Using SAML for Kibana Single Sign On

Jochen Kressin · 2018-08-16

Search Guard has full support for SAML. That means you can use your favorite identity provider like Keycloak, Okta or Auth0 to implement easy Kibana Single Sign-On. In this post, we walk you through the steps to set up SAML authentication with Okta.

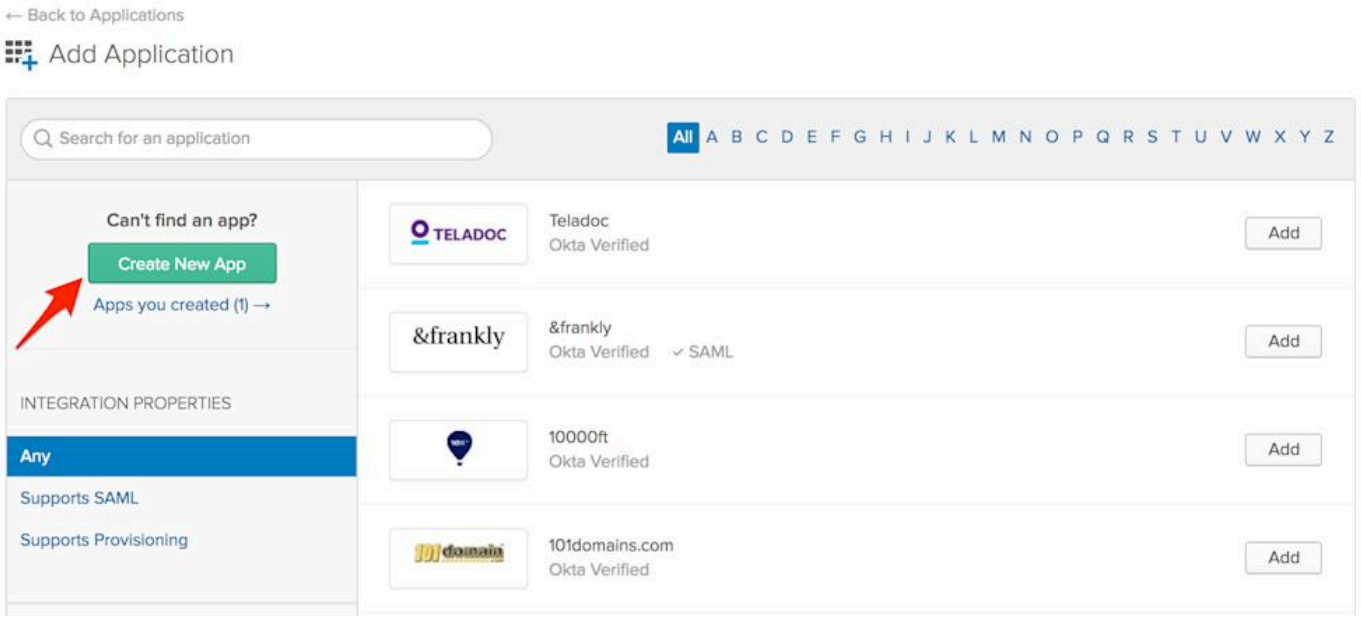
Prerequisites

To use SAML you need to install Search Guard v23 or above and the Search Guard Kibana plugin version 14 or above. In this article, we will use Elasticsearch 6.3.2, so we install

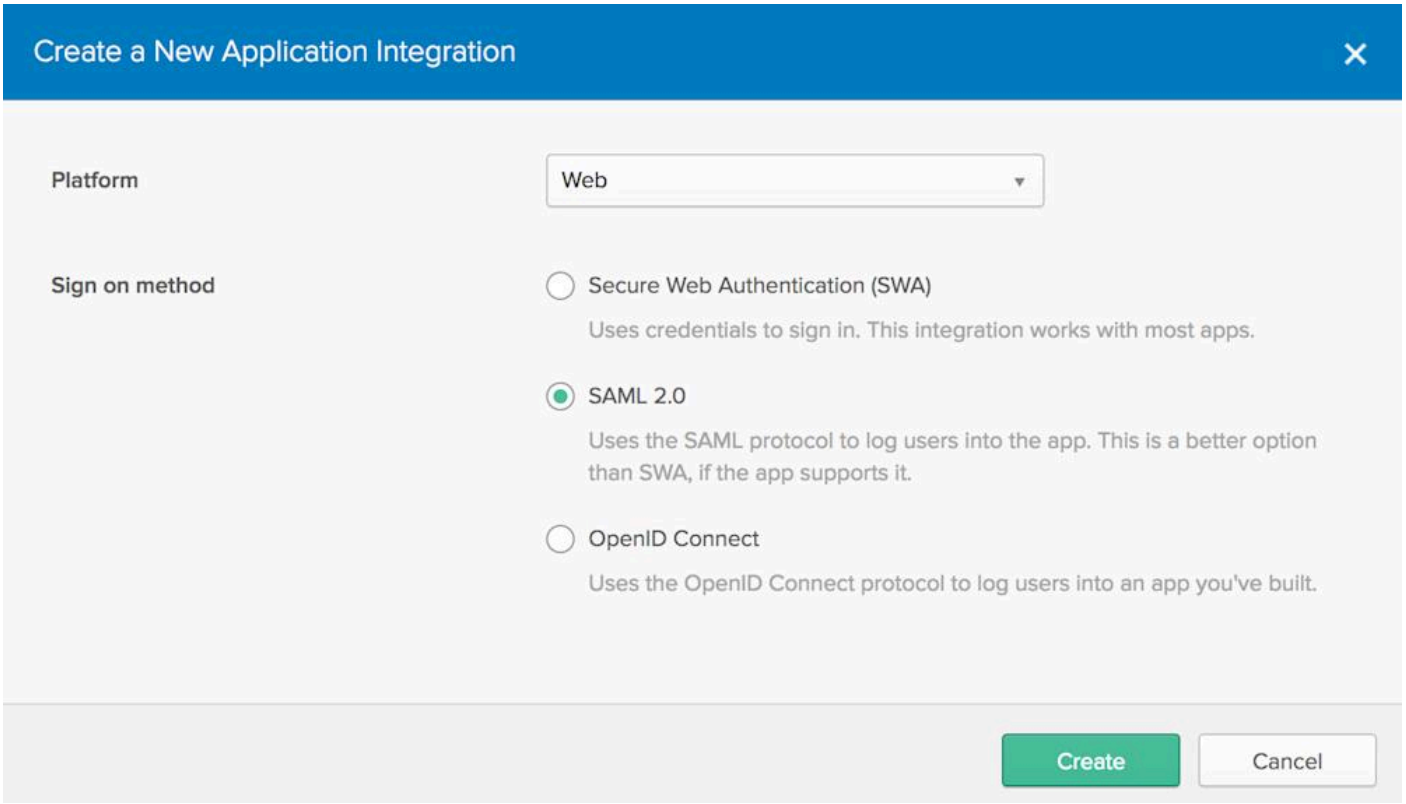
- Search Guard 6.3.2-23.0
- Kibana Plugin 6.3.2-14

Setting up a SAML application in Okta

To use Okta as our identity provider for Kibana we need to first set up a new application with SAML support. After logging in to Okta, click on *Application → Add Application → Create new application*.



On the next screen, we choose "Web app" as type and "SAML 2.0" as the authentication method.

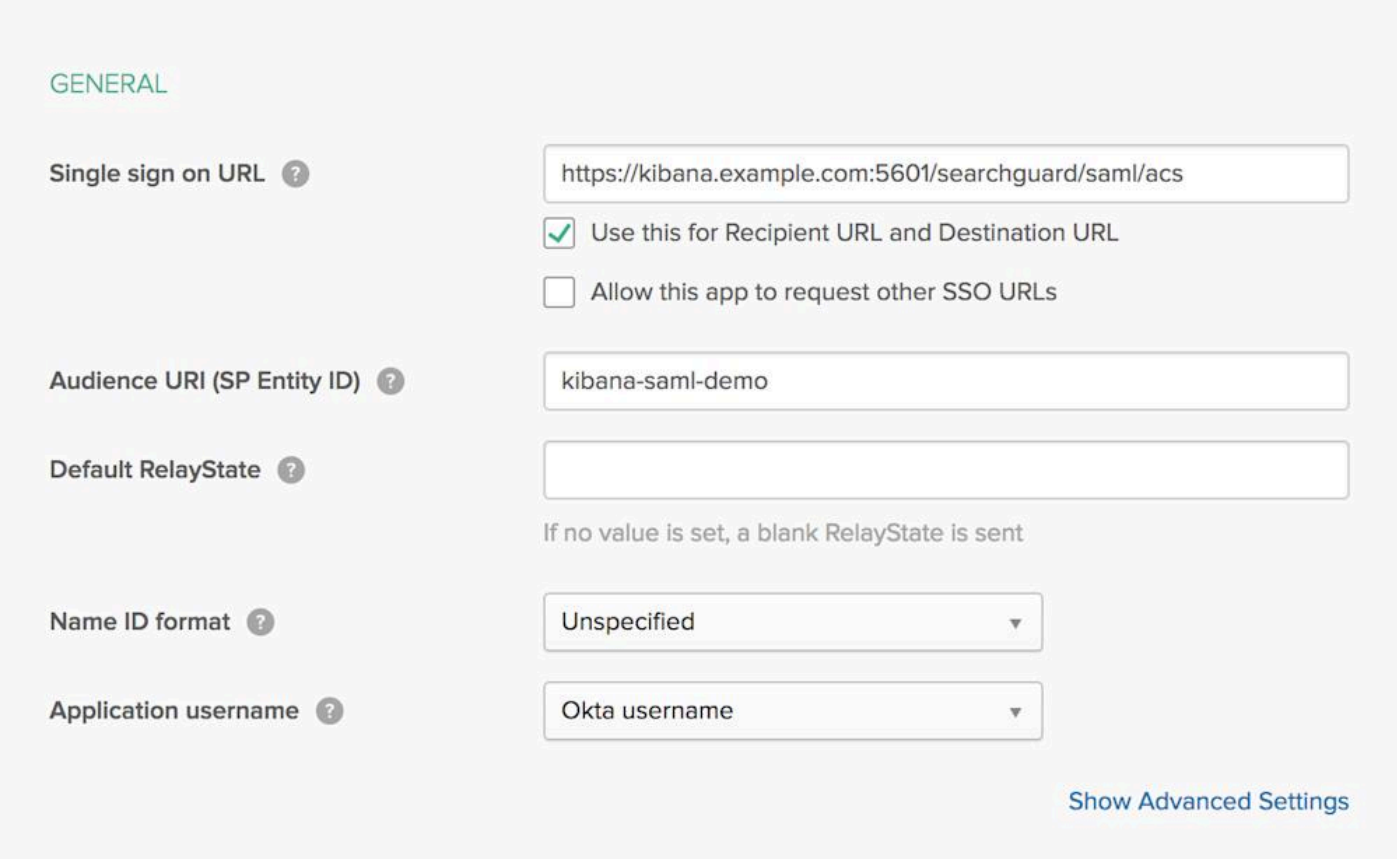


After we have chosen a name for our new application and clicking next, we are taken to the SAML settings screen. The two crucial settings here are the *Single sign on URL* and the *Audience URI (SP Entity ID)*

The *Single sign on URL* specifies the URI where Okta should redirect the request after successful login. This endpoint in Kibana is provided by the Search Guard Kibana plugin, and has the format:

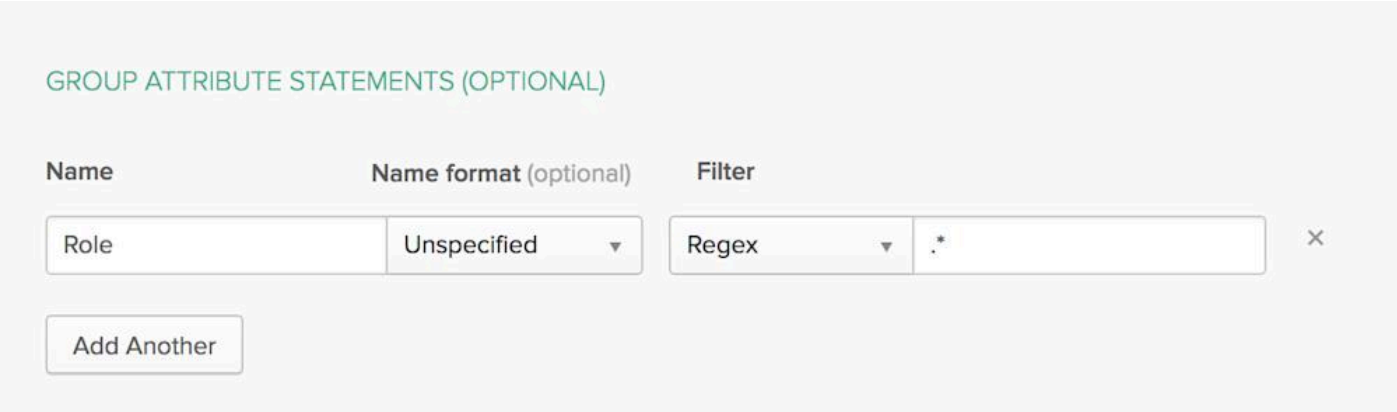
```
https://<kibana_base_url>:<kibana_port>/searchguard/saml/acs
```

Note that Okta requires that the redirect URI be https. Unencrypted HTTP will not work. We will configure Kibana to use HTTPS instead of HTTP in later steps. The *Audience URI (SP Entity ID)* is the identifier for the service provider. You can choose any name here, for this demo we use *kibana-saml-demo*. We need this name later on when we configure SAML support in Search Guard.



Adding roles to the SAML response

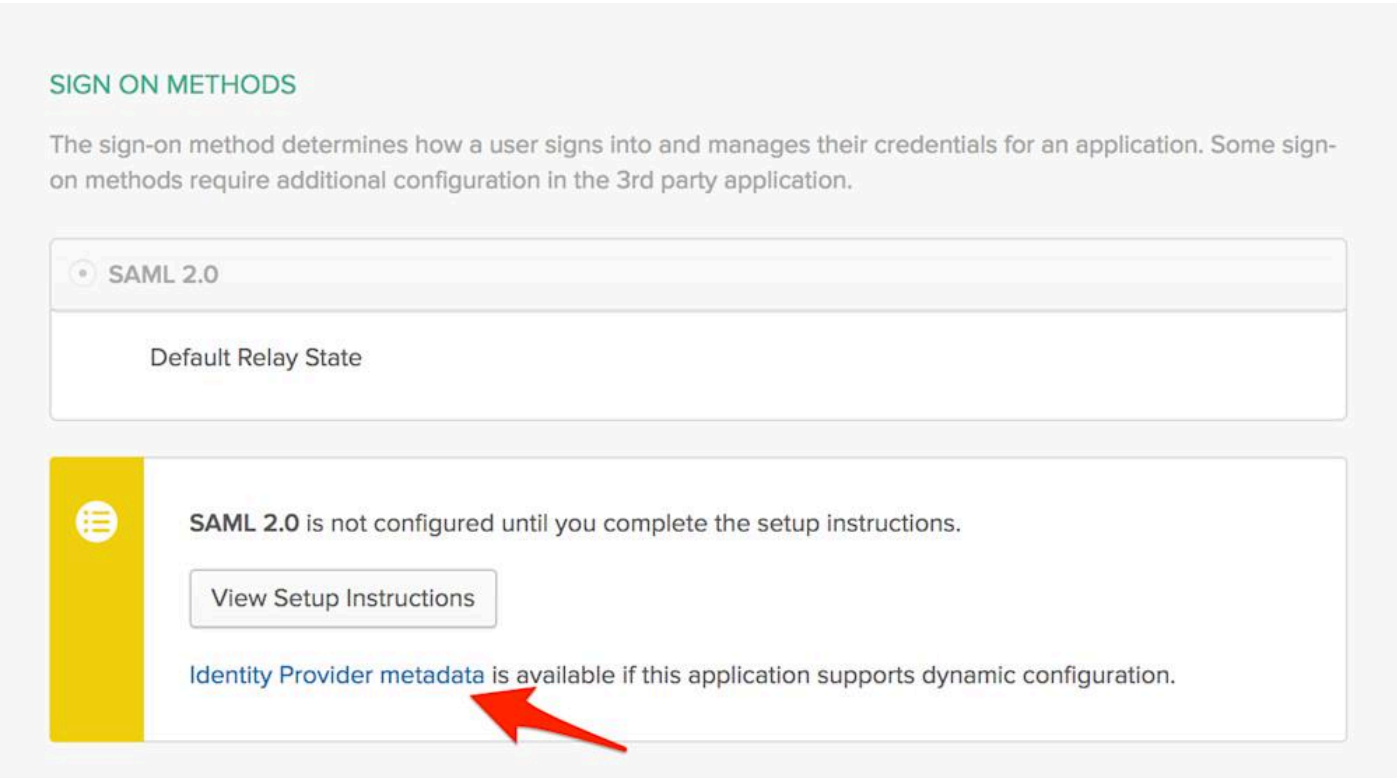
Since we want to use the Okta groups of a user to assign Search Guard roles, we need to make sure that these groups are added to the SAML response when the user logs in. In Okta this is achieved by adding Group Attribute Statements. The *Name* field defines the name of the SAML assertion that will hold the groups of the user. Since we want to add all groups of the user to the SAML assertion, we will use a regular expression as a filter that matches all groups:



Metadata endpoint

After finishing the setup wizard, we're taken to the *Settings* screen. Here we get access to the *SAML metadata*. This metadata makes it possible to auto-configure the client application and contains information about endpoints, formats, encryption and ciphers.

We click on *Identity Provider metadata* and save the XML file in the config directory of Elasticsearch. We name the file *okta.xml*:

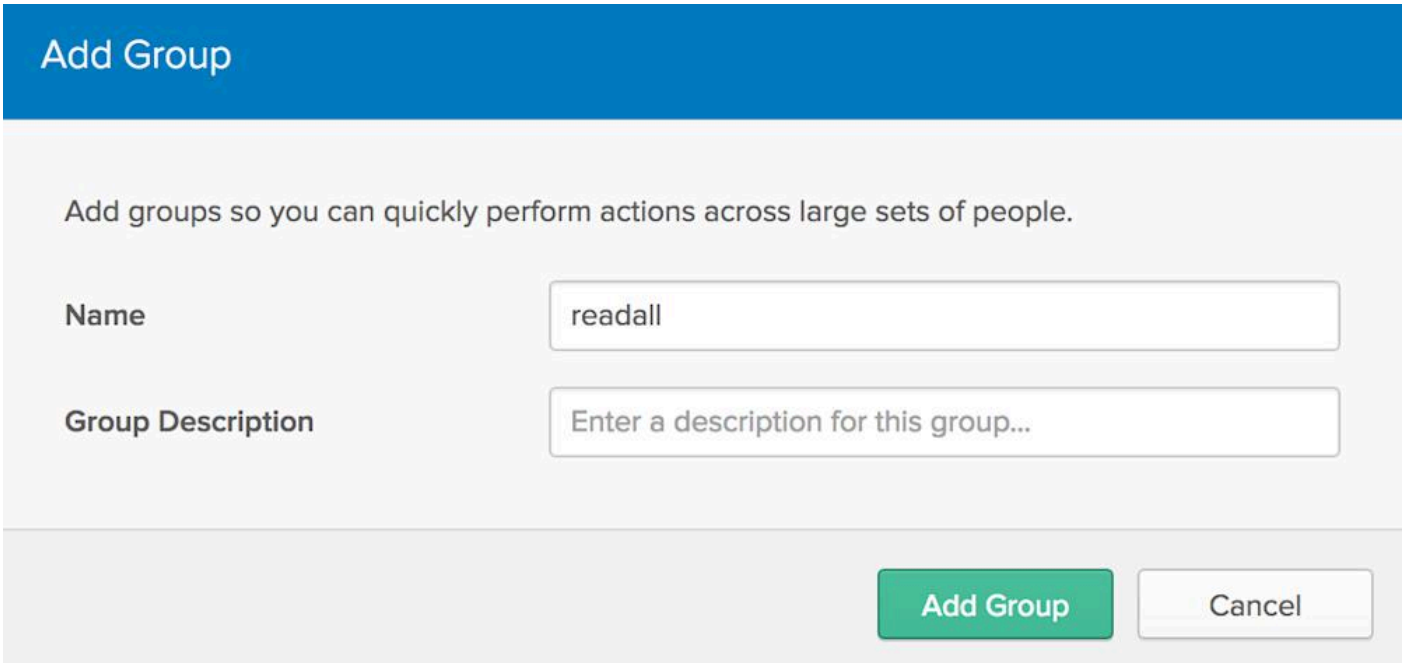


Adding users and groups

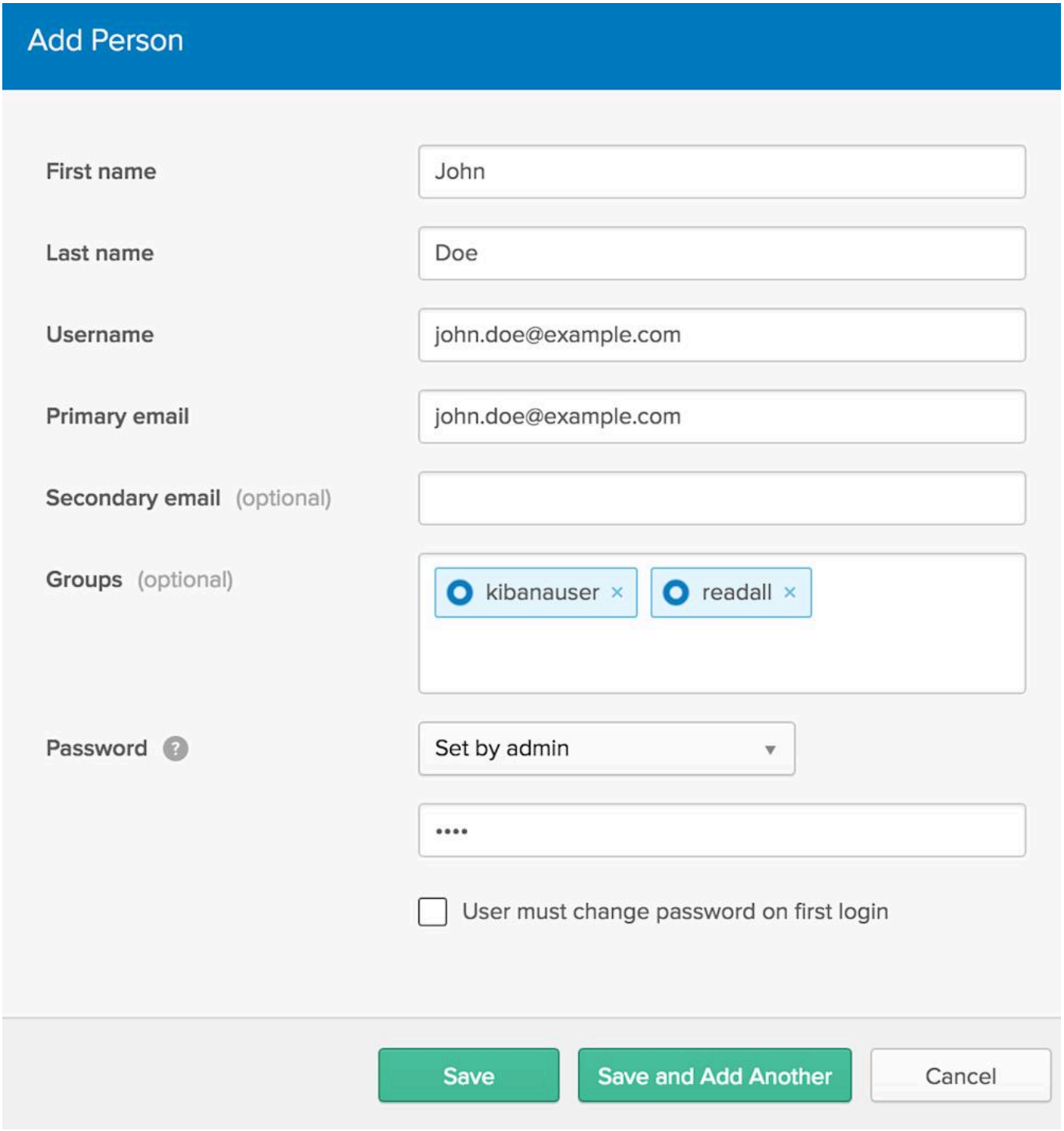
Now it's time to add a user and the groups o the user. We will start with the groups. Our goal is to create a user with read-only access to all indices and access to Kibana. Since we use the Search Guard demo configuration, we want the user to have the following backend roles:

- kibanauser
- read_all

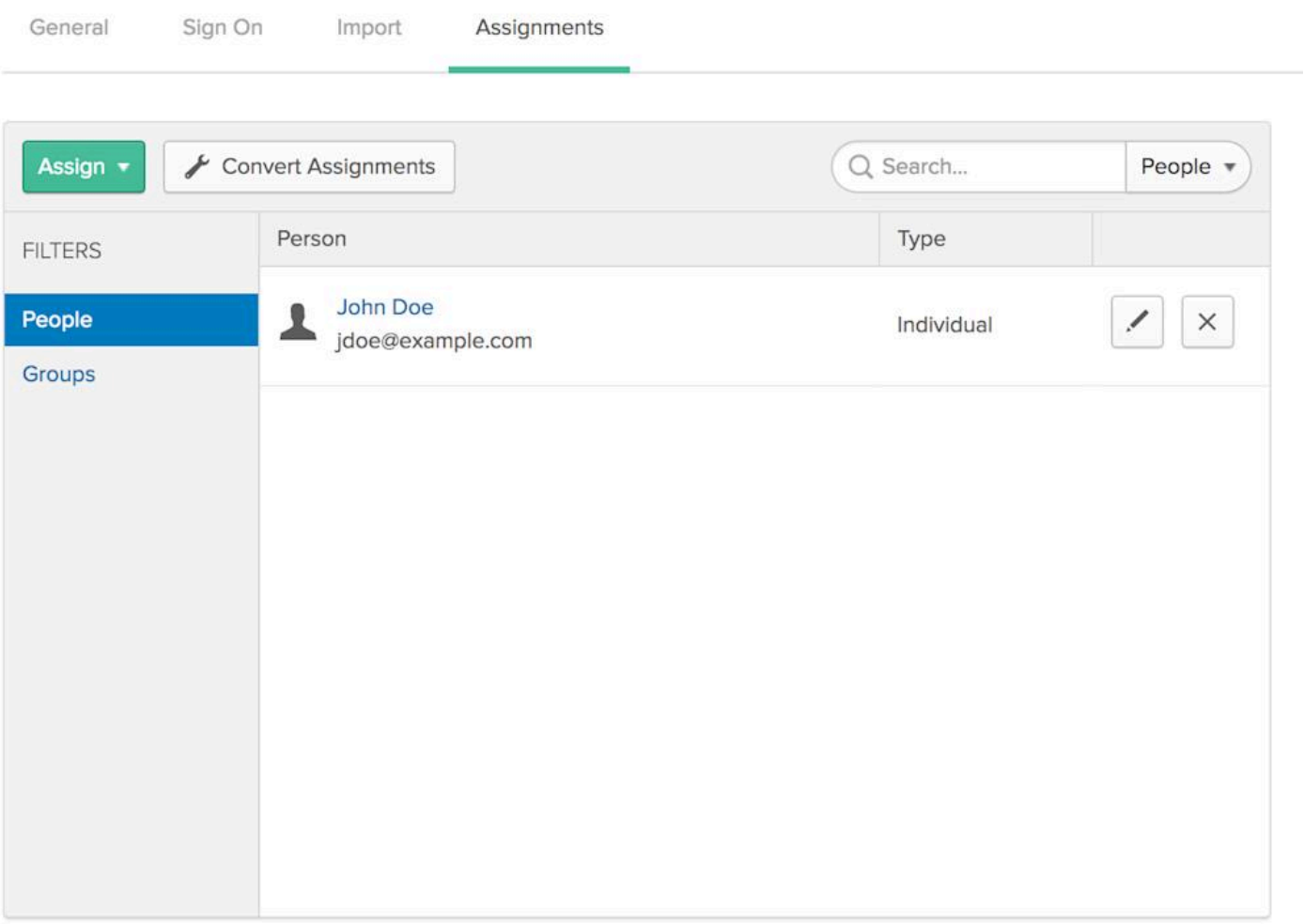
These backend roles are mapped to the corresponding Search Guard roles in the demo configuration. So we click on *Directory → Groups → Add Group* to create the *kibanauser* and *read*all_ group.



We then create a user by clicking on *Directory → People → Add Person*. We assign the newly created roles kibanaserver and readall.



Finally, we need to assign the user jdoe to the Kibana SAML Demo application:



Configure SAML for Elasticsearch

Now we have everything in place for configuring SAML for both Elasticsearch and Kibana. Since Search Guard can read and make use of the published SAML metadata, the actual configuration is pretty straight-forward.

Our goal is to use SAML for Kibana, but still being able to access Elasticsearch API directly. So we configure two authentication domains, one leveraging Basic Authentication and the internal user database, and one for SAML support:

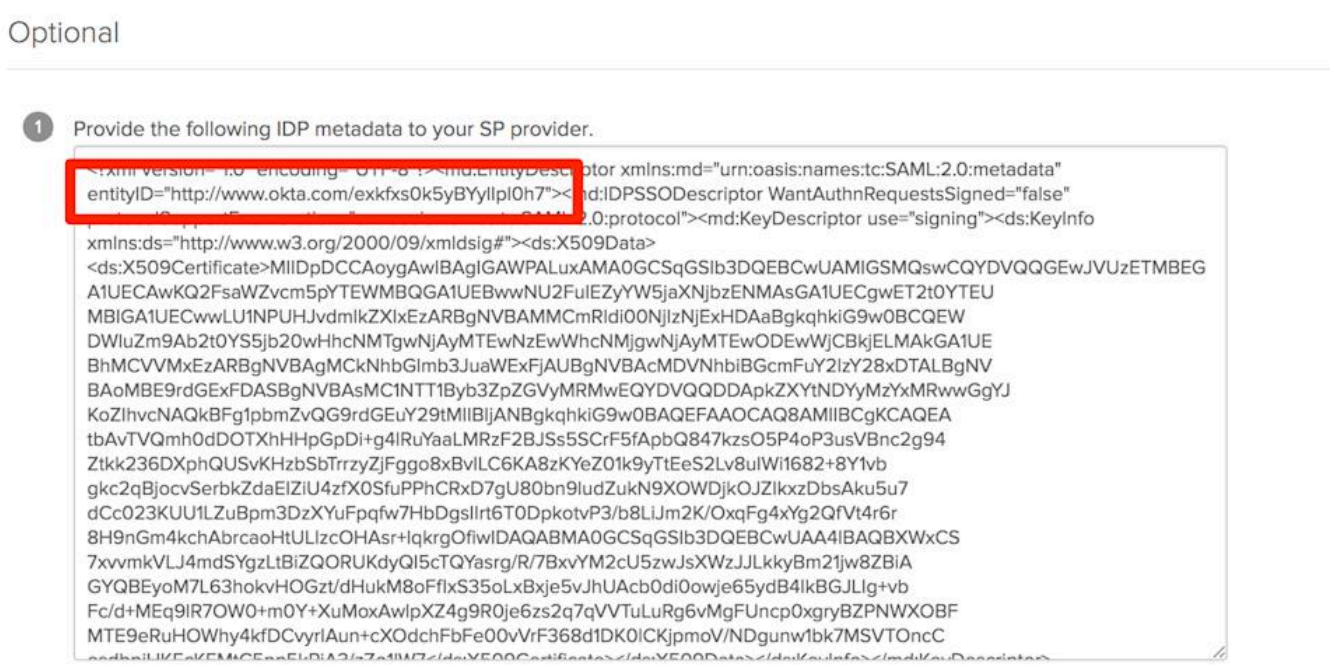
```
authc:
  basic_internal_auth_domain:
    enabled: true
    order: 0
    http_authenticator:
      type: basic
      challenge: false
    authentication_backend:
      type: intern
  saml_auth_domain_okta:
     http_enabled: true
     transport_enabled: true
     order: 1
     http_authenticator:
        type: 'saml'
        challenge: true
        config:
            idp:
              metadata_file: okta.xml
              entity_id: http://www.okta.com/exkflvnptwlnzRkp40h7
            sp:
              entity_id: kibana-saml-demo
            kibana_url: https://kibana.example.com:5601/
            roles_key: Role
            exchange_key: 'peuvgOLrjzuhXfRUwzU...'
```

idp.metadata_file

This is name of the metadata XML file we downloaded from Okta. The file must be placed in the *config* directory of Elasticsearch. The *metadata_file* path must be relative to the *config* directory.

idp.entity_id

This is the unique ID of your identity provider. You can find this ID for example in the SAML metadata:



sp.entity_id

This is the service provider entity ID. We configured this name under *Audience URI (SP Entity ID)* when we set up the SAML application in Okta.

kibana_url

This is the base URL of your Kibana installation.

roles_key

The attribute name under which the roles can be found in the SAML response. We have used the name *Role* when we configured the group attribute statements during setup.

exchange_key

SAML, unlike other protocols like JWT or Basic Authentication, is not meant to be used for exchanging user credentials with each request. Therefore Search Guard trades the heavy-weight SAML response for a light-weight JSON web token. This token is signed with an exchange key that you can choose freely. The key should be at least 32 characters long.

Configure SAML for Kibana

Since most of the heavy lifting is done by Search Guard, the Kibana configuration is straightforward. We need to configure SAML as the authentication type, and whitelist the redirect and logout URL:

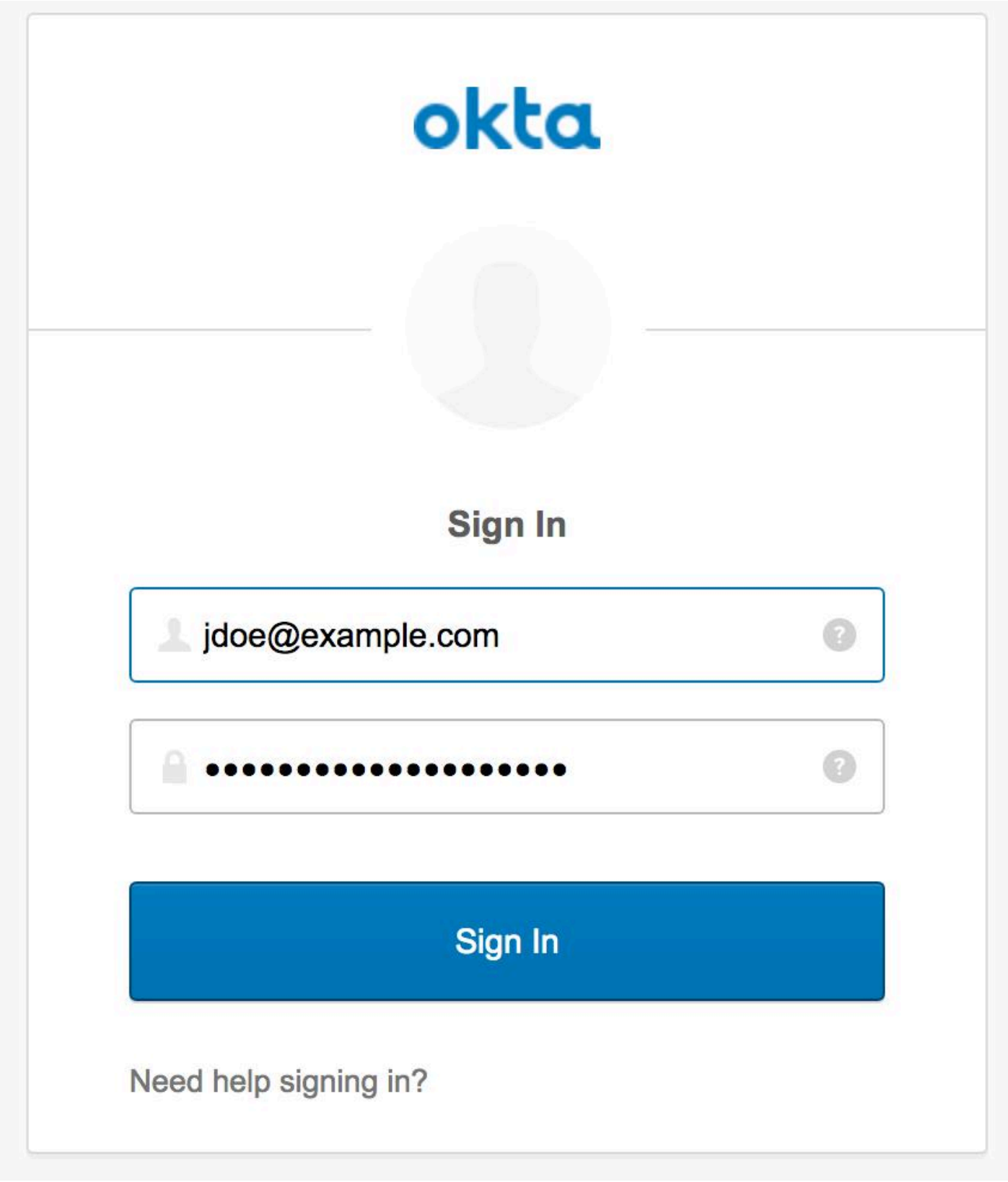
```
searchguard.auth.type: "saml"
server.xsrf.whitelist: ["/searchguard/saml/acs", "/searchguard/saml/logout"]
```

Since Okta requires the service provider (in our case Kibana) to use HTTPS, we configure the SSL certificate and key as well:

```
server.ssl.enabled: true
server.ssl.certificate: ./config/kibana.crtfull.pem
server.ssl.key: ./config/kibana.key.pem
```

Trying it out

Now we're ready to try our installation. We access Kibana via https://kibana.example.com:5601 and get redirected to the Okta login page:



After we log in, we get redirected to Kibana and can start using it.

Summary

In this article, we demonstrated how to setup SAML authentication for Kibana Single Sign-On. We used Okta as our identity provider. Since SAML is an industry standard, there is a wide variety of IdPs to

choose from, including Auth0, Keycloak or OneLogin.

Where to go next

- Read the official SAML configuration documentation
- If you have trouble configuring SAML, refer to our SAML troubleshooting guide
- If you have any question, ask them on our Search Guard Forum

Image: shutterstock / jirsak

This article was published on the Search Guard blog: https://search-guard.com/blog/kibana-elasticsearch-saml/