Задача 4. Ansible

Подготовка

Ссылки на репозитории:

1. ab-haproxy - https://github.com/Fittske/ab-haproxy
2. ab-logstash - https://github.com/Fittske/ab-logstash
3. ab-webui - https://github.com/Fittske/ab-webui

Vagrant

Подготовим Vagrantfile для создания небходимых виртуальных машин на Ubuntu 18.04.

```
Vagrant.configure("2") do |config|

  config.vm.define "vm1" do |vm1|
      vm1.vm.box = "bento/ubuntu-18.04"
      vm1.vm.network "forwarded_port", guest: 80, host: 8080
      vm1.vm.network "public_network", ip: "192.168.1.10"

      vm1.vm.provider "virtualbox" do |v|
          v.cpus = 2
          v.memory = 4096
      end

      vm1.vm.provision :ansible do |ansible|
          ansible.playbook = "./ab-haproxy/playbook.yml"
      end
  end

  config.vm.define "vm2" do |vm2|
      vm2.vm.box = "bento/ubuntu-18.04"
      vm2.vm.network "forwarded_port", guest: 81, host: 8081
      vm2.vm.network "forwarded_port", guest: 9600, host: 9600 # Logtash
      vm2.vm.network "forwarded_port", guest: 9200, host: 9200 # Elasticsearch
      vm2.vm.network "public_network", ip: "192.168.1.11"

      vm2.vm.provider "virtualbox" do |v|
          v.cpus = 2
          v.memory = 4096
      end

      vm2.vm.provision :ansible do |ansible|
          ansible.playbook = "./ab-logstash/playbook.yml"
      end
  end

  config.vm.define "vm3" do |vm3|
      vm3.vm.box = "bento/ubuntu-18.04"
      vm3.vm.network "forwarded_port", guest: 82, host: 8082
      vm3.vm.network "forwarded_port", guest: 5601, host: 5601 # Kibana
      vm3.vm.network "public_network", ip: "192.168.1.12"

      vm3.vm.provider "virtualbox" do |v|
          v.cpus = 2
          v.memory = 4096
      end

      vm3.vm.provision :ansible do |ansible|
          ansible.playbook = "./ab-webui/playbook.yml"
      end
  end
end
```

Виртуальная машина ab-haproxy

Создадим Ansible плейбук для данной ВМ и подготовим соответствующие роли.
`nano playbook.yml`

```
- name: Initial config
  hosts: all
  become: yes
  roles:
    - apt
    - ntp
    - monit
    - haproxy
```

APT

Роль предназначена для добавления необходимых пакетов и обновления из репозиториев установленных по умолчанию пакетов.

Инициируем роль:
`ansible galaxy init apt`
Результат:

```
ruslan@ruslan-Z690-UD:~/IT-2023/task_4/ab-haproxy/roles/apt$ ll
итого 36
drwxr-xr-x 8 root   root   4096 ноя 13 23:19 ./
drwxrwxr-x 3 ruslan ruslan 4096 ноя 13 23:19 ../
drwxr-xr-x 2 root   root   4096 ноя 13 23:19 defaults/
drwxr-xr-x 2 root   root   4096 ноя 13 23:19 handlers/
drwxr-xr-x 2 root   root   4096 ноя 13 23:19 meta/
-rw-r--r-- 1 root   root   1328 ноя 13 23:19 README.md
drwxr-xr-x 2 root   root   4096 ноя 13 23:19 tasks/
drwxr-xr-x 2 root   root   4096 ноя 13 23:19 tests/
drwxr-xr-x 2 root   root   4096 ноя 13 23:19 vars/
ruslan@ruslan-Z690-UD:~/IT-2023/task_4/ab-haproxy/roles/apt$
```

DEFAULTS

Раздел defaults позволяет устанавливать переменные по умолчанию для включенных или зависимых ролей.

Отредактируем файл `main.yml` следующим образом:

```
---
# defaults file for apt
# The upgrade mode to use. See
https://docs.ansible.com/ansible/latest/modules/apt_module.html for available
options.
apt_upgrade: "dist"
```

TASKS

Раздел tasks содержит один или несколько файлов с задачами. Эти задачи могут напрямую ссылаться на файлы и шаблоны, содержащиеся в соответствующих каталогах внутри роли, без необходимости указывать полный путь к файлу.

Отредактируем файл `main.yml` следующим образом:

```
---
# tasks file for apt
- name: Update package cache & dist-upgrade packages
  apt:
    upgrade: "{{ apt_upgrade }}"
    update_cache: yes
    cache_valid_time: 3600
    force_apt_get: true
    autoclean: true

- name: Clean apt.
  apt:
    autoclean: yes

- name: Update package cache after upgrade.
  apt:
    update_cache: yes
```

NTP

Роль предназначена для обновления времени и настройки синхронизации времени по cron 1 раз в сутки с любого общедоступного сервера.

Инициируем роль:
`ansible galaxy init ntp`

DEFAULTS

Укажем переменную для общедоступного сервера времени.

Отредактируем файл `main.yml` следующим образом:

```
---
# defaults file for ntp
ntpserver: ntp.ultracoder.org
```

TASKS

Отредактируем файл `main.yml` следующим образом:

```
---
# tasks file for ntp
- name: install ntpdate
  apt:
    pkg: ntpdate
    state: present

- name: time synchronization
  ansible.builtin.shell:
    cmd: sudo ntpdate -u {{ ntpserver }}

- name: Setup CRON job for time synchronization
  ansible.builtin.cron:
    name: "Synchronize time"
    minute: "*"
    weekday: "*"
    hour: "24"
    job: "sudo ntpdate -u {{ ntpserver }}"
    user: root
    cron_file: time_update.cron
```

MONIT

Роль предназначена для установки и настройки monit для само-мониторинга виртуальной машины.

Внутри роли добавим каталог templates:
`mkdir templates`

TEMPLATES

Раздел предназначен для шаблонов, которые будут использоваться для генерирации файлов на удалённом хосте.

Создадим файл конфигурации для Monit и укажем правила перезагрузки Haproxy.

`haproxy.conf`:

```
check process haproxy with pidfile /var/run/haproxy.pid
group root
start program = "/etc/init.d/haproxy start"
stop program = "/etc/init.d/haproxy stop"

if failed host 127.0.0.1 port 80 then restart

if 5 restarts within 5 cycles then timeout
```

TASKS

`main.yml`:

```
---
# tasks file for monit
- name: install monit
  apt:
    pkg: monit
    state: present

- name: start monit
  service:
    name: monit
    state: started

- name: configure monit
  template:
    src: haproxy.conf
    dest: /etc/monit/conf.d/haproxy
    mode: 0644

- name: restart monit
  service:
    name: monit
    state: restarted
```

HAPROXY

Роль предназначена для установки и настройки haproxy в качестве балансировщика для веб-иинтерфейса.

DEFAULTS

Укажем переменные для конфигурации haproxy:

1. Разрешим бинд нелокальных ip адресов
2. Установим настройки таймаута

`main.yml`:

```
---
# defaults file for haproxy
haproxy_allow_bind_non_local_ip: true

haproxy_connect_timeout: 5000

haproxy_client_timeout: 50000

haproxy_server_timeout: 50000
```

HANDLERS

Раздел предназначен для обработчиков.

Добавим обработчик для перезагрузки haproxy.

`main.yml`:

```
---
# handlers file for haproxy
- name: Reload haproxy
  systemd:
    name: "haproxy"
    state: reloaded
```

TEMPLATES

Добавим шаблон для конфигурации haproxy.

Настроим проксирование на ВМ, где будет работать kibana.

`haproxy.cfg`:

```
global
    log /dev/log  local0
    user haproxy
    group haproxy
    daemon

defaults
    log global
    mode  http
    option  httplog
    option  dontlognull
    timeout connect {{ haproxy_connect_timeout }}
    timeout client {{ haproxy_client_timeout }}
    timeout server {{ haproxy_server_timeout }}

frontend web
  bind *:80
  default_backend kibana

backend kibana
  mode http
  server kibana1 192.168.1.12:5601 check
```

Виртуальная машина ab-logstash

Создадим Ansible плейбук для данной ВМ и подготовим соответствующие роли.
`nano playbook.yml`

```
- name: Initial config
  hosts: all
  become: yes
  roles:
    - apt
    - ntp
    - monit
    - elasticsearch
    - logstash
```

Роли APT, NTP, MONIT по настройкам аналогичны ВМ ab-haproxy, за исключением правил для monit (перезапуск настроен для elasticsearch).

LOGSTASH

Роль предназначена для установки logstash, который собирает данные из различных источников, преобразовывает их и отправляет в нужное место назначения.

HANDLERS

Добавим обработчик для перезапуска.

`main.yml`:

```
---
# handlers file for logstash
- name: restart logstash
  service:
     name: logstash
     state: restarted
```

TEMPLATES

Добавим шаблон для конфигурации logstash.

Настроим его таким образом, чтобы logstash мог получать данные от rsyslog на порту 10514 и отправлял их в elasticsearch.

`ab-logstash.conf`:

```
# This input block will listen on port 10514 for logs to come in.
# host should be an IP on the Logstash server.

input {
    udp {
      host => "192.168.1.11"
      port => 10514
      codec => "json"
      type => "rsyslog"
    }
}

filter { }

# This output block will send all events of type "rsyslog" to Elasticsearch at
the configured
# host and port into daily indices of the pattern, "rsyslog-YYYY.MM.DD"
output {
    if [type] == "rsyslog" {
        elasticsearch {
            hosts => [ "192.168.1.11:9200" ]
        }
    }
}
```

TASKS

Добавим задачи для установки logstash. В качестве репозитория будем использовать зеркало Яндекса.

`main.yml`:

```
---
# tasks file for logstash

- name: Install apt-transport-https
  apt:
    pkg: apt-transport-https
    state: latest

- name: Save repo
  ansible.builtin.shell: echo "deb [trusted=yes]
https://mirror.yandex.ru/mirrors/elastic/8/ stable main" | sudo tee
/etc/apt/sources.list.d/elastic-8.x.list

- name: Install Logstash
  apt:
    name: logstash
    state: present
    update_cache: true

- name: Setup conf
  template:
    dest: /etc/logstash/conf.d/ab-logstash.conf
    src: ab-logstash.conf
    force: yes
    mode: 0644
  become: yes
  notify:
    - restart logstash
```

ELASTICSEARCH

Роль предназначена для установки и настройки elasticsearch, в котором будут храниться данные логов с возможностью поиска.

HANDLERS

Добавим обработчки для перезапуска.

`main.yml`:

```
---
# handlers file for elasticsearch
- name: restart elasticsearch
  service:
    name: elasticsearch
    state: restarted
```

TEMPLATES

Создадим шаблон конфигурации для elastic:

1. Укажем IP адрес для привязки.
2. Укажем список хостов для обнаружения.
3. Укажем порт.
4. Отключим опции безопасности.

`elasticsearch.yml`:

```
...
network.host: 192.168.1.11
http.port: 9200
discovery.seed_hosts: ["127.0.0.1", "[::1]"]
xpack.security.enabled: false
...
```

TASKS

Добавим задачи для установки и конфигурации elasticsearch.

`main.yml`:

```
---
# tasks file for elasticsearch
- name: Install apt-transport-https
  apt:
    pkg: apt-transport-https
    state: latest

- name: Save repo
  ansible.builtin.shell: echo "deb [trusted=yes]
https://mirror.yandex.ru/mirrors/elastic/8/ stable main" | sudo tee
/etc/apt/sources.list.d/elastic-8.x.list

- name: Install elasticsearch
  become: yes
  apt:
    name: elasticsearch
    state: present
    update_cache: true

- name: Running elastic with systemd
  ansible.builtin.shell: sudo /bin/systemctl daemon-reload && sudo
/bin/systemctl enable elasticsearch.service

- name: Ensure service is enabled
  systemd:
    name: "elasticsearch"
    enabled: true
    state: started

- name: Setup yml
  template:
    src: elasticsearch.yml
    dest: /etc/elasticsearch/elasticsearch.yml
  notify:
    - restart elasticsearch

- name: Reset security settings
  ansible.builtin.shell: sudo rm /etc/elasticsearch/elasticsearch.keystore &&
sudo /usr/share/elasticsearch/bin/elasticsearch-keystore create
  notify:
    - restart elasticsearch
```

Виртуальная машина ab-webui

Создадим Ansible плейбук для данной ВМ и подготовим соответствующие роли.

`nano playbook.yml`

```
- name: Initial config
  hosts: all
  become: yes
  roles:
    - apt
    - ntp
    - monit
    - rsyslog
    - kibana
    - nginx
```

Роли APT, NTP, MONIT по настройкам аналогичны BM ab-haproxy, за исключением правил для monit (перезапуск настроен для rsyslog).

RSYSLOG

Роль предназначена для установки rsyslog в качестве транслятора для стандартных системных логов.

HANDLERS

Добавим обработчик для перезапуска rsyslog.

`main.yml`:

```
---
# handlers file for rsyslog
- name: Restart rsyslog service
  service:
    name: rsyslog
    state: restarted
```

TEMPLATES

Создадим конфигурационный файл для rsyslog.

Укажем порты для приема системных логов для UCP и TCP протоколов в разделе `MODULES'.

`rsyslog.conf`:

```
# /etc/rsyslog.conf configuration file for rsyslog
#
# For more information install rsyslog-doc and see
# /usr/share/doc/rsyslog-doc/html/configuration/index.html
#
# Default logging rules can be found in /etc/rsyslog.d/50-default.conf


#################
#### MODULES ####
#################

module(load="imuxsock") # provides support for local system logging
#module(load="immark")  # provides --MARK-- message capability

# provides UDP syslog reception
module(load="imudp")
input(type="imudp" port="514")

# provides TCP syslog reception
module(load="imtcp")
input(type="imtcp" port="514")

# provides kernel logging support and enable non-kernel klog messages
module(load="imklog" permitnonkernelfacility="on")

###########################
#### GLOBAL DIRECTIVES ####
###########################

#
# Use traditional timestamp format.
# To enable high precision timestamps, comment out the following line.
#
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat

# Filter duplicated messages
$RepeatedMsgReduction on

#
# Set the default permissions for all log files.
#
$FileOwner syslog
$FileGroup adm
$FileCreateMode 0640
$DirCreateMode 0755
$Umask 0022
$PrivDropToUser syslog
$PrivDropToGroup syslog
#
```

```
# Where to place spool and state files
#
$WorkDirectory /var/spool/rsyslog

#
# Include all config files in /etc/rsyslog.d/
#
$IncludeConfig /etc/rsyslog.d/*.conf
```

Также создадим директорию 'rsyslog.d', в которую добавим два файла:

1. `01-json-template.conf`
   Elasticsearch требует, чтобы передаваемые данные были в формате JSON, поэтому используем данный шаблон для преобразования логов.
2. `60-output.conf`
   Данный шаблон используется для отправки данных в logstash на соответствующую ВМ.

`01-json-template.conf`:

```
template(name="json-template" type="list") {
    constant(value="{")
      constant(value="\"@timestamp\":\"")     property(name="timereported"
dateFormat="rfc3339")
      constant(value="\",\"message\":\"")     property(name="msg"
format="json")
      constant(value="\",\"sysloghost\":\"")  property(name="hostname")
      constant(value="\",\"programname\":\"") property(name="programname")
    constant(value="\"}\n")
}
```

`60-output.conf`:

```
# This line sends all lines to defined IP address at port 10514,
# using the "json-template" format template
*.*                         @192.168.1.11:10514;json-template
```

TASKS

Добавим задачи для установки rsyslog и его конфигурации.

`main.yml`:

```
---
# tasks file for rsyslog
- name: Install rsyslog package
  apt:
    name: rsyslog
    state: present

- name: Start and enable rsyslog
  systemd:
    name: rsyslog
    state: started
    enabled: yes


- name: Configure rsyslog settings
  template:
    src: rsyslog.conf
    dest: /etc/rsyslog.conf
  notify:
    - Restart rsyslog service

- name: Configure json template
  template:
    src: rsyslog.d/01-json-template.conf
    dest: /etc/rsyslog.d/01-json-template.conf
  notify:
    - Restart rsyslog service

- name: Configure output
  template:
    src: rsyslog.d/60-output.conf
    dest: /etc/rsyslog.d/60-output.conf
  notify:
    - Restart rsyslog service
```

KIBANA

Данная роль предназначена для установки Kibana для визуализации логов из logstash.

TEMPLATES

Подготовим шаблон для kibana.

Для этого создадим файл `kibana.yml` и укажем адрес, к которому она будет привязана. Для обеспечения удаленного доступа укажем non-loopback адрес.

`kibana.yml`:

```
...
server.host: "0.0.0.0"
...
```

TASKS

Создадим задачи для установки и конфигурации kibana.

`main.yml`:

```
---
# tasks file for kibana
- name: Ensure dependencies are installed.
  apt:
    name:
      - apt-transport-https
      - gnupg2
    state: present

- name: Save repo
  ansible.builtin.shell: echo "deb [trusted=yes]
https://mirror.yandex.ru/mirrors/elastic/8/ stable main" | sudo tee
/etc/apt/sources.list.d/elastic-8.x.list

- name: Install Kibana
  apt:
    name: kibana
    state: present
    update_cache: true

- name: Set kibana base configuration
  template:
    src: kibana.yml
    dest: /etc/kibana/kibana.yml
    mode: 0644

- name: Set permissions to kibana.yml
  become: true
  ansible.builtin.shell: chmod 777 /etc/kibana/*

- name: Add to autostart
  become: true
  ansible.builtin.shell: systemctl daemon-reload && systemctl enable
kibana.service && systemctl start kibana.service

- name: Ensure Kibana is started and enabled
  service:
    name: kibana
    state: started
    enabled: true
```

NGINX

Данная роль предназначена для установки nginx и настройки reverse proxy на kibana.

HANDLERS

Добавим обработчик для перезапуска nginx.

`main.yml`:

```
---
# handlers file for nginx
- name: restart nginx
  service:
    name: nginx
    state: restarted
```

TEMPLATES

Добавим конфигурацию для nginx с настройкой обратного проксирования на kibana.

`kibana.conf`:

```
server {
  listen      80;
  #listen      [::]:80 ipv6only=on;
  server_name _;

  auth_basic "Restricted Access";
  auth_basic_user_file /etc/nginx/.kibana-user;

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

TASKS

Добавим задачи для установки nginx и его конфигурации.

`main.yml`:

```
---
# tasks file for nginx
- name: Update apt cache.
  apt:
    update_cache: yes
    cache_valid_time: 86400
  changed_when: false

- name: Ensure nginx is installed.
  apt:
    name: nginx
    state: present

- name: Copy nginx configuration in place.
  template:
    src: kibana.conf
    dest: /etc/nginx/conf.d/kibana.conf
    owner: root
    group: root
    mode: 0644
  notify:
    - restart nginx
```

Запуск и проверка

1. Запускаем виртуальные машины:
   `vagrant up`
2. Убеждаемся, что роли для каждой из ВМ успешно отработали:

```
PLAY RECAP *********************************************************************
vm1                        : ok=18   changed=13   unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

==> vm2: Importing base box 'bento/ubuntu-18.04'...
==> vm2: Matching MAC address for NAT networking...
==> vm2: Checking if box 'bento/ubuntu-18.04' version '202303.13.0' is up to date...
==> vm2: Setting the name of the VM: vagrant_vm2_1700679139776_90098
==> vm2: Fixed port collision for 22 => 2222. Now on port 2200.
==> vm2: Clearing any previously set network interfaces...
==> vm2: Available bridged network interfaces:
1) enp4s0
2) br-e7273ec4747a
3) docker0
4) br-75eeb0fbf5f6
==> vm2: When choosing an interface, it is usually the one that is
==> vm2: being used to connect to the internet.
==> vm2:
    vm2: Which interface should the network bridge to?
```

```
changed: [vm2]

TASK [elasticsearch : Ensure service is enabled] ******************************
changed: [vm2]

TASK [elasticsearch : Setup yml] **********************************************
changed: [vm2]

TASK [elasticsearch : Reset security settings] ********************************
changed: [vm2]

RUNNING HANDLER [logstash : restart logstash] *********************************
changed: [vm2]

RUNNING HANDLER [elasticsearch : restart elasticsearch] ***********************
changed: [vm2]

PLAY RECAP ********************************************************************
vm2                        : ok=22   changed=18   unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

==> vm3: Importing base box 'bento/ubuntu-18.04'...
==> vm3: Matching MAC address for NAT networking...
==> vm3: Checking if box 'bento/ubuntu-18.04' version '202303.13.0' is up to date...
==> vm3: Setting the name of the VM: vagrant_vm3_1700679437740_71579
==> vm3: Fixed port collision for 22 => 2222. Now on port 2201.
==> vm3: Clearing any previously set network interfaces...
==> vm3: Available bridged network interfaces:
1) enp4s0
2) br-e7273ec4747a
3) docker0
4) br-75eeb0fbf5f6
==> vm3: When choosing an interface, it is usually the one that is
==> vm3: being used to connect to the internet.
==> vm3:
    vm3: Which interface should the network bridge to?
```

```
TASK [rsyslog : Configure output] **********************************************
ok: [vm3]

TASK [kibana : Ensure dependencies are installed.] *****************************
ok: [vm3]

TASK [kibana : Save repo] ******************************************************
changed: [vm3]

TASK [kibana : Install Kibana] *************************************************
ok: [vm3]

TASK [kibana : Add to autostart] ***********************************************
changed: [vm3]

TASK [kibana : Ensure Kibana is started and enabled] ***************************
ok: [vm3]

TASK [nginx : Update apt cache.] ***********************************************
ok: [vm3]

TASK [nginx : Ensure nginx is installed.] **************************************
ok: [vm3]

TASK [nginx : Copy nginx configuration in place.] ******************************
changed: [vm3]

RUNNING HANDLER [nginx : restart nginx] ****************************************
changed: [vm3]

PLAY RECAP ********************************************************************
vm3                        : ok=25   changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ruslan@ruslan-Z690-UD:~/IT-2023/task_4/vagrant/ab-webui/roles/nginx/tasks$
```

3. Подключаемся к vm1:

   `vagrant ssh vm1`

4. Проверим состояние haproxy:

```
vagrant@vagrant:/etc/haproxy$ sudo systemctl status haproxy
● haproxy.service - HAProxy Load Balancer
   Loaded: loaded (/lib/systemd/system/haproxy.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-11-23 18:37:45 UTC; 22s ago
     Docs: man:haproxy(1)
           file:/usr/share/doc/haproxy/configuration.txt.gz
  Process: 28303 ExecReload=/bin/kill -USR2 $MAINPID (code=exited, status=0/SUCCESS)
  Process: 28302 ExecReload=/usr/sbin/haproxy -f $CONFIG -c -q $EXTRAOPTS (code=exited, status=0/SUCCESS)
  Process: 28375 ExecStartPre=/usr/sbin/haproxy -f $CONFIG -c -q $EXTRAOPTS (code=exited, status=0/SUCCESS)
 Main PID: 28376 (haproxy)
    Tasks: 2 (limit: 4656)
   CGroup: /system.slice/haproxy.service
           ├─28376 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid
           └─28377 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid

Nov 23 18:37:45 vagrant systemd[1]: Starting HAProxy Load Balancer...
Nov 23 18:37:45 vagrant haproxy[28376]: Proxy web started.
Nov 23 18:37:45 vagrant haproxy[28376]: Proxy kibana started.
Nov 23 18:37:45 vagrant systemd[1]: Started HAProxy Load Balancer.
vagrant@vagrant:/etc/haproxy$
```

5. Подключимся к vm2 и проверим порты для logstash и elasticsearch:

```
vagrant@vagrant:/etc/logstash$ sudo lsof -i -P -n | grep LISTEN | grep 9200
java      29178   elasticsearch  434u  IPv6  66017      0t0  TCP 192.168.1.11:9200 (LIS
vagrant@vagrant:/etc/logstash$
```

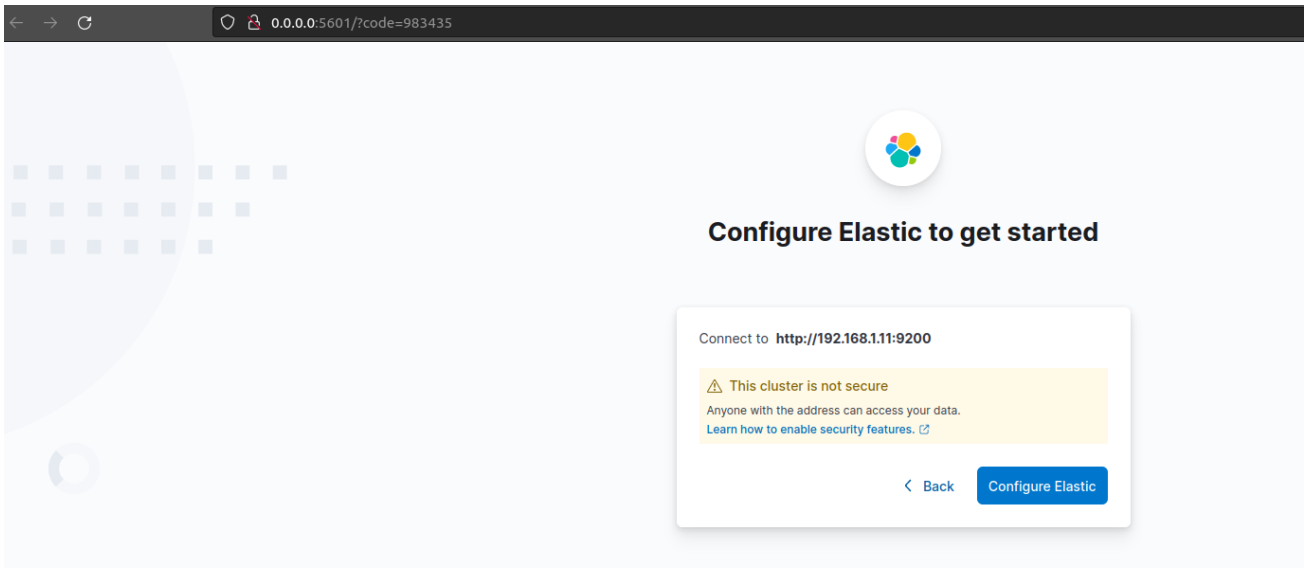
```
vagrant@vagrant:/etc/logstash$ sudo lsof -i -P -n | grep LISTEN | grep 9600
java      29377        logstash   79u  IPv6  67134      0t0  TCP 127.0.0.1:9600 (LISTEN
vagrant@vagrant:/etc/logstash$
```

```
vagrant@vagrant:/etc/logstash/conf.d$ netstat -na | grep 10514
udp        0      0 127.0.0.1:10514         0.0.0.0:*
```

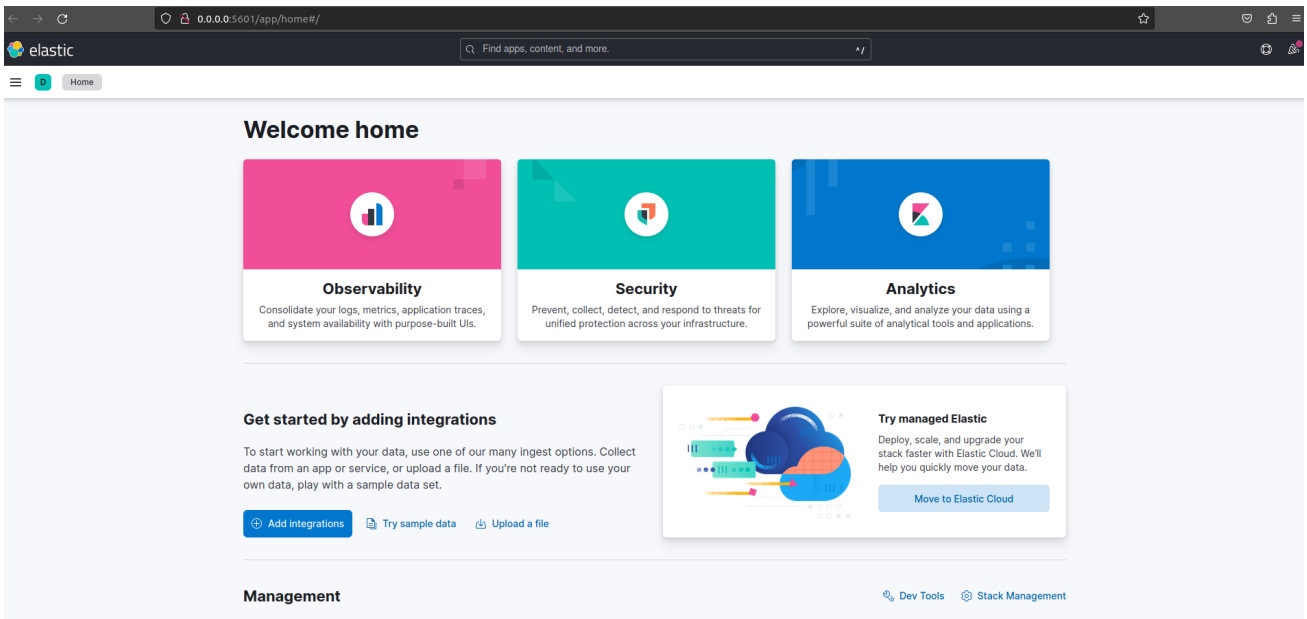
6. Подключимся к vm3 и проверим порты для kibana:

```
ruslan@ruslan-Z690-UD:~/IT-2023/task_4/vagrant$ vagrant ssh vm3
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-206-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system is built by the Bento project by Chef Software
More information can be found at https://github.com/chef/bento
Last login: Wed Nov 22 19:09:50 2023 from 10.0.2.2
vagrant@vagrant:~$ sudo lsof -i -P -n | grep LISTEN | grep 5601
node      30525          kibana   20u  IPv4  74137      0t0  TCP 127.0.0.1:5601 (LISTEN)
vagrant@vagrant:~$
```

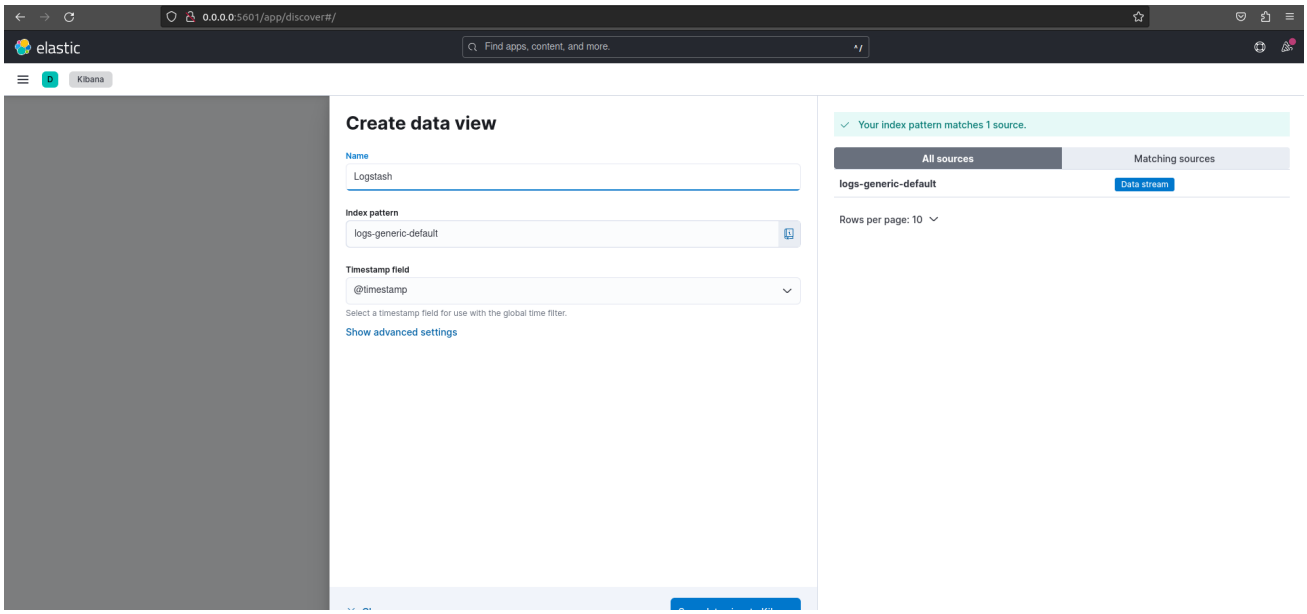
7. Проверим работоспособность kibana в браузере:



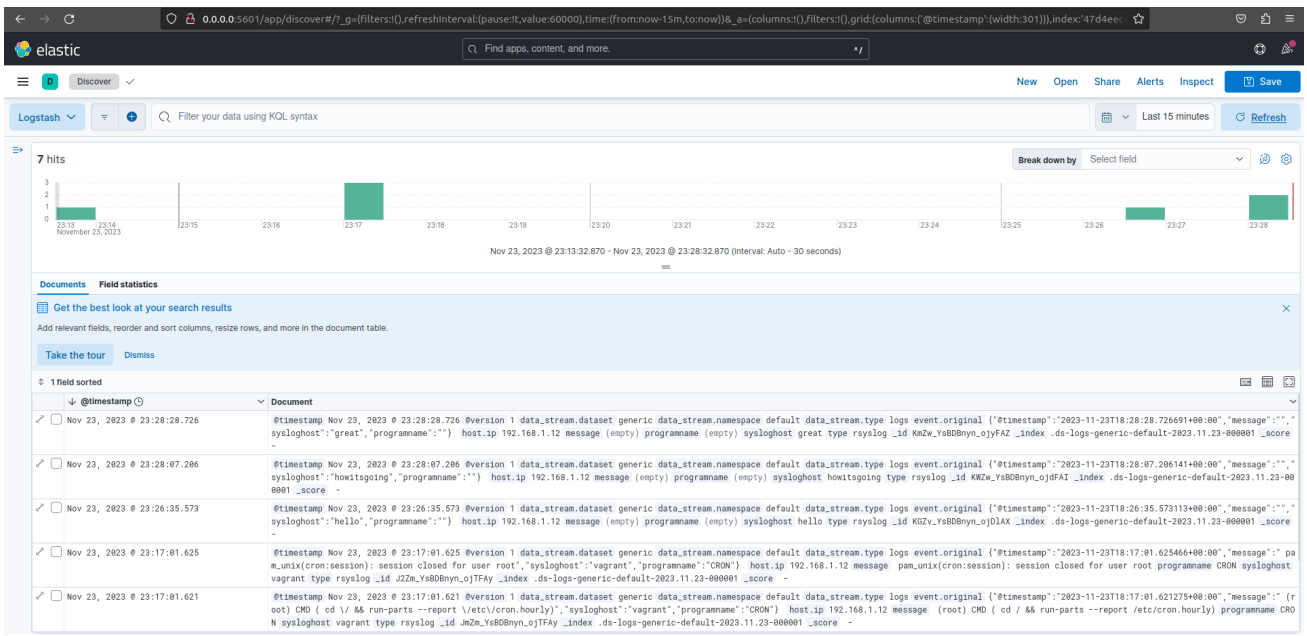
8. Указываем необходимый адрес для подключения и видим стартовую страницу:



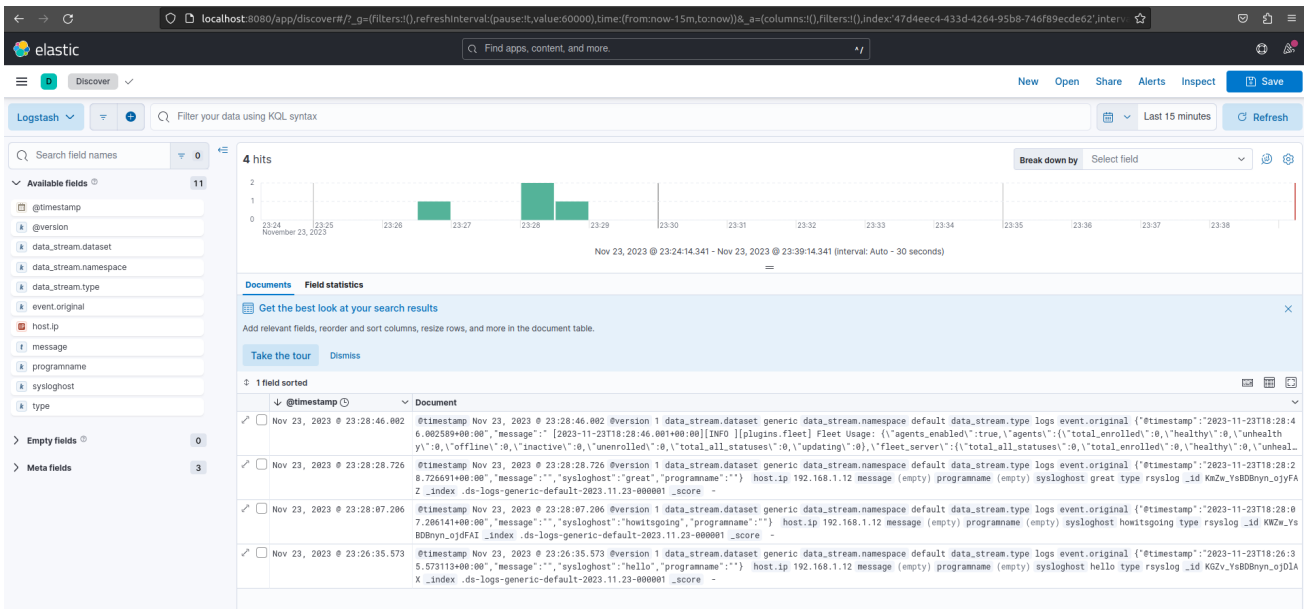
9. Далее создаем data view для logstash:



10. Переходим в созданный data view, где видим, что отображаются наши системные логи:



11. Далее проверим доступность kibana с ВМ ab-haproxy. Для этого в браузере откроем localhost на порте 8080:


localhost:8080/app/discover#/?_g=(filters:!(),refreshInterval:(pause:!t,value:60000),time:(from:now-15m,to:now))&_a=(columns:!(),filters:!(),index:'47d4eec4-433d-4264-95b8-746f89ecde62',interv
elastic
Find apps, content, and more.
Discover
New
Open
Share
Alerts
Inspect
Save
Logstash
Filter your data using KQL syntax
Last 15 minutes
Refresh
Search field names
Available fields
11
@timestamp
@version
data_stream.dataset
data_stream.namespace
data_stream.type
event.original
host.ip
message
programname
sysloghost
type
Empty fields
0
Meta fields
3
4 hits
Break down by
Select field
Nov 23, 2023 @ 23:24:14.341 - Nov 23, 2023 @ 23:39:14.341 (interval: Auto - 30 seconds)
Documents
Field statistics
Get the best look at your search results
Add relevant fields, reorder and sort columns, resize rows, and more in the document table.
Take the tour
Dismiss
1 field sorted
@timestamp
Document
Nov 23, 2023 @ 23:28:46.002
Nov 23, 2023 @ 23:28:28.726
Nov 23, 2023 @ 23:28:07.206
Nov 23, 2023 @ 23:26:35.573