Deploying Autoscaling CI/CD Runners

Kane, Ubiquiti Inc.

No deep tech today.

Just a quick demo
for everyone.

Agenda

01

Why GitHub Actions?

02

What Do we Need?

03

Get Your Hands Dirty.

04

Keep Moving On!

Manual Deployment Era


Manual Operations


Error-Prone

Low Efficiency

2000-2010s Every Deployment is an Adventure

🔧 Automation Tools Rise

Jenkins

Travis CI

TeamCity

💻 Infrastructure Maintenance 🔧 Complex Configuration

☁ Cloud CI/CD Revolution

GitHub Actions

GitLab CI

Azure DevOps

🚀 Zero Maintenance Ready to Use

Hosted runners

Linux, macOS, Windows, ARM, GPU, and containers make it easy to build and test all your projects. Run directly on a VM or inside a container. Use your own VMs, in the cloud or on-prem, with self-hosted runners.

Matrix builds

Save time with matrix workflows that simultaneously test across multiple operating systems and versions of your runtime.

Any language

GitHub Actions supports Node.js, Python, Java, Ruby, PHP, Go, Rust, .NET, and more. Build, test, and deploy applications in your language of choice.

Live logs

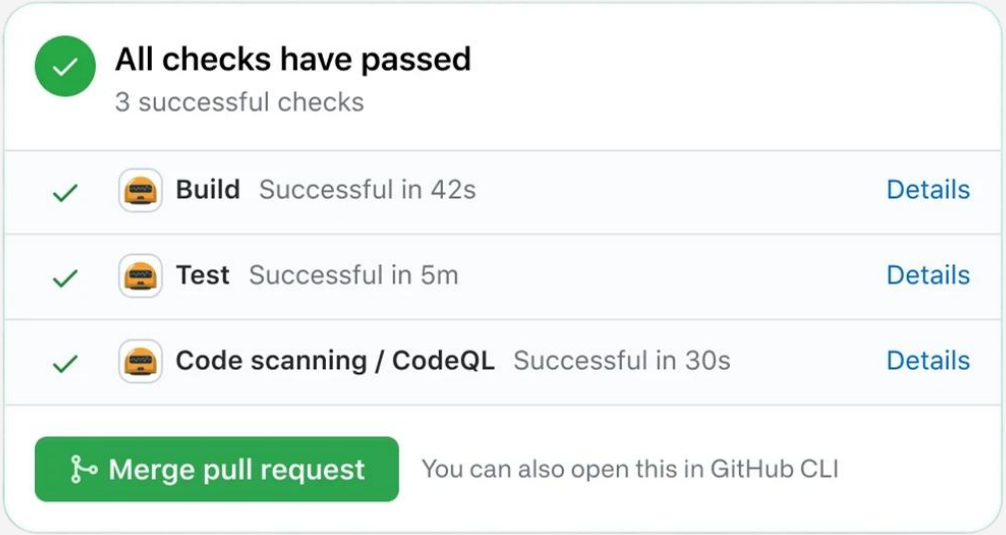
See your workflow run in realtime with color and emoji. It's one click to copy a link that highlights a specific line number to share a CI/CD failure.

Built in secret store

Automate your software development practices with workflow files embracing the Git flow by codifying it in your repository.

Multi-container testing

Test your web service and its DB in your workflow by simply adding some docker-compose to your workflow file.



Run workflow on any

event

Pipeline script

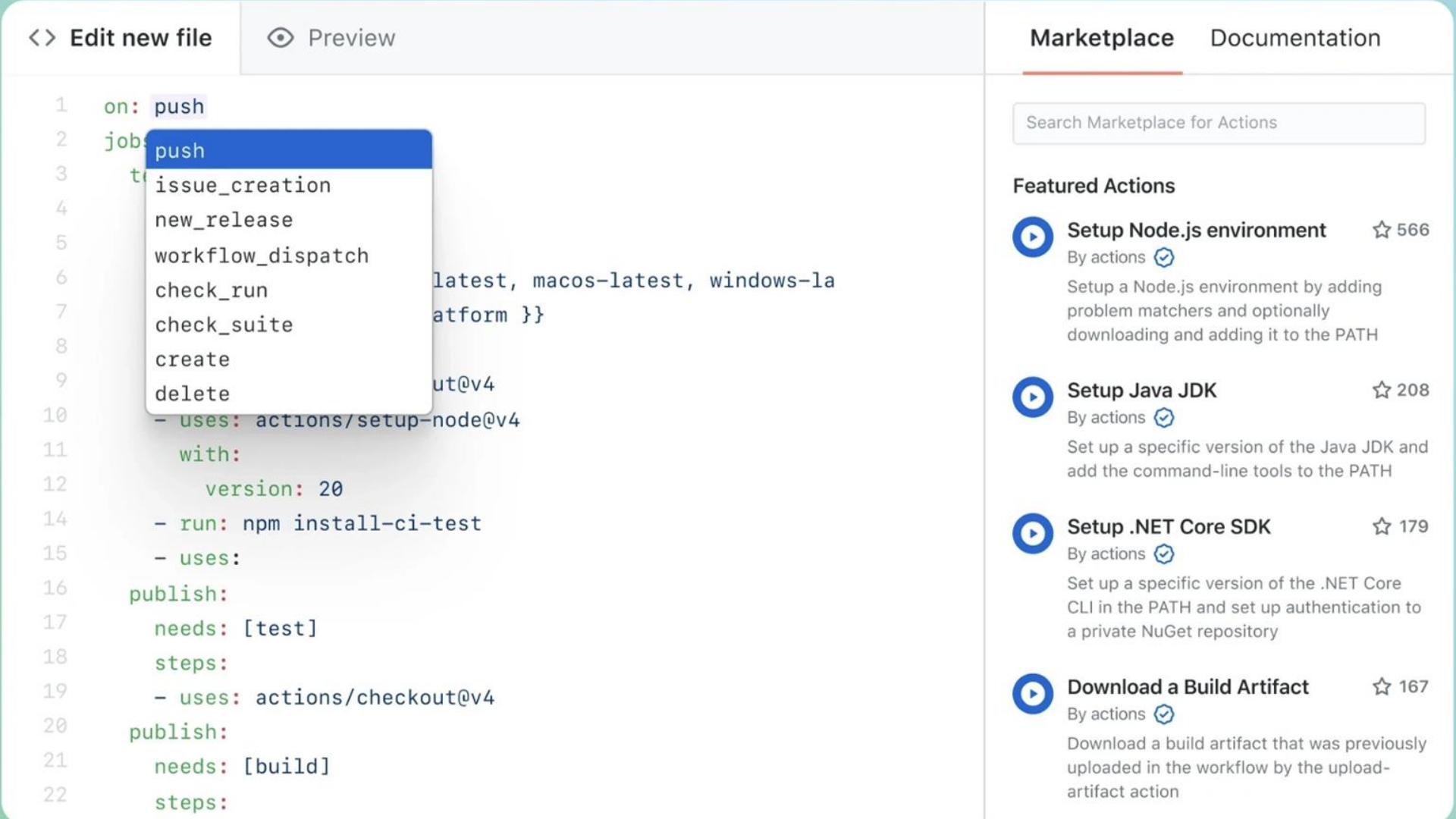
Script ?

```
node {
    def mvnHome
    stage('Preparation') { // for display purposes
        // Get some code from a GitHub repository
        git 'https://github.com/jglick/simple-maven-project-with-tests.git'
        // Get the Maven tool.
        // ** NOTE: This 'M3' Maven tool must be configured
        // **       in the global configuration.
        mvnHome = tool 'M3'
    }
    stage('Build') {
        // Run the maven build
        withEnv(["MVN_HOME=$mvnHome"]) {
            if (isUnix()) {
                sh '"$MVN_HOME/bin/mvn" -Dmaven.test.failure.ignore clean package'
            } else {
                bat(/"%MVN_HOME%\bin\mvn" -Dmaven.test.failure.ignore clean package
            }
        }
    }
    stage('Results') {
        junit '**/target/surefire-reports/TEST-*.xml'
        archiveArtifacts 'target/*.jar'
    }
}
```

Pipeline script

Script ?

```
node {
    def mvnHome
    stage('Preparation') { // for display purposes
        // Get some code from a GitHub repository
        git 'https://github.com/jglick/simple-maven-project-with-tests.git'
        // Get the Maven tool.
        // ** NOTE: This 'M3' Maven tool must be configured
        // **       in the global configuration.
        mvnHome = tool 'M3'
    }
    stage('Build') {
        // Run the maven build
        withEnv(["MVN_HOME=$mvnHome"]) {
            if (isUnix()) {
                sh '"$MVN_HOME/bin/mvn" -Dmaven.test.failure.ignore clean package'
            } else {
                bat(/"%MVN_HOME%\bin\mvn" -Dmaven.test.failure.ignore clean package
            }
        }
    }
    stage('Results') {
        junit '**/target/surefire-reports/TEST-*.xml'
        archiveArtifacts 'target/*.jar'
    }
}
```

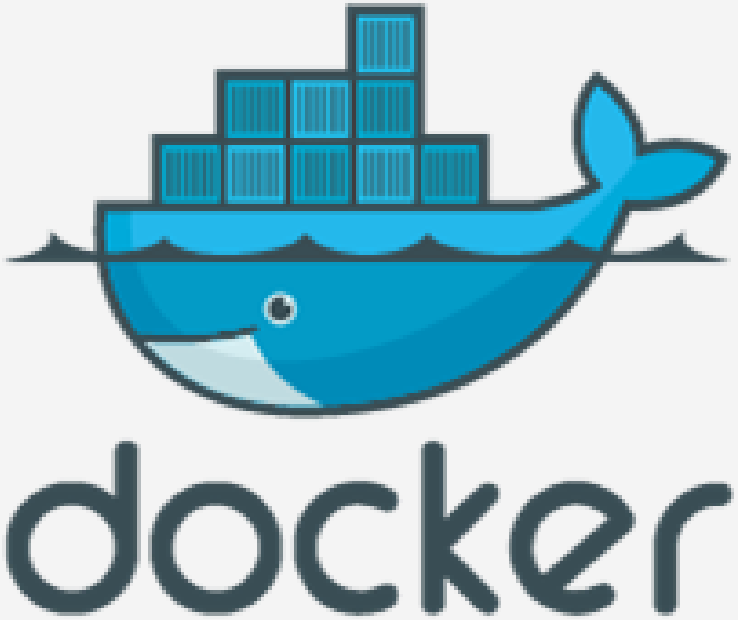

Edit new file
Preview
Marketplace
Documentation
on: push
jobs
push
issue_creation
new_release
workflow_dispatch
check_run
check_suite
create
delete
latest, macos-latest, windows-la
atform }}
ut@v4
- uses: actions/setup-node@v4
with:
version: 20
- run: npm install-ci-test
- uses:
publish:
needs: [test]
steps:
- uses: actions/checkout@v4
publish:
needs: [build]
steps:
Search Marketplace for Actions
Featured Actions
Setup Node.js environment
566
By actions
Setup a Node.js environment by adding problem matchers and optionally downloading and adding it to the PATH
Setup Java JDK
208
By actions
Set up a specific version of the Java JDK and add the command-line tools to the PATH
Setup .NET Core SDK
179
By actions
Set up a specific version of the .NET Core CLI in the PATH and set up authentication to a private NuGet repository
Download a Build Artifact
167
By actions
Download a build artifact that was previously uploaded in the workflow by the upload-artifact action

What Do We Needs?

1. GitHub Account
2. Docker Engine
3. Minikube
4. Helm
5. K9s

Docker Engine

```
$ docker -v
```



https://docs.docker.com/engine/install/

Minikube

$ minikube version

https://minikube.sigs.k8s.io/docs/start/?arch=%2Fmacos%2Farm64%2Fstable%2Fbinary+download

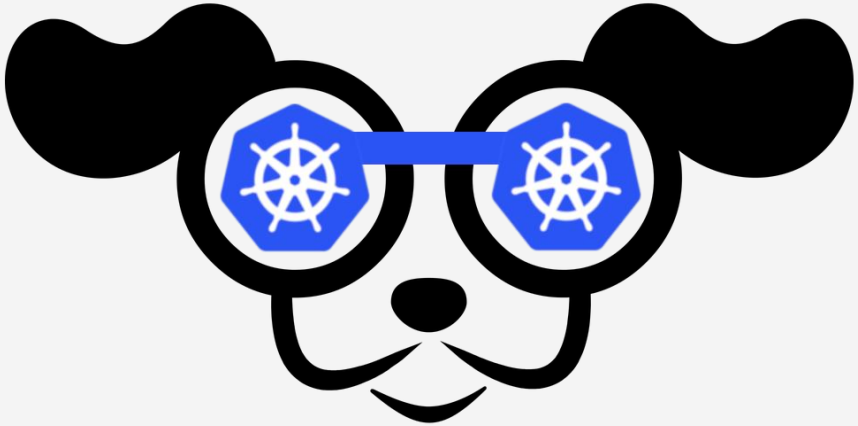
Helm

```
$ helm version
```

https://helm.sh/docs/intro/install/

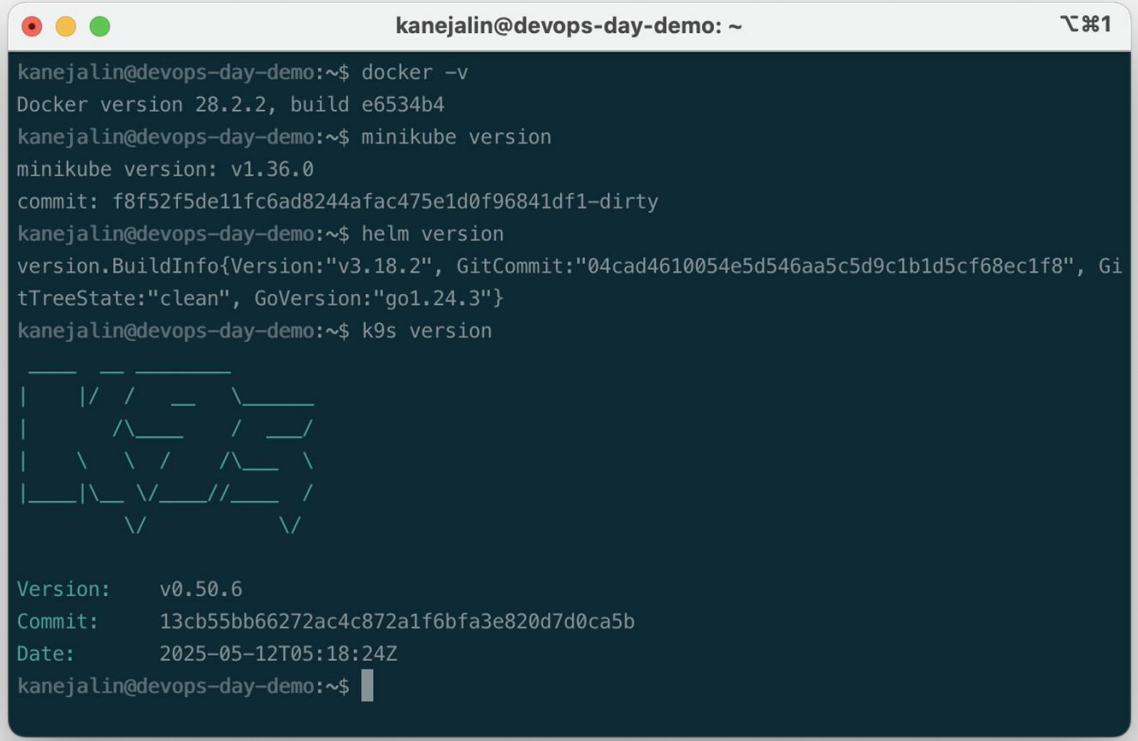
K9s

$ k9s version

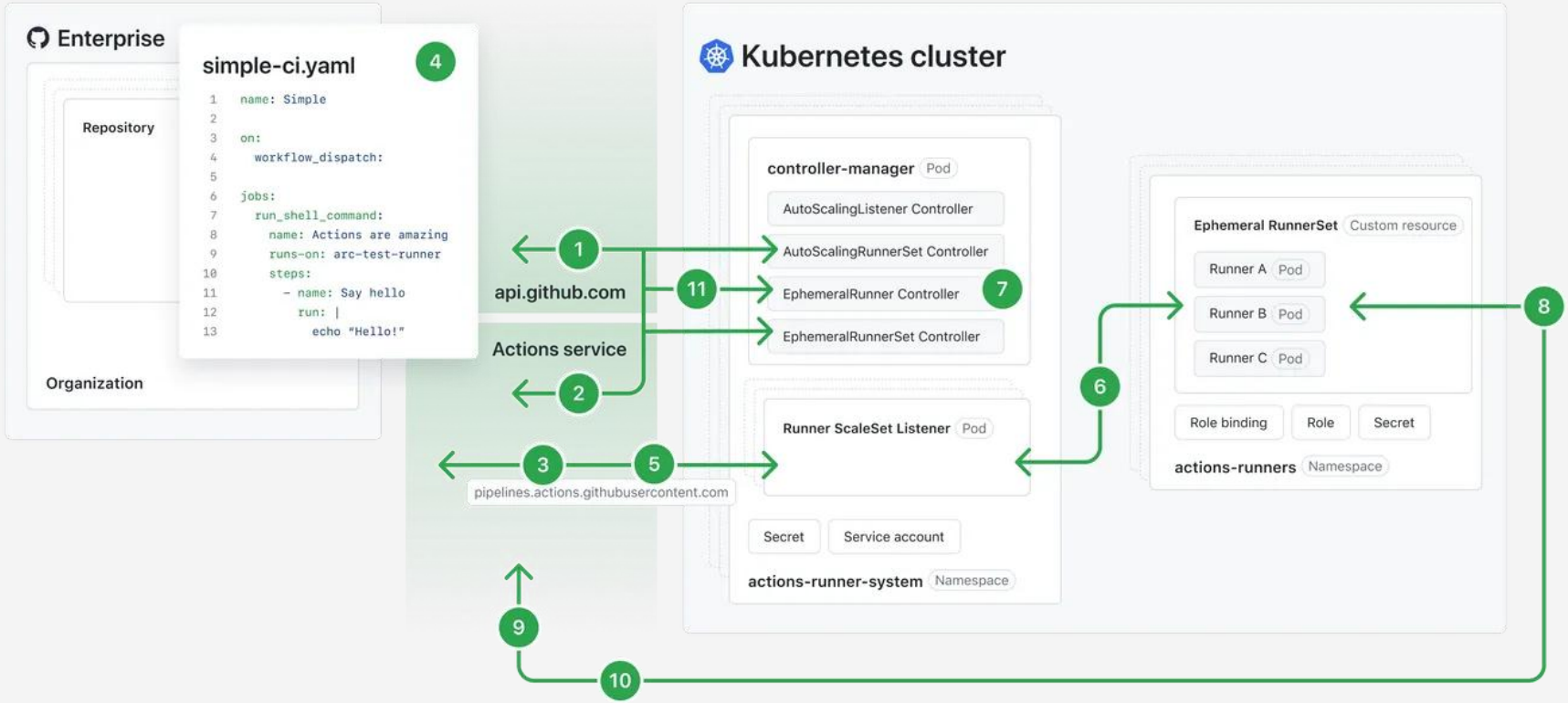


https://k9scli.io/topics/install/

10 Mins for Installation



Actions Runner Controller



ARC Setup Steps

1. Prepare GitHub API Key
2. Start K8s by minikube
3. Installing Actions Runner Controller
4. Clone Demo Repository
5. Configuring a runner scale set

Prepare GitHub API Key

Create a personal access token (classic) with the required scopes. The required scopes are different depending on whether you are registering runners at the repository or organization level. For more information on how to create a personal access token (classic), see Managing your personal access tokens.

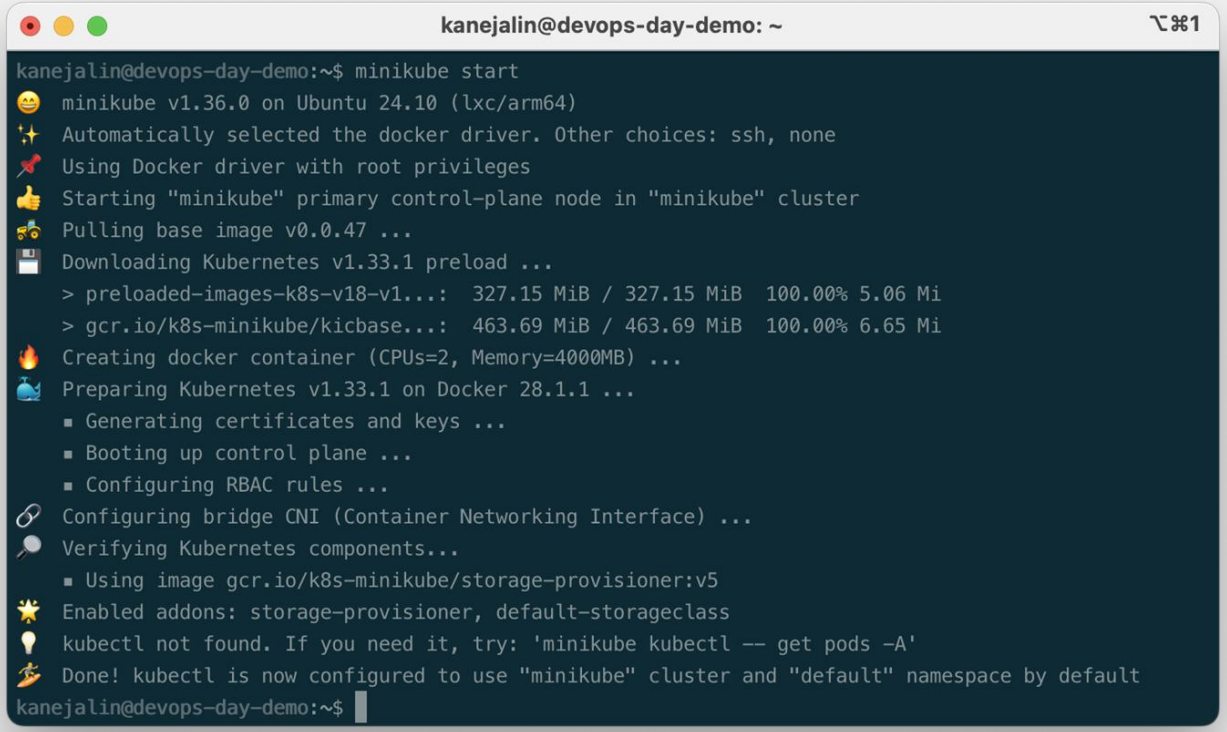
The following is the list of required personal access token scopes for ARC runners.

- Repository runners: `repo`
- Organization runners: `admin:org`

Start K8s by minikube

```
$ minikube start
```

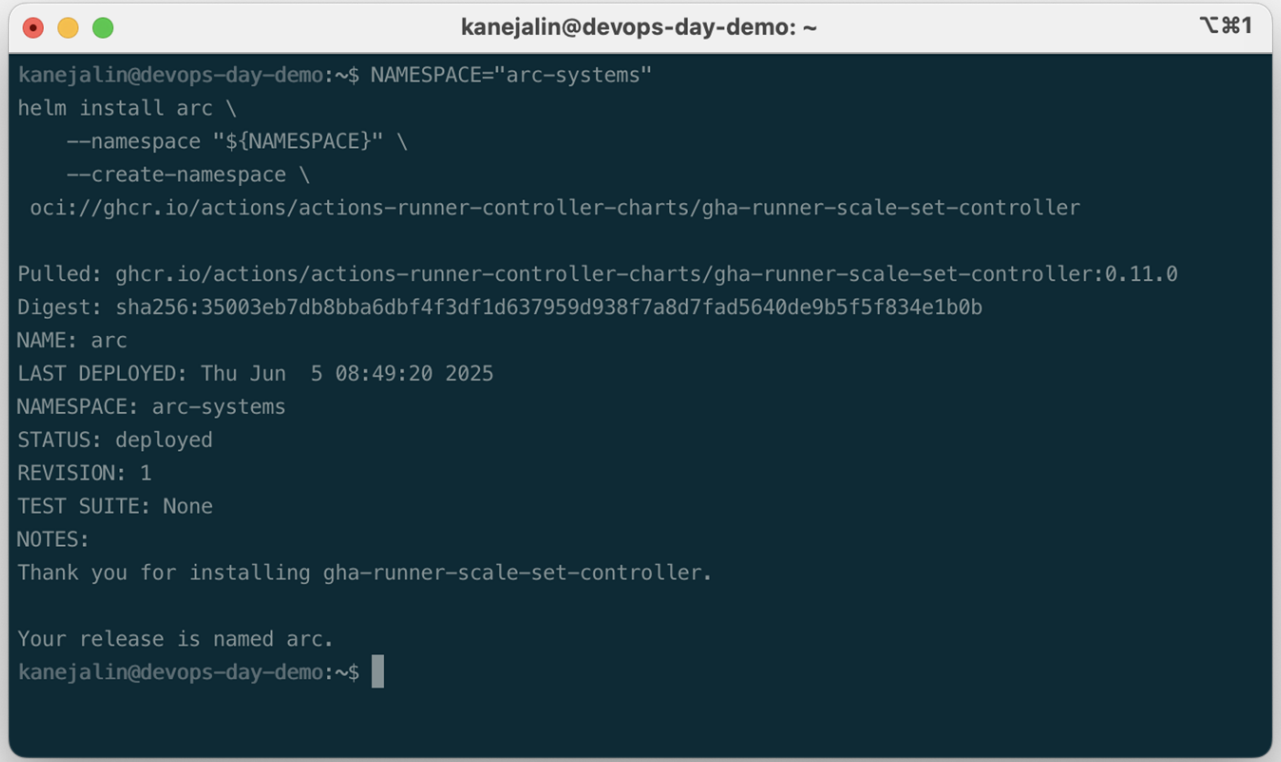
3 Mins for minikube



Installing ARC

```
NAMESPACE="arc-systems"
helm install arc \
    --namespace "${NAMESPACE}" \
    --create-namespace \
 oci://ghcr.io/actions/actions-runner-controller-charts/gha-runner-scale-set-controller
```

3 Mins for ARC

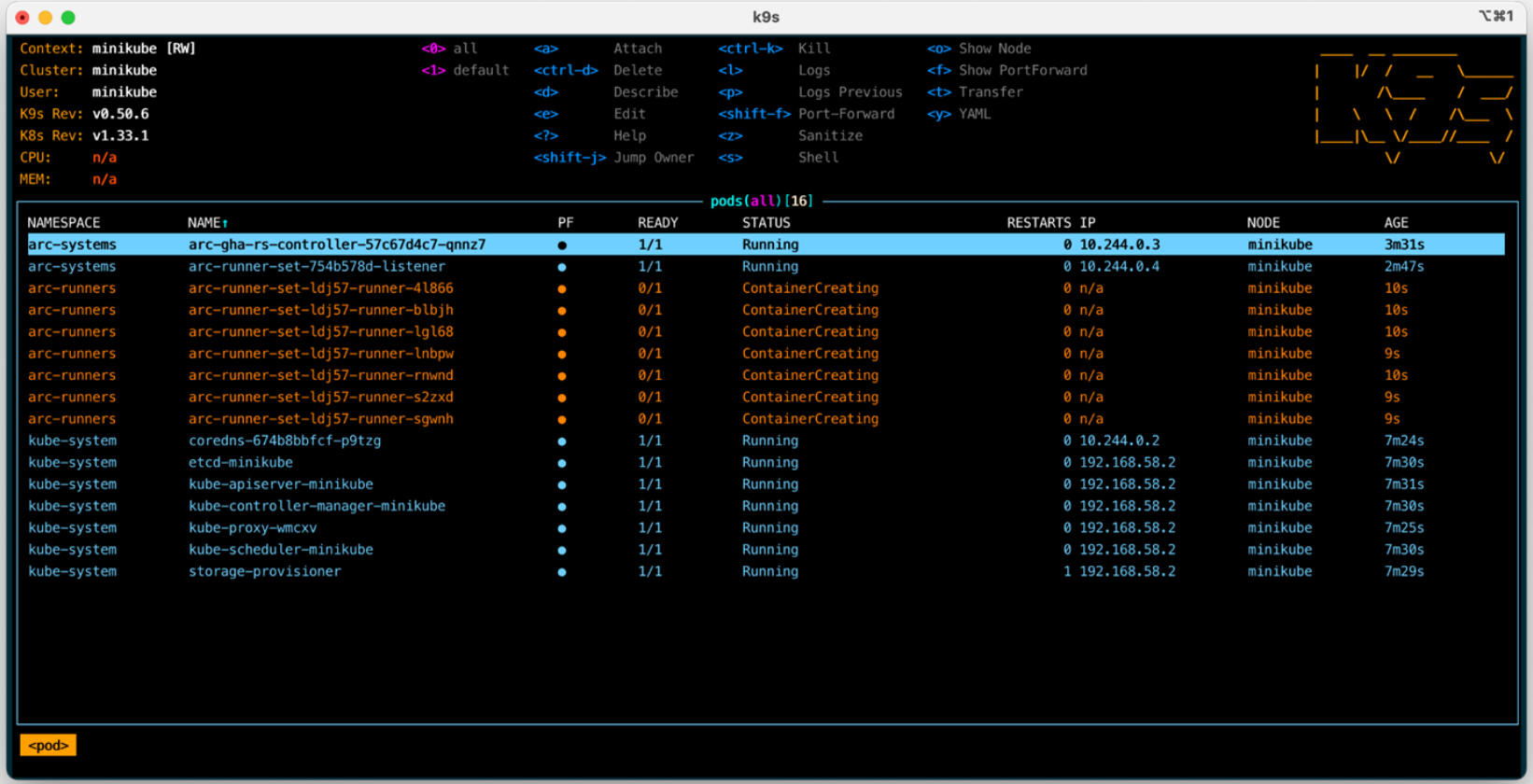


Clone Demo Repository

Clone this repository

Configuring a runner scale set

```
INSTALLATION_NAME="arc-runner-set"
NAMESPACE="arc-runners"
GITHUB_CONFIG_URL="https://github.com/<your_enterprise/org/repo>"
GITHUB_PAT="<PAT>"
helm install "${INSTALLATION_NAME}" \
    --namespace "${NAMESPACE}" \
    --create-namespace \
    --set githubConfigUrl="${GITHUB_CONFIG_URL}" \
    --set githubConfigSecret.github_token="${GITHUB_PAT}" \
    oci://ghcr.io/actions/actions-runner-controller-charts/gha-runner-scale-set
```


k9s
⌥⌘1
Context: minikube [RW]
Cluster: minikube
User: minikube
K9s Rev: v0.50.6
K8s Rev: v1.33.1
CPU: n/a
MEM: n/a
<0> all
<1> default
<a> Attach
<ctrl-d> Delete
<d> Describe
<e> Edit
<?> Help
<shift-j> Jump Owner
<ctrl-k> Kill
<l> Logs
<p> Logs Previous
<shift-f> Port-Forward
<z> Sanitize
<s> Shell
<o> Show Node
<f> Show PortForward
<t> Transfer
<y> YAML
pods(all)[16]
NAMESPACE NAME↑ PF READY STATUS RESTARTS IP NODE AGE
arc-systems arc-gha-rs-controller-57c67d4c7-qnnz7 1/1 Running 0 10.244.0.3 minikube 3m31s
arc-systems arc-runner-set-754b578d-listener 1/1 Running 0 10.244.0.4 minikube 2m47s
arc-runners arc-runner-set-ldj57-runner-4l866 0/1 ContainerCreating 0 n/a minikube 10s
arc-runners arc-runner-set-ldj57-runner-blbjh 0/1 ContainerCreating 0 n/a minikube 10s
arc-runners arc-runner-set-ldj57-runner-lgl68 0/1 ContainerCreating 0 n/a minikube 10s
arc-runners arc-runner-set-ldj57-runner-lnbpw 0/1 ContainerCreating 0 n/a minikube 9s
arc-runners arc-runner-set-ldj57-runner-rnwnd 0/1 ContainerCreating 0 n/a minikube 10s
arc-runners arc-runner-set-ldj57-runner-s2zxd 0/1 ContainerCreating 0 n/a minikube 9s
arc-runners arc-runner-set-ldj57-runner-sgwnh 0/1 ContainerCreating 0 n/a minikube 9s
kube-system coredns-674b8bbfcf-p9tzg 1/1 Running 0 10.244.0.2 minikube 7m24s
kube-system etcd-minikube 1/1 Running 0 192.168.58.2 minikube 7m30s
kube-system kube-apiserver-minikube 1/1 Running 0 192.168.58.2 minikube 7m31s
kube-system kube-controller-manager-minikube 1/1 Running 0 192.168.58.2 minikube 7m30s
kube-system kube-proxy-wmcxv 1/1 Running 0 192.168.58.2 minikube 7m25s
kube-system kube-scheduler-minikube 1/1 Running 0 192.168.58.2 minikube 7m30s
kube-system storage-provisioner 1/1 Running 1 192.168.58.2 minikube 7m29s
<pod>

Monitoring


Runner Namespace
arc-runners
ARC System Namespace
arc-systems
Startup Duration
Wait Time
50 min
15 min
7 min
3.50 min
1.83 min
1.17 min
30 s
15 s
8 s
4 s
500 ms
0 s
02:05
02:10
1.50
19.5
Job Execution
Time
1.15
20.6
Workqueue depth
13
Average Execution Time
arc-test/main.yml@main
24.5 s
arc-test/main.yml@test
22 s
arc-test/main.yml@test2
24 s
Active Runners
8
Idle Runners
0
Pending Runners
11
Desired Runners
20
Container I/O
Bytes
200 kB
0 B
Write
Read
Listeners
1
Registered Runners
15
Assigned Jobs
26
Running Jobs
8
Container Pod Status
Ready
Waiting
Completed
Error
9
6
5
0

Awesome Actions

Awesome Actions  

> A curated list of awesome things related to GitHub Actions.

Actions are triggered by GitHub platform events directly in a repo and run on-demand workflows either on Linux, Windows or macOS virtual machines or inside a container in response. With GitHub Actions you can automate your workflow from idea to production.

WE WANT YOU!

Positions

1. F0128-Software Development Engineer in Test (Switch)
2. F0032 - Software Test Engineer

Thank you

Questions? Contact

www.linkedin.com/in/jiananlin1104