

Azure Deployment

VM Anlegen

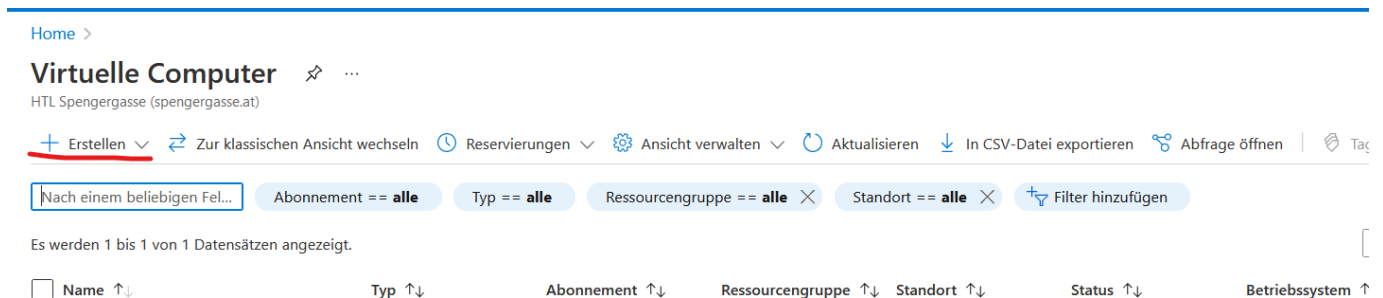
1. Auf portal.azure.com mit der Spengergasse-Email anmelden

WICHTIG: Während der Anmeldung niemals Kreditkartendaten angeben. Falls Azure das verlangt nochmal prüfen ob ihr ein Studentenkonto erstellt habt.

2. Zum **Virtuelle Computer** Menü gehen



Auf erstellen klicken, dannach auf **Azure VM**



Wähle eine bestehende Ressourcengruppe aus oder erstelle ggf. eine neue. Leg einen beliebigen Namen fest. Am besten mit deinem Kürzel oder intialen um doppelte Benennungen zu vermeiden.

Virtuellen Computer erstellen ...

wie z. B. Ordner zum Organisieren und Verwalten all Ihrer Kessourcen.

Abonnement * ⓘ

Azure for Students

Ressourcengruppe * ⓘ

asphost_group (neu)

[Neues Element erstellen](#)

Instanzendetails

Name des virtuellen Computers * ⓘ

asphost

Region * ⓘ

(Europe) West Europe


Verfügbarkeitsoptionen ⓘ

Keine Infrastrukturredundanz erforderlich.


Sicherheitstyp ⓘ

Standard

Image * ⓘ

 Ubuntu Server 20.04 LTS - x64 Gen2

[Alle Images anzeigen](#) | [VM-Generation konfigurieren](#)

 Dieses Image ist mit zusätzlichen Sicherheitsfunktionen kompatibel. [Klicken Sie hier, um zum Sicherheitstyp „Vertrauenswürdiger Start“ zu wechseln.](#)

VM-Architektur ⓘ

☐ Arm64

☒ x64

Erstelle eine VM entweder mit Ubuntu oder alternativ mit einer Distribution mit der du dich gut auskennst. Zum authentifizieren ist ein SSH-Schlüssel empfehlenswert da man sich kein Passwort merken muss.

Größe * ⓘ Standard_B1s - 1 vcpu, 1 GiB Arbeitsspeicher (8,76 \$/Monat) (für kostenlose ... Alle Größen anzeigen

Aktivieren des Ruhezustands (Vorschau) ⓘ ☐

Administratorkonto

Authentifizierungstyp ⓘ ☒ Öffentlicher SSH-Schlüssel ☐ Kennwort

i Azure generiert jetzt automatisch ein SSH-Schlüsselpaar für Sie, das zur späteren Verwendung speicherbar ist. Mithilfe des SSH-Schlüsselpaars können Sie schnell, einfach und sicher eine Verbindung zu Ihrer VM herstellen.

Benutzername * ⓘ azureuser ✓

Quelle für öffentlichen SSH-Schlüssel Neues Schlüsselpaar generieren ✓

Schlüsselpaarname * asphost_key ✓

Bei den Eingangsport brauchen wir SSH (22) und HTTP (80). Über Port 80 werden wir später auf unsere Webanwendung zugreifen. Über Port 22 können wir eine Konsole der VM erreichen. Wenn du willst kannst du auch versuchen HTTPS einzurichten, ist aber für diese Übung nicht notwendig.

Regeln für eingehende Ports

Wählen Sie aus, welche VM-Netzwerkports über das öffentliche Internet zugänglich sind. Auf der Registerkarte "Netzwerk" können Sie einen begrenzteren oder differenzierten Netzwerkzugriff angeben.

Öffentliche Eingangsports * ⓘ ☐ Keine ☒ Ausgewählte Ports zulassen

Eingangsports auswählen * HTTP (80), SSH (22) ✓

- ☒ HTTP (80)
- ☐ HTTPS (443)
- ☒ SSH (22)

Klicke dann auf **Überprüfen + erstellen** die Zusammenfassung sollte folgendermaßen aussehen:

Virtuellen Computer erstellen ...

✓ Überprüfung erfolgreich

Grundeinstellungen Datenträger Netzwerk Verwaltung Überwachung Erweitert Tags Überprüfen + erstellen

i Die unten angegebenen Kosten sind eine Schätzung und nicht der endgültige Preis. Bitte verwenden Sie [Preisrechner](#) für alle Ihre Preisanforderungen.


Preis

1 X Standard B1s
von Microsoft
[Nutzungsbedingungen](#) | [Datenschutzrichtlinie](#)

Abonnementguthaben anwendbar ⓘ
0.0120USD/Std.
[Preise für andere VM-Größen](#)






Nachdem die Ressource erstellt wurde bekommt man folgende Bestätigung:


[Home](#) >


 **CreateVm-canonical.0001-com-ubuntu-server-focal-2-20240416072814** | Übersicht ⚙ ...


Bereitstellung


<<

 Löschen  Abbrechen  Erneut bereitstellen  Herunterladen  Aktualisieren



 Übersicht

 Eingaben

 Ausgaben

 Vorlage

✓ Ihre Bereitstellung wurde abgeschlossen.

 Bereitstellungsname: CreateVm-canonical.0001-com-ubuntu-server... Startzeit: 16.4.2024, 07:32:44
Abonnement: [Azure for Students](#) Korrelations-ID: 05e48bf6-92cf-4855-840d-c7033250e195 
Ressourcengruppe: [asphost_group](#)

▽ Bereitstellungsdetails

^ Nächste Schritte

[Automatisches Herunterfahren einrichten](#) Empfohlen

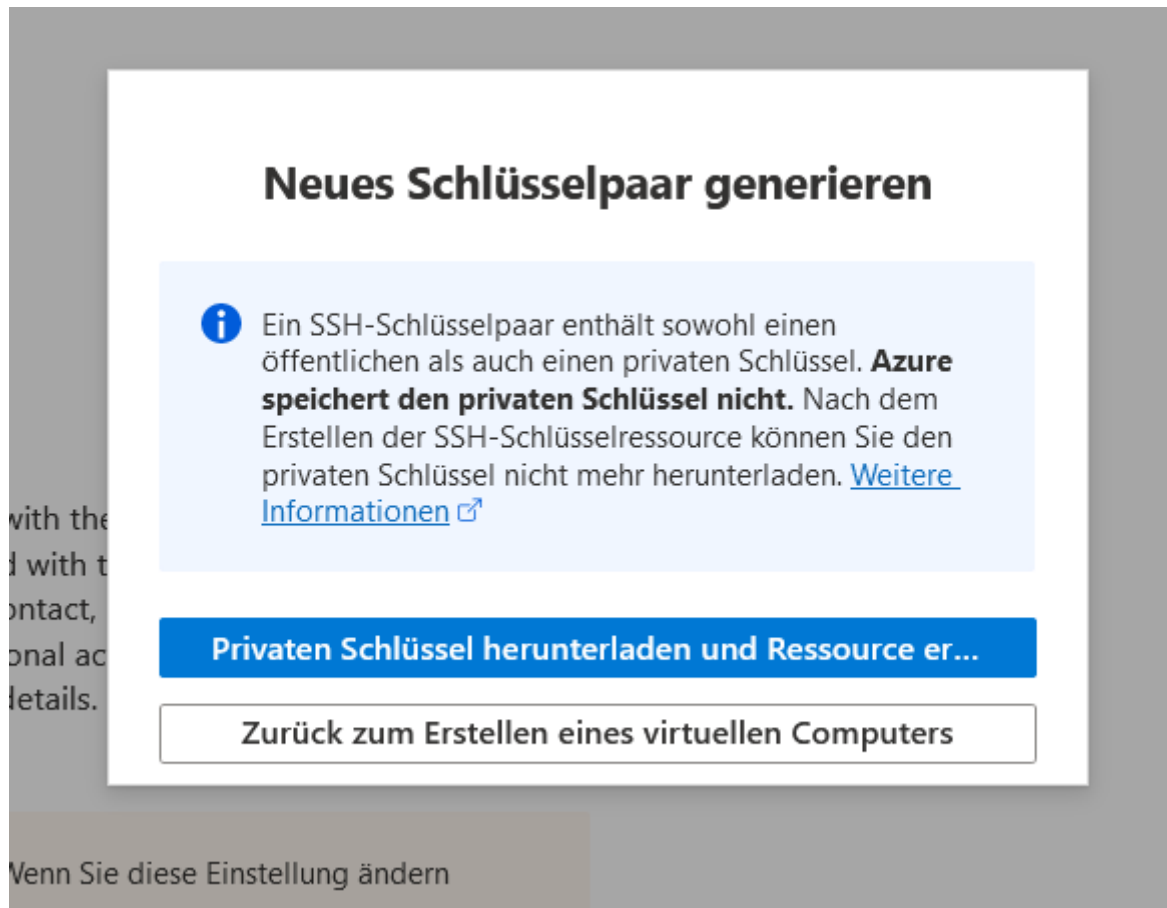
[VM-Integrität, Leistung und Netzwerkabhängigkeiten überwachen](#) Empfohlen

[Skript in VM ausführen](#) Empfohlen

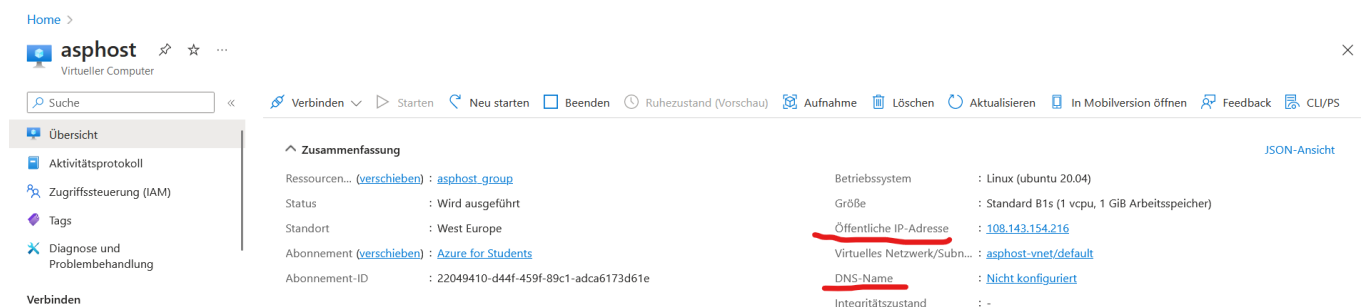
Zu Ressource wechseln

Weitere VM erstellen

WICHTIG beim Klick auf **Erstellen** erstellt Azure uns einen privaten Schlüssel in einer pem-Datei. Diesen brauchen wir um uns einloggen zu können. Unbedingt herunterladen!



In der Übersichtsseite können wir zum einen die aktuelle IP unserer VM einsehen, als auch einen DNS-Namen festlegen:



Mit folgendem Befehl können wir uns jetzt in eine Konsole der VM einloggen:

```
ssh -i <dein-schlüsselname>.pem azureuser@<ip-deiner-vm>
```

In meinem Fall (**bei dir anders**) sieht der Befehl so aus:

```
ssh -i asphost_key.pem azureuser@108.143.154.216
```

Bevor man sich zum ersten mal einloggt fragt einen SSH ob man der Gegenseite vertraut.

```
The authenticity of host '108.143.154.216 (108.143.154.216)' can't be established.
ED25519 key fingerprint is SHA256:gspN+LirQ+6X4C3m9EX1Fc6fbMxv/GQ66FPvyFnXYUU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Hier muss man **yes** eingeben bevor eine Verbindung aufgebaut wird. Für den am Anfang häufigen Fehler **Unprotected Key file** gibt es hier: <https://superuser.com/questions/1296024/windows-ssh-permissions-for-private-key-are-too-open> Lösungsansätze.

Erscheint dann eine Konsole mit grünem Text, hat der Login funktioniert.

Daten übertragen in die VM

Im nächsten Schritt müssen wir unser Projekt in die VM übertragen. Dazu eignet sich der Befehl **sftp**. Der Name steht für SSH File Transfer Protocol. Da FTP unverschlüsselt ist sollte man davon absehen es in seiner normalen Form zu verwenden.

Ähnlich wie bei SSH ist die Befehlsstruktur hier:

```
sftp -i <dein-schlüsselname>.pem azureuser@<ip-deiner-vm>
```

Dannach erscheint der Prompt **sftp>** welcher FTP-Befehle entgegen nimmt. Mit **put <dateiname>.zip** können wir eine Datei von einem lokalen Ordner in die VM verschieben.

Nachdem die Projektdateien verschoben sind können wir uns wieder mit SSH auf der VM einloggen und mit der Installation beginnen. Zunächst sollte man wie bei jeder neuen Linux-Installation ein update durchführen. Das funktioniert unter Ubuntu mit

```
sudo apt-get update -y
```

Ist das Update fertig können wir diverse Programme installieren. Wir benötigen Docker fürs Ausführen sowie unzip um die zip-Datei zu entpacken. Als kleines Hilfsprogramm noch dos2unix.

```
sudo apt install unzip dos2unix -y
```

```
sudo snap install docker
```

Nun können wir mit **unzip <dateiname>.zip** unser Projekt auspacken. Um in den darin erstellten Ordner zu wechseln gibt es das Kommando **cd <dein-projektordner>**.

compose ausführen

Das funktioniert - wie unter Windows auch - mit **docker-compose up**. Innerhalb des Containers wird dann das **Dockerfile** aufgebaut und ausgeführt.

Port 8080 in compose ummappen

Wie in einem oberen Screenshot ersichtlich werden in Azure nur die Ports 22, 80 und 443 veröffentlicht. Da unsere Anwendung jedoch auf dem Port 8080 arbeitet müssen wir den Netzwerkverkehr umleiten.

Durch verändern der docker-compose.yml-Datei ist das ganz einfach möglich. Wie der innere auf den äußeren Port umgeleitet wird, wird immer folgendermaßen angegeben: **<aussenport>:<innenport>**. Statt **8080:8080** müssen wir also **80:8080** schreiben.

So könnte in deinem Projektordner die Datei docker-compose.yml aussehen:

```
version: "3"
services:
  dotnet:
    build: .
    ports:
      - "80:8080"
```

Webservice aufrufen

Rufe die aktuelle IP (oder in meinem Fall habe ich eine Domain hinterlegt) auf und sieh nach ob eine Antwort zurückkommt. Bei mir wird folgendes JSON retourniert:

