Kubernetes Community Days
MUNICH 2024

OUR SPONSORS

metalstack cloud
CISCO
consol Enterprising IT
APE FACTORY
paloalto NETWORKS
VEEAM
dynatrace
Pulumi
StackState
camp to camp
splunk> a cisco company
mindcurv Part of Accenture Song
spectro cloud
Akamai
ISOVALENT now part of cisco
EXOSCALE
TIGERA
STEADFORCE
ADN
CSP²
DGi
REPLY LIQUID
white duck
QA|WARE SOFTWARE ENGINEERING

Speaker: Koray Oksay Batuhan Apaydın
Company: Kubermatic Trendyol

No More YAML Soup: Taking Control with Dagger's Pipeline-as-Code Philosophy

No More YAML Soup: Taking Control with Dagger's Pipeline-as-Code Philosophy

Batuhan Apaydın

Platform Engineer at trendyol and Software Supply Chain Security Enthusiastic 🧐

Best Sigstore Evangelist Award 2022 🏆

Kubestronaut 🏆

CNCF Ambassador

Docker Captain

KCD Istanbul Organizer 🚀

linkedin.com/in/bthnapydin

@developerguyba

developer-guy

Kubernetes Community Days
MUNICH 2024

Koray Oksay

Kubernetes Consultant and Instructor @Kubermatic working remotely from Istanbul

CNCF Ambassador

Instructor for Linux Foundation Kubernetes trainings

#sig-k8s-infra

KCD Istanbul Organizer

linkedin.com/in/korayoksay

@korayoksay

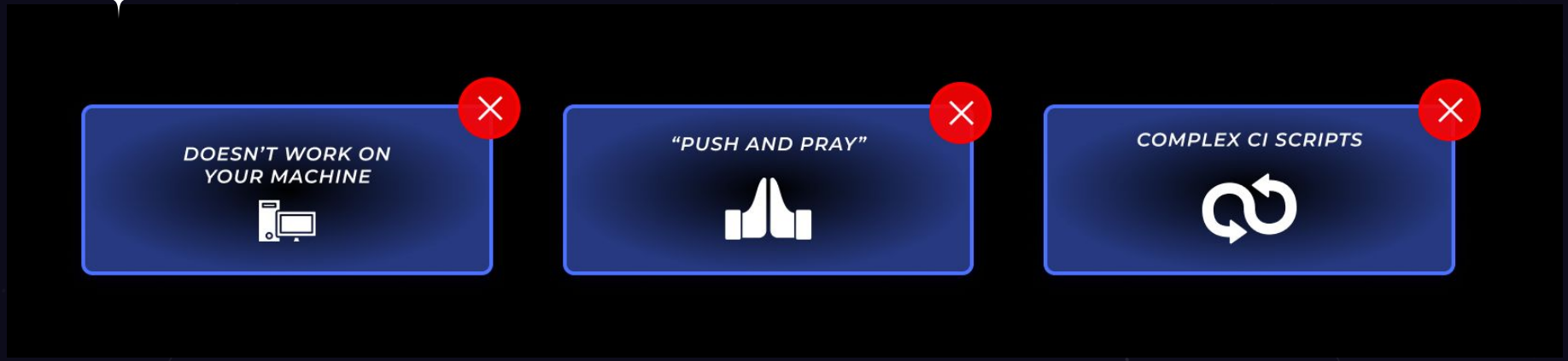
koksay

Kubernetes Community Days
MUNICH 2024


ONE DOES NOT SIMPLY HAVE GREEN PIPELINES
ON THE 1ST ATTEMPT
imgflip.com


Kubernetes Community Days
MUNICH 2024

Current CI/CD Problems



Solomon Hykes
@solomonstre

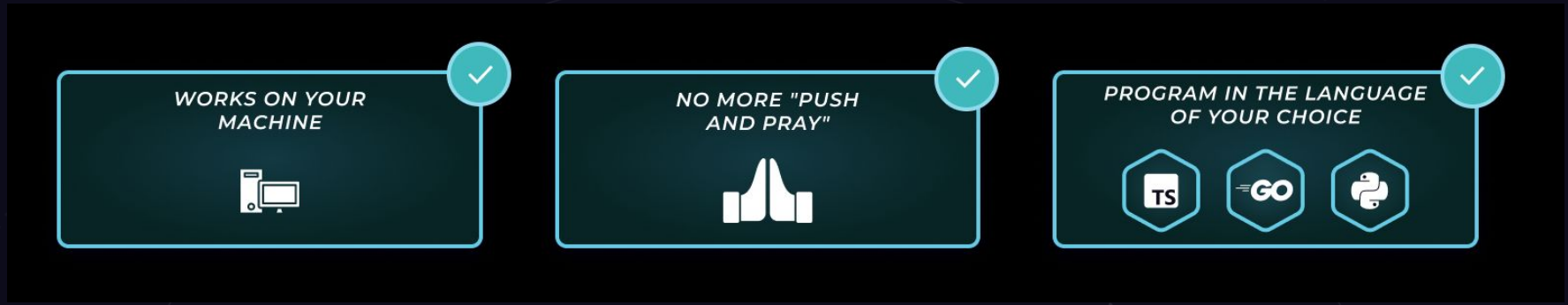
Your CI/CD pipeline should start on the developer's laptop. If it only starts after a git push, you're slowing your team down and throwing money down the drain.

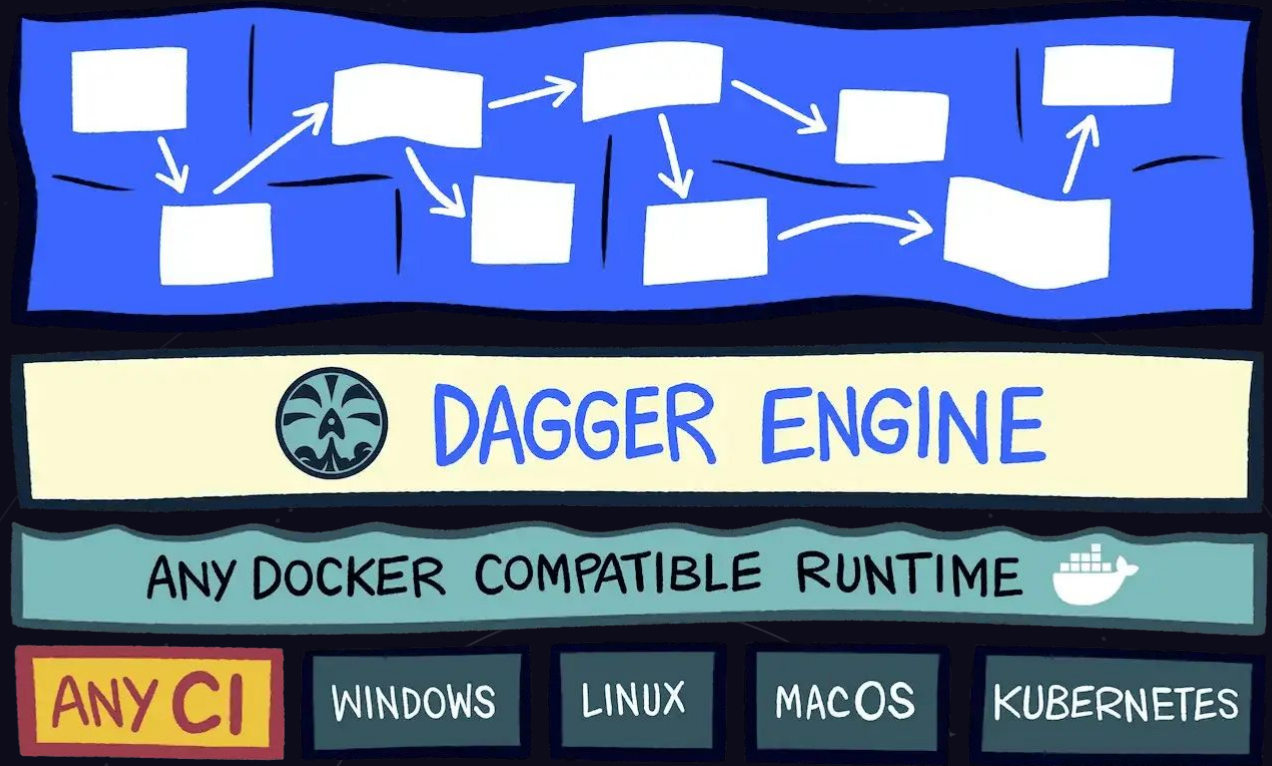
CI/CD "shift left" is the lowest-hanging fruit for engineering efficiency in teams of 20+ engineers IMO.

9:29 PM · Apr 20, 2023 · **388.9K** Views

83 225 1.3K 486

CI/CD Solutions with Dagger





Kubernetes Community Days
MUNICH 2024

The Dagger Platform

Integrations (https://dagger.io/integrations)

GitLabCI

Documentation: Use less YAML and gain more control by calling Dagger functions in GitLab CI/CD

Tekton

Documentation: Using Dagger with Tekton CI

Argo Workflows

Documentation: Code sample demonstrates how to integrate Dagger with Argo Workflows

Azure Pipelines

Documentation: Unlock local execution and the power of TypeScript, Python, or Go

Kubernetes

Daggerverse modules: Dagger Functions that range from "dev" to "deploy" on Kubernetes

GitHub Actions

Documentation: Call Dagger Functions from Actions workflows using the dagger-for-github action

Helm

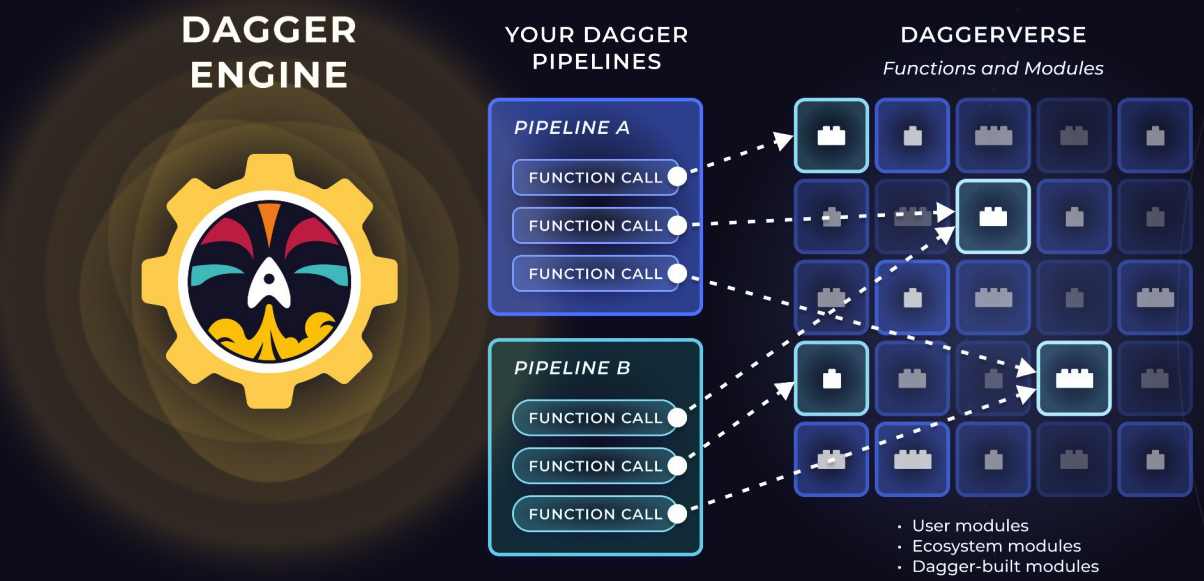
Daggerverse Module: Use Helm Charts help define, install, and upgrade complex Kubernetes applications

Amazon EKS

Daggerverse module: Leverage managed Kubernetes in AWS to run your Dagger Pipelines

Google Cloud

Daggerverse modules: Running Dagger with GCP and GoogleCloud Run

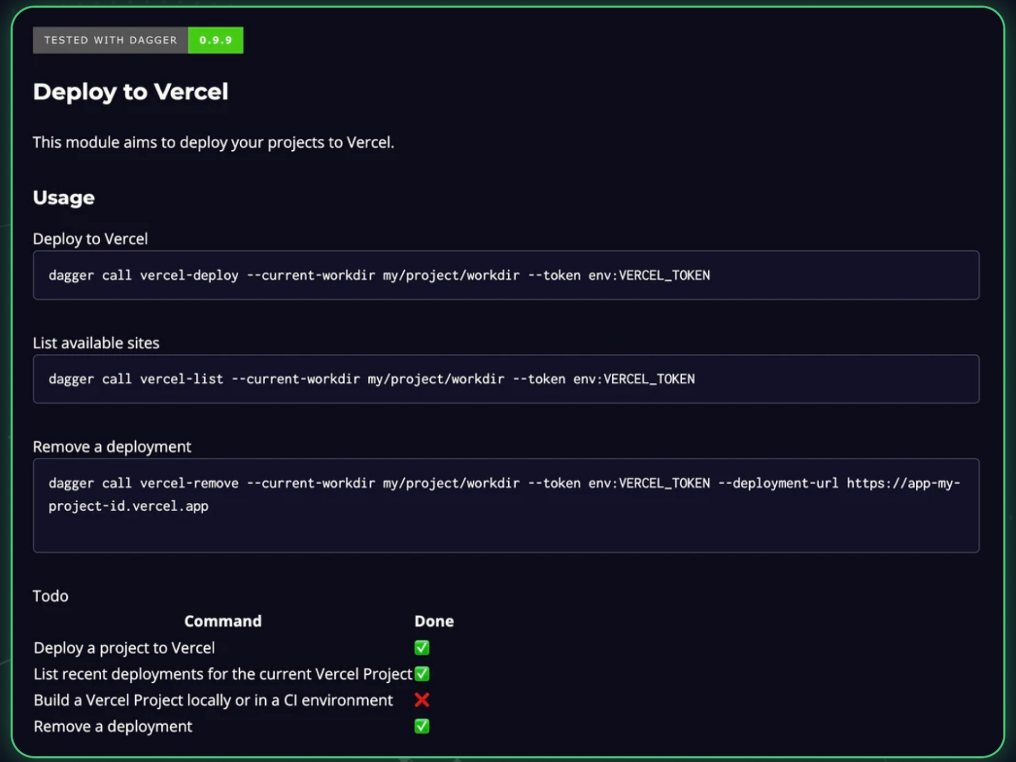


No more YAML soup

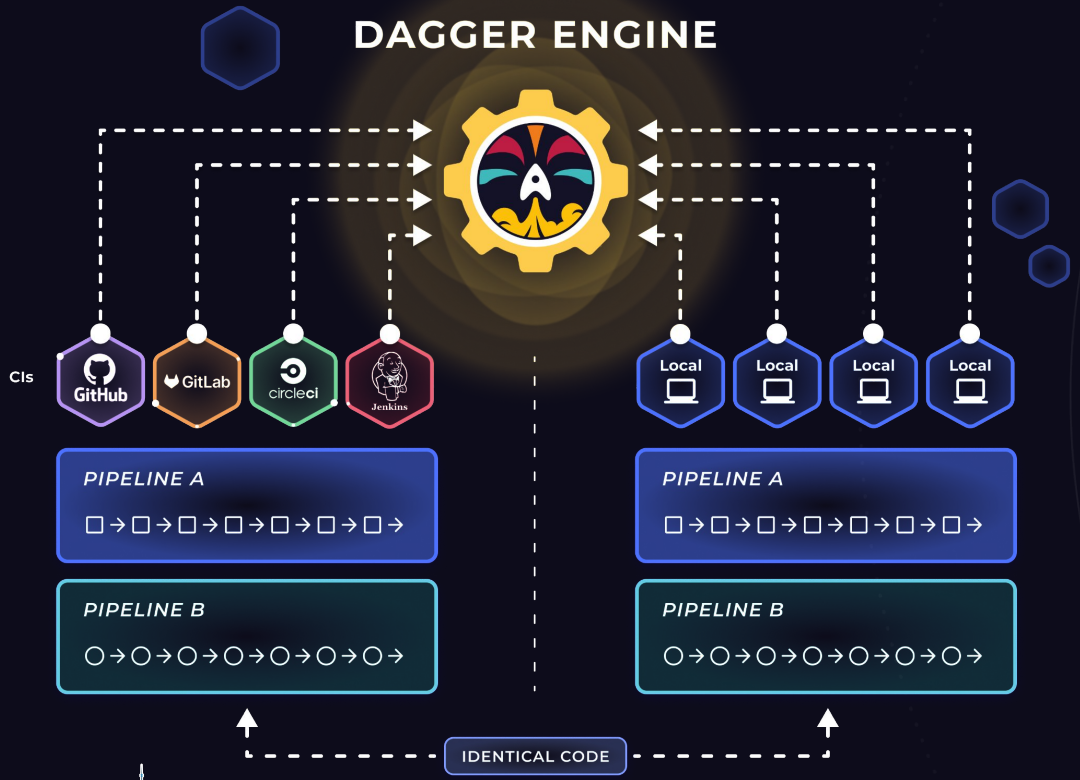
Replace complex CI scripts with a programmable platform

Standardized Dagger Functions

Pipelines just chain Dagger Functions - built by your team or by the community



Kubernetes Community Days
MUNICH 2024



Eliminate Push And Pray

If it works on your laptop
it'll work in CI

Cached
For Speed

Avoid unnecessary rebuilds and test reruns when nothing has changed

```
func (g *Golang) Base(version string) *Golang {
    mod := dag.CacheVolume("gomodcache")
    build := dag.CacheVolume("gobuildcache")
    image := fmt.Sprintf("golang:%s", version)
    c := dag.Container().
        From(image).
        WithMountedCache("/go/pkg/mod", mod).
        WithMountedCache("/root/.cache/go-build", build)
    g.Ctr = c
    return g
}
```

```typescript
import { dag, Container, Directory, object, func } from "@dagger.io/dagger"

@object()
// eslint-disable-next-line @typescript-eslint/no-unused-vars
class Ci {

  /**
   * example usage: "dagger call ci --source ."
   */
  @func()
  async ci(source: Directory): Promise<string> {
    // Use Golang module to configure project
    var goProject = dag.golang().withProject(source)

    // Run Go tests using Golang module
    await goProject.test()

    // Get container with built binaries using Golang module
    var image = await goProject.buildContainer()

    // Push image to a registry using core Dagger API
    var ref = await image.publish("ttl.sh/demoapp:1h")

    // Scan image for vulnerabilities using Trivy module
    return dag.trivy().scanContainer(dag.container().from(ref))
  }
}
```

Multi-Language

Pipelines in the same language as your app. Each Dagger Function is just an API call away.

Demo Time

https://github.com/deveoper-guy/kcd-munich-2024-demo

Kubernetes Community Days
MUNICH 2024

Thank you!

Questions?