ECS -> orchestratore minimale di container, lightweight rispetto a Kubernetes originale  
EKS -> distro di Amazon di Kubernetes, orchestratore a tutti gli effetti

EC2 -> servizio amazon per creare VM  
Fargate-> servizio amazon per configurare VM serverless  
  
ECS si puo’ usare con EC2 o con Fargate  
  
ECR-> il registro immagini docker di Amazon  
RDB -> servizio AWS per creare database relazionali  
Parameter Store -> servizio in cloud AWS per salvare le mappe di parametri  
Secret Manager -> servizio in cloud AWS per salvare i secrets  
  
Se configuri le dipendenze di spring boot per amazon cloud, puoi interagire con alcuni dei servizi di cui sopra.  
Ad esempio se le metti come dipendenze maven, e poi crei un file bootstrap.properties, quando l’applicativo spring boot sale cerca nel bootstrap properties per esempio la convenzione dei file di configurazione da iniettare nell’environment (un po’ come avviene nei deployment file per kubernetes).  
La convenzione delle properties quindi nel parameter store per AWS è :  
/prefix/appname/nome.del.parametro

Appname è il nome che si configura nell ‘application.properties del progetto spring es  
spring.application.name=letitout  
quindi per esempio se la nostr applicazione è definita con name letitout e sul Parameter Store di AWS

Mettiamo  
/config/letitout/userdatabase.hostname  
  
allora nel mio application properties potro’ avere nella stringa dell’url del db per esempio

spring.datasource.url=jdbc:mariadb://${**userdatabase.hostname**}:3306/dbname  
visto che viene fatto l’injection di userdatabase.hostname che si prende dal parameter store AWS sul cloud.  
  
NB: per altri servizi è necessario fornire parametri specifici nell’application.properties per AWS, ad esempio tipicamente la coppia di key/name per l’authorization (access key name e value)

Su AWS se si crea un **task definition + un container** (le specs del container) associato, se vengono deployate in un **cluster** (creato sempre su AWS) diventano un **Servizio** AWS