# **ECR Now Deployment and Keycloak Integration Guide**

**Version:** 3.0  
 **Last Updated:** July 2025

## **1. Prerequisites**

Ensure the following components are installed and configured on your system:

|  |  |  |
| --- | --- | --- |
| **Tool** | **Version (Minimum)** | **Description** |
| **Docker** | 20.10+ | Container runtime |
| **Docker Compose** | 3.8+ | Multi-container orchestration |
| **Keycloak** | 26.1.1+ | Identity and Access Management |

ℹ️ For Windows, ensure Docker Desktop has WSL integration enabled.

## **2. Keycloak Setup (Development Mode)**

Launch Keycloak using Docker:

docker run --name keycloak -p 9000:8080 \  
 -e KEYCLOAK\_ADMIN=admin \  
 -e KEYCLOAK\_ADMIN\_PASSWORD=admin \  
 quay.io/keycloak/keycloak:26.1.1 start-dev

* Access: [http://localhost:9000](http://localhost:9000/)
* Credentials:
  + **Username:** admin
  + **Password:** admin

## **🔐 3. Create Realm and Client in Keycloak**

### **🔹 Realm Creation**

1. Log into the Keycloak Admin Console.
2. Click **Create Realm**.
3. Set name as: ecrNow and click **Create**.

### **🔹 Client Setup**

1. Navigate to **Clients > Create**.
2. Fill:
   1. **Client ID**: ecrNowClient
   2. **Client Type**: OpenID Connect
   3. **Client Authentication**: Enabled
3. Save and navigate to the **Credentials** tab to **copy the Client Secret**.

### **🔹 Export Realm Configuration**

1. Navigate to **Realm Settings > Export**.
2. Enable:
   1. *Export Clients*
   2. *Export Roles*
3. Save as:  
    C:/keycloak-data/ecrNow-realm.json

## **📁 4. Application Directory Structure**

Ensure the following folder hierarchy under /home/ubuntu/ecr-now/:

/home/ubuntu/ecr-now:  
├── ecrNow.log  
├── eRSDv2\_specification\_bundle.json  
├── kars/  
├── bsa-output/  
├── custom-queries/

## **5. Configure and Integrate P12/JKS Keystore**

### **📄 Reference:**

Refer to :

[**Generating and Converting Certificates with OpenSSL and Keytool.pdf**](https://github.com/drajer-health/eCRNow/blob/new_cql_upgrade_rb/documents/eCRNowDeploymentsWithKeyCloak/Latest%20Generating%20and%20Converting%20Certificates%20with%20OpenSSL%20and%20Keytool.pdf)**📌 Keystore Location**

/home/ubuntu/ecr-now/secret/ecr-now.p12

### **🔒 Set Permissions**

chmod 640 /home/ubuntu/ecr-now/secret/ecr-now.p12

### **🛠 Update Docker Compose.yaml**

services:  
 ecr-now:  
 environment:  
 - jwks.keystore.location=/keys/ecr-now.p12  
 - jwks.keystore.password=<p12 keystore password>  
 volumes:  
 - /home/ubuntu/ecr-now/secret:/keys:ro

### **Verify the Keystore**

keytool -list -v   
 -keystore ecr-now.p12  
 -storetype PKCS12   
 -storepass [your-password]

## **6. Launch ECR Now via Docker**

From the root directory containing docker-compose.yml, run:

docker-compose up -d

Services Started:

* 🔸 PostgreSQL → postgres\_container
* 🔸 Keycloak → keycloak
* 🔸 Backend API → ecr-now
* 🔸 Frontend UI → ecrNow-ui

✅ Validate with:

docker ps

## **7. Access URLs**

|  |  |
| --- | --- |
| **Service** | **URL** |
| **Backend API** | [http://localhost:8081](http://localhost:8081/) |
| **Frontend UI** | [http://localhost:3000](http://localhost:3000/) |
| **Keycloak Console** | [http://localhost:9000](http://localhost:9000/) |

Ensure that:

REACT\_APP\_ECR\_BASE\_URL=http://localhost:8081

## **8. UI Configuration**

Once the UI is accessible:

📄 Refer to:  
 [**eCR Now App Configuration Guide\_Release3.0.docx**](https://github.com/drajer-health/eCRNow/blob/new_cql_upgrade_rb/documents/eCR%20Now%20App%20Configuration%20Guide_Release3.0.docx)

Update:

* Health Settings
* Public Health Authorities (PHA)

## **9. Troubleshooting**

### **🧾 Realm Import Fails**

* Ensure ecrNow-realm.json exists under C:/keycloak-data/
* Manually import if auto-import fails

### **PostgreSQL Errors**

* Double-check:
  + POSTGRES\_DB
  + POSTGRES\_USER
  + POSTGRES\_PASSWORD
* Restart the container:

bash

CopyEdit

docker restart postgres\_container

### **UI or API Not Loading**

* Backend: <http://localhost:8081>
* Frontend: <http://localhost:3000>
* Validate API in browser or Postman

### **🔐 Keycloak Issues**

* Confirm:
  + keycloak.auth.server
  + keycloak.client.id
  + keycloak.client.secret