Harbor

Hashicorp Harbor Unleashed: A Complete Guide with Docker-Compose Installation

Hashicorp Harbor is an open-source cloud-native registry that secures, stores, and scans container images. Originally developed to enhance the security and management of containerized artifacts, Harbor is now a popular choice for organizations looking for a robust, secure, and scalable solution for managing their images and artifacts. In this article, we will explore Harbor's key features, its various use cases, and provide step-by-step Docker-Compose installation and basic setup instructions.

Key Features of Hashicorp Harbor

1. **Security and Vulnerability Scanning**

Harbor integrates with Clair and Trivy for security scanning, ensuring that container images do not contain known vulnerabilities. It can be configured to automatically scan images at intervals and before deployment, safeguarding your environment from malicious code or security flaws.

2. **Role-Based Access Control (RBAC)**

With Harbor, you can easily manage permissions for different users or teams. Administrators can assign specific roles to users or teams, controlling access to projects and images based on the organization’s needs.

3. **Replication**

Harbor allows the replication of images across multiple Harbor instances. This feature ensures redundancy and availability, useful in multi-data-center environments, or for teams collaborating across regions.

4. **Image Retention Policies**

Harbor supports automated image retention policies. This helps in managing storage efficiently by ensuring that old, unused images are purged while keeping critical ones available.

5. **Extensibility and Integration**

Harbor seamlessly integrates with Docker, Kubernetes, Helm, and other container orchestration systems. It also supports Webhooks, which allows for integration with CI/CD pipelines or other automation workflows.

6. **Audit Logging**

Every action performed within Harbor is logged. Audit logs allow administrators to monitor user activity, ensuring transparency and aiding compliance with organizational or regulatory requirements.

7. **Multi-Tenancy Support**

Harbor provides multi-tenant capabilities through project isolation. Each project can have its own set of users, permissions, and images, which is ideal for larger organizations or service providers managing different clients or teams.

8. **Notary for Image Signing**

Harbor integrates with Notary to support digital signing of container images. This feature guarantees the authenticity and integrity of images, ensuring that only trusted images are deployed.

9. **Interoperability**

Harbor is OCI-compliant, meaning it supports a broad range of cloud-native tools and technologies. It can work seamlessly with other registries and services, making it a versatile choice for managing containerized applications.

Use Cases for Hashicorp Harbor

1. **Enterprise-Scale Container Management**: Harbor is ideal for enterprises running multiple teams and applications that need centralized management of container images with strong security controls and governance.
2. **Multi-Region Deployment**: The replication feature ensures that images are available across various regions or data centers, reducing latency and improving reliability for distributed applications.
3. **Secure CI/CD Pipelines**: Harbor's integration with vulnerability scanners and image signing makes it the go-to choice for organizations with security-sensitive workflows, particularly when integrated into CI/CD pipelines.
4. **Private Cloud Solutions**: For organizations running on-premises infrastructure, Harbor offers a self-hosted solution for managing container images without relying on external public registries.

Installation Instructions Using Docker Compose

To install and configure Harbor using Docker Compose, follow these steps:

Prerequisites

1. Docker installed on your system.
2. Docker Compose installed.
3. Access to a domain name (optional, but recommended for SSL setup).

Step 1: Download Harbor

Next, download Harbor from the official GitHub repository:

```
wget https://github.com/goharbor/harbor/releases/download/v2.4.2/harbor-online-installer-v2.4.2.tgz
tar xzvf harbor-online-installer-v2.4.2.tgz
cd harbor
```

Step 2: Configure Harbor

Modify the `harbor.yml` file to suit your environment. This configuration file includes settings for domain names, database credentials, and other key settings.

```
vi harbor.yml
```

Key fields to configure:

- **Hostname**: The domain or IP where Harbor will be accessed.
- **HTTPS**: Enable if you plan to use SSL (strongly recommended).
- **Database**: Ensure that the PostgreSQL settings match your environment.

Step 3: Install and Start Harbor

Once the configuration is set, you can install Harbor using the provided installation script:

```
./install.sh
```

Once Harbor is installed, start the service using Docker Compose:

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

Step 4: Access Harbor

You can access Harbor by navigating to the domain or IP address you configured in `harbor.yml`. The default login credentials are:

- **Username**: `admin`
- **Password**: `Harbor12345`

After the first login, be sure to change the default password for security reasons.

Basic Setup Instructions

Step 1: Create a New Project

Once logged in, create a new project where you'll store your container images. A project is a logical grouping of container images that can be private or public, depending on your needs.

1. Navigate to **Projects** > **New Project**.
2. Name your project and configure its access level (public or private).
3. Click **OK**.

Step 2: Push and Pull Docker Images

To push an image to your Harbor instance, tag your Docker image with the Harbor domain and project name, then push it to the registry.

```
docker tag your-image:latest your-harbor-domain/project-name/your-image:latest
docker push your-harbor-domain/project-name/your-image:latest
```

To pull an image from Harbor, use the following:

```
docker pull your-harbor-domain/project-name/your-image:latest
```

Step 3: Set Up Vulnerability Scanning

Harbor integrates with Clair and Trivy for vulnerability scanning. To enable scanning:

1. Go to **Administration** > **Vulnerability Scanning**.
2. Select the scanner you want to use and configure scan policies.
3. Schedule automated scans or manually trigger a scan for specific projects or images.

Step 4: Integrate with CI/CD

Harbor supports Webhooks and other CI/CD tools, enabling seamless integration into your DevOps pipelines. You can set up Webhooks to trigger image builds, tests, or deployments based on specific events.

Conclusion

Hashicorp Harbor offers a powerful and secure solution for managing container images and artifacts. With features like vulnerability scanning, replication, and role-based access control, it is ideal for enterprises, small teams, or anyone looking to manage containers in a scalable and efficient way. Harbor integrates smoothly with a variety of DevOps tools and provides a seamless experience for securing and managing container images. With Docker Compose, setting up Harbor is straightforward and customizable, making it a must-have tool for containerized environments.

Revision #2
Created 2024-09-17 13:15:25 UTC by thesabear
Updated 2024-09-17 13:38:08 UTC by thesabear