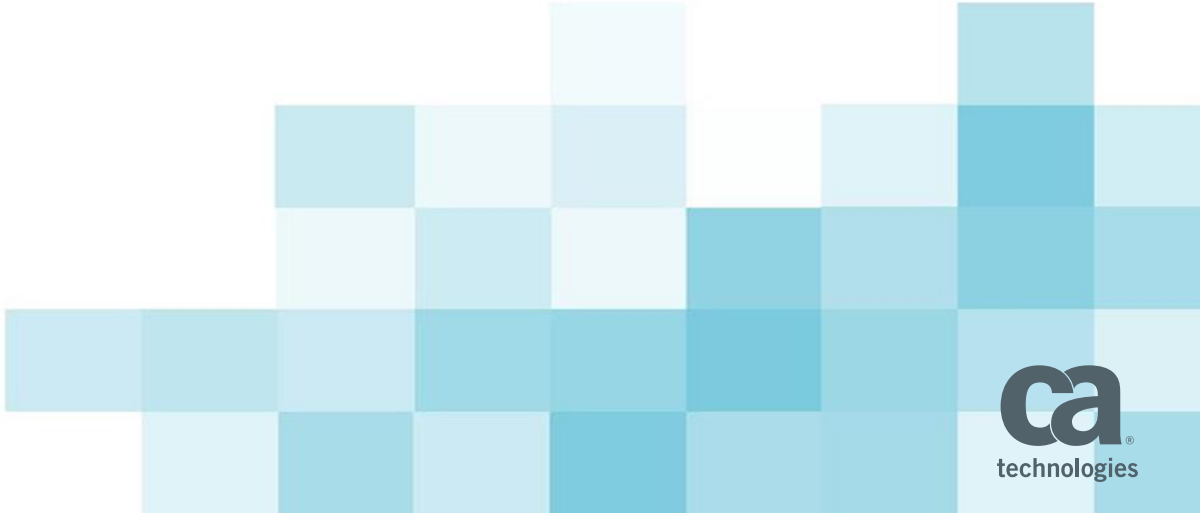
Project Whetstone Status Update

Use cloud native technology to
deploy in minutes and operate at scale

29 Apr 2016



ca technologies

For Informational Purposes Only

Terms of this Presentation

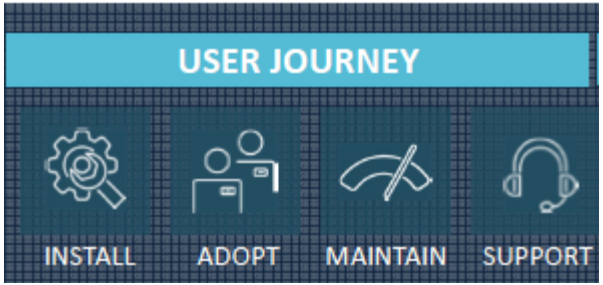
Certain information in this presentation may outline CA's general product direction. This presentation shall not serve to (i) affect the rights and/or obligations of CA or its licensees under any existing or future license agreement or services agreement relating to any CA software product; or (ii) amend any product documentation or specifications for any CA software product. This presentation is based on current information and resource allocations as of January 1, 2016, and is subject to change or withdrawal by CA at any time without notice. The development, release and timing of any features or functionality described in this presentation remain at CA's sole discretion.

Notwithstanding anything in this presentation to the contrary, upon the general availability of any future CA product release referenced in this presentation, CA may make such release available to new licensees in the form of a regularly scheduled major product release. Such release may be made available to licensees of the product who are active subscribers to CA maintenance and support, on a when and if-available basis. The information in this presentation is not deemed to be incorporated into any contract.

Agenda

1. GOALS
2. DELIVERY MODEL
3. USE CASES
4. STATUS OF PRODUCTS
5. TECHNICAL PREVIEW
6. SUMMARY

The User Journey



- CA is traditionally good at selling (The **Buyer Journey**)
- CA is traditionally poor at CX (The **User Journey**)
  - Install, Adopt, Maintain, Support
- Key **User Journey** challenges we have to improve for CX:
  - ***Large platform support matrix***: hard for engineering to test, hard to support
  - ***Complex installations***: difficult for customers to deploy & maintain
    - **Limits new NCV opportunities in channel markets**
  - ***Fragile upgrade process***: every deployment environment is different

Containers and The User Journey

- **Reduced support matrix (SSO has 35+):** Use shared base containers aligned across products
- **Container platform (Kubernetes):** Run anywhere, reducing complexity
- **Cloud & private data center ready**: AWS, Azure, Google, VMware
- **No installations, only downloads:** What we build and test is what the customer runs
- **Separation of data/config from binaries:** Easier to debug. Easier to support. Easier to patch
- **Central logging capability:** Kubernetes has built in Elasticsearch, Logstash, Kibana (ELK) support
- **Remote Engineer:** Containerized component can be built for all products to use
- **Rolling Upgrade/Rollback support:** Kubernetes natively provides this capability

So I licensed a CA Security product

How do I use it?

Infrastructure

Choose where you run your products: on-premises or cloud.

Configuration

Fill in the blanks for required settings.

Structured configuration file* for all the advanced options.

Fully extensible for custom data and modules.

Runtime

High Availability is pre-configured.

Elastic scale-up and scale-down is built-in.

Rolling upgrade* is built-in.

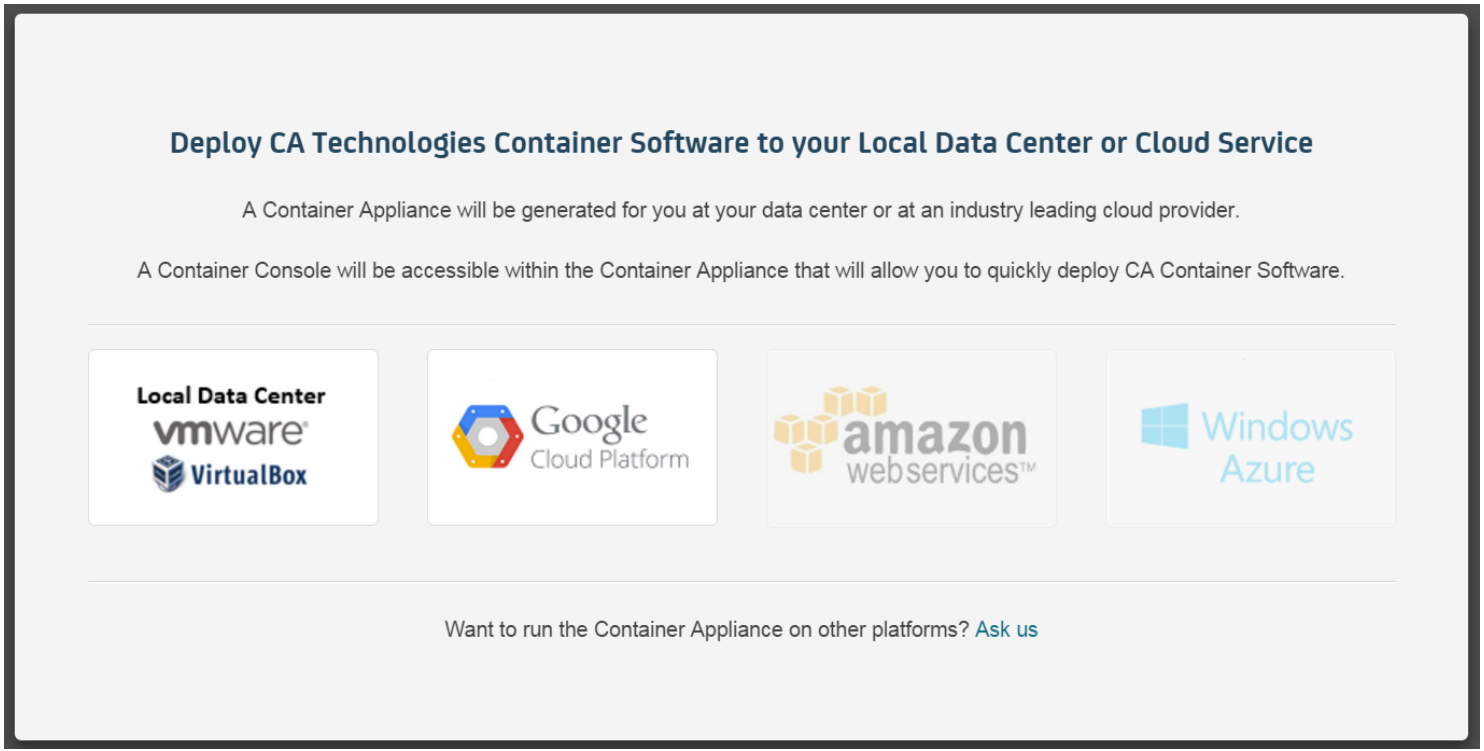
Public services are exposed on known ports.

Integrates with existing systems.

* We're working on this right now

Infrastructure

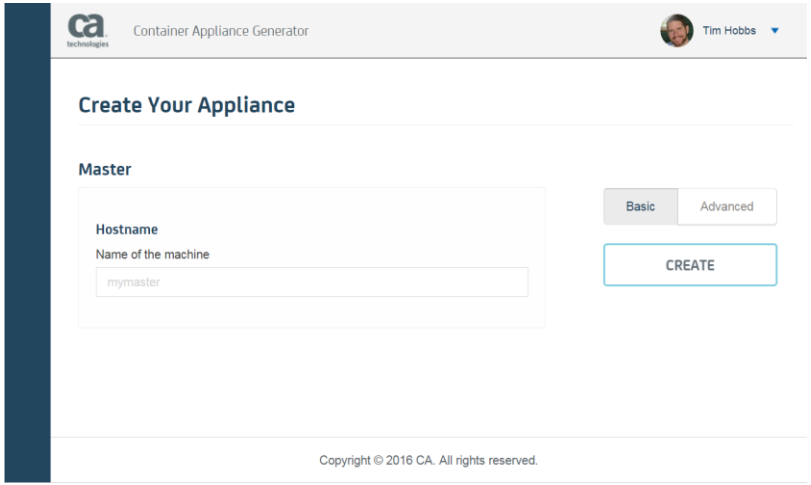
Same functionality, multiple infrastructures

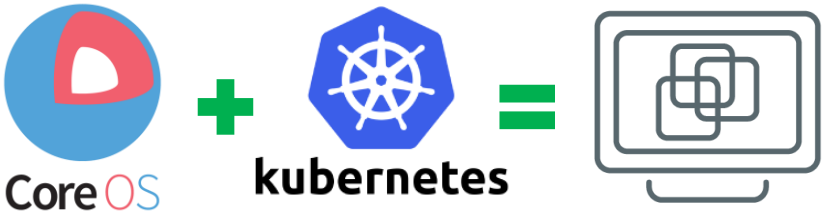


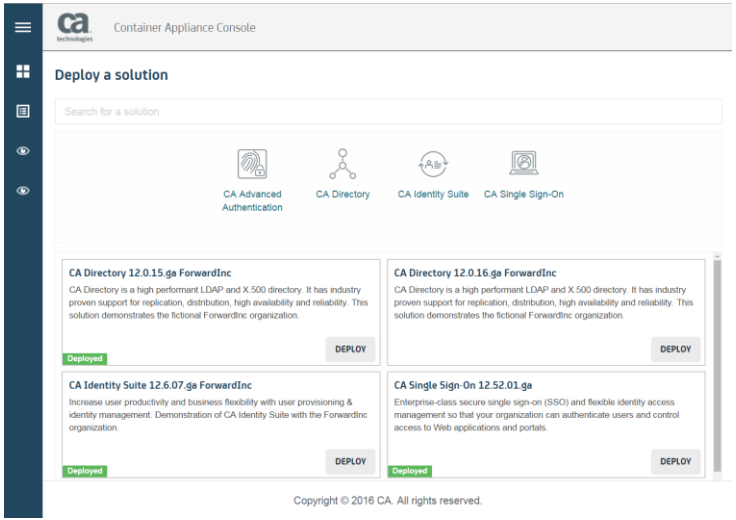
ca technologies

Infrastructure

When you prefer virtual machines…

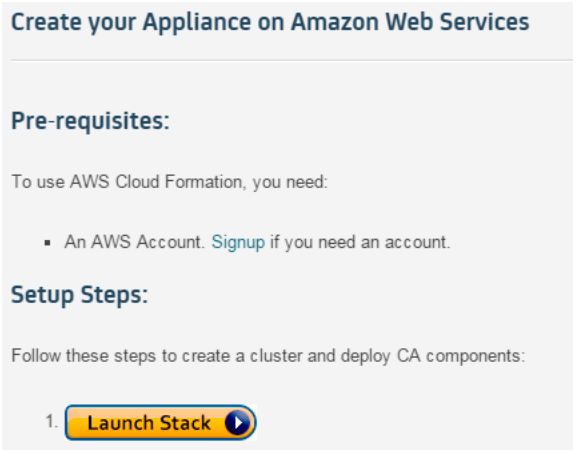


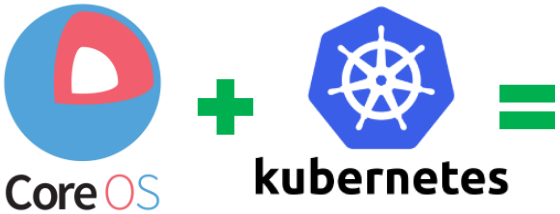


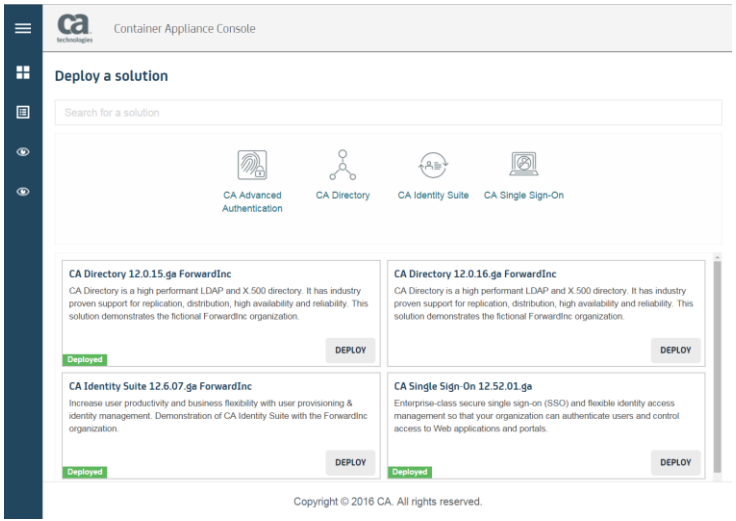


Infrastructure

When you prefer the Amazon cloud…







* We're working on this right now

Infrastructure

When you prefer the Google cloud…

Create your Appliance on Google Container Engine (GKE)

Pre-requisites:

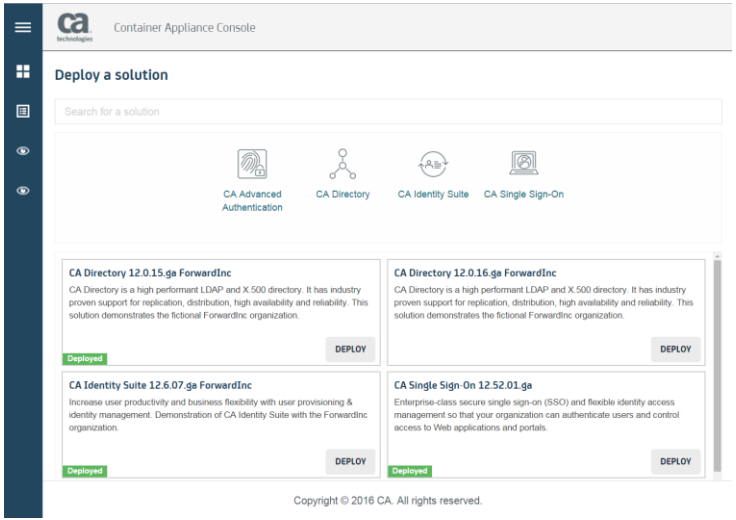
To access Google Container Engine, you need:

- A Google Account. Signup if you need an account.
- Billing enabled for your account with Google Compute. Enable billing in case you use all of your free trial period.

Setup Steps:

Follow these steps to create a cluster and deploy CA components:

kubernetes = CONTAINER ENGINE



* We're working on this right now

Infrastructure

When you prefer OpenShift…

Add CA Solutions to OpenShift

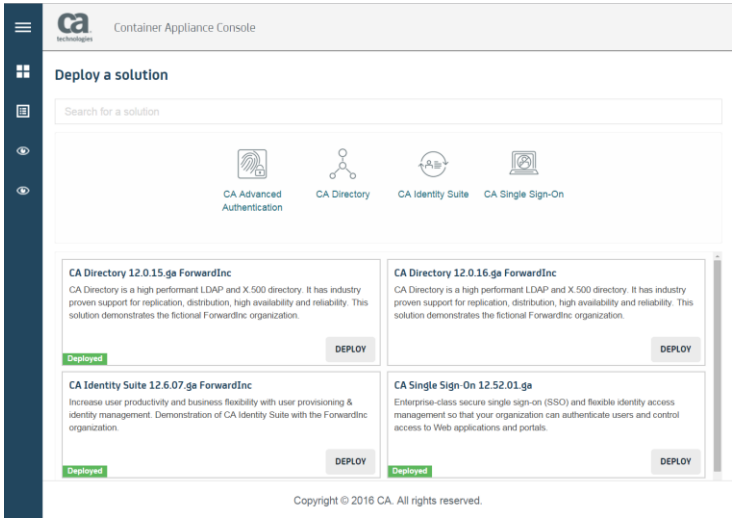
Pre-requisites:

- A working OpenShift installation.

Setup Steps:

Follow these steps to access the Container Console:

PROJECT ATOMIC + kubernetes = OPENSHIFT



* We're working on this right now

ca technologies

Configuration

Get started with minimal configuration

Most settings are already configured

Add your custom values

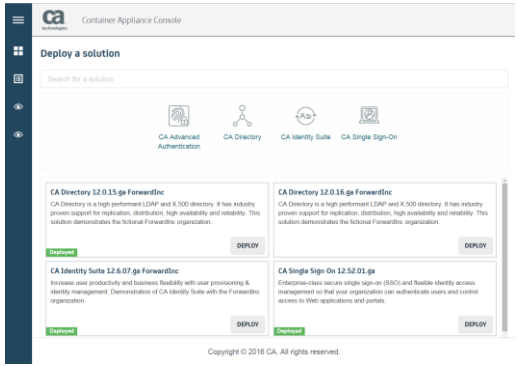
Include any policies, roles, or data to initialize the system

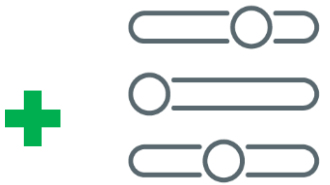
Choose URL during deployment, or upload* a zip file

* We're working on this right now

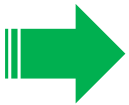
Runtime

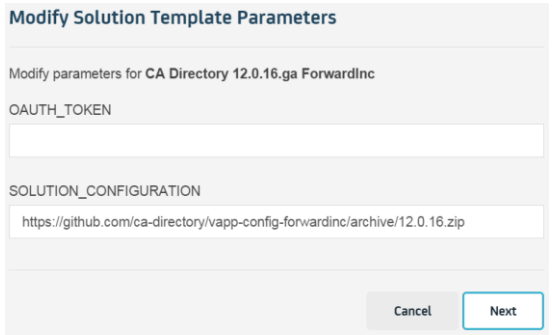
Click, choose, and use

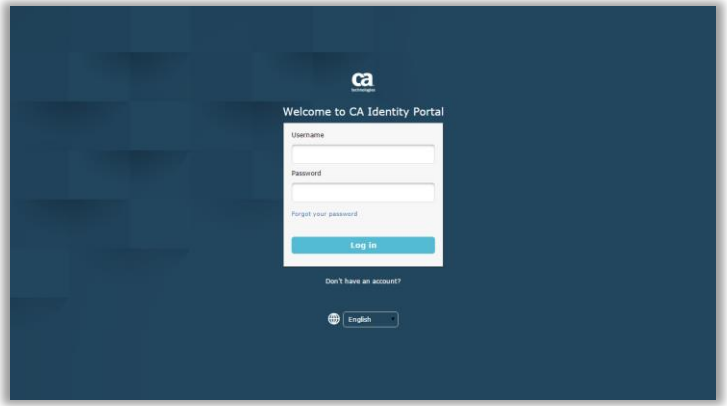




Configuration







Use Cases

Common challenges across all products

- Simplify Installation – scalable infrastructure delivered as a Virtual Machine
  - Pre-installed containers with pre-connected components provides **known environment** using **best-practice architecture**
- Simplify Configuration
  - Iteration 1: Apply configuration properties and data with flexible scripting
  - Iteration 2: Apply structured configuration and data with opinionated scripting
- Simplify Upgrade – Upgrade configuration and product with minimal interaction
- High Availability – Pre-configured for an “HA first” mentality
- Elastic Scaling – Unexpected benefit of containerization with orchestration

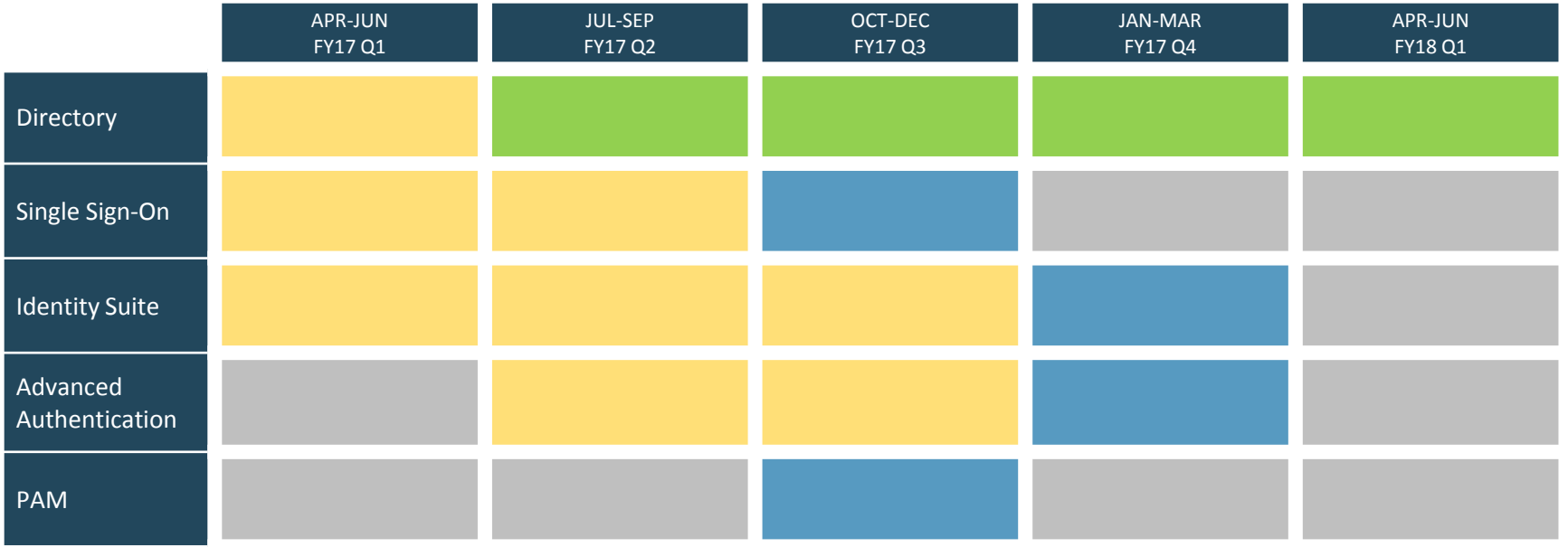
Metrics

Evaluate success with measurement

- Performance
  - Bare-metal, virtual machines, containers
- Installation and Configuration time
  - Demo, Proof of Concept, Customer
- Upgrade time
  - Minor release, Major release, hotfix
- Operating cost
  - Bare-metal, virtual machines, containers

Product Readiness Plan

Components running within containers



ca technologies

Product Readiness

CA Directory Highlights

- Automatic failure recovery
  - If any DSA fails, a new DSA starts and data is **automatically replicated**
- Directory (LDAP) requests are **automatically load balanced** to all DSAs
- DSAs can be **manually scaled** using the command line (limit of 200)
  - To turn up 6 dsas: **`kubectl scale rc dxgrid-data-rc --replicas=6`**
- Centralized logging across all DSAs using ELK stack
- Next steps:
  - Automatic scaling based on workload
  - Performance testing with high performance docker networking

Product Readiness

SSO Highlights

- Master and Worker policy servers
- Agent requests are **automatically load balanced** to all worker policy servers
  - External agents use pre-configured Host Configuration Object (HCO) to access scalable policy server on known ports (4444x)
- Worker policy servers can be **manually scaled** using the command line
  - To turn up 6 workers: **`kubectl scale rc policy-server --replicas=6`**
- Next steps:
  - Scalable access gateway
  - Reuse highly available CA Directory solution as session store, policy store

Summary

A Few Words to Review

Key topics

Cloud-native infrastructure is a new paradigm.

Starter configurations shorten time-to-value.

Findings

An efficient decision for the entire DevOps lifecycle.

Encourages iterative improvement.

Experiences

Many customers and partners are interested in reducing cost of ownership.

Many customers and partners are interested in using containers.

Tim Hobbs
Advisor, Product Management
Timothy.Hobbs@ca.com