

M346, IaC.

Thema:

Infrastructure as Code

Lernziele

- Sie verstehen die Vorteile von IaC
- Sie unterscheiden den deklarativen und imperativen Ansatz

Aufgabe 1: IaC | Einzelarbeit

Lesen Sie den Artikel [Was ist IaC](#) und beantworten Sie folgende Fragen:

Was ist IaC?

Infrastructure as Code (IaC) ist ein Ansatz zur Verwaltung und Bereitstellung von IT-Infrastruktur durch automatisierte Skripte und Konfigurationsdateien, anstatt durch manuelle Prozesse.
Mit IaC werden Clouddienste wie EC2, S3... durch code und nicht manuelle Prozesse bereitgestellt

Was sind die Beweggründe für IaC

Die Beweggründe für IaC sind die Automatisierung, Konsistenz und Skalierbarkeit der IT-Infrastruktur, um Effizienz zu steigern und Fehler zu reduzieren

jedes Mal, wenn der code ausgeführt wird, erhält man dieselbe Umgebung

keine undokumentierten Ad-hoc Änderungen

Versionskontrolle

Automatisierung --> Cloud native Ansatz

Was ist der Unterschied zwischen dem deklarativen und dem imperativen Ansatz?

Der deklarative Ansatz definiert den gewünschten Zustand der Infrastruktur, während der imperativen Ansatz die konkreten Schritte zur Erreichung dieses Zustands festlegt. Beim deklarativen Ansatz wird die Konfiguration beschrieben, und ein IaC-Tool wendet sie an, während beim imperativen Ansatz die Befehle definiert werden, die in der richtigen Reihenfolge ausgeführt werden müssen.

-dekla

Was ist ein wichtiger Teil von IaC, um die Codefiles zu verwalten?

Ein wichtiger Teil von IaC, um die Code-Dateien zu verwalten, ist die Versionskontrolle. Damit können Sie Änderungen nachverfolgen, verschiedenen Versionen Ihrer Infrastrukturkonfigurationen verwalten und Zusammenarbeit ermöglichen, was die Verwaltung und den Austausch von Code erleichtert

Versionskontrolle kann via git-Repo laufen