

1. Was ist AWS CloudFormation und welche Hauptprobleme löst es?

CloudFormation bedient sich dem Prinzip der Infrastruktur als Code und ist eine Hilfestellung beim einrichten oder abschalten einer schon vorgefertigten Blaupause, für eine Infrastruktur. Somit kann Zeit gespart und effizienter gearbeitet werden.

2. Wie unterscheiden sich imperatives und deklaratives Modellieren in Bezug auf die Infrastruktur als Code, welches davon trifft auf AWS Cloudformation zu.

Deklarativ heißt, dass der Nutzer festlegt, wie der Zustand am Ende aussehen soll und Imperativ heißt, dass der Nutzer eine Folge von Schritten angibt, die zum gewünschten Ergebnis führen.

3. Was ist ein AWS CloudFormation Template und in welchen Formaten kann es geschrieben werden?

Das Template ist quasi ein Code, der eine Infrastruktur einrichtet und kann im JSON und YAML Format geschrieben werden.

4. Was sind Stacks in AWS CloudFormation und wie helfen sie bei der Verwaltung von AWS-Ressourcen?

Cloud Formation Stacks sind quasi das Template oder besser gesagt der Zusammenschluss von AWS Ressourcen im Template/ Blaupause.

Sie helfen halt dabei Zeit zu sparen und effizienter zu arbeiten, abgesehen davon lassen sich die Kosten sehr gut schätzen.

5. Was ist AWS Elastic Beanstalk und für welche Art von Anwendungen ist es am besten geeignet?

Elastic Beanstalk ist ein Service, der bei der Einrichtung der Infrastruktur hilft, sodass sich Entwickler mehr auf das Programmieren konzentrieren können. Hier wurde Terraform genannt und hat das Format .tf

6. Wie unterscheidet sich AWS Elastic Beanstalk von anderen AWS-Diensten wie Amazon EC2 oder AWS CloudFormation?

Beanstalk wurde für Entwickler entworfen, um diese Zeit zu sparen oder besser gesagt, ihre Zeit mit der Produktion effizienter zu gestalten. Darüber hinaus ist Beanstalk ein Service, der bei der Einrichtung hilft und andere AWS Ressourcen aktiviert, dabei entstehen lediglich Kosten für die besagten Ressourcen.

7. Welche Bereitstellungsstrategien bietet AWS Elastic Beanstalk an, und wie unterscheiden sie sich voneinander?

Blue and Green Deployment, es werden zeitgleich zwei separate Umgebungen betrieben, wobei in der grünen Umgebung die neusten Updates sind und nachdem sie erfolgreich getestet wurden, automatisch auf die blaue Umgebung geschaltet.

Rolling Deployment ist, wenn Instanzen mit der selben Anwendung nach und nach aktualisiert werden, um Ausfallzeiten zu minimieren.

8. Was ist Git, und warum wird es in der Softwareentwicklung verwendet?

Git ist ein Open Source Versionskontrollsystem, welches Entwicklern ermöglicht, Änderungen an Daten zu verfolgen und zu verwalten.

9. Was ist ein Branch in Git, und für welche Zwecke wird er typischerweise verwendet?

Ein Branch ist eine Abzweigung vom Main Branch, ein Branch ist eine Kopie des Main und bietet somit die Möglichkeit am Code zu arbeiten, ohne dabei den Code im Main Branch einzuschränken.

10. Was sind einige Best Practices für das Nutzen von Git-Branches in einem Projekt?

Aussagekräftige Namen für alle Branches verwenden

Nebenbranches für kommende Änderungen anlegen

Nach erfolgreicher Prüfung Nebenbranch in Mainbranch mergen und Nebenbranch löschen

Ansonsten add, commit und push - optimalerweise pull request vom Mainbranch

11. Was ist ein Merge Konflikt?

Wenn beim Zusammenführen vom Neben Branch zum Main Branch sich Zeilen im Code überschneiden.

RIP.

