# Global Protect Basic lab use case

This lab activity delves into the use case of ansible ad-hoc commands using the RAW Module for connecting to devices that don’t have python enabled/supported. Ansible is an open-source automation tool that can manage the configuration of network devices. Ad-hoc commands in Ansible are commands that can be run individually to perform quick tasks, instead of executing playbooks which contain a series of tasks.

In order to run Ansible ad-hoc commands, a couple of things are needed:

1. Ansible installed on a control node - This can be your personal computer or a dedicated server. This control node communicates with your network devices. In this lab we use GNS3 provided automation appliance. The GNS3 network container provides the popular tools used for network automation: Netmiko, NAPALM, Pyntc, and Ansible.

2. Python - Ansible is written in Python and requires Python on the control node. (Again because we are using the Network appliance we don’t have to worry about this)

3. Network modules - Ansible uses network modules (like ios\_command for Cisco IOS devices) to interact with network devices.

Here's a sample lab with Ansible ad-hoc commands used on Cisco devices:

The main steps in the lab are:

1. \*\*Install Ansible on the control node\*\*: You can do this via a package manager like apt or yum. This lab assumes you have the control node setup

2. \*\*Set up the host file\*\*: Ansible uses an inventory file to track which hosts are part of your network.

3. \*\*Test connectivity to the router/switch\*\*: You can ping the Cisco router using Ansible's `ping` module.

4. \*\*Run commands on the router\*\*: You can use ad-hoc commands to run commands on the router.

5. \*\*Filter output\*\*: You can filter the output you get back from Ansible to find what you want.

6. \*\*Save output\*\*: You can save output to a file using the ad-hoc commands

We will use the following topology:

![](media/842293be652a76a123dff255f4e69ff9.png)

Here are some of the information on lab devices

|  |  |
| --- | --- |
| **Hostname** |  |
| **Network Automation** | GNS3 Network appliance @192.168.100.1 |
| **S1** | Cisco Switch running vios\_l2 Version 15.2 @192.168.100.10 |
| **S2** | Cisco Switch running vios\_l2 Version 15.2 @192.168.100.20 |
| **S3** | Cisco Switch running vios\_l2 Version 15.2 @192.168.100.30 |
| **S4** | Cisco Switch running vios\_l2 Version 15.2 @192.168.100.40 |
| **S5** | Cisco Switch running vios\_l2 Version 15.2 @192.168.100.50 |

Prerequires:

* GNS3
* Ansible appliance
* Cisco images for GNS3

## Step 1: Configure Palo Alto

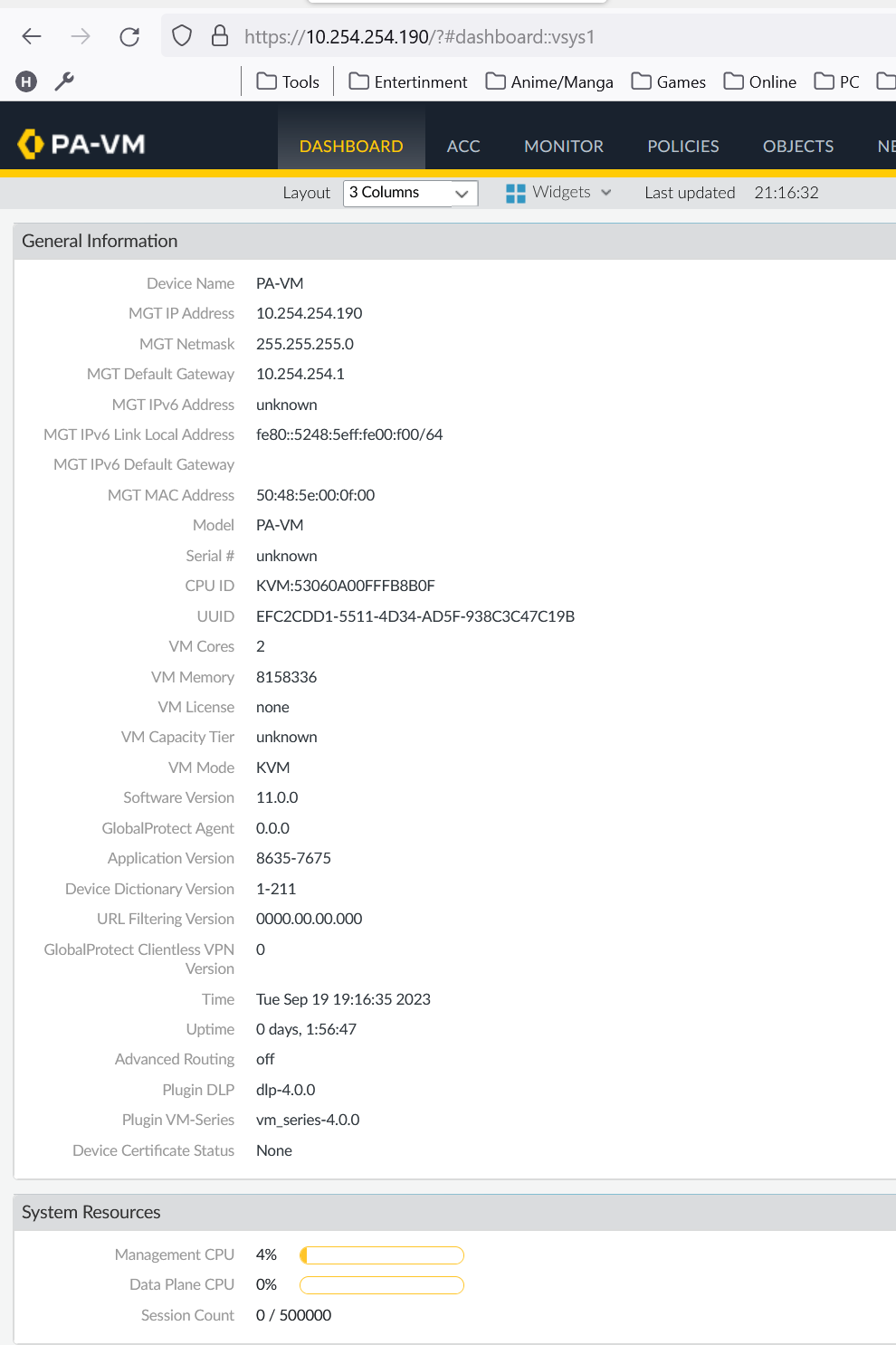
First we have to configure the the firewall for GUI access. Use the below command and make sure that the address you use is part of your network;

nset deviceconfig system ip-address 10.254.254.190 netmask 255.255.255.0 default-gateway 10.254.254.1 type static

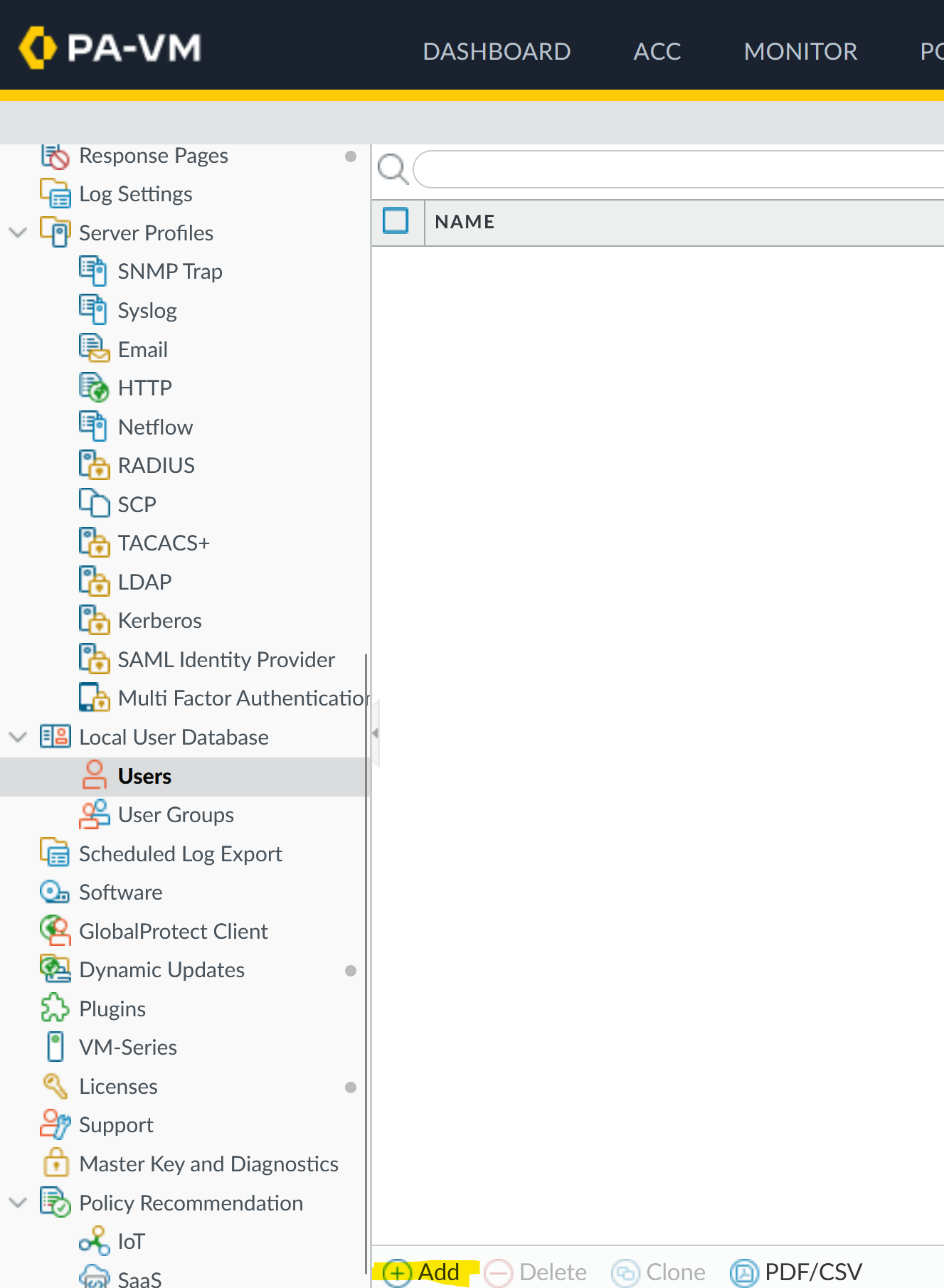
## 

Visit your palo alto through the GUI using https (use the ip you just configured)

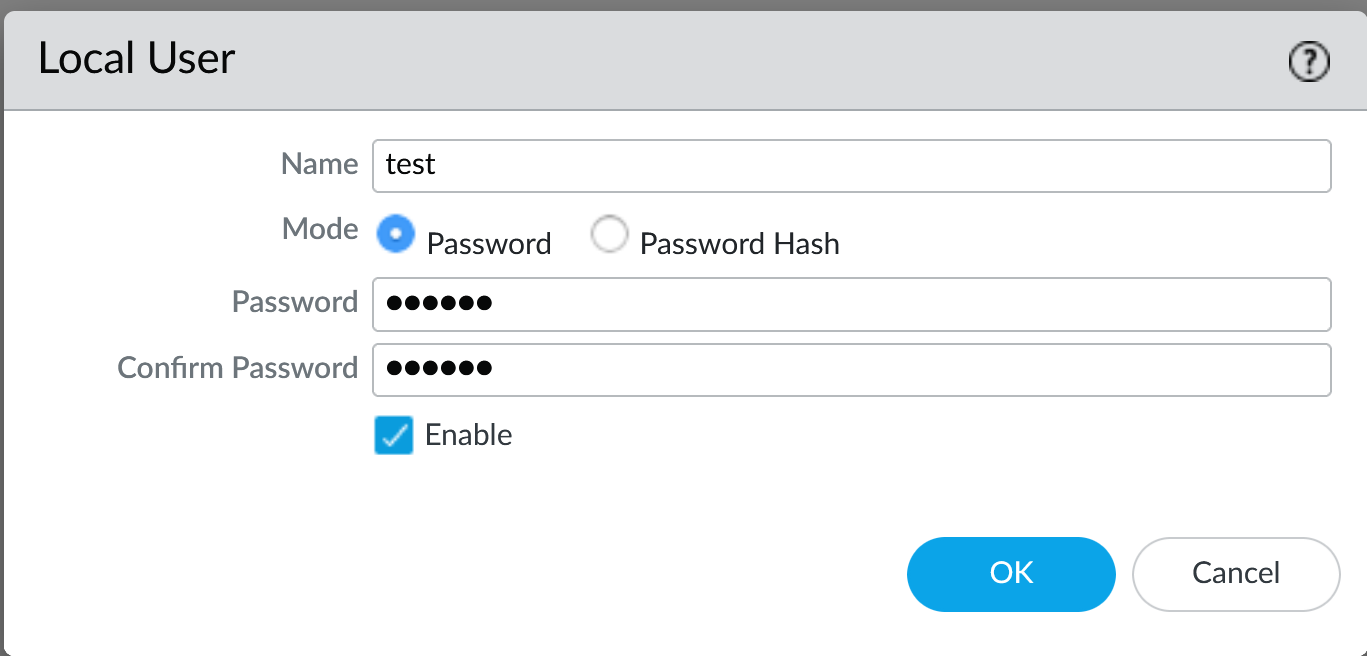
https://10.254.254.190



2. Create the local user. This is needed since we are going to use local user for Global protect

* Navigate -> Device -> Local User Database -> Users
* Then click on Add
* 

Create a local user for testing use. I used Name: test with a password: 12345678



1. **Define Authentication profile.**

* Navigate -> Device Authentication Profile [ Then click on Add]
* Name: GP-Auth-Profile
* Type: Local Database
* Navigate -> Advanced -> Add

## 

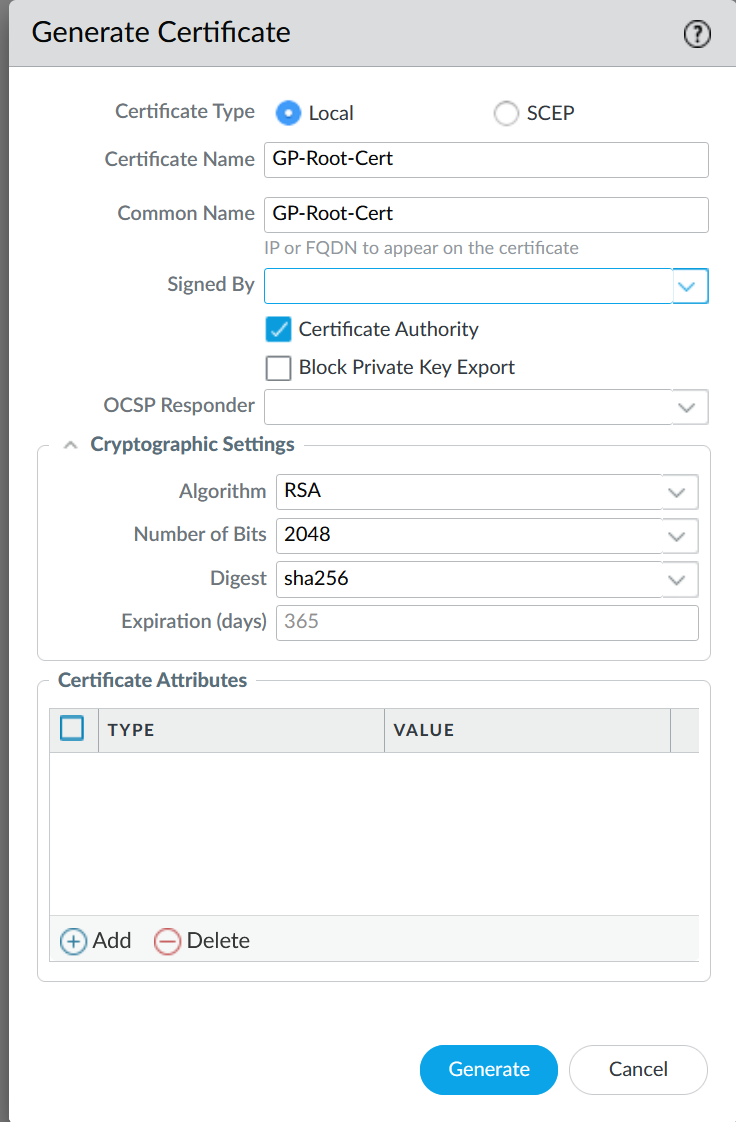
Commit your changes

## Step 2: Generate Certificate

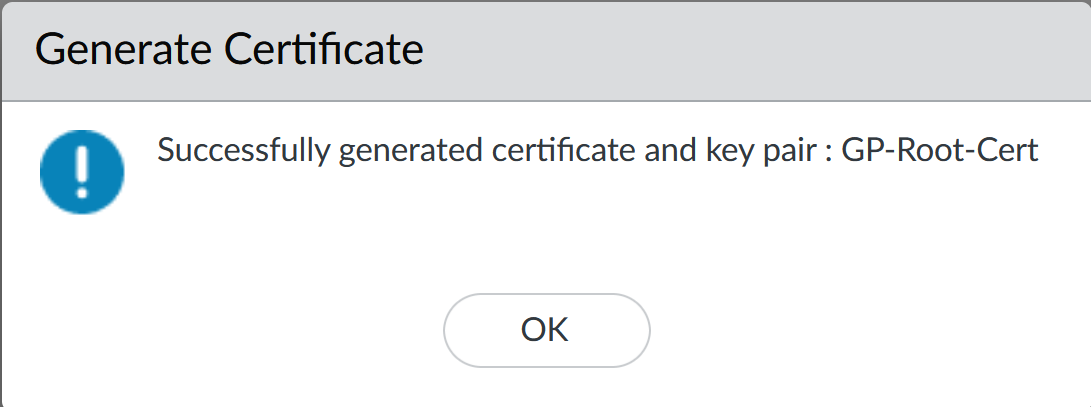
We have to generate a certificate and than export and add it into our browser.

* Navigate -> Device -> Certificate Management -> Certificates
* [ Then click on Generate]

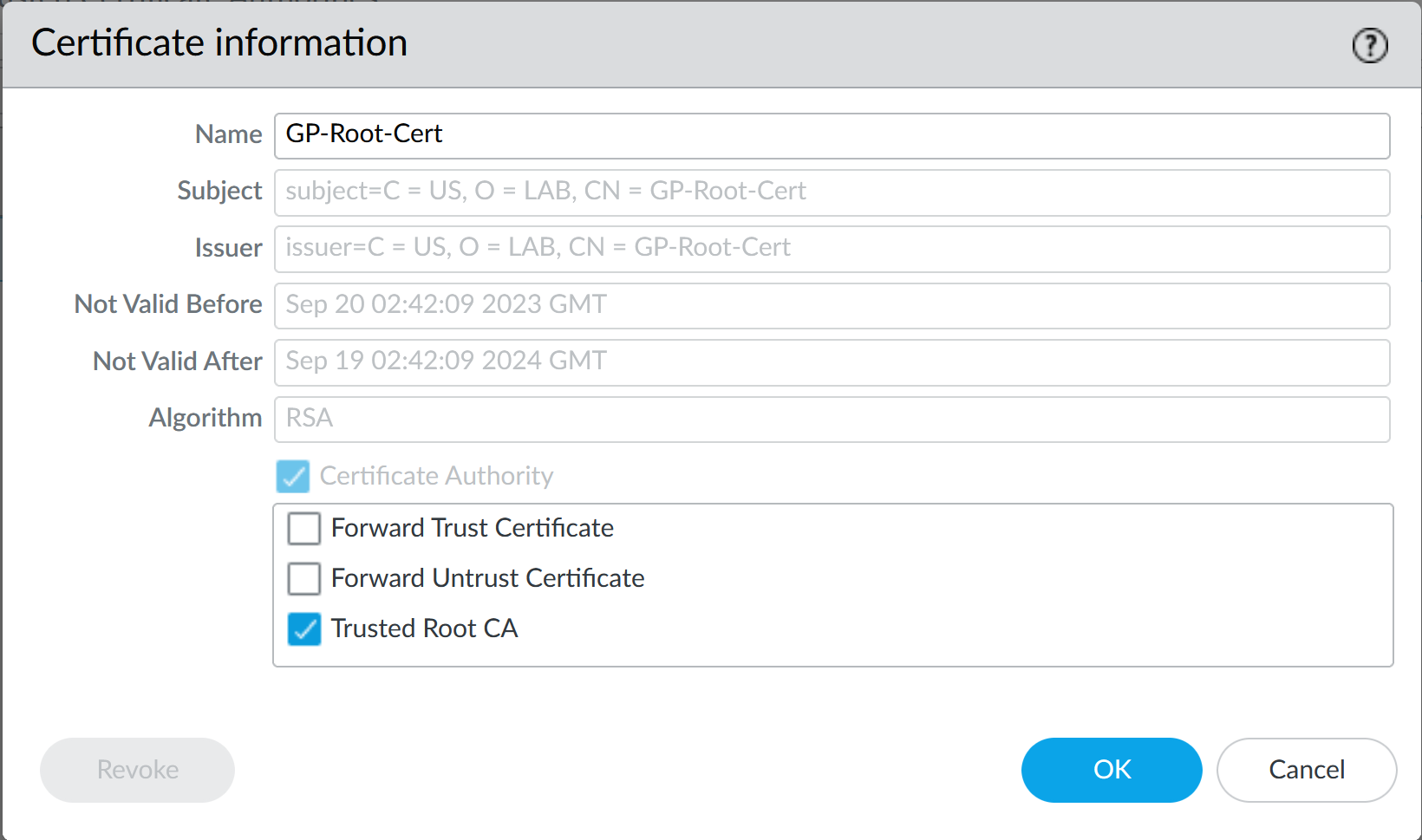
|  |  |
| --- | --- |
| Certificate Name: | GP-Root-Cert |
| Common Name: | GP-Root-Cert |
| Certificate Authority | Make sure this is checked |
| Certificate Attrubutes: | Add in what you want |



Click on Generate to Generate the certificate and you should get the below



Click on the Generated Cert and Make sure to check the box “Trusted Root CA”

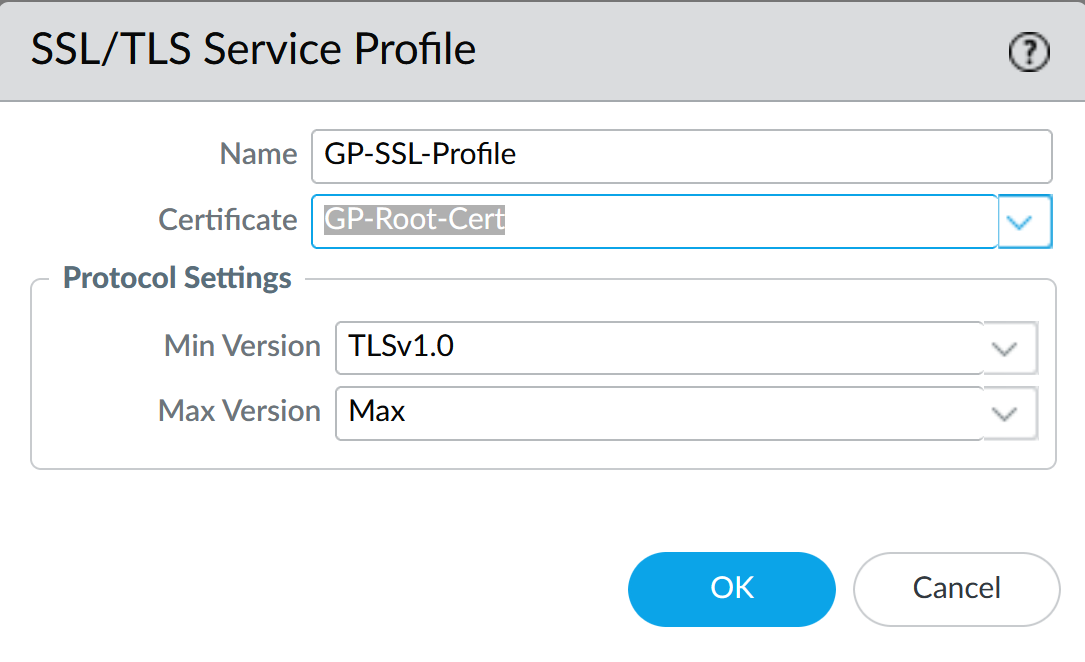


2. **Create SSL service Profile. This will be needed for users trying to connect**

Navigate -> Device -> Certificate Management -> SSL/TLS Service Profile

[ Then click on Add]

|  |  |
| --- | --- |
| Name: | GP-SSL-Profile |
| Certificate | GP-Root-Cert |

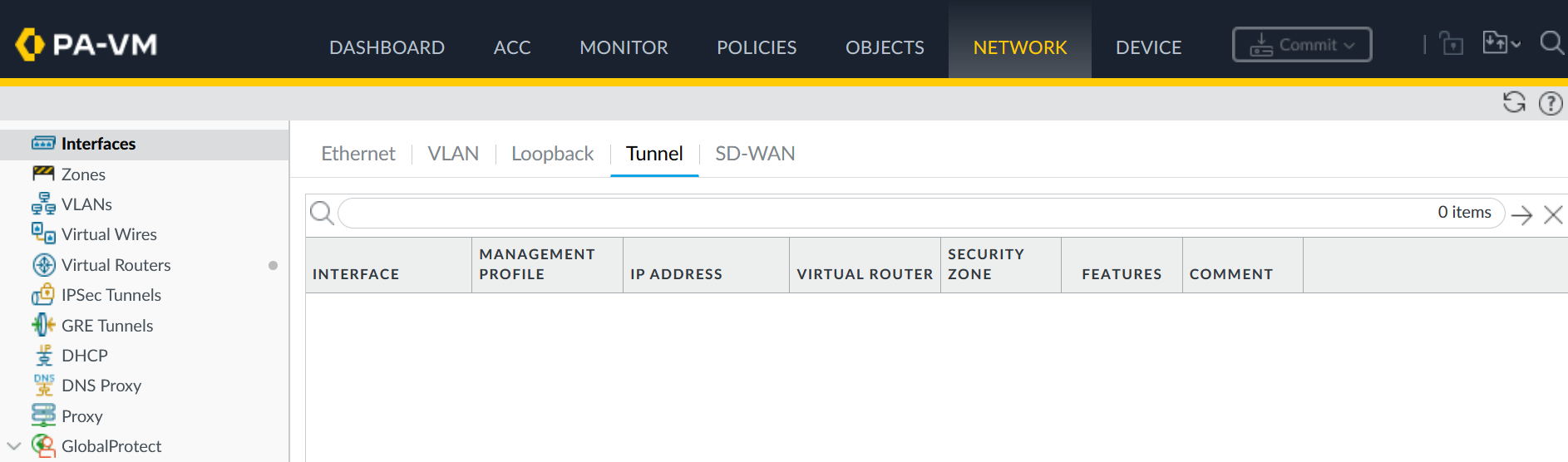


Click on “OK”

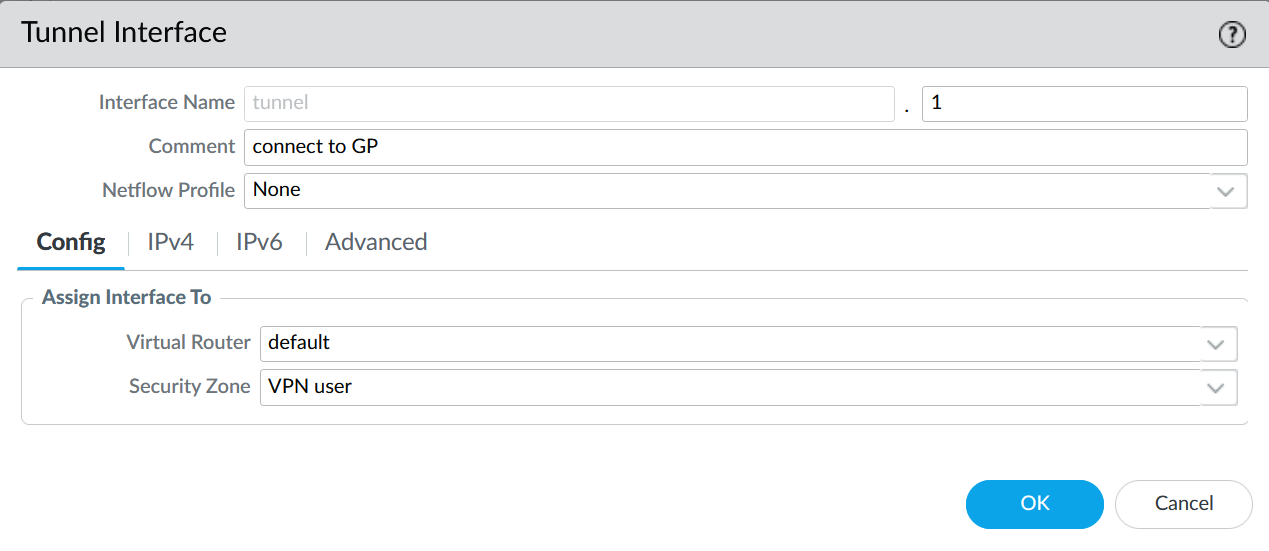
Commit your changes

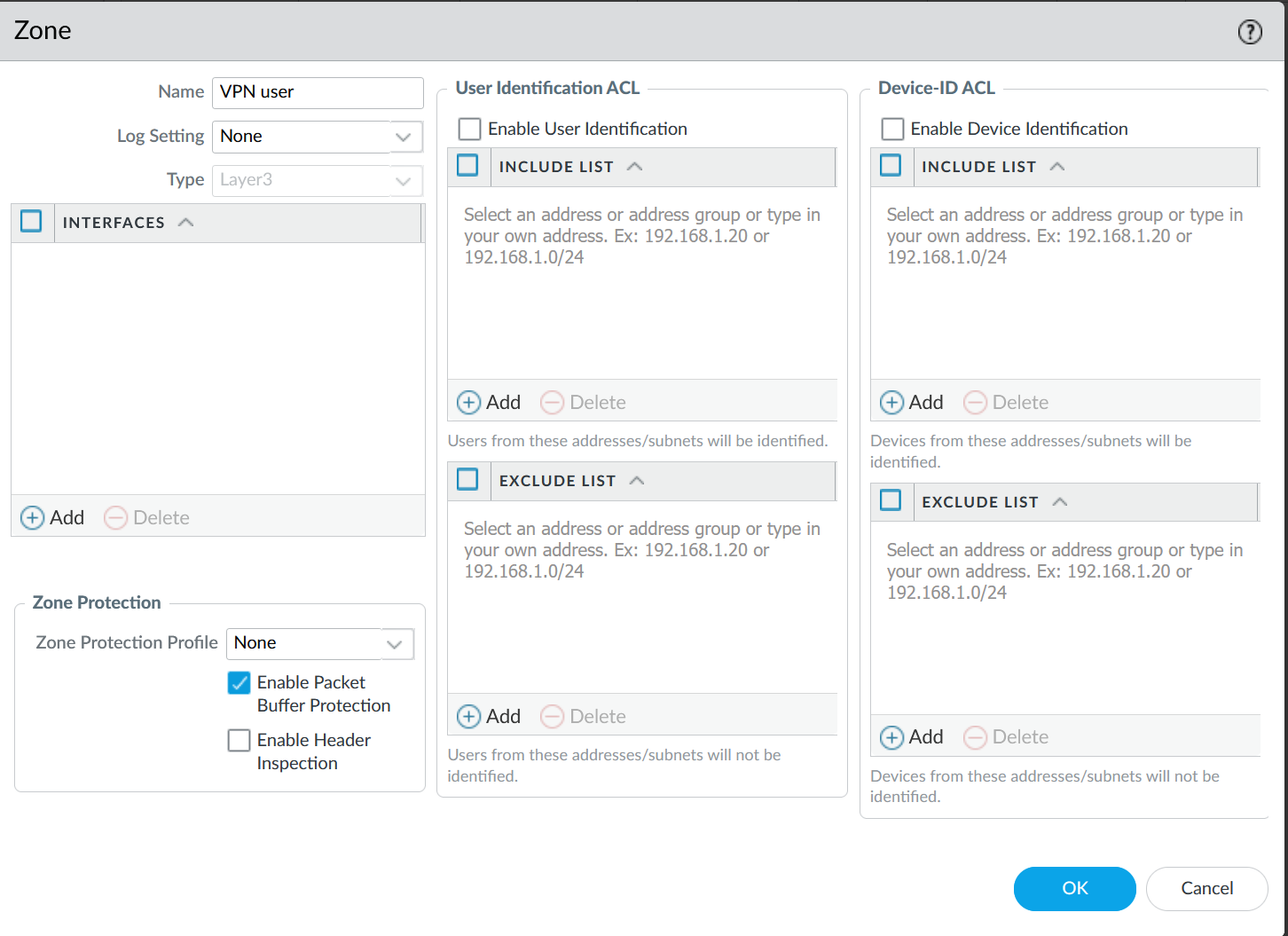
## Step 3: Configure tunnel and Interfaces

Navigate to Network -> Interfaces -> Tunnel ( Then click on Add)



|  |  |
| --- | --- |
| Interface Name: | 1 |
| Comment: | Connect to GP |
| Virtual Router: | default |
| Security Zone: | Create new zone (VPN user) |



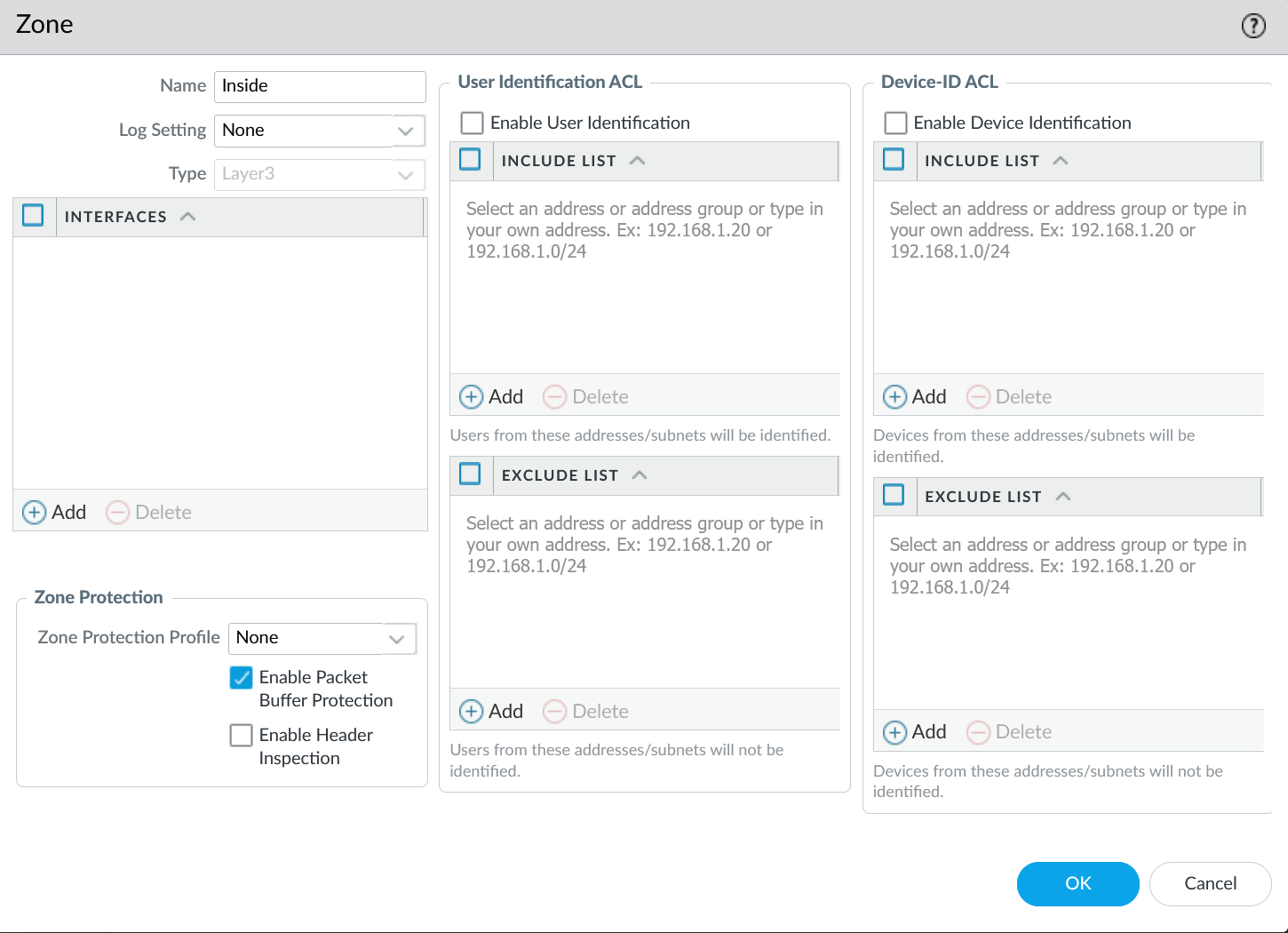


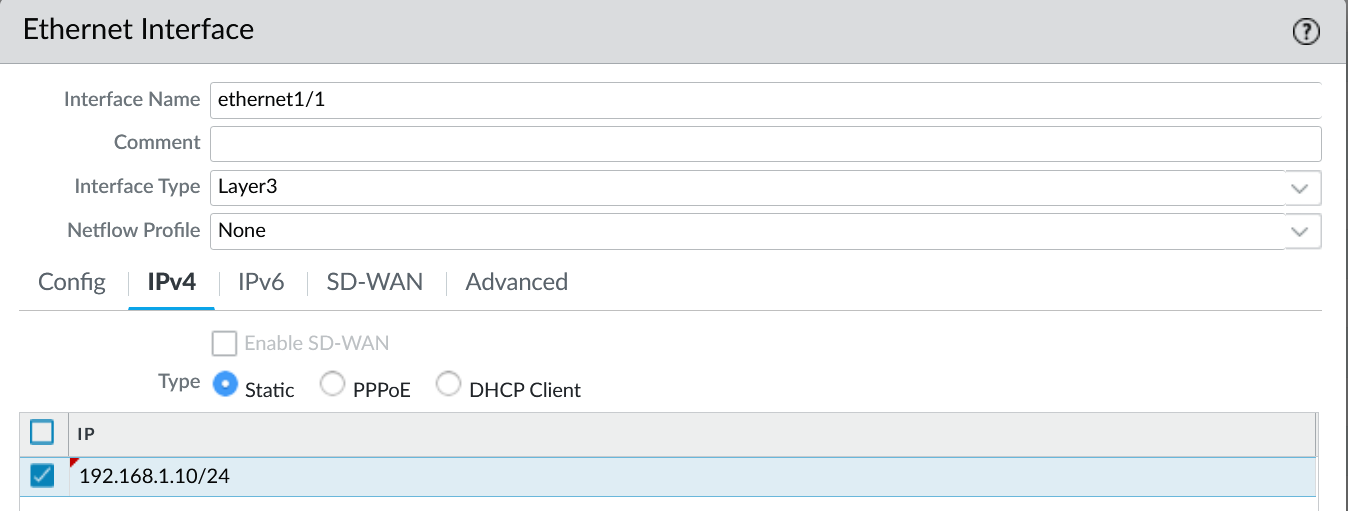
Click “OK”

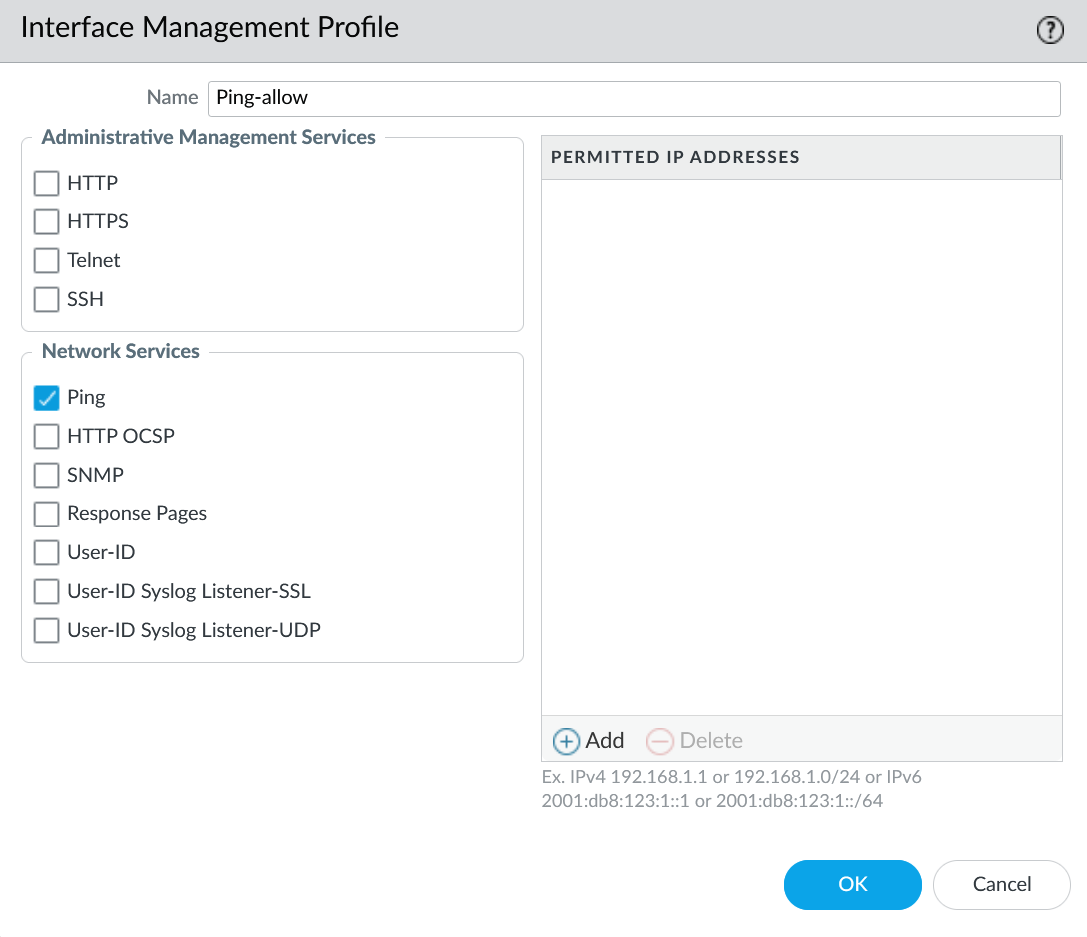
Configure Interfaces

Navigate to Network -> Interfaces -> Ethernet1/1

|  |  |
| --- | --- |
| Config -> Interface Type | Layer3 |
| Config -> Virtual Router | default |
| Config -> Security Zone | Inside (Create) |
| IPV4 -> static | 192.168.1.10/24 |
| Advanced ->Management Profile | Ping-allow (create and check only ping) |







Navigate to Network -> Interfaces -> Ethernet1/2

|  |  |
| --- | --- |
| Config -> Interface Type | Layer3 |
| Config -> Virtual Router | default |
| Config -> Security Zone | Outside (Create) |
| IPV4 -> static | 200.1.1.10/30 |
| Advanced ->Management Profile | Ping-allow |

Navigate to Network -> Interfaces -> Ethernet1/3

|  |  |
| --- | --- |
| Config -> Interface Type | Layer3 |
| Config -> Virtual Router | default |
| Config -> Security Zone | DMZ (Create) |
| IPV4 -> static | 10.1.1.10/24 |
| Advanced ->Management Profile | Ping-allow |

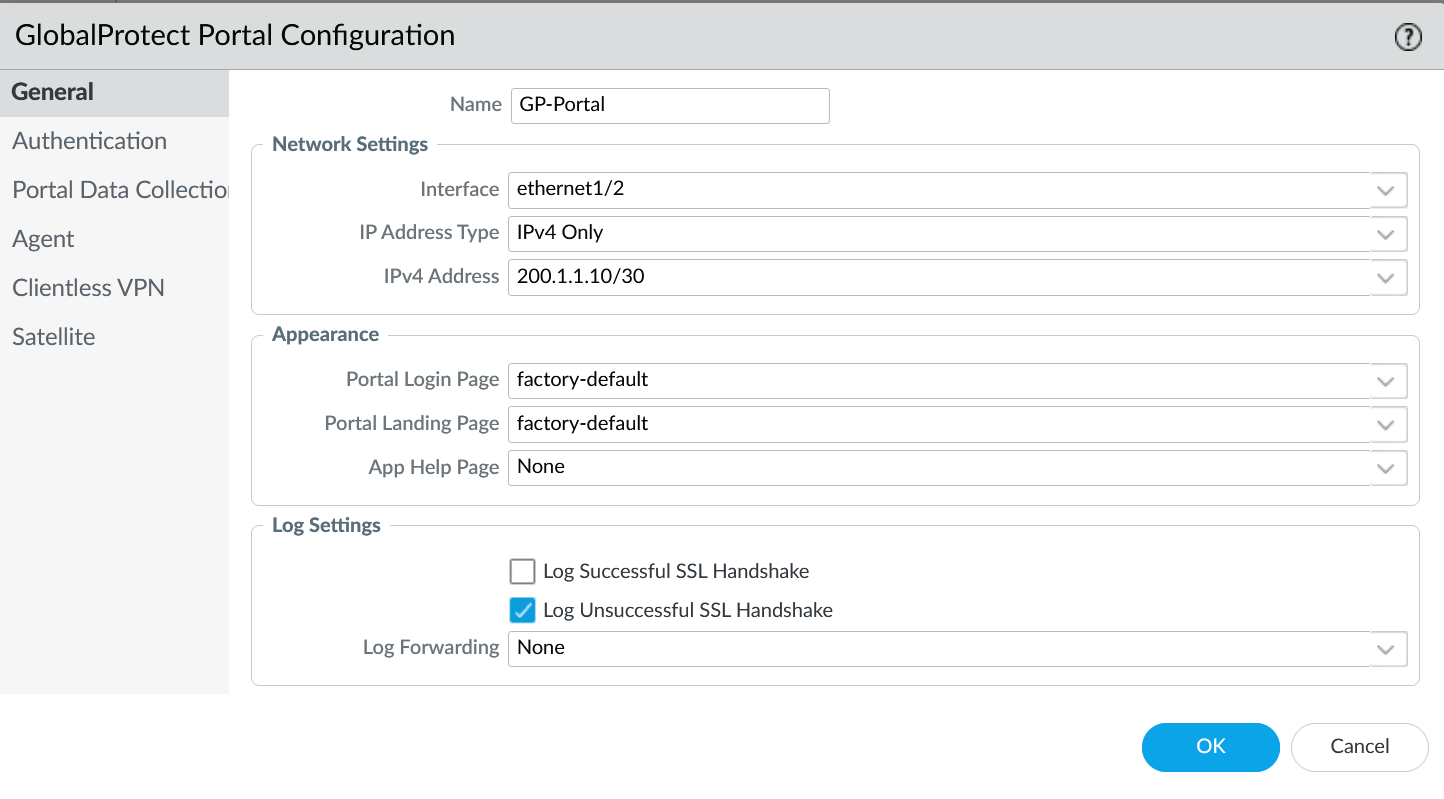
(Note depending on what PA version output may look different)

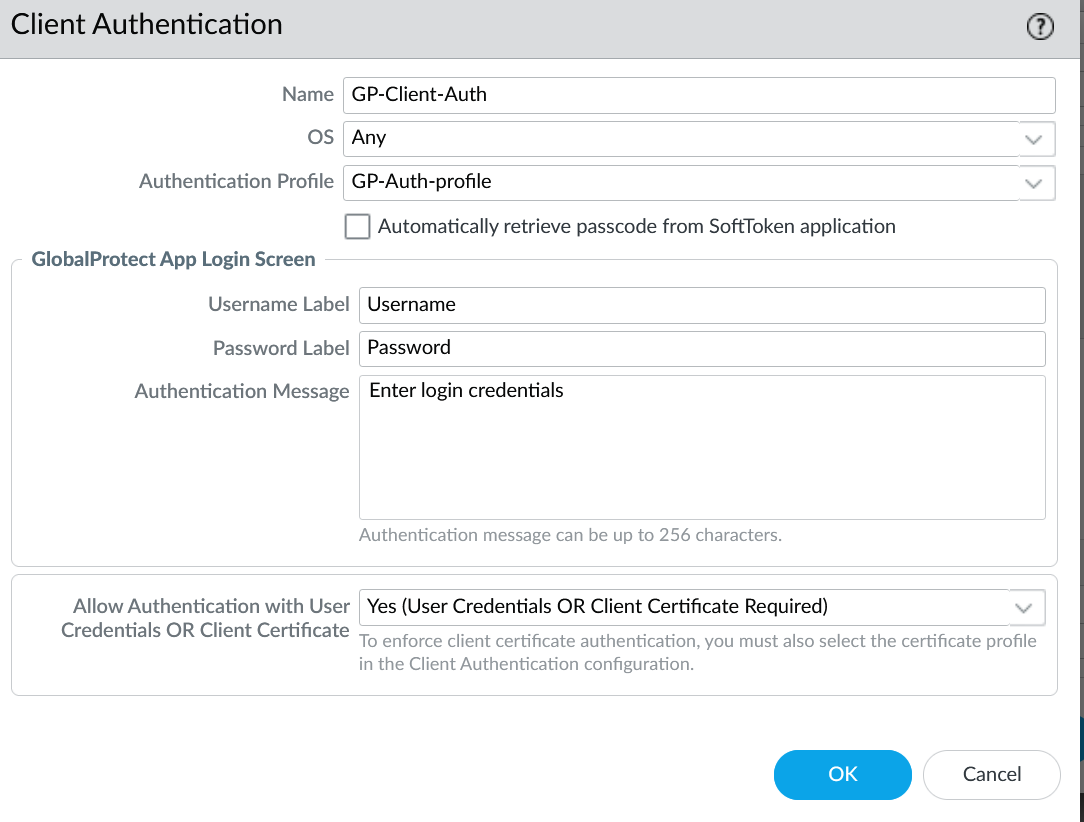
## Step 4: Portal configs

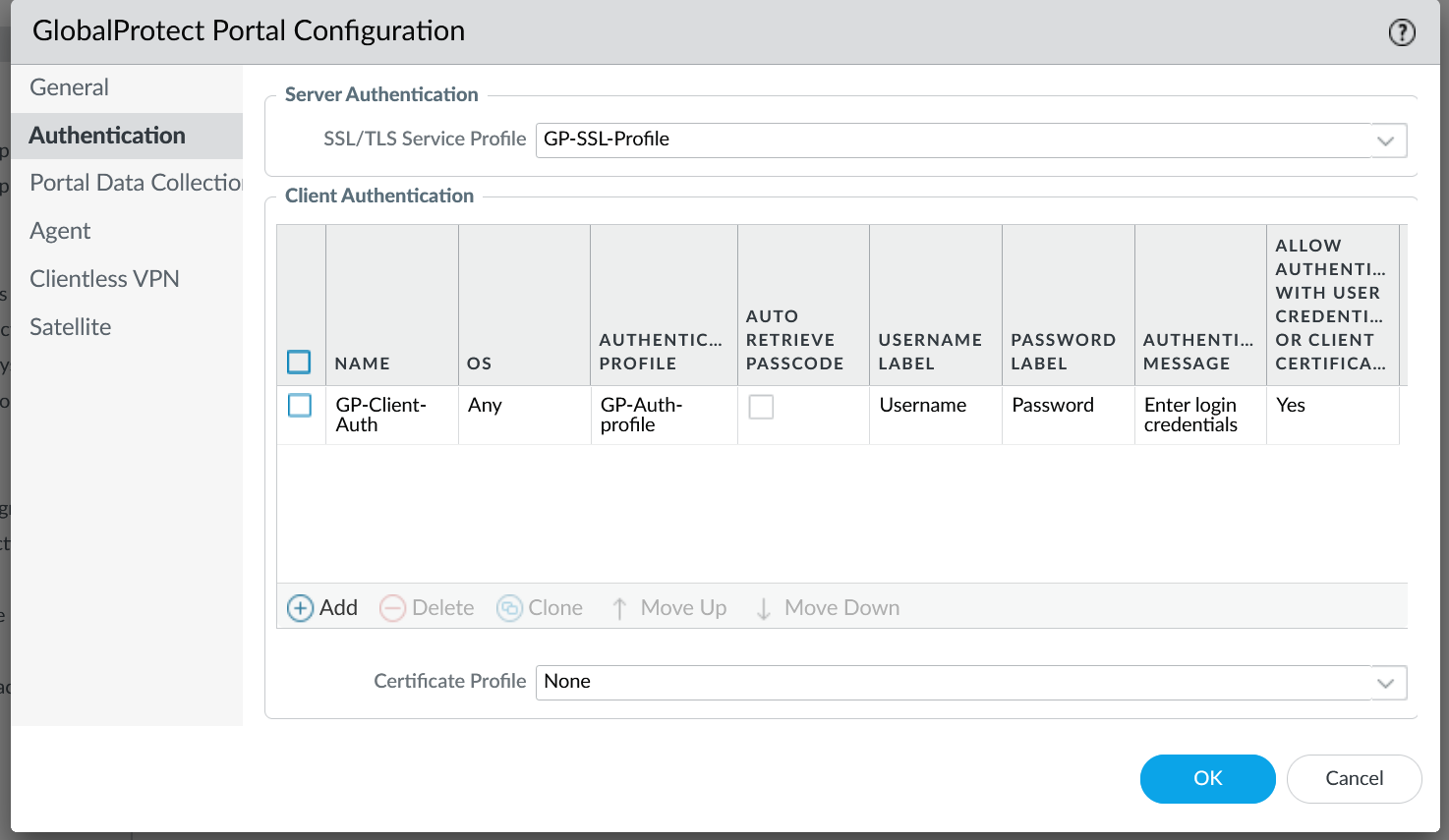
Configure portal

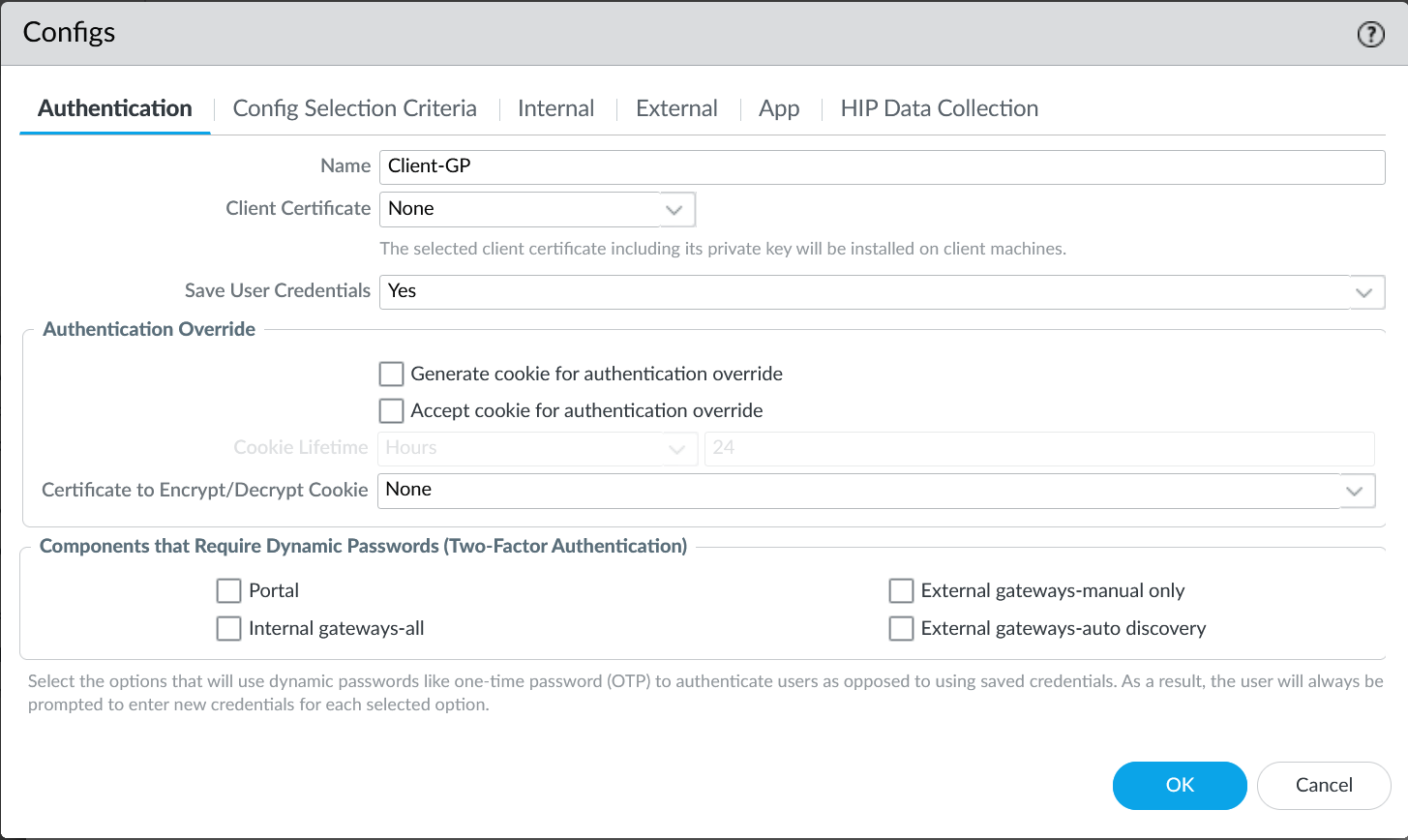
Navigate to Network -> GlobalProtect -> Portals (Then click Add)

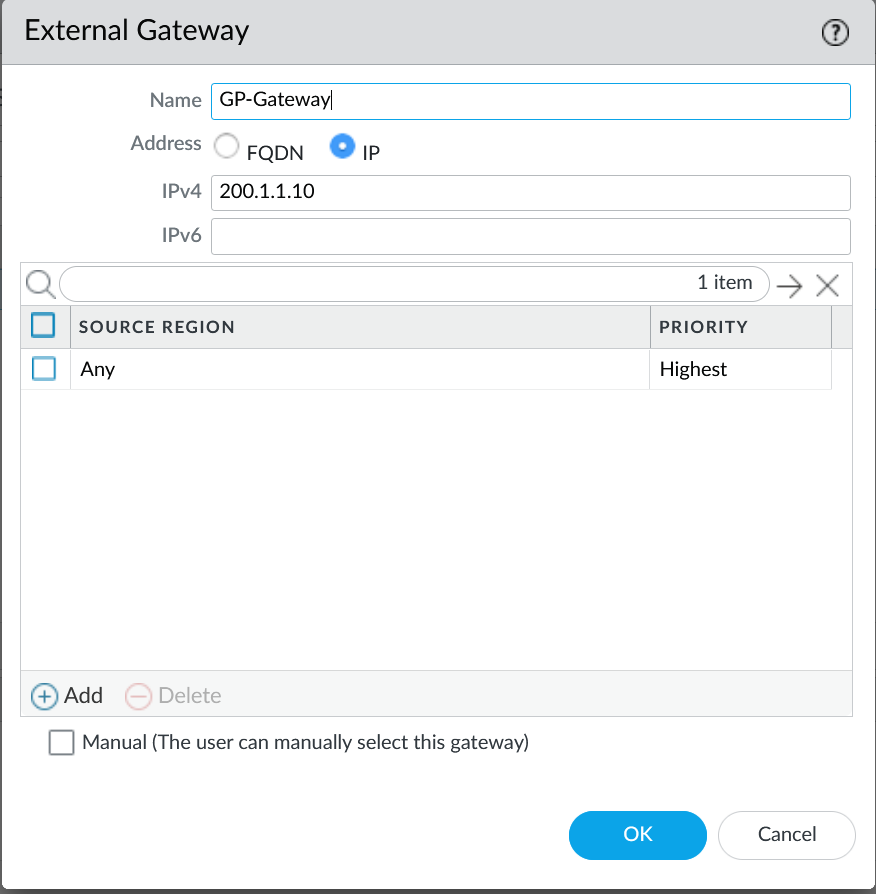
|  |  |
| --- | --- |
| General -> Name | GP-Portal |
| General -> Interface | Ethernet1/2 |
| General -> IPv4 Address | 200.1.1.10/30 |
| Authentication -> SSL/TLS | GP-SSL-Profile |
| Authentication -> **Add** | Name: GP-Client-Auth  Auth profile: GP-Auth-profile  Allow Auth: Yes  [click ok] |
| Authentication -> Agent -> **Add** | Name: Client-GP |
| Authentication -> Agent -> Client-GP -> **External -> Add** | Name: GP-Gateway  Ipv4: 200.1.1.10  Add -> Source (Any)  [click ok] |
| Authentication -> Agent -> Client-GP -> **App** | Connect Method: on Demand  [click ok]  [click ok]  to finish the GP portal configuration |

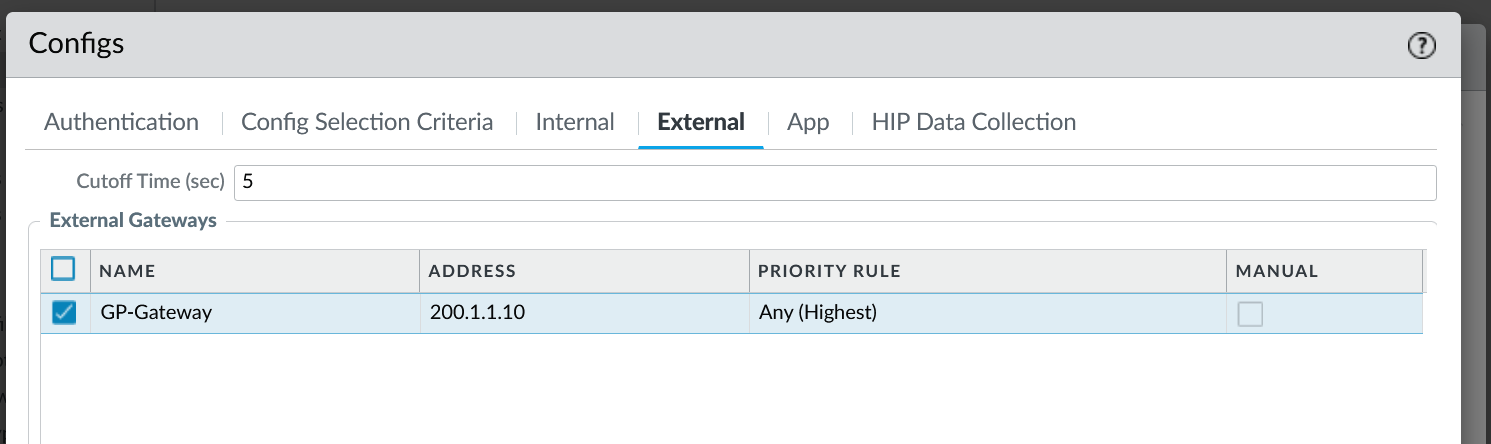


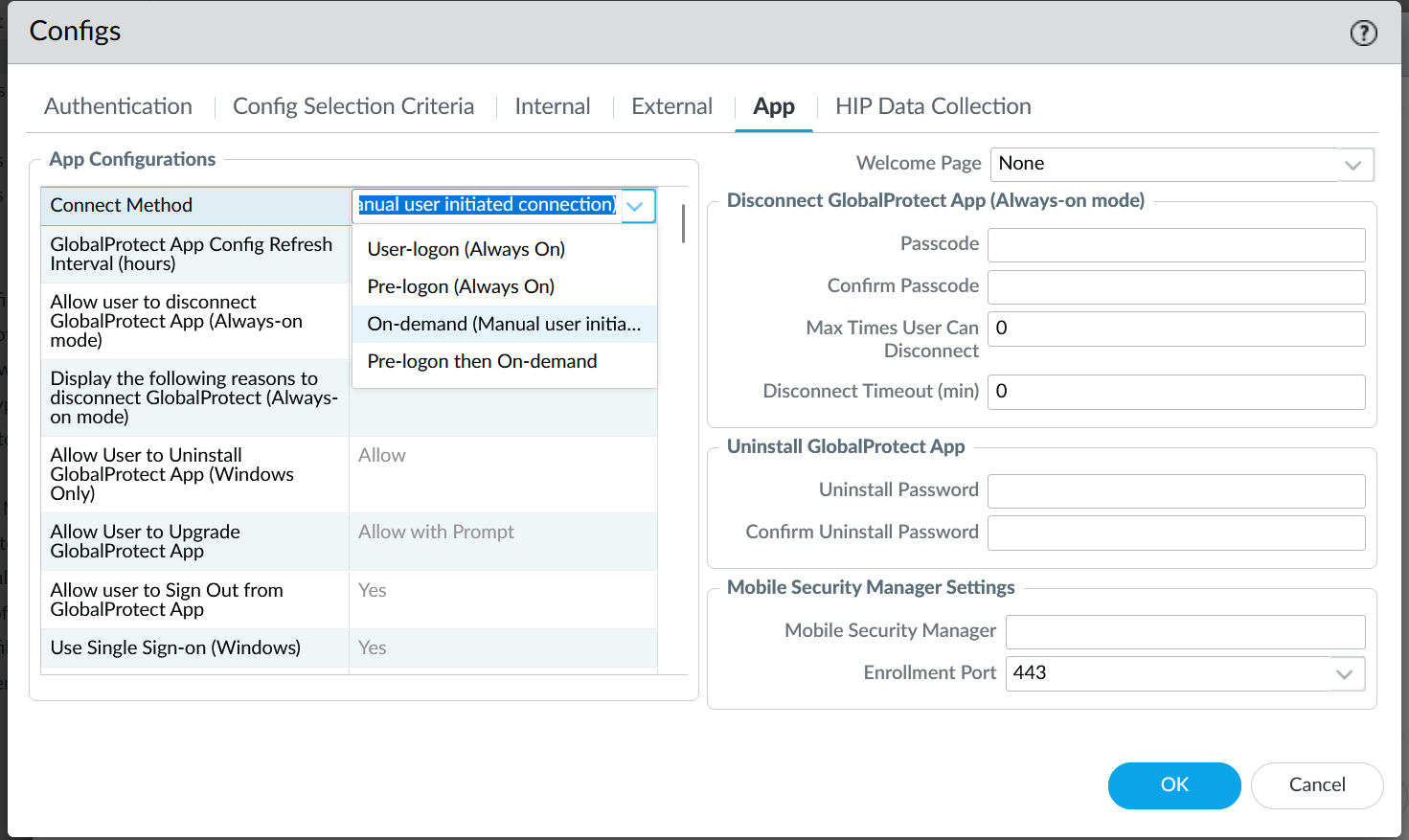








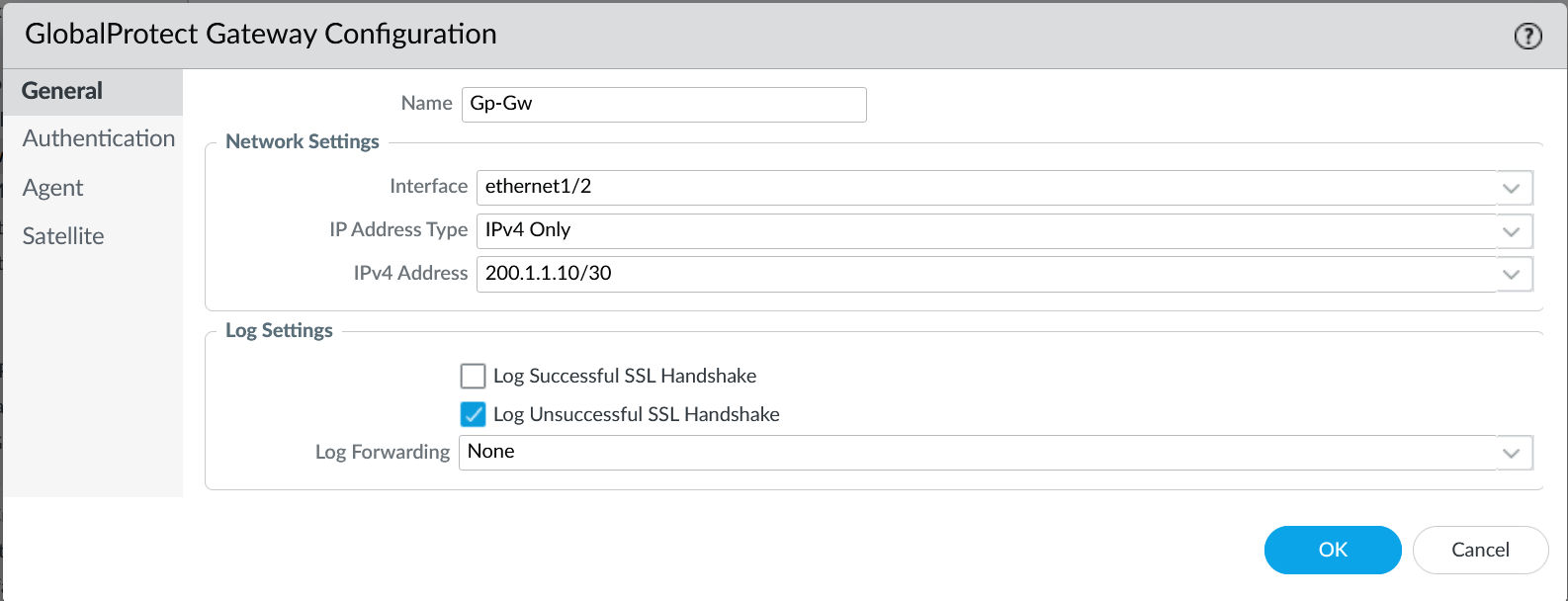


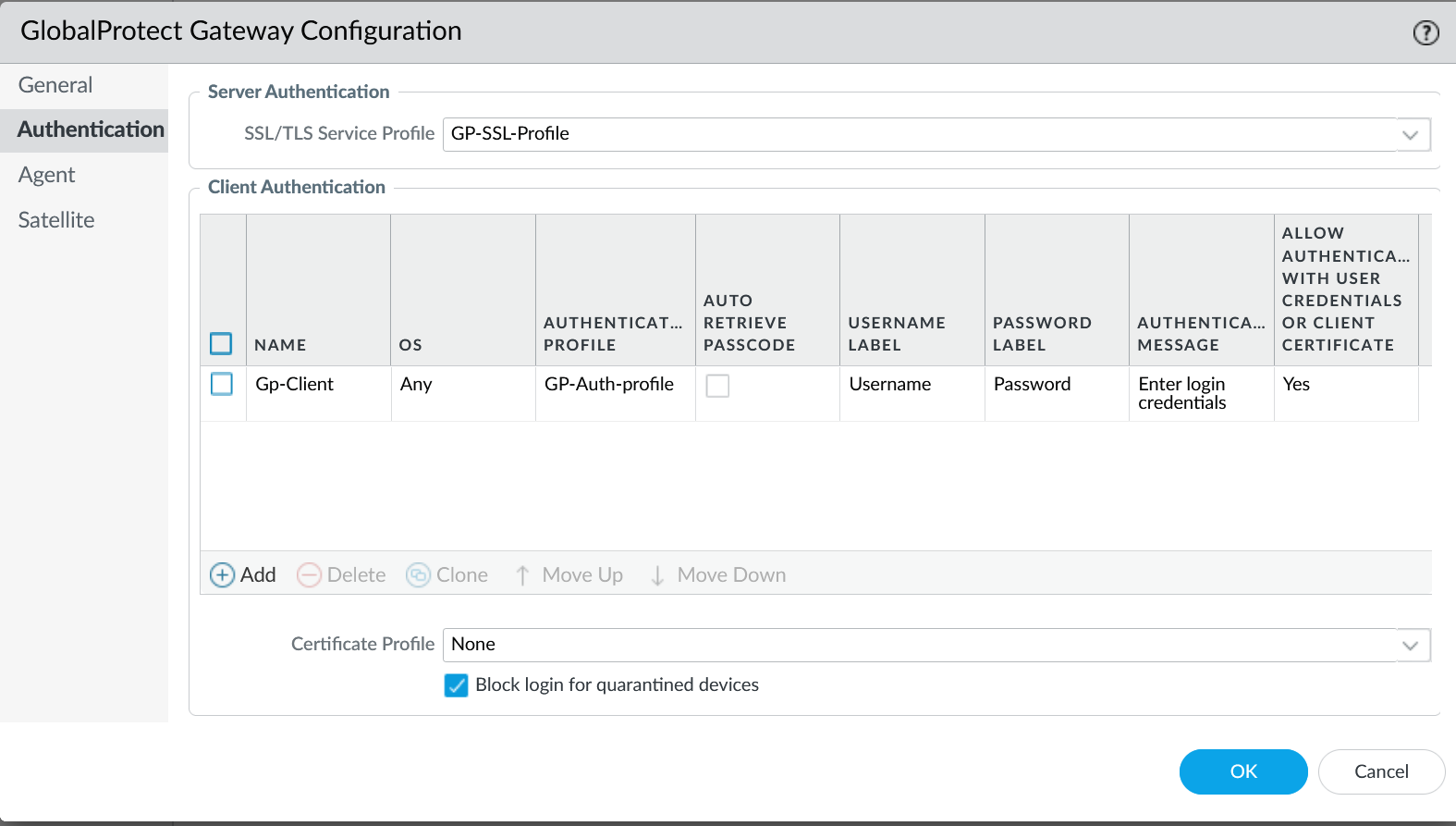


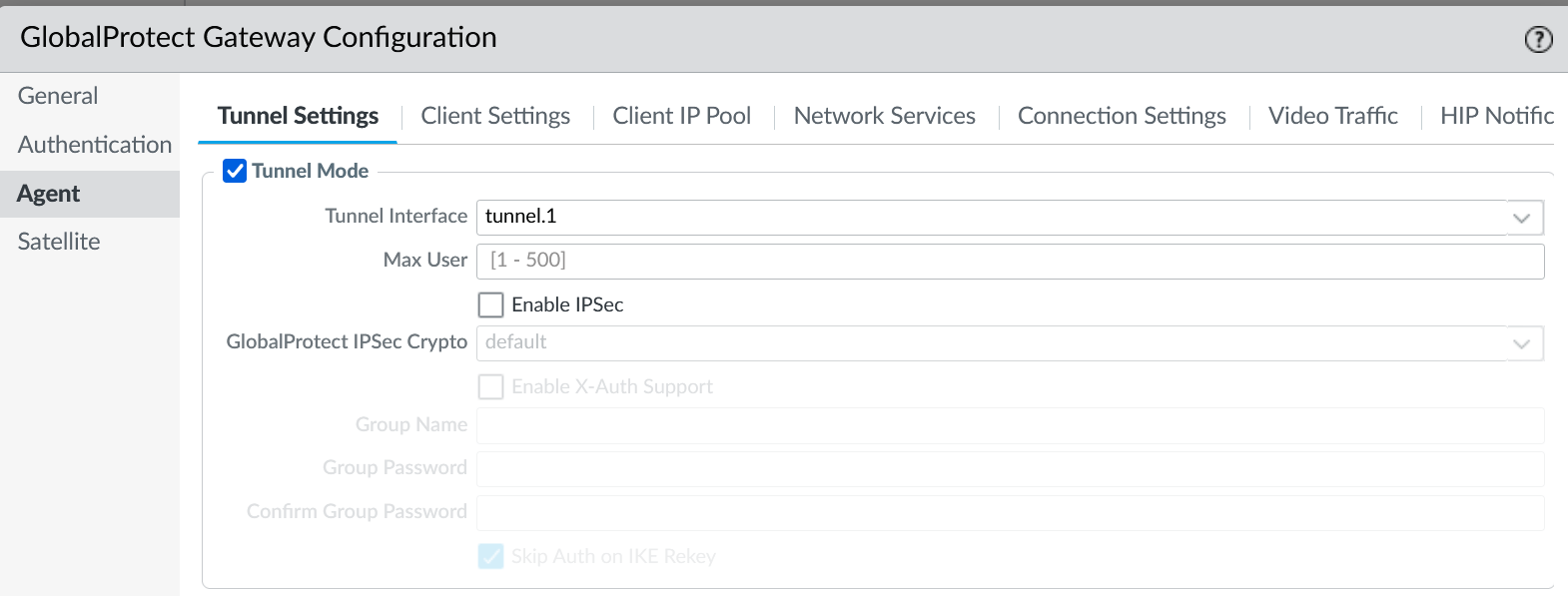
## Step 5: Gateway config

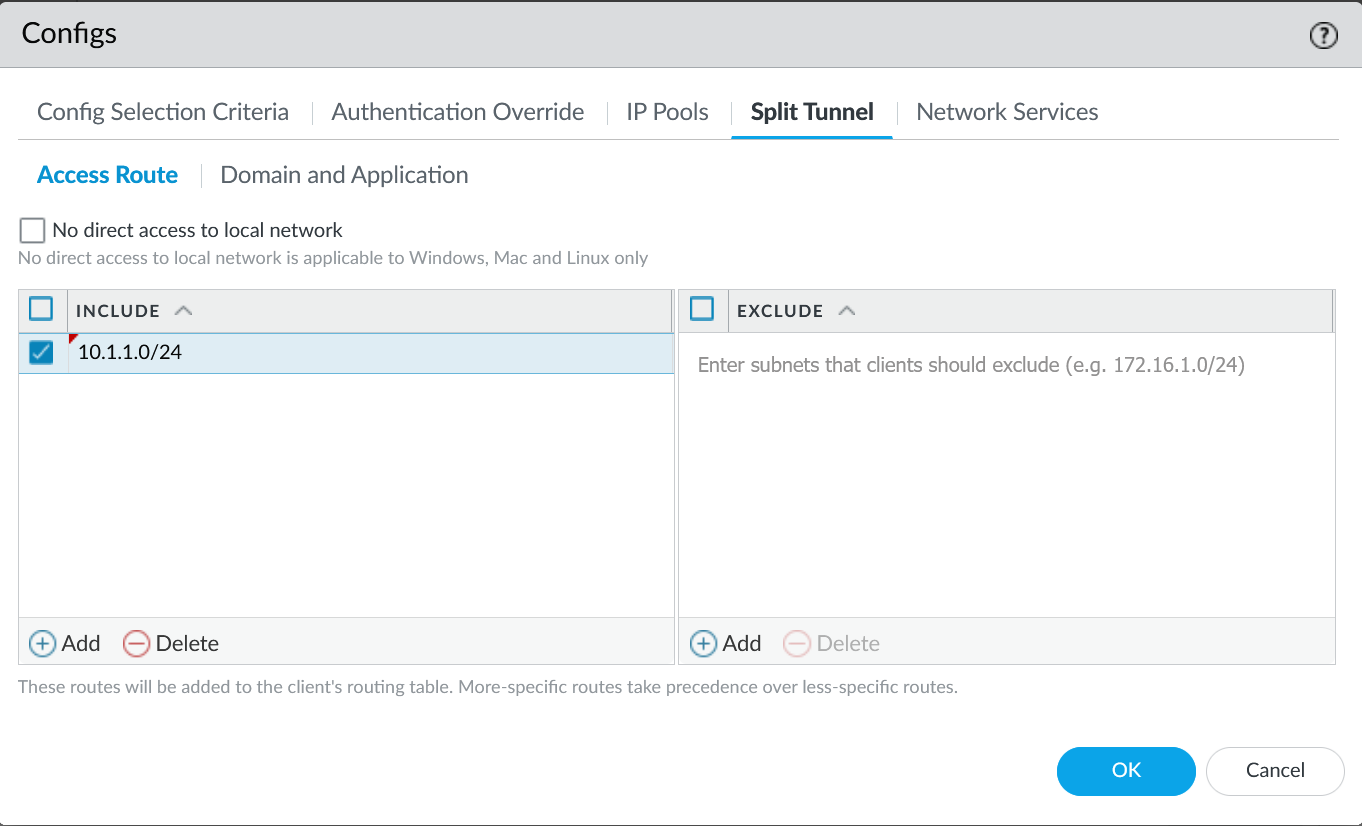
Navigate to Network -> GlobalProtect -> Gateways (Then click Add)

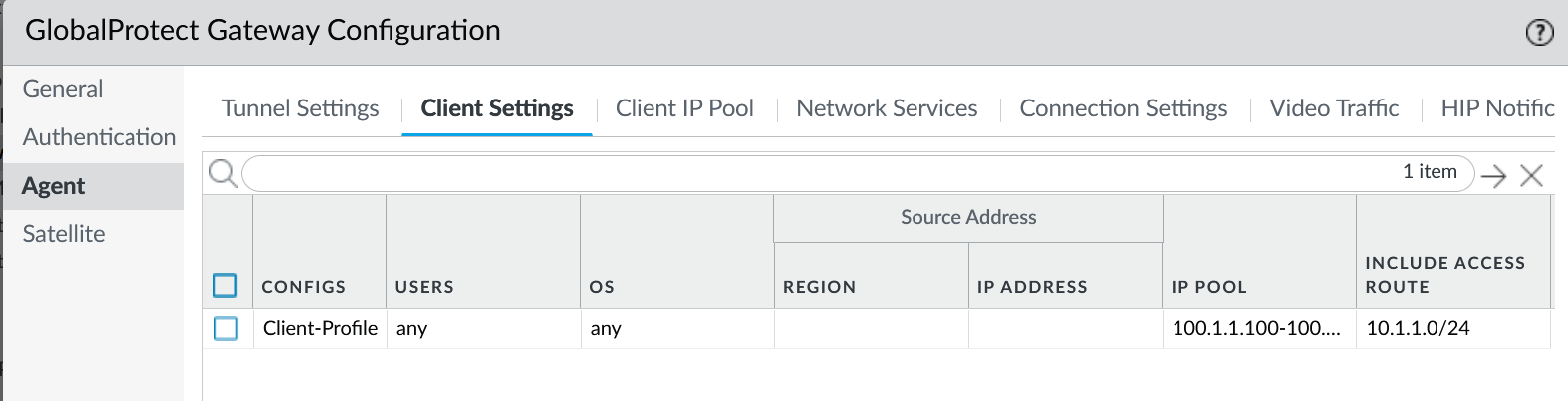
|  |  |
| --- | --- |
| General -> Name: | Name: Gp-Gw  Interface: ethernet ½  Ipv4: 200.1.1.10/30 |
| General -> Authentication: | SSL/TLS Profile: GP-SSL-Profile +**Add** |
| General -> Authentication -> Add | Name: Gp-Client Authentication  Profile : GP-Auth-profile  Allow Authentication: Yes  [ click ok] |
| General -> Agent -> Tunnel Settings | [Check] Tunnel mode  Tunnel Interface: tunnel.1  Disable IPsec |
| General -> Agent -> Client Settings | **Add**  Config Selection -> Name: Client-Profile  IP Pools -> 100.1.1.100-100.1.1.200  Split Tunnel -> 10.1.1.0/24  Click OK Click Ok  than commit |

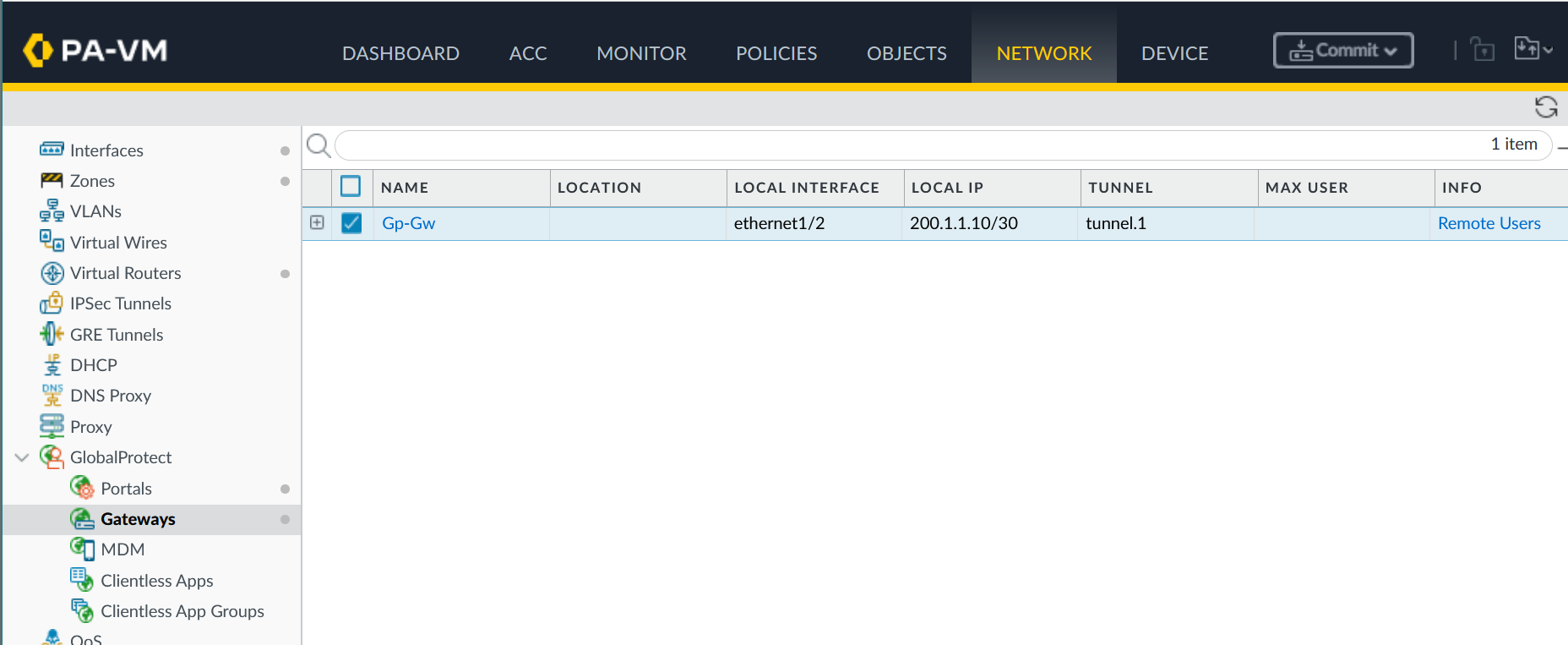












Navigate to Network -> Interfaces -> Tunnel.1 than configure tunnel

Assign a IPv4 address to the tunnel interface

|  |  |
| --- | --- |
| IPv4 | 10.1.1.10/24 |

## 

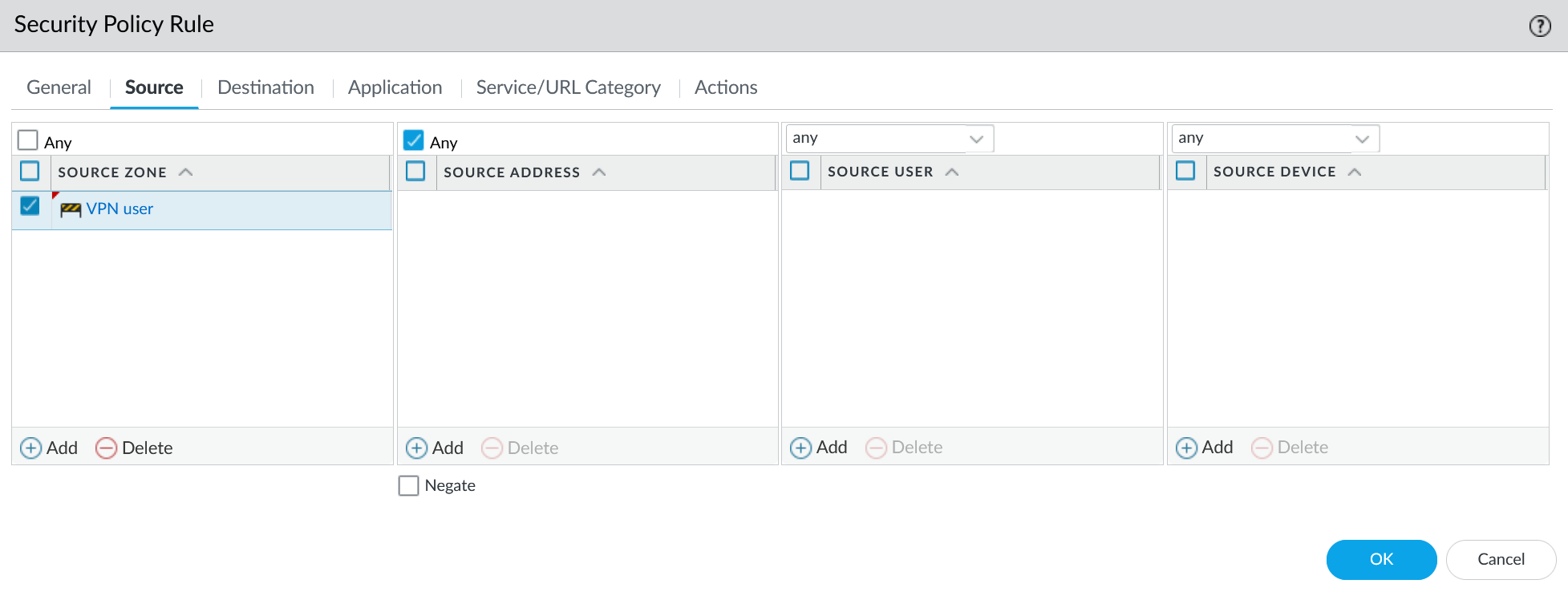
Click [ok] and commit

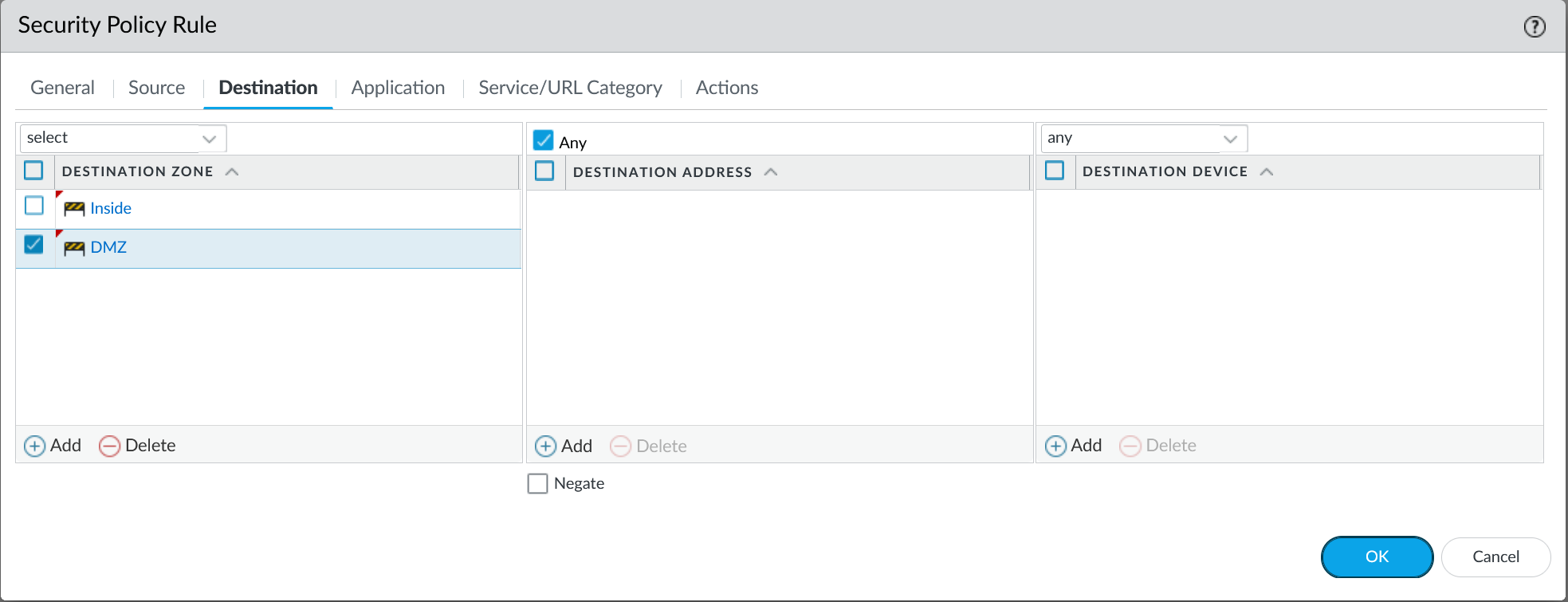
## Step 5: Define Policy

We have to define security policy to allow traffic from vpn user to internal.

Navigate to Policies -> Security (Then click Add)

|  |  |
| --- | --- |
| Security Policy Rule -> General | Name: GP-VPN |
| Security Policy Rule -> Source | SOURCE ZONE:  VPN user |
| Security Policy Rule -> Destination | DESTINATION ZONE  Inside  DMZ |
| Security Policy Rule -> Application | Any |
| Security Policy Rule -> Actions | Allow |



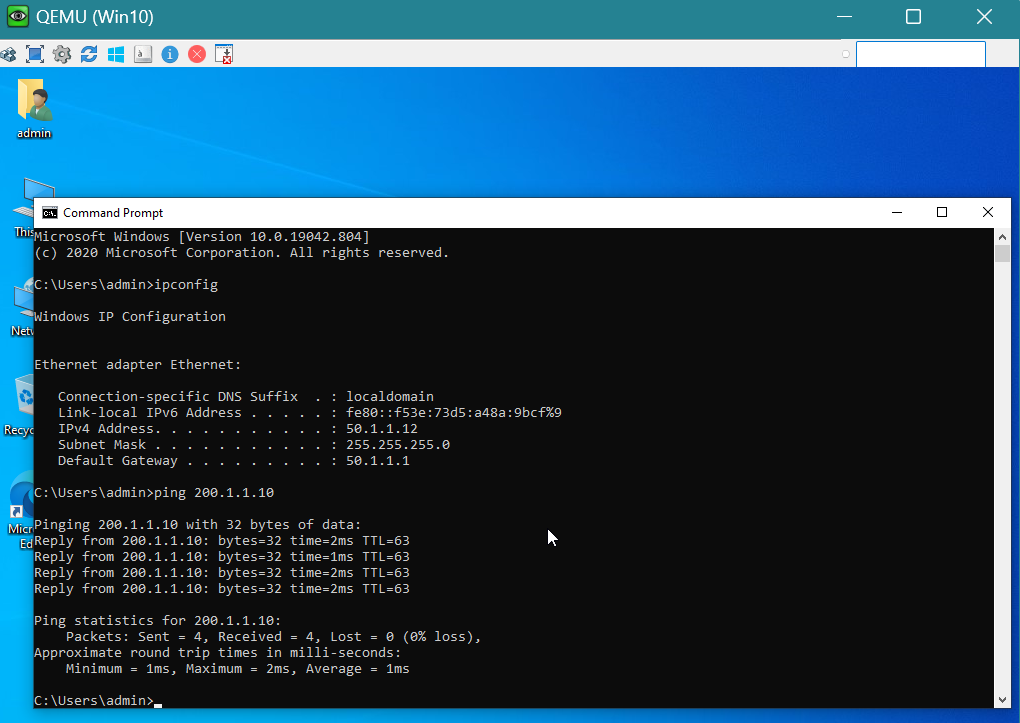


Commit the changes

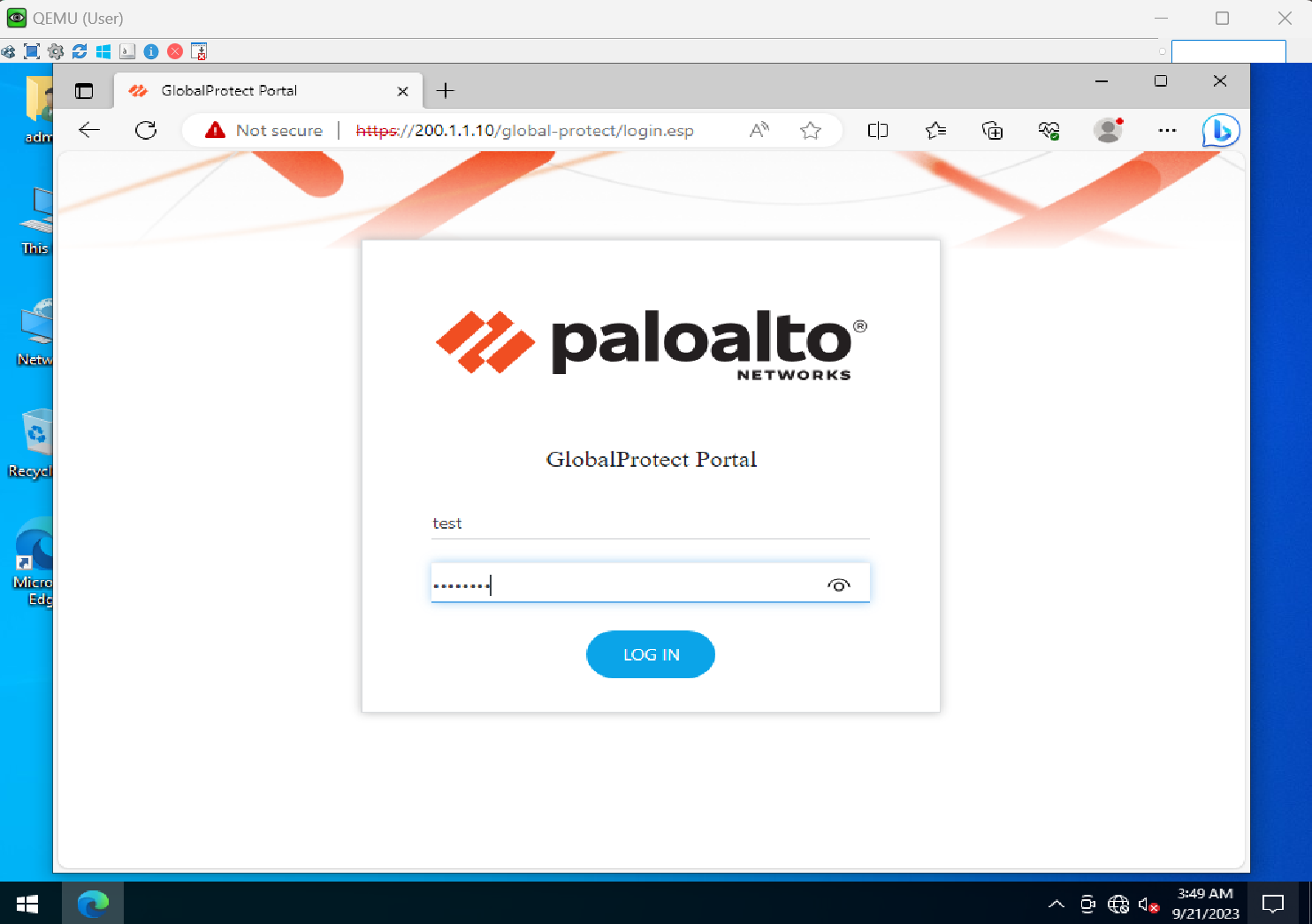
## Step 6: Verify reachability to Global protect and download client vpn

First we need to verify that we can reach our palo alto firwall. The management profile that was configured before will allow user to ping the external interface of the firewall.

From the user PC ping 10.1.1.10

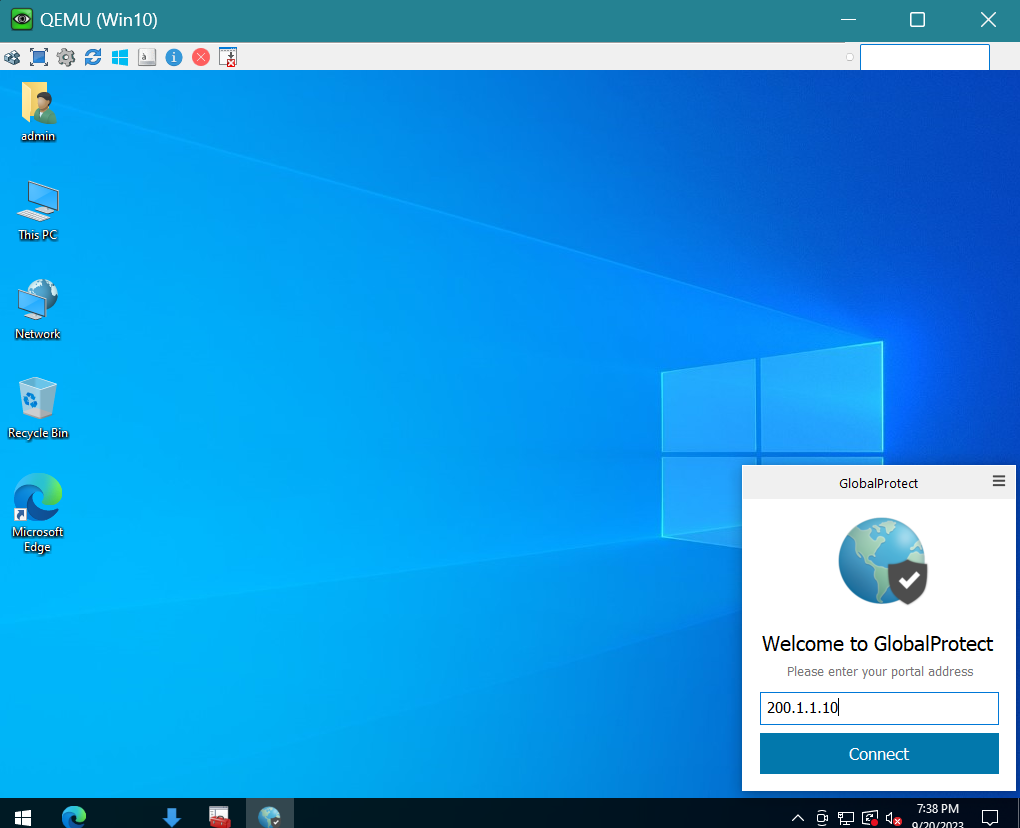


Verify that you are able to reach the Global protect login page to download the global protect client vpn.

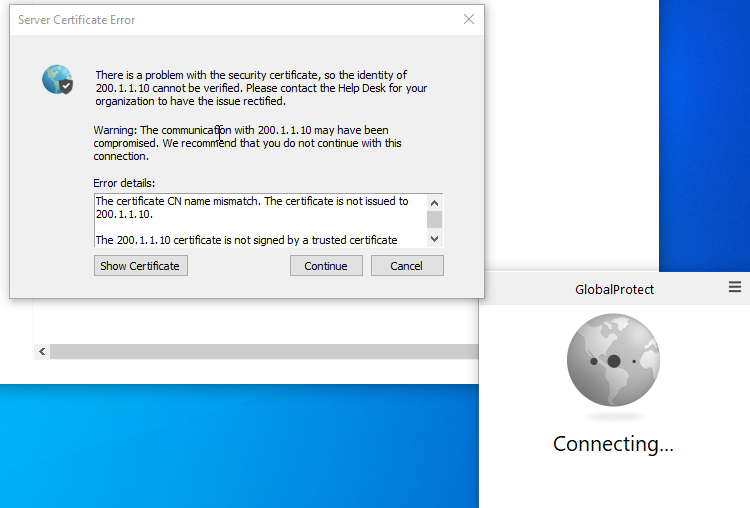


If you cannot or if palo a lot is no licensed you will have to obtain the client vpn from the internet and download it that way (not this assumes you using this in a lab. Not a good idea to use old global protect client vpn in production.)

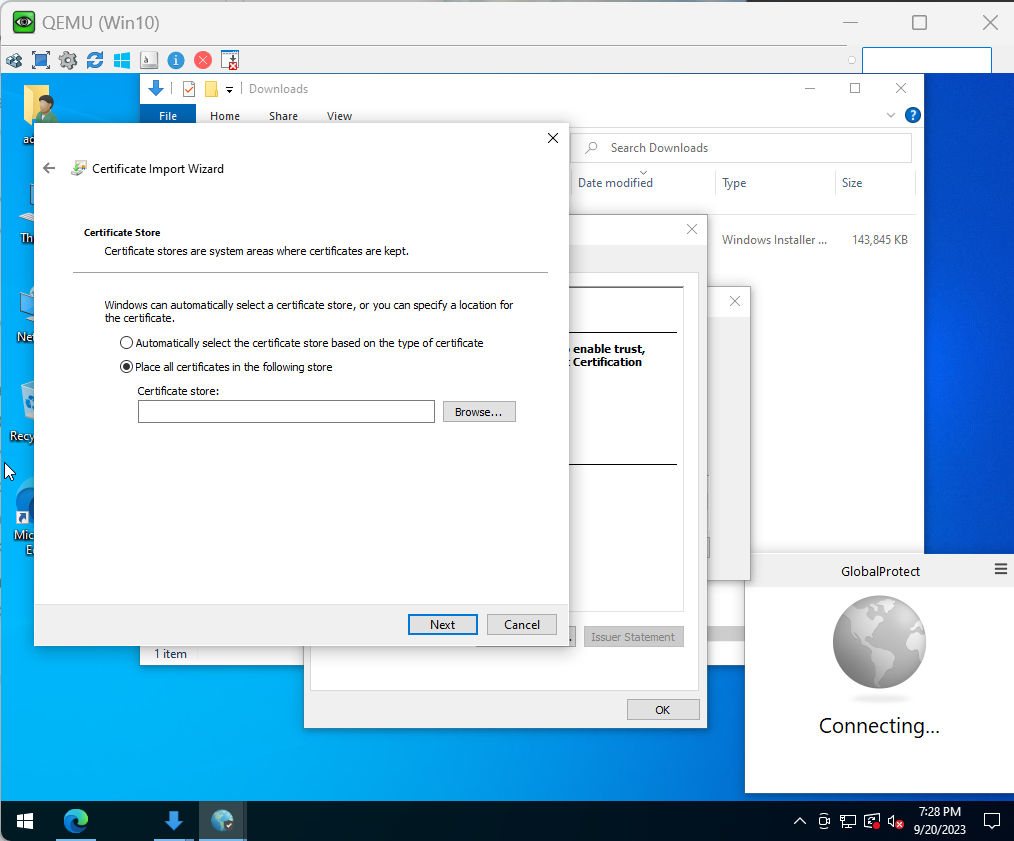
Install global protect and type in 200.1.1.10 for the Portal address.



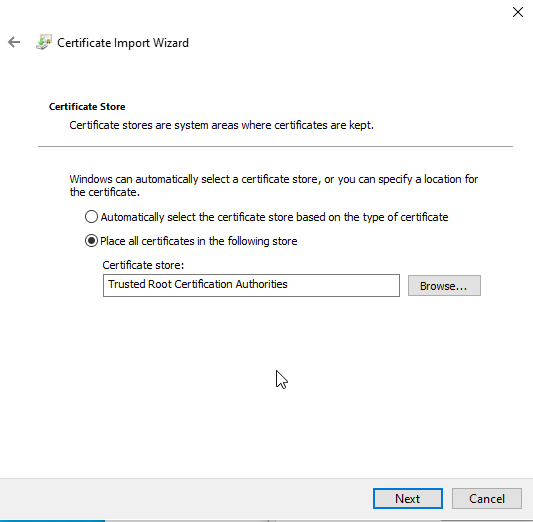
You should receive a Server Certificate error



Click on Show Certificate -> Install Certificate -> **Current user** -> Place all certification in the following store



Click on Trusted Root Certificate

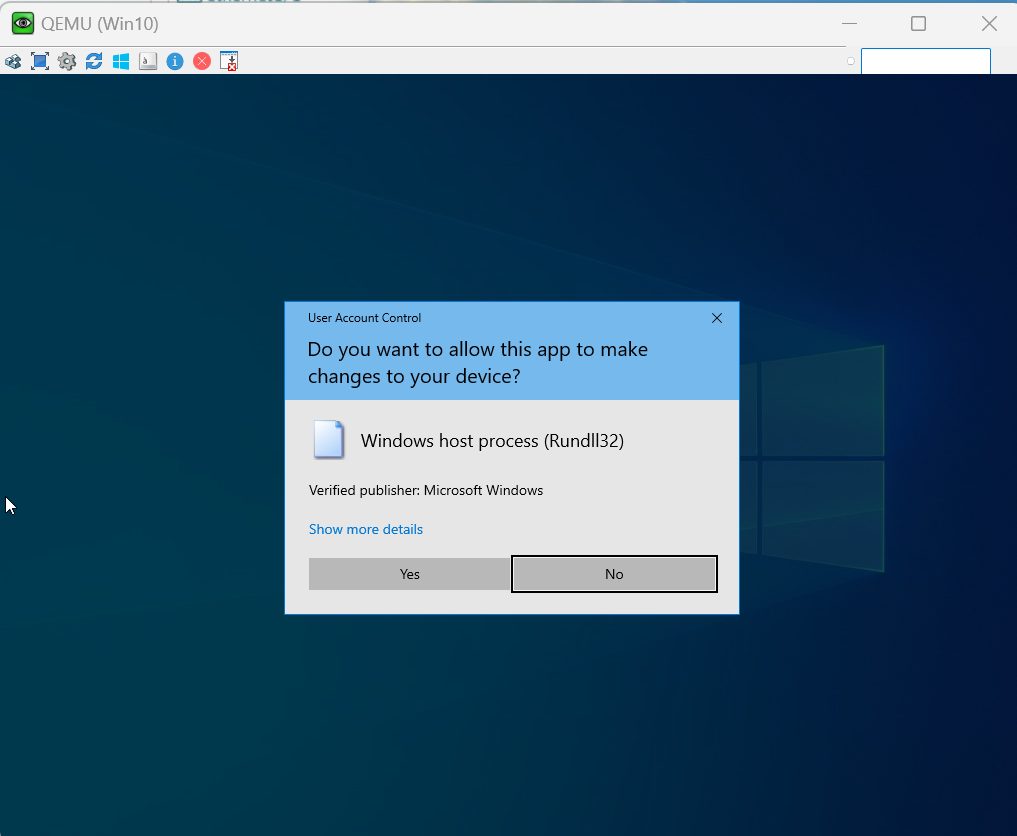


Click -> Next -> Finish -> Yes on the security warning and the import should be successful

## 

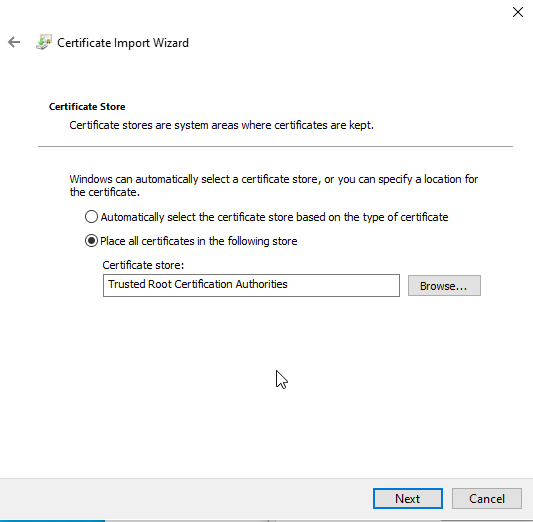
Click -> ok -> ok -> Show Certificate

Install Certificate -> **Local Machine** -> You may get a prompt click on yes



Place all certification in the following store

Click on Trusted Root Certificate



Click -> Next -> Finish -> Yes on the security warning and the import should be successful

Click -> ok -> ok -> Continue

Login in with you local user

Username: test

Password: 12345678

## Step 7: Use ansible for troubleshooting

1. You can use ansible Ad-hoc commands for troubleshooting. It’s a matter of what you are looking for and filtering for. You can use the below command as an example:

ansible all -i ./gns3hosts -m raw -a "show arp" -u user -k | grep 'CHANGED\|3e66'

![](media/7ccf556812c9c40e4defd4358e5a53fb.png)

**`-a "show arp**"`: Specifies the actual command to be run on the target hosts. In this case, the command is `show arp`, which typically displays the Address Resolution Protocol (ARP) table of the system.

**`| grep 'CHANGED\|3e66'**`: The output of the Ansible command is piped (`|`) into the `grep` command. `grep` is a command-line utility for searching plain-text data sets for lines that match a regular expression. In this case, it's used to filter and display only the lines of the output that contain the string 'CHANGED' or '3e66'.

So, in short, this command will ask for a password, then connect as the 'user' to all hosts defined in the `gns3hosts` file, run the `show arp` command on each of them, and then filter the output to display only the lines that contain 'CHANGED' or '3e66'.

Finished!