#RedHatOSD

RED HAT FORUMS

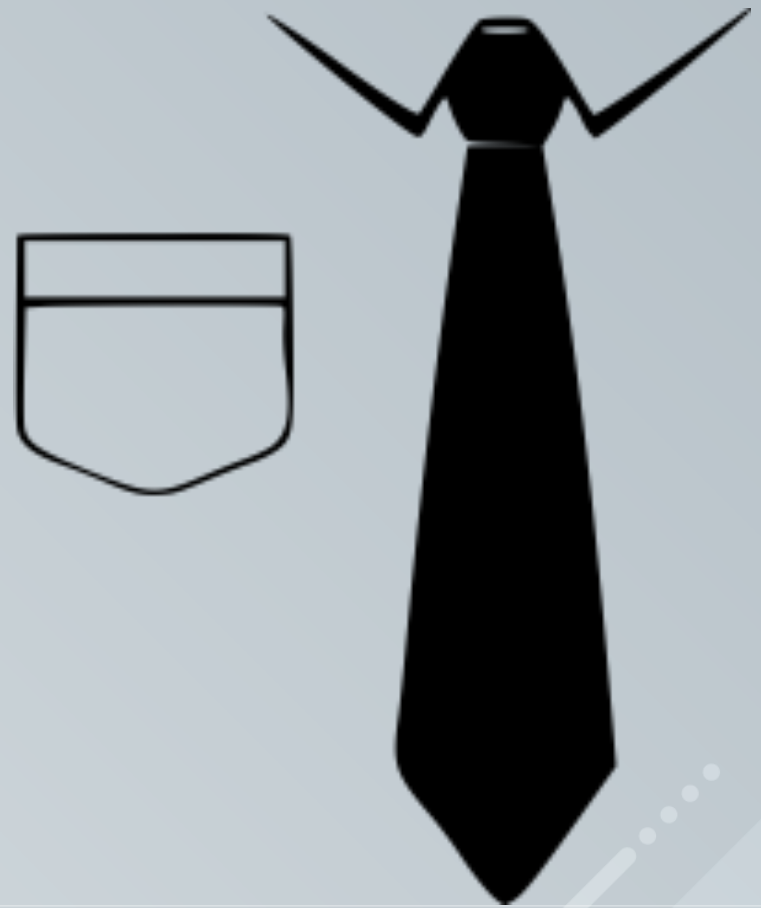
# CI/CD Redefined

Simone Gotti
Principal Software Architect
03/12/2019

Red Hat | SORINT lab

# WHO AM I?

- 20 years experience on IT
- Open Source Software Engineer and Architect
- +8K contributions in the last year on github



# LET’S START

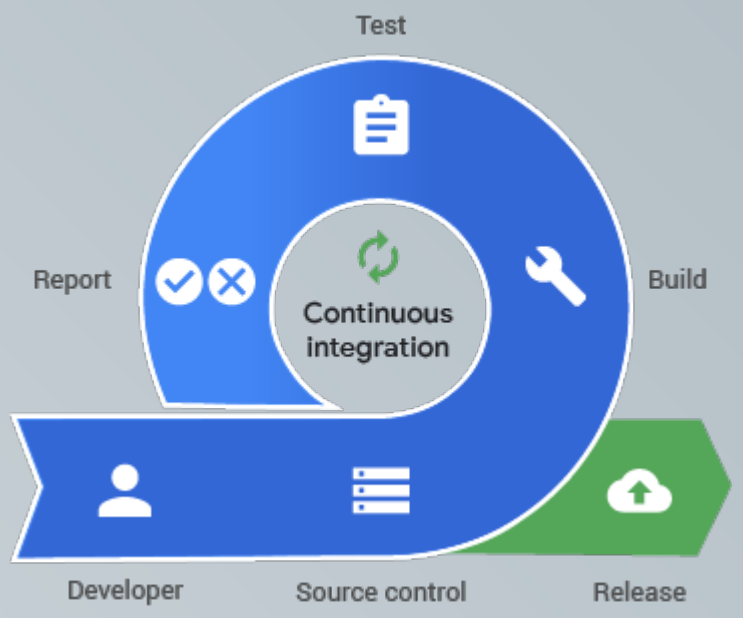
# AGENDA

- CI/CD Definition
- CI/CD Tools
- Background
- Why create a **New Tool**
- **New Tool** architecture
- My favourite feature
- Resources

# CI DEFINITION

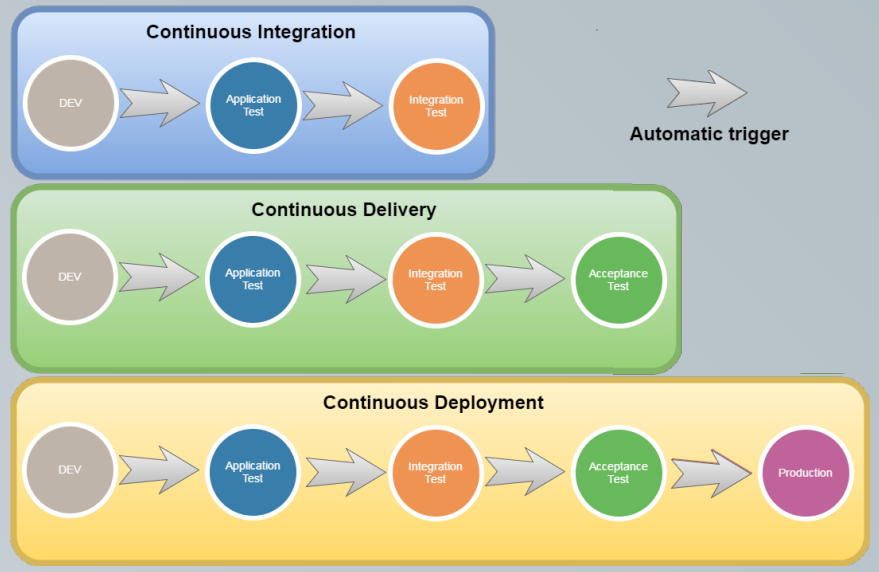
**Continuous Integration** (**CI**) is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early.

By integrating regularly, you can detect errors quickly, and locate them more easily.

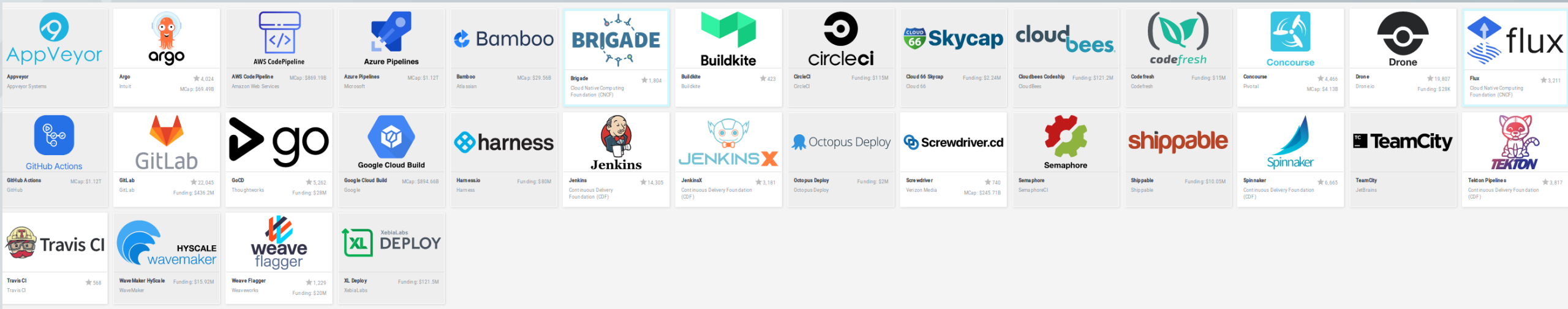


# CD DEFINITION

**Continuous delivery** (CD or CDE) is a software engineering approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time and, when releasing the software, doing so manually.
It aims at building, testing, and releasing software with greater speed and frequency. The approach helps reduce the cost, time, and risk of delivering changes by allowing for more incremental updates to applications in production.



# TOOLS


AppVeyor
Appveyor
Appveyor Systems
argo
Argo
Intuit
★ 4,024
MCap: $69.49B
AWS CodePipeline
AWS Code Pipeline
Amazon Web Services
MCap: $869.19B
Azure Pipelines
Azure Pipelines
Microsoft
MCap: $1.12T
Bamboo
Bamboo
Atlassian
MCap: $29.56B
BRIGADE
Brigade
Cloud Native Computing Foundation (CNCF)
★ 1,804
Buildkite
Buildkite
Buildkite
★ 423
circleci
CircleCI
CircleCI
Funding: $115M
CLOUD 66 Skycap
Cloud 66 Skycap
Cloud 66
Funding: $2.24M
cloudbees
Cloudbees Codeship
CloudBees
Funding: $121.2M
codefresh
Code fresh
Codefresh
Funding: $15M
Concourse
Concourse
Pivotal
★ 4,466
MCap: $4.13B
Drone
Drone
Drone.io
★ 19,807
Funding: $28K
flux
Flux
Cloud Native Computing Foundation (CNCF)
★ 3,211
GitHub Actions
GitHub Actions
GitHub
MCap: $1.12T
GitLab
GitLab
GitLab
★ 22,045
Funding: $436.2M
go
GoCD
Thoughtworks
★ 5,262
Funding: $28M
Google Cloud Build
Google Cloud Build
Google
MCap: $894.66B
harness
Harness.io
Harness
Funding: $80M
Jenkins
Jenkins
Continuous Delivery Foundation (CDF)
★ 14,305
JENKINS X
JenkinsX
Continuous Delivery Foundation (CDF)
★ 3,181
Octopus Deploy
Octopus Deploy
Octopus Deploy
Funding: $2M
Screwdriver.cd
Screwdriver
Verizon Media
★ 740
MCap: $245.71B
Semaphore
Semaphore
SemaphoreCI
shippable
Shippable
Shippable
Funding: $10.05M
Spinnaker
Spinnaker
Continuous Delivery Foundation (CDF)
★ 6,665
TeamCity
TeamCity
JetBrains
TEKTON
Tekton Pipelines
Continuous Delivery Foundation (CDF)
★ 3,817
Travis CI
Travis CI
Travis CI
★ 568
HYSCALE
wavemaker
Wave Maker HyScale
WaveMaker
Funding: $15.92M
weave flagger
Weave Flagger
Weaveworks
★ 1,229
Funding: $20M
XebiaLabs
DEPLOY
XL Deploy
XebiaLabs
Funding: $121.5M

# Background

- We innovate (CI/CD, DevOps, CLOUD, AGILE)
- We do consulting to implement CI/CD
- We used the tools contained in the slide tools and we found many problems about:
  - Segregation of duty
  - Methodologies and Opinionated Tool
  - Security
  - Cloud Native (scalability, single instance...)
  - Power to the developers no wall no silos
- With SorintOSS, we have created the solution to the problems….

# WELCOME


CI/CD Redefined

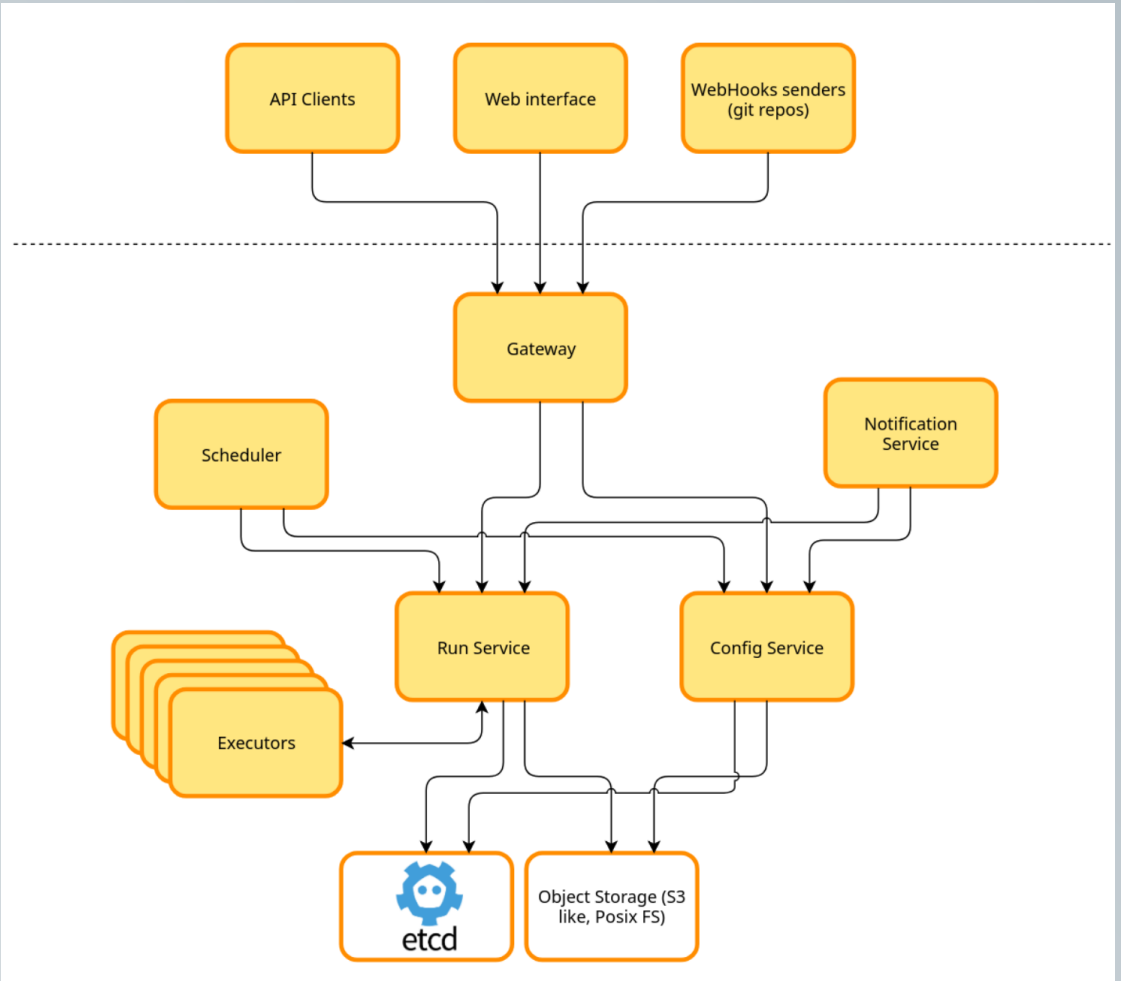
# WHY AGOLA?

- Development Speed
  - Git based workflow
  - Advanced workflows (runs) composed of many related tasks, containerized, reproducible, restartable (from start or just from failed tasks)
  - Works with any programming language/environment.
  - A unique tool for Dev Ops
  - Simple/standard/powerful run definition language (yaml, json, jsonnet, starlark)
  - User direct runs
  - Testable runs
- Deployment Speed
  - Run restart from failed tasks
- Reliability
  - At most once execution (avoid concurrent deploy issues like done by other tools)
- Development/Deployment Quality
  - Every tasks is run inside a "container"

# WHY AGOLA?

- Segregation of duties
  - Organizations, teams, RBAC (coming soon)
- Security
  - Secrets and variables
  - Run and tasks approval
- Simplicity
  - High available and scalable by nature
  - Deploy everywhere (bare metal, docker, kubernetes)
  - Cloud Native

# AGOLA ARCHITECTURE



# My Favorite Feature

## Common Problem

Users usually run the software tests manually on their workstation before committing and pushing to git. This usually requires a lot of resources, tests aren't executed in a clean environment and usually only part of the tests can be executed locally.

# My Favorite Feature

Solution: **USER DIRECT RUNS**

With agola we wanted to improve this workflow letting users execute the runs (also before committing and pushing).
These runs are called user direct runs and are executed on the agola instances in the same environment as a project run. In this way users are able to test their software and runs in the same way they will be tested when pushing them or opening a pull request. All of this won't require a super powerful workstation.

# RESOURCES

GitHub → https://github.com/agola-io/agola
Site → https://agola.io/
Forum → https://talk.agola.io/

#RedHatOSD

# RED HAT FORUMS

## THANK YOU

linkedin.com/company/Red-Hat

facebook.com/RedHatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat | SORINT lab