ssh-keygen # генерим на хосте ssh , 3 enter

заходим в папку .ssh

vim id\_rsa.pub # копируем через vi ключ публичный и вставляем в файл на нужно агенте в папку

vi authorized\_keys # вставили

ssh ansible@192.168.0.7 # проверяем заходит ли в ssh где ansible пользователь@IP\_adress

Дальше пишем inventory файл

--

elasticsearch: # группа

hosts:

ubuntu: # название хоста любое

ansible\_connection: ssh # по ssh

ansible\_user: ansible # юзер в линуксе ansible

ansible\_host: 192.168.0.7 # ip

kibana:

hosts:

ubuntu:

ansible\_connection: ssh

ansible\_user: ansible

ansible\_host: 192.168.0.7

далее закидываем в папку files java

после заходим в папку group\_vars создаем папку kibana

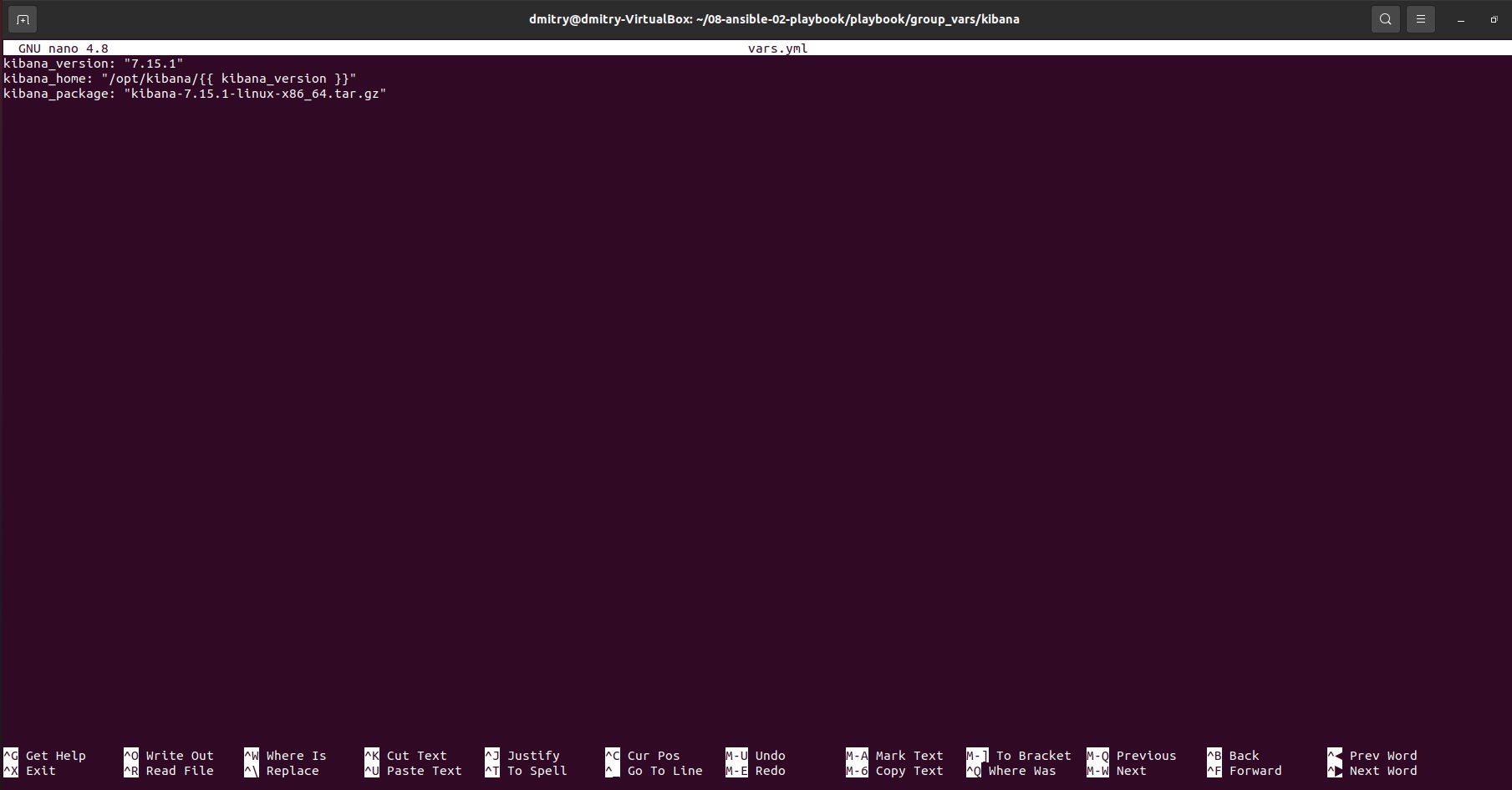
создаем файл vars.yml

kibana\_version: "7.15.1"

kibana\_home: "/opt/kibana/{{ kibana\_version }}"

kibana\_package: "kibana-7.15.1-linux-x86\_64.tar.gz"

см. скрин “kibana.jpg”



Далее заходим в папку файл и скачиваем tar.gz

dmitry@dmitry-VirtualBox:~/08-ansible-02-playbook/playbook/files$ curl –O <https://artifacts.elastic.co/downloads/kibana/kibana-7.15.1-linux-x86_64.tar.gz> #Одна строчка

далее заходим в плейбук site.yml

дописываем плейбук kibana чтобы он брал из папки files kibana 7.15.1 как указано в groups\_vars

- name: Install Kibana

hosts: kibana

become: true

tasks:

- name: Upload .tar.gz file containing binaries from local storage

copy:

src: "{{ kibana\_package }}"

dest: "/tmp/kibana-{{ kibana\_version }}-linux-x86\_64.tar.gz"

# register: download\_kibana\_binaries

# until: download\_kibana\_binaries is succeeded

tags: kibana

- name: Ensure installation dir exists

become: true

file:

state: directory

path: "{{ kibana\_home }}"

tags: kibana

- name: Extract kibana in the installation directory

become: true

unarchive:

copy: false

src: "/tmp/kibana-{{ kibana\_version }}-linux-x86\_64.tar.gz"

dest: "{{ kibana\_home }}"

extra\_opts: [--strip-components=1]

creates: "{{ kibana\_home }}/bin/kibana"

tags:

- skip\_ansible\_lint

- kibana

- name: Export environment variables

become: true

template:

src: kibana.sh.j2

dest: /etc/profile.d/kibana.sh

tags: kibana

- name: Recursively take ownership of a directory

become: yes

file:

path: "{{ kibana\_home }}"

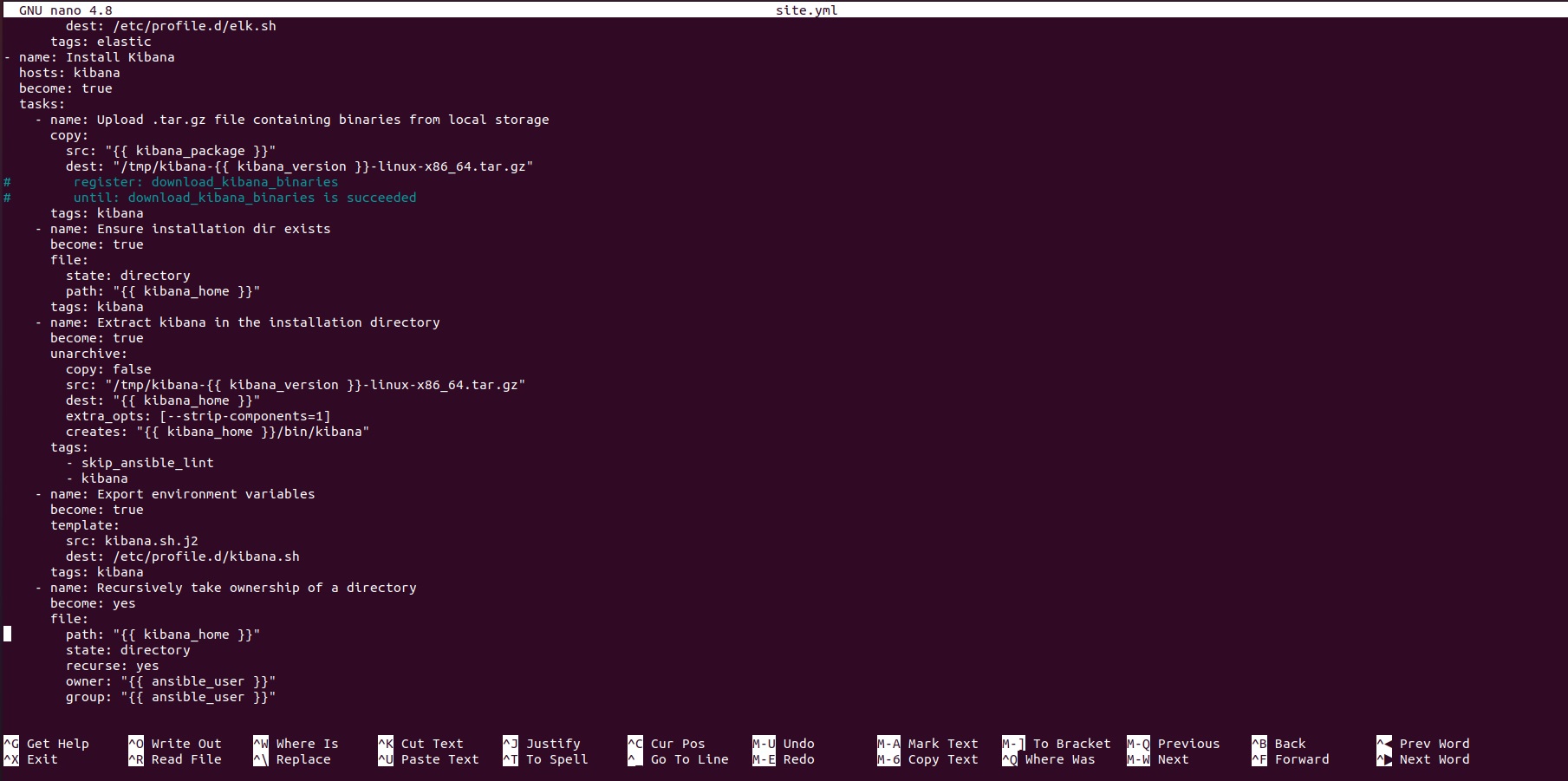
state: directory

recurse: yes

owner: "{{ ansible\_user }}"

group: "{{ ansible\_user }}"

см. скрин “kibana playbook.jpg”



Итоговый плейбук получился

---

- name: Install Java

hosts: all

tasks:

- name: Set facts for Java 11 vars

set\_fact:

java\_home: "/opt/jdk/{{ java\_jdk\_version }}"

tags: java

- name: Upload .tar.gz file containing binaries from local storage

copy:

src: "{{ java\_oracle\_jdk\_package }}"

dest: "/tmp/jdk-{{ java\_jdk\_version }}.tar.gz"

register: download\_java\_binaries

until: download\_java\_binaries is succeeded

tags: java

- name: Ensure installation dir exists

become: true

file:

state: directory

path: "{{ java\_home }}"

tags: java

- name: Extract java in the installation directory

become: true

unarchive:

copy: false

src: "/tmp/jdk-{{ java\_jdk\_version }}.tar.gz"

dest: "{{ java\_home }}"

extra\_opts: [--strip-components=1]

creates: "{{ java\_home }}/bin/java"

tags:

- java

- name: Export environment variables

become: true

template:

src: jdk.sh.j2

dest: /etc/profile.d/jdk.sh

tags: java

- name: Install Elasticsearch

hosts: elasticsearch

become: true

tasks:

- name: Upload tar.gz Elasticsearch from remote URL

get\_url:

url: "https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-{{ elastic\_version }}-linux-x86\_64.tar.gz"

dest: "/tmp/elasticsearch-{{ elastic\_version }}-linux-x86\_64.tar.gz"

mode: 0755

timeout: 60

force: true

validate\_certs: false

register: get\_elastic

until: get\_elastic is succeeded

tags: elastic

- name: Create directrory for Elasticsearch

file:

state: directory

path: "{{ elastic\_home }}"

tags: elastic

- name: Extract Elasticsearch in the installation directory

become: true

unarchive:

copy: false

src: "/tmp/elasticsearch-{{ elastic\_version }}-linux-x86\_64.tar.gz"

dest: "{{ elastic\_home }}"

extra\_opts: [--strip-components=1]

creates: "{{ elastic\_home }}/bin/elasticsearch"

tags:

- elastic

- name: Set environment Elastic

become: true

template:

src: templates/elk.sh.j2

dest: /etc/profile.d/elk.sh

tags: elastic

- name: Install Kibana

hosts: kibana

become: true

tasks:

- name: Upload .tar.gz file containing binaries from local storage

copy:

src: "{{ kibana\_package }}"

dest: "/tmp/kibana-{{ kibana\_version }}-linux-x86\_64.tar.gz"

# register: download\_kibana\_binaries

# until: download\_kibana\_binaries is succeeded

tags: kibana

- name: Ensure installation dir exists

become: true

file:

state: directory

path: "{{ kibana\_home }}"

tags: kibana

- name: Extract kibana in the installation directory

become: true

unarchive:

copy: false

src: "/tmp/kibana-{{ kibana\_version }}-linux-x86\_64.tar.gz"

dest: "{{ kibana\_home }}"

extra\_opts: [--strip-components=1]

creates: "{{ kibana\_home }}/bin/kibana"

tags:

- skip\_ansible\_lint

- kibana

- name: Export environment variables

become: true

template:

src: kibana.sh.j2

dest: /etc/profile.d/kibana.sh

tags: kibana

- name: Recursively take ownership of a directory

become: yes

file:

path: "{{ kibana\_home }}"

state: directory

recurse: yes

owner: "{{ ansible\_user }}"

заходим в teamplates и создаем файл kibana.sh.j2

dmitry@dmitry-VirtualBox:~/08-ansible-02-playbook/playbook/templates$ touch kibana.sh.j2

добавляем переменную PATH

# Warning: This file is Ansible Managed, manual changes will be overwritten on next playbook run.

#!/usr/bin/env bash

export KIBANA\_HOME={{ kibana\_home }}

export PATH=$PATH:$KIBANA\_HOME/bin

Запускаем плейбук

dmitry@dmitry-VirtualBox:~/08-ansible-02-playbook/playbook$ ansible-playbook -i inventory/prod.yml site.yml

см. скрин “final playbook.jpg”

