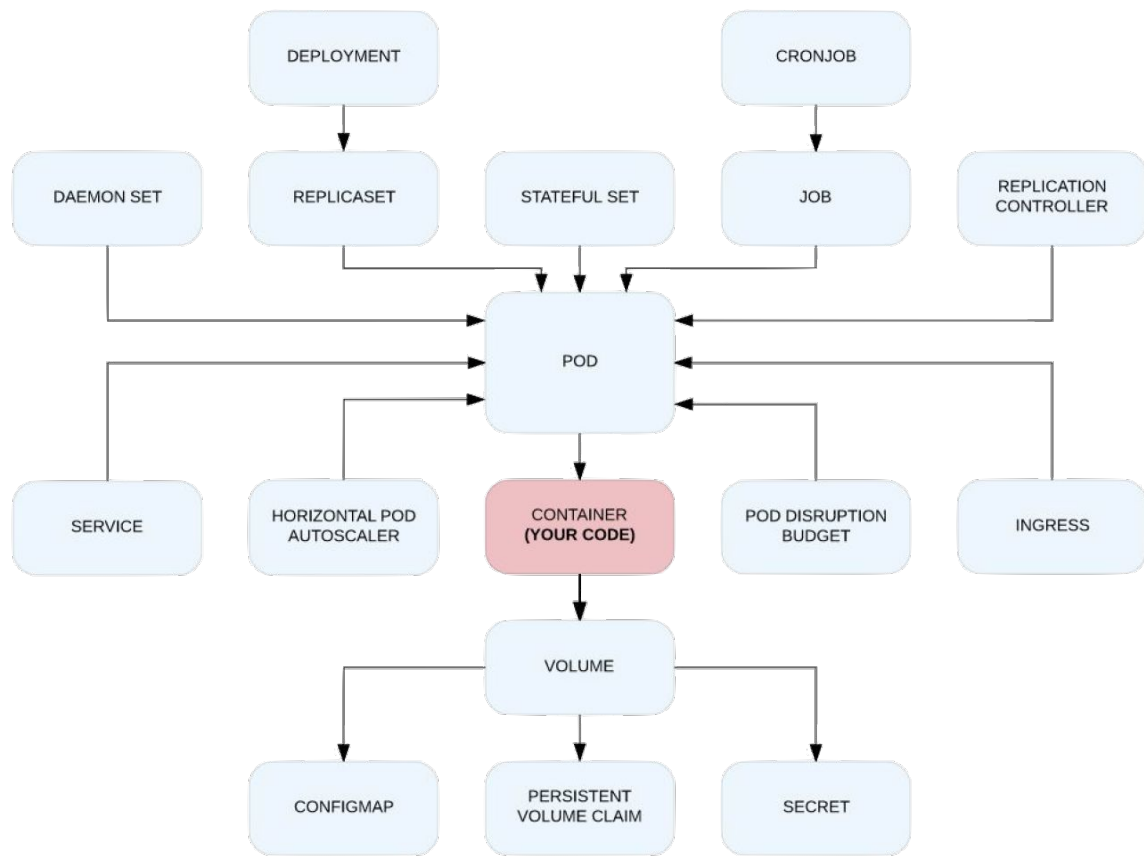


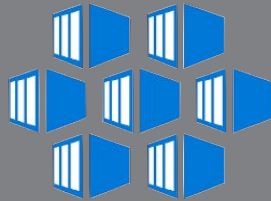
ThoughtWorks®

K8S EFFECTS

Rise of the Containers Workshop

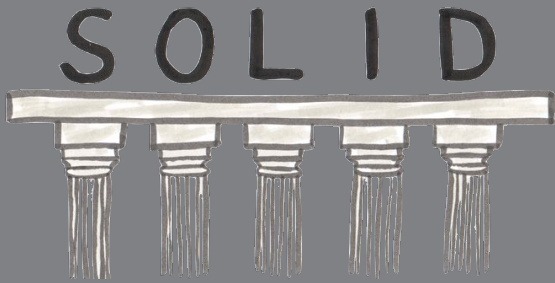
Kubernetes Objects





k8s as a Service

Kubernetes gives you the freedom to move your applications between the different cloud providers, kubernetes distributions and service providers.

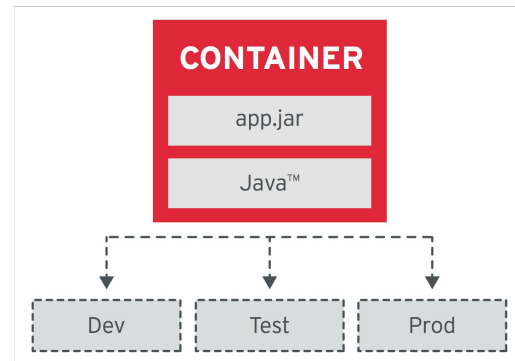
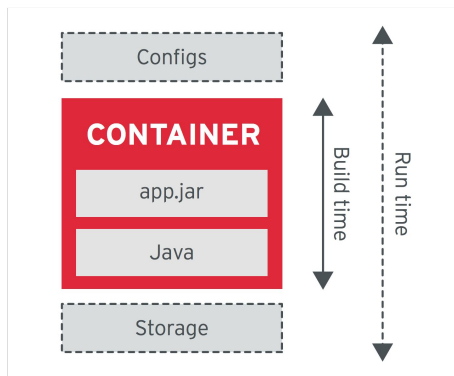
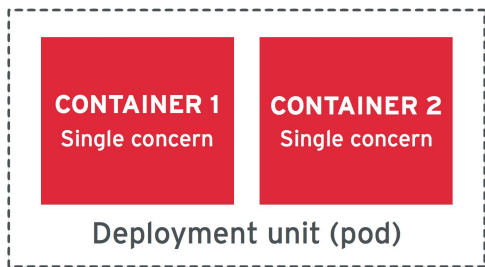


Container Design Principles

Similarly to the SOLID principles that were introduced by Robert C. Martin, which represent guidelines for writing better object-oriented software, there are also design principles for creating better-containerized applications.

<https://www.redhat.com/cms/managed-files/cl-cloud-native-container-design-whitepaper-f8808kc-201710-v3-en.pdf>

Build time design principles



SINGLE CONCERN PRINCIPLE

every container should address a single concern and do it well

SELF CONTAINMENT PRINCIPLE

a container should rely only on the presence of the Linux kernel and have additional libraries added to it at build time

IMAGE IMMUTABILITY PRINCIPLE

containerized applications are meant to be immutable, and once built are not changed between different environments

Runtime design principles



HIGH OBSERVABILITY PRINCIPLE

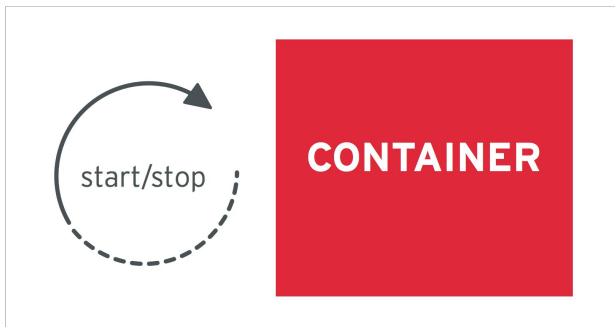
every container must implement all necessary APIs to help the platform observe and manage the application in the best way possible



LIFECYCLE CONFORMANCE PRINCIPLE

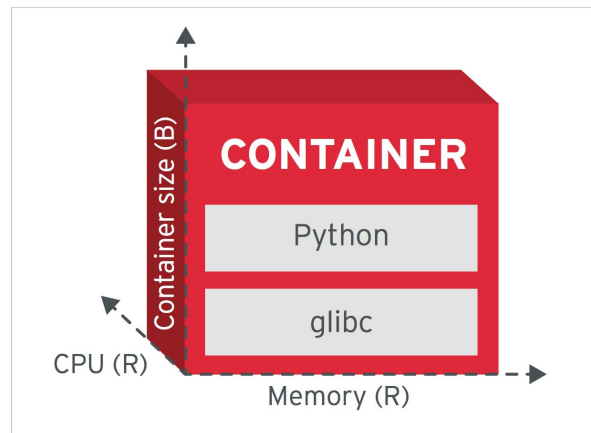
a container should have a way to read the events coming from the platform and conform by reacting to those events

Runtime design principles cont...



PROCESS DISPOSABILITY PRINCIPLE

containerized applications need to be as ephemeral as possible and ready to be replaced by another container instance at any point in time



RUNTIME CONFINEMENT PRINCIPLE

every container should declare its resource requirements and it is also important that the application stay confined to the indicated resource requirements

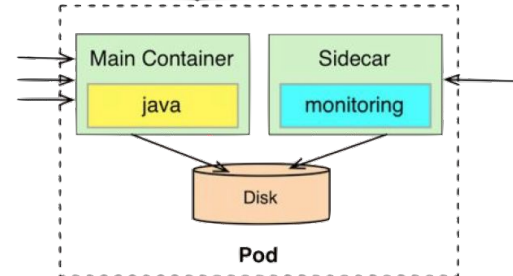
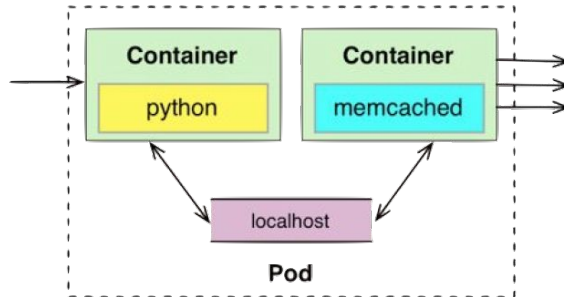
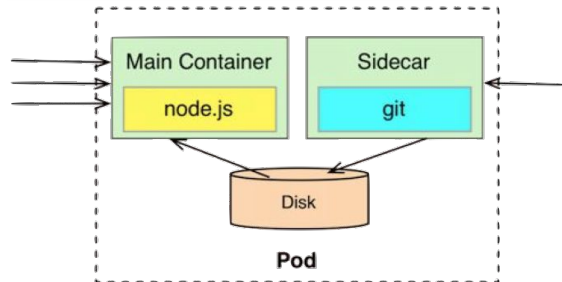
FM Factory Method							A Adapter
PT Prototype	S Singleton				CR Chain of Responsibility	CP Composite	D Decorator
AF Abstract Factory	TM Template Method	CD Command	MD Mediator	O Observer	IN Interpreter	PX Proxy	FA Facade
BU Builder	SR Strategy	MM Memento	ST State	IT Iterator	V Visitor	FL Flyweight	BR Bridge

K8S Design Patterns

Similarly to the SOLID principles that were introduced by Robert C. Martin, which represent guidelines for writing better object-oriented software, there are also design principles for creating better-containerized applications.

<https://www.redhat.com/cms/managed-files/cl-cloud-native-container-design-whitepaper-f8808kc-201710-v3-en.pdf>

Single pod multi-container patterns



SIDECAR PATTERN

a sidecar container extends and enhances the functionality of a pre-existing container without changing it

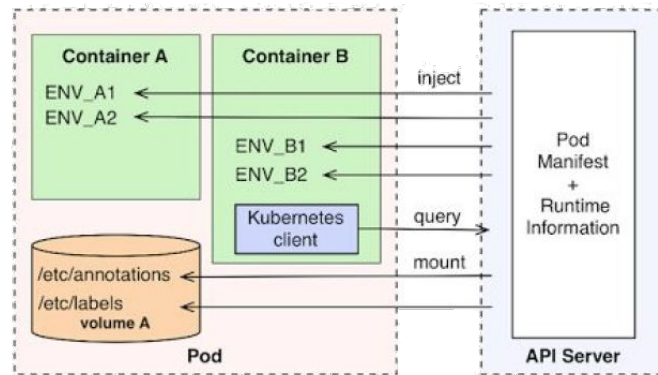
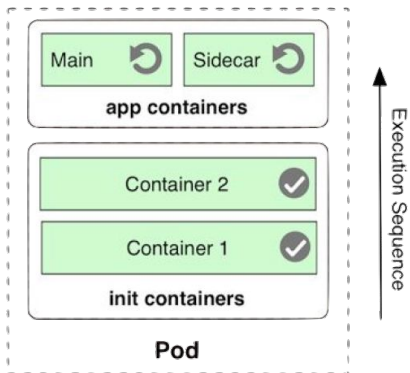
AMBASSADOR PATTERN

this pattern hides complexity and provides a unified view of the world to your container

ADAPTER PATTERN

an Adapter is kind of reverse Ambassador and provides a unified interface to a pod from the outside world

Single container patterns



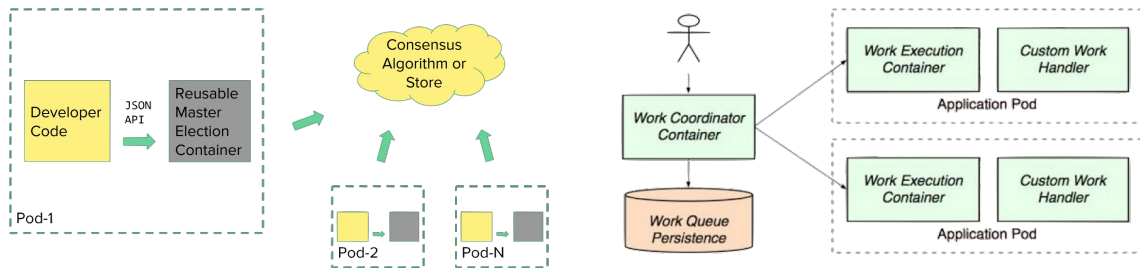
INITIALIZER PATTERN

init containers allow separation of initialization related tasks from the main application logic

SELF AWARENESS PATTERN

describes occasions where an application needs to introspect and get metadata about itself and the environment where it is running

Distributed container patterns



LEADER ELECTION PATTERN

to do ...

WORK QUEUE PATTERN

a generic work queue pattern based on containers allows taking arbitrary processing code packaged as a container, and arbitrary data, and build a complete work queue system

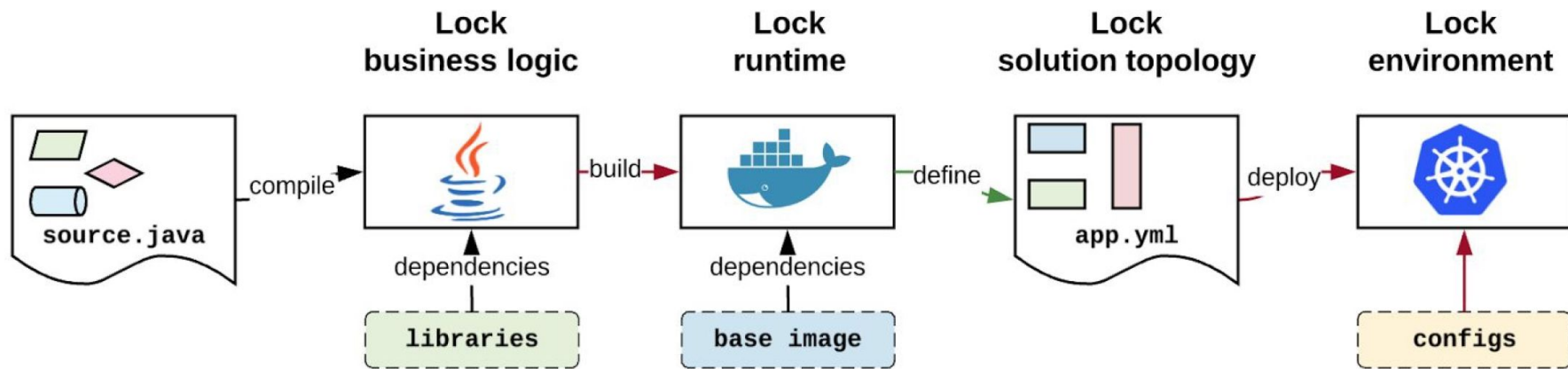
CUSTOM CONTROLLER PATTERN

a controller watches for changes to objects and act on those changes to drive the cluster to a desired state. This reconciliation pattern can be used to implement custom logic and extend the functionality of the platform

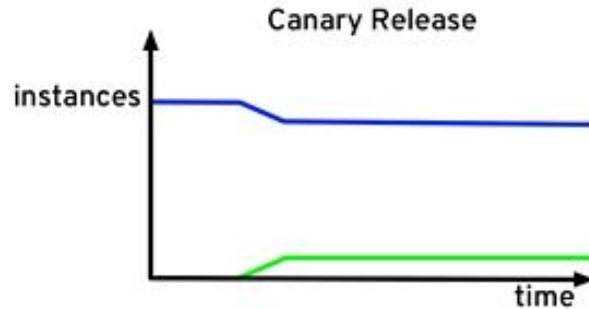
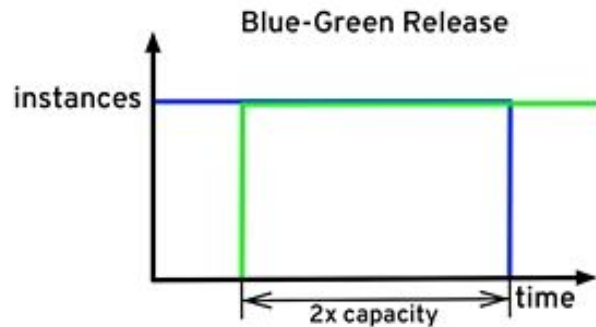
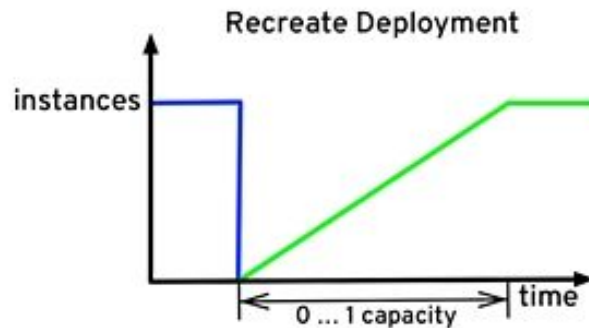
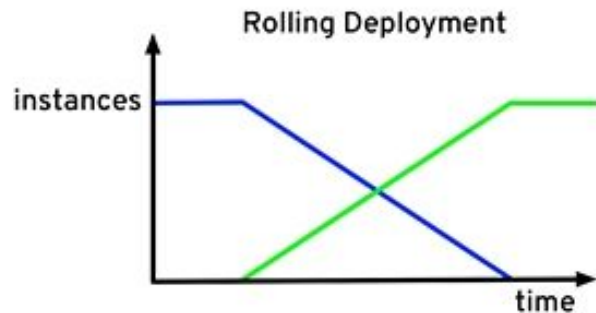
Container best Practices & Techniques

- Aim for small images
- Support arbitrary user IDs
- Mark important ports
- Use volumes for persistent data
- Set image metadata
- Synchronize host and image
- Log to STDOUT and STDERR

CI & CD with containers



K8S deployment strategies





Docker & Kubernetes
makes **infrastructure as code** easy

THANK YOU

For questions or suggestions:

Sunit Parekh

sunitp@thoughtworks.com