

Lambda, Cloudwatch ec2 stop/start

Lambda ?

ec2 Lambda 1ms

1. Lambda IAM

** Lambda

[Policy]

: Lambda_policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "ec2:Describe*",
        "ec2:Start*",
        "ec2:Stop*"
      ],
      "Resource": "*"
    }
  ]
}
```

[Role]

: Lambda_role

policy(Lambda_policy) 가

2. Lambda

Author from scratch

Lambda Function : StartEC2Instance / StopEC2Instance

Runtime : Python3.8

Permissions - Use an existing role - role

[Lambda Code]

```
import boto3
region = 'ap-northeast-2'
instances = []
ec2_r = boto3.resource('ec2')
ec2 = boto3.client('ec2', region_name=region)
//
for instance in ec2_r.instances.all():
    instances.append(instance.id)

def lambda_handler(event, context):
    ec2.start_instances(InstanceIds=instances)
    print('started your instances: ' + str(instances))
```

ec2

```
import boto3
region = 'ap-northeast-2'
instances = []
ec2_r = boto3.resource('ec2')
ec2 = boto3.client('ec2', region_name=region)
```

```
for instance in ec2_r.instances.all():
    for tag in instance.tags:
        if tag['Key'] == 'auto-schedule':
            if tag['Value'] == 'auto':
                instances.append(instance.id)
```

```
def lambda_handler(event, context):
    ec2.start_instances(InstanceIds=instances)
    print('start your instances: ' + str(instances))
```

가

Lambda - Deploy - Test

- 1) IAM Policy, Lambda code
- 2) Lambda-General configuration - Timeout 3(default) -> 15
- 3) region ec2 Instance state가 1EA terminate
가 .

3. CloudWatch

Event - Rules - Go to Amazon EventBridge

step1)

 : StartEC2Instance
Event bus : default
Rule type : Schedule

step2)

GMT cron

step3)

target types : AWS service
Select a target : Lambda function
Function : Lambda Function (StartEC2Instance)
++ 가 가

step4)

() 4. Lambda code /

[SNS]

topic
subscription
 Protocol : Email
 Endpoint :

[Lambda]

Configuration - Destinations - Add destination
source : Asynchronous invocation
Condition : On Failure / On Success
Destination Type : SNS topic
Destination : topic

CentOS 7

(2)

CentOS 7

CentOS 7

가

/

1.

```
##
```

```
yum update [ ]
```

```
##
```

```
yum list updates
```

2.

```
# epel-release :
                                                    disable
    enablerepo
```

```
vi /etc/yum.repos.d/epel.repo
enabled=1 0
```

```
[epel]
...
enabled=0
...
```

```
yum repolist
yum --enablerepo=epel install [    ]
```

```
# net-tools :
ifconfig,
                                                    netstat
                                                    IP
```

```
<ifconfig>
ifconfig -a
ifconfig [interface] up
ifconfig [interface] down
```

```
<netstat>
netstat -nap
netstat -an | grep [Port]
netstat -nlpt
```

```
# unzip : zip
```

```
unzip [file_name].zip
unzip -l [file_name].zip
unzip -t [file_name].zip
```

```
# wget :
                                                    가
                                                    HTTP, HTTPS, FTP
```

```
wget -O [    ] [URL    ]
```

```
wget --no-check-certificate [URL ]
```

```
# curl : HTTP, FTP
```

```
Web wget
```

```
curl -o [ ] [URL ]
```

```
curl -T [ ] [ IP]
```

```
curl -L [URL ]
```

```
# chrony : NTP Server/Client ntpd
```

```
## NTP IP 가
```

```
vi /etc/chrony.conf
```

```
server [NTP server IP] iburst <iburst =  
>
```

```
systemctl restart chronyd  
chronyc sources
```

```
# gcc, gcc-c++ : C , C++
```

```
# openssl-devel : openssl openssl
```

```
# htop : (  
)
```

```
PID
```

USER
PR
NI
VIRT
RES
SHR
S
%CPU
%MEM
TIMR+
COMMAND

iftop :

iftop -i eno1
iftop -f "dst port 22"

dstat : I/O
가 .

dstat -tcdml

sysstat : sar, iostat

<iostat> : CPU , .
iostat -d 3
iostat -c 3

<sar> : /var/log/sa sa

.
sar -u
sar -r
sar -dp
sar -n DEV

lsof :

```
lsof -u [      ]
lsof -i
lsof -c [      ]
```

```
# psmisc : proc
killall, pstree
```

fuser,

```
<fuser> :                umount
```

```
,                kill
```

```
fuser -v [      ]
```

```
fuser -ck [      ]
```

```
<killall> :
```

```
killall -i [      ]
```

```
killall -v [      ]
```

```
killall -w [      ]
```

```
<pstree> :                Tree
```

```
pstree -anp
```

3.

```
# mlocate :                find
```

```
updatedb
```

```
locate [      ]
```

```
locate -n [    ] [      ]
```

```
# ncat :                가
```

```
<      >
```

```
ncat -l [Port]
```

```
ncat -lk [Port]
```

```
<          >
```

```
ncat [Server IP] [Port]
```

```
# whois :                IP
```



```

whois [          ]
whois [IP       ]

# cloud-utils-growpart : LVM                root
가
.

growpart [        ] [          ]
resize2fs [        ]

# tcping :                TCP            ping
.

tcping [Server IP] [Port]

```

About OOM Killer ?

Kernel 5.4.0-104-generic

OOM(Out Of Memory) ?

Linux swap , 가 가
 Over Commit 가 .

OOM(Out Of Memory) Killer ?

Linux 가 Out Of Memory가 , OOM Score
 Kill Linux Kernel
 OOM Killer /var/log/

oom_killer

Log

```
$ cat /var/log/syslog | grep oom
Mar  7 19:14:00 zabbix-node01 kernel: [1132818.054201]
ib_log_writer invoked oom-killer:
gfp_mask=0x100cca(GFP_HIGHUSER_MOVABLE), order=0,
oom_score_adj=0
OOM Score          OOM Killer
```

1. + fork()
- 2.
3. 가 , root (super user)
4. nice 1 Score 2 가
5. /proc/[PID]/oom_score_adj (가 가)
6. /proc/[PID]/oom_adj (가 가)
OOM Score .

```
# oom_score
```

```
$ cat /proc/890081/oom_score
```

```
1048
```

```
# 890081 PID , OOM Score 1048 .
```

```
oom_adj / oom_score_adj
```

```
가 OOM Score 가 OOM Killer
```

```
가 , oom_adj /oom_score_adj , OOM
```

```
Score 가 .
```

```
oom_adj -17 ~ 15 , -17 OOM Killer Disable
```

```
가 .
```

```
oom_score_adj -1000 ~ 1000 , OOM Score
```

```
oom_score_adj 가 .(oom_scoer_adj -1000 oom_adj
```

```
-17 .)
```

```
oom_adj / oom_score_adj
```

```
oom_score
```

```
OOM Score
```

```
oom_score_adj
```

```
OOM Score
```

```
.
```

```
# oom_score
```

```
$ cat /proc/890081/oom_score
```

```
1048
```

```
# 890081 PID , OOM Score 1048 .
```

```

# oom_score_adj          OOM Score
$ echo -1000 > /proc/890081/oom_score_adj

$ cat /proc/890081/oom_score_adj
-1000
# oom_score_adj          가          -1000

#          oom_score / oom_adj
$ cat /proc/890081/oom_score
0
$ cat /proc/890081/oom_adj
-17
# oom_score_adj          , oom_adj          -17 (OOM Killer
Disable)
# oom_score          1048 -> 0          .
가          , overcommit          . over commit
.

```

over commit

```

$ cat /etc/sysctl.conf | grep overcommit_memory
vm.overcommit_memory = 1 # 0~2          가          .
# 0 = Heuristic overcommit. Default          ,          (
)          over commit          .
# 1 =          over commit          . OOM Killer가          ,
.
# 2 = vm.overcommit_ratio          over commit
.
#          가
$ cat /etc/sysctl.conf | grep overcommit_ratio
vm.overcommit_ratio = 90
#          vm.overcommit_memory가 2          가          .
# 90%          + swap          OOM Killer
가          .
#
$ systemctl -w
,          commit          가          over commit
?
/proc/meminfo          sysstat          .          .

```

```
# commit
$ cat /proc/meminfo | grep Commit
CommitLimit: 30229156 kB
# vm.overcommit_memory 2 vm.overcommit_ratio
commit 가
Committed_AS: 64267776 kB
# commit
```

sysstat commit

```
$ apt install -y sysstat
# sysstat
```

```
# sar
$ sar -r 1
Linux 5.4.0-104-generic (zabbix-node02) 04/14/2022
_x86_64_ (16 CPU)
```

```
09:25:17 AM kbmemfree kbavail kbmemused %memused kbbuffers
kbcached kbcommit %commit kbactive kbinact kbdirty
09:25:18 AM 208884 2446760 20804112 85.73 990436
1441416 64262056 196.79 19538260 3304352 1508
```

```
# %commit commit
# 100% commit , over commit
# sar (CPU, Memory, I/O)
```

```
# -r Memory , 1 1
```

```
Linux Memory Commit / Memory Over Commit ?
Memory Commit
가 가
```

```
A 가 가 A 가 ,
A ,
Memory Commit ( ) . , 가
?
```

가

1. A 가

2.

- , 가

3. (Fragmentation)가

- : RAM

가 가

-

.

Memory Over Commit

, Memory Over() Commit

.

over commit

, over commit 가 oom killer

가

.

가

.

[CKA] #2. Pod - 1

[CKA] #2. Pod (1)

CKA

가 URL

Pod ?

Kubernetes

, Pod

Container

```
Pod Network 가 /pause
Pod , 가
Pod Network , /pause 가
```

Pod Create

```
# yaml dry run

$ kubectl run hello --image=nginx --dry-run=client -o yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: hello
  name: hello
spec:
  containers:
  - image: nginx
    name: hello
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}

# | ( ) (apply) -

kubectl run hello2 --image=nginx --dry-run=client -o yaml |
kubectl apply -f -
kubectl get pods -o wide

# nodeName 가 pod 가
Taint
yaml nodeName 가
$ kubectl run hi --image=nginx --dry-run=client -o yaml >
hi.yaml
apiVersion: v1
kind: Pod
metadata:
```

```

creationTimestamp: null
labels:
  run: hi
  name: hi
spec:
  containers:
  - image: nginx
    name: hi
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
  nodeName: user-controller      ## Node
status: {}

```

```

#          YAML          Pod
$ kubectl create -f hi.yaml
$ kubectl get pods -o wide
NAME                                READY   STATUS    RESTARTS   AGE
IP                                  NODE                                NOMINATED NODE   READINESS GATES
hello-776c774f98-894tt             1/1    Running   0          18h
192.168.153.193                    user-worker                          <none>          <none>
hi                                  1/1    Running   0          3m10s
192.168.136.5                      user-controller                      <none>          <none>

```

Pod status

```
$ kubectl describe pod hi
```

```
# Pod . -- kubectl arg --
```

```
$ kubectl exec -it hi -- /bin/bash
root@hi:/#
```

Pod

```
# --replicas . deployment
```

```
$ kubectl create deployment web --image=nginx --replicas=3
deployment.apps/web created
```

```
$ kubectl get pods -o wide | grep -i web
web-76b56fd968-c2pk9      1/1      Running    0          11s
192.168.153.217    user-worker    <none>    <none>
web-76b56fd968-chr4w    1/1      Running    0          11s
192.168.136.6      user-controller <none>    <none>
web-76b56fd968-mmdfn    1/1      Running    0          11s
192.168.153.218    user-worker    <none>    <none>
```

Pod log

```
# Pod info
```

```
kubectl describe pod hi
```

```
# Pod log
```

```
kubectl logs hi
```

```
# journal Log(kubelet)
```

```
sudo journalctl -u kubelet
```

```
#          Log          .
hi POD      Container      .
/pause      가 Pod          . nginx      가
```

```
$ sudo docker ps -a | grep -i hi
```

```
d507d8b298c3    nginx          "/docker-
entrypoint...."    26 hours ago    Up 26 hours
k8s_hi_hi_default_6a1464a1-0fea-4ff8-a5c6-426afe281173_0
d8fb1a992247    k8s.gcr.io/pause:3.6    "/pause"
26 hours ago    Up 26 hours
k8s_POD_hi_default_6a1464a1-0fea-4ff8-a5c6-426afe281173_0
```

```
# nginx Service Container info
```

```
$ sudo docker inspect d507d8b298c3
```

```
$ sudo docker logs d507d8b298c3
```

```
# Pod Network info
```

```
$ sudo docker inspect d8fb1a992247
```

```
$ sudo docker logs d8fb1a992247
```

```
# Container
```

```
$ sudo docker exec -it d507d8b298c3 ls
```

가


```
bin    docker-entrypoint.d    home    media    proc    sbin    tmp
boot  docker-entrypoint.sh    lib     mnt     root    srv     usr
dev    etc                      lib64   opt     run     sys     var
```

```
$ sudo docker exec -it d507d8b298c3 /bin/bash
root@hi:/#
```

```
#          Net          가          nsenter          .
ip
$ $ sudo docker exec -it d507d8b298c3 ip addr
OCI runtime exec failed: exec failed: container_linux.go:380:
starting container process caused: exec: "ip": executable file
not found in $PATH: unknown
```

```
PID
$ sudo docker inspect --format '{{ .State.Pid }}' d507d8b298c3
1244489
```

```
nsenter          PID          (pod namespace가
)          가          .
$ sudo nsenter -t 1244489 -n ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group
default qlen 1000
    link/ipip 0.0.0.0 brd 0.0.0.0
4: eth0@if12: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1480 qdisc
noqueue state UP group default
    link/ether c6:3d:04:5d:80:82 brd ff:ff:ff:ff:ff:ff link-
netnsid 0
    inet 192.168.136.5/32 scope global eth0
        valid_lft forever preferred_lft forever
```

Pod delete

```
#          Pod
$ kubectl delete hi
```

```
# Pod Delete. replicas 가 .
$ kubectl delete deployment web
deployment.apps "web" deleted
$ kubectl get pods -o wide | grep -i web
-
```

Pod : <https://kubernetes.io/docs/concepts/workloads/pods/>
Pod Networking :
<https://www.digitalocean.com/community/tutorials/how-to-inspect-kubernetes-networking>