

Documentation Technique: Virtualisation Middle VM (MiVM)

Racem MOALLA

August 9, 2024

Contents

1	Introduction	2
2	Configuration du Réseau dans OpenStack	2
2.1	Création du Réseau	2
2.2	Connexion des Réseaux via un Routeur	2
2.3	Description de la Topologie Réseau	2
2.4	Connections et Communications	2
2.5	Diagramme de la Topologie Réseau	2
3	Création des VM dans OpenStack	3
4	Installation des Prérequis	3
4.1	Installer Python et les dépendances nécessaires	3
4.2	Fichier <code>requirements.txt</code>	4
5	Description des Composants	4
5.1	Simulateur	4
5.2	Gateway	4
5.3	Forward to Edge	4
5.4	prétraitement	4
5.5	DecisionFinale	4
6	Flux de Données	5
7	Conclusion	5

1 Introduction

Cette documentation technique détaille le processus de création, configuration et implémentation d'une solution hybride utilisant les ressources du Cloud et de l'Edge pour maximiser les performances. La solution repose sur une machine virtuelle intermédiaire (MiVM) qui coordonne les tâches entre le Cloud et l'Edge, assurant ainsi une interaction simplifiée pour les utilisateurs.

2 Configuration du Réseau dans OpenStack

2.1 Création du Réseau

Pour configurer le réseau dans OpenStack, vous devez créer les réseaux suivants :

- **Réseau Public** : Ce réseau est accessible publiquement, probablement l'Internet.
- **Réseau Privé (PrivateN)** : Ce réseau est utilisé pour des communications internes sécurisées.
- **Réseau Edge (EdgeN)** : Ce réseau sert de pont entre le réseau privé et le réseau cloud. Il est connecté au réseau privé via un routeur et contient les noeuds Edge.
- **Réseau Cloud (CloudN)** : Ce réseau héberge les services et ressources dans le cloud.

2.2 Connexion des Réseaux via un Routeur

Pour permettre la communication entre le réseau privé sécurisé et le réseau Edge, vous devez configurer un routeur dans OpenStack. Le routeur assure le routage des paquets entre ces réseaux, permettant ainsi à la MVM de communiquer directement avec EdgeVM.

Cause : L'utilisation d'un routeur permet d'isoler les réseaux pour des raisons de sécurité tout en facilitant la communication entre eux, maximisant ainsi les performances et la sécurité de la solution.

2.3 Description de la Topologie Réseau

- **MiVM (middle virtual machinet)**
 - Située dans le réseau PrivateN.
 - Connectée au réseau EdgeN via le routeur.
- **EdgeVM (Machine Virtuelle Edge)**
 - Située dans le réseau EdgeN.
 - Connectée aux réseaux PrivateN via le routeur et dispose également d'une adresse publique.
- **CloudVM (Machine Virtuelle Cloud)**
 - Située dans le réseau CloudN et dispose d'une adresse publique.

2.4 Connections et Communications

- **MVM < – > EdgeVM**
 - Communique via les adresses IP privées par le routeur.
- **EdgeVM < – > CloudVM**
 - Communique via les adresses IP publiques.

2.5 Diagramme de la Topologie Réseau

Pour une meilleure compréhension, voici un diagramme représentant la topologie réseau et les communications entre les différentes VM :

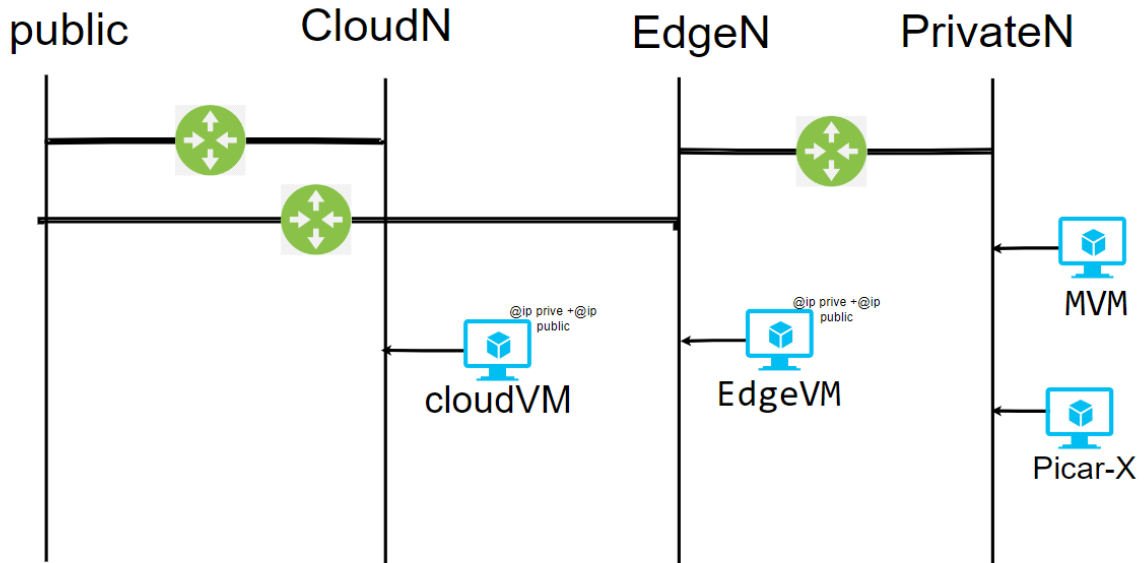


Figure 1: Diagramme de la Topologie Réseau

3 Création des VM dans OpenStack

Après avoir configuré les réseaux, vous pouvez créer les VM nécessaires :

- **MVM (Middle Virtual Machine)**
 - Nom : MiVM
 - Image : Sélectionnez une image appropriée (par exemple, Ubuntu 20.04)
 - Taille : m1.medium
- **EdgeVM**
 - Nom : EdgeVM
 - Image : Sélectionnez une image appropriée (par exemple, Ubuntu 20.04)
 - Taille : m1.medium
- **CloudVM**
 - Nom : CloudVM
 - Image : Sélectionnez une image appropriée (par exemple, Ubuntu 20.04)
 - Taille : m1.large

4 Installation des Prérequis

Pour installer les prérequis nécessaires au fonctionnement de cette solution, suivez ces étapes :

4.1 Installer Python et les dépendances nécessaires

Connectez-vous à chaque VM (MVM, EdgeVM, CloudVM, Picar-X) et exécutez les commandes suivantes :

```
sudo apt install -y python3 python3-pip
pip3 install --no-cache-dir --break-system-packages -r requirements.txt
```

4.2 Fichier requirements.txt

Ce fichier contient toutes les dépendances nécessaires à l'exécution des scripts. Assurez-vous de le placer dans le répertoire de chaque VM avant d'exécuter les commandes ci-dessus.

5 Description des Composants

5.1 Simulateur

Le simulateur génère des paquets de données et les envoie à la Gateway.

Chemin du Code : `Picar-X/simulateur_de_donnees.py`

Fonctionnement :

- Génère des données de simulation représentant les capteurs de la voiture.
- Envoie ces données à la Gateway .

5.2 Gateway

La Gateway reçoit les données des véhicules et les transmet à la VM intermédiaire.

Chemin du Code : `Picar-X/gateway.py`

Fonctionnement :

- Reçoit des paquets de données des véhicules connectés.
- Transmet ces données à la MVM via une requête HTTP POST.

5.3 Forward to Edge

Le service Forward to Edge reçoit les données de la Gateway et les transmet à EdgeVM pour prétraitement.

Chemin du Code : `Code_du_MiVM/forwordToEdge.py`

Fonctionnement :

- Reçoit les données de la Gateway.
- Transmet ces données à EdgeVM pour prétraitement via une requête HTTP POST.

5.4 prétraitement

Le service Analyse sur EdgeVM effectue le prétraitement des données avant de les envoyer à CloudVM.

Chemin du Code : `Code_du_EdgeVM/prétraitement.py`

Fonctionnement :

- Effectue le prétraitement des données reçues.
- Envoie les données prétraitées à CloudVM pour un traitement intensif.

5.5 DecisionFinale

Le service Decision sur CloudVM effectue le traitement intensif et renvoie les résultats à la Gateway.

Chemin du Code : `Code_du_CloudVM/decisionFinale.py`

Fonctionnement :

- Reçoit les données prétraitées d'EdgeVM.
- Effectue un traitement intensif des données.
- Renvoie les résultats à la Gateway pour action.

6 Flux de Données

Le flux de données dans cette solution se déroule comme suit :

1. Les véhicules envoient des données à la Gateway sur Picar-X.
2. La Gateway transmet les données à la MVM.
3. La MVM transmet les données à EdgeVM pour prétraitement.
4. EdgeVM effectue le prétraitement et envoie les données à CloudVM pour un traitement intensif.
5. CloudVM traite intensivement les données et renvoie les résultats à la voiture.

Ce flux de données garantit que les tâches nécessitant une faible latence sont traitées près de la source des données (Edge), tandis que les tâches nécessitant des ressources importantes sont traitées dans le Cloud.

7 Conclusion

Cette documentation technique a décrit la création, la configuration et l'implémentation d'une solution hybride Cloud-Edge utilisant des VM dans OpenStack. En virtualisant les ressources des environnements Cloud et Edge via une MiVM intermédiaire, nous avons pu optimiser les performances et simplifier l'interaction utilisateur. La solution comprend des services de prétraitement et de traitement intensif, assurant une gestion efficace des données et des ressources.

Pour toute question ou suggestion, n'hésitez pas à me contacter.