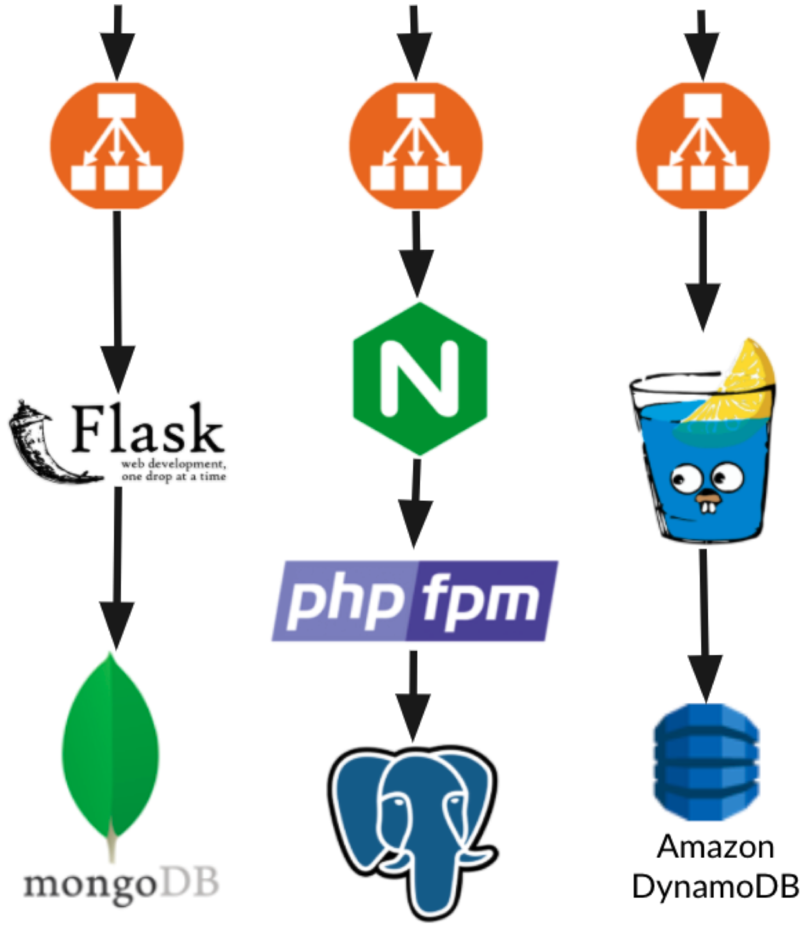
Why You Should Standardize Workflows, Not Tools

Mazedur Rahman
Lead Software Engineer
Airbus UTM

DEVOPS WORLD by CloudBees

Engineering Teams' Dreams

- One Programming Language
- One MVC Framework
- One REST API Framework
- One Database Technology
- One Caching Technology
- One CI
- One Deployment Tool
- ...



Engineering Teams' Regrets

- We wrote everything using a single programming language
- We allowed engineers to use only one specific MVC/REST API Framework
- We use a specific DBMS/Cache
- We can't iterate/test pipeline logic outside our "controlled" CI/CD environment
- We are tightly coupled with a specialized deployment tooling
- ...

Engineering Teams' Mistakes

- We can't iterate/test pipeline logic outside our "controlled" CI/CD environment
  - We built a **Unicorn**
    - We can't develop outside the controlled environment
    - We can't debug outside the controlled environment

Engineering Teams' Mistakes: Programming in CI/CD Tool

```
    stage("Determine new version") {
         steps {
             script {
                 env.DEPLOY_VERSION = sh(returnStdout: true, script: "docker run --rm -v '${env.WORKSPACE}':/repo:ro
softonic/ci-version:0.1.0 --compatible-with package.json").trim()
                 env.DEPLOY_MAJOR_VERSION = sh(returnStdout: true, script: "echo '${env.DEPLOY_VERSION}' | awk -F'[ .]'
'{print \$1}'").trim()
                 env.DEPLOY_COMMIT_HASH = sh(returnStdout: true, script: "git rev-parse HEAD | cut -c1-7").trim()
                 env.DEPLOY_BUILD_DATE = sh(returnStdout: true, script: "date -u +'%Y-%m-%dT%H:%M:%SZ'").trim()
                 env.DEPLOY_STACK_NAME = "${env.STACK_PREFIX}-v${env.DEPLOY_MAJOR_VERSION}"
                 env.IS_NEW_VERSION = sh(returnStdout: true, script: "[ '${env.DEPLOY_VERSION}' ] && echo 'YES'").trim()
             }
         }
    }
```

DEVOPSWORLD by CloudBees

Engineering Teams’ Mistakes: Coupling with Deployment Tool

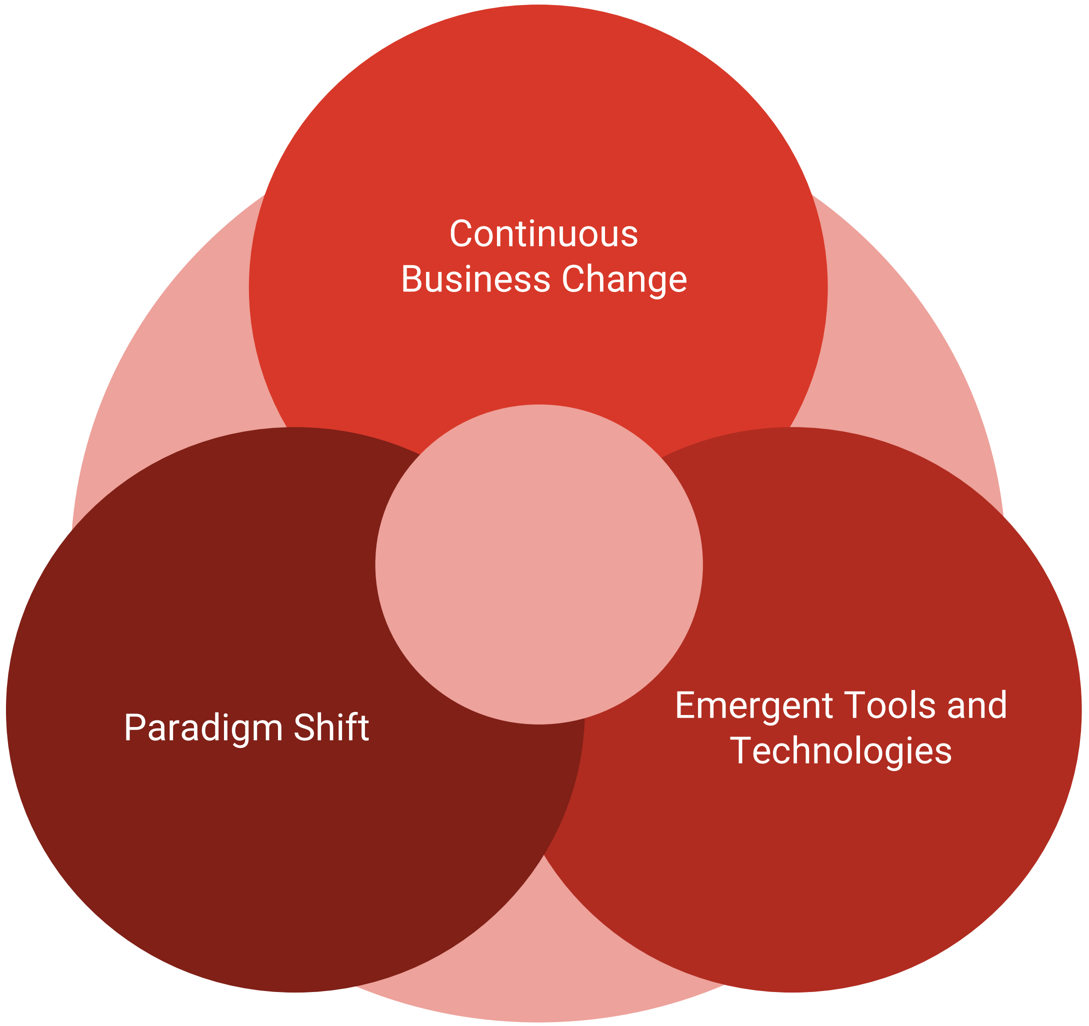
- All our provisioning and deployment done using one tool
  - We can’t extend it for new deployment needs, we didn’t foresee the new future
    - Puppet/Chef to orchestrate containers?

puppet

CHEF™

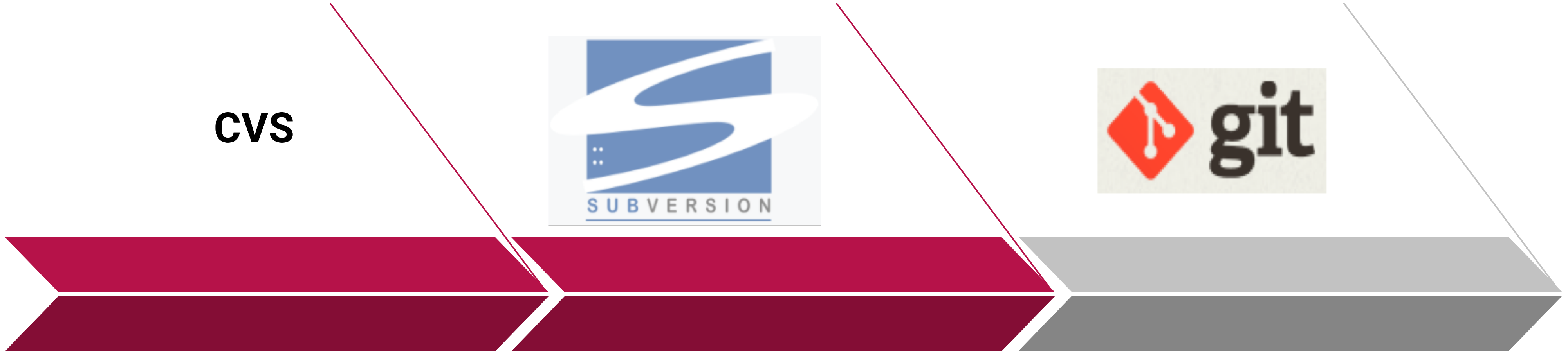
☹

Unpleasant Reality

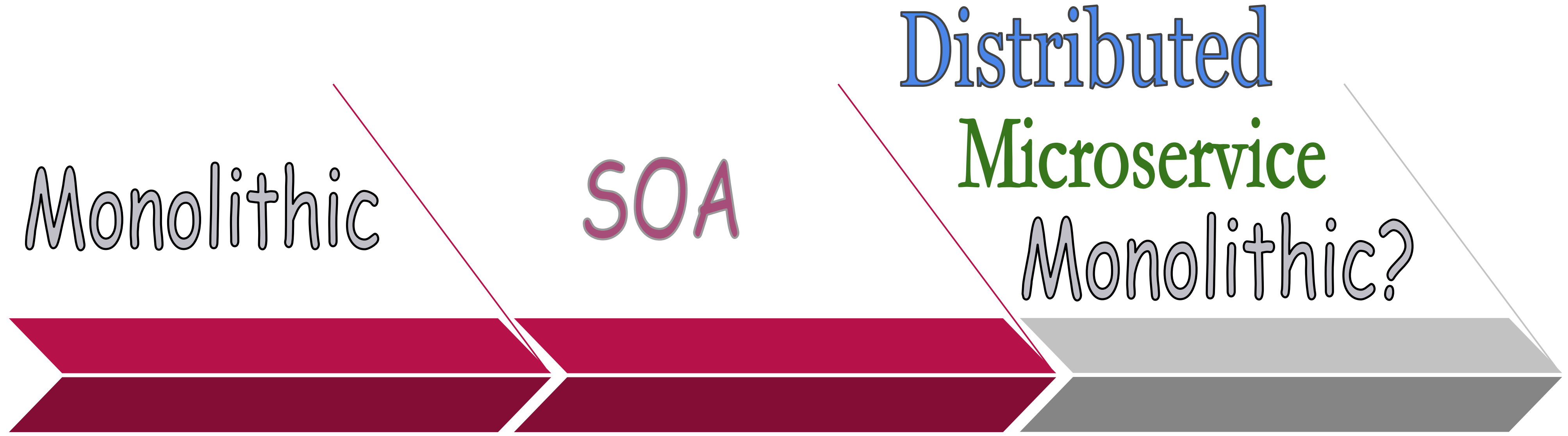


DEVOPSWORLD by CloudBees

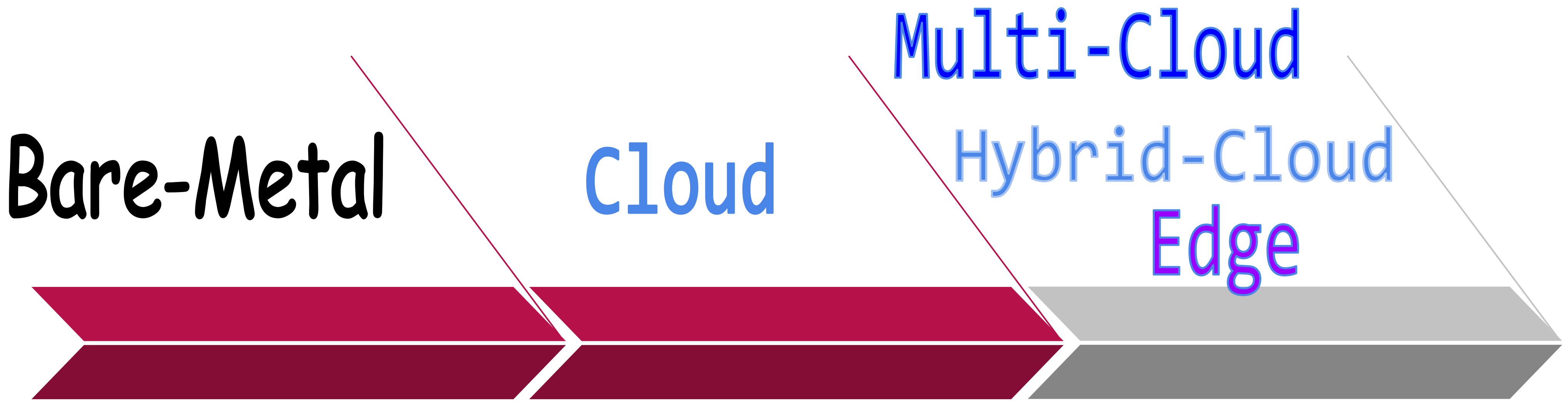
Unpleasant Reality: Version Control



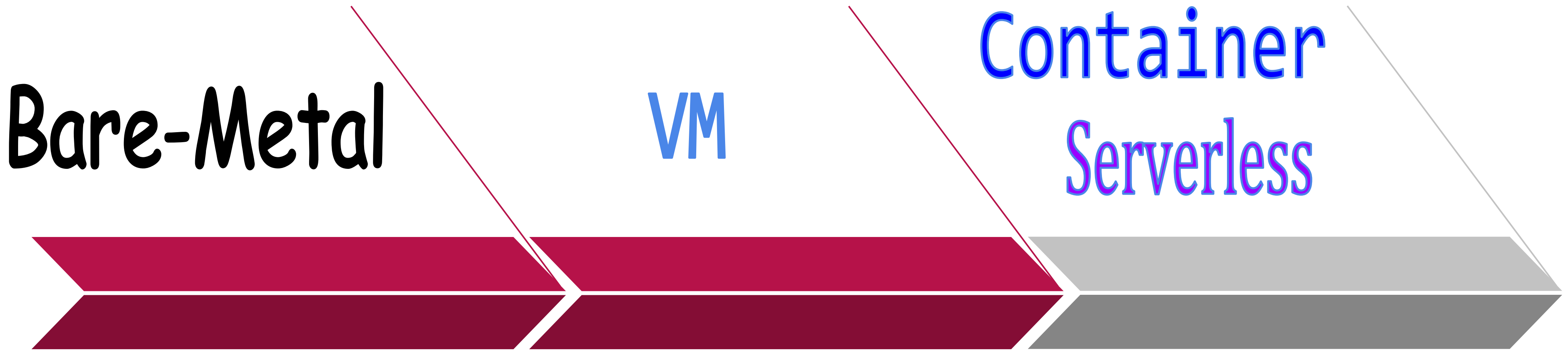
Unpleasant Reality: Architecture



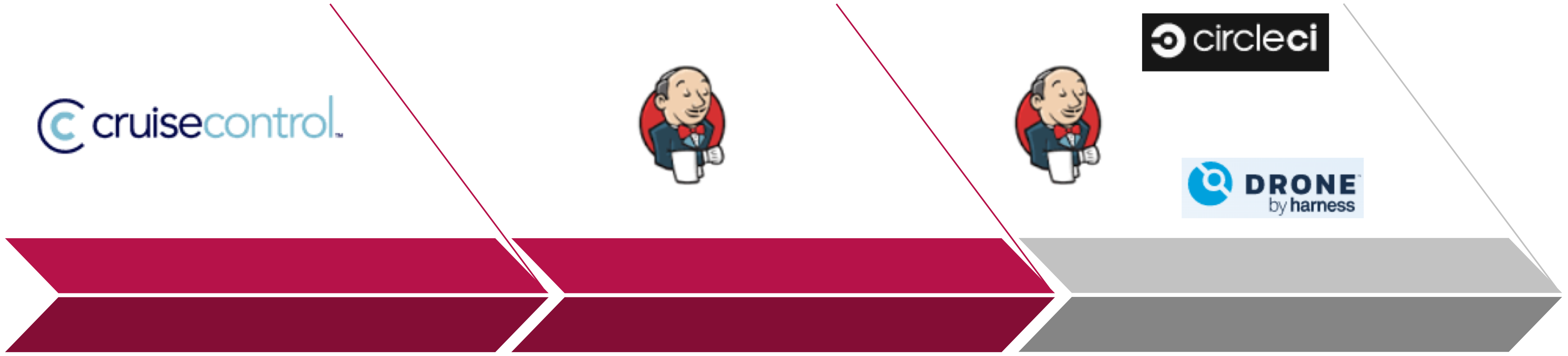
Unpleasant Reality: Infrastructure



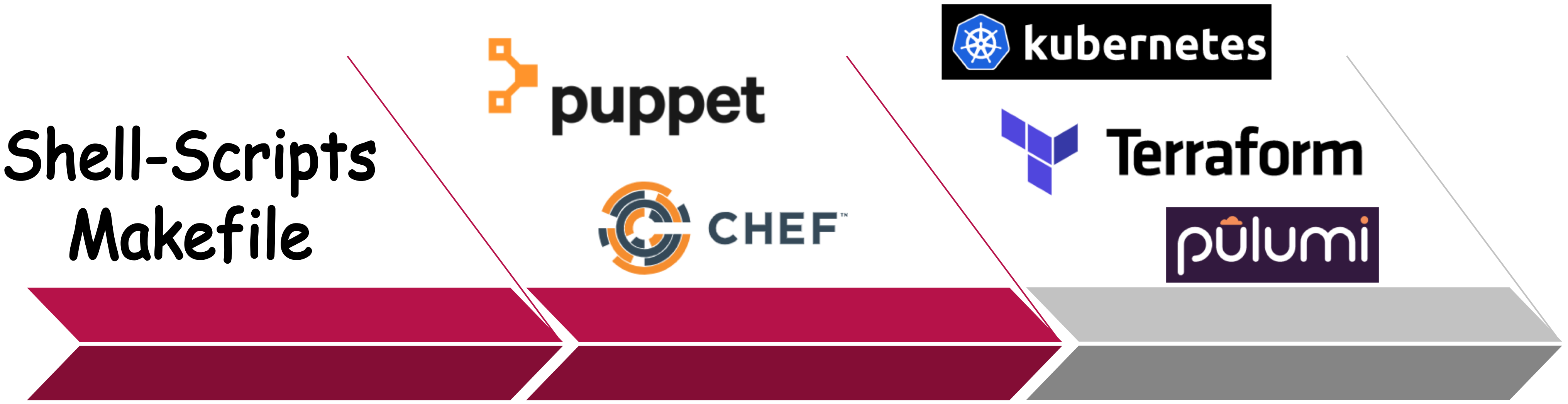
Unpleasant Reality: Workload Runtime



Unpleasant Reality: CI/CD



Unpleasant Reality: Provisioning and Config Management



Our World at a Glance: CNCF Hosted Tools - Graduated

Kubernetes
Orchestration

Prometheus
Monitoring

Envoy
Network Proxy

CoreDNS
Service Discovery

containerd
Container Runtime

Fluentd
Logging

Jaeger
Distributed Tracing

Vitess
Storage

TUF
Software Update Spec

Helm
Package Management

Harbor
Registry

Source: cncf.io

Our World at a Glance: CNCF Hosted Tools - Incubating

OpenTracing Distributed Tracing API	gRPC Remote Procedure Call	CNI Networking API	Notary Security	NATS Messaging	Linkerd Service Mesh
Rook Storage	etcd Key/Value Store	Open Policy Agent	CRI-O Container Runtime	TiKV Key/Value Store	CloudEvents Serverless
Falco Container Security	Argo Continuous Integration & Deployment	Dragonfly Image Distribution	SPIFFE Identity Spec	SPIRE Identity	Contour High performance ingress controller

Our World at a Glance: CNCF Hosted Tools - Sandbox

Telepresence
Tooling

OpenMetrics
Metrics Spec

Cortex
Monitoring

Buildpacks
Packaging Spec

Virtual Kubelet
Nodeless

KubeEdge
Edge

Brigade
Scripting

Network Service Mesh

OpenTelemetry
Telemetry Specification

OpenEBS
Storage

Thanos
Monitoring

Flux
GitOps

in-toto
Security

Strimzi
Kafka Operator

KubeVirt
VM Operator

Longhorn
Storage

ChubaoFS
Storage

KEDA
Event-driven autoscaling

DEVOPSWORLD by CloudBees

Our World at a Glance: CNCF Hosted Tools - Archived

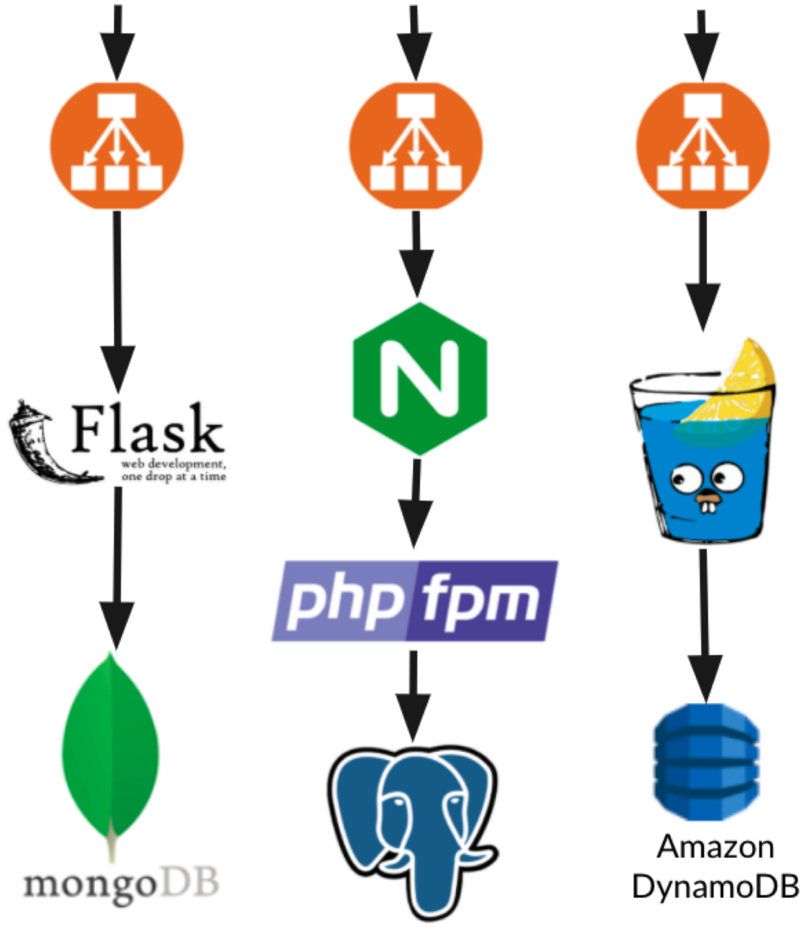
rkt

Container Runtime

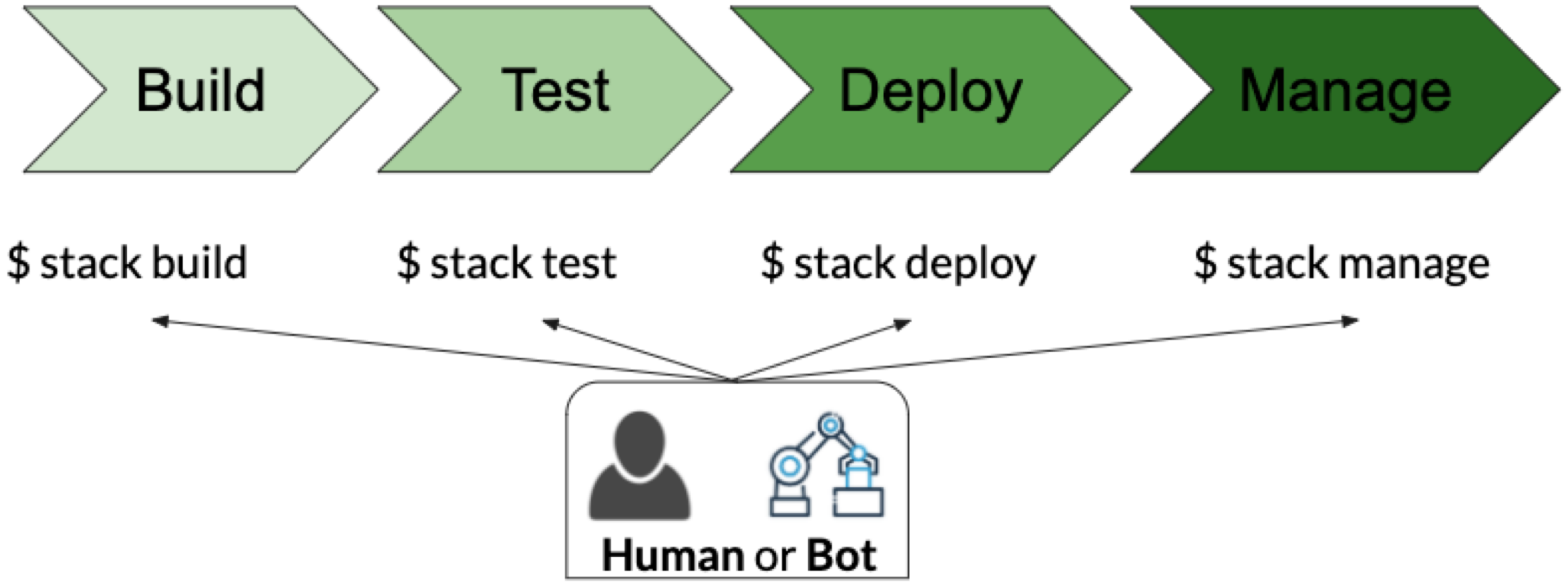
Who’s next?

Our Domains

- **Deal with a set of stacks**
  - ∑(LB, Web Server, App, DB, ...)
- **Flows**
  - Build the stack
  - Test everything works
  - Deploy to target environment
  - Keep running the stack



Standardize Workflows, Not Tools



Wrap Your Workflows: Let’s call it “Stack Pattern”

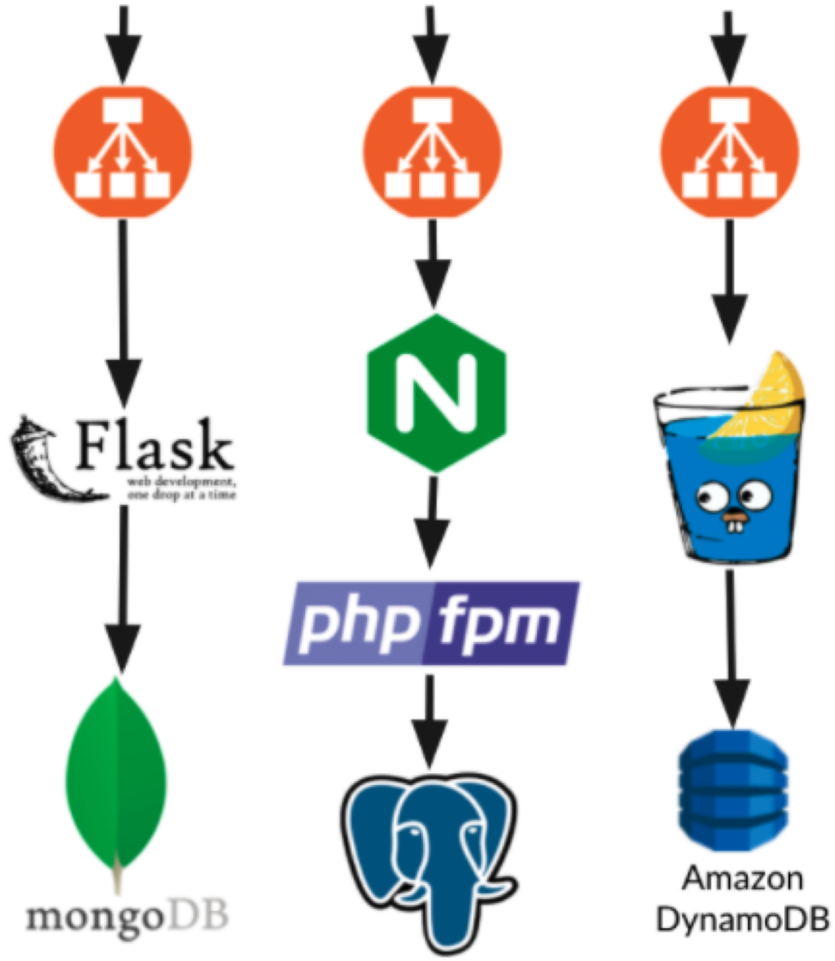
```
Usage: stack <command>
  help          Show this help

Build:
  build         Build artifacts for this stack
  push          Push artifacts to registry

Test:
  test          Run all tests for this stack

Deploy:
  deploy        Deploy the stack
  destroy       Teardown the stack

Manage:
  dump          Take a data dump
  enter         Enter app container
  restore       Restore the data from dump
```



Keep Your Workflows Composable, Swappable, Testable

```
Usage: stack <command>
  help          Show this help

Build:
  build         Build artifacts for this stack
  push          Push artifacts to registry

Test:
  test          Run all tests for this stack

Deploy:
  deploy        Deploy the stack
  destroy       Teardown the stack

Manage:
  dump          Take a data dump
  enter         Enter app container
  restore       Restore the data from dump
```

```
Usage: stack deploy <command>
  all           Deploy all resources
  cache         Deploy a cache service
  db            Deploy a database server
  ing           Deploy ingress resources
  pods          Deploy pods
  tls           Deploy TLS resources
```

Thank You!

DEVOPS WORLD by CloudBees