

Ellexus Container Checker user manual

(Version 2.11.0)

Ellexus Container Checker automatically generates an I/O report for your Docker application.

Quick start

Step 1: Purchase a Container Checker AMI instance and log in

Purchase the AMI image from AWS marketplace:

```
$ ssh -i "your-key-pair.pem" ubuntu@<aws-instance-hostname>
```

Step 2: Upload the docker container that you want to check and run it under Container Checker

Use `ellexus-docker-run` command instead of `docker run` to start your Docker container:

```
$ ellexus-docker-run [DOCKER_RUN_OPTIONS] IMAGE [COMMAND] [ARG...]
```

Step 3: When the container has finished running read the Container Checker report

After the container exits, the I/O report is generated under `./ellexus-TIMESTAMP-healthcheck`

The report contains:

- `report.txt`: the main report;
- `dependencies.txt`: a list of dependencies that your application used;
- `programs.csv`: detailed information about each process in your application.

Table of Contents

Quick start.....	1
1.0 Purchasing an Ellexus Container Checker AMI instance.....	2
1.1 Logging on Ellexus Container Checker AMI instance.....	2
2.0 Uploading your Docker container to your AWS instance.....	2
2.1 Running your Docker container under Container Checker.....	2
3.0 Accessing the Container Checker report.....	3
3.1 What can I expect to see in the report?.....	4
3.2 Which applications can you use Ellexus Container Checker on?.....	4

1.0 Purchasing an Ellexus Container Checker AMI instance

Ellexus Container Checker can be purchased from Amazon Marketplace using your AWS account. The size of EC2 instance to run on depends on the Docker container that you are running, but we would recommend at least 2 GiB of RAM. The AWS sizing guidelines which can be found at <https://aws.amazon.com/ec2/instance-types/> may be helpful in making this decision.

1.1 Logging an Ellexus Container Checker AMI instance

After you have purchased Ellexus Container Checker AMI, please log in with ssh on the AWS instance as ubuntu user (AWS may suggest using root account, but please use ubuntu instead):

```
$ ssh -i "your-key-pair.pem" ubuntu@<aws-instance-hostname>
```

You can use all the usual Docker commands on the instance, e.g., to log in to a Docker registry and to manage your containers and images.

2.0 Uploading your Docker container to your AWS instance

Docker CE is installed on the Ellexus Container Checker AMI, so you can use all the usual Docker commands to install your container.

For a Docker image that you only have locally, run the following to export the image from your local machine and upload it

3.0 Accessing the Container Checker report

After the container exits, the I/O report is generated under `ellexus-TIMESTAMP-healthcheck` subdirectory, which contains 3 files:

- `report.txt`: the main report;
- `dependencies.txt`: a list of dependencies that your application used;
- `programs.csv`: detailed information about each process in your application.

The report is human readable and has recommendations for changes that could be made to your application.

You can also execute a Docker container in interactive mode and run commands. For example:

```
$ ellexus-docker-run -it ubuntu
$ root@8344f5cd7fc1:/# <run-commands-to-trace>
$ root@8344f5cd7fc1:/# exit
...
Ellexus Healthcheck report generated: ./ellexus-TIMESTAMP-
healthcheck
```

You can also interrupt a running container with Ctrl-C to generate a report:

```
$ ellexus-docker-run -p 8086:8086 -p 8083:8083 \
                    -e INFLUXDB_ADMIN_ENABLED=true influxdb
...
[I] 2017-09-07T13:50:44Z Sending usage statistics to
usage.influxdata.com
^C[I] 2017-09-07T13:51:02Z Signal received, initializing clean
shutdown...
...
Ellexus Healthcheck report generated: ./ellexus-TIMESTAMP-healthcheck
```

3.1 What can I expect to see in the report?

The report has three sections:

- 1 - 2 Summary
- 3 - 7 Common bad I/O patterns and performance issues
- 8 - 13 Summary of I/O performance for instance sizing and bottleneck detection.

You can generate the output documentation by running:

```
$ ellexus-healthcheck.sh --docs
```

This will generate a full report.txt file, so you can see what could potentially be reported when running your application under Ellexus Container Checker. Where necessary, explanations should inform you as to what is reported, and how.

3.2 Which applications can you use Ellexus Container Checker on?

Ellexus Container Checker supports all applications that use glibc, which includes nearly all applications. It doesn't matter what language they are written in. Python, Perl, Java, C, PHP, Javascript etc will all work automatically.

Go applications won't work and Docker containers that use other C libraries, such as musl-libc, uClibc or dietlibc, are not currently supported so talk to us if you want support for those libraries. An example of such an application is Docker adminer image, which uses musl-libc. If you don't know if you are using these then you are probably not.

Very minimal images, such as Docker hello-world, cannot be checked with Container Checker either, because they don't contain a program that runs from the main shell.

If you need advice about using Ellexus Container Checker on your Docker application, please contact support@ellexus.com.