



Red Hat OpenStack Platform 13

オーバークラウドの既存 Red Hat Ceph クラスターとの統合

オーバークラウドでスタンドアロンの Red Hat Ceph Storage を使用するための設定

Red Hat OpenStack Platform 13 オーバークラウドの既存 Red Hat Ceph クラスターとの統合

オーバークラウドでスタンドアロンの Red Hat Ceph Storage を使用するための設定

OpenStack Team
rhos-docs@redhat.com

法律上の通知

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

本ガイドでは、Red Hat OpenStack Platform director を使用して、オーバークラウドを既存のスタンドアロン Red Hat Ceph クラスタと統合する方法について説明します。

目次

多様性を受け入れるオープンソースの強化	3
第1章 はじめに	4
1.1. CEPH STORAGE の定義	4
1.2. シナリオの定義	4
第2章 オーバークラウドノードの準備	5
2.1. CEPH STORAGE のデプロイメント前の検証	5
2.2. 既存の CEPH STORAGE クラスターの設定	5
2.3. STACK ユーザーの初期化	7
2.4. ノードの登録	7
2.5. ノードの手動でのタグ付け	9
第3章 既存 CEPH クラスターとの統合	11
3.1. ASSIGNING NODES AND FLAVORS TO ROLES	12
3.2. オーバークラウドのデプロイ	13
第4章 オーバークラウドへのアクセス	15

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#) をご覧ください。

第1章 はじめに

Red Hat OpenStack Platform (RHOSP) director は、**オーバークラウド** と呼ばれるクラウド環境を作成します。director を使用して、オーバークラウドの追加機能を設定することができます。これらの追加機能の1つに、Red Hat Ceph Storage との統合が挙げられます。これには、director で作成した Ceph Storage クラスターおよび既存の Ceph Storage クラスターの両方が含まれます。

1.1. CEPH STORAGE の定義

Red Hat Ceph Storage は、優れたパフォーマンス、信頼性、スケーリングを提供するように設計された、分散型のデータオブジェクトストアです。非構造化データに対応しており、クライアントが新しいタイプのオブジェクトインターフェイスと従来のインターフェイスを同時に使用できる分散型のオブジェクトストアは、今後のストレージのあるべき姿です。すべての Ceph デプロイメントの中核となる Ceph Storage クラスターは、2種類のデーモンで設定されます。

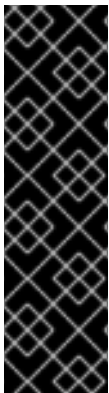
Ceph Object Storage デーモン

Ceph Object Storage デーモン (OSD) は、Ceph クライアントの代わりにデータを格納します。また、Ceph OSD は Ceph ノードの CPU とメモリーを使用して、データの複製、リバランス、復旧、監視、レポート作成を実行します。

Ceph Monitor

Ceph モニター (MON) は、ストレージクラスターの現在の状態が含まれる Ceph Storage クラスターのマッピングのマスターコピーを管理します。

Red Hat Ceph Storage に関する詳細は、[Red Hat Ceph Storage Architecture Guide](#) を参照してください。



重要

本ガイドでは、Block Storage を使用した Ceph の設定方法について説明します。

Ceph Object Gateway (RGW) に関する注記: この機能は、本リリースでは **テクノロジープレビュー** として提供しているため、Red Hat では全面的にはサポートしていません。これは、テスト用途にのみご利用いただく機能で、実稼働環境にデプロイすべきではありません。テクノロジープレビュー機能についての詳しい情報は、[対象範囲の詳細](#) を参照してください。

Ceph File System (CephFS) には対応していません。

1.2. シナリオの定義

本ガイドでは、既存の Ceph Storage クラスターをオーバークラウドと統合する手順について説明します。この場合、director はストレージ用途に Ceph Storage クラスターを使用するようにオーバークラウドを設定します。クラスターそのものは、オーバークラウド設定とは独立して管理およびスケーリングします。

第2章 オーバークラウドノードの準備

本章のシナリオでは、オーバークラウドは6台のノードで設定されます。

- 高可用性のコントローラーノード3台
- コンピュートノード3台

director は、独自のノードで設定される独立した Ceph Storage クラスタをオーバークラウドに統合します。このクラスタは、オーバークラウドとは別々に管理されます。たとえば、Ceph Storage クラスタは、OpenStack Platform director ではなく Ceph 管理ツールを使用してスケールアップします。詳しくは、[Red Hat Ceph Storage](#) のドキュメントを参照してください。

2.1. CEPH STORAGE のデプロイメント前の検証

オーバークラウドのデプロイメントが失敗しないようにするには、必要なパッケージがサーバーに存在することを確認します。

2.1.1. ceph-ansible パッケージバージョンの確認

アンダークラウドには Ansible ベースの検証が含まれ、これを実行してオーバークラウドをデプロイする前に潜在的な問題を特定することができます。これらの検証は、典型的な問題が発生する前にそれらを特定し、オーバークラウドのデプロイメントの失敗を回避するのに役立ちます。

手順

ceph-ansible パッケージの修正バージョンがインストールされていることを確認してください。

```
$ ansible-playbook -i /usr/bin/tripleo-ansible-inventory /usr/share/openstack-tripleo-validations/validations/ceph-ansible-installed.yaml
```

2.1.2. 事前にプロビジョニングされたノード用のパッケージの確認

オーバークラウドのデプロイメントで事前にプロビジョニングされたノードを使用する場合には、Ceph サービスをホストするオーバークラウドノードに必要なパッケージがサーバーにあることを確認することができます。

事前にプロビジョニングされたノードの詳細は、[事前にプロビジョニングされたノードを使用した基本的なオーバークラウドの設定](#) を参照してください。

手順

サーバーに必要なパッケージが含まれていることを確認します。

```
ansible-playbook -i /usr/bin/tripleo-ansible-inventory /usr/share/openstack-tripleo-validations/validations/ceph-dependencies-installed.yaml
```

2.2. 既存の CEPH STORAGE クラスタの設定

1. お使いの環境に適した Ceph クラスタに以下のプールを作成します。

- **volumes:** OpenStack Block Storage (cinder) のストレージ
- **images:** OpenStack Image Storage (glance) のストレージ

- **vms**: インスタンスのストレージ
- **backups**: OpenStack Block Storage Backup (cinder-backup) のストレージ
- **metrics**: OpenStack Telemetry Metrics (gnocchi) のストレージ
以下のコマンドは指針として使用してください。

```
[root@ceph ~]# ceph osd pool create volumes PGNUM
[root@ceph ~]# ceph osd pool create images PGNUM
[root@ceph ~]# ceph osd pool create vms PGNUM
[root@ceph ~]# ceph osd pool create backups PGNUM
[root@ceph ~]# ceph osd pool create metrics PGNUM
```

PGNUM は **配置グループ** の数に置き換えます。1 OSD につき 100 程度を推奨します。たとえば、OSD の合計数を 100 で乗算して、レプリカ数で除算します (**osd pool default size**)。適切な値を判断するには [Ceph Placement Groups \(PGs\) per Pool Calculator](#) を使用することを推奨します。

2. 以下の機能を指定して **client.openstack** ユーザーを Ceph クラスターに作成します。

- **cap_mgr**: "allow *"
- **cap_mon**: profile rbd
- **cap_osd**: profile rbd pool=volumes, profile rbd pool=vms, profile rbd pool=images, profile rbd pool=backups, profile rbd pool=metrics
以下のコマンドは指針として使用してください。

```
[root@ceph ~]# ceph auth add client.openstack mgr 'allow *' mon 'profile rbd' osd 'profile rbd pool=volumes, profile rbd pool=vms, profile rbd pool=images, profile rbd pool=backups, profile rbd pool=metrics'
```

3. **client.openstack** ユーザー向けに作成された Ceph クライアントキーをメモします。

```
[root@ceph ~]# ceph auth list
...
[client.openstack]
key = AQC+vYNXgDAgAhAAc8UoYt+OTz5uhV7ltLdwUw==
caps mgr = "allow *"
caps mon = "profile rbd"
caps osd = "profile rbd pool=volumes, profile rbd pool=vms, profile rbd pool=images, profile rbd pool=backups, profile rbd pool=metrics"
...
```

この例の **key** 値 (**AQC+vYNXgDAgAhAAc8UoYt+OTz5uhV7ltLdwUw==**) は Ceph クライアントキーです。

4. Ceph Storage クラスターの **ファイルシステム ID** をメモします。この値は、クラスターの設定ファイルにある **fsid** の設定で指定されています (**[global]** のセクション下)。

```
[global]
fsid = 4b5c8c0a-ff60-454b-a1b4-9747aa737d19
...
```



注記

Ceph Storage クラスターの設定ファイルに関する詳しい情報は、[Red Hat Ceph Storage Configuration Guide](#) の [Configuration Reference](#) を参照してください。

Ceph クライアントキーおよびファイルシステム ID はいずれも、後ほど3章 [既存 Ceph クラスターとの統合](#) で使用します。

2.3. STACK ユーザーの初期化

stack ユーザーとして director ホストにログインし、以下のコマンドを実行して director の設定を初期化します。

```
$ source ~/stackrc
```

このコマンドでは、director の CLI ツールにアクセスする認証情報が含まれる環境変数を設定します。

2.4. ノードの登録

ノード定義のテンプレート (**instackenv.json**) は JSON ファイル形式で、ノード登録用のハードウェアおよび電源管理の情報が含まれています。以下に例を示します。

```
{
  "nodes":[
    {
      "mac":[
        "bb:bb:bb:bb:bb:bb"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"pxe_ipmitool",
      "pm_user":"admin",
      "pm_password":"p@55w0rd!",
      "pm_addr":"192.0.2.205"
    },
    {
      "mac":[
        "cc:cc:cc:cc:cc:cc"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"pxe_ipmitool",
      "pm_user":"admin",
      "pm_password":"p@55w0rd!",
      "pm_addr":"192.0.2.206"
    },
    {
      "mac":[
        "dd:dd:dd:dd:dd:dd"
      ],

```

```
    "cpu": "4",
    "memory": "6144",
    "disk": "40",
    "arch": "x86_64",
    "pm_type": "pxe_ipmitool",
    "pm_user": "admin",
    "pm_password": "p@55w0rd!",
    "pm_addr": "192.0.2.207"
  },
  {
    "mac": [
      "ee:ee:ee:ee:ee:ee"
    ],
    "cpu": "4",
    "memory": "6144",
    "disk": "40",
    "arch": "x86_64",
    "pm_type": "pxe_ipmitool",
    "pm_user": "admin",
    "pm_password": "p@55w0rd!",
    "pm_addr": "192.0.2.208"
  }
  {
    "mac": [
      "ff:ff:ff:ff:ff:ff"
    ],
    "cpu": "4",
    "memory": "6144",
    "disk": "40",
    "arch": "x86_64",
    "pm_type": "pxe_ipmitool",
    "pm_user": "admin",
    "pm_password": "p@55w0rd!",
    "pm_addr": "192.0.2.209"
  }
  {
    "mac": [
      "gg:gg:gg:gg:gg:gg"
    ],
    "cpu": "4",
    "memory": "6144",
    "disk": "40",
    "arch": "x86_64",
    "pm_type": "pxe_ipmitool",
    "pm_user": "admin",
    "pm_password": "p@55w0rd!",
    "pm_addr": "192.0.2.210"
  }
]
}
```

テンプレートを作成したら、stack ユーザーのホームディレクトリーにファイルを保存します (/home/stack/instackenv.json)。stack ユーザーを初期化し、続いて **instackenv.json** を director にインポートします。

```
$ source ~/stackrc
$ openstack overcloud node import ~/instackenv.json
```

このコマンドでテンプレートをインポートして、テンプレートから director に各ノードを登録します。
カーネルと ramdisk イメージを各ノードに割り当てます。

```
$ openstack overcloud node configure <node>
```

director でのノードの登録、設定が完了しました。

2.5. ノードの手動でのタグ付け

各ノードを登録した後は、ハードウェアを検査して、ノードを特定のプロファイルにタグ付けする必要があります。プロファイルタグにより、ノードがフレーバーに照合され、そのフレーバーはデプロイメントロールに割り当てられます。

新規ノードを検査してタグ付けするには、以下のステップに従います。

1. ハードウェアのイントロスペクションをトリガーして、各ノードのハードウェア属性を取得します。

```
$ openstack overcloud node introspect --all-manageable --provide
```

- **--all-manageable** オプションは、管理状態のノードのみをイントロスペクションします。上記の例では、すべてのノードが対象です。
- **--provide** オプションは、イントロスペクション後に全ノードを **active** の状態にリセットします。



重要

このプロセスが最後まで実行されて正常に終了したことを確認してください。ベアメタルノードの場合には、通常 15 分ほどかかります。

2. ノード一覧を取得して UUID を把握します。

```
$ openstack baremetal node list
```

3. 各ノードの **properties/capabilities** パラメーターに **profile** オプションを追加して、ノードを特定のプロファイルに手動でタグ付けします。
たとえば、3つのノードが **control** プロファイルを使用し、別の3つのノードが **compute** プロファイルを使用するようにタグ付けするには、以下のコマンドを実行します。

```
$ ironic node-update 1a4e30da-b6dc-499d-ba87-0bd8a3819bc0 add
properties/capabilities='profile:control,boot_option:local'
$ ironic node-update 6faba1a9-e2d8-4b7c-95a2-c7fbdcd12129a add
properties/capabilities='profile:control,boot_option:local'
$ ironic node-update 5e3b2f50-fcd9-4404-b0a2-59d79924b38e add
properties/capabilities='profile:control,boot_option:local'
$ ironic node-update 484587b2-b3b3-40d5-925b-a26a2fa3036f add
properties/capabilities='profile:compute,boot_option:local'
$ ironic node-update d010460b-38f2-4800-9cc4-d69f0d067efe add
```

```
properties/capabilities='profile:compute,boot_option:local'  
$ ironic node-update d930e613-3e14-44b9-8240-4f3559801ea6 add  
properties/capabilities='profile:compute,boot_option:local'
```

profile オプションを追加すると、適切なプロファイルにノードをタグ付けします。



注記

手動でのタグ付けの代わりに、Automated Health Check (AHC) ツールを使用し、ベンチマークデータに基づいて、多数のノードに自動でタグ付けします。

第3章 既存 CEPH クラスターとの統合

director の提供する heat テンプレートコレクションには、オーバークラウドをデプロイするのに必要なテンプレートおよび環境ファイルがすでに含まれます。

デプロイメント時(「[オーバークラウドのデプロイ](#)」)にこの環境ファイルを呼び出し、デプロイ中のオーバークラウドに既存の Ceph クラスターを統合します。

- `/usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml`

手順

1. director は **ceph-ansible** を使用して既存 Ceph クラスターとの統合を行います。デフォルトでは **ceph-ansible** はアンダークラウドにインストールされません。以下のコマンドを入力して、アンダークラウドに ceph-ansible パッケージをインストールしてください。

```
sudo yum install -y ceph-ansible
```

2. 統合を設定するには、Ceph クラスターの詳細を director に提供する必要があります。カスタム環境ファイルを使用して、`/usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml` で使用されるデフォルト設定をオーバーライドします。

3. 以下のカスタム環境ファイルを作成します。
`/home/stack/templates/ceph-config.yaml`

4. このファイルに **parameter_defaults:** ヘッダーを追加します。

```
parameter_defaults:
```

5. このヘッダーの下に、`/usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml` を上書きするすべてのパラメーターを設定します。少なくとも、以下のパラメーターを設定する必要があります。

- **CephClientKey:** Ceph Storage クラスターの Ceph クライアントキー。これは、先ほど「[既存の Ceph Storage クラスターの設定](#)」で取得した **key** の値です。たとえば、**AQDLOh1VgEp6FRAAFzT7Zw+Y9V6JJExQAsRnRQ==** です。
- **CephClusterFSID:** Ceph Storage クラスターのファイルシステム ID。これは、先ほど「[既存の Ceph Storage クラスターの設定](#)」で取得した Ceph Storage クラスター設定ファイルの **fsid** の値ですたとえば、**4b5c8c0a-ff60-454b-a1b4-9747aa737d19** です。
- **CephExternalMonHost:** Ceph Storage クラスターの全 MON ホストの IP をコンマ区切りにしたリスト (例: **172.16.1.7, 172.16.1.8**)。

```
parameter_defaults:
  CephClientKey: AQDLOh1VgEp6FRAAFzT7Zw+Y9V6JJExQAsRnRQ==
  CephClusterFSID: 4b5c8c0a-ff60-454b-a1b4-9747aa737d19
  CephExternalMonHost: 172.16.1.7, 172.16.1.8
```

6. 必要な場合は、以下のパラメーターと値を使用して、OpenStack プールとクライアントユーザーの名前を設定します。

- **CephClientUserName:** **openstack**

- **NovaRbdPoolName: vms**
 - **CinderRbdPoolName: volumes**
 - **GlanceRbdPoolName: images**
 - **CinderBackupRbdPoolName: backups**
 - **GnocchiRbdPoolName: metrics**
7. カスタム環境ファイルにオープンクラウドパラメーターを追加することもできます。たとえば、**neutron** のネットワーク種別を **vxlan** に設定するには、**parameter_defaults** に以下の設定を追加します。

```
NeutronNetworkType: vxlan
```

3.1. ASSIGNING NODES AND FLAVORS TO ROLES

オープンクラウドのデプロイメントのプランニングでは、各ロールに割り当てるノード数とフレーバーを指定する必要があります。すべての Heat テンプレートのパラメーターと同様に、これらのロールの仕様はカスタム環境ファイル (ここでは [3章既存 Ceph クラスターとの統合](#) の `/home/stack/templates/ceph-config`) の **parameter_defaults** セクションで宣言する必要があります。

この設定には、以下のパラメーターを使用します。

表3.1 オープンクラウドノードのロールとフレーバー

Heat テンプレートのパラメーター	説明
ControllerCount	スケールアウトするコントローラーノード数
OvercloudControlFlavor	コントローラーノードに使用するフレーバー (control)
ComputeCount	スケールアウトするコンピューターノード数
OvercloudComputeFlavor	コンピューターノードに使用するフレーバー (compute)

たとえば、オープンクラウドが各ロール (Controller および Compute) に 3 つずつノードをデプロイするように設定するには、**parameter_defaults** に以下の設定を追加します。

```
parameter_defaults:
  ControllerCount: 3
  ComputeCount: 3
  OvercloudControlFlavor: control
  OvercloudComputeFlavor: compute
```



注記

Heat テンプレートのパラメーターのより詳細な一覧は、[director のインストールと使用方法](#) の CLI ツールを使用したオープンクラウドの作成を参照してください。

3.2. オーバークラウドのデプロイ



注記

アンダークラウドのインストール時に、**undercloud.conf** ファイルに **generate_service_certificate=false** を設定します。設定しない場合は、オーバークラウドのデプロイ時にトラストアンカーを挿入する必要があります。トラストアンカーの挿入方法についての詳細は、[オーバークラウドの高度なカスタマイズガイドのオーバークラウドのパブリックエンドポイントでの SSL/TLS の有効化](#) を参照してください。

オーバークラウドの作成には、**openstack overcloud deploy** コマンドに追加の引数を指定する必要があります。以下に例を示します。

```
$ openstack overcloud deploy --templates \
  -e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml \
  -e /home/stack/templates/ceph-config.yaml \
  -e --ntp-server pool.ntp.org \
```

上記のコマンドは、以下のオプションを使用します。

- **--templates:** デフォルトの Heat テンプレートコレクション (**/usr/share/openstack-tripleo-heat-templates/**) からオーバークラウドを作成します。
- **-e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml:** 既存の Ceph クラスターをオーバークラウドに統合するように director を設定します。
- **-e /home/stack/templates/ceph-config.yaml:** **-e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml** の設定するデフォルトを上書きするためのカスタム環境ファイルを追加します。ここでは、[3章既存 Ceph クラスターとの統合](#)で作成したカスタム環境ファイルです。
- **--ntp-server pool.ntp.org:** NTP サーバーを設定します。

ヒント

アンサーファイル を使用して、すべてのテンプレートおよび環境ファイルを呼び出すこともできます。たとえば、以下のコマンドを使用して、同一のオーバークラウドをデプロイすることができます。

```
$ openstack overcloud deploy \
  --answers-file /home/stack/templates/answers.yaml \
  --ntp-server pool.ntp.org
```

この場合、アンサーファイル **/home/stack/templates/answers.yaml** の内容は以下のようになります。

```
templates: /usr/share/openstack-tripleo-heat-templates/
environments:
  - /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml
  \
  - /home/stack/templates/ceph-config.yaml \
```

詳細は、[オーバークラウド作成時の環境ファイルの追加](#) を参照してください。

オプションの完全な一覧を表示するには、以下を実行します。

```
$ openstack help overcloud deploy
```

詳細は、**director** のインストールと使用方法の [CLI ツールを使用したオーバークラウドの作成](#) を参照してください。

オーバークラウドの作成プロセスが開始され、director によりノードがプロビジョニングされます。このプロセスは完了するまで多少時間がかかります。オーバークラウドの作成のステータスを確認するには、**stack** ユーザーとして別のターミナルを開き、以下のコマンドを実行します。

```
$ source ~/stackrc  
$ openstack stack list --nested
```

この設定では、オーバークラウドが外部の Ceph Storage クラスターを使用するように設定します。このクラスターは、オーバークラウドから独立して、管理される点に注意してください。たとえば、Ceph Storage クラスターは、OpenStack Platform director ではなく Ceph 管理ツールを使用してスケールリングします。

第4章 オーバークラウドへのアクセス

director は、director ホストからオーバークラウドに対話するための設定を行い、認証をサポートするスクリプトを作成します。director は、このファイル (**overcloudrc**) を **stack** ユーザーのホームディレクトリーに保存します。このファイルを使用するには、以下のコマンドを実行します。

```
$ source ~/overcloudrc
```

これにより、director ホストの CLI からオーバークラウドと対話するために必要な環境変数が読み込まれます。director のホストとの対話に戻るには、以下のコマンドを実行します。

```
$ source ~/stackrc
```