

## Linux task1

Если задание требует предоставить какую-то отчетность (выгрузки логов, файлы, скриншоты) - именуйте файлы по такому шаблону.

**<блок>\_<your\_secondname>\_<задача>\_<подпункт задачи>.file\_extension**

Например: **linux\_ivanov\_task10\_pipeline.txt**

Всякий раз, когда вы создаете виртуальные машины, на каждой виртуальной машине hostname должен быть изменен со стандартного на такой:

**<your\_secondname>\_<virtual\_machine\_name>**

Например: **ivanov\_a**

**Важно: Можно делать снапшоты и бекапы состояний виртуальных машин в процессе выполнения заданий по блокам.**

## Критерии выполнения задания

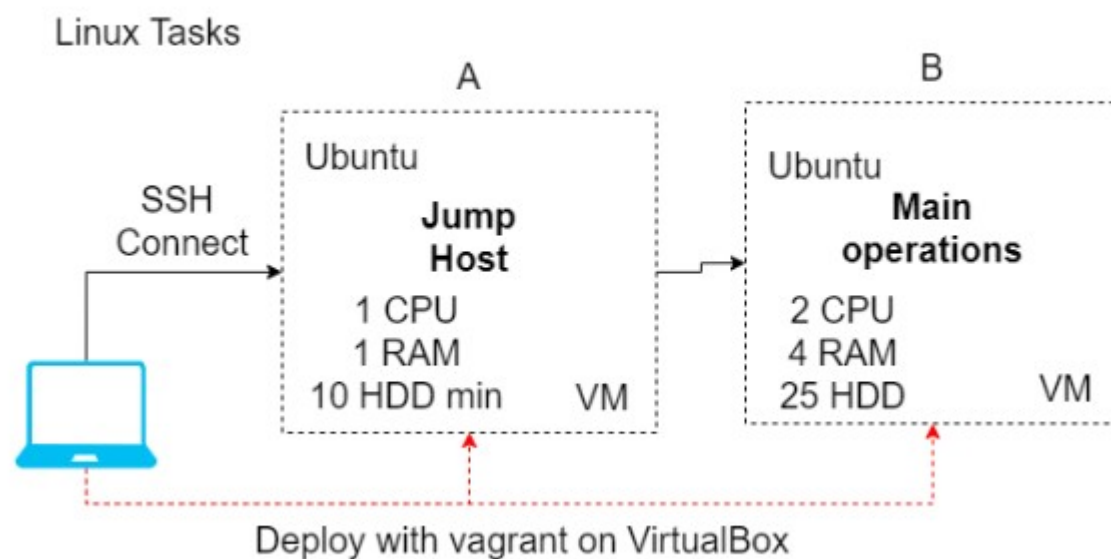
Выполнены все задания. Сформирована папка с ответами по каждому заданию. Архив с ответами загружен на проверку.

Task\_1:

Развернуть виртуальную машину (A) через Vagrant

В качестве отчета предоставить Vagrantfile и лог его работы. Лог выгрузить в файл **linux\_<your\_secondname>\_task\_1.txt**

## Схема



Vagrant консольная утилита, которая управляет виртуальными машинами. По умолчанию Vagrant работает с системой виртуализации VirtualBox. Такая связка доступна в операционных системах Windows, Mac и Linux.

В папке проекта создается файл Vagrantfile с описанием конфигурации виртуальной машины. В котором указывается операционная система, выделяемые ресурсы, сетевые настройки и скрипты по установке и настройке дополнительных программ.

Команда `vagrant up` на основе Vagrantfile создает и запускает виртуальную машину. Во время ее работы папка проекта синхронизируется между основной операционной системой и операционной системой виртуальной машины.

Разработка ведется в основной операционной системе в привычных программах (браузер, IDE и т.д.), а код запускается в изолированном окружении виртуальной машины.

Достаточно поделиться Vagrantfile с членами команды, чтобы они так же легко и быстро развернули среду разработки с теми же зависимостями и настройками.

Дальнейшие шаги я проделывал на Fedora Linux и буду описывать именно для этой ОС. Отличий для других дистрибутивов Linux думаю не будет.

## Установка

Сначала скачайте и установите [VirtualBox](#) для своей операционной системы. Затем скачайте и установите [Vagrant](#).

Откройте терминал если вы используете Linux или Mac. В Windows установите и запустите эмулятор [Git BASH](#). Для проверки успешной установки введите команду:

```
vagrant -v
```

Если видите что-то вроде этого, установка прошла успешно:

```
Vagrant 2.3.3
```

## Использование

Создадим папку проекта и перейдем в нее:

```
// создание папки проекта  
mkdir NeoflexProject
```

```
// переход в папку  
cd NeoflexProject
```

# Создание Vagrantfile

Теперь займемся созданием виртуальной машины. Введите в терминале:

```
vagrant init -m bento/ubuntu-22.04
```

Команда `vagrant init` создает файл Vagrantfile в текущей директории. В нем на языке Ruby описывается конфигурация виртуальной машины. Синтаксис простой и понятный даже если вы никогда не использовали Ruby. Содержимое созданного файла Vagrantfile:

```
Vagrant.configure("2") do |config|  
  config.vm.box = "bento/ubuntu-22.04"  
end
```

При создании Vagrantfile мы указали название бокса `bento/ubuntu-22.04`. Бокс это образ операционной системы, который так же может содержать установленные программы (LAMP, Python и т.д).

Разработчики Vagrant рекомендуют использовать образы [Bento](#), которые хранятся в специальном [каталоге](#) вместе с остальными образами.

Перед созданием виртуальной машины добавим еще несколько настроек в [Vagrantfile](#):

```
1  # -*- mode: ruby -*-  
2  # vi: set ft=ruby :  
3  
4  # Задаем имя машины A и её hostname  
5  A_NODE_NAME = 'machine.a' # Указываем имя виртуальной машины  
6  A_NODE_HOSTNAME = 'suleymanov.a' # Указываем hostname виртуальной машины  
7  A_NODE_IP = '192.168.56.110' # Задаем IP адрес виртуальной машине  
8  
9  # Задаем конфигурацию машины A  
10 MEM = 1024 # Указываем количество оперативной памяти  
11 CPU = 1 # Указываем количество ядер процессора  
12  
13 # Задаем параметры для создания машины A  
14 Vagrant.configure("2") do |config|  
15   config.vm.box = "bento/ubuntu-22.04" # Указываем название дистрибутива виртуальной машины  
16   config.vm.provider "virtualbox" do |v|  
17     v.memory = MEM # Присваиваем значение ОЗУ из конфигурации машины A  
18     v.cpus = CPU # Присваиваем значение ЦПУ из конфигурации машины A  
19   end  
20  
21   # Конфигурируем машину A  
22   config.vm.define A_NODE_NAME do |a|  
23     a.vm.hostname = A_NODE_HOSTNAME # Присваиваем значение hostname из конфигурации имени машины A  
24     a.vm.network :private_network, ip: A_NODE_IP # Присваиваем IP адрес из конфигурации машины A  
25     a.vm.synced_folder ".", "/vagrant", type: "virtualbox" # Создаем общую папку между виртуальной и основной машиной  
26     a.vm.provider "virtualbox" do |v|  
27       v.name = A_NODE_NAME # Присваиваем значение hostname из конфигурации имени hostname A  
28     end  
29   end  
30 end
```

Сделал скрин потому что не знаю как корректно вставить код в PDF файл => Этот Vagrantfile вы можете найти в моем репозитории)

- `config.vm.box` — базовый образ. В нашем случае Ubuntu 22.04
- `config.vm.provider` — система виртуализации. Доступные провайдеры: VirtualBox, VMware, Hyper-V, Docker и др.
- `a.vm.hostname` — имя хоста.
- `a.vm.synced_folder` — синхронизация папок. Первым параметром передается путь хостовой (основной) машины, вторым параметром передается путь гостевой (виртуальной машины)
- `a.vm.network` — настройки сети. Мы настроили статический ip-адрес для нашей машины

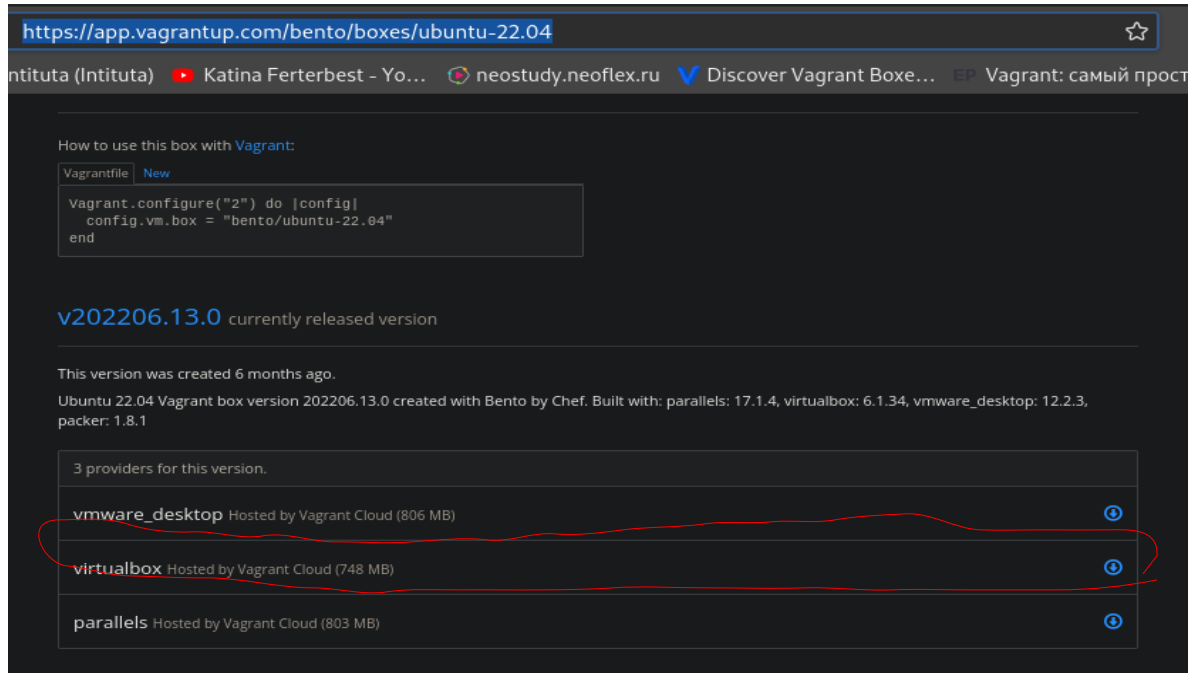
Создаем виртуальную машину. В первый раз процесс займет больше времени, Vagrant должен скачать образ с операционной системой. Выполните команду, которая создаст и запустит виртуальную машину:

**vagrant up**

**Внимание! Vagrant закрыл доступ к своему хранилищу образов ОС для пользователей из России!**

При попытке ввода команды `vagrant up` вы получите 404 ошибку. Есть два пути для решения проблемы:

1. Подключить VPN. Расширение для браузера не пойдет, нужен vpn встроенный в систему, например OpenVPN или Wireguard.
2. Вручную скачать образ ОС, например [bento/ubuntu-22.04](https://bento.io/boxes/ubuntu-22.04) для VirtualBox



3. В терминале переходим в папку со скачанным образом, называться он будет примерно так: `ddc462a2-13f7-4652-bd78-18515defd419.Kk06-9xC`, по-этому переименуем его на `bento/ubuntu-22.04`

```
mv ddc462a2-13f7-4652-bd78-18515defd419.Kk06-9xC bento/ubuntu-22.04
```

4. Теперь добавим образ ОС в Vagrant. Находясь в директории с образом ОС, в терминале введем команду

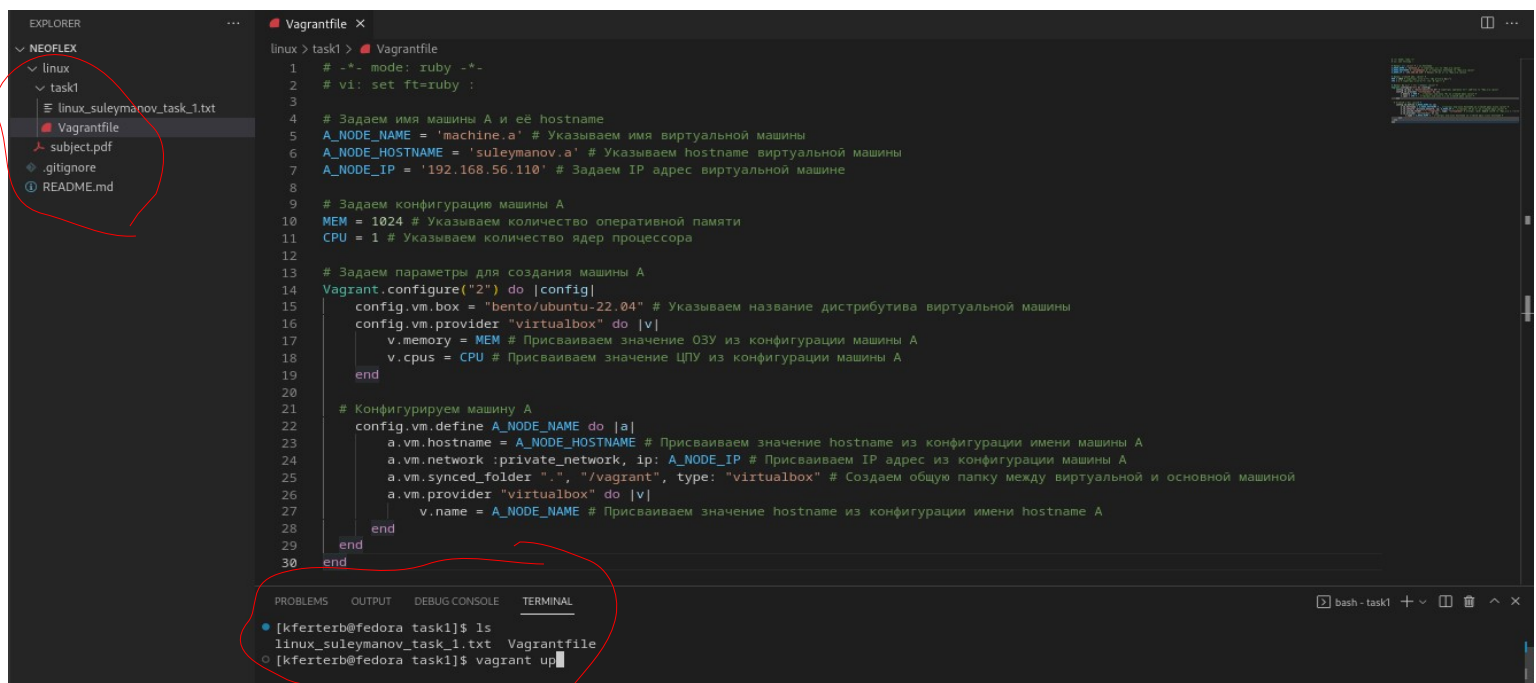
```
vagrant box add bento/ubuntu-22.04
```

**Вagrant ответит нам следующим выводом:**

A `Vagrantfile` has been placed in this directory. You are now ready to `vagrant up` your first virtual environment! Please read the comments in the Vagrantfile as well as documentation on [vagrantup.com](https://www.vagrantup.com) for more information on using Vagrant.

5. Вернемся в нашу папку с проектом где лежит **Vagrantfile** и в терминале введем команду

```
vagrant up
```

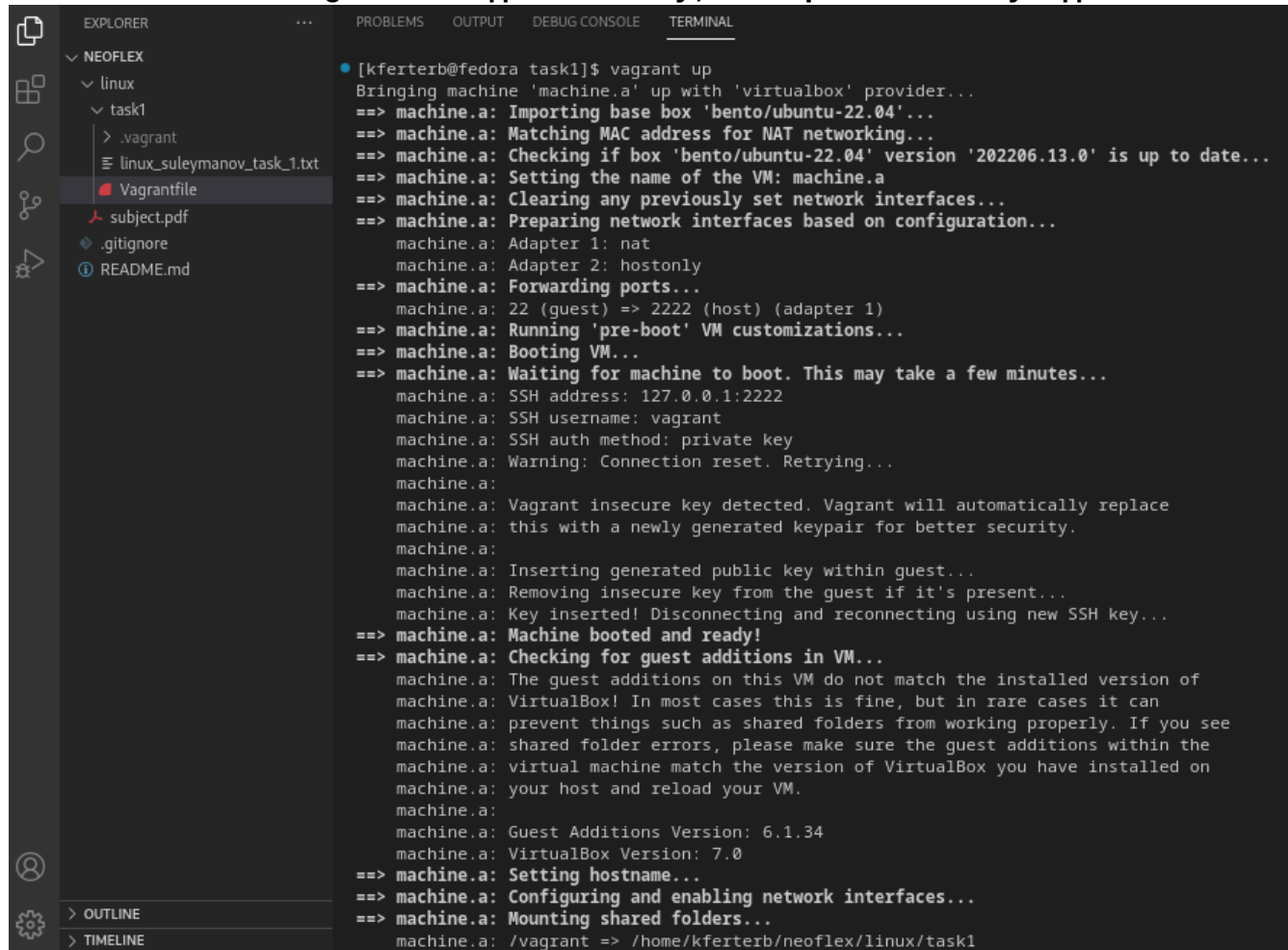


```
1  # -*- mode: ruby -*-
2  # vi: set ft=ruby :
3
4  # Задаем имя машины A и её hostname
5  A_NODE_NAME = 'machine.a' # Указываем имя виртуальной машины
6  A_NODE_HOSTNAME = 'suleymanov.a' # Указываем hostname виртуальной машины
7  A_NODE_IP = '192.168.56.110' # Задаем IP адрес виртуальной машине
8
9  # Задаем конфигурацию машины A
10 MEM = 1024 # Указываем количество оперативной памяти
11 CPU = 1 # Указываем количество ядер процессора
12
13 # Задаем параметры для создания машины A
14 Vagrant.configure("2") do |config|
15   config.vm.box = "bento/ubuntu-22.04" # Указываем название дистрибутива виртуальной машины
16   config.vm.provider "virtualbox" do |v|
17     v.memory = MEM # Присваиваем значение ОЗУ из конфигурации машины A
18     v.cpus = CPU # Присваиваем значение ЦПУ из конфигурации машины A
19   end
20
21   # Конфигурируем машину A
22   config.vm.define A_NODE_NAME do |a|
23     a.vm.hostname = A_NODE_HOSTNAME # Присваиваем значение hostname из конфигурации имени машины A
24     a.vm.network :private_network, ip: A_NODE_IP # Присваиваем IP адрес из конфигурации машины A
25     a.vm.synced_folder ".", "/vagrant", type: "virtualbox" # Создаем общую папку между виртуальной и основной машиной
26     a.vm.provider "virtualbox" do |v|
27       v.name = A_NODE_NAME # Присваиваем значение hostname из конфигурации имени hostname A
28     end
29   end
30 end
```

```
[kferterb@fedora task1]$ ls
linux_suleymanov_task_1.txt  Vagrantfile
[kferterb@fedora task1]$ vagrant up
```

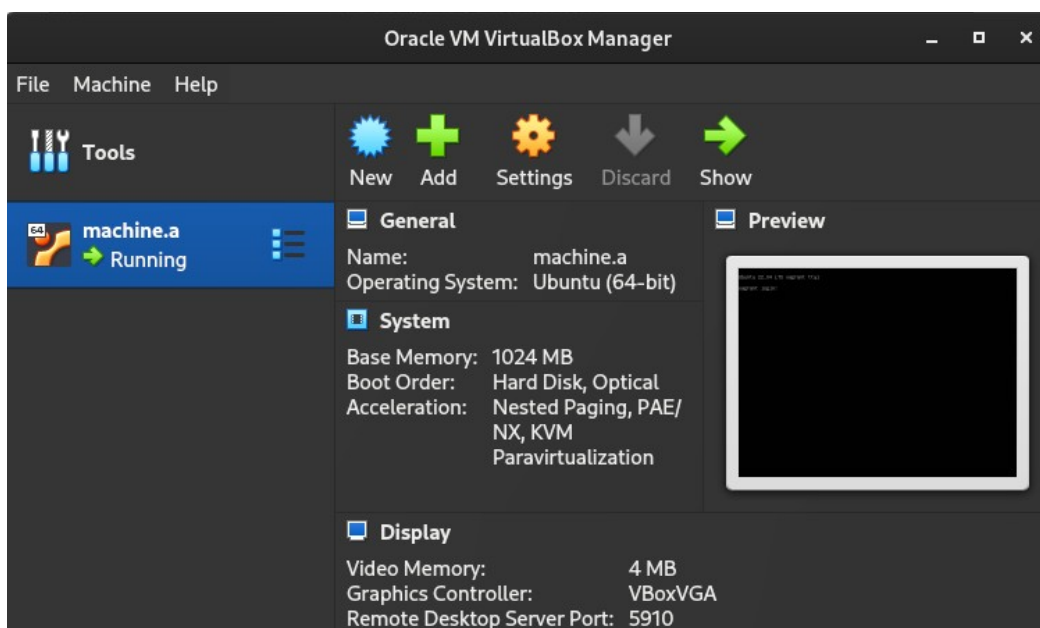
Продолжение ниже =)

После того как Vagrant создаст машину, в терминале мы увидим логи:



```
[kferterb@fedora task1]$ vagrant up
Bringing machine 'machine.a' up with 'virtualbox' provider...
==> machine.a: Importing base box 'bento/ubuntu-22.04'...
==> machine.a: Matching MAC address for NAT networking...
==> machine.a: Checking if box 'bento/ubuntu-22.04' version '202206.13.0' is up to date...
==> machine.a: Setting the name of the VM: machine.a
==> machine.a: Clearing any previously set network interfaces...
==> machine.a: Preparing network interfaces based on configuration...
machine.a: Adapter 1: nat
machine.a: Adapter 2: hostonly
==> machine.a: Forwarding ports...
machine.a: 22 (guest) => 2222 (host) (adapter 1)
==> machine.a: Running 'pre-boot' VM customizations...
==> machine.a: Booting VM...
==> machine.a: Waiting for machine to boot. This may take a few minutes...
machine.a: SSH address: 127.0.0.1:2222
machine.a: SSH username: vagrant
machine.a: SSH auth method: private key
machine.a: Warning: Connection reset. Retrying...
machine.a:
machine.a: Vagrant insecure key detected. Vagrant will automatically replace
machine.a: this with a newly generated keypair for better security.
machine.a:
machine.a: Inserting generated public key within guest...
machine.a: Removing insecure key from the guest if it's present...
machine.a: Key inserted! Disconnecting and reconnecting using new SSH key...
==> machine.a: Machine booted and ready!
==> machine.a: Checking for guest additions in VM...
machine.a: The guest additions on this VM do not match the installed version of
machine.a: VirtualBox! In most cases this is fine, but in rare cases it can
machine.a: prevent things such as shared folders from working properly. If you see
machine.a: shared folder errors, please make sure the guest additions within the
machine.a: virtual machine match the version of VirtualBox you have installed on
machine.a: your host and reload your VM.
machine.a:
machine.a: Guest Additions Version: 6.1.34
machine.a: VirtualBox Version: 7.0
==> machine.a: Setting hostname...
==> machine.a: Configuring and enabling network interfaces...
==> machine.a: Mounting shared folders...
machine.a: /vagrant => /home/kferterb/neoflex/linux/task1
```

Так же, если мы откроем Virtualbox, то увидим нашу созданную машину



Мы можем зайти на нашу виртуальную машину. По умолчанию логин/пароль: vagrant/vagrant

По заданию необходимо сохранить лог развертывания виртуальной машины в файл:

<блок>\_<your\_secondname>\_<задача>\_<подпункт задачи>.file\_extension

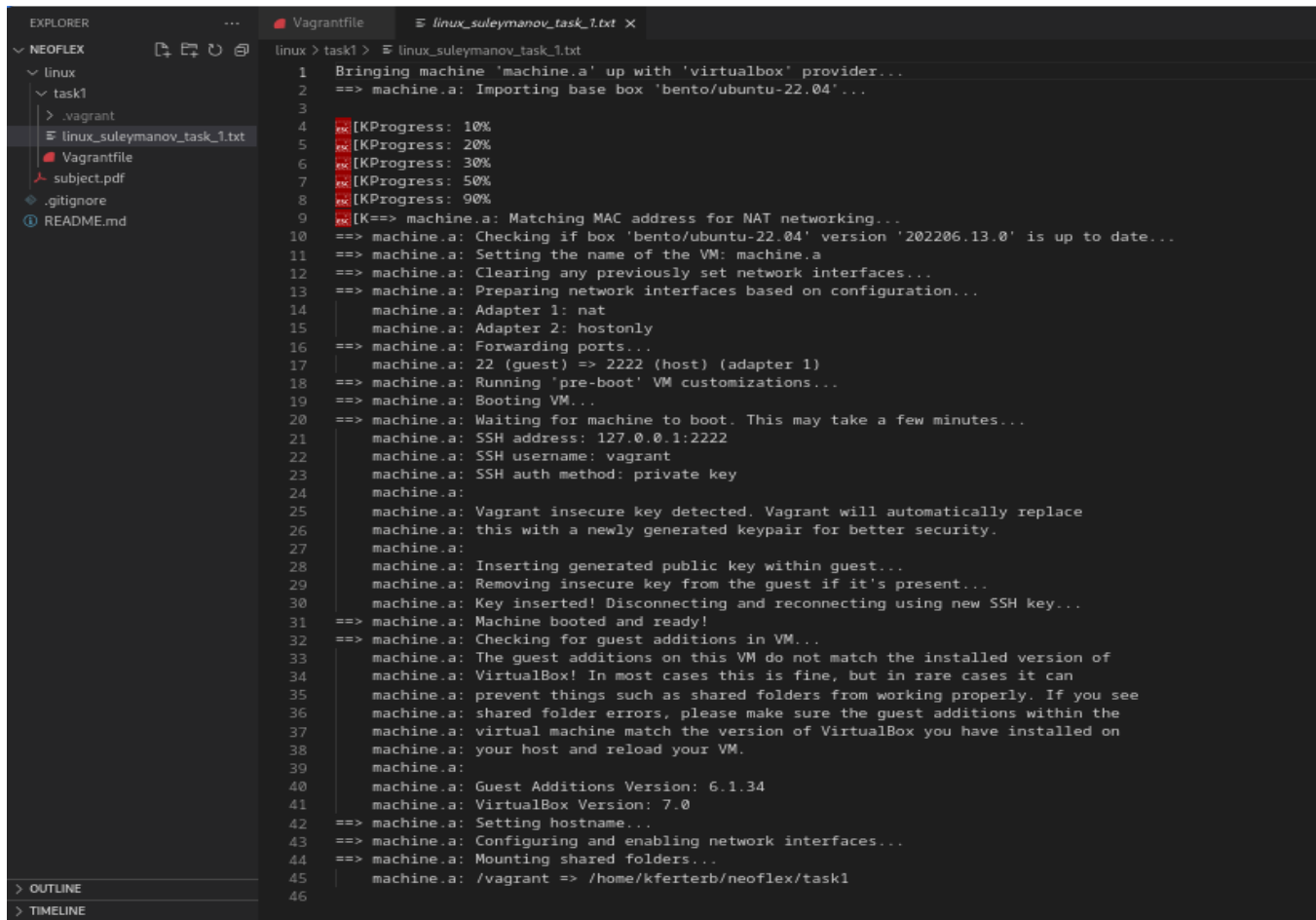
Например: **linux\_ivanov\_task10\_pipeline.txt**

Для этого нужно сделать следующие шаги:

1. Удалим нашу созданную виртуальную машину с помощью команды  
`vagrant destroy`  
и подтвердим удаление Y
2. Снова поднимим нашу виртуальную машину, но укажем файл в который запишется наш лог развертывания  
`vagrant up | cat > linux_ваша_фамилия_task_1.txt`

Так мы перенаправили поток вывода информации из терминала в текстовый файл, который появится в текущей папке. При выполнении развертывания мы не увидим никакой информации в терминале, потому что она сразу пишется в файл.

После окончания развертывания виртуальной машины, можем открыть файл с логами



```
EXPLORER    ...    Vagrantfile    linux_suleymanov_task_1.txt X
└─ linux
  └─ task1
    └─ .vagrant
      └─ linux_suleymanov_task_1.txt
        └─ Vagrantfile
          └─ subject.pdf
            └─ .gitignore
              └─ README.md

linux > task1 > linux_suleymanov_task_1.txt
1 Bringing machine 'machine.a' up with 'virtualbox' provider...
2 ==> machine.a: Importing base box 'bento/ubuntu-22.04'...
3
4 [KProgress: 10%]
5 [KProgress: 20%]
6 [KProgress: 30%]
7 [KProgress: 50%]
8 [KProgress: 90%]
9 [K=> machine.a: Matching MAC address for NAT networking...
10 ==> machine.a: Checking if box 'bento/ubuntu-22.04' version '202206.13.0' is up to date...
11 ==> machine.a: Setting the name of the VM: machine.a
12 ==> machine.a: Clearing any previously set network interfaces...
13 ==> machine.a: Preparing network interfaces based on configuration...
14 machine.a: Adapter 1: nat
15 machine.a: Adapter 2: hostonly
16 ==> machine.a: Forwarding ports...
17 machine.a: 22 (guest) => 2222 (host) (adapter 1)
18 ==> machine.a: Running 'pre-boot' VM customizations...
19 ==> machine.a: Booting VM...
20 ==> machine.a: Waiting for machine to boot. This may take a few minutes...
21 machine.a: SSH address: 127.0.0.1:2222
22 machine.a: SSH username: vagrant
23 machine.a: SSH auth method: private key
24 machine.a:
25 machine.a: Vagrant insecure key detected. Vagrant will automatically replace
26 machine.a: this with a newly generated keypair for better security.
27 machine.a:
28 machine.a: Inserting generated public key within guest...
29 machine.a: Removing insecure key from the guest if it's present...
30 machine.a: Key inserted! Disconnecting and reconnecting using new SSH key...
31 ==> machine.a: Machine booted and ready!
32 ==> machine.a: Checking for guest additions in VM...
33 machine.a: The guest additions on this VM do not match the installed version of
34 machine.a: VirtualBox! In most cases this is fine, but in rare cases it can
35 machine.a: prevent things such as shared folders from working properly. If you see
36 machine.a: shared folder errors, please make sure the guest additions within the
37 machine.a: virtual machine match the version of VirtualBox you have installed on
38 machine.a: your host and reload your VM.
39 machine.a:
40 machine.a: Guest Additions Version: 6.1.34
41 machine.a: VirtualBox Version: 7.0
42 ==> machine.a: Setting hostname...
43 ==> machine.a: Configuring and enabling network interfaces...
44 ==> machine.a: Mounting shared folders...
45 machine.a: /vagrant => /home/kferterb/neoflex/task1
46
```

Если нашли ошибки или недочеты в инструкции, или у вас что-то не получается пишите мне в телеграме @intituta  
С радостью помогу всем =)