**Project Name: 13 - Using Caching**

**Overview**

This project demonstrates the use of **caching in GitHub Actions** to optimize CI/CD workflows for a React application. The workflow:

* Calculates a unique cache key based on the package-lock.json file.
* Conditionally downloads cached dependencies using the actions/cache action.
* Installs dependencies only when the cache is not available.
* Builds, tests, and lints the project in separate jobs to showcase modular workflow design.

*(****Include a screenshot of the Actions tab showing the triggered workflow and jobs here.****)*

**Workflow File**

Below is the complete YAML configuration for the caching workflow:

yaml

Copy code

name: 13 - Using Caching

on:

workflow\_dispatch:

inputs:

use-cache:

description: Whether to execute cache step

type: boolean

default: true

node-version:

description: Node version

type: choice

options:

- 18.x

- 20.x

- 21.x

default: 20.x

jobs:

install-deps:

runs-on: ubuntu-latest

defaults:

run:

working-directory: 13-caching/react-app

outputs:

deps-cache-key: ${{ steps.cache-key.outputs.CACHE\_KEY }}

steps:

- name: Checkout code

uses: actions/checkout@v4

- name: Setup Node

uses: actions/setup-node@v3

with:

node-version: ${{ github.event.inputs.node-version }}

- name: Calculate cache key

id: cache-key

run: |

echo "CACHE\_KEY=deps-node-modules-${{ hashFiles('13-caching/react-app/package-lock.json') }}" >> "$GITHUB\_OUTPUT"

- name: Download cached dependencies

uses: actions/cache@v3

if: ${{ github.event.inputs.use-cache }}

id: cache

with:

path: 13-caching/react-app/node\_modules

key: deps-node-modules-${{ hashFiles('13-caching/react-app/package-lock.json') }}

- name: Install dependencies

if: steps.cache.outputs.cache-hit != 'true'

run: npm ci

build:

runs-on: ubuntu-latest

needs: install-deps

defaults:

run:

working-directory: 13-caching/react-app

steps:

- name: Checkout code

uses: actions/checkout@v4

- name: Setup Node

uses: actions/setup-node@v3

with:

node-version: ${{ github.event.inputs.node-version }}

- name: Download cached dependencies

uses: actions/cache@v3

if: ${{ github.event.inputs.use-cache }}

id: cache

with:

path: 13-caching/react-app/node\_modules

key: ${{ needs.install-deps.outputs.deps-cache-key }}

- name: Testing

run: npm run test

- name: Linting

run: echo "Linting..."

lint-test:

runs-on: ubuntu-latest

needs: install-deps

defaults:

run:

working-directory: 13-caching/react-app

steps:

- name: Checkout code

uses: actions/checkout@v4

- name: Setup Node

uses: actions/setup-node@v3

with:

node-version: ${{ github.event.inputs.node-version }}

- name: Download cached dependencies

uses: actions/cache@v3

if: ${{ github.event.inputs.use-cache }}

id: cache

with:

path: 13-caching/react-app/node\_modules

key: ${{ needs.install-deps.outputs.deps-cache-key }}

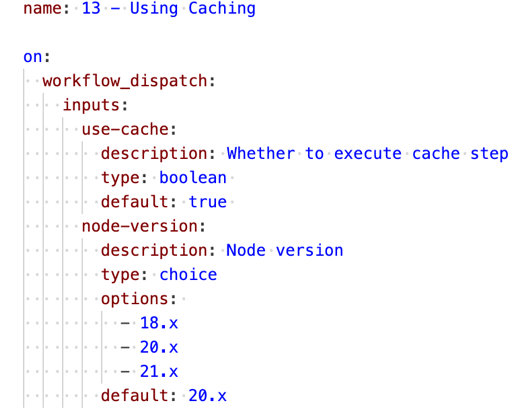
- name: Building

run: npm run build

*(****Include a screenshot of your repository's YAML file view in GitHub for reference here.****)*

**Workflow Breakdown**

1. **Trigger Events**
   * The workflow starts using workflow\_dispatch.
   * Takes user inputs (use-cache and node-version) for dynamic configuration.

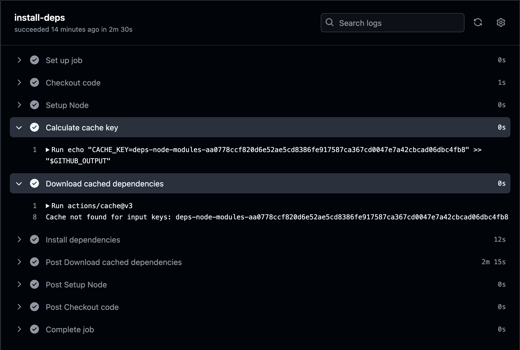


1. **Install Dependencies Job (install-deps)**

Responsibilities:

* + Checks out the code.
  + Sets up Node.js with the user-selected version.
  + Calculates a cache key.
  + Downloads dependencies from the cache or installs them if the cache is unavailable.

(**screenshot of the "install-deps" job's steps, showing the cache key calculation and cache usage (hit/miss) here.**)

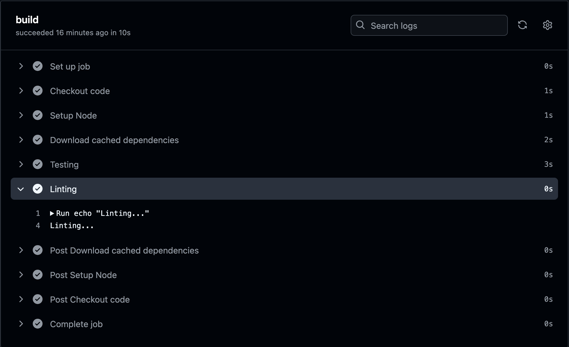
**

1. **Build Job (build)**

Responsibilities:

* + Depends on the install-deps job.
  + Reuses the cached dependencies.
  + Builds the project and runs linting.

(**A screenshot of the "build" job's steps, especially the build and linting outputs here.**)

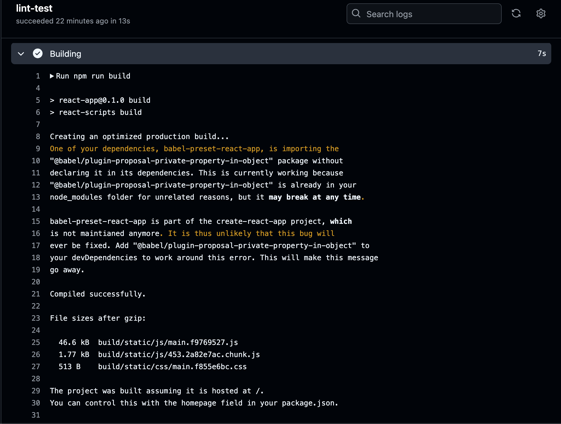


1. **Lint-Test Job (lint-test)**

Responsibilities:

* + Similar to build, depends on install-deps for caching.
  + Runs build, lint, and test commands.

( **A screenshot of the "lint-test" job's steps, highlighting the test and linting outputs here.**)



**Expected Workflow Behavior**

1. **Cache Hit:**
   * Shows "Cache restored from..." message in the install-deps job.
   * Workflow execution time is reduced.

*(****Include a screenshot of the "Cache restored" log entry here.****)*

1. **Cache Miss:**
   * Shows "Cache not found" message in the install-deps job.
   * Dependencies are freshly installed.

*(****Include a screenshot of the "Cache not found" and dependency installation log here.****)*

**Results and Benefits**

* Workflow execution time is optimized when caching is enabled.
* Demonstrates the effective use of GitHub Actions caching for dependency management.

*(****Include a screenshot of the final workflow summary showing all jobs successfully completed here.****)*