

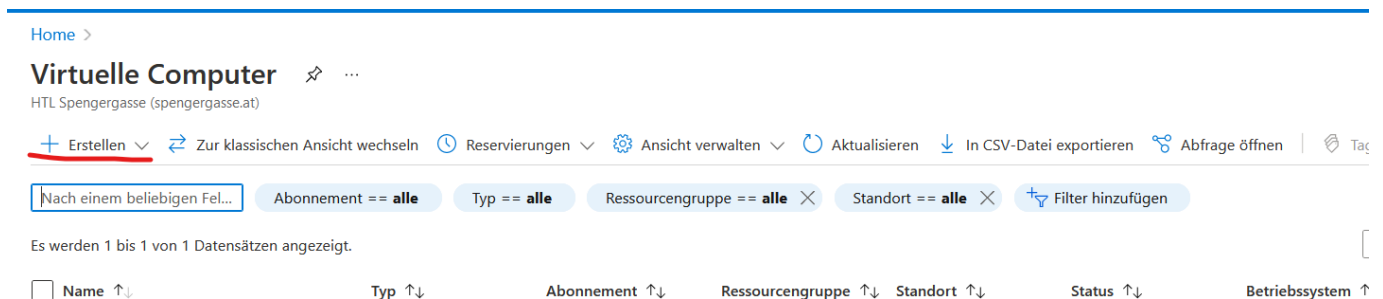
# Azure Deployment

## VM Anlegen

1. Auf [portal.azure.com](https://portal.azure.com) mit der Spengergasse-Email anmelden
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 ...

## Grundeinstellungen    Datenträger    Netzwerk    Verwaltung    Erweitert    Tags    Überprüfen + erstellen

Erstellen Sie einen virtuellen Computer, auf dem Linux oder Windows ausgeführt wird. Wählen Sie ein Image aus dem Azure Marketplace aus, oder verwenden Sie ein eigenes benutzerdefiniertes Image. Füllen Sie die Registerkarte "Grundlagen" aus, und wählen Sie dann "Überprüfen + erstellen", um einen virtuellen Computer mit Standardparametern bereitzustellen, oder überprüfen Sie zur vollständigen Anpassung die einzelnen Registerkarten. [Weitere Informationen](#)

### Projektdetails

Wählen Sie das Abonnement aus, um bereitgestellte Ressourcen und Kosten zu verwalten. Verwenden Sie Ressourcengruppen wie z. B. Ordner zum Organisieren und Verwalten all Ihrer Ressourcen.

Abonnement *	<input type="text" value="Azure subscription 1"/>
Ressourcengruppe *	<input type="text" value="4xbgm-mis"/>

[Neues Element erstellen](#)

### Instanzendetails

Name des virtuellen Computers *	<input type="text" value="misVM"/>
Region *	<input type="text" value="(Europe) West Europe (für kostenlose Dienste berechtigt)"/>
Verfügbarkeitsoptionen	<input type="text" value="Keine Infrastrukturredundanz erforderlich."/>
Sicherheitstyp	<input type="text" value="Standard"/>

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.

Image *	<input type="text" value="Ubuntu Server 20.04 LTS – Gen2 (für kostenlose Dienste berechtigt)"/>
Azure Spot-Instanz	<input type="checkbox"/>
Größe *	<input type="text" value="Standard_D2ads_v5 - 2 vcpus, 8 GiB Arbeitsspeicher (91,25 \$/Monat)"/>

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

### Administratorkonto

Authentifizierungstyp	<input checked="" type="radio"/> Öffentlicher SSH-Schlüssel <input type="radio"/> Kennwort
<div><p>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.</p></div>	
Benutzername *	<input type="text" value="azureuser"/>
Quelle für öffentlichen SSH-Schlüssel	<input type="text" value="Neues Schlüsselpaar generieren"/>
Schlüsselpaarname *	<input type="text" value="misVM_key"/>

Bei den Eingangsports brauchen wir alle verfügbaren Ports. Über Port 80 werden wir später auf unsere Webanwendung zugreifen. Über Port 22 können wir eine Konsole der VM erreichen.

Eingangsports auswählen \*

HTTP (80), HTTPS (443), SSH (22) ✓

☒ HTTP (80)

☒ HTTPS (443)

☒ SSH (22)

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

Grundeinstellungen

Datenträger

Netzwerk

Verwaltung

Erweitert

Tags

**Überprüfen + erstellen**

#### PRODUCT DETAILS

1 X Standard D2ads v5

by Microsoft

[Terms of use](#) | [Privacy policy](#)

Subscription credits apply ⓘ

**0,1250 USD/hr**

[Pricing for other VM sizes](#)

#### TERMS

By clicking "Erstellen", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the [Azure Marketplace Terms](#) for additional details.

**⚠ Sie haben SSH Port(s) für das Internet geöffnet.** Dies wird nur für Tests empfohlen. Wenn Sie diese Einstellung ändern möchten, wechseln Sie zurück zur Registerkarte "Grundlagen".

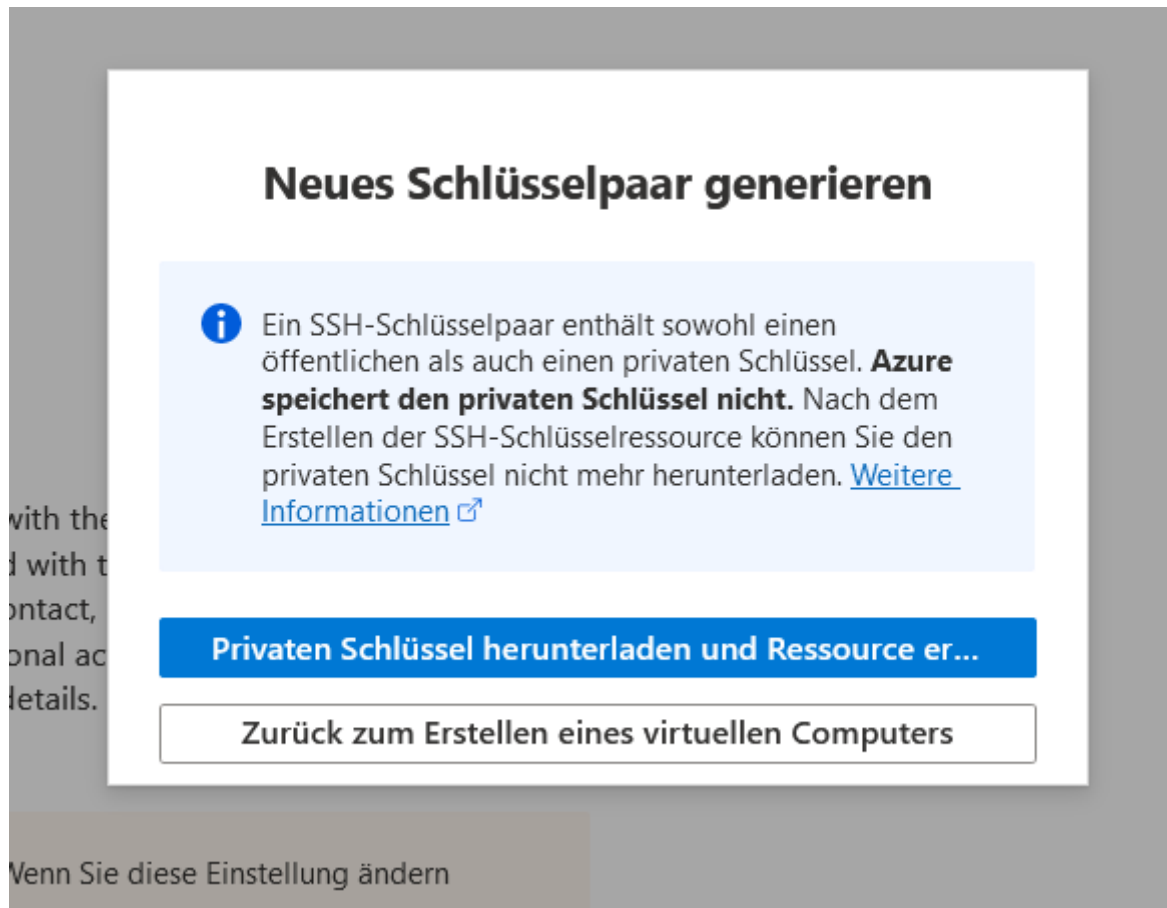
**Erstellen**

< Zurück

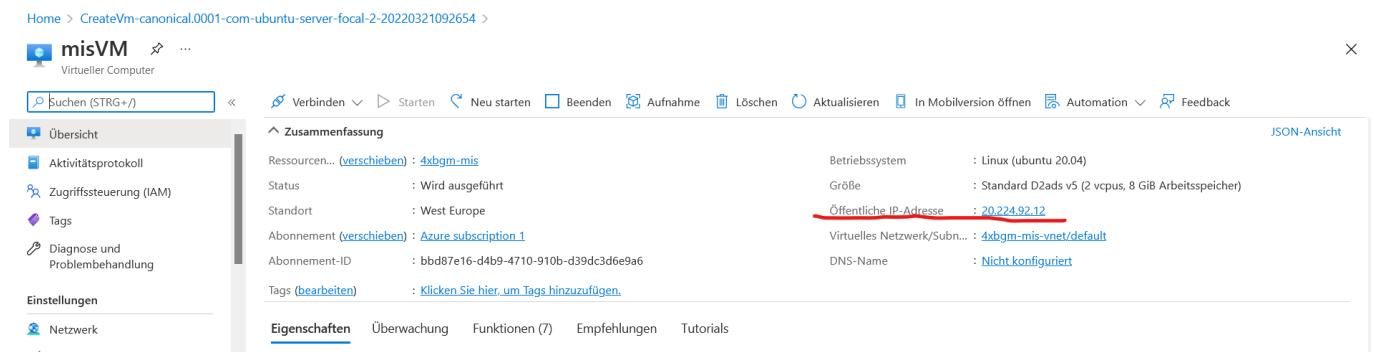
Weiter >

[Vorlage zur Automatisierung herunterladen](#)

**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 die aktuelle IP unserer VM einsehen:



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 .\misVM_key.pem azureuser@20.224.92.12
```

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

### Daten übertragen in die VM

Im nächsten Schritt müssen wir user Projekt in die VM übertragen. Dazu eignet sich der Befehl **sftp**.

Ähnlich wie bei SSH ist die Befehlsstruktur hier:

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

Dannach erschreint 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 funktionier unter Ubuntu mit

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

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

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

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

## jar-Datei bauen

Die jar-Datei kann man mit dem gewrappten maven `mvnw` bauen. Der Befehl dafür lautet:

```
./mvnw clean package -DskipTests
```

Falls die Datei nicht als ausführbare Datei deklariert ist kann man dies unter Linux mit

```
chmod +x mvnw
```

 ändern.

## compose ausführen

Sobald eine unter Linux paketierte jar-Datei vorliegt können die Container gestartet werden. Das funktioniert - wie unter Windows auch - mit `docker-compose up`

### Port 8080 in compose ummappen

Wie in einem oberen Screenshot ersichtlich werden in Azure nur die Ports 22, 80 und 443 veröffentlicht. Da unsere Spring-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 foldendermaßen angegeben: `<aussenport>:<innenport>`. Statt `8080:8080` müssen wir also `80:8080` schreiben.

## 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:

← → ↻ xbgmcloud.westeurope.cloudapp.azure.com/api/patient

JSON Rohdaten Kopfzeilen

Speichern Kopieren Alle einklappen Alle ausklappen 🔍 JSON durchsuchen

▼ 0:

- id: "123"
- text: null
- identifier: []
- active: true
- name: []
- telecom: []
- gender: "male"
- birthDate: null
- deceasedBoolean: false
- deceasedDateTime: null
- address: []

▼ 1:

- id: "643221yu"
- text: null
- ▼ identifier:
  - ▼ 0:
    - id: "123lasd"
    - use: "usual"
    - type: null
    - system: "urn:oid:1.2.36.146.595.217.0.1"
    - value: "12345"
    - period: null