Project Protocol – Milestone 2

Team 1

Alexander Nachtmann,
Markus Rösner,
Max Sinnl and
Stephanie Rauscher

Infrastructure Overview

In contrast to our Infrastructure Specification Protocol, we have made a few changes to our infrastructure. All servers use **Ubuntu version 22.04** as their operating system.

Instance Name	Container Type	IP	Domain	Packages	Subnet
Gitlab Server	T2.medium	Private IP: 10.0.1.49 Public IP (elastic): 54.88.228.103	git.team01.at	GitLab 16.6.1	Public
Bastion Host	T2.micro	Private IP: 10.0.2.61 Public IP always changes after restart	bastionhost.team01.at	-	Public
Gitlab Runner	T2.small	Private IP: 10.0.2.70	gitrunner.team01.at	GitLab Runner 16.6.1	Private
Main DNS-Server	T2.micro	Private IP: 10.0.2.49	dnsmain.team01.at	BIND 9	Private
Backup DNS Server	T2.micro	Private IP: 10.0.2.48	dnsbackup.team01.at	BIND 9	Private

For this milestone, we have omitted the configuration of the LDAP server.

Routing Tables

Public Subnet Route Table

Destination IP	Target name
10.0.0.0/16	VPC
0.0.0.0/0	Internet Gateway

Destination IP 10.0.0.0/16 - Target: VPC
This entry indicates that any traffic destined for an IP address within the 10.0.0.0/16 range) should be routed internally within the VPC. It essentially means that all IPs in this range are part of the VPC network. This is a standard entry for internal network routing within the VPC.

Destination IP 0.0.0.0/0 - Target: Internet Gateway
This entry is for routing all other traffic (not destined for the internal VPC network) to the Internet Gateway. The destination IP 0.0.0.0/0 represents all IP addresses not covered by more specific routes.

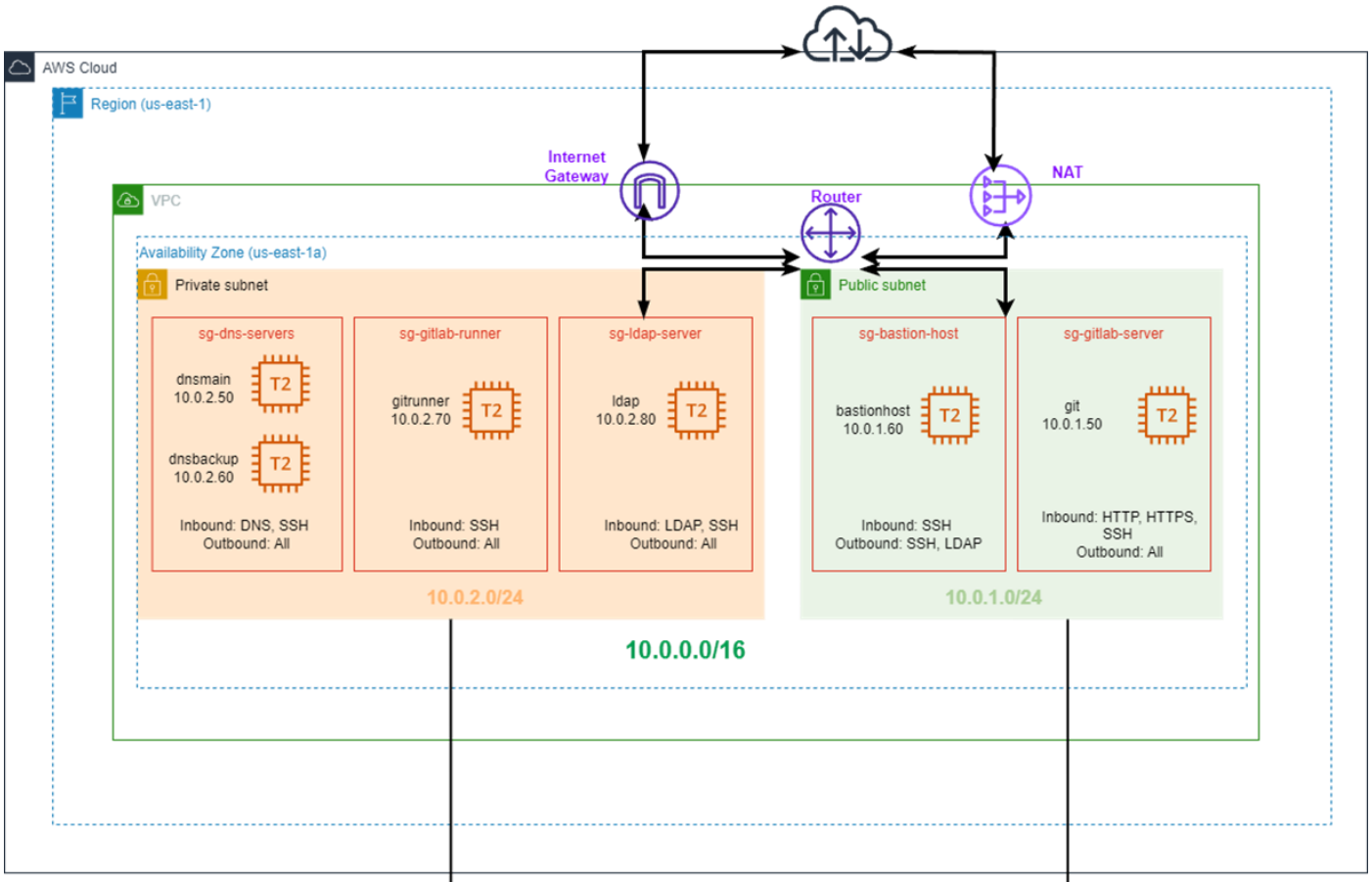
Private Subnet Route Table

Destination IP	Target name
0.0.0.0/0	NAT Gateway
10.0.0.0/16	VPC

Destination IP 0.0.0.0/0 - Target: NAT Gateway
This route directs all traffic that is not for local destinations (i.e., any destination not within the VPC) to a Network Address Translation (NAT) Gateway. It's used for allowing instances in the Private Subnet to access the internet for updates or downloads, but not allowing incoming internet traffic to initiate connections with those instances.

Destination IP 10.0.0.0/16 - Target: VPC
Similar to the Public Subnet, this entry ensures that traffic destined for the VPC's internal IP range is kept within the VPC network.

Network Topology Graph



Private Route Table	
Destination	*Target*
0.0.0.0/0	NAT
10.0.0.0/16	VPC

Public Route Table	
Destination	*Target*
10.0.0.0/16	VPC
0.0.0.0/0	igw

Security Groups

The (updated) Security Groups we used for our Project were:

Inbound Rules

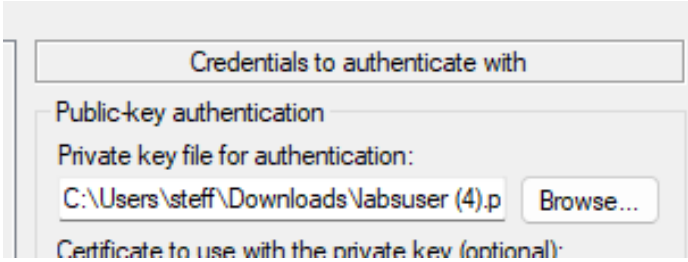
SG Name	Type	Port Range	Source
scg-gitlab-server	HTTP, HTTPS, SSH	TCP 80, TCP443 TCP 22	0.0.0.0/0 0.0.0.0/0 10.0.1.61/32
scg-bastion-host	SSH	TCP 22	10.0.0.0/16
scg-gitlab-runner	SSH	TCP 22	10.0.1.61/32
scg-dnsservers	DNS, SSH	TCP/UDP 53 TCP 22	10.0.0.0/16 10.0.1.61/32

Outbound Rules

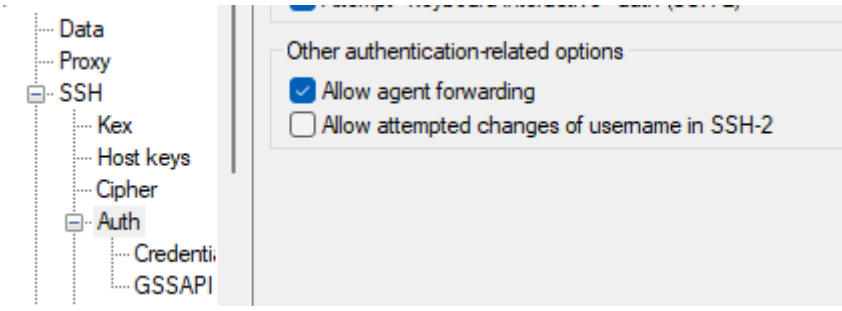
SG Name	Type	Port Range	Source
scg-gitlab-server	All Destinations	-	0.0.0.0/0
scg-bastion-host	SSH	TCP 22	10.0.0.0/16
scg-gitlab-runner	All Destinations	-	0.0.0.0/0
scg-dns-servers	All Destinations	-	0.0.0.0/0

Configuration of Services

First, we start by connecting to the Bastion Host (Jump Server) via SSH. We upload the .ppk file from AWS as our credentials in SSH, under 'Auth' -> 'Credentials'.



and tick the box labeled 'Allow Agent Forwarding' in SSH -> Auth.



We use the public IP of the Bastion Host to connect to the instance. From there, we can connect to every other instance in our infrastructure using the command *ssh ubuntu@IP*.

Configuration of the Primary DNS Server

From the Jump Server, we connect to the Primary DNS Server using the command `ssh ubuntu@10.0.2.49` (10.0.2.49 being the IP of our Primary DNS Server). From there, we use the commands

`sudo apt update` and

`sudo apt install bind9` for installing BIND on our DNS Server.

To ensure BIND is running, we use:

```
sudo systemctl start named
```

```
sudo systemctl enable named
```

Then we edit the **/etc/bind/named.conf** file to define our zones:

```
sudo nano /etc/bind/named.conf
```

```
ubuntu@ip-10-0-2-49: ~
  GNU nano 6.2                    /etc/bind/named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
zone "team01.at" {
    type master;
    file "/etc/bind/db.team01.at";

    allow-transfer { 10.0.2.48; };
};
```

Next, we create our zone file **/etc/bind/db.team01.at** with all of our DNS Zones:

```
sudo nano /etc/bind/db.team01.at
```

```
ubuntu@ip-10-0-2-49: ~
  GNU nano 6.2                    /etc/bind/db.team01.at
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     dnsmain.team01.at. admin.team01.at. (
                              6         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      dnsmain.team01.at.
@       IN      NS      dnsbackup.team01.at.
dnsmain IN      A       10.0.2.49
dnsbackup       IN      A       10.0.2.48
bastionhost IN  A       10.0.1.61
git     IN      A       10.0.1.49
git     IN     A       54.88.228.103
gitrunner IN   A       10.0.2.70
ldap    IN     A       10.0.2.80
                                [ Read 21 lines ]
^G Help         ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location
^X Exit         ^R Read File    ^\ Replace      ^U Paste        ^J Justify      ^/ Go To Line
```

we check the configuration for errors:

```
sudo named-checkconf
sudo named-checkzone team01.at /etc/bind/zones/db.team01.at
```

Restart BIND to apply changes:

```
sudo systemctl restart named
```

To avoid local resolving Problems, we navigate to the File

```
sudo nano /etc/systemd/resolved.conf
```

```
ubuntu@ip-10-0-2-49: ~
  GNU nano 6.2                    /etc/systemd/resolved.conf
#  This file is part of systemd.
#
#  systemd is free software; you can redistribute it and/or modify it under the
#  terms of the GNU Lesser General Public License as published by the Free
#  Software Foundation; either version 2.1 of the License, or (at your option)
#  any later version.
#
# Entries in this file show the compile time defaults. Local configuration
# should be created by either modifying this file, or by creating "drop-ins" in
# the resolved.conf.d/ subdirectory. The latter is generally recommended.
# Defaults can be restored by simply deleting this file and all drop-ins.
#
# Use 'systemd-analyze cat-config systemd/resolved.conf' to display the full config.
#
# See resolved.conf(5) for details.

[Resolve]
# Some examples of DNS servers which may be used for DNS= and FallbackDNS=:
# Cloudflare: 1.1.1.1#cloudflare-dns.com 1.0.0.1#cloudflare-dns.com 2606:4700:4700:>
# Google:     8.8.8.8#dns.google 8.8.4.4#dns.google 2001:4860:4860::8888#dns.google>
# Quad9:      9.9.9.9#dns.quad9.net 149.112.112.112#dns.quad9.net 2620:fe::fe#dns.q>
DNS=10.0.2.49 10.0.2.48
#FallbackDNS=
#Domains=
#DNSSEC=no
#DNSOverTLS=no
```

We will edit the file and, under the 'DNS=' section, insert the IPs of both our DNS servers.

We also create the File `sudo nano /etc/netplan/99-custom-dns.yaml` with the following content

```
ubuntu@ip-10-0-2-49: ~
  GNU nano 6.2                /etc/netplan/99-custom-dns.yaml
network:
  version: 2
  ethernets:
    eth0:
      nameservers:
        addresses: [10.0.2.49, 10.0.2.48]
      dhcp4-overrides:
        use-dns: false
        use-domains: false
```

use these commands to apply changes

```
sudo netplan generate
```

```
sudo netplan apply
```

Restart BIND

```
sudo systemctl restart bind9
```

Testing

To test if our zones are set up correctly, we use the following command from another server in our VPC (from our Gitlab Server with the IP 10.0.1.49):

```
nslookup bastionhost.team01.at 10.0.2.49
```

```
ubuntu@ip-10-0-1-49:~$ nslookup bastionhost.team01.at 10.0.2.49
Server:         10.0.2.49
Address:        10.0.2.49#53

Name:   bastionhost.team01.at
Address: 10.0.1.61
```

We also try it on our DNS Server

```
nslookup git.team01.at 10.0.2.49
```

```
ubuntu@ip-10-0-2-49:~$ nslookup git.team01.at 10.0.2.49
Server:         10.0.2.49
Address:        10.0.2.49#53

Name:   git.team01.at
Address: 54.88.228.103
Name:   git.team01.at
Address: 10.0.1.49
```

And we query our Main DNS Server one last time

```
dig bastionhost.team01.at @10.0.2.49
```

```
ubuntu@ip-10-0-2-49: ~

ubuntu@ip-10-0-2-49:~$ dig bastionhost.team01.at @10.0.2.49

; <<>> DiG 9.18.18-0ubuntu0.22.04.1-Ubuntu <<>> bastionhost.team01.at @10.0.2.49
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46824
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 6a5d291cfc5c2c40010000006576da6dc874d4765e6e8c71 (good)
;; QUESTION SECTION:
;bastionhost.team01.at.         IN      A

;; ANSWER SECTION:
bastionhost.team01.at.  604800  IN      A       10.0.1.61

;; Query time: 0 msec
;; SERVER: 10.0.2.49#53(10.0.2.49) (UDP)
;; WHEN: Mon Dec 11 09:46:21 UTC 2023
;; MSG SIZE  rcvd: 94

ubuntu@ip-10-0-2-49:~$
```

Since we can see it returning the IP of our Bastion Host and Gitlab Server, our setup works as planned.

For also making DNS reverse lookup work, we do the following:

We locate the **/etc/bind/named.conf.local** file

`sudo nano /etc/bind/named.conf.local`

and edit it like this:

```
 GNU nano 6.2                    /etc/bind/named.conf.local
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
//
// Do any local configuration here
zone "2.0.10.in-addr.arpa" {
    type master;
    file "/etc/bind/db.2.0.10";
    allow-transfer { 10.0.2.48; };
};

zone "1.0.10.in-addr.arpa" {
    type master;
    file "/etc/bind/db.1.0.10";
    allow-transfer { 10.0.2.48; };
};
```

The file **db.2.0.10** is used for defining reverse zone files in the private subnet, while the file **db.1.0.10** is for defining reverse zone files in the public subnet. We specify 'type master;' because this server is our main DNS, and we set 'allow-transfer' to enable the transfer of all zone files later to our backup DNS server.

We create both files and then edit them with the reverse zone files for our subnets:

`sudo nano /etc/bind/db.2.0.10` Private Subnet

```
ubuntu@ip-10-0-2-49: ~
 GNU nano 6.2                    /etc/bind/db.2.0.10
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     dnsmain.team01.at. admin.team01.at. (
                            2           ; Serial
                       604800           ; Refresh
                        86400           ; Retry
                      2419200           ; Expire
                       604800 )         ; Negative Cache TTL
;
@       IN      NS      dnsmain.team01.at.
49      IN      PTR     dnsmain.team01.at.
48      IN      PTR     dnsbackup.team01.at.
70      IN     PTR     gitrunner.team01.at.
80      IN     PTR     ldap.team01.at.
```

`sudo nano /etc/bind/db.1.0.10` Public Subnet

```
ubuntu@ip-10-0-2-49: ~
  GNU nano 6.2                          /etc/bind/db.1.0.10
$TTL    604800
@       IN      SOA     dnsmain.team01.at. admin.team01.at. (
                            3         ; Serial
                       604800         ; Refresh
                        86400         ; Retry
                      2419200         ; Expire
                       604800 )       ; Negative Cache TTL
;
@       IN      NS      dnsmain.team01.at.
@       IN      NS       dnsbackup.team01.at.
51      IN      PTR     bastionhost.team01.at.
49      IN      PTR     git.team01.at.
```

We will also configure our DNS server to forward queries to external DNS servers. If there is no local record for the external domain, the query is forwarded to the configured forwarder DNS servers, which in our case are Google's DNS servers at 8.8.8.8 and 8.8.4.4. Once Google's DNS servers retrieve the response, they send this information back to our DNS server.

Additionally, we use 'allow-query-cache' to avoid DNS cache problems.

`sudo nano /etc/bind/named.conf.options`

```
ubuntu@ip-10-0-2-49: ~
  GNU nano 6.2                /etc/bind/named.conf.options *
options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.


    forwarders {
        8.8.8.8; // Google's DNS
       8.8.4.4; // googles secondary dns
    };
    forward only;

  // Cache access control
    allow-query-cache { any; };

        //========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================
        dnssec-validation auto;

        listen-on-v6 { any; };
};

^G Help      ^O Write Out  ^W Where Is   ^K Cut      ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste    ^J Justify    ^/ Go To Line
```

Testing

We will now verify the functionality of reverse lookup and DNS forwarding.

Reverse Lookup:

`dig -x 10.0.2.70 @10.0.2.49` (from Bastion Host)

```
ubuntu@ip-10-0-1-61: ~
Connection to 10.0.2.49 closed.
ubuntu@ip-10-0-1-61:~$ dig -x 10.0.2.70 @10.0.2.49

; <<>> DiG 9.18.12-0ubuntu0.22.04.3-Ubuntu <<>> -x 10.0.2.70 @10.0.2.49
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60145
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: e48251b18ddd850b010000006576ddac7b4e7d796fd90ca7 (good)
;; QUESTION SECTION:
;70.2.0.10.in-addr.arpa.                IN      PTR

;; ANSWER SECTION:
70.2.0.10.in-addr.arpa. 604800  IN      PTR     gitrunner.team01.at.

;; Query time: 0 msec
;; SERVER: 10.0.2.49#53(10.0.2.49) (UDP)
;; WHEN: Mon Dec 11 10:00:12 UTC 2023
;; MSG SIZE  rcvd: 112

ubuntu@ip-10-0-1-61:~$
```

`dig +short -x 10.0.2.70 @10.0.2.49` (from Bastion Host)

```
ubuntu@ip-10-0-1-61:~$ dig +short -x 10.0.2.70 @10.0.2.49
gitrunner.team01.at.
ubuntu@ip-10-0-1-61:~$
```

`dig +short -x 10.0.2.70 @10.0.2.49` (from Gitlab Server)

`dig +short -x 10.0.1.61 @10.0.2.49` (from Gitlab Server)

```
ubuntu@ip-10-0-1-49:~$ dig +short -x 10.0.2.70 @10.0.2.49
gitrunner.team01.at.
ubuntu@ip-10-0-1-49:~$ dig +short -x 10.0.1.61 @10.0.2.49
bastionhost.team01.at.
ubuntu@ip-10-0-1-49:~$
```

Forward DNS Lookup for an External Domain

`dig www.google.com @10.0.2.49`

```
ubuntu@ip-10-0-1-49: ~
buntu@ip-10-0-1-49:~$ dig www.google.com @10.0.2.49

 <<>> DiG 9.18.12-0ubuntu0.22.04.3-Ubuntu <<>> www.google.com @10.0.2.49
; global options: +cmd
; Got answer:
; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26840
; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1

; OPT PSEUDOSECTION:
 EDNS: version: 0, flags:; udp: 1232
 COOKIE: 13a1b1b3d62f5751010000006576f3a1dfda1cc4ac21525d (good)
; QUESTION SECTION:
www.google.com.                         IN      A

; ANSWER SECTION:
ww.google.com.          247     IN      A       172.253.63.99
ww.google.com.          247     IN      A       172.253.63.103
ww.google.com.          247     IN      A       172.253.63.104
ww.google.com.          247     IN      A       172.253.63.105
ww.google.com.          247     IN      A       172.253.63.106
ww.google.com.          247     IN      A       172.253.63.147

; Query time: 7 msec
; SERVER: 10.0.2.49#53(10.0.2.49) (UDP)
; WHEN: Mon Dec 11 11:33:53 UTC 2023
; MSG SIZE  rcvd: 167
```

From these tests, we can see that the reverse lookup for our main DNS server is functioning properly.

Configuring the Backup DNS Server

We disconnect from our Primary DNS using `exit` and connect to our Backup DNS using `ssh ubuntu@10.0.2.48`

We start configuring the Backup DNS Server like we did with the primary Server previously with the following commands

`sudo apt update` and

`sudo apt install bind9` for installing BIND on our DNS Server.

We create a new directory

`sudo mkdir /var/cache/bind/slaves`

`sudo chown bind:bind /var/cache/bind/slaves`

Then, we modify our **named.conf.local** file to match the configuration shown in the screenshot. By setting 'type slave;', we designate this server as our Backup DNS (Slave) Server.

```
sudo nano /etc/bind/named.conf.local
```

```
ubuntu@ip-10-0-2-48: ~
  GNU nano 6.2                    /etc/bind/named.conf.local
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

// Secondary DNS configuration for reverse zones
zone "2.0.10.in-addr.arpa" {
    type slave;
    file "slaves/db.2.0.10";
    masters { 10.0.2.49; }; // IP address of the primary DNS server
};

zone "1.0.10.in-addr.arpa" {
    type slave;
    file "slaves/db.1.0.10";
    masters { 10.0.2.49; }; // IP address of the primary DNS server
};
```

We also edit the **named.conf** file

```
sudo nano /etc/bind/named.conf
```

```
ubuntu@ip-10-0-2-48: ~
  GNU nano 6.2                    /etc/bind/named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
zone "team01.at" {
    type slave;
    file "slaves/db.team01.at";
    masters { 10.0.2.49; }; // IP address of the primary DNS server
};
```

There is no need to copy paste the zone files, since they will transfer automatically. After editing both files, we restart BIND

```
sudo systemctl restart bind9
```

If the zone transfer worked, the files on the secondary server should look similar to this

```
ubuntu@ip-10-0-2-48: ~
  GNU nano 6.2                    /var/cache/bind/slaves/db.team01.at
^@^@^@^B^@^@^@^Aes□□^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@Y^@^A^@^F^@^@^@       :□^@
^@^A=^@^@^@/^@^A^@^A^@^@^@      :□^@^@^@^A^@^U  dnsbackup^Fteam01^Bat^@^@^D
^@^B0^@^@^@-^@^A^@^A^@^@^@      :□^@^@^@^A^@^S^Gdnsmain^Fteam01^Bat^@^@^D
^@^B1^@^@^@)^@^A^@^A^@^@^@      :□^@^@^@^A^@^O^Cgit^Fteam01^Bat^@^@^D
^@^A1^@^@^@/^@^A^@^A^@^@^@      :□^@^@^@^A^@^U  gitrunner^Fteam01^Bat^@^@^D
^@^BF^@^@^@*^@^A^@^A^@^@^@      :□^@^@^@^A^@^P^Dldap^Fteam01^Bat^@^@^D
^@^BP
```

Testing

We test if our Backup DNS Server (10.0.2.48) can take queries as well.

```
nslookup git.team01.at 10.0.2.48
```

```
ubuntu@ip-10-0-1-49:~$ nslookup git.team01.at 10.0.2.48
Server:         10.0.2.48
Address:        10.0.2.48#53

Name:   git.team01.at
Address: 10.0.1.49

ubuntu@ip-10-0-1-49:~$
```

```
dig -x 10.0.1.61 @10.0.2.48
```

```
ubuntu@ip-10-0-1-49:~$ dig -x 10.0.1.61 @10.0.2.48

; <<>> DiG 9.18.12-0ubuntu0.22.04.3-Ubuntu <<>> -x 10.0.1.61 @10.0.2.48
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7576
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 477b96f4dcacf39c010000006576f57322e3f32253fb9200 (good)
;; QUESTION SECTION:
;61.1.0.10.in-addr.arpa.                IN      PTR

;; ANSWER SECTION:
61.1.0.10.in-addr.arpa. 604800  IN      PTR     bastionhost.team01.at.

;; Query time: 0 msec
;; SERVER: 10.0.2.48#53(10.0.2.48) (UDP)
;; WHEN: Mon Dec 11 11:41:39 UTC 2023
;; MSG SIZE  rcvd: 114

ubuntu@ip-10-0-1-49:~$
```

As evidenced by these outputs, the Backup Server also responds to DNS queries, confirming that the zone transfer was successful.

Gitlab Server

Add the GitLab package repository and install the package:

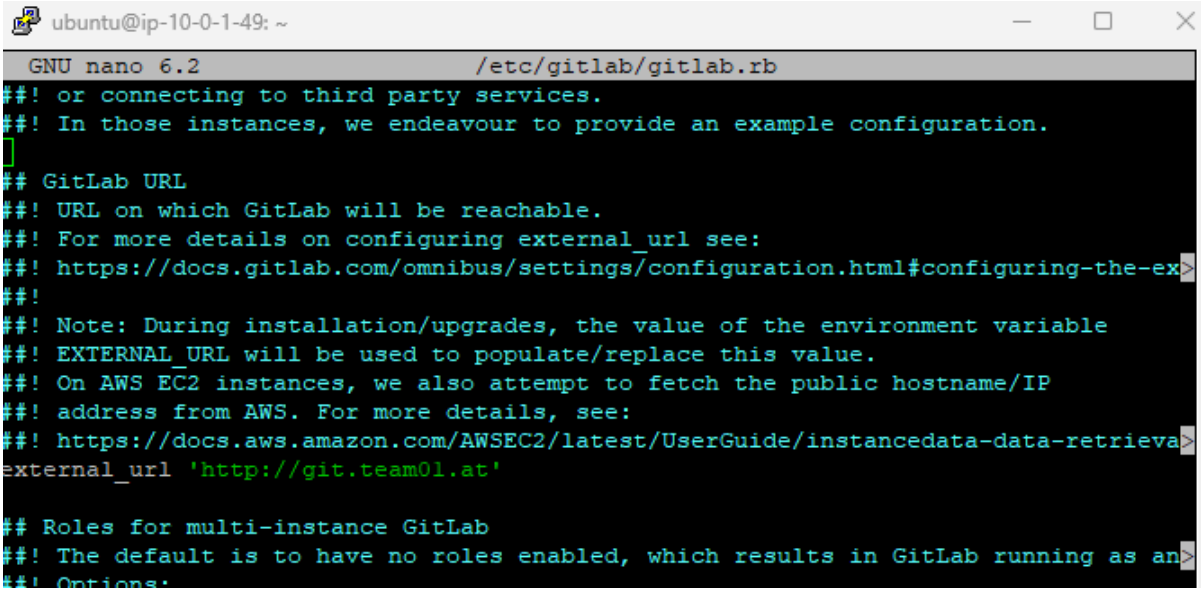
```
curl
https://packages.gitlab.com/install/repositories/gitlab/gitlab
-ce/script.deb.sh | sudo bash
```

```
sudo apt-get install gitlab-ce
```

Locate to

```
sudo nano /etc/gitlab/gitlab.rb
```

And edit the file at the section where you are prompted to enter your URL (Your GitLab domain)



Now GitLab is installed, and the interface should already be visible at your public IP for the GitLab Server.

To change the password for the root user, enabling login access, follow these steps:

Open the GitLab Rails Console

```
sudo gitlab-rails console -e production
```

Change the root Password

```
user = User.find_by(username: 'root')
```

```
user.password = 'new_password'
```

```
user.password_confirmation = 'new_password'
```

```
user.save!
```

Exit the Rails Console

```
exit
```

Gitlab Runner

First, run server Updates:

```
sudo apt update
```

Install Repository & GitLab Runner

```
curl -L
https://packages.gitlab.com/install/repositories/runner/gitlab
-runner/script.deb.sh | sudo bash
```

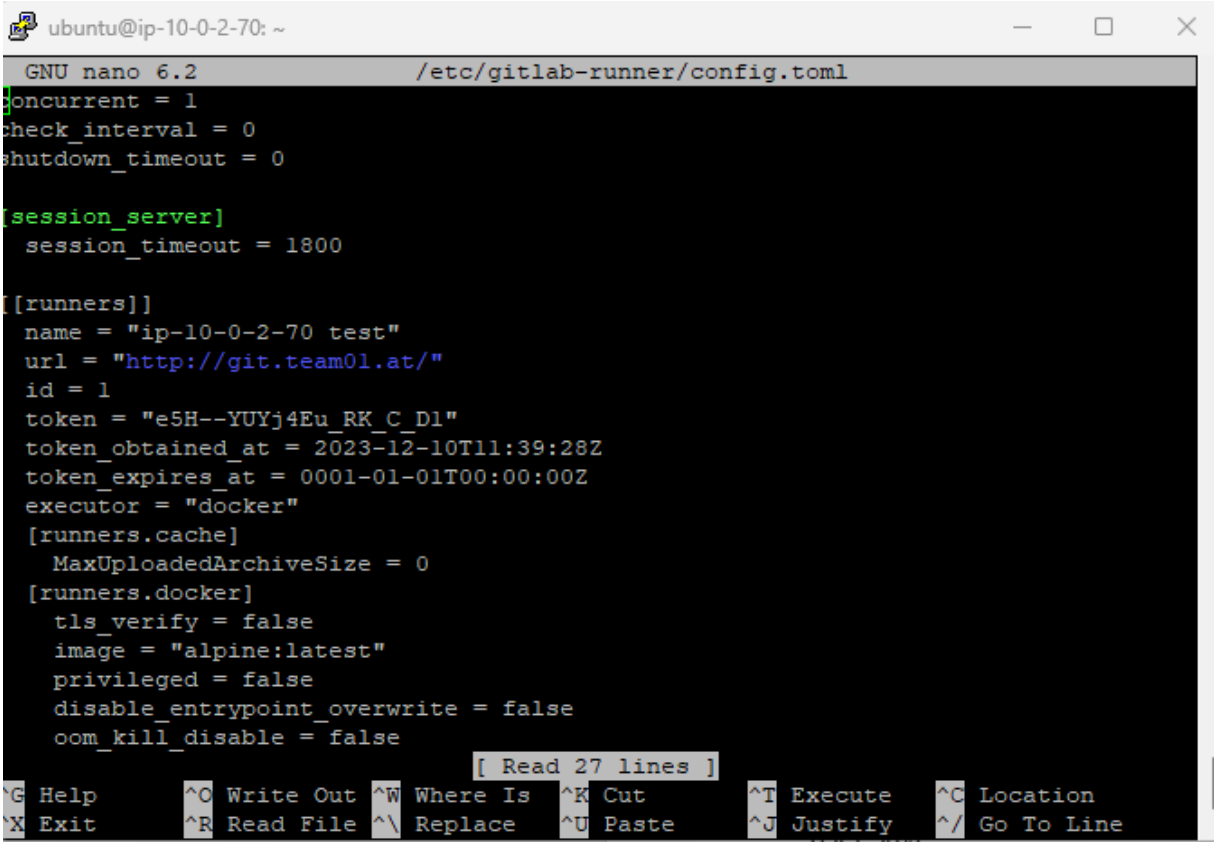
```
sudo apt-get install gitlab-runner
```

Register the GitLab Runner. You will be asked for the Domain of your External URL, your GitRunner Token and optional Tags.

```
sudo gitlab-runner register
```

If everything went well, you will see your Configuration of the Runner in the **config.toml** File.

```
sudo nano /etc/gitlab-runner/config.toml
```



Creating a Repository on GitLab Server

Go to your GitLab Server in your Browser and log in.
Navigate to "Projects" and click "New project" or "Create project".
Enter the project name (e.g., "video").
Set Visibility Level to "Private".
Click "Create project".

Setting Up Your Local Repository:
Create a new directory and initialize it as a Git repository:

```
mkdir video
```

```
cd video
```

```
git init
```

Rename the default branch to "video" (optional):

```
git branch -m video
```

Add the remote GitLab repository:

```
git remote add origin http://git.team01.at/root/video.git
```

Create a new file, stage, and commit it:

```
echo "Hello World" > hello.txt
git add hello.txt
git commit -m "Add video hello.txt"
```

Push the commit to the GitLab repository:

```
git push -u origin video
```

Cloning the repository

To clone a repository in a folder:

```
git clone http://git.team01.at/root/video.git <name of folder>
```