

Auth0 IdP を利用した Qlik Sense Enterprise on Kubernetes の設定

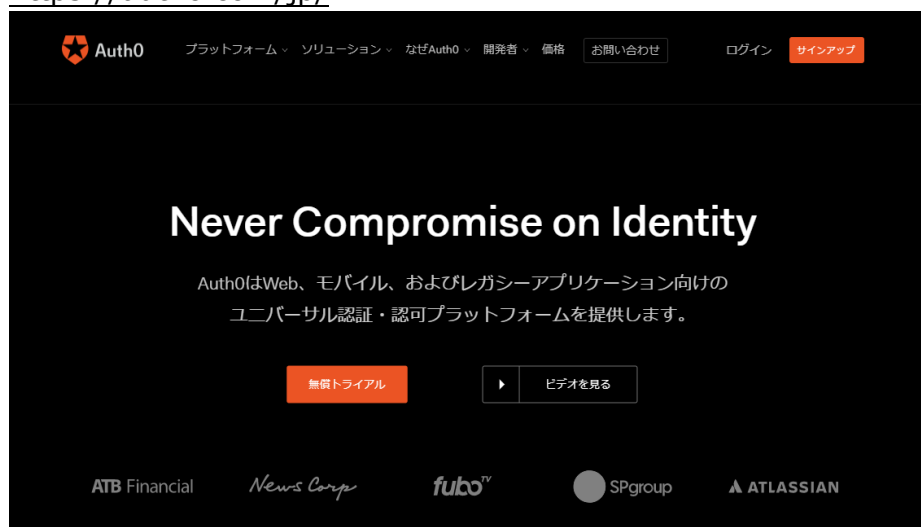
Qlik Sense Enterprise マルチクラウド演習

1.	Auth0へのアカウント登録	2
2.	Auth0アプリケーションの作成	4
3.	ユーザーデータベースの設定とユーザーの登録	6
4.	Qlik Sense Enterprise on Kubernetes側（Minikubeを利用の場合）でのAuth0を使った認証の設定	12
5.	認証エラーの場合のQlik Sense Enterprise on Kubernetes側のログの確認	18

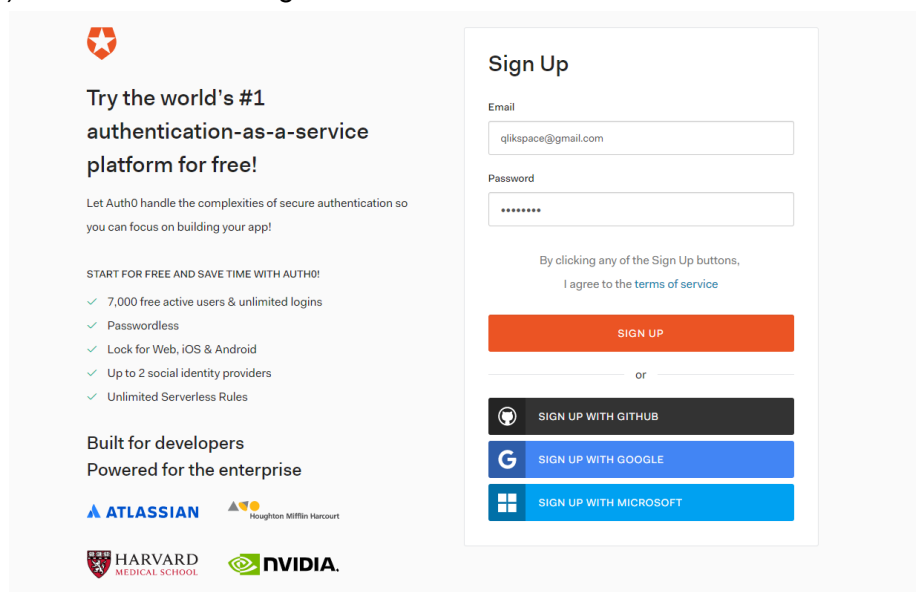
1.Auth0 へのアカウント登録

1) 以下のサイトにアクセスし、「無償トライアル」をクリックします。

<https://auth0.com/jp/>



2) メール、Github、Google もしくは Microsoft のアカウントでサインアップします。



3) 任意の Tenant Domain を入力し、Region を選択して「Next」をクリックします。

TENANT DOMAIN

qlikspace .au.auth0.com

To help you easily explore our product, we've selected a tenant domain name for you. Although you can't rename a tenant, you can always add more tenants to your account (for staging or production environments) later.

REGION

AU EU US

We can host all of your data in any of these regions. Useful if you need to comply with EU Data Protection Directive

NEXT

4) 項目を任意に入力し、「Create Account」をクリックします。

ACCOUNT TYPE

Are you creating this account for yourself or on behalf of a company?

Company Personal

COMPANY NAME **EMPLOYEES**

qlikspace 1-49

ROLE

Select your most applicable role.

Developer

MAIN CHALLENGE

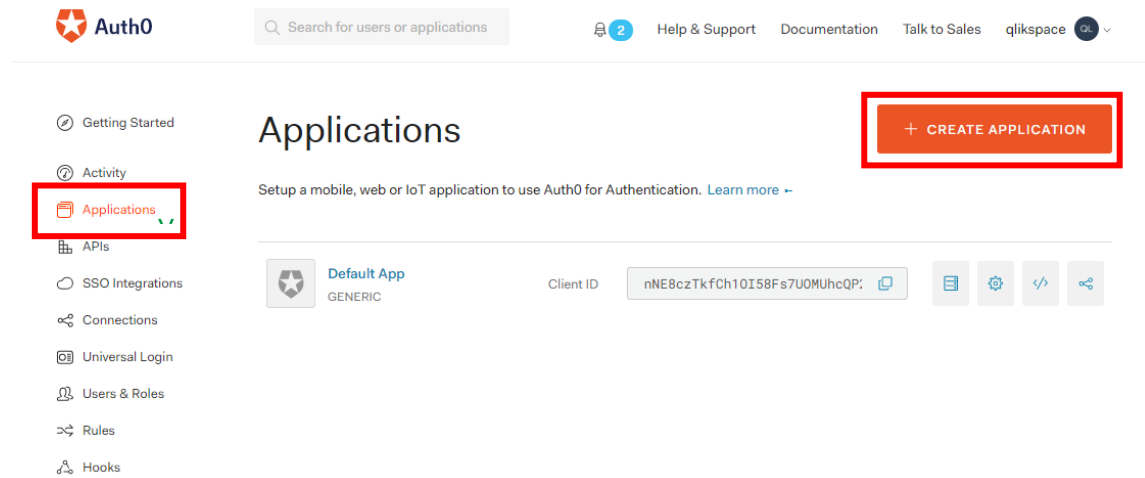
What's the main challenge you need to solve with Auth0?

Add auth to my app

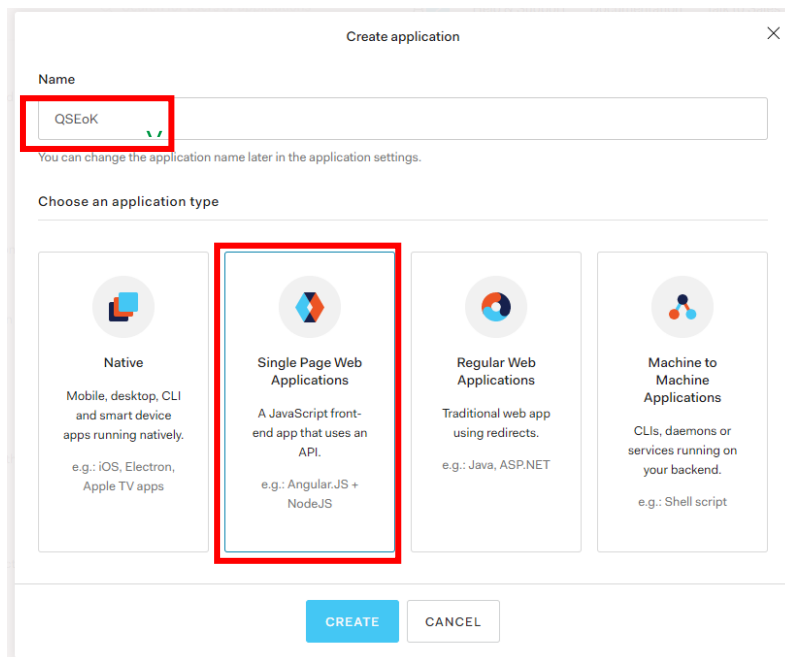
CREATE ACCOUNT

2.Auth0 アプリケーションの作成

1) 左側のメニューから「Applications」を選択し、「Create Application」をクリックします。

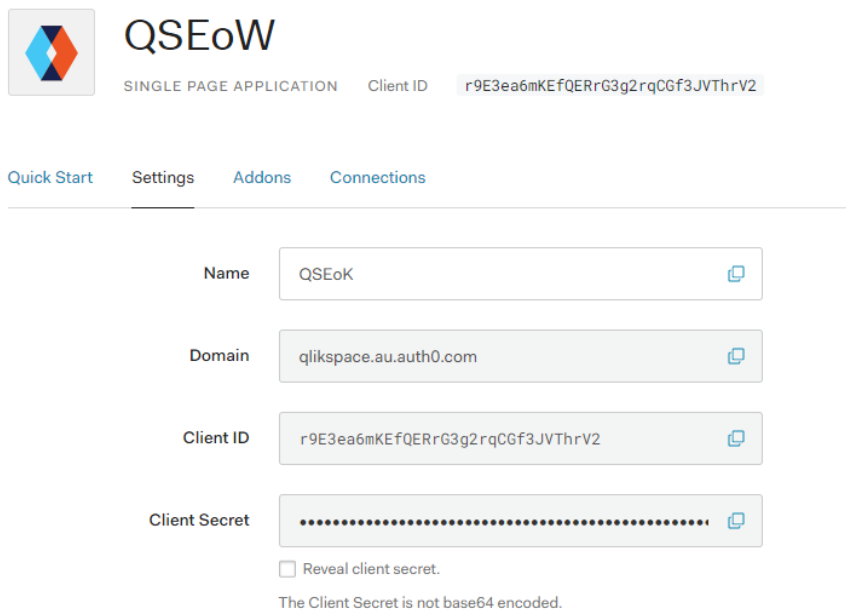


2) 「Name」に「QSEoK」と入力し、「Single Page Web Applications」を選択して「Create」をクリックします。



- 3) 「Settings」タブを開き、後続のステップで利用しますので「Domain」「Client ID」「Client Secret」をメモに残してください。

← Back to Applications



QSEoW
SINGLE PAGE APPLICATION Client ID r9E3ea6mKEfQERrG3g2rqCGf3JVThrV2

Quick Start Settings Addons Connections

Name QSEoK

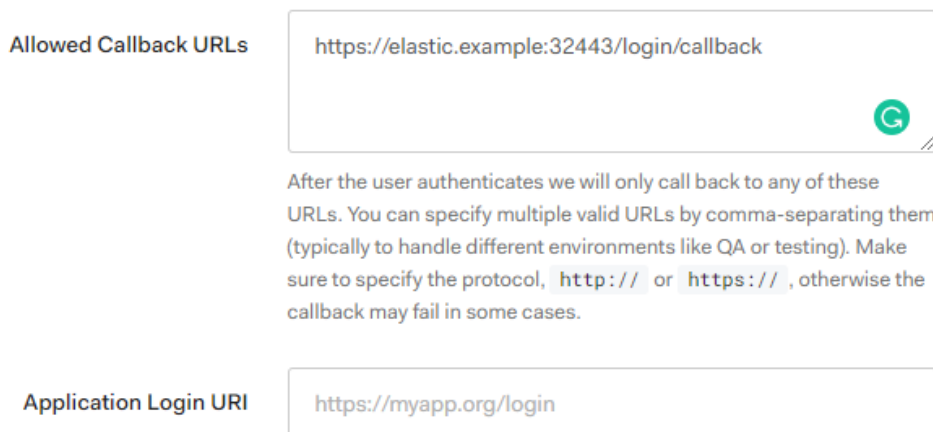
Domain qlikspace.au.auth0.com

Client ID r9E3ea6mKEfQERrG3g2rqCGf3JVThrV2

Client Secret

Reveal client secret.
The Client Secret is not base64 encoded.

- 4) 下にスクロールし、「Allowed Callback URLs」に「<https://elastic.example:32443/login/callback>」を入力します。(Azure の場合にはポート番号は不要です。また AWS では取得した URL に /login/callback を付加した URL をここで指定します。) Auth0 では登録された URL にのみコールバックを行い、ここでは Qlik Sense Enterprise on Kubernetes の URL に /login/callback を付加した URL を入力します。



Allowed Callback URLs https://elastic.example:32443/login/callback

After the user authenticates we will only call back to any of these URLs. You can specify multiple valid URLs by comma-separating them (typically to handle different environments like QA or testing). Make sure to specify the protocol, `http://` or `https://`, otherwise the callback may fail in some cases.

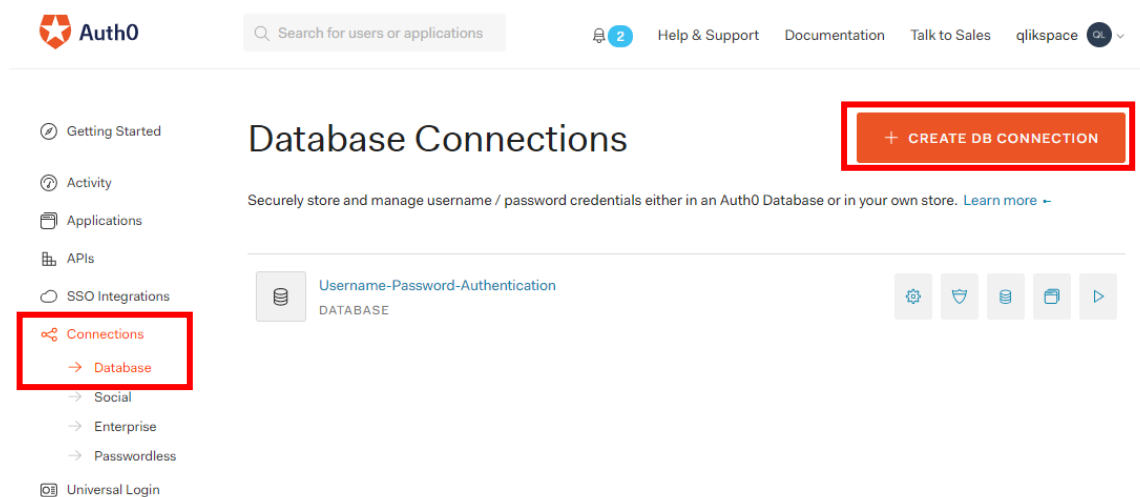
Application Login URI https://myapp.org/login

5) 下までスクロールして「Save Changes」をクリックします。



3. ユーザーデータベースの設定とユーザーの登録

1) 左側メニューから「Connections」>「Database」を選択し、「Create DB Connection」をクリックします。



2) 「Name」に「QSEoK-Users」と入力し「Create」をクリックします。

Getting Started

Activity

Applications

APIs

SSO Integrations

Connections

→ Database

→ Social

→ Enterprise

→ Passwordless

Universal Login

Users & Roles

→ Users

→ Roles

Rules

Hooks

Multifactor Auth

Emails

Logs

Anomaly Detection

Extensions

Get Support

New Database Connection

Settings

Name

QSEoK-Users

Must start and end with an alphanumeric character and can only contain alphanumeric characters and '-'. Can't have more than 35 characters.

Requires Username

Requires the user to provide a username in addition to email.

Username length

Set the minimum and maximum values allowed for a user to have as username.

YOU NEED TO ENABLE REQUIRES USERNAME IN ORDER TO USE THE USERNAME LENGTH SETTINGS.

Min Max

Disable Sign Ups

Check this if you want to prevent sign ups to your application. You will still be able to create users with your api credentials or from the dashboard.

CREATE CANCEL

3) 「Application」タブを開き、先ほど作成した「QSEoK」のアプリケーションのスイッチを有効化します。

QSEoK-Users

Database

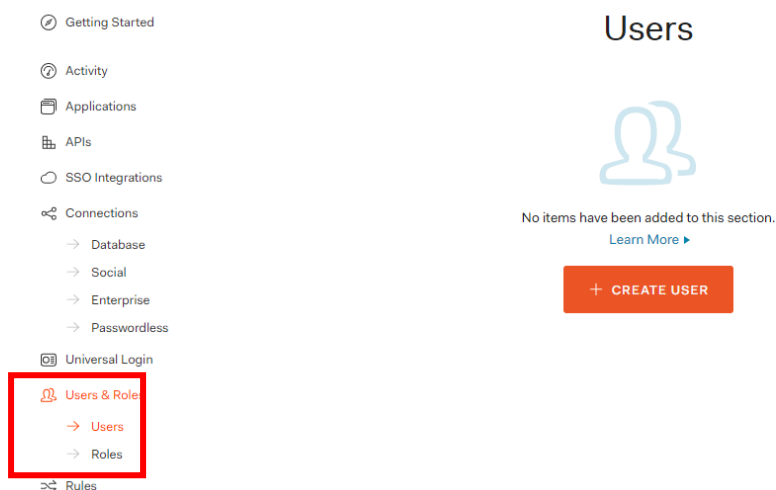
Settings Password Policy Custom Database Applications Try Connection

Applications using this connection.

Default App
GENERIC

QSEoK
SINGLE PAGE APPLICATION

4) 左側メニューから「Users & Roles」>「Users」を選択し、「Create User」をクリックします。

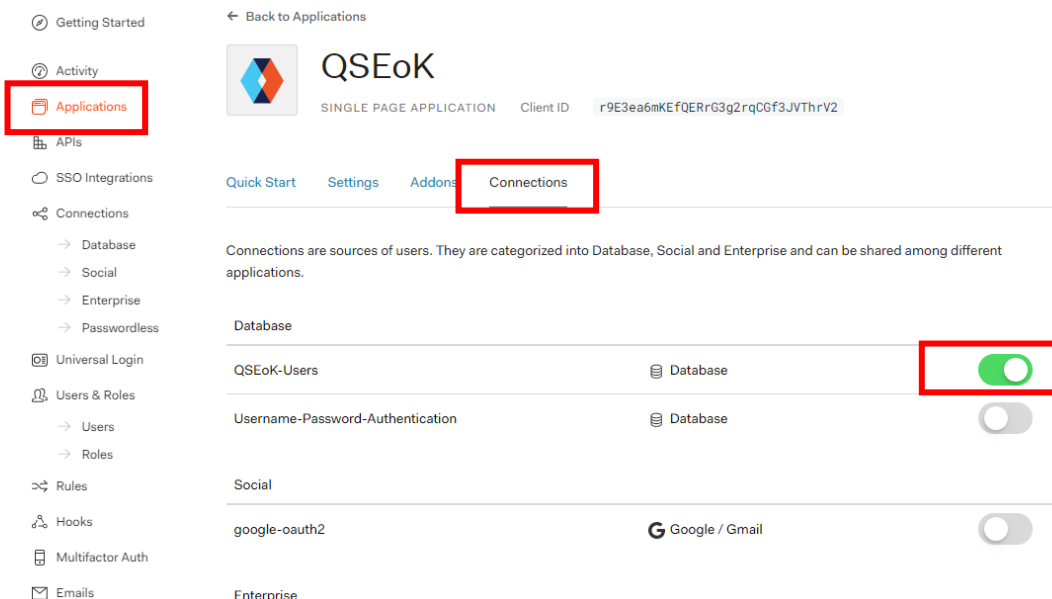


5) ユーザー情報を入力します。(メールは実在しないメールでも問題ありません)「Connection」には先ほど作成した「QSEoK-Users」を選択して「Create」をクリックします。

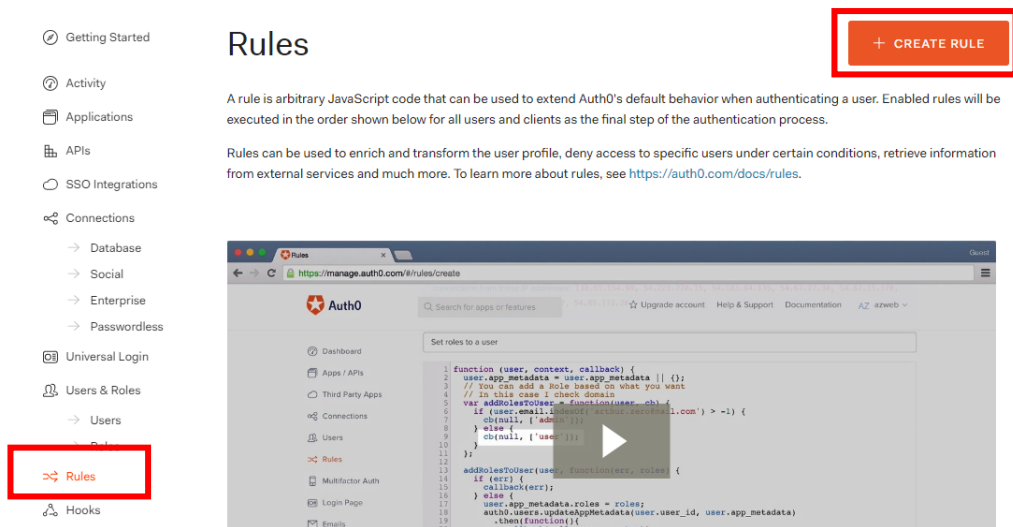
6) このユーザーにグループの設定を行います。スクロールし、「user_metadata」に以下の入力を行い「Save」をクリックします。他のユーザーを作成する場合には同様の作業を繰り返します。

```
{
  "groups": [
    "Everyone"
  ]
}
```

7) 左側メニューから「Applications」を選択して先ほど作成した「QSEoK」のアプリケーションを選択し、「Connections」タブをクリックして作成した「QSEoE-Users」の Database のみを有効化して他は無効化します。



8) 左側メニューから「Rules」を選択して「Create Rule」をクリックします。

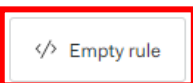


9) 「Empty rule」を選択します。

Pick a rule template

These are pre-made rules for common use cases that you can use or adapt to suit your needs.

Empty



Access Control


- 10) 「Script」を以下の通り変更し、「Name」を「Add groups claim」に変更して「Save Changes」をクリックします。これにより認証に Auth0 から Qlik Sense Enterprise に渡されるメタデータにグループ情報が含まれます。

```
function (user, context, callback) {
  if((user.user_metadata || {}).groups){
    context.idToken['https://qlik.com/groups'] = user.user_metadata.groups;
  }
  callback(null, user, context);
}
```

Name

Script

```
1 function (user, context, callback) {
2   if((user.user_metadata || {}).groups){
3     context.idToken['https://qlik.com/groups'] = user.user_metadata.groups;
4   }
5   callback(null, user, context);
6 }
```

 Please note that trying or debugging the rule will save it as well, overwriting its code.

[SAVE CHANGES](#) [▶ TRY THIS RULE](#) [🔄 INSTALL REAL-TIME LOGS](#)

- 11) 同様の手順でもう一つ別の Empty rule を作成し、「Script」を以下の通り変更して「Name」を「Add email to subject」に変更し、「Save Changes」をクリックします。これにより文字列で表現される User ID を理解しやすい email で置き換えて表現します。

```
function (user, context, callback) {
  context.idToken['https://qlik.com/sub'] = user.email;
  callback(null, user, context);
}
```

Name

Add email to subject

Script

```

1 function (user, context, callback) {
2   context.idToken['https://qlik.com/sub'] = user.email;
3   callback(null, user, context);
4 }
5 |

```

⚠ Please note that **trying** or **debugging** the rule will save it as well, overwriting its code.

SAVE CHANGES TRY THIS RULE INSTALL REAL-TIME LOGS

- 12) ユーザーログインのテストを行います。「Connections」>「Database」を選択して、先ほど作成した「QSEoK-Users」をクリックします。

Getting Started Activity Applications APIs SSO Integrations **Connections** Universal Login

Database Social Enterprise Passwordless

Database Connections

+ CREATE DB CONNECTION

Securely store and manage username / password credentials either in an Auth0 Database or in your own store. [Learn more](#) -

QSEoK-Users DATABASE	⚙️ 🛡️ 🗄️ 📄 ▶️
Username-Password-Authentication DATABASE	⚙️ 🛡️ 🗄️ 📄 ▶️

- 13) 「Try Connection」をクリックします。

Getting Started Activity Applications APIs SSO Integrations **Connections** Universal Login

Database Social

← Back to Database Connections

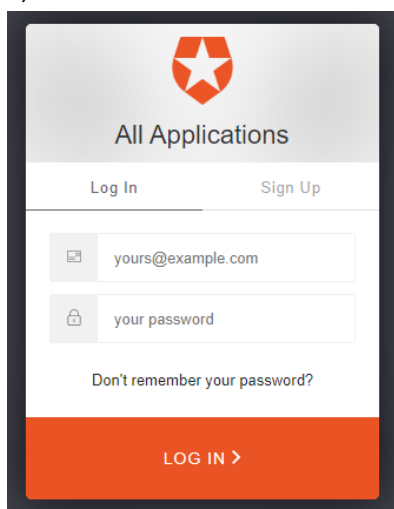
QSEoK-Users

Database

Settings Password Policy Custom Database Applications **Try Connection**

These settings will only affect this particular database connection. If you need to add custom beha

- 14) ログイン画面が表示されますので、先ほど追加したユーザーの Email とパスワードを入力します。



- 15) ログインが成功し、メタデータに追加したルールに従って groups に Everyone、sub に Email が含まれていることを確認します。

```
{
  "sub": "auth0|5d6642b022eec70d5f038737",
  "nickname": "harley",
  "name": "harley@qlikcloud.com",
  "picture": "https://s.gravatar.com/avatar/8d63c12ea784276cee64f3eda03e7b11?s=480&r=pg&d=https%3A%2F%2Fcdn.auth0.com%",
  "updated_at": "2019-08-28T10:56:07.757Z",
  "https://qlik.com/groups": [
    "Everyone"
  ],
  "https://qlik.com/sub": "harley@qlikcloud.com"
}
```

4. Qlik Sense Enterprise on Kubernetes 側 (Minikube を利用の場合) での Auth0 を使った認証の設定

ここでは Minikube を使った場合の Qlik Sense Enterprise on Kubernetes 側の設定方法をご説明します。Azure Kubernetes Service (AKS) や Amazon Elastic Kubernetes Service (EKS) などの設定を行う場合にはそれぞれのドキュメントをご参照ください。

- 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) エディタを利用して以下の内容の values-auth0.yaml ファイルを作成します。尚、以下は Minikube を利用した場合の例になりますので、Azure Kubernetes Service(AKS)や Amazon Elastic Kubernetes Service(EKS)などを利用するなどの場合には別紙をご参照の上、そこで指定された YAML ファイルを利用してください。

前段の演習「01_QSE on Kubernetes の導入」で NFS Provisioner の設定を既に行っているか、いないかで利用すべきファイルの内容が異なります。(どちらを利用すべきか不明な場合は「① NFS Provisioner の設定を行っていない場合」をご利用ください。) また、黄色でハイライトされている「discoveryUrl」、「clientId」、「clientSecret」、「postLogoutRedirectUri」の値は前段の手順でメモした Auth0 上のアプリケーションの「Domain」、「Client ID」、「Client Secret」の値に基づいて書き換える必要があります。

① NFS Provisioner の設定を行っていない場合

```
#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

identity-providers:
  secrets:
  idpConfigs:
    - discoveryUrl: "https://<Domain>/.well-known/openid-configuration"
      clientId: "r9E3ea6mKEfQERrG3g2rqCGf3JVThrV2"
      clientSecret :
"BZLGoB5ubfI1l2mwBVfxUkX654QiEJTiaXOGEtcTxr7CO8D5yNVFYHWp6Tm5zD1g"
      realm: "Auth0"
      postLogoutRedirectUri: "https://<Domain>/v2/logout/"
      hostname: "elastic.example"
      claimsMapping:
        client_id: [ "client_id", "azp" ]
        groups: "/https:~1~1qlik.com~1groups"
        sub: ["/https:~1~1qlik.com~1sub","sub"]

```

② NFS Provisioner の設定を行っている場合

```

#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
```



```
identity-providers:
  secrets:
    idpConfigs:
      - discoveryUrl: "https://<Domain>/well-known/openid-configuration"
        clientId: "r9E3ea6mKEfQERrG3g2rqCGf3JVThrV2"
        clientSecret :
          "BZLGoB5ubfI1I2mwBVfxUkX654QiEJTiaXOGEtcTxr7CO8D5yNVFYHWp6Tm5zD1g"
        realm: "Auth0"
        postLogoutRedirectUri: "https://<Domain>/v2/logout/"
        hostname: "elastic.example"
        claimsMapping:
          client_id: [ "client_id", "azp" ]
          groups: "/https:~1~1qlik.com~1groups"
          sub: ["/https:~1~1qlik.com~1sub","sub"]
```

3) Qlik Sense Enterprise on Kubernetes を展開します。

```
helm upgrade --install qliksense qlik/qliksense -f values-auth0.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
```

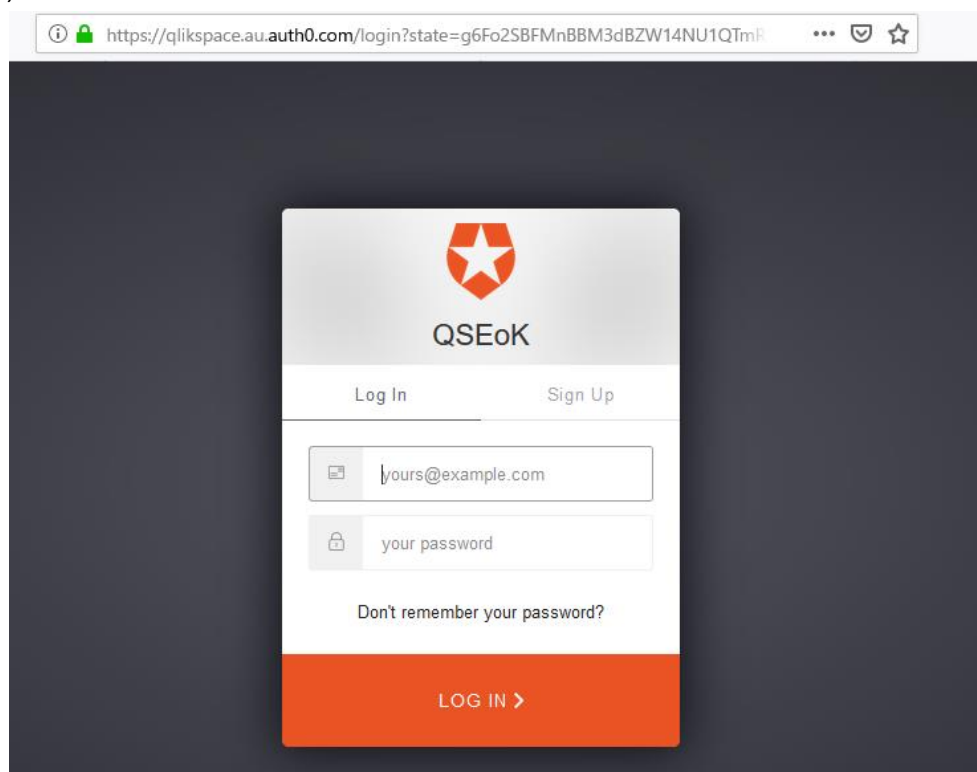
以下のコマンドを実行すると定期的な繰り返し処理で上記のコマンドを実行することも可能です。

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

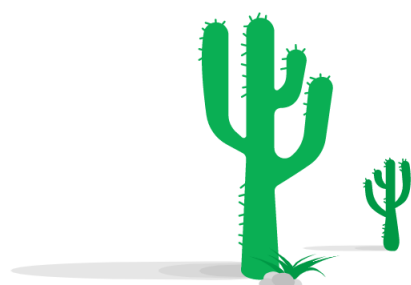
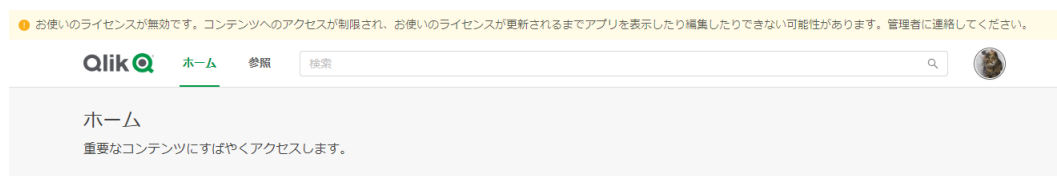
5) Windows のホストコンピューター上でブラウザを開き、以下の URL にアクセスします。

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

6) Auth0 のログイン画面が表示されますので、先ほど追加したユーザーの Email とパスワードを入力します。



7) 以下のようなハブのトップ画面が表示されます。これで環境のセットアップが完了しましたので、別紙「03_Qlik Sense Enterprise on Kubernetes 上の管理コンソールでのライセンス有効化」の手順に従ってサイトのライセンスを有効化します。



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。

5. 認証エラーの場合の Qlik Sense Enterprise on Kubernetes 側のログの確認

1) 以下のコマンドを実行し、edge-auth の Pod の ID を確認します。

```
kubectl get pods | grep auth
```

```
[root@minikube ~]# kubectl get pods | grep auth
qliksense-edge-auth-78f4f76cb8-722d9          2/2      Running   9          17h
```

2) 以下のコマンドで edge-auth Pod の edge-auth コンテナのログを表示します。(必要に応じて grep で絞り込みを行います。)

```
kubectl logs qliksense-edge-auth-78f4f76cb8-722d9 edge-auth
```

※ 以下のように-fフラグを追加するとストリーム表示となります。

```
kubectl logs qliksense-edge-auth-78f4f76cb8-722d9 edge-auth -f
```

3) 以下のようなログが表示され、ブラウザからの URL リクエストやエラーの内容が表示されます。

```
{"ip":"172.17.0.1","url":"/login/callback?code=***MASKED***&state=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyZnAiOiJlbnV4cXVpMU1SekItemtYTGdFeWh0LTdKOEhtNVFaVlk1ZFJSeG5seFRGMlNBIiwicmQiOiJodHRwczovL2VsYXN0aWMuZXhhbXBsZTozMjQ0My8iLCJpYXQiOiE1NjcwMDExNDksImV4cCI6MTU2NzYwNTk0OX0.Amna7F8boR8r11RNq0fPssTJVyE72OlfEizB6D2GcT0","method":"GET","query":[{"code","***MASKED***"}, {"state","eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyZnAiOiJlbnV4cXVpMU1SekItemtYTGdFeWh0LTdKOEhtNVFaVlk1ZFJSeG5seFRGMlNBIiwicmQiOiJodHRwczovL2VsYXN0aWMuZXhhbXBsZTozMjQ0My8iLCJpYXQiOiE1NjcwMDExNDksImV4cCI6MTU2NzYwNTk0OX0.Amna7F8boR8r11RNq0fPssTJVyE72OlfEizB6D2GcT0"}],"pid":1,"type":"edge-auth","version":"2.38.4","buildInfo":{"SHA":"d2953be","name":"edge-auth","version":"2.38.4","buildTime":"2019-06-21T19:13:26.093Z"},"module":"services/identityProvider","error":"AssertionError [ERR_ASSERTION]: id_token issued in the future","stack":"AssertionError [ERR_ASSERTION]: id_token issued in the future\n    at Client.validateIdToken (/usr/src/app/node_modules/openid-client/lib/client.js:586:7)\n    at grant.then.then.tokenset (/usr/src/app/node_modules/openid-client/lib/client.js:423:32)\n    at process._tickCallback (internal/process/next_tick.js:68:7)","code":"EAS-7","logTraceId":"qCa0rzS8mNtZncBjmbS1p5XE","headers":{"host":"elastic.example:32443","x-request-id":"ef98a974f98691d380b9df096615c03c","x-real-
```

```
ip":"172.17.0.1","x-forwarded-for":"172.17.0.1","x-forwarded-
host":"elastic.example:32443","x-forwarded-port":"443","x-forwarded-
proto":"https","x-original-
uri":"/login/callback?code=***MASKED***&state=eyJhbGciOiJIUzI1NiIsInR5cCI6Ikp
XVCJ9.eyJyZnAiOiJVQWJpMU1SekItemtYTdGdFeWh0LTdKOEhtNVFaVlk1ZFJSeG5seFR
GMINBIiwicmQiOiJodHRwczovL2VsYXN0aWMuZXhhbXBsZTozMjQ0My8iLCJpYXQiOiJE
1NjcwMDExNDksImV4cCI6MTU2NzYwNTk0OX0.Amna7F8boR8r11RNq0fPssTJVyE72
OlfEizB6D2GcT0","x-scheme":"https","user-agent":"Mozilla/5.0 (Windows NT 10.0;
WOW64; rv:68.0) Gecko/20100101
Firefox/68.0","accept":"text/html,application/xhtml+xml,application/xml;q=0.9,*/*;
q=0.8","accept-language":"en-US,en;q=0.5","accept-encoding":"gzip, deflate,
br","upgrade-insecure-
requests":"1"}, "logseverity":"ERROR", "level":"error", "message":"Failure during IDP
callback: AssertionError [ERR_ASSERTION]: id_token issued in the
future", "timestamp":"2019-08-28T14:06:20.941Z
```



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

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.