Step-by-step: install from scratch: AWS CDK 2.0

Quick facts (why these matter)

- CDK v2 requires **Node.js 22.x or later** (use an Active LTS). (AWS Documentation)
- The CDK CLI is installed via npm (`aws-cdk`) — you can install globally or run it per-project with `npx` . (AWS Documentation)
- Before you `cdk deploy` you must **bootstrap** your AWS environment (creates the S3/ECR assets bucket etc.). (AWS Documentation)
- When authoring code you add the CDK library to your project (`aws-cdk-lib` / `constructs` for most languages). (AWS Documentation)
- The CDK CLI uses whatever AWS credentials your AWS CLI is configured with (`aws configure` / SSO). Verify credentials first. (AWS Documentation)

Step-by-step: install from scratch

1) Install Node.js (recommended: use nvm so you don't need `sudo`)

(Use Node 22.x+ as required.)

Linux / macOS (recommended via nvm)

```
# install nvm (if not already)
curl -fsSL https://raw.githubusercontent.com/nvm-sh/nvm/master/install.sh | bash
# reload shell (or open a new terminal)
export NVM_DIR="$HOME/.nvm"; [ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"
# install Node 22 (LTS major)
nvm install 22
nvm use 22
node -v    # should show v22.x.x
npm -v
```

Windows

- Either download & run the Windows installer from Node.js — Run JavaScript Everywhere (choose current LTS / 22.x), **or** install **nvm-windows** (recommended if you switch versions frequently).
- Then confirm: `node -v` and `npm -v` .

(Why nvm? avoids global `sudo` installs and makes switching Node versions easy.)

> Source: CDK docs require Node.js 22.x or later. (AWS Documentation)

---

2) Install (or update) the AWS CLI and configure credentials

You must have working AWS credentials before deploying with CDK.

```
# verify AWS CLI is installed
aws --version || echo "install aws cli (see AWS docs)"
```

```

# configure credentials (interactive)
aws configure
# or, for SSO:
aws configure sso
```

Place valid credentials or configure SSO/profiles as required. The CDK CLI will use your AWS CLI credentials (default profile) unless you pass `--profile`. (AWS Documentation)

---

3) Install the CDK CLI (two options)

Option A — Global install (simple)

```
# global install (latest CDK v2 by default)
npm install -g aws-cdk

# verify
cdk --version
```

If you get permission errors on Linux, the docs show `sudo npm install -g aws-cdk` as an option — but prefer nvm to avoid `sudo`. (AWS Documentation)

Option B — Per-project (preferred for multiple versions)

In each project, install locally and run with `npx`:

```
# in project dir
npm install --save-dev aws-cdk
# then run:
npx aws-cdk --version

```

Using `npx`/local installs is safer for multi-version projects. (AWS Documentation)

---

4) Verify the CLI

```
cdk --version
```

If it prints a version number, CLI is installed. (AWS Documentation)

---

5) Bootstrap your AWS environment (required before deploying)

Run from your local machine with the credentials/profile you plan to deploy from:

```
# default profile, default region in AWS config
cdk bootstrap

# or specify account/region and profile
cdk bootstrap aws://ACCOUNT_ID/REGION --profile myprofile

```

Bootstrapping creates the CDKToolkit stack (S3 bucket, ECR repo, roles) needed for asset deployments. Do this for every account/region you intend to deploy to. (AWS Documentation)

---

6) Create your first app (TypeScript & Python examples)

TypeScript (recommended for examples)

```
mkdir my-cdk-app && cd my-cdk-app
```

```
cdk init app --language typescript

# (cdk init creates package.json; then)
npm install aws-cdk-lib constructs
# build / synth
npm run build       # generated by init
cdk synth
# preview diff
cdk diff
# deploy
cdk deploy --profile myprofile

```

The CDK library to import in TS is `aws-cdk-lib`. (AWS Documentation)

Python

```
mkdir my-cdk-py && cd my-cdk-py
cdk init app --language python

# cdk init will create a .venv; if not:
python -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt   # OR
pip install aws-cdk-lib constructs
cdk synth
cdk deploy

```

`cdk init` scaffolds requirements; for Python you need Python 3.7+. (AWS Documentation)

---

7) Useful CDK commands (quick reference)

- `cdk synth` — synthesize CloudFormation template.
- `cdk diff` — show differences between local and deployed stacks.
- `cdk deploy` — deploy stack to AWS (uses your AWS CLI creds/profile).
- `cdk bootstrap` — prepare environment for deployments (S3 + ECR + roles). (AWS Documentation)

---

8) Common gotchas & troubleshooting

- **Permission error on `npm install -g`** → use `nvm` or run `sudo npm install -g aws-cdk` (docs mention sudo as a fallback). (AWS Documentation)
- **“Unable to resolve AWS account to use”** → means your AWS profile/region/credentials aren’t set or passed; check `aws configure list` and use `--profile`. (Common fix: specify `--profile` when bootstrapping/deploying.) (Stack Overflow)
- **SSM parameter /cdk-bootstrap/... not found** → you haven’t bootstrapped that account/region; run `cdk bootstrap` for that environment. (Repost)