

**Technische Berufsschule Zürich****TBZ**

- [KN03: IaaS - Virtuelle Server](#)
 - [A\) AWS Kurs](#)
 - [a\) Lab 4.1 - EC2](#)
 - [b\) Lab 4.2 - S3](#)
 - [B\) Zugriff mit SSH-Key](#)
 - [C\) Cloud-init Datei Verstehen](#)
 - [D\) SSH-Key und Cloud-init](#)
 - [Quellen](#)

KN03: IaaS - Virtuelle Server

Beachten Sie die [allgemeinen Informationen zu den Abgaben](#).

Wir werden nun mit einem public cloud Anbieter arbeiten. Wir verwenden Amazon Web Services (AWS). Aber alles was wir hier umsetzen, bieten auch andere public cloud Anbieter (z. B. Microsoft Azure).

Wir verwenden teilweise die AWS Schulungsumgebung für diese Kompetenz. Sie haben eine Einladung für zwei Kurse bekommen, die wir folgend verwenden werden.

Ein paar zusätzliche notwendige Grundlagen finden Sie in der Datei [Cloud Computing in Gitlab](#).

Sie erlernen den grundsätzlichen Umgang mit AWS. Anschliessend werden Sie lernen wie sie sich mit einem SSH Key einloggt, gefolgt vom Verständnis für Cloud-init. Am Schluss vermischen Sie SSH-Key mit cloud-init Konfiguration.

A) AWS Kurs

Öffnen Sie den Kurs "**AWS Academy Introduction to Cloud: Semester 1**" und arbeiten Sie sich durch die ersten 4 Module (Global Infrastructure, Structures of the Cloud, AWS Console, Virtual Servers). Sie müssen die Knowledge Checks nicht durchführen, dürfen aber.

Die ersten drei Module sind informativ. Das vierte Modul sind praktische Übungen. Sie lernen mit virtuellen Maschinen umzugehen.

a) Lab 4.1 - EC2

Führen Sie alle Schritte durch. Sie erstellen die Screenshots der verschiedenen Bereiche für Ihre Abgabe **nachdem** Sie **alle** Schritte des Moduls durchgeführt haben. Screenshots:

- HTML-Seite, inkl. URL
- Liste der EC2-Instanzen
- Details der *Web Server*-Instanz (öffentliche IP sichtbar)

- Security-Group: Liste der Inbound-Regeln.

Schreiben Sie **erklärende Worte** hinzu, so dass erklärt wird, was auf den Screenshots zu sehen ist.

b) Lab 4.2 - S3

Führen Sie alle Schritte durch. Sie erstellen die Screenshots der verschiedenen Bereiche für Ihre Abgabe **nachdem** Sie **alle** Schritte des Moduls durchgeführt haben. Screenshots:

- Liste der Buckets.
- HTML-Seite, inkl. URL.
- Liste der Dateien im Bucket.
- Eigenschaften von "Static website hosting".

Schreiben Sie **erklärende Worte** hinzu, so dass erklärt wird, was auf den Screenshots zu sehen ist.

B) Zugriff mit SSH-Key

Wechseln Sie nun den Kurs und verwenden "**Lerner Lab**". Ab sofort arbeiten Sie immer im **Lerner Lab**, ausser explizit anders angegeben.

Unter [Quellen](#) finden Sie ein Dokument, welches die Unterschiede zwischen Passwort und SSH-Key beschreibt. Lesen Sie sich ein. Wir werden grundsätzlich kein Passwort verwenden und immer SSH Keys.

Erstellen Sie nun wieder eine neue **Ubuntu**-Instanz in AWS mit den Standard-Einstellungen und den folgenden zusätzlich:

- Bei "Key pair" erstellen Sie **zwei** Key-Value Pairs! Geben Sie Namen wie <Ihr-Name>-1 und <Ihr-Name>-2. Stellen Sie sicher, dass Sie beide Schlüssel herunterladen.
- Bei "Key pair" wählen Sie dann den **ersten** Key aus um weiterzufahren.

Speicherort der privaten (und öffentlichen) Schlüssel muss ein Pfad sein, der eingeschränkte Rechte hat (nur Ihr persönlicher Account). Normalerweise wird der Pfad `C:\Users\<ihrBenutzer>\.ssh` verwendet.

Der Befehl für den Zugriff sieht nun folgendermassen aus. Der Parameter `-o` verhindert, dass Sie ein Timeout kriegen.

```
ssh <user>@<server> -i <path-to-privatekey>\<private-key-file>.pem -o
ServerAliveInterval=30
# Standard-Benutzername ist ubuntu!
# Beispiel: ssh ubuntu@121.12.3.1 -i c:\Users\nussle\.ssh\yves-1.pem -o
ServerAliveInterval=30
```

Zeigen Sie, dass sie sich **nur** mit **dem ausgewählten** Schlüssel einloggen können.

Hinweis: AWS speichert ihren privaten Schlüssel **nicht**. Sie können den nicht wiederherstellen. Auf dem Ubuntu-Server liegt immer nur der öffentliche Schlüssel. Seien Sie also vorsichtig und verlieren Sie den Schlüssel nicht.

Abgaben:

- Screenshot mit dem ssh-Befehl und des Resultats unter Verwendung des **ersten** Schlüssels
- Screenshot mit dem ssh-Befehl und des Resultats unter Verwendung des **zweiten** Schlüssels
- Screenshot der Instanz-Detail (oder Liste), so dass der verwendete Schlüssel sichtbar ist.

C) Cloud-init Datei Verstehen

Die wichtigsten Informationen zu Cloud-init und YAML [finden Sie in der Theorie](#).

Laden Sie diese [cloud-init Datei](#) herunter. Erklären Sie alle Zeilen der Cloud-init Datei in folgendem Schema.

Achtung: Viele Werte sind nicht definiert von cloud-init, sondern von Linux/Ubuntu. z.B. *sudo*: *<sudo-Regeln>*. Die Sudo-Regeln sind definiert in Linux (sudoers-file).

```
#cloud-config
users: # Collection von Benutzern
- name: ubuntu # Der Benutzername des ersten Benutzers
  sudo: ALL=(ALL) NOPASSWD:ALL # sudo-Regeln für diesen Benutzer
#...
#...
```

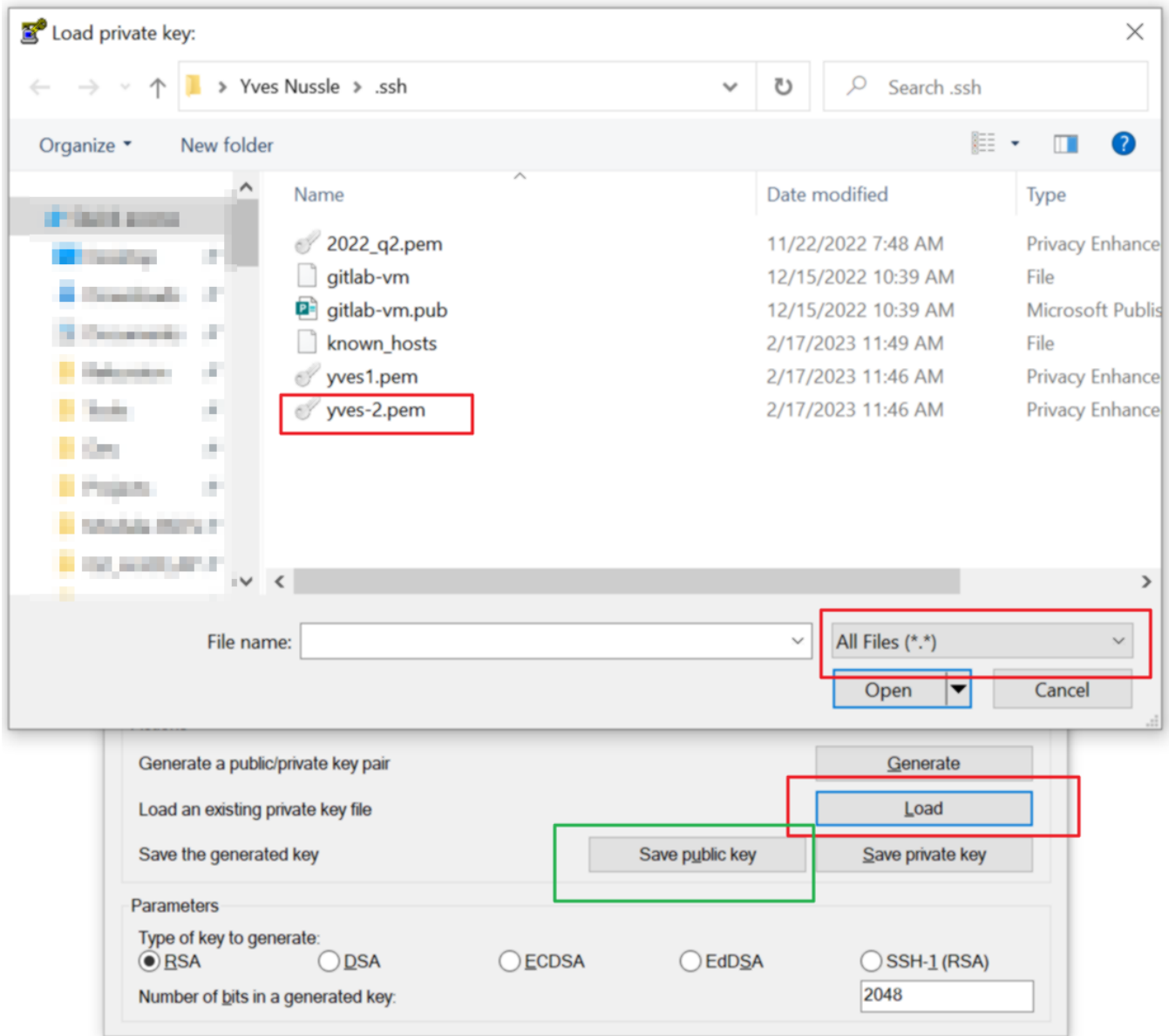
Sie werden eine Cloud-init Konfiguration in Teil D) verwenden. Was die Cloud-init Datei alles kann, sehen Sie dann in der KN03.

Abgaben:

- Die dokumentierte YAML-**Datei** (nicht als Screenshot)

D) SSH-Key und Cloud-init

Anstatt, dass Sie den SSH-Key im GUI auswählen, können Sie ihn auch im Cloud-Init mitgeben. Dies werden wir folgend tun. Installieren Sie puttygen, falls Sie es noch nicht haben (unter Quellen finden Sie den Link). Extrahieren Sie den öffentlichen Schlüssel (für beide Keys) wie folgt.



Erstellen Sie eine neue Instanz in AWS mit den folgenden Einstellungen:

- Ubuntu 22.04
- Bei "Key pair" verwenden Sie ihren **zweiten** Schlüssel. Sie müssen hier etwas auswählen. AWS lässt Sie nicht weiter ohne Auswahl.
- Keine Änderungen bei den restlichen Einstellungen.
- aber: verwenden Sie die [cloud-init Datei](#) und ersetzen Sie den Schlüssel mit ihrem **ersten** öffentlich Schlüssel. Beachten Sie das Format `ssh-rsa <ihr-Schlüssel-ohne-zeilenumbrüche> aws-key`
- Sie können ihre Cloud-init Konfiguration einfach in dem Feld "user data" in der Sektion "Advanced details" reinkopieren - ähnlich wie Sie es in Teil A) gemacht haben.

Hinweis: In der vorherigen Aufgaben hatten Sie in diesem Feld ("user data") ein Bash-Skript ausgeführt. Dies weil die erste Zeile auf ein Bash-Skript hinweist. Wenn Sie nun aber ihre YAML-Datei in das Feld kopieren, wird AWS automatisch das richtige installieren.

Sie verwenden nun also beide Schlüssel (einmal via cloud-init, einmal via Eigenschaften im GUI).

Zeigen Sie nun, dass Sie **nur mit dem Key aus der Cloud-Init Datei** einloggen können.

Abgaben:

- Ihre angepasste Cloud-init Konfiguration als **Datei** im Git-Repository.
- Ein Screenshot der Details oder Liste der Instanz, welcher den **verwendeten** Key zeigt.
- Screenshot mit dem ssh-Befehl und des Resultats unter Verwendung des **ersten** Schlüssels.
- Screenshot mit dem ssh-Befehl und des Resultats unter Verwendung des **zweiten** Schlüssels.

ACHTUNG:

- **Ab sofort fügen Sie immer** den **öffentlichen Schlüssel für Lehrpersonen** hinzu (Sie können jeweils mehrere öffentliche Schlüssel verwenden).

Häufige Fehlerquellen:

- Sie vergessen die erste Zeile in der Cloud-init Datei **#cloud-config**. Diese ist notwendig!
- Einrückungen wurden nicht korrekt erstellt. Das Format können Sie hier überprüfen: <https://www.yamllint.com/>.
- Sie können auf dem Server die Log-Datei für Fehler durchsuchen: **/var/log/cloud-init-output.log**
- Das Format des SSH-Schlüssels ist falsch. Korrekt ist: **ssh-key XXXXXXXXXX aws-key**.

Quellen

Thorntech: [Passwords vs. SSH keys - what's better for authentication?](#)

puttygen: [Puttygen download](#)