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 --input-file *<file>*
Read inputs from a .env-style file

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

act -g
Draw the workflow graph

act --validate
Validate workflows

act --detect-event
Use the workflow's first event type

act -w
Re-run when local files change

act -C *<dir>*
Set the working directory

SECRETS AND VARIABLES

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

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 --var-file *<file>*
Load variables from a file

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

act -r
Reuse containers to keep state

act -b
Bind-mount the directory, do not copy

act --rm
Remove containers after a failure

act --privileged
Run containers privileged

act --container-options *<opts>*
Extra docker options for job containers

IMAGES AND CONTAINERS (cont.)

act --network *<name>*
Set the docker network

OUTPUT AND OFFLINE

act -v
Verbose output

act --action-offline-mode
Use cached actions and images

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

CONFIGURATION

~/.config/act/actrc
User-wide default flags (XDG)

~/.actrc
Home default flags

.actrc
Per-project default flags

Default images for ubuntu-latest

Size	Image
Micro	node:16-buster-slim
Medium	catthehacker/ubuntu:act-latest
Large	catthehacker/ubuntu:full-latest

Pick the image size on first run; the medium image is a compromise that lacks many tools the hosted runner has.

RECIPES

Pull request check with a token

```
act pull_request -j test \
  -s GITHUB_TOKEN="$(gh auth
token)" \
  -e pr.json
```

Project defaults in .actrc

```
-P ubuntu-latest=catthehacker/
ubuntu:act-latest
--container-architecture=linux/
amd64
--action-offline-mode
```

One matrix cell with artifacts

```
act push --matrix node:20 \
  --artifact-server-path
$PWD/.artifacts
```

Skim jobs then run one

```
act -l
act -n -j build
act -j build --secret-
file .secrets
```