

# Qlik Sense Enterprise on Kubernetes の Minikube 上への導入

Qlik Sense Enterprise マルチクラウド演習

1.	CentOSへのログイン	2
2.	Kubectlでのクラスタの確認	2
3.	Helmの導入と設定	2
4.	Qlik Sense Enterprise on Kubernetesの導入(開発モードを使った基本設定)	3
5.	Qlik Sense Enterprise on Kubernetesの導入(永続ストレージを使った設定)	8

## 1. CentOS へのログイン

- 1) この文書では以降の導入作業を root ユーザーで実施します。Windows ホスト上で PowerShell を開き、コンソールから以下で SSH アクセスを行います。パスワードには「osboxes.org」を入力します。

```
ssh root@elastic.example
```

## 2. Kubectl でのクラスタの確認

- 1) Minikube のステータスを確認します。

```
minikube status
```

Status が Stopped となっている場合には以下を実行して Minikube を実行し、再度 minikube status コマンドを実行してステータスが Running となっていることを確認します。

```
minikube start --memory 6144 --cpus=2 --vm-driver=none
```

- 2) 以下を実行し、minikube と表示されて kubectl が既に作成した minikube のクラスタを参照していることを確認します。

```
kubectl config current-context
```

- 3) 以下のコマンドで利用可能なクラスタを表示されることが可能です。尚、クラスタの参照先を変更するには「kubectl config set-cluster <クラスタ名>」を利用します。

```
kubectl config get-clusters
```

## 3. Helm の導入と設定

- ※ 手順詳細は以下のサイトが参考となります。ここでは最低限必要な手順をご説明します。

<https://github.com/helm/helm/blob/master/docs/install.md>

- 1) 以下を実行して Helm をインストールします。

```
curl -LO https://git.io/get_helm.sh
chmod 700 get_helm.sh
./get_helm.sh
```

- 2) Helm を利用するには Helm クライアントからの要求を処理する Tiller Pod を立ち上げる必要があります。まず Service Account を作成します。

```
kubectl create serviceaccount --namespace kube-system tiller
```

3) Tiller を動作させる Service Account に権限を付与します。

```
kubectl create clusterrolebinding tiller-cluster-rule --clusterrole=cluster-admin --serviceaccount=kube-system:tiller
```

Helm クライアントの初期化を Tiller のデプロイを行います。

```
helm init --upgrade --wait
```

4) Tiller Pod の Service Account の設定を行います。

```
kubectl patch deploy --namespace kube-system tiller-deploy -p '{"spec":{"template":{"spec":{"serviceAccount":"tiller"}}}}'
```

5) 以下を実行し、Tiller Pod が起動していることを確認します。

```
kubectl get pods -n kube-system | egrep "NAME|tiller"
```

```
[root@minikube ~]# kubectl get pods -n kube-system | egrep "NAME|tiller"
NAME                                READY   STATUS    RESTARTS   AGE
tiller-deploy-8557598fbc-mh6t6      1/1     Running   0           4m19s
```

6) Qlik の Helm チャートレポジトリを追加します。

```
helm repo add qlik https://qlik.bintray.com/stable
```

7) 以下を実行してレポジトリの一覧を確認します。

```
helm repo list
```

```
[root@minikube ~]# helm repo list
NAME    URL
stable  https://kubernetes-charts.storage.googleapis.com
local   http://127.0.0.1:8879/charts
qlik    https://qlik.bintray.com/stable
```

## 4. Qlik Sense Enterprise on Kubernetes の導入(開発モードを使った基本設定)

この項では、主にデモ環境での利用を目的とした IdP や永続ストレージへのデータ(アプリなどのファイルやレポジトリ DB 内のデータ)の保存を行わない簡易な環境の導入を行います。

1) エディタを利用して以下の内容の values.yaml ファイルを作成します。

```
#This setting enables dev mode to include a local MongoDB install
devMode:
  enabled: true
```

```

#This setting accepts the EULA for the product
engine:
  acceptEULA: "yes"

# MINIKUBE SPECIFIC SETTINGS (dont not use with other K8
providers)_____
elastic-infra:
nginx-ingress:
  controller:
    service:
      type: NodePort
      nodePorts:
        https: 32443
      extraArgs.report-node-internal-ip-address: ""

hub:
  ingress:
    annotations:
      nginx.ingress.kubernetes.io/auth-signin:
https://$host:32443/login?returnto=$request_uri

management-console:
  ingress:
    annotations:
      nginx.ingress.kubernetes.io/auth-signin:
https://$host:32443/login?returnto=$request_uri

edge-auth:
  oidc:
    redirectUri: https://elastic.example:32443/login/callback

```

2) 以下を実行します。

```
helm upgrade --install qliksense-init qlik/qliksense-init
```

3) 以下を実行して Qlik Sense Enterprise on Kubernetes を展開します。

```
helm upgrade --install qliksense qlik/qliksense -f values.yaml
```

- 4) 以下を実行し、展開した Pod の状況を確認します。STATUS が ContainerCreating から全て Running となり、READY で全てのコンテナが立ち上がるまで待ちます。

```
kubectl get pods
```

以下のコマンドを実行すると定期的な繰り返し処理で上記のコマンドを実行することも可能です。

```
while [ 1 ]; do kubectl get pods; sleep 5; clear; done
```

```

NAME                                READY   STATUS    RESTARTS   AGE
aliksense-audit-9996f4b76-zsp5z     1/1    Running   8           22m
aliksense-chronos-88b87fbf-dwtf      2/2    Running   0           22m
aliksense-chronos-worker-55cc9dc57-6jcpv  1/1    Running   6           22m
aliksense-collections-84bd94cbd-zd8dg  1/1    Running   0           22m
aliksense-data-connector-odbc-cmd-6887674dc6-7dvbm  1/1    Running   0           22m
aliksense-data-connector-odbc-rlid-86bf45cff8-wd6gb  1/1    Running   0           22m
aliksense-data-connector-qwc-cmd-567f88f8bc-sx58m  1/1    Running   7           22m
aliksense-data-connector-qwc-rlid-576679cc58-bhjrj  1/1    Running   7           22m
aliksense-data-connector-rest-cmd-7f8ddb7ccc-k7f5t  1/1    Running   0           22m
aliksense-data-connector-rest-rlid-788fccc4b59-5rmzb  1/1    Running   0           22m
aliksense-data-prep-5ccbc8664-stanw  1/1    Running   0           22m
aliksense-data-rest-source-7f494f849f-qrkxm  1/1    Running   0           22m
aliksense-dcaas-8ffbfdfcb-8vs87      1/1    Running   0           22m
aliksense-dcaas-redis-master-0       1/1    Running   0           22m
aliksense-dcaas-redis-metrics-7994787d44-bl9ha  1/1    Running   0           22m
aliksense-dcaas-web-59b886cb7b-5jrnk  1/1    Running   0           22m
aliksense-edge-auth-78f4f76cb8-h9nxt  2/2    Running   1           22m
aliksense-encryption-7f57ffc656-lfdqz  1/1    Running   0           22m
aliksense-engine-67f85d8c85-f2ba8    1/1    Running   8           22m
aliksense-feature-flags-5584949df7-j9mgw  1/1    Running   0           22m
aliksense-groups-df8baf476-kpvxj     1/1    Running   5           22m
aliksense-hub-d7984454c-8l8rs        1/1    Running   0           22m
aliksense-identity-providers-569c8c8d4c-cwsjd  1/1    Running   0           22m
aliksense-insights-6d7c588b47-a8pvl  1/1    Running   0           22m
aliksense-keys-59c554955d-vf28j     1/1    Running   0           22m
aliksense-licenses-7d8dcbd575-qh6nz  2/2    Running   0           22m
aliksense-locale-5f9dcb8cf9-knqhx    1/1    Running   0           22m
aliksense-management-console-5944d84748-p9mrv  1/1    Running   0           22m
aliksense-mongodb-64c94bf67f-lwgd4   1/1    Running   0           22m
aliksense-nats-0                      2/2    Running   0           22m
aliksense-nats-streaming-0           2/2    Running   3           22m
aliksense-nats-streaming-1           2/2    Running   0           4m42s
aliksense-nats-streaming-2           2/2    Running   0           4m7s
aliksense-nginx-ingress-controller-86cdf9b898-kfghs  1/1    Running   0           22m
aliksense-odag-78d89f5c4-g2sjt       2/2    Running   0           22m
aliksense-policy-decisions-5d5876f685-ksv7m  1/1    Running   0           22m
aliksense-precedents-846484c4fd-cx6ff  1/1    Running   0           22m
aliksense-qix-data-connection-804bc88bdb-w6zpl  1/1    Running   0           22m
aliksense-qix-datafiles-967fcb88c-jaws2  1/1    Running   0           22m
aliksense-qix-sessions-5df58784b-vmj7x  1/1    Running   0           22m
aliksense-redis-master-0             1/1    Running   0           22m
aliksense-redis-slave-8589c98556-bmp7l  1/1    Running   1           22m
aliksense-reload-task-6dbcf9df9c-5xbtx  1/1    Running   0           22m
aliksense-reloads-56ff4b87bd-6ahw5   1/1    Running   7           22m
aliksense-reporting-6687fd5bcf-jb17z  4/4    Running   0           22m
aliksense-resource-library-54754b7fb-bv7dd  1/1    Running   0           22m
aliksense-sense-client-5945fc77df-w25hg  1/1    Running   0           22m
aliksense-spaces-665f976886-lhg84     1/1    Running   0           22m
aliksense-temporary-contents-586667c5fc-fv5hr  1/1    Running   0           22m
aliksense-tenants-577c46c55f-tqlmq   1/1    Running   0           22m
aliksense-users-7446657456-cbfkg     1/1    Running   0           22m

```

主なサービス(Pod)の説明は以下となります。

サービス	説明
Chronos	スケジューラ バック エンド サービス。
Cloud hub	Qlik Cloud Services と Qlik Sense Enterprise on Kubernetes のユーザーに対するハブの機能を処理します。
Collections	ハブに送られたコンテンツを編成し、構造化します。また、アクセス制御ルールにも適用されます。
datafiles	アプリのロード中にアクセス可能なデータ ファイルのアップロード/管理をユーザーに許可します。
edge-auth	外部 ID プロバイダーと連携して、展開へのログイン時にユーザーを認証するサービス。内部リソースへの安全なアクセスを認証するチケットも管理します。
elastic-infra	MongoDB、Redis、Traefik、nginx-ingress といった非 Qlik サービスを含んだコレクション。MongoDB、Redis、nginx-ingress、Helm パッケージ マネージャーを使って、Kubernetes クラスター上で elastic-infra 展開を起動します。クラウド環境内で必要なすべてのコンポーネントと機能を接続するための基本リソースを起動します。
Engine	すべてのアプリケーションの計算とロジックを処理します。
Feature Flags	高度なシナリオにおいて機能のオンとオフを切り替えます。
Insights	シートの [共有] ダイアログを支援する機能に関する REST サービス。このサービスは、共有リソースへの持続的な固定リンクの生成、追跡、および供給を行うことで、Qlik Sense 洞察の共有を担います。固定リンクは、さまざまなソーシャル メディアで共有できます。
Licenses	ライセンス サービスは、Qlik Cloud Services と Qlik Sense Enterprise on Kubernetes でユーザー ライセンスを供与するために使用されます。
Locale	クライアントでのユーザー ロケールの選択を処理します。
Mira	展開内のエンジン、その現在の正常性、アプリケーションの可用性について検出サービスを提供します。
ODAG	Qlik Sense アプリの On-demand アプリ生成用のサービス。
Policy Decision	Qlik オブジェクト (アプリなど) に対して ABAC セキュリティ評価を実行する Qlik Cloud Services および Qlik Sense Enterprise on Kubernetes で一連のルールを処理します。Rules Service と呼ばれることもあります。ルール エンジン用に REST API を、ルールベースのポリシー用に管理 API を使用し、QRS ルール エンジン、Policy Decision Service を置き換えます。
qix-data-connection	接続管理に関するエンジン リクエストを処理します。
qix-sessions	ユーザー セッション トラフィックをエンジン サービスにルーティングします。
Reporting	データおよびチャート画像を含むレポートの生成を実行します。
Resource Library	テーマや拡張機能などのコンテンツをサポートするための、汎用リソース ストレージ サービス。
Tenant	テナント (ユーザー) 情報を保存し、返すために使用されます。
temporary-contents	一時的に使用可能になっているリソースを管理します。
User	ユーザー情報を管理及び取得します。
Sense Client	Qlik Sense Enterprise では開発者により、QCS および Qlik Sense Enterprise on Kubernetes では消費者によって実行される、Qlik Sense クライアントのデスクトップおよび Web ブラウザー インスタンス。

5) Windows のホストコンピューター上でブラウザを開き、以下の URL にアクセスします。

<https://elastic.example:32443/>

6) 以下のような警告画面が表示されますので、そのままアクセスを行います。

この接続ではプライバシーが保護されません

elastic.example では、悪意のあるユーザーによって、パスワード、メッセージ、クレジットカードなどの情報が盗まれる可能性があります。詳細

NET::ERR\_CERT\_AUTHORITY\_INVALID

一部のシステム情報とページのコンテンツを Google に送信して、セーフ ブラウジングの改善にご協力ください。プライバシー ポリシー

[詳細情報を表示しない](#) [セキュリティで保護されたページに戻る](#)

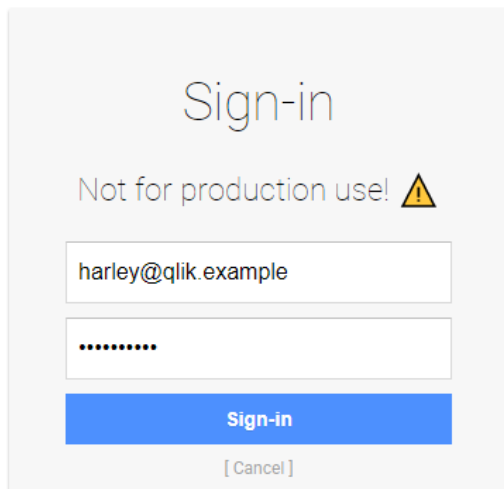
このサーバーが elastic.example であることを確認できませんでした。このサーバーのセキュリティ証明書は、ご使用のパソコンのオペレーティング システムによって信頼されているものではありません。原因としては、不適切な設定や、悪意のあるユーザーによる接続妨害が考えられます。

elastic.example にアクセスする (安全ではありません)

7) サインインの画面が表示されますので、以下のユーザーID・パスワードを使ってログインします。(ここでは外部 IdP の設定を行わず、内部に含まれている簡易なテスト用の OIDC を使った認証を行っています。)

ユーザーID: harley@qlik.example

パスワード: Password1!

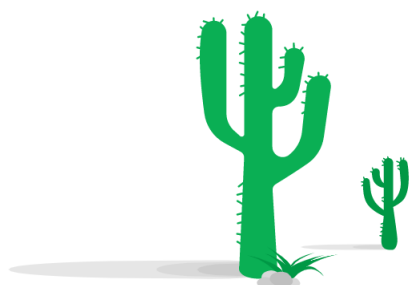
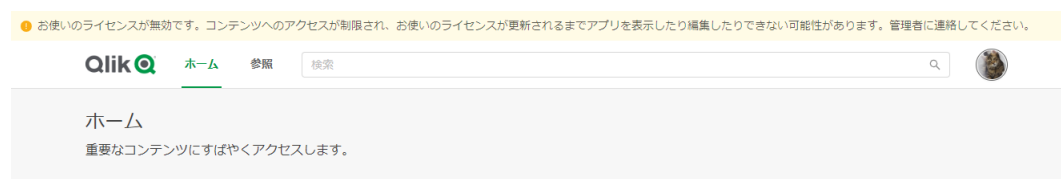
A sign-in dialog box with a light gray background. At the top, it says "Sign-in" in a large font. Below that, it says "Not for production use!" followed by a yellow warning triangle icon. There are two input fields: the first contains the email address "harley@qlik.example" and the second contains a series of dots representing a password. Below the input fields are two buttons: a blue "Sign-in" button and a smaller "[ Cancel ]" button.

※ デフォルトで以下のようなテストユーザーが登録されています。

メールアドレス	パスワード	グループ
harley@qlik.example	Password1!	'Everyone', 'Sales'
barb@qlik.example	Password1!	'Everyone', 'Support'
quinn@qlik.example	Password1!	'Everyone', 'Accounting'
sim@qlik.example	Password1!	'Everyone', 'Accounting'
phillie@qlik.example	Password1!	'Everyone', 'Marketing', 'Sales'
peta@qlik.example	Password1!	'Everyone', 'Engineering'
marne@qlik.example	Password1!	'Everyone', 'Marketing'
sibylla@qlik.example	Password1!	'Everyone', 'Accounting'
evan@qlik.example	Password1!	'Everyone', 'Engineering'
franklin@qlik.example	Password1!	'Everyone', 'Sales'



- 8) 以下のようなハブのトップ画面が表示されます。この状態では IdP を使わず、テスト利用目的に内部的に用意されたユーザーを使ってログオンしている状況となります。



こちらは空白になっています。

こちらに表示できるものはまだありません。アプリの表示が許可されているか、またアプリを利用できるかどうか、管理者に確認してください。

この環境でライセンスを適用してそのまま利用するには「04\_Qlik Sense Enterprise on Kubernetes 上の管理コンソールでのライセンス有効化」の説明に従ってライセンスの適用を行います。ただし、ライセンスを適用したユーザーが RootAdmin としての管理者権限が付与されますので、一旦ライセンスを適用後に認証を IdP での認証に切り替えると、この RootAdmin のユーザーとしてログオンが出来なくなってしまいますので、チュートリアルをそのまま進めるにはライセンスの適用を行わずにそのまま進んでください。

## 5. Qlik Sense Enterprise on Kubernetes の導入(永続ストレージを使った設定)

- 1) 既に前章の「4. Qlik Sense Enterprise on Kubernetes の導入(開発モードを使った基本設定)」で環境を構築済みの場合には、次の演習に進む前に一旦展開した Qlik Sense Enterprise on Kubernetes を削除します。まず、展開されている helm のリリースを確認します。

```
helm list
```

- 2) リリースを削除します。

```
helm del --purge qliksense
```

- 3) NFS 用のディレクトリを作成します。

```
mkdir /var/nfs/qlik -p  
chmod 777 /var/nfs/qlik
```

- 4) 作成したディレクトリを NFS 用に Export します。エディタを利用して「/etc/exports」ファイル内に以下の一行を追加します。

```
/var/nfs/qlik 192.168.56.0/24(rw, sync, no_subtree_check)
```

- 5) NFS サーバー関連のサービスを実行します。

```
sudo systemctl start rpcbind
sudo systemctl start nfs-server
sudo systemctl start nfs-lock
sudo systemctl start nfs-idmap
sudo systemctl enable nfs-server
```

- 6) 以下を実行し、NFS が Export されたことを確認します。(追加されていない場合には `exportfs -a` を実行します。)

```
exportfs -v
```

```
[root@minikube ~]# exportfs -v
/var/nfs/qlik 192.168.56.0/24(sync, wdelay, hide, no_subtree_check, sec=sys, rw, secure, root_squash, no_all_squash)
```

- 7) NFS のマウントを確認します。マウント先のフォルダを作成します。

```
mkdir -p /nfs/qlik
```

- 8) マウントを実行し、マウントできたことを確認します。

```
mount -t nfs 192.168.56.100:/var/nfs/qlik /nfs/qlik
mount | grep /var/nfs/qlik
```

- 9) NFS Provisioner を作成します。

```
helm install stable/nfs-client-provisioner --set nfs.server=192.168.56.100 --set
nfs.path=/var/nfs/qlik --name nfs-client
kubectl patch storageclass nfs-client -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

- 10) NFS Provisioner の Storage Class が作成されたことを確認します。

```
kubectl get sc
```

```
[root@minikube ~]# kubectl get sc
NAME                                PROVISIONER                                AGE
nfs-client (default)                cluster.local/nfs-client-nfs-client-provisioner 3m22s
standard (default)                  k8s.io/minikube-hostpath                    20h
```

11) エディタを利用して以下の内容の values-ps.yaml ファイルを作成します。

```
#This setting enables dev mode to include a local MongoDB install
devMode:
  enabled: true

#This setting accepts the EULA for the product
engine:
  acceptEULA: "yes"

#These setting specifies the storage for the services, in this case, NFS
global:
  persistence:
    storageClass: nfs-client

#MongoDB is set to persist with NFS storage so that data will not be lost if the
container is destroyed
mongodb:
  persistence:
    enabled: true
    accessMode: ReadWriteMany
    storageClass: nfs-client

# MINIKUBE SPECIFIC SETTINGS (dont not use with other K8
providers)_____
elastic-infra:
  nginx-ingress:
    controller:
      service:
        type: NodePort
        nodePorts:
          https: 32443
        extraArgs.report-node-internal-ip-address: ""

hub:
  ingress:
    annotations:
      nginx.ingress.kubernetes.io/auth-signin:
https://$host:32443/login?returnto=$request_uri
```

```
management-console:
ingress:
  annotations:
    nginx.ingress.kubernetes.io/auth-signin:
      https://$host:32443/login?returnto=$request_uri

edge-auth:
  oidc:
    redirectUri: https://elastic.example:32443/login/callback
```

12) 以下を実行します。

```
helm upgrade --install qliksense-init qlik/qliksense-init
```

13) インストールを実行します。

```
helm upgrade --install qliksense qlik/qliksense -f values-ps.yaml
```

14) 以下を実行し、展開した Pod の状況を確認します。STATUS が ContainerCreating から全て Running となるまで待ちます。

```
kubectl get pods
```

以下のコマンドを実行すると定期的な繰り返し処理で上記のコマンドを実行することも可能です。

```
while [ 1 ]; do kubectl get pods; sleep 5; clear; done
```

15) 全ての Pod が Running となったのを確認後、以下のコマンドで作成された Persistent Volume、Persistent Volume Claim を確認することができます。NFS Provisioner により動的にプロビジョニングが行われて作成されていることが確認でき、これらに Qlik Sense の永続化が必要なデータが格納されています。

```
kubectl get pv
```

```
kubectl get pvc
```

```
[root@minikube ~]# kubectl get pv
NAME                                     CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM                                     STORAGECLASS  REASON  AGE
pvc-09186597-8053-4355-8818-808889252988  8Gi       RWO           Delete          Bound   default/redis-data-qliksense-redis-master-0  standard   130m
pvc-219ba005-a60f-49a3-b416-7fce3392ac73    5Gi       RWX           Delete          Bound   default/qliksense-qix-datafiles             nfs-client  14m
pvc-65c3fa19-3db9-4899-af97-d680ba98befb     8Gi       RWO           Delete          Bound   default/qliksense-mongodb                   nfs-client  14m
pvc-c09fb91e-42ca-4d8b-b97b-27b51dc6969    5Gi       RWX           Delete          Bound   default/qliksense-resource-library          nfs-client  14m
pvc-f969d9da-6951-41d4-962e-7989a7e09c82    5Gi       RWX           Delete          Bound   default/qliksense-engine                     nfs-client  14m
pvc-fb07820d-f80a-4206-bd7c-c89a853ef25d    8Gi       RWO           Delete          Bound   default/redis-data-qliksense-dcaas-redis-master-0  standard   130m
pvc-fb7531b3-b30a-40a9-9597-2997a44dc931    5Gi       RWX           Delete          Bound   default/qliksense-temporary-contents        nfs-client  14m
[root@minikube ~]# kubectl get pvc
NAME                                     STATUS  VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS  AGE
qliksense-engine                         Bound   pvc-f969d9da-6951-41d4-962e-7989a7e09c82  5Gi       RWX           nfs-client    14m
qliksense-mongodb                         Bound   pvc-65c3fa19-3db9-4899-af97-d680ba98befb  8Gi       RWO           nfs-client    14m
qliksense-qix-datafiles                   Bound   pvc-219ba005-a60f-49a3-b416-7fce3392ac73  5Gi       RWX           nfs-client    14m
qliksense-resource-library                Bound   pvc-c09fb91e-42ca-4d8b-b97b-27b51dc6969  5Gi       RWX           nfs-client    14m
qliksense-temporary-contents              Bound   pvc-fb7531b3-b30a-40a9-9597-2997a44dc931  5Gi       RWX           nfs-client    14m
redis-data-qliksense-dcaas-redis-master-0 Bound   pvc-fb07820d-f80a-4206-bd7c-c89a853ef25d  8Gi       RWO           standard      130m
redis-data-qliksense-redis-master-0      Bound   pvc-09186597-8053-4355-8818-808889252988  8Gi       RWO           standard      130m
```

16) また、以下を実行することで先ほど作成した NFS のフォルダにデータの実態が確認されていることを確認  
できます。

```
cd /var/nfs/qlik/  
ls
```



#### About Qlik

Qlik is on a mission to create a data-literate world, where everyone can use data to solve their most challenging problems. Only Qlik's end-to-end data management and analytics platform brings together all of an organization's data from any source, enabling people at any skill level to use their curiosity to uncover new insights. Companies use Qlik products to see more deeply into customer behavior, reinvent business processes, discover new revenue streams, and balance risk and reward. Qlik does business in more than 100 countries and serves over 48,000 customers around the world.

[qlik.com](http://qlik.com)

**NOTE – Please ensure you get always the latest copyright line from the brand portal.** © 2018 QlikTech International AB. All rights reserved. Qlik®, Qlik Sense®, QlikView®, QlikTech®, Qlik Cloud®, Qlik DataMarket®, Qlik Analytics Platform®, Qlik NPrinting®, Qlik Connectors®, Qlik GeoAnalytics®, Qlik Core®, Associative Difference®, Lead with Data™, Qlik Data Catalyst™, Qlik Associative Big Data Index™ and the QlikTech logos are trademarks of QlikTech International AB that have been registered in one or more countries. Other marks and logos mentioned herein are trademarks or registered trademarks of their respective owners.