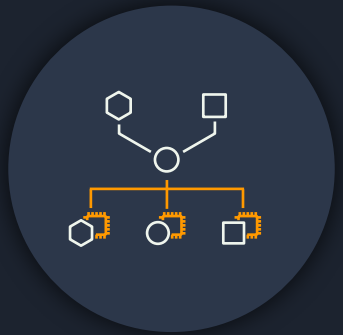
Opciones de escaladores de contenedores con mejores prácticas

Iago Banov
Sr. Specialist Solutions Architect, Containers - LATAM

Puntos problemáticos



Escalabilidad

Alta latencia

Disponibilidad

Objetivo afectado

Amazon EKS: ¿por qué?

Amazon EKS le permite crear aplicaciones confiables, estables y seguras en cualquier entorno.

Proteja Kubernetes Upstream

Usa las API estándar de Kubernetes. Funciona con herramientas comunitarias. Parcheado y protegido por AWS.

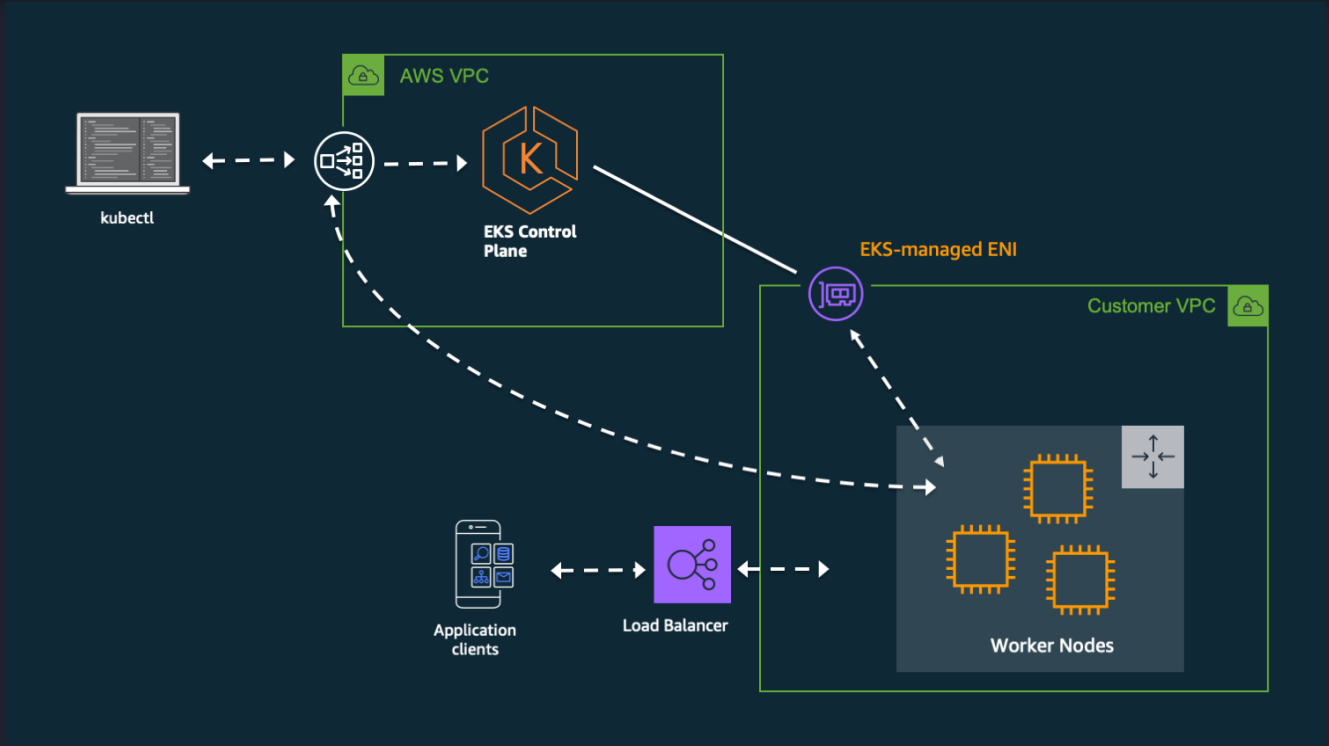
Alta disponibilidad

Diseñados para cargas de trabajo de producción, todos los clústeres tienen una alta disponibilidad. Respaldado por un SLA del 99.95%.

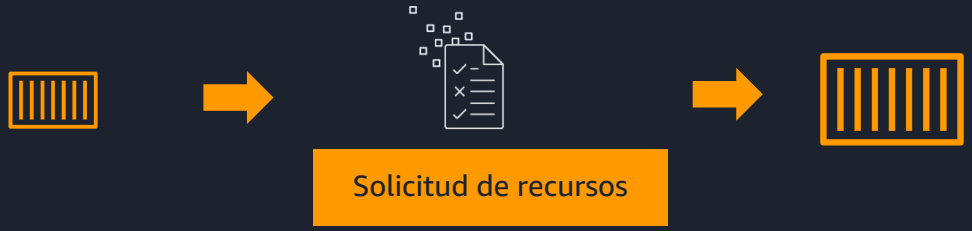
Integrado

con el ecosistema de AWS: redes de VPC, equilibrio de carga elástico, permisos de IAM, CloudWatch y más

Arquitectura de clústeres EKS



Aplicación de escalado automático



Vertical Pod Autoscaler (VPA)

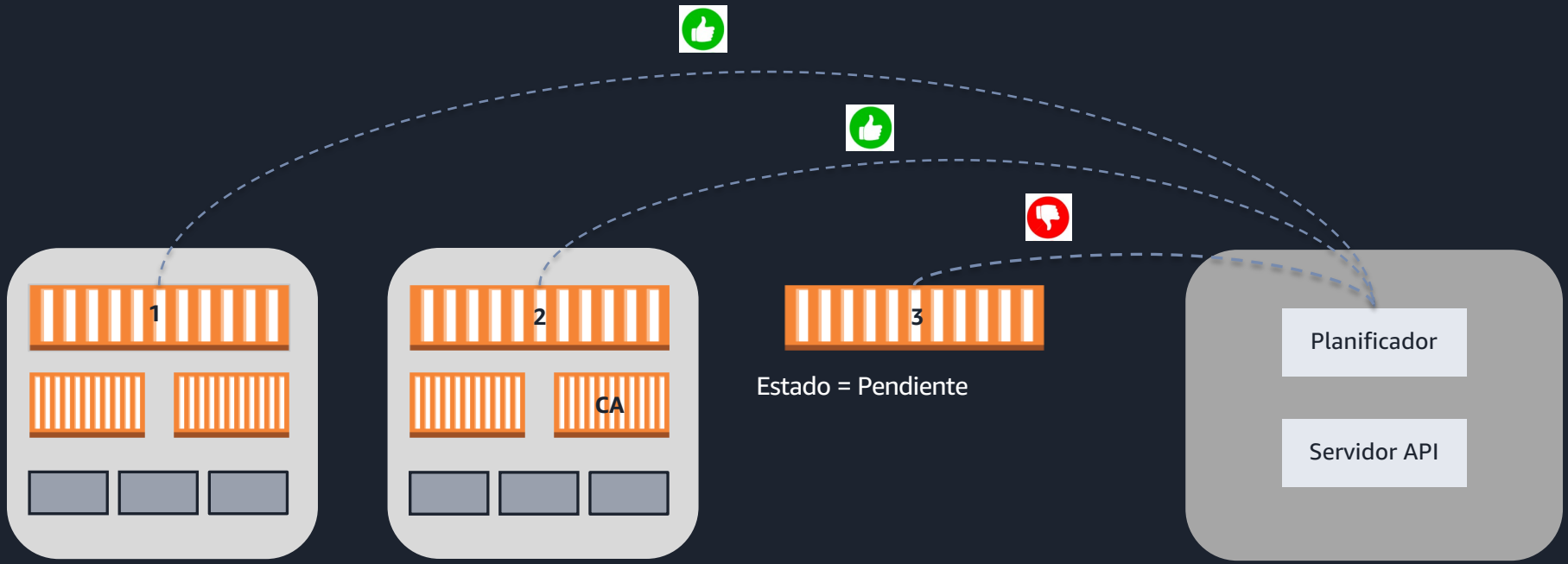
- Asigna más recursos
- Ajusta automáticamente las reservas de CPU y memoria de tus Pods

Horizontal Pod Autoscaler (HPA)

- Asigna más pods
- Los controles HPA se escalan mediante Deployment and ReplicationSet

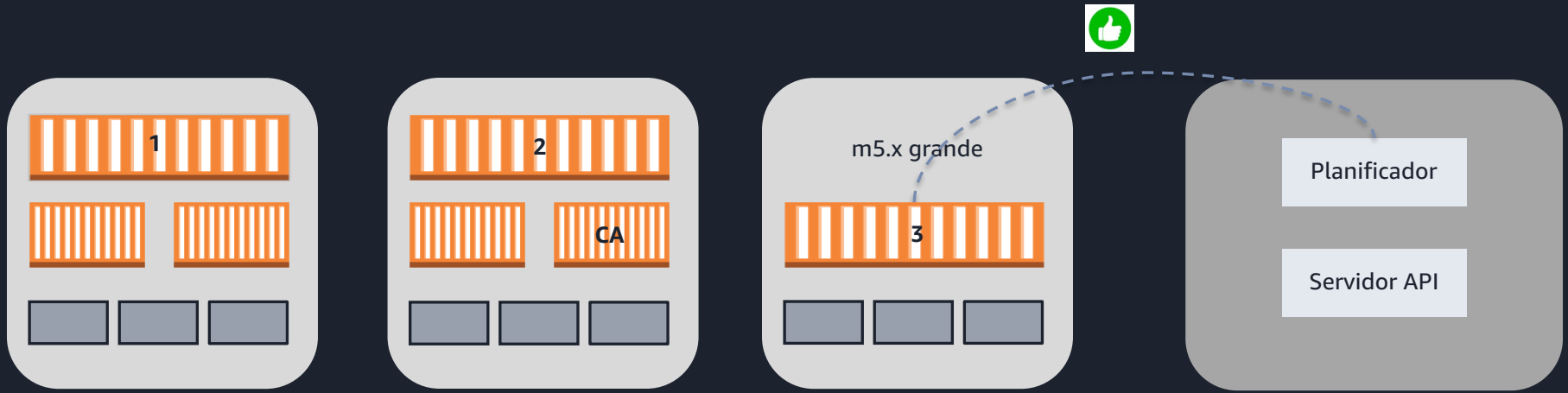
¿Cómo funciona el escalamiento de clústeres?

- 2 de los pods están programados para ejecutarse en las instancias existentes
- El tercer pod queda en estado pendiente debido a la falta de recursos de la CPU



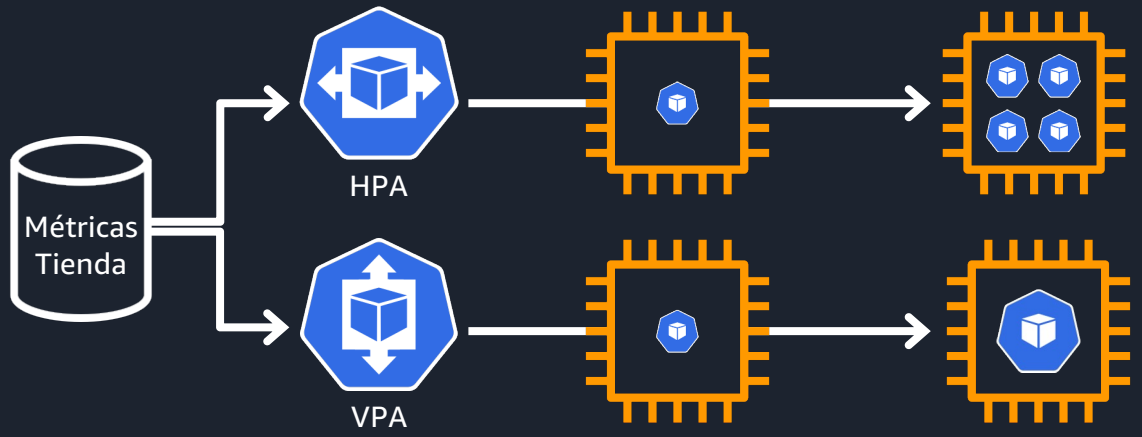
¿Cómo funciona el escalamiento de clústeres?

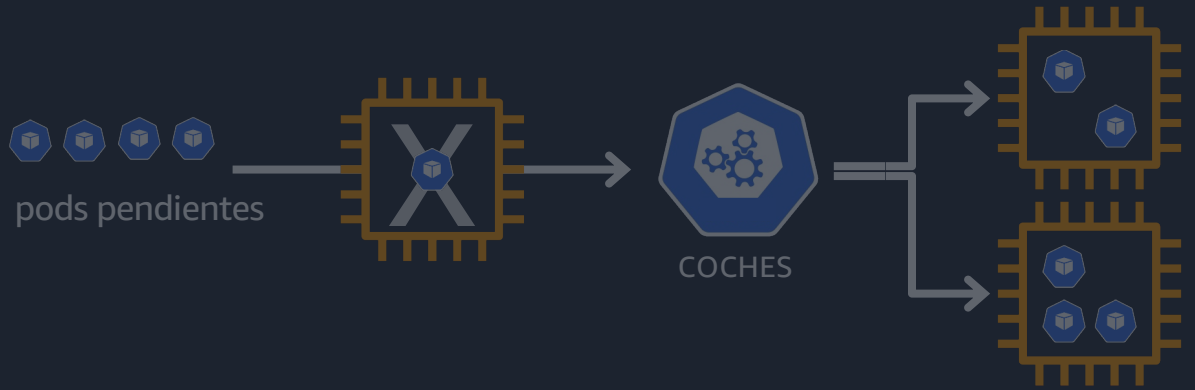
- El Cluster Autoscaler le dice a ASG que escale
- Se lanza una instancia EC2 del mismo tipo.
- El tercer pod está programado para ejecutarse en la nueva instancia



Escalado automático de Kubernetes

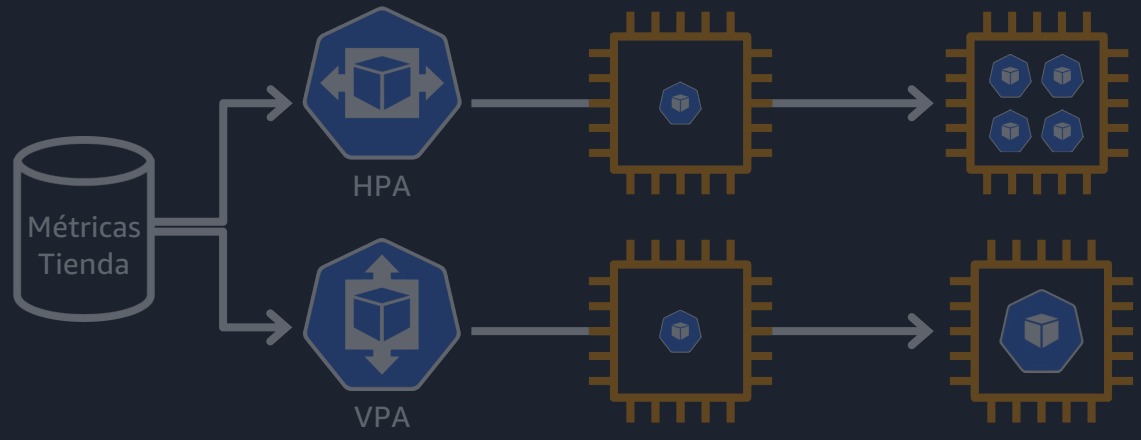
1. Auto scaling de containers horizontales (HPA)
2. Auto scaling de containers verticales (VPA)
3. Cluster Autoscaler (CAS)

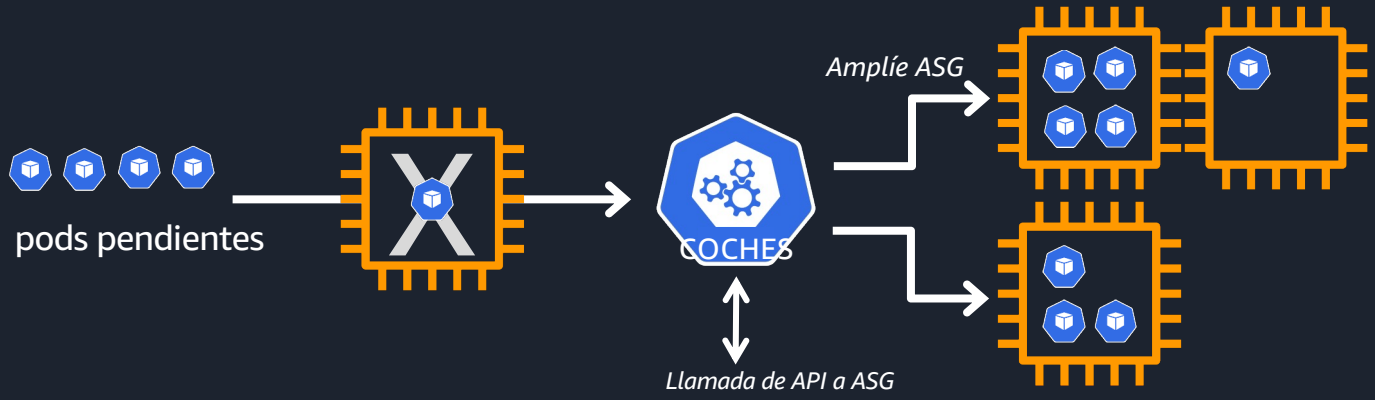




Escalado automático de Kubernetes

1. Auto scaling de containers horizontales (HPA)
2. Auto scaling de containers verticales (VPA)
3. Cluster Autoscaler (CAS)





Limitaciones del Cluster Autoscaler

- La estrategia de Autoscaler se centra en el uso de EC2 Autoscaling Group
- Asume que los tipos de instancias son idénticos en un grupo de Nodes
- Los tipos de instancias mixtos deben ser lo más homogéneos posible en cuanto a CPU y memoria
- Necesita varios grupos de Nodes para admitir diferentes tipos de instancias
- La mejor práctica es tener un grupo de Nodes por AZ
- Conduce a una gran proliferación de grupos de Nodes en grandes clústeres

AWS Fargate

Presentación de AWS Fargate con EKS

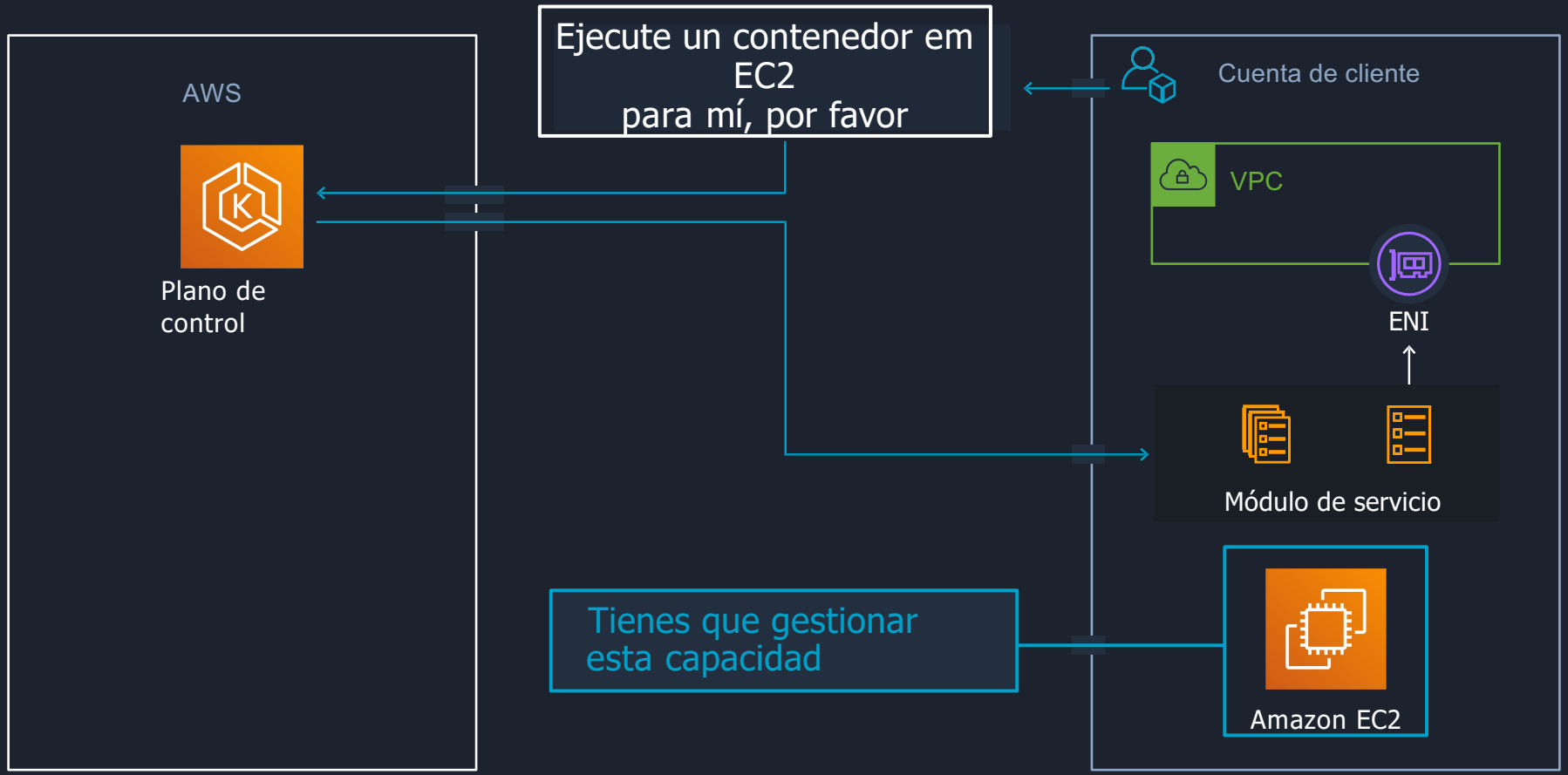
Tu Aplicaciones en contenedores

Serverless
Administrado por AWS, sin instancias de EC2 que aprovisionar, escalar o administrar: modelo de operaciones sencillo

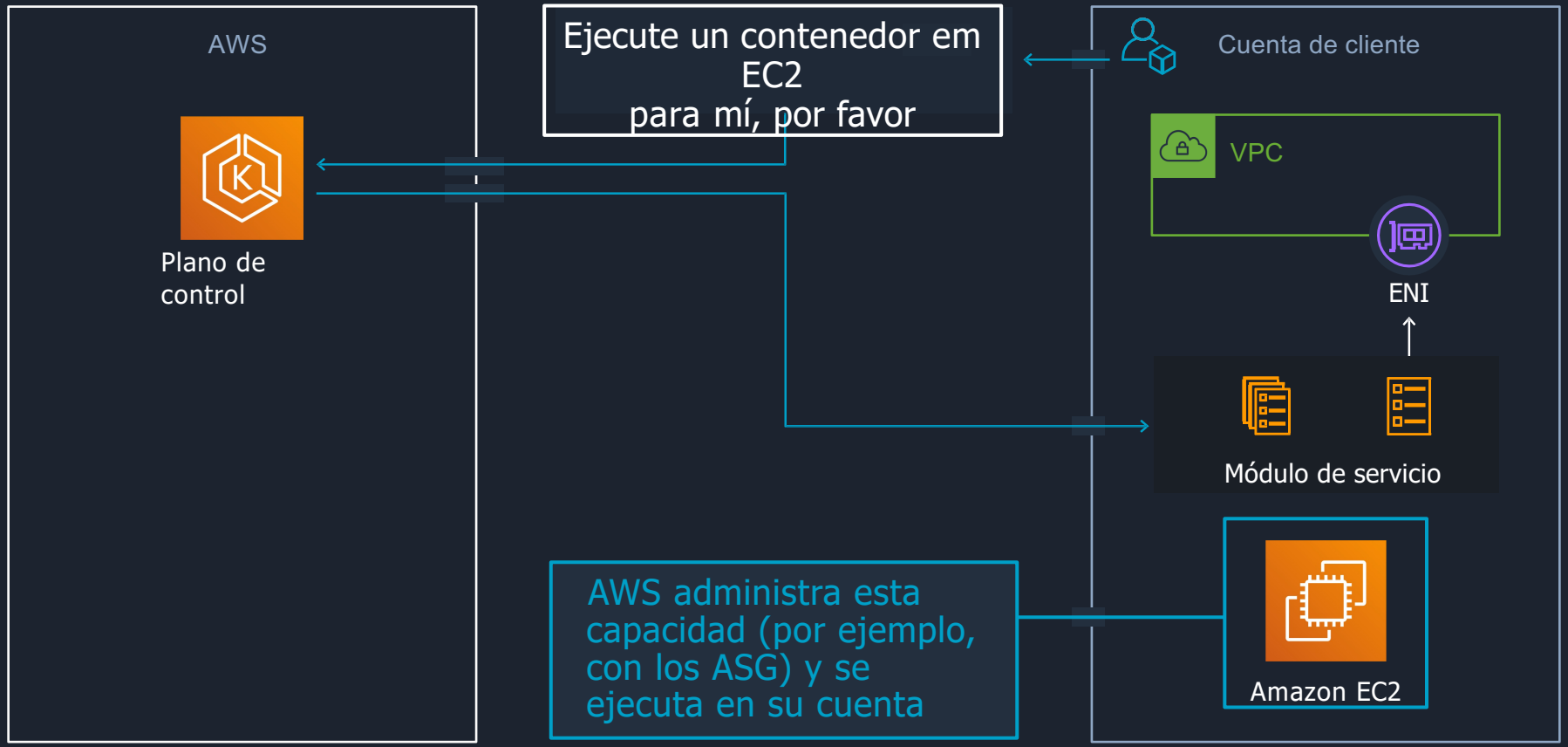
Seguro de forma predeterminada
Aislado, parcheado y compatible para ejecutar las cargas de trabajo más sensibles

Ahorros
Modelo elástico de precios bajo demanda con asignaciones de recursos granulares que brinda a los clientes la posibilidad de adquirir solo la vCPU y la memoria necesarias cuando la necesitan sin el exceso

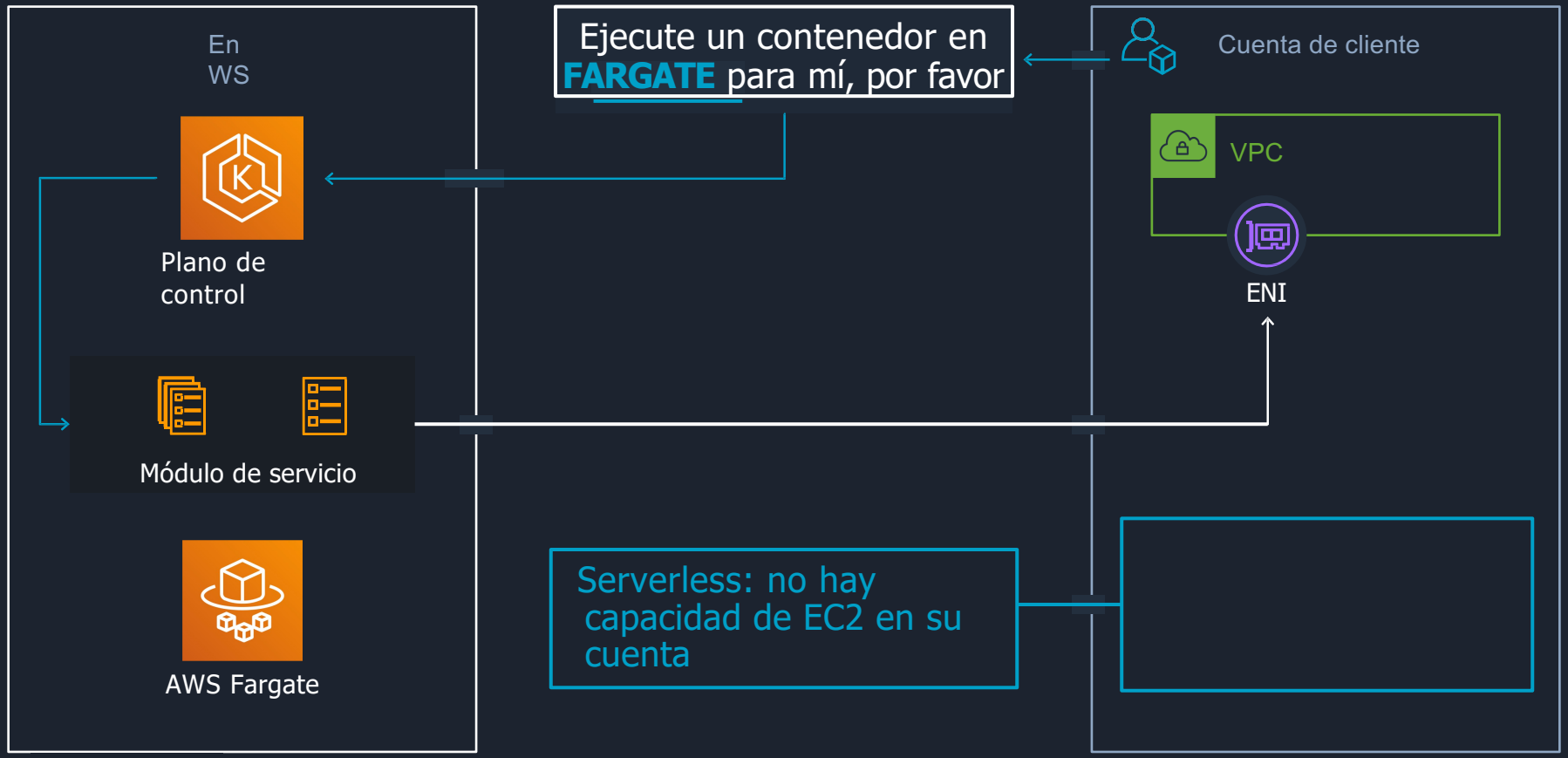
EKS con EC2


Ejecute un contenedor em EC2
para mí, por favor
AWS
Plano de control
Cuenta de cliente
VPC
ENI
Módulo de servicio
Tienes que gestionar esta capacidad
Amazon EC2

EKS con Manage Node Groups



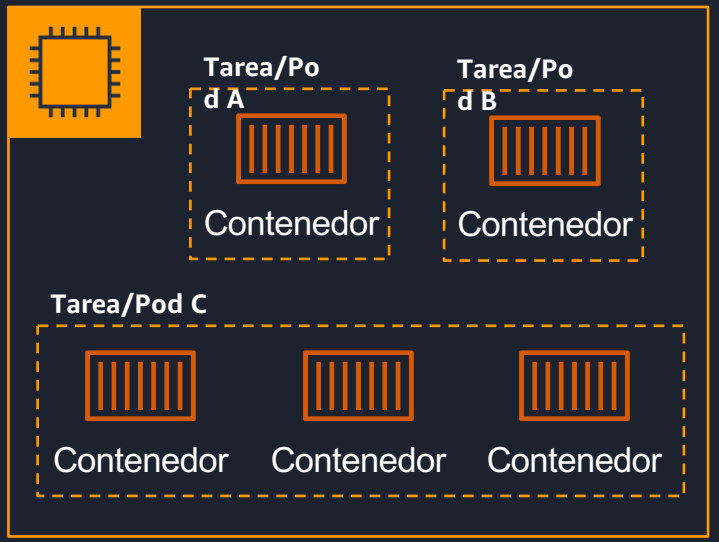
EKS con Fargate



Suministre y pague los recursos que necesita

Nodo EC2 estándar

Es difícil lograr el equilibrio entre número de contenedores y capacidad de recursos



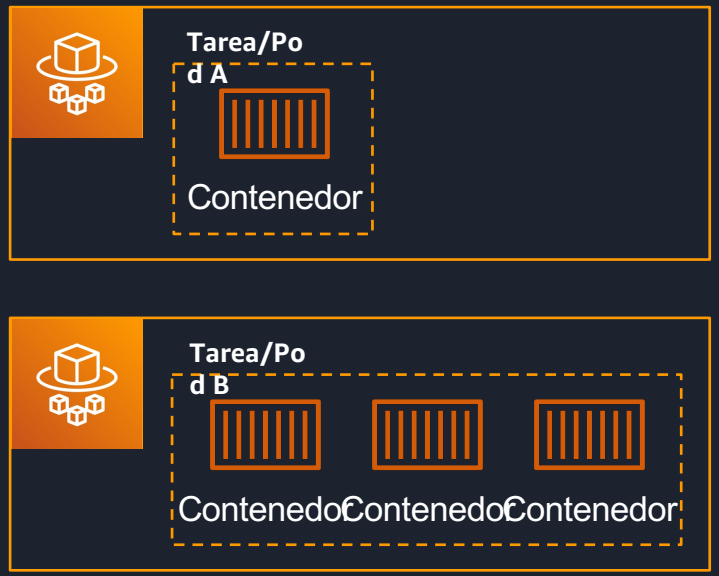
Utilización de recursos

Suele pagar por grandes cantidades de recursos de EC2 no utilizados en varios Nodes del clúster

Suministre y pague los recursos que necesita

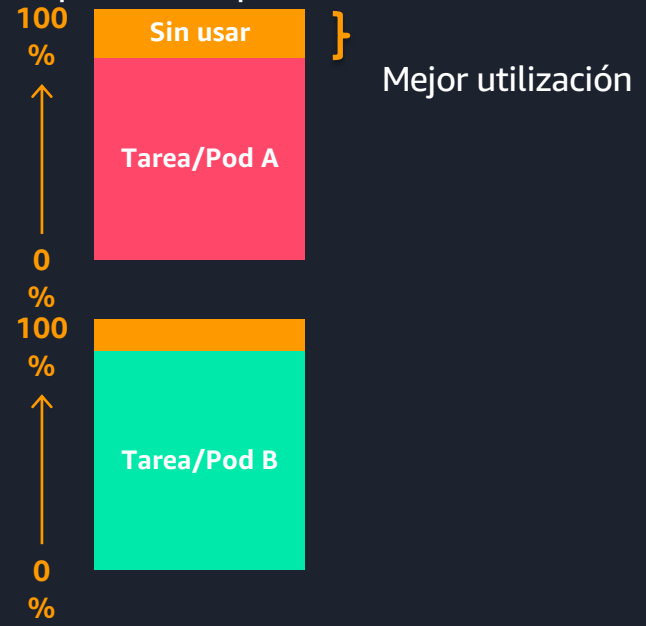
Instancia de Fargate

Ajusta el tamaño específico de cada instancia a la memoria y los recursos de vCPU necesarios

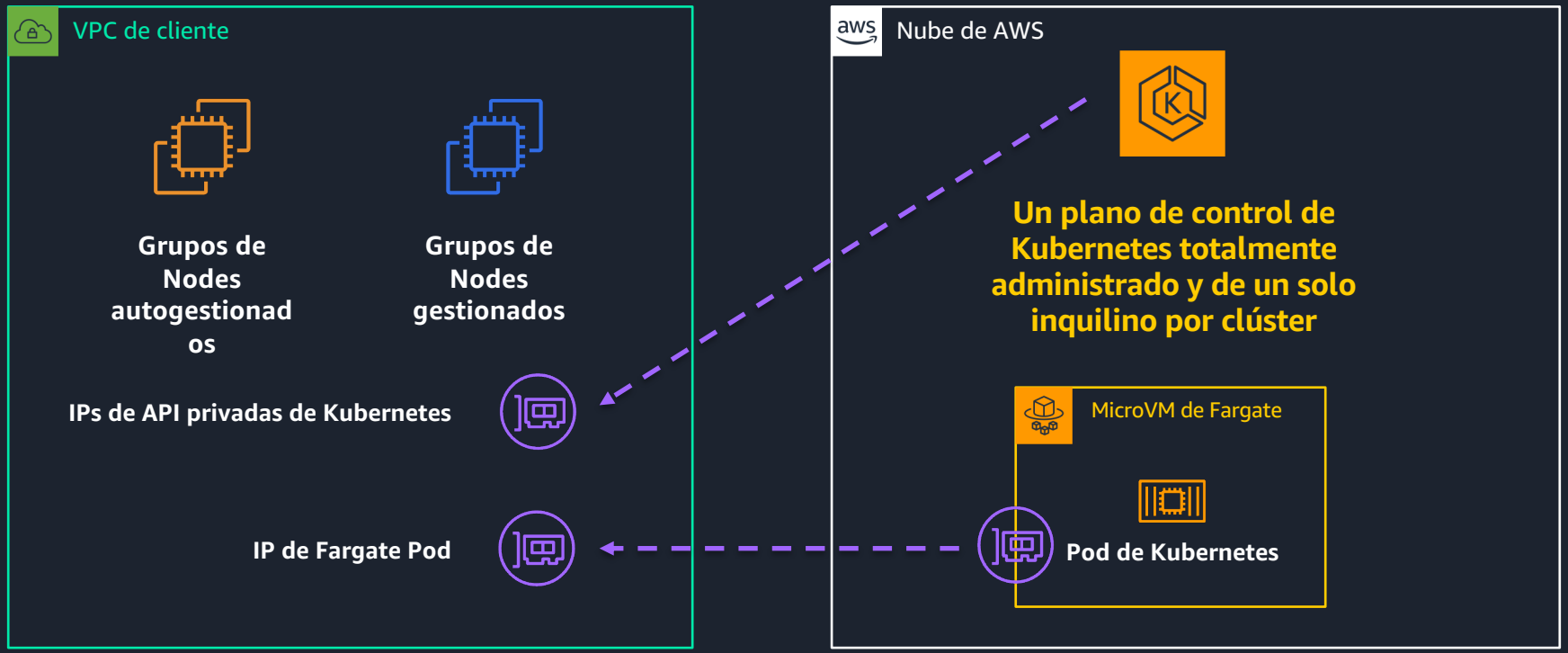


Utilización de recursos

Establezca un control más detallado sobre los recursos con Fargate, minimizando la capacidad desperdiciada



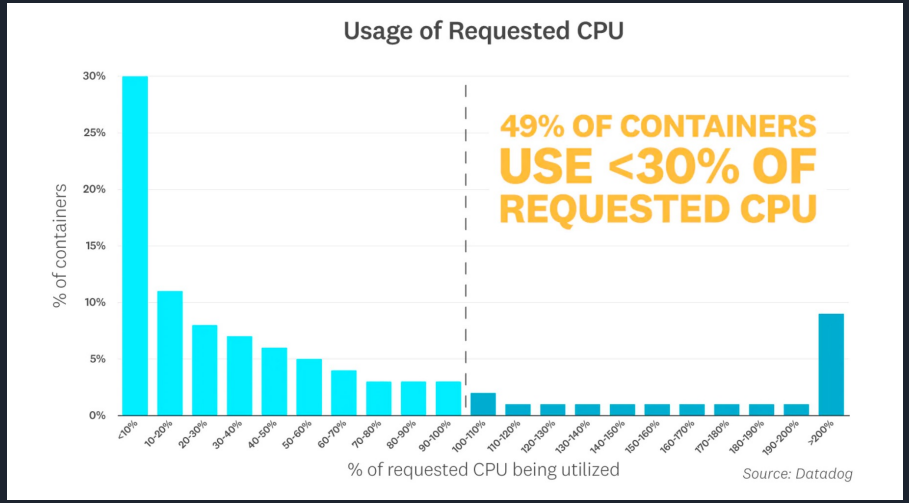
Arquitectura de alto nivel de Amazon EKS



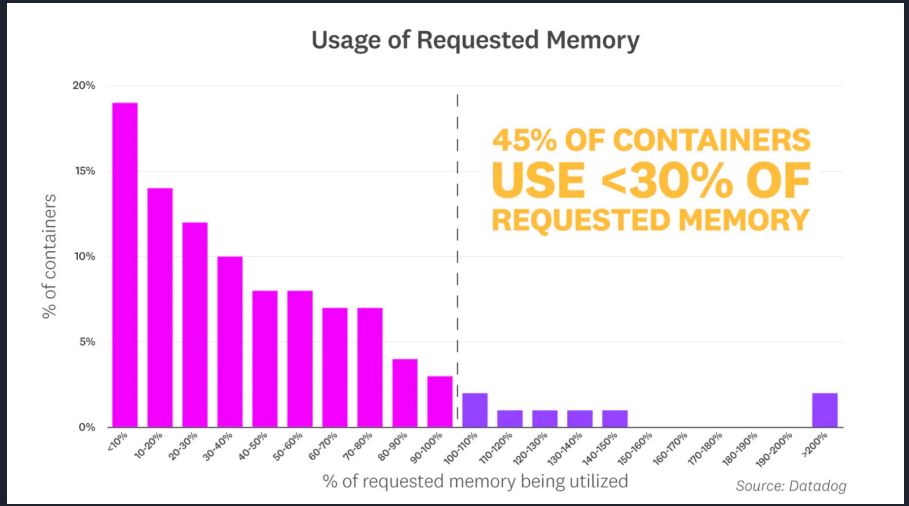
Elaboración de perfiles de aplicaciones

Métricas de uso medio de contenedores

CPU



Memoria

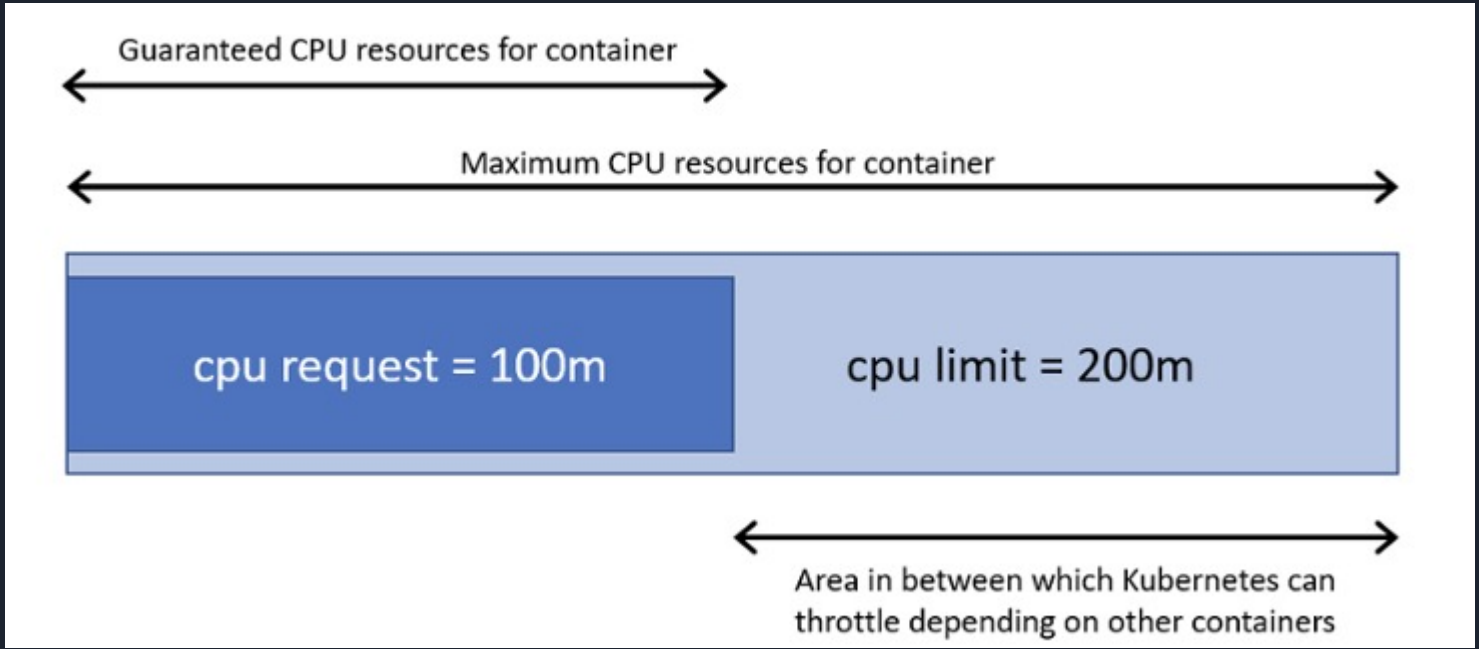


Elaboración de perfiles de aplicaciones

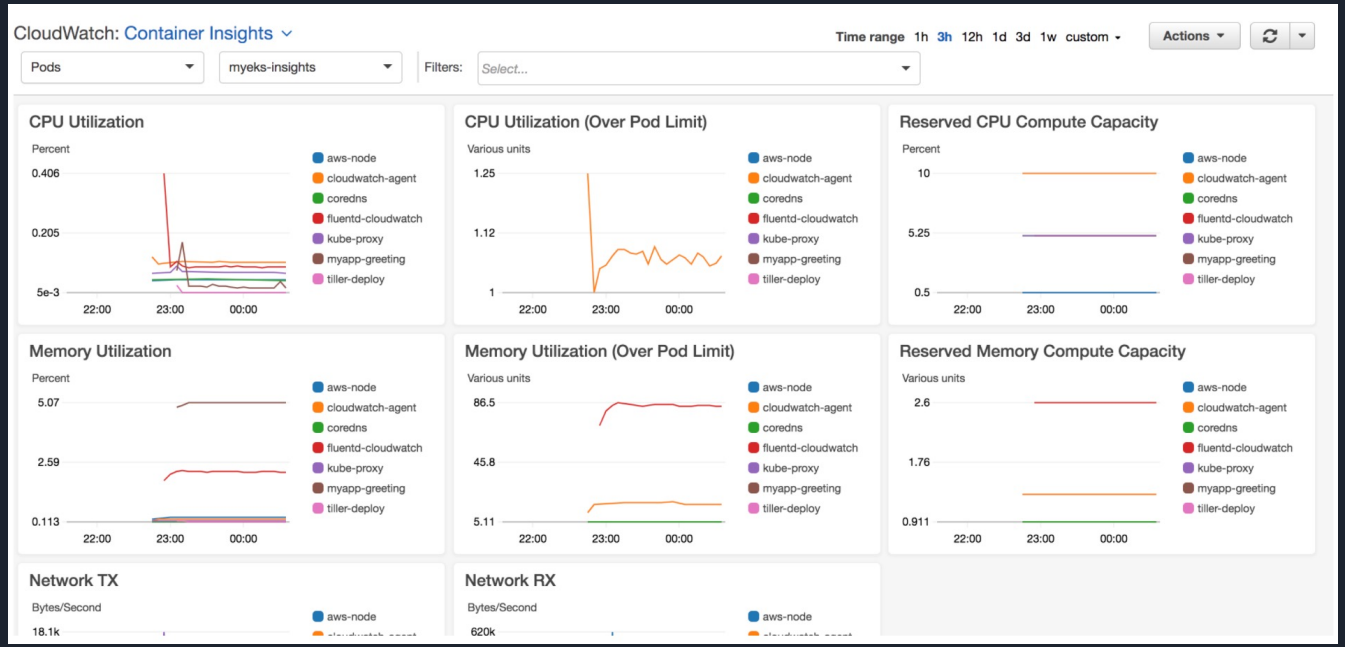
En raras ocasiones, la asignación de CPU y memoria para una carga de trabajo será correcta la primera vez;

La creación de perfiles continuos le ayuda a ajustar las asignaciones de recursos para evitar el sobreaprovisionamiento.

Solicitudes y límites



Comprenda el uso de los recursos



Mejores prácticas para la creación de perfiles de aplicaciones:

- Identifique los requisitos de recursos mediante la creación de perfiles de las aplicaciones
- Realice perfiles continuos para ajustarlos a lo largo del tiempo
- Aplicaciones de prueba de carga para determinar los requisitos de recursos a escala
- Cuando sea necesario, implemente límites

Mejores prácticas

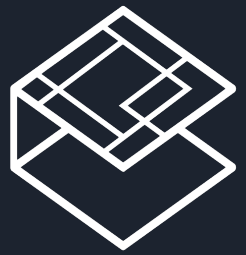
Configuración de red IPv4 del Pod

Modelo de red de Kubernetes

Cada pod tiene su propia IP

Los contenedores del mismo Pod comparten la red (IP)

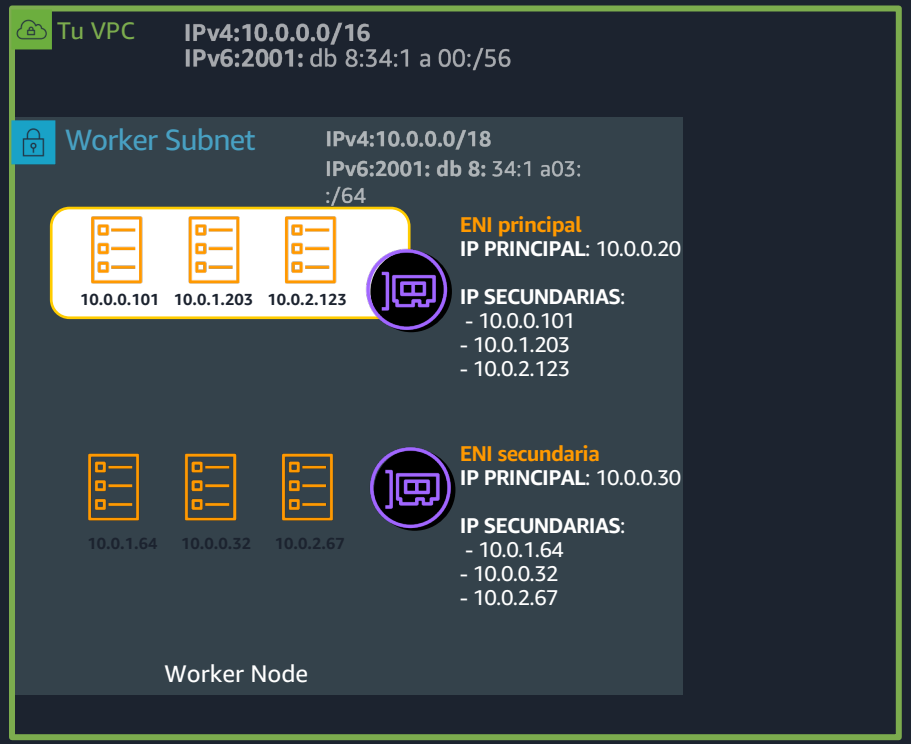
Los pods se comunican con otros pods sin NAT



CNI

Plugin de
interfaz de red de contenedores

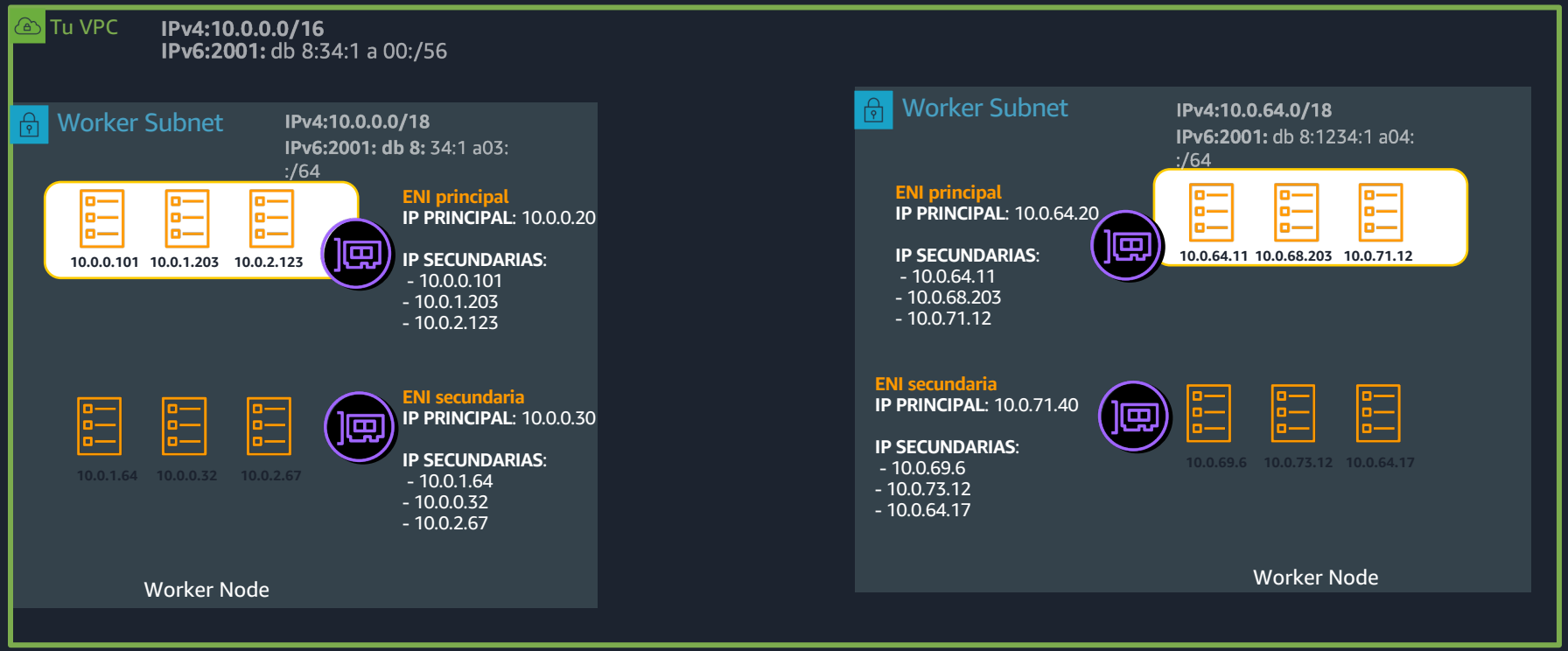
Redes IPv4 Pod: complemento CNI



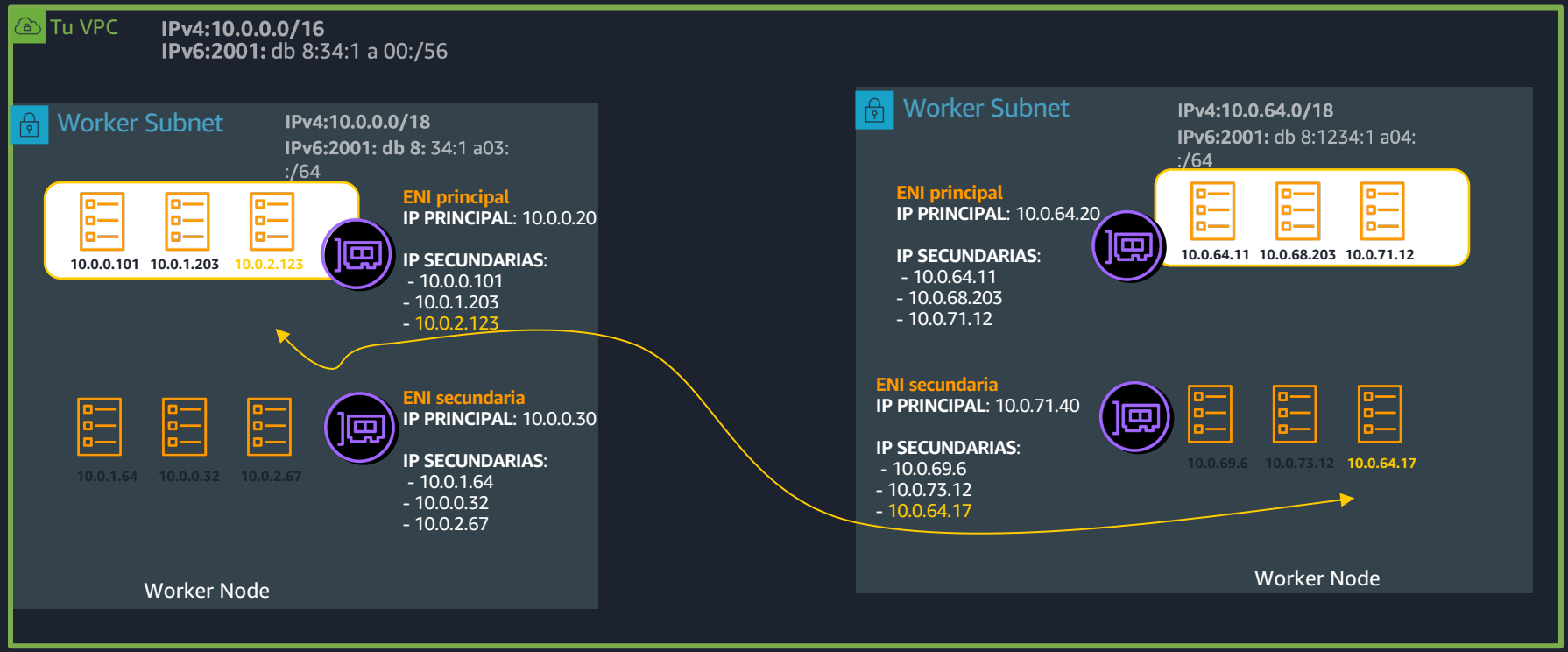
Complemento Amazon VPC Container Network Interface (CNI) se utiliza para
- Creación y conexión de ENI a Nodes
- Asignación de IP secundarias a los Pods

Alternativas a Amazon VPC CNI
- Calico de Tigera
- Cilium de Isovalent
- Weave Net de Weaveworks
- Antrea de VMware

Redes IPv4 de Pod: Pod a Pod



Redes IPv4 de Pod: Pod a Pod



Amazon CNI: opciones de configuración

- Delegación de prefijos

```
kubectl set env daemonset aws-node -n kube-system enable_prefix_delegation=Verdadero/falso
```

- security group para Pods

```
kubectl set env daemonset aws-node -n kube-system enable_pod_eni=Verdadero/falso
```

Delegación de prefijos de Amazon VPC

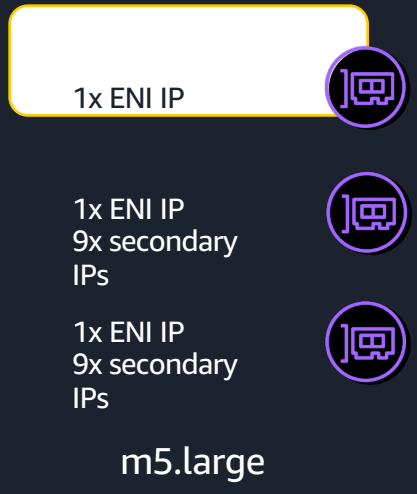
Permite asignar un prefijo a un ENI EC2

- Bloque /28 para IPv4 (16 direcciones IPv4)

- Bloque /80 para IPv6 (280 billones de direcciones IPv6)

La delegación de prefijos solo se admite en instancias de Nitro

IPs secundarios en Pods

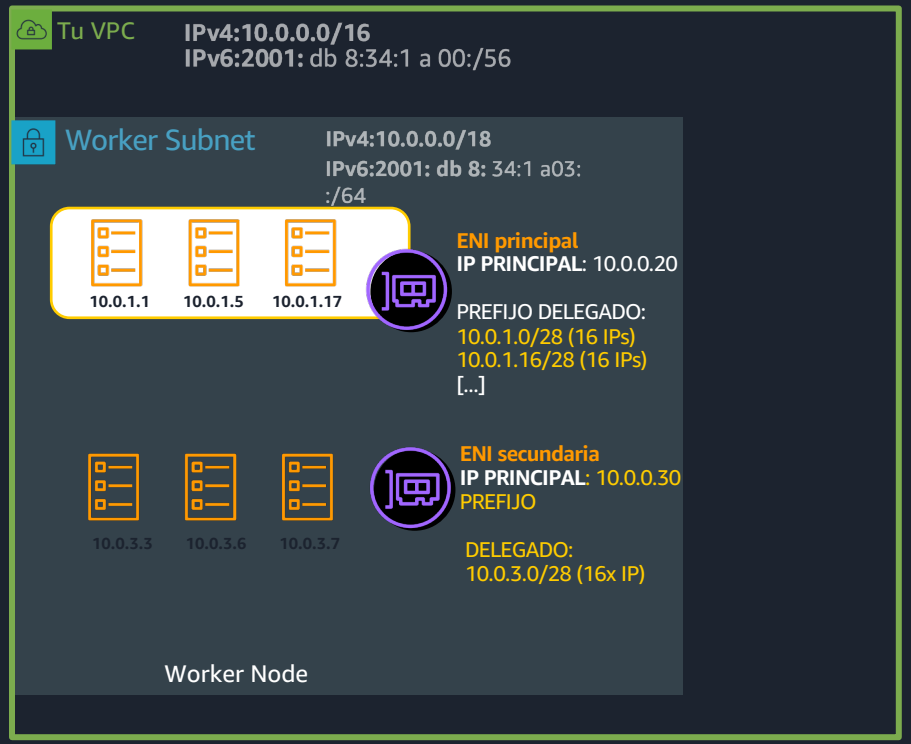


Instance type	Maximum network interfaces	Private IPv4 addresses per interface
m5.large	3	10

MAX PODS = (Number of network interfaces × [the number of IP addresses per network interface – 1]) + 2

29

Redes IPv4 Pod: delegación de prefijos



Beneficios

- Mayor densidad de pods
- Se requieren menos llamadas de API al plano de control de EC2

```
kubectl set env daemonset aws-node -n kube-system ENABLE_PREFIX_DELEGATION = true
```

¡Gracias!

¡Gracias!