APACHE AIRAVATA

Introduction to First Half Course Topics

Marlon Pierce, Suresh Marru

Science Gateways Research Center, Indiana University

marpierc@iu.edu, smarru@iu.edu

Monolithic Applications: Traditional Software Releases

- Software releases occur in discrete increments
- Software runs on clients' systems
- Releases may be frequent but they are still distinct
  - Firefox
  - OS system upgrades
- Traditional release cycles
  - Extensive testing
  - Alpha, beta, release candidates, and full releases
- Extensive recompiling and testing required after code changes
- Code changes require the entire release cycle to be repeated

MicroServices: Software as a Service

- Does your software run as a service?
- Do you run this service yourself? Or does another part of your organization run the service that your team develops?
- Traditional release cycles don't work well
  - May make releases many times per day
  - Test-release-deploy takes too long
- You can be a little more tolerant of bugs discovered after release if you can fix quickly or roll back quickly.
- Get new features and improvements into production quickly.

Science gateways provide software as a service

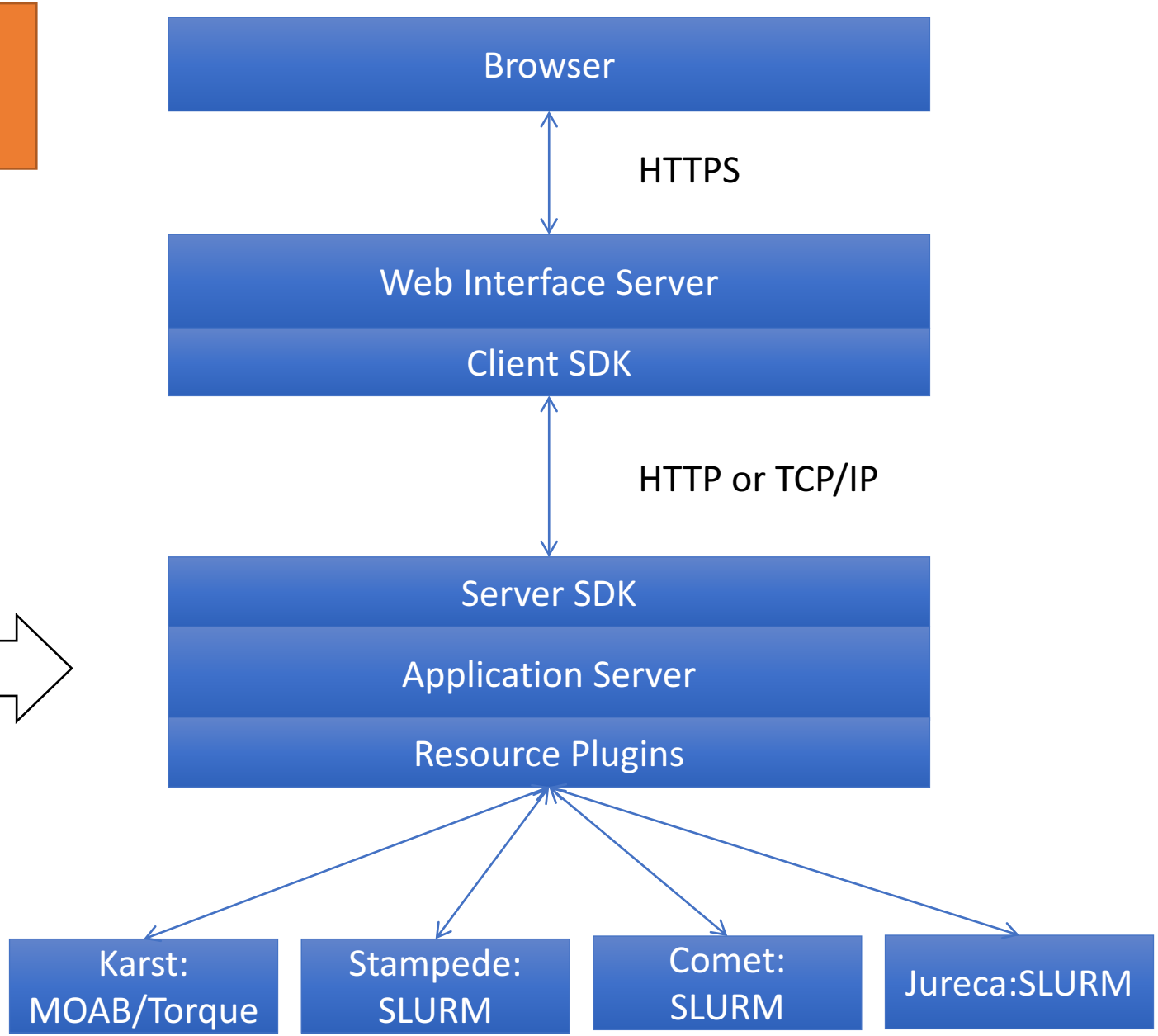
Science Gateways and Microservices

What Is a Microservice?

- Develop a single application as a suite of small services
- Each service runs in its own process
- Services communicate with lightweight mechanisms
  - “Often an HTTP resource API”
  - But that has some problems
  - Messaging and hybrid approaches
- These services are built around business capabilities
- Independently deployable by fully automated deployment machinery.
- Minimum of centralized management of these services,
  - May be written in different programming languages
  - May use different data storage technologies.

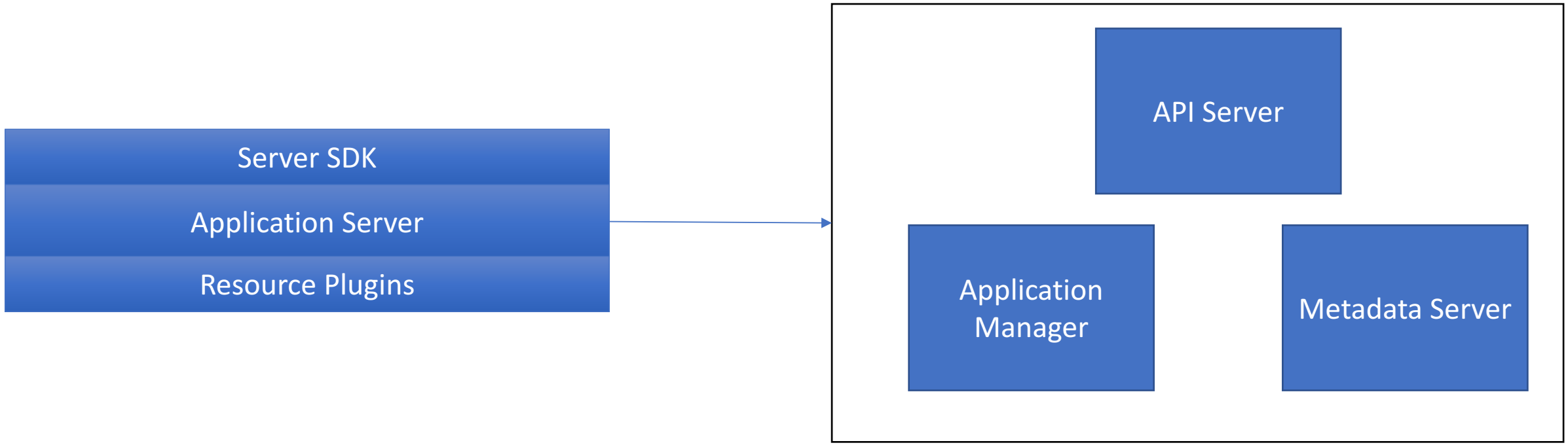
http://martinfowler.com/articles/microservices.html

Recall the Gateway Octopus Diagram

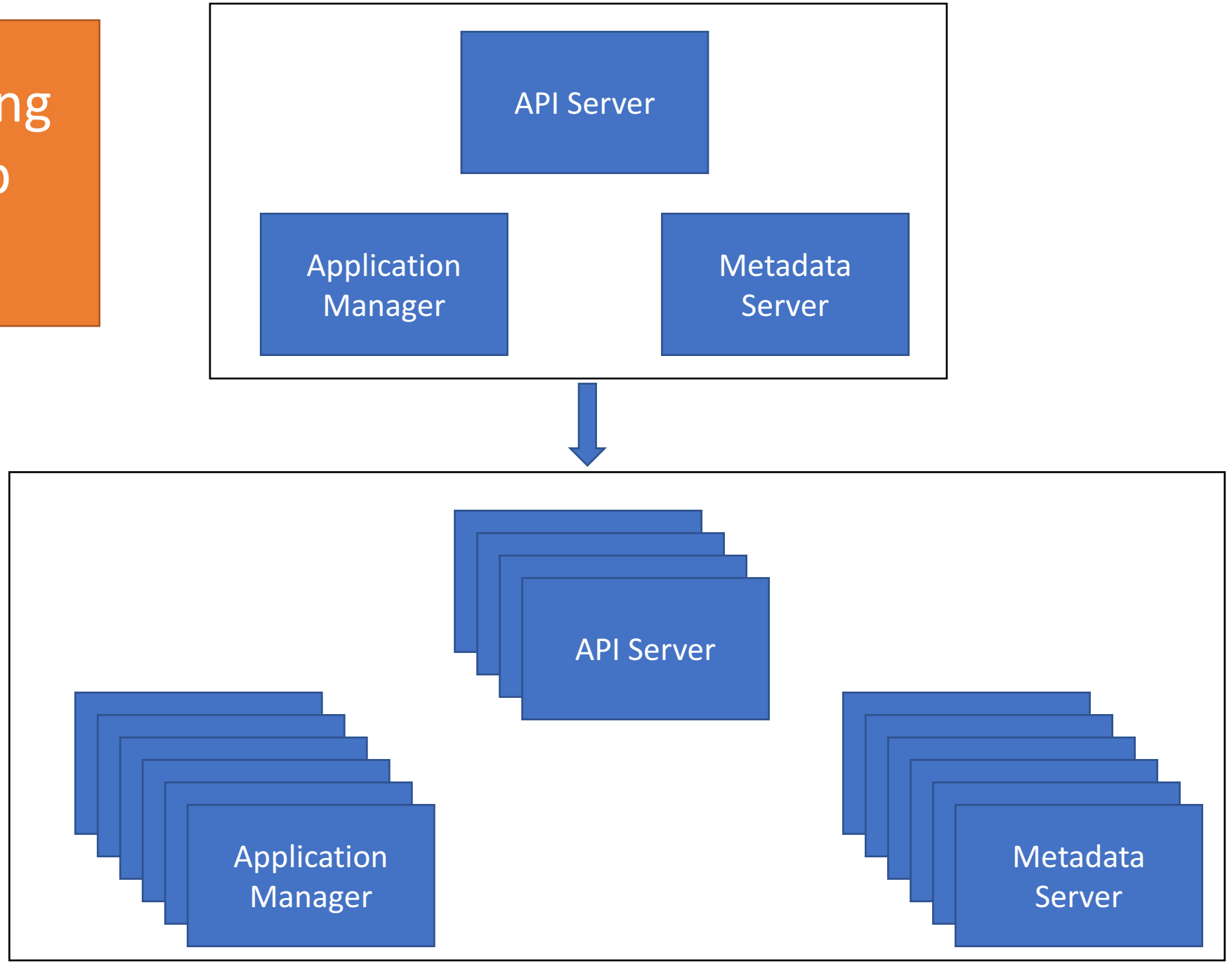


We will focus on this piece

Basic Components of the Gateway App Server



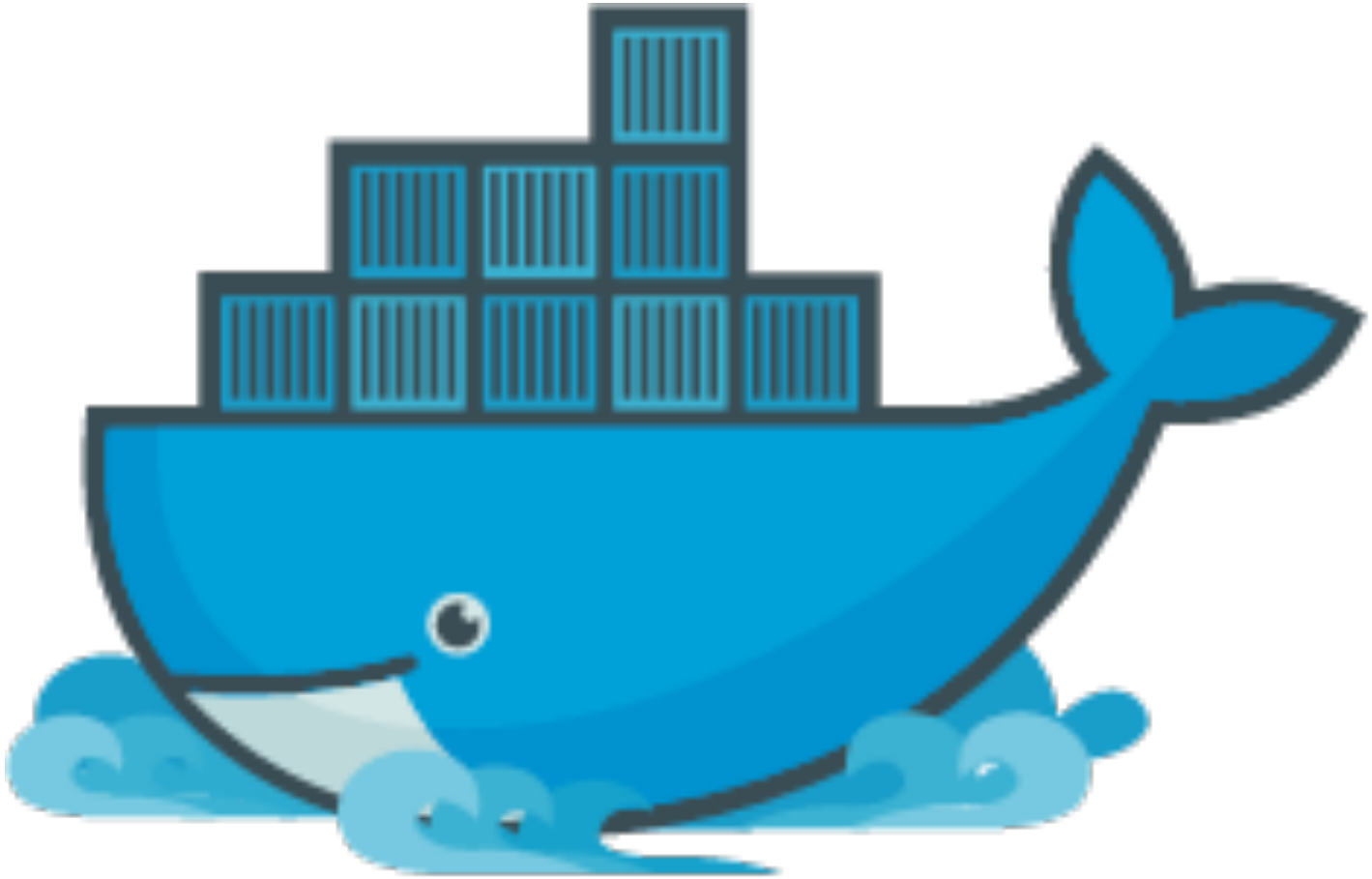
Decoupling the App Server



How Do We Package and Where Do We Run All Those Microservices?

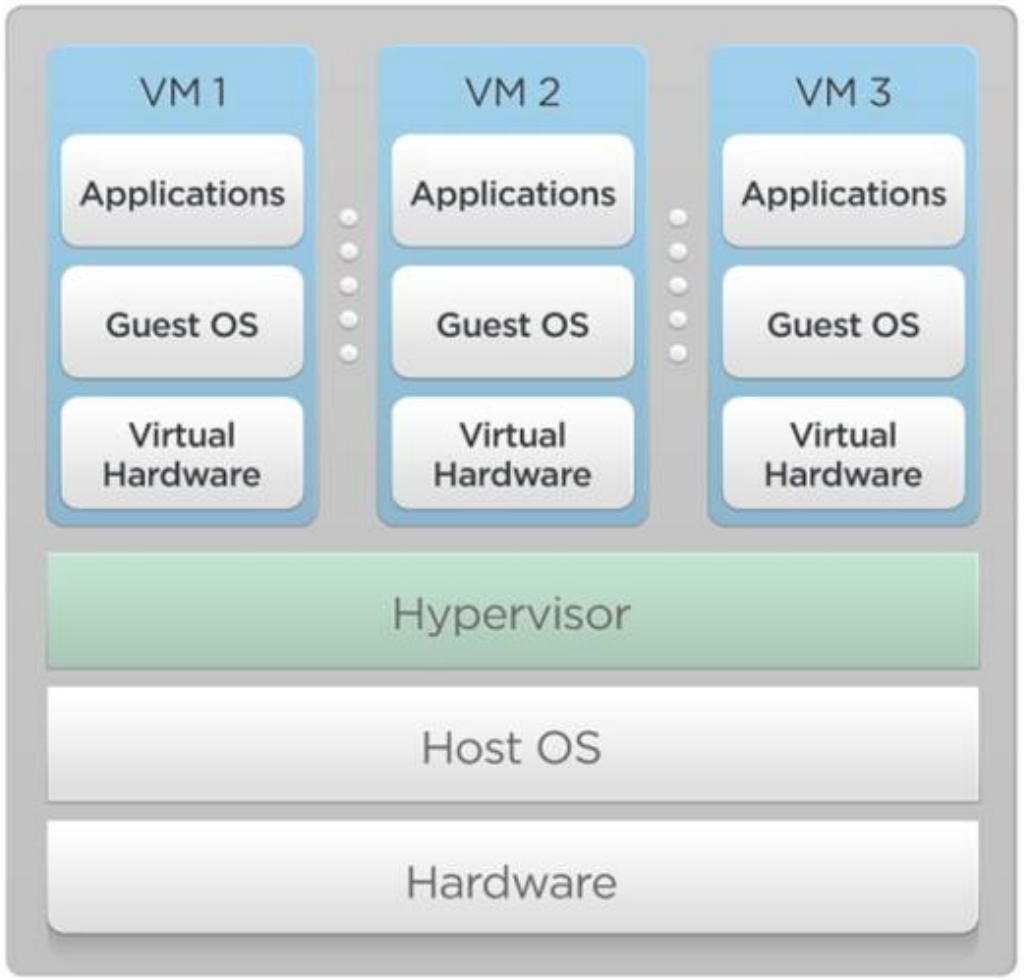
On the Cloud? In the Matrix?

Virtualization, Containers, Docker



Hypervisor Virtualization

- Hypervisors provide software emulated hardware and support multiple OS tenants.
- Type 1 Hypervisors run directly on the hardware
  - Kernel-based Virtual Machine (KVM)
  - Xen: Amazon uses (or used) this
- Type 2 Hypervisors run as another program in the HostOS
  - Virtual Box
  - VMWare



Virtualization Benefits and Drawbacks

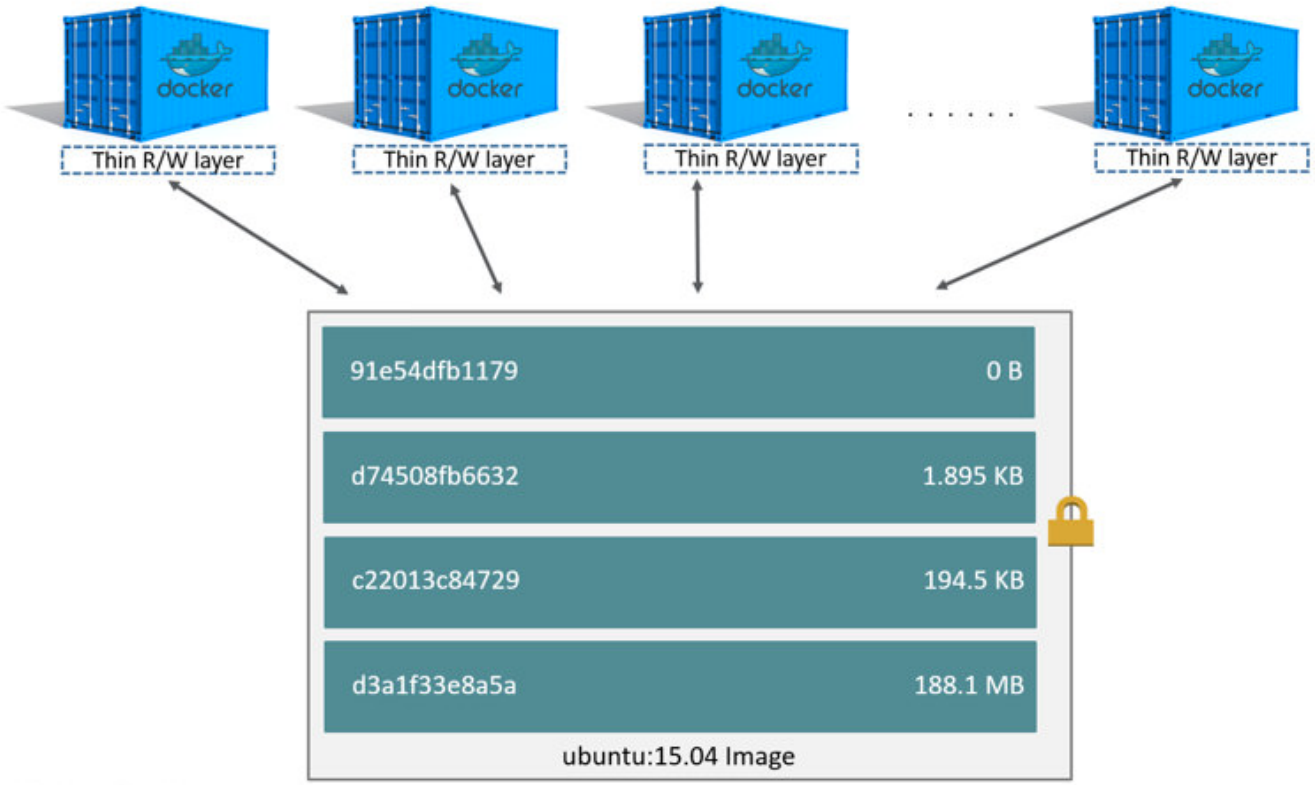
- Makes your development, testing, and deployment environments the same
- Supports Continuous Integration and Deployment

- Adds overhead
  - Relatively slow to create and destroy VMs
  - Virtualized hardware and networking are slower than real systems
  - Uses memory, disk, and CPU resources

You can run 1 VM per microservice, but there is a better way.

Docker and Other Linux Containers

- Docker replaces the need for DIY containers
- Docker uses LXC containerization + AuFS
  - AuFS lets Docker containers share common files
  - AuFS also allows you to have efficient version control of container images



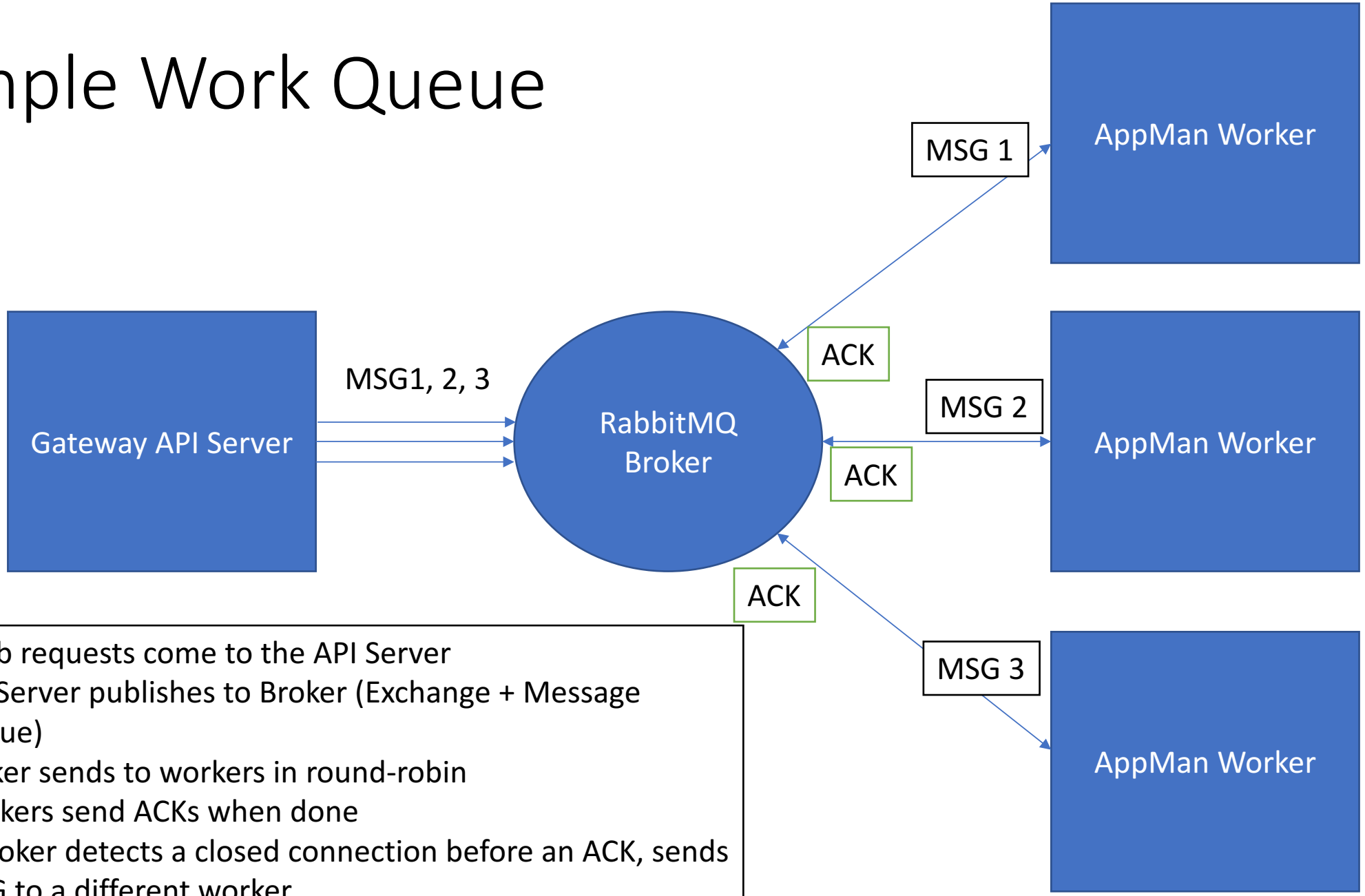
Because each container has its own thin writable container layer, and all changes are stored in this container layer, multiple containers can share access to the same underlying image and yet have their own data state.

How Do Microservices Communicate?

Push, pull, etc

Messaging Systems: RabbitMQ, Apache Kafka

Simple Work Queue



- 3 Job requests come to the API Server
- API Server publishes to Broker (Exchange + Message Queue)
- Broker sends to workers in round-robin
- Workers send ACKs when done
- If Broker detects a closed connection before an ACK, sends MSG to a different worker

How Can Components Expose their APIs and Data Models to Other Components?

And can we make this programming language independent?

API and Metadata Model Design

Apache Thrift ™

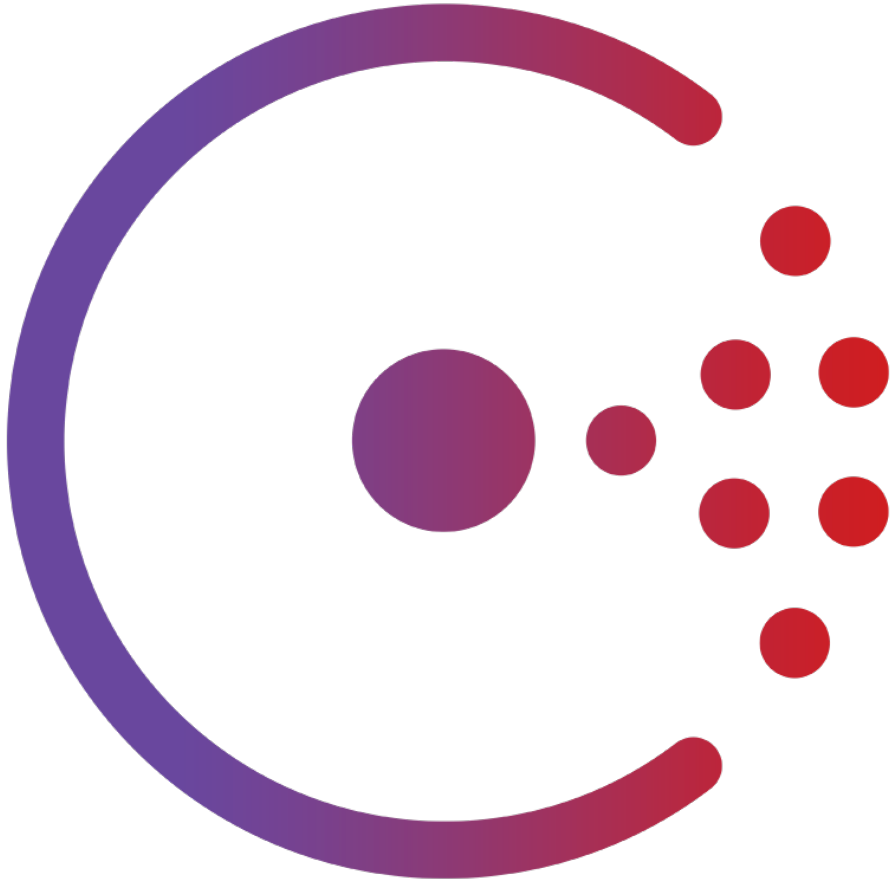
RESTful API

GET PUT POST DELETE

How Can I Discover, Monitor, and Manage Services?

Can we learn some lessons from distributed systems research?

Distributed State Management: Zookeeper, Consul



How Do I Manage Logs from Microservices

And detect if there are problems


Log Files
Logstash
Customized Agent
elastic
kibana

How Can I Secure my API Server and Microservices?

How do I manage user identities, authentication and authorization?

Security: OAuth2 and OpenIDConnect

How Does the Gateway Interact with Computing Clouds and Supercomputers?

How can we work reliably with unreliable resources?

Running Jobs Overview

Researcher

Login

Command Line

When you connect to a resource, you are on a login node shared by many users.

Batch Script

Commands for code execution, copy input files to scratch,… Specify number/type of nodes, length of run, output directory, …

Data

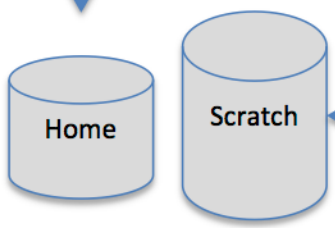
Job

Login Nodes

Use for tasks such as file editing, code compilation, data backup, and job submission.

*Store project files such as source code, scripts, and **small** input data sets to your Home directory.*

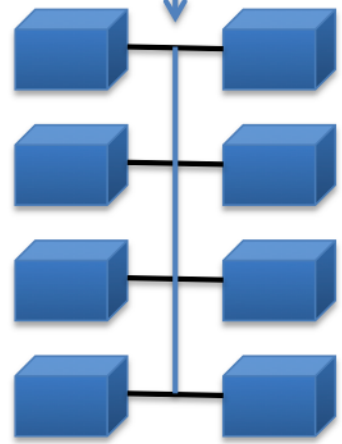
Run jobs by submitting your batch script to the compute nodes using the "qsub" command.



File System

Data

Read/write data from compute nodes to your Scratch directory.



Compute Nodes

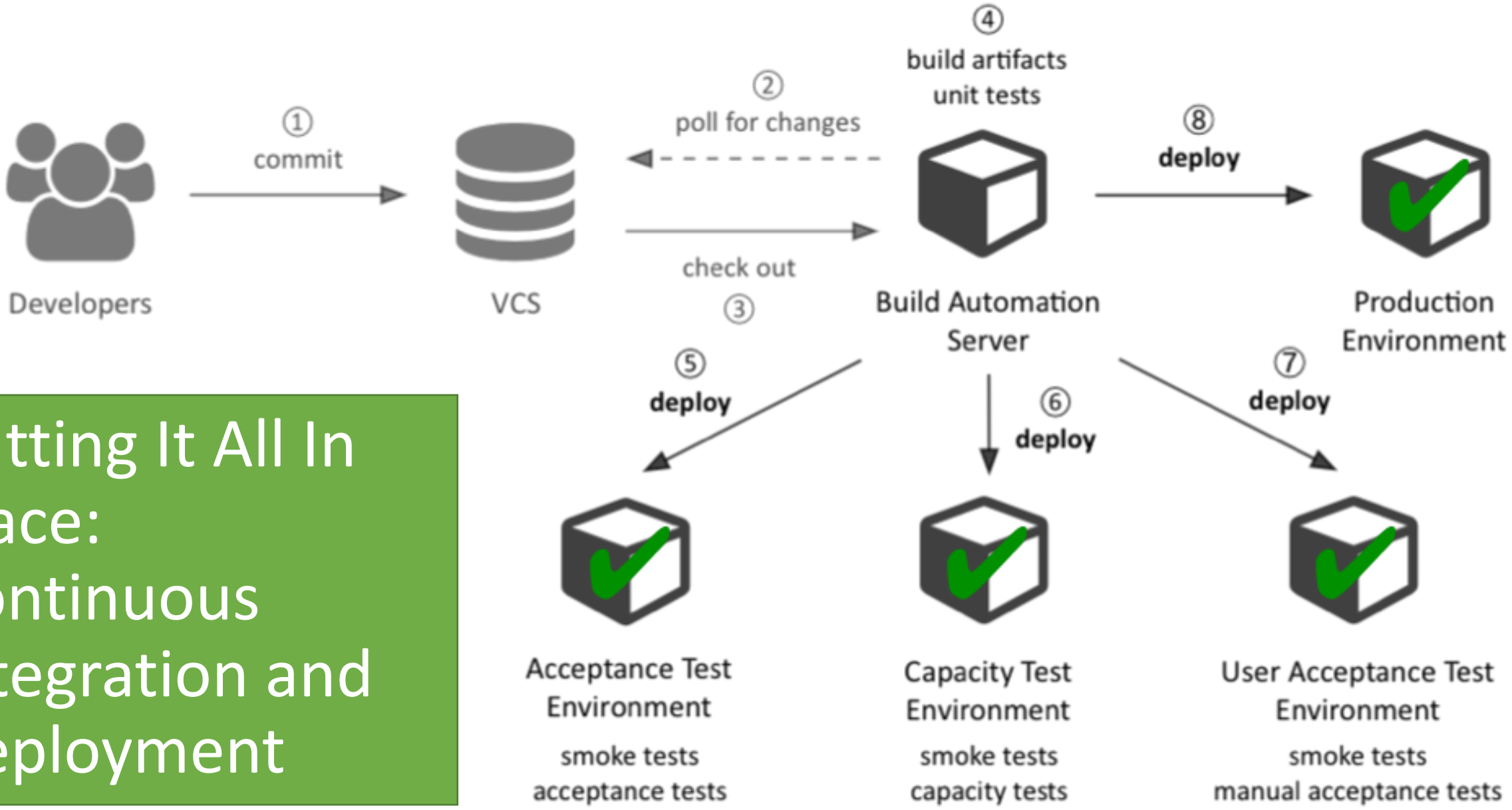
Your job is submitted to a queue and will wait in line until nodes are available. Queues are managed by a job scheduler that enables jobs to run efficiently.

Resource Management and Scheduling: Mesos, Aurora, and Torque

How Can We Automate All of This?

How can we make our infrastructure reproducible?

Putting It All In Place: Continuous Integration and Deployment



The Scientific Method as an Ongoing Process

Next Semester: Science!