

Plan

1. Wiederholung
2. Essential Services
3. Best Practices
4. Well architected Framework
5. kleines Quiz

Wiederholung

Erwartungen - Lernziel

- Ein Grundverständnis dafür, was ein paar der wichtigsten AWS Services machen
- Grundverständnis für Design Prinzipien
- Wissen was die 6 Säulen des Well Architected Framework sind
- Grundverständnis dafür was die 6 Säulen bedeuten

“featured” → viel benutzte AWS Services

Amazon EC2

Virtual servers in the cloud

Amazon Simple Storage Service (S3)

Scalable storage in the cloud

Amazon Aurora

High performance managed relational database with full MySQL and PostgreSQL compatibility

Amazon DynamoDB

Managed NoSQL database

Amazon RDS

Managed relational database service for PostgreSQL, MySQL, MariaDB, SQL Server, Oracle, and Db2

AWS Lambda

Run code without thinking about servers

Amazon VPC

Isolated cloud resources

Amazon Lightsail

Launch and manage virtual private servers

Amazon SageMaker

Build, train, and deploy machine learning models at scale

AWS Cloudformation



Code infrastructure

Code your infrastructure from scratch with the CloudFormation template language, in either YAML or JSON format, or start from many available sample templates



Amazon S3

Check out your template code locally, or upload it into an S3 bucket



AWS CloudFormation

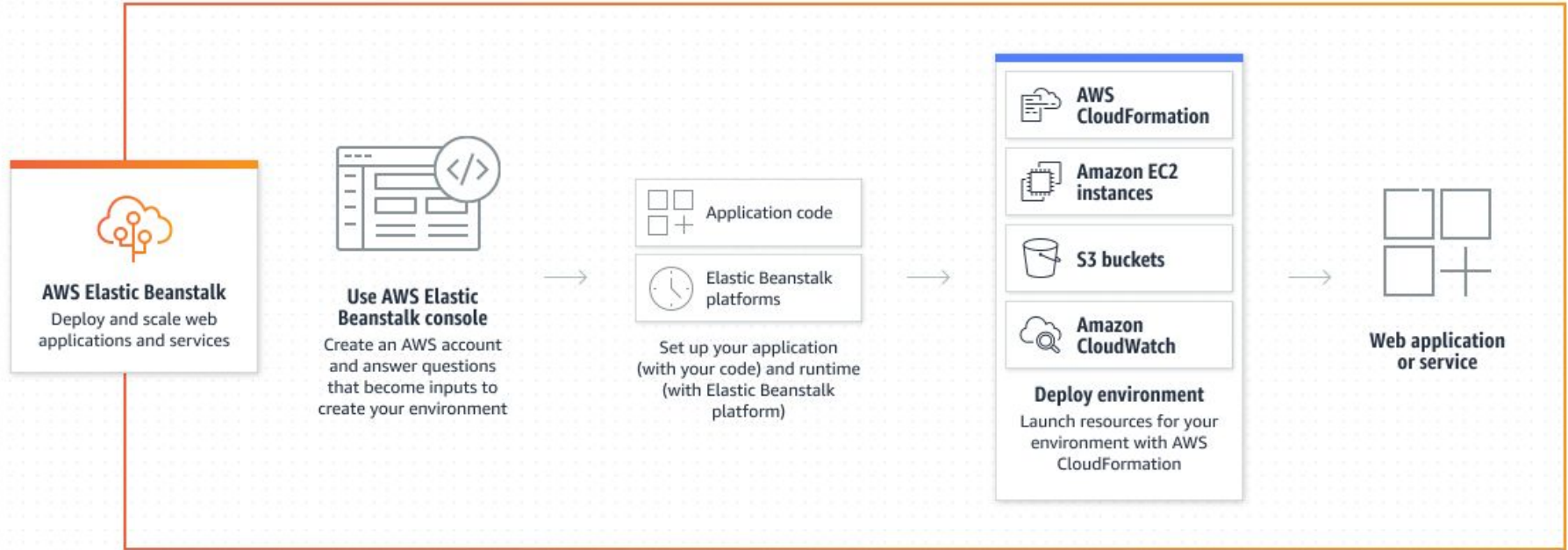
Use AWS CloudFormation via the browser console, command line tools or APIs to create a stack based on your template code



Output

AWS CloudFormation provisions and configures the stacks and resources you specified on your template

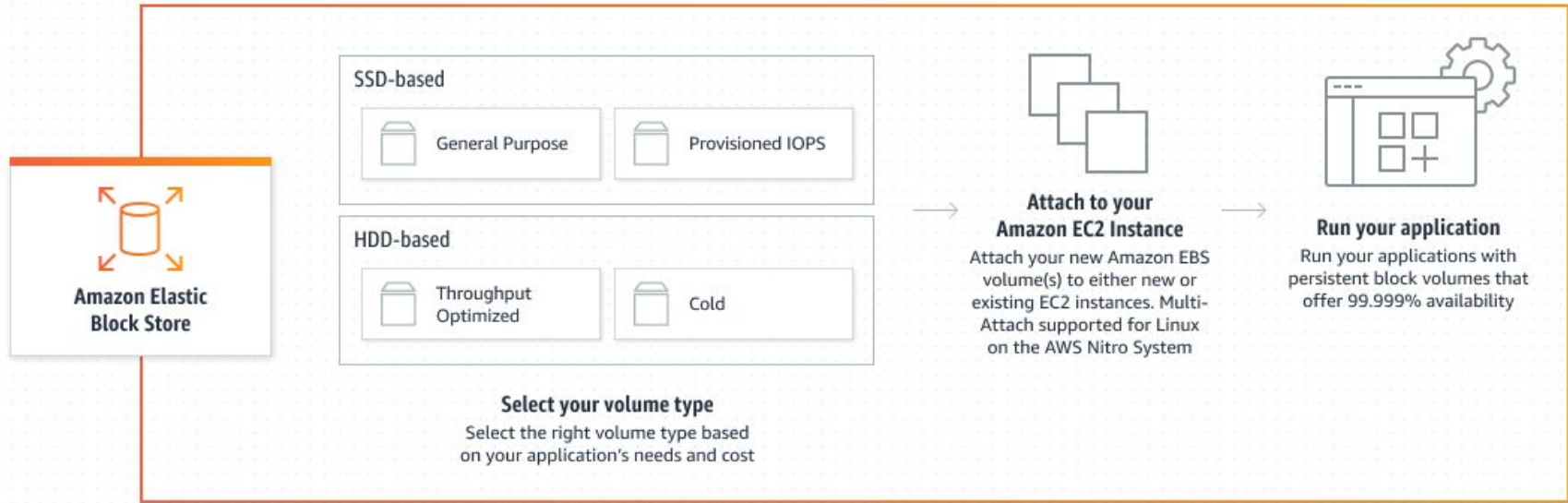
AWS Elastic Beanstalk



Amazon Elastic Block Store

- Speichern ausserhalb von Instanzen
- gebunden an availability zones

- Verhalten: "Netzwerk USB stick"



Design Prinzipien & Well architected Framework

AWS Best Practices - Design Prinzipien

- nicht schätzen, wie viel Kapazität man braucht - Autoscaling
- Systeme im Produktionsumfang testen
- Automatisierung der Erstellung der Architektur - CloudFormation
- Entwicklung der Architektur erlauben - keine Monolithen
- Verbessern durch Stresstests - Netflix Chaos Monkey

AWS Best Practices - Design Prinzipien

- Skalierbarkeit (Scalability): vertikal und horizontal
- Wegwerfbare Ressourcen (Disposable Resources): Servers sollen ersetzbar und einfach konfigurierbar sein
- Automatisierung (Automation): IaaS, Auto Scaling
- Lockere Kopplung (Loose Coupling):
 - Monoliten (schlecht) sind große Applikationen die so designt sind, dass sie sich nicht ohne weiteres unterteilen lassen.
 - Applikationen in kleinere locker gekoppelte Komponenten unterteilen (gut)
 - Der Ausfall einer Komponente sollte nicht zum Ausfall weiterer führen
- Dienste keine Server (Services, not Servers): (Nachhaltigkeit)
 - nicht nur EC2 nutzen
 - managed services, Datenbanken, serverless nutzen

Well Architected Framework - 6 Säulen

1. Operational Excellence
2. Security
3. Reliability
4. Performance Efficiency
5. Cost Optimization
6. Sustainability

1. Operational Excellence
2. Sicherheit
3. Zuverlässigkeit
4. Leistungseffizienz
5. Kostenoptimierung
6. Nachhaltigkeit

Operational Excellence

- umfasst die Fähigkeit, Systeme zu betreiben und zu überwachen, um einen geschäftlichen Nutzen zu erzielen und unterstützende Prozesse und Verfahren kontinuierlich zu verbessern
- Design Prinzipien:
 - Operationen als Code ausführen - Infrastructure as code
 - Dokumentation kommentieren - die Erstellung von kommentierter Dokumentation nach jedem Build automatisieren
 - häufige, kleine, reversible Änderungen vornehmen - sodass man im Notfall jeden Fehler rückgängig machen kann
 - Abläufe häufig verbessern
 - Fehler erwarten
 - Aus Fehlern lernen

Operational Excellence - Services

Vorbereitung:

- AWS Config
- AWS Cloudformation

Betrieb:

- AWS Cloudformation
- AWS Config
- AWS CloudTrail
- AWS CloudWatch
- AWS X-Ray

Entwicklung:

- AWS CloudFormation
- AWS CodeBuild
- AWS CodeCommit
- AWS CodeDeploy
- AWS CodePipeline

Wir werden Entwicklung mit Github oder Gitlabs (oder Bitbucket) machen.

Sicherheit - Security

- Die Fähigkeit Information und Systeme zu schützen und trotzdem Werte zu generieren für das Unternehmen. Durch Gefahreneinschätzungen (***risk assessment***) und durch Minderungsstrategien (***mitigation strategies***)
- Design Prinzipien
 - eine starke Identitätsgrundlage schaffen - Zentralisierung der Rechteverwaltung und Verringerung der Abhängigkeit von langfristigen Berechtigungsnachweisen - Prinzip der geringsten Berechtigung (***principle of least privilege***) - ***IAM*** → ***identity access management***
 - Rückverfolgbarkeit - Integration von Protokollen und Metriken in Systeme, um automatisch reagieren und Maßnahmen ergreifen zu können
 - Sicherheit auf allen Ebenen anwenden - wie Edge-Netzwerk, VPC, Subnetz, Load Balancer, jede Instanz, jedes Betriebssystem und jede Anwendung
 - best practices der Sicherheit implementieren
 - Schutz von Daten bei der Übertragung und im Ruhezustand (***in transit and at rest***) - Verschlüsselung, Tokenisierung und Zugangskontrolle
 - Menschen von Daten fernhalten - die Notwendigkeit des direkten Zugriffs oder der manuellen Verarbeitung von Daten verringern oder beseitigen
 - Vorbereitung auf Sicherheitsereignisse - Durchführung von Simulationen zur Reaktion auf Zwischenfälle und Einsatz von Automatisierungstools zur Beschleunigung von Erkennung, Untersuchung und Wiederherstellung

Sicherheit - Security

- Identitätsmanagement: IAM, AWS-STS, MFA token, AWS Organizations
- “Detective Controls”: AWS Config, AWS CloudTrail, Amazon CloudWatch, Inspektor (Betrachtet einzelne Instanzen)
- Schutz der Infrastruktur: Amazon CloudFront (DDoS), Amazon VPC(schützt durch NACL), AWS Shield (DDoS), WAF (WebApplication Firewall), Inspektor (Betrachtet einzelne Instanzen)

Erklärung DDoS: Distributed Denial of Service

Wenn mein System auf 10.000 Nutzer ausgelegt und ich 1.000.000 anfragen von verschiedenen IPs bekomme.

z.B. ich sage 1.000.000 Leute, dass sie einen Link öffnen sollen → Webseite bricht zusammen (wenn sie darauf nicht ausgelegt ist)

Aufgaben für Breakouts

1. Operational Excellence
2. Security
- 3. Reliability**
- 4. Performance Efficiency**
- 5. Cost Optimization**
- 6. Sustainability**

1. Operational Excellence
2. Sicherheit
- 3. Zuverlässigkeit**
- 4. Leistungseffizienz**
- 5. Kostenoptimierung**
- 6. Nachhaltigkeit**

Aufgabe in 4 Gruppen:

Jede Gruppe hat ein eigenes Thema **3, 4, 5** oder **6**.

1. Sucht euch online und hauptsächlich von der AWS Seite ressourcen die jeweiligen Design Prinzipien beschreiben.
2. Sucht AWS Services raus, die zur Umsetzung dieser Prinzipien helfen.
3. Fasst die Ergebnisse zu 2 - 3 Seiten Präsentation zusammen. (Slides oder Docs)

Zuverlässigkeit - Reliability

On-site	IaaS	PaaS	SaaS
Applications	Applications	Applications	Applications
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
O/S	O/S	O/S	O/S
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking



You manage



Service provider manages