Création d'un cluster OpenHPC sur Raspberry

Table of Contents

:

OpenHPC est un projet collaboratif de la Fondation Linux dont la mission est de fournir une collection de référence de composants logiciels HPC open source et de meilleures pratiques, abaissant les obstacles au déploiement, à l'avancement et à l'utilisation des méthodes et outils HPC modernes.

Installer le système d'exploitation de base

Télécharger l'image de la distribution sur https://people.centos.org/pgreco/CentOS-Userland-8-aarch64-RaspberryPI-Minimal-4/CentOS-Userland-8-aarch64-RaspberryPI-Minimal-4-sda.raw.xz

Graver l'image sur la carte SD en utilisant l'une des méthodes décrites ici.

Insérer la carte SD dans le Raspberry PI, allumer le Raspberry PI puis se connecter avec l'utilisateur par défaut :

```
utilisateur : root
```

```
    mot de passe : centos
```

Mettre le système d'exploitation à jour. Depuis le terminal :

```
yum -y update
yum -y upgrade
```

Configurer le Wi-Fi

Avant de commencer le processus d'installation des composants **OpenHPC**, plusieurs considérations supplémentaires doivent être intégrées dans la configuration de l'hôte.

Premièrement, la recette d'installation ici suppose que le nom d'hôte peut être résolu localement.

```
[sms]# echo ${sms_ip} ${sms_name} >> /etc/hosts
```

Bien qu'il soit théoriquement possible d'activer SELinux sur un cluster provisionné avec Warewulf, même l'utilisation du mode permissif peut être problématique et il est recommandé de désactiver SELinux sur l'hôte SMS maître. Si les composants SELinux sont installés localement, la commande `selinuxenabled` peut être utilisée pour déterminer si SELinux est actuellement activé. Si elle est activé, il faut désactiver.

Enfin, les services d'approvisionnement reposent sur les protocoles réseau DHCP, TFTP et HTTP. En fonction de la configuration locale sur l'hôte SMS, les règles de pare-feu par défaut peuvent interdire ces services. Par conséquent, le pare-feu local exécuté sur l'hôte doit être désactivé. S'il est installé, la valeur par défaut le service de pare-feu peut être désactivé comme suit :

```
[sms]# systemctl disable firewalld
[sms]# systemctl stop firewalld
```

Installer les composants OpenHPC

Une fois le système de base installé et démarré, l'étape suivante consiste à ajouter les packages OpenHPC souhaités sur le serveur maître afin de fournir des services d'approvisionnement et de gestion des ressources pour le reste du cluster. Less sous-sections suivantes mettent en évidence ce processus.

Activer le référentiel OpenHPC pour une utilisation locale

Pour commencer, activer l'utilisation du référentiel **OpenHPC** en l'ajoutant à la liste locale des référentiels de packages disponibles.

Cela nécessite un accès réseau du serveur maître au référentiel **OpenHPC**, ou alternativement, que le référentiel **OpenHPC** soit mis en miroir localement. Dans les cas où une connectivité réseau externe est disponible, **OpenHPC** fournit un package ohpc-release qui inclut des clés GPG pour la signature du package et l'activation du dépôt. L'exemple qui suit illustre l'installation du package ohp-

release directement à partir du serveur de construction **OpenHPC**.

```
yum install
http://repos.openhpc.community/OpenHPC/2/CentOS_8/aarch64/ohpc-release-2-1.el8.aarch64.rpm
```

Il peut être utile ou nécessaire d'avoir une copie locale des référentiels OpenHPC. Pour faciliter ce besoin, des archives tar autonomes sont fournies - une contenant un référentiel de packages binaires ainsi que tout les mises à jour disponibles, et un contenant un référentiel de RPMS source. Les fichiers tar contiennent également un simple script bash pour configurer le gestionnaire de packages pour utiliser le référentiel local après le téléchargement. Pour l'utiliser, il suffit de déballer l'archive tar où on souhaite héberger le référentiel local et exécuter le script make repo.sh. Les fichiers tar pour cette version peut être trouvé à http://repos.openhpc.community/dist/2.3

En plus du référentiel de packages OpenHPC, l'hôte maître nécessite également un accès à la base standard dépôts de distribution du système d'exploitation afin de résoudre les dépendances nécessaires. Pour CentOS8.4, les exigences sont d'avoir accès aux référentiels BaseOS, Appstream, Extras, PowerTools et EPEL pour lesquels des miroirs sont disponibles gratuitement en ligne par exemple http://mirror.centos.org/centos-8/8/

Cela dépend du référentiel CentOS Extras, qui est livré avec CentOS et est généralement activé par défaut. En revanche, le référentiel CentOS PowerTools est généralement désactivé dans une installation standard,mais peut être activé comme suit :

```
yum install dnf-plugins-core
yum config-manager --set-enabled powertools
```

La collection d'instructions de ligne de commande qui suivent dans ce guide, lorsqu'elles sont combinées avec des entrées de site local,peut être utilisé pour mettre en œuvre une installation et une configuration de système sans système d'exploitation. Le format de ces commandes est destiné à être utilisable par couper-coller direct (avec substitution de variable pour les paramètres spécifiques au site).
Alternativement, le package de documentation OpenHPC (docs-ohpc) inclut un script de modèle qui inclut un résumé de toutes les commandes utilisées ici. Ce script peut être utilisé en conjonction avec un simple fichier texte pour définir les variables locales.

Ajouter des services de provisionnement sur le nœud maître

Avec le référentiel **OpenHPC** activé, on peut maintenant commencer à ajouter les composants souhaités sur le serveur maître.

Ce référentiel fournit un certain nombre d'alias qui regroupent les composants logiques afin d'aider dans ce processus. Pour ajouter la prise en charge des services de provisionnement, les commandes

suivantes illustrent l'ajout d'un package de base commun suivi du système d'approvisionnement Warewulf.

```
# Installer les méta-paquets de base
yum -y install ohpc-base
yum -y install ohpc-warewulf
```

De nombreuses configurations BIOS de serveur ont un démarrage réseau PXE configuré comme option principale dans le démarrage par défaut. Si les nœuds de calcul ont un périphérique différent du premier de la séquence, l'ipmitool peut être utilisé pour activer PXE.
`ipmitool -E -I lanplus -H ${bmc_ipaddr} -U root chassis bootdev pxe options=persistent`

Les systèmes HPC reposent sur des horloges synchronisées dans tout le système et le protocole NTP peut être utilisé pour faciliter cette synchronisation. Pour activer les services NTP sur l'hôte SMS avec un serveur spécifique ${ntp server},et autoriser ce serveur à servir de serveur de temps local pour le cluster, exécuter ce qui suit :

```
systemctl enable chronyd.service
echo "server ${ntp_server}" >> /etc/chrony.conf
echo "allow all" >> /etc/chrony.conf
systemctl restart chronyd
```

L'option “allow all” spécifiée pour le démon chrony time autorise tous les serveurs du réseau local à pouvoir se synchroniser avec l'hôte SMS. Alternativement, on peut restreindre l'accès à des plages d'adresses IP fixes et à un sous-réseau local de classe B en utilisant la commande suivante:`allow 192.168.0.0/16`

Ajouter des services de gestion des ressources sur le nœud maître

OpenHPC fournit plusieurs options pour la gestion des ressources distribuées. La commande suivante ajoute le les Slurm composants au serveur du gestionnaire de charge de travail de l'hôte maître choisi(les composants côté client seront ajoutés à l'image de calcul correspondante dans une étape ultérieure)

```
# Installer le méta-paquet du serveur slurm
yum -y install ohpc-slurm-server
# Utiliser le fichier fourni par ohpc pour démarrer la configuration SLURM
cp /etc/slurm/slurm.conf.ohpc /etc/slurm/slurm.conf
# Identifier le nom d'hôte du gestionnaire de ressources sur l'hôte maître
```

```
perl -pi -e "s/ControlMachine=\S+/ControlMachine=${sms_name}/"
/etc/slurm/slurm.conf
```

Il existe une grande variété d'options de configuration et de plugins disponibles pour Slurm et l'exemple de configuration illustré ci-dessus cible une installation assez basique. En particulier, les données d'achèvement des travaux seront stockées dans un fichier texte (/var/log/slurm jobcomp.log) qui peut être utilisé pour enregistrer des informations comptables simples.

SLURM requiert l'énumération des caractéristiques matérielles physiques pour les nœuds de calcul sous son contrôle.

En particulier, trois paramètres de configuration se combinent pour définir des ressources de calcul consommables : **Sockets**, **CoresPerSocket** et **ThreadsPerCore**. Le fichier de configuration par défaut fourni via OpenHPC suppose, 8 cœurs par socket et deux threads par cœur pour cet exemple à 4 nœuds. Si cela ne reflète pas la configuration matérielle mise en oeuvre, mettre à jour le fichier de configuration dans /etc/slurm/slurm.conf en conséquence pour qu'il corresponde au matériel particulier.

Le projet SLURM fournit un outil de configuration en ligne facile à utiliser qui peut être consulté ici.

Terminer la configuration de base de Warewulf pour le nœud maître

À ce stade, tous les packages nécessaires pour utiliser Warewulf sur l'hôte principal doivent être installés.

Il faut mettre à jour plusieurs fichiers de configuration afin de permettre à Warewulf de fonctionner avec CentOS8.4 et de prendre en charge l'approvisionnement local à l'aide d'une seconde interface privée.

Par défaut, Warewulf est configuré pour provisionner sur l'interface eth1, les étapes ci-dessous décrivent comment le remplacer par une interface potentiellement nommée autrement spécifiée par ${sms eth internal}.

```
# Configurer le provisionnement Warewulf pour utiliser l'interface interne
souhaitée
 perl -pi -e "s/device = eth1/device = ${sms_eth_internal}/"
/etc/warewulf/provision.conf

# Activer l'interface interne pour le provisionnement
ip link set dev ${sms_eth_internal} up
ip address add ${sms_ip}/${internal_netmask} broadcast + dev
${sms_eth_internal}

# Redémarrer/activer
[sms]# systemctl enable httpd.service
```

```
[sms]# systemctl restart httpd
[sms]# systemctl enable dhcpd.service
[sms]# systemctl enable tftp.socket
[sms]# systemctl start tftp.socket
```

Définir l'image de calcul pour le provisionnement

Une fois les services d'approvisionnement activés, l'étape suivante consiste à définir et à personnaliser une image système qui peut être ensuite utilisé pour provisionner un ou plusieurs nœuds de calcul.

Construire l'image initiale

La version OpenHPC de Warewulf inclut des améliorations spécifiques permettant la prise en charge de CentOS8.4.

Les étapes suivantes illustrent le processus de création d'une image par défaut minimale à utiliser avec Warewulf. On commence en définissant une structure de répertoire sur l'hôte maître qui représentera le système de fichiers racine du calculateur nœud. L'emplacement par défaut de cet exemple est dans /opt/ohpc/admin/images/centos8.4.

Warewulf est configuré par défaut pour accéder à un référentiel externe (mirror.centos.org) pendant le wwmkchroot traiter. Si l'hôte maître ne peut pas atteindre les miroirs CentOS publics, ou si on préfère utiliser un miroir mis en cache, définir la variable d'environnement ${YUM MIRROR} sur l'emplacement de dépôt souhaité avant d'exécuter la commande wwmkchroot ci-dessous. Par exemple:

```
# Remplacer le référentiel du système d'exploitation par défaut (facultatif)
- définir la variable YUM_MIRROR sur l'emplacement du référentiel souhaité
export YUM_MIRROR=${BOS_MIRROR}

# Définir l'emplacement du chroot
export CHROOT=/opt/ohpc/admin/images/centos8.4

# Construire l'image chroot initiale
wwmkchroot -v centos-8 $CHROOT

# Activer les référentiels OpenHPC et EPEL dans chroot
dnf -y --installroot $CHROOT install epel-release
cp -p /etc/yum.repos.d/OpenHPC*.repo $CHROOT/etc/yum.repos.d
```

Ajouter des composants OpenHPC

Le processus wwmkchroot utilisé à l'étape précédente est conçu pour fournir une configuration CentOS8.4 minimale.

Ensuite, ajouter des composants supplémentaires pour inclure les services client de gestion des ressources, la prise en charge NTP et d'autres packages supplémentaires pour prendre en charge l'environnement OpenHPC par défaut. Ce processus augmente l'installation basée sur wwmkchroot pour modifier l'image de provisionnement de base et les référentiels OpenHPC pour résoudre les demandes d'installation de packages. On commence par installer quelques paquets communs:

```
# Installer le méta-paquet de base du nœud de calcul
yum -y --installroot=$CHROOT install ohpc-base-compute
```

Pour accéder aux référentiels distants par nom d'hôte (et non par adresses IP), l'environnement chroot doit être mis à jour pour activer la résolution DNS. En supposant que l'hôte maître dispose d'une configuration DNS fonctionnelle, l'environnement chroot peut être mis à jour avec une copie de la configuration comme suit :

```
cp -p /etc/resolv.conf $CHROOT/etc/resolv.conf
```

Maintenant, on peut inclure des composants requis supplémentaires à l'instance de calcul, y compris le gestionnaire de ressources de prise en charge des modules client, NTP et environnement de développement.

```
# copier les fichiers d'informations d'identification dans $CHROOT pour
garantir la cohérence
# des uid/gids pour slurm/munge. Celles-ci seront synchronisées avec les
futures mises à jour
# via le système d'approvisionnement.
cp /etc/passwd /etc/group $CHROOT/etc

# Ajouter un méta-paquet de support client Slurm et activer munge
yum -y --installroot=$CHROOT install ohpc-slurm-client
chroot $CHROOT systemctl enable munge

# Enregistrer le serveur Slurm avec les calculs (en utilisant l'option "sans
configuration")
 echo SLURMD_OPTIONS="--conf-server ${sms_ip}" >
$CHROOT/etc/sysconfig/slurmd

# Ajout de la prise en charge du protocole NTP (Network Time Protocol)
yum -y --installroot=$CHROOT install chrony

# Identifier l'hôte maître en tant que serveur NTP local
echo "server ${sms_ip}" >> $CHROOT/etc/chrony.conf

# Ajouter des pilotes de noyau (correspondant à la version du noyau sur le
nœud SMS)
yum -y --installroot=$CHROOT install kernel-`uname -r`

# Inclure l'environnement utilisateur des modules
yum -y --installroot=$CHROOT install lmod-ohpc
```

Personnaliser la configuration du système

Avant d'assembler l'image, il faut effectuer toute personnalisation supplémentaire dans l'environnement chroot créé pour l'instance de calcul souhaitée. Les étapes suivantes documentent le processus d'ajout d'une clé ssh locale créée par Warewulf pour prendre en charge l'accès à distance et permettre le montage NFS d'un système de fichier $HOME et le chemin d'installation public d'OpenHPC (/opt/ohpc/pub) qui sera hébergé par l'hôte maître dans cet exemple de configuration.

```
# Initialiser la base de données warewulf et ssh_keys
base de données wwinit
wwinit ssh_keys

# Ajouter les montages client NFS de /home et /opt/ohpc/pub à l'image de
base
echo "${sms_ip}:/home /home nfs nfsvers=3,nodev,nosuid 0 0" >>
$CHROOT/etc/fstab
echo "${sms_ip}:/opt/ohpc/pub /opt/ohpc/pub nfs nfsvers=3,nodev 0 0" >>
$CHROOT/etc/fstab

# Exporter les packages publics /home et OpenHPC depuis le serveur maître
echo "/home *(rw,no_subtree_check,fsid=10,no_root_squash)" >> /etc/exports
echo "/opt/ohpc/pub *(ro,no_subtree_check,fsid=11)" >> /etc/exports
exportfs -a
systemctl restart nfs-server
systemctl enable le serveur nfs
```

Personnalisation supplémentaire (facultatif)

Cette section met en évidence les personnalisations supplémentaires courantes qui peuvent éventuellement être appliquées à l'environnement du cluster local. Ces personnalisations incluent :

- Il est recommandé de restreindre l'accès ssh sur les nœuds de calcul pour autoriser uniquement l'accès par les utilisateurs qui ont un actif travail associé au nœud, cela peut être activé via l'utilisation d'un module d'authentification enfichable (PAM) fourni dans le cadre de l'installation du package Slurm

```
echo "account required pam_slurm.so" >> $CHROOT/etc/pam.d/sshd
```

- Il est souvent souhaitable de consolider les informations de journalisation du système. pour le cluster dans un emplacement central, à la fois pour faciliter l'accès aux données et pour réduire l'impact de stocker des données dans l'empreinte mémoire du nœud de calcul sans état. Les commandes suivantes mettent en évidence les étapes nécessaires pour configurer les nœuds de calcul pour transmettre leurs journaux au SMS et permettre au SMS d'accepter ces demandes de journal.

```
# Configurer SMS pour recevoir des messages et recharger la configuration
rsyslog
```

```
echo 'module(load="imudp")' >> /etc/rsyslog.d/ohpc.conf
echo 'input(type="imudp" port="514")' >> /etc/rsyslog.d/ohpc.conf
systemctl restart rsyslog

# Définir la destination de transfert du nœud de calcul
echo "*.* @${sms_ip}:514" >> $CHROOT/etc/rsyslog.conf
echo "Target=\"${sms_ip}\" Protocol=\"udp\"" >> $CHROOT/etc/rsyslog.conf

# Désactiver la plupart des journaux locaux sur les calculs. Les journaux
d'urgence et de démarrage resteront sur les nœuds de calcul
perl -pi -e "s/^\*\.info/\\#\*\.info/" $CHROOT/etc/rsyslog.conf
perl -pi -e "s/^authpriv/\\#authpriv/" $CHROOT/etc/rsyslog.conf
perl -pi -e "s/^mail/\\#mail/" $CHROOT/etc/rsyslog.conf
perl -pi -e "s/^cron/\\#cron/" $CHROOT/etc/rsyslog.conf
perl -pi -e "s/^uucp/\\#uucp/" $CHROOT/etc/rsyslog.conf
```

- **Nagios** est un package de surveillance de réseau d'infrastructure open source conçu pour surveiller les serveurs, les commutateurs et divers services et offre des fonctionnalités d'alerte définies par l'utilisateur pour le monitoring divers aspects d'un cluster HPC. Le démon Nagios de base et une variété de plugins de surveillance sont fournis par la distribution du système d'exploitation sous-jacent et les commandes suivantes peuvent être utilisées pour installer et configurer un serveur Nagios sur le nœud maître, et ajouter la possibilité d'exécuter des tests et de collecter des métriques à partir de nœuds de calcul. Cet exemple de configuration simple est destiné à illustrer la définition d'un groupe d'hôtes de calcul et d'activer une vérification ssh pour les calculs.

```
# Installer nagios, nrep et tous les plugins disponibles sur l'hôte
principal
yum -y install --skip-broken nagios nrpe nagios-plugins-*

# Installer nrpe et un exemple de plugin dans l'image du nœud de calcul
yum -y --installroot=$CHROOT install nrpe nagios-plugins-ssh

# Activer et configurer le démon Nagios NRPE dans l'image de calcul
chroot $CHROOT systemctl enable nrpe
perl -pi -e "s/^allowed_hosts=/# allowed_hosts=/"
$CHROOT/etc/nagios/nrpe.cfg
echo "nrpe : ${sms_ip} : ALLOW" >> $CHROOT/etc/hosts.allow
echo "nrpe : ALL : DENY" >> $CHROOT/etc/hosts.

# Copier l'exemple de fichier de configuration Nagios pour définir un groupe
de calcul et vérifier ssh
# (Il faut le modifier pour ajouter tous les hôtes de calcul souhaités)
cp /opt/ohpc/pub/examples/nagios/compute.cfg /etc/nagios/objects

# Enregistrer le fichier de configuration avec nagios
echo "cfg_file=/etc/nagios/objects/compute.cfg" >> /etc/nagios/nagios.cfg

# Mettre à jour l'emplacement du binaire de messagerie pour les commandes
d'alerte
```

```
perl -pi -e "s/ \/bin\/mail/ \/usr\/bin\/mailx/g"
/etc/nagios/objects/commands.cfg

# Mettre à jour l'adresse e-mail du contact pour les alertes
perl -pi -e "s/nagios\@localhost/root\@${sms_name}/"
/etc/nagios/objects/contacts.cfg

# Ajout de la commande check_ssh pour les hôtes distants
[sms]# echo command[check_ssh]=/usr/lib64/nagios/plugins/check_ssh localhost
$CHROOT/etc/nagios/nrpe.cfg

# définir le mot de passe pour nagiosadminpouvoir se connecter à l'interface
web
htpasswd -bc /etc/nagios/passwd nagiosadmin ${nagios_web_password}

# Activer Nagios sur le maître et configurer
systemctl enable nagios
systemctl start nagios

# Mettre à jour les autorisations sur la commande ping pour permettre à
l'utilisateur de nagios de s'exécuter
chmod u+s `which ping`
```

- **ClusterShell** est une bibliothèque Python basée sur les événements pour exécuter des commandes en parallèle à travers les nœuds du cluster. Installation et configuration de base définissant trois groupes de nœuds (adm, compute,et all) est le suivant :

```
# Installer ClusterShell
yum -y installer clustershell

# Configurer les définitions de nœuds
cd /etc/clustershell/groups.d
mv local.cfg local.cfg.orig
echo "adm: ${sms_name}" > local.cfg
echo "compute: ${compute_prefix}[1-${num_computes}]" >> local.cfg
echo "all: @adm,@compute" >> local.cfg
```

- **Genders** est une base de données de configuration de cluster statique ou une base de données de typage de nœuds utilisée pour la gestion de la configuration des clusters. D'autres outils et utilisateurs peuvent accéder à la base de données des genres afin de prendre des décisions sur l'endroit où une action, ou même quelle action, est appropriée en fonction des types ou « genres ». Des valeurs peuvent également être attribuées à et extraites d'un genre pour fournir une granularité supplémentaire. L'exemple suivant met en évidence l'installation et la configuration de deux genres : compute et bmc.

```
# Installer genders
yum -y install genders-ohpc

# Générer un exemple de fichier de genre
echo -e "${sms_name}\tsms" > /etc/genders
```

```
for ((i=0; i<$num_computes; i++)) ; do
    echo -e "${c_name[$i]}\tcompute,bmc=${c_bmc[$i]}"
done >> /etc/genders
```

- **Magpie** contient un certain nombre de scripts pour aider à exécuter une variété de données volumineuses frameworks logiciels dans les environnements de files d'attente HPC. Les exemples incluent Hadoop, Spark, Hbase, Storm, Pig, Mahout, Phoenix, Kafka, Zeppelin et Zookeeper:

```
# Installer Magpie
yum -y install magpie-ohpc
```

- **ConMan** est un programme de gestion de console série conçu pour prendre en charge un grand nombre de consoles et d'utilisateurs simultanés. Il prend en charge la sortie de périphérique de console de journalisation et la connexion pour calculer les consoles de nœuds via IPMI serial-over-lan. L'installation et l'exemple de configuration sont décrits ci-dessous.

```
# Installer conman pour fournir un front-end aux consoles de calcul et à la
sortie des journaux
yumm -y installer conman-ohpc

# Configurer conman pour les calculs (notez que votre mot de passe IPMI est
requis pour l'accès à la console)
for ((i=0; i<$num_computes; i++)) ; do
    echo -n 'CONSOLE name="'${c_name[$i]}'" dev="ipmi:'${c_bmc[$i]}'" '
    echo 'ipmiopts="'U:${bmc_username},P:${IPMI_PASSWORD:-
undefined},W:solpayloadsize'"'
done >> /etc/conman.conf

# Activer et démarrer conman
systemctl enable conman
systemctl start conman
```

Une option de démarrage du noyau supplémentaire est généralement nécessaire pour activer la sortie de la console série. Cette option est mis une fois que les nœuds de calcul ont été enregistrés auprès du système d'approvisionnement.

- Les gestionnaires de ressources prévoient souvent un « contrôle de l'état des nœuds » périodique à effectuer sur chaque nœud de calcul pour vérifier que le nœud fonctionne correctement. Les nœuds qui sont déterminés « non sain » peut être marqué comme étant en panne ou hors ligne afin d'empêcher la planification ou l'exécution de tâches sur celles-ci. Cela permet d'augmenter la fiabilité et le débit d'un cluster en réduisant les échecs de travail évitables dus à une mauvaise configuration, panne matérielle, etc. OpenHPC distribue **NHC** pour répondre à cette exigence. Dans un scénario typique, le script du pilote **NHC** est exécuté périodiquement sur chaque nœud de calcul par la ressource démon client du gestionnaire. Il charge son fichier de configuration pour déterminer quels contrôles doivent être exécutés sur le nœud (en fonction de son nom d'hôte). Chaque vérification de correspondance est exécutée, et

si un échec est rencontré, **NHC** quittera avec un message d'erreur décrivant le problème. Il peut également être configuré pour marquer les nœuds hors ligne afin que le planificateur n'affectera pas de tâches aux nœuds défectueux, réduisant ainsi le risque d'échecs de tâches induits par le système.

```
# Installer NHC sur les nœuds maître et de calcul
yum -y installer nhc-ohpc
yum -y --installroot=$CHROOT install nhc-ohpc

# S'inscrire au programme de bilan de santé du SLURM
echo "HealthCheckProgram=/usr/sbin/nhc" >> /etc/slurm/slurm.conf
echo "HealthCheckInterval=300" >> /etc/slurm/slurm.conf # s'exécute toutes
les cinq minutes
```

Importer des fichiers

Le système Warewulf inclut des fonctionnalités pour importer des fichiers arbitraires depuis le serveur de provisionnement pour la distribution aux hôtes gérés. C'est une façon de distribuer les informations d'identification des utilisateurs aux nœuds de calcul. Pour importer localement les informations d'identification basées sur des fichiers, exécuter ce qui suit:

```
wwsh file import /etc/passwd
wwsh file import /etc/group
wwsh file import /etc/shadow
```

De même, pour importer la clé cryptographique requise par la bibliothèque d'authentification Munge pour être disponible sur tout hôte dans le pool de gestion des ressources, exécuter ce qui suit :

```
wwsh file import /etc/munge/munge.key
```

Finalisation de la configuration de provisionnement

Warewulf utilise un processus de démarrage en deux étapes pour le provisionnement des nœuds via la création d'une image d'amorçage qui est utilisé pour initialiser le processus et une capsule de système de fichiers de nœud virtuel contenant l'image système complète.

Cette section met en évidence la création des images d'approvisionnement nécessaires, suivie de l'enregistrement des nœuds de calcul.

Assembler l'image d'amorçage

L'image d'amorçage comprend le noyau d'exécution et les modules associés, ainsi que quelques scripts simples pour terminer le processus d'approvisionnement. Les commandes suivantes mettent en évidence l'inclusion de pilotes supplémentaires et la création de l'image d'amorçage basée sur le

noyau en cours d'exécution.

```
# (Facultatif) Inclut les pilotes des mises à jour du noyau ;
# nécessaire si l'activation de modules de noyau supplémentaires sur les
calculs
export WW_CONF=/etc/warewulf/bootstrap.conf
echo "drivers += updates/kernel/" >> $WW_CONF

# Construire une image d'amorçage
[sms]# wwbootstrap `uname -r`
```

Assembler l'image du système de fichiers de nœud virtuel (VNFS)

Avec les personnalisations du site local en place, l'étape suivante utilise la commande wwvnfs pour assembler un VNFS capsule à partir de l'environnement chroot défini pour l'instance de calcul.

```
wwvnfs --chroot $CHROOT
```

Enregistrer les nœuds pour le provisionnement

En vue de l'approvisionnement, on peut maintenant définir les paramètres réseau souhaités pour quatre nœuds de calcul avec le système d'approvisionnement sous-jacent et redémarrez le service DHCP. Notez l'utilisation de noms de variables pour les noms d'hôte de calcul, les adresses IP de nœud et les adresses MAC souhaités qui doivent être modifiés pour s'adapter les paramètres locaux et le matériel.

Par défaut, Warewulf utilise des noms d'interface réseau de la variété eth# et ajoute les arguments de démarrage du noyau pour maintenir ce schéma sur les noyaux plus récents. Par conséquent, lors de la spécification de l'interface de provisionnement via la variable de provision $eth, il doit suivre cette convention.

Alternativement, si on préfére utiliser le schéma de nommage de l'interface réseau prévisible (par exemple, des noms comme en4s0f0), en plus des étapes sont incluses pour modifier les arguments de démarrage du noyau par défaut et supprimer l'interface nommée eth# après l'amorçage afin que le processus d'initialisation normal puisse l'afficher à nouveau en utilisant le nom souhaité.

La dernière étape de ce processus associe l'image VNFS assemblée dans les étapes précédentes avec la nouvelle nœuds de calcul définis, en utilisant les fichiers d'informations d'identification de l'utilisateur et la clé Munge qui ont été importés.

```
# Définir l'interface d'approvisionnement comme périphérique réseau par
défaut
echo "GATEWAYDEV=${eth_provision}" > /tmp/network.$$
wwsh -y fichier import /tmp/network.$$ --name network
wwsh -y file set network --path /etc/sysconfig/network --mode=0644 --uid=0
```

```
# Ajouter des nœuds au magasin de données Warewulf
for ((i=0; i<$num_computes; i++)) ; do
    wwsh -y node new ${c_name[i]} --ipaddr=${c_ip[i]} --hwaddr=${c_mac[i]} -
D ${eth_provision}
done

# Étape supplémentaire requise si pour utiliser une interface réseau
prévisible dans le
# schémas de nommage (par exemple en4s0f0). Ignorer si on utilise des noms
de style eth#.
export kargs="${kargs} net.ifnames=1,biosdevname=1"
wwsh provision set --postnetdown=1 "${compute_regex}"

# Définir l'image de provisionnement pour les hôtes
wwsh -y provision set "${compute_regex}" --vnfs=centos8.4 --bootstrap=`uname
-r` \
--files=dynamic_hosts,passwd,group,shadow,munge.key,network
```

Warewulf inclut un utilitaire nommé wwnodescan pour enregistrer automatiquement de nouveaux nœuds de calcul par rapport à l'approche d'ajout de nœud qui nécessite que les adresses MAC matérielles soient collectées à l'avance. Avec wwnodescan, les nœuds seront ajoutés à la base de données Warewulf dans l'ordre dans lequel leurs requêtes DHCP sont reçu par le maître, il faut donc veiller à démarrer les nœuds dans l'ordre que l'on souhaite voir conservé dans la Base de données Warewulf. L'adresse IP fournie sera incrémentée une fois chaque nœud trouvé et l'utilitaire se fermera une fois que tous les nœuds spécifiés auront été trouvés. Un exemple d'utilisation est mis en évidence ci-dessous :

```
wwnodescan --netdev=${eth_provision} --ipaddr=${c_ip[0]} --
netmask=${internal_netmask} \
--vnfs=centos8.4 --bootstrap=uname -r --listen=${sms_eth_internal}
${c_name[0]}-${c_name[3]}
```

```
# Redémarrer dhcp / mettre à jour PXE`
systemctl restart dhcpd\\ `
wwsh pxe update
```

Arguments de noyau facultatifs

L'environnement d'exécution du conteneur Charliecloud nécessite l'activation de l'option de mappage des espaces de noms d'utilisateurs. Ceci permet aux applications exécutées avec le privilège root à l'intérieur d'un conteneur, mais qu'elles s'exécutent sous une forme différente, généralement non privilégié, sur l'hôte. Bien que généralement considéré comme mature et sûr, CentOS considère qu'il s'agit d'un Technology Preview, il doit donc être activé manuellement avec les arguments du noyau. On l'activera sur nœuds de calcul, mais cela serait également requis sur le SMS si on souhaite y créer et exécuter des conteneurs.

```
# Définir les arguments du noyau du nœud à supespaces de noms d'utilisateur
de port
export kargs="${kargs} namespace.unpriv_enable=1"

# Augmenter la limite par utilisateur du nombre d'espaces de noms
d'utilisateurs pouvant être créés
echo "user.max_user_namespaces=15076" >> $CHROOT/etc/sysctl.conf

# reconstruire VNFS
wwvnfs --chroot $CHROOT
```

Les workflows typiques de Charliecloud sont basés sur des conteneurs Docker, mais il n'est pas strictement nécessaire d'installer Docker lui-même sur la ressource HPC. Un modèle courant consiste à créer le conteneur Docker sur un ordinateur portable ou VM et télécharger le résultat dans le cluster pour une utilisation avec Charliecloud.

Lorsqu'on a activer **ConMan**, une configuration supplémentaire de warewulf est nécessaire comme suit :

```
# Définir les arguments du noyau du nœud pour prendre en charge la console
SOL
wwsh -y provision set "${compute_regex}" --console=ttyS1,115200
```

Si des composants ont été ajoutés aux arguments de ligne de commande du noyau de démarrage pour les nœuds de calcul, la commande suivante est requise pour stocker la configuration dans Warewulf :

```
# Définir les arguments de ligne de commande du noyau du nœud de calcul
facultatifs.
wwsh -y provision set "${compute_regex}" --kargs="${kargs}"
```

Configurer éventuellement le provisionnement avec état

Warewulf exécute normalement par défaut l'image VNFS assemblée hors de la mémoire système dans un état sans état configuration. Alternativement, Warewulf peut également être utilisé pour partitionner et formater le stockage persistant de telle sorte que l'image VNFS peut être installée localement sur le disque de manière dynamique. Cela nécessite cependant qu'un chargeur de démarrage (GRUB) soit ajouté à l'image comme suit :

```
# Ajouter le chargeur de démarrage GRUB2 et réassembler l'image VNFS
yum -y --installroot=$CHROOT install grub2
wwvnfs --chroot $CHROOT
```

L'activation des nœuds avec état nécessite également des paramètres supplémentaires spécifiques

au site et liés au disque dans la configuration Warewulf et plusieurs exemples de scripts de partitionnement sont fournis dans la distribution.

```
# Sélectionner (et personnaliser) l'exemple de disposition en parties
approprié
cp /etc/warewulf/filesystem/examples/gpt_example.cmds
/etc/warewulf/filesystem/gpt.cmds
wwsh provision set --filesystem=gpt "${compute_regex}"
wwsh provision set --bootloader=sda "${compute_regex}"
```

Ceux qui provisionnent des nœuds de calcul en mode UEFI installeront un ensemble de packages légèrement différent dans le VNFS. Warewulf fournit également un exemple de disposition de système de fichiers EFI.

```
# Ajouter le chargeur de démarrage GRUB2 et réassembler l'image
VNFS
yum -y --installroot=$CHROOT installer grub2-efi grub2-efi-modules
wwvnfs --chroot $CHROOT
cp /etc/warewulf/filesystem/examples/efi_example.cmds
/etc/warewulf/filesystem/efi.cmds
wwsh provision set --filesystem=efi "${compute_regex}"
wwsh provision set --bootloader=sda "${compute_regex}"
```

Lors du redémarrage ultérieur des nœuds modifiés, Warewulf partitionnera et formatera le disque pour héberger l' image VNFS souhaitée. Une fois l'image installée sur le disque, warewulf peut être configuré pour utiliser le local des nœuds de stockage en tant que périphérique de démarrage.

```
# Configurer le démarrage local (après un provisionnement réussi)
wwsh provision set --bootlocal=normal "${compute_regex}"
```

Démarrage des nœuds de calcul

À ce stade, le serveur maître doit pouvoir démarrer les nœuds de calcul nouvellement définis. En supposant que les paramètres BIOS du nœud de calcul sont configurés pour démarrer sur PXE, tout ce qui est nécessaire pour lancer le processus de provisionnement consiste à éteindre et rallumer chacun des hôtes souhaités à l'aide de l'accès IPMI. Les commandes suivantes utilisent l'utilitaire ipmitool pour lancer les réinitialisations d'alimentation sur chacun des quatre hôtes de calcul. L'utilitaire nécessite que la variable d'environnement IPMI PASSWORD soit définie avec le mot de passe BMC local pour fonctionner de manière interactive.

```
for ((i=0; i<${num_computes}; i++)) ; do
    ipmitool -E -I lanplus -H ${c_bmc[$i]} -U ${bmc_username} -P
${bmc_password} chassis power reset
done
```

Une fois lancé, le processus de démarrage devrait prendre moins de 5 minutes (selon les heures de publication du BIOS) et on peut vérifier que les hôtes de calcul sont disponibles via ssh ou via des outils ssh parallèles vers plusieurs hôtes. Pour par exemple, exécuter une commande sur les hôtes de calcul nouvellement imagés à l'aide de pdsh, exécuter la commande suivante :

```
pdsh -w c[1-4] uptime
c1 05:03am up 0:02, 0 users, load average: 0.20, 0.13, 0.05
c2 05:03am up 0:02, 0 users, load average: 0.20, 0.14, 0.06
c3 05:03am up 0:02, 0 users, load average: 0.19, 0.15, 0.06
c4 05:03am up 0:02, 0 users, load average: 0.15, 0.12, 0.05
```

Alors que les fichiers pxelinux.0 et lpxelinux.0 fournis avec Warewulf pour permettre le démarrage réseau prennent en charge une large gamme de matériel, certains hôtes peuvent démarrer de manière plus fiable ou plus rapide en utilisant les versions BOS fournies via le paquet syslinux-tftpboot. Si vous rencontrez des problèmes PXE, envisagez de remplacer le pxelinux.0 et lpxelinux.0 fournis avec warewulf-provision-ohpc avec les versions de syslinux-tftpboot.