

using remote SSH with Codex app

ai

software-engineering

for a while, the answer to "can Codex app work against a remote box over SSH?" was basically "not yet." that changed on **April 16, 2026**: OpenAI announced alpha support for remote devboxes over SSH, and the docs now have a real setup page.

if you want the clean path now, do this:

1. add a real host alias to `~/.ssh/config`
2. make sure `ssh your-alias` works from the same machine running Codex
3. install and authenticate `codex` on the remote host
4. open `Settings > Connections` in the app and choose the remote project folder

one important detail from the docs: the app bootstraps the remote side through your remote user's login shell. so if `codex` is not on `PATH` there, or your shell init does something weird, the connection can fail even though plain SSH works.

[if the Connections menu still doesn't show up](#)

the issue thread is still useful here. the official docs say SSH remote connections are in alpha and rolling out gradually, so people are clearly seeing different states in stable builds.

the two workarounds showing up in the latest comments are:

- add this to `~/.codex/config.toml`, then restart the app:

toml

```
[features]
remote_connections = true
```

- if that still does not do it, use the beta appcast build once. the shared feed is `https://persistent.oaistatic.com/codex-app-beta/appcast.xml`, and on April 16, 2026 it was serving `26.415.20818`. a few people reported that they could add the SSH connection in beta and keep using it from stable afterward.

two gotchas that seem real

first: keep the remote `codex` CLI updated. one of the latest comments specifically calls out `0.121.0` on the remote machine as important for recovering connections cleanly.

second: watch your shell init files. one tester traced `ECONNRESET` and websocket `1006` failures to a `~/.bashrc` that always did `exec zsh -l`. because Codex starts the remote side with a login interactive shell, that extra handoff can break bootstrap. guarding the handoff with `BASH_EXECUTION_STRING` fixed it:

```
>_ bash

if [ -x "$(command -v zsh)" ] && [ -z "${BASH_EXECUTION_STRING:-}" ]; then
  exec zsh -l
fi
```

that's the current state

so yeah, remote SSH in Codex app is finally real. the official path exists now, but the issue comments are still worth reading because they capture the messy part of a live alpha rollout: hidden flags, beta builds, and shell-init edge cases.

also: stick to the docs' network model. use SSH port forwarding or a VPN / Tailscale path, and do not expose the remote app server directly to the internet.

SOURCES

[OpenAI Codex remote connections docs \(https://developers.openai.com/codex/remote-connections\)](https://developers.openai.com/codex/remote-connections)

-

OpenAI: Codex for (almost) everything (<https://openai.com/index/codex-for-almost-everything/>)

OpenAI announcement comment on issue #10450 (<https://github.com/openai/codex/issues/10450#issuecomment-4262271609>)

GitHub issue #10450 (<https://github.com/openai/codex/issues/10450>)

Codex beta appcast (<https://persistent.oaistatic.com/codex-app-beta/appcast.xml>)