Elastic Stack 5.2 with X-Pack
Ubuntu Server 16.04.1 Build Instructions

PREFACE:
This guide is under the assumption that you are building a Elasticsearch cluster for a production environment; NOT one instance of Elasticsearch for a small scale of data. This guide is under the assumption that you have read the documentation on elastic.co about installing their products. When in doubt go to their really easy to follow guide on their website. There are many setting options and required changes to be made based on multiple factors such as; operating system version, product version, etc. This guide is designed to walk you through building a headless server for the Elastic Stack on Ubuntu Server 16.04.1 with Elastic Stack 5.0.

STEP 1 – INSTALL UBUNTU SERVER 16.04.1

- Partition Hard Drive Space
  - During the initial installation of the OS, ensure to select the custom partitioning setup option.
    - For HDD < 2TB - Set all space available to mount at /
    - For HDD > 2TB – select guided partition then delete the swap and /; then combine the available space for /.
    - NO SWAP!

- Packages Selection
  - Select the following when prompted for packages to install by using the spacebar to select the correct packages:
    - Manual Packages
    - Openssh Server
    - Standard System Utilities

STEP 2 – INSTALLATION OF SOFTWARE

- Update/upgrade:
  - sudo apt-get –y update && apt-get -y upgrade
  - sudo apt-get -y dist-upgrade

- Install Oracle Java Version 8
  - sudo add-apt-repository ppa:webupd8team/java
  - sudo apt-get update
  - sudo apt-get install oracle-java8-installer
  - sudo apt-get –y update && apt-get -y upgrade

- Install Elasticsearch via Debian Package
  - wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-5.2.2.deb
  - sudo dpkg -i elasticsearch-5.2.2.deb
    - If you want to set up as a service run these commands
      - sudo /bin/systemctl daemon-reload
      - sudo /bin/systemctl enable elasticsearch.service

- Install Logstash via Debian Package
  - wget https://artifacts.elastic.co/downloads/logstash/logstash-5.2.2.deb
  - sudo dpkg -i logstash-5.2.2.deb
    - To run as a service run:
      - sudo /bin/systemctl daemon-reload
      - sudo /bin/systemctl enable logstash.service

- Install Kibana via Debian Package
  - wget https://artifacts.elastic.co/downloads/kibana/kibana-5.2.2-amd64.deb
  - sudo dpkg -i kibana-5.2.2-amd64.deb
  - NOTE: When Security is Enabled for the X-Pack (Default) for Elasticsearch and Kibana, the default login username = elastic and the password = changeme
    - Running Kibana as a service run these commands:
      - sudo /bin/systemctl daemon-reload
      - sudo /bin/systemctl enable kibana.service

- Install X-Pack for Elasticsearch
  - cd /usr/share/elasticsearch
  - sudo bin/elasticsearch-plugin install x-pack

- Install X-Pack for Kibana
  - cd /usr/share/kibana
  - sudo bin/kibana-plugin install x-pack

- Install Translate Plugin and TLD-plugin for Logstash
  - cd /usr/share/logstash
  - sudo bin/logstash-plugin install logstash-filter-translate
  - sudo bin/logstash-plugin install logstash-filter-tld

- Install some extra stuff that will be helpful down the road
  - apt-get install curl htop git python-pip
  - pip install elastisticsearch-curator

STEP 3 – CONFIGURATION OF OPERATION SYSTEM (SYSTEMD)

- Modify the file descriptors (default for Ubuntu Server is 1024) – changes will not take effect until rebooting the system. Edit the limits.conf file.
  - vi /etc/security/limits.conf
  - Add the following at the bottom of the file (this sets the file descriptors to unlimited for all users and root)
    - *   soft nofile 65536
    - *   hard nofile 65536
    - root soft nofile 65536
    - root hard nofile 65536

- When using Debian packages on systems that use systemd, system limits must be specified via systemd. The systemd service file usr/lib/systemd/system/elasticsearch.service contains the limits that are applied by default. To override these, add a file called /etc/systemd/system/elasticsearch.service.d/elasticsearch.conf and specify any changes in that file, such as: LimitMEMLOCK=infinity
    - mkdir –p /etc/systemd/system/elasticsearch.service.d/
    - vi /etc/systemd/system/elasticsearch.service.d/elasticsearch.conf
    - Add the following line in the newly created file
        - LimitMEMLOCK=infinity

- Add Elasticsearch to the Sudo group (needed so that JVM can apply changes such as locking heap in memory)
    - usermod –aG sudo elasticsearch

- REBOOT
    - shutdown –r now
    - init 6

STEP 4 – CONFIGURATION OF SOFTWARE

Elasticsearch:

- Required Configuration Changes for Elasticsearch:
    - Set the minimum and maximum values for dedicate heap space for JVM. -Xms and –Xmx. Set these variables to 50% of the available RAM on your operating system (gb) but no more than 31 GB's. For example if you OS has 20 GB of RAM then set these values at 10g [DO NOT exceed 31gb of RAM]
        - vi /etc/Elasticsearch/jvm.options
            - -Xms=10g
            - -Xmx=10g

    - Uncomment the following and enter information appropriately
        - cluster.name: whate_ever_you_want
            - the cluster name must match on all your nodes in order to form a cluster
        - node.name: what_ever_you_want
            - I recommend something like name and purpose IE. Node1-EM, EM meaning Eligible Master
        - network.host: IP_ADDRESS_of_the_host
            - If using for local testing you can set to localhost
        - If you are building a cluster with mulptiple nodes, you must uncomment the following variable and enter the IP address of all the nodes in your cluster. IE:
            - discovery.zen.ping.unicast.hosts: ["Ip_of_otherNode1", "IP_of_otherNode2", "etc", "etc"]

- If building a cluster, TO AVOID SPLIT BRAIN use this formula to calculate the following variable setting [(master_eligible_nodes / 2) +1]. If not building a cluster, leave this setting commented in.
  - discovery.zen.minimum_master_nodes: 2

- Elasticsearch nodes have many different functionalities/duties. If you have a one node cluster don't worry about setting the following settings because the default settings for Elasticsearch are Master Eligible. But you are building a robust cluster you must designate the appropriate roles for you cluster. It is recommended to have a minimum of 2 Master eligible nodes in a cluster. Read the following link if you are confused: https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-node.html Add the following anywhere in the yml to set what type of node you are configuring:
  - Master Only Node
    - node.master: true
    - node.data: false
    - node.ingest: false

  - Master Eligible Node
    - node.master: true
    - node.data: true
    - node.ingest: true

  - Data Node
    - node.master: false
    - node.data: true
    - node.ingest: false

  - Coordinate Node
    - node.master: false
    - node.data: false
    - node.ingest: false

  - Tribe Node
    - Go to elastic.co (beyond the scope of this guide)

  - Ingest Node
    - Go to elastic.co (beyond the scope of this guide)

- If you need to disable Elasticsearch security feature that is a part of the XPACK add the following in the Elasticsearch.yml
  - nano /etc/elasticsearch/elasticsearch.yml
  - Add these into the yml
    - xpack.security.enabled: false

Kibana:

- Configure Kibana (Basic Configuration). It is recommended to dedicate a host just for Kibana. Kibana will be your Web Front End. Depending on how many users will be querying your datastore, it may be necessary to build a Elasticsearch coordinate node and drop Kibana on that host (instead of on a Eligible Master node). This will increase overall performance in your cluster by separating duties. Meaning that the a Master node can worry about master work, and a coordinate node can worry about the users queries.
  - Edit the kibana.yml file
    - vi /etc/kibana/kibana.yml

  - Uncomment and edit the following variables in the kibana.yml.
    - nano /etc/kibana/kibana.yml
    - server.host: "IP_of_coordinate_node"
      - For local testing you can leave at "localhost"
    - elasticsearch.url: http://IP_of_coordinate_Node:9200
      - For local testing you can leave at "localhost"
    - server.name: "Whatever_you_want"

Logstash:

- Configure Logstash (Basic Configuration)
  - Edit the jvm settings file
    - nano /etc/logstash/jvm.options

  - Set the minimum and maximum values for dedicate heap space for JVM. -Xms and –Xmx. Set these variables to 50% of the available RAM on your operating system (gb) IF LOGSTASH IS ON A DEDICATED HOST independent from elasticsearch, but no more than 31 GB's if Logstash is on a dedicated host independent from elasticsearch. For example, if you OS has 20 GB of RAM then set these values at 10g [DO NOT exceed 31gb of RAM]. IF LOGSTASH IS ON THE SAME HOST AS ELASTICSEARCH… Then set the JVM value to 20-25% percent of the RAM on the host.
    - vi /etc/Elasticsearch/jvm.options
      - -Xms=10g
      - -Xmx=10g

  - Uncomment and edit the following variables in the Logstash.yml as appropriate.
    - Uncomment and set you batchsize. This means how many events per worker you want each worker to process at a time.
      - pipeline.batch.size: 7500

    - Uncomment pipeline.batch.delay
      - pipeline.batch.delay: 60

    - Leave the rest of the yml as is unless you want to tweak some settings (Advanced)

- Download the filters from my github.
  - cd ~
  - git clone https://github.com/jeriel20/elastic5
  - copy the correct logstash filter into /etc/logstash/conf.d/ For Example:
    - cp elastic5/logstash_new/filters/10_bro_from_fileV3.conf /etc/logstash/conf.d/
    - cd /etc/logstash/conf.d
    - chmod 777 10_bro_from_fileV3.conf
    - Edit the path of the bro logs and the since_db path

- Check to make sure the configuration file has no errors
  - /usr/share/logstash/bin/logstash –path.settings=/etc/logstash -f /etc/logstash/conf.d/10_bro_from_fileV3.conf -t

- If the output does not come back with a "CONFIGURATION OK" then you need to look at the logstash log to see what happened.
  - tail -n 50 /var/log/logstash/logstash-plain.log | less -S

- Once the config is good to go start logstash manually or using systemd
  - /usr/share/logstash/bin/logstash –path.settings=/etc/logstash -f /etc/logstash/conf.d/10_bro_from_fileV3.conf

  OR
  - systemctl start logstash

- Now you can go to Kibana Web Front end to set your index pattern to index based off of the @timestamp metadata
  - Open a web browser and go to Kibana. http://localhost:5601 or http://IP_ADDRESS:5601
    - Select @timestamp for the index pattern for logstash-*

STEP 5 – CONFIGURE NETWORK INTERFACE

- Edit config but before you do ensure to annotate the actual interface name by running ifconfig command. You should see something that starts with eno.
  - sudo vi /etc/network/interfaces

```
# Loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eno3ps
iface eno3ps inet static
    address 192.168.1.101
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255
```

gateway 192.168.1.1

STEP 6 – POST INSTALL AND CONFIGURATION CHECKS

- Elasticsearch uses a hybrid mmapfs / niofs directory by default to store its indices. The default operating system limits on mmap counts is likely to be too low, which may result in out of memory exceptions. After rebooting run the following command as root and ensure that the value is shows 262144
  - sysctl vm.max_map_count

  - To set this value permanently, update the vm.max_map_count setting in /etc/sysctl.conf or you can set the value temporarily with the following command:
    - sysctl -w vm.max_map_count=262144

- Verify JVM heap got locked into memory:
  - free –h (RAM used should display as half of the memory in use – only if Elasticsearch binary has been started)

- Verify the daemons you have started bind to the host IP (as required):
  - netstat –plunt
    - Look for port 9200, 9300 (elasticsearch) and 5601 (Kibana) binded to IP of the host

    OR
    - lsof -i

- Verify daemon status if you enabled it in systemd
  - systemctl status elasticsearch
  - systemctl status logstash
  - systemctl status kibana

STEP 7 – BOOT NODES SYSTEMATICALLY

- Ensure to boot the node you want to be a master first, then boot the other nodes one by one.
- Use Kibana X-Pack monitoring view to ensure proper cluster formation.
- For advanced enterprise cluster wide settings to optimize for bulk indexing or search optimization, read the guides on elasics.co
- By default Elasticsearch is not setup for bulk indexing.