

NAME

iocage - jail manager amalgamating ZFS, VNET and resource limits

SYNOPSIS

```

iocage fetch
iocage create [-c] [release=RELEASE]
iocage clone UUID
iocage list [-t]
iocage start UUID
iocage stop UUID
iocage console UUID
iocage get [property | all ] UUID
iocage set property=value UUID
iocage cap UUID
iocage uncap UUID
iocage inuse UUID
iocage snapshot UUID
iocage snaplist UUID
iocage snapremove snapshotname UUID
iocage defaults
iocage version | --version
iocage help

```

DESCRIPTION

iocage is a system administration tool for jails designed to simplify jail management tasks. It abstracts away the management of ZFS backed jails running VNET networking with optional support for resource limits.

NOTE: only VNET enabled jails are supported! The older shared IP based jails are intentionally not supported for various reasons - mostly VNET enabled jails provide better isolation.

Each jail has a unique ID (*UUID*) automatically generated at creation time. Using *UUID*'s as jail names means that a jail can be replicated in a distributed environment with greater flexibility. This also eliminates potential naming clashes on large scale deployments.

To ease jail identification a TAG field is included in *list* mode which can be *set* to any string (hostname, label, note, etc.). By default the TAG field contains the creation date and time stamp.

Properties are stored inside ZFS custom fields. This eliminates the need for any configuration files and jails can be easily moved with ZFS send and receive preserving *all* of their properties automatically.

iocage relies on ZFS, VIMAGE (VNET) and has the following minimal requirements:

- o FreeBSD 10.0-RELEASE amd64
- o bridge interfaces (bridge0,bridge1) add:


```
cloned_interfaces="bridge0 bridge1" to /etc/rc.conf
```
- o Kernel compiled with:

options	VIMAGE
options	RACCT
options	RCTL

SUBCOMMANDS

fetch [*release=RELEASE*]

fetch has to be executed as the first command on a pristine system. By default *fetch* will download the host node's *RELEASE* for deployment. If other releases are required this can be changed with supplying the required release property.

Example: `iocage fetch release=9.2-RELEASE`

create [-c] [release=RELEASE] [property=value] [property=value] ...

By default create will deploy a new jail based on the host operating system's release. This can be changed by specifying the release option. If the -c switch is specified the jail will be cloned from the release. Default is to create a fully independent jail set.

Example: `iocage create tag=www01`

clone [property=value] ... UUID

Clone jail identified by UUID (ZFS clone). All properties will be reset on the clone, defaults can be overridden by specifying properties on the fly.

Example: `iocage clone tag=www02 UUID`

list [-t]

List all jails, if -t is specified list only templates.

df

List disk space related information. Available fields:

CRT - compression ratio
RES - reserved space
QTA - disk quota
USE - used space
AVA - available space

start UUID

Start jail identified by UUID.

stop UUID

Stop jail identified by UUID.

console UUID

Console access, drop into jail.

get [property | all] UUID

Get named property or if all is specified dump all properties known to iocage.

To display whether resource limits are enforced for jail:

`iocage get rlimits UUID`

set property=value UUID

Set a property to value.

cap UUID

Reapply resource limits on jail while it is running.

uncap UUID

Release all resource limits, disable limits on the fly.

inuse UUID

Display consumed resources for jail.

snapshot UUID

Create a ZFS snapshot for jail.

snaplist UUID

List all snapshots belonging to jail.

snapremove snapshotname UUID

Destroy snapshot.

defaults

Display all defaults set in iocage itself.

version | --version

List version number.

help

List quick help.

PROPERTIES

For more information on properties please check the relevant man page which is noted next to each *property*.

interfaces

By default there are two interfaces specified with their bridge association. Up to four interfaces are supported. Interface configurations are separated by commas.

Default: vnet0:bridge0,vnet1:bridge1

host_hostname

Default: UUID. See jail(8) for more details.

exec_fib

Default: 0. jail(8)

devfs_ruleset

Default: 4. jail(8)

mount_devfs

Default: 1. jail(8)

exec_start

Default: /bin/sh /etc/rc. jail(8)

exec_stop
Default: /bin/sh /etc/rc.shutdown. jail(8)

exec_prestart
Default: /usr/bin/true. jail(8)

exec_prestop
Default: /usr/bin/true. jail(8)

exec_poststop
Default: /usr/bin/true. jail(8)

exec_poststart
Default: /usr/bin/true. jail(8)

exec_clean
Default: 1. jail(8)

exec_timeout
Default: 60. jail(8)

stop_timeout
Default: 30. jail(8)

exec_jail_user
Default: root. jail(8)

exec_system_jail_user
Default: 0. jail(8)

exec_system_user
Default: root. jail(8)

mount_fdscfs
Default: 1. jail(8)

enforce_statfs
Default: 2. jail(8)

children_max
Default: 0. jail(8)

login_flags
Default: -f root. jail(8)

securelevel
Default: 3. jail(8)

allow_set_hostname

Default: 1. jail(8)

allow_sysvipc

Default: 0. jail(8)

allow_raw_sockets

Default: 0. jail(8)

allow_chflags

Default: 0. jail(8)

allow_mount

Default: 0. jail(8)

allow_mount_devfs

Default: 0. jail(8)

allow_mount_nullfs

Default: 0. jail(8)

allow_mount_procfs

Default: 0. jail(8)

allow_mount_tmpfs

Default: 0. jail(8)

allow_mount_zfs

Default: 0. jail(8)

allow_quotas

Default: 0. jail(8)

allow_socket_af

Default: 0. jail(8)

host_hostuuid

Default: UUID. jail(8)

tag

Custom string for aliasing jails.

Default: date@time

template

This property controls whether the jail is a template. Templates are not auto-started by iocage.

Default: no

boot

If set to on jail will be auto-started at boot time.

Default: off

notes

Custom notes.

Default: none

owner

The owner of the jail, can be any string.

Default: root.

priority

Start priority at boot time, smaller value means higher priority.

Default: 99.

last_started

Last successful start time.

type

Currently only jail is supported - this is for future use.

Default: jail

hostid

The UUID of the host node. Jail won't start if this property differs from the actual UUID of the host node. This is to safeguard jails from being started on different nodes in case they are periodically replicated across nodes.

Default: UUID of the host (/etc/hostid)

release

The RELEASE used at creation time.

compression

Default: lz4. zfs(8)

origin

This is only set for clones. zfs(8)

quota

Quota for jail.

Default: 15G. zfs(8)

mountpoint

Path for the jail's root filesystem.

Default: set to jail's root. zfs(8)

compressratio

Compression ratio. zfs(8)

available

Available space in jail's dataset. zfs(8)

used

Used space by jail. zfs(8)

dedup

Deduplication for jail.

Default: off. zfs(8)

reservation

Reserved space for jail.

Default: none. zfs(8)

sync_target

This is for future use, currently not supported.

sync_tgt_zpool

This is for future use, currently not supported.

rlimits

If on resource limits will be enforced.

Default: off

cpuset

CPU affinity and limits. Please consult cpuset(1) for details.

Default: off

RESOURCE LIMITS

Resource limits (except cpuset and rlimits) use the following value field formatting in the *property*:
limit:action - to better understand what this means please read **rctl**(8) before enabling any limits.

The following resource limits are supported:

cpuset

Default: off. cpuset(1) for more details.

memoryuse

Default: 8G:log. rctl(8)

memorylocked
Default: off. rctl(8)

vmemoryuse
Default: off. rctl(8)

maxproc
Default: off. rctl(8)

cputime
Default: off. rctl(8)

pcpu
Default: off. rctl(8)

datasize
Default: off. rctl(8)

stacksize
Default: off. rctl(8)

coredumpsize
Default: off. rctl(8)

openfiles
Default: off. rctl(8)

pseudoterminals
Default: off. rctl(8)

swapuse
Default: off. rctl(8)

nthr
Default: off. rctl(8)

msgqueued
Default: off. rctl(8)

msgqsize
Default: off. rctl(8)

nmsgq
Default: off. rctl(8)

nsemop
Default: off. rctl(8)


```
nshm
    Default: off. rctl(8)

shmsize
    Default: off. rctl(8)

wallclock
    Default: off. rctl(8)
```

EXAMPLES

Set up **iocage** from scratch:

```
iocage fetch
```

Create first jail:

```
iocage create
```

List jails:

```
iocage list
```

Start jail:

```
iocage start UUID
```

Turn on resource limits and apply them:

```
iocage set rlimits=on UUID
iocage cap UUID
```

Display resource usage:

```
iocage inuse UUID
```

HINTS

iocage marks a ZFS pool in the pool's comment field and identifies the active pool for use based on this string.

Don't forget to add the node's physical NIC into one of the bridges if you need outside connection. Also read **bridge(4)** how traffic is handled if you are not familiar with this concept (in a nutshell: bridge behaves like a network switch).

PF firewall is not supported inside VNET jails as of July 2014. PF can be enabled for the host however. IPFW is fully supported.

Property validation is not handled by **iocage** (to keep it simple) so please make sure your custom values are supported before configuring any properties.

The actual jail name in the **jls(8)** output is *set* to *ioc-UUID*. This is a required workaround as jails will refuse to *start* with **jail(8)** when name starts with a "0".

SEE ALSO

jail(8), **ifconfig(8)**, **epair(4)**, **bridge(4)**, **jexec(8)**, **zfs(8)**, **zpool(8)**, **rctl(8)**, **cpuset(1)**

BUGS

Nothing is perfect, please kindly report them.

AUTHORS

Peter Toth <peter.toth198@gmail.com>