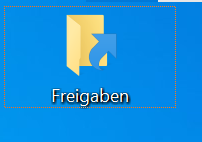
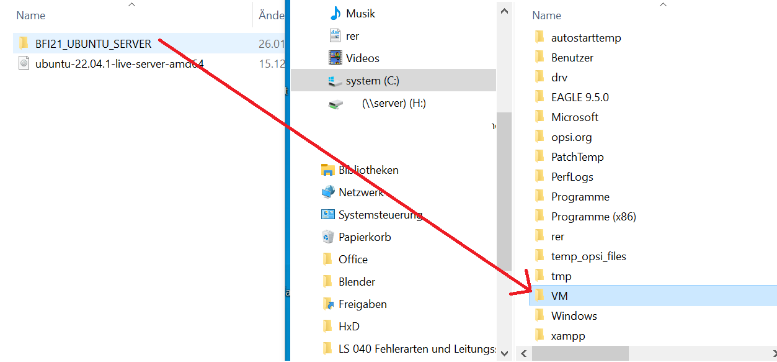
1. Netzlaufwerk öffnen. Die Verknüpfung „Freigabe“ auf dem Desktop öffnen.



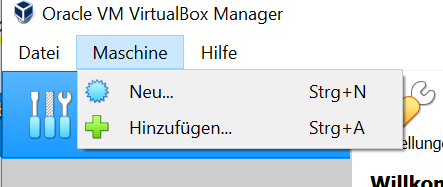
* 1. In das Klassenlaufwerk gehen. Danach in den Ordner LF5 wechseln



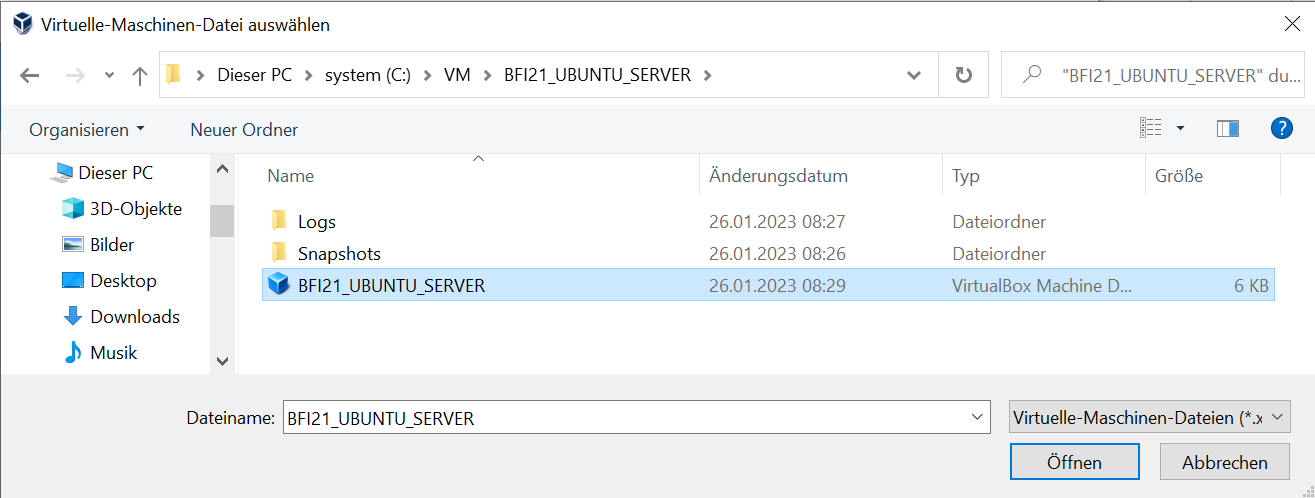
* 1. Virtuelle Maschine „BFI21\_UBUNTU\_SERVER\_TEST“ vom Netzlaufwerk nach „C:\VM\“ kopieren



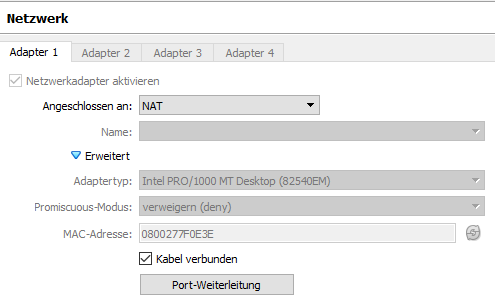
1. Virtuelle Maschine in Virtual Box importieren
   1. Virtual Box öffnen -> Maschine -> Hinzufügen



* 1. In den Ordner C:\VM\ wechseln und virtuelle Maschine auswählen.



1. Virtuelle Maschine starten.
   1. Login: „bbsadmin“ – Passwort: „bbsadmin“ (jeweils ohne „“)
2. Die Maschine so einrichten, dass der Netzwerkmodus NAT verwendet wird und gleichzeitig der Zugriff über einen SSH-Client wie „putty“ möglich ist.
   1. Dokumentation über einen Screenshot mit den relevanten Virtual Box Einstellungen.



* 1. Dokumentation über einen Screenshot mit den relevanten Putty Verbindungseinstellungen. 

1. Aktualisieren Sie das Betriebssystem. Geben Sie die notwendigen Shell-Befehle ein.

**sudo apt-get update**

1. Installieren Sie den github Shellclient. Geben Sie die notwendigen Shell-Befehle ein.

**sudo apt-get install git**

1. Verbinden Sie den github Shellclient mit Ihrem Github-Account.
   1. Geben Sie die notwendigen Shell-Befehle ein.

**git config—global user.name**

**git config—global user.email**

**ssh - keygen**

* 1. Dokumentation über einen Screenshot mit den relevanten Einstellungen auf der github-Webseite.

1. Fügen Sie meinem github Repo eine Datei „vorname\_nachname.txt“ mit dem Inhalt BZTG hinzu. (Erstellen Sie einen pull request)
2. Installieren und einrichten von Docker.
   1. Installieren Sie das docker Paket docker.io
   2. Fügen Sie den aktuellen Benutzer der Gruppe „docker“ hinzu.
3. In dieser Aufgabe sollen Sie einen Docker Image und Container erstellen und dieses zu gihhub hochladen.
   1. Erstellen Sie ein github Repo „myStaticWebSrv“.
   2. Laden Sie das Repo herunter.
   3. Erstellen Sie der lokalen Kopie des Repo " myStaticWebSrv " die Dateien:
      * Dockerfile
      * docker\_build.sh
      * docker\_run.sh
      * index.html
   4. Fügen Sie den folgenden Inhalt den Dateien hinzu:

|  |  |
| --- | --- |
|  | Inhalt Dockerfile:  FROM alpine  RUN apk update &&\  apk upgrade &&\  apk add apache2 libxml2-dev apache2-utils  COPY index.html /var/www/localhost/htdocs/index.html  CMD ["/usr/sbin/httpd","-D","FOREGROUND"]  # Start httpd  #ENTRYPOINT ["/usr/sbin/httpd"]  EXPOSE 80 443 |
|  | Inhalt index.html:  <html><body><h1>BZTG at work</h1></body></html> |
|  | Inhalt docker\_build.sh:  sudo docker build -t myalpinehttpd . |
|  | Inhalt docker\_run.sh:  docker stop cnt\_simple\_apache  docker rm cnt\_simple\_apache  docker run -d -p 80:80 --name cnt\_simple\_apache myalpinehttpd |

* 1. Machen Sie die Dateien „docker\_build.sh“ und „docker\_run.sh“ ausführbar.

sudo usermod -aG docker $USER

* 1. Laden Sie alle Dateien und Änderungen auf github hoch