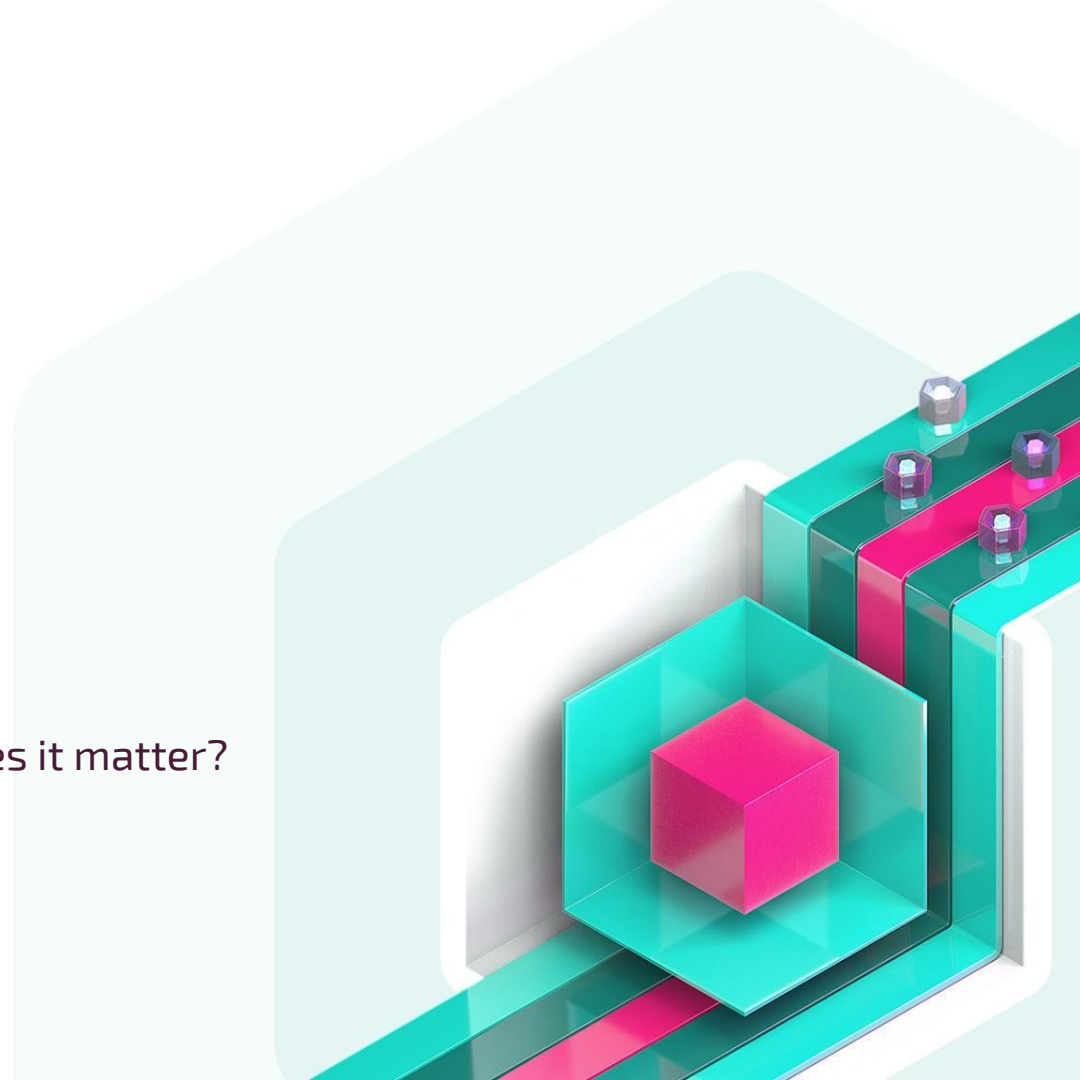




PaaS vs KaaS

What's the difference, and when does it matter?



Who am I?



Nick Chase

Head of Technical Content at Mirantis

Nick Chase is Head of Technical Content for Mirantis and a former member of the Kubernetes release team. He is a former software developer and author or co-author of more than a dozen books on various programming topics, including the *OpenStack Architecture Guide*, *Understanding OPNFV*, and *Machine Learning for Mere Mortals*.

Agenda

- KaaS and PaaS: what are they?
- PaaS strengths and weaknesses
- KaaS strengths and weaknesses
- How to decide which to use
- Q&A

This webinar is for you if...

- You make technology decisions
- You're not **necessarily** technical (but it's OK if you are)
- You're just curious

For more information

Download the comparison table from <http://bit.ly/kaas-paas-comparison>



KaaS and PaaS: What are they?

The old hierarchy

IAAS -> PAAS -> SAAS

The new hierarchy

IAAS -> KAAS -> PAAS -> SAAS

Define: Platform as a Service

A system that abstracts away hardware and software deployment, enabling developers to simply mix and match components for application creation.

Some PaaSes you may have heard of

- CloudFoundry
- OpenShift
- Google App Engine
- AWS Elastic Beanstalk
- Windows Azure
- Heroku
- Force.com
- Apache Stratos

What is a KaaS?

A system that enables developers to deploy and/or manage Kubernetes clusters in a self service manner.

Some KaaSes you may have heard of

- Rancher
- Platform9
- Mirantis KaaS
- VMware Kubernetes Management Platform
- Rackspace KaaS
- Amazon Elastic Kubernetes Service (EKS)
- Google Kubernetes Engine (GKE)
- Azure Kubernetes Service (AKS)

Telling them apart

???



PaaS strengths and weaknesses

Strength: Application Catalog

OPENSIFT CONTAINER PLATFORM Service Catalog

Search Catalog

Browse Catalog

Deploy Image Import YAML / JSON Select from Project

All Languages Databases Middleware CI/CD Other

Filter 207 Items

.NET

.NET Core

amp-apicast-wildcard-router

.NET

.NET Core + PostgreSQL (Persistent)

amp-pvc

.NET

.NET Core Example

Apache HTTP Server

.NET

.NET Core Runtime Example

Apache HTTP Server (httpd)

3scale-gateway

CakePHP + MySQL

My Projects

+ Create Project

Create Project

* Name
myfirstapp
A unique name for the project.

Display Name
My Project

Description
A short description.

Cancel Create

Recently Viewed

Weakness: Application Catalog

OPENSIFT CONTAINER PLATFORM Service Catalog

Search Catalog

Browse Catalog

Deploy Image Import YAML / JSON Select from Project

All Languages Databases Middleware CI/CD Other

Filter 207 Items

.NET

.NET Core

amp-apicast-wildcard-router

.NET

.NET Core + PostgreSQL (Persistent)

amp-pvc

.NET

.NET Core Example

Apache HTTP Server

.NET

.NET Core Runtime Example

Apache HTTP Server (httpd)

3scale-gateway

CakePHP + MySQL

My Projects [+ Create Project](#)

Create Project

* Name
myfirstapp
A unique name for the project.

Display Name
My Project

Description
A short description.

Cancel Create

Recently Viewed



Strength: Simplified Developer Experience

CONTAINER PL

Search

databases

.NET C

Deploy Image

Image

Results

1

2

* Add to Project

myfirstapp

Deploy an existing image from an image stream tag or image registry.

☐ Image Stream Tag

Namespace / Image Stream : Tag

☒ Image Name

redhatworkshops/welcome-php

To deploy an image from a private repository, you must [create an image pull secret](#) with your image registry credentials. [Learn More](#)

redhatworkshops/welcome-php

2 years ago, 175.9 MiB, 5 layers

17

Strength: Operator control

- Operator makes architectural decisions for the PaaS
- Operator installs the PaaS
- Operator updates the PaaS
- Operator upgrades the PaaS
- Operator controls access to the PaaS

Weakness: Operator control

- Developers are reliant on operators for all those things

Strength: Self contained architecture

- Typically comes with all the pieces-parts
- Application catalog is pre-installed
- May include CI/CD infrastructure, IAM, etc.

Weakness: Self contained architecture

- Can't choose your own pieces-parts
- Dependant on how all those pieces are installed/fit together

Strength: Built in security

- May include built in security policies/restrictions

Weakness: Built in security

- May include built in security policies/restrictions
- Lack of control over those security policies/restrictions



KaaS strengths and weaknesses

Strength: Developer control

- Developers can choose what applications they want
- Developers can spin up what resources they need
- Developers can upgrade/update their clusters
- Developers can control security

Weakness: Developer control

- Developers can choose what applications they want
- Developers can spin up what resources they need
- Developers can upgrade/update their clusters
- Developers can control security

Strength: More flexible scalability

- A small cluster can be scaled on small machines
- No need to scale the entire system for one tenant

Weakness: More flexible scalability

- Cluster sprawl is a real possibility

Easier upgrades/updates

- Developers are in control
- KaaS can handle the process
- Only single clusters are affected

Strength: Vendor neutrality

- Not reliant on any proprietary/system specific API
- Can be migrated to any other Kubernetes system/provider
- Can be used with multiple Kubernetes providers

Strength: Multicloud apps

- Multiple clusters is the very definition, after all!
- Can create clusters on multiple clouds/providers, even without those providers knowing anything about KaaS



How to decide which to use

Questions to ask yourself

- What are you trying to accomplish with your automation?

Questions to ask yourself

- What are you trying to accomplish with your automation?
- How much of a guardrail do your developers need/want?

Questions to ask yourself

- What are you trying to accomplish with your automation?
- How much of a guardrail do your developers need/want?
- How are you going to handle upgrades/updates?

Questions to ask yourself

- What are you trying to accomplish with your automation?
- How much of a guardrail do your developers need/want?
- How are you going to handle upgrades/updates?
- How many tenants/applications do you have?

Questions to ask yourself

- What are you trying to accomplish with your automation?
- How much of a guardrail do your developers need/want?
- How are you going to handle upgrades/updates?
- How many tenants/applications do you have?
- How much flexibility do you need in choosing your architecture?

Questions to ask yourself

- What are you trying to accomplish with your automation?
- How much of a guardrail do your developers need/want?
- How are you going to handle upgrades/updates?
- How many tenants/applications do you have?
- How much flexibility do you need in choosing your architecture?
- Do you anticipate needing different clouds? Multiple clouds?

Questions to ask yourself

- What are you trying to accomplish with your automation?
- How much of a guardrail do your developers need/want?
- How are you going to handle upgrades/updates?
- How many tenants/applications do you have?
- How much flexibility do you need in choosing your architecture?
- Do you anticipate needing different clouds? Multiple clouds?
- How predictable are your plans?



THANK YOU

Download a KaaS vs. PaaS comparison table
<http://bit.ly/kaas-paas-comparison>

Includes Mirantis KaaS, Red Hat OpenShift,
Rancher, Canonical and Pivotal