# Milestone 1: Infrastruktur-Spezifikation

Hammerschmidt, Rentenberger, Schodl, Weidinger

28. November 2024

## Inhaltsverzeichnis

# 1 Netzwerktopologie

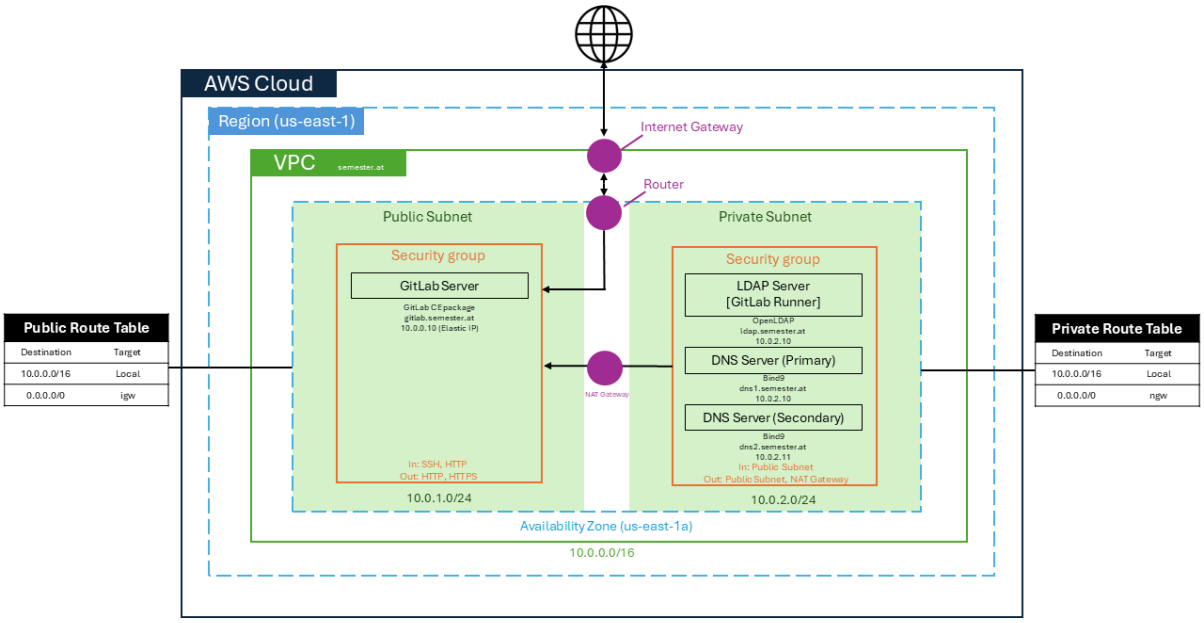


Abbildung 1: Netzwerktopologie der Infrastruktur

| **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

Die Security Groups definieren, welche Dienste und Ports erreichbar sind:

| **Dienst** | **Eingehend (Ingress)** | **Ausgehend (Egress)** |
|---|---|---|
| DNS | Port 53 (UDP/TCP) von privatem Subnetz | Alle Ports ins VPC-Netz |
| GitLab | Port 80/443 (HTTP/HTTPS) von bestimmten IPs | Alle Ports ins VPC-Netz |
| GitLab Runner | Port 8093 von GitLab-Server | Alle Ports ins VPC-Netz |
| LDAP | Port 389 (LDAP) nur aus privatem Subnetz | Alle Ports ins VPC-Netz |

Tabelle 2: Geplante Security Groups und Regeln

| Inbound SGRs | DNS | Bastion | LDAP/GitLab-R | GitLab-Server |
|---|---|---|---|---|
| HTTP | Nein | Nein | Nein | Ja (0.0.0.0/0) |
| HTTPS | Nein | Nein | Ja (10.0.0.0/24) | Ja (0.0.0.0/0) |
| SSH | Ja (10.0.0.0/24) | Ja (0.0.0.0/0) | Ja (0.0.0.0/0) | Ja (0.0.0.0/0) |
| DNS (UDP) | Ja (10.0.0.0/24) | Nein | Nein | Nein |
| DNS (TCP) | Ja (10.0.0.0/24) | Nein | Nein | Nein |
| LDAP | Nein | Nein | Ja (0.0.0.0/0) | Ja (0.0.0.0/0) |
| ALL ICMP | Ja (0.0.0.0/0) | Nein | Nein | Ja (10.0.0.0/24) |

Tabelle 3: Eingehende Sicherheitsgruppenregeln (SGRs) für verschiedene Dienste

| Outbound SGRs | DNS | Bastion | LDAP/GitLab-R | GitLab-Server |
|---|---|---|---|---|
| HTTP | Nein | Nein | Ja (0.0.0.0/0) | Nein |
| HTTPS | Nein | Nein | Ja (0.0.0.0/0) | Nein |
| SSH | Nein | Nein | Ja (0.0.0.0/0) | Nein |
| DNS (UDP) | Nein | Nein | Ja (0.0.0.0/0) | Nein |
| DNS (TCP) | Nein | Nein | Ja (0.0.0.0/0) | Nein |
| LDAP | Nein | Nein | Ja (0.0.0.0/0) | Ja (0.0.0.0/0) |
| ALL ICMP | Nein | Nein | Ja (0.0.0.0/0) | Nein |
| ALL Traffic | Ja (0.0.0.0/0) | Ja (0.0.0.0/0) | Nein | Ja (0.0.0.0/0) |

Tabelle 4: Ausgehende Sicherheitsgruppenregeln (SGRs) für verschiedene Dienste

| Serverinstanz | Betriebssystem | Dienste |
|---|---|---|
| DNS-Server 1 | Ubuntu 22.04 LTS | BIND (Primärer DNS) |
| DNS-Server 2 | Ubuntu 22.04 LTS | BIND (Sekundärer DNS) |
| GitLab Server | Ubuntu 22.04 LTS | GitLab CE |
| GitLab Runner | Ubuntu 22.04 LTS | GitLab Runner |
| LDAP Server | Ubuntu 22.04 LTS | OpenLDAP |

Tabelle 5: Dienste-Zuordnung zu Server-Instanzen

# 3 Dienste-Zuordnung zu Server-Instanzen

# 4 Spezifikationen der eingesetzten Systeme

| Server | OS | Packages | Version |
|---|---|---|---|
| Preferred DNS Server | Ubuntu Server | bind9, bind9utils | BIND 9.18.19 / Ubuntu 22.04.3 LTS |
| Alternative DNS Server | Ubuntu Server | bind9, bind9utils | BIND 9.18.19 / Ubuntu 22.04.3 LTS |
| LDAP Server | Ubuntu Server | slapd, ldap-utils | OpenLDAP 2.6.6 |
| GitLab Runner | Ubuntu Server | - | Ubuntu 22.04.3 LTS |
| GitLab Server | Ubuntu Server | GitLab CE | GitLab CE / Ubuntu 22.04.3 LTS |

Tabelle 6: Server-Spezifikationen: Betriebssystem, Pakete und Instanztypen

- **Betriebssystem**: Ubuntu 22.04 LTS (64-bit)
- **DNS-Server**: BIND 9.18

- **GitLab**: GitLab CE 15.x
- **GitLab Runner**: Version kompatibel mit GitLab CE 15.x
- **LDAP**: OpenLDAP 2.5.x
- **Optionale Überwachung**: AWS CloudWatch zur Protokollierung und Überwachung.

# 5 Tests

## DNS Resolution Testing

- **Objective:** Ensure the BIND server correctly resolves domain names.
- **Method:** Use the `dig` command to query the DNS server for known domains. Check `A`, `AAAA`, `MX`, and `NS` records:

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

- **Expected Outcome:** Each query returns the correct IP addresses and record details.

## Forward and Reverse DNS Lookup

- **Objective:** Verify that forward and reverse lookups are functional.
- **Method:**
  - Use `dig` for forward lookup (domain to IP):

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

  - Use `dig -x` for reverse lookup (IP to domain):

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

- **Expected Outcome:** Accurate mapping between domain names and IP addresses.

## Zone Transfer Test

- **Objective:** Ensure zone transfers between primary and secondary DNS servers are working.
- **Method:** Initiate a zone transfer using `dig AXFR` and check logs for the transfer:

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

- **Expected Outcome:** Zone data is accurately replicated between primary and secondary servers.

### DNS Failover Testing

- **Objective:** Assess the resilience and reliability of the DNS service under failure conditions.
- **Method:**
  - Simulate a failure of the primary DNS server.
  - Monitor the response from the secondary DNS server:

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

  - Record any downtime experienced during the transition.
- **Expected Outcome:** The secondary DNS server seamlessly takes over with little to no disruption in DNS resolution.

## Stress Testing

- **Objective:** Test the performance of the server under high load.
- **Method:** Use tools such as `dnsperf` to simulate a high number of DNS queries:

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

- **Expected Outcome:** The server maintains accuracy and performs well under the load.

# 6 Rollen und Verantwortlichkeiten im Team

| **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 7: Rollen und Verantwortlichkeiten

# 7 FQDN und interne Domain

Die interne Domain wird als **intern.local** festgelegt. Beispiele:

- DNS-Server: `dns1.intern.local`, `dns2.intern.local`.
- GitLab: `gitlab.intern.local`.

## Optional: Monitoring

AWS CloudWatch wird zur Protokollierung und Überwachung genutzt:

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