

Session 01

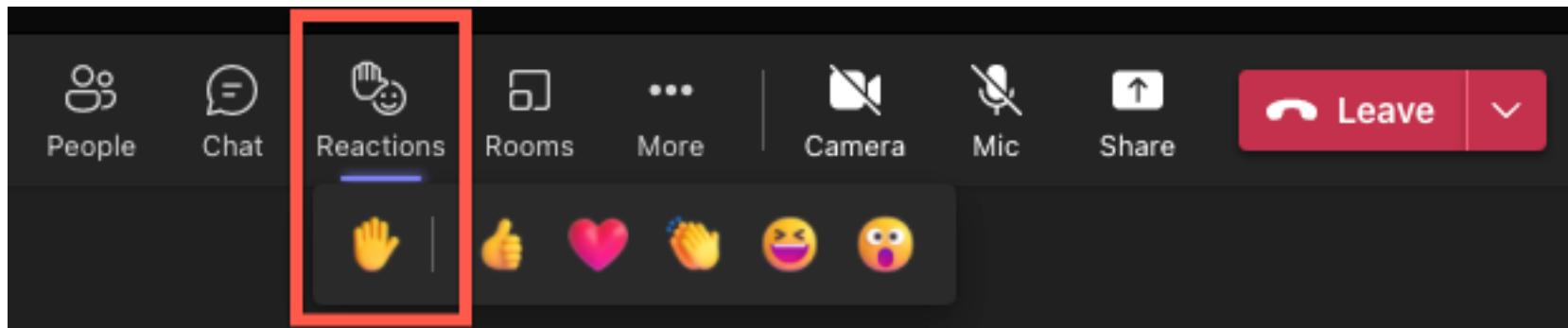
Course introduction

Anton Boyko
Microsoft Regional Director
Microsoft Azure MVP
me@boykoant.pro



Housekeeping

- Please keep yourself muted unless you are participating in the conversation, so we can have a more clear recording.
- If you have any questions, comments or want to discuss something – raise your hand.



Course intro

About me

18 years in IT

Worked as Developer, DevOps, Manager, Architect

Working with cloud since 2009

Working with Microsoft Azure since 2011

Microsoft Azure MVP since 2014

Microsoft Regional Director since 2019

Live long and prosper 

About course

- DevOps and DevOps practices
- DevOps culture
- CI/CD, IaC, Configuration management, etc
- Virtualization and containerization, Docker
- Microsoft Azure public cloud
 - IaaS, PaaS, SaaS
 - Cloud architecture pillars
 - Managed services
- Performance testing and chaos engineering
- Keeping quality of your setup under control

Course goal

Teach you how to do DevOps in Azure?



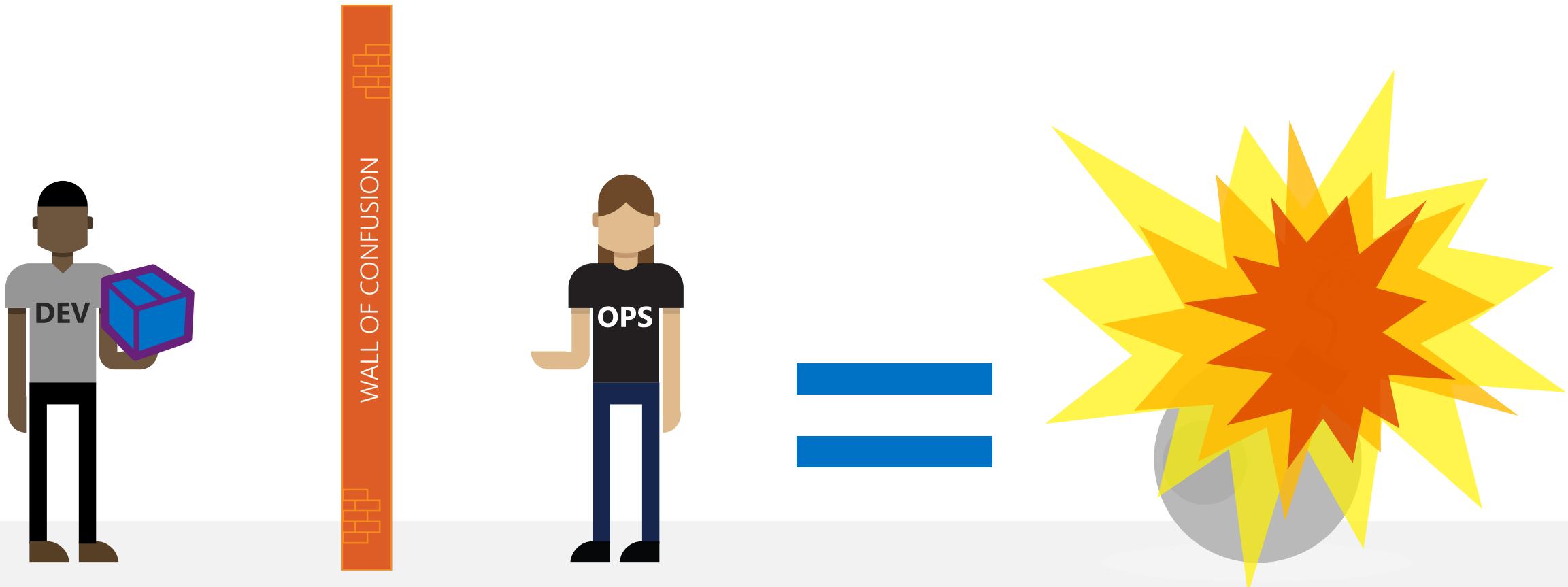
Yes / No



Teach you how you can teach yourself.

What is DevOps in general?

Traditional Development and Operations



“DevOps is
development
and operations
collaboration”

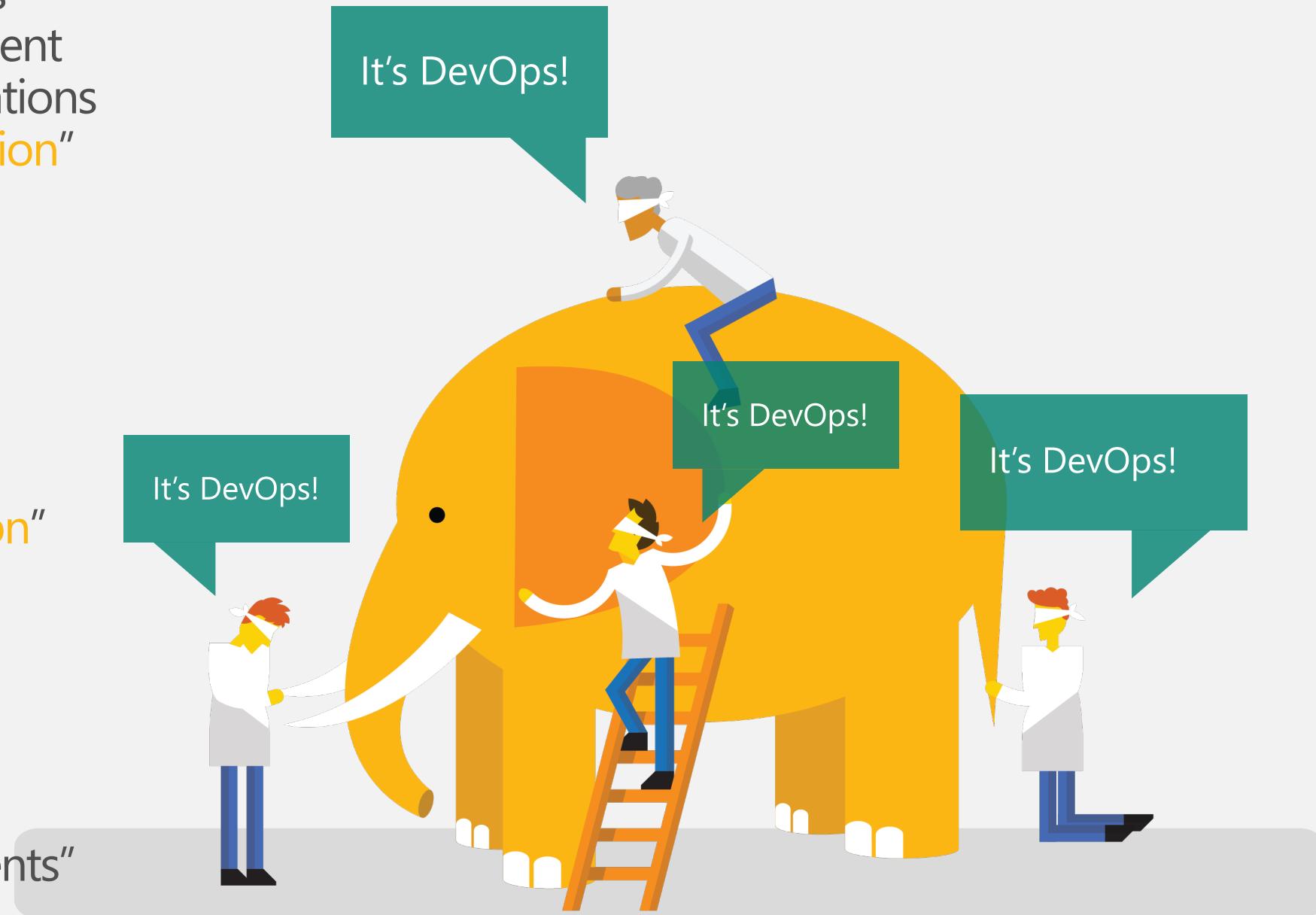
“DevOps
is using
automation”

“DevOps
is **small**
deployments”

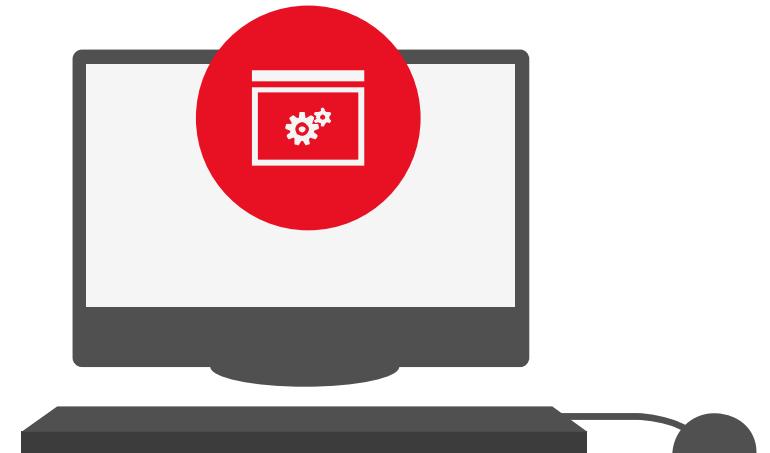
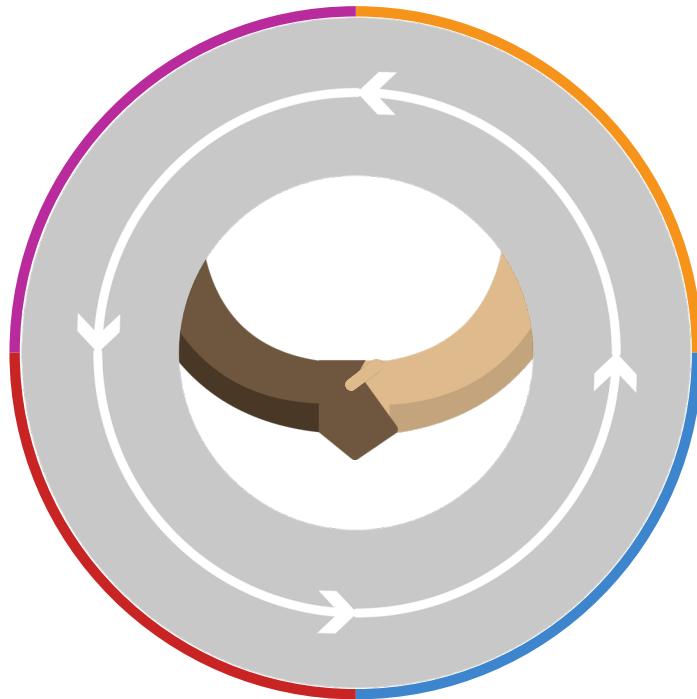
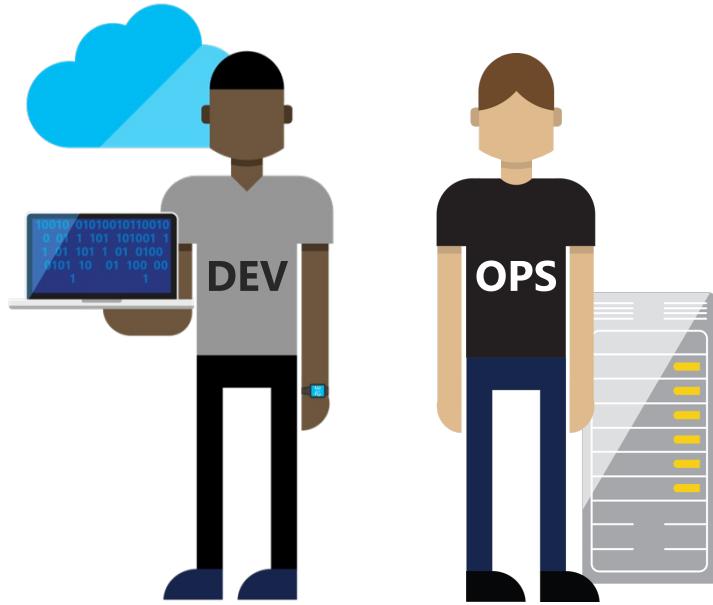
“DevOps is
treating your
infrastructure
as code”

“DevOps
is feature
switches”

“Kanban
for Ops?”



DevOps: the three stage conversation



1 | People

2 | Process

3 | Products

"I'm allergic to
manual labor"



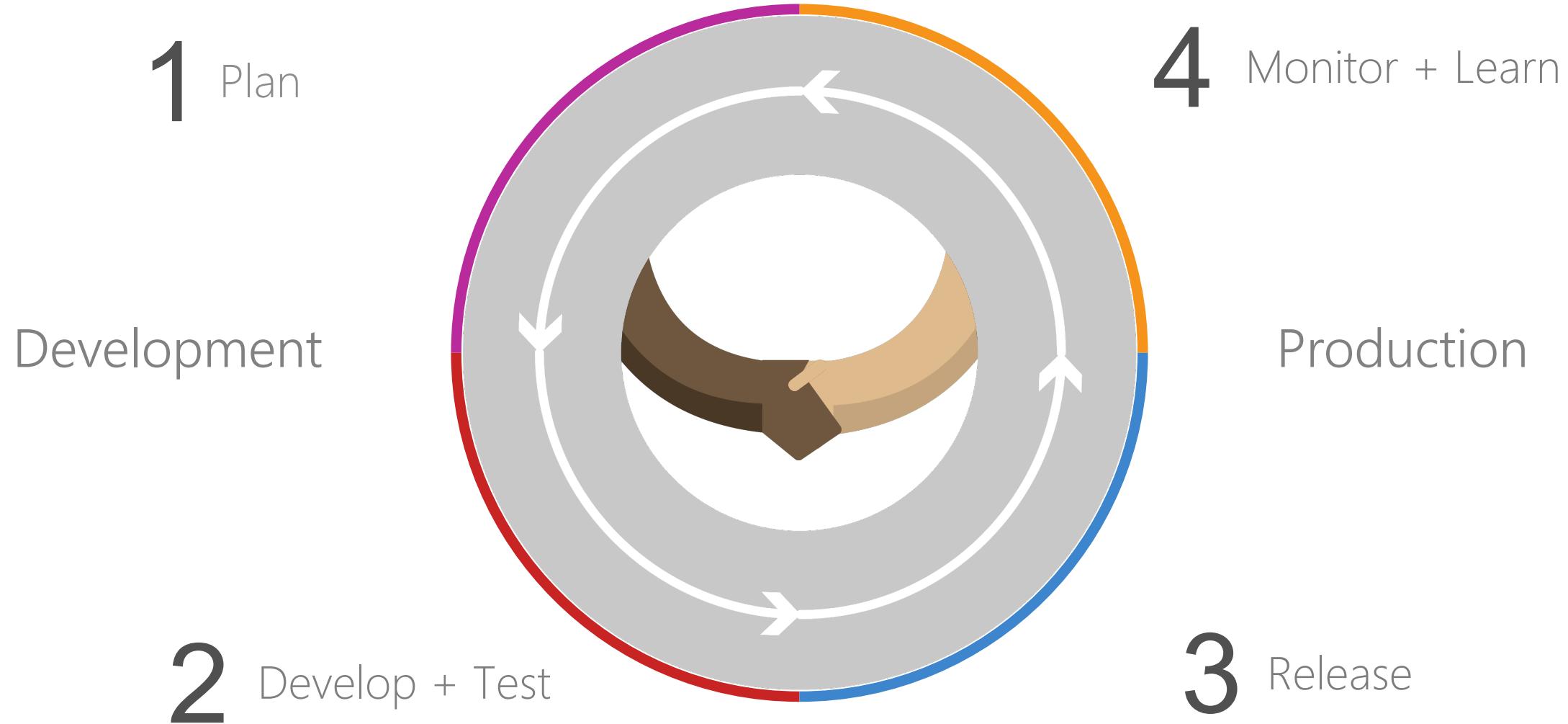
"If you know your server's name, it's your pet. But we are dealing with cattle."



List of DevOps Practices

- Infrastructure as Code (IaC)
- Continuous Integration
- Automated Testing
- Continuous Deployment
- Release Management
- App Performance Monitoring
- Load Testing & Auto-Scale
- Availability Monitoring
- Change/Configuration Management
- Feature Flags
- Automated Environment De-Provisioning
- Self Service Environments
- Automated Recovery (Rollback & Roll-Forward)
- Hypothesis Driven Development
 - Testing in Production
 - Fault Injection
 - Usage Monitoring/User Telemetry

DevOps

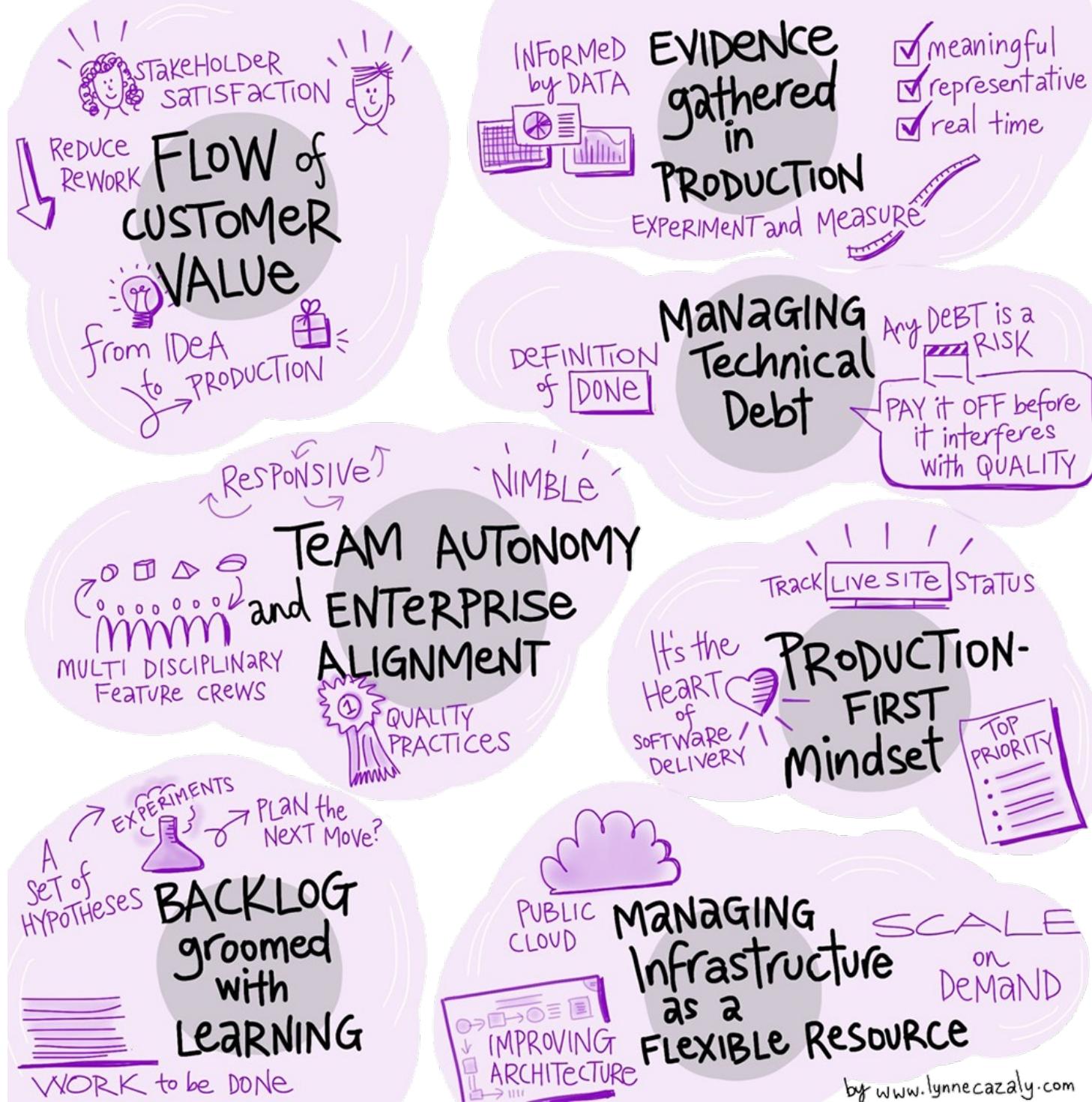


DevOps Maturity Assessment

OUR 7 HABITS FOR DEVOPS SUCCESS

Based on our experiences from our own transformation to a DevOps culture, we have identified **DevOps habits** and related **DevOps practices** that will lead to a true shift to DevOps in organizations of any size.

<http://DevOpsAssessment.azurewebsites.net/>



Source control

GitOps

A single answer for multiple questions.

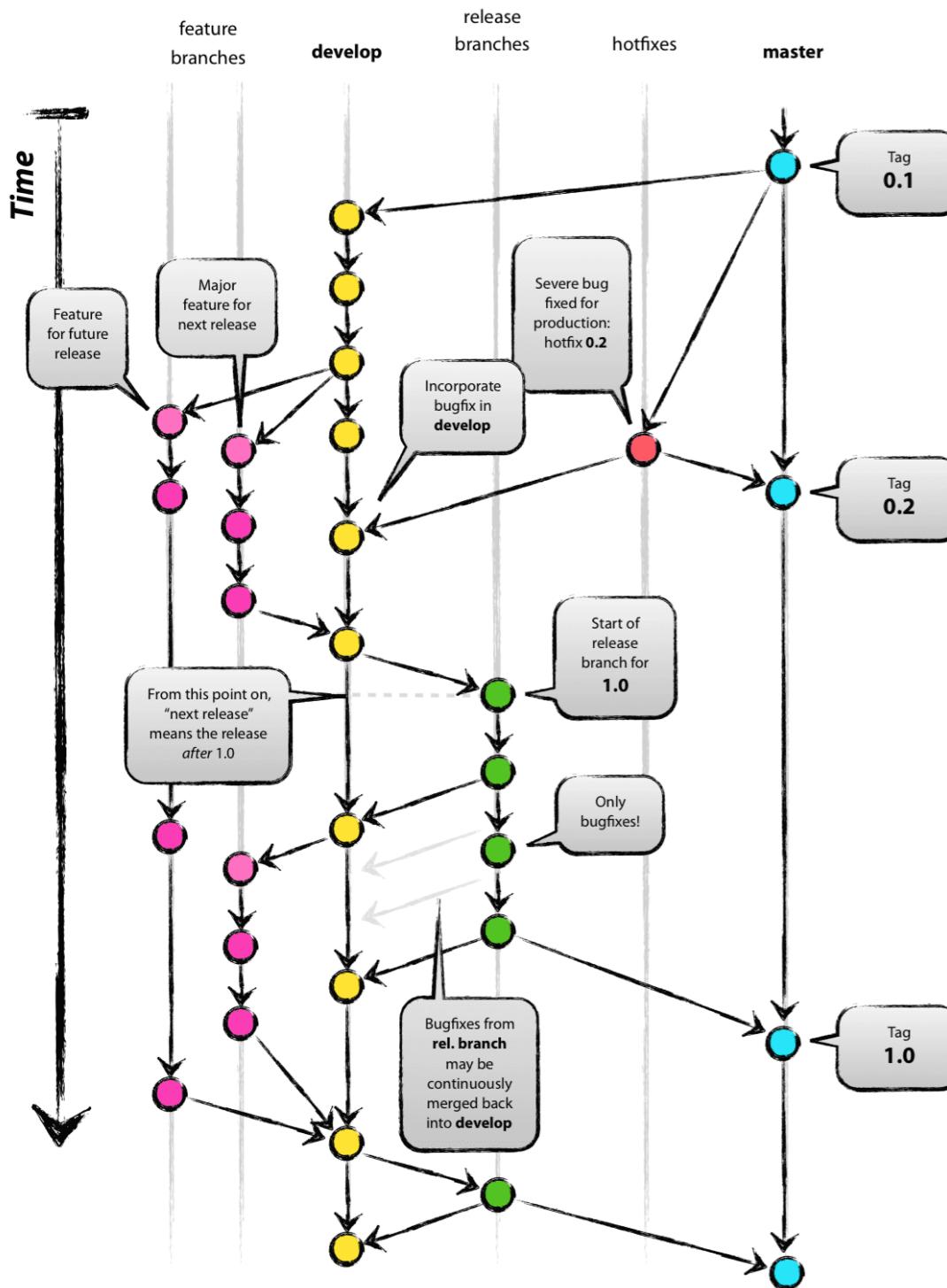
- How many environments do we have?
- How are our environments configured?
- What software is installed on each environment?
- What was the reason to introduce this or that change to the environment?
- How to ensure that all our environments are consistent?

Q: Who likes to write documentation?

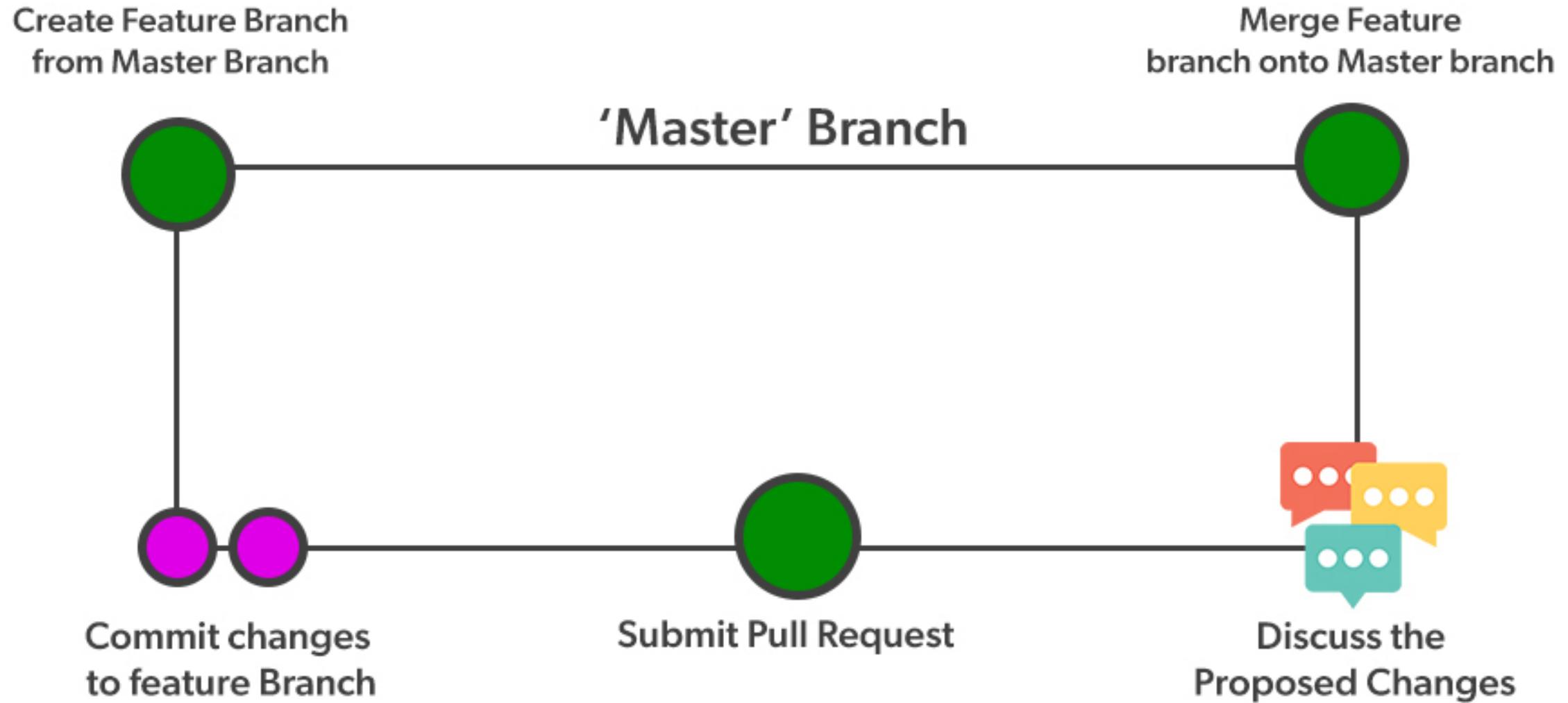
Q: Who likes to read documentation?



Git flow



GitHub flow



Q: Which one is better?

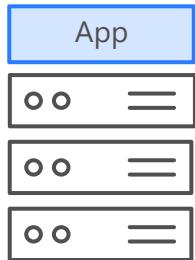
A: It depends.



DEM

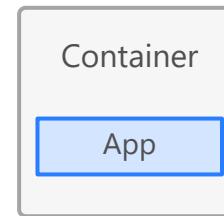
Virtualization vs containerization

What is a **container**?



Virtual machines

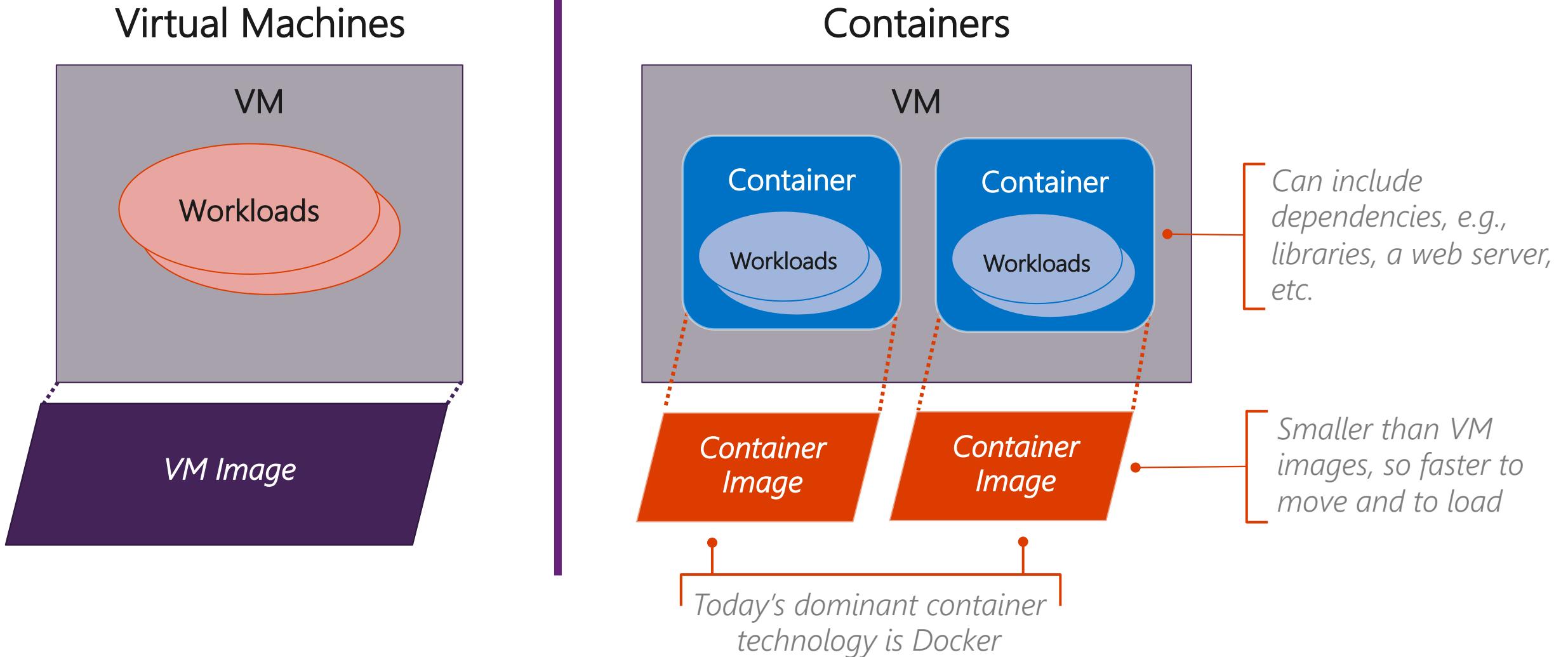
- Virtualize the hardware
- VMs as units of scaling



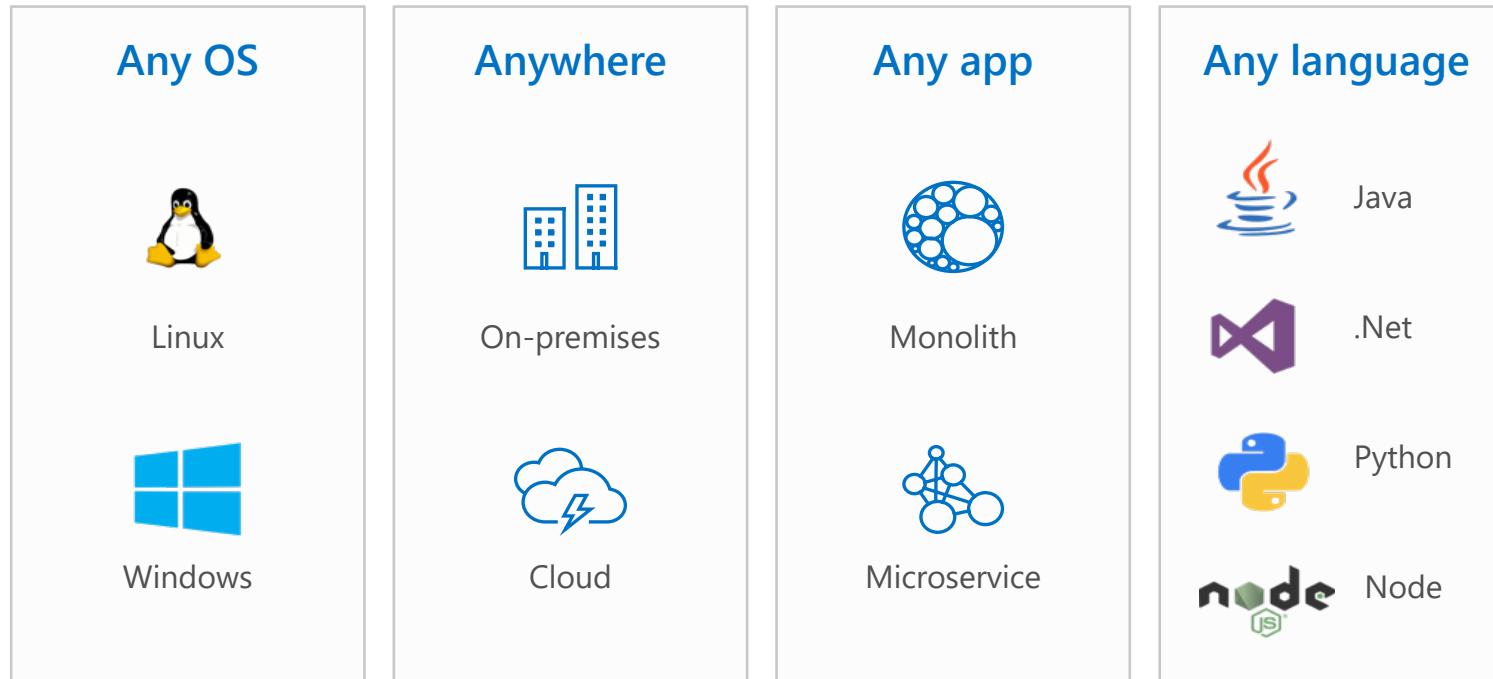
Containers

- Virtualize the operating system
- Applications as units of scaling

Illustrating Containers



The benefits of using containers



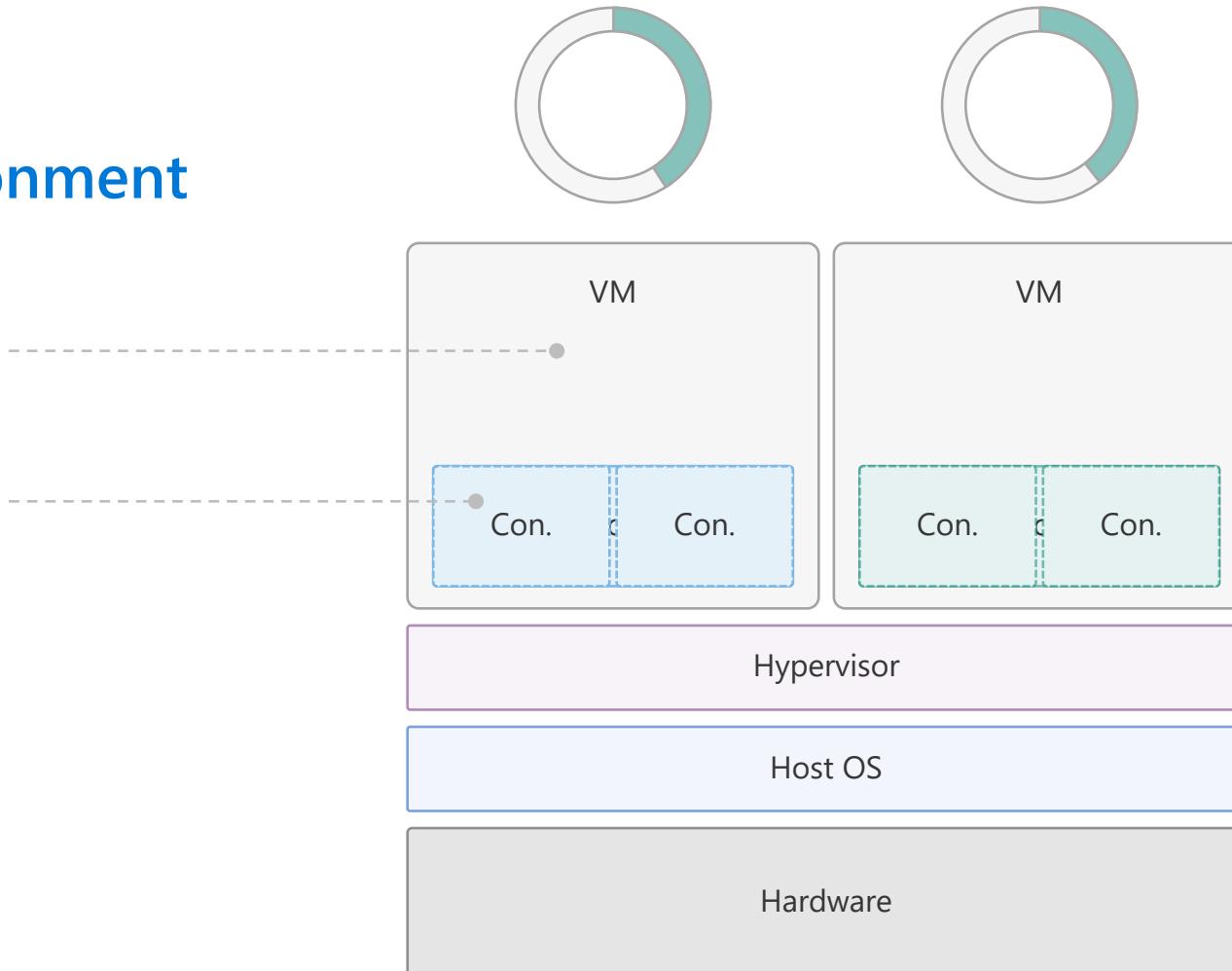
The container **advantage**

Traditional virtualized environment

Low utilization of container resources

Containerization of applications and their dependencies for portability

From dev to production agility across development and operations teams



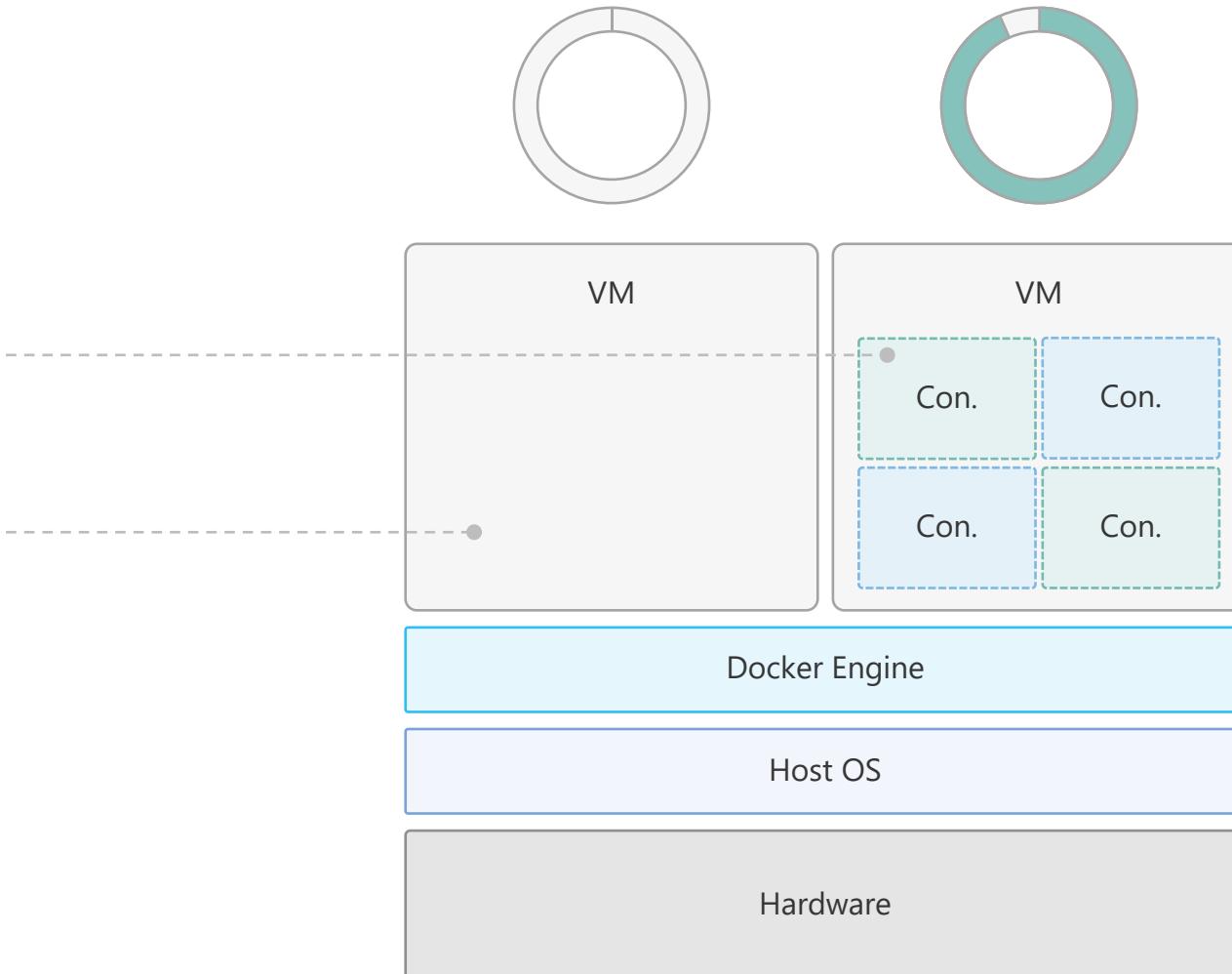
The container **advantage**

Containerized environment

Migrate containers and their dependencies to underutilized VMs for improved density and isolation

Decommission unused resources for efficiency gains and cost savings

Container is lighter weight and faster to scale dynamically



Docker and Kubernetes

What is docker?

An open-source container runtime
Mac, Windows and Linux support

```
# The world's simplest Dockerfile
$ cat Dockerfile
FROM scratch
COPY hello /
CMD ["/hello"]

# Let's create a docker image "tagged" hello-world
$ docker build -t hello-world .

# And run it...
$ docker run hello-world
```

DEM

The elements of **orchestration**



Scheduling



Affinity/anti-affinity



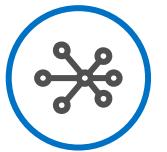
Health monitoring



Failover



Scaling



Networking



Service discovery



Coordinated app upgrades

Kubernetes

Portable

Public, private, hybrid,
multi-cloud

Extensible

Modular, pluggable,
hookable, composable

Self-healing

Auto-placement, auto-restart,
auto-replication, auto-scaling



Deploy your
applications quickly
and predictably

Scale your
applications on
the fly

Roll out
new features
seamlessly

Limit hardware
usage to required
resources only

Q: Can I do DevOps
without Docker or
Kubernetes?

A: Yes. Totally. Absolutely.



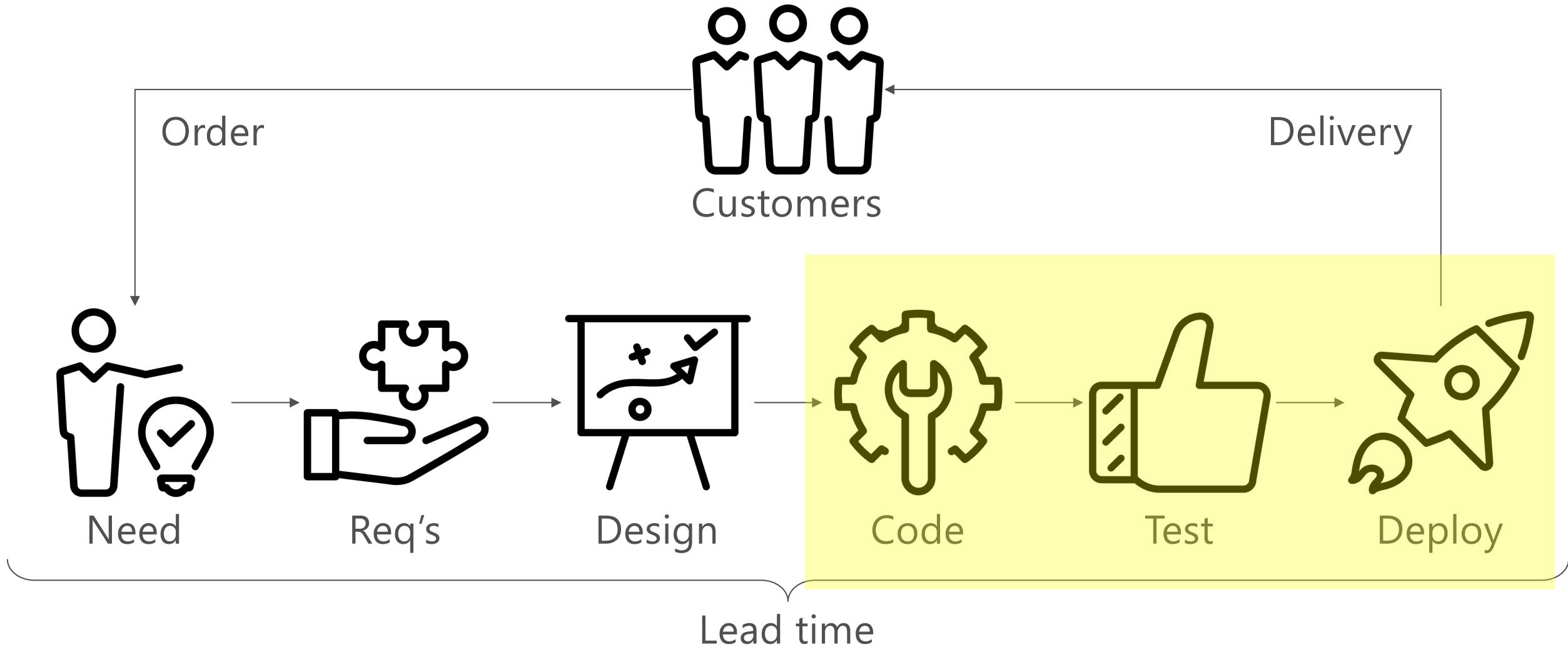
Waste removal exercise

Pop quiz

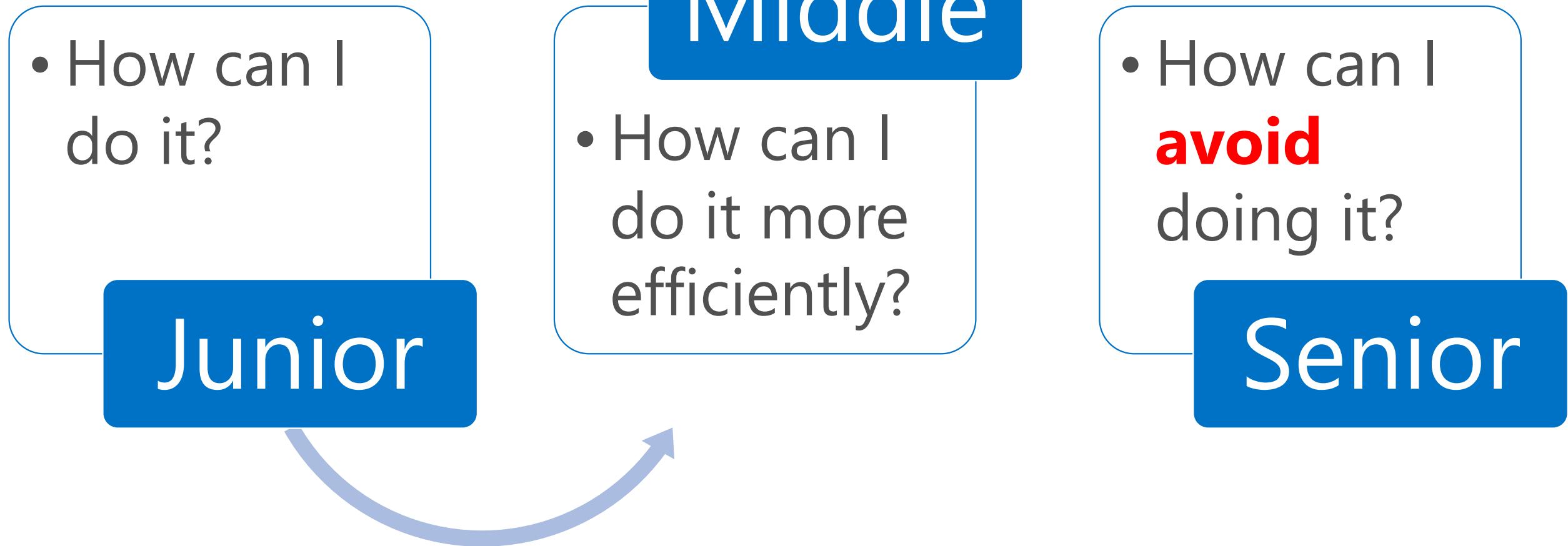
Which DevOps practice is more important?

1. Build automation.
2. Deployment automation.
3. Test automation.
4. Infrastructure provision automation.
5. Backup automation.

Value stream analysis



Engineer maturity levels



Use cases

1. Development iteration takes 2 weeks, testing iteration takes 6 weeks.
2. SaaS single tenant application.
3. No deployments, no incidents.
4. Cross-platform command line tool.

Questions?



Homework

Create accounts

- GitHub personal and organization accounts.
- Docker Hub personal account.

Install

- Docker Desktop (<https://www.docker.com/products/docker-desktop/>).
- Git SCM (<https://git-scm.com>).
- Sourcetree (optional, <https://www.sourcetreeapp.com>).
- Visual Studio Code (<https://code.visualstudio.com>).
- Docker by Microsoft (<https://marketplace.visualstudio.com/items?itemName=ms-azuretools.vscode-docker>).

Do (1)

- Create a public GitHub repo, name it “static-website-01”.
- Create a simple HTML file.
For example, “hello world from <first name> <last name>!”
- Create an image that would use either nginx or httpd to host a single page website (previously created HTML file).
- Push created image to the Docker Hub.
- The abovementioned HTML file and Dockerfile should be pushed to GitHub repo.
- Invite the **boyko-ant** GitHub user as a co-owner to your organization.

Do (2)

- Enable Hyper-V on your local PC.
- Create a VM (1 CPU, 2 Gb RAM, 32 Gb disk), name it “github-runner-01”, install Ubuntu Server 22.04 LTS (<https://ubuntu.com/download/server>) on it.

Deadline

Recommended – by the end of the day 05.08.2022



Maximum – by the end of the day 12.08.2022

