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

## **Prerequisites**

Before running the application, ensure you have the following installed on your system:

* **Docker** (version 20.10+ recommended)
* **Docker Compose** (version 3.8+ recommended)
* **Keycloak** (version 26.1.1+ recommended)
* **Environment Variables File** (env.list)

## **Creating Keycloak Realm and Clients**

### **Run Keycloak in Development Mode**

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

### **Login to Keycloak Admin Console**

1. Open a browser and go to <http://localhost:9000>
2. Login with credentials:
   1. **Username**: admin
   2. **Password**: admin

### **Create a New Realm**

1. Click **Create Realm**
2. Set **Realm Name** as ecrNow and save

### **Create a Client**

1. Navigate to **Clients** and click **Create**
2. Set **Client ID** to ecrNowClient
3. Choose **Client Type** as **OpenID Connect**
4. Enable **Client authentication**
5. Save the client

### **Configure Client Credentials**

1. Under **Credentials** tab, copy the **Client Secret**

### **Export the Realm Configuration**

1. Go to **Realm Settings > Export**
2. Select **Export Clients and Roles**
3. Click **Export** and save the file as C:/keycloak-data/ecrNow-realm.json

## **Setting Up the Environment**

### **Prepare Keycloak Realm JSON File**

Place your ecrNow-realm.json file in the directory C:/keycloak-data/. This file is required for importing Keycloak realm settings into your instance.

### **Prepare Environment Variables**

Create a file named env.list at C:/ecr-now/. Add the required environment variables, such as database credentials, security keys, and Keycloak configurations.

### **Unzip ECR Now Application**

1. Unzip the ecr-now.zip file and place it in the C:/ drive.
2. Navigate to the env.list file inside the ecr-now folder and adjust the variables as shown below:

LOGGING\_FILE\_LOC=/c/ecr-now/ecrNow.log  
ESRD\_FILE\_LOC=/c/ecr-now/eRSDv2\_specification\_bundle.json  
SCHEMATRON\_FILE\_LOC=/c/ecr-now/schematron/CDAR2\_IG\_PHCASERPT\_R2\_STU1.1\_SCHEMATRON.sch  
SCHEMA\_FILE\_LOC=/c/ecr-now/schemas/CDA\_SDTC.xsd  
KAR\_DIRECTORY=/c/ecr-now/kars  
BSA\_OUTPUT\_DIRECTORY=/c/ecr-now/bsa-output/  
CUSTOM\_QUERY\_DIRECTORY=/c/ecr-now/custom-queries  
DATABASE\_URL=jdbc:postgresql://postgres\_container:5432/ecrnow

## **Running the Docker Containers**

### **Navigate to the Docker Compose File Location**

cd path/to/docker-compose.yml

### **Start the Services**

docker-compose up -d

The -d flag runs the containers in detached mode. This will start the following services:

* **PostgreSQL Database** (postgres)
* **Keycloak Authentication Service** (keycloak)
* **ECR Now Backend** (ecr-now)
* **ECR Now UI** (ecrNow-ui)

### **Verify Running Containers**

docker ps

Ensure all services are up and running.

## **Accessing the Services**

### **ECR Now Backend API**

* **URL**: <http://localhost:8081>
* **Authentication**: Uses Keycloak authentication

### **ECR Now UI**

* **URL**: <http://localhost:3000>
* Interacts with the backend via <http://ecr-now:8081>

## **Stopping the Services**

To stop all running containers:

docker-compose down

## **Troubleshooting**

1. **Keycloak Not Importing Realm Automatically**
   1. Ensure that ecrNow-realm.json exists in C:/keycloak-data/.
   2. Manually import the realm through the Keycloak Admin Console.
2. **Database Connection Issues**
   1. Verify that POSTGRES\_DB, POSTGRES\_PASSWORD, and jdbc.username/jdbc.password match.
   2. Restart PostgreSQL container:

docker restart postgres\_container

1. **UI Not Loading Correctly**
   1. Ensure the backend (ecr-now) is running and accessible at <http://localhost:8081>.
   2. Check that REACTS\_APP\_ECR\_BASE\_URL is correctly set.
2. **Keycloak Authentication Errors**
   1. Verify that keycloak.auth.server is set correctly.
   2. Ensure keycloak.client.id and keycloak.client.secret match the Keycloak settings.

This completes the setup and configuration guide for running ECR Now with Keycloak authentication