ThoughtWorks®

DOCKER'S NOT JUST FOR PRODUCTION

Using containers for your development environment

Charles Korn

Development environment?

THE PROBLEM

How to ensure your sanity?

#1:

Consistent builds

#2:
Fast builds

#3:

Easy to set up

#4:
Easy to use

#5:

Easy to establish and maintain

#6:
Make efficient use of resources

#7:

Easy to reproduce old builds

#8:
Enable team autonomy

EIGHT CRITERIA FOR SANITY

1. Consistent builds
2. Fast builds
3. Easy to set up
4. Easy to use
5. Easy to establish and maintain
6. Make efficient use of resources
7. Easy to reproduce old builds
8. Enable team autonomy

Solution #1:

INSTALL EVERYTHING LOCALLY

Image source: quickmeme.com

It works

Good local development experience

Possible to script setup / version check

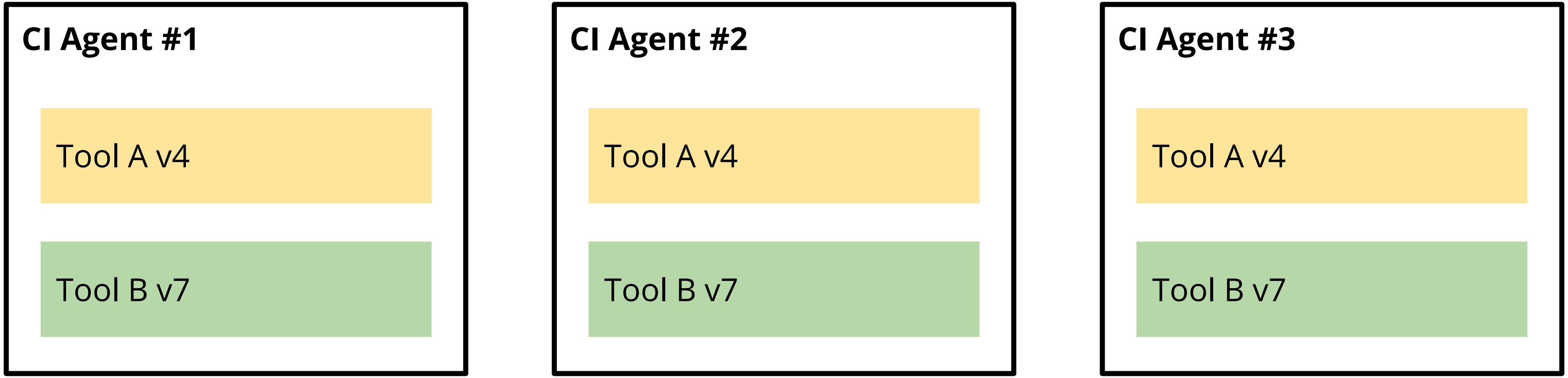
Possible to script setup / version check

…but it usually isn’t

What happens if you’re working on multiple codebases at the same time?

Image source: shauntmax30.com

Shared CI infrastructure == :(



CI Agent #1

Tool A v4

Tool B v7

CI Agent #2

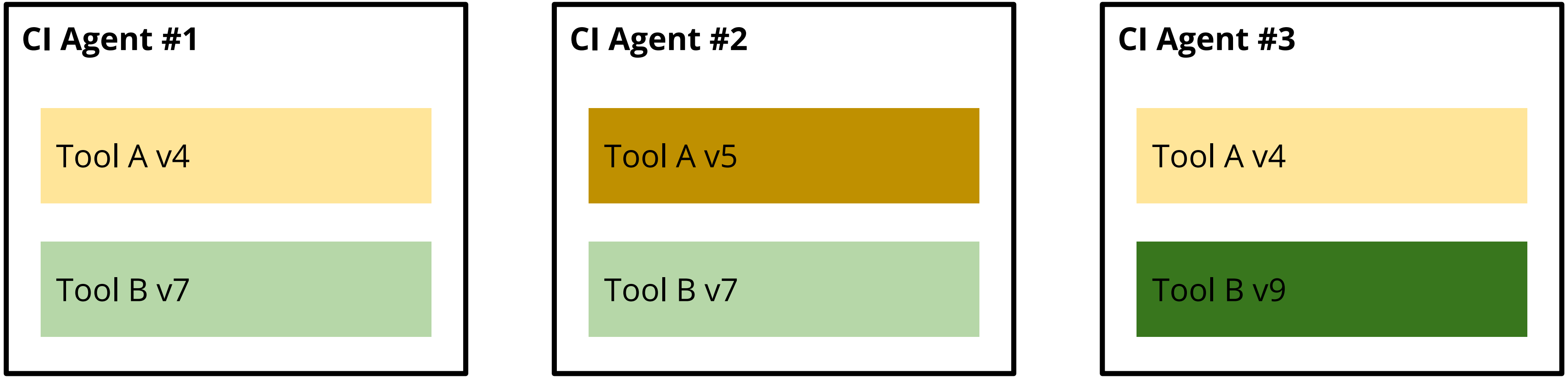
Tool A v4

Tool B v7

CI Agent #3

Tool A v4

Tool B v7



CI Agent #1

Tool A v4

Tool B v7

CI Agent #2

Tool A v5

Tool B v7

CI Agent #3

Tool A v4

Tool B v9

Consistent builds	☆
Fast builds	☆☆☆☆☆
Easy to set up	😭 to ☆☆☆
Easy to use	☆☆☆☆☆
Easy to establish and maintain	😭 to ☆☆☆
Make efficient use of resources	☆☆☆
Easy to reproduce old builds	😭 to ☆☆
Enable team autonomy	☆☆

Solution #2:

DEVELOPMENT VM

VAGRANT

```
vagrant up
vagrant provision
vagrant destroy
```

Automation

Isolation

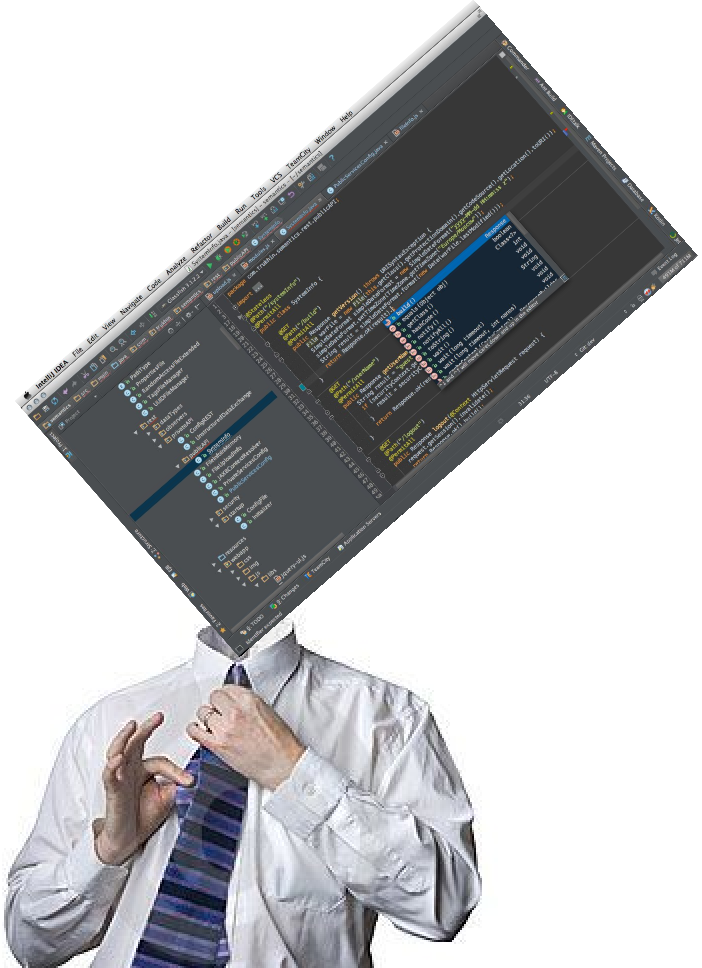
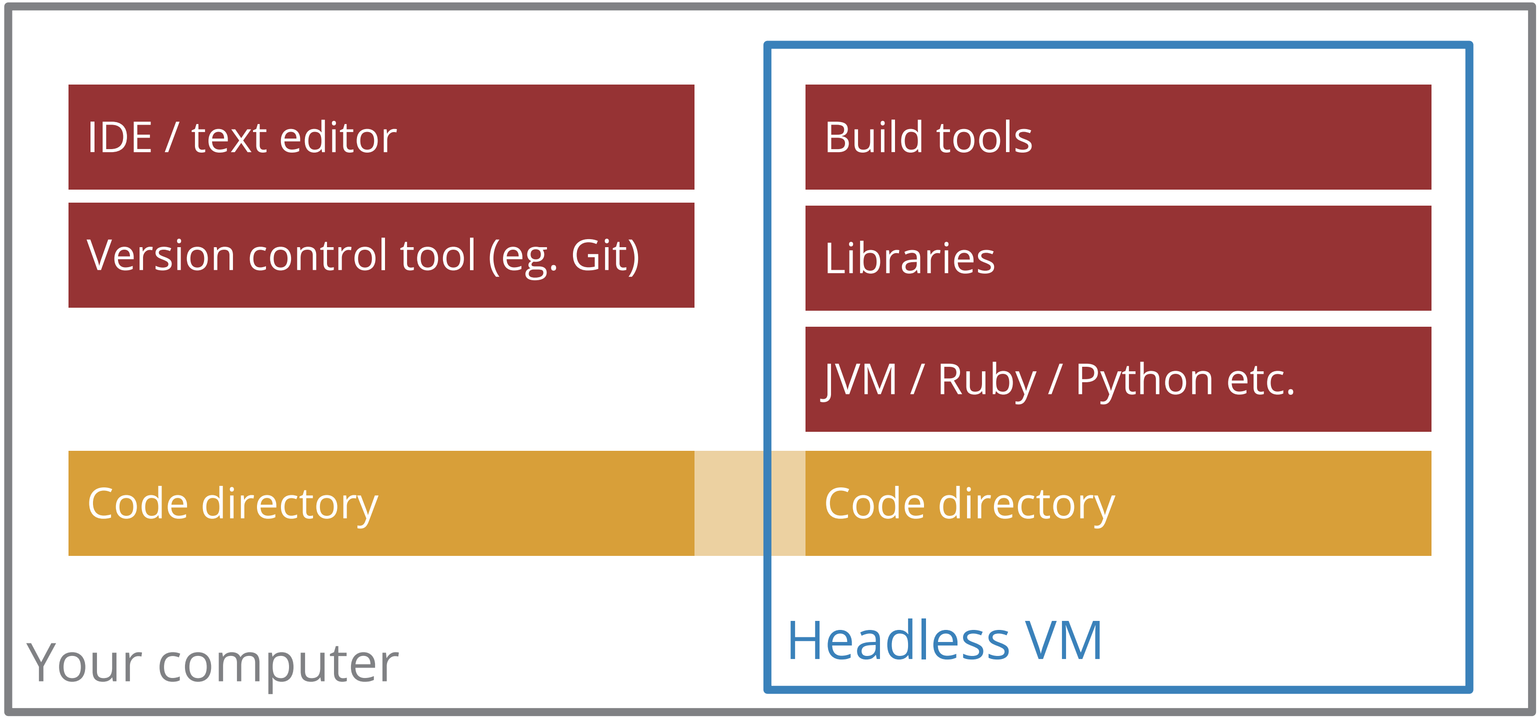


Image sources: dreamstime.com, mikebeas.com, wikimedia.org



IDE / text editor

Version control tool (eg. Git)

Code directory

Your computer

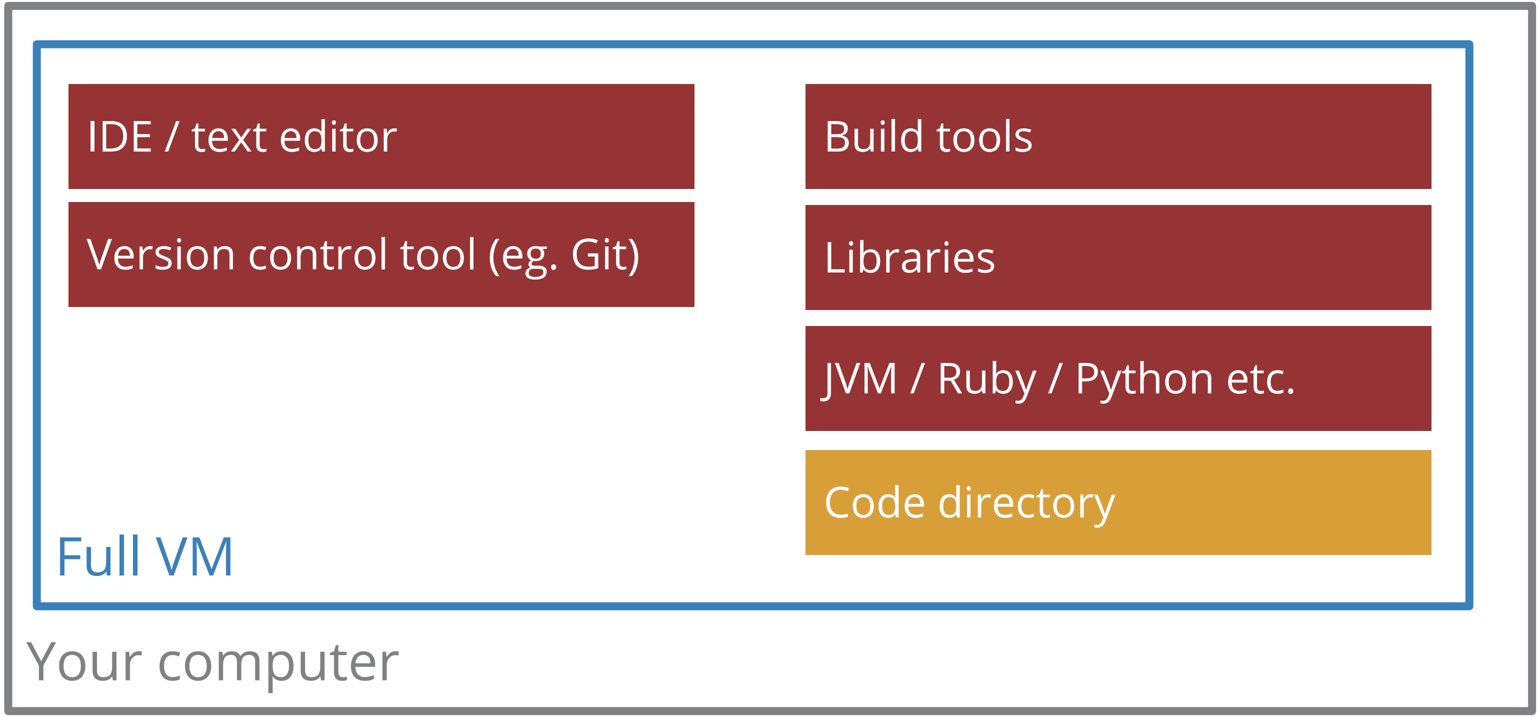
Build tools

Libraries

JVM / Ruby / Python etc.

Code directory

Headless VM



IDE / text editor

Version control tool (eg. Git)

Build tools

Libraries

JVM / Ruby / Python etc.

Code directory

Full VM

Your computer

Setup time

Resource-hungry

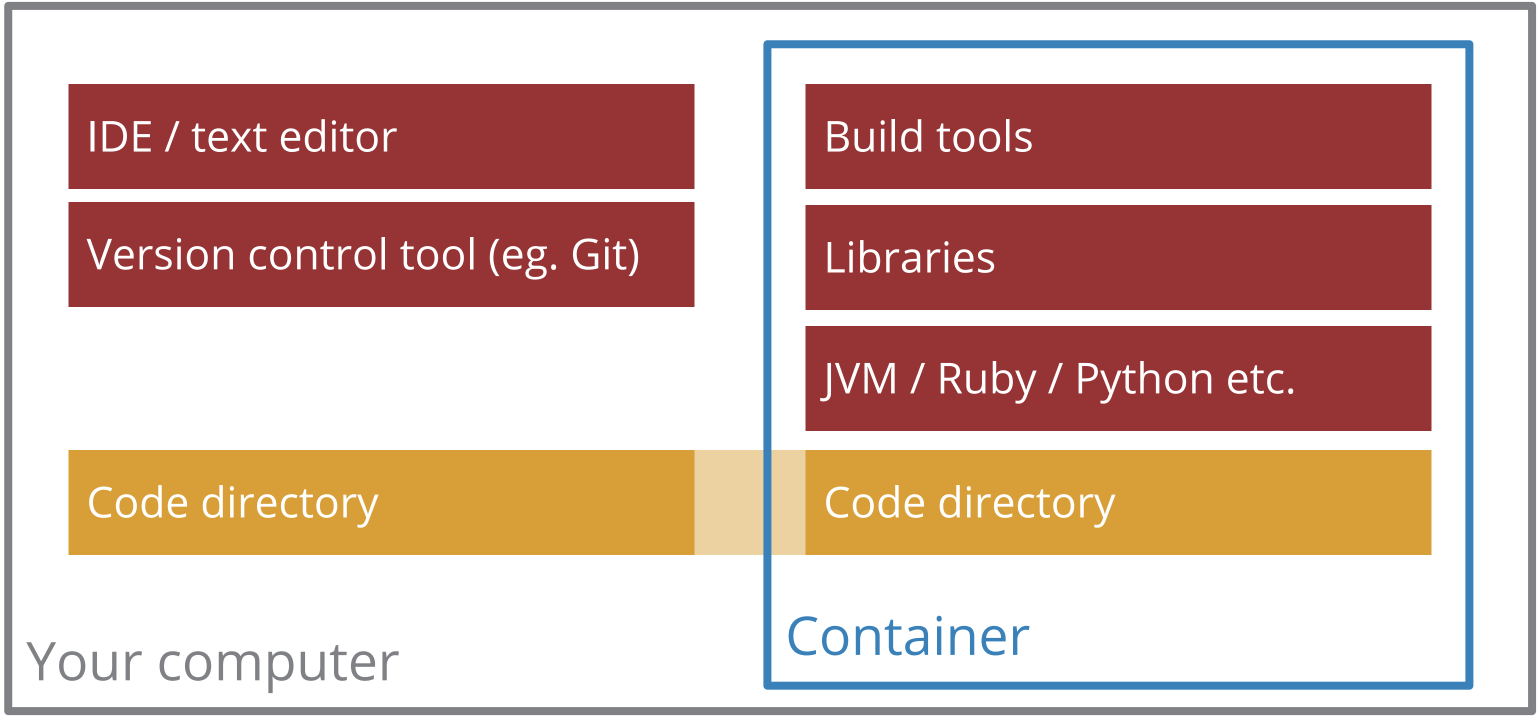
Configuration drift

Consistent builds	☆☆☆
Fast builds	☆☆☆☆☆
Easy to set up	☆☆☆☆☆
Easy to use	☆☆☆(☆)
Easy to establish and maintain	☆☆☆☆☆
Make efficient use of resources	☆☆☆
Easy to reproduce old builds	☆☆☆☆☆
Enable team autonomy	☆☆☆

Solution #3:

CONTAINERISED ENVIRONMENT

Image source: memecaptain.com



IDE / text editor

Version control tool (eg. Git)

Code directory

Your computer

Build tools

Libraries

JVM / Ruby / Python etc.

Code directory

Container

Lightweight

Ephemeral

…which makes caching impossible

Quick setup

Easy to update

```
docker build \
   --tag cool-app-dev-env:v3 \
   dev-env

docker run --rm -it -v $PWD:/code -w /code \
   cool-app-dev-env:v3 \
   ./gradlew build
```

```
FROM ubuntu:16.10

RUN apt-get update
RUN apt-get install -y openjdk-8-jdk
```

```
FROM ubuntu:16.10

RUN apt-get update && apt-get install -y \
   openjdk-8-jdk=8u111-b14-2ubuntu0.16.10.2
```

Don’t need to use Docker in production

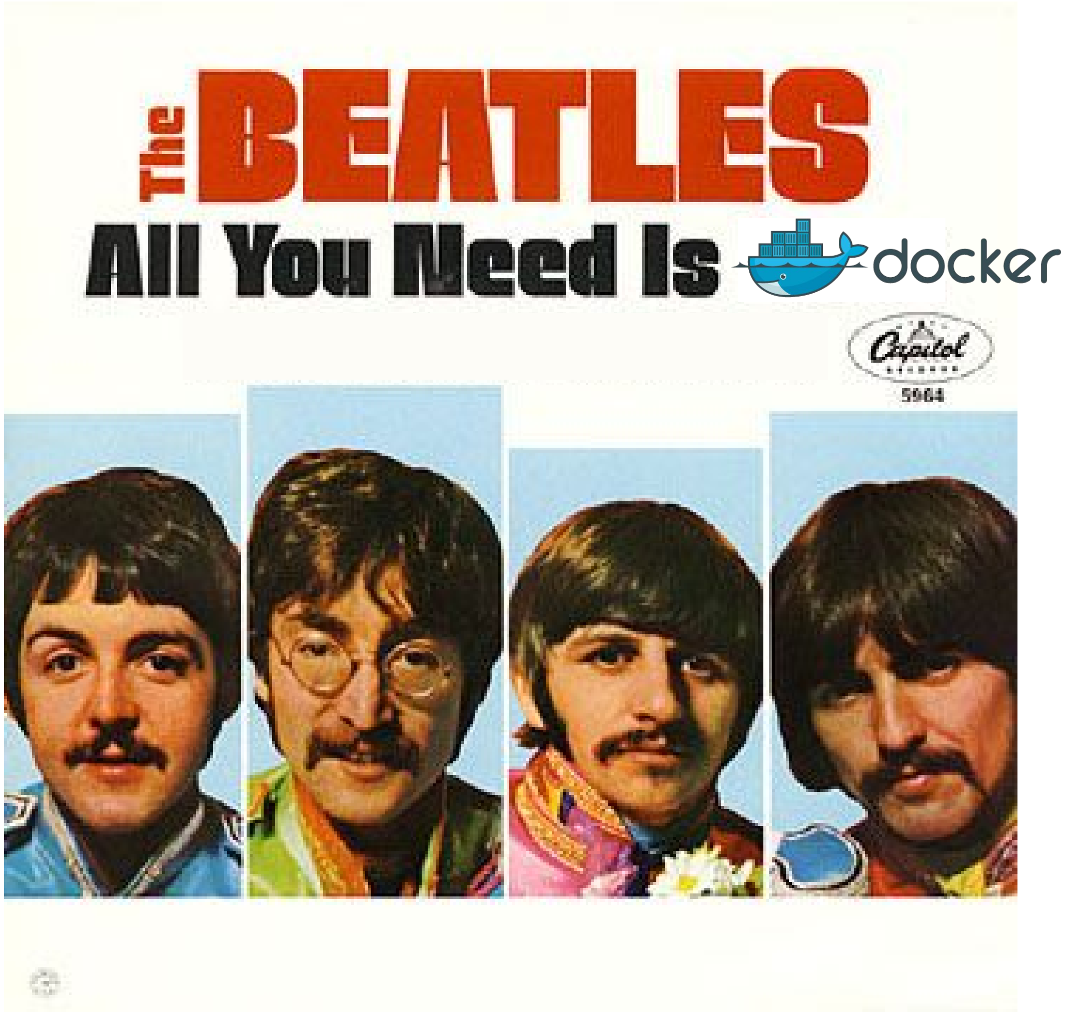


Image sources: wikimedia.org, docker.com *but…*

DEMO

Consistent builds	☆☆☆☆☆
Fast builds	☆☆☆☆☆
Easy to set up	☆☆☆☆☆
Easy to use	☆☆☆
Easy to establish and maintain	☆☆☆☆☆
Make efficient use of resources	☆☆☆☆☆
Easy to reproduce old builds	☆☆☆☆☆
Enable team autonomy	☆☆☆☆☆

THANK YOU

Charles Korn

ckorn@thoughtworks.com

charleskorn.com

ThoughtWorks®