# Version History

|  |  |  |
| --- | --- | --- |
| Version | Date | Release Note |
| V0.1 | 10-Mar-2020 | Initial Draft |
| V1.0 | 2-Apr-2020 | 1st Release |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

index

1 Version History 2

2 Exercise 1 4

2.1 ELASTICSERACH INstallation (optional) 4

2.2 logstash installation (optional) 7

2.3 KIBANA INSTALL (optional) 8

2.4 Port Mapping 10

3 Start Vagrant box 11

4 Kibana Searching 12

# Exercise 1

The exercise will guide you how to use Kibana

Prerequisite

1. A Vagrant box for ELK installation

## ELASTICSERACH INSTALLATION (optional)

#Download the signing key and add elasticsearch repository

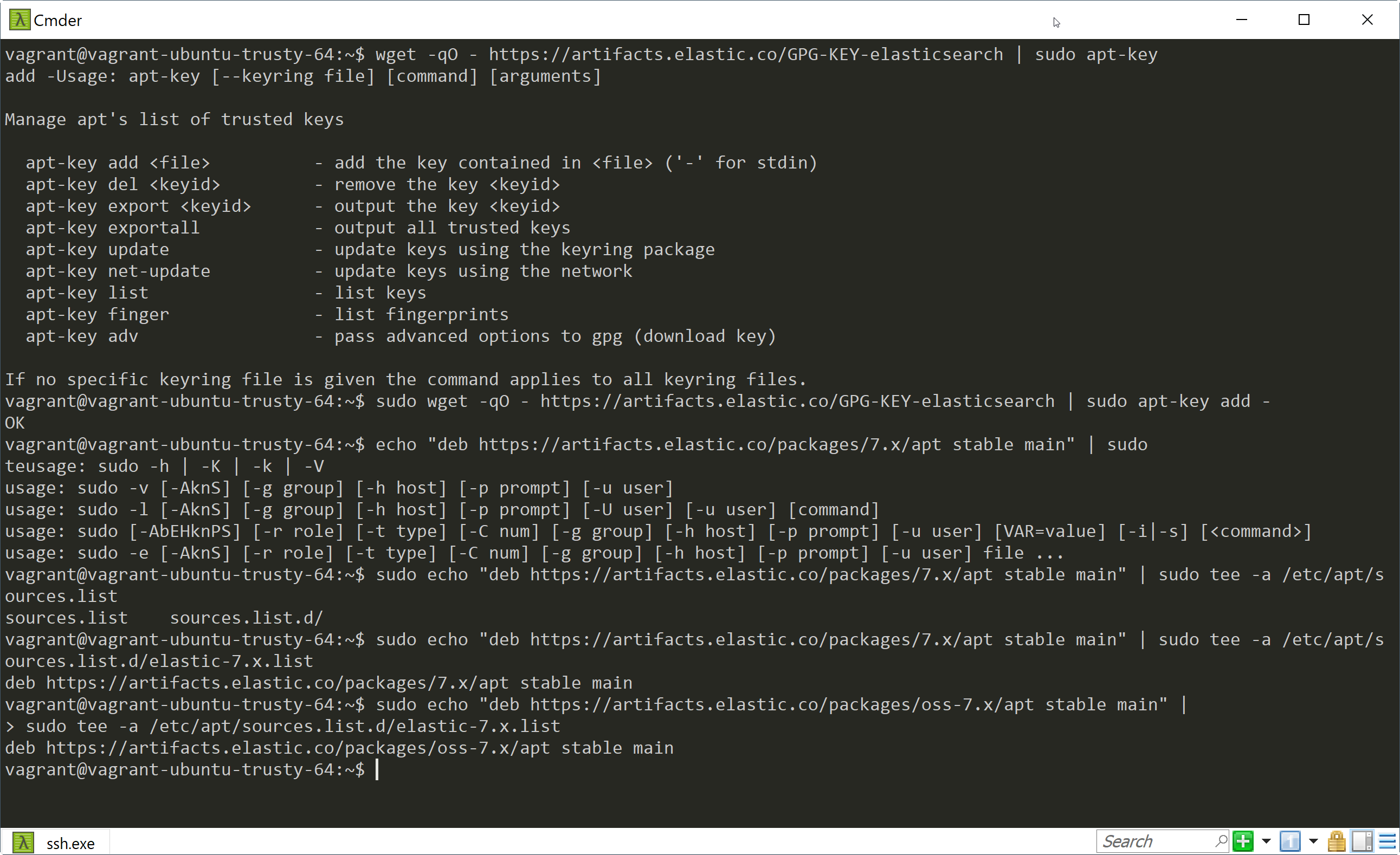
wget -qO – <https://artifacts.elastic.co/GPG-KEY-elasticsearch> | sudo apt-key add -

sudo echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo

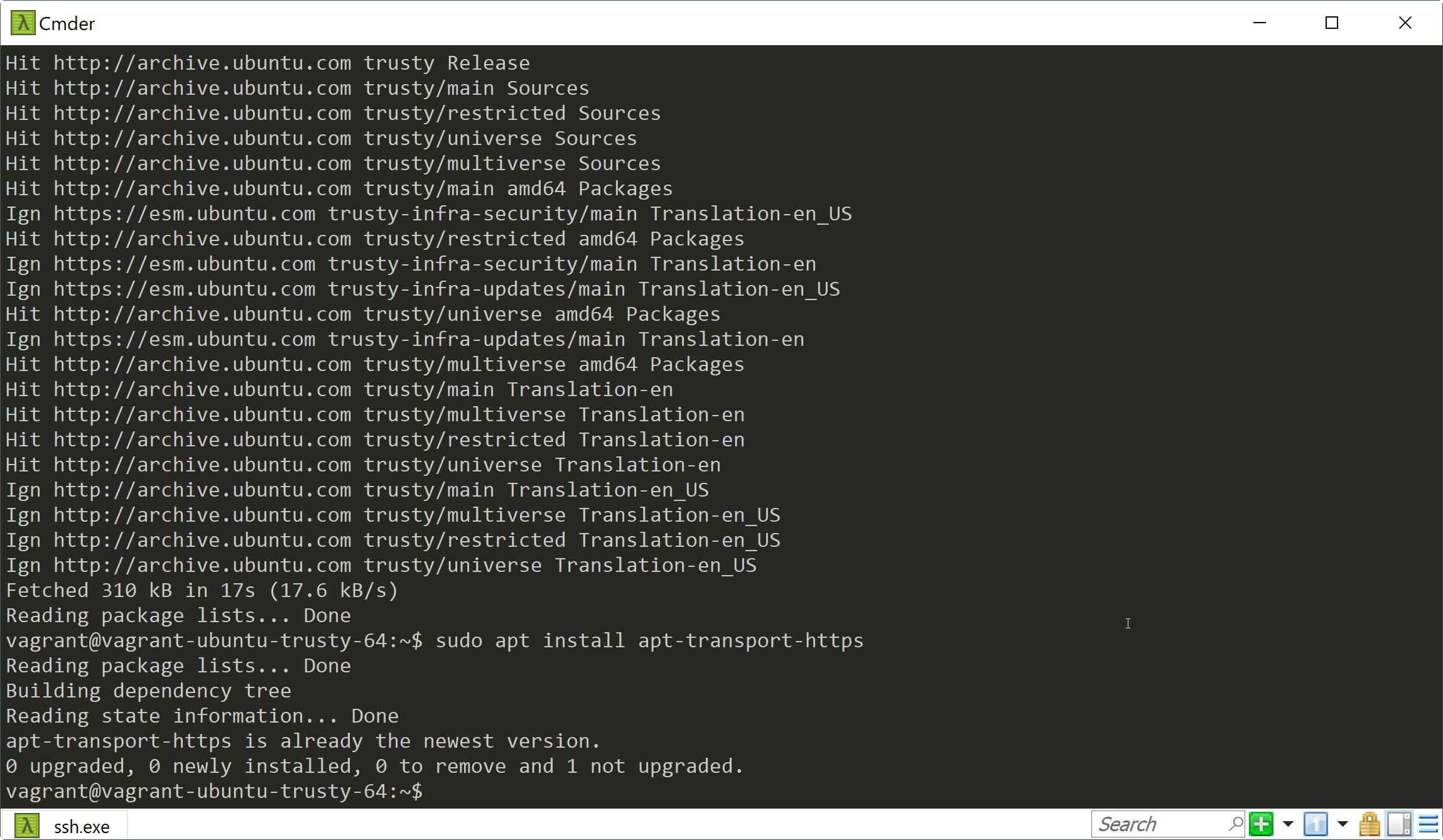
tee -a /etc/apt/sources.list.d/elastic-7.x.list

sudo echo "deb https://artifacts.elastic.co/packages/oss-7.x/apt stable main" |

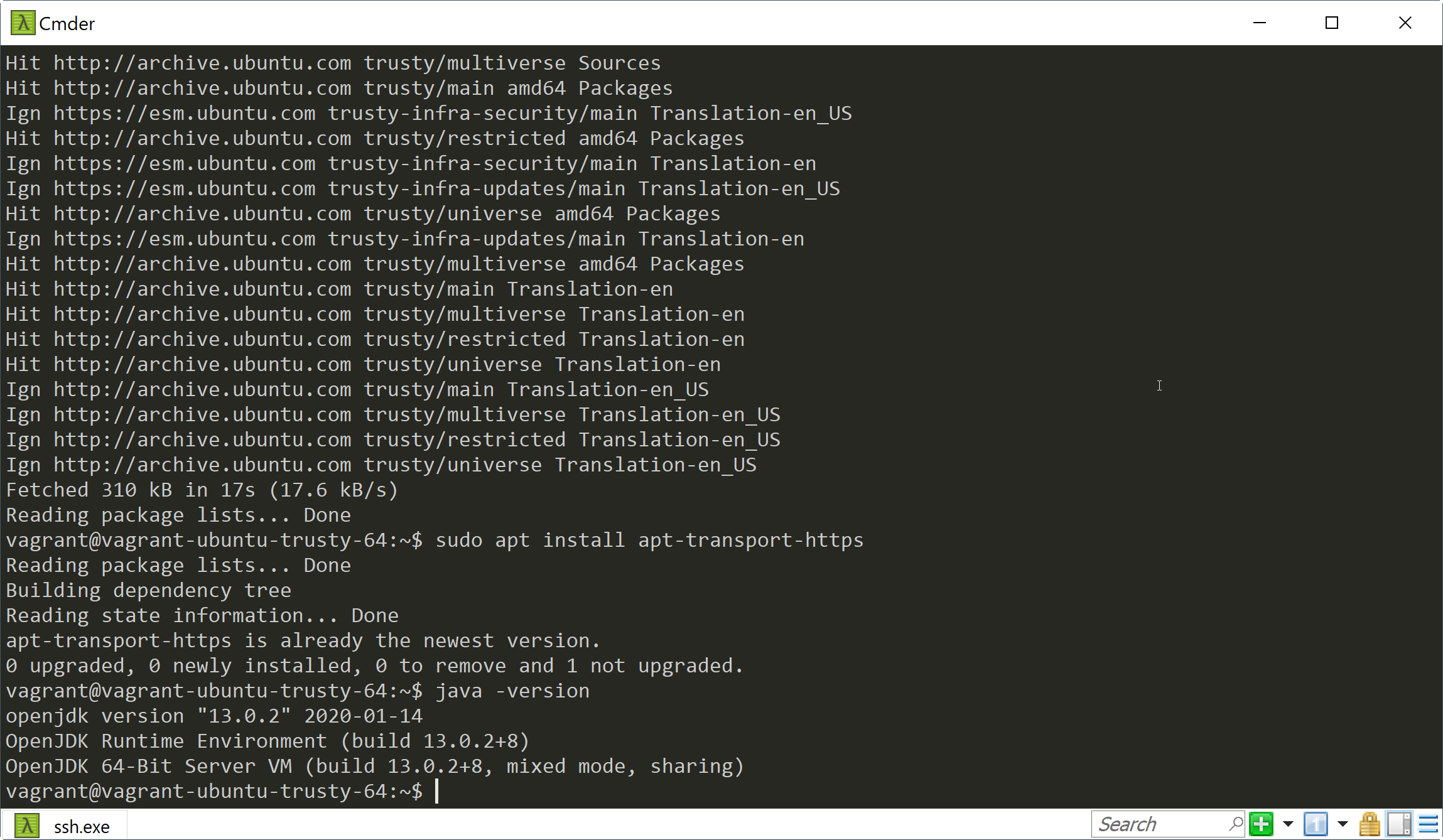
sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list



sudo apt update

sudo apt install apt-transport-https

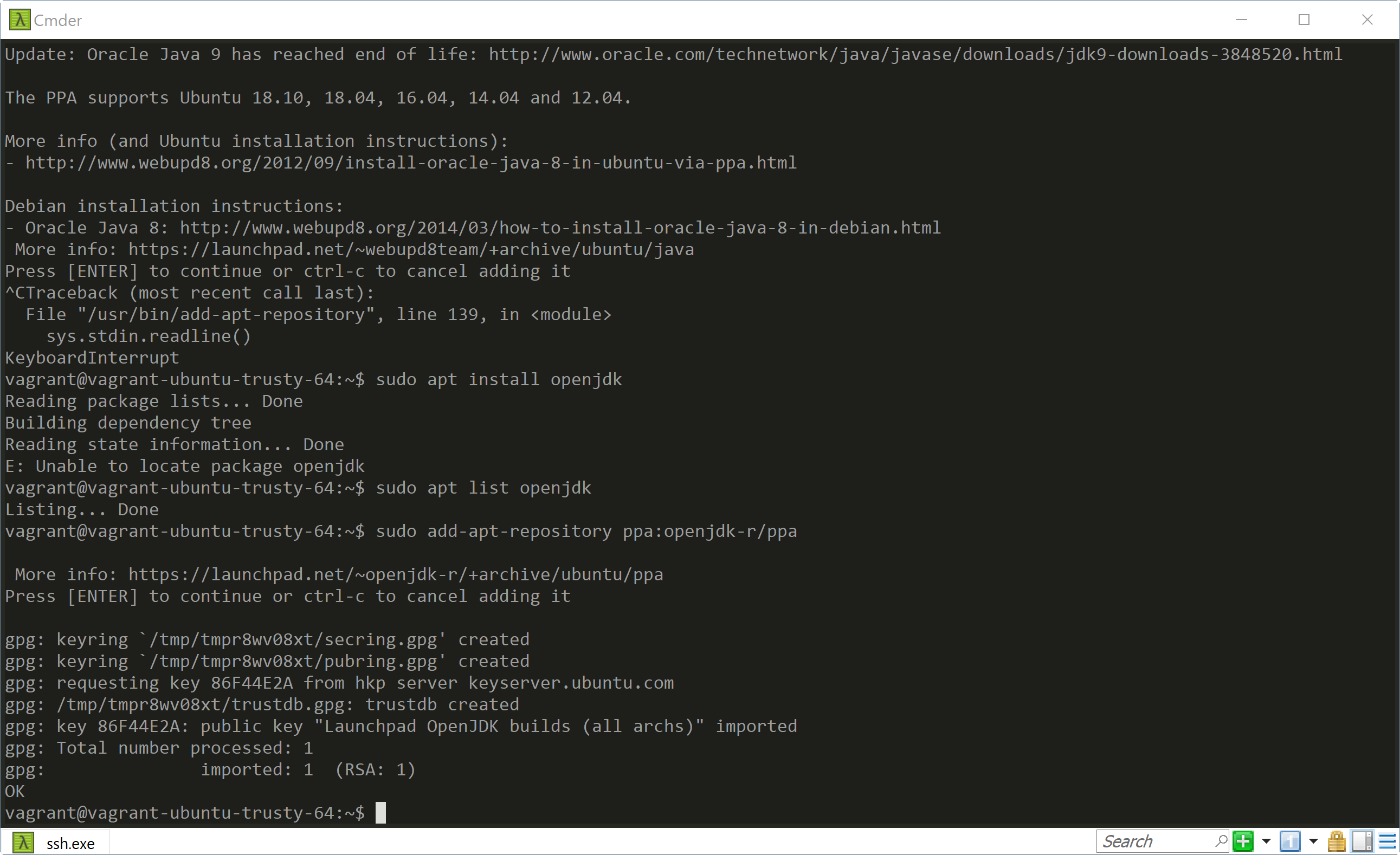
Verify java if installed

Enter **java -version**

If not exist, install java

sudo add-apt-repository ppa:openjdk-r/ppa

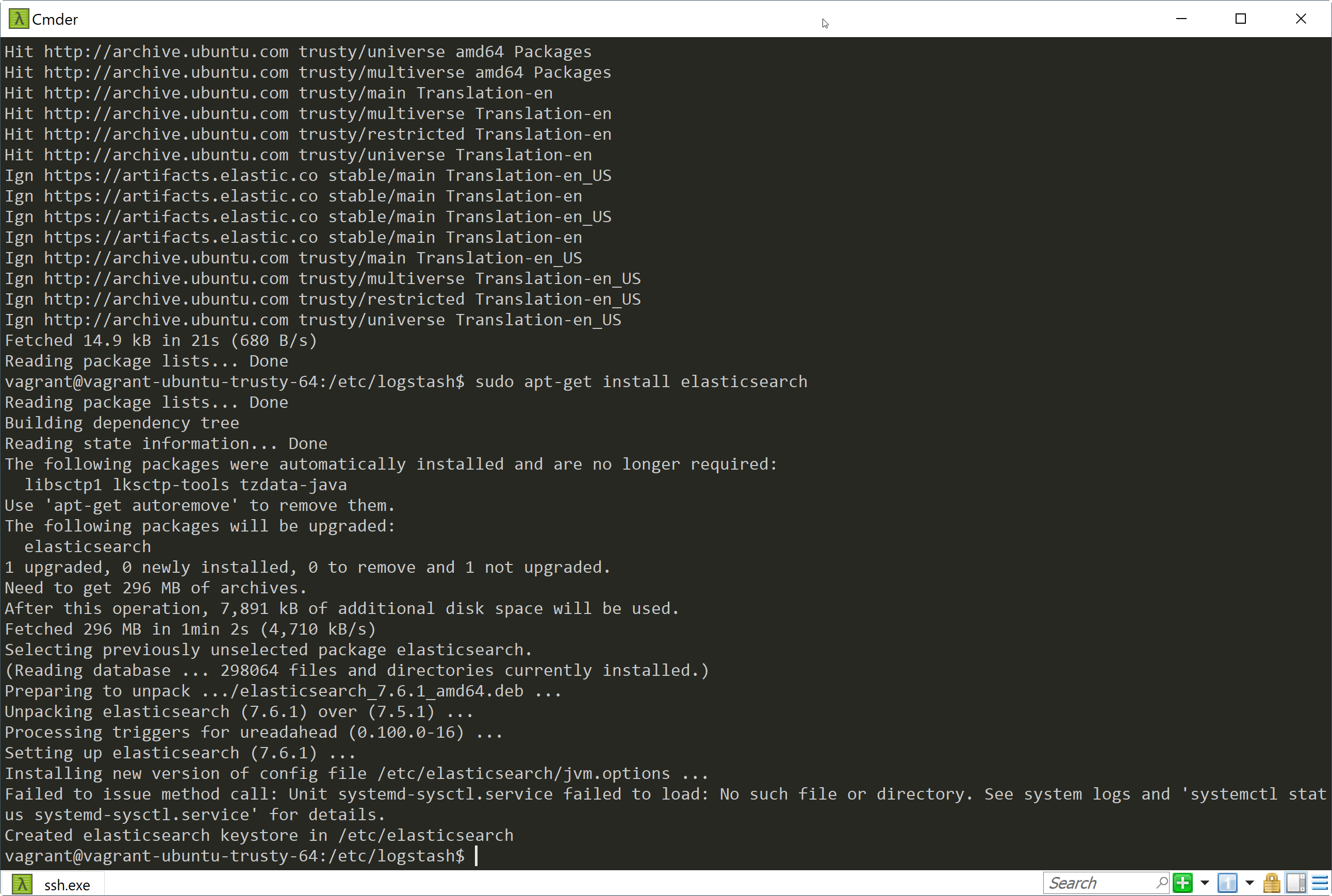
sudo apt-get update

sudo apt-get install openjdk-8-jdk

### Install Elasticsearch

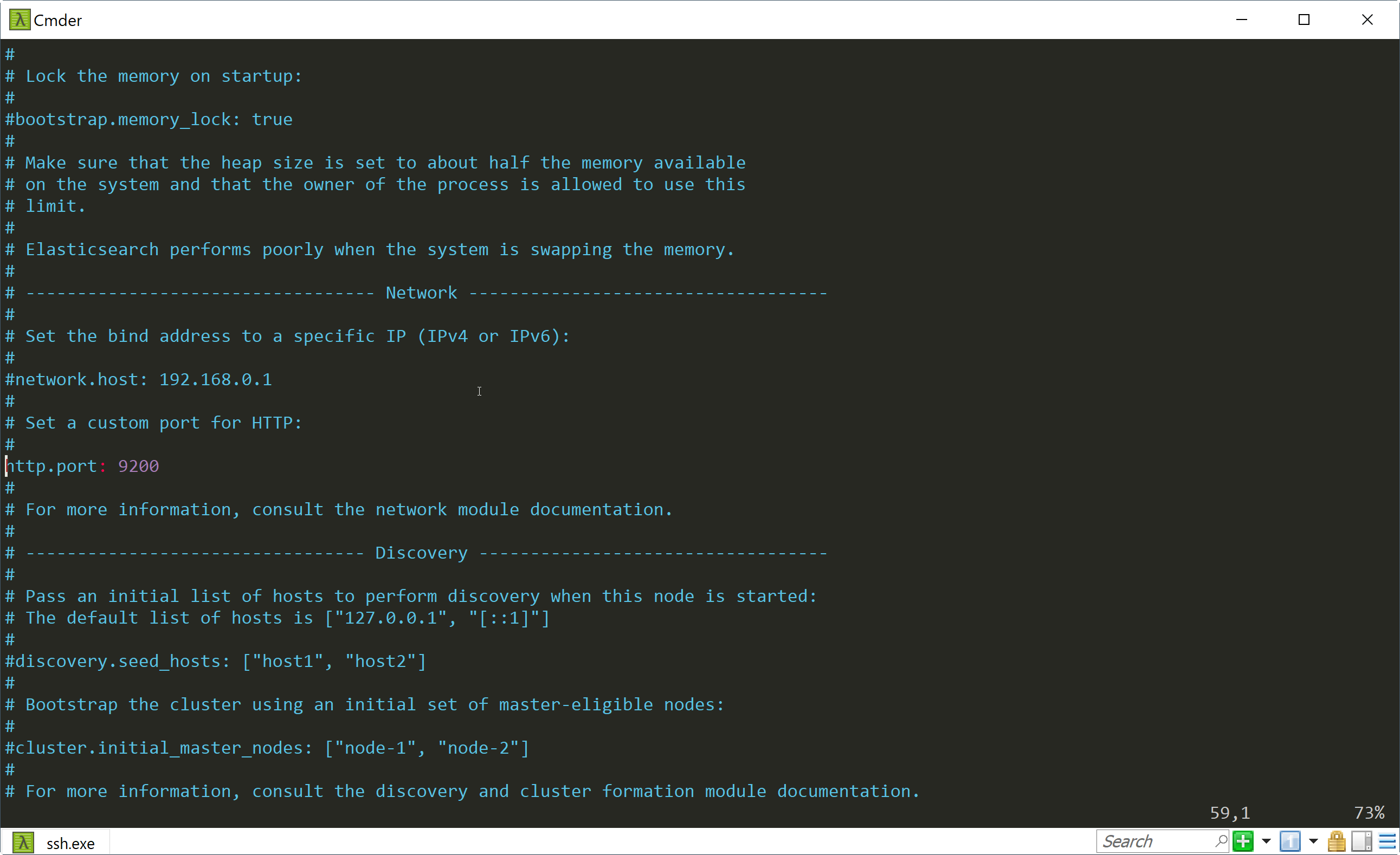
sudo apt-get update

sudo apt-get install elasticsearch



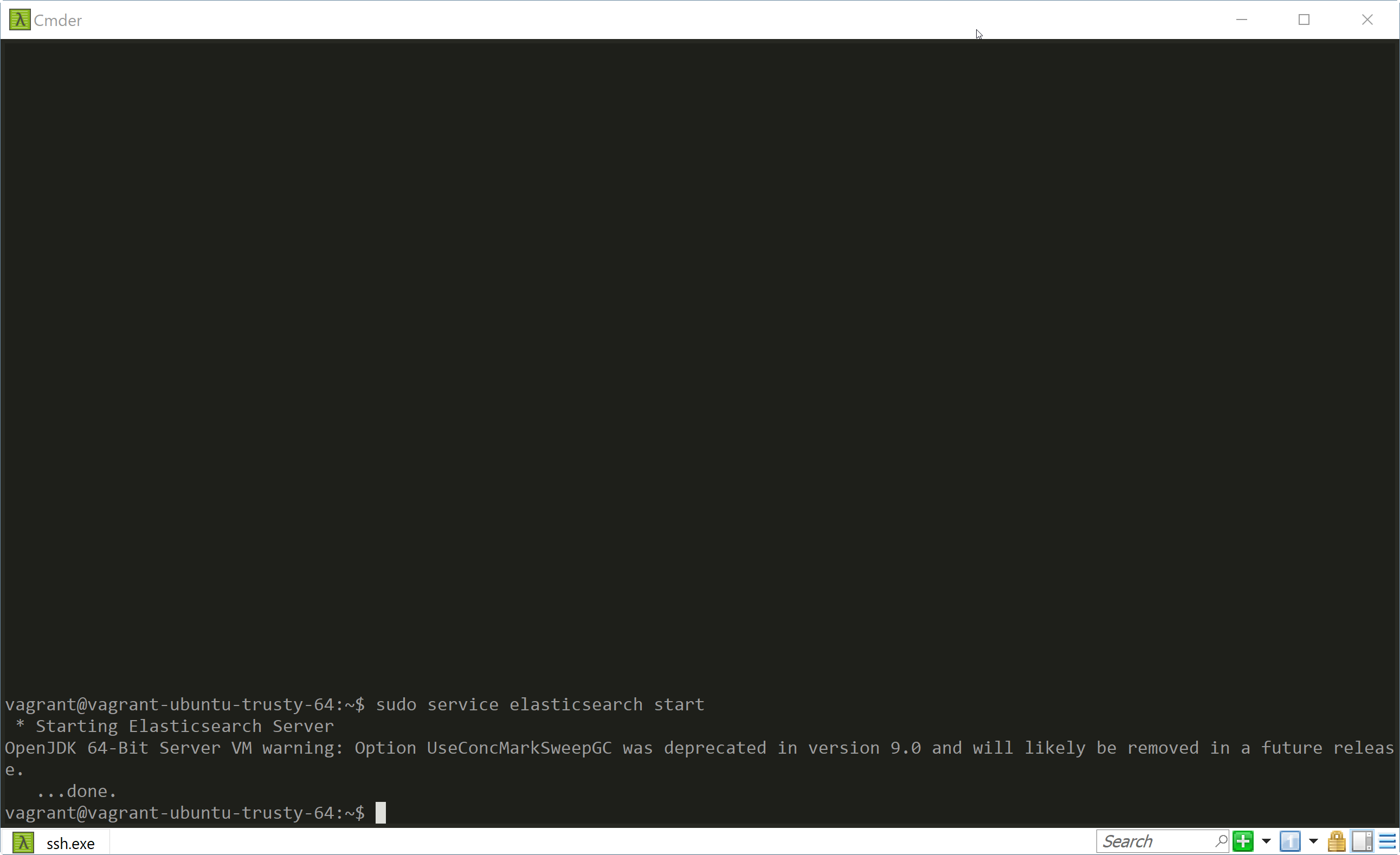
### Configure Elasticsearch

sudo vi /etc/elasticsearch/elasticsearch.yml

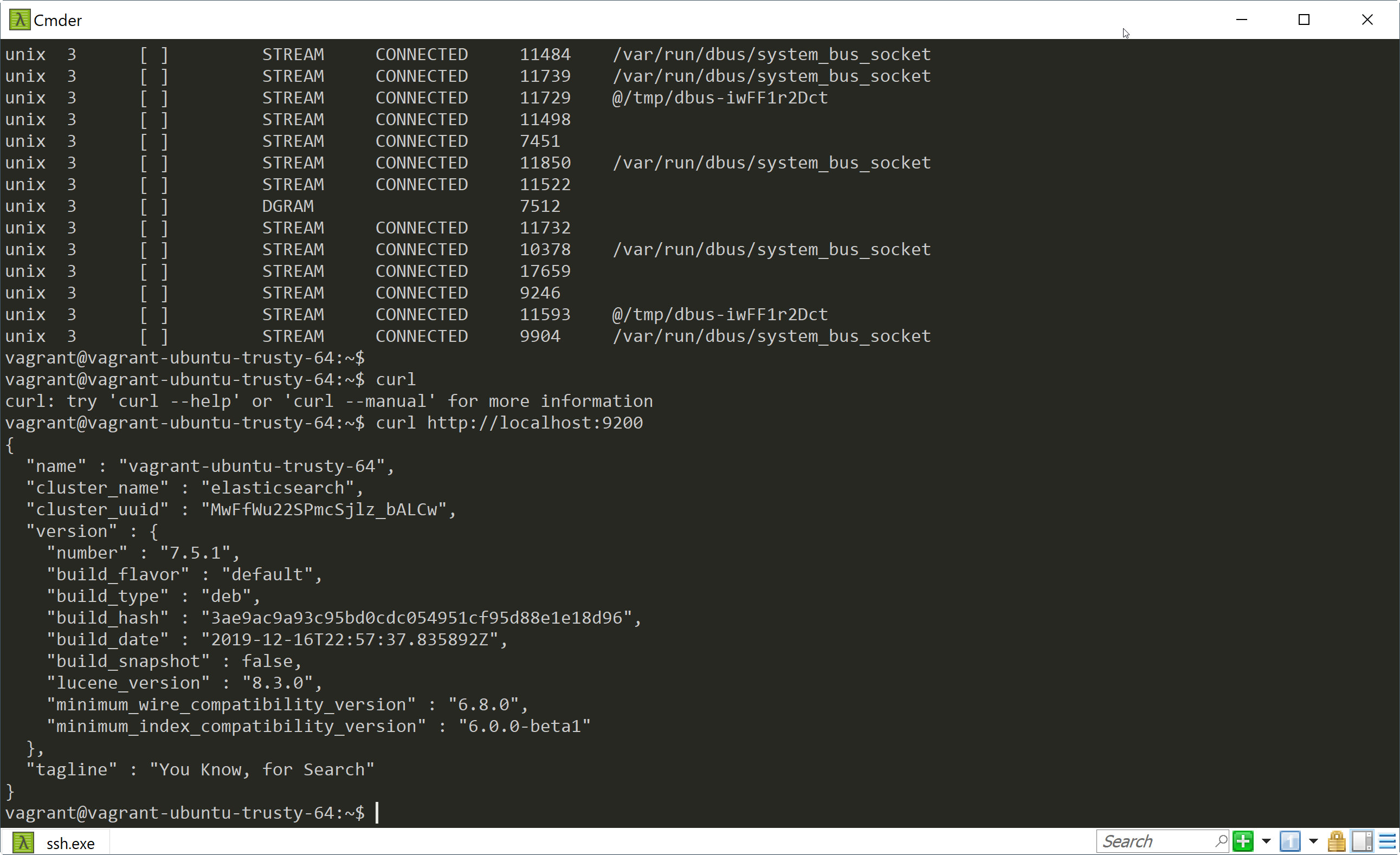
Go to the network Module and Enable the port to 9200

Start Elasticsearch service

Sudo service elasticsearch start



Verify Elasticsearch is up and run

Curl http://localhost:9200

## LOGSTASH INSTALLATION (optional)

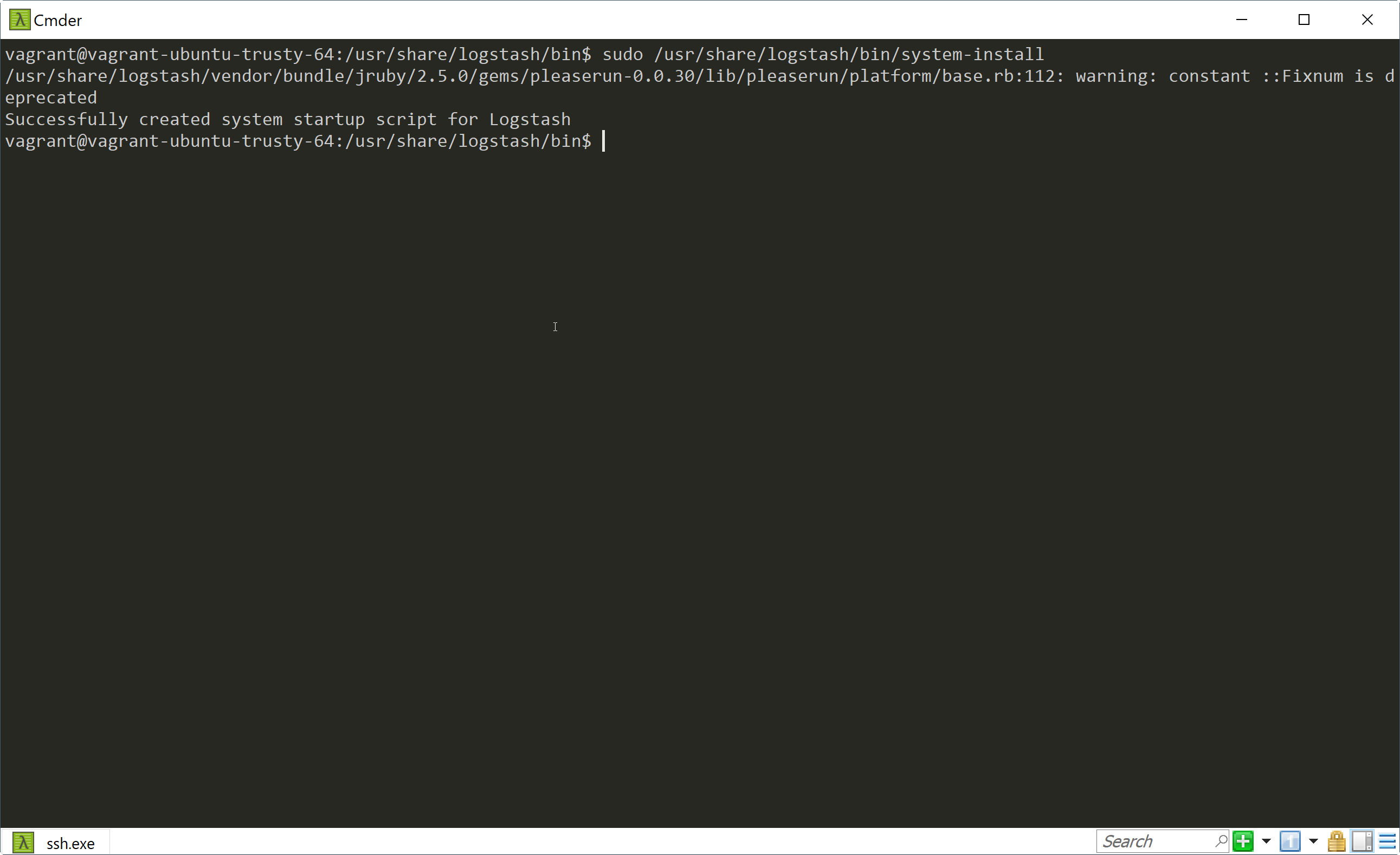
### Get Elastic repository package

### Logstash installation

sudo apt-get update

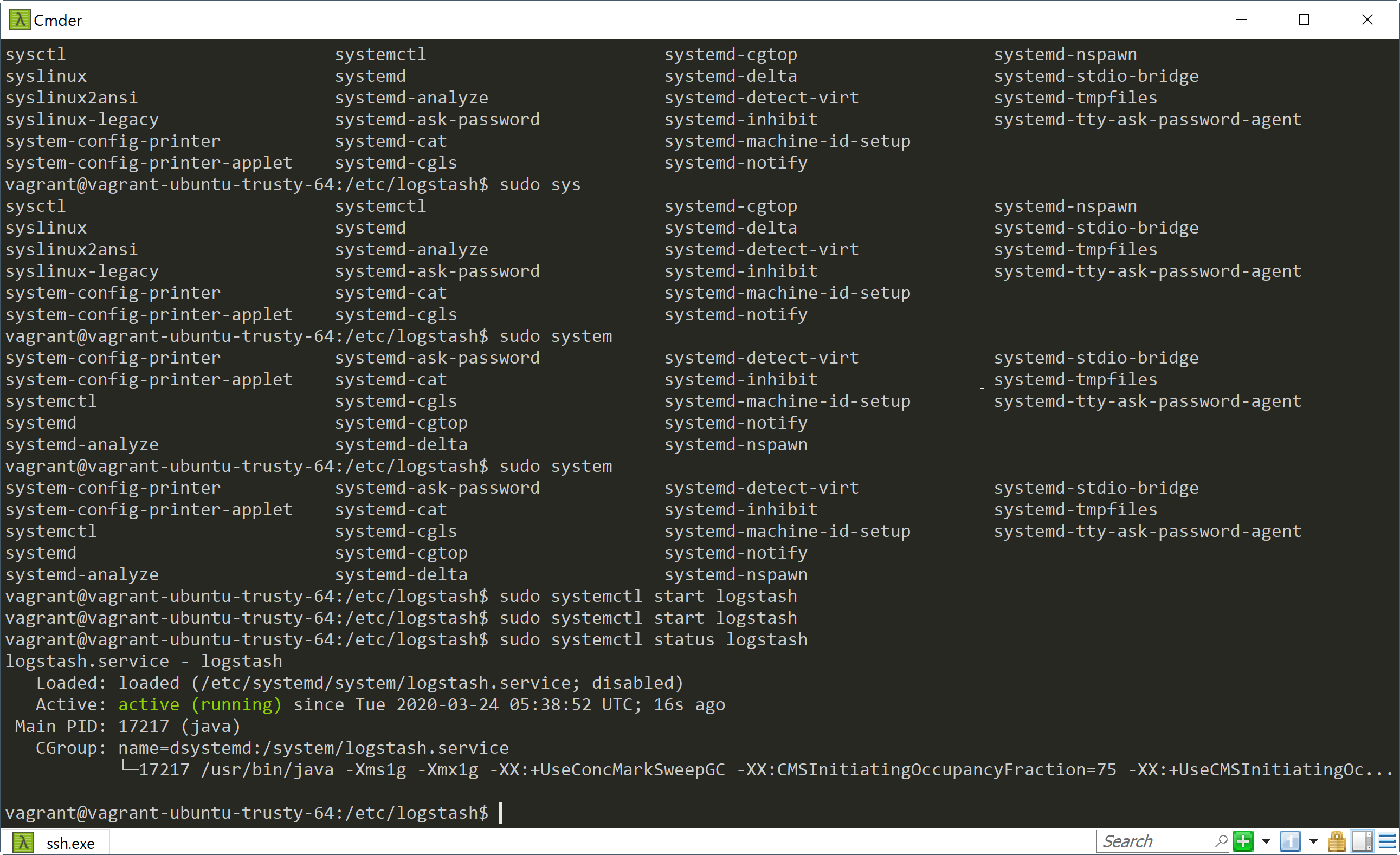
sudo apt-get install logstash

Service installations

sudo /usr/share/logstash/bin/system-install

Start Logstash

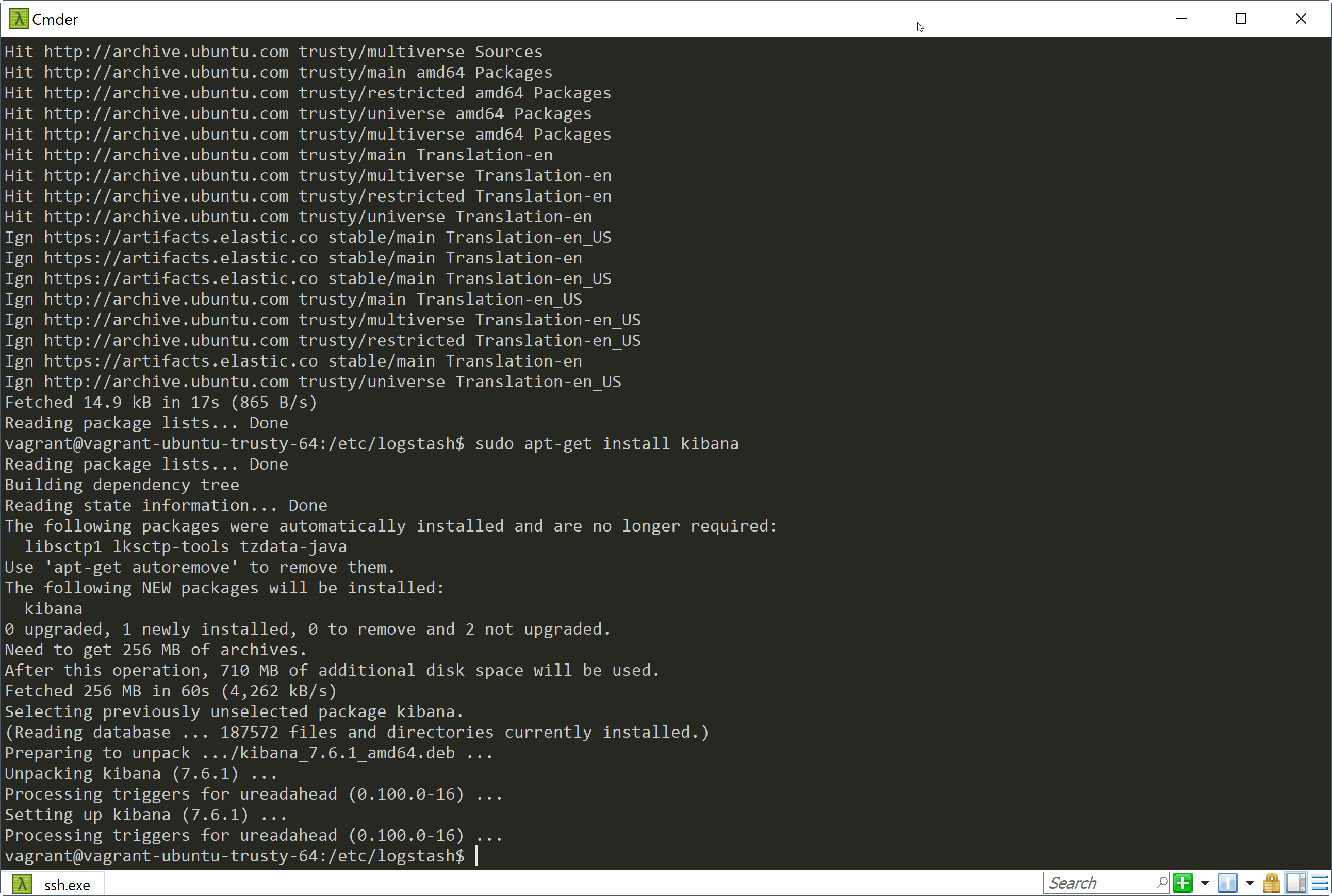
sudo systemctl status logstash



## KIBANA INSTALL (optional)

### KIBANA installation

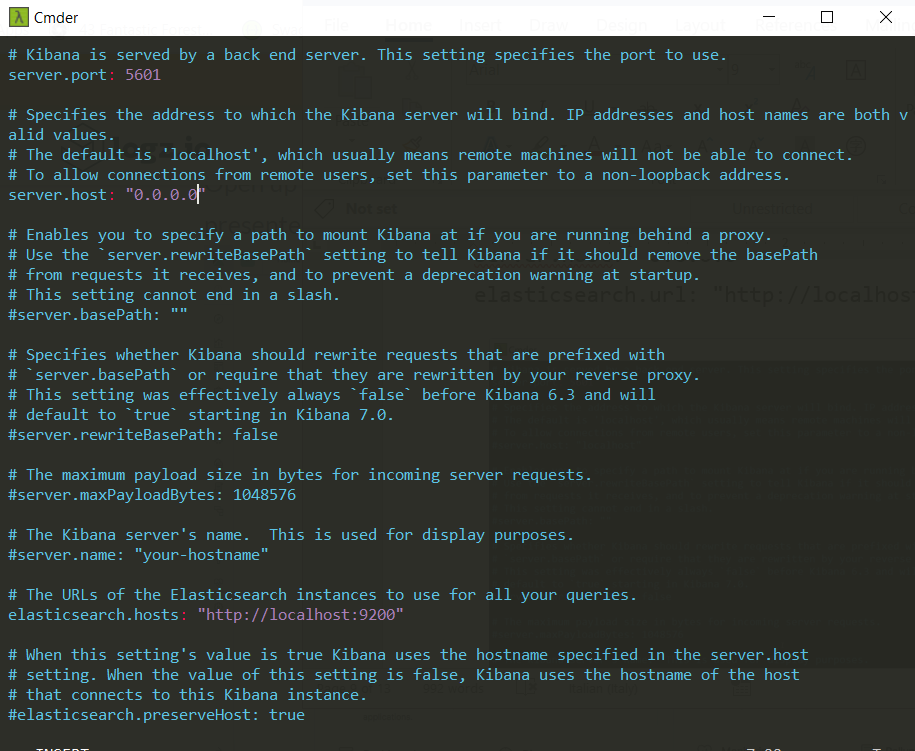
sudo apt-get install kibana



Edit /etc/kibana/kibana.yml and define following parameters to allow Kibana bind to all IP addresses

server.port: 5601

server.host: “0.0.0.0”

elasticsearch.url: "http://localhost:9200"

Start service

sudo service kibana start

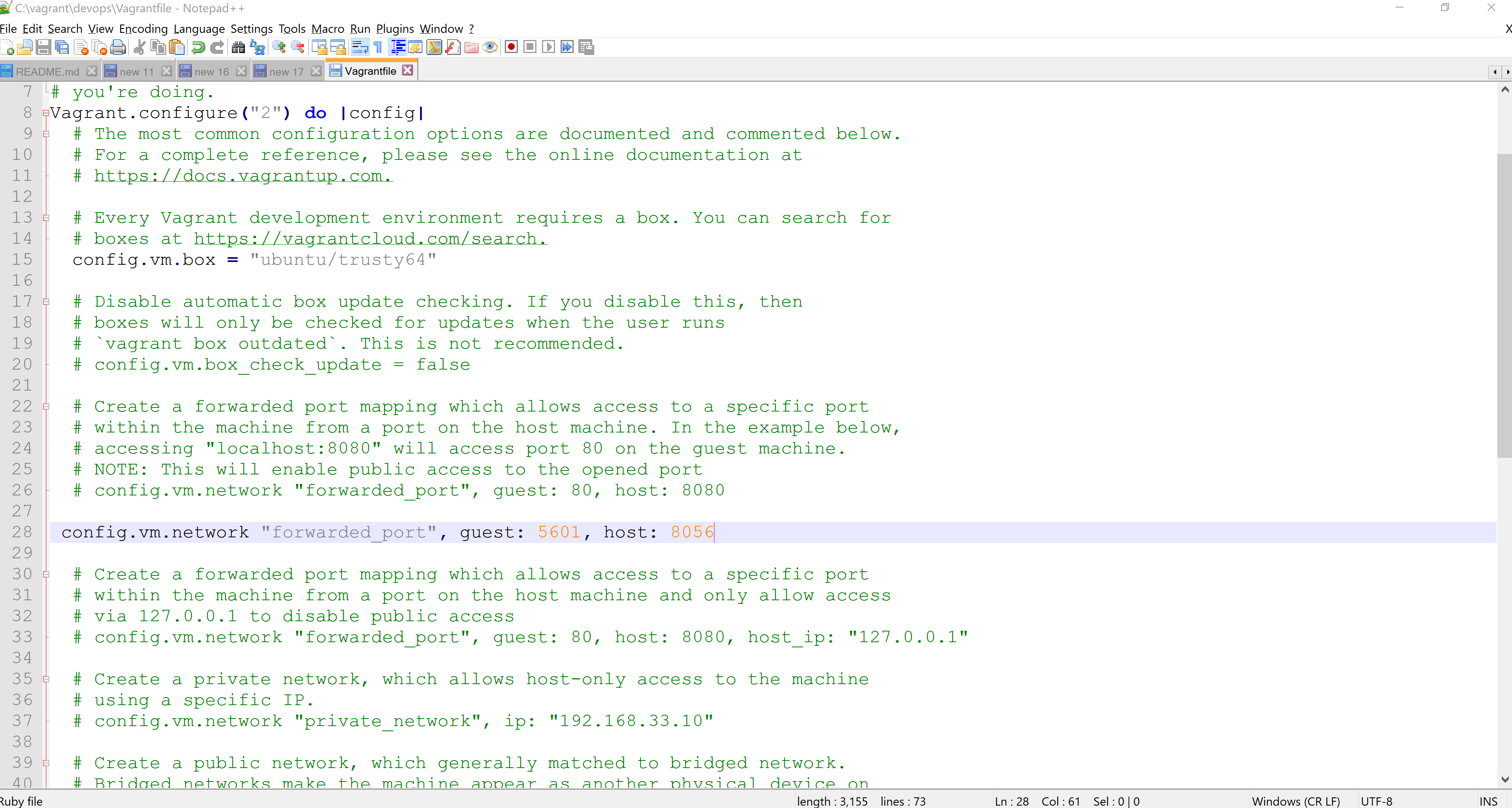
## Port Mapping

Create a port forward in Vagrant so that you can access the guest port from your laptop

Edit the file vagrantfile

Add new line and save the file

config.vm.network "forwarded\_port", guest: 5601, host: 8056



### Restart the Vagrant box

After configure, we needed to reload ther vagrant box and view then view the new port mapping

vagrant reload

vagrant port

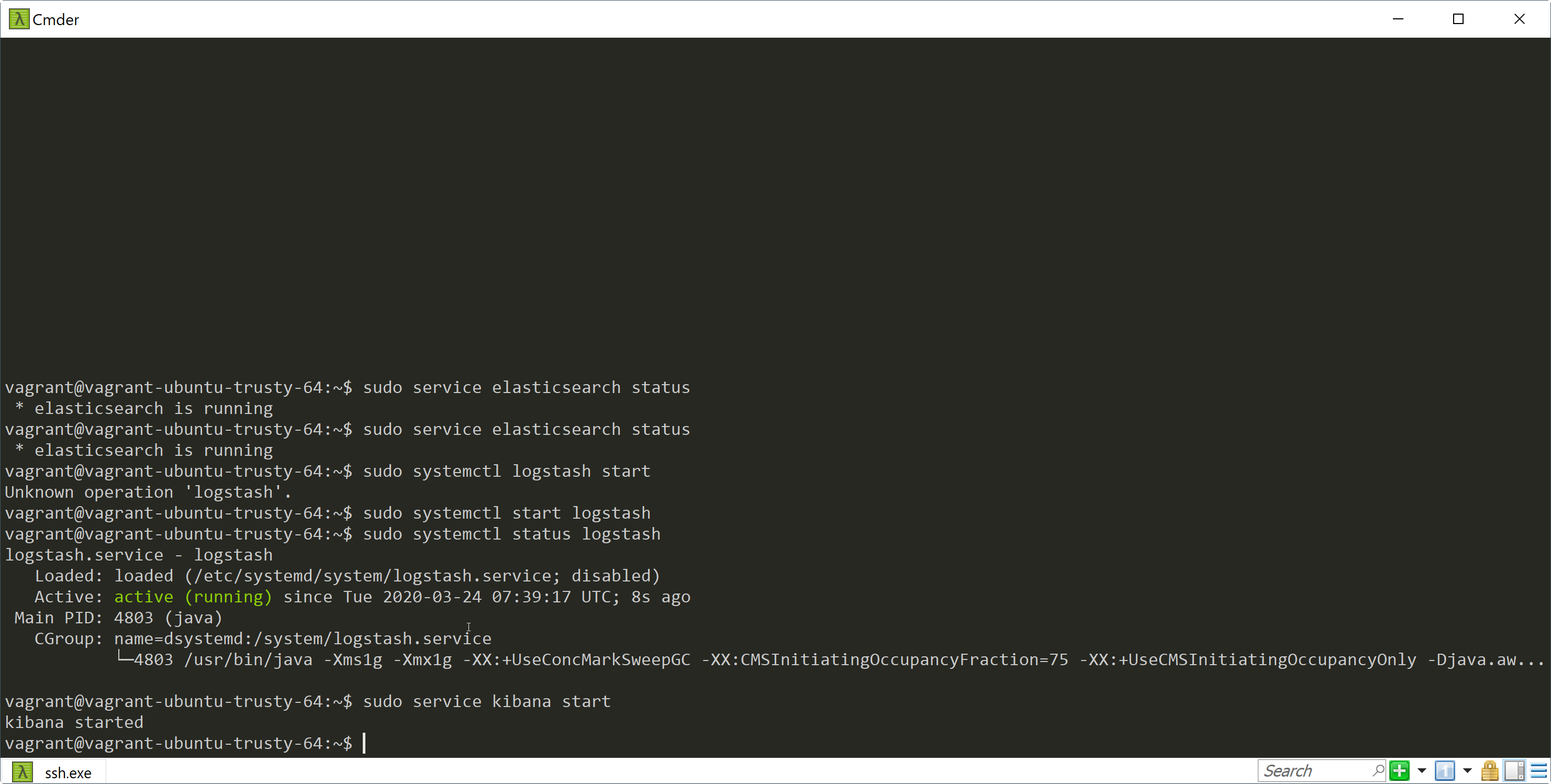
After login to the box through ssh, we can restart all ELK services

vagrant ssh

sudo service elasticsearch start

sudo systemctl start logstash

sudo service kibana start

p

Verify Kibana started and ready to use

sudo tail -f /varlog/kibana/kibana.stdout

# Start Vagrant box

1. Copy api-training.box to your vagrant folder
2. Start cmder.exe
3. Execute vagrant init
4. Modify VagrantFile to add config.vml.box\_url with your training.box path. Assume api-training.box and VagrantFile is in the same folder

config.vm.box\_url = "api-training.box"

1. Execute vagrant up
2. Execute vagrant ssh
3. Start elasticsearch service, logstash kibana service

sudo service elasticsearch start

sudo systemctl start logstash

sudo service kibana start

# Kibana Searching

Kibana querying is an art unto itself, and there are various methods for performing searches on your data. This step will describe some of the most common search methods as well as some tips and best practices that should be memorized for optimized user experience.

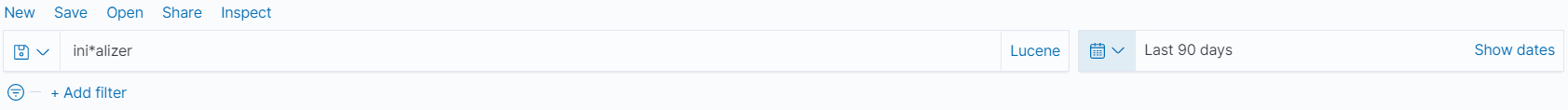
### Access Kibana

http://localhost:8056

### Free-Text Search

Free text search works within all fields—including the \_source  field, which includes all the other fields. If no specific field is indicated in the search, the search will be done on all of the fields that are being analyzed.

In the search field at the top of the Discover page, run these searches and examine the result **(set the time parameter on the top right of the dashboard to the past 90 days to capture more data. Search Option is Lucene**):



Run these searches and examine the result

* Inventories
* inventories
* inven\*
* “initializer bean”
* initializer bean

#### **Tips and Gotchas**

1. Text searches are not case sensitive. This means that " inventories" and "Inventories" will return the same results. When you put the text within double quotes (""), you are looking for an exact match, which means that the exact string must match what is inside the double quotes. This is why [initializer bean] and ["initializer bean "] will return different results
2. Kibana wildcard searches—you can use the wildcard symbols [\*] or [?] in searches. [\*] means any number of characters, and [?] means only one character

### Field-Level Searches

Another common search in Kibana is field-level queries, sued for searching for data inside specific fields. To use this type of search that, you need to use the following format:

<fieldname>:search

As before, run the following searches to see what you get (some will purposely return no results):

level:debug

level:info

logger\_name:com.exercise.orders.api.impl.OrdersApiDelegateImpl name:Chr\*

message:"httpStatus: 200"

\_exists\_:message

\_exists\_:date

#### Tips and Gotchas

* Field-level searches depend on the type of field. (All fields are not analyzed by default, which means that searches are case-sensitive and cannot use wildcard searches. The reason we save all of the fields as "not analyzed" is to save space in the index because the data is also duplicated in an analyzed field called  \_source )
* Using the \_exists\_   prefix for a field will search the documents to see if the field exists.

### Logical Statements

* You can use logical statements in searches in these ways:
* order AND inventory
* order OR inventory
* (order and inventory) OR services
* -order
* !order
* +order
* NOT order

#### **Tips and gotchas**

* You need to make sure that you use the proper format such as capital letters to define logical terms like AND or OR
* You can use parentheses to define complex, logical statements
* You can use -,! and NOT to define negative terms

### Kibana Special Characters

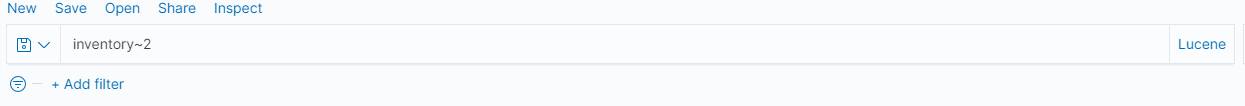
All special characters need to be properly escaped. The following is a list of all available special characters:

+ - && || ! ( ) { } [ ] ^ " ~ \* ? : \

### Proximity searches

Proximity searches are an advanced feature of Kibana that takes advantage of the Lucene query language.

[inventory~2] means a search for all the terms that are within two changes from [inventory]. This means that all category will be matched.



#### Tips and gotchas

Proximity searches use a lot of system resources and often trigger internal circuit breakers in Elasticsearch. If you try something such as [inv~10], it is likely not to return any results due to the amount of memory used to perform this specific search. 

### Kibana Filtering Dialog

To assist users in searches, recent versions include a filtering dialog that allows easier forming of Kibana search syntax.

To use the dialog, simply click the Add a filter + button under the search box and begin experimenting with the conditionals.

Power users can also enter Elasticsearch queries using the Query DSL.

END of Document