Volta Node toolchain manager

START HERE

`volta` **`install node`**
Default Node (latest LTS)

`volta` **`which`** *`<binary>`*
Real binary behind a shim

`volta` **`install node@`***`<version>`*
Default Node, pinned

`volta` **`setup`**
Fix PATH setup

`volta` **`pin node@`***`<version>`*
Pin Node in project

`volta` **`pin npm@`***`<version>`*
Pin npm in project

`volta` **`list`**
What is active?

`volta` **`run --node`** *`<version>`* *`<cmd>`*
One-off other Node

node / npm / tsc you type a command → `shim on PATH` → **Volta shim** in ~/.volta/bin → `looks up` → **package.json "volta"** else your default → `runs` → **Pinned binary** right Node engine

CONCEPTS

Volta swaps tool versions per directory: a pin in package.json wins, otherwise your default applies. No shell hook or use command needed.

Global packages are pinned to the Node they were installed with and keep it until you reinstall them.

Commit the "volta" block in package.json so the whole team and CI get the same toolchain.

INSTALL VOLTA

`curl https://get.volta.sh | bash`
Install on Linux, macOS, WSL

`winget install Volta.Volta`
Install on Windows

`volta` **`setup`**
Add shims to PATH for your user
Edits existing shell profiles and sets VOLTA_HOME. The Unix installer already runs it.

`volta` **`--version`**
Show Volta version

`volta` **`completions`** *`<shell>`*
Print shell completions
Shells: bash, zsh, fish, powershell, elvish. Add -o <file> to write, -f to overwrite.

INSTALL VOLTA (cont.)

Manual setup: export VOLTA_HOME=$HOME/.volta and put $VOLTA_HOME/bin first on PATH.

DEFAULT TOOLS

`volta` **`install node`**
Default to the latest LTS Node

`volta` **`install node@`***`<version>`*
Default to an exact or partial version
node@22 picks a matching release; node@22.5.1 is exact.

`volta` **`install npm@`***`<version>`*
Default npm instead of the bundled one

`volta` **`install yarn@`***`<version>`*
Default Yarn

`volta` **`install`** *`<package>`*
Install a global package binary
Also works as npm install --global; Volta tracks the binaries either way.

`volta` **`install`** *`<package>`***`@`***`<range>`*
Global package at a semver range

`volta` **`uninstall`** *`<package>`*
Remove a global package or tool

DEFAULT TOOLS (cont.)

`volta` **`fetch`** *`<tool>`***`@`***`<version>`*
Download to cache only, no default
Not put on PATH; useful to warm a cache for offline use.

install changes your default everywhere outside pinned projects; use pin for a single project.

PIN PER PROJECT

`volta` **`pin node@`***`<version>`*
Pin Node in package.json

`volta` **`pin node@lts`**
Pin the current LTS Node

`volta` **`pin npm@`***`<version>`*
Pin npm

`volta` **`pin yarn@`***`<range>`*
Pin Yarn

`volta` **`pin pnpm@`***`<version>`*
Pin pnpm (needs VOLTA_FEATURE_PNPM=1)

What pin writes

```
"volta": {
  "node": "20.16.0",
  "npm": "10.8.1"
}
```

pin only handles Node and package managers. Change library versions with npm install or yarn add.

INSPECT

`volta` **`list`**
Show the active toolchain

`volta` **`list all`**
Everything installed

`volta` **`list node`**
Installed Node versions

`volta` **`list --default`**
Show your defaults only

`volta` **`list`** *`<package>`*
Look up a package or binary

`volta` **`list --format plain`**
Script-friendly output
Plain is the default when stdout is not a terminal.

`volta` **`which`** *`<binary>`*
Real path behind a shim

RUN ONCE

`volta` **`run --node`** *`<version>`* *`<cmd>`*
Run with another Node
Version flags go before the command; everything after it is passed to the command.

`volta` **`run --npm`** *`<version>`* *`<cmd>`*
Run with another npm

`volta` **`run --yarn`** *`<version>`* *`<cmd>`*
Run with another Yarn

`volta` **`run --bundled-npm`** *`<cmd>`*
Use the npm bundled with Node

RUN ONCE (cont.)

`volta` **`run --no-yarn`** *`<cmd>`*
Run with Yarn disabled

`volta` **`run --env`** *`<NAME=value>`* *`<cmd>`*
Set an env var for the run

MONOREPOS AND MIRRORS

Share pins with extends

```
# packages/cli/package.json
"volta": { "extends": "../../
package.json" }
```

With extends, settings merge and the current file wins; volta pin writes to the nearest package.json.

`~/.volta/hooks.json`
System-wide hooks file (mirrors, proxies)
Per project: <root>/.volta/hooks.json. Hooks cover node, npm and yarn: index, latest, distro.

`VOLTA_FEATURE_PNPM=1`
Enable experimental pnpm support
Global pnpm installs are unsupported; uninstall an old Volta-managed pnpm first.

COMMON OPTIONS

`--verbose`
More diagnostics (any command)

COMMON OPTIONS (cont.)

`--quiet`
Less output (any command)

`volta` **`help`** *`<command>`*
Help for one command

`rm -rf ~/.volta`
Remove Volta (Unix)
Then delete the two Volta lines from your shell profiles and open a new terminal.

RECIPES

Set up a project

```
cd my-app
volta pin node@lts
volta pin npm@<version>
git add package.json && git
commit -m "Pin toolchain"
```

Test on another Node

```
volta run --node 18 npm test
volta run --node 22 npm test
# default and pin stay
unchanged
```

Fresh machine to working Node

```
curl https://get.volta.sh |
bash
# open a new terminal
volta install node
volta list
```

Debug the wrong version

```
volta list
volta which node
# project pin in package.json
beats the default
```