

# Ansible / Containers

**Pavel Slama**

**Principal Software Engineer @ Red Hat**

[pslama@redhat.com](mailto:pslama@redhat.com)

# Ansible - demo

- Running ansible playbook against freshly provisioned Vms through Vagrant
- Base CentOS 8 installation
- Ansible will install and configure Apache web server and simple site

# Ansible

- IT Infrastructure automation
  - Provisioning
  - Configuration management
  - Application deployment
- 
- Simple compared to other solutions
  - Doesn't need remote agent, just ssh/keys
  - Controls all hosts in parallel
  - Whole config stored in version control
- 
- Strong community
  - Idempotence



ANSIBLE

# Ansible - basics

- **Inventory**
  - Hosts, Variables
- **Playbooks**
  - List of tasks to be done
  - Various modules (ssh keys, package management, mounts, git, ..)
- **Templates**
  - Jinja2
  - Config or any other files, can use variables
- **Variables**
  - Structure
  - Several levels from global down to single host
- **Files**
  - Prepared files to copy to target machines

# Ansible - roles

- Reusing existing configuration (web server, firewall, monitoring,..)
- Stored in version control
- Able to reference specific version
- Ansible Galaxy – repository of existing common configurations

# Ansible - secrets

- Passwords, key files – you don't want to share
- Ansible Vault – in place encryption

# Ansible - demo

- Walk through example
- Modify inventory – add another webserver
- Add template / inventory variable
- Recover from disaster

# Ansible Tower

- Web UI for Ansible
- User management, Permissions
- API
- Job templates
- User self-service





# Summary, Q/A

## **Ansible**

- Infrastructure automation
- Provisioning, configuration
- Inventory, playbooks, roles, templates, variables

# Containers / Openshift - history

- Bare metal
- OS Virtualization
- Cloud
- Containers

# Containers

## **Bare metal**

- Hard to scale
- Ineffective HW usage (dimensed for peak load)

# Containers

## OS Virtualization

- HW abstraction
- Separated OS
- Best achievable separation
- OS overhead
- Possible different OS on host/guest
- Persistent, Stateful

# Containers

## Cloud

- Based on Virtualization
- Self service
- Billing
- Effective usage – pay for what you use, shared HW
- Management, APIs - Automation
- Hybrid cloud
- Companies able to scale to public clouds

# Containers

## Containers

- Nothing new in linux world (since 2000), but needed to perfect the separation (chroot, cgroups, namespaces, selinux)

## What is container?

- Standalone app/software package with everything it needs to run:
  - Code, libraries, tools
  - Settings – mounts, exposed network ports
- Set of separated processes running on host OS
- Lightweight
  - Almost no overhead
  - Scale very quickly

# Openshift - demo

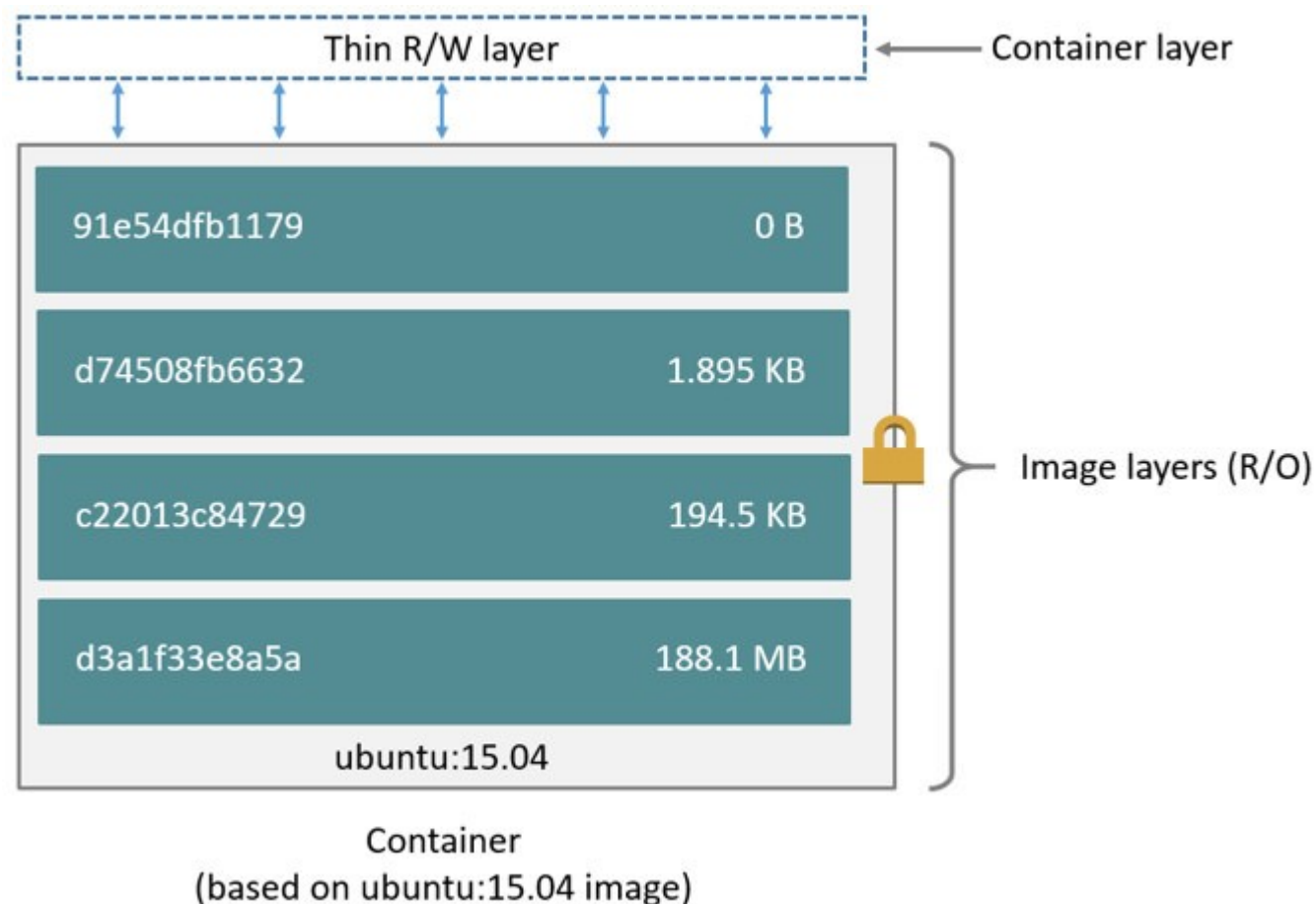
- Spin up CakePHP App from source repo



# Containers - Packaging

- Standardization – Open Container Initiative

- Layers

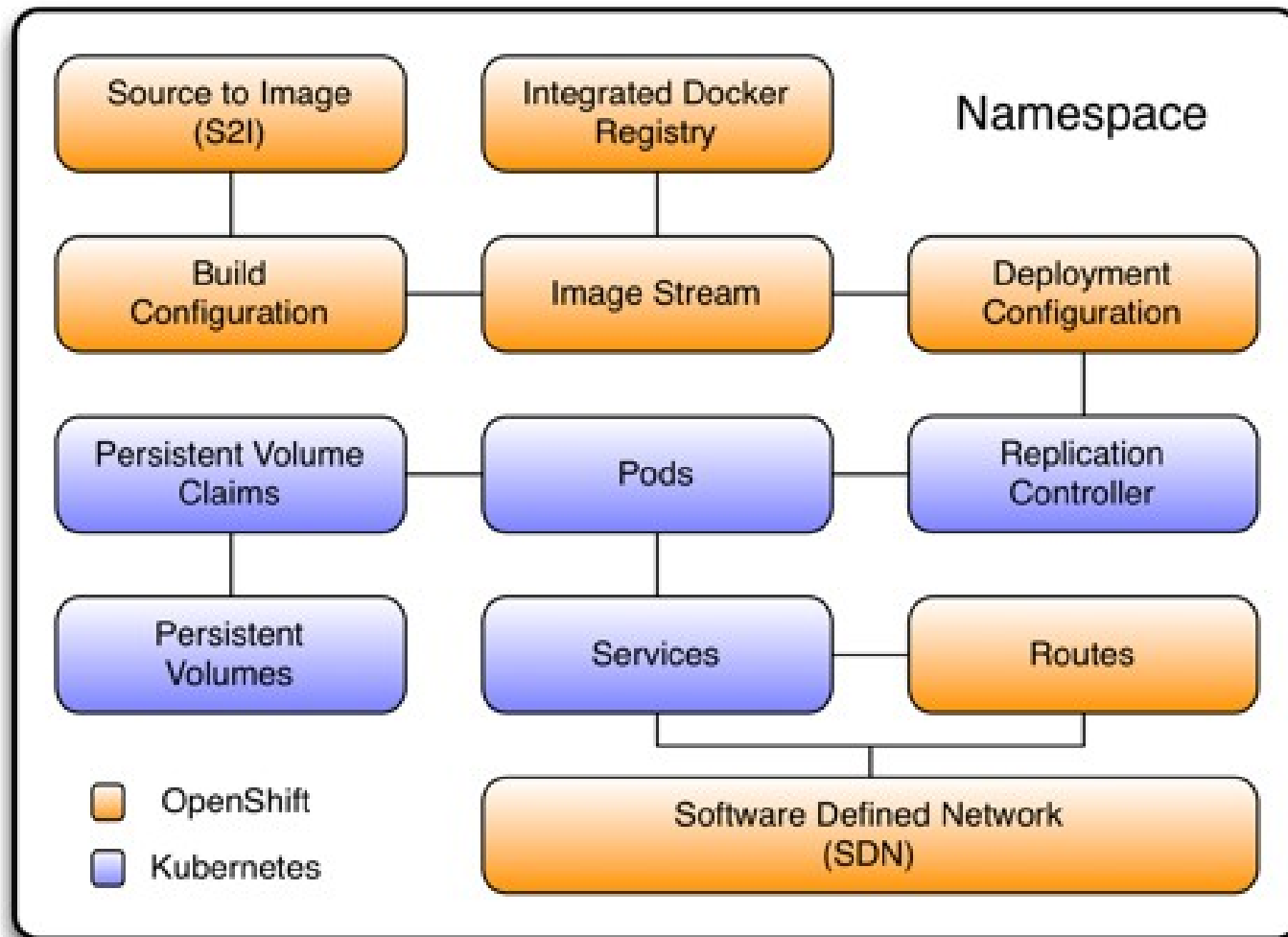




# Containers

- Different paradigm – containers are stateless
- Everything is temporary/ephemeral
- Container can be killed anytime
  - Example: DB container killed in the middle of transaction
- Declarative environment
  - Describe the world, we'll make it happen

# Openshift - Orchestration



# Openshift - S2I

- Source To Image
- Builder container that can assembly images from source code
- Can consume various languages, frameworks, packaging:
  - PHP (cake,.. )
  - Python (django, flask,..)
  - Java
  - ..
- Based on source repo contents – does the right thing to collect dependencies

# Openshift - Developers

- Openshift uses Kubernetes and add integration with developers' workflow
- Image builder, integration with source control
- Rolling updates
- Automation – hooks
- Dev → qa → prod promotion
- CI/CD
- Example – PHP App running in Openshift

# Openshift - demo

- Setup github → openshift webhook
- Update the code, let the openshift deploy automatically
- Kill running pods, let the openshift recover

# Summary, Q/A

## **Containers**

- Lightweight standalone application
- Layered packaging
- Set of separated processes and environment

## **Openshift**

- Container platform built on top of kubernetes
- Allows rapid development and deployment
- Rolling updates, self healing, CI/CD
- S2I – builds app images from source