

Gruppenarbeit 1 - Cloud Fundamentals beim Provider

Alexander van Schie & Oli Dias

March 11, 2019

Contents

1 Hands-On: Hello (Cloud) World	3
1.1 Installationsanleitung	3
1.1.1 Openshift Account erstellen	3
1.1.2 Erstellen, Builden und Deployment der Applikation	4
2 Analyse: OSSM-Definition	5
2.1 On-demand & Self-service	6
2.2 Scalable	6
2.3 Measurable	7
3 Konzept: Cloud Computing Patterns	8
4 Hands-On: Self Information	9
5 Analyse: Preisrecherche	9
6 Analyse: Preisvergleich eigenes Hosting, IaaS und PaaS	9

1 Hands-On: Hello (Cloud) World

1.1 Installationsanleitung

Das Ziel ist es, eine Asp.Net-Core Hello-World Applikation mittels Openshift Online zu builden und deployen. Am Schluss dieser Installationsanleitung sollte dies möglich sein.

Für das Einrichten von Openshift Online müssen grob folgende Schritte durchgeführt werden:

- Account und Projekt auf der Plattform erstellen
- GitHub Repository der Asp.Net-Core Applikation mit Openshift Projekt verbinden
- Applikation auf Openshift builden
- Applikation auf Openshift deployen

1.1.1 Openshift Account erstellen

Zuerst muss ein Account auf <https://manage.openshift.com/signin> erstellt werden. Danach kann zwischen den in Abbildung 1 vorgeschlagenen Plänen ausgewählt werden. Wir benutzen den Openshift Online Starter Plan.

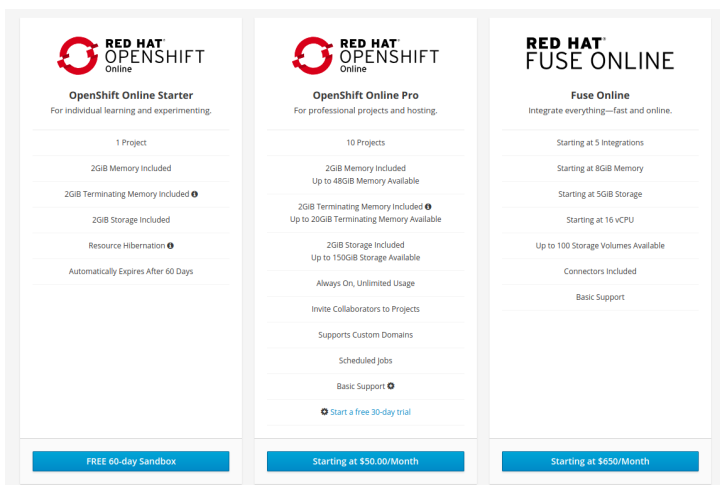


Figure 1: Gewählter Plan Openshift

Dass der Account verifiziert werden kann, muss eine Telefonnummer hinterlegt werden, auf welche darauffolgend einen Bestätigungscode zugeschickt wird.

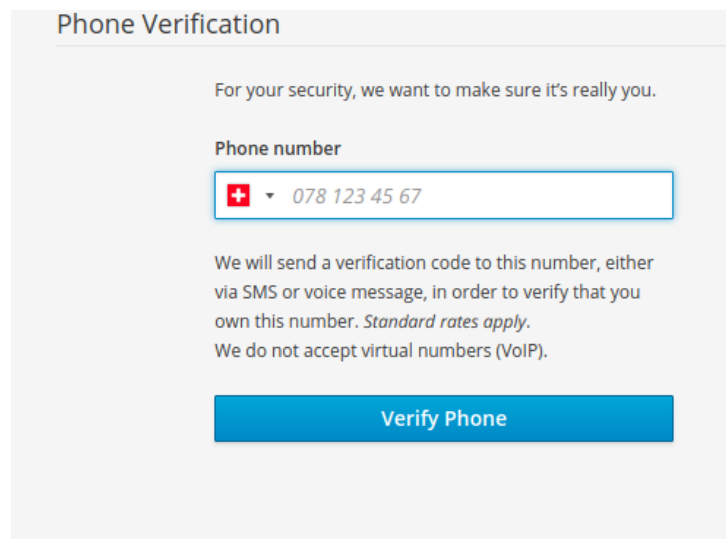
The image shows a 'Phone Verification' form. At the top, it says 'Phone Verification'. Below that, a message reads: 'For your security, we want to make sure it's really you.' There is a section labeled 'Phone number' with a dropdown menu showing a Swiss flag and the number '078 123 45 67'. Below the input field, a message states: 'We will send a verification code to this number, either via SMS or voice message, in order to verify that you own this number. Standard rates apply. We do not accept virtual numbers (VoIP).' At the bottom, there is a blue button labeled 'Verify Phone'.

Figure 2: Telefon Verifikation

Wurde diese eingegeben und verifiziert, erscheint eine Übersicht über das bestellte Produkt wie in Abbildung 3 angezeigt. Daraufhin kann die Subscription bestätigt werden. Kurz nach dem Bestätigen sollte ein Bestätigungsmail eintreffen. Nach dem Bestätigen dieser kann bereits die Web Console geöffnet werden. Es wird ein Katalog mit allen Produkten von Openshift Online dargestellt (Abbildung 4).

Das Erstellen des Openshift Accounts ist somit abgeschlossen und die Plattform ist für das Builden und Deployen von Applikationen bereit.

1.1.2 Erstellen, Builden und Deployment der Applikation

In der Web-Console können wir nun auf .NET Core Projekt klicken. Daraufhin erscheint ein Wizard, dem wir Schritt für Schritt folgen können. Falls das GitHub-Repository schon während dem Wizard hinzugefügt werden soll, muss es bereits existieren und sichtbar sein.

Entsprechende Felder müssen gemäss Abbildung 5 ausgefüllt sein. Vorsicht ist bei der .NET Version geboten; wir verwenden die Version 2.2 von .NET Core.

Sobald das Projekt in Openshift erstellt wurde, startet der Build. Womöglich schlägt der Build aufgrund von fehlender .s2i-Konfiguration (Source-2-Image) fehl. Um diesen Fehler zu beheben, muss Openshift gesagt werden, wo das Startup-Projekt liegt. Dazu muss ein Ordner und File mit dem Namen .s2i/environment erstellt werden. Dies beinhaltet folgendes:

```
1 DOTNET_STARTUP_PROJECT=HelloWorld-netcore/HelloWorld  
-netcore.csproj
```

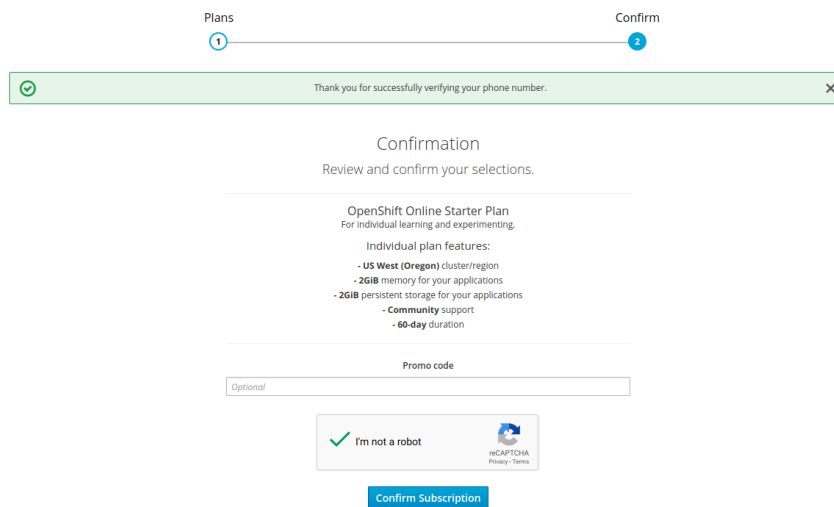


Figure 3: Übersicht des abgeschlossenen Plans

Wichtig ist weiter zu beachten, dass die .net Versionen (dotnet sowie NuGet-Pakete) mit denjenigen von Openshift Cloud kompatibel sind.

War der Build erfolgreich, muss noch das Deployment konfiguriert werden. Dazu kommt ein weiteres File mit dem Namen `run` ins `.s2i` Verzeichnis. Darin muss die Applikation noch gestartet werden. Dies funktioniert so:

```
1 exec dotnet run
```

Ist auch dieser Schritt vollbracht, kann im Control Panel des Projektes zu Deployments → Routes navigiert werden. Dort erscheint eine Tabelle, wo der Hostname bereits angegeben ist und womit nun die Asp.Net-Core Applikation vom Internet her erreichbar ist.

2 Analyse: OSSM-Definition

Damit sich jemand als Cloud Computing Provider ausgeben kann, sollten folgende Charakteristiken erfüllt sein:

- On-demand
- Self-service
- Scalable

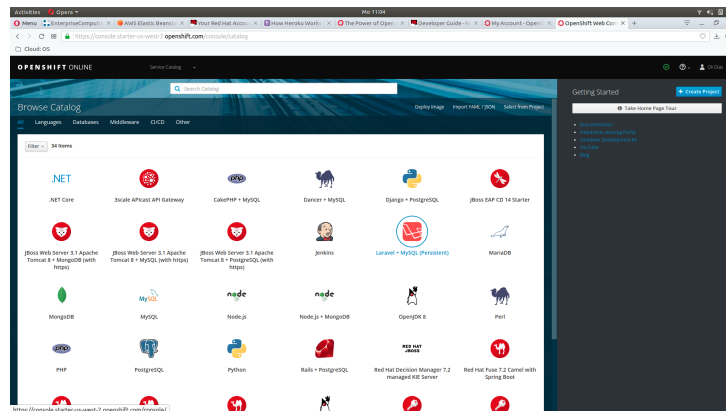


Figure 4: Katalog von OpenShift

- Measurable

In den folgenden Kapiteln erläutern wir, wie OpenShift diese umsetzt.

2.1 On-demand & Self-service

Auf der Startseite findet man eine Katalog aller möglichen Projekttypen. Nach einem Klick auf den gewünschten Projekttyp erscheint ein Dialog, in welchem die spezifische Konfiguration vorgenommen werden kann. Gleichzeitig wird geprüft, ob die definierte Konfiguration plausibel ist. Ist dies der Fall, kann das Projekt erstellt werden. Nach wenigen Sekunden ist das Projekt unter der Rubrik "My Projects" ersichtlich.

Die Projektübersicht bietet neben einigen generellen Informationen die Möglichkeit zum Build und Deployment.

2.2 Scalable

Die Nutzung von OpenShift erfordert, dass man sich für ein Abonnement entscheidet. Neben den kostenlosen Einführungsangeboten bedarf die langfristige Nutzung das Abonnement "OpenShift Online Pro". Im Standard bekommt man hierfür folgende Ressourcen:

- 10 Projects
- 2 GB Memory
- 2 GB Terminating Memory
- 2 GB Storage

.NET Core

✕

Information

Configuration

Results

1

2

3

* Project name

helloworld-aspnetcore

A unique name for the project.

Project Display Name

helloworld-aspnetcore

Project Description

Simple asp.net-core hello world application

Version

2.2 — latest

* Application Name

helloworld-aspnetcore

* Git Repository

https://github.com/olddias/Cloud-netCore_HelloWorld

Cancel

< Back

Create

Um mehr als 10 Projekte zu verwalten bedarf es einem neuen Abonnement. Falls mehr Arbeitsspeicher oder Speicher nötig ist, kann das aktuelle Abonnement angepasst werden, was natürlich einen Einfluss auf den Preis hat. Trotzdem gibt es folgende Begrenzungen:

- Arbeitsspeicher: maximal 48 GB
- Speicher: maximal 150 GB

2.3 Measurable

Die aktuelle Nutzung der Ressourcen kann lediglich auf Projektstufe eingesehen werden. Diese Übersicht ist ziemlich einfach gehalten, lediglich genutzter Arbeitsspeicher und Speicher werden im Verhältnis zum Maximum angezeigt.

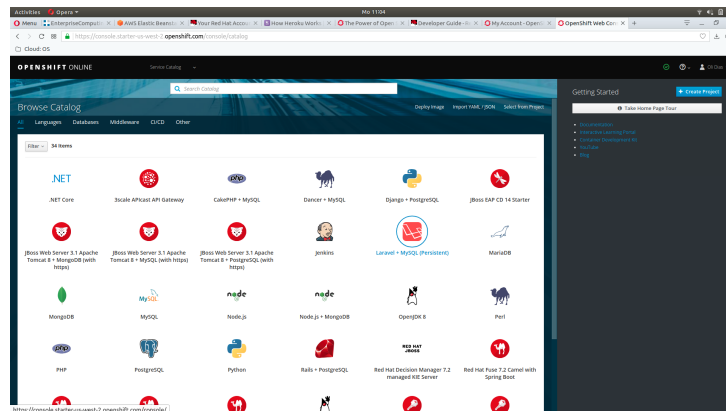
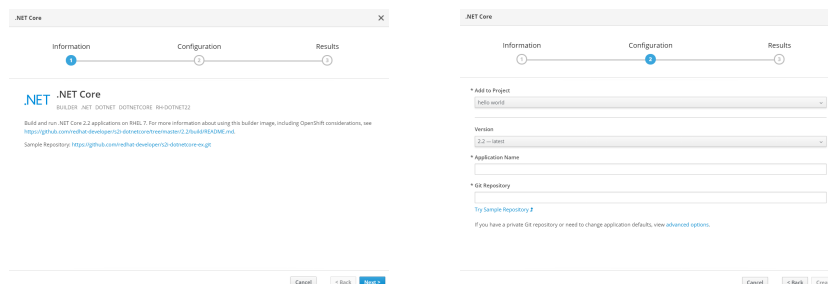


Figure 7: Auswahlkatalog aller möglichen Projekttypen



(a) label 1

(b) label 2

Figure 8: Setup Dialog

3 Konzept: Cloud Computing Patterns

Openshift bietet Entwicklern mit dem Konzept PaaS eine Plattform an, auf welcher eine App relative einfach deployed werden kann. Dies bringt den Vorteil, dass der Entwickler sich nicht mit der Komplexität der Building-/Deploying Infrastruktur auseinandersetzen muss. Die Applikation läuft anschließend auf sogenannten "Pods", welche vergleichbar mit Docker-Container sind. Einerseits kann die Anzahl Pod's pro Projekt manuell festgelegt und geändert werden. Zudem gibt es die Möglichkeit für Pod Autoscaling. Sobald ein Pod bis zu einem gewissen Grad ausgelastet wird, kommt ein zusätzlicher Pod in Aktion, sofern die maximale Anzahl definierter Pod's nicht erreicht wurde.

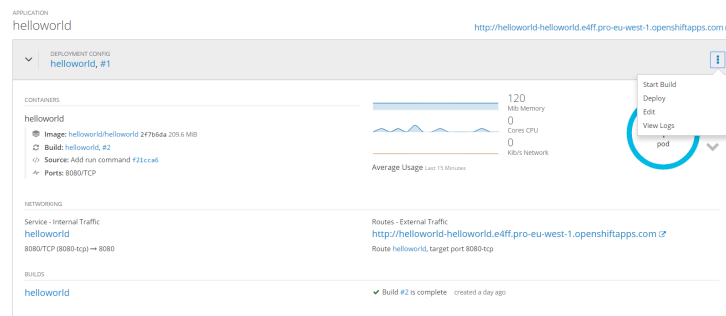


Figure 9: Ansicht der Projektübersicht

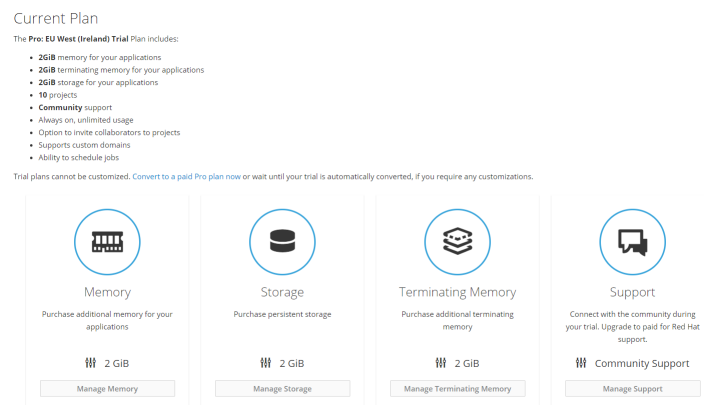


Figure 10: Verwaltung der Ressourcen eines Abonnements

4 Hands-On: Self Information

5 Analyse: Preisrecherche

6 Analyse: Preisvergleich eigenes Hosting, IaaS und PaaS

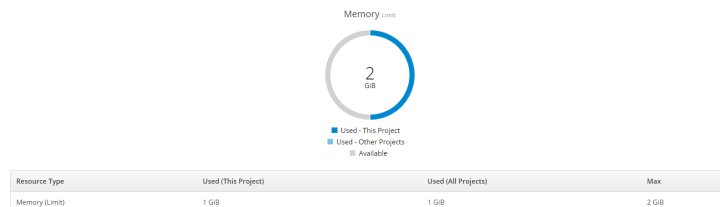


Figure 11: Übersicht der genutzten Ressourcen

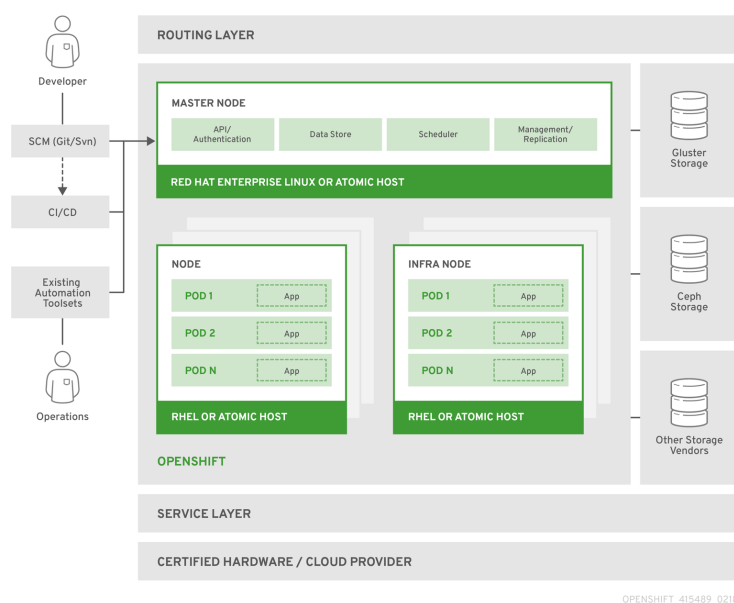


Figure 12: Architekturübersicht der OpenShift-Plattform