Keycloak Identity and Access Management Server

by Code Creator

Customer Instructions

Private, self-hosted identity and access management on AWS with automated first boot, generated administrator credentials, HTTPS access, and backup and restore helpers.

Software	Keycloak | PostgreSQL 18.4 | Nginx 1.30.4
Operating system	Ubuntu 26.04 LTS, AMD64
Validated first boot	2 minutes 48 seconds on t3.xlarge; allow up to 6 minutes

Purpose: Launch the AMI, retrieve the generated administrator credentials, open the Keycloak administration console, create realms and clients, and operate the server safely.

Private by design: The software runs inside your AWS account. The AMI does not contain Code Creator AWS access keys, shared administrator passwords, or a seller-hosted identity service.

Quick Start

Step	Action	Success
1	Launch with a public IPv4 address and allow TCP 22, 80, and 443.	EC2 status checks pass.
2	Connect with the Amazon private key selected at launch and SSH username **ubuntu**	SSH prompt opens.
3	Allow first boot to finish, then display FIRST_LOGIN.txt.	HTTPS URL and generated password appear.
4	Open the administration URL and accept the self-signed certificate warning.	Keycloak login page appears.
5	Sign in as keycloak-admin and create the first realm or client.	Administration console opens.

1. Product Overview

This AMI provides a self-hosted Keycloak identity and access management server inside your AWS account. It is designed for teams that need centralized authentication, single sign-on, standards-based application integration, user federation, multi-factor authentication, and administrative control without building the initial server stack manually.

The Code Creator package adds the AWS-ready Ubuntu image, Docker deployment, PostgreSQL database, Nginx HTTPS proxy, customer-specific first-boot automation, generated administrator credentials, operating helpers, and backup and restore tools.

Included capability	Customer value
Automated first boot	Generates unique administrator and database credentials, creates the HTTPS certificate, initializes PostgreSQL, and starts Keycloak automatically.
Keycloak administration	Manage realms, users, groups, roles, authentication flows, identity providers, and application clients.
Standards-based integration	Use OpenID Connect, OAuth 2.0, and SAML 2.0 with compatible applications and services.
Private PostgreSQL database	Database traffic remains on an internal Docker network and is not exposed on the EC2 host.
Operational helpers	Check status, view logs, restart services, display credentials, and create or restore backups.
AWS account ownership	The instance, data, network, backups, and AWS charges remain under the customer's AWS account.

Important architecture note: This release is a single-instance deployment. It is suitable for evaluation, development, internal services, and controlled production use where the customer accepts a single EC2 host. High availability requires additional architecture outside this AMI.

2. AWS Launch Requirements

Recommended EC2 configuration

Setting	Guidance
Instance type	t3.large recommended. t3.medium is the minimum evaluation size. t3.xlarge was used for final timing validation.
Storage	30 GiB gp3 root volume, or larger when retaining substantial audit, realm, or backup data.
Architecture	AMD64 / x86_64.
Public address	Enable a public IPv4 address for the validated public-IP access path.
IAM role	No IAM role is required for the validated single-instance configuration.
Key pair	Select an Amazon EC2 key pair that you control. Keep the private key secure.

Security group inbound rules

Protocol	Port	Source recommendation	Purpose
TCP	22	Your trusted IP or CIDR	SSH administration.
TCP	80	0.0.0.0/0 and ::/0 when IPv6 is used	Redirects browser traffic to HTTPS.
TCP	443	0.0.0.0/0 and ::/0 when IPv6 is used	Keycloak HTTPS access.

Do not expose internal ports: Keep TCP 5432, 8080, 8443, and 9000 closed in the EC2 security group. These ports are used internally by PostgreSQL and Keycloak.

3. First Boot and Administrator Login

Wait for first boot

First boot detects the new public IPv4 address, creates unique database and administrator passwords, generates a self-signed HTTPS certificate for the instance IP, initializes the database, creates the permanent Keycloak administrator, removes the temporary bootstrap administrator, and starts the Docker services.

The final AMI completed first boot in 2 minutes 48 seconds on a t3.xlarge instance. Allow up to 6 minutes on smaller or busier instances. Wait until both EC2 status checks pass before troubleshooting.

Do not interrupt first boot: Restarting or stopping the instance while provisioning is active can delay setup. The completion file is /var/lib/codecreator/keycloak-firstboot.done.

Connect by SSH

Use the Amazon private key selected during launch and the SSH username **ubuntu**. Replace the placeholders below with your key file and instance public IPv4 address.

```
chmod 400 /path/to/your-key.pem
```

Linux and macOS only. Windows permissions are handled differently.

```
ssh -i /path/to/your-key.pem ubuntu@PUBLIC_IP
```

Required SSH username: Always use ubuntu for this AMI. Do not use root, ec2-user, admin, or keycloak-admin for SSH.

Display the generated login information

```
cat /home/ubuntu/FIRST_LOGIN.txt
```

The file contains the HTTPS administration URL, instance public IP, permanent administrator username, generated password, and common server commands.

Protect the password: FIRST_LOGIN.txt contains administrator credentials. Store and share it securely. Do not paste the password into tickets, chat messages, public logs, or screenshots.

Open Keycloak in a browser

Open the administration URL shown in FIRST_LOGIN.txt. The format is:

```
https://PUBLIC_IP/admin/
```

The certificate is self-signed and tied to the public IPv4 address generated during first boot. Your browser will display a certificate warning. Review the address, choose the advanced option, and continue only when it matches your EC2 instance.

Sign in with:

- Username: **keycloak-admin**
- Password: the generated value in **/home/ubuntu/FIRST_LOGIN.txt**

Successful login opens the Keycloak administration console with the master realm available.

4. Initial Keycloak Setup

The AMI creates only the permanent server administrator and the default master realm. Create separate realms for applications and users rather than placing normal application users in the master realm.

1. Sign in to the administration console.
2. Open the realm selector and choose Create realm.
3. Enter a realm name and create the realm.
4. Create application clients using the protocol required by the application.
5. Create or federate users, groups, roles, and authentication policies.
6. Record client IDs, redirect URIs, issuer URLs, and secrets in your application's protected configuration.

> **Issuer format:** For a realm named example, the OpenID Connect issuer is https://PUBLIC_IP/realms/example.

Keycloak supports OpenID Connect, OAuth 2.0, SAML 2.0, identity brokering, LDAP and Active Directory federation, multi-factor authentication, WebAuthn, custom themes, role-based access, and configurable authentication flows. Configure only the capabilities required by your environment.

5. Server Operations

Run these commands through SSH as the ubuntu user. Commands that require elevated privileges include sudo.

Task	Command
Check service and first-boot status	`keycloak-status`
Display the generated administration credentials	`keycloak-credentials`
View recent product logs	`keycloak-logs`
Restart the full Keycloak stack safely	`sudo keycloak-restart`

Docker Compose location

The application files are stored under:

```
/opt/codecreator/keycloak
```

The protected environment file is:

```
/etc/codecreator/keycloak.env
```

Do not casually edit protected files: Incorrect changes to compose.yaml, the protected environment file, the reverse proxy, or generated certificates can prevent login or break token issuer URLs. Create a backup before material changes.

Restart behavior

Docker and the product containers start automatically after an EC2 reboot. The first-boot provisioning service is idempotent: after successful setup, it detects the completion marker and does not replace the customer's credentials or database.

6. Backup and Restore

Create a backup

```
sudo keycloak-backup
```

The helper creates a compressed archive under /opt/codecreator/keycloak/backups. The archive includes a PostgreSQL database dump, the matching administrator credential record, a manifest, and checksums.

Sensitive backup: The backup contains identity data and administrator information. Copy it to protected storage, restrict access, and follow your organization's retention and encryption requirements.

Restore a backup

```
sudo keycloak-restore /path/to/keycloak-backup.tar.gz
```

Restore replaces the current Keycloak database with the selected backup. The helper first creates an automatic safety backup, stops the application services, recreates the database, restores the data and matching administrator record, then starts and validates the runtime.

Restore interruption: Do not terminate the instance or interrupt the restore command. Confirm the server is healthy with keycloak-status after the restore finishes.

7. Security and Deployment Notes

- SSH uses the Amazon key pair selected at launch. Password-based SSH authentication is disabled.
- The Keycloak administrator password is generated uniquely on first boot. No shared default administrator password is included.
- PostgreSQL, Keycloak application, and Keycloak management ports remain internal to the instance.
- The validated public interface uses a self-signed certificate tied to the public IPv4 address.
- The software and customer data remain inside the customer's AWS account.
- The customer is responsible for AWS security groups, IAM, backups, monitoring, patch planning, data protection, and application-level Keycloak configuration.

Public IP stability

Important: The certificate and Keycloak issuer URLs are generated for the public IPv4 address detected during first boot. If that address changes after an EC2 stop/start, browser certificate validation and application issuer URLs may no longer match.

For longer-lived deployments, plan a stable endpoint before connecting applications. Common approaches include an Elastic IP for a stable IPv4 address or a separately engineered load balancer and managed certificate design. Changing the hostname, proxy, issuer, or TLS architecture is outside the validated public-IP quick-start configuration and should be tested before production use.

Updates

Test operating system, Keycloak, PostgreSQL, Nginx, Docker, and configuration updates before applying them to an important deployment. Create a backup first. Major application or image changes may require a new Code Creator AMI revision rather than an in-place modification.

8. Troubleshooting and Support

First boot is not complete

Check the service and log:

```
sudo systemctl status codecreator-keycloak-firstboot.service --no-pager
```

```
sudo tail -n 100 /var/log/codecreator/keycloak-firstboot.log
```

Normal first boot is usually complete within 3 to 6 minutes. Confirm the instance has a public IPv4 address, internet access to required Ubuntu services is available, Docker is active, and both EC2 status checks have passed.

Browser does not open

- Confirm TCP 80 and 443 are allowed in the EC2 security group.
- Confirm the browser URL uses the current public IPv4 address.
- Accept the self-signed certificate warning only after verifying the address.
- Run keycloak-status and keycloak-logs through SSH.

Administrator login fails

- Read the current credentials with keycloak-credentials or cat /home/ubuntu/FIRST_LOGIN.txt.
- Use the username keycloak-admin. Do not use the SSH username ubuntu in the browser.
- Check for copied spaces or line breaks in the password.
- Confirm the administration URL ends with /admin/.

Server health and restart

```
keycloak-status
```

```
keycloak-logs
```

```
sudo keycloak-restart
```

Official resources

Keycloak Documentation: https://www.keycloak.org/documentation

Keycloak Server Administration Guide: https://www.keycloak.org/docs/latest/server_admin/

Keycloak Release Notes: https://www.keycloak.org/docs/latest/release_notes/

AWS EC2 Linux SSH Guide: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/connect-linux-inst-ssh.html

Docker Compose Documentation: https://docs.docker.com/compose/