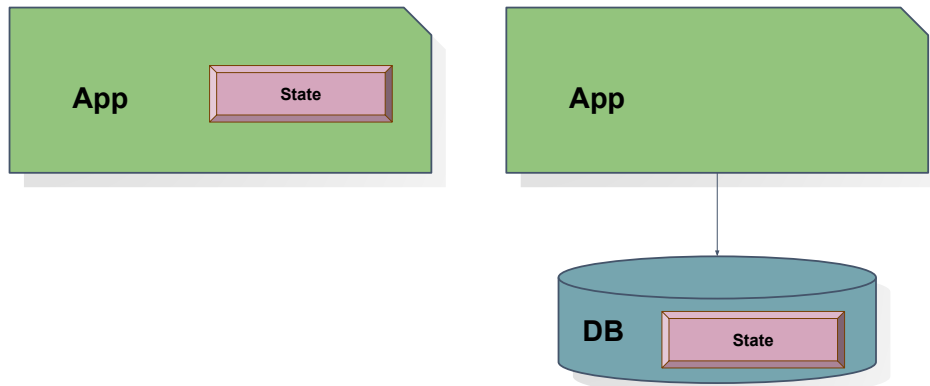


# Docker Volumes





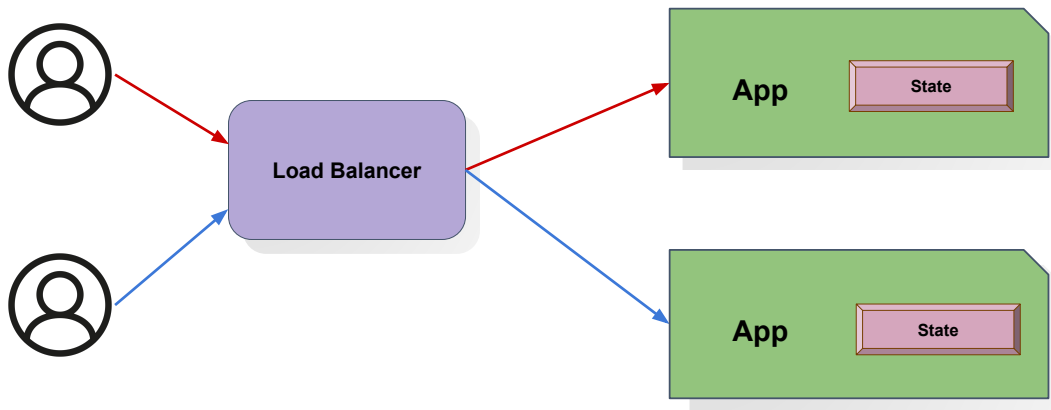
- **Statefulness**
- Types of Volumes
- Volume Plugins
- Demo



The first thing we should go over is stateful vs stateless applications. The state of a system is the truth in the system, like who is logged in, database data, where players are in a game world, etc. If this data is lost, then the system will be in a different state which may cause issues such as users getting logged out, orders being lost, or worse. A stateful application is one that has this data local to the app, such as in memory or on disk. A stateless app instead doesn't rely on state or refers to it elsewhere.

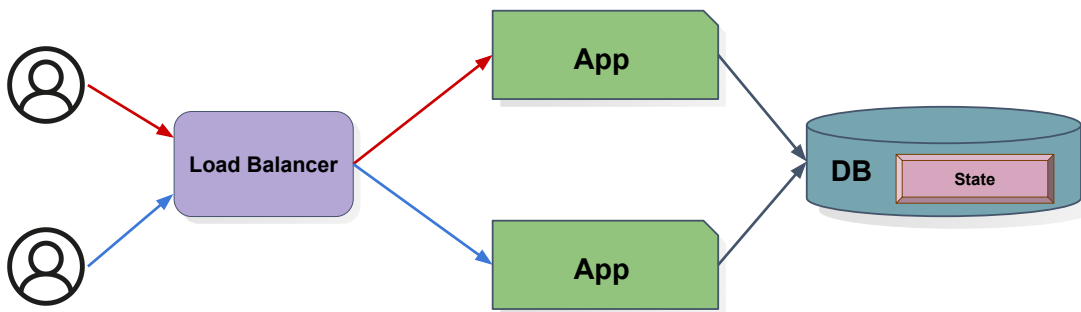
Classic examples of a stateful app:

- \* Databases
- \* CI/CD Systems
- \* CRM and content management apps



The problem with colocated state is that if we want to scale our application horizontally, as in adding more instances of our app, then we must either send a user to the same instance each time (depending on the app) or we cannot scale out the application at all. We also have a problem if the application crashes or becomes unavailable as we have impacted any users relying on its state. This is especially critical with containers because any information on local disk will be gone when the pod terminates.

If we horizontally scale a stateful application, we must use 'sticky sessions' or 'session affinity' to ensure our users witness a consistent state. This can create hotspots as well as single points of failure that guarantee that some percentage of users would be impacted by an outage.



It's much better to have a stateless app. This app may still depend on some state data but it does not live local to the app. Perhaps it is broken out into a database. This allows us to have multiple instances of the application running in different locations, all referring to the same state.

If we can scale our application horizontally, then we can achieve greater availability, scalability, and we may even be able to achieve no-downtime updates. If we lose an instance of the app, no big deal, make a new one. If we need to handle more workload, just increase capacity by having more pods.



- Statefulness
- **Types of Volumes**
- Volume Plugins
- Demo

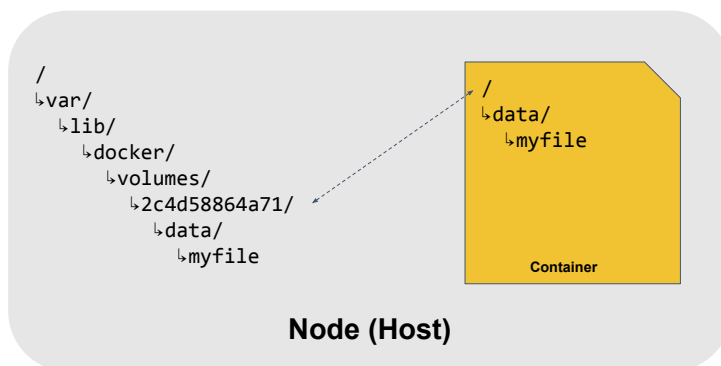


## Types of Volumes

- per container (ephemeral)
- shared storage (same host, persistent)
- multi-host shared storage (persistent)



```
$ docker run container  
  
$ #Inside Container  
$ mkdir /data  
$ touch /data/myfile
```

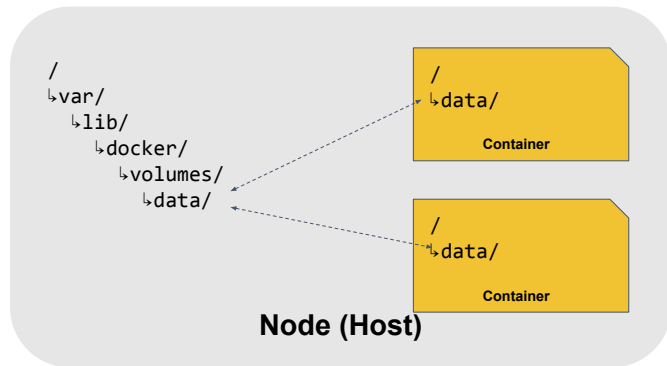


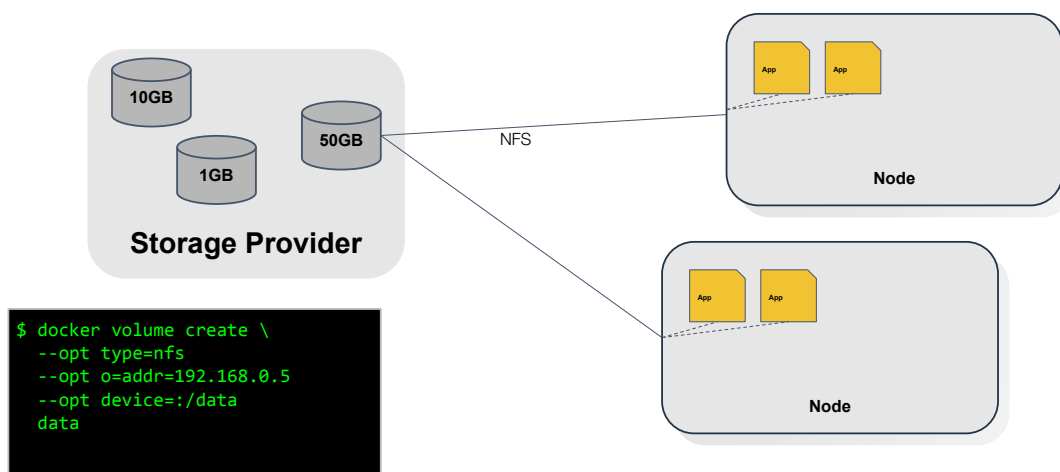




## Shared Storage (same host)

```
$ docker volume create data  
$ docker run container \  
-v data:/data  
$ docker run container \  
-v data:/data
```







- Statefulness
- Types of Volumes
- **Volume Plugins**
- Demo

[https://docs.docker.com/engine/extend/legacy\\_plugins/#volume-plugins](https://docs.docker.com/engine/extend/legacy_plugins/#volume-plugins)



# Using Volumes



## Additional Resources



- Docker Storage video - [https://www.youtube.com/watch?v=y5wMbA\\_T0tA](https://www.youtube.com/watch?v=y5wMbA_T0tA)
- List of volume plugins - [https://docs.docker.com/engine/extend/legacy\\_plugins/#volume-plugins](https://docs.docker.com/engine/extend/legacy_plugins/#volume-plugins)