ITSM Automation NG Express

Software release version: 2017.04

Get Started

Document release date: April 2017
Product release date: April 2017

Hewlett Packard Enterprise

Legal Notices

Warranty

The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted rights legend

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

Documentation updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to: https://softwaresupport.hpe.com/.

This site requires that you register for an HPE Passport and to sign in. To register for an HPE Passport ID, click **Register** on the HPE Software Support site or click **Create an Account** on the HPE Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HPE sales representative for details.

Support

Visit the HPE Software Support site at: https://softwaresupport.hpe.com.

This website provides contact information and details about the products, services, and support that HPE Software offers.

HPE Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support website to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HPE Support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HPE Passport user and to sign in. Many also require a support contract. To register for an HPE Passport ID, click **Register** on the HPE Support site or click **Create an Account** on the HPE Passport login page.

To find more information about access levels, go to: https://softwaresupport.hpe.com/web/softwaresupport/access-levels.

HPE Software Solutions Now accesses the HPESW Solution and Integration Portal website. This site enables you to explore HPE Product Solutions to meet your business needs, includes a full list of Integrations between HPE Products, as well as a listing of ITIL Processes. The URL for this website is https://softwaresupport.hpe.com/.

Contents

1 Overview of ITOM Container Deployment Foundation 4
1.1 New and agile software delivery platform 4
1.2 Architecture 4
2 Overview of ITSM Automation (ITSMA) 6
2.1 Capabilities 6
2.2 Containerization 8
3 Personas 9
3.1 ITOM CDF personas 9
3.2 ITSMA personas 9
4 Language support 11
5 Glossary 12

This section provides resources to help you understand ITOM Container Deployment Foundation (CDF) and the ITSMA suite (ITSMA NG Express).

- Overview of ITOM Container Deployment Foundation
- Overview of ITSM Automation (ITSMA)
- Personas
- Language support
- Glossary

Overview of ITOM Container Deployment Foundation

The ITSMA suite must be deployed on HPE ITOM Container Deployment Foundation (ITOM CDF), which powers the deployment of the following container-based software suites and and drives significantly the overall time to value of customers:

- IT Service Management Automation (ITSMA)
- IT Event Correlation and Management (OpsBridge)
- Data Center Automation (DCA)
- Helion Cloud Management (HCM)

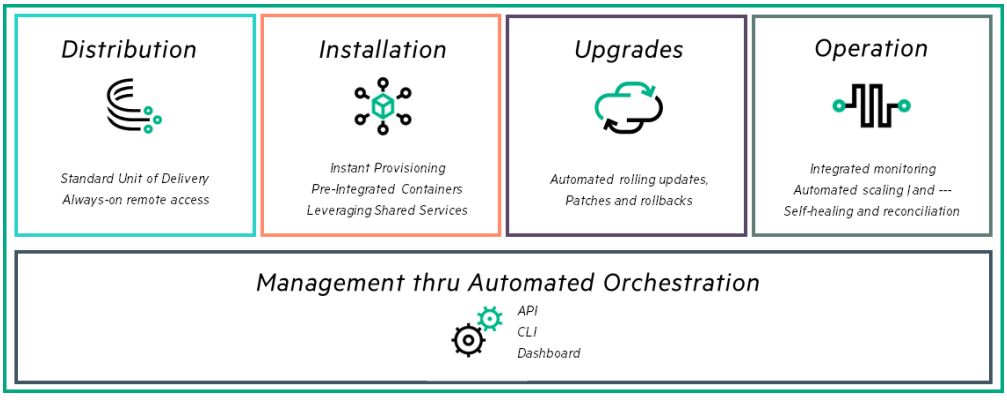
> For a better understanding of the terms used in this help center, see Glossary.

New and agile software delivery platform

The installation and deployment of complex software packages in an IT environment has been traditionally very complex and expensive, involving many teams and additional resources, such as professional services. Time is also consumed for any post-deployment activities such as integration work. Upgrades have also been very slow to be adopted with very large regression cycles gated by many environments until updates finally are pushed to your production environment.

By building a new and agile software delivery platform, alongside modernized software, ITOM CDF allows customers to install pre-integrated suite capabilities. Not only Day-1 type of operations have been resolved immediately due to the nature of a container based infrastructure, the same platform allows for easier access and deployment and operation of subsequent upgrades.

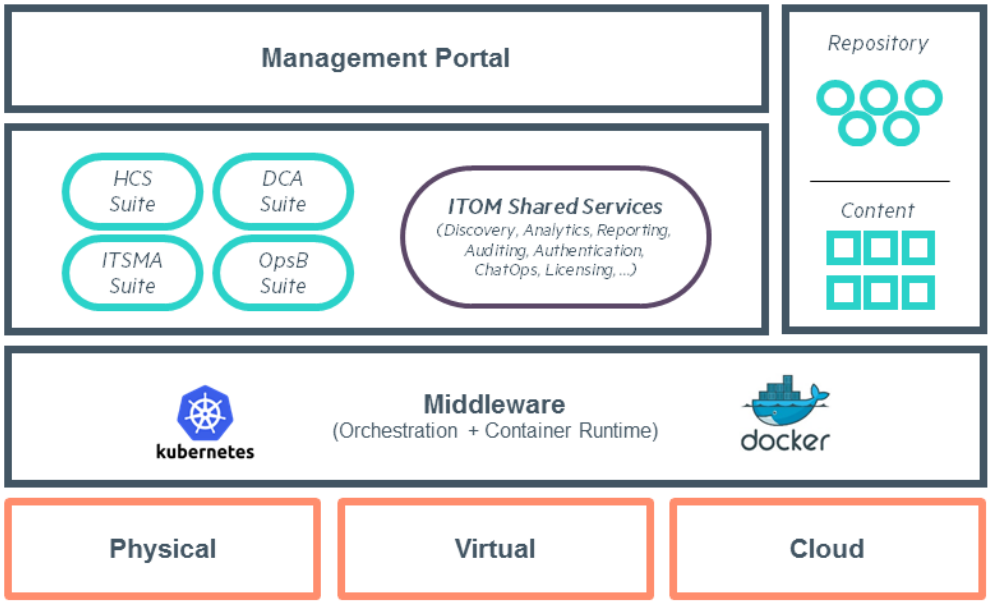
The distribution unit of the software delivery is container-based, leveraging the speed and format of the new containerized environment. By bundling an orchestration layer to bootstrap and manage the lifecycle of many suite-related containers, you are able to standardize on deployment, upgrade and patching, scaling and rollbacks.



Architecture

ITOM CDF is a container deployment foundation that is built to run on many environments, offering the right level of flexibility. Deploy your IT Operations Management suite either on a bare metal or a virtual environment. ITOM CDF is using cloud native toolset, such as Docker technologies and Kubernetes from the Cloud Native Computing Foundation (CNCF) that allows the management of container-based applications at scale.

The following figure depicts the ITOM CDF architecture.



This foundation comes with additional services, including:

- A secured configuration store powered by **Vault**
- A networking layer managed by **Flannel**
- A distributed configuration database - **etcd** (runs on all of the Docker hosts) provides a unified view - file system type database to store the configuration information for the container platform.
- **A set of shared services** such as Identity Management – to plug against an existing enterprise authentication store such as LDAP or Microsoft Director, or the Licensing service to track license consumption throughout your suite
- A **Repository and Content** server. A local Container registry service handles the management of our container image delivery.
- A **Management Portal**. This is a web-based application and dashboard that you will use to install and manage suites, manage the container images required to bootstrap the suite, and manage the underlying infrastructure.

Overview of ITSM Automation (ITSMA)

The IT Service Management Automation (ITSMA) suite is a comprehensive and fully integrated IT Service Management software suite that enables IT to improve service levels, balance resources, control costs, and mitigate risk exposure to the organization. It enables you to manage IT services using a "lifecycle" approach, with consistent improvement built into the governance model.

As the Next-Generation (NG) Express edition of the ITSMA suite, ITSMA NG Express offers the following benefits:

- Quick and easy container-based deployment, scaling, and management
- Seamless integration and unified user experience
- Improved end user and Service Desk efficiency, as well as enhanced problem identification by leveraging big data based Smart Analytics
- Quick time to value through ITIL aligned out-of-box best practices, codeless configuration, and KPIs
- Robust IT management processes with Universal CMDB (UCMDB) supported workflows

Capabilities

ITSMA NG Express provides comprehensive capabilities described in the following table.

Capability	Description	Base product version
ITSMA Service Management	Service Management provides a tool for service desk and help desk management that quickly, efficiently handles change and incident management while integrating a broad range of ITSM capabilities, Big Data, and social collaboration to enable your workforce with connected intelligence.	Service Manager Web Tier 9.52

ITSMA Smart Analytics	Built on HPE Service Manager (SM) and using an OEM-licensed version of HPE IDOL, Smart Analytics provides a big data based Service Desk solution. This powerful SM-IDOL integration drives automation further into ITSM processes by mining unstructured data and extracting information from different types of data. Smart Analytics can help you achieve the following goals: • Improve the processes of Help Desk management • Reduce the time and effort expended on request submittals by end users and IT professionals • Accelerate the process of problem management • Improve search experience across a variety of internal and external content Smart Analytics provides the following features: Smart Ticket, Hot Topic Analytics, and Smart Search. **Smart Ticket** With Smart Ticket, you can quickly submit a Service Desk record by just entering a description or attaching a picture. Smart Analytics intelligently populates other fields, such as category or affected services, by extracting and analyzing the content that you entered in the record. **Hot Topic Analytics** Hot Topic Analytics intelligently displays an interactive diagram that indicates the hot topics among recent records in the system. Hot Topic Analytics enables you to easily discover common issues, identify escalation candidates, and create new records for escalation based on the selected candidates. **Smart Search** Smart Search enables you to search across a variety of content, including Service Manager records (such as Incidents and Changes), SharePoint documents, static web pages, and KM documents. You can integrate multiple knowledge libraries by configuring different search connectors, so that Service Manager users can search all the information that they can access.	Service Manager Smart Analytics 9.52
ITSMA Service Portal	Built on the modern technology of HPE Propel, Service Portal enables IT departments to offer their services in an online shopping experience, similar to what users experience today at popular online retailers. With Service Portal, end users can order, track, and manage their IT services, access knowledge articles, initiate chat conversations, and complete surveys.	Service Manager Service Portal 9.52
ITSMA Mobility	Connects your organization's people and information by providing your team access to the Change Management, Incident Management, and Service Desk applications provided by the Service Management capability through the use of smartphones.	Service Manager Mobility Client 9.52
ITSMA CMDB	Provides a central repository for the configuration information that is gathered from the Data Flow Management (DFM) process and various third-party applications and tools. It enables you to manage configuration items (CIs) in your organization.	CMS 10.32

ITSMA CMDB Browser	A lightweight client to access CMDB data, which provides a simple and intuitive search for Configuration Items (CIs) in CMDB and displays important data in the context of the selected CI. It is an ideal tool for providing quick access to specific CI information. For each CI, relevant data is presented and gathered into information widgets (for example, Properties, Environment, and Impact Simulation widgets). Data is presented by default in a Preview mode, with the option to view more comprehensive data in an Expanded mode.	CMS 10.32
ITSMA Suite Configuration	Provides a user interface for the suite administrator to configure and administer the suite, such as configuring an external LDAP server and configuring debugging settings.	ITSMA 2017.04

This release of ITSMA NG Express also uses the following shared HPE components:

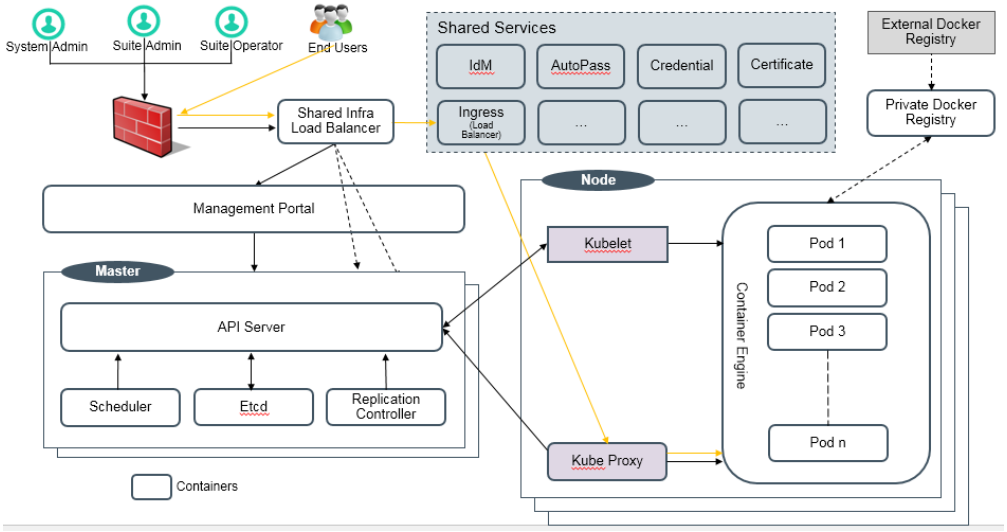
- HPE AutoPass 9.40
- HPE Identity Manager (IdM) 1.15.2

Containerization

ITSMA NG Express must be deployed and running on HPE ITOM Container Deployment Foundation (CDF), which is based on Docker containers and Kubernetes (K8S) orchestration. ITSMA NG Express comprises a bunch of microservices running in containers. For more information, see Overview of ITOM Container Deployment Foundation.

Personas

The following figure shows the personas of ITOM Container Deployment Foundation (CDF) and ITSMA NG Express.



ITOM CDF personas

ITOM CDF has two types of personas.

- IT Administrator
  Manages ITOM and all suite products, as well as the grow/shrink functions, and adding and removing working nodes (machines). The IT Administrator is a super administrator. This user has ability to request or add resources and has wide access permissions.

  > The out-of-box System Administrator user account is **admin**/ **cloud** (the initial password "cloud" must be changed at the initial login). After login as **admin**, you can add more users of the two types.

- Suite Administrator
  Manages a specific suite product. The Suite Administrator does not have access to the **ITOM SUITES** menu and **ADMIN** menu. This user can only access the RESOURCES menu and all the children menus under **RESOURCES**, including **Namespaces**, **Workloads**, **Service and Discovery**, **Persistent Volume Claims**, **Configuration**, and all the relevant subsidiaries.

ITSMA personas

The ITSMA suite has the following types of personas.

- ESS (Employee Self-Service) user:
  Has access to the Service Portal capability only.

> In a test system that uses the internal LDAP server, the following out-of-box user accounts are available: **falcon**, **kquan**, **sdapprover**, **Amy.Lopez**, and **Aaron.Caffrey**, whose passwords are all **123456**.

- Non-ESS user:
  Has full access to the ITSMA suite, including the **Suite Configuration** capability. Non-ESS users include the suite administrator, IT agents, Service Management/CMDB administrators, and so on.

  > The suite administrator **sysadmin** is a seeded user stored in the ITSMA IdM database. Its initial password is specified during the suite installation and can be changed after the suite administrator log in to ITSMA.

Language support

ITSMA NG Express 2017.04 supports the following languages:

- English
- German
- French
- Spanish

> The ITSMA user interface is displayed according to the user's browser language; however, if the user's browser language is not supported, the user interface is displayed in English.

> Be aware that, if using Firefox or Win7 IE11, users must select <Language> instead of "<Language> (<Country>)" when setting their browser language.

Glossary

The following table describes the terminology used in this help center.

Annotation	A key/value pair that can hold larger (compared to a label), and possibly not human-readable, data, intended to store non-identifying auxiliary data, especially data manipulated by tools and system extensions. Efficient filtering by annotation values is not supported. See http://kubernetes.io/docs/user-guide/annotations/.
Cluster	A cluster is a set of physical or virtual machines and other infrastructure resources used by Kubernetes to run your applications. A single cluster should be able to satisfy the needs of multiple users or groups of users (henceforth a 'user community'). Each user community wants to be able to work in isolation from other communities. Each user community has its own: • resources (pods, services, replication controllers, etc.) • policies (who can or cannot perform actions in their community) • constraints (this community is allowed this much quota, etc.) A cluster operator may create a Namespace for each unique user community. Kubernetes coordinates a highly available cluster of computers that are connected to work as a single unit. The abstractions in Kubernetes allow you to deploy containerized applications to a cluster without tying them specifically to individual machines. To make use of this new model of deployment, applications need to be packaged in a way that decouples them from individual hosts: they need to be containerized. Containerized applications are more flexible and available than in past deployment models, where applications were installed directly onto specific machines as packages deeply integrated into the host. Kubernetes automates the distribution and scheduling of application containers across a cluster in a more efficient way. Kubernetes is an open-source platform and is production-ready. A Kubernetes cluster consists of two types of resources: • The Master coordinates the cluster • Nodes are the workers that run applications See http://kubernetes.io/docs/admin/cluster-management/.
Config Map	The ConfigMap API resource holds key-value pairs of configuration data that can be consumed in pods or used to store configuration data for system components such as controllers. ConfigMap is similar to Secrets, but designed to more conveniently support working with strings that do not contain sensitive information. See http://kubernetes.io/docs/user-guide/configmap/.
Configuration files	The configuration files are just standard pod definition in json or yaml format in specific directory.

Container	A container (Linux Container) at its core is an allocation, portioning, and assignment of host (compute) resources such as CPU Shares, Network I/O, Bandwidth, Block I/O, and Memory (RAM) so that kernel level constructs may jail-off, isolate or “contain” these protected resources so that specific running services (processes) and namespaces may solely utilize them without interfering with the rest of the system. These processes could be lightweight Linux hosts based on a Linux image, multiple web severs and applications, a single subsystem like a database backend, to a single process such as ‘echo “Hello”’ with little to no overhead. Commonly known as “operating system-level virtualization” or “OS Virtual Environments” containers differ from hypervisor level virtualization. The main difference is that the container model eliminates the hypervisor layer, redundant OS kernels, binaries, and libraries needed to typically run workloads in a VM. Containers are based on operating-system-level virtualization rather than hardware virtualization. These containers are isolated from each other and from the host: they have their own filesystems, they can’t see each others’ processes, and their computational resource usage can be bounded. They are easier to build than VMs, and because they are decoupled from the underlying infrastructure and from the host filesystem, they are portable across clouds and OS distributions. Because containers are small and fast, one application can be packed in each container image. This one-to-one application-to-image relationship unlocks the full benefits of containers. With containers, immutable container images can be created at build/release time rather than deployment time, since each application doesn’t need to be composed with the rest of the application stack, nor married to the production infrastructure environment. Generating container images at build/release time enables a consistent environment to be carried from development into production. Similarly, containers are vastly more transparent than VMs, which facilitates monitoring and management. This is especially true when the containers’ process lifecycles are managed by the infrastructure rather than hidden by a process supervisor inside the container. Finally, with a single application per container, managing the containers becomes tantamount to managing deployment of the application. See http://kubernetes.io/docs/whatisk8s/.
Deamon sets	A Daemon Set ensures that all (or some) nodes run a copy of a pod. As nodes are added to the cluster, pods are added to them. As nodes are removed from the cluster, those pods are garbage collected. Deleting a Daemon Set will clean up the pods it created. If you are running clustered Kubernetes and are using static pods to run a pod on every node, you should probably be using a DaemonSet! Static pods are managed directly by kubelet daemon on a specific node, without API server observing it. It does not have associated any replication controller, kubelet daemon itself watches it and restarts it when it crashes. There is no health check though. Static pods are always bound to one kubelet daemon and always run on the same node with it. Kubelet automatically creates so-called mirror pod on Kubernetes API server for each static pod, so the pods are visible there, but they cannot be controlled from the API server. See http://kubernetes.io/docs/admin/daemons/
etcd	etcd is a highly-available key value store which Kubernetes uses for persistent storage of all of its REST API objects.

Image	Each container in a pod has its own image. Currently, the only type of image supported is a Docker Image. You create your Docker image and push it to a registry before referring to it in a Kubernetes pod. The image property of a container supports the same syntax as the docker command does, including private registries and tags. The following docker images will be shared across ITOM suites: HPE Linux, Vertica, Ingress, Apache HTTPD, PorstgreSQL, ZooKeeper, Kafka, Redis, Nginx, and more. See http://kubernetes.io/docs/user-guide/images/ and https://docs.docker.com/engine/tutorials/dockerimages/.
ITOM	IT Operations Management
Ingress	An Ingress is a collection of rules that allow inbound connections to reach the cluster services. It can be configured to give services externally-reachable urls, load balance traffic, terminate SSL, offer name based virtual hosting etc. Users request ingress by POSTing the Ingress resource to the API server. An Ingress controller is responsible for fulfilling the Ingress, usually with a loadbalancer, though it may also configure your edge router or additional frontends to help handle the traffic in an HA manner. See http://kubernetes.io/docs/user-guide/ingress/.
job	A job creates one or more pods and ensures that a specified number of them successfully terminate. As pods successfully complete, the job tracks the successful completions. When a specified number of successful completions is reached, the job itself is complete. Deleting a Job will cleanup the pods it created. A simple case is to create one Job object in order to reliably run one Pod to completion. The Job object will start a new Pod if the first pod fails or is deleted (for example due to a node hardware failure or a node reboot). A Job can also be used to run multiple pods in parallel. Running an example Job
Kubernetes	Kubernetes is an open-source platform for automating deployment, scaling, and operations of application containers across clusters of hosts. See http://kubernetes.io/docs/.
Label	A label is a key/value pair that is attached to objects, such as pods, to convey a user-defined identifying attribute. Labels can be used to organize and to select subsets of resources. Labels can be attached to objects at creation time and subsequently added and modified at any time. Each object can have a set of key/value labels defined. Each Key must be unique for a given object. For example, you can use the labels when you install a suite so you can categorize nodes to serve certain functions. For example, so you can install a suite only on certain nodes. This helps you partition your cluster. See http://kubernetes.io/docs/user-guide/labels/.

Name	All objects in the Kubernetes REST API are unambiguously identified by a Name and a UID. Names are generally client-provided. Only one object of a given kind can have a given name at a time (i.e., they are spatially unique). By convention, the names of Kubernetes resources should be up to maximum length of 253 characters and consist of lower case alphanumeric characters, -, and ., but certain resources have more specific restrictions. See http://kubernetes.io/docs/user-guide/identifiers/.
Namespace	A Namespace is a mechanism to partition resources created by users into a logically named group. A namespace is like a prefix to the name of a resource. Namespaces help different projects, teams, or customers to share a cluster, such as by preventing name collisions between unrelated teams. There are two default namespaces: • **default.** The default namespace for objects with no other namespace • **kube-system.** The namespace for objects created by the Kubernetes system See http://kubernetes.io/docs/admin/namespaces/.
Node	A node is a physical or virtual machine running Kubernetes, onto which pods can be scheduled. A Kubernetes cluster consists of two types of resources: • The Master coordinates the cluster • Nodes are the workers that run applications A node is a VM or a physical computer that serves as a worker machine in a Kubernetes cluster. Each node has a Kubelet, which is an agent for managing the node and communicating with the Kubernetes master. The node should also have tools for handling container operations, such as Docker or rkt. A Kubernetes cluster that handles production traffic should have a minimum of three nodes. See http://kubernetes.io/docs/admin/node/.
NFS	Network File System
PetSet	A PetSet is a Controller that provides a unique identity to its Pods. It provides guarantees about the ordering of deployment and scaling.

Persistent Volumes	A PersistentVolume (PV) is a piece of networked storage in the cluster that has been provisioned by an administrator. It is a resource in the cluster just like a node is a cluster resource. PVs are volume plugins like Volumes, but have a lifecycle independent of any individual pod that uses the PV. This API object captures the details of the implementation of the storage, be that NFS, iSCSI, or a cloud-provider-specific storage system. A PersistentVolumeClaim (PVC) is a request for storage by a user. It is similar to a pod. Pods consume node resources and PVCs consume PV resources. Pods can request specific levels of resources (CPU and Memory). Claims can request specific size and access modes (e.g, can be mounted once read/write or many times read-only). See http://kubernetes.io/docs/user-guide/persistent-volumes/.
Pod	A pod is a co-located group of containers and volumes. They are scheduled onto the same host. Pods serve as units of scheduling, deployment, and horizontal scaling/replication. Pods share fate, and share some resources, such as storage volumes and IP addresses. See http://kubernetes.io/docs/user-guide/pods/.
Registry	The HPE Registry is a server-side application that stores and lets you distribute Docker images. In the production environment, we use the public Docker Registry to replace the HPE Registry
Replica Set	A Replica Set ensures that a specified number of pod “replicas” are running at any given time. However, a Deployment is a higher-level concept that manages Replica Sets and provides declarative updates to pods along with a lot of other useful features. Therefore, we recommend using Deployments instead of directly using Replica Sets, unless you require custom update orchestration or don’t require updates at all. See http://kubernetes.io/docs/user-guide/replicasets/.
Replication Controller	A replication controller ensures that a specified number of pod replicas are running at any one time. It both allows for easy scaling of replicated systems and handles re-creation of a pod when the machine it is on reboots or otherwise fails. See http://kubernetes.io/docs/user-guide/replication-controller/.
Secret	A secret stores sensitive data, such as authentication tokens, which can be made available to containers upon request. See http://kubernetes.io/docs/user-guide/secrets/.
Selector	A selector is an expression that matches labels in order to identify related resources, such as which pods are targeted by a load-balanced service. See http://kubernetes.io/docs/user-guide/labels/.

Service	A service defines a set of pods and a means by which to access them, such as single stable IP address and corresponding DNS name (such as a web service or API server) that directs and loads-balances traffic to the set of pods that it covers. See http://kubernetes.io/docs/user-guide/services/.
Vault	HashiCorp Vault is used to See https://www.vaultproject.io.
Volume	A volume is a directory, possibly with some data in it, which is accessible to a Container as part of its filesystem. Kubernetes volumes build upon Docker Volumes, adding provisioning of the volume directory and/or device. Docker also has a concept of volumes, though it is somewhat looser and less managed. In Docker, a volume is simply a directory on disk or in another container. Lifetimes are not managed and until very recently there were only local-disk-backed volumes. Docker now provides volume drivers, but the functionality is very limited for now (e.g. as of Docker 1.7 only one volume driver is allowed per container and there is no way to pass parameters to volumes). A Kubernetes volume, on the other hand, has an explicit lifetime - the same as the pod that encloses it. Consequently, a volume outlives any containers that run within the Pod, and data is preserved across Container restarts. Of course, when a Pod ceases to exist, the volume will cease to exist, too. Perhaps more importantly than this, Kubernetes supports many type of volumes, and a Pod can use any number of them simultaneously. See http://kubernetes.io/docs/user-guide/volumes/
Worker node	Worker nodes run the applications. See http://kubernetes.io/docs/admin/node/.

Send documentation feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on ITSMA documents (ITSMA 201704)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to ovdoc-ITSM@hpe.com.

We appreciate your feedback!