

Giving storage access in swarm

1. Update an already existing service with a volume using the command below.

```
docker service update --mount-add  
type=volume,source=<volume_name>,target=/<destination> <service_name>
```

e.g- docker service update --mount-add type=volume,source=new-vol,target=/apps test

```
root@ip-172-31-23-36:/home/ubuntu# docker service update --mount-add type=volume,source=new-vol,target=/apps test  
test  
overall progress: 3 out of 3 tasks  
1/3: running [=====>]  
2/3: running [=====>]  
3/3: running [=====>]  
verify: Service converged  
root@ip-172-31-23-36:/home/ubuntu#
```

i-0efc1c15a6ac477fa (Manager)
PublicIPs: 54.221.85.119 PrivateIPs: 172.31.23.36

Now let's inspect it

And we can see target and source is /apps and new-vol

```
aws Services Search [Alt+S]  
Message: update completed  
Placement:  
UpdateConfig:  
  Parallelism: 1  
  On failure: pause  
  Monitoring Period: 5s  
  Max failure ratio: 0  
  Update order: stop-first  
RollbackConfig:  
  Parallelism: 1  
  On failure: pause  
  Monitoring Period: 5s  
  Max failure ratio: 0  
  Rollback order: stop-first  
ContainerSpec:  
  Image: nginx:latest@sha256:e3ffd9d807cce9d9f973faff2e420b05243b49fd241b576a3de929bb3362cb60  
  Init: false  
Mounts:  
  Target: /apps  
  Source: new-vol  
  ReadOnly: false  
  Type: volume  
Resources:  
  Networks: my-net  
Endpoint Mode: vip  
root@ip-172-31-23-36:/home/ubuntu#
```

i-0efc1c15a6ac477fa (Manager)
PublicIPs: 54.221.85.119 PrivateIPs: 172.31.23.36