HPE Reference Configuration for securing containers with Aqua Security

Securing container-based applications for production deployment

Reference Architecture

Contents

Executive summary

Organizations looking to deploy containers at scale are looking for technologies and best practices to secure their container environment from development to operations. Docker Enterprise Edition (EE) has inherent capabilities that can secure the container infrastructure. Aqua Security complements these features by providing security monitoring and analytics at the container application layer. This paper provides guidance for deploying and configuring Aqua's security platform to protect a Docker EE cluster on HPE Synergy. It describes:

- The capabilities of Aqua Security and how its features can be used to detect and prevent container attacks, and help IT organizations enforce security policies and compliance.
- How to leverage HPE Synergy strengths in rapid provisioning using HPE Synergy Image Streamer to deploy a Docker worker pre-configured with the Aqua monitoring agent

Docker brings application portability, security and efficiency to enterprises with a production-ready Containers-as-a-Service (CaaS) platform. The combination of Docker Enterprise Edition Advanced with HPE Composable Infrastructure allows organizations to build a high-performance and resource-efficient platform that supports large scale deployment of diverse container workloads. With Aqua, IT security can have full visibility and container protection over all phases of the container lifecycle from development to deployment on HPE Synergy.

Target audience: This document is intended for IT architects, systems integrators, and partners who are planning to deploy and manage Docker containers on a large scale running on Hewlett Packard Enterprise Synergy Composable Infrastructure.

Document purpose: The purpose of this document is to describe a best practice scenario for securing Docker container applications for production deployment using the Aqua Security software. Readers can use this document to achieve the following goals:

- Gain insight into how to leverage Aqua Security to protect containers running on the HPE Synergy platform.
- Learn how to rapidly deploy the Docker workers with Aqua Security built-in using HPE Synergy Image Streamer technology.

Introduction

This Reference Configuration builds on the solution described in HPE Reference Configuration for Docker Enterprise Edition (EE) Standard on HPE Synergy with HPE Synergy Image Streamer. This solution includes a Docker Enterprise Edition Advanced deployment along with the Aqua Container Security Platform (CSP) on HPE Composable Infrastructure and includes best practices for the use of Aqua Security to secure your Docker EE environment (Basic, Standard, or Advanced).

The HPE Synergy platform is designed to bridge traditional and cloud-native applications with the implementation of composable infrastructure. Composable infrastructure combines the use of fluid resource pools, made up of compute, storage, and fabric with software-defined intelligence. Composable pools of compute, storage and fabric can be intelligently and automatically combined to support any workload. The resource pools can be flexed to meet the needs of any business application. The use of HPE Synergy Image Streamer in the solution allows for rapid image and application changes to multiple compute nodes in an automated process.

HPE Synergy provides an ideal Docker deployment platform allowing for rapid deployment of additional Docker Universal Control Plane (UCP) swarm workers to meet changing workload demands. The introduction of HPE Synergy Image Streamer provides the capability to automatically deploy new compute servers pre-configured with the Aqua Enforcer immediately with true stateless images which integrate server hardware configuration with operating environment images in just minutes. This allows for seamless expansion and contraction of physical resource utilization providing optimal use of data center infrastructure.

The challenge: Securing containerized applications

The increasingly widespread use of software containers represents a step-change in application development and computing infrastructure, and as such, introduces new and varied security challenges.

There are several factors at play: The massive use of open source components, the DevOps speed of development and deployment, and the runtime stack where containers share the same OS kernel – all combine to increase the risk posed by software vulnerabilities, lack of adequate access control, changeable network topologies and new attack vectors.

The relative immaturity of the technologies that comprise the new container stack also leaves many unaddressed gaps. From a process standpoint, security ownership is unclear since security teams are often uninitiated in container technology. In the worst case, this lack of clarity creates a complete ownership vacuum – hamstringing security decisions. In the best case, it creates a "shift left" of responsibility, forcing developers and DevOps teams to address security on their own, and do their best to secure their container deployments as early in the process as possible, yet without a full understanding of the resulting operational security stance.

Traditional network-based security tools and other point solutions that only address part of the problem are insufficient; they fail to provide comprehensive visibility and control over the entire process, resulting in a partial understanding of the security posture. This leaves organizations open to attacks and internal threats.

Aqua developed its Container Security Platform (CSP) to offer a container-native and holistic approach to container security. Furthermore, the Aqua Platform integrates with container management tools to implement security best practices as an integral and automated part of the process. Customers deploying Docker container solutions on HPE Synergy can benefit from a combination of security protection from Docker Enterprise Edition and Aqua.

Introduction to Aqua

Aqua CSP design principles

Aqua Security enables enterprises to secure their virtual container environments from development to production, accelerating container adoption and bridging the gap between DevOps and IT security. The Aqua Platform tackles the complex challenge of securing containerized applications while adhering to a set of principles that would ensure the effectiveness and usability of the solution including:

Full lifecycle security

Most DevOps and security professionals would agree that it is important and easier to secure containers as early as possible in their lifecycle. Aqua enables this “shift left” methodology and secures a container DevOps process that starts with image development and continuous integration (CI), continues with deployment and orchestration, and once in production continues to monitor and improve. Focusing on the development process only is insufficient, and focusing on runtime without understanding the container payload is not effective. Combining the two in one seamless platform is what ensures effective security while minimizing false positives.

Automation

One reason to choose automation is that manual processes are prone to human error. But in the context of containers, it is the scale and frequency of updates that make automation not only better, but indispensable. Every change in a container image or its deployment should not require manual adjustment or the overhead of security management would become untenable. Since the Docker container build-ship-run cycle is highly automated, it is possible to integrate security into it with very little overhead.

Smart defaults

Customers want flexibility and control. But during implementation many customers stay with the default provided by the vendor because they cannot be bothered with tweaking it – and those defaults may not necessarily be tuned for security but rather for convenience. Therefore, having smart defaults that are suitable for most customer requirements are critical. It also allows the customer to benefit from enhanced security from day one of using the Aqua Platform. For example, default security profiles for the most popular open-source container images would typically be sufficient for most workloads using those containers, and only need to be tweaked for specific use cases.

Layered security model

Providing layers of security is one of the key principles of any security policy – it helps avoid single points of failure, and creates fallback solutions in case one of the layers fails to fulfill its purpose, creating defense in depth. Passive measures, such as hardening and access control mechanisms, should also be layered with active measures that monitor for anomalous behavior and generate alerts or preventive action.

Cyber intelligence

The threat landscape for containers constantly changes, as it does everywhere else in cyber security. Since containers are a new and evolving technology, there are significant new features and updates with every release that repeatedly change the attack surface. To keep up with those changes, the Aqua Platform has cyber intelligence updates fed into it on a daily basis. Aqua constantly updates its threat intelligence based on a variety of open sources, commercial sources, and in-house research.

Actionable insights

Many security solutions can point out vulnerabilities, or alert on suspicious behavior, but stop short of providing preventive measures or processes that lead to remediation. With Aqua, insights are actionable, whether automatically as part of the security policy, or in supporting user-driven processes and collaboration that drive to resolution.

Security at scale

Some of the largest scale applications in the world run on containers. These workloads can be highly variable and deploy from zero to thousands of hosts in minutes, then winding back down to zero. A security solution must, therefore, be able to scale rapidly and with usability in mind.

Aqua use cases

Five Aqua use cases are described in this Reference Configuration:

- Continuous Image Assurance
- Continuous Integration / Continuous Deployment (CI/CD) integration
- Container Role Based Access Control
- Runtime Protection
- Regulatory Compliance

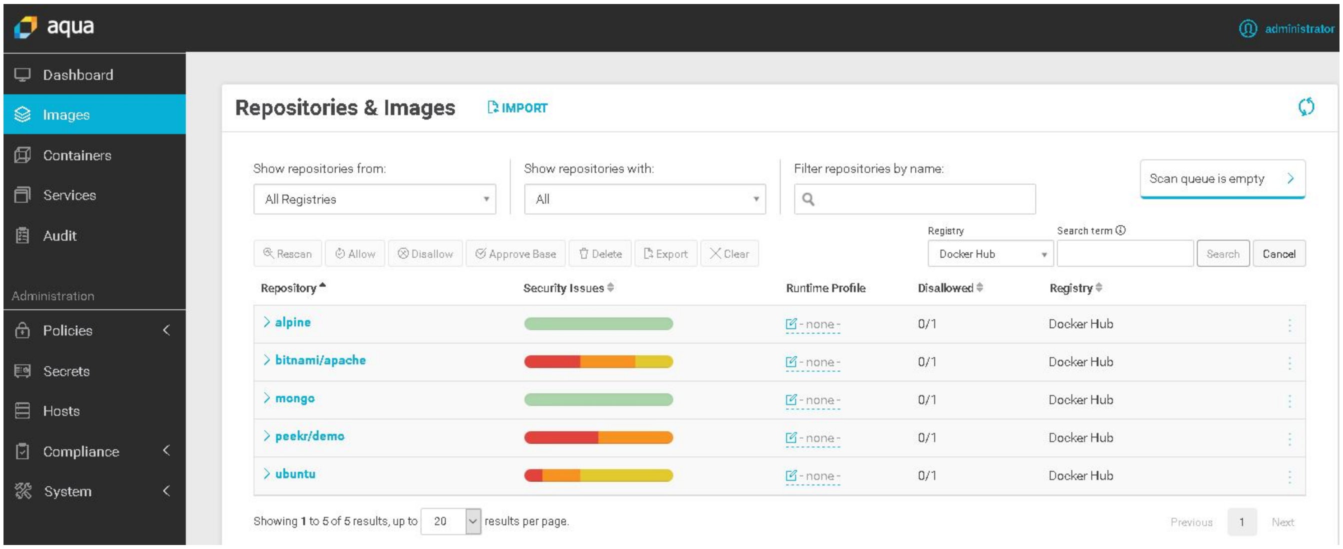
Continuous Image Assurance use case

Aqua's Continuous Image Assurance functionality covers the integrity and security of container images from the time they are created until they are deployed. This enforces the proper use of a container image within the organization, according to Aqua image assurance policy. Organizations must keep their production environment safe, only trusted and verified images should be allowed to run. This can be easily achieved with Aqua image assurance policy where users can define security controls on images and Aqua will enforce and block any image that doesn't meet the security requirements from running.

Aqua scans the images for known vulnerabilities and security best practice issues based on a continuously updated feed from multiple sources. For vulnerability scanning, Aqua supports identifying vulnerabilities in OS packages (RPM, Deb and Alpine), programming language components (Java® JavaScript, Python, Ruby, etc.) and binaries copied directly to the image. In addition to the vulnerability scanning, Aqua verifies that the image does not contain sensitive data (e.g., SSH keys, private keys, passwords) and that the image was built according to security best practices.

Administrators can define Image Assurance policies, which based on the scanning results will mark an image as "disallowed" for running. The following is an example of security controls that an administrator can use as part of the Image Assurance policy:

- Block images based on vulnerabilities found
- Block images with Common Vulnerability Scoring System (CVSS) score higher than X
- Common Vulnerabilities and Exposures (CVE) blacklist – block images with specific CVEs
- Package blacklist – block images with specific packages
- Block images with sensitive data
- Block images configured with user root



Repositories & Images

Show repositories from: All Registries

Show repositories with: All

Filter repositories by name:

Registry: Docker Hub

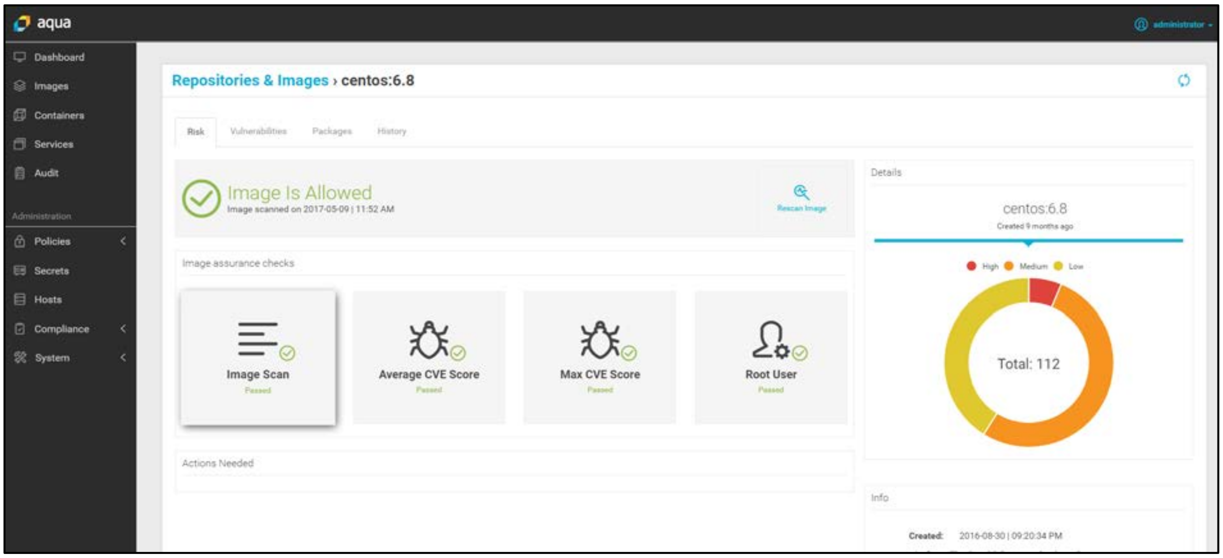
Search term

Repository	Security Issues	Runtime Profile	Disallowed	Registry
alpine	🟩	- none -	0/1	Docker Hub
bitnami/apache	🟥🟧🟨	- none -	0/1	Docker Hub
mongo	🟩	- none -	0/1	Docker Hub
peekr/demo	🟥🟧	- none -	0/1	Docker Hub
ubuntu	🟥🟧🟨	- none -	0/1	Docker Hub

Showing 1 to 5 of 5 results, up to 20 results per page.

Figure 1: Aqua repositories and Images

As new vulnerabilities are discovered every day, Aqua continuously re-validates scanned images on a daily basis, updating its scan results based on latest threat intelligence feeds.



Repositories & Images › centos:6.8

Image Is Allowed
Image scanned on 2017-05-09 | 11:52 AM

Image assurance checks

Image Scan — Passed
Average CVE Score — Passed
Max CVE Score — Passed
Root User — Passed

Actions Needed

centos:6.8
Created 9 months ago

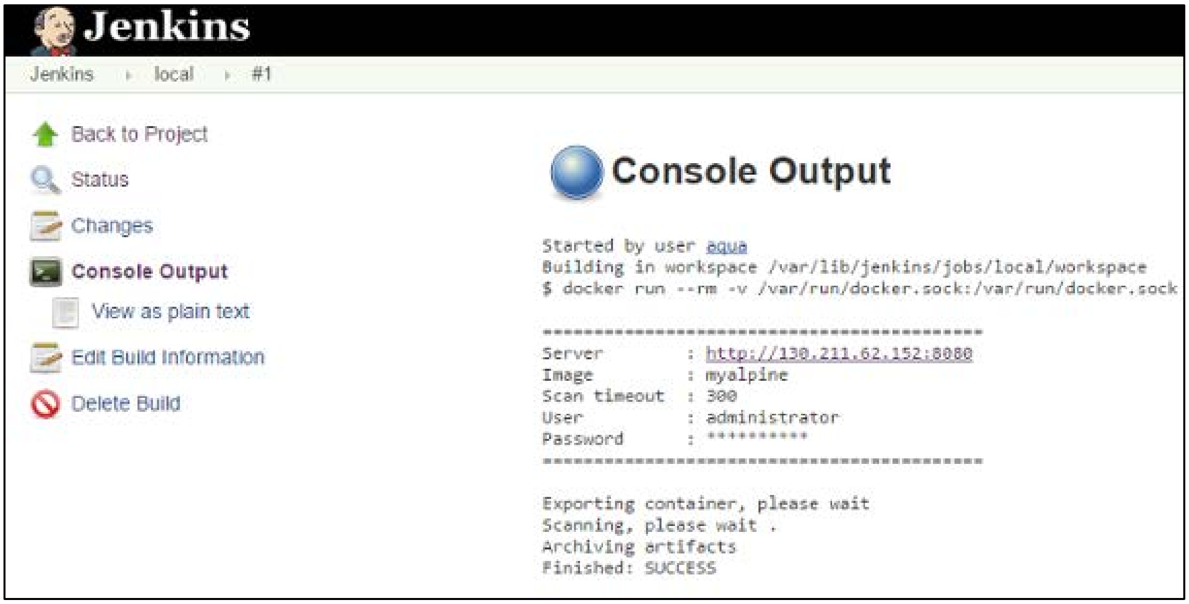
Created: 2016-08-30 | 09:20:34 PM

Figure 2: Aqua scans images for vulnerabilities

Continuous Integration / Continuous Deployment (CI/CD) integration use case

Container security should be part of the continuous integration and continuous deployment processes. Aqua has integrations with many popular CI/CD tools, such as Jenkins, Microsoft® Visual Studio Team System (VSTS), Atlassian Bamboo, JetBrains TeamCity and GoCD, usually via native plug-ins. These integrations allow CI/CD users to define Aqua as an automated step in their process, triggering vulnerability scans as images are built.

Aqua Security assessment can be configured to fail a build in case it does not pass Image Assurance policy checks defined in the Aqua Security Command Center. Adding an image assurance stage to the build process before images are pushed to the registry ensures production safety.



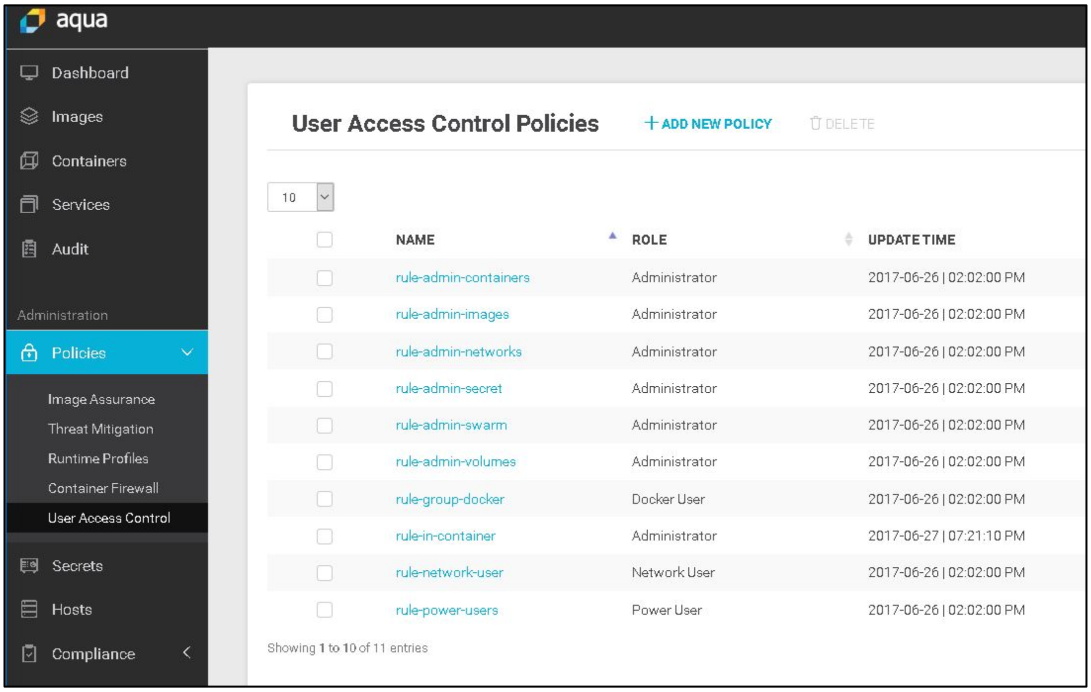
Console Output

```
Started by user aqua
Building in workspace /var/lib/jenkins/jobs/local/workspace
$ docker run --rm -v /var/run/docker.sock:/var/run/docker.sock

===========================================
Server        : http://130.211.62.152:8080
Image         : myalpine
Scan timeout  : 300
User          : administrator
Password      : **********
===========================================

Exporting container, please wait
Scanning, please wait .
Archiving artifacts
Finished: SUCCESS
```

Figure 3: CI/CD integration with Aqua Security

Container Role Based Access Control use case

With Aqua, IT administrators can set granular role-based user access control per container, image, service, and network device or storage volume as shown in Figure 4. This allows IT to limit access to these resources by user or user group, and set specifically what type of access privileges they would have, e.g., full administrator, auditor, backup operator, and more.



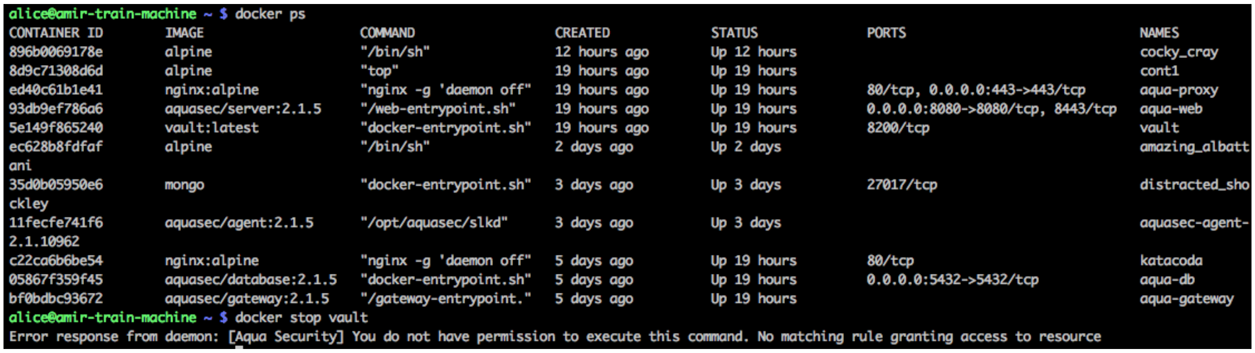
User Access Control Policies

□	NAME	ROLE	UPDATE TIME
□	rule-admin-containers	Administrator	2017-06-26 | 02:02:00 PM
□	rule-admin-images	Administrator	2017-06-26 | 02:02:00 PM
□	rule-admin-networks	Administrator	2017-06-26 | 02:02:00 PM
□	rule-admin-secret	Administrator	2017-06-26 | 02:02:00 PM
□	rule-admin-swarm	Administrator	2017-06-26 | 02:02:00 PM
□	rule-admin-volumes	Administrator	2017-06-26 | 02:02:00 PM
□	rule-group-docker	Docker User	2017-06-26 | 02:02:00 PM
□	rule-in-container	Administrator	2017-06-27 | 07:21:10 PM
□	rule-network-user	Network User	2017-06-26 | 02:02:00 PM
□	rule-power-users	Power User	2017-06-26 | 02:02:00 PM

Showing 1 to 10 of 11 entries

Figure 4: Aqua user access control policies

One example of User Access Control is to define specific users to have "Auditor" rights on containers, meaning they can look at their logs but not perform any other administrative actions on any container as shown in Figure 5.



```
alice@amir-train-machine ~ $ docker ps
CONTAINER ID        IMAGE                    COMMAND                  CREATED             STATUS              PORTS                               NAMES
896b0069178e        alpine                   "/bin/sh"                12 hours ago        Up 12 hours                                             cocky_cray
8d9c71308d6d        alpine                   "top"                    19 hours ago        Up 19 hours                                             cont1
ed40c61b1e41        nginx:alpine             "nginx -g 'daemon off"   19 hours ago        Up 19 hours         80/tcp, 0.0.0.0:443->443/tcp        aqua-proxy
93db9ef786a6        aquasec/server:2.1.5     "/web-entrypoint.sh"     19 hours ago        Up 19 hours         0.0.0.0:8080->8080/tcp, 8443/tcp    aqua-web
5e149f865240        vault:latest             "docker-entrypoint.sh"   19 hours ago        Up 19 hours         8200/tcp                            vault
ec628b8fdfaf        alpine                   "/bin/sh"                2 days ago          Up 2 days                                               amazing_albatt
ani
35d0b05950e6        mongo                    "docker-entrypoint.sh"   3 days ago          Up 3 days           27017/tcp                           distracted_sho
ckley
11fecfe741f6        aquasec/agent:2.1.5      "/opt/aquasec/slkd"      3 days ago          Up 3 days                                               aquasec-agent-
2.1.10962
c22ca6b6be54        nginx:alpine             "nginx -g 'daemon off"   5 days ago          Up 19 hours         80/tcp                              katacoda
05867f359f45        aquasec/database:2.1.5   "docker-entrypoint.sh"   5 days ago          Up 19 hours         0.0.0.0:5432->5432/tcp              aqua-db
bf0bdbc93672        aquasec/gateway:2.1.5    "/gateway-entrypoint."   5 days ago          Up 19 hours                                             aqua-gateway
alice@amir-train-machine ~ $ docker stop vault
Error response from daemon: [Aqua Security] You do not have permission to execute this command. No matching rule granting access to resource
```

Figure 5: User with auditor rights has view-only privileges

Runtime Protection use case

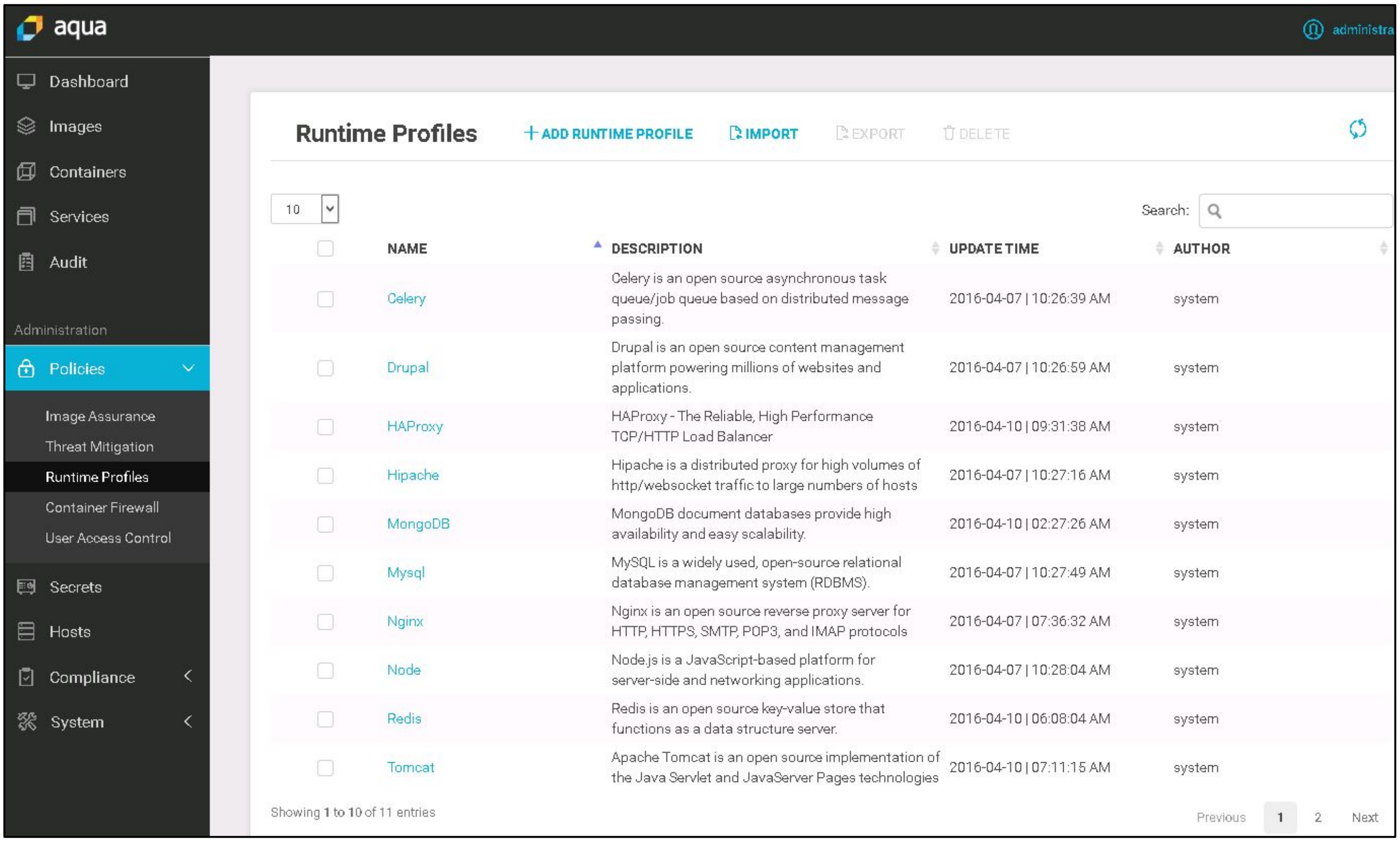
During runtime, and especially when containers are deployed in production environments, having vulnerability-free images is a good starting point, but is insufficient to stop attacks, user privilege abuse and malicious container activity. The Aqua Platform employs a variety of mechanisms to further harden, tighten and monitor container behavior. The Platform proactively detects and prevents malicious container behavior. The system administrator can choose whether to run in "detect mode", that will only generate alerts, or in "prevent mode", that will automatically take preventive action to stop prohibited or suspicious container behavior.

- Behavioral Runtime Profiles

  At the heart of the Aqua security model are container runtime profiles that are generated automatically based on behavioral analytics of a container's behavior. Since containers typically perform simple, limited operations, their behavior is easier to predict than the behavior of monolithic applications. Runtime profiles whitelist normal behavior, and make it easy to detect anomalies and escalation attempts, with very few false positives. Behavioral runtime profiles may include which read-only files can be accessed by the container, inbound and outbound network connections, a list of allowed executables, whether the container can run as root, what storage volumes are allowed, as well as resource limits on simultaneous processes, CPU and memory usage.

- Official Image Default Profiles

  Aqua provides "out of the box" security profiles for popular official images, such as MongoDB, Redis, Nginx, etc., as shown in Figure 6. These security profiles are used to provide secure defaults for deployment, even without any container behavior profiling. The default profiles can also be assigned automatically to other images that have similar functionality and can therefore benefit from similar "least privilege" settings.



Runtime Profiles

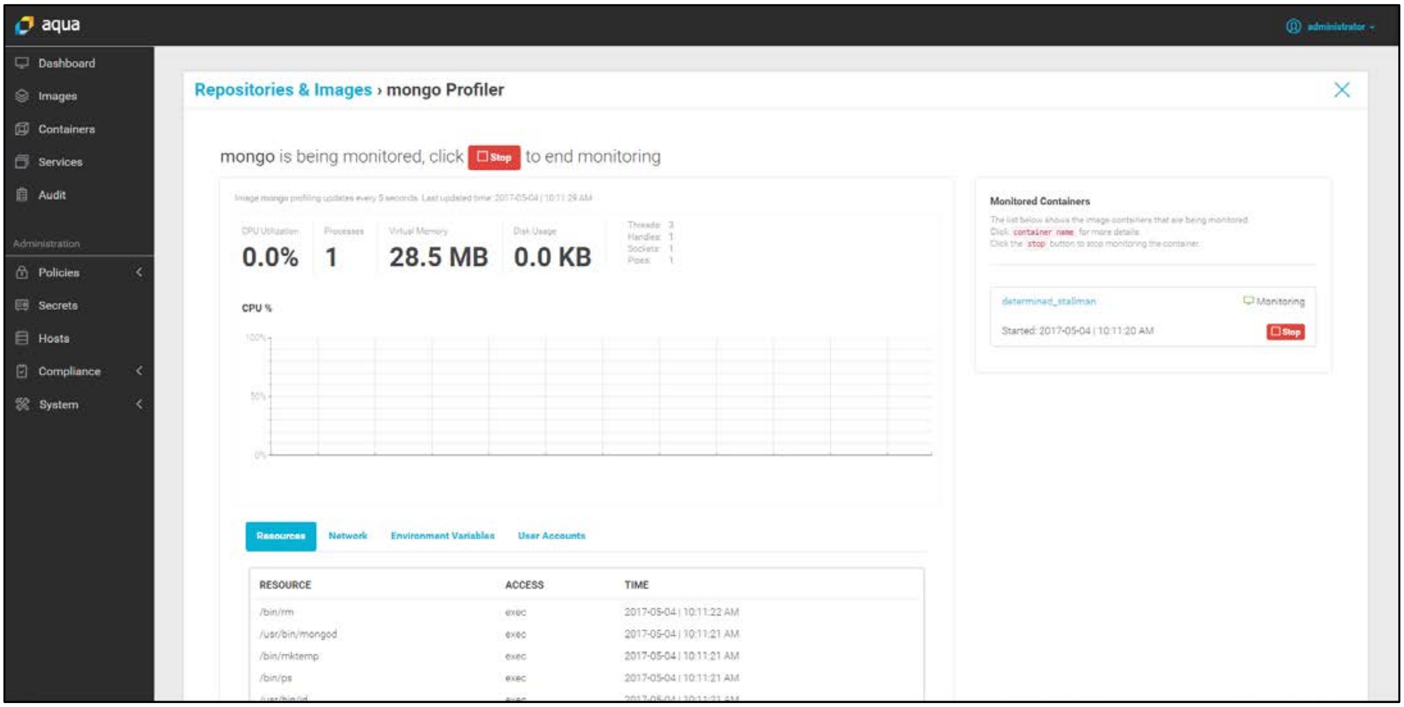
☐	NAME	DESCRIPTION	UPDATE TIME	AUTHOR
☐	Celery	Celery is an open source asynchronous task queue/job queue based on distributed message passing.	2016-04-07 | 10:26:39 AM	system
☐	Drupal	Drupal is an open source content management platform powering millions of websites and applications.	2016-04-07 | 10:26:59 AM	system
☐	HAProxy	HAProxy - The Reliable, High Performance TCP/HTTP Load Balancer	2016-04-10 | 09:31:38 AM	system
☐	Hipache	Hipache is a distributed proxy for high volumes of http/websocket traffic to large numbers of hosts	2016-04-07 | 10:27:16 AM	system
☐	MongoDB	MongoDB document databases provide high availability and easy scalability.	2016-04-10 | 02:27:26 AM	system
☐	Mysql	MySQL is a widely used, open-source relational database management system (RDBMS).	2016-04-07 | 10:27:49 AM	system
☐	Nginx	Nginx is an open source reverse proxy server for HTTP, HTTPS, SMTP, POP3, and IMAP protocols	2016-04-07 | 07:36:32 AM	system
☐	Node	Node.js is a JavaScript-based platform for server-side and networking applications.	2016-04-07 | 10:28:04 AM	system
☐	Redis	Redis is an open source key-value store that functions as a data structure server.	2016-04-10 | 06:08:04 AM	system
☐	Tomcat	Apache Tomcat is an open source implementation of the Java Servlet and JavaServer Pages technologies	2016-04-10 | 07:11:15 AM	system

Showing 1 to 10 of 11 entries

Figure 6: Aqua default security profiles

- Cyber Threat Mitigation

  Aqua's Cyber Intelligence team has identified a collection of malicious behaviors, such as port scanning, fork bombs, and connecting to ill-reputed external IP addresses. Aqua's cyber threat mitigation features automatically protect against these behaviors, acting as an additional layer of defense.



mongo is being monitored, click □ Stop to end monitoring

Image mongo profiling updates every 5 seconds. Last updated time: 2017-05-04 | 10:11:29 AM

CPU Utilization	Processes	Virtual Memory	Disk Usage
0.0%	1	28.5 MB	0.0 KB

Threads: 3
Handles: 1
Sockets: 1
Pipes: 1

Monitored Containers

The list below shows the image containers that are being monitored.
Click `container name` for more details.
Click the `stop` button to stop monitoring the container.

determined_stallman □ Monitoring
Started: 2017-05-04 | 10:11:20 AM □ Stop

RESOURCE	ACCESS	TIME
/bin/rm	exec	2017-05-04 | 10:11:22 AM
/usr/bin/mongod	exec	2017-05-04 | 10:11:21 AM
/bin/mktemp	exec	2017-05-04 | 10:11:21 AM
/bin/ps	exec	2017-05-04 | 10:11:21 AM
/usr/bin/id	exec	2017-05-04 | 10:11:21 AM

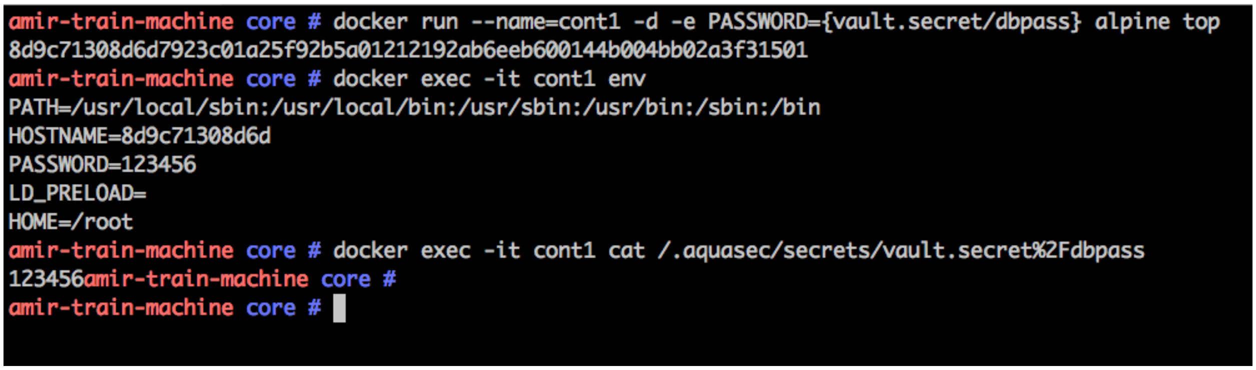
Figure 7: Monitoring containers with Aqua

- Secrets Management

  In many environments, there is a need to pass sensitive information like passwords, connection strings or tokens into a container. A sensitive piece of information is called a "secret" and Aqua provides central management and secure distribution of secrets into running containers. Aqua users can define a secret in the Aqua Management console, and assign access control policies that authorize users or groups to run containers that make use of the secret.

  When a secret is used, its value will be automatically injected into the container, either as an environment variable or file. The value of the secret is not visible outside the container, is not stored on disk, and is encrypted in transit.

  Aqua also integrates with several secrets stores such as HashiCorp Vault, Amazon Web Services (AWS) Key Management Service (KMS) and Microsoft Azure key store. This enables the use of secrets that are already defined in the secret stores and to distribute their values through Aqua's access control system to containers.

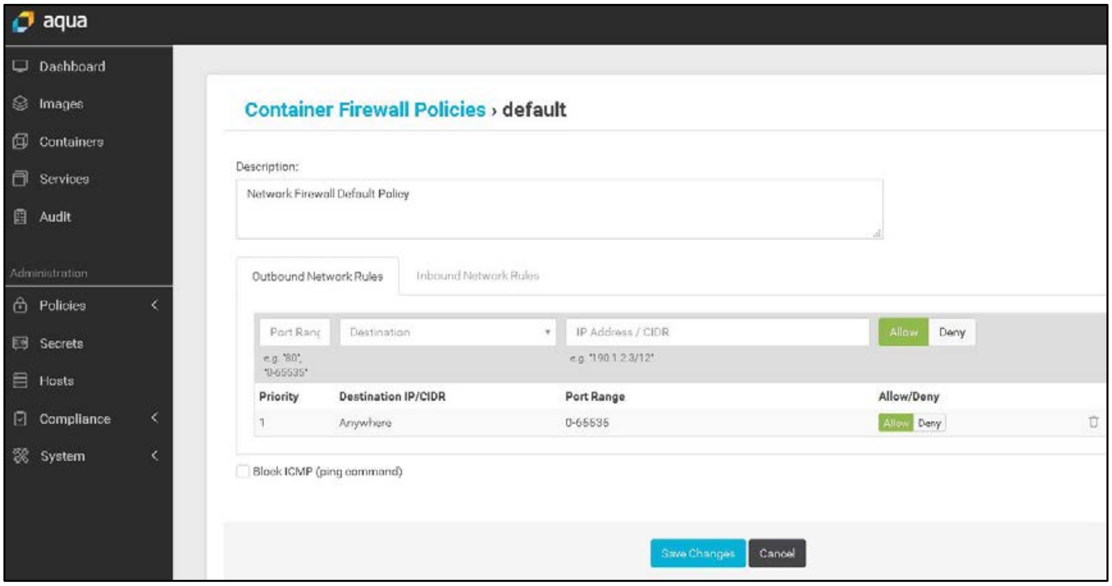


```
amir-train-machine core # docker run --name=cont1 -d -e PASSWORD={vault.secret/dbpass} alpine top
8d9c71308d6d7923c01a25f92b5a01212192ab6eeb600144b004bb02a3f31501
amir-train-machine core # docker exec -it cont1 env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=8d9c71308d6d
PASSWORD=123456
LD_PRELOAD=
HOME=/root
amir-train-machine core # docker exec -it cont1 cat /.aquasec/secrets/vault.secret%2Fdbpass
123456amir-train-machine core #
amir-train-machine core # 
```

Figure 8: Docker secrets are only visible within the container

- Container Firewall (Nano Segmentation)

  Aqua can set application-level network policies that determine firewall rules for network access between containers, and allow or prevent specific containers from having any inbound or outbound network connections as shown in Figure 9. This allows for greater isolation between containers that have no need to communicate with each other, preventing “East-West” lateral movement attacks or creating network boundaries across services, where users can control which networks are accessible for each service.



Container Firewall Policies › default

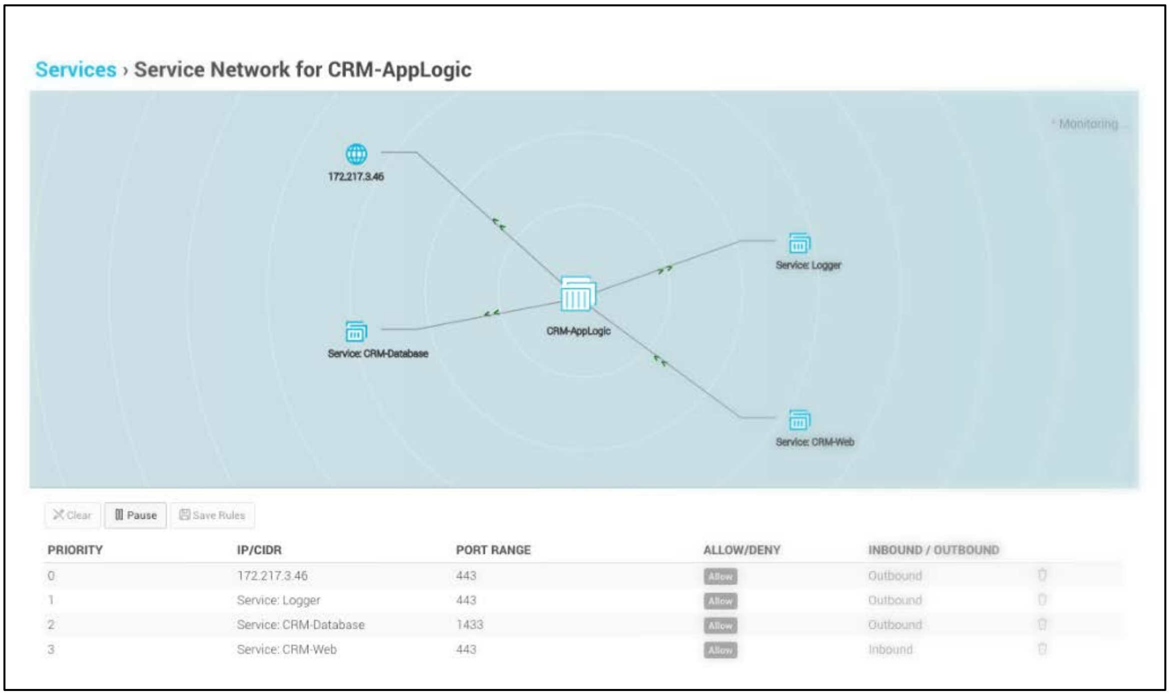
Description:
Network Firewall Default Policy

Priority	Destination IP/CIDR	Port Range	Allow/Deny
1	Anywhere	0-65535	Allow Deny

□ Block ICMP (ping command)

Figure 9: Firewall Policy creation

Additionally, users can automatically block network connections between containers that were not linked to each other using the Docker “link” command. For example, users can create a firewall rule that only allows containers which are members of the “Database” service to be accessible from containers which are members of the "Web" service. Services and external network connections can be visualized in the network topology map. (Figure 10)



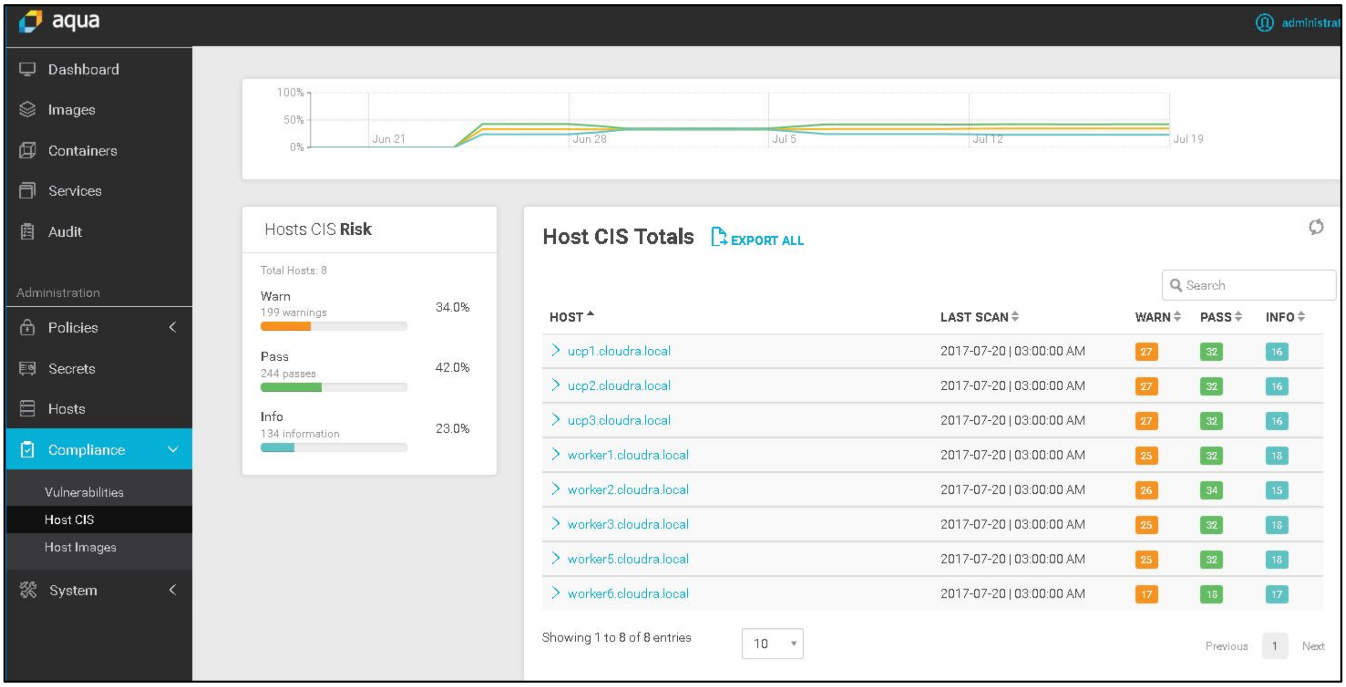
PRIORITY	IP/CIDR	PORT RANGE	ALLOW/DENY	INBOUND / OUTBOUND
0	172.217.3.46	443	Allow	Outbound
1	Service: Logger	443	Allow	Outbound
2	Service: CRM-Database	1433	Allow	Outbound
3	Service: CRM-Web	443	Allow	Inbound

Figure 10: Dynamically updated Visual Network Topology

Regulatory Compliance use case

Aqua provides snapshots and event data required for regulatory compliance where containers are used for critical applications that must adhere to industry and corporate governance. Compliance information is provided around several key areas – image vulnerability and configuration status, runtime environment status, and container runtime events such as start/stop and user access.

With Aqua's integration into security information and event management (SIEM) systems such as ArcSight, Splunk, Sumo Logic and IBM QRadar, this data can be rolled up into higher-level reports for mixed environments.



Hosts CIS **Risk**

Total Hosts: 8

Warn
199 warnings 34.0%

Pass
244 passes 42.0%

Info
134 information 23.0%

Host CIS Totals

HOST	LAST SCAN	WARN	PASS	INFO
ucp1.cloudra.local	2017-07-20 | 03:00:00 AM	27	32	16
ucp2.cloudra.local	2017-07-20 | 03:00:00 AM	27	32	16
ucp3.cloudra.local	2017-07-20 | 03:00:00 AM	27	32	16
worker1.cloudra.local	2017-07-20 | 03:00:00 AM	25	32	18
worker2.cloudra.local	2017-07-20 | 03:00:00 AM	26	34	15
worker3.cloudra.local	2017-07-20 | 03:00:00 AM	25	32	18
worker5.cloudra.local	2017-07-20 | 03:00:00 AM	25	32	18
worker6.cloudra.local	2017-07-20 | 03:00:00 AM	17	18	17

Showing 1 to 8 of 8 entries 10

Previous 1 Next

Figure 11: CIS Benchmark report for hosts running Docker containers

Solution overview

This Reference Configuration provides a solution architected for Docker Enterprise Edition Advanced with Aqua Container Security Platform on hybrid bare metal and virtualized HPE Composable Infrastructure and integrations to provide automated server provisioning and container management as shown in Figure 12. This creates a scalable, secure, and highly available platform for a Docker Enterprise Edition Advanced deployment.

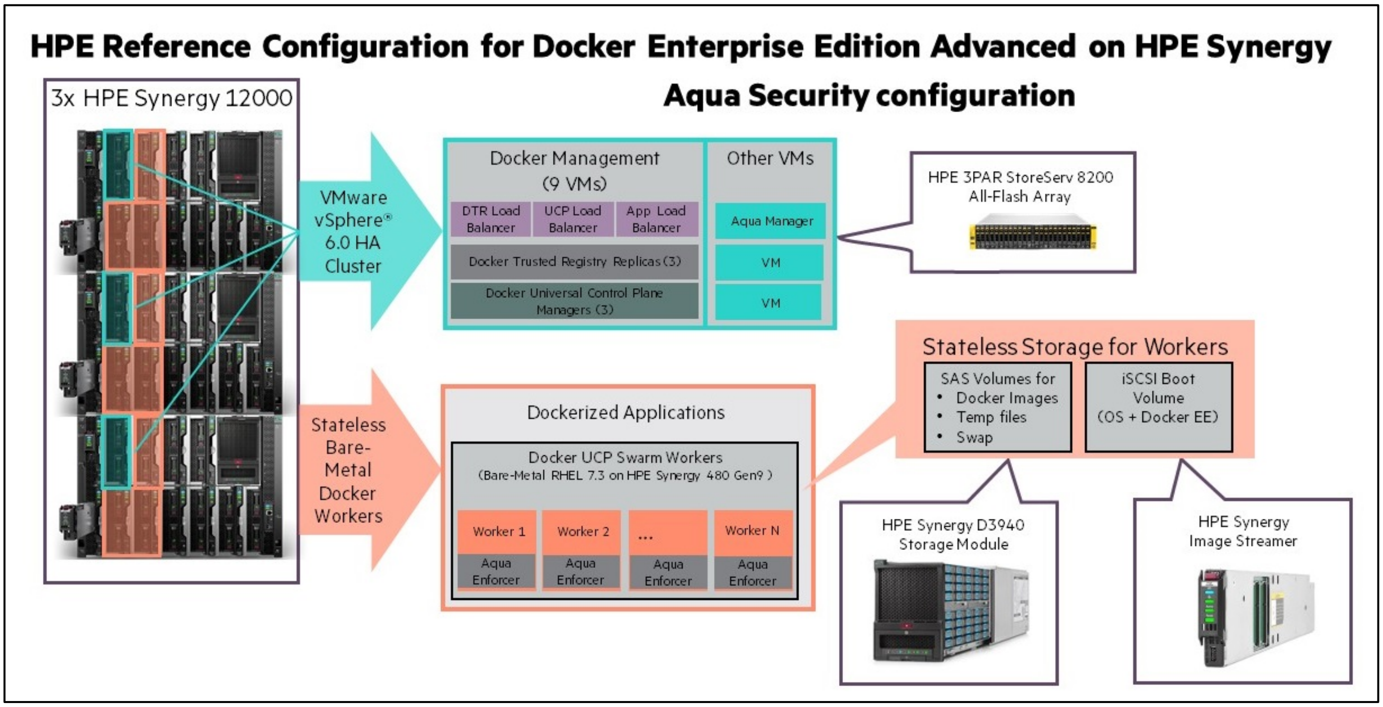

HPE Reference Configuration for Docker Enterprise Edition Advanced on HPE Synergy
Aqua Security configuration
3x HPE Synergy 12000
VMware vSphere® 6.0 HA Cluster
Stateless Bare-Metal Docker Workers
Docker Management (9 VMs)
DTR Load Balancer
UCP Load Balancer
App Load Balancer
Docker Trusted Registry Replicas (3)
Docker Universal Control Plane Managers (3)
Other VMs
Aqua Manager
VM
VM
HPE 3PAR StoreServ 8200 All-Flash Array
Stateless Storage for Workers
SAS Volumes for
• Docker Images
• Temp files
• Swap
iSCSI Boot Volume (OS + Docker EE)
Dockerized Applications
Docker UCP Swarm Workers (Bare-Metal RHEL 7.3 on HPE Synergy 480 Gen9)
Worker 1
Worker 2
...
Worker N
Aqua Enforcer
Aqua Enforcer
Aqua Enforcer
Aqua Enforcer
HPE Synergy D3940 Storage Module
HPE Synergy Image Streamer

Figure 12: Solution overview: Docker Enterprise Edition Advanced on HPE Synergy with Aqua Security

This solution uses a Docker Enterprise Edition Advanced deployment consisting of Docker Universal Control Plane (UCP), Docker Trusted Registry (DTR), and Docker Engine. The Docker management nodes and the Aqua Manager will be running as virtual machines on a VMware® vSphere cluster as shown in Figure 12. For high availability (HA) and load balancing purposes, Docker recommends multiple hosts for Docker UCP, DTR, and load balancers. Running these components initially in virtual instead of physical hosts reduces overall hardware requirements and promotes better resource utilization. HPE Composable Infrastructure provides a scalable virtualization platform for the VMware vSphere environment. The VMware Distributed Resource Scheduler (DRS) clusters will provide resource management and additional high availability for the Docker management plane which is running on virtual machines. The Docker UCP worker nodes are running on bare metal HPE Synergy 480 Gen9 Compute Modules and can be efficiently scaled out using HPE Synergy Image Streamer to rapidly deploy additional servers to the solution.

Golden images and automated deployment plans configured and stored in HPE Synergy Image Streamer enable rapid deployment of stateless bare metal Docker UCP swarm workers with Aqua Enforcer containers which immediately join the Docker swarm cluster as the server is powered on. The Aqua Enforcers automatically connect to the Aqua Command Center as shown in Figure 13. In order to efficiently utilize the limited amount of storage on the HPE Synergy Image Streamer, the Docker container images, when pulled from the Docker Trusted Registry, are deployed on HPE Synergy D3940 Storage Module volumes which are presented as local storage to the compute servers.

Aqua solution components

The Aqua Container Security Platform is composed of the following components:

- **Aqua Command Center** – The Aqua Command Center is used for managing security policies and viewing container security alerts and vulnerabilities.
- **Aqua Gateway** – The Aqua Gateway is used to communicate with Aqua Enforcers and client components.
- **Aqua Enforcer** – The Aqua Enforcer is a container deployed on every Docker host. The Aqua Enforcer adds protection to all the containers running on the same host.
- **Aqua Scanner-CLI** – The scanner-cli client is used to initiate a scan of local and remote images, and receive the results locally. This component is used mainly in the CI/CD pipeline.
- **Aqua Cyber Intelligence** – Aqua Cyber Intelligence is a cloud service that supplies threat intelligence information to the Aqua deployments.

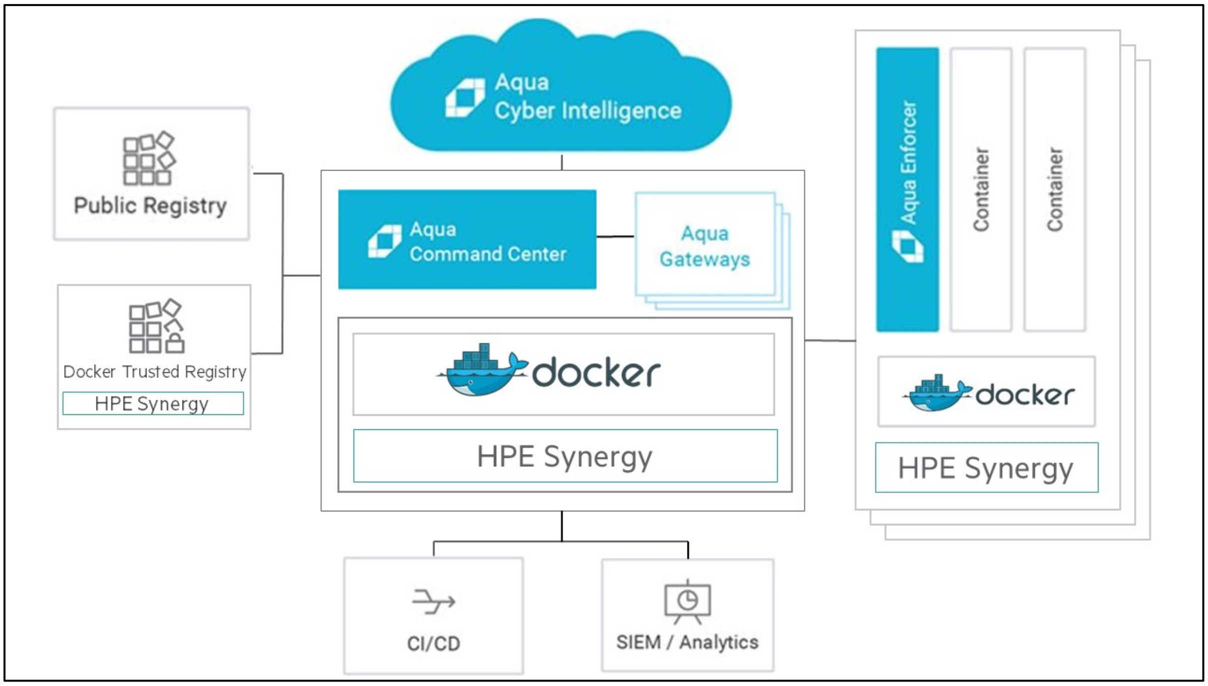


Figure 13: Aqua Container Security Platform on HPE Synergy

For this Reference Configuration, the Aqua Command Center and Aqua Gateway were deployed on a single virtual machine and configured to allow automated Aqua Enforcer container deployments. The Aqua Enforcer is deployed as part of the Docker Worker node deployment by HPE Synergy Image Streamer. Details on the deployment of the Aqua Command Center and Aqua Enforcer are found in Appendix C: Configuring the Aqua Management Console and Enforcer.

Solution components

The solution hardware and software components used to build the HPE Reference Configuration for Docker Enterprise Edition Advanced with Aqua Security on HPE Composable Infrastructure are detailed in this section.

Solution hardware

The following hardware components were utilized in this Reference Configuration as listed in Table 1.

Table 1: Hardware components

Component	Purpose
HPE Synergy 12000 Frame	Infrastructure for compute, storage, fabric and management
HPE Synergy Composer	Infrastructure management
HPE Synergy Image Streamer	Infrastructure deployment
HPE Synergy 480 Gen9 Compute Modules	Docker virtualization and bare metal hosts
HPE Synergy D3940 Storage Module	Storage for Docker hosts

HPE Synergy

HPE Synergy, the first Composable Infrastructure, empowers IT to create and deliver new value instantly and continuously. This single infrastructure reduces operational complexity for traditional workloads and increases operational velocity for the new breed of applications and services. Through a single interface, HPE Synergy composes compute, storage and fabric pools into any configuration for any application. It also enables a broad range of applications from bare metal to virtual machines to containers, and operational models like hybrid cloud and DevOps. HPE Synergy enables IT to rapidly react to new business demands.

HPE Synergy Frames contain a management appliance called the HPE Synergy Composer which hosts HPE OneView. HPE Synergy Composer manages the composable infrastructure and delivers:

- Fluid pools of resources, where a single infrastructure of compute, storage and fabric boots up ready for workloads and demonstrates self-assimilating capacity.
- Software-defined intelligence, with a single interface that precisely composes logical infrastructures at near-instant speeds; and demonstrates template-driven, frictionless operations.
- Unified API access, which enables simple line-of-code programming of every infrastructure element; easily automates IT operational processes; and effortlessly automates applications through infrastructure deployment.

HPE Synergy Composer

HPE Synergy Composer provides the enterprise-level management to compose and deploy system resources to your application needs. This management appliance uses software-defined intelligence with embedded HPE OneView to aggregate compute, storage, and fabric resources in a manner that scales to your application needs, instead of being restricted to the fixed ratios of traditional resource offerings. HPE Synergy template-based provisioning enables fast time to service with a single point for defining compute module state, pooled storage, network connectivity, and boot image.

HPE OneView is a comprehensive unifying platform designed from the ground up for converged infrastructure management. A unifying platform increases the productivity of every member of the internal IT team across servers, storage, and networking. By streamlining processes, incorporating best practices, and creating a new holistic way to work, HPE OneView provides organizations with a more efficient way to work. It is designed for open integration with existing tools and processes to extend these efficiencies.

HPE OneView is instrumental for the deployment and management of HPE compute modules and networking. It collapses infrastructure management tools into a single resource-oriented architecture that provides direct access to all logical and physical resources of the solution. Logical resources include server profiles and server profile templates, enclosures and enclosure groups, and logical interconnects and logical interconnect groups. Physical resources include server hardware blades and rack servers, networking interconnects, and computing resources.

The HPE OneView converged infrastructure platform offers a uniform way for administrators to interact with resources by providing a RESTful API foundation. The RESTful APIs enable administrators to utilize a growing ecosystem of integrations to further expand the advantages of the integrated resource model that removes the need for the administrator to enter and maintain the same configuration data more than once and keep all versions up to date. It encapsulates and abstracts many underlying tools behind the integrated resource model, so the administrator can operate with new levels of simplicity, speed, and agility to provision, monitor, and maintain the solution.

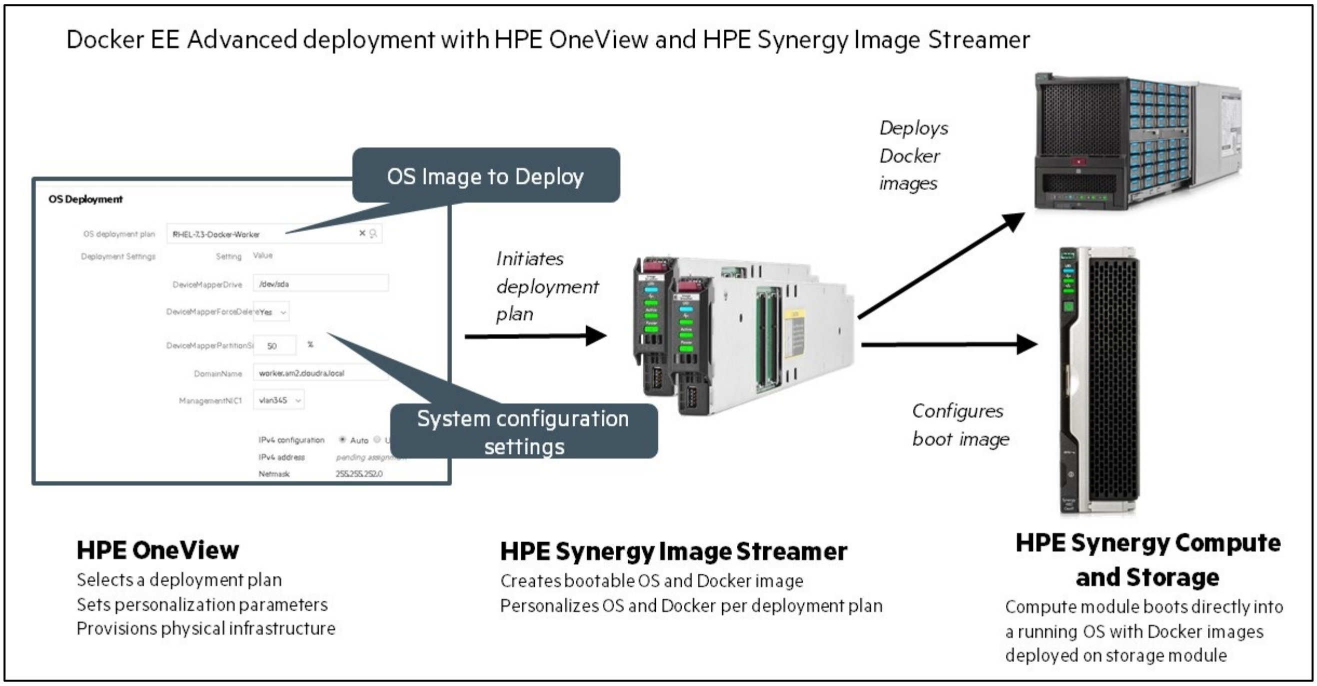
HPE Synergy Image Streamer

HPE Synergy Image Streamer is a new approach to deployment and updates for composable infrastructure. This management appliance works with HPE Synergy Composer for fast software-defined control over physical compute modules with operating system and application provisioning. HPE Synergy Image Streamer enables true stateless computing combined with the capability for image lifecycle management. This management appliance rapidly deploys and updates infrastructure.

HPE Synergy Image Streamer adds a powerful dimension to “infrastructure as code”—the ability to manage physical servers like virtual machines. In traditional environments, deploying an OS and applications or hypervisor is time consuming because it requires building or copying the software image onto individual servers, possibly requiring multiple reboot cycles. In HPE Synergy, the tight integration of HPE Synergy Image Streamer with HPE Synergy Composer enhances server profiles with images and personalities for true stateless operation.

HPE Synergy Composer, powered by HPE OneView, captures the physical state of the server in the server profile. HPE Synergy Image Streamer enhances this server profile (and its desired configuration) by capturing your golden image as the “deployed software state” in the form of bootable image volumes. These enhanced server profiles and bootable OS plus application images are software structures (“infrastructure as code”)—no compute module hardware is required for these operations. The bootable images are stored on redundant HPE Synergy Image Streamer appliances, and they are available for fast implementation onto multiple compute modules at any time. This enables bare metal compute modules to boot directly into a running OS with applications and multiple compute modules to be quickly updated.

Figure 14 shows how an HPE OneView server profile is configured to deploy a compute module with Docker Enterprise Edition Advanced installed. The Server Profile specifies the required networking, storage and firmware as well as the OS deployment plan from HPE Synergy Image Streamer. The physical state and deployed software state are maintained separate from the physical compute module. The physical compute module does not need to retain any state.


Docker EE Advanced deployment with HPE OneView and HPE Synergy Image Streamer
OS Image to Deploy
System configuration settings
Initiates deployment plan
Deploys Docker images
Configures boot image

OS Deployment

OS deployment plan: RHEL-7.3-Docker-Worker

Deployment Settings — Setting: Value

DeviceMapperDrive: /dev/sda

DeviceMapperForceDele: Yes

DeviceMapperPartitionSi: 50 %

DomainName: worker.am2.cloudra.local

ManagementNIC1: vlan345

IPv4 configuration: ● Auto ○ U

IPv4 address: pending assignment

Netmask: 255.255.252.0

HPE OneView
Selects a deployment plan
Sets personalization parameters
Provisions physical infrastructure

HPE Synergy Image Streamer
Creates bootable OS and Docker image
Personalizes OS and Docker per deployment plan

HPE Synergy Compute and Storage
Compute module boots directly into a running OS with Docker images deployed on storage module

Figure 14: Deploying Docker workers with HPE OneView and HPE Synergy Image Streamer

HPE Synergy D3940 Storage Module

The HPE Synergy D3940 Storage Module provides a fluid pool of storage resources for the composable infrastructure. Additional capacity for compute modules is easily provisioned and intelligently managed with integrated data services for availability and protection. The 40 SFF drive bays per storage module can be populated with 12 G SAS or 6 G SATA drives.

The HPE Synergy D3940 Storage Module provides local storage to compute resources and can meet the demands of a wide range of data workloads. Pooled storage resources provide the flexibility and performance needed to accommodate a wide range of workloads. Changes such as updating firmware are automatically implemented with the infrastructure online significantly reducing errors and delivering real-time compliance.

For this Reference Configuration the HPE Synergy D3940 Storage Module was used for Docker images. The server profiles for Docker worker nodes include the configuration of local storage where the Docker volumes are mounted as shown in Figure 14. Configuring the storage in this manner minimizes the use of limited storage on the HPE Synergy Image Streamer.

HPE Synergy 480 Gen9 Compute Module

The HPE Synergy 480 Gen9 Compute Module delivers superior capacity, efficiency, and flexibility in a two-socket, half-height form factor to support demanding workloads. The HPE Synergy 480 Gen9 Compute Module provides a composable compute resource that can be self-discovered, quickly provisioned, easily managed, and seamlessly redeployed to deliver the right compute capacity for changing workload needs.

Solution management software

The following software components were utilized in this Reference Configuration as listed in the tables below.

Table 2: Docker Enterprise Edition Advanced subscription components

Component	Version
Docker Universal Control Plane (UCP)	2.1.3
Docker Trusted Registry (DTR)	2.2.4
Docker Enterprise Edition Engine	17.03

Table 3: Hewlett Packard Enterprise solution management software

Component	Version
HPE Synergy Composer	3.00.07
HPE Synergy Image Streamer	3.00.05
HPE Synergy Image Streamer artifacts for Docker + Aqua	RHEL-7.3-Docker-Worker-With-AquaSec-2017-06-16.zip

Table 4: Aqua software components

Component	Version
Aqua Command Center	2.1.6
Aqua Enforcer	2.1.6

Docker Enterprise Edition Advanced

Docker Enterprise Edition Advanced provides Containers-as-a-Service (CaaS) to enterprises with a production-ready platform supported by Docker and hosted locally behind the firewall.

Docker Enterprise Edition Advanced enables enterprises to containerize applications, and deploy them across the underlying infrastructure of choice, in a way that is efficient and secure. This enterprise container platform from Docker spans across the application lifecycle with tooling for both developers and IT operations and support from Docker or HPE. The platform is available in three different versions: Basic, Standard, and Advanced. This Reference Configuration focuses on Docker Enterprise Edition Advanced.

Docker Enterprise Edition Advanced features include:

- Built-in clustering and orchestration via Docker Swarm and Compose
- Integrated image and container management
- Cryptographic image signing enabled via Docker Content Trust to protect against man in the middle attacks
- Policy enforcement for image signing and run only signed images in production
- Integrated secrets management for a least privilege access strategy
- Granular role based access controls to manage users and teams access to all system components
- Flexible load balancing with built-in routing mesh for L4 or L7 load balancing
- Application health checks configuration within GUI
- Support for all Docker API and commands
- Image scanning via Docker Security Scanning
- GUI and CLI usability

Docker Enterprise Edition Advanced delivers a universal, platform-agnostic container runtime with built-in orchestration, networking and volumes for container-based applications. It offers open APIs for automation, extensibility and integrations into existing systems like LDAP/AD, monitoring, logging and more. The solution comes with Hewlett Packard Enterprise technical product support if the license is procured through Hewlett Packard Enterprise.

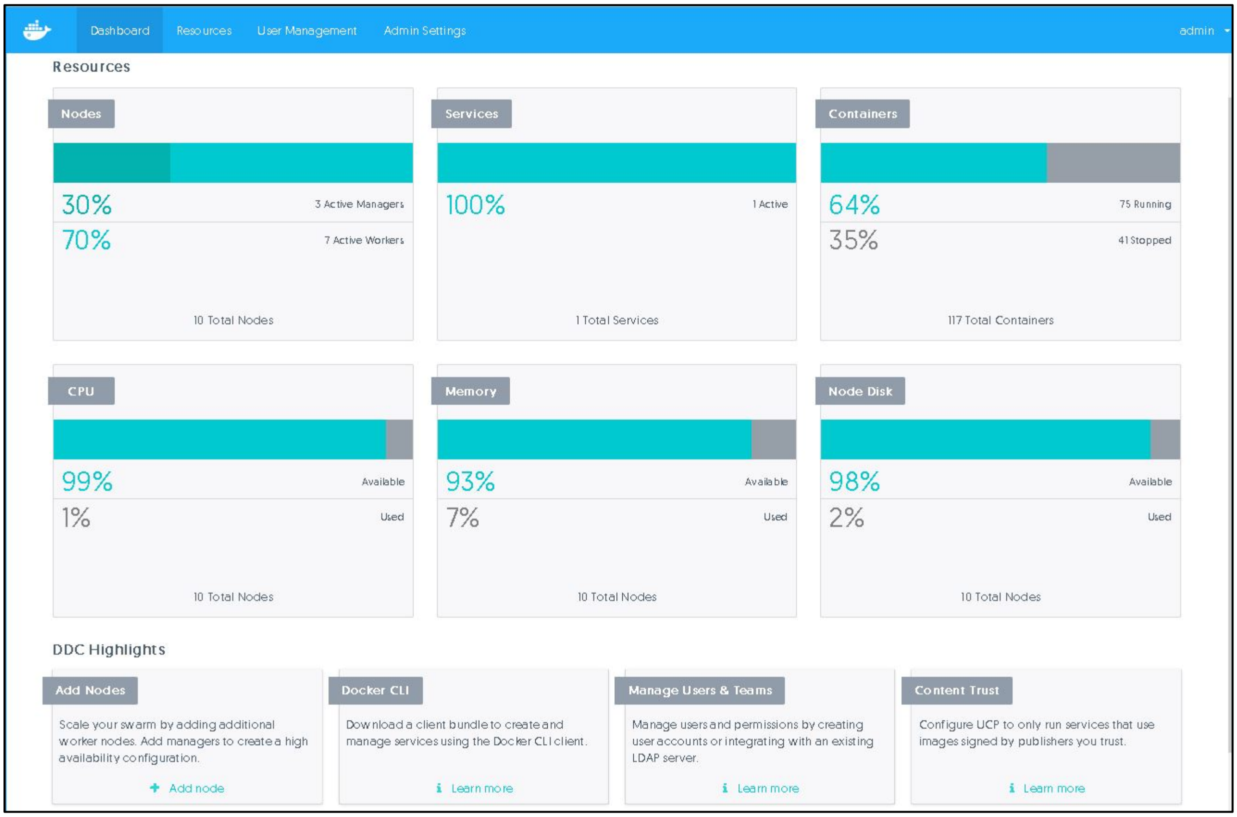
This Reference Configuration includes production grade deployments of the core components of the Docker Enterprise Edition Advanced subscription which are listed in Table 2 above.

Universal Control Plane (UCP), the Docker management layer within Docker Enterprise Edition Advanced, serves as the primary means of managing and monitoring the Docker Enterprise Edition Advanced deployment described in this Reference Configuration. The dashboard shown in Figure 15, provides a high-level overview of the nodes, services and containers running in a UCP cluster. Detailed information is available by exploring each area.

The UCP management interface controls actions through the use or roles and permissions. This enables a common interface for use by different teams.

The UCP management console enables operators to manage and configure the container environment and perform tasks such as:

- Manage worker node deployments
- Deploy services
- Manage service secrets
- Integrate with an external LDAP/AD service
- Manage container storage volumes and networks



Resources

Nodes
30% 3 Active Managers
70% 7 Active Workers
10 Total Nodes

Services
100% 1 Active
1 Total Services

Containers
64% 75 Running
35% 41 Stopped
117 Total Containers

CPU
99% Available
1% Used
10 Total Nodes

Memory
93% Available
7% Used
10 Total Nodes

Node Disk
98% Available
2% Used
10 Total Nodes

DDC Highlights

Add Nodes
Scale your swarm by adding additional worker nodes. Add managers to create a high availability configuration.
Add node

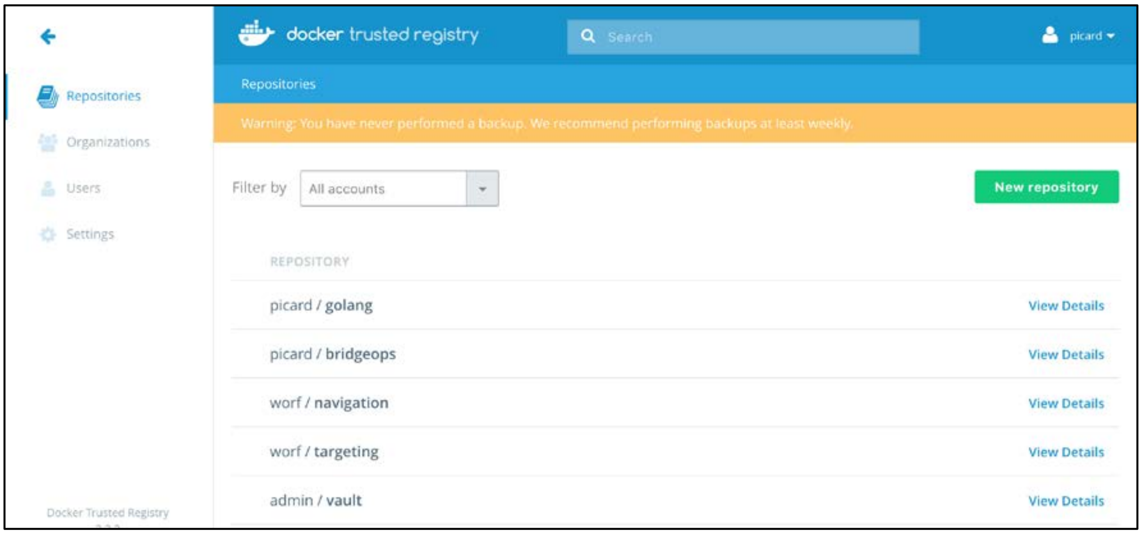
Docker CLI
Download a client bundle to create and manage services using the Docker CLI client.
Learn more

Manage Users & Teams
Manage users and permissions by creating user accounts or integrating with an existing LDAP server.
Learn more

Content Trust
Configure UCP to only run services that use images signed by publishers you trust.
Learn more

Figure 15: Docker UCP dashboard

The on-premises secure image registry that is part of Docker Enterprise Edition Advanced is Docker Trusted Registry (DTR). DTR includes security features such as image signing to enable a secure container lifecycle and image scanning through Docker Security Scanning available as a part of the Docker Enterprise Edition Advanced. It also allows for more efficient resource utilization by allowing for hard delete of orphaned images, as well as processes with image content cache and webhooks.



Warning: You have never performed a backup. We recommend performing backups at least weekly.

REPOSITORY	
picard / golang	View Details
picard / bridgeops	View Details
worf / navigation	View Details
worf / targeting	View Details
admin / vault	View Details

Figure 16: Docker Trusted Registry

The Docker Enterprise Edition Advanced installation for this Reference Configuration consists of three primary types of nodes that can be deployed on a mixture of virtual machines and physical servers:

- UCP manager nodes: The managers schedule container workloads to run on the cluster, and maintain the workload requirements by spinning up containers that have gone down. UCP also provides a routing mesh enabling the traffic to be routed to the correct container node. UCP can be managed from a CLI or a UI.
- DTR worker nodes: Docker Trusted Registry nodes provide a highly available image repository including image scanning services with a management UI.
- UCP swarm worker nodes: These nodes run container workloads.

The Docker UCP swarm manager orchestrates and schedules containers on the entire cluster. Docker Enterprise Edition supports the use of three, five, or seven UCP swarm manager nodes for failover and state preservation.

How Aqua complements Docker EE Advanced

While Docker Enterprise Edition Advanced offers many security features, it is not (nor is it intended to be) a full security platform for container-based applications. From a security standpoint, Docker is primarily concerned with securing the platform itself and its related systems (Docker daemon, registry, control plane, and orchestrator), but are oblivious to the workloads running in containers, i.e., the application layer. Aqua's runtime protection provides active, automated security controls that identify and prevent unauthorized activities, and provide a full audit trail for compliance and forensics.

Note

This section describes the use of Docker EE Advanced. Aqua Security Platform does not require the use of Docker EE Advanced. Aqua Security Platform can be used to secure Docker EE Basic, Standard, or Advanced.

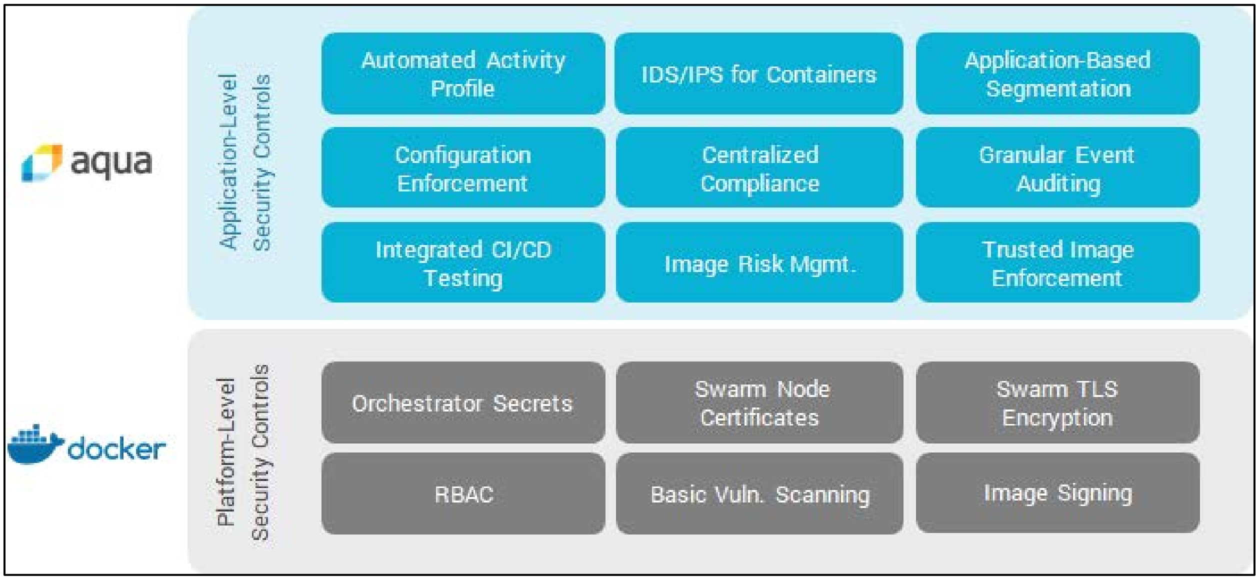


Figure 17: Aqua application security extends Docker platform security

Benefits of using Aqua with Docker

The benefits of using Aqua with Docker include:

- **Full Lifecycle Security:** Secure the entire lifecycle of a containerized application with consistent policies and controls, from image build to container deployment and runtime, with tools that automate and facilitate remediation.
- **Visibility and compliance:** Get single-pane-of-glass view on the state of container compliance anywhere in your environment. Instantly know if you are non-compliant. Get full audit trail of access and security-related events.
- **Active Countermeasures:** Go beyond passive preventive measures, monitoring container behavior in the application context, and alerting on or blocking unpermitted activities.
- **Defense in Depth:** Avoid single point of failure security, ensuring that if any layer was somehow compromised, an attack can be contained and mitigated.
- **Segregation of Duties:** Enable security teams to control the security posture of container deployments, while empowering DevOps teams to automate security into their pipeline.

Configuration guidelines for Docker with Aqua Security

Securing container environments on HPE Synergy with Aqua – Best practices

This section describes best practices for using the Aqua Container Security Platform to secure Docker containers deployed on HPE Synergy.

- Integrate Aqua Security into CI/CD pipeline

  Pushing an image into registry before scanning it will expose the development environment to potential hazards. Organizations should shift left their security controls, implementing them as early as possible in the development process. Integrating Aqua Security into the CI/CD pipeline will ensure that the development process is monitored and secured, keeping the production registry safe.

- Continuous scan of registry

  Aqua Cyber Intelligence is continuously updated from multiple vulnerability sources. Aqua can also scan the registry daily to comply with security best practices to run automated vulnerability scans frequently. This will allow organizations to track the progress of remediation efforts, identify new risks as well as reprioritize the remediation of vulnerabilities based on new intelligence gathered.

- Apply strong defaults

  Organizations should apply strong default security controls on their containers for instant protection of their running containers:

  - Default Runtime profile – Runtime profiles enforce security rules on repositories during runtime. Users should immediately apply a runtime profile to provide protection for containers running from newly pulled images. This can be achieved by creating a generic runtime profile and applying it as the default runtime profile. If a dedicated runtime profile is not attached to a specific repository, Aqua Enforcers will use the default runtime profile.
  - Default Container firewall – Aqua can limit the network connectivity between containers by applying a firewall-like concept for the container environment. This capability allows creating network boundaries across services, where you can control which networks are accessible for each service. Just like the default runtime profile, users should create a default generic container firewall policy to limit container network activity.
  - Threat mitigation – Users should apply threat mitigation controls such as preventing port scanning inside containers, preventing outbound communication from containers to IP addresses with known bad reputation, and preventing fork bombs inside containers.

- Customize least privileges in runtime

  After applying default security controls users can and should optimize them to apply a least privileges security runtime profile. Aqua automatic profiler will monitor and analyze running containers to create a runtime profile based on that container activity. The created profile is based on vital components usage, resources, and network settings, this creates a least privileges security runtime profile which the Aqua administrator can then edit for even more granular security controls.

- Container network segmentation

  In order to “limit the blast radius” in case of an attack, and to prevent lateral movement on the network, it is advised to limit how containers network with each other and with other systems and services. Aqua’s nano-segmentation feature makes this possible by automatically identifying legitimate container network communication and using that information to create firewall rules that operate at the container level, and follow containers regardless of their location.

HPE Synergy configuration

The three HPE Synergy frames used for this Reference Configuration include HA deployment of HPE Synergy Composer and HPE Synergy Image Streamer as shown in Figure 18. A detailed description of this solution is found in HPE Reference Configuration for Docker Enterprise Edition (EE) Standard on HPE Synergy with HPE Synergy Image Streamer.

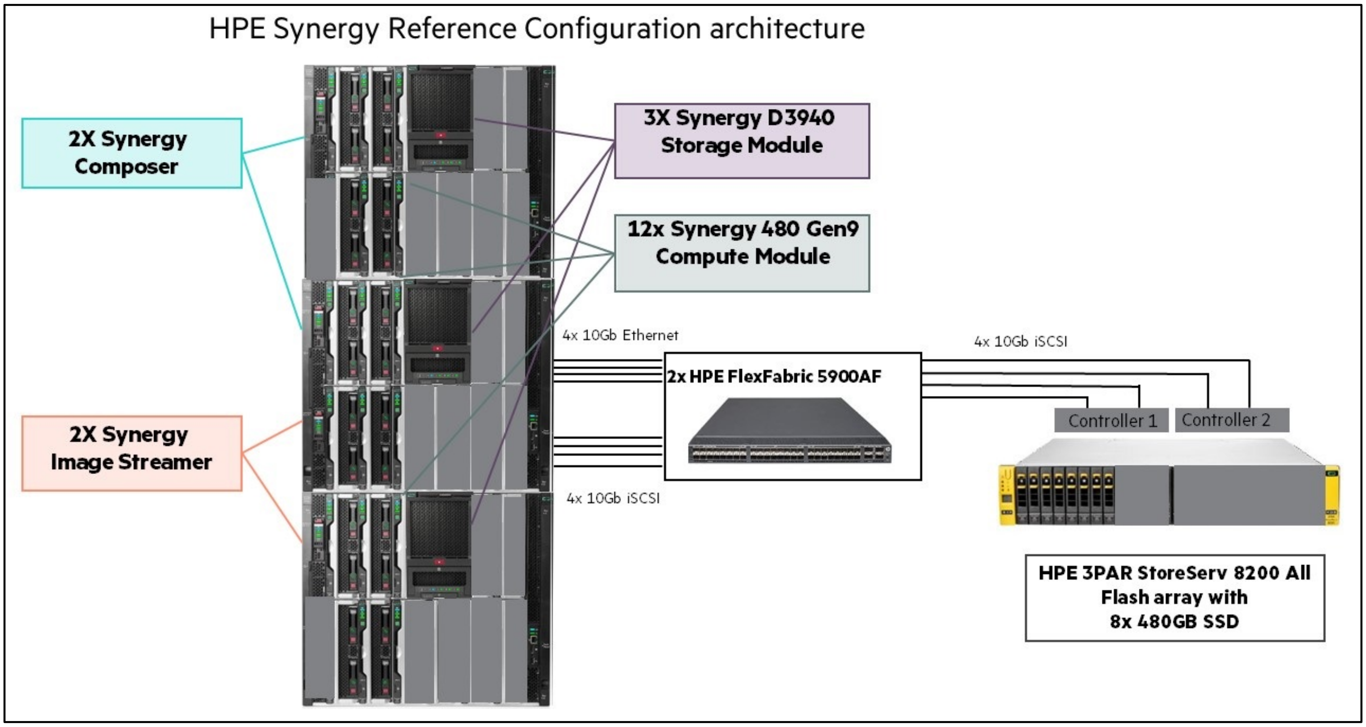

HPE Synergy Reference Configuration architecture
2X Synergy Composer
3X Synergy D3940 Storage Module
12x Synergy 480 Gen9 Compute Module
4x 10Gb Ethernet
4x 10Gb iSCSI
2x HPE FlexFabric 5900AF
4x 10Gb iSCSI
Controller 1
Controller 2
2X Synergy Image Streamer
HPE 3PAR StoreServ 8200 All Flash array with 8x 480GB SSD

Figure 18: HPE Synergy and HPE 3PAR StoreServ 8200 configuration

Best practices for this Reference Configuration

This Reference Configuration extends the configuration described in HPE Reference Configuration for Docker Enterprise Edition (EE) Standard on HPE Synergy with HPE Synergy Image Streamer with the addition of security provided by Aqua. The best practices for Docker Enterprise Edition Standard installation and the use of HPE Image Streamer are described in this document.

Downloading HPE Synergy Image Streamer artifacts for Docker with Aqua Security

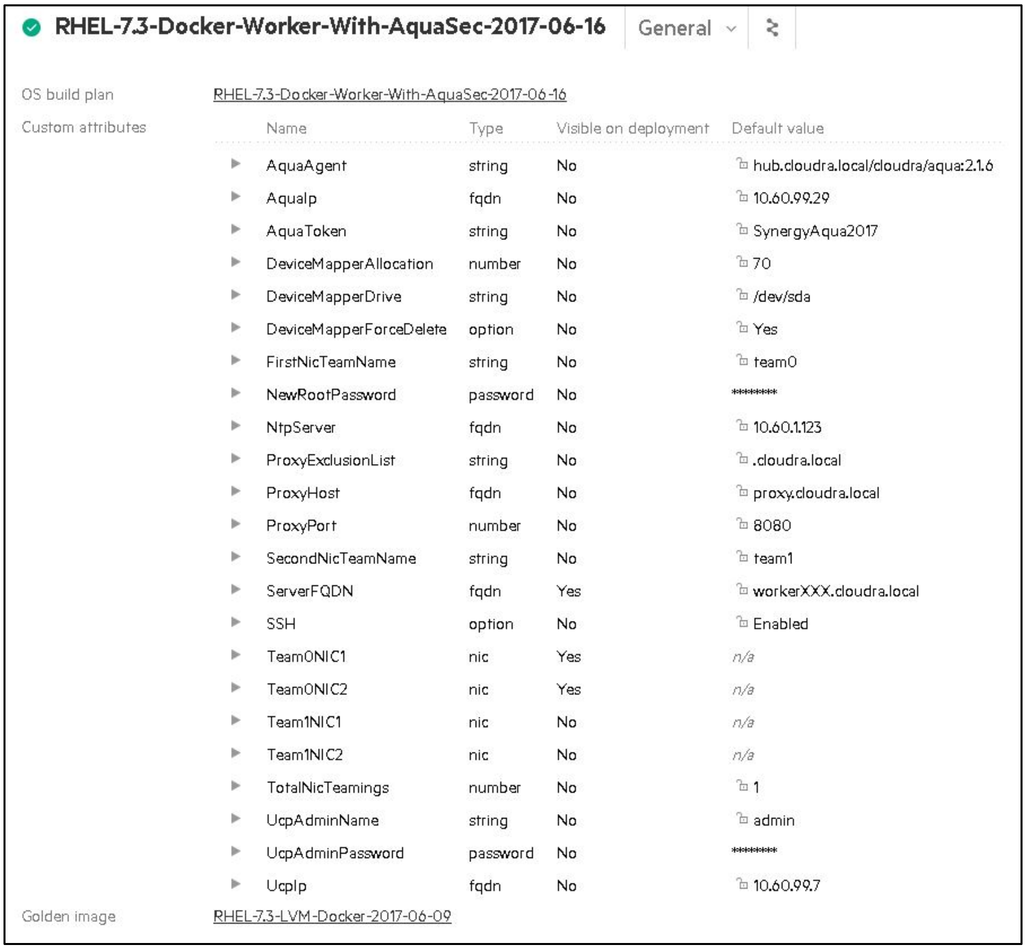
Hewlett Packard Enterprise has created HPE Synergy Image Streamer artifacts and published them to HPE's GitHub location for customer download. The following artifact bundle was used in the development of this Reference Configuration, RHEL-7.3-Docker-Worker-With-AquaSec-2017-06-16.zip

Import the downloaded artifact bundle into HPE Synergy Image Streamer and extract the contents. The artifact bundle includes:

- Deployment plan
- OS build plans
- Plan scripts

Customize the HPE supplied RHEL-7.3-Docker-Worker with Aqua Security deployment plan

The deployment plan for Docker Workers as shown in Figure 19 needs to be copied and customized before it can be used. Make a copy of the plan and edit the values for the Plan Attributes to match the settings for your environment. You will probably want to hide many of the custom attributes and their values so that they are pre-configured when a server profile is created. For example, if you always specify the same proxy server, set this value and hide the corresponding custom attributes and values. The Aqua settings should be pre-configured with the location of the Enforcer image in your local DTR as well as the address of the Command Center and the Enforcer token. Details on the custom attribute definitions and values are found in Appendix B: UCP worker node deployment plan custom attributes. Set the default for the golden image to use your newly-created Red Hat® Enterprise Linux® (RHEL) 7.3 Docker Worker golden image.



RHEL-7.3-Docker-Worker-With-AquaSec-2017-06-16

OS build plan: RHEL-7.3-Docker-Worker-With-AquaSec-2017-06-16

Custom attributes

Name	Type	Visible on deployment	Default value
AquaAgent	string	No	hub.cloudra.local/cloudra/aqua:2.1.6
AquaIp	fqdn	No	10.60.99.29
AquaToken	string	No	SynergyAqua2017
DeviceMapperAllocation	number	No	70
DeviceMapperDrive	string	No	/dev/sda
DeviceMapperForceDelete	option	No	Yes
FirstNicTeamName	string	No	team0
NewRootPassword	password	No	********
NtpServer	fqdn	No	10.60.1.123
ProxyExclusionList	string	No	.cloudra.local
ProxyHost	fqdn	No	proxy.cloudra.local
ProxyPort	number	No	8080
SecondNicTeamName	string	No	team1
ServerFQDN	fqdn	Yes	workerXXX.cloudra.local
SSH	option	No	Enabled
Team0NIC1	nic	Yes	n/a
Team0NIC2	nic	Yes	n/a
Team1NIC1	nic	No	n/a
Team1NIC2	nic	No	n/a
TotalNicTeamings	number	No	1
UcpAdminName	string	No	admin
UcpAdminPassword	password	No	********
UcpIp	fqdn	No	10.60.99.7

Golden image: RHEL-7.3-LVM-Docker-2017-06-09

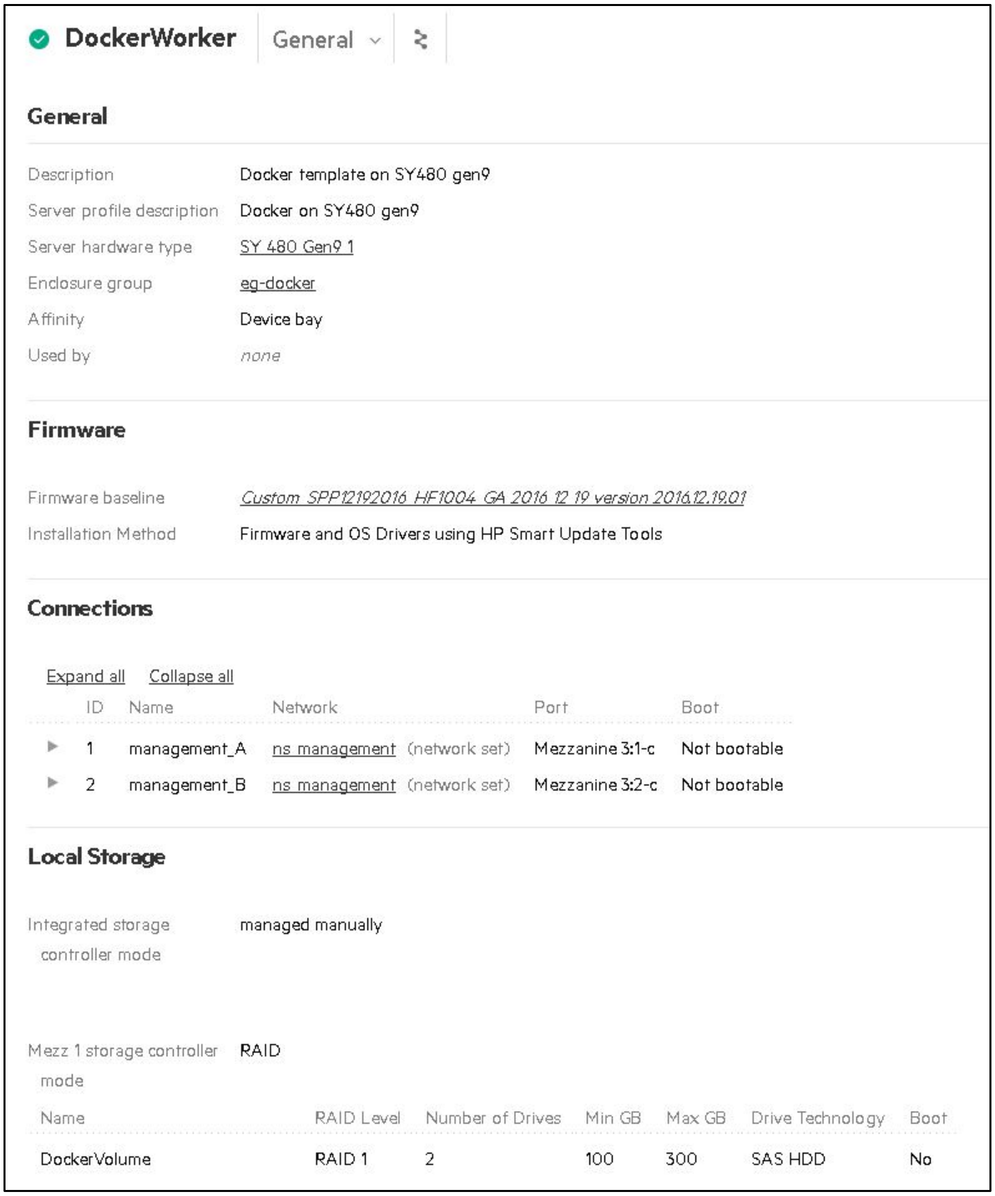
Figure 19: HPE Synergy Image Streamer deployment plan for Docker worker plus Aqua Enforcer custom attributes

Create server profiles in HPE OneView

It is a best practice to create a server profile template for your Docker Workers and use this to create the server profiles. By using the Server Profile Template feature within HPE OneView you can specify and maintain a single configuration for the system firmware, BIOS, and boot-order at time of initial deployment as well as orchestrate updates to that configuration as needed. This provides a location to centrally manage and update configuration settings, such as system firmware, and provides assurance that each server is running with the same configuration and has event and health data being exposed up to HPE OneView.

A server profile template for Docker Workers is shown in Figure 20. The local storage shown as **DockerVolume** is the storage on the HPE Synergy D3940. The storage controller mode must be set to RAID to configure redundant storage with RAID level 1. This storage will be used for Docker containers.

The Connections section in the server profile only includes the management network set used by Docker. **Do not** include the iSCSI network used for Image Streamer in the profile template. This network will automatically be included for you when the server profile is created.



DockerWorker

General

Description	Docker template on SY480 gen9
Server profile description	Docker on SY480 gen9
Server hardware type	SY 480 Gen9 1
Enclosure group	eg-docker
Affinity	Device bay
Used by	*none*

Firmware

Firmware baseline	*Custom SPP12192016 HF1004 GA 2016 12 19 version 2016.12.19.01*
Installation Method	Firmware and OS Drivers using HP Smart Update Tools

Connections

Expand all Collapse all

ID	Name	Network	Port	Boot
1	management_A	ns management (network set)	Mezzanine 3:1-c	Not bootable
2	management_B	ns management (network set)	Mezzanine 3:2-c	Not bootable

Local Storage

Integrated storage controller mode	managed manually
Mezz 1 storage controller mode	RAID

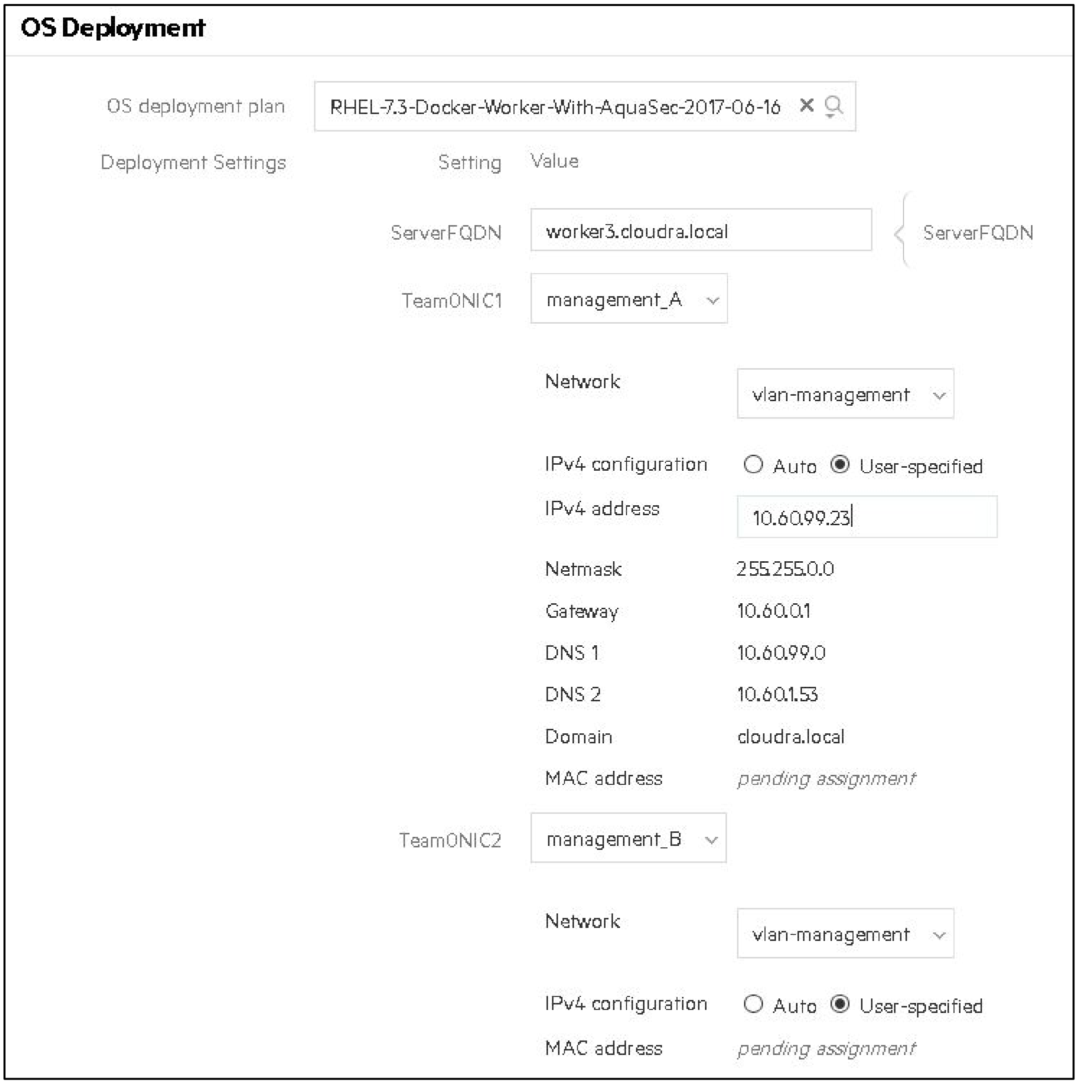
Name	RAID Level	Number of Drives	Min GB	Max GB	Drive Technology	Boot
DockerVolume	RAID 1	2	100	300	SAS HDD	No

Figure 20: HPE OneView Server Profile Template for Docker Worker

Once the server profile template creation is complete it can be used to create server profiles to deploy your Docker Workers.

- Create a new server profile from the Docker Worker template you created.
- Specify the OS deployment plan you created in Customize the HPE supplied RHEL-7.3-Docker-Worker with Aqua Security deployment plan
- Set all custom attributes you have made visible when you created the deployment plan. Minimally you will need to set:
  - The ServerFQDN as the fully qualified domain name of the worker node
  - The values for the Team0NIC1 and Team0NIC2 to the names of the Docker Ethernet Traffic networks with a User-specified static IP address.

Note
At the time of publication there is a restriction to select a User Specified static IP address for the management network which will allow for dynamically changing the OS deployment plan when updates to golden images or other deployment scripts are needed.



OS Deployment

OS deployment plan: RHEL-7.3-Docker-Worker-With-AquaSec-2017-06-16

Deployment Settings

Setting	Value	
ServerFQDN	worker3.cloudra.local	ServerFQDN
Team0NIC1	management_A	

Team0NIC1:

Network: vlan-management

IPv4 configuration: ○ Auto ● User-specified

IPv4 address: 10.60.99.23

Netmask: 255.255.0.0

Gateway: 10.60.0.1

DNS 1: 10.60.99.0

DNS 2: 10.60.1.53

Domain: cloudra.local

MAC address: *pending assignment*

Team0NIC2: management_B

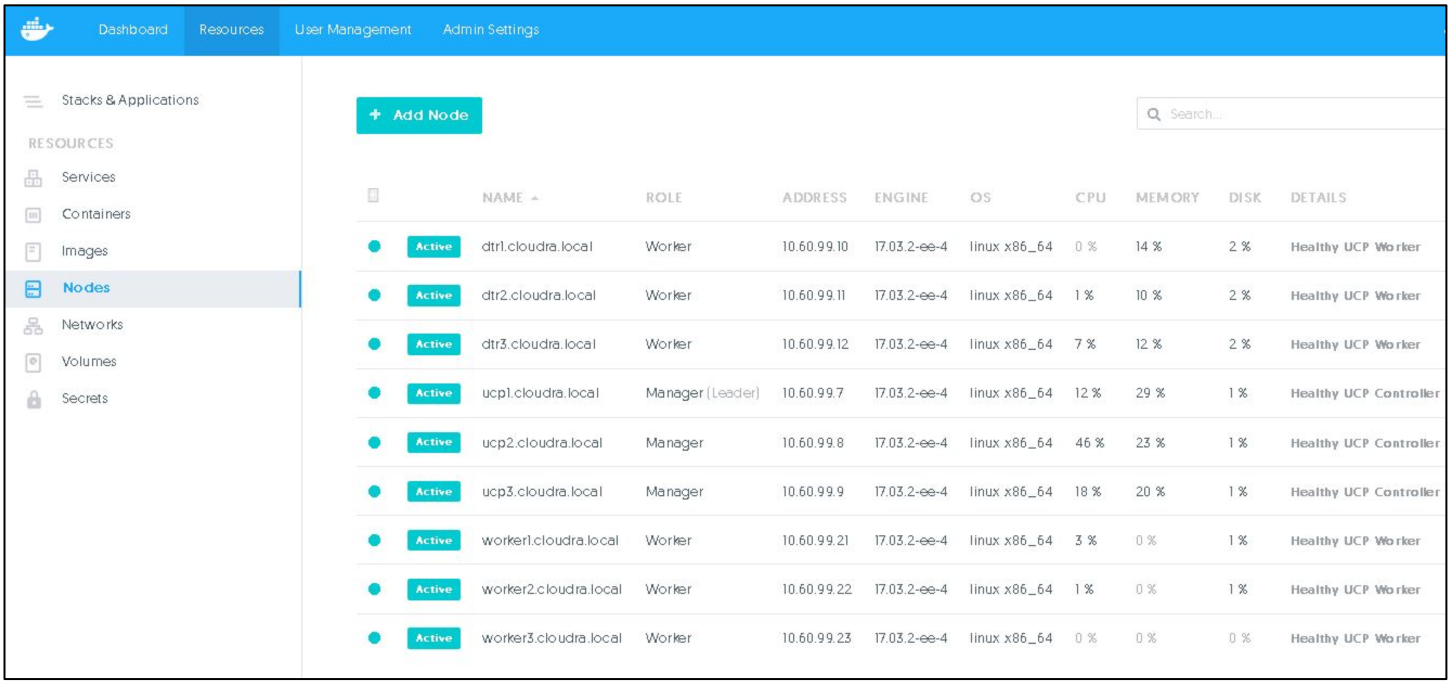
Network: vlan-management

IPv4 configuration: ○ Auto ● User-specified

MAC address: *pending assignment*

Figure 21: Docker Worker server profile creation

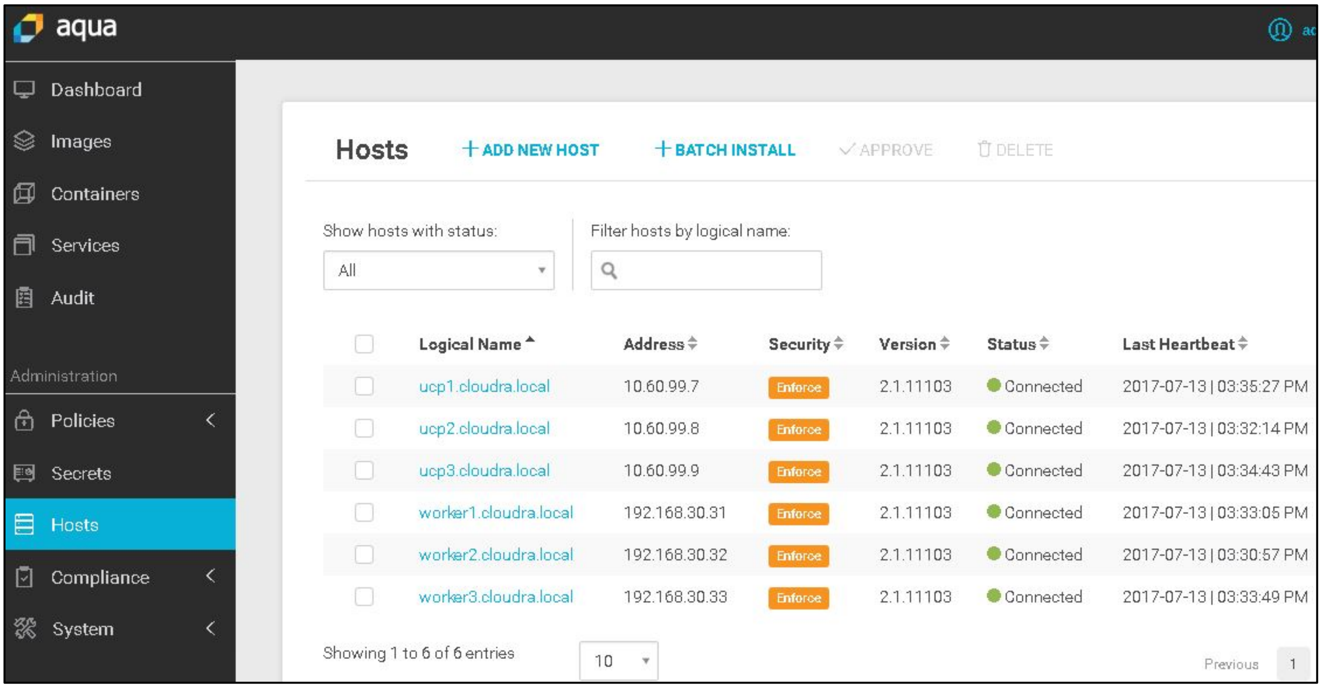
Once the server profile is assigned to an available server, the deployment process will begin. A smart clone of the golden image is created as a new volume and presented to the server. Power up the server. The server will automatically boot the newly created volume. The build plan and plan scripts that are part of the RHEL-7.3-Docker-Worker deployment plan will automatically customize the server with all the networking, storage connectivity, firewall, and other host settings. The plan scripts automate the completion of the Docker Worker configuration and joining the swarm cluster. The plan scripts also deploy the Aqua Enforcer which registers the new host with the Aqua Command Center. Once the scripts have completed, the newly added Docker Worker is visible in Docker UCP and available to host new container workloads as shown in Figure 22.



		NAME	ROLE	ADDRESS	ENGINE	OS	CPU	MEMORY	DISK	DETAILS
●	Active	dtr1.cloudra.local	Worker	10.60.99.10	17.03.2-ee-4	linux x86_64	0 %	14 %	2 %	Healthy UCP Worker
●	Active	dtr2.cloudra.local	Worker	10.60.99.11	17.03.2-ee-4	linux x86_64	1 %	10 %	2 %	Healthy UCP Worker
●	Active	dtr3.cloudra.local	Worker	10.60.99.12	17.03.2-ee-4	linux x86_64	7 %	12 %	2 %	Healthy UCP Worker
●	Active	ucp1.cloudra.local	Manager (Leader)	10.60.99.7	17.03.2-ee-4	linux x86_64	12 %	29 %	1 %	Healthy UCP Controller
●	Active	ucp2.cloudra.local	Manager	10.60.99.8	17.03.2-ee-4	linux x86_64	46 %	23 %	1 %	Healthy UCP Controller
●	Active	ucp3.cloudra.local	Manager	10.60.99.9	17.03.2-ee-4	linux x86_64	18 %	20 %	1 %	Healthy UCP Controller
●	Active	worker1.cloudra.local	Worker	10.60.99.21	17.03.2-ee-4	linux x86_64	3 %	0 %	1 %	Healthy UCP Worker
●	Active	worker2.cloudra.local	Worker	10.60.99.22	17.03.2-ee-4	linux x86_64	1 %	0 %	1 %	Healthy UCP Worker
●	Active	worker3.cloudra.local	Worker	10.60.99.23	17.03.2-ee-4	linux x86_64	0 %	0 %	0 %	Healthy UCP Worker

Figure 22: Docker UCP view of automatically provisioned Docker workers

The new Docker Worker is automatically deployed with the Aqua Enforcer and the policies defined in the Aqua Management console are automatically enforced on the new worker node as shown in Figure 23. Note as shown below that the Aqua Enforcer is installed on all worker nodes as well as on the UCP nodes. It is optional to install Aqua Enforcer on the DTR nodes.



Show hosts with status: All

Filter hosts by logical name:

□	Logical Name	Address	Security	Version	Status	Last Heartbeat
□	ucp1.cloudra.local	10.60.99.7	Enforce	2.1.11103	● Connected	2017-07-13 | 03:35:27 PM
□	ucp2.cloudra.local	10.60.99.8	Enforce	2.1.11103	● Connected	2017-07-13 | 03:32:14 PM
□	ucp3.cloudra.local	10.60.99.9	Enforce	2.1.11103	● Connected	2017-07-13 | 03:34:43 PM
□	worker1.cloudra.local	192.168.30.31	Enforce	2.1.11103	● Connected	2017-07-13 | 03:33:05 PM
□	worker2.cloudra.local	192.168.30.32	Enforce	2.1.11103	● Connected	2017-07-13 | 03:30:57 PM
□	worker3.cloudra.local	192.168.30.33	Enforce	2.1.11103	● Connected	2017-07-13 | 03:33:49 PM

Showing 1 to 6 of 6 entries

Figure 23: Aqua automatically manages newly deployed worker nodes

Node maintenance, upgrade best practices and performance benefits of HPE Image Streamer

Best practices for node maintenance and upgrade as well as performance benefits of HPE Image Streamer are detailed in HPE Reference Configuration for Docker Enterprise Edition (EE) Standard on HPE Synergy with HPE Synergy Image Streamer.

Summary

This document has described the benefits of integrating Aqua Security with Docker Enterprise Edition Advanced on HPE Synergy using HPE OneView and HPE Synergy Image Streamer. HPE Synergy provides an ideal platform to deploy Docker with Aqua Security. This integration of security from Aqua into the Docker environment secures the entire lifecycle of containerized applications from image build to container deployment and runtime actively remediating threats as they are identified. Aqua provides active monitoring and prevention capabilities that provide defense in depth, and prevent misuse or breach of the containerized application by both unauthorized and privileged users.

Customers deploying Docker containers on large scale environments should consider HPE Synergy with Aqua Security as the secure deployment infrastructure for their enterprise solutions.

Appendix A: Bill of materials

The following BOMs contain electronic license to use (E-LTU) parts. Electronic software license delivery is now available in most countries. HPE recommends purchasing electronic products over physical products (when available) for faster delivery and for the convenience of not tracking and managing confidential paper licenses. For more information, please contact your reseller or an HPE representative.

Note

Part numbers are at time of publication and subject to change. The bill of materials does not include complete support options or other rack and power requirements. If you have questions regarding ordering, please consult with your HPE Reseller or HPE Sales Representative for more details. hpe.com/us/en/services/consulting.html

Table 5: Bill of materials

Quantity	Part number	Description
		Rack and network infrastructure
1	BW908A	HPE 42U 600x1200mm Enterprise Shock Rack
4	AF522A	HPE Intelligent 8.6kVA/L15-30P/NA/J PDU
1	HC790A	HPE Integration Center Routg Service FIO
1	BW932A	HPE 600mm Rack Stabilizer Kit
1	BW909A	HPE 42U 1200mm Side Panel Kit
1	JG505A	HPE 59xx CTO Switch Solution
2	JG510A	HPE 5900AF 48G 4XG 2QSFP+ Switch
4	JD096C	HPE X240 10G SFP+ SFP+ 1.2m DAC Cable
2	JC680A	HPE 58x0AF 650W AC Power Supply
2	JC682A	HPE 58x0AF Bck(pwr) Frt(prt) Fan Tray
		Synergy 12000 Frame components
3	797740-B21	HPE Synergy12000 CTO Frame 1xFLM 10x Fan
3	798096-B21	HPE Synergy 12000F 6x 2650W AC Ti FIO PS
2	804353-B21	HPE Synergy Composer
3	804942-B21	HPE Synergy Frame Link Module
1	804938-B21	HPE Synergy 12000 Frame Rack Rail Option
1	804943-B21	HPE Synergy 12000 Frame 4x Lift Handle
18	TK738A	HPE 2.0m 250V 16A C19-C20 Sgl IPD Jpr Crd
2	804937-B21	HPE Synergy Image Streamer
		Synergy 480 Gen9 Compute Module components
12	732352-B21	HPE SY 480 Gen9 CTO Cmpt Mdl
12	826994-B21	HPE Synergy 480 Gen9 E5-2683v4 Kit
12	826994-L21	HPE Synergy 480 Gen9 E5-2683v4 Kit
96	805351-B21	HPE 32GB 2Rx4 PC4-2400T Memory
12	759557-B21	HPE Smart Array P542D/2GB Controller
12	814068-B21	HPE Smart Array P240nr/1GB Controller
12	777430-B21	HPE Synergy 3820C 10/20Gb CNA
12	782958-B21	HPE 96W Smart Stor Battery 260mm Cbl Kit

Quantity	Part number	Description
		Synergy fabric components
2	794502-B23	HPE VC SE 40Gb F8 Module
4	779218-B21	HPE Synergy 20Gb Interconnect Link Mod
6	755985-B21	HPE Synergy 12Gb SAS Connection Module
		Synergy composable storage components
3	835386-B21	HPE Synergy D3940 CTO Storage Module
3	757323-B21	HPE Synergy D3940 IO Adapter
30	785067-B21	HPE 300GB 12G SAS 10K 2.5in SC ENT HDD
		Cables and transceivers
8	804101-B21	HPE Synergy Interconnect Link 3m AOC
2	720199-B21	HPE BLc 40G QSFP+ QSFP+ 3m DAC Cable
8	720193-B21	HPE BLc QSFP+ to SFP+ Adapter
8	455883-B21	HPE BLc 10G SFP+ SR Transceiver
8	AJ837A	HPE 15m Multi-mode OM3 LC/LC FC Cable
9	861412-B21	HPE CAT6A 4ft Cbl
2	838327-B21	HPE Synergy Dual 10GBASE-T QSFP+ 30m RJ45 Transceiver
		HPE 3PAR StoreServ 8200 with iSCSI adapters and accessories
1	K2Q36B	HPE 3PAR 8200 2N+SW Storage Field Base
2	H6Z10A	HPE 3PAR 8000 2-pt 10Gb iSCSI/FCoE Adptr
8	K2P88B	HPE 3PAR 8000 480GB+SW Non-AFC SFF SSD
1	HA114A1	HPE Installation and Startup Service
1	HA114A1 5XU	HPE Startup 3PAR 8200 2N Fld Int Bas SVC
1	K2R29A	HPE 3PAR StoreServ RPS Service Processor
1	H1K92A3	HPE 3Y Proactive Care 24x7 Service
1	H1K92A3 W3G	HPE 3PAR 8200 2N+SW Storage Base Support
8	H1K92A3 X8G	HPE 3PAR 8000 480GB+SW LFF SSD Supp
1	H1K92A3 YNW	HPE 3PAR StoreServ RPS Service Proc Supp
2	H1K92A3 YTN	HPE 3PAR 8000 2-pt 10Gb FCoE Adptr Supp
1	L7F20AAE	HPE 3PAR All-in S-sys SW Current E-Media
1	L7F22AAE	HPE 3PAR All-in M-sys SW Current E-Media
1	C7535A	HPE RJ45 to RJ45 Cat5e Black M/M 7.6ft 1-pack Data Cable
8	H0JD6A1	HPE 3PAR SSD Extended Replacement SVC
1	HA124A1	HPE Technical Installation Startup SVC
1	HA124A1 5QW	HPE Startup 3PAR Vrt Cpy Lvl1 Tier 1 SVC
1	HA124A1 56X	HPE Startup 3PAR 8K Mlt Sys PM PP RC SVC
		VMware license for management cluster
6	BD715AAE	VMw vSphere EntPlus 1P 3yr E-LTU

Alternate solution components

Table 6: Alternate Docker licensing options (pick one) for the HPE Synergy 480 servers

Quantity	Part number	Description
15	Q7D86AAE	HPE Docker Ent Adv 1yr 9x5 E-LTU
15	Q7D87AAE	HPE Docker Ent Adv 3yr 9x5 E-LTU
15	Q7D88AAE	HPE Docker Ent Adv 1yr 24x7 E-LTU
15	Q7D89AAE	HPE Docker Ent Adv 3yr 24x7 E-LTU

Note

Each server or VM installed with the Docker EE Engine requires a Docker Enterprise Edition license. Aqua Security Platform can secure Docker EE Basic, Standard or Advanced. Docker Enterprise Edition Advanced was used for this Reference Configuration. At minimum, 15 Docker Enterprise Edition licenses are required for the UCP deployment as outlined in this Reference Configuration, as six Docker Enterprise Edition Advanced licenses are required for the management VMs (UCP swarm managers and DTR replicas) and nine licenses are required for the UCP swarm workers.

Aqua Security Platform licensing

The Aqua licensing model is based on the number of Aqua Enforcers in use. Each Enforcer provides a full set of functionalities to ensure the security of the entire container lifecycle – from production to development. The Aqua Enforcer must be deployed on every host that needs to be protected by Aqua including UCP nodes. Deploying Aqua Enforcer on DTR nodes is optional. Licenses can be purchased from aquasec.com/hpe.

Appendix B: UCP worker node deployment plan custom attributes

The RHEL-7.3-Docker-Worker-2017-04-18 deployment plan provided in the downloaded artifact bundle requires changes to the custom attributes used to deploy the Docker worker node. The following custom attributes are included:

Table 7: Custom Attributes for Docker worker deployment

Name	Description	Visible	Sample value
AquaAgent	Path to the Docker Trusted Registry repository containing the Aqua Enforcer	Yes	hub.cloudra.local/cloudra/aqua:2.1.6
AquaIP	Fully qualified domain name or IP address of the Aqua Management server	Yes	10.10.10.101
AquaToken	The installation token specified during Aqua Manager configuration which allows for new host batch installation. This is the BATCH_INSTALL_TOKEN supplied during Aqua Manager installation.	Yes	
DeviceMapperAllocation	Percentage of the D3940 disk drive which is allocated for Docker volume images. The rest of the storage is available for the Docker root directory and can be used for volume creation. It should be noted that the volumes created on local storage are only available to containers running on the same host.	Yes	70
DeviceMapperDrive	The filesystem location for the D3940 data drive.	Yes	/dev/sda
DeviceMapperForceDelete	Forces a disk wipe so logical volumes can be re-used. This custom attribute is hidden and must be set to yes or the Docker containers may not load properly.	No	yes
FirstNicTeamName	Unique name for the NIC Team used for Docker networking. This name will be used by the plan scripts to create the NIC team.	Yes	Team0
NewRootPassword	Password to be configured for the root user. Note that when you assign the server profile you need to re-enter the password for security purposes.	Yes	
NtpServer	IP address for the NTP server. An NTP server is required or communication between containers could fail.	Yes	10.10.10.100

Name	Description	Visible	Sample value
ProxyExclusionList	A comma separated list of domain names which will be used to create the no_proxy environment variable on the deployed server. If you don't have a proxy in your environment, or use a transparent proxy, edit the OS build plan and remove the step which configures the proxy (070-Docker-configure-proxy).	Yes	
ProxyHost	Proxy host IP address if needed. If no proxy is needed, remove the plan script (070-Docker-configure-proxy) from the OS build plan. Do not include http in the ProxyHost name.	Yes	proxy.acme.com
ProxyPort	Proxy port used with the proxy host.	Yes	8080
SecondNicTeamName	Reserved for future use.	Yes	
ServerFQDN	Fully qualified domain name	Yes	server1.acme.com
SSH	Enable the SSH service and root login over SSH. The default is Enabled.	Yes	enabled
Team0NIC1	A static IP address for the Docker worker host on your production network. Select Static User Assigned for this address.	Yes	
Team0NIC2	This NIC is used for teaming. Leave the default value of static. No IP address is needed.	Yes	
Team1NIC1	Reserved for future use.	No	
Team1NIC2	Reserved for future use.	No	
TotalNicTeamings	The total number of NIC teams set by the plan scripts. Leave this value set to 1.	No	1
UcpAdminName	The administrator account for accessing UCP	Yes	admin
UcpAdminPassword	Password for UCP administrator user	Yes	
UcpIp	IP address for UCP Load Balancer	Yes	

As noted in Table 7 above, the OS build plan delivered as part of the artifact bundle for Docker includes a step to configure the proxy host. If a proxy host is not required in your environment, remove XXX-Docker-configure-proxy from your OS build plan, (XXX = 070 at the time of writing).

Appendix C: Configuring the Aqua Management Console and Enforcer

The system requirements for deploying the Aqua Management console can be found in the Aqua documentation. Note that this site is restricted and only available when a license is purchased. For this Reference Configuration, the Quick-Start Aqua Console Installation was followed. The console, database and gateway components are deployed as Docker containers on a single virtual machine.

Installation of the Aqua Management console components requires:

- Docker 1.10 or later
- 4 GB RAM
- 3 GB available disk space. Additional disks might be needed, depending on the scanned image sizes.
- Database: Postgres 9.5.x, either already available, or the one supplied as a container during Aqua console installation.

Installation of the Aqua Enforcer requires:

- Docker 1.10 or later
- 150 MB RAM
- 0.03 CPU cores

The automation for Aqua Enforcer container installation expects that the Aqua Enforcer image has been pushed to your local DTR. By pulling the Aqua Enforcer to your local DTR, you eliminate the need to supply credentials in the Synergy Image Streamer plan scripts to pull the Enforcer during deployment from the external Aqua Docker repository. In order to automate deployment of the Aqua Enforcer, the Aqua console must be configured to allow automated installation of Aqua Enforcers. Change the default docker-compose.yml settings for BATCH_INSTALL prior to installing the Aqua console. A sample setting is shown in the example below.

```
### Modify below lines to create a batch install profile for Aqua Enforcer
  - BATCH_INSTALL_TOKEN=SynergyAqua2017
  - BATCH_INSTALL_NAME=SynergyAgents
  - BATCH_INSTALL_GATEWAY=aqua.cloudra.local
  - BATCH_INSTALL_ENFORCE_MODE=y
```

The parameters are configured as follows:

- BATCH_INSTALL_TOKEN: The value entered will be supplied as a custom attribute in the Image Streamer plan script used to deploy the Enforcer. This is the AquaToken in Table 7.
- BATCH_INSTALL_NAME: Name of the Host Batch Install. This can be any value.
- BATCH_INSTALL_GATEWAY: The FQDN or IP address of the server hosting the Aqua gateway container. In our Reference Configuration this is the same as the hostname for the virtual machine hosting the Aqua management console.
- BATCH_INSTALL_ENFORCE_MODE: Set to y or n to set the security mode.

Note
The Host Batch Install settings shown above can also be set or modified within the Aqua console after deployment.

Deployment of the Aqua Enforcer has been automated in the Image Streamer plan scripts created for this solution. The plan scripts require that the Aqua Enforcer image is pulled down from the Aqua repository and stored in a local DTR. The location of the image in the local DTR is defined as a configuration attribute for the Image Streamer plan script.

Deployment of the Aqua Enforcers is automated by an Image Streamer plan script. The plan script requires the following Custom Attributes:

- AquaIp: The IP Address for the Aqua Management console.
- AquaToken: The value specified in the BATCH_INSTALL_TOKEN as described above.
- AquaAgent: The location of the Aqua Enforcer in the local DTR. For example dtr.cloud.local/cloudra/aqua/2.1.6

When a Docker Worker is deployed, the Aqua Enforcer is automatically deployed and the newly added host is visible from the Aqua console as shown in Figure 23.

Resources and additional links

Aqua Container Security Platform, aquasec.com/products/aqua-container-security-platform

Docker Enterprise Edition, http://docs.docker.com/enterprise

Docker Trusted Registry documentation, http://docs.docker.com/datacenter/dtr/2.2/guides

Docker Universal Control Plane documentation, http://docs.docker.com/datacenter/ucp/2.1/guides

Docker Reference Architecture: Universal Control Plane 2.0 Service Discovery and Load Balancing, http://success.docker.com/Architecture/Docker_Reference_Architecture%3A_Universal_Control_Plane_2.0_Service_Discovery_and_Load_Balancing

HA Proxy, haproxy.org

HPE GitHub Reference Architectures, https://github.com/HewlettPackard/image-streamer-reference-architectures

HPE Networking, hpe.com/networking

HPE OneView, hpe.com/oneview

HPE Reference Architectures, hpe.com/info/ra

HPE Reference Configuration for securing Docker on HPE hardware, http://h20195.www2.hpe.com/V2/GetDocument.aspx?docname=a00020437enw

HPE Servers, hpe.com/servers

HPE Storage, hpe.com/storage

HPE Synergy, hpe.com/synergy

HPE Technology Consulting Services, hpe.com/us/en/services/consulting.html

To help us improve our documents, please provide feedback at hpe.com/contact/feedback

Sign up for updates