HOSTING

miNET

L'hébergement cloud à MiNET HOSTING

Seaweedbrain
14.06.2023

Table of contents

01 Petit historique

02 Présentation de base

03 L'architecture réseau

04 La plateforme web

05 Sécurité et dangers

06 Les projets futurs

HOSTING minet

01

Petit historique

Hosting, dans sa forme actuel

01.04.2021

par D.C.

2 ans

de bêta, après quelques mois de coupure complète
(raisons de sécurités)

03.04.2023

version 1.0

HOSTING
minet

Depuis 2021 ...

Développement de fonctionnalités

Lourd travail de correction

HOSTING

minet

02

Présentation basique

Hosting au fond c’est quoi ?

Hosting : il y en a pour tout le monde 😋

Réseau

Administration reseau de toutes les VMs, enjeux de sécurité

DevOps

Application web pour les adherents et les admins pour contrôler 100% des VMs

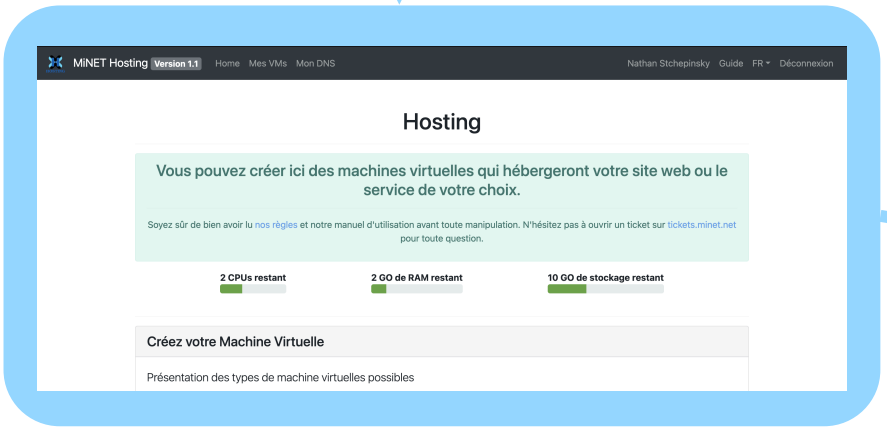
Sys admin

Full CI/CD, 2 hyperviseurs en charge de l'hébergement de toutes les machines des adherents. Déploiement automatique

HOSTING

minet

Hosting : Comment ça marche ?



Web Application

CI/CD

API de proxmox

PROXMOX

Accès Internet

SSH

WWW

HOSTING

minet

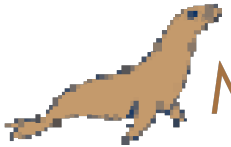
Hosting : Ça marche avec quoi ?

hosting.minet.net

CI/CD

API

Frontend



BDD

Hebergement

Aomine - Kars - Wammu

DNS

HashiCorp Packer

Génération des templates

Hosting

Bienvenue sur le panel d'administration Hosting
Ici vous pouvez gérer toutes les VM et entrées DNS

bg

Statistiques

VM créées	VM actives	Entrées DNS
112	85	62

HOSTING
minet

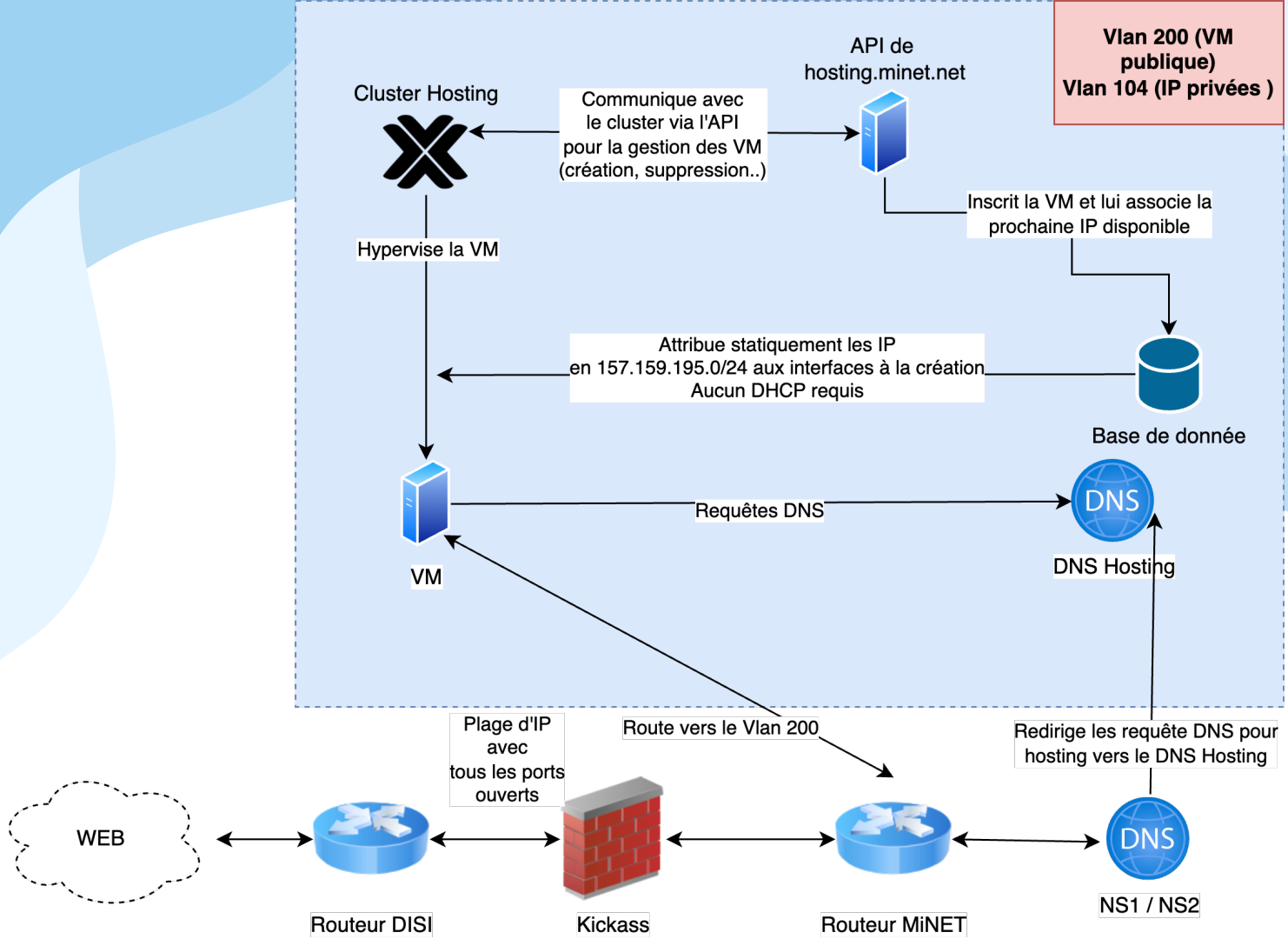
03

L’architecture réseau

Comment que c’est que ça communique toussa ?

HOSTING

minet


Vlan 200 (VM publique)
Vlan 104 (IP privées)
API de hosting.minet.net
Cluster Hosting
Communique avec le cluster via l'API pour la gestion des VM (création, suppression..)
Inscrit la VM et lui associe la prochaine IP disponible
Hypervise la VM
Attribue statiquement les IP en 157.159.195.0/24 aux interfaces à la création
Aucun DHCP requis
Base de donnée
Requêtes DNS
DNS
DNS Hosting
VM
Plage d'IP avec tous les ports ouverts
Route vers le Vlan 200
Redirige les requête DNS pour hosting vers le DNS Hosting
WEB
Routeur DISI
Kickass
Routeur MiNET
DNS
NS1 / NS2

Les VMs

Pool
replicated_3_times_hosting

Tous les soirs

Debian 11 avec fail2ban installé par défaut

Accès SSH par clé et comptes gérés par Cloud-init

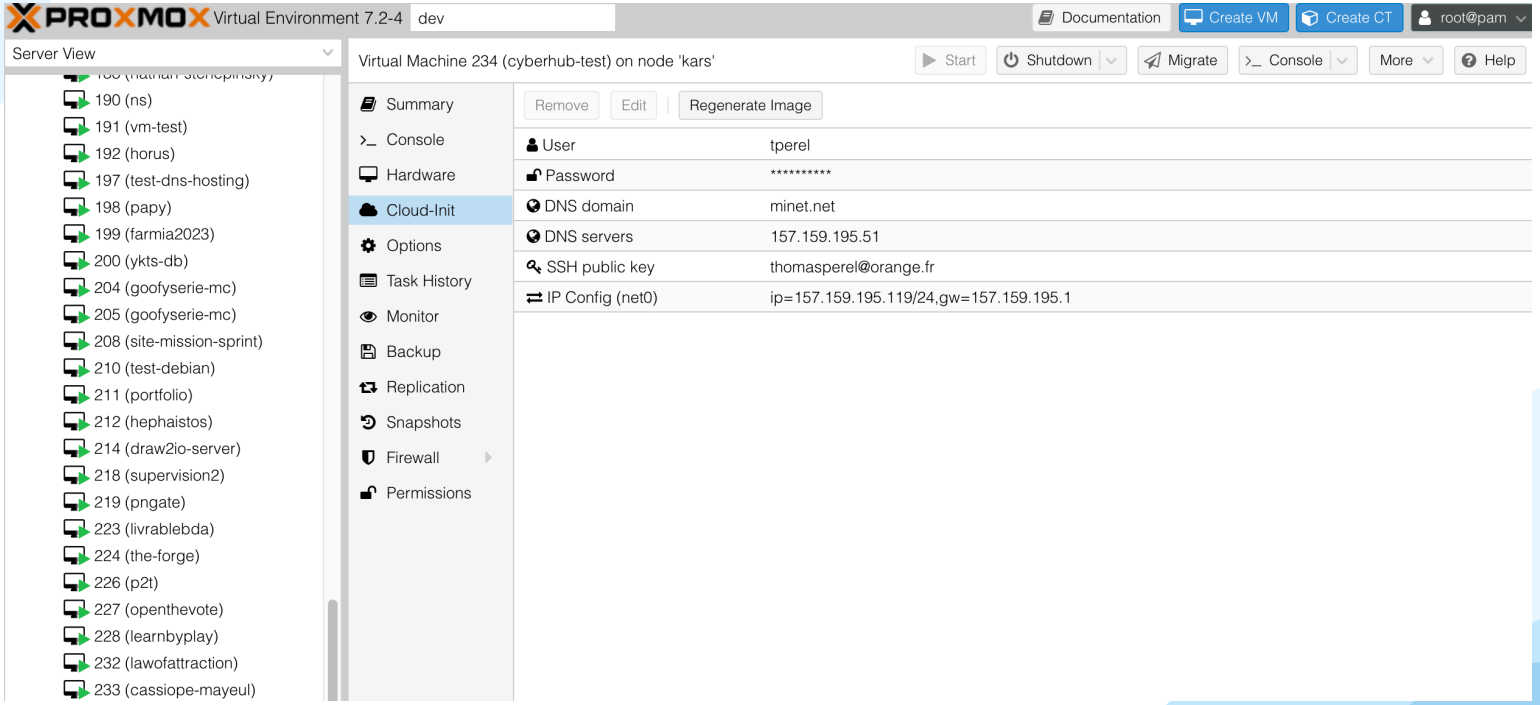
qemu-agent installé et activé

Clones de template proxmox pré-défini

HOSTING
minet

Cloud init

~~ou initialiseur infonuagique~~

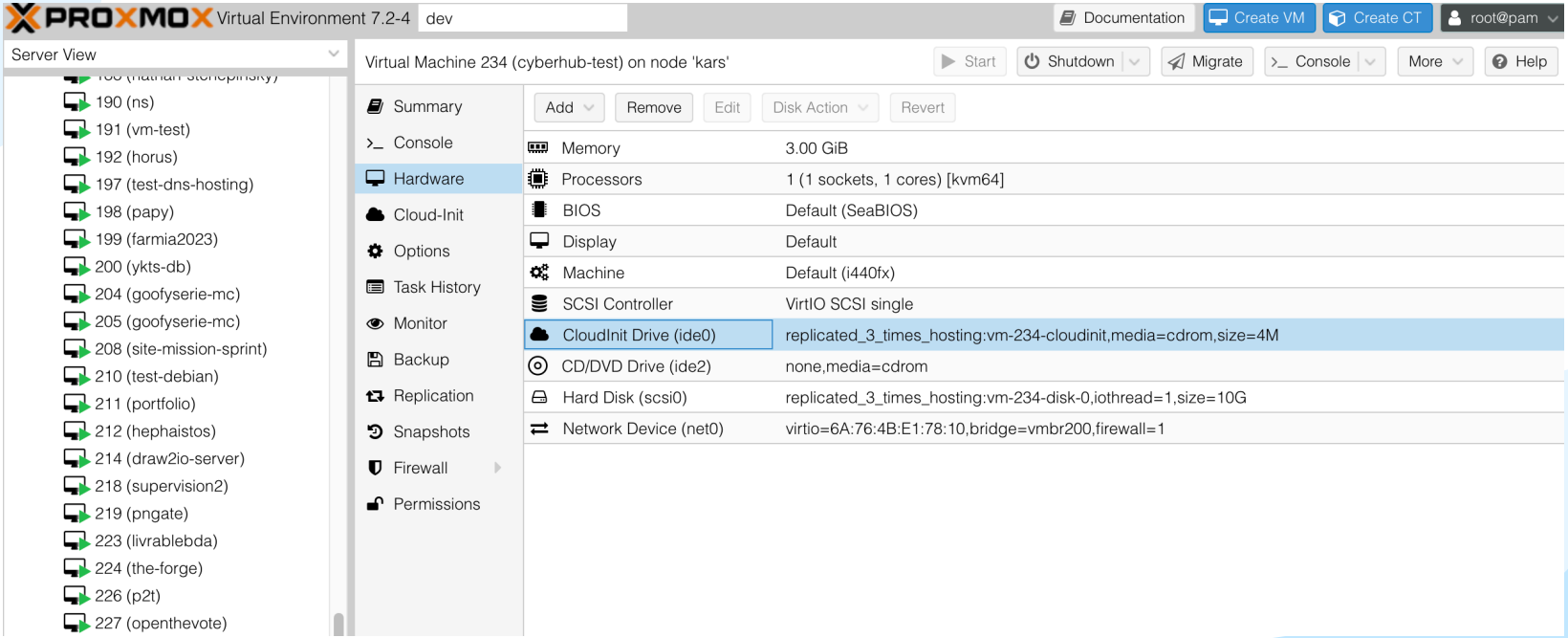


HOSTING

m.net

Cloud init

~~ou initialiseur infonuagique~~



HOSTING

Cloud init

~~ou initialiseur infonuagique~~

```yaml
manage_etc_hosts: true

users:
   - default

disable_root: true

apt_update: true

# Upgrade the instance on first boot
apt_upgrade: true

# Reboot after package install/update if necessary
apt_reboot_if_required: true

# Install useful packages
packages:
 - vim
 - fail2ban

# Write out new SSH daemon configuration. Standard debian 11 configuration
# apart from forbidding root login and disabling password authentication
write_files:
 - path: /etc/ssh/sshd_config
   content: |
      PermitRootLogin no
      PubkeyAuthentication yes
      PasswordAuthentication no
      PermitEmptyPasswords no
      ChallengeResponseAuthentication no
      UsePAM yes
      X11Forwarding yes
      PrintMotd no
      AcceptEnv LANG LC_*
      Subsystem   sftp  /usr/lib/openssh/sftp-server

```

```yaml

# The modules that run in the 'init' stage
cloud_init_modules:
 - seed_random
 - write-files
 - set_hostname
 - update_hostname
 - update_etc_hosts
 - ca-certs
 - users-groups
 - ssh

# The modules that run in the 'config' stage
cloud_config_modules:
 - set-passwords
 - ntp
 - timezone
 - disable-ec2-metadata

# The modules that run in the 'final' stage
cloud_final_modules:
 - package-update-upgrade-install
 - scripts-vendor
 - scripts-per-once
 - scripts-per-boot
 - scripts-per-instance
 - scripts-user
 - ssh-authkey-fingerprints
 - final-message

system_info:
   # This will affect which distro class gets used
   distro: debian
   # Default user name + that default users groups (if added/used)
   default_user:
     name: debian
     lock_passwd: true
     gecos: Debian
     groups: [adm, audio, cdrom, dialout, dip, floppy, netdev, plugdev, sudo, video]
     sudo: ["ALL=(ALL:ALL) ALL"]
     shell: /bin/bash
   paths:
      cloud_dir: /var/lib/cloud/
      templates_dir: /etc/cloud/templates/
      upstart_dir: /etc/init/
   package_mirrors:
     - arches: [default]
       failsafe:
         primary: http://deb.debian.org/debian
         security: http://security.debian.org/
   ssh_svcname: ssh
```

HOSTING

m.net

Processus de creation de VM

PROXMOX

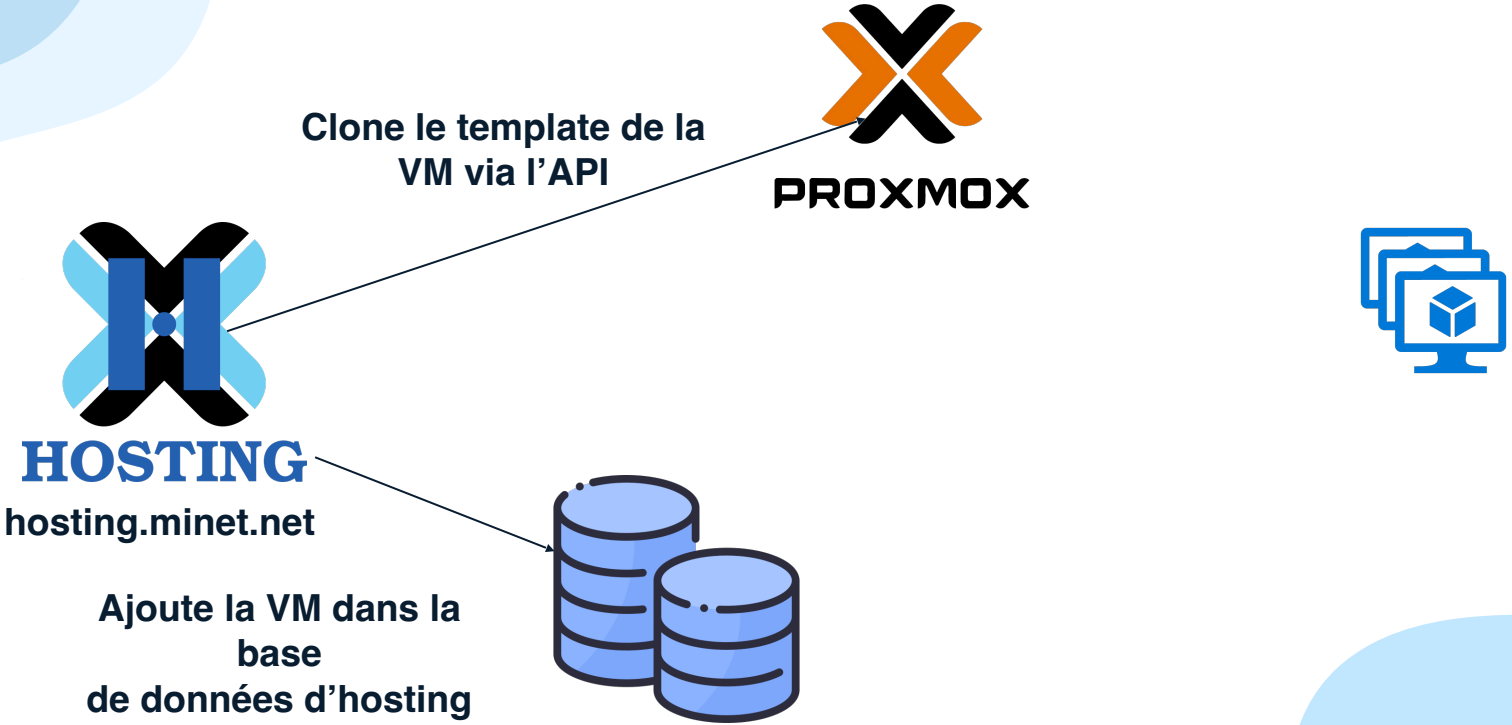
HOSTING

hosting.minet.net

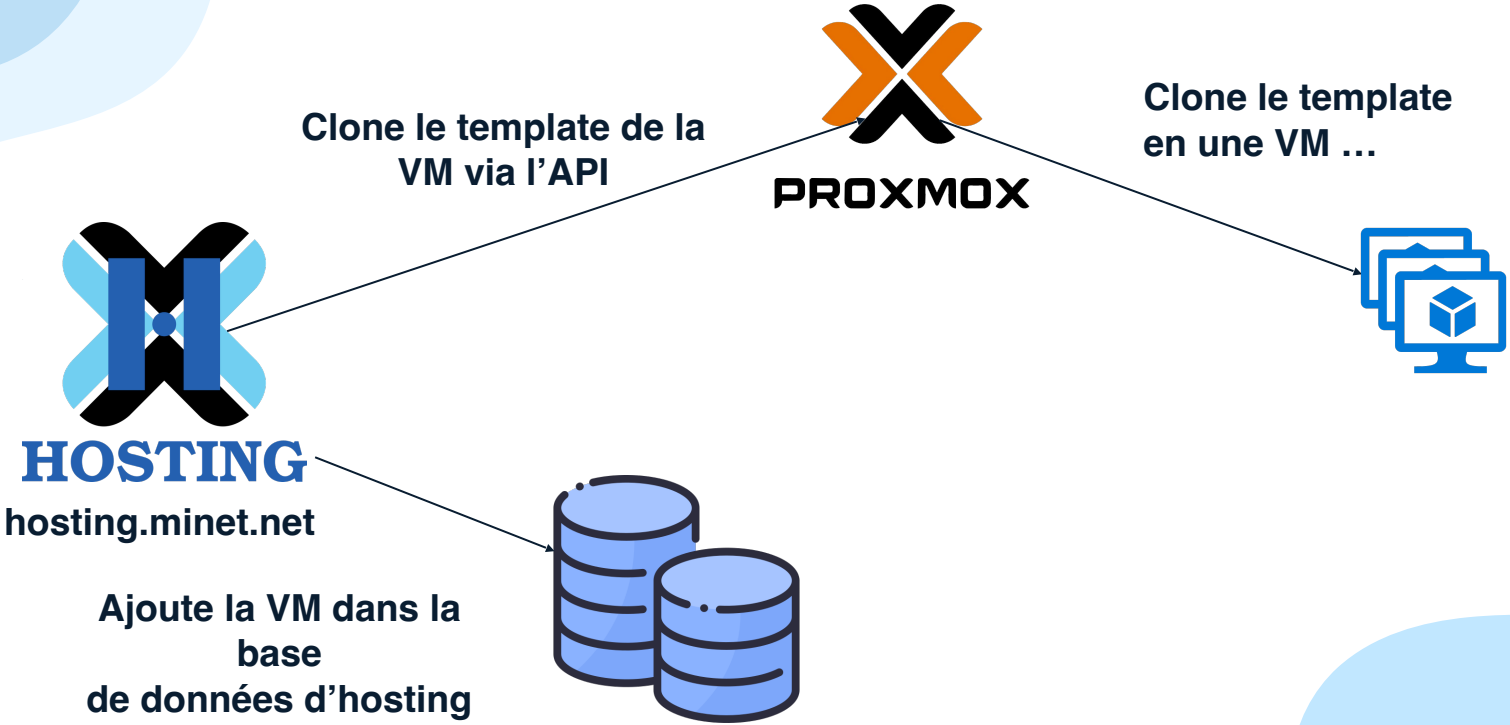
HOSTING

m.net

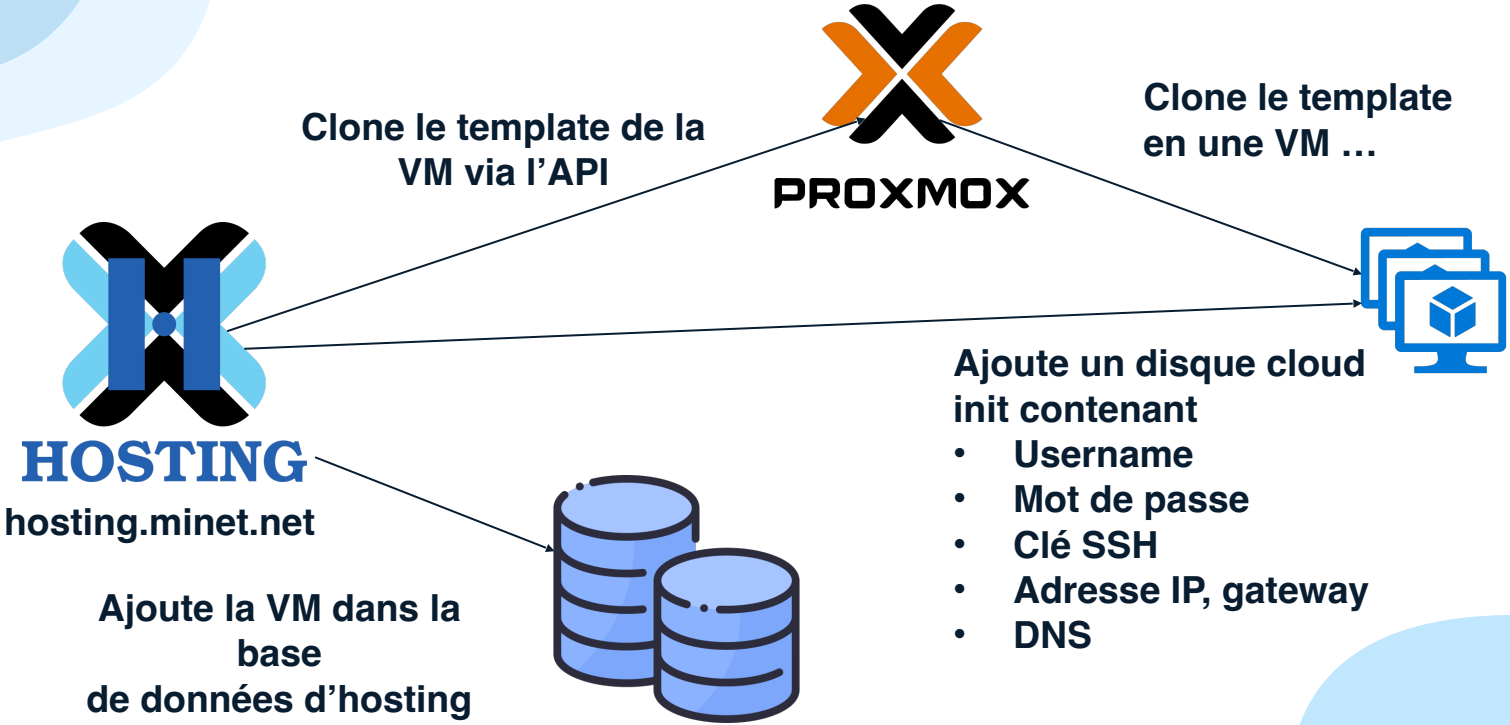
Processus de creation de VM



Processus de creation de VM



Processus de creation de VM



Processus de creation de VM

BOOT

Clone le template de la VM via l'API

PROXMOX

Clone le template en une VM …

HOSTING

hosting.minet.net

Applique le script cloud init au boot

Ajoute un disque cloud init contenant

- Username
- Mot de passe
- Clé SSH
- Adresse IP, gateway
- DNS

Ajoute la VM dans la base de données d'hosting

HOSTING
m.net

Processus d'ajout d'une entrée DNS

BIND 9

HOSTING

hosting.minet.net

DNS de MiNET

HOSTING

minet

Processus d'ajout d'une entrée DNS

Ajouter l'entrée dans le DNS d'hosting

BIND 9

HOSTING

hosting.minet.net

Ajouter la VM dans la base de données d'hosting

DNS de MiNET

HOSTING

minet

Processus d'ajout d'une entrée DNS

Ajouter l'entrée dans le DNS d'hosting

BIND 9

Redirige toutes les requêtes DNS pour h.minet.net

HOSTING

hosting.minet.net

Ajouter la VM dans la base de données d'hosting

DNS de MiNET

D’un point de vue adressage

IPs appartenant à l’école

192.168.104.0/24

157.159.195.0/24

- **157.159.195.1-157.159.195.10**
- **157.159.195.11-157.159.195.255**

D’un point de vue adressage

157.159.0.0/16 IPs appartenant à l’école

192.168.104.0/24

157.159.195.0/24

- **157.159.195.1-157.159.195.10**
- **157.159.195.11-157.159.195.255**

D’un point de vue adressage

157.159.0.0/16 IPs appartenant à l’école

192.168.104.0/24 Réseau administration privé

157.159.195.0/24

- **157.159.195.1-157.159.195.10**
- **157.159.195.11-157.159.195.255**

D’un point de vue adressage

157.159.0.0/16 IPs appartenant à l’école

192.168.104.0/24 Réseau administration privé

157.159.195.0/24 Réseau public, VLAN 200

- **157.159.195.1-157.159.195.10**
- **157.159.195.11-157.159.195.255**

D’un point de vue adressage

157.159.0.0/16 IPs appartenant à l’école

192.168.104.0/24 Réseau administration privé

157.159.195.0/24 Réseau public, VLAN 200

- **157.159.195.1-157.159.195.10** IP réservée pour l’administration
- **157.159.195.11-157.159.195.255**

D’un point de vue adressage

157.159.0.0/16 IPs appartenant à l’école

192.168.104.0/24 Réseau administration privé

157.159.195.0/24 Réseau public, VLAN 200

- **157.159.195.1-157.159.195.10** IP réservée pour l’administration
- **157.159.195.11-157.159.195.255** IP attribuées aux VMs

04

La plateforme web

Objectif : dépasser les 25 000 contributions sur le repo github

04

La plateforme web

Objectif : dépasser les 25 000 contributions sur le repo github

Ouais je flex, mais c'est ma formation, donc je
fais ce que je veux

HOSTING
minet

04

La plateforme web

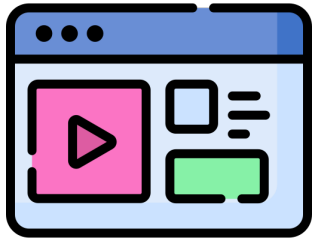
Objectif : dépasser les 25 000 contributions sur le repo github

Ouais je flex, mais c'est ma formation, donc je
fais ce que je veux

Nicolas, tg

HOSTING
minet

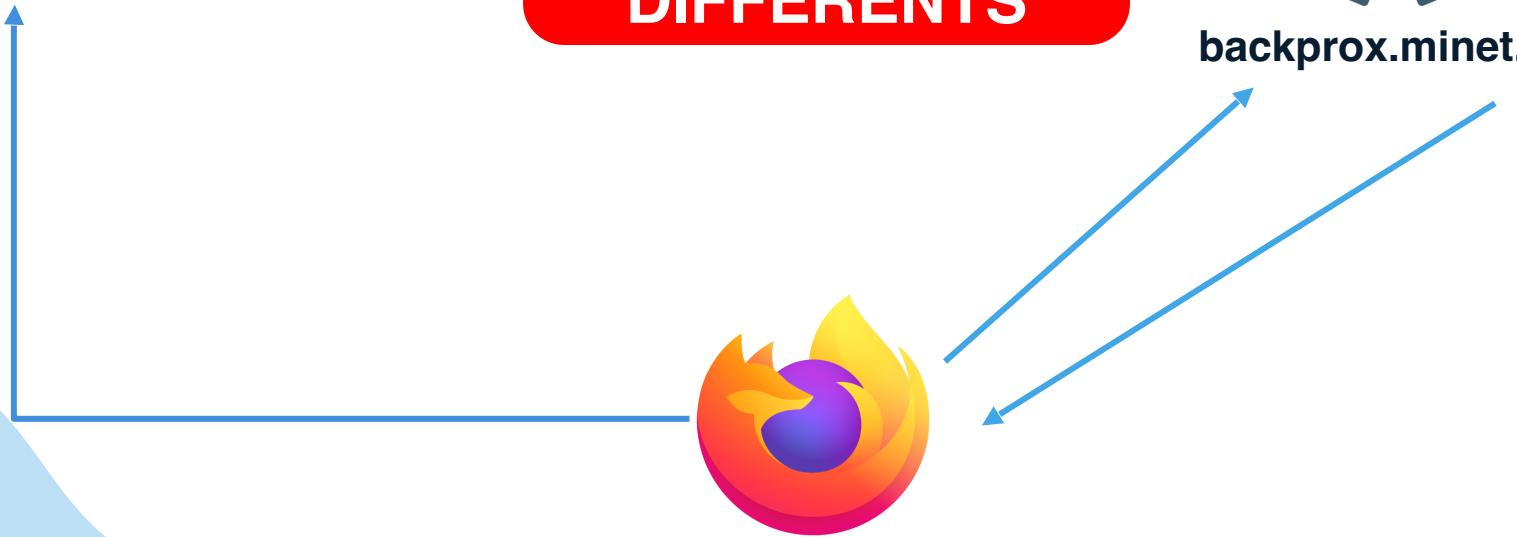
Comment que c'est que ça marche ?



hosting.minet.net

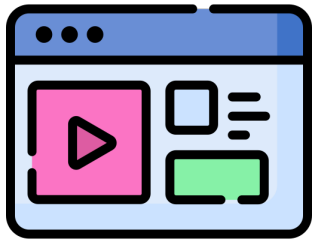
2 SERVEURS WEB DIFFÉRENTS

backprox.minet.net



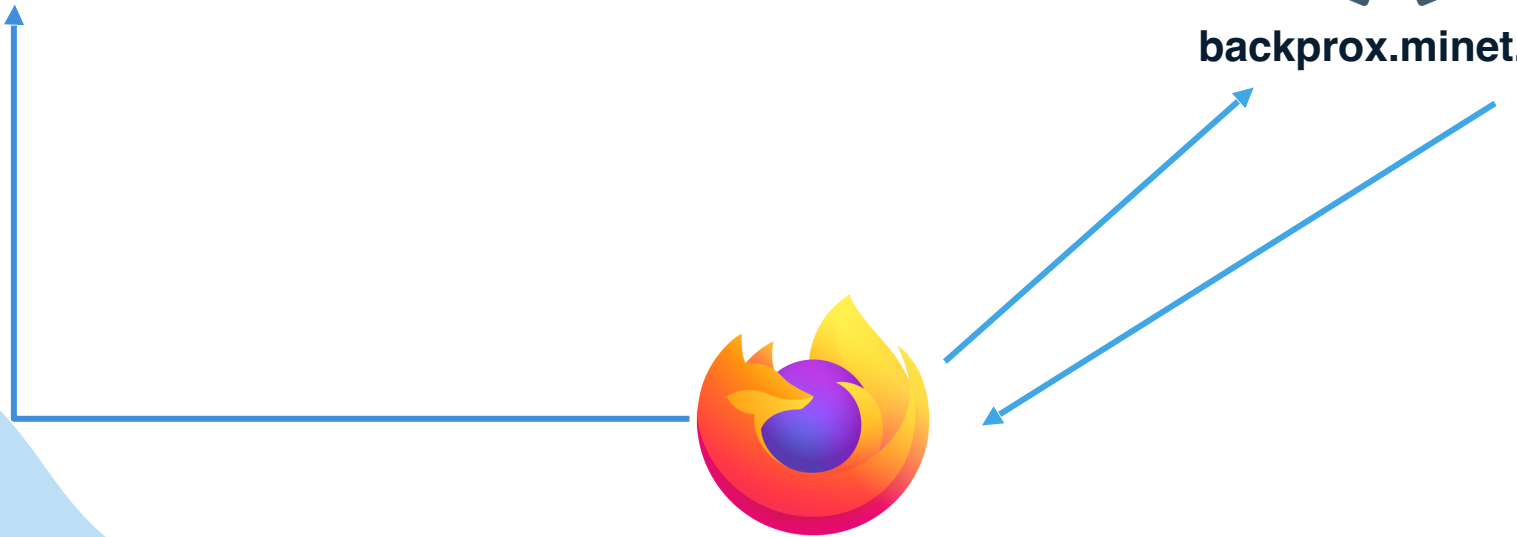
HOSTING minet

Comment que c'est que ça marche ?



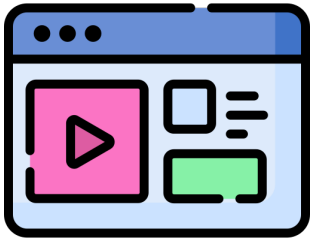
hosting.minet.net

backprox.minet.net



HOSTING minet

Comment que c'est que ça marche ?



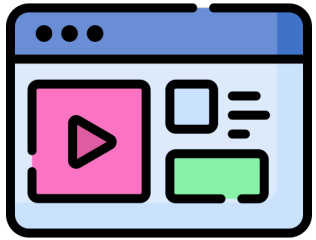
hosting.minet.net

backprox.minet.net

https://hosting.minet.net/vms

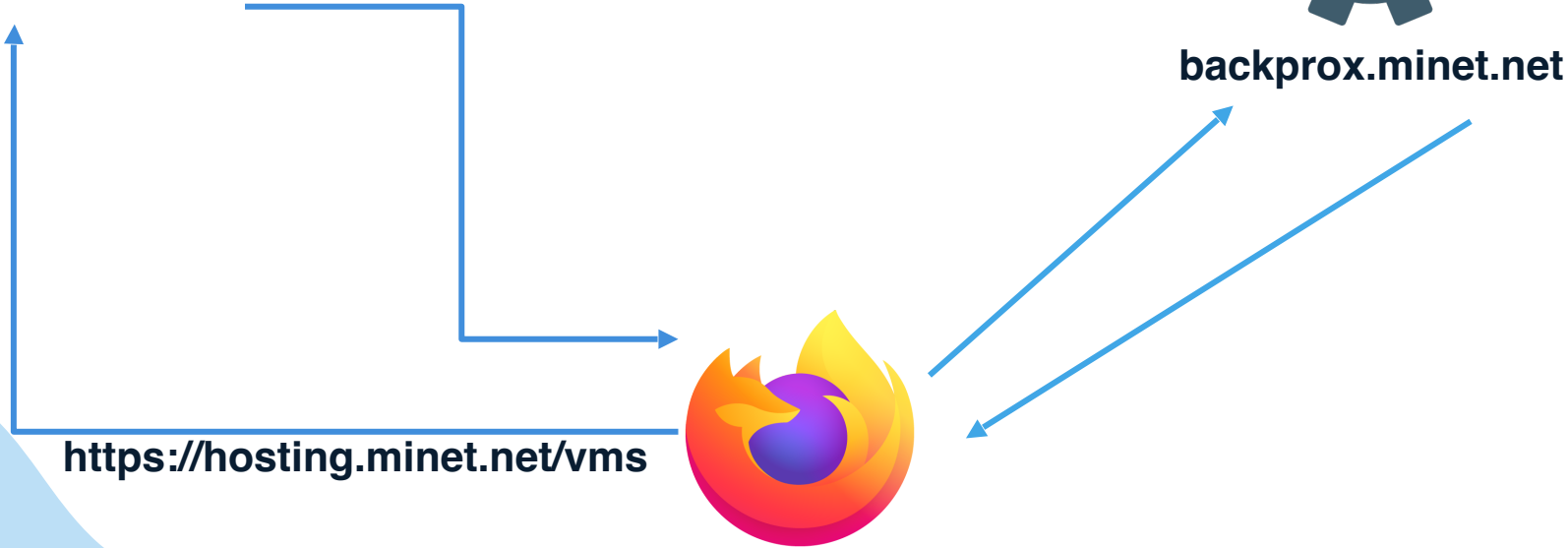
HOSTING minet

Comment que c'est que ça marche ?



hosting.minet.net

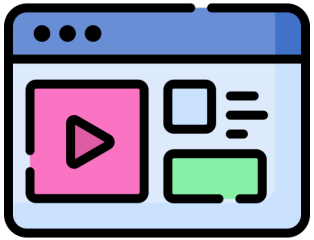
backprox.minet.net



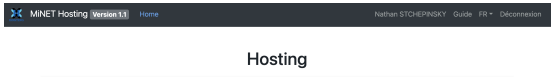
https://hosting.minet.net/vms

HOSTING minet

Comment que c'est que ça marche ?



hosting.minet.net

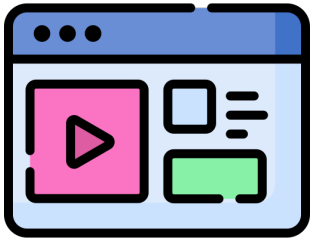


backprox.minet.net

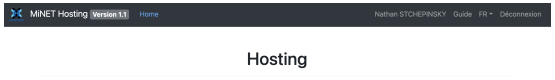
https://hosting.minet.net/vms

HOSTING
minet

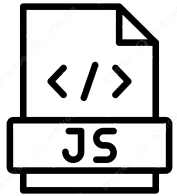
Comment que c'est que ça marche ?



hosting.minet.net



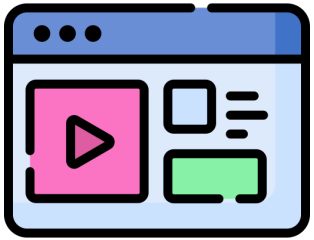
backprox.minet.net



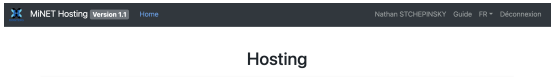
https://hosting.minet.net/vms

HOSTING
minet

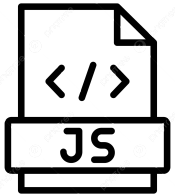
Comment que c'est que ça marche ?



hosting.minet.net



backprox.minet.net

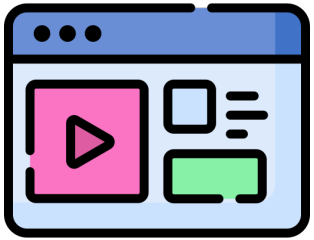


https://backprox.minet.net/api/1.0.0/vms

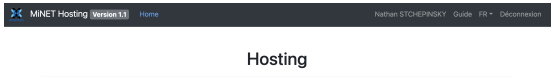
https://hosting.minet.net/vms

HOSTING
minet

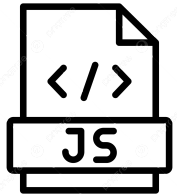
Comment que c'est que ça marche ?



hosting.minet.net



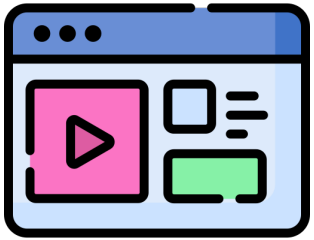
backprox.minet.net



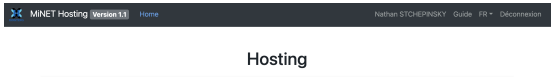
https://hosting.minet.net/vms

HOSTING
minet

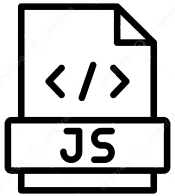
Comment que c'est que ça marche ?

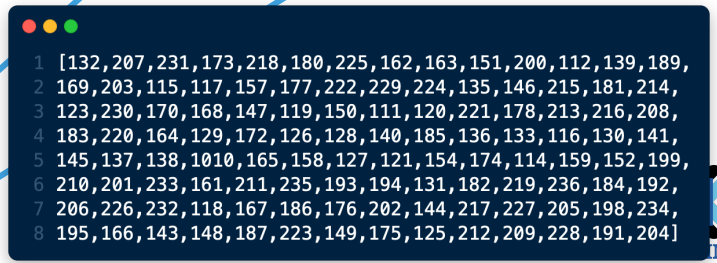


hosting.minet.net



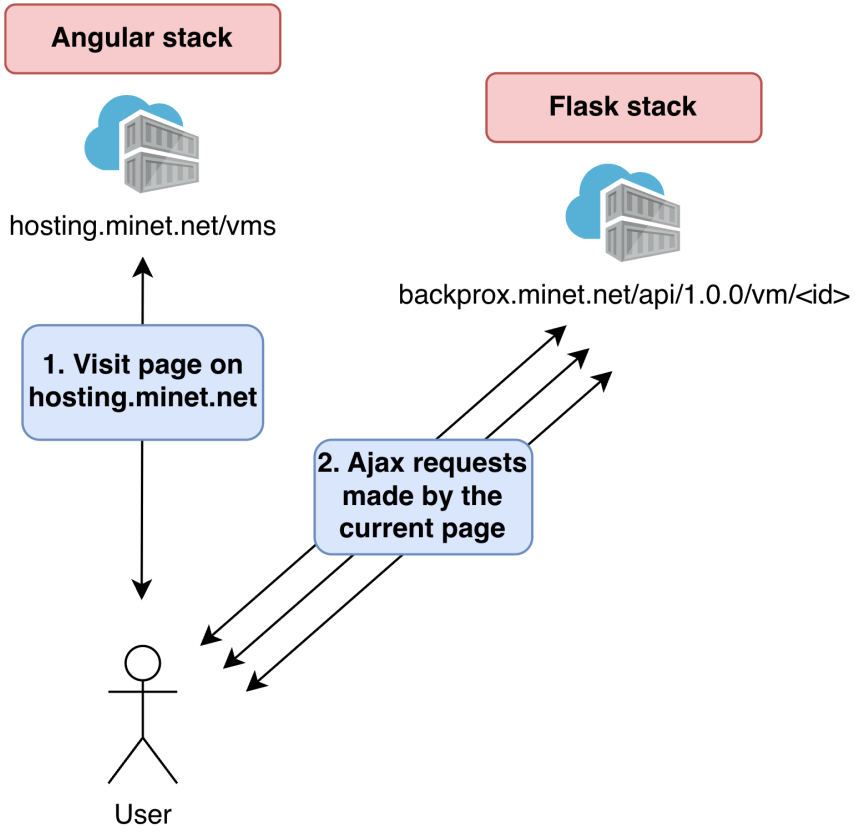
backprox.minet.net





https://hosting.minet.net/vms

Comment que c'est que ça marche ?



P’tit quizz

Allez sur hosting.minet.net et connectez vous en tant qu’adhérent.

Combien de requêtes votre navigateur fait-il à l’API ?

- 0
- 8
- 9
- 10

P’tit quizz

Allez sur hosting.minet.net et connectez vous en tant qu’adhérent.

Combien de requêtes votre navigateur fait-il à l’API ?

- 0
- 8
- 9
- 10

P’tit quizz

Allez sur hosting.minet.net et connectez vous en tant qu’adhérent.

Combien de requêtes votre navigateur fait-il à l’API ?

0

8

Oui je sais trou du cul, tu as plus de requêtes parce que tu as déjà des VMs, mais qui l’a demandé ?

9

10

HOSTING

minet

P’tit quizz

Allez sur hosting.minet.net et connectez vous en tant qu’adhérent.

Combien de requêtes votre navigateur fait-il à l’API ?

0

8

Oui je sais trou du cul, tu as plus de requêtes parce que tu as déjà des VMs, mais qui l’a demandé ?

9

10

On regarde ensemble ?

HOSTING

minet

Partie I
Le frontend

app-routing.module.ts : les routes

```
const routes: Routes = [
  {path: '', component: HomeComponent},
  {path: 'vms', component: VmsComponent},
  {path: 'vms/:vmid', component: VmComponent},
  {path: 'dns', component: DnsComponent},
  {path: 'legal', component: LegalComponent},
  {path: 'manual', component: ManualComponent},
  {path: 'history', component: HistoryComponent},
  {path: '**', redirectTo: ''},
];
```

Partie I
Le frontend

Les components

Contient tout le code HTML d'un composant
(le corps la page des VMs, home, etc …)

Est intégré directement dans l'index.html, en fonction de la route

HOSTING
minet

Partie I
Le frontend

index.html : la base

```html
<!doctype html>
<html>
      <head></head>

      <body>
            <app-root></app-root>
      </body>
</html>
```

<app-root>

app.component.html

```html
<app-navbar></app-navbar>

<ng-template></ng-template>

<app-footer></app-footer>

```

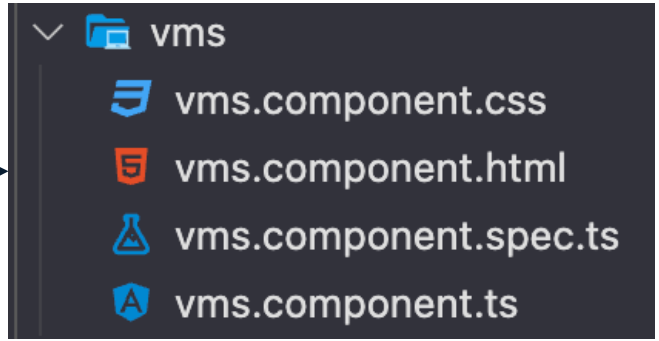
Partie I
Le frontend

app.component.html

```
1 <app-navbar></app-navbar>
2
3 <ng-template></ng-template>
4
5 <app-footer></app-footer>
6
```

<ng-template>

component



ANGULAR

HOSTING
minet

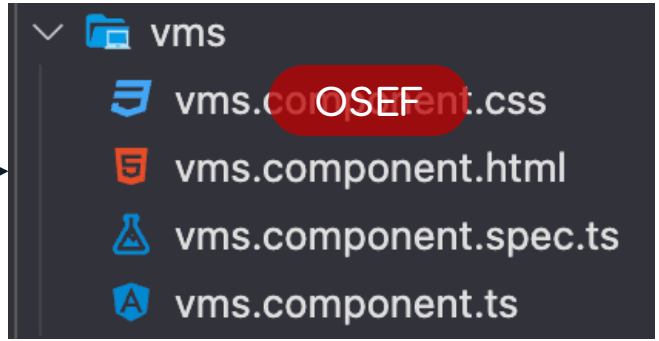
Partie I
Le frontend

app.component.html

```
1 <app-navbar></app-navbar>
2
3 <ng-template></ng-template>
4
5 <app-footer></app-footer>
6
```

<ng-template>

component



ANGULAR

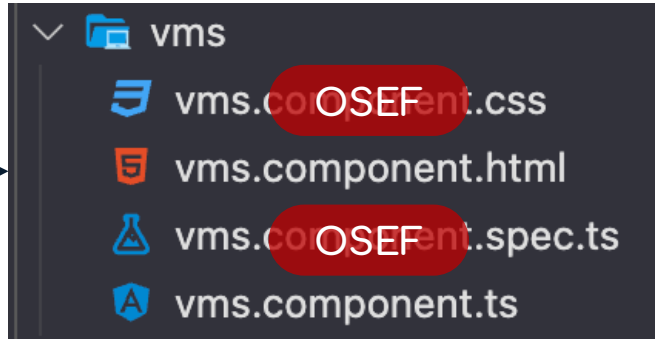
Partie I
Le frontend

app.component.html

```
1 <app-navbar></app-navbar>
2
3 <ng-template></ng-template>
4
5 <app-footer></app-footer>
6
```

<ng-template>

component



ANGULAR

Partie I
Le frontend

component.ts
Typescript TABASSE
javascript

Bon, écoute-moi bien. TypeScript déchire carrément la gueule de JavaScript, et je vais te dire pourquoi.
D'abord, TypeScript est comme le grand frère badass de JavaScript. Il lui met une bonne claque avec son système de typage statique. Tu vois, JavaScript est un langage permissif, il te laisse te fourrer dans n'importe quelle galère sans te prévenir. Mais TypeScript, lui, il te dit "Hé, attention ! Tu essaies de mélanger des types comme un idiot !". Il t'oblige à déclarer les types de tes variables, tes paramètres de fonction et tes retours de valeurs. Ça veut dire moins de bugs stupides qui se produisent lors de l'exécution, parce que TypeScript les repère avant même que tu lances ton code de merde.
Ensuite, TypeScript, c'est le mec qui sait comment bien gérer la programmation orientée objet. Il sait comment utiliser les classes, l'héritage, les interfaces et tout le bordel. Ça rend ton code beaucoup plus organisé et compréhensible. Fini le bordel incompréhensible de JavaScript où tu ne sais pas qui fait quoi. TypeScript te permet de structurer ton code de façon claire et de définir des contrats entre les différentes parties de ton programme. C'est comme mettre de l'ordre dans la putain de jungle de JavaScript.
Et attends, y'a encore mieux ! TypeScript a un système de compilation qui transforme ton code TypeScript en JavaScript utilisable par les navigateurs et les moteurs JavaScript. Tu peux donc utiliser les dernières fonctionnalités de JavaScript tout en profitant des améliorations de TypeScript. En plus, ça optimise ton code pour qu'il soit plus court et plus rapide. Ça donne des performances de ouf, mec.
Pour couronner le tout, TypeScript a une putain de communauté qui déchire tout. Tous les grands frameworks comme Angular, React et Vue.js le supportent à fond. Ça veut dire que tu peux utiliser les meilleures technologies tout en bénéficiant des avantages de TypeScript. C'est comme si tu rejoignais une bande de mecs badass qui savent ce qu'ils font.
Donc voilà, en gros, TypeScript met une sacrée raclée à JavaScript. Son système de typage statique, sa programmation orientée objet, sa compilation et sa communauté de malades font de lui le choix ultime. Si tu veux coder comme un vrai pro et mettre JavaScript à genoux, choisis TypeScript.

ANGULAR

ChatGPT

Partie I
Le frontend

component.html
Intègre utilise et est utilisé par typescript

```
<div class="container text-center" *ngIf="!loading && (user.chartevalidated || user.admin)">
    <h1 class="text-center mb-2">VM: {{ user.vms[0].name }}</h1>

    <div class="alert"
         [ngClass]="{
                    'alert-danger': deletionStatus == 'deleting' || vm_has_error,
                    'alert-secondary' : user.vms[0].status === 'stopped',
                    'alert-info': user.vms[0].status === 'creating',
                    'alert-warning': user.vms[0].status === 'booting',
                    'alert-success': user.vms[0].status === 'running' || deletionStatus == 'deleted'
                    }">
    </div>
</div>
```

HOSTING
minet

Partie I
Le frontend

Les pages

- **/ : Home -** Création de VMs
- **/vms : Listes des VMs –** Liste de toutes les VMs pour les admins
- **/vms/<vmid> : Vue particulière d'une**
- **/dns : Liste des entrées DNS –** Liste de toutes les entrées DNS pour les admins

HOSTING
minet

Partie I
Le frontend

Les fonctionalitées

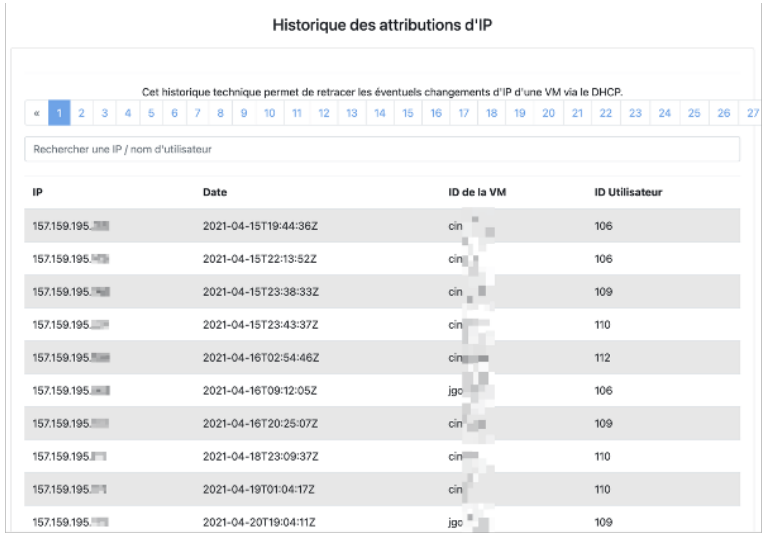
- **/ : Home -** Création de VMs
- **/vms : Listes des VMs –** Liste de toutes les VMs pour les admins
- **/vms/<vmid> : Vue particulière d'une**
- **/dns : Liste des entrées DNS –** Liste de toutes les entrées DNS pour les admins

HOSTING
minet

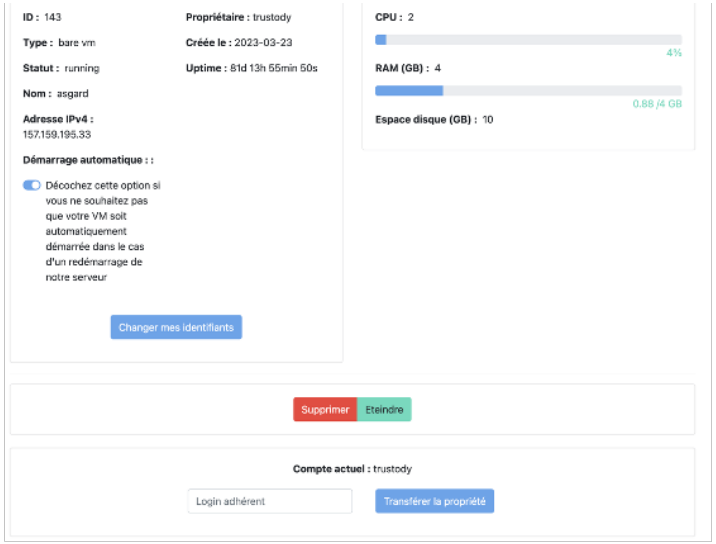
Partie I
Le frontend

Les fonctionnalités

Historique des IPs



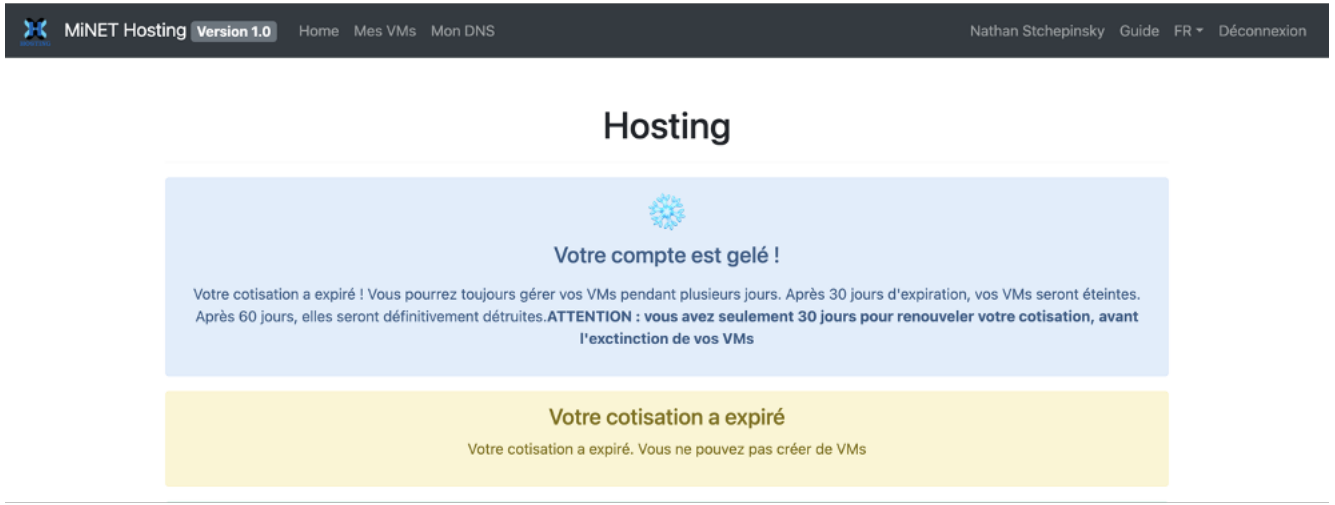
Transfert de propriété



Partie I
Le frontend

Les fonctionnalités

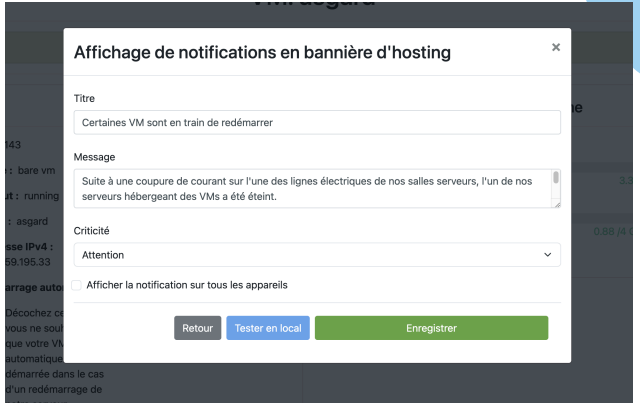
Différents niveau de freeze de compte :
le freeze_state

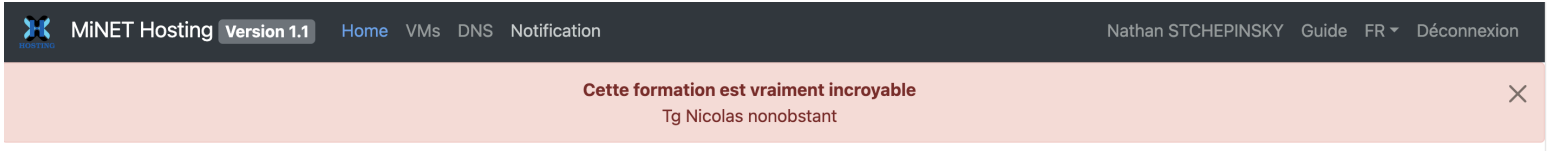


Partie I
Le frontend

Les fonctionnalités

Les notifications par bannières :



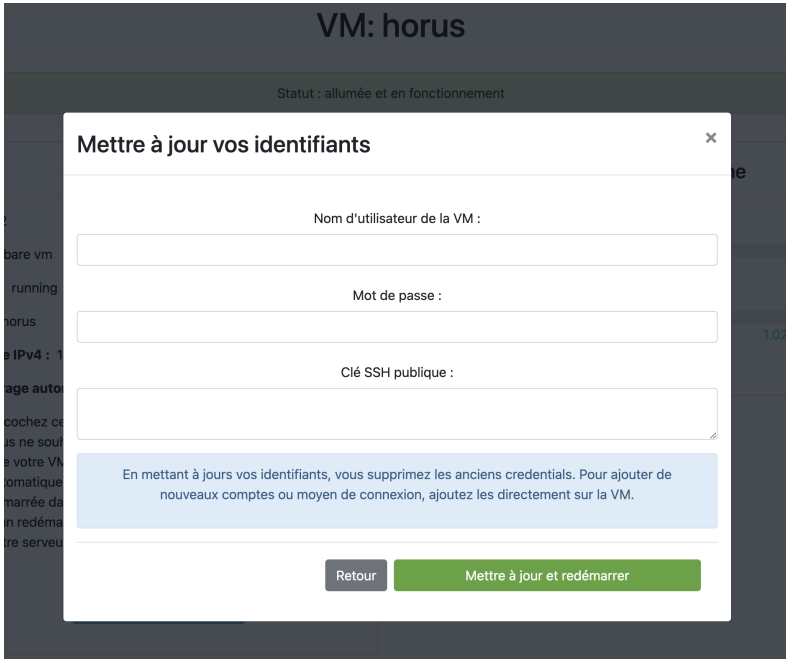


Hosting

Partie I
Le frontend

Les fonctionnalités

Update des credentials d'une VM



Partie I
Le frontend

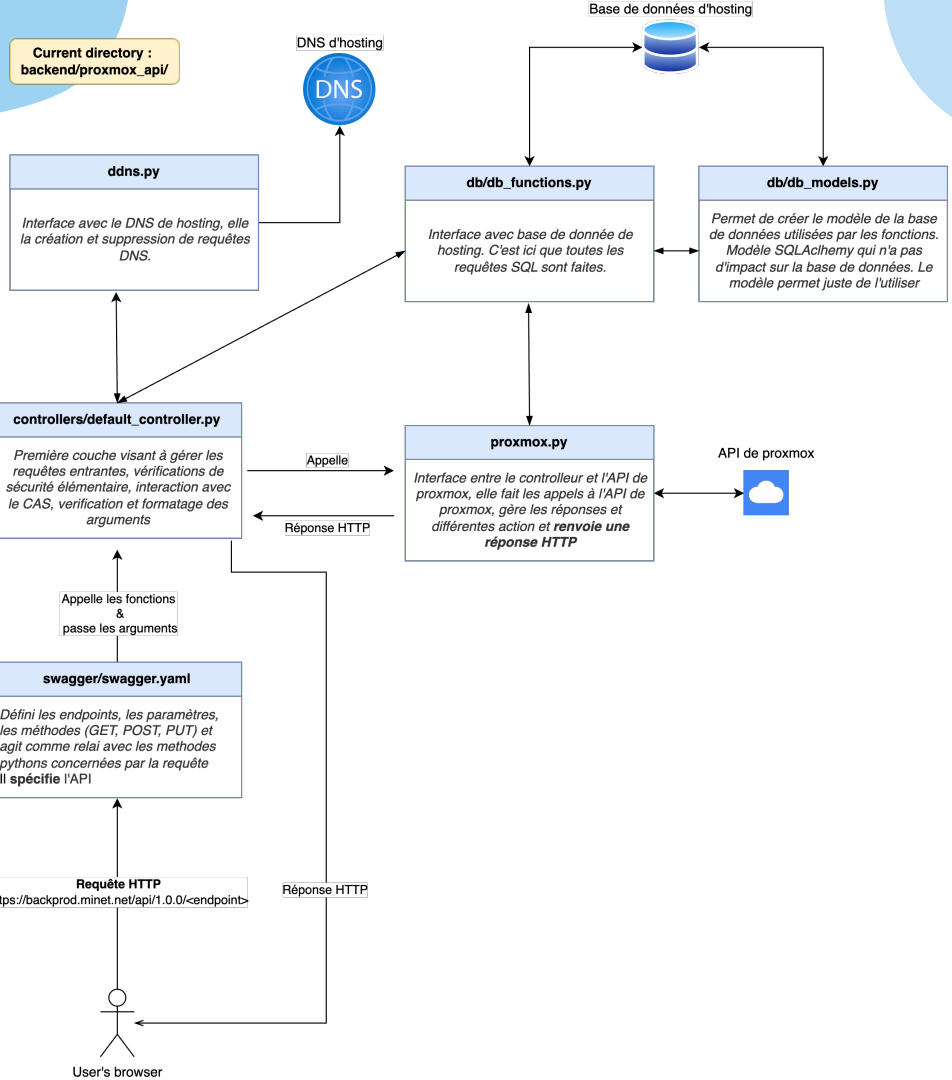
La fonctionnalité principale :
Les ✨ressources dynamiques✨

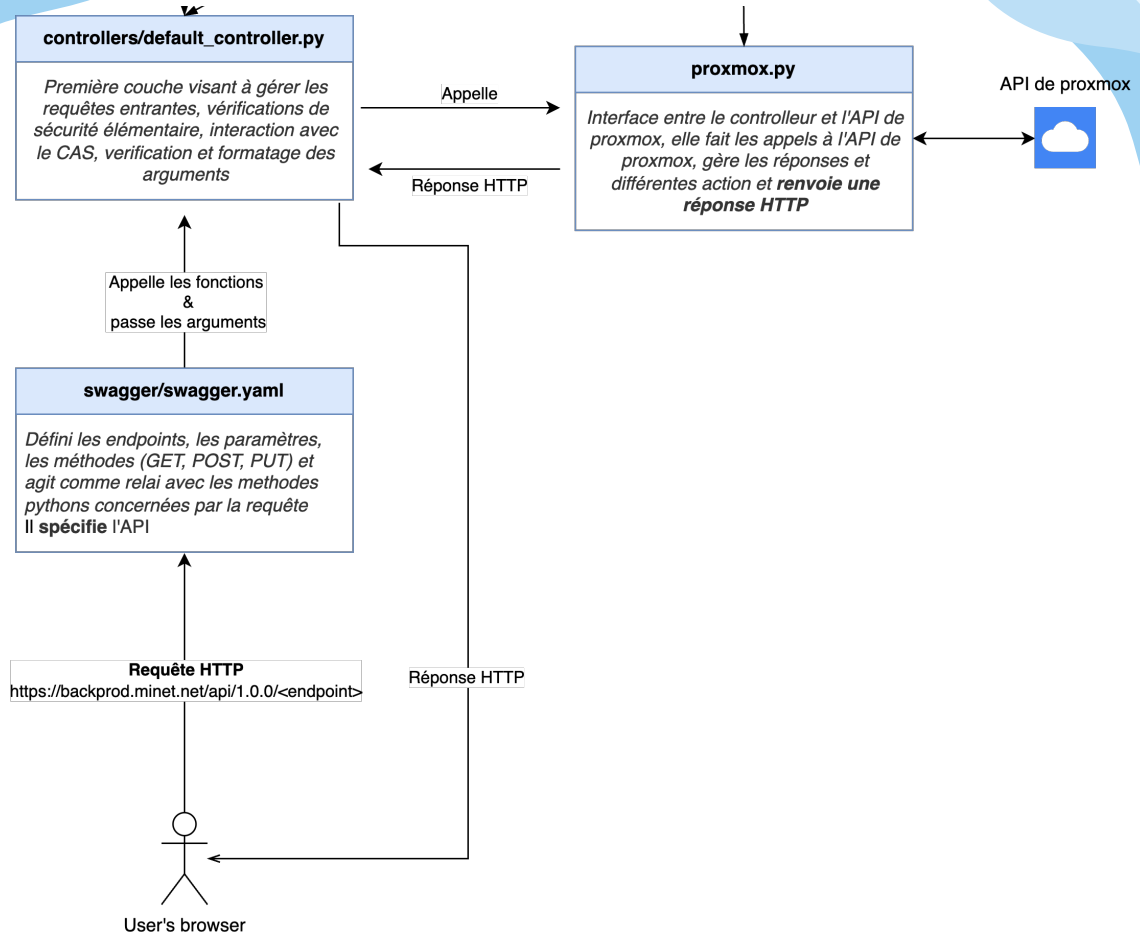
6 CPUs restant **9 GO de RAM restant** **30 GO de stockage restant**

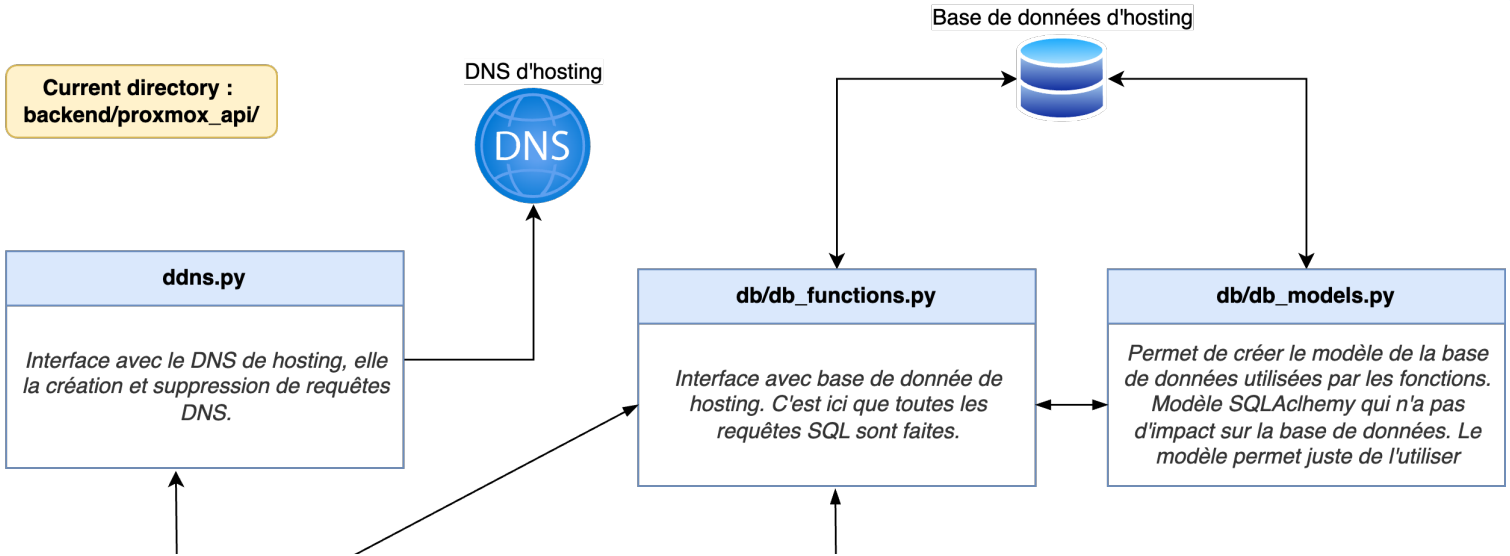
HOSTING minet

Partie II
L'API

https://backprox.minet.net/api/1.0.0/ui


Current directory :
backend/proxmox_api/
DNS d'hosting
DNS
Base de données d'hosting
ddns.py
Interface avec le DNS de hosting, elle la création et suppression de requêtes DNS.
db/db_functions.py
Interface avec base de donnée de hosting. C'est ici que toutes les requêtes SQL sont faites.
db/db_models.py
Permet de créer le modèle de la base de données utilisées par les fonctions. Modèle SQLAclhemy qui n'a pas d'impact sur la base de données. Le modèle permet juste de l'utiliser
controllers/default_controller.py
Première couche visant à gérer les requêtes entrantes, vérifications de sécurité élémentaire, interaction avec le CAS, verification et formatage des arguments
Appelle
Réponse HTTP
proxmox.py
Interface entre le controlleur et l'API de proxmox, elle fait les appels à l'API de proxmox, gère les réponses et différentes action et renvoie une réponse HTTP
API de proxmox
Appelle les fonctions
&
passe les arguments
swagger/swagger.yaml
Défini les endpoints, les paramètres, les méthodes (GET, POST, PUT) et agit comme relai avec les methodes pythons concernées par la requête
Il spécifie l'API
Requête HTTP
https://backprod.minet.net/api/1.0.0/<endpoint>
Réponse HTTP
User's browser

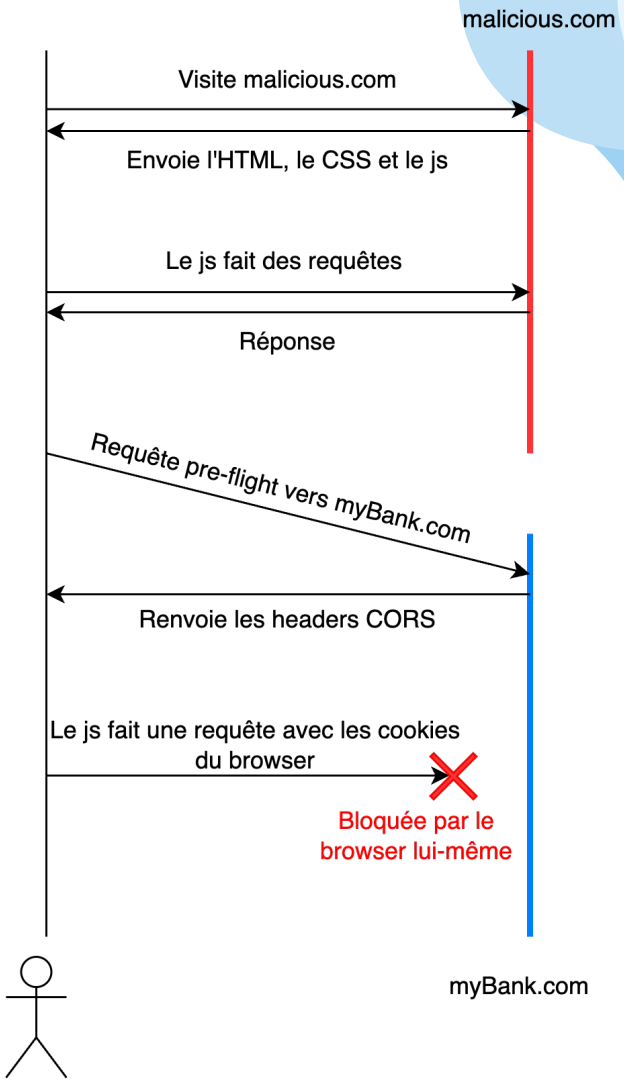

controllers/default_controller.py
Première couche visant à gérer les requêtes entrantes, vérifications de sécurité élémentaire, interaction avec le CAS, verification et formatage des arguments
Appelle
Réponse HTTP
proxmox.py
Interface entre le controlleur et l'API de proxmox, elle fait les appels à l'API de proxmox, gère les réponses et différentes action et renvoie une réponse HTTP
API de proxmox
Appelle les fonctions & passe les arguments
swagger/swagger.yaml
Défini les endpoints, les paramètres, les méthodes (GET, POST, PUT) et agit comme relai avec les methodes pythons concernées par la requête
Il spécifie l'API
Requête HTTP
https://backprod.minet.net/api/1.0.0/<endpoint>
Réponse HTTP
User's browser



HOSTING
minet

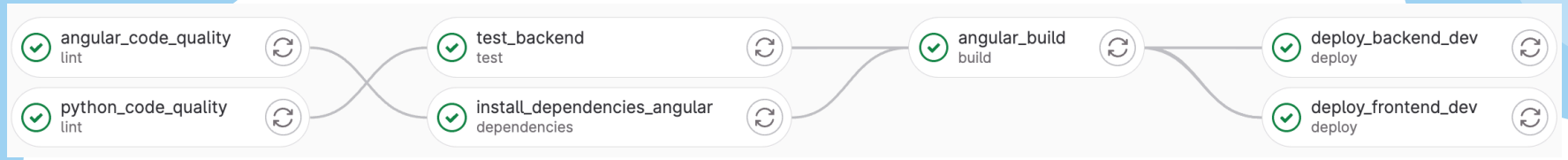
Partie II
L'API

Petit digression, les CORS



Partie II
L'API

Les tests de code



Partie II
L'API
Les tests de code

test_backend

69 tests | 0 failures | 0 errors | 100% success rate | 352.03s

Tests

Suite	Name	Filename	Status	Duration	Details
test.integration.test_vm_life_10G	test_valid_vm_creation		✓	61.45s	View details
test.integration.test_vm_life_30G	test_valid_vm_creation		✓	56.14s	View details
test.integration.test_vm_life_20G	test_valid_vm_creation		✓	50.28s	View details
test.integration.test_vm_life_20G	test_vm_start		✓	37.72s	View details
test.integration.test_vm_life_30G	test_vm_start		✓	34.95s	View details

Partie II
L'API

Les tests de code

69 tests

test_backend

69 tests | 0 failures | 0 errors | 100% success rate | 352.03s

Tests

Suite	Name	Filename	Status	Duration	Details
test.integration.test_vm_life_10G	test_valid_vm_creation		✓	61.45s	View details
test.integration.test_vm_life_30G	test_valid_vm_creation		✓	56.14s	View details
test.integration.test_vm_life_20G	test_valid_vm_creation		✓	50.28s	View details
test.integration.test_vm_life_20G	test_vm_start		✓	37.72s	View details
test.integration.test_vm_life_30G	test_vm_start		✓	34.95s	View details

Partie II
L'API

Les tests de code

69 tests

test_backend

69 tests | 0 failures | 0 errors | 100% success rate | 352.03s

Tests

Suite	Name	Filename	Status	Duration	Details
test.integration.test_vm_life_10G	test_valid_vm_creation		✓	61.45s	View details
test.integration.test_vm_life_30G	test_valid_vm_creation		✓	56.14s	View details
test.integration.test_vm_life_20G	test_valid_vm_creation		✓	50.28s	View details
test.integration.test_vm_life_20G	test_vm_start		✓	37.72s	View details
test.integration.test_vm_life_30G	test_vm_start		✓	34.95s	View details

46% de coverage

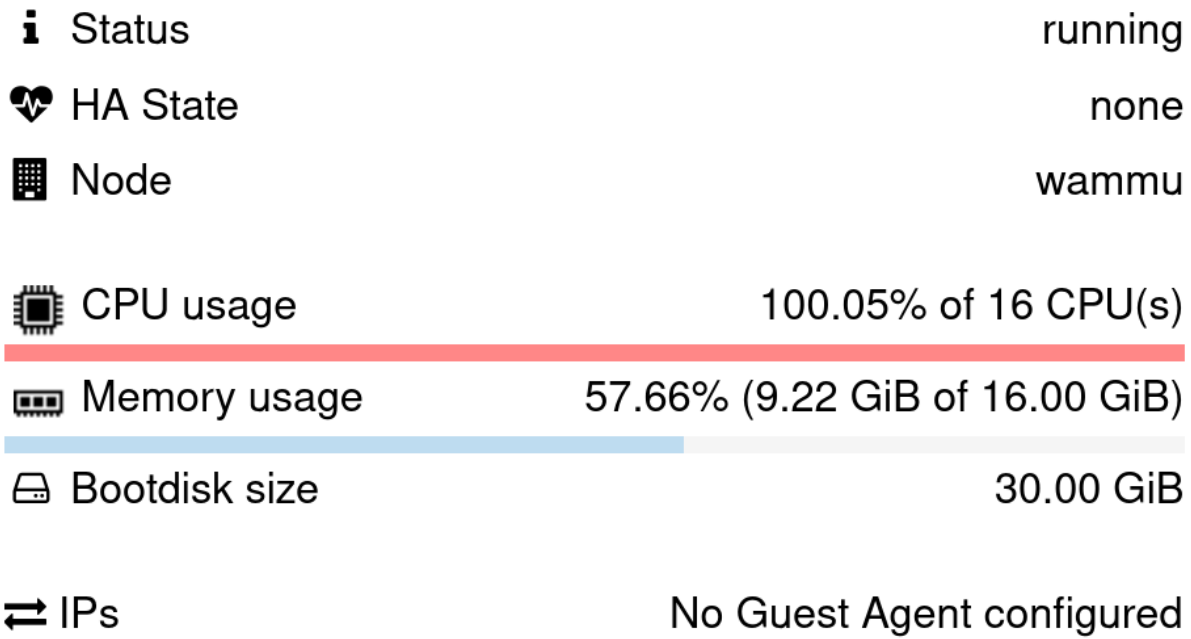
HOSTING
minet

05

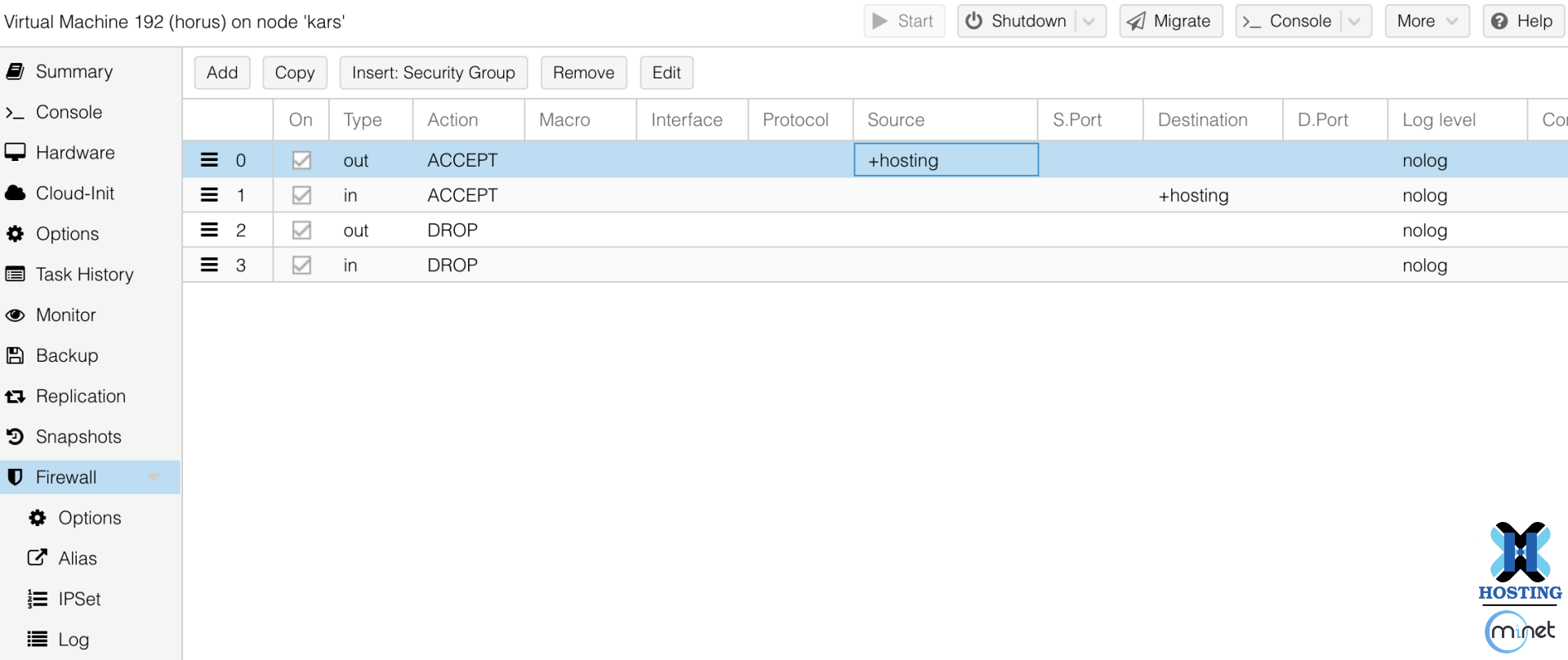
Sécurité et dangers

L’un des points le plus important

Le risque ...


Status
running
HA State
none
Node
wammu
CPU usage
100.05% of 16 CPU(s)
Memory usage
57.66% (9.22 GiB of 16.00 GiB)
Bootdisk size
30.00 GiB
IPs
No Guest Agent configured

Sécurité anti-spoofing : le firewall de proxmox ....

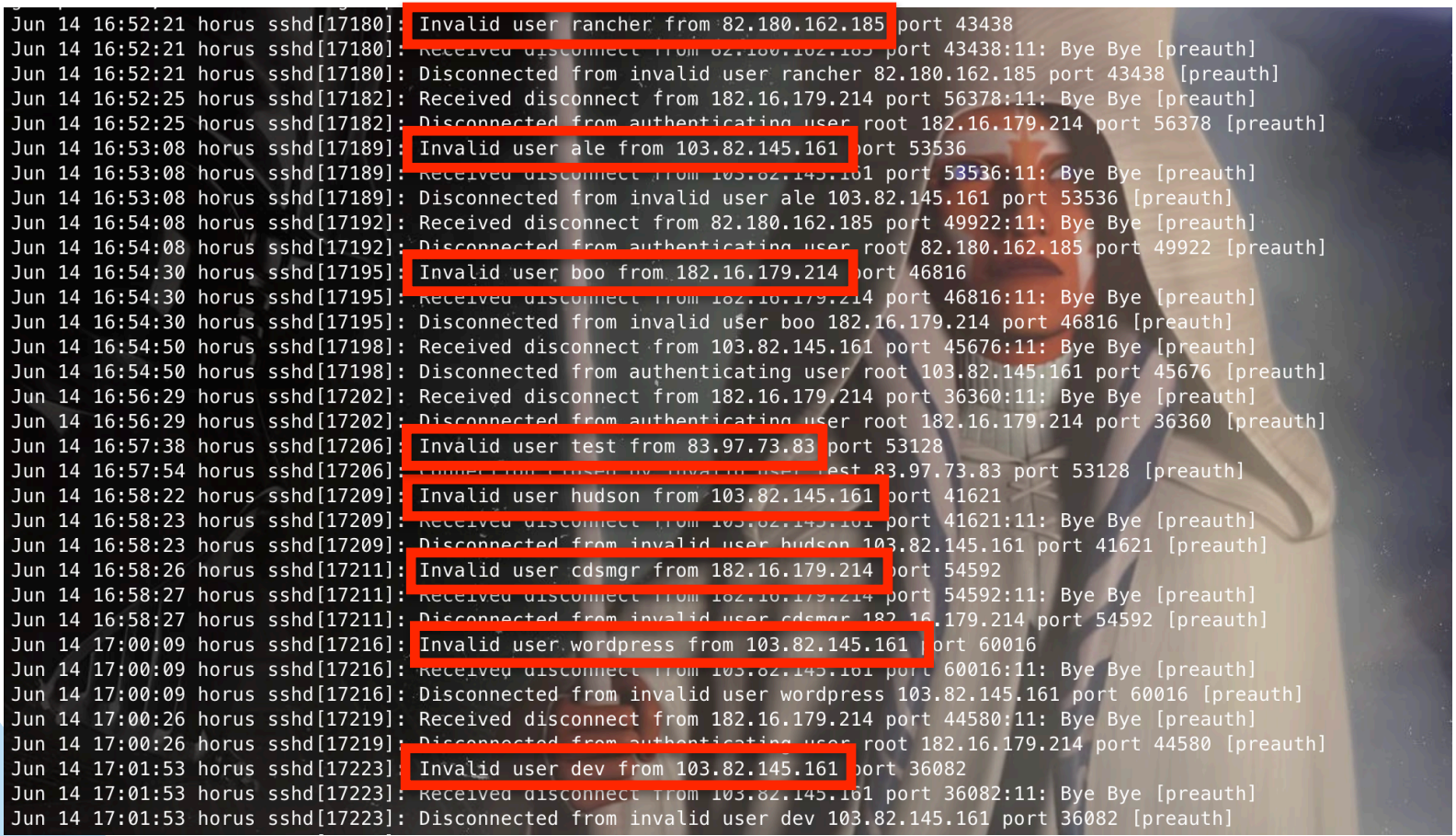


Sécurité anti-spoofing :
… depuis le code directement

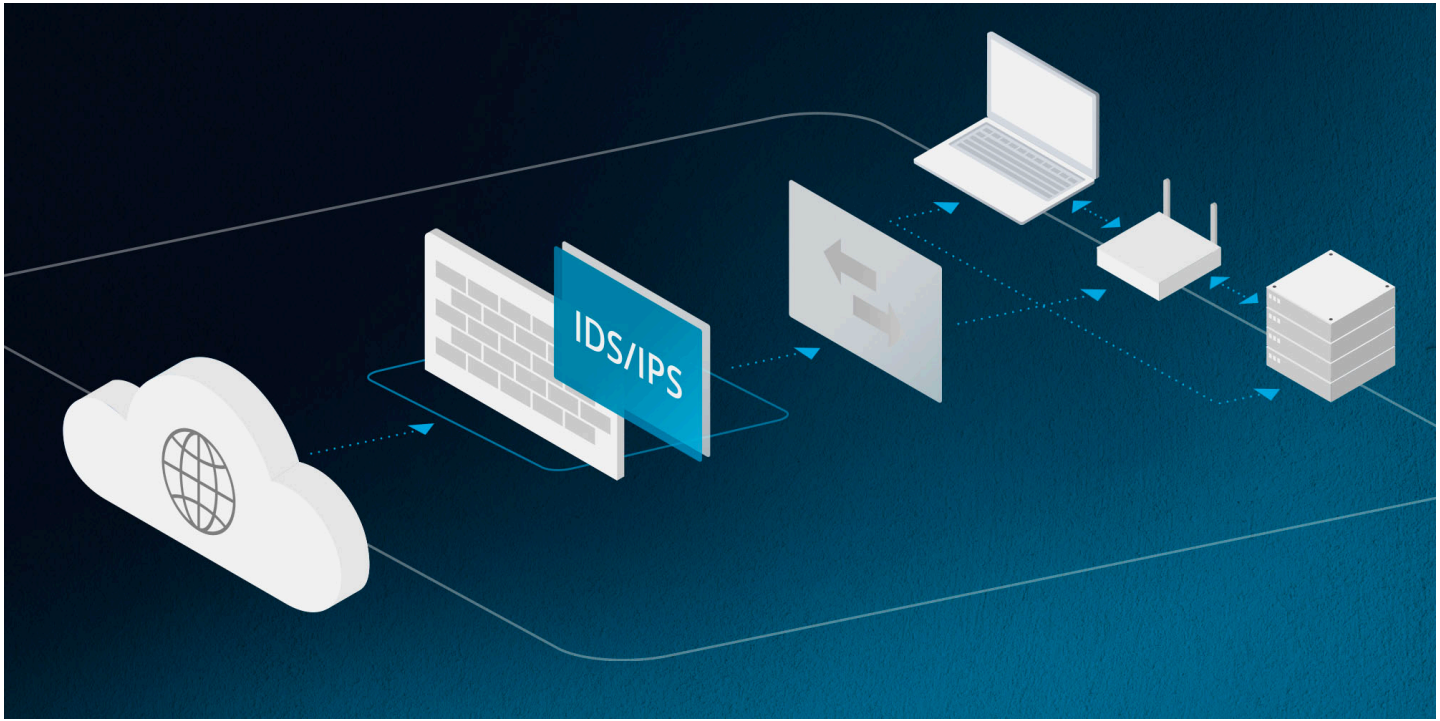
```
350         # order matters
351         proxmox.nodes(node).qemu(vmid).firewall.options.put(enable=1)
352         proxmox.nodes(node).qemu(vmid).firewall.options.put(ipfilter=0)    "ipfilter": Unknown word.
353         proxmox.nodes(node).qemu(vmid).firewall.options.put(policy_in="ACCEPT")
354         proxmox.nodes(node).qemu(vmid).firewall.rules.post(action="DROP", type="out", log="nolog", enable=1) # OUT DROP -log
            nolog
355         proxmox.nodes(node).qemu(vmid).firewall.rules.post(action="DROP", type="in", log="nolog", enable=1) # IN DROP -log nolog
356         proxmox.nodes(node).qemu(vmid).firewall.rules.post(action="ACCEPT", type="in", dest="+hosting", log="nolog", enable=1)
            # IN ACCEPT -dest +hosting -log nolog
357         proxmox.nodes(node).qemu(vmid).firewall.rules.post(action="ACCEPT", type="out", source="+hosting", log="nolog",
            enable=1) # OUT ACCEPT -source +hosting -log nolog
358
```

HOSTING
minet

Brute force : le nmap



Intrusion Detection & Prevention System



Piste de projet : Déployer *suricata* uniquement sur la hosting ?

Security by design

La particularité du design frontend/API

06

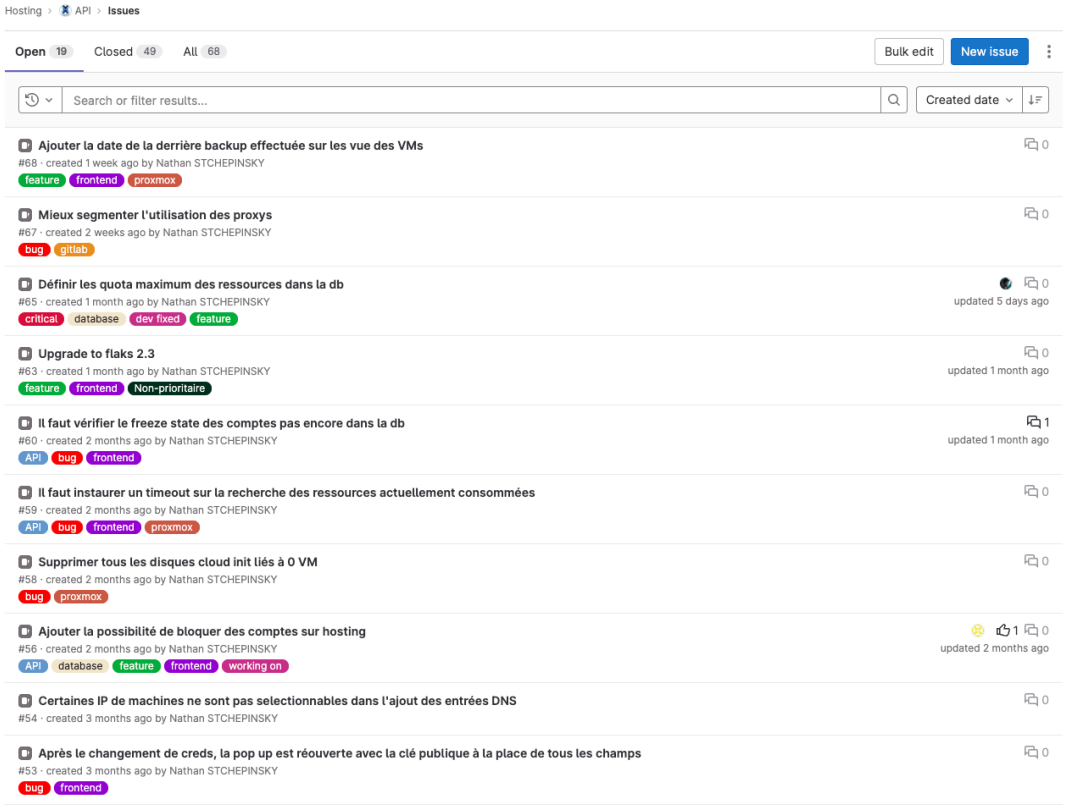
Les projets futurs

C’est CADEAU

Attention aux alternatives

Des milliards de projets encore à faire !

https://gitlabint.priv.minet.net/hosting/api/-/issues



HOSTING

minet


Kahoot!
COUCOU
de Rocamadour
LAMAS-TECH