

Elastic Beanstalk

Gregor Bückendorf

3. November 2017

Was ist Elastic Beanstalk?

Was ist Elastic Beanstalk?

- Ein Weg für Anwendungsentwickler Webapps hochverfügbar in die Amazon Cloud zu bringen, ohne sich um Resources kümmern zu müssen

Was ist Elastic Beanstalk?

- Ein Weg für Anwendungsentwickler Webapps hochverfügbar in die Amazon Cloud zu bringen, ohne sich um Resources kümmern zu müssen
- Ein Werkzeug zum verwalten von Deployment Environments in der Amazon Cloud

Was ist Elastic Beanstalk?

- Ein Weg für Anwendungsentwickler Webapps hochverfügbar in die Amazon Cloud zu bringen, ohne sich um Resources kümmern zu müssen
- Ein Werkzeug zum verwalten von Deployment Environments in der Amazon Cloud
- 4 AWS Resource Types

Die Application

AWS::ElasticBeanstalk::Application



Die Application

AWS::ElasticBeanstalk::Application

- Ein Container für Application Versions



Die Application

AWS::ElasticBeanstalk::Application

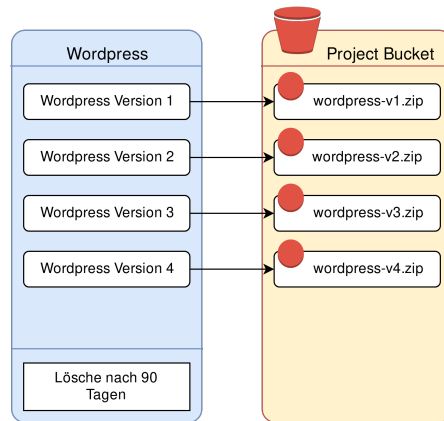
- Ein Container für Application Versions
- Regeln zur automatischen Löschung alter Versionen



Die Version

AWS::ElasticBeanstalk::ApplicationVersion

Ein Link zu einem Objekt in einem S3-Bucket

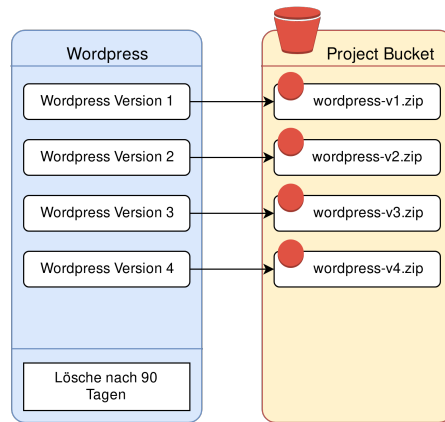


Die Version

AWS::ElasticBeanstalk::ApplicationVersion

Ein Link zu einem Objekt in einem S3-Bucket

- wird hier Sourcebundle genannt

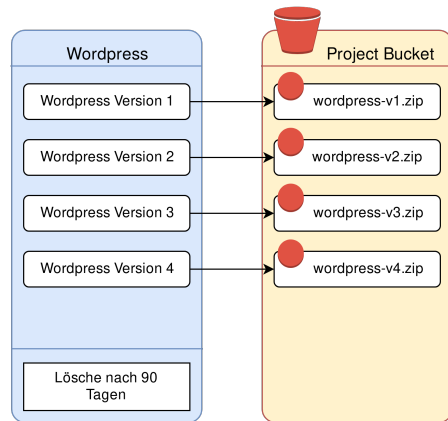


Die Version

AWS::ElasticBeanstalk::ApplicationVersion

Ein Link zu einem Objekt in einem S3-Bucket

- wird hier Sourcebundle genannt
- ist meistens eine .zip-Datei

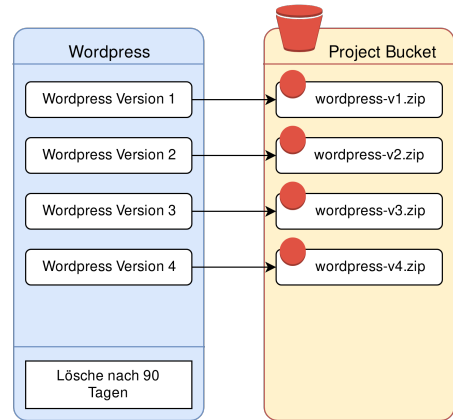


Die Version

AWS::ElasticBeanstalk::ApplicationVersion

Ein Link zu einem Objekt in einem S3-Bucket

- wird hier Sourcebundle genannt
- ist meistens eine .zip-Datei
- enthält die eigentliche Applikation

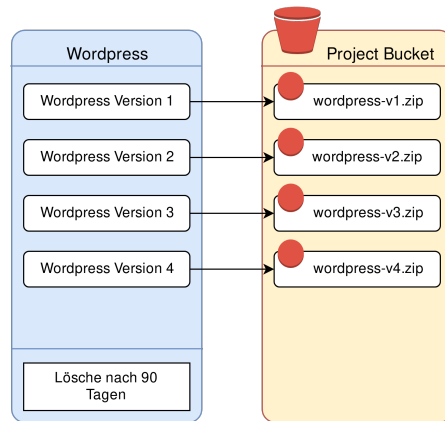


Die Version

AWS::ElasticBeanstalk::ApplicationVersion

Ein Link zu einem Objekt in einem S3-Bucket

- wird hier Sourcebundle genannt
- ist meistens eine .zip-Datei
- enthält die eigentliche Applikation (z.B. den Inhalt von /var/www/html)

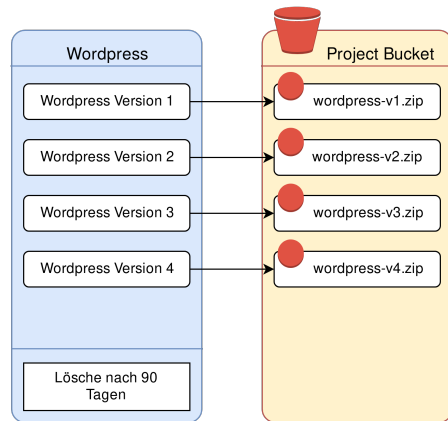


Die Version

AWS::ElasticBeanstalk::ApplicationVersion

Ein Link zu einem Objekt in einem S3-Bucket

- wird hier Sourcebundle genannt
- ist meistens eine .zip-Datei
- enthält die eigentliche Applikation (z.B. den Inhalt von /var/www/html)
- kann auch Konfigurationen enthalten



Die Environment

AWS::ElasticBeanstalk::Environment

Die Environment

AWS::ElasticBeanstalk::Environment

- Zentrale Resource des Deployments

Die Environment

AWS::ElasticBeanstalk::Environment

- Zentrale Resource des Deployments
- Erstellt und verwaltet einen Cloudformation Stack

Die Environment

AWS::ElasticBeanstalk::Environment

- Zentrale Resource des Deployments
- Erstellt und verwaltet einen Cloudformation Stack
- Verbindet:
 - Application Version

Die Environment

AWS::ElasticBeanstalk::Environment

- Zentrale Resource des Deployments
- Erstellt und verwaltet einen Cloudformation Stack
- Verbindet:
 - Application Version
 - Environment Tier
 - Bestimmt aus welchen Ressourcen der Cloudformation Stack besteht
 - z.Z. WebServer oder Worker

Die Environment

AWS::ElasticBeanstalk::Environment

- Zentrale Resource des Deployments
- Erstellt und verwaltet einen Cloudformation Stack
- Verbindet:
 - Application Version
 - Environment Tier
 - Bestimmt aus welchen Ressourcen der Cloudformation Stack besteht
 - z.Z. WebServer oder Worker
 - Platform
 - Spezielles AMI, das auf den Instanzen der zentralen Autoscaling Group der Environment läuft
 - Wird auch Solution Stack genannt
 - z.B. PHP, Python, Docker, Java ...

Die Environment

AWS::ElasticBeanstalk::Environment

- Zentrale Resource des Deployments
- Erstellt und verwaltet einen Cloudformation Stack
- Verbindet:
 - Application Version
 - Environment Tier
 - Bestimmt aus welchen Ressourcen der Cloudformation Stack besteht
 - z.Z. WebServer oder Worker
 - Platform
 - Spezielles AML, das auf den Instanzen der zentralen Autoscaling Group der Environment läuft
 - Wird auch Solution Stack genannt
 - z.B. PHP, Python, Docker, Java ...
- Configuration Options

Configuration Options

- Hauptweg um die festen Bestandteile einer Environment anzupassen.

Configuration Options

- Hauptweg um die festen Bestandteile einer Environment anzupassen.
- Bestehen aus Key-Value-Pairs in Namespaces, z.B.
aws:elb:listener ListenerProtocol = HTTP

Configuration Options

- Hauptweg um die festen Bestandteile einer Environment anzupassen.
- Bestehen aus Key-Value-Pairs in Namespaces, z.B.
aws:elb:listener ListenerProtocol = HTTP
- Dinge, die sich über Configuration Options einstellen lassen:

Configuration Options

- Hauptweg um die festen Bestandteile einer Environment anzupassen.
- Bestehen aus Key-Value-Pairs in Namespaces, z.B.
aws:elb:listener ListenerProtocol = HTTP
- Dinge, die sich über Configuration Options einstellen lassen:
 - Das VPC in das Ressourcen gesetzt werden

Configuration Options

- Hauptweg um die festen Bestandteile einer Environment anzupassen.
- Bestehen aus Key-Value-Pairs in Namespaces, z.B.
aws:elb:listener ListenerProtocol = HTTP
- Dinge, die sich über Configuration Options einstellen lassen:
 - Das VPC in das Ressourcen gesetzt werden
 - Die Autoscaling Group (Größe, Trigger, Instanztypen, EC2Keys u.s.w.)

Configuration Options

- Hauptweg um die festen Bestandteile einer Environment anzupassen.
- Bestehen aus Key-Value-Pairs in Namespaces, z.B.
aws:elb:listener ListenerProtocol = HTTP
- Dinge, die sich über Configuration Options einstellen lassen:
 - Das VPC in das Ressourcen gesetzt werden
 - Die Autoscaling Group (Größe, Trigger, Instanztypen, EC2Keys u.s.w.)
 - Individuelle Variablen, die an die Anwendung weiter gegeben werden (bei PHP z.B. durch das \$_SERVER Array)

Configuration Options

- Hauptweg um die festen Bestandteile einer Environment anzupassen.
- Bestehen aus Key-Value-Pairs in Namespaces, z.B.
aws:elb:listener ListenerProtocol = HTTP
- Dinge, die sich über Configuration Options einstellen lassen:
 - Das VPC in das Ressourcen gesetzt werden
 - Die Autoscaling Group (Größe, Trigger, Instanztypen, EC2Keys u.s.w.)
 - Individuelle Variablen, die an die Anwendung weiter gegeben werden (bei PHP z.B. durch das `$_SERVER` Array)
 - Der Loadbalancer

Configuration Options

- Hauptweg um die festen Bestandteile einer Environment anzupassen.
- Bestehen aus Key-Value-Pairs in Namespaces, z.B.
aws:elb:listener ListenerProtocol = HTTP
- Dinge, die sich über Configuration Options einstellen lassen:
 - Das VPC in das Ressourcen gesetzt werden
 - Die Autoscaling Group (Größe, Trigger, Instanztypen, EC2Keys u.s.w.)
 - Individuelle Variablen, die an die Anwendung weiter gegeben werden (bei PHP z.B. durch das `$_SERVER` Array)
 - Der Loadbalancer
 - Ob die Environment überhaupt einen Loadbalancer hat, oder SingleInstance sein soll

Configuration Options

- Hauptweg um die festen Bestandteile einer Environment anzupassen.
- Bestehen aus Key-Value-Pairs in Namespaces, z.B.
aws:elb:listener ListenerProtocol = HTTP
- Dinge, die sich über Configuration Options einstellen lassen:
 - Das VPC in das Ressourcen gesetzt werden
 - Die Autoscaling Group (Größe, Trigger, Instanztypen, EC2Keys u.s.w.)
 - Individuelle Variablen, die an die Anwendung weiter gegeben werden (bei PHP z.B. durch das `$_SERVER` Array)
 - Der Loadbalancer
 - Ob die Environment überhaupt einen Loadbalancer hat, oder SingleInstance sein soll
 - ...

.ebextensions

Konfigurationen an der ApplicationVersion

Wenn im Rootverzeichnis einer ApplicationVersion ein Directory mit dem Namen `.ebextensions` existiert, werden YAML- und JSON-Dateien für die Konfiguration von Environments ausgelesen, wenn diese ApplicationVersion ausgerollt wird.

.ebextensions

Konfigurationen an der ApplicationVersion

Wenn im Rootverzeichnis einer ApplicationVersion ein Directory mit dem Namen `.ebextensions` existiert, werden YAML- und JSON-Dateien für die Konfiguration von Environments ausgelesen, wenn diese ApplicationVersion ausgerollt wird.

Erlaubte Sektionen:

.ebextensions

Konfigurationen an der ApplicationVersion

Wenn im Rootverzeichnis einer ApplicationVersion ein Directory mit dem Namen `.ebextensions` existiert, werden YAML- und JSON-Dateien für die Konfiguration von Environments ausgelesen, wenn diese ApplicationVersion ausgerollt wird.

Erlaubte Sektionen:

- `option_settings` Configuration Options

.ebextensions

Konfigurationen an der ApplicationVersion

Wenn im Rootverzeichnis einer ApplicationVersion ein Directory mit dem Namen `.ebextensions` existiert, werden YAML- und JSON-Dateien für die Konfiguration von Environments ausgelesen, wenn diese ApplicationVersion ausgerollt wird.

Erlaubte Sektionen:

- `option_settings` Configuration Options
- `packages`, `groups`, `users`, `sources`, `files`, `commands`, `services`

.ebextensions

Konfigurationen an der ApplicationVersion

Wenn im Rootverzeichnis einer ApplicationVersion ein Directory mit dem Namen `.ebextensions` existiert, werden YAML- und JSON-Dateien für die Konfiguration von Environments ausgelesen, wenn diese ApplicationVersion ausgerollt wird.

Erlaubte Sektionen:

- `option_settings` Configuration Options
- `packages`, `groups`, `users`, `sources`, `files`, `commands`, `services`
- `container_commands` Wie Commands aus `AWS::Cloudformation::Init`, aber werden ausgeführt nachdem der Rest der Anwendung ausgerollt wurde

.ebextensions

Konfigurationen an der ApplicationVersion

Wenn im Rootverzeichnis einer ApplicationVersion ein Directory mit dem Namen `.ebextensions` existiert, werden YAML- und JSON-Dateien für die Konfiguration von Environments ausgelesen, wenn diese ApplicationVersion ausgerollt wird.

Erlaubte Sektionen:

- `option_settings` Configuration Options
- `packages`, `groups`, `users`, `sources`, `files`, `commands`, `services`
- `container_commands` Wie Commands aus `AWS::Cloudformation::Init`, aber werden ausgeführt nachdem der Rest der Anwendung ausgerollt wurde
- `Resources` Zusätzliche Cloudformation Resources die in der Environment erstellt werden. Auf in der Environment gesetzte Configuration Options kann mit `Fn::GetOptionSetting` zugegriffen werden.

Die 4. Resource: Das Configuration Template

AWS::ElasticBeanstalk::ConfigurationTemplate

- Speicherformat für Configuration Options

Die 4. Resource: Das Configuration Template

AWS::ElasticBeanstalk::ConfigurationTemplate

- Speicherformat für Configuration Options
- Lässt sich einer Environment anhängen

Die 4. Resource: Das Configuration Template

AWS::ElasticBeanstalk::ConfigurationTemplate

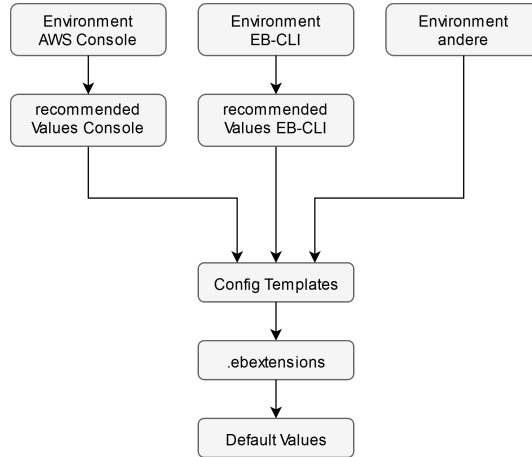
- Speicherformat für Configuration Options
- Lässt sich einer Environment anhängen
- Kann auch die Platform für die Environment festlegen

Die 4. Resource: Das Configuration Template

AWS::ElasticBeanstalk::ConfigurationTemplate

- Speicherformat für Configuration Options
- Lässt sich einer Environment anhängen
- Kann auch die Platform für die Environment festlegen
- Kann zusätzlich zu eigenen Configuration Options auch bei der Erstellung Configuration Options aus einer bestehenden Environment oder einem anderen Configuration Template erben

Configuration Option Precedence



Wordpress vorbereiten

```
$ wget https://wordpress.org/latest.tar.gz  
$ tar -xvzf ./latest.tar.gz wordpress  
$ vim wordpress/wp-config.php
```

wp-config.php

```
<?php
define('DB_NAME', $_SERVER['RDS_DB_NAME']);
define('DB_USER', $_SERVER['RDS_USERNAME']);
define('DB_PASSWORD', $_SERVER['RDS_PASSWORD']);
define('DB_HOST', $_SERVER['RDS_HOSTNAME'] . ':' . $_SERVER['RDS_PORT']);
...
```

eb init

```
$ eb init
```

```
Select a default region
```

- 1) us-east-1 : US East (N. Virginia)
 - 2) us-west-1 : US West (N. California)
 - 3) us-west-2 : US West (Oregon)
 - 4) eu-west-1 : EU (Ireland)
- ```
(default is 3): 4
```

```
Select an application to use
```

- 1) [ Create new Application ]
- ```
(default is 1): 1
```

```
Enter Application Name
```

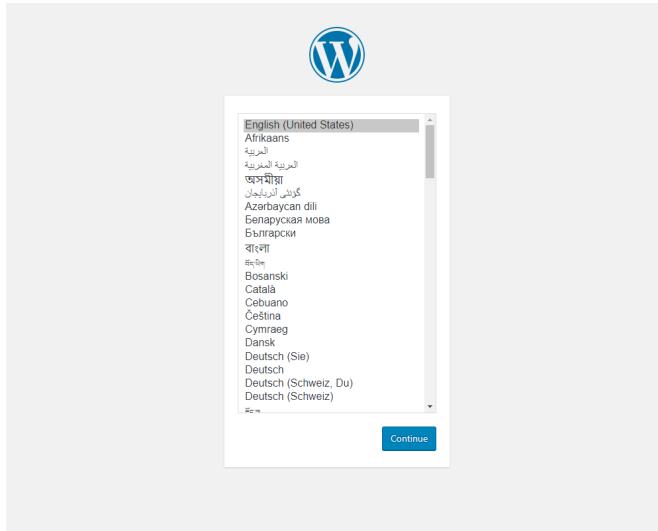
```
(default is "wordpress"): wordpress
```

```
Application wordpress has been created.
```

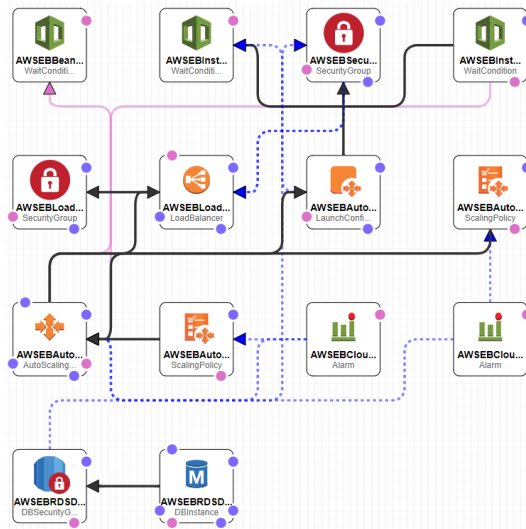
eb create

```
$ eb create -db wordpress1
Enter an RDS DB username (default is "ebroot"):
Enter an RDS DB master password:
Creating application version archive "app-171102_151050".
Uploading: [#####] 100% Done...
--- Waiting for Application Versions to be pre-processed ---
Finished processing application version app-171102_151050
Environment details for: wordpress1
  Application name: wordpress
  Region: eu-west-1
  Deployed Version: app-171102_151050
  Environment ID: e-jgxsxns3dk
  Platform: arn:aws:eb:eu-west-1::platform/PHP 5.4 running on Amazon Linux/2.5.0
  Tier: WebServer-Standard
  CNAME: UNKNOWN
  Updated: 2017-11-02 15:11:00.901000+00:00
Printing Status:
INFO: createEnvironment is starting.
INFO: Using eb-eu-west-1-163962199350 as Amazon S3 bucket for environment data.
...
INFO: Environment health has transitioned from Pending to Ok. Initialization completed 2
seconds ago and took 9 minutes.
INFO: Successfully launched environment: wordpress1
```

Ergebnisse



Ergebnisse!



Die Datenbank: Down the rabbit hole

```
eb create -db wordpress1
```


Die Datenbank: Down the rabbit hole

```
eb create -db wordpress1
```

Wie haben sie das gemacht?

Die Datenbank: Down the rabbit hole

```
eb create -db wordpress1
```

Wie haben sie das gemacht?

Templates für Templates

Probleme mit Cloudformation

Wir schreiben ein Cloudformation Template,

Probleme mit Cloudformation

Wir schreiben ein Cloudformation Template, das einen Stack beschreibt,

Probleme mit Cloudformation

Wir schreiben ein Cloudformation Template, das einen Stack beschreibt, das ein Werkzeug definiert,

Probleme mit Cloudformation

Wir schreiben ein Cloudformation Template, das einen Stack beschreibt, das ein Werkzeug definiert, das ein Cloudformation Template aus anderen Templates baut,

Probleme mit Cloudformation

Wir schreiben ein Cloudformation Template, das einen Stack beschreibt, das ein Werkzeug definiert, das ein Cloudformation Template aus anderen Templates baut, um die Ressourcen zu definieren, auf denen unsere Applikation laufen soll.

Probleme mit Cloudformation

Wir schreiben ein Cloudformation Template, das einen Stack beschreibt, das ein Werkzeug definiert, das ein Cloudformation Template aus anderen Templates baut, um die Ressourcen zu definieren, auf denen unsere Applikation laufen soll.

Application Versions sollen von Elastic Beanstalk selber verwaltet werden

Probleme mit Cloudformation

Wir schreiben ein Cloudformation Template, das einen Stack beschreibt, das ein Werkzeug definiert, das ein Cloudformation Template aus anderen Templates baut, um die Ressourcen zu definieren, auf denen unsere Applikation laufen soll.

Application Versions sollen von Elastic Beanstalk selber verwaltet werden, aber Cloudformation Stacks sind idempotent.

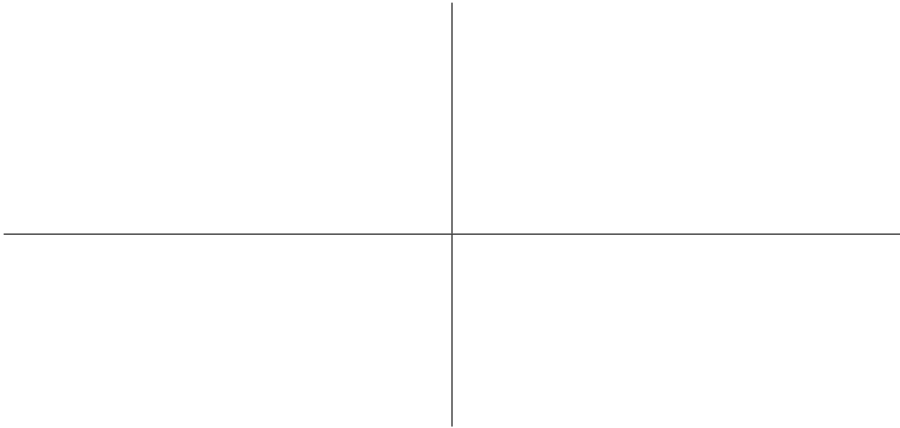
Probleme mit Cloudformation

Wir schreiben ein Cloudformation Template, das einen Stack beschreibt, das ein Werkzeug definiert, das ein Cloudformation Template aus anderen Templates baut, um die Ressourcen zu definieren, auf denen unsere Applikation laufen soll.

Application Versions sollen von Elastic Beanstalk selber verwaltet werden, aber Cloudformation Stacks sind idempotent.

Die anderen Elastic Beanstalk Ressourcen haben unterschiedliche Lebenszyklen.

Ein Lösungsansatz



Ein Lösungsansatz

Application Stack

Definiert:

- Application
- Dev Configuration Template
- Prod Configuration Template

Lebenszyklus: lang

Ein Lösungsansatz

Application Stack

Definiert:

- Application
- Dev Configuration Template
- Prod Configuration Template

Lebenszyklus: lang

Dev Environment Stack

Definiert:

- VPC mit 2 Subnetzen
- Eine RDS Instanz
- Die Development Environment

Lebenszyklus: kurz

Prod Network Stack

Definiert:

- VPC mit 4 Subnetzen
- Multi-AZ RDS Instanz

Lebenszyklus: mittel

Ein Lösungsansatz

Application Stack

Definiert:

- Application
- Dev Configuration Template
- Prod Configuration Template

Lebenszyklus: lang

Dev Environment Stack

Definiert:

- VPC mit 2 Subnetzen
- Eine RDS Instanz
- Die Development Environment

Lebenszyklus: kurz

Prod Network Stack

Definiert:

- VPC mit 4 Subnetzen
- Multi-AZ RDS Instanz

Lebenszyklus: mittel

Prod Environment Stack

(2 Abhängigkeiten)

Definiert:

- Die Production Environment

Lebenszyklus: kurz

Dinge, die ich (bis jetzt) ausgelassen habe.

env.yaml

Dinge, die ich (bis jetzt) ausgelassen habe.

env.yaml

Environment Groups

Dinge, die ich (bis jetzt) ausgelassen habe.

env.yaml

Environment Groups

Custom Platforms

Dinge, die ich (bis jetzt) ausgelassen habe.

env.yaml

Environment Groups

Custom Platforms (Packer Templates)

Ein Wort der Warnung

Ein Wort der Warnung

Dinge haben mehrere Namen

Ein Wort der Warnung

Dinge haben mehrere Namen

- Configuration Template = Saved Configuration

Ein Wort der Warnung

Dinge haben mehrere Namen

- Configuration Template = Saved Configuration
- Environment Tier/Type
- Solution Stack = !(Custom) Platform
- (Application) Version Lifecycle Config / (Application) Resource Lifecycle Config / Lifecycle Policy

Ein Wort der Warnung

Dinge haben mehrere Namen

- Configuration Template = Saved Configuration
- Environment Tier/Type
- Solution Stack = !(Custom) Platform
- (Application) Version Lifecycle Config / (Application) Resource Lifecycle Config / Lifecycle Policy

Es gibt für alles eine Ausnahme

Fazit