Réseau Zéro

Le Réseau de zéro

Bonjour à toutes et tous ! □□

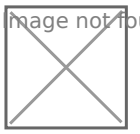
Cet article est un prologue à un autre article qui sera dédié au VPN Wireguard.

Les deux parties devaient à l'origine faire partie d'un même article.

Mais au vue de la longueur déjà conséquente de celui-ci. J'ai décidé d'en faire deux.

L'article que vous avez sous les yeux est donc une initiation au Réseau et à ses concepts de bases qui seront nécessaires à la compréhension du fonctionnement et de la configuration de Wireguard.

Je vous souhaites une bonne lecture. □□



#Réseau

> Qu'est ce qu'un réseau ?

Un réseau est un ensemble de machines connectées. Ces machines s'échangent des informations aux moyen de protocoles de communications.

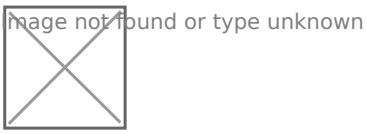
Il existe diverses façons de constituer un réseau.

Chaque noeud peut-être ce que l'on veut: un ordinateur portable, un serveur, un smartphone, un PC de bureau, une box internet et même une cafetière connectée si l'on veut ! □□

De même les "traits" qui relient nos noeuds peuvent aussi être de natures diverses. Nous pouvons connecter nos machine avec un bon vieux câble ethernet, mais aussi en Wi-Fi ou 4G/5G.

#Interface

Si l'on zoom sur un des noeuds nous allons observer ceci.



Les "rectangles colorés" constituent les interfaces de nos noeuds.

Une `interface` est une porte d'entrée/sortie d'une machine et lui permet de discuter sur le réseau.

Une même machine peut posséder plusieurs interfaces.

Pour les désigner on leur donne des noms:

- `eth0`, `eth1` : pour les branchement par câble
- `wlan0` : pour l'antenne Wi-Fi

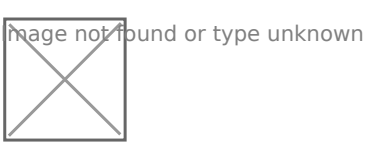
En fonction des système d'exploitation ces dénominations peuvent varier.

#Les concentrateurs

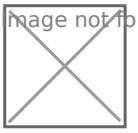
Afin de permettre de connecter plusieurs machines entre-elles sans multiplier le nombre d'interfaces nécessaires.

Nous allons brancher l'ensemble des noeuds qui désirent discuter entre eux sur machine commune appelée `concentrateur`.

Il existe plusieurs type de concentrateurs.



Ceux dits "intelligents", cela veut dire qu'ils sont capables de comprendre les messages qui leurs sont transmis et de modifier le traffic en fonction de règles, on y retrouve majoritairement les routeurs dont les boxes internet.



Et ceux qui sont "non-intelligents", il ne servent que de "passe-plats", ils ne comprennent que la destination de ce qu'ils doivent transmettre. C'est ce qu'on appelle communément un switch.

#Sous réseau

La grande force des réseaux est qu'ils peuvent discuter avec d'autres réseaux.

On appelle se procéder `l'interconnexion`.

Pour cela, un réseau 1 discute avec un réseau 2 en utilisant comme intermédiaire un réseau neutre.

Très schématiquement, ici nous avons le fonctionnement d'Internet. Un ensemble de sous-réseaux communiquants.

#Passerelle

Ce qui se passe à l'interconnexion entre réseau est aussi intéressante.

Zoomons un peu:

Nous avons une machine qui possède deux interfaces, une **eth1** et une eth0.
L'interface eth0 appartient au réseau d'interconnexion et l'interface **eth1** à un autre réseau (bleu ou rouge).

Ce genre de noeud possédant des interfaces dans plusieurs réseaux est appelé une `passerelle` ou `gateway`.

Ici nous nous limitons à l'interconnexion de deux réseaux, mais il est possible d'en interconnecter bien plus sur une même passerelle.

#Le protocole IP

Je vous ai dis que dans un réseau il y avait des messages qui transitaient sur les liaisons entre noeuds.

Mais un peu comme pour la Poste, il faut que l'on sache qui habite où.

C'est ainsi que l'on a créé le protocole internet ou `IP`.

Il existe plusieurs versions de ce protocole. Deux sont actuellement en usage:

- l'`IPv4` : très largement répandu
- l'`IPv6` : en cours d'adoption depuis 2008 (LOL □□).

L'IPv6 a été mise en place pour faire face à phénomène de raréfaction des adresses IPv4.

En effet il existe un nombre fini d'adresses, et comme chaque utilisateur doit posséder une adresse unique, comme dans la vraie vie avec la Poste pour recevoir son courrier. Plus le nombre d'utilisateurs sur internet a grandit plus le nombre d'adresses disponibles a diminué.

La principale différence entre la version 4 et la version 6 est le nombre de bits utilisé pour écrire l'adresse.

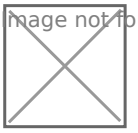
- IPv4 => 32 bits
- IPv6 => 128 bits

On va revenir sur cette notion de bit très vite.

Etant donné que l'on est pas encore prêt à utiliser l'IPv6 en 2021, je vais majoritairement vous parler de ce que je connais et utilise, l'IPv4.

#Adressage IPv4

Une adresse IPv4 est composée de 4 octets séparés par des points.



On appelle cette notation, la notation décimale pointée d'une adresse IPv4.

Cette notation avec des points n'est valable que pour des humains, la machine elle le comprend comme des nombres et plus précisément comme des nombres binaires.

Les machines ne manipulent pas de chaines de caractères car leur traitement est lent et couteux en ressources.

Pour ceux qui aurait des lacunes avec cette notation, je vous propose cette excellente ressource.

Un octet est un paquet de 8 bits, et un bit est la brique élémentaire de l'informatique permettant de représenter absolument tout lorsque l'on manipule des données en memoire.

La notation décimale existe afin d'être plus facilement manipulée par des humains.

Si l'on devais écrire la même adresse du point de vue de la machine on aurait plutôt ceci.

```
11000000.10101000.00000001.00000001
```

Et encore ici, je vous conserve les séparateurs, la machine ne les vois même pas et donc si on se ramène à une valeur décimale notre adresse vaut `3232235777`.

Une autre adresse comme `10.0.0.2` vaut `167772162` du point de vue de la machine.

Ce qui n'est pas très pratique à se rappeler pour nous pauvres êtres faits de chair et de sang. □□

#Masque de réseau

Je vous ai dit qu'un réseau est un ensemble de machines qui discutent entre elles.

> Mais comment expliquer à un ordinateur ce qu'est un réseau et si oui ou non telle ou telle machine appartient à ce réseau ?

C'est tout le but d'un masque réseau.

Cela fonctionne un peu comme un indicatif de numéro national.

Image not found or type unknown

Un même numéro de téléphone correspond à différents usagers en fonction de leur indicatif national.

En réseau on va réaliser la même chose.

Image not found or type unknown

Ici en rouge nous allons avoir "l'indicatif du réseau" et en bleu "le numéro de l'usager".

Notre "usager" étant la machine connectée au réseau.

On remarque que le dernier numéro peut-être identique d'une machine à l'autre mais ne pas correspondre à la même machine.

Toutes les machines possédant un même indicatif peuvent discuter entre elles.

En réseau, on va définir cet indicatif d'une manière un peu spéciale.

Comme vu précédemment sur la partie sur l'adressage la machine ne manipule que des nombres.

> Comment faire pour définir que les machines `192.168.1.2` et `192.168.1.3` sont dans le même réseau mais pas la machine `192.168.9.4` ?

Les concepteurs de l'IPv4 ont créé un concept astucieux: le masque de réseau.

L'idée est de profiter des propriétés des nombres binaires et une en particulier appelée le `ET binaire` .

```
# Table de vérité
1 & 1 = 1
1 & 0 = 0
0 & 1 = 0
0 & 0 = 0
```

On va effectuer cette opération sur chaque bit d'une adresse.

Image not found or type unknown

La partie du haut est ce que l'on désire masquer et la partie du bas le masque.

Tout ce qui est à l'aplomb d'un `1` est conservé, sinon il sera transformé en `0`.

Comme pour les adresses on peut définir nos masques avec une notation décimale pointée.

Entre les adresses `192.168.1.2` et `192.168.1.3`, nous voyons que la racine commune est `192.168.1` qui n'est pas partagée par `192.168.9.4`.

Cela semble une bonne méthode pour départager nos adresses.

Nous voulons conserver les 3 premiers octets de nos adresses et supprimer le quatrième.

Nous avons vu que les `1` des masques conservaient les données et les `0` les supprimaient.

Comme nous voulons conserver des octets complets, nous cherchons un masque constitué uniquement de `1`.

Image not found or type unknown

A l'inverse le quatrième octet ne doit pas être conservé, on cherche un masque uniquement constitué de `0`.

Image not found or type unknown

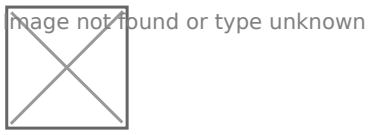
Prenons maintenant la notation décimale de ces 2 masques:

Image not found or type unknown

Et maintenant écrivons le masque complet:

```
masque = 255.255.255.0
```

Si l'on applique celui ci à nos adresses:

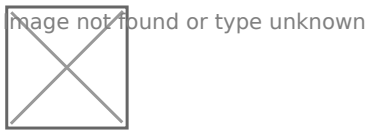


Nous n'avons plus qu'à comparer les adresses masquée entre elles.

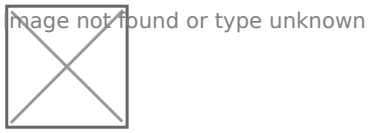
Si elles sont identiques, elles appartiennent au même réseau et discutent entre elles. Sinon ce n'est pas le cas.

En faisant nos calculs de masques, on observe bien que les deux premières adresses sont dans le même réseau mais ce n'est pas le cas de la dernière.

Pour définir si une adresse à appartient ou non à un réseau on va comparer si l'adresse testée masquée est égale à l'adresse du réseau masquée.



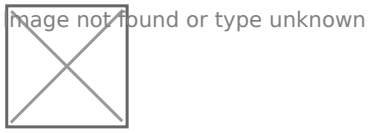
On peut réaliser la même opération sur des adresses différentes, des masques différents et des réseaux différents.



#Notation CIDR

Les informaticiens sont fainéants de nature et écrire des masques toute la journée n'est pas très enrichissant.

Cette situation fastidieuse a poussé les concepteurs de l'IPv4 à inventer une notation basée sur le nombre de `1` consécutif d'un masque.



Et donc pour notre réseau

```
adresse : 192.168.1.0
masque : 255.255.255.0
```

On écrira plutôt:

```
192.168.1.0/24
```

Remarque

Ce réseau peut s'écrire indifféremment `192.168.1.1/24` ou `192.168.1.8/24` ou `192.168.1.0/24`, cela n'a pas d'importance. Seule compte l'adresse masquée.

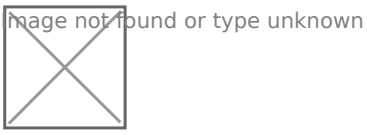
#Plage adressable

Maintenant que nous savons comment définir si une adresse appartient ou non à un réseau, il est temps de parler de plage d'adresses.

De la même manière que dans une rue avec une longeur défini. Le nombre de maisons et donc de numéros dans la rue est lui aussi fini.

Un octet est un paquet de 8 bits. Ce qui signifie que la valeur la plus basse qu'il peut prendre est `0` et la plus haute `255`.

Si on reprend notre réseau `192.168.1.0/24` on obtient:

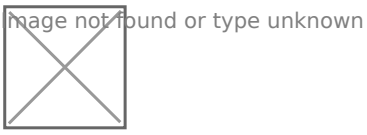


Deux adresses sont un peu spéciales:

- `192.168.1.0` : l'adresse du réseau, ne peut pas être utilisée par une machine
- `192.168.1.255` : l'adresse de broadcast, on y revient dans la suite, ne peut pas non plus être utilisée par une machine.

Tout le reste est appelée, `plage adressage` du réseau `192.168.1.0/24`.

Bien-sûr on peut réaliser le même décompte des adresses possibles dans un réseau différent.



#Adresse de broadcast

L'adresse de broadcast est l'une des deux adresses qui existent sur les réseaux mais qu'il n'est pas possible d'obtenir comme IP d'une machine.

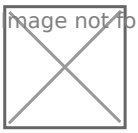
Cette adresse est toujours la dernière de la plage du réseau.

On calcule cette adresse via une formule qui utilise deux opérateurs.

```
#La négation
~ 1 = 0
~ 0 = 1

#Le OU binaire
1 | 1 = 1
1 | 0 = 1
0 | 1 = 1
0 | 0 = 0
```

Par exemple:



Cette adresse est utilisée pour transmettre des informations à l'ensemble des machines connectées sur le réseau.

Lorsqu'un switch reçoit un message qui a pour destinataire l'adresse de broadcast, il retransmet le message sur tous les ports.

C'est pour cela qu'elle n'est pas utilisé pour adresser une machine en particulier car elle désigne toutes les machines du réseau par définition.

#Les passerelles de sous-réseaux

Maintenant que nous sommes des champions du réseau il est temps de les faire parler entre eux.

Si vous vous rappelez bien on a parlé des passerelles réseau plus haut.

Et bien c'est leur heure de gloire.

Nos passerelles possèdent 2 adresses IP dans deux sous-réseaux différents.

Une adresse dans le réseau bleu et une adresse dans le réseau vert.

C'est ainsi que fonctionne les boxes Internet, elles ont à la fois une IP dite publique qui leur permet de se connecter à Internet. Mais aussi une IP privée généralement `192.168.1.1` qui permet de communiquer avec des pareils de la maison.

Cette box internet est la passerelle de votre réseau domestique.

Pour l'instant les sous-réseaux ne communiquent pas entre-eux il manque un mécanisme permettant de transmettre les informations, du réseau bleu vers le réseau vert et inversement.

#Routage

Ce mécanisme se nomme le routage.

Il consiste en une série de règles qui définissent comment les paquets entrants et sortants d'une passerelles doivent se comporter.

Ainsi que les échanges qui sont réalisés entre les interfaces de la passerelle.

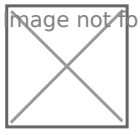
#Ports logiques

Sur une machine il tourne généralement plusieurs applications, le souci est de savoir à qui l'on doit remettre le message.

Pour reprendre l'exemple de la Poste. Chaque immeuble possède un numéro unique mais plusieurs appartements.

Ici le numéro d'immeuble serait l'IP de la machine et le numéro de porte, le port logique.

On défini ce couple de numéros par une notation standardisée, qui sépare l'adresse de la machine du numéro de port par un `:`.



Les numéros de ports sont en parti standardisés et correspondent généralement des applications bien définies.

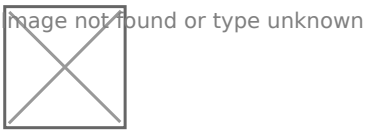
Par exemple un serveur HTTP aura comme de port `80` et un serveur sécurisé HTTP, le fameux HTTPS, le `443`.

Il existe une liste qui répertorie les numéros de port les plus connus.

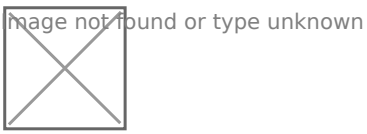
#Routage de ports

Maintenant que nous sommes capable de distinguer des messages ayant pour destination plusieurs applications situées sur une même machine.

Nous allons pouvoir router nos messages en fonction de nos besoins.



Par exemple rediriger tout le flux HTTP/S vers une machine A et les mails vers un machine B.



On a un flux d'entré de l'interface `eth0` qui sont composé de messages tous adressés à l'IP `10.0.0.2` mais sur des ports différents.

Nous routons les messages pour qu'ils soient transmis vers la bonne destination.

- les port `80` et `443` vers la machine `192.168.1.2`
- le port `25` vers la machine `192.168.1.3`

Le port `22` n'étant routé nulle part, les messages sont supprimés et ne seront perdu.

A l'inverse tous les messages venant du switch sont librement renvoyés vers l'interface `eth0`.

#Réseau étendu et local

La distinction entre réseau local (**L**ocal **A**rea **N**etwork) et le réseau étendu (**W**ide **A**rea **N**etwork) réside principalement dans sa taille en nombre de machines et en surface couverte.

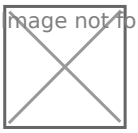
Le premier va concerné des petits réseaux domestiques ou d'entreprise. Un petit milliers de machines dans le cas général.

Tandis que le second s'intéresse aux réseau à l'échelle d'une région, d'un pays, d'un continent ou même de la planète entière.

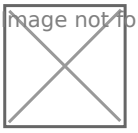
Le WAN le plus célèbre s'appelle Internet.

Comme nos machines sur notre réseau `192.168.1.1/24` notre réseau va lui aussi posséder une adresse dans WAN Internet.

Cette adresse est appellée `IP publique`, elle permet à l'ensemble des machines qui sont connectées à Internet de vous contacter et vous de pouvoir explorer Internet !



Si l'on schématise extrêmement grossièrement le fonctionnent d'Internet, on peut créer un schéma comme celui-ci:



Nous avons deux passerelle qui ont chacune deux interfaces: une bleu qui le rélie au **LAN** et une rouge qui le relie au **WAN** (Internet).

Le réseau de gauche possède sur Internet l'adresse IP publique `101.42.77.12` et le réseau droite `43.12.65.9`.

Lorsque le réseau de gauche veut discuter avec des machines dans le réseau de droite, il adresse ses message sur l'IP `43.12.65.9`.

Ensuite, en fonction des règles de routage définie sur la passerelle de droite, les messages arrivent à destination.

Si l'on veut faire l'inverse, il suffit d'adresser ses messages non plus à `43.12.65.9` mais à `101.42.77.12`.

#DNS

#Nom de domaine

Avant de pouvoir parler de DNS nous allons devoir expliquer la notion de nom de domaine.

Je vous ai expliqué que la notation décimale des adresses IP était faite pour les humains, pour qu'il puisse se rappeller plus facilement que d'utiliser la notation binaire ou même entière.

Mais si vous demandez tata Jacqueline d'aller sur Google et qu'il faut qu'elle se rappelle de l'IP publique du serveur Google par exemple : `216.58.208.195`. Il risque d'y avoir peu de personne qui retrouveront Google sur Internet.

C'est pour cette raison qu'a été créé le concept de `nom de domaine`.

Son principe de fonctionnement est associer une chaîne de caractères à une adresse IP.

Par exemple:

```
google.fr -> 216.58.208.195
facebook.com -> 157.240.14.35
```

On peut même avoir plusieurs IPs associées à un même nom de domaine.

```
twitter.com -> 104.244.42.129 et 104.244.42.193
```

> “ Le problème c'est où stocké ces associations pour les mettre à disposition du reste d'Internet ?

C'est là qu'intervient le serveur de nom de domaine autrement appelé **D**omain **N**ame **S**erver.

#Le serveur DNS

Est une machine comme une autre, située dans un LAN qui dispose comme tous les autres réseaux d'une adresse publique.

A ceci près qu'un serveur DNS possède un service qui écoute sur le port `53`.

Ce service possède la table d'association entre les noms de domaines et les IP publiques dont on parlait tout à l'heure.

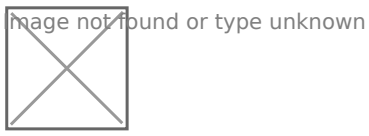
Il répond à l'ensemble des demandes venant de l'extérieur en renvoyant pour le nom de domaine demander, l'IP publique qui lui est associée.

Une autre différence est que l'on ne joint pas le serveur DNS par son nom de domaine mais par son IP publique. Sinon on se retrouverait dans le paradoxe de l'oeuf et de la poule.

Par exemple si un utlisateur situé dans un LAN, mettons une maison, souhaite se connecter à `twitter.com`.

Il va d'abord joindre le serveur DNS qu'il connait, ici le `1.1.1.1`. Qui va lui répondre que l'IP publique du LAN contenant le site `twitter.com` se trouve à l'IP `104.244.42.129`.

Il va ensuite faire une requête vers `104.244.42.129` pour atteindre le LAN de twitter.



#TCP/UDP

Ce qui nous fais poser une autre question:

> Comment fait-on pour s'assurer que les informations aillent bien au bon endroit ?

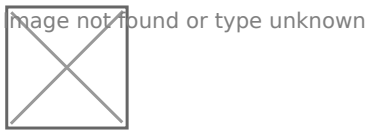
Dans cette partie on ne détaillera pas complètement le fonctionnement. Je voudrai juste que vous ayez l'intution de ce qui se passe.

Les réseaux communiquant au travers de IP utilise deux principales manières: le TCP et l'UDP.

#UDP

L'**U**ser **D**atagram **P**rotocol d'abord, c'est la manière la plus simple des deux d'acheminer un message.

Imaginez ça comme un pigeon voyageur, vous envoyez un message et c'est tout. Vous ne saurez jamais si ce message a bien été reçu par le destinataire.



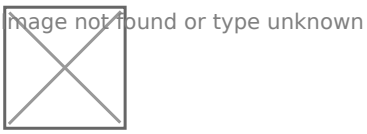
Ce protocole est utilisé lorsque les messages sont répétés et peu importants.

Son intérêt est d'être rapide et peu contraignant. On envoie les messages et adviennent que pourra.

On l'utilise par exemple dans les jeux vidéos pour connaître la position d'un joueur lors d'une partie en ligne par exemple. On peut se permettre de perdre la position du joueur à la frame 1001 puisque la frame 1002 va corriger le souci.

#TCP

Le second, le **T**ransmission **C**ontrol **P**rotocol est plus à rapprocher à une discussion par talkie-walkie.



Il y a une notion de retour d'information, en anglais on nomme ceci acknowledgement (reconnaissance).

Ce qui si signifie que pour tout les messages qui vont être envoyés. Nous allons attendre un retour du destinataire qui va reconnaître qu'il a bien reçu le message.

Et si l'on veut une autre analogie, on peut rapprocher cela d'une lettre recommandée. L'accusé de réception fait foi du bon transport du message et de sa réception.

Par contre, tout comme dans la vraie vie, une lettre avec accusé de réception est plus lourde et si elle se perd en chemin ou que le destinataire n'est pas trouvée. Elle est renvoyé à l'expéditeur.

Expéditeur qui essaiera de renvoyer le message.

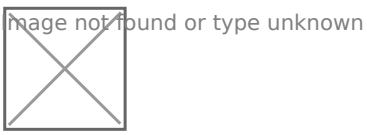
Ce protocole est utilisé dans le cas ou chaque message compte. La perte d'un message peut avoir des conséquence sur la cohérence de la communication.

Comme avec nos talkie-walkie, si la qualité de réception n'est pas bonne. Il va manquer des phrases ou des bouts de phrases. Et ainsi brouiller le sens de la conversation.

Faire répéter son interlocuteur est parfois essentiel.

#VPN

Comme disait Mufasa



Notre Royaume étant notre LAN, et le WAN le Cimetière d'éléphants rempli de hyènes qui vous veulent du mal.

Imaginons que nous voulions connecter deux LANs entre eux. Si ces LANs sont séparés géographiquement, ils devront forcément passer par le WAN pour communiquer.

Or, il se peut que les données qui vont y transiter soient sensibles et nécessite des sécurités pour éviter que des personnes mal-intentionnés ne puisse y avoir accès.

Nous devons donc trouver un moyen de faire transiter nos messages de manière sécurisée.

Une des techniques qui a été mise au point pour y arriver se nomme le **V**irtual **P**rivate **N**etwork ou autrement appelé IPsec.

L'idée est de créer une interface réseau virtuelle dans chacun des deux LANs que l'on souhaite relier.

L'interface est dite virtuelle car elle ne concerne ni une interface ethernet, ni une interface Wi-Fi.

Sur le schéma dessous elle est représentée par l'interface tun0.

La liaison entre ces deux interfaces virtuelles, va créer dans le WAN une sorte de câble réseau virtuelle.

Cette liaison est appelée un `tunnel` et de la même manière qu'un tunnel de montagne, il permet d'accéder rapidement d'un côté et de l'autre de celui-ci.

Une autre contrainte d'un tunnel est d'acheminer de manière sécurisée les messages qui y transitent.

L'idée est que si une personne qui n'est pas habilité à lire les messages tente de le faire, il ne verra que du bruit numérique.

#Conclusion

Ouf ! On est arrivé au bout. □□

Nous possédons tous les concepts de bases du réseau pour comprendre le fonctionnement du VPN Wireguard et sa configuration.

J'espère que l'article vous a plu.

Je me suis particulièrement amusé à vulgariser du mieux que j'ai pu tout ça.

La prochaine partie arrive très vite.

On s'attaquera au gros du morceau:

- Chiffrement asymétrique
- Configuration du VPN
- Cas d'utilisation
- Automatisation de la gestion des utilisateurs

Merci de m'avoir lu et à la prochaine! □□

CommentsPowered by GitLab & Vssue

Revision #1
Created 12 October 2021 19:21:06 by garfi
Updated 12 October 2021 19:21:27 by garfi