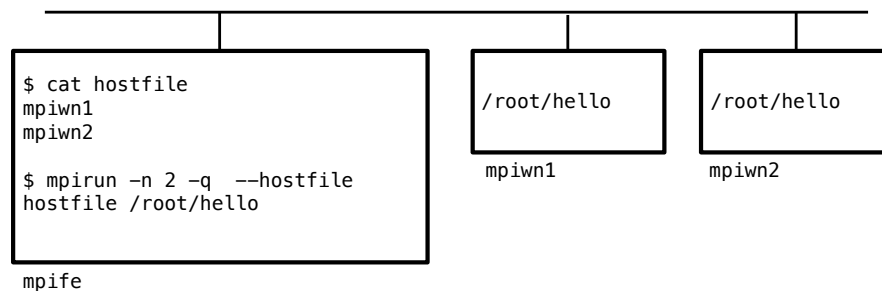
Plataformas de Gestión de Contenedores

Máster Universitario en Computación en la Nube y de Altas Prestaciones
Departamento de Sistemas Informáticos y Computación (DSIC)
2022-23 – Examen Práctico – 40% de la calificación total
Duración 2:30h



Nombre: _____

Queremos desplegar un clúster MPI sobre contenedores en Docker y Kubernetes, de forma que desde un contenedor podamos lanzar un programa MPI y que cada uno de los procesos MPI se ejecute en un contenedor diferente, según el esquema siguiente:



Para ello, se deben realizar los siguientes pasos:

- 1) Crear las Imágenes de Docker apropiadas.
- 2) Ejecutar una prueba con varios contenedores Docker en local.
- 3) Crear los recursos Kubernetes apropiados para ejecutarlo.

1. Crear las imágenes de Docker (25%)

Para instalar MPI, es necesario: seguir los siguientes pasos

- Necesitamos una imagen de Docker con un servidor sshd instalado (p.e. la imagen `rastasheep/ubuntu-sshd:18.04`). El servidor `sshd` se puede arrancar mediante el comando `/usr/sbin/sshd` utilizando la opción `-D` para que funcione en primer plano si fuera necesario), y configurar la conexión ssh sin contraseña con un par de claves, según los pasos siguientes (desde dentro del contenedor):
 - `ssh-keygen -q -N '' -f ~/.ssh/id_rsa`
 - `cp /root/.ssh/id_rsa.pub /root/.ssh/authorized_keys`
 - Comprobar: `ssh -i /root/.ssh/id_rsa localhost ls -l /`
- Después deberemos Instalar MPI en el contenedor con los siguientes comandos
 - `apt-get update`
 - `apt-get install -y openmpi-bin libopenmpi-dev`
- Finalmente registraremos la imagen del contenedor modificada como una imagen propia y la subiremos a Docker Hub

Referencia de la imagen modificada en Docker Hub

Contenido de la clave privada utilizada (/root/.ssh/id_rsa):

2. Ejecutar una prueba con varios contenedores Docker en local (30%)

Una vez disponemos de un contenedor Docker preparado para ejecutar MPI, procedemos a hacer una prueba ejecutando directamente tres contenedores en la máquina local y conectándolos de forma apropiada. Para ello, habrá que hacer los siguientes pasos:

- Desplegar un contenedor bajo el nombre “mpife”. Entrar en él, compilar el programa siguiente mediante el comando `mpicc -o hello hello.c`. Si necesitas un editor, lo puedes instalar con `apt-get install -y vim`.

```
#include <mpi.h>
#include <stdio.h>
#include <unistd.h>
int main(int argc, char **argv) {
    int rank;
    char hostname[256];
    MPI_Init(&argc,&argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    gethostname(hostname,255);
    printf("Hello world! I am process number: %d on host %s\n",
           rank, hostname);
    MPI_Finalize();
    return 0;
}
```

- Ejecutar el programa mediante `mpirun -n 2 --allow-run-as-root -q /root/hello`. Anotar el resultado en el cuadro que se indica (revisa que el directorio y el nombre del programa “hello” coincidan en tu caso).
-

Para poder ejecutar en paralelo, deberemos:

- Ejecutar dos contenedores más con los nombres `mpiwn1` y `mpiwn2`.
- Crear un fichero de “hostfile” en el contenedor `mpife` con las IPs locales de los contenedores `mpiwn1` y `mpiwn2`. Recuerda que las IPs se pueden obtener ejecutando el comando `ip addr` dentro de un contenedor. Si este comando no está disponible, se puede instalar con `apt-get install -y iproute2`. También las puedes obtener sin instalar ningún paquete inspeccionando la red correspondiente.
- Copiar el ejecutable en ambos contenedores (p.e. mediante `scp` o mediante `docker cp`) en la misma ubicación que en el contenedor `mpife`.
- Comprobar que desde el contenedor `mpife` se puede acceder sin contraseña a los contenedores `mpiwn1` y `mpiwn2` (necesario para que se incluyan sus IPs en el fichero `known_hosts`).
- Ejecutar mediante el comando `mpirun -n 8 -q --allow-run-as-root --machinefile hostfile /root/hello`.

Rellenar la siguiente información:

Comandos para la copia del fichero en los contenedores	
Contenido de hostfile	Resultado de la ejecución

3. Crear los recursos Kubernetes apropiados para ejecutarlo (30%)

En este último apartado crearemos los objetos apropiados para desplegar el clúster en un entorno Kubernetes. Por motivos de tiempo, haremos las siguientes simplificaciones:

- La configuración del fichero hostfile se hará a posteriori, una vez desplegado el clúster.
- El lanzamiento de proceso implica que el ejecutable esté en todos los contenedores, para lo que se realizará manualmente la copia del fichero en todos ellos.

Indicar en el siguiente cuadro el (o los) ficheros YAML necesarios para el despliegue:

Desplegar el cluster, entrar en el contenedor elegido como front-end, compilar y ejecutar el programa como se ha descrito en el paso anterior

Comandos para el lanzamiento de la aplicación en Kubernetes	
Contenido de hostfile	Resultado de la ejecución

4. Mejora de la configuración y acceso a los ejecutables en Kubernetes (15%)

En el apartado anterior se han introducido dos simplificaciones para facilitar su implementación:

- Rellenar el fichero hostfile a posteriori una vez desplegado el clúster.
- Copia manual del ejecutable en todos los contenedores.

Indica (sin que haga falta implementarlo) que soluciones te parecen adecuadas para que no haga falta copiar manualmente el ejecutable y para automatizar la creación del fichero hostfile. Indica los objetos Kubernetes que utilizarías, el orden de creación y qué acciones serían necesarias.