

	update README...	27742d4 · 13 hours ago
	update README.md ...	2 months ago
	allow cross-compilin...	6 months ago
	fix: update metric de...	2 months ago
	Download static cUR...	last year
	Drop v3 metrics fro...	last month
	update helm release...	last year
	Update docs links th...	4 months ago
	Bump Go version in ...	2 months ago
	add all missing go.m...	2 years ago
	fix: information discl...	last year
	Migrate golanglint-ci...	9 months ago
	Replace Minio refs in...	6 years ago
	Fix evaluation of Ne...	9 months ago
	add CNAME for chart...	4 years ago
	cleanup markdown ...	3 years ago
	chore: Adjust setup ...	last year
	update dependencie...	9 months ago
	update scripts pointi...	2 months ago
	simplify dockerfiles a...	4 years ago
	move to go1.24 (#21...	8 months ago
	move to go1.24 (#21...	8 months ago
	move to go1.24 (#21...	8 months ago
	fix: add Dockerfile.re...	4 years ago
	update license chan...	4 years ago
	update with newer p...	8 months ago
	update NOTICE to 20...	2 years ago
	add pull requests eti...	7 months ago
	update README.md ...	13 hours ago
	cleanup markdown ...	3 years ago
	cleanup markdown ...	3 years ago
	Set theme jekyll-the...	4 years ago
	cleanup unneeded fi...	6 years ago
	update scripts pointi...	2 months ago
	Bump Go version in ...	2 months ago
	Use new gofumpt (#...	3 months ago
	Add MinIO server he...	4 years ago
	update helm release...	last year
	Use go mod tool to i...	8 months ago
	add community cont...	2 years ago

Maintenance Mode

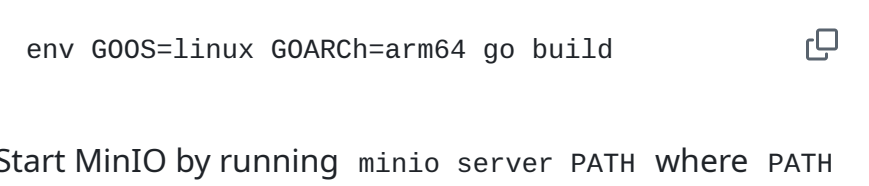
This project is currently under maintenance and is not accepting new changes.

- The codebase is in a maintenance-only state
- No new features, enhancements, or pull requests will be accepted
- Critical security fixes may be evaluated on a case-by-case basis
- Existing issues and pull requests will not be actively reviewed
- Community support continues on a best-effort basis through [Slack](#)

For enterprise support and actively maintained versions, please see [MinIO AIStor](#).

MinIO Quickstart Guide

slack channel 33744docker pulls 1.8Glicense AGPL V3



MinIO is a high-performance, S3-compatible object storage solution released under the GNU AGPL v3.0 license. Designed for speed and scalability, it powers AI/ML, analytics, and data-intensive workloads with industry-leading performance.

- S3 API Compatible – Seamless integration with existing S3 tools
- Built for AI & Analytics – Optimized for large-scale data pipelines
- High Performance – Ideal for demanding storage workloads.

This README provides instructions for building MinIO from source and deploying onto baremetal hardware. Use the [MinIO Documentation](#) project to build and host a local copy of the documentation.

MinIO is Open Source Software

We designed MinIO as Open Source software for the Open Source software community. We encourage the community to remix, redesign, and reshape MinIO under the terms of the AGPLv3 license.

All usage of MinIO in your application stack requires validation against AGPLv3 obligations, which include but are not limited to the release of modified code to the community from which you have benefited. Any commercial/proprietary usage of the AGPLv3 software, including repackaging or reselling services/features, is done at your own risk.

The AGPLv3 provides no obligation by any party to support, maintain, or warranty the original or any modified work. All support is provided on a best-effort basis through Github and our [Slack](#) channel, and any member of the community is welcome to contribute and assist others in their usage of the software.

MinIO [AIStor](#) includes enterprise-grade support and licensing for workloads which require commercial or proprietary usage and production-level SLA/SLO-backed support. For more information, [reach out for a quote](#).

Source-Only Distribution

Important: The MinIO community edition is now distributed as source code only. We will no longer provide pre-compiled binary releases for the community version.

Installing Latest MinIO Community Edition

To use MinIO community edition, you have two options:

1. **Install from source** using `go install github.com/minio/minio@latest` (recommended)
2. **Build a Docker image** from the provided Dockerfile

See the sections below for detailed instructions on each method.

Legacy Binary Releases

Historical pre-compiled binary releases remain available for reference but are no longer maintained:

- GitHub Releases: <https://github.com/minio/minio/releases>
- Direct downloads: <https://dl.min.io/server/minio/release/>

These legacy binaries will not receive updates. We strongly recommend using source builds for access to the latest features, bug fixes, and security updates.

Install from Source

Use the following commands to compile and run a standalone MinIO server from source. If you do not have a working Golang environment, please follow [How to install Golang](#). Minimum version required is [go1.24](#)

```
go install github.com/minio/minio@latest
```

You can alternatively run `go build` and use the `GOOS` and `GOARCH` environment variables to control the OS and architecture target. For example:

```
env GOOS=linux GOARCH=arm64 go build
```

Start MinIO by running `minio server PATH` where `PATH` is any empty folder on your local filesystem.

The MinIO deployment starts using default root credentials `minioadmin:minioadmin`. You can test the deployment using the MinIO Console, an embedded web-based object browser built into MinIO Server. Point a web browser running on the host machine to <http://127.0.0.1:9000> and log in with the root credentials. You can use the Browser to create buckets, upload objects, and browse the contents of the MinIO server.

You can also connect using any S3-compatible tool, such as the MinIO Client `mc` commandline tool:

```
mc alias set local http://localhost:9000 minio:
mc admin info local
```

See [Test using MinIO Client mc](#) for more information on using the `mc` commandline tool. For application developers, see <https://docs.min.io/enterprise/aistor-object-store/developers/sdk/> to view MinIO SDKs for supported languages.

Note

Production environments using compiled-from-source MinIO binaries do so at their own risk. The AGPLv3 license provides no warranties nor liabilities for any such usage.

Build Docker Image

You can use the `docker build` command to build a Docker image on your local host machine. You must first [build MinIO](#) and ensure the `minio` binary exists in the project root.

The following command builds the Docker image using the default `Dockerfile` in the root project directory with the repository and image tag `myminio:minio`

```
docker build -t myminio:minio .
```

Use `docker image ls` to confirm the image exists in your local repository. You can run the server using standard Docker invocation:

```
docker run -p 9000:9000 -p 9001:9001 myminio:m
```

Complete documentation for building Docker containers, managing custom images, or loading images into orchestration platforms is out of scope for this documentation. You can modify the `Dockerfile` and `dockerscripts/docker-entrypoint.sh` as-needed to reflect your specific image requirements.

See the [MinIO Container](#) documentation for more guidance on running MinIO within a Container image.

Install using Helm Charts

There are two paths for installing MinIO onto Kubernetes infrastructure:

- Use the [MinIO Operator](#)
- Use the community-maintained [Helm charts](#)

See the [MinIO Documentation](#) for guidance on deploying using the Operator. The Community Helm chart has instructions in the folder-level README.

Test MinIO Connectivity

About

MinIO is a high-performance, S3 compatible object store, open sourced under GNU AGPLv3 license.

[min.io/download?license=agp...](#)

go kubernetes cloud
storage s3 k8s
cloudnative objectstorage
multi-cloud cloudstorage
amazon-s3
multi-cloud-kubernetes

- Readme
- AGPL-3.0 license
- Code of conduct
- Contributing
- Security policy
- Activity
- Custom properties
- 58.9k stars
- 636 watching
- 6.5k forks

Report repository

Releases 521

Security/CVE RELEAS... Latest on Oct 16

+ 520 releases

Used by 405

 + 397

Contributors 506



+ 492 contributors

Languages

Go 99.0% Other 1.0%

Test using MinIO Console

MinIO Server comes with an embedded web based object browser. Point your web browser to <http://127.0.0.1:9000> to ensure your server has started successfully.

Note

MinIO runs console on random port by default, if you wish to choose a specific port use `--console-address` to pick a specific interface and port.

Test using MinIO Client `mc`

`mc` provides a modern alternative to UNIX commands like `ls`, `cat`, `cp`, `mirror`, `diff` etc. It supports filesystems and Amazon S3 compatible cloud storage services.

The following commands set a local alias, validate the server information, create a bucket, copy data to that bucket, and list the contents of the bucket.

```
mc alias set local http://localhost:9000 minio:   
mc admin info  
mc mb data  
mc cp ~/Downloads/mydata data/  
mc ls data/
```

Follow the MinIO Client [Quickstart Guide](#) for further instructions.

Explore Further

- [The MinIO documentation website](#)
- [MinIO Erasure Code Overview](#)
- [Use `mc` with MinIO Server](#)
- [Use `minio-go` SDK with MinIO Server](#)

Contribute to MinIO Project

Please follow MinIO [Contributor's Guide](#) for guidance on making new contributions to the repository.

License

- MinIO source is licensed under the [GNU AGPLv3](#).
- MinIO [documentation](#) is licensed under [CC BY 4.0](#).
- [License Compliance](#)

