

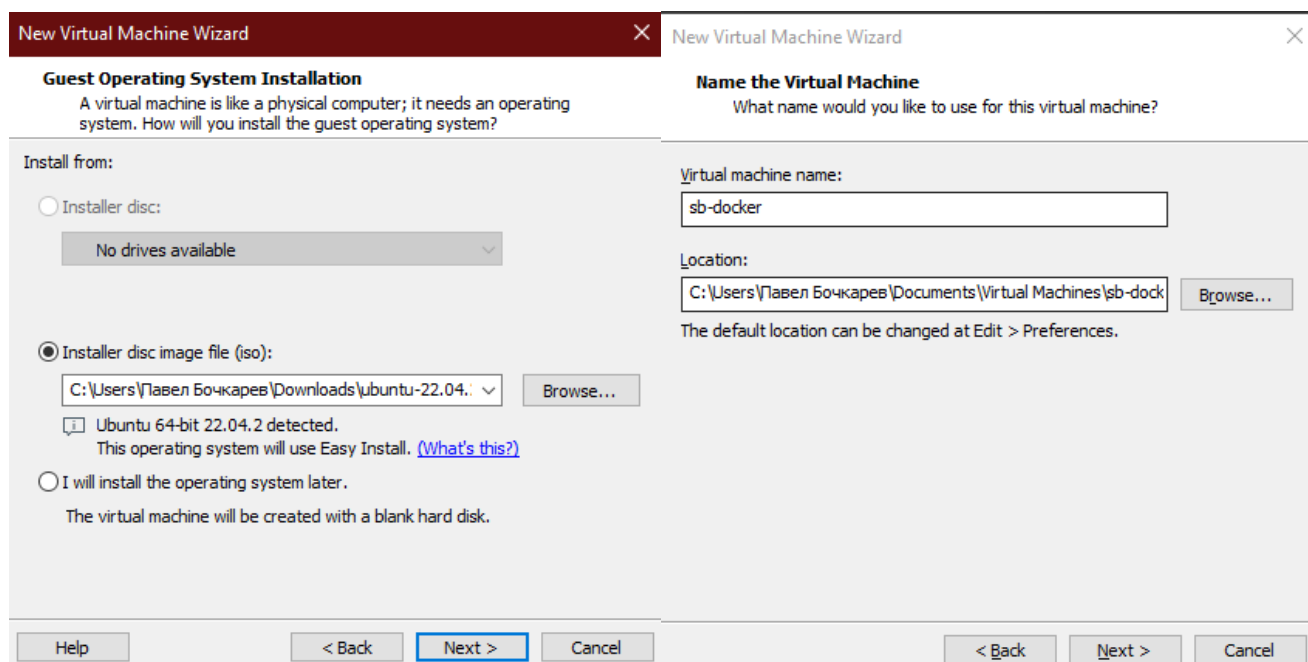
Оглавление

<i>Настройка виртуальных машин</i>	<i>2</i>
<i>Установка Ansible</i>	<i>5</i>
<i>Подготовка к написанию инвентаря.</i>	<i>5</i>
<i>Пишем inventory.....</i>	<i>6</i>
<i>Проверим работу Ansible</i>	<i>6</i>
<i>Напишем Ansible Playbook</i>	<i>7</i>
<i>Запустим наш Playbook.....</i>	<i>10</i>

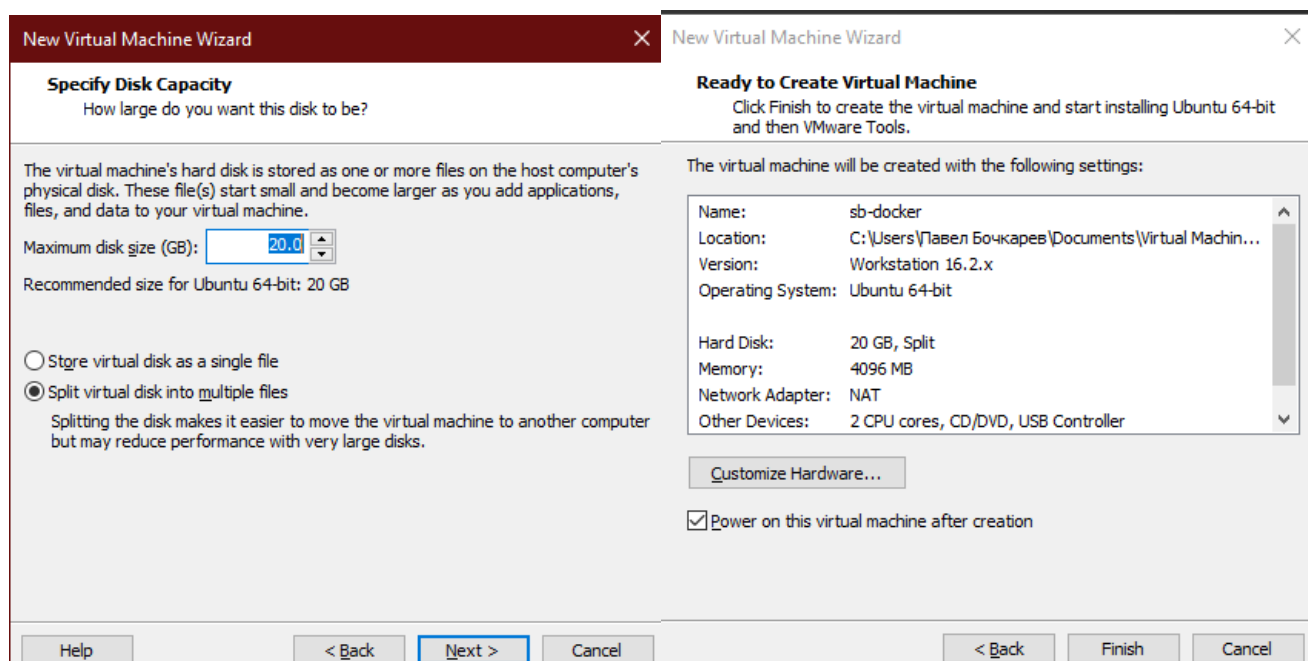
Настройка виртуальных машин

В данном разделе я приведу пример раскатки VM с помощью VMWare Workstation для Docker машины, однако те же действия я повторял и для Ansible VM.

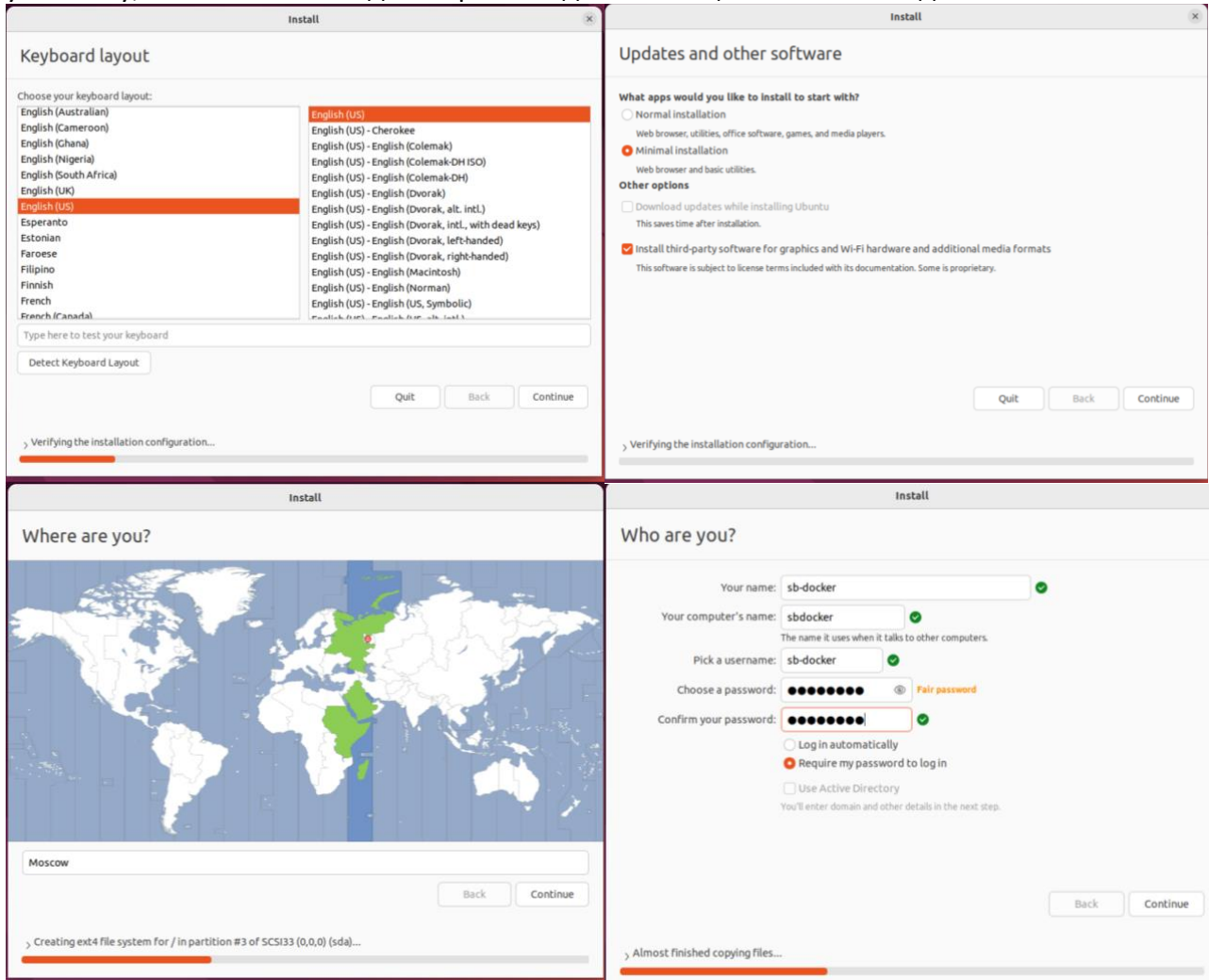
Создадим новую VM. Для этого подкинем десктопный образ Ubuntu 22.04. Назовем машину – sb-docker.



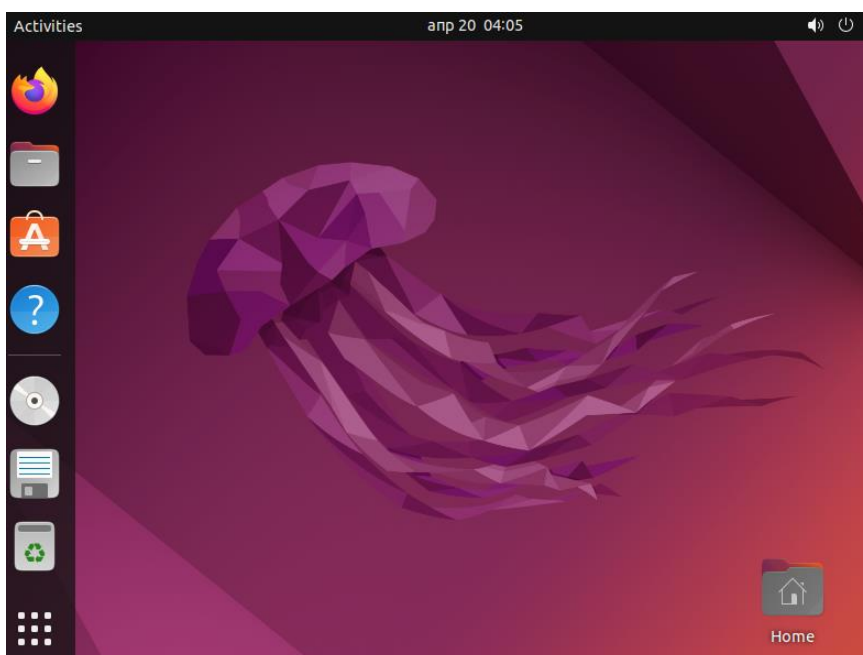
Выделим 20GB памяти постоянной, 4GB RAM, 2CPU (можно меньше, но это оптимально как по мне).



Далее включим ВМ и проследуем стандартные настройки ОС. Выберем минимальную установку, так как никакого доп.софта нам для наших целей не необходимо.



Все готово.



Для более удобной работы можно установить open-vm-tools. Это позволит нам drag-n-drop'ить. Файлы между нашей машиной и VM, а также добавит общий буфер обмена, что очень полезно☺. Далее скриншотов не буду прикладывать, необходимо просто выполнить следующие команды.

```
sudo apt update  
sudo apt upgrade  
sudo apt install open-vm-tool-desktop (да-да, именно десктоп, так как у нас GUI)
```

Также установим openssh-server

```
sudo apt install openssh-server
```

Базово-минимальная настройка для комфортного выполнения текущей задачи готово, однако для других целей необходимо будет настраивать чуть больше.

Установка Ansible

Установку будем выполнять в соответствии с официальной документацией:

https://docs.ansible.com/ansible/latest/installation_guide/installation_distros.html

```
sudo apt update
sudo apt install software-properties-common
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install ansible
```

Подготовка к написанию инвентаря.

В этой части отчета мы будем устанавливать соединение по SSH с помощью ключа.

В части с установкой VM мы уже установили SSH сервер, поэтому сейчас мы будем генерировать SSH ключ Ansible хоста:

```
ansible-sb@ansible-sb:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ansible-sb/.ssh/id_rsa):
Created directory '/home/ansible-sb/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ansible-sb/.ssh/id_rsa
Your public key has been saved in /home/ansible-sb/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:ovEndaG0Jo8q66hfEHbDk+Vw3yODcUOM9/Maw11YVdw ansible-sb@ansible-sb
The key's randomart image is:
+---[RSA 3072]-----+
|  . +++      .*|
|  . *. =00    E|
|  o * o.+.+   o|
|  . o o   +oo  .|
|  . . . S .o .|
|  . + = .. o .|
|  . + = o  =   |
|o  . = o  o    |
|*=o.. .        |
+---[SHA256]-----+
```

После этого нам необходимо его перекинуть на Docker хост:

```
ansible-sb@ansible-sb:~$ ssh-copy-id sb-docker@192.168.37.134
The authenticity of host '192.168.37.134 (192.168.37.134)' can't be established.
ED25519 key fingerprint is SHA256:sq9DiRbmPK36vhXv77YN3MqVm2CU0XBcSY8nji6dPxY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
sb-docker@192.168.37.134's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'sb-docker@192.168.37.134'"
and check to make sure that only the key(s) you wanted were added.
```

Проверяем:

```
ansible-sb@ansible-sb:~$ ssh sb-docker@192.168.37.134
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-32-generic x86_64)

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

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

127 updates can be applied immediately.
68 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
```

Готово, теперь мы можем подключаться к нашей VM без пароля.

Пишем inventory

Так как у нас всего 1 VM, то наш файл будет очень простым. Создадим в удобной нам директории файл inventory.ini. Сделаем это с помощью команды:

```
nano inventory.ini
```

И напишем там следующее:

```
[demo]
192.168.37.134
```

Сохраняемся и готово.

Проверим работу Ansible

Для этого выполним команду и вывод должен быть таким.

```
ansible-sb@ansible-sb:~$ ansible all -i ~/inventory.ini -m ping
192.168.37.134 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

Напишем Ansible Playbook

Наш playbook сразу установит Docker и раскатит в нем nginx контейнер в соответствии с заданием. Ниже приведу сам скрипт с комментариями:

```
- name: Deploy Nginx with Docker #Название нашего Playbook
hosts: demo #Группа хостов, на которую раскатим
become: yes #Так как для установки Docker нужны права суперюзера, то этот установим этот параметр
tasks: #список задач
  - name: Установить пакеты для работы с apt-репозиториями
    ansible.builtin.apt:
      name: "{{ item }}"
      state: present
      update_cache: yes
    loop:
      - apt-transport-https
      - ca-certificates
      - curl
      - gnupg
      - lsb-release
      - software-properties-common

  - name: Добавить GPG ключ Docker
    ansible.builtin.apt_key:
      url: https://download.docker.com/linux/ubuntu/gpg
      state: present

  - name: Добавить Docker репозиторий
    ansible.builtin.apt_repository:
      repo: "deb [arch=amd64] https://download.docker.com/linux/ubuntu {{ ansible_lsb.codename }}
stable"
      state: present

  - name: Установить Docker
    ansible.builtin.apt:
      name: docker-ce
      state: present
      update_cache: yes

  - name: Убедиться, что служба Docker запущена и включена при загрузке
    ansible.builtin.systemd:
      name: docker
      state: started
      enabled: yes

  - name: Создать директорию для конфигурационного файла Nginx
    ansible.builtin.file:
```

```
path: /opt/nginx_conf
state: directory
```

- name: Разместить конфигурационный файл Nginx
ansible.builtin.copy:
src: nginx.conf
dest: /opt/nginx_conf/nginx.conf
mode: '0644'
- name: Запустить контейнер Nginx с Docker
ansible.builtin.docker_container:
name: nginx_container
image: nginx:latest
state: started
pull: yes
ports:
- "8080:80"
volumes:
- "/opt/nginx_conf/nginx.conf:/etc/nginx/nginx.conf"

Также приведу пример нашего nginx.conf

```
user nginx;
worker_processes auto;

error_log /var/log/nginx/error.log warn;
pid /var/run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/access.log main;

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;

    keepalive_timeout 65;

    gzip on;
    gzip_disable "msie6";
```



```
server {  
    listen    80;  
    server_name localhost;  
  
    location / {  
        root /usr/share/nginx/html;  
        index index.html index.htm;  
    }  
  
    error_page 500 502 503 504 /50x.html;  
    location = /50x.html {  
        root /usr/share/nginx/html;  
    }  
}
```

Запустим наш Playbook

Добавим в команду использование наших файлов playbook и inventory, а также аргумент «спроси пароль».

```
ansible-sb@ansible-sb:~$ ansible-playbook -i inventory.ini playbook.yml --ask-become-pass
BECOME password:

PLAY [Deploy Nginx with Docker] *****

TASK [Gathering Facts] *****
ok: [192.168.37.134]

TASK [Установить пакеты для работы с apt-репозиториями] *****
changed: [192.168.37.134] => (item=apt-transport-https)
ok: [192.168.37.134] => (item=ca-certificates)
changed: [192.168.37.134] => (item=curl)
ok: [192.168.37.134] => (item=gnupg)
ok: [192.168.37.134] => (item=lsb-release)
ok: [192.168.37.134] => (item=software-properties-common)

TASK [Добавить GPG ключ Docker] *****
changed: [192.168.37.134]

TASK [Добавить Docker репозиторий] *****
changed: [192.168.37.134]

TASK [Установить Docker] *****
changed: [192.168.37.134]

TASK [Убедиться, что служба Docker запущена и включена при загрузке] *****
ok: [192.168.37.134]

TASK [Создать директорию для конфигурационного файла Nginx] *****
changed: [192.168.37.134]

TASK [Разместить конфигурационный файл Nginx] *****
changed: [192.168.37.134]

TASK [Запустить контейнер Nginx с Docker] *****
changed: [192.168.37.134]

PLAY RECAP *****
192.168.37.134 : ok=9 changed=7 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

Готово, теперь проверим работу.

