



# Podman and MinIO

Will Dinyes  
6 Dec 2022

# Agenda

- MinIO Object Storage
- Get to the POC
- Secret Sauce
- Questions



# MinIO Enterprise Grade Feature Set



## S3 Compatible

Amazon S3 API is the de facto standard for object storage. MinIO implements Amazon S3 v2/v4 API.



## Object Locking

Ensures that an object cannot be deleted or overwritten for a set period of time. Certified by Cohasset Associates for SEC 17a-4(f), FINRA 4511(c) and CFTC 1.31(c)-(d)



## Single Sign-on

MinIO Server integrates with Identity Providers such as WSO2, Keycloak, Okta, Ping Identity to allow applications or users to authenticate and use Object Storage.



## Encryption & Tamper-proof

MinIO provides confidentiality, integrity and authenticity assurances for encrypted data with negligible performance overhead.



## Lambda Compute

MinIO server triggers Lambda functions through event notification service. OCR, audit compliance are good examples of lambda computing.



## Erasure Code & Bitrot Protection

You may lose up to half the number of drives and still recover from it. Data protection code is accelerated using SIMD instructions on x64 and ARM CPUs.



## Object Lifecycle Management

Automate data lifecycle management activities such as changing tiers (NVMe -> HDD), updating policies, transitioning and deleting data.



## Server Side Bucket Replication

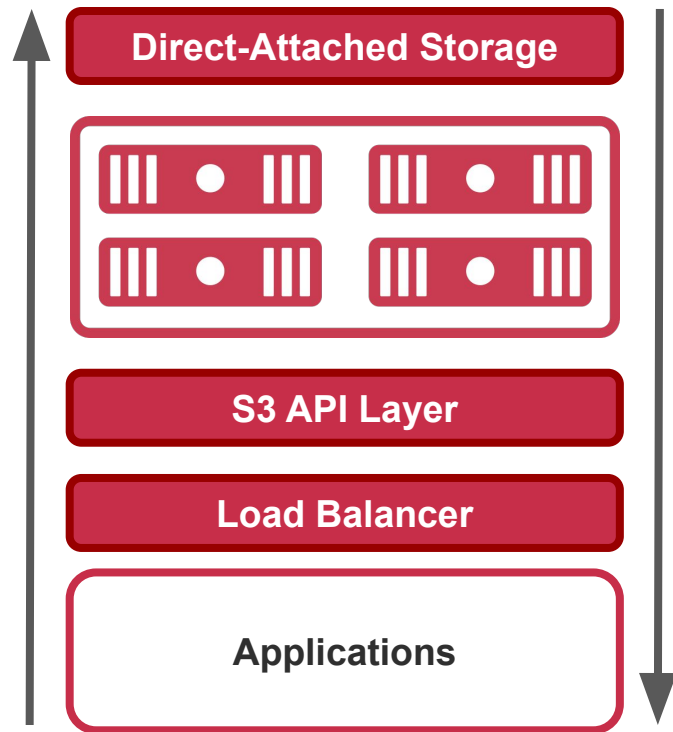
Works in Active-Active and Active-Passive setup enabling automated application failover. Supports object locking/retention across source and destination buckets.

# Topology of a Deployment

A production MinIO deployment consists at least 4 MinIO hosts pooling compute and storage resources into a single aggregated object storage resource.

Each MinIO server has a complete picture of the distributed topology, such that an application can connect to any node in the deployment and perform S3 operations.

Applications typically interact with a load balancer using a "Least Connections" methodology for routing application requests to MinIO hosts



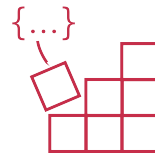
# MinIO Deployment Use Cases



Big Data/Machine Learning  
environments



HDFS replacements



High performance data  
lake/warehouse infrastructure



Cloud native applications  
(replacing file + block)



Multi-cloud environments  
(portability)



Endpoint for streaming  
workloads

**By deploying for performance, long term archival storage and disaster recovery are effectively free.**



# Everything All At Once

# Get to the POC

Rapid development cycles mean getting that *POC* done fast ...

- So you can get the *MVP* done fast ...
  - So you can get the *update* fast ...

# Challenge: Cloud POC on a hacker budget

Do you want to pay Amazon, or Google, or ...

Until you can deploy?

NO!

But you're going to the cloud eventually, right?



# Start as you intend to Finish

MinIO lets you deploy a fully S3 compatible environment...

- Locally!
- And Open Source!
- Licensing and cloud migration can come later, if your POC pans out
- ***WITHOUT CHANGING A LINE OF CODE!***



# Podman: your secret weapon

# MinIO and Podman make it simple

Don't spend too much time deploying MinIO for a POC

Use a container (of course!!!)

Single drive, single server is great for development

Podman and MinIO work great together!

# The Secret Sauce

MinIO is for your data storage, all objects all the time

Podman lets you mount volumes (even if you're not developing on Linux)

Keep your data safe between updates

Make the updates simple with a container

# — Here's the hard way

Podman on Mac is fast, simple, easy

Podman Desktop makes it even easier

But, what about the data?

# Get the Engine Right

Tell the Podman Machine where your data will be ...

```
podman machine init -v /local/data:/Minio/data
```

**/local/data** is on your host system

**/Minio/data** is created in the podman machine

# Fire Up The Container

```
podman run \  
  -p 9000:9000 \  
  -p 9090:9090 \  
  --name minio \  
  -v /Minio/data:/data \  
  -e "MINIO_ROOT_USER=admin" \  
  -e "MINIO_ROOT_PASSWORD=ifyouusethispasswordeveryonewilllaughatyou" \  
  quay.io/minio/minio server /data --console-address ":9090"
```

# Time is not on your side...

MinIO is very time sensitive

If your Mac goes to sleep, your Podman machine will have a time slip

Just remember to restart both the Machine and the Container if you see:

The difference between the request time and the server's time is too large.



# ...but everything else is!

Write your code to S3 API but connect to your local container

Your data is safely exfiltrated so it survives the container reaping/updates

- MinIO does plenty of updates
- As does Podman

When you're ready, move to S3 without changing your code

- *Or, even better, deploy MinIO on your cloud provider of choice while still using an S3 compatible object storage layer!*



Thanks for attending!

Questions?