Institut Puig Castellar
Santa Coloma de Gramenet

ASIX1617L Comparació de sistemes de monitorització de xarxes

CFGS Administració de Sistemes Informàtics i Xarxes

Adrià Lora Larroya

2 ASIX
(Administració De Sistemes Informàtics i Xarxes)

01/06/2017

B) GNU Free Documentation License (GNU FDL)

RESUM

El projecte constarà de fer una comparativa entre varis dels sistemes de monitorització de xarxes que hi han ara al mercat. Farem una instal·lació prèvia d'ells i també una serie de proves per veure la estabilitat del programa i quins errors dona.

L'objectiu del projecte serà que un cop instal·lats tots els sistemes de monitorització, poguem decidir quin sistema es millor per a la monitorització a empreses, quins podem utilitzar per a educació, quins son els mes punters, els mes fàcils de configurar etc.

Farem una planificació prèvia del projecte mitjançant un diagrama de Gantt. Seguidament atenent-nos a les pautes del diagrama farem una selecció de diferents sistemes de monitorització tant graficadors unicament com sistemes tot en un. Un cop feta la selecció seguirem una serie de pasos a seguir:

- Instal·lació i configuració
- Proves bàsiques de configuració
- Anàlisi de l'aplicació
- Valoració especifica i conjunta

Finalment es farà un gràfica on es veuran les puntuacions del 1 al 5 en diferents característiques, proves i modificacions.

Índex

1. Introducció

1.1 Context i justificació del Treball

El punt de partida del treball serà: Instal·lar i comparar diferents programes de monitorització de xarxes i valorar i decidir quins son els millors per a cada lloc.

Es un tema rellevant perque així treus molta feina d'investigació per a gent que vulgui monitoritzar les seves màquines.

El resultat que es vol obtenir es una valoració de totes les aplicacións de monitoreig i fer un anàlisi final de totes les instal·lacions.

1.2 Objectius del Treball

Els objectius a assolir en aquest projecte són:

1. Fer un mètode de selecció agafant els millors programes de monitoreig trobats per la xarxa.

2. Realitzar la presa de requeriments, anàlisi i instal·lació de les aplicacions.

3. Realitzar l'instal·lació de les diferents aplicacions segons el grau de dificultat

4. Valorar totes les passes que hem seguit per a la instal·lació del programari de cara a la valoració final.

5. Aprendre a gestionar el temps, ja que cada aplicació requerirà un temps d'instal·lació diferent i un grau de dificultat major o menor.

6. Documentar tot el procés d'instal·lació i configuració del programari emprat i posteriorment fer una serie de proves per veure el nivell de rendiment.

1.3 Enfocament i mètode seguit

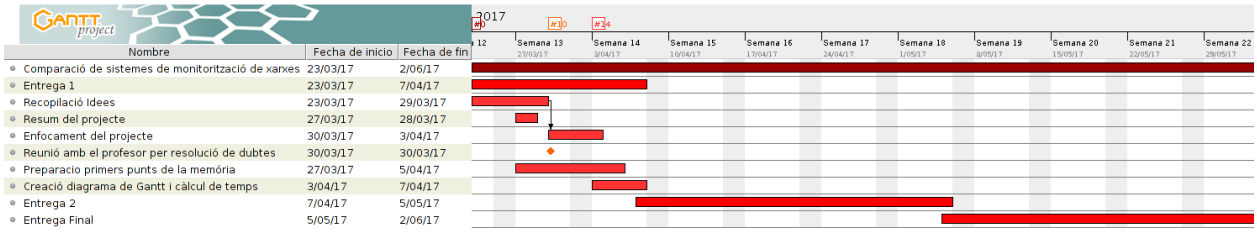
He optat per fer un projecte diferent, fent una tasca d'investigació bastant engorrosa ja que hem d'anar buscant programa a programa i fer una valoració de quin programa de monitorització es l'adequat per a les grans empreses, quin es el millor per l'aprenentatge etc.

Per poder assolir els objectius mes ràpidament i tenir una estratègia solida, anteriorment hem fet un estudi de les possibles idees a afegir al projecte.

També hem fet una presa de requeriments

1.4 Planificació del projecte

Per la planificació del projecte hem fet prèviament una recopilació de possibles idees i tasques a fer i les hem organitzat per fer el diagrama de Gantt.



Aquest es el diagrama de Gantt de la primera entrega. El diagrama anirà augmentant conforme el projecte vagi avançant ja que només tenim constància de les tasques fetes fins ara i del temps que em pot portar fer-ho. Al fer el projecte sol, no tindrem un diagrama de recursos ja que jo faig totes les tasques.

A continuació podem veure a la llista de tasques amb la seva descripció, la durada i també qui la desenvolupa. Això va bé per a grups ja que així podem veure quines tasques desenvolupa cadascú.

1.5 Breu sumari de productes obtinguts

A continuació farem un breu resum dels programes que en principi farem servir per al projecte.

Collectd → És un dimoni que bàsicament el que fa és capturar certes estadístiques del sistema i bolca aquestes dades en un gràfic. Una de les seves principals característiques és que a diferència d'altres sistemes similars collectd no val de l'ús de crontab per col·lectar aquestes dades, sinó que compta amb el seu propi dimoni per a aquest propòsit.

Graphite → Graphite és una eina de monitorització preparada per córrer en maquinari commodity que s'encarrega de bàsicament 2 coses:

- Emmagatzemar dades temporals numèrics
- Renderitzar gràfics d'aquestes dades sota demanda

Graphite es distribueix sota llicència Apache 2.0 i s'utilitza en grans empreses. Graphite es compon de 3 components de programari:

- Carbon → Dimoni twisted que s'encarrega d'escoltar de dades temporals
- Whisper → llibreria per a emmagatzematge de dades temporals
- Graphite webapp-→ webapp Django que renderitza gràfics sota demanda usant Caire.

Si voleu aprofundir podeu accedir a la seva documentació aquí.
Serà la base de dades que en aquest cas agafarà les dades de collectd per a funcionar.

Grafana → Grafana és programari Open Source d'anàlisi de mètriques i una suite de visualització. S'utilitza més comunament per a la visualització de les dades de periodes de temps per a la infraestructura i aplicacions analítiques, però molts ho fan servir en altres dominis que inclouen sensors industrials, domòtica, el clima i el control de processos.

També és el Dashboard que s'encarrega de mostrar tota la informació que Graphite té emmagatzemat en les bases de bades.

Munin → És una eina multiplataforma basada en web, utilitzada en el monitoratge dels recursos en xarxa. Està dissenyada per a ser plug and play.

La seva arquitectura és bastant senzilla: un servidor que centralitza les dades enviades pels agents instal·lats a cada client. Permet monitoritzar molts paràmetres i visualitzar-los en còmodes gràfiques.

Tota la informació generada es pot veure a través del web des de qualsevol part. Està implementat en Perl i alliberat sota llicència GPL versió 2 de la Free Software Foundation

Cacti → És una solució completa que dissenya gràfics de la xarxa, dissenyat per aprofitar el poder d'emmagatzematge de dades de RRDtool i amb una bona funcionalitat gràfica. Cacti proporciona una poller ràpida, plantilles gràfiques avançades, múltiples mètodes d'adquisició de dades i gestió d'usuaris presenta fora de la caixa. Tot això està embolicat en una interfície intuïtiva i fàcil d'usar interfície que tingui sentit per a LAN de mida instal·lacions de fins a xarxes complexes amb milers de dispositius.

Nagios → És un sistema de monitorització de gran abast que permet a les organitzacions identificar i resoldre els problemes d'infraestructura de TI abans que afectin els processos crítics de negoci.

També hi ha forks i altres productes comercials. Mentre que algunes persones els agrada Nagios, hi ha altres que no. Una cosa que es pot dir, és que ja que moltes persones fan servir el producte, ja que hi ha bastant suport online si tens qualsevol dubte.

Icinga2 →Icinga 2 és una xarxa de codi obert sistema que verifica la disponibilitat dels recursos de la xarxa de monitorització, notifica als usuaris de les interrupcions, i genera dades de rendiment per a la presentació d'informes. Icinga 2 és escalable i extensible i pot controlar grans, entorns complexos a través de múltiples ubicacions.

ELK STACK →És una col·lecció de tres productes: Elasticsearch, Logstash i Kibana fets per elastic. Elasticsearch és una base de dades NoSQL que es basa en el motor de cerca Lucene. Logstash és una eina de recol·lecció de dades de registre que accepta entrades de diverses fonts, executa diferents transformacions, i exporta les dades a diversos objectius. Kibana és una capa de visualització que funciona per sobre de Elasticsearch.

2. Monitorització

2.1 Que és?

El monitoratge de servidors consisteix en la vigilància de tots els serveis actius que una màquina ofereix per Internet. Els serveis poden ser: web, correu electrònic, missatgeria instantània, etc.

El monitoratge pot ser tant intern com extern. En el cas del monitoratge intern, la vigilància es realitza des de la mateixa xarxa on està instal·lat el servidor. Quan la monitorització és externa, s'utilitza una plataforma d'un proveïdor de serveis que es troba fora de la nostra xarxa (normalment és una xarxa d'equips distribuïda per tot el món).

El monitoratge extern és molt més fiable, ja que és independent dels problemes que pot haver dins de la xarxa on es troba l'equip a vigilar.

Una altra raó que dóna suport la seva major fiabilitat és que els sistemes comercials compten amb sistemes compostos per servidors distribuïts per tot el món i, per tant, menys sensibles a problemes puntuals en alguna de les xarxes on es troben instal·lats.

Moltes de les eines que instal·larem tenen un protocol en comú, el protocol SNMP.

2.2 Diferents maneres d'enfocament

No tots els programes són iguals: alguns tan sols s'encarreguen de recopilar dades (que poden ser des del rendiment d'un ordinador -el que t'interessarà més- fins a dades de sensors, etc).

Altres s'encarreguen de rebre per la xarxa aquestes dades recopilades i / o guardar-los de forma permanent en alguna base de dades específica (les anomenades Time-Based BD, de les quals un format molt estès és el RRDTool) i altres s'encarreguen d'oferir aquests dades gràficament en forma de "dashboard" editable amb gràfiques i coloraines.

Existeixen diferents tipus de maneres de recopilar les dades:

- **Recol·lectors** → El que fan els recol·lectors es monitoritzar (sempre en la màquina que estiguin instal·lats)el rendiment dels seus 4 recursos hardware clàssics (cpu,ram,disc,xarxa) + l'estat de varis softwares servidors que també estiguin instal·lats a la màquina (start/stop/logs apache,mysql,etc) i a mes a mes poden enviar alertes (o no, segons el programa) a un determinat destí (syslog, mail....) al superar registres definits.

  Per exemple collectd o collectl (una variant molt semblant) el que fan es cada 10 segons, recullen la informació del sistema i la desen en un arxiu RDD (existeix el RDDTool per a gestionar aquests tipus d'arxius).

- **Graficadors** → Son programes que el que fan es fer gràfiques a partir de unes dades prèviament donades. Poden agafar les dades des de mòduls de programació prèviament definits de python, rubi etc.

- **Tot en un** → Els programes tot en un com cacti o munin, fan varies funcions a la vegada. Monitoreigen tant els serveis locals com els serveis que estiguin en xarxa. També consta d'una interfície de xarxa on podem veure i gestionar diferents dades .

  Utilitza una arquitectura mestre/node, en el qual el mestre és l'encarregat d'emmagatzemar la informació que llegeix periòdicament dels nodes. La instal·lació i configuració són relativament fàcils si aquesta es fa a debian (encara que ho farem a ubuntu).

2.2 Protocol SNMP

El protocol SNMP(Simple Network Management Protocol) és un protocol de la capa d'aplicació que facilita l'intercanvi d'informació d'administració entre dispositius de xarxa. Forma part del conjunt de protocols TCP/IP. El protocol SNMP permet als administradors supervisar el funcionament de la xarxa, cercar i resoldre els seus problemes, i planificar el seu creixement.

Les versions de SNMP més utilitzades són dues: SNMP versió 1 (SNMPv1) i SNMP versió 2 (SNMPv2). Les dues versions tenen característiques en comú, però SNMPv2 ofereix millores com, per exemple, operacions addicionals. Nosaltres utilitzarem la versió 2 per a la nostra feina d'investigació.

Existeix una nova versió (SNMPv3) que ofereix canvis significatius en relació als seus predecessors, sobretot en aspectes de seguretat. Tot i això, no ha estat gaire acceptat per la indústria. Ja que no es molt estable i es mes insegura.

2.2.1 SNMP v2

SNMP versió 2 és una evolució de la versió inicial, SNMPv1. Originalment, SNMPv2 es va publicar com un conjunt de normes d'Internet proposades en 1993. Actualment, es tracta d'un projecte de normes. Igual que amb les funcions SNMPv1, SNMPv2 dins de les especificacions de l'Estructura de Gestió d'Informació (SMI).

En teoria, SNMPv2 ofereix una sèrie de millores a SNMPv1, incloent operacions de protocol addicionals.

En cuant a l'estructura d'Informació de Gestió (SMI) defineix les regles per a la descripció de la informació de gestió, utilitzant ASN.1.

El SNMPv2 SMI fa algunes addicions i millores als tipus de dades de SNMPv1 SMI específiques, com ara la inclusió de cadenes de bits, les adreçes de xarxa, i els comptadors. Les cadenes de bits es defineixen només en SNMPv2 i comprenen zero o més bits amb el nom que especifiqui un valor. Les adreces de xarxa representen una adreça d'una família de protocols particulars.

SNMPv1 suporta adreces IP única de 32 bits, però en canvi SNMPv2 pot suportar altres tipus d'adreces també. Els comptadors són enters no negatius que augmenten fins que arriben a un valor màxim i després torna a zero. En SNMPv1, s'especifica una mida comptador de 32 bits no com a SNMPv2 que es defineixen comptadors de 32 bits i 64 bits.

La seguretat del protocol SNMP no té capacitats d'autenticació, el que resulta en la vulnerabilitat, una varietat d'amenaces a la seguretat. Aquests inclouen ocurrències d'emmascarament, modificació de la informació, de seqüència del missatge i modificacions de sincronització, i de la divulgació.

El masquerading per exemple consta d'una entitat no autoritzada d'intentar realitzar operacions de gestió assumint la identitat d'una entitat de gestió autoritzada. A causa de que el protocol SNMP no implementa l'autenticació, molts proveïdors no implementen operacions Set de SNMP que el que fa es que es redueixi a un centre de monitorització.

A continuació farem una comparació de la gran diferència entre la versió 1 i la versió 2 del protocol SNMP, el format del missatge i les capçaleres es mantenen pràcticament igual però el que canvia es la part del Getbulk PDU que només ho te el SNMP v2.

Els missatges SNMP contenen dues parts: una capçalera de missatge i una unitat de protocol de dades (PDU).

Capçalera del missatge	Contingut PDU

Les capçaleres dels missatges a SNMP contenen dos camps:

- Nombre de versió → Especifica la versió de SNMP utilitzat. (v1,v2,v3)

- Nom de la comunitat →Defineix un entorn d'accés per a un grup de SMN. Els noms de comunitat serveixen com una forma feble d'autenticació, ja que els dispositius que no coneixen el nom de la comunitat adequada estan exclosos de les operacions de SNMP.

Ara passarem a explicar el contingut PDU dels missatges SNMP. L'estructura es la mateixa , però lògicament amb algunes millores respecte a la versió 1 ja sigui per major velocitat de transmissió o major qualitat del contingut del missatge. L'estructura seria una mica semblant a això:

PDU Type	Agent Address	Generic trap type	Specific trap code	Time stamp	Object 1 value 1	Object 2 value 2

Variables asociativas

3. Instal·lació de programari

3.1 Icinga 2

ICINGA2 és un sistema de monitorització de codi obert que comprova la disponibilitat dels recursos de xarxa, serveis, informa als usuaris de les interrupcions i genera dades de rendiment per a la presentació d'informes. És una forma avançada de Nagios que també veurem al projecte i té una millor interfície web en comparació amb ella.

Es desenvolupa amb una interfície web fàcil d'utilitzar tant amb més opcions i és més sensible i personalitzable. Per sobre de tot, la comunicació entre el servidor de supervisió i els nodes client s'ha tornat més segur en aquesta versió. Es una de les grans avantatges respecte a la versió 1 d'icinga.

Nosaltres farem la instal·lació de la última versió per veure les millores respecte a la primera versió i també veure amb quina altre suite la podem integrar. Així que agafarem una màquina virtual soyuz proporcionada pel professor, i instal·larem icinga2. El que intentarem serà integrar el que ens reculli icinga2 a un dashborad de grafana.

Primer de tot abans de fer l'instal·lació, instal·larem un paquet que es necessari per després.

```
usuario@soyuz:~$ sudo apt-get install software-properties-common
[sudo] password for usuario:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  python3-software-properties
```

Seguidament afegim el repositori en qüestió i fem un update per que s'apliquin els canvis.

```
usuario@soyuz:~$ sudo add-apt-repository ppa:formorer/icinga
 This PPA provides Icinga 1, Icinga 2 and Icinga web Packages for Ubuntu. They are directly derived from
the Debian Packages that I maintain within Debian.
 Más información: https://launchpad.net/~formorer/+archive/ubuntu/icinga
Pulse [Intro] para continuar o ctrl-c para cancelar

gpg: anillo «/tmp/tmpnhdbd6ba/secring.gpg» creado
gpg: anillo «/tmp/tmpnhdbd6ba/pubring.gpg» creado
gpg: solicitando clave 36862847 de hkp servidor keyserver.ubuntu.com
gpg: /tmp/tmpnhdbd6ba/trustdb.gpg: se ha creado base de datos de confianza
gpg: clave 36862847: clave pública "Launchpad PPA for Alexander Wirt" importada
gpg: Cantidad total procesada: 1
gpg:               importadas: 1  (RSA: 1)
OK
```

Ara si un cop fet l'adició del repositori, podem començar a instal·lar icinga2.

```
usuario@soyuz:~$ sudo apt-get install icinga2
[sudo] password for usuario:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

Un cop instal·lat, el que farem serà activar el servei a mà ja que per defecte està en loaded.

```
usuario@soyuz:~$ systemctl start icinga2.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to start 'icinga2.service'.
Authenticating as: usuario,,, (usuario)
Password:
==== AUTHENTICATION COMPLETE ===
```

Ara si mirem l'estat del servei i veiem que está correctament funcionant.

```
usuario@soyuz:~$ sudo service icinga2 status
● icinga2.service - Icinga host/service/network monitoring system
   Loaded: loaded (/lib/systemd/system/icinga2.service; enabled; vendor preset: enabled)
   Active: active (running) since jue 2017-04-27 18:46:47 CEST; 1 weeks 0 days ago
 Main PID: 13133 (icinga2)
    Tasks: 16
   Memory: 13.4M
      CPU: 56.604s
   CGroup: /system.slice/icinga2.service
```

Ara el que farem serà activar la llista de funcions que tenim per defecte a icinga2, i també en desactiva unes altres que necessitarem mes endavant, que respecte a la versió 1 han estat millorades.

```
usuario@soyuz:~$ sudo icinga2 feature list
[sudo] password for usuario:
Disabled features: api command compatlog debuglog gelf graphite influxdb livestatus opentsdb perfdata sta
tusdata syslog
Enabled features: checker mainlog notification
```

Mes endavant activarem la funció de graphite i si fem mes proves també la d'influxdb. Ara el que farem serà instal·lar un plugin de nagios que el que ens permet es recollir les dades del servei i es bastant integrable amb icinga2 ja que es una adaptació de nagios.

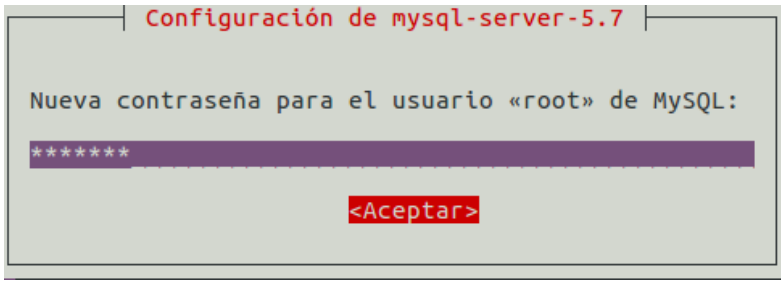
```
usuario@soyuz:~$ sudo apt-get install nagios-plugins
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libarchive13 libavahi-client3 libavahi-common-data libavahi-common3 libcups2 libdbi1 libldb1
  libmysqlclient20 libnet-snmp-perl libpq5 libpython-stdlib libpython2.7 libpython2.7-minimal
  libpython2.7-stdlib libsensors4 libsmbclient libsnmp-base libsnmp30 libtalloc2 libtdb1 libtevent0
  libtirpc1 libwbclient0 monitoring-plugins monitoring-plugins-standard mysql-common python
  python-crypto python-ldb python-minimal python-samba python-talloc python-tdb python2.7
  python2.7-minimal rpcbind samba-common samba-common-bin samba-libs smbclient snmp
```

Com podem veure, s'instal·len moltes dependències i si ens fixem la ultima dependència a instal·lar es la del protocol snmp que tots comparteixen.

Ara necessitarem una base de dades externa per poder emmagatzemar les dades. Jo m'he decantat per MySQL com a base de dades externa. Per tant, hem d'instal·lar el mòdul de MySQL IDO que s'utilitza per a la interfície web Icinga2. S'utilitza per exportar tota la informació de configuració i estat a la base de dades. Necessitem instal·lar MySQL al nostre servidor, si no està instal·lat abans.

```
usuario@soyuz:~$ sudo apt-get install mysql-server-5.7
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl libevent-core-2.0-5 libfcgi-perl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl
  libio-html-perl liblwp-mediatypes-perl libtimedate-perl liburi-perl mysql-client-5.7
  mysql-client-core-5.7 mysql-server-core-5.7
```

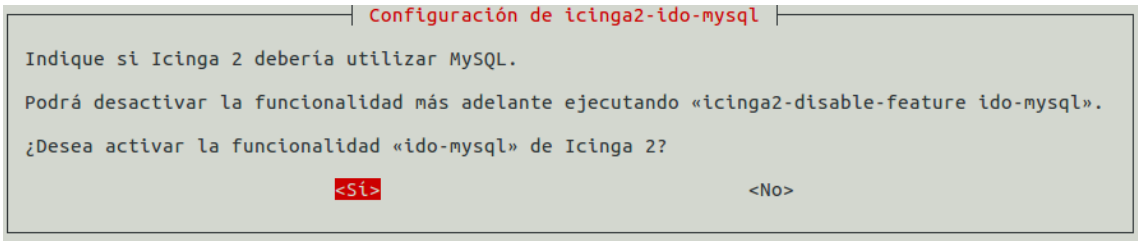
I abans de completar l'instal·lació ens demanarà la contrasenya de root de MYSQL:



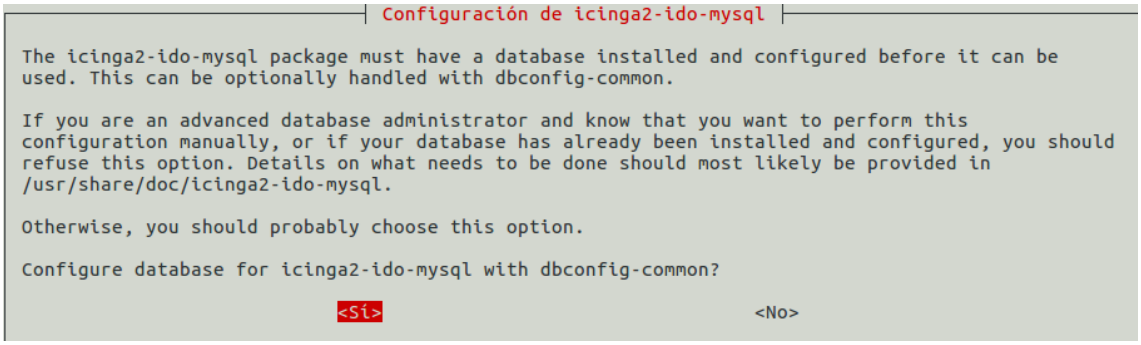
Seguidament instal·larem la dependència IDO que es el mòdul que s'utilitza per a la interfície d'icinga2 com hem dit abans.

```
usuario@soyuz:~$ sudo apt-get install icinga2-ido-mysql
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  dbconfig-common
Paquetes sugeridos:
  dbconfig-mysql | dbconfig-pgsql | dbconfig-sqlite | dbconfig-sqlite3 | dbconfig-no-thanks
  default-mysql-server | mysql-server
Se instalarán los siguientes paquetes NUEVOS:
  dbconfig-common icinga2-ido-mysql
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 112 no actualizados.
```

Quan s'estigui instal·lant, ens demanarà si volem integrar la funció de la base de dades mysql amb icinga2 i li diem que si



El següent pas serà dir-li que ens guardi tot el que recol·lecti a una base de dades creada automàticament a mysql (l'opció està per defecte).



Per últim ens demanarà la contrasenya de l'usuari root un altre cop i ja haurem instal·lat el plugin. Un cop instal·lat el mòdul. Cal configurar la nostra base de dades MySQL per acceptar els valors utilitzant aquest mòdul.

```
usuario@soyuz:~$ sudo mysql -u root -p
[sudo] password for usuario:
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 5.7.18-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Crearem una base de dades amb privilegis per realitzar totes les accions anomenada "Icinga2" amb l'usuari i contrasenya que volguem.

```
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, DROP, CREATE VIEW, INDEX, EXECUTE ON icinga2.*TO 'icinga2'@'
localhost' IDENTIFIED BY 'icinga123';
Query OK, 0 rows affected, 1 warning (0,00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0,00 sec)
```

Ara activarem el servei IDO per que la configuració s'apliqui correctament.

```
usuario@soyuz:~$ sudo icinga2 feature enable ido-mysql
Enabling feature ido-mysql. Make sure to restart Icinga 2 for these changes to take effect.
```

El següent que mirarem serà que el servei que acabem d'activar s'hagi instal·lat correctament, per això fem un cat del fitxer

```
usuario@soyuz:~$ sudo cat /etc/icinga2/features-enabled/ido-mysql.conf
[sudo] password for usuario:
/**
 * The db_ido_mysql library implements IDO functionality
 * for MySQL.
 */

library "db_ido_mysql"

object IdoMysqlConnection "ido-mysql" {
  user = "icinga2",
  password = "usuario",
  host = "localhost",
  database = "icinga2"
}
```

El que farem ara un cop instal·lada la base de dades serà la configuració de l'apartat web d'icinga2. En la versió d'Ubuntu 16, PHP 7.0 és la versió per defecte, hi ha molts més problemes de compactabilitat d'Icinga2 amb PHP 7.0. Per tant, abans d'aquesta instal·lació, hem d'instal·lar PHP versió 5.6. Per a la instal·lació de PHP 5.6 en aquest servidor, necessito habilitar un repositori anomenat "Ondrej / php".

```
usuario@soyuz:~$ sudo add-apt-repository ppa:ondrej/php
gpg: anillo «/tmp/tmpz919eto5/secring.gpg» creado
gpg: anillo «/tmp/tmpz919eto5/pubring.gpg» creado
gpg: solicitando clave E5267A6C de hkp servidor keyserver.ubuntu.com
gpg: /tmp/tmpz919eto5/trustdb.gpg: se ha creado base de datos de confianza
gpg: clave E5267A6C: clave pública "Launchpad PPA for Ondřej Surý" importada
gpg: Cantidad total procesada: 1
gpg:               importadas: 1  (RSA: 1)
OK
```

I seguidament instal·lem la versió 5.6 comentada anteriorment.

```
usuario@soyuz:~$ sudo apt-get install php5.6
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Nota, seleccionando «php5.6-json» para la expresión regular «php5.6»
Nota, seleccionando «php5.6-common» para la expresión regular «php5.6»
```

Un cop instal·lat el plugin, podrem veure que la versió de php ha camviat amb la comanda *php -v* :

```
usuario@soyuz:~$ php -v
PHP 5.6.30-10+deb.sury.org~xenial+2 (cli)
Copyright (c) 1997-2016 The PHP Group
Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies
    with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2016, by Zend Technologies
```

Ara el que farem serà executar el mysql per crear la base de dades però del apartat web:

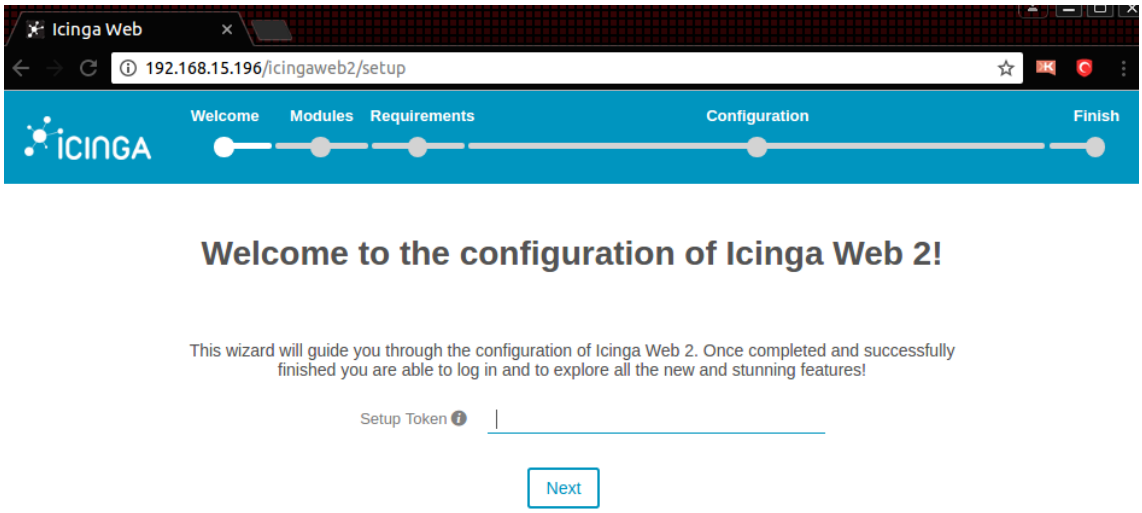
```
mysql> CREATE DATABASE icingawebdb;
Query OK, 1 row affected (0,02 sec)

mysql> GRANT SUPER ON *.* TO 'icingaweb'@'localhost' IDENTIFIED BY 'icinga123';
Query OK, 0 rows affected, 1 warning (0,00 sec)

mysql> GRANT SELECT, INSERT, UPDATE, DELETE, DROP, CREATE VIEW, INDEX, EXECUTE ON icingawebdb.*TO 'icinga
web'@'localhost' IDENTIFIED BY 'icinga123';
Query OK, 0 rows affected, 1 warning (0,00 sec)
```

Estem fent mes o menys las mateixes passes que quan vam instal·lar el servidor d'icinga2. Primerament crearem la base de dades d'icingaweb, després assignarem els permisos de superusuari al usuari icingaweb, i finalment crearem una consulta on definim les accions que aquest usuari pot fer.

Ara el que farem serà instal·lar per fi el paquet web d'icinga2. Per fer-ho executem la següent comanda:

```
usuario@soyuz:~$ sudo apt-get install icingaweb2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  fontconfig-config fonts-dejavu fonts-dejavu-core fonts-dejavu-extra ghostscript gsfonts icingacli
  icingaweb2-common icingaweb2-module-doc icingaweb2-module-monitoring imagemagick-common
  libcupsfilters1 libcupsimage2 libfftw3-double3 libfontconfig1 libgd3 libgomp1 libgs9 libgs9-common
  libijs-0.35 libjbig0 libjbig2dec0 libjpeg-turbo8 libjpeg8 liblcms2-2 liblqr-1-0 libltdl7
  libmagickcore-6.q16-2 libmagickwand-6.q16-2 libpaper-utils libpaper1 libtiff5 libwebp5 libxpm4
  libxslt1.1 php-dompdf php-font-lib php-gd php-htmlpurifier php-icinga php-imagick php-intl php-ldap
  php-mysql php-xml php7.1-common php7.1-gd php7.1-intl php7.1-ldap php7.1-mysql php7.1-xml
  poppler-data sdop ttf-dejavu-core
```

Instal·larem bastants paquets addicionals ja que la interfície web requereix molts paquets. Ara si tot ha anat bé, podrem configurar el plugin des de la interfície web teclejant a la barra de direccions la següent URL "http://IPSERVIDOR//icingaweb2/setup".



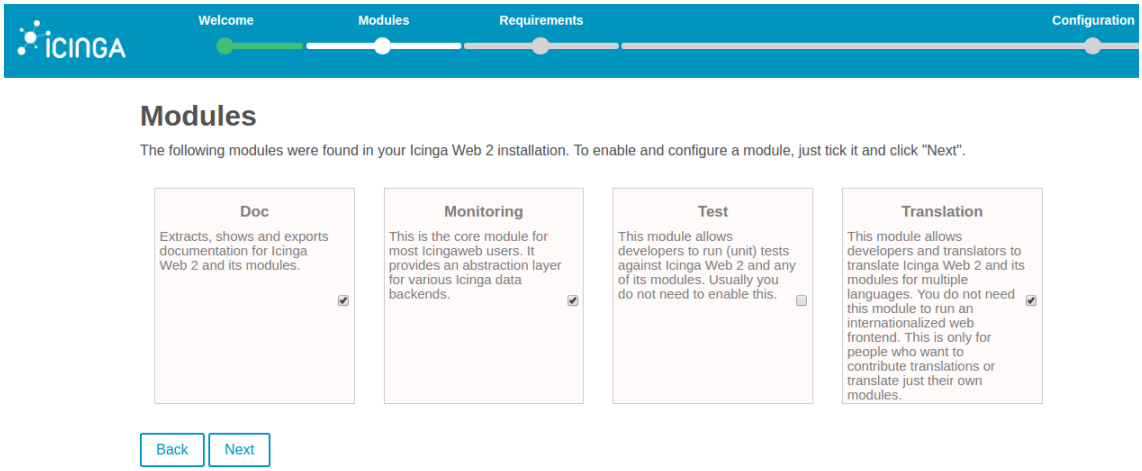
Ara un cop estem a la interfície web, el que farem serà generar el token per sincronitzar l'icingacli amb l'icingaweb. Els passos a seguir son els següents:

```
usuario@soyuz:~$ cd /usr/local/src/
usuario@soyuz:/usr/local/src$ sudo addgroup --system icingaweb2
addgroup: El grupo `icingaweb2' ya existe como grupo del sistema. Saliendo.
usuario@soyuz:/usr/local/src$ sudo usermod -a -G icingaweb2 www-data
usuario@soyuz:/usr/local/src$ sudo icingacli setup config directory --group icingaweb2;
Successfully created configuration directory /etc/icingaweb2
```

Un cop comprovem que el grup existeix i que la configuració de directoris d'icingaweb s'han sincronitzat amb èxit, passarem a generar el token:

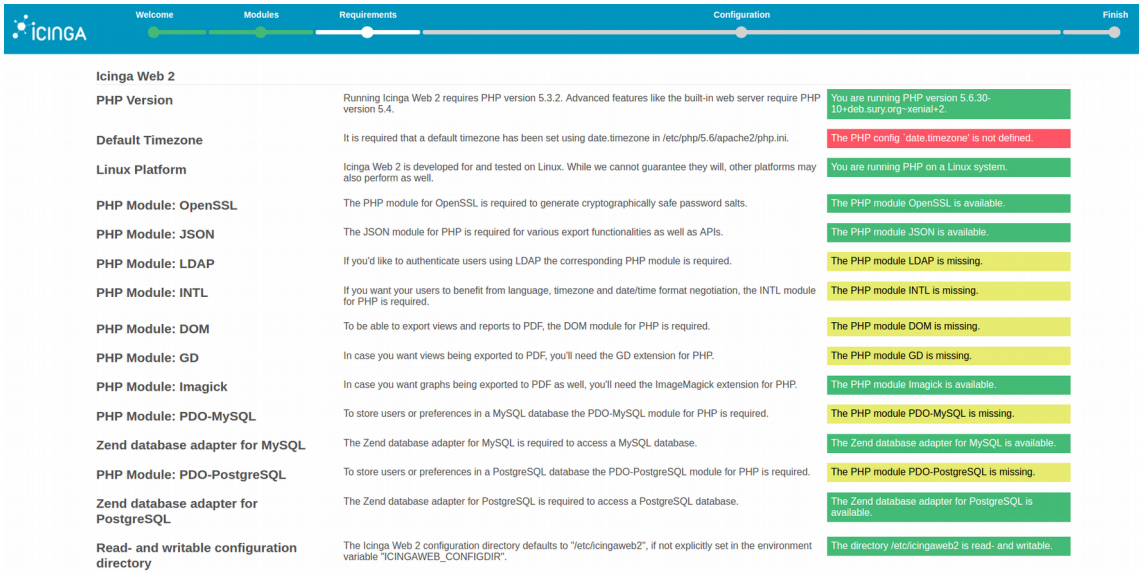
```
usuario@soyuz:/usr/local/src$ sudo icingacli setup token create;
The newly generated setup token is: 24d7dbe43ce3b8e3
```

Amb aquest token l'introduïm a l'interficie web ens trobem amb la següent opció, que es la selecció dels mòduls d'icinga.

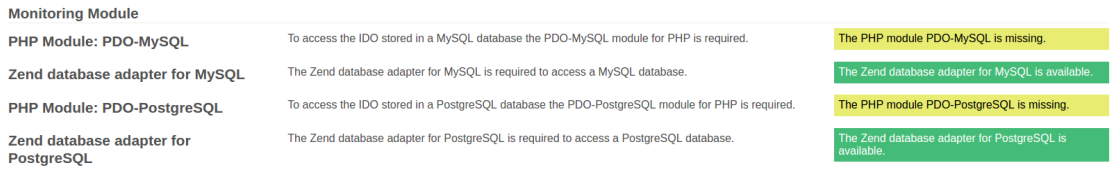


Com podem veure, instal·lem tots els mòduls a excepció del test (no es requereix). La següent tasca serà instal·lar els diferents mòduls de php per poder seguir endavant.

Aquí podem veure quins mòduls ens falten per instal·lar a l'apartat d' icinga web2 :



També podem veure els mòduls que falten de monitorització :



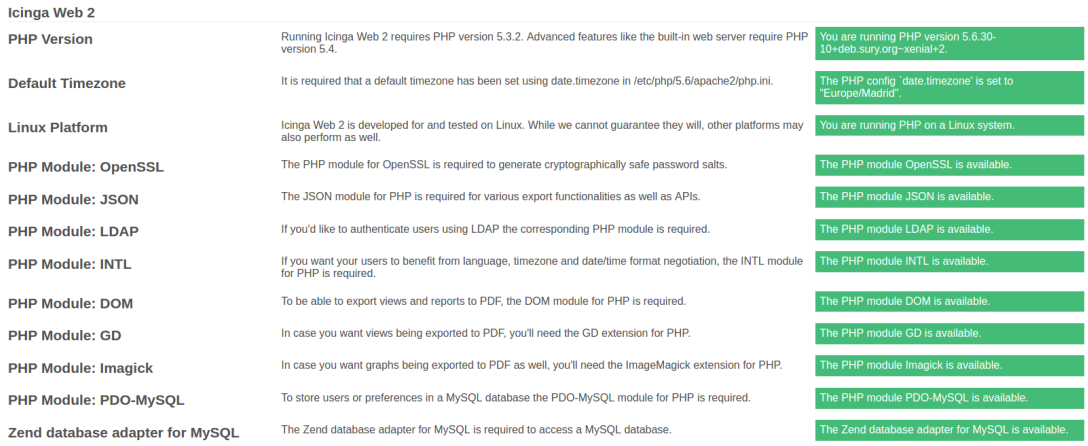
Per poder seguir endavant, cal instal·lar tots els mòduls que falten, en aquest cas els que estan amb color groc.

```
usuario@soyuz:/usr/local/src$ sudo apt-get install php5.6-gd php5.6-json  php5.6-dba php5.6-intl php5.6-l
dap php5.6-pdo-mysql php5.6-imagick php5.6-dom
[sudo] password for usuario:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

Perquè tinguin efecte tots els mòduls instal·lats, el que farem serà actualitzar la zona horària (date.timezone) del fitxer php.ini d'apache.

```
[Date]
; Defines the default timezone used by the date functions
; http://php.net/date.timezone
date.timezone = 'Europe/Madrid'
```

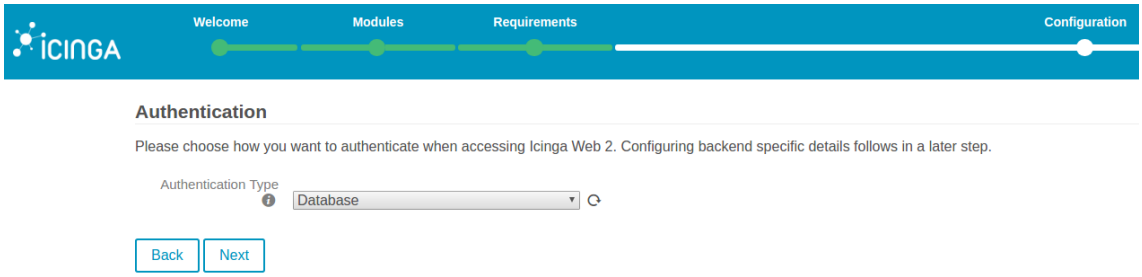
Fem un reinici d'apache i veurem que els mòduls s'han actualitzat.



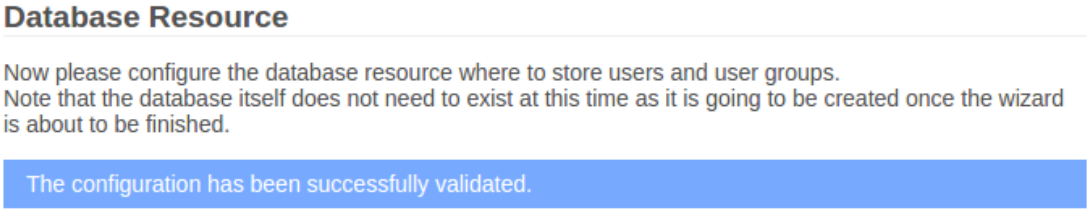
Icinga Web 2

PHP Version	Running Icinga Web 2 requires PHP version 5.3.2. Advanced features like the built-in web server require PHP version 5.4.	You are running PHP version 5.6.30-10+deb.sury.org~xenial+2.
Default Timezone	It is required that a default timezone has been set using date.timezone in /etc/php/5.6/apache2/php.ini.	The PHP config 'date.timezone' is set to "Europe/Madrid".
Linux Platform	Icinga Web 2 is developed for and tested on Linux. While we cannot guarantee they will, other platforms may also perform as well.	You are running PHP on a Linux system.
PHP Module: OpenSSL	The PHP module for OpenSSL is required to generate cryptographically safe password salts.	The PHP module OpenSSL is available.
PHP Module: JSON	The JSON module for PHP is required for various export functionalities as well as APIs.	The PHP module JSON is available.
PHP Module: LDAP	If you'd like to authenticate users using LDAP the corresponding PHP module is required.	The PHP module LDAP is available.
PHP Module: INTL	If you want your users to benefit from language, timezone and date/time format negotiation, the INTL module for PHP is required.	The PHP module INTL is available.
PHP Module: DOM	To be able to export views and reports to PDF, the DOM module for PHP is required.	The PHP module DOM is available.
PHP Module: GD	In case you want views being exported to PDF, you'll need the GD extension for PHP.	The PHP module GD is available.
PHP Module: Imagick	In case you want graphs being exported to PDF as well, you'll need the ImageMagick extension for PHP.	The PHP module Imagick is available.
PHP Module: PDO-MySQL	To store users or preferences in a MySQL database the PDO-MySQL module for PHP is required.	The PHP module PDO-MySQL is available.
Zend database adapter for MySQL	The Zend database adapter for MySQL is required to access a MySQL database.	The Zend database adapter for MySQL is available.

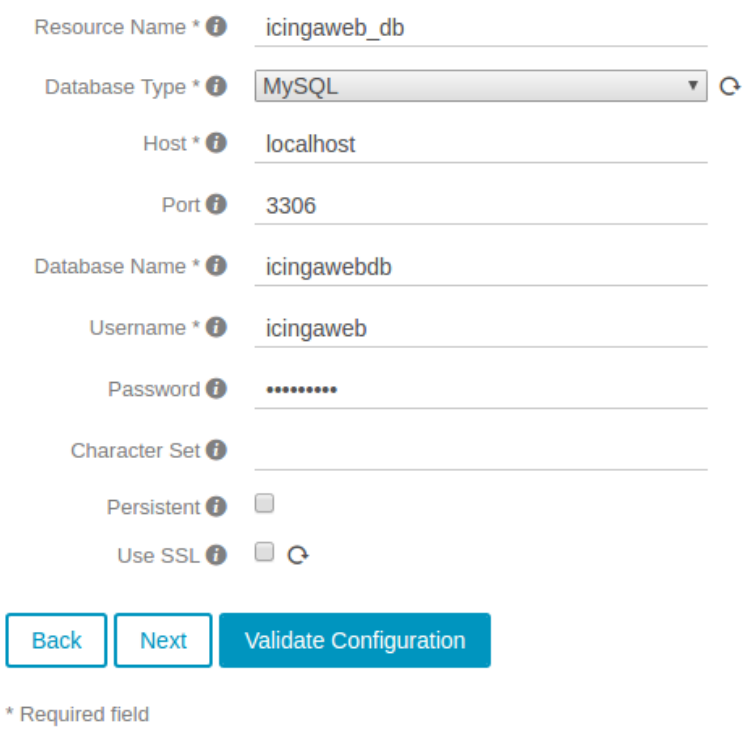
S'han d'anar instal·lant tots els mòduls de php que ens facin falta per a la resolució d'aquest apartat. Sempre també reiniciar apache2 per a que s'apliquin els serveis. Si tot ha anat bé, el que farem serà escollir el tema de la base de dades.



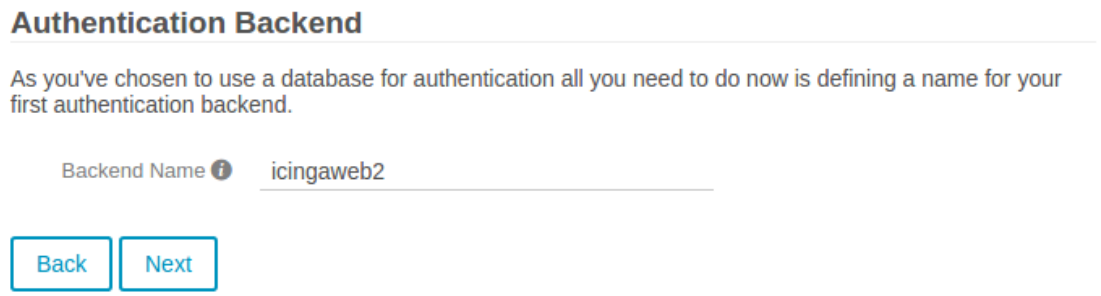
Al escollir el tema Database, hem d'emplenar una serie de camps per autenticar amb la base de dades, tota aquesta configuració la hem feta prèviament a l'instal·lació. Si hem posat bé les credencials, ens sortirà el següent missatge:



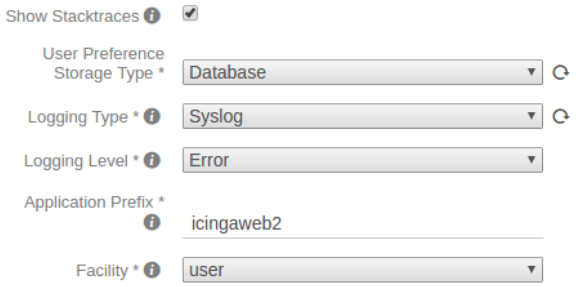
La configuració de les dades es la següent:



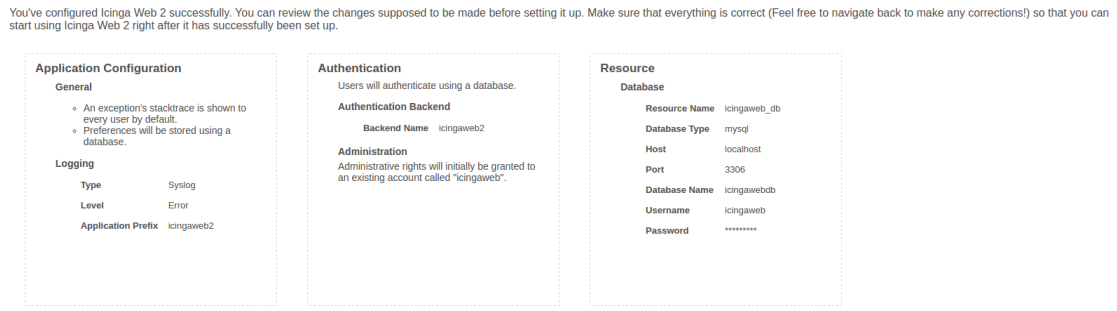
Ara passarem a configurar el backend. Automàticament s'importa el nom de l'interficie web instal·lada per consola anteriorment.



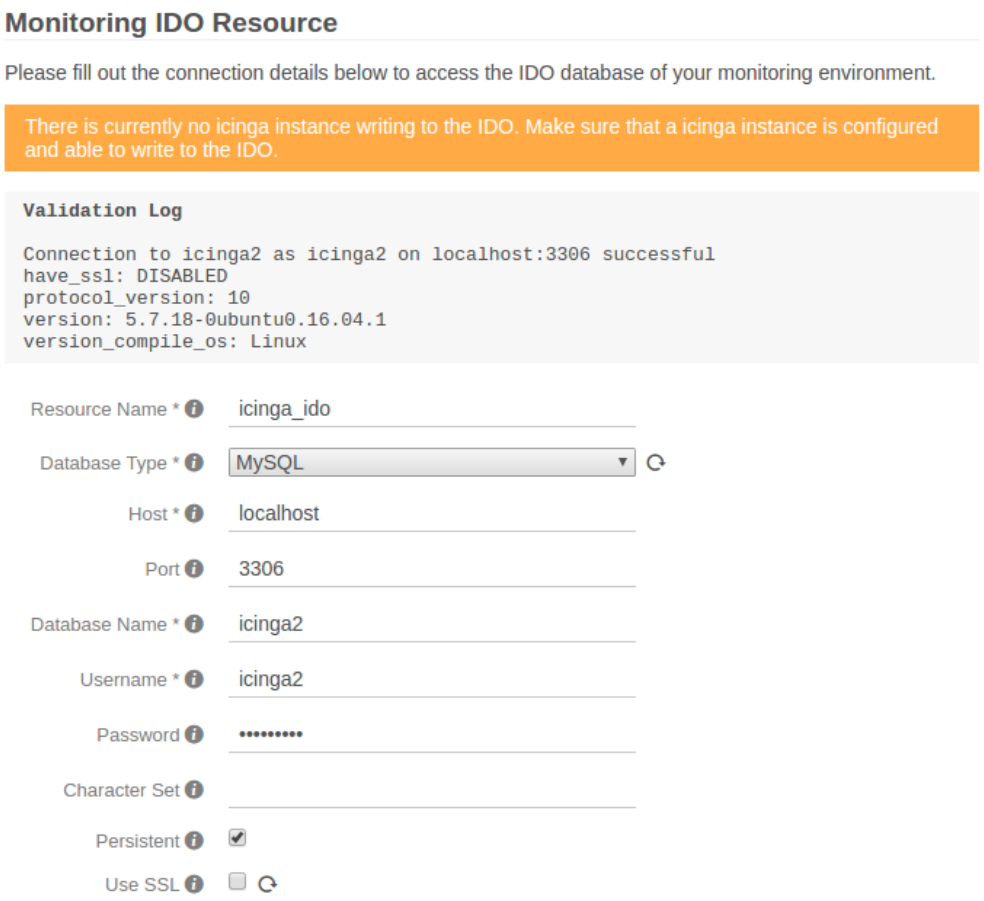
Seguidament, ficarem l'usuari i la contrasenya d'administrador de la part del backend. Si tot va bé haurem de configurar la part d'aplicació amb les dades que venen per defecte (tot això ja està anteriorment configurat).



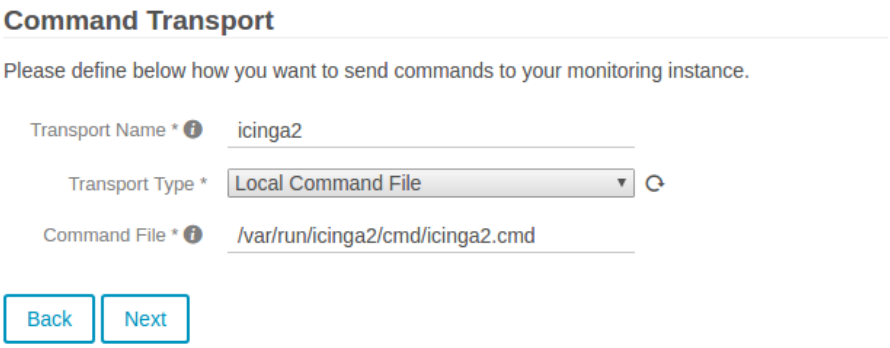
Finalment tenim una pantalla amb la revisió de tota la configuració de icinga, per mirar que tots els camps estiguin correctament configurats.



Seguidament si tot ha anat bé, configurarem la part de la monitorització del backend. La configuració es la següent:



Ara veurem quina es la ruta de seguiment de execució de la comanda de transport :

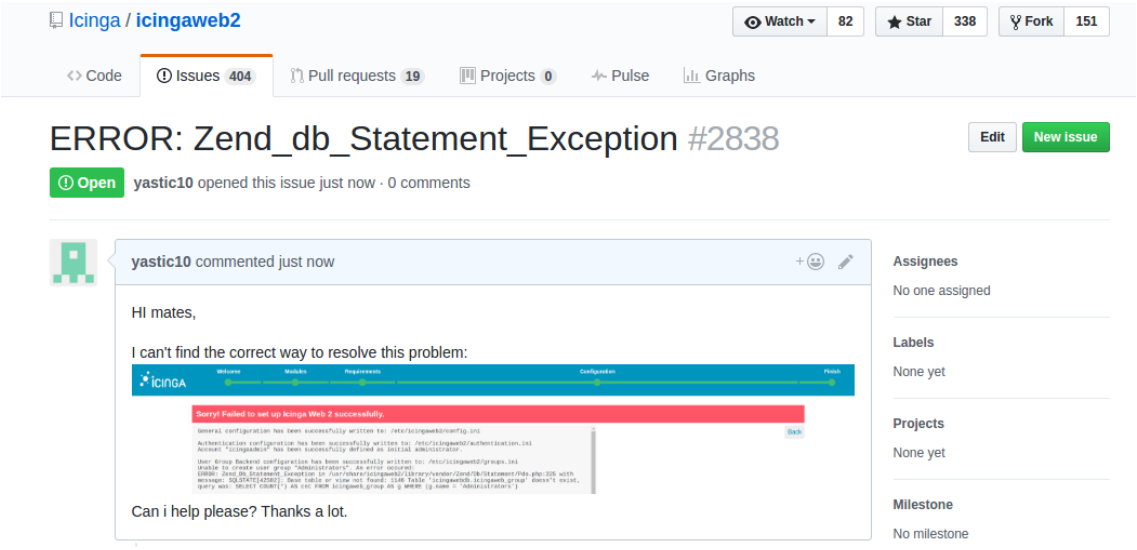


Finalment, veurem el resum de la configuració per verificar tots els detalls :

Si tot ha anat be hauria de sortir la pàgina que ens diu que icinga2 ha estat correctament instal·lat i llest per a funcionar, però en el nostre cas hem tingut el següent error:

El nostre error es que degut a que la instal·lació d'icingaweb2 ha sigut erronea es veu que no s'han creat les taules necesaries a la base de dades i per tant està intentant autentificarse amb un usuari i un password que no està allotjat a cap lloc.

El que hem fet ha sigut escriure al github d'icinga per a veure si ens poden resoldre el nostre dubte ja que hi ha gent que ha tingut el mateix error i no ha pogut fer l'instal·lació amb èxit.



Hem demanat ajuda mitjançant el github d'icingaweb2 que es on tenim el problema, i esperem resposta ja que la comunitat sempre intenta ajudar en tot el possible. Aquí tenim la nova issue creada:

3.2 ELK STACK

Un dels programaris que hem trobat interessants per a instal·lar es la pila ELK és a dir, Elasticsearch, Logstash i Kibana). També intentarem configurar la pila perquè pugin reunir i visualitzar els logs de sistema, dels sistemes en un lloc centralitzat, utilitzant Filebeat.

Ara passarem a fer un breu resum de que es cada eina i quines accions poden dur a terme:

- **Logstash** és una eina de codi obert per a la recollida, anàlisi i emmagatzemar registres per al seu ús futur.
- **Kibana** és una interfície web que pot ser utilitzat per buscar i veure els registres que Logstash ha indexat.

Les dues eines es basen en **Elasticsearch**, que s'utilitza per emmagatzemar els registres.

El registre centralitzat pot ser molt útil quan es tracta d'identificar els problemes amb els servidors o aplicacions, ja que li permet buscar a través de tots els seus registres en un sol lloc. També és útil perquè permet identificar els problemes que abasten múltiples servidors mitjançant la correlació dels seus registres durant un període de temps específic.

És possible utilitzar Logstash per recopilar registres de tota mena, però anem a limitar l'abast d'aquest tutorial per a la recol·lecció de syslog.

Així doncs la nostra configuració de pila ELK té quatre components principals:

- Logstash → El component de servidor de Logstash que processa diaris d'enviament
- Elasticsearch → emmagatzema tots els registres
- Kibana → interfície web per a la recerca i visualització de registres, que serà a través de proxy Nginx
- Filebeat → Instal·lat en servidors del client que enviaran als seus registres a Logstash, Filebeat serveix com un agent de transport de registres que utilitza el llenyataire protocol de xarxa per a comunicar-se amb Logstash

Un cop feta una mica l'introducció de que es aquesta pila, començarem per crear les maquines. Crearem una màquina Ubuntu Server 16.04 amb adaptador pont i ara instal·larem una serie de dependències. Primerament començarem per afegir una serie de repositoris per instal·lar java8.

```
usuario@soyuz:~$ sudo add-apt-repository -y ppa:webupd8team/java
[sudo] password for usuario:
gpg: anillo «/tmp/tmperv_zww4/secring.gpg» creado
gpg: anillo «/tmp/tmperv_zww4/pubring.gpg» creado
gpg: solicitando clave EEA14886 de hkp servidor keyserver.ubuntu.com
gpg: /tmp/tmperv_zww4/trustdb.gpg: se ha creado base de datos de confianza
gpg: clave EEA14886: clave pública "Launchpad VLC" importada
gpg: no se encuentran claves totalmente fiables
gpg: Cantidad total procesada: 1
gpg:               importadas: 1  (RSA: 1)
OK
```

Seguidament instal·larem el paquet java8 instaler. Un cop instal·lat si acceptem els termes i condicions, descomprimirem el fitxer.

```
usuario@soyuz:~$ sudo apt-get -y install oracle-java8-installer
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  binutils gsfonts gsfonts-x11 java-common libfontenc1 libxfont1 oracle-java8-set-default x11-common
  xfonts-encodings xfonts-utils
```

```
     0K ........ ........ ........ ........ ........ ........  1% 3,95M 44s
  3072K ........ ........ ........ ........ ........ ........  3% 2,77M 53s
  6144K ........ ........ ........ ........ ........ ........  5% 3,71M 49s
  9216K ........ ........ ........ ........ ........ ........  6% 3,09M 50s
 12288K ........ ........ ........ ........ ........ ........  8% 4,32M 47s
 15360K ........ ........ ........ ........ ........ ........ 10% 5,09M 43s
 18432K ........ ........ ........ ........ ........ ........ 11% 4,00M 42s
 21504K ........ ........ ........ ........ ........ ........ 13% 4,66M 40s
 24576K ........ ........ ........ ........ ........ ........ 15% 4,06M 39s
```

Ara el que farem serà importar la clave pública GPG d'elastisearch:

```
usuario@soyuz:~$ wget -qO - https://packages.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
OK
```

Un cop importada la clau GPG, el següent que farem sera importar les llistes de fonts de Elasticsearch, son els següents paquets:

```
usuario@soyuz:~$ echo "deb http://packages.elastic.co/elasticsearch/2.x/debian stable main" | sudo tee -a
 /etc/apt/sources.list.d/elasticsearch-2.x.list
deb http://packages.elastic.co/elasticsearch/2.x/debian stable main
```

Ara si podem instal·lar elasticsearch tenint prèviament configurades les llistes i la clau gpg pública.

```
usuario@soyuz:~$ sudo apt-get -y install elasticsearch
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  elasticsearch
```

Ara un cop instal·lat elasticsearch, el que farem serà editar el fitxer .yml perquè poguem tenir connexió des de el nostre host i betar l'entrada de l'exterior per evitar possibles atacs:

```
# ---------------------------------- Network -----------------------------------
#
# Set the bind address to a specific IP (IPv4 or IPv6):
#
 network.host: localhost
#
# Set a custom port for HTTP:
#
# http.port: 9200
```

Un cop modificat l'arxiu amb els paràmetres per que poguem fer bé la connexió, i perquè la gent de fora no pugui fer de les seves, reiniciarem el servei i l'activarem a l'inici de la màquina.

```
usuario@soyuz:~$ sudo systemctl restart elasticsearch.service
usuario@soyuz:~$ sudo systemctl daemon-reload
usuario@soyuz:~$ sudo systemctl enable elasticsearch.service
Synchronizing state of elasticsearch.service with SysV init with /lib/systemd/systemd-sysv-install...
Executing /lib/systemd/systemd-sysv-install enable elasticsearch
Created symlink from /etc/systemd/system/multi-user.target.wants/elasticsearch.service to /usr/lib/system
d/system/elasticsearch.service.
```

Si tot ha anat bé ja tindrem instal·lat elasticsearch, faltarà configurar-lo a fons quan tinguem totes les suites instal·lades. El que farem a continuació serà instal·lar Kibana. El podem instal·lar fent l'adició dels repositoris oficials de la suite:

```
usuario@soyuz:~$ echo "deb http://packages.elastic.co/kibana/4.5/debian stable main" | sudo tee -a /etc/a
pt/sources.list
deb http://packages.elastic.co/kibana/4.5/debian stable main
```

Seguidament ja podem instal·lar kibana. Es una suite que té molta configuració per defecte amb elasticsearch com ja hem comentat anteriorment.

```
usuario@soyuz:~$ sudo apt-get -y install kibana
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  kibana
```

Quan ja tinguem el paquet instal·lat, el que farem serà configurar la connexió a la nostra maquina en el fitxer .yml que tenim a continuació:

```
  GNU nano 2.5.3                Archivo: /opt/kibana/config/kibana.yml

# Kibana is served by a back end server. This controls which port to use.
# server.port: 5601

# The host to bind the server to.
 server.host: "localhost"
```

Com hem fet amb elasticsearch el que farem serà reiniciar el servidor de kibana, i configurar-lo perquè arrenqui a l'inici.

```
usuario@soyuz:~$ sudo systemctl daemon-reload
usuario@soyuz:~$ sudo systemctl enable kibana
Synchronizing state of kibana.service with SysV init with /lib/systemd/systemd-sysv-install...
Executing /lib/systemd/systemd-sysv-install enable kibana
usuario@soyuz:~$ sudo systemctl start kibana
```

A causa de que hem configurat Kibana ens escolti per localhost, hem de configurar un proxy invers per permetre l'accés extern a ell. Utilitzarem Nginx per canviar una mica, ja que sempre fem servir apache.

```
usuario@soyuz:~$ sudo apt-get -y install nginx
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

Per entrar a l'interfície web de kibana, utilitzem la política SSL per això crearem un usuari i una contrasenya amb la següent comanda:

```
usuario@soyuz:~$ echo "kibanaadmin:`openssl passwd -apr1`" | sudo tee -a /etc/nginx/htpasswd.users
Password:
Verifying - Password:
kibanaadmin:$apr1$sdiRuMMA$8kKIP0kSFLDCnCA7GsOWQ/
```

Ara obrirem l'arxiu de configuració d'nginx i configurarem una serie de paràmetres que són els següents:

```
server {
    listen 80;

    kibana_server example.com;

    auth_basic "Restricted Access";
    auth_basic_user_file /etc/nginx/htpasswd.users;

    location / {
        proxy_pass http://localhost:5601;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

Això configurarà Nginx per dirigir el trànsit HTTP del servidor per a l'aplicació Kibana, que està escoltant en localhost pel port 5601. A més, Nginx utilitzarà el htpasswd.users arxiu, que hem creat anteriorment, i requereixen l'autenticació bàsica.

Ara reiniciarem el servidor per comprovar que no tenim cap error de sintaxi, si en tenim algun ho podem mirar amb la comanda journalctl -xe.

```
usuario@soyuz:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

El paràmetre -t el que fa es fer-nos un test del servidor tant de sintaxi com de directives.

Ara si apliquem les regles per defecte del nostre firewall ufw podríem entrar a kibana per el nostre nom de domini (FQDN)

```
usuario@soyuz:~$ sudo ufw allow 'Nginx Full'
Reglas actualizadas
Reglas actualizadas (v6)
```

Ara passarem a instal·lar Logstash. El podrem trobar als mateixos repositoris que l'elasticsearch, pero per si decàs fem un altre cop la següent comanda:

```
usuario@soyuz:~$ echo "deb http://packages.elastic.co/logstash/2.3/debian stable main" | sudo tee -a /etc
/apt/sources.list
deb http://packages.elastic.co/logstash/2.3/debian stable main
```

I instal·lem el paquet:

```
usuario@soyuz:~$ sudo apt-get install logstash
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  logstash
```

Ja que utilitzarem Filebeat als registres dels logs del nostre client a la nostra servidor ELK, hem de crear un certificat i una clau. El certificat és utilitzat per Filebeat per verificar la identitat del servidor ELK. Ara crearem els directoris per guardar els certificats :

```
usuario@soyuz:~$ sudo mkdir -p /etc/pki/tls/certs
usuario@soyuz:~$ sudo mkdir /etc/pki/tls/private
```

Seguidament afegirem la nostra IP privada per a resoldre per ssl en aquest cas posarem la IP de localhost.

```
usuario@soyuz:~$ sudo nano /etc/ssl/openssl.cnf
```

```
[ v3_ca ]
subjectAltName = IP: 127.0.0.1
```

El que farem ara serà un cop afegida la IP , generar els certificats SSL mitjançant aquestes dues comandes:

```
usuario@soyuz:~$ cd /etc/pki/tls
usuario@soyuz:/etc/pki/tls$ sudo openssl req -config /etc/ssl/openssl.cnf -x509 -days 3650 -batch -nodes
-newkey rsa:2048 -keyout private/logstash-forwarder.key -out certs/logstash-forwarder.crt
Generating a 2048 bit RSA private key
.....................................+++
.....................................................+++
writing new private key to 'private/logstash-forwarder.key'
-----
```

Ara passarem a configurar Logstash, per poder fer-ho hem de saber que la configuració consisteix en 3 seccions: entrades, sortides i filtres. Ara seguidament crearem un fitxer de configuració anomenat *02-beats-input.conf* on podrem establir la nostra entrada filebeat:

```
  GNU nano 2.5.3          Archivo: /etc/logstash/conf.d/02-beats-input.conf

input {
  beats {
    port => 5044
    ssl => true
    ssl_certificate => "/etc/pki/tls/certs/logstash-forwarder.crt"
    ssl_key => "/etc/pki/tls/private/logstash-forwarder.key"
  }
}
```

Al fitxer d'entrada es on assignarem la clau ssl creada anteriorment, també assignarem el port d'entrada 5044. Ara haurem d'obrir el port mitjançant el firewall :

```
usuario@soyuz:~$ sudo ufw allow 5044
[sudo] password for usuario:
Reglas actualizadas
Reglas actualizadas (v6)
```

El següent que farem ara serà crear un filtre per als registres del sistema amb un arxiu anomenat *10-syslog-filter.conf* que quedaria de la següent manera:

```
  GNU nano 2.5.3          Archivo: /etc/logstash/conf.d/10-syslog-filter.conf              Modificado

filter {
  if [type] == "syslog" {
    grok {
      match => { "message" => "%{SYSLOGTIMESTAMP:syslog_timestamp} %{SYSLOGHOST:syslog_hostname} %{DATA:$
      add_field => [ "received_at", "%{@timestamp}" ]
      add_field => [ "received_from", "%{host}" ]
    }
    syslog_pri { }
    date {
      match => [ "syslog_timestamp", "MMM  d HH:mm:ss", "MMM dd HH:mm:ss" ]
    }
  }
}
```

Aquest filtre cerca els registres que estan etiquetats com a tipus "syslog" (per Filebeat), i tractarà d'utilitzar grok per analitzar els registres de registre del sistema d'entrada perquè sigui estructurat i amb possibilitat de consulta.

Per últim crearem un fitxer que emmagatzemarà les dades d'elasticsearch que estiguin apuntant a localhost:9200 utilitzant filebeat.

```
  GNU nano 2.5.3       Archivo: /etc/logstash/conf.d/30-elasticsearch-output.conf

output {
  elasticsearch {
    hosts => ["localhost:9200"]
    sniffing => true
    manage_template => false
    index => "%{[@metadata][beat]}-%{+YYYY.MM.dd}"
    document_type => "%{[@metadata][type]}"
  }
}
```

Si tota la configuració està correctament, el que farem ara serà amb una comanda, mirar que la configuració del logsgtash estigui correctament carregada.

```
usuario@soyuz:~$ sudo /opt/logstash/bin/logstash --configtest -f /etc/logstash/conf.d/
Configuration OK
```

Fem un reinici del servei i ja tindrem la configuració bàsica de logstash feta.

```
usuario@soyuz:~$ sudo systemctl restart logstash
usuario@soyuz:~$ sudo systemctl enable logstash
logstash.service is not a native service, redirecting to systemd-sysv-install
Executing /lib/systemd/systemd-sysv-install enable logstash
```

Ara ja passarem a la part mes important per a mi que es carregar els dashborads a kibana. El primer que hem de fer es baixar-nos tots els dashborads i descomprimir-los.

```
usuario@soyuz:~$ cd ~
usuario@soyuz:~$ curl -L -O https://download.elastic.co/beats/dashboards/beats-dashboards-1.2.2.zip
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  890k  100  890k    0     0   340k      0  0:00:02  0:00:02 --:--:--  341k
```

```
usuario@soyuz:~$ sudo unzip beats-dashboards-*.zip
Archive:  beats-dashboards-1.2.2.zip
670cd5d9d5911a4cbaf800a51935c76d8ce7076c
   creating: beats-dashboards-1.2.2/
 extracting: beats-dashboards-1.2.2/.gitignore
  inflating: beats-dashboards-1.2.2/CHANGELOG.md
  inflating: beats-dashboards-1.2.2/Makefile
  inflating: beats-dashboards-1.2.2/README.md
   creating: beats-dashboards-1.2.2/dashboards/
   creating: beats-dashboards-1.2.2/dashboards/dashboard/
  inflating: beats-dashboards-1.2.2/dashboards/dashboard/HTTP.json
  inflating: beats-dashboards-1.2.2/dashboards/dashboard/MongoDB-performance.json
  inflating: beats-dashboards-1.2.2/dashboards/dashboard/MySQL-performance.json
  inflating: beats-dashboards-1.2.2/dashboards/dashboard/Packetbeat-Dashboard.json
  inflating: beats-dashboards-1.2.2/dashboards/dashboard/PgSQL-performance.json
  inflating: beats-dashboards-1.2.2/dashboards/dashboard/Thrift-performance.json
  inflating: beats-dashboards-1.2.2/dashboards/dashboard/Topbeat-Dashboard.json
  inflating: beats-dashboards-1.2.2/dashboards/dashboard/Winlogbeat-Dashboard.json
```

Seguidament executant l'script ./load.sh ens carregarà varies dashborads.

Ara ens baixarem la plantilla de Filebeat desde Github, i seguidament carregarem tota la informació amb la comanda curl al nostre localhost.

```
usuario@soyuz:~$ curl -O https://gist.githubusercontent.com/thisismitch/3429023e8438cc25b86c/raw/d8c479e2
a1adcea8b1fe86570e42abab0f10f364/filebeat-index-template.json
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   991  100   991    0     0   1197      0 --:--:-- --:--:-- --:--:--  1196
usuario@soyuz:~$ curl -XPUT 'http://localhost:9200/_template/filebeat?pretty' -d@filebeat-index-template.
json
{
  "acknowledged" : true
}
```

Si ens surt el missatge "acknowledged" : true vol dir que la configuració esta correcte i que s'ha afegit correctament.

Ara configurarem el client per a que puguin enviar peticions al servidor. Primer de tot, copiarem el certificat a la maquina, ho fem des de el servidor:

```
usuario@soyuz:~$ scp /etc/pki/tls/certs/logstash-forwarder.crt usuario@192.168.15.150:/tmp
The authenticity of host '192.168.15.150 (192.168.15.150)' can't be established.
ECDSA key fingerprint is SHA256:VYndnj+XQGmRut/3KnuY1aOTjOkDCflLiq5hiHCn9p8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.15.150' (ECDSA) to the list of known hosts.
usuario@192.168.15.150's password:
logstash-forwarder.crt                                        100% 1249     1.2KB/s   00:00
```

Seguidament un cop passat el certificat, ens anem al client i creem una carpeta anomenada certs per emmagatzemar el certificat.

```
usuario@soyuz:~$ sudo mkdir -p /etc/pki/tls/certs
[sudo] password for usuario:
usuario@soyuz:~$ sudo cp /tmp/logstash-forwarder.crt /etc/pki/tls/certs/
```

Ara instal·larem el paquet filebeat al client, pero abans de tot crearem una serie de llistes:

```
usuario@soyuz:~$ echo "deb https://packages.elastic.co/beats/apt stable main" |  sudo tee -a /etc/apt/sou
rces.list.d/beats.list
deb https://packages.elastic.co/beats/apt stable main
```

Al client també instal·lem la clau GPG amb la següent comanda:

```
usuario@soyuz:~$ wget -qO - https://packages.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
OK
```

Ara cambiarem el destí dels logs per a Logstash. Es canvia al fitxer /etc/filebeat/filebeat.yml

```
    paths:
        - /var/log/auth.log
        - /var/log/syslog
    #   - /var/log/*.log
```

Ara en la línea document_type la posarem de tipus syslog que es el filtre que tenim amb Logstash.

```
    # Type to be published in the 'type' field. For Elasticsearch output,
    # the type defines the document type these entries should be stored
    # in. Default: log
    document_type: syslog
```

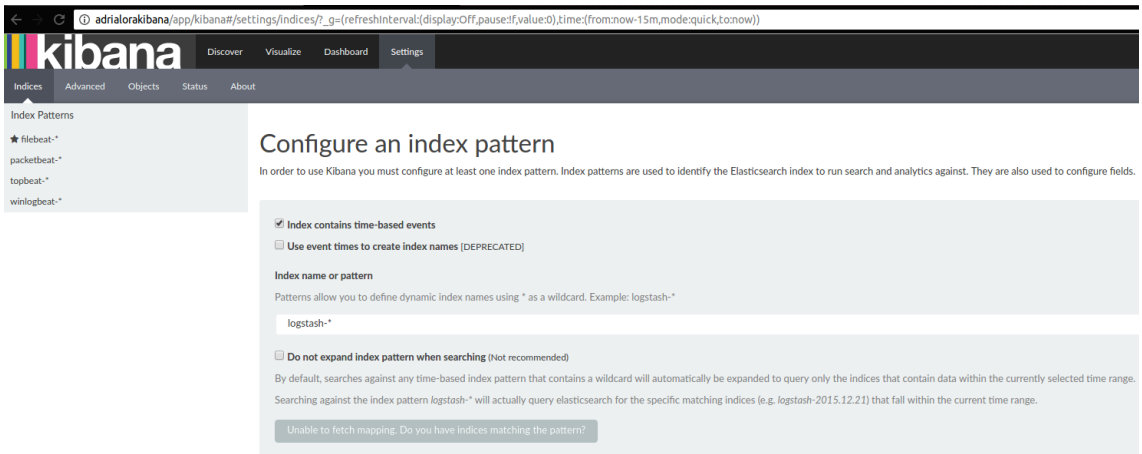
A continuació indicarem la direcció del servidor a l'apartat " Logstash as output " que es la sortida on el port es el correcte el 5044. Hem de comentar abans la línea semblant que posa elasticsearch.

```
### Logstash as output
logstash:
  # The Logstash hosts
  hosts: ["192.168.15.178:5044"]
  bulk_max_size: 1024
```

Un cop afegida la IP del servidor, activarem el TLS per indicar la ruta del nostre certificat

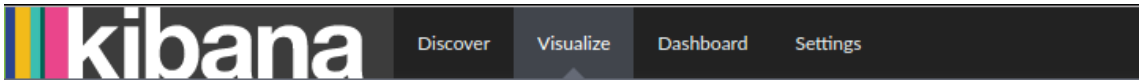
```
# Optional TLS. By default is off.
tls:
  # List of root certificates for HTTPS server verifications
  certificate_authorities: ["/etc/pki/tls/certs/logstash-forwarder.crt"]
```

Aquesta línea configura filebeat per utilitzar el certificat SSL que hem creat al nostre servidor ELK. Ara reiniciem el servidor filebeat i podrem comprovar que posant la nostra direcció FQDN anteriorment configurada accedirem al portal web de Kibana.



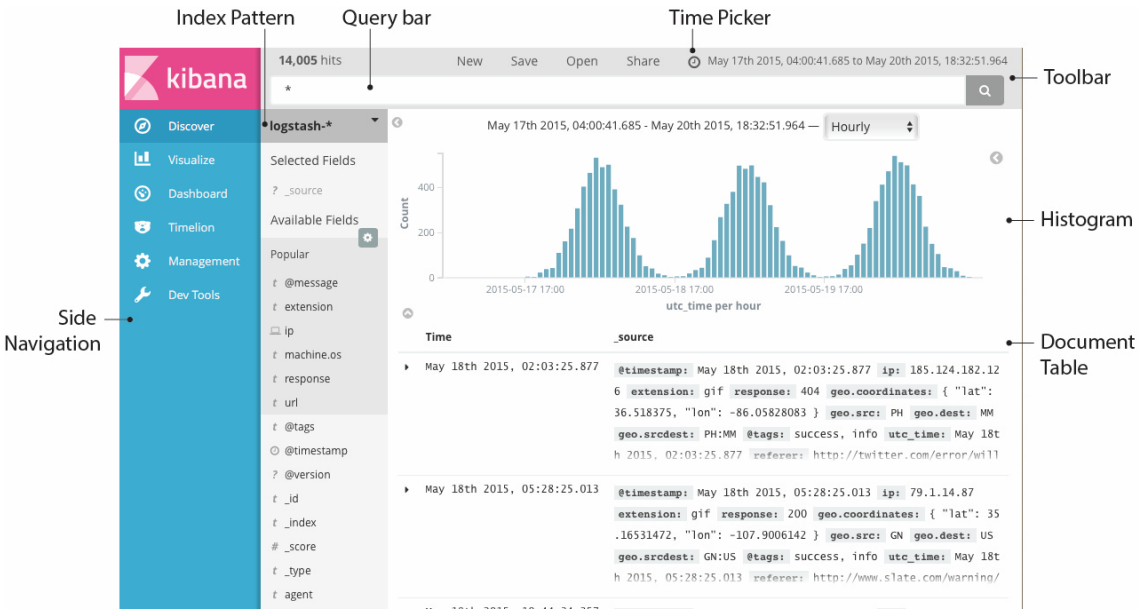
Un cop haguem posat les credencials usuari → kibanaadmin i el password que hem configurat anteriorment, ens sortirà aquest portal web.

Ara passarem a mirar les varies opcions que tenim a la nostra interfície a kibana. Consta de 4 parts principals on podem fer una serie d'accions:

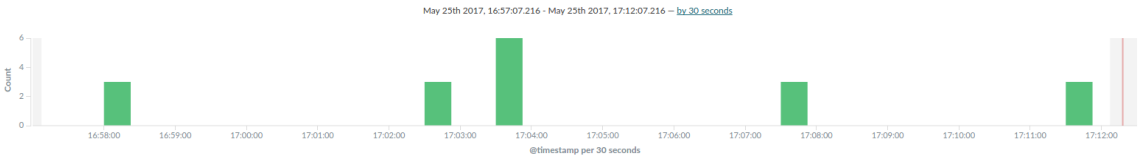


Discover → Tenim accés a tots els documents en tots els índexs que coincideix amb el patró d'índex seleccionat (en el nostre cas filebeat). Podem enviar consultes de cerca, filtrar els resultats de cerca, i veure les dades del document. També podem veure el nombre de documents que coincideixen amb la consulta de cerca i obtenir estadístiques dels valors del camp.

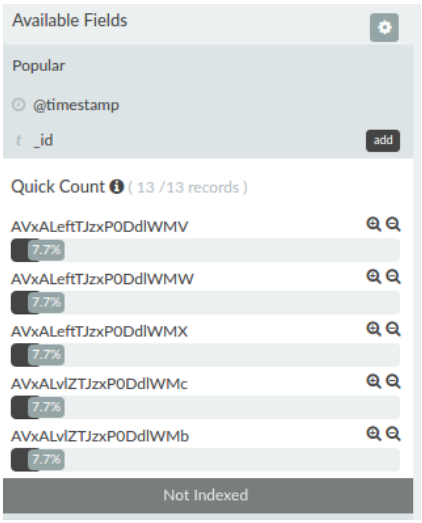
Si un camp està configurat per al patró d'índex seleccionat, la distribució de documents a través del temps es mostra en l'histograma a la part superior de la pàgina.



En aquest apartat, es on gracies a elasticsearch i logstash tenim tots els registres generals de la maquina client, tant els logs com una gràfica a la part superior que ens mostra el timestamp del rang de temps que nosaltres indiquem.

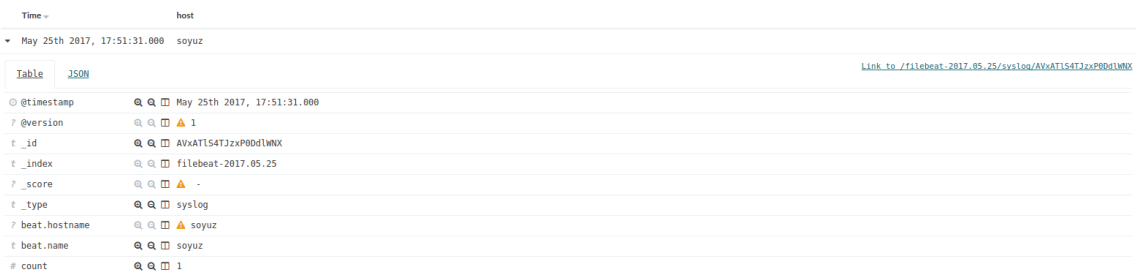


Seguidament a la part esquerra de la pantalla, tenim tota la serie de camps desplegats que ens ha proporcionat logstash. Com es a nivell de log, podem arribar a analitzar les dades a un nivell de capa molt baix.



Si fem click a cada camp, s'ens desplegara per veure mes a fons els percentatges del camp etc.

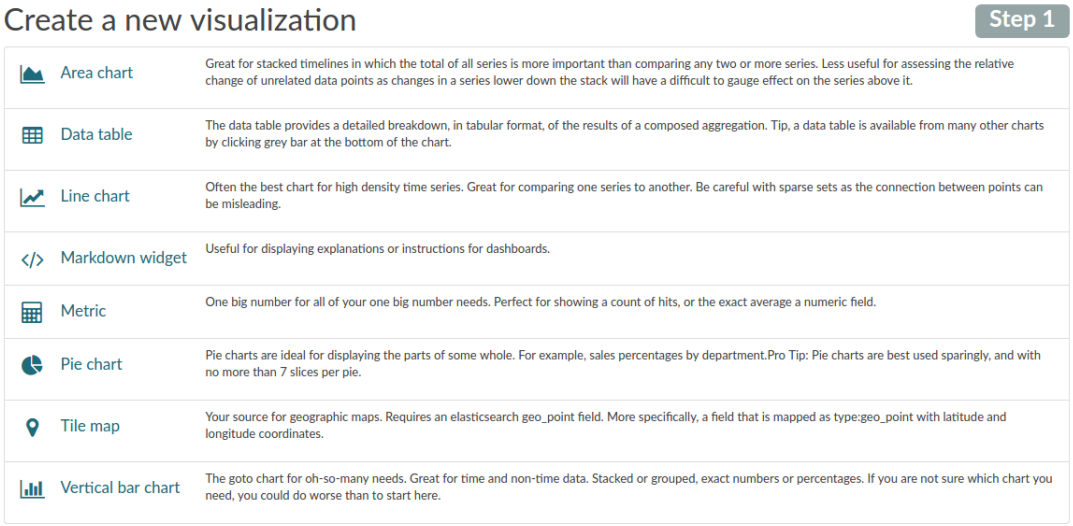
També podem agregar camps de la part esquerra i s'ens mostraràn els logs a la pàgina:



Visualize → Aquest camp ens permet crear visualitzacions de les dades en els seus índexs a Elasticsearch. A continuació, podem crear quadres de comandament que mostren les visualitzacions relacionades.

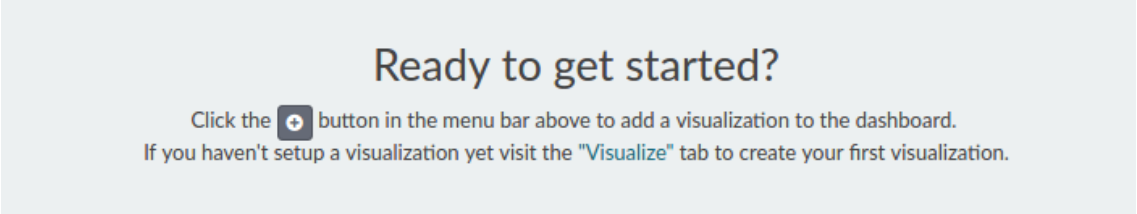
Les visualitzacions es basen en consultes d'Elasticsearch. Mitjançant l'ús d'una sèrie d'agregacions, Elasticsearch per extreure i processar les dades, pot crear gràfics que mostren les tendències, els pics, i paràmetres addicionals.

Podem crear visualitzacions de una cerca desada o crear una nova consulta de cerca.

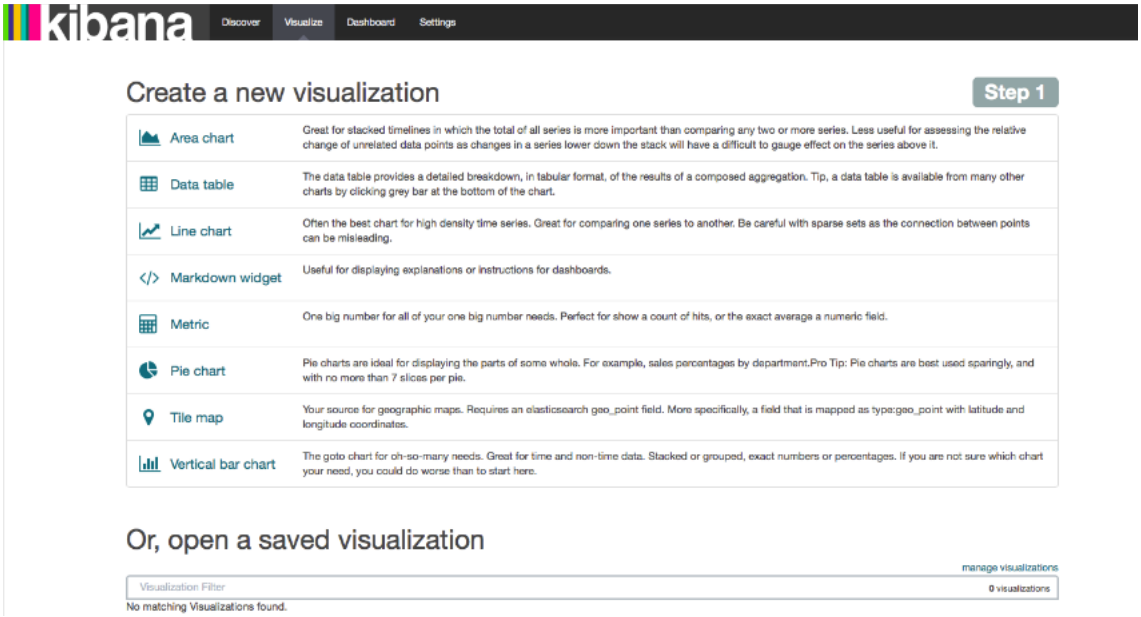


Hi han molts exemples que podem utilitzar per a fer gràfiques de mostra, això si si tenim instal·lats els plugins necessaris.

Dashboard → Aqui es on disenyarem els nostres gràfics. Per començar quan vas al panell, et surt el següent missatge:

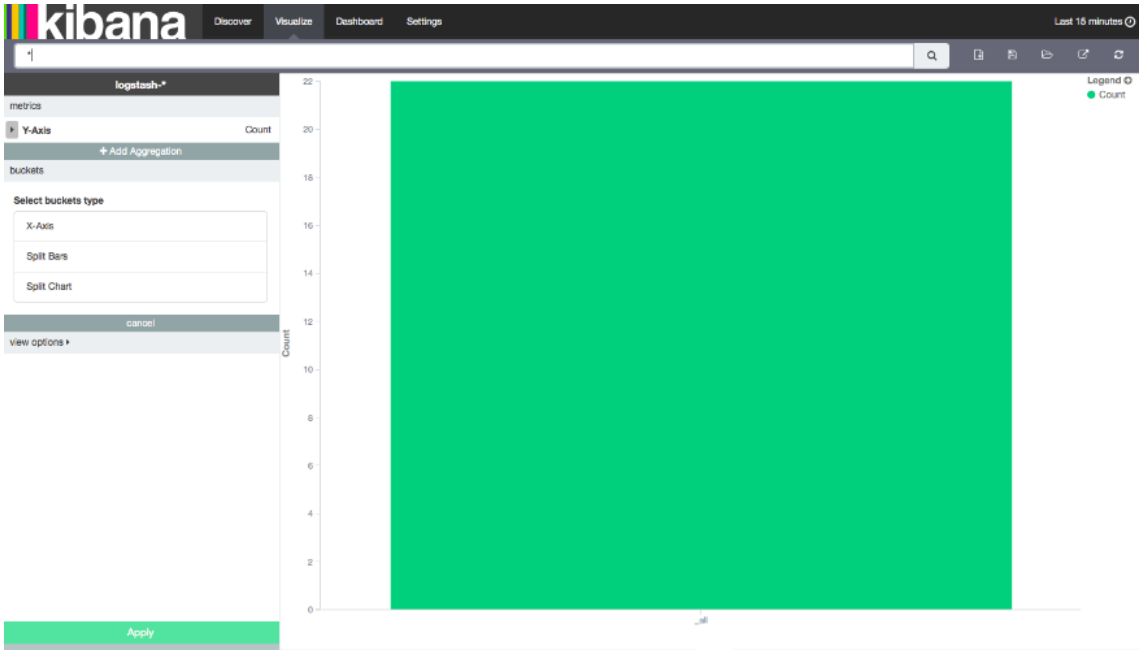


Podem importar un dashborad ja seleccionat anteriorment, o crear-ne un de nou. Nosaltres seleccionarem Visualize i veurem la següent pantalla:



Podem seleccionar qualsevol opció, nosaltres seleccionarem l'última opció Vertical bar chart i ens preguntarà el següent:

Al no tenir cap cerca creada, seleccionem From a new search i veurem un gràfic tot en verd, el que farem ara serà configurar-lo.



Aquí mostra tots els resultats per defecte i sense cap tipus d'agrupació.

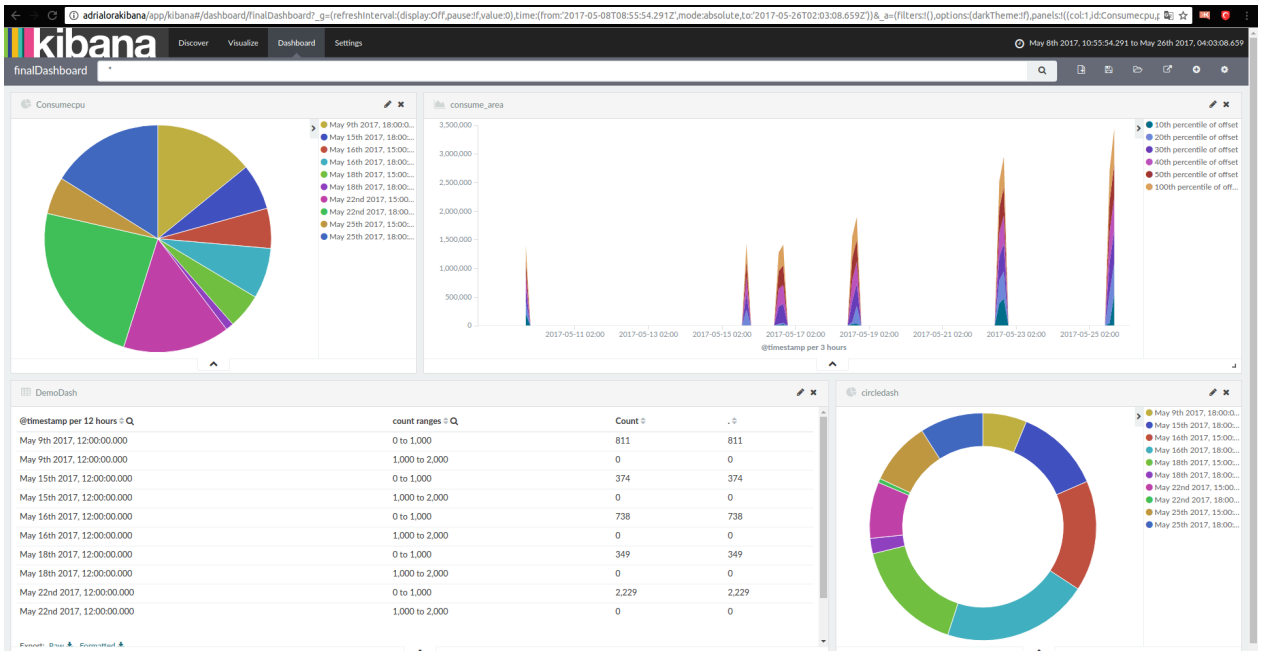
Per poder veure els accessos per minut al nostre servidor necessitem afegir un valor a l'Eix-X. Per a això punxem on diu X-Axis i en Aggregation seleccionem el tipus Date Histogram.

I en els nous camps que es mostraran:

Field: @timestamp
Interval: Minute

Donem a Apply i desem els cambis.

Un cop explicat per sobre com es crea un dashboard, ara veurem una captura de pantalla on veurem el gràfic principal que hem creat.



Aquí hem creat un petit dashboard que es un consumecpu, podem veure el percentatge de consumició de la cpu durant els dies que nosaltres li diguem. El següent dashboard que tenim, es l'activitat de les diferents parts de la memòria. Per últim els dos dashboards de la part baixa es el camp timestamp mencionat anteriorment i fent estadístiques amb ells.

L' script per consumir CPU es el següent:

```
fulload() {

        dd if=/dev/zero of=/dev/null |
        dd if=/dev/zero of=/dev/null |
        dd if=/dev/zero of=/dev/null |
        dd if=/dev/zero of=/dev/null &
 };

fulload; read;
killall dd
```

El que fa es executar més càrrega a més núclis.

3.3 GGC STACK

GCC Stack es la pila de Grafana, Graphite i Collectd es molt semblant a la pila ELK però aquesta ens proporciona una major facilitat d'importació dels dashboards, i gracies a la nova interfície de Grafana, major facilitat de configuració. L'altre pila es centraba mes en els logs, però aquesta pila es centra mes en la recol·lecció " pura " de les dades.

Començarem per instal·lar Graphite i fer la corresponent configuració. Graphite treballa per defecte amb la base de dades SQLite però he tingut varis problemes amb aquesta base de dades així que utilitzarem Postgresql per instal·lar la pila GGC, peró que es graphite? És una eina de monitorització que es pot executar en qualsevol plataforma. Els equips que utilitzen graphite per fer un seguiment del rendiment dels seus llocs web, aplicacions, serveis d'oficina i servidors en xarxa. Això va marcar el començament d'una nova generació d'eines de seguiment, pel que és més fàcil que mai per emmagatzemar, recuperar, compartir i visualitzar les dades.

Primer de tot fem un update del sistema i seguidament instal·lem la part web i el plugin graphite-carbon:

```
usuario@graphite:~$ sudo apt-get install graphite-web graphite-carbon
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

Seguidament instal·larem Postgresql amb dues dependències necessaries:

```
usuario@graphite:~$ sudo apt-get install postgresql libpq-dev python-psycopg2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

Ara en autenticarem amb l'usuari postgres i crearem un usuari anomenat graphiteuser i dues bases de dades una per graphite i una altre per grafana.

```
postgres@graphite:/home/usuario$ createuser graphiteuser --pwprompt
Enter password for new role:
Enter it again:
postgres@graphite:/home/usuario$ createdb -O graphiteuser graphite_db
postgres@graphite:/home/usuario$ createdb -O graphiteuser grafana_db
```

Ara hem de configurar el fitxer local_settings.py de graphite ja que no utilitzarem SQLite, hem de cambiar els paràmetres als que hem creat al pas anterior:

```
DATABASES = {
    'default': {
        'NAME': 'graphite_db',
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'USER': 'graphiteuser',
        'PASSWORD': 'postgres',
        'HOST': '127.0.0.1',
        'PORT': ''
    }
}
```

També haurem d'activar la zona horària que ens pertany i també activar l'autenticació per poder guardar els gràfics.

```
TIME ZONE = 'Europe/London'
```

```
## REMOTE_USER authentication. See: https://docs.djangoproject.com/en/dev/howto/auth-remote-user/
USE_REMOTE_USER_AUTHENTICATION = True
```

Un cop configurat els fitxers, el que farem será sincronitzar la base de dades:

```
usuario@graphite:~$ sudo graphite-manage syncdb
/usr/lib/python2.7/dist-packages/graphite/settings.py:249: UserWarning: SECRET_KEY is set to an unsafe de
fault. This should be set in local_settings.py for better security
  warn('SECRET_KEY is set to an unsafe default. This should be set in local_settings.py for better securi
ty')
Operations to perform:
  Synchronize unmigrated apps: account, cli, render, whitelist, metrics, url_shortener, dashboard, compos
er, events, browser
  Apply all migrations: admin, contenttypes, tagging, auth, sessions
Synchronizing apps without migrations:
  Creating tables...
    Creating table account_profile
    Creating table account_variable
    Creating table account_view
    Creating table account_window
    Creating table account_mygraph
    Creating table dashboard_dashboard
    Creating table events_event
    Creating table url_shortener_link
    Running deferred SQL...
```

Ara configurarem el plugin carbon. El primer que farem serà al fitxer de configuració global que és /etc/carbon/carbon.conf, caldria editar-lo per configurar carbon perquè escolti per l'interficie de xarxa 192.168.3.12

Seguidament copiem la configuració al directori que percotca.

```
usuario@graphite:~$ sudo sed -i.bak s/0.0.0.0/192.168.3.12/g /etc/carbon/carbon.conf
usuario@graphite:~$ sudo cp /usr/share/doc/graphite-carbon/examples/storage-aggregation.conf.example /etc
/carbon/storage-aggregation.conf
```

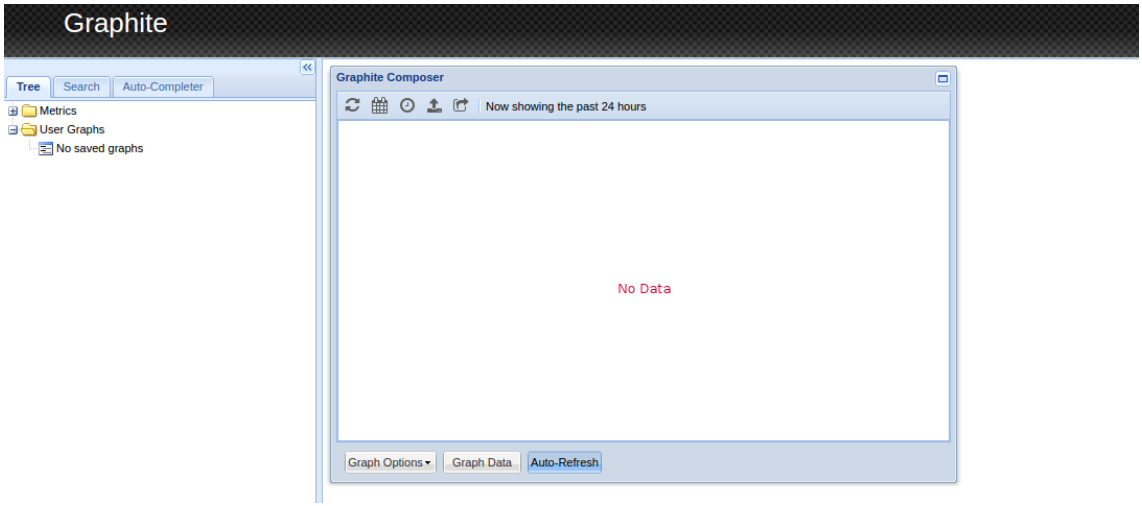
Per últim activarem carbon al iniciar el sistema, per això editem el fitxer /etc/default/graphite-carbon:

```
  GNU nano 2.5.3                Archivo: /etc/default/graphite-carbon

# Change to true, to enable carbon-cache on boot
CARBON_CACHE_ENABLED=true
```

Ara iniciarem el servei carbon per poder després configurar-lo amb grafana i collectd:

```
usuario@graphite:~$ sudo service carbon-cache restart
[sudo] password for usuario:
```

Ara si posem a la nostra barra de direccions la IP del nostre Graphite hauria de sortir l'interficie web.



Com podem veure ens surt com que no tenim dades. Per això hem de configurar el nostre collectd amb la IP que apunti a la màquina virtual de Graphite. Primerament parlarem una mica que es collectd.

Collect, és un dimoni que bàsicament el que fa és capturar certes estadístiques del sistema i bolca aquestes dades en un gràfic. Una de les seves principals característiques és que a diferència d'altres sistemes similars collectd no val de l'ús de crontab per col·lectar aquestes dades, sinó que compta amb el seu propi dimoni per a aquest propòsit.

És extensible mitjançant plugins i per defecte ja ve amb diversos d'ells. Podeu fer-li un cop d'ull a la documentació oficial, que és bastant completa: http://collectd.org/documentation.shtml

El que farem ara es agafar un entorn web com graphite que sigui el encarregat de recollir les dades que li passa collectd i representarles en un gràfic, finalment també instal·larem Grafana per crear dashboards a gran escala i més específics.

Comencem actualitzant el index de paquets local

```
usuario@collectd:~$ sudo apt-get update
[sudo] password for usuario:
Des:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]
Obj:2 http://es.archive.ubuntu.com/ubuntu xenial InRelease
Des:3 http://es.archive.ubuntu.com/ubuntu xenial-updates InRelease [102 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu xenial-backports InRelease [102 kB]
Des:5 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages [268 kB]
```

Seguidament instal·lem el programa collectd i el plugin utils:

Amb aquest paquets també s'instal·laran una interficie de control auxiliar. Ara el que farem serà configurar el fitxer principal de configuració de collectd.

```
usuario@collectd:~$ sudo apt-get install collectd collectd-utils
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
collectd ya está en su versión más reciente (5.5.1-1build2).
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  linux-headers-4.4.0-21 linux-headers-4.4.0-21-generic linux-image-4.4.0-21-generic linux-image-extra-4.4.0-21-generic
```

Nosaltres activarem tots els plugins que volguem utilitzar (si no estan activats cal escriure una linea com les que veiem a continuació) per poder recol·lectar dades.

```
</Plugin>

LoadPlugin disk
LoadPlugin df
LoadPlugin interface
LoadPlugin network
LoadPlugin users
LoadPlugin cpu
LoadPlugin entropy
LoadPlugin load
LoadPlugin memory
LoadPlugin swap
LoadPlugin uptime
LoadPlugin mysql
LoadPlugin processes
LoadPlugin sensors
LoadPlugin exec
LoadPlugin write_graphite
```

Com podem veure hem afegit el plugin de write_graphite que ara configurarem amb la IP del host on instal·lem graphite i alguns paràmetres més com el port que serà el per defecte, i el prefix que serà el nostre recol·lector.

```
<Plugin write_graphite>
    <Carbon>
        Host "192.168.15.189"
        Port "2003"
        Prefix "collectd."
        StoreRates true
        AlwaysAppendDS false
        EscapeCharacter "_"
    </Carbon>
</Plugin>
```

Podem activar tants plugins com volguem sempre al mateix fitxer. Podem definir esquemes de mètriques peró això aniria a graphite.

Per ultim, instal·larem la nova versió de grafana. Ho instal·larem amb unes comandes básiques que veiem a continuació, peró abans parlarem una mica de grafana.

Grafana és una aplicació de codi obert que ens permet, fàcilment, desenvolupar quadres de comandaments sobre mètriques basades en sèries de temps, tant per mostrar i analitzar infraestructura (servidors, routers, sensors industrials, elements domòtics, etc.) com aplicacions pròpies (veure demo) . Sol anar acompanyada d'altres aplicacions / plugins que la complementen; elasticsearch o influxdb1, per exemple, per proporcionar-li les fonts de dades que alimenten les gràfiques i quadres de comandament creats. I podem crear els nostres propis plugins.

Grafana suporta 3 tipus de connectors:

- Panell → el quadre de comandament pròpiament dit. Podem crear tots els que vulguem.
- Datasource → proporciona les dades als panells.
- App-→ una aplicació completa dins de grafana que pot tenir els seus propis panells, datasources, etc.

Començem per actualitzar la llista de repositoris d'ubuntu per poder instal·lar l'ultima versió de grafana.

```
usuario@newgrafana:~$ echo 'deb https://packagecloud.io/grafana/stable/debian/ wheezy main' |  sudo tee -
a /etc/apt/sources.list
[sudo] password for usuario:
deb https://packagecloud.io/grafana/stable/debian/ wheezy main
```

Seguidament a la página de grafana tenim una URL de descárrega per si decás no s'ens descarrega bé el paquet:

```
usuario@newgrafana:~$ get https://s3-us-west-2.amazonaws.com/grafana-releases/release/grafana_4.3.1_amd64.deb
--2017-05-29 17:30:13--  https://s3-us-west-2.amazonaws.com/grafana-releases/release/grafana_4.3.1_amd64.
deb
Resolviendo s3-us-west-2.amazonaws.com (s3-us-west-2.amazonaws.com)... 52.218.160.40
Conectando con s3-us-west-2.amazonaws.com (s3-us-west-2.amazonaws.com)[52.218.160.40]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 47222394 (45M) [application/x-debian-package]
Grabando a: “grafana_4.3.1_amd64.deb”

grafana_4.3.1_amd64.deb     28%[=========>                          ]  12,87M  1,36MB/s    eta 45s
```

```
usuario@newgrafana:~$ sudo dpkg -i grafana_4.3.1_amd64.deb
```

Un cop instal·lat, anirem a la configuració per a poder accedir a la base de dades, en aquest cas la configurarem amb postgresql. El fitxer es /etc/grafana/grafana.ini

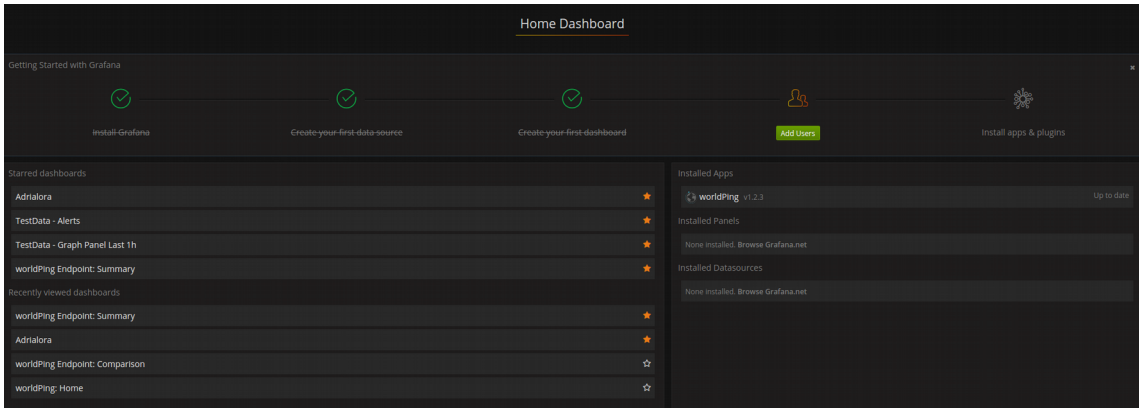
```
# Either "mysql", "postgres" or "sqlite3", it's your choice
;type = postgres
;host = 127.0.0.1:5432
;name = grafana
;user = postgres
# If the password contains # or ; you have to wrap it with trippel quotes. Ex """#password;"""
;password = postgres
```

Ara haurem de integrar a grafana, graphite. Pero abans d'això el que farem será reiniciar el servidor i si tot ha anat bé, veurem al status que esta sense errors.

```
usuario@newgrafana:~$ sudo service grafana-server restart
usuario@newgrafana:~$ sudo service grafana-server status
● grafana-server.service - Grafana instance
   Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; disabled; vendor preset: enabled)
   Active: active (running) since lun 2017-05-29 15:34:06 CEST; 2h 8min ago
     Docs: http://docs.grafana.org
 Main PID: 22104 (grafana-server)
    Tasks: 9
   Memory: 10.5M
      CPU: 15.553s
   CGroup: /system.slice/grafana-server.service
           └─22104 /usr/sbin/grafana-server --config=/etc/grafana/grafana.ini --pidfile= cfg:default.path
```

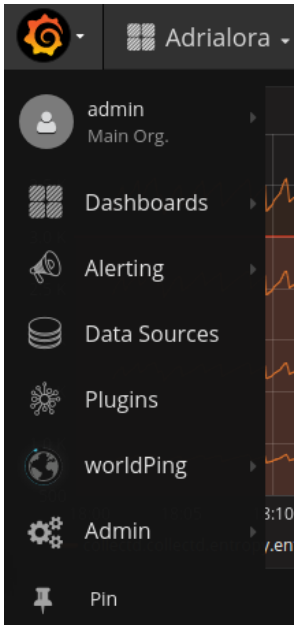
Si introduïm la nostra direcció IP seguit del port 3000 ens apareixerá el login de grafana.

Si introduïm les credencials (per defecte admin, admin) ens apareixerá la finestra principal de l'interficie web.



Aquesta interficie es la principal on veiem el sdiferents dashboards que tenim a grafana. Podem marcarlos amb una ★ per tenir controlats els nostres dashboards preferits.

Ara pasarem a explicar una mica per sobre el menú principal. Tenim els següents apartats:



Perfil d'usuari → Es divideix en dues parts:

- You: Aqui podem trobar tota la informació del perfil i l'opció de tancar la sessió.
- Main Org : Aqui trobarem l'apartat de preferències, adició d'usuaris i creacions de noves organitzacións.

Dashboards → Aqui está tot el tema dels gràfics, l'importació i exportació dels dashboards i tot el que té que veure amb ells.

Alerting → Apartat per crear alertes tant dels dashboards com d'errors del sistema.

Data Sources → Aqui es on escollirem amb quin recopilador treballar els dashborads (collectd, carbon, influxDB etc).

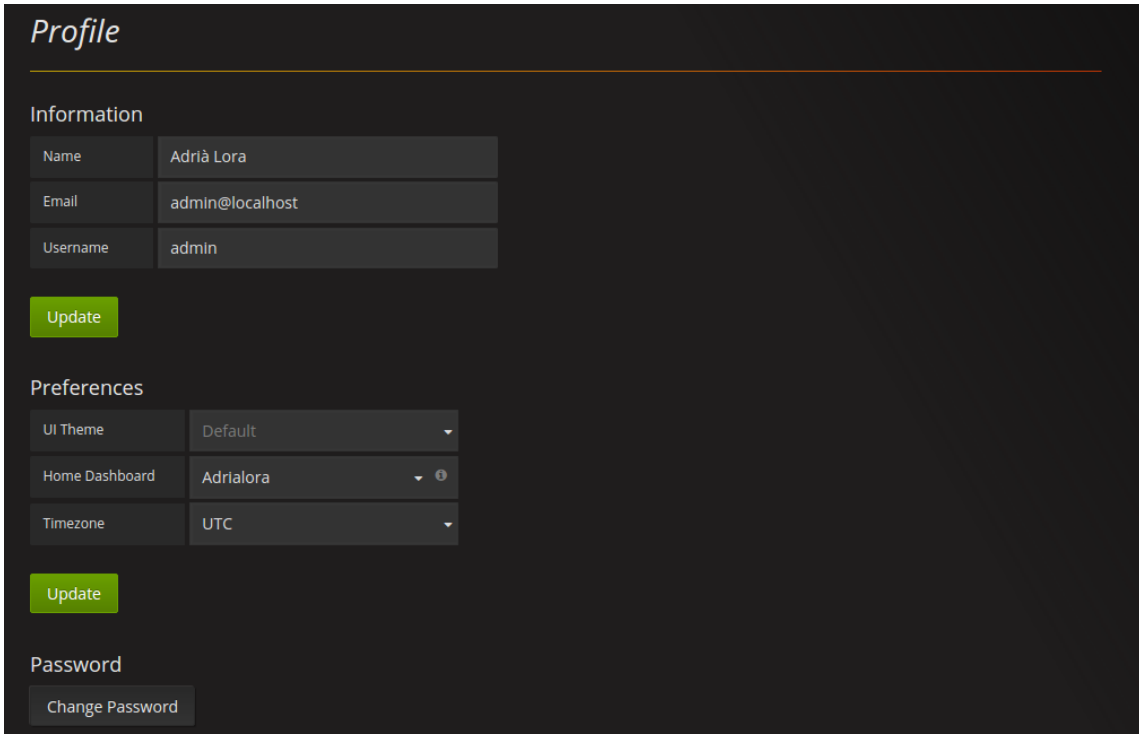
Plugins → Es un dels apartats mes importants de Grafana. Aqui podem importar plugins per fer diferents accions amb la monitorització de les dades

WorldPing → Es un plugin instal·lat a Grafana que mes endavant explicarem per que serveix.

Admin → Es la part de tota la configuració a nivell d'administració.

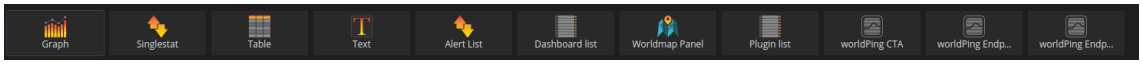
Ara pasarem a explicar mes en profunditat els apartats mencionats anteriorment.

Perfil d'usuari → Al perfil d'usuari podem veure com configurar tota la informació tant del nom, correu electrònic, nom d'usuari etc. També podem tocar el tema de preferències, el dashboard d'inici i la zona horaria. Per últim tenim opció a cambiar el password.



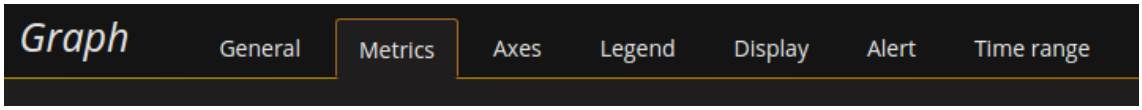
Dashboards → Per crear un dashboard al nou grafana el primer que hem de fer es anar a l'opció New + Create New ens sortirá una area de treball en blanc.

Despres a la part superior esquerra de la pantalla, tindrem un menú amb una serie d'opcions. Nosaltres el primer que farem será afegir un panell (Add Panel) i li direm que volem un gráfic (Graph).



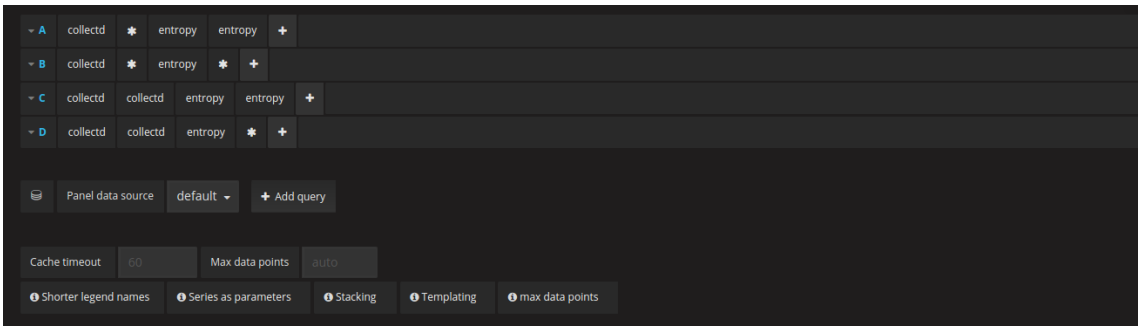
Com veiem a la imatge, si piquem al títol del gràfic, ens surten unes opcions, de les quals seleccionem Edit.

Com podem veure, tenim una serie d'apartats:

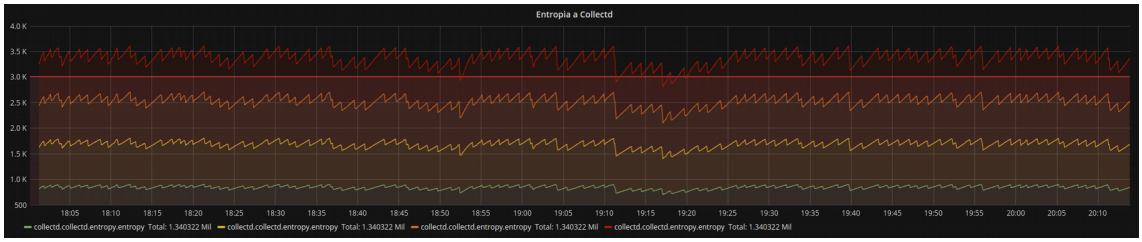


General → Aquest apartat general, es el que ens permet posar un titol, diferents maneres d'espaiat al gráfic, si el volem amb fons transparent o no , etc.

Metrics → En aquest apartat es on formem les querys. Les podem generar de dues maneres, ja sigui mitjançant desplegables (mètode mes fàcil) o mitjançant consultes amb sintaxi grafana.

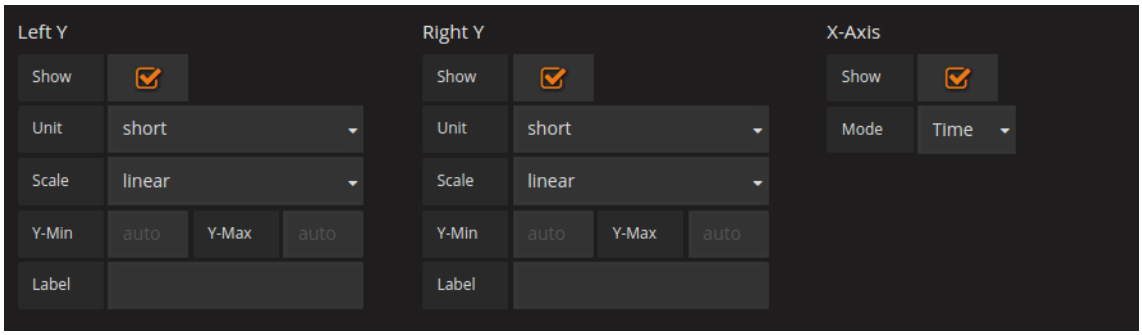


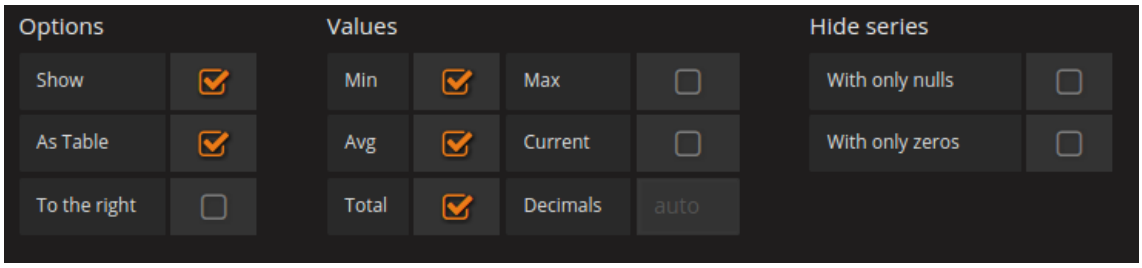
Com podem veure, primer seleccionem al desplegable el recol·lector de dades que volem, en aquest cas tenim o collectd o carbon, i seguidament anem agafant les dades que volem consultar per formar el grafic.



A l'imatge veiem el gráfic de la entropia que generem entre els 4 nuclis del nostre PC. Podem fer tantes consultes com vulguem graficar en el mateix gràfic.

Area & Grid → En aquest apartat es on controlem tot el tema de les dades a nivell estadistic. Es a dir, quin nom té el tipus de dades que mostrem, quines son les maximes, minimes, mitjanes de les dades, podem ordenar les dades de la forma que volguem.

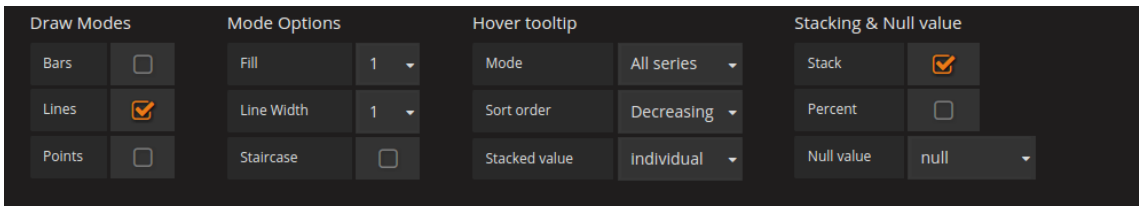




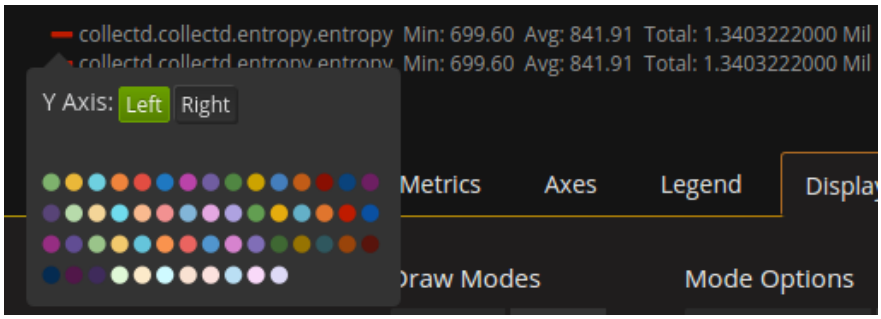
També una cosa molt curiosa es que podem fer una mini-taula amb les mitajes, maximes o minimes de les dades representades anterioment.

min	avg	total
700	842	1.340322 Mil
700	842	1.340322 Mil
700	842	1.340322 Mil
700	842	1.340322 Mil

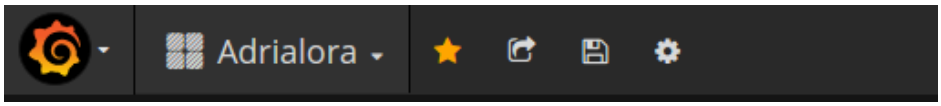
Display Styles →Aquí trobarem diferents opcions a la hora de modificar el gràfic ja siguin dades o l'estil i ordre del gràfic. Una pestanya molt interessant es la que posa stack que serveix per separar les diferents dades al gràfic i poder analitzarles millor.



A line options podem donar-li estil, com més color d'àrea etc. Una de les opcions que no tenim al menú però que si es sobre els estils es el cambi de color a les línies de gràfic de les dades que es fa fent click al color que veiem a la imatge.



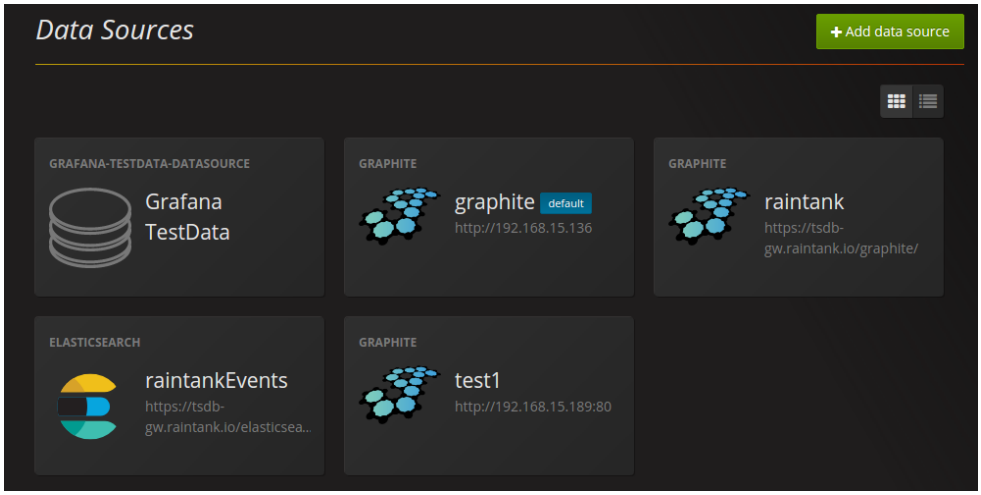
Doncs ja hem vist com crear un dashboard i quines son les multiples opcions que tenim per la creació. Es poden crear també taules, rankings de dades, entre d'altres opcions. Un cop tenim el dashboard creat, el guardem amb l'opció que tenim a la part superior de la pantalla



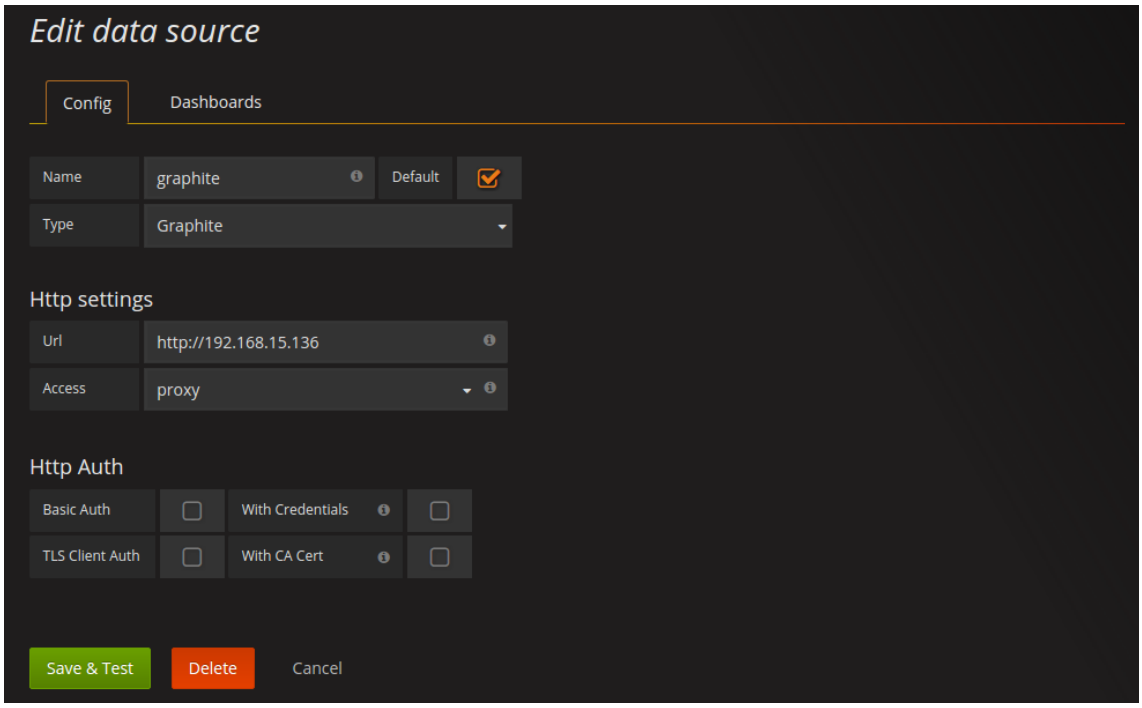
i també el podem afegir a la nostra llista de preferits amb l'icona de la estrella que veiem a la captura anterior.

Alerting → Aquesta es l'area que tenim per configurar també les alertes però a nivell general. Es molt semblant a la configuració d'alertes dels dashboards Podem veure-ho a la següent imatge:

Data Sources → Es l'apartat on tindrem una llista dels recolectors de dades que tenim configurats a grafana. Jo tinc les següents :



Per defecte hem escollit graphite (ho podem veure que está per defecte) que té la següent configuració:



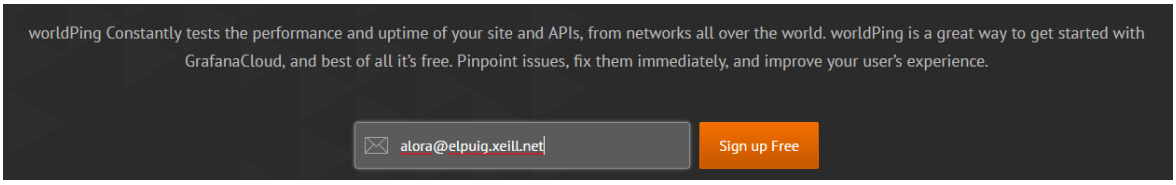
Aqui definim la IP del servidor de graphite, el tipus d'accés ja sigui per TLS-SSL. Podem tenir fins autenticació per HTTP.

Plugins → Hi han infinitats de plugins a Grafana. Tenim tant Panels, Data sources o Apps. Nosaltres ens centrarem en el plugin que hem instal·lat anomenat WorldPing.

WorldPing és un plugin per a Grafana que posa a prova contínuament, el temps de resposta i les alertes sobre el rendiment global i la disponibilitat de les pàgines web que nosaltres li indiquem perquè pugui identificar problemes, solucionar-los immediatament, i millorar l'experiència de l'usuari.

Podem utilitzar worldPing per obtenir una visió en temps real del rendiment i la disponibilitat de qualsevol pàgina web. El bo que té aquest plugin es que t'alerta en temps real, qualsevol anomalia ja sigui una caiguda o que la pàgina no respongui temporalment, per ajudar a l'usuari a solucionar el problema mes aviat.

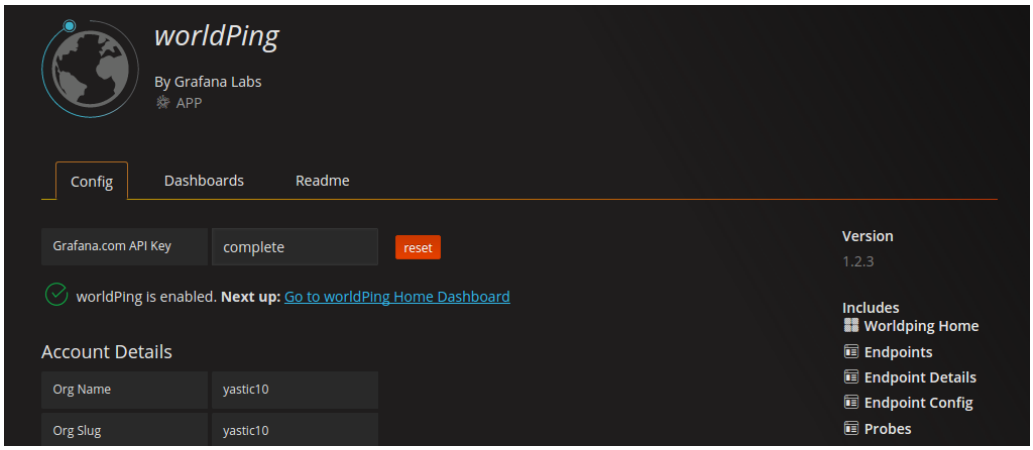
Caracteristiques → Aquest plugin es molt fácil d'instal·lar, només cal iniciar sessió a la web de grafana (si no tenim compte ens la creem perquè será necessaria per la instal·lació del plugin)



i un cop tinguem la sessió iniciada li donem a 

per començar l'instal·lació. El primer pas que hem de fer es la comanda per instal·lar el plugin en el sistema (captura) i un cop ja l'haguem executat, seguidament anirem a l'apartat plugins de la nostra pàgina de Grafana i farem click a 

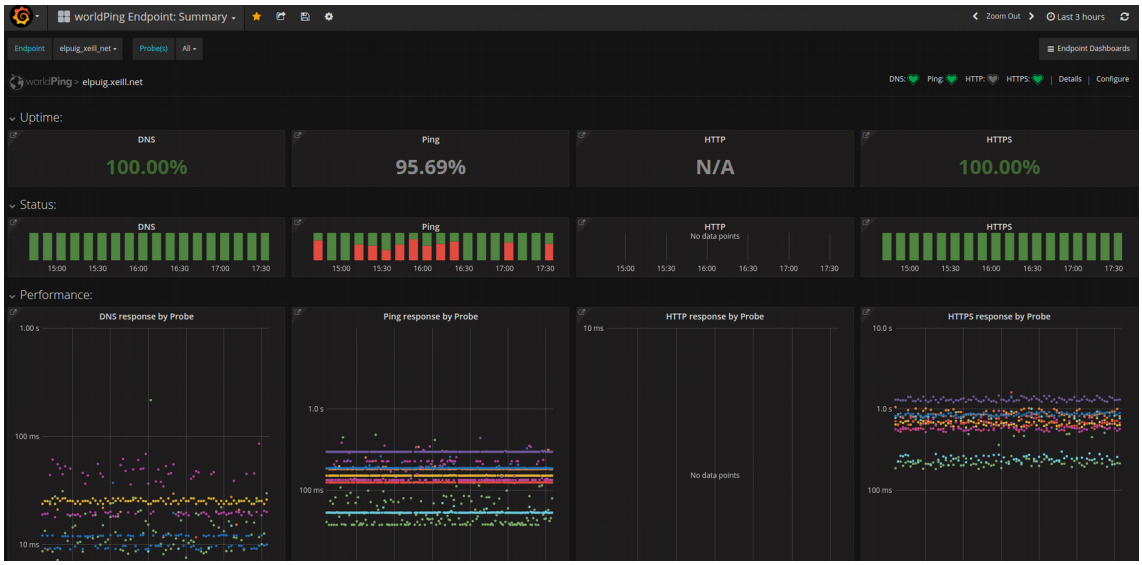
per activar el plugin. Es possible que a alguns usuaris ens demani una API Key que la podrem trobar fácilment iniciant sessió a la pàgina web de grafana.



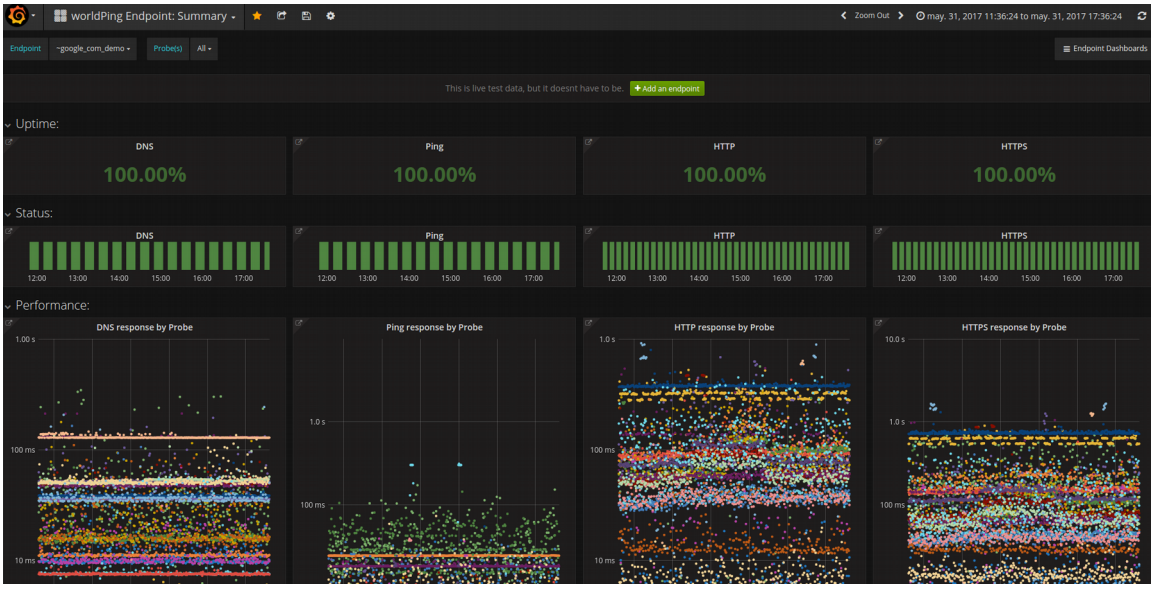
Protocols compatibles → El que podem visualitzar als dashboards de Worldping es el següent:

- el temps d'activitat i rendiment de ping (per exemple. latència, pèrdua, fluctuació)
- el temps d'activitat i rendiment DNS (per exemple. latència, les respostes)
- HTTP i HTTPS (temps d'activitat i rendiment)

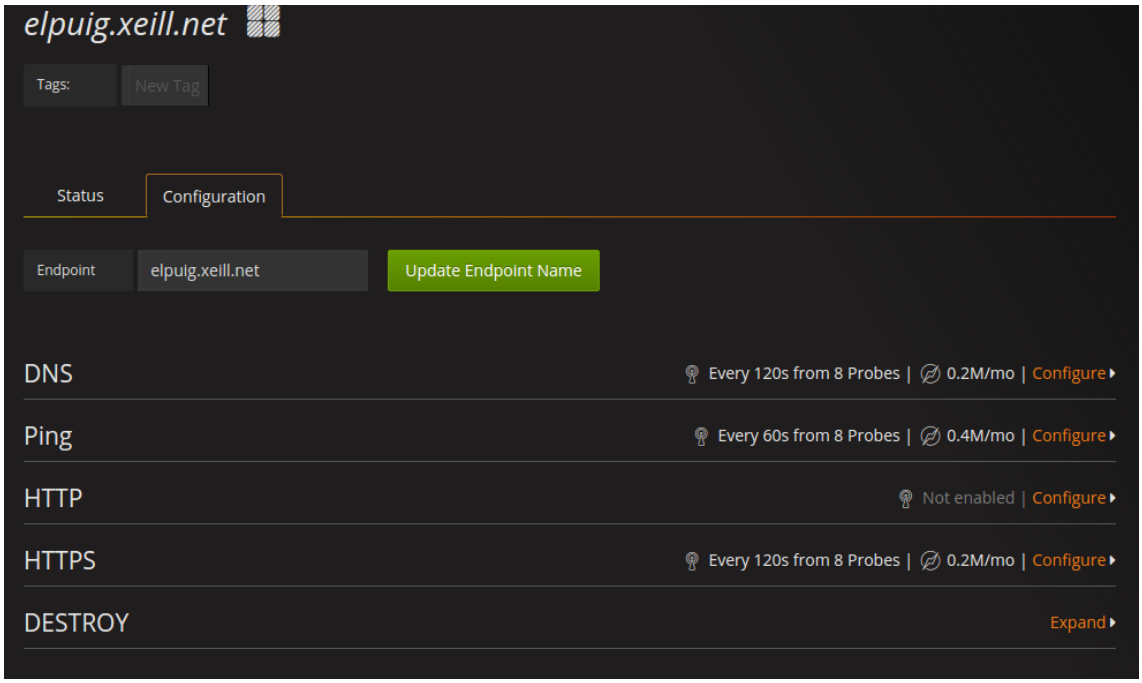
Es pot configurar cada punt final a les nostres necessitats exactes que es una mica el que hem fet, per exemple hem afegit la web del centre per veure com es comporta i hem deshabilitat el protocol HTTP per que no l'utilitza.



Aquí veiem dues imatges de el comportament del plugin amb una gran web com es Google.es i la compararem amb la nostra web del centre. (captura x2)Com podem veure hem personalitzat la configuració al dashboard del centre desactivant el protocol HTTP.



A la part superior dreta de la pantalla tenim l'opció configure que es on podrem configurar tots els paràmetres tant activar com desactivar protocols, com afegirne de nous.



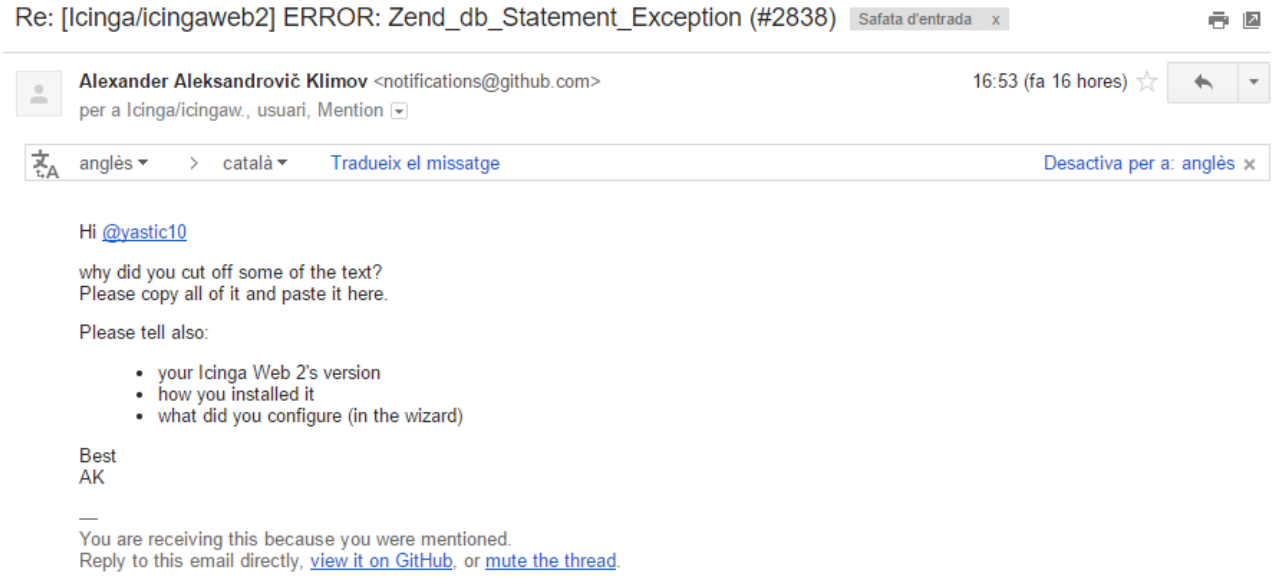
Alertes → El control d'alertes es molt important a la hora de la monitorització. Per això nosaltres podem configurar alertes de cualsevol gràfic per tindre constància d'alguna caiguda o anomalia a la xarxa.

Els errors es validen a través de múltiples ubicacions per reduir els falsos positius.

A continuació veurem el dashboard principal que tenim configurats a Grafana. Començarem per el Dashboard principal on integrem a la part alta de la pantalla l'entropia generada per els nuclis de la CPU, seguidament una taula amb els diferents estats de la memória i a la part baixa el temps de resposta i el plugin de WorldPing.

4.Gestió D'errors

Icinga2 → Just un dia abans de la entrega final, em van respondre a la issue penjada sobre l'error d'icinga2 que tenia (problema de autenticació) . En 24h no em va donar temps a respondre a la seva resposta ja que tenia temes que acabar per lliurar la memòria en bon estat.



Re: [Icinga/icingaweb2] ERROR: Zend_db_Statement_Exception (#2838) Safata d'entrada x

Alexander Aleksandrovič Klimov <notifications@github.com> 16:53 (fa 16 hores)

per a Icinga/icingaw., usuari, Mention

anglès › català Tradueix el missatge Desactiva per a: anglès ×

Hi @yastic10

why did you cut off some of the text?
Please copy all of it and paste it here.

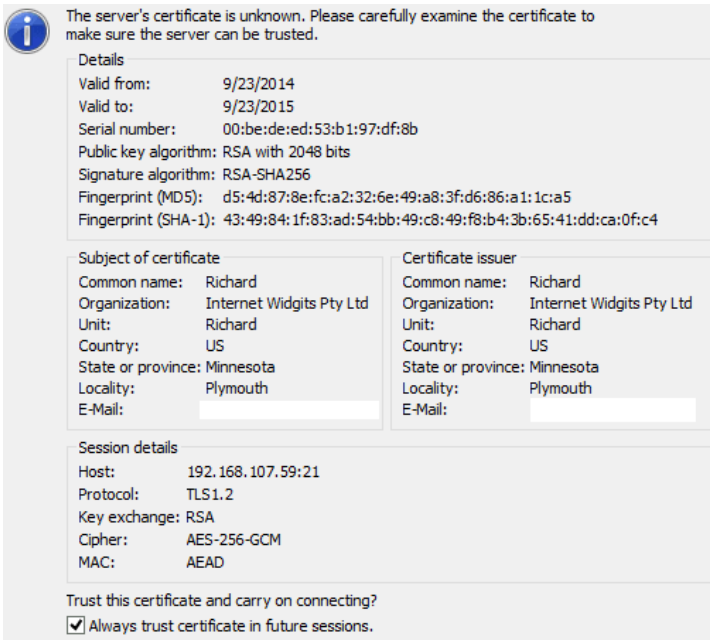
Please tell also:

- your Icinga Web 2's version
- how you installed it
- what did you configure (in the wizard)

Best
AK

—
You are receiving this because you were mentioned.
Reply to this email directly, view it on GitHub, or mute the thread.

ELK Stack → Vem tindre un error amb l'autenticació del client ELK amb el servidor, no es conectaven i el problema estaba en que habiem d'activar les conexions desconegudes ja que els certificats per defecte només agafaba la IP de localhost (la de loopback 127.0.0.1)



The server's certificate is unknown. Please carefully examine the certificate to make sure the server can be trusted.

Details

Valid from:	9/23/2014
Valid to:	9/23/2015
Serial number:	00:be:de:ed:53:b1:97:df:8b
Public key algorithm:	RSA with 2048 bits
Signature algorithm:	RSA-SHA256
Fingerprint (MD5):	d5:4d:87:8e:fc:a2:32:6e:49:a8:3f:d6:86:a1:1c:a5
Fingerprint (SHA-1):	43:49:84:1f:83:ad:54:bb:49:c8:49:f8:b4:3b:65:41:dd:ca:0f:c4

Subject of certificate

Common name:	Richard
Organization:	Internet Widgits Pty Ltd
Unit:	Richard
Country:	US
State or province:	Minnesota
Locality:	Plymouth
E-Mail:	

Certificate issuer

Common name:	Richard
Organization:	Internet Widgits Pty Ltd
Unit:	Richard
Country:	US
State or province:	Minnesota
Locality:	Plymouth
E-Mail:	

Session details

Host:	192.168.107.59:21
Protocol:	TLS1.2
Key exchange:	RSA
Cipher:	AES-256-GCM
MAC:	AEAD

Trust this certificate and carry on connecting?
☑ Always trust certificate in future sessions.

L'error que donaba era semblant a aquest, era un error de certificat desconegut peró amb la diferència que no reconeixa un altre IP que la local.

5. Pròxims Objectius

Integrar Elasticsearch amb Grafana ja que son dues de les eines que he utilitzat per separat i m'agradaria probarles juntes.

Acabar de arreglar l'error d'icinga2 i probarlo ja que es bastant diferent a les dos suites probades

Probar grafana amb influxDB ja que te molt bones critiques

Acabar de probar a fons grafana per poder implementar-lo a la llarga a nivell empresarial.

Instal·lar la pila TICK Stack que també obté molt bones critiques i per falta de temps no he pogut du a terme.

Integrar les maquines virtuals VMWare que hi ha a la meva empresa amb kibana o grafana.

6.Conclusió

Ha sigut un projecte molt educatiu a nivell personal perquè era una cosa que tenia pensada implementar a la meva empresa. Hi ha hagut una investigació prèvia molt extensa, descartant molt software i quedant-me realment amb el que creia que seria mes útil per a mi. La falta de temps ha perjudicat molt a la hora d'aconseguir objectius al projecte, ja que vaig estar molt de temps amb errors d'instal·lació a icinga2 i he agut d'apretar al tram final.

La meva valoració ha estat satisfactòria ja que m'ha resultat molt productiu i m'he ficat de plé en el món de la monitorització. I m'ha sorprès de tot el que es pot arribar a monitoritzar, tant a nivell de logs com a nivell de hardware.

Uns quants dels objectius que teníem que era comparar softwares diferents per la monitorització ha estat aconseguit, comparant GGC Stack amb ELK Stack i la veritat que em sento molt satisfet d'on he pogut arribar jo sol.

També ha sigut difícil treballar sol estant acostumat a treballar amb equip, i l'organització ha de ser mes estricte a la hora de treballar.

7. Bibliografia

COLLECTD

https://www.digitalocean.com/community/tutorials/how-to-configure-collectd-to-gather-system-metrics-for-graphite-on-ubuntu-14-04

http://packages.ubuntu.com/search?keywords=collectd

https://collectd.org/download.shtml

GRAPHITE

https://www.digitalocean.com/community/tutorials/how-to-install-and-use-graphite-on-an-ubuntu-14-04-server

https://gist.github.com/albertohm/5697429

https://www.hostedgraphite.com/docs/dashboards/grafana-dashboards.html

GRAFANA

https://grafana.com/grafana

http://giverhell.com/2016/07/25/installing-configuring-grafana-ubuntu/

https://community.rackspace.com/products/f/25/t/6800

https://worldping.raintank.io/docs/

http://play.grafana.org/dashboard/db/grafana-play-home?orgId=1&from=1496210603342&to=1496212968665

https://grafana.com/grafana/download

ICINGA E ICINGA 2

https://docs.icinga.com/icinga2/latest/doc/module/icinga2/toc

https://docs.icinga.com/

http://linoxide.com/monitoring-2/install-configure-icinga-linux/

https://lowendbox.com/blog/server-monitoring-with-icinga-2-part-1-the-server-ubuntu-host/

MUNIN

http://munin-monitoring.org/

https://www.digitalocean.com/community/tutorials/how-to-install-the-munin-monitoring-tool-on-ubuntu-14-04

http://munin-monitoring.org/wiki/munin.conf

CACTI

http://www.cacti.net/

http://www.cacti.net/downloads/docs/html/unix_configure_cacti.html

NAGIOS

https://www.nagios.org/

https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/3/en/objectdefinitions.html

https://www.digitalocean.com/community/tutorials/how-to-install-nagios-4-and-monitor-your-servers-on-ubuntu-14-04

https://en.wikipedia.org/wiki/Nagios

http://www.tecmint.com/how-to-add-linux-host-to-nagios-monitoring-server/

ELK STACK

https://www.quora.com

https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elk-stack-on-ubuntu-16-04

https://www.elastic.co/downloads

https://www.elastic.co/guide/en/elastic-stack/current/installing-elastic-stack.html

https://linuxacademy.com/howtoguides/posts/show/topic/12167-elk-stack-50-installation-and-configuration-part-2-kibana-filebeat

https://linoxide.com/ubuntu-how-to/setup-elk-stack-ubuntu-16/

https://www.elastic.co/guide/en/logstash/current/configuration.html

https://www.elastic.co/guide/en/beats/filebeat/master/configuring-ingest-node.html

https://logz.io/blog/siem-dashboard-aws-elk-stack/

https://blog.socialcops.com/engineering/granular-geospatial-dashboards-elk-stack/

https://logz.io/learn/complete-guide-elk-stack/