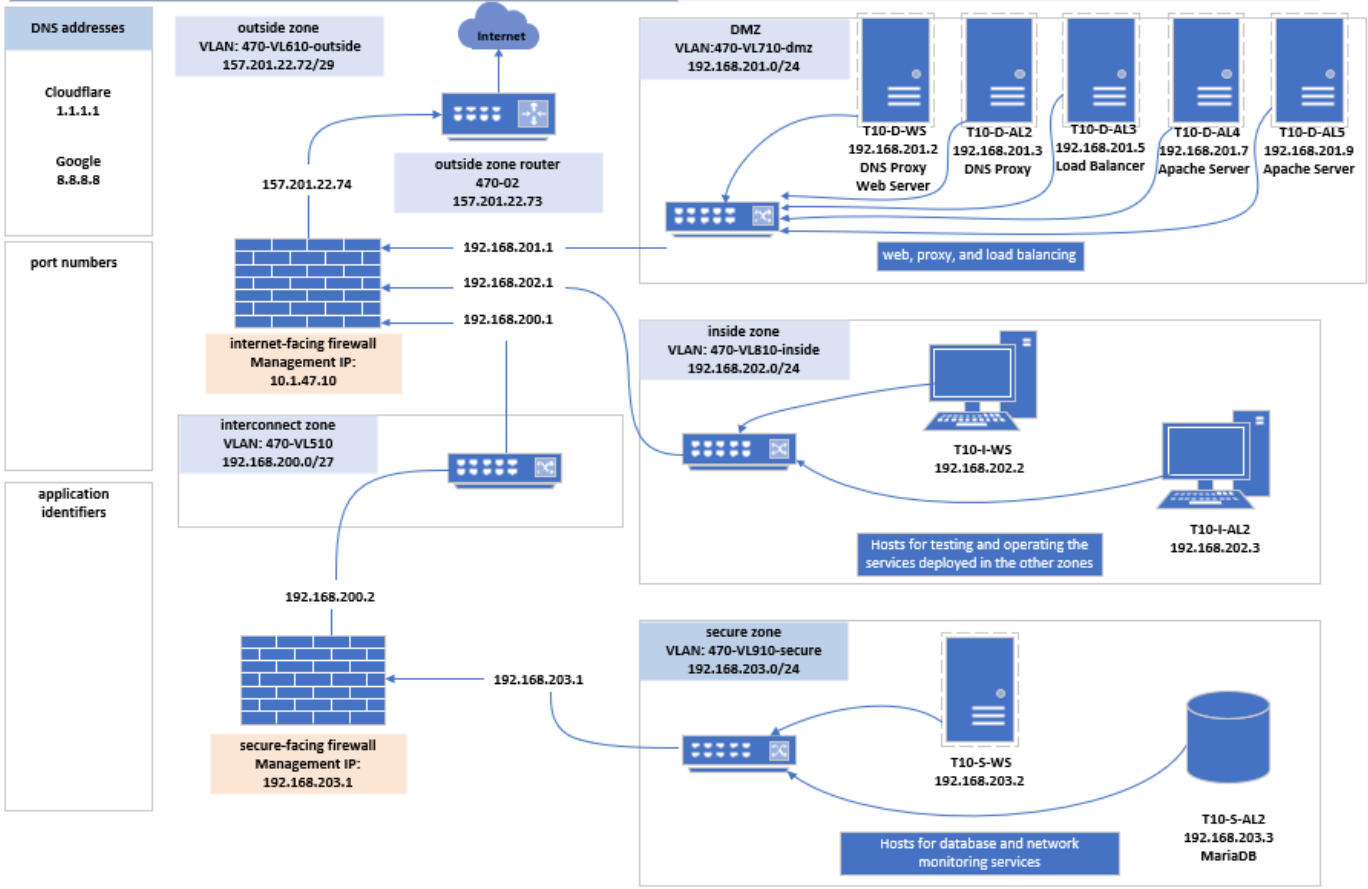
Tiered app

By Carlos Gerez, Christopher Ditto , and Mark Riley Slik

cit470
Task: Diagram

Team 10 Layer 3: outside zones' public IPv4 address assignments

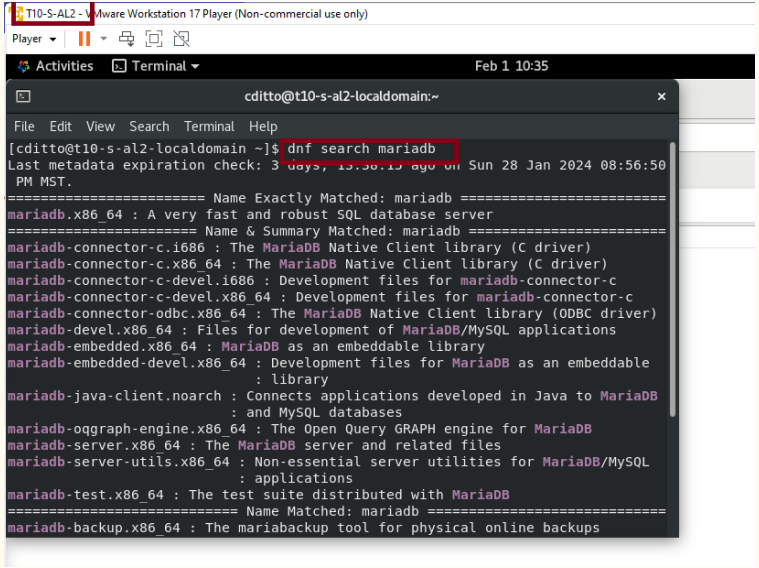
public space (IPv4 subnet ID)	router	firewall (dynamic NAT)	static NAT	(broadcast)
157.201.22.72/29	157.201.22.73	157.201.22.74 470t10ra.cit.byui.edu	157.201.22.75- 157.201.22.78	157.201.22.79

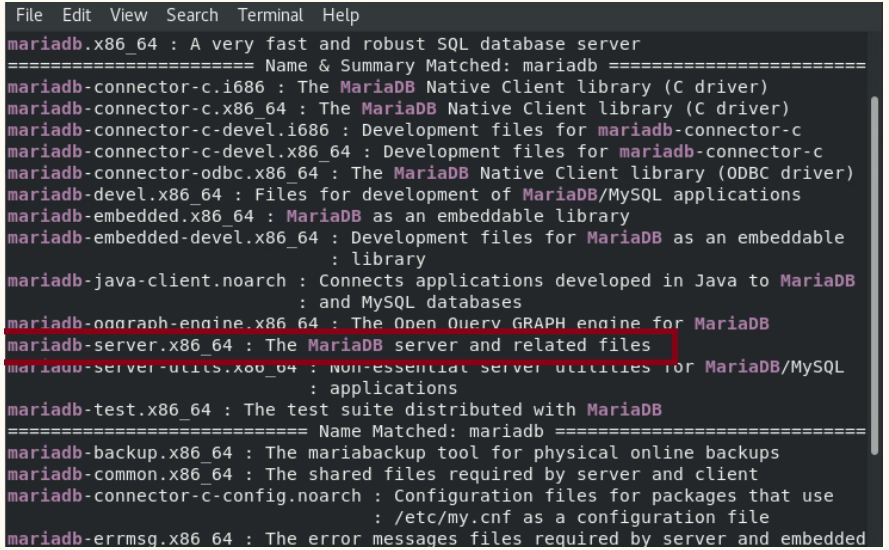


Mariadb installation

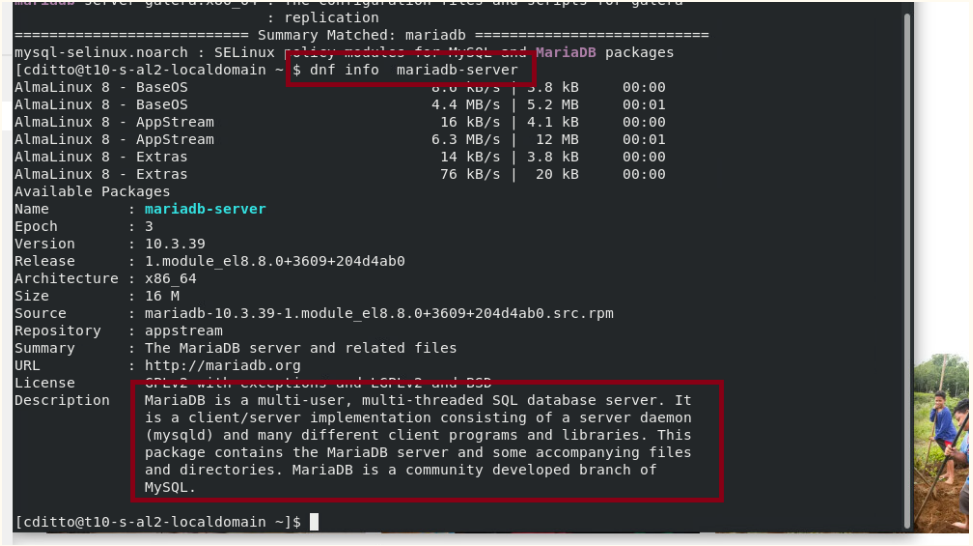
First search to see what options we have as mariadb. Mariadb-server is the one we will use.

```
dnf search mariadb
```

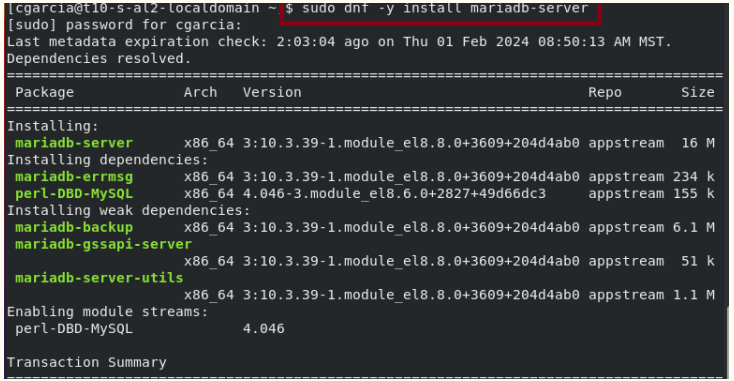




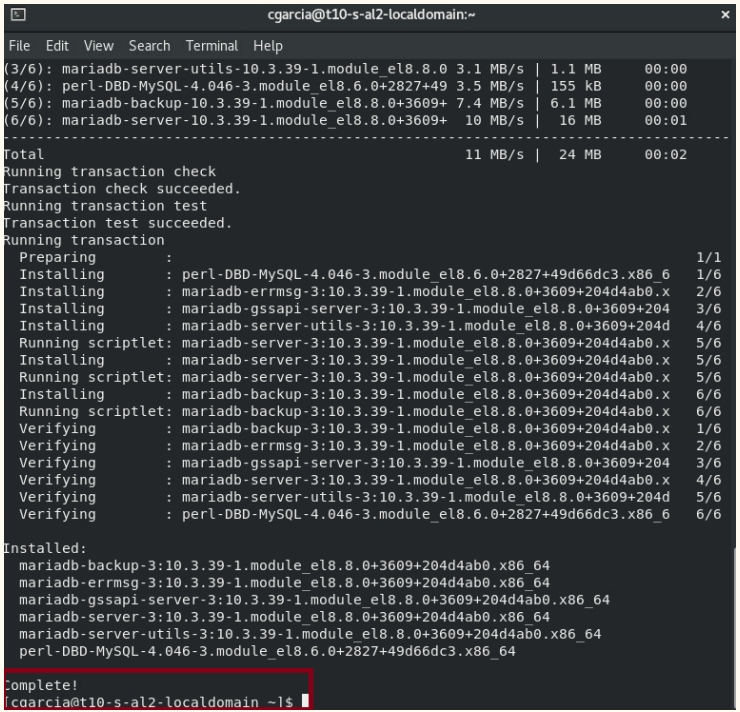
We can see his info and in the second command install it.




```
dnf info mariadb

sudo dnf -y install mariadb-server
```

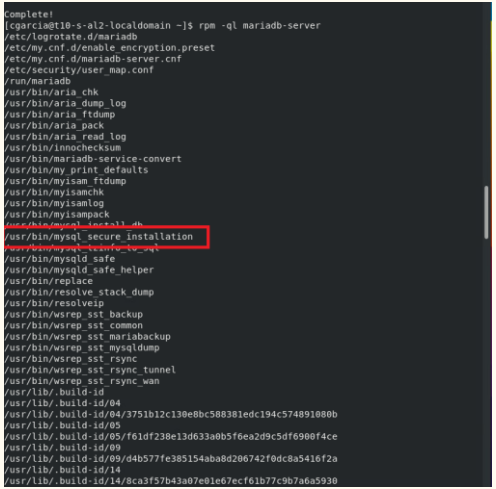


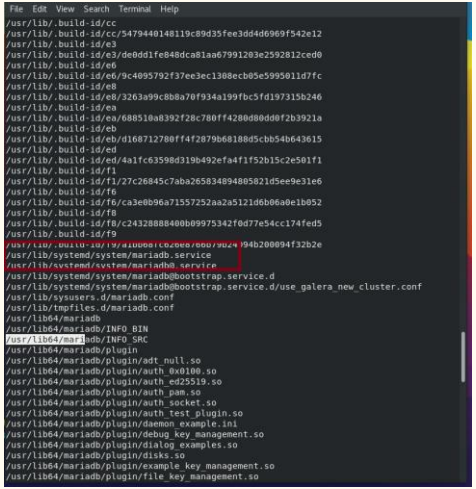
Installation is complete, check for the following files in mariadb package using rpm -ql mariadb-server



/usr/lib/systemd/system/mariadb.service

/user/bin/mysql_secure_installation

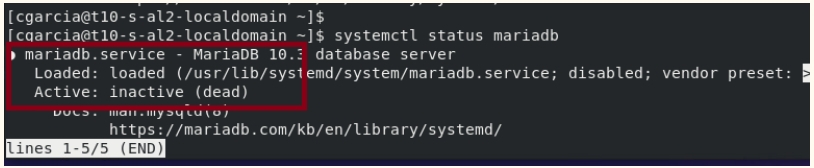


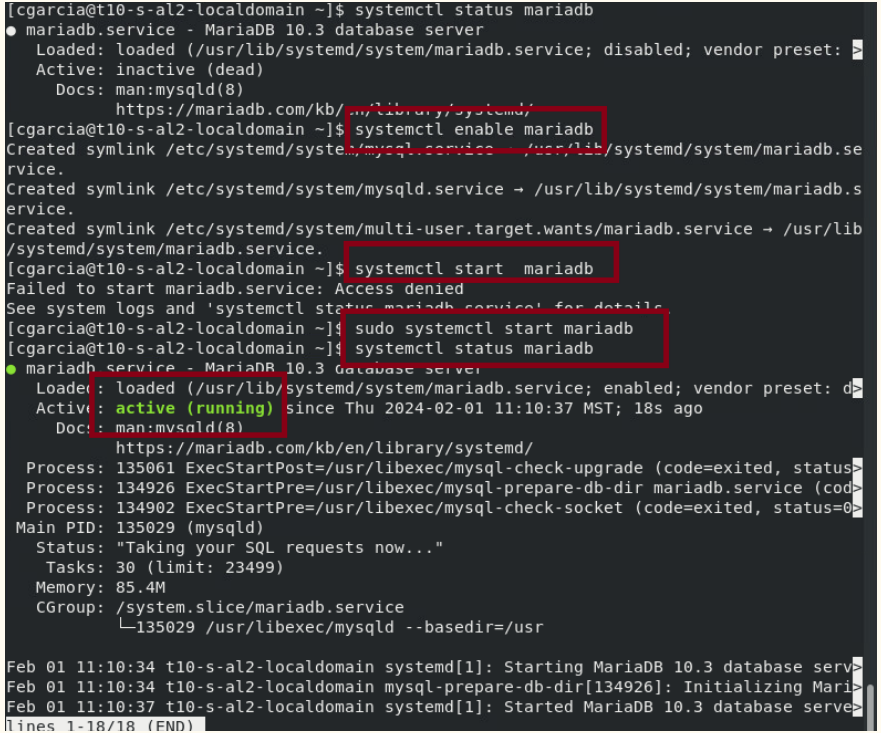


Check status and enable/start the service

Use:

```
systemctl status mariadb

sudo systemctl enable mariadb

sudo systemctl start mariadb
```

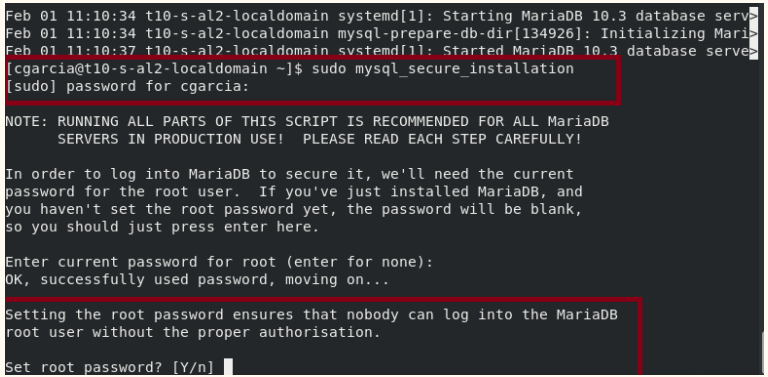




Use my_sql_secure installation to initialize security settings:

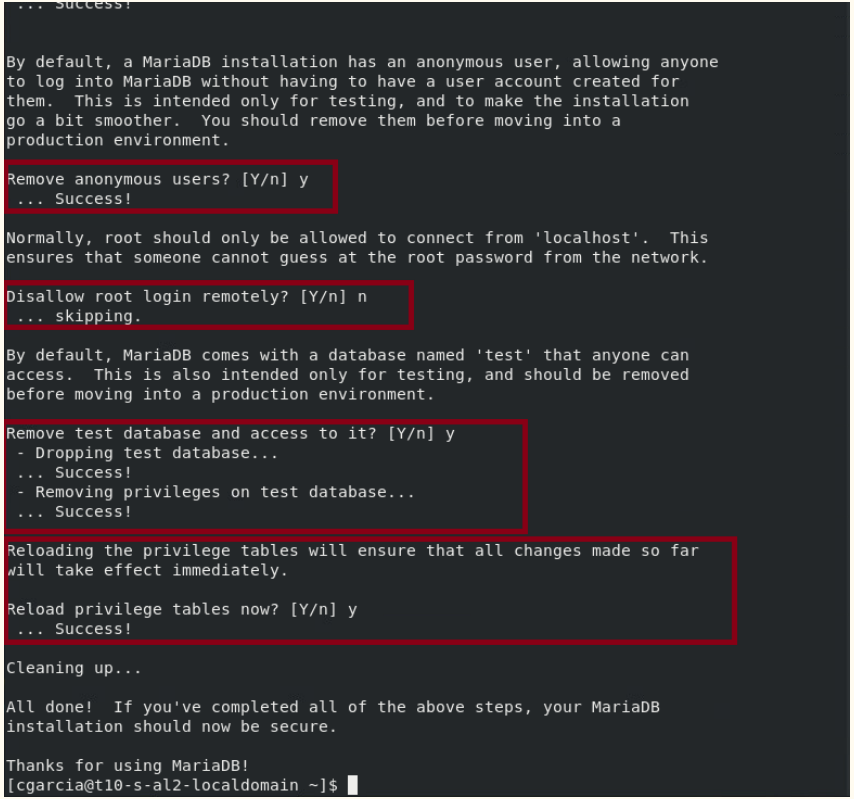
```
sudo mysql_secure_installation
```

Answer the prompts from the interactive script.



Use my_sql_secure installation to initialize security settings:

Answer the prompts from the interactive script.

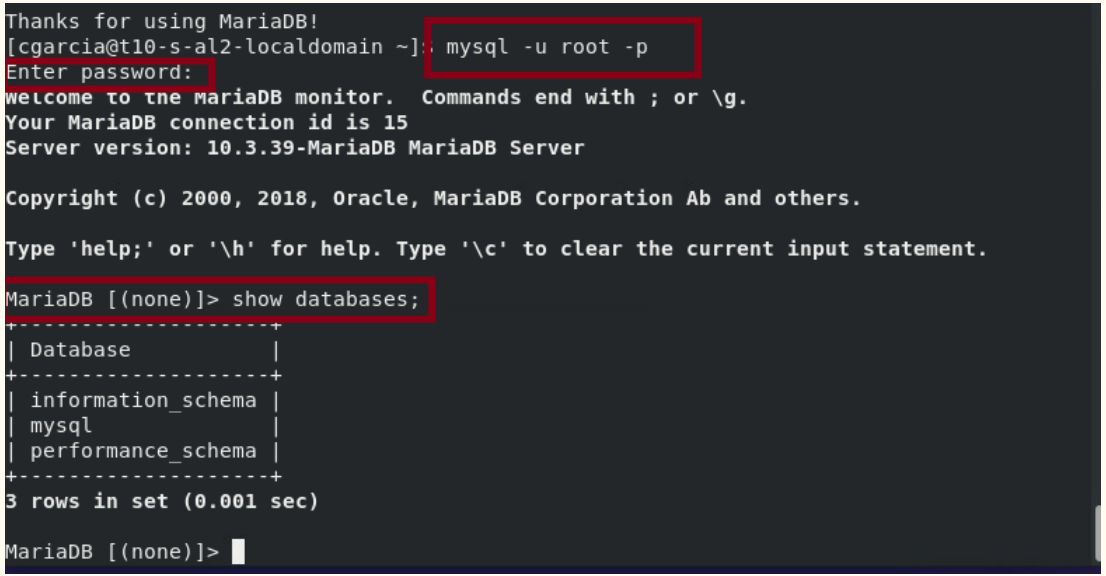


Connect to the database server with the MariaDB root account. (The necessary -p command-line option tells the client to [p]rompt you for the password.)

```
mysql -u root -p
```

List the available databases, remember the “;” :

```
show databases;
```



Create a new database call q2a and create q2auser with passwords and privileges.

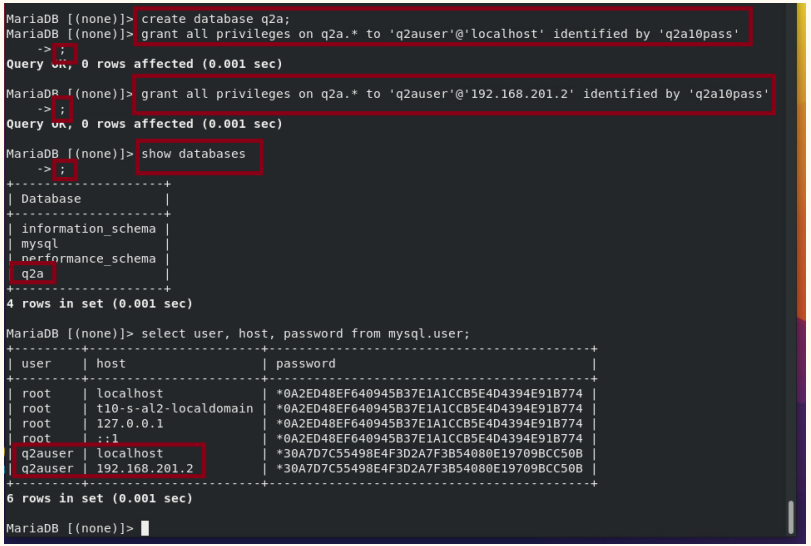
```
create database q2a;

grant all privileges on q2a.* to
'q2auser'@'localhost' identified by
'q2apass';
```

User from DMZ zone:

```
grant all privileges on q2a.* to
'q2auser'@'192.168.201.2' identified by
'q2apass';
```

See results:

```
show databases;

select user, host, password from mysql.user;
```

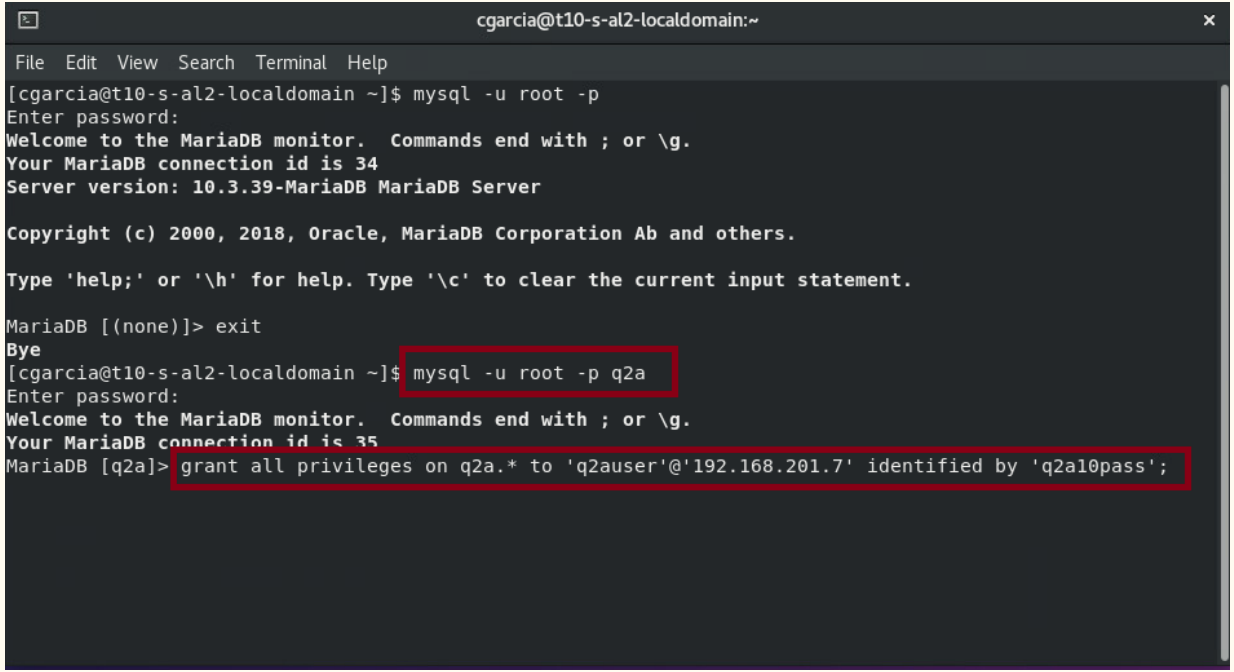


Create another user in the right machine.

```
mysql -u root -p q2a
```

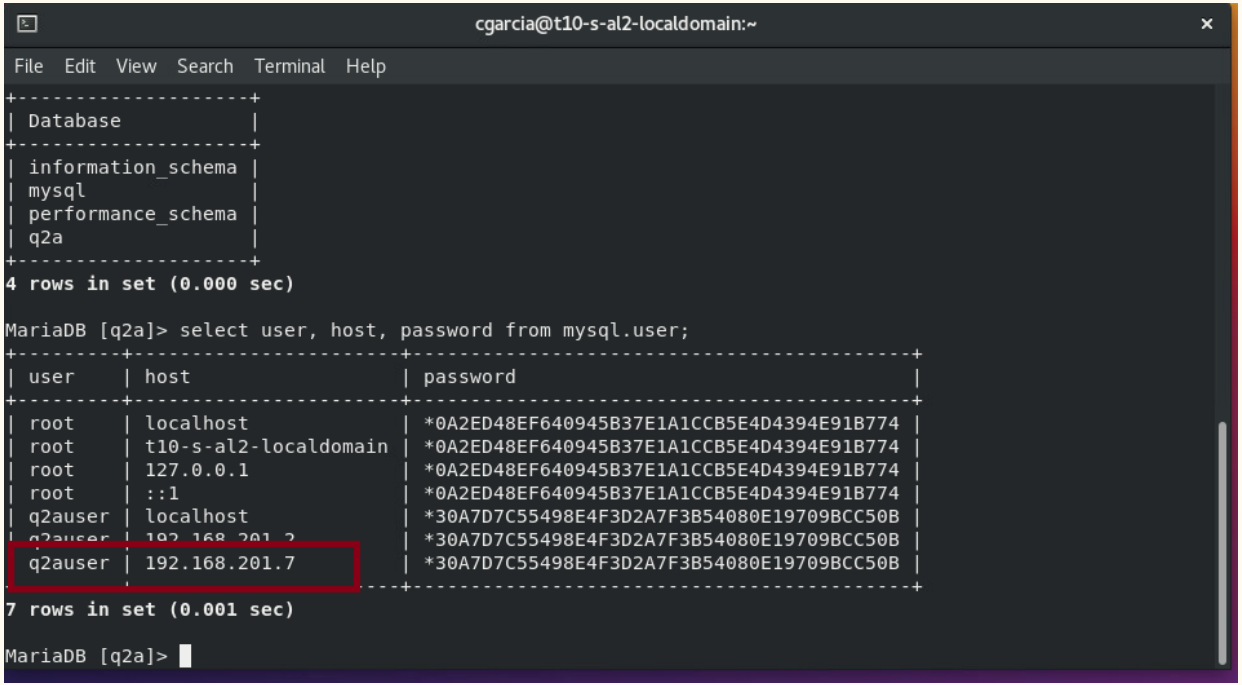
```
grant all privileges on q2a.*
to ‘q2auser’@’192.168.201.7’
Identified by ‘q2a10pass’;
```

This shows how to create new users and Add privileges to connect from other machines.



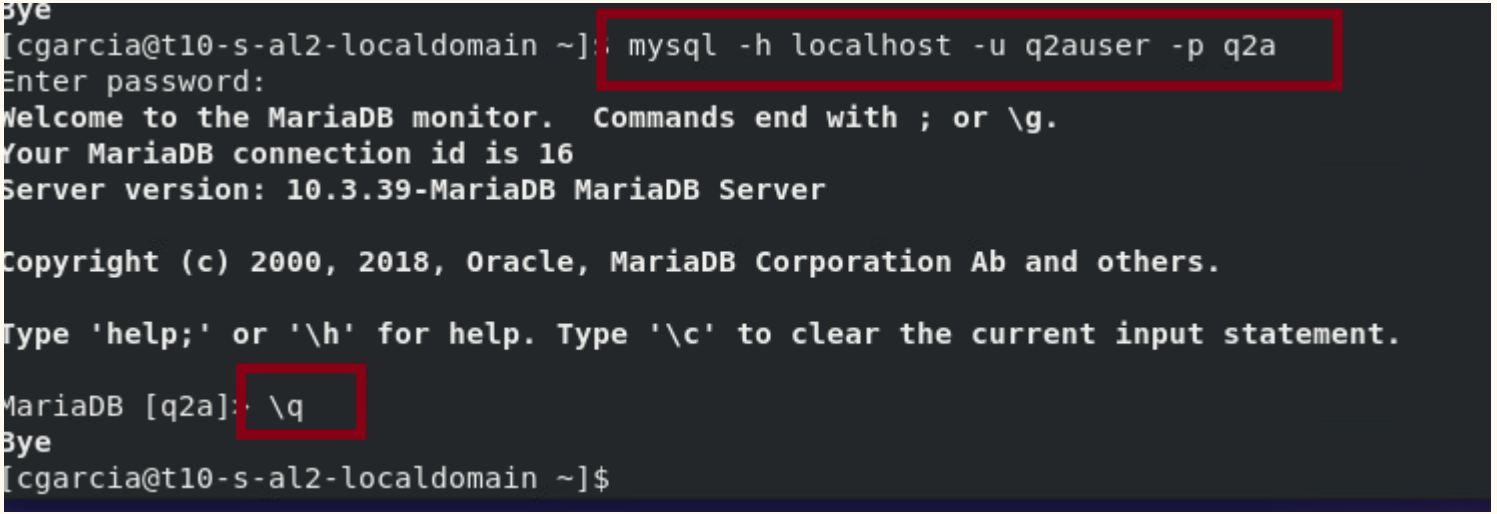
The new user created previously shows in the database.

```
select user, host, password from mysql.user;
```



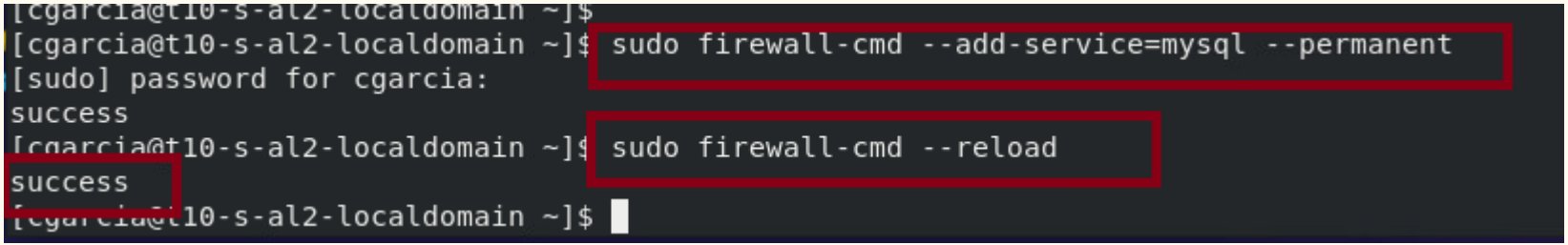
Connect locally to the q2a database as q2auser

```
mysql -h localhost -u q2auser -p q2a
```



The following adjustments will allow other hosts to connect to MariaDB:

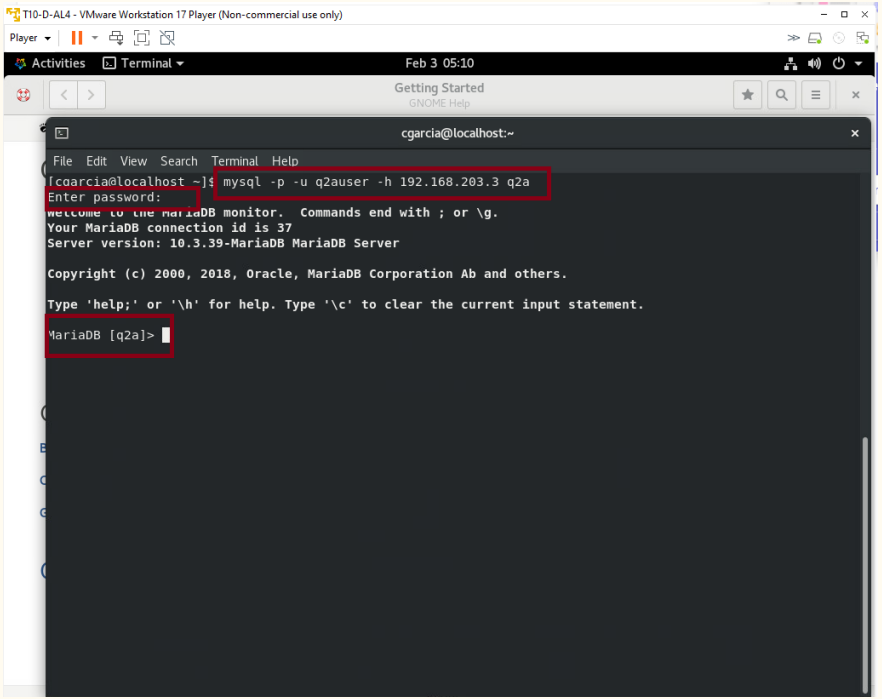
```
sudo firewall-cmd --add-service=mysql --permanent

sudo firewall-cmd --reload
```



Remote connection from the DMZ machine where the apache server resides.

```
mysql -p -u q2auser -h 192.168.203.3 q2a
```

Description:
-p ask for password
-u username (q2auser)
-h host ip address(192.168.203.3)
(q2a) the name of the database



Apache server installation

Install Apache Server

Download, install, and initialize Apache HTTPD server software.

```
[cditto@localhost ~]$ dnf search apache
```

```
[cditto@localhost ~]$ sudo dnf -y install httpd
```

```
Installed:
  almalinux-logos-httpd-84.5-1.el8.noarch
  apr-1.6.3-12.el8.x86_64
  apr-util-1.6.1-9.el8.x86_64
  apr-util-bdb-1.6.1-9.el8.x86_64
  apr-util-openssl-1.6.1-9.el8.x86_64
  httpd-2.4.37-62.module_el8.9.0+3646+acd210d0.x86_64
  httpd-filesystem-2.4.37-62.module_el8.9.0+3646+acd210d0.noarch
  httpd-tools-2.4.37-62.module_el8.9.0+3646+acd210d0.x86_64
  mod_http2-1.15.7-8.module_el8.9.0+3660+29a7abf6.3.x86_64

Complete!
```

Install Apache Server

Backup a copy of the primary configuration file /etc/httpd/conf/httpd.conf

```
cditto@localhost ~]$ cd /etc/httpd/conf
```

Change configuration file by changing the Listen directive to attach the service to the virtual ethernet. To do this open nano and change the Listener directive to the Host server IP address and port 80

```
[cditto@localhost ~]$ sudo nano /etc/httpd/conf/httpd.conf
```

```
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 192.168.201.7:80
```

```
[cditto@localhost conf]$ sudo systemctl restart httpd
```

Install Apache Server

Check the server status.

Notice it is listening to the IP address and port we configured earlier.

```
[cditto@localhost conf]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor pres>
   Active: active (running) since Fri 2024-02-02 12:53:29 MST; 5min ago
     Docs: man:httpd.service(8)
 Main PID: 42400 (httpd)
   Status: "Running, listening on: 192.168.201.7 port 80"
    Tasks: 213 (limit: 23499)
   Memory: 29.1M
   CGroup: /system.slice/httpd.service
           ├─42400 /usr/sbin/httpd -DFOREGROUND
           ├─42401 /usr/sbin/httpd -DFOREGROUND
           ├─42402 /usr/sbin/httpd -DFOREGROUND
           ├─42403 /usr/sbin/httpd -DFOREGROUND
           └─42404 /usr/sbin/httpd -DFOREGROUND

Feb 02 12:53:29 localhost.localdomain systemd[1]: Starting The Apache HTTP Serv>
Feb 02 12:53:29 localhost.localdomain httpd[42400]: AH00558: httpd: Could not r>
Feb 02 12:53:29 localhost.localdomain systemd[1]: Started The Apache HTTP Serve>
Feb 02 12:53:29 localhost.localdomain httpd[42400]: Server configured, listenin>
```

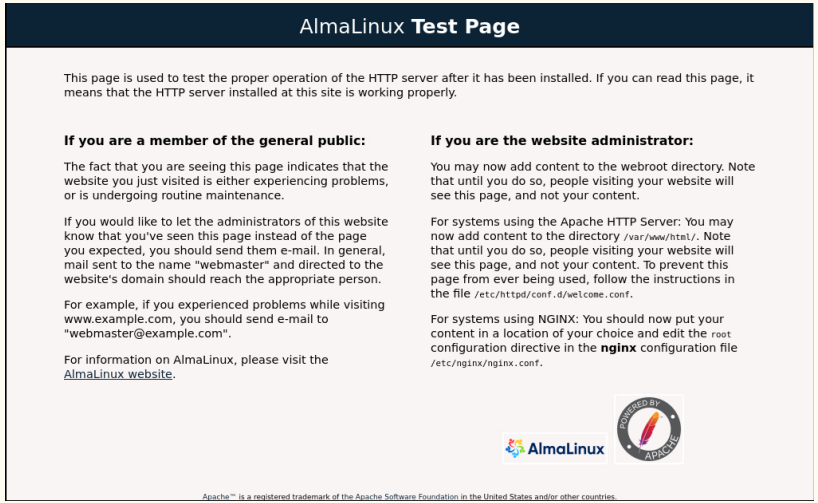
Install Apache Server

Enable service to start at startup.

```
[cditto@localhost conf]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
```

Open browser and surf to the host servers webpage. You should see the new installation “Test Page”

```
[cditto@localhost conf]$ sudo systemctl start httpd
```



Download & Install PHP

Download and install PHP.

Restart Apache.

```
[cditto@localhost ~]$ sudo dnf install php
```

```
Installed:
  nginx-filesystem-1:1.14.1-9.module_el8.3.0+2165+af250afe.alma.noarch
  php-7.2.24-1.module_el8.3.0+2010+7c76a223.x86_64
  php-cli-7.2.24-1.module_el8.3.0+2010+7c76a223.x86_64
  php-common-7.2.24-1.module_el8.3.0+2010+7c76a223.x86_64
  php-fpm-7.2.24-1.module_el8.3.0+2010+7c76a223.x86_64

Complete!
```

```
[cditto@localhost ~]$ sudo systemctl restart httpd
```

Download & Install PHP

Check Apache server status.

Notice php-fpm is activated without having to enable it.

```
[cditto@localhost ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
  Drop-In: /usr/lib/systemd/system/httpd.service.d
           └─php-fpm.conf
   Active: active (running) since Fri 2024-02-02 13:47:02 MST; 27s ago
     Docs: man:httpd.service(8)
 Main PID: 45585 (httpd)
   Status: "Running, listening on: 192.168.201.7 port 80"
    Tasks: 213 (limit: 23499)
   Memory: 37.0M
   CGroup: /system.slice/httpd.service
           ├─45585 /usr/sbin/httpd -DFOREGROUND
           ├─45593 /usr/sbin/httpd -DFOREGROUND
           ├─45595 /usr/sbin/httpd -DFOREGROUND
           ├─45596 /usr/sbin/httpd -DFOREGROUND
           └─45598 /usr/sbin/httpd -DFOREGROUND

Feb 02 13:47:02 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
Feb 02 13:47:02 localhost.localdomain httpd[45585]: AH00558: httpd: Could not reliably determine the server's>
Feb 02 13:47:02 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
Feb 02 13:47:02 localhost.localdomain httpd[45585]: Server configured, listening on: 192.168.201.7 port 80
```

Download & Install PHP

Check php-fpm status.

In a later assignment Q2A will specify that we need to add the extension MySQLi but when we search for a MySQL extension we only find MySQLnd in Alma Linux.

```
[cditto@localhost ~]$ sudo systemctl status php-fpm
● php-fpm.service - The PHP FastCGI Process Manager
   Loaded: loaded (/usr/lib/systemd/system/php-fpm.service; disabled; vendor preset: disabled)
   Active: active (running) since Fri 2024-02-02 13:47:01 MST; 11min ago
 Main PID: 45577 (php-fpm)
   Status: "Processes active: 0, idle: 5, Requests: 0, slow: 0, Traffic: 0req/sec"
    Tasks: 6 (limit: 23499)
   Memory: 9.0M
   CGroup: /system.slice/php-fpm.service
           ├─45577 php-fpm: master process (/etc/php-fpm.conf)
           ├─45578 php-fpm: pool www
           ├─45579 php-fpm: pool www
           ├─45580 php-fpm: pool www
           ├─45581 php-fpm: pool www
           └─45582 php-fpm: pool www

Feb 02 13:47:01 localhost.localdomain systemd[1]: Starting The PHP FastCGI Process Manager...
Feb 02 13:47:01 localhost.localdomain systemd[1]: Started The PHP FastCGI Process Manager.
```

```
[cditto@localhost ~]$ dnf search php-mysql
```

```
php-mysqlnd.x86_64 : A module for PHP applications that use MySQL databases
[cditto@localhost ~]$ dnf info php-mysqlnd
Last metadata expiration check: 2:34:23 ago on Fri 02 Feb 2024 11:28:18 AM MST.
Available Packages
```

Download & Install PHP

MySQLnd is an updated extension from MySQLi. It is fine to use it in this installation. Install this package.

```
[cditto@localhost ~]$ sudo dnf -y install php-mysqlnd
```

```
Installed:
  php-mysqlnd-7.2.24-1.module_el8.3.0+2010+7c76a223.x86_64
  php-pdo-7.2.24-1.module_el8.3.0+2010+7c76a223.x86_64

Complete!
[cditto@localhost ~]$
```

Test the PHP Installation

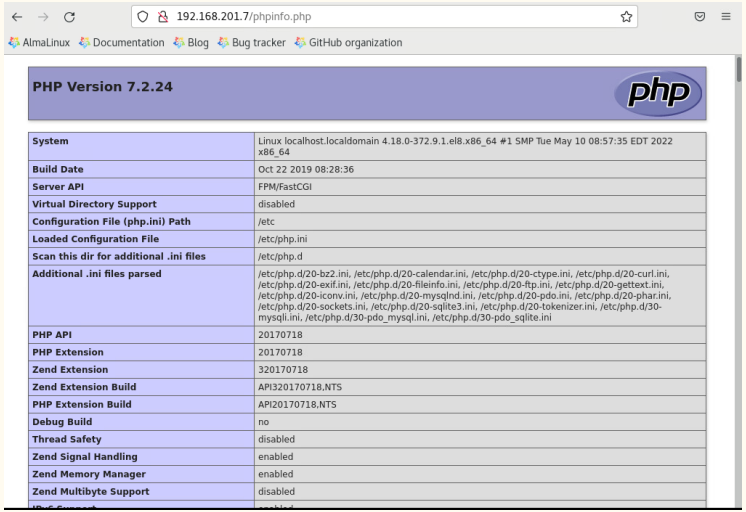
Create a PHP file in the Apache documentroot folder /var/www/html

Add the line <?php phpinfo(); ?> to the file.

Use the web browser to surf to that file, using the URL http://<DMZ server IP address>/phpinfo.php

```
[cditto@localhost ~]$ sudo nano /var/www/html/phpinfo.php
```

```
  GNU nano 2.9.8

<?php phpinfo(); ?>
```

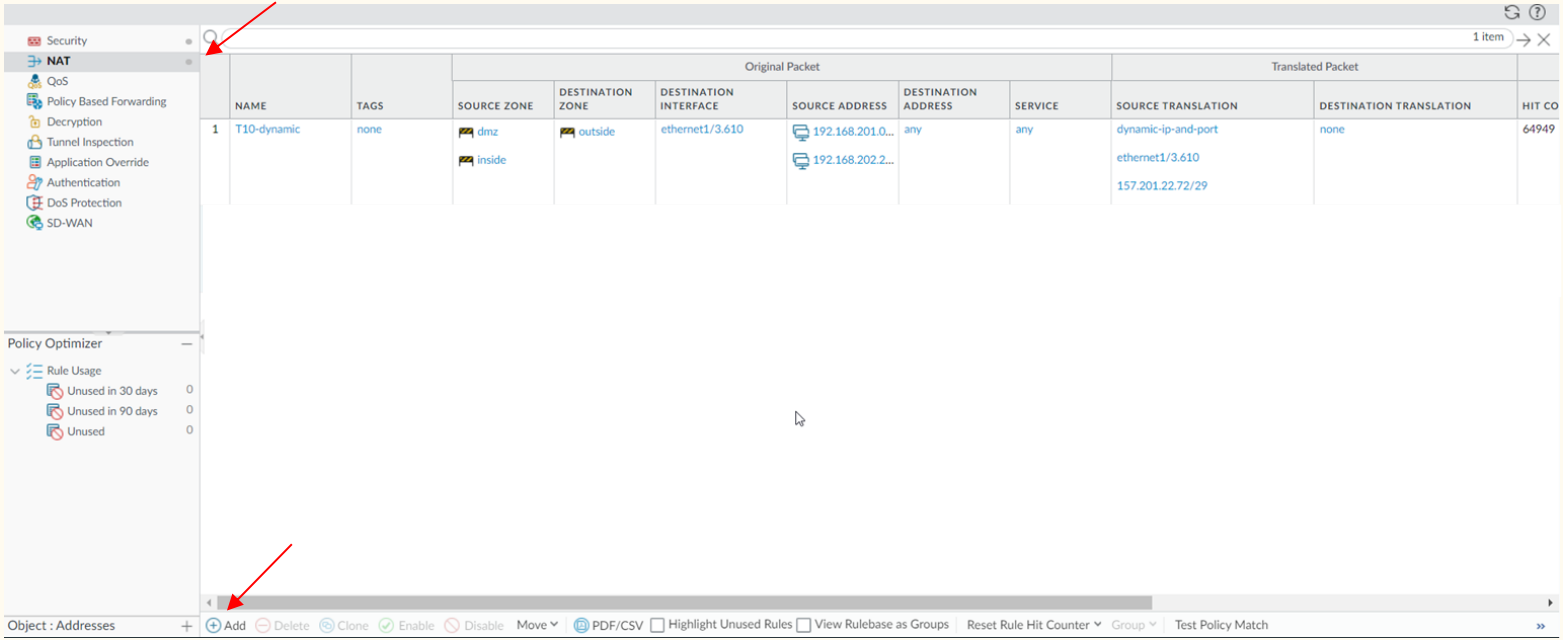


System	Linux localhost.localdomain 4.18.0-372.9.1.el8.x86_64 #1 SMP Tue May 10 08:57:35 EDT 2022 x86_64
Build Date	Oct 22 2019 08:28:36
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-mysqlnd.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled

Configure Policies on Internet Facing Firewalls

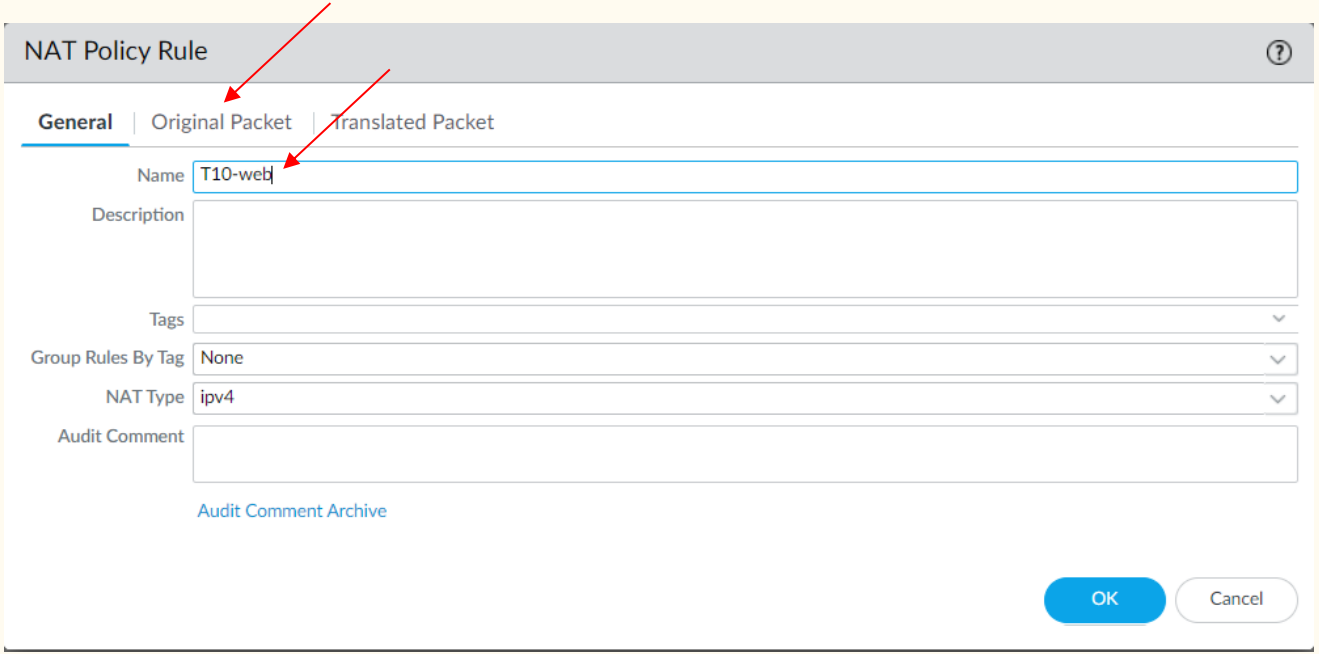
Configure Static NAT.

Under the “Policies” tab, select “NAT” and then select “ADD”



Configure Policies on Internet Facing Firewalls

Add a name to your NAT policy.



Configure Policies on Internet Facing Firewalls

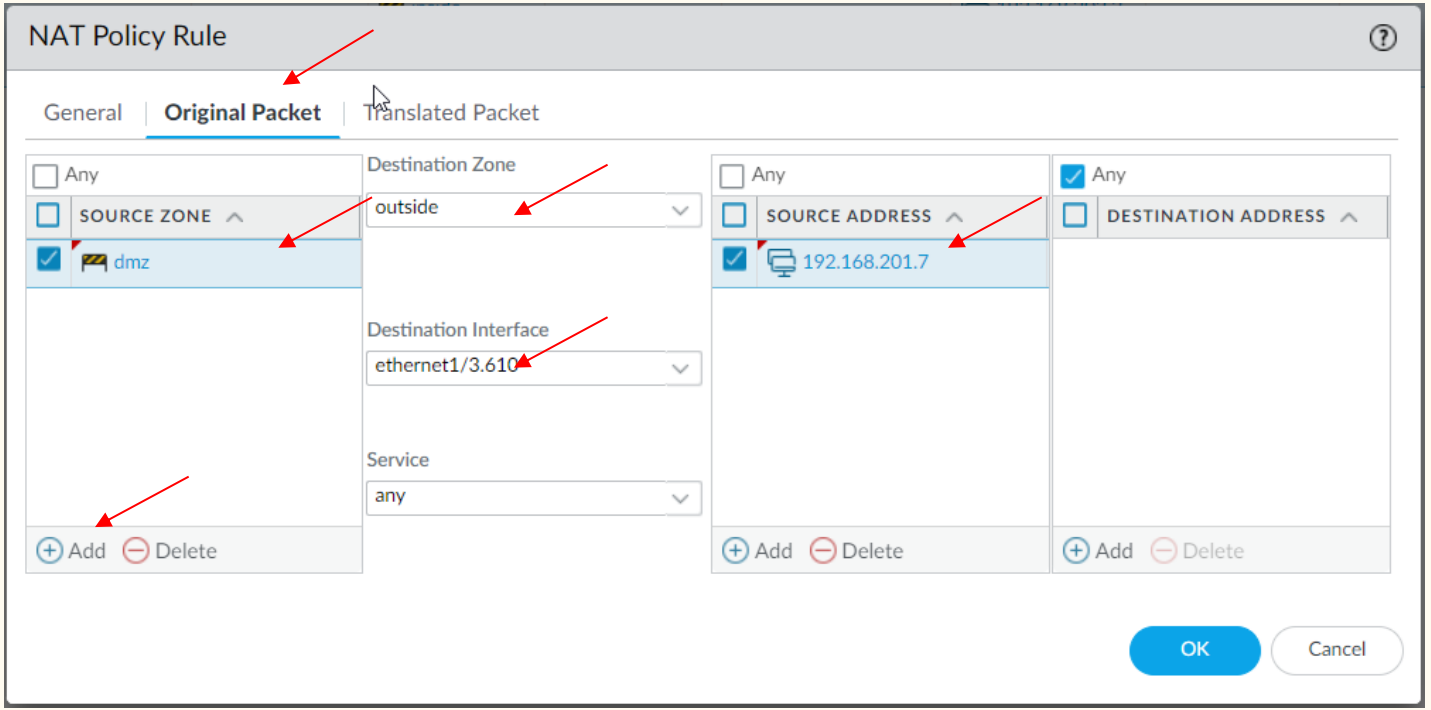
Select the “Original Packet” tab.

Change the “Source Zone” to DMZ.

Select the dropdown in the “Destination Zone” Select outside.

Select the dropdown for the “Destination Interface” then select the VPN interface in the outside zone.

Under “Source Address” add the IP address of the apache server.



Configure Policies on Internet Facing Firewalls

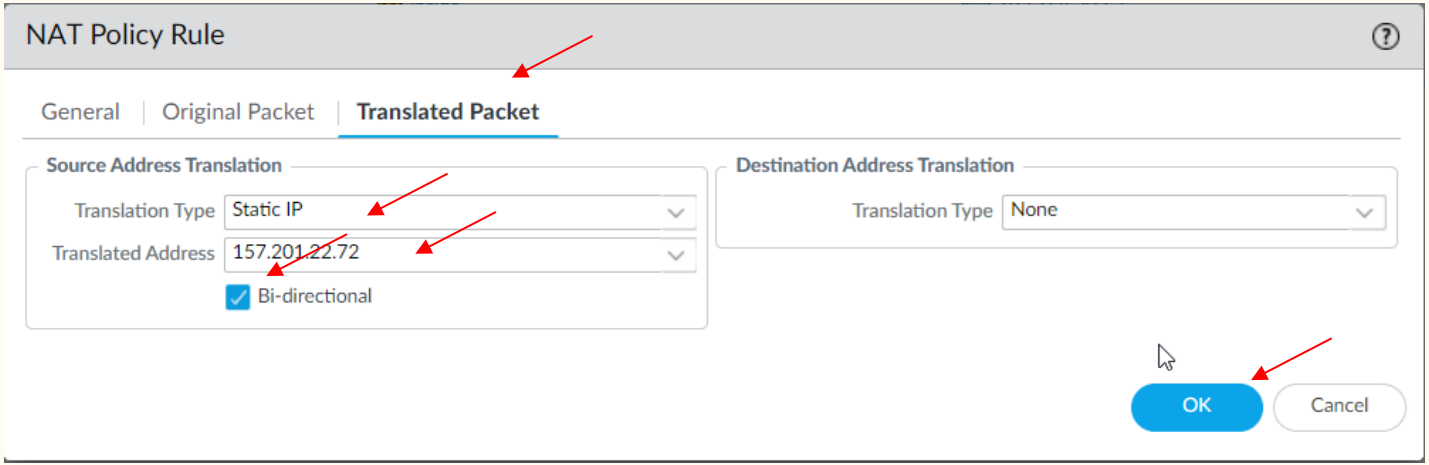
Select the “Translated Packet” tab.

Under “Translation Type” select the dropdown and then Static IP.

Under the “Translated Address” enter the public IP address.

Check th “Bi-directional box” so the rule translates both ways.

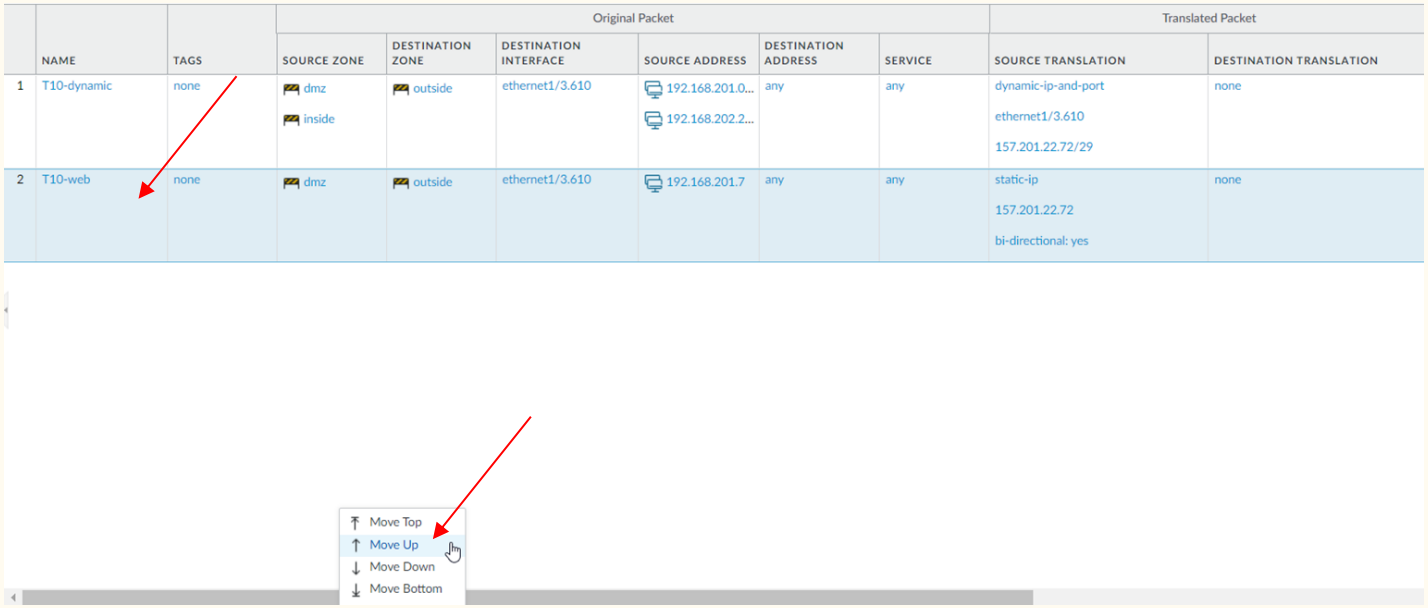
Click “OK”



Configure Policies on Internet Facing Firewalls

Highlight the new rule and select “move up” from the “Move” interface at the bottom of the web-interface.

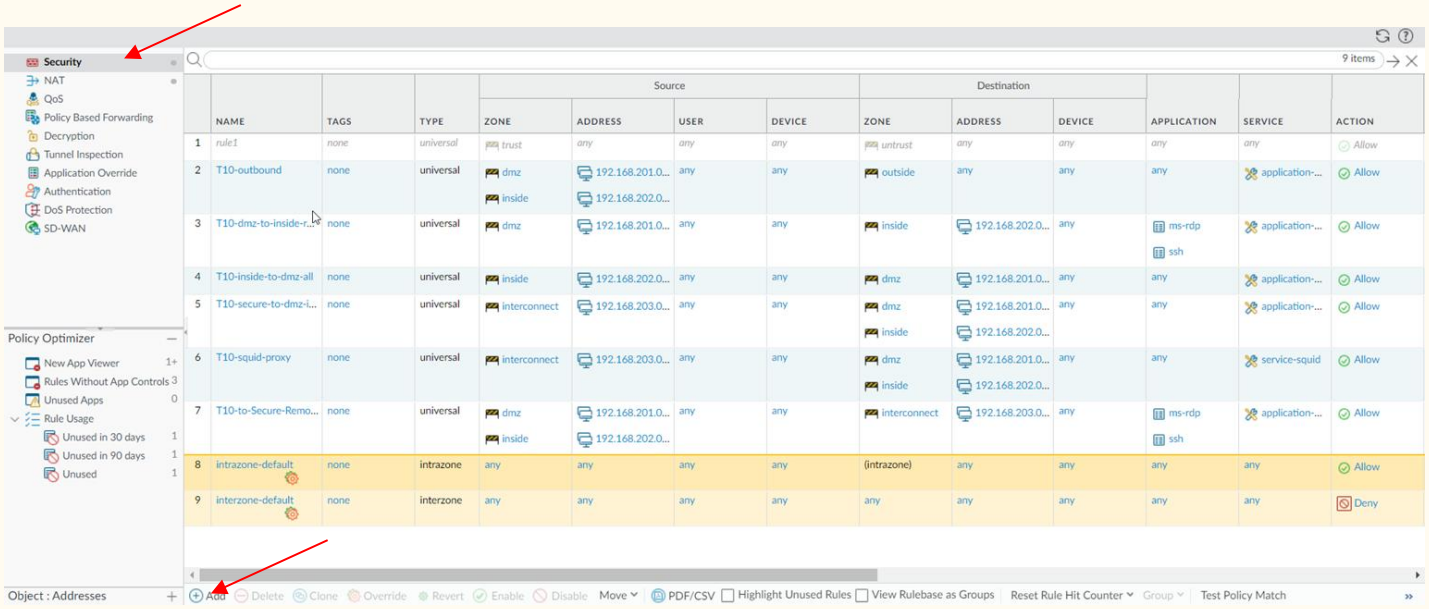
The rule needs to be before the team’s Dynamic rule.



Configure Policies on Internet Facing Firewalls

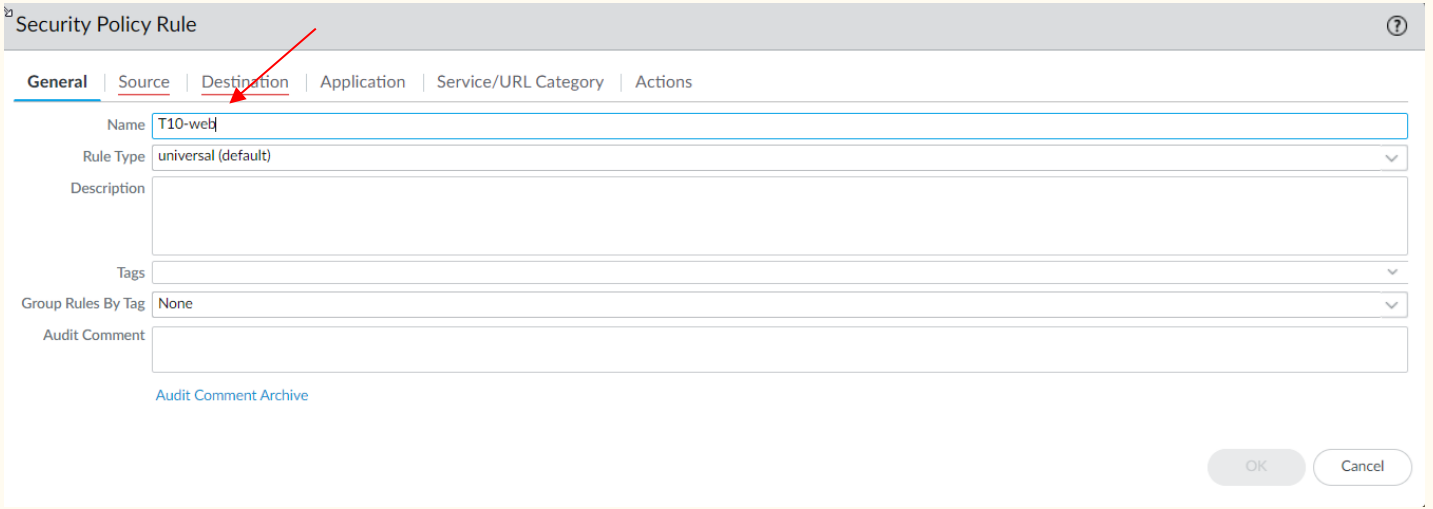
Configure a web-server security policy.

Select Security on the left just above NAT then click on Add.



Configure Policies on Internet Facing Firewalls

Under General add the team name followed by “web” under the “Name” selection.



Configure Policies on Internet Facing Firewalls

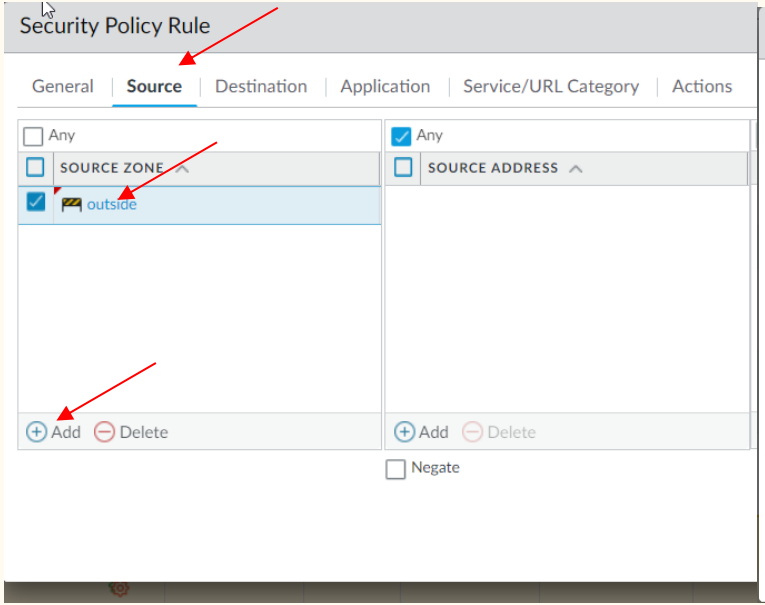
Select the “Source” tab.

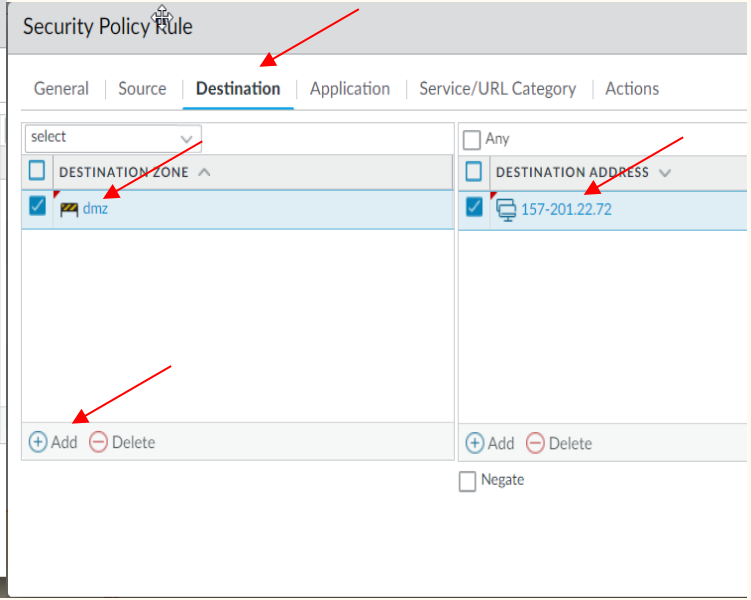
Under the “Source Zone” click on “Add” then select “outside”.

Select the “Destination” tab

Under the “Destination Zone” Select “Add” then select outside.

Under “Destination Address” add the public IP address of the web server.



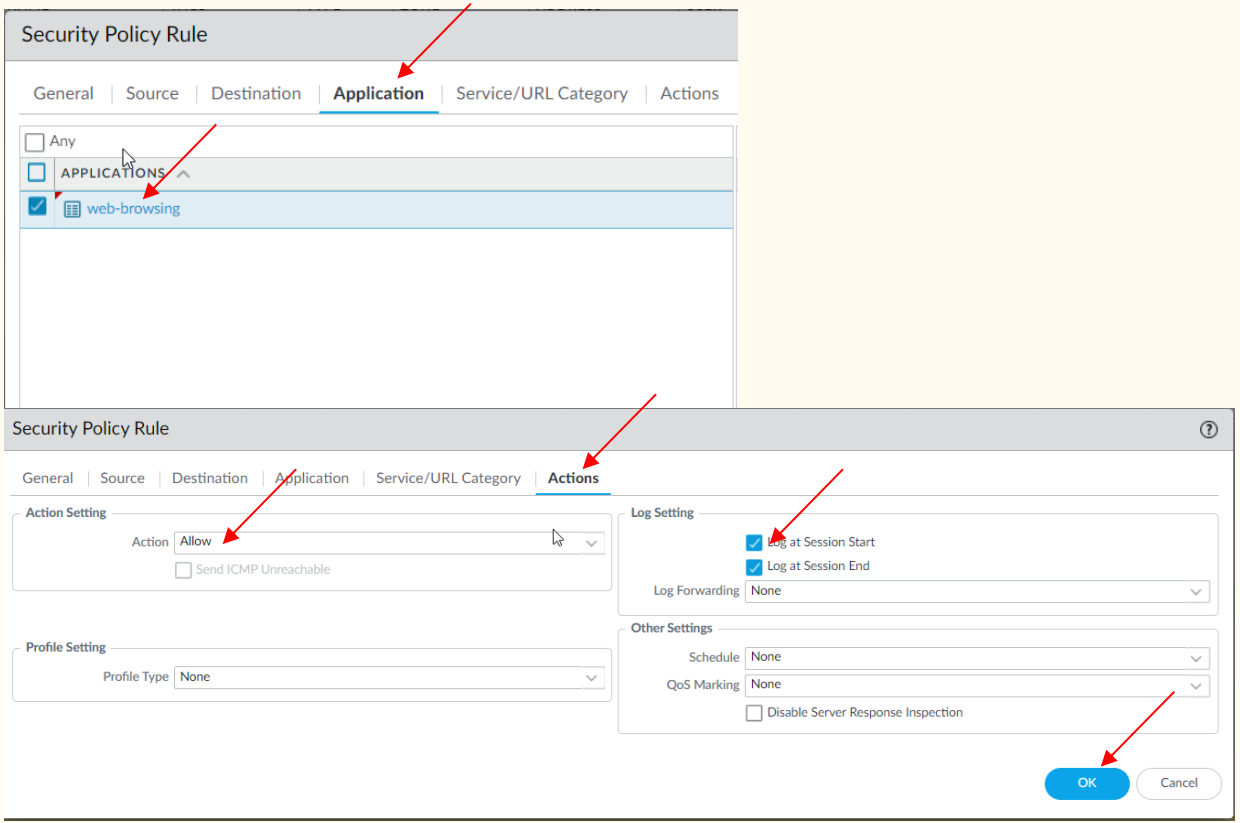


Configure Policies on Internet Facing Firewalls

Select the “Applications” tab, and specify web-browsing.

Select the “Actions” tab

Commit your changes.



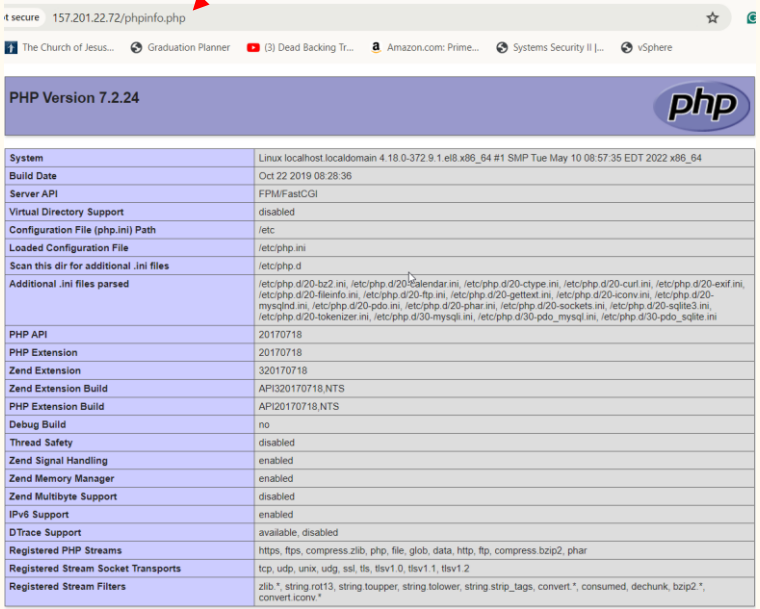
Configure Policies on Internet Facing Firewalls

Configure Apache server to accept HTTP and HTTPS requests through the firewall.

Enter http://<web server's public IP address>/phpinfo.php.

You should see the same info page that you saw at the in of the PHP test section of this instruction sequence.

```
[cditto@localhost /]$ sudo firewall-cmd --zone=public --permanent --add-service=http
success
[cditto@localhost /]$ sudo firewall-cmd --zone=public --permanent --add-service=https
success
```



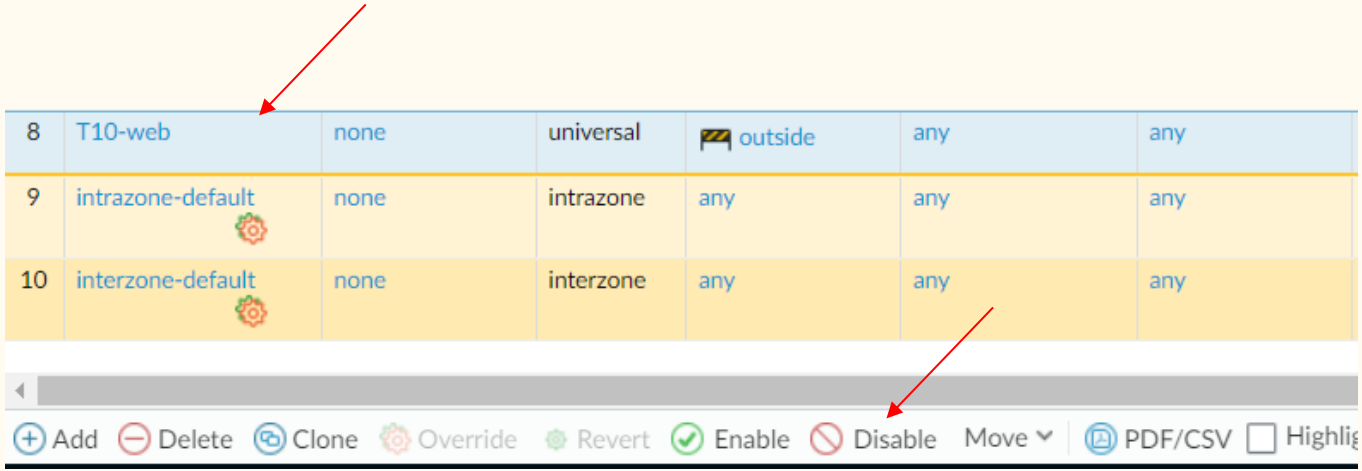
System	Linux localhost.localdomain 4.18.0-372.9.1.el8.x86_64 #1 SMP Tue May 10 08:57:35 EDT 2022 x86_64
Build Date	Oct 22 2019 08:28:36
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-mysqlnd.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, compress.bzip2, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, bzip2.*, convert.iconv.*

Configure Policies on Internet Facing Firewalls

Until the Question2Answer web app is installed and configured, it is a bad idea to leave the web server exposed to untrusted connections. Disable T10-Web rule until after the installation and configuration.

Highlight the T10-web rule then click “Disable” at the bottom of the web display.

Commit the change.



Q2A Internet-Facing Firewall Configuration

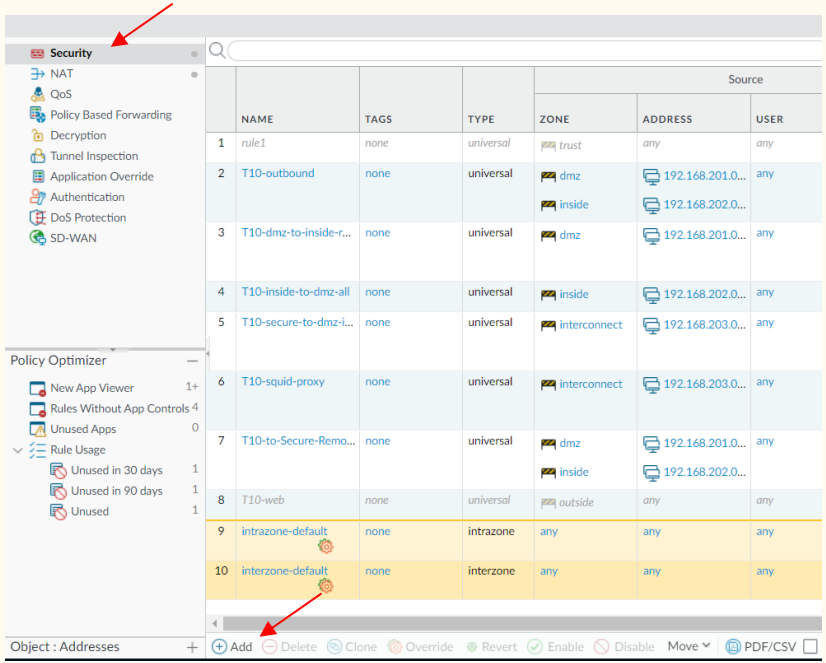
Check to make sure that qtauser can log into MariahDB.

This will result in an error until secure-facing firewall is configured.

Create a rule on the Palo Alto firewall that allows the application mysql from the web server in the DMZ through to the interconnect zone.

On the top left of the web display click on “Security” and then select “Add” at the bottom of the display

```
[cditto@t10-s-al2-localdomain ~]$ mysql -p -u q2auser -h 192.168.201.2
Enter password:
ERROR 2002 (HY000): Can't connect to MySQL server on '192.168.201.2' (115)
```



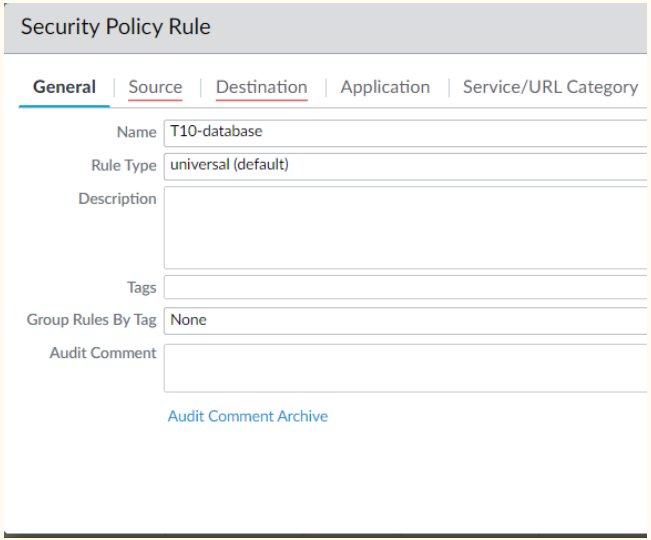
Q2A Internet-Facing Firewall Configuration

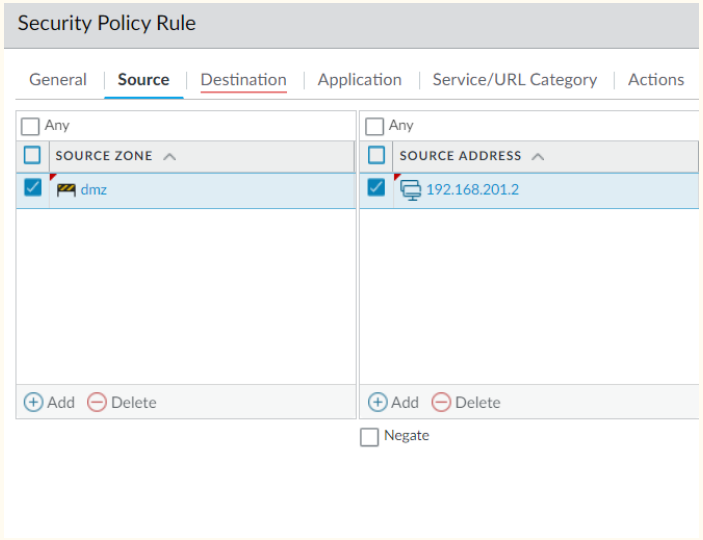
Under the General tab select Name and enter your (team's name)-database.

Select the "Source" tab.

Under Source Zone Click on "Add" and then select DMZ.

Under Source Address Select "Add" then add the address of your teams DMZ web server.



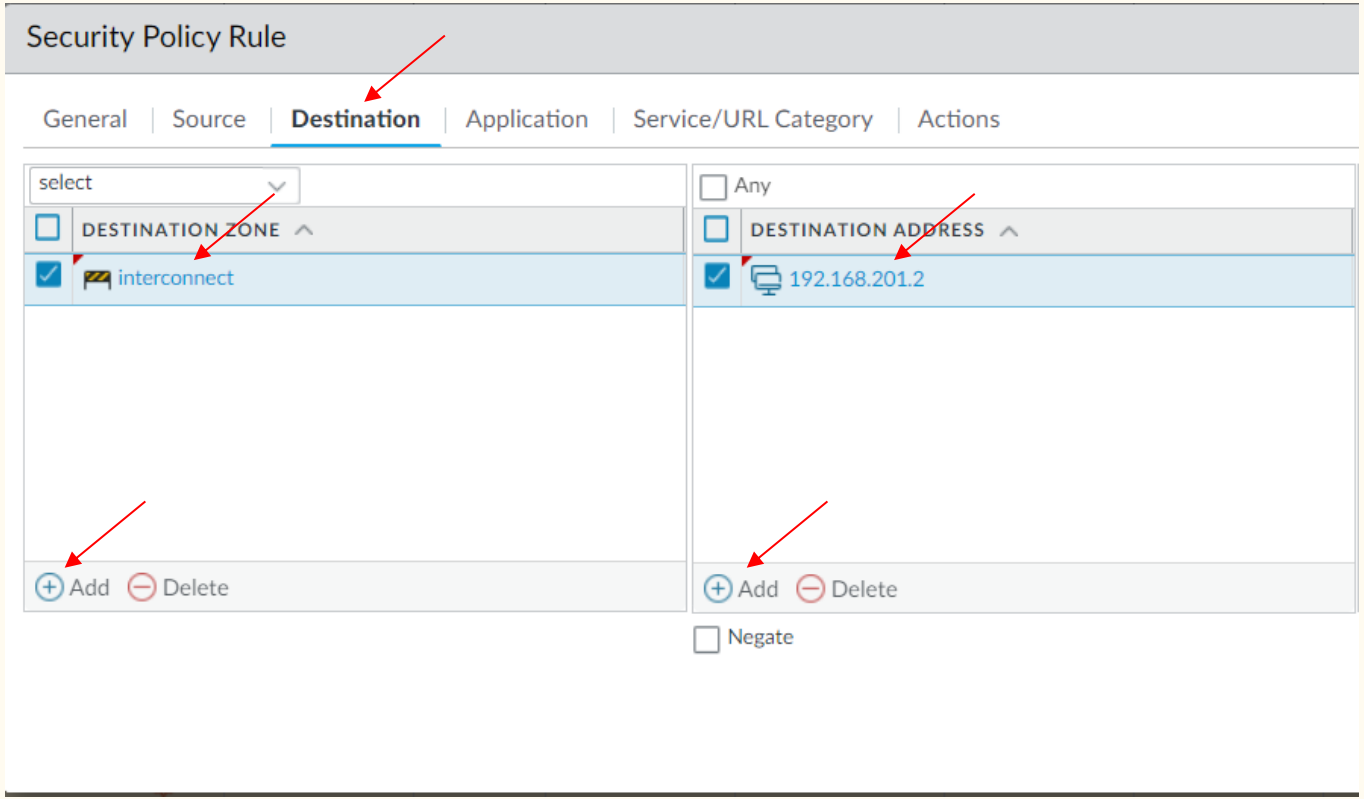


Q2A Internet-Facing Firewall Configuration

Select the “Destination” tab.

Under the Destination Zone, click “Add” then select “interconnect.

Under Destination Address, click “Add” then enter the IP address to your team’s DMZ web server.



Q2A Internet-Facing Firewall Configuration

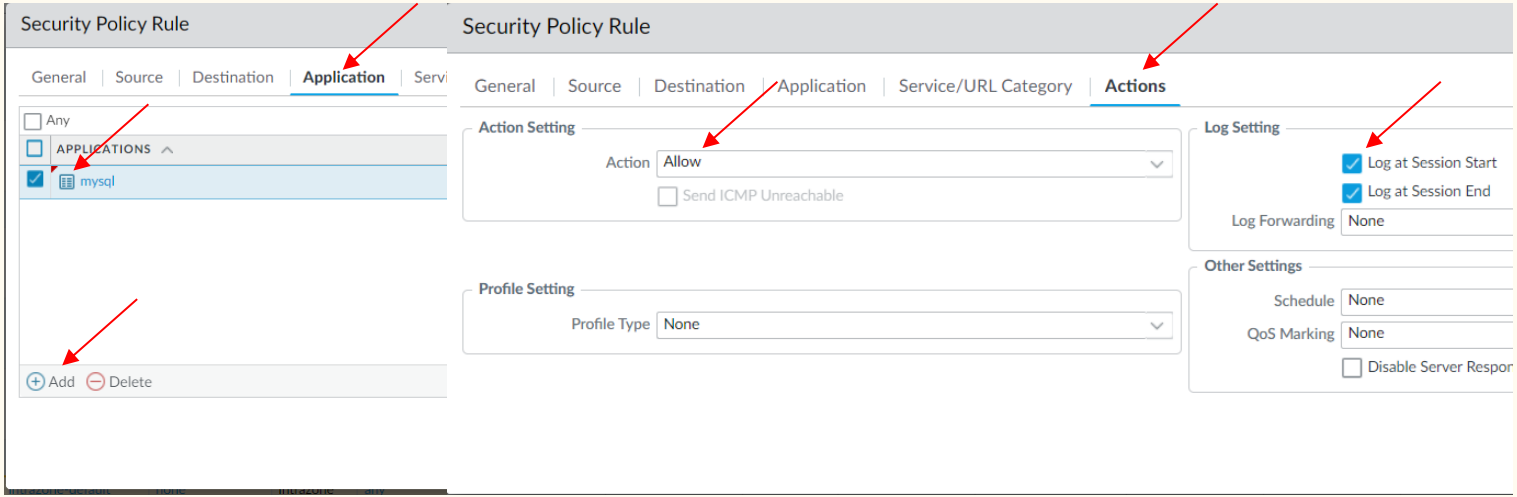
Select the Application tab.

Under “Applications”Click “Add” then select “mysql”.

Select the Actions tab.

Make sure the Action setting is set to “Allow” and check the “Log at Session Start check box.

Click OK and Commit your changes.



9	T10-database	none	universal	dmz	192.168.201.2	any	any	interconnect	192.168.201.2	any	mysql	application-...	Allow

Q2A Secure-Facing Firewall Configuration

Login to FortiGate.

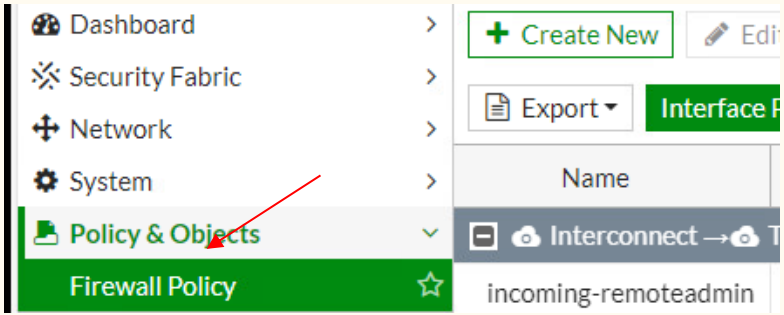
Select “Firewall Policy” from the right side menu.

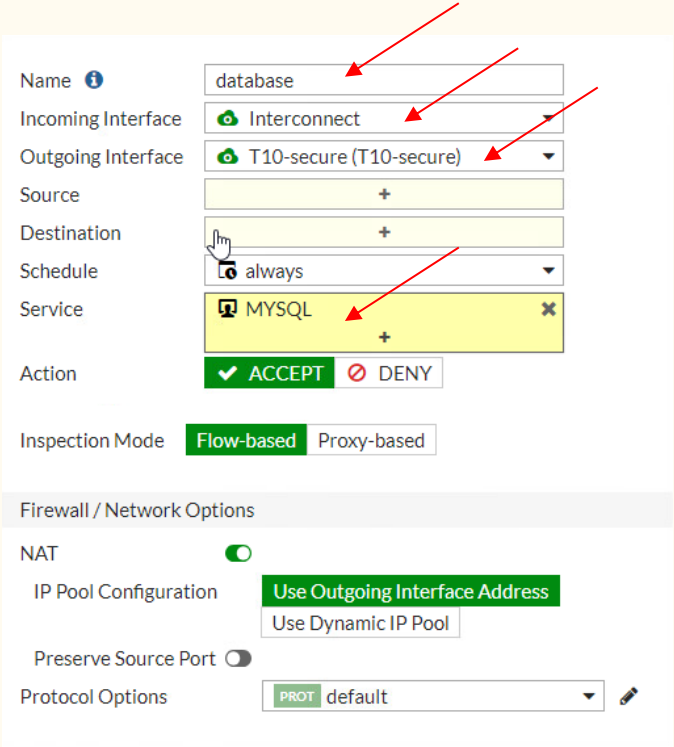
Click “Create New”.

Name the new rule “database”.

Under Incoming Interface select “Interconnect”

Under Outgoing Interface select (Team number)-secure Under Service select “mysql”.





Q2A Secure-Facing Firewall Configuration

For the Source and Destination selections you must create two new address objects.

Select the Source selection window then select “Create”.

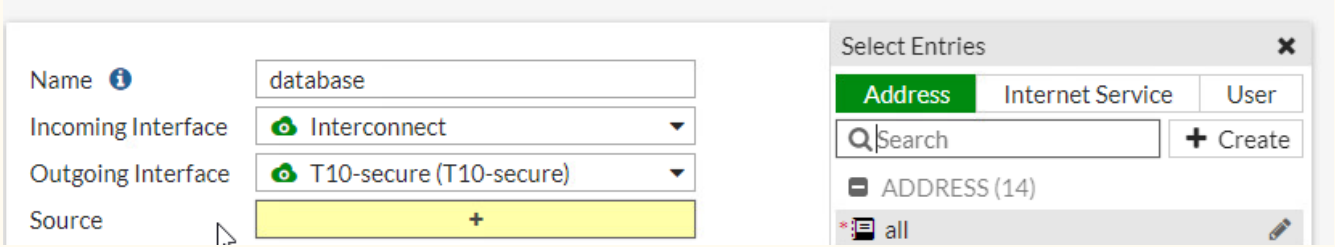
Select “Address”.

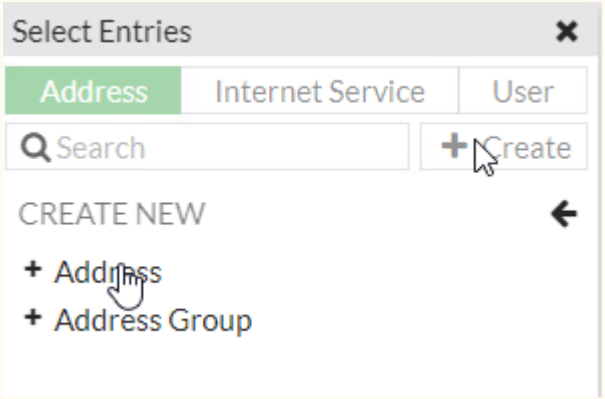
Enter “dmz-web” for the name.

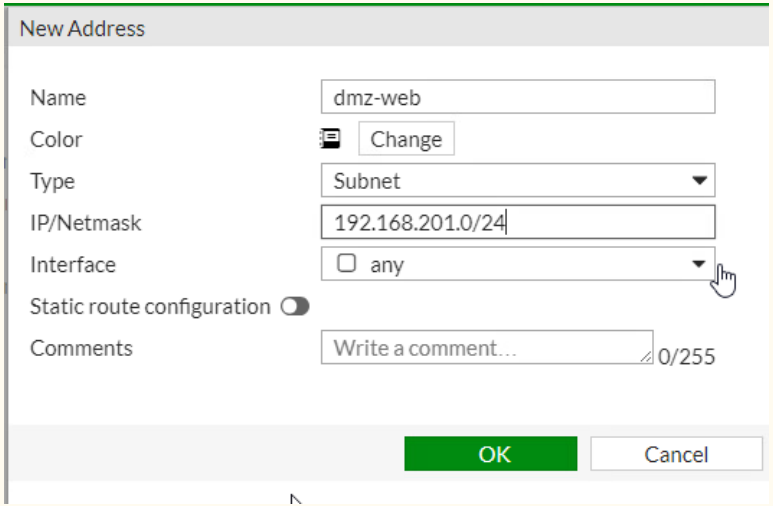
Enter the IP address of the team’s DMZ and subnet.

Click OK.

Repeat steps for the Destination selection.



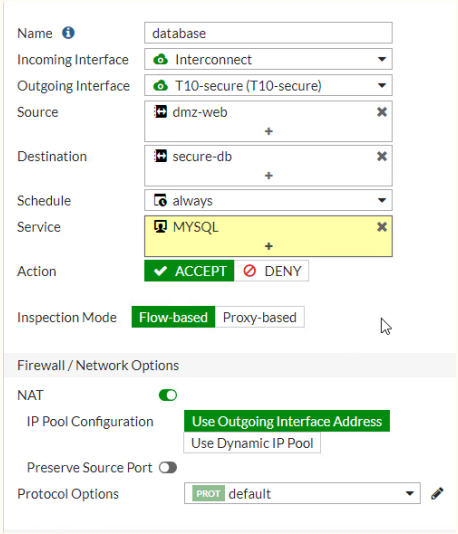




Q2A Secure-Facing Firewall Configuration

Click OK.

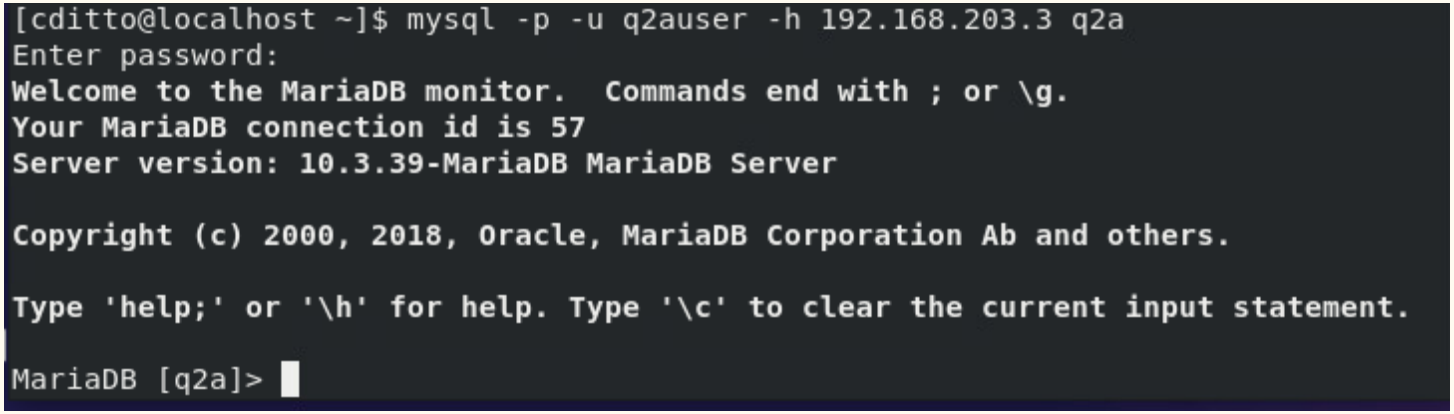
Disable NAT



incoming-remoteadmin	MZ Inside	Secure	always	RDP SSH	ACCEPT	Disabled
database	dmz-web	secure-db	always	MYSQL	ACCEPT	Disabled

Q2A Installation & Configuration

Login to the web server in the DMZ and check that you are able to connect to MariaDB located on the Secure zone host.

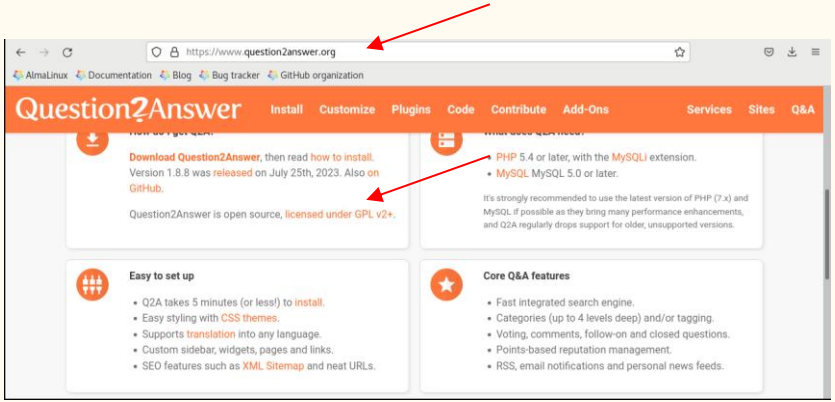


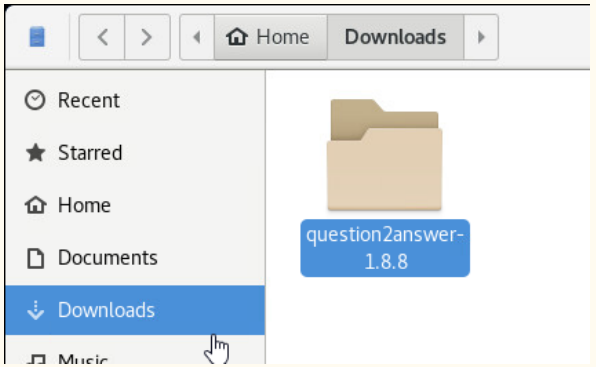
Q2A Installation & Configuration

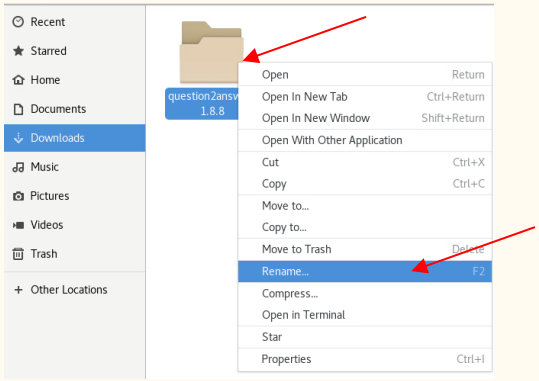
Open the browser on the Apache server and download the Question2Answer package from the Question2Answer website.

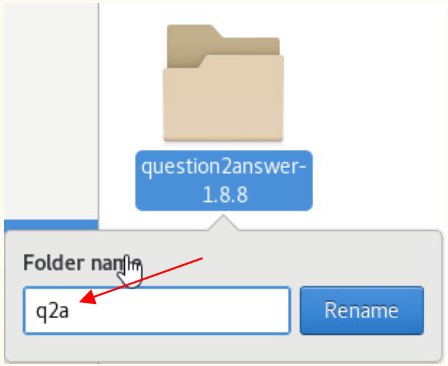
Extract the file into the downloads folder.

Rename the top file something easier like q2a.





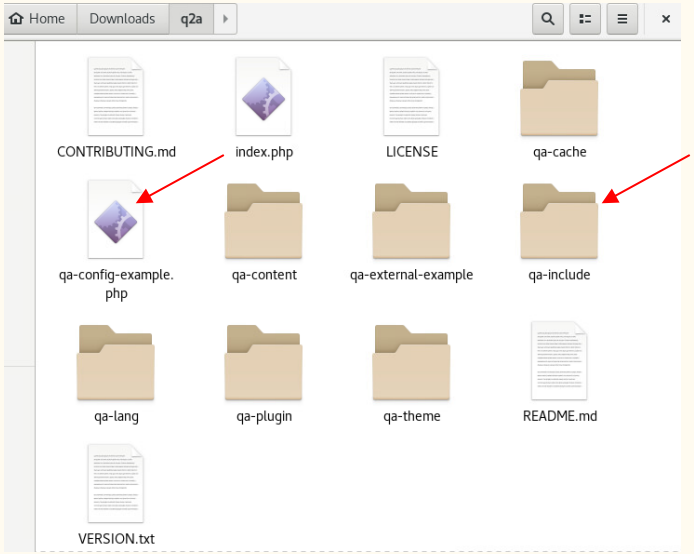


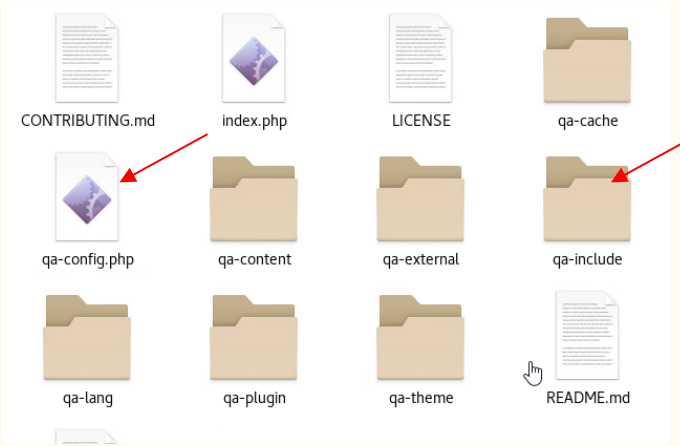


Q2A Installation & Configuration

Open the q2a folder and change the names on the example files as per the instructions on the Question2Answer website.

There is a hidden example file as well you will need to change.

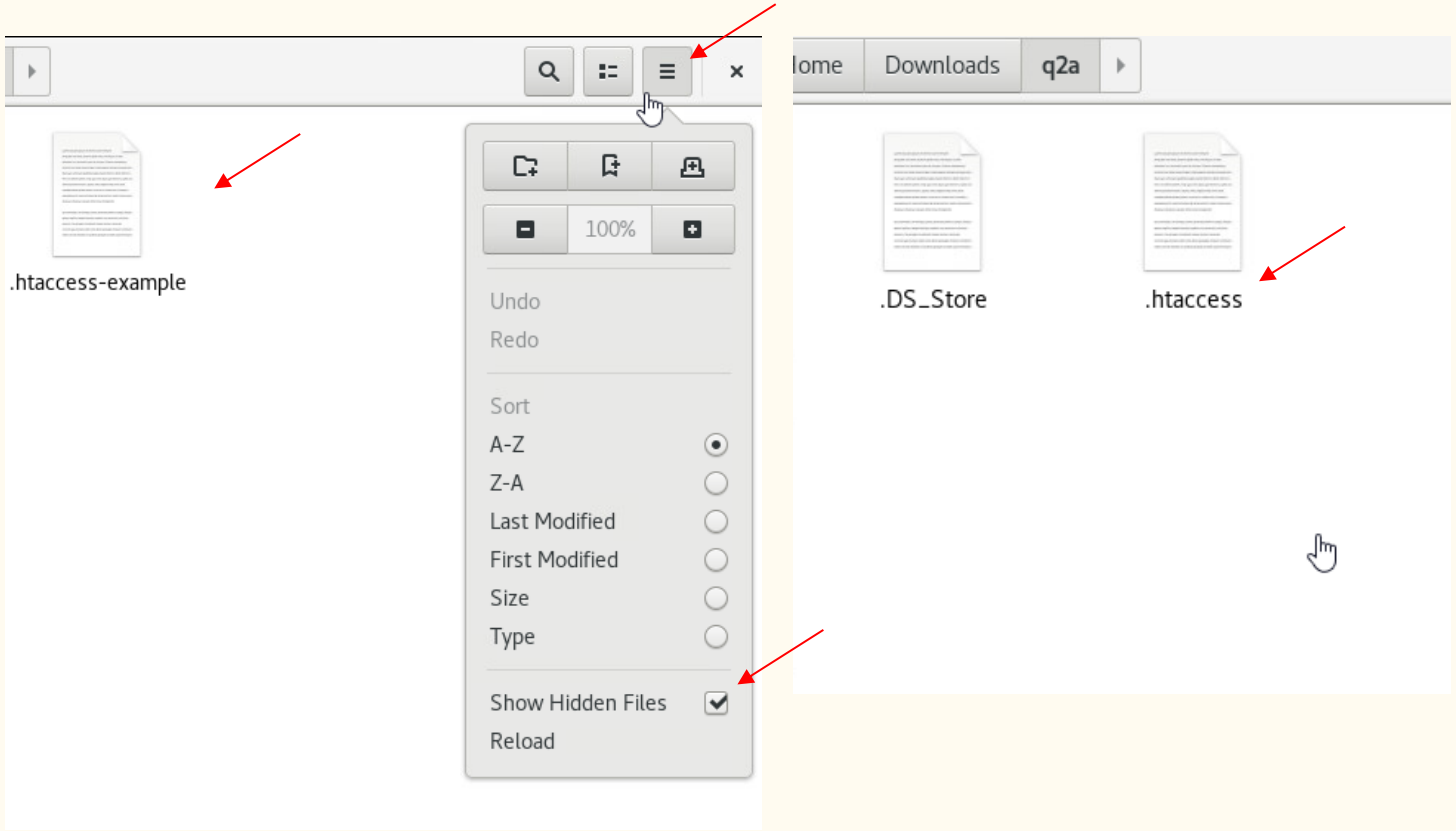




Q2A Installation & Configuration

There is a hidden example file as well you will need to change.

Select the hamburger on the top right corner and check the box show hidden files. Now remove example from the filename.



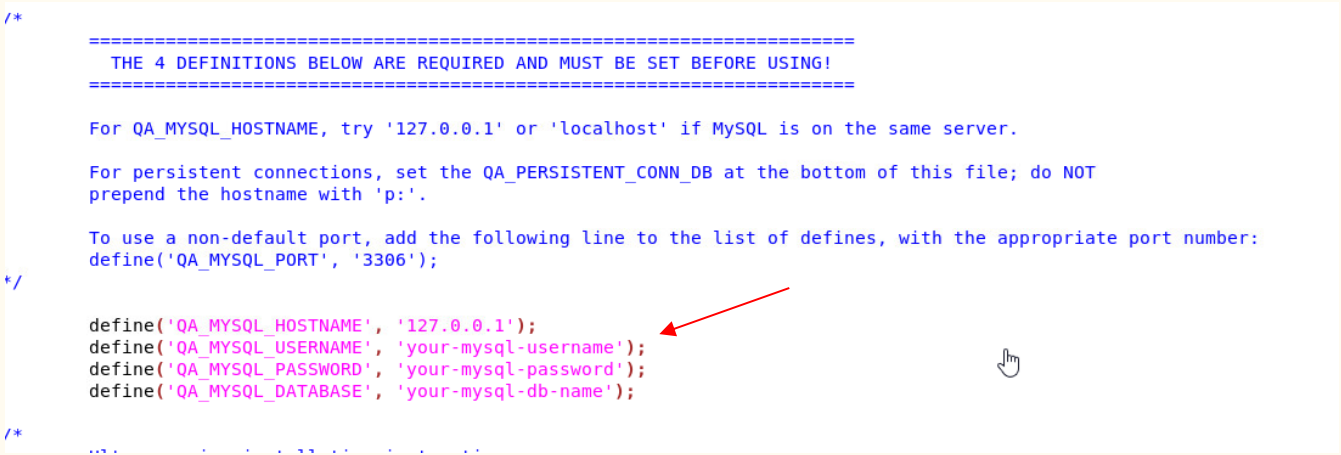
Q2A Installation & Configuration

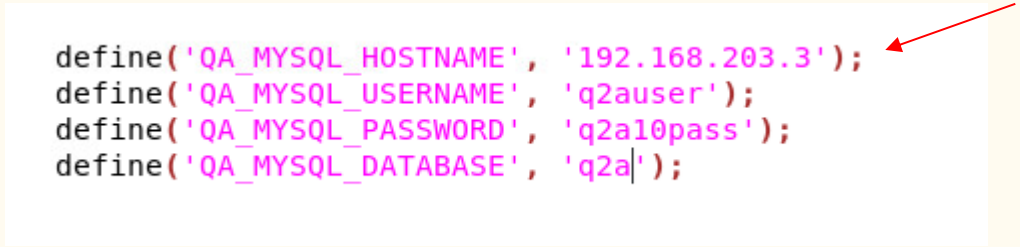
Open the config file and

Scroll down to the PHP code that defines the database configuration file.

Change the settings to fit your teams database IP, username, password, and database name..

Save your changes by clicking save in the top right corner.





Q2A Installation & Configuration

Move the Question2Answer files to the web server's "DocumentRoot" folder using the following commands. The last two commands move the hidden files we explored earlier.

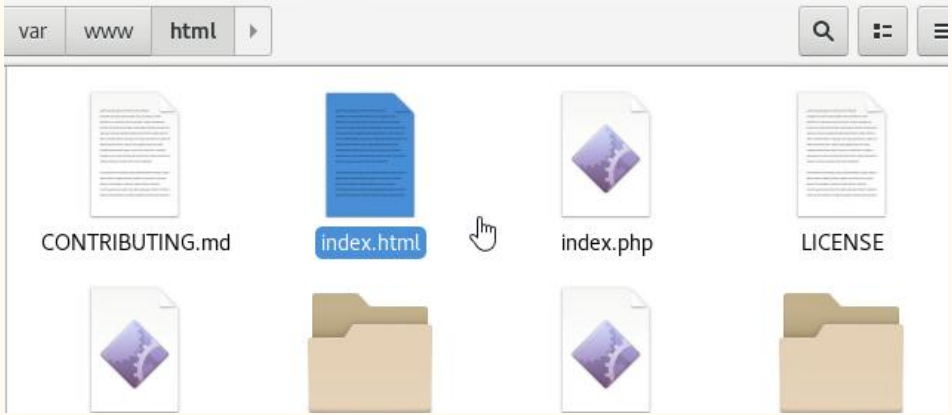
```
[cditto@localhost q2a]$ sudo mv * /var/www/html
```

```
[cditto@localhost q2a]$ sudo mv .htaccess /var/www/html
```

```
[cditto@localhost q2a]$ sudo mv .DS_Store /var/www/html
```

Q2A Installation & Configuration

In the web server's "DocumentRoot" folder, remove the index.html file so that the web server will serve the Question2Answer's index.php code instead.




```
[cditto@localhost q2a]$ sudo rm /var/www/html/index.html
```

Q2A Installation & Configuration

The Q2A files that were moved from the download directory to the html directory were automatically labeled when they were placed in the download folder. These files need to be restored to the correct labels based on the context of where they reside in the file system.

```
[cditto@localhost html]$ ls -Z
unconfined_u:object_r:user_home_t:s0 CONTRIBUTING.md
unconfined_u:object_r:user_home_t:s0 index.php
unconfined_u:object_r:user_home_t:s0 LICENSE
unconfined_u:object_r:user_home_t:s0 qa-cache
unconfined_u:object_r:user_home_t:s0 qa-config.php
unconfined_u:object_r:user_home_t:s0 qa-content
unconfined_u:object_r:user_home_t:s0 qa-external
unconfined_u:object_r:user_home_t:s0 qa-include
unconfined_u:object_r:user_home_t:s0 qa-lang
unconfined_u:object_r:user_home_t:s0 qa-plugin
unconfined_u:object_r:user_home_t:s0 qa-theme
unconfined_u:object_r:user_home_t:s0 README.md
unconfined_u:object_r:user_home_t:s0 VERSION.txt
```

```
[cditto@localhost www]$ restorecon -R html
```

```
onfined_u:object_r:httpd_sys_content_t:s0 CONTRIBUTING.md
onfined_u:object_r:httpd_sys_content_t:s0 index.php
onfined_u:object_r:httpd_sys_content_t:s0 LICENSE
onfined_u:object_r:httpd_sys_content_t:s0 qa-cache
onfined_u:object_r:httpd_sys_content_t:s0 qa-config.php
onfined_u:object_r:httpd_sys_content_t:s0 qa-content
onfined_u:object_r:httpd_sys_content_t:s0 qa-external
onfined_u:object_r:httpd_sys_content_t:s0 qa-include
onfined_u:object_r:httpd_sys_content_t:s0 qa-lang
onfined_u:object_r:httpd_sys_content_t:s0 qa-plugin
onfined_u:object_r:httpd_sys_content_t:s0 qa-theme
onfined_u:object_r:httpd_sys_content_t:s0 README.md
onfined_u:object_r:httpd_sys_content_t:s0 VERSION.txt
```

Q2A Installation & Configuration

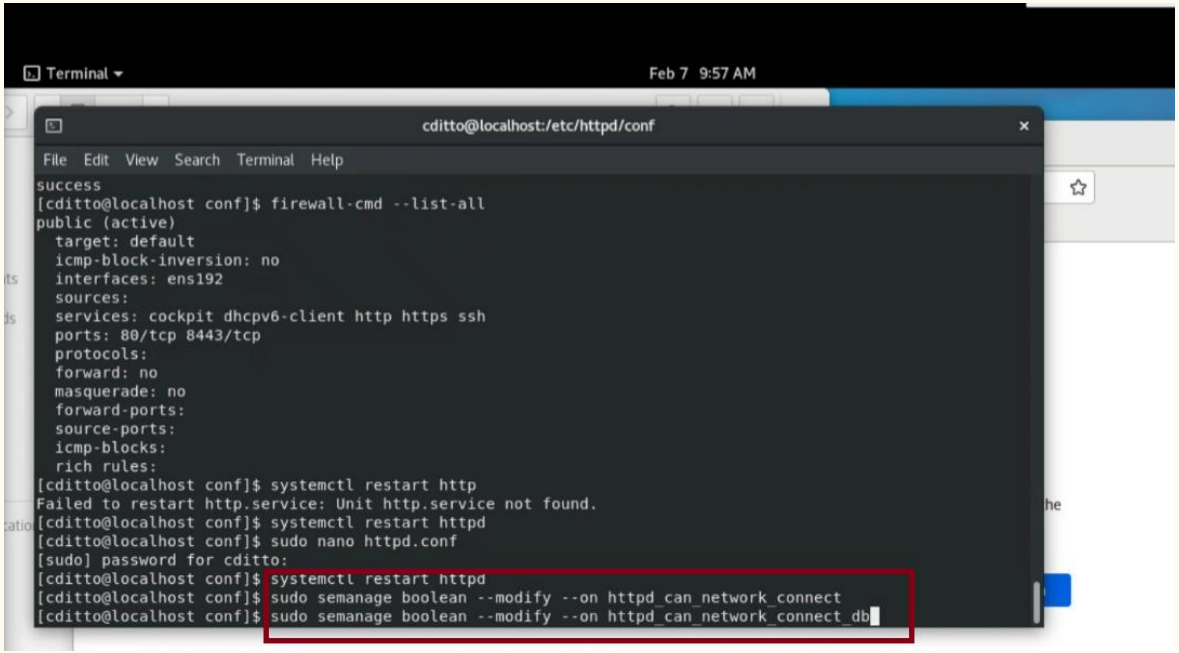
In this app, PHP uses json.

Download json to the machine.

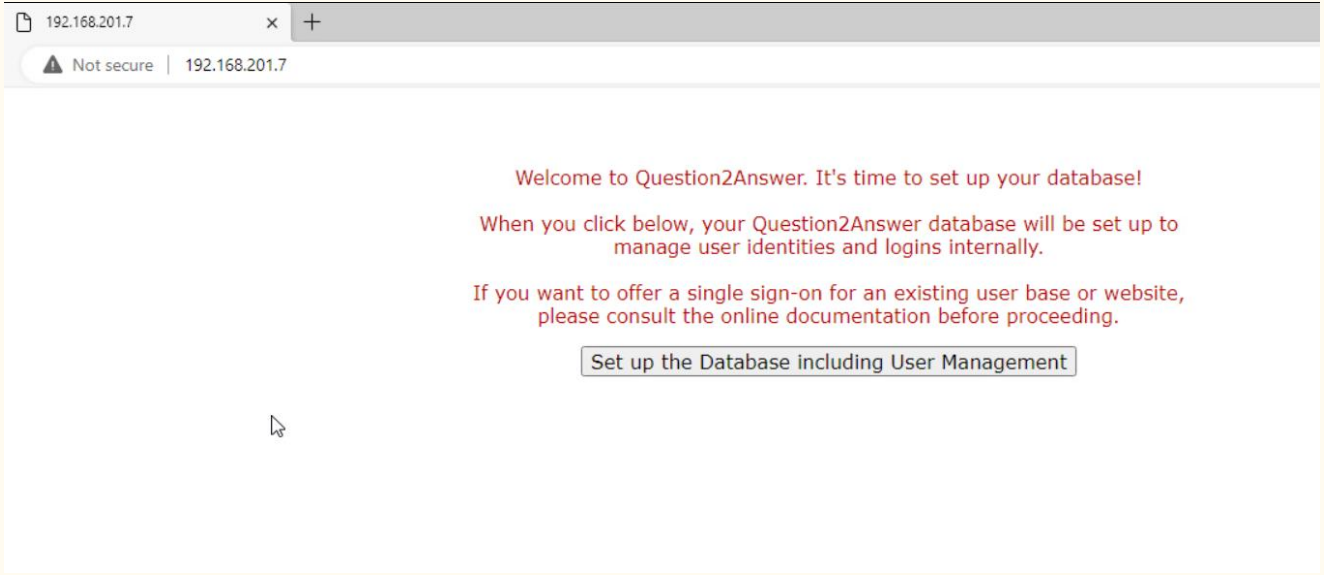
Restart the machine so the PHP engine will notice its new functions.

```
[cditto@localhost html]$ sudo dnf -y install php-json
```

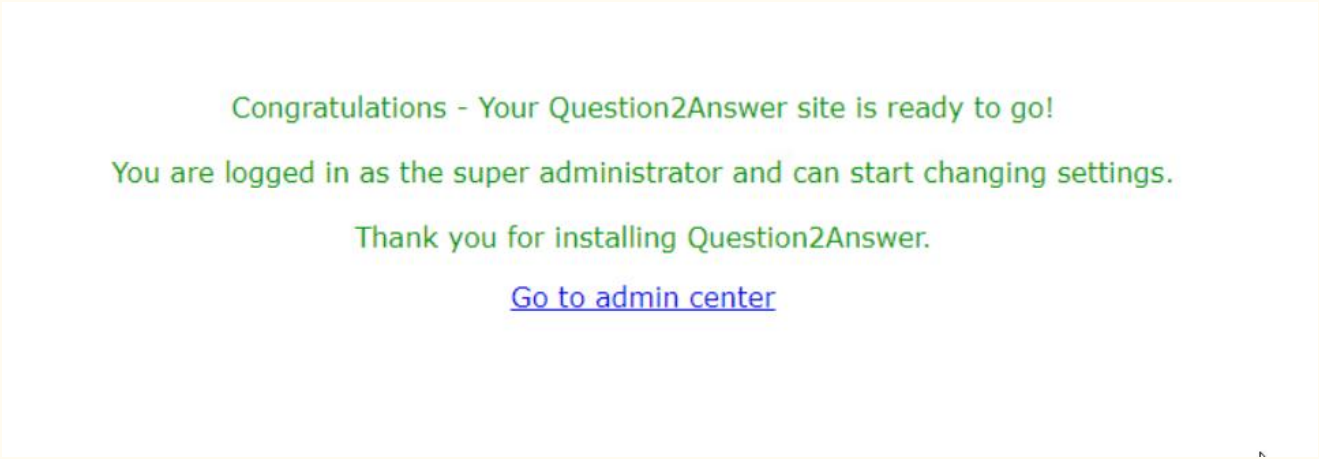
```
Installed:
  php-json-7.2.24-1.module_el8.3.0+2010+7c76a223.x86_64
```


Terminal
Feb 7 9:57 AM
cditto@localhost:/etc/httpd/conf
File Edit View Search Terminal Help
success
[cditto@localhost conf]$ firewall-cmd --list-all
public (active)
target: default
icmp-block-inversion: no
interfaces: ens192
sources:
services: cockpit dhcpv6-client http https ssh
ports: 80/tcp 8443/tcp
protocols:
forward: no
masquerade: no
forward-ports:
source-ports:
icmp-blocks:
rich rules:
[cditto@localhost conf]$ systemctl restart http
Failed to restart http.service: Unit http.service not found.
[cditto@localhost conf]$ systemctl restart httpd
[cditto@localhost conf]$ sudo nano httpd.conf
[sudo] password for cditto:
[cditto@localhost conf]$ systemctl restart httpd
[cditto@localhost conf]$ sudo semanage boolean --modify --on httpd_can_network_connect
[cditto@localhost conf]$ sudo semanage boolean --modify --on httpd_can_network_connect_db

Q2A Installation & Configuration

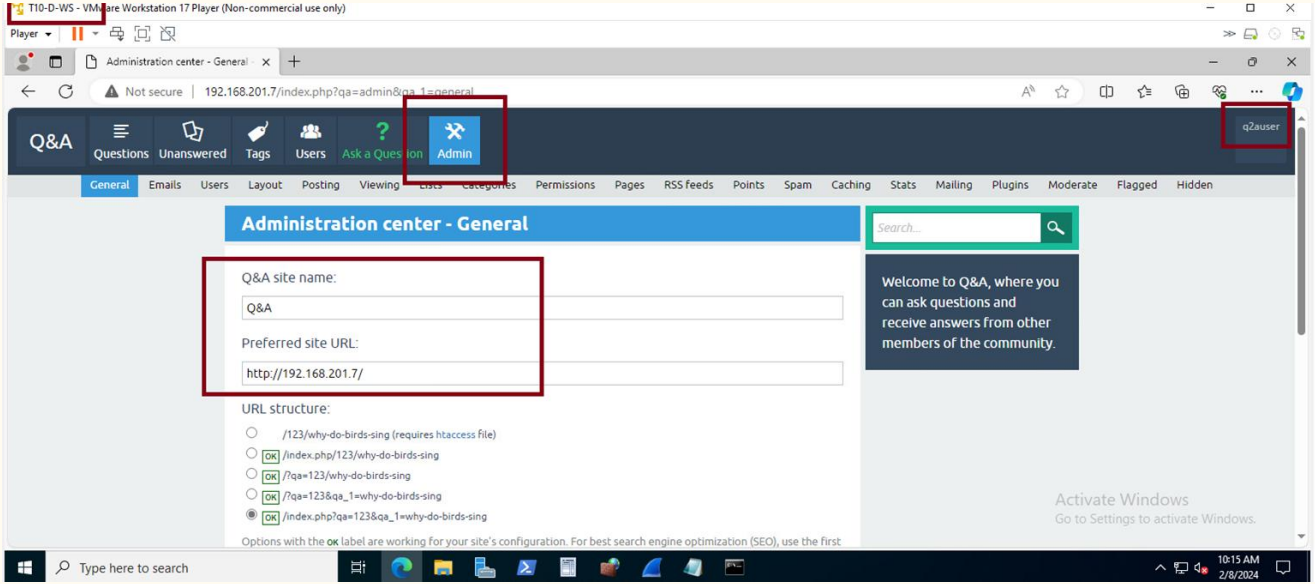


Q2A Installation & Configuration



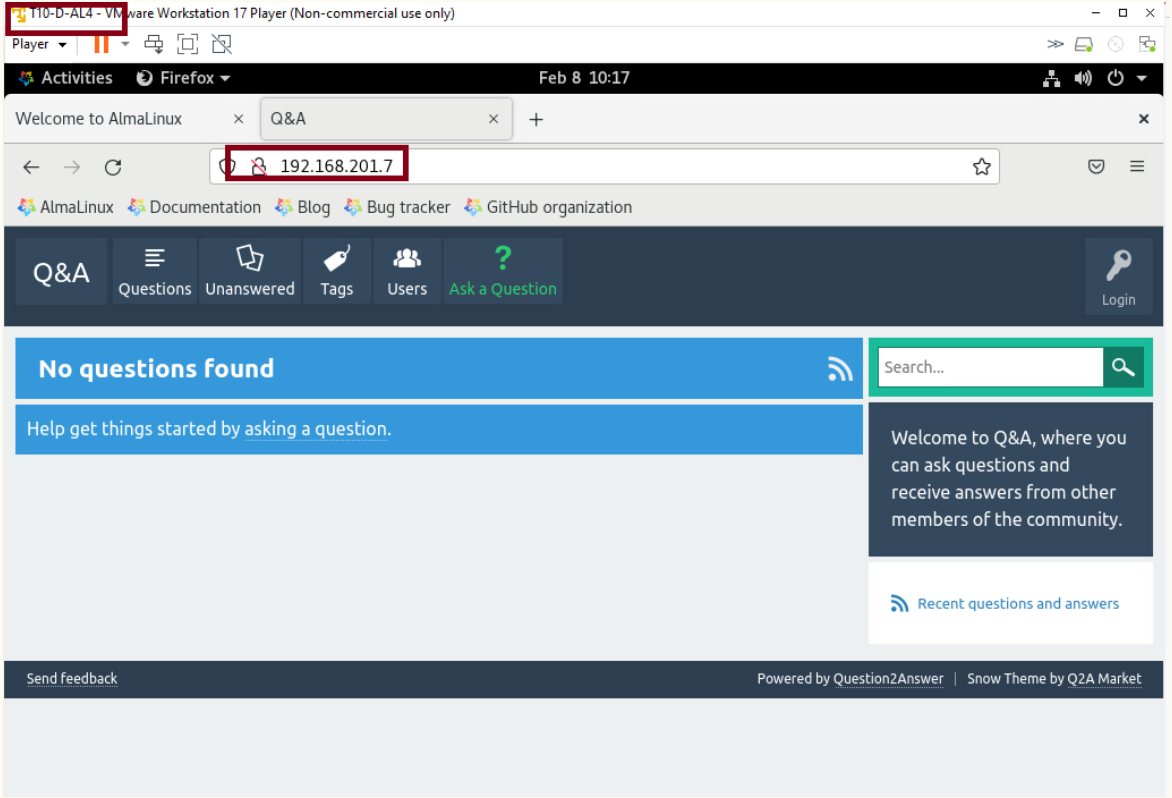
Q2A Installation & Configuration

Q2a page open in windows system in the DMZ zone



Q2A Installation & Configuration

Q2a page open in Linux system in the DMZ zone. This system is the one that host the apache server.



Q2A Installation & Configuration

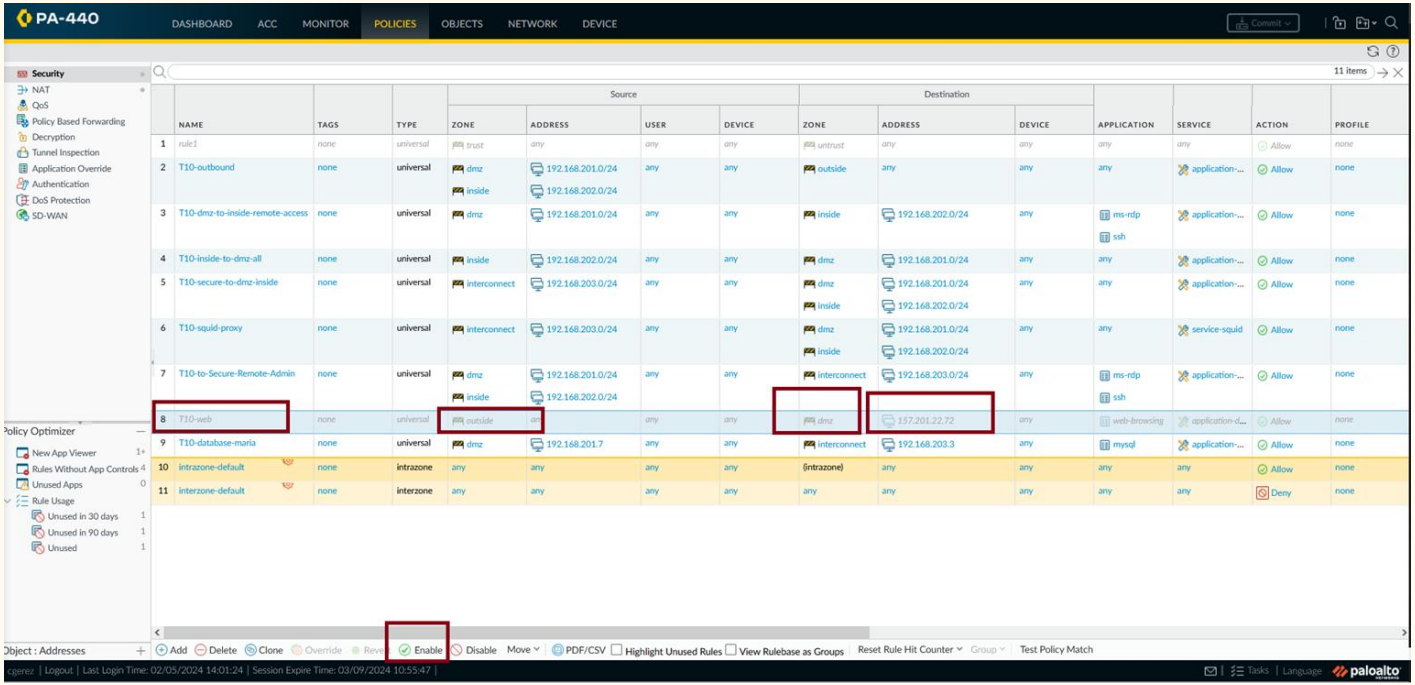
Restrict access to phpinfo.php .

Since we expose our app to the untrusted internet before we restrict access using chown command .

```
sudo chown -R /var/www/html/phpinfo.php
```

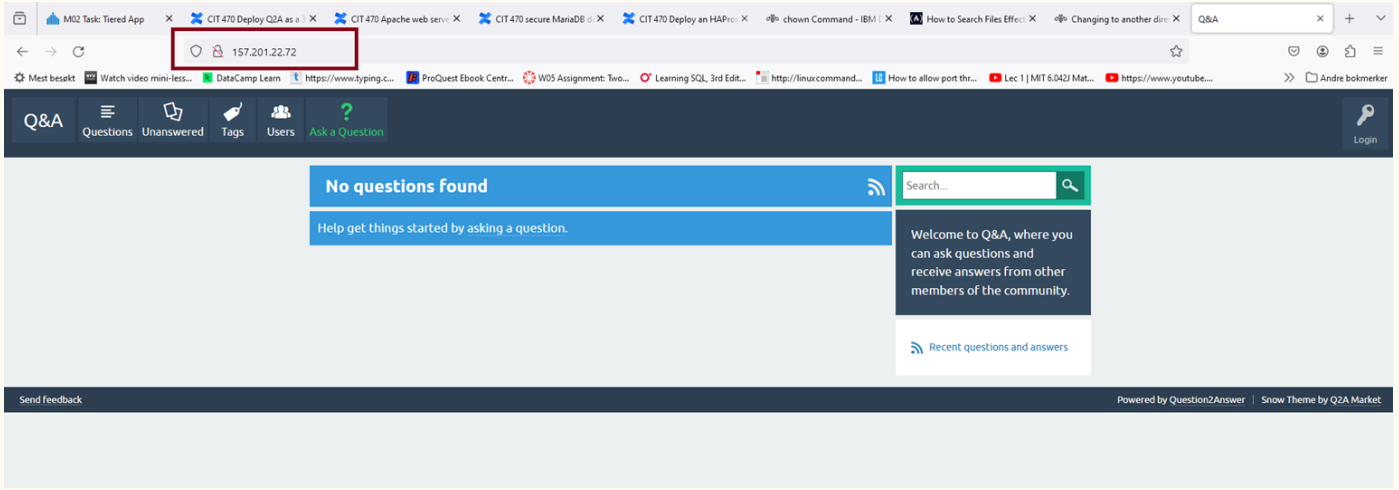
Q2A Installation & Configuration

Enable the web browsing rule to test the internet connection from outside.
Remember to commit the changes after enable the rule.



Q2A Installation & Configuration

Access the q2a from a computer outside our network. Use the browser and put the outside address, in this case: 157.201.22.72

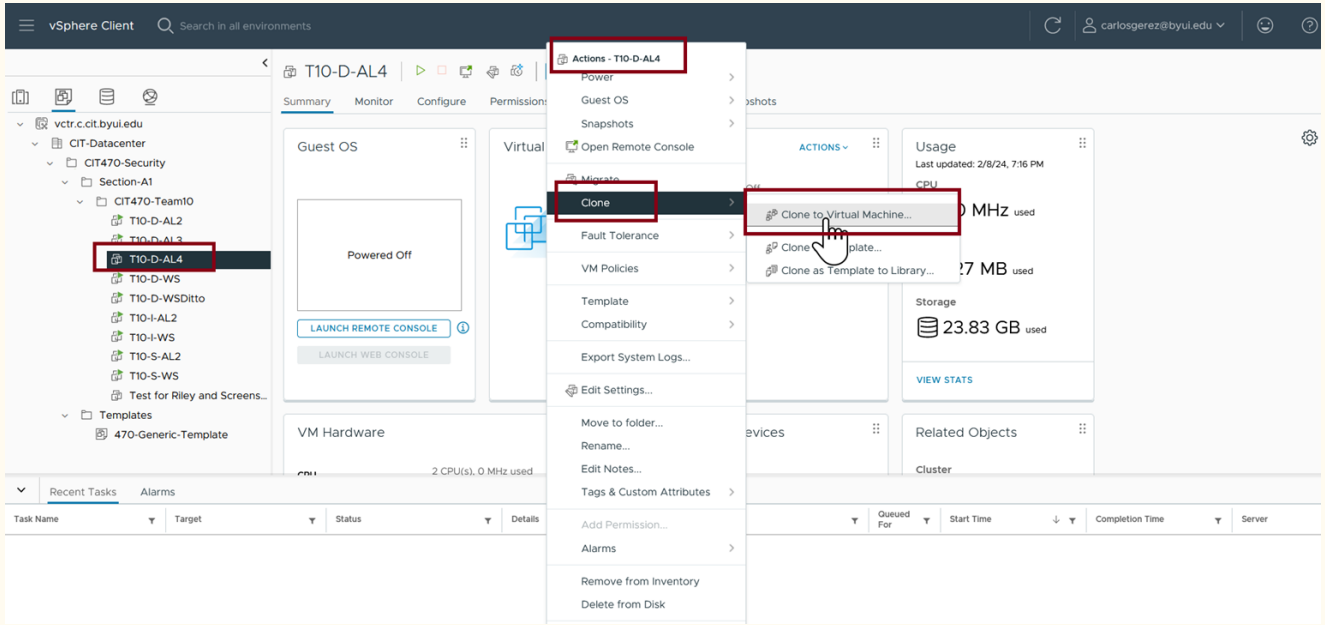


HAProxy load balancer configuration

Cloning the machine that holds the apache server.

Before start, power off the machine that will be cloned.

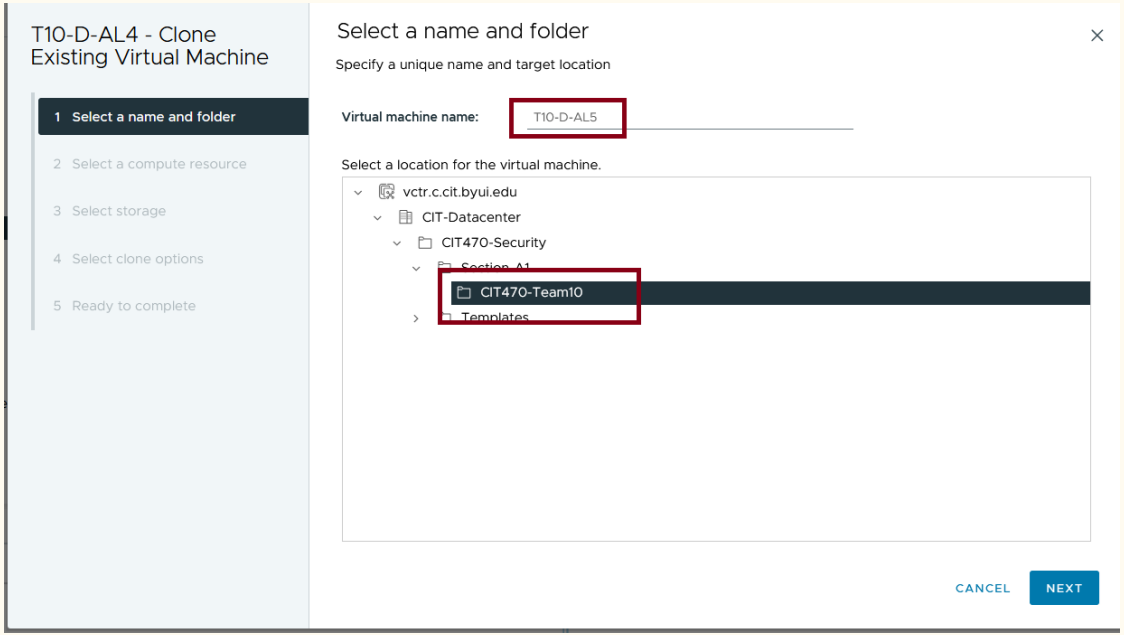
Use actions tab in sphere to get to the clone options menu.



Cloning the machine that holds the apache server.

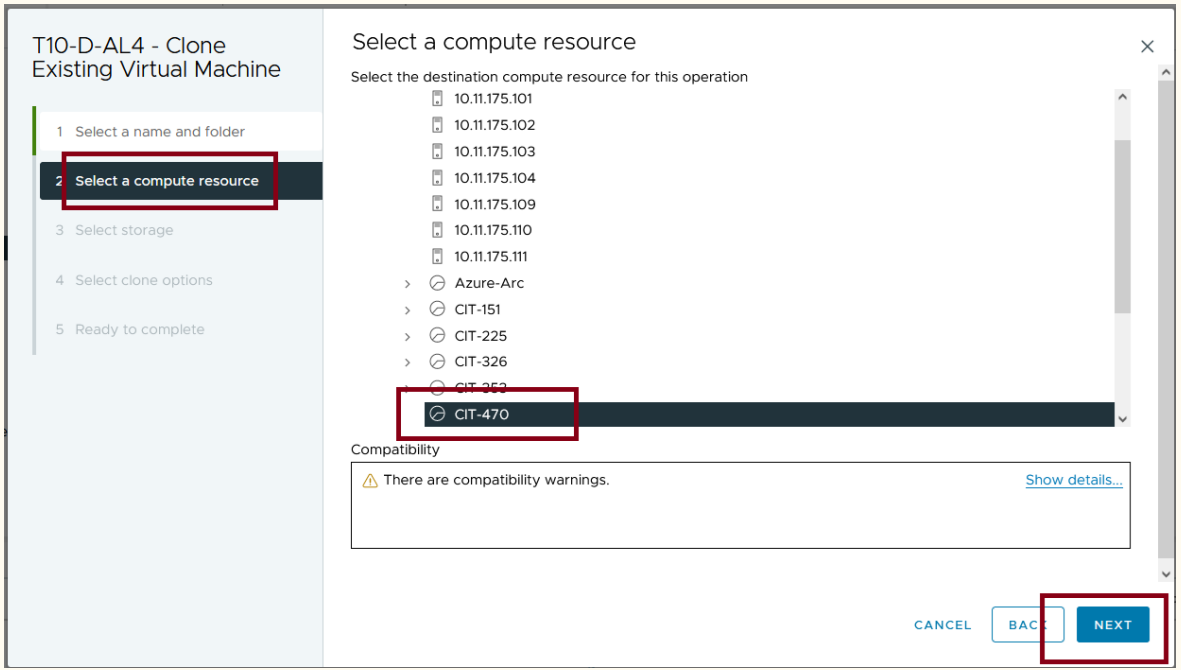
From now on follow the indications in the screen.

First select a name and a folder were the machine will reside. Press next.



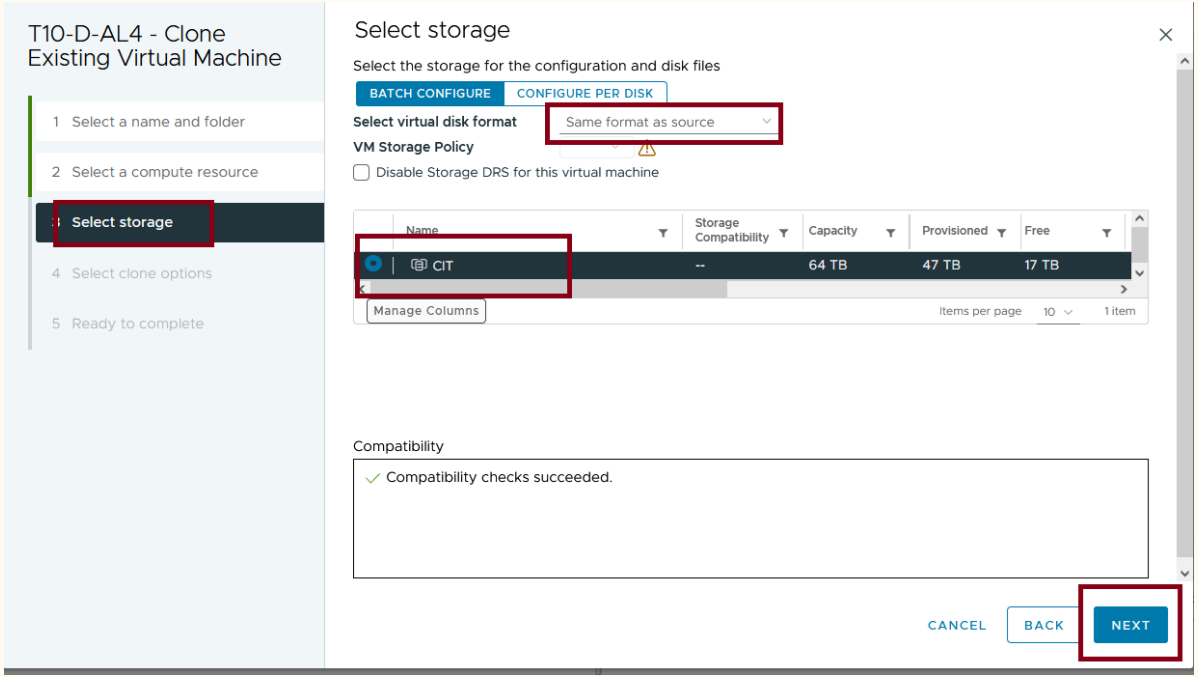
Cloning the machine that holds the apache server.

Select a computer resource, in our class the assigned for our class. Press next.



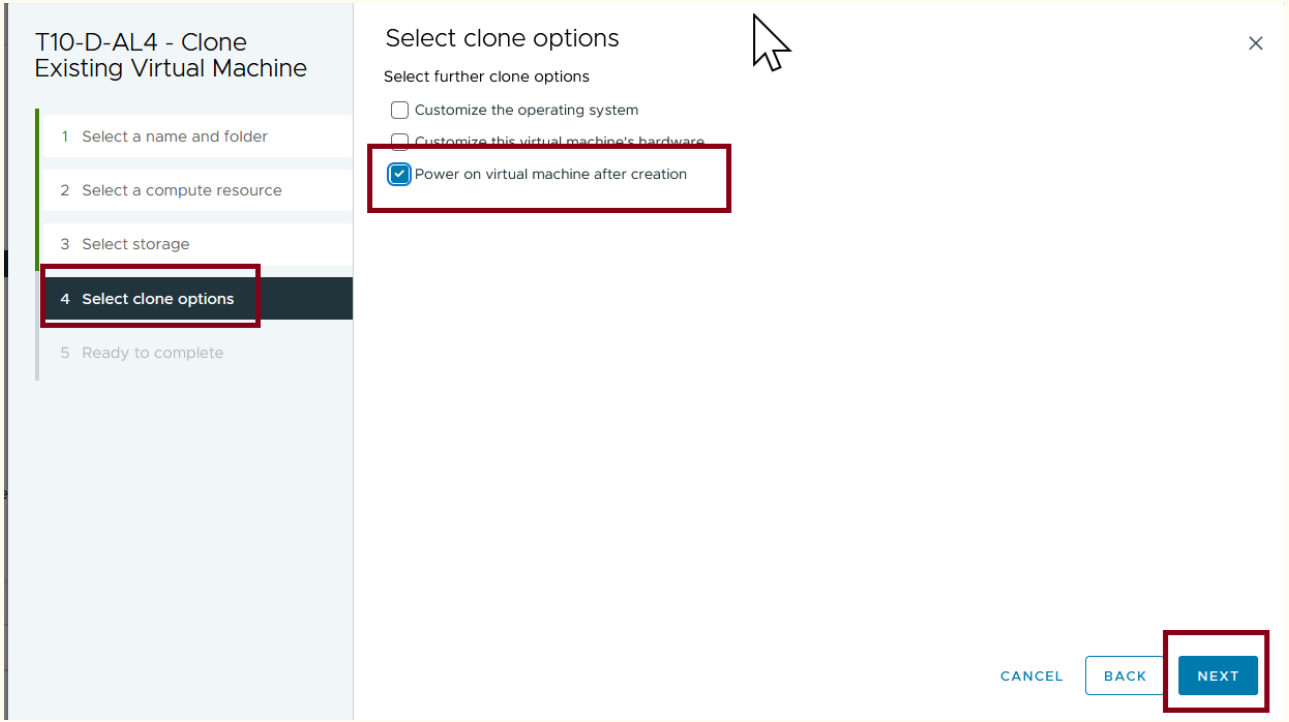
Cloning the machine that holds the apache server.

Select the storage and click in same format as source. Press next.



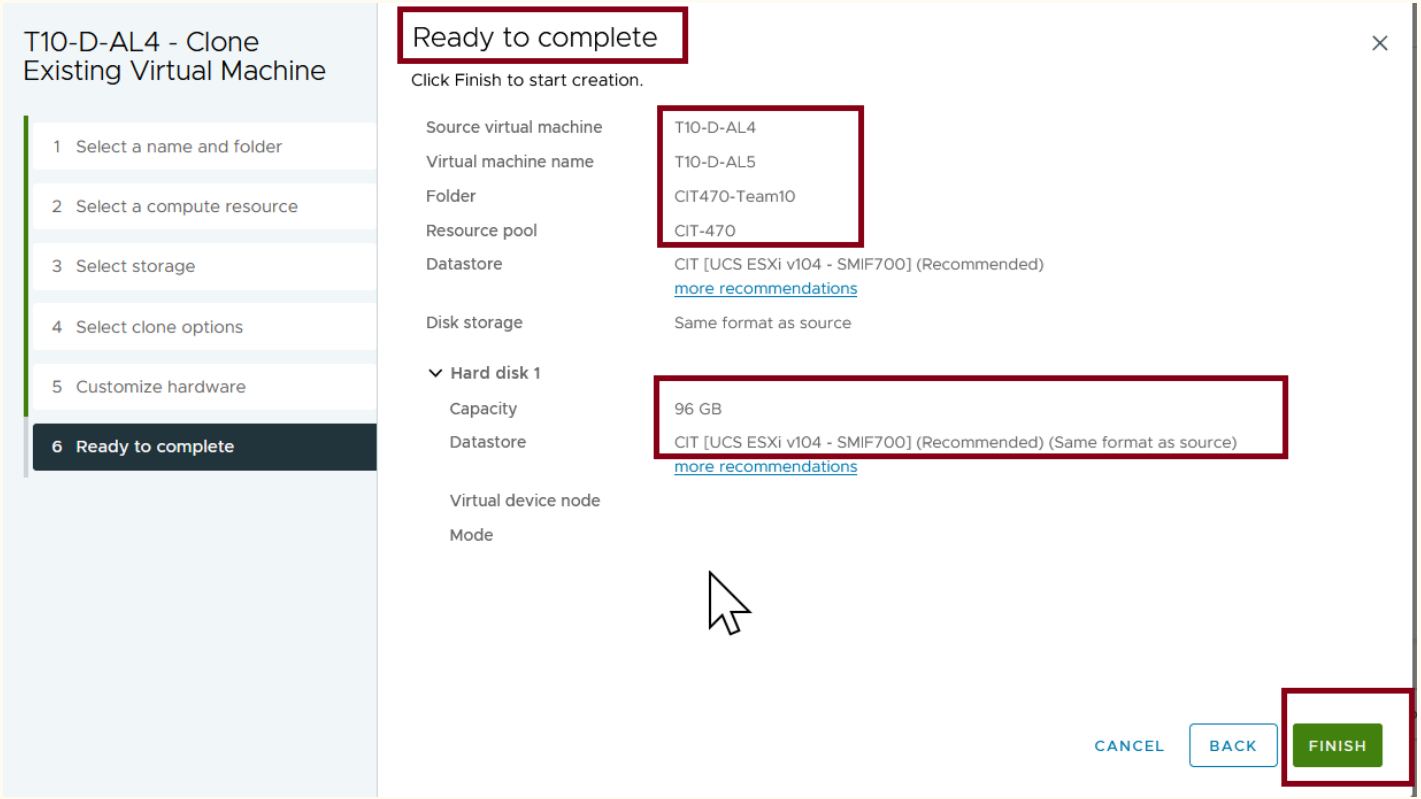
Cloning the machine that holds the apache server.

On clone options select power on virtual machine after creation.Press next.



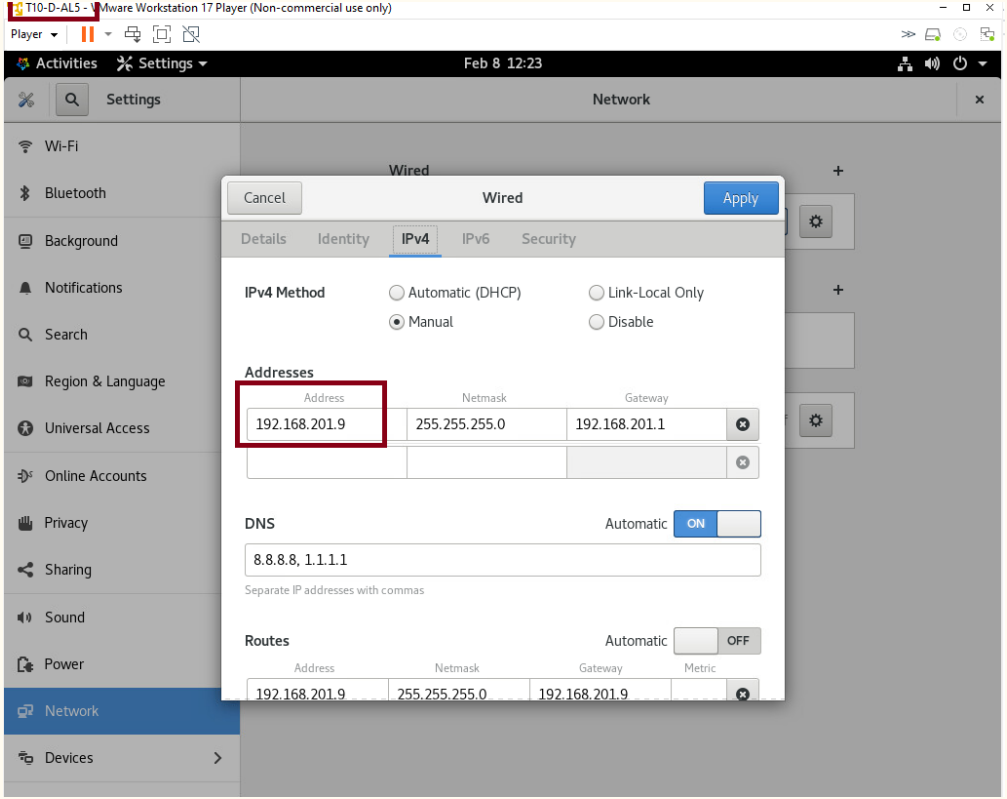
Cloning the machine that holds the apache server.

Review and press finish.



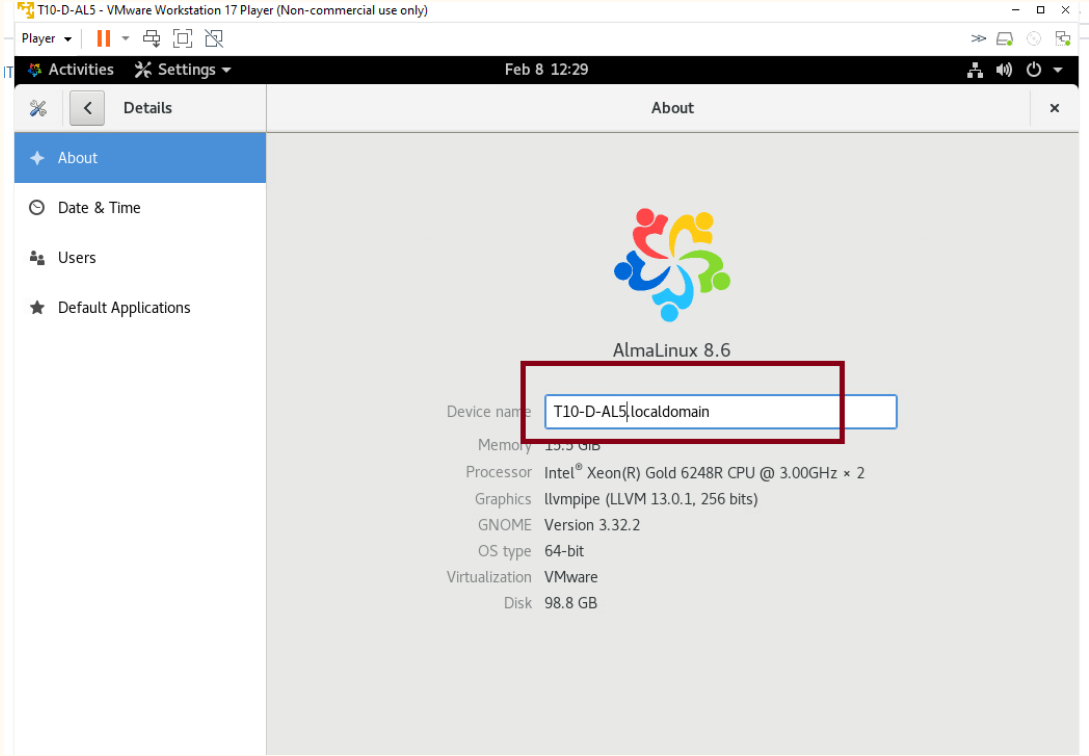
Cloning the machine that holds the apache server.

Change the ip addresses to the new machine address.



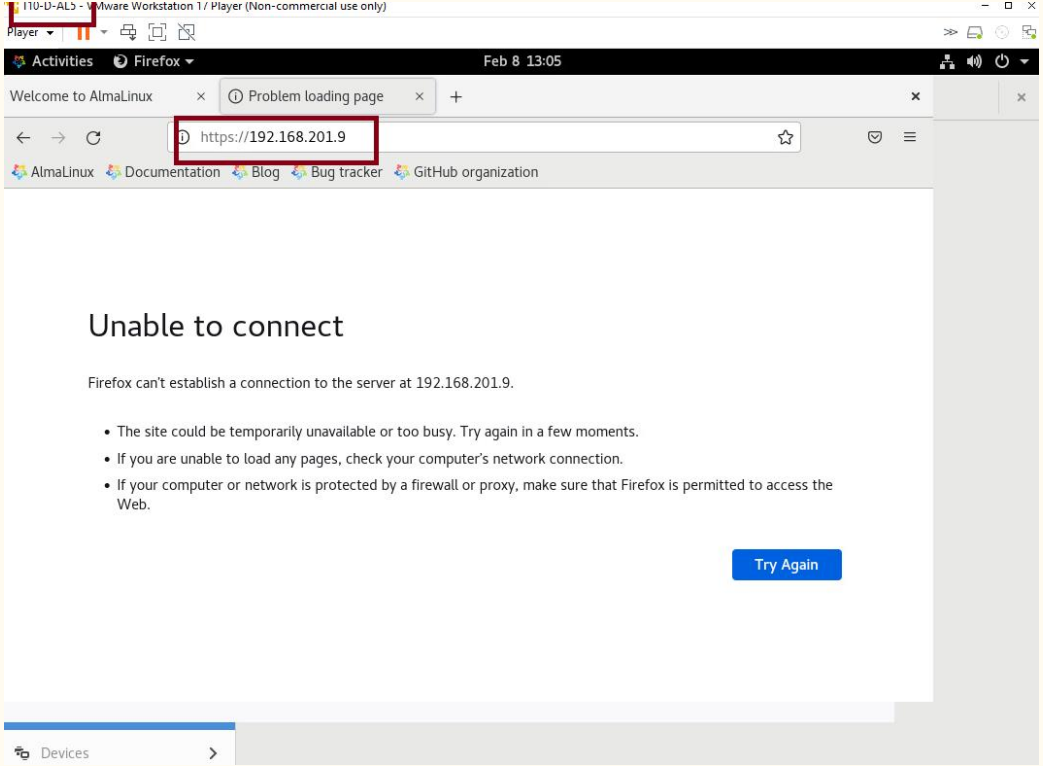
Cloning the machine that holds the apache server.

Change the name
to the correct
new name.



Configuring the app in the cloned machine.

When trying to connect to the server in the cloned machine we cannot.



Configuring the app in the cloned machine.

We has to edit the configuration file to put the new listening address. The file we has to update is httpd.conf. We use the following commands:

1. change to the directory:

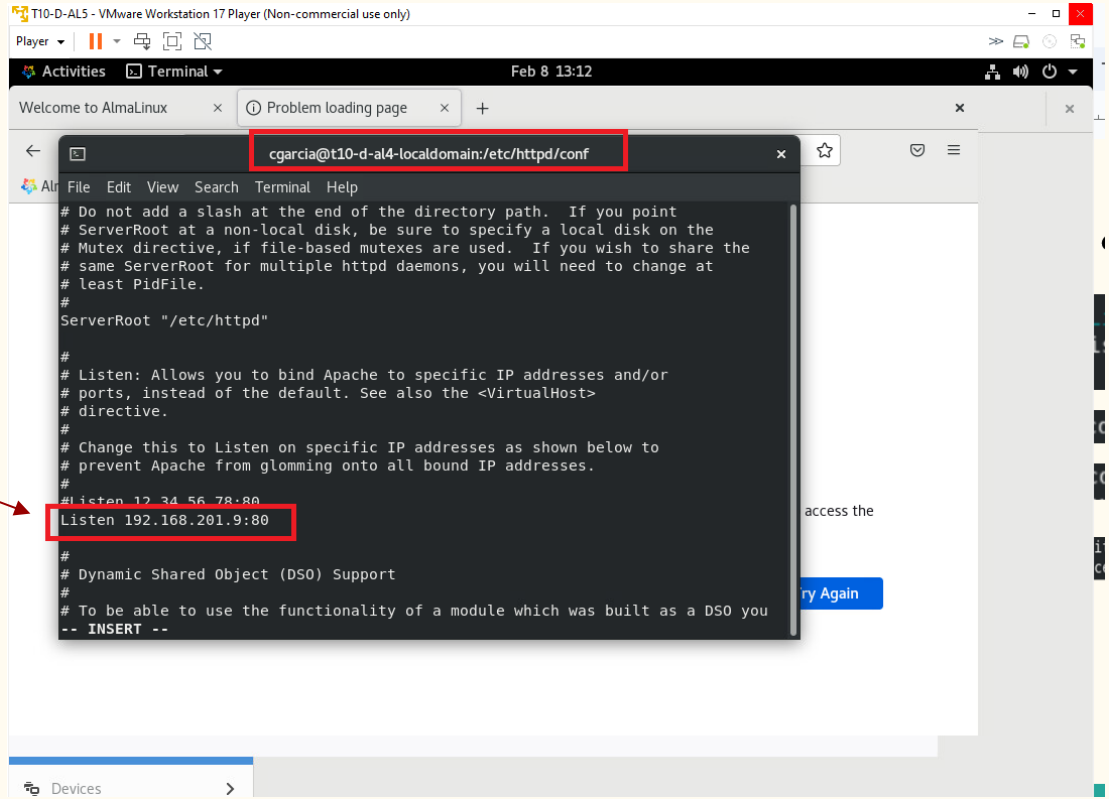
```
cd /etc/httpd/conf
```

1. edit the file with vi and change the Listen address.

```
sudo vi httpd.conf
```

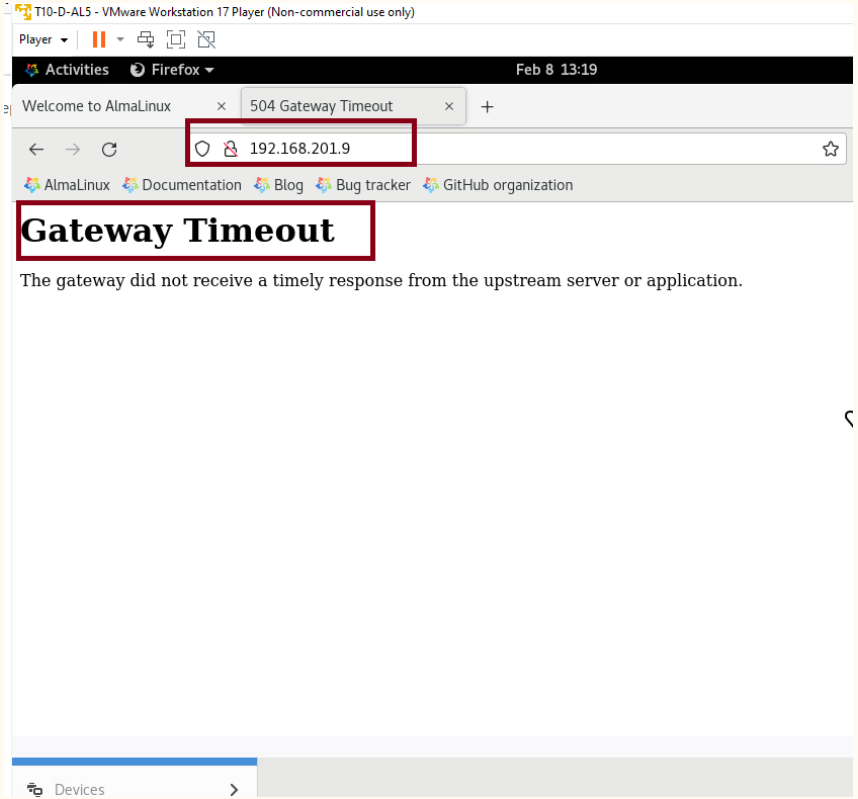
1. restart the service to apply the changes.

```
systemctl restart httpd
```



Configuring the app in the cloned machine.

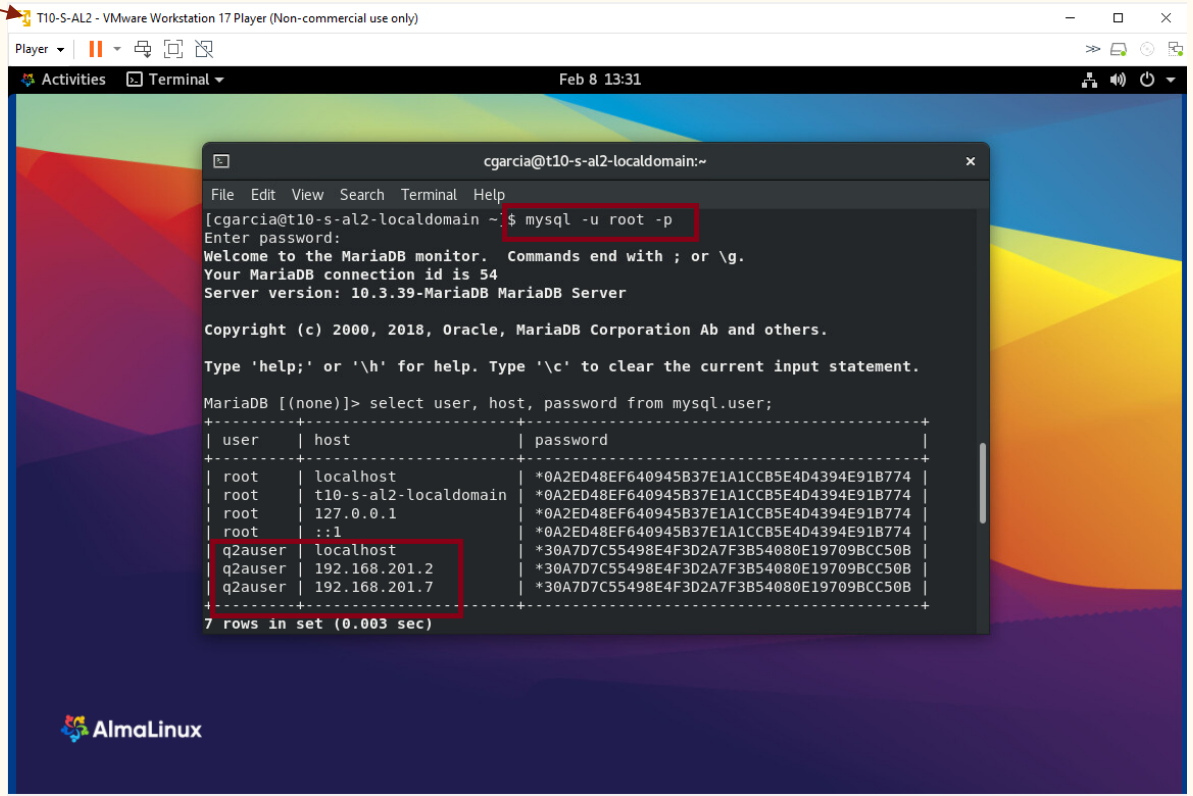
We get this new message because the mariadb database is not yet configured to connect with this machine.



Configuring the app in the cloned machine.

In the secure zone machine that host the mariadb we must configure the access from the new server. The first command give access as root to the database:

```
mysql -u root -p
```



Configuring the app in the cloned machine.

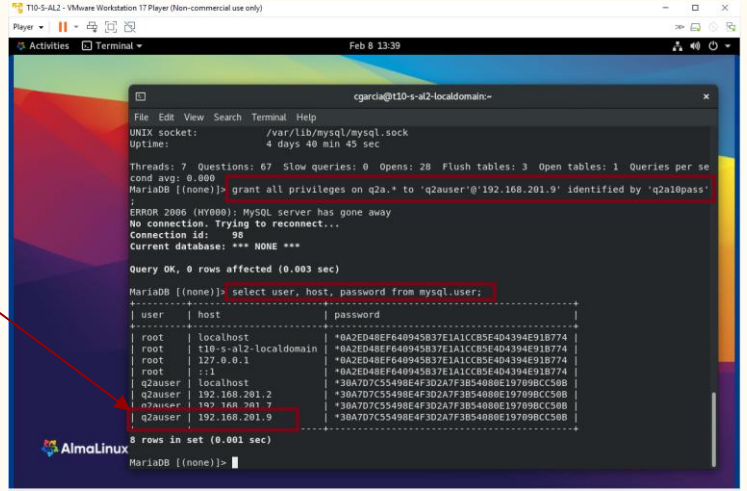
To give access use the following command:

```
grant all privileges on q2a.* to q2auser'@'192.168.201.9' identified by 'q2a10pass';
```

Then you can use the following command to check the result:

```
select user, host, password from mysql.user;
```

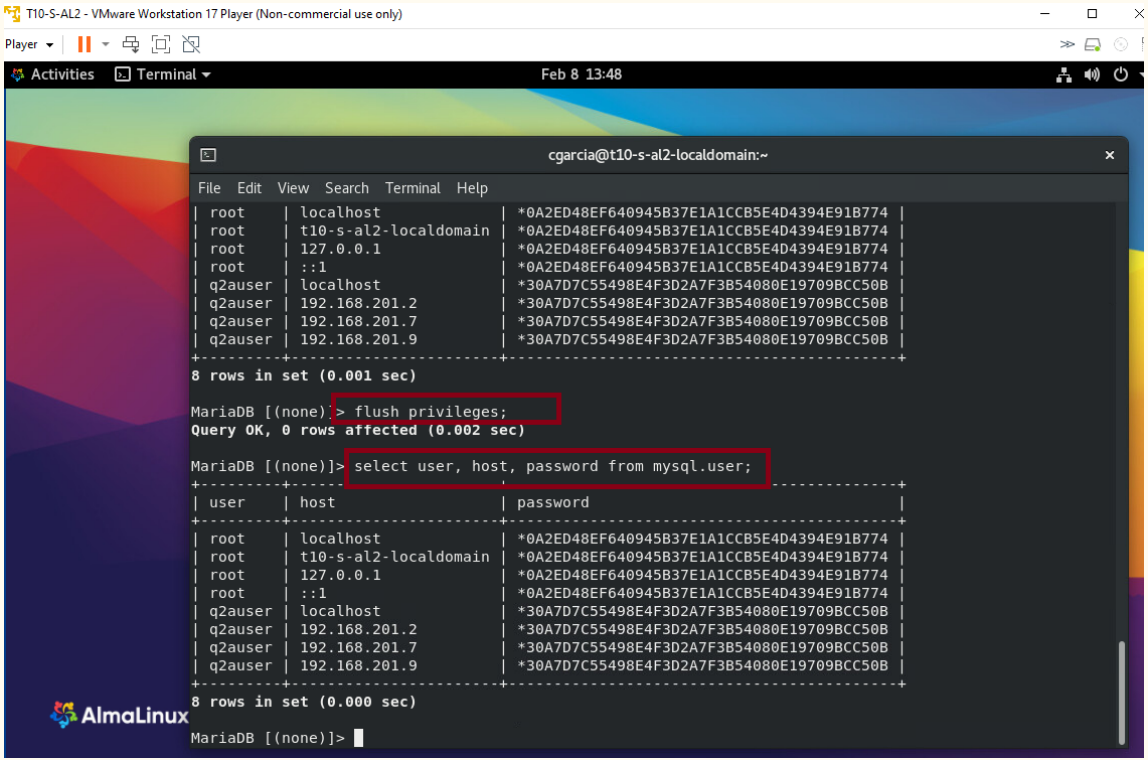
In the last line appears the new access.



Configuring the app in the cloned machine.

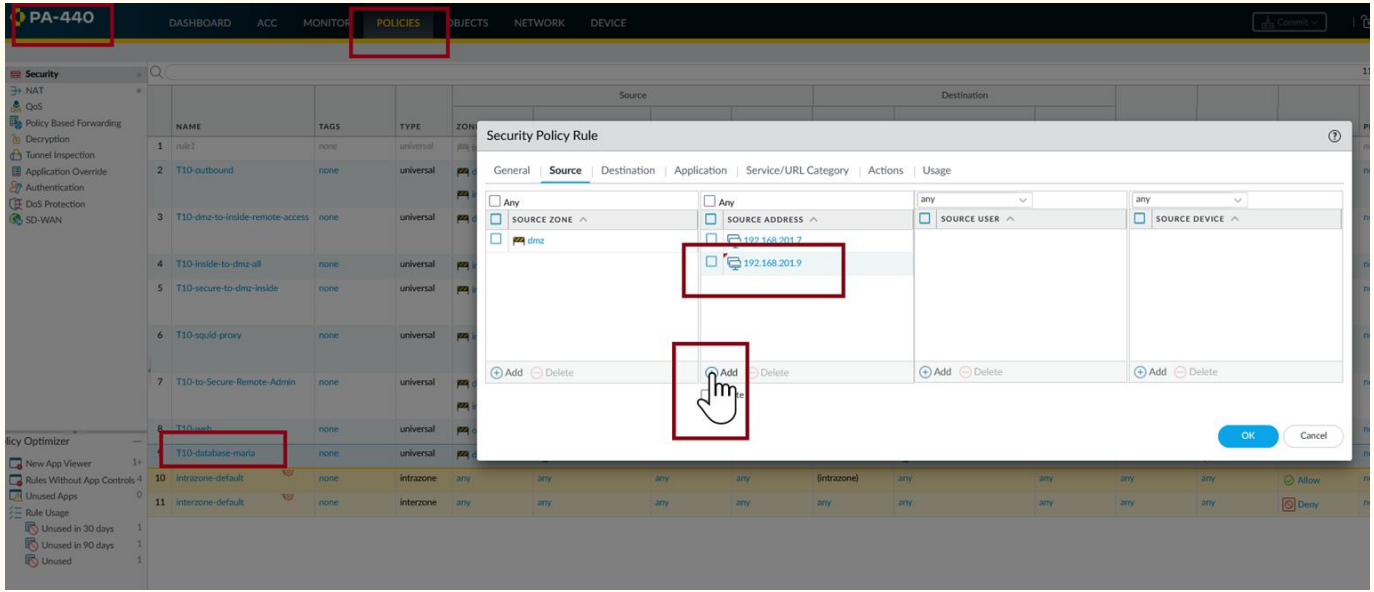
Remember to flush those privileges:

```
flush privileges;
```



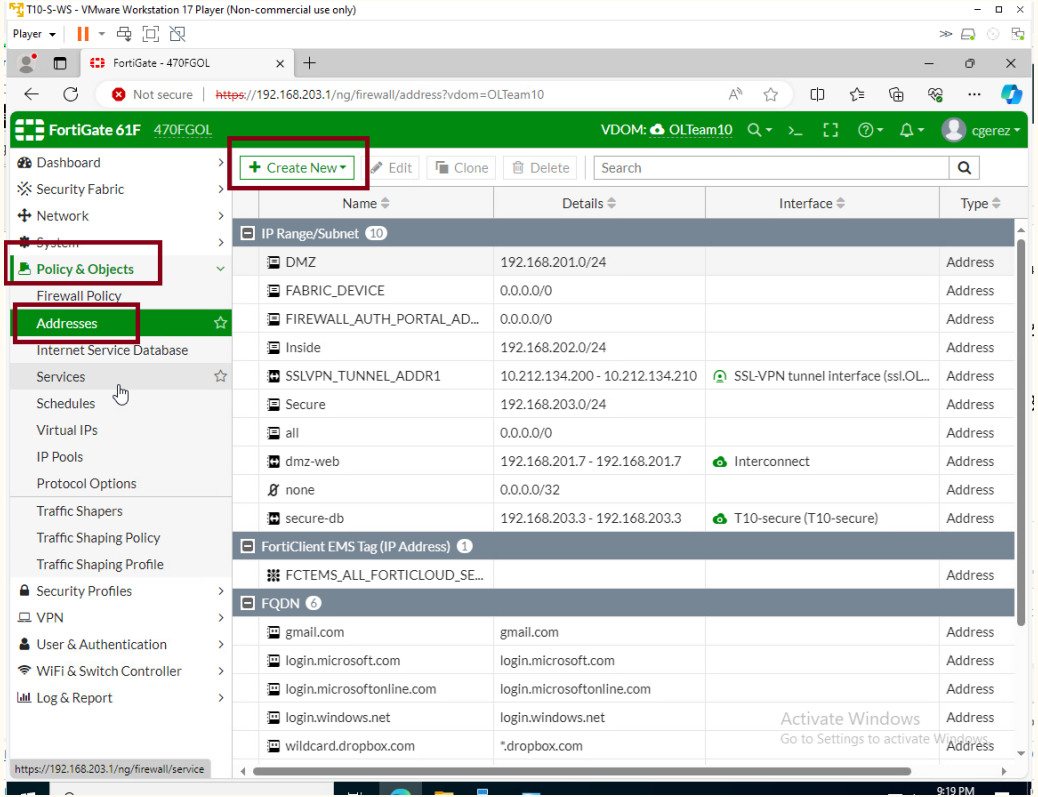
Configuring the app in the cloned machine.

We has to give access also in the Firewalls. On the Palo Alto policies add the new address to the rule to allow mysql from the dmz zone. In policies select the rule, and in the source tab add the new address. Don't forget to commit your changes.



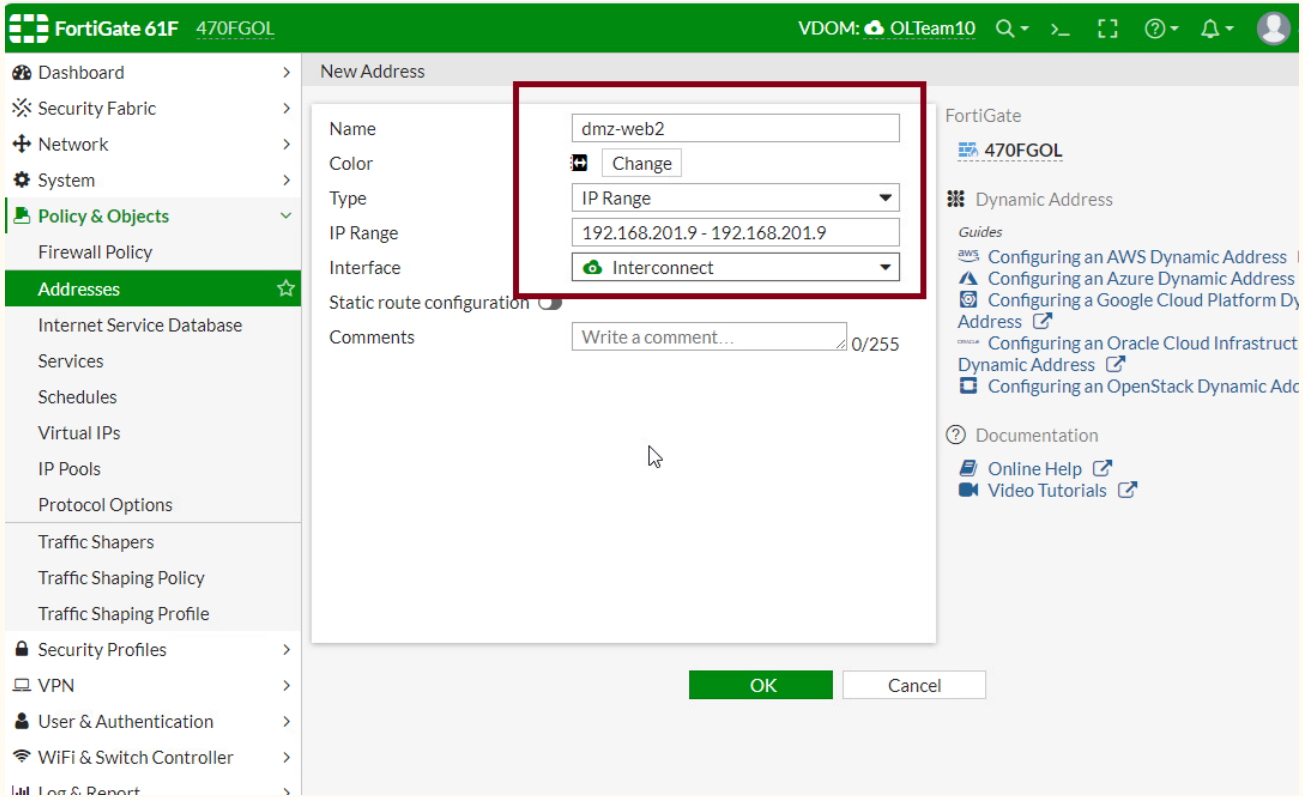
Configuring the app in the cloned machine.

Then to add access in the Fortigate, opened from a machine in the secure zone create a new address object to add to the rule that allows connections from the DMZ.In Policy & Objects tab, select Addresses, and the Create New tab.



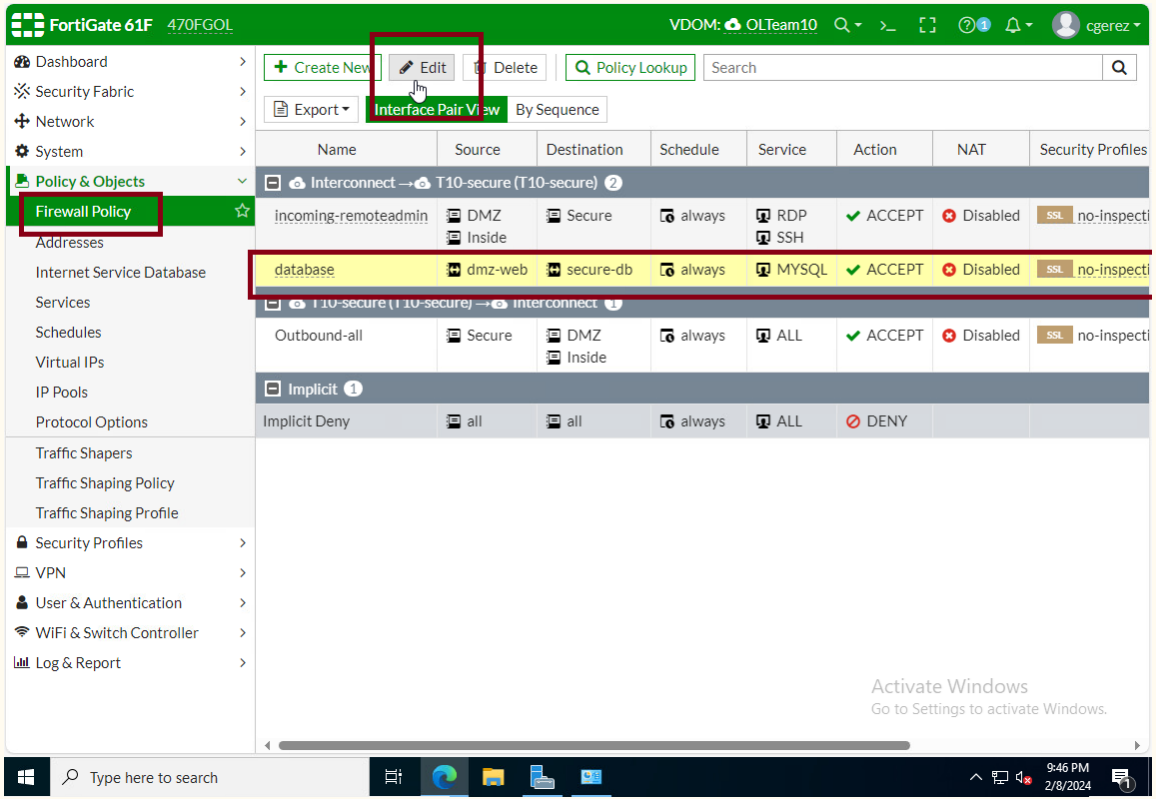
Configuring the app in the cloned machine.

Then create the new address as the one created before to allow connections for the other machine and press OK.



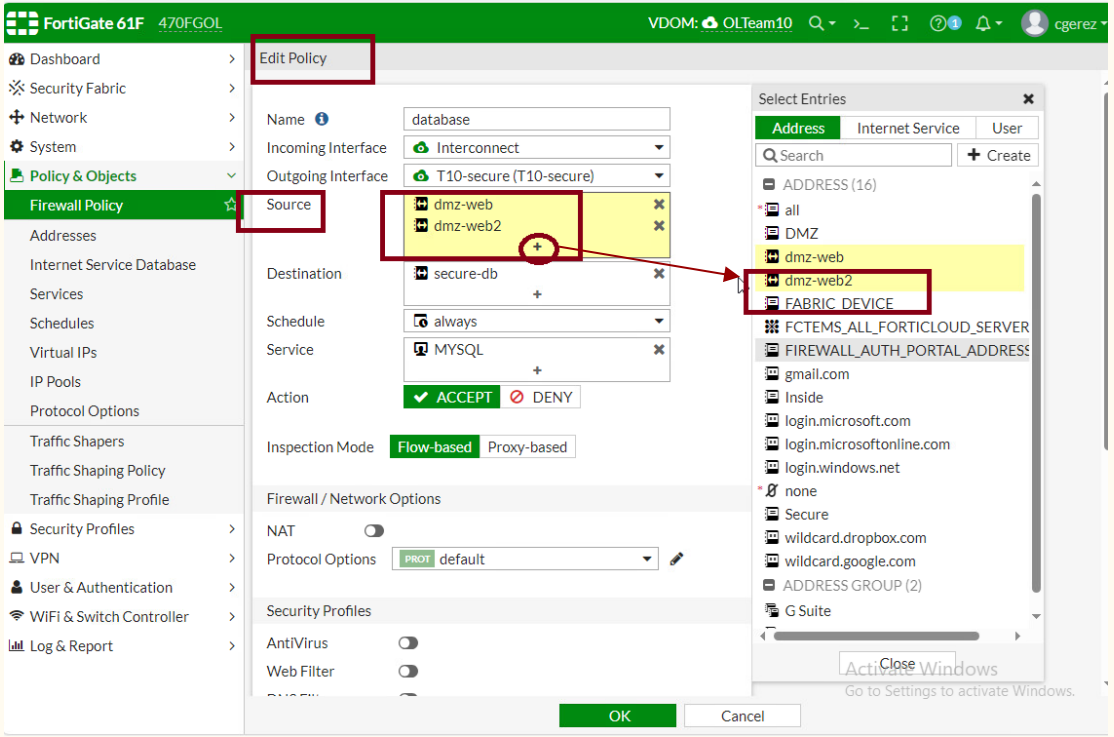
Configuring the app in the cloned machine.

Then select Firewall Policy tab and select the database rule by clicking into that line. Then select Edit.



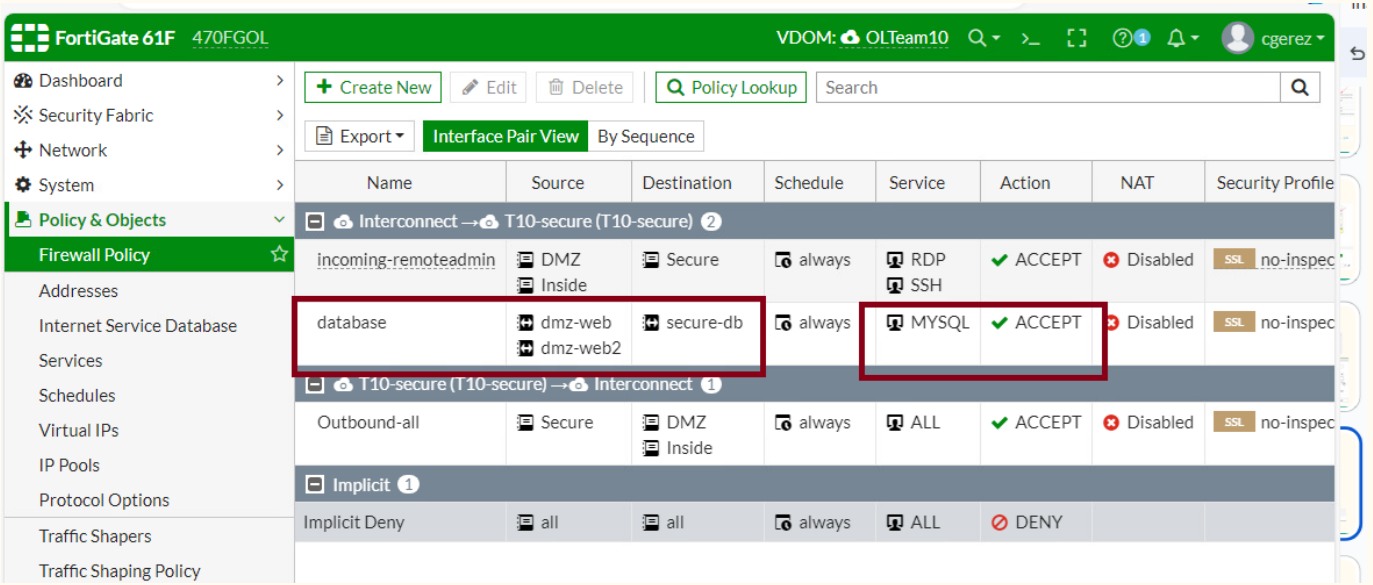
Configuring the app in the cloned machine.

A Edit Policy interface will be open and in the line of the Source select the + sign that opens another tab with a list. In that list select the new address (dmz-web2) that you created and finish with OK.



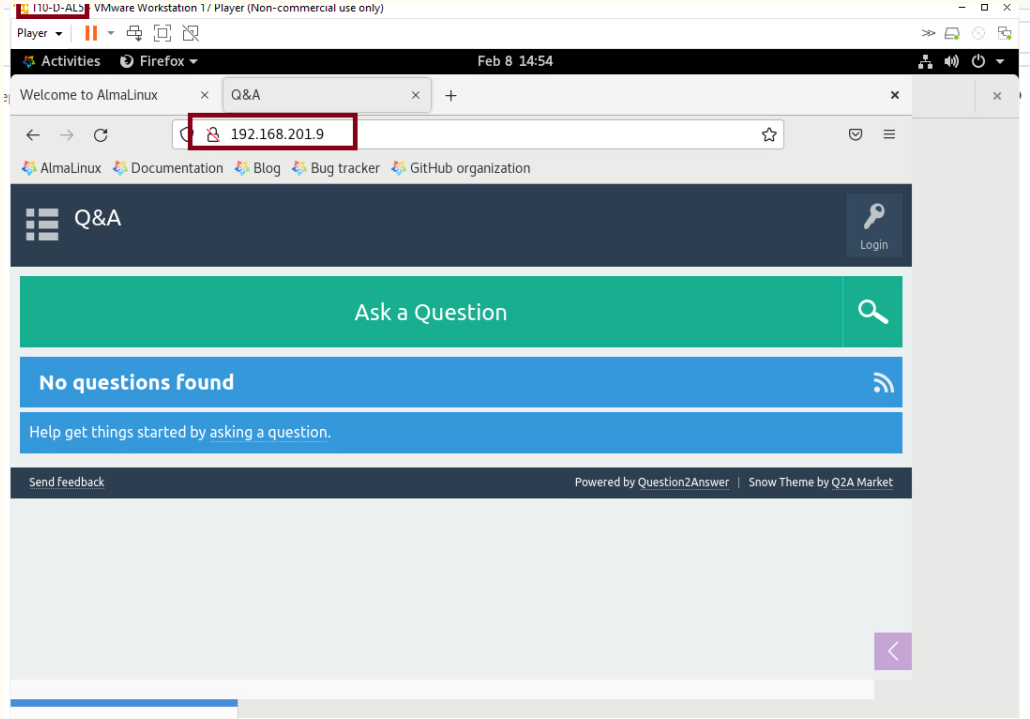
Configuring the app in the cloned machine.

This is how it should looks at the end. If the line still is colored , just reload the page with the option at the bottom of the creen, and it should look as in the picture.



Configuring the app in the cloned machine.

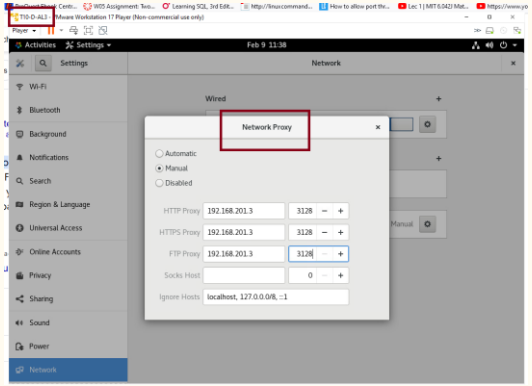
Now from any machine on the dmz zone you are able to start the app with the address of the new machine.Then, the app is working as it should from the DMZ zone now in the new cloned machine.

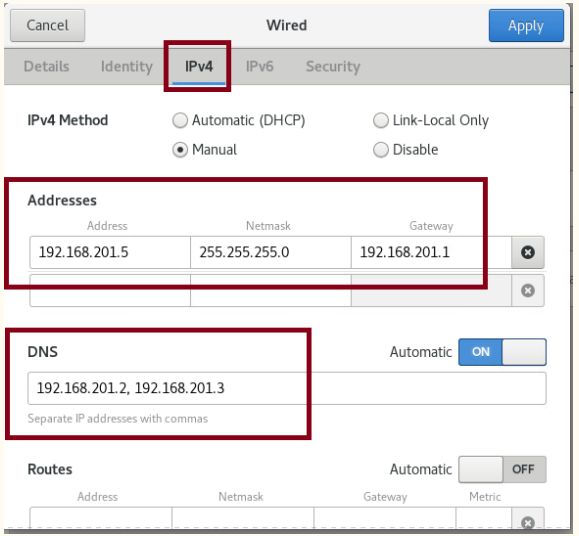


Configuring a load balancer in Alma Linux T10-D-AL3.

We have an extra machine created for testing purposes that is up and ready to install software, then we will use this machine. In case a new machine were necessary, refer to the first presentation on installing VM's.

Here you can see the machine settings for connectivity.





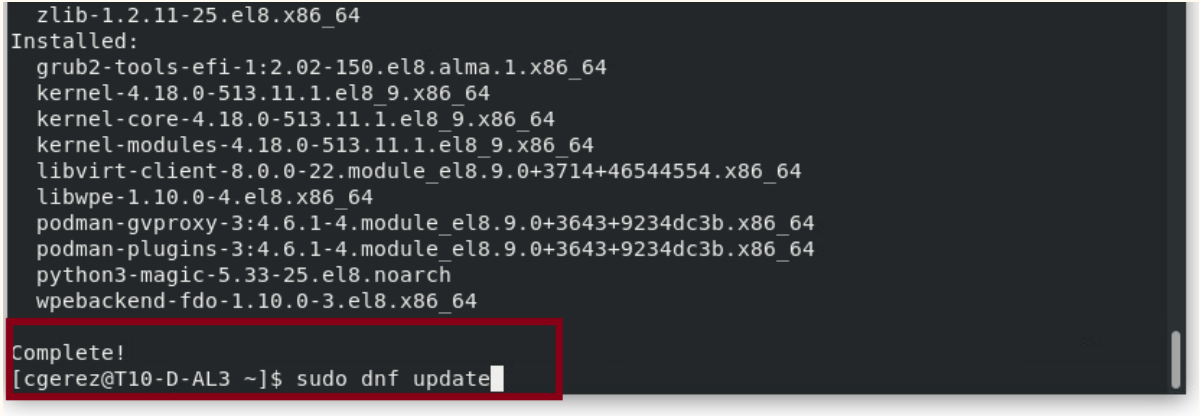
Configuring a load balancer in Alma Linux T10-D-AL3.

Since I already have internet connectivity, I first upgrade and update the system to start with the last version and patches.

we used:

```
sudo dnf upgrade almalinux-release
```

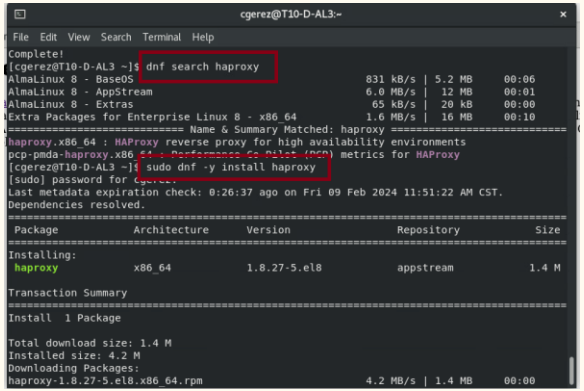
```
sudo dnf update
```

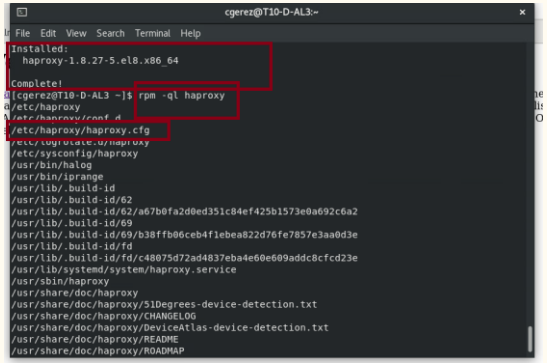


Configuring a load balancer in Alma Linux T10-D-AL3.

Here are the commands to find install and examine haproxy in alma linux.

```
dnf search haproxy

sudo dnf -y install haproxy

rpm -ql haproxy
```

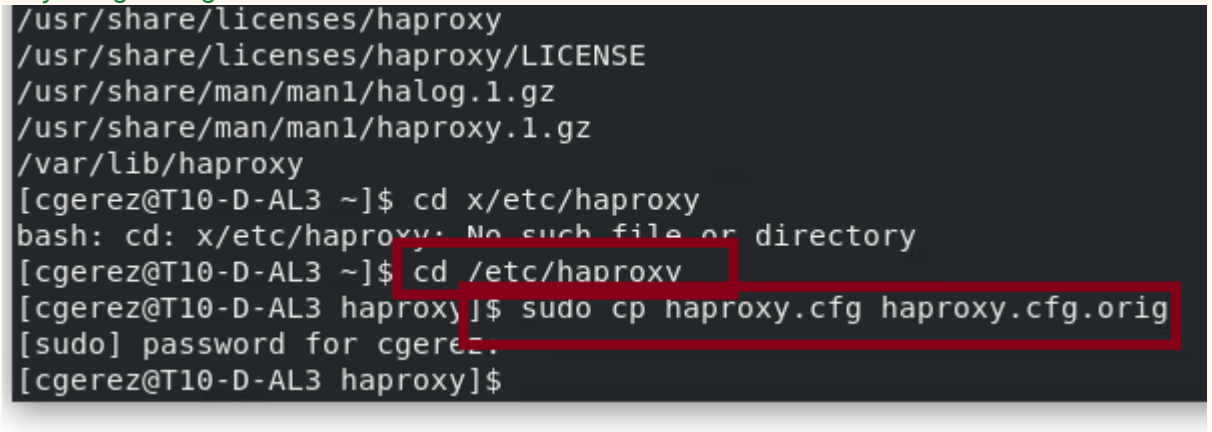




Configuring a load balancer in Alma Linux T10-D-AL3.

The configuration file `/etc/haproxy/haproxy.cfg`
usually contains a demo config, which we won't use.
For best-practices sake, make a backup of the config:

```
cd /etc/haproxy

sudo cp haproxy.cfg haproxy.cfg.orig
```



Configuring a load balancer in Alma Linux T10-D-AL3.

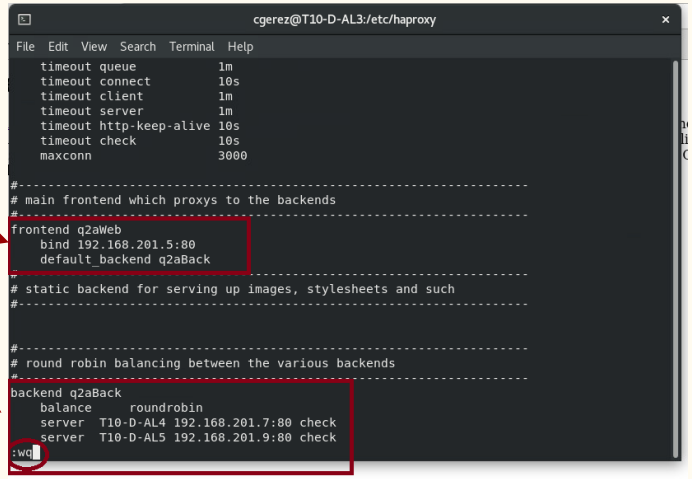
Edit the file with vi:

```
sudo vi haproxy.cfg
```

Find and delete all of the “frontend” and “backend” configuration stanzas, and replace them with a frontend and backend suitable to balance your original and cloned web servers.

```
frontend q2aWeb
    bind 192.168.201.5:80
    default_backend q2aBack


backend q2aBack
    balance roundrobin
    server T10-D-AL4 192.168.201.7:80 check
    server T10-D-AL5 192.168.201.9:80 check
```

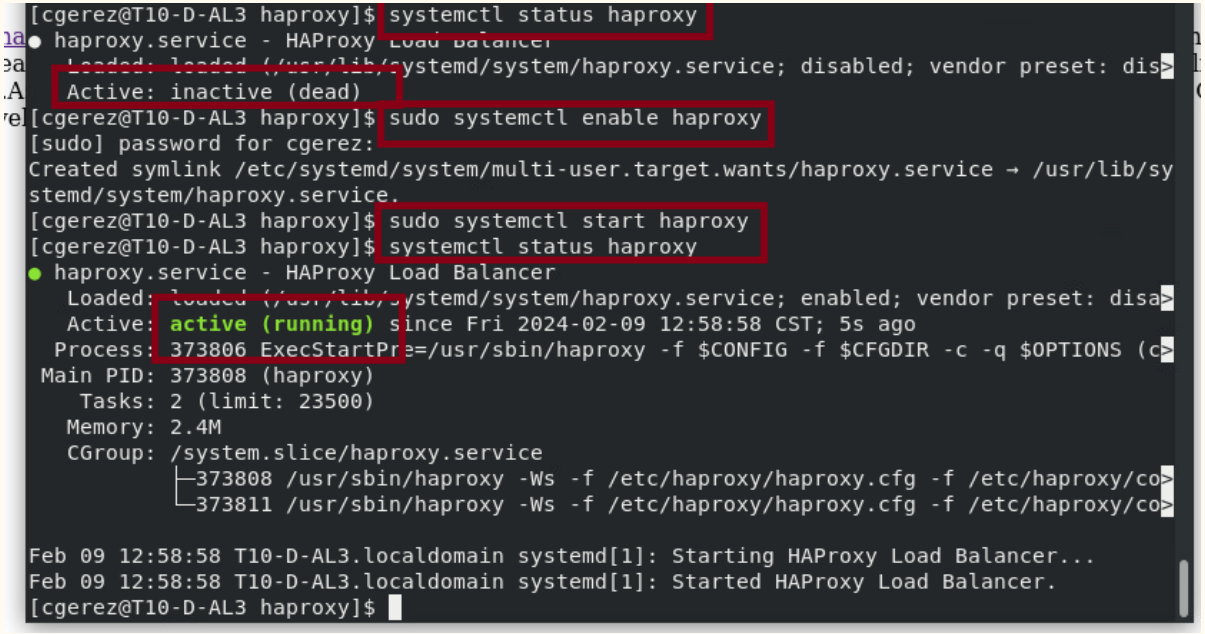
This configure round robin in our balancer, haproxy will listen for connections in this system and forwarder taking turns to each of the servers.



Configuring a load balancer in Alma Linux T10-D-AL3.

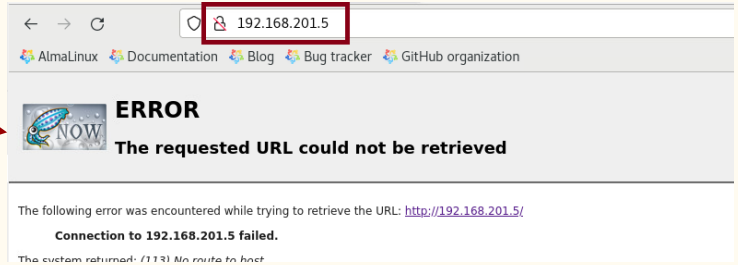
As all new services starts being disabled and inactive, we use this commands to check and start the service:

```
systemctl status haproxy

sudo systemctl enable haproxy

sudo systemctl start haproxy

systemctl status haproxy
```



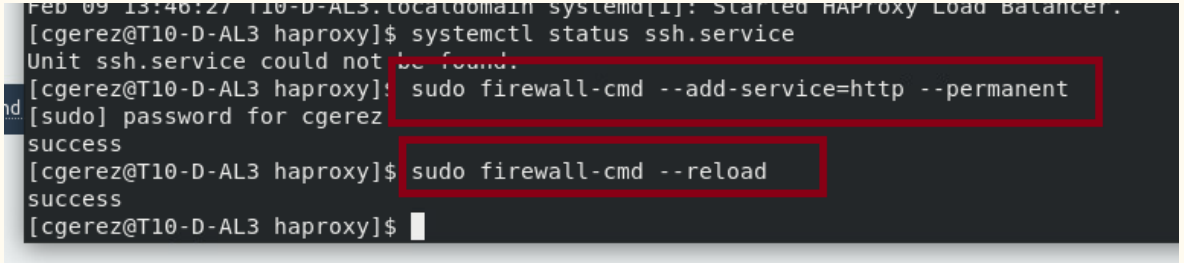
Configuring a load balancer in Alma Linux T10-D-AL3.

Is possible that you get this screen after all configurations are done. To solve this problem we has to configure the firewall in the balancer host machine to allow the http service with:




```
sudo firewall-cmd --add-service=http --permanent
```

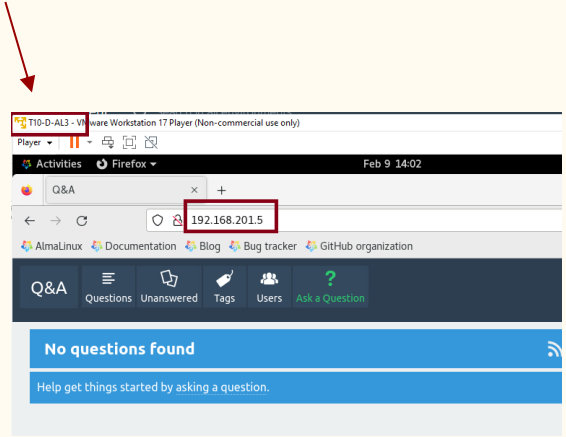
```
sudo firewall-cmd --reload
```

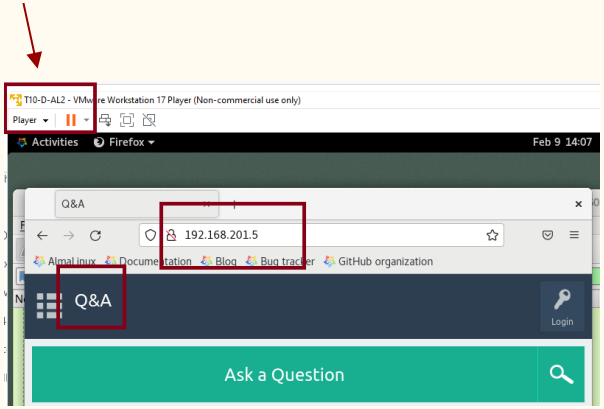


Configuring a load balancer in Alma Linux T10-D-AL3.

Now we can access the app through the load balancer in the ip 192.168.201.5

Here you can see access from the load balancer host and from another endpoint in the dmz.

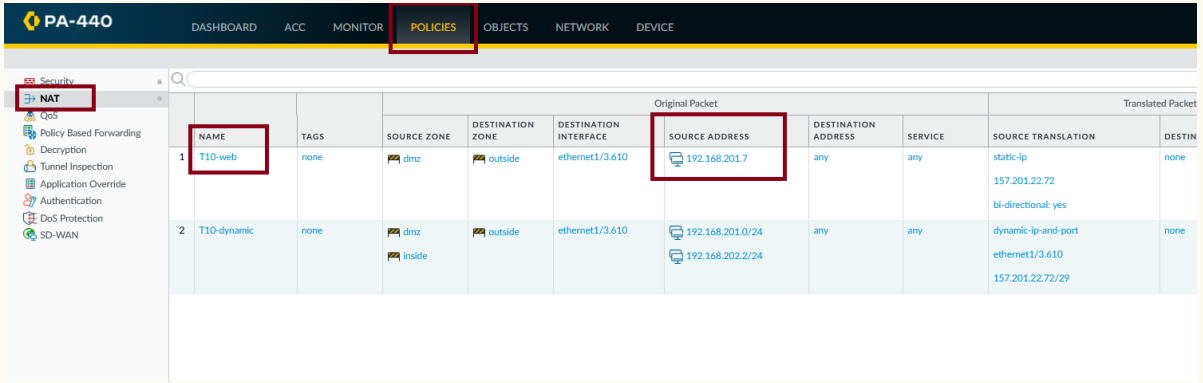


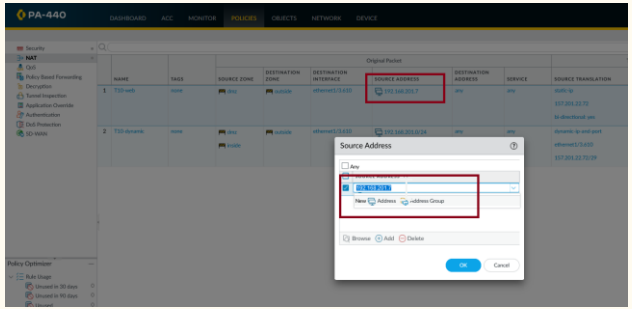


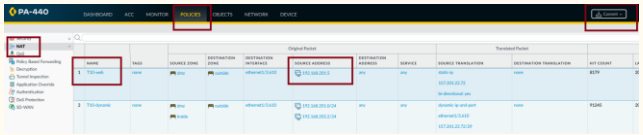
Test HAProxy, and verify that is balancing properly.

In the Palo Alto firewall we will adjust the rule to give clients in the internet to access the app. We are opening the balancer machine to untrust areas.

On NAT select the T10-web rule and change the source address to point to the load balancer hosting machine by selecting the address and change the address in the window that pop up. Remember to commit the changes.



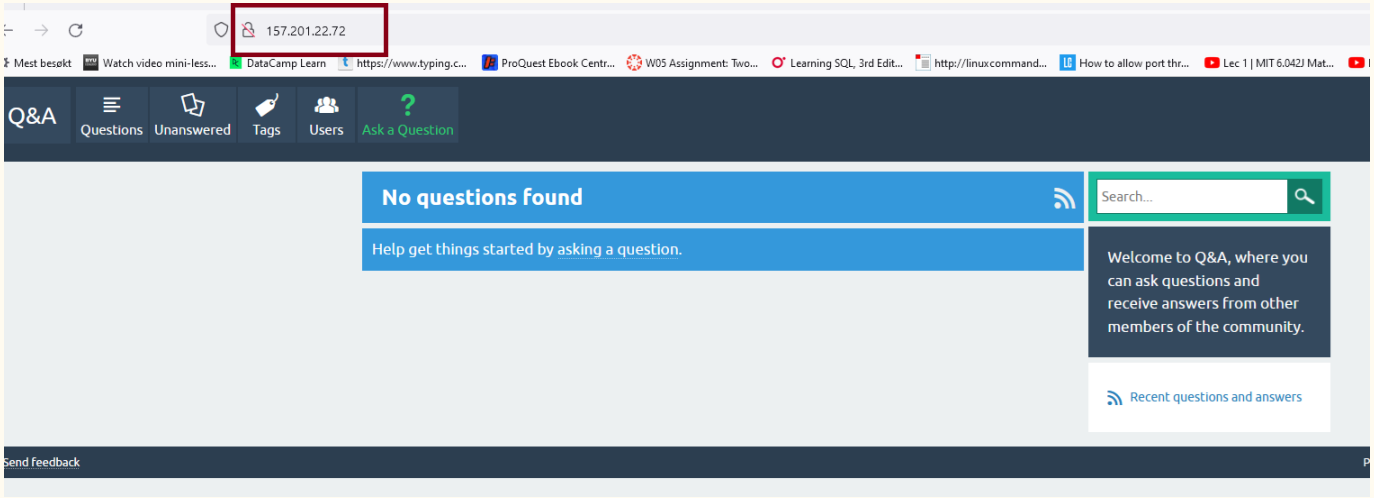




Test HAProxy, and verify that is balancing properly.

Now the app is accessible from the internet through the load balancer that alternate the servers use.

Now we explore several ways to test that this is occurring.

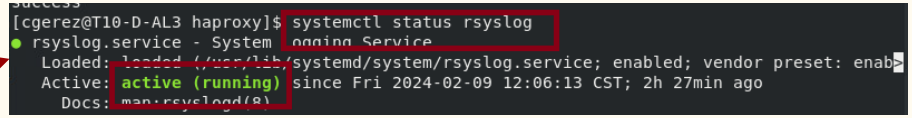


Test HAProxy, and verify that is balancing properly.

First option is use rsyslog to collect logs of the service.

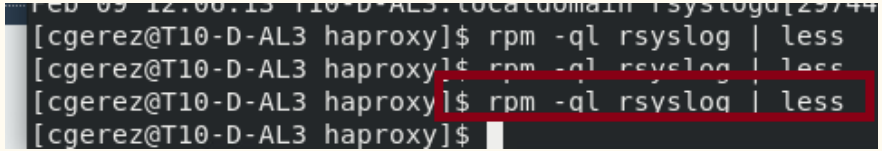
Check that rsyslog is working:

```
systemctl status rsyslog
```



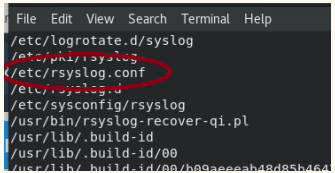
Find his configuration file

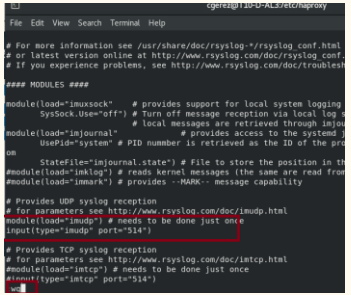
```
rpm -ql rsyslog | less
```



Edit his configuration file with vi and uncomment this lines:

```
sudo vi /etc/rsyslog.conf
```



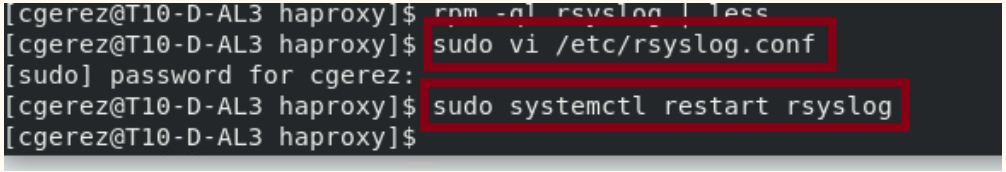



```
#module(load="imudp")
```

```
#input(type="imudp" port="514")
```

Then restart the service and check on the logs.

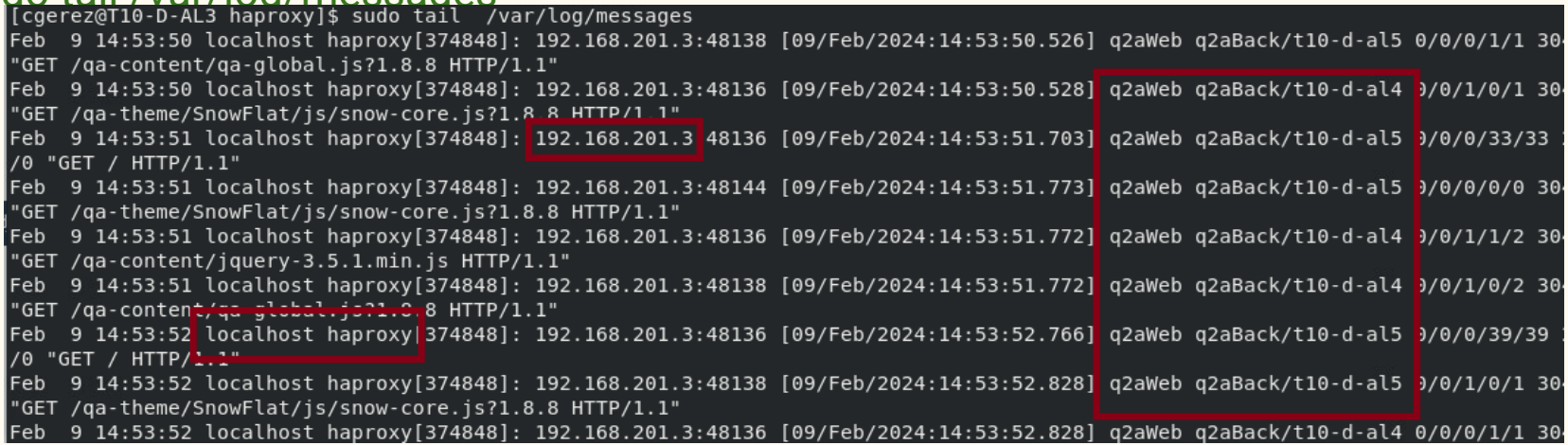
```
sudo systemctl restart rsyslog
```



Test HAProxy, and verify that is balancing properly.

With the previous configuration one can see the logs generated by the app with this command. The request take turns in each server.You can see machines al4 and al5 alternate.

sudo tail /var/log/messages



Test HAProxy, and verify that is balancing properly.

As a good practice we will redirect the logs to a separate file. We edit again the configuration file:

```
sudo vi /etc/rsyslog.conf
```

Find this rule:i

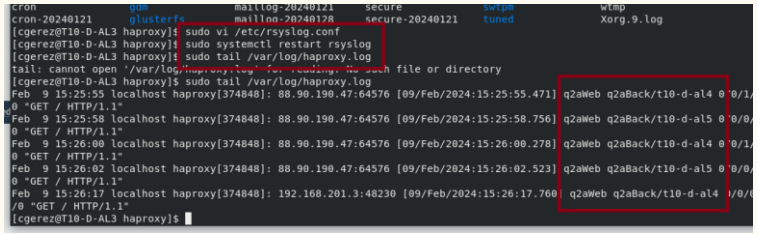
```
*.info;mail.none;authpriv.none;cron.none    /var/log/messages
```

And change it to:

```
*.info;mail.none;authpriv.none;cron.none;local2.none    /var/log/messages
```

And add a new rule that sends local2 facility to haproxy.log

```
local2.*    /var/log/haproxy.log
```



As before save and restart the service will apply the changes.

```
sudo systemctl restart rsyslog
```

Test HAProxy, and verify that is balancing properly.

There are 2 optional ways to check the alternate of the servers on this task. Today I will leave here. If you feel to continue are welcome. Look ar the end of the class tutorial where stay optional.

(***Optional***) Syslogs and event logs are reliable tools for troubleshooting server software, but sometimes it just feels more satisfying to see load balancing evidence in the web browser client. Here's one way to do this:

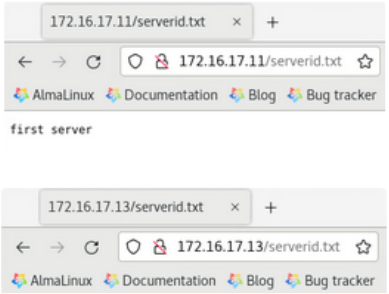
On the first web server, create a "static page" file that contains some identifying content. Example:

```
echo "first server" | tee -a /var/www/html/serverid.txt
```

On the second web server, create the same file, but put different content in that file:

```
echo "second server" | tee -a /var/www/html/serverid.txt
```

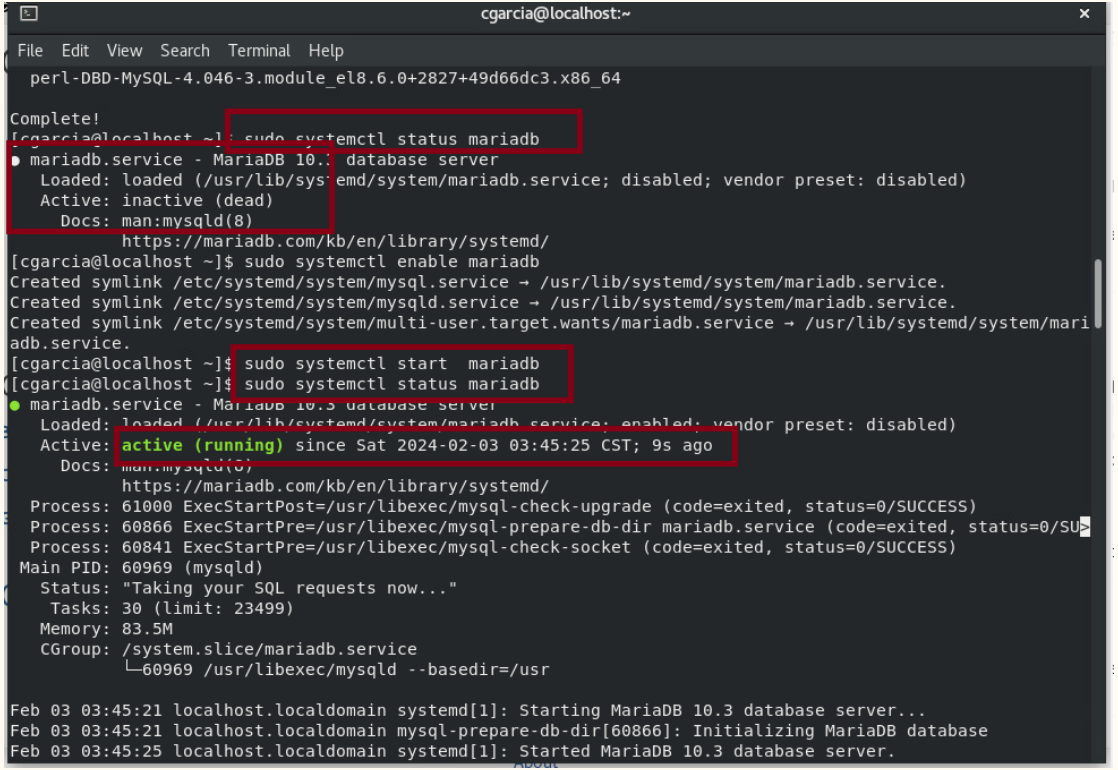
Launch a browser on a DMZ VM, and verify the contents each identifier file on its respective host:



Challenges we faced

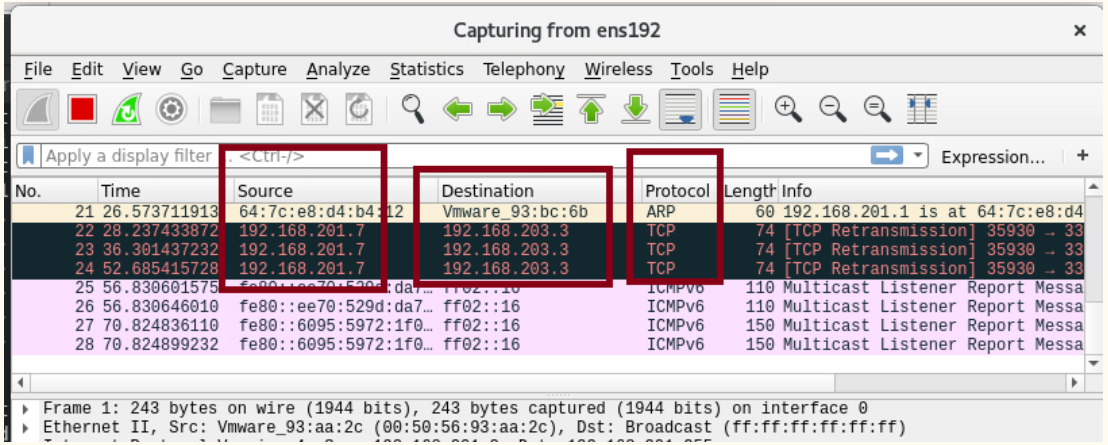
If mariadb is not running use the following commands.

Use the command sudo systemctl status mariadb to look if is enable and had start. If not use the same command changing to enable and start, to start the service.



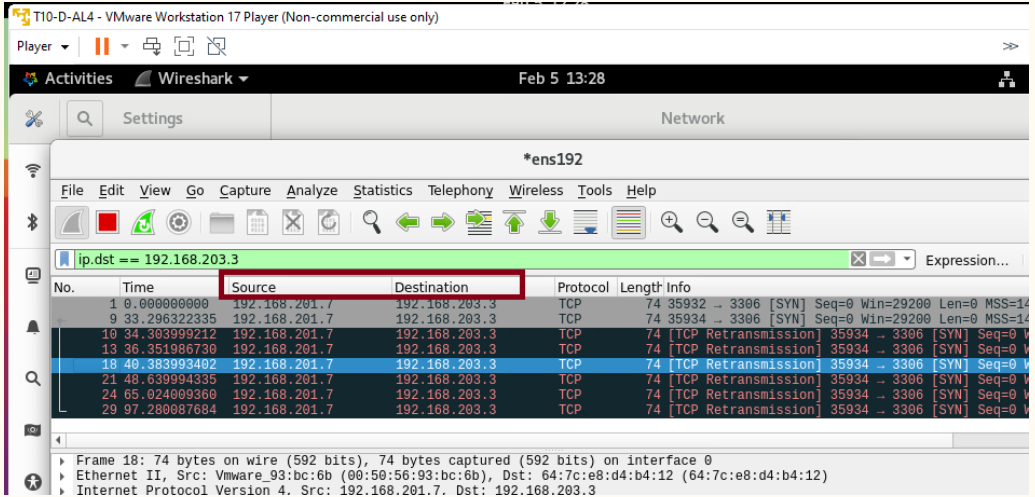
We experienced connections problems from the DMZ with the database in the secure zone.

We used wireshark to test why we didn't have app connections between the DMZ and the secure zone. The firewall was already configured with a policy to allow mysql.



Connections problems from the DMZ zone.

Filtering our capture from Wireshark on the apache DMZ machine we saw that the packages were send but somehow not receiving responses. Black color is an indicator of something not working in the connections.

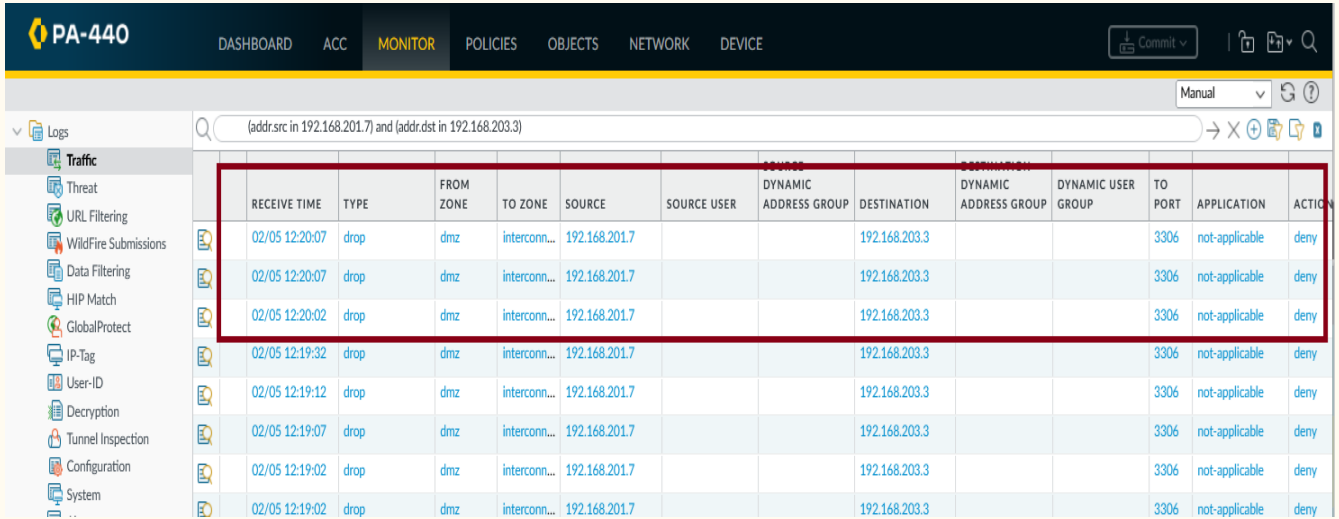


In the Palo Alto firewall the packages were dropped, and not allowed.

Checking in the Palo Alto monitor tab and filtering between the 2 address that we were interested in, we find some dropped packages when trying the connections.

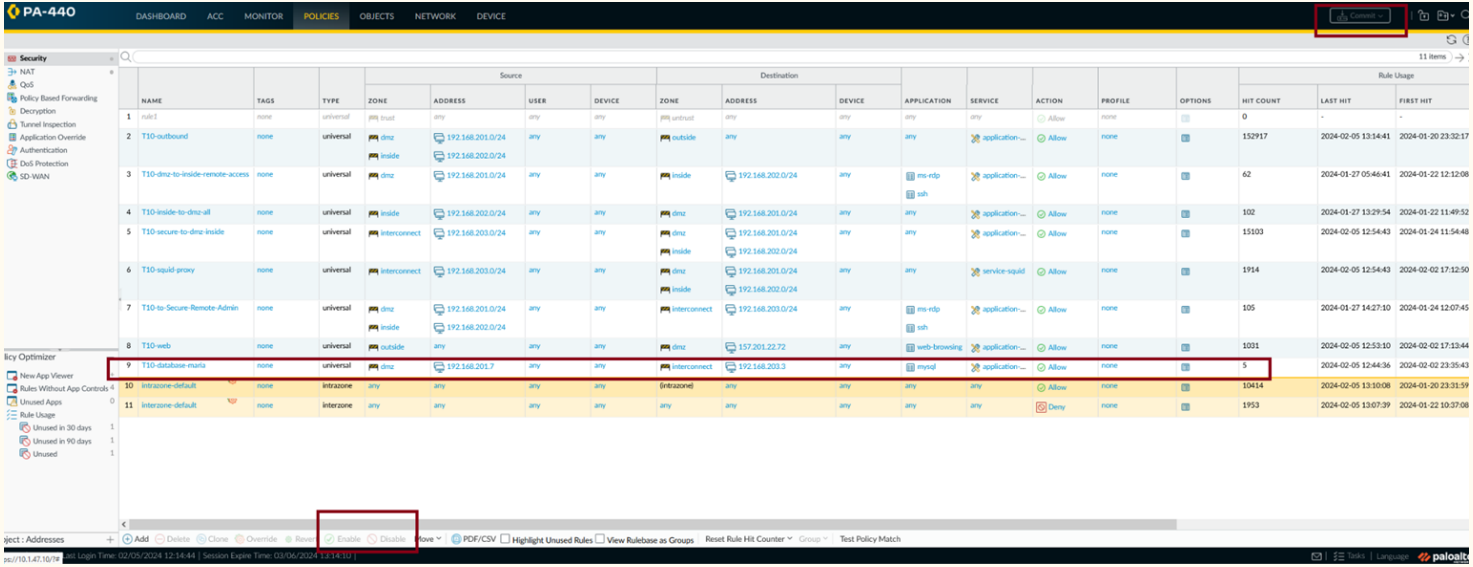
Then we review the policy and found that was disabled.

After enable and commit it start to work again. See next slide.



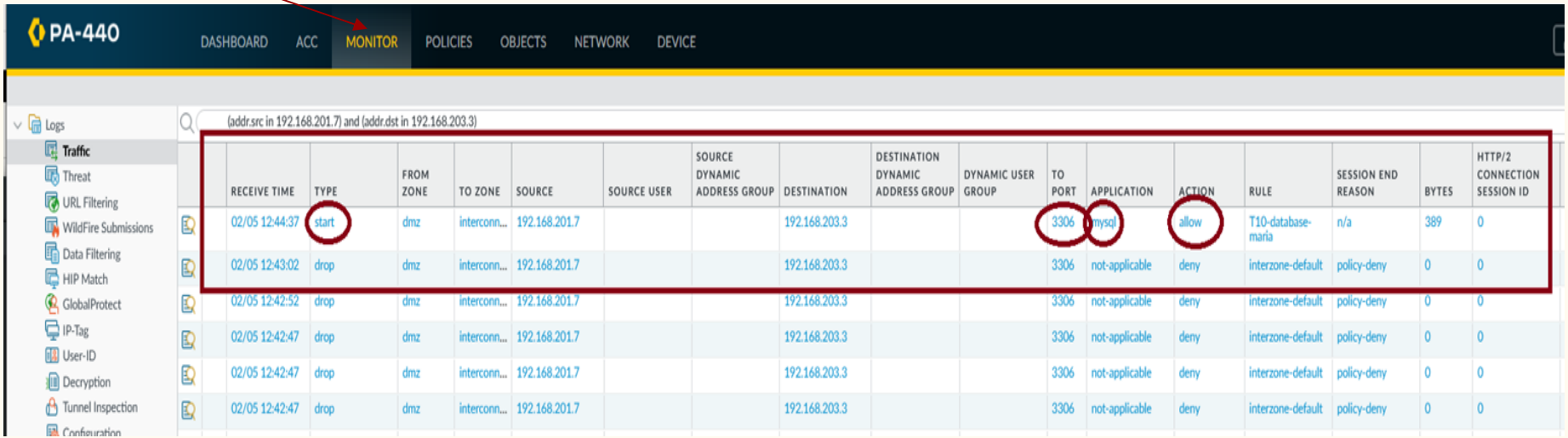
The allow policy to mysql from DMZ to secure working.

Is always good to know where to find the tabs to enable or disable a policy to troubleshoot, and remember to commit changes.Here in policies tab and after highlight the rule that is grey (disabled), we use the bottom line tab enable to get it to the right blue color again. But remember that not change is done before we committed.



Results before and after the policy was enabled.

The packages now flow through the firewall as we can see in the monitor of the firewall.



RECEIVE TIME	TYPE	FROM ZONE	TO ZONE	SOURCE	SOURCE USER	SOURCE DYNAMIC ADDRESS GROUP	DESTINATION	DESTINATION DYNAMIC ADDRESS GROUP	DYNAMIC USER GROUP	TO PORT	APPLICATION	ACTION	RULE	SESSION END REASON	BYTES	HTTP/2 CONNECTION SESSION ID
02/05 12:44:37	start	dmz	interconn...	192.168.201.7			192.168.203.3			3306	mysql	allow	T10-database-maria	n/a	389	0
02/05 12:43:02	drop	dmz	interconn...	192.168.201.7			192.168.203.3			3306	not-applicable	deny	interzone-default	policy-deny	0	0
02/05 12:42:52	drop	dmz	interconn...	192.168.201.7			192.168.203.3			3306	not-applicable	deny	interzone-default	policy-deny	0	0
02/05 12:42:47	drop	dmz	interconn...	192.168.201.7			192.168.203.3			3306	not-applicable	deny	interzone-default	policy-deny	0	0
02/05 12:42:47	drop	dmz	interconn...	192.168.201.7			192.168.203.3			3306	not-applicable	deny	interzone-default	policy-deny	0	0
02/05 12:42:47	drop	dmz	interconn...	192.168.201.7			192.168.203.3			3306	not-applicable	deny	interzone-default	policy-deny	0	0

After enable the firewall policy that allow transmission. Connections were established.

This is a screenshot of wireshark monitor that shows the reconnected status. Is filtered by the destination address 192.168.203.3