

| Auftrag 0002

- [Der Auftrag](#)
- [DNS Server installieren](#)
- [Konfiguration von Bind](#)
- [DNS Zonen erstellen](#)
- [Logging](#)

Der Auftrag

Die Firma **HighEnd Elektro AG** möchte einen eigenen DNS-Resolver für das Unternehmen verwenden, damit man diesbezüglich nicht von einem anderen Unternehmen abhängig ist. Das Unternehmen hat bereits vor ein paar Jahren die gesamte Server-Infrastruktur in die Cloud ausgelagert und verfügt über keine lokale Server-Infrastruktur mehr. Aus diesem Grund soll die Installation und der Betrieb komplett von Ihrer Firma bereitgestellt werden. Die HighEnd Elektro AG stellt konkrete Anforderungen, welche der DNS-Resolver erfüllen muss:

- Der DNS-Resolver muss grundsätzlich von überall her für die Mitarbeiter aus dem Internet erreichbar sein.
- Der DNS-Resolver darf nicht als öffentlicher DNS-Resolver fungieren, aus diesem Grund darf der DNS-Resolver nur Anfragen aus folgenden Netzwerken beantworten
 - 31.10.147.0/24
 - 80.74.144.0/24
 - Ihr eigenes Netz bez. IP-Adresse von dem Sie arbeiten.
- Der DNS-Resolver soll neben der reinen Resolving-Funktion auch noch weitere Aufgaben übernehmen. So hat das Unternehmen eine Intranet-Webseite mit der IP 80.74.136.2. Diese soll über den Domain-Namen **high-end.intern** erreichbar sein, wenn der DNS-Resolver verwendet wird.
- Das Unternehmen möchte, dass gewisse Domains aus dem Unternehmen heraus nicht aufgerufen werden können und daher vom Resolver blockiert werden. Es handelt sich hierbei um folgende Domains:
 - facebook.com (inkl. www)
 - youtube.com (inkl. www)
 - tiktok.com (inkl. www)
 - instagram.com (inkl. www)

Um die Wartbarkeit des Systems zu gewährleisten, wird eine vollständige technische Dokumentation erwartet. Diese beinhaltet eine Dokumentation des Installations-Vorgangs, die relevanten Konfigurationen sowie ein Test-Protokoll, welches die Funktionstüchtigkeit des Systems bei der Übergabe belegt.

Erwartete Abgabe

Folgende Abgaben werden erwartet:

- Vollständig installierter und getesteter DNS-Server
- Sämtliche relevanten Konfigurationen vom DNS-Server im Format .txt
- Technische Dokumentation gem. Vorlage als Word oder PDF:
 - [Vorlage_technische_Dokumentation_m123.docx](#)
 - [Anleitung zum Testen von Systemen und Objekten für Berufsschulen.docx](#)

Vorführung im Unterricht

In der Unterrichtswoche nach der Abgabe, wird jede Gruppe das System kurz präsentieren. Dabei wird getestet, ob die wichtigsten Funktionen erfüllt werden. Weiter wird jedes Mitglied der Gruppe ein bis zwei Fragen zur Umsetzung des Systems beantworten müssen.

Formalitäten

Bei voller Zufriedenheit der Arbeit erwartet Sie einen Lohn von CHF 9000.

Der Zeitaufwand für die Arbeit beträgt ungefähr 3 bis 4 Lektionen.

DNS Server installieren

In dieser Übung werden Sie einen DNS-Server installieren und konfigurieren. Hierzu werden Sie auf der AWS Academy Lab-Umgebung einen virtuellen Server (EC2 Instanz) starten. Auf diesem virtuellen Server werden Sie den weitverbreiteten Open-Source DNS Server [Bind](#) installieren.

Sicherheitsgruppe erstellen

Als erstes muss in AWS eine Sicherheitsgruppe erstellt werden. Mit dieser Sicherheitsgruppe wird die Firewall des Servers konfiguriert. Hierbei ist es wichtig, dass wir den eingehenden Datenverkehr für DNS-Pakete zulassen.

1. Starten Sie die [AWS Academy Lab-Umgebung](#).
2. Klicken Sie auf **Services** -> **Datenverarbeitung** um die EC2 Übersicht zu öffnen.
3. Klicken Sie in der Gruppe **Ressourcen** auf **Sicherheitsgruppen** um zur Übersicht mit den Sicherheitsgruppen zu gelangen.

Ressourcen Info Globale Ansicht von EC2 ↗ ↻

Sie verwenden die folgenden Amazon EC2-Ressourcen in der Region USA Ost (Nord-Virginia):

Instances (ausgeführt)	0	Dedicated Hosts	0	Elastic IP-Adressen
Instances	0	Load Balancers	0	Platzierungsgruppen
Schlüsselpaare	4	Sicherheitsgruppen	1	Snapshots
Volumes	0			

4. Klicken Sie auf den orangenen Button **Sicherheitsgruppe erstellen**.

- Wählen Sie einen passenden Namen wie z. B. DNS-Server. Diese Sicherheitsgruppe können Sie dann für alle DNS-Server verwenden, welche Sie in der AWS-Umgebung erstellen. Setzen Sie ebenfalls eine aussagekräftige Beschreibung wie "Gewährt den Zugriff fuer DNS-Dienste von extern" (Sonderzeichen sind nicht erlaubt)
- Klicken Sie nun unter **Regeln für eingehenden Datenverkehr** auf **Regel hinzufügen**.
- Fügen Sie nun drei Regeln hinzu. Jeweils eine für **DNS (TCP)**, **DNS (UDP)**, und **SSH**. Setzen Sie bei allen drei Regeln die erlaubte Quelle auf **Anywhere IPv4**:

Regeln für eingehenden Datenverkehr [Info](#)

Typ Info	Protokoll Info	Portbereich Info	Quelle Info	Beschreibung – optional Info
DNS (UDP)	UDP	53	Anywh... 0.0.0.0/0	
DNS (TCP)	TCP	53	Anywh... 0.0.0.0/0	
SSH	TCP	22	Anywh... 0.0.0.0/0	

[Regel hinzufügen](#)

- Bei Regeln für ausgehenden Datenverkehr sollte bereits eine Regel existieren für den gesamten Datenverkehr. Diese können Sie so belassen:

Regeln für ausgehenden Datenverkehr [Info](#)

Typ Info	Protokoll Info	Portbereich Info	Ziel Info	Beschreibung – optional Info
Gesamter Datenverkehr	Alle	Alle	Benutz... 0.0.0.0/0	

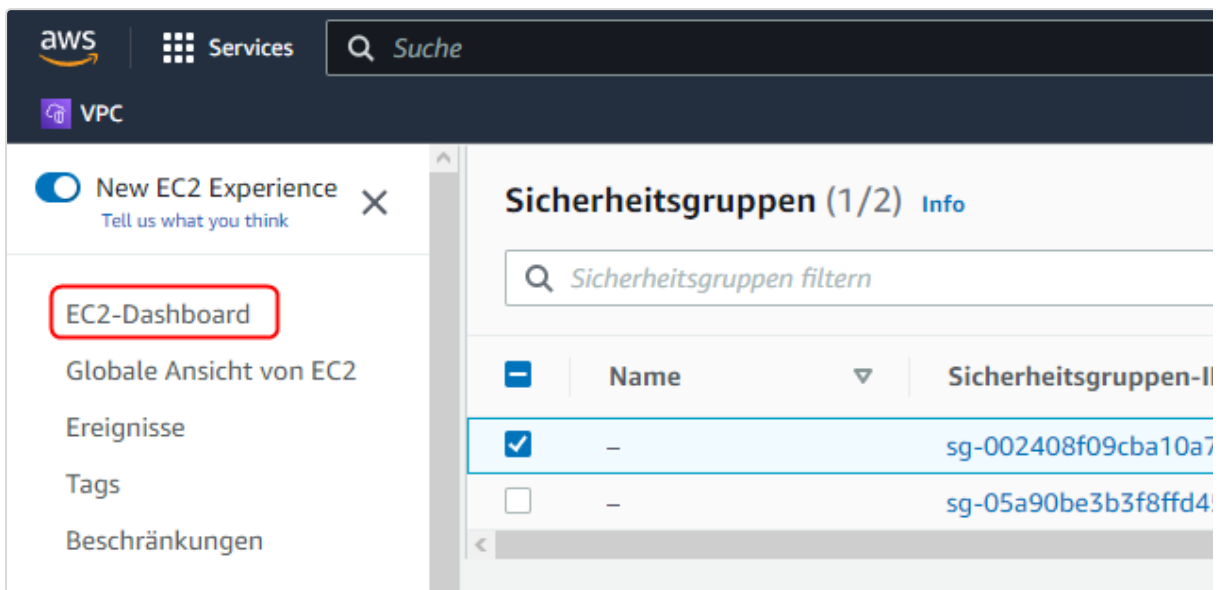
[Regel hinzufügen](#)

- Bestätigen Sie nun mit dem orangen Button **Sicherheitsgruppe erstellen**

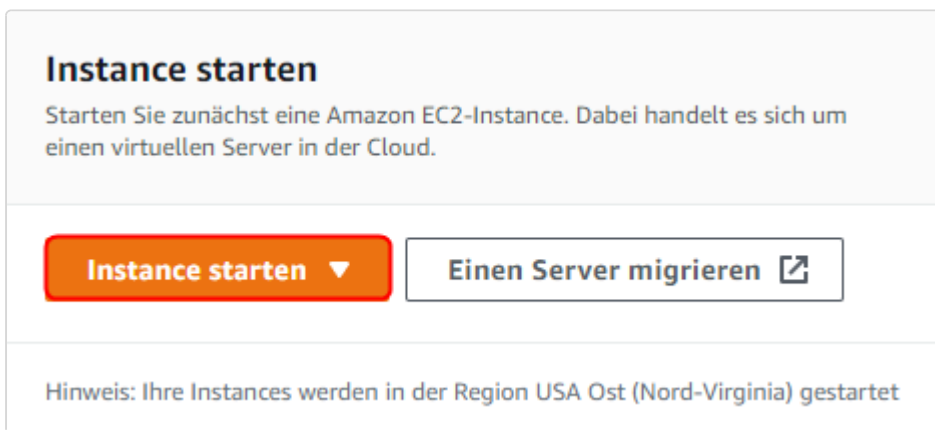
VM anlegen

Nun können Sie als nächsten Schritt eine neue Virtuelle Maschine (EC2 Instanz) anlegen:

- Wechseln Sie wieder in das EC2-Dashboard. Alternativ können Sie im Menü aber auch direkt zu Instances wechseln:



2. Klicken Sie nun auf den Button Instance starten:



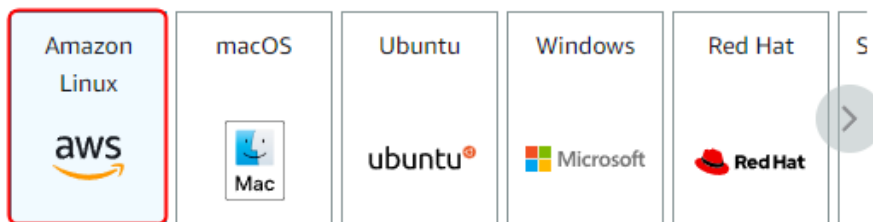
3. Wählen Sie einen passenden Namen für Ihren DNS Server.
4. Wählen Sie nun Amazon Linux als Server Image aus:

▼ Anwendungs- und Betriebssystemabbilder (Amazon Machine Image) [Info](#)

Ein AMI ist eine Vorlage, die die Softwarekonfiguration (Betriebssystem, Anwendungsserver und Anwendungen) enthält, die Starten Ihrer Instance benötigt wird. Suchen Sie nach AMIs, wenn Sie unten nicht sehen, wonach Sie suchen

 Durchsuchen Sie unseren vollständigen Katalog mit Tausenden von Anwendungs- und Betriebssystemab

Schnellstart




Weitere AMIs
durchsuchen

Einschließlich von AMI
aus AWS, Marketplace
und der Community

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type Zur kostenlosen Nutzung berechtigt
ami-0b0dcb5067f052a63 (64-Bit (x86)) / ami-01b5ec3ed8678d8b7 (64-Bit (Arm))
Virtualisierung: hvm ENA-aktiviert: true Root-Gerätetyp: ebs

Beschreibung

Amazon Linux 2 Kernel 5.10 AMI 2.0.20221103.3 x86_64 HVM gp2

Architektur

AMI-ID

64-Bit (x86) ▼

ami-0b0dcb5067f052a63

Verifizierter Anbieter

5. Wählen Sie den Instanz-Typ **t2.medium** aus.
6. Wählen Sie Ihren vorgängig erstellten Gruppen-SSH-Key aus, damit dieser bei der EC2-Instanz hinterlegt wird.

▼ Schlüsselpaar (Anmeldung) [Info](#)

Sie können ein Schlüsselpaar verwenden, um eine sichere Verbindung zu Ihrer Instance herzustellen. Stellen Sie sicher, dass Sie Zugriff auf das ausgewählte Schlüsselpaar haben, bevor Sie die Instance starten.

Schlüsselpaarname - Pflichtfeld

GruppenName ▼



[Neues Schlüsselpaar erstellen](#)

7. Nun können Sie im Abschnitt Netzwerk-Einstellungen die zuvor erstellte Sicherheitsgruppe hinzufügen.

▼ Netzwerkeinstellungen [Info](#)

[Bearbeiten](#)

Netzwerk [Info](#)

vpc-058c2581c8a78fc68

Subnetz [Info](#)

Keine Präferenz (Standard-Subnetz in jeder Availability Zone)

Öffentliche IP automatisch zuweisen [Info](#)

Aktivieren

Firewall (Sicherheitsgruppen) [Info](#)

Eine Sicherheitsgruppe ist eine Reihe von Firewall-Regeln, die den Datenverkehr für Ihre Instance steuern. Fügen Sie Regeln hinzu, damit bestimmter Datenverkehr Ihre Instance erreichen kann.

☐ Sicherheitsgruppe erstellen

☒ Vorhandene Sicherheitsgruppe auswählen

Sicherheitsgruppen [Info](#)

Sicherheitsgruppen auswählen ▲



☐ default
VPC: vpc-058c2581c8a78fc68

sg-002408f09cba10a78

☐ DNS-Server
VPC: vpc-058c2581c8a78fc68

sg-05a90be3b3f8ffd45



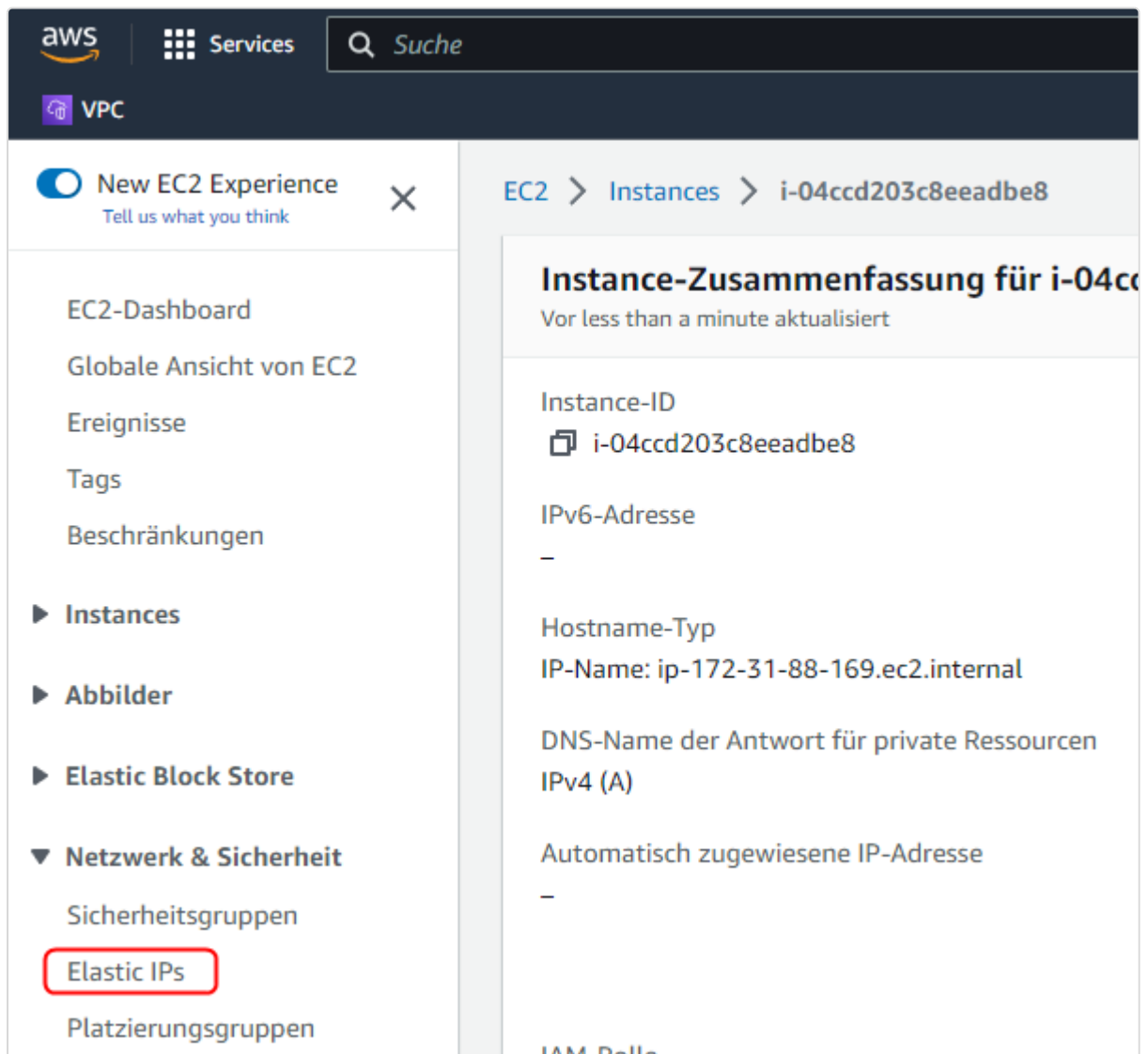
[Sicherheitsgruppen vergleichen](#)

8. Weisen Sie der EC2-Instanz eine statische IP-Adresse zu.

Statische IP-Adresse zuweisen

Standardmässig erhält eine VM keine statische IP-Adresse in AWS. Bei jedem Neustart einer Instanz wird dynamisch eine neue IP-Adresse zugewiesen. Dies ist bei einem Server ungünstig, da dieser immer unter der selben IP-Adresse erreichbar sein sollte. Statische IP-Adressen bei AWS sind eine kostenpflichtige Dienstleistung. Aus diesem Grund müssen Sie nun zu erst eine statische IP-Adresse erstellen und diese dann der erstellten VM zuweisen.

1. Navigieren Sie im Menü zum Punkt **Elastic IPs** in der Gruppe **Netzwerk & Sicherheit**



2. Klicken Sie auf den orangenen Button **Elastic IP-Adresse zuweisen**.
3. Die bereits gesetzten Einstellungen können Sie so belassen. Bestätigen Sie nun mit dem Button **Zuweisen**.
4. Selektieren Sie die neu erstellte IP-Adresse und wählen Sie unter **Aktionen** die Option **Elastic IP-Adresse zuordnen**:

Elastic IP-Adressen (1/1)					Aktionen ▲	Elastic IP-Adresse zuwe...
Elastic IP-Adressen filtern					Details einblenden	Elastic IP-Adressen freigeben
☒	Name ▼	Zugewiesene IPv4-...	Typ		Elastic IP-Adresse zuordnen	
☒	–	44.194.159.38	Öffentliche IP-Adresse		us-east-1	

5. Wählen Sie nun unter Instance die zuvor erstellte VM aus:

Elastic IP-Adresse: 44.194.159.38

Ressourcentyp
 Wählen Sie die Art der Ressource aus, der die Elastic IP-Adresse zugeordnet werden soll.

☒ Instance
 ☐ Netzwerkschnittstelle

⚠ Wenn Sie eine Elastic IP-Adresse einer Instance zuordnen, die bereits einer anderen Elastic IP-Adresse zugeordnet ist, wird die Zuordnung für die zuvor zugeordnete Elastic IP-Adresse aufgehoben. Sie bleibt Ihrem Konto jedoch weiterhin zugewiesen. [Weitere Informationen](#)

Instance

i-04ccd203c8eeadb8 (DNS-SERVER) - running

Private IP-Adresse
 Die private IP-Adresse, der die Elastic IP-Adresse zugeordnet werden soll.

Neuzuordnung
 Geben Sie an, ob die Elastic IP-Adresse einer neuen Ressource zugeordnet werden kann, wenn sie bereits einer anderen Ressource zugeordnet ist.

☐ Neuzuordnung dieser elastischen IP-Adresse zulassen

6. Bestätigen Sie die Aktion mit dem Button **Zuordnen**.

7. Wechseln Sie zurück zur Instanz. Nun sollte die öffentliche IP-Adresse zugeordnet sein:

Instance-Zusammenfassung für i-04ccd203c8eeadb8 (DNS-SERVER) Info	
Vor less than a minute aktualisiert	
Instance-ID i-04ccd203c8eeadb8 (DNS-SERVER)	Öffentliche IPv4-Adresse 44.194.159.38 offene Adresse
IPv6-Adresse –	Instance-Status Läuft

Verbindung mit einem SSH-Client

Es ist am bequemsten per SSH auf einem Server zu arbeiten. Dazu haben Sie ein lokales Terminal auf Ihrem Gerät, welches eine SSH-Verbindung zum Server aufbaut.

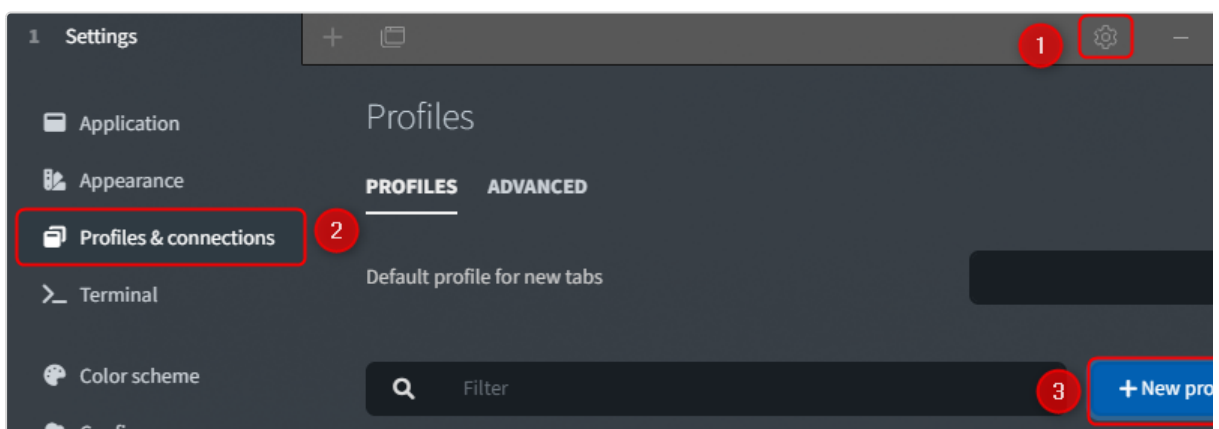
Das kryptographische Netzwerkprotokoll Secure Shell (SSH) ermöglicht es, Netzwerkdienste über unsichere Netzwerke sicher zu betreiben. Mit SSH kann man eine lokale Kommandozeile auf einen entfernten Rechner setzen, auf welchem dann die Ausgaben der entfernten Konsole angezeigt und lokale Tastatureingaben an den entfernten Rechner gesendet werden. Dadurch kann man z. B. einen Server, der in einem entfernten Rechenzentrum steht, fernwarten. Mit der neueren Protokollversion SSH-2 sind weitere Funktionen wie die Datenübertragung per SFTP verfügbar.

Sie dürfen natürlich jedes Terminal verwenden. Diese Anleitung zeigt die Konfiguration anhand vom Tabby Terminal. Es ist aber auch möglich die Verbindung mit Putty oder einem

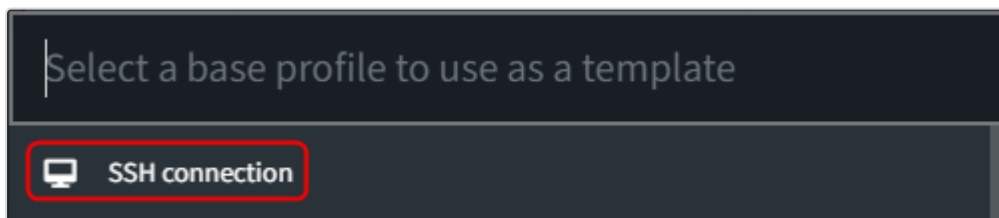
1. Laden Sie die Software Tabby Terminal von folgender Seite herunter und installieren Sie diese: <https://tabby.sh>

Laden Sie die Version **tabby-1.0.187-setup-x64.exe** herunter.

2. Starten Sie die Software und öffnen Sie die Einstellungen. Navigieren Sie dort zu **Profiles & connections** und fügen Sie ein neues Profil hinzu:



3. Sie müssen nun ein Basis-Profil als Vorlage auswählen. Nehmen Sie hier **SSH connection**:



4. Nun tragen Sie **ihre öffentliche IP-Adresse** (44.194.159.38 ist nur ein Beispiel) sowie den

Port 22 ein. Als Benutzername setzen Sie **ec2-user**. Als Authentifizierungsmethode Wählen Sie **Key**. Nun können Sie über den Button **Add a private key** den Gruppenschlüssel hinzufügen.

SSH DNS-Server

Name

SSH DNS-Server

Group

Ungrouped

Icon

fas fa-desktop



Color

#000000

Disable dynamic tab title

Connection name will be used instead

☐

GENERAL

PORTS

ADVANCED

CIPHERS

LOGIN SCRIPTS

Host

44.194.159.38

Port

22

Username

ec2-user

Authentication method

Auto

Password

Key

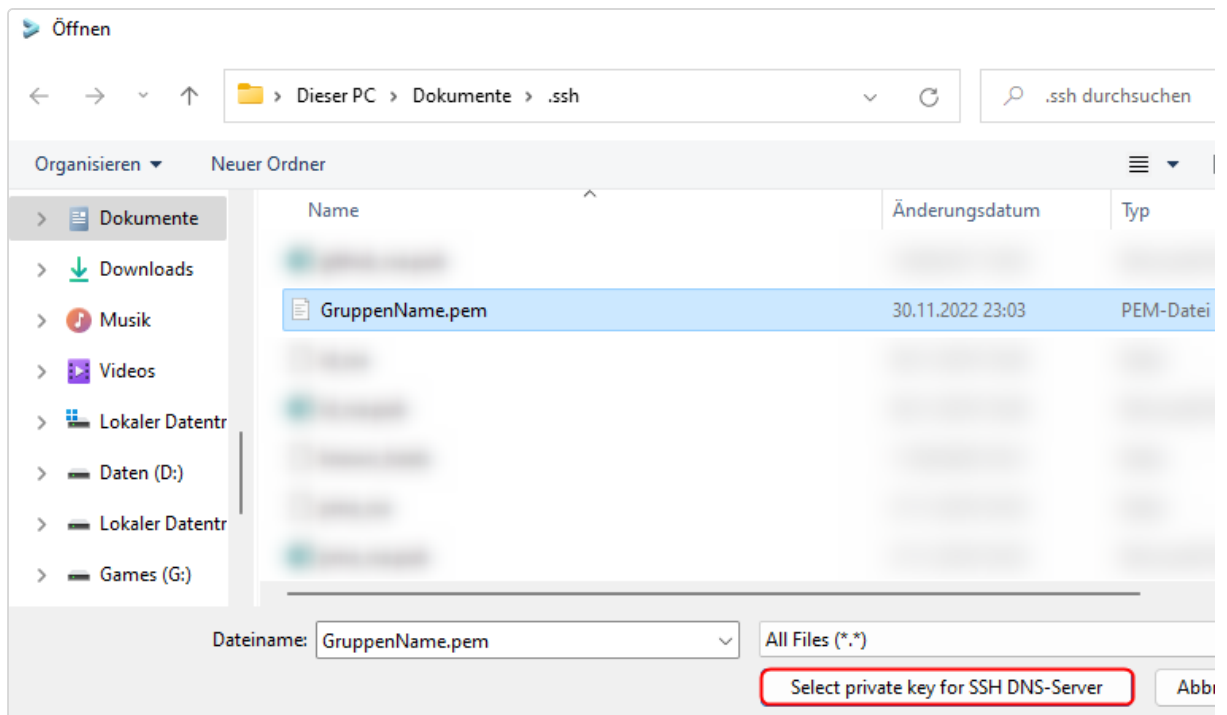
Agent

Interactive

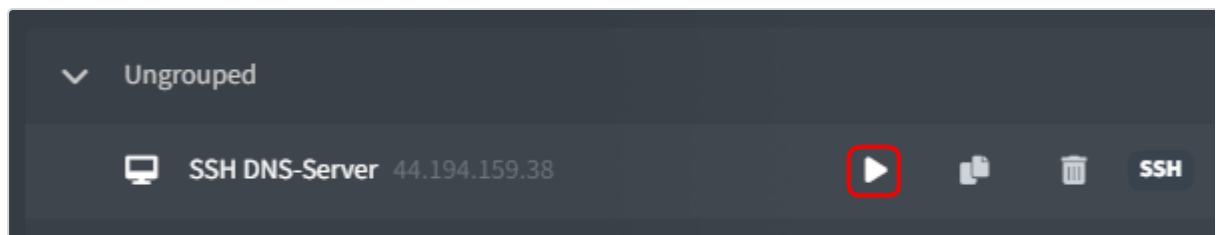
Private keys

Add a private key

5. Wählen Sie hier den Grupenschlüssel, welchen Sie vom Gruppenleiter erhalten haben sollten:



6. Speichern Sie das Profil und starten Sie anschliessend die Verbindung:



7. Möglicherweise müssen Sie nun den Host-Fingerprint bestätigen. Speichern Sie den Fingerprint und fahren sie fort.
8. Die Verbindung sollte nun möglich sein und Sie sollten den Willkommens-Bildschirm sehen:

The screenshot shows an SSH terminal window with a dark theme. The title bar at the top reads '1 ec2-user@ip-172-31-7...'. Below the title bar, the terminal status bar shows 'ec2-user@44.194.159.38 :22' and icons for 'Reconnect', 'SFTP', and 'Ports'. The terminal output shows the SSH connection process: 'Connecting to 44.194.159.38', 'Host key fingerprint:', 'SHA256 02b905178b0b6f61e850358d33acbd070387215f7988086fe2cc9124f8d5deef', and 'Loading private key'. Below this is the Amazon Linux 2 logo and the text 'Amazon Linux 2 AMI'. A URL 'https://aws.amazon.com/amazon-linux-2/' is displayed. The terminal prompt is '[ec2-user@ip-172-31-72-77 ~]\$'.

DNS Server (Bind) installieren

1. Verbinden Sie sich per SSH von Ihrem lokalen Terminal auf den Server.
2. Wechseln Sie direkt in den Administrator-Modus mit **sudo su**
3. Installieren Sie nun über die Paketverwaltung **yum** die Software **bind**:

```
yum install bind
```

Yum wird Ihnen dann eine Auflistung der zu installierenden bind-Version inkl. aller Abhängigkeiten anzeigen. Bestätigen Sie die Installation indem Sie **y** eintippen.

```
Install 1 Package (+8 Dependent packages)
Total download size: 4.2 M
Installed size: 11 M
Is this ok [y/d/N]: y
```

Sie können nachvollziehen, wie **yum** nun die Software bind installiert.

4. Verifizieren Sie nun, dass bind installiert ist. Mit dem Befehl **yum list installed** können Sie eine Liste aller installierte Softwarepakete einsehen:

```
yum list installed
```

Sie sehen nun eine Liste mit allen installierten Softwarepaketen

```
# yum list installed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Installed Packages
GeoIP.x86_64                               1.5.0-11.amzn2.0.2      installed
PyYAML.x86_64                              3.10-11.amzn2.0.2      installed
acl.x86_64                                 2.2.51-14.amzn2        installed
acpid.x86_64                               2.0.19-9.amzn2.0.1     installed
amazon-linux-extras.noarch                2.0.1-1.amzn2          installed
amazon-linux-extras-yum-plugin.noarch     2.0.1-1.amzn2          installed
amazon-ssm-agent.x86_64                   3.1.1732.0-1.amzn2     installed
[...]
```

Wie Sie sicherlich bemerkt haben, ist die Liste sehr lange. Aus diesem Grund werden Sie nun den Output mit dem Befehl **grep** nach **bind** filtern. Hierzu fügen Sie hinter dem bereits bestehenden Befehl einen weiteren Befehl an, mit dem Zeichen **|** übergeben Sie die Ausgabe aus dem vorhergehenden Befehl an den nachfolgenden.

```
yum list installed | grep bind
```

Sie erhalten nun eine List, welche nach dem Stichwort **bind** gefiltert ist.

```
# yum list installed | grep bind
bind.x86_64                                32:9.11.4-26.P2.amzn2.5.2 @amzn2-core
bind-export-libs.x86_64                   32:9.11.4-26.P2.amzn2.5.2 installed
bind-libs.x86_64                          32:9.11.4-26.P2.amzn2.5.2 installed
bind-libs-lite.x86_64                     32:9.11.4-26.P2.amzn2.5.2 installed
bind-license.noarch                       32:9.11.4-26.P2.amzn2.5.2 installed
bind-utils.x86_64                         32:9.11.4-26.P2.amzn2.5.2 installed
rpcbind.x86_64                             0.2.0-44.amzn2         installed
```

Nun ist die Liste etwas übersichtlicher.

Nun ist die Software bind installiert. Im nächsten Kapitel werden Sie den Dienst starten und die Konfiguration genauer unter die Lupe nehmen.

Verwendete Befehle

yum: Mit der Paketverwaltung yum können Sie ganz einfach Software auf Ihrem RHEL-Basierten Linux installieren (rhel, centos, Amazon linux, fedora). Yum organisiert die Software in sogenannten Repositories. Dies sind Verzeichnisse, in denen bekannte Software zur Verfügung gestellt wird. In den Standard-Repositories sind die am meisten verwendeten Softwarepakete vorhanden. Weiter können aber zusätzliche Software-Repositories eingebunden werden, damit auch eine spezielle Software installiert werden kann. Im Rahmen dieses Moduls ist dies aber nicht nötig.

yum stellt eine Reihe von Befehlen zur Verfügung, um Software auf dem Server zu verwalten:

- `yum list <package>` Liste der installierten und verfügbaren Paketen <package>
- `yum list all` Liste aller installierten und verfügbaren Pakete
- `yum list available <package>` Liste aller verfügbaren und installierbaren Pakete
- `yum list updates <package>` Liste aller verfügbaren Pakete <package> , die aktueller als die installierten sind
- `yum list installed <package>` Liste aller installierten Pakete <package>
- `yum info <package>` Kurzbeschreibung zu installierten und verfügbaren Paketen <package>
- `yum search <zeichenkette>` Paketnamen und Beschreibungen durchsuchen nach <zeichenkette>
- `yum install <package>` Installiere die aktuellste Version der Pakete <package> (inkl. der abhängigen!)
- `yum check-update` gibt es aktuellere Pakete in den Repos?
- `yum update` aktualisiere alle z.Z. installierten Pakete
- `yum update <package>` aktualisiere die Pakete <package> (inkl. der abhängigen!)
- `yum erase <package>` deinstalliere Pakete <package> (inkl. aller abhängigen Pakete!)
- `yum remove <package>` deinstalliere Pakete <package> (inkl. aller abhängigen Pakete!)

In anderen Linux-Distributionen kommen andere Paketverwaltungen zum Einsatz. Debian derivative (debian, ubuntu, kali linux) haben mit **apt-get** ein sehr ähnliches Tool zur Verfügung.

grep: Der Befehl grep ist ein Linux-Befehl, der dazu verwendet wird, um Zeichenfolgen in Textdateien zu suchen. Es durchsucht Dateien nach bestimmten Zeichenketten, die angegeben wurden, und gibt alle Zeilen zurück, in denen es die angegebene Zeichenfolge gefunden hat. Es kann auch dazu verwendet werden, um einzelne Dateien oder mehrere Dateien nach den angegebenen Zeichenketten zu durchsuchen.

pipe: Dies ist ein Begriff, der sich aus dem englischen Wort "Pipeline" ableitet, was übersetzt so viel wie "Rohrleitung" bedeutet. Unter Linux bildet eine Pipe einen Datenstrom zwischen zwei Prozessen, die nicht immer miteinander verbunden sind. Dies bedeutet, dass das Ergebnis (Ausgabe) eines Programms als Eingabe für ein anderes Programm genutzt werden kann. Dadurch können größere Aufgaben in kleinere Teilaufgaben aufgeteilt werden, um eine bessere Übersicht zu erhalten.

Konfiguration von Bind

Wir schauen uns nun die Konfiguration von Bind an. Konfigurationen unter Linux sind in der Regel einfache Text-Dateien, welche die nötigen Anweisungen für die Software enthalten.

Grundkonfiguration

Die Konfiguration des BIND-Servers erfolgt über das Verzeichnis **/etc**. Nach der Installation sollten sich hier folgende Dateien finden:

```
named.conf
named.iscdlv.key
named.rfc1912.zones
named.root.key
```

Der Prozess von BIND ist als **named** bekannt. Daher beziehen sich viele der Dateien auf "named" anstelle von "BIND".

Die Datei **named.conf** ist die Haupt-Konfiguration von Bind. Öffnen Sie die Konfiguration nun mit dem Befehl **nano**.

Sie werden nun feststellen, dass die Konfiguration für Bind in gewissen Gruppen zusammengefasst ist, welche durch geschweifte Klammern **{ }** umschlossen werden. In der Gruppe **options** finden Sie die Grundkonfigurationen.

Zeile	Bedeutung
<code>listen-on port 53 { 127.0.0.1; };</code>	Definiert auf welcher IP-Adresse und welchem Port der DNS-Server eingehende Verbindungen annehmen soll. Bei 127.0.0.1 handelt es sich um die localhost Adresse des Servers, der Dienst wird daher nur Verbindungen vom eigenen Server annehmen.
<code>listen-on-v6 port 53 { ::1; };</code>	Dasselbe gibt es auch in einer separaten Variante für IPv6
<code>allow-query { localhost; };</code>	Definiert, wer (welche IP-Adresse) eine DNS-Abfrage stellen darf. Standardmässig ist der Wert auf localhost gesetzt. Der Dienst wird daher nur DNS-Abfragen vom eigenen Server beantworten.
<code>directory</code>	In dem angegebenen Verzeichnis werden die Dateien mit den DNS-Informationen (Zonen) abgelegt.
<code>recursion yes;</code>	Definiert ob der Server rekursive DNS-Abfragen bearbeitet oder nicht. Standardmässig ist diese Option auf yes wodurch der Dienst als DNS-Resolver fungiert. Soll der Dienst nur als autoritativer Nameserver arbeiten, dann muss diese Option auf no gesetzt werden.

Die grundlegenden Einstellungen sind soweit in Ordnung. Es muss aber eine Einstellung angepasst werden, damit der DNS-Server Anfragen von unserer IP-Adresse überhaupt annimmt.

Im Kontext von Einstellungen unter Linux spricht man bei den einzelnen Einstellungsmöglichkeiten von Direktiven.

- Die **listen-on** Direktive ist derzeit auf 127.0.0.1 gesetzt. Aus diesem Grund nimmt der DNS-Server keine Verbindungen von extern an. Um das zu ändern, setzen Sie die Direktive auf **any**. Den Port 53 können Sie belassen, da dies der Standard-Port für DNS ist:

```
listen-on port 53 { any; };
```

Damit ist die Basis-Konfiguration abgeschlossen. Im nächsten Schritt werden wir nun eine DNS-Zone einrichten.

Starten des Dienstes

Bis jetzt ist der DNS-Dienst noch nicht gestartet. Mit dem Befehl **systemctl** können wir einen Dienst steuern:

```
systemctl enable named[] Aktiviert den Dienst und trägt diesen im Autostart ein
systemctl disable named[] Deaktiviert den Dienst und entfernt diesen aus dem Autostart
systemctl start named[] Startet den Dienst
systemctl stop named[] Stoppt den Dienst
systemctl restart named[] Startet den Dienst neu
```

Verschaffen Sie sich einen weitergehenden Überblick über systemctl und wie damit ein Dienst verwaltet werden kann auf folgender Seite: <https://www.digitalocean.com/community/tutorials/how-to-use-systemctl-to-manage-systemd-services-and-units-de>

Hilfe Hilfe, der Dienste startet nicht!

Bei fehlerhaften Konfigurationen kann es sein, dass ein Serverdienst nicht starten kann. Hier gibt es verschiedenen Möglichkeiten die Ursache zu finden. Nachfolgend ist eine Meldung ersichtlich, dass der Dienst named (bind) nicht gestartet werden konnte:

```
# systemctl start named.service
Job for named.service failed because the control process exited with error code. See "systemctl
status named.service" and "journalctl -xe" for details.
```

Möglichkeit 1: Systemctl gibt Ihnen einen kurzen Auszug der letzten Informationen aus dem Log zurück mit **systemctl status named.service**. Dies sieht wie folgt aus:

```
# systemctl status named.service
● named.service - Berkeley Internet Name Domain (DNS)
   Loaded: loaded (/usr/lib/systemd/system/named.service; enabled; vendor preset: disabled)
   Active: failed (Result: exit-code) since Wed 2022-12-14 15:10:12 UTC; 9s ago
     Process: 4379 ExecStop=/bin/sh -c /usr/sbin/rndc stop > /dev/null 2>&1 || /bin/kill -TERM $MAINPID (code=exited, status=0/SUCCESS)
     Process: 4317 ExecReload=/bin/sh -c /usr/sbin/rndc reload > /dev/null 2>&1 || /bin/kill -HUP $MAINPID (code=exited, status=0/SUCCESS)
     Process: 3093 ExecStart=/usr/sbin/named -u named -c ${NAMEDCONF} $OPTIONS (code=exited, status=0/SUCCESS)
     Process: 4390 ExecStartPre=/bin/bash -c if [ ! "$DISABLE_ZONE_CHECKING" == "yes" ]; then /usr/sbin/named-checkconf -z "$NAMEDCONF"; else echo "Checking of zone files is disabled"; fi (code=exited, status=1/FAILURE)
    Main PID: 3095 (code=exited, status=0/SUCCESS)

Dec 14 15:10:12 ip-172-31-72-77.ec2.internal systemd[1]: Starting Berkeley Internet Name Domain (DNS)...
Dec 14 15:10:12 ip-172-31-72-77.ec2.internal bash[4390]: /etc/named.conf:62: missing ';' before end of file
Dec 14 15:10:12 ip-172-31-72-77.ec2.internal systemd[1]: named.service: control process exited, code=exited status=1
Dec 14 15:10:12 ip-172-31-72-77.ec2.internal systemd[1]: Failed to start Berkeley Internet Name Domain (DNS).
Dec 14 15:10:12 ip-172-31-72-77.ec2.internal systemd[1]: Unit named.service entered failed state.
Dec 14 15:10:12 ip-172-31-72-77.ec2.internal systemd[1]: named.service failed.
```

Es wird ersichtlich, dass in auf der Zeile 62 in der Datei /etc/named.conf wohl ein Zeichen fehlt.

Möglichkeit 2: Manchmal möchten Sie auch das gesamte Log einsehen. Bind loggt standardmässig alle Informationen in das Log **/var/log/messages**. Hier können Sie einfach nach dem Stichwort **named** suchen:

```
# grep named /var/log/messages
Dec 14 15:09:15 ip-172-31-72-77 named[3095]: received SIGHUP signal to reload zones
Dec 14 15:09:15 ip-172-31-72-77 named[3095]: loading configuration from '/etc/named.conf'
Dec 14 15:09:15 ip-172-31-72-77 named[3095]: /etc/named.conf:62: missing ';' before end of file
Dec 14 15:09:15 ip-172-31-72-77 named[3095]: reloading configuration failed: failure
```

Auch hier ist die Ursache ersichtlich.

Möglichkeit 3: Bind bietet ein Tool, dass uns erlaubt eine Konfiguration auch zu testen bevor man einen Neustart vornimmt. Mit `named-checkconf` kann während der Laufzeit geprüft werden, ob ein Konfigurationsfehler vorliegt. So kann dieser Fehler behoben werden, **bevor** der Dienst neu gestartet wird. So kann ein Ausfall des Dienstes verhindert werden.

```
# named-checkconf
/etc/named.conf:62: missing ';' before end of file
```

Diese Möglichkeit funktioniert natürlich nur bei Fehlern, welche durch ein Konfigurationsproblem verursacht werden. Wenn andere Fehler vorliegen wie z. B. zu wenig Speicher oder RAM, dann müssen die Informationsn aus dem Log beigezogen werden.

Zugriffsbeschränkung

Tipp: Testen Sie eine DNS-Abfrage wenn der Dienst jeweils gestartet und beendet ist. Notieren Sie die Unterschiede und halten Sie diese für Ihr Testprotokoll wie auch Troubleshooting-Möglichkeiten in Ihrer Dokumentation fest.

Erhalten Sie beim Testen einen REFUSED Fehler? Wenn ja, dann läuft Ihr DNS-Dienst aber es fehlt noch eine Konfiguration.

Was für eine Fehlermeldung haben Sie erhalten? Welche Option wird dafür verantwortlich sein? Prüfen Sie den Abschnitt Grundkonfiguration.

Tipp: Die Erkenntnisse aus der obigen Fragestellung gehört in die Dokumentation zum Testing und Troubleshooting.

Bind erlaubt es, zu steuern, welchen IP-Adressen eine DNS-Abfrage stellen dürfen. Hierzu kann eine Liste mit erlaubten IP-Adressen angelegt werden. Diese sollte am besten im Verzeichnis `/etc/named/` erstellt werden. z. B.

```
# cat /etc/named/zugriff.acl
acl "erlaubte-ips" {
    1. 2. 3. 4;
    2. 3. 4. 5;
    localhost;
    localnets;
};
```

Ihre derzeitige IP-Adresse können Sie unter <https://ip.metanet.ch> herausfinden.

Damit diese Zugriffsliste auch verwendet wird, muss diese nun in der Konfiguration **/etc/named.conf** hinterlegt werden.

Sie können diese ganz unten bei den bestehenden Includes ergänzen:

```
include "/etc/named.rfc1912.zones";  
include "/etc/named.root.key";  
include "/etc/named/zugriff.acl";
```

Zu guter Letzt, müssen Sie nur noch die Zugriffsliste in der Direktive allow-query angeben:

```
allow-query { erlaubte-ips; };
```

Testen Sie anschliessend erneut und halten Sie Ihre Ergebnisse im Dokument fest.

DNS Zonen erstellen

Diese Anleitung beschreibt, wie Sie eine generische DNS-Zone erstellen können. Sie benötigen ggf. mehrere DNS-Zonen um die Anforderungen Ihres Auftrags zu erfüllen. Legen Sie als erstes die nachfolgende Test-Zone an und anschliessend die benötigten DNS-Zonen.

DNS-Zonen

Eine DNS-Zone benötigt zwei Teile:

- Einen Eintrag in der Datei `/etc/named.conf`, welcher dem DNS-Server mitteilt das eine Zone existiert und was für ein Typ die Zone ist.
- Der eigentliche Inhalt der Zone, welcher als eigene Datei abgelegt ist.

Named.conf

Fügen Sie nun ganz am Ende der Datei `/etc/named.conf` folgenden Abschnitt ein:

```
zone "test.tbz" {  
    type master;  
    file "/var/named/test.tbz";  
};
```

- Sie teilen damit dem DNS-Server mit, dass nun die Konfiguration für die DNS-Zone **test.tbz** folgt.
- Dabei setzen Sie den Typ der DNS-Zone mit der Direktive **type** auf **master**.
- Mit der Direktive **file** geben Sie den Pfad zur Datei mit dem Inhalt der DNS Zone an. Diese Datei existiert derzeit noch nicht und muss im nächsten Schritt von uns angelegt werden.

Bei vielen Zonen wird die Datei named.conf schnell mal unübersichtlich gross. Aus diese Grund empfiehlt es sich hier analog der Zugriffsliste eine separate Datei mit den DNS-Zonen unter /etc/named/ (z. B. /etc/named/dns.zones) zu erstellen und diese mit include einzubinden. **Setzen Sie diese Lösung für Ihre benötigten DNS-Zonen um.**

Das Zonen-File

Wechseln Sie nun in das Verzeichnis /var/named und erstellen Sie eine neue Datei mit dem Namen **test.tbz**

In dieser Datei fügen Sie nun den nachfolgenden Inhalt ein:

```
$TTL      600
@         IN      SOA     ns1.test.tbz. admin.test.tbz. (
                        2          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
@         IN      NS      ns1.test.tbz.
@         IN      A       8.8.8.8
ns1[]IN[]A[]<IP des DNS-Servers>
```

Sehen wir uns das Zonen-File etwas genauer an:

- Die TTL (Time to life) definiert, wie lange ein DNS-Resolver die Zonen-Informationen im Cache behalten wird. Eine niedrige TTL wird die Last auf dem Server erhöhen, da ein DNS-Resolver zur Abfrage der selben DNS-Zone mehr Requests stellen wird. Der Wert wird in Sekunden angegeben.
- Der Origin definiert den Namen der DNS-Zone.
- Es folgt ein Block mit den Basis Zonen-Informationen, hier ist vor allem die **Serial** relevant. Bei der Serial handelt es sich sozusagen um eine Version der Zone, bei jeder Anpassung muss die Serial erhöht werden, damit Slave-DNS-Server eine Änderung in der Zone erkennen können.

DNS-Zone überprüfen.

Ist der Inhalt einer DNS-Zone nicht korrekt, kann der DNS-Dienst nicht mehr gestartet werden. Aus diesem Grund sollte die Zone vor einem Neustart überprüft werden.

```
named-checkzone test.tbz /var/named/test.tbz
```

DNS-Zone testen

Verwenden Sie die bekannten Tools um zu testen, ob die von Ihnen angelegten DNS-Zonen funktionieren. Im einen weiterführenden Test, können Sie Ihren Nameserver auch bei Ihrem Windows hinterlegen und die betreffenden Domains aufrufen.

■ Logging

Abschliessend richten Sie noch die Log-Funktion für den Dienst ein. Das Log enthält alle relevanten Informationen aus dem Betrieb des Dienstes und erlaubt im Nachhinein Fehler nachzuvollziehen.

Passen Sie als erstes den Abschnitt **logging** in der Datei **/etc/named.conf** wie folgt an:

```

logging {
    channel default_file {
        file "/var/log/named/default.log" versions 3 size 5m;
        severity dynamic;
        print-time yes;
    };

    channel resolver_file {
        file "/var/log/named/resolver.log" versions 3 size 5m;
        severity debug 11;
        print-time yes;
    };

    channel queries_file {
        file "/var/log/named/queries.log" versions 3 size 5m;
        severity debug 11;
        print-time yes;
    };

    channel database_file {
        file "/var/log/named/database.log" versions 3 size 5m;
        severity debug 11;
        print-time yes;
    };

    category default { default_file; };
    category queries { queries_file; };
    category database { database_file; };
    category resolver { resolver_file; };

};

```

Nun müssen die dazugehörigen Log-Dateien erstellt und mit den richtigen Berechtigungen versehen werden:

```
mkdir -p /var/log/named/

touch /var/log/named/default.log
touch /var/log/named/resolver.log
touch /var/log/named/queries.log
touch /var/log/named/database.log

chown named: /var/log/named/default.log
chown named: /var/log/named/resolver.log
chown named: /var/log/named/queries.log
chown named: /var/log/named/database.log
```

Anschliessend können Sie den Dienst neu starten und in das Verzeichnis **/var/log/named** wechseln.

Sie haben nun verschiedene Möglichkeiten, die Informationen in den Logs auszuwerten:

tail -f: Mit dem Befehl "tail -f" können Sie die Log-Informationen fortlaufend ausgeben. Mit der Tastenkombination CTRL + C können Sie die Ausgabe wieder schliessen.

```
tail -f /var/log/named/resolver.log
```

Wenn Sie nur nach den Informationen einer bestimmten Domain suchen, macht es Sinn wenn Sie die Ausgabe mit **grep** filtern:

```
tail -f /var/log/named/resolver.log | grep domain.ch
```

Mit dem * als Wildcard-Zeichen, können Sie auch gleichzeitig mehrere Logdateien ausgeben:

```
tail -f /var/log/named/*.log
```

grep: Mit grep können Sie auch direkt innerhalb der Dateien suchen. Damit durchsuchen Sie dann den gesamten Inhalt der betreffenden Logdatei:

```
grep domain.ch /var/log/named/resolver.log
```

Mit dem * als Wildcard-Zeichen, können Sie auch gleichzeitig mehrere Logdateien durchsuchen:

```
grep domain.ch /var/log/named/*.log
```

Bedeutung Log

- Default: Im Default-Log werden die normalen BIND-Informationen geloggt.
- Queries: Hier werden die eingehenden DNS-Anfragen verzeichnet.
- Database: Hier werden abfragen in die lokale DNS-Datenbank (Cache) verzeichnet.
- Resolver: Hier werden die DNS-Abfragen verzeichnet, welcher der Resolver auf andere DNS-Server macht.

Dokumentation

Notieren Sie in der Dokumentation welche Logs sich wo befinden und was diese bedeuten. Machen Sie ggf. auch ein zwei Beispiele und beschreiben Sie, was im Log aufgezeichnet wird.