
CLOUD
INSIGHTS
09. März 2022

Cloud Security in der Praxis – Am Beispiel von AWS

9. März 2022

ORGANISATORISCHES

Diese Online-Veranstaltung wird aufgezeichnet.

Wir nutzen das freundliche „DU“ ☺

Nutzt den Chat Bereich (auch anonym), um eure Fragen zu stellen.

Im Nachgang erhaltet ihr einen Link zur Aufzeichnung und Präsentation.

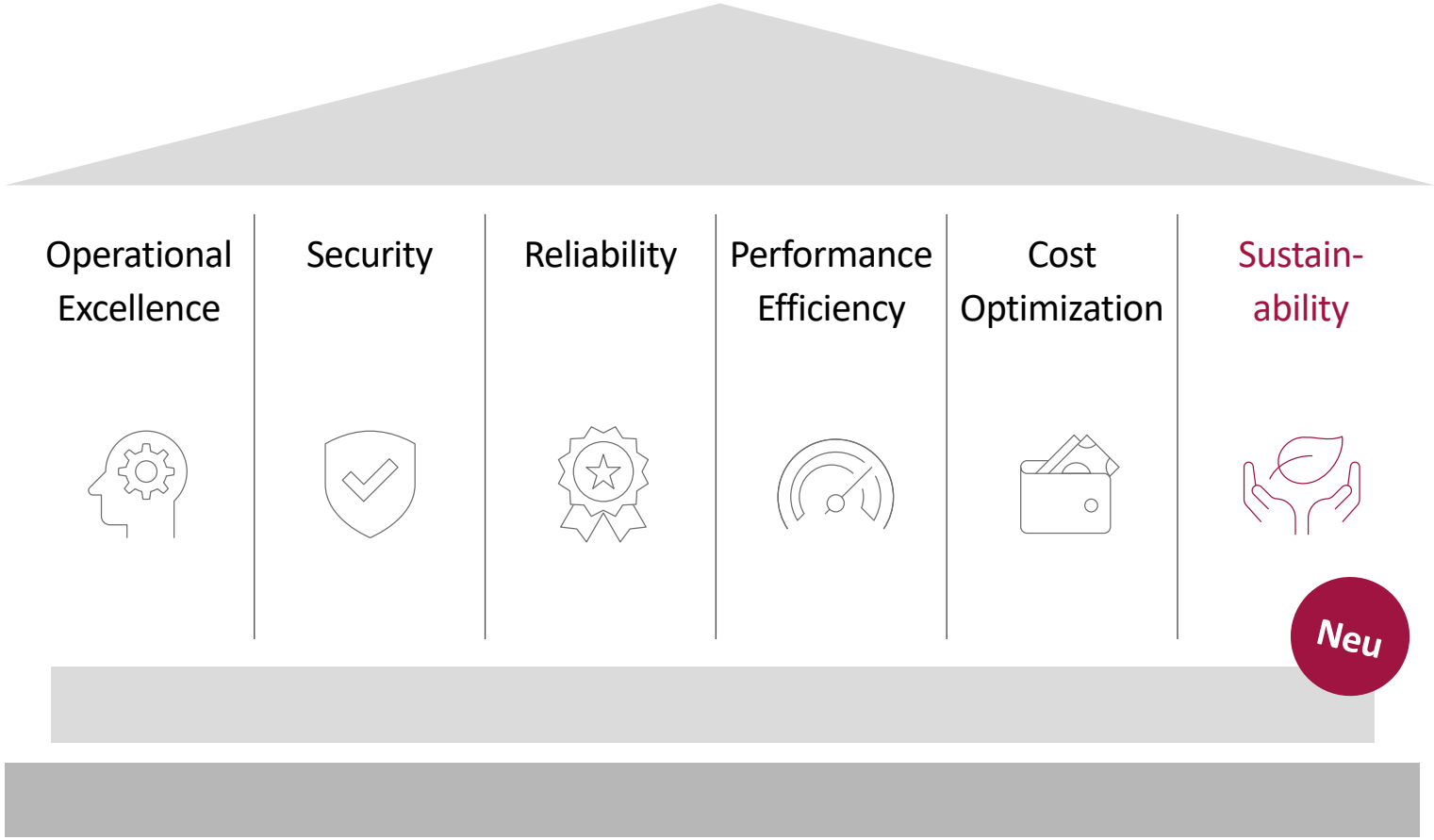
Kurzer Einblick „AWS Well-Architected Framework“ – Säule Sicherheit

Das AWS Well-Architected Framework mit seinen 6 Säulen

Das AWS Well-Architected Framework besteht aus 6 Säulen, welche uns Konzepte und Patterns für die Architektur von Workloads auf AWS an die Hand geben.

Diese Säulen sind:

- Operational Excellence
- Sicherheit
- Zuverlässigkeit
- Leistung und Effizienz
- Kostenoptimierung
- Nachhaltigkeit

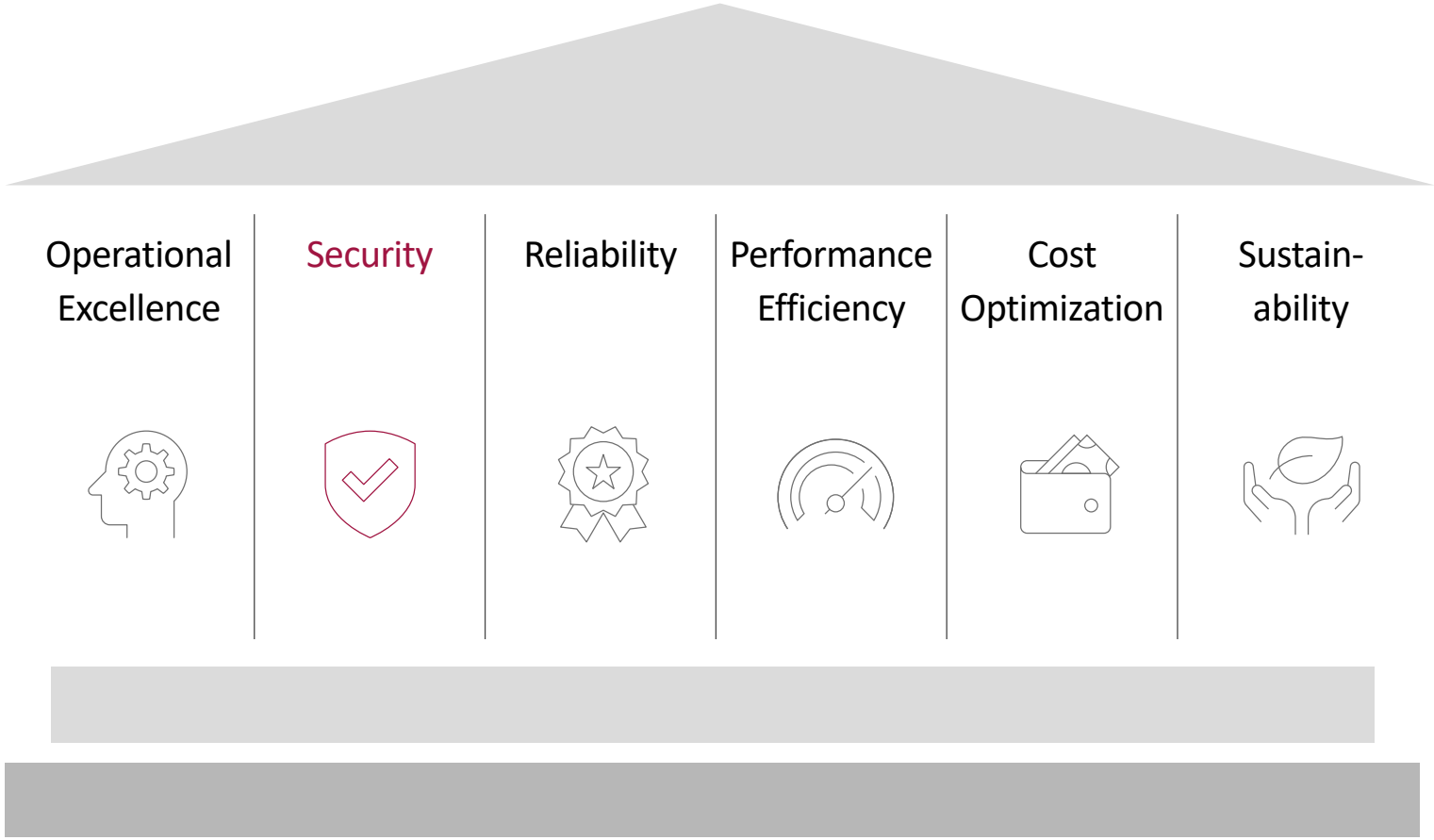


Quelle: https://docs.aws.amazon.com/wellarchitected/latest/framework/welcome.html

Das AWS Well-Architected Framework mit seinen ~~5~~ 6 Säulen

Innerhalb dieses Vortrages wollen wir uns nun mit der Säule Sicherheit weiter beschäftigen.

Diese besteht aus verschiedensten Teilgebieten und Aspekten, wovon wir uns nachfolgend diverse anschauen.



Einblick in bestimmte Aspekte der Säule Sicherheit

Grundlagen	• Geteilte Verantwortung beim Thema Sicherheit
Identity/Access Management	• Verwaltung von Identitäten und deren Rechten
Schutz von Ressourcen	• Schutz von Ressourcen innerhalb eines Netzwerks auf AWS

Einblick in bestimmte Aspekte der Säule Sicherheit

Grundlagen

- Geteilte Verantwortung beim Thema Sicherheit

Identity/Access Management

- Verwaltung von Identitäten und deren Rechten

Schutz von Ressourcen

- Schutz von Ressourcen innerhalb eines Netzwerks auf AWS

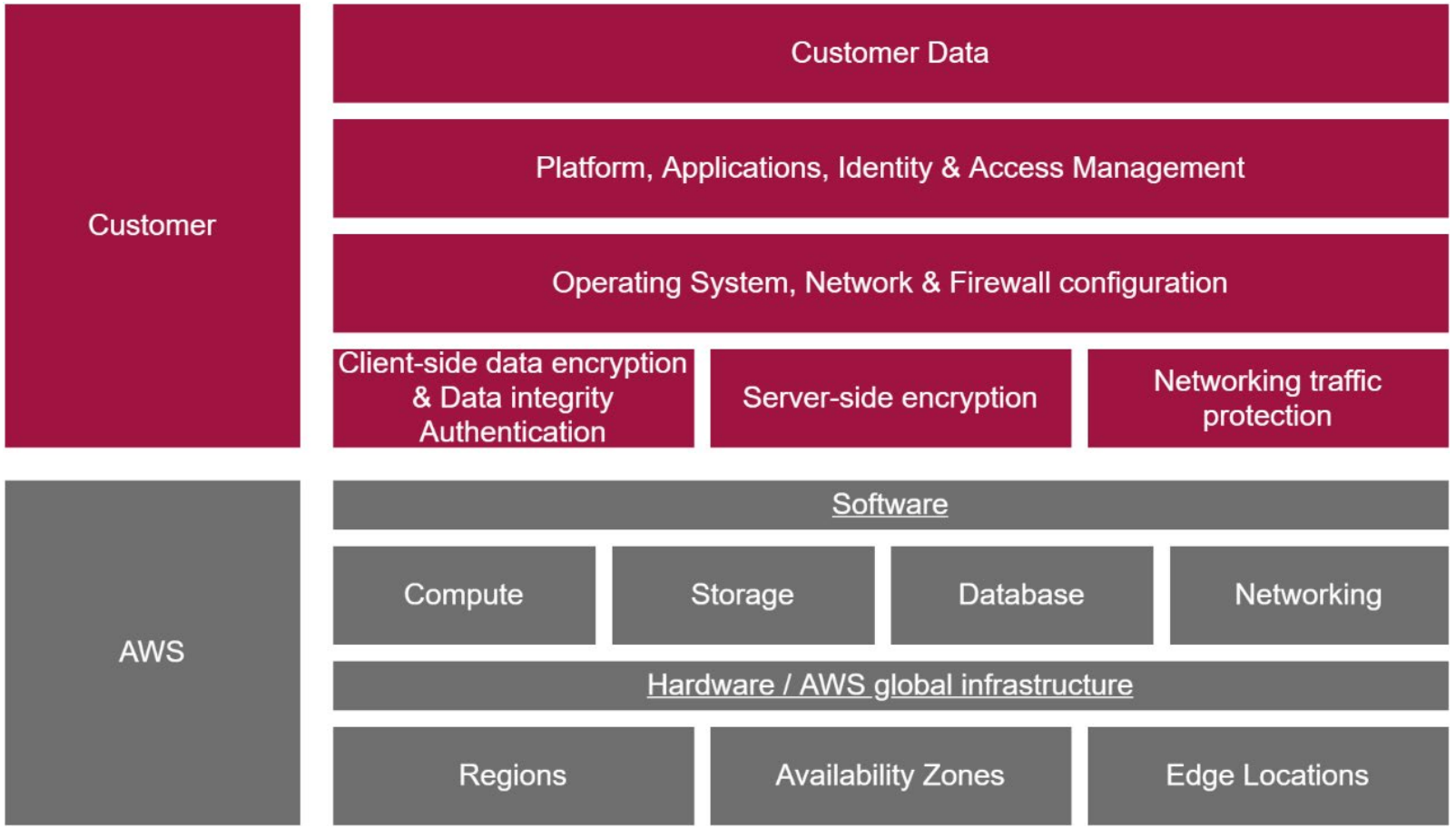
Grundlagen

Bevor wir unseren Workload auf AWS absichern können, müssen wir uns erstmal bewusst werden, für was wir auf AWS verantwortlich sind. Hier existiert eine **geteilte Verantwortung** mit AWS.

Für AWS bedeute das vor allem die Verantwortlichkeit für die vorhandene Infrastruktur, Zugängen zu Gebäuden oder die korrekte Funktionsweise der Software der Services. Sie sind somit verantwortlich für die **Sicherheit der Cloud**.

Für den Kunden bedeutet es die Verantwortlichkeit für die Daten und deren Verschlüsselung, Konfigurationen von Netzwerken und Betriebssystemen und Identity Management des Accounts. Er ist somit verantwortlich für die **Sicherheit in der Cloud**.

Grundlagen


Customer
Customer Data
Platform, Applications, Identity & Access Management
Operating System, Network & Firewall configuration
Client-side data encryption & Data integrity Authentication
Server-side encryption
Networking traffic protection
AWS
Software
Compute
Storage
Database
Networking
Hardware / AWS global infrastructure
Regions
Availability Zones
Edge Locations

Grundlagen

Kann ich auch irgendwo die AWS Compliance Reports einsehen?

Sollte die Notwendigkeit bestehen die Sicherheit- und Compliance-Reports von AWS einzusehen, wird dafür der Dienst **AWS Artifact** angeboten. Dieser beinhaltet verschiedenste Reports von 3rd-party Auditoren.

Quelle https://aws.amazon.com/artifact/

Einblick in bestimmte Aspekte der Säule Sicherheit

Grundlagen

- Geteilte Verantwortung beim Thema Sicherheit

Identity/Access Management

- Verwaltung von Identitäten und deren Rechten

Schutz von Ressourcen

- Schutz von Ressourcen innerhalb eines Netzwerks auf AWS

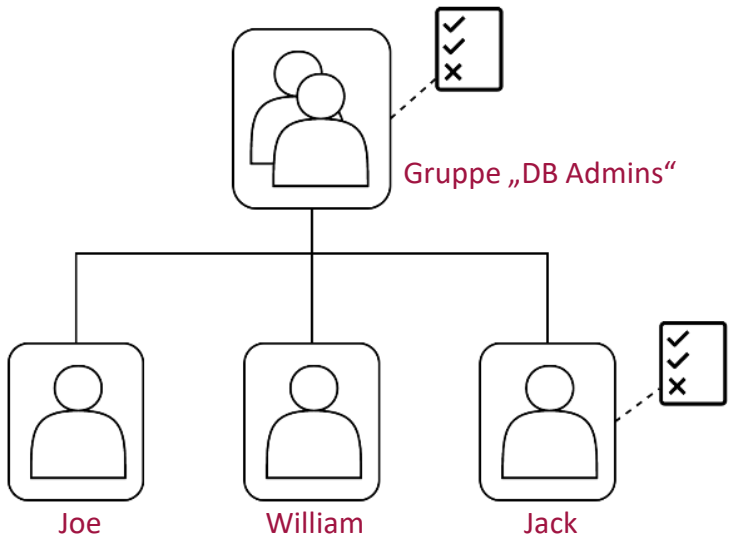
Verwaltung von Identitäten und deren Rechten

Zuallererst: Gebt nicht jedem Nutzer auf dem AWS Account alle verfügbaren Rechte!

Auf AWS können jedem Nutzer einzelnen oder Gruppen von Nutzern dedizierte Rechte zugewiesen werden mit Hilfe des IAM (Identity & Access Management) Services.

- Ein IAM **Nutzer** ist somit eine Identität, welche mit den Ressourcen auf dem AWS Account interagiert
- Eine IAM **Gruppe** ist eine Sammlung von IAM Nutzern
- Eine IAM **Policy** ist ein Dokument, welches die Rechte eines Nutzer oder einer Gruppe festlegt

Jedem Nutzer sollten genau die Rechte gegeben werden, welche er benötigt, um die ihm zugedachte Aufgabe zu erfüllen.



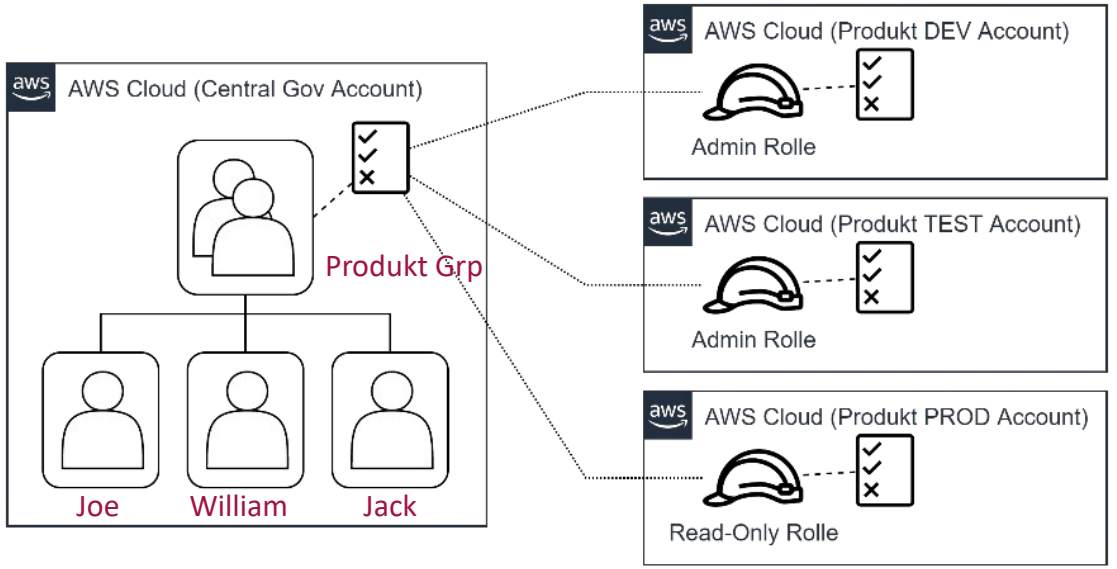
Verwaltung von Identitäten und deren Rechten

Was ist, wenn ich aber jemandem nur temporäre Rechte geben möchte?

Dafür gibt es auf AWS das Konzept einer sogenannten IAM **Rolle**.

Diese Rolle kann angenommen werden um die daran hinterlegten Berechtigungen temporär anzunehmen. **Wichtig:** Dabei werden alle Berechtigungen, die der Nutzer direkt oder durch eine Gruppe hat, abgelegt.

Diese temporäre Art der Berechtigungsattestierung kann z.B. genutzt werden, um Nutzern eines anderen AWS Accounts Rechte auf dem eigenen zu geben, ohne für diese extra Nutzer mit Name/Passwort anzulegen.



Einblick in bestimmte Aspekte der Säule Sicherheit

Grundlagen

- Geteilte Verantwortung beim Thema Sicherheit

Identity/Access Management

- Verwaltung von Identitäten und deren Rechten

Schutz von Ressourcen

- Schutz von Ressourcen innerhalb eines Netzwerks auf AWS

Schutz von Ressourcen innerhalb eines Netzwerks auf AWS

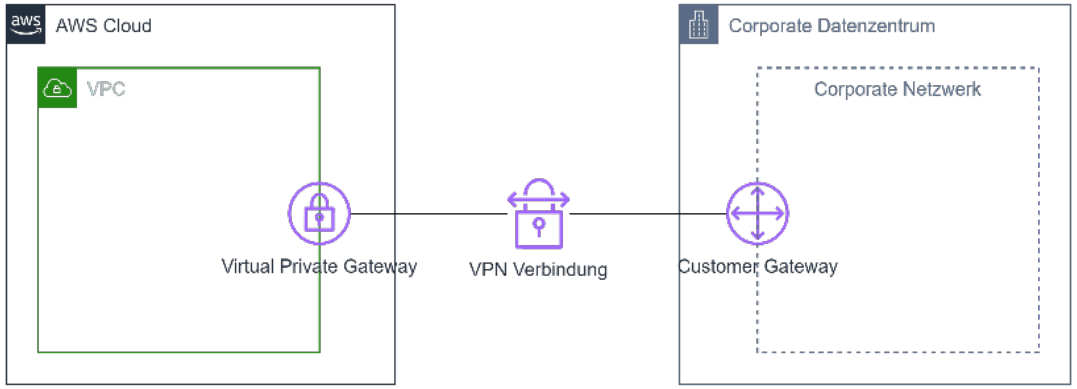
Wie kann ich mein on-premises Netzwerk mit AWS verbinden für eine hybride Cloud Lösung?

Sollte die Notwendigkeit eine sichere Verbindung zwischen dem Netzwerk des lokalen Datenzentrums mit AWS gibt es verschiedene Möglichkeiten.

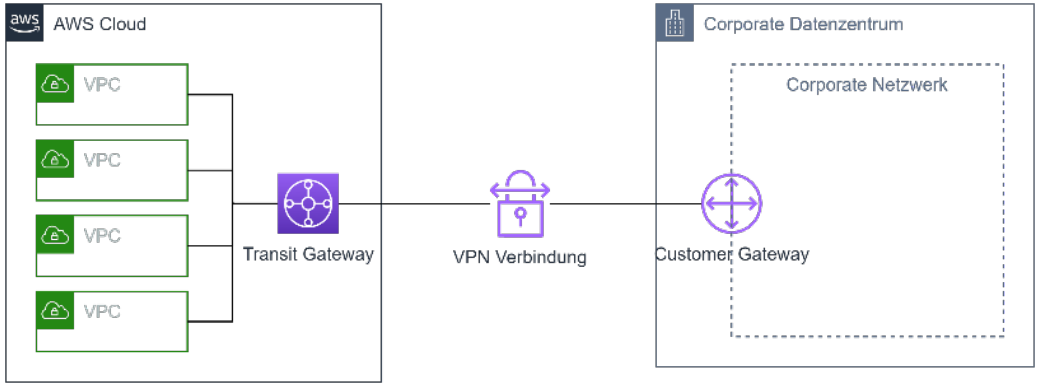
Die erste davon ist eine VPN Verbindung, welche ein sogenanntes **Site-to-Site-VPN** definiert. Dafür wird auf Seite des Kundennetzwerkes eine **Customer Gateway** benötigt. Auf AWS Seite je nach Anzahl der zu verbindenden Netzwerke und die Kardinalität dieser (1-to-1, 1-to-N, M-to-1, M-to-N) ein **Virtual Private Gateway** oder ein **Transit Gateway**.

Schutz von Ressourcen innerhalb eines Netzwerks auf AWS

Beispiel einer 1-to-1 VPN Verbindung



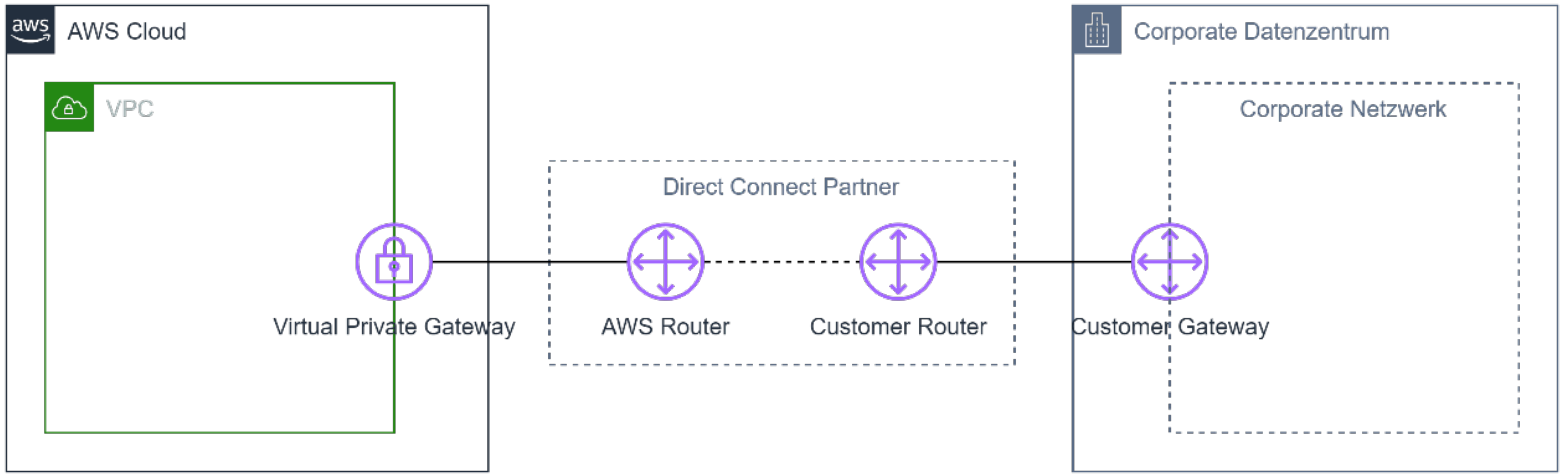
Beispiel einer M-to-1 VPN Verbindung



Schutz von Ressourcen innerhalb eines Netzwerks auf AWS

Sollten die Anforderungen über eine VPN Verbindung hinausgehen, kann auch ein **Direct Connect** als Verbindung zwischen dem on-premises und AWS Netzwerk verwendet werden.

Dabei können die beiden Netzwerke über einen Direct Connect Partner mit einer dedizierten Glasfaserleitung verbunden werden. Somit wird im Gegensatz zur VPN Verbindung das Internet vermieden. Diese Direct Connect Partner sind meist große Telekommunikationsanbieter in den einzelnen Ländern.



Schutz von Ressourcen innerhalb eines Netzwerks auf AWS

Der Netzwerkverkehr innerhalb eines AWS VPC kann dann mit verschiedenen Layern abgesichert werden, wie zum Beispiel **Security Groups** oder **NACLs** (Network Access Control List).

Beide Lösungen bieten die Möglichkeit einer **Firewall** im VPC, welche den Traffic zu den Ressourcen filtert, und beide können miteinander kombiniert werden.

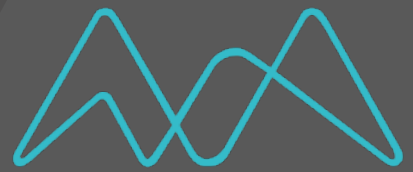
Type	Source	Protocol	Port Range	Description
HTTP	0.0.0.0/0	TCP	80	Erlaube eingehende HTTP Verbindungen von überall
HTTPS	0.0.0.0/0	TCP	443	Erlaube eingehende HTTPS Verbindungen von überall

Beispiel Inbound Regeln einer Security Group

Beispiel Inbound Regeln eines NACL

Rule #	Type	Source	Protocol	Port Range	Allow/Deny
18	HTTP	0.0.0.0/0	TCP	80	Allow
19	HTTPS	0.0.0.0/0	TCP	443	Allow
*	All traffic	0.0.0.0/0	All	All	Deny

Schutz von Daten in AWS S3 mit Hilfe von Amazon Macie



Schutz von Daten mit Amazon Macie

- Mit Amazon Macie können vertrauliche Daten in Amazon S3 Buckets erkannt werden
- Dabei wird Machine Learning und Musterabgleich verwendet
- Es existieren vordefinierte Identifier, wie z.B. Credentials, Bankdaten oder personenbezogene Daten, aber es können auch eigene Identifier definiert werden

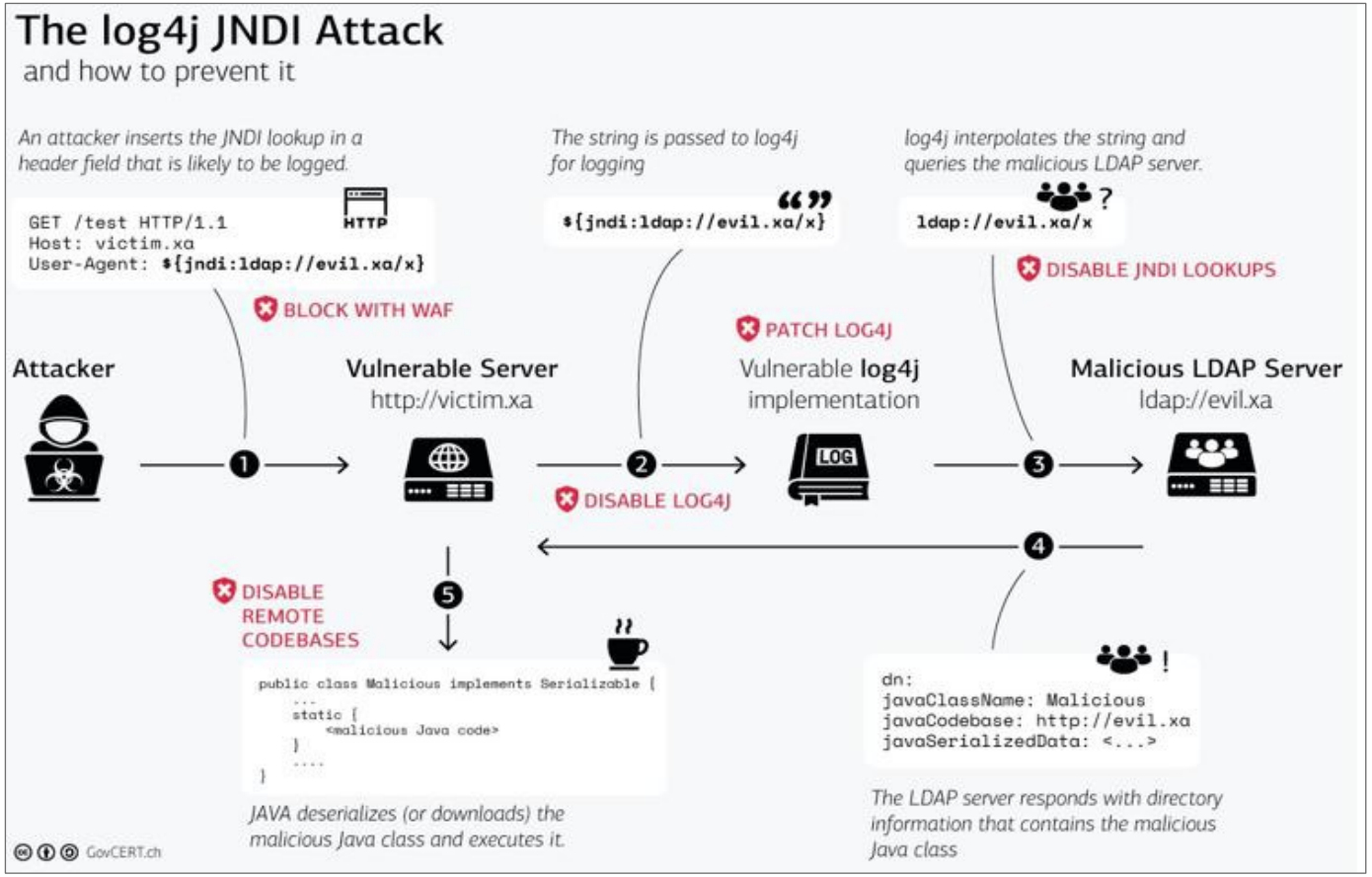
Quelle: https://aws.amazon.com/macie

Schutz von Daten mit Amazon Macie

Demo

AWS-Sicherheitsdienste zum Schutz, zur Erkennung und zur Reaktion auf die Log4j-Schwachstelle

Verwendung von AWS-Sicherheitsdiensten zum Schutz vor, zur Erkennung und zur Reaktion auf die Log4j-Schwachstelle (CVE-2021-44228)



Quelle: https://aws.amazon.com/de/blogs/security/using-aws-security-services-to-protect-against-detect-and-respond-to-the-log4j-vulnerability/

AWS WAF

- Mit der **AWS Web Application Firewall (WAF)** können HTTP(S)-Anfragen an folgende AWS-Dienste überwacht bzw. blockiert werden: Amazon CloudFront, Amazon API Gateway REST API, Application Load Balancer, AWS AppSync GraphQL API
- Für die AWS WAF stehen **von AWS verwaltete Regeln** bereit, die Schutz vor bekannten Anwendungsschwachstellen oder anderem unerwünschten Datenverkehr bieten
- So stellt z.B. der Regelsatz `AWSManagedRulesKnownBadInputsRuleSet` die Regel `Log4JRCE` bereit, mit deren Hilfe HTTP(S)-Anfragen auf das Vorhandensein der Log4j-Schwachstelle untersucht werden können; Beispielmuster beinhalten `${jndi:ldap://example.com/}`

➔ Verdächtige HTTP(S)-Anfragen entsprechend blockieren

Quelle: https://docs.aws.amazon.com/waf/latest/developerguide/waf-chapter.html

Amazon Route 53

Amazon Route 53

- **Amazon Route 53** ist ein hochverfügbarer und skalierbarer DNS-Dienst
- Mit Hilfe der **Route 53 Resolver DNS Firewall** kann der ausgehende DNS-Verkehr eines logisch isolierten virtuellen Netzwerks (Amazon VPC) gefiltert und kontrolliert werden
- Es stehen **von AWS verwaltete Listen von Domain-Namen** bereit, die mit bösartigen Aktivitäten oder anderen potentiellen Bedrohungen in Verbindung gebracht werden; z.B. wurde (und wird weiterhin) die `AWSManagedDomainsMalwareDomainList` mit Domain-Namen aktualisiert, die als Host für Malware identifiziert wurden, die in Verbindung mit der Log4j-Schwachstelle verwendet wird

➔ Verdächtige Domains entsprechend blockieren

\+ Verwendung externer, nicht vertrauenswürdiger DNS-Server verhindern (Route 53 Resolver DNS Firewall + GuardDuty)

\+ Protokollierung von DNS-Anfragen (Route 53 Resolver)

Quelle: https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resolver-dns-firewall-managed-domain-lists.html

Amazon Inspector

- **Amazon Inspector** ist ein automatisierter und skalierbarer Dienst zur kontinuierlichen Erkennung von Software-Schwachstellen
- Mit Hilfe des Dienstes können Amazon EC2-Instanzen sowie Container-Images in Amazon ECR auf bekannte Schwachstellen geprüft werden
- Sobald AWS dem Dienst eine Unterstützung zur Erkennung der Log4j-Schwachstelle hinzugefügt hatte, wurde von Amazon Inspector geprüft, ob Log4j
  - Auf (vom AWS Systems Manager verwalteten) EC2-Instanzen über den Paketmanager des Betriebssystems installiert ist
  - Als Maven-Dependency in einem Container-Image in ECR vorhanden ist

Quelle: https://aws.amazon.com/inspector

Amazon GuardDuty

- **Amazon GuardDuty** ist ein Dienst zur Bedrohungserkennung, welcher AWS-Konten/-Workloads kontinuierlich auf bösartige Aktivitäten überwacht
- Der intelligente Dienst nutzt maschinelles Lernen, Anomalie-Erkennung und von Drittanbietern bereitgestellte Bedrohungsdaten, um potentielle Bedrohungen zu identifizieren und zu priorisieren
- GuardDuty liefert detaillierte Informationen zu verdächtigen Aktivitäten, deren Ursache z.B. durch die vorhandene Integration mit **Amazon Detective** schnell analysiert werden kann
- Das Team hinter GuardDuty hat dem Dienst Indikatoren für eine Kompromittierung im Zusammenhang mit der Ausnutzung der Log4j-Schwachstelle hinzugefügt
- Hierzu überwacht der Dienst z.B. Versuche, verdächtige IP-Adressen oder DNS-Einträge zu erreichen, und kann durch Anomalie-basierte Verhaltensanalysen verdächtige Aktivitäten nach dem Ausnutzen der Schwachstelle feststellen

Quelle: https://aws.amazon.com/guardduty

Amazon CodeGuru

Amazon CodeGuru

Untrusted input causes log injection High

User-provided inputs must be sanitized before they are logged. An attacker can use unsanitized input to break a log's integrity, forge log entries, or bypass log monitors.

Detector ID
python/log-injection@v1.0

Common Weakness Enumeration (CWE)
CWE-117

Category
Security

Tags
data-integrity # injection

Noncompliant example

```
1  def logging_noncompliant():
2      filename = input("Enter a filename: ")
3      # Noncompliant: unsanitized input is logged.
4      logger.info("Processing %s", filename)
5
```

Compliant example

```
1  def logging_compliant():
2      filename = input("Enter a filename: ")
3      if re.match(r'^[\w_ -\.]+$', filename):
4          # Compliant: input is validated before logging.
5          logger.info("Processing %s", filename)
6
```

Quelle: https://aws.amazon.com/about-aws/whats-new/2022/02/amazon-codeguru-reviewer-detects-apache-log4j

AWS Systems Manager: Patch Manager

- **Patch Manager** ist eine Funktionalität des **AWS Systems Manager**-Dienstes, um den Patch-Prozess vom Dienst verwalteter Knoten (z.B. EC2-Instanzen) mit sicherheitsrelevanten und anderen Updates zu automatisieren
- Falls im Patch Manager kritische Patches zur sofortigen Installation eingestellt sind, verfügen vom Systems Manager-Dienst verwaltete EC2-Instanzen bereits über einen Patch für das (über den Paketmanager des Betriebssystems installierte) Log4j-Paket
- Für EKS-Cluster steht ein von AWS entwickelter Hot-Patch auf JVM-Ebene zur Verfügung, der JNDI-Lookups für von der Schwachstelle betroffene Versionen der Log4j-Bibliothek deaktiviert!
  Die betroffenen Container-Images können dann in ECR entsprechend aktualisiert werden, sodass sie eine gepatchte Log4j-Version verwenden …

Quelle: https://aws.amazon.com/systems-manager

Fragen?

CLOUD INSIGHTS

Kommende Online-Seminare

#digitalisierenmitmehrwert

23.03.2022 I Matthias Meyer

Hochskalierbare Cloud-Architekturen am Beispiel von Cassandra

---

06.04.2022 I Victor Ionescu & Walter Knaub

Azure DevOps services: From Dev to DevOps – Let the journey begin

---

01.06.2022 I Alexander Bätz

Effiziente Unternehmenssteuerung und bessere Marktchancen durch den Einsatz von SAP Analytics Tools

Registrierungslink:

www.msg.group/cloud#events

Kontakt

Philipp Dühring
Senior IT Consultant
Seit 2016 bei der msg
philipp.duehring@msg.group

Stefan Kurz
Lead IT Consultant
Seit 2008 bei der msg
stefan.kurz@msg.group

msg systems ag
Robert-Bürkle-Straße 1
85737 Ismaning

+49 89 96101-0

value – inspired by people