



KN06: Skalierung

- A) Installation App (20%)
- B) Vertikale Skalierung (10%)
- C) Horizontale Skalierung (40%)
- D) Auto Scaling (30%)
- E) Zusatz für Profis

KN06: Skalierung

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

In dieser Kompetenz werden Sie eine Instanzen horizontal, vertikal und automatisiert skalieren.

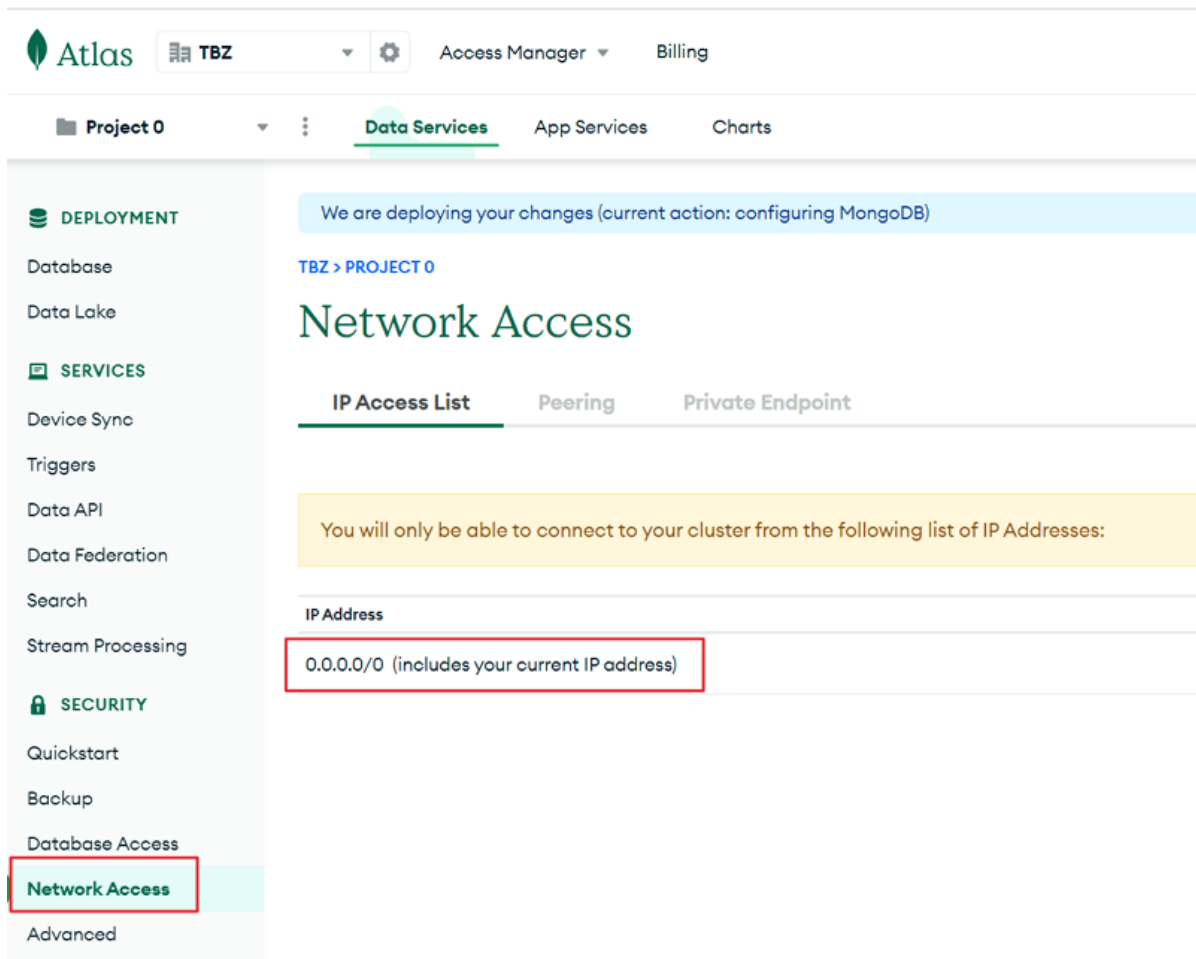
[Lesen Sie sich in die Theorie ein, auch in die Dokumente in der Quelle.](#)

A) Installation App (20%)

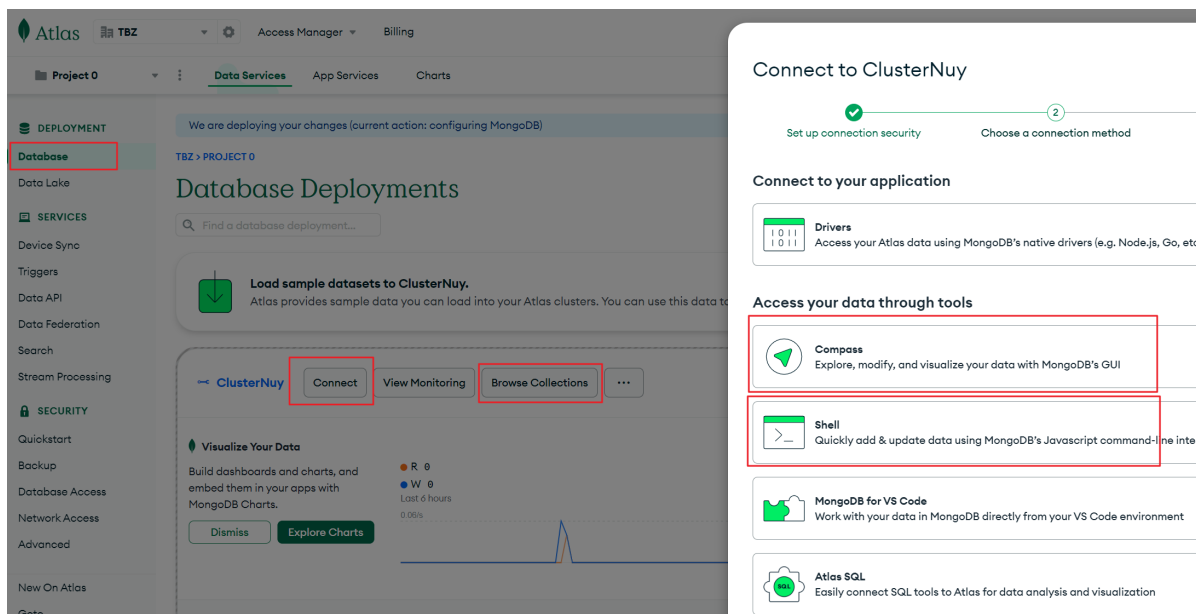
Im Ordner für [KN06](#) finden Sie Cloud-init Dateien. Diese Dateien installieren einen Datenbank- und einen Webserver für entweder eine .NET Applikation. Speziell dabei ist, dass sowohl die Applikation mit eigenem Webserver kommt, die auf Port 5000 hören. Bisher hatten wir Apache als Web-Server verwendet. Nun werden wir Nginx einsetzen. Sie möchten ja nicht, dass ihre Applikation nur über den Port 5000 aufgerufen werden kann, sondern über den normalen Port 80. Nginx wird daher als [Reverse Proxy](#) eingesetzt, welcher die Anfragen weiterleitet.

a) MongoDB als Datenbank einrichten

Diese neue Applikation verwendet MongoDB als Datenbank und wird mit MongoDB Atlas gehostet (SaaS). Sie benötigen also daher keinen zusätzlichen Datenbank-Server. Richten Sie sich ein [kostenloses Konto ein](#). Stellen Sie sicher, dass auf diese Datenbank zugegriffen werden kann, indem Sie den entsprechenden IP Range angeben. Den gleichen Schritt mussten Sie mit mariaDB ausführen in KN03.



Ihr Cloud-Init Skript für den Webserver wird via Shell (MongoDb Shell) auf die Datenbank zugreifen. Falls Sie ein graphisches UI zusätzlich möchten, verwenden Sie MongoDB Compass. Mit der Option *Browse Collections* haben Sie aber auch die Möglichkeit Ihre Datenbank(en) via Browser anzuschauen.



b) Web Server installieren auf AWS

Installieren Sie nun wieder einen Web-Server, aber nun mit einer komplexeren Applikation. Verwenden Sie die Standardeinstellungen für die Instanz, ohne statische IP. Als Grundlage haben Sie einen Web-API .NET Code mit Swagger aktiviert.

Verwenden Sie die vorgegebene Cloud-Init Konfiguration. **Achtung:** Sie müssen gewisse Platzhalter ersetzen (markiert mit den Zeichen `[[...]]`). Stellen Sie sicher, dass Sie die korrekte IP aus Ihrem Netzwerk-Interface verwenden. Versuchen Sie die Cloud-Init zu verstehen und damit wird auch klar was Sie ersetzen.

Wichtig: Es kann sein, dass Sie ein paar Minuten warten müssen bis alles installiert ist.

.NET Applikation

- Infos zum Nachlesen: [https://learn.microsoft.com/en-us/aspnet/core/host-and-deploy/linux-n](https://learn.microsoft.com/en-us/aspnet/core/host-and-deploy/linux-nginx?view=aspnetcore-7.0&tabs=linux-ubuntu)
[ginx?view=aspnetcore-7.0&tabs=linux-ubuntu](https://learn.microsoft.com/en-us/aspnet/core/host-and-deploy/linux-nginx?view=aspnetcore-7.0&tabs=linux-ubuntu)
- Swagger-URL: <http://ihre-ip/swagger/index.html>

Abgaben:

- Kurze Erklärung in **eigenen Worten** was ein Reverse Proxy ist.
- Screenshot der zeigt, dass Sie die Swagger-URL aufrufen können.
- Screenshot der zeigt, dass sie den Endpoint `GetProducts` aufrufen können (via Swagger) und auch ein korrektes Resultat bekommen.
- Screenshot einer der MongoDB Collections mit Auszug aus dem Inhalt.

B) Vertikale Skalierung (10%)

Skalieren Sie nun ihren Webserver und erweitern Sie die Disk auf **20GB**. **Erklären** Sie die Schritte die notwendig waren. Geht dies im laufenden Betrieb?

Skalieren Sie nun auch weitere Ressourcen und Verwenden Sie die den Instanztyp **t2.medium**. **Erklären** Sie die Schritte, die notwendig waren. Geht dies im laufenden Betrieb?

Abgaben:

- Vorher-Nachher Screenshots der Instanz-Ressourcen
- Erklärungen.

C) Horizontale Skalierung (40%)

Erstellen Sie einen Load Balancer und fügen Sie diesem **zwei** Instanzen des Web Servers hinzu. Suchen Sie **selbstständig** nach Ressourcen. Überdenken Sie ihre anderen Objekte wie *statischen IPs, Sicherheitsgruppen*, etc. Was macht nun Sinn in ihrer neuen Umgebung? Richten Sie Ihre Umgebung entsprechend ein.

Rufen Sie nun die Load-Balancer URL auf und zeigen Sie, dass Swagger und der Endpoint auch mit dieser URL funktioniert. Sie haben nun eine Domain und keine IP mehr. Nehmen sie an, dass diese Applikation unter der URL *app.tbz-m346.ch* verfügbar sein soll. Wie müssten Sie den DNS konfigurieren, damit dies funktioniert?

Abgaben:

- Erklärungen wie oben verlangt in Git.
- Sie zeigen ihre laufende Umgebung der Lehrperson.

- Seien Sie bereit Auskunft zu erteilen über ihr Vorgehen und die erstellten Objekte wie *Load Balancer, Target Group, Health Check, IPs, Sicherheitsgruppen, Listener u. a.*

D) Auto Scaling (30%)

Erstellen Sie ein Auto-Scaling, welches mit ihrem Load Balancer zusammenarbeitet. Als Ziel sollen jeweils 2 Instanzen verfügbar sein, maximal aber 5. Bearbeiten Sie diese Aufgabe aber nur, wenn die *Health Checks* bei C) grün sind.

Testen Sie, ob Ihre Autoskalierung funktioniert, indem Sie die bestehenden Webserver einfach herunterfahren.

Abgaben:

- Zeigen Sie ihre laufende Umgebung der Lehrperson. Sie dürfen die Umgebung aus C) hier erweitern, ohne, dass Sie vorher abgeben.
- Seien Sie bereit Auskunft zu erteilen über ihr Vorgehen und die erstellten Objekte wie *Template, Health Checks, u.a.*

E) Zusatz für Profis

In Teil B hatten Sie das Cloud-Init angepasst und Variablen ersetzt. Schauen Sie sich das Cloud-Init nochmals genau an. Welcher Teil macht hier überhaupt keinen Sinn in einer produktiven Umgebung?

Tipp: Die Datenbank verändert sich mit der Zeit.