

(Analysis on)

Container Migration|Kubernetes

July 19, 2018

Sachin Jagtap

Email: Sachin.Jagtap@click2cloud.net

Document Version – V1.0

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Name | Description of Change |
| 1.0 | 19/07/2018 | Sachin Jagtap | Initial draft created |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Reviewers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Version Approved | Role | Date |
| Rajiv Khobragade | 1.0 | PM | 19/07/2018 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Container Migration 3](#_Toc519775961)

[CRIU Limitations 3](#_Toc519775962)

[Limitation: 3](#_Toc519775963)

[Use Docker experimental 4](#_Toc519775964)

[Issues: 4](#_Toc519775965)

[Tools Found: 4](#_Toc519775966)

[CloudSlang 4](#_Toc519775967)

[Steps to migrate 4](#_Toc519775968)

[References: 6](#_Toc519775969)

# **Container Migration**

There are several problems that container migration can address:

* **Downtime during hardware maintenance:** When a system administrator needs to upgrade hardware, it`s very painful to migrate all the customers from one hardware node to another hardware node, and in many cases, it's just impossible without downtime.
* **Unbalanced cluster load:** When one hardware node becomes overloaded, the rebalance process can require implementation of specific application patterns narrowing the choice of workloads that can be hosted in the cluster.
* **Troubles within a cloud:** Today there are so many clouds on the market and sometimes they have downtimes, change pricing policy or degrade in the quality of services. And in most cases, it is a problem to easily migrate the application from one cloud provider to another.

Please find the below tools with the help of which we can migrate containers.

* [Virtuozzo](https://virtuozzo.com/) - this team created the live migration technology for containers; they were pioneers in this direction and today already offer a production-ready containers engine with live migration.
* [runC](http://runc.io/), from the Open Containers Initiative, is another upcoming container engine solution with live migration based on [CRIU](https://criu.org/Main_Page).
* [Jelastic](https://jelastic.com/) is offering containers orchestration platform that provides live migration of production applications across hardware regions, data centers and cloud vendors.

# **CRIU Limitations**

One of the most obvious and hardest requirements for using CRIU is that used libraries (on the source and destination system) of the checkpointed and restored process must be exactly the same. The libraries are not newly loaded on the destination system of the restore. The restored binary expects all library provided functions to be at the (exact) same memory address as before. If a used library function is at a different memory location – the restored process will crash. Although this sounds like a severe restriction… it is not as fatal as it sounds. For example, when using CRIU to migrate a container (i.e. a “[fancy process](http://crunchtools.com/container-myths-redux/)”) the container will often include not only the actual application to be migrated – it will also include the required libraries.

Another limitation of note: CRIU cannot (currently) be used to migrate applications which are directly accessing hardware through ioctl(). If such an application needs to be checkpointed and restarted or migrated CRIU provides an interface to create plugins which can be used to extract the state of the hardware on the source system and then put the hardware back into the same state during restore.

It is also important to remember that it is not possible to checkpoint processes which are already being ptrace’d (e.g., gdb, strace).

# **Limitation:**

1. Checkpointed and restored process must be exactly the same.
2. The libraries are not newly loaded on the destination system of the restore.
3. The restored binary expects all library provided functions to be at the (exact) same memory address as before.

# **Use Docker experimental**

Experimental features are now included in the standard Docker binaries as of version 1.13.0. To enable experimental features, start the Docker daemon with the --experimental flag or enable the daemon flag in the /etc/docker/daemon.jsonconfiguration file:

{

"experimental": true

}

You can check to see if experimental features are enabled on a running daemon using the following command:

$ docker version -f '{{.Server.Experimental}}'

true

# **Issues:**

Starting docker in experimental mode is giving following issue.

ddb3c5103d64ce5be 
fbbdf7a62e0f48-node1:—# export cid=$(docker run 
busybox 
root@c7 
docker: Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the 
See •docker run - -help' . 
tail 
docke r 
/dev/null) 
daemon running? . View O)gs 
click2doud-Sachin-poc 
Alibaba 
Hong 
Cloud 
All 
exceptions 
Manage 
17.06.2- 
c7ddb3c5103d64ce5befbbdf7a62eOf48 
Kong 
Cluster 
Running 
j6crj3qsn6xj544nvfv1t 
Monitor 
Delete 
More 

# **Tools Found:**

1. CloudSlang
2. CloudEndure
3. Cloudify
4. OneOps
5. Jelastic

## **CloudSlang**

### **Steps to migrate**

1. Install Docker

2. Install CRIU

3. Run a redis container

4. Save the container to a tar file:

    docker export redis > redis.tar

5. Stop and delete the container

6. Create a folder for the redis  runc container.

7. Create a rootfs subfolder, and extract redis.tar to rootfs:

     cd redis

     tar -C rootfs -xf redis.tar

8. Create a json specification file for the container:

     cd redis

     docker-runc spec

9. Edit config.json and

     a. change terminal to false:

     "process": {

                    "terminal": false,

                    "user": {},

                   "args": [

                            "redis-server"

                    ],

    b. change readonly to false

     "root": {

                    "path": "rootfs",

                   "readonly": false

            },

10. Create folders for your dump and predump images (mkdir redis/dump;mkdir redis/predump).

11. Start the container: docker-runc start redis

12. On the target host: Install Docker and CRIU, create the folders, copy the redis.tar and config.json files.

    and extract redis.tar to the rootfs.

    Note: Do not start the container in the target host!

13. In the source host, insert some data to your redis database:

     docker-runc exec redis redis-cli set cloudslang super-cool

    you are now ready to run the migrate.sl flow

14. When the flow is done, run the following command on the target docker host:

    docker-runc exec redis redis-cli get cloudslang

# References:

<https://forums.docker.com/t/container-migration/28951>

<https://circleci.com/blog/checkpoint-and-restore-docker-container-with-criu/>

<https://forums.docker.com/t/migrate-docker-container-from-on-premise-to-cloud-azure/33039>

<https://forums.docker.com/t/moving-a-docker-container-to-another-machine/5605>

<https://stackoverflow.com/questions/47288677/container-migration-in-docker>

<https://stackoverflow.com/questions/42487859/docker-container-migration>

<https://rhelblog.redhat.com/2016/09/26/from-checkpointrestore-to-container-migration/>

<https://criu.org/Docker>

[https://criu.org/Installation#Installing\_from\_packages](https://criu.org/Installation" \l "Installing_from_packages)

<https://medium.com/@TarunChinmai/criu-installation-f277cda14ce0>

<https://medium.com/@tigranbs/container-is-live-ok-lets-move-it-1022abcb6250>

<https://technology.amis.nl/2018/04/08/first-steps-with-docker-checkpoint-to-create-and-restore-snapshots-of-running-containers/>

<https://github.com/CloudSlang/cs-content/blob/master/content/io/cloudslang/docker/runc/examples/migrate_container.sl>