Authelia SSO Integration for Grafana and Gitea with Monitoring and Logging

Overview

This comprehensive guide provides step-by-step instructions for setting up Single Sign-On (SSO) integration using Authelia with Grafana and Gitea, including monitoring and logging capabilities. All configuration files and secrets are available in the GitHub repository for easy setup and cloning.

Repository: https://github.com/PrasadP744/sre-task.git

Prerequisites

- AWS account (free tier sufficient)
- Terraform (optional)
- Public-facing DNS with IP address
- Docker Compose with all configurations
- Basic understanding of Docker and containerization

Architecture Components

- **Authelia**: Authentication and authorization server
- **Grafana**: Monitoring and visualization platform
- **Gitea**: Git service with web interface
- **Nginx**: Reverse proxy and load balancer
- **Prometheus**: Metrics collection
- **Loki**: Log aggregation
- **Promtail**: Log shipping agent

Step 1: Initial Server Setup

EC2 Instance Configuration

Minimum Requirements:

- RAM: 1GB (recommended: add 2GB swap from storage)
- Storage: 20GB allocated for entire mount
- Instance Type: AWS free tier eligible

Note: 1GB RAM may cause performance issues, so adding swap space is highly recommended.

Docker Installation

Install Docker from the official website following the standard installation process for your operating system.

Terraform Provisioning (Optional)

If using Terraform for infrastructure provisioning, ensure you have:

- AWS CLI configured
- AWS Access Key ID and Secret Access Key generated

Required Ports

Configure security groups to allow the following ports:

Port	Protocol	Purpose
80	HTTP	Web traffic
443	HTTPS	Secure web traffic
22	SSH	Remote access

Repository Cloning

Clone the project repository on your server:

```bash
git clone https://github.com/PrasadP744/sre-task.git
cd sre-task
```

Step 2: DNS Configuration

DNS Provider Setup

Choose one of the following DNS providers:

- **Cloudflare** (Paid)
- **AWS Route 53** (Paid)
- **AWS Public DNS** (Free) - tested and working

Configuration Files to Update

Update your DNS name in the following files:

1. **docker-compose.yml**
2. **authelia/configuration.yml**
3. **.nginx.conf.d/default.conf**

Required URL Configurations

Update the following URLs according to your DNS name:

- Root URL
- Domain URL
- SSH domain
- OAuth URLs for Grafana and Gitea
- Subpaths where necessary

Step 3: Secret Generation and Configuration

Authelia Secrets Generation

Execute the provided script to generate required secrets:

```bash
./generate-password.sh
```

This script generates:

- `AUTHELIA_JWT_SECRET`
- `AUTHELIA_SESSION_SECRET`
- `AUTHELIA_STORAGE_ENCRYPTION_KEY`

Secret Placement

Update the generated secrets in:

- `docker-compose.yml`
- `authelia/configuration.yml`

User Password Hash Generation

Generate password hashes for users in `authelia/users_database.yml`:

```bash
# Generate password hashes for users (you need to install authelia CLI or use online generator)
echo "🔐 Note: You need to generate proper password hashes for users_database.yml"
echo "   Use: docker run --rm authelia/authelia:latest authelia crypto hash generate argon2 --p
```

Example command for user passwords:

```bash
docker run --rm authelia/authelia:latest authelia crypto hash generate argon2 --password 'admir
```

Where to update the generated hashes:

- **Authelia Users**: Update password hashes in `authelia/users_database.yml`
- **Grafana**: Client secret goes in `docker-compose.yml` under Grafana environment variables
- **Gitea**: Client secret goes in `docker-compose.yml` under Gitea environment variables
- **Authelia OIDC**: Client secrets go in `authelia/configuration.yml` under OIDC clients configuration

⚠️ **Important:** Save all secrets and passwords for future reference.

Step 4: HMAC and Private Key Generation

HMAC Secret Generation

Generate HMAC secret for `authelia/configuration.yml`:

```bash
docker run --rm authelia/authelia:latest authelia crypto rand --length 72 --charset alphanumeri
```

Private Key Generation

Generate private key for OIDC configuration in `authelia/configuration.yml`:

Option 1:

bash

```
authelia crypto pair rsa generate --bits 2048 --directory /path/to/keys/
```

Option 2:

bash

```
openssl genrsa -out oidc_private_key.pem 2048
```

Step 5: Client Secret Generation

Password Hash Generation

Generate password hashes for Grafana and Gitea client secrets:

bash

```
docker run --rm authelia/authelia:latest authelia crypto hash generate pbkdf2 --password 'admir
```

Important Notes:

- Each client can have only one secret
- Generate separate secrets for Grafana and Gitea
- Place client secrets in OIDC configuration for both services
- Update secrets in docker-compose file

⚠️ **Critical:** Save all passwords and secrets in a secure location.

Step 6: SSL Certificate Configuration

Certificate Requirements

Certificates must be generated with SAN (Subject Alternative Name) standard. Non-SAN certificates will cause errors.

Certificate Generation

Use the provided `cert.conf` file with your specific configurations:

bash

```
openssl req -x509 -newkey rsa:2048 -nodes -days 365 \
  -keyout privkey.pem -out fullchain.pem \
  -config cert.conf \
  -extensions v3_req
```

Certificate Placement

1. Place generated certificates in the `SSL` folder
2. Copy the same certificate `fullchain.pem` to the `certs` folder as `ca-certificates.crt`:

   bash

   ```
   cp SSL/fullchain.pem certs/ca-certificates.crt
   ```

3. Ensure DNS names match the certificate host configuration

Important: The `ca-certificates.crt` file enables Gitea to trust the certificate.

Step 7: Final Configuration

Authelia Configuration Updates

- Update DNS URLs in Authelia configuration
- Configure session timeout according to requirements
- Set token expiration as needed

Cluster Startup

Start your Docker cluster:

bash

```
docker-compose up -d
```

Step 8: Service Integration

Gitea OIDC Integration

Web UI Configuration Steps

1. **Access Gitea Web Interface**

- Navigate to your Gitea instance: `https://your-domain.com/gitea`
- Log in using generated credentials or create a new user account

2. **Navigate to Authentication Settings**
   - Go to **Site Administration** (admin user required)
   - Select **Authentication Sources**
   - Click **Add Authentication Source**

3. **Configure OAuth2/OIDC Provider**
   - **Authentication Type**: Select `OAuth2`
   - **Authentication Name**: `Authelia SSO` (or preferred name)
   - **OAuth2 Provider**: Select `OpenID Connect`

4. **Authelia Integration Settings**
   - **Client ID**: Use the client ID configured in `authelia/configuration.yml`
   - **Client Secret**: Use the client secret generated in Step 5
   - **OpenID Connect Auto Discovery URL**: `https://your-domain.com/api/oidc/.well-known/openid_configuration`
   - **Icon URL**: (optional) Add Authelia icon URL
   - **Scopes**: `openid profile email groups`

5. **Advanced Configuration**
   - **Claim Name**: `preferred_username` or `name`
   - **Claim Username**: `preferred_username`
   - **Claim Email**: `email`
   - **Claim Groups**: `groups` (if using group-based access)
   - **Group Claim Value**: Configure based on your Authelia group settings
   - **Admin Group**: Specify admin group name if applicable
   - **Restricted Group**: Configure restricted access groups if needed

6. **Additional Settings**
   - **Enable**: Check this box to activate the authentication source
   - **Synchronize Groups**: Enable if you want to sync groups from Authelia
   - **Skip Local 2FA**: Check if you want to skip local 2FA (since Authelia handles it)

7. **Test Configuration**
   - Click **Test** to verify the connection

- Ensure all settings are correct before saving
- Click **Add Authentication Source** to save

Post-Configuration Steps

1. **Verify SSO Login**
   - Log out of Gitea
   - Navigate to login page
   - You should see "Sign in with Authelia SSO" option
   - Test login with Authelia credentials
2. **User Management**
   - First-time users will be automatically created
   - Admin users can manage user permissions
   - Group memberships will sync based on Authelia configuration

Configuration Reference

For detailed Authelia configuration, refer to the Authelia documentation: **Gitea | Integration | Authelia**

Example Authelia OIDC Client Configuration:

```yaml
identity_providers:
  oidc:
    clients:
      - id: gitea
        description: Gitea
        secret: '$pbkdf2-sha512$your-generated-secret-hash'
        redirect_uris:
          - 'https://your-domain.com/gitea/user/oauth2/authelia/callback'
        scopes:
          - 'openid'
          - 'profile'
          - 'email'
          - 'groups'
```

Grafana Configuration

Grafana integration includes:

- Loki for log aggregation
- Promtail for log shipping
- Prometheus for metrics collection
- Basic and intermediate collection shipped to Grafana

Step 9: Monitoring and Dashboards

Grafana Dashboards

Recommended Dashboards:

- **Dashboard ID: 17802** - For Gitea monitoring
- **Custom Authelia Dashboard** - JSON file available in GitHub repository
- **Container Service Logs** - Choose appropriate dashboards based on requirements

Dashboard Configuration

Access Grafana web UI to:

- Import recommended dashboards
- Create custom dashboards
- Configure log and metric collections
- Set up alerting rules

Troubleshooting

Common Issues

1. **Certificate Errors**: Ensure SAN standard certificates are used
2. **DNS Resolution**: Verify DNS names match certificate configuration
3. **Memory Issues**: Add swap space if using minimal RAM
4. **Port Conflicts**: Ensure required ports are open and available

Verification Steps

1. Check all services are running: `docker-compose ps`
2. Verify certificate validity
3. Test DNS resolution
4. Confirm OIDC integration functionality

Security Considerations

- Store all secrets securely
- Use strong passwords for all accounts
- Regularly update certificates before expiration
- Monitor access logs for unusual activity
- Keep Docker images updated

Maintenance

Regular Tasks

- Monitor certificate expiration dates
- Review and rotate secrets periodically
- Update Docker images for security patches
- Backup configuration files and secrets
- Monitor system resources and performance

Backup Strategy

- Configuration files
- SSL certificates and keys
- User databases
- Docker volumes containing persistent data

Support and Resources

- **GitHub Repository**: https://github.com/PrasadP744/sre-task.git
- **Authelia Documentation**: Official Authelia docs
- **Docker Documentation**: Official Docker guides
- **Grafana Documentation**: Official Grafana resources

Conclusion

This guide provides a complete setup for Authelia SSO integration with Grafana and Gitea, including comprehensive monitoring and logging capabilities. Follow each step carefully and ensure all secrets are

properly secured. The provided GitHub repository contains all necessary configuration files and additional resources for successful deployment.