

Examen 20 pags.

IP publica de la máquina esta en el IG (internet gateway) - no se puede conseguir con un ip a.

Dentro vpc - solo IP privados

R53_zone \Rightarrow crea DNS.

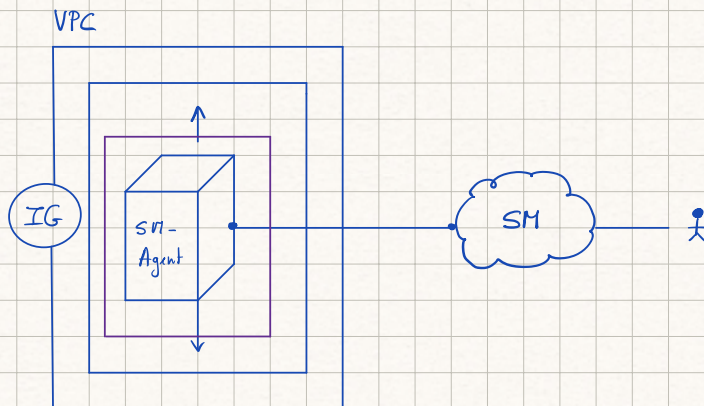
Autoescalng group al caer máquina \rightarrow crea una \rightarrow la registra en el R53.

Budgets (Billing Console)

Administración de máquinas: ssh / También se puede por consola.

Systems Manager (servicio amazon, CORE)

SM-agent



Session Manager \rightarrow añadir parches

Kubernetes

Muchas app mismos recursos → cuello de botella

Ahora = una aplicación en cada máquina

2 Problema = uso medio de recursos (20%). 80% recursos inutilizados pero pagados.

ej. BD 97% inutilizados 3% usados
oracle

Como evitarlo? Distribuyendo los recursos.

Disco duro



Imagen

ej



Registro: BD de discos duros (Registry)
2 de imagenes

Mecanismos que permiten restringir registros = linux controll group

Compartir en las mismas máquinas aplicaciones = Contenedor

BD que contuviese todos los recursos + "inteligencia"

Se debe adaptar cada contenedor para cada plataforma - el multcloud no existe

EKS - crear Kubernetes (AWS)

eksctl create cluster (crear cluster y mantenerlo)

Kubectl → lo configuraremos para que pueda hablarle al cluster de Kubernetes

Namespace = mecanismo administrativo

Pods = una app se ejecuta usando pods. Serie de contenedores.

Replicaset 1 = lanza copias de las apps, de los pods

Replicaset 2 = lanza otro replicaset si queremos modificar los contenedores

Deployments = crea un replicaset que crean los pods

Services = balanceador de carga para poder acceder a los pods.

Lo pondremos todo dentro del mismo fichero pero debería ser un recurso un fichero.

EOF = end of file

De una pipeline de despliegue cual es la acción más importante? Rollback