Microsoft

Microsoft Ignite

Learn.

Connect.

Explore.

Microsoft

BRK2392

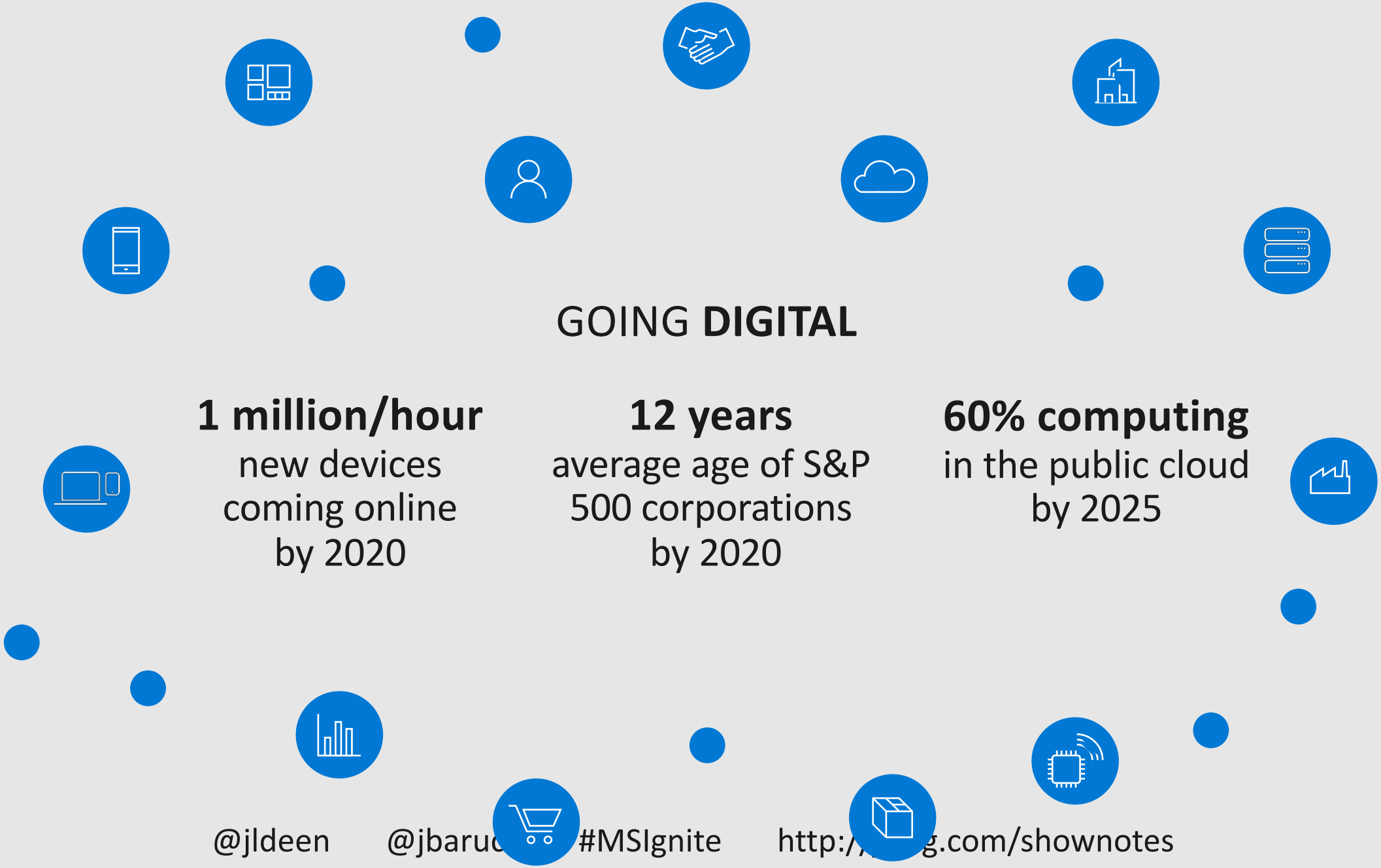
Master Tooling for Containers

Baruch Sadogursky and Jessica deen

@jldeen @jbaruch #MSIgnite http://jfrog.com/shownotes

About us

- Jessica Deen
  - Developer Advocate, Azure
  - @jldeen
- Baruch Sadogursky
  - Developer Advocate, JFrog
  - @jbaruch
- Shownotes
  - http://jfrog.com/shownotes



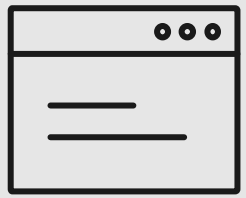
GOING **DIGITAL**

1 million/hour
new devices
coming online
by 2020

12 years
average age of S&P
500 corporations
by 2020

60% computing
in the public cloud
by 2025

What we hear from developers



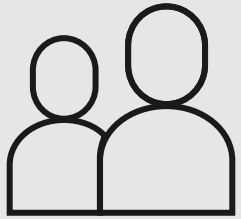
I need to create applications at a competitive rate without worrying about IT

New applications run smoothly on my machine but malfunction on traditional IT servers

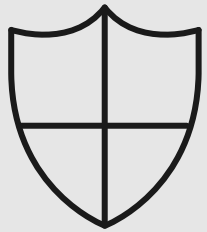
My productivity and application innovation become suspended when I have to wait on IT

What we hear from IT

I need to manage servers and maintain compliance with little disruption



I'm unsure of how to integrate unfamiliar applications, and I require help from developers

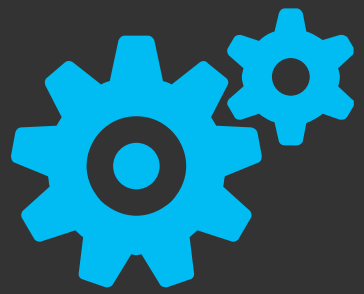


I'm unable to focus on both server protection and application compliance

IT stress points

Security threats

Datacenter efficiency



Supporting innovation

Cloud is a new way to think about a datacenter

Traditional model

Dedicated infrastructure for each application

Purpose-built hardware

Distinct infrastructure and operations teams

Customized processes and configurations

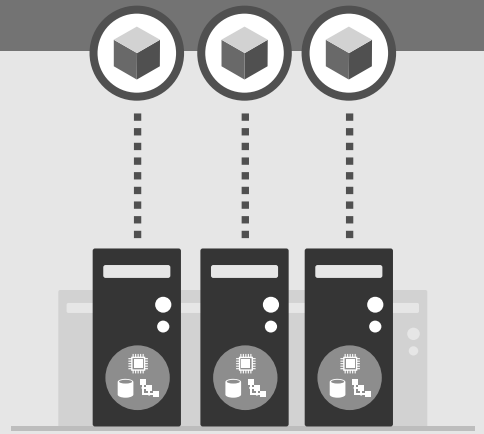
Cloud model

Loosely coupled apps and micro-services

Industry-standard hardware

Service-focused DevOps teams

Standardized processes and configurations

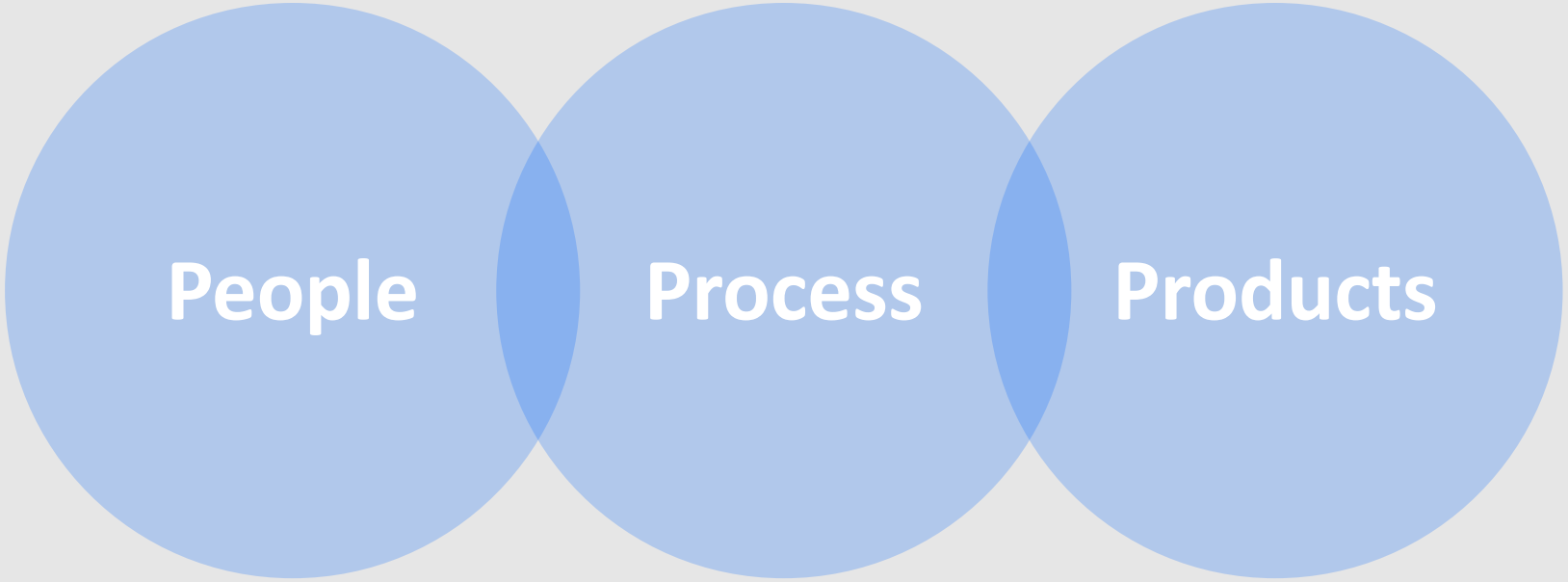


Servers

Services

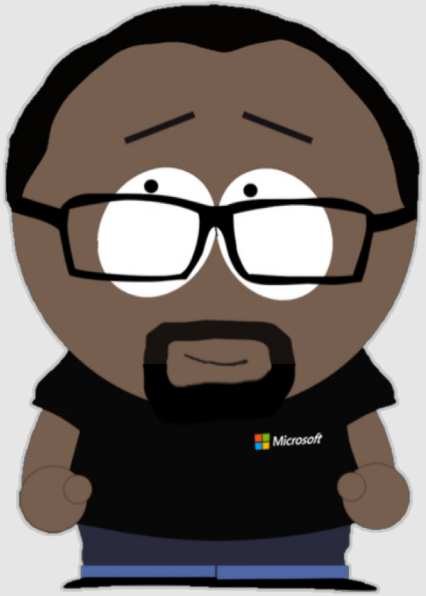
Microsoft

DevOps: The Three Stage Conversation



DevOps is the union of people, process, and products to enable continuous delivery of value to our end users.

-Donovan Brown



http://bit.ly/WhatIs-DevOps

Key DevOps Practices

Infrastructure as Code	Continuous Integration	Continuous Deployment
Automated Testing	Release Management	Performance Monitoring
Availability Monitoring	Load Testing & Auto Scale	Automated Recovery (Rollback & Roll Forward)

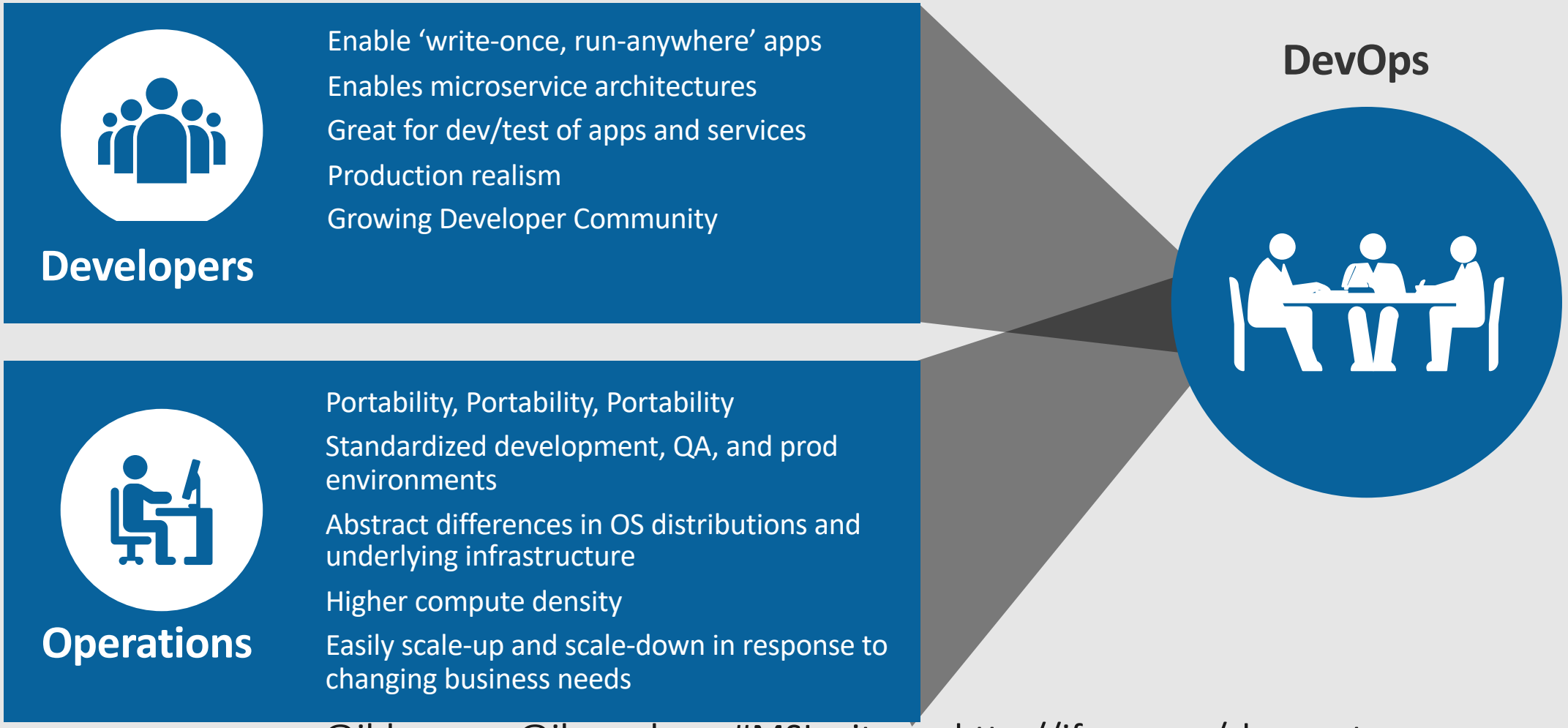
DevOps Benefits

IT Performance Metrics

	2015	2016	2017
Deployment Frequency	30x more frequent	200x more frequent	46x more frequent
Lead Time for Changes	200x faster	2,555x faster	440x faster
Mean Time to Recover (MTTR)	168x faster	24x faster	96x faster
Change Failure Rate		3x lower (1/3 as likely)	5x lower (1/5 as likely)

Source: https://puppetlabs.com

Why Containers?



What is a Container?

Not a real thing. An application delivery mechanism with **process isolation** based on several **Linux kernel** features.

Namespaces (what a process can see)

- **PID**
- **Mount**
- **Network**
- **UTS**
- **IPC**
- **User**
- **Cgroup**

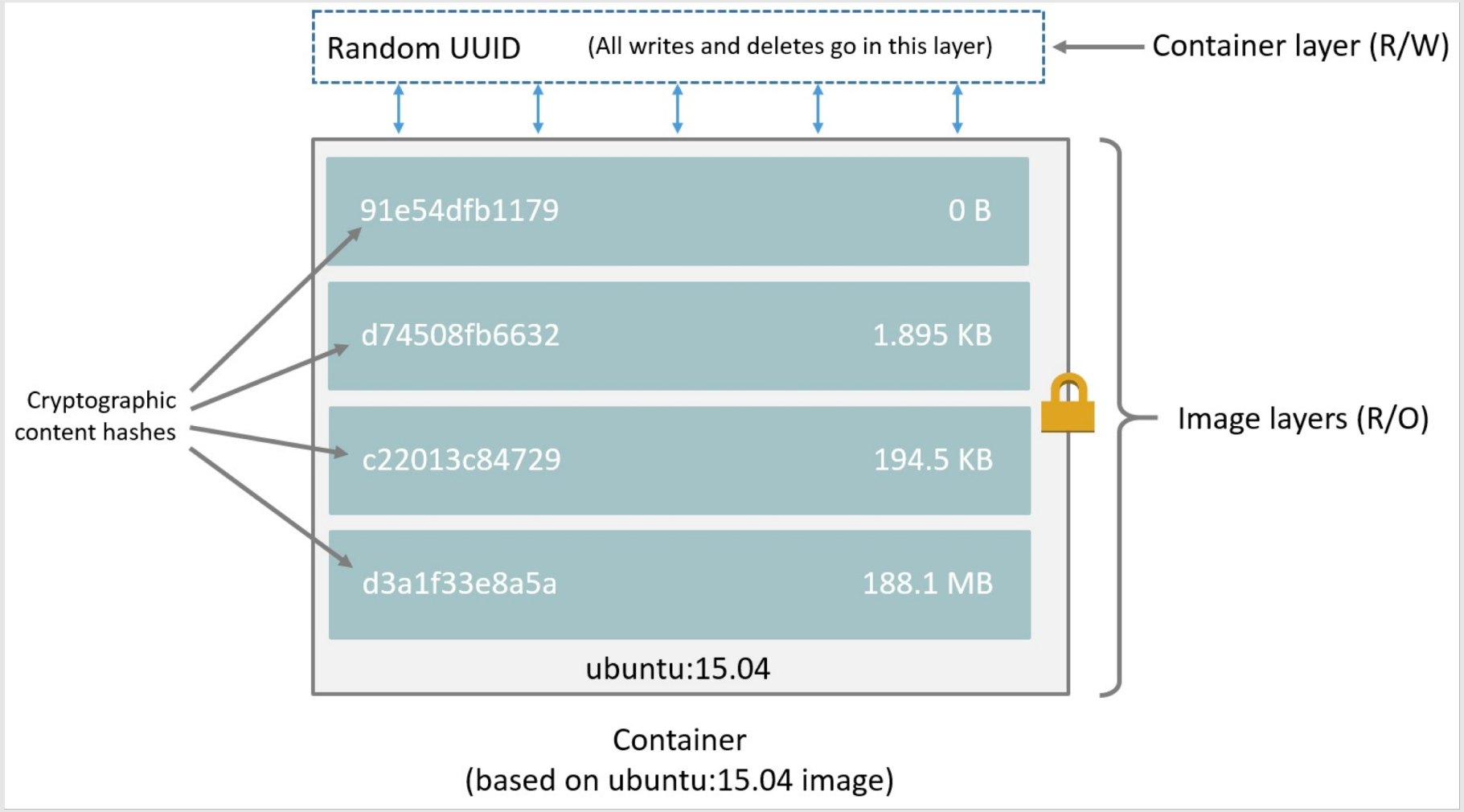
Cgroups (what a process can use)

- **Memory**
- **CPU**
- **Blkio**
- **Cpuacct**
- **Cpuset**
- **Devices**
- **Net_prio**

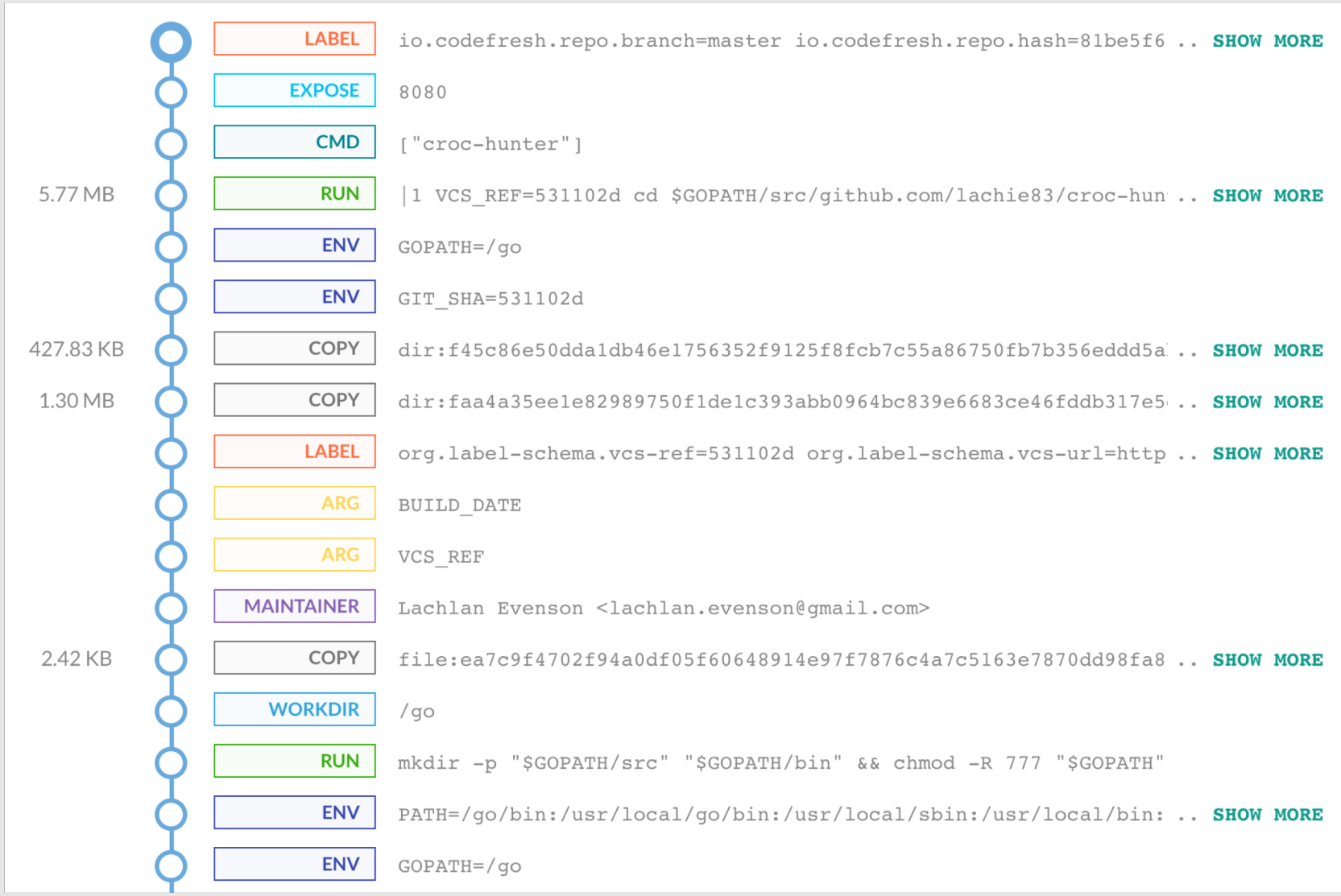
What is

- Open Source Container Runtime
- Mac, Linux, Windows Support
- Command Line Tool
- “Dockerfile” format
- The Docker image format with layered filesystem

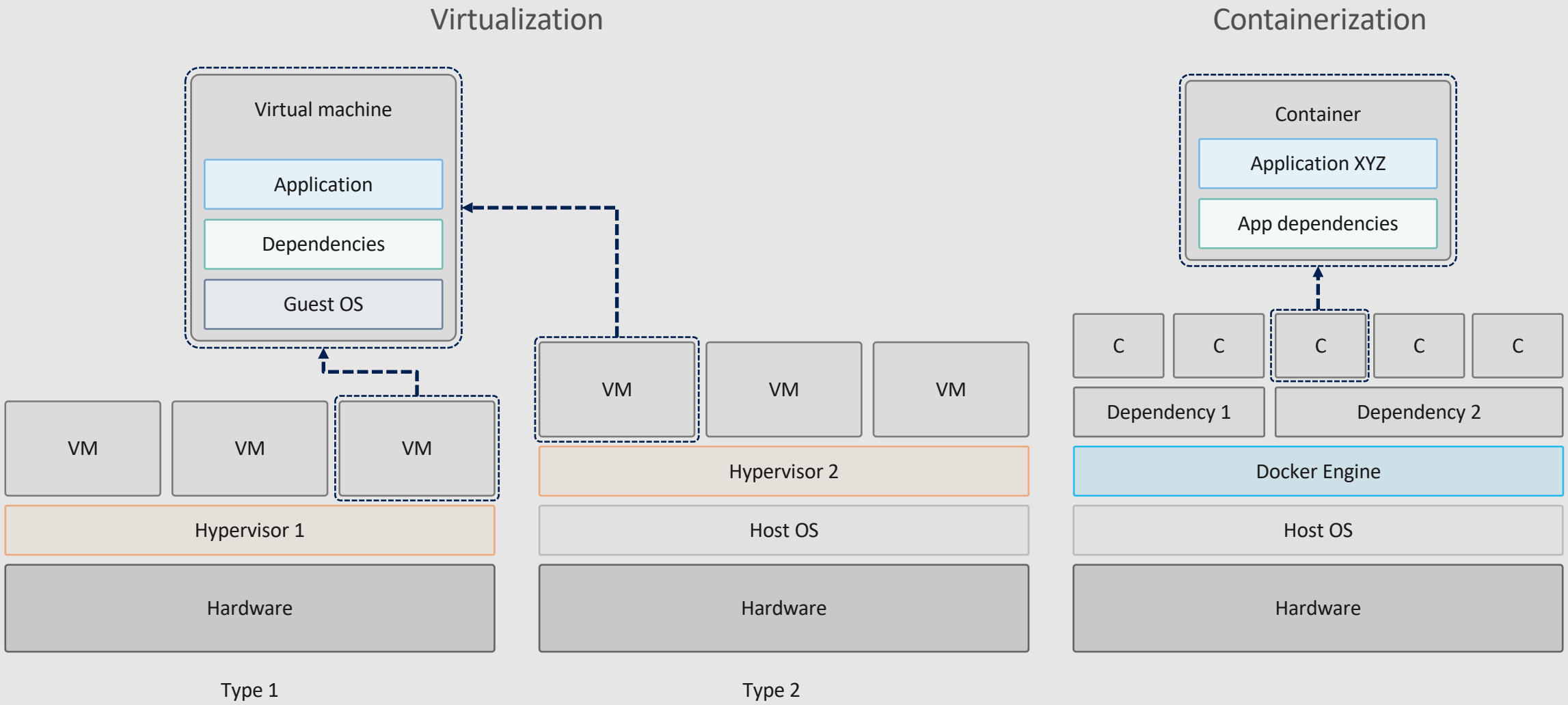
Docker Layered Filesystem



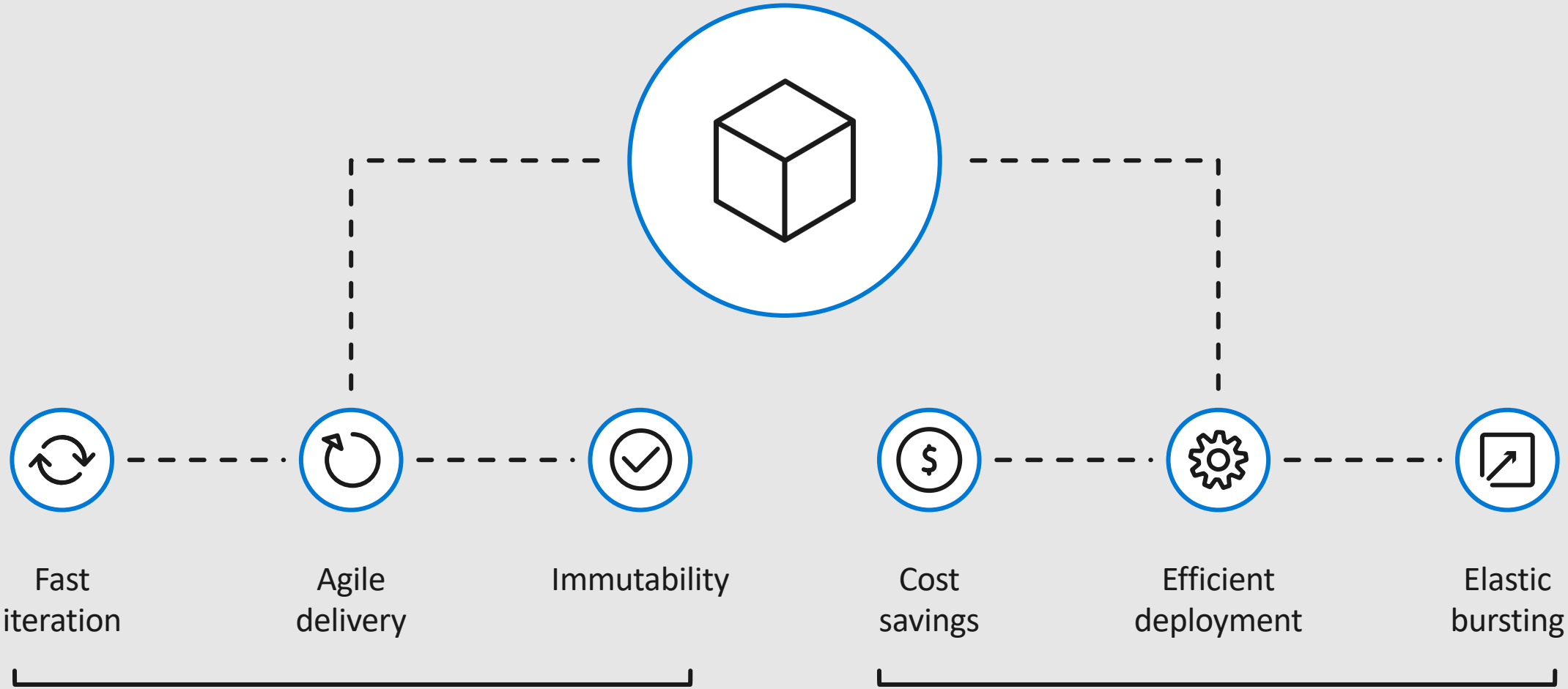
Docker Layered Filesystem



Virtualization versus containerization



The container advantage

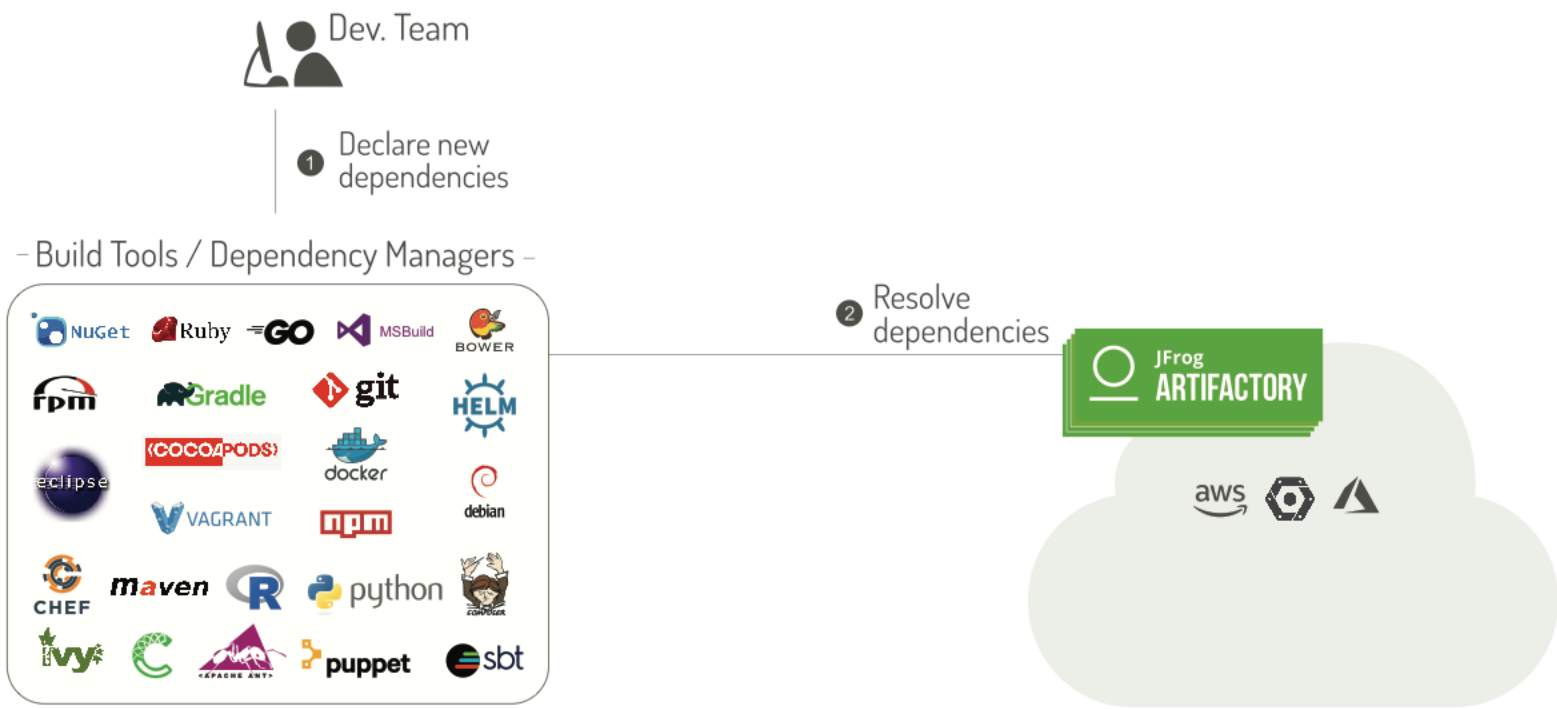


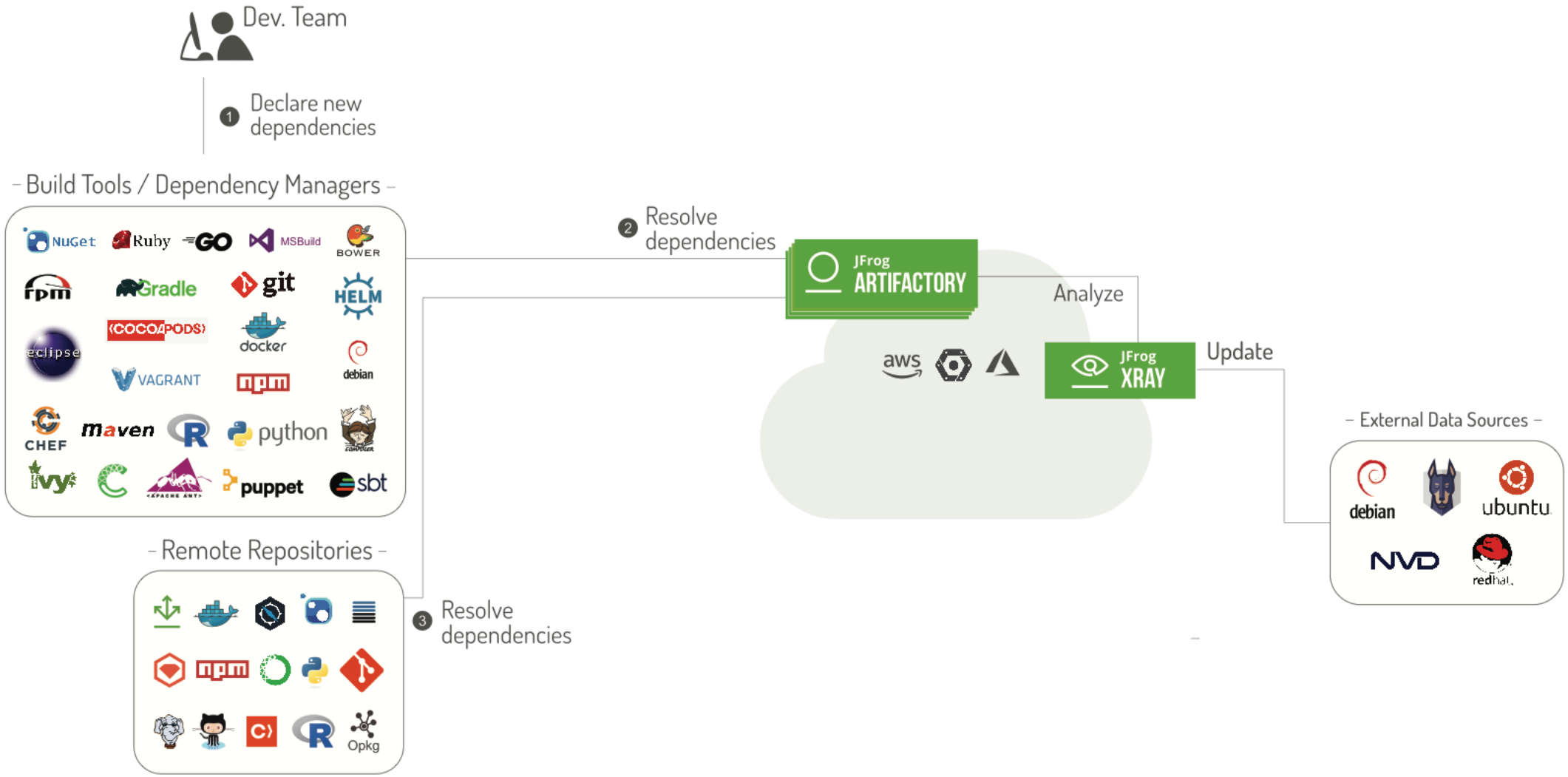
@jldeen @jbaruch #MSIgnite http://jfrog.com/shownotes

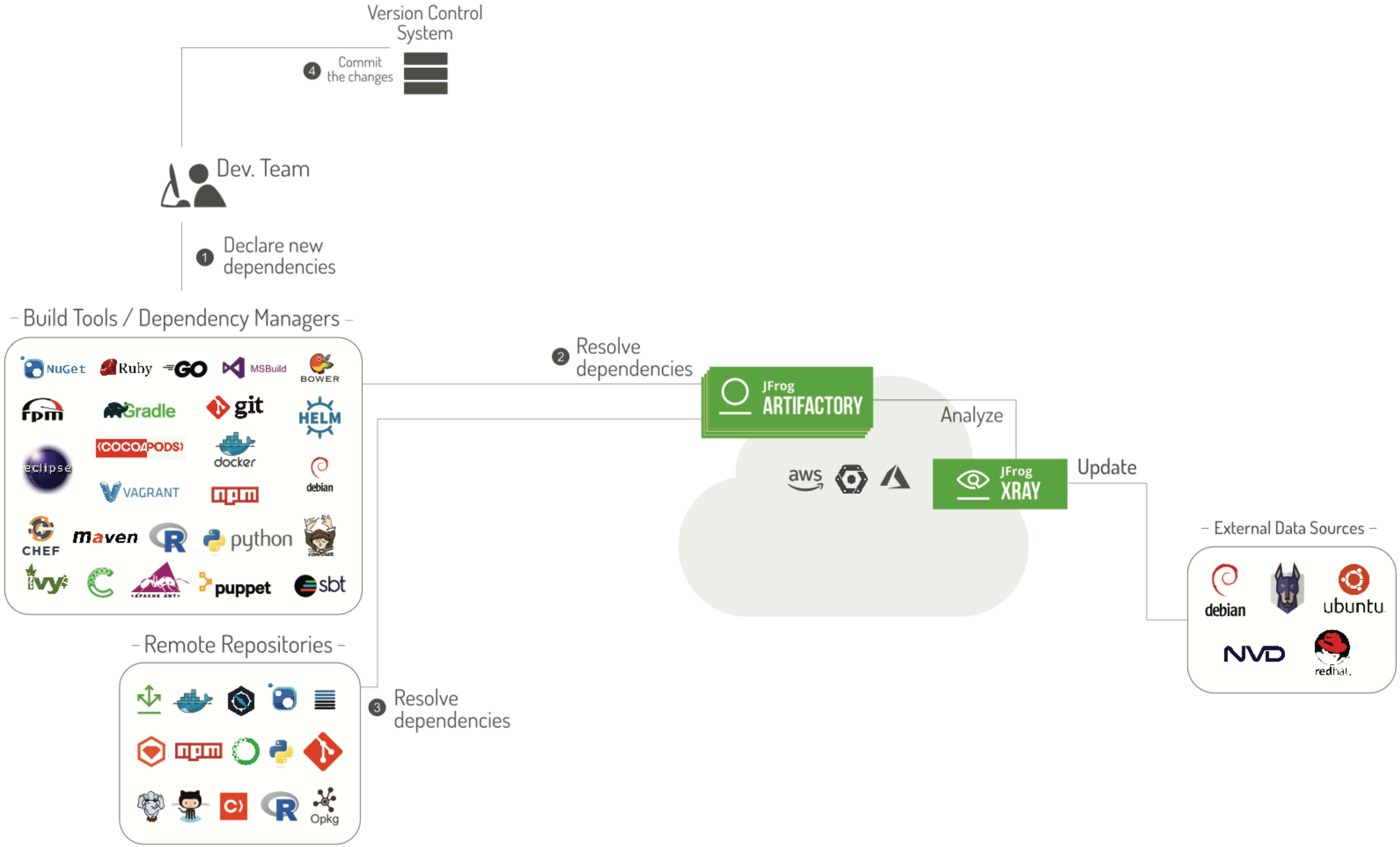
Demo

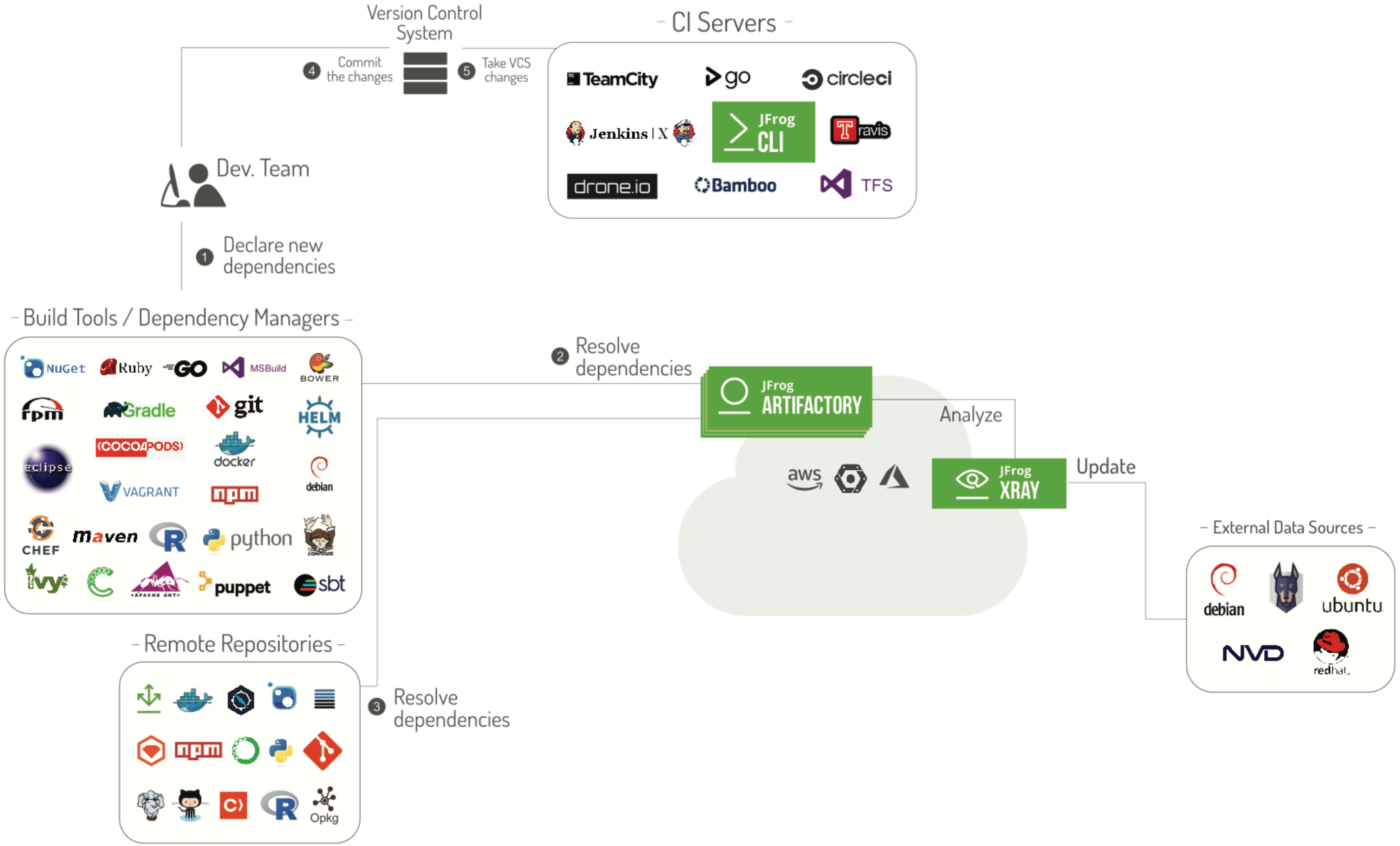

Dev. Team

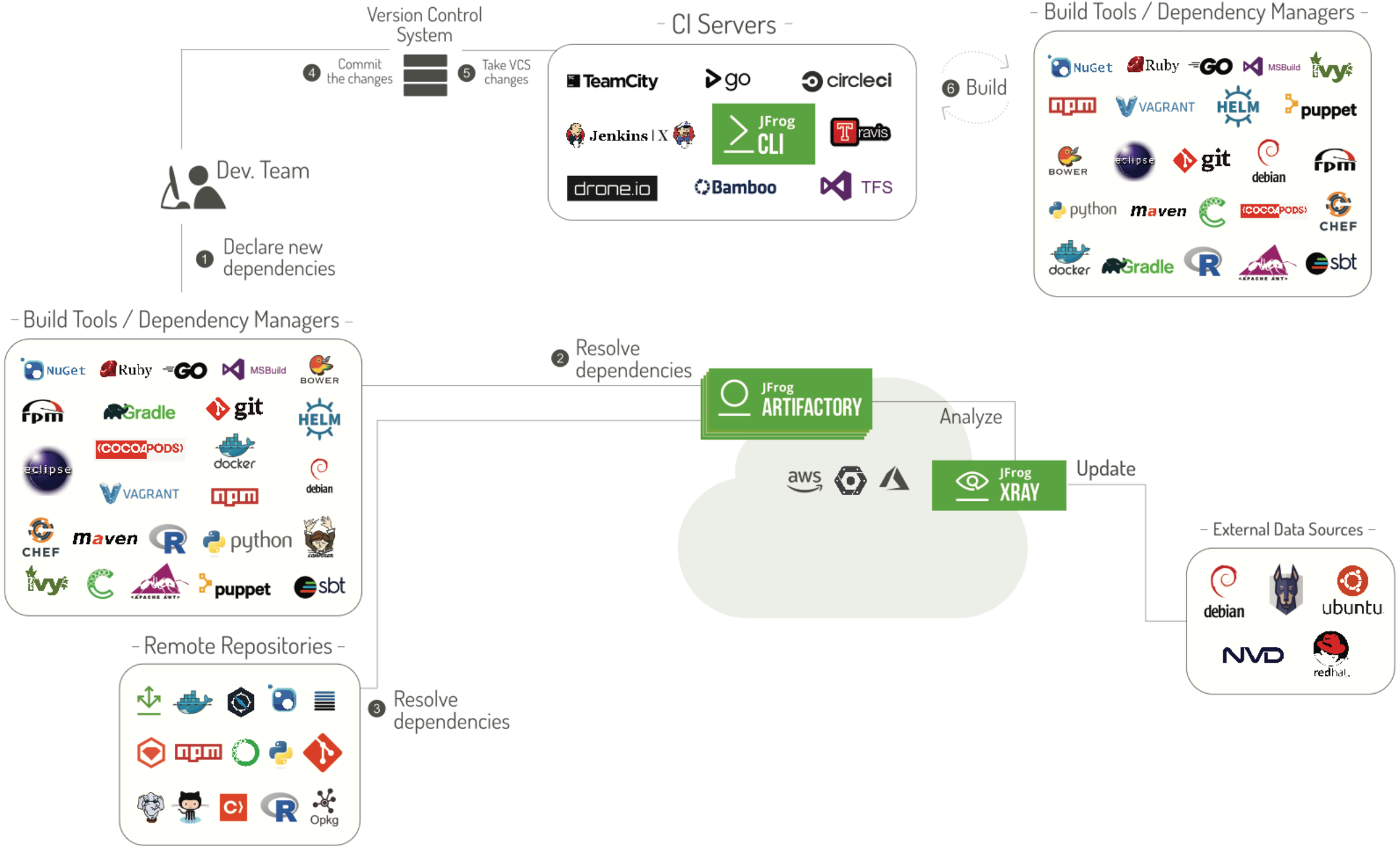

Dev. Team
1
Declare new dependencies
Build Tools / Dependency Managers
NuGet
Ruby
GO
MSBuild
BOWER
fpm
Gradle
git
HELM
COCOAPODS
docker
eclipse
debian
VAGRANT
npm
CHEF
maven
python
ivy
puppet
sbt

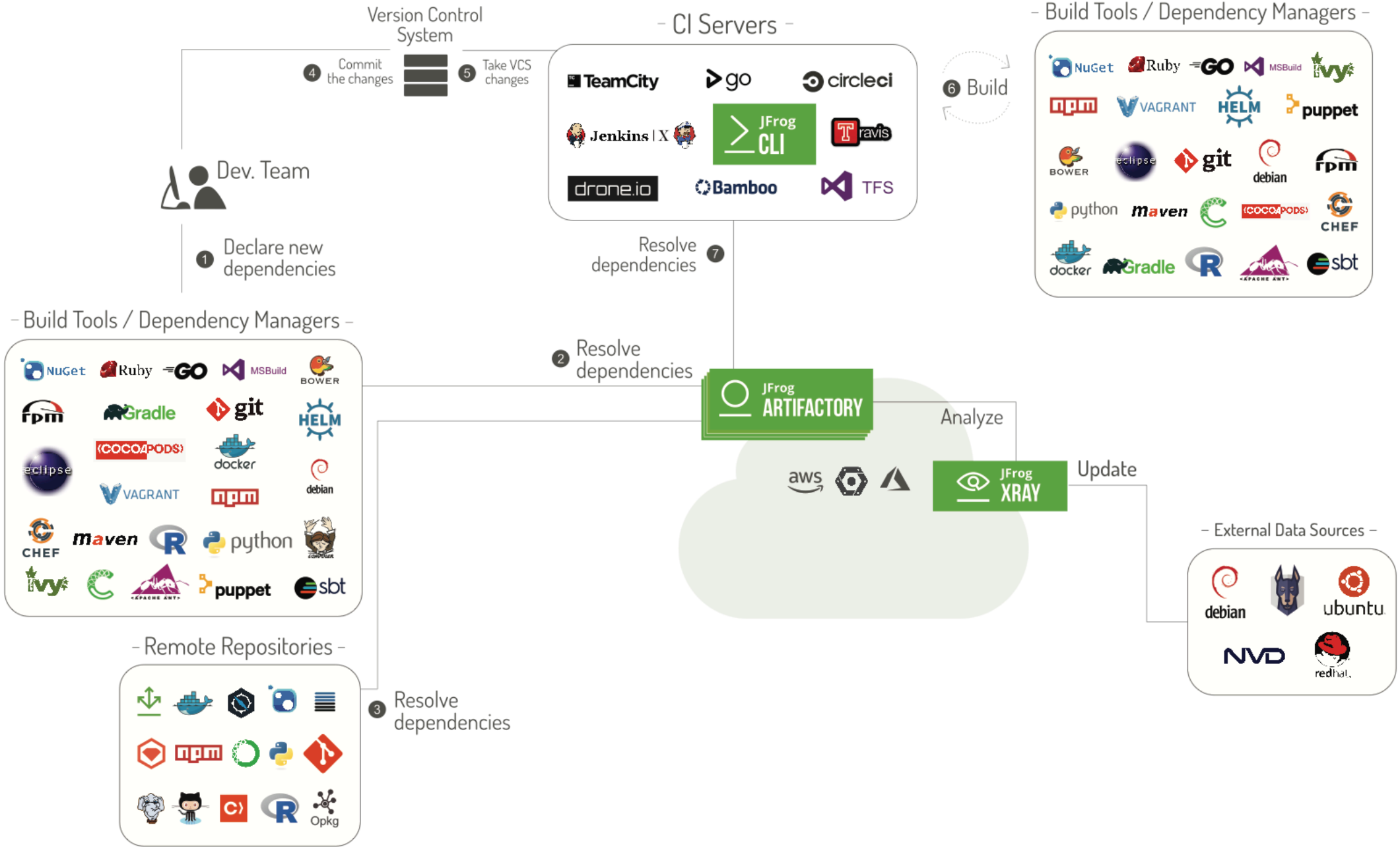

Dev. Team
1 Declare new dependencies
Build Tools / Dependency Managers
NuGet
Ruby
GO
MSBuild
BOWER
rpm
Gradle
git
HELM
COCOAPODS
eclipse
docker
debian
VAGRANT
npm
CHEF
maven
R
python
ivy
puppet
sbt
2 Resolve dependencies
JFrog
ARTIFACTORY
aws

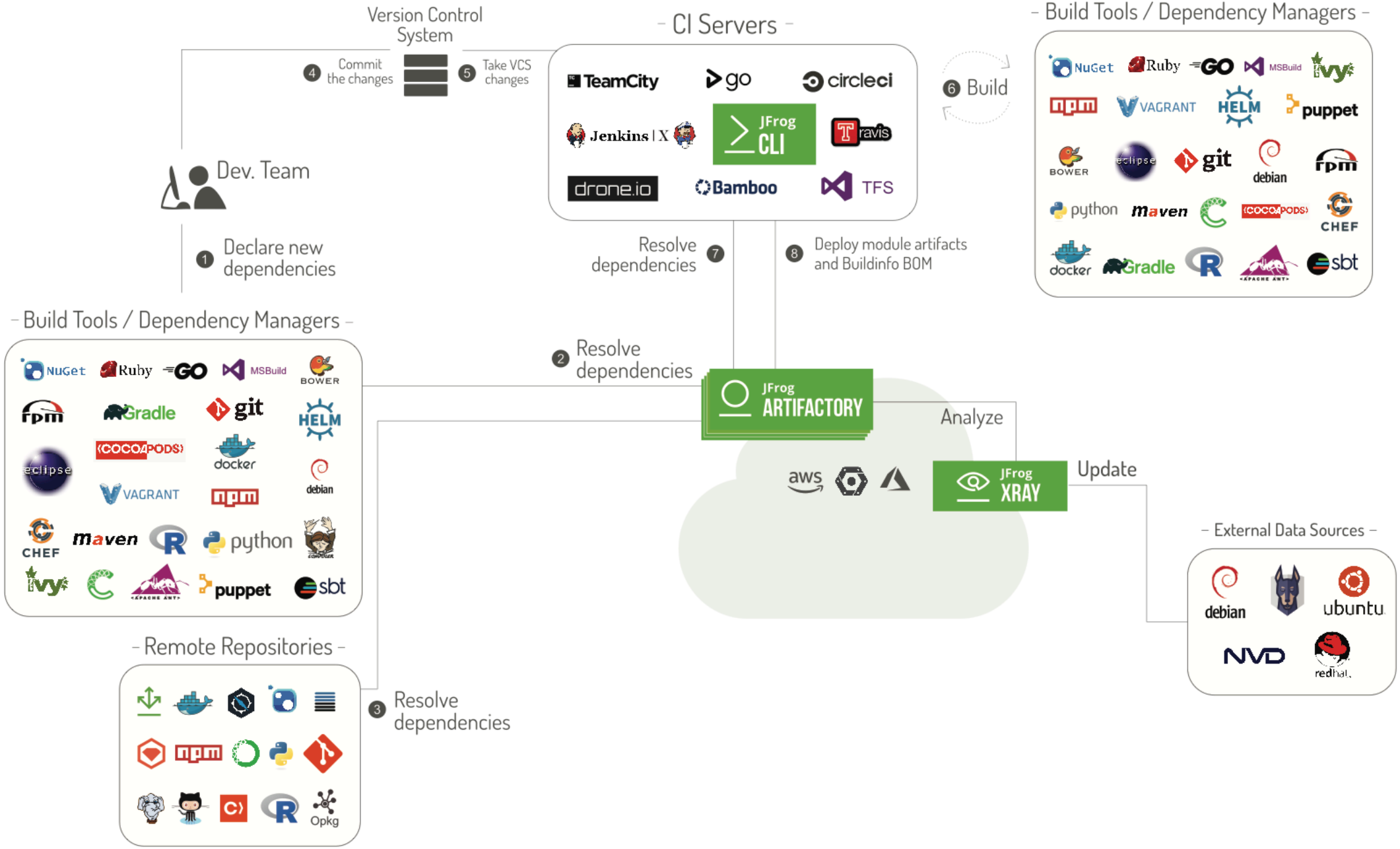

Dev. Team
1 Declare new dependencies
– Build Tools / Dependency Managers –
NuGet
Ruby
GO
MSBuild
BOWER
fpm
Gradle
git
HELM
eclipse
COCOAPODS
docker
debian
VAGRANT
npm
CHEF
maven
python
ivy
puppet
sbt
2 Resolve dependencies
JFrog ARTIFACTORY
Analyze
aws
JFrog XRAY
Update
– External Data Sources –
debian
ubuntu
NVD
redhat.
– Remote Repositories –
npm
Opkg
3 Resolve dependencies

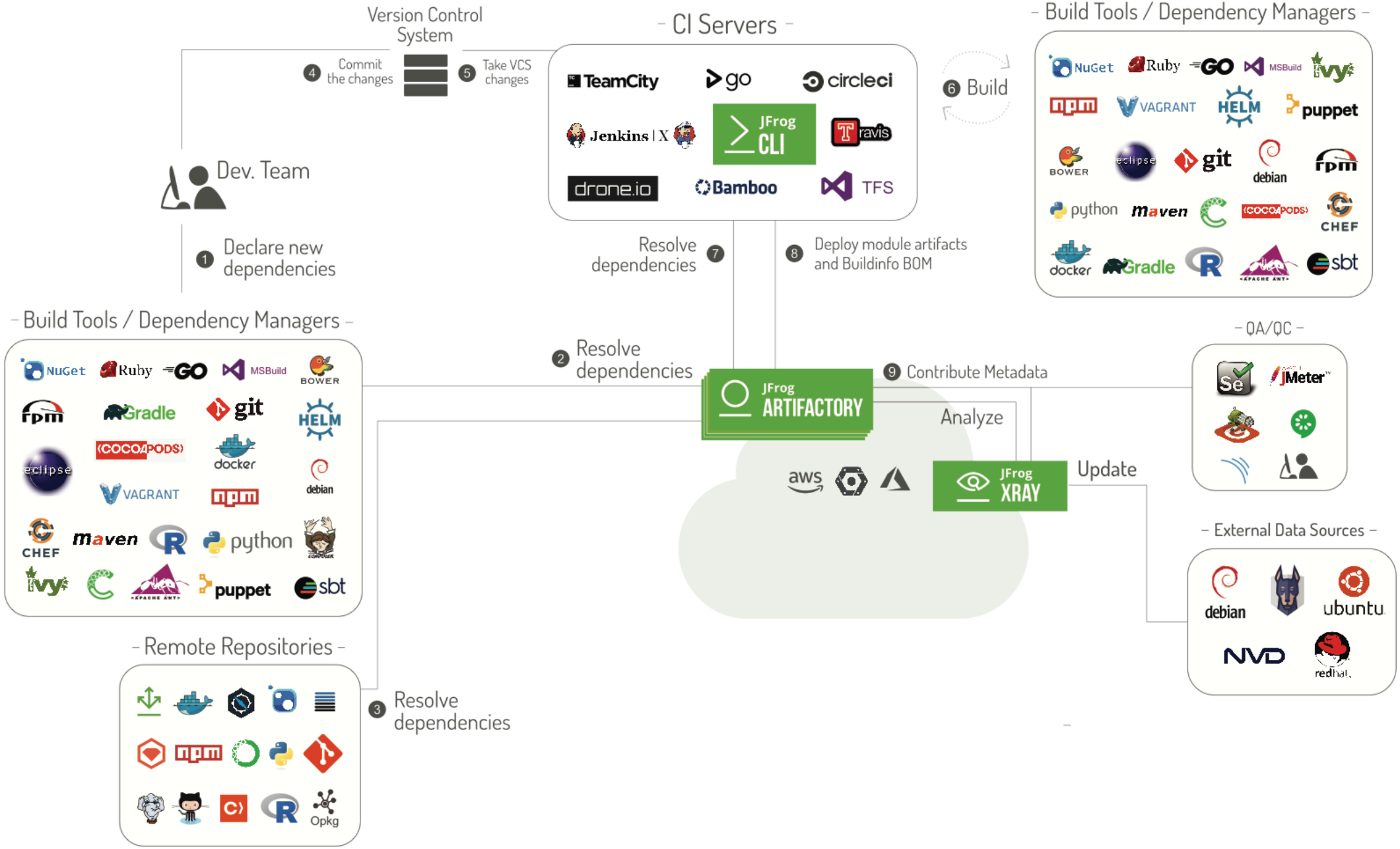

Version Control System
4 Commit the changes
Dev. Team
1 Declare new dependencies
– Build Tools / Dependency Managers –
NuGet
Ruby
GO
MSBuild
BOWER
fpm
Gradle
git
HELM
eclipse
COCOAPODS
docker
debian
VAGRANT
npm
CHEF
maven
python
ivy
puppet
sbt
2 Resolve dependencies
JFrog ARTIFACTORY
Analyze
aws
JFrog XRAY
Update
– External Data Sources –
debian
ubuntu
NVD
redhat.
– Remote Repositories –
npm
Opkg
3 Resolve dependencies

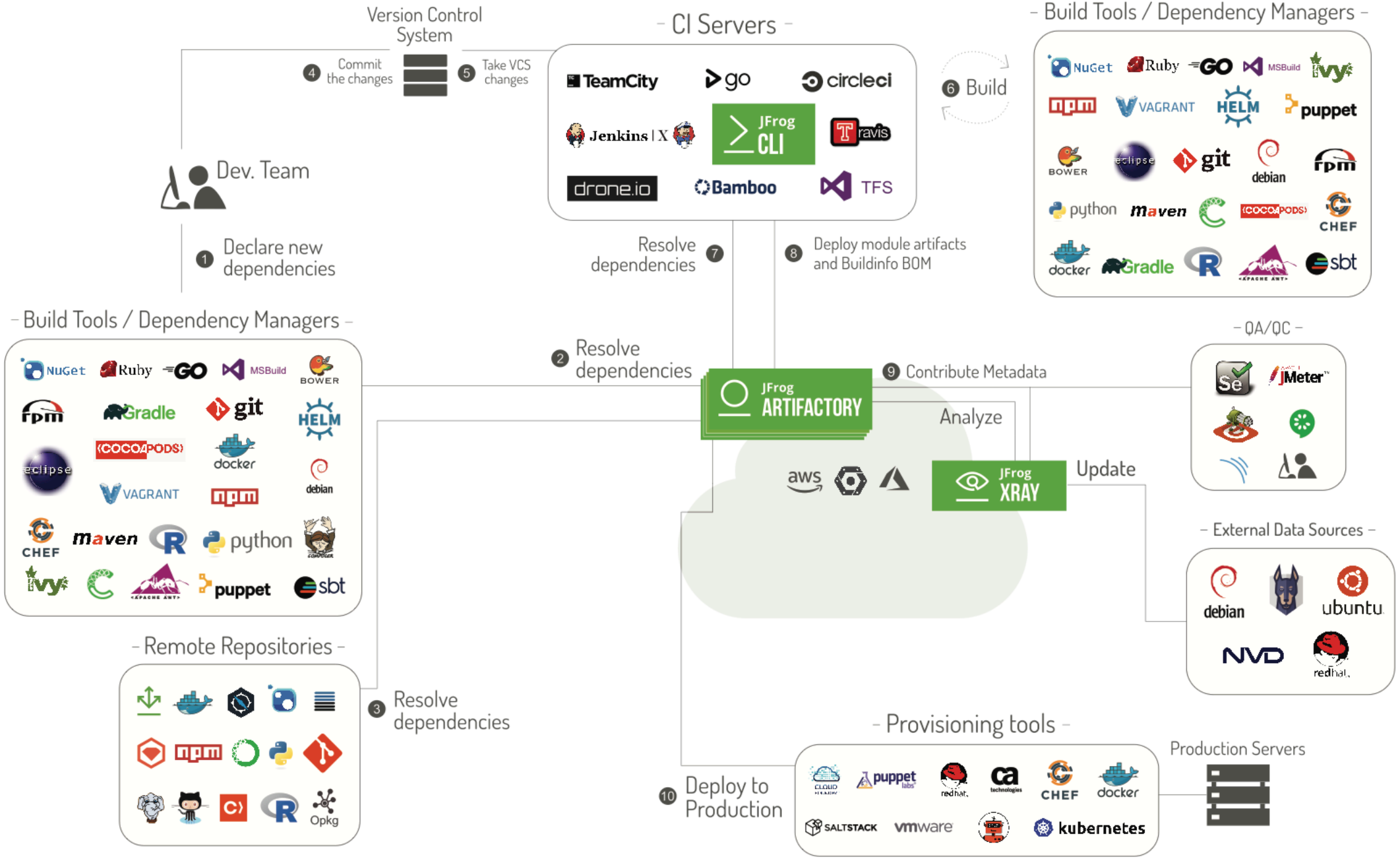

Version Control System
4 Commit the changes
5 Take VCS changes
CI Servers
TeamCity
go
circleci
Jenkins | X
JFrog CLI
Travis
drone.io
Bamboo
TFS
Dev. Team
1 Declare new dependencies
Build Tools / Dependency Managers
NuGet
Ruby
GO
MSBuild
BOWER
fpm
Gradle
git
HELM
COCOAPODS
docker
eclipse
debian
VAGRANT
npm
CHEF
maven
python
ivy
puppet
sbt
2 Resolve dependencies
JFrog ARTIFACTORY
Analyze
aws
JFrog XRAY
Update
External Data Sources
debian
ubuntu
NVD
redhat
Remote Repositories
npm
Opkg
3 Resolve dependencies

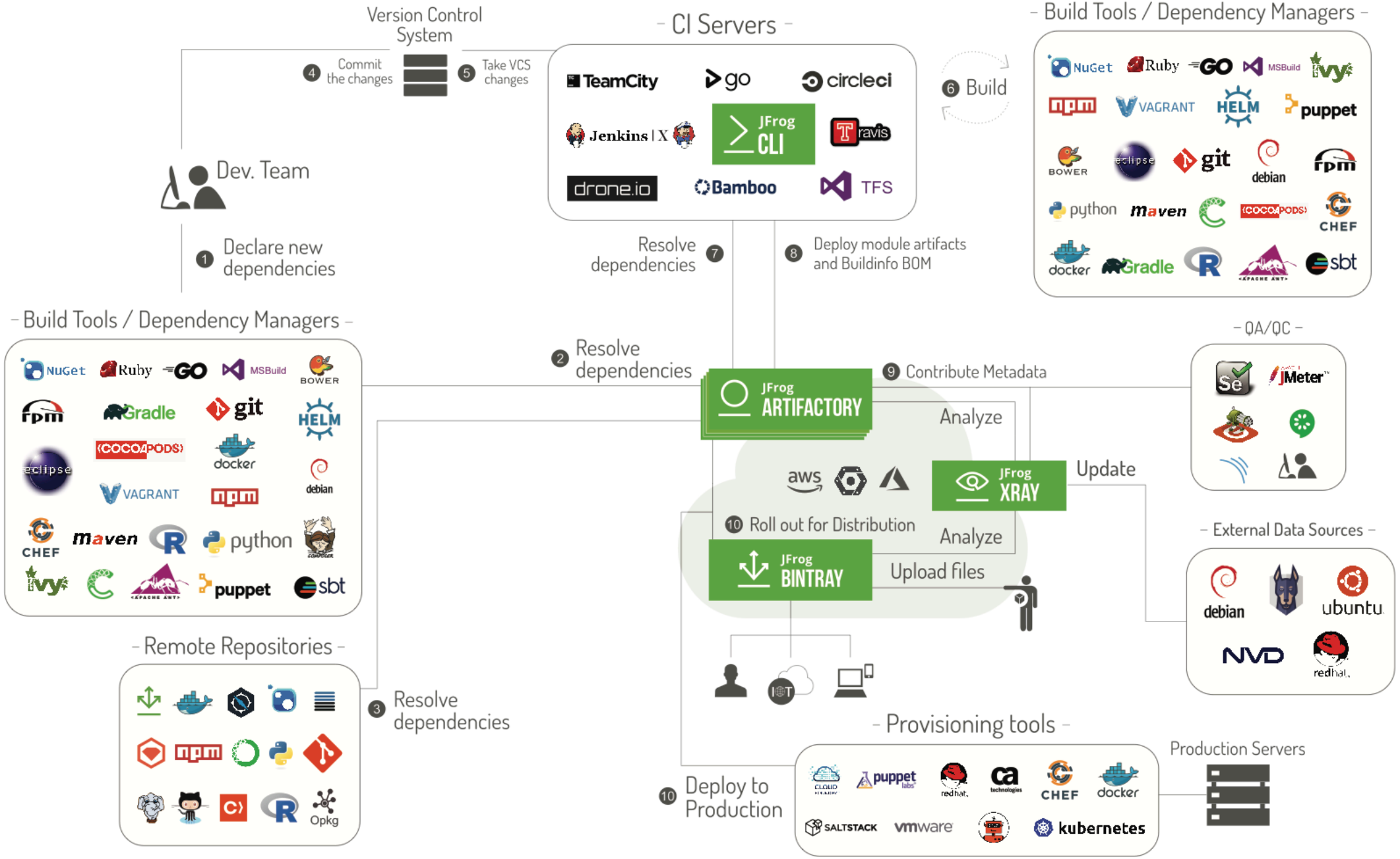

Version Control System
4 Commit the changes
5 Take VCS changes
- CI Servers -
TeamCity
go
circleci
Jenkins | X
JFrog CLI
Travis
drone.io
Bamboo
TFS
6 Build
- Build Tools / Dependency Managers -
NuGet
Ruby
GO
MSBuild
ivy
npm
VAGRANT
HELM
puppet
BOWER
eclipse
git
debian
rpm
python
maven
COCOAPODS
CHEF
docker
Gradle
R
Apache Ant
sbt
Dev. Team
1 Declare new dependencies
- Build Tools / Dependency Managers -
2 Resolve dependencies
JFrog ARTIFACTORY
Analyze
aws
JFrog XRAY
Update
- External Data Sources -
debian
ubuntu
NVD
redhat.
- Remote Repositories -
3 Resolve dependencies
Opkg

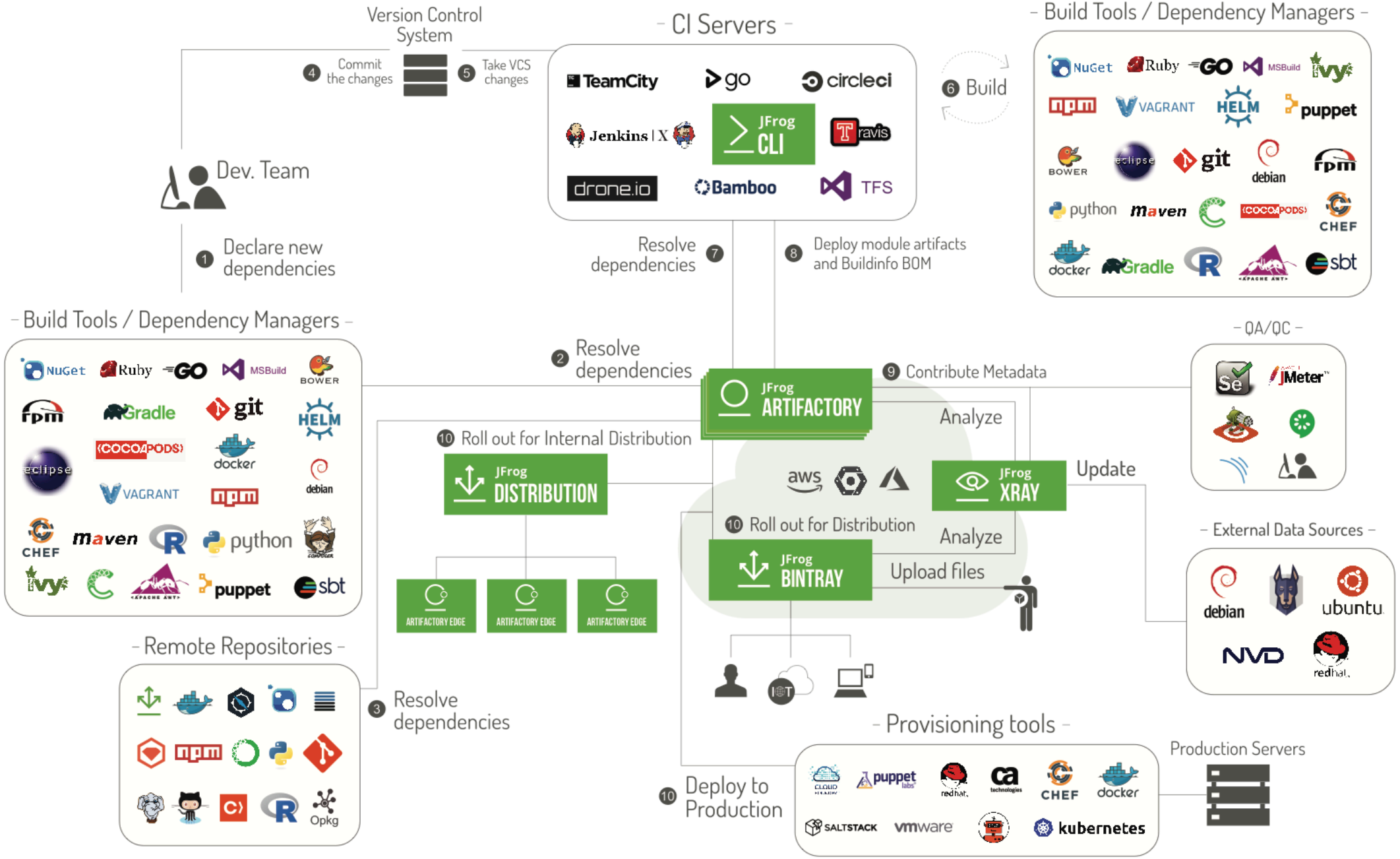

Version Control System
4 Commit the changes
5 Take VCS changes
- CI Servers -
TeamCity
go
circleci
Jenkins | X
JFrog CLI
Travis
drone.io
Bamboo
TFS
6 Build
- Build Tools / Dependency Managers -
NuGet
Ruby
GO
MSBuild
ivy
npm
VAGRANT
HELM
puppet
BOWER
eclipse
git
debian
rpm
python
maven
COCOAPODS
CHEF
docker
Gradle
R
Apache ANT
sbt
Dev. Team
1 Declare new dependencies
Resolve dependencies 7
- Build Tools / Dependency Managers -
2 Resolve dependencies
JFrog ARTIFACTORY
Analyze
aws
JFrog XRAY
Update
- External Data Sources -
debian
ubuntu
NVD
redhat.
- Remote Repositories -
3 Resolve dependencies
Opkg


Version Control System
4 Commit the changes
5 Take VCS changes
- CI Servers -
TeamCity
go
circleci
Jenkins | X
JFrog CLI
Travis
drone.io
Bamboo
TFS
6 Build
- Build Tools / Dependency Managers -
NuGet
Ruby
GO
MSBuild
ivy
npm
VAGRANT
HELM
puppet
BOWER
eclipse
git
debian
rpm
python
maven
COCOAPODS
CHEF
docker
Gradle
R
Apache Ant
sbt
Dev. Team
1 Declare new dependencies
Resolve dependencies 7
8 Deploy module artifacts and Buildinfo BOM
- Build Tools / Dependency Managers -
2 Resolve dependencies
JFrog ARTIFACTORY
Analyze
aws
JFrog XRAY
Update
- External Data Sources -
debian
ubuntu
NVD
redhat.
- Remote Repositories -
3 Resolve dependencies
Opkg

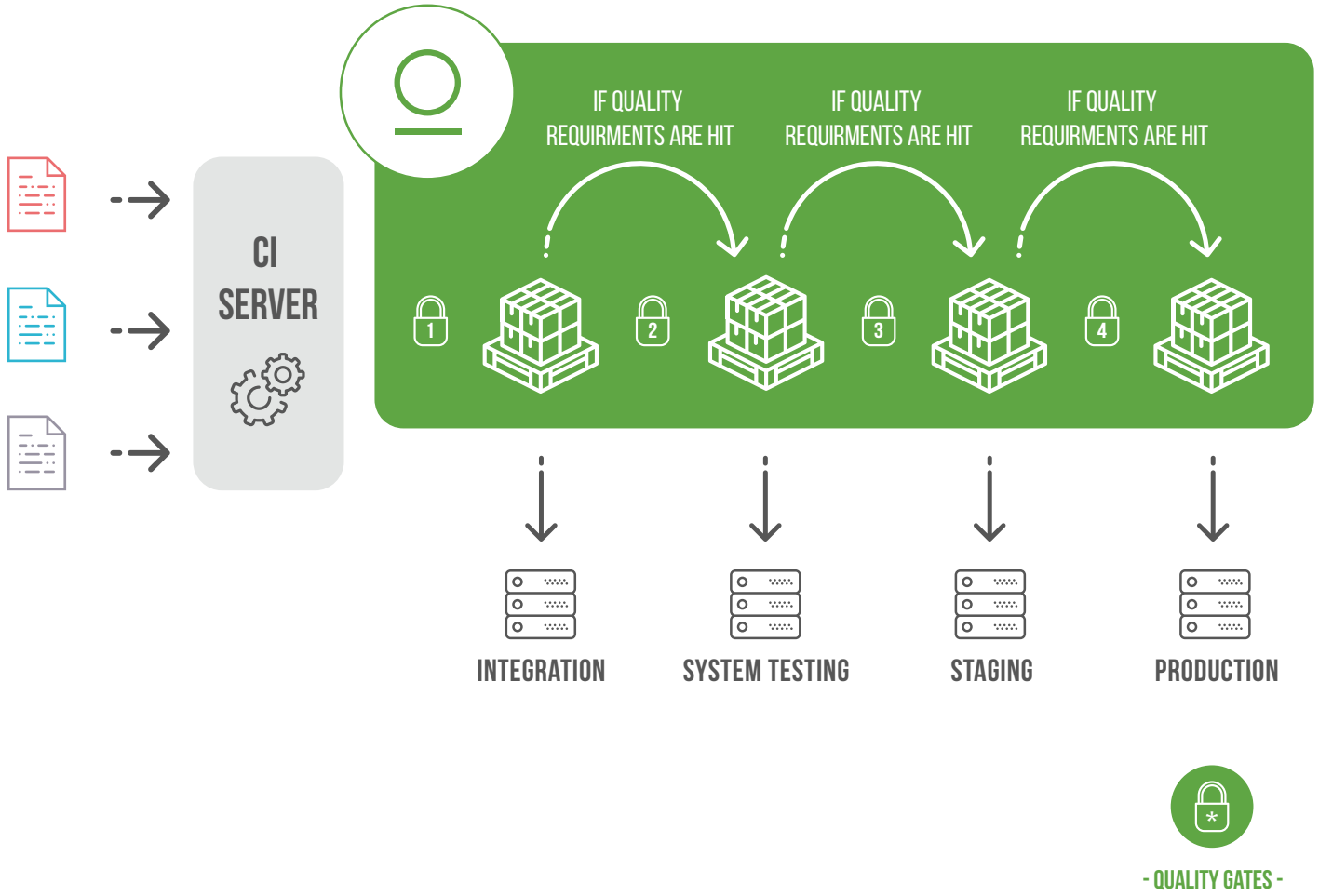

Version Control System
4 Commit the changes
5 Take VCS changes
- CI Servers -
TeamCity
go
circleci
Jenkins | X
JFrog CLI
Travis
drone.io
Bamboo
TFS
6 Build
- Build Tools / Dependency Managers -
Dev. Team
1 Declare new dependencies
Resolve dependencies 7
8 Deploy module artifacts and Buildinfo BOM
- Build Tools / Dependency Managers -
2 Resolve dependencies
JFrog ARTIFACTORY
9 Contribute Metadata
Analyze
JFrog XRAY
Update
- QA/QC -
- External Data Sources -
- Remote Repositories -
3 Resolve dependencies


Version Control System
4 Commit the changes
5 Take VCS changes
- CI Servers -
TeamCity
go
circleci
Jenkins | X
JFrog CLI
Travis
drone.io
Bamboo
TFS
6 Build
- Build Tools / Dependency Managers -
NuGet
Ruby
GO
MSBuild
ivy
npm
VAGRANT
HELM
puppet
BOWER
eclipse
git
debian
rpm
python
maven
COCOAPODS
CHEF
docker
Gradle
R
ANT
sbt
Dev. Team
1 Declare new dependencies
Resolve dependencies 7
8 Deploy module artifacts and Buildinfo BOM
- Build Tools / Dependency Managers -
2 Resolve dependencies
JFrog ARTIFACTORY
9 Contribute Metadata
Analyze
aws
JFrog XRAY
Update
- QA/QC -
Selenium
JMeter
- External Data Sources -
ubuntu
NVD
redhat
- Remote Repositories -
Opkg
3 Resolve dependencies
- Provisioning tools -
CLOUD FOUNDRY
SALTSTACK
vmware
kubernetes
10 Deploy to Production
Production Servers

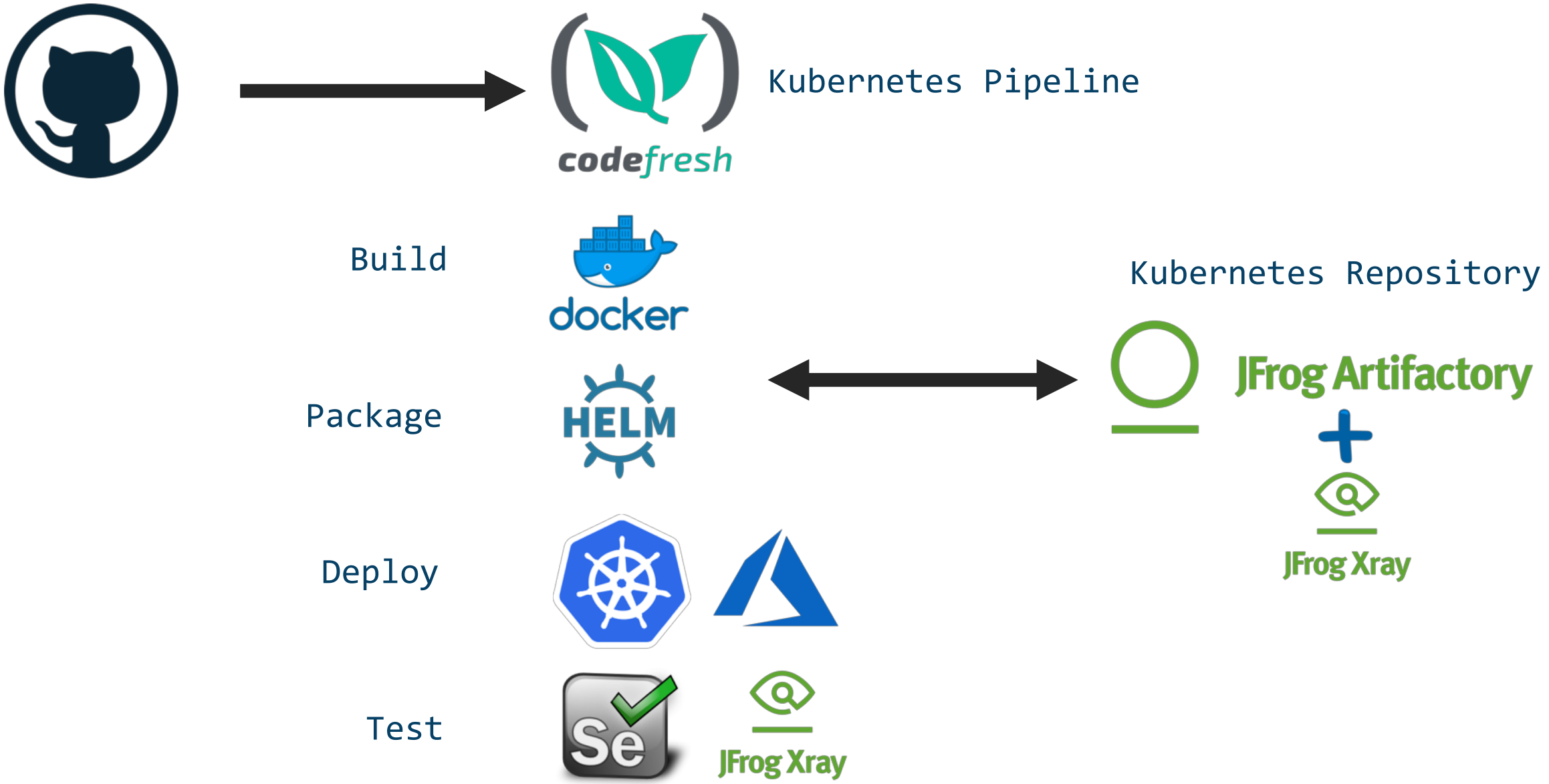

Version Control System
4 Commit the changes
5 Take VCS changes
- CI Servers -
TeamCity
go
circleci
Jenkins | X
JFrog CLI
Travis
drone.io
Bamboo
TFS
6 Build
- Build Tools / Dependency Managers -
NuGet
Ruby
GO
MSBuild
ivy
npm
VAGRANT
HELM
puppet
BOWER
eclipse
git
debian
rpm
python
maven
COCOAPODS
CHEF
docker
Gradle
R
ANT
sbt
Dev. Team
1 Declare new dependencies
Resolve dependencies 7
8 Deploy module artifacts and Buildinfo BOM
- Build Tools / Dependency Managers -
2 Resolve dependencies
JFrog ARTIFACTORY
9 Contribute Metadata
- QA/QC -
Selenium
JMeter
Analyze
aws
JFrog XRAY
Update
10 Roll out for Distribution
Analyze
JFrog BINTRAY
Upload files
- External Data Sources -
debian
ubuntu
NVD
redhat
- Remote Repositories -
Opkg
3 Resolve dependencies
- Provisioning tools -
puppet labs
redhat
ca technologies
CHEF
docker
SALTSTACK
vmware
kubernetes
10 Deploy to Production
Production Servers


Version Control System
4 Commit the changes
5 Take VCS changes
- CI Servers -
TeamCity
go
circleci
Jenkins | X
JFrog CLI
Travis
drone.io
Bamboo
TFS
6 Build
- Build Tools / Dependency Managers -
NuGet
Ruby
GO
MSBuild
ivy
npm
VAGRANT
HELM
puppet
BOWER
eclipse
git
debian
rpm
python
maven
COCOAPODS
CHEF
docker
Gradle
R
ANT
sbt
Dev. Team
1 Declare new dependencies
Resolve dependencies 7
8 Deploy module artifacts and Buildinfo BOM
- Build Tools / Dependency Managers -
2 Resolve dependencies
JFrog ARTIFACTORY
9 Contribute Metadata
- QA/QC -
Selenium
JMeter
Analyze
10 Roll out for Internal Distribution
JFrog DISTRIBUTION
aws
JFrog XRAY
Update
10 Roll out for Distribution
Analyze
JFrog BINTRAY
Upload files
ARTIFACTORY EDGE
ARTIFACTORY EDGE
ARTIFACTORY EDGE
- External Data Sources -
debian
ubuntu
NVD
redhat
- Remote Repositories -
npm
Opkg
3 Resolve dependencies
- Provisioning tools -
puppet labs
redhat
ca technologies
CHEF
docker
SALTSTACK
vmware
kubernetes
10 Deploy to Production
Production Servers

Promotion pipeline



What did we just do?



Container Orchestration: Kubernetes

@jldeen @jbaruch #MSIgnite http://jfrog.com/shownotes

What is Kubernetes?

Open source container orchestrator that automates deployment, scaling, and management of applications.

Features include:

- **Automatic bin packing**
- **Self-healing**
- **Horizontal scaling**
- **Service discovery**
- **Load balancing**
- **Automated rollouts and rollbacks**
- **Secret and configuration management**

- **Designed by Google**
  - **Based on their system used to run BILLIONS of containers per week**
- **Over 2,300 contributors**
- **Graduated from CNCF**

Who is using Kubernetes?


Bloomberg
box
CapitalOne
COMCAST
DENSO
ebay
GitHub
cruise
Goldman Sachs
indeed
intuit
Kuelap
Layer
Morgan Stanley
NAIC
National Association of Insurance Commissioners
Nasdaq
NCSOFT
NIPR
NATIONAL INSURANCE PRODUCER REGISTRY
Olark
Pinterest
PUSHER
reddit
ricardo.ch
salesforce
SAP Concur
shopify
showmax
Spotify
Spredfast
STEELHOUSE
textkernel
The New York Times
ticketmaster
twilio
twitter
vevo
WIKIMEDIA FOUNDATION
woorank
WP engine
zalando
zendesk
JFrog


Walmart
CONCUR
viacom
buffer
Home Office
Arkena
Goldman Sachs
monzo
EVE ONLINE
Pearson
The New York Times
box
OpenAI
ticketmaster
SKY
COMCAST
Bloomberg
Pokémon GO
SOUNDCLOUD
unacast.

Azure Kubernetes Service (AKS)

@jldeen @jbaruch #MSIgnite http://jfrog.com/shownotes

Your Kubernetes Cluster Managed by Azure

@jldeen @jbaruch #MSIgnite http://jfrog.com/shownotes

Why AKS?

Easy to use:

- **Fastest path to Kubernetes on Azure**
- **Up and running with 3 simple commands**
- **I argue there are 2.5 commands**

Easy to manage:

- **Automated upgrades and patching**
- **Easily scale the cluster up and down**
- **Self-healing control plane**

Uses open APIs – 100% upstream Kubernetes

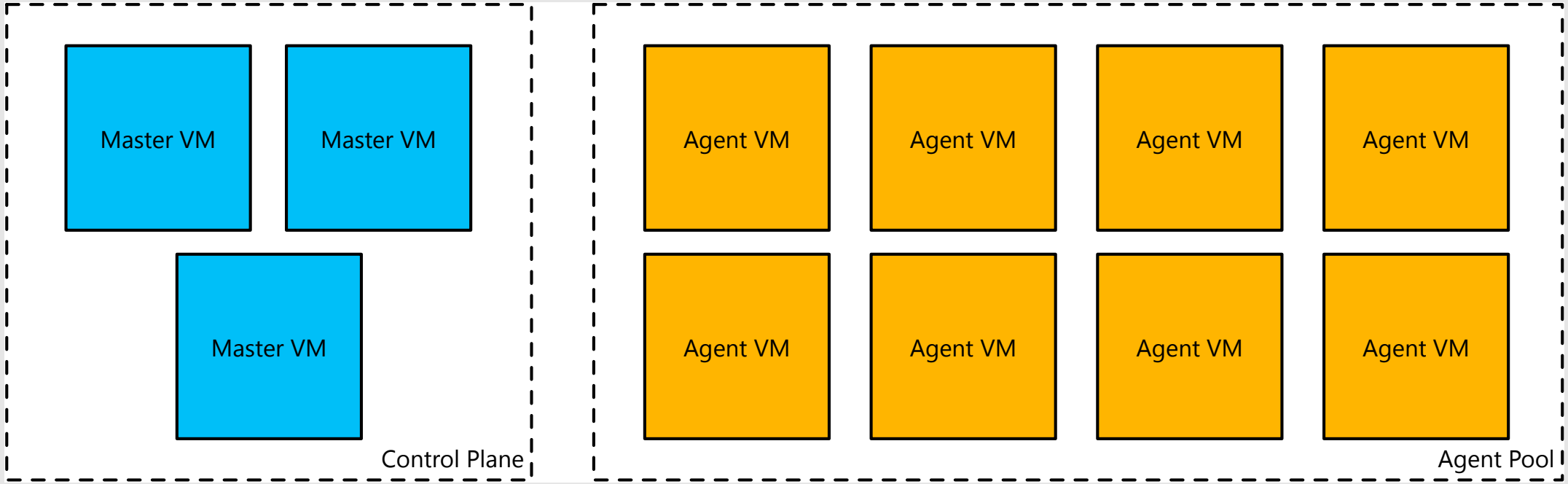
Getting Started with AKS

```
$ az aks create -g myResourceGroup -n myCluster --generate-ssh-keys
\ Running ..

$ az aks install-cli
Downloading client to /usr/local/bin/kubectl ..

$ az aks get-credentials -g myResourceGroup -n myCluster
Merged "myCluster" as current context ..

$ kubectl get nodes
NAME                       STATUS    AGE       VERSION
aks-mycluster-36851231-0   Ready     4m        v1.8.1
aks-mycluster-36851231-1   Ready     4m        v1.8.1
aks-mycluster-36851231-2   Ready     4m        v1.8.1
```

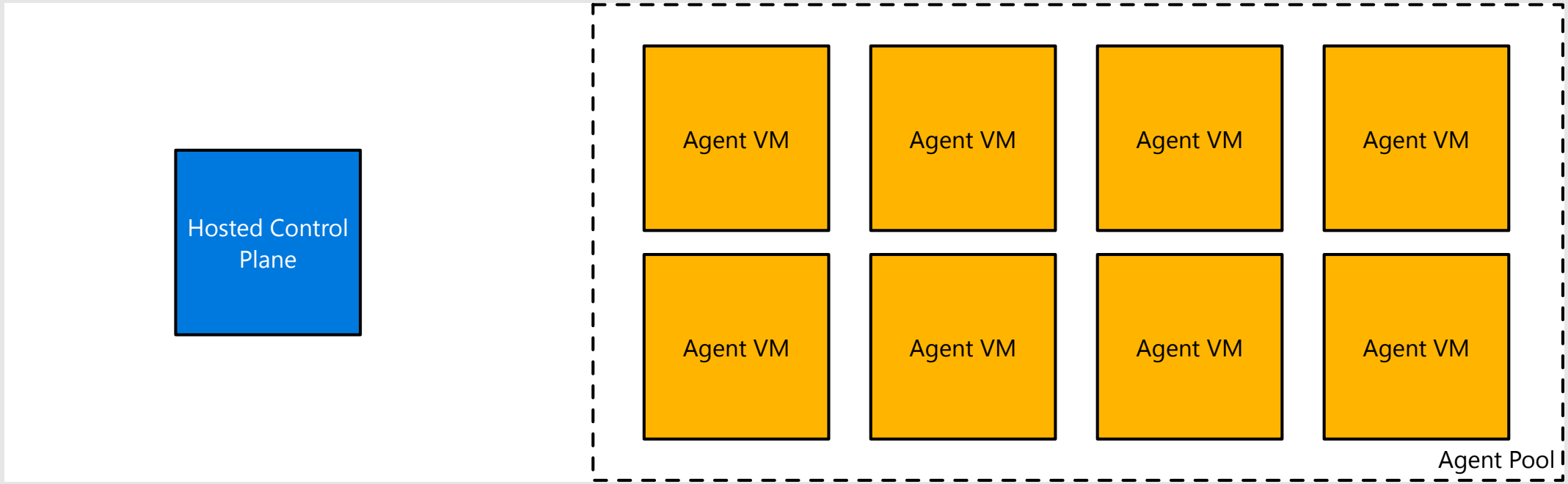
Managing an AKS Cluster

```
$ az aks list –o table
Name                 Location    ResourceGroup     KubernetesRelease
ProvisioningState
-----------------  ----------  -------------  -------------------  -------------
------
myCluster            westus2     myResourceGroup                1.7.7  Succeeded

$ az aks upgrade -g myResourceGroup -n myCluster –-kubernetes-version 1.8.1
\ Running ..

$ kubectl get nodes
NAME                          STATUS    AGE       VERSION
aks-mycluster-36851231-0   Ready     12m       v1.8.1
aks-mycluster-36851231-1   Ready     8m        v1.8.1
aks-mycluster-36851231-2   Ready     3m        v1.8.1

$ az aks scale -g myResourceGroup -n myCluster --agent-count 10
\ Running ..
```

Kubernetes without AKS



Kubernetes with AKS



Azure Container Service (AKS)

Azure Container Instances (ACI)

Azure Container Registry

Open Service Broker API (OSBA)

Release Automation Tools

Release automation tools

Simplifying the Kubernetes experience

Streamlined Kubernetes development

The package manager for Kubernetes

Event-driven scripting for Kubernetes

Visualization dashboard for Brigade

Azure Container Service (AKS)

Azure Container Instances (ACI)

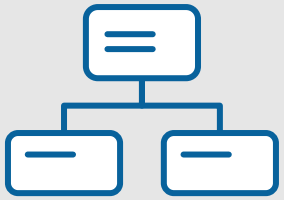
Azure Container Registry

Open Service Broker API (OSBA)

Release Automation Tools

Helm

The best way to find, share, and use software built for Kubernetes



Manage complexity

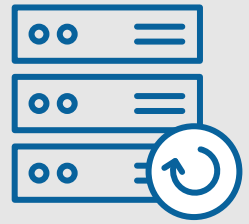
Charts can describe complex apps; provide repeatable app installs, and serve as a single point of authority

Easy updates

Take the pain out of updates with in-place upgrades and custom hooks

Simple sharing

Charts are easy to version, share, and host on public or private servers



Rollbacks

Use `helm rollback` to roll back to an older version of a release with ease

@jldeen @jbaruch #MSIgnite http://jfrog.com/shownotes

Azure Container Service (AKS)

Azure Container Instances (ACI)

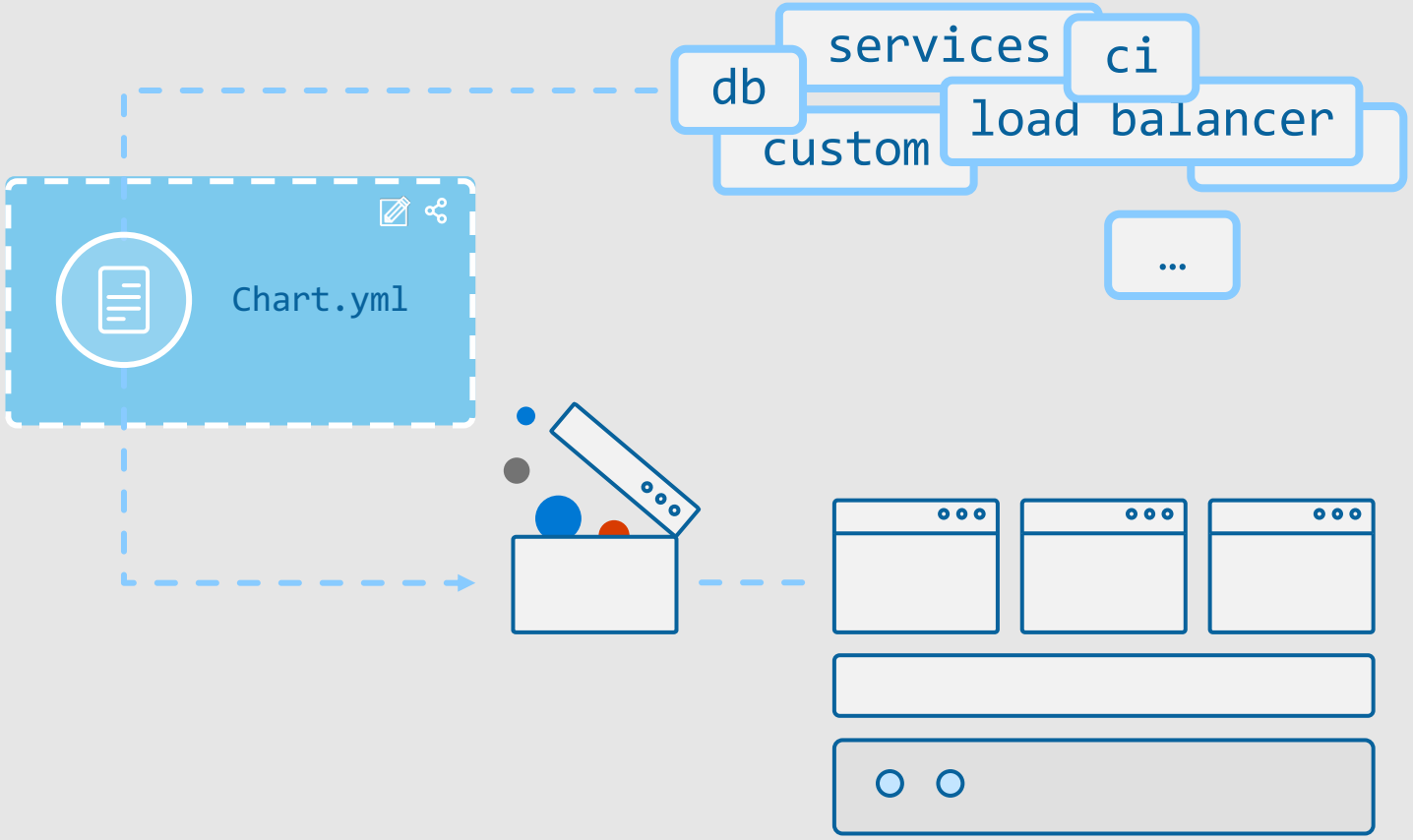
Azure Container Registry

Open Service Broker API (OSBA)

Release Automation Tools

Helm

Helm Charts helps you define, install, and upgrade even the most complex Kubernetes application



Azure Container Service (AKS)

Azure Container Instances (ACI)

Azure Container Registry

Open Service Broker API (OSBA)

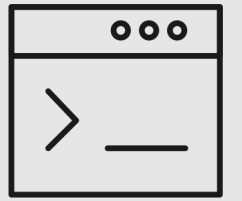
Release Automation Tools

Draft

Simple app development and deployment – into any Kubernetes cluster

Simplified development

Using two simple commands, developers can now begin hacking on container-based applications without requiring Docker or even installing Kubernetes themselves



Language support

Draft detects which language your app is written in, and then uses packs to generate a Dockerfile and Helm Chart with the best practices for that language

@jldeen @jbaruch #MSIgnite http://jfrog.com/shownotes

Demo

5 Kubernetes Best Practices

- Build small containers
- Application architecture
  - Use Namespaces
  - Use helm charts
  - RBAC
- Implement Health checks
- Set requests and limits
- Be mindful of your services
  - Map external services
  - Don’t rely on load balancers

Q&A and Shownotes

- @jldeen
- @jbaruch
- #MSIgnite
- http://jfrog.com/shownotes
  - Slides
  - Video
  - All the links
  - Comments, Ratings
  - Raffle