AWS container services common problems and solutions

LVEE 2017

JUNE 1, 2017

General background

AWS (Amazon Web Services - Cloud Computing Services) have several container-based services. These services are very-good connected in AWS infrastructure.

Reliable tested list of container services in AWS

- EC2 CS (Container service of AWS)
- Lambda (Service of micro-computation in AWS)
- CodeBuild (Service of CI in AWS)

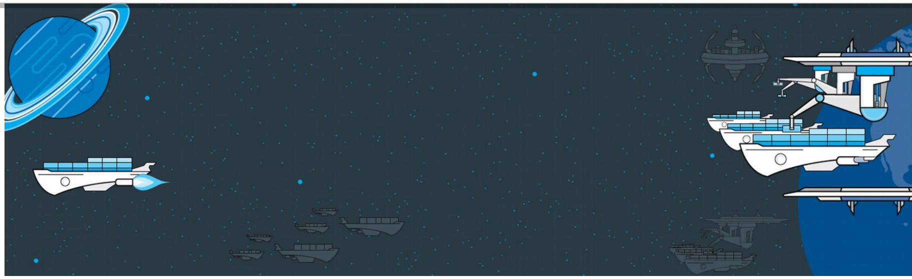
Details of service implementation

1. It is docker-based
2. Docker runs on Amazon Linux VM
3. Container properties can be heavy customized
4. Containers can be built outside AWS (except lambda)

Amazon Linux

docker

EC2 CS



Advantages

- Very heavily customized
- Organized in “services” - count of HA running containers
- Internal support for HA (Application ELB)

Disadvantages

- Require hard limitations for Memory, CPU, etc.
- It have generic problems running on the same docker host (e.g. ports)
- It can’t distribute evenly containers by data-centers (availability zones)
- Require reserved space for service restart (especially for cloud-formation)

Lambda

Advantages

- Easy to create and run
- Very low cost of mini-operations
- Runs in HA environment (on different AZ)

Disadvantages

- Due to little memory only minimal computation is allowed
- Does not support inbound TCP or UDP
- Debugging is terrible due to log location (CloudWatch)
- Binary distributions running is complicated
- SSH access from service possible but not easy

Code Build

Advantages

- Build outside any AWS service
- Designed as “Black Box” - with inputs and outputs on S3
- Supports customized container for build, can do almost anything
- Supports git (e.g. github) input

Disadvantages

- Without customized container almost useless
- Debugging is terrible due to log location (CloudWatch)
- Inputs/outputs (e.g. git) very limited
- Process of build continues even on error after overall exec reporting it
- Takes a lot of time because of input/output/container start

Summary

1. AWS services is based on Open Source Linux and Docker
2. These services is very limited comparing to the Open Source
3. By integration inside AWS service infrastructure standard usage is easy
4. Non-default usage is either very difficult or limited
5. At the moment none of this services can replace non-containerized ones