



Red Hat OpenStack Platform

11

ロギング、モニタリング、トラブル シューティングガイド

OpenStack のロギング、モニタリング、トラブルシューティングの詳細ガイド

OpenStack Team

Red Hat OpenStack Platform 11 ログイング、モニタリング、トラブルシューティングガイド

OpenStack のログイング、モニタリング、トラブルシューティングの詳細ガイド

OpenStack Team
rhos-docs@redhat.com

法律上の通知

Copyright © 2017 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, 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 Software Collections 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

本ガイドは、Red Hat OpenStack Platform 環境のロギングおよびモニタリング、さらには問題の解決方法について詳しく説明します。

目次

第1章 本ガイドについて	4
第2章 ロギング	5
2.1. OPENSTACK サービスのログファイル	5
2.1.1. Bare Metal Provisioning (Ironic) のログファイル	5
2.1.2. Block Storage (cinder) ログファイル	5
2.1.3. Compute (nova) ログファイル	6
2.1.4. Dashboard (horizon) ログファイル	6
2.1.5. Data Processing (sahara) ログファイル	7
2.1.6. Database as a Service (trove) ログファイル	8
2.1.7. Identity サービス (keystone) ログファイル	8
2.1.8. Image Service (glance) ログファイル	8
2.1.9. Networking (neutron) ログファイル	9
2.1.10. Object Storage (swift) ログファイル	9
2.1.11. Orchestration (heat) ログファイル	10
2.1.12. Shared Filesystem サービス (manila) ログファイル	10
2.1.13. Telemetry (ceilometer) ログファイル	11
2.1.14. 補足サービスのログファイル	12
2.2. ロギングオプションの設定	12
2.3. リモートロギングのインストールおよび設定	13
2.3.1. リモートロギングの概要	13
2.3.2. rsyslog サーバーのインストール	13
2.3.3. 集中ロギングサーバーでの rsyslog の設定	13
2.3.4. 個々のノードでの rsyslog 設定	14
2.3.5. rsyslog サーバーの起動	14
第3章 TELEMETRY サービスを使用したモニタリング	16
3.1. 既存のアラームの表示	16
3.2. アラームの作成	16
3.3. アラームの無効化または削除	17
3.4. 測定の表示	18
3.5. 新規測定の作成	18
3.6. クラウドの使用状況の測定	18
3.7. L3 キャッシュの使用状況の表示	18
3.8. インスタンスのディスクアクティビティの監視	18
3.9. リソースタイプの管理	20
3.10. TIME-SERIES-DATABASE-AS-A-SERVICE の使用	21
3.10.1. Time-Series-Database-as-a-Service の実行	21
3.10.2. WSGI アプリケーションとしての実行	21
3.10.3. metricd ワーカー	22
3.10.4. Time-Series-Database-as-a-Service の監視	22
3.10.5. Time-Series-Database-as-a-Service のバックアップと復元	22
第4章 トラブルシューティング	23
4.1. サポート	23
4.2. IDENTITY クライアント (KEYSTONE) の接続性における問題のトラブルシューティング	23
4.3. OPENSTACK NETWORKING に関する問題のトラブルシューティング	24
4.4. ダッシュボードでのネットワークまたはルータータブの表示に関するトラブルシューティング	25
4.5. DASHBOARD でのインスタンス起動エラーに関するトラブルシューティング	25
4.6. DASHBOARD の KEYSTONE V3 認証のトラブルシューティング	26
4.7. OPENSTACK DASHBOARD: RED HAT ACCESS タブ	28
4.7.1. 検索	29
4.7.2. ログ	--

4.7.2. ログ	30
4.7.3. サポート	31

第1章 本ガイドについて

警告

現在 Red Hat では、本リリースのガイドに記載の情報および手順のレビューを行なっています。

本書は、https://access.redhat.com/documentation/ja-jp/red_hat_openstack_platform/?version=10 で提供している Red Hat OpenStack Platform 10 のドキュメントをベースにしています。

Red Hat OpenStack Platform 11 でサポートが必要な場合には、Red Hat サポートまでお問い合わせください。

本ガイドは、Red Hat OpenStack Platform 環境で利用可能なログインおよびモニタリング機能の概要と発生する可能性のある問題のトラブルシューティングの方法を説明します。

第2章 ログイン

Red Hat OpenStack Platform は、特定のログファイルに情報メッセージを書き込みます。このメッセージは、トラブルシューティングやシステムイベントのモニタリングに使用することができます。



注記

個別のログファイルをサポートケースに手動で添付する必要はありません。必要な情報はすべて **sosreport** で自動的に収集されます。この件に関しては、「[4章 トラブルシューティング](#)」で説明しています。

2.1. OPENSTACK サービスのログファイル

OpenStack のコンポーネントごとに、実行サービス固有のファイルを含む個別のログインディレクトリーがあります。

2.1.1. Bare Metal Provisioning (Ironic) のログファイル

サービス	サービス名	ログへのパス
OpenStack Ironic API	openstack-ironic-api.service	/var/log/ironic/ironic-api.log
OpenStack Ironic Conductor	openstack-ironic-conductor.service	/var/log/ironic/ironic-conductor.log

2.1.2. Block Storage (cinder) ログファイル

サービス	サービス名	ログへのパス
Block Storage API	openstack-cinder-api.service	/var/log/cinder/api.log
Block Storage バックアップ	openstack-cinder-backup.service	/var/log/cinder/backup.log
情報メッセージ	cinder-manage コマンド	/var/log/cinder/cinder-manage.log
Block Storage スケジューラ	openstack-cinder-scheduler.service	/var/log/cinder/scheduler.log

サービス	サービス名	ログへのパス
Block Storage ボリューム	openstack-cinder-volume.service	/var/log/cinder/volume.log

2.1.3. Compute (nova) ログファイル

サービス	サービス名	ログへのパス
OpenStack Compute API サービス	openstack-nova-api.service	/var/log/nova/nova-api.log
OpenStack Compute 証明書サーバー	openstack-nova-cert.service	/var/log/nova/nova-cert.log
OpenStack Compute サービス	openstack-nova-compute.service	/var/log/nova/nova-compute.log
OpenStack Compute コンダクターサービス	openstack-nova-conductor.service	/var/log/nova/nova-conductor.log
OpenStack Compute VNC コンソールの認証サーバー	openstack-nova-consoleauth.service	/var/log/nova/nova-consoleauth.log
情報メッセージ	nova-manage コマンド	/var/log/nova/nova-manage.log
OpenStack Compute NoVNC プロキシサービス	openstack-nova-novncproxy.service	/var/log/nova/nova-novncproxy.log
OpenStack Compute スケジューラーサービス	openstack-nova-scheduler.service	/var/log/nova/nova-scheduler.log

2.1.4. Dashboard (horizon) ログファイル

サービス	サービス名	ログへのパス
------	-------	--------

サービス	サービス名	ログへのパス
特定のユーザーとの対話ログ	Dashboard インターフェース	/var/log/horizon/horizon.log

Apache HTTP サーバーは、Dashboard Web インターフェース用に追加のログファイルを複数使用します。これらのファイルは、Web ブラウザーまたはコマンドラインクライアント (keystone、nova) を使用してアクセスできます。以下のログファイルは、Dashboard の使用のトラッキングや、問題診断に役立ちます。

目的	ログへのパス
処理済みの HTTP 要求すべて	/var/log/httpd/horizon_access.log
HTTP エラー	/var/log/httpd/horizon_error.log
管理者ロールの API 要求	/var/log/httpd/keystone_wsgi_admin_access.log
管理者ロールの API エラー	/var/log/httpd/keystone_wsgi_admin_error.log
メンバーロールの API 要求	/var/log/httpd/keystone_wsgi_main_access.log
メンバーロールの API エラー	/var/log/httpd/keystone_wsgi_main_error.log



注記

同じホスト上で実行する他の Web サービスから報告されたエラーを格納する **/var/log/httpd/default_error.log** もあります。

2.1.5. Data Processing (sahara) ログファイル

サービス	サービス名	ログへのパス
Sahara API サーバー	openstack-sahara-all.service openstack-sahara-api.service	/var/log/sahara/sahara-all.log /var/log/messages

サービス	サービス名	ログへのパス
Sahara Engine サーバー	openstack-sahara-engine.service	/var/log/messages

2.1.6. Database as a Service (trove) ログファイル

サービス	サービス名	ログへのパス
OpenStack Trove API サービス	openstack-trove-api.service	/var/log/trove/trove-api.log
OpenStack Trove Conductor サービス	openstack-trove-conductor.service	/var/log/trove/trove-conductor.log
OpenStack Trove guestagent サービス	openstack-trove-guestagent.service	/var/log/trove/logfile.txt
OpenStack Trove taskmanager サービス	openstack-trove-taskmanager.service	/var/log/trove/trove-taskmanager.log

2.1.7. Identity サービス (keystone) ログファイル

サービス	サービス名	ログへのパス
OpenStack Identity サービス	openstack-keystone.service	/var/log/keystone/keystone.log

2.1.8. Image Service (glance) ログファイル

サービス	サービス名	ログへのパス
OpenStack Image Service API サーバー	openstack-glance-api.service	/var/log/glance/api.log
OpenStack Image Service レジストリーサーバー	openstack-glance-registry.service	/var/log/glance/registry.log

2.1.9. Networking (neutron) ログファイル

サービス	サービス名	ログへのパス
OpenStack Neutron DHCP エージェント	neutron-dhcp-agent.service	/var/log/neutron/dhcp-agent.log
OpenStack Networking レイヤー 3 エージェント	neutron-l3-agent.service	/var/log/neutron/l3-agent.log
メタデータエージェントサービス	neutron-metadata-agent.service	/var/log/neutron/metadata-agent.log
メタデータの名前空間プロキシ	なし	/var/log/neutron/neutron-ns-metadata-proxy- UUID .log
Open vSwitch エージェント	neutron-openvswitch-agent.service	/var/log/neutron/openvswitch-agent.log
OpenStack Networking サービス	neutron-server.service	/var/log/neutron/server.log

2.1.10. Object Storage (swift) ログファイル

OpenStack Object Storage は、システムのロギング機能にのみ、ログを送信します。



注記

デフォルトでは、Object Storage ログファイルはすべて local0、local1、local2 syslog 機能を使用して /var/log/swift/swift.log に送られます。

Object Storage のログメッセージは主に、REST API サービスのログメッセージと、バックグラウンドデーモンのログメッセージの 2 つのカテゴリに分類されます。API サービスのメッセージには、API 要求ごとに 1 行含まれています。これは、一般的に使用されている HTTP サーバーによく似た形式となっており、フロントエンド (プロキシ) およびバックエンド (アカウント、コンテナ、オブジェクト) の両サービスがこのようなメッセージをポストします。デーモンメッセージは、(API サービスのものとは比べ) 構造化されておらず、通常、定期的なタスクを実行するデーモンに関する情報が人間が判読できる形で含まれています。ただし、Object Storage のどの部分がメッセージを生成するかに関わらず、ソースのアイデンティティは必ず行頭に記載されます。

プロキシメッセージ例

```
Apr 20 15:20:34 rhev-a24c-01 proxy-server: 127.0.0.1 127.0.0.1
20/Apr/2015/19/20/34 GET
```

```
/v1/AUTH_zaitcev%3Fformat%3Djson%26marker%3Dtestcont HTTP/1.0 200 -
python-swiftclient-2.1.0 AUTH_tk737d6... - 2 - txc454fa8ea4844d909820a-
0055355182 - 0.0162 - - 1429557634.806570053 1429557634.822791100
```

バックグラウンドデーモンからのアドホックメッセージ例

```
Apr 27 17:08:15 rhev-a24c-02 object-auditor: Object audit (ZBF). Since
Mon Apr 27 21:08:15 2015: Locally: 1 passed, 0 quarantined, 0 errors
files/sec: 4.34 , bytes/sec: 0.00, Total time: 0.23, Auditing time:
0.00, Rate: 0.00
Apr 27 17:08:16 rhev-a24c-02 object-auditor: Object audit (ZBF)
"forever" mode completed: 0.56s. Total quarantined: 0, Total errors: 0,
Total files/sec: 14.31, Total bytes/sec: 0.00, Auditing time: 0.02,
Rate: 0.04
Apr 27 17:08:16 rhev-a24c-02 account-replicator: Beginning replication
run
Apr 27 17:08:16 rhev-a24c-02 account-replicator: Replication run OVER
Apr 27 17:08:16 rhev-a24c-02 account-replicator: Attempted to replicate
5 dbs in 0.12589 seconds (39.71876/s)
Apr 27 17:08:16 rhev-a24c-02 account-replicator: Removed 0 dbs
Apr 27 17:08:16 rhev-a24c-02 account-replicator: 10 successes, 0
failures
```

2.1.11. Orchestration (heat) ログファイル

サービス	サービス名	ログへのパス
OpenStack Heat API サービス	openstack-heat-api.service	/var/log/heat/heat-api.log
Openstack Heat エンジンサービス	openstack-heat-engine.service	/var/log/heat/heat-engine.log
Orchestration サービスのイベント	なし	/var/log/heat/heat-manage.log

2.1.12. Shared Filesystem サービス (manila) ログファイル

サービス	サービス名	ログへのパス
OpenStack Manila API サーバー	openstack-manila-api.service	/var/log/manila/api.log
OpenStack Manila スケジューラー	openstack-manila-scheduler.service	/var/log/manila/scheduler.log

サービス	サービス名	ログへのパス
OpenStack Manila 共有サービス	openstack-manila-share.service	/var/log/manila/share.log



注記

Manila Python ライブラリーからの情報の一部は `/var/log/manila/manila-manage.log` にロギングすることもできます。

2.1.13. Telemetry (ceilometer) ログファイル

サービス	サービス名	ログへのパス
OpenStack ceilometer 通知エージェント	openstack-ceilometer-notification.service	/var/log/ceilometer/agent-notification.log
OpenStack ceilometer アラーム評価	openstack-ceilometer-alarm-evaluator.service	/var/log/ceilometer/alarm-evaluator.log
OpenStack ceilometer アラーム通知	openstack-ceilometer-alarm-notifier.service	/var/log/ceilometer/alarm-notifier.log
OpenStack ceilometer API	httpd.service	/var/log/ceilometer/api.log
情報メッセージ	MongoDB integration	/var/log/ceilometer/ceilometer-dbsync.log
OpenStack ceilometer 中央エージェント	openstack-ceilometer-central.service	/var/log/ceilometer/central.log
OpenStack ceilometer コレクション	openstack-ceilometer-collector.service	/var/log/ceilometer/collector.log
OpenStack ceilometer コンピュートエージェント	openstack-ceilometer-compute.service	/var/log/ceilometer/compute.log

2.1.14. 補足サービスのログファイル

以下のサービスは、中核となる OpenStack コンポーネントにより使用されており、サービスごとに独自のログのディレクトリーとファイルが存在します。

サービス	サービス名	ログへのパス
メッセージブローカー (RabbitMQ)	rabbitmq-server.service	/var/log/rabbitmq/rabbit@short_hostname.log /var/log/rabbitmq/rabbit@short_hostname-sasl.log (Simple Authentication and Security Layer 関連のログメッセージ)
データベースサーバー (MariaDB)	mariadb.service	/var/log/mariadb/mariadb.log
ドキュメント指向データベース (MongoDB)	mongod.service	/var/log/mongodb/mongodb.log
仮想ネットワークス イッチ (Open vSwitch)	openvswitch- nonetwork.service	/var/log/openvswitch/ovsdb-server.log /var/log/openvswitch/ovs-vsitchd.log

2.2. ロギングオプションの設定

コンポーネントごとに、別のロギング設定が適切な設定ファイルに保管されています。たとえば、Compute ではこれらのオプションは `/etc/nova/nova.conf` に設定されます。

- ※ デバッグを有効化することで、情報ロギングのレベルを上げます。このオプションにより、取得する情報量が大幅に増加するため、この機能を一時的にだけ使用するか、またはログの回転設定を先に確認するようにしてください。

```
debug=True
```

- ※ 詳細ロギングを有効化します。

```
verbose=True
```

- ※ ログファイルのパスを変更します。

```
log_dir=/var/log/nova
```

- ※ ログを中央の syslog サーバーに送信します。

```
use_syslog=True
syslog_log_facility=LOG_USER
```




注記

タイムスタンプの設定やログのフォーマットなどのオプションも利用できます。追加のログインオプションについてはコンポーネントの設定ファイルを確認してください。

2.3. リモートログインのインストールおよび設定

2.3.1. リモートログインの概要

すべてのシステムは、直面した問題やアクションを記録するログファイルを作成して更新します。多くのシステムが含まれる分散またはクラウドコンピューティング環境では、中央ロケーションでこれらのログファイルをまとめることで、デバッグを簡素化します。

rsyslog サービスにより、集中ログインサーバーを実行したり、個別システムがログファイルを集中ログインサーバーに送信するように設定したりする機能が提供されます。これは、システムの **リモートログイン** 設定と呼ばれます。

2.3.2. rsyslog サーバーのインストール

rsyslog パッケージは、集中ログインサーバーとして使用予定のシステムおよびログの送信元として設定する全システムにインストールする必要があります。これには、**root** ユーザーとしてログインして **rsyslog** パッケージをインストールします。

```
# yum install rsyslog
```

rsyslog パッケージがインストールされ、設定の準備ができました。

2.3.3. 集中ログインサーバーでの rsyslog の設定

集中ログインサーバーとして使用予定のシステム上で、以下の手順に記載するステップを行うようにしてください。また、すべてのステップは **root** ユーザーとしてログインして実行する必要があります。

1. SELinux が **rsyslog** トラフィックを許可するように設定します。

```
# semanage port -a -t syslogd_port_t -p udp 514
```

2. テキストエディターで **/etc/rsyslog.conf** ファイルを開きます。

- a. ファイルに以下の行を追加して、ログの保存先を定義します。

```
$template TmplMsg, "/var/log/%HOSTNAME%/%PROGRAMNAME%.log"
$template TmplAuth, "/var/log/%HOSTNAME%/%PROGRAMNAME%.log"

authpriv.*    ?TmplAuth
*.info,mail.none,authpriv.none,cron.none    ?TmplMsg
```

- b. ファイル内の以下の行から、行頭のコメント文字 (#) を削除します。

```

#$ModLoad imudp
#$UDPServerRun 514

```

- c. `/etc/rsyslog.conf` ファイルへの変更を保存します。

環境内の他のシステムからのログファイルを受信し、保管するための集中ログサーバーの設定が完了しました。

2.3.4. 個々のノードでの rsyslog 設定

以下の手順に記載するステップを各システムに適用して、集中ログサーバーにログを送信するように設定します。これらのステップはすべて **root** ユーザーとしてログインして実行する必要があります。

1. `/etc/rsyslog.conf` を編集して以下を追加し、集中ログサーバーのアドレスを指定します。

```
*.* @YOURSERVERADDRESS:YOURSERVERPORT
```

YOURSERVERADDRESS は集中ログサーバーのアドレスに、**YOURSERVERPORT** は、rsyslog サービスをリッスンするポートに置き換えます。以下はその例です。

```
*.* @192.168.20.254:514
```

または

```
*.* @@log-server.example.com:514
```

@ が 1 つの場合は、転送プロトコルに UDP が指定されます。TCP 転送プロトコルを指定するには、@@ を使用してください。

重要

上記の例で、ワイルドカード文字 (*) を使用している箇所は、全ログファシリティからの全ログプライオリティのログエントリをリモートの rsyslog サーバーに送信する必要があることを rsyslog に対して示しています。

より厳密なフィルターをログファイルに適用する方法についての説明は、rsyslog 設定ファイル (`rsyslog.conf`) の man ページを参照してください。このページには、`man rsyslog.conf` のコマンドを実行するとアクセスすることができます。

2. **rsyslog** サービスが起動または再起動されると、システムは全ログメッセージを集中ログサーバーに送信します。

2.3.5. rsyslog サーバーの起動

rsyslog サービスは、集中ログサーバーと、そのサーバーにログ記録を試みるシステムの両方で実行する必要があります。

以下の手順に記載するステップは **root** ユーザーとしてログインして実行する必要があります。

1. rsyslog サービスを起動します。

```
# service rsyslog start
```

2. 今後 rsyslog サービスが自動的に起動するように設定します。

```
# chkconfig rsyslog on
```

rsyslog サービスが起動されました。サービスは、ローカル設定に基づいて、ログの送受信を開始します。

第3章 TELEMETRY サービスを使用したモニタリング

Telemetry との対話には、**aodh** および **gnocchi** コマンドを使用することができます。これにより、アラーム、メトリック、測定を管理することができます。

3.1. 既存のアラームの表示

既存の Telemetry アラームを表示するには、以下のように **aodh** コマンドを使用します。

```
# aodh alarm list
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
| alarm_id           | type           |
| name               | state          | severity | enabled
|
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
| 922f899c-27c8-4c7d-a2cf-107be51ca90a |
| gnocchi_aggregation_by_resources_threshold | iops-monitor-read-requests
| insufficient data | low          | True     |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
```

リソースに割り当てた計測を表示するには、以下のようにリソースの **UUID** を指定します (インスタンス、イメージ、ボリュームなど)。

```
# gnocchi resource show 5e3fcbe2-7aab-475d-b42c-a440aa42e5ad
```

3.2. アラームの作成

aodh を使用して、しきい値に達した時点でアクティベートされるアラームを作成することができます。この例では、個別のインスタンスの CPU 使用率が 80% を超えた場合に、アラームがアクティベートされ、ログエントリが追加されます。クエリーを使用して、特定のインスタンスの ID (**94619081-abf5-4f1f-81c7-9cedaa872403**) をモニタリング目的で分離します。

```
# aodh alarm create --type gnocchi_aggregation_by_resources_threshold
--name cpu_usage_high --metric cpu_util --threshold 80 --aggregation-
method sum --resource-type instance --query '{"=": {"id": "94619081-
abf5-4f1f-81c7-9cedaa872403"}}' --alarm-action 'log://'
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
| Field           | Value
|
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
| aggregation_method | sum
|
| alarm_actions      | [u'log://']
|
```

```

| alarm_id | b794adc7-ed4f-4edb-ace4-88cbe4674a94
| comparison_operator | eq
| description |
gnocchi_aggregation_by_resources_threshold alarm rule |
| enabled | True
| evaluation_periods | 1
| granularity | 60
| insufficient_data_actions | []
| metric | cpu_util
| name | cpu_usage_high
| ok_actions | []
| project_id | 13c52c41e0e543d9841a3e761f981c20
| query | {"=": {"id": "94619081-abf5-4f1f-81c7-
9cedaa872403"}} |
| repeat_actions | False
| resource_type | instance
| severity | low
| state | insufficient data
| state_timestamp | 2016-12-09T05:18:53.326000
| threshold | 80.0
| time_constraints | []
| timestamp | 2016-12-09T05:18:53.326000
| type |
gnocchi_aggregation_by_resources_threshold |
| user_id | 32d3f2c9a234423cb52fb69d3741dbbc
+-----+-----+
-----+

```

既存のしきい値アラームを編集するには、**aodh alarm update** コマンドを使用します。アラームのしきい値を 75% に増加させるには、以下のコマンドを実行します。

```
# aodh alarm update --name cpu_usage_high --threshold 75
```

3.3. アラームの無効化または削除

アラームを無効にするには、以下のコマンドを使用します。

```
# aodh alarm update --name cpu_usage_high --enabled=false
```

アラームを削除するには、以下のコマンドを使用します。

```
# aodh alarm delete --name cpu_usage_high
```

3.4. 測定の表示

特定のリソースの測定をすべて表示するには、以下のコマンドを使用します。

```
# gnocchi measures show --resource-id UUID METER_NAME
```

タイムスタンプの範囲内における特定のリソースの測定のみを表示するには以下のコマンドを使用します。

```
# gnocchi measures show --aggregation mean --start START_TIME --end  
STOP_TIME --resource-id UUID METER_NAME
```

START_TIME および **END_TIME** は **iso-dateThh:mm:ss** の形式で指定してください。

3.5. 新規測定の作成

測定を使用して、Telemetry サービスにデータを送信することができます。以前に定義した計測と一致する必要はありません。以下に例を示します。

```
# gnocchi measures add -m 2015-01-12T17:56:23@42 --resource-id UUID  
METER_NAME
```

3.6. クラウドの使用状況の測定

以下の例では、各プロジェクトの全インスタンスの平均メモリー使用状況を表示します。

```
gnocchi measures aggregation --resource-type instance --groupby  
project_id -m memory
```

3.7. L3 キャッシュの使用状況の表示

Intel ハードウェアと libvirt バージョンが **キャッシュモニタリングテクノロジー (CMT)** をサポートする場合には、**cpu_l3_cache** の計測を使用して、インスタンスが使用する L3 キャッシュの量を監視することができます。

3.8. インスタンスのディスクアクティビティの監視

以下の例では、aodm アラームを使用して、特定のプロジェクトに含まれる全インスタンスの累積ディスクアクティビティを監視する方法を紹介します。

1. 既存のプロジェクトをレビューして、監視する必要のあるプロジェクトの適切な UUID を選択します。この例では **admin** テナントを使用します。

■

```
$ openstack project list
+-----+-----+
| ID                               | Name     |
+-----+-----+
| 745d33000ac74d30a77539f8920555e7 | admin    |
| 983739bb834a42ddb48124a38def8538 | services |
| be9e767afd4c4b7ead1417c6dfedde2b | demo     |
+-----+-----+
```

2. プロジェクトの UUID を使用して、**admin** テナント内のインスタンスが生成する読み込み要求すべての **sum()** を分析するアラームを作成します (クエリーは、**--query** パラメーターを使用してさらに制約を加えることができます)。

```
# aodh alarm create --type gnocchi_aggregation_by_resources_threshold -
-name iops-monitor-read-requests --metric disk.read.requests.rate --
threshold 42000 --aggregation-method sum --resource-type instance --
query '{"=": {"project_id": "745d33000ac74d30a77539f8920555e7"}}'
+-----+-----+
-----+
| Field                               | Value   |
|                                       |         |
+-----+-----+
-----+
| aggregation_method                 | sum     |
| alarm_actions                      | []      |
| alarm_id                          | 192aba27-d823-4ede-a404-7f6b3cc12469 |
| comparison_operator                | eq      |
| description                        |         |
gnocchi_aggregation_by_resources_threshold alarm rule |
| enabled                            | True    |
| evaluation_periods                 | 1       |
| granularity                         | 60      |
| insufficient_data_actions          | []      |
| metric                             | disk.read.requests.rate |
| name                               | iops-monitor-read-requests |
| ok_actions                         | []      |
| project_id                         | 745d33000ac74d30a77539f8920555e7 |
| query                              | {"=": {"project_id":
"745d33000ac74d30a77539f8920555e7"}} |
| repeat_actions                    | False   |
| resource_type                      | instance |
| severity                          | low     |
```

```

| state | insufficient data
| state_timestamp | 2016-11-08T23:41:22.919000
| threshold | 42000.0
| time_constraints | []
| timestamp | 2016-11-08T23:41:22.919000
| type |
| gnocchi_aggregation_by_resources_threshold |
| user_id | 8c4aea738d774967b4ef388eb41fef5e
+-----+-----+
-----+

```

3.9. リソースタイプの管理

以前にハードコードされている Telemetry リソースタイプは、**gnocchi** クライアントで管理することができます。gnocchi クライアントを使用して、リソースタイプを作成、表示、削除することができます。gnocchi API を使用して属性を更新または削除することができます。

1. 新しい **resource-type** を作成します。

```

$ gnocchi resource-type create testResource01 -a
bla:string:True:min_length=123
+-----+-----+
-----+
| Field | Value |
|-----+-----|
| attributes/bla | max_length=255, min_length=123, required=True, type=string |
| name | testResource01 |
| state | active |
+-----+-----+
-----+

```

2. **resource-type** の設定をレビューします。

```

$ gnocchi resource-type show testResource01
+-----+-----+
-----+
| Field | Value |
|-----+-----|
| attributes/bla | max_length=255, min_length=123, required=True, type=string |
| name | testResource01 |

```



```
|
| state          | active
|
+-----+-----+
-----+
```

3. **resource-type** を削除します。

```
$ gnocchi resource-type delete testResource01
```



注記

リソースが使用している場合には、リソースタイプを削除することはできません。

3.10. TIME-SERIES-DATABASE-AS-A-SERVICE の使用

Time-Series-Database-as-a-Service (gnocchi) はマルチテナントのメトリックおよびリソースのデータベースです。大規模なメトリックを格納する一方でオペレーターやユーザーにメトリックおよびリソースの情報へのアクセスを提供します。

現在、TSDaaS は認証に Identity サービスを、データの保存に Ceph と Object Storage を使用します。

TDSaaS は **statsd** プロトコルと互換性のある **statsd** デーモンを提供し、**gnocchi-statsd** と呼ばれるネットワークで送信されるメトリックをリスンすることができます。TDSaaS での **statsd** サポートを有効化するには、設定ファイルで **[statsd]** オプションを設定する必要があります。リソース ID パラメーターは、全メトリックがアタッチされる主要な一般リソース、リソースとメトリックに関連付けられるユーザーとプロジェクト ID、メトリックの作成に使用するアーカイブポリシー名として使用されます。

メトリックは **gnocchi-statsd** に送信されるため、すべてのメトリックは動的に作成され、指定の名前を設定したリソース ID にアタッチします。TSDaaS のインストールおよび設定に関する詳しい情報は、<https://access.redhat.com/documentation/ja/red-hat-enterprise-linux-openstack-platform/> から『**手動インストール手順**』の「**Time-Series-Database-as-a-サービスのインストール**」の章を参照してください。

3.10.1. Time-Series-Database-as-a-Service の実行

HTTP サーバーとメトリックデーモンを実行して、Time-Series-Database-as-a-Service (TSDaaS) を実行します。

```
# gnocchi-api
# gnocchi-metricd
```

3.10.2. WSGI アプリケーションとしての実行

mod_wsgi または他の WSGI アプリケーションなど WSGI サービスで TSDaaS を実行できます。TSDaaS で提供される **gnocchi/rest/app.wsgi** により、WSGI アプリケーションとして Gnocchi を有効化できます。

TSDaaS API 層は、WSGI を使用して実行します。つまり、Apache **httpd** および **mod_wsgi** または **uwsgi** などの別の HTTP デーモンを使用して実行できるということです。CPU の数に合わせて

プロセスやスレッド数を設定してください。通常は **1.5 × CPU の数** です。サーバーが 1 台では十分でない場合には、新たに API サーバーを起動して異なるマシン上にでも Gnocchi をスケールアウトできます。

3.10.3. metricd ワーカー

デフォルトでは **gnocchi-metricd** デーモンは、すべての CPU 電源機能をチェックしてメトリックの集計を算出する時の CPU の使用率を最大化します。**gnocchi status** コマンドを使用して HTTP API を照会し、メトリック処理のクラスターのステータスを取得します。このコマンドにより、処理するメトリック数が表示されます。これは、**gnocchi-metricd** の処理バックログとして知られています。このバックログが増え続けている限り、**gnocchi-metricd** は受信するメトリックの量を処理できるということです。処理する測定値の数が継続的に増えている場合には、**gnocchi-metricd** デーモンの数を (一時的に) 増やす必要があります。実行できる metricd デーモンやサーバー数に制約はありません。

3.10.4. Time-Series-Database-as-a-Service の監視

HTTP API の **/v1/status** エンドポイントは、処理する測定値の数 (測定値のバックログ) などさまざまな情報を返し、簡単に監視することができます。HTTP サーバーと **gnocchi-metricd** デーモンは実行中であり、ログに警告内容が書きこまれていないことを確認できると、全体的なシステムのヘルスが良好であることが分かります。

3.10.5. Time-Series-Database-as-a-Service のバックアップと復元

障害から回復できるように、インデックスとストレージの両方をバックアップする必要があります。つまり、データベースダンプ (PostgreSQL または MySQL) を作成して、データストレージ (Ceph、Swift、またはファイルシステム) のスナップショットかコピーを作成してください。復元の手順は、インデックスとストレージのバックアップを復元して、必要に応じて TSDaaS を再インストールしてから、再起動します。

第4章 トラブルシューティング

本章では、Red Hat OpenStack Platform のトラブルシューティングに役立つログ記録およびサポート情報について記載します。

4.1. サポート

クライアントコマンドが失敗した場合には、Red Hat テクニカルサポートまでご連絡ください。その際には、発生した問題についての状況説明、コンソールの全出力、およびコンソールの出力で参照されているすべてのログファイル、問題のある (可能性のある) ノードからの **sosreport** を提供してください。たとえば、コンピュートレベルで問題が発生した場合には Nova ノードで **sosreport** を実行します。ネットワークの問題の場合は、Neutron ノードでユーティリティーを実行します。一般的なデプロイメントの問題の場合は、クラウドコントローラー上で **sosreport** を実行するのがベストです。

sosreport コマンド (**sos** パッケージ) についての情報は、[「Red Hat Enterprise Linux 4.6 以降における sosreport の役割と取得方法」](#)の記事を参照してください。

ヒントについては `/var/log/messages` ファイルも確認します。

4.2. IDENTITY クライアント (KEYSTONE) の接続性における問題のトラブルシューティング

Identity クライアント (**keystone**) が Identity サービスにコンタクトできない場合には、次のようなエラーが返されます。

```
Unable to communicate with identity service: [Errno 113] No route to host. (HTTP 400)
```

この問題をデバッグするには、以下にあげる一般的な原因を確認してください。

Identity サービスが稼働していない場合

Identity サービスが **httpd.service** 内で実行されるようになりました。Identity サービスをホストしているシステム上で、サービスのステータスを確認します。

```
# systemctl status httpd.service
```

サービスがアクティブでない場合には、root ユーザーとしてログインして起動します。

```
# systemctl start httpd.service
```

ファイアウォールが適切に設定されていない場合

ファイアウォールがポート **5000** と **35357** で TCP トラフィックを許可するように設定されていない可能性があります。そのような場合の修正方法については『[手動インストール手順](#)』の「[Identity サービスのトラフィックを許可するためのファイアウォール設定](#)」を参照してください。

サービスエンドポイントが正しく定義されていない場合

Identity サービスをホストするシステムで、エンドポイントが正しく定義されているかどうかを確認します。

1. 管理トークンを取得します。

```
# grep admin_token /etc/keystone/keystone.conf
admin_token = 91f0866234a64fc299db8f26f8729488
```

2. Identity サービスの正しい管理エンドポイントを決定します。

```
http://IP:35357/VERSION
```

IP は Identity サービスをホストするシステムの IP アドレスまたはホスト名に置き換えます。VERSION は、使用中の API バージョンに置き換えます (v2.0 または v3)。

3. 事前に定義されている Identity サービス関連の環境変数の設定を解除します。

```
# unset OS_USERNAME OS_TENANT_NAME OS_PASSWORD OS_AUTH_URL
```

4. 管理トークンとエンドポイントを使用して、Identity サービスとの認証を行います。Identity サービスのエンドポイントが正しいことを確認してください。以下に例を示します。

```
# openstack endpoint list --os-
token=91f0556234a64fc299db8f26f8729488 --os-
url=https://osp.lab.local:35357/v3/ --os-identity-api-
version 3
```

一覧表示された Identity サービスの **publicurl**、**internalurl**、および **adminurl** が正しいことを確認してください。特に、各エンドポイント内にリストされている IP アドレスとポート番号が正しく、ネットワーク上で到達可能であるようにしてください。

これらの値が正しくない場合には、『手動インストール手順』の「管理者アカウントおよび Identity サービスエンドポイントの作成」の項で、正しいエンドポイントの追加方法についての説明を参照してください。正しいエンドポイントが追加されたら、誤ったエンドポイントは **openstack** コマンドの **endpoint-delete** アクションを使用して削除します。

```
# openstack endpoint delete
2d32fa6feecc49aab5de538bdf7aa018 --os-
token=91f0866234a64fc299db8f26f8729488 --os-
url=https://osp.lab.local:35357/v3/ --os-identity-api-
version 3
```

TOKEN および **ENDPOINT** は、上記のステップで特定した値に置き換えます。ID は **endpoint-list** アクションにより一覧表示される、削除対象のエンドポイントに置き換えます。

4.3. OPENSTACK NETWORKING に関する問題のトラブルシューティング

本項では、OpenStack Networking サービスに関する問題のトラブルシューティングに使用することができるさまざまなコマンドと手順について説明します。

ネットワークデバイスのデバッグ

- ※ **ip a** コマンドで、全物理/仮想デバイスを表示します。
- ※ **ovs-vsctl show** コマンドで、仮想スイッチ内のインターフェースとブリッジを表示します。
- ※ **ovs-dpctl show** コマンドで、スイッチ上のデータパスを表示します。

ネットワークパケットの追跡

- ※ **tcpdump** コマンドで、パケットが通過しない場所を確認します。

```
# tcpdump -n -i INTERFACE -e -w FILENAME
```

INTERFACE は、パケットが通過できない箇所を確認するためのネットワークインターフェース名に置き換えます。このインターフェース名には、ブリッジまたはイーサネットデバイスの名前を使用することができます。

-e フラグで、リンクレベルヘッダーがダンプされるようにします (その場合には、**vlan** タグが表示されます)。

-w フラグはオプションです。出力をファイルに書き込む場合にのみ使用することができます。使用しない場合には、その出力は標準出力 (**stdout**) に書き込まれます。

tcpdump についての詳細は、**man tcpdump** のコマンドで man ページを開いて参照してください。

ネットワーク名前空間のデバッグ

- ※ **ip netns list** コマンドで、既知のネットワーク名前空間をすべて一覧表示します。
- ※ **ip netns exec** コマンドで、特定の名称空間内のルーティングテーブルを表示します。

```
# ip netns exec NAMESPACE_ID bash
# route -n
```

bash シェルで **ip netns exec** コマンドを起動し、それ以降に実行するコマンドが **ip netns exec** コマンドを実行しなくても呼び出されるようにします。

4.4. ダッシュボードでのネットワークまたはルータータブの表示に関するトラブルシューティング

Networks および **Routers** のタブは、OpenStack Networking を使用するよう環境が設定されている場合にのみ表示されます。現在、デフォルトでは、Packstack ユーティリティによって Nova ネットワークがデプロイされるため、この方法でデプロイされた環境には、これらのタブは表示されない点に特に注意してください。

OpenStack Networking が環境にデプロイされているにもかかわらずタブが表示されない場合には、Identity サービスでサービスエンドポイントが正しく定義されて、ファイアウォールがそのエンドポイントへのアクセスを許可し、サービスが稼働していることを確認してください。

4.5. DASHBOARD でのインスタンス起動エラーに関するトラブルシューティング

ノ ュ ノ 1 ノ ノ

ダッシュボードを使用したインスタンス起動時に操作が失敗した場合には、汎用の **ERROR** メッセージが表示されます。実際の原因を究明するには、コマンドラインツールを使用する必要があります。

nova list でインスタンスの一意識別子を確認します。次にその識別子を **nova show** コマンドの引数として使用します。返される項目の 1 つがエラー条件となります。最も一般的な値は **NoValidHost** です。

このエラーは、インスタンスをホストするのに十分なリソースが利用できる有効なホストがないことを示しています。この問題を回避するには、より小さなインスタンスサイズを選択するか、その環境のオーバーコミットの上限を高くする方法を検討してください。



注記

インスタンスをホストするには、コンピュータノードで CPU および RAM リソースが使用可能なだけでなく、インスタンスに関連付けられる一時ストレージ用に十分なディスク領域がある必要もあります。

4.6. DASHBOARD の KEYSTONE V3 認証のトラブルシューティング

`django_openstack_auth` は、Django の `contrib.auth` フレームワークと連携する、プラグ可能な Django 認証バックエンドで、OpenStack Identity サービス API に対してユーザー認証を行います。`django_openstack_auth` は、トークンオブジェクトを使用して、ユーザーおよび Keystone 関連の情報をカプセル化し、Dashboard は、トークンオブジェクトを使用して Django ユーザーオブジェクトを再構築します。

現在、トークンオブジェクトは以下を格納します。

- ※ keystone トークン
- ※ ユーザー情報
- ※ 範囲
- ※ ロール
- ※ サービスカタログ

Dashboard は、ユーザーセッションデータの処理に Django のセッションフレームワークを使用します。以下は、利用可能な各種セッションバックエンド一覧です。これらは、`local_settings.py` ファイルの `SESSION_ENGINE` 設定で制御されます。

- ※ ローカルメモリーキャッシュ
- ※ Memcached
- ※ データベース
- ※ キャッシュされたデータベース
- ※ クッキー

特に署名付きクッキーのセッションバックエンドが使用されている場合、多数またはすべてのサービスが一度に有効化された場合など、クッキーのサイズが制限に到達して、Dashboard へのログインに失敗する可能性があります。クッキーサイズが増加する理由の1つとして、サービスカタログが挙げられます。多くのサービスが登録されるにつれ、サービスカタログのサイズも増加します。

このようなシナリオでは (特に keystone v3 認証を使用している場合)、セッショントークン管理を向上するため、Dashboard へログインするための以下の設定を含めてください。

1. `/usr/share/openstack-dashboard/openstack_dashboard/settings.py` では、以下の設定を追加します。

```
DATABASES =
{
  'default':
  {
    'ENGINE': 'django.db.backends.mysql',
    'NAME': 'horizondb',
    'USER': 'User Name',
    'PASSWORD': 'Password',
    'HOST': 'localhost',
  }
}
```

2. 同じファイルで、`SESSION_ENGINE` を以下に変更します。

```
SESSION_ENGINE = 'django.contrib.sessions.backends.cached_db'
```

3. `mysql` コマンドを使用してデータベースサービスに接続します。USER は、接続に使用するユーザー名に置き換えます。また、USER は root ユーザー (または正しいパーミッション「create db」を持つユーザー) でなければなりません。

```
# mysql -u USER -p
```

4. Horizon データベースを作成します。

```
mysql > create database horizondb;
```

5. `mysql` クライアントを終了します。

```
mysql > exit
```

6. 以下のコマンドで、`openstack_dashboard` ディレクトリーに移動して、データベースを同期します。

```
# cd /usr/share/openstack-dashboard/openstack_dashboard
$ ./manage.py syncdb
```

スーパーユーザーを作成する必要はないため、質問には「n」と回答します。

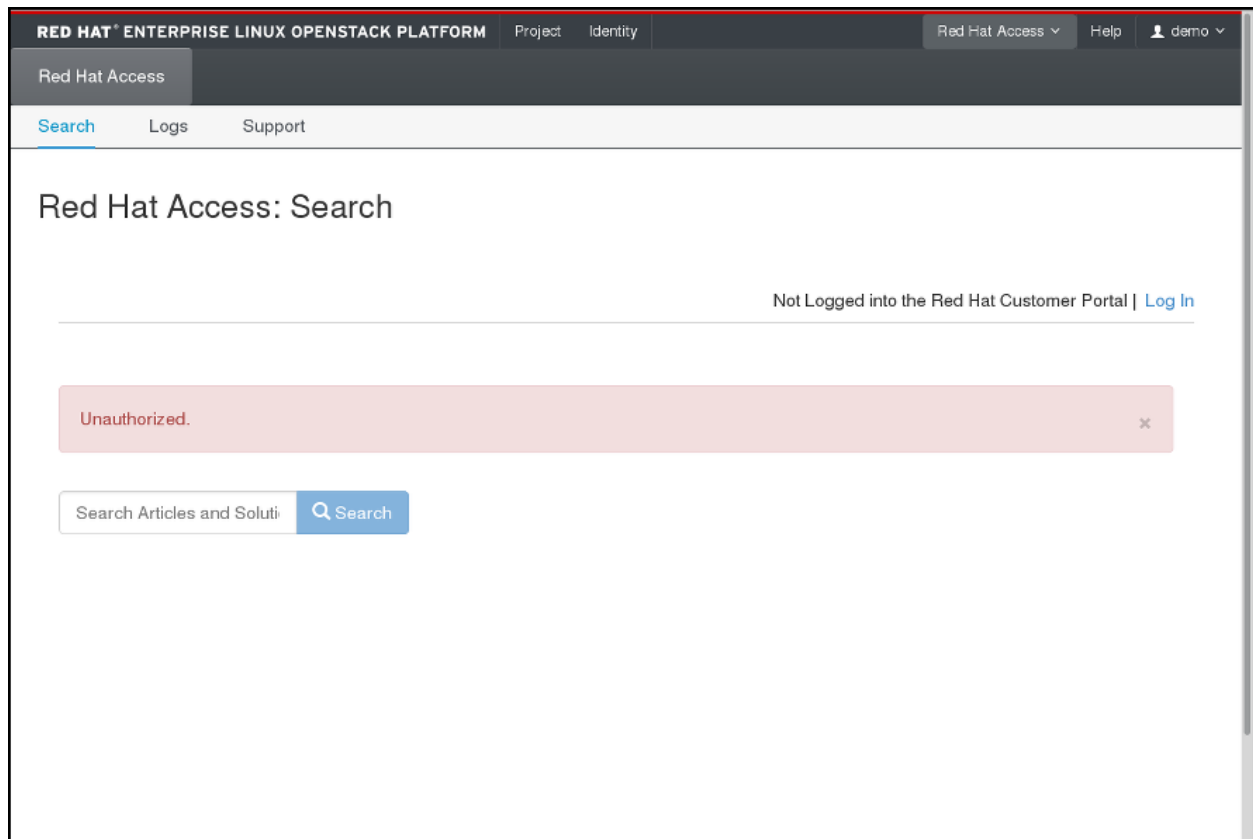
7. Apache http サーバーを再起動します。Red Hat Enterprise Linux の場合は以下を実行します。

```
#service httpd restart
```

4.7. OPENSTACK DASHBOARD: RED HAT ACCESS タブ

Red Hat Access タブ (OpenStack Dashboard の一部) では、Red Hat カスタマーポータルから記事やソリューションの検索、確認、インスタンスからのログ表示や診断、カスタマーサポートケースの対応ができます。

図4.1 Red Hat Access タブ



重要

Red Hat Access タブの機能を使用するには、ブラウザで Red Hat カスタマーポータルにログインする必要があります。

ログインされていない場合には、以下の手順でログインしてください。

1. **ログイン** をクリックします。
2. Red Hat のログイン情報を入力します。
3. Red Hat パスワードを入力します。
4. **サインイン** をクリックします。

フォームは以下のとおりです。

図4.2 Red Hat カスタマーポータルへのログイン

Sign into the Red Hat Customer Portal

Red Hat Access makes it easy for you to self-solve issues, diagnose problems, and engage with us via the Red Hat Customer Portal. To access Red Hat Customer Portal resources, you must enter valid portal credentials.

Red Hat Login

Password

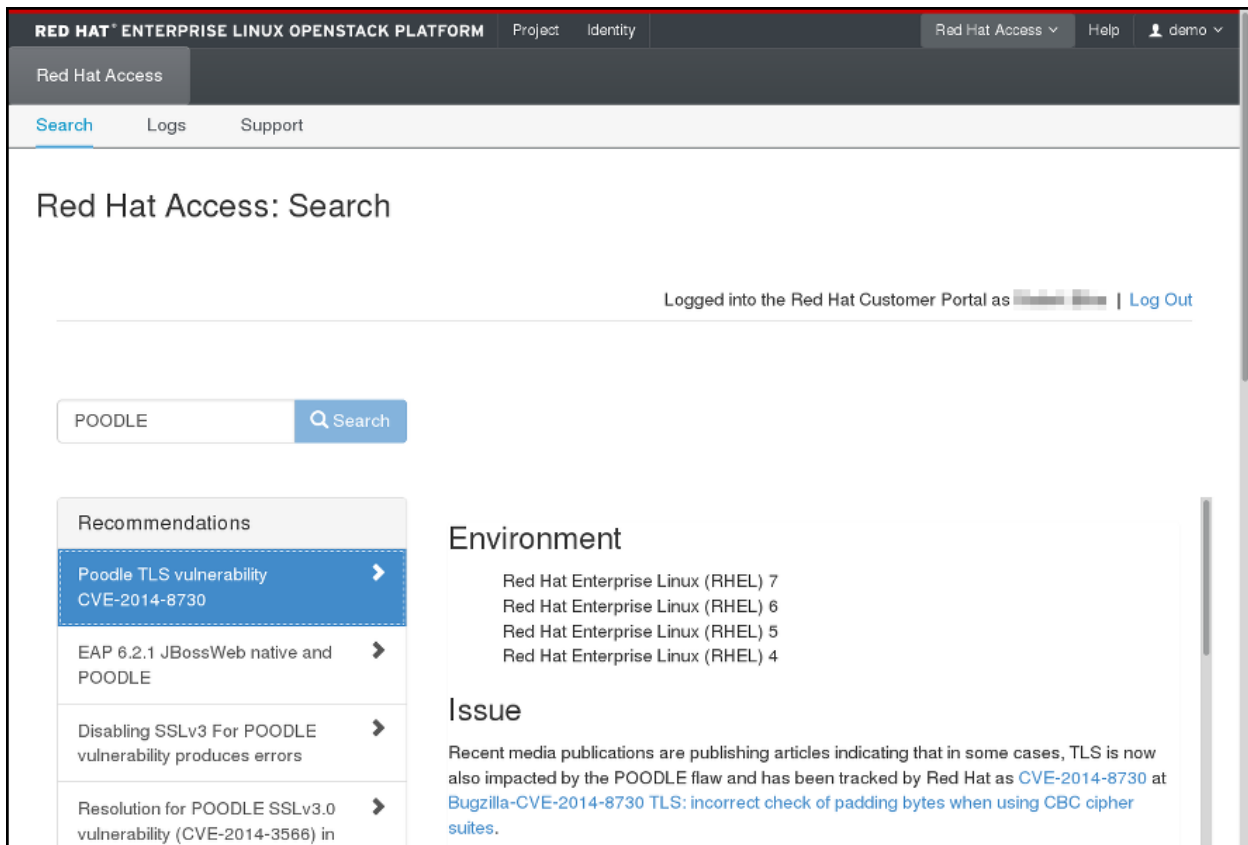
Note: Red Hat Customer Portal credentials differ from the credentials used to log into this product.

この時点でログインしないと、認証が必要な機能の1つを使用する際に Red Hat ログインとパスワードが要求されます。

4.7.1. 検索

1つまたは複数の検索キーワードを入力して、Red Hat カスタマーポータルからの記事やソリューションを検索できます。関連の記事やソリューションのタイトルが表示されます。タイトルをクリックして、指定の記事またはソリューションを表示します。

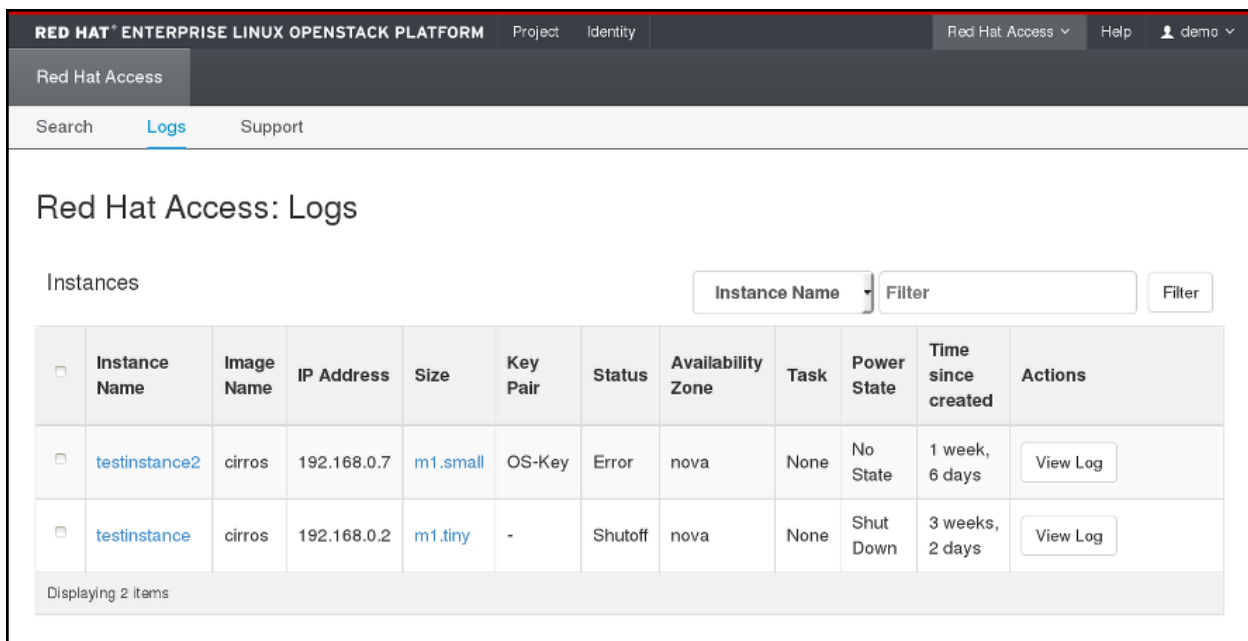
図4.3 Red Hat Access タブの検索結果の例



4.7.2. ログ

ここから、OpenStack インスタンスからのログを確認することができます。

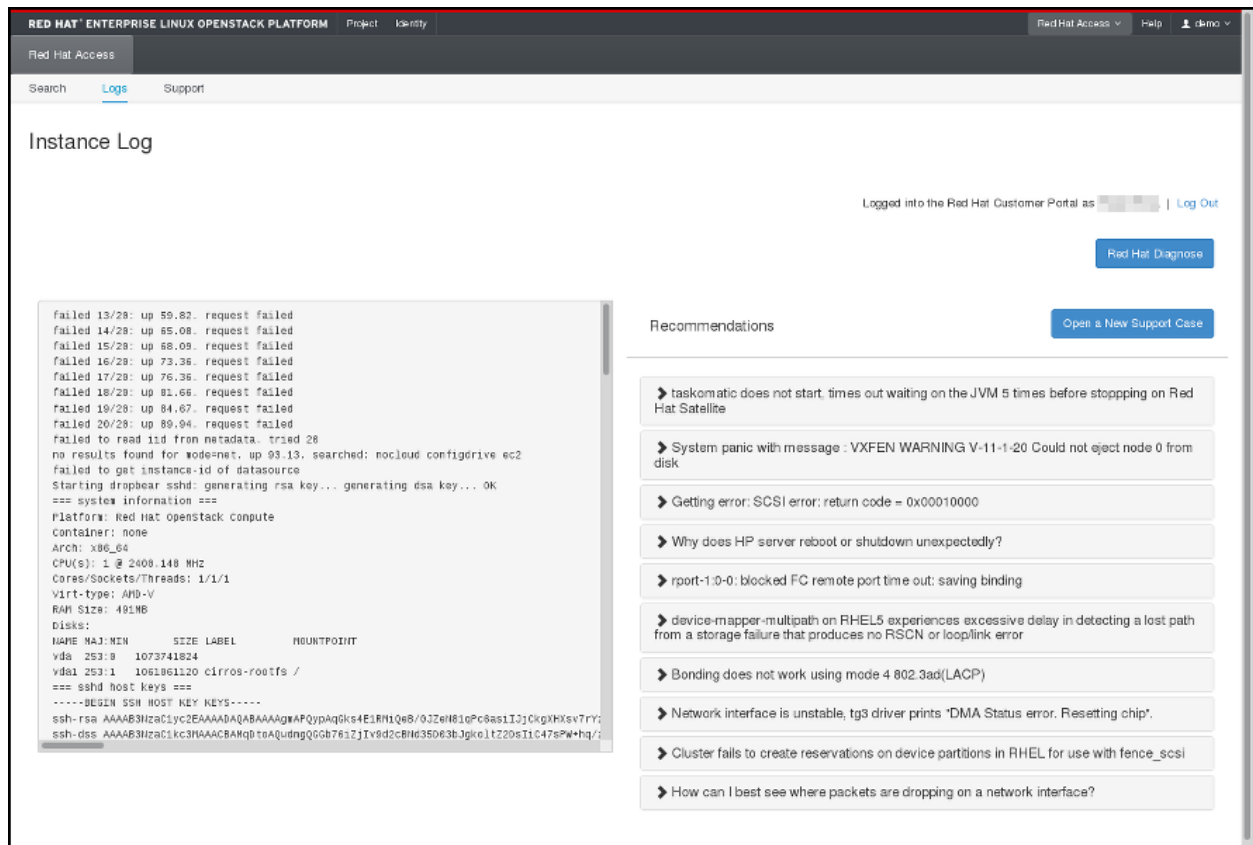
図4.4 Red Hat Access タブでのインスタンスのログ



表から希望のインスタンスを検索します。多数のインスタンスがある場合には、名前、状態、イメージ ID、フレーバー ID 別にフィルタリングすることができます。チェックするインスタンスの **アクション** の欄で **ログの表示** をクリックします。

インスタンスのログが表示されたら、**Red Hat 診断** をクリックして、コンテンツに関連した提案ソリューションを取得することができます。

図4.5 Red Hat Access タブでのインスタンスのログ

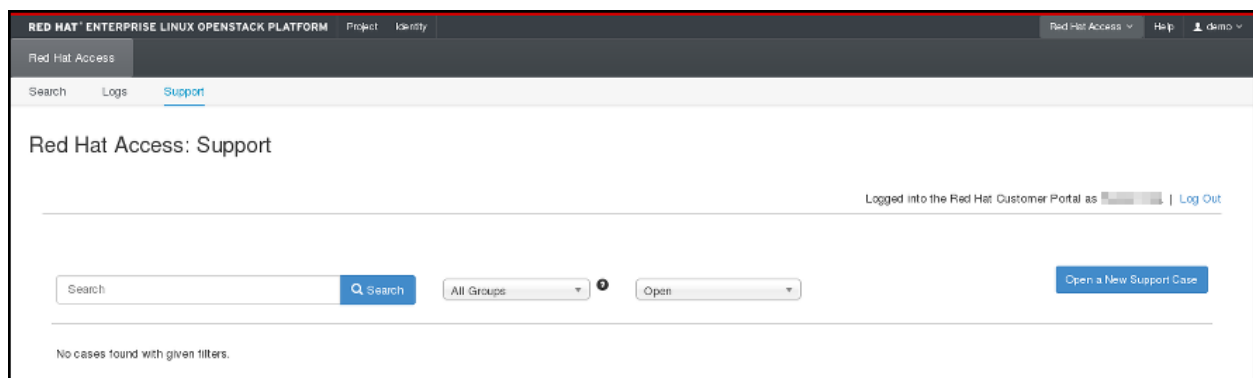


提案されたソリューションで役に立つものがない場合や、問題が正しくロギングされていない場合には、サポートケースを新規作成 をクリックして、問題を Red Hat サポートに報告してください。

4.7.3. サポート

Red Hat Access タブの最後のオプションでは、Red Hat カスタマーポータルをサポートケースを検索することができます。

図4.6 サポートケースの検索



また、適切なボタンをクリックして、以下のページでフォームに入力して新規サポートケースを開くことも可能です。

図4.7 サポートケースの新規作成

