
libvirtd.conf

Fichier de configuration pour libvirtd

Options

listen_tls à 0, écoute les connexions non chiffrées, à 1, écoute les connexions chiffrées

tls_port Port d'écoute pour les connexions tcp sécurisés.

tcp_port Port d'écoute pour les connexions tcp non-sécurisés.

listen_addr Adresse d'écoute pour les connexions tcp

mdns_adv à 1, active l'avertissement mDNS des services client.

mdns_name par défaut "Virtualization Host HOSTNAME", où HOSTNAME est substitué par le nom court de la machine.

unix_sock_group Groupe propriétaire du socket

unix_sock_ro_perms Permissions du socket RO unix, utilisé pour la supervision des VM

unix_sock_rw_perms Permissions du socket RW unix, utilisé pour la gestion complète des VM.

unix_sock_dir Répertoire où créer/trouver le socket.

auth_unix_ro Définis le schéma d'authentification pour les socket RO unix (none,sasl,polkit)

auth_unix_rw Définis le schéma d'authentification pour les socket RW unix (none,sasl,polkit)

auth_tcp Définis le schéma d'authentification pour les sockets TCP (none,sasl,polkit)

auth_tls Définis le schéma d'authentification pour les sockets TCP sécurisés (none,sasl,polkit)

access_drivers = ["nop" | "polkit"] Par défaut un utilisateur authentifié est autorisé à accéder a toutes les API. Les pilotes d'accès peuvent y placer des restrictions. Par défaut, le driver 'nop' est activé, signifiant qu'aucun contrôle d'accès n'est fait une fois le client authentifié. Sinon, 'polkit' peut être utilisé

key_file Clé privée du service

cert_file Chemin des fichiers de certificat du serveur

ca_file Certificat de CA à utiliser

crl_file CRL à utiliser

tls_no_sanity_certificate À 1, désactive la vérification de son certificat

tls_no_verify_certificate À 1, désactive la vérification des certificats client

tls_allowed_dn_list = ["DN1", "DN2"] Une liste blanche de DN X.509. Si c'est une liste vide, personne ne peut se connecter.

sasl_allowed_username_list = ["joe@EXAMPLE.COM", "fred@EXAMPLE.COM"] Une liste blanche de noms d'utilisateurs sasl autorisé. Si c'est une liste vide, personne ne peut se connecter.

max_clients = 5000 Nombre de clients simultanés autorisés

max_queued_clients = 1000 Longueur de la file d'attente de connexion autorisés

max_anonymous_clients = 20 Longueur de la file d'attente de clients acceptés mais pas encore authentifiés

min_workers = 5 Limite minimum de clients prévu, au dessus de cette limite, d'autres threads sont créés.

max_workers = 20 Limite maximum. Généralement max_workers == max_clients

prio_workers Nombre de clients prioritaires. Si tous les clients du pool de workers sont bloqués, certains appels marqués en haute priorité peuvent être exécutés dans ce pool.

max_requests = 20 Limite totale d'appels RPC concurrents. Devrait être au moins aussi grand que max_workers. Au delà de cette limite, les requêtes RPC seront lues en mémoire et mises en file d'attente. Cela impacte l'utilisation de la mémoire, chaque requête nécessitant 256Ko de mémoire.

max_client_requests = 5 Limite de requête concurrentes par client. Pour éviter qu'un client monopolise le serveur, devrait être une petite fraction de max_requests.

log_level = 3 Niveau de logs (4 errors, 3 warnings, 2 information, 1 debug)

log_filters="3 :remote 4 :event" Filtre le logs. Permet de sélectionner différents niveaux de log pour une catégorie de logs donnée. Plusieurs filtres peuvent être définis dans un simple @filters Le format est :

x :name

x :+name où name est une chaîne qui est matchée avec un nom de fichier source (ex : "remote", "qemu", ou "util/json") "+" indique à libvirt de logger les stack trace pour chaque message, et x est le niveau minimum où les messages matchant sont loggés :

1 : DEBUG

2 : INFO

3 : WARNING

4 : ERROR

log_outputs="3 :syslog :libvirtd" Une sortie est un emplacement où sauver les informations de log. plusieurs sorties peuvent être définis. Le format est :

x :stderr Erreur standard

x :syslog :name Utilise syslog

x :file :file_path Sort dans un fichier

x :journald Utilise journald

x est le niveau minimum, agissant comme filtre :

1 :DEBUG

2 :INFO

3 :WARNING

4 :ERROR

audit_level = 2 Permet d'altérer le sous-système d'audit :

0 Désactive l'audit

1 Active l'audit, seulement si permis par l'hôte

2 Active l'audit, et qui si désactivé par l'hôte

audit_logging = 1 À 1, les messages d'audit seront envoyés via l'infrastructure de logging libvirt. Défaut : 0

host_uuid = "00000000-0000-0000-0000-000000000000" Fournis l'UUID de l'hôte ici dans le cas où la commande 'dmidecode -s system-uuid' ne fournit pas d'uuid valide.

keepalive_interval = 5 Permet à libvirtd de détecter les connexions client cassé ou morts. Un keepalive est envoyé à un client après cet interval (en seconde) d'inactivité pour vérifier s'il répond.

keepalive_count = 5 Nombre maximum de message keepalive sans réponses à envoyer avant de fermer automatiquement la connexion.

keepalive_required = 1 À 1, libvirtd refuse de parler aux clients qui ne supportent pas le protocole keepalive. Défaut : 0.