AUTHX

Java PaaS – Enabling CI, CD, and DevOps

AuthX Overview

Who We Are?

- Digital Engagement Company offering Technical and Marketing Services with proven success supporting Fortune 1000 companies.
- We partner with industry leading CXM technology companies to offer our clients a tailored strategy that complements their stage of the digital marketing journey.
  - Technical Strategy and Consulting
  - Software Implementation
  - Optimization and Managed Services
- We have a passion for Authenticity – for our clients and their customers, our employees and our partners.

Why We're Different?

- We are bridge builders – helping our clients focus needs on the business
- Our vision is focused on what's next for our clients (not now)
- Flexible engagement style

Who am I?

Director/Sr. Cloud Architect with AuthX Consulting (www.authxconsulting.com)
- In charge of our Managed Cloud Services offerings

Most Recent Past: Solutions Architect with CloudBees
- Spoke at OpenStack Summit: Dynamic Jenkins Slaves in OpenStack (jclouds and Rackspace)

Early Adopter and Fast Follower of Java, Web, and Cloud technologies

LinkedIn: http://www.linkedin.com/in/iamjimmyray

Blog: http://www.techsand.com/ Avoiding Tech-sand

Today's Agenda

- Starting Point
- PaaS
- CI
- CD
- DevOps
- PaaS Futures?

Our Starting Point

- CI underpinned by Jenkins OSS and Jenkins Enterprise (JE) from CloudBees
- Java PaaS underpinned primarily by CloudBees DEV@cloud and RUN@cloud (for now, at least)
- Traditional IT compute resources provisioning is slow, and wasteful.
- There are tradeoffs with moving to the cloud. PaaS is not a panacea for poorly performing IT.
- If you are not using cloud technologies, you are behind.

PaaS

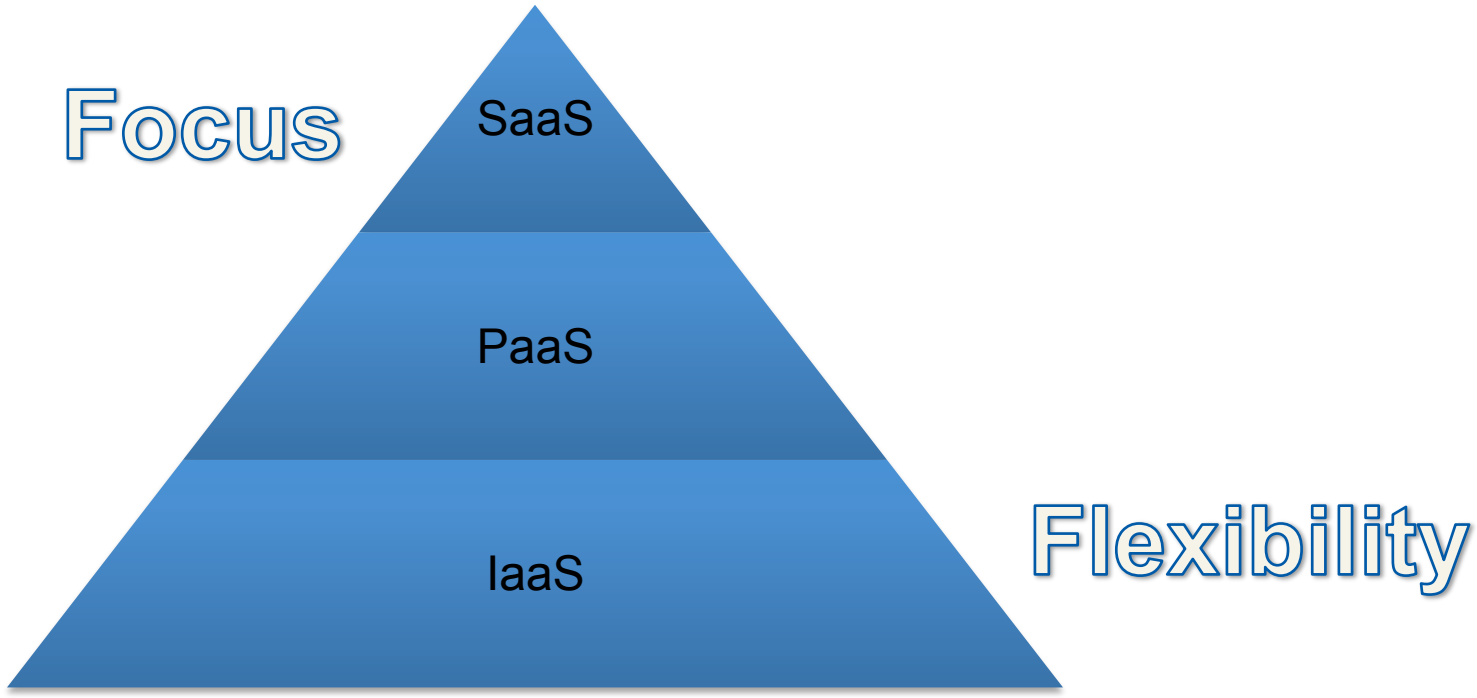
Platform as a Service, According to NIST:

“The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.”

PaaS

- Platforms, running on IaaS, providing a solution stack (Tomcat, MySQL, etc.)
  - No longer need platform experts
- Customer uses platform to develop and/or run their applications.
  - Less customization than IaaS, more than SaaS
- Flexibility, Adaptability, Remote Collaboration
- Customers do not install the platform software or manage the servers.
- Not as flexible as IaaS, not as narrowly focused as SaaS

PaaS vs. IaaS vs. SaaS



Java PaaS Expectations

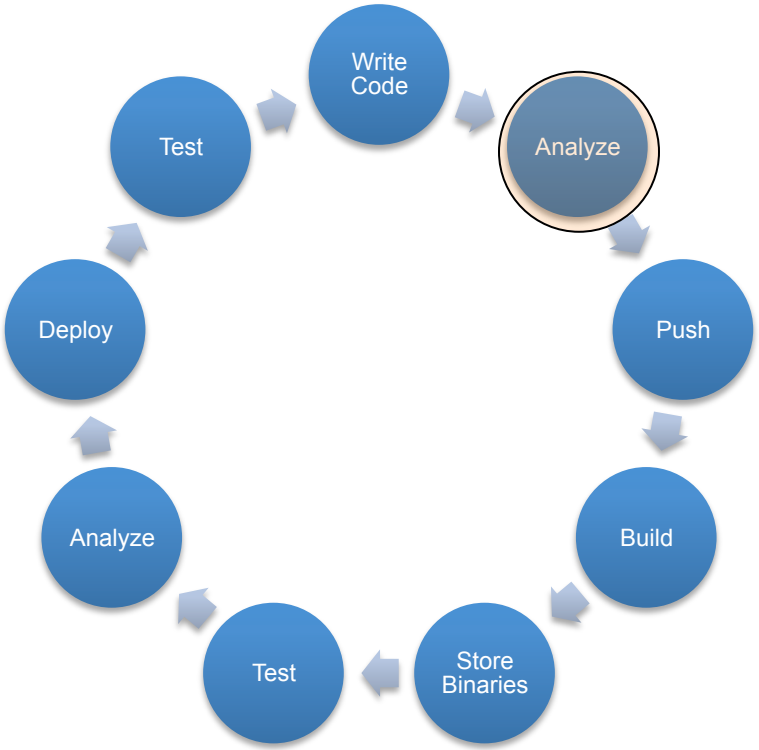
- PaaS is NOT IaaS
- You will not get access to SSH into your servers
- You may manage your applications, and application container sizes
- You may subscribe to tools and services, but you will not manage the application stacks or tools.
- Customization may be possible, via a web UI, an API/SDK, or even via support tickets (ie: SSL routers, etc.)

Continuous Integration

- Development practice and a set of patterns
- Developers integrate (check-in, merge, commit, push, etc.) their code early and often
  - Multiple times a day
- Code is then verified by automated unit testing and static analysis
- End result is early success or failure
- The foundation of good agile software development.

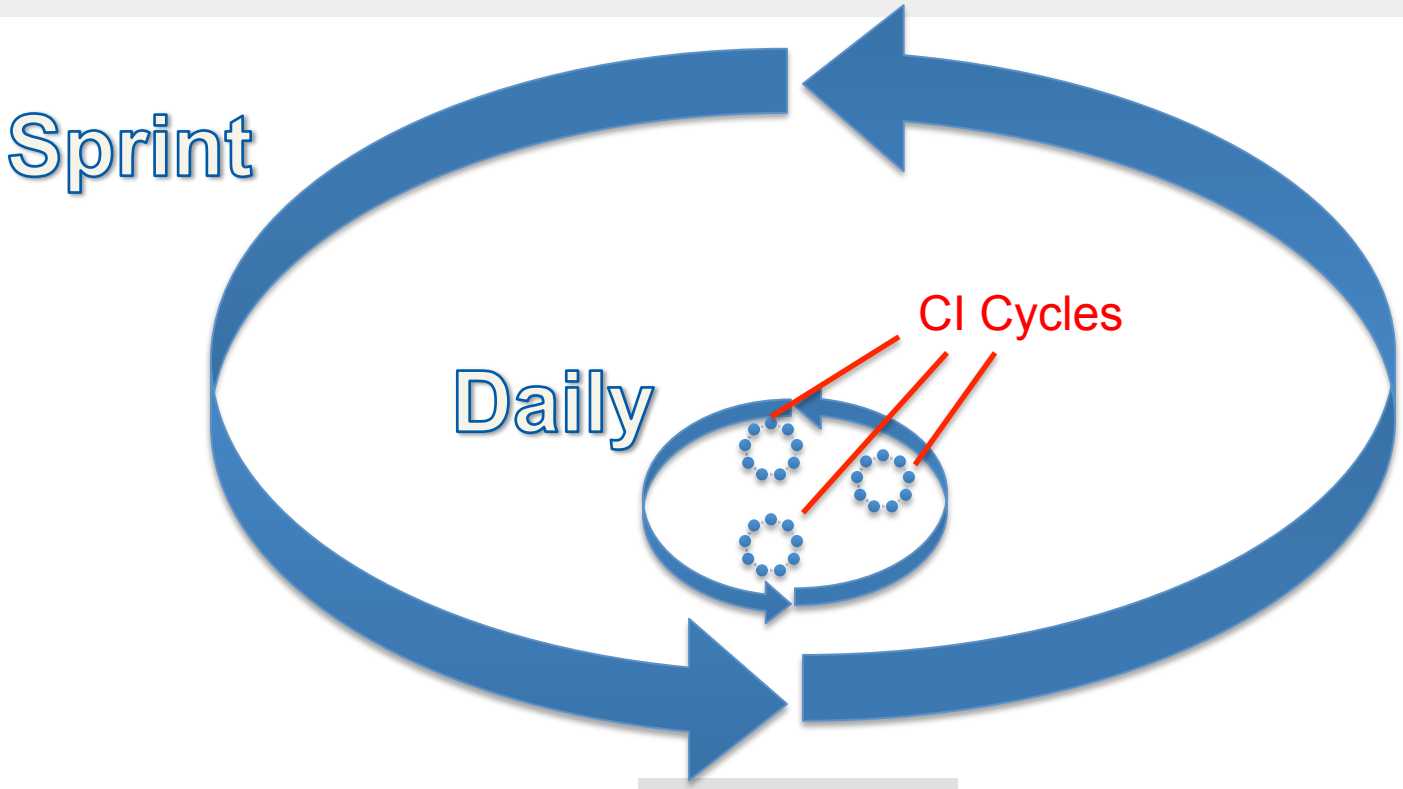
"Do not fear mistakes. There are none." - *Miles Davis*

CI Process



AUTHX

Agile Software Cycle



Demo

- CI with Java PaaS
- CloudBees Java PaaS
  - http://www.cloudbees.com/

Why to Choose PaaS for CI

- PaaS vendor will manage CI toolset for you
- You concentrate on being productive, and not installing, administering, and updating CI toolset

- Alternatives:
  - Run CI tools (all those tools) in IaaS cloud, or on premise, or use hybrid approach.

- Trade-offs:
  - Giving up control for simplicity

Why not to Choose PaaS for CI

- Security – It simply may not match your needs.
- Data Compliance (check with your cloud vendors, each layer)
  - PII
  - PHI (HIPAA Business Associate or conduit)
  - PCI
- Integration to existing resources
- Look into on-premise or hybrid solutions

PaaS Hybrid CI Solutions

“I would like to use a PaaS solution, and let someone else manage my CI tools, but I need for my builds to run locally within my firewall.”

- Use a hybrid solution: JE in CloudBees DEV@cloud, with On-Premise Executors (OPE)
  - OPEs run locally on JE slaves. They require JE in DEV@cloud
  - The OPE slave initiates the connection to the JE server in DEV@cloud. So, there is no need for static IPs on the PaaS side and IP whitelisting on premise.
  - The OPE slave is a Jenkins CLI slave, using the jenkins-cli.jar
  - **Note: If your builds will be triggered from SCM check-ins/pushes, CI running in in the cloud must still be able to connect to that SCM to poll/receive those triggers**.

PaaS Cloud Bursting

“I would like to have an on premise CI solution with the ability to burst builds in to cloud as needed.”

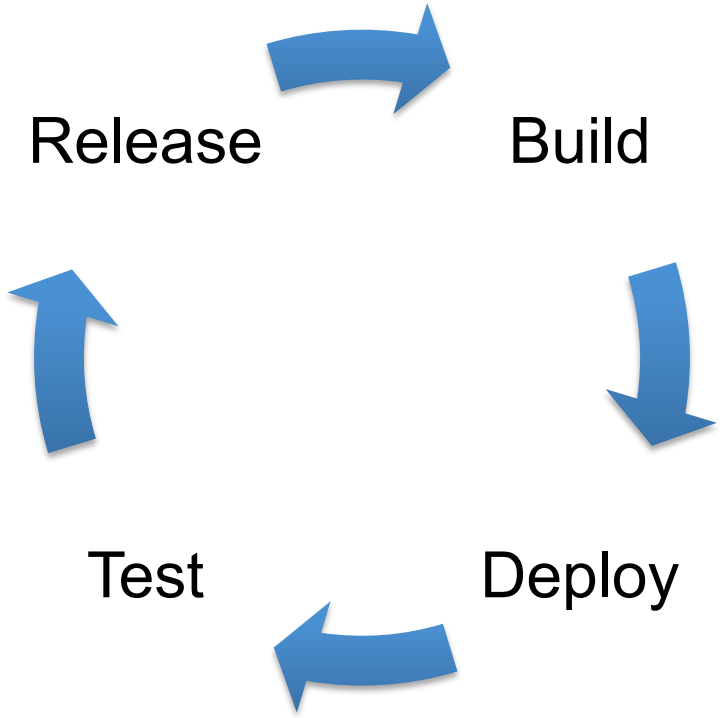
- Use the CloudBees Cloud Connector with JE on-premise
- Use the Jenkins jclouds plugin to push slaves to Rackspace, AWS, etc.

Continuous Delivery

- A set of applied patterns, for delivering software to multiple environments.
- Defines the repeatable activities and orchestrations
- The idea is to reduce the size of deployments, while increasing the frequency.
- Underpinned by the same practices in CI, Agile, and Lean
  - Like CI, CD is designed to reduce uncertainty, and waste

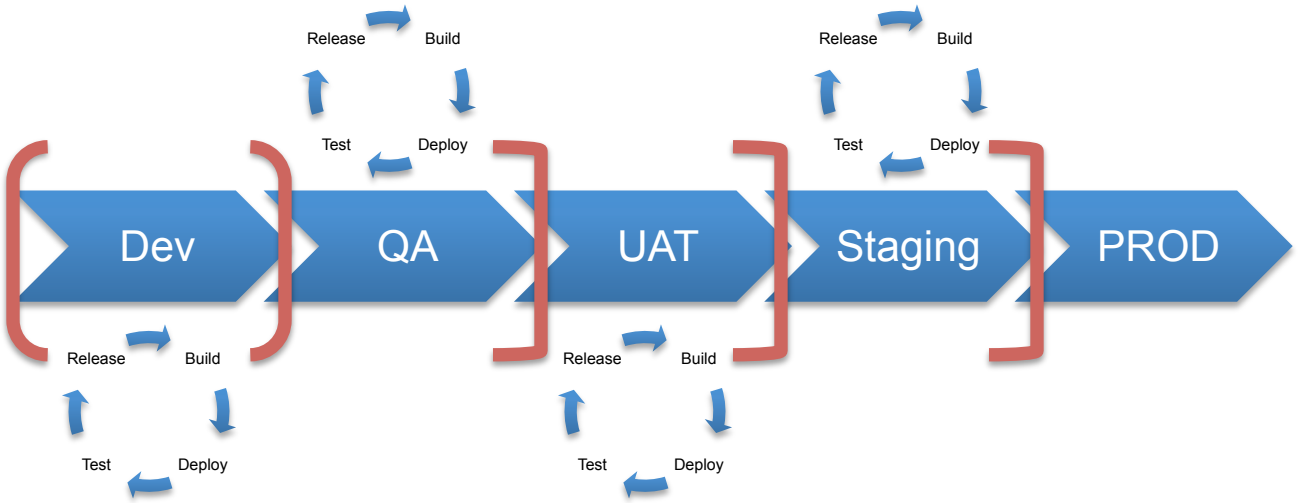
Faster, Better, Cheaper – *CloudBees 2014*

Continuous Delivery



Continuous Delivery

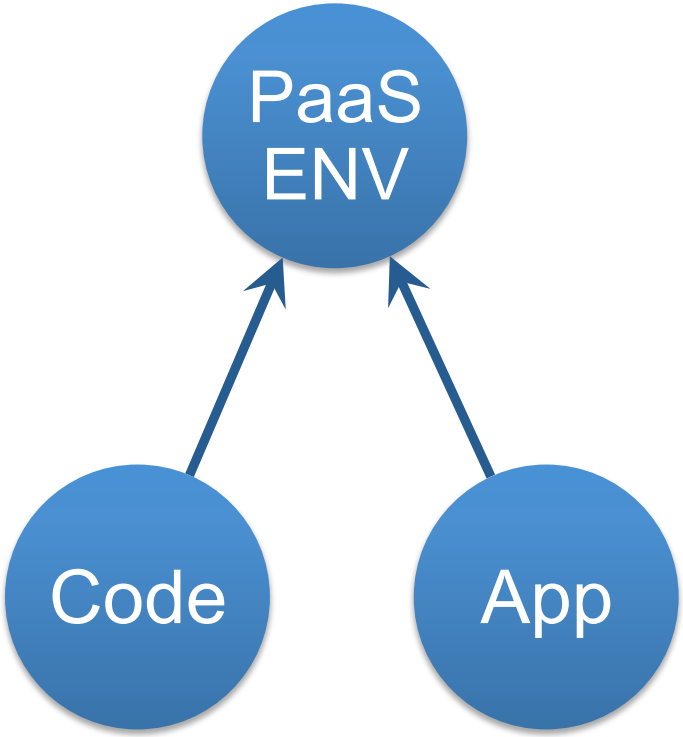
- Example Environment Progression



PaaS Environments

- Environments in PaaS take on a new meaning.
- They are no longer static environments waiting for applications to be deposited and configured within them.
- The environment now becomes the logical intersection between the codebase (branch/tag) and the application declaration.
- With RUN@cloud, It is as simple as copying an existing Jenkins job, changing the name, and changing the application ID.**

PaaS Environments



Demo

- CD with Java PaaS
- Java PaaS Environments in CloudBees

PaaS Landscape Major Change

- CloudBees is dropping RUN@cloud, their PaaS Runtime environment.
  - Announced 2014-09-11
  - They became “The Enterprise Jenkins Company”
  - http://www.cloudbees.com/faq-customers-runcloud
  - Migration: https://docs.google.com/document/d/1M1gqvtHXZh_YqEaRKJgk9Lv4spy5pcauMAGNL3uL8qM/edit

No More RUN@cloud

- What does this mean for the PaaS market?
  - For the best solution for Jenkins-based PaaS CI, I would still use CloudBees
  - For PaaS runtime deployments, there are other options (Heroku, OpenShift, Engine Yard, AWS EC2, AWS Elastic Beanstalk, Jelastic, Google Cloud Services)
  - Jenkins integration with CloudBees On-Premise Executors, jclouds, Xebia Labs DeployIt, AWS Deployer, etc.

Why Drop PaaS Runtime?

- The lines are now blurred between IaaS/PaaS/SaaS
  - PaaS and SaaS vendors now provide customizations via APIs, SDKs
  - IaaS vendors are also offering PaaS-like services
- Development in the cloud is not the same for all projects and all technologies.
- Cloud services are now, more than ever, commoditized, and spread across the IaaS/PaaS/SaaS landscape.
  - à la carte is the new consumption model

AWS Elastic Beanstalk as PaaS

- Supports multiple platforms: IIS, Node.js, PHP, Python, Ruby, Tomcat, Docker
- Don’t immediately see what you need…consider using Docker
  - In fact, Docker might be the PaaS Runtime killer
  - Consider what PaaS vendors are already doing in the background with Puppet and Chef.

- Trade-offs
  - Users must create the AWS Beanstalk application first, before CloudBees Jenkins can deploy to it.

Demo

- Blue-Green Deployments with CloudBees SDK – While it Lasts!!

DevOps

- A collaborative approach that combines folks that write software, with folks that provide secured, reliable, and scalable environments and tools.
- It is meant to break down traditional barriers, while allowing developers to learn from operations and vice-versa.
- They perform very different tasks and activities
  - Sometimes at odds with each other
  - BUT…they should have the same goals…agility.
- Lines are sometimes blurred with Configuration as Code (Puppet, Chef, Docker, etc.)

DevOps Barriers

- Developers are measured by feature delivery, and defects remediated.
- Operations is measured by system stability, security, reliability, availability, etc.
- So, the barriers are organically built by competing concerns.

- PaaS can help erase those barriers, but PaaS is notorious for their lack of DevOps tools.

DevOps Past


It worked fine in Dev and Test, it's an Ops problem now.
We need to upgrade our servers. It should have no affect on the applications.

PaaS & DevOps

- PaaS reduces provisioning tasks load on Ops folks, more so than IaaS
- PaaS provides stacks and tools for quick compute resource provisioning
- SaaS provides tools for Ops concerns, such as sever monitoring, security, and log analysis
  - You can also use other SaaS solutions (CopperEgg, Nagios, etc.)
- PaaS provides tools to make developers more productive without placing a huge burden on finances or human resources

- **Lessen the impacts of Dev and Ops incidents**

Private PaaS

- Not all PaaS solutions are private.
- Apprenda, Stackato, Stratalux, offer private PaaS solutions.

Alternatives to PaaS

- Managing iron
  - Maybe cost effective for larger companies, but barrier to business for smaller ones.
- Virtualized Servers
- Private Clouds (Rackspace, Helion,…OpenStack)
- Vagrant
- IaaS
- Configuration as Code
- Docker
- AWS OpsWorks

- PaaS and IaaS ARE NOT mutually exclusive.

PaaS for DevOps – VisualOps

- VisualOps is a promising tool – Currently Free
- Shares similar goals as Cloud Formation in terms of templating AWS infrastructure.
- Different than OpsWorks (AWS)
  - VisualOps is more of a top-down approach, stressing architecture first, then AWS infrastructure
- Users must provide sensitive info about AWS account to use this PaaS offering.
- Users need to know networking and AWS IaaS
- Demo

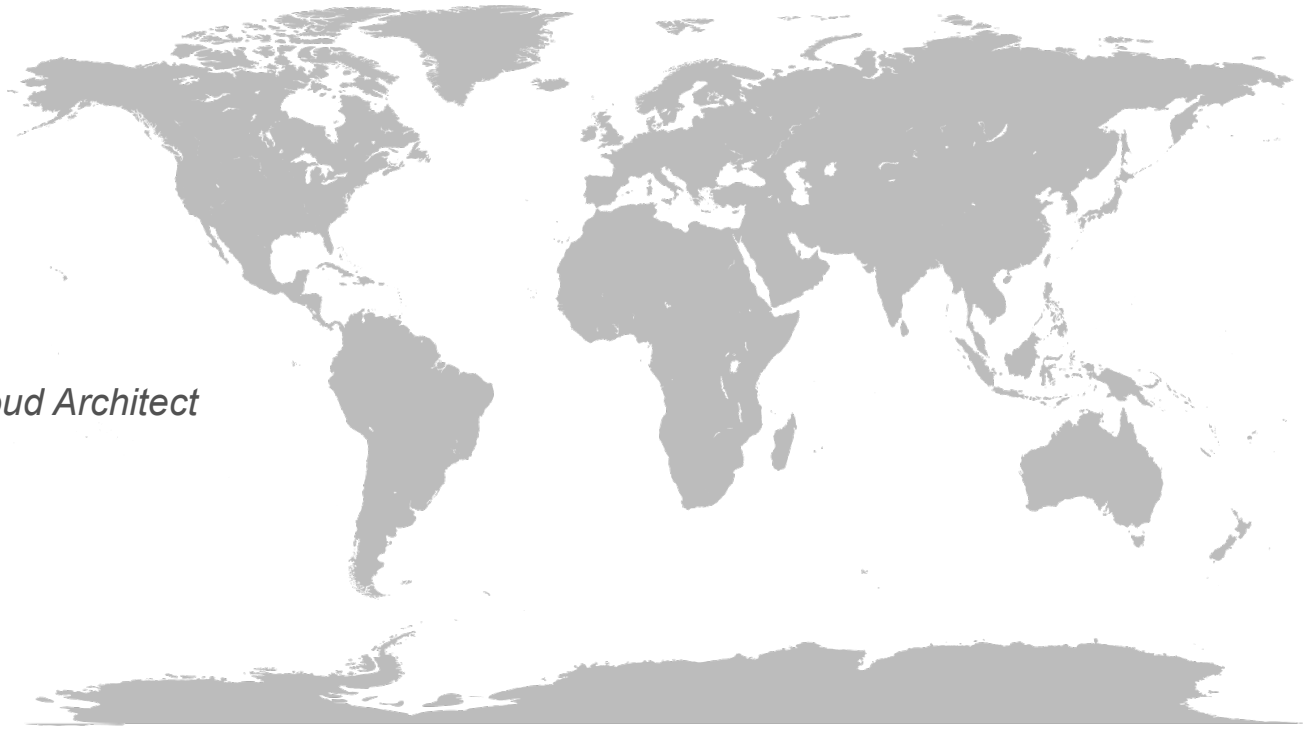
Selecting a PaaS Vendor

- How do they handle density vs. isolation?
  - LXC, ZFS, Docker?
- How do they handle security? Compliance?
- Features?
- Supported Frameworks and Languages?
- Availability?
- Extensibility (APIs, SDKs, web UI)?
- What IaaS or other infrastructure are they using?
- Can you avoid PaaS or stack lock-in?
- What changes to my application stack will I need to make?

Questions

AUTHX

Thank **you.**



Jimmy Ray, *Director/Sr. Cloud Architect*

jray@authxconsulting.com
@authxconsulting

AUTHX