

# docker

## The Customization of Docker from a Containerbie

Evelyn Xu    [evelyn.xu@daocloud.io](mailto:evelyn.xu@daocloud.io)

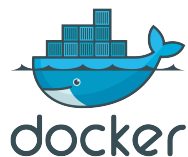
<http://blog.daocloud.io/dockerby-evelyn/>

# Content

---



- Problem Statement
- The Challenge
- The Thought Process
- Source Code Travel
- Conclusion



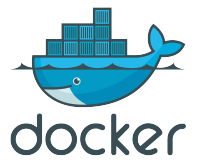
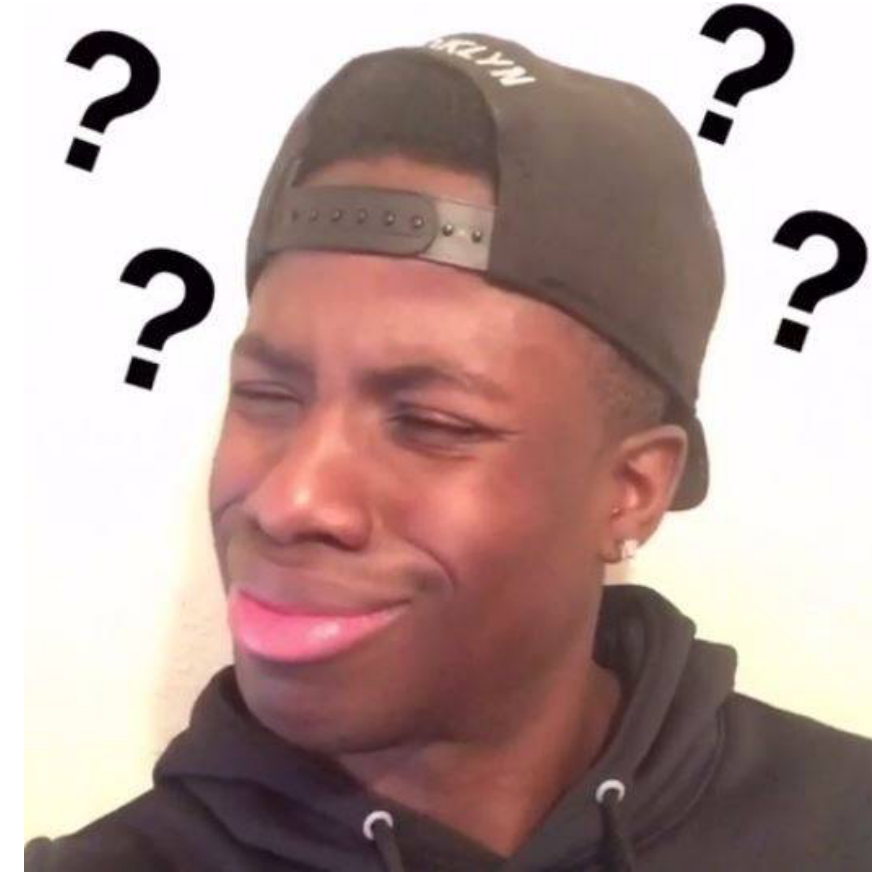
# Problem Statement

---

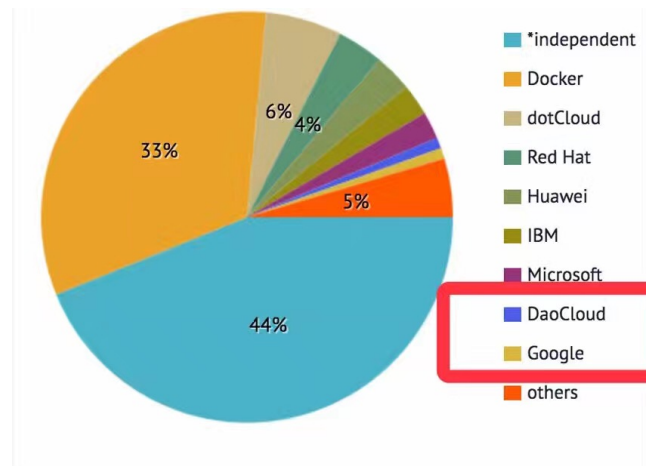
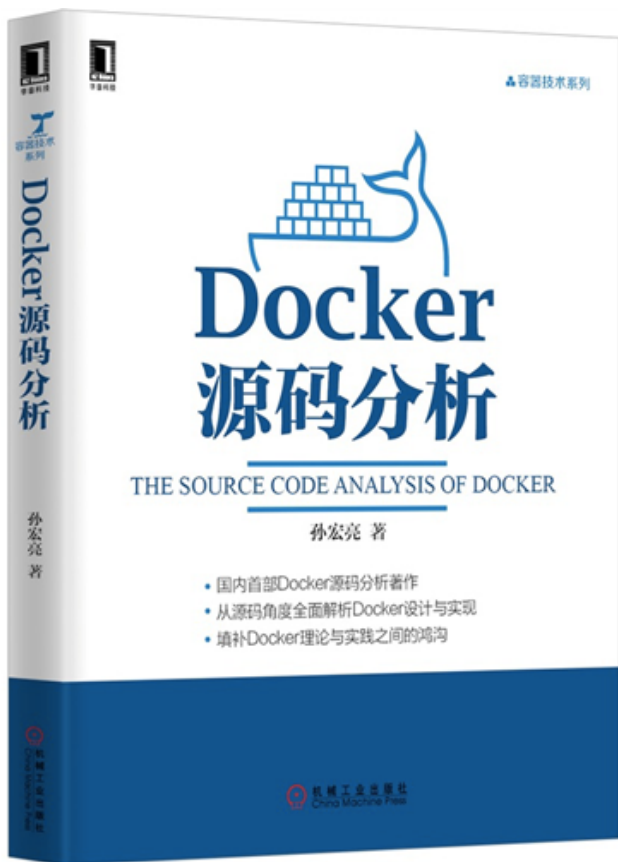


Allen :

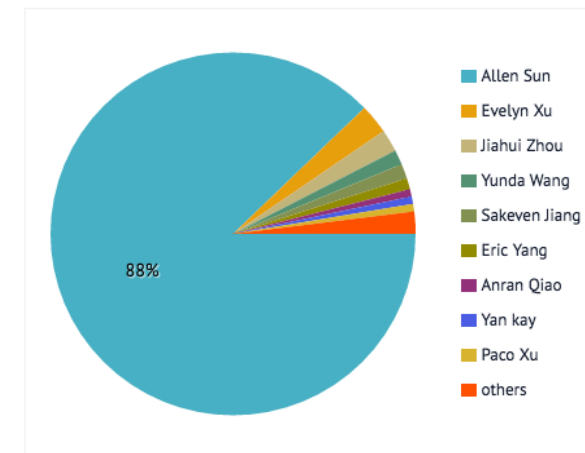
Try to **customize** the Docker, restrict the number of volume created by each Daemon !!!



- DaoCloud Tech Partner
- Author of 《 Docker Source Code Analysis》
- Docker Global Maintainer



Contribution by contributors



Show 10 entries Search:

| # | Company         | Commits |
|---|-----------------|---------|
|   | *independent    |         |
| 1 | Docker          |         |
| 2 | dotCloud        |         |
| 3 | Red Hat         |         |
| 4 | Huawei          |         |
| 5 | IBM             |         |
| 6 | Microsoft       | 745     |
| 7 | DaoCloud        | 296     |
| 8 | Google          | 295     |
| 9 | ZTE Corporation | 271     |

公司名称：DaoCloud  
全球排名：7  
Commits：296

Show 10 entries Search:

| #  | Contributor   | Commits |
|----|---------------|---------|
| 1  | Allen Sun     | 266     |
| 2  | Evelyn Xu     |         |
| 3  | Jiahui Zhou   |         |
| 4  | Yunda Wang    |         |
| 5  | Sakeven Jiang |         |
| 6  | Eric Yang     |         |
| 7  | Anran Qiao    | 2       |
| 8  | Yan kay       | 2       |
| 9  | Paco Xu       | 2       |
| 10 | Zhenglun Xu   | 2       |

Maintainer：Allen Sun  
Commits：256

Showing 1 to 10 of 14 entries First Previous 1 2 Next Last

# The Challenge



What is Linux Container ?

---



What is Docker ?

---



Docker vs Virtual Machine

---

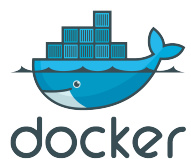
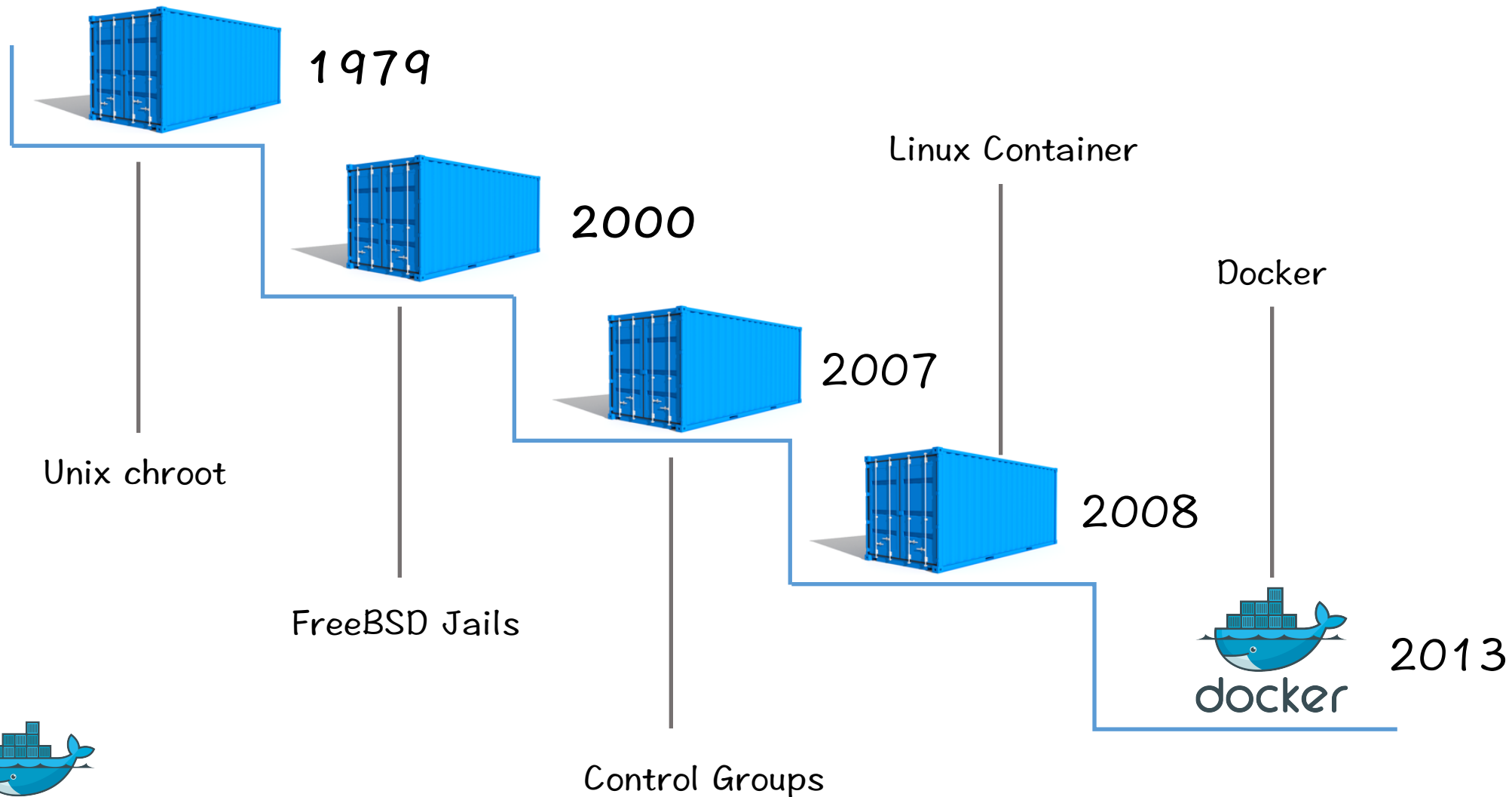


What is Volume ?

---



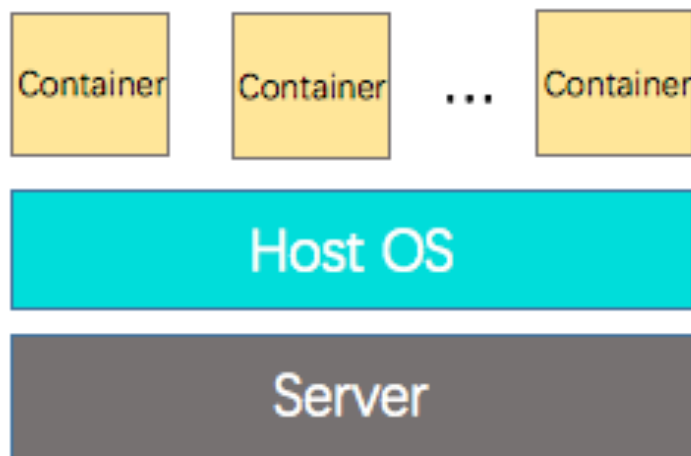
# The Challenge



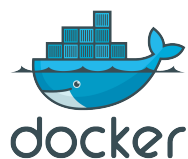
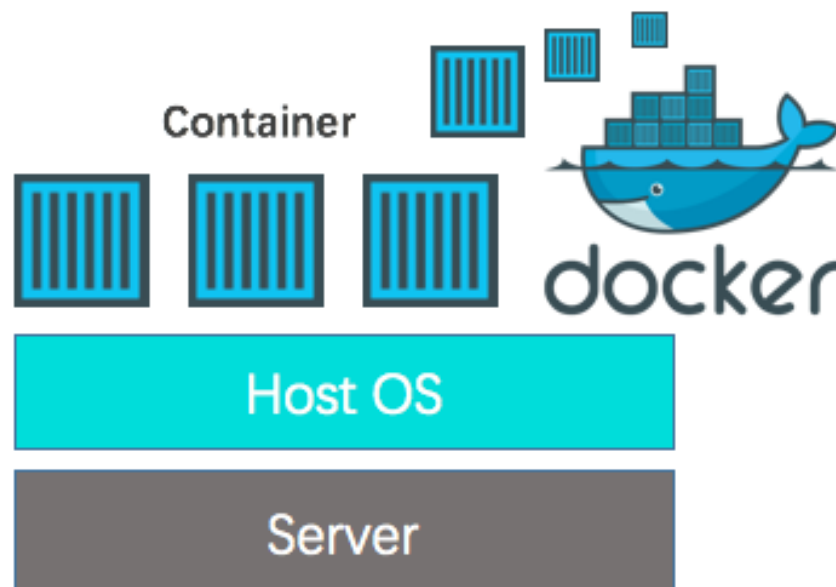
# The Thought Process



Container :



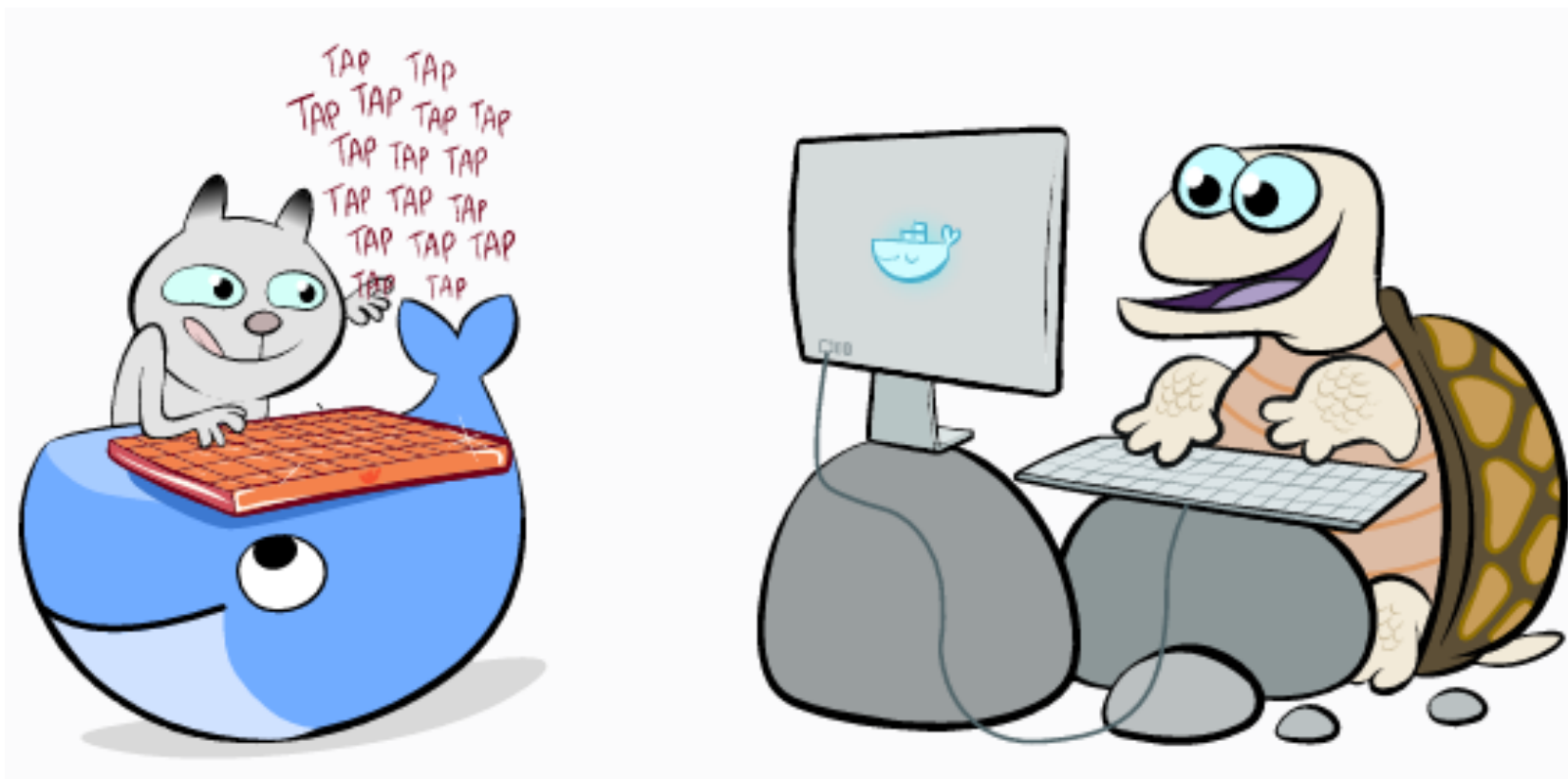
Docker :



# The Thought Process



Docker – Build, Ship and Run Any APP, Anywhere



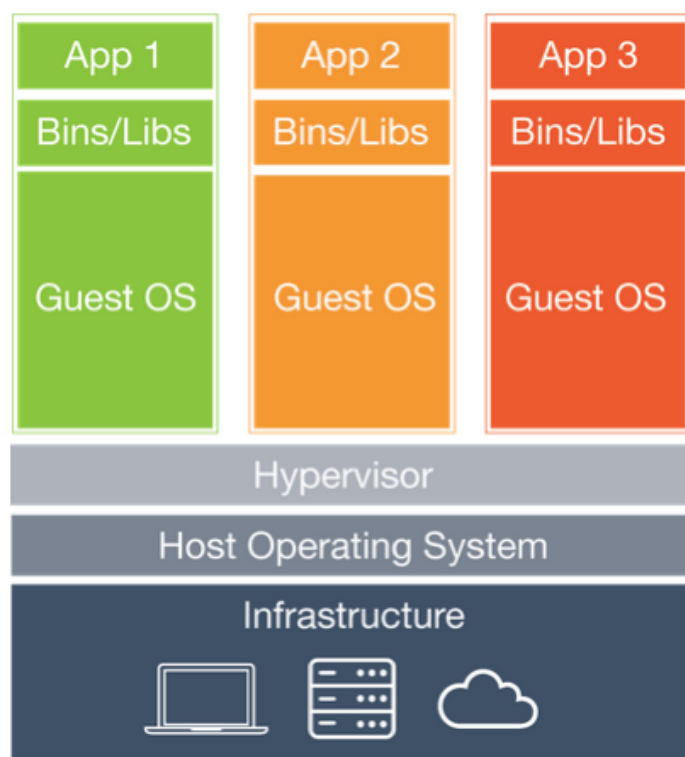


# The Thought Process

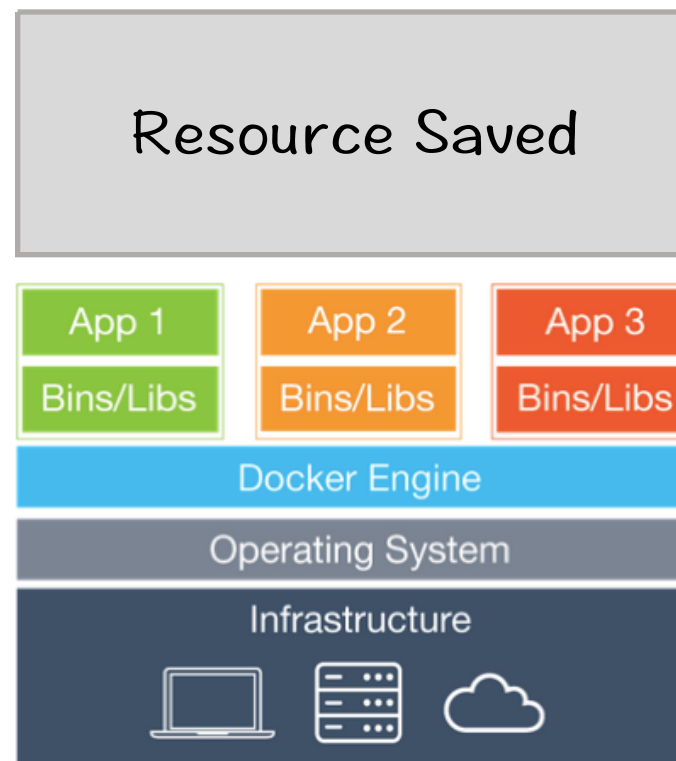


## Docker vs Virtual Machine

VM



Resource Saved



Docker





docker



DaoCloud



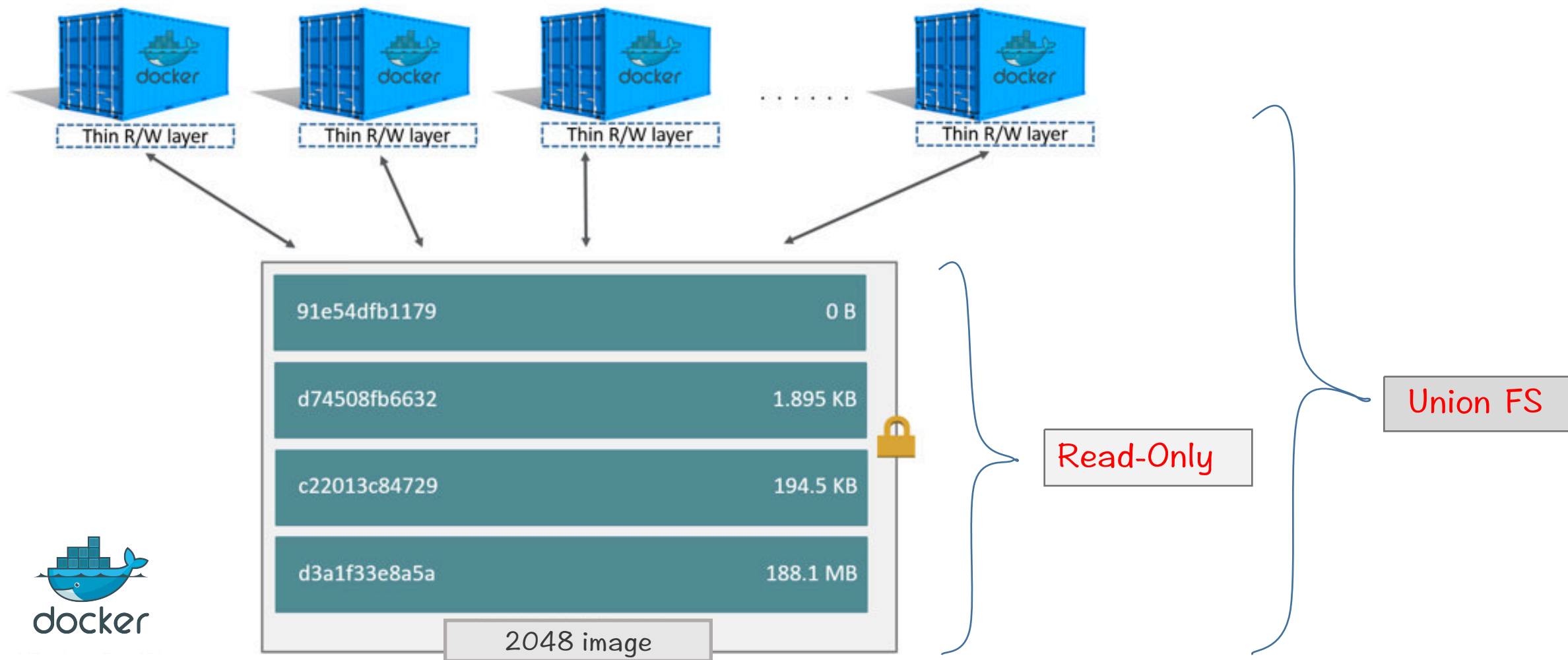
Demo1:

Run Game 2048 in a Docker Container



docker

# The Thought Process



# The Thought Process



## General Volume :

A [Windows](#) system with two physical hard disks.

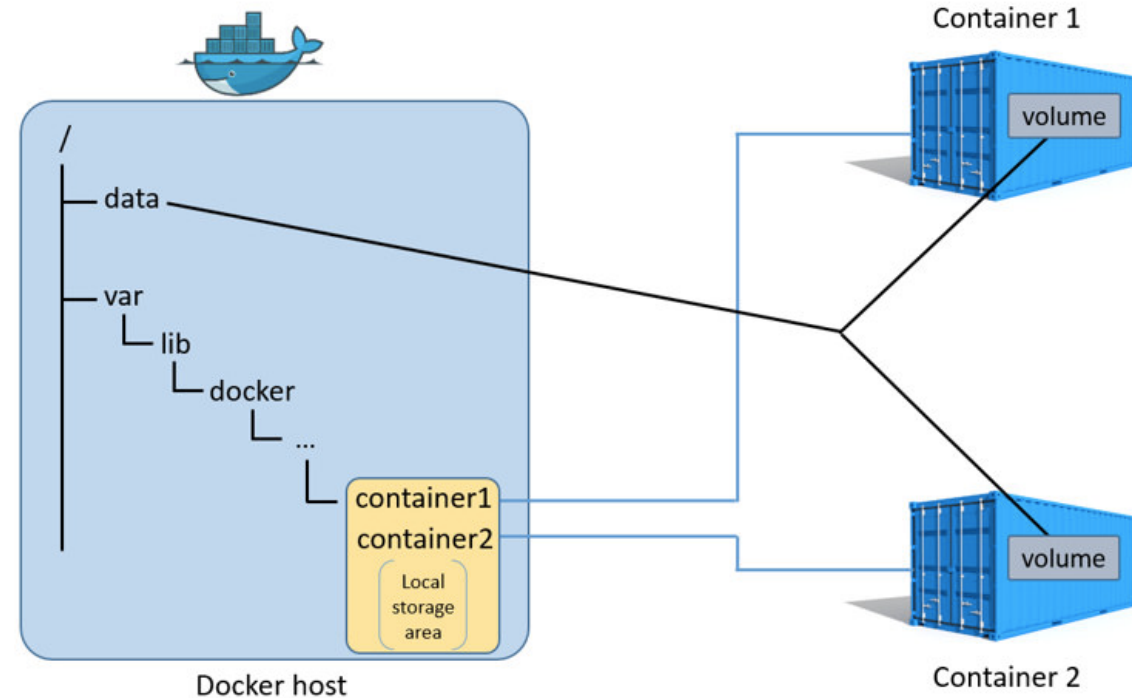
| Physical disk | Partition   | Filesystem | Driver letter |
|---------------|-------------|------------|---------------|
| Hard Disk 1   | Partition 1 | NTFS       | C:            |
|               | Partition 2 | FAT 32     | D:            |
| Hard Disk 2   | Partition 1 | FAT 32     | E:            |

- "C:", "D:", and "E:" are volumes.
- Hard Disk 1 and Hard Disk 2 are physical disks.
- Any of these can be called a "drive".



## Docker Volume :

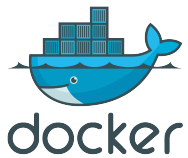
A specially-designated directory within one or more containers that bypasses the [Union File System](#)





Demo2:

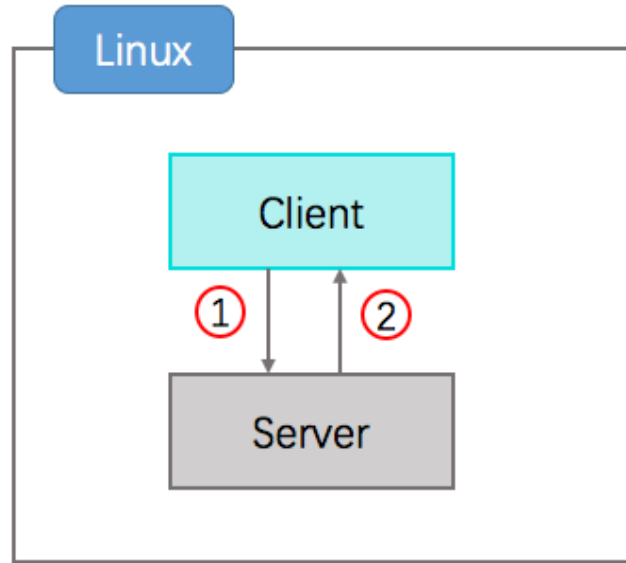
Create A Volume in Docker Container



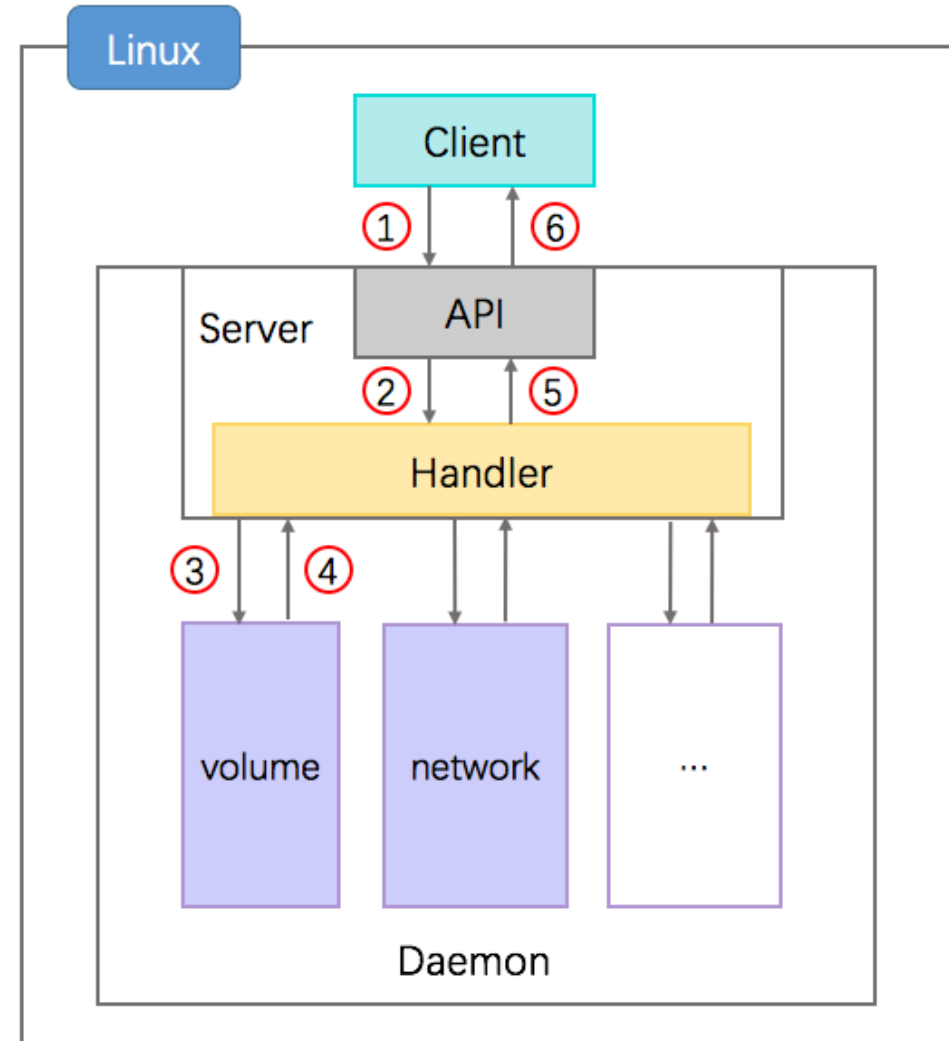
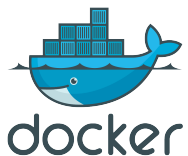
# The Thought Process



## Docker Architecture



C/S Architecture



<https://github.com/docker/docker>



This repository Search Pull requests Issues Gist

docker / docker Watch 3,145 Star 40,609 Fork 12,238

Code Issues 2,297 Pull requests 145 Projects 3 Wiki Pulse Graphs

Docker - the open-source application container engine <https://www.docker.com>



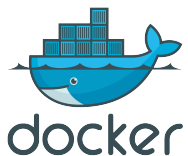
Branch: master New pull request Create new file Upload files Find file Clone or download

Switch branches/tags

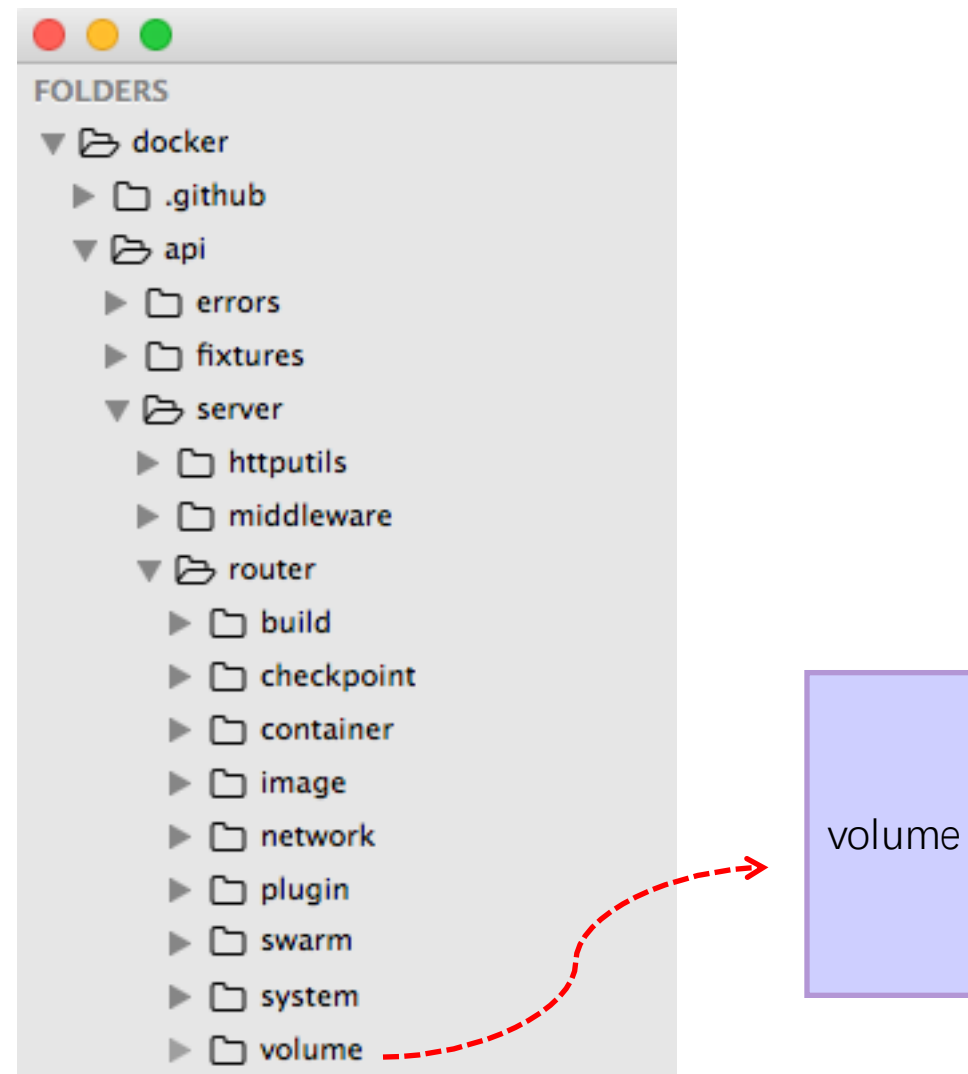
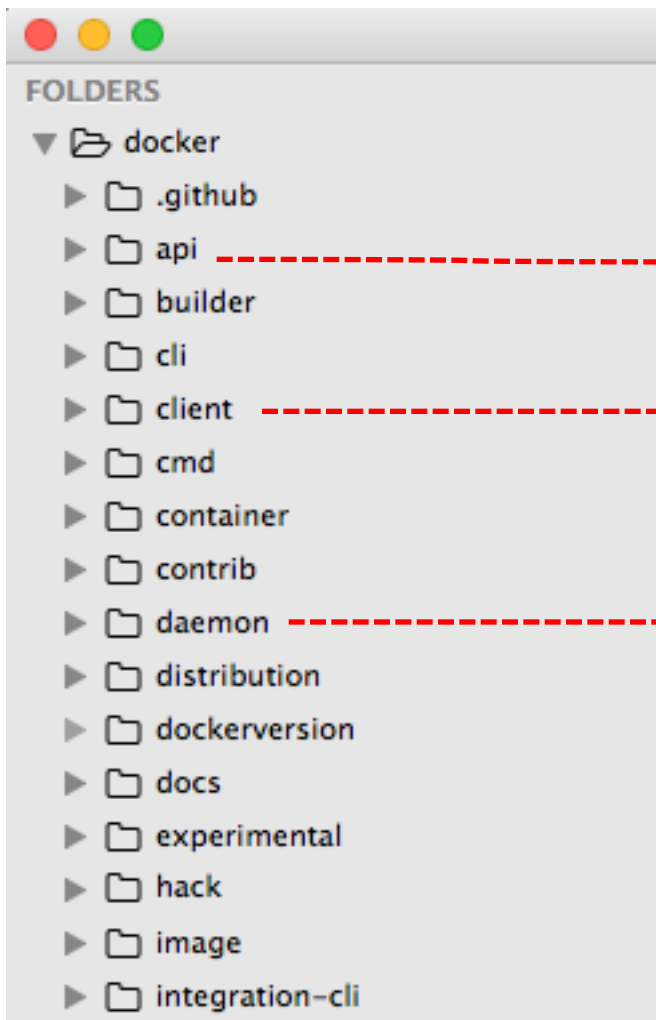
Filter branches/tags

| Branches       | Tags                                                                   |
|----------------|------------------------------------------------------------------------|
| 1.12.x         |                                                                        |
| 1.13.x         |                                                                        |
| <b>17.03.x</b> |                                                                        |
| docs           |                                                                        |
| ✓ master       |                                                                        |
| contrib        | Merge pull request #31635 from christopher-gibson/master 5 hours ago   |
| daemon         | Merge pull request #31500 from dperny/fix-service-logs-cli 3 hours ago |
| distribution   | Merge pull request #30047 from duglin/distError 4 days ago             |

| Commit                                                        | Time                              |
|---------------------------------------------------------------|-----------------------------------|
| request #31500 from dperny/fix-service-logs-cli               | Latest commit 1d46080 3 hours ago |
| SUE_TEMPLATE.md                                               | 5 months ago                      |
| request #31710 from sanimej/drillerr                          | 12 hours ago                      |
| request #31177 from allencloud/donot-allow-subsecond-in-do... | 6 days ago                        |
| request #31500 from dperny/fix-service-logs-cli               | 3 hours ago                       |
| request #31710 from sanimej/drillerr                          | 12 hours ago                      |
| hand options that are related to Windows                      | 2 days ago                        |
| request #30807 from chchliang/networkproject                  | 8 days ago                        |



# Source Code Travel



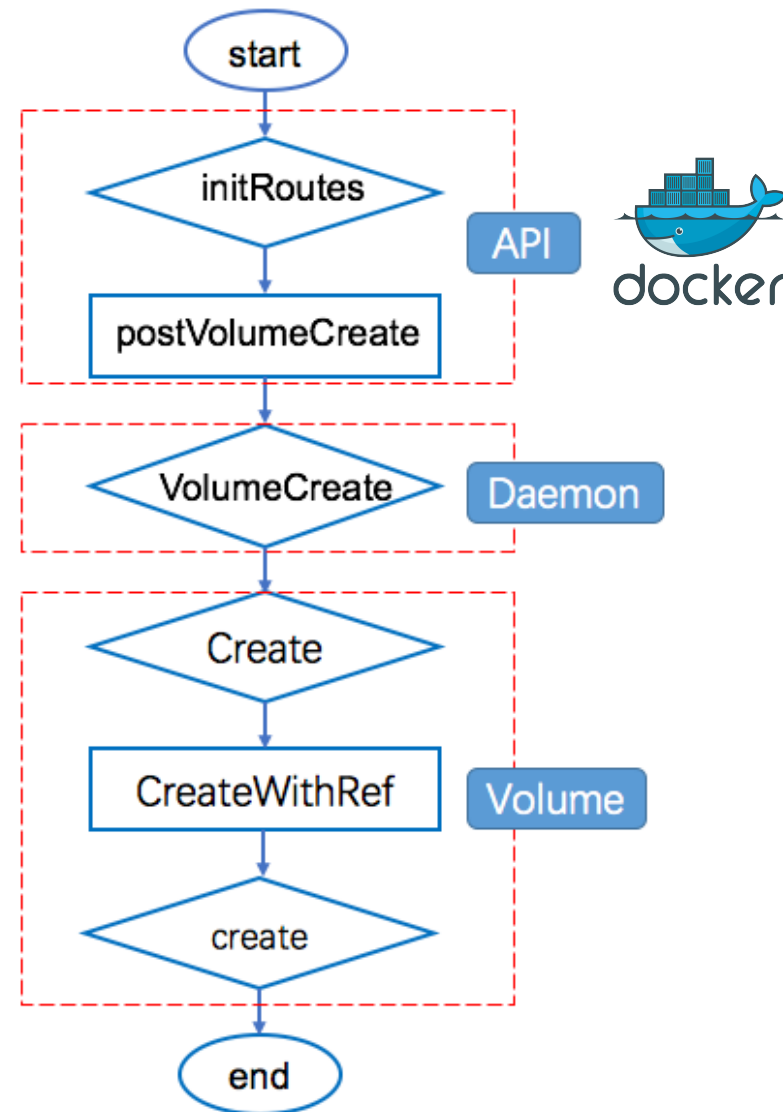


# Source Code Travel



```
func (r *volumeRouter) initRoutes() {
    r.routes = []router.Route{
        // GET
        router.NewGetRoute("/volumes", r.getVolumesList),
        router.NewGetRoute("/volumes/{name:.*}", r.getVolumeByName),
        // POST
        router.NewPostRoute("/volumes/create", r.postVolumesCreate),
        router.NewPostRoute("/volumes/prune", r.postVolumesPrune),
        // DELETE
        router.NewDeleteRoute("/volumes/{name:.*}", r.deleteVolumes),
    }
}

func (v *volumeRouter) postVolumesCreate(ctx context.Context, w http.ResponseWriter,
    r *http.Request, vars map[string]string) error {
    .....
    volume, err := v.backend.VolumeCreate(req.Name, req.Driver, req.DriverOpts, req.Labels)
    if err != nil {
        return err
    }
    return httputils.WriteJSON(w, http.StatusOK, volume)
}
```



# Source Code Travel



```
// Daemon holds information about the Docker daemon.
```

```
type Daemon struct {
```

```
    ID             string
    repository     string
    containers     container.Store
```

```
    .....
```

```
    EventsService *events.Events
```

```
    netController libnetwork.NetworkController → 网络
```

```
    volumes       *store.VolumeStore → 存储卷
```

```
    .....
```

```
}
```

```
// VolumeCreate creates a volume with the specified name, driver, and opts
```

```
// This is called directly from the Engine API
```

```
func (daemon *Daemon) VolumeCreate(name, driverName string, opts,
```

```
labels map[string]string) (*types.Volume, error) {
```

```
    if name == "" {
```

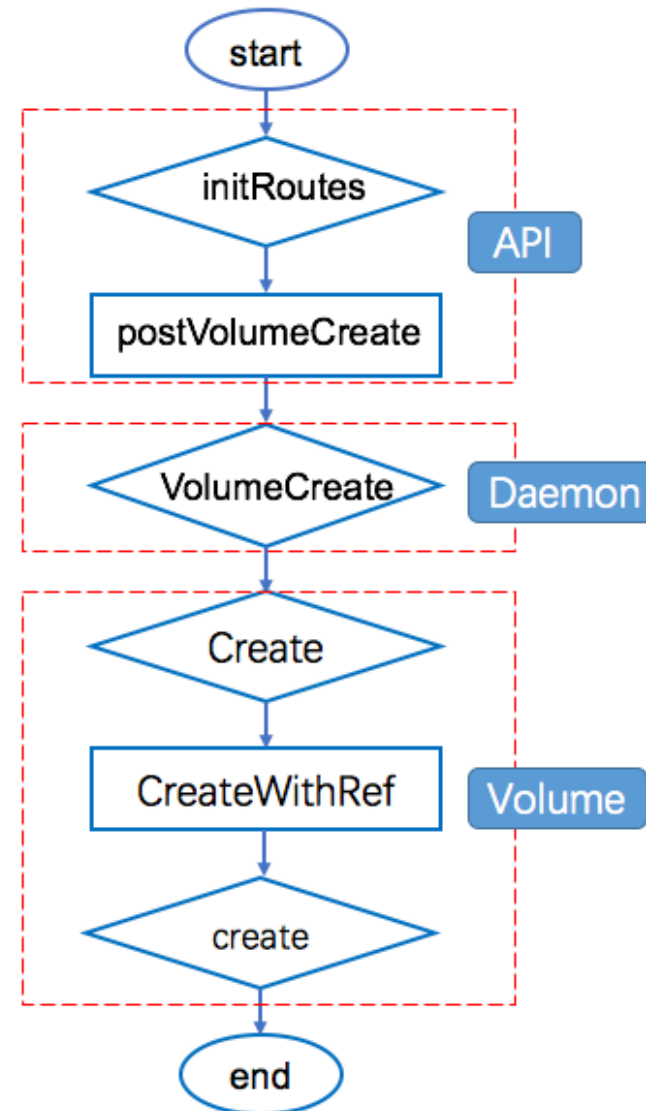
```
        name = stringid.GenerateNonCryptoID()
```

```
    }
```

```
    v, err := daemon.volumes.Create(name, driverName, opts, labels)
```

```
    .....
```

```
}
```



# Source Code Travel



docker



DaoCloud

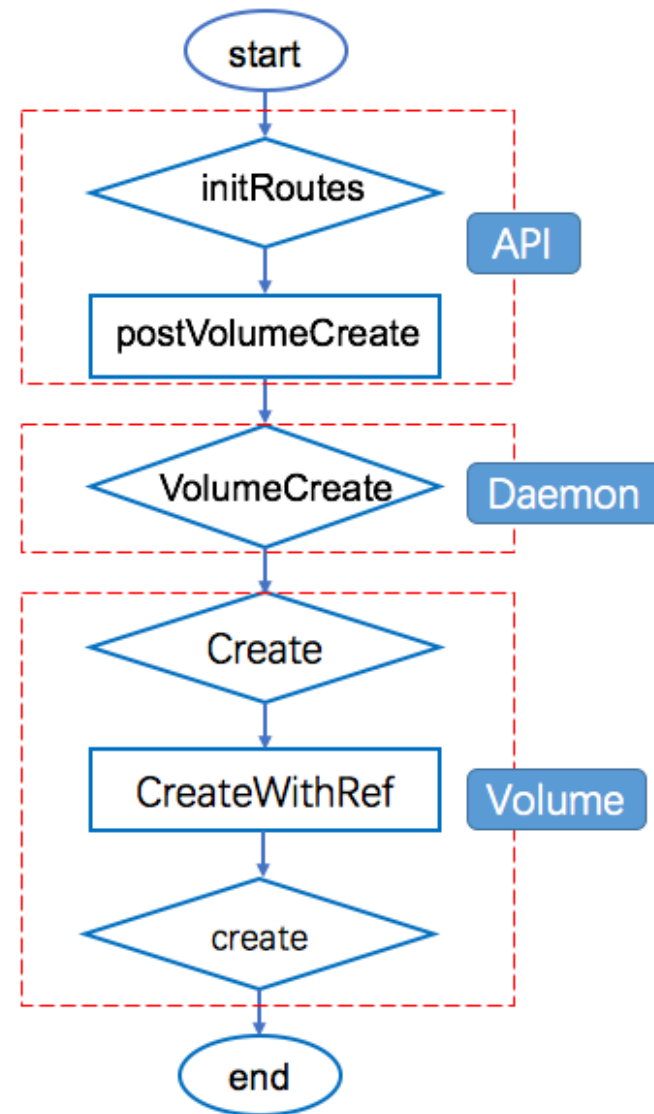


不止于技术

```
func (s *VolumeStore) Create(name, driverName string, opts,
labels map[string]string) (volume.Volume, error) {
return s.CreateWithRef(name, driverName, "", opts, labels)
}

func (s *VolumeStore) CreateWithRef(name, driverName, ref string, opts,
labels map[string]string) (volume.Volume, error) {
.....
v, err := s.create(name, driverName, opts, labels)
.....
}

func (s *VolumeStore) create(name, driverName string, opts,
labels map[string]string) (volume.Volume, error) {
// Validate the name in a platform-specific manner
valid, err := volume.IsVolumeNameValid(name)
if err != nil {
return nil, err
}
if !valid {
return nil, &OpErr{Err: errInvalidName, Name: name, Op: "create"}
}
.....
return volumeWrapper{v, labels, vd.Scope(), opts}, nil
}
```



docker

# Source Code Travel



```
// VolumeStore is a struct that stores the list of volumes available and
// keeps track of their usage counts
type VolumeStore struct {
    .....

    refs map[string]map[string]struct{}

    labels map[string]map[string]string

    options map[string]map[string]string
    db      *bolt.DB

    max int } -----> 声明 max 变量

// New initializes a VolumeStore to keep
// reference counting of volumes in the system.
func New(rootPath string) (*VolumeStore, error) {
    vs := &VolumeStore{
        locks: &locker.Locker{},
        names:  make(map[string]volume.Volume),
        refs:   make(map[string]map[string]struct{}),
        labels: make(map[string]map[string]string),
        options: make(map[string]map[string]string),

        max: 5, } -----> 为 max 变量赋初值
}
```

// define a vols in create() function  
var vols []volume.Volume

// call list() to return the vols  
vols, \_, \_ := s.list()

// calculate current volume number  
number := len(vols)

// add a judgment statement  
if number < s.max

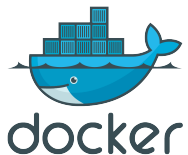
# Conclusion

---



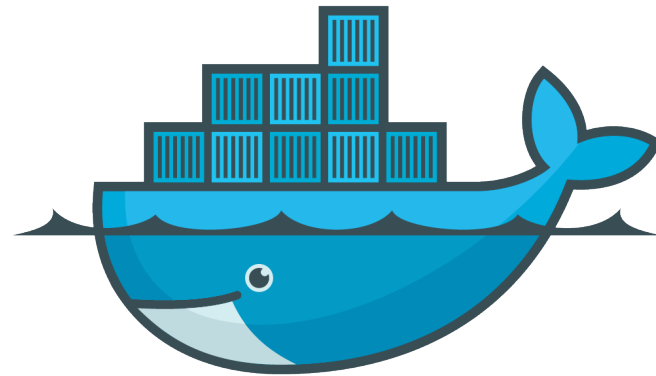
1. Docker is a \_\_\_\_\_ to run container ?
2. For users, Docker is a \_\_\_\_\_ architecture ?
3. How to travel in a source code ocean ?
4. How to become a Docker committer ?

**Disappointing, but not fatal**





Thanks For Your Time



docker