

# Parcial 2

## Sistemas Distribuidos

**Nombre Estudiante:** Emmanuel Bolaños Ocampo

**Código Estudiante:** A00309828

**Repositorio:** <https://github.com/emmanuel8664/sd-exam2/blob/master/A00309828/Informe.md>

### Examen 2

**Universidad ICESI**

**Curso:** Sistemas Distribuidos

**Docente:** Daniel Barragán C.

**Tema:** Automatización de infraestructura (Docker)

**Correo:** daniel.barragan at correo.icesi.edu.co

### Objetivos

- Realizar de forma autónoma el aprovisionamiento automático de infraestructura
- Diagnosticar y ejecutar de forma autónoma las acciones necesarias para lograr infraestructuras estables
- Integrar servicios ejecutándose en nodos distintos

### Prerrequisitos

- Docker
- docker-compose

### Nota Importante

El parcial debe realizarse de forma individual, en caso de encontrar copia a otros compañeros se anulará su examen. Podrá usar cualquier material de clase o realizar consultas en Internet. Una vez termine correctamente la parte práctica se le pedirá

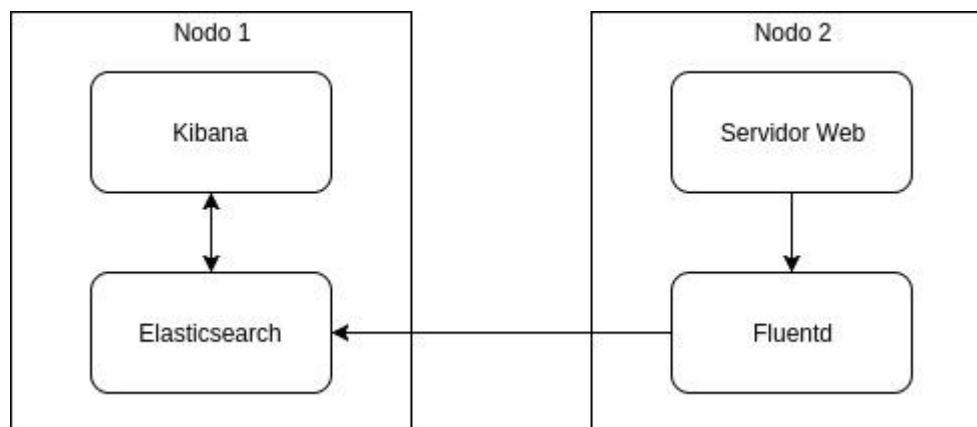
que realice la respectiva sustentación de su solución, esta sustentación tendrá un factor de 0 a 1 sobre el valor del punto.

## Descripción

El stack ELK es un paquete de tres herramientas open source de la empresa Elastic. Las herramientas son Elasticsearch, Logstash y Kibana. Estas tres herramientas son proyectos independientes pero pueden ser usadas en conjunto para desplegar un ambiente de monitoreo de infraestructura.

Deberá realizar el aprovisionamiento de un ambiente compuesto por los siguientes elementos:

- En un servidor ejecutar: Un contenedor encargado de almacenar logs por medio de la aplicación Elasticsearch, un contenedor con la herramienta encargada de visualizar la información de los logs por medio de la aplicación Kibana
- En uno o varios servidores ejecutar: un contenedor web y un contenedor encargado de hacer la conversión de logs por medio de la aplicación Fluentd



En el repositorio de github del curso se encuentran ejemplos de docker y docker-compose los cuales pueden ser consultados para construir su solución.

**Consigne los comandos de Linux necesarios para el aprovisionamiento de los servicios solicitados. En este punto no debe incluir archivos tipo Dockerfile solo se requiere que usted identifique los comandos o acciones que debe automatizar**

```
docker run -d --name=elastic elasticsearch
```

```
docker run -d --name=kibana --link elastic:elasticsearch -p 5601:5601 kibana
```

```
FROM fluent/fluentd:v0.12
```

```
RUN ["gem", "install", "fluent-plugin-elasticsearch", "--no-rdoc", "--no-ri",  
"--version", "1.9.5"]
```

```
docker build -t fluentd-elastic .
```

```
docker run --name=fluentd --link elastic:elasticsearch -v  
/fluentd/conf:/fluentd/etc -e FLUENTD_CONF=fluent.conf fluentd-elastic
```

```
docker run -d -p 80:80 --name=apache --log-driver=fluentd --log-opt  
fluentdaddress=172.17.0.4:24224 --log-opt tag=httpd.access httpd
```

**Escriba los archivos Dockerfile para cada uno de los servicios solicitados junto con los archivos fuente necesarios. Tenga en cuenta consultar buenas prácticas para la elaboración de archivos Dockerfile.**

```
FROM fluent/fluentd:v0.12
```

```
RUN ["gem", "install", "fluent-plugin-elasticsearch", "--no-rdoc", "--no-ri",  
"--version", "1.9.5"]
```

**Escriba el archivo docker-compose.yml necesario para el despliegue de la infraestructura. No emplee configuraciones deprecated.**

```
version: '2'  
services:  
  web:  
    image: httpd  
    ports:  
      - "80:80"  
    links:  
      - fluentd  
    logging:  
      driver: "fluentd"  
    options:  
      fluentd-address: localhost:24224  
      tag: httpd.access  
    fluentd:  
      build: ./fluentd  
    volumes:  
      - ./fluentd/conf:/fluentd/etc  
    links:  
      - "elasticsearch"  
    ports:
```

- "24224:24224"
- "24224:24224/udp"

elasticsearch:

image: elasticsearch

kibana:

image: kibana

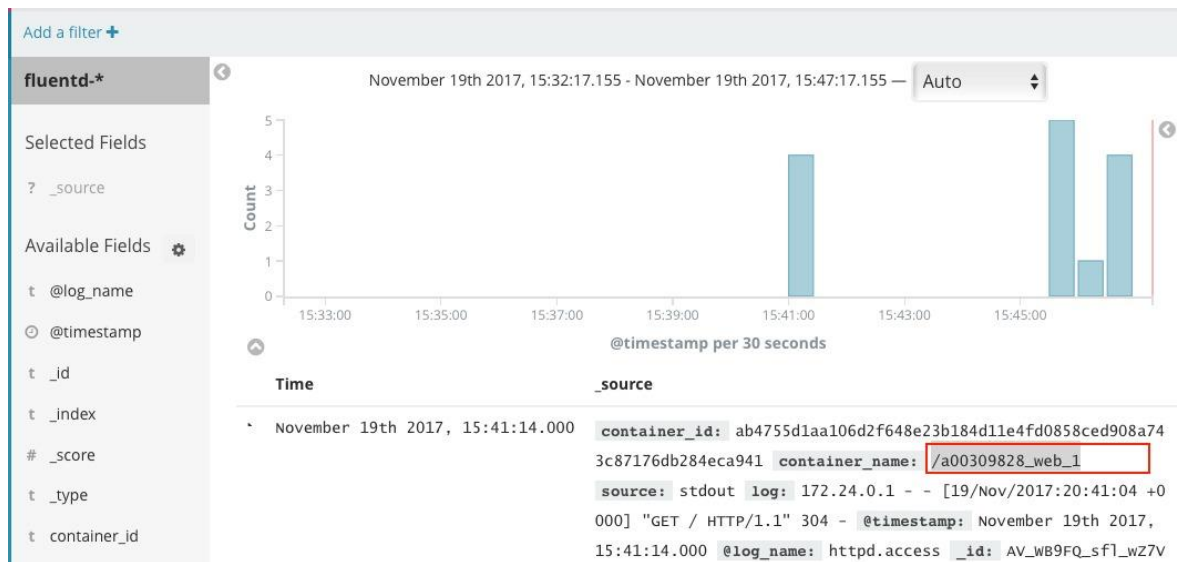
links:

- "elasticsearch"

ports:

- "5601:5601"

**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.**

Cuando se quieren enviar Logs a fluentd es necesario conocer la ip del contenedor de fluentd. En principio, se intentó usar la opción de "link" de docker pero no dio resultado. Como solución se decidió exponer el servicio de fluentd a localhost

```

fluentd:
  build: ./fluentd
  volumes:
    - ./fluentd/conf:/fluentd/etc
  links:
    - "elasticsearch"
  ports:
    - "24224:24224"

```

... y hacer referencia a este en vez de al contenedor.

```
web:
  image: httpd
  ports:
    - "80:80"
  links:
    - fluentd
  logging:
    driver: "fluentd"
    options:
      fluentd-address: localhost:24224
      tag: httpd.access
```