act Run Actions locally

CONCEPTS

> act reads .github/workflows, builds a plan for the event and runs each job in a Docker container. Docker must be running; with no event name it runs push.

Event	push, pull_request ...
Workflows	.github/workflows
Jobs	filtered by -j / -W
Containers	image per runs-on label

What act does with one command

RUN

act
Run the push event

act *<event>*
Run a given event (pull_request, schedule)

act -l
List workflows and jobs for the event

act -j *<job>*
Run only one job by ID

act -W *<path>*
Use a workflow file or directory

act -n
Dry run: validate without containers

act -e *<event.json>*
Supply an event payload file

act workflow_dispatch --input *<NAME>=<value>*
Set a dispatch input

act --matrix *<key>:<value>*
Limit matrix to combinations

act -w
Re-run when local files change

SECRETS AND VARIABLES

act -s *<NAME>*
Read secret from env or prompt

SECRETS AND VARIABLES (cont.)

act -s *<NAME>=<value>*
Set a secret inline

act --secret-file *<file>*
Load secrets from a .env file

act -s GITHUB_TOKEN="$(gh auth token)"
Use the gh CLI token

act --var *<NAME>=<value>*
Set a vars.* variable

act --env *<NAME>=<value>*
Set an environment variable

act --env-file *<file>*
Load env from a file

> act sets ACT in the environment. Guard steps with if: ${{ !env.ACT }}, or jobs with an event file containing {"act": true}.

IMAGES AND CONTAINERS

act -P *<label>=<image>*
Map a runs-on label to an image

act -P *<label>*=**-self-hosted**
Run a label directly on the host

act --pull=false
Use local images, skip pulling

act --container-architecture *<os/arch>*
Force architecture, e.g. linux/amd64

OUTPUT AND OFFLINE

act -v
Verbose output

act --artifact-server-path *<dir>*
Enable artifact upload/download

CONFIGURATION

.actrc
Per-project default flags