

NAME

un – execute code in secure sandboxed containers

SYNOPSIS

un [*OPTIONS*] *source_file*

un session [*OPTIONS*]

un service *command* [*OPTIONS*]

DESCRIPTION

un executes code in 42+ programming languages via the unsandbox.com API. Code runs in isolated LXD containers with configurable network access and resource limits.

The tool supports three modes of operation:

Code Execution

Run source files directly. Language is auto-detected from file extension or shebang.

Interactive Sessions

Open shell or REPL sessions with optional persistence via tmux/screen.

Services

Manage long-lived containers with exposed ports for running applications.

OPTIONS**General Options**

-e *KEY=VALUE*

Set environment variable. Can be used multiple times.

-f *FILE*

Add input file to the sandbox. Can be used multiple times.

-a

Return and save artifacts (compiled binaries, generated files).

-o *DIR* Output directory for artifacts (default: current directory).

-k *KEY*

API key. Can also be set via UNSANDBOX_API_KEY environment variable.

-n *MODE*

Network mode: **zerotrust** (default, isolated) or **semitrusted** (internet access).

-h

Show help message.

Session Options

-s, --shell *SHELL*

Shell or REPL to use (default: bash).

-l, --list

List active sessions.

--attach *ID*

Reconnect to existing session by ID or container name.

--kill *ID*

Terminate a session by ID or container name.

--audit

Record session for auditing. Saves terminal output to artifacts.

--tmux

Enable session persistence with tmux. Allows disconnect and reconnect.

--screen

Enable session persistence with screen. Allows disconnect and reconnect.

Service Options

- name** *NAME*
Service name (creates a new service when provided).
- ports** *PORTS*
Comma-separated ports to expose (e.g., 80,443).
- bootstrap** *CMD*
Bootstrap command, file path, or URL to run on startup.
- l, --list**
List all services.
- info** *ID*
Get service details.
- logs** *ID*
Get bootstrap logs.
- sleep** *ID*
Freeze a service (pause execution, preserve state).
- wake** *ID*
Unfreeze a service.
- destroy** *ID*
Destroy a service permanently.
- redploy** *ID*
Re-run bootstrap script (requires **--bootstrap**).

SHELLS AND REPLS**Available Shells**

bash, dash, sh, zsh, fish, ksh, tcsh, csh, elvish, xonsh, ash

Available REPLs

python3, bpython, ipython, node, ruby, irb, lua, php, perl, guile, ghci, erl, iex, sbcl, clisp, r, julia, clojure

SESSION BEHAVIOR**Default (no multiplexer)**

Session terminates immediately on disconnect. Clean exit, container released.

With **--tmux or **--screen****

Session persists on disconnect. Reconnect anytime with **--attach**.

MULTIPLEXER SHORTCUTS**tmux****Ctrl+b d**

Detach (disconnect, keep session alive)

Ctrl+b c

Create new window

Ctrl+b n/p

Next/previous window

Ctrl+b 0-9

Switch to window by number

Ctrl+b %

Split pane vertically

Ctrl+b

Split pane horizontally

Ctrl+b o

Switch between panes

screen**Ctrl+a d**

Detach (disconnect, keep session alive)

Ctrl+a c

Create new window

Ctrl+a n/p

Next/previous window

Ctrl+a 0-9

Switch to window by number

Ctrl+a S

Split horizontally

Ctrl+a |

Split vertically

Ctrl+a Tab

Switch between regions

SUPPORTED LANGUAGES

Interpreted: python, javascript, typescript, ruby, perl, php, lua, bash, r, elixir, erlang, tcl, scheme, powershell, clojure, commonlisp, crystal, groovy, deno, awk, raku

Compiled: c, cpp, go, rust, java, kotlin, cobol, fortran, d, zig, nim, v, objc, dart, julia, haskell, ocaml, fsharp, csharp, prolog, forth

Aliases: js/node → javascript, ts → typescript, lisp → commonlisp

RESOURCE LIMITS

CPU 1 vCPU per container

Memory

2 GB per container

Processes

500 per container

I/O Rate (interactive sessions only)

Interactive sessions via streaming API or CLI are terminated if stdin/stdout exceeds rate limits (prevents runaway processes).

ENVIRONMENT**UNSANDBOX_API_KEY**

API key for authentication. Required for most operations.

EXAMPLES**Code Execution**

```
# Execute Python script
un script.py
```

```
# With environment variable
un -e DEBUG=1 script.py
```

```
# With input file
un -f data.csv process.py
```

```
# Save compiled artifacts
un -a -o ./bin main.c
```

```
# With network access
un -n semitrusted fetch.py
```

Interactive Sessions

```
# Start bash session (terminates on disconnect)
un session
```

```
# Bash with tmux (can reconnect)
un session --tmux
```

```
# Python REPL with screen
un session --shell python3 --screen
```

```
# List active sessions
un session --list
```

```
# Reconnect to session
un session --attach unsb-vm-12345
```

```
# Terminate session
un session --kill unsb-vm-12345
```

```
# Audited session (records output)
un session --audit -o ./logs
```

Services

```
# Create web service
un service --name web --ports 80,443 \
  --bootstrap "python -m http.server 80"
```

```
# List services
un service --list
```

```
# Get service info/logs
un service --info abc123
un service --logs abc123
```

```
# Freeze/unfreeze
un service --sleep abc123
un service --wake abc123
```

```
# Destroy
un service --destroy abc123
```

```
# Redeploy with updated bootstrap
un service --redeploy abc123 --bootstrap ./setup.sh
```

EXIT STATUS

0	Success
1	General error
2	API error or network failure

FILES

/tmp/artifacts/
Default location for artifacts within containers.

SEE ALSO

tmux(1), screen(1), ssh(1)

AUTHOR

unsandbox@unturf.com

COPYRIGHT

Copyright © 2025 unturf.com. All rights reserved.