

**MTENDER SYSTEM DOCUMENTATION**

**Central Database Unit**

**Administrator Roll-out Guide**

Table of Contents

[1 Document objective 5](#_Toc27752520)

[2 MTender CDU Infrastructure 6](#_Toc27752521)

[4 MTender CDU components scheme 7](#_Toc27752522)

[4.1 Apache Cassandra-MD 8](#_Toc27752523)

[4.1.1 md.ac-node1 8](#_Toc27752524)

[4.1.3 md.ac-node2 14](#_Toc27752525)

[4.1.5 md.ac-node3 17](#_Toc27752526)

[4.2 Postres 20](#_Toc27752527)

[4.2.1 Connect 20](#_Toc27752528)

[4.2.2 User management 21](#_Toc27752529)

[4.2.4 Postgres replication / eAuction-3 23](#_Toc27752530)

[4.3 Apache Kafka-MD 25](#_Toc27752531)

[4.3.1 md.ak-node1 25](#_Toc27752532)

[4.3.2 md.ak-node2 27](#_Toc27752533)

[4.3.4 md.ak-node3 30](#_Toc27752534)

[4.3.6 md.kafka-acl 32](#_Toc27752535)

[4.5 BPE-dns-MD 44](#_Toc27752536)

[4.5.1 Connection 44](#_Toc27752537)

[4.5.2 Setup 44](#_Toc27752538)

[4.6 Docker-node1 46](#_Toc27752539)

[4.6.1 Connection 46](#_Toc27752540)

[4.6.2 Install docker 46](#_Toc27752541)

[4.6.3 Install docker CE 46](#_Toc27752542)

[4.8 md.ELK-Stack 47](#_Toc27752543)

[4.8.1 How it works 47](#_Toc27752544)

[4.8.2 Install OpenJDK 47](#_Toc27752545)

[4.8.3 Install Elasticsearch 47](#_Toc27752546)

[4.8.4 Install Kibana 49](#_Toc27752547)

[4.8.5 Install Logstash 49](#_Toc27752548)

[4.8.6 Nginx virtualhost 10.73.10.10 server (single access point) 50](#_Toc27752549)

[4.8.7 Kibana configuration 51](#_Toc27752550)

[4.9 md.Nginx-frontend 54](#_Toc27752551)

[4.9.1 auth.mtender.gov.md 54](#_Toc27752552)

[4.9.2 bpe.mtender.gov.md 54](#_Toc27752553)

[4.9.3 gitlab.mtender.gov.md 55](#_Toc27752554)

[4.9.4 mconnect.mtender.gov.md 55](#_Toc27752555)

[4.9.5 auction.mtender.gov.md 56](#_Toc27752556)

[4.9.6 md.kibana.eprocurement.systems 56](#_Toc27752557)

[4.9.7 zabbix.mtender.gov.md 57](#_Toc27752558)

[4.9.8 operations.mtender.gov.md 57](#_Toc27752559)

[4.9.9 portainer.eprocurement.systems 58](#_Toc27752560)

[4.9.10 public.mtender.gov.md 58](#_Toc27752561)

[4.9.11 storage.mtender.gov.md 59](#_Toc27752562)

[4.11 md.storage-service 60](#_Toc27752563)

[4.11.1 Connection 60](#_Toc27752564)

[4.11.2 Install JRE8 60](#_Toc27752565)

[4.11.3 cqlsh: client to connect to the Cassandra cluster 60](#_Toc27752566)

[4.11.4 Launch script for Docker container 60](#_Toc27752567)

[4.13 md.glusterFS 61](#_Toc27752568)

[4.13.1 Provisional configuration of the server side 61](#_Toc27752569)

[4.13.2 Setup of server side 62](#_Toc27752570)

[4.13.3 Provisional configuration of the client side 63](#_Toc27752571)

[4.13.4 Setup of client side 64](#_Toc27752572)

[4.13.5 Adding / Removing Nodes 64](#_Toc27752573)

[4.15 md.Swarm-Cluster 68](#_Toc27752574)

[4.15.1 Connection 68](#_Toc27752575)

[4.16 md.Zabbix-v3.4 69](#_Toc27752576)

[4.16.1 Connection 69](#_Toc27752577)

[4.16.2 Preparing the environment 69](#_Toc27752578)

[4.16.3 Create a group and user 69](#_Toc27752579)

[4.16.4 Install dependencies 70](#_Toc27752580)

[4.16.5 Creating a Make File 71](#_Toc27752581)

[4.16.6 Making edits 71](#_Toc27752582)

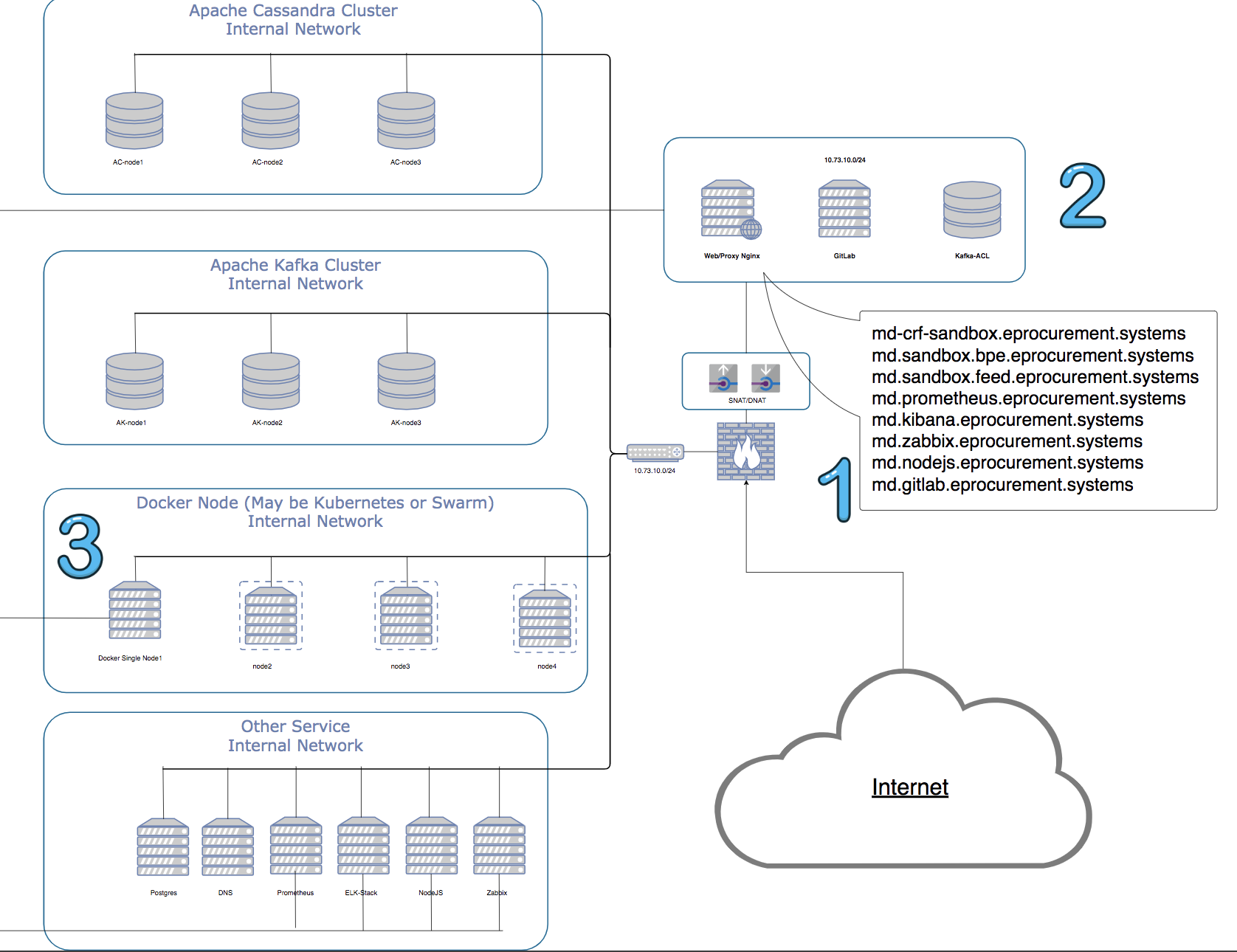
[4.16.7 Install mysql-server-5.7 72](#_Toc27752583)

[4.16.9 Install PHP-7.2 PHP-FPM-7.2 74](#_Toc27752584)

# Document objective

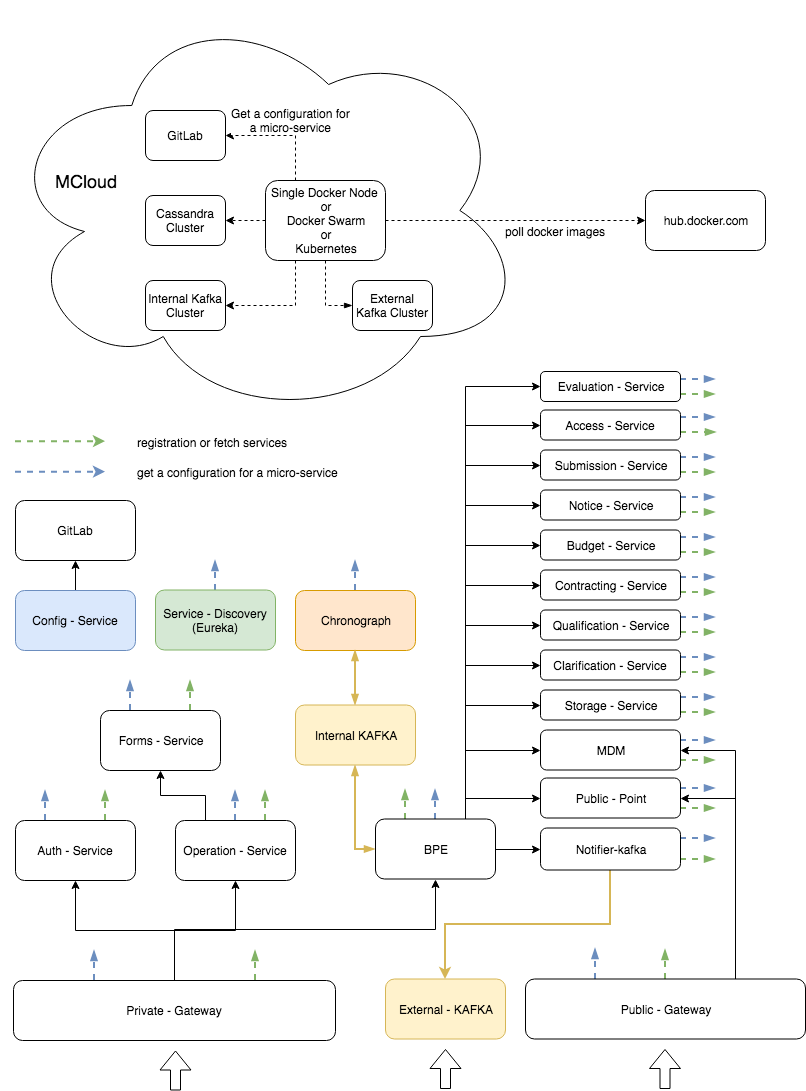
This document is to be used as an Administrator Rollout Guide for the MTender infrastructure. The document describes the current configuration of the production servers.

# MTender CDU Infrastructure

****

# 

# MTender CDU components scheme

[](https://www.draw.io/#G18qpICO2w7nZUNYu301GAK8637EG468Qt)

## Apache Cassandra-MD

md.ac-node1

#### Connect

##### Login to MCloud Production tenant

|  |
| --- |
| ip: 10.73.10.7 login: \*\*\*\*\*\* pass: \*\*\*\*\*\* |

##### Login for Cassandra (root-level)

Use root-level credentials for Cassandra to login

|  |
| --- |
| user: \*\*\*\*\*\* pass: \*\*\*\*\*\*   # cqlsh -u caroot 10.73.10.7 Password: Connected to eProcurement-MD at 10.73.10.7:9042. [cqlsh 5.0.1 | Cassandra 3.9.0 | CQL spec 3.4.2 | Native protocol v4] Use HELP **for** help. caroot@cqlsh> |

##### Login for Cassandra (caclient)

Connect to node-1 using Cassandra caclient credentials

|  |
| --- |
| user: \*\*\*\*\*\* pass: \*\*\*\*\*\*   # cqlsh -u caclient 10.73.10.8 Password: Connected to eProcurement-MD at 10.73.10.8:9042. [cqlsh 5.0.1 | Cassandra 3.9.0 | CQL spec 3.4.2 | Native protocol v4] Use HELP **for** help. |

#### /etc/rc.local

|  |
| --- |
| /usr/sbin/cassandra -R 1>/dev/null |

#### Cassandra cluster setup

###### Setup of UTC time-zone

|  |
| --- |
| # ln -sf /usr/share/zoneinfo/Europe/Kiev /etc/localtime # hwclock --systohc --utc # apt-get install ntpdate -y # ntpdate -bs pool.ntp.org |

###### Setup of JDK

|  |
| --- |
| root@AC-node1:~# apt-get install default-jdk root@AC-node1:~# java -version openjdk version "1.8.0\_151" OpenJDK Runtime Environment (build 1.8.0\_151-8u151-b12-0ubuntu0.16.04.2-b12) OpenJDK 64-Bit Server VM (build 25.151-b12, mixed mode) |

Cassandra - 3.9 latest repository

|  |
| --- |
| root@AC-node1:~# echo "deb http://debian.datastax.com/datastax-ddc 3.9 main" | tee -a /etc/apt/sources.list.d/cassandra.sources.list root@AC-node1:~# apt-get update |

Key registration exception

|  |
| --- |
| W: GPG error: http://debian.datastax.com/datastax-ddc 3.9 Release: The following signatures couldn't be verified beca use the public key is not available: NO\_PUBKEY 350200F2B999A372 W: The repository 'http://debian.datastax.com/datastax-ddc 3.9 Release' is not signed. N: Data from such a repository can't be authenticated and is therefore potentially dangerous to use. N: See apt-secure(8) manpage **for** repository creation and user configuration details.   root@AC-node1:~# apt-key adv --keyserver keys.gnupg.net --recv-keys 350200F2B999A372 root@AC-node1:~# apt-get update Reading package lists... Done |

Setup of curl directly to Cassandra

|  |
| --- |
| root@AC-node1:~# apt-get install curl root@AC-node1:~# apt-get install datastax-ddc |

Status

|  |
| --- |
| root@AC-node1:~# service cassandra status  cassandra.service - LSB: distributed storage system **for** structured data  Loaded: loaded (/etc/init.d/cassandra; bad; vendor preset: enabled)  Active: active (exited) since Tue 2017-12-12 10:54:37 CET; 30s ago |

Exception in /var/log/cassandra/system.log

|  |
| --- |
| ERROR [main] 2017-12-12 10:58:14,034 CassandraDaemon.java:747 - Local host name unknown: java.net.UnknownHostException: AC-node1: AC-node1: Name or service not known |

Solution: add config /etc/hosts имя ac-node1.mtender.gov.md

|  |
| --- |
| root@AC-node1:~# hostname -f root@AC-node1:~# /etc/init.d/cassandra restart |

Cluster status

|  |
| --- |
| root@AC-node1:~# nodetool status Datacenter: datacenter1 ======================= Status=Up/Down |/ State=Normal/Leaving/Joining/Moving -- Address Load Tokens Owns (effective) Host ID Rack UN 127.0.0.1 103.52 KiB 256 100.0% b56bdf9a-e516-4420-84ec-0803e79e345b rack1 |

#### Node configuration and preparation for cluster

1. Stop Cassandra

|  |
| --- |
| root@AC-node1:~# /etc/init.d/cassandra stop Stopping cassandra (via systemctl): cassandra.service.  root@AC-node1:~# ps -ef | grep cassandra |

1. Drop data

|  |
| --- |
| root@AC-node1:~# rm -rf /var/lib/cassandra/\* |

1. Generate tokens for each node: <https://www.geroba.com/cassandra/cassandra-token-calculator/>

|  |
| --- |
| -9223372036854775808 -3074457345618258603 3074457345618258602 |

Use this method for generating tokens when you are not using virtual nodes (vnodes) and using the Murmur3Partitioner (default). This partitioner uses a maximum possible range of hash values from -2 63 to +2 63 -1.

To calculate tokens for this partitioner:

|  |
| --- |
| python -c 'print [str(((2\*\*64 / number\_of\_tokens) \* i) - 2\*\*63) for i in range(number\_of\_tokens)]' For example, to generate tokens **for** 6 nodes:   python -c 'print [str(((2\*\*64 / 6) \* i) - 2\*\*63) for i in range(6)]'   The command displays the token **for** each node: # array of tokens generated |

#### Configurations in /etc/cassandra/cassandra.yaml

Full node-1 config (without comments and empty lines)

|  |
| --- |
| cluster\_name: 'eProcurement-MD' initial\_token: # token hinted\_handoff\_enabled: **true** hinted\_handoff\_throttle\_in\_kb: 1024 max\_hints\_delivery\_threads: 2 hints\_flush\_period\_in\_ms: 10000 max\_hints\_file\_size\_in\_mb: 128 batchlog\_replay\_throttle\_in\_kb: 1024 authenticator: PasswordAuthenticator authorizer: AllowAllAuthorizer role\_manager: CassandraRoleManager roles\_validity\_in\_ms: 2000 permissions\_validity\_in\_ms: 2000 credentials\_validity\_in\_ms: 2000 partitioner: org.apache.cassandra.dht.Murmur3Partitioner data\_file\_directories:  - /var/lib/cassandra/data commitlog\_directory: /var/lib/cassandra/commitlog cdc\_enabled: **false** disk\_failure\_policy: stop commit\_failure\_policy: stop prepared\_statements\_cache\_size\_mb: thrift\_prepared\_statements\_cache\_size\_mb: key\_cache\_size\_in\_mb: key\_cache\_save\_period: 14400 row\_cache\_size\_in\_mb: 0 row\_cache\_save\_period: 0 counter\_cache\_size\_in\_mb: counter\_cache\_save\_period: 7200 saved\_caches\_directory: /var/lib/cassandra/saved\_caches commitlog\_sync: periodic commitlog\_sync\_period\_in\_ms: 10000 commitlog\_segment\_size\_in\_mb: 32 seed\_provider:  - class\_name: org.apache.cassandra.locator.SimpleSeedProvider  parameters:  - seeds: # MCloud Productive tenant ip concurrent\_reads: 32 concurrent\_writes: 32 concurrent\_counter\_writes: 32 concurrent\_materialized\_view\_writes: 32 memtable\_allocation\_type: heap\_buffers index\_summary\_capacity\_in\_mb: index\_summary\_resize\_interval\_in\_minutes: 60 trickle\_fsync: **false** trickle\_fsync\_interval\_in\_kb: 10240 storage\_port: 7000 ssl\_storage\_port: 7001 listen\_address: 10.73.10.7 start\_native\_transport: **true** native\_transport\_port: 9042 start\_rpc: **false** rpc\_interface: eth0 rpc\_port: 9160 rpc\_keepalive: **true** rpc\_server\_type: sync thrift\_framed\_transport\_size\_in\_mb: 15 incremental\_backups: **false** snapshot\_before\_compaction: **false** auto\_snapshot: **true** column\_index\_size\_in\_kb: 64 column\_index\_cache\_size\_in\_kb: 2 compaction\_throughput\_mb\_per\_sec: 16 sstable\_preemptive\_open\_interval\_in\_mb: 50 read\_request\_timeout\_in\_ms: 5000 range\_request\_timeout\_in\_ms: 10000 write\_request\_timeout\_in\_ms: 2000 counter\_write\_request\_timeout\_in\_ms: 5000 cas\_contention\_timeout\_in\_ms: 1000 truncate\_request\_timeout\_in\_ms: 60000 request\_timeout\_in\_ms: 10000 cross\_node\_timeout: **false** endpoint\_snitch: RackInferringSnitch endpoint\_snitch: SimpleSnitch dynamic\_snitch\_update\_interval\_in\_ms: 100 dynamic\_snitch\_reset\_interval\_in\_ms: 600000 dynamic\_snitch\_badness\_threshold: 0.1 request\_scheduler: org.apache.cassandra.scheduler.NoScheduler server\_encryption\_options:  internode\_encryption: none  keystore: conf/.keystore  keystore\_password: cassandra  truststore: conf/.truststore  truststore\_password: cassandra client\_encryption\_options:  enabled: **false**  optional: **false**  keystore: conf/.keystore  keystore\_password: cassandra internode\_compression: dc inter\_dc\_tcp\_nodelay: **false** tracetype\_query\_ttl: 86400 tracetype\_repair\_ttl: 604800 enable\_user\_defined\_functions: **false** enable\_scripted\_user\_defined\_functions: **false** windows\_timer\_interval: 1 transparent\_data\_encryption\_options:  enabled: **false**  chunk\_length\_kb: 64  cipher: AES/CBC/PKCS5Padding  key\_alias: testing:1  key\_provider:  - class\_name: org.apache.cassandra.security.JKSKeyProvider  parameters:  - keystore: conf/.keystore  keystore\_password: cassandra  store\_type: JCEKS  key\_password: cassandra tombstone\_warn\_threshold: 1000 tombstone\_failure\_threshold: 100000 batch\_size\_warn\_threshold\_in\_kb: 5 batch\_size\_fail\_threshold\_in\_kb: 50 unlogged\_batch\_across\_partitions\_warn\_threshold: 10 compaction\_large\_partition\_warning\_threshold\_mb: 100 gc\_warn\_threshold\_in\_ms: 1000 |

To allocate more memory volume:

|  |
| --- |
| vim /etc/cassandra/jvm.options |

Remove comment from the following lines:

|  |
| --- |
| -Xms4G -Xmx4G |

#### Directives

<http://docs.datastax.com/en/cassandra/2.1/cassandra/configuration/configCassandra_yaml_r.html>

md.ac-node2

#### Connect

##### Login to MCloud Production tenant

|  |
| --- |
| ip: 10.73.10.9 login: \*\*\*\*\*\* pass: \*\*\*\*\*\* |

##### Login for Cassandra (root-level)

Use root-level credentials for Cassandra to login

|  |
| --- |
| user: \*\*\*\*\*\* pass: \*\*\*\*\*\*   # cqlsh -u caroot 10.73.10.7 Password: Connected to eProcurement-MD at 10.73.10.7:9042. [cqlsh 5.0.1 | Cassandra 3.9.0 | CQL spec 3.4.2 | Native protocol v4] Use HELP **for** help. caroot@cqlsh> |

##### Login for Cassandra (caclient)

Connect to node-2 using Cassandra caclient credentials

|  |
| --- |
| user: \*\*\*\*\*\* pass: \*\*\*\*\*\*   # cqlsh -u caclient 10.73.10.8 Password: Connected to eProcurement-MD at 10.73.10.8:9042. [cqlsh 5.0.1 | Cassandra 3.9.0 | CQL spec 3.4.2 | Native protocol v4] Use HELP **for** help. |

##### Configurations in /etc/cassandra/cassandra.yaml

Full node-2 config (without comments and empty lines)

|  |
| --- |
| cluster\_name: 'eProcurement-MD' initial\_token: -3074457345618258603 hinted\_handoff\_enabled: **true** hinted\_handoff\_throttle\_in\_kb: 1024 max\_hints\_delivery\_threads: 2 hints\_flush\_period\_in\_ms: 10000 max\_hints\_file\_size\_in\_mb: 128 batchlog\_replay\_throttle\_in\_kb: 1024 authenticator: PasswordAuthenticator authorizer: AllowAllAuthorizer role\_manager: CassandraRoleManager roles\_validity\_in\_ms: 2000 permissions\_validity\_in\_ms: 2000 credentials\_validity\_in\_ms: 2000 partitioner: org.apache.cassandra.dht.Murmur3Partitioner data\_file\_directories:  - /var/lib/cassandra/data commitlog\_directory: /var/lib/cassandra/commitlog cdc\_enabled: **false** disk\_failure\_policy: stop commit\_failure\_policy: stop prepared\_statements\_cache\_size\_mb: thrift\_prepared\_statements\_cache\_size\_mb: key\_cache\_size\_in\_mb: key\_cache\_save\_period: 14400 row\_cache\_size\_in\_mb: 0 row\_cache\_save\_period: 0 counter\_cache\_size\_in\_mb: counter\_cache\_save\_period: 7200 saved\_caches\_directory: /var/lib/cassandra/saved\_caches commitlog\_sync: periodic commitlog\_sync\_period\_in\_ms: 10000 commitlog\_segment\_size\_in\_mb: 32 seed\_provider:  - class\_name: org.apache.cassandra.locator.SimpleSeedProvider  parameters:  - seeds: "10.73.10.7" concurrent\_reads: 32 concurrent\_writes: 32 concurrent\_counter\_writes: 32 concurrent\_materialized\_view\_writes: 32 memtable\_allocation\_type: heap\_buffers index\_summary\_capacity\_in\_mb: index\_summary\_resize\_interval\_in\_minutes: 60 trickle\_fsync: **false** trickle\_fsync\_interval\_in\_kb: 10240 storage\_port: 7000 ssl\_storage\_port: 7001 listen\_address: 10.73.10.9 start\_native\_transport: **true** native\_transport\_port: 9042 start\_rpc: **false** rpc\_interface: eth0 rpc\_port: 9160 rpc\_keepalive: **true** rpc\_server\_type: sync thrift\_framed\_transport\_size\_in\_mb: 15 incremental\_backups: **false** snapshot\_before\_compaction: **false** auto\_snapshot: **true** column\_index\_size\_in\_kb: 64 column\_index\_cache\_size\_in\_kb: 2 compaction\_throughput\_mb\_per\_sec: 16 sstable\_preemptive\_open\_interval\_in\_mb: 50 read\_request\_timeout\_in\_ms: 5000 range\_request\_timeout\_in\_ms: 10000 write\_request\_timeout\_in\_ms: 2000 counter\_write\_request\_timeout\_in\_ms: 5000 cas\_contention\_timeout\_in\_ms: 1000 truncate\_request\_timeout\_in\_ms: 60000 request\_timeout\_in\_ms: 10000 cross\_node\_timeout: **false** endpoint\_snitch: RackInferringSnitch dynamic\_snitch\_update\_interval\_in\_ms: 100 dynamic\_snitch\_reset\_interval\_in\_ms: 600000 dynamic\_snitch\_badness\_threshold: 0.1 request\_scheduler: org.apache.cassandra.scheduler.NoScheduler server\_encryption\_options:  internode\_encryption: none  keystore: conf/.keystore  keystore\_password: cassandra  truststore: conf/.truststore  truststore\_password: cassandra client\_encryption\_options:  enabled: **false**  optional: **false**  keystore: conf/.keystore  keystore\_password: cassandra internode\_compression: dc inter\_dc\_tcp\_nodelay: **false** tracetype\_query\_ttl: 86400 tracetype\_repair\_ttl: 604800 enable\_user\_defined\_functions: **false** enable\_scripted\_user\_defined\_functions: **false** windows\_timer\_interval: 1 transparent\_data\_encryption\_options:  enabled: **false**  chunk\_length\_kb: 64  cipher: AES/CBC/PKCS5Padding  key\_alias: testing:1  key\_provider:  - class\_name: org.apache.cassandra.security.JKSKeyProvider  parameters:  - keystore: conf/.keystore  keystore\_password: cassandra  store\_type: JCEKS  key\_password: cassandra tombstone\_warn\_threshold: 1000 tombstone\_failure\_threshold: 100000 batch\_size\_warn\_threshold\_in\_kb: 5 batch\_size\_fail\_threshold\_in\_kb: 50 unlogged\_batch\_across\_partitions\_warn\_threshold: 10 compaction\_large\_partition\_warning\_threshold\_mb: 100 gc\_warn\_threshold\_in\_ms: 1000 |

To allocate more memory volume:

|  |
| --- |
| vim /etc/cassandra/jvm.options |

Remove comment from the following lines:

|  |
| --- |
| -Xms4G -Xmx4G |

md.ac-node3

#### Connect

##### Login to MCloud Production tenant

|  |
| --- |
| ssh ip: 10.73.10.8 login: \*\*\*\*\*\* pass: \*\*\*\*\*\* |

##### Login for Cassandra (root-level)

Use root-level credentials for Cassandra to login

|  |
| --- |
| user: \*\*\*\*\*\* pass: \*\*\*\*\*\*   # cqlsh -u caroot 10.73.10.8 Password: Connected to eProcurement-MD at 10.73.10.7:9042. [cqlsh 5.0.1 | Cassandra 3.9.0 | CQL spec 3.4.2 | Native protocol v4] Use HELP **for** help. caroot@cqlsh> |

##### Login for Cassandra (caclient)

Connect to node-3 using Cassandra caclient credentials

|  |
| --- |
| user: \*\*\*\*\*\* pass: \*\*\*\*\*\*  # cqlsh -u caroot 10.73.10.8  Password: Connected to eProcurement-MD at 10.73.10.8:9042. [cqlsh 5.0.1 | Cassandra 3.9.0 | CQL spec 3.4.2 | Native protocol v4] Use HELP **for** help. |

#### /etc/rc.local

|  |
| --- |
| /usr/sbin/cassandra -R 1>/dev/null |

#### Configuration in /etc/cassandra/cassandra.yaml

Full node-3 config (without comments and empty lines)

|  |
| --- |
| cluster\_name: 'eProcurement-MD' initial\_token: 3074457345618258602 hinted\_handoff\_enabled: **true** hinted\_handoff\_throttle\_in\_kb: 1024 max\_hints\_delivery\_threads: 2 hints\_flush\_period\_in\_ms: 10000 max\_hints\_file\_size\_in\_mb: 128 batchlog\_replay\_throttle\_in\_kb: 1024 authenticator: PasswordAuthenticator authorizer: AllowAllAuthorizer role\_manager: CassandraRoleManager roles\_validity\_in\_ms: 2000 permissions\_validity\_in\_ms: 2000 credentials\_validity\_in\_ms: 2000 partitioner: org.apache.cassandra.dht.Murmur3Partitioner data\_file\_directories:  - /var/lib/cassandra/data commitlog\_directory: /var/lib/cassandra/commitlog cdc\_enabled: **false** disk\_failure\_policy: stop commit\_failure\_policy: stop prepared\_statements\_cache\_size\_mb: thrift\_prepared\_statements\_cache\_size\_mb: key\_cache\_size\_in\_mb: key\_cache\_save\_period: 14400 row\_cache\_size\_in\_mb: 0 row\_cache\_save\_period: 0 counter\_cache\_size\_in\_mb: counter\_cache\_save\_period: 7200 saved\_caches\_directory: /var/lib/cassandra/saved\_caches commitlog\_sync: periodic commitlog\_sync\_period\_in\_ms: 10000 commitlog\_segment\_size\_in\_mb: 32 seed\_provider:  - class\_name: org.apache.cassandra.locator.SimpleSeedProvider  parameters:  - seeds: "10.73.10.7" concurrent\_reads: 32 concurrent\_writes: 32 concurrent\_counter\_writes: 32 concurrent\_materialized\_view\_writes: 32 memtable\_allocation\_type: heap\_buffers index\_summary\_capacity\_in\_mb: index\_summary\_resize\_interval\_in\_minutes: 60 trickle\_fsync: **false** trickle\_fsync\_interval\_in\_kb: 10240 storage\_port: 7000 ssl\_storage\_port: 7001 listen\_address: 10.73.10.8 start\_native\_transport: **true** native\_transport\_port: 9042 start\_rpc: **false** rpc\_interface: eth0 rpc\_port: 9160 rpc\_keepalive: **true** rpc\_server\_type: sync thrift\_framed\_transport\_size\_in\_mb: 15 incremental\_backups: **false** snapshot\_before\_compaction: **false** auto\_snapshot: **true** column\_index\_size\_in\_kb: 64 column\_index\_cache\_size\_in\_kb: 2 compaction\_throughput\_mb\_per\_sec: 16 sstable\_preemptive\_open\_interval\_in\_mb: 50 read\_request\_timeout\_in\_ms: 5000 range\_request\_timeout\_in\_ms: 10000 write\_request\_timeout\_in\_ms: 2000 counter\_write\_request\_timeout\_in\_ms: 5000 cas\_contention\_timeout\_in\_ms: 1000 truncate\_request\_timeout\_in\_ms: 60000 request\_timeout\_in\_ms: 10000 cross\_node\_timeout: **false** endpoint\_snitch: RackInferringSnitch dynamic\_snitch\_update\_interval\_in\_ms: 100 dynamic\_snitch\_reset\_interval\_in\_ms: 600000 dynamic\_snitch\_badness\_threshold: 0.1 request\_scheduler: org.apache.cassandra.scheduler.NoScheduler server\_encryption\_options:  internode\_encryption: none  keystore: conf/.keystore  keystore\_password: cassandra  truststore: conf/.truststore  truststore\_password: cassandra client\_encryption\_options:  enabled: **false**  optional: **false**  keystore: conf/.keystore  keystore\_password: cassandra internode\_compression: dc inter\_dc\_tcp\_nodelay: **false** tracetype\_query\_ttl: 86400 tracetype\_repair\_ttl: 604800 enable\_user\_defined\_functions: **false** enable\_scripted\_user\_defined\_functions: **false** windows\_timer\_interval: 1 transparent\_data\_encryption\_options:  enabled: **false**  chunk\_length\_kb: 64  cipher: AES/CBC/PKCS5Padding  key\_alias: testing:1  key\_provider:  - class\_name: org.apache.cassandra.security.JKSKeyProvider  parameters:  - keystore: conf/.keystore  keystore\_password: cassandra  store\_type: JCEKS  key\_password: cassandra tombstone\_warn\_threshold: 1000 tombstone\_failure\_threshold: 100000 batch\_size\_warn\_threshold\_in\_kb: 5 batch\_size\_fail\_threshold\_in\_kb: 50 unlogged\_batch\_across\_partitions\_warn\_threshold: 10 compaction\_large\_partition\_warning\_threshold\_mb: 100 gc\_warn\_threshold\_in\_ms: 1000 |

To allocate more memory volume:

|  |
| --- |
| vim /etc/cassandra/jvm.options |

Remove comment from the following lines:

|  |
| --- |
| -Xms4G -Xmx4G |

## Postres

Connect

|  |
| --- |
| ip: 10.73.10.5 ssh login: \*\*\*\*\*\* pass: \*\*\*\*\*\*   Postgres: login: \*\*\*\*\*\* pass: \*\*\*\*\*\*   root@postgres:~# psql --host localhost -Upgsql -p5432 --dbname template1 apt-get update add-apt-repository "deb http://apt.postgresql.org/pub/repos/apt/ $(lsb\_release -sc)-pgdg main" wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add - apt-get update apt-get install postgresql-9.6 |

Configure the server to work on the network, otherwise it will be available only from the address 127.0.0.1. Open the *postgresql.conf* file and edit it

|  |
| --- |
| netstat -nltp | grep post tcp 0 0 127.0.0.1:5432 0.0.0.0:\* LISTEN 3138/postgres   vim /etc/postgresql/9.6/main/postgresql.conf #listen\_addresses = 'localhost' # what IP address(es) to listen on; listen\_addresses = '\*' # what IP address(es) to listen on;   /etc/init.d/postgresql restart [ ok ] Restarting postgresql (via systemctl): postgresql.service.   netstat -nltp | grep post tcp 0 0 0.0.0.0:5432 0.0.0.0:\* LISTEN 62712/postgres |

Define who should have access to the server. This is configured in the *pg\_hba.conf* file

|  |
| --- |
| vim /etc/postgresql/9.6/main/pg\_hba.conf # IPv4 local connections: host all all 127.0.0.1/32 md5 host all postgres 10.73.10.0/24 md5 ### Access for Docker Swarm login: authadmin host all authadmin 10.73.10.6/32 md5 host all authadmin 10.73.10.13/32 md5 host all authadmin 10.73.10.19/32 md5   ### Access for Docker Swarm login: chronoadmin host all chronoadmin 10.73.10.6/32 md5 host all chronoadmin 10.73.10.13/32 md5 host all chronoadmin 10.73.10.19/32 md5 ### Access for Docker Swarm login: mdmadmin host all mdmadmin 10.73.10.6/32 md5 host all mdmadmin 10.73.10.13/32 md5 host all mdmadmin 10.73.10.19/32 md5 ### Access for Docker Swarm login: bpeadmin host all bpeadmin 10.73.10.6/32 md5 host all bpeadmin 10.73.10.13/32 md5 host all bpeadmin 10.73.10.19/32 md5 ### Access for Slave-server. Login: replicator host replication replicator 10.73.10.15/32 md5 |

User management

#### Creating a role and assigning root privileges

|  |
| --- |
| su postgres -c psql postgres psql (9.6.10) Type "help" **for** help.   postgres=# CREATE ROLE pgsql WITH SUPERUSER LOGIN PASSWORD 'E8\_kmdUV\_2dtEsX'; CREATE ROLE postgres=# \q |

#### Password change for user pgsql (aka root)

In the *pg\_hba.conf* file, add the line

|  |
| --- |
| vim /etc/postgresql/9.6/main/pg\_hba.conf   # TYPE DATABASE USER ADDRESS METHOD   # "local" is for Unix domain socket connections only #local all all peer local all pgsql trust |

Next, restart postgres and then run the password change request:

Attention: The request will work only if the necessary role is created and there are privileges for base on template1

|  |
| --- |
| psql -Upgsql template1 -c "alter user pgsql with password 'E8\_kmdUV\_2dtEsX'; |

Check connection

|  |
| --- |
| psql --host localhost -Upgsql -p5432 --dbname template1 Password **for** user pgsql: psql (9.6.10) SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off) Type "help" **for** help.   template1=# |

Now go in */etc/postgresql/9.4/main/pg\_hba.conf* and return the changes to the previous form and restart postgres

Postgres replication / eAuction-3

##### Connection

|  |
| --- |
| ssh ip: 10.73.10.15 login: \*\*\*\*\*\*\*\*\*\* pass: \*\*\*\*\*\*\*\*\*\* |

##### Master setting

when setting up slave, you will need to write the same parameters:

|  |
| --- |
| vi /etc/postgresql/9.6/main/postgresql.conf wal\_level = replica max\_wal\_senders = 10 wal\_keep\_segments = 128 |

The value of max\_wal\_senders cannot exceed the value of max\_connections.

###### Create a user to replicate

|  |
| --- |
| su postgres -c psql postgres postgres=# CREATE ROLE replicator WITH REPLICATION PASSWORD 'D6K7jxW36qrpJ7xs' LOGIN; postgres=# \q |

###### Open access to the server to the user replicator co slave server

|  |
| --- |
| ### Access for Slave-server. Login: replicator host replication replicator 10.73.10.15/32 md5 |

Where 10.73.10.15/32 is the address of the slave server. / 32 indicates that access should be granted only for this address. Please note that "DATABASE" is specified as "replication". This is not a database name, but a system value.

###### Restart master

|  |
| --- |
| /etc/init.d/postgresql restart [ ok ] Restarting postgresql (via systemctl): postgresql.service. |

##### Slave setting

We stop the server and correct the configuration:

|  |
| --- |
| /etc/init.d/postgresql stop [ ok ] Restarting postgresql (via systemctl): postgresql.service. vi /etc/postgresql/9.6/main/postgresql.conf hot\_standby = on |

We also make the configuration the same as on master, which is marked as necessary for slave. We check that this is the slave server, and clear the data\_directory (if you are not sure, it is better to transfer the contents of this directory to another location and delete it later):

|  |
| --- |
| rm -rf /var/lib/postgresql/9.6/main/\* |

Copy the current state from master to slave and enter the password of the replicator user (the command is executed on the slave server):

|  |
| --- |
| su postgres -c "pg\_basebackup -h 10.73.10.5 -D /var/lib/postgresql/9.6/main -R -P -U replicator --xlog-method=stream" |

Where 10.73.10.5 - master address. Add the ability to the slave server to become a master (necessary to increase fault tolerance):

|  |
| --- |
| vi /var/lib/postgresql/9.6/main/recovery.conf   trigger\_file = '/var/lib/postgresql/9.6/main/make\_me\_master' |

Under normal conditions, this file should not be. It is created manually when the master server is unavailable, so that the slave becomes a master, and allows recording to the database. After the master is restored, there will be data out of sync on the servers. A solution to this problem will be written below.

###### Start slave

|  |
| --- |
| /etc/init.d/postgresql start [ ok ] Starting postgresql (via systemctl): postgresql.service. |

You can check if replication is working by running the following commands at slave:

|  |
| --- |
| ps aux | grep receiver postgres 4264 0.0 0.2 315292 11852 ? Ss 16:42 0:00 postgres: 9.6/main: wal receiver process streaming 0/1C2DD380 |

and at master:

|  |
| --- |
| ps aux | grep sender postgres 3559 0.0 1.1 304872 11064 ? Ss 16:42 0:00 postgres: 9.6/main: wal sender process replicator 10.73.10.15(51174) streaming 0/1C2DD380 |

If the master server crashes on slave, you need to do:

|  |
| --- |
| touch /var/lib/postgresql/9.6/main/make\_me\_master |

Slave will automatically become master, without restarting PostgreSQL, and the system will continue to work.

In case of recovery of the master's functionality,

1. Stop the service (/etc/init.d/postgresql stop);
2. Repeat the above steps, namely:
   1. add user replicator to slave;
   2. fix pg\_hba.conf configuration to slave;
   3. restart slave;
   4. clear data\_directory to master;
   5. copy the current state from slave;
3. stop slave;
4. rename recovery.done to recovery.conf (mv /var/lib/postgresql/9.6/main/recovery.done /var/lib/postgresql/9.6/main/recovery.conf);
5. run master;
6. run slave;
7. check that master is master, and slave is slave.

## Apache Kafka-MD

md.ak-node1

#### Connect

##### Login to MCloud Production tenant using md.ak-node1 credentials

|  |
| --- |
| ip: 10.73.10.12 login: \*\*\*\*\*\* pass: \*\*\*\*\*\* |

##### Setup of UTC time-zone

|  |
| --- |
| # ln -sf /usr/share/zoneinfo/Europe/Chisinau /etc/localtime # hwclock --systohc --utc # apt-get install ntpdate -y # ntpdate -bs pool.ntp.org |

##### /etc/rc.local

|  |
| --- |
| cd /home/kafka && nohup bin/zookeeper-server-start.sh config/zookeeper.properties & #cd /home/kafka && nohup bin/kafka-server-sasl-start.sh config/server-sasl.properties & cd /home/kafka && nohup bin/kafka-server-start.sh config/server.properties & |

Denote host files on test machines:

##### /etc/hosts

|  |
| --- |
| 127.0.0.1 localhost 10.73.10.12 ak-node1.mtender.gov.md 10.73.10.11 ak-node2.mtender.gov.md 10.73.10.24 ak-node3.mtender.gov.md   # The following lines are desirable for IPv6 capable hosts ::1 localhost ip6-localhost ip6-loopback ff02::1 ip6-allnodes ff02::2 ip6-allrouters |

#### Setup

##### Java Development Kit:

|  |
| --- |
| # apt-get install default-jdk |

##### Kafka

|  |
| --- |
| # wget http://apache.volia.net/kafka/1.0.0/kafka\_2.12-1.0.0.tgz # tar xfv kafka\_2.12-1.0.0.tgz |

##### Components setup

###### Zookeeper

|  |
| --- |
| root@static:/home/kafka# mkdir -p /var/zookeeper/data root@static:/home/kafka# vim config/zookeeper.properties dataDir=/var/zookeeper/data # the port at which the clients will connect clientPort=2181 # disable the per-ip limit on the number of connections since this is a non-production config maxClientCnxns=0 server.1=ak-node1.mtender.gov.md:2888:3888 server.2=ak-node2.mtender.gov.md:2888:3888 server.3=ak-node3.mtender.gov.md:2888:3888 initLimit=5 syncLimit=2 |

Create a unique identifier on all machines. In our case, the first node is described, we execute on it:

|  |
| --- |
| root@AK-node1:/home/kafka# echo "01" > /var/zookeeper/data/myid |

###### Kafka

|  |
| --- |
| root@AK-node1:/home/kafka# vim config/server.properties   broker.id=01 log.dirs=/var/log/kafka-logs zookeeper.connect=ak-node1.mtender.gov.md:2181,ak-node2.mtender.gov.md:2181,ak-node3.mtender.gov.md:2181 |

Run zookeeper on all nodes:

|  |
| --- |
| root@AK-node1:/home/kafka# nohup bin/zookeeper-server-start.sh config/zookeeper.properties & |

Run the broker on all nodes:

|  |
| --- |
| root@AK-node1:/home/kafka# nohup bin/kafka-server-start.sh config/server.properties & |

Create Kafka topic:

|  |
| --- |
| root@AK-node1:/home/kafka# bin/kafka-topics.sh \ --create \ --zookeeper ak-node1.mtender.gov.md:2181,ak-node2.mtender.gov.md:2181,ak-node3.mtender.gov.md:2181 \ --replication-factor 3 \ --partitions 1 \ --topic uStudioTopTest   Created topic "uStudioTopTest". |

Run producer on the first node, accept the message on the third node

|  |
| --- |
| root@AK-node1:/home/kafka# bin/kafka-console-producer.sh \ --broker-list ak-node1.mtender.gov.md:9092,ak-node2.mtender.gov.md:9092,ak-node3.mtender.gov.md:9092 \ --topic uStudioTopTest >hello, my Kafka! |

On all nodes, add to /etc/rc.local

|  |
| --- |
| cd /home/kafka && nohup bin/zookeeper-server-start.sh config/zookeeper.properties & cd /home/kafka && nohup bin/kafka-server-start.sh config/server.properties & |

**In order for kafka-server to start, you need to have zookeeper running on at least two nodes. Therefore, if the entire cluster is physically turned off at the same time, we start manually zookeeper, and then manually kafka-server**

md.ak-node2

##### Login to MCloud Production tenant using *md.ak-node2* credentials

|  |
| --- |
| ip: 10.73.10.11 login: \*\*\*\*\*\* pass: \*\*\*\*\*\* |

##### /etc/rc.local

|  |
| --- |
| cd /home/kafka && nohup bin/zookeeper-server-start.sh config/zookeeper.properties & #cd /home/kafka && nohup bin/kafka-server-sasl-start.sh config/server-sasl.properties & cd /home/kafka && nohup bin/kafka-server-start.sh config/server.properties & |

Denote host files on test machines:

##### /etc/hosts

|  |
| --- |
| 127.0.0.1 localhost 10.73.10.12 ak-node1.mtender.gov.md 10.73.10.11 ak-node2.mtender.gov.md 10.73.10.24 ak-node3.mtender.gov.md   # The following lines are desirable for IPv6 capable hosts ::1 localhost ip6-localhost ip6-loopback ff02::1 ip6-allnodes ff02::2 ip6-allrouters |

##### Setup of UTC time-zone

|  |
| --- |
| # ln -sf /usr/share/zoneinfo/Europe/Chisinau /etc/localtime # hwclock --systohc --utc # apt-get install ntpdate -y # ntpdate -bs pool.ntp.org |

#### Setup

##### Java Development Kit:

|  |
| --- |
| # apt-get install default-jdk |

##### Kafka

|  |
| --- |
| # wget http://apache.volia.net/kafka/1.0.0/kafka\_2.12-1.0.0.tgz # tar xfv kafka\_2.12-1.0.0.tgz |

##### Components setup

###### Zookeeper

|  |
| --- |
| root@static:/home/kafka# mkdir -p /var/zookeeper/data root@static:/home/kafka# vim config/zookeeper.properties dataDir=/var/zookeeper/data # the port at which the clients will connect clientPort=2181 # disable the per-ip limit on the number of connections since this is a non-production config maxClientCnxns=0 server.1=ak-node1.mtender.gov.md:2888:3888 server.2=ak-node2.mtender.gov.md:2888:3888 server.3=ak-node3.mtender.gov.md:2888:3888 initLimit=5 syncLimit=2 |

Create a unique identifier on all machines. In our case, the first node is described, we execute on it:

|  |
| --- |
| root@AK-node1:/home/kafka# echo "01" > /var/zookeeper/data/myid |

###### Kafka

|  |
| --- |
| root@AK-node1:/home/kafka# vim config/server.properties   broker.id=02 log.dirs=/var/log/kafka-logs zookeeper.connect=ak-node1.mtender.gov.md:2181,ak-node2.mtender.gov.md:2181,ak-node3.mtender.gov.md:2181 |

Run zookeeper on all nodes:

|  |
| --- |
| root@AK-node1:/home/kafka# nohup bin/zookeeper-server-start.sh config/zookeeper.properties & |

Run the broker on all nodes:

|  |
| --- |
| root@AK-node1:/home/kafka# nohup bin/kafka-server-start.sh config/server.properties & |

md.ak-node3

##### Login to MCloud Production tenant using *md.ak-node3* credentials

|  |
| --- |
| ip: 10.73.10.24 login: \*\*\*\*\*\* pass: \*\*\*\*\*\* |

##### /etc/rc.local

|  |
| --- |
| cd /home/kafka && nohup bin/zookeeper-server-start.sh config/zookeeper.properties & #cd /home/kafka && nohup bin/kafka-server-sasl-start.sh config/server-sasl.properties & cd /home/kafka && nohup bin/kafka-server-start.sh config/server.properties & |

Denote host files on test machines:

##### /etc/hosts

|  |
| --- |
| 127.0.0.1 localhost 10.73.10.12 ak-node1.mtender.gov.md 10.73.10.11 ak-node2.mtender.gov.md 10.73.10.24 ak-node3.mtender.gov.md   # The following lines are desirable for IPv6 capable hosts ::1 localhost ip6-localhost ip6-loopback ff02::1 ip6-allnodes ff02::2 ip6-allrouters |

##### Setup of UTC time-zone

|  |
| --- |
| # ln -sf /usr/share/zoneinfo/Europe/Chisinau /etc/localtime # hwclock --systohc --utc # apt-get install ntpdate -y # ntpdate -bs pool.ntp.org |

#### Setup

##### Java Development Kit:

|  |
| --- |
| # apt-get install default-jdk |

##### Kafka

|  |
| --- |
| # wget http://apache.volia.net/kafka/1.0.0/kafka\_2.12-1.0.0.tgz # tar xfv kafka\_2.12-1.0.0.tgz |

##### Components setup

###### Zookeeper

|  |
| --- |
| root@static:/home/kafka# mkdir -p /var/zookeeper/data root@static:/home/kafka# vim config/zookeeper.properties dataDir=/var/zookeeper/data # the port at which the clients will connect clientPort=2181 # disable the per-ip limit on the number of connections since this is a non-production config maxClientCnxns=0 server.1=ak-node1.mtender.gov.md:2888:3888 server.2=ak-node2.mtender.gov.md:2888:3888 server.3=ak-node3.mtender.gov.md:2888:3888 initLimit=5 syncLimit=2 |

Create a unique identifier on all machines. In our case, the first node is described, we execute on it:

|  |
| --- |
| root@AK-node1:/home/kafka# echo "01" > /var/zookeeper/data/myid |

###### Kafka

|  |
| --- |
| root@AK-node1:/home/kafka# vim config/server.properties   broker.id=03 log.dirs=/var/log/kafka-logs zookeeper.connect=ak-node1.mtender.gov.md:2181,ak-node2.mtender.gov.md:2181,ak-node3.mtender.gov.md:2181 |

Run zookeeper on all nodes:

|  |
| --- |
| root@AK-node1:/home/kafka# nohup bin/zookeeper-server-start.sh config/zookeeper.properties & |

Run the broker on all nodes:

|  |
| --- |
| root@AK-node1:/home/kafka# nohup bin/kafka-server-start.sh config/server.properties & |

md.kafka-acl

##### Login to MCloud Production tenant

|  |
| --- |
| ip: 10.73.10.21 login: tenantadmin pass: uPuAqPUa8w |

##### Kafka user credentials

Use Kafka user credentials

##### /etc/rc.local

|  |
| --- |
| cd /home/kafka && nohup bin/zookeeper-server-sasl-start.sh config/zookeeper-sasl.properties & cd /home/kafka && nohup bin/kafka-server-sasl-start.sh config/server-sasl.properties & |

##### Setup of UTC time-zone

|  |
| --- |
| # ln -sf /usr/share/zoneinfo/Europe/Chisinau /etc/localtime # hwclock --systohc --utc # apt-get install ntpdate -y # ntpdate -bs pool.ntp.org |

##### Setup

###### Java Development Kit

|  |
| --- |
| # apt-get install default-jdk |

###### Kafka

|  |
| --- |
| # wget http://apache.volia.net/kafka/1.0.0/kafka\_2.12-1.0.0.tgz # tar xfv kafka\_2.12-1.0.0.tgz |

##### Configuration

###### zookeeper: config/zookeeper-sasl.properties

|  |
| --- |
| dataDir=/var/zookeeper/data clientPort=2181 maxClientCnxns=0 authProvider.1=org.apache.zookeeper.server.auth.SASLAuthenticationProvider requireClientAuthScheme=sasl jaasLoginRenew=3600000 |

###### consumer: config/consumer-sasl.properties

|  |
| --- |
| security.protocol=SASL\_PLAINTEXT sasl.mechanism=PLAIN #zookeeper.connect=localhost:2181 #zookeeper.connection.timeout.ms=6000 #group.id=test-consumer-group group.id=console-consumer-1262 |

###### config/producer-sasl.properties

|  |
| --- |
| security.protocol=SASL\_PLAINTEXT sasl.mechanism=PLAIN bootstrap.servers=localhost:9092 #compression.type=none |

###### config/server-sasl.properties

|  |
| --- |
| security.inter.broker.protocol=SASL\_PLAINTEXT sasl.mechanism.inter.broker.protocol=PLAIN sasl.enabled.mechanisms=PLAIN   authorizer.class.name=kafka.security.auth.SimpleAclAuthorizer allow.everyone.if.no.acl.found=**false** auto.create.topics.enable=**false** broker.id=0 listeners=SASL\_PLAINTEXT://:9092 advertised.listeners=SASL\_PLAINTEXT://feed.mtender.gov.md:9092 #advertised.listeners=SASL\_PLAINTEXT://md.sandbox.feed.eprocurement.systems:9092 num.network.threads=3 num.io.threads=8 socket.send.buffer.bytes=102400 socket.receive.buffer.bytes=102400 socket.request.max.bytes=104857600 #advertised.host.name=md.sandbox.feed.eprocurement.systems advertised.host.name=feed.mtender.gov.md num.partitions=1 num.recovery.threads.per.data.dir=1 log.flush.interval.messages=30000000 log.flush.interval.ms=1800000 log.retention.minutes=30 log.segment.bytes=1073741824 log.retention.check.interval.ms=300000 delete.topic.enable=**true** zookeeper.connect=localhost:2181 zookeeper.connection.timeout.ms=6000 offsets.topic.replication.factor=1 super.users=User:admin |

###### config/connect-standalone.properties

|  |
| --- |
| bootstrap.servers=feed.mtender.gov.md:9092 key.converter=org.apache.kafka.connect.json.JsonConverter value.converter=org.apache.kafka.connect.json.JsonConverter key.converter.schemas.enable=**true** value.converter.schemas.enable=**true**  internal.key.converter=org.apache.kafka.connect.json.JsonConverter internal.value.converter=org.apache.kafka.connect.json.JsonConverter internal.key.converter.schemas.enable=**false** internal.value.converter.schemas.enable=**false** offset.storage.file.filename=/tmp/connect.offsets offset.flush.interval.ms=10000 |

###### config/jaas-kafka-client.conf

|  |
| --- |
| KafkaClient {  org.apache.kafka.common.security.plain.PlainLoginModule required  username="\*\*\*\*\*\*"  password="\*\*\*\*\*\*"; };   Client {  org.apache.kafka.common.security.plain.PlainLoginModule required  username="\*\*\*\*\*\*"  password="\*\*\*\*\*\*"; }; |

###### config/jaas-kafka-server.conf

|  |
| --- |
| KafkaServer {  org.apache.kafka.common.security.plain.PlainLoginModule required  username="\*\*\*\*\*\*"  password="\*\*\*\*\*\*"  user\_admin="\*\*\*\*\*\*"  user\_7e75b11a-5538-43b2-9a75-4155eb597589="\*\*\*\*\*\*"  user\_17c1c77c-62a8-4499-9b45-e49e285e6ffe="\*\*\*\*\*\*"  user\_3b0e5206-bb50-470e-8dc5-d4de5e1c86f7="\*\*\*\*\*\*"  user\_6ac7e396-35b7-42a1-9c5e-a8f7349d5c5f="\*\*\*\*\*\*"  user\_2cd423a4-d1a0-4df8-9a0d-1c24bc199e66="\*\*\*\*\*\*"  user\_67bf521c-7e83-4f31-adad-962347ee1495="\*\*\*\*\*\*"  user\_YodaAdmin="\*\*\*\*\*\*"; };   Client {  org.apache.kafka.common.security.plain.PlainLoginModule required  username="\*\*\*\*\*\*"  password="\*\*\*\*\*\*"; }; |

###### config/jaas-zookeeper.conf

|  |
| --- |
| Server {  org.apache.kafka.common.security.plain.PlainLoginModule required  username="\*\*\*\*\*\*"  password="\*\*\*\*\*\*"  user\_admin="\*\*\*\*\*\*"  user\_7e75b11a-5538-43b2-9a75-4155eb597589="\*\*\*\*\*\*"  user\_17c1c77c-62a8-4499-9b45-e49e285e6ffe="\*\*\*\*\*\*"  user\_3b0e5206-bb50-470e-8dc5-d4de5e1c86f7="\*\*\*\*\*\*"  user\_6ac7e396-35b7-42a1-9c5e-a8f7349d5c5f="\*\*\*\*\*\*"  user\_2cd423a4-d1a0-4df8-9a0d-1c24bc199e66="\*\*\*\*\*\*"  user\_67bf521c-7e83-4f31-adad-962347ee1495="\*\*\*\*\*\*"  user\_YodaAdmin="\*\*\*\*\*\*"; }; |

###### bin/zookeeper-server-sasl-start.sh

|  |
| --- |
| #!/bin/bash   **if** [ $# -lt 1 ]; **then**  echo "USAGE: $0 [-daemon] zookeeper.properties"  exit 1 **fi** base\_dir=$(dirname $0)   **if** [ "x$KAFKA\_LOG4J\_OPTS" = "x" ]; **then**  export KAFKA\_LOG4J\_OPTS="-Dlog4j.configuration=file:$base\_dir/../config/log4j.properties" **fi**   **if** [ "x$KAFKA\_HEAP\_OPTS" = "x" ]; **then**  export KAFKA\_HEAP\_OPTS="-Xmx512M -Xms512M" **fi**   EXTRA\_ARGS=${EXTRA\_ARGS-'-name zookeeper -loggc'}   COMMAND=$1 **case** $COMMAND **in**  -daemon)  EXTRA\_ARGS="-daemon "$EXTRA\_ARGS  shift  ;;  \*)  ;; **esac**   #exec $base\_dir/kafka-run-class.sh $EXTRA\_ARGS org.apache.zookeeper.server.quorum.QuorumPeerMain "$@" exec $base\_dir/kafka-run-class.sh $EXTRA\_ARGS -Djava.security.auth.login.config=$base\_dir/../config/jaas-zookeeper.conf org.apache.zookeeper.server.quorum.QuorumPeerMain "$@" |

###### bin/kafka-server-sasl-start.sh

|  |
| --- |
| #!/bin/bash **if** [ $# -lt 1 ]; **then**  echo "USAGE: $0 [-daemon] server.properties [--override property=value]\*"  exit 1 **fi** base\_dir=$(dirname $0) **if** [ "x$KAFKA\_LOG4J\_OPTS" = "x" ]; **then**  export KAFKA\_LOG4J\_OPTS="-Dlog4j.configuration=file:$base\_dir/../config/log4j.properties" **fi** **if** [ "x$KAFKA\_HEAP\_OPTS" = "x" ]; **then**  export KAFKA\_HEAP\_OPTS="-Xmx1G -Xms1G" **fi** EXTRA\_ARGS=${EXTRA\_ARGS-'-name kafkaServer -loggc'} COMMAND=$1 **case** $COMMAND **in**  -daemon)  EXTRA\_ARGS="-daemon "$EXTRA\_ARGS  shift  ;;  \*)  ;; **esac** #exec $base\_dir/kafka-run-class.sh $EXTRA\_ARGS kafka.Kafka "$@" exec $base\_dir/kafka-run-class.sh $EXTRA\_ARGS -Djava.security.auth.login.config=$base\_dir/../config/jaas-kafka-server.conf kafka.Kafka "$@" |

##### Users setup

###### config/jaas-kafka-YodaAdmin.conf

|  |
| --- |
| KafkaClient {  org.apache.kafka.common.security.plain.PlainLoginModule required  username="\*\*\*\*\*\*"  password="\*\*\*\*\*\*"; }; |

###### bin/kafka-console-producer-sasl-YodaAdmin.sh

|  |
| --- |
| #!/bin/bash **if** [ "x$KAFKA\_HEAP\_OPTS" = "x" ]; **then**  export KAFKA\_HEAP\_OPTS="-Xmx512M" **fi**   base\_dir=$(dirname $0) exec $base\_dir/kafka-run-class.sh $EXTRA\_ARGS -Djava.security.auth.login.config=$base\_dir/../config/jaas-kafka-YodaAdmin.conf kafka.tools.ConsoleProducer "$@" |

###### bin/kafka-console-consumer-sasl-7e75b11a-5538-43b2-9a75-4155eb597589.sh

|  |
| --- |
| #!/bin/bash **if** [ "x$KAFKA\_HEAP\_OPTS" = "x" ]; **then**  export KAFKA\_HEAP\_OPTS="-Xmx512M" **fi** base\_dir=$(dirname $0) exec $base\_dir/kafka-run-class.sh $EXTRA\_ARGS -Djava.security.auth.login.config=$base\_dir/../config/jaas-kafka-client-7e75b11a-5538-43b2-9a75-4155eb597589.conf kafka.tools.ConsoleConsumer "$@" |

###### config/jaas-kafka-client-7e75b11a-5538-43b2-9a75-4155eb597589.conf

|  |
| --- |
| KafkaClient {  org.apache.kafka.common.security.plain.PlainLoginModule required  username="\*\*\*\*\*\*"  password="\*\*\*\*\*"; }; |

###### bin/kafka-console-consumer-sasl-17c1c77c-62a8-4499-9b45-e49e285e6ffe.sh

|  |
| --- |
| #!/bin/bash **if** [ "x$KAFKA\_HEAP\_OPTS" = "x" ]; **then**  export KAFKA\_HEAP\_OPTS="-Xmx512M" **fi** base\_dir=$(dirname $0) exec $base\_dir/kafka-run-class.sh $EXTRA\_ARGS -Djava.security.auth.login.config=$base\_dir/../config/jaas-kafka-client-17c1c77c-62a8-4499-9b45-e49e285e6ffe.conf kafka.tools.ConsoleConsumer "$@" |

###### config/jaas-kafka-client-17c1c77c-62a8-4499-9b45-e49e285e6ffe.conf

|  |
| --- |
| KafkaClient {  org.apache.kafka.common.security.plain.PlainLoginModule required  username="\*\*\*\*\*\*"  password="\*\*\*\*\*\*"; }; |

###### bin/kafka-console-consumer-sasl-3b0e5206-bb50-470e-8dc5-d4de5e1c86f7.sh

|  |
| --- |
| #!/bin/bash **if** [ "x$KAFKA\_HEAP\_OPTS" = "x" ]; **then**  export KAFKA\_HEAP\_OPTS="-Xmx512M" **fi** base\_dir=$(dirname $0) exec $base\_dir/kafka-run-class.sh $EXTRA\_ARGS -Djava.security.auth.login.config=$base\_dir/../config/jaas-kafka-client-3b0e5206-bb50-470e-8dc5-d4de5e1c86f7.conf kafka.tools.ConsoleConsumer "$@" |

###### config/jaas-kafka-client-3b0e5206-bb50-470e-8dc5-d4de5e1c86f7.conf

|  |
| --- |
| KafkaClient {  org.apache.kafka.common.security.plain.PlainLoginModule required  username="\*\*\*\*\*\*"  password="\*\*\*\*\*\*"; }; |

###### bin/kafka-console-consumer-sasl-6ac7e396-35b7-42a1-9c5e-a8f7349d5c5f.sh

|  |
| --- |
| #!/bin/bash **if** [ "x$KAFKA\_HEAP\_OPTS" = "x" ]; **then**  export KAFKA\_HEAP\_OPTS="-Xmx512M" **fi** base\_dir=$(dirname $0) exec $base\_dir/kafka-run-class.sh $EXTRA\_ARGS -Djava.security.auth.login.config=$base\_dir/../config/jaas-kafka-client-6ac7e396-35b7-42a1-9c5e-a8f7349d5c5f.conf kafka.tools.ConsoleConsumer "$@" |

###### config/jaas-kafka-client-6ac7e396-35b7-42a1-9c5e-a8f7349d5c5f.conf

|  |
| --- |
| KafkaClient {  org.apache.kafka.common.security.plain.PlainLoginModule required  username="\*\*\*\*\*"  password="\*\*\*\*\*\*\*\*"; }; |

###### bin/kafka-console-consumer-sasl-2cd423a4-d1a0-4df8-9a0d-1c24bc199e66.sh

|  |
| --- |
| #!/bin/bash **if** [ "x$KAFKA\_HEAP\_OPTS" = "x" ]; **then**  export KAFKA\_HEAP\_OPTS="-Xmx512M" **fi** base\_dir=$(dirname $0) exec $base\_dir/kafka-run-class.sh $EXTRA\_ARGS -Djava.security.auth.login.config=$base\_dir/../config/jaas-kafka-client-2cd423a4-d1a0-4df8-9a0d-1c24bc199e66.conf kafka.tools.ConsoleConsumer "$@" |

###### config/jaas-kafka-client-2cd423a4-d1a0-4df8-9a0d-1c24bc199e66.conf

|  |
| --- |
| KafkaClient {  org.apache.kafka.common.security.plain.PlainLoginModule required  username="\*\*\*\*\*"  password="\*\*\*\*\*\*\*\*"; }; |

###### bin/kafka-console-consumer-sasl-67bf521c-7e83-4f31-adad-962347ee1495.sh

|  |
| --- |
| #!/bin/bash   **if** [ "x$KAFKA\_HEAP\_OPTS" = "x" ]; **then**  export KAFKA\_HEAP\_OPTS="-Xmx512M" **fi** base\_dir=$(dirname $0) exec $base\_dir/kafka-run-class.sh $EXTRA\_ARGS -Djava.security.auth.login.config=$base\_dir/../config/jaas-kafka-client-67bf521c-7e83-4f31-adad-962347ee1495.conf kafka.tools.ConsoleConsumer "$@" |

###### config/jaas-kafka-client-67bf521c-7e83-4f31-adad-962347ee1495.conf

|  |
| --- |
| KafkaClient {  org.apache.kafka.common.security.plain.PlainLoginModule required  username="\*\*\*\*\*"  password="\*\*\*\*\*\*\*\*"; }; |

###### config/consumer-sasl-7e75b11a-5538-43b2-9a75-4155eb597589.properties

|  |
| --- |
| security.protocol=SASL\_PLAINTEXT sasl.mechanism=PLAIN #zookeeper.connect=localhost:2181 #zookeeper.connection.timeout.ms=6000 #group.id=test-consumer-group group.id=yptendermd |

###### config/consumer-sasl-17c1c77c-62a8-4499-9b45-e49e285e6ffe.properties

|  |
| --- |
| sasl.mechanism=PLAIN #zookeeper.connect=localhost:2181 #zookeeper.connection.timeout.ms=6000 #group.id=test-consumer-group group.id=achizitiimd |

###### config/consumer-sasl-3b0e5206-bb50-470e-8dc5-d4de5e1c86f7.properties

|  |
| --- |
| security.protocol=SASL\_PLAINTEXT sasl.mechanism=PLAIN #zookeeper.connect=localhost:2181 #zookeeper.connection.timeout.ms=6000 #group.id=test-consumer-group group.id=achizitiimd1 |

###### config/consumer-sasl-6ac7e396-35b7-42a1-9c5e-a8f7349d5c5f.properties

|  |
| --- |
| security.protocol=SASL\_PLAINTEXT sasl.mechanism=PLAIN #zookeeper.connect=localhost:2181 #zookeeper.connection.timeout.ms=6000 #group.id=test-consumer-group group.id=achizitiimd2 |

###### config/consumer-sasl-2cd423a4-d1a0-4df8-9a0d-1c24bc199e66.properties

|  |
| --- |
| security.protocol=SASL\_PLAINTEXT sasl.mechanism=PLAIN #zookeeper.connect=localhost:2181 #zookeeper.connection.timeout.ms=6000 #group.id=test-consumer-group group.id=elicitatiemd |

###### config/consumer-sasl-67bf521c-7e83-4f31-adad-962347ee1495.properties

|  |
| --- |
| security.protocol=SASL\_PLAINTEXT sasl.mechanism=PLAIN #zookeeper.connect=localhost:2181 #zookeeper.connection.timeout.ms=6000 #group.id=test-consumer-group group.id=ofertamd |

##### Run zookeeper

|  |
| --- |
| nohup bin/zookeeper-server-sasl-start.sh config/zookeeper-sasl.properties & # netstat -nltp | grep 2181 tcp 0 0 0.0.0.0:2181 0.0.0.0:\* LISTEN 60833/java |

##### Run kafka-server

|  |
| --- |
| nohup bin/kafka-server-sasl-start.sh config/server-sasl.properties & # netstat -nltp | grep 9092 tcp 0 0 0.0.0.0:9092 0.0.0.0:\* LISTEN 61400/java |

##### Create topics

|  |
| --- |
| bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic b015f078-536f-4094-852b-9d55dcfcd1da bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic a0d93286-5da1-4346-97ee-4dac1de0cc28 bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic 3564a5a2-9e66-41b1-801e-ccbaeac8be5f bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic 7474bc62-cac7-46a5-81ec-88b0d6be8504 |

##### List of topics

|  |
| --- |
| # bin/kafka-topics.sh --list --zookeeper localhost:2181 3564a5a2-9e66-41b1-801e-ccbaeac8be5f 7474bc62-cac7-46a5-81ec-88b0d6be8504 a0d93286-5da1-4346-97ee-4dac1de0cc28 b015f078-536f-4094-852b-9d55dcfcd1da |

##### Granting permissions for producer - allows you to set permissions on describe

|  |
| --- |
| bin/kafka-acls.sh --authorizer kafka.security.auth.SimpleAclAuthorizer --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:YodaAdmin --producer --topic b015f078-536f-4094-852b-9d55dcfcd1da   bin/kafka-acls.sh --authorizer kafka.security.auth.SimpleAclAuthorizer --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:YodaAdmin --producer --topic a0d93286-5da1-4346-97ee-4dac1de0cc28   bin/kafka-acls.sh --authorizer kafka.security.auth.SimpleAclAuthorizer --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:YodaAdmin --producer --topic 3564a5a2-9e66-41b1-801e-ccbaeac8be5f   bin/kafka-acls.sh --authorizer kafka.security.auth.SimpleAclAuthorizer --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:YodaAdmin --producer --topic 7474bc62-cac7-46a5-81ec-88b0d6be8504 |

##### Granting permissions to read and write to the topic

|  |
| --- |
| bin/kafka-acls.sh --authorizer kafka.security.auth.SimpleAclAuthorizer --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:YodaAdmin --operation Read --operation Write --topic b015f078-536f-4094-852b-9d55dcfcd1da   bin/kafka-acls.sh --authorizer kafka.security.auth.SimpleAclAuthorizer --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:YodaAdmin --operation Read --operation Write --topic a0d93286-5da1-4346-97ee-4dac1de0cc28   bin/kafka-acls.sh --authorizer kafka.security.auth.SimpleAclAuthorizer --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:YodaAdmin --operation Read --operation Write --topic 3564a5a2-9e66-41b1-801e-ccbaeac8be5f   bin/kafka-acls.sh --authorizer kafka.security.auth.SimpleAclAuthorizer --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:YodaAdmin --operation Read --operation Write --topic 7474bc62-cac7-46a5-81ec-88b0d6be8504 |

##### List of permissions for topics and consumers

|  |
| --- |
| bin/kafka-acls.sh --authorizer kafka.security.auth.SimpleAclAuthorizer --authorizer-properties zookeeper.connect=localhost:2181 --list |

##### Additional groups

If it is necessary that each user read messages independently of a group, then we divide these users into different groups. This is done so that when reading information from the topic by user-1, the index for user-2 should not be omitted. In other words, so that users do not intercept messages from the same topic and have the opportunity to receive this message for all users, regardless of who connected at what time.

###### Add permissions for new group

|  |
| --- |
| bin/kafka-acls.sh --authorizer kafka.security.auth.SimpleAclAuthorizer --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:7e75b11a-5538-43b2-9a75-4155eb597589 --operation Read --group yptendermd   bin/kafka-acls.sh --authorizer kafka.security.auth.SimpleAclAuthorizer --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:17c1c77c-62a8-4499-9b45-e49e285e6ffe --operation Read --group achizitiimd   bin/kafka-acls.sh --authorizer kafka.security.auth.SimpleAclAuthorizer --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:3b0e5206-bb50-470e-8dc5-d4de5e1c86f7 --operation Read --group achizitiimd1   bin/kafka-acls.sh --authorizer kafka.security.auth.SimpleAclAuthorizer --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:6ac7e396-35b7-42a1-9c5e-a8f7349d5c5f --operation Read --group achizitiimd2   bin/kafka-acls.sh --authorizer kafka.security.auth.SimpleAclAuthorizer --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:2cd423a4-d1a0-4df8-9a0d-1c24bc199e66 --operation Read --group elicitatiemd   bin/kafka-acls.sh --authorizer kafka.security.auth.SimpleAclAuthorizer --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:67bf521c-7e83-4f31-adad-962347ee1495 --operation Read --group ofertamd |

###### Permissions to read from topics for each user

|  |
| --- |
| bin/kafka-acls.sh --authorizer kafka.security.auth.SimpleAclAuthorizer --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:7e75b11a-5538-43b2-9a75-4155eb597589 --operation Read --topic b015f078-536f-4094-852b-9d55dcfcd1da   bin/kafka-acls.sh --authorizer kafka.security.auth.SimpleAclAuthorizer --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:17c1c77c-62a8-4499-9b45-e49e285e6ffe --operation Read --topic a0d93286-5da1-4346-97ee-4dac1de0cc28   bin/kafka-acls.sh --authorizer kafka.security.auth.SimpleAclAuthorizer --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:3b0e5206-bb50-470e-8dc5-d4de5e1c86f7 --operation Read --topic a0d93286-5da1-4346-97ee-4dac1de0cc28   bin/kafka-acls.sh --authorizer kafka.security.auth.SimpleAclAuthorizer --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:6ac7e396-35b7-42a1-9c5e-a8f7349d5c5f --operation Read --topic a0d93286-5da1-4346-97ee-4dac1de0cc28   bin/kafka-acls.sh --authorizer kafka.security.auth.SimpleAclAuthorizer --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:2cd423a4-d1a0-4df8-9a0d-1c24bc199e66 --operation Read --topic 3564a5a2-9e66-41b1-801e-ccbaeac8be5f   bin/kafka-acls.sh --authorizer kafka.security.auth.SimpleAclAuthorizer --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:67bf521c-7e83-4f31-adad-962347ee1495 --operation Read --topic 7474bc62-cac7-46a5-81ec-88b0d6be8504 |

###### Check the performance

User YodaAdmin writes posts in topics

|  |
| --- |
| bin/kafka-console-producer-sasl-YodaAdmin.sh --broker-list localhost:9092 --topic b015f078-536f-4094-852b-9d55dcfcd1da --producer.config=/home/kafka/config/producer-sasl.properties   bin/kafka-console-producer-sasl-YodaAdmin.sh --broker-list localhost:9092 --topic a0d93286-5da1-4346-97ee-4dac1de0cc28 --producer.config=/home/kafka/config/producer-sasl.properties   bin/kafka-console-producer-sasl-YodaAdmin.sh --broker-list localhost:9092 --topic 3564a5a2-9e66-41b1-801e-ccbaeac8be5f --producer.config=/home/kafka/config/producer-sasl.properties   bin/kafka-console-producer-sasl-YodaAdmin.sh --broker-list localhost:9092 --topic 7474bc62-cac7-46a5-81ec-88b0d6be8504 --producer.config=/home/kafka/config/producer-sasl.properties |

###### Receiving a message

|  |
| --- |
| bin/kafka-console-consumer-sasl-7e75b11a-5538-43b2-9a75-4155eb597589.sh --bootstrap-server localhost:9092 --topic b015f078-536f-4094-852b-9d55dcfcd1da --from-beginning --consumer.config=config/consumer-sasl-7e75b11a-5538-43b2-9a75-4155eb597589.properties   bin/kafka-console-consumer-sasl-17c1c77c-62a8-4499-9b45-e49e285e6ffe.sh --bootstrap-server localhost:9092 --topic a0d93286-5da1-4346-97ee-4dac1de0cc28 --from-beginning --consumer.config=config/consumer-sasl-17c1c77c-62a8-4499-9b45-e49e285e6ffe.properties   bin/kafka-console-consumer-sasl-3b0e5206-bb50-470e-8dc5-d4de5e1c86f7.sh --bootstrap-server localhost:9092 --topic a0d93286-5da1-4346-97ee-4dac1de0cc28 --from-beginning --consumer.config=config/consumer-sasl-3b0e5206-bb50-470e-8dc5-d4de5e1c86f7.properties   bin/kafka-console-consumer-sasl-6ac7e396-35b7-42a1-9c5e-a8f7349d5c5f.sh --bootstrap-server localhost:9092 --topic a0d93286-5da1-4346-97ee-4dac1de0cc28 --from-beginning --consumer.config=config/consumer-sasl-6ac7e396-35b7-42a1-9c5e-a8f7349d5c5f.properties  bin/kafka-console-consumer-sasl-2cd423a4-d1a0-4df8-9a0d-1c24bc199e66.sh --bootstrap-server localhost:9092 --topic 3564a5a2-9e66-41b1-801e-ccbaeac8be5f --from-beginning --consumer.config=config/consumer-sasl-2cd423a4-d1a0-4df8-9a0d-1c24bc199e66.properties   bin/kafka-console-consumer-sasl-67bf521c-7e83-4f31-adad-962347ee1495.sh --bootstrap-server localhost:9092 --topic 7474bc62-cac7-46a5-81ec-88b0d6be8504 --from-beginning --consumer.config=config/consumer-sasl-67bf521c-7e83-4f31-adad-962347ee1495.properties |

## 

## BPE-dns-MD

### Connection

|  |
| --- |
| ssh \*\*\*\*\*\*\*\* pass: \*\*\*\*\*\*\* |

### Setup

#### update and install

|  |
| --- |
| # apt-get update # apt-get install bind9 dnsutils |

#### /etc/resolv.conf

|  |
| --- |
| nameserver 127.0.0.1 |

#### /etc/bind/named.conf

|  |
| --- |
| include "/etc/bind/named.conf.options"; include "/etc/bind/named.conf.local"; include "/etc/bind/named.conf.default-zones"; include "/etc/bind/named.conf.zones"; |

#### /etc/bind/named.conf.zones

|  |
| --- |
| zone "mtender.gov.md" { type master; file "/etc/bind/zones/master/mtender.gov.md"; }; |

#### /etc/bind/zones/master/mtender.gov.md

|  |
| --- |
| $TTL 3600 @ IN SOA ns.mtender.gov.md. ns2.mtender.gov.md. ( 2018061000 ; Serial 900 ; Refresh 900 ; Retry 3600 ; Expire 3600 ; Minimum ) @ IN NS ns.mtender.gov.md. @ IN NS ns2.mtender.gov.md. @ IN A 10.73.10.25 www IN CNAME @ ns IN A 10.73.10.25 ns2 IN A 10.73.10.25 ac-node1 IN A 10.73.10.7 ac-node2 IN A 10.73.10.9 ac-node3 IN A 10.73.10.8 ak-node1 IN A 10.73.10.12 ak-node2 IN A 10.73.10.11 ak-node3 IN A 10.73.10.24 postgresql IN A 10.73.10.5 openvpn IN A 10.73.10.4 logstash IN A 10.73.10.17 nginx IN A 10.73.10.10 mconnect IN A 10.73.10.10 zabbix IN A 10.73.10.22 feed IN A 10.73.10.21 docker-node1 IN A 10.73.10.18 docker-node2 IN A 10.73.10.19 prometheus IN A 10.73.10.20 gitlab IN A 10.73.10.23 jenkins IN A 10.73.10.26 nodejs IN A 10.73.10.15 zabbix IN A 10.73.10.10 auction IN A 10.73.10.14 |

#### restart

|  |
| --- |
| # /etc/init.d/bind9 restart root@MFB3WEB026:~# netnetstat -nltp | grep 53 tcp 0 0 10.73.10.25:53 0.0.0.0:\* LISTEN 873/named tcp 0 0 127.0.0.1:53 0.0.0.0:\* LISTEN 873/named tcp 0 0 127.0.0.1:953 0.0.0.0:\* LISTEN 873/named |

#### Check from another server on the internal network

|  |
| --- |
| root@MFB3WEB002:~# dig @10.73.10.25 ac-node1.mtender.gov.md ; <<>> DiG 9.10.3-P4-Ubuntu <<>> @10.73.10.25 ac-node1.mtender.gov.md ; (1 server found) ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55453 ;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3   ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;ac-node1.mtender.gov.md. IN A   ;; ANSWER SECTION: ac-node1.mtender.gov.md. 3600 IN A 10.73.10.7   ;; AUTHORITY SECTION: mtender.gov.md. 3600 IN NS ns2.mtender.gov.md. mtender.gov.md. 3600 IN NS ns.mtender.gov.md.   ;; ADDITIONAL SECTION: ns.mtender.gov.md. 3600 IN A 10.73.10.25 ns2.mtender.gov.md. 3600 IN A 10.73.10.25   ;; Query time: 0 msec ;; SERVER: 10.73.10.25#53(10.73.10.25) ;; WHEN: Tue Jun 19 09:45:46 EEST 2018 ;; MSG SIZE rcvd: 174 |

## Docker-node1

Connection

|  |
| --- |
| ip: 10.73.10.18 ssh login: \*\*\*\*\*\*\*\* pass: \*\*\*\*\*\*\*\* sudo -s |

Install docker

##### Update cache of packet manager

|  |
| --- |
| # apt-get update # apt-get install apt-transport-https ca-certificates curl software-properties-common |

##### Docker’s official GPG key:

|  |
| --- |
| # curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add - OK |

##### Add repository

|  |
| --- |
| # add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" |

Install docker CE

|  |
| --- |
| # apt-get update # apt-get install docker-ce # docker run hello-world |

## 

## md.ELK-Stack

* E - Elasticsearch - a large non-relational and fast storage that scales perfectly
* L - Logstash - resource-intensive app with a bunch of features for parsing an unstructured log
* K - Kibana - web interface for Elasticsearch

How it works

On each virtual machine there is a Filebeat, in which it is written in the config which log files to watch for it. As soon as it detects changes, it sends it to Logstash line by line. Logstash receives in itself an input string sent to it by Filebeat. Runs it through all the filters, determines what to do with it and sends it to the desired output, to the desired Eraser index. The eraser receives the log, indexes it, making it searchable, and catches the timestamp field - it turns out the time-series table. The user enters Kibana, enters in the search bar what he needs to find there, and finds it. Or does not find.

#### Connection

|  |
| --- |
| internal ip: 10.73.10.17 ssh login: \*\*\*\*\*\*\*\* ssh pass: \*\*\*\*\*\*\*\*   http://md.kibana.eprocurement.systems login: \*\*\*\*\*\*\*\* pass: \*\*\*\*\*\*\*\* |

Install OpenJDK

#### JDK8

|  |
| --- |
| # apt-get update # add-apt-repository -y ppa:webupd8team/java # apt-get update # apt-get -y install openjdk-8-jre |

Install Elasticsearch

#### Set up an internal key:

|  |
| --- |
| # wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add - OK |

#### Package with the necessary dependencies for the following installation of Elastic

|  |
| --- |
| # apt-get install apt-transport-https |

#### Elastic repository. Update and installation

|  |
| --- |
| # echo "deb https://artifacts.elastic.co/packages/6.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-6.x.list deb https://artifacts.elastic.co/packages/6.x/apt stable main # apt-get update # apt-get install elasticsearch |

#### Edit config

|  |
| --- |
| # vim /etc/elasticsearch/elasticsearch.yml   network.host: 10.73.10.17 http.port: 9200 |

#### Startup

|  |
| --- |
| # /bin/systemctl daemon-reload # /bin/systemctl enable elasticsearch.service # /etc/init.d/elasticsearch restart |

#### Demon's initial check

|  |
| --- |
| # netstat -nltp | grep java   tcp 0 0 10.73.10.17:9300 0.0.0.0:\* LISTEN 60875/java tcp 0 0 10.73.10.17:9200 0.0.0.0:\* LISTEN 60875/java |

#### Curl check

|  |
| --- |
| # curl -XGET '10.73.10.17:9200/?pretty' {  "name" : "cmzfVg-",  "cluster\_name" : "\*\*\*\*\*\*\*\*",  "cluster\_uuid" : "\*\*\*\*\*\*\*\*",  "version" : {  "number" : "6.4.0",  "build\_flavor" : "default",  "build\_type" : "deb",  "build\_hash" : "595516e",  "build\_date" : "2018-08-17T23:18:47.308994Z",  "build\_snapshot" : **false**,  "lucene\_version" : "7.4.0",  "minimum\_wire\_compatibility\_version" : "5.6.0",  "minimum\_index\_compatibility\_version" : "5.0.0"  },  "tagline" : "You Know, for Search" } |

Install Kibana

#### Set up an public key:

|  |
| --- |
| # wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add - Ok |

#### Startup

|  |
| --- |
| # apt-get install kibana # /bin/systemctl daemon-reload # /bin/systemctl enable kibana.service |

#### Configuration file

|  |
| --- |
| # vim /etc/kibana/kibana.yml   server.port: 5601 server.host: "10.73.10.17" elasticsearch.url: "http://10.73.10.17:9200" |

#### Launch and test

|  |
| --- |
| # /etc/init.d/kibana restart # netstat -nltp | grep node tcp 0 0 10.73.10.17:5601 0.0.0.0:\* LISTEN 61307/node |

Install Logstash

Since public key and repositories are already there:

|  |
| --- |
| # apt-get install logstash |

#### Configuration and start

|  |
| --- |
| # vim /etc/logstash/conf.d/simple.conf    input {  tcp {  port => 5044  }  udp {  port => 5044  } } filter {  json {  source => "message"  } } output {  elasticsearch {  hosts => [ 'localhost' ]  user => '\*\*\*\*\*\*\*\*'  password => '\*\*\*\*\*\*\*\*'  } } |

#### Startup

|  |
| --- |
| # /bin/systemctl enable logstash.service # service logstash start # service logstash status # lsof -i | grep logstash java 6555 logstash 99u IPv4 28378 0t0 TCP \*:5044 (LISTEN) java 6555 logstash 100u IPv4 27240 0t0 UDP \*:5044 java 6555 logstash 106u IPv4 28402 0t0 TCP localhost:9600 (LISTEN) |

Nginx virtualhost 10.73.10.10 server (single access point)

#### htpasswd

|  |
| --- |
| # htpasswd /etc/nginx/.htpasswd elk New password: 2C4Kf9j4h\_o51hiN Re-type new password: 2C4Kf9j4h\_o51hiN |

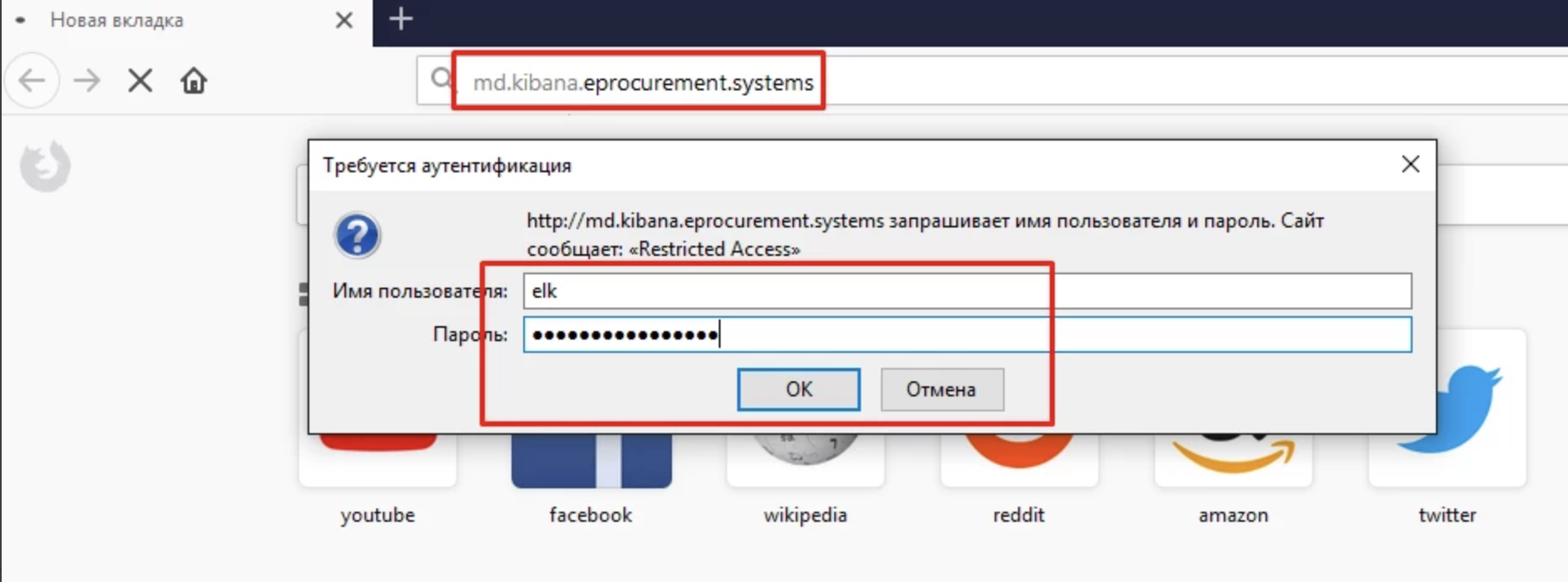
#### Virtualhost nginx:

|  |
| --- |
| # vim /etc/nginx/sites-available/md.kibana.conf   server {  listen 80;  server\_name md.kibana.eprocurement.systems;    auth\_basic "Restricted Access";  auth\_basic\_user\_file /etc/nginx/.htpasswd;    location / {  proxy\_pass http://10.73.10.17:5601;  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;  }  } |

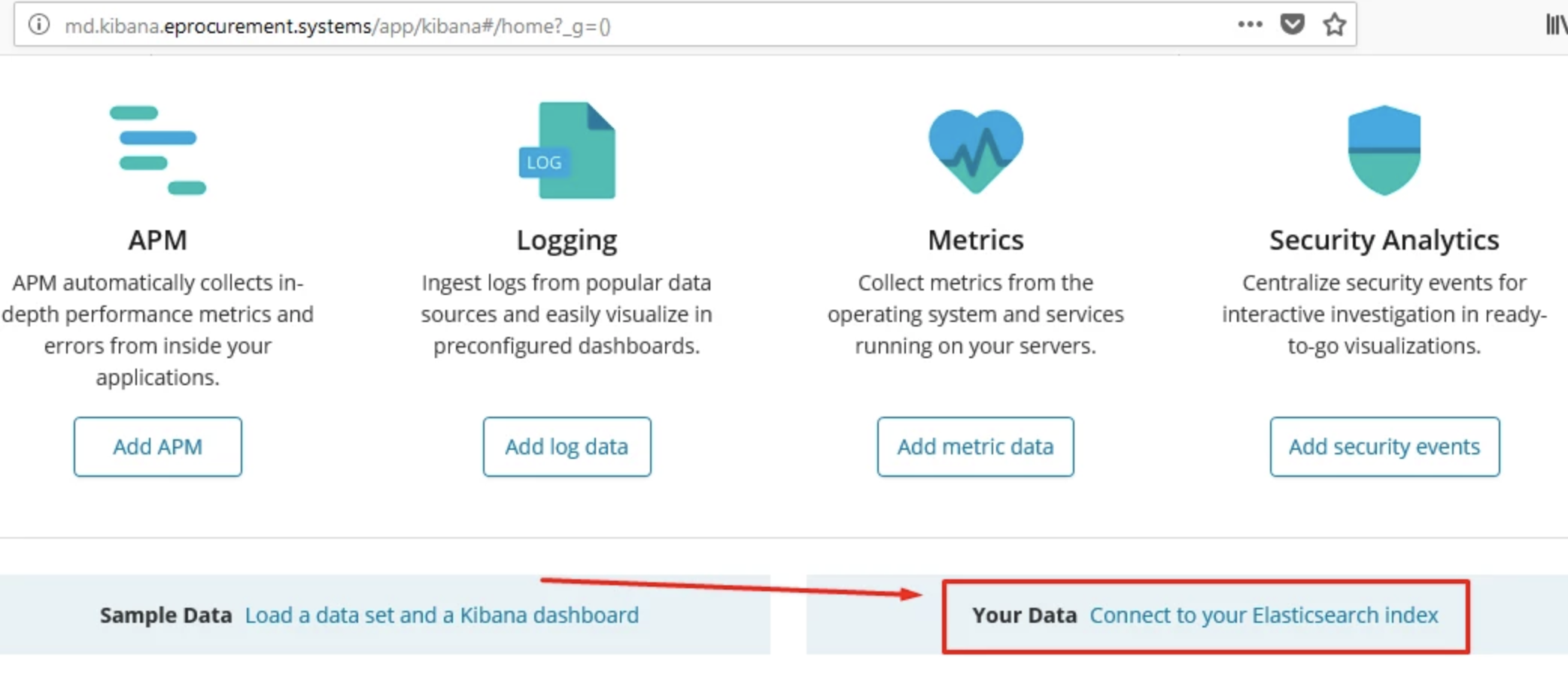
##### Start and test

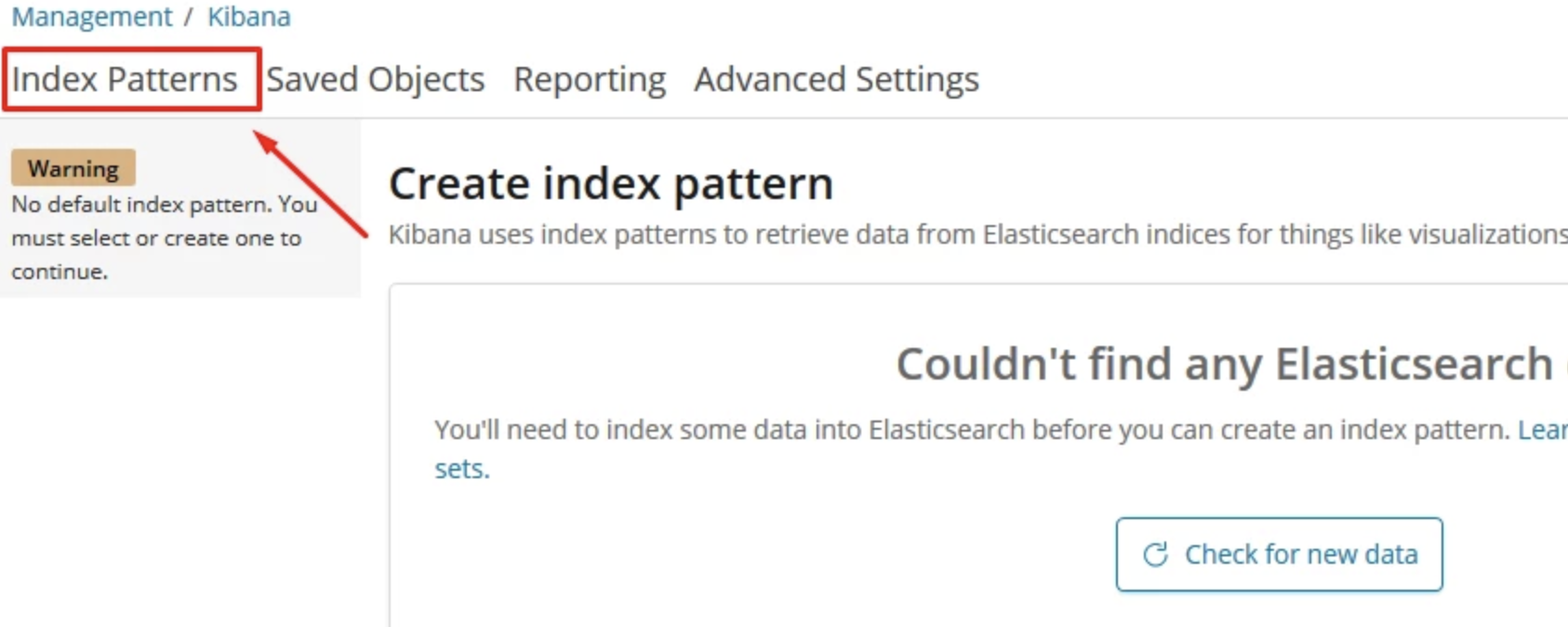
|  |
| --- |
| # ln -s /etc/nginx/sites-available/md.kibana.conf /etc/nginx/sites-enabled/md.kibana.conf # /etc/init.d/nginx restart Restarting nginx (via systemctl): nginx.service. # netstat -nltp | grep nginx tcp 0 0 0.0.0.0:80 0.0.0.0:\* LISTEN 10680/nginx -g daem |

Kibana configuration

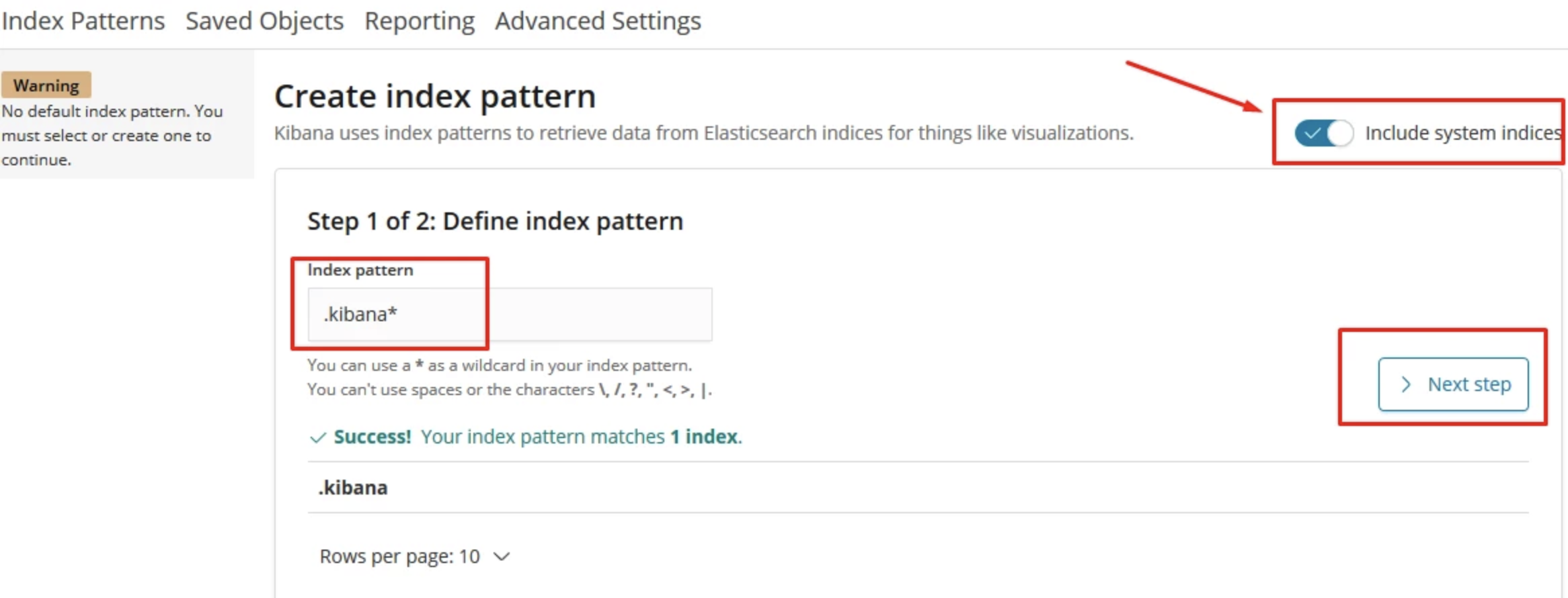


##### Create an index

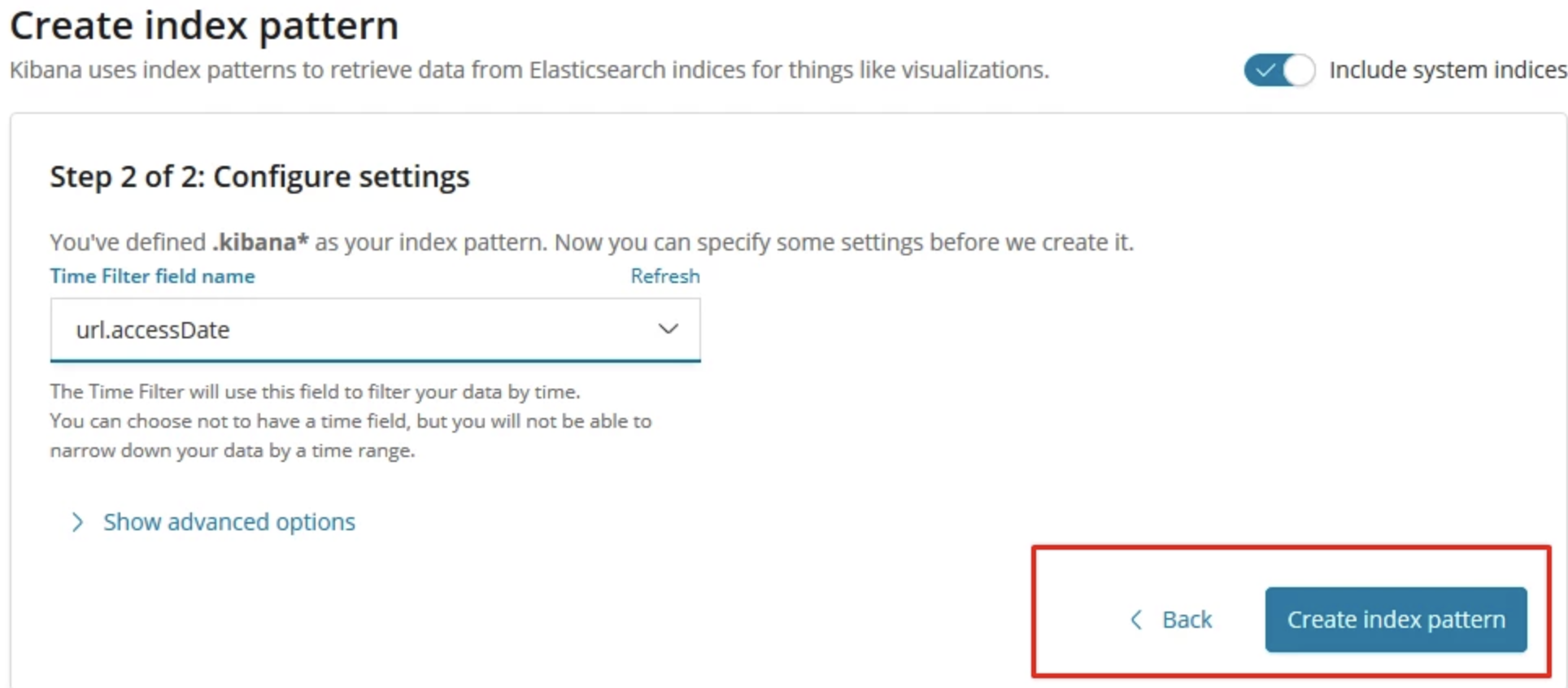




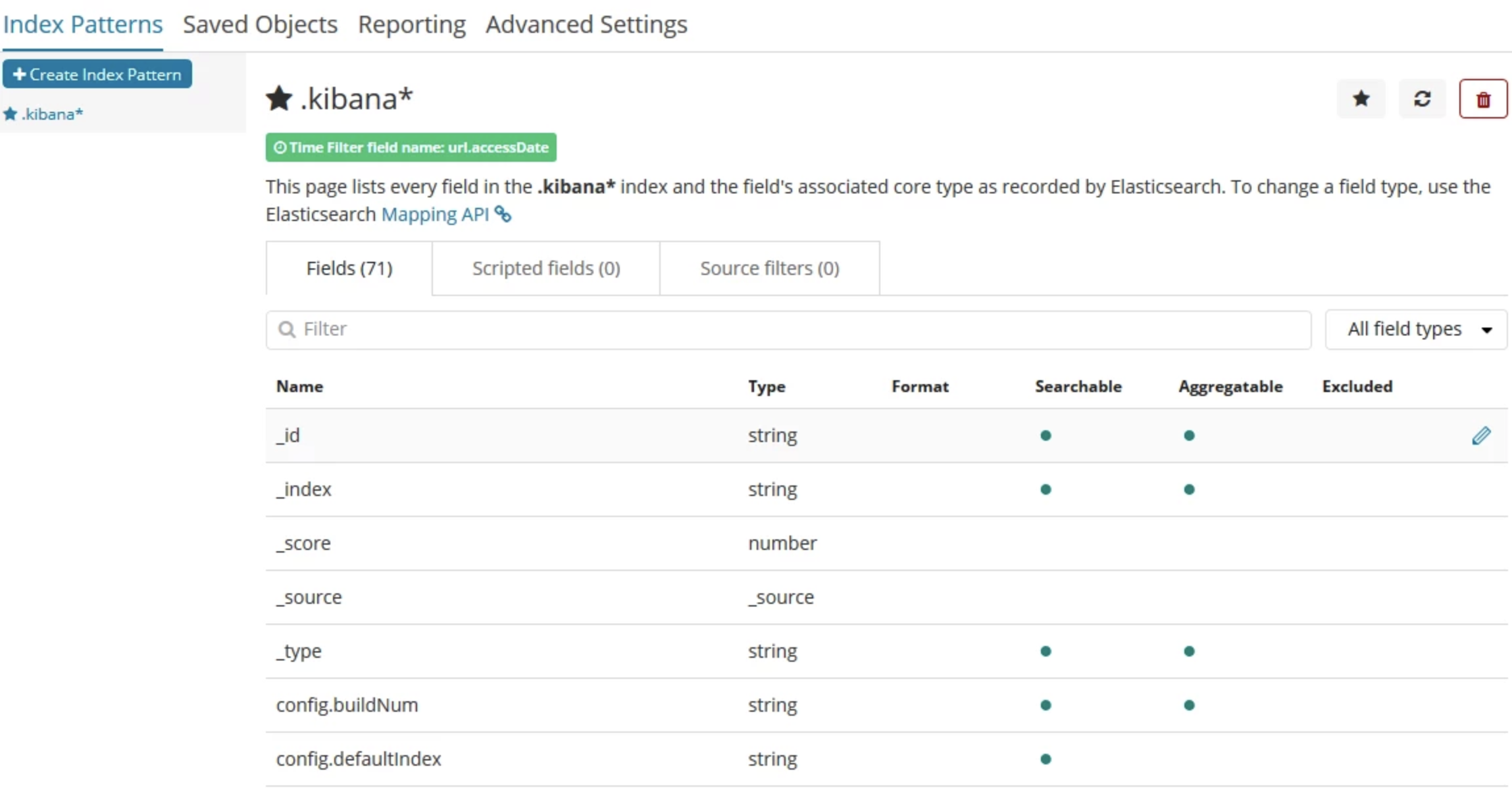
Insert *.kibana* and click *Next Step*



Next - dropdown menu



and click *Create Index Pattern*



In a same way all other indexes and other patterns can be created depending on the received data.

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

## md.Nginx-frontend

auth.mtender.gov.md

|  |
| --- |
| upstream authDockerSwarm {  server 10.73.10.19:8900 weight=1 max\_fails=4 fail\_timeout=600s;  server 10.73.10.6:8900 weight=1 max\_fails=4 fail\_timeout=600s;  server 10.73.10.13:8900 weight=1 max\_fails=4 fail\_timeout=600s; }  server {  listen \*:80;  server\_name auth.mtender.gov.md www.auth.mtender.gov.md;  location / {  proxy\_pass http://authDockerSwarm/api/v1/auth/$request\_uri; # proxy\_pass http://10.0.20.116:8900;  proxy\_set\_header Host $host;  proxy\_set\_header X-REAL-IP $remote\_addr;  proxy\_connect\_timeout 600;  proxy\_send\_timeout 600;  proxy\_read\_timeout 600;  proxy\_http\_version 1.1;  proxy\_set\_header Upgrade $http\_upgrade;  proxy\_set\_header Connection 'upgrade';  proxy\_cache\_bypass $http\_upgrade; # proxy\_set\_header Authorization "";  } } |

bpe.mtender.gov.md

|  |
| --- |
| upstream bpeDockerSwarm {  server 10.73.10.19:8900 weight=1 max\_fails=4 fail\_timeout=600s;  server 10.73.10.6:8900 weight=1 max\_fails=4 fail\_timeout=600s;  server 10.73.10.13:8900 weight=1 max\_fails=4 fail\_timeout=600s; }   server {  listen \*:80;  server\_name bpe.mtender.gov.md www.bpe.mtender.gov.md;  location / {  proxy\_pass http://bpeDockerSwarm/api/v1/$request\_uri;  proxy\_set\_header Host $host;  proxy\_set\_header X-REAL-IP $remote\_addr;  proxy\_connect\_timeout 600;  proxy\_send\_timeout 600;  proxy\_read\_timeout 600;  proxy\_http\_version 1.1;  proxy\_set\_header Upgrade $http\_upgrade;  proxy\_set\_header Connection 'upgrade';  proxy\_cache\_bypass $http\_upgrade; # proxy\_set\_header Authorization "";  } } |

gitlab.mtender.gov.md

|  |
| --- |
| server {  listen \*:80;  server\_name gitlab.mtender.gov.md;  access\_log off;  error\_log /dev/null;  auth\_basic "Restricted Access";  auth\_basic\_user\_file /etc/nginx/.htpasswd;   # ssl on; # ssl\_certificate /etc/letsencrypt/live/gitlab.mtender.gov.md/fullchain.pem; # ssl\_certificate\_key /etc/letsencrypt/live/gitlab.mtender.gov.md/privkey.pem;  location / {  proxy\_pass http://10.73.10.23;  proxy\_set\_header Host $host;  proxy\_set\_header X-REAL-IP $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_set\_header X-Forwarded-Proto $scheme;  proxy\_set\_header Authorization "";  } } |

mconnect.mtender.gov.md

|  |
| --- |
| upstream mconnectDockerSwarm {  server 10.73.10.19:8920 weight=1 max\_fails=4 fail\_timeout=600s;  server 10.73.10.6:8920 weight=1 max\_fails=4 fail\_timeout=600s;  server 10.73.10.13:8920 weight=1 max\_fails=4 fail\_timeout=600s; }   server {  listen \*:80;  server\_name mconnect.mtender.gov.md www.mconnect.mtender.gov.md;  location / { # proxy\_pass http://185.108.182.239:8044/api/v1/budget/$request\_uri;  proxy\_pass http://mconnectDockerSwarm/api/v1/mconnect/$request\_uri;  proxy\_set\_header Host $host;  proxy\_set\_header X-REAL-IP $remote\_addr;  proxy\_connect\_timeout 600;  proxy\_send\_timeout 600;  proxy\_read\_timeout 600;  proxy\_http\_version 1.1;  proxy\_set\_header Upgrade $http\_upgrade;  proxy\_set\_header Connection 'upgrade';  proxy\_cache\_bypass $http\_upgrade;  } } |

##### 

auction.mtender.gov.md

|  |
| --- |
| server { listen \*:443;  server\_name auction.mtender.gov.md;  ssl on;  ssl\_certificate /etc/nginx/ssl/wildcard-mntender.gov.md-fullchain.pem;  ssl\_certificate\_key /etc/nginx/ssl/wildcard-mntender.gov.md-privkey.pem;   # auth\_basic "Restricted Access"; # auth\_basic\_user\_file /etc/nginx/.htpasswd; # location /index2.html { # root /var/www/nginx; # index index.html; # expires 1M; # access\_log off; # add\_header Cache-Control "public"; # }  location / {  proxy\_pass http://10.73.10.14;  proxy\_http\_version 1.1;  proxy\_set\_header Upgrade $http\_upgrade;  proxy\_set\_header Connection 'upgrade';  proxy\_set\_header Host $host;  proxy\_cache\_bypass $http\_upgrade;  }  } server {  listen \*:80;  server\_name auction.mtender.gov.md;  rewrite ^(.\*)$ https://auction.mtender.gov.md$1 permanent; } |

md.kibana.eprocurement.systems

|  |
| --- |
| server {  listen 80;  server\_name md.kibana.eprocurement.systems;    auth\_basic "Restricted Access";  auth\_basic\_user\_file /etc/nginx/.htpasswd;    location / {  proxy\_pass http://10.73.10.17:5601;  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;  }  } |

zabbix.mtender.gov.md

|  |
| --- |
| server {  listen 443;  server\_name zabbix.mtender.gov.md;    auth\_basic "Restricted Access";  auth\_basic\_user\_file /etc/nginx/.htpasswd;    ssl on;  ssl\_certificate /etc/nginx/ssl/wildcard-mntender.gov.md-fullchain.pem;  ssl\_certificate\_key /etc/nginx/ssl/wildcard-mntender.gov.md-privkey.pem;    location / {  proxy\_pass http://10.73.10.22:81;  proxy\_http\_version 1.1;  proxy\_set\_header Upgrade $http\_upgrade;  proxy\_set\_header Connection 'upgrade';  proxy\_set\_header Host $host;  proxy\_cache\_bypass $http\_upgrade;  }  } server {  listen 80;  server\_name zabbix.mtender.gov.md;  rewrite ^(.\*)$ https://zabbix.mtender.gov.md$1 permanent; } |

operations.mtender.gov.md

|  |
| --- |
| upstream operationsDockerSwarm {  server 10.73.10.19:8900 weight=1 max\_fails=4 fail\_timeout=600s;  server 10.73.10.6:8900 weight=1 max\_fails=4 fail\_timeout=600s;  server 10.73.10.13:8900 weight=1 max\_fails=4 fail\_timeout=600s; } server {  listen \*:80;  server\_name operations.mtender.gov.md www.operations.mtender.gov.md;  location / {  proxy\_pass http://operationsDockerSwarm/api/v1/operations; # proxy\_pass http://10.0.20.116:8900;  proxy\_set\_header Host $host;  proxy\_set\_header X-REAL-IP $remote\_addr;  proxy\_connect\_timeout 600;  proxy\_send\_timeout 600;  proxy\_read\_timeout 600;  proxy\_http\_version 1.1;  proxy\_set\_header Upgrade $http\_upgrade;  proxy\_set\_header Connection 'upgrade';  proxy\_cache\_bypass $http\_upgrade; # proxy\_set\_header Authorization "";  } } |

portainer.eprocurement.systems

|  |
| --- |
| upstream DockerSwarmPortainer {  server 10.73.10.19:9000 weight=1 max\_fails=4 fail\_timeout=600s;  server 10.73.10.6:9000 weight=1 max\_fails=4 fail\_timeout=600s;  server 10.73.10.13:9000 weight=1 max\_fails=4 fail\_timeout=600s; }   server {  listen \*:443;  server\_name portainer.eprocurement.systems;  ssl on;  ssl\_certificate /etc/letsencrypt/live/portainer.eprocurement.systems/fullchain.pem;  ssl\_certificate\_key /etc/letsencrypt/live/portainer.eprocurement.systems/privkey.pem;  auth\_basic "Restricted Access";  auth\_basic\_user\_file /etc/nginx/.htpasswd;    location / {  proxy\_pass http://DockerSwarmPortainer;  set $token $arg\_token;  proxy\_set\_header Host $host;  proxy\_set\_header X-REAL-IP $remote\_addr;  proxy\_connect\_timeout 600;  proxy\_send\_timeout 600;  proxy\_read\_timeout 600;  proxy\_http\_version 1.1;  proxy\_set\_header Upgrade $http\_upgrade;  proxy\_set\_header Connection 'upgrade';  proxy\_cache\_bypass $http\_upgrade;  proxy\_set\_header Authorization "";  proxy\_set\_header Authorization "Bearer $token";  } }   server {  listen 80;  server\_name portainer.eprocurement.systems;  rewrite ^(.\*)$ https://portainer.eprocurement.systems$1 permanent; } |

public.mtender.gov.md

|  |
| --- |
| upstream DockerSwarmPublic {  server 10.73.10.19:8910 weight=1 max\_fails=4 fail\_timeout=600s;  server 10.73.10.6:8910 weight=1 max\_fails=4 fail\_timeout=600s;  server 10.73.10.13:8910 weight=1 max\_fails=4 fail\_timeout=600s; } server {  listen \*:443;  server\_name public.mtender.gov.md;  ssl on;  ssl\_certificate /etc/nginx/ssl/wildcard-mntender.gov.md-fullchain.pem;  ssl\_certificate\_key /etc/nginx/ssl/wildcard-mntender.gov.md-privkey.pem;  location / {  proxy\_pass http://DockerSwarmPublic;  proxy\_set\_header Host $host;  proxy\_set\_header X-REAL-IP $remote\_addr;  proxy\_connect\_timeout 600;  proxy\_send\_timeout 600;  proxy\_read\_timeout 600;  proxy\_http\_version 1.1;  proxy\_set\_header Upgrade $http\_upgrade;  proxy\_set\_header Connection 'upgrade';  proxy\_cache\_bypass $http\_upgrade;  } } server {  listen \*:80;  server\_name public.mtender.gov.md;  rewrite ^(.\*)$ https://public.mtender.gov.md$1 permanent; } |

storage.mtender.gov.md

|  |
| --- |
| upstream StorageDockerSwarm {  server 10.73.10.19:8900 weight=1 max\_fails=4 fail\_timeout=600s;  server 10.73.10.6:8900 weight=1 max\_fails=4 fail\_timeout=600s;  server 10.73.10.13:8900 weight=1 max\_fails=4 fail\_timeout=600s; }   server {  listen \*:80;  server\_name storage.mtender.gov.md www.storage.mtender.gov.md;  client\_max\_body\_size 55m;  location / {  proxy\_pass http://StorageDockerSwarm/api/v1/storage/$request\_uri;  proxy\_set\_header Host $host;  proxy\_set\_header X-REAL-IP $remote\_addr;  proxy\_connect\_timeout 600;  proxy\_send\_timeout 600;  proxy\_read\_timeout 600;  proxy\_http\_version 1.1;  proxy\_set\_header Upgrade $http\_upgrade;  proxy\_set\_header Connection 'upgrade';  proxy\_cache\_bypass $http\_upgrade; # proxy\_set\_header Authorization "";  } } |

## 

## md.storage-service

Connection

|  |
| --- |
| internal ip: 10.73.10.27 ssh login: \*\*\*\*\*\*\* ssh pass: \*\*\*\*\*\*\* |

Install JRE8

|  |
| --- |
| add-apt-repository -y ppa:webupd8team/java apt-get update apt-get -y install openjdk-8-jre |

cqlsh: client to connect to the Cassandra cluster

|  |
| --- |
| echo "deb http://debian.datastax.com/datastax-ddc 3.9 main" | tee -a /etc/apt/sources.list.d/cassandra.sources.list apt-key adv --keyserver keys.gnupg.net --recv-keys 350200F2B999A372 apt-get update apt-get install datastax-ddc-tools |

Launch script for Docker container

|  |
| --- |
| docker run \  --name="storage" \  --env \_\_PROP\_spring\_cloud\_config\_uri=http://IP-OF-THE-SWARM:PORT-CONFIG-SERVER-IN-SWARM (Example: --env \_\_PROP\_spring\_cloud\_config\_uri=http://10.10.10.100:8700) \  --env \_\_PROP\_spring\_profiles\_active=default \  --env \_\_PROP\_spring\_cloud\_config\_label=COMMIT-OF-THE-CONFIGURATION-STORAGE (Example: --env \_\_PROP\_spring\_cloud\_config\_label=aa56f07) \  --env \_\_PROP\_cassandra\_username=CASSANDRA-USERNAME \  --env \_\_PROP\_cassandra\_password=CASSANDRA-PASSWORD \  --publish published=8080,target=8080 \  --dns IP-OF-THE-INNER-DNS (Example: --dns 10.10.10.200)\  -v FILE-STORE-LOCATION:/storage (Example: -v /storage:/storage)\  eprocurementsystems/storage:TAG (Example: eprocurementsystems/storage:1.1.1.0a6292e) |

## 

## md.glusterFS

**When working with the client and copying a large number of files, instead of cp and mv, you need to use rsync!**

GlusterFS is a distributed file system that runs on top of an existing file system. Designed for organizing access and storage of microservice storage data.

Data storage method: distributed-replicated implies that if there are 2x nodes, the files are distributed between x nodes, and copies of the files are stored on the remaining nodes. Such a storage system allows you to not lose access to data even if x nodes fall (provided that unpaired nodes are unavailable).

In this case, 4 nodes are configured, which implies the distribution of data on 2 of them, and mirroring on two more. The order of specifying addresses during configuration matters: each even node is a mirror of the previous one.

Each part of glusterFS can work not only on separate nodes. Two directories can be located on one server, which will be independent of each other and in which either different files or the same ones (mirror) will be located, but in this case there is no fault tolerance as such.

The following settings are performed on each server node!

Provisional configuration of the server side

Add the alias and addresses for all server nodes into /etc/hosts

|  |
| --- |
| # vi /etc/hosts   10.73.10.28 gluster1.mtender.gov.md gluster1 10.73.10.29 gluster2.mtender.gov.md gluster2 10.73.10.30 gluster3.mtender.gov.md gluster3 10.73.10.32 gluster4.mtender.gov.md gluster4 |

Create directories for the repository. For convenience, we will use the names according to the functional load:

###### gluster1

|  |
| --- |
| **# mkdir /storage\_g1** |

###### gluster2

|  |
| --- |
| **# mkdir /storage\_g1\_repl** |

###### gluster3

|  |
| --- |
| **# mkdir /storage\_g3** |

###### gluster3

|  |
| --- |
| # mkdir /storage\_g3\_repl |

Half of the files will be stored on the gluster1 node and replicated on the gluster2 node. Similarly for the second half of the files on the nodes gluster3 and gluser4.

Setup of server side

|  |
| --- |
| # add-apt-repository ppa:gluster/glusterfs-3.13 # apt update # apt install glusterfs-server |

Setting up the server side (performed on gluster1). VOLUME\_NAME needs to be replaced with the repository name. It can be anything.

|  |
| --- |
| # gluster   gluster> peer probe gluster2 peer probe: success. gluster> peer probe gluster3 peer probe: success. gluster> peer probe gluster4 peer probe: success. gluster> peer status Number of Peers: 3   Hostname: gluster3 Uuid: (uuid) State: Peer **in** Cluster (Connected)   Hostname: gluster2 Uuid: (uuid) State: Peer **in** Cluster (Connected)   Hostname: gluster4 Uuid: (uuid) State: Peer **in** Cluster (Connected)   gluster> volume create VOLUME\_NAME replica 2 transport tcp gluster1:/storage\_g1 gluster2:/storage\_g1\_repl gluster3:/storage\_g3 gluster4:/storage\_g3\_repl Creation of volume VOLUME\_NAME has been successful. Please start the volume to access data. gluster> volume start VOLUME\_NAME Starting volume VOLUME\_NAME has been successful gluster> volume info   Volume Name: VOLUME\_NAME Type: Distributed-Replicate Volume ID: fb85ac55-4e85-4777-a6c7-95465b48cdfa Status: Started Snapshot Count: 0 Number of Bricks: 2 x 2 = 4 Transport-type: tcp Bricks: Brick1: gluster1:/storage\_g1 Brick2: gluster2:/storage\_g1\_repl Brick3: gluster3:/storage\_g2 Brick4: gluster4:/storage\_g2\_repl Options Reconfigured: nfs.disable: on   gluster> volume status Status of volume: VOLUME\_NAME Gluster process TCP Port RDMA Port Online Pid ------------------------------------------------------------------------------ Brick gluster1:/storage\_g1 49152 0 Y 2441 Brick gluster2:/storage\_g1\_repl 49152 0 Y 2314 Brick gluster3:/storage\_g3 49152 0 Y 2696 Brick gluster4:/storage\_g3\_repl 49153 0 Y 2195 Self-heal Daemon on localhost N/A N/A Y 3117 Self-heal Daemon on gluster3 N/A N/A Y 2818 Self-heal Daemon on gluster4 N/A N/A Y 2321 Self-heal Daemon on gluster2 N/A N/A Y 2848   Task Status of Volume VOLUME\_NAME ------------------------------------------------------------------------------ There are no active volume tasks |

Uuid may (and will) be different.

The following settings are performed on each client node!

Provisional configuration of the client side

In the file / etc / hosts, write the name of the current node (if not already entered), and the addresses of the server nodes. The configuration is illustrated by docker-swarm-node1. For other clients, you need to change the address, domain and name of the client.

|  |
| --- |
| # vi /etc/hosts   10.73.10.19 swarm-node1.mtender.gov.md swarm-node1 10.73.10.28 gluster1.mtender.gov.md gluster1 10.73.10.29 gluster2.mtender.gov.md gluster2 10.73.10.30 gluster3.mtender.gov.md gluster3 10.73.10.32 gluster4.mtender.gov.md gluster4 |

Prepare the mount points. For docker-swarm-X nodes and storage service, this is the directory / var / lib / docker / volumes / storage-vol / \_data

You also need to mount gluster to the storage-service node. Mount point / storage.

Make sure that this directory exists and is empty. Otherwise, you will need to create it, or save a copy of the files for further work with them.

If you need to mount the storage in a different way, you need to specify the required path.

Setup of client side

|  |
| --- |
| # add-apt-repository ppa:gluster/glusterfs-3.13 # apt update # apt install glusterfs-client |

Set up automatic storage mounting when the node starts:

|  |
| --- |
| # vi /etc/fstab   10.73.10.28:/VOLUME\_NAME /var/lib/docker/volumes/storage-vol/\_data glusterfs defaults,\_netdev 0 0 |

Mount storage

|  |
| --- |
| # mount -a # mount | grep gluster 10.73.10.28:/VOLUME\_NAME on /var/lib/docker/volumes/storage-vol\_data type fuse.glusterfs (rw,relatime,user\_id=0,group\_id=0,default\_permissions,allow\_other,max\_read=131072,\_netdev) |

Adding / Removing Nodes

The file system supports horizontal scaling to increase performance. Nodes or directories in this case are added only in pairs. At the same time, the storage is rebalanced and some of the files migrate from existing servers to new ones. To expand the cluster, you need to configure two more servers according to the example above. The example will use one server with two independent directories. When the new server is configured, you need to add on all servers of the glusterFS cluster its address and name in / etc / hosts. After that, all settings are performed on any node that is already in the cluster (in this case, the actions were performed on the gluster2 node).

|  |
| --- |
| # gluster gluster> peer status Number of Peers: 3   Hostname: gluster3 Uuid: 134d3b4b-7a5f-49ca-814e-3ee1197d5859 State: Peer **in** Cluster (Connected)   Hostname: gluster1.mtender.gov.md Uuid: 3d53ac81-1a19-4d9b-bbb0-1411336640ed State: Peer **in** Cluster (Connected) Other names: gluster1.mtender.gov.md   Hostname: gluster4 Uuid: 6d7fbd7a-ce27-4a8a-acc0-c5e9241ee7a1 State: Peer **in** Cluster (Connected)   gluster> peer probe gluster5 peer probe: success. gluster> peer status Number of Peers: 4   Hostname: gluster3 Uuid: 134d3b4b-7a5f-49ca-814e-3ee1197d5859 State: Peer **in** Cluster (Connected)   Hostname: gluster1.mtender.gov.md Uuid: 3d53ac81-1a19-4d9b-bbb0-1411336640ed State: Peer **in** Cluster (Connected) Other names: gluster1.mtender.gov.md Hostname: gluster4 Uuid: 6d7fbd7a-ce27-4a8a-acc0-c5e9241ee7a1 State: Peer **in** Cluster (Connected) Hostname: gluster5 Uuid: 8700ccd7-6b7b-4e8d-85c3-cc2b53f23cea State: Peer **in** Cluster (Connected) gluster> volume add-brick VOLUME\_NAME replica 2 gluster5:/storage\_g5 gluster5:/storage\_g5\_repl force volume add-brick: success gluster> volume status Status of volume: VOLUME\_NAME Gluster process TCP Port RDMA Port Online Pid Brick gluster1:/storage\_g1 49152 0 Y 2441 Brick gluster2:/storage\_g1\_repl 49152 0 Y 2314 Brick gluster3:/storage\_g3 49152 0 Y 2696 Brick gluster4:/storage\_g3\_repl 49153 0 Y 2195 Brick gluster5:/storage\_g5 49152 0 Y 1983 Brick gluster5:/storage\_g5\_repl 49153 0 Y 2004 Self-heal Daemon on localhost N/A N/A Y 5975 Self-heal Daemon on gluster3 N/A N/A Y 5886 Self-heal Daemon on gluster1 N/A N/A Y 6137 Self-heal Daemon on gluster4 N/A N/A Y 5266 Self-heal Daemon on gluster5 N/A N/A Y 2026 Task Status of Volume VOLUME\_NAME ------------------------------------------------------------------------------ There are no active volume tasks gluster> volume rebalance VOLUME\_NAME start volume rebalance: VOLUME\_NAME: success: Rebalance on VOLUME\_NAME has been started successfully. Use rebalance status command to check status of the rebalance process. ID: c6bbc8aa-5497-4357-9363-44a8170b22fe |

After adding a new node to volume and rebuilding, we can observe that part of the files migrated to the new server.

|  |
| --- |
| root@gluster5:~# ls /storage\_g5 \_data root@gluster5:~# ls /storage\_g5\_repl/ \_data root@gluster5:~# ls /storage\_g5\_repl/\_data/ 00 0e 1b 2b 3a 49 56 62 6f 7c 8c 9a a8 b6 c2 d0 dd e9 f6 01 0f 1d 2c 3b 4a 57 63 70 7d 8d 9b a9 b7 c4 d1 de ea f7 02 11 1e 2d 3d 4b 58 64 71 7f 8f 9c aa b8 c5 d2 df eb f8 03 12 1f 2e 3e 4d 59 65 72 80 90 9d ab b9 c6 d3 e0 ed f9 05 13 20 2f 40 4e 5a 66 73 82 91 9f ac ba c7 d4 e1 ee fa 07 14 21 30 41 4f 5b 67 74 83 92 a0 ad bb c8 d5 e2 ef fb 08 15 22 32 42 50 5c 68 75 85 93 a1 ae bc c9 d6 e3 f0 fc 09 16 23 34 44 51 5d 69 76 86 94 a2 af bd ca d7 e4 f1 fd 0a 17 24 35 45 52 5e 6a 77 87 95 a3 b0 be cb d8 e5 f2 fe 0b 18 28 37 46 53 5f 6b 78 88 96 a4 b1 bf cc d9 e6 f3 ff 0c 19 29 38 47 54 60 6d 7a 89 97 a5 b3 c0 cd da e7 f4 0d 1a 2a 39 48 55 61 6e 7b 8b 99 a7 b5 c1 ce dc e8 f5 |

Removing a node is performed from any server located in the cluster, excluding the server that must be deleted:

|  |
| --- |
| # gluster   gluster> volume status Status of volume: VOLUME\_NAME Gluster process TCP Port RDMA Port Online Pid ------------------------------------------------------------------------------ Brick gluster1:/storage\_g1 49152 0 Y 1466 Brick gluster2:/storage\_g1\_repl 49152 0 Y 1458 Brick gluster3:/storage\_g3 49152 0 Y 1485 Brick gluster4:/storage\_g3\_repl 49152 0 Y 1515 Brick gluster5:/storage\_g5 49152 0 Y 1479 Brick gluster5:/storage\_g5\_repl 49153 0 Y 1501 Self-heal Daemon on localhost N/A N/A Y 1455 Self-heal Daemon on gluster5 N/A N/A Y 1456 Self-heal Daemon on gluster2 N/A N/A Y 1449 Self-heal Daemon on gluster4 N/A N/A Y 1483 Self-heal Daemon on gluster3 N/A N/A Y 1462   gluster> volume remove-brick VOLUME\_NAME replica 2 gluster5:/storage\_g5 gluster5:/storage\_g5\_repl start volume remove-brick start: success ID: bfe48e7d-1ec1-4b87-a809-d73a04b50c52 |

After executing the delete command, gluster will start transferring files from the deleted node to the remaining ones. You can monitor the progress with the command:

|  |
| --- |
| gluster> volume remove-brick VOLUME\_NAME replica 2 gluster5:/storage\_g5 gluster5:/storage\_g5\_repl status  Node Rebalanced-files size scanned failures skipped status run time **in** h:m:s  --------- ----------- ----------- ----------- ----------- ----------- ------------ --------------  gluster5 0 0Bytes 1301 0 0 **in** progress 0:04:23 The estimated time **for** rebalance to complete will be unavailable **for** the first 10 minutes.   gluster> volume remove-brick VOLUME\_NAME replica 2 gluster5:/storage\_g5 gluster5:/storage\_g5\_repl status  Node Rebalanced-files size scanned failures skipped status run time **in** h:m:s  --------- ----------- ----------- ----------- ----------- ----------- ------------ --------------  gluster5 0 0Bytes 2160 0 0 completed 0:07:39 |

After the end, you can commit the changes and disable peer

|  |
| --- |
| gluster> volume remove-brick VOLUME\_NAME replica 2 gluster5:/storage\_g5 gluster5:/storage\_g5\_repl commit Removing brick(s) can result **in** data loss. Do you want to Continue? (y/n) Y volume remove-brick commit: success Check the removed bricks to ensure all files are migrated. If files with data are found on the brick path, copy them via a gluster mount point before re-purposing the removed brick.   gluster> peer detach gluster5 peer detach: success   gluster> peer status Number of Peers: 3   Hostname: gluster4 Uuid: 6d7fbd7a-ce27-4a8a-acc0-c5e9241ee7a1 State: Peer **in** Cluster (Connected)   Hostname: gluster3 Uuid: 134d3b4b-7a5f-49ca-814e-3ee1197d5859 State: Peer **in** Cluster (Connected)   Hostname: gluster2 Uuid: 703a1056-be4e-4e85-9040-d526b919f054 State: Peer **in** Cluster (Connected)   gluster> volume status Status of volume: VOLUME\_NAME Gluster process TCP Port RDMA Port Online Pid ------------------------------------------------------------------------------ Brick gluster1:/storage\_g1 49152 0 Y 1466 Brick gluster2:/storage\_g1\_repl 49152 0 Y 1458 Brick gluster3:/storage\_g3 49152 0 Y 1485 Brick gluster4:/storage\_g3\_repl 49152 0 Y 1515 Self-heal Daemon on localhost N/A N/A Y 1838 Self-heal Daemon on gluster4 N/A N/A Y 1691 Self-heal Daemon on gluster2 N/A N/A Y 1623 Self-heal Daemon on gluster3 N/A N/A Y 2078   Task Status of Volume VOLUME\_NAME ------------------------------------------------------------------------------ There are no active volume tasks |

## 

## md.Swarm-Cluster

Connection

|  |
| --- |
| Node1 (manager) ip: 10.73.10.19 login: \*\*\*\*\*\*\* pass: \*\*\*\*\*\*\*   Node2 (worker) ip: 10.73.10.6 login: \*\*\*\*\*\*\* pass: \*\*\*\*\*\*\*   Node3 (worker) ip: 10.73.10.13 login: \*\*\*\*\*\*\* pass: \*\*\*\*\*\*\* |

#### For all nodes

|  |
| --- |
| apt-get update apt-get install apt-transport-https software-properties-common ca-certificates -y wget https://download.docker.com/linux/ubuntu/gpg && apt-key add gpg echo "deb [arch=amd64] https://download.docker.com/linux/ubuntu xenial stable" >> /etc/apt/sources.list.d/docker.list apt-get update apt-get install docker-ce -y systemctl restart docker && systemctl enable docker |

#### Node-1 (manager)

|  |
| --- |
| docker swarm init --advertise-addr 10.73.10.19 Swarm initialized: current node (dxu44lekrn740rotjxipwovf1) is now a manager. To add a worker to this swarm, run the following command:  docker swarm join --token SWMTKN-1-5m39oqlzpa76rz3ltri16e720sc4a0bpuxntz1wl4tipen55yw-dw6q42s8w1i3sc3anuvbf4loj 10.73.10.19:2377 To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions. |

#### Add node2 and node3 in cluster.

|  |
| --- |
| docker swarm join --token SWMTKN-1-5m39oqlzpa76rz3ltri16e720sc4a0bpuxntz1wl4tipen55yw-dw6q42s8w1i3sc3anuvbf4loj 10.73.10.19:2377 |

#### Update node1:

|  |
| --- |
| docker node ls ID HOSTNAME STATUS AVAILABILITY MANAGER STATUS ENGINE VERSION dxu44lekrn740rotjxipwovf1 \* MFB3WEB020 Ready Active Leader 18.06.1-ce 9zgxy01hytb8ixx4kaoyejg4j MFB3WEB030 Ready Active 18.06.1-ce pqty69ff6m1d2vhbe7xole0nv MFB3WEB031 Ready Active 18.06.1-ce |

## md.Zabbix-v3.4

Connection

|  |
| --- |
| ip: 10.73.10.22 login: \*\*\*\*\*\* pass: \*\*\*\*\*\*   https://zabbix.mtender.gov.md http://10.73.10.22:81/ basic\_auth: \*\*\*\*\*\* / \*\*\*\*\*\*   login: \*\*\*\*\*\* pass: \*\*\*\*\*\* |

Preparing the environment

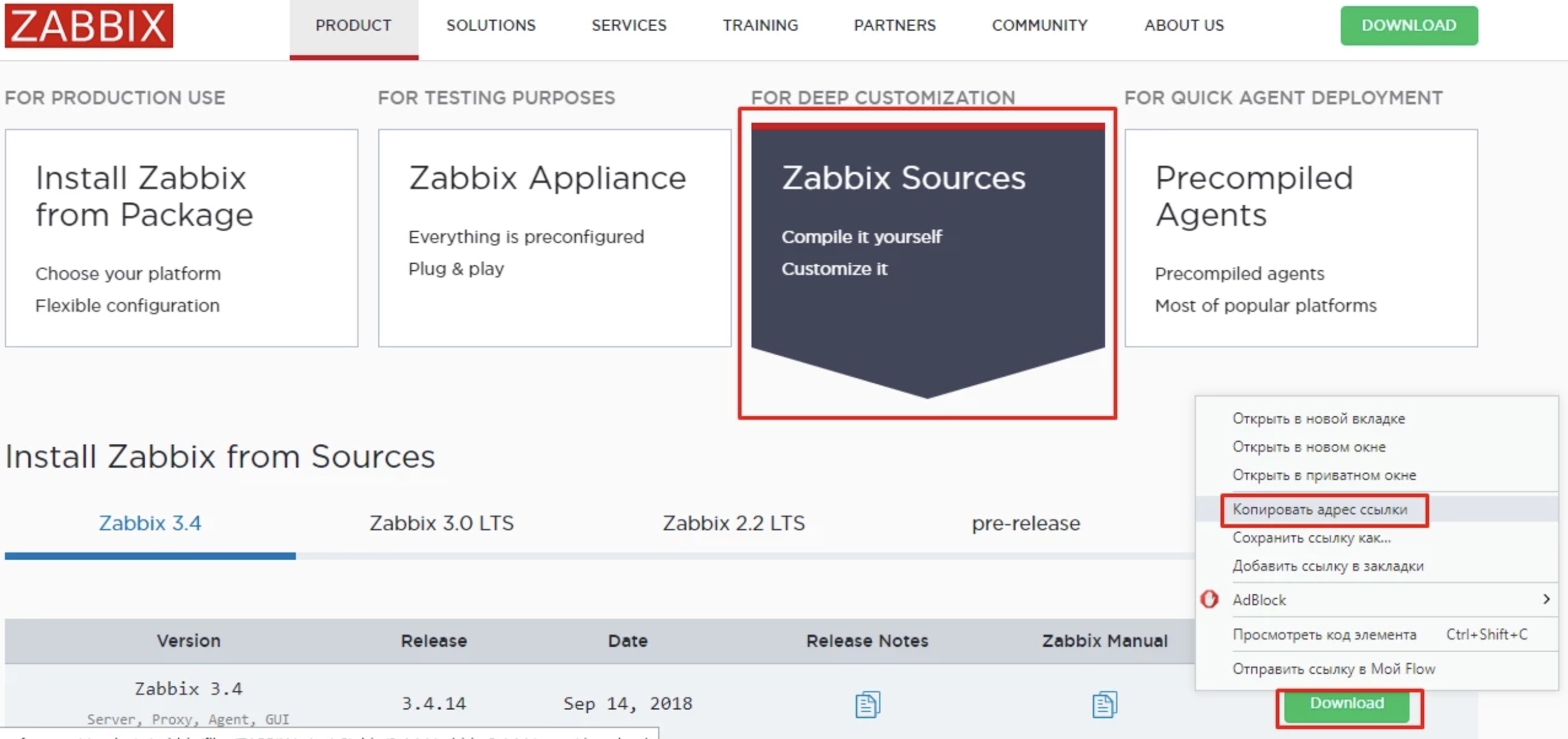
Install gcc and make - these packages are needed for future compilation of our zabbix from source.

|  |
| --- |
| apt-get update apt-get install gcc make |

Create a group and user

|  |
| --- |
| groupadd zabbix useradd -r -g zabbix zabbix |

An archive with source codes is required, which can be taken at <https://www.zabbix.com/download>



|  |
| --- |
| wget https://sourceforge.net/projects/zabbix/files/ZABBIX%20Latest%20Stable/3.4.14/zabbix-3.4.14.tar.gz tar xf zabbix-3.4.14.tar.gz cd zabbix-3.4.14/ |

In order to see possible compilation options, execute the command.

|  |
| --- |
| ./configure --help |

Install dependencies

|  |
| --- |
| apt-get install default-jdk |

|  |
| --- |
| apt-get install libmysqlclient-dev |

|  |
| --- |
| apt-get install libiksemel-dev |

|  |
| --- |
| apt-get install libcurl4-openssl-dev |

|  |
| --- |
| apt-get install libsnmp-dev snmptrapd |

|  |
| --- |
| apt-get install libopenipmi-dev |

|  |
| --- |
| apt-get install libxml2-dev |

|  |
| --- |
| apt-get install libssh2-1-dev |

|  |
| --- |
| apt-get install libldap2-dev |

|  |
| --- |
| apt-get install libevent-dev |

|  |
| --- |
| apt-get install curl |

|  |
| --- |
| apt-get install libcurl4-openssl-dev |

|  |
| --- |
| apt-get install libpcre++-dev |

|  |
| --- |
| apt-get autoremove |

Creating a Make File

|  |
| --- |
| ./configure --enable-server --enable-agent --enable-java --enable-ipv6 --with-mysql --with-jabber=/usr --with-libcurl --with-iodbc --with-net-snmp --with-ssh2 --with-openipmi --with-ldap |

After the configuration was successful, we build and install

|  |
| --- |
| make install |

After compilation, we need to do the creation of loaded scripts

|  |
| --- |
| cp misc/init.d/debian/zabbix-server /etc/init.d/ cp misc/init.d/debian/zabbix-agent /etc/init.d/ |

Making edits

#### /etc/init.d/zabbix-server:

|  |
| --- |
| #!/bin/sh ### BEGIN INIT INFO # Provides: zabbix-server # Required-Start: $remote\_fs $network # Required-Stop: $remote\_fs # Default-Start: 2 3 4 5 # Default-Stop: 0 1 6 # Should-Start: mysql # Should-Stop: mysql # Short-Description: Start zabbix-server daemon ### END INIT INFO   NAME=zabbix\_server DAEMON=/usr/local/sbin/${NAME} DESC="Zabbix server daemon" PID=/tmp/$NAME.pid . . . |

#### /etc/init.d/zabbix-agent:

|  |
| --- |
| #!/bin/sh ### BEGIN INIT INFO # Provides: zabbix-agent # Required-Start: $local\_fs $remote\_fs # Required-Stop: $local\_fs $remote\_fs # Should-Start: $all # Should-Stop: $all # Default-Start: 2 3 4 5 # Default-Stop: 0 1 6 # Short-Description: Start/stop Zabbix-agent # Description: Start/stop Zabbix-agent ### END INIT INFO   NAME=zabbix\_agentd DAEMON=/usr/local/sbin/${NAME} DESC="Zabbix agent daemon" PID=/tmp/$NAME.pid . . . |

Daemons should load at startup, so we’ll register autoload.

|  |
| --- |
| /usr/sbin/update-rc.d -f zabbix-server defaults /usr/sbin/update-rc.d -f zabbix-agent defaults |

Install mysql-server-5.7

|  |
| --- |
| apt-get install mysql-server mysql-client mysql\_secure\_installation |

#### MySQL access

|  |
| --- |
| login: \*\*\*\*\* pass: \*\*\*\*\* |

#### MySQL database and user for Zabbix

|  |
| --- |
| mysql -uroot -p Enter password: Welcome to the MySQL monitor. Commands end with ; or \g.   mysql> create database zabbix character set utf8 collate utf8\_bin; Query OK, 1 row affected (0.24 sec)   mysql> grant all privileges on zabbix.\* to zabbix@localhost identified by 'p4ylMbd97#mnszeI3ftpl5X'; Query OK, 0 rows affected (0.05 sec)   mysql> quit Bye |

#### Dumps, schemes and data:

|  |
| --- |
| cd ~/zabbix-3.4.14/database/mysql/ mysql -uroot -p -b zabbix < schema.sql mysql -uroot -p -b zabbix < images.sql mysql -uroot -p -b zabbix < data.sql |

|  |
| --- |
| vim /usr/local/etc/zabbix\_server.conf |

Find the DBPassword line, remove the comment and add the password:

|  |
| --- |
| DBPassword=\*\*\*\*\*\*\*\*\*\*\*\* |

Install PHP-7.2 PHP-FPM-7.2

|  |
| --- |
| apt-get install python-software-properties add-apt-repository ppa:ondrej/php apt-get update apt-get install php7.2 php7.2-fpm php-pear php7.2-curl php7.2-dev php7.2-gd php7.2-mbstring php7.2-zip php7.2-mysql php7.2-xml |

##### The user on behalf of which php-fpm will work

|  |
| --- |
| useradd monitor -d /dev/null -s /usr/sbin/nologin |

##### fpm settings

|  |
| --- |
| vim /etc/php/7.2/fpm/pool.d/monitor.conf   [zabbix.monitor.center] pm = dynamic pm.start\_servers = 1 pm.min\_spare\_servers = 1 pm.max\_children = 5 pm.max\_spare\_servers = 5 php\_admin\_value[display\_errors] = stderr php\_admin\_value[log\_errors] = On listen = /var/run/php72-fpm-zabbix.monitor.sock listen.mode = 0660 listen.owner = monitor listen.group = nginx user = monitor group = monitor chdir = / |

##### Start fpm

|  |
| --- |
| /etc/init.d/php7.2-fpm restart [ ok ] Restarting php7.2-fpm (via systemctl): php7.2-fpm.service. ls -l /var/run/php72-fpm-zabbix.monitor.sock srw-rw---- 1 monitor www-data 0 Sep 19 21:04 /var/run/php72-fpm-zabbix.monitor.sock |

##### Copy files for zabbix frontend

|  |
| --- |
| mkdir -p /var/www/monitor/domains/zabbix.monitor.center cd ~/zabbix-\*/frontends cp -a php/\* /var/www/monitor/domains/zabbix.monitor.center/ chown -R monitor:monitor /var/www/monitor |

##### Install nginx and configure virtualhost

|  |
| --- |
| wget http://nginx.org/keys/nginx\_signing.key apt-key add nginx\_signing.key   echo 'deb http://nginx.org/packages/mainline/ubuntu/ '$(lsb\_release -cs)' nginx' > /etc/apt/sources.list.d/nginx.list apt-get update apt-get install nginx nginx -v   nginx version: nginx/1.15.3 |

##### Virtualhost

|  |
| --- |
| vim /etc/nginx/conf.d/zabbix.mtender.gov.md.conf   server {  listen 80;  server\_name zabbix.mtender.gov.md;    access\_log /var/log/nginx/zabbix.monitor.center\_access.log combined;  error\_log /var/log/nginx/zabbix.monitor.center\_error.log;    root /var/www/monitor/domains/zabbix.monitor.center;  index index.php;  location ~\* ^.+\.(jpg|jpeg|gif|png|ico|zip|gz|rar|bz2|xls|html|exe|pdf|txt|wav|bmp|js|swf|css|xml)$ {  root /var/www/monitor/domains/zabbix.monitor.center;  }    location ~ \.php$ {  include fastcgi\_params;  fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;  fastcgi\_pass unix:/var/run/php72-fpm-zabbix.monitor.sock;  }  } |

|  |
| --- |
| nginx -t nginx: the configuration file /etc/nginx/nginx.conf syntax is ok nginx: configuration file /etc/nginx/nginx.conf test is successful   /etc/init.d/nginx restart |

##### php.ini settings

|  |
| --- |
| vim /etc/php/7.2/fpm/php.ini   post\_max\_size = 32M max\_execution\_time = 600 max\_input\_time = 600 date.timezone = Europe/Chisinau |

##### Installation of additional modules

|  |
| --- |
| apt-get install php7.2-bcmath apt-get install php7.2-ldap /etc/init.d/php7.2-fpm restart |