# ENTORN ELK (1 master – 2 esclaus)

## INSTANCIES AWS – ELASTICSEARCH & KIBANA

Al compte de AWS ja estaran configurats les instancies però per els esclaus s’han de crear la imatge de Docker perquè depenen de la IP i no la tenim estàtica. MASTER=Elasticsearch\_TEST Esclau1= Elasticsearch\_node2 Esclau2= Elasticsearch\_node3.

### MASTER

El màster ja estaria configurat al tenir una IP fixa. Però si fos necessari crear la imatge de Docker, els sources estan a: “/home/ec2-user/AWS/docker\_full\_elasticsearch\_ssh\_marvel” i per compilar seria “docker build -t elasticsearch\_ssh\_marvel .”.

La comanda per córrer per primera vegada el màster seria: “docker run -d --ulimit nofile=98304:98304 --name some-elasticsearch\_ssh\_marvel -p 9200:9200 -p 9300:9300 -p 2222:22 -p 8888:8888 -v /home/ec2-user/AWS/Elasticsearch/ElasticsearchData:/usr/share/elasticsearch/data elasticsearch\_ssh\_marvel”, però en principi això no es necessari i únicament seria necessari arrancar el Docker amb start:

sudo docker start some-elasticsearch\_ssh\_marvel some-kibana

\*\*En cas de que no funciones Kibana s’eliminaria i es crearia de nou la imatge: “sudo docker rm –f some-kibana” i després eliminar la imatge “sudo docker rmi kibana\_marvel” i crear de nou el Kibana:

“sudo docker build -t kibana\_marvel /home/ec2-user/AWS/docker\_kibana\_marvel “ i “docker run -d --name some-kibana --link some-elasticsearch\_ssh\_marvel:elasticsearch -p 5601:5601 kibana\_marvel”.

### ESCLAUS

Els esclaus depenen de IPs i al no tenir IP fixa(es podria configurar i no es necesitaria tot això) s’ha de crear una imatge nova cada cop que s’arranca la maquina. Primerament s’ha de “compilar” la imatge Docker, però abans s’ha de canviar la IP que es trobar a:

/home/ec2-user/AWS/docker\_full\_elasticsearch\_ssh\_marvel/config/elasticsearch.yml

Es mira la IP de la EC2 de AWS y es canvia la línia del arxiu:

network.host: 0.0.0.0

network.publish\_host: **IP\_AWS\_ESCLAU**

cluster.name: mycluster

node.name: node\_${HOSTNAME}

node.master: false

node.data: true

http.port : 9200

tcp.port : 9300

discovery.zen.ping.multicast.enabled: false

discovery.zen.ping.unicast.hosts: ["52.203.178.151:9300"]

Un cop configurat el arxiu es compila la imatge: **(\*POSIBLE ERROR)**

sudo docker build -t elasticsearch\_ssh\_marvel /home/ec2-user/AWS/docker\_full\_elasticsearch\_ssh\_marvel

Seguidament s’ha de executar el Docker:

sudo docker run --ulimit nofile=98304:98304 --name some-elasticsearch\_ssh\_marvel -p 9200:9200 -p 9300:9300 -p 2222:22 -p 8888:8888 -v /home/ec2-user/AWS/Elasticsearch/ElasticsearchData:/usr/share/elasticsearch/data elasticsearch\_ssh\_marvel

\*POSIBLE ERROR: que la imatge ja estigui creada i un Docker de la imatge també estigui creat per això primer s’ha de eliminar el Docker: sudo docker rm -f some-elasticsearch\_ssh\_marvel” i després eliminar la imatge “sudo docker rmi elasticsearch\_ssh\_marvel”

## INSTANCIES AWS – LOGSTASH

### CREAR EC2 AWS – LOGSTASH

Es crea la maquina EC2 en AWS. Després s’instal·larà Dockers es descarregarà de github i es prepararà l’entorn:

#Instalar Dockers

sudo yum update -y

sudo yum install -y docker

sudo service docker start

# Instalar Github y descargar de nuestra repo

sudo yum install -y git

git clone https://github.com/jcanopui/awsELK

mv /home/ec2-user/awsELK/AWS /home/ec2-user/AWS/

sudo rm -r /home/ec2-user/awsELK/

######

#Generar los ficheros

sudo rm -rf /home/ec2-user/AWS/logstash

mv /home/ec2-user/AWS/logstash\_pruebas\_de\_carga/logstash/ /home/ec2-user/AWS/logstash

chmod -x /home/ec2-user/AWS/logs/log\_peticiones\_exactas/log\_peticiones\_exactas.jar

java -jar /home/ec2-user/AWS/logs/log\_peticiones\_exactas/log\_peticiones\_exactas.jar peticiones /home/ec2-user/AWS/logs/log\_peticiones\_exactas/services.properties 25000

mv /home/ec2-user/AWS/logstash/logs/rastreator-delta.log.1 /home/ec2-user/AWS/logstash/logs/rastreator-delta1.log

mv /home/ec2-user/AWS/logstash/logs/rastreator-delta.log.2 /home/ec2-user/AWS/logstash/logs/rastreator-delta2.log

mv /home/ec2-user/AWS/logstash/logs/rastreator-delta.log.3 /home/ec2-user/AWS/logstash/logs/rastreator-delta3.log

mkdir /home/ec2-user/AWS/logstash/logs/logstash1\_1kk\_1

mkdir /home/ec2-user/AWS/logstash/logs/logstash2\_1kk\_1

cp /home/ec2-user/AWS/logstash/logs/rastreator-delta1.log /home/ec2-user/AWS/logstash/logs/rastreator-delta2.log /home/ec2-user/AWS/logstash/logs/rastreator-delta3.log /home/ec2-user/AWS/logstash/logs/rastreator-delta.log /home/ec2-user/AWS/logstash/logs/logstash1\_1kk\_1/

mv /home/ec2-user/AWS/logstash/logs/rastreator-delta1.log /home/ec2-user/AWS/logstash/logs/rastreator-delta2.log /home/ec2-user/AWS/logstash/logs/rastreator-delta3.log /home/ec2-user/AWS/logstash/logs/rastreator-delta.log /home/ec2-user/AWS/logstash/logs/logstash2\_1kk\_1/

### PROVA DE CARREGA

Per les proves de carrega un cop tenim les EC2 creades i preparades s’ha de configurar. Es tindra els logs en dos carpetes “logstash1\_1kk\_n” y “logstash2\_1kk\_n” on la n seria el numero de la la prova. Tot això es per que no hi hagi problemes amb les variables de entorn del docker i s’enviïn totes les traces.

També s’ha de configurar en els dos fitxers “logstash1\_1kk.conf” y “logstash2\_1kk.conf”.

Cada un estarà dirigit a un directori:

input {

stdin {

type => "stdin-type"

}

file {

type => "syslog-ng-3"

# Wildcards work, here :)

path => [ "/config-dir/logs/**logstash1\_1kk\_1**/\*.log"]

start\_position => "beginning"

# sincedb\_path => "/dev/null"

}

}

També és important posar la IP a la que enviarem els logs:

output {

elasticsearch { hosts => ["ec2-52-203-178-151.compute-1.amazonaws.com:9200"] }

}

Un cop configurat seria nomes arrancar el Logstash:

sudo docker run -it --name logstash1 --rm -v /home/ec2-user/AWS/logstash:/config-dir logstash logstash -f /config-dir/logstash1\_1kk.conf

sudo docker run -it --name logstash2 --rm -v /home/ec2-user/AWS/logstash:/config-dir logstash logstash -f /config-dir/logstash2\_1kk.conf

Per repetir les proves s’ha de clicar Control+C en les dos terminals y seguidament Control+C per tancar el Docker del tot. Després s’ha de renombrar les carpetes:

“/home/ec2-user/AWS/logstash/logs/logstash1\_1kk**\_1**” -> “/home/ec2-user/AWS/logstash/logs/logstash1\_1kk**\_2**”

“/home/ec2-user/AWS/logstash/logs/logstash2\_1kk**\_1**”-> “/home/ec2-user/AWS/logstash/logs/logstash2\_1kk**\_2**”

Un cop canviat els noms, s’ha de canviar la configuració:

logstash2\_1kk.conf y logstash1\_1kk.conf on subtituirem: “path => [ "/config-dir/logs/logstash1\_1kk\_1/\*.log"]” per “path => [ "/config-dir/logs/logstash1\_1kk\_2/\*.log"]” i “path => [ "/config-dir/logs/logstash2\_1kk\_1/\*.log"]” per “path => [ "/config-dir/logs/logstash2\_1kk\_2/\*.log"]” i ja es podria tornar a executar els Logstash.

# Possibles problemes

## GENERALES

Al executar la comanda Docker es possible que respongui:

“docker: Cannot connect to the Docker daemon. Is the docker daemon running on this host?.”

S’afegeix “sudo” davant de la comanda i ja funcionaria..

## NO ENVIA TRACES LOGSTASH

En cas de que Logstash no envií traces la manera d’arreglar-lo seria eliminar tot lo que esta relacionat amb el programa Dockers:

#Unistall Docker

sudo service docker stop

sudo yum erase -y docker

sudo rm -r /var/lib/docker/

E instal·lar de nou:

#Install Docker

sudo yum update -y

sudo yum install -y docker

sudo service docker start