

Wordpress Stack bei AWS

Auftrag

Wordpress-Stack bei AWS

Mittwoch, 2. November 2022 08:59

Um einzelne IaaS- und auch PaaS-Lösungen von AWS näher kennenzulernen, nehmen Sie Wordpress dort in Betrieb. Im Prinzip eignen sich viele Open Source Applikationen dafür, um einen klassischen LAMP-Stack in folgendem Umfang in Betrieb zu nehmen - Wordpress wird hier einfach exemplarisch verwendet, da es sehr einfach zu installieren ist und es nicht um die Installation an sich geht, sondern um die Betriebsarchitektur des Stacks:

- Application Server - EC2-Instanz mit Linux, Apache Webserver, PHP
- Database Server - RDS (Relational Database Service) mit MySQL
- Storage - Vorerst in der EC2-Instanz, optionale Auslagerung auf EBS

Auftrag

1. Installation von Wordpress auf EC2 und RDS

RDS ist eine managed Database, d.h. im Gegensatz zu EC2, wo die volle Verantwortung über den OS- und Application-Layer bei Ihnen liegt, nutzen wir für die Datenbank eine PaaS-Lösung.

Verwenden Sie hierzu das folgende Tutorial:

<https://aws.amazon.com/de/getting-started/hands-on/deploy-wordpress-with-amazon-rds/>

Für die Schnellen: Binden Sie EBS als **zusätzlichen** Storage ein und konfigurieren Sie die Installation so, dass die Daten darauf gespeichert werden.

Erklärung der Unterschiede: <https://aws.amazon.com/de/premiumsupport/knowledge-center/instance-store-vs-ebs/>

Für die Einbindung gibt es kein Tutorial, aber da findet man einiges an Hinweisen im Internet, v.a. in der offiziellen Dokumentation von AWS.

2. Einrichtung des Backups der EC2-Instanz

Da dies nun eine öffentlich erreichbare Web-Applikation ist, kann es natürlich sein, dass sie gehackt wird oder aus irgendwelchen Gründen nicht mehr funktioniert. Hierfür ist es wichtig, regelmässige Backups einzurichten. Beginnen Sie mit demjenigen für EC2:

https://aws.amazon.com/de/getting-started/hands-on/amazon-ec2-backup-and-restore-using-aws-backup/?TRK=gs_card

Testen Sie den Restore! Ein Backup, von dem Sie nicht wissen, ob der Restore funktioniert, ist nichts wert.

3. Einrichtung des Backups der RDS

Auch die DB muss regelmässig gesichert werden. Richten Sie dort ebenfalls (ohne Tutorial) das Backup ein und testen Sie den Restore.

Für die Schnellen: Falls Sie EBS als Storage eingebunden haben, muss dieser natürlich auch gesichert werden. Richten Sie dies ein - danach erübrigt sich eigentlich das Backup der

EC2-Instanz, da bei Löschung dieser, kein Datenverlust auftauchen sollte (bei richtiger Konfiguration).

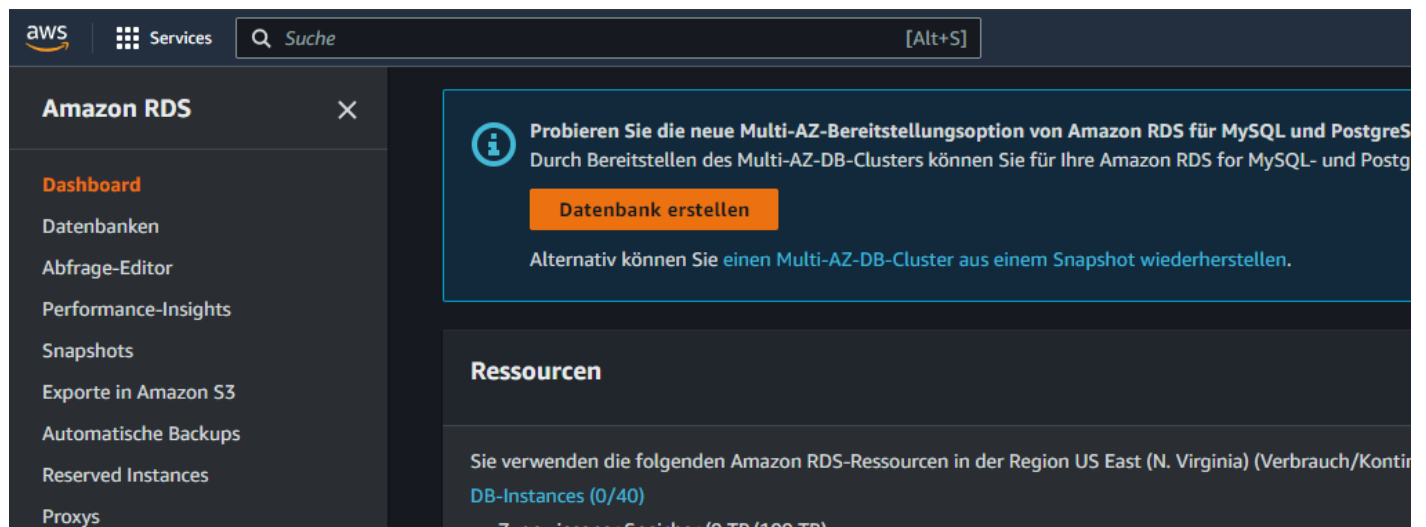
4. Reflexionsfrage

Wenn Sie alle Punkte erledigt haben, spricht man von einer sogenannten *Stateless Architecture*. Erklären Sie in **eigenen** Worten, was dies bedeutet:

WordPress mit Amazon RDS bereitstellen







Erstellen einer MySQL-Datenbank mit RDS

Um eine Datenbank zu erstellen, muss unter Amazon Cloud nach RDS suchen und dann auf «Datenbank erstellen» gehen.



Als Engine-Typ benutzen wir MySQL

Engine-Typ [Informationen](#)

<input type="radio"/> Amazon Aurora 	<input checked="" type="radio"/> MySQL 	<input type="radio"/> MariaDB 
<input type="radio"/> PostgreSQL 	<input type="radio"/> Oracle 	<input type="radio"/> Microsoft SQL Server 

Als Vorlage benutzen man „Kostenloses Kontingent“

Vorlagen
Wählen Sie eine Beispielvorlage aus, die zu Ihrem Anwendungsfall passt.

<input type="radio"/> Produktion Verwenden Sie Standardwerte für hohe Verfügbarkeit und schnelle, kontinuierliche Leistung.	<input type="radio"/> Entwicklung/Test Diese Instance dient zur Verwendung für die Entwicklung außerhalb einer Produktionsumgebung.	<input checked="" type="radio"/> Kostenloses Kontingent Mit dem kostenlosen Kontingent für RDS können Sie neue Anwendungen entwickeln, vorhandene Anwendungen testen oder praktische Erfahrung mit Amazon RDS sammeln. Informationen
---	---	---

In den Einstellungen ändern man die DB-Instance-Kennung zu „WordPress“, den Hauptbenutzernamen zu „admin“ (falls anders) und das Passwort kann man selbst wählen.

Einstellungen

DB-Instance-Kennung [Informationen](#)

Geben Sie einen Namen für Ihre DB-Instance ein. Der Name muss für alle DB-Instances Ihres AWS-Kontos in der aktuellen AWS-Region eindeutig sein.

WordPress

Bei der DB-Instance-Kennung wird zwischen Groß- und Kleinschreibung unterschieden, sie wird jedoch komplett in Kleinbuchstaben gespeichert (wie z. B. in „meinedbinstance“). Einschränkungen: Zwischen 1 und 60 alphanumerische Zeichen oder Bindestriche. Muss mit einem Buchstaben beginnen. Darf keine zwei aufeinanderfolgenden Bindestriche enthalten. Darf nicht mit einem Bindestrich enden.

▼ Einstellungen für Anmeldeinformationen

Hauptbenutzername [Informationen](#)

Geben Sie eine Anmelde-ID für den Hauptbenutzer Ihrer DB-Instance ein.

admin

1 bis 16 alphanumerische Zeichen. Muss mit einem Buchstaben beginnen.

☐ Ein Passwort automatisch erstellen

Amazon RDS kann ein Passwort für Sie generieren oder Sie können Ihr eigenes Passwort angeben.

Hauptpasswort [Informationen](#)

.....



Einschränkungen: Mindestens 8 druckbare ASCII-Zeichen. Folgende Zeichen dürfen nicht enthalten sein: / (Schrägstrich), ' (einzelne Anführungszeichen), " (doppelte Anführungszeichen) und @ (At-Zeichen).

Passwort bestätigen [Informationen](#)

.....



Als nächstes öffnet man den Reiter „Zusätzliche Konfiguration“

► Zusätzliche Konfiguration

Datenbankoptionen, Verschlüsselung aktiviert, Datensicherung aktiviert, Backtrack deaktiviert, Wartung, CloudWatch Logs, Löschschutz deaktiviert.

Dort ändert man noch den „Anfänglicher Datenbankname“ zu wordpress

▼ Zusätzliche Konfiguration

Datenbankoptionen, Verschlüsselung aktiviert, Datensicherung aktiviert, Backtrack deaktiviert, Wartung, CloudWatch Logs, Löschschutz deaktiviert.

Datenbankoptionen

Anfänglicher Datenbankname [Informationen](#)

Wenn Sie keinen Datenbanknamen angeben, wird von Amazon RDS keine Datenbank erstellt.

Wenn dies alles Konfiguriert ist drückt man auf Datenbank erstellen


Geschätzte monatliche Kosten

Das kostenlose Kontingent für Amazon RDS steht Ihnen 12 Monate lang zur Verfügung. Mit dem kostenlosen Kontingent können Sie in jedem Kalendermonat die folgenden Amazon RDS-Ressourcen kostenlos verwenden:

- 750 Stunden für Amazon RDS in einer Single-AZ-Instance vom Typ „db.t2.micro“, „db.t3.micro“ oder „db.t4g.micro“.
- 20 GB Universal-Speicher (SSD)
- 20 GB Speicher für automatische Sicherungen und alle vom Benutzer gestarteten DB-Snapshots

[Weitere Informationen zum kostenlosen Nutzungskontingent für AWS](#) [↗](#)

Wenn das kostenlose Kontingent abgelaufen ist oder Ihre Nutzung der Anwendung das kostenlose Nutzungskontingent überschreitet, zahlen Sie einfach die üblichen nutzungsabhängigen Service-Tarife. Einzelheiten dazu finden Sie auf der Seite mit den [Preisangaben für Amazon RDS](#) [↗](#).

 Sie müssen sicherstellen, dass Sie über alle erforderlichen Rechte für Produkte oder Services von Drittanbietern verfügen, die Sie mit AWS-Services verwenden.

Abbrechen

Datenbank erstellen

Erstellen einer EC2-Instance

Um eine EC2-Instanz zu erstellen, sucht man in der Suche nach EC2 und geht dann wenn man es gefunden hat auf „Instance starten“

aws Services Suche [Alt+S]

New EC2 Experience Tell us what you think

EC2-Dashboard

- Globale Ansicht von EC2
- Ereignisse
- Tags
- Beschränkungen
- ▼ **Instances**
 - Instances
 - Instance-Typen
 - Startvorlagen
 - Spot-Anfragen
 - Savings Plans
 - Reserved Instances
 - Dedicated Hosts
 - Geplante Instances
 - Kapazitätsreservierungen

Ressourcen

Sie verwenden die folgenden Amazon EC2-Ressourcen in der Region USA Ost (Nord-Virginia):

Instances (ausgeführt)	0	Dedicated Hosts	0
Load Balancers	0	Platzierungsgruppen	0
Snapshots	0	Volumes	0

📘 Mithilfe des AWS Launch Wizard können Sie ganz einfach Verfügbarkeitsgruppen von Microsoft SQL Server Always On konfigurieren, bereitstellen und ih

Instance starten

Starten Sie zunächst eine Amazon EC2-Instance. Dabei handelt es sich um einen virtuellen Server in der Cloud.

Instance starten ▼ **Einen Server migrieren** ↗

Hinweis: Ihre Instances werden in der Region USA Ost (Nord-Virginia) gestartet

Als Name kann man nehmen, was man will, Ich habe mich für „EC2-WordPress“ entschieden.

Name und Tags Info

Name

EC2-WordPress [Weitere Tags hinzufügen](#)

Als Image kann man im Grunde jedes nehmen, Ich würde aber „Amazon Linux“ benutzen da die Befehle auf einem anderen OS eine andere Syntax haben.

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

S

>

[Weitere AMIs durchsuchen](#)

Einschließlich von AMIs aus AWS, Marketplace und der Community

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type Zur kostenlosen Nutzung berechtigt

ami-0b0dcb5067f052a63 (64-Bit (x86)) / ami-01b5ec3ed8678d8b7 (64-Bit (Arm))

Virtualisierung: hvm ENA-aktiviert: true Root-Gerätetyp: ebs

Für die Kostenlose Benutzung würde ich die „t2.micro“ nehmen da diese 750 Stunden pro Monat kostenlos benutzbar ist.

▼ Instance-Typ [Info](#)

Instance-Typ

t2.micro

Zur kostenlosen Nutzung berechtigt

Familie: t2 1 vCPU 1 GiB Arbeitsspeicher

On-Demand Linux Preise: 0.0116 USD pro Stunde

On-Demand Windows Preise: 0.0162 USD pro Stunde

[Instance-Typen
vergleichen](#)

Als Schlüsselpaar (SSH Key) kann man entweder einen neuen erstellen oder einen bereits vorhandenes verwenden.

▼ Schlüsselpaar (Anmeldung) [Info](#)

Sie können ein Schlüsselpaar verwenden, um eine sichere Verbindung zu Ihrer Instance herzustellen. Stellen Sie sicher, dass Sie Zugriff auf das ausgewählte Schlüsselpaar haben, bevor Sie die Instance starten.

Schlüsselpaarname - Pflichtfeld

WordPress



[Neues Schlüsselpaar
erstellen](#)

In den Netzwerkeinstellungen ändert man zur Einfachheit halber den Namen und die Beschreibung zur Erkennung der Sicherheitsgruppe. Konfiguriere dann 2 Sicherheitsgruppenregel:

1. Um mit SSH auf die VM zu kommen (Am besten der Quelltyp «Meine IP», man kann aber natürlich auch «Überall» benutzen)
2. Die zweite wäre das http Traffic von überall auf die spätere Wordpressseite Zugriff hat.

Netzwerkeinstellungen

Name der Sicherheitsgruppe - *pflichtfeld*

EC2-WordPress

Diese Sicherheitsgruppe wird allen Netzwerkschnittstellen hinzugefügt. Der Name kann nicht bearbeitet werden, nachdem die Sicherheitsgruppe erstellt wurde. Die maximale Länge beträgt 255 Zeichen. Gültige Zeichen: a-z, A-Z, 0-9, Leerzeichen und _-:/()#,@!+= &;[]\$*

Beschreibung - *pflichtfeld* [Info](#)

Sicherheitsgruppe fuer EC2-WordPress Instanz

Eingehende Sicherheitsgruppenregeln

▼ Sicherheitsgruppenregel 1 (TCP, 22, 85.2.134.238/32)

Entfernen

Typ [Info](#)

ssh ▼

Protokoll [Info](#)

TCP

Portbereich [Info](#)

22

Quellentyp [Info](#)

Meine IP ▼

Name [Info](#)

Q CIDR, Präfixliste oder Sicherhei

85.2.134.238/32 X

Beschreibung - *optional* [Info](#)

z. B. SSH für Admin Desktop

▼ Sicherheitsgruppenregel 2 (TCP, 80, 0.0.0.0/0, WordPress)

Entfernen

Typ [Info](#)

HTTP ▼

Protokoll [Info](#)

TCP

Portbereich [Info](#)

80

Quellentyp [Info](#)

Überall ▼

Quelle [Info](#)

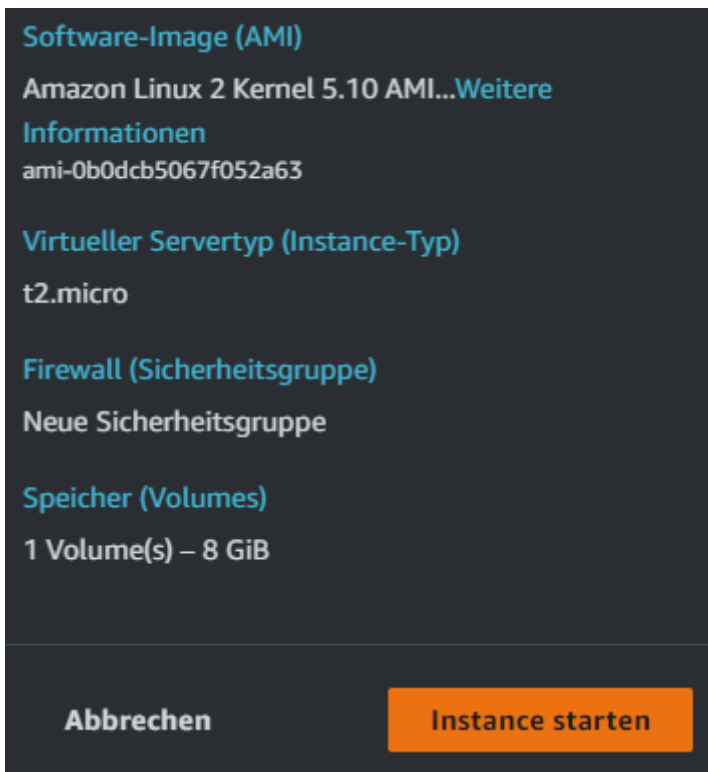
Q CIDR, Präfixliste oder Sicherhei

0.0.0.0/0 X

Beschreibung - *optional* [Info](#)

WordPress

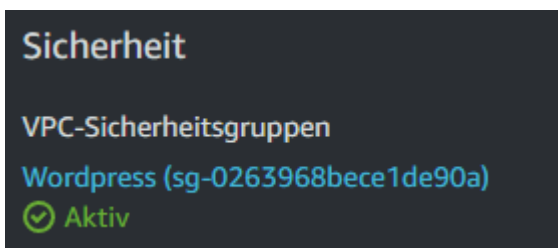
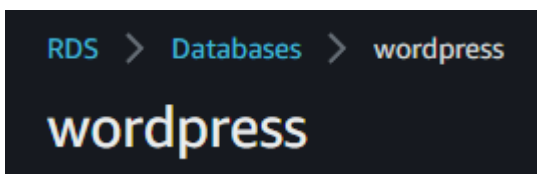
Am Schluss sollten die Konfigurationen ungefähr so angezeigt werden. Danach kann man auf „Instance starten“ drücken.



Konfiguration ihrer RDS-Datenbank

Datenbank Sicherheitsgruppe anpassen

Um die Sicherheitsgruppe der Datenbank so anzupassen das nur die EC2-Instanz eine Verbindung aufbauen kann muss man auf die Datenbank gehen und dann nach VPC-Sicherheitsgruppen suchen. Wenn man diese gefunden hat, drückt man auf diese drauf und fügt eine neue Regel für den eingehenden Datenverkehr hinzu



Diese Regel sollte ungefähr so aussehen wie die auf der linken Seite.

Regeln für eingehenden Datenverkehr bearbeiten Info

Regeln für eingehenden Datenverkehr steuern den eingehenden Datenverkehr, der die Instance erreichen darf.

Regeln für eingehenden Datenverkehr Info

ID der Sicherheitsgruppenregel	Typ <small>Info</small>	Protokoll <small>Info</small>	Portbereich <small>Info</small>	Quelle <small>Info</small>	Beschreibung – optional <small>Info</small>	
-	<div>MYSQL/Aurora</div>	<div>TCP</div>	<div>3306</div>	<div>Benutzerdefi...</div>	<div>Q</div>	<div>EC2-WordPress td MySQL Database</div>
				<div>sg-03d9795ce685f1790 X</div>		<div>Löschen</div>

Regel hinzufügen

Abbrechen

Änderungen in der Vorschau anzeigen

Regeln speichern

Per SSH auf EC2 Instanz Verbinden

Wie man sich mit der EC2 Instanz verbindet, kann man auf der linken Seite erkennen.

```
C:\Users\manuel>ssh -i "C:\Users\manuel\Downloads\WordPress.pem" ec2-user@44.202.160.95
The authenticity of host '44.202.160.95 (44.202.160.95)' can't be established.
ED25519 key fingerprint is SHA256:Hf06IaxmRmjWLj2ZKHbNyv9W0SLv535wjLFGWzd6HvM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '44.202.160.95' (ED25519) to the list of known hosts.
```

```
 _ | _ | _ )
 _ | ( _ /   Amazon Linux 2 AMI
---|\---|---
```

```
https://aws.amazon.com/amazon-linux-2/
19 package(s) needed for security, out of 38 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-94-181 ~]$ |
```

MySQL installieren, auf Datenbank verbinden und User erstellen

```
sudo yum update
sudo yum install -y mysql
```

Nachdem man das Repository geupdatet und MySQL installiert hat kann man sich auf die Datenbank verbinden, um den Wordpress MySQL User zu erstellen. Dafür muss man die Endpunkt Adresse herausfinden die man in der AWS-Konsole unter RDS/Databases/"WordPress Datenbank" findet.

RDS > Databases > wordpress

wordpress

Endpunkt und Port

Endpunkt

wordpress.csj4fgsm0wsp.us-east-1.rds.amazonaws.com

Port

3306

Um die Endpunkt Adresse hinzuzufügen, benutzt man den Befehl: `export MYSQL_HOST=`

```
[ec2-user@ip-172-31-94-181 ~]$ export MYSQL_HOST=wordpress.csj4fgsm0wsp.us-east-1.rds.amazonaws.com
[ec2-user@ip-172-31-94-181 ~]$ |
```

Mit diesen Befehlen verbindet man sich auf die Datenbank und erstellt einen Benutzer mit dem Namen „wordpress“ der „ALL PRIVILEGES“ hat.

```
[ec2-user@ip-172-31-94-181 ~]$ mysql --user=admin --password=yaMM5Bi6AmbL8GT3 wordpress
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 22
Server version: 8.0.28 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [wordpress]> |
```

```
MySQL [wordpress]> CREATE USER 'wordpress' IDENTIFIED BY 'pCAD3sLSjiRRRL9i';
Query OK, 0 rows affected (0.01 sec)

MySQL [wordpress]> GRANT ALL PRIVILEGES ON wordpress.* TO wordpress;
Query OK, 0 rows affected (0.00 sec)

MySQL [wordpress]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

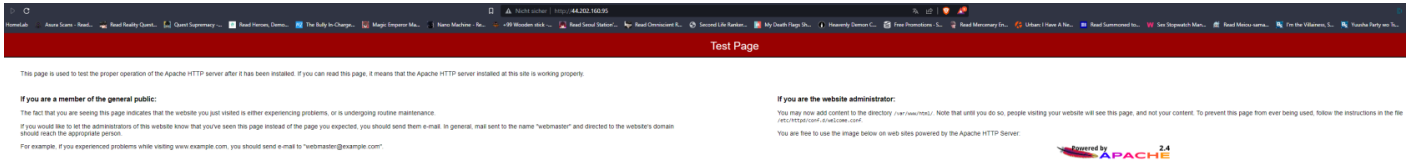
MySQL [wordpress]> Exit
Bye
[ec2-user@ip-172-31-94-181 ~]$ |
```

Konfigurieren von WordPress auf EC2

Webservice Installieren

Hier installiert man den Webserver von Apache.

```
sudo yum install -y httpd  
sudo service httpd start
```



WordPress herunterladen und Installieren

Man lädt das neuste WordPress .tar.gz herunter und entpackt es.

```
wget https://wordpress.org/latest.tar.gz  
tar -xzf latest.tar.gz
```

```
[ec2-user@ip-172-31-94-181 ~]$ ls  
latest.tar.gz  wordpress
```

Nachher geht man in das Directory wordpress und kopiert die Sample config zur wirklichen WordPress config.

In dem wp-config.php File ändert man die Database Settings zu den eigenen

```
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'wordpress' );

/** Database password */
define( 'DB_PASSWORD', 'pCAD3sLSjiRRRL9i|' );

/** Database hostname */
define( 'DB_HOST', 'wordpress.csj4fgsm0wsp.us-east-1.rds.amazonaws.com' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

Hier fügt man noch die API secret-keys ein damit man sicherer ist. Diese können in [diesem Link](#) generiert werden.

```
define('AUTH_KEY', 'RNp|Wj}fT6[-B0]p2m>*L D5pP4DAK+mvgKm&▷sB+APE5uLiW$DOH#Q[Jdhi-P(}');
define('SECURE_AUTH_KEY', 'F)W|+eYWxcF0b8[{V2]ZiR|-+0u7ba%I QQ>gw%_WEo7n1TIrsiVb[I/DPnYNV2-');
define('LOGGED_IN_KEY', 'FZ 7bTR,7J~G;B{0d#hBe6^DDjM5^Ugpi||+T^Dum_Us(0e+LsB#:/+|[+uN)#[');
define('NONCE_KEY', 'Wa:+{yt!gZg>|Weu1zSy3xe;|^3/Yh4:t-P|6z9ETMD~Sa~I(b@d)-Cz=}p^{u>');
define('AUTH_SALT', 's73Q(:9+v:|Ie,+u}Mo^{acd($U0++`#SrXv zsQ6c+@IMrHw+0*pAJ*B]W^lSah');
define('SECURE_AUTH_SALT', 'Jdoe2@51h~8u^@fcfzc[6Ua!|2Y+9IW-tXL5^FH%QLk}|6!yJ&$ZAQG&_KaotX|F');
define('LOGGED_IN_SALT', 'VoTB[Q5l +Z0db4-lSffRXZK/ET>W8K,`(_Vb|e[#1W{yI(It DI!UcL7k[ d:I'L');
define('NONCE_SALT', 'y`D;`@=LYtHof;3t)npTsLd+RLsTbh$}A$4bP4,?0Aj&Pkh(cg[Mm}P6C1c+!-1!');
```

```
sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
cd /home/ec2-user
sudo cp -r wordpress/* /var/www/html/
sudo service httpd restart
```

Und wenn man die letzten Konfigurationen, die man oben sieht, ausführt. Sollte am Schluss im Browser die WordPress Seite zu sehen sein.

Nicht sicher | http://44.202.160.95/wp-admin/install.php

WordPress

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title

Username

Username can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password [Show](#)

Important: You will need this password to log in. Please store it in a secure location.

Your Email

Double-check your email address before continuing.

Search engine visibility ☐ Discourage search engines from indexing this site

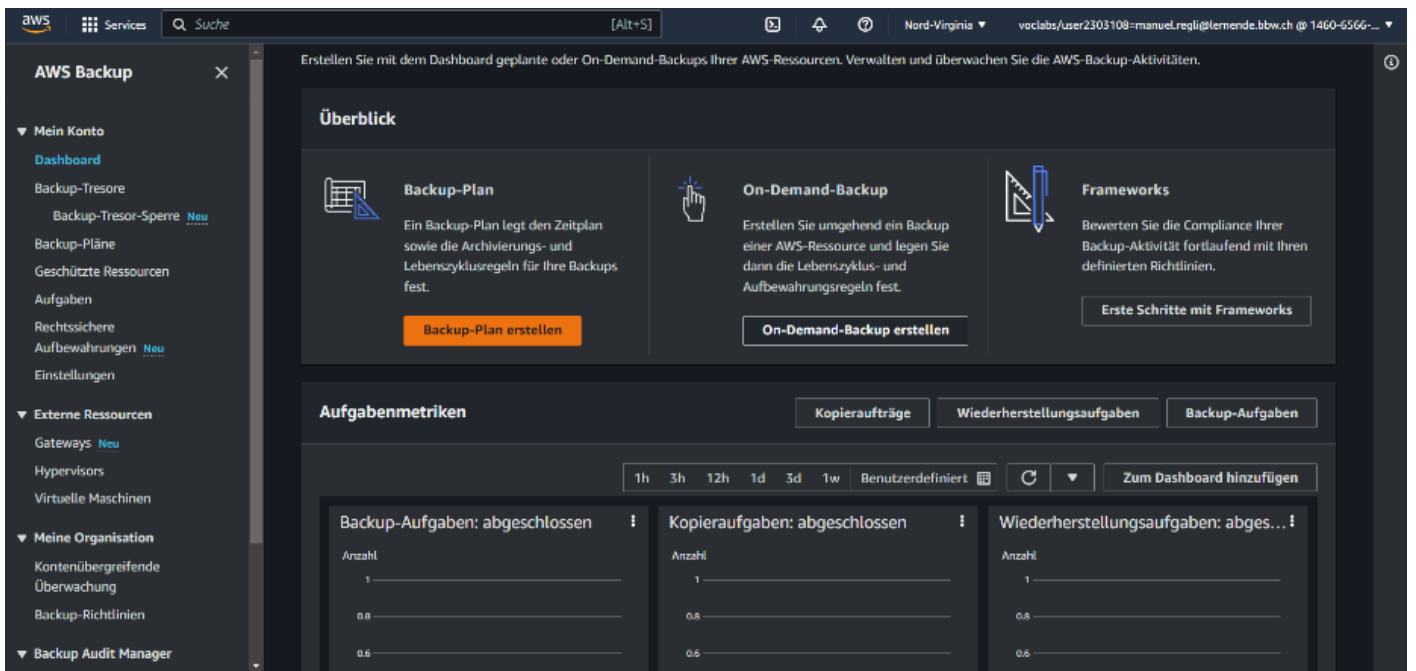
It is up to search engines to honor this request.

[Install WordPress](#)

Backup

Amazon EC2 Backup and Restore using AWS Backup

Gehe auf den AWS Backup Service und erstelle einen On-Demand-Backup.



Auf den Ressourcentyp wählt man entweder EC2 und RDS und dann die Instance-ID des RDS (WordPress) und EC2 (WordPress). Die Aufbewahrungszeitraum soll 7 Tage betragen und der Backup-Tresor ist WordPress (kann auch Default sein). Leider geht diese Methode nicht da wir nicht die nötigen Berechtigungen haben.

Einstellungen

Ressourcentyp

EC2 ▼

Instance-ID

i-0da3be2c716eb24e4 ▼



Backup-Fenster

☒ Jetzt Backup erstellen

Startet innerhalb von 1 Stunde.

☐ Backup-Fenster anpassen

Aufbewahrungszeitraum [Info](#)

Tage ▼

7

Backup-Tresor [Info](#)

Geben Sie den Backup-Tresor an, in dem dieses Backup durchgeführt wird.

WordPress ▼

Neuen Backup-Tresor erstellen

IAM-Rolle [Info](#)

Geben Sie die IAM-Rolle an, die AWS Backup beim Erstellen und Verwalten von Backups in Ihrem Namen übernimmt.

☒ Standard-Rolle

Wenn die AWS Backup-Standardrolle nicht vorhanden ist, wird eine Rolle mit den richtigen Berechtigungen für Sie erstellt.

☐ Wählen Sie eine IAM-Rolle

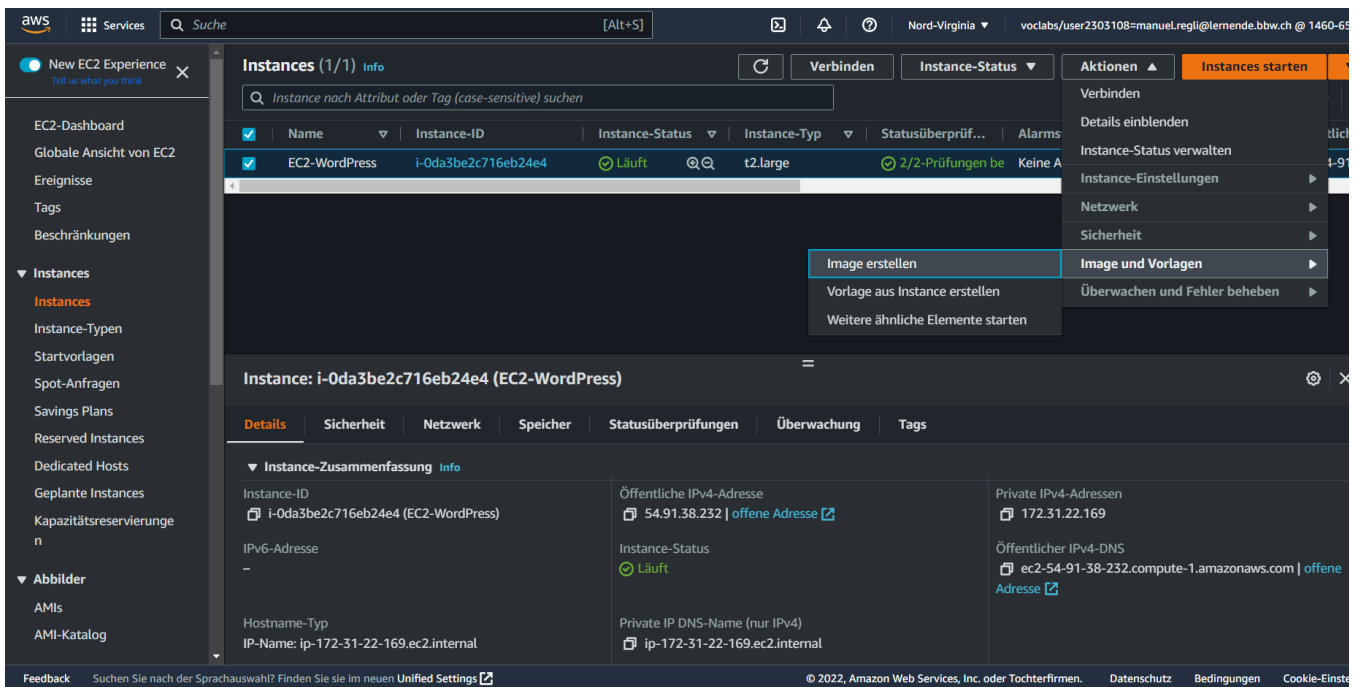
► Tags den Wiederherstellungspunkten hinzugefügt - *optional*

AWS Backup kopiert Tags bei der Erstellung von der geschützten Ressource zum Wiederherstellungspunkt. Sie können zusätzliche Tags

EC2 Snapshot

Hier sind die Schritte, um ein Snapshot-Image von einer EC2-Instanz zu erstellen:

1. Öffnen Sie das AWS Management Console und melden Sie sich mit Ihrem AWS-Konto an.
2. Navigieren Sie zum Bereich "EC2" und wählen Sie die Instanz aus, von der Sie ein Snapshot-Image erstellen möchten.
3. Klicken Sie auf die Schaltfläche "Action" und wählen Sie "Image" > "Create Image" aus.



4. Geben Sie im Dialogfeld "Create Image" einen Namen und eine Beschreibung für das Snapshot-Image ein.
5. Wählen Sie "No Reboot" aus, wenn Sie möchten, dass die Instanz während des Erstellens des Snapshot-Images läuft. Wählen Sie "Reboot" aus, wenn Sie möchten, dass die Instanz neu gestartet wird, bevor das Snapshot-Image erstellt wird.
6. Klicken Sie auf die Schaltfläche "Create Image".

Image-Name

EC2-WordPress

Maximal 127 Zeichen. Kann nach der Erstellung nicht mehr geändert werden.

Image-Beschreibung – optional

Ein Backup des EC2-WordPress

Maximal 255 Zeichen

Kein Neustart

☐ Aktivieren

Instance-Volumes

Volume-Typ	Gerät	Snapshot	Größe	Volume-Typ	IOPS	Durchsatz	Bei Beenden löschen	Verschlüsselt
EBS	/dev/...	Neuen Snapshot vom V...	8	EBS-Allzweck-SSD - gp2	100		<input checked="" type="checkbox"/> Aktivieren	<input type="checkbox"/> Aktivieren

[Volume hinzufügen](#)

Während des Abbilderstellungsprozesses erstellt Amazon EC2 einen Snapshot von jedem der oben genannten Volumes.

Tags – optional

Ein Tag ist eine Markierung, die Sie einer AWS-Ressource zuordnen. Jedes Tag besteht aus einem Schlüssel und einem optionalen Wert. Sie können Tags verwenden, um Ressourcen zu suchen und zu filtern oder um Ihre AWS-Kosten nachzuverfolgen.

☒ Bild und Snapshots zusammen kennzeichnen
Kennzeichnen Sie das Bild und die Snapshots mit demselben Tag.

☐ Bild und Snapshots separat kennzeichnen
Kennzeichnen Sie das Bild und die Snapshots mit verschiedenen Tags.

7. Das Snapshot-Image wird im Hintergrund erstellt und kann in der Liste der AMIs (Amazon Machine Images) im AWS Management Console angezeigt werden, wenn es abgeschlossen ist.

The screenshot shows the AWS Management Console interface for Amazon Machine Images (AMIs). The left sidebar contains navigation links for various AWS services. The main content area displays the details for a specific AMI with the ID 'ami-0762924aa633caad1'.

Amazon Machine Images (AMIs) (1/1) Info			
Name	AMI-ID	Name des AMI	Quelle
-	ami-0762924aa633caad1	EC2-WordPress	146065668584/EC2-WordPress

AMI-ID: ami-0762924aa633caad1			
Details	Berechtigungen	Speicher	Tags
AMI-ID ami-0762924aa633caad1	Image-Typ machine	Plattformdetails Linux/UNIX	Stammgerätetyp EBS
Name des AMI EC2-WordPress	Konto-ID des Besitzers 146065668584	Architektur x86_64	Nutzungsvorgang RunInstances
Stammgerätename /dev/xvda	Status Verfügbar	Quelle 146065668584/EC2-WordPress	Virtualisierungstyp hvm
Startmodus -	Zustandsgrund -	Erstellungsdatum Mon Dec 19 2022 09:43:38 GMT+0100	Kernel-ID -

Es ist wichtig zu beachten, dass das Snapshot-Image nur die Inhalte der EBS-Volumes sichert, die an die EC2-Instanz angefügt sind. Wenn Sie andere Daten auf der Instanz haben, die nicht in einem EBS-Volume gespeichert sind, müssen Sie diese Daten separat sichern. Sie können zum Beispiel ein Backup von der EC2-Instanz auf S3 speichern, um zusätzliche Sicherheit zu gewährleisten.

Von Snapshot Volume Wiederherstellen:

Snapshot-ID

📄 snap-0cf02af311536f614

Volume-Typ Info

Allzweck-SSD (gp2)

Größe (GiB)

8

Min.: 1 GiB, Max.: 16384 GiB. Der Wert muss eine Ganzzahl sein.

IOPS

100 / 3000


Durchsatz (MiB/s) Info

Nicht zutreffend

Availability Zone Info

us-east-1a

Schnelle Snapshot-Wiederherstellung Info

 Für ausgewählten Snapshot nicht aktiviert

Verschlüsselung Info

☐ Dieses Volume verschlüsseln

Volume erstellen

aws

Services

Suche

[Alt+S]

Nord-Virginia

voclabs/user2303108=manuel.regli@lernende.bb.w.ch @ 1460-6566-...

Startvorlagen

Spot-Anfragen

Savings Plans

Reserved Instances

Dedicated Hosts

Geplante Instances

Kapazitätsreservierung

Abbilder

AMIs

AMI-Katalog

Elastic Block Store

Sie können jetzt Richtlinien für Amazon Data Lifecycle Manager erstellen, um die Snapshot-Verwaltung direkt von diesem Bildschirm zu automatisieren. Wählen Sie zunächst die zu sichernden Volumes aus, dann wählen Sie **Actions**, **Create snapshot lifecycle policy** (Erstellen einer Lebenszyklusrichtlinie erstellen) aus. Weitere Informationen finden Sie im entsprechenden [Knowledge-Center-Artikel](#).

Volumes (2)

Suchen

Aktionen

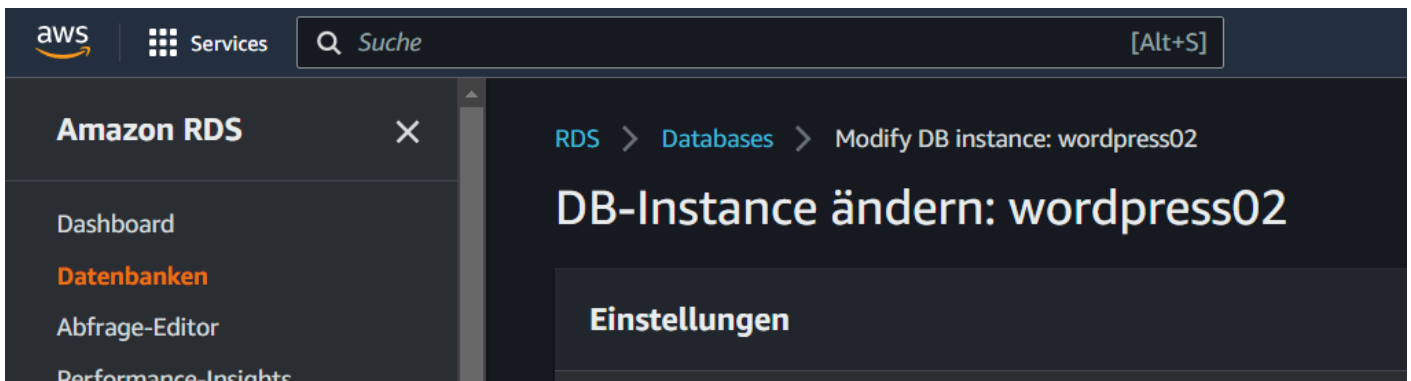
Volume erstellen

<

1

>

<input type="checkbox"/>	Name	Volume-ID	Typ	Größe	IOPS	Durchsatz	Snapshot	Erstellt	Availability Zone
<input type="checkbox"/>	-	vol-02d69f20f55936f29	gp2	8 GiB	100	-	snap-097c82c...	2022/12/19 09:24 GMT+1	us-east-1
<input type="checkbox"/>	-	vol-0b661356b322bc022	gp2	8 GiB	100	-	snap-0cf02af3...	2023/01/09 09:28 GMT+1	us-east-1



RDS Backup

RDS Backup Automatisch

Um die automatischen Backups für Ihre RDS-Instanz zu nutzen, müssen Sie diese zunächst einrichten. Hier sind die Schritte, wie Sie das tun können:

1. Öffnen Sie die Amazon RDS-Konsole und wählen Sie die Instanz aus, für die Sie Backups einrichten möchten.

Datensicherung

☒ Automatische Backups aktivieren
Erstellt einen Point-in-Time-Snapshot Ihrer Datenbank

⚠ Bitte beachten Sie, dass automatisierte Backups derzeit nur für die InnoDB-Speicher-Engine unterstützt werden. Nähere Informationen zu MyISAM erhalten Sie [hier](#).

Aufbewahrungszeitraum für Backups [Informationen](#)

Die Anzahl der Tage (1-35), für die automatische Sicherungen aufbewahrt werden.

7

▼

Tage

Backup-Fenster [Informationen](#)

Der tägliche Zeitraum (in UTC), in dem RDS automatische Backups erstellt.

☒ Fenster auswählen

☐ Keine Präferenz

Startzeit

07

▼

 :

14

▼

 UTC

Dauer

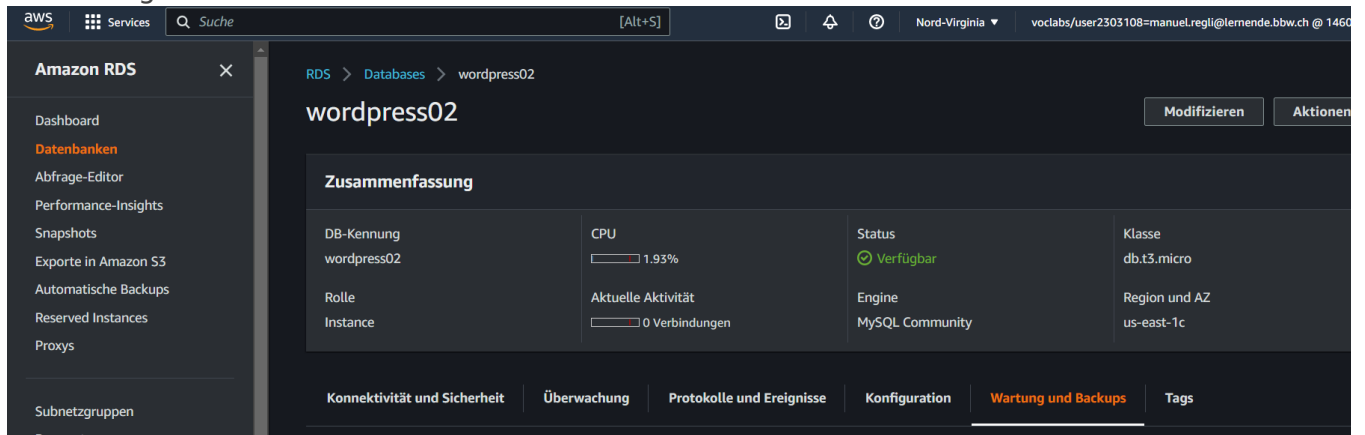
0.5

▼

 Stunden

2. Klicken Sie auf die Registerkarte "Backups" und dann auf "Edit Backup Retention Period".
3. Geben Sie die Anzahl der Tage an, für die Sie Backups aufbewahren möchten, und klicken Sie dann auf "Save Changes". Standardmäßig werden Backups für eine Retention-Zeit von

einem Tag erstellt.

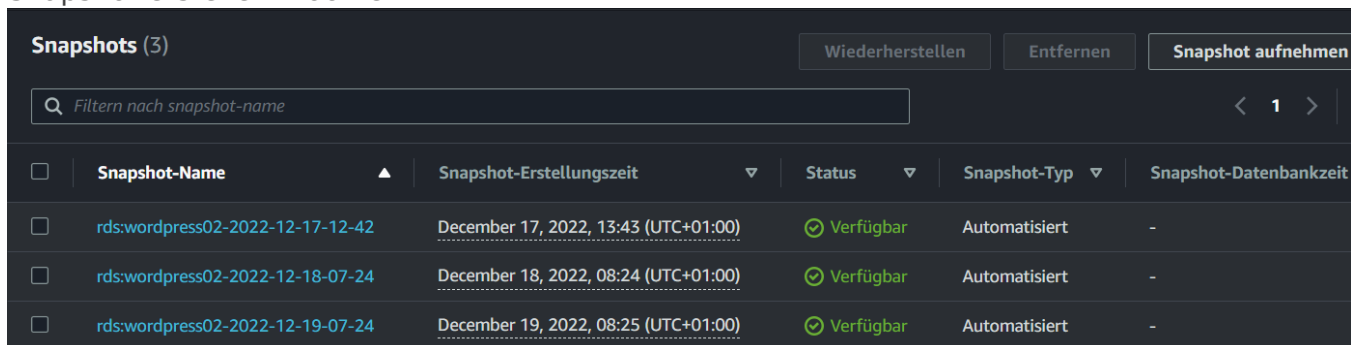


Wichtig: Wenn Sie die Retention-Zeit auf 0 Tage ändern, werden automatische Backups deaktiviert. Sobald die automatischen Backups eingerichtet sind, werden täglich Backups Ihrer RDS-Instanz erstellt und für die angegebene Anzahl von Tagen aufbewahrt. Sie können diese Backups in der RDS-Konsole unter der Registerkarte "Backups" anzeigen und verwalten.

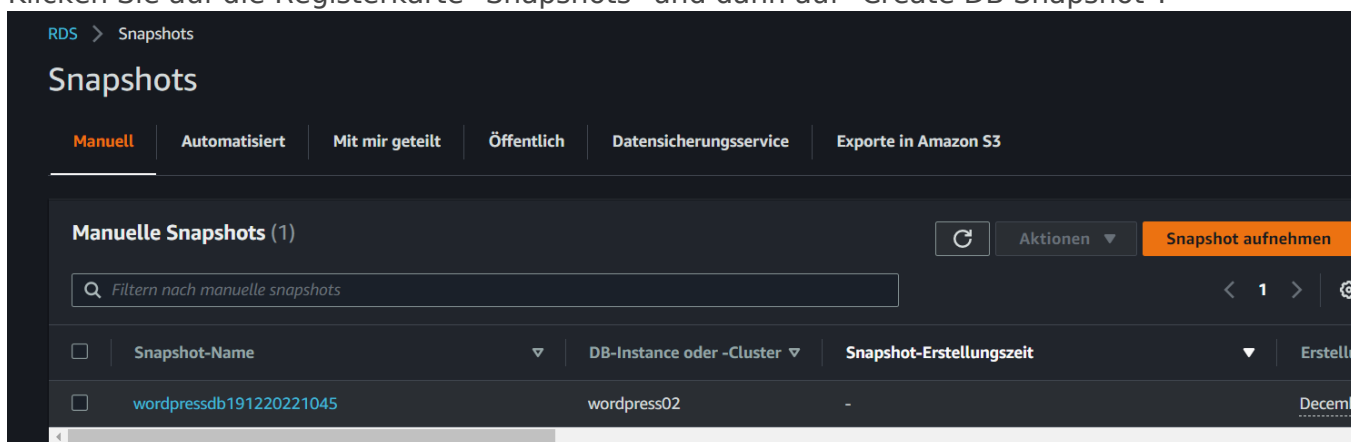
RDS Backup Manuell

Um manuelle Snapshots von Ihrer RDS-Instanz zu erstellen, folgen Sie diesen Schritten:

1. Öffnen Sie die Amazon RDS-Konsole und wählen Sie die Instanz aus, für die Sie einen Snapshot erstellen möchten.



2. Klicken Sie auf die Registerkarte "Snapshots" und dann auf "Create DB Snapshot".



3. Geben Sie einen Namen für den Snapshot an und klicken Sie auf "Create DB Snapshot".

Der Snapshot-Prozess kann einige Zeit in Anspruch nehmen, abhängig von der Größe Ihrer Datenbank. Während des Prozesses wird Ihre RDS-Instanz möglicherweise kurzzeitig nicht

verfügbar sein. Sobald der Snapshot erstellt wurde, wird er in der Liste der Snapshots angezeigt, die Sie in der RDS-Konsole unter der Registerkarte "Snapshots" finden. Sie können auf den Namen des Snapshots klicken, um weitere Informationen anzuzeigen und ihn zu verwalten. Hinweis: Manuell erstellte Snapshots bleiben solange erhalten, bis Sie sie manuell löschen. Es ist daher wichtig, dass Sie regelmäßig überflüssige Snapshots löschen, um Speicherplatz zu sparen.

Von Backup Datenbank erstellen:

[RDS](#) > [Databases](#) > wordpress02

wordpress02

ModifizierenAktionen ▼

Zusammenfassung

DB-Kennung wordpress02	CPU <div>1.78%</div>	Status ✔ Verfügbar	Klasse db.t3.micro
Rolle Instance	Aktuelle Aktivität <div>0 Verbindungen</div>	Engine MySQL Community	Region und AZ us-east-1c

Konnektivität und SicherheitÜberwachungProtokolle und EreignisseKonfigurationWartung und BackupsTags

Snapshots (10)

WiederherstellenEntfernenSnapshot aufnehmen

Filtern nach snapshot-name

< 1 2 > ⚙

<input type="checkbox"/>	Snapshot-Name ▲	Snapshot-Erstellungszeit ▼	Status ▼	Snapshot-Typ ▼	Snapshot-Datenbankzeit ▼
<input type="checkbox"/>	rd:wordpress02-2023-01-01-07-25	January 01, 2023, 08:25 (UTC+01:00)	✔ Verfügbar	Automatisiert	-
<input type="checkbox"/>	rd:wordpress02-2023-01-02-07-25	January 02, 2023, 08:25 (UTC+01:00)	✔ Verfügbar	Automatisiert	-
<input checked="" type="checkbox"/>	rd:wordpress02-2023-01-03-07-25	January 03, 2023, 08:25 (UTC+01:00)	✔ Verfügbar	Automatisiert	-
<input type="checkbox"/>	rd:wordpress02-2023-01-04-07-25	January 04, 2023, 08:25 (UTC+01:00)	✔ Verfügbar	Automatisiert	-
<input type="checkbox"/>	rd:wordpress02-2023-01-05-07-25	January 05, 2023, 08:26 (UTC+01:00)	✔ Verfügbar	Automatisiert	-

Wenn man Wiederherstellen drückt kann man eine Neue DB erstellen mit den Informationen

Datenbanken

Gruppenressourcen ↗

ModifizierenAktionen ▼Wiederherstellung aus S3Datenbank erstellen

Filtern nach datenbanken

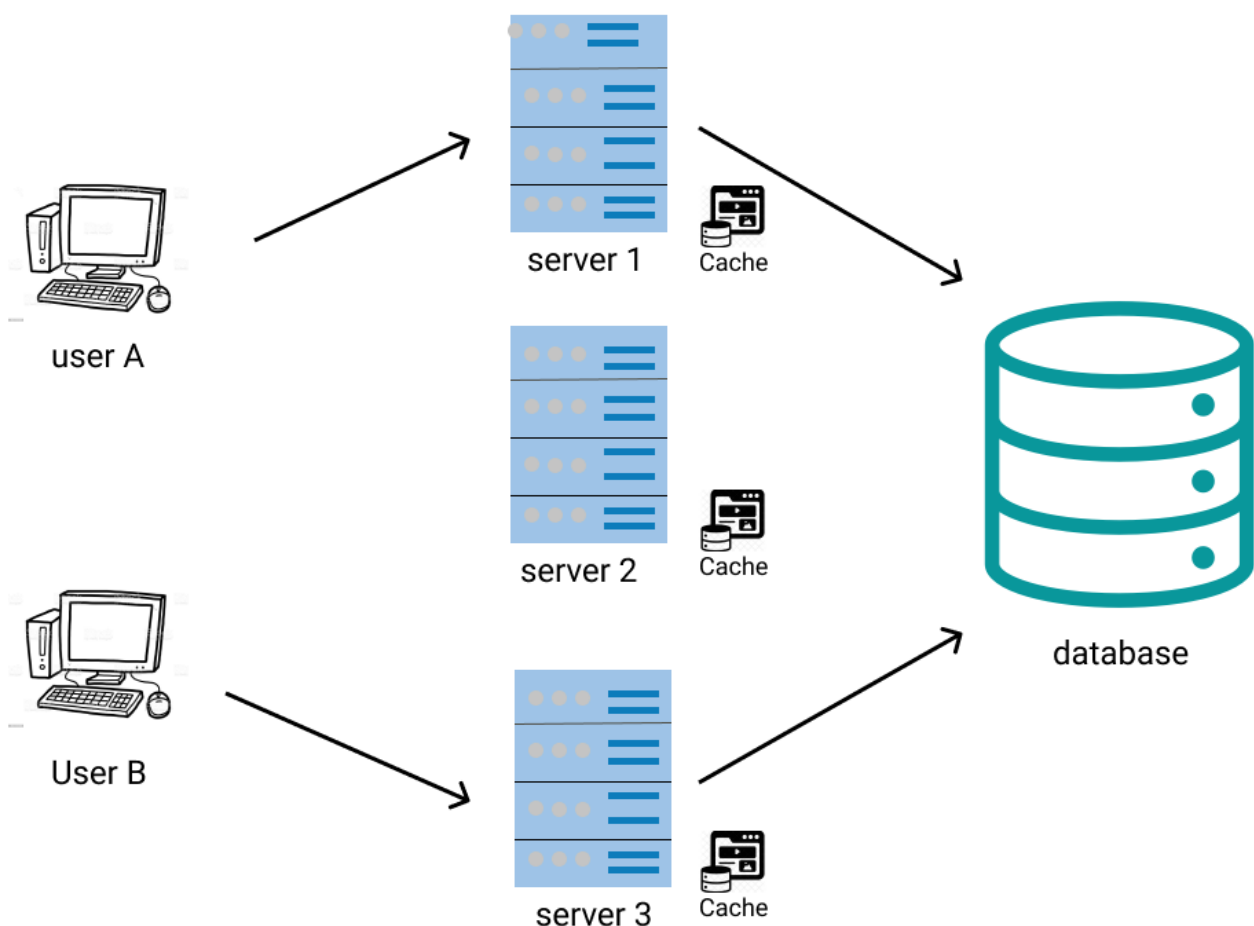
< 1 > ⚙

<input type="checkbox"/>	DB-Kennung ▲	Rolle ▼	Engine ▼	Region und AZ ▼	Größe ▼	Status ▼	CPU
<input type="radio"/>	wordpress-backup	Instance	MySQL Community	-	db.m6i.large	🕒 Wird erstellt	-
<input type="radio"/>	wordpress02	Instance	MySQL Community	us-east-1c	db.t3.micro	✔ Verfügbar	<div></div>

Was Bedeutet Stateless Architecture:

Eine stateless architecture bedeutet, dass keine Informationen über den Zustand einer Anwendung auf dem Server gespeichert werden. Dies bedeutet, dass jeder Request, den ein Benutzer an die Anwendung sendet, alle notwendigen Informationen enthält, um die Anforderung zu verarbeiten. Der Server muss keine Informationen aus einem vorherigen Request speichern, um den aktuellen Request zu verarbeiten.

Ein Beispiel für eine stateless architecture wäre eine Wordpress-Installation, bei der alle Benutzerinformationen in der Datenbank gespeichert werden und jeder Request, den ein Benutzer an die Anwendung sendet, die erforderlichen Informationen von der Datenbank abrufen, um die Anforderung zu verarbeiten. Die Anwendung muss keine Informationen über den Zustand des Benutzers auf dem Server speichern, um Anforderungen von diesem Benutzer zu verarbeiten.



Bash Script

Da mir diese Schritte aber zu lang gingen habe ich noch ein kurzes Bash Script geschrieben.
(Zuunterst sieht man noch einen Link zum Test Video)

```
#!/bin/bash

sudo yum -y update
sudo yum install -y mysql
sudo yum install -y httpd
sudo service httpd start
wget https://wordpress.org/latest.tar.gz
tar -xzf latest.tar.gz
cp wordpress/wp-config-sample.php wordpress/wp-config.php

# Set the new values for the database variables
new_db_name='wordpress'
new_db_user='wordpress'
new_db_password='5ztDByGKq49aKm3S'
new_db_host='wordpressdb.csj4fgsm0wsp.us-east-1.rds.amazonaws.com'

# Replace the old values in the wp-config.php file
sed -i "s/define( 'DB_NAME', 'database_name_here' );/define( 'DB_NAME', '$new_db_name' );/" wordpress/wp-config.php
sed -i "s/define( 'DB_USER', 'username_here' );/define( 'DB_USER', '$new_db_user' );/" wordpress/wp-config.php
sed -i "s/define( 'DB_PASSWORD', 'password_here' );/define( 'DB_PASSWORD', '$new_db_password' );/" wordpress/wp-config.php
sed -i "s/define( 'DB_HOST', 'localhost' );/define( 'DB_HOST', '$new_db_host' );/" wordpress/wp-config.php

# Ersetze die Zeilen mit den 'unique phrase' durch die neuen Werte
sed -i -s 's{define( \"AUTH_KEY\", \"put your unique phrase here\" );}{define(\"AUTH_KEY\", \"0ozpHN_tE]U?[L.)kvnCDrC-H*qI:UgzQvV9Ga+)k+*~RIZZZHD,kB>[<Y6|RjXC\" );{' wordpress/wp-config.php
sed -i -s 's{define( \"SECURE_AUTH_KEY\", \"put your unique phrase here\" );?define(\"SECURE_AUTH_KEY\", \"{G!dia+f!={U>8vI%Ch0Y-{/xO~Eew^q_ETOyG]~EUeK;C2v3JRx8)N[vi[tO}<\" );?' wordpress/wp-config.php
sed -i -s 's{define( \"LOGGED_IN_KEY\", \"put your unique phrase here\" );}{define(\"LOGGED_IN_KEY\", \"!*^!0Q?teip2%gg8W=ssNTeB,M?7/++U7bj`GHN9/-i#Y<{n`hK5INO[gOB~|#y\" );}' wordpress/wp-config.php
sed -i -s 's{define( \"NONCE_KEY\", \"put your unique phrase here\" );!define(\"NONCE_KEY\", \"W*L:-+4kkn4{hX_8TH`P3Z0yq1==vy)4M~OBRkj*e}u:1ttc_]Kb8HmMV+|yO>1 \" );!' wordpress/wp-config.php
sed -i -s 's{define( \"AUTH_SALT\", \"put your unique phrase here\" );}{define(\"AUTH_SALT\",
```

```

\"*vb,woc2_=4|[5$di(?Ata;{vQ6,!nf3OT.#( #<poS7AdkX_BU1Px~~8nY!y[1x\" );}' wordpress/wp-config.php
sed -i -s 's{define( '\"SECURE_AUTH_SALT\"\", '\"put your unique phrase here\"
);}define('\"SECURE_AUTH_SALT\",
\"*o:<yV#gpxt/cS#HrsHeX?[A*;hm+!`oufT}WZWBum1Bn::0hM6jyNZScN<tC)! \" );{' wordpress/wp-config.php
sed -i -s 's}define( '\"LOGGED_IN_SALT\"\", '\"put your unique phrase here\" ');}define('\"LOGGED_IN_SALT\",
 '\"DY+PZ8;3?*~/~GnaQk)T<=47o|)x#Ht(o,^Zs3K,{*UtX2CBB[la;1Y!0:h]Q^#C\" ');}' wordpress/wp-config.php
sed -i -s 's}define( '\"NONCE_SALT\"\", '\"put your unique phrase here\" ');}define('\"NONCE_SALT\",
 '\"abD~m;/Ws0NJI]EodiE[{a,x|DU/W%w/aK15j Ee?]> 7L-i$u[?C,5=qt<q{8\" ');}' wordpress/wp-config.php

```

```
sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
```

```
sudo cp -r wordpress/* /var/www/html/
```

```
sudo service httpd restart
```

Video: <https://youtu.be/b2QCiFxmGvw>

Ansible Playbook

Da mir diese Schritte aber zu lang gingen habe ich noch ein kurzes Ansible Playbook geschrieben. (Zuunterst sieht man noch einen Link zum Test Video)

```

---
- name: Install and configure WordPress
  hosts: all
  become: true
  vars:
    new_db_name: wordpress
    new_db_user: wordpress
    new_db_password: 5ztDByGKq49aKm3S
    new_db_host: wordpress02.csj4fgsm0wsp.us-east-1.rds.amazonaws.com

  tasks:
    - name: Update package manager cache and install MySQL, Apache, and PHP
      package:
        name:
          - mysql
          - httpd
          - amazon-linux-extras

```

```
register: pkg_result
```

service:

```
state: started
```

```
get_url:
```

```
dest: /tmp/latest.tar.gz
```

unarchive:

```
dest: /tmp
```

```
remote_src: yes
```

```
shell: cp /tmp/wordpress/wp-config-sample.php /tmp/wordpress/wp-config.php
```

replace:

```
regexp: "define\\((\\s*'DB_NAME',\\s*\\.?'\\s*\\)"
```

```
replace: "define( 'DB_NAME', '{ { new_db_name } }' );"
```

replace:

```
regexp: "define\\((\\s*'DB_USER',\\s*\\.?'\\s*\\)"
```

```
replace: "define( 'DB_USER', '{{ new_db_user }}' );"
```

replace:

```
regexp: "define\\((\\s*'DB PASSWORD',\\s*'.*'\\s*\\)"
```

```
replace: "define( 'DB_PASSWORD', '{{ new_db_password }}' );"
```

```

replace:
  path: /tmp/wordpress/wp-config.php
  regexp: "define\\(\\s*'DB_HOST',\\s*.*'\\s*\\)"
  replace: "define( 'DB_HOST', '{ new_db_host }' );"

- name: Set unique phrases in wp-config.php
  replace:
    path: /tmp/wordpress/wp-config.php
    regexp: "define\\( 'AUTH_KEY',\\s+'put your unique phrase here' \\);"
    replace: "define('AUTH_KEY', '0ozpHN_tE]U?[L.)kvnCDrC-
H*ql:UgzQvV9Ga+)k+*~RIZZZHD,kB>[<Y6|RjXC');"
  replace:
    path: /tmp/wordpress/wp-config.php
    regexp: "define\\( 'SECURE_AUTH_KEY',\\s+'put your unique phrase here' \\);"
    replace: "define('SECURE_AUTH_KEY', '{G!dia+f!={U>8vI%Ch0Y-{/xO~Eew^q_ETOyG]~EUeK;C2v3JRx8)N
[vi[tO}<');;"
  replace:
    path: /tmp/wordpress/wp-config.php
    regexp: "define\\( 'LOGGED_IN_KEY',\\s+'put your unique phrase here' \\);"
    replace: "define('LOGGED_IN_KEY', '!*^!0Q?teip2%gg8W=ssNTeB,M?7/+ ++U7bj`GHN9/-
i#Y<{n`hK5!NO[gOB~|#y');;"
  replace:
    path: /tmp/wordpress/wp-config.php
    regexp: "define\\( 'NONCE_KEY',\\s+'put your unique phrase here' \\);"
    replace: "define('NONCE_KEY', 'W*L:-
+4kkn4{hX_8TH`P3Z0yq1==vy)4M~OBRk]*e}u:1ttc_}Kb8HmMV+|yO>1 ');;"
  replace:
    path: /tmp/wordpress/wp-config.php
    regexp: "define\\( 'AUTH_SALT',\\s+'put your unique phrase here' \\);"
    replace: "define('AUTH_SALT', '*vb,woc2_=4|[5$di(?Ata;{vQ6,!nf3OT.#(
#<poS7AdkX_BU1Px~~8nY!y[1x');"
  replace:
    path: /tmp/wordpress/wp-config.php
    regexp: "define\\( 'SECURE_AUTH_SALT',\\s+'put your unique phrase here' \\);"
    replace: "define('SECURE_AUTH_SALT',
'*o:<yV#gpxt/cS#HrsHeX?[A*;hm+!`oufT}WZW Bum1Bn.;0hM6jyNZScN<tC)! ');;"
  replace:
    path: /tmp/wordpress/wp-config.php
    regexp: "define\\( 'LOGGED_IN_SALT',\\s+'put your unique phrase here' \\);"

```

```
replace: "define('LOGGED_IN_SALT',
'DY+PZ8;3?*~/~GnaQk)T<=47o|)x#Ht(o,^Zs3K,{*UtX2CBB[la;1Y!0:h]Q^#C');"
replace:
path: /tmp/wordpress/wp-config.php
regexp: "define\\( 'NONCE_SALT',\\s+'put your unique phrase here' \\);"
replace: "define('NONCE_SALT', 'abD~m;/Ws0Nj]l]EodiE[{a,x|DU/W%w/aK15j Ee?]> 7L-i$u[?C,5=qt<q{8'});"

- name: Install lamp-mariadb10.2-php7.2
command: "sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2"

- name: copy files task
shell: sudo cp -r /tmp/wordpress/* /var/www/html/

- name: Restart Apache service
service:
  name: httpd
  state: restarted
```

Video: <https://youtu.be/Nb9mF5vab7M>

Revision #1

Created 15 December 2023 12:39:32 by Manuel Regli

Updated 15 December 2023 12:50:27 by Manuel Regli