corosync.conf

Fichier de configuration de corosync

Description

Le fichier de configuration consiste de directives en accolades. Les choix possibles sont :

totem {} Contient les options de configuration pour le protocole totem

logging {} Contient les options de configuration pour le logging

quorum {} Contient les options de configuration pour le quorum

nodelist {} Contient les options de configuration pour les nœuds dans le cluster

qb {} Contient les options de configuration liées à libqb

Options totem

ringnumber Spécifie le numéro pour l'interface. En utilisant le protocole ring redondant, chaque interface devrait avoir un numéro ring séparé pour identifier les membres du protocole que l'interface utilise pour le ring. doit commencer à 0.

bindnetaddr Spécifie l'adresse réseaux auquel corosync devrait se lier. Doit être une IP dans le système, ou un réseau.

broadcast Optionnel et peut être mis à yes. Permet d'utiliser le broadcast pour les communications.

mcastaddr Adresse de multicast pour corosync. Le défaut devrait fonctionner sur la plupart des réseaux. Éviter 224.x.x.x parce que c'est une adresse multicast de config.

mcastport Port UDP pour le multicast.

ttl Si le cluster fonctionne sur un réseau routé, un TTL de 1 sera trop petit.

version Spécifie la version du fichier de configuration. Actuellement la seule version valide est 2.

clear_node_high_bit Optionnel et est seulement utile quand aucun nodeid n'est spécifié. Certains client corosync nécessitent un nodeid 32-bits signé supérieur à 0, cependant corosync utilise tous les 32-bits de l'espace d'adressage IPv4 en générant un nodeid. À yes, cette options force le MSB à 0.

crypto_hash Spécifie l'algorithme d'authentification HMAC utilisé pour authentifier tous les messages. Les valeurs valides sont : none, md5, sha1, sha256, sha384 et sha512.

crypto_cipher Spécifie l'algorithme de chiffrement à utiliser pour chiffrer les messages. Les valeurs valides sont : none, aes256, aes192, aes128, et 3des.

rrp_mode Spécifie le mode de ring redondant, qui peut être : none, active, ou passive. La réplication active offre les latences les plus faibles mais est moins performant. La réplication passive peut presque doubler la vitesse du protocole totem. none spécifie que seul l'interface réseau sera utilisé pour opérer le protocole totem.

netmtu Spécifie l'unité de transmission maximum réseaux. Pour définir cette valeur au-delà de 1500, le support des jumbo frames est nécessaire sur tous les nœuds.

transport Contrôle le mécanisme de transport utilisé. Si l'interface auquel corosync est lié est une interface RDMA, le paramètre "iba" peut être spécifié. Pour éviter l'utilisation du multicast, un paramètre de transport unicast "udpu" peut être spécifié. Cela nécessite de spécifier la liste des membres dans la directive nodelist. Défaut : udp.

cluster_name Spécifie le nom du cluster et son utilisation pour la génération automatique d'adresse multicast.

config_version Spécifie la version du fichier de configuration. C'est convertit en entier non-signé 64-bits. Par défaut : 0. Permet d'éviter de joindre d'anciens nœud sans configuration à jours.

ip_version Spécifie la version d'IP à utiliser pour les communications. ipv4 ou ipv6.

token Ce timeout spécifié en millisecondes le délai au delà duquel une perte de token est déclaré. C'est le temps passé à détecter une panne d'un processeur dans la configuration courante. Réformer une nouvelle configuration prend 50 ms en plus de ce timeout.

token_coefficient Utilisé seulement quand la section **nodelist** est spécifiée et contient au moins 3 nœuds. Le vrai timeout de token est calculé par **token + (number_of_nodes -2) * token_coefficient**. Cela permet au cluster de s'agrandir sans changer manuellement le timeoute de token à chaque fois qu'un nœud est ajouté. Défaut : 650ms

token_retransmit Ce timeout spécifie, en ms, après combien de temps avant de recevoir un token le token est retransmis. Ce sera automatiquement calculé si le token est modifié. Il n'est pas recommandé d'altérer cette valeur. Défaut : 238ms.

hold Ce timeout spécifie, en ms, combien de temps le token devrait être maintenu quand le protocole est en sous-utilisation. Il n'est pas recommandé d'altérer ce paramètre. Défaut : 180ms

token_retransmits_before_loss_const Cette valeur identifie combien de token retransmis devraient être tentés avnt de former une nouvelle configuration. Si cette valeur est définie, la retransmission et le maintient sera automatiquement calculé depuis retransmits_before_loss et token. Défaut : 4 retransmissions.

join Ce timeout spécifie, en ms, combien de temps attendre les messages join dans le protocole membership.

send_join Ce timeout spécifie, en ms, une plage entre 0 et send_join d'attente avant d'envoyer un message join. Pour les configuration inférieur à 32 nœuds, ce paramètre n'est pas nécessaire. Pour les ring plus grands, ce paramètre est nécessaire pour s'assurer que l'interface n'est pas surchargée avec des messages join. Une valeur raisonnable pour les grands ring (128 nœuds) est de 80ms. Défaut : 0

consensus Ce timeout spécifie, en ms, combien de temps attendre pour que le consensus soit achevé avant de commencer un nouveau tour de configuration de membership. a valeur minimum pour le consensus doit être 1.2 * token. Ce calcul est automatique si cette option n'est pas spécifiée

merge Ce timeout spécifie, en ms, combien de temps attendre avant de vérifier une partition quand aucun trafique multicast n'a été envoyé. Si un trafique multicast est envoyé, la detection du merge se produit automatiquement en tant que fonction du protocole. Défaut : 200ms

downcheck Ce timeout spécifie, en ms, combien de temps attendre avant de vérifie qu'une interface réseaux est de nouveau disponible après une indisponibilité. Défaut : 1000ms

fail_recv_constf Cette constante spécifie combien de rotations du token sans recevoir de messages quand les message devaient être reçus peuvent se produire avant qu'une nouvelle configuration ne soit formée. Défaut : 2500 échecs de réception de message.

seqno_unchanged_const Cette constante spécifie combient de rotation du token sans trafique multicast devrait se produire avant que le timer hold soit démarré. Défaut : 30 rotations.

heartbeat_failures_allowed Mécanisme HeartBeating. Configure le mécanisme HeartBeating pour une détection d'erreur plus rapide. Garder en mémoire qu'engager ce mécanisme dans un réseau avec beaucoup de perte peut causer les déclaration de fausse perte vu que le mécanisme ne se base que sur le réseau. Défaut : 0 (désactivé)

max_network_delay Mécanisme HeartBeating. Cette constante spécifie, en ms, le délai approximatif que le réseau prend pour transporter un packet d'une machine à une autre. Cette valeur doit être définie par les ingénieurs système. Défaut : 50ms

window_size Cette constante spécifie le nombre de messages qui peuvent être envoyés en une rotation de token. Si tous les processeurs traitent de manière égale, cette valeur peut être grande (300). pour réduire la latence dans les grand ring (>=16), le défaut est un compromis de sûreté. Si un ou plusieurs processeur sont lents, window_size ne devrait pas être supérieur à 256000 / netmtu pour éviter la surchage des tampon. Défaut : 50 messages.

max_messages Cette constante spécifie le nombre maximum de messages qui peuvent être envoyés par un processeur à la réception du token. Ce paramètre est limité à 256000 / netmtu pour empêcher la saturation des tampons de transmission.

miss_count_const Définie le nombre maximum de fois à la réception d'un token qu'un message est vérifié pour retransmission avant qu'une retransmission se produise. Défaut : 5 messages.

rrp_problem_count_timeout Spécifie le temps en ms d'attente avant de décrémenter le compteur de problème de 1 pour un ring particulier pour s'assurer qu'un lien n'est pas marqué en panne. (Défaut : 2000 ms)

rrp_problem_count_threshold Spécifie le nombre de fois qu'un problème est détecté avec un lien avec de définir ce lien en faute. Une fois en faute, plus aucune donnée n'est émise sur ce lien. Également, le compteur de problème n'est plus décrémenté quand le timeout de compteur de problème expire. Un problème est détecté quand tous les tokens d'un processeur n'a pas été reçus dans le rrp_token_expired_timeout. Les rrp_problem_count_threshold * rrp_token_expired_timeout devrait être au moins à 50ms inférieur au timeout du token, ou une reconfiguration complète peut se produire. Défaut : 10 problèmes.

rrp_problem_count_mcast_threshold Spécifie le nombre de fois qu'un problème est détecté avec un multicast avant de définis le lien en faute pour un mode rrp passif. Cette variable n'est pas utilisée pour un mode rrp actif.

rrp_token_expired_timeout Spécifie le temps en ms pour incrémenter le compteur de problème pour le rrp après avoir reçu un token de tous les rings pour un processeur particulier. Cette valeur est automatiquement calculée du timeout de token et problem_count_threshold mais peut être écrasé. Défaut : 47ms

rrp_autorecovery_check_timeout Spécifie le temps en ms pour vérifier si le ring en panne peut être auto-récupéré.

Options logging

timestamp Spécifie si un horodatage est placé dans les messages de log. Défaut : off

fileline spécifique que le fichier et la ligne devrait être affichés. Défaut : off

function_name Spécifie que le nom de la fonction devrait être affiché. Défaut : off

to_stderr

to_logfile

to_syslog spécifient la sortie de logging. peut être yes ou no. Défaut : syslog et stderr

logfile si to_logfile est à yes, cette option spécifie le chemin du fichier de log.

logfile_priority Priorité du logfile pour un sous-système particulier. Ignoré si debug est on. (alert,crit, debug, emerg, err, info, notice, warning) Défaut : info.

syslog_facility Type de facilité syslog (daemon, local0, local1, local2, local3, local4, local5, local6 et local7). Défaut : daemon

syslog_priority Log level pour le sous-système particulier (alert, crit, debug, emerg, err, info, notice, warning) Défaut : info

debug Active le debug. Défaut : off.

Options logger_subsys

Toutes les options de la directive logging sont valide, plus :

subsys Spécifie le nom du sous-système pour lequel le logging est spécifié. C'est le nom utilisé par un service dans l'appel log_init. Cette directive est requise.

Options quorum

provider seul corosync_votequorum est supporté.

Options nodelist

ringX_addr Spécifie l'adresse IP d'un des nœuds. X est le numéro du ring.

nodeid Requis pour IPv6, optionnel pour IPv4. valeur 32bits spécifiant l'identifiant du nœud délivré au service de cluster membership. 0 est réservé et ne doit pas être utilisé.

Options qb

ipc_type Type d'IPC à utiliser. Peut être native, shm et socket. Native signifie soit shm soit socket, en fonction de ce qui est supporté par l'OS. shm est généralement plus rapide, mais nécessite d'allouer un tampon ring dans /dev/shm.