

1. DISTRIBUCIÓN

Podemos distinguir entre dos sistemas: distribuidos y en red.

1.1. SISTEMAS DISTRIBUIDOS VS SISTEMAS EN RED

- Sistemas distribuidos: es un conjunto de computadores separados físicamente y conectados entre sí por una red de comunicaciones distribuida; cada máquina posee sus componentes de hardware y software que el usuario percibe como un solo sistema.
- Sistemas en red: Es aquel que utiliza los recursos de una sola computadora central, es decir, su memoria, CPU, disco y periféricos. La computadora en sí misma puede controlar todos los periféricos directamente (si están físicamente conectados con la computadora central), o conectados a través de un servidor de terminal.

1.2. VENTAJAS DE LOS SISTEMAS DISTRIBUIDOS Y SISTEMAS EN RED

Ventajas de los sistemas distribuidos:

- Económicos.
- Trabajo en conjunto.
- Mayor confiabilidad.

Ventajas de los sistemas en red:

- Mejor aprovechamiento de los recursos individuales.
- Mayor poder de cómputo a menor precio.
- Menor redundancia.

1.3. CLOUD FOUNDRY COMO SISTEMA DISTRIBUIDO

Diego es un sistema distribuido que le permite ejecutar y escalar N cantidad de aplicaciones y tareas en contenedores a través de una cantidad de celdas. Aquí están las principales características y atributos de Diego:

- Es responsable de ejecutar y monitorear las imágenes compatibles con OCI, las aplicaciones independientes y las tareas implementadas en Cloud Foundry.
- Al residir en el núcleo de Cloud Foundry, Diego se encarga de la programación, ejecución y supervisión de tareas y procesos de larga ejecución (aplicaciones) que residen dentro de contenedores administrados.
- Es agnóstico tanto para la interacción del cliente como para la implementación en tiempo de ejecución.
- Asegura que las aplicaciones sigan ejecutándose al conciliar el estado deseado con el estado real mediante el establecimiento de una consistencia eventual, la recuperación automática y los ciclos de retroalimentación cerrados.
- Cuenta con un entorno de ejecución genérico compuesto de acciones y backends para permitir el soporte de múltiples cargas de trabajo diferentes basadas en Windows y Linux.

Cuando da un paso atrás y considera los desafíos inherentes con cualquier sistema distribuido, las soluciones a los desafíos de coherencia y orquestación proporcionados por Diego son extremadamente elegantes. Diego ha sido diseñado para hacer que el subsistema de tiempo de ejecución de contenedores de Cloud Foundry sea modular y genérico.

La mayoría de los usuarios de Cloud Foundry (por ejemplo, desarrolladores) no interactúan con Diego directamente. Los desarrolladores interactúan solo con la API de Cloud Foundry, conocida como CAPI. Sin embargo, comprender el tiempo de ejecución del contenedor de Diego es esencial para los Operadores de Plataforma porque, como operador, debe interactuar con Diego por consideraciones clave, como los requisitos de resistencia y la resolución de problemas de las aplicaciones.

1.3.1. LA IMPORTANCIA DE LA GESTION DE CONTENEDORES

Para que una imagen de contenedor (como una imagen compatible con OCI) se ejecute como un proceso aislado, se requiere una capa de administración de contenedor que pueda crear, ejecutar y administrar el proceso de contenedor. Como una API de administración de contenedores más generalizada, Cloud Foundry usa Garden para admitir una variedad de tecnologías de contenedores.

A través de Garden, Diego ahora puede admitir cualquier formato de imagen de contenedor que admita la API de Garden. Además, también ha agregado soporte para ejecutar contenedores en cualquier tecnología de contenedor basada en Garden, incluidos los backends de contenedor basados en Linux y Windows que implementan la API de Garden. [Ilustración 1](#) ilustra la capacidad de Diego para admitir múltiples artefactos de aplicación y formatos de imagen de contenedor.

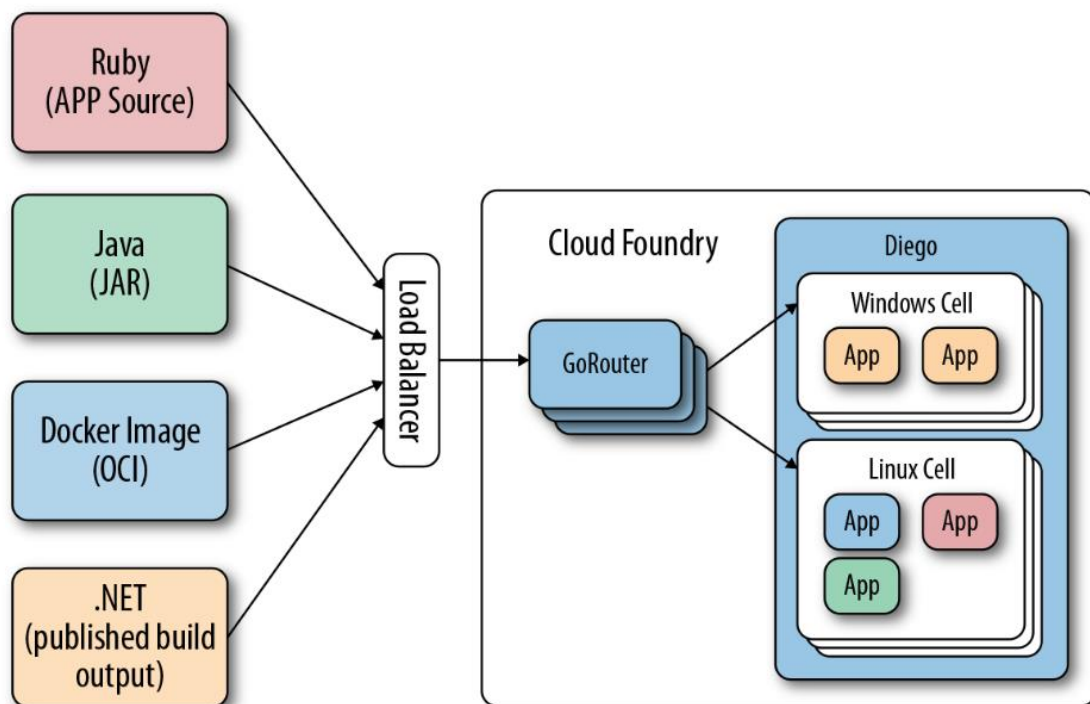


Ilustración 1

1.3.2. ARQUITECTURA DE DIEGO

Diego está compuesto por varios microservicios que residen en varios componentes, como puede verse en la [Ilustración 2](#).

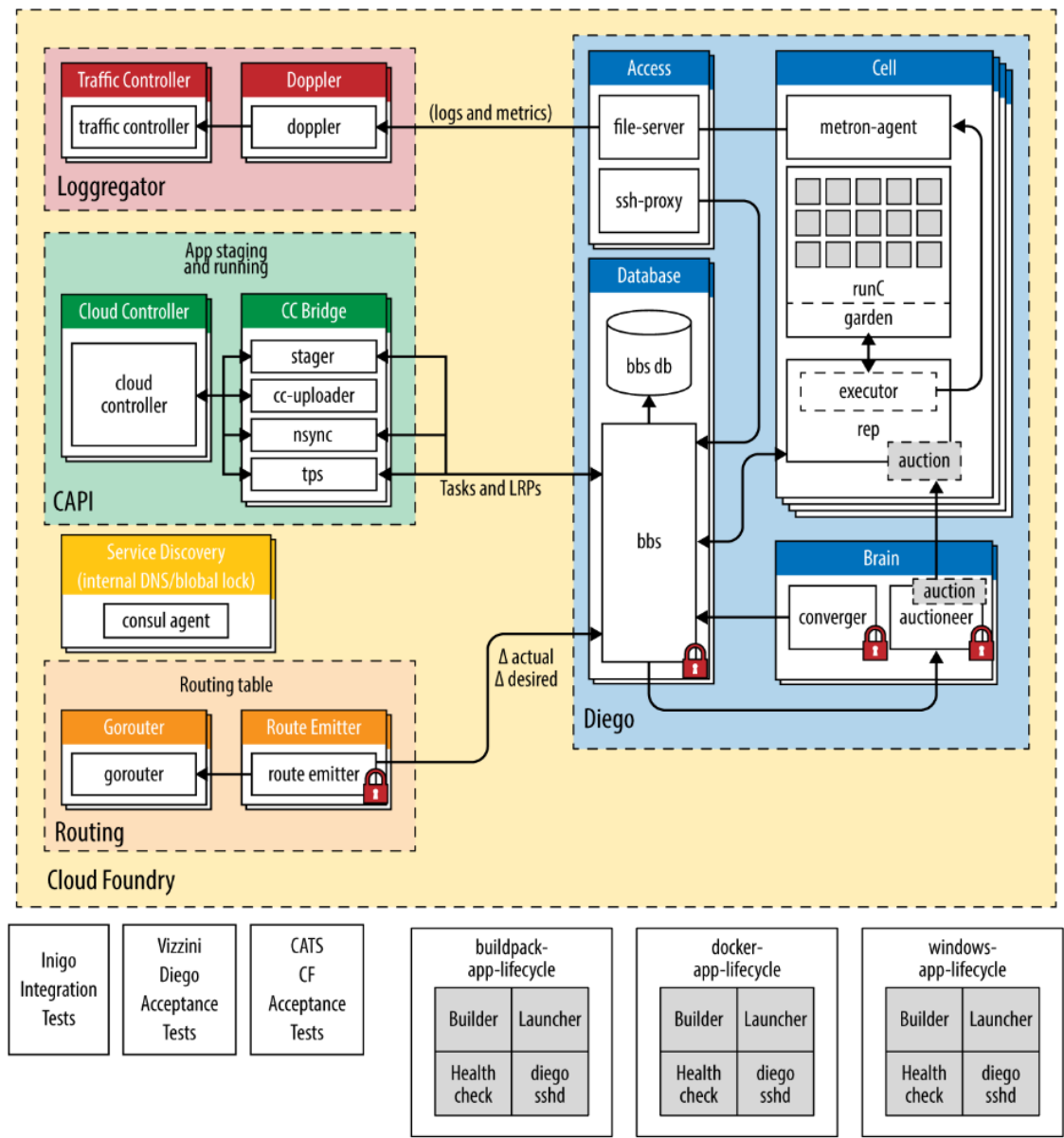


Ilustración 2: Componentes de Diego

Podemos agrupar los componentes, ampliamente, de la siguiente manera:

- Una capa de Cloud Foundry de componentes orientados al usuario (componentes con los que el usuario de la plataforma, interactuará directamente).
- La capa de Diego Container Runtime (componentes con los que interactúan los componentes centrales de Cloud Foundry).

1.3.3. INTERACTUANDO CON DIEGO

Los usuarios de Cloud Foundry no interactúan con Diego directamente; interactúan con los componentes orientados al usuario de Cloud Foundry, que luego interactúan con Diego en nombre del usuario. Aquí están los componentes orientados al usuario de Cloud Foundry que funcionan en conjunto con Diego:

- Componentes CAPI: la API de Cloud Foundry (Cloud Controller y el CC-Bridge).
- El sistema de registro definido por los agentes Loggregator y Metron.
- Enrutamiento (GoRouter, TCPRouter y el Emisor de ruta).

En conjunto, estos componentes de Cloud Foundry son responsables de lo siguiente:

- Política de aplicación.
- Cargar artefactos, droplets y metadatos de la aplicación en un blobstore.
- Tráfico de enrutamiento y manejo de tráfico de aplicaciones.
- Gestión de usuarios, incluida la interacción del usuario final a través de los comandos de API de Cloud Foundry.

Diego se conecta sin problemas a estos diferentes componentes de Cloud Foundry para ejecutar aplicaciones y tareas, enrutar el tráfico a sus aplicaciones y permitir la recuperación de los registros necesarios.