CI/CD CONCEPTS IN THE REAL LIFE

DEFINITION OF CI/CD, WHY WE NEED TO USE CI/CD

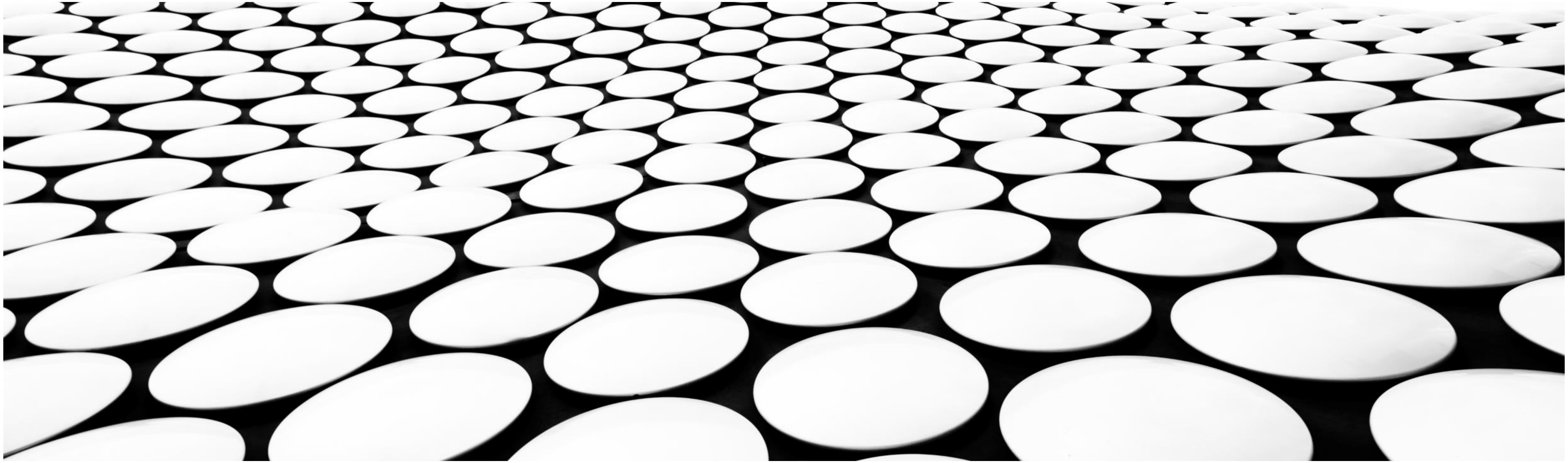
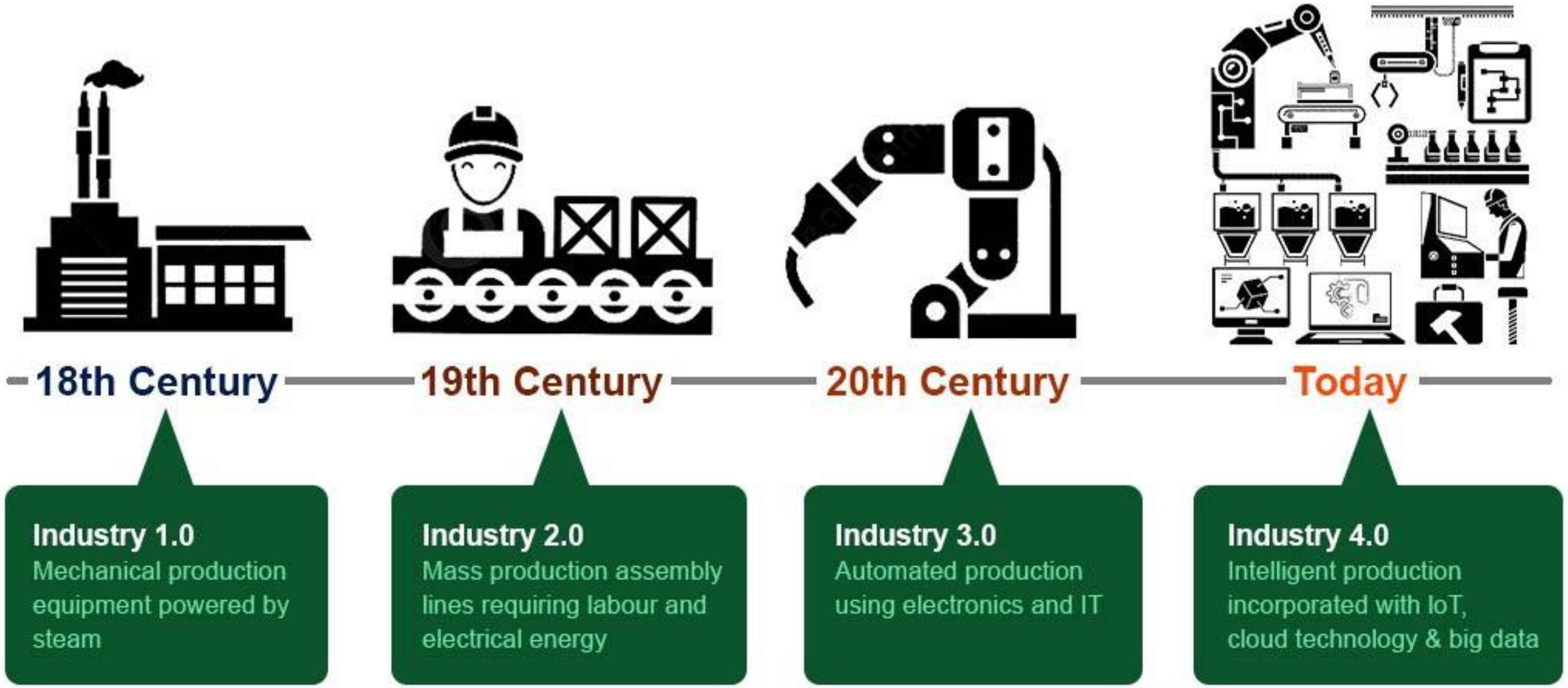


TABLE OF CONTENTS

- The evolution of industry
- What is CI/CD
- CI/CD on the business view
- Q/A

THE EVOLUTION OF INDUSTRY



18th Century

Industry 1.0
Mechanical production equipment powered by steam

19th Century

Industry 2.0
Mass production assembly lines requiring labour and electrical energy

20th Century

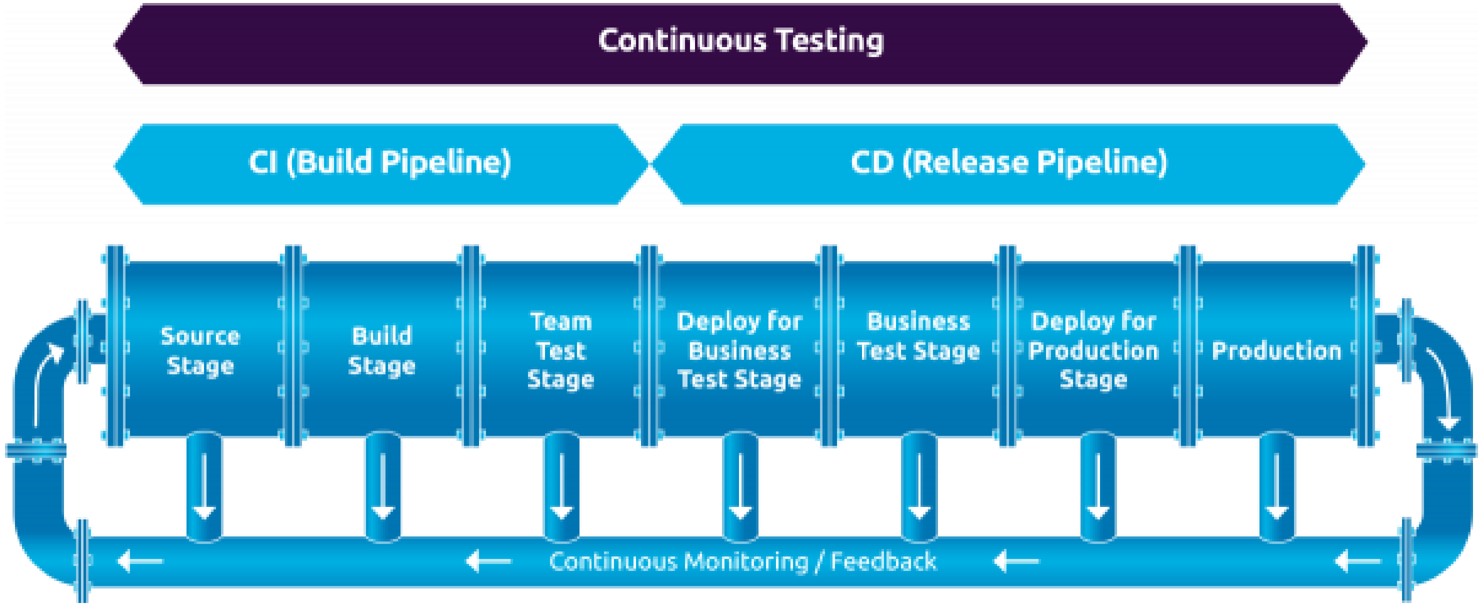
Industry 3.0
Automated production using electronics and IT

Today

Industry 4.0
Intelligent production incorporated with IoT, cloud technology & big data

CI/CD IS ONE OF KEY COMPONENT ON THE 4.0 INDUSTRY

WHAT IS CI/CD



The full flow of CI/CD

WHAT IS CI/CD

- CI/CD is the abbreviation of Continuous Integration / Continuous Deployment. CI/CD is seen as the backbone to enable DevOps.
- CI/CD automates much or all of the manual human intervention traditionally needed to get new code from a commit into production, encompassing the build, test (including integration tests, unit tests, and regression tests), and deploy phases, as well as infrastructure provisioning.

WHAT IS CI

- **Continuous integration** is the practice of integrating all your code changes into the main branch of a shared source code repository early and often, automatically testing each change when you commit or merge them, and automatically kicking off a build.
- With continuous integration, errors and security issues can be identified and fixed more easily, and much earlier in the development process.
- Some common CI-related phases might include*:*
  - Compile
  - Unit Test
  - Static Analysis
  - Dependency vulnerability testing
  - Store artifact

WHAT IS CD

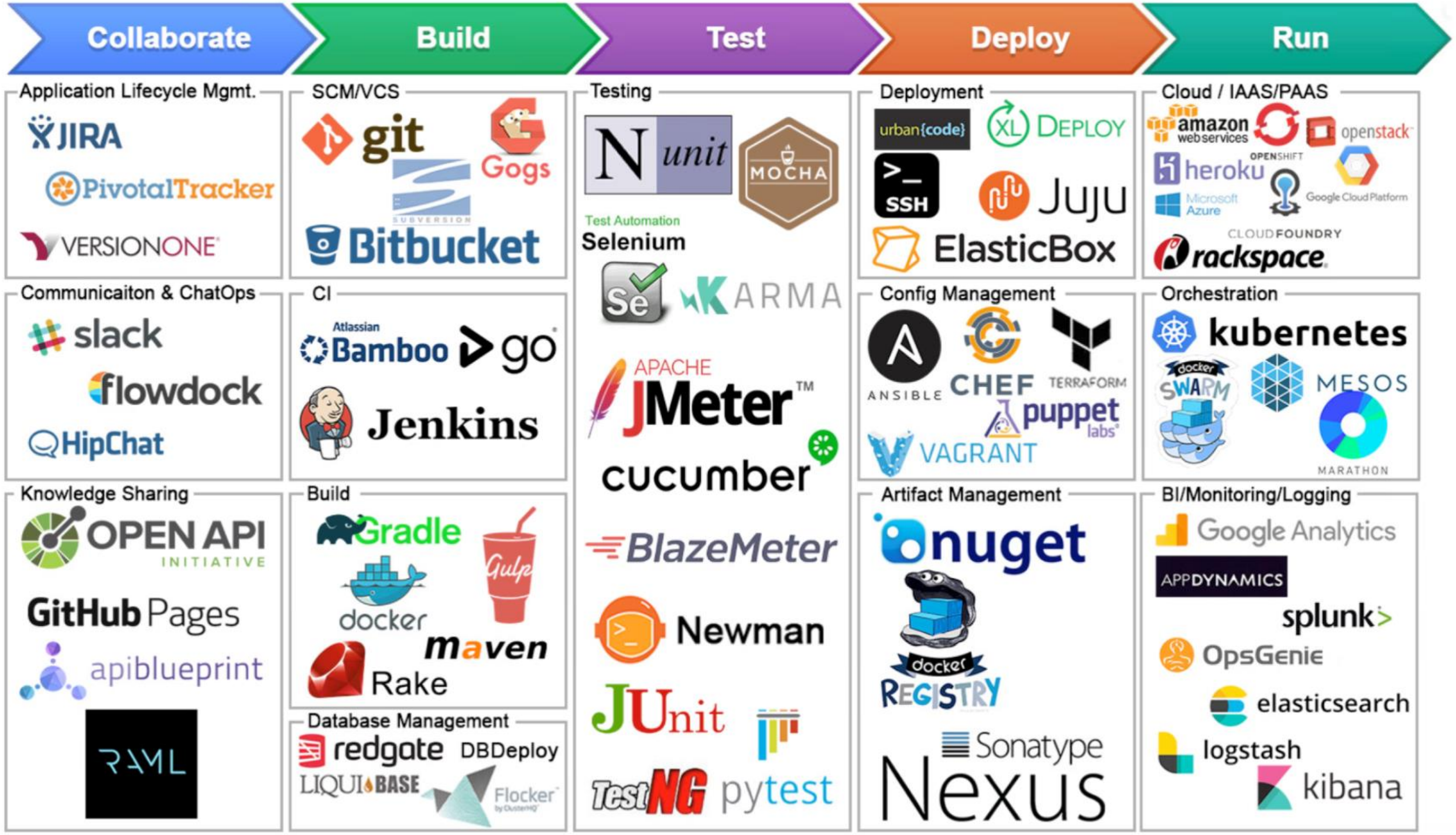
- Continuous delivery is a software development practice that works in conjunction with CI to automate the infrastructure provisioning and application release process.
- Some common CD-related phases might include:
  - Creating infrastructure
  - Provisioning servers
  - Copying files
  - Promoting to production
  - Smoke Testing (aka Verify)
  - Rollbacks

CI/CD ON THE BUSINESS

CI/CD FROM BUSINESS VIEW

Reason	Value	Technical side
Less developer time on issues from new developer code	Reduce Cost	Catch Compile Errors After Merge
New value-generating features released more quickly	Increase Revenue	Faster and More Frequent Production Deployments
Less time to market	Increase Revenue	Deploy to Production Without Manual Checks
Reduced downtime from a deploy-related crash or major bug	Protect Revenue	Automated Smoke Tests
Quick undo to return production to working state	Protect Revenue	Automated Rollback Triggered by Job Failure
Less bugs in production and less time in testing	Avoid Cost	Catch Unit Test Failures
Less infrastructure costs from unused resources	Reduce Cost	Automate Infrastructure Cleanup
Less human error, Faster deployments	Avoid Cost	Automate Infrastructure Creation

CI/CD TOOLS



Collaborate

Application Lifecycle Mgmt.
- JIRA
- PivotalTracker
- VERSIONONE

Communicaiton & ChatOps
- slack
- flowdock
- HipChat

Knowledge Sharing
- OPEN API INITIATIVE
- GitHub Pages
- apiblueprint
- RAML

Build

SCM/VCS
- git
- Gogs
- SUBVERSION
- Bitbucket

CI
- Atlassian Bamboo
- go
- Jenkins

Build
- Gradle
- Gulp
- docker
- maven
- Rake

Database Management
- redgate
- DBDeploy
- LIQUIBASE
- Flocker

Test

Testing
- Nunit
- MOCHA
- Test Automation Selenium
- Se
- KARMA
- APACHE JMeter™
- cucumber
- BlazeMeter
- Newman
- JUnit
- TestNG
- pytest

Deploy

Deployment
- urban{code}
- XL DEPLOY
- SSH
- Juju
- ElasticBox

Config Management
- ANSIBLE
- CHEF
- TERRAFORM
- puppet labs
- VAGRANT

Artifact Management
- nuget
- docker REGISTRY
- Sonatype Nexus

Run

Cloud / IAAS/PAAS
- amazon web services
- OPENSHIFT
- openstack
- heroku
- Microsoft Azure
- Google Cloud Platform
- CLOUD FOUNDRY
- rackspace

Orchestration
- kubernetes
- docker SWARM
- MESOS
- MARATHON

BI/Monitoring/Logging
- Google Analytics
- APPDYNAMICS
- splunk>
- OpsGenie
- elasticsearch
- logstash
- kibana

Q&A

Thank You