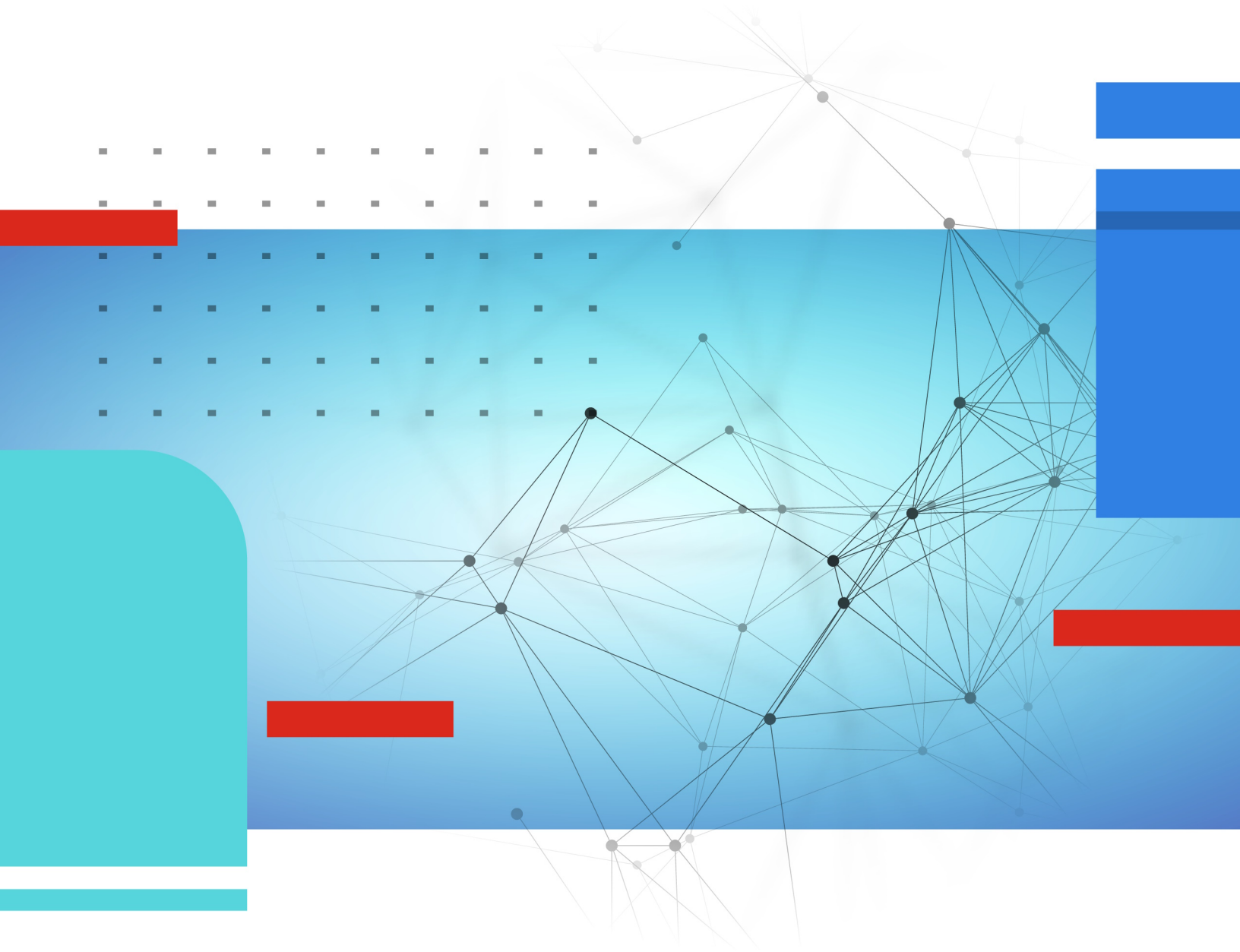
Getting Started (LXC)

Container FortiOS 7.2.2



FORTINET DOCUMENT LIBRARY
https://docs.fortinet.com

FORTINET VIDEO LIBRARY
https://video.fortinet.com

FORTINET BLOG
https://blog.fortinet.com

CUSTOMER SERVICE & SUPPORT
https://support.fortinet.com

FORTINET TRAINING & CERTIFICATION PROGRAM
https://www.fortinet.com/training-certification

FORTINET TRAINING INSTITUTE
https://training.fortinet.com

FORTIGUARD LABS
https://www.fortiguard.com

END USER LICENSE AGREEMENT
https://www.fortinet.com/doc/legal/EULA.pdf

FEEDBACK
Email: techdoc@fortinet.com

June 13, 2025
Container FortiOS 7.2.2 Getting Started (LXC)
87-722-971807-20250613

TABLE OF CONTENTS

Change Log

Date	Change Description
2025-06-10	Initial release.
2025-06-13	Updated Getting the container image on page 6.

Introduction

Container FortiOS provides NGFW firewall features, including security policies, IPS inspection, application control, URL filtering, and antivirus in a container-deployed format.

It supports Linux Containers (LXC), Docker, and Kubernetes.

This guide provides information about the installation and configuration of Container FortiOS version 7.2.2, build 265 on LXC.

Deploying Container FortiOS

This section provides an overview of the procedures for deployment of Container FortiOS.

As container environments vary widely, this document provides basic instructions for deployment to LXC and does not provide in-depth information about configuration of LXC itself.

The basic steps for deployment are as follows:

1. Get the container image.
2. Mount the image.
3. Optionally, configure the default bridge.
4. Specify the persistent storage location.
5. Optionally, specify startup configuration.
6. Configure the container.
7. Start the container.
8. Upload a valid license.

Getting the container image

After purchasing a Container FortiOS license, submit a ticket through the Customer Service & Support site. The Technical Assistance center (TAC) team will then provide you with the appropriate image file.

For more information about submitting a ticket, see Technical Tip: How to create a ticket for Fortinet TAC.

LXC images use the following naming convention:

```
cFOS_<CPU Arch>_<Container Type>-v<Major Version>-build<build number>-<Company>.squashfs
```

For example, image `cFOS_ARM64_LXC-v7-build265-FORTINET.squashfs` was built for LXC running on an 64-bit ARM CPU device. The major version is 7 and build number is 265.

Mounting the container image

The LXC container image contains a root file system in Squashfs format.

Mount it to a directory with the following command:

```
mount <image_file> <mount_point> -t squashfs -o loop
```

Once the image is mounted successfully to `mount_point`, you will find two top level items in that directory: a config file and a rootfs directory. This directory structure is required by LXC.

The configuration file is a template and must be customized for your specific environment.

Configuring the default bridge

After installation, LXC creates a default bridge device `lxcbr0` and all the necessary networking rules to route traffic to the default gateway. By default, `lxcbr0` is on subnet `10.0.3.0/24`.

Container FortiOS applies security inspection over any traffic flows cross the container. The default `lxcbr0` device can be used to bridge the egress interface (`port0`) inside the container.

To change the subnet (for example to `192.168.200.0/24`), update the LXC configuration in `/etc/default/lxc-net`. For example:

```
LXC_BRIDGE="lxcbr0"
LXC_ADDR="192.168.200.1"
LXC_NETMASK="255.255.255.0"
LXC_NETWORK="192.168.200.0/24"
LXC_DHCP_RANGE="192.168.200.2,192.168.200.254"
LXC_DHCP_MAX="253"
```

Restart the lxc-net service to apply the changes:

- Using `systemd`:

  ```
  sudo systemctl restart lxc-net.service
  ```

- Using service manager:

  ```
  sudo service lxc-net restart
  ```

Specifying persistent storage

Container FortiOS containers require persistent local storage. For each container, specify a local directory on the host.

Specifying startup configuration

You may specify a license and configuration to be applied when the container is created. The license must be provided for the configuration to be applied.

You may specify either a full configuration file or a partial configuration file, but not both.

Full configuration file

A full backup configuration file can be exported from a running container:

```
exec config backup <file name> [password]
```

The full configuration file must be encrypted with a password in order to be applied to a container with a different data directory.

Partial configuration file

A partial configuration file can be applied at startup, but it cannot be encrypted.

Specifying configuration and license at startup

To specify configuration and license at startup:

1. Copy the backup file to the `data` directory and name it `cfos.conf`.

   When using an encrypted configuration file at startup, save the backup password in a file named `cfos.key` in the data directory.

   You may also restore a partial configuration using a `cfos-partial.conf` file.
2. Copy the Container FortiOS license file to the `data` directory and name it `cfos.lic`.
   The license and configuration files are read and applied to the container when it runs, then deleted automatically.

Configuring the container

Update the sample configuration file with values appropriate to your environment.

The sample configuration file inside the squashfs image is as follows:

```
# Container specific configuration
lxc.signal.halt = SIGUSR1
lxc.signal.reboot = SIGTERM
lxc.tty.max = 1
lxc.pty.max = 1
#lxc.cap.drop = sys_module mac_admin mac_override sys_time
lxc.mount.auto = cgroup:mixed proc:mixed sys:mixed
lxc.mount.entry = shm dev/shm tmpfs defaults,create=dir 0 0
lxc.mount.entry = mqueue dev/mqueue mqueue defaults,optional,create=dir 0 0
lxc.mount.entry = tmpfs tmp tmpfs defaults 0 0
lxc.mount.entry = tmpfs run tmpfs defaults 0 0
lxc.mount.entry = {DATA_DIR} data none bind,create=dir  0 0
#lxc.mount.entry = /sys/kernel/security sys/kernel/security none ro,bind,optional 0 0
```

```
lxc.rootfs.path = dir:{CN_ROOTFS}
lxc.uts.name = {CN_NAME}
lxc.environment = FCN_DNS={CN_DNS}
# Network configuration
lxc.net.0.type = veth
lxc.net.0.name = port0
lxc.net.0.veth.pair = vethport0
lxc.net.0.link = {PORT0_BRIDGE}
lxc.net.0.flags = up
lxc.net.0.ipv4.address = {PORT0_IP}
lxc.net.0.ipv4.gateway = {GATEWAY}
lxc.net.1.type = veth
lxc.net.1.name = port1
lxc.net.1.veth.pair = vethport1
lxc.net.1.link = {PORT1_BRIDGE}
lxc.net.1.flags = up
lxc.net.1.ipv4.address = {PORT1_IP}
```

Copy the sample configuration file and update all the variables inside curly braces with your specific values as follows:

Variable	Description
DATA_DIR	A directory on the host that is mounted into the container as persistent storage.
CN_ROOTFS	The path to the root file system of the container.
CN_NAME	The container hostname.
CN_DNS	The DNS server the container uses to access the internet.
PORT0_BRIDGE	The interface addressto be used for network traffic on port0.
PORT0_IP	The IP address range for this interface.
GATEWAY	The IPv4 address of the gateway inside the container.
PORT1_BRIDGE	The interface to be used for network traffic on port0.
PORT1_IP	The IP address range for this interface.

`DATA_DIR` and `CN_DNS` are specific to Container FortiOS.

If you are assigning multiple interfaces to the Container FortiOS container, enable IP forwarding by adding the following line to the configuration file:

```
lxc.hook.autodev: sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

For more information about LXC container configuration options, see https://man7.org/linux/man-pages/man5/lxc.container.conf.5.html or enter `man lxc.container.conf` at the Linux command prompt.

In this config, `port0` and `port1` are virtual ethernet ports (`veth`). Other network types, such as MACVLAN and IPVLAN, can also be used.

The port names can be changed. For example, you can use `wan` and `lan` for `port0` and `port1`.

If you change these names, you must update the `fcnlxc` script, if used.

Starting the container

Because the LXC image is a squash file system image, not a fully extracted file system, a shell cript, `fcnlxc`. is provided to help import and run the image. It wraps common LXC commands such as `lxc-start` and `lxc-stop`.

To manage the container with the fcnlxc script:

1. Import the container image using the following command:

   ```
   ./fcnlxc import [option]... [FOS container image]
   ```

   Set the following options:

   - **`-n`**: Set the container name. Default: `fcn`. (`CN_NAME` in the configuration file.)
   - **`-p`**: Set the persistent storage directory for the container. (`DATA_DIR` in the configuration file.)
   - **`-m`**: Set the mount directory for the base image.
   - **`-r`**: Set the container top level root directory. (`CN_ROOTFS` in the configuration file.)
   - **`-d`**: Set the container DNS server. Default: `8.8.8.8`. (`CN_DNS` in the configuration file.)
   - **`-w`**: Set interface `port0` IP address. Default: `10.0.3.10/24 10.0.3.255`. (`PORT0_IP` in the configuration file.)
   - **`-l`**: Set interface `port1` IP address. Default: `10.0.4.10/24 10.0.4.255`. (`PORT1_IP` in the configuration file.)
   - **`-c`**: Use this specific LXC template config file.

   Below is an example `import` command. In this example, all DNS and IP addresses use default values.

   ```
   ./fcnlxc import -n fcntest -p /tmp/data ./FOS_X64_LXC-v1-build0001-FORTINET.squashfs
   ```

2. Run the container using the following command:

   ```
   ./fcnlxc start [option]... [lxc_options]
   ```

   Below is an example `start` command. With the LXC `-F` option, the container will run in the foreground. The default LXC option is to run in the background in daemon mode.

   ```
   ./fcnlxc start -n fcntest -F
   ```

3. Use the following command to attach to the instance:

   ```
   sudo ./fcnlxc console -n fcntest
   ```

4. To exit from container instance, press `CTRL + A + Q`.
5. To stop the container, do one of the following:
   - In the container CLI, enter the following command:

     ```
     execute shutdown
     ```

   - In the host shell, enter the following command:

     ```
     sudo ./fcnlxc stop -n fcntest
     ```

When AppArmor is enabled and you try to launch a container in LXC, you may see error messages similar to the one below.

```
lxc-start: fcntest: lsm/apparmor.c: apparmor_prepare: 1111 If you really
want to start this container, set
lxc-start: fcntest: lsm/apparmor.c: apparmor_prepare: 1112
lxc.apparmor.allow_incomplete = 1
lxc-start: fcntest: lsm/apparmor.c: apparmor_prepare: 1113 in your
container configuration file
lxc-start: fcntest: start.c: lxc_init: 845 Failed to initialize LSM
```

To solve this issue, edit the `/etc/lxc/default.conf` file and add the following line:

```
lxc.apparmor.allow_incomplete = 1
```

Restart the `lxc-net` service to apply the change.

Using Container FortiOS

This section provides an overview of the initial steps for connecting to and configuring the running Container FortiOS container.

Connecting to the Container FortiOS CLI

Container FortiOS provides access to the FortiOS shell for CLI usage as well as the underlying Linux shell.

The Container FortiOS CLI is based on the FortiOS CLI, but has fewer available options.

To connect to the running Container FortiOS container:

In the host shell, enter the following command:

```
lxc-console -n <container_name>
```

The initial username is `admin` with an empty password. Use `config system admin` to set a password.

To enter the Linux shell, use the following command:

```
sysctl sh
```

Connecting to the REST API

Container FortiOS provides a REST API to perform configuration and monitoring operations. The API provides a subset of the FortiOS API.

The API is accessible by default on port 443 at any of the container interfaces. If configured to require a token, all requests must include the API token.

For example, the following examples get the antivirus settings:

```
curl -H "Authorization: Bearer rkMJd3SdLhb8UFBan987CnIrmPBLfaIj"
https://localhost/api/v2/cmdb/antivirus/settings
```

```
curl https://localhost/api/v2/cmdb/antivirus/settings?access_
token=rkMJd3SdLhb8UFBan987CnIrmPBLfaIj
```

To see the available options for each object, append `?action=schema` to the url.

For full details on the available API actions and access, see the Container FortiOS REST API documentation on FNDN.

Uploading a license

You must upload a valid Container FortiOS license before many features are available.

To upload a license:

1. Open the license file in a text editor and copy the full contents.
2. In the container CLI, enter the following command:

   ```
   exec import-license "<content_of_license_file>"
   ```

   Container FortiOS validates the license.

More information

Additional Container FortiOS documentation is available in the Fortinet Documentation Library.

FortiOS documentation

Configuration and administration of Container FortiOS is very similar to FortiOS.

The following FortiOS documentation may be helpful:

- FortiOS Administration Guide
- FortiOS CLI Reference
- FortiOS REST API Reference on FNDN

www.fortinet.com