

Arbeit 1: Evaluation eines Cloud-Providers

Silvan Lendi, Leonardo Ravani

18.03.2025

Contents

1. Arbeitsauftrag 1	2
Vorstellung Railway	2
Zu beantwortende Fragen	2
2. Arbeitsauftrag 2	3
OSSM-Analyse	3
3. Arbeitsauftrag 3	4
Zu beantwortende Fragen	4
Vergleichstabelle	4
4. Auftrag 4 - Analyse Preisrecherche.	6
Zu beantwortende Fragen	6
5. Arbeitsauftrag 5	8
Zu beantwortende Fragen	8
6. Empfehlungen	9
tmpdoc	9

1. Arbeitsauftrag 1

Vorstellung Railway

Railway ist eine Cloud-Plattform die einem erlaubt, vereinfacht Applikationen zu bauen, deployen und zu skalieren. Services lassen sich über Docker-images starten oder aber auch über source code aus einem Github Repository. Ebenso lassen sich mit wenigen Klicks Datenbanken hochfahren und mit Services vernetzen.

Railway bietet einen modernen, intuitiven visuellen Canvas welcher Übersicht über die Services bietet und gleichzeitig die Oberfläche bereitstellt seine Services zu konfigurieren.

Zu beantwortende Fragen

Frage: Können alle Teammitglieder die Applikation deployen?

Antwort: Ja, alle Teammitglieder können die Applikation starten, entfernen und auch bearbeiten.

Frage: Können Sie Benutzer und Zugriffsrechte fürs Deployment verwalten?

Antwort: Nein. Man kann User zum Projekt einladen welche entweder read-only oder read-write access haben. Eine granulare einstellung ist nicht möglich.

Frage: Bietet der Provider die Möglichkeit z.B. ein GitHub-/GitLab-Repository zu verbinden und dieses automatisch zu deployen?

Antwort: Ja, das ist mehr oder weniger der hauptaspekt von Railway

Frage: Weshalb haben Sie diesen Provider für die Evaluation ausgewählt?

Antwort: Railway ist sehr einfach zu bedienen und schnell auf zu setzen, zumindest beim Deployment eines Github Projektes. Weiter gibt es eine relativ grosszügige "Free Trial" bei der man 500MB RAM und geteilte VCPUs benutzen kann. Zuletzt war uns beiden dieser Cloud-Provider unbekannt welches die Arbeit spannender gestaltete.

Frage: Welche Möglichkeiten haben Sie, die Applikation zu starten und zu stoppen?

Antwort: Auf der Website railway.com hat man die Möglichkeit das Deployment einer Applikation zu entfernen und wieder zurück zu laden. Dies ist sehr unschön und die Webseite bietet ansonsten keine weiteren Möglichkeiten in der Trial-Version. Wenn man allerdings das CLI für eine lokale Instanz verwendet kann man mit "railway up" für Start und "railway down" für Stop die Kontrolle etwas besser übernehmen.

Frage: Gibt es die Möglichkeit, das Deployment zu automatisieren? Welche Schritte wären dazu nötig?

Antwort: Ja, out of the box wird Railway ein GitHub-Projekt sofort deployen und mit automatisierte Deployments konfigurieren. Railway kann automatisch Änderungen im Repository erkennen (z.B. bei einem neuen commit) und die Applikation neu deployen.

Frage: Wo können Sie das Log der Applikation betrachten? Sehen Sie auch die einzelnen HTTP Requests und Responses?

Antwort: Wenn man auf dem Dashboard auf der Railway Webseite das deployte Projekt anklickt, kann man die laufende Instanz anklicken und sieht da nebst weiteren Details einen Tab für je Build-, Deploy- und HTTP-Logs.

Frage: Worin sehen Sie die Vor- und Nachteile Ihres Providers verglichen mit a9s?

Antwort: Die Vorteile sind: Benutzerfreundlichkeit - Applikation lässt sich einfach und schnell aufsetzen und überwachen. Automatisierung - Projekte sind von Anfang an automatisiert. Logs - Die Logdateien sind übersichtlich und einfach einsehbar im Web-Portal. Kosten - Für kleine Projekte ist die Verwendung absolut kostenlos.

Die Nachteile sind: Einstellungsmöglichkeiten im "Backend" - Railway ist eine Managed Plattform,

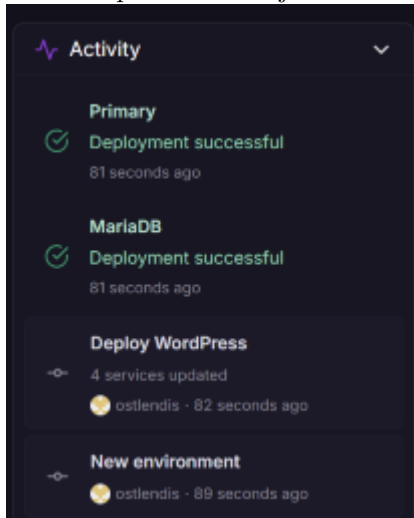
die die Infrastruktur komplett abstrahiert und bietet keine Möglichkeit, unterliegende Instanzen oder Netzwerkeinstellungen direkt zu konfigurieren.

2. Arbeitsauftrag 2

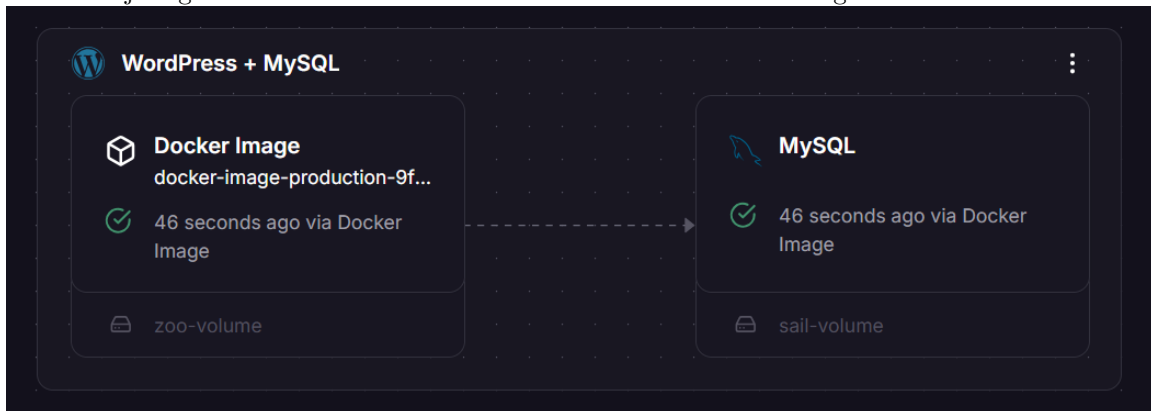
OSSM-Analyse

On-Demand Der On-Demand Aspekt ist auf Railway klar gewährleistet. Nach dem Anmelden oder registrieren kann man direkt loslegen indem ein Template gestartet wird oder ein Github-Repo verknüpft. Kurze Zeit später (bis zu 2 Minuten) ist die Applikation funktionsfähig.

Zum Beispiel lässt sich jederzeit das Wordpress Template starten:



Innerhalb Sekunden und ohne warten lässt sich ein Service starten. Es wurde zuerst von einem leeren Projekt gestartet und schliesslich sind alle Services funktionsfähig.



Self-Service Bei Railway lassen sich alle On-Demand Services via Self-Service selbst managen. Man hat gewisse Einschränkungen wenn man sich auf dem Trial-Plan befindet im Gegensatz zu einem bezahlten Plan befindet. Man müsste also zuerst seinen Plan upgraden, will man einen Service vom bezahlten Plan nutzen. Allerdings lässt sich auch dies innerhalb von kurzer Zeit ohne direkte Kommunikation mit Railway machen. Man kann also beliebig Services starten, ohne sich mit dem Provider absprechen zu müssen.

Dies lässt sich ebenfalls aus den obigen Bildern entnehmen, da in kurzer Zeit ein neues Projekt angelegt wurde, und danach Datenbank- und Webserver hochgefahren wurden welche kurze Zeit

später vollständig funktionieren.

Skalierbarkeit Bei Railway lässt sich sowohl horizontal als auch vertikal skalieren. Vertikale Skalierbarkeit ist durch die Anzahl CPU-Kerne und die Menge an Arbeitsspeicher erreichbar, ist aber eingeschränkt, auf welchem Zahl-Plan man momentan fährt. Ist man gratis unterwegs, hat man keine Möglichkeit vertikal zu skalieren. Beim Hobby und Pro Plan sind 8GB Arbeitsspeicher und 8 CPU's respektive 32GB Arbeitsspeicher und 32 CPU's pro Service verfügbar. Siehe Pläne. Die Horizontale Skalierbarkeit im Vergleich zu grösseren Anbietern auch minimalistisch. Ist kein Dateisystem am Service angehängt und es sind keine Cronjobs definiert, lassen sich die Services manuell um eine gewünschte Anzahl replizieren und somit Skalieren. Gleichzeitig gibt es die Möglichkeit, einen Service in den Serverless-Mode zu schalten. Bei diesem Modus werden anhand von verschiedenen Kriterien die Aktivität am Service gemessen und je nach Last die Anzahl Replikas hoch und hinunter gefahren. Siehe Scaling

Messbarkeit Die mit dem Hosten verbundenen Kosten sind sehr transparent und einfach gestaltet. Bei Arbeitsspeicher, Prozessorkernen und Festplattenspeicher zahlt man pro Menge mal die Zeit. Ausgehendes Traffic bezahlt man pro Kilobyte. Siehe Pricing

3. Arbeitsauftrag 3

Zu beantwortende Fragen

Frage: Wieviel RAM steht Ihrer Cloud-Instanz maximal zur Verfügung?

Antwort: Maximal 21 megabyte

Frage: Von welchem Hersteller stammt die JVM?

Antwort: Oracle Corporation

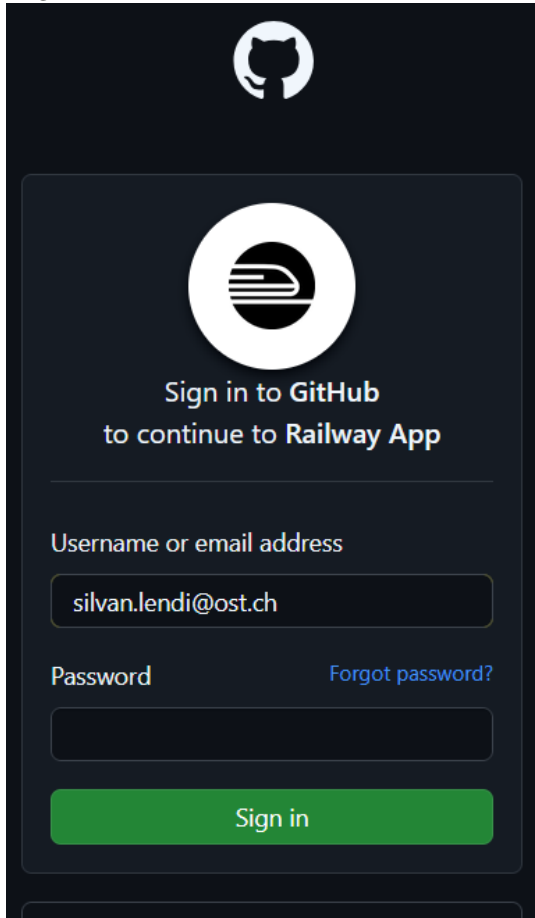
Frage: Welche IP-Adresse hat der Server laut der Self-Information Applikation? Antwort: 10.250.10.254

Vergleichstabelle

Railway	a9s
Registrierung und Anmeldung	

Railway

Login via Github auf der Website



a9s

Registrierung via Email. Vorteil: kein weiterer Account nötig. | Nachteil: Account ist abhängig von GitHub. |

Deployment

Über GUI auf dem Canvas gewünschtes Repo auswählen und automatisch deployen.

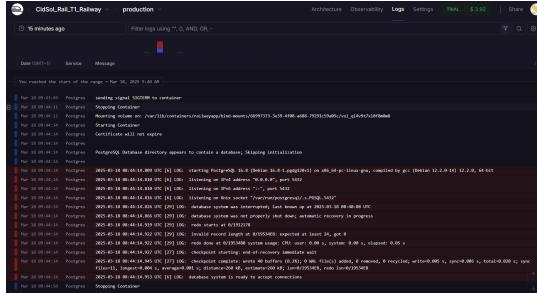
Vorteile: Railway benötigt keine CLI (ist aber eine Möglichkeit). Build passiert automatisch und funktioniert.

Nachteile: Keine.

Operations (Start, Stop, Logs)

Railway

Starten eines Services mittels Deployment. Logs sind im Deployment oder im Log-Menü sichtbar. Stoppen lässt sich ein Service über das Deployment: Deployment entfernen zum Stoppen und redeployen, um den Service wieder zu starten.



logs image

Antworten zu den Fragen

Public Domain am Service anhängen und URL aufrufen. Informationen entnehmen.

a9s

Vorteile: Keine CLI benötigt.

Initiales Starten ist wesentlich einfacher mittels Canvas. Detaillierte und aggregierte Logs.

Nachteil: Start-Stopp eines Services ist versteckt und umständlich.

Vorteil: Point-and-click statt IP von der CLI kopieren.

Nachteil: Zusätzlicher Schritt, Domain dem Service hinzuzufügen.

4. Auftrag 4 - Analyse Preisrecherche.

Zu beantwortende Fragen

Welche Preismodelle stehen zur Verfügung? Einfachheitshalber gibt es nur 3 Subscription-Pläne, "Hobby" für 5/Monat, "Pro" für 20/Monat und "Enterprise" für einen custom Betrag im Monat welcher nicht genau deklariert wurde. Die Pläne bieten folgende Limitationen:

Plan	RAM	CPU	Ephemeral Storage	Volume Storage
Trial	0.5 GB	2 vCPU	1 GB	0.5 GB
Hobby	8 GB	8 vCPU	10 GB	5 GB
Pro	32 GB	32 vCPU	100 GB	50 GB**
Enterprise	128 GB	64 vCPU	100 GB	2 TB**

Zusätzlich zu den Subscription Kosten gibt es noch Kosten anhand der Nutzung:

Resource	Resource Price
RAM	\$10 / GB / month (\$0.000231 / GB / minute)
CPU	\$20 / vCPU / month (\$0.000463 / vCPU / minute)
Network Egress	\$0.05 / GB (\$0.000000047683716 / KB)
(Optional Add-on) Volume Storage	\$0.15 / GB / month (\$0.000003472222222 / GB / minute)

Zuletzt ist es auch möglich anstelle einer Subscription ein “Committed Spending” zu beziehen welches viel höhere Monatliche Kosten hat, jedoch keine weiteren dynamischen Kosten, solange man im Rahmen des Pakets bleibt:

Add-on Feature	Commitment Spend	Description
2 TB RAM / 112 vCPU	\$500/month	Access to increased computing resources at a committed monthly spend level.
RBAC	\$500/month	Role-Based Access Control to manage user permissions and access.
SLOs	\$500/month	Service Level Objectives to ensure and track application performance.
90-day log history	\$500/month	Extended log retention for better historical analysis and auditing.
HIPAA BAAs	\$1,000/month	HIPAA Business Associate Agreements for compliant health data handling. Requires a year commitment paid monthly
Dedicated Hosts	\$10,000/month	Custom dedicated infrastructure for enhanced performance and control.

Wie kann bezahlt werden? Wenn man die Subscription ändern möchte muss man dazu die Kreditkarte angeben, wobei andere Zahlungsmethoden nicht möglich sind. Wenn man allerdings die Enterprise Subscription bezahlen möchte ist auch per Rechnung gestattet.

Gibt es ein Gratis-Angebot für Einsteiger? Ja es gibt eine “Full” und eine “Limited” Trial-Version. Der einzige unterschied zwischen den beiden ist, dass die Limited-Version keinen Code deployen kann sondern nur Datenbanken. Wenn man seinen Account verifiziert wird man von der Limited- zu der Full-Version heraufgestuft. Die Einschränkungen sind bereits in der vorherigen Fragen ersichtlich mit 0.5GB RAM, 2 vCPU, 1GB Ephemeral Storage und 0.5GB Volume Storage.

Welche Ressourcen werden abgerechnet? Es werden als Dynamische Kosten zusätzlich zur Subscription immer RAM, CPU, Network Egress und falls man noch zusätzlichen Volume Storage möchte auch der Storage abgerechnet.

Was kostet Sie das Hosting der Anwendung aus der Ausgangslage über drei Jahre? 500\$ * 12 für ersten jedes Monats (Enterprise minimum monatlicher Preis)

Eigentlich wäre die Hobby Subscription genug aber bei einem Downgrade (zweiter Tag jedes Monats) werden die Änderungen erst nächsten Monat angepasst, somit muss man bei der Enterprise Version bleiben. Die Enterprise Version hat auf der Webseite keinen Preis und muss durch die Besprechung des optimalen Plans mit einem Mitarbeiter der Railway Firma per E-Mail herausgefunden werden.

Es sollte kein zusätzlicher Speicher benötigt sein allerdings ist dies ebenfalls nicht auf der Webseite deklariert wie viel beim Mindestangebot der Enterprise Version dabei ist.

10\$ * 8 * 12 für RAM

20\$ * 4 * 12 || 20\$ * 8 * 12 für CPU Cores wobei nicht genau beschrieben steht wie viele Cores ein vCPU hat und sein kann, dass es nur ein halber Core eines physischen CPUs ist. Bei der finalen Rechnung gehe ich davon aus, dass ein vCPU ein Core eines physischen CPUs ist.

0.05\$ * 1000 für Netzwerk Traffic

Für den ersten Tag jedes Monats müssen zusätzlich noch (20\$ * 28 (CPU) + 10\$ * 120 (RAM)) / 30 * 12 (Es ist nirgendwo geschrieben wie die Berechnung eines einzigen Tags angegangen wird. Die Vermutung ist, dass es durch 30 mal die Anzahl Tage gerechnet wird).

Somit wäre der Gesamtpreis 3224\$ im Jahr, was für die Ausgangslage von 3 Jahren 9672\$ bedeutet.

Welche Schwierigkeiten treten bei dem Vergleich auf? Der Vergleich zwischen einer normalen Subscription und den "Committed Spend Tiers" ist schwierig da bei diesen Tiers sowie bei der Enterprise Version die genauen Kosten und was alles enthalten ist nicht beschrieben wird. Wenn man etwas grösseres in die Cloud stellen möchte muss man mit einer solchen personalisierten Lösung arbeiten. Weiter ist es mühsam, dass man beim Downgrade der Subscription die teurere bis ende Monat behält. Somit muss man schlussendlich viel zu viel für eine kleine Applikation mit kurzen Peaks bezahlen.

5. Arbeitsauftrag 5

Zu beantwortende Fragen

Gibt es ein oder mehrere Service Level Objectives (SLOs)? Weder auf railway.com noch auf docs.railway.com gab es irgendwelche Informationen zu den SLO oder SLA. Die einzige Erwähnung war im Bereich Support in den Docs, wobei es nur um die acknowledgement time eines Support-Falls geht. Nach einigen Tagen erhielten wir noch keine Antwort auf eine E-Mail bei welcher wir Kontakt aufnehmen wollten um weitere Details über die SLO/SLA zu erfahren.

Werden die fünf «Questions to Ask» aus dem SLA White Paper von Dimension Data beantwortet? Nein, da kein SLA verfügbar ist.

Fehlen aus Ihrer Sicht im White Paper von Dimension Data bestimmte SLA-Evaluationskriterien, würden Sie also andere Fragen stellen? Wir würden höchstens noch zusätzliche Fragen stellen. Informationen welche wichtig wären und ebenfalls nirgendwo auffindbar waren sind die Themen, Datenverlust durch den Anbieter, Sicherheit und Privacy und weitere

details zur Performance. Bei der Performance wäre noch interessant wie latency und throughput gehandhabt wird.

Schätzen Sie Existenz und Einhaltung von SLAs als wichtig ein a) bei der Entscheidung, mit einer Anwendung in die Cloud zu gehen und b) einen Provider auszuwählen? Wir empfinden die Existenz und vorallem die Einhaltung der SLA als sehr wichtig bei beiden Punkten aber nur zu bestimmten Situationen. Zuerst bei der Wahl ob man überhaupt in die cloud möchte: Wenn ein Service permanent online sein muss und dies bei einem Cloud Provider fast nie gegeben ist (oder zu teuer sein könnte), könnte es sich mehr lohnen das Problem selbst in die Hand zu nehmen. Wenn man allerdings eine Applikation hat welche durch die Cloud zugänglich sein sollte, jedoch sehr selten gebraucht wird, spielt auch kein Punkt in der SLA eine Rolle. Die SLA ist ein Vertrag und man sollte davon ausgehen können, dass dieser auf Seite Provider eingehalten wird. Bei dem zweiten Teil, kommt es ebenfalls darauf an ob die Applikation etwas lockerer mit den Anforderungen umgehen kann oder nicht. Hier gilt ebenfalls, wenn eine Applikation zum Beispiel immer zugänglich sein sollte, sollte man sich einen Provider aussuchen welcher möglichst viele nines in den SLA angibt. Bei beiden Fragen spielt die Applikation die Hauptrolle, wobei die SLA jedoch auch zum Zuge kommt.

Gibt es neben den SLAs weitere Dokumente, die man als zukünftiger Cloud-Consumer sorgfältig lesen sollte? Die Datenschutzrichtlinien sowie die Nutzungsbedingungen und Sicherheitsrichtlinien.

6. Empfehlungen

Mit dem einfachen und schnellen setup eignet es sich hervorragend für kleinere Projekte. Dabei spielt es keine Rolle ob es eine native cloud- oder eine lokale-App oder sogar nur eine Datenbank ist. Dadurch, dass das System selbst aus einem Git Projekt eine Instanz erstellt und laufen lässt, benötigt man keine Änderungen an einer lokalen Applikation zu machen.

Mit den personalisierten Angeboten kann es sich auch für grössere Projekte lohnen, dazu muss man allerdings viel und detailliert mit dem Provider kommunizieren da nur sehr wenige Informationen online zur Verfügung stehen.

Falls man jedoch hohen Wert auf Leistung, Verfügbarkeit, etc. setzt, ist die fehlende SLA ein Dorn im Auge, wobei man eher zu einem grossen Anbieter wechseln sollte.

tmpdoc

- Logged into railway with github account
- looked at nextjs example
- created github repo and copied files from previous java exercise into repo
- created railway project and selected github repo as source
- building and deploying went automatic, zero input and configuration required, WTF
- created a domain so app is now accessible from web