

Mini Elastic Container Service (ECS)

Sistemas Operativos

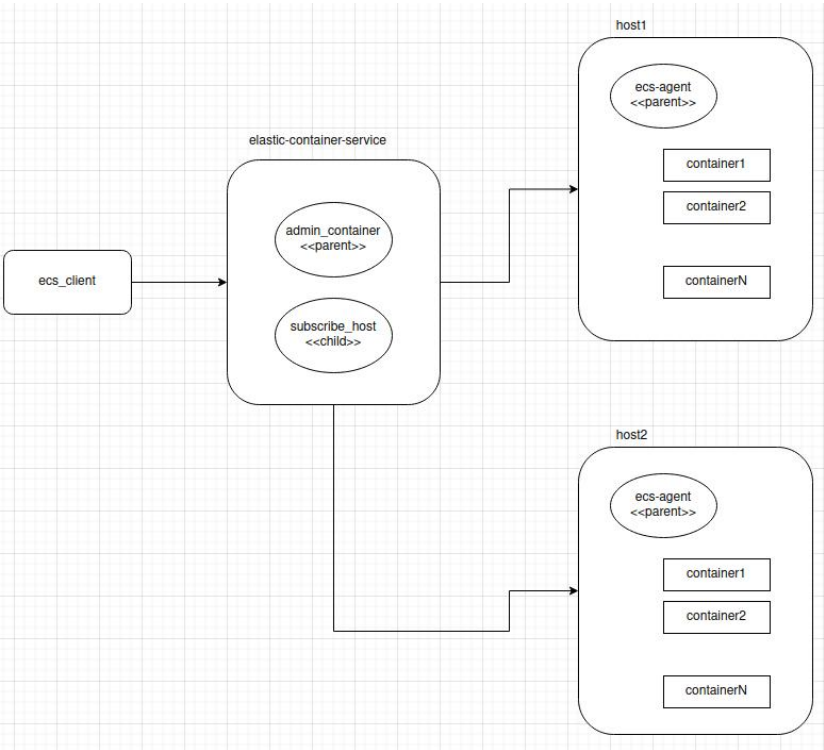
February 1, 2022

1 DESCRIPCIÓN DE LA PRIMERA PARTE DEL PROYECTO DE CLASE

El objetivo de este proyecto es diseñar y desarrollar un servicio de contenedores, que permita crear, administrar y monitorear contenedores docker de manera sencilla.

Para esto el proyecto estará dividido en varias etapas y en cada una de ellas se deberá ir agregando nuevas funcionalidades alineadas con los conceptos vistos en clase. Para esta primera parte se implementara la base del servicio de contenedores usando los conceptos de procesos, comunicación entre procesos y docker.

La arquitectura general del proyecto es la siguiente:



host El host es la maquina donde se crearan los contenedores, para efectos de prueba puede ser una maquina virtual o la maquina que tenga disponible el estudiante.

ecs-agent Este agente es el encargado de crear, listar, detener o borrar contenedores. Este proceso solo recibe instrucciones del elastic-container-service por medio del proceso `admin_container`. Como la creación de un contenedor no todo el tiempo es inmediata, el agente recibe la petición y responde con un status de recibido y ejecuta la acción.

Cuando un agente se inicia en un host, este debe reportarse enviando una petición al proceso `subscribe_host` del elastic-container-service.

elastic-container-service Este servicio es el encargado de recibir las peticiones del cliente y ejecutarlas. Este servicio constara de dos procesos, el `admin_container` y el `subscribe_host`, los cuales se crearan usando `fork`.

El `admin_container` es el proceso encargado de recibir las peticiones para manejar los contenedores (crear, listar, detener y borrar contenedores). El `subscribe_host`, es un proceso encargado de recibir peticiones desde los hosts y lleva un control de los hosts disponibles para crear contenedores. Solo los agentes que corren en el host interactuan con el proceso `subscribe_host`.

ecs-client El `ecs-client` es un SDK para python (interactua solo con el proceso `admin_container`) y ofrecerá la forma de enviar peticiones de administración a los contenedores. Las peticiones que puede enviar son las que expone el servicio de contenedores (elastic-container-service), las cuales solo serán:

- crear contenedor
- listar contenedores
- detener contenedor
- borrar contenedor

Al finalizar esta primera entrega se desea obtener:

- Una implementación funcional del sistema (El código deberá estar en GitHub o GitLab).
- Debe permitir validar la administración de distintos contenedores docker.
- Debe permitir correr dos agentes al mismo tiempo (así sea en la misma maquina) y que el `admin_container` decida cual usar.
- Debe poder demostrar el uso de los conceptos de procesos y comunicación entre procesos vistos en clase.

Sobre la entrega:

- Puede ser trabajado en parejas.
- Fecha máxima de entrega: Una semana después de terminado el tema de comunicación entre procesos