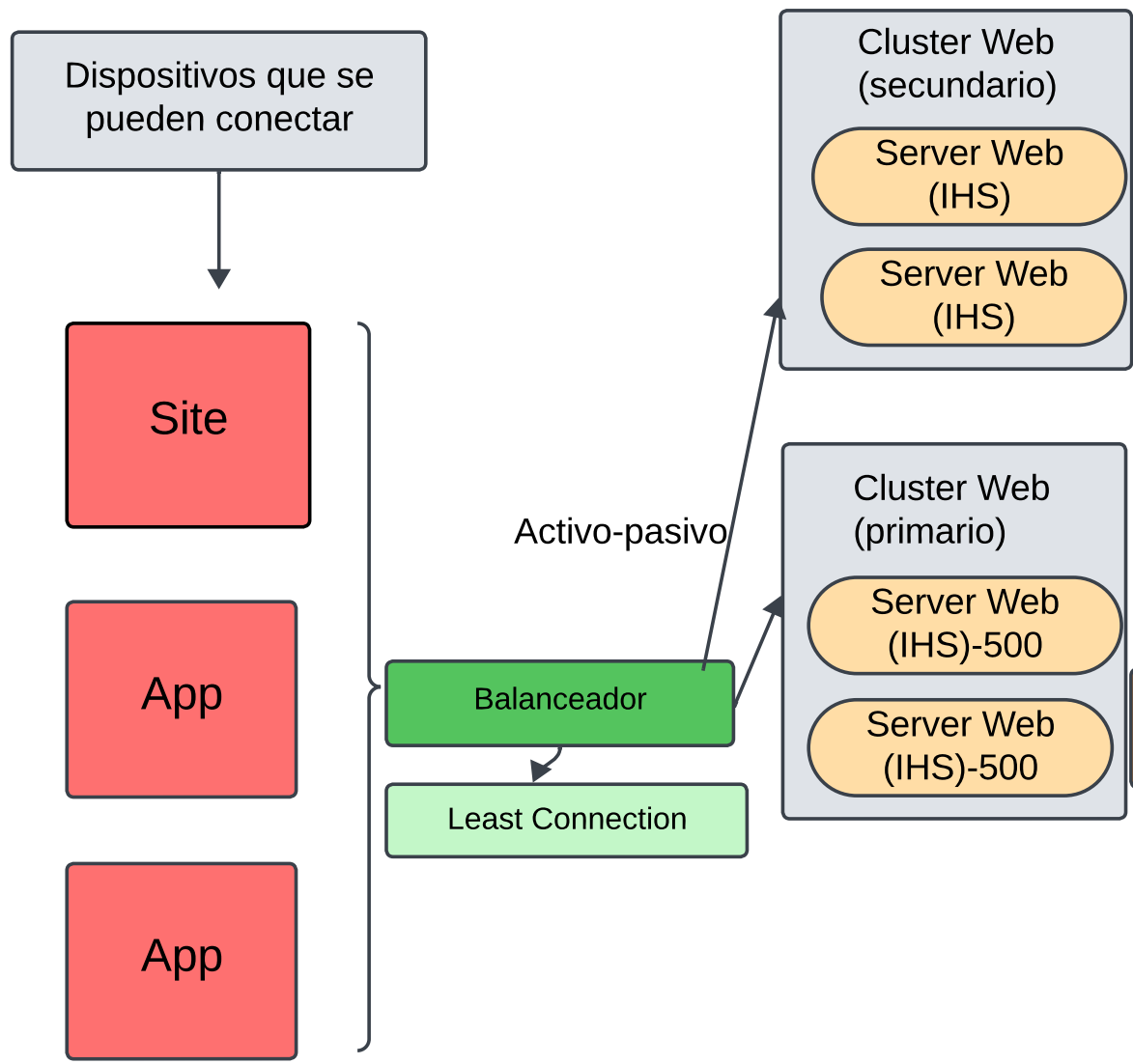
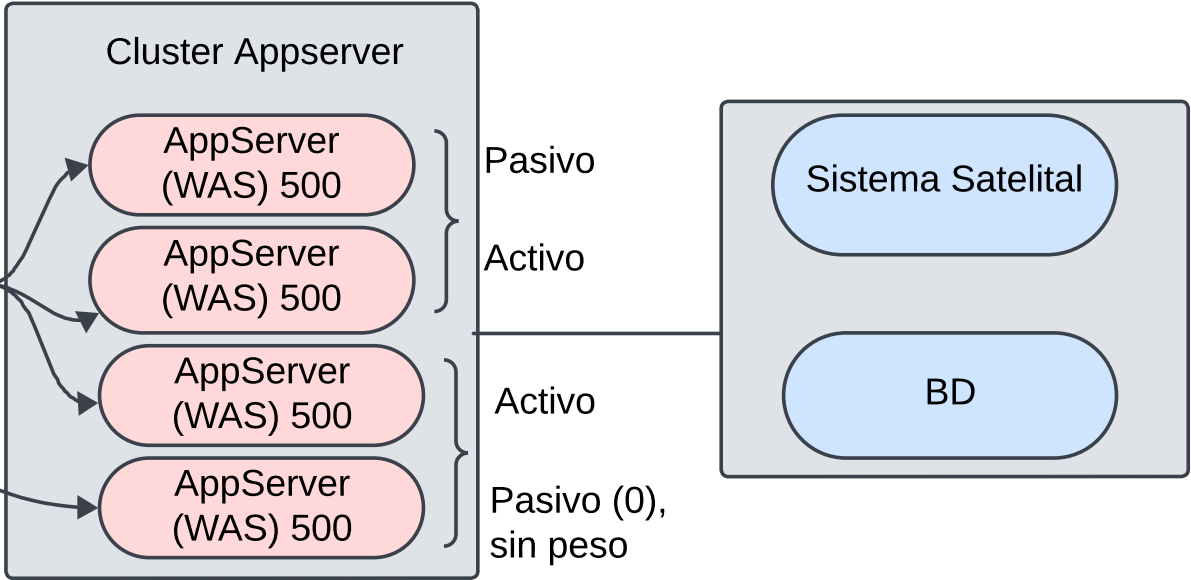


Escenario 1



Escenario 2

DMZ (porque va a estar expuesto el servicio y necesita estar protegido).



Escenario 1* : Este esquema esta pensando ante un disaster recovery, los cluster tienen la misma capacidad pero están dentro de un esquema failover, significa que yo tengo un activo-pasivo y que cuando no reaccione el cluster primario, se vaya al otro(secundario), esto ante el caso de que el licenciamiento nos permita tener un esquema de alta disponibilidad o esquema de Disaster Recovery. El cluster primario maneja la carga activa, y el cluster secundario asume el control solo si el cluster primario completo falla. Mientras tanto, si el cluster secundario falla, el cluster primario sigue operando normalmente, lo cual asegura que siempre tengas alta disponibilidad del servicio.

● Capa donde está expuesta el sitio de la página principal que va a ser nuestro servidor web y las conexiones concurrentes se van a distribuir entre dos servidores, lo configure como para que reciba 500 peticiones c/u .

● Teniendo en cuenta que las tareas para cambiar los estados de los envíos pueden variar significativamente en el tiempo que tardan (por ejemplo, algunas demoran 1 segundo y otras hasta 5 segundos o más), el método **Least Connection** sería una mejor elección. Esto aseguraría que las nuevas solicitudes sean enviadas a los servidores menos ocupados, evitando la sobrecarga de un solo servidor.

● Se asigna balanceo , que es Round Robin con pesos. Las solicitudes se distribuyen cíclicamente entre los servidores, pero la cantidad de solicitudes que cada servidor recibe, está proporcionalmente ajustada por su peso.

● Sistema satelital . Utilizaría: **Sistema de monitoreo y login concurrentes** con **Prometheus + Grafana**, para monitorear el tiempo de ejecución de las tareas concurrentes, la carga de CPU, el uso de memoria, y cualquier otra métrica clave.

Logging de Transiciones de Estados: ELK Stack (Elasticsearch, Logstash, Kibana) puede ser utilizado para centralizar y analizar logs relacionados con las transiciones de estados en mercado-envíos.

Sistema de Cache Distribuido.: Uso de **Redis** para almacenar en caché estados intermedios o resultados parciales de las tareas, reduciendo la carga sobre las bases de datos y mejorando el tiempo de respuesta del endpoint /api/status.

Sistema de Colas de Mensajes (Message Queue):Gestión de Tareas Asíncronas con **RabbitMQ** o **Apache Kafka** pueden gestionar las comunicaciones asíncronas entre Sistemas y mercado-envíos, asegurando que los mensajes que indican transiciones de estados se procesen de manera confiable.

Escenario 2*: Configuré un ambiente como para que en caso que haya un crush o fallo, siga funcionando. Entonces yo puedo recibir conexiones en algún momento y tener un plan de contingencia en donde yo pueda venir y cambiarle un sólo parámetro, actualizarlo y listo.

Monitoreo:

Para monitorear las urls/endpoints del proyecto, utilizaría herramientas como **Prometheus**, **Grafana**, **Nagios**, o **Zabbix** con el objetivo de Verificar que las URLs estén disponibles y respondiendo dentro de un tiempo de respuesta aceptable.

Herramientas como **New Relic**, **Datadog**, o **AppDynamics** para monitorear rendimiento .Pruebas de Carga y Stress:Para simular cargas de tráfico y verificar cómo se comportan las URLs bajo condiciones de alta carga. con herramientas como **Apache JMeter** o **Gatling**

Endpoint de Tareas Concurrentes:

• **URL:** /api/tarea-concurrente: Este endpoint es crucial porque maneja la ejecución de tareas concurrentes. Monitorear su disponibilidad y rendimiento, asegura que las tareas se procesen correctamente.

Endpoint de Estado:

• **URL:** /api/status: Permite consultar el estado de los envíos sin interferir con la ejecución de las tareas para asegurarse de que este endpoint esté operativo, y garantizar que se pueda consultar el estado de las tareas en cualquier momento.

Endpoints de "Mareo Envíos":

• **URL:** /shipplings/{shippingId} (PATCH requests): Éstos endpoints permiten actualizar el estado de los envíos. Monitorear estas URL asegura que las actualizaciones de estado se realicen correctamente y a tiempo.