

**NAME**

**un** – execute code in secure sandboxed containers

**SYNOPSIS**

**un** [*OPTIONS*] *source\_file*  
**un** [*OPTIONS*] **-s** *LANG* '*code*'  
**un session** [*OPTIONS*]  
**un service** [*OPTIONS*]  
**un snapshot** [*OPTIONS*]  
**un image** [*OPTIONS*]  
**un languages** [**--json**]  
**un key** [**--extend**]

**DESCRIPTION**

**un** executes code in 42+ programming languages via the [unsandbox.com](https://unsandbox.com) API. Code runs in isolated LXD containers with configurable network access and resource limits.

The tool supports the following modes of operation:

**Code Execution**

Run source files or inline code. 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 automatic HTTPS. Each service gets a public URL at **NAME.on.unsandbox.com** with a free TLS certificate provisioned automatically via Let's Encrypt. Custom domains supported via CNAME.

**Snapshots**

Save and restore container state for sessions and services.

**Images** Publish, share, and spawn from container images.

**Languages**

List available execution languages.

**Key Management**

Validate API keys and extend their expiration.

**AUTHENTICATION**

All API requests use HMAC-SHA256 signature authentication with a public/secret key pair.

Credentials are loaded in priority order:

1. Command-line flags (**-p** and **-k**)
2. Environment variables (**UNSANDBOX\_PUBLIC\_KEY** and **UNSANDBOX\_SECRET\_KEY**)
3. Config file (**~/unsandbox/accounts.csv**)

The config file format is one key pair per line:

```
public_key,secret_key
```

Get API keys from <https://unsandbox.com/pricing>.

**OPTIONS****General Options**

**-s, --shell** *LANG*

Specify language. If the argument is not a file, the default is bash. When used with a non-file argument, the argument is treated as inline code.

- e** *KEY=VALUE*  
Set environment variable. Can be used multiple times.
- f** *FILE*  
Add input file to **/root/input/** in the sandbox. Can be used multiple times.
- F** *FILE*  
Add input file with its path preserved (not flattened to **/root/input/**).
- a**  
Return and save artifacts (compiled binaries, generated files).
- o** *DIR*  
Output directory for artifacts (default: current directory).
- p** *KEY*  
Public key for authentication. Can also be set via **UNSANDBOX\_PUBLIC\_KEY**.
- k** *KEY*  
Secret key for authentication. Can also be set via **UNSANDBOX\_SECRET\_KEY**.
- n** *MODE*  
Network mode: **zerotrust** (default, fully isolated) or **semitrusted** (internet access via egress proxy).
- v** *N*, **--vcpu** *N*  
vCPU count 1–8. Each vCPU gets 2GB RAM. Default: 1.
- y**  
Skip confirmation for large uploads (>1GB).
- 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.
- freeze** *ID*  
Freeze a session (pause execution, preserve state).
- unfreeze** *ID*  
Unfreeze a frozen session.
- 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.
- snapshot** *ID*  
Create a snapshot of a session.
- restore** *SNAPSHOT*  
Restore session from a snapshot.
- snapshot-name** *NAME*  
Name for the snapshot.

**--hot** Take snapshot without freezing (live snapshot).

### Service Options

**--name** *NAME*

Service name. Creates **https://NAME.on.unsandbox.com** with automatic TLS certificate.

**--ports** *PORTS*

Comma-separated ports to expose (e.g., 80,443).

**--domains** *DOMAINS*

Comma-separated custom domains. Add a CNAME record pointing to **on.unsandbox.com** and TLS certificates are provisioned automatically.

**--type** *TYPE*

Service type: minecraft, mumble, teamspeak, source, tcp, udp.

**--bootstrap** *CMD*

Bootstrap command to run on startup.

**--bootstrap-file** *FILE*

Upload local file as bootstrap script.

**-e, --env** *KEY=VAL*

Set environment variable (stored encrypted in vault).

**--env-file** *FILE*

Load environment variables from .env file (stored encrypted).

**-f** *FILE*

Upload file to /tmp/ in the service container.

**-l, --list**

List all services.

**--info** *ID*

Get service details.

**--tail** *ID*

Get last 9000 lines of bootstrap logs.

**--logs** *ID*

Get all bootstrap logs.

**--download-logs** *ID FILE*

Download all logs to a file.

**--execute** *ID CMD*

Run a command in a running service.

**--freeze** *ID*

Freeze a service (pause execution, preserve state).

**--unfreeze** *ID*

Unfreeze a service.

**--destroy** *ID*

Destroy a service permanently.

**--lock** *ID*

Lock a service to prevent accidental deletion.

**--unlock** *ID*

Unlock a service.

**--resize** *ID*

Resize a service (requires **--vcpu**).

- redeploy *ID***  
Re-run bootstrap script.
- snapshot *ID***  
Create a snapshot of a service.
- restore *SNAPSHOT***  
Restore service from a snapshot.

### Service Environment Vault

- env status *ID***  
Show vault status for a service.
- env set *ID***  
Set vault from **--env-file** or stdin.
- env export *ID***  
Export vault contents to stdout.
- env delete *ID***  
Delete the vault.

### Snapshot Options

- l, --list**  
List all snapshots.
- info *ID***  
Get snapshot details.
- delete *ID***  
Delete a snapshot.
- lock *ID***  
Lock snapshot to prevent deletion.
- unlock *ID***  
Unlock snapshot.
- clone *ID***  
Clone snapshot to a new session or service.
- type *TYPE***  
Clone type: **session** or **service** (used with **--clone**).
- name *NAME***  
Name for cloned service (used with **--clone**).
- shell *SHELL***  
Shell for cloned session (used with **--clone**).
- ports *PORTS***  
Ports for cloned service (used with **--clone**).

### Image Options

- l, --list**  
List all images.
- info *ID***  
Get image details.
- delete *ID***  
Delete an image.
- lock *ID***  
Lock image to prevent deletion.

- unlock** *ID*  
Unlock image.
- publish** *ID*  
Publish image from a service or snapshot (requires **--source-type**).
- source-type** *TYPE*  
Source type: **service** or **snapshot** (used with **--publish**).
- visibility** *ID VISIBILITY*  
Set image visibility: **private**, **unlisted**, or **public**.
- spawn** *ID*  
Spawn a new service from an image.
- clone** *ID*  
Clone an image.
- name** *NAME*  
Name for spawned service or cloned image.
- ports** *PORTS*  
Ports for spawned service.

**Languages Options****(no options)**

List all available languages, one per line.

- json** Output as JSON array.

**Key Options****(no options)**

Validate API key and display status, tier, rate limits, and expiration.

- extend**  
Open browser to extend API key expiration.

**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, swift, scala

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

Use **un languages** to see the current list from the API.

**NETWORK MODES****zerotrust**

Default. No network access. Container is fully isolated.

**semitrusted**

Internet access through egress proxy. Container can reach external services.

**RESOURCE LIMITS**

**CPU** 1 vCPU per container (configurable up to 8 with **-v**).

**Memory**

2 GB per vCPU (e.g., 4 vCPUs = 8 GB RAM).

**Processes**

500 per container.

**I/O Rate**

Interactive sessions are terminated if stdin/stdout exceeds rate limits.

**ENVIRONMENT****UNSANDBOX\_PUBLIC\_KEY**

Public key for HMAC authentication (format: unsb-pk-xxxx-xxxx-xxxx-xxxx).

**UNSANDBOX\_SECRET\_KEY**

Secret key for HMAC authentication (format: unsb-sk-xxxxx-xxxxx-xxxxx-xxxxx).

**FILES**

**~/unsandbox/accounts.csv**  
Credential store. One public\_key,secret\_key pair per line.

**~/unsandbox/languages.json**  
Cached language list (refreshed hourly).

**/root/input/**  
Default location for input files within containers.

**/root/artifacts/**  
Default location for output artifacts within containers.

**EXAMPLES****Code Execution**

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

# Execute inline bash command
un -s bash 'echo hello'

# 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

# Multi-vCPU (4 vCPUs, 8GB RAM)
un -v 4 heavy_computation.py
```

**Interactive Sessions**

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

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

# Python REPL with network access
un session -s python3 -n semitrusted

# List active sessions
un session --list

# Reconnect to a session
un session --attach sandbox-abc123

# Terminate a session
un session --kill sandbox-abc123

# Snapshot a session
un session --snapshot sandbox-abc123 --snapshot-name before-changes
```

**Services**

```
# Create a web service (live at https://myapp.on.unsandbox.com)
un service --name myapp --ports 80 \
  --bootstrap "python -m http.server 80"

# List all services
un service --list

# Get service info
un service --info srv-abc123

# View logs
un service --tail srv-abc123

# Execute command in a service
un service --execute srv-abc123 'ls -la'

# Freeze/unfreeze
un service --freeze srv-abc123
un service --unfreeze srv-abc123

# Destroy a service
un service --destroy srv-abc123

# Set environment variables
un service env set srv-abc123 --env-file .env
```

**Snapshots**

```
# List all snapshots
un snapshot --list

# Clone snapshot to a new service
un snapshot --clone snap-abc123 \
  --type service --name myapp --ports 80

# Clone snapshot to a new session
un snapshot --clone snap-abc123 \
  --type session --shell bash

# Delete a snapshot
un snapshot --delete snap-abc123
```

**Images**

```
# List all images
un image --list

# Publish an image from a service
un image --publish srv-abc123 \
  --source-type service --name my-image

# Set image visibility
un image --visibility img-abc123 public

# Spawn a new service from an image
un image --spawn img-abc123 \
  --name new-service --ports 80,443
```

```
# Clone an image
un image --clone img-abc123 --name my-copy
```

**Languages**

```
# List all supported languages
un languages
```

```
# Output as JSON
un languages --json
```

**Key Management**

```
# Check API key status
un key
```

```
# Extend key expiration (opens browser)
un key --extend
```

**EXIT STATUS**

```
0      Success
1      General error
2      API error or network failure
```

**SEE ALSO**

**tmux(1)**, **screen(1)**, **ssh(1)**, **curl(1)**

**BUGS**

Report bugs at <https://github.com/russellballestrini/un-inception/issues>.

**AUTHOR**

unsandbox@unturf.com

**COPYRIGHT**

Copyright © 2025–2026 unturf.com. All rights reserved.