Ricardo Delgado – A00268540  
<https://github.com/rocco522/all-exams/tree/master/2017_b/sd/exam1>

# Parcial 1 Sistemas Distribuidos

**Consigne los comandos de Linux necesarios para el aprovisionamiento de los servicios solicitados.**

Para la instalación de ELK se han usado los siguientes comandos y se han consignado en los scripts de cada servidor (fuente: <https://github.com/ICESI/so-monitoring>)

Aprovisionamiento de Elasticsearch

Instalar el openjdk de Java

yum install java-1.8.0-openjdk.x86\_64

Descargar e instalar la llave pública

rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch

Crear el archivo con el repositorio de elasticsearch

vi /etc/yum.repos.d/elasticsearch.repo

[elasticsearch-5.x]

name=Elasticsearch repository for 5.x packages

baseurl=https://artifacts.elastic.co/packages/5.x/yum

gpgcheck=1

gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch

enabled=1

autorefresh=1

type=rpm-md

Instalar elasticsearch

yum install elasticsearch

Aprovisionamiento de logstash

Crear el archivo con el repositorio de logstash

vi /etc/yum.repos.d/logstash.repo

[logstash-5.x]

name=Elastic repository for 5.x packages

baseurl=https://artifacts.elastic.co/packages/5.x/yum

gpgcheck=1

gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch

enabled=1

autorefresh=1

type=rpm-md

Instalar logstash

yum install logstash

Aprovisionamiento de Kibana

Crear el archivo con el repositorio de kibana

vi /etc/yum.repos.d/kibana.repo

[kibana-5.x]

name=Kibana repository for 5.x packages

baseurl=https://artifacts.elastic.co/packages/5.x/yum

gpgcheck=1

gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch

enabled=1

autorefresh=1

type=rpm-md

Instalar kibana

yum install kibana

Instalación de filebeat (Cliente)

Descargar e instalar la llave pública

sudo rpm --import https://packages.elastic.co/GPG-KEY-elasticsearch

Crear el archivo con el repositorio de filebeat

vi /etc/yum.repos.d/elastic.repo

[elastic-5.x]

name=Elastic repository for 5.x packages

baseurl=https://artifacts.elastic.co/packages/5.x/yum

gpgcheck=1

gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch

enabled=1

autorefresh=1

type=rpm-md

Instalar filebeat

sudo yum install filebeat

Iniciar filebeat en el arranque

sudo chkconfig --add filebeat

**Escriba el archivo Vagrantfile para realizar el aprovisionamiento, teniendo en cuenta definir: maquinas a aprovisionar, interfaces solo anfitrión, interfaces tipo puente, declaración de cookbooks, variables necesarias para plantillas**

# -\*- mode: ruby -\*-

# vi: set ft=ruby :

VAGRANTFILE\_API\_VERSION = "2"

Vagrant.configure(VAGRANTFILE\_API\_VERSION) do |config|

config.ssh.insert\_key = false

config.vm.define :elastic\_search do |es|

es.vm.box = "centos/7"

es.vm.network :private\_network, ip: "192.168.56.101"

#es.vm.network "public\_network", bridge: "eth4", ip:"192.168.131.100", netmask: "255.255.255.0"

es.vm.provider :virtualbox do |vb|

vb.customize ["modifyvm", :id, "--memory", "1024","--cpus", "1", "--name", "es" ]

end

#config.vm.provision "shell", inline: <<-SHELL

# sudo yum install screen -y

#SHELL

config.vm.provision :chef\_solo do |chef|

chef.install = false

chef.cookbooks\_path = "cookbooks"

chef.add\_recipe "elastic\_search"

#chef.add\_recipe "haproxy"

#chef.json = {"courses":[

#{"id":"001", "name":"Sistemas distribuídos-s1" },

#{"id":"002", "name":"Redes convergentes-s1" },

#{"id":"003", "name":"Computación en la nube-s1" },

#{"id":"004", "name":"Sistemas operativos-s1" }

#]}

chef.json = {"ip" => "192.168.56.101"}

end

end

config.vm.define :logstash\_server do |ls|

ls.vm.box = "centos/7"

ls.vm.network :private\_network, ip: "192.168.56.102"

#wb.vm.network "public\_network", bridge: "eth4", ip:"192.168.131.100", netmask: "255.255.255.0"

ls.vm.provider :virtualbox do |vb|

vb.customize ["modifyvm", :id, "--memory", "1024","--cpus", "1", "--name", "ls" ]

end

config.vm.provision :chef\_solo do |chef|

chef.install = false

chef.cookbooks\_path = "cookbooks"

chef.add\_recipe "logstash"

chef.json = {"ip" => "192.168.56.102"}

end

end

config.vm.define :kibana\_server do |ks|

ks.vm.box = "centos/7"

ks.vm.network :private\_network, ip: "192.168.56.103"

#wb.vm.network "public\_network", bridge: "eth4", ip:"192.168.131.100", netmask: "255.255.255.0"

ks.vm.provider :virtualbox do |vb|

vb.customize ["modifyvm", :id, "--memory", "1024","--cpus", "1", "--name", "ks" ]

end

config.vm.provision :chef\_solo do |chef|

chef.install = false

chef.cookbooks\_path = "cookbooks"

chef.add\_recipe "kibana"

#chef.add\_recipe "haproxy"

chef.json = {"ip" => "192.168.56.103", "url" => "http://192.168.56.101:9200"}

end

end

config.vm.define :web\_server do |wb1|

wb1.vm.box = "centos/7"

wb1.vm.network :private\_network, ip: "192.168.56.104"

#wb.vm.network "public\_network", bridge: "eth4", ip:"192.168.131.100", netmask: "255.255.255.0"

wb1.vm.provider :virtualbox do |vb|

vb.customize ["modifyvm", :id, "--memory", "1024","--cpus", "1", "--name", "wb1" ]

end

config.vm.provision :chef\_solo do |chef|

chef.install = false

chef.cookbooks\_path = "cookbooks"

chef.add\_recipe "httpd"

chef.add\_recipe "filebeat"

chef.json = {"ip" => "192.168.56.104"}

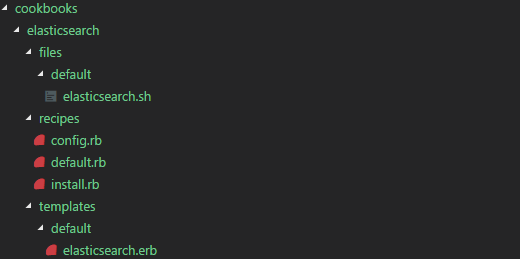
end

end

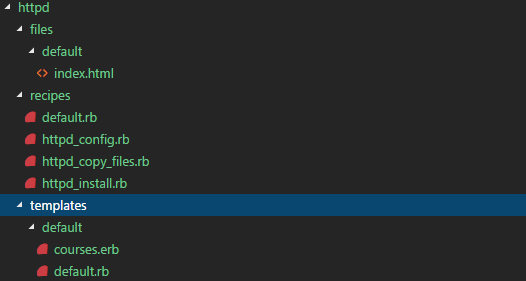
end

**Escriba los cookbooks necesarios para realizar la instalación de los servicios solicitados**

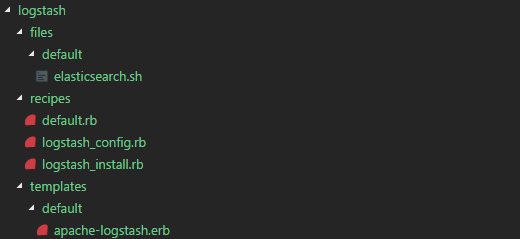
Los cookbooks utilizados para la instalación de los servicios, son los siguientes:







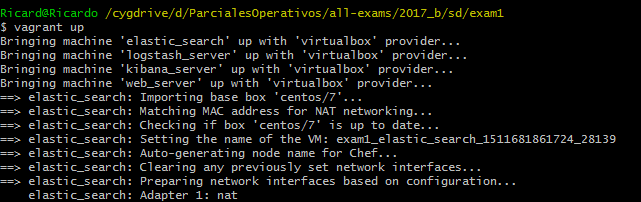


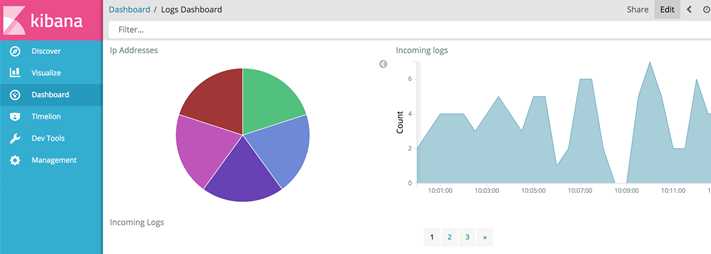


**El informe debe publicarse en un repositorio de github el cual debe ser un fork de https://github.com/ICESI-Training/sd-exam1 y para la entrega deberá hacer un Pull Request (PR) respetando la estructura definida. El código fuente y la url de github deben incluirse en el informe (15%). Tenga en cuenta publicar los archivos para el aprovisionamiento**

<https://github.com/rocco522/all-exams/tree/master/2017_b/sd/exam1>

**Incluya evidencias que muestran el funcionamiento de lo solicitado**





**Documente algunos de los problemas encontrados y las acciones efectuadas para su solución al aprovisionar la infraestructura y aplicaciones**

Tuve problemas aprovisionando los servicios utilizando bash, pues no tenía mucha experiencia creando los scripts que corrieran de manera automática.

Hubo problemas en el funcionamiento de vagrant en Windows que no logré resolver. Tuve que trabajar sobre una VM Ubuntu.