

Installation et configuration d'OpenStack (Kilo) dans Oracle® Solaris

ORACLE

Référence: E74913
Juin 2016

Référence: E74913

Copyright © 2014, 2016, Oracle et/ou ses affiliés. Tous droits réservés.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf stipulation expresse de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, accorder de licence, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est livré sous licence au Gouvernement des Etats-Unis, ou à quiconque qui aurait souscrit la licence de ce logiciel pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer un risque de dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour des applications dangereuses.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée de The Open Group.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers, sauf mention contraire stipulée dans un contrat entre vous et Oracle. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation, sauf mention contraire stipulée dans un contrat entre vous et Oracle.

Accessibilité de la documentation

Pour plus d'informations sur l'engagement d'Oracle pour l'accessibilité à la documentation, visitez le site Web Oracle Accessibility Program, à l'adresse <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Accès aux services de support Oracle

Les clients Oracle qui ont souscrit un contrat de support ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> ou le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> si vous êtes malentendant.

Table des matières

Utilisation de la présente documentation	11
1 Nouveautés d'OpenStack (Kilo) sur Oracle Solaris	13
Fonctionnalités basées sur le composant principal	13
Fonctionnalités liées à Cinder	13
Fonctionnalités liées à Nova	14
Fonctionnalités liées à Neutron	15
Ajout de fonctionnalités Oracle Solaris	15
Autres sources d'informations	16
2 Mise à niveau vers OpenStack 2015.1.2 (Kilo)	17
Notes de mise à niveau	17
Procédures de mise à niveau	17
▼ Procédure de mise à niveau de Havana vers Kilo	18
▼ Procédure de mise à niveau de Juno vers Kilo	19
▼ Tâches à effectuer après la mise à niveau	19
3 Installation sur plusieurs systèmes pour une configuration OpenStack multinoeud	23
Présentation de l'architecture à trois noeuds	23
Etapas préliminaires	26
Préparation des noms d'hôtes, variables et mots de passe	26
Exemple de script Keystone	27
A propos de la modification des fichiers de configuration	28
Optimisation de l'utilisation de la mémoire	28
Configuration du serveur NTP	29
▼ Configuration du serveur NTP	29
Configuration du noeud de contrôleur	30
Configuration du client NTP	30
Installation de MySQL	31

Installation de Keystone	32
Installation de Glance	33
Installation de Nova	35
Installation d'Horizon	36
Installation de Cinder	37
Installation de Neutron	38
Configuration du noeud de calcul	41
▼ Configuration du noeud de calcul	42
▼ Activation de l'accès à la console	44
Configuration du noeud de stockage	46
▼ Configuration du noeud de stockage par bloc	46
Configuration du stockage d'objets Swift	47
▼ Configuration du noeud de service pour le contrôleur proxy Swift	48
▼ Configuration du noeud de stockage d'objets	49
4 Tâches de post-installation et de configuration	51
Préparation du réseau externe pour les projets OpenStack	51
A propos du routeur de fournisseur	51
Création du réseau externe	53
Préparation des images pour le référentiel Glance	59
Création des images	60
Affichage des informations sur les images	62
Utilisation d'un script de création d'image Glance	63
5 Utilisation du cloud	65
Accès au tableau de bord OpenStack	65
▼ Accès au tableau de bord OpenStack	65
Exploration du tableau de bord	66
Création de projets et d'utilisateurs	68
▼ Création d'un projet et affectation des utilisateurs	68
▼ Alimentation d'un projet avec des utilisateurs existants	69
Création de réseaux internes pour les projets	71
▼ Configuration d'un réseau pour un projet	71
▼ Association d'une adresse IP flottante à un projet	74
Création et initialisation d'une instance de machine virtuelle	74
▼ Création d'une paire de clés SSH	74
▼ Création d'une instance de machine virtuelle	75
▼ Ajout d'utilisateurs à une instance de machine virtuelle	78
Gestion des variantes	80

Affichage des informations sur les variantes	80
Modification des spécifications de variante	81
Gestion des instances de machine virtuelle	83
Migration et évacuation des instances de machine virtuelle	84
Redimensionnement d'une instance de machine virtuelle	86
6 Options pour la configuration et le déploiement de Cinder	91
Déploiement de systèmes distants pour le stockage	91
Configuration du fichier <code>cinder.conf</code>	92
Octroi des droits à l'utilisateur désigné	95
Activation de l'hôte distant en tant que cible	96
Spécification de volumes d'initialisation pour les noeuds de calcul	96
▼ Création de volumes de stockage root pour des instances de calcul	97
Utilisation du pilote NFS Cinder	98
▼ Procédure d'utilisation du pilote NFS Cinder	98
Utilisation d'OpenStack avec Oracle ZFS Storage Appliance	100
À propos d'Oracle ZFS Storage Appliance	100
Configurer OpenStack avec Oracle ZFSSA	100
▼ Configuration d'Oracle ZFSSA pour OpenStack	102
7 Options pour le déploiement de Neutron	107
Déploiement de Neutron dans une zone de noyau	107
▼ Installation du composant Neutron dans une zone de noyau	109
Affichage des informations d'adresse MAC et de VID	110
Affichage à partir de la machine virtuelle invitée	110
Affichage à partir de l'hôte	110
8 Utilisation d'Ironic	113
A propos du composant Ironic	113
Installation et configuration d'Ironic	114
▼ Installation et configuration d'Ironic	114
Présentation : Déploiement sans système d'exploitation avec Ironic	119
Utilisation d'Ironic pour un déploiement sans système d'exploitation	121
▼ Déploiement sans système d'exploitation à partir d'un fichier UAR	121
▼ Mise hors service d'un noeud	124
9 Utilisation de Heat	127
A propos du composant Heat	127

Installer Heat	128
▼ Configuration de Heat	128
A propos des modèles HOT	128
Utilisation de Heat avec Cloudbase-Init	130
▼ Initialisation automatique d'une image invitée	130
10 Dépannage d'OpenStack	133
Obtention de l'aide sur la ligne de commande	133
Limitations connues	134
Examen des fichiers journaux	135
Examen et résolution des problèmes	138
Création d'un réseau	139
Installation et configuration de l'instance de machine virtuelle	139
Problème lié à Horizon	142
Problèmes d'évolutivité	143
Démontage de réseaux	143
▼ Suppression de la configuration réseau sur Neutron	143
▼ Suppression des Vports	144
Conseils et astuces généraux sur le débogage	144
Sites utiles	145
A Fichiers de configuration et services OpenStack courants	147
Fichiers de configuration	147
Fichiers Cinder	147
Fichiers Glance	147
Fichiers Keystone	147
Fichiers Neutron	148
Fichiers Nova	148
Fichiers Horizon	148
Fichiers Swift	148
Services SMF d'Openstack	149
Cinder	149
Glance	149
Keystone	149
Neutron	149
Nova *	149
Swift	150
B Commandes OpenStackClient	151

OpenStackClient (OSC)	151
Index	153

Utilisation de la présente documentation

- **Présentation** – Décrit la procédure d'installation de la version actuelle d'OpenStack et de déploiement de machines virtuelles OpenStack sur les systèmes Oracle Solaris pris en charge.
- **Public visé** – Administrateurs de grands systèmes d'installation.
- **Connaissances requises** – Administration réseau et grands systèmes Oracle Solaris. Etre familiarisé avec OpenStack est très utile.

Bibliothèque de documentation produit

La documentation et les ressources de ce produit et des produits associés sont disponibles sur le site Web <http://www.oracle.com/pls/topic/lookup?ctx=E69399>.

Commentaires

Faites part de vos commentaires sur cette documentation à l'adresse : <http://www.oracle.com/goto/docfeedback>.

Nouveautés d'OpenStack (Kilo) sur Oracle Solaris

Oracle Solaris 11.3 SRU 9 inclut la version OpenStack 2015.1.2 (Kilo) dans son package générique. Ce chapitre répertorie les nouvelles fonctionnalités présentes dans cette version Kilo d'OpenStack.

- ["Fonctionnalités basées sur le composant principal" à la page 13](#)
- ["Autres sources d'informations" à la page 16](#)

Fonctionnalités basées sur le composant principal

Cette section décrit les fonctionnalités introduites dans les composants principaux de la version Kilo d'OpenStack.

Fonctionnalités liées à Cinder

Les fonctionnalités suivantes ont été ajoutées pour Cinder :

- Utilisation du stockage SAN distant
La prise en charge de réseaux de stockage (SAN) vous permet de déployer le service Cinder à distance. Pour plus d'informations, reportez-vous à la section ["Déploiement de systèmes distants pour le stockage" à la page 91](#).
- Prise en charge d'opérations de sauvegarde et de restauration de volumes
Le service SMF de sauvegarde Cinder est désormais activé dans Oracle Solaris. Ainsi, les volumes non associés peuvent être sauvegardés et restaurés entre des back-ends configurés. Swift est le seul back-end pris en charge actuellement.
- Prise en charge de la migration de volumes Cinder
Cinder profite des opérations ZFS pour migrer les volumes Cinder. Les processus d'envoi et de réception ZFS permettent la migration de volumes entre différents back-ends Cinder configurés. Si la cible de la migration se trouve sur le même zpool que la source, une opération de renommage ZFS est utilisée. La prise en charge de la migration de volumes Cinder se limite actuellement à un seul système.

- La gestion de volumes dans Cinder a été améliorée à l'aide d'une option *manage*, afin que vous puissiez importer des volumes créés en dehors de la fonctionnalité Cinder. Une fois que ces volumes ont été importés, vous pouvez les administrer dans le cloud comme s'il s'agissait de volumes Cinder ordinaires.

De même, vous pouvez rendre les volumes Cinder invisibles à l'aide de l'option *unmanage* qui permet de désactiver l'accès. L'option ne supprime pas les volumes. Vous pouvez ainsi réactiver l'accès aux volumes en les réimportant.

La fonctionnalité Gérer/Ne pas gérer est disponible à la fois sur le tableau de bord Horizon et sur la ligne de commande.
- Un pilote Cinder ZFSSA mis à jour avec une nouvelle propriété est disponible dans l'implémentation Kilo actuelle. Pour régler vos paramètres de configuration ZFSSA dans OpenStack, reportez-vous aux instructions de la section "[Tâches à effectuer après la mise à niveau](#)" à la page 19.
- Le pilote de volumes NFS OpenStack Cinder pour Solaris est pris en charge. Vous pouvez créer un volume de type *nfs*. L'accès aux fichiers NFS est défini avec l'utilisateur et le groupe Cinder. Toutefois, notez qu'actuellement la prise en charge de ce pilote se limite aux zones de noyau.

Pour plus d'informations, reportez-vous à la section "[Utilisation du pilote NFS Cinder](#)" à la page 98.
- Dans une configuration comportant plusieurs back-ends, vous pouvez spécifier le volume d'initialisation pour chaque noeud de calcul que vous créez. Pour configurer Cinder avec cette fonctionnalité, reportez-vous à la section "[Spécification de volumes d'initialisation pour les noeuds de calcul](#)" à la page 96.

Fonctionnalités liées à Nova

Les fonctionnalités suivantes ont été ajoutées pour Nova :

- Migration en direct sécurisée

La prise en charge de la migration en direct, qui est une fonctionnalité des zones Oracle Solaris, a été étendue aux instances de machine virtuelle sur les noeuds Nova. Pour plus d'informations sur la migration de noeuds en direct, reportez-vous à la section "[Migration et évacuation des instances de machine virtuelle](#)" à la page 84.
- Prise en charge de l'évacuation d'instances

En cas de défaillance de l'hôte ou de désactivation de services sur l'hôte, vous pouvez déplacer une instance vers un autre noeud en vue de sa récupération à l'aide de la commande `nova evacuate`. Notez que la prise en charge de l'évacuation est disponible uniquement si les périphériques root sont sur le stockage partagé. En outre, l'évacuation n'est prise en charge que pour les zones de noyau et pas pour les zones non globales.
- Fonctionnalité de redimensionnement des instances de machine virtuelle

Vous pouvez redimensionner des instances de machine virtuelle en modifiant leurs types d'instance. Les nouveaux types d'instance fournissent des propriétés différentes aux

instances de machine virtuelle, telles que la capacité de la CPU, la mémoire et d'autres ressources. Pour plus d'informations, reportez-vous à la section "[Redimensionnement d'une instance de machine virtuelle](#)" à la page 86.

Fonctionnalités liées à Neutron

Les fonctionnalités suivantes ont été ajoutées pour Neutron :

- Fonction Neutron dans la zone de noyau
La prise en charge d'adresses MAC dynamiques et de VID dans des zones vous permet d'installer Neutron dans une zone de noyau. Pour plus d'informations, reportez-vous à la section "[Déploiement de Neutron dans une zone de noyau](#)" à la page 107.
- VPN as a Service
VPN as a Service (VPNaaS) est pris en charge via Neutron. En outre, le profil "Network IPsec Management" est ajouté aux profils qui ont déjà été affectés à Neutron. Ce profil permet à l'administrateur de gérer les services SMF IPsec et IKE.

Ajout de fonctionnalités Oracle Solaris

Ces ajouts de fonctionnalités représentent des améliorations apportées au pilote d'OpenStack pour Oracle Solaris. Ces améliorations ont déjà été implémentées sur le projet en amont principal.

Prise en charge de cloudbase-init

Le service cloudbase-init facilite l'initialisation et la configuration de systèmes d'exploitation invités dans le cloud. Les tâches incluent la création d'utilisateurs, la génération de mots de passe, la configuration de réseaux statiques, le nom d'hôte, les clés publiques SSH et les scripts de données utilisateur. Le fichier de configuration pour le service est `/etc/cloudbase-init.conf`.

La version Oracle Solaris de cloudbase-init s'exécute en tant que service SMF, `application/cloudbase-init`, et est activée par défaut. Les scripts qui sont exportés par le biais des données utilisateur exécutent généralement des tâches de configuration d'application et de système nécessitant un accès privilégié. Par conséquent, le service cloudbase-init s'exécutera en tant qu'utilisateur root, ainsi que les scripts de données utilisateur.

Le package cloudbase-init n'est inclus dans aucun package de groupe standard. Les utilisateurs doivent installer le package uniquement dans des images qui sont expressément destinées à un déploiement dans des environnements de cloud.

Remarque - Actuellement, dans cette version d'OpenStack, le fichier `/etc/cloudbase-init.conf` active uniquement le module d'extension `UserData`.

Pour plus d'informations sur `cloudbase-init`, accédez à la page <http://cloudbase-init.readthedocs.io/en/latest/tutorial.html>.

Implémentation d'OpenStackClient

OpenStackClient (OSC) est un client de la communauté OpenStack qui combine tous les ensembles des commandes du composant dans un shell unique avec une structure de commande uniforme. Ainsi, alors que les versions précédentes contenaient des commandes basées sur les composants, telles que `keystone user-list`, `glance image-show`, `neutron net-list`, etc., la plupart de ces commandes sont exécutées avec `openstack` comme commande principale, par exemple `openstack user list`.

Dans la version Kilo actuelle, toutes les commandes `keystone` sont obsolètes. Des alertes appropriées sont générées si vous utilisez la commande `keystone`.

Pour plus d'informations sur OSC, reportez-vous à la page <http://docs.openstack.org/developer/python-openstackclient/index.html>.

Pour obtenir la liste des commandes précédentes et de leurs équivalents dans OSC, reportez-vous à l'[Annexe B, Commandes OpenStackClient](#).

Autres sources d'informations

Reportez-vous également aux informations fournies par les notes de version Kilo de la communauté OpenStack à l'adresse <https://wiki.openstack.org/wiki/ReleaseNotes/Kilo>.

Pour obtenir la liste des problèmes liés à l'implémentation Oracle Solaris actuelle de cette version d'OpenStack, consultez le fichier `Readme` correspondant à la version d'OpenStack dans [My Oracle Support \(https://support.oracle.com\)](https://support.oracle.com).

◆◆◆ CHAPITRE 2

Mise à niveau vers OpenStack 2015.1.2 (Kilo)

Ce chapitre décrit la procédure de mise à niveau vers la version Kilo d'OpenStack à partir d'une version antérieure.

- ["Notes de mise à niveau" à la page 17](#)
- ["Procédures de mise à niveau" à la page 17](#)

Notes de mise à niveau

Tenez compte des meilleures pratiques suivantes lors de la mise à niveau :

- Créez des sauvegardes, en particulier si vous ne mettez à niveau que la configuration OpenStack et non l'ensemble du système d'exploitation.
- Si vous mettez à niveau uniquement la configuration OpenStack, un nouvel environnement d'initialisation est créé avec la version mise à niveau d'OpenStack. Initialisez à partir du nouvel environnement d'initialisation une fois la mise à niveau terminée.
- Dans une configuration multinoeud, mettez d'abord à jour le noeud de contrôleur, puis le noeud de stockage, suivi du reste des noeuds.
- Dans Kilo, les paramètres d'utilisateur invité et de mot de passe par défaut pour RabbitMQ sont désactivés. Si vous avez utilisé le paramètre de mot de passe par défaut dans la version précédente d'OpenStack, remplacez-le par un mot de passe sécurisé.

Procédures de mise à niveau

La procédure de mise à niveau vers Kilo dépend de la version d'OpenStack à partir de laquelle vous démarrez. Une mise à niveau directe de Havana vers Kilo n'est pas prise en charge. Si votre version actuelle est Havana, vous devez d'abord effectuer une mise à niveau vers Juno, puis une autre de Juno vers Kilo.

Remarque - Lors de la mise à niveau, vous devez utiliser la commande `pkg update` pour lancer l'opération. N'utilisez pas `pkg install`.

Le déroulement de la mise à niveau est le suivant :

- A partir d'une configuration Havana existante, démarrez à la section "[Procédure de mise à niveau de Havana vers Kilo](#)". Passez ensuite aux "[Tâches à effectuer après la mise à niveau](#)".
- A partir d'une configuration Juno existante, démarrez à la section "[Procédure de mise à niveau de Juno vers Kilo](#)". Passez ensuite aux "[Tâches à effectuer après la mise à niveau](#)".

▼ Procédure de mise à niveau de Havana vers Kilo

Utilisez cette procédure si votre version actuelle d'OpenStack est Havana exécuté sous Oracle Solaris 11.2 SRU 11 ou une version antérieure.

Si votre configuration OpenStack est basée sur plusieurs noeuds, vous devez effectuer cette procédure sur chacun d'eux, en commençant par le contrôleur.

Les mises à niveau Oracle Solaris créent un nouvel environnement d'initialisation pour le système d'exploitation. A l'issue de la mise à niveau, le nouvel environnement d'initialisation est activé. Lors de la réinitialisation, le système s'amorce sur le nouvel environnement d'initialisation.

1. **Si le système exécute Oracle Solaris 11.2 SRU 11 ou une version antérieure, procédez comme suit :**
 - a. **Mettez à niveau le système d'exploitation vers Oracle Solaris 11.3 au minimum.**

Remarque - A ce stade, n'effectuez pas encore de mise à niveau vers Oracle Solaris 11.3 SRU 9.

Le package Oracle Solaris 11.3 inclut les packages OpenStack Juno.

Pour connaître la procédure de mise à niveau vers Oracle Solaris 11.3, reportez-vous à la section "[How to Update a System Running 11.1 or 11.2 to Oracle Solaris 11.3](#)" du manuel [Updating to Oracle Solaris 11.3](#).

- b. **Vérifiez si les informations relatives à la migration de la base de données Neutron s'appliquent à votre configuration.**

Reportez-vous à la page [Migrating Neutron Database from sqlite to MySQL for Oracle OpenStack for Oracle Solaris](#). Suivez les instructions de l'entrée de blog si le scénario s'applique à votre configuration.

2. **Mettez à niveau le système d'exploitation vers Oracle Solaris 11.3 SRU 9.**

Pour plus d'informations sur la mise à jour vers la version Oracle Solaris 11.3 SRU 9, connectez-vous à votre compte MOS à l'adresse <https://support.oracle.com>. A partir de l'index des mises à jour du référentiel support (SRU) d'Oracle Solaris 11.3 (ID de doc 2045311.1), accédez au fichier Readme SRU9.

3. Amorcez le système sur le nouvel environnement d'initialisation.

Étapes suivantes Terminez la mise à niveau en effectuant toutes les étapes des "[Tâches à effectuer après la mise à niveau](#)" à la page 19.

▼ Procédure de mise à niveau de Juno vers Kilo

Utilisez cette procédure si votre version actuelle d'OpenStack est Juno.

Si votre configuration OpenStack est basée sur plusieurs noeuds, vous devez effectuer cette procédure sur chacun d'eux, en commençant par le contrôleur.

Les mises à niveau Oracle Solaris créent un nouvel environnement d'initialisation pour le système d'exploitation. A l'issue de la mise à niveau, le nouvel environnement d'initialisation est activé. Lors de la réinitialisation, le système s'amorce sur le nouvel environnement d'initialisation.

1. Mettez à niveau le système d'exploitation vers Oracle Solaris 11.3 SRU 9.

Pour plus d'informations sur la mise à jour vers la version Oracle Solaris 11.3 SRU 9, consultez le fichier Readme correspondant à votre version.

2. Effectuez les "[Tâches à effectuer après la mise à niveau](#)" pour terminer la mise à niveau.

▼ Tâches à effectuer après la mise à niveau

Après la mise à niveau vers la version actuelle d'Oracle Solaris, exécutez les étapes restantes pour achever la mise à niveau vers OpenStack Kilo.

1. Migrez les personnalisations Horizon vers la version Kilo.

- a. **Transférez les paramètres de personnalisation de `/etc/openstack_dashboard/local_settings.py.old` vers `/etc/openstack_dashboard/local_settings.py`.**

b. Mettez en commentaire des lignes supplémentaires dans le fichier `/etc/openstack_dashboard/local_settings.py` si l'une des conditions suivantes s'applique à votre configuration :

- Vous disposez d'une configuration OpenStack mononoeud à des fins d'évaluation.
- Votre configuration Horizon n'utilise pas SSL.

Voyez l'exemple suivant :

```
# SECURE_PROXY_SSL_HEADER = ('HTTP_X_FORWARDED_PROTOCOL', 'https')
# CSRF_COOKIE_SECURE = True
# SESSION_COOKIE_SECURE = True
```

c. Copiez les exemples de fragments de configuration Horizon Apache dans le répertoire `Apache conf.d`.

Vous devez copier uniquement l'exemple de fragment correspondant au protocole que vous utilisez. Exécutez l'une des commandes suivantes :

- Si vous utilisez HTTP :

```
# cp /etc/apache2/2.4/samples-conf.d/openstack-dashboard-http.conf /etc/
  apache2/2.4/conf.d
```

- Si vous utilisez TLS :

```
# cp /etc/apache2/2.4/samples-conf.d/openstack-dashboard-tls.conf /etc/
  apache2/2.4/conf.d
```

2. Si vous disposez d'une configuration multinoeud, mettez à jour `/etc/rabbitmq/rabbitmq.config` avec la ligne affichée en gras :

```
% FHC read buffer has been disabled by default in later versions of
%RabbitMQ.
[
  {rabbit, [
    {fhc_read_buffering, false},
    {loopback_users, []}
  ]}
].
```

3. Mettez à jour le service Cinder v2.

Effectuez ces étapes sur le noeud où Keystone est en cours d'exécution. Un exemple de sortie est inclus pour chaque commande que vous exécutez.

a. Créez le service Cinder v2.

```
controller# openstack --os-url http://$CONTROLLER_ADMIN_NODE:35357/v2.0 \
  --os-token ADMIN \
  service create --name cinderv2 \
```

```

--description "Cinder Volume Service v2" volumev2
+-----+-----+
| Field      | Value                               |
+-----+-----+
| description | Cinder Volume Service v2          |
| enabled     | True                                |
| id          | 2ee6fefbdc4f06bcb0e36e0e4dd9c3    |
| name        | cinderv2                            |
| type        | volumev2                            |
+-----+-----+

```

b. Créez les points d'extrémité.

```

controller# openstack --os-url http://$CONTROLLER_ADMIN_NODE:35357/v2.0 \
--os-token ADMIN
endpoint create \
--region RegionOne \
--publicurl "http://$CONTROLLER_ADMIN_NODE:8776/v2/\$(tenant_id)s" \
--adminurl "http://$CONTROLLER_ADMIN_NODE:8776/v2/\$(tenant_id)s" \
--internalurl "http://$CONTROLLER_ADMIN_NODE:8776/v2/\$(tenant_id)s" cinderv2

```

```

+-----+-----+
| Field      | Value                               |
+-----+-----+
| adminurl   | http://controller-node:8776/v2/\$(tenant_id)s |
| id         | 1b8cd962b12342429cdedb0c7e5d0440          |
| internalurl | http://controller-node:8776/v2/\$(tenant_id)s |
| publicurl  | http://controller-node:8776/v2/\$(tenant_id)s |
| region     | RegionOne                                  |
| service_id | 2ee6fefbdc4f06bcb0e36e0e4dd9c3          |
| service_name | cinderv2                                    |
| service_type | volumev2                                    |
+-----+-----+

```

c. Vérifiez que cinderv2 existe dans la liste des points d'extrémité :

```

controller# openstack --os-url http://$CONTROLLER_ADMIN_NODE:35357/v2.0 --os-token
ADMIN endpoint list

```

```

+-----+-----+-----+-----+
| ID                               | Region | Service Name | Service Type |
+-----+-----+-----+-----+
| 6891354066f84268968c8498f5f6d51b | RegionOne | neutron      | network      |
| 03121908d41e4efa98748fde8ca6d057 | RegionOne | heat         | orchestration |
| b69e4f0373ff4a8f9560dc2644d891ba | RegionOne | glance       | image        |
| 1e6c7f52dcd34a27b7ccac98918f19f1 | RegionOne | ec2          | ec2          |
| e3236915a3dd4098b9e8e0853b5a5af2 | RegionOne | keystone     | identity     |
| fe8870c3e6ac4b529aa7ce7563fc24a4 | RegionOne | heat-cfn     | cloudformation |
| aa931a795f2c4c0ca637e0e4c351cf07 | RegionOne | swift        | object-store |
| 1b8cd962b12342429cdedb0c7e5d0440 | RegionOne | cinderv2     | volumev2     |
| 618a8edba487417c91d0de7f3bcc786d | RegionOne | cinder       | volume       |
| 4c79d020189a44d383bdc15033a942c4 | RegionOne | nova         | compute      |
+-----+-----+-----+-----+

```

4. Redémarrez le service Apache.

```
# svcadm restart apache24
```

5. Démarrez le service de filtre IP s'il n'est pas en cours d'exécution.

```
controller# svcadm enable -rs ipfilter
```

6. Si vous utilisez ZFSSA pour le stockage, réglez /etc/cinder.cinder.conf pour qu'il utilise une nouvelle propriété de pilote.

La propriété `zfssa_initiator_config` répertorie plusieurs initiateurs, ou plusieurs groupes d'initiateurs, et remplace `zfssa_initiator_group` qui est obsolète dans la version Kilo d'OpenStack.

a. Répertoriez plusieurs initiateurs pour la nouvelle propriété, au format suivant :

```
zfssa_initiator_config = {
  'init-grp1': [
    {'iqn': 'iqn1' , 'user': 'user' , 'password': 'password'},
    {'iqn': 'iqn2' , 'user': 'user' , 'password': 'password'}
  ],
  'init-grp2': [
    {'iqn': 'iqn3' , 'user': 'user' , 'password': 'password'}
  ]
}
```

Par exemple, si deux groupes d'initiateurs, Group A et Group B, sont créés sur l'appareil ZFS Storage Appliance, vous les définirez comme suit :

```
zfssa_initiator_config = {
  'GroupA': [
    {'iqn': 'iqn.1986-03.com.sun:01:0a43b9f9dcfd5.570d7fd1' , 'user': 'test1' ,
    'password': 'password1234'},
    {'iqn': 'iqn.1986-03.com.sun:01:0a43b9f9dcfd5.570d7fd2' , 'user': '' ,
    'password': ''}
  ],
  'GroupB': [
    {'iqn': 'iqn.1986-03.com.sun:01:0a43b9f9dcfd5.570d7fd3' , 'user': '' ,
    'password': ''}
  ]
}
```

b. Mettez en commentaire les paramètres obsolètes suivants dans le fichier :

- `zfssa_initiator_group`
- `zfssa_initiator`

c. Redémarrez les services Cinder.

```
controller# svcadm restart cinder-volume:default
```

Installation sur plusieurs systèmes pour une configuration OpenStack multinoeud

Ce chapitre explique comment installer une configuration OpenStack multinoeud. Les thèmes ci-dessous sont abordés :

- ["Présentation de l'architecture à trois noeuds" à la page 23](#)
- ["Etapes préliminaires" à la page 26](#)
- ["Configuration du noeud de contrôleur" à la page 30](#)
- ["Configuration du noeud de calcul" à la page 41](#)
- ["Configuration du noeud de stockage" à la page 46](#)
- ["Configuration du stockage d'objets Swift" à la page 47](#)

Présentation de l'architecture à trois noeuds

Les configurations mononoeuds sont utiles pour tester OpenStack en tant que produit et pour vous familiariser avec ses fonctionnalités. En revanche, une configuration à un seul noeud ne convient pas dans un environnement de production. Pour cet environnement, vous devez installer et configurer OpenStack sur plusieurs systèmes ou noeuds.

Chaque cloud n'a besoin que d'une instance de tableau de bord, d'un magasin d'images et d'un service d'identité. Chaque cloud peut avoir n'importe quel nombre d'instances de stockage et de calcul. Évaluez chaque composant par rapport à vos besoins pour un déploiement cloud particulier afin de déterminer si ce composant doit être installé sur un noeud distinct, ainsi que le nombre de noeuds de chaque type dont vous avez besoin.

L'architecture décrite dans ce chapitre est déployée sur les trois systèmes suivants :

- Noeud de contrôleur – Noeud où la plus grande partie des services OpenStack partagés et d'autres outils sont exécutés. Le noeud de contrôleur fournit les API, la planification et d'autres services partagés pour le Cloud. Le noeud de contrôleur contient le tableau de bord, le magasin d'images et le service d'identité. En outre, le service de gestion des calculs Nova ainsi que le serveur Neutron sont également configurés dans ce noeud.

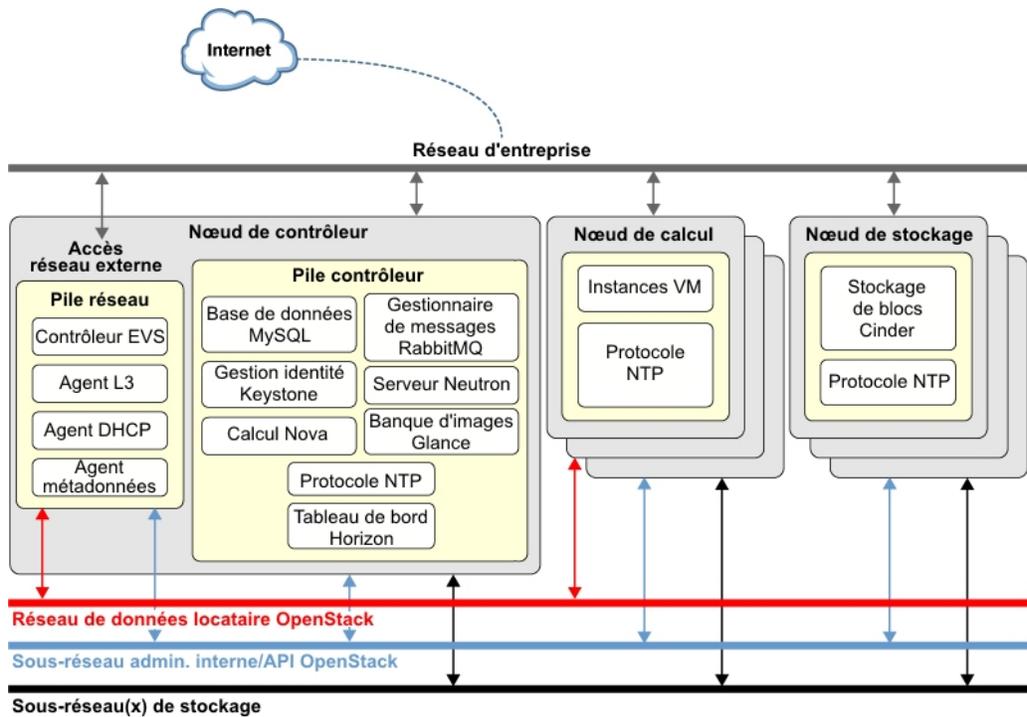
- Noeud de calcul – Noeud où les instances de machine virtuelle (aussi appelées instances de calcul Nova) sont installées. Ce noeud exécute le démon de calcul qui gère les instances de machine virtuelle concernées.
- Noeud de stockage – Noeud qui héberge les données.

Cette architecture à trois noeuds est une façon parmi d'autres de déployer OpenStack sur plusieurs systèmes. Grâce à la flexibilité d'OpenStack, vous pouvez distribuer ses composants en utilisant d'autres architectures. Ainsi, vous devez planifier la configuration du cloud avant de commencer l'installation. Pour plus d'informations sur la planification, reportez-vous à la rubrique [Planification d'une configuration OpenStack](#).

Remarque - Pour partitionner un serveur Oracle SPARC et effectuer une configuration multinoeud d'OpenStack sur le serveur qui exécute OVM Server for SPARC (LDMs), reportez-vous à [Multi-node Solaris 11.2 OpenStack on SPARC Servers](#). Cet article concerne spécifiquement la version Havana d'OpenStack. Toutefois, les grandes lignes de la procédure s'appliquent également à la version actuelle.

La figure suivante illustre une vue de haut niveau de l'architecture décrite dans ce chapitre.

FIGURE 1 Architecture de référence de la configuration à trois noeuds

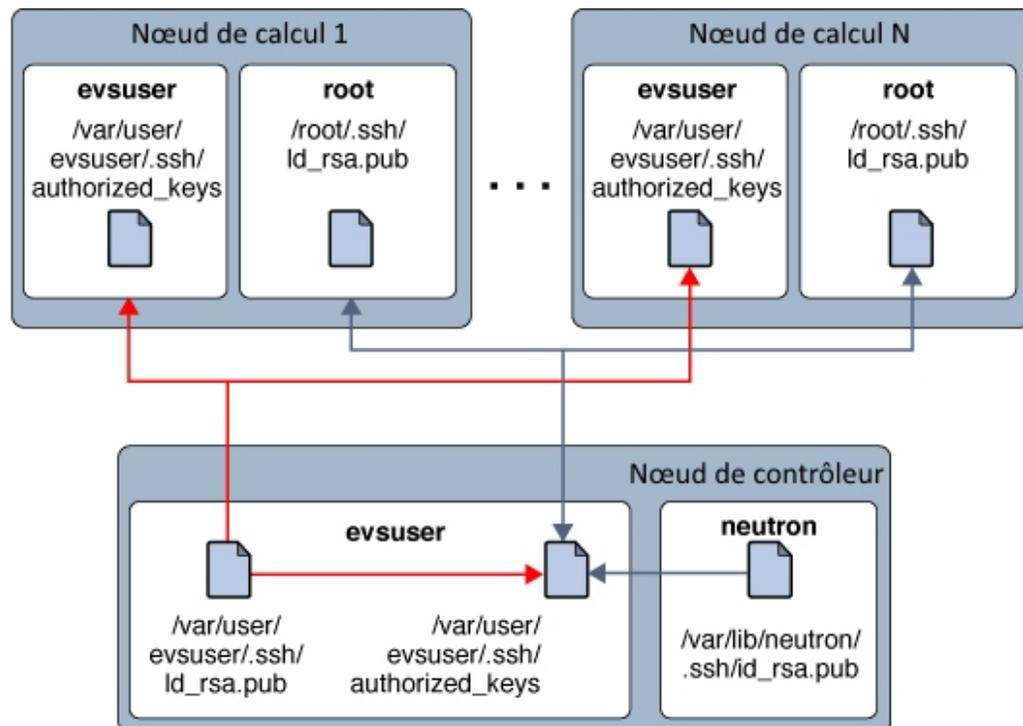


Cette figure utilise Cinder pour illustrer le noeud de stockage. Toutefois, le service de stockage d'objets Swift peut être configuré de manière similaire.

Dans Oracle Solaris, le commutateur virtuel élastique (EVS) constitue le back-end du réseau OpenStack. Il facilite la communication entre les instances de machine virtuelle situées sur des réseaux VLAN ou VXLAN. Les instances de machine virtuelle peuvent se trouver sur le même noeud de calcul ou être réparties entre plusieurs noeuds de calcul. Pour plus d'informations sur les commutateurs virtuels élastiques (EVS), consultez le manuel *Gestion de la virtualisation réseau et des ressources réseau*. Ce manuel figure dans la bibliothèque correspondant à votre version d'Oracle Solaris dans la page [Operating Systems Documentation](#).

Pour que les différents noeuds communiquent entre eux, il faut que les clés publiques SSH de `evsuser`, `neutron` et `root` dans le noeud de contrôleur soient présentes dans chaque fichier `authorized_keys` de l'utilisateur `evsuser` dans tous les noeuds de calcul configurés. Reportez-vous à l'image suivante qui illustre la répartition des clés publiques SSH. L'image suppose que plusieurs noeuds de calcul ont été configurés.

FIGURE 2 Répartition des clés SSH du contrôleur EVS



Pour une liste des paramètres de configuration {ENT OST} utiles pour les déploiements d'{ENT OST} sur les systèmes Oracle Solaris, voir [Annexe A, Fichiers de configuration et services OpenStack courants](#).

Etapas préliminaires

Cette section présente certains éléments à prendre en considération avant d'implémenter la configuration OpenStack multinoeud.

Préparation des noms d'hôtes, variables et mots de passe

Dans une configuration multinoeud, vous utilisez plusieurs interfaces réseau dédiées aux différents sous-réseaux que vous créez pour le cloud. Assurez-vous que vous avez préparé les noms d'hôte de ces interfaces. Incluez ces noms et leurs adresses IP dans le fichier `/etc/hosts` des systèmes ou dans la configuration DNS.

Par exemple, vous pouvez créer les noms d'hôte suivants pour gérer différents types de trafic réseau.

- `host-on` pour le réseau OpenStack qui héberge le trafic administratif et des API.
- `host-tn` pour le réseau de projet qui héberge le trafic entre les noeuds de calcul et le routeur L3.
- `host-en` pour le trafic réseau externe.

Lorsque vous configurez les services OpenStack dans des noeuds différents, créez des variables pour faciliter la tâche, comme dans les exemples suivants :

- `$CONTROLLER_ADMIN_NODE` - Nom d'hôte de l'interface ou adresse IP (dans le noeud de contrôleur) à laquelle les services administratifs OpenStack sont rattachés.
- `$CONTROLLER_ADMIN_NODE_IP` - Adresse IP du port de contrôleur qui gère le trafic et les services administratifs OpenStack.
- `$COMPUTE_ADMIN_NODE_IP` - Adresse IP du port de calcul qui gère le trafic et les services administratifs OpenStack.
- `$VOLUME_IP` - Nom d'hôte du noeud de contrôleur.

Les mots de passe sont également requis dans les processus de configuration. Voici un exemple de liste des mots de passe que vous devez préparer.

- Mot de passe `root` pour la base de données MySQL
- Mot de passe de l'utilisateur `admin`
- Mots de passe de base de données des services OpenStack :

- Service de gestion de l'identité
- Service d'image
- Service de calcul
- Base de données de tableau de bord
- Base de données de stockage par bloc
- Base de données de mise en réseau
- Base de données d'orchestration
- Mots de passe des utilisateurs des services OpenStack :
 - glance
 - nova
 - cinder
 - neutron
 - heat

Remarque - Vous pouvez également attribuer un mot de passe commun pour un groupe d'utilisateurs ou de services. Quel que soit le système utilisé pour l'attribution des mots de passe, veuillez à respecter les meilleures pratiques pour sécuriser votre environnement. Reportez-vous à la section *Sécurisation des systèmes et des périphériques connectés* dans la bibliothèque correspondant à votre version d'Oracle Solaris dans la [Operating Systems Documentation](#).

Exemple de script Keystone

Pour alimenter rapidement la base de données Keystone, vous pouvez utiliser l'exemple de script `/usr/demo/openstack/keystone/sample_data.sh`. Pour vous aider à démarrer, le script effectue les tâches de base suivantes :

- Il crée les projets initiaux suivants :
 - `service` sous lequel les services de base ou principaux sont créés.
 - `demo` dans lequel l'utilisateur `admin` est créé avec le mot de passe par défaut `secrete`.
- Il alimente la base de données Keystone.
- Crée les principaux services suivants :
 - `cinder`
 - `cinder2`
 - `ec2`
 - `glance`
 - `keystone`
 - `neutron`
 - `nova`

- swift
- heat

Des noms d'utilisateur et des mots de passe correspondent à chaque service, à l'exception du service Keystone, pour lequel aucun utilisateur n'est créé. Par défaut, les noms d'utilisateur, les mots de passe et les noms de service sont identiques. Par exemple, pour le service Cinder, le nom d'utilisateur et le mot de passe est `cinder`, `glance` pour le service Glance, etc. Vous pouvez éventuellement créer des mots de passe personnalisés pour remplacer ces mots de passe par défaut dans le script. Vous pouvez également définir un seul mot de passe pour tous les services dans le script. Apportez toutes les modifications nécessaires au script avant de l'exécuter pour démarrer Keystone.

Remarque - Veillez à vérifier le script pour plus d'informations sur les paramètres qui peuvent être définis pour l'environnement considéré. Remplacez les paramètres par défaut dans le script, selon vos préférences.

Dans l'ensemble de ce document, les procédures supposent que, à l'exception des mots de passe, l'exemple du script de données soit utilisé sans révision et que tous les paramètres par défaut du script soient appliqués à la configuration cloud.

A propos de la modification des fichiers de configuration

Une grande partie de la configuration d'OpenStack concerne la modification des fichiers de configuration des composants. Dans ce document, seuls les paramètres sélectionnés sont identifiés pour la configuration dans chaque fichier `*.conf` ou `*.ini`. Ces paramètres sélectionnés sont le minimum requis pour que la configuration cloud fonctionne. Toutefois, vérifiez l'intégralité du contenu de chaque fichier de configuration pour garantir que tous les paramètres propres à votre configuration cloud spécifique sont correctement configurés.

Optimisation de l'utilisation de la mémoire

Pour une meilleure gestion de l'utilisation de la mémoire entre ZFS et les applications dans Oracle Solaris 11, définissez le paramètre `usr_reserve_hint_pct` sur le noeud, comme indiqué dans l'exemple suivant :

```
# echo "set user_reserve_hint_pct=80" >>/etc/system.d/site:kernel-zones-reserve
# reboot
```

où `site` peut correspondre à votre entreprise.

Définissez également ce paramètre sur les différents noeuds OpenStack.

Pour plus d'informations sur ce paramètre, connectez-vous à votre compte MOS à l'adresse <https://support.oracle.com> et consultez le Document 1663862.1, *Memory Management Between ZFS and Applications in Oracle Solaris 11.2*.

Configuration du serveur NTP

L'installation de NTP (Network Time Protocol) est facultative, mais vivement recommandée. NTP permet de garantir la cohérence de l'heure du jour sur tous les noeuds du cloud. Si vous activez NTP dans un réseau, configurez les noeuds de service pour obtenir l'heure sur laquelle ils sont sur le réseau.

- Si la multidiffusion est activée sur les sous-réseaux IP sur lesquels résident les noeuds de service, vous pouvez tirer parti du réseau IP pour utiliser la multidiffusion afin de configurer NTP.
- Si la multidiffusion n'est pas activée sur les sous-réseaux IP sur lesquels résident les noeuds de service, configurez NTP manuellement.

L'utilisation du protocole NTP implique la configuration du serveur et du client NTP. En général, le serveur NTP est un système distinct des autres systèmes sur lesquels vous configurez OpenStack. Le client NTP est installé et configuré sur les noeuds ou systèmes qui hébergent les composants OpenStack.

Pour plus d'informations sur le protocole NTP, reportez-vous à la documentation à l'adresse <http://www.ntp.org/documentation.html>.

▼ Configuration du serveur NTP

Le serveur NTP se trouve sur un système distinct des noeuds OpenStack.

1. Installez le package NTP.

```
ntp-server# pkg install ntp
```

2. Installez le fichier de configuration.

```
ntp-server# cp /etc/inet/ntp.server /etc/inet/ntp.conf
```

3. Modifiez le fichier `/etc/inet/ntp.conf` en configurant les mots-clés `server` et `driftfile`.

Par exemple :

```
server 127.127.1.0 prefer
...
```

```
driftfile /var/ntp/ntp.drift
```

Remarque - 127.127.1.0 n'est **pas** une adresse IP. Il s'agit d'un format utilisé pour faire référence à une horloge qui fournit l'heure exacte au serveur. Assurez-vous de lire les commentaires dans le fichier `ntp.conf`, qui explique le mot-clé `server`.

4. **Créez le fichier `/var/ntp/ntp.drift` comme vous l'avez défini dans l'étape précédente.**

```
ntp-server# touch /var/ntp/ntp.drift
```

5. **Démarrez le service `ntp`.**

```
ntp-server# svcadm enable ntp
```

Configuration du noeud de contrôleur

Le noeud de contrôleur comporte un service de tableau de bord, un magasin d'images et un service d'identité. Ce noeud inclut également MySQL, RabbitMQ, ainsi que des services de calcul, de stockage de bloc et de mise en réseau.

Pour configurer le noeud de contrôleur, installez les composants et services OpenStack sur le système à l'aide de la commande suivante :

```
controller# pkg install openstack
```

Une fois l'installation du package terminée, configurez les services à exécuter sur le noeud. La liste suivante décrit les tâches à accomplir pour configurer le noeud de contrôleur :

- ["Configuration du client NTP" à la page 30](#)
- ["Installation de MySQL" à la page 31](#)
- ["Installation de Keystone" à la page 32](#)
- ["Installation de Glance" à la page 33](#)
- ["Installation de Nova" à la page 35](#)
- ["Installation d'Horizon" à la page 36](#)
- ["Installation de Cinder" à la page 37](#)
- ["Installation de Neutron" à la page 38](#)

Configuration du client NTP

Vous installez le service client NTP sur chaque noeud de service de votre déploiement cloud.

▼ Configuration du client NTP

Cette section part du principe que vous avez déjà configuré le serveur NTP comme indiqué à la section "[Configuration du serveur NTP](#)" à la page 29.

1. Créez le fichier de configuration client.

```
controller# cp /etc/inet/ntp.client /etc/inet/ntp.conf
```

2. Dans le fichier de configuration client, supprimez les marques de commentaire d'une ou de plusieurs options serveur et indiquez le nom spécifique ou l'adresse IP du serveur NTP.

Par exemple, si le nom d'hôte du serveur NTP que vous avez configuré est `system1`, le fichier de configuration ressemble à l'exemple suivant :

```
# multicastclient 224.0.1.1
...
server system1.example.com iburst
# server server_name2 iburst
# server server_name3 iburst
```

3. Activez le service `ntp`.

```
controller# svcadm enable ntp
```

Installation de MySQL

De nombreux services OpenStack permettent à une base de données d'assurer le suivi des ressources critiques, leur utilisation et d'autres informations. Pour stocker ces informations, utilisez des bases de données telles que la base de données MySQL, en particulier pour les configurations multinoeuds.

▼ Installation d'une base de données MySQL

1. Activez les services RabbitMQ.

```
controller# svcadm enable rabbitmq
controller# svcadm restart rad:local
```

2. (Facultatif) Si vous utilisez une adresse IP dédiée pour le trafic d'administration et d'API, ajoutez-la dans le fichier `/etc/mysql/5.5/my.cnf` :

```
bind-address=${CONTROLLER_ADMIN_NODE_IP}
```

3. Activez le service MySQL.

```
controller# svcadm enable mysql
```

4. Définissez le mot de passe root du serveur MySQL.

```
controller# mysqladmin -u root password MySQL-root-password
```

5. Configurez MySQL.

Créez les tables qui seront utilisées par OpenStack. Accordez des privilèges aux services sur le noeud de contrôleur afin de fournir un accès exclusif à ces bases de données.

```
controller# mysql -u root -p
Enter password: MySQL-root-password
mysql> drop database if exists nova;
mysql> drop database if exists cinder;
mysql> drop database if exists glance;
mysql> drop database if exists keystone;
mysql> drop database if exists neutron;
mysql> drop database if exists heat;
mysql> create database cinder default character set utf8 default collate utf8_general_ci;
mysql> grant all privileges on cinder.* to 'cinder'@'$CONTROLLER_ADMIN_NODE' identified
  by 'service-password';
mysql> create database glance default character set utf8 default collate utf8_general_ci;
mysql> grant all privileges on glance.* to 'glance'@'$CONTROLLER_ADMIN_NODE' identified
  by 'service-password';
mysql> create database keystone default character set utf8 default collate
  utf8_general_ci;
mysql> grant all privileges on keystone.* to 'keystone'@'$CONTROLLER_ADMIN_NODE'
  identified by 'service-password';
mysql> create database nova default character set utf8 default collate utf8_general_ci;
mysql> grant all privileges on nova.* to 'nova'@'$CONTROLLER_ADMIN_NODE' identified by
  'service-password';
mysql> create database neutron default character set utf8 default collate
  utf8_general_ci;
mysql> grant all privileges on neutron.* to 'neutron'@'$CONTROLLER_ADMIN_NODE' identified
  by 'service-password';
mysql> create database heat default character set utf8 default collate utf8_general_ci;
mysql> grant all privileges on heat.* to 'heat'@'$CONTROLLER_ADMIN_NODE' identified by
  'service-password';
mysql> flush privileges;
mysql> quit
```

Installation de Keystone

Le service Keystone doit être installé et configuré sur le noeud de contrôleur. Cette procédure utilise l'exemple de script décrit dans ["Exemple de script Keystone" à la page 27](#). Consultez la section avant d'utiliser le script.

▼ Installation et configuration de Keystone

1. Créez le jeton partagé pour Keystone et les autres services OpenStack.

Ce jeton se compose d'une chaîne aléatoire de caractères. Notez que la commande `openssl` vous invite à saisir les composants qui constituent la clé, par exemple le pays, l'Etat, etc.

```
controller# openssl rand -hex 10
token-string
```

2. Configurez le jeton pour une variable shell.

```
controller# export MY_SERVICE_TOKEN=token-string
```

où `token-string` représente la sortie générée par la commande de l'étape précédente.

3. Modifiez les paramètres du fichier `/etc/keystone/keystone.conf`.

La configuration doit ressembler à l'exemple suivant .

```
[DEFAULT]
admin_token = token-string
...
[database]
connection = mysql://keystone:service-password@$CONTROLLER_ADMIN_NODE/keystone

[oslo_messaging_rabbit]
rabbit_host=$CONTROLLER_ADMIN_NODE
```

4. Activez le service MSF Keystone.

```
controller# svcadm enable keystone
```

5. Alimentez la base de données Keystone à l'aide de l'exemple de script Keystone.

Veillez à vérifier et modifier le script en fonction de vos préférences avant de l'exécuter. Les procédures supposent que l'exemple de script n'est pas personnalisé.

```
controller# CONTROLLER_PUBLIC_ADDRESS=$CONTROLLER_ADMIN_NODE \
CONTROLLER_ADMIN_ADDRESS=$CONTROLLER_ADMIN_NODE \
CONTROLLER_INTERNAL_ADDRESS=$CONTROLLER_ADMIN_NODE \
SERVICE_TOKEN=$MY_SERVICE_TOKEN \
ADMIN_PASSWORD=admin-password \
SERVICE_PASSWORD=service-password \
/usr/demo/openstack/keystone/sample_data.sh
```

Installation de Glance

Glance requiert la configuration de certaines informations d'authentification ainsi que des informations de localisation des services MySQL et RabbitMQ.

▼ Configuration et installation de Glance

1. Configurez Glance en éliminant des marques de commentaire ou en définissant des paramètres dans les fichiers de configuration suivants :

- /etc/glance/glance-api.conf

```
[DEFAULT]
registry_host = $CONTROLLER_ADMIN_NODE

auth_strategy = keystone
default_publisher_id = image.$CONTROLLER_ADMIN_NODE

[database]
connection = mysql://glance:service-password@$CONTROLLER_ADMIN_NODE/glance

[keystone_authtoken]
auth_uri = http://$CONTROLLER_ADMIN_NODE:5000/v2.0
identity_uri = http://$CONTROLLER_ADMIN_NODE:35357
admin_user = glance
admin_password = service-password
admin_tenant_name = service

[oslo_messaging_rabbit]
rabbit_host=$CONTROLLER_ADMIN_NODE
```

- /etc/glance/glance-cache.conf

```
[DEFAULT]
auth_url = http://$CONTROLLER_ADMIN_NODE:5000/v2.0/
admin_user = glance
admin_password = service-password
admin_tenant_name = service
```

- /etc/glance/glance-registry.conf

```
[DEFAULT]
default_publisher_id = image.$CONTROLLER_ADMIN_NODE

[database]
connection = mysql://glance:service-password@$CONTROLLER_ADMIN_NODE/glance

[keystone_authtoken]
auth_uri = http://$CONTROLLER_ADMIN_NODE:5000/v2.0
identity_uri = http://$CONTROLLER_ADMIN_NODE:35357
admin_user = glance
admin_password = service-password
admin_tenant_name = service
```

```

[oslo_messaging_rabbit]
rabbit_host=${CONTROLLER_ADMIN_NODE}

```

- /etc/glance/glance-scrubber.conf

```

[DEFAULT]
auth_url = http://${CONTROLLER_ADMIN_NODE}:5000/v2.0/
identity_uri = http://${CONTROLLER_ADMIN_NODE}:35357
admin_user = glance
admin_password = service-password
admin_tenant_name = service

[database]
connection=mysql://glance:service-password@${CONTROLLER_ADMIN_NODE}/glance

```

2. Activez les services SMF Glance.

```
controller# svcadm enable -rs glance-api glance-db glance-registry glance-scrubber
```

Installation de Nova

Cette section traite de la configuration des services de point d'extrémité Nova et non du noeud de calcul.

▼ Installation et configuration de Nova

1. Configurez Nova en éliminant des marques de commentaire ou en définissant les paramètres du fichier /etc/nova/nova.conf .

```

[DEFAULT]
my_ip=${CONTROLLER_ADMIN_NODE_IP}
host=${CONTROLLER_ADMIN_NODE}
firewall_driver=nova.virt.firewall.NoopFirewallDriver

[database]
connection = mysql://nova:service-password@${CONTROLLER_ADMIN_NODE}/nova

[glance]
host=${CONTROLLER_ADMIN_NODE}

[keystone_authtoken]
auth_uri=http://${CONTROLLER_ADMIN_NODE}:5000/v2.0/
identity_uri=http://${CONTROLLER_ADMIN_NODE}:35357/
admin_user=nova
admin_password=service-password
admin_tenant_name=service

```

```
[neutron]
url=http://$CONTROLLER_ADMIN_NODE:9696
admin_username=neutron
admin_password=service-password
admin_tenant_name=service
admin_auth_url=http://$CONTROLLER_ADMIN_NODE:5000/v2.0

[oslo_messaging_rabbit]
rabbit_host=$CONTROLLER_ADMIN_NODE
```

2. Définissez les paramètres dans le fichier `/etc/nova/api-paste.ini`.

```
[filter:authtoken]
admin_user = nova
admin_password = service-password
admin_tenant_name = service
auth_uri = http://$CONTROLLER_ADMIN_NODE:5000/v2.0/
identity_uri = http://$CONTROLLER_ADMIN_NODE:35357
```

3. Activez les services SMF Nova.

```
controller# svcadm enable -rs nova-conductor
controller# svcadm enable -rs nova-api-osapi-compute \
nova-cert nova-scheduler
```

Installation d'Horizon

Horizon joue le rôle de portail Web pour OpenStack.

▼ Configuration d'Horizon

1. Définissez la configuration Horizon pour SSL/TLS.

a. Générez les certificats à utiliser par Horizon.

Les commandes suivantes génèrent des certificats auto-signés pour Horizon et copient le fichier de configuration du tableau de bord OpenStack sur le répertoire de fichiers de configuration Apache. Pour plus d'informations sur la création de certificats autosignés, reportez-vous à la foire aux questions Apache : [SSL/TLS Strong Encryption: FAQ](#).

```
controller# export DASHBOARD=/etc/openstack_dashboard
controller# openssl req -new -x509 -nodes \
-out horizon.crt -keyout horizon.key
```

A ce stade, indiquez les informations demandées, telles que le pays, l'état, la ville, la société, l'organisation, le nom et l'adresse électronique. Passez ensuite au déplacement de la clé.

```

controller# mv horizon.crt horizon.key ${DASHBOARD}
controller# chmod 0644 ${DASHBOARD}/*
controller# chown websrvd:websrvd ${DASHBOARD}/*

controller# sed \
-e "/SSLCertificateFile/s:/path.*:${DASHBOARD}/horizon.crt:" \
-e "/SSLCACertificateFile/d" \
-e "/SSLCertificateKeyFile/s:/path.*:${DASHBOARD}/horizon.key:" \
< /etc/apache2/2.4/samples-conf.d/openstack-dashboard-tls.conf \
> /etc/apache2/2.4/conf.d/openstack-dashboard-tls.conf

```

- b. Dans le fichier `/etc/apache2/2.4/conf.d/openstack-dashboard-tls.conf`, indiquez l'adresse du site et le nom de serveur du package Horizon dans les paramètres suivants :

```

RedirectPermanent /horizon https://controller-fqdn/horizon
ServerName controller-fqdn

```

2. Démarrez le service Apache.

```
controller# systemctl enable apache24
```

Installation de Cinder

La configuration Cinder doit indiquer au moins les informations suivantes :

- Informations d'autorisations pour authentification auprès de Keystone.
- Classe des volumes à créer.

▼ Installation et configuration de Cinder

Les étapes de cette procédure concernent la configuration des services de point d'extrémité Cinder, pas le noeud Cinder ni le noeud de volume.

1. Configurez Cinder en éliminant les marques de commentaire ou en définissant les paramètres appropriés dans le fichier `/etc/cinder/cinder.conf`.

Pour le paramètre `volume_driver`, plusieurs pilotes sont à votre disposition pour la sélection. Dans l'exemple suivant, seul le pilote sélectionné pour `volume_driver` s'affiche. Les autres pilotes disponibles qui sont commentés sont exclus.

```

[DEFAULT]
volume_driver=cinder.volume.drivers.solaris.zfs.ZFSISCSIDriver
my_ip=${CONTROLLER_ADMIN_NODE}

[database]

```

```
connection = mysql://cinder:service-password@$CONTROLLER_ADMIN_NODE/cinder
```

```
[keystone_authtoken]
auth_uri = http://$CONTROLLER_ADMIN_NODE:5000/v2.0
identity_uri = http://$CONTROLLER_ADMIN_NODE:35357
admin_user = cinder
admin_password = service-password
admin_tenant_name = service
```

```
[oslo_messaging_rabbit]
rabbit_host=$CONTROLLER_ADMIN_NODE
```

2. Configurez les paramètres dans le fichier `/etc/cinder/api-paste.ini`.

```
[filter:authtoken]
admin_tenant_name = service
admin_user = cinder
admin_password = service-password
```

3. Si les cibles iSCSI sont configurées, activez les services SMF correspondants.

```
controller# svcadm enable iscsi/target stmf
```

4. Activez les services SMF Cinder.

```
controller# svcadm enable -rs cinder-db
controller# svcadm enable -rs cinder-api cinder-scheduler
```

Voir aussi Reportez-vous également à [How to Build OpenStack Block Storage on ZFS](#).

Installation de Neutron

Dans l'architecture décrite dans ce chapitre, le service de l'API Neutron s'exécute sur le noeud de contrôleur.

▼ Installation et configuration de Neutron

1. Configurez Neutron en éliminant des marques de commentaire ou en définissant des paramètres dans les fichiers de configuration suivants :

- `/etc/neutron/neutron.conf`

```
[DEFAULT]
host=$CONTROLLER_ADMIN_NODE

[keystone_authtoken]
auth_uri = http://$CONTROLLER_ADMIN_NODE:5000/v2.0
```

```

identity_uri = http://$CONTROLLER_ADMIN_NODE:35357
admin_user = neutron
admin_password = service-password
admin_tenant_name = service

[database]
connection = mysql://neutron:service-password@$CONTROLLER_ADMIN_NODE/neutron

[oslo_messaging_rabbit]
rabbit_host=$CONTROLLER_ADMIN_NODE

```

- /etc/neutron/plugins/evs/evs_plugin.ini

```

[EVS]
evs_controller = ssh://evsuser@$CONTROLLER_ADMIN_NODE

```

- /etc/neutron/dhcp_agent.ini

```

[DEFAULT]
evs_controller = ssh://evsuser@$CONTROLLER_ADMIN_NODE

```

- /etc/neutron/l3_agent.ini

```

evs_controller = ssh://evsuser@$CONTROLLER_ADMIN_NODE

```

Remarque - Par défaut, les réseaux internes de projets dans le cloud sont isolés les uns des autres. Les réseaux d'un projet peuvent uniquement communiquer entre eux, pas avec les réseaux d'autres projets. Pour permettre à tous les réseaux du cloud de se connecter les uns aux autres, quel que soit le projet auquel ils appartiennent, modifiez le fichier `/etc/neutron/l3_agent.ini` à l'aide de la définition de paramètre suivante :

```
allow_forwarding_between_networks = true
```

2. Configurez les paires de clés SSH à utiliser.

a. Créez les paires de clés SSH pour les utilisateurs `evsuser`, `neutron` et `root`.

```

controller# su - evsuser -c "ssh-keygen -N '' \
-f /var/user/evsuser/.ssh/id_rsa -t rsa"
controller# su - neutron -c "ssh-keygen -N '' -f /var/lib/neutron/.ssh/id_rsa -t rsa"
controller# ssh-keygen -N '' -f /root/.ssh/id_rsa -t rsa

```

b. Combinez les clés SSH des utilisateurs `evsuser`, `neutron` et `root` dans le fichier `authorized_keys` d'`evsuser`.

```

controller# cat /var/user/evsuser/.ssh/id_rsa.pub \
/var/lib/neutron/.ssh/id_rsa.pub /root/.ssh/id_rsa.pub >> \
/var/user/evsuser/.ssh/authorized_keys

```

c. Testez les connexions SSH de façon à accepter les empreintes qui seront enregistrées dans le fichier `known_host`.

Sélectionnez Yes pour chaque invite de confirmation.

```
controller# su - evsuser -c "ssh evsuser@$CONTROLLER_ADMIN_NODE true"
controller# su - neutron -c "ssh evsuser@$CONTROLLER_ADMIN_NODE true"
controller# ssh evsuser@$CONTROLLER_ADMIN_NODE true
```

3. Configurez le commutateur virtuel élastique (EVS).

Remarque - Les opérations décrites ci-après permettent de configurer le commutateur EVS pour un réseau basé sur un VLAN.

Pour savoir comment configurer un réseau basé sur un VXLAN, consultez le manuel *Gestion de la virtualisation réseau et des ressources réseau dans Oracle Solaris* dans la bibliothèque correspondant à votre version de la page [Operating Systems Documentation](#). Dans ce manuel, lisez tout particulièrement la section *Cas d'utilisation : configuration d'un commutateur virtuel élastique pour un locataire*.

Un exemple de configuration de réseaux plats est décrit dans la page https://blogs.oracle.com/openstack/entry/configuring_the_neutron_l3_agent.

a. Définissez la propriété EVS qui indique l'emplacement du contrôleur EVS.

```
controller# evsadm set-prop -p controller=ssh://evsuser@$CONTROLLER_ADMIN_NODE
```

b. Configurez les propriétés `l2-type`, `vlan-range` et `uplink-port` du contrôleur EVS.

```
controller# evsadm set-controlprop -p property=value
```

Comme présenté dans [Figure 1, "Architecture de référence de la configuration à trois noeuds"](#), vous verrez généralement plusieurs interfaces réseau pour servir les différents sous-réseaux. Lorsque vous définissez la propriété `uplink-port`, vous pouvez fractionner le réseau VLANS sur les ports réseau multiples dédiés aux sous-réseaux.

Les exemples suivants illustrent comment configurer les propriétés EVS, notamment le fractionnement des VLAN. Vous pouvez éventuellement utiliser la commande finale pour afficher toutes les propriétés EVS.

Remarque - Définissez d'abord la plage de VLAN avant de répartir les VLAN sur les ports réseau. Sinon, vous ne pouvez pas configurer la propriété `uplink-port`.

```
controller# evsadm set-controlprop -p l2-type=vlan
controller# evsadm set-controlprop -p vlan-range=1,200-300
controller# evsadm set-controlprop -p uplink-port=net0,vlan-range=1
```

```

controller# evsadm set-controlprop -p uplink-port=net1,vlan-range=200-250
controller# evsadm set-controlprop -p uplink-port=net2,vlan-range=251-300

controller# evsadm show-controlprop -o all

```

4. Activez le transfert IP.

```

controller# ipadm set-prop -p forwarding=on ipv4
controller# ipadm set-prop -p forwarding=on ipv6

```

5. Démarrez le service de filtre IP.

```

controller# svcadm enable -rs ipfilter

```

6. Activez le service de serveur Neutron.

```

controller# svcadm enable -rs neutron-server neutron-dhcp-agent

```

Configuration du noeud de calcul

Vous installez les instances de machine virtuelle (VM) dans le noeud de calcul ainsi que le démon nova-compute. Les instances VM fournissent un large éventail de services, notamment des applications Web et des fonctionnalités d'analyse. Vous pouvez configurer autant de noeuds de calcul que nécessaire pour votre Cloud.

Pour configurer le noeud de calcul, installez les composants et services OpenStack sur le système à l'aide de la commande suivante :

```

compute# pkg install openstack

```

Une fois l'installation du package terminée, configurez les services à exécuter sur le noeud.

Remarque - Pour une meilleure gestion de l'utilisation de la mémoire entre ZFS et les applications dans Oracle Solaris 11, définissez le paramètre `usr_reserve_hint_pct` sur le noeud, comme indiqué dans l'exemple suivant :

```

# echo "set user_reserve_hint_pct=80" >>/etc/system.d/site:kernel-zones-reserve
# reboot

```

où *site* est un identificateur aléatoire, par exemple le nom de votre société.

Définissez également ce paramètre sur les différents noeuds OpenStack.

Pour plus d'informations sur ce paramètre, connectez-vous à votre compte MOS à l'adresse <https://support.oracle.com> et consultez le Document 1663862.1, *Memory Management Between ZFS and Applications in Oracle Solaris 11.2*.

▼ Configuration du noeud de calcul

1. Configurez le client NTP.

Voir "[Configuration du client NTP](#)" à la page 30.

2. Redémarrez RAD (Démon d'accès distant).

Nova utilise RAD pour communiquer avec la structure Oracle Solaris Zones.

```
compute1# svcadm restart rad:local
```

3. Configurez Nova en éliminant les marques de commentaire ou en définissant les paramètres suivants dans le fichier `/etc/nova/nova.conf`.

```
[DEFAULT]
my_ip=${COMPUTE_ADMIN_NODE_IP}
host=${COMPUTE_ADMIN_NODE_X}
firewall_driver=nova.virt.firewall.NoopFirewallDriver
keystone_ec2_url=http://${CONTROLLER_ADMIN_NODE:5000}/v2.0/ec2tokens

[database]
connection = mysql://nova:service-password@${CONTROLLER_ADMIN_NODE}/nova

[glance]
host=${CONTROLLER_ADMIN_NODE}

[keystone_authtoken]
auth_uri=http://${CONTROLLER_ADMIN_NODE:5000}/v2.0/
identity_uri=http://${CONTROLLER_ADMIN_NODE:35357}/
admin_user=nova
admin_password=service-password
admin_tenant_name=service

[neutron]
url=http://${CONTROLLER_ADMIN_NODE:9696}
admin_username=neutron
admin_password=service-password
admin_tenant_name=service
admin_auth_url=http://${CONTROLLER_ADMIN_NODE:5000}/v2.0

[oslo_messaging_rabbit]
rabbit_host=${CONTROLLER_ADMIN_NODE}
```

4. Définissez les paramètres dans le fichier `/etc/nova/api-paste.ini`.

```
[filter:authtoken]
admin_user = nova
admin_password = service-password
admin_tenant_name = service
auth_uri = http://${CONTROLLER_ADMIN_NODE:5000}/v2.0/
identity_uri = http://${CONTROLLER_ADMIN_NODE:35357}
```

5. Configurez EVS sur le noeud de calcul.

a. Assurez-vous que le package EVS est installé.

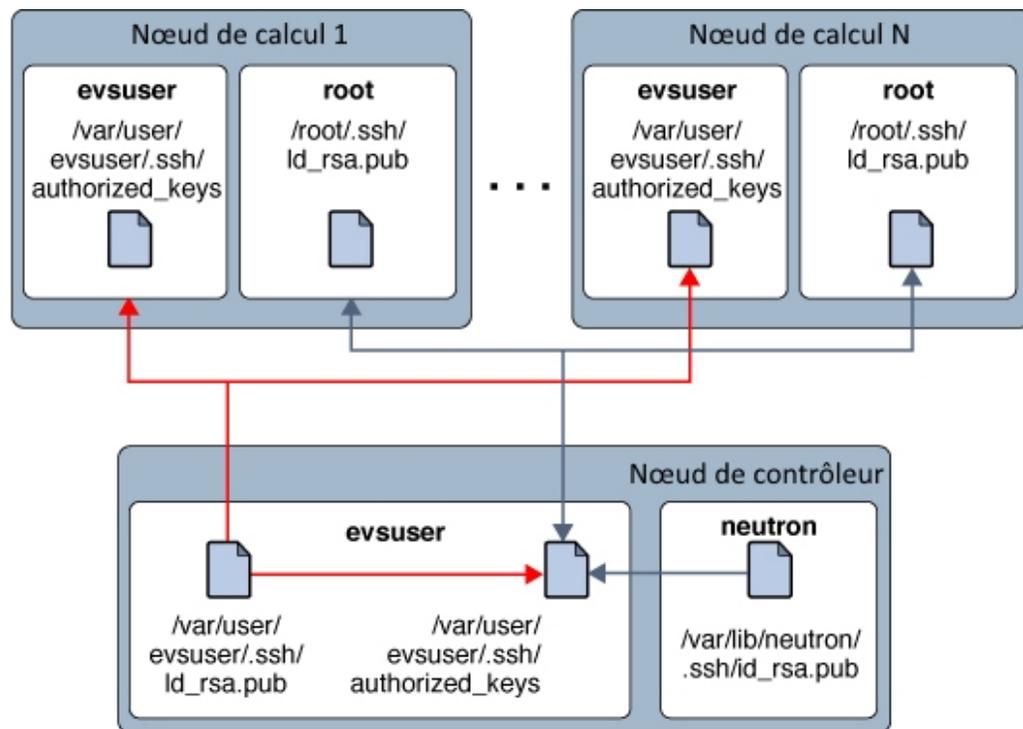
```
compute1# pkg info evs
```

b. Indiquez l'emplacement du contrôleur EVS.

```
compute1# evsadm set-prop -p controller=ssh://evsuser@$CONTROLLER_ADMIN_NODE
```

6. Configurez les communications entre le contrôleur et les noeuds de calcul.

La répartition des clés SSH, qui établit la communication entre les noeuds, se présente comme dans la figure suivante :



a. Créez une clé publique SSH sur le noeud de calcul pour l'utilisateur root.

```
compute1# ssh-keygen -N '' -f /root/.ssh/id_rsa -t rsa
```

b. (Facultatif) Vérifiez le contenu de la clé SSH.

```
compute1# cat /root/.ssh/id_rsa.pub
```

- c. Copiez la clé SSH `/root/.ssh/id_rsa.pub` sur le noeud de contrôleur.
- d. Sur le noeud de contrôleur, ajoutez la clé SSH au fichier `authorized_keys` de l'utilisateur `evsuser`.

```
controller# cat location/id_rsa.pub >> /var/user/evsuser/.ssh/authorized_keys
```

- e. (Facultatif) Vérifiez que la clé SSH en provenance du noeud de calcul a été ajoutée au fichier `authorized_keys`.

```
controller# cat /var/user/evsuser/.ssh/authorized_keys
```

La sortie doit inclure le contenu de `/root/.ssh/id_rsa.pub` que vous avez généré dans le noeud de calcul.

- f. Testez les connexions SSH du noeud de calcul au contrôleur et acceptez le stockage des empreintes dans le fichier `known_host`.

Sélectionnez Yes à l'invite de confirmation.

```
compute1# ssh evsuser@$CONTROLLER_ADMIN_NODE true
```

- 7. Activez le service de calcul Nova.

```
compute1# svcadm enable nova-compute
```

▼ Activation de l'accès à la console

Cette procédure permet d'utiliser une console d'instance de machine virtuelle dans un navigateur en fonction d'une demande de l'utilisateur.

- 1. Effectuez les étapes ci-dessous dans chaque noeud de calcul en fonction du scénario qui s'applique.
 - Si les utilisateurs peuvent accéder aux adresses IP du noeud de calcul dans le réseau orienté public, définissez les paramètres suivants dans la section [DEFAULT] du fichier `/etc/nova/nova.conf`.

```
[DEFAULT]
...
vnc_enabled = true
vncserver_listen = 0.0.0.0
novncproxy_port = 6080
novncproxy_base_url =http://FQDN:6080/vnc_auto.html
```

```
novncproxy_host = 0.0.0.0
...
```

où *FQDN* représente le nom de domaine complet ou l'adresse IP du noeud de calcul.

- **Si le noeud de calcul se trouve dans un réseau privé, définissez les paramètres suivants dans la section [DEFAULT] du fichier `/etc/nova/nova.conf`.**

```
[DEFAULT]
...
vnc_enabled = true
vncserver_listen = internal-IP
novncproxy_port=6080
novncproxy_base_url = http://public-IP:6080/vnc_auto.html
vncserver_proxycient_address = internal-IP
```

- *internal-IP* : adresse IP du noeud de calcul sur le réseau interne.
- *public-IP* : adresse IP publique de l'hôte du contrôleur.

2. **Exécutez les sous-étapes suivantes si les adresses IP du noeud de calcul sont accessibles à partir du réseau orienté public. Sinon, passez à l'étape suivante.**

- a. **Activez le service `nova-novncproxy`.**

```
compute# svcadm enable nova-novncproxy
```

- b. **Redémarrez le service `nova-compute`.**

```
compute# svcadm restart nova-compute
```

3. **Effectuez les étapes ci-dessous dans chaque noeud de contrôleur en fonction du scénario qui s'applique.**

- **Si les utilisateurs peuvent accéder aux adresses IP du noeud de calcul dans le réseau orienté public, activez le service `nova-consoleauth`.**

```
controller# svcadm enable nova-consoleauth
```

- **Si le noeud de calcul se trouve dans un réseau privé, effectuez les étapes suivantes :**

- a. **Définissez les paramètres suivants dans la section [DEFAULT] du fichier `/etc/nova/nova.conf`.**

```
novncproxy_base_url=http://public-IP:6080/vnc_auto.html
```

où *public-IP* est l'adresse IP publique de l'hôte du contrôleur.

- b. **Activez les services Nova comme suit :**

```
controller# svcadm enable nova-consoleauth
controller# svcadm enable nova-novncproxy
```

Configuration du noeud de stockage

Le noeud de contrôleur constitue le référentiel de toutes les données traitées dans le paramétrage d'OpenStack.

Pour configurer le noeud de calcul, installez les composants et services OpenStack sur le système à l'aide de la commande suivante :

```
storage# pkg install openstack
```

Une fois l'installation du package terminée, configurez les services à exécuter sur le noeud.

Remarque - Pour une meilleure gestion de l'utilisation de la mémoire entre ZFS et les applications dans Oracle Solaris 11, définissez le paramètre `usr_reserve_hint_pct` sur le noeud, comme indiqué dans l'exemple suivant :

```
# echo "set user_reserve_hint_pct=80" >>/etc/system.d/site:kernel-zones-reserve
# reboot
```

où `site` est un identificateur aléatoire, par exemple le nom de votre société.

Définissez également ce paramètre sur les différents noeuds OpenStack.

Pour plus d'informations sur ce paramètre, connectez-vous à votre compte MOS à l'adresse <https://support.oracle.com> et consultez le Document 1663862.1, *Memory Management Between ZFS and Applications in Oracle Solaris 11.2*.

▼ Configuration du noeud de stockage par bloc

Cette procédure décrit une configuration type de stockage par blocs. Pour les autres options permettant de configurer le composant de stockage, voir [Chapitre 6, Options pour la configuration et le déploiement de Cinder](#).

1. **Configurez le client NTP.**
Voir "[Configuration du client NTP](#)" à la page 30.
2. **Configurez Cinder en éliminant les marques de commentaire ou en définissant les paramètres appropriés dans le fichier `/etc/cinder/cinder.conf`.**

```
[DEFAULT]
san_is_local=true
volume_driver=cinder.volume.drivers.solaris.zfs.ZFSISCSIDriver
my_ip=$VOLUME_IP
glance_host=$CONTROLLER_ADMIN_NODE
zfs_volume_base=cinder/cinder

[database]
connection = mysql://cinder:service-password@$CONTROLLER_ADMIN_NODE/cinder

[keystone_authtoken]
auth_uri = http://$CONTROLLER_ADMIN_NODE:5000/v2.0
identity_uri = http://$CONTROLLER_ADMIN_NODE:35357
admin_user = cinder
admin_password = service-password
admin_tenant_name = service

[oslo_messaging_rabbit]
rabbit_host=$CONTROLLER_ADMIN_NODE
```

3. Définissez les paramètres dans le fichier `/etc/cinder/api-paste.ini`.

```
[filter:authtoken]
admin_tenant_name = service
admin_user = cinder
admin_password = service-password
```

4. Démarrez les services Cinder..

```
storage# svcadm enable -rs cinder-db cinder-volume:default cinder-volume:setup
storage# svcadm enable -rs iscsi/target
```

Configuration du stockage d'objets Swift

Swift est le projet de stockage d'objets OpenStack. Il offre un logiciel de stockage cloud qui vous permet de stocker et d'extraire des données volumineuses à l'aide d'une simple API. Le service est idéal pour stocker des données non structurées qui peuvent s'agrandir sans limite.

Pour plus d'informations sur Swift, reportez-vous au chapitre portant sur le stockage d'objets dans le [OpenStack Cloud Administrator Guide](#) de la communauté OpenStack.

La documentation de la communauté OpenStack recommande au moins 6 nœuds pour un déploiement Swift en mode de production. Ces nœuds sont composés d'un contrôleur proxy Swift et de 5 nœuds de contrôleur Swift. Ce guide, toutefois, décrit un déploiement à 3 nœuds cohérent avec l'architecture à 3 nœuds sur laquelle les procédures de configuration précédentes ont été basées. Vous pouvez ajouter des nœuds de contrôleur ultérieurement si besoin est.

▼ Configuration du noeud de service pour le contrôleur proxy Swift

Cette tâche suppose que vous ayez déjà installé les packages OpenStack dans le noeud désigné pour Swift. Reportez-vous à la commande d'installation à utiliser à la section "[Configuration du noeud de stockage](#)" à la page 46.

1. Installez les packages Swift.

```
proxy-node # pkg install swift swiftclient
```

2. Créez des jeux de données ZFS.

```
proxy-node # /usr/sbin/zfs create -o mountpoint=none rpool/export/swift
proxy-node # /usr/sbin/zfs create -o mountpoint=/srv rpool/export/swift/srv
proxy-node # /usr/sbin/zfs create -p rpool/export/swift/srv/node/disk0
proxy-node # /usr/bin/chown -R swift:swift /srv
```

3. Effectuez les vidages octaux suivants.

Conservez les valeurs des vidages. Ces deux valeurs sont appelées \$OD_1 et \$OPD_2 dans les étapes suivantes.

```
proxy-node # od -t x8 -N 8 -A n < /dev/random
proxy-node # od -t x8 -N 8 -A n < /dev/random
```

4. Modifiez le fichier /etc/swift/swift.conf à l'aide des paramètres suivants.

```
[swift-hash]
swift_hash_path_suffix = $OD_1
swift_hash_path_prefix = $OD_2
```

5. Modifiez le fichier /etc/swift/proxy-server.conf à l'aide des paramètres suivants :

```
[DEFAULT]
bind_port = 8080

[filter:tempauth]
use = egg:swift#tempauth

operator_roles = admin, swiftoperator

[filter:authtoken]
auth_uri = http://$CONTROLLER_IP:5000/
identity_uri = http://$CONTROLLER_IP:35357
admin_user = swift
admin_password = swiftpass
admin_tenant_name = service

[filter:cache]
memcache_servers = $CONTROLLER_IP:11211
```

```
[oslo_messaging_rabbit]
rabbit_host=${CONTROLLER_ADMIN_NODE}
```

6. Activez le démon memcached.

```
proxy-node # svcadm enable -rs memcached
```

7. Créez les anneaux.

```
proxy-node # cd /etc/swift
proxy-node # swift-ring-builder account.builder create 18 3 1
proxy-node # swift-ring-builder container.builder create 18 3 1
proxy-node # swift-ring-builder object.builder create 18 3 1
proxy-node # swift-ring-builder account.builder add r1z1-$$STORAGE_IP_1:6002/disk0 100
proxy-node # swift-ring-builder container.builder add r1z1-$$STORAGE_IP_1:6001/disk0 100
proxy-node # swift-ring-builder object.builder add r1z1-$$STORAGE_IP_1:6000/disk0 100
proxy-node # swift-ring-builder account.builder add r1z1-$$STORAGE_IP_2:6002/disk0 100
proxy-node # swift-ring-builder container.builder add r1z1-$$STORAGE_IP_2:6001/disk0 100
proxy-node # swift-ring-builder object.builder add r1z1-$$STORAGE_IP_2:6000/disk0 100
proxy-node # swift-ring-builder account.builder rebalance
proxy-node # swift-ring-builder container.builder rebalance
proxy-node # swift-ring-builder object.builder rebalance
proxy-node # >chown -R swift:swift /etc/swift
```

8. Activez le service Swift.

```
proxy-node # svcadm enable swift-proxy-server
```

▼ Configuration du noeud de stockage d'objets

Répétez cette procédure pour chaque noeud de contrôleur d'objets à configurer.

1. Installez les packages Swift.

```
storage-node # pkg install swift swiftclient
```

2. Créez des jeux de données ZFS.

```
storage-node # /usr/sbin/zfs create -o mountpoint=none rpool/export/swift
storage-node # /usr/sbin/zfs create -o mountpoint=/srv rpool/export/swift/srv
storage-node # /usr/sbin/zfs create -p rpool/export/swift/srv/node/disk0
storage-node # /usr/bin/chown -R swift:swift /srv
```

3. Copiez les fichiers à partir du noeud de serveur proxy comme suit :

- a. Copiez le fichier `/etc/swift/swift.conf` à partir du noeud de serveur proxy vers le répertoire `/etc/swift` du noeud actuel.

b. Copiez les fichiers suivants à partir du noeud de serveur proxy vers le répertoire `/etc/swift` du noeud actuel.

- `account.ring.gz`
- `container.ring.gz`
- `object.ring.gz`

4. Activez le service duplicateur Swift.

```
storage-node # svcadm enable swift-replicator-rsync
```

5. Définissez la propriété du répertoire `/etc/swift` du noeud actuel.

```
storage-node # chown -R swift:swift /etc/swift
```

6. Activez tous les services Swift.

```
storage-node # for x in `svcs -a -o SVC | fgrep swift | \
  egrep "account|container|object" | sort` \
  do \
    echo Starting $x \
    svcadm enable $x \
  done
```

7. Sur le noeud de contrôleur, autorisez les utilisateurs à accéder aux services Swift et à les utiliser.

a. Définissez des variables shell globales pour Swift.

```
controller# export OS_USERNAME=swift
controller# export OS_PASSWORD=service-password
controller# export OS_PROJECT_NAME=service
controller# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0
```

b. Ajoutez le rôle Keystone `swiftoperator`.

```
controller# openstack role create --name swiftoperator
```

c. Affectez le rôle `swiftoperator` aux utilisateurs autorisés des services Swift.

```
controller# openstack user role add --user user-name \
  --role swiftoperator --project tenant-ID
```

Étapes suivantes Pour permettre aux utilisateurs d'utiliser le cloud, terminez les préparations décrites dans le chapitre [Chapitre 4, Tâches de post-installation et de configuration](#).

◆◆◆ 4 CHAPITRE 4

Tâches de post-installation et de configuration

Ce chapitre décrit les procédures permettant de terminer l'installation et la configuration initiale d'OpenStack. Ce chapitre comprend les sections suivantes :

- ["Préparation du réseau externe pour les projets OpenStack" à la page 51](#)
- ["Préparation des images pour le référentiel Glance" à la page 59](#)

Préparation du réseau externe pour les projets OpenStack

Le réseau externe assure la connexion entre les réseaux privés dans le cloud et le réseau public.

A propos du routeur de fournisseur

Le routeur assure la connectivité des instances de machine virtuelle de projet avec des réseaux plus larges. Le routeur est partagé par tous les réseaux de projet. Etant donné qu'il n'y a qu'un routeur, les réseaux de projet ne peuvent pas utiliser le chevauchement des adresses IP.

Le routeur effectue une translation d'adresse réseau (NAT) bidirectionnelle sur l'interface qui connecte le routeur au réseau externe. Les projets peuvent avoir autant d'adresses IP flottantes (IP publiques) que nécessaires ou qu'autorisées par le quota d'IP flottantes. Ces IP flottantes sont associées aux instances de machine virtuelle qui requièrent une connectivité avec l'extérieur.

La création du routeur requiert la configuration de l'agent L3 Neutron. Cet agent crée automatiquement des mappages NAT biunivoques entre les adresses affectées aux instances Nova et les adresses IP flottantes. L'agent L3 permet aussi la communication entre réseaux privés.

Par défaut, le routage entre les réseaux privés qui font partie du même projet est désactivé. Pour modifier ce comportement, définissez `allow_forwarding_between_networks` sur `True` dans le fichier de configuration `/etc/neutron/l3_agent.ini`. Redémarrez le service SMF `neutron-l3-agent` après avoir défini le paramètre.

▼ Configuration du routeur pour le réseau externe

Cette procédure explique comment créer le routeur pour le réseau externe. Une partie des étapes nécessite la modification d'un fichier de configuration. Aussi, il est plus pratique d'utiliser la fenêtre de terminal pour cette procédure que le tableau de bord Horizon.

Vous effectuez les étapes suivantes sur le noeud où le service Neutron est installé. Ce document localise le service sur le noeud de contrôleur, en fonction de l'exemple d'architecture décrit dans les chapitres précédents.

Avant de commencer Prenez soin de terminer la configuration Neutron telle que décrite dans la section "[Installation et configuration de Neutron](#)" à la page 38.

1. Démarrez le service de filtre IP s'il est désactivé.

```
controller# svcadm enable -rs ipfilter
```

2. Activez le transfert d'IP sur l'hôte, s'il ne l'est pas.

```
controller# ipadm set-prop -p forwarding=on ipv4
```

3. Définissez les variables shell globales pour Neutron.

```
controller# export OS_USERNAME=neutron
controller# export OS_PASSWORD=service-password
controller# export OS_PROJECT_NAME=service
controller# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0
```

4. Créez le routeur de fournisseur.

```
controller# neutron router-create router-name
```

La commande affiche le nom du routeur avec le code correspondant. Utilisez ce code pour mettre à jour le fichier de configuration à l'étape suivante.

5. Vérifiez que le fichier `/etc/neutron/l3_agent.ini` contient les paramètres suivants.

```
router_id=routerID    ID obtenu à l'étape précédente.
```

6. Activez le service SMF `neutron-l3-agent`.

```
controller# svcadm enable neutron-l3-agent
```

7. (Facultatif) Affichez les informations sur le routeur.

Une fois que vous avez ajouté le réseau externe au routeur, des informations supplémentaires sont ajoutées sur le routeur.

```
controller# neutron router-show router-name
```

Exemple 1 Création du routeur

Cet exemple explique comment créer le routeur pour le réseau externe.

```

controller# svcadm enable -rs ipfilter

controller# ipadm set-prop -p forwarding=on ipv4
controller# ipadm set-prop -p forwarding=on ipv6

controller# export OS_USERNAME=neutron
controller# export OS_PASSWORD=service-password
controller# export OS_PROJECT_NAME=service
controller# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0

controller# neutron router-create ext-router
Created a new router:
+-----+-----+
| Field                | Value                                |
+-----+-----+
| admin_state_up       | True                                  |
| external_gateway_info |                                       |
| id                   | f89b24ed-42dd-48b0-8f4b-fd41887a3370 |
| name                 | ext-router                           |
| status               | ACTIVE                               |
| project_id           | 7d1caf0854b24becb28df5c5cabf72cc    |
+-----+-----+

```

A ce stade, vous devez mettre à jour le paramètre `router_ID` dans le fichier `/etc/neutron/l3_agent.ini`.

```
router_id = f89b24ed-42dd-48b0-8f4b-fd41887a3370
```

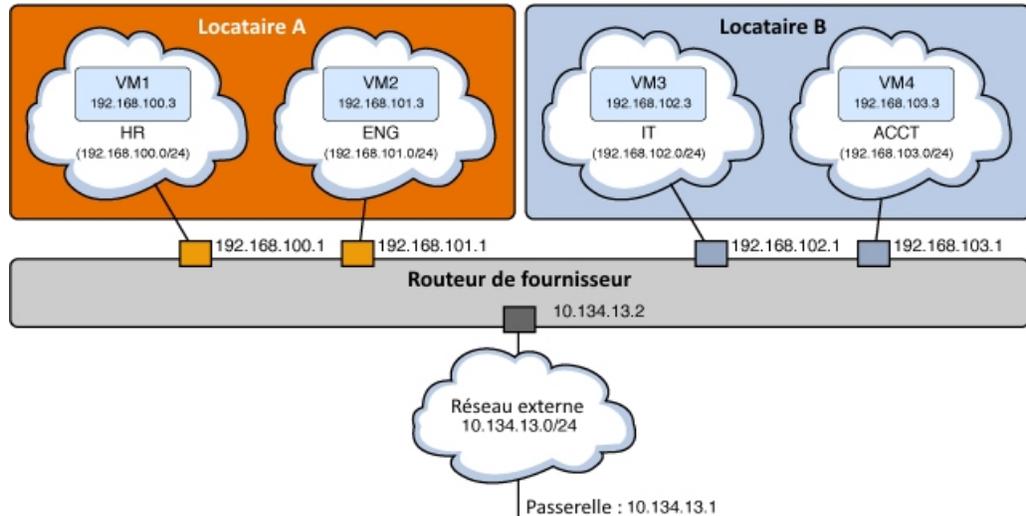
Puis, vous devez activer le service de l'agent L3.

```
controller# svcadm enable neutron-l3-agent
```

Création du réseau externe

Après avoir créé le routeur, vous devez configurer le réseau externe. Au moyen de la passerelle du réseau externe, les utilisateurs des réseaux internes du cloud peuvent se connecter à Internet.

La figure suivante illustre comment un seul routeur prend en charge les communications réseau des projets du cloud.

FIGURE 3 Routeur de fournisseur avec modèle de réseaux privés

La figure présente les éléments suivants :

- Deux projets (Locataire A et Locataire B)
- Quatre machines virtuelles (VM1, VM2, VM3 et VM4)
- Quatre sous-réseaux (HR, ENG, IT et ACCT)
- Routeur
- Réseau externe

Quand vous fournissez un accès de réseau interne au réseau externe, l'adresse IP de la machine virtuelle est mappée à l'une des adresses flottantes que vous affectez au réseau externe.

▼ Création d'un réseau externe

Cette procédure explique comment créer un réseau virtuel qui représente un réseau externe. Ce réseau virtuel n'utilise pas DHCP. A la place, les adresses IP flottantes sont créées et affectées aux projets et utilisées par les instances de machine virtuelle Nova sous ces projets. Ces étapes créent un type de réseau VLAN, mais la procédure s'applique à la création d'autres types de réseau, tels qu'un réseau plat.

Vous pouvez créer le réseau externe indépendamment de la création des réseaux internes.

Avant de commencer Terminez la configuration du commutateur virtuel élastique. Pour plus d'informations, reportez-vous à la section "[Installation et configuration de Neutron](#)" à la page 38, notamment aux étapes de la configuration EVS.

1. Définissez les variables shell globales pour Neutron.

```
controller# export OS_USERNAME=neutron
controller# export OS_PASSWORD=service-password
controller# export OS_PROJECT_NAME=service
controller# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0
```

2. (Facultatif) Obtenez la plage de VLAN.

```
controller# evsadm show-controlprop -p vlan-range
```

3. Créez un réseau externe.

```
controller# neutron net-create --provider:network_type=vlan \
--provider:segmentation_id=VLAN-nbr \
--router:external network-name
```

Cette étape suppose qu'un réseau VLAN a été créé en suivant la procédure indiquée à la section "[Installation et configuration de Neutron](#)" à la page 38. `segmentation_id` est l'ID de VLAN du réseau VLAN, dont la plage a été définie lors de la configuration d'EVS.

Remarque - Si vous créez un réseau plat, vous n'avez pas à spécifier un ID segmentation.

4. Créez le sous-réseau du réseau externe.

Le pool d'allocation se compose d'une plage d'adresses IP flottantes qui est affectée au sous-réseau.

```
controller# neutron subnet-create --name subnet-name --disable-dhcp \
--allocation-pool start=start-IP,end=end-IP \
network-name subnet-IP
```

5. Ajoutez le réseau externe au routeur.

```
controller# neutron router-gateway-set router-name ext-network-ID
```

Remarque - Par défaut, la SNAT est activée quand vous exécutez cette commande. Lorsque le mode SNAT est activé, les machines virtuelles du réseau privé peuvent accéder au réseau externe. Toutefois, les instances elles-mêmes ne peuvent pas être accédées en dehors du cloud. Pour désactiver la SNAT, spécifiez l'option `--disable-snat` pour la sous-commande `neutron router-gateway-set`.

6. (Facultatif) Affichez les informations sur le routeur.

```
controller# neutron router-show router-name
```

Exemple 2 Création du réseau externe

Cet exemple explique comment créer le réseau externe pour le préparer à l'utilisation par des réseaux internes dans le cloud.

Pour créer un réseau plat, reportez-vous à l'exemple fourni dans la section 2 de <https://blogs.oracle.com/openstack/tags/juno>.

```

controller# export OS_USERNAME=neutron
controller# export OS_PASSWORD=service-password
controller# export OS_PROJECT_NAME=service
controller# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0

controller# evsadm show-controlprop -p vlan-range
PROPERTY      PERM VALUE      DEFAULT  HOST
vlan-range    rw  1,200-300    --      --

controller# neutron net-create --router:external \
--provider:network_type=vlan --provider:segmentation_id=1 ext_network
Created a new network:
+-----+
| Field                | Value                                     |
+-----+
| admin_state_up       | True                                     |
| id                   | 08cf49c8-f28f-49c1-933d-bdb1017e0294    |
| name                 | ext_network                             |
| provider:network_type | vlan                                     |
| provider:segmentation_id | 1                                       |
| router:external      | True                                     |
| shared               | False                                    |
| status               | ACTIVE                                  |
| subnets             |                                          |
| project_id           | 7d1caf0854b24becb28df5c5cabf72cc       |
+-----+

controller# neutron subnet-create --name ext_subnet --disable-dhcp \
--allocation-pool start=10.134.10.8,end=10.134.10.254 \
ext_network 10.134.10.0/24
Created a new subnet:
+-----+
| Field                | Value                                     |
+-----+
| allocation_pools     | {"start": "10.134.10.8", "end": "10.134.10.254"} |
| cidr                 | 10.134.10.0/24                           |
| dns_nameservers      |                                          |
| enable_dhcp          | False                                    |
| gateway_ip           | 10.134.13.1                               |
| host_routes          |                                          |
| id                   | fce503ff-f483-4024-b122-f2524e3edae1    |
| ip_version           | 4                                         |
| ipv6_address_mode    |                                          |
| ipv6_ra_mode         |                                          |
| name                 | ext_subnet                                |
+-----+

```

```

| network_id      | 08cf49c8-f28f-49c1-933d-bdb1017e0294 |
| project_id     | 7d1caf0854b24becb28df5c5cabf72cc   |
+-----+-----+
controller# neutron router-gateway-set ext-router 08cf49c8-f28f-49c1-933d-bdb1017e0294
Set gateway for router ext-router

controller# neutron router-show ext-router
+-----+-----+
| Field          | Value                                |
+-----+-----+
| admin_state_up | True                                 |
| external_gateway_info | {"network_id": "08cf49c8-f28f-49c1-933d-bdb1017e0294", |
|                  | "enable_snat": true,                |
|                  | "external_fixed_ips":                |
|                  | [{"subnet_id": "fce503ff-f483-4024-b122-f2524e3edae1", |
|                  | "ip_address": "10.134.10.8"}]} |
| id             | f89b24ed-42dd-48b0-8f4b-fd41887a3370 |
| name           | ext-router                           |
| status         | ACTIVE                                |
| project_id     | 7d1caf0854b24becb28df5c5cabf72cc   |
+-----+-----+

```

- Voir aussi
- ["Respect de la configuration de l'agent L3" à la page 58.](#)
 - ["Limitations connues" à la page 134.](#)

▼ Connectivité externe aux réseaux internes

Utilisez cette procédure pour activer un réseau interne pour accéder au réseau public étendu. Cette procédure suppose que des réseaux internes existent déjà pour des projets spécifiques. Pour créer un réseau interne de projet à l'aide du tableau de bord, reportez-vous à la section ["Création de réseaux internes pour les projets" à la page 71.](#)

Avant de commencer Avant de poursuivre, obtenez le nom du sous-réseau qui nécessite l'accès public.

1. Définissez les variables shell globales pour Neutron.

```

controller# export OS_USERNAME=neutron
controller# export OS_PASSWORD=service-password
controller# export OS_PROJECT_NAME=service
controller# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0

```

2. Identifiez l'ID du sous-réseau qui requiert l'accès externe.

```

controller# neutron subnet-list | grep subnet-name

```

3. (Facultatif) Obtenez le nom du routeur.

```

controller# neutron router-list

```

4. Ajoutez l'ID du sous-réseau comme interface pour le routeur.

```
controller# neutron router-interface-add router-name subnetID
```

Exemple 3 Connexion d'un réseau interne au réseau externe

Cet exemple utilise le réseau interne HR qui a été créé dans la section "[Configuration d'un réseau pour un projet](#)" à la page 71. Le réseau HR, dont le sous-réseau est HR_Subnet, requiert un accès au réseau public.

```
controller# neutron subnet-list | grep HR_Subnet
| b6feff42-36aa-4235- | HR_Subnet | 10.132.30.0/24 | {"start": "10.132.30.2", |
| 9fe0-ac5de6b43af3 | | | "end": "10.132.30.254"} |
```

```
controller# neutron router-list
+-----+-----+-----+
| id | name | external_gateway_info |
+-----+-----+-----+
| f89b24ed-42dd-48b0- | ext-router | {"network_id": "6c4c1823-a203- |
| 8f4b-fd41887a3370 | | 43b1-9674-ddb5ff4185fc", |
| | | "enable_snat": true, |
| | | "external_fixed_ips": |
| | | [{"subnet_id": "83d9b40f-cc61- |
| | | 4696-b22e-b4cbc2aa3872", |
| | | "ip_address": "10.132.10.8"}]} |
+-----+-----+-----+
```

```
controller# neutron router-interface-add ext-router b6feff42-36aa-4235-9fe0-ac5de6b43af3
Added interface b6feff42-36aa-4235-9fe0-ac5de6b43af3 to router ext-router.
```

▼ Respect de la configuration de l'agent L3

Vous pouvez utiliser des commandes de filtre IP, telles que ipf, ippool, et ipnat et des commandes de mise en réseau, telles que dladm et ipadm pour respecter et résoudre les problèmes de la configuration effectuée par neutron-l3-agent.

1. Affichez les VNIC créées par neutron-l3-agent.

```
network# dladm show-vnic
LINK OVER SPEED MACADDRESS MACADDRTYPE VIDS
l3i7843841e_0_0 net1 1000 2:8:20:42:ed:22 fixed 200
l3i89289b8e_0_0 net1 1000 2:8:20:7d:87:12 fixed 201
l3ed527f842_0_0 net0 100 2:8:20:9:98:3e fixed 0
```

2. Affichez les adresses IP créée par neutron-l3-agent.

```
network# ipadm
NAME CLASS/TYPE STATE UNDER ADDR
l3ed527f842_0_0 ip ok -- --
```

```

l3ed527f842_0_0/v4 static ok -- 10.134.10.8/24
l3ed527f842_0_0/v4a static ok -- 10.134.10.9/32
l3i7843841e_0_0 ip ok -- --
l3i7843841e_0_0/v4 static ok -- 192.168.100.1/24
l3i89289b8e_0_0 ip ok -- --
l3i89289b8e_0_0/v4 static ok -- 192.168.101.1/24

```

3. Affichez les règles de filtrage IP.

```

network# ipfstat -io
empty list for ipfilter(out)
block in quick on l3i7843841e_0_0 from 192.168.100.0/24 to pool/4386082
block in quick on l3i89289b8e_0_0 from 192.168.101.0/24 to pool/8226578
network# ippool -l
table role = ipf type = tree number = 8226578
{ 192.168.100.0/24; };
table role = ipf type = tree number = 4386082
{ 192.168.101.0/24; };

```

4. Affichez les règles de filtrage NAT.

```

network# ipnat -l
List of active MAP/Redirect filters:
bimap l3ed527f842_0_0 192.168.101.3/32 -> 10.134.13.9/32
List of active sessions:
BIMAP 192.168.101.3 22 <- -> 10.134.13.9 22 [10.132.146.13 36405]

```

Préparation des images pour le référentiel Glance

Une image représente le fondement des instances de machine virtuelle dans le cloud. Une image est un simple fichier qui contient un disque virtuel sur lequel est installé un système d'exploitation amorçable. Elle fournit un modèle pour la création d'une ou de plusieurs machines virtuelles. Ainsi, pour provisionner une machine virtuelle sur le cloud, vous devez d'abord créer une image.

Glance, le service d'image d'OpenStack, assure les services de stockage, de découverte, d'enregistrement et de livraison pour les images disque et serveur. Un *serveur de registre* est un service d'image fournissant les informations de métadonnées de l'image aux clients. L'*image cache* est utilisée par le service d'image pour obtenir des images sur l'hôte local au lieu de devoir effectuer un nouveau téléchargement à partir du serveur d'images chaque fois qu'une image est demandée.

Vous pouvez charger plusieurs images dans le référentiel Glance. A titre de recommandation, téléchargez les images de différents types de système que vous déploierez sur le cloud. Par exemple, créez des images archivées de zones non globales, de zones de noyau et de zones globales. Vous pouvez ensuite déployer une machine virtuelle de l'un de ces types rapidement en sélectionnant le modèle approprié.

Création des images

Dans Oracle Solaris, vous utilisez la fonctionnalité Unified Archives pour créer des images OpenStack. La commande `archiveadm` permet de créer de nouvelles UA à partir de zones globales, non globales et de noyau.

L'UA peut être une archive de clone ou une archive de récupération. Une archive de *clone* se base sur l'environnement d'initialisation (BE) actif. Cette archive ne comprend aucune information de configuration système provenant de l'instance de système d'exploitation, notamment des environnements d'initialisation inactifs. A la place, les programmes d'installation imposent une reconfiguration ou utilisent les informations de configuration que vous avez indiquées dans un profil de configuration système (SC). Une archive de *récupération* inclut tous les environnements d'initialisation et informations de configuration système. Par conséquent, si vous voulez inclure toutes les informations du système dans une UA, vous devez créer une archive de récupération. Pour plus d'informations sur les UA, consultez le manuel *Utilisation de Unified Archives pour la récupération du système et le clonage*. Ce manuel figure dans la bibliothèque correspondant à votre version d'Oracle Solaris dans la page [Systems Operation Documentation](#).

Plus tard, quand la configuration est totalement opérationnelle, vous pouvez également créer une image en créant un instantané d'une instance de machine virtuelle existante. Pour ce cas, l'instance de machine virtuelle est déjà dans le cloud. En conséquence, la commande à utiliser est `nova image-create` au lieu de `archiveadm`. La commande `nova` prend un instantané d'une instance de machine virtuelle en cours d'exécution pour créer l'image.

Vous pouvez également utiliser des images personnalisées pour les sauvegardes de données ou pour assurer le secours d'une instance de machine virtuelle. Une *image de secours* est un type spécial d'image qui est initialisée lorsqu'une instance de machine virtuelle est placée en mode secours. Une image de secours permet à un administrateur de monter les systèmes de fichiers pour l'instance de machine virtuelle afin de résoudre le problème.

Dans Oracle Solaris, vous créez une image OpenStack en trois phases :

1. Créez la zone.
2. Créez l'UA de la zone.
3. Téléchargez l'UA vers Glance.

Ces phases sont combinées dans la procédure suivante.

▼ Création d'une image pour OpenStack

A l'exception de la syntaxe de commande pour créer la zone, tout le reste des étapes peut être utilisé pour la création et le téléchargement des images des zones non globales et des zones de noyau.

Dans cette procédure, l'étape pour la création des zones fournit uniquement les commandes de base. Pour des instructions complètes lors de la création de zones, reportez-vous au document *Création et utilisation de zones Oracle Solaris* relatif à l'installation, la mise hors tension, l'arrêt, la désinstallation et le clonage de zones non globales. Ce manuel figure dans la bibliothèque correspondant à votre version d'Oracle Solaris dans la page [Operating Systems Documentation](#).

1. Sur n'importe quel système, créez la zone puis connectez-vous à la zone.

Après la connexion, fournissez les informations quand vous y êtes invité.

```
global# zonecfg -z zone-name create
global# zoneadm -z zone-name install
global# zoneadm -z zone-name boot
global# zlogin -C zone-name
```

Remarque - Cette étape peut prendre un certain temps.

2. Activez root SSH pour l'accès de connexion root OpenStack.

```
global# zlogin zone-name
root@zone-name# sed /^PermitRootLogin/s/no$/without-password/ < /etc/ssh/sshd_config
> /system/volatile/sed.$$
root@zone-name# cp /etc/ssh/sshd_config /etc/ssh/sshd_config.orig
root@zone-name# cp /system/volatile/sed.$$ /etc/ssh/sshd_config
root@zone-name# exit
```

3. Créez l'UA pour la zone.

```
global# archiveadm create -z zone-name /var/tmp/archive-name.uar
```

4. Transférez l'UA vers le système où Glance est installé.

Ce document suppose que Glance est sur le noeud de contrôleur.

5. Définissez les variables shell globales pour Glance.

```
controller# export OS_USERNAME=glance
controller# export OS_PASSWORD=service-password
controller# export OS_PROJECT_NAME=service
controller# export OS_AUTH_URL=$CONTROLLER_ADMIN_NODE:5000/v2.0
```

6. Téléchargez l'UA vers le référentiel Glance.

```
controller# glance --os-image-api-version 2 image-create \
--container-format bare --disk-format raw \
--visibility visibility-mode --name "image-name" \
--property architecture=system-arch \
--property hypervisor_type=solariszones \
--property vm_mode=solariszones --file path-to-archive-file
```

<i>system-arch</i>	Architecture du système qui peut être x86_64 ou sparc64.
<i>visibility-mode</i>	Portée de l'accessibilité des images qui peut être public ou private.

Affichage des informations sur les images

Pour afficher des informations d'image, vous pouvez utiliser soit la commande `nova`, soit la commande `glance`.

```
$ nova image-list
+-----+-----+-----+-----+
| ID                | Name                | Status | Server |
+-----+-----+-----+-----+
| 4dfbfd4f-2de5-4251-832c-e35a4a4145ee | Solaris Non-global Zone | ACTIVE |        |
+-----+-----+-----+-----+
```

La commande `glance image-list` affiche des informations supplémentaires, y compris le format de disque, le format de conteneur et la taille des différentes images.

Les commandes `nova image-show` et `glance image-show` affichent des informations sur une image spécifique. Chaque commande génère une sortie différente pour l'image.

```
$ nova image-show 'Solaris Non-global Zone'
+-----+-----+
| Property          | Value               |
| OS-EXT-IMG-SIZE:size | 845025280           |
| created           | 2015-11-19T14:46:38Z |
| id                | 4dfbfd4f-2de5-4251-832c-e35a4a4145ee |
| metadata architecture | x86_64              |
| metadata hypervisor_type | solariszones        |
| metadata vm_mode    | solariszones        |
| minDisk           | 0                   |
| minRam            | 0                   |
| name              | Solaris Non-global Zone |
| progress          | 100                 |
| status            | ACTIVE              |
| updated           | 2015-11-19T14:46:42Z |
+-----+-----+
```

```
$ glance image-show 'Solaris Non-global Zone'
+-----+-----+
| Property          | Value               |
+-----+-----+
| Property 'architecture' | x86_64              |
| Property 'hypervisor_type' | solariszones        |
| Property 'vm_mode'      | solariszones        |
| checksum            | ba9b9eeddb467833d725c8750a46e004 |
| container_format     | bare                |
+-----+-----+
```

```

| created_at          | 2015-11-19T14:46:38 |
| deleted            | False               |
| disk_format        | raw                 |
| id                 | 4dfbfd4f-2de5-4251-832c-e35a4a4145ee |
| is_public          | True                |
| min_disk           | 0                   |
| min_ram            | 0                   |
| name               | Solaris Non-global Zone |
| owner              | 7d1caf0854b24becb28df5c5cabf72cc |
| protected          | False               |
| size               | 845025280           |
| status             | active              |
| updated_at         | 2015-11-19T14:46:42 |
+-----+-----+

```

Remarque - Vous pouvez obtenir les mêmes informations sur l'image à partir du tableau de bord Horizon.

Utilisation d'un script de création d'image Glance

La commande `glance image-create` permet de télécharger l'image et de définir toutes les valeurs de propriété à la fois. Le script suivant montre comment s'assurer que vous avez téléchargé l'image avec la propriété `architecture` définie sur l'architecture de l'hôte en cours :

```

#!/bin/ksh

# Upload Unified Archive image to glance with proper Solaris decorations

arch=$(archiveadm info -p $1|grep ^archive|cut -d '|' -f 4)

if [[ "$arch" == "i386" ]]; then
    imgarch=x86_64
else
    imgarch=sparc64
fi

name=$(basename $1 .uar)

export OS_USERNAME=glance
export OS_PASSWORD=glance
export OS_TENANT_NAME=service
export OS_AUTH_URL=http://controller-name:5000/v2.0

glance image-create --name $name --container-format bare --disk-format raw --owner service
--file $1 --is-public True --property architecture=$imgarch --property
hypervisor_type=solariszones
--property vm_mode=solariszones --progress

```


Utilisation du cloud

Ce chapitre explique l'exécution de différentes tâches administratives sur le cloud. Pour effectuer ces tâches, vous pouvez utiliser le tableau de bord ou les lignes de commande. Sur le tableau de bord, les tâches figurant sous l'onglet Projet ne requièrent que le rôle de membre alors que les tâches de l'onglet Admin requièrent des privilèges d'administration. Pour effectuer l'ensemble de ces tâches pour un projet lors d'une session de connexion unique sur le tableau de bord, vous devez disposer à la fois des rôles de membre et d'administrateur pour ce projet.

Ce chapitre comprend les sections suivantes :

- ["Création de projets et d'utilisateurs" à la page 68](#)
- ["Création de réseaux internes pour les projets" à la page 71](#)
- ["Création et initialisation d'une instance de machine virtuelle" à la page 74](#)
- ["Gestion des variantes" à la page 80](#)
- ["Gestion des instances de machine virtuelle" à la page 83](#)

Accès au tableau de bord OpenStack

Une fois les tâches de configuration d'installation et suivant l'installation de OpenStack terminées, connectez-vous au tableau de bord OpenStack pour visualiser les ressources disponibles dans le cloud.

▼ Accès au tableau de bord OpenStack

1. **Connectez-vous à n'importe quel système pouvant se connecter au système OpenStack.**
2. **Configurez votre navigateur.**
 - a. **Activez JavaScript.**

b. Conservez les cookies.

3. Dans le champ d'emplacement ou d'adresse du navigateur, entrez l'emplacement suivant :

`http://system/horizon/`

Le `system` est soit le nom soit l'adresse IP du système OpenStack où l'archive d'ensemble OpenStack a été installée, et qui exécute le service Horizon OpenStack sous le serveur web Apache.

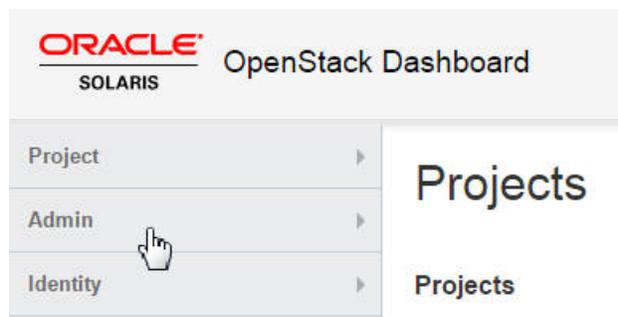
Si vous avez installé l'archive d'ensemble sur une zone de noyau, le système OpenStack est la zone de noyau, et `system` est le nom ou l'adresse IP de la zone de noyau.

4. Indiquez les informations suivantes sur l'écran de connexion :

- Nom utilisateur : `admin`
- Mot de passe : `secrete`

Exploration du tableau de bord

Au départ, connectez-vous au tableau de bord Horizon en tant qu'utilisateur `admin` pour ouvrir la page d'accueil demo du projet. Comme `admin` dispose de privilèges d'administration sur `demo`, la page affiche 3 onglets sur le panneau de gauche : `Project`, `Admin` et `Identity`. Sans privilèges d'administration, un utilisateur ne verrait que les onglets `Project` et `Identity`.



La page Vue d'ensemble Résumé de l'utilisation du panneau Admin est la vue par défaut de l'administrateur du cloud.

FIGURE 4 Fenêtre de présentation d'administration du tableau de bord OpenStack

Les choix proposés sur le panneau de l'administrateur permettent la fonctionnalité suivante :

- Une vue globale des instances Nova et des volumes Cinder utilisés à l'intérieur du cloud
- La capacité à afficher et modifier les définitions de variantes qui définissent les caractéristiques de l'instance de machine virtuelle telles que :
 - Le nombre de CPU virtuelles
 - La quantité de mémoire
 - L'espace disque affecté
 - La marque de la zone Oracle Solaris sous-jacente : `solaris` pour les zones non globales ou `solaris-kz` pour les zones de noyau
- La possibilité de créer des routeurs et des réseaux virtuels en vue de leur utilisation par les administrateurs du cloud
- La possibilité d'afficher et de modifier des projets en regroupant et en isolant l'appartenance des ressources de calcul virtuelles.
- La possibilité d'afficher et de modifier les utilisateurs, qui sont les personnes ou les services qui utilisent les ressources du cloud.

L'UA OpenStack pour la configuration en mode monoeud fournit les ressources préconfigurées suivantes :

- Deux images : Zone non globale Solaris et Zone de noyau Solaris
- Deux projets ou locataires : `demo` et `service`

Par défaut, le projet `demo` est créé avec un utilisateur unique, `admin`, en tant que membre.

Le projet `service` est un projet utilisé par l'administrateur du cloud pour créer des ressources qui seront partagées entre plusieurs projets. Par exemple, dans le document, le routeur Neutron est créé dans le projet `service` de sorte que le routeur soit partagé par tous les projets. Vous ne devez pas utiliser le projet `service` à toute autre fin dans votre configuration OpenStack. Les services OpenStack communiquent les uns avec les autres au moyen d'utilisateurs spécifiques du service, qui disposent tous de privilèges d'administration dans le projet `service`.

- Dix variantes

Pour voir les images préconfigurées dans le référentiel qui ont été incluses dans OpenStack Unified Archive, cliquez sur l'un des deux onglets suivants :

- Admin -> Système -> onglet Images.
- Projet -> Calcul -> onglet Images.

Pour répertorier les types d'instance Oracle Solaris automatiquement disponibles, cliquez sur Admin > Système > onglet Types d'instance.

Les présentations vidéo suivantes fournissent un aperçu complet du tableau de bord :

- [The OpenStack Dashboard - Part 1](#)
- [The OpenStack Dashboard - Part 2](#)

Pour plus d'informations sur les tâches que vous pouvez effectuer sur le tableau de bord, reportez-vous au [Chapitre 5, Utilisation du cloud](#).

Création de projets et d'utilisateurs

Votre première connexion à OpenStack en tant qu'utilisateur admin vous amène à la page d'accueil du projet demo. A partir de ce projet, vous pouvez procéder à la création d'autres projets.

▼ Création d'un projet et affectation des utilisateurs

Utilisez cette procédure pour créer de nouveaux projets ou locataires et les renseigner avec de nouveaux utilisateurs.

1. **Dans votre navigateur, connectez-vous en tant qu'administrateur du cloud sur l'URL similaire au lien suivant :**

`http://system/horizon/`

La page d'accueil du projet demo s'affiche.

Pour une brève présentation du tableau de bord, voir "[Exploration du tableau de bord](#)" à la page 66.

2. **Sur le panneau de gauche, cliquez sur l'onglet Identité -> Projets.**

Les projets par défaut demo et service sont affichés.

3. **Cliquez sur Créer un projet.**
4. **Sur l'onglet Informations du projet, spécifiez le nom et la description du projet.**
Une fois créé, le nouveau projet est ajouté à la liste des projets.
5. **Cliquez sur l'onglet Identité -> Utilisateurs.**
Les utilisateurs par défaut pour demo et service sont affichés.
6. **Cliquez sur Créer un utilisateur.**
7. **Indiquez les informations requises dans les champs appropriés.**
 - a. **Indiquez le nouveau nom d'utilisateur et le mot de passe affecté.**
 - b. **Dans le menu déroulant Projet primaire, sélectionnez le projet auquel appartiendra le nouvel utilisateur.**
 - c. **(Facultatif) Dans le menu déroulant Rôle, sélectionnez le rôle pour l'utilisateur du projet.**
Par défaut, un nouvel utilisateur d'un projet a un rôle de membre.

▼ Alimentation d'un projet avec des utilisateurs existants

Utilisez cette procédure pour ajouter des utilisateurs existants à un projet nouvellement créé.

1. **Sur le panneau de gauche de l'écran par défaut, cliquez sur l'onglet Identité -> Projets.**
2. **Cliquez sur Editer les utilisateurs pour le projet auquel vous voulez ajouter des utilisateurs existants.**
La boîte de dialogue Editer le projet s'affiche.
3. **Sous la liste Utilisateurs, cliquez sur le signe plus (+) à droite du nom d'utilisateur que vous voulez ajouter au projet.**
Par défaut, l'utilisateur ajouté a un rôle de membre sur ce projet.

Remarque - N'ajoutez pas ces utilisateurs du projet service à d'autres projets :

- glance
 - cinder
 - swift
 - neutron
 - nova
 - ec2
-

4. **(Facultatif) Pour modifier le rôle de n'importe quel utilisateur du projet, procédez comme suit :**
 - a. **Dans la liste des membres du projet, ouvrez le menu déroulant pour l'utilisateur dont vous voulez modifier le rôle.**
 - b. **Sélectionnez le nouveau rôle que vous voulez affecter à l'utilisateur.**

Dans l'exemple suivant, les membres du projet actuel sont a1user et admin. L'utilisateur admin dispose de privilèges d'administration sur le projet. Vous pouvez affecter à la fois les rôles member et admin à un utilisateur.

Création de réseaux internes pour les projets

Un projet peut avoir plusieurs réseaux internes, chacun servant une instance de machine virtuelle correspondante. Par défaut, la communication utilisateur n'est limitée qu'au sein du réseau. Votre site doit avoir un réseau opérationnel pour que vous puissiez configurer les réseaux cloud.

▼ Configuration d'un réseau pour un projet

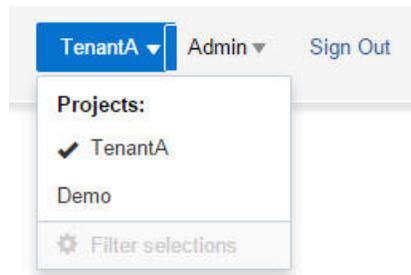
Pour créer des réseaux internes pour un projet, vous devez au minimum être membre du projet. Vous n'avez pas besoin de privilèges d'administration pour effectuer cette procédure.

1. Dans votre navigateur, connectez-vous au tableau de bord Horizon à l'aide de l'URL similaire à l'adresse suivante :

<http://system/horizon/>

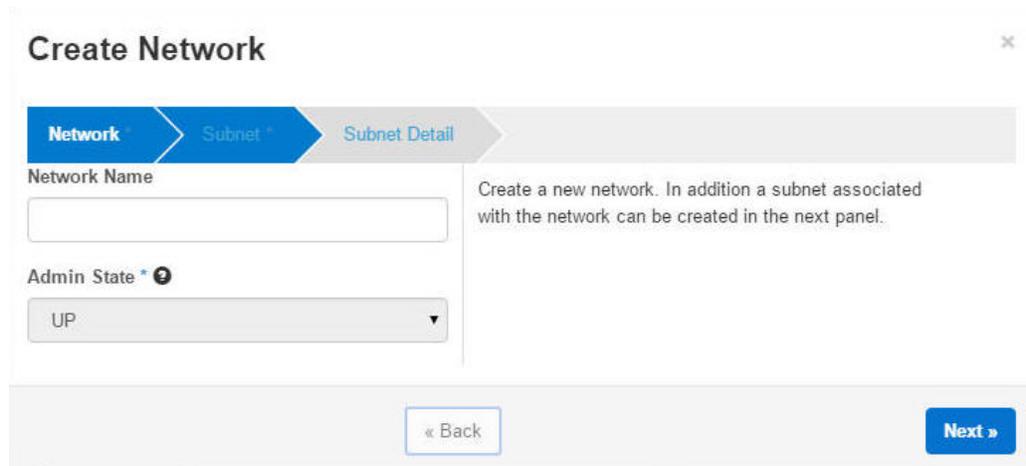
2. **Assurez-vous d'être connecté au bon projet en vérifiant le nom du projet dans le coin supérieur droit de l'écran par défaut.**

L'exemple suivant montre que TenantA est sélectionné parmi les deux projets disponibles. L'utilisateur admin est actuellement connecté.



3. **Sur le panneau de gauche, cliquez sur Projet -> Réseau -> onglet Réseaux, puis sur Créer un réseau.**

La boîte de dialogue Créer un réseau s'affiche.



4. **Cliquez sur chaque onglet et fournissez les informations requises.**

Remarque - Cliquez sur Suivant pour afficher l'écran de chaque onglet.

Chaque onglet vous invite à entrer les informations suivantes :

- Onglet Réseau
 - Nom du réseau
 - Admin State – acceptez la valeur par défaut.
- Onglet Sous-réseau
 - Nom du sous-réseau
 - Adresse réseau
 - Version IP
 - IP de la passerelle – laissez vide pour accepter la valeur par défaut.
- Onglet Détails du sous-réseau
 - Utiliser DHCP – à désélectionner si vous n'utilisez pas DHCP.
 - Pools d'allocation
 - Serveurs DNS
 - Routes d'hôte

Après avoir fourni les informations et cliqué sur le bouton Créer, l'écran Réseaux affiche les réseaux et leurs sous-réseaux comme dans l'exemple suivant :

Networks

Networks

<input type="checkbox"/>	Name	Subnets Associated
<input type="checkbox"/>	HR	hr_subnet 10.132.30.0/24
<input type="checkbox"/>	ENG	eng_subnet 10.132.35.0/24

Displaying 2 items

Étapes suivantes Si vous voulez qu'un réseau interne se connecte au réseau public, ajoutez ce sous-réseau au réseau externe du cloud. Reportez-vous à la section "[Connectivité externe aux réseaux internes](#)" à la page 57.

▼ Association d'une adresse IP flottante à un projet

Une partie de la configuration du réseau externe concerne la création des adresses IP flottantes. Reportez-vous à la section "[Création d'un réseau externe](#)" à la page 54. Utilisez cette procédure pour allouer certaines de ces adresses IP à un projet.

1. **Sur le panneau de gauche, cliquez sur Projet -> Calcul -> onglet Accès et Sécurité.**

2. **Cliquez sur l'onglet IP flottantes.**

3. **Cliquez sur le bouton Allouer une adresse IP au projet.**

La boîte de dialogue Allouer une IP flottante apparaît.

4. **Dans le menu déroulant, sélectionnez un pool à partir duquel allouer l'IP flottante.**

5. **Cliquez sur le bouton Allocation d'IP dans la boîte de dialogue.**

Une adresse IP est ajoutée à la liste des IP flottantes. Vous pouvez allouer autant d'adresses IP que vous le souhaitez ou qu'autorisées par le quota.

Création et initialisation d'une instance de machine virtuelle

Pour effectuer les procédures de cette section, vous devez au minimum être un membre valide du projet. Vous n'avez pas besoin de privilèges d'administration.

▼ Création d'une paire de clés SSH

1. **Dans votre navigateur, connectez-vous au tableau de bord Horizon à l'aide de l'URL similaire à l'adresse suivante :**

`http://system/horizon/`

2. **Assurez-vous d'être connecté au bon projet en vérifiant le nom du projet dans le coin supérieur droit de l'écran par défaut.**

3. **Sur le panneau de gauche, cliquez sur Projet -> Calcul -> onglet Accès et Sécurité.**

4. **Sur l'onglet Paires de clés, déterminez si vous devez créer une paire de clés ou l'importer.**
 - **Créez une paire de clés.**
 - a. **Cliquez sur le bouton Créer une paire de clés.**
 - b. **Spécifiez un nom dans le champ Nom de la paire de clés.**
 - c. **Cliquez sur le bouton Créer une paire de clés.**

La nouvelle paire de clés est téléchargée automatiquement. Si ce n'est pas le cas, cliquez sur le lien indiqué pour télécharger la paire de clés.
La nouvelle paire de clés est répertoriée dans l'onglet Paire de clés du panneau Accès et Sécurité.
 - **Importez une paire de clés.**
 - a. **Cliquez sur le bouton Importer une paire de clés.**
 - b. **Spécifiez un nom dans le champ Nom de la paire de clés.**
 - c. **Dans une fenêtre de terminal, copiez le contenu du fichier `.ssh/id_rsa.pub` de l'utilisateur `root` et collez-le dans le champ Clé publique.**
 - d. **Cliquez sur le bouton Importer une paire de clés.**

La nouvelle paire de clés est répertoriée dans l'onglet Paire de clés du panneau Accès et Sécurité.

▼ Création d'une instance de machine virtuelle

Avant de commencer

Assurez-vous que vous disposez d'une paire de clés SSH. Reportez-vous à "[Création d'une paire de clés SSH](#)" à la page 74.

Assurez-vous qu'un réseau interne est défini. Reportez-vous à la section "[Création de réseaux internes pour les projets](#)" à la page 71.

1. **Dans votre navigateur, connectez-vous au tableau de bord Horizon à l'aide de l'URL similaire à l'adresse suivante :**

`http://system/horizon/`
2. **Assurez-vous d'être connecté au bon projet en vérifiant le nom du projet dans le coin supérieur droit de l'écran par défaut.**

3. **Sur le panneau de gauche, cliquez sur Projet -> Calcul -> onglet Instances, puis sur Lancer une instance.**

La boîte de dialogue Lancer une instance s'affiche.

Launch Instance
✕

Details

Access & Security

Networking

Availability Zone

nova ▾

Instance Name

Flavor

Oracle Solaris kernel zone - tiny ▾

Instance Count

1

Instance Boot Source

--- Select source --- ▾

Specify the details for launching an instance.

The chart below shows the resources used by this project

Flavor Details

Name	Oracle Sola...
VCPUs	1
Root Disk	10 GB
Ephemeral Disk	0 GB
Total Disk	10 GB
RAM	2,048 MB

Project Limits

Number of Instances

Number of VCPUs

Total RAM

Cancel

Launch

4. **Fournissez les informations quand vous y êtes invité pour chaque onglet.**

Indiquez les informations pour les champs suivants :

- Onglet Détails
 - Nom de l'instance

- Type d'instance - dans le menu déroulant, sélectionnez le type d'instance approprié. Si ce système OpenStack est une zone de noyau et non un système à chaud, vous devez alors sélectionner une variante de zone non globale.
- Source de l'instance de démarrage - dans le menu déroulant, sélectionnez Démarrer depuis une image. Sélectionnez le nom de l'image que vous voulez utiliser. Si ce système OpenStack est une zone de noyau et non un système à chaud, vous devez alors sélectionner une image de zone non globale.

Le type d'instance doit correspondre au type d'image. Par exemple, si l'image est celle d'une zone non globale, le type d'instance doit également être une zone non globale.

- Onglet Accès et Sécurité – sélectionnez la paire de clés à utiliser.
- Onglet Réseaux – parmi les réseaux disponibles, sélectionnez celui auquel vous voulez associer la nouvelle machine virtuelle.

5. Cliquez sur le bouton Lancer dans la partie inférieure de la boîte de dialogue.

La nouvelle instance de machine virtuelle est créée, installée et amorcée.

Cette étape peut prendre un certain temps.

6. Associez l'adresse IP flottante à la nouvelle instance de machine virtuelle.

Vous pouvez effectuer les opérations suivantes pendant que la nouvelle instance de machine virtuelle est en cours d'installation. L'instance de machine virtuelle doit avoir une adresse IP flottante associée pour que les utilisateurs puissent s'y connecter.

a. Dans le menu déroulant de la colonne Actions, sélectionnez Associer une adresse IP flottante.

La boîte de dialogue Gérer les associations d'IP flottantes s'ouvre.

b. Sélectionnez une adresse dans la menu déroulant d'adresses IP.

Si aucune adresse IP n'est disponible, cliquez sur le bouton +. Reportez-vous à ["Association d'une adresse IP flottante à un projet" à la page 74.](#)

c. Sélectionnez le port correspondant à la machine virtuelle que vous avez créée.

d. Cliquez sur le bouton Associer dans la partie inférieure de la boîte de dialogue.

Étapes suivantes

- Cliquez sur Instances puis sur le nom de l'instance pour afficher les informations détaillées de l'instance et le journal de console de l'instance. Rechargez la page pour voir les mises à jour du journal.
- Cliquez sur Volumes pour afficher les volumes Cinder qui ont été créés.
- Cliquez sur Topologie réseau pour obtenir une représentation du réseau du cloud, y compris tous les segments de sous-réseau, les routeurs virtuels et les instances actives.

- Cliquez sur Images & Clichés pour visualiser les archives d'ensemble qui ont été téléchargés dans le magasin d'images Glance.
- Lorsque l'installation d'une nouvelle instance de machine virtuelle est terminée et est en statut Actif, connectez-vous à l'instance. La commande suivante se connecte à la zone en tant que root à l'aide de la paire de clés et de l'adresse IP flottante :

```
# ssh root@floating-IP-address
```

▼ Ajout d'utilisateurs à une instance de machine virtuelle

Dans Oracle Solaris, les instances de machine virtuelle utilisent la technologie des zones d'Oracle Solaris pour vous permettre de provisionner des machines virtuelles dans le cloud. Pour ajouter des utilisateurs à l'instance de machine virtuelle, vous devez exécuter des commandes en tant qu'administrateur de zone. Ces étapes ne sont pas prises en charge dans le tableau de bord. En conséquence, vous devez avoir accès à une fenêtre de terminal.

Avant de commencer

Obtenez l'adresse IP flottante du réseau externe à laquelle l'instance de machine virtuelle est associée en cliquant sur Admin -> Système -> onglet Instances dans le tableau de bord.

1. Dans une fenêtre de terminal, répertoriez les instances de machine virtuelle.

```
# zoneadm list -cv
```

Les noms des machines virtuelles dans le cloud ont le préfixe instance.

2. Connectez-vous à la zone spécifique.

```
# zlogin zonename
```

3. Créez un répertoire personnel pour l'utilisateur.

```
root@zone# mkdir -p /export/home/username
```

4. Créez l'utilisateur.

```
root@zone# useradd -d home-dir options
```

où *home-dir* est le répertoire que vous avez créé pour l'utilisateur. Pour les autres options que vous pouvez utiliser avec la commande `useradd`, reportez-vous à la page de manuel [useradd\(8\)](#).

5. Pour créer le mot de passe de l'utilisateur, exécutez la commande suivante et suivez les invites.

```
root@zone# passwd username
```

6. (Facultatif) Vérifiez que le mot de passe a été créé.

```
root@zone# grep username /etc/passwd
```

7. Pour créer le mot de passe root, exécutez la commande suivante et suivez les invites.

```
zone# passwd root
```

8. Quittez la zone et déconnectez-vous.**9. Connectez-vous à Secure Shell pour la machine virtuelle.**

```
# ssh username@floating-IP
```

où *floating-IP* est l'adresse IP flottante associée à la machine virtuelle.

Exemple 4 Ajout d'un utilisateur à une instance de machine virtuelle

Dans cet exemple, le nom d'utilisateur *jsmith* est ajouté en tant qu'utilisateur de VM1.

```
# zoneadm list -cv
ID NAME          STATUS  PATH                                BRAND  IP
0  global          running /                                      solaris shared
6  instance-00000006 running /system/zones/instance-00000006 solaris excl
-  myzone          installed /system/zones/myzone              solaris excl
```

```
# zlogin instance-00000006
[Connected to zone 'instance-00000006' pts/3]
Last login: Wed Jan  6 14:31:18 2016 on pts/2
Oracle Corporation      SunOS 5.11      11.3      September 2015
```

```
root@VM1# mkdir -p /export/home/jsmith
root@VM1# useradd -d /export/home/jsmith -m -s /usr/bin/bash jsmith
```

L'utilisateur *jsmith* est créé avec *bash* comme shell par défaut.

```
root@VM1# passwd jsmith
New Password: password
Re-enter new Password: password
passwd: password successfully changed for jsmith
```

```
root@VM1# passwd root
New Password: password
Re-enter new Password: password
passwd: password successfully changed for root
```

```
root@VM1# exit
logout
```

```
[Connection to zone 'instance-00000006' pts/3 closed]
```

```
# ssh jsmith@10.132.10.9
```

Gestion des variantes

Une *variante* est un type d'instance de machine virtuelle, ou modèle matériel virtuel. Une variante spécifie un ensemble de ressources d'une machine virtuelle, telles que le nombre de CPU virtuelles, la quantité de mémoire, et l'espace disque, affectées à une instance de machine virtuelle. Sur Oracle Solaris, la variante inclut aussi la marque de la zone sous-jacente : `solaris` pour les zones non globales et `solaris-kz` pour les zones de noyau. Un exemple de variante d'instance est une zone de noyau avec 16 CPU virtuelles et 16384 Mo de RAM.

Pour obtenir des informations générales sur les variantes, reportez-vous à la section [Variantes](#) du *OpenStack Cloud Administrator Guide*.

Affichage des informations sur les variantes

Quand vous vous connectez au tableau de bord en tant qu'administrateur de cloud, vous pouvez afficher les types d'instance disponibles à partir de l'onglet Admin -> Système -> Types d'instance.

FIGURE 5 Types d'instance dans Oracle Solaris pour Oracle OpenStack

<input type="checkbox"/>	Flavor Name	VCPUs	RAM	Root Disk	Ephemeral Disk	Swap Disk	ID	Public	Metadata	Actions
<input type="checkbox"/>	Oracle Solaris kernel zone - tiny	1	2048MB	10GB	0GB	0MB	1	Yes	Yes	Edit Flavor ▾
<input type="checkbox"/>	Oracle Solaris non-global zone - tiny	1	2048MB	10GB	0GB	0MB	6	Yes	Yes	Edit Flavor ▾
<input type="checkbox"/>	Oracle Solaris non-global zone - small	4	3072MB	20GB	0GB	0MB	7	Yes	Yes	Edit Flavor ▾
<input type="checkbox"/>	Oracle Solaris kernel zone - small	4	4096MB	20GB	0GB	0MB	2	Yes	Yes	Edit Flavor ▾
<input type="checkbox"/>	Oracle Solaris non-global zone - medium	8	4096MB	40GB	0GB	0MB	8	Yes	Yes	Edit Flavor ▾
<input type="checkbox"/>	Oracle Solaris kernel zone - medium	8	8192MB	40GB	0GB	0MB	3	Yes	Yes	Edit Flavor ▾
<input type="checkbox"/>	Oracle Solaris non-global zone - large	16	8192MB	40GB	0GB	0MB	9	Yes	Yes	Edit Flavor ▾
<input type="checkbox"/>	Oracle Solaris kernel zone - large	16	16384MB	40GB	0GB	0MB	4	Yes	Yes	Edit Flavor ▾
<input type="checkbox"/>	Oracle Solaris non-global zone - xlarge	32	16384MB	80GB	0GB	0MB	10	Yes	Yes	Edit Flavor ▾
<input type="checkbox"/>	Oracle Solaris kernel zone - xlarge	32	32768MB	80GB	0GB	0MB	5	Yes	Yes	Edit Flavor ▾

Displaying 10 items

Pour plus d'informations sur ces colonnes, reportez-vous à [OpenStack Command-Line Interface Reference](#).

Modification des spécifications de variante

Sous la colonne Actions de chaque type d'instance, cliquez sur l'une des trois premières options pour afficher les propriétés de ce type d'instance ainsi que pour pouvoir les modifier. Trois actions sont disponibles :

- L'option `Editer le type d'instance` affiche des informations sur le type d'instance ainsi que les propriétés que vous pouvez modifier. Sur l'onglet `Accès au type d'instance`, vous pouvez limiter les projets qui peuvent accéder au type d'instance. Le paramètre par défaut est `Aucun`, qui signifie que le type d'instance est public et que tous les projets peuvent y accéder.
- L'option `Modifier l'accès` ouvre directement l'onglet `Accès au type d'instance` du type d'instance afin que vous puissiez modifier le paramètre d'accès.
- L'option `Mettre à jour les métadonnées` permet de modifier les métadonnées du type d'instance.

Une fois que vous avez modifié les spécifications d'un type d'instance, ces changements s'appliquent à tous les invités créés ultérieurement qui utiliseront le type d'instance modifié.

Le tableau de bord ne permet pas d'apporter toutes les modifications au type d'instance. Par exemple, les clés de la propriété `extra_specs` peuvent uniquement être révisées avec la ligne de commande. Les clés de la propriété `font` référence à un jeu de propriétés de zone qui sont généralement configurées avec la commande `zonecfg` et qui sont prises en charge dans OpenStack.

Les clés suivantes sont prises en charge dans les types d'instance de zones de noyau et de zones non globales :

- `zonecfg:bootargs`
- `zonecfg:brand`
- `zonecfg:hostid`
- `zonecfg:cpu-arch`

Les clés suivantes ne sont prises en charge que dans les types d'instance de zones non globales :

- `zonecfg:file-mac-profile`
- `zonecfg:fs-allowed`
- `zonecfg:limitpriv`

Pour une description de ces propriétés de configuration de zone, reportez-vous à la page de manuel [zonecfg\(8\)](#).

Remarque - Toutes les propriétés de configuration de zone ne sont pas prises en charge dans OpenStack

En outre, la clé `sc_profile` n'est modifiable qu'à partir de la ligne de commande. Vous utilisez cette clé pour spécifier un profil de configuration système pour le type d'instance.

Pour modifier un type d'instance à partir de la ligne de commande, utilisez la syntaxe suivante :

```
nova flavor-key flavor action key=value [key=value ...]
```

flavor Le nom ou l'ID de la variante.

action set ou unset

key=value La *key* est le nom de la spécification. La *value* est la nouvelle valeur pour cette spécification. Si l'*action* est unset, n'indiquez que la *key*.

Par exemple, pour définir un fichier de configuration système spécifique pour le huitième type d'instance de la liste (Oracle Solaris kernel zone - large), exécutez la commande suivante :

```
$ nova flavor-key 4 set sc_profile=/system/volatile/profile/sc_profile.xml
```

Pour plus d'informations sur la suppression et la création de variantes, reportez-vous au [OpenStack Admin User Guide](#).

▼ Modification de la propriété `extra_specs` d'un type d'instance

1. Définissez les variables shell globales pour Neutron.

```
controller# export OS_USERNAME=nova
controller# export OS_PASSWORD=service-password
controller# export OS_PROJECT_NAME=service
controller# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0
```

2. Affichez les types d'instance disponibles.

```
controller# nova flavor-list
```

3. Notez l'ID du type d'instance que vous modifiez.

4. Modifiez la clé `extra_specs` de ce type d'instance.

```
controller# nova flavor-key flavor action es-key=value
```

où *es-key* se rapporte à la clé spécifique de la propriété `extra_specs`.

5. (Facultatif) Affichez les propriétés du type d'instance.

```
controller# nova flavor-show flavor
```

Exemple 5 Modification de la clé `zonecfg:bootargs`

Cet exemple montre comment modifier la clé `zonecfg:bootargs` du type d'instance Oracle Solaris non-global zone - medium, dont l'ID est 8.

Pour préserver l'espace, les colonnes `RTX_Factor` et `Is_Public` sont modifiées à partir de l'exemple de sortie `nova flavor-list` suivant.

```
controller# nova flavor-list
```

ID	Name	Memory_MB	Disk	Ephemeral	Swap	VCPUs
1	Oracle Solaris kernel zone - tiny	2048	10	0		1
10	Oracle Solaris non-global zone - xlarge	16384	80	0		32
2	Oracle Solaris kernel zone - small	4096	20	0		4
3	Oracle Solaris kernel zone - medium	8192	40	0		8
4	Oracle Solaris kernel zone - large	16384	40	0		16
5	Oracle Solaris kernel zone - xlarge	32768	80	0		32
6	Oracle Solaris non-global zone - tiny	2048	10	0		1
7	Oracle Solaris non-global zone - small	3072	20	0		4
8	Oracle Solaris non-global zone - medium	4096	40	0		8
9	Oracle Solaris non-global zone - large	8192	40	0		16

```
controller# nova flavor-key 8 set zonecfg:bootargs=-v
```

```
controller# nova flavor-show 8
```

Property	Value
OS-FLV-DISABLED:disabled	False
OS-FLV-EXT-DATA:ephemeral	0
disk	40
extra_specs	{"zonecfg:brand": "solaris"} u'zonecfg:bootargs': u'-v'}
id	8
name	Oracle Solaris non-global zone - medium
os-flavor-access:is_public	True
ram	4096
rxtx_factor	1.0
swap	
vcpus	8

bootargs modifié

Gestion des instances de machine virtuelle

Cette section décrit la procédure de modification des instances de machine virtuelle que vous avez créées sur le cloud, telles que la migration ou le redimensionnement des machines virtuelles et la définition des options d'amorçage de machine virtuelle.

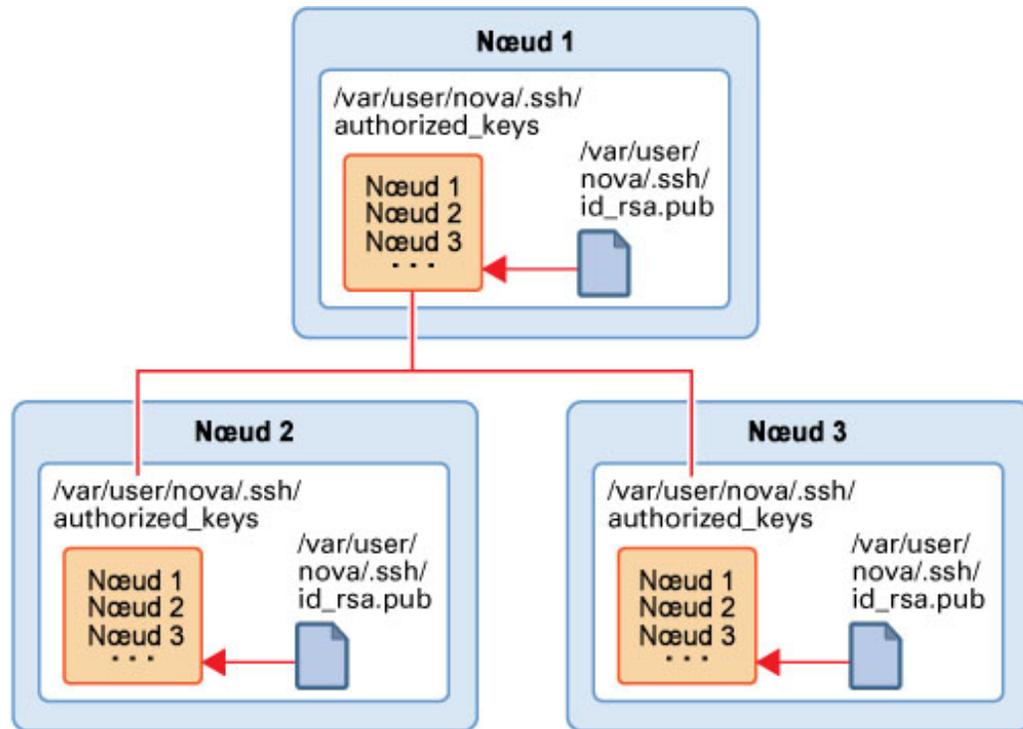
Migration et évacuation des instances de machine virtuelle

Dans Oracle OpenStack pour Oracle Solaris, les instances de machine virtuelle sont des zones de noyau ou des zones non globales, qui sont des environnements virtuels denses et évolutifs d'Oracle Solaris. La prise en charge de la migration en direct des zones a été étendue aux instances de machine virtuelle Nova.

Vous pouvez démarrer la migration en direct à l'aide du tableau de bord du service Horizon ou de la commande `nova`. Le planificateur sélectionne l'hôte cible de la migration parmi les noeuds participants. Pour garantir la sécurité, le mécanisme de migration sélectionne automatiquement l'algorithme de chiffrement approprié pour effectuer la migration. Toutefois, un paramètre dans le fichier `/etc/nova/nova.conf` vous permet de sélectionner le chiffrement que vous préférez utiliser.

Pour plus d'informations sur la migration en direct des zones de noyau, reportez-vous au manuel *Création et utilisation des zones de noyau d'Oracle Solaris* dans la bibliothèque correspondant à votre version d'Oracle Solaris dans la [Operating Systems Documentation](#). Reportez-vous également aux pages de manuel [zoneadm\(8\)](#) et [solaris-kz\(7\)](#).

Pour que la migration des noeuds réussisse, vous devez vous assurer que la clé SSH de chaque noeud de calcul est ajoutée au fichier des clés autorisées de chaque noeud. Ainsi, les noeuds partagent des fichiers identiques de clés autorisées, comme l'illustre la figure suivante :



Pour préparer la migration des instances de machine virtuelle, procédez comme suit :

1. Sur chaque noeud, créez une clé SSH.

```
# su - nova -c "ssh-keygen -N '' -f /var/user/nova/.ssh/id_rsa -t rsa"
```

2. Réunissez tous les fichiers de clés provenant des différents noeuds dans un emplacement commun dans l'un des noeuds.
3. Combinez toutes les clés dans le fichier `authorized_keys`.

Par exemple :

```
# cat nova(1)/id_rsa.pub nova(n)/id_rsa.pub >> /var/user/nova/.ssh/authorized_keys
```

où `nova(1)` à `nova(n)` représentent les clés SSH des noeuds participants.

4. Distribuez le fichier `authorized_keys` dans les répertoires `/var/user/nova/.ssh` de tous les autres noeuds participants.
5. En option, spécifiez le cryptage à utiliser lors de la migration dans le paramètre `live_migration_cipher` du fichier `/etc/nova/nova.conf` de chaque noeud de calcul. Toutefois, si vous préférez que le processus sélectionne automatiquement un cryptage approprié, ne définissez pas le paramètre.

Pour effectuer la migration en direct d'un serveur actif sur une autre machine, utilisez la syntaxe suivante après avoir défini les variables de shell globales :

```
# nova live-migration server [host]
```

où *server* est le nom ou l'ID du serveur, tandis que le paramètre facultatif *host* est le nom du serveur de destination.

En cas de défaillance du noeud de l'instance actuelle, ou si le service Nova est lui-même désactivé pendant un certain temps, vous pouvez déplacer ou évacuer l'instance et la reconstruire sur un autre noeud. Ainsi, vous pouvez récupérer le noeud.

Remarque - Vous pouvez évacuer uniquement des zones de noyau. L'évacuation est prise en charge dans les configurations où les périphériques root sont sur le stockage partagé.

Pour effectuer la migration en direct de toutes les instances VM d'un hôte vers un autre, utilisez la syntaxe suivante après avoir défini les variables de shell globales :

```
# nova host-evacuate-live [--target-host target] server
```

Redimensionnement d'une instance de machine virtuelle

La taille d'une machine virtuelle est indiquée par le type d'instance sur la base duquel la machine virtuelle est lancée. Pour les étapes permettant de créer une instance de machine virtuelle, reportez-vous à la section "[Création d'une instance de machine virtuelle](#)" à la page 75. La figure suivante présente les détails d'un exemple de machine virtuelle hr_vm1 tel qu'affiché sur le tableau de bord Horizon.

FIGURE 6 Taille de l'instance de machine virtuelle

Instances

Instances

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Size
<input type="checkbox"/>	hr_vm1	Solaris Non-global Zone	10.132.20.5 10.132.10.10	Oracle Solaris non-global zone - tiny

Displaying 1 item

La figure indique que la taille de hr_vm1 est la taille qui a été définie pour le type d'instance Oracle Solaris non-global zone - tiny. Pour afficher des détails sur un type d'instance spécifique, cliquez sur le nom de l'instance :

- RAM = 2 Go
- VCPU = 1 VCPU
- Disque = 10 Go

Le redimensionnement d'une instance implique l'utilisation d'un type d'instance différent pour l'instance.

Par défaut, le processus de redimensionnement crée l'instance redimensionnée sur un nouveau noeud. Toutefois, si vous procédez au redimensionnement vers un type d'instance de zone non globale, vous devez définir le processus de redimensionnement pour qu'il s'effectue sur le même noeud. Les zones non globales doivent être de la même version que leur zone globale d'origine. Le redimensionnement vers un type d'instance de zone non globale et la création de l'instance dans un autre noeud risquent de placer l'instance dans un noeud qui a une version de zone globale différente. Pour prévenir ce risque, modifiez le paramètre suivant dans le fichier `/etc/nova/nova.conf` qui crée l'instance récemment dimensionnée sur le même noeud :

```
allow_resize_to_same_host=true
```

Remarque - Le risque ne s'applique pas aux zones de noyau. Vous pouvez donc redimensionner en toute sécurité une zone de noyau vers un autre noeud.

▼ Redimensionnement d'une instance de machine virtuelle

Avant de commencer

Vous devez connaître la taille actuelle de l'instance de machine virtuelle que vous voulez modifier. Vous pouvez obtenir ces informations à partir du tableau de bord. Pour un exemple, voir [Figure 6, "Taille de l'instance de machine virtuelle"](#).

1. Définissez les variables shell globales.

```
# export OS_USERNAME=nova
# export OS_PASSWORD=service-password
# export OS_PROJECT_NAME=service
# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0"
```

2. Si vous procédez au redimensionnement vers un type d'instance de zone non globale, modifiez le paramètre suivant dans le fichier `/etc/nova/nova.conf` comme suit :

```
allow_resize_to_same_host=true
```

3. Affichez la liste des types d'instance.

```
# openstack flavor list
```

4. Notez le numéro de l'ID du nouveau type d'instance que vous voulez que l'instance de machine virtuelle utilise.

5. Modifiez l'instance de machine virtuelle pour qu'elle utilise le nouveau type d'instance en référant le numéro de l'ID du type d'instance.

```
# openstack server resize --flavor flavor-ID instance-name
```

Par exemple, supposons qu'actuellement, la taille de l'instance `hr_vm1` soit définie sur le type d'instance `Oracle Solaris non-global zone - tiny`, dont le numéro d'ID est 6. Vous voulez redimensionner l'instance pour le type d'instance `Oracle Solaris non-global zone - medium`, dont le numéro d'ID est 8. Vous devez taper ce qui suit :

```
# openstack server resize --flavor 8 hr_vm1
```

Avant de spécifier le nom de l'instance, vous pouvez insérer des options supplémentaires telles que `--wait` pour attendre la fin du redimensionnement ou `--confirm` pour confirmer la fin du redimensionnement du serveur.

6. (Facultatif) Vérifiez sur le tableau de bord que la taille de l'instance a changé pour le nouveau type d'instance.

Pour effectuer cette vérification, vous pouvez également utiliser la commande suivante :

```
# openstack server show instance-name
```

Vérifiez que la valeur du champ `flavor` spécifie le nouveau type d'instance de l'étape précédente.

L'exemple suivant illustre la sortie extraite. L'instance `hr_vm1` est créée dans un projet appelé `TenantA` et a été redimensionnée pour `Oracle Solaris non-global zone - medium`.

```
# export OS_USERNAME=admin
# export OS_PASSWORD=admin-password
# export OS_PROJECT_NAME=TenantA
# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0"

# openstack server show hr_vm1
+-----+-----+
| Property      | Value                                     |
+-----+-----+
| ...           |                                           |
| created       | 2016-01-26T12:38:47Z                     |
| flavor        | Oracle Solaris non-global zone - medium (8) |
| ...           |                                           |
+-----+-----+
```


Options pour la configuration et le déploiement de Cinder

Ce chapitre décrit les autres méthodes de configuration de Cinder et du composant de stockage de votre installation OpenStack.

Remarque - Les procédures de la section "[Configuration du noeud de stockage](#)" à la page 46 qui permettent de configurer le stockage de blocs demeurent la méthode par défaut pour configurer le noeud de stockage. Les options de ce chapitre proposent des alternatives pour le déploiement du composant de stockage. Toutefois, il n'est pour l'instant pas recommandé d'utiliser simultanément la configuration de stockage par défaut et la configuration alternative pour installer Cinder sur votre structure de cloud.

Ce chapitre comprend les sections suivantes :

- "[Déploiement de systèmes distants pour le stockage](#)" à la page 91
- "[Spécification de volumes d'initialisation pour les noeuds de calcul](#)" à la page 96
- "[Utilisation du pilote NFS Cinder](#)" à la page 98
- "[Utilisation d'OpenStack avec Oracle ZFS Storage Appliance](#)" à la page 100

Déploiement de systèmes distants pour le stockage

Dans les versions précédentes d'OpenStack sans prise en charge du réseau de stockage (SAN), le service de volume Cinder devait être configuré pour s'exécuter sur l'hôte cible si le pilote iSCSI ZFS était utilisé.

Grâce à la prise en charge des réseaux de stockage (SAN) dans Cinder, vous pouvez utiliser SSH pour configurer plusieurs back-ends hôte de stockage et définir leur type de volume. Dans ce type de déploiement, le package Cinder ainsi que l'ensemble des services Cinder sont installés uniquement sur l'hôte de l'initiateur, qui est généralement l'hôte faisant également office de noeud de calcul.

Il n'est pas nécessaire d'installer le package OpenStack sur les hôtes cible distants. Ces hôtes fournissent simplement les disques LUN à l'hôte de l'initiateur basé sur COMSTAR.

Pour déployer correctement des systèmes distants pour le stockage, vous devez respecter les exigences suivantes :

- Configuration du fichier `/etc/cinder/cinder.conf`.
- Octroi du profil de droits approprié à l'utilisateur désigné.
- Installation manuelle de packages supplémentaires.

Les sections suivantes présentent ces exigences en détail.

Configuration du fichier `cinder.conf`

Après l'installation du package Cinder sur l'hôte de l'initiateur, modifiez la section `[DEFAULT]` du fichier `/etc/cinder/cinder.conf`.

Reportez-vous à l'exemple suivant pour comprendre les définitions figurant dans le fichier de configuration.

```
[DEFAULT]
my_ip = localhost
osapi_volume_workers = 1
auth_strategy = keystone
#num_volume_device_scan_tries = 6
os-volume-api-version = 2
scheduler_driver=cinder.scheduler.filter_scheduler.FilterScheduler

enabled_backends=zfsdriver-1, zfsdriver-2, zfsdriver-3, zfsdriver-4, zfsdriver-5

[zfsdriver-1]
volume_group=zvolumes-1
volume_driver=cinder.volume.drivers.solaris.zfs.ZFSVolumeDriver
volume_backend_name=ZFS_LOCAL
zfs_volume_base = rpool/cinder
san_is_local = True
debug=true
verbose=true
[zfsdriver-2]
volume_group=zvolumes-2
volume_driver=cinder.volume.drivers.solaris.zfs.ZFISCSIDriver
volume_backend_name=ZFS_REMOTE
zfs_volume_base = rpool/cinder
san_ip = 10.134.13.38
san_login = user-name
san_password = password
debug=true
verbose=true
[zfsdriver-3]
volume_group=zvolumes-3
volume_driver=cinder.volume.drivers.solaris.zfs.ZFSFCDriver
```

```

volume_backend_name=ZFS_REMOTE_FC
zfs_volume_base = rpool/fc
san_ip = 10.134.13.38
san_login = user-name
san_password = password
debug=true
verbose=true
[zfsdriver-4]
volume_group=zvolumes-4
volume_driver=cinder.volume.drivers.solaris.zfs.ZFSISCSIDriver
volume_backend_name=ZFS_REMOTE
zfs_volume_base = rpool/cinder/zq
san_ip = 10.134.13.38
san_login = user-name
san_password = password
debug=true
verbose=true
[zfsdriver-5]
volume_group=zvolumes-5
volume_driver=cinder.volume.drivers.solaris.zfs.ZFSISCSIDriver
volume_backend_name=ZFS_REMOTE
zfs_volume_base = rpool/zq
san_ip = 10.134.63.182
san_login = user-name
san_password = password
debug=true
verbose=true

```

enabled_backends Répertorie les hôtes back-end activés. Dans cet exemple, 5 hôtes back-end sont définis.

Les hôtes back-end utilisent 3 pilotes ZFS (ZFSVolumeDriver, ZFSISCSIDriver, ZFSFCDriver) sur 3 hôtes. Parmi ces hôtes, l'un est local (localhost) tandis que les autres sont distants (10.134.13.38 et 10.134.63.182).

volume_backend_name Définit un type de volume avec le nom indiqué. Ce paramètre identifie le type de volume. Toutefois, vous devez créer le type de volume manuellement à l'aide des commandes suivantes :

```

# cinder type-create vol-type
# cinder type-key vol-type set volume_backend_name=backend-name
# cinder create --display-name display --volume-type vol-type

```

Ces commandes exécutent respectivement les actions suivantes :

- Création d'un type de volume.
- Affectation du nom de back-end au nouveau type de volume.
- Création d'un volume en fonction du nouveau type de volume.

D'après l'exemple de fichier de configuration Cinder précédent, vous devez saisir les commandes suivantes :

```
# cinder type-create type-remote
# cinder type-key type-remote set volume_backend_name=ZFS_REMOTE
# cinder create --display-name t1 --volume-type type-remote
```

La dernière commande crée le volume t1 sur l'un des back-ends avec le nom ZFS_REMOTE en fonction des règles de filtrage.

Vous devez exécuter le même ensemble de commandes pour créer des types de volume pour ZFS_LOCAL et ZFS_REMOTE_FC.

`zfs_volume_base` Spécifie l'ensemble de données de base pour les nouveaux volumes ZFS sur chaque back-end de volume.

`san_is_local`
`san_ip san_login`
`san_password` Paramètres du pilote SAN sur lesquels sont basés tous les pilotes ZFS. Ces paramètres doivent être définis pour activer l'exécution de commandes sur un hôte back-end localement ou à distance avec SSH. Pour chaque back-end, définissez les paramètres SAN de l'une des deux façons suivantes :

- Définissez uniquement `san_is_local = True`
- Définissez `san_ip`, `san_login` et `san_password` ensemble.

Ne définissez pas les quatre paramètres SAN.

`debug=true`
`verbose=true` Paramètres facultatifs pour le débogage. Vous pouvez omettre la configuration de ces paramètres.

Une fois que Cinder est correctement configuré, la liste des services affiche l'état de chaque service.

```
# cinder service-list
+-----+-----+-----+-----+-----+-----+-----+
| Binary | Host | Zone | Status | State | Updated_at | Disabled |
|         |      |     |        |      |             | Reason   |
+-----+-----+-----+-----+-----+-----+-----+
| cinder- |      |     |        |      | 2015-10-13 |          |
| backup  | host-2 | nova | enabled | up   | T19:22:45.000000 | None     |
|         |      |     |        |      |             |          |
| cinder- |      |     |        |      | 2015-10-13 |          |
| scheduler | host-2 | nova | enabled | up   | T19:22:43.000000 | None     |
|         |      |     |        |      |             |          |
| cinder- |      |     |        |      | 2015-10-13 |          |
| volume  | host-2 | nova | enabled | down | T18:31:41.000000 | None     |
|         |      |     |        |      |             |          |
| cinder- | host-2 |     |        |      | 2015-10-13 |          |
| volume  | @zfsdriver-1 | nova | enabled | up   | T19:22:46.000000 | None     |
|         |      |     |        |      |             |          |
| cinder- | host-2 |     |        |      | 2015-10-13 |          |
| volume  | @zfsdriver-2 | nova | enabled | up   | T19:22:47.000000 | None     |
|         |      |     |        |      |             |          |
```

cinder-	host-2				2015-10-13		
volume	@zfsdriver-3	nova	enabled	up	T19:22:48.000000	None	
cinder-	host-2				2015-10-13		
volume	@zfsdriver-4	nova	enabled	up	T19:22:47.000000	None	
cinder-	host-2				2015-10-13		
volume	@zfsdriver-5	nova	enabled	up	T19:22:48.000000	None	
cinder-	host-2				2015-10-13		
volume	@zfsdriver-6	nova	enabled	down	T18:32:55.000000	None	

Octroi des droits à l'utilisateur désigné

Pour que l'utilisateur défini par `san_login` puisse utiliser les cibles distantes, il doit disposer d'un profil de droits approprié. L'exemple suivant illustre la création du profil de droits pour l'utilisateur.

```
# useradd -s /usr/bin/pfbash -m jdoe
# passwd jdoe password
# profiles -p "Cinder Storage management"
profiles:Cinder Storage management> set desc="Cinder Storage management on target host"
profiles:Cinder Storage management> add profiles="File System Management"
profiles:Cinder Storage management> add auths="solaris.smf.modify.stmf"
profiles:Cinder Storage management> add cmd=/usr/sbin/itadm
profiles:Cinder Storage management:itadm> set euid=0
profiles:Cinder Storage management:itadm> end
profiles:Cinder Storage management> add cmd=/usr/sbin/fcadm
profiles:Cinder Storage management:itadm> set privs=file_dac_read,sys_devices
profiles:Cinder Storage management:itadm> end
profiles:Cinder Storage management> add cmd=/usr/sbin/fcinfo
profiles:Cinder Storage management:itadm> set privs=file_dac_read,sys_devices
profiles:Cinder Storage management:itadm> end
profiles:Cinder Storage management> add cmd=/usr/sbin/stmfadm
profiles:Cinder Storage management:stmfadm> set euid=0
profiles:Cinder Storage management:stmfadm> end
profiles:Cinder Storage management> add cmd=/usr/lib/rad/module/mod_zfsmgr.so.1
profiles:Cinder Storage management:mod_zfsmgr.so.1> set privs={zone}:/system/volatile/*, \
sys_config,sys_mount
profiles:Cinder Storage management:mod_zfsmgr.so.1> end
profiles:Cinder Storage management> add cmd=/usr/sbin/zfs
profiles:Cinder Storage management:zfs> set priv=sys_config,sys_mount
profiles:Cinder Storage management:zfs> end
profiles:Cinder Storage management> exit

# usermod -P "Cinder Storage management" jdoe
```

Pour plus d'informations sur les profils et les droits, reportez-vous au manuel *Sécurisation des utilisateurs et des processus dans Oracle Solaris* dans la bibliothèque correspondant à votre version d'Oracle Solaris dans la [Operating Systems Documentation](#).

Activation de l'hôte distant en tant que cible

Pour activer l'hôte distant en tant que cible, effectuez les étapes suivantes :

1. Installez le package `group/feature/storage-server` sur les hôtes distants.

```
# pkg install storage-server
```

2. Activez les services suivants à la fois sur l'initiateur et les hôtes distants :

- `svc:/system/stmf:default`
- `svc:/network/iscsi/target:default`
- `svc:/system/rad:remote`

Par exemple :

```
remote-host# svcadm enable stmf
remote-host# svcadm enable -r svc:/network/iscsi/target:default
remote-host# svcadm enable rad
```

3. Initialisez et configurez la liste de contrôle d'accès (ACL) pour le paramètre `zfs_volume_base` défini.

Par exemple, supposons que la définition `zfs_volume_base=rpool/fc` figure dans le fichier de configuration. Vous devez ensuite exécuter les commandes suivantes :

```
# chmod A+user:cinder:add_subdirectory:allow /rpool/fc
# zfs allow cinder clone,create,destroy,mount,snapshot rpool/fc
```

Spécification de volumes d'initialisation pour les noeuds de calcul

Dans une configuration utilisant plusieurs back-ends de stockage, ou dans un environnement comportant plusieurs racks ou dispositifs de stockage ZFS ZS3, vous voulez contrôler l'emplacement du volume root pour chaque nouvelle instance que vous créez. Dans OpenStack pour Oracle Solaris, ce contrôle s'effectue par le biais de deux paramètres dans le fichier `/etc/nova/nova.conf`.

▼ Création de volumes de stockage root pour des instances de calcul

Notez que cette procédure fait partie de la tâche générale consistant à définir plusieurs back-ends distants, comme décrit à la section "[Déploiement de systèmes distants pour le stockage](#)" à la page 91. Cette procédure utilise donc les mêmes exemples.

1. Définissez les back-ends Cinder dans le fichier `/etc/cinder/cinder.conf`.

Reportez-vous à l'exemple de la section "[Configuration du fichier cinder.conf](#)" à la page 92, en particulier à la spécification de `enabled_backends` ainsi qu'à la définition du nom de volume pour chaque back-end.

2. Si votre configuration utilise une zone de disponibilité Cinder, définissez-la également dans le fichier de configuration Cinder.

Par exemple :

```
[DEFAULT]
...
storage_availability_zone=cinder_az
```

3. Pour chaque back-end défini dans le fichier de configuration, créez les volumes correspondants, comme présenté dans l'exemple suivant :

```
# cinder type-create type-remote
# cinder type-key type-remote set volume_backend_name=ZFS_REMOTE
```

Vous devez exécuter le même ensemble de commandes pour créer d'autres types de volume si nécessaire. Le pilote `solariszones` crée alors le volume d'initialisation Cinder réel pour l'instance Nova.

4. Sur le noeud Cinder, redémarrez le service Cinder.

```
# svcadm restart cinder-volume:default
```

5. Dans le fichier `/etc/nova/nova.conf` de chaque noeud de calcul, définissez l'un des paramètres suivants, ou les deux, d'après le fichier de configuration Cinder.

- `boot_volume_type`
- `boot_volume_az`

Par exemple, en fonction des étapes précédentes, modifiez le fichier de configuration Nova comme suit :

```
boot_volume_type=type-remote
boot_volume_az=cinder_az
```

6. Redémarrez le service Nova sur le noeud de calcul.

```
compute-node# svcadm restart nova-compute
```

Utilisation du pilote NFS Cinder

Le pilote Cinder pour NFS est pris en charge dans Oracle Solaris. Ce pilote mappe les primitives Cinder et l'API vers le stockage d'arrière-plan fondamental. Ce pilote utilise en particulier NFS comme stockage d'arrière-plan.

Le pilote Cinder est chargé du provisionnement et d'autres tâches de manipulation administrative des périphériques de stockage uniquement. Toutefois, le pilote lui-même ne se trouve pas dans le chemin de données des opérations d'E/S. Le pilote n'autorise pas une instance à accéder à un périphérique de stockage au niveau du bloc. Des fichiers sont créés à la place sur un partage NFS et mappés vers des instances. Chaque fichier NFS fonctionne comme un périphérique en mode bloc.

Remarque - Actuellement, ce pilote prend en charge uniquement les zones de noyau. N'utilisez pas ce pilote dans des zones non globales.

▼ Procédure d'utilisation du pilote NFS Cinder

Avant de commencer

Pour utiliser le pilote, vous devez disposer d'un serveur NFS afin de créer des partages NFS pour les clients. Ce document ne donne aucune instruction de configuration pour un serveur NFS. Reportez-vous à une autre documentation NFS pour configurer le serveur.

Un seul serveur NFS suffit généralement. Toutefois, vous pouvez disposer de plusieurs serveurs NFS si nécessaire.

1. Ajoutez les partages NFS disponibles au fichier `/etc/cinder/nfs_shares`.

Répertoriez les partages NFS au format `host:share`. Par exemple :

```
nfs-server-system1:/scratch/volume1
nfs-server-system2:/scratch/volume2
```

2. Modifiez le fichier `/etc/cinder/cinder.conf`.

a. Spécifiez le pilote NFS à utiliser.

```
volume_driver=cinder.volume.drivers.solaris.nfs.ZfsNfsVolumeDriver
```

b. Vérifiez que le paramètre `nfs_shares_config` fait référence au fichier que vous avez utilisé à l'[Étape 1](#).

```
nfs_shares_config=/etc/cinder/nfs_shares
```

c. Configurez les autres paramètres NFS connexes si nécessaire.

- `nfs_mount_attempts` – nombre maximal de tentatives de montage d'un partage NFS avant la génération d'une erreur.
- `nfs_mount_point_base` – répertoire principal des points de montage de partage NFS.
- `nfs_oversub_ratio` – limite maximale du rapport entre l'espace de volume alloué et l'espace disponible dans un volume de destination. En cas de dépassement du rapport, le volume de destination n'est plus valide.
- `nfs_sparsed_volumes` – la valeur par défaut `True` crée les volumes sous forme de fichiers fragmentés. Sinon, les volumes sont créés sous forme de fichiers ordinaires.
- `nfs_used_ratio` – pourcentage d'utilisation réelle du volume sous-jacent auquel de nouveaux volumes ne peuvent plus être alloués au volume cible.
- `nfs_round_robin` – la valeur par défaut `True` planifie un tourniquet sur les partages NFS. Si ce paramètre n'est pas défini, le partage NFS disposant de la plus grande quantité d'espace disponible sera sélectionné pour le placement de volume.

3. Redémarrez le service Cinder.

```
# svcadm cinder-volume restart
```

Après le démarrage du service, les répertoires pour les partages NFS sont ajoutés au répertoire principal des points de montage du partage NFS.

4. Créez des volumes.

Remarque - Lors de l'utilisation du pilote NFS Cinder, la création d'instantanés n'est pas prise en charge.

a. Définissez les variables de shell nécessaires.

Par exemple :

```
# export OS_USERNAME=nova
# export OS_PASSWORD=service-password
# export OS_PROJECT_NAME=service
# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0
```

b. Créez le volume.

Par exemple :

```
# nova volume-create --display-name nfsvol 5
```

c. (Facultatif) Affichez les propriétés du volume.

Par exemple :

```
# nova volume-show nfsvol
```

Utilisation d'OpenStack avec Oracle ZFS Storage Appliance

Cette section décrit une configuration OpenStack utilisant Oracle ZFS Storage Appliance (ZFSSA) comme stockage d'arrière-plan.

À propos d'Oracle ZFS Storage Appliance

La gamme de produits Oracle ZFS Storage Appliance offre des services de données de fichiers et de données par blocs performants à des clients réseau, ainsi qu'un éventail de services de données pouvant être appliqués aux données stockées sur le système, y compris les technologies suivantes :

- Analyse – Vous permet d'observer de façon dynamique le comportement de votre système en temps réel et d'afficher les données sous forme de graphique.
- Pool de stockage hybride ZFS - Permet de gérer différents périphériques tels que les périphériques facultatifs de mémoire flash, les disques à grande capacité et faible consommation et la mémoire DRAM comme une hiérarchie de données unique.
- Prise en charge de divers matériel

Le package `cloud/openstack/cinder` d'Oracle OpenStack pour Oracle Solaris inclut le pilote Oracle ZFSSA iSCSI Cinder. Le pilote permet au ZFSSA d'être utilisé de façon transparente sous la forme d'une ressource de stockage en mode bloc pour le composant Cinder. En particulier, le pilote permet de créer des volumes iSCSI pouvant être alloués par un serveur Cinder à toute machine virtuelle instanciée par le service Nova. Pour utiliser Oracle ZFSSA pour le stockage, assurez-vous que le système s'exécute au minimum avec la version 2013.1.2.0 du logiciel ZFSSA.

Configurer OpenStack avec Oracle ZFSSA

Cette section ne couvre pas en détails les procédures de configuration d'Oracle ZFSSA. Pour plus d'informations sur le système, y compris les étapes de configuration, reportez-vous aux sources suivantes :

- Documentation du produit Oracle ZFSSA sur la [Oracle Help Center storage documentation page](#)

- [Using the Oracle ZFS Storage Appliance as Storage Back End for OpenStack Cinder](http://www.oracle.com/technetwork/server-storage/sun-unified-storage/documentation/openstack-cinder-zfssa-120915-2813178.pdf) (<http://www.oracle.com/technetwork/server-storage/sun-unified-storage/documentation/openstack-cinder-zfssa-120915-2813178.pdf>). Ce livre blanc détaille la configuration du système de stockage à utiliser dans le cloud.

Avant d'exécuter les procédures de configuration d'OpenStack avec Oracle ZFSSA, assurez-vous d'avoir créé le sous-réseau qui hébergera le trafic du stockage des données de l'hôte. Le scénario de ce chapitre utilise la figure [Figure 1, "Architecture de référence de la configuration à trois noeuds"](#) comme base de départ, mais avec une différence. Ici, le trafic des données de stockage est hébergé sur deux sous-réseaux, avec net2 et net3 comme interfaces pour ce trafic. Les deux sous-réseaux assurent la performance et la disponibilité du service de stockage de données.

Côté ZFSSA

Oracle ZFSSA utilise iSCSI pour échanger des commandes SCSI sur le réseau. Pour configurer la communication entre les noeuds iSCSI, vous devez avoir à disposition les informations suivantes :

- Nom DNS ou adresse IP du noeud SCSI. Le noeud SCSI comprend des noeuds initiateur et cible.
- Port Internet TCP iSCSI sur le noeud cible. Par défaut, le numéro de port est 3260.
- Noms qualifiés iSCSI (IQN) des noeuds cible et initiateur.
- Informations d'authentification facultatives utilisant le protocole CHAP (Challenge Handshake Authentication Protocol)

Pour obtenir les informations iSCSI appropriées pour cette étape préparatoire, utilisez la commande `iscsiadm` d'Oracle Solaris.

Ce document ne décrit pas les étapes de configuration du système. Pour les procédures de préparation du système à une utilisation opérationnelle, reportez-vous à la documentation de la version appropriée dans la [documentation des produits de stockage](#) d'Oracle.

Côté OpenStack

Le pilote OpenStack ZFSSA Cinder crée et gère les volumes en créant les LUN associés sur le système, en configurant les propriétés appropriées de ces LUN et en gérant la visibilité des LUN pour les noeuds de calcul OpenStack et les instances d'invité sur lesquels les volumes sont utilisés.

Après le paramétrage d'Oracle ZFSSA, vous pouvez ensuite configurer le pilote ZFSSA iSCSI Cinder.

▼ Configuration d'Oracle ZFSSA pour OpenStack

La procédure se sert du workflow `cinder.akwf` qui effectue les tâches suivantes :

- Création de l'utilisateur s'il n'existe pas.
- Définition des autorisations de rôle pour effectuer les opérations du pilote Cinder.
- Activation du service RESTful si le service est désactivé.

Avant de commencer

Configuration d'un pool sur Oracle ZFS Storage Appliance. Vous pouvez décider d'utiliser un pool préexistant.

1. Exécutez le workflow `cinder.akwf` à l'aide d'une des méthodes suivantes :

- **Exécutez le workflow à partir de la CLI.**

```
zfssa:maintenance workflows> download
zfssa:maintenance workflows download (uncommitted)> show
Properties:
    url = (unset)
    user = (unset)
    password = (unset)

zfssa:maintenance workflows download (uncommitted)> set url= "url-to-cinder.akwf-file"
    url = "url-to-cinder.akwf-file"
zfssa:maintenance workflows download (uncommitted)> commit
Transferred 2.64K of 2.64K (100%) ... done

zfssa:maintenance workflows> ls
Properties:
    showhidden = false

Workflows:

WORKFLOW      NAME                                     OWNER SETID ORIGIN
VERSION
workflow-000 Clear locks                                     root  false Oracle Corporation
1.0.0
workflow-001 Configuration for OpenStack Cinder Driver root  false Oracle Corporation
1.0.0

zfssa:maintenance workflows> select workflow-001

zfssa:maintenance workflow-001> execute
zfssa:maintenance workflow-001 execute (uncommitted)>

zfssa:maintenance workflow-001 execute (uncommitted)> set name=user
    name = user
zfssa:maintenance workflow-001 execute (uncommitted)> set password=password
    password = password
zfssa:maintenance workflow-001 execute (uncommitted)> commit
```

User openstack created.

Notez que pour *user* et *password*, les valeurs se présentent tel qu'elles sont définies pour les paramètres `san_login` et `san_password` dans le fichier `cinder.conf`.

- **Exécutez le workflow à partir de la BUI.**
 - a. **Sélectionnez Maintenance -> Workflow, et utilisez l'icône plus (+) pour télécharger un nouveau workflow.**
 - b. **Cliquez sur le bouton Parcourir et sélectionnez le fichier `cinder.akwf`.**
 - c. **Cliquez sur le bouton TELECHARGER pour terminer le téléchargement du workflow.**
 - d. **Cliquez sur la nouvelle ligne qui apparaît sur la page Workflows BUI pour exécuter le workflow du pilote Cinder.**

Le workflow invite à entrer un nom d'utilisateur et un mot de passe. Ce nom d'utilisateur et ce mot de passe seront également utilisés dans le fichier `cinder.conf` en tant que `san_login` et `san_password`.

2. Définissez les paramètres dans le fichier `/etc/cinder/cinder.conf`.

Spécifiez les propriétés requises suivantes dans le fichier `cinder.conf` :

Remarque - La liste suivante est partielle. Assurez-vous de passer en revue et de définir toutes les propriétés du fichier de configuration nécessaires à la réalisation de votre configuration spécifique.

- `volume_driver` – Assurez-vous que `cinder.volume.drivers.zfssa.zfssaiscsi.ZFSSAISCSIDriver` n'est pas commentée. Assurez-vous que les trois autres sélections sont commentées.
- `san_ip` – Nom ou adresse IP de l'hôte de gestion ZFSSA.
- `san_login` – Nom de l'utilisateur Cinder sur le ZFSSA.
- `san_password` – Mot de passe de l'utilisateur Cinder sur le ZFSSA.
- `zfssa_pool` – Le pool à utiliser pour allouer les volumes.
- `zfssa_target_portal` – Portail cible iSCSI ZFSSA (au format `data-IP:port`). Le port par défaut est 3260.
- `zfssa_project` – Nom du projet ZFSSA. Si le projet n'existe pas sur l'appareil, un projet avec le même nom est créé par le pilote au démarrage. Ce projet contient tous les volumes créés par le pilote. Des propriétés ZFSSA supplémentaires sont utilisées pour définir les caractéristiques des volumes (la taille de bloc, par exemple) et leur accès (par exemple, initiateurs, cibles, sécurité).

- `zfssa_initiator_config` - Propriété répertoriant plusieurs initiateurs, ou plusieurs groupes d'initiateurs. Cette propriété remplace le paramètre `zfssa_initiator_group` précédent, qui est obsolète dans la version Kilo d'OpenStack.

Pour répertorier plusieurs initiateurs, utilisez le format suivant :

```
zfssa_initiator_config = {
  'init-grp1': [
    {'iqn':'iqn1' , 'user':'user' , 'password':'password'},
    {'iqn':'iqn2' , 'user':'user' , 'password':'password'}
  ],
  'init-grp2': [
    {'iqn':'iqn3' , 'user':'user' , 'password':'password'}
  ] }
```

Pour obtenir un exemple spécifique de la procédure permettant de répertorier les initiateurs sur cette propriété, reportez-vous à l'[Exemple 6, "Utilisation de la propriété de pilote `zfssa_initiator_config`"](#).

- `zfssa_target_interfaces` – Les interfaces de réseau des cibles iSCSI ZFSSA. Exécutez la commande suivante pour afficher les interfaces :

```
zfssa:configuration net interfaces> show
Interfaces:

INTERFACE STATE CLASS LINKS   ADDR5      LABEL
e1000g0  up   ip    e1000g0  1.10.20.30/24  Untitled Interface
```

- `connection` – Définissez ce paramètre comme suit :

```
connection=mysql://cinder:service-password@controller-fqdn/cinder
```

3. Vérifiez que le service ZFSSA iSCSI est en ligne.

Si le service ZFSSA iSCSI n'est pas en ligne, utiliser la BUI ou la CLI dans l'appareil pour l'activer. L'exemple suivant montre comment utiliser la CLI dans l'appareil :

```
zfssa:> configuration services iscsi
zfssa:configuration services iscsi> enable
zfssa:configuration services iscsi> show
Properties:
<status> = online
...
```

4. Activez les services SMF de volume Cinder.

```
controller# svcadm enable cinder-volume:default cinder-volume:setup
```

Exemple 6 Utilisation de la propriété de pilote `zfssa_initiator_config`

Cet exemple montre comment répertorier plusieurs initiateurs pour la propriété `zfssa_initiator_config` dans le fichier de configuration Cinder.

Dans l'exemple, deux groupes d'initiateurs, Group A et Group B, sont créés sur l'appareil ZFS Storage Appliance. Répertoriez ces initiateurs dans le fichier `/etc/cinder/cinder.conf` comme suit :

```
zfssa_initiator_config = {
    'GroupA':[
        {'iqn':'iqn.1986-03.com.sun:01:0a43b9fdcf5.570d7fd1', 'user':'test1',
        'password':'password1234'},
        {'iqn':'iqn.1986-03.com.sun:01:0a43b9fdcf5.570d7fd2', 'user':'', 'password':''}
    ],
    'GroupB':[
        {'iqn':'iqn.1986-03.com.sun:01:0a43b9fdcf5.570d7fd3', 'user':'', 'password':''}
    ]
}
```


Options pour le déploiement de Neutron

Dans le modèle de configuration OpenStack à trois noeuds qui est utilisé pour cette documentation, le composant Neutron est installé conjointement avec le noeud de contrôleur dans un système unique (voir la [Figure 1, "Architecture de référence de la configuration à trois noeuds"](#)). Ce chapitre décrit la procédure d'installation du composant Neutron à l'intérieur d'une zone de noyau, indépendamment des autres composants principaux du système. Les thèmes ci-dessous sont abordés :

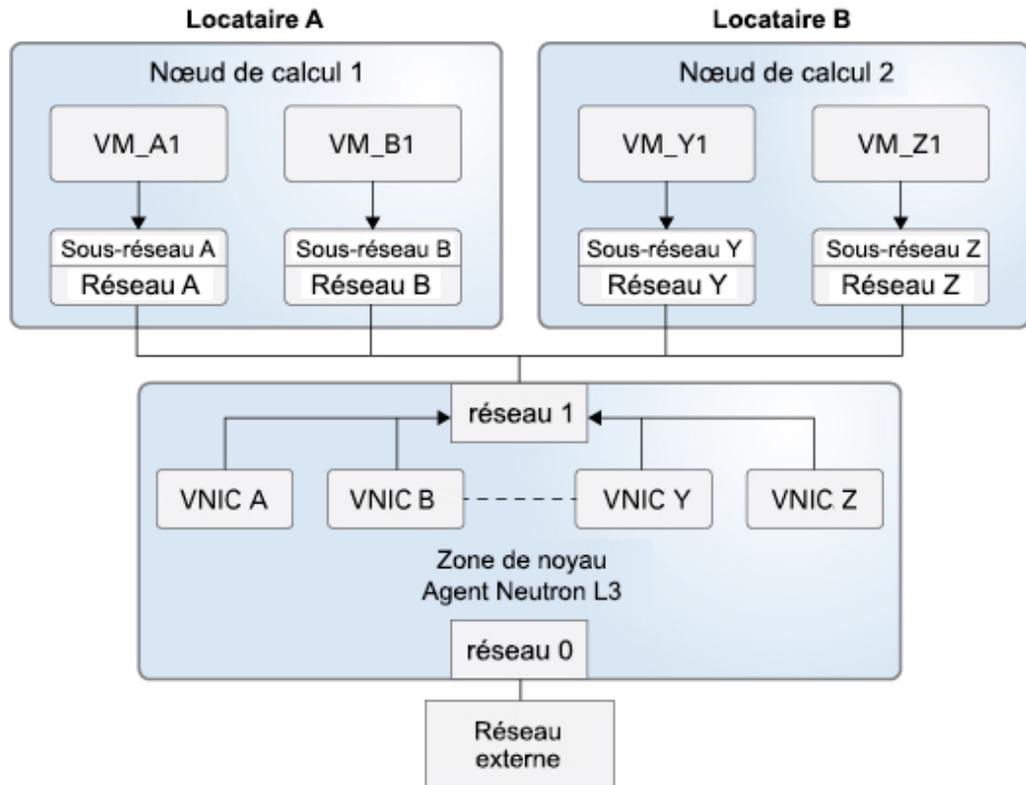
- ["Déploiement de Neutron dans une zone de noyau" à la page 107](#)
- ["Affichage des informations d'adresse MAC et de VID" à la page 110](#)

Déploiement de Neutron dans une zone de noyau

Dans les versions précédentes d'Oracle Solaris, l'impossibilité d'affecter des adresses de façon dynamique empêchait l'installation de Neutron dans des zones de noyau. Les propriétés de ressources de zone récemment introduites ne sont pas soumises à cette limitation.

La figure suivante illustre le déploiement de Neutron dans une zone de noyau.

FIGURE 7 Déploiement de Neutron dans une zone de noyau



Dans la figure, étant donné que des instances de machine virtuelle sont créées sur les noeuds de calcul, comme VM_A1, VM_B1, etc., l'agent L3 dans la zone de noyau configure les cartes VNIC correspondantes pour leurs réseaux respectifs. Avec des adresses dynamiques et des VID, l'agent peut automatiquement gérer les réseaux des instances de machine virtuelle lors de leur lancement par l'administrateur de cloud.

La prise en charge d'adresses MAC dynamiques et de VID est activée en définissant deux propriétés de ressources de zone :

- allowed-mac-address, qui a été ajoutée à la ressource anet`mac.
- allowed-vlan-ids, qui a été ajoutée à la ressource anet`vlan.

Remarque - Ces propriétés sont disponibles uniquement dans la marque solaris-kz.

▼ Installation du composant Neutron dans une zone de noyau

Utilisez cette procédure si vous voulez que le composant Neutron soit placé dans une zone de noyau isolée plutôt qu'il partage la zone globale avec d'autres composants OpenStack.

Les étapes de cette procédure se concentrent uniquement sur les configurations Neutron. Pour obtenir des instructions plus détaillées sur la configuration de zones de noyau, consultez la documentation relative aux zones appropriée.

1. Exécutez la procédure de création d'une zone de noyau.

Pour consulter les instructions complètes, reportez-vous à la section relative à la configuration d'une zone de noyau dans le manuel *Création et utilisation des zones de noyau d'Oracle Solaris*. Ce manuel figure dans la bibliothèque correspondant à votre version d'Oracle Solaris dans la page [Operating Systems Documentation](#).

2. Dans la zone de noyau, affectez une liste de préfixes d'adresse MAC que l'agent L3 peut affecter dynamiquement à des cartes VNIC.

La longueur du préfixe doit être comprise entre 1 et 5 octets, par exemple fa:16:3f ou fa:80:20:21:22.

Exécutez une nouvelle commande add pour chaque préfixe. Par exemple :

```
# zonecfg -z kernel-zone
zonecfg:kernel-zone> add anet
zonecfg:kernel-zone:anet> add mac
zonecfg:kernel-zone:anet:mac> add allowed-mac-address prefix
zonecfg:kernel-zone:anet:mac> add allowed-mac-address prefix
...
zonecfg:kernel-zone:anet:mac> end
zonecfg:kernel-zone:anet> end
zonecfg:kernel-zone>
```

3. Dans la zone de noyau, définissez une plage d'ID de VLAN que l'agent L3 peut affecter dynamiquement à des cartes VNIC.

Exécutez une nouvelle commande add pour chaque plage d'ID de VLAN autorisée. Par exemple :

```
# zonecfg -z kernel-zone
zonecfg:kernel-zone> add anet
zonecfg:kernel-zone:anet> add vlan
zonecfg:kernel-zone:anet:vlan> add allowed-vlan-ids id-range
zonecfg:kernel-zone:anet:vlan> add allowed-vlan-ids id-range
...
zonecfg:kernel-zone:anet:vlan> end
zonecfg:kernel-zone:anet> end
zonecfg:kernel-zone>
```

Au lieu de fournir une plage, vous pouvez également indiquer le mot-clé `any` pour la propriété `allowed-vlan-ids`. L'agent L3 affectera ensuite un ID de VLAN valide à une carte VNIC créée par l'agent.

4. Dans la zone de noyau, exécutez la procédure d'installation et de configuration de Neutron.

Pour obtenir des instructions, reportez-vous à la section "[Installation et configuration de Neutron](#)" à la page 38.

Affichage des informations d'adresse MAC et de VID

Les différentes commandes affichent diverses informations relatives à l'adresse MAC et au VID. La commande que vous utilisez varie également selon que vous vous trouvez sur la machine virtuelle invitée ou sur l'hôte.

Affichage à partir de la machine virtuelle invitée

A l'intérieur de l'instance de machine virtuelle, vous pouvez afficher la plage d'adresses MAC et de VID disponibles pour une utilisation par la machine virtuelle, à l'aide de la commande `dladm show-phys`. Pour afficher ces propriétés, vous devez utiliser l'option `-o` avec les colonnes que vous souhaitez voir sur la sortie. Les noms de colonne `ALLOWED-ADDRESSES` et `ALLOWED-VIDS` affichent les plages d'adresses MAC et de VID. Par exemple :

```
VM-instance# dladm show-phys -o link,media,device,allowed-addresses,allowed-vids
LINK  MEDIA  DEVICE  ALLOWED-ADDRESSES  ALLOWED-VIDS
net0  Ethernet  zvnet0  fa:16:3f,          100-199,
      fa:80:20:21:22  400-498,500
```

Affichage à partir de l'hôte

Lorsque vous êtes en dehors de l'instance de machine virtuelle, vous pouvez afficher la plage d'adresses MAC et d'ID de VLAN à l'aide de la commande `zonecfg info` ou `zonecfg export`. Vous pouvez éventuellement utiliser l'option `-r` avec chacune des commandes.

Les exemples suivants présentent des extraits de la sortie plus complète susceptible d'être générée par les commandes.

- `zonecfg info` ou `zonecfg -r info`
- `global-zone# zonecfg -z kernel-zone -r info`

```

anet:
...
  mac:
...
    allowed-mac-address: fa:16:3f
    allowed-mac-address: fa:80:20:21:22
...
  vlan:
...
    allowed-vlan-ids: 100-199
    allowed-vlan-ids: 400-498
    allowed-vlan-ids: 500
...

```

- `zonecfg export` ou `zone -r export`

```

global-zone# zonecfg -z kernel-zone -r export
add anet
...
add mac
add allowed-mac-address: fa:16:3f
add allowed-mac-address: fa:80:20:21:22
...
end
add vlan:
add allowed-vlan-ids: 100-199
add allowed-vlan-ids: 400-498
add allowed-vlan-ids: 500
end

```

Les commandes de zone présentent la plage d'adresses MAC ou de VID disponibles.

Pour voir quelles adresses et quels VID sont réellement utilisés, exécutez la commande `dladm show-vnic -m`. Dans l'exemple suivant, les informations d'adresse et d'ID réellement utilisées se fondent sur l'exemple de sortie précédent généré par les commandes `zonecfg`.

```

global-zone# dladm show-vnic -m
LINK          OVER   SPEED  MACADDRESSES  MACADDRTYPES  IDS
kz1/net0      net0   1000   2:8:20:31:ab:46  random        VID:0,100-109
              2:8:20:ad:29:e8  random
              fa:80:20:21:22:00 random
              fa:80:20:21:22:ff random
              fa:16:3f:0:0:1   random
              fa:16:3f:0:0:2   random

```

La sortie indique que 4 cartes VNIC ont été créées par l'agent. Deux cartes VNIC utilisent les adresses de la plage `fa:80:20:21:22`, tandis que les deux autres utilisent les adresses de la plage `fa:16:3f`. Vous pouvez également déduire de la sortie qu'il existe actuellement 4 machines virtuelles invitées dans les noeuds de calcul. Les machines virtuelles utilisent dix VID en tout.

Pour plus de détails sur les commandes `zonecfg` et `dladm`, reportez-vous aux pages de manuel [zonecfg\(8\)](#) et [dladm\(8\)](#).

Utilisation d'Ironic

Ce chapitre décrit le composant Ironic tel qu'implémenté et pris en charge dans Oracle Solaris. Les thèmes ci-dessous sont abordés :

- ["A propos du composant Ironic" à la page 113](#)
- ["Installation et configuration d'Ironic" à la page 114](#)
- ["Présentation : Déploiement sans système d'exploitation avec Ironic" à la page 119](#)
- ["Utilisation d'Ironic pour un déploiement sans système d'exploitation" à la page 121](#)

A propos du composant Ironic

Les chapitres précédents ont décrit les composants OpenStack principaux pour créer un cloud. Des composants supplémentaires offrent d'autres services qui sont associés à l'administration du cloud. Ce chapitre traite du composant Ironic disponible dans la version Kilo.

Alors que les composants principaux d'OpenStack permettent de provisionner les machines virtuelles ou les instances de machine virtuelle, Ironic offre des services pour l'inscription, le provisionnement et la mise hors service des instances ou noeuds sans système d'exploitation. Ironic utilise des technologies courantes, telles que l'initialisation PXE et IPMI, pour prendre en charge une grande variété de matériels que vous pouvez provisionner. En outre, un mécanisme de pilote enfichable permet à Ironic de gérer et de prendre en charge le matériel propre au fournisseur.

Pour plus d'informations sur Ironic et ses avantages, reportez-vous à la [developper documentation](#) d'Ironic sur le site Web de la communauté OpenStack.

Ironic comprend trois composants principaux. Dans Oracle Solaris, ces composants sont livrés sous forme de services SMF. Le tableau suivant répertorie et décrit ces composants.

Composant	Description	Service SMF
Service d'API Ironic OpenStack	Service fournissant une API RESTful qui permet aux opérateurs et autres services d'interagir avec les noeuds sans système d'exploitation gérés.	svc:/application/openstack/ironic/ironic-api

Composant	Description	Service SMF
Service de conducteur Ironic OpenStack	Contrôleur principal effectuant le provisionnement réel des noeuds sans système d'exploitation en utilisant la référence et les pilotes propres au fournisseur. Les services de conducteur et d'API communiquent en utilisant RPC.	svc:/application/openstack/ironic/ironic-conductor
Service de base de données Ironic OpenStack	Service SMF non persistant utilisé pour créer et synchroniser la base de données back-end d'Ironic.	svc:/application/openstack/ironic/ironic-db

Installation et configuration d'Ironic

Vous pouvez utiliser Ironic comme un composant autonome sans autres services OpenStack. Ou, vous pouvez le déployer avec d'autres composants OpenStack, généralement sur le noeud de calcul. La configuration dépend de la manière dont vous déployez Ironic.

Le provisionnement des instances sans système d'exploitation Oracle Solaris exige un programme d'installation automatisée (AI). Le programme d'installation automatisée peut être situé sur le même noeud que le service Ironic. Vous pouvez également utiliser le programme d'installation automatisée sur un serveur distant pour les opérations effectuées avec Ironic.

Pour plus d'informations sur la configuration et l'utilisation du programme d'installation automatisée, consultez le guide d'installation approprié dans la bibliothèque correspondant à votre version d'Oracle Solaris dans la [Operating Systems Documentation](#).

▼ Installation et configuration d'Ironic

Avant de commencer

Si vous déployez Ironic avec les autres composants OpenStack, assurez-vous que vous avez d'abord configuré ces composants principaux. Les configurations Keystone et Glance doivent au moins être terminées avant que vous ne puissiez travailler dans Ironic.

De même, assurez-vous que la base de données d'Ironic est configurée. Cette procédure suit l'architecture de référence à trois noeuds de la figure [Figure 1, "Architecture de référence de la configuration à trois noeuds"](#) où la base de données est configurée sur le noeud de contrôleur, avec les autres bases de données OpenStack.

1. **Créez la base de données Ironic.**
 - a. **Définissez la variable shell globale pour le noeud d'administrateur de contrôleur.**

```
controller# export CONTROLLER_ADMIN_NODE=controller-node
```

où *controller-node* peut être l'adresse IP ou le nom d'hôte du contrôleur. Pour plus d'informations sur la définition des variables, reportez-vous à la section "[Préparation des noms d'hôtes, variables et mots de passe](#)" à la page 26.

b. Créez la base de données avec les commandes suivantes :

```
controller# mysql -u root -p
Enter password: MySQL-root-password
mysql> create database ironic default character set utf8 default collate
utf8_general_ci;
mysql> grant all privileges on ironic.* to 'ironic'@'$CONTROLLER_ADMIN_NODE'
identified by 'service-password';
mysql> flush privileges;
mysql> quit
```

2. Installez le package Ironic OpenStack.

```
# pkg install ironic ironicclient rabbitmq
# pkg update stevedore stevedore-27
```

3. Installez le serveur AI et les outils d'administration.

Utilisez la même commande que vous installez le package du serveur AI localement avec Ironic ou à distance sur un hôte différent.

```
# pkg install pkg:/install/installadm
```

4. Si le programme d'installation automatisée est situé à distance, configurez l'utilisateur Ironic sur ce serveur.

```
remote-AI# useradd -d /var/lib/ironic -m -g 88 -u 88 \
-P "Install Service Management" ironic
remote-AI# passwd ironic
New Password: password
Re-enter new Password: password
```

Remarque - Si le programme d'installation automatisée est installé localement avec Ironic, le processus d'installation Ironic crée automatiquement l'utilisateur *ironic* sur le système.

5. Créez et administrez la clé SSH pour l'utilisateur Ironic.

- **Si le programme d'installation automatisée est installé localement avec Ironic, exécutez les commandes suivantes sur ce système :**

```
ironic-localhost# su - ironic
ironic-localhost# mkdir /var/lib/ironic/.ssh
ironic-localhost# ssh-keygen -N '' -t rsa \
-f /var/lib/ironic/.ssh/id_rsa
ironic-localhost# cat /var/lib/ironic/.ssh/id_rsa.pub > \
```

```
/var/lib/ironic/.ssh/authorized_keys
```

- **Si le programme d'installation automatisée est situé à distance, effectuez les étapes suivantes :**

- a. **Dans le programme d'installation automatisée distant, exécutez les commandes suivantes :**

```
remote-AI# su - ironic
remote-AI# mkdir /var/lib/ironic/.ssh
remote-AI# ssh-keygen -N '' -t rsa \
-f /var/lib/ironic/.ssh/id_rsa
remote-AI# cat /var/lib/ironic/.ssh/id_rsa.pub > \
/var/lib/ironic/.ssh/authorized_keys
```

- b. **Dans l'hôte Ironic, exécutez les commandes suivantes :**

```
ironic-localhost# mkdir /var/lib/ironic/.ssh
ironic-localhost# scp ironic@AI-server:~/.ssh/id_rsa /var/lib/ironic/.ssh
ironic-localhost# scp ironic@AI-server:~/.ssh/id_rsa.pub /var/lib/ironic/.ssh
ironic-localhost# cat /var/lib/ironic/.ssh/id_rsa.pub > \
/var/lib/ironic/.ssh/authorized_keys
ironic-localhost# chown -R ironic:ironic /var/lib/ironic/.ssh
```

où *AI-server* peut être l'adresse IP ou le nom d'hôte du serveur AI.

6. Modifiez le fichier `/etc/ironic/ironic.conf`.

Notez les points suivants concernant la configuration de certains paramètres dans le fichier :

<code>auth_strategy</code>	Sous la section <code>DEFAULT</code> , spécifiez <code>noauth</code> si vous utilisez Ironic en tant que composant autonome. Spécifiez <code>keystone</code> si vous déployez Ironic avec d'autres composants OpenStack.
<code>server</code>	Sous la section <code>ai</code> , spécifiez <code>localhost</code> si le programme d'installation automatisée est local avec Ironic. Si le programme d'installation automatisée est distant, spécifiez l'adresse IP ou le nom d'hôte de ce serveur.
<code>connection</code>	Sous la section <code>[database]</code> , spécifiez la connexion à la base de données MySQL.
<code>glance_host</code>	Sous la section <code>glance</code> , spécifiez l'adresse IP ou le nom d'hôte du serveur Glance si le paramètre <code>auth_strategy</code> est défini sur <code>keystone</code> .
<code>glance_api_servers</code>	De même, sous la section <code>glance</code> , spécifiez les noms d'hôte ou les adresses IP ainsi que les numéros de port correspondants, par exemple <code>192.168.0.150:9292</code> . Ne spécifiez pas <code>localhost</code> .

Utilisez l'exemple de fichier suivant comme guide pour configurer tous les paramètres requis pour le fonctionnement d'Ironic. Notez que dans l'exemple de fichier, la variable *glance-serverIP* représente l'adresse IP ou le nom d'hôte du serveur Glance.

```
[DEFAULT]
enabled_drivers=solaris
auth_strategy= le paramètre varie selon qu'Ironic est autonome ou ne l'est pas
pybasedir = /usr/lib/python2.6/vendor-packages/ironic
bindir = /usr/lib/ironic
host = ironic

[ai]
server= le paramètre varie selon que le programme d'installation automatisée est local ou distant
username=ironic
port=22
timeout=10
ssh_key_file=/var/lib/ironic/.ssh/id_rsa
deploy_interval=30

[api]
port=6385

[conductor]
api_url=http://localhost:6385/
heartbeat_timeout=60
heartbeat_interval=60
sync_power_state_interval=300
check_provision_state_interval=120

[database]
connection= mysql://ironic:service-password@$CONTROLLER_ADMIN_NODE/ironic

[solaris_ipmi]
imagecache_dirname = /var/lib/ironic/images
imagecache_lock_timeout = 60

[glance]
glance_host = glance-serverIP
glance_port = 9292
glance_protocol = http
glance_api_servers = glance-serverIP:port
auth_strategy = pour qu'Ironic utilise Keystone pour interagir avec Glance, spécifiez keystone

[keystone_authtoken] à configurer si sous DEFAULT, auth_strategy = keystone
auth_host = localhost
auth_uri = http://$CONTROLLER_ADMIN_NODE:5000/v2.0
identity_uri = http://$CONTROLLER_ADMIN_NODE:35357
admin_user = ironic
admin_password = service-password
admin_tenant_name = tenant
signing_dir = $state_path/keystone-signing

[neutron]
```

`auth_strategy =` *pour qu'Ironic utilise Keystone pour interagir avec Neutron, spécifiez keystone*

```
[oslo_messaging_rabbit]
rabbit_host=${CONTROLLER_ADMIN_NODE}
```

7. Assurez-vous que les services SMF pour Ironic sont en cours d'exécution.

```
ironic-localhost# svcs -a | grep rabbitmq
ironic-localhost# svcs -a | grep ironic
```

Si Ironic est en mode autonome, il se peut que vous ayez besoin d'activer les services manuellement.

```
ironic-localhost# svcadm enable rabbitmq
ironic-localhost# svcadm enable ironic-db
ironic-localhost# svcadm enable ironic-api ironic-conductor
```

8. (Facultatif) Testez l'utilitaire de ligne de commande.

a. Définissez les variables shell globales pour Ironic.

- **Si le paramètre `auth_strategy` est défini sur `noauth`, définissez les variables shell comme suit :**

```
ironic-localhost# export OS_AUTH_TOKEN=fake-token
ironic-localhost# export IRONIC_URL=http://localhost:6385
```

- **Si le paramètre `auth_strategy` est défini sur `keystone`, définissez les variables shell comme suit :**

Si le service Keystone est dans un noeud différent d'Ironic, spécifiez alors le nom d'hôte ou l'adresse IP du noeud dans lequel Keystone est installé, pour `OS_AUTH_URL`.

```
ironic-localhost# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0
ironic-localhost# export OS_PROJECT_NAME=service
ironic-localhost# export OS_USERNAME=ironic
ironic-localhost# export OS_PASSWORD=service-password
ironic-localhost# export IRONIC_URL=http://localhost:6385/
```

b. Exécutez une ligne de commande Ironic.

```
# ironic driver-list
+-----+-----+
| Supported driver(s) | Active host(s) |
+-----+-----+
| solaris             | ironic         |
+-----+-----+
```

Actuellement, `solaris` est le seul pilote activé et testé dans Ironic. Toutefois, vous pouvez ajouter des pilotes à la liste en incluant leur nom dans la section `[enabled_drivers]` du

fichier `/etc/ironic/ironic.conf`. Après avoir ajouté les pilotes, vous devez redémarrer les services SMF pour `ironic-api` et `ironic-conductor`.

Présentation : Déploiement sans système d'exploitation avec Ironic

Avec le pilote `solaris` Ironic, vous pouvez provisionner un noeud sans système d'exploitation en utilisant la fonction Unified Archive (UAR) d'Oracle Solaris ou son Image Packaging System (IPS). Lorsque vous créez un noeud, vous transmettez les informations au pilote `solaris` au moyen d'éléments de noeud configurables.

Le tableau suivant répertorie les éléments utilisés pour créer des noeuds. L'élément `driver_info/archive_uri` s'applique aux noeuds de provisionnement avec un fichier UAR. Les éléments restants du tableau s'appliquent aux noeuds de provisionnement en utilisant la fonction IPS.

Élément	Description	Exemple
<code>driver_info/archive_uri</code>	URI de l'archive Unified Archive avec laquelle la cible sans système d'exploitation est provisionnée.	<code>http://host.example.com/sol-11_3-x86.uar</code>
<code>driver_info/ai_service</code>	Nom du service AI à utiliser	<code>default-x86</code>
<code>driver_info/publishers</code>	Liste des éditeurs IPS, séparés par le signe Plus (+) et utilisant le format de dénomination <code>publisher-name@origin</code> .	<code>solaris@http://ipkg.us.oracle.com/solaris/N/dev+userland@http://my.example.repo</code>
<code>driver_info/fmri</code>	Liste des FMRI du package à installer, séparés par le signe Plus (+).	<code>pkg:/group/system/solaris-small-server+pkg:/cloud/openstack/nova</code>
<code>driver_info/install_profiles</code>	Liste des URI des profils de configuration à appliquer à l'environnement d'installation. Les URI sont séparés par le signe Plus (+).	<code>http://host.example.com/profile1.xml+glance://glance-image</code>
<code>driver_info/sc_profiles</code>	Liste des URI des profils de configuration système à appliquer au système installé. Les URI sont séparés par le signe Plus (+).	<code>http://host.example.com/profile1.xml+glance://glance-image</code>
<code>driver_info/manifest</code>	URI du manifeste AI à utiliser pour provisionner la cible sans système d'exploitation.	<code>http://host.example.com/my-manifest.xml</code>
<code>driver_info/ipmi_address</code>	Adresse IP ou nom d'hôte de la console série à connecter à ILOM sur le noeud à provisionner.	<code>192.168.2.200</code>
<code>driver_info/ipmi_username</code>	Nom d'utilisateur pour la connexion IPMI.	<code>root</code>

Elément	Description	Exemple
driver_info/ ipmi_password	Mot de passe pour la connexion IPMI.	<i>password</i>

Si vous provisionnez un noeud en utilisant un fichier UAR, vous avez seulement besoin de fournir des informations pour `driver_info/archive_uri`. L'URI de l'archive peut utiliser un des schémas indiqués dans la liste suivante. Les mêmes options de schéma peuvent être utilisées pour spécifier les profils si vous décidez de provisionner le noeud en utilisant IPS.

- `file://`
- `http://`
- `https://`
- `glance://`

Remarque - Bien que Glance soit généralement utilisé pour stocker les images d'installation, l'ajout de profils au stockage est acceptable.

Si vous provisionnez un noeud en utilisant IPS, assurez-vous en premier lieu qu'au moins un service AI par défaut existe pour l'architecture du noeud, comme illustré dans l'exemple suivant :

```
# installadm list
Service Name          Status Arch  Type Alias Aliases Clients Profiles Manifests
-----
default-i386          on   i386  pkg  yes  0      0      0      1
default-sparc         on   i386  pkg  yes  0      0      0      1
ironic-x86             on   i386  pkg  no   0      0      0      1
ironic-sparc          on   i386  pkg  no   0      0      0      1

# installadm list -vcn ironic-x86
There are no clients configured for service 'ironic-x86'.

# installadm list -vmn ironic-x86
Service Name          Manifest Name Status  Criteria
-----
ironic-x86            orig_default  default none
ironic-sparc          orig_default  default none
```

Lorsque vous utilisez IPS pour provisionner des noeuds, il est facultatif de spécifier l'élément `driver_info/ai_service`. Si vous omettez le nom du service AI, le service AI par défaut pour l'architecture de ce noeud est utilisé.

Si vous spécifiez des packages personnalisés pour `driver_info/fmri`, vous devez également spécifier les éditeurs pour l'élément `driver_info/publishers`.

Utilisation d'Ironic pour un déploiement sans système d'exploitation

La tâche suivante présente les étapes de base, de la création du noeud jusqu'au déploiement réel.

Pour plus d'informations sur l'utilisation de la fonctionnalité d'archive Unified Archive d'Oracle Solaris, reportez-vous à *Using Unified Archives for System Recovery and Cloning in Oracle Solaris*. Ce manuel figure dans la [library](#) correspondant à votre version d'Oracle Solaris.

▼ Déploiement sans système d'exploitation à partir d'un fichier UAR

Pour un exemple concret utilisant les étapes suivantes, reportez-vous à l'[Exemple 7](#), "Déploiement d'un noeud en utilisant un fichier UAR".

Avant de commencer

Assurez-vous que le fichier UAR à utiliser existe déjà.

1. **Créez les variables suivantes pour pouvoir utiliser l'utilitaire de ligne de commande Ironic.**
 - IP : adresse IP utilisée pour se connecter à l'ILOM des noeuds.
 - USER : généralement, l'utilisateur est root.
 - PASS : le mot de passe root.
 - HOST_MAC : l'adresse MAC du système.
2. **Créez le noeud Ironic.**

```
# ironic node-create options
```

Les propriétés du nouveau noeud sont affichées, notamment l'UUID du noeud.
3. **Créez une variable pour l'UUID du noeud afin de faciliter la référence.**
4. **Mettez à jour le noeud en spécifiant l'emplacement du fichier UAR à installer.**

```
# ironic node-update options
```
5. **Créez un port associé pour ce noeud.**

```
# ironic port-create options
```
6. **(Facultatif) Validez les champs que vous avez spécifiés pour le noeud.**

```
# ironic node-validate options
```

7. Provisionnez le noeud.

```
# ironic node-set-provision-state options
```

8. (Facultatif) Affichez le statut du déploiement.

```
# ironic node-show options
```

Remarque - Le résultat est différent si vous exécutez la commande pendant le provisionnement plutôt qu'après la fin du processus.

Exemple 7 Déploiement d'un noeud en utilisant un fichier UAR

Cet exemple est basé sur les conditions suivantes :

- Informations de base sur l'ordinateur hébergeant le noeud
 - Nom d'hôte : mynewnode.example.com
 - Architecture : x86
 - Adresse IP : 1.1.1.1
 - Adresse MAC : 01:02:03:04:05:06
- Informations de base sur l'hôte ILOM
 - Nom d'hôte : mynewnode-aa.example.com
 - Adresse IP : 2.2.2.2
 - Utilisateur : root
 - Mot de passe : password
- Nom du fichier UAR : myuar.server/sol11-3-x86.uar

```
# export ILOM_IP=2.2.2.2
# export ILOM_USER=root
# export ILOM_PASS=password
# export HOST_MAC=01:02:03:04:05:06

# ironic node-create -d solaris -i ipmi_address=$ILOM_IP \
  -i ipmi_username=$ILOM_USER -i ipmi_password=$ILOM_PASS
```

Le noeud est créé

Property	Value
uuid	4eacbfde-4977-4d8c-8043-8cbe8f876187
driver_info	{u'ipmi_address': u'2.2.2.2', u'ipmi_username': u'root', u'ipmi_password': u'password'}
extra	{}
driver	solaris
chassis_uuid	None
properties	{}

```
# export NODE=4eacbfde-4977-4d8c-8043-8cbe8f876187    L'UUID est enregistré
```

```

# ironic node-update $NODE \
  add driver_info/archive_uri=http://myuar.server/sol11-3-x86.uar
+-----+
| Property          | Value                                     |
+-----+
| instance_uuid     | None                                     |
| target_power_state | None                                     |
| properties        | {}                                       |
| maintenance       | False                                    |
| driver_info       | {u'archive_uri': u'http://myuar.server/sol11-3-x86.uar', |
|                   | u'ipmi_address': u'2.2.2.2', u'ipmi_username': u'root', |
|                   | u'ipmi_password': u'password'}         |
| extra            | {}                                       |
| last_error        | None                                     |
| created_at        | 2014-10-03T15:38:43+00:00              |
| target_provision_state | None                                     |
| driver            | solaris                                  |
| updated_at        | 2014-10-03T15:53:04+00:00              |
| instance_info     |                                           |
| chassis_uuid      | None                                     |
| provision_state    | None                                     |
| reservation       | None                                     |
| power_state        | None                                     |
| console_enabled    | False                                    |
| uuid              | 4eacbfde-4977-4d8c-8043-8cbe8f876187   |
+-----+

# ironic port-create -n $NODE -a $HOST_MAC
+-----+
| Property | Value                                     |
+-----+
| node_uuid | 4eacbfde-4977-4d8c-8043-8cbe8f876187 |
| extra     | {}                                       |
| uuid      | 4c765ab0-2529-4463-a51b-e5544dd15a32 |
| address   | 01:02:03:04:05:06                       |
+-----+

# ironic node-validate $NODE
+-----+
| Interface | Result | Reason          |
+-----+
| console   | None   | not supported   |
| deploy    | True   |                 |
| management | True   |                 |
| power     | True   |                 |
+-----+

# ironic node-set-provision-state $NODE active   Le noeud est provisionné

# ironic node-show $NODE
+-----+
| Property          | Value                                     |
+-----+
| instance_uuid     | None                                     |

```

```

|target_power_state      | None
|properties              | {}
|maintenance            | False
|driver_info            | {u'archive_uri': u'http://myuar.server/sol11-3-x86.uar',
|                        | u'ipmi_address': u'2.2.2.2', u'ipmi_username': u'root',
|                        | u'ipmi_password': u'password'}
|extra                  | {}
|last_error             | None
|created_at             | 2014-10-03T15:38:43+00:00
|target_provision_state | deploy_complete
|driver                 | solaris
|updated_at             | 2014-10-03T15:53:04+00:00
|instance_info          |
|chassis_uuid           | None
|provision_state        | active
|reservation            | None
|power_state            | power on
|console_enabled        | False
|uuid                   | 4eacbfde-4977-4d8c-8043-8cbe8f876187
+-----+-----+

```

Si vous avez exécuté la commande `ironic node-show` pendant le provisionnement, la valeur `provision_state` indique un statut différent de `active`.

▼ Mise hors service d'un noeud

Utilisez également cette procédure en cas d'échec d'une opération de provisionnement.

1. Définissez le noeud sur un état supprimé.

```
# ironic node-set-provision-state $NODE deleted
```

2. Affichez les informations du noeud.

Dans l'exemple suivant, les propriétés ayant un astérisque (*) indiquent l'état supprimé du noeud.

```

# ironic node-show $NODE
+-----+-----+
| Property          | Value
+-----+-----+
|instance_uuid      | None
|target_power_state | None
|properties          | {}
|maintenance        | True
|driver_info        | {u'archive_uri': u'http://myuar.server/sol11-3-x86.uar',
|                    | u'ipmi_address': u'2.2.2.2', u'ipmi_username': u'root',
|                    | u'ipmi_password': u'password'}
|extra              | {}
|last_error         | None
+-----+-----+

```

*

created_at	2014-10-03T15:38:43+00:00		
target_provision_state	None		*
driver	solaris		
updated_at	2014-10-03T15:53:04+00:00		
instance_info	{}		
chassis_uuid	None		*
provision_state	None		*
reservation	None		
power_state	power off		*
console_enabled	False		
uuid	4eacbfde-4977-4d8c-8043-8cbe8f876187		
+-----+			

3. Si vous essayez de résoudre l'échec du processus de provisionnement, validez les champs du noeud.

Les problèmes liés à n'importe quel élément driver_info spécifié pour le noeud sont identifiés dans la colonne Reason.

```
# ironic node-validate $NODE
```

Interface	Result	Reason	
console	None	not supported	
deploy	True		
management	True		
power	True		

Utilisation de Heat

Ce chapitre décrit le composant Heat tel qu'implémenté et pris en charge dans Oracle Solaris. Les thèmes ci-dessous sont abordés :

- ["A propos du composant Heat" à la page 127](#)
- ["Installer Heat" à la page 128](#)
- ["A propos des modèles HOT" à la page 128](#)
- ["Utilisation de Heat avec Cloudbase-Init" à la page 130](#)

A propos du composant Heat

Heat est le moteur d'orchestration d'OpenStack qui permet de déployer des applications pour le cloud basées sur les modèles Heat Orchestration que vous créez. Ces modèles sont également appelés modèles HOT.

Un modèle HOT vous permet de créer différents types de ressource OpenStack, notamment des instances, des IP flottantes, des volumes et des utilisateurs. Les modèles vous permettent également de déployer des fonctionnalités avancées telles que la haute disponibilité des instances, la mise à l'échelle automatique des instances et les piles imbriquées. Ainsi, Heat permettrait à tous les principaux projets OpenStack de recevoir une base élargie d'utilisateurs. A l'aide des modèles, Heat vous permet de remplacer l'implémentation par défaut des ressources en utilisant à la place les paramètres qui sont transmis du modèle au moteur.

Heat fournit ses services via une API de services Web RESTful. Comme pour toutes les applications OpenStack, ce sont les interfaces WSGI (Python) qui sont utilisées et les applications sont configurées ensemble à l'aide de Paste. Les points d'extrémité HTTP de l'application sont constitués des pipelines du middleware Web Server Gateway Interface (WSGI). Heat utilise deux points d'extrémité en particulier : le port 8004 pour l'API Heat et le port 8000 pour l'outil Heat Cloud Formation.

La configuration de Heat est contrôlée par le fichier de configuration `/etc/heat/heat.conf`. A ce stade, aucun paramètre de configuration Solaris n'existe dans le principal fichier de configuration Heat.

Pour plus d'informations sur le composant Heat, reportez-vous à la [Heat documentation](#) de la communauté OpenStack.

Installer Heat

Dans une configuration standard, vous installez les services Heat sur le même noeud que Keystone. Le package Heat est automatiquement inclus si vous avez installé OpenStack sur le noeud avec la commande suivante :

```
# pkg install openstack
```

▼ Configuration de Heat

Avant de commencer

Vous devez d'abord configurer Keystone comme décrit à la section "[Installation et configuration de Keystone](#)" avant d'effectuer cette tâche.

1. **Configurez Heat en éliminant des marques de commentaire ou en définissant des paramètres dans `/etc/heat/heat.conf` :**

```
[database]
connection = mysql://heat:service-password@$CONTROLLER_ADMIN_NODE/heat

[keystone_authtoken]
auth_uri = http://$CONTROLLER_ADMIN_NODE:5000/v2.0
identity_uri = http://$CONTROLLER_ADMIN_NODE:35357
admin_user = heat
admin_password = service-password
admin_tenant_name = tenant

[oslo_messaging_rabbit]
rabbit_host=$CONTROLLER_ADMIN_NODE
```

2. **Activez le service Heat.**

```
controller# svcadm enable -rs heat-api heat-db heat-engine \
heat-api-cfn heat-api-cloudwatch
```

A propos des modèles HOT

Pour que le composant Heat organise les différentes applications cloud composites dans votre configuration d'OpenStack, vous devez définir un modèle HOT (Heat Orchestration Template).

Un modèle HOT contient des spécifications que vous devez suivre. Les paramètres que vous fournissez sont lus lors de l'exécution du processus pour créer des types de ressource et d'autres fonctionnalités avancées.

Pour les spécifications du modèle HOT et leurs descriptions, reportez-vous à la rubrique http://docs.openstack.org/developer/heat/template_guide/hot_spec.html#hot-spec.

Pour plus d'informations sur la création d'un modèle HOT, reportez-vous à la rubrique http://docs.openstack.org/developer/heat/template_guide/hot_guide.html#hot-guide.

Pour déployer un modèle HOT, utilisez la commande suivante :

```
# heat stack-create -f template stack-name
```

template Chemin d'accès complet du fichier de modèle que vous traitez.

stack-name Pile que vous voulez créer.

Vous pouvez également insérer des valeurs de paramètre supplémentaires dans la commande, au format `-P key1=value1;key2=value2...`

L'exemple suivant présente le contenu du modèle HOT `stack3` qui crée trois réseaux privés avec des sous-réseaux.

Remarque - La ligne `heat_template_version: 2013-05-23` doit apparaître en haut du modèle.

```
heat_template_version: 2013-05-23
description: Create a few networks.

resources:
  heat_net1:
    type:OS::Neutron::Net
    properties:
      name:heat_net1

  heat_subnet1:
    type:OS::Neutron::Subnet
    properties:
      name:heat_subnet1
      network_id: { get_resource: heat_net1 }
      cidr: 192.168.50.0/24

  heat_net2:
    type:OS::Neutron::Net
    properties:
      name: heat_net2

  heat_subnet2:
    type:OS::Neutron::Subnet
    properties:
```

```
name:heat_subnet2
network_id: { get_resource: heat_net2 }
cidr: 192.168.51.0/24

heat_net3:
type:OS::Neutron::Net
properties:
name: heat_net3

heat_subnet3:
type:OS::Neutron::Subnet
properties:
name:heat_subnet3
network_id: { get_resource: heat_net3 }
cidr: 192.168.52.0/24
```

Utilisation de Heat avec Cloudbase-Init

Cloudbase-Init est un service tiers qui initialise automatiquement de nouvelles images invitées dans le cloud, telles que la création d'utilisateurs, la génération de mots de passe, l'exécution de scripts, etc. Oracle Solaris OpenStack prend en charge ce service dans Kilo mais le limite uniquement au module d'extension UserData.

Le package Cloudbase-Init n'est inclus dans aucun package de groupe standard. Vous devez installer ce package uniquement sur des systèmes dont les images sont expressément destinées à un déploiement dans des environnements de cloud.

▼ Initialisation automatique d'une image invitée

Avant de commencer

Vous devez avoir effectué les opérations suivantes :

- Créé une archive Unified Archive (UA) du système dont vous voulez déployer l'image dans le cloud. Vérifiez que le package cloudbase-init est inclus dans ce système avant de créer l'UA.

Pour obtenir des instructions sur la création d'une UA, reportez-vous au chapitre *Utilisation des archives Unified Archives* dans le guide *Utilisation de Unified Archives pour la récupération du système et le clonage dans Oracle Solaris*. Ce guide figure dans la bibliothèque correspondant à votre version d'Oracle Solaris dans la [Operating Systems Documentation](#).

- Dans la section des données utilisateur d'un modèle Heat, indiquez les informations de données utilisateur. Enregistrez le modèle sous forme de fichier *.yaml.

1. Téléchargez l'image contenant le package Cloudbase-Init dans Glance.

2. Exécutez la commande suivante :

```
# heat stack-create -f yml-template \
  -P key_name=your key name;image=image name\
  private_net=name of tenant private network stack-name
```

Exemple 8 Modèle Heat utilisant Cloudbase-Init

L'exemple suivant présente un modèle Heat qui sera traité à l'aide de Cloudbase-Init. Les informations que vous devez fournir sont affichées en gras et les informations de données utilisateur sont marquées à des fins pratiques. Le nom du modèle est `test.yml`.

```
heat_template_version: 2013-05-23
```

```
description: HOT template to deploy one server into an existing neutron tenant
network and assign a floating IP address so it's routable from the public network.
```

```
parameters:
```

```
  key_name:Server1Key
```

```
    type: string
```

```
    description: Name of keypair to assign to server
```

```
  image:Solaris Non-global Zone
```

```
    type: string
```

```
    description: Name of image to use for server
```

```
  flavor:8
```

```
    type: string
```

```
    description: Flavor to use for server
```

```
    default: 1
```

```
  public_net:
```

```
    type: string
```

```
    description:
```

```
      Name or ID of public network for which floating IP address will
be allocated
```

```
    default: oracle
```

```
  private_net:HR
```

```
    type: string
```

```
    description: Name or ID of private network into which server is
deployed
```

```
resources:
```

```
  server1:
```

```
    type: OS::Nova::Server
```

```
    properties:
```

```
      name: { get_param: 'OS::stack_name' }
```

```
      image: { get_param: image }
```

```
      flavor: { get_param: flavor }
```

```
      key_name: { get_param: key_name }
```

```
      networks:
```

```
        - port: { get_resource: server1_port }
```

```
      user_data_format: RAW
```

```
----- Beginning of user data section -----
```

```
      user_data:
```

```
        str_replace:
```

```
    template: |
        #!/bin/ksh
        print "This is a test."

server1_port:
  type: OS::Neutron::Port
  properties:
    network: { get_param: private_net }

server1_floating_ip:
  type: OS::Neutron::FloatingIP
  properties:
    floating_network: { get_param: public_net }
    port_id: { get_resource: server1_port }

outputs:
  server1_private_ip:
    description: IP address of server1 in private network
    value: { get_attr: [ server1, first_address ] }
  server1_public_ip:
    description: Floating IP address of server1 in public network
    value: { get_attr: [ server1_floating_ip, floating_ip_address ] }
```

Pour déployer le modèle, saisissez les informations suivantes :

```
# heat stack-create -f test.yaml -P key_name=Server1Key \
  -P image=Solaris Non-global Zone\
...-P flavor=8
  -P private_net=HR teststack
```

Notez que les valeurs spécifiques de chaque option de la commande proviennent des informations figurant dans le fichier de modèle `test.yaml`. Le réseau public est `oracle`, qui représente la valeur par défaut.

Dépannage d'OpenStack

Ce chapitre décrit les problèmes OpenStack associés à la version en cours. Il fournit également des conseils de dépannage et des solutions pour résoudre les problèmes de base que vous pourriez rencontrer lors de la configuration. Il aborde les sujets suivants :

- ["Obtention de l'aide sur la ligne de commande" à la page 133](#)
- ["Limitations connues" à la page 134](#)
- ["Examen des fichiers journaux" à la page 135](#)
- ["Examen et résolution des problèmes" à la page 138](#)
- ["Conseils et astuces généraux sur le débogage" à la page 144](#)
- ["Sites utiles" à la page 145](#)

Obtention de l'aide sur la ligne de commande

Dans OpenStack, les commandes correspondent aux composants OpenStack. Par exemple, la commande `nova` s'applique aux opérations de calcul, la commande `cinder` au stockage et la commande `neutron` à la mise en réseau.

Pour obtenir de l'aide sur l'utilisation de ces commandes, notamment la syntaxe correcte, les sous-commandes prises en charge, les options possibles, etc., utilisez la commande `command-component help`, telle que `nova help` ou `neutron help`. Vous pouvez filtrer la liste des sous-commandes possibles à utiliser avec la commande `command-component` à l'aide de la commande `grep`. Par exemple, pour répertorier les sous-commandes `neutron` liées aux routeurs, saisissez la commande suivante :

```
# neutron help | grep router
l3-agent-list-hosting-router List L3 agents hosting a router.
l3-agent-router-add        Add a router to a L3 agent.
l3-agent-router-remove     Remove a router from a L3 agent.
net-gateway-connect       Add an internal network interface to a router.
router-create             Create a router for a given tenant.
router-delete             Delete a given router.
router-gateway-clear      Remove an external network gateway from a router.
router-gateway-set        Set the external network gateway for a router.
router-interface-add      Add an internal network interface to a router.
router-interface-delete   Remove an internal network interface from a router.
```

<code>router-list</code>	List routers that belong to a given tenant.
<code>router-list-on-l3-agent</code>	List the routers on a L3 agent.
<code>router-port-list</code>	List ports that belong to a given tenant, with specified
<code>router.</code>	
<code>router-show</code>	Show information of a given router.
<code>router-update</code>	Update router's information.

Puis, pour obtenir des détails spécifiques sur une sous-commande, telle que `router-list` qui identifie les routeurs dans le cloud, saisissez la commande suivante :

```
# neutron help router-list
```

En raison de l'implémentation d'OpenStackClient (OSC), certaines commandes basées sur les composants sont obsolètes. Utilisez à la place `openstack` comme commande principale avec la sous-commande appropriée. Pour obtenir une brève description d'OSC, reportez-vous à la section "[Implémentation d'OpenStackClient](#)" à la page 16.

Pour obtenir plus d'informations sur la commande `openstack` et ses sous-commandes, utilisez l'une des commandes suivantes :

- `openstack help subcommand`
- `openstack --help`

Si vous entrez `openstack` sans sous-commandes, vous passez en mode interactif, ce qui vous permet de saisir `help [subcommand]` pour obtenir des informations. Pour quitter le mode interactif, saisissez `quit`.

Pour plus d'informations sur OSC, reportez-vous à la page <http://docs.openstack.org/developer/python-openstackclient/index.html>.

Pour obtenir la liste des commandes précédentes et de leurs équivalents dans OSC, reportez-vous à l'[Annexe B, Commandes OpenStackClient](#).

Limitations connues

Problèmes connus avec OpenStack (Kilo) dans Oracle Solaris :

- Seuls les noeuds Nova exécutant Oracle Solaris sont entièrement pris en charge, car Neutron ne prend en charge qu'un seul module d'extension pour la virtualisation du réseau.
- L'ajout de volumes Cinder en pièce jointe n'est actuellement pas pris en charge dans les zones non globales.
- Les instances de machine virtuelle doivent exécuter la version actuelle d'Oracle Solaris.
- La sauvegarde de Cinder n'est pas prise en charge.

Le service `cinder-backup` est installé en même temps que le package `cinder`. Toutefois, dans un déploiement Cinder par défaut tel que celui décrit à la section "[Configuration du](#)

noeud de stockage" à la page 46, le service ne fonctionne pas pour l'instant pour le volume backup_volume.

- Sur la boîte de dialogue du lancement de l'instance, seule l'initialisation à partir de l'image est prise en charge pour la source d'initialisation de l'instance. Dans le menu Projet -> Images & Clichés -> Actions, UploadToImage n'est pas pris en charge.
- Les liaisons de données VXLAN ne sont pas prises en charge en tant que valeur pour l'option external_network_dataLink dans le fichier /etc/neutron/l3_agent.ini. Si vous définissez une liaison de données VXLAN en tant que valeur de l'option external_network_dataLink, l'agent Neutron L3 ne parviendra pas à créer et à raccorder une VNIC sur le réseau externe.
- Vous devez utiliser la ligne de commande pour modifier le quota d'une ressource réseau pour un projet.

Le quota d'une ressource réseau ne peut pas être modifié dans Horizon. Vous pouvez utiliser le tableau de bord Horizon pour créer un projet ou modifier des ressources non-réseau d'un projet existant. Pour modifier le quota des réseaux, des sous-réseaux, des ports, des routeurs ou des adresses IP flottantes, vous devez utiliser la commande `neutron quota-update`.

Même lorsque vous modifiez une ressource non-réseau, vous pouvez voir le message d'erreur suivant. Vous pouvez ignorer ce message. Contrairement à ce qu'indique le message, le quota de la ressource non-réseau a été appliqué.

Error: Modified project information and members, but unable to modify project quotas.

- SMF et Openstack peuvent renvoyer des états de service différents.

L'exemple suivant montre que le service nova-cert est désactivé dans OpenStack, même si SMF présente ce service comme étant online :

```
root@c190-133:~# nova service-disable c190-133 nova-cert
+-----+-----+-----+
| Host    | Binary  | Status  |
+-----+-----+-----+
| c190-133 | nova-cert | disabled |
+-----+-----+-----+
root@c190-133:~# svcs nova-cert
STATE          STIME      FMRI
online         21:14:11  svc:/application/openstack/nova/nova-cert:default
```

Examen des fichiers journaux

Les services SMF ainsi que divers processus Oracle Solaris produisent des fichiers journaux que vous pouvez consulter pour trouver les messages d'erreur ou encore collecter plus d'informations sur les messages affichés à l'écran. Les fichiers journaux du service SMF contiennent d'utiles informations de débogage.

Comme OpenStack est en général installé sur plusieurs systèmes, les fichiers journaux que vous avez besoin de consulter se trouvent à différents emplacements. Pour effectuer un dépannage plus systématique, examinez les journaux noeud par noeud.

Pour savoir comment résoudre les problèmes liés aux services SMF, reportez-vous au manuel *Gestion des services système dans Oracle Solaris* dans la bibliothèque correspondant à votre version d'Oracle Solaris dans la [Operating Systems Documentation](#).

Pour afficher les journaux des services, assurez-vous d'avoir l'autorisation appropriée. Prenez le profil RBAC approprié afin de visualiser les fichiers journaux de service OpenStack ou d'utiliser la commande `pfedit` pour modifier les fichiers de configuration du service OpenStack. Les profils pouvant être affectés sont les suivants :

- Gestion OpenStack en mode "block storage"
- Gestion de calcul OpenStack
- Gestion des identités OpenStack
- Gestion des images OpenStack
- Gestion du réseau OpenStack
- Gestion de stockage d'objets OpenStack
- Gestion OpenStack

Les commandes générales utilisées pour le dépannage sont les suivantes :

- Pour savoir quels services OpenStack sont en cours d'exécution sur un noeud spécifique :

```
# svcs -a | grep openstack
```

- Pour afficher la liste des services qui pourraient être en mode de maintenance :

```
# svcs -x
svc:/application/openstack/swift/swift-replicator-rsync:
    default (OpenStack Swift Replication Service)
State: maintenance since Fri May 22 04:06:11 2015
Reason: Start method exited with $SMF_EXIT_ERR_FATAL.
    See: http://support.oracle.com/msg/SMF-8000-KS
    See: rsync(1)
    See: rsyncd.conf(5)
    See: /var/svc/log/application-openstack-swift-swift-replicator-rsync:default.log
Impact: This service is not running.
```

Si un service est en mode de maintenance, examinez son fichier journal.

- Pour identifier le journal correspondant à un service OpenStack spécifique :

```
# svcs -L openstack-service
```

Par exemple :

```
# svcs -L neutron-server
```

```
/var/svc/log/application-openstack-neutron-neutron-server:default.log
```

Si vous avez l'autorisation appropriée, vous pouvez combiner des options. Ainsi, `-Lv` permet à la fois d'identifier et d'afficher le journal du service.

- Pour identifier immédiatement des instances d'erreur enregistrées dans un journal spécifique, vous pouvez utiliser des commandes UNIX courantes telles que `grep`.

```
# grep keyword `svcs -L openstack-service`
```

Vous pouvez rechercher les occurrences de `error` (erreur), `warning` (avertissement) et d'autres mots-clés importants pour lire directement les messages d'erreur.

- Pour vérifier les propriétés EVS lors de la résolution de problèmes réseau, utilisez les diverses sous-commandes `evsadm`, par exemple `evsadm show-prop`.

Les journaux suivants contiennent souvent des informations utiles pour le dépannage :

- `nova-compute`
- `nova-scheduler`
- `cinder-scheduler`
- `neutron-server`

Outre les fichiers journaux des services SMF, vous pouvez consulter les journaux du répertoire `/var/log`. Comme d'autres processus Oracle Solaris, les services OpenStack génèrent leurs propres fichiers journaux dans les répertoires `/var/log/openstack-service`.

Par exemple, les fichiers journaux du magasin d'images OpenStack se trouvent dans `/var/log/glance`. Des problèmes relatifs à la création et au démarrage d'une instance de machine virtuelle peuvent être consignés dans le répertoire `/var/log/zones`. Les journaux de messages sont stockés sous `/var/log/rabbitmq/rabbit@hostname.log`.

La plupart des fichiers de configuration d'OpenStack se trouvent sous le nom de service OpenStack dans le répertoire `/etc`. Ainsi, les fichiers de configuration de mise en réseau OpenStack sont dans `/etc/neutron`. Les fichiers de configuration de Horizon sont dans `/etc/openstack_dashboard`. Ceux concernant Nova sont dans `/etc/nova`, etc. Vous pouvez utiliser ces fichiers pour dépanner un service spécifique en définissant les paramètres suivants ou en éliminant les marques de commentaire dans le fichier de configuration du service considéré :

- `debug=true`
- `verbose=true`

Ces paramètres permettent de voir davantage de résultats pour les opérations concernées par ce fichier de configuration. Reportez-vous aux tableaux d'options de configuration dans "Common Configuration Parameters for OpenStack" dans la rubrique <http://www.oracle.com/technetwork/articles/servers-storage-admin/getting-started-openstack-os11-2->

[2195380.html](#) et dans la *Référence de configuration d'OpenStack* sur le site de documentation [OpenStack](#).

Remarque - Certaines commandes de service OpenStack acceptent également l'option `--debug`. Cette option équivaut à définir `debug=true` dans un fichier de configuration.

Examen et résolution des problèmes

Cette section décrit quelques problèmes possibles lors de l'installation et de la configuration d'OpenStack.

L'exemple suivant montre une erreur en rapport avec le tableau de bord.

```
Error: Unauthorized: Unable to retrieve usage information.
Error: Unauthorized: Unable to retrieve quota information.
Error: Unauthorized: Unable to retrieve project list information.
Error: Unauthorized: Unable to retrieve instance list information.
```

Ces messages peuvent indiquer que la clé d'hôte RSA a été modifiée et n'a pas été propagée à tous les composants. Pour plus d'informations sur la configuration de clés RSA, reportez-vous à "[Installation et configuration de Neutron](#)" et à "[Configuration du noeud de calcul](#)".

Le rapport d'erreur suivant peut être inclus dans le journal `nova-scheduler`.

```
controllor# grep error `svcs -L nova-scheduler`
2014-12-03 12:49:19.271 3475 TRACE
nova.openstack.common.rpc.common error: [Errno 32] Broken pipe
```

Une erreur de pipe cassé est généralement consignée lorsque vous actualisez un service OpenStack et pas les autres. Si vous avez effectué des modifications dans les fichiers de configuration d'un noeud, actualisez tous les services sur ce noeud. La commande suivante redémarre des services qui sont en ligne mais ont besoin d'être actualisés.

```
controllor# svcs `*openstack*` | grep online \
| awk -e '{print $3}' | xargs svcadm restart
```

Une erreur peut aussi être due à un manque de ressources. Pendant la création d'une instance de machine virtuelle, le journal `nova-compute` peut afficher des messages de ce type :

```
[abc-123-def-456] Build of instance
abc-123-def-456 aborted: Image
xyz-987-uvw-654 is unacceptable: Image query failed.
Possibly invalid or corrupt. Log file location: controllor:/tmp/archive_log.4249
```

Par la suite, le journal va également indiquer `out of space/storage`. Utilisez la commande `top` pour voir les ressources disponibles sur le système. Si le système a moins de 1 Go de mémoire, il se peut que vous deviez en ajouter.

Création d'un réseau

Lors de la configuration d'un réseau interne et de ses composants, le message suivant peut s'afficher en permanence sur l'écran de la console :

```
To: root@controller.mydomain.com
From: neutron@controller.mydomain.com
Subject: *** SECURITY information for controller ***
Content-Length: 143
```

```
controller: datetime : neutron user NOT in sudoers; TTY = unknown ; PWD=var/lib/neutron;
user=ROOT ; COMMAND=command:
```

Pour éviter la génération de ce message, modifiez le fichier `/etc/neutron/neutron.conf` comme suit :

1. Supprimez la mise en commentaire de la ligne correspondant à l'option `root_helper`.
2. Vérifiez que le paramètre est défini sur `none`.

```
root_helper =
```

Installation et configuration de l'instance de machine virtuelle

Les problèmes abordés dans cette section sont spécifiques aux instances de machine virtuelle.

Instance de machine virtuelle en statut d'erreur

Une instance de machine virtuelle pourra être en statut d'erreur car vous tentez de l'installer sur un système hôte qui a une architecture différente. Dans ce cas, vous risquez de ne pas recevoir un message d'erreur vous indiquant précisément que les architectures ne correspondent pas. Pour éviter ce problème, assurez-vous d'avoir correctement défini la propriété `architecture` de l'image lorsque vous téléchargez celle-ci depuis le magasin d'images Glance. Si vous utilisez Horizon pour télécharger une image, vous devez définir les propriétés de l'image après les téléchargements. Vous pouvez également utiliser la ligne de commande pour télécharger l'image et définir les valeurs de propriété en une seule commande `glance image-create`. Pour un exemple, voir "[Préparation des images pour le référentiel Glance](#)" à la page 59.

Les valeurs de propriété d'instance de machine virtuelle ne correspondent pas aux valeurs de propriété de la zone

Certaines informations qu'OpenStack retourne sur une instance de machine virtuelle ne correspondent pas aux informations retournées par Oracle Solaris sur la zone correspondante.

Les informations affichées dans Horizon ou par la commande `nova` peuvent ne pas correspondre aux informations affichées par la commande `zoneadm` ou par d'autres commandes Oracle Solaris.

Nom	Le nom d'une instance de machine virtuelle tel qu'affiché dans Horizon ou par la commande <code>nova list</code> est le nom que vous avez affecté lors de la création de l'instance, tel que <code>example-instance</code> . Le nom de la zone affiché par la commande <code>zoneadm list</code> est identique à <code>instance-00000001</code> . Utilisez la commande <code>nova show</code> pour déterminer quelles zones sont associées à quelles instances de machine virtuelle. Dans la sortie <code>nova show</code> , la valeur de la propriété <code>OS-EXT-SRV-ATTR:instance_name</code> est le nom de la zone, et la valeur de la propriété <code>name</code> est le nom de l'instance de machine virtuelle.
UUID	L'UUID d'une instance de machine virtuelle telle qu'affichée dans Horizon ou par la commande <code>nova show</code> ne correspond pas à l'UUID de la même zone telle qu'affichée par la commande <code>zoneadm list -p</code> . L'UUID affichée par la commande <code>zoneadm</code> est un identificateur différent de celui utilisé pour Nova.
CPU	Le nombre de CPU virtuelles d'une instance de machine virtuelle tel qu'affiché dans Horizon est le nombre de CPU <i>plafonnées</i> qui sont virtualisées uniquement dans la mesure du nombre de fractions de CPU pouvant être utilisées par l'instance. Ce chiffre ne fournit pas d'observabilité à l'intérieur de l'instance plafonnée. La commande <code>psrinfo</code> signale les CPU dédiées qui sont allouées à la zone.
Mémoire	La quantité de mémoire d'une instance de machine virtuelle telle qu'affichée dans Horizon peut être différente de la quantité de mémoire que la commande <code>prtconf</code> affiche lorsque vous êtes connecté à cette instance de machine virtuelle. Horizon indique la quantité de mémoire spécifiée par la variante utilisée pour créer l'instance de machine virtuelle. La commande <code>prtconf</code> signale l'ensemble de la mémoire système.
Stockage	La quantité de stockage d'une instance de machine virtuelle telle qu'affichée dans Horizon peut être différente de la quantité de stockage affichée lorsque vous êtes connecté à cette instance de machine virtuelle, sauf si l'instance de machine virtuelle est une zone non globale utilisant ZOSS (Zones on Shared Storage).

Problèmes liés aux informations d'identification

Dans certains cas, vous pouvez recevoir des messages d'erreur relatifs aux informations d'identification incorrectes qui vous empêchent d'émettre des commandes de service. Par

exemple, lors de l'exécution d'une commande `glance`, il est possible que le message d'erreur suivant s'affiche :

```
Invalid OpenStack Identity credentials.
```

La cause à l'origine de ce message peut être différente à chaque fois. En conséquence, vous devez consulter les journaux pour limiter les causes probables. Par exemple, pour le service `glance`, consultez le contenu des journaux de service SMF Glance. Le fichier `/var/log/glance/api.log` peut signaler ce qui suit :

```
WARNING keystonemiddleware.auth_token [-] Authorization failed for token
```

Si vous avez défini `Debug = True` et `Verbose = True` dans les fichiers de configuration `glance`, vous verrez d'autres détails dans le fichier `/var/svc/log/application-openstack-glance-glance-api:default.log`, notamment :

```
DEBUG keystonemiddleware.auth_token [-] Received request from user:
  user_id None, project_id None, roles None service: user_id None,
  project_id None, roles None
__call__/_usr/lib/python2.7/vendor-packages/keystonemiddleware/auth_token.py:821
```

Vous pouvez explorer les zones suivantes pour résoudre le problème.

- Vérifiez les fichiers de configuration du service pour vous assurer que les paramètres appropriés sont correctement définis.
- Vérifiez que les variables shell globales sont correctement définies pour le service. Pour le service Glance, par exemple, les variables suivantes devraient être définies :
 - `OS_USERNAME=glance`
 - `OS_PASSWORD=service-password`
 - `OS_PROJECT_NAME=service`
 - `OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0`

Si la commande continue d'échouer et que les mêmes messages d'erreur sont affichés, il vous faudra peut-être recréer l'utilisateur du service pour générer de nouvelles informations d'identification. Etudiez l'exemple suivant :

```
# export OS_USERNAME=admin
# export OS_PASSWORD=service-password
# export OS_PROJECT_NAME=project
# export OS_AUTH_URL=http://$CONTROLLER_ADMIN_NODE:5000/v2.0

# openstack user list
```

Dans la sortie de cette commande, notez le numéro de l'ID de l'utilisateur du service altéré. Vous devrez supprimer cet utilisateur et le recréer avec des informations d'identification correctes :

```
# openstack user delete user-ID

# openstack user create --name glance --password service-password
# openstack user role add --user=glance --project=service --role=admin
```

Problème lié à Horizon

Une fois que vous avez lancé vos instances de machine virtuelle, le tableau de bord Horizon risque d'être inaccessible et le message d'erreur 404 Not Found s'affiche. Le journal des services Apache contient alors l'entrée suivante :

```
Oct 13 16:13:53 Executing start method (" /lib/svc/method/http-apache24 start"). ]
Apache version is 2.4
(125) Address already in use: AH000/2: make_sock: could not bind to address [::]:80
Oct 13 16:13:55 Method "start" exited with status 0. ]
```

Le journal indique que l'adresse ne peut pas se connecter au port 80 car ce dernier est occupé.

A partir de la version Kilo, OpenStack utilise Apache 2.4 au lieu d'Apache 2.2 dans les versions OpenStack précédentes. Vérifiez que la version correcte d'Apache est activée et en mode écoute sur ce port.

Pour libérer le port, procédez comme suit :

1. Obtenez l'ID de processus qui est actuellement à l'écoute sur le port.

```
# netstat -uan -f inet | grep "*.*80"
```

Cette procédure suppose que la configuration utilise les adresses IPv4 et que les processus faisant appel au port 80 sont à l'écoute sur toutes ces adresses. Si le processus écoute le trafic IPv6, la commande risque de ne donner aucun résultat.

2. En fonction de l'ID de processus, identifiez le processus ou le service réel. Vous pouvez utiliser l'une des commandes suivantes :

```
# svcs -p | egrep "online|pid http" | ggrep -B1 pid
```

ou

```
# ps -lf -p pid
```

3. Si la version incorrecte d'Apache utilise le port, désactivez le service.
4. Activez la version correcte d'Apache pour la version Kilo.

Si la version correcte d'Apache est en mode maintenance, supprimez d'abord le service avant de l'activer.

L'exemple suivant décrit comment libérer le port 80 et basculer vers la version Apache correcte.

```
# netstat -uan -f inet | grep "*.*80"
*.80      *.*      root      5560 httpd    0      0 128000    0 LISTEN
*.8080    *.*      webservd   1124 java     0      0 128000    0 LISTEN
*.8009    *.*      webservd   1124 java     0      0 128000    0 LISTEN

# svcs -p | egrep "online|5560 http" | ggrep -B1 5560
online    Aug_31   svc:/network/http:apache22
          Sep_09   5560 httpd
```

```
# svcadm disable apache22
# svcadm clear apache24
# svcadm enable apache24
```

Problèmes d'évolutivité

Par défaut, RabbitMQ impose une limite de descripteur de fichier de 255. Cette limite peut facilement empêcher l'évolutivité supplémentaire du cloud une fois que vous avez créé quelques noeuds de calcul. Pour éviter ce blocage, augmentez la valeur de la limite dans le fichier `/etc/rabbitmq/rabbitmq-env.conf` :

```
# Increase soft limit on file descriptors for RabbitMQ
ulimit -n 8192
```

Démontage de réseaux

Si vous rencontrez des problèmes lors de la configuration de Neutron sur le noeud Réseau et devez peut-être annuler la configuration pour recommencer, suivez la procédure ci-dessous. En fonction du point auquel vous devez commencer à annuler la configuration, suivez la séquence décrite dans la procédure.

▼ Suppression de la configuration réseau sur Neutron

1. Effectuez cette étape sur le tableau de bord Horizon.

- a. Dissociez toutes les adresses IP flottantes.

- b. Supprimez toutes les adresses IP flottantes.

2. Dans une fenêtre du terminal, tapez les commandes suivantes :

```
# neutron router-gateway-clear router-ID external-network-ID
```

```
# neutron router-interface-delete router-ID subnet-ID
```

- a. Pour supprimer l'interface de passerelle du routeur, tapez la commande suivante :

```
# neutron router-gateway-interface-delete router-ID external-network-ID
```


- Utilisez la commande `tail -30 `svcs -L service-name`` qui vous fournit des informations provenant du journal SMF. Si vous avez activé le débogage ainsi que la journalisation détaillée, vous devrez peut-être augmenter le nombre de lignes que vous spécifiez pour la commande `tail`.
- Les processus Horizon passent par Apache. Par conséquent, pour diagnostiquer Horizon, activez `debug = True` dans le fichier `/etc/openstack_dashboard/local_settings.py`. Les erreurs Django sont ensuite générées sur le page Web.
- Les composants Nova sont basés sur les zones d'Oracle Solaris. Vous pouvez donc vous reporter aux journaux figurant dans `/var/log/zones` pour résoudre des problèmes liés à Nova.

Sites utiles

Reportez-vous aux sites suivants pour des conseils de dépannage concernant la résolution des différents problèmes OpenStack.

- <https://blogs.oracle.com/openstack/>
- <https://ask.openstack.org/en/questions/>
- <https://raymii.org/s/tags/openstack.html>



Fichiers de configuration et services OpenStack courants

Cette annexe répertorie les fichiers de configuration des principaux composants OpenStack ainsi que les services SMF OpenStack.

Fichiers de configuration

Fichiers Cinder

- `/etc/cinder/api_paste.ini`
- `/etc/cinder/cinder.conf`

Fichiers Glance

- `/etc/glance/glance-api.conf`
- `/etc/glance/glance-cache.conf`
- `/etc/glance/glance-registry.conf`
- `/etc/glance/glance-scrubber.conf`
- `/etc/glance/glance-registry-paste.ini`
- `/etc/glance/glance-api-paste.ini`

Fichiers Keystone

- `/etc/keystone/keystone.conf`
- `/etc/keystone/keystone-paste.ini`

Fichiers Neutron

- `/etc/neutron/neutron.conf`
- `/etc/neutron/dhcp_agent.ini`
- `/etc/neutron/l3_agent.ini`
- `/etc/neutron/api-paste.ini`
- `/etc/neutron/metadata_agent.ini`
- `/etc/neutron/plugins/evs/evs_plugin.ini`

Fichiers Nova

- `/etc/nova/nova.conf`
- `/etc/nova/api_paste.ini`

Fichiers Horizon

- `/etc/openstack_dashboard/local_settings.py`
- `/etc/apache2/2.4/httpd.conf`
- `/etc/apache2/2.4/conf.d/openstack-dashboard-http.conf`
ou
`/etc/apache2/2.4/conf.d/openstack-dashboard-tls.conf`

Fichiers Swift

- `/etc/swift/swift.conf`
- `/etc/swift/account-server.conf`
- `/etc/swift/container-server.conf`
- `/etc/swift/object-server.conf`
- `/etc/swift/proxy-server.conf`
- `/etc/swift/rsyncd.conf`

Services SMF d'Openstack

Cinder

```
svc:/application/openstack/cinder/cinder-db:default
svc:/application/openstack/cinder/cinder-backup:default
svc:/application/openstack/cinder/cinder-scheduler:default
svc:/application/openstack/cinder/cinder-api:default
svc:/application/openstack/cinder/cinder-volume:setup
svc:/application/openstack/cinder/cinder-volume:default
```

Glance

```
svc:/application/openstack/glance/glance-db:default
svc:/application/openstack/glance/glance-registry:default
svc:/application/openstack/glance/glance-scrubber:default
svc:/application/openstack/glance/glance-api:default
```

Keystone

```
svc:/application/openstack/keystone:default
```

Neutron

```
svc:/application/openstack/neutron/neutron-server:default
svc:/application/openstack/neutron/neutron-dhcp-agent:default
svc:/application/openstack/neutron/neutron-metadata-agent:default
svc:/application/openstack/neutron/neutron-l3-agent:default
```

Nova *

```
svc:/application/openstack/nova/nova-objectstore:default
svc:/application/openstack/nova/nova-consoleauth:default
svc:/application/openstack/nova/nova-novncproxy:default
svc:/application/openstack/nova/nova-api-metadata:default
svc:/application/openstack/nova/nova-api-ec2:default
svc:/application/openstack/nova/nova-api-osapi-compute:default
svc:/application/openstack/nova/nova-conductor:default
svc:/application/openstack/nova/nova-cert:default
svc:/application/openstack/nova/nova-compute:default
svc:/application/openstack/nova/nova-scheduler:default
```

* D'autres services Nova peuvent également être répertoriés selon la configuration du noeud de calcul.

Swift

```
svc:/application/openstack/swift/swift-object-expirer:default
svc:/application/openstack/swift/swift-account-reaper:default
svc:/application/openstack/swift/swift-container-replicator:default
svc:/application/openstack/swift/swift-account-replicator:default
svc:/application/openstack/swift/swift-object-auditor:default
svc:/application/openstack/swift/swift-container-updater:default
svc:/application/openstack/swift/swift-container-sync:default
svc:/application/openstack/swift/swift-object-updater:default
svc:/application/openstack/swift/swift-account-auditor:default
svc:/application/openstack/swift/swift-replicator-rsync:default
svc:/application/openstack/swift/swift-container-auditor:default
svc:/application/openstack/swift/swift-object-replicator:default
svc:/application/openstack/swift/swift-container-reconciler:default
svc:/application/openstack/swift/swift-container-server:default
svc:/application/openstack/swift/swift-object-server:default
svc:/application/openstack/swift/swift-proxy-server:default
svc:/application/openstack/swift/swift-account-server:default
```

Commandes OpenStackClient

OpenStackClient (OSC)

OpenStackClient rassemble l'ensemble des commandes pour les API Calcul, Identité, Image, Stockage d'objets et Stockage de blocs dans un shell unique avec une structure de commande uniforme. Le nouvel ensemble de commandes utilise `openstack` comme commande principale au lieu des noms de composants qui étaient utilisés dans les versions précédentes d'OpenStack. L'implémentation du client a commencé avec la version Kilo. Cette annexe mappe les sous-commandes OpenStack précédentes vers le nouveau client.

Remarque - La liste des équivalents est partielle.

Pour obtenir plus d'informations sur la commande `openstack` et ses sous-commandes, utilisez l'une des commandes suivantes :

- `openstack help subcommand`
- `openstack --help`

Si vous entrez `openstack` sans sous-commandes, vous passez en mode interactif, ce qui vous permet de saisir `help [subcommand]` pour obtenir des informations. Pour quitter le mode interactif, saisissez `quit`.

Pour obtenir la liste détaillée des sous-commandes, des options et de la syntaxe `openstack`, reportez-vous à la section <http://docs.openstack.org/developer/python-openstackclient/index.html>.

TABLEAU 1 Commandes OpenStackClient équivalentes

Commande principale	Sous-commande	Sous-commandes équivalentes pour la commande principale <code>openstack</code>	Remarques
<code>cinder</code>			
	<code>type-create</code>	<code>volume type create</code>	
	<code>type-key</code>	<code>volume type set/volume type unset</code>	

Commande principale	Sous-commande	Sous-commandes équivalentes pour la commande principale openstack	Remarques
	service-list	service list	
glance			
	image-create	image create	API version 1 uniquement
	image-show	image show	
keystone			
	user-list	user list	
	user-role-add	user role add	
	user-role-list	user role list	
neutron			
	net-create	network create	
	router-create	router create??	
	router-list	router list??	
	router-show	router show??	
	subnet-list	subnet list	
nova			
	flavor-key	flavor set/flavor unset	dans la version OpenStack Liberty
	flavor-list	flavor list	
	flavor-show	flavor show	
	image-create	image create	API version 1 uniquement
	image-list	image list	
	image-show	image show	
	migrate	server migrate	
	resize	server resize	
	service-disable	compute service set/compute service unset	Keystone version 3 uniquement
	show	server show	

Index

A

- Accès à la console, 44
- Adresses IP flottantes, 74
 - Voir aussi* Réseau
- Adresses MAC dynamiques, 109
- Affichage des adresses MAC et des VID autorisés, 110
- Agent L3, 51, 58, 107
- allowed-mac-address, 108
- allowed-vlan-ids, 108
- archiveadm Commande, 60

B

- Base de données, 31

C

- Cinder
 - Configuration de systèmes de stockage distants, 91
 - Définition de listes de contrôle d'accès sur des hôtes distants, 96
 - Fichiers de configuration, 37, 46, 92, 103
 - Hôtes de stockage cible, 91
 - Installation, 37
 - Options de déploiement, 91
 - Pilote iSCSI Cinder pour ZFSSA, 100
 - Prise en charge d'un réseau de stockage, 91
 - Prise en charge NFS, 98
 - Profil de droits utilisateur, 95
 - Sauvegarde, 134
 - Utilisation de ZFSSA comme stockage d'arrière-plan, 102
 - Zone de disponibilité, 97
- Clés SSH
 - Dans la configuration EVS, 43
 - Migration d'instance, 85

- Cloudbase-Init, 130

- Commutateur virtuel élastique (EVS)

- Clés SSH, 43
 - Configuration, 43
 - evsadm Commande, 52
 - evsadm, commande, 54

D

- Déploiement sans système d'exploitation *Voir* Sans système d'exploitation

E

- Evolutivité, 143

F

- Fichier de configuration TLS, 37

G

- Glance
 - A propos de , 59
 - Affichage des informations d'image, 62
 - Fichiers de configuration, 34
 - Installation, 33
 - Script de création et de téléchargement, 63

H

- Heat
 - A propos de, 127
 - Avec Cloudbase-Init, 130
 - installation, 128
- Horizon

- Accès, 65
- Affichage des types d'instance, 80
- Configuration d'accès SSL, 36
- Liste de projets, 67
- Présentation, 66
- Présentation vidéo, 68

I

- ID de VLAN dynamiques, 109
- Image de secours, 60
- Images, 59
 - Voir aussi* Instances de machine virtuelle
 - A propos de, 68
 - Affichage des informations, 62
 - Image cache, 59
 - Instantanés, 60
 - Sauvegarde, 60
 - Secours, 60
 - Serveur de registre, 59
- Installation mononoeud d'OpenStack
 - Avec OVM Server for SPARC, 24
- Installation OpenStack
 - Configuration multinoeud, 23
- Installation OpenStack multinoeud
 - Architecture de référence à trois noeuds, 23
 - Préparation, 26
- Instance *Voir* Instances de machine virtuelle
- Instances de machine virtuelle, 74
 - Voir aussi* Nova
 - Ajout d'utilisateurs, 78
 - Création, 74
 - Images, 59, 68
 - Instantanés, 60
 - Migration, 14, 84
 - Paires de clés, 74
 - Redimensionnement, 14, 86
 - Sauvegarde, 60
 - Secours, 60
 - Types d'instance, 68
 - Variantes, 67, 80
- Instances de machine virtuelles
 - connexion, 78
- Instantanés, 60
- Ironic
 - Composants, 113

- Configuration AI, 114
- Déploiement sans système d'exploitation, 119
- Déploiement sans système d'exploitation à partir d'un fichier UAR, 121
- Echecs lors du provisionnement, 124
- Fichier `/etc/ironic/ironic.conf`, 116
- Installation et configuration, 114
- Mise hors service d'un noeud, 124
- Service, 113
- Utilisation des fonctions UAR et IPS, 120

K

- Keystone
 - Fichiers de configuration, 33
 - Installation, 32
 - `sample_data.sh` Script, 27

L

- LDoms, 24
- Locataires *Voir* Projets

M

- Machine virtuelle *Voir* Instances de machine virtuelle
- Machine virtuelle de cloud *Voir* Instances de machine virtuelle
- Migration de volumes Cinder, 13
- Migration en direct, 14, 84
- Mise à niveau
 - A partir de Havana, 18
 - A partir de Juno, 19
 - Considérations, 17
- Modèles d'instances *Voir* Variantes
- Modèles matériels *Voir* Variantes
- Modèles pour le matériel *Voir* Variantes
- MySQL, 31

N

- Neutron
 - Agent L3, 51, 58, 107
 - Création d'un routeur, 52
 - Dans des zones de noyau, 107

- Fichiers de configuration, 38
 - Installation, 38
 - Réseau externe, 53
 - Routeur, 51
 - Utilisation d'adresses MAC dynamiques et de VID, 107
 - Noeud de calcul, 23
 - Voir aussi* Nova
 - Configuration, 41
 - Récupération, 86
 - Noeud de contrôleur
 - A propos de, 23
 - Configuration, 30
 - Noeud de stockage
 - A propos de, 23
 - Configuration, 46
 - Plusieurs hôtes back-end, 91
 - Nova
 - Affichage des informations d'image, 62
 - Configuration, 41
 - Création d'instances de machine virtuelle, 74
 - Evacuation, 14, 86
 - Fichiers de configuration, 35, 42
 - Installation sur le noeud de contrôleur, 35
 - Migration d'instances, 14
 - Migration des instances, 84
 - Récupération de noeuds de calcul, 86
 - Redimensionnement, 86
 - Volume d'initialisation, 96
 - NTP (Network Time Protocol)
 - Client, 30
 - Fichier de configuration client, 31
 - Fichier de configuration serveur, 29
 - Serveur, 29
- O**
- Optimisation de la mémoire, 28
 - Options de débogage, 137
 - OVM Server for SPARC, 24
- P**
- Prise en charge d'un réseau de stockage dans Cinder, 91
 - Prise en charge de cloudbase-init, 15
 - Profil de configuration système *Voir* Profil SC
 - Profil de droits pour l'utilisateur Cinder, 95
 - Profil SC, 60, 82
 - Projets
 - Ajout d'utilisateurs, 68, 69
 - Création, 68
 - Modification du rôle de l'utilisateur, 69
 - Par défaut, 67
- R**
- RAD (Démon d'accès distant), 42
 - Réseau
 - Association d'adresses IP flottantes, 74
 - Connexion des réseaux internes au réseau externe, 57
 - Création d'un réseau externe, 54
 - Création du réseau externe, 51
 - Création du routeur, 52
 - Interne, 71
 - Sous-réseaux, 71
 - Réseau plat, 54
 - Routeur de fournisseur *Voir* Neutron
 - Routeur pour réseau externe, 52
- S**
- sample_data.sh Script, 27
 - Sans système d'exploitation, 119
 - Voir aussi* Ironic
 - Déploiement avec Ironic, 119
 - Déploiement en utilisant un fichier UAR, 121
 - Éléments de noeud configurables, 119
 - Mise hors service, 124
 - Utilisation des fonctions UAR et IPS, 120
 - Script pour la création et le téléchargement d'une image, 63
 - Serveur de registre, 59
 - Service SMF neutron-l3-agent , 52
 - SPARC, 24
 - SQLite, 31
 - Stockage SAN, 13
 - Structure de zones, 42
 - Swift
 - A propos de, 25
 - Fichiers de configuration, 48

Systèmes de stockage distants, 91

Voir aussi Cinder

Activation en tant que cibles, 96

Listes de contrôle d'accès (ACL), 96

Services SMF requis, 96

Voir aussi Cinder

Stockage d'arrière-plan pour OpenStack, 100

Utilitaire de workflow, 102

zonecfg:bootargs, option, 83

Zones de noyau hébergeant Neutron, 107

T

Tableau de bord

Affichage des images, 68

Affichage des types d'instance, 68, 80

Connexion, 65

Création d'une instance de machine virtuelle, 74

Création de réseaux internes, 71

Liste de projets, 67

Présentation vidéo, 68

Translation d'adresse réseau (NAT)

A propos de, 51

Désactivation d'une NAT sécurisée, 55

Types d'instance

A propos de, 68

extra_specs, propriété, 81

Modification des propriétés, 81

U

Unified Archives

Création d'images OpenStack avec la commande ,
60

user_reserve_hint_pct, 28

Utilitaire de gestion des services (SMF)

Services requis pour les hôtes cible, 96

V

Variantes, 80

Voir aussi Instances de machine virtuelle

Volume

Migration, 13

Sauvegarde et restauration, 13

Volumes NFS, 98

Z

ZFS Storage Appliance (ZFSSA), 100