

# Milestone 1: Infrastruktur-Spezifikation

Hammerschmidt, Rentenberger, Schodl, Weidinger

28. November 2024

## Inhaltsverzeichnis

<b>1</b>	<b>Netzwerktopologie</b>	<b>2</b>
<b>2</b>	<b>Geplante Security Groups und Regeln</b>	<b>3</b>
<b>3</b>	<b>Spezifikationen der eingesetzten Systeme</b>	<b>4</b>
<b>4</b>	<b>Tests</b>	<b>5</b>
<b>5</b>	<b>Rollen und Verantwortlichkeiten im Team</b>	<b>8</b>
<b>6</b>	<b>Monitoring</b>	<b>8</b>
	<b>Abbildungsverzeichnis</b>	<b>9</b>
	<b>Tabellenverzeichnis</b>	<b>10</b>

# 1 Netzwerktopologie

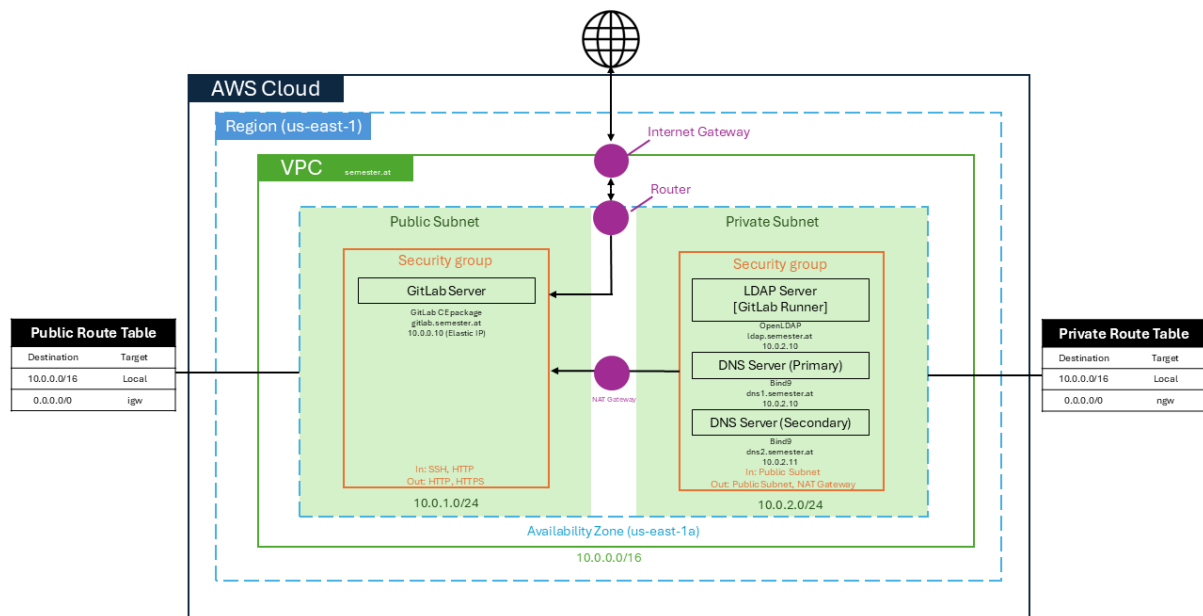


Abbildung 1: Netzwerktopologie der Infrastruktur

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

<b>Inbound SGRs</b>	<b>DNS</b>	<b>Bastion</b>	<b>LDAP/GitLab-R</b>	<b>GitLab-Server</b>
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

<b>Outbound SGRs</b>	<b>DNS</b>	<b>Bastion</b>	<b>LDAP/GitLab-R</b>	<b>GitLab-Server</b>
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
- **Monitoring:** 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 `dnstperf` verwenden, um eine hohe Anzahl von DNS-Abfragen zu simulieren:

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

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

### subsection\*LDAP-Authentifizierungstest

- **Ziel:** Funktionalität des Authentifizierungssystems überprüfen.
- **Methode:** Benutzeranmeldungen über LDAP versuchen. Tests mit gültigen und ungültigen Anmeldedaten durchführen.
- **Erwartetes Ergebnis:** Anmeldeversuche sollten für gültige Anmeldedaten akzeptiert und für ungültige Anmeldedaten abgelehnt werden.

## LDAP-Suche und -Filterung

- **Ziel:** Genauigkeit der LDAP-Suche und -Filterung bestätigen.
- **Methode:** Suchen nach bestimmten Benutzergruppen oder Attributen durchführen und Filter testen.
- **Erwartetes Ergebnis:** Für jede Suche und jeden Filter werden die korrekten Daten zurückgegeben.

## Benutzer- und Gruppenverwaltung

- **Ziel:** Testen der Erstellung und Änderung von Benutzern und Gruppen.
- **Methode:** Gruppen und Benutzer erstellen sowie deren Attribute ändern. Änderungen im LDAP-Verzeichnis überwachen.
- **Erwartetes Ergebnis:** Alle Änderungen werden korrekt im LDAP-Verzeichnis angezeigt.

## LDAP-Integrationstest

- **Ziel:** Testen, ob die Integration der GitLab-Authentifizierung mit LDAP funktioniert.
- **Methode:** Anmeldungen bei GitLab mit LDAP-Anmeldedaten (verschiedene Rollen) durchführen.
- **Erwartetes Ergebnis:** Benutzeranmeldungen werden mit LDAP-Anmeldedaten akzeptiert.

## Repository-Operationen

- **Ziel:** Testen, ob Standard-Git-Operationen innerhalb von GitLab funktionieren.
- **Methode:** Neben der Erstellung von Repositories werden Pull-, Push- und Merge-Operationen getestet. Zusätzlich werden Branches erstellt und gelöscht.
- **Erwartetes Ergebnis:** Änderungen entsprechen den erwarteten Ergebnissen der Git-Operationen.

## CI/CD-Pipeline-Test

- **Ziel:** Funktionalität der CI/CD-Pipeline überprüfen.
- **Methode:** Geänderten Code pushen und die automatische Ausführung der CI/CD-Pipeline beobachten. Erfolg des Builds und der Bereitstellung prüfen.
- **Erwartetes Ergebnis:** Codeänderungen werden automatisch gebaut und fehlerfrei bereitgestellt.

## Lasttest

- **Ziel:** Leistung von GitLab unter hoher Last bewerten.
- **Methode:** Mehrere Benutzer simulieren, die gleichzeitig Git-Operationen durchführen.
- **Erwartetes Ergebnis:** GitLab hält die Leistungsniveaus auch bei gleichzeitiger Nutzung aufrecht.

## 5 Rollen und Verantwortlichkeiten im Team

<b>Samuel Hammerschmidt</b>	<b>Lorenz Rentenberger</b>	<b>Nikolas Schodl</b>	<b>Alexander Weidinger</b>
LDAP Server	DNS Server (CloudWatch)	GitLab Server	GitLab Server
AWS Cloud Config	AWS Cloud Config	AWS Cloud Config	AWS Cloud Config
Tests LDAP	Tests DNS	Tests GitLab	Tests GitLab

Tabelle 5: Team-Aufgaben und Zuständigkeiten

## 6 Monitoring

AWS CloudWatch wird zur Protokollierung und Überwachung genutzt:

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



# Abbildungsverzeichnis

1	Netzwerktopologie der Infrastruktur . . . . .	2
---	---	---

## Tabellenverzeichnis

1	Netzwerkdienste und IP-Zuordnung . . . . .	2
2	Eingehende Sicherheitsgruppenregeln (SGRs) für verschiedene Dienste . .	3
3	Ausgehende Sicherheitsgruppenregeln (SGRs) für verschiedene Dienste .	3
4	Server-Spezifikationen: Betriebssystem, Pakete und Instanztypen . . . . .	4
5	Team-Aufgaben und Zuständigkeiten . . . . .	8