Milestone 1: Infrastruktur-Spezifikation

Hammerschmidt, Rentenberger, Schodl, Weidinger

28. November 2024

Inhaltsverzeichnis

1 Netzwerktopologie

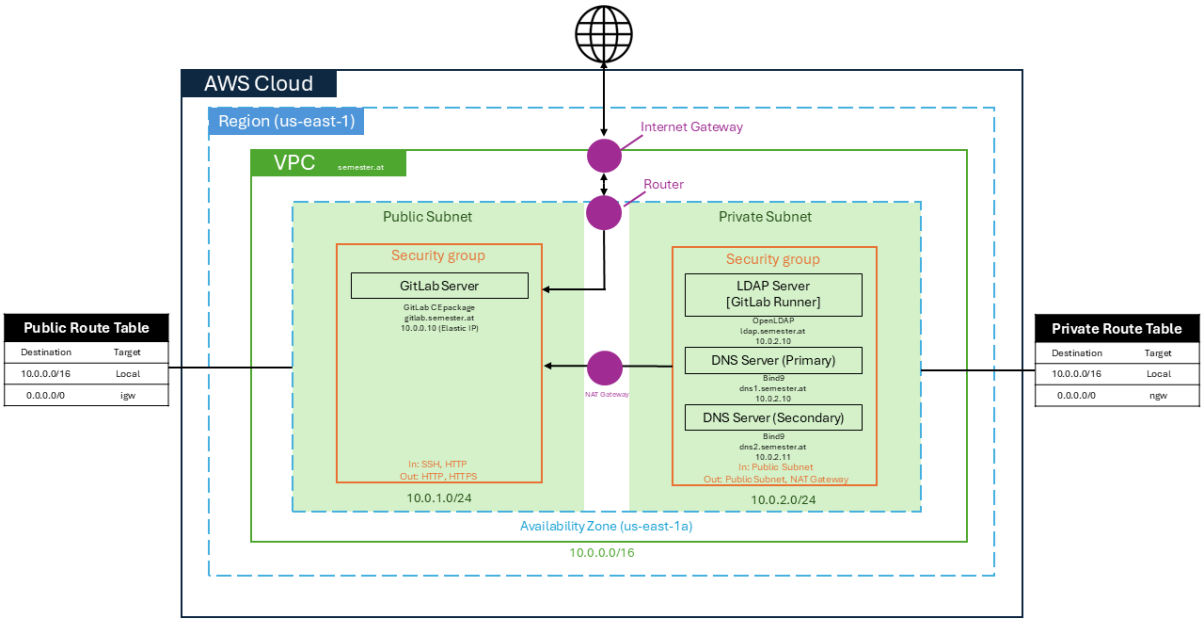

Abbildung 1: Netzwerktopologie der Infrastruktur

Dienst	**Subnetztyp**	**IP-Adresse**	**FQDN**
Primärer DNS	Privates Subnetz	10.0.1.10	dns1.semester.at
Sekundärer DNS	Privates Subnetz	10.0.1.11	dns2.semester.at
GitLab Server	Öffentliches Subnetz	10.0.0.10	gitlab.semester.at
LDAP Server	Privates Subnetz	10.0.2.10	server-ldap.semester.at

Tabelle 1: Netzwerkdienste und IP-Zuordnung

2 Geplante Security Groups und Regeln

Inbound SGRs	**DNS**	**Bastion**	**LDAP/GitLab-R**	**GitLab-Server**
HTTP	Nein	Nein	Nein	Ja
HTTPS	Nein	Nein	Ja	Ja
SSH	Ja	Ja	Ja	Ja
DNS (UDP)	Ja	Nein	Nein	Nein
DNS (TCP)	Ja	Nein	Nein	Nein
LDAP	Nein	Nein	Ja	Ja
ALL ICMP	Ja	Nein	Nein	Ja

Tabelle 2: Eingehende Sicherheitsgruppenregeln (SGRs) für verschiedene Dienste

Outbound SGRs	**DNS**	**Bastion**	**LDAP/GitLab-R**	**GitLab-Server**
HTTP	Nein	Nein	Ja	Nein
HTTPS	Nein	Nein	Ja	Nein
SSH	Nein	Nein	Ja	Nein
DNS (UDP)	Nein	Nein	Ja	Nein
DNS (TCP)	Nein	Nein	Ja	Nein
LDAP	Nein	Nein	Ja	Ja
ALL ICMP	Nein	Nein	Ja	Nein
ALL Traffic	Ja	Ja	Nein	Ja

Tabelle 3: Ausgehende Sicherheitsgruppenregeln (SGRs) für verschiedene Dienste

3 Spezifikationen der eingesetzten Systeme

Server	OS	Packages	Version	Server Instance
Primärer DNS Server	Ubuntu Server	bind9, bind9utils	BIND 9 / Ubuntu 24.04 LTS	T3.micro
Sekundärer DNS Server	Ubuntu Server	bind9, bind9utils	BIND 9 / Ubuntu 24.04 LTS	T3.micro
LDAP Server	Ubuntu Server	slapd, ldap-utils	OpenLDAP 2.6	T3.micro
GitLab Runner	Ubuntu Server	-	Ubuntu 24.04 LTS	T3.micro
GitLab Server	Ubuntu Server	GitLab CE	GitLab CE / Ubuntu 24.04 LTS	T2.Large

Tabelle 4: Server-Spezifikationen: Betriebssystem, Pakete und Instanztypen

- **Betriebssystem**: Ubuntu 24.04 LTS LTS (64-bit)
- **DNS-Server**: BIND 9.x
- **GitLab**: GitLab CE 15.x
- **GitLab Runner**: Version kompatibel mit GitLab CE 15.x
- **LDAP**: OpenLDAP 2.6.x
- **Optionale Überwachung**: AWS CloudWatch zur Protokollierung und Überwachung.

4 Tests

DNS Resolution Testing

- **Ziel:** Sicherstellen, dass der BIND-Server Domain-Namen korrekt auflöst.
- **Methode:** Verwenden des `dig`-Befehls, um den DNS-Server nach bekannten Domains abzufragen. Überprüfen der `A`, `AAAA`, `MX` und `NS` Records:

```
dig @<DNS-server> example.com [A, AAAA, MX, NS]
```

- **Erwartetes Ergebnis:** Jede Abfrage liefert die richtigen IP-Adressen und Record-Details.

Forward and Reverse DNS Lookup

- **Ziel:** Überprüfen, dass Vorwärts- und Rückwärts-Abfragen funktionieren.
- **Methode:**
  - Verwenden von `dig` für die Vorwärtsabfrage (Domain zu IP):

```
dig @<DNS-server> example.com A
```

  - Verwenden von `dig -x` für die Rückwärtsabfrage (IP zu Domain):

```
dig @<DNS-server> -x [192.0.2.1]
```

- **Erwartetes Ergebnis:** Genaues Mapping zwischen Domain-Namen und IP-Adressen.

Zone Transfer Test

- **Ziel:** Sicherstellen, dass Zonentransfers zwischen primären und sekundären DNS-Servern funktionieren.
- **Methode:** Einen Zonentransfer mit `dig AXFR` anstoßen und die Logs auf den Transfer überprüfen:

```
dig @<primary-DNS-server> example.com AXFR
```

- **Erwartetes Ergebnis:** Zonendaten werden korrekt zwischen primären und sekundären Servern repliziert.

DNS Failover Testing

- **Ziel:** Die Resilienz und Zuverlässigkeit des DNS-Dienstes unter Ausfallbedingungen bewerten.
- **Methode:**
  - Einen Ausfall des primären DNS-Servers simulieren.
  - Die Antwort des sekundären DNS-Servers überwachen:

```
dig @<secondary-DNS-server> example.com A
```

  - Etwaige Ausfallzeiten während des Übergangs protokollieren.
- **Erwartetes Ergebnis:** Der sekundäre DNS-Server übernimmt nahtlos mit wenig bis gar keiner Unterbrechung der DNS-Auflösung.

Stress Testing

- **Ziel:** Die Leistung des Servers unter hoher Last testen.
- **Methode:** Tools wie `dnsperf` verwenden, um eine hohe Anzahl von DNS-Abfragen zu simulieren:

```
dnsperf -s <primary-DNS-IP> -d queries.txt -l 30
```

- **Erwartetes Ergebnis:** Der Server bleibt auch unter Last genau und leistungsfähig.

Rolle	Verantwortlichkeiten
Netzwerk-Architekt	Planung und Einrichtung der VPC und Subnetze
DevOps-Ingenieur	Bereitstellung von GitLab und CI/CD
Systemadministrator	Konfiguration von DNS- und LDAP-Servern
QA-Ingenieur	Testen der Dienste und Sicherheitskonfiguration

Tabelle 5: Rollen und Verantwortlichkeiten

5 Rollen und Verantwortlichkeiten im Team

6 Monitoring

AWS CloudWatch wird zur Protokollierung und Überwachung genutzt:

- Überwachung der CPU-, Speicher- und Netzwerknutzung.
- Automatische Alarme bei Ausfällen.