Documentation for builtin roles

https://www.elastic.co/guide/en/elasticsearch/reference/8.18/built-in-roles.html

# Creating a role and users, and giving privileges

Click on sandwich menu > click stack management > click roles under security tab > click on Create role A screenshot of a computer

AI-generated content may be incorrect.

Click on create after giving these privilegesA screenshot of a chat

AI-generated content may be incorrect.

Click on sandwich menu > click stack management > click users under security tab > click on Create usersA screenshot of a computer

AI-generated content may be incorrect.

Create the user A screenshot of a computer

AI-generated content may be incorrect.

Earlier we had create the kibana\_system through the api using curl

curl -X POST "https://localhost:9200/\_security/user/kibana\_system/\_password" -u elastic:BMV=YdeAn-QtVm5xageh -H "Content-Type: application/json" -d '{"password": "123456789"}' -k

and now we just created the user and it’s role through kibana

A white background with blue text

AI-generated content may be incorrect.

Let’s re-edit the readonly user and give it more privileges and update the userA screenshot of a computer

AI-generated content may be incorrect.

Open the browser in-cognito to login with the user readonly A screenshot of a login screen

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

as you can see this role has readonly privileges, it cant make edits. You cant manage the indexes or use the devtools. A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

Roles gives properties to the user for what they can do or they cant do. It is a similar concept as in aws IAM roles policies.

# Realm in elasticsearch for authentication

You can add multiple roles authentication tools or realms with the elasticsearch. Native realms is the xpack.security

In **Elasticsearch**, a **realm** is a component used for **authentication and user lookup**. When a user tries to log in, Elasticsearch checks each configured realm in order to **authenticate** the user and retrieve their roles.

### 📌 **Definition**

A **realm** is a plugin or module that connects Elasticsearch to an external identity provider or source of user data (like LDAP, file-based users, or native users).

### 📊 **Table: Types of Realms in Elasticsearch**

| **Realm Type** | **Description** | **Common Use Case** | **Configuration Prefix** |
| --- | --- | --- | --- |
| native | Uses the internal Elasticsearch user store | Default for managing users via Kibana or API | xpack.security.authc.realms.native |
| file | Users and roles defined in local files (users and users\_roles) | Static, fallback user definitions | xpack.security.authc.realms.file |
| ldap | Connects to an LDAP directory like Active Directory | Centralized enterprise authentication | xpack.security.authc.realms.ldap |
| active\_directory | Special LDAP support for Windows AD | Best for Windows-based orgs with AD | xpack.security.authc.realms.active\_directory |
| pki | Uses client certificates for authentication | Environments with strong certificate-based auth | xpack.security.authc.realms.pki |
| saml | Uses SAML 2.0 identity providers | Single sign-on (SSO) via third-party IDPs (e.g., Okta, ADFS) | xpack.security.authc.realms.saml |
| oidc | OpenID Connect support | SSO using providers like Google, Azure, Keycloak | xpack.security.authc.realms.oidc |
| kerberos | Uses the Kerberos protocol | Environments with Kerberos tickets (common in Linux/Windows hybrid) | xpack.security.authc.realms.kerberos |
| custom | Custom-developed realms via plugin | Specialized enterprise systems | xpack.security.authc.realms.custom |

<https://www.elastic.co/docs/deploy-manage/users-roles/cluster-or-deployment-auth/authentication-realms>

## Example using configuring realms using LDAP:

<https://www.elastic.co/docs/deploy-manage/users-roles/cluster-or-deployment-auth/ldap#ldap-realm-configuration>

You have to mention this configuration on all the master nodes in order for you to be able to use it with the ldap

Side note: the url going to be your organization url, so you are going to use it according to your organization configurations.

A screenshot of a computer program

AI-generated content may be incorrect.

In older version of elasticsearch you could define the native realms, in new one it automatically selects the native realms so you don’t have to configure. (you can research how older version use to define the native xpack realms)

# On creating spaces:

There are multiple ways through which you can create the space Click on the space icon > click on manage space. Or you can search that in global search of kibana. Or you you can click sandwich menu > click on stack management > under Kibana click on spaces > click on Create space A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

## What is space?

Space concept is similar to have to how namespace works in Kubernetes. It is used to segregate or categorize the group of applications in a same group. Suppose tomorrow I am going to have 10 different project, I cant add those all projects in one space or else it will be difficult for me to manage them on granular level. To manage the project more finely, I will create a 10 different space so that I can keep them separate from each other.

## Let’s create a space:

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

As you can see there are two spaces created now. A screenshot of a computer

AI-generated content may be incorrect.

# 7 node cluster

A screenshot of a computer

AI-generated content may be incorrect.

A diagram of a network

AI-generated content may be incorrect.

Working with co-ordinator nodes

Node 1,2,3 above is a master node 3,4 below is a coordinator node, and 1,2 below are data nodes

A diagram of a diagram

AI-generated content may be incorrect.

# Instruction for removing elasticsearch cleanup

Currently, I am running 3 nodes elasticsearch and on node 1 I have both elasticsearch and kibana. We will provision 4 more nodes and we will clean up first 3 nodes:

To clean up elasticsearch we will have to remove all the directories log, config, service, remove the installation each and everything.

We will learn how each node will communicate with each other as well in next document:

Instructions for removing elasticsearch: (not this is not going to be your day to day practice)

1. **Stop Elasticsearch Service**  
   First, stop the Elasticsearch service if it is running.

sh

CopyEdit

sudo systemctl stop elasticsearch

1. **Uninstall Elasticsearch Package**  
   Use apt to remove the Elasticsearch package.

sh

CopyEdit

sudo apt-get remove --purge elasticsearch

The --purge option will remove the package and its configuration files.

1. **Remove Elasticsearch Directories**  
   You may also want to delete Elasticsearch directories, including data and logs. Be careful with this step, as it will remove all stored data.

sh

CopyEdit

sudo rm -rf /etc/elasticsearch

sudo rm -rf /var/lib/elasticsearch

sudo rm -rf /var/log/elasticsearch

1. **Remove Elasticsearch User and Group**  
   If you want to clean up the system further, you can remove the Elasticsearch user and group created during the installation.

sh

CopyEdit

sudo deluser elasticsearch

sudo delgroup elasticsearch

1. **Verify Removal**  
   Finally, you can verify that Elasticsearch has been removed by checking the service status.

All the major organizations use the sso, because they cannot create 100 of the users and the passwords each time. For that reason you have to provide a ldap realms configured in your elastic cluster so that it can authenticate form there. To enable the realms, you will have to configure it in the elsticsearch.yml file.