User Management Guide

This guide covers creating, managing, and organizing Cognito users for OpenSearch Dashboards and the Subscription Editor access.

Prerequisites

- AWS CLI v2.13+ installed and configured
- Stack deployed with dashboards enabled (Essential tier or above)
- IAM permissions: cognito-idp:Admin* actions on the User Pool
- Refer to readme.txt that accompanies cmd files, running each cmd without parameters provide usage.

Creating Users

Users are created via the users(.cmd/.sh) helper script. Each user receives a temporary password and must set a new one on first login.
Create the admin user first, it had root-level user permissions.

Syntax:

```
users.cmd <stack-name> create <email> [temp-password] [region]
```

Examples:

```
users.cmd LogProcessor create admin@example.com
users.cmd LogProcessor create viewer@example.com "TempPass99$!@!#$" us-east-1
```

If no temporary password is provided, a default is used. The user will be forced to change it on first login.

Other user actions:

```
users.cmd <stack-name> list                          -- List all users
users.cmd <stack-name> delete <email>                -- Delete a user
users.cmd <stack-name> reset <email> [password]      -- Reset password
users.cmd <stack-name> disable <email>               -- Disable (block login)
users.cmd <stack-name> enable <email>                -- Re-enable
```

Managing Groups

Groups let you organize users. Use the groups.cmd helper script.

Creating groups:

```
groups.cmd <stack-name> create-group admin "Full access editors"
groups.cmd <stack-name> create-group viewer "Read-only dashboard access"
```

Assigning users to groups:

```
groups.cmd <stack-name> add-to-group otheradmin@example.com admin
groups.cmd <stack-name> add-to-group analyst@example.com viewer
```

Removing users from groups:

```
groups.cmd <stack-name> remove-from-group analyst@example.com viewer
```

Listing:

```
groups.cmd <stack-name> list-groups                -- All groups
groups.cmd <stack-name> list-users                 -- All users
groups.cmd <stack-name> list-users admin           -- Users in a group
```

Deleting a group:

```
groups.cmd <stack-name> delete-group viewer
```

Subscription Editor Roles (RBAC)

The Subscription Editor enforces role-based access control using groups in advanced tier and above.

Supported group:

Role	Behavior
admin	Full access – can Save, Sync, Rollback, Import patterns, and edit all fields
viewer	Read-only – can view subscriptions and patterns but Save and Sync buttons are disabled. Server rejects write attempts.
(not set)	Access denied.

Important notes:

- Group changes take effect on next login (user must close browser/incognito and re-authenticate)
- ALB sessions last 1 hour – existing sessions retain the old role until expiry
- The editor displays the user's email and role in the header bar
- Enforcement is both client-side (disabled buttons) and server-side (403 Forbidden)

Recommended Setup

For a typical team deployment:

1. Create groups:

```
groups.cmd LogProcessor create-group admin "Full access - can edit subscriptions and patterns"
groups.cmd LogProcessor create-group viewer "Read-only dashboard access"
```

2. Create users:

```
users.cmd LogProcessor create admin@example.com
users.cmd LogProcessor create analyst@example.com
users.cmd LogProcessor create dba@example.com "Temp*88=888Pass123!@#!!!" us-east-1
```

3. Assign groups:

```
groups.cmd LogProcessor add-to-group otheradmin@example.com admin
groups.cmd LogProcessor add-to-group analyst@example.com viewer
groups.cmd LogProcessor add-to-group analyst@example.com viewer
```

Single Sign-on
Manage external SSO (SAML/OIDC) identity providers on the Cognito via the sso(.cmd/.sh) helper script
Requires: bash, AWS CLI, Cognito UserPool deployed (essential+ tier).

After adding an IdP, users see a "Sign in with <name>" button.
Assign groups after first login: groups.cmd <stack> add-to-group <email> admin

1. setup:
setup-saml: Add a SAML identity provider (Okta, ADFS, etc.)
setup-oidc: Add an OIDC identity provider (Azure AD, Google, etc.)

sso.cmd <stack-name> setup-saml <name> <metadata-url>
sso.cmd <stack-name> setup-oidc <name> <issuer> <client-id> <client-secret>

2. List:
List configured identity providers.

sso.cmd <stack-name> list

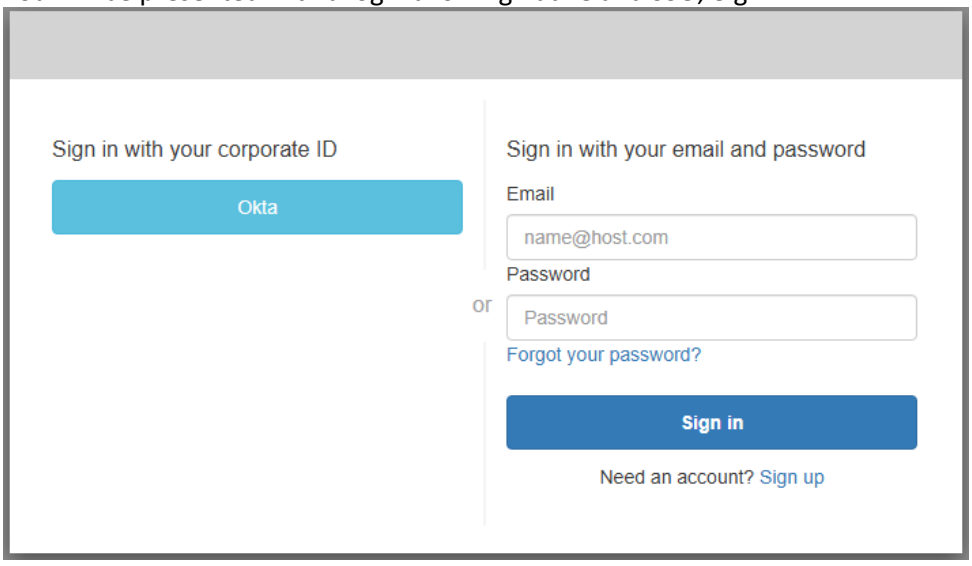
3. Remove:
Remove an identity provider.

sso.cmd <stack-name> remove <name>

3. Get Info:
Show ACS URL, Entity ID, and Redirect URI for IdP setup.

sso.cmd <stack-name> info

You will be presented with a login allowing native and SSO, e.g.



First Login

When a user logs in for the first time:

- Navigate to the Dashboards URL (from stack outputs or your custom domain)
- Enter the email and temporary password
- Set a new password (must meet policy: 12+ chars, uppercase, digit, symbol)
- If MFA is enabled (OPTIONAL or ON), configure an authenticator app
- Access is granted to OpenSearch Dashboards and the Subscription Editor

MFA Configuration

MFA is configured at the stack level via the DashboardsMfa parameter:

- **OFF** – No MFA required
- **OPTIONAL** – Users can enable MFA in their profile
- **ON** – MFA required for all users (TOTP only, no SMS)

Enterprise tier defaults to ON (required). Users configure MFA using any TOTP authenticator app (Google Authenticator, Authy, 1Password, etc.).

Password Policy

All user pools enforce:

- Minimum 12 characters
- At least one uppercase letter
- At least one digit
- At least one symbol

Troubleshooting

"User Pool not found" error:

- Ensure the stack has dashboards enabled (Essential tier or above)
- Verify the stack name is correct (case-sensitive)
- Check you have a valid ACM certificate configured

User cannot log in:

- Check user status with users.cmd <stack> list
- If status is FORCE_CHANGE_PASSWORD, the user hasn't completed first login
- If disabled, re-enable with users.cmd <stack> enable <email>
- Verify IP is in the DashboardsAllowedCidr range

Security Notes

- Self-signup is disabled – only administrators can create users
- All authentication flows through Cognito + ALB – no direct OpenSearch access
- Session tokens expire after 1 hour (Cognito default)
- Failed login attempts are throttled by Cognito automatically
- User activity is logged in Cognito advanced security (if enabled)