



**Sistema modular de adquisición, gestión y visualización web de  
métricas de desempeño para centros de cómputo de alto  
rendimiento.  
Caso de estudio: CECAD**

**Steven Sierra Forero**

Universidad Distrital Francisco José de Caldas.  
Ingeniería de Sistemas  
Bogotá, Colombia  
Marzo, 2019



**Sistema modular de adquisición, gestion y visualización web de  
métricas de desempeño para centros de cómputo de alto  
rendimiento.Caso de estudio: CECAD**

**Steven Sierra Forero**

Tesis o trabajo de grado presentada(o) como requisito parcial para optar al título de:  
**Ingeniero de Sistemas**

Director(a):  
PhD. Nelson Enrique Vera Parra

Línea de Investigación:  
Nombrar la línea de investigación en la que enmarca la tesis o trabajo de investigación  
Grupo de Investigación:  
Nombrar el grupo en caso que sea posible

Universidad Distrital Francisco José de Caldas.  
Facultad, Ingeniería de Sistemas  
Bogota, Colombia  
2019



## Dedicatoria

A mi hija: Abby Sierra Cubillos.

La preocupación por el hombre y su destino siempre debe ser el interés primordial de todo esfuerzo técnico. Nunca olvides esto entre tus diagramas y ecuaciones.

Albert Einstein



# Agradecimientos

Esta sección es opcional, en ella el autor agradece a las personas o instituciones que colaboraron en la realización de la tesis o trabajo de investigación. Si se incluye esta sección, deben aparecer los nombres completos, los cargos y su aporte al documento.

Al profesor Nelson Vera Parra por la paciencia.





## Resumen

El objeto del presente proyecto es el diseño, implementación y evaluación de un sistema modular de adquisición, gestión y visualización web de métricas de rendimiento para centros de cómputo de alto desempeño. Métricas tales como: uso de CPU, utilización de memoria RAM, espacio en disco, conteo de procesos, tráfico de red, operaciones de I/O en disco, cantidad de hilos, uso de memoria en GPU serán gestionadas mediante un motor de búsqueda como Elasticsearch y serán visualizadas haciendo uso de herramientas como Kibana. El sistema tendrá un diseño de naturaleza modular con el fin de adaptarse a las diferentes naturalezas de los centros de cómputo de alto desempeño y a su continuo crecimiento e inclusión de nuevas tecnologías. La implementación realizada será un caso de estudio específicamente para el CECAD.

# Contenido

<b>Agradecimientos</b>	<b>VII</b>
<b>Lista de Figuras</b>	<b>XII</b>
<b>Lista de Tablas</b>	<b>1</b>
<b>1 Introducción</b>	<b>2</b>
1.1 ¿Qué es un centro de computación de alto desempeño?	2
1.2 ¿Por que es importante monitorear el desempeño en centros de cómputo de alto desempeño?	3
1.3 Objetivos	3
1.4 Pregunta problema e hipótesis de la investigación	4
1.5 Como se organiza esta investigación	5
<b>2 Centros de Computación de Alto desempeño</b>	<b>6</b>
2.1 Centros de Computación de Alto desempeño	6
2.2 CECAD, Centro de Computación de Alto Desempeño de la Universidad Distrital Francisco José de Caldas	7
<b>3 Elastic Search, Logstash y Kibana</b>	<b>8</b>
3.1 Elastic Search	8
3.2 Logstash	8
3.2.1 Arquitectura de Logstash	8
3.3 Kibana	9
<b>4 Arquitectura</b>	<b>10</b>
4.1 Recolección de Métricas	13
4.1.1 Tipos de Métricas	13
4.1.2 Metricbeat	13
4.1.3 Collectd	13
4.1.4 Statsd	13
4.2 Gestión de Métricas	13
4.3 Persistencia	14
4.4 Comunicación	15

4.5	Protocolos de operabilidad . . . . .	15
4.5.1	Operabilidad en Red . . . . .	15
4.6	Visualización . . . . .	16
<b>5</b>	<b>Implementación de SMMD</b>	<b>17</b>
5.0.1	Requisitos mínimos del sistema . . . . .	17
5.1	Fuentes de Instalación . . . . .	18
5.2	Instalación . . . . .	19
5.2.1	Instalación del Servidor SMMD . . . . .	19
5.2.2	Aprovisionamiento de Nodos . . . . .	21
5.3	Uso . . . . .	23
5.3.1	Visualización de Métricas de Desempeño . . . . .	23
5.3.2	Ajustes generales . . . . .	24
5.3.3	Plugins . . . . .	24
<b>6</b>	<b>Evaluación de SMMD</b>	<b>25</b>
6.0.1	Caso práctico CECAD . . . . .	25
<b>7</b>	<b>Conclusiones</b>	<b>26</b>
<b>8</b>	<b>Principales aportes y divulgación</b>	<b>27</b>
<b>9</b>	<b>Recomendaciones y Futuros Trabajos</b>	<b>28</b>
9.1	tal . . . . .	28
	<b>Glossary</b>	<b>29</b>
<b>10</b>	<b>Referencias</b>	<b>30</b>

# Lista de Figuras

3-1	Logstash. . . . .	8
4-1	Modelo General del Sistema SMMD. . . . .	12
4-2	Comunicacion Cliente - Servidor. . . . .	16
5-1	Instalación de SMMD en Nodos Cliente. . . . .	18
5-2	Ejemplo, resultado de la ejecución de docker-compose up, iniciación de servicios	20
5-3	Pantalla de inicio de sesión en Kibana . . . . .	20
5-4	Instalación de SMMD en Nodos Cliente. . . . .	21
5-5	Aprovisionamiento de SMMD en los nodos cliente. . . . .	22
5-6	Pantalla Principal de SMMD . . . . .	23
6-1	Dashboard Personalizado para la medición de métricas recibidas por StatsD	25

# Lista de Tablas

1-1 Organización de la Investigación . . . . .	5
--	---

# 1 Introducción

## 1.1. ¿Qué es un centro de computación de alto desempeño?

El término “Computación de Alto desempeño” hace referencia a la distribución de la carga computacional para la solución de problemas científicos haciendo uso de tecnologías que implementan la distribución de carga en un conjunto de recursos computacionales coordinados llamados clusters. De acuerdo a (*What is high performance computing?*, 2019) la computación de alto desempeño generalmente se refiere a la práctica de agregar poder computacional de tal forma que este entrega mucho más rendimiento que cualquier recurso de computo individual con el propósito de solucionar problemas en diversos campos como la ingeniería, la ciencias básicas y la economía.

Como cualquier recurso computacional un sistema de computación de alto desempeño debe abordarse desde sus componentes más básicos, en esencia se puede comprender desde los componentes de cualquier sistema de computo en términos de sus elementos de hardware tales como memoria, procesador, almacenamiento y sus elementos de software tales como sistema operativo y aplicaciones de usuario. Partiendo desde estos componentes básicos, podemos afirmar que un centro de computación de alto desempeño es la suma de todos los recursos de hardware de diferentes nodos los cuales a través de un circuito lógico construido por el sistema operativo, la comunicación en red y el software componen un único sistema multipropósito, en el cual se pueden procesar grandes volúmenes de información, ejecutar diversos algoritmos y procesamiento de cálculos con gran velocidad y alto rendimiento.

Para dar idea de la magnitud de recursos de hardware que se pueden conformar en un centro de computo de alto desempeño, para Noviembre de 2018, en los Estados Unidos la supercomputadora número 1 Power System AC922 de IBM tiene 2,397,824 núcleos de procesador y una memoria RAM de 2,801,664 Gigabytes alcanzando un máximo teórico de 200,795 Tera flops por segundo (*Top 500 The List*, 2019) lo que da cuenta del poder computacional con el que se pueden solucionar diversos problemas de la ciencia y la ingeniería.

## **1.2. ¿Por que es importante monitorear el desempeño en centros de cómputo de alto desempeño?**

Los centros de cómputo de alto desempeño son recursos totalmente necesarios en entidades académicas, investigativas o comerciales, que requieran el almacenamiento y procesamiento de altos volúmenes de datos que exijan tareas intensivas y complejas. El potencial de cada uno de esos centros de cómputo se basa en las especificaciones y configuración de su infraestructura y en el buen uso que se haga de ésta por parte de las aplicaciones que se ejecutan allí. La gestión y administración de un centro de cómputo de alto rendimiento y la alta eficiencia con la cual las aplicaciones sacan provechos de la infraestructura depende en gran parte del control continuo que se tenga sobre el uso de los recursos con los cuales se cuenta, sin embargo este es uno de los problemas en los cuales los investigadores invierten mucho tiempo para encontrar herramientas de software que permitan la obtención de métricas básicas, las herramientas usadas son diversas y dependen de las necesidades del actor, de la infraestructura, de la tecnología y del software. Esto conlleva a que tanto la administración del centro como cada uno de los proyectos ejecutados en él instalen, configuren y en algunos casos desarrollen herramientas específicas de monitoreo de métricas de desempeño agregando esfuerzo y tiempo que podría ser utilizado en la resolución del problema científico, ejemplo evidente en el caso de los proyectos de bioinformática desarrollados en el convenio del GICOGÉ con el Instituto de Genética de la Universidad Nacional.

## **1.3. Objetivos**

El objeto del presente proyecto es el diseño, implementación y evaluación de un sistema modular de adquisición, gestión y visualización web de métricas de rendimiento para centros de cómputo de alto desempeño. Métricas tales como: uso de CPU, utilización de memoria RAM, espacio en disco, conteo de procesos, tráfico de red, operaciones de I/O en disco, cantidad de hilos, uso de memoria en GPU serán gestionadas mediante un motor de búsqueda como Elasticsearch y serán visualizadas haciendo uso de herramientas como Kibana. El sistema tendrá un diseño de naturaleza modular con el fin de adaptarse a las diferentes naturalezas de los centros de cómputo de alto desempeño y a su continuo crecimiento e inclusión de nuevas tecnologías. La implementación realizada será un caso de estudio específicamente para el CECAD. Crear un modelo de sistema modular para la adquisición, gestión y visualización web para métricas de desempeño en centros de cómputo de alto rendimiento. Construir una implementación del sistema modelado para el caso específico del CECAD. Evaluar el sistema implementado mediante su uso por parte de los proyectos vigentes en el CECAD. Generar proyecciones y prospectivas de uso e innovación (mejoras, optimización y transformación) del sistema.

## 1.4. Pregunta problema e hipótesis de la investigación

¿Qué especificaciones debe cumplir un sistema modular de adquisición, gestión y visualización web de métricas de desempeño para centros de cómputo de alto rendimiento?

Un sistema modular de adquisición, gestión y visualización, debe ser capaz de utilizarse de una manera sencilla y amigable, en donde el investigador no tenga la necesidad de invertir el tiempo en encontrar la combinación de herramientas y tecnologías que le permitan conocer métricas básicas necesarias para la medición y comparación de rendimiento en el procesamiento de datos y ejecución de algoritmos.

El sistema debe ser capaz de entregar métricas básicas tales como el uso de procesador, memoria, almacenamiento en una interfaz sencilla y de fácil acceso. Debido los intereses particulares entre quienes administran los centros de cómputo de alto rendimiento y quienes hacen uso de ellos, es necesario que la herramienta de adquisición, gestión y visualización sea independiente de tal manera que se pueda instalar, configurar y ejecutar con los ajustes que requiera el investigador sin la necesidad de interferir con el software de administración del centro de cómputo, permitiéndole al investigador trabajar de manera autónoma y aislada dando espacio para que diversos proyectos se ejecuten en el mismo centro de cómputo sin interferencia alguna.



## 1.5. Como se organiza esta investigación

El proyecto esta planteado en tres fases principales las cuales son: Diseño, Implemetación y Evaluación, dichas fases están organizadas en capítulos donde se describen las actividades y los modelos que componen el sistema de monitoreo y administración de métricas de desempeño.

Fase	Descripción	Capítulo
Diseño	Estado del arte sobre monitoreo en centros de cómputo de alto desempeño	1
	Definición de tecnologías y metodología	2
	Diseño de la arquitectura general del sistema.	3
Implementación	Configuración del entorno	4
	Instalación	4.1
	Ejecución de módulos y verificación	
Evaluación	Verificación del sistema y métricas	
	Conclusiones	1
	Aportes y divulgación	

**Tabla 1-1:** Organización de la Investigación

## 2 Centros de Computación de Alto desempeño

### 2.1. Centros de Computación de Alto desempeño

Un centro de computación de alto desempeño es una colección de computadoras denominadas nodos las cuales se encuentran interconectadas a través de una red de alta velocidad establecidas en un arquitectura específica, dichos nodos se clasifican según las tareas que se desarrollen en estos. Una de las principales motivaciones en el uso de Centros de Computación de alto desempeño, es la paralelización de algoritmos para la obtención del mejor desempeño, reduciendo tiempo de computo haciendo y haciendo un uso eficiente de los recursos en el procesamiento de grandes volúmenes de información.

En el área de la computación de alto desempeño se abarcan los campos científicos y técnicos en el estudio de las supercomputadoras (Nielsen, 2016) , el sitio Top 500 (*Top 500 The List*, 2019) mantiene una lista actualizada de las quinientas mejores computadoras en el mundo y las clasifica según su rendimiento, este es medido en FLOPS<sup>1</sup>, para Noviembre de 2018 la supercomputadora número 1 fue Power System AC922 de IBM con 2,397,824 núcleos de procesador y una memoria RAM de 2,801,664 Gigabytes alcanzando un máximo teórico de rendimiento computacional de 200,795 Tera flops por segundo.

---

<sup>1</sup>Floating point operations per second, Cantidad de operaciones aritméticas de punto flotante por segundo

## **2.2. CECAD, Centro de Computación de Alto Desempeño de la Universidad Distrital Francisco José de Caldas**

El centro de computación de alto desempeño de la Universidad Distrital Francisco José de Caldas, es un laboratorio destinado a la investigación en donde se realizan distintos experimentos y pruebas, este es un espacio destinado a la investigación y el aprendizaje, allí se realizan pruebas de nuevas tecnologías y se desarrollan distintos proyectos de investigación provenientes de los grupos de investigación de pregrado, maestría y doctorado.(Cáliz, 2019). La estructura general del CECAD está compuesta por la siguiente infraestructura:

1. Un cluster HPC homogéneo
2. Cluster HPC heterógeno respaldado por 1 nodo GPU con una tarjeta NVIDIA Tesla K-80 y 4 Servidores Intel Xeon Phi z200 7230.
3. Nube privada respaldada 36 equipos Dell R610, 4 equipos Dell R900 con un total de almacenamiento de 60 TB.
4. Nube privada de pruebas, esta nube se encuentra basada en openstackl, la cual se utilizara para probar nuevos servicios y actualizaciones.
5. Distema de archivos distribuido
6. Sistema de Respaldo implementado a través de un SAN Controller service.

## 3 Elastic Search, Logstash y Kibana

### 3.1. Elastic Search

Elasticsearch es un servidor de búsqueda basado en Lucene. Provee un motor de búsqueda de texto completo, distribuido y con capacidad de multi-tenencia con una interfaz web RESTful y con documentos JSON. Elasticsearch está desarrollado en Java y está publicado como código abierto bajo las condiciones de la licencia Apache. Elasticsearch puede ser usado para buscar todo tipo de documentos. La búsqueda es escalable y casi en tiempo real, soportando multitenencia. Es distribuido, haciendo que los índices se puedan dividir en fragmentos y cada uno teniendo cero o más réplicas. Cada nodo alberga uno o más fragmentos, actuando como un coordinador para delegar operaciones a los fragmentos correctos. El rebalanceo y ruteo se realizan automáticamente. Utiliza Lucene e intenta hacer todas sus funciones disponibles a través de JSON y Java API. Soporta facetado y percolación, que puede ser útil para notificar si nuevos documentos coinciden con consultas registradas. Otra funcionalidad llamada gateway maneja la persistencia a largo plazo del índice por ejemplo, se puede recuperar un índice del gateway en caso de una caída del servidor. Soporta peticiones GET en tiempo real y esto lo hace válido para una solución NoSQL pero carece de transacciones distribuidas.

### 3.2. Logstash

#### 3.2.1. Arquitectura de Logstash

Logstash es una herramienta para la administración de logs. Esta herramienta se puede utilizar para recolectar, parsear y guardar los logs para futuras búsquedas. La aplicación se encuentra basada en jRuby y requiere de Java Virtual Machine para correr. Como corre en

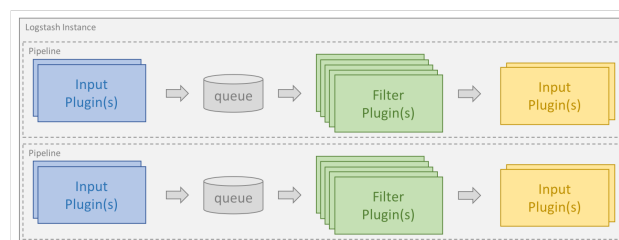


Figura 3-1: Logstash.

JVM puede ser ejecutada en cualquier Sistema Operativo que corra JVM (Linux, Mac OS X, Windows). En Logstash y con una infraestructura distribuida, cada servidor web debe ser configurado para correr Lumberjack (es opcional pero altamente recomendado para economizar recursos). Lumberjack hace un forward de los logs a un servidor corriendo Logstash con una entrada de Lumberjack. Como Lumberjack requiere SSL, los logs van a ser encriptados del servidor web al servidor de logs central. Logstash is an open source, server-side data processing pipeline that ingests data from a multitude of sources simultaneously, transforms it, and then sends it to your favorite “stash.” (Ours is Elasticsearch, naturally.) Logstash soporta un número de entradas, códecs, filtros y salidas. Las entradas son las fuentes de datos. Los códecs esencialmente convierten un formato de entrada en un formato aceptado por Logstash, así como también transforman del formato de Logstash al formato deseado de salida. Estos son utilizados comúnmente si la fuente de datos no es una línea de texto plano. Los filtros son acciones que se utilizan para procesar en los eventos y permiten modificarlos o eliminar eventos luego de ser procesados. Finalmente, las salidas son los destinos donde los datos procesados deben ser derivados.

### 3.3. Kibana

Kibana hace parte del grupo de aplicaciones liberadas por Elastic Search (Sharma, 2016) para el análisis de datos a través de logs, comúnmente es conocido como “ELK Stack”, el cual refiere al uso de Elastic Search y Kibana en conjunto para la presentación y análisis de información a través de tableros de mando con estadísticas y series de tiempo.

## 4 Arquitectura

El sistema modular para la adquisición, gestión y visualización web de métricas de desempeño, está compuesto por cuatro capas cada una con una función específica dentro de la representación y gestión de las métricas de desempeño que se presentan al usuario.

Los componentes del SMMD se clasifican de la siguiente manera:

### 1. Recolectores de métricas

- a) Collectd, un proceso que se ejecuta en segundo plano a nivel del nodo en donde se ejecutan los algoritmos o el procesamiento de datos, el cual se encarga de tomar todas las métricas de desempeño a través de plugins que se activan en un archivo de configuración.
- b) Metricbeat, un proceso que se ejecuta en segundo plano a nivel del nodo en donde se ejecutan los algoritmos o el procesamiento de datos el cual toma métricas básicas del sistema operativo y las envía al recolector de logs.

### 2. Recolector de Logs

- a) Logstash, es un proceso que se ejecuta en el servidor, en este caso en el equipo que utiliza el investigador al cual los nodos envían la información de métricas de desempeño.

### 3. Persistencia de Datos

- a) Elastic Search, representa la capa de persistencia de datos y procesamiento, una vez la información ha sido recibida desde Logstash, elastic search se encarga de procesar y persistir la información en un formato de documento de tipo JSON <sup>1</sup>

### 4. Presentación

- a) Kibana, es un visualizador de datos, que permite explorar la información que se encuentra en Elastic Search y provee herramientas de fácil uso para la creación de paneles de control y vistas generales para la visualización de métricas de desempeño. Utiliza los paneles de control pre-definidos por metric beat para representar el uso de recursos tales como memoria, procesador y almacenamiento con actualizaciones en tiempo real.

---

<sup>1</sup>JavaScript Object Notation, un formato de intercambio de datos basado en el estándar ECMA-262 del lenguaje de programación Javascript

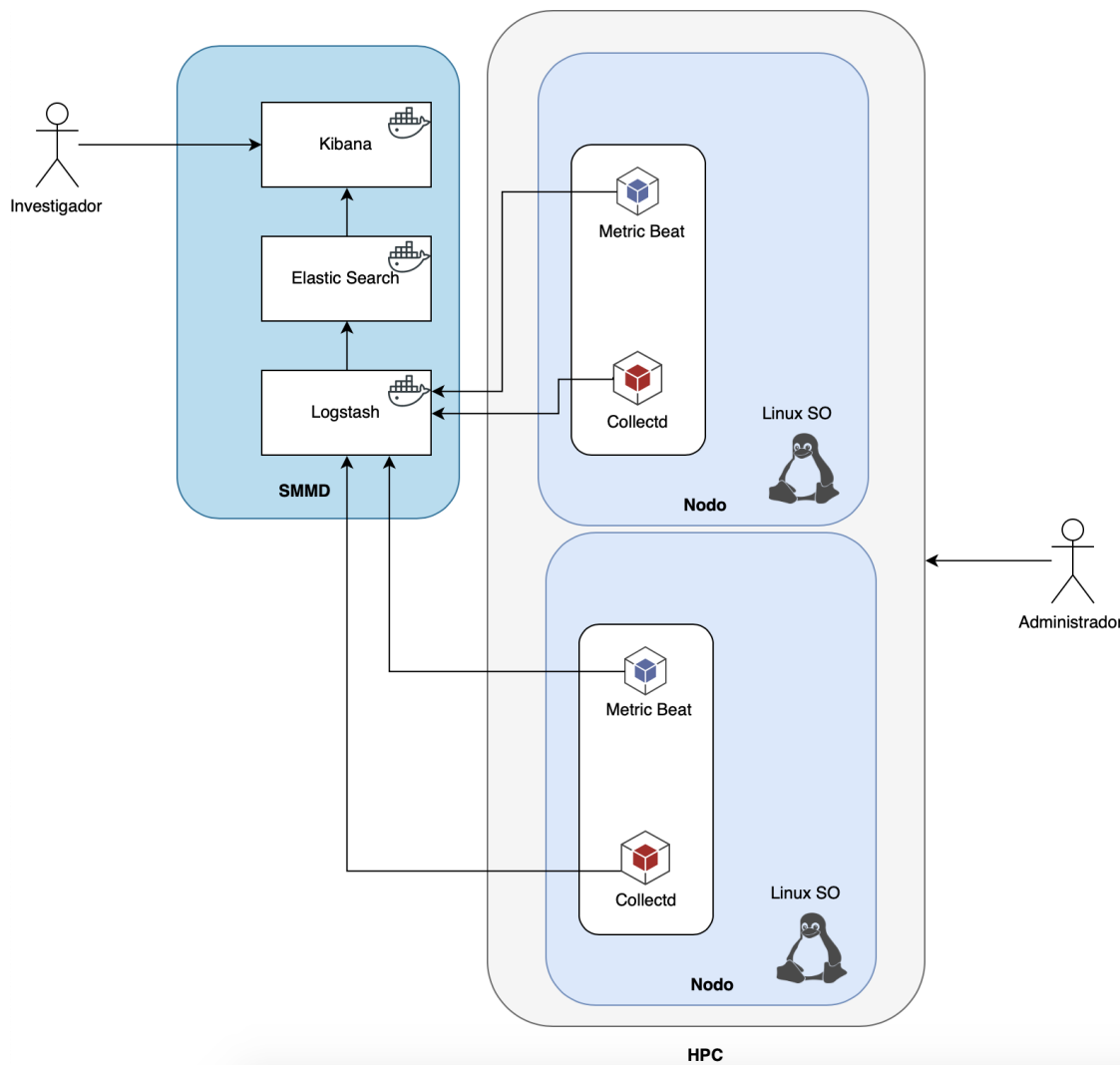
Tal como se puede observar en la figura 4-1 SMMD está modelado en una arquitectura de tipo cliente/servidor en donde el cliente está representado por los nodos que son asignados en el cluster al investigador para que este utilice sus recursos en la ejecución de programas o procesamiento de datos. El administrador provee de manera transparente al investigador dichos nodos con sistema operativo previamente instalado, que en el caso particular de este proyecto y acorde a los proyectos que se manejan actualmente en el CECAD, es un sistema operativo Linux en alguna de las dos distribuciones CentOS 7 o Ubuntu Xenial 16.04.

Todo nodo que sea monitoreado utilizando SMMD, contiene dos colectores de métricas Collectd y Metric beat respectivamente los cuales envían información al servidor central, el cual está bajo el control de investigador y en donde se encuentran los componentes de recolección de logs, persistencia de datos y visualización. A partir de configuraciones específicas para el recolector de métricas Collectd, el investigador está en la capacidad de incluir diferentes plugins que le permitirán visualizar diferentes tipos de métricas de acuerdo a su necesidad particular. A nivel del servidor que como bien se mencionó anteriormente está bajo el control del investigador, los componentes de recolección de logs, persistencia de datos y presentación, están aislados del sistema de archivos del sistema operativo y corren en un espacio de usuario alterno gracias al uso de contenedores Docker que están orquestados utilizando la herramienta docker-compose<sup>1</sup>, los cuales no tienen dependencia directa las librerías o software del sistema operativo permitiendo así la operabilidad en distintos sistemas operativos que soporten la tecnología Docker, tales como MacOS, Windows y Linux en distintas arquitecturas de procesador como los son x86, amd64, ARM, ppc64le (IBM Power) y s390x (IBM Z). Adicionalmente como consecuencia de la independencia de estos componentes con librerías propias del sistema operativo, permite que la instalación sea mucho más sencilla para el investigador sin la necesidad que este tenga que preocuparse por instalar dependencias en su sistema, con la simple instalación de Docker el investigador tendrá a su disposición el componente del servidor sin inconveniente alguno.

A continuación se presentará en detalle cada uno de los componentes de las capas de SMMD en donde se mencionará en detalle la interoperabilidad entre dichas tecnologías que permite el funcionamiento de SSMD.

---

<sup>1</sup><https://docs.docker.com/compose>, Compose es una herramienta para definir y correr aplicaciones de múltiples contenedores en Docker



**Figura 4-1:** Modelo General del Sistema SMMD.



## 4.1. Recolección de Métricas

### 4.1.1. Tipos de Métricas

En el espacio del tipo de métricas que SMMD soporta actualmente se suman los dos grandes conjuntos de métricas que provee el sistema a través de las herramientas Metricbeat y Collectd, cada uno es una tecnología separada e independiente con propósitos e implementación complementamente distintos los cuales se describen a continuación.

Las métricas soportadas son

### 4.1.2. Metricbeat

Metricbeat<sup>1</sup> conocido también como Elastic Metricbeat, hace parte de la suite de herramientas de Elastic<sup>2</sup> es un software que corre como servicio o demonio el cual permite procesar métricas de desempeño básicas como: uso de CPU, carga promedio del sistema, memoria, uso de red y lecto-escritura en disco. Provee la capacidad de habilitar módulos para procesar métricas de otros sistemas como HAProxy, Apache Kafka, Kubernetes, MongoDB, MySQL entre otros. Debido a que su instalación es muy sencilla y su integración con otros productos de la misma compañía es buena, es la opción estándar para la obtención de métricas básicas de desempeño.

### 4.1.3. Collectd

Existen diversos tipos de software para la recolección de métricas de desempeño, para el caso particular de este proyecto se utilizará **Collectd**, un software de tipo demonio<sup>3</sup>,

### 4.1.4. Statsd

Statsd es un demonio de muy fácil uso

## 4.2. Gestión de Métricas

Logstash es un motor de procesamiento y recopilación de datos basado en plugins. Incluye una gran variedad de plugins que permiten configurarlo fácilmente para recopilar, procesar y reenviar datos en muchas arquitecturas diferentes. **Registrar fuente y actualizar**

<sup>1</sup><https://www.elastic.co/products/beats/metricbeat>

<sup>2</sup>Elastic NV, compañía Europea fundada por Shay Banon en 2012 conocida antiguamente como Elasticsearch, es una compañía dedicada a ofrecer servicios de tipo SaaS (Software como servicio) para el análisis y búsqueda de logs principalmente, servicios usados por compañías reconocidas como: eBay, Cisco, Microsoft y la NASA).

<sup>3</sup>Un programa de computadora que corre como un proceso en segundo plano el cual no está bajo el control directo del usuario actual del sistema

El procesamiento se organiza en uno o más pipelines. En cada pipeline, uno o más plugins de entrada reciben o recopilan datos que luego se colocan en una cola interna. De manera predeterminada, esta es pequeña y se almacena en la memoria, pero puede configurarse para ser más amplia y permanecer en el disco para mejorar la confiabilidad y la persistencia.

Los hilos de procesamiento leen datos de la cola y los procesan a través de cualquier plugin de filtro configurado en secuencia. Logstash viene de fábrica con una gran cantidad de plugins destinados a tipos específicos de procesamiento, y así es como se analizan, procesan y enriquecen los datos.

Una vez que se procesan los datos, los hilos de procesamiento envían los datos a los plugins de salida correspondientes, que son los responsables de formatear y enviar los datos hacia adelante, por. ej., a Elasticsearch.

Los plugins de entrada y salida también pueden tener un plugin de códec configurado. Esto permite formatear datos antes de colocarlos en la cola interna o antes de enviarlos a un plugin de salida.

### 4.3. Persistencia

Elasticsearch es un motor de búsqueda basado en Lucene. Lucene es una API gratuita y de código abierto para hallar informaciones y se usa mucho para crear motores de búsqueda.

Elasticsearch, a través de interfaces web HTTP y documentos JSON, permite interactuar de forma sencilla con su núcleo y realizar búsquedas de texto completo muy eficaces.

La misión de Elasticsearch parte de una convicción de Shay Banon, el fundador: “Search is something that any application should have”. Basándose en este credo, Banon trabajó durante años para llevar su compañía a la cima de la escena informática mundial.

Hoy Elasticsearch es un sistema distribuido que escala horizontalmente, basado en nodos que a su vez se dividen en clústeres. La comunicación hacia y desde los clústeres se realiza a través de las REST API que utilizan HTTP. Las aplicaciones de los clientes que utilizan este proyecto se pueden escribir en cualquier lenguaje. La arquitectura subyacente es obviamente invisible para el usuario, que percibe todo como una entidad única, aunque la naturaleza distribuida de la pila hace que los procesos se interconecten entre sí a través de un intercambio continuo de mensajes entre los nodos. Estos últimos tienen tareas muy específicas que se dividen en autonomía, dependiendo de las configuraciones que se le da a los distintos balances de carga. Si los recursos establecidos al principio no son suficientes, Elasticsearch puede “ampliar” su capacidad de procesamiento activando nuevos nodos y creando nuevos clústeres, con total autonomía; de aquí su naturaleza escalable.

## 4.4. Comunicación

## 4.5. Protocolos de operabilidad

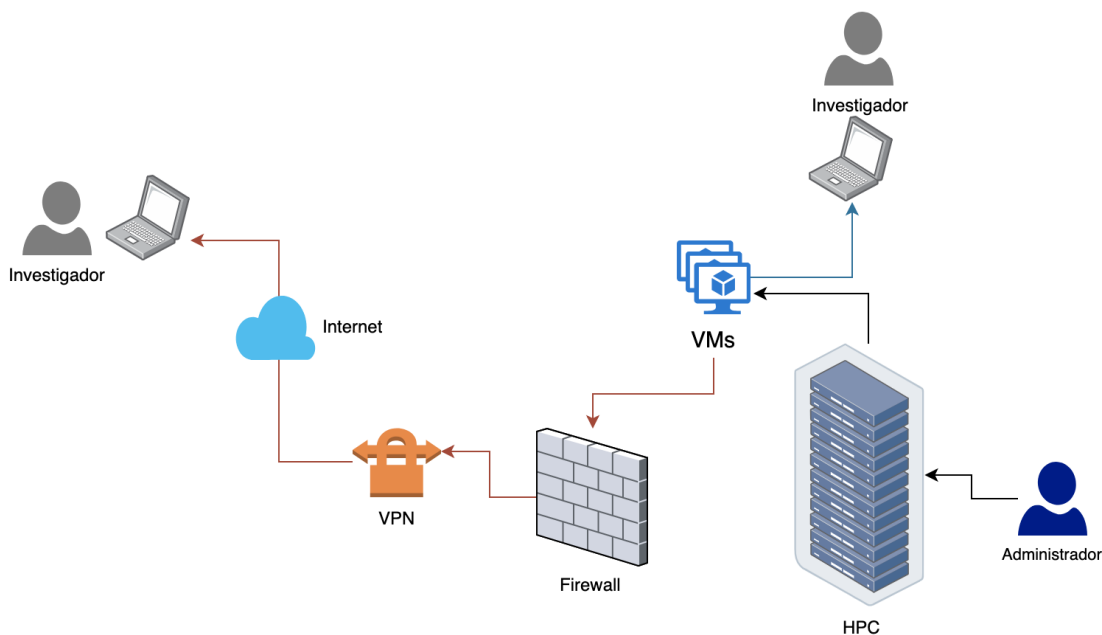
Aquí que protocolo utiliza logstash, collectd y metric beat.

### 4.5.1. Operabilidad en Red

En el aspecto de la comunicación entre cliente y servidor, SMMD permite dos configuraciones básicas como se puede observar en la figura [4-2](#)

1. Usuario externo: En este escenario el investigador se encuentra en un red externa fuera del rango IPs accesible necesario para que las máquinas virtuales asignadas por el administrador envíen información periodica al servidor. La línea roja en el diagrama [4-2](#) representa el flujo de la información desde el interior de la red del centro de computo hasta el servidor principal SMMD controlado por el investigador. Para este escenario, el investigador deberá solicitar al administrador del HPC acceso a la red interna a través de un VPN con el fin de establecer una LAN virtual entre los nodos asignados en el cluster y el equipo servidor que recibirá el reporte de las métricas de desempeño.
2. Usuario interno: En este escenario el usuario se encuentra físicamente en la LAN del centro de computo, y comparte una misma subred lo cual hace que los nodos asignados vean la dirección ip del servidor y puedan enviar la información de métricas de desempeño.

Como se mencionará en el capítulo de instalación, los valores de red a configurar en los nodos son dinámicos y permiten ajustarse a cualquier de los dos escenarios mencionados anteriormente.



**Figura 4-2:** Comunicacion Cliente - Servidor.

## 4.6. Visualización

Kibana es una plataforma de análisis y visualización de código abierto diseñada para trabajar con Elasticsearch. Utiliza Kibana para buscar, ver e interactuar con datos almacenados en los índices de Elasticsearch. Puede realizar análisis de datos avanzados y visualizar fácilmente sus datos en una variedad de gráficos, tablas y mapas.

Kibana facilita la comprensión de grandes volúmenes de datos. Su sencilla interfaz basada en navegador le permite crear y compartir rápidamente paneles dinámicos que muestran cambios en las consultas de Elasticsearch en tiempo real.

La configuración de Kibana es muy fácil. Puedes instalar Kibana y comenzar a explorar sus índices Elasticsearch en minutos - sin código, sin infraestructura adicional requerida.

# 5 Implementación de SMMD

A continuación se describe la estructura de la implementación del Sistema Modular de métricas de desempeño en el CECAD. Se describe una arquitectura de tipo cliente/servidor, donde el servidor es el equipo del investigador, y los clientes son los nodos que le han sido asignados para trabajar dentro del centro de computación de alto desempeño.

## 5.0.1. Requisitos mínimos del sistema

Para hacer uso de SSMD el sistema operativo deberá tener los siguientes requerimientos mínimos para poder ser instalado y utilizado. Estos requisitos se dividen en requisitos del servidor y cliente.

### Requisitos del Servidor

Los siguientes son los requisitos de software para instalar el software servidor de SSMD.

1. Docker Community Edition. Disponible para su descarga en <https://docs.docker.com/install>
2. Python 2.7. Disponible para su descarga en <https://www.python.org/downloads/>
3. Ansible. Ver [https://docs.ansible.com/ansible/latest/installation\\_guide/intro\\_installation.html](https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html)
4. Requisitos opcionales
  - Virtualbox <https://www.virtualbox.org/wiki/Downloadsre>
  - Vagrant <https://www.vagrantup.com/downloads.html>

### Requisitos del Cliente

Los siguientes son los requisitos de software para instalar el software servidor de SSMD.

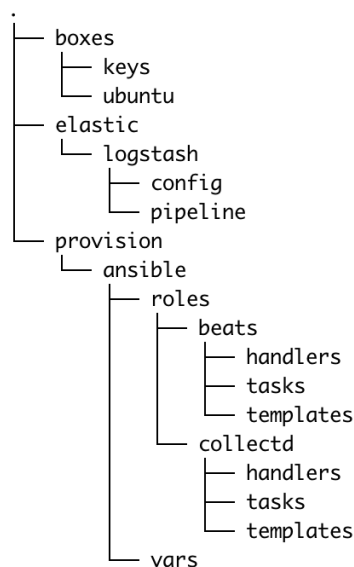
1. Python 2.7. Disponible para su descarga en <https://www.python.org/downloads/>
2. Acceso SSH habilitado para conexiones salientes.
3. Sistema Operativo CentOS 7 o Ubuntu 16.04 LTS (Xenial Xerus)

- CentosOS: <https://www.centos.org/download/>
- Ubuntu: <http://releases.ubuntu.com/16.04/>

## 5.1. Fuentes de Instalación

El código fuente de este proyecto se encuentra en el repositorio público <https://github.com/bizoru/ssmd>, este puede ser descargado desde cualquier lugar y está disponible con uso de licencia GNU GPL v3.<sup>1</sup>

La estructura del código se puede visualizar en la Figura 5-1



**Figura 5-1:** Instalación de SMMD en Nodos Cliente.

A continuación se describe la estructura de archivos que se encuentran en el código fuente de acuerdo con su funcionalidad y propósito.

<sup>1</sup>Licencia Pública General GNU <https://www.gnu.org/licenses/gpl-3.0.en.html>

Las carpetas que se encuentran en el código fuente de acuerdo con la figura 5-1 son:

- boxes

Esta carpeta contiene las imágenes de [Vagrant](#) para hacer la demostración de SMMD sin necesidad de tener nodos externos. Contiene las llaves SSH <sup>2</sup> que se utilizarán para acceder a los nodos y la definición de los mismos.

- elastic

Esta carpeta contiene el archivo principal de [docker-compose](#) el cual contiene las imágenes de Logstash, Kibana, y Elastic Search, aquí se encuentra toda la configuración de puertos, nombre del cluster y contraseña de acceso para el sistema de SMMD a través de Kibana.

- provision

Esta carpeta contiene los scripts<sup>3</sup> de aprovisionamiento, allí se encuentran divididos en unidades lógicas llamados roles, estos definen que componente del cliente se va a instalar, para este caso, el demonio Collectd (Ver sección 4.1.3) y MetricBeat (Ver sección 4.1.2). También encontramos la definición de las variables de cada uno de los programas a instalar, y las configuraciones específicas tanto para Collectd como para metricbeat.

## 5.2. Instalación

La instalación de SMMD consta de dos etapas: La primera etapa consiste en la descarga de la fuente de instalación, (Ver sección 5.1) preparar los requisitos de instalación del servidor de acuerdo al sistema operativo del investigador, posteriormente se deben subir los servicios de Logstash (Ver sección 3.2), Elastic Search (Ver sección 3.1 ) y Kibana (Ver sección 3.3).

### 5.2.1. Instalación del Servidor SMMD

Es importante tener en cuenta los siguientes puntos en consideración para que el proceso de instalación del servidor se pueda realizar a cabo sin problema:

1. Habilitar las reglas de firewall para que pueda tener comunicación con los diferentes nodos.

Para proceder con la iniciación de los servicios de Kibana, ElasticSearch y Logstash realice los siguientes pasos:

---

<sup>2</sup>SSH, Secure Shell, Protocolo para operar computadoras a través de la red de manera segura.

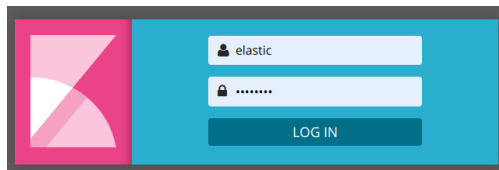
<sup>3</sup>Conjunto de [scripts](#) que permite la automatización de procesos.

1. Ingrese a la carpeta `elastic`
2. Ejecute el comando `docker-compose up`

```
investigador@computadora.local: docker-compose up
Starting elastic_logstash_1 ... done
Starting elastic_kibana_1 ... done
Starting elastic_elasticsearch_1 ... done
Attaching to elastic_logstash_1, elastic_elasticsearch_1, elastic_kibana_1
logstash_1 | 2019/04/28 02:27:06 Setting 'node.name' from environment.
^[[Zelasticsearch_1 | [2019-04-28T02:27:10,671][INFO ][o.e.n.Node ] [ilos] initializing ...
elasticsearch_1 | [2019-04-28T02:27:10,839][INFO ][o.e.e.NodeEnvironment ] [ilos] using [1] data paths, mount
s [[/ (overlay)]], net usable_space [73.4gb], net total_space [466.3gb], spins? [possibly], types [overlay]
elasticsearch_1 | [2019-04-28T02:27:10,839][INFO ][o.e.e.NodeEnvironment ] [ilos] heap size [1.9gb], compress
ed ordinary object pointers [true]
elasticsearch_1 | [2019-04-28T02:27:10,882][INFO ][o.e.n.Node ] [ilos] node name [ilos], node ID [
RYvd4s4JQxeVCq4BrV0Y-Q]
elasticsearch_1 | [2019-04-28T02:27:10,882][INFO ][o.e.n.Node ] [ilos] version[5.6.3], pid[1], bui
ld[1a2f265/2017-10-06T20:33:39.012Z], OS[Linux/4.15.0-47-generic/amd64], JVM[Oracle Corporation/OpenJDK 64-Bit Se
rver VM/1.8.0_141/25.141-b16]
elasticsearch_1 | [2019-04-28T02:27:10,882][INFO ][o.e.n.Node ] [ilos] JVM arguments [-Xms2g, -Xmx
2g, -XX:+UseConcMarkSweepGC, -XX:CMSInitiatingOccupancyFraction=75, -XX:+UseCMSInitiatingOccupancyOnly, -XX:+Alwa
ysPreTouch, -Xss1m, -Djava.awt.headless=true, -Dfile.encoding=UTF-8, -Djna.nosys=true, -Djdk.io.permissionsUseCan
onicalPath=true, -Dio.netty.noUnsafe=true, -Dio.netty.noKeySetOptimization=true, -Dio.netty.recycler.maxCapacityP
erThread=0, -Dlog4j.shutdownHookEnabled=false, -Dlog4j2.disable.jmx=true, -Dlog4j.skipJansi=true, -XX:+HeapDumpOn
OutOfMemoryError, -Des.cgroups.hierarchy.override=/, -Des.path.home=/usr/share/elasticsearch]
elasticsearch_1 | [2019-04-28T02:27:14,222][INFO ][o.e.p.PluginsService ] [ilos] loaded module [aggs-matrix-
stats]
elasticsearch_1 | [2019-04-28T02:27:14,222][INFO ][o.e.p.PluginsService ] [ilos] loaded module [ingest-commo
n]
elasticsearch_1 | [2019-04-28T02:27:14,222][INFO ][o.e.p.PluginsService ] [ilos] loaded module [lang-express
ion]
elasticsearch_1 | [2019-04-28T02:27:14,222][INFO ][o.e.p.PluginsService ] [ilos] loaded module [lang-groovy]
elasticsearch_1 | [2019-04-28T02:27:14,222][INFO ][o.e.p.PluginsService ] [ilos] loaded module [lang-mustach
e]
elasticsearch_1 | [2019-04-28T02:27:14,222][INFO ][o.e.p.PluginsService ] [ilos] loaded module [lang-painles
s]
elasticsearch_1 | [2019-04-28T02:27:14,222][INFO ][o.e.p.PluginsService ] [ilos] loaded module [parent-join]
elasticsearch_1 | [2019-04-28T02:27:14,222][INFO ][o.e.p.PluginsService ] [ilos] loaded module [percolator]
elasticsearch_1 | [2019-04-28T02:27:14,222][INFO ][o.e.p.PluginsService ] [ilos] loaded module [reindex]
elasticsearch_1 | [2019-04-28T02:27:14,222][INFO ][o.e.p.PluginsService ] [ilos] loaded module [transport-ne
ttv31
```

**Figura 5-2:** Ejemplo, resultado de la ejecución de `docker-compose up`, iniciación de servicios

El proceso tomará algún tiempo en completarse una vez esta haya finalizado podrá acceder la interfaz de administración ingresando a <http://localhost:5601>, el usuario es *elastic* y la contraseña es *changeme*.



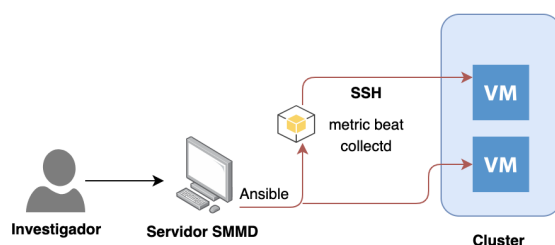
**Figura 5-3:** Pantalla de inicio de sesión en Kibana

Con estos pasos se ha completado la instalación del servidor central donde el investigador podrá acceder al panel de administración y visualizar las métricas de desempeño.



### 5.2.2. Aprovisionamiento de Nodos

SMMD provee una herramienta para hacer la instalación de Metricbeat y collectd, a través de un script de aprovisionamiento dicho script establecerá una conexión SSH hacia los nodos en el cluster y utilizando Ansible hará la instalación del software requerido y configurará los valores necesarios para la comunicación con el servidor central.



**Figura 5-4:** Instalación de SMMD en Nodos Cliente.

Las siguientes son consideraciones importantes ha tener en cuenta para aprovisionar los nodos que enviarán las métricas de desempeño al nodo central.

1. El servidor SMMD deberá estar en la misma red en la que están los nodos, con el fin de poder recibir las métricas de desempeño a través de la red.
2. Cada nodo deberá tener una llave de acceso [SSH](#) en el archivo `./ssh/authorized_keys` con un usuario con permisos de administración.
3. Habilitar el usuario super administrador y llaves SSH
4. Habilitar reglas de firewall para habilitar comunicación externa con el servidor central.
5. Configuración especial en métricas de collectd que se requieran para utilizar en Elastic Search.

Como se puede ver en la figura [5-4](#) el investigador ejecuta el código de aprovisionamiento de [Ansible](#) y este se encarga de hacer la instalación de los paquetes collectd y metricbeat en los nodos o máquinas virtuales que le han sido asignadas en el cluster HPC.

El proceso de aprovisionamiento utiliza el script que se encuentra en la ruta `/provision/provision.sh`, este script se ejecuta de la siguiente manera:

```
export ELASTIC_SEARCH_SERVER=<ip>
export LOGSTASH_SERVER=<ip>
provision.sh <host>
```

Donde las variables para `ELASTIC_SEARCH_SERVER` y `LOGSTASH_SERVER` en el campo de `ip` debe asignarse el valor de la `ip` pública del servidor en el cual se está ejecutando el script de aprovisionamiento.

```
steven mars: ~/Dropbox/UDistrital/Tesis/smmd/provision ./provision-all.sh
Provisionando con Metricbeat, Collectd y StatsD

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [nodol-ubuntu.local]

TASK [collectd : Say hello] *****
ok: [nodol-ubuntu.local] => {
  "msg": "Hello World Ansible! Ubuntu"
}

TASK [collectd : Setup collectd for Ubuntu / Debian] *****
ok: [nodol-ubuntu.local] => {
  "msg": "This is Ubuntu! latest"
}

TASK [collectd : Install collectd] *****
changed: [nodol-ubuntu.local]
```

**Figura 5-5:** Aprovisionamiento de SMMD en los nodos cliente.

Como se puede ver en la figura 5-5 este es el resultado que entrega la consola en el momento de ejecutar el aprovisionamiento, al final del proceso, todos los nodos que hayan sido provisionados, tendrán activos los servicios de `metricbeat`, `collectd` y `statsd`.

5.3.    **Uso**

A continuación se describe el uso de SMMD

**5.3.1.    Visualización de Métricas de Desempeño**

asdf Seleccionar las métricas de desempeño de cada nodo según el nombre de host asignado.

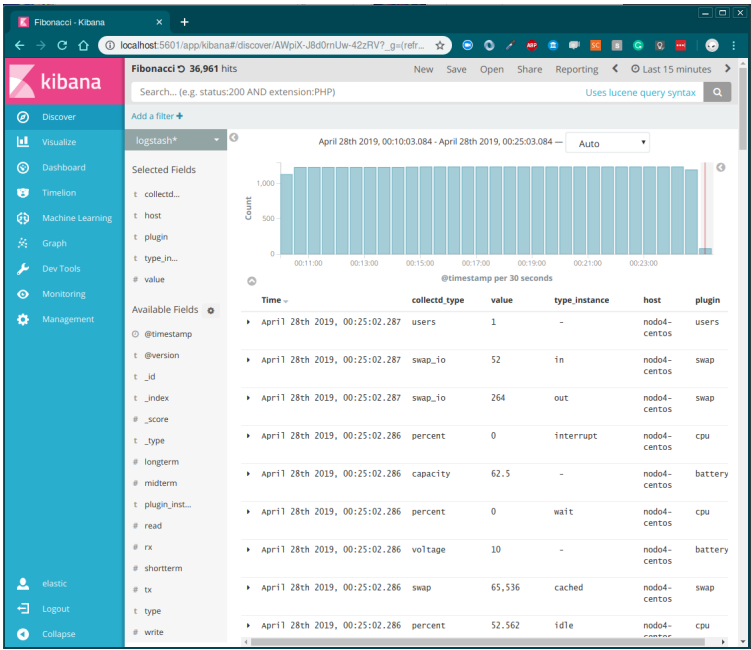


Figura 5-6: Pantalla Principal de SMMD

### **5.3.2. Ajustes generales**

Ajustes generales a nivel del motor de base de datos no relacional Elastic Search para el almacenamiento de la información de las métricas de desempeño.

### **5.3.3. Plugins**

Ajustes generales a nivel de plugin del lado del cliente. Inclusión de nuevas métricas y valores específicos en la configuración del demonio collectd.

## 6 Evaluación de SMMD

Una vez instalada la aplicación de SMMD en el CECAD se mostrará a continuación sus funcionalidades principales y su manejo.

### 6.0.1. Caso práctico CECAD

En la figura se puede observar la creación de un dashboard personalizado el cual muestra los valores de las métricas `fibonacci-count`, `fibonacci-average`, `fibonacci-sum`, valores que son enviados a través de la librería cliente de statsd en python.

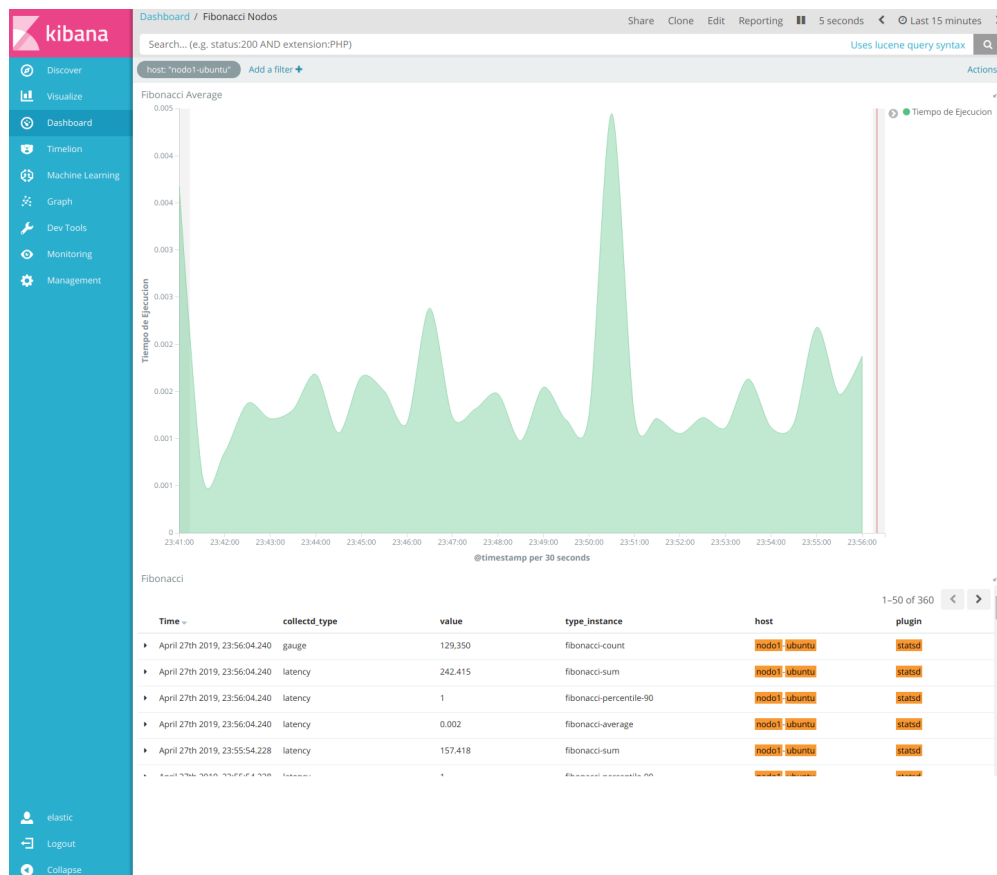


Figura 6-1: Dashboard Personalizado para la medición de métricas recibidas por StatsD

## **7 Conclusiones**

## **8 Principales aportes y divulgación**

# 9 Recomendaciones y Futuros Trabajos

## 9.1. tal

asdfasfdasf adsf asdf adsf ads fdas fdas a sdf



# Glosario

**Ansible** Que es un script. [21](#)

**docker-compose** A strange animal. [19](#)

**script** Que es un script. [19](#)

**SSH** Que es un script. [20](#)

**Vagrant** Un sistema open source para la construcción y mantenimiento de ambientes desarrollo portables a través del uso de diferentes Hypervisors como Virtualbox <http://www.virtualbox.com>. [19](#)

## 10 Referencias

Cáliz, R. (2019). *¿qué es el cecad?* Descargado de <http://cecad.udistrital.edu.co/index.php/home/que-es-el-cecad>

Nielsen, F. (2016). Introduction to hpc with mpi for data science. *Undergraduate Topics in Computer Science*. doi: 10.1007/978-3-319-21903-5

*Top 500 the list*. (2019, Apr). Descargado de <https://www.top500.org/>

*What is high performance computing?* (2019). Descargado de <https://insidehpc.com/hpc-basic-training/what-is-hpc/>