

-a|--abstract-socket

force UNIX socket in abstract mode (Linux only)

--add-header

automatically add HTTP headers to response

--alarm

create a new alarm, syntax: <alarm> <plugin:args>

--alarm-freq

tune the anti-loop alarm system (default 3 seconds)

--alarm-list

list enabled alarms

--allowed-modifiers

comma separated list of allowed modifiers

--app

*** deprecated ***

-A|--sharedarea

create a raw shared memory area of specified pages

--async

enable async mode with specified cores

--attach-daemon

attach a command/daemon to the master process (the command has to not go in background)

--autoload

try to automatically load plugins when unknown options are found

--auto-procname

automatically set processes name to something meaningful

--auto-snapshot

automatically make workers snapshot after reload

--auto-weight

set weight of the instance (used by clustering/lb/subscriptions) automatically

--backtrace-depth

set backtrace depth

-b|--buffer-size

set internal buffer size

--binary-path

force binary path

--cache-blocksize

set cache blocksize

--cache

create a shared cache containing given elements

--cache-expire-freq

set the frequency of cache sweeper scans (default 3 seconds)

--cache-no-expire

disable auto sweep of expired items

--cache-report-freed-items

constantly report the cache item freed by the sweeper (use only for debug)

--cache-server

enable the threaded cache server

--cache-server-threads

set the number of threads for the cache server

--cache-store

enable persistent cache to disk

--cache-store-sync

set frequency of sync for persistent cache

--callable

set default WSGI callable name

--cap

set process capability

--carbon-freq

set carbon push frequency in seconds (default 60)

--carbon-id

set carbon id

--carbon-max-retry

set maximum number of retries in case of connection errors (default 1)

--carbon-no-workers

disable generation of single worker metrics

--carbon

push statistics to the specified carbon server

--carbon-retry-delay

set connection retry delay in seconds (default 7)

--carbon-timeout

set carbon connection timeout in seconds (default 3)

--catch-exceptions

report exception has http output (discouraged)

-c|--cgi-mode

force CGI-mode for plugins supporting it

-C|--chmod-socket

chmod-socket

--cflags

report uWSGI CFLAGS (useful for building external plugins)

--cgroup-opt

set value in specified cgroup option

--cgroup

put the processes in the specified cgroup

--chdir2

chdir to specified directory after apps loading

--chdir

chdir to specified directory before apps loading

--cheaper-algo

choose to algorithm used for adaptive process spawning)

--cheaper-algo-list

list enabled cheapers algorithms

--cheaper-initial

set the initial number of processes to spawn in cheaper mode

--cheaper-overload

increase workers after specified overload

--cheaper

set cheaper mode (adaptive process spawning)

--cheaper-step

number of additional processes to spawn at each overload

--cheap

set cheap mode (spawn workers only after the first request)

--check-cache

check for response data in the cache

--check-interval

set the interval (in seconds) of master checks

--check-static

check for static files in the specified directory

--check-static-docroot

check for static files in the requested DOCUMENT_ROOT

--chown-socket

chown unix sockets

--chroot

chroot() to the specified directory

--clock-list

list enabled clocks

--clock

set a clock source

--close-on-exec

set close-on-exec on sockets (could be required for spawning processes in requests)

--cluster

join specified uWSGI cluster

--cluster-log

send a log line to the cluster

--cluster-nodes

get nodes list from the specified cluster

--cluster-reload

send a reload message to the cluster

--command-mode

force command mode

--connect-and-read

connect to a socket and wait for data from it

--cpu-affinity

set cpu affinity

--cron

add a cron task

--daemonize2

daemonize uWSGI after app loading

-d|--daemonize

daemonize uWSGI

--declare-option

declare a new uWSGI custom option

--die-on-idle

shutdown uWSGI when idle

--die-on-term

exit instead of brutal reload on SIGTERM

--disable-write-exception

disable exception generation on write()/writev()

--dlopen

blindly load a shared library

--dump-options

dump the full list of available options

--early-emperor

spawn the emperor as soon as possible

--emperor-broodlord

run the emperor in BroodLord mode

--emperor-freq

set the Emperor scan frequency (default 3 seconds)

--emperor-magic-exec

prefix vassals config files with exec:// if they have the executable bit

--emperor-max-throttle

set max throttling level (in milliseconds) for bad behaving vassals (default 3 minutes)

--emperor-pidfile

write the Emperor pid in the specified file

--emperor-procname

set the Emperor process name

--emperor-required-heartbeat

set the Emperor tolerance about heartbeats

--emperor

run the Emperor

--emperor-stats

run the Emperor stats server

--emperor-throttle

set throttling level (in milliseconds) for bad behaving vassals (default 1000)

--emperor-tyrant

put the Emperor in Tyrant mode

--endfor

(opt logic) end for cycle

--endif

(opt logic) end if

--env

set environment variable

--eval

eval python code

--evil-reload-on-as

force the master to reload a worker if its address space is higher than specified megabytes

--evil-reload-on-rss

force the master to reload a worker if its rss memory is higher than specified megabytes

--exec-as-root

run the specified command before privileges drop

--exec-as-user-atexit

run the specified command before app exit and reload

--exec-as-user

run the specified command after privileges drop

--exec-in-jail

run the specified command in jail after initialization

--exec-post-jail

run the specified command after jailing

--exec-pre-app

run the specified command before app loading

--exec-pre-jail

run the specified command before jailing

--exit-on-reload

force exit even if a reload is requested

--extract

fetch/dump any supported address to stdout

--farm

add a mule farm

--fastcgi-socket

bind to the specified UNIX/TCP socket using FastCGI protocol

--fastrouter-cheap

run the fastrouter in cheap mode

--fastrouter-events

set the maximum number of concurrent events

--fastrouter-fallback

fallback to the specified node in case of error

--fastrouter-gracetime

retry connections to dead static nodes after the specified amount of seconds

--fastrouter-harakiri

enable fastrouter harakiri

--fastrouter-post-buffering-dir

put fastrouter buffered files to the specified directory

--fastrouter-post-buffering

enable fastrouter post buffering

--fastrouter-processes

prefork the specified number of fastrouter processes

--fastrouter-quiet

do not report failed connections to instances

--fastrouter

run the fastrouter on the specified port

--fastrouter-stats

run the fastrouter stats server

--fastrouter-subscription-server

run the fastrouter subscription server on the specified address

--fastrouter-timeout

set fastrouter timeout

--fastrouter-to

forward requests to the specified uwsgi server (you can specify it multiple times for load balancing)

--fastrouter-use-base

use a base dir for fastrouter hostname->server mapping

--fastrouter-use-cache

use uWSGI cache as hostname->server mapper for the fastrouter

--fastrouter-use-cluster

load balance to nodes subscribed to the cluster

--fastrouter-use-code-string

use code string as hostname->server mapper for the fastrouter

er

--fastrouter-use-pattern

use a pattern for fastrouter hostname->server mapping

--fastrouter-use-socket

forward request to the specified uwsgi socket

--fastrouter-zerg

attach the fastrouter to a zerg server

--file-serve-mode

set static file serving mode

--flock2

lock the specified file after logging/daemon setup, exit if locked

--flock

lock the specified file before starting, exit if locked

--flock-wait2

lock the specified file after logging/daemon setup, wait if locked

--flock-wait

lock the specified file before starting, wait if locked

--forkbomb-delay

sleep for the specified number of seconds when a forkbomb is detected

--for

(opt logic) for cycle

--freebind

put socket in freebind mode

--ftok

set the ipcsem key via ftok() for avoiding duplicates

--gevent

a shortcut enabling gevent loop engine with the specified number of async cores and optimal parameters

--gid

setgid to the specified group/gid

--grunt

enable grunt mode (in-request fork)

--harakiri-no-arh

do not enable harakiri during after-request-hook

--harakiri-verbose

enable verbose mode for harakiri

--heartbeat

announce healthiness to the emperor

-h|--help

show this help

-H|--home

set PYTHONHOME/virtualenv

--http

add an http router/server on the specified address

--http-cheap

run the http router in cheap mode

--http-events

set the number of concurrent http async events

--http-harakiri

enable http router harakiri

--http-keepalive

experimental HTTP keepalive support (non-pipelined) requests (requires backend support)

--http-manage-expect

manage the Expect HTTP request header

--http-modifier1

set uwsgi protocol modifier1

--http-processes

set the number of http processes to spawn

--http-raw-body

blindly send HTTP body to backends (required for WebSockets and Icecast support)

--https

add an https router/server on the specified address with specified certificate and key

--https-export-cert

export uwsgi variable HTTPS_CC containing the raw client certificate

--http-socket

bind to the specified UNIX/TCP socket using HTTP protocol

--http-stats

run the http router stats server

--http-stud-prefix

expect a stud prefix (1byte family + 4/16 bytes address) on connections from the specified address

--http-subscription-server

enable the subscription server

--http-timeout

set internal http socket timeout

--http-to

forward requests to the specified node (you can specify it multiple times for lb)

--http-to-https

add an http router/server on the specified address and redirect all of the requests to https

--http-use-base

use the specified base for mapping requests to unix sockets

--http-use-cache

use uWSGI cache as key->value virtualhost mapper

--http-use-code-string

use code string as hostname->server mapper for the http router

--http-use-pattern

use the specified pattern for mapping requests to unix sockets

--http-var

add a key=value item to the generated uwsgi packet

--http-zerg

attach the http router to a zerg server

--idle

set idle mode (put uWSGI in cheap mode after inactivity)

--if-dir

(opt logic) check for directory existence

--if-env

(opt logic) check for environment variable

--if-exists

(opt logic) check for file/directory existence

--if-file

(opt logic) check for file existence

--if-opt

(opt logic) check for option

--if-reload

(opt logic) check for reload

--ignore-script-name

ignore SCRIPT_NAME

--ignore-sigpipe

do not report (annoying) SIGPIPE

--ignore-write-errors

do not report (annoying) write()/writev() errors

--imperial-monitor-list

list enabled imperial monitors

--import

import a python module

--include

include the specified file as immediate configuration

--inherit

use the specified file as config template

--ini

load config from ini file

--ini-paste

load a paste.deploy config file containing uwsgi section

--ini-paste-logged

load a paste.deploy config file containing uwsgi section (load loggers too)

-i|--single-interpreter

do not use multiple interpreters (where available)

-j|--json

load config from json file

-J|--test

test a module import

--ksm

enable Linux KSM

--lazy-apps

load apps in each worker instead of the master

--lazy

set lazy mode (load apps in workers instead of master)

--ldap

load configuration from ldap server

--ldap-schema

dump uWSGI ldap schema

--ldap-schema-ldif

dump uWSGI ldap schema in ldif format

-L|--disable-logging

disable request logging

--limit-as

limit processes address space/vsz

--limit-nproc

limit the number of spawnable processes

--limit-post

limit request body

-l|--listen

set the socket listen queue size

--lock-engine

set the lock engine

--locks

create the specified number of shared locks

--log-4xx

log requests with a 4xx response

--log-5xx

log requests with a 5xx response

--log-alarm

raise the specified alarm when a log line matches the specified regexp, syntax: <alarm>[,alarm...] <regexp>

--log-backupname

set logfile name after rotation

--log-big

log requestes bigger than the specified size

--logdate

prefix logs with date or a strftime string

--log-drain

drain (do not show) log lines matching the specified regexp

--logfile-chmod

chmod logfiles

--logfile-chown

chown logfiles

--log-filter

show only log lines matching the specified regexp

--log-format

set advanced format for request logging

--logformat-strftime

apply strftime to logformat output

--logger-list

list enabled loggers

--logger

set/append a logger

--log-master-bufsize

set the buffer size for the master logger. bigger log messages will be truncated

--log-master

delegate logging to master process

--log-maxsize

set maximum logfile size

--log-micros

report response time in microseconds instead of milliseconds

--log-prefix

prefix logs with a string

--log-reopen

reopen log after reload

--log-route

log to the specified named logger if regexp applied on logline matches

--log-sendfile

log sendfile requests

--log-slow

log requests slower than the specified number of milliseconds

--log-socket

send logs to the specified socket

--log-syslog

log to syslog

--logto2

log to specified file or udp address after privileges drop

--logto

set logfile/udp address

--log-truncate

truncate log on startup

--log-x-forwarded-for

use the ip from X-Forwarded-For header instead of REMOTE_ADDR

--log-zero

log responses without body

--log-zeromq

send logs to a zeromq server

--loop-list

list enabled loop engines

--loop

select the uWSGI loop engine

--manage-script-name

automatically rewrite SCRIPT_NAME and PATH_INFO

--map-socket

map sockets to specific workers

--master-as-root

leave master process running as root

--max-apps

set the maximum number of per-worker applications

--max-fd

set maximum number of file descriptors (requires root privileges)

--mimefile

set mime types file path (default /etc/mime.types)

-M|--master

enable master process

-m|--memory-report

enable memory report

--mode

set uWSGI custom mode

--mount

load application under mountpoint

--mule

add a mule

--mule-harakiri

set harakiri timeout for mule tasks

--mules

add the specified number of mules

--multicast

subscribe to specified multicast group

--multicast-ttl

set multicast ttl

--nagios

nagios check

--namespace-keep-mount

keep the specified mountpoint in your namespace

--namespace-net

add network namespace

--namespace

run in a new namespace under the specified rootfs

--need-app

exit if no app can be loaded

--never-swap

lock all memory pages avoiding swapping

--no-default-app

do not fallback to default app

--no-defer-accept

disable deferred-accept on sockets

--no-fd-passing

disable file descriptor passing

--no-initgroups

disable additional groups set via initgroups()

--no-orphans

automatically kill workers if master dies (can be dangerous for availability)

--no-server

force no-server mode

--no-site

do not import site module

--no-threads-wait

do not wait for threads cancellation on quit/reload

--offload-threads

set the number of offload threads to spawn (per-worker, default 0)

-O|--optimize

set python optimization level

--paste

load a paste.deploy config file

--paste-logger

enable paste fileConfig logger

--pause

pause an instance

--pcre-jit

enable pcre jit (if available)

--pidfile2

create pidfile (after privileges drop)

--pidfile

create pidfile (before privileges drop)

--ping

ping specified uwsgi host

--ping-timeout

set ping timeout

--plugins-dir

add a directory to uWSGI plugin search path

--plugins-list

list enabled plugins

--plugins

load uWSGI plugins

--post-buffering-bufsize

set buffer size for read() in post buffering mode

--post-buffering

enable post buffering

--post-pymodule-alias

add a python module alias after uwsgi module initialization

-p|--processes

spawn the specified number of workers/processes

--print

simple print

--prio

set processes/threads priority

--privileged-binary-patch-arg

patch the uwsgi binary with a new command and arguments (before privileges drop)

--privileged-binary-patch

patch the uwsgi binary with a new command (before privileges drop)

--procname-append

append a string to process names

--procname-master

set master process name

--procname-prefix

add a prefix to the process names

--procname-prefix-spaced

add a spaced prefix to the process names

--procname

set process names

--profiler

enable the specified profiler

--propagate-touch

over-engineering option for system with flaky signal management

--protocol

force the specified protocol for default sockets

--pump

uWSGI OPTIONS (1.4.5)

13 of 19

load a pump app

--pyargv

manually set sys.argv

--py-auto-reload-ignore

ignore the specified module during auto-reload scan (can be specified multiple times)

--py-auto-reload

monitor python modules mtime to trigger reload (use only in development)

--pymodule-alias

add a python alias module

--py-programname

set python program name

--pyshell-oneshot

run an interactive python shell in the uWSGI environment (one-shot variant)

--pyshell

run an interactive python shell in the uWSGI environment

--pythonpath

add directory (or glob) to pythonpath

--python

run a python script in the uWSGI environment

--python-version

report python version

--py-tracebacker

enable the uWSGI python tracebacker

-Q|--spooler

run a spooler on the specified directory

--queue-blocksize

set queue blocksize

--queue

enable shared queue

--queue-store

enable persistent queue to disk

--queue-store-sync

set frequency of sync for persistent queue

--rawrouter-cheap

run the rawrouter in cheap mode

--rawrouter-harakiri

enable rawrouter harakiri

--rawrouter-max-retries

set the maximum number of retries/fallbacks to other nodes

--rawrouter-processes

prefork the specified number of rawrouter processes

--rawrouter

run the rawrouter on the specified port

--rawrouter-stats

run the rawrouter stats server

--rawrouter-subscription-server

run the rawrouter subscription server on the specified address

--rawrouter-timeout

set rawrouter timeout

--rawrouter-use-base

use a base dir for rawrouter hostname->server mapping

--rawrouter-use-cache

use uWSGI cache as hostname->server mapper for the rawrouter

--rawrouter-use-code-string

use code string as hostname->server mapper for the rawrouter

--rawrouter-use-pattern

use a pattern for rawrouter hostname->server mapping

--rawrouter-xclient

use the xclient protocol to pass the client address

--rawrouter-zerg

attach the rawrouter to a zerg server

--reload-mercy

set the maximum time (in seconds) a worker can take to reload/shutdown

--reload-on-as

reload if address space is higher than specified megabytes

--reload-on-exception

reload a worker when an exception is raised

--reload-on-exception-repr

reload a worker when a specific exception type+value (language-specific) is raised

--reload-on-exception-type

reload a worker when a specific exception type is raised

--reload-on-exception-value

reload a worker when a specific exception value is raised

--reload-on-rss

reload if rss memory is higher than specified megabytes

--reload-os-env

force reload of os.environ at each request

--reload

reload an instance

--remap-modifier

remap request modifier from one id to another

--resume

resume an instance

--reuse-port

enable REUSE_PORT flag on socket (BSD only)

-R|--max-requests

reload workers after the specified amount of managed requests

--route

add a route

--route-host

add a route based on Host header

--route-qs

add a route based on QUERY_STRING

--router-list

list enabled routers

--route-uri

add a route based on REQUEST_URI

--rrdtool

collect requests data in the specified rrd file

--rrdtool-freq

set collect frequency

--rrdtool-max-ds

set maximum number of data sources

-r|--reaper

call waitpid(-1,...) after each request to get rid of zombies

--shared-import

import a python module in all of the processes

--shared-socket

create a shared sacket for advanced jailing or ipc

--show-config

show the current config reformatted as ini

--signal-bufsize

set buffer size for signal queue

--signal

send a uwsgi signal to a server

--skip-zero

skip check of file descriptor 0

--smart-attach-daemon2

attach a command/daemon to the master process managed by a pidfile (the command has to NOT daemonize)

--smart-attach-daemon

attach a command/daemon to the master process managed

by a pidfile (the command has to daemonize)

--snmp-community

set the snmp community string

--snmp

enable the embedded snmp server

--so-keepalive

enable TCP KEEPALIVES

--so-send-timeout

set SO_SNDTIMEO

--spooler-chdir

chdir() to specified directory before each spooler task

--spooler-external

map spoolers requests to a spooler directory managed by an external instance

--spooler-harakiri

set harakiri timeout for spooler tasks

--spooler-import

import a python module in the spooler

--spooler-max-tasks

set the maximum number of tasks to run before recycling a spooler

--spooler-ordered

try to order the execution of spooler tasks

--spooler-processes

set the number of processes for spoolers

--spooler-quiet

do not be verbose with spooler tasks

--sqlite3

load config from sqlite3 db

-S|--set

set a custom placeholder

--ssl-verbose

be verbose about SSL errors

-s|--socket

bind to the specified UNIX/TCP socket using default protocol

--start_response-nodelay

send WSGI http headers as soon as possible (PEP violation)

--static-expire-mtime

set the Expires header based on filename regexp and file mtime

--static-expire-path-info-mtime

set the Expires header based on PATH_INFO regexp and file mtime

--static-expire-path-info

set the Expires header based on PATH_INFO regexp

--static-expire

set the Expires header based on filename regexp

--static-expire-type-mtime

set the Expires header based on content type and file mtime

--static-expire-type

set the Expires header based on content type

--static-expire-uri-mtime

set the Expires header based on REQUEST_URI regexp and file mtime

--static-expire-uri

set the Expires header based on REQUEST_URI regexp

--static-index

search for specified file if a directory is requested

--static-map2

like static-map but completely appending the requested resource to the docroot

--static-map

map mountpoint to static directory (or file)

--static-skip-ext

skip specified extension from staticfile checks

--stats

enable the stats server on the specified address

--stats-http

prefix stats server json output with http headers

--stats-minified

minify statistics json output

--stats-pusher-default-freq

set the default frequency of stats pushers

--stats-push

push the stats json to the specified destination

--stop

stop an instance

--subscribe-freq

send subscription announce at the specified interval

--subscribe-to

subscribe to the specified subscription server

--subscription-algo

set load balancing algorithm for the subscription system

--subscription-dotsplit

try to fallback to the next part (dot based) in subscription key

--subscriptions-sign-check

set digest algorithm and certificate directory for secured subscription system

--subscriptions-sign-check-tolerance

set the maximum tolerance (in seconds) of clock skew for secured subscription system

--subscription-tolerance

set tolerance for subscription servers

--suspend

suspend an instance

-s|--uwsgi-socket

bind to the specified UNIX/TCP socket using uwsgi protocol

-T|--enable-threads

enable threads

-t|--harakiri

set harakiri timeout

--threaded-logger

offload log writing to a thread

--threads

run each worker in prethreaded mode with the specified number of threads

--thread-stacksize

set threads stacksize

--thunder-lock

serialize accept() usage (if possible)

--touch-logreopen

trigger log reopen if the specified file is modified/touched

--touch-logrotate

trigger logrotation if the specified file is modified/touched

--touch-reload

reload uWSGI if the specified file is modified/touched

--udp

run the udp server on the specified address

--ugreen

enable ugreen coroutine subsystem

--ugreen-stacksize

set ugreen stack size in pages

--uid

setuid to the specified user/uid

--umask

set umask

--underrferred-shared-socket

create a shared sacket for advanced jailing or ipc (underrferred mode)

--unenv

unset environment variable

--unprivileged-binary-patch-arg

patch the uwsgi binary with a new command and arguments (after privileges drop)

--unprivileged-binary-patch

patch the uwsgi binary with a new command (after privileges drop)

--unshare

unshare() part of the processes and put it in a new namespace

--unsubscribe-on-graceful-reload

force unsubscribe request even during graceful reload

--upload-progress

enable creation of .json files in the specified directory during a file upload

--vacuum

try to remove all of the generated file/sockets

--vassals-inherit

add config templates to vassals config

--vassal-sos-backlog

ask emperor for sos if backlog queue has more items than the value specified

--vassals-start-hook

run the specified command before each vassal starts

--vassals-stop-hook

run the specified command after vassals death

--version

print uWSGI version

--vhost

enable virtualhosting mode (based on SERVER_NAME variable)

--vhost-host

enable virtualhosting mode (based on HTTP_HOST variable)

-v|--max-vars

set the amount of internal iovec/vars structures

--web3

load a web3 app

--weight

weight of the instance (used by clustering/lb/subscriptions)

-w|--module

load a WSGI module

--worker-exec

run the specified command as worker

--worker-mount

load application under mountpoint in the specified worker or after workers spawn

--write-errors-exception-only

only raise an exception on write errors giving control to the app itself

--write-errors-tolerance

set the maximum number of allowed write errors (default: no tolerance)

--wsgi-env-behaviour

set the strategy for allocating/deallocating the WSGI env

--wsgi-file

load .wsgi file

--wsgi-lite

load a wsgi-lite app

-x|--xmlconfig

load config from xml file

-y|--yaml

load config from yaml file

--zerg

attach to a zerg server

--zerg-fallback

fallback to normal sockets if the zerg server is not available

--zergpool

start a zergpool on specified address for specified address

--zerg-server

enable the zerg server on the specified UNIX socket

--zeromq

create a zeromq pub/sub pair

-z|--socket-timeout

set internal sockets timeout