Docker Swarm Cheatsheet

Node management

- Initialize a swarm: `docker swarm init`
- List swarm nodes: `docker node ls`
- Get the command for new nodes to join a swarm: `docker swarm join-token worker` or `docker swarm join-token worker` . The command prints the required command and token: E.g. To add a worker to this swarm, run the following command:

```
docker swarm join \
--token SWMTKN-1-49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv-
8vxv8rssmk743ojnwacrr2e7c \
192.168.99.100:2377
```

- Check service manger reachability: `docker node inspect manager-node-name --format "{{ .ManagerStatus.Reachability }}"`
- Check node state: `docker node inspect node-name --format "{{ .Status.State }}"`
- Put a node in maintenance mode: `docker node update --availability drain node_name`
- Activate a node (after maintenance): `docker node update --availability active node_name`
- Add a label: `docker node update --label-add key=value node_name`
- Remove a label: `docker node update --label-rm key node_name`
- Search label: `docker node inspect node_name | grep Labels -C5`

Service management

- List services (manager node): `docker service ls`
- Describe services (manager node): `docker service ps service_name`
- Inspect a service: `docker service inspect service_name`
- Scale a service: `docker service scale service_name=N`
- Remove service: `docker service rm service_name`

Stack management

- Deploy stack from docker-compose file: `docker stack deploy -c docker-compose.yml stack_name`
- List stacks: `docker stack ls`
- List stack services: `docker stack services stack_name`
- List stack tasks: `docker stack ps stack_name`

- Remove stack : `docker stack rm stack_name`

Network management

- List networks: `docker network ls`
- Create overlay network: `docker network create -d overlay network_name`
- Remove network: `docker network rm network_name`

Monitor services

- Docker stats: `docker stats`
- Service logs: `docker service logs service_name`
- Cluster-wide monitoring and log collection with Sematext Docker Agent on Swarm: Create a Docker App in Sematext Cloud to get monitoring tokens and use them instead the placeholders `YourContainerToken` , `YourInfraToken` and `YourLogsToken` in commands below. Follow the instructions in the UI, which shows the following command with your tokens:

```
docker service create --mode global --name st-agent \
--restart-condition any \
--mount type=bind,src=/var/run,dst=/var/run/ \
--mount type=bind,src=/usr/lib,dst=/host/usr/lib \
--mount type=bind,src=/sys/kernel/debug,dst=/sys/kernel/debug \
--mount type=bind,src=/proc,dst=/host/proc,readonly \
--mount type=bind,src=/etc,dst=/host/etc,readonly \
--mount type=bind,src=/sys,dst=/host/sys,readonly \
-e NODE_NAME={{.Node.Hostname}} \
-e CONTAINER_TOKEN=YourContainerToken \
-e INFRA_TOKEN=YourInfraToken \
-e JOURNAL_DIR=/var/run/st-agent \
-e LOGGING_REQUEST_TRACKING=false \
-e LOGGING_WRITE_EVENTS=false \
-e LOGGING_LEVEL=info \
sematext/agent:latest
```

Collect all container logs:

```
docker service create --mode global --name st-logagent \
--restart-condition any \
--mount type=bind,src=/var/run/docker.sock,dst=/var/run/docker.sock \
-e LOGS_TOKEN=YourLogsToken \
sematext/logagent:latest
```