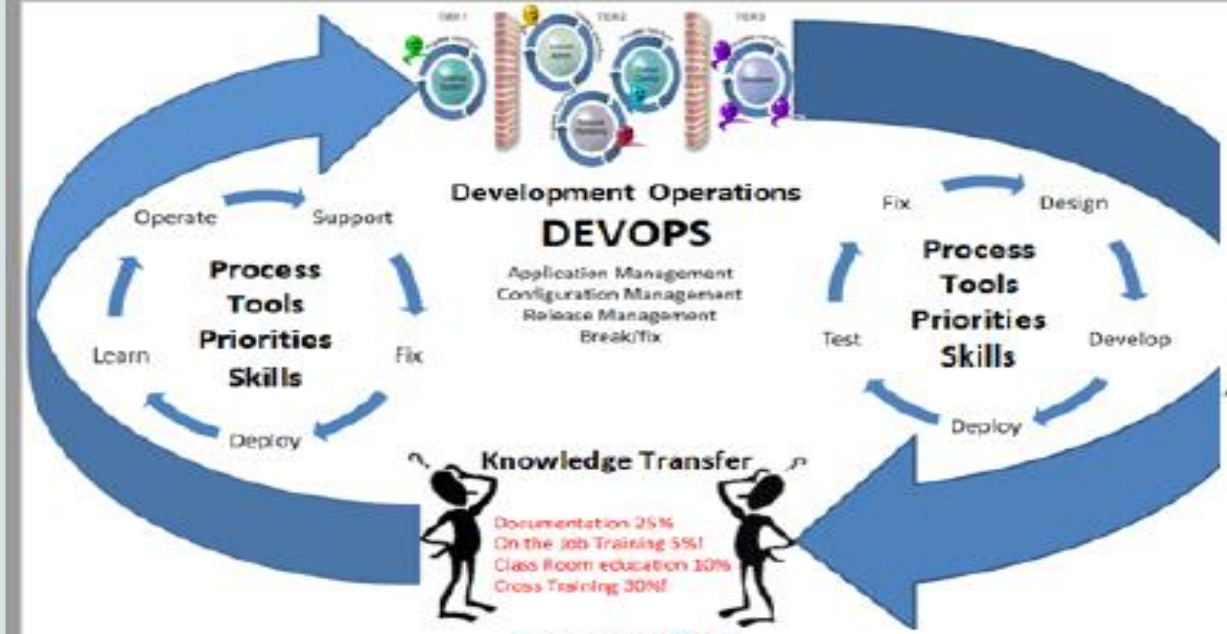
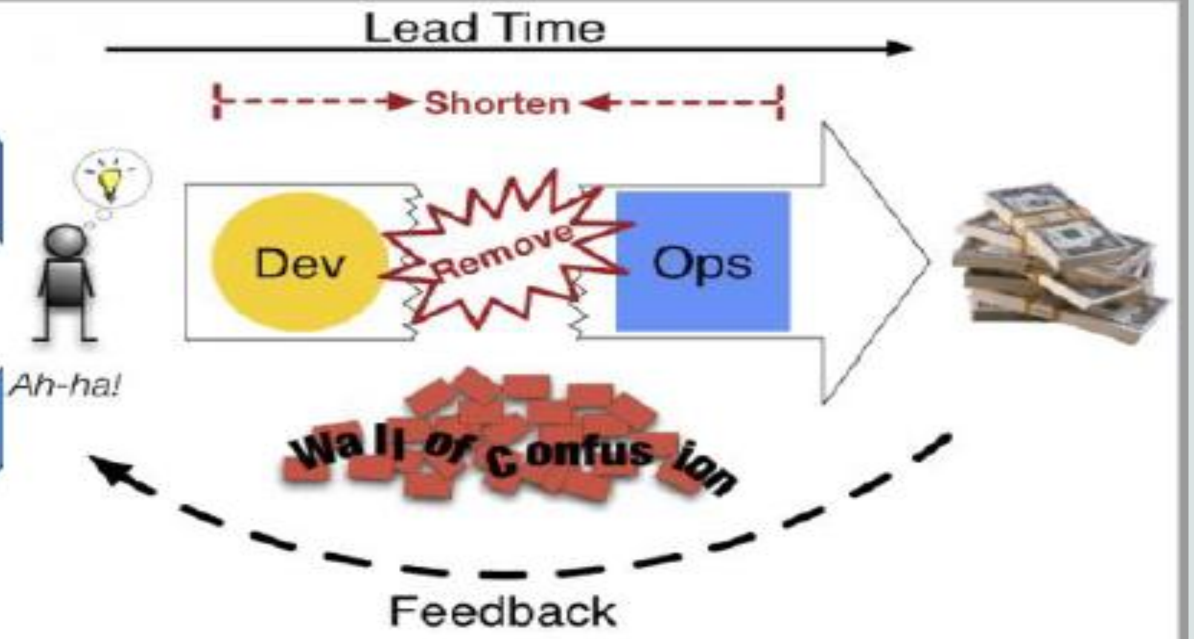
DevOps

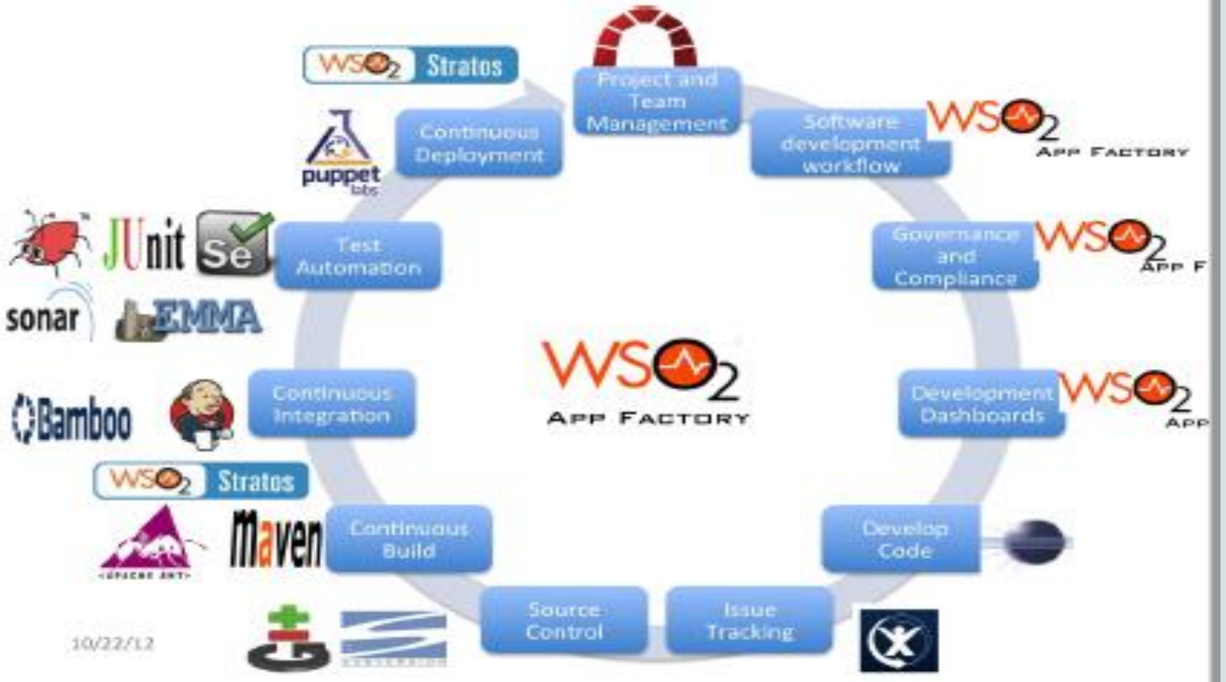
Questions I will answer today?

- **What is DevOps?**
- **What Problems Will DevOps Help Me Solve?**
- **How Do I Get Started?**
- **What Mistakes Can I Avoid?**
- **Cloud and DevOps?**
- **How can we use the DevOps as a Service?**
- **What are the current cloud technology/approaches/tools follow the DevOps?**


Development Operations
DEVOPS
Application Management
Configuration Management
Release Management
Break/fix
Operate
Support
Fix
Deploy
Learn
Process
Tools
Priorities
Skills
Fix
Design
Develop
Deploy
Test
Knowledge Transfer
Documentation 25%
On the Job Training 5%!
Class Room education 10%
Cross Training 30%!


Lead Time
Shorten
Ah-ha!
Dev
Remove
Ops
Wall of Confusion
Feedback

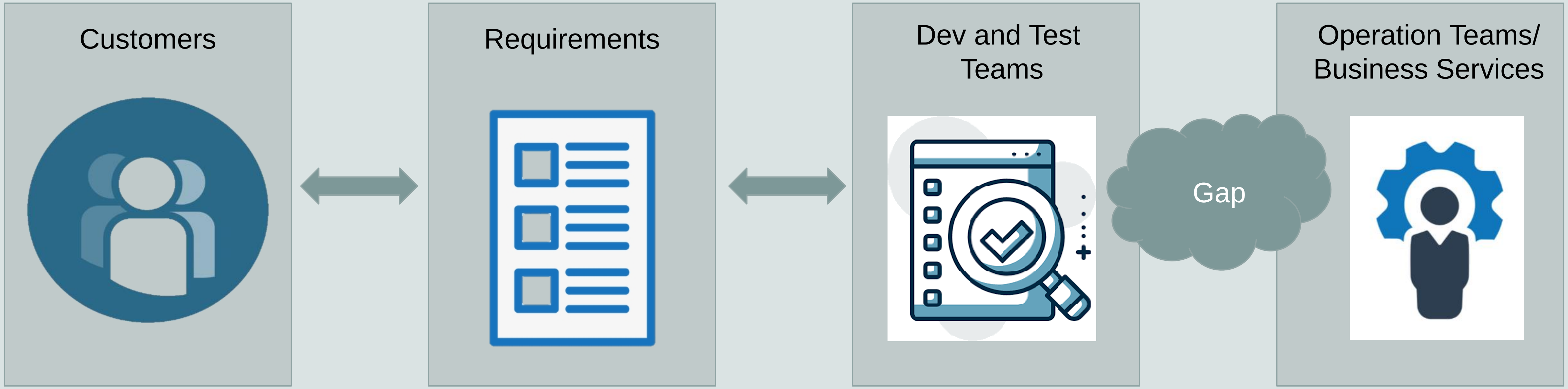
Develop/Test
Steer
Lean
Agile
Deploy
Operate


WSO2 Stratos
Project and Team Management
Software development workflow
WSO2 App Factory
Continuous Deployment
puppet labs
Governance and Compliance
Test Automation
JUnit
Se
sonar
EMMA
WSO2 App Factory
Development Dashboards
Bamboo
Continuous Integration
WSO2 Stratos
Continuous Build
maven
Develop Code
Source Control
Issue Tracking
10/22/12

The Problem In A Nutshell

- Everything needs software.
- Software runs on a server to become a service.
- Delivering a service from inception to its users is too slow and error-prone.
- There are internal friction points that make this the case.
- This loses you money. (Delay = loss)
- Therefore IT is frequently the bottleneck in the transition of “concept to cash.”

Why DevOps?


Customers
Requirements
Dev and Test Teams
Gap
Operation Teams/ Business Services

Symptoms

- Defects released into production, causing outage
- Inability to diagnose production issues quickly
- Problems appear in some environments only

- Long delays while dev, QA, or another team waits on resource or response from other teams
- “Manual error” is a commonly cited root cause
- Releases slip/fail
- Quality of life issues in IT

Why Does This Problem Exist?

- "Business-IT Alignment?"
- The business has demanded the wrong things out of IT
  - Cost sensitive
  - Risk averse
- IT has metastasized over time into a form to give the business what it's said it wants
  - Centralized and monolithic
  - Slow and penny wise, pound foolish

Why gaps?

DEV VIEW

- Dev systems may not be the same as production system.
- Developers will have faster turn around time w.r.t features.
- Not very much concerned about the infrastructural as well as deployment impact.

OPS VIEW

- Rewarded mainly for uptime.
- Lesser turn around time w.r.t feature deployment and testing due to large number of dev builds coming their way.
- Very much concerned about the infrastructural as well as deployment impact.

DevOps Defined

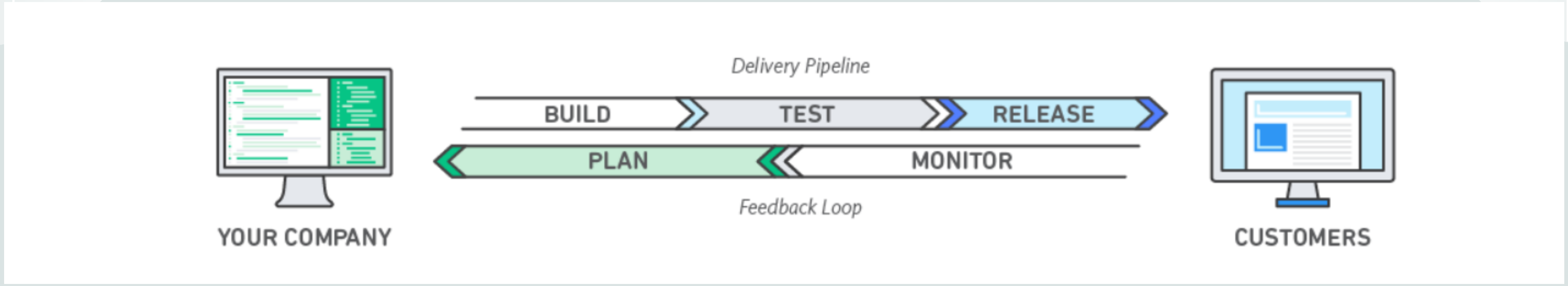
- DevOps is the practice of operations and development engineers participating together in the entire service lifecycle, from design through the development process to production support.
- DevOps is also characterized by operations staff making use many of the same techniques as developers for their systems work.

DevOps Defined

- DevOps is the practice of operations and development engineers participating together in the entire service lifecycle, from design through the development process to production support.
- DevOps is also characterized by operations staff making use many of the same techniques as developers for their systems work.

Dev and Ops

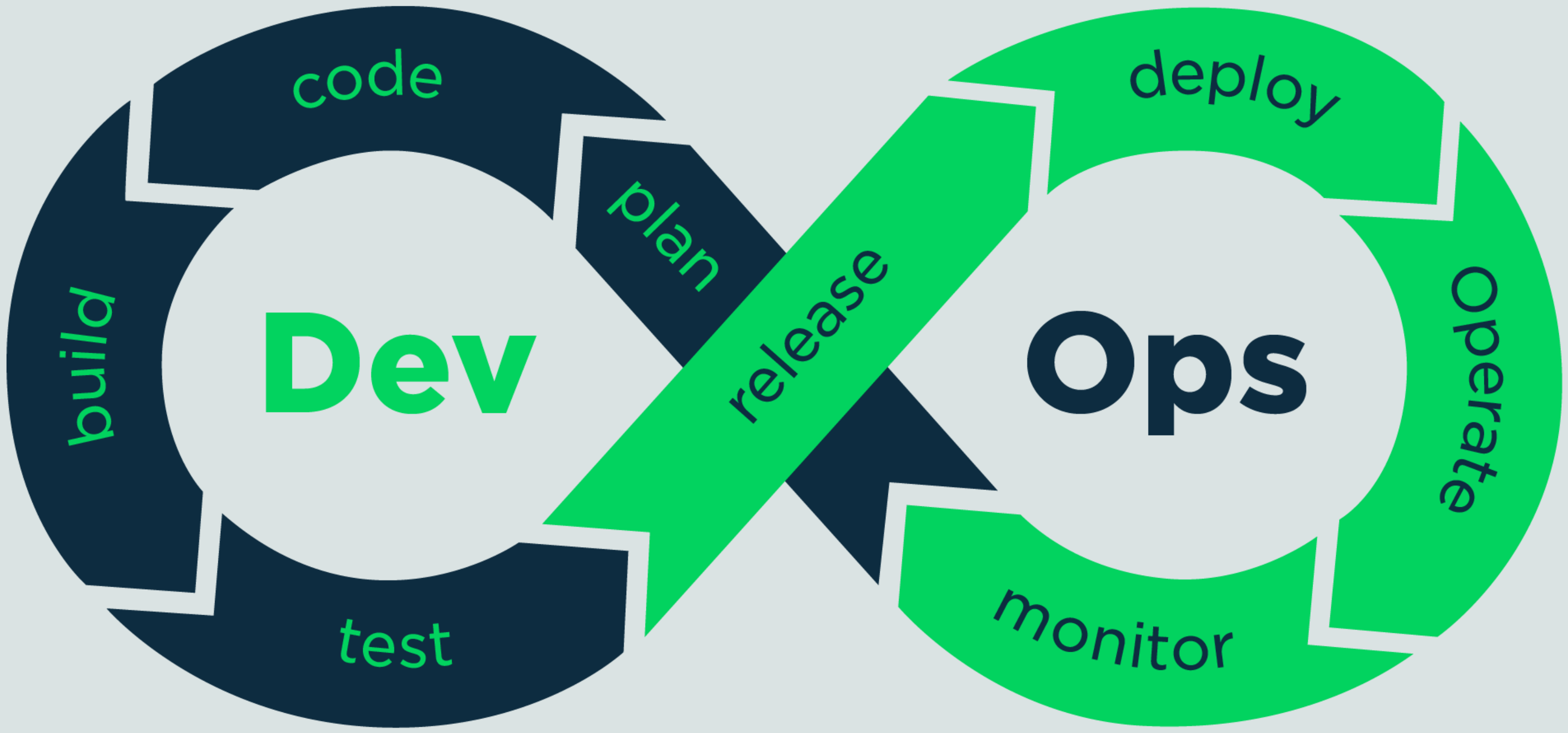
- Developers work with Ops to understand the impact of code changes.
- Developers now work more closely with production-equivalent systems.
- Developers focus on metrics required by Ops team.
- Closely monitors the Dev – Test – Prod pipeline for each deployment with immediate feedback.
- Better collaboration and communication.



What is DevOps?

- DevOps is a set of practices that combines software development and IT operations. It aims to shorten the systems development life cycle and provide continuous delivery with high software quality.
- It stresses communication and integration between developers and IT professionals.
- It reduces risk, reduces cost and improves quality.

What is DevOps?



DevOps Concepts

DevOps Principles

DevOps Practices

DevOps Tools

DevOps Principles

- The Three Ways
  - Systems Thinking
  - Amplify Feedback Loops
  - Culture of Continual Experimentation
- CAMS
  - Culture – People > Process > Tools
  - Automation – Infrastructure as Code
  - Measurement – Measure Everything
  - Sharing – Collaboration/Feedback
- Informed by the values in the Agile Manifesto and Lean Theory of Constraints

DevOps Practices

- Version Control For All
- Automated Testing
- Proactive Monitoring and Metrics
- Kanban/Scrum
- Visible Ops/Change Management
- Configuration Management
- Incident Command System
- Continuous Integration/Deployment/Delivery
- “Put Developers On Call”
- Virtualization/Cloud/Containers
- Toolchain Approach
- Transparent Uptime/Incident Retrospectives

Some DevOps Practices

- Continuous Integration (CI)

Continuous Integration is a software development practice in which the developers frequently integrate their code modifications.

- Continuous Testing (CT)

Continuous Testing is a software testing practice that involves automated software testing at all the stages of the software development life cycle.

- Continuous Delivery (CD)

Continuous Delivery is a software development procedure where code modifications are automatically released into the production environment.

How are DevOps and Cloud related?

CLOUD

- Cloud technology ensures that companies can access an endless range of features and solutions at a speed that suits them.
- With the cloud, there’s no limit on how much functionality a company can access.

DEVOPS

- In the same way, DevOps supports an agile environment for everyone involved. Both solutions have their own benefits to consider for speed and productivity.
- However, when the cloud and DevOps come together, they enhance each other’s strengths to create an even more compelling offering.

DevOps as a Service

- With DevOps as a Service, you are relocating some of those enterprise resources to the cloud. Rather than using discrete software development tools for development, DevOps as a Service collects the development tools into a single toolset that is hosted in the cloud.
- The objective is to ensure that developers use a common toolset and that every action is tracked to promote continuous delivery, continuous integration, and business value.

The Pros of DevOps as a Service

- Cloud-based DevOps makes it easier to collaborate, putting all the tools in the cloud so they can be accessed by users anywhere.
- DevOps as a Service means faster testing and deployment. Generally, using cloud services enables increased release frequency. It also gives developers more computing power and data storage as they need it.
- Using DevOps as a Service hides the complexities of data and information flow, which means DevOps team members can focus on their specific tools without having to understand the entire tool chain.

The Pros of DevOps as a Service

- Using cloud services is a more data-driven process where everyone uses the same data set. This lends itself to better documentation and tighter quality control.
- DevOps as a Service does not have to function on its own; it can coexist with internal DevOps development and deployment processes. Using DevOps as a Service simply makes it easier to offload specific aspects of a project for better collaboration and faster turnaround.

The Cons of DevOps as a Service

- Outsourcing a DevOps infrastructure requires a specific level of software development expertise, including an in-depth understanding of integration, infrastructure, and orchestrating workflow. You need the experts along with the tools for DevOps as a Service to succeed.
- Can you really clone a production environment in the cloud? Cloning an enterprise infrastructure for test purposes is complex and can lead to unforeseen compatibility problems.
- Security is always a concern. The security team is usually not part of DevOps, and the DevOps team tends to choose speed over security when developing software.

DevOps Tool

- DevOps tool is an application that helps automate the software development process.
- DevOps tool enables teams to automate most of the software development processes like build, conflict management, dependency management, deployment, etc. and helps reduce manual efforts.

Some DevOps Tools

- QuerySurge

QuerySurge is the smart data testing solution that is the first-of-its-kind full DevOps solution for continuous data testing.

1. Seamlessly integrates into the DevOps pipeline for continuous testing
2. Verifies large amounts of data quickly
3. Detects requirements and code changes, updates tests accordingly and alerts team members of said changes
4. Provides detailed data intelligence & data analytics

The Cloud Hosting

- The way we orchestrate infrastructure had to change as well. A higher level of dynamism and elasticity was required. That became especially evident with the emergence of cloud hosting providers like Amazon Web Services (AWS) and, later on, Azure and GCE.

Infrastructure as code

- Still, the question remains. How do we manage infrastructure in the cloud with all the benefits it gives us? How do we handle its highly dynamic nature? tools like CloudFormation or agnostic solutions like Terraform. When combined with tools that allow us to create images, they represent a new generation of configuration management.

📌 Manual infrastructure drawbacks

1. Configuring infrastructure manually is very error-prone. If you want to try out a new infrastructure configuration, you have to change the environment by hand. If the change is correct, you have to remember what steps were involved in making that change and then manually apply them to your other environments. If you do not like the change, you have to remember how to roll the environment back to its original state. Since the process is manual, the changes are often not made the same to each environment which is one of the reasons that environments end up differing and have their own personalities.(change in multiple environments)

Manual infrastructure drawbacks

2. Time consumption

3. Dependency:
   Cannot destroy a piece of infrastructure if another piece depends on it.

Terraform

- Terraform solves all of the problems mentioned by defining infrastructure in code.
- Essentially, it takes the infrastructure that you have defined in code and makes it real.
- The Terraform is that it does not ask you how to get from the infrastructure you have to the infrastructure you want.
- It simply asks you what you want the world to look like and then does the hard work.

What Terraform can do

1. Configure multiple components using Terraform
   - ➢ This means that you can configure multiple components and infrastructures in a single project even though they sit in multiple clouds using the same language (HCL).
2. Terraform community.
3. Destroy an environment using Terraform.
4. Add Terraform to an existing infrastructure.

Chef and Puppet vs. Terraform

- Chef and Puppet are configuration management tools.
- They are designed to configure and manage the software that is running on a machine (infrastructure) that already exists.

Deployment Process

- Deployment changes frequently and more than infrastructure due to new releases.
- Application might need JDK6 and the other JDK7.
- A new release of the first one might require JDK to be upgraded to a new version.
- Upgrading OS.

Earlier days of deployment

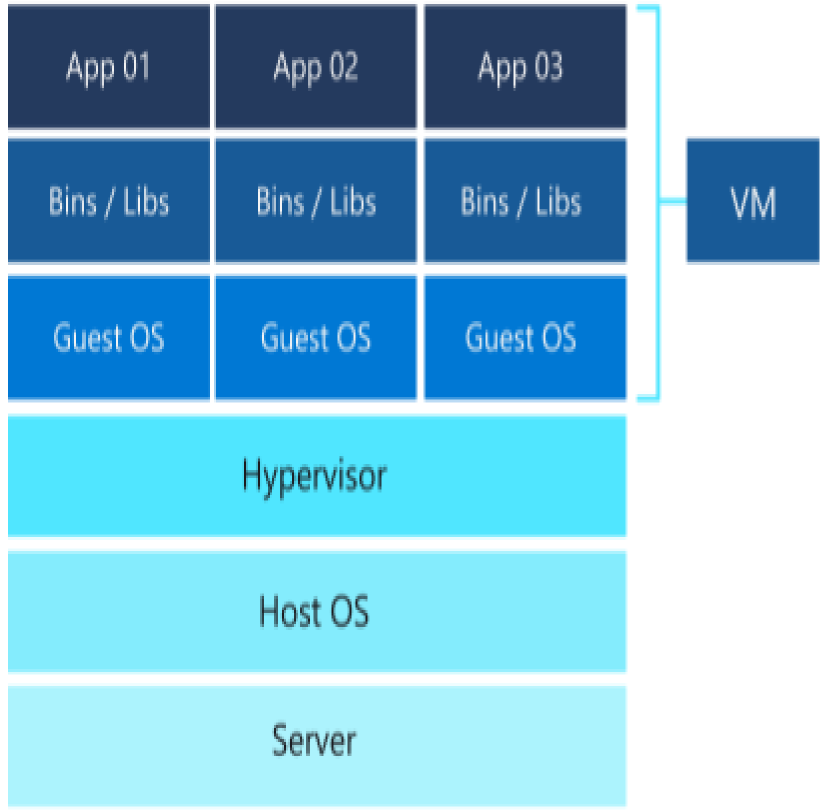
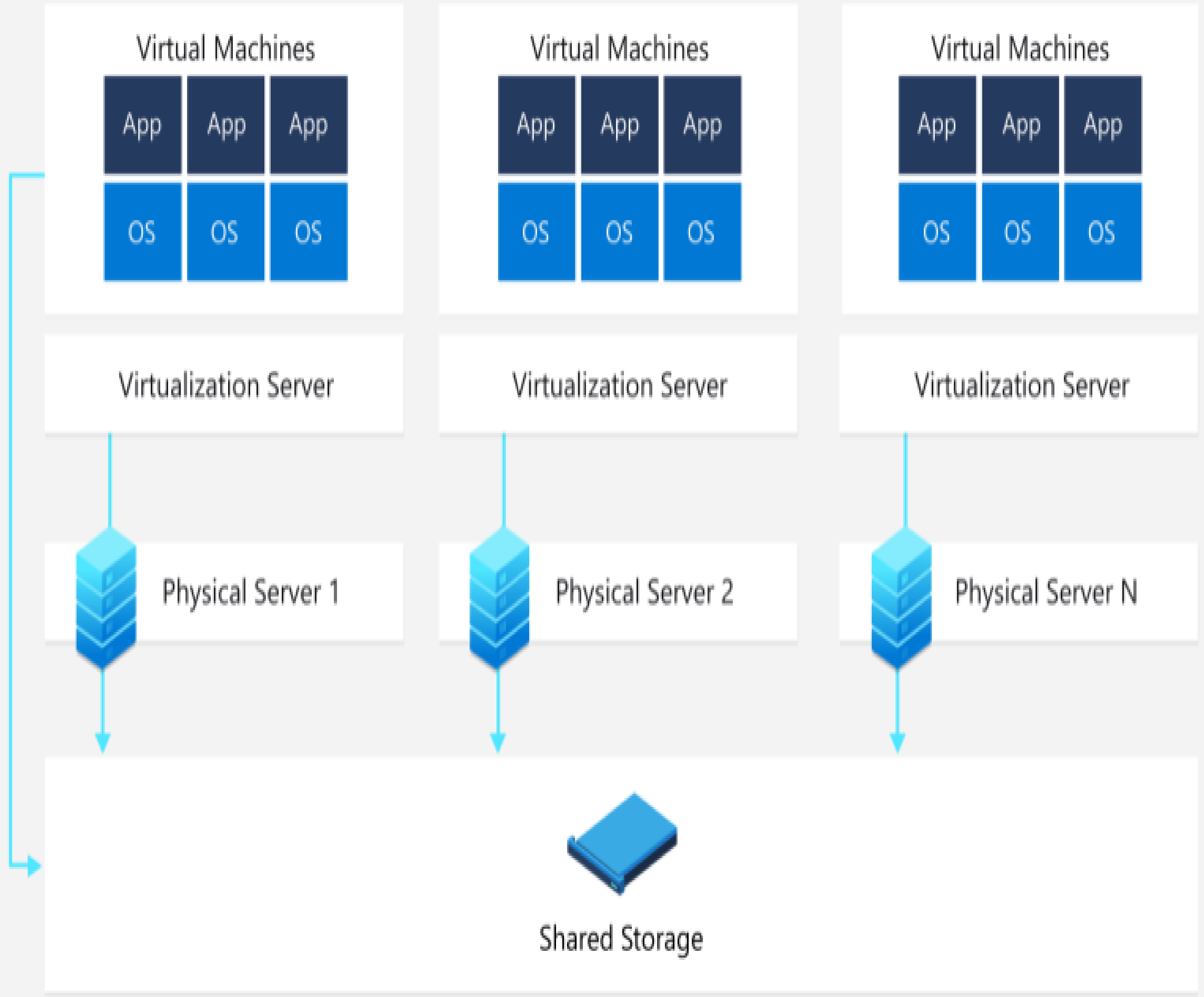
- Programmer goes to server building and install his application.
- The best we could do at that time was to zip files that form a release. More likely, we’d manually copy files from one place to another.

The Solution

- Today, containers are the preferable way to package and deploy services.
- They are the answer to immutability we were so desperately trying to implement.
- They provide necessary isolation of processes, optimized resource utilization, and quite a few other benefits. And yet, we already realized that we need much more.

Virtual Machines

- A virtual machine, is no different than any other physical computer like a laptop, smart phone, or server.
- It has a CPU, memory, disks to store your files, and can connect to the internet if needed. While the parts that make up your computer (called hardware) are physical and tangible, VMs are often thought of as virtual computers or software-defined computers within physical servers, existing only as code.
- VMs opened the doors to dynamic infrastructure where VMs are continuously created and destroyed.


App 01
App 02
App 03
Bins / Libs
Bins / Libs
Bins / Libs
Guest OS
Guest OS
Guest OS
VM
Hypervisor
Host OS
Server

Virtual Machines
App
OS
Virtualization Server
Physical Server 1
Physical Server 2
Physical Server N
Shared Storage

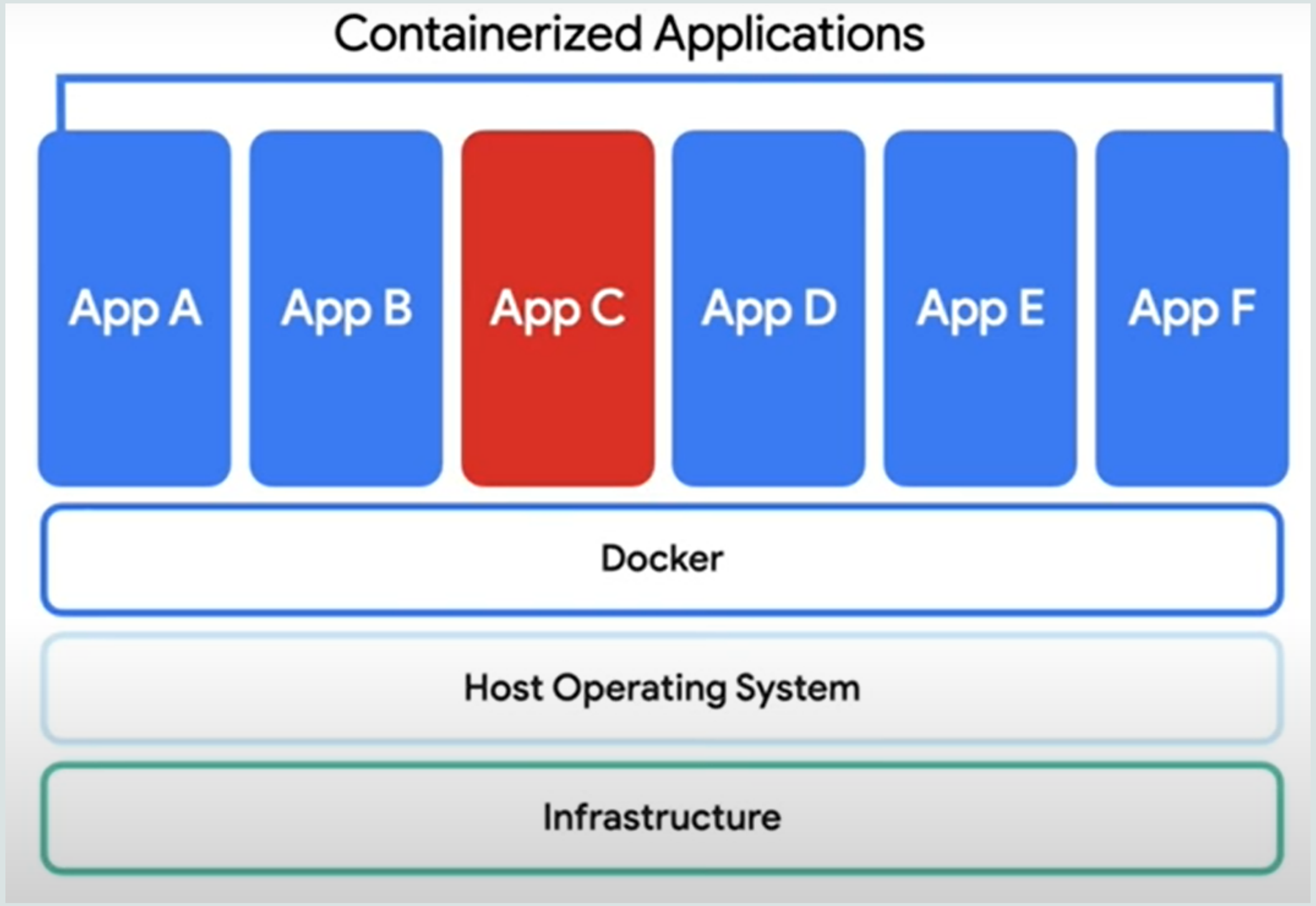
Containers

- The application is made of files served by an HTTP server (Apache here, but it could be Kestrel, IIS, NGINX, …), a runtime (PHP 5.6 here).
- Without containers, the dependencies and files are all placed together on a server. Since managing these dependencies is time-consuming, similar apps are typically grouped on the same server, sharing their dependencies.

Containers

- Containers solve this problem since each app will run inside its own container with its own dependencies. Your typical server would look like:

Containers Problems?

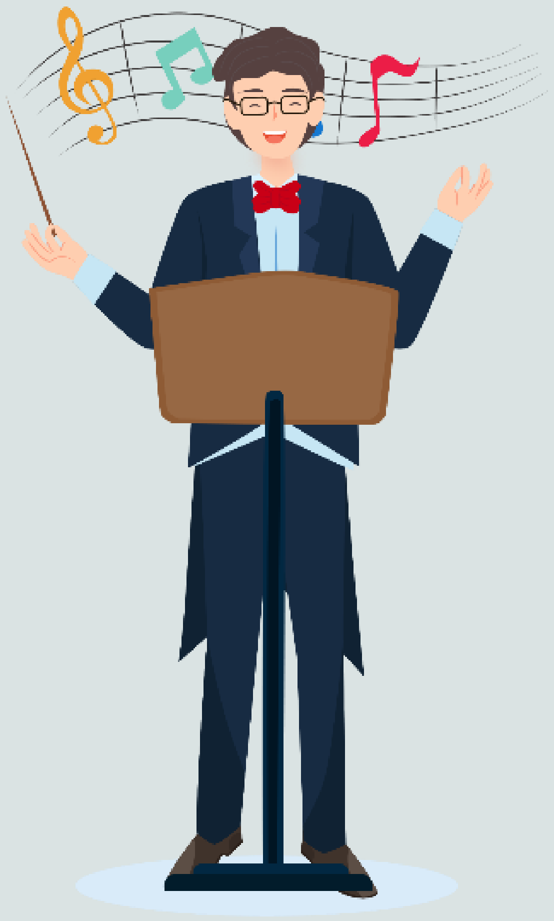


Container Schedulers

- It makes sure that resources are used efficiently and within constraints.
- It makes sure that services are (almost) always running.
- It provides fault tolerance and high availability.
- It makes sure that the specified number of replicas are deployed.
- It makes sure that the desired state requirement of a service or a node is (almost) always fulfilled.

What Is Kubernetes?

- Container Orchestrator that make sure that each container is where it’s suppose to be, and that the containers can work together.



Like conductor that manage everything in an orchestra

Resources

- https://www.educative.io/path/devops-for-developers
- https://aws.amazon.com/containers/
- https://www.forbes.com/sites/forbestechcouncil/2017/07/21/the-relationship-between-the-cloud-and-devops/?sh=1f011f4c2957