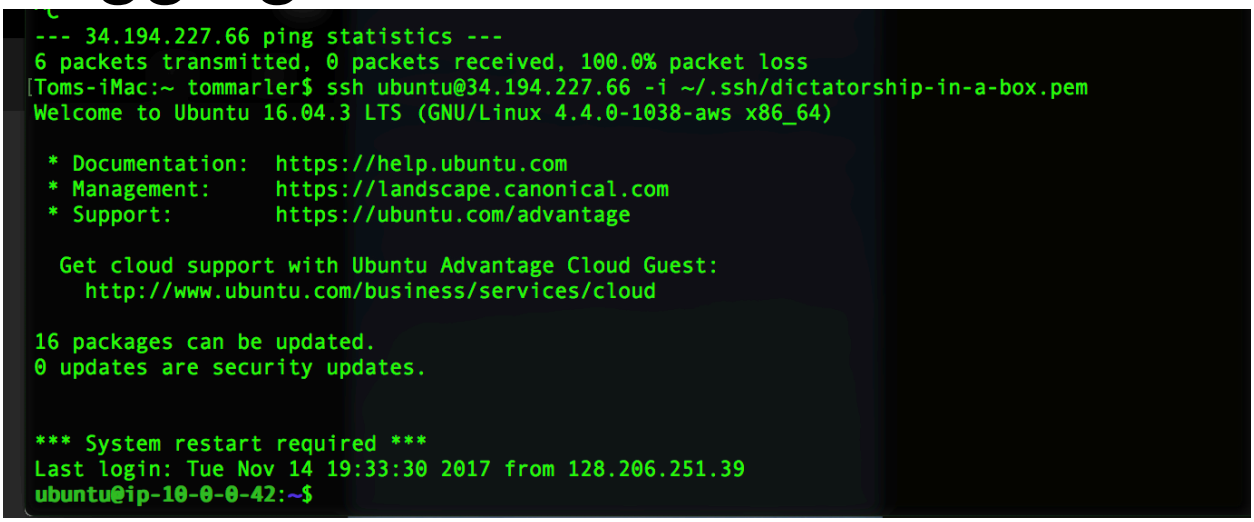
Logging in to AWS instance



ServerContainer

- Navigate to /ServerContainer and run
  - build -t server/securityonion
  - **Network-Research**/EvilBox/Images/**SecurityOnionServices.pn**

```
$PREFIX/etc/node.cfg  -- configure network interface to monitor
$PREFIX/etc/networks.cfg -- configure local networks
$PREFIX/etc/broctl.cfg -- change MailTo address and the log rotation
```

BRO INSTALL

- Now lets get started on the Bro IDS Installation under Ubuntu 16.04

Grab the required packages using apt.

```
apt install cmake make gcc g++ flex bison libpcap-dev libssl-dev python-dev swig zlib1g-dev libgeoip-dev
```

As you can see we have included the `libgeoip-dev` package as we are going to configure our installation with GeoIP support.

```
wget http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz
wget http://geolite.maxmind.com/download/geoip/database/GeoLiteCityv6-beta/GeoLiteCityv6.dat.gz
gzip -d GeoLiteCity.dat.gz
gzip -d GeoLiteCityv6.dat.gz
```

Now move the GeoIP files over to the default location `/usr/share/GeoIP/`, we need to rename them to match the location that Bro is expecting.

```
mv GeoLiteCity.dat /usr/share/GeoIP/GeoIPCity.dat
mv GeoLiteCityv6.dat /usr/share/GeoIP/GeoIPCityv6.dat
```

Install Bro on Ubuntu from package

```
sh -c "echo 'deb http://download.opensuse.org/repositories/network:/bro/xUbuntu_16.04/ /' > /etc/apt/sources
.list.d/bro.list"
apt update
apt install bro
```

Install Bro on Ubuntu from source

Download the source, extract and use the standard **configure, make, make install**.

```
wget https://www.bro.org/downloads/bro-2.5.1.tar.gz

tar zxvf bro-2.5.1.tar.gz
cd bro-2.5.1
./configure
make
make install
```

No errors? Good now add bro to your PATH.

```
export PATH=/usr/local/bro/bin:$PATH
```

You can also add PATH=/usr/local/bro/bin:$PATH to your ~/.profile file in your home directory to make the change permanent.

Bro is a powerful tool, to get started quickly we will follow the guide on the project page.

Edit the following files before starting:

```
$PREFIX/etc/node.cfg  -- configure network interface to monitor
$PREFIX/etc/networks.cfg -- configure local networks
$PREFIX/etc/broctl.cfg -- change MailTo address and the log rotation
```

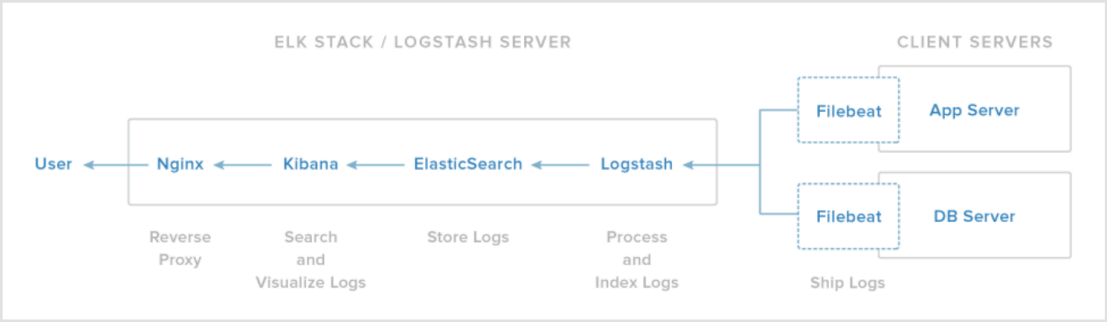
To start the program simply enter broctl at a shell.

```
====================|  Bro Build Summary  |=====================

Build type:        RelWithDebInfo
Build dir:         /root/bro/build
Install prefix:    /usr/local/bro
Bro Script Path:   /usr/local/bro/share/bro
Debug mode:        false

CC:                /usr/bin/cc
CFLAGS:             -Wall -Wno-unused -O2 -g -DNDEBUG
CXX:               /usr/bin/c++
CXXFLAGS:           -Wall -Wno-unused -std=c++11 -O2 -g -DNDEBUG
CPP:               /usr/bin/c++

Broker:            false
Broker Python:     false
Broccoli:          true
Broctl:            true
Aux. Tools:        true

GeoIP:             true
gperftools found:  false
        tcmalloc:  false
       debugging:  false
jemalloc:          false

=================================================================

-- Configuring done
-- Generating done
-- Build files have been written to: /root/bro/build
root@d8909c5d3e43:~/bro#
```

ELKStack

Our ELK stack setup has four main components:

- **Logstash**: The server component of Logstash that processes incoming logs
- **Elasticsearch**: Stores all of the logs
- **Kibana**: Web interface for searching and visualizing logs, which will be proxied through Nginx
- **Filebeat**: Installed on client servers that will send their logs to Logstash, Filebeat serves as a log shipping agent that utilizes the *lumberjack* networking protocol to communicate with Logstash



Java 8 Install

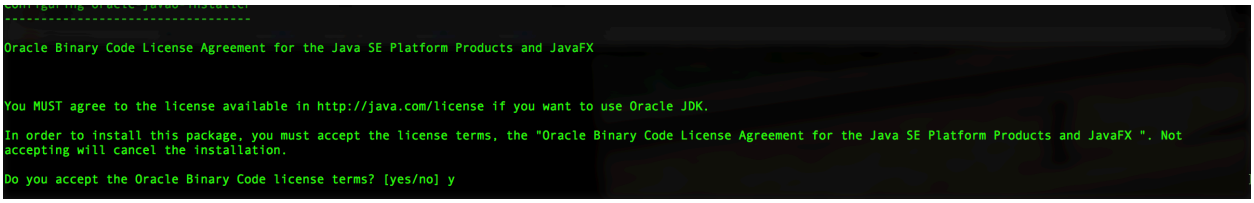
I am trying to download java8 it is a dependency that the Elkstack relies on

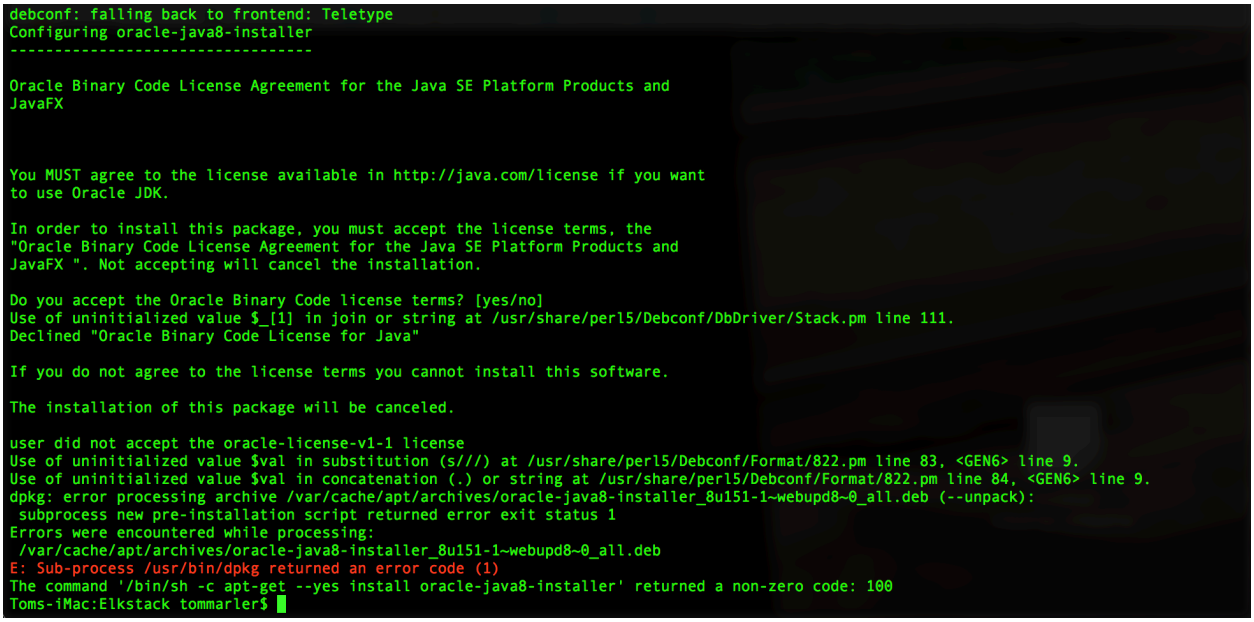
- Elkstack includes - Kibana, Logstash and elastic search
  - 1. Build Dockerfile for elkstack install
    - update and upgrade system and app ppa

```
## Elk Stack
FROM ubuntu:16.04

CMD "This is from the Metric Container debian:latest with ELk Stack"

# Update the System
RUN apt-get -y update && apt-get -y upgrade \
    software-properties-common \
    nano

# Install Java
RUN add-apt-repository -y ppa:webupd8team/java
RUN apt-get update
RUN apt-get -y install oracle-java8-installer
```

  - 2. Run docker build -t metric/elkstack . received an error and try to manually installed Java8 by /bin/bash

- For some reason Oracle needs the user to type yes and I guess -y command does not work for this.





Tire

> **--force-yes**
> Force yes. This is a dangerous option that will cause **apt-get** to continue without prompting if it is doing something potentially harmful. It should not be used except in very special situations. Using **--force-yes** can potentially destroy your system!
> Configuration Item: *APT::Get::force-yes*.

-y, **--yes**, **--assume-yes**
Automatic yes to prompts. Assume "yes" as answer to all prompts and run non-interactively. If an undesirable situation, such as changing a held package or removing an essential package, occurs then **apt-get** will abort.
Configuration Item: *APT::Get::Assume-Yes*.

Solution
https://askubuntu.com/questions/190582/installing-java-automatically-with-silent-option
If OpenJDK/OpenJRE works fine for you, I recommend using that package instead as suggested by @SAM. However, some software really requires Oracle's JDK/JRE. This answer is how to silence the license question with the Oracle package from the PPA.

First, let's recognize the question asked is a *feature* of the package, created by the developer.

```
oracle-java7-installer (7u7-0~webupd8~4) maverick; urgency=medium

  * removed cookie file use or else the PPA stays disabled
  * request the user to accept the Oracle license before installation
 -- Alin Andrei <webupd8@gmail.com>   Tue, 04 Sep 2012 14:18:29 +0200
```

As @Nate indicated in his answer, there should be a silent option. And there is. Do this before installing it:

```
$ echo debconf shared/accepted-oracle-license-v1-1 select true | \
  sudo debconf-set-selections
$ echo debconf shared/accepted-oracle-license-v1-1 seen true | \
  sudo debconf-set-selections
```

This sets the value of the debconf key to true, but also marks it as seen by the user. Now this question should not appear!

How did I find this?

In the source of the package, I tracked this down in the `oracle-java7-installer.preinst` file:

```
license=oracle-license-v1-1

# snip

db_get shared/accepted-$license
if [ "$RET" = "true" ]; then
    echo "$license license has already been accepted" >&2
    exit 0
fi
```

Apparantly, it uses debconf's value for the key `shared/accepted-oracle-license-v1-1` to check whether the user has already accepted the license. If it is, the script will exit gracefully and allow the installation to continue without asking you the question. We should now just tell debconf you already accept the Oracle Licence 1.1.

Please refer to the manpage of `debconf-set-selections` on more details, but this is the example for your issue and works similar for other packages. What other keys do you have on your system in debconf's database? Install `debconf-utils` and do

```
$ sudo debconf-get-selections
```

Then grep for more keys you need to set in your automated installation. This is way more flexible than using `-y` with `apt-get` as it gives you the opportunity to set other than default settings on installation times.

Java Installed

- docker build -t metric/elkstack
- docker run -itd metric/elkstack /bin/bash
- docker attach metric/elkstack
- java -version

```
root@92f9d4f12acb:/#
root@92f9d4f12acb:/# java -version
java version "1.8.0_151"
Java(TM) SE Runtime Environment (build 1.8.0_151-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.151-b12, mixed mode)
root@92f9d4f12acb:/#
```

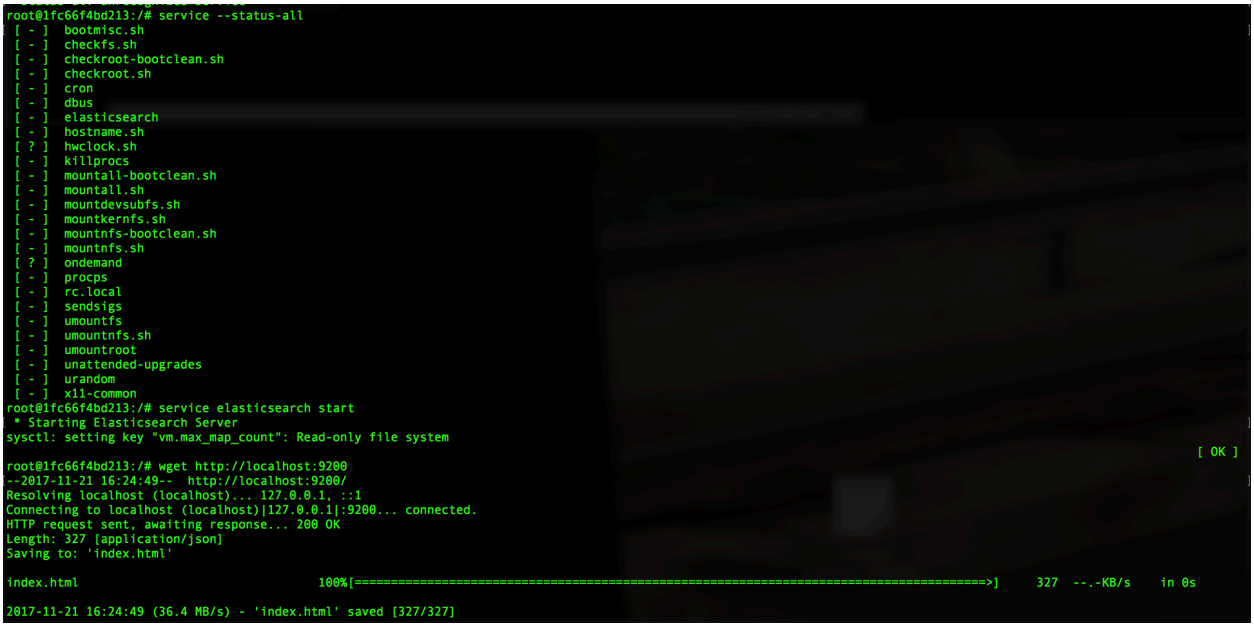
```
# ElasticSearch repo added
## Get elasticsearch signed key
### Update and upgrade
RUN echo "deb https://artifacts.elastic.co/packages/5.x/apt stable main" | tee -a /etc/apt/sources.list.d/elastic-5.x.lis
RUN wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | apt-key add - \
    && apt-get update && apt-get upgrade

RUN apt-get -y install elasticsearch \
    && service elasticsearch start \
    && service elasticsearch status
```

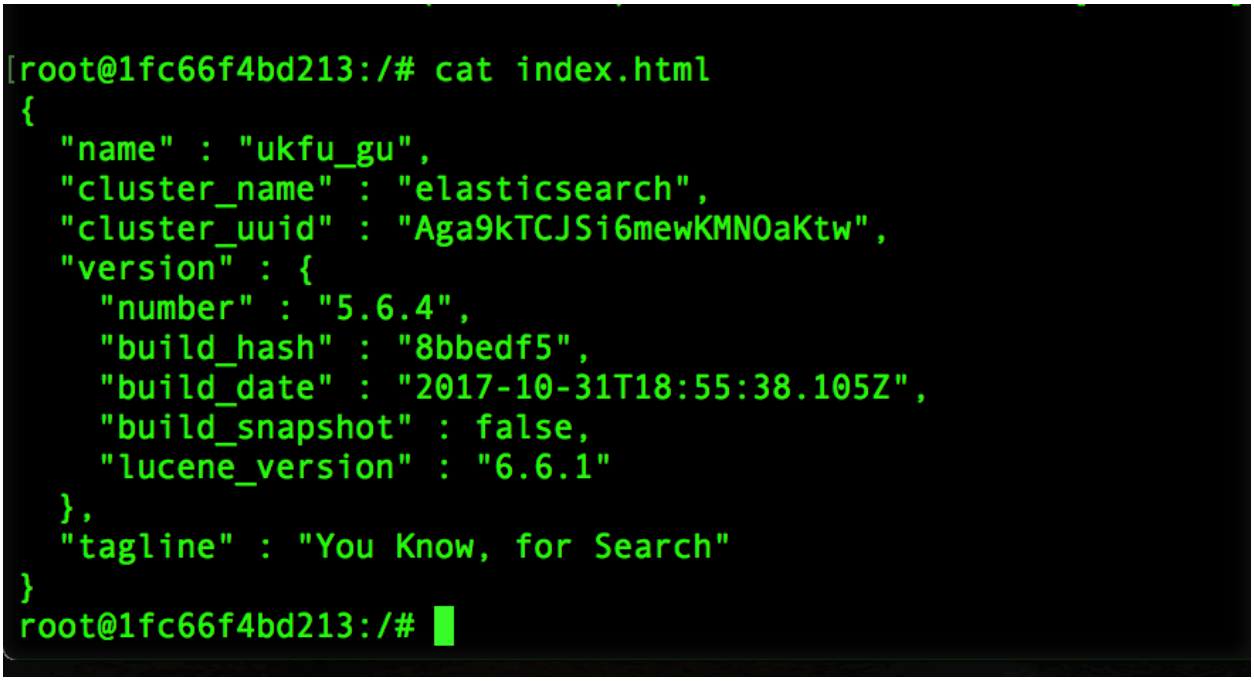
```
Failed to connect to bus: No such file or directory
Processing triggers for systemd (229-4ubuntu21) ...
 * Starting Elasticsearch Server
sysctl: setting key "vm.max_map_count": Read-only file system
   ...done.
 * elasticsearch is running
 ---> effcf316edf4
Removing intermediate container 5954e695689f
Successfully built effcf316edf4
Successfully tagged metric/elkstack:latest
Toms-iMac:Elkstack tommarler$
```

Problem: Service elastic search start in docker file is not running with container starts, even though pictures above tells different story

Quick Fix: docker run -itd elkstack /bin/bash -> docker attach elkstack

- manually start elastic search with -> service elastic search start
- curl localhost:9200 to issue elasticsearch is working properly





Install Kibana

- sudo apt-get install kibana
- runs on port 5600

- wget http://localhost:5600 -> cat index.html