**Installing OpenSearch and OpenSearch Dashboard on Windows**

**Objectives**

1.clone dashboard repository.

2.create docker compose file .

**Prerequisites**

1. Ensure Git is installed on your system.
2. Install Node.js (latest stable version) and Yarn package manager.
   * Use the following commands for installation:



Steps

1. **Clone the OpenSearch Dashboard repository**:

git clone [git@github.com:opensearch-project/OpenSearch-Dashboards.git -b 2.18.0](mailto:git@github.com:opensearch-project/OpenSearch-Dashboards.git%20-b%202.18.0)

1. **Navigate to the cloned directory:**

cd OpenSearch-Dashboards

1. **Bootstrap the environment:**

yarn osd bootstrap

The osd bootstrap command will install the project's dependencies and build all internal packages and plugins. Bootstrapping is necessary any time you need to update packages, plugins, or dependencies, and it's recommended to run it anytime you sync with the latest upstream changes.

4.***Update corepack to the latest version***

npm i -g corepack

Corepack is a Node.js tool that acts as a **package manager manager**. It allows you to manage and switch between different Node.js package managers like **Yarn** without having to install them globally. Corepack is included in Node.js versions and above but needs to be enabled manually (or updated) via this command.

* npm: The Node.js package manager.
* i: Short for install, it installs a package.
* -g: Installs the package globally, making it accessible system-wide.
* corepack: The name of the package being installed.

5**. *Install the correct version of yarn***

corepack install

**Modify the OpenSearch Dashboard Configuration**

Edit the opensearch.yml file with the following settings:

**Opensearch\_dashbord.yaml**

# OpenSearch Dashboards is served by a backend server. This setting specifies the port to use.

server.port: 5611

# Specifies the address to which the OpenSearch Dashboards server will bind.

server.host: "localhost"

# The URLs of the OpenSearch instances to use for all your queries.

opensearch.hosts: ["https://localhost:9002"]

# If your OpenSearch is protected with basic authentication, provide the credentials here.

opensearch.username: "admin"

opensearch.password: "Infopercept@321"

# Optional setting to specify the path to the PEM file for the certificate authority for your OpenSearch instance.

# Uncomment and replace with the path if you're using custom certificates.

# opensearch.ssl.certificateAuthorities: [ "/path/to/your/CA.pem" ]

# To disregard the validity of SSL certificates, set this to 'none' (not recommended for production).

opensearch.ssl.verificationMode: none

**OpenSearch Setup with Docker**

**Steps**

1. **Create a directory named opensearch-docker:**

mkdir opensearch-docker

cd opensearch-docker

1. **Create a docker-compose.yml file in this directory with the following content:**

Docker-compose.yaml

version: '3'

services:

opensearch-node1:

image: opensearchproject/opensearch:latest

container\_name: opensearch-node1

environment:

- cluster.name=opensearch-cluster # Name the cluster

- node.name=opensearch-node1 # Name the node

- discovery.seed\_hosts=opensearch-node1 # Only this node in the setup

- cluster.initial\_cluster\_manager\_nodes=opensearch-node1 # Single-node manager

- bootstrap.memory\_lock=true # Disable JVM heap memory swapping

- "OPENSEARCH\_JAVA\_OPTS=-Xms512m -Xmx512m" # Set JVM heap size

- OPENSEARCH\_INITIAL\_ADMIN\_PASSWORD=Infopercept@321 # Set admin password

ulimits:

memlock:

soft: -1

hard: -1

nofile:

soft: 65536

hard: 65536

volumes:

- opensearch-data:/usr/share/opensearch/data # Persistent data storage

ports:

- 9002:9200 # Map container's 9200 port to host's 9002 port for REST API access

- 9600:9600 # Performance Analyzer (optional, can be removed)

networks:

- opensearch-net

volumes:

opensearch-data:

networks:

opensearch-net:

1. **Start the OpenSearch service:**

docker compose up -d

**Testing the Setup**

* OpenSearch Dashboard: Access the dashboard in your browser at http://localhost:5611.
* OpenSearch: Verify the API endpoint is running by navigating to <http://localhost:9002>