

CLOUD FOUNDRY PLATFORM AS A SERVICE ON VBLOCK SYSTEM

Enabled by Pivotal CF, VCE Vblock Systems, EMC Unified Infrastructure Manager, and VMware vCenter Operations Manager

- Deploy an application platform designed for cloud computing environments and delivered as a service
- Provide elastic and flexible IaaS to support scalability of PaaS environment
- Monitor performance, analyze root cause and impact, manage capacity

EMC Solutions

Abstract

This white paper explores the interoperability of technology components of PaaS and IaaS cloud platforms and provisioning solutions that enable service providers to deploy, manage, and scale cloud-based application infrastructure services. This document describes how to build a PaaS service with Cloud Foundry enabled by Pivotal CF on Vblock Systems while ensuring scalability and elasticity of the underlying cloud infrastructure including the storage, compute, and networking layers.



TECHNOLOGY
ALLIANCE

February 2014

Copyright © 2014 EMC Corporation. All Rights Reserved.

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

The information in this publication is provided as is. EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com.

EMC², EMC, and the EMC logo are registered trademarks or trademarks of EMC Corporation in the United States and other countries.

All other trademarks used herein are the property of their respective owners.

Part Number H12825

Table of contents

Executive summary	5
Business case	5
Go to market	5
Solution overview	5
Key components	6
Key benefits	7
Introduction	8
Purpose	8
Scope	8
Audience	8
Terminology	8
Solution architecture	10
PaaS: Cloud Foundry and Pivotal CF	12
Open PaaS	12
Modularity	12
Elastic Runtime	13
Application services	15
IaaS: Vblock System	17
Vblock System	17
Unified Infrastructure Manager	18
UIM provisioning	19
Provisioning Pivotal CF PaaS	22
Overview	22
Deploying the Pivotal CF Operations Manager appliance	22
Deploying Pivotal CF with Operations Manager	24
Configure vSphere infrastructure for Pivotal CF	25
Configure and add Elastic Runtime	26
Configure and add application services	28
Deploy Pivotal CF Instance	28
Cloud Provider Interface	30
Deploying applications to Pivotal CF	31
Preparing to deploy an application	31
Using services	32

Vblock System infrastructure elastic provisioning.....	34
Provisioning additional ESX hosts to managed service.....	34
Expanding storage pools with Unisphere	36
Increasing the capacity of provisioned block storage	37
Decommissioning a server or storage volume	39
Monitoring and capacity planning.....	43
UIM monitoring.....	43
VMware vCenter Operations Manager	46
Capacity forecasting.....	46
Conclusion	51
References.....	54
EMC	54
Pivotal	54
VCE	54
VMware.....	54

Executive summary

Business case

Platform as a service (PaaS) is a category of cloud computing services that provides an application development platform as a service. In the PaaS model, the cloud provider offers and manages programming languages, frameworks, libraries, services, and tools for the end user to create and deploy applications. The cloud provider also manages and controls the underlying cloud infrastructure, including network, servers, operating systems, and storage, while the end user has control over the deployed applications and possibly the configuration settings for the application hosting environment. Therefore, PaaS automates the configuration, deployment, and management of applications, allowing the end user to focus on core application development and innovation. In contrast to a provider of software as a service, a PaaS service provider does not manage the applications themselves.

Although starting from a lower level, PaaS is perhaps the most dynamic market within the cloud computing stack, with a large number of vendors targeting the space and new competitors entering constantly.

With a \$1B market size in 2013 and an annual growth rate of 46 percent, PaaS offers a great opportunity to service providers, especially those who already have experience in offering cloud services. The challenge is to pick the right kind of services and technologies from the available options.

Go to market

Service providers can approach the PaaS space from multiple perspectives: Software-as-a-service (SaaS) providers may want to open the platform underlying their application offering to developers to allow them to easily integrate into the SaaS environment. The solution described in this white paper can support such an offering.

Another approach is for service providers with existing infrastructure-as-a-service (IaaS) offerings to extend them to include platform services. For do this, they need to provide agile, elastic, seamless, on-demand IaaS provisioning to support fast growth and the demand for flexible scalability of the PaaS environment. This document suggests solutions to overcome challenges that IaaS service providers face when they enhance their offerings to include PaaS.

A third category is “pure play” PaaS providers who brand their own PaaS offerings but do not operate their own infrastructures, which means that they act more as cloud brokers for infrastructure providers. While Pivotal CF supports that model well, that scenario is not the focus of this paper, which instead explores a tighter IaaS/PaaS integration.

Solution overview

This document describes how to build a PaaS service with Pivotal CF on Vblock Systems while ensuring scalability and elasticity of the underlying cloud infrastructure, including the storage, compute, and networking layers.

Cloud computing enables service providers to seamlessly deliver infrastructure services to customers while reducing power consumption, saving space, maintaining reliability, and reducing the overall cost to serve. An IaaS architecture based on EMC and VCE technology helps IT service providers offer customized services that meet the fast-changing business needs of their end users.

PaaS is a cloud service offering that enables innovative, agile development and fast and efficient deployment of business applications. In this model, the cloud provider offers and manages programming languages, frameworks, libraries, services, and tools for the end user to create and deploy applications.

To support such comprehensive, dynamic, and fast-growing development environments, service providers must ensure that the underlying cloud compute infrastructure provides availability, scalability, flexibility, and performance to the PaaS platform. This document presents a solution to these challenges, describing how to build a scalable Pivotal CF PaaS environment with an interoperable and easy-to-deploy underlying IaaS using EMC solutions and the VCE Vblock System.

This document provides an architectural overview of Pivotal CF and describes the provisioning processes and solutions involved in the deployment of Pivotal CF instances. It focuses on the interaction with provisioning of the underlying infrastructure resources using EMC® Unified Infrastructure Manager (UIM). It provides an overview of monitoring and capacity planning to enable a service provider to gather information about the current service availability and to predict future resource requirements, which will demand scaling of the environment. The document addresses how to use UIM to seamlessly provide more resources to a Pivotal CF instance. UIM lets you provision additional VMware ESX hosts to VMware vCenter clusters and extend existing Virtual Machine File System (VMFS) stores without disruption after the addition of new physical disks to the disk pool.

Key components

Key solution components include:

- Pivotal CF—Pivotal CF is the leading enterprise PaaS, powered by Cloud Foundry. It delivers an always-available, turnkey experience for scaling and updating PaaS on the private cloud, allowing agile development teams to rapidly update and scale applications. This solution features:
 - Pivotal CF Elastic Runtime Service—A complete, scalable runtime environment, extensible to most modern frameworks or languages running on Linux. It provides deployed applications with built-in services and enables them to automatically bind to new data services through a service broker or to an existing user-provided service. This enables application architects and developer teams to reduce time-to-market by shortening the software delivery cycle.
 - Pivotal CF Operations Manager—The industry's first turnkey enterprise PaaS management platform with IaaS integration. It enables zero-downtime patching and updates to the platform without service interruption.
 - Pivotal One Services—Pivotal enterprise data and application services that are available for automatic application binding and service provisioning:
 - Pivotal HD Service—Hadoop environment
 - Pivotal AX Service—Analytics services
 - Pivotal RabbitMQ Service—Message broker for asynchronous messaging service between applications
 - Pivotal MySQL Service—Multitenant, single-instance MySQL databases

- EMC UIM
 - Discovers and provisions Vblock System components (compute, storage, and networks) as a single entity
 - Manages the infrastructure service lifecycle: activate/provision, change, monitor, deactivate/release, decommission
 - Connects to VMware vSphere/vCenter Servers, creates ESX clusters, and makes hosts and datastores available for virtual machine creation
 - Provides access to Vblock System components for configuration, change management, and compliance tracking
 -
 - Simplifies VPLEX deployment by automating provisioning of the underlying storage, creating the VPLEX virtual volumes and stretch clusters within VMware vCenter
 - Monitors availability of Vblock System components and forwards alerts/events to enterprise monitoring systems
 - Displays dynamic topology views of UIM services, what Vblock Systems they reside on, and what virtual machines/vApps are running on them
- VMware vCenter Operations Manager—Provides operations dashboards for viewing the health, risk, and efficiency of your infrastructure, performance management and capacity forecasting, and optimization capabilities.
- EMC Unisphere® unified storage management—Manages daily tasks across the VNX series and legacy storage systems with an easy-to-use wizard that provides an intuitive, context-based approach to configuring storage, creating replicas, monitoring the environment, and managing host connections.

Key benefits

The key benefits of the solution for service providers are:

- Quick deployment of PaaS services
- Integration of Pivotal CF PaaS with Vblock System IaaS
- Dynamic scaling of compute, storage, and networking resources for PaaS

The key benefits for application developers and deployers are:

- Platform offers scalability and high availability.
- Developers can focus on implementing business logic.
- Service providers manage application operations.
- Solution provides at a reduced cost an elastic, scalable infrastructure sized to meet demand.

Introduction

Purpose

The purpose of this white paper is to provide an overview of how to deploy and operate a Pivotal CF-based PaaS on a scalable and dynamic Vblock System-based IaaS, with the goal of helping service providers design their PaaS offerings. It explores the interoperability of the Pivotal CF PaaS platform with an underlying Vblock System and IaaS based on UIM, and VMware vCenter Operations Manager. This platform allows service providers to:

- Deploy, manage, and monitor PaaS-based services
- Ensure scalability, elasticity, and flexibility of PaaS-based services

Scope

This document provides an overview of Pivotal CF and explains how to deploy and operate a Pivotal CF instance on a Vblock System infrastructure. It also shows how to seamlessly provide more resources to a Pivotal CF instance by provisioning additional ESX hosts to vCenter clusters and by extending existing datastores without disruption after adding new physical disks to a disk pool. For detailed product installation information, refer to the relevant product documentation.

Audience

This white paper is for EMC employees, partners, and customers, including IT planners, virtualization architects, and administrators, and any others involved in evaluating, acquiring, managing, operating, or designing a PaaS infrastructure environment using EMC and VCE technologies.

The reader should be familiar with the concepts and operations related to virtualization technologies and their use in a cloud infrastructure.

Terminology

Table 1 lists terminology used in this paper.

Table 1. Terminology

Term	Definition
Pivotal CF Operations Manager	Pivotal CF management platform
Pivotal CF	Pivotal's enterprise platform-as-a-service powered by Cloud Foundry
Droplet Execution Agent (DEA)	A component of the Pivotal CF platform that manages the lifecycle of applications hosted on a Pivotal CF instance.
Infrastructure as a service (IaaS)	A complete IT infrastructure consumed as a service. Users or tenants access a portion of a consolidated pool of federated resources to create and use their own compute infrastructure as needed, when needed, and how needed.
Platform as a service (PaaS)	A compute environment accessed as needed by a service provider over a network. PaaS is used to develop and run software as an alternative to designing, building, and installing an in-house development and production environment.

Term	Definition
UIM	EMC Unified Infrastructure Manager
vCenter Operations Manager	VMware monitoring solution

Solution architecture

Figure 1 shows an overview of PaaS and IaaS deployment stages and the required interoperability between involved solution components. The infrastructure layer is the fundamental IaaS layer and consists of core infrastructure components and associated management and monitoring services. UIM integration with Cisco UCS Manager and underlying core infrastructure components provide a service-based IaaS environment.

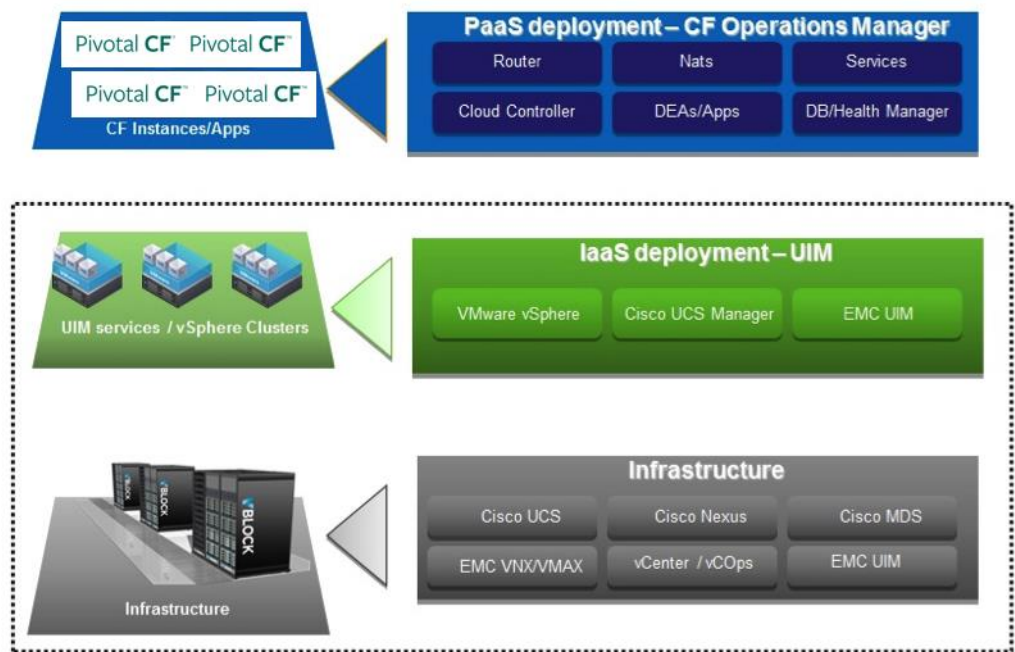


Figure 1. Solution layers

The foundation of the Pivotal CF PaaS environment is open-source technology Cloud Foundry, a modern application platform that supports the development, deployment, and operation of cloud-era applications.

Pivotal CF provides heterogeneous application services, supports multiple programming languages and frameworks, and automates the deployment of applications and their underlying back-end services across diverse cloud infrastructures.

One of Cloud Foundry's advantages is being agnostic about the underlying IaaS, which minimizes the potential risk of vendor lock-in. At the same time, this presents some challenges in terms of adequately monitoring and planning the capacity of the underlying IaaS platform to support the scaling and rapid expansion of applications and associated services.

The Vblock System, as the IaaS component of this solution, can support this kind of elastic, nondisruptive, on-demand scaling of compute, storage, and networking resources for the Pivotal CF platform.

More specifically, UIM provides simplified management for the Vblock Systems, accelerates infrastructure service delivery, and enhances flexibility and elasticity with functions such as:

- Selective ESX server provisioning/decommissioning
- Datastore addition and decommission
- Nondisruptive block-based LUN expansion

The sections of this document dedicated to the provisioning of Vblock System resources describe these functions in more detail.

In addition to provisioning and integration of PaaS services and the underlying IaaS, the environment must include implementation of adequate end-to-end performance and availability monitoring solutions. Predicting resource demands and available capacity is crucial in large-scale, fast-growing, and dynamic PaaS environments. For that purpose, advanced features of vCenter Operations Manager for capacity metering and trending, right-sizing and resource optimization, creating scenarios, and modeling are described in the sections of this document related to monitoring.

Along with capacity, availability is important to ensure defined service-level agreements (SLAs). In addition to the sections on monitoring, this white paper includes more details on UIM and functionalities such as root cause analysis, monitoring availability of Vblock System components as a single entity, and complete real-time topology views of physical, logical, and virtual abstractions and relationships.

PaaS: Cloud Foundry and Pivotal CF

Open PaaS

At the core of EMC's PaaS solutions is Cloud Foundry, the industry's open PaaS. Pivotal makes Cloud Foundry available as a turnkey enterprise PaaS called Pivotal CF.

Cloud Foundry creates a flexible and adaptable PaaS framework with the following characteristics:

- Multiple languages—Cloud Foundry supports many programming languages such as Ruby, Java, Scala, Node.js, Erlang, Python, and PHP. As more languages emerge, they are easily added using buildpacks, a flexible approach to assembling runtime environments. A wide ecosystem of industry players, including IBM and Heroku, provide buildpacks, ensuring support for virtually any language.
- Multiple frameworks—Cloud Foundry supports several popular frameworks including Rails, Sinatra, Spring, Grails, Express, and Lift. New frameworks are continuously being added.
- Multiple services—Application services, such as databases and message queues, are part of a real-world application deployment environment. Cloud Foundry allows the dynamic management of application services such as MySQL, Postgres, MongoDB, Redis, and RabbitMQ.
- Multiple clouds—Cloud Foundry is designed to run on any kind of underlying infrastructure cloud, which means it supports vSphere and vCloud as well as other environments like OpenStack and AWS.
- Public, private, hybrid:—A reservation that some have about PaaS is the potential lock-in to one particular service provider. Cloud Foundry addresses this concern by supporting multiple deployment options. Should application owners want to move an application in-house they can do so by deploying their own private Cloud Foundry environment. For development purposes, Cloud Foundry can be deployed on a laptop, from a memory stick.
- Open source—An open PaaS can only be truly open if its source can be accessed, modified, and augmented by the community and if the community can steer the project. Cloud Foundry provides that environment with contributions and participation from a broad network of technology and service providers including NTT, Verizon, Savvis, SAP, and IBM.

Modularity

To achieve the flexibility required by an open PaaS, Cloud Foundry is designed as a modular system with well-defined APIs and frameworks for extensions.

As Figure 2 shows, Cloud Foundry is designed to be extended in three dimensions:

- Runtime environments and frameworks
- Application services
- Cloud infrastructures

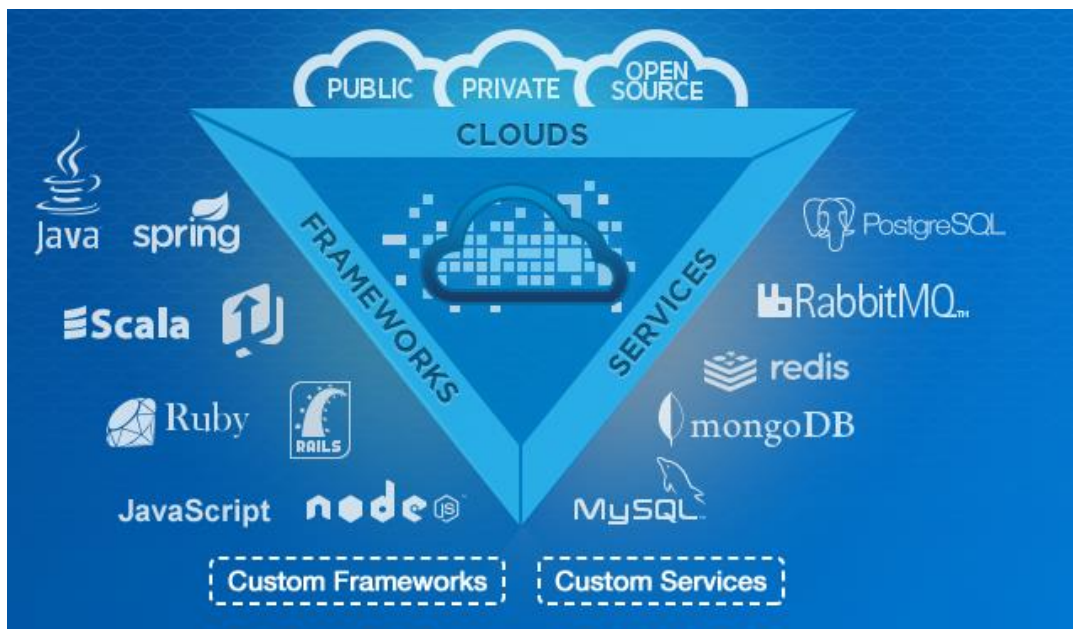


Figure 2. Cloud Foundry extensibility

For a service provider, these three differentiable service areas are specified in PaaS service definitions. Providing additional runtimes and services can attract and retain different user groups. This needs to be balanced with the effort and cost of keeping a larger number of environments continuously up to date for the users.

Because the underlying cloud infrastructure is invisible to the PaaS users, it is not a service differentiator from the users' point of view. However, using synergies in the integration with the IaaS infrastructure can lead to improved commercial offers to customers and increased margins. The level of abstraction that the Cloud Foundry cloud provider interface provides also opens the opportunity for service providers to step into a cloud broker role. They do not need to run all the IaaS capabilities themselves; they could also use external offerings. This can provide technical benefits in terms of scaling out but also allows for different service plans. For example, a service provider could offer a higher-value plan served directly from the service provider's infrastructure and a more commodity-type plan that could be delivered from an external vendor.

Elastic Runtime

All PaaS platforms share a common architecture pattern: a routing function at the front end, a grid of runtimes at the core, and back-end services. Figure 3 shows the main components of the core of Pivotal CF, the Elastic Runtime capability.

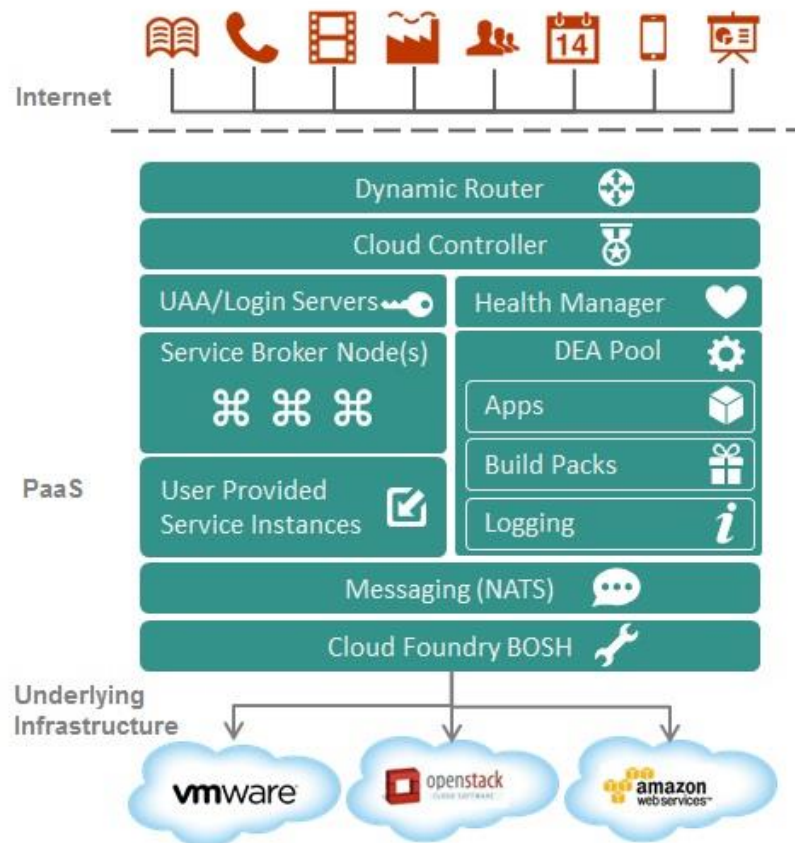


Figure 3. Pivotal CF Elastic Runtime logical architecture

The routing function provides the front end to the PaaS users, receives and analyzes requests, and determines where in the architecture to route the requests.

Application developers or deployers issue application lifecycle requests such as deploying an application from the VMC command-line tool or from the Eclipse STS plug-in. They upload their application source files using the CF CLI or plug-ins for popular IDE and build tools like Eclipse, Maven, Gradle, Jenkins, and Bamboo. The requests are first routed to the Cloud Controller instances. The Cloud Controller configures and manages the Pivotal CF environment and the applications and services running in it.

Deployed applications receive built-in services for horizontal scaling, load balancing, DNS, automated health management, and logging, resulting in a dramatic reduction in the number of vendors and integrations required for continuous software delivery.

Pivotal CF uses a flexible approach called buildpacks to dynamically assemble and configure a complete runtime environment for executing a particular class of applications. Rather than specifying how to run applications, developers can rely on buildpacks to detect, download, and configure the appropriate runtimes, containers, and libraries.

Because buildpacks are extensible to most modern runtimes and frameworks running on Linux, enterprises can deploy applications written in nearly any language to Pivotal CF. Pivotal's buildpacks for Java, Ruby, and Node are part of a broad buildpack

provider ecosystem that ensures constant updates and maintenance for virtually any language.

Pivotal CF routes end-user requests to instances of applications deployed out of a pool of execution environments called Droplet Execution Agents (DEAs), exemplifying elastic runtimes. Applications deployed on Cloud Foundry in turn use the services (such as databases) that are provided by the platform and instantiated by the application developers or deployers.

The Cloud Foundry components communicate with each other through a high-speed messaging infrastructure and are monitored by a Health Manager.

Application services

Middleware capabilities, such as databases or message queues, are provided to applications as services in Cloud Foundry. Service providers can differentiate their Cloud Foundry offerings by which services they make available to customers. When deploying a Cloud Foundry instance, the service provider administrator must register these application services with Cloud Foundry.

The Pivotal One platform provides a number of prepackaged services. These include:

- **Pivotal HD Service**—Builds, manages, and scales Hadoop as a natively integrated Pivotal CF service. Via the Service Broker, applications can bind to this service automatically assigning capacity in the Hadoop Distributed File System (HDFS), a database in HAWQ, and a resource queue in YARN. This reduces development cycle time by eliminating common Hadoop development complexities around deployment, security, networking, and resource management.
- **Pivotal AX Service**—Offers a self-service analytics environment to teams responsible for the creation, collection, storage, querying, and visualization of data. Lowers the barriers of entry to analyze Pivotal CF deployed applications and relevant supporting data to discover and communicate meaningful patterns that affect profitability and future product direction. Pivotal AX Service is analytics software that is built on Pivotal HD. It deploys and scales as a Cloud Foundry service.
- **Pivotal RabbitMQ Service**—Increases application speed, scalability, and reliability by delivering asynchronous messaging to applications. A message broker for applications running on Pivotal CF, RabbitMQ Service applications can integrate with other Pivotal CF applications and with applications outside Pivotal CF using the service broker.
- **Pivotal MySQL Service**—Provisions multitenant, single-instance MySQL databases suitable for rapid application development and testing.

To allow applications access to services, the application first has to be bound to the service. At runtime, when the application runs on Cloud Foundry, the environment contains a `VCAP_SERVICES` variable that has information about all the services bound to the application. The content of this variable is available in a Java Script Object Notation (JSON) document.

The JSON document is a list of service types with a list of the provisioned services bound to the application. Usually, there is just one instance of a service type, but you can create and bind multiple instances of a service type. The JSON document lists

each service with a number of parameters, including a name and label, as well as a credentials object that contains all of the information your application requires to access the service. Figure 4 shows an example of a JSON document.

```
VCAP_SERVICES=
{
  cleardb-n/a: [
    {
      name: "cleardb-1",
      label: "cleardb-n/a",
      plan: "spark",
      credentials: {
        name: "ad_c6f4446532610ab",
        hostname: "us-cdbr-east-03.cleardb.com",
        port: "3306",
        username: "b5d435f40dd2b2",
        password: "ebfc00ac",
        uri: "mysql://b5d435f40dd2b2:ebfc00ac@us-cdbr-east-03.cleardb.com:3306/ad_c6f4446532610ab",
        jdbcUrl: "jdbc:mysql://b5d435f40dd2b2:ebfc00ac@us-cdbr-east-03.cleardb.com:3306/ad_c6f4446532610ab"
      }
    }
  ],
  cloudamqp-n/a: [
    {
      name: "cloudamqp-6",
      label: "cloudamqp-n/a",
      plan: "lemur",
      credentials: {
        uri: "amqp://ksvyjmiv:IvN6dCdZmeQD400ZPKpulY0aLxIhe8wo@lemur.cloudamqp.com/ksvyjmiv"
      }
    },
    {
      name: "cloudamqp-dev-9dbc6",
      label: "cloudamqp-dev-n/a",
      plan: "lemur",
      credentials: {
        uri: "amqp://vhuklnxa:9lNFxpTuJsAdTts98vQIdKHw3MoJyMyV@lemur.cloudamqp.com/vhuklnxa"
      }
    }
  ],
  rediscloud-n/a: [
    {
      name: "rediscloud-1",
      label: "rediscloud-n/a",
      plan: "20mb",
      credentials: {
        port: "17546",
        host: "pub-redis-17546.MatanCluster.ec2.garantiadata.com",
        password: "1M5zd3QfVi9nUyya"
      }
    }
  ]
}
```

Figure 4. Example of a JSON document

Regardless of which programming language or framework you use, you must get the VCAP_SERVICES variable from the environment, parse it, and extract the connection information and credentials for the service you want to access. Use this data to connect to the service via the library, module, or driver.

IaaS: Vblock System

Vblock System

The IaaS cloud service model represents an on-demand provisioned, service-based, multitenant cloud infrastructure including network, servers, operating systems, and storage. IaaS represents the foundation on which service providers build their additional service offerings. The level to which IaaS supports rapid elastic provisioning and scalability defines the scope of functionalities and the quality of services built on top of it, as well as their SLA definitions.

Vblock System is a prepackaged system, assembled and configured for fast deployment of the vSphere virtualization platform with associated compute, storage, and network resources. Using the UIM solution with the Vblock Systems provides the fastest path to infrastructure deployment, significantly streamlining ongoing provisioning operations and enabling maximum flexibility for future enhancements and scale.

Figure 5 shows the architectural overview and topology of all the hardware components. It comprises the foundation of an IaaS platform where all the components are configured with high availability and redundancy to satisfy availability SLAs. Vblock System is the platform that enables the seamless extension of this environment. A modular architecture facilitates scaling: You can add compute capacity to the existing Vblock System (scale up), or you can add more Vblock Systems (scale out) into the environment. The UIM solution automates the manual steps needed to provision infrastructure components.

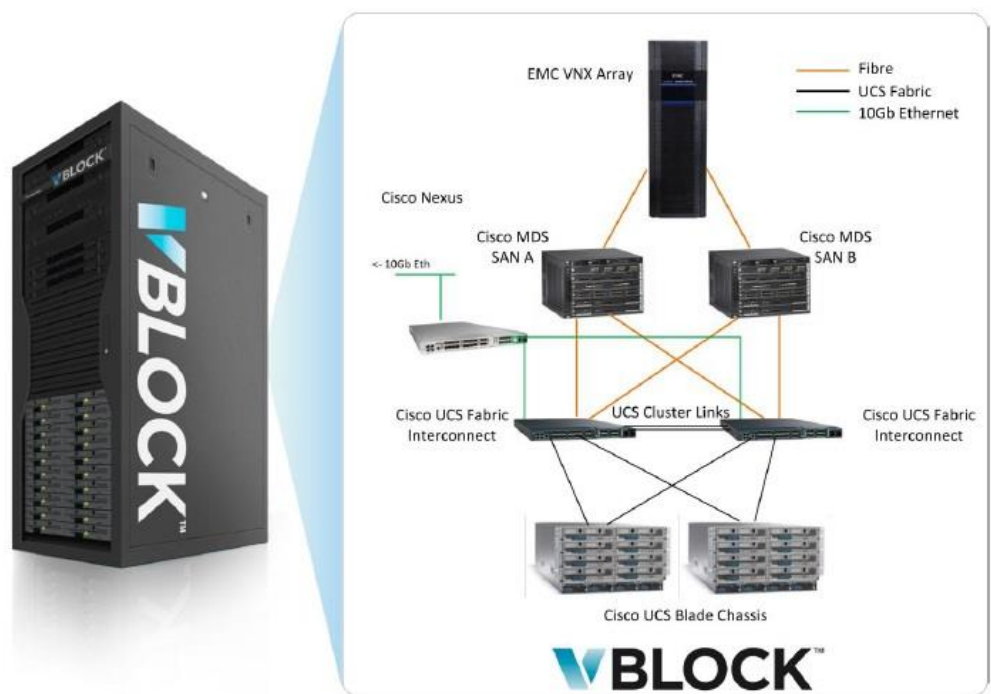


Figure 5. Physical overview of Vblock System

Unified Infrastructure Manager

The EMC UIM is a provisioning solution built for Vblock Systems. It automates infrastructure deployment and bare-metal provisioning, performing nonintrusive elastic provisioning to add or release compute resources within a given infrastructure service. Figure 6 shows in detail the components of the UIM solution and the correlation between them.

The UIM administrator configures and provisions network access, storage, and compute (blade) profiles. The combination of the three is managed as a single entity and presents a service offering template. When they are combined in a service template, resources can be reserved, deployed, allocated, and managed as a single entity through the UIM interface. vSphere hosts with associated storage and networking are assigned to the resource cluster in VMware vSphere through service provisioning.

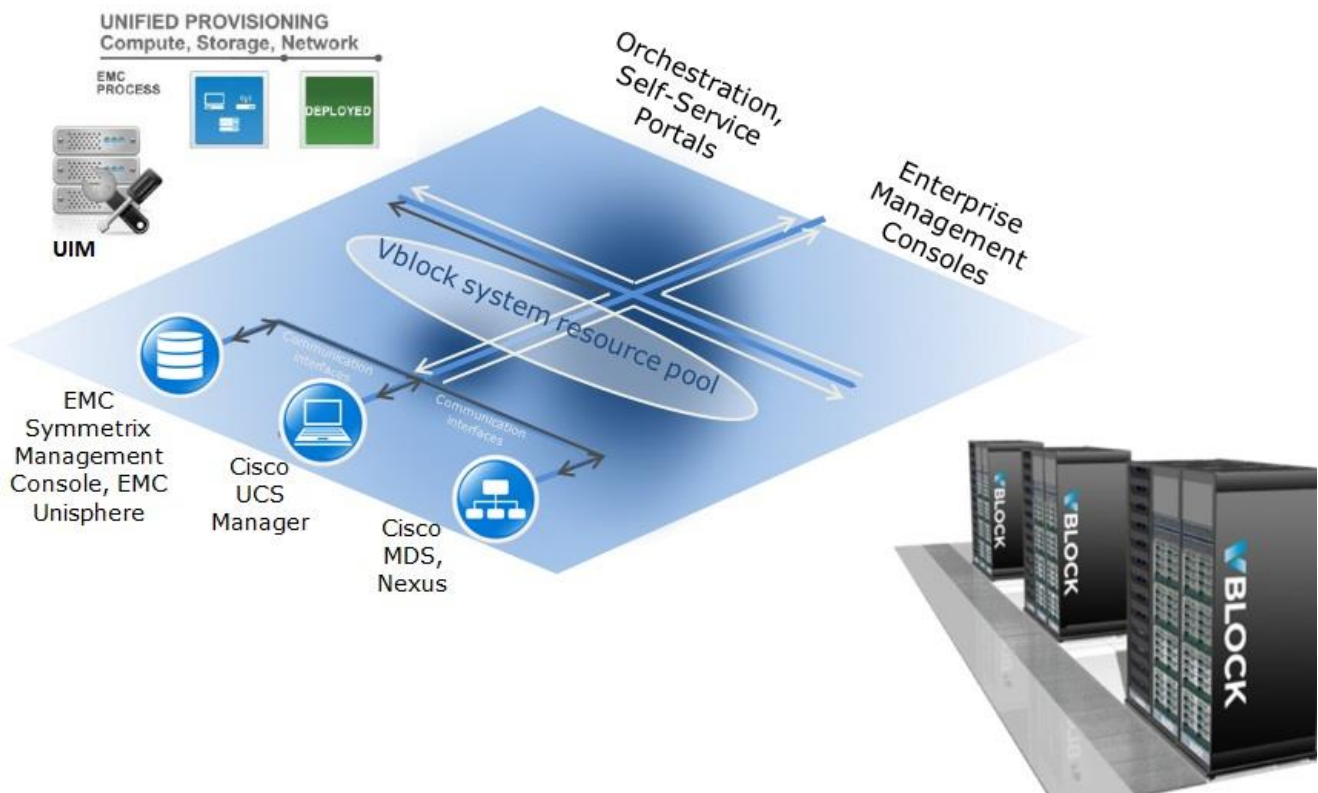


Figure 6. UIM solution components

UIM enables monitoring of your VblockSystem. With UIM you can:

- Monitor the availability of components within an individual Vblock System and across multiple Vblock Systems
- Create dynamic topology views of UIM services and the Vblock Systems on which they reside
- Determine root cause for faults in the Vblock System
- Provide impact analysis on higher-level abstractions

UIM provisioning

The UIM solution allows you to create infrastructure service offerings or infrastructure “templates” needed to build IT services and form the basis for delivering IaaS. After you define your service offerings you can automate infrastructure deployment and bare-metal provisioning, and perform nonintrusive elastic provisioning to add or release components within a given infrastructure service, as shown in Figure 7.



Figure 7. UIM service lifecycle

UIM supports the following functions that enable elastic service provisioning:

- Discovery and provisioning of Vblock System components (compute, storage, and networks) as a single entity
- Creation of infrastructure services to automate provisioning, building, and management of provisioned resources
- Administration and management of Vblock System resources and resource allocation
- Connectivity to VMware vSphere/vCenter Servers, synchronization, and integration of provisioned ESX hosts and datastores with a vCenter Server
- Modification of the infrastructure services: add or remove compute, storage, or networking components
- Enablement of access to Vblock System components for configuration, change management, and compliance tracking

UIM services are created from service offerings. Before a service offering can be created, the Vblock System must be discovered and the Vblock System resources must be configured for UIM automation. Vblock System discovery, resource configuration, and creation of service offerings are performed from the UIM Center user interface.

The UIM Service Catalog allows you to define and present a consistent set of infrastructure services that can be delivered by the Vblock System. These services define a compute profile (number of compute blades, type of blade), storage profile (size of boot partition, size and number of datastores, service grades for storage), network profile (VLAN ID, QoS, PIN Group), and operating system, as shown in Figure 8.

Servers					
	Number of Servers	Grade	Description	Minimum Blades	Maximum Blades
	1	Silver		1	4
Storage Devices	Type	Grade	Description	Size GB	
	Block Data	Vblock ESX HA		10	
	Block Data	500 GB pool 1		500	
Storage Limits	Grade	Description	1 ▲	Minimum Size GB	Maximum Size GB
	500 GB pool 1			500	1000
	Vblock ESX Boot			20	1000
Networks	NIC	QoS	Pin Group	VLANs	
	1			vblock_esx_nfsnas,vblock_esx_vmotion,vblock_esx_mgmt	
	2			vblock_esx_mgmt,vblock_esx_nfsnas,vblock_esx_vmotion	

Figure 8. UIM service offering

The service offering is a template that can be used multiple times to provision different infrastructure services. When provisioning is executed, interaction with the Vblock System devices occurs. An extensive amount of provisioning operations is performed, including creation of the service profile for the UCS blade servers, the LUN creation, zoning, and vNIC and vHBA configuration. After provisioning has been completed, all configurations are validated to ensure they match the expected values.

With hardware components provisioned, UIM continues to deploy and configure the selected version of the vSphere hypervisor to provisioned blade servers.

The service provider administrator performs a manual deployment and initial configuration of vCenter Server, networking (through VMware vSphere Distributed Switch [VDS], for example), and vSphere datacenter/cluster infrastructure components in the vSphere environment.

UIM integrates with the VMware vCenter management server. To establish connection between components, enter the IP address of the respective vCenter servers and administrative credentials in UIM service properties. UIM establishes a relationship with a defined vCenter server and performs configuration and synchronization of provisioned vSphere hosts in clusters in the vSphere datacenter.

When the UIM service provisioning process is complete, service activation makes the provisioned vSphere hosts and associated infrastructure available for use.

When implemented on top of the core infrastructure layer, the UIM solution features an elastic and agile IaaS capable of hosting dynamic PaaS instances and providing vSphere resources as needed. If additional compute or storage resources are required for the PaaS environment, you can easily provision new or expand existing

IaaS resources by creating new or modifying existing service properties using the UIMGUI.

Along with the need for elastic provisioning to accommodate the growth and dynamic change of PaaS environments, infrastructure performance and capacity monitoring are crucial. The process of deploying new Pivotal CF instances or upgrading existing ones rapidly provisions or removes a large number of virtual machines and services. Understanding the consequences to capacity and performance of the underlying IaaS is important; therefore, some of the UIM infrastructure monitoring features and vCenter Operations Management Suite capacity-forecasting features are detailed in [Monitoring and capacity planning](#).

Deployed on the provisioned vSphere infrastructure, Pivotal CF Operations Manager represents an orchestration and integration mechanism that provides turnkey deployment and updates of Pivotal CF components to create a fully functional platform. [Provisioning Pivotal CF PaaS](#) on [page 22](#) describes these activities in more detail.

Provisioning Pivotal CF PaaS

Overview

This section provides an overview of provisioning and deployment processes and solutions from the platform or service provider's perspective.

The workflow shown in Figure 9 represents the scope and flow of processes involved in the deployment of a Pivotal CF instance.

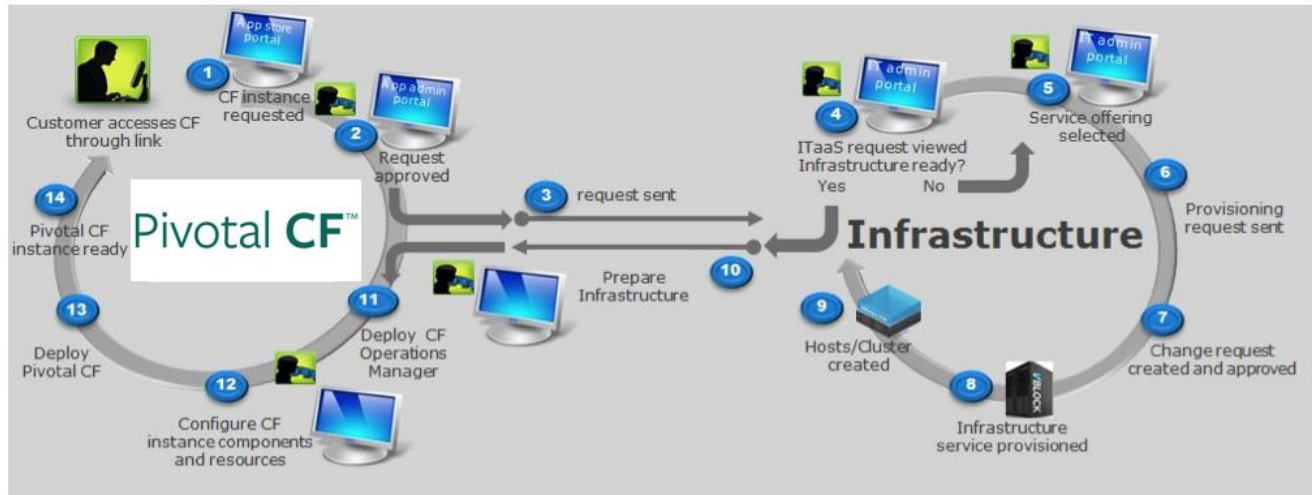


Figure 9. Processes involved in delivering PaaS and IaaS

Following the request initiation for a new Pivotal CF instance and approval processes on the service provider side, the service provider administrator provisions the appropriate UIM services. The provisioned resources are associated to vCenter clusters. The service provider administrator performs manual configurations and fulfills prerequisites before deploying the new Pivotal CF instance.

Next, the Pivotal CF Operations Manager is deployed. Pivotal CF Operations Manager is a virtual appliance that provides a simple GUI for release engineering, continuous deployment, monitoring, orchestration, and lifecycle management of large-scale distributed services. CF Operations Manager enables direct PaaS integration with the underlying IaaS and is used to deploy new Pivotal CF PaaS services and components, and to update existing ones. Pivotal CF contains a number of components designed to make the system horizontally scalable. This means that a Pivotal CF instance can have one or more copies of each component to meet the load needs.

Deploying the Pivotal CF Operations Manager appliance

From the vSphere Client you deploy the Pivotal CF Operations Manager OVF file to the vSphere datacenter/cluster that will provide vSphere infrastructure for the Pivotal CF instance, as shown in Figure 10.

Deploy OVF Template

Properties
Customize the software solution for this deployment.

[Source](#)
[OVF Template Details](#)
[End User License Agreement](#)
[Name and Location](#)
[Storage](#)
[Disk Format](#)
Properties
 Ready to Complete

IP Address
The IP address for the Pivotal CF Installer. Leave blank if DHCP is desired.

Netmask
The netmask for the Pivotal CF Installer's network. Leave blank if DHCP is desired.

Default Gateway
The default gateway address for the Pivotal CF Installer's network. Leave blank if DHCP is desired.

DNS
The domain name servers for the Pivotal CF Installer (comma separated). Leave blank if DHCP is desired.

NTP Servers
Comma-delimited list of NTP servers

Admin Password
This password is used to SSH into the Pivotal CF Installer. The username is 'tempest'.
 Enter password
 Confirm password

Figure 10. Deploying Pivotal CF Operations Manager appliance

After you deploy the CF Operations Manager appliance, you can deploy a production large-scale instance of the Pivotal CF platform. To configure and deploy Pivotal CF components, log in to the web interface using the IP address of the deployed CF Operations Manager, as shown in Figure 11.

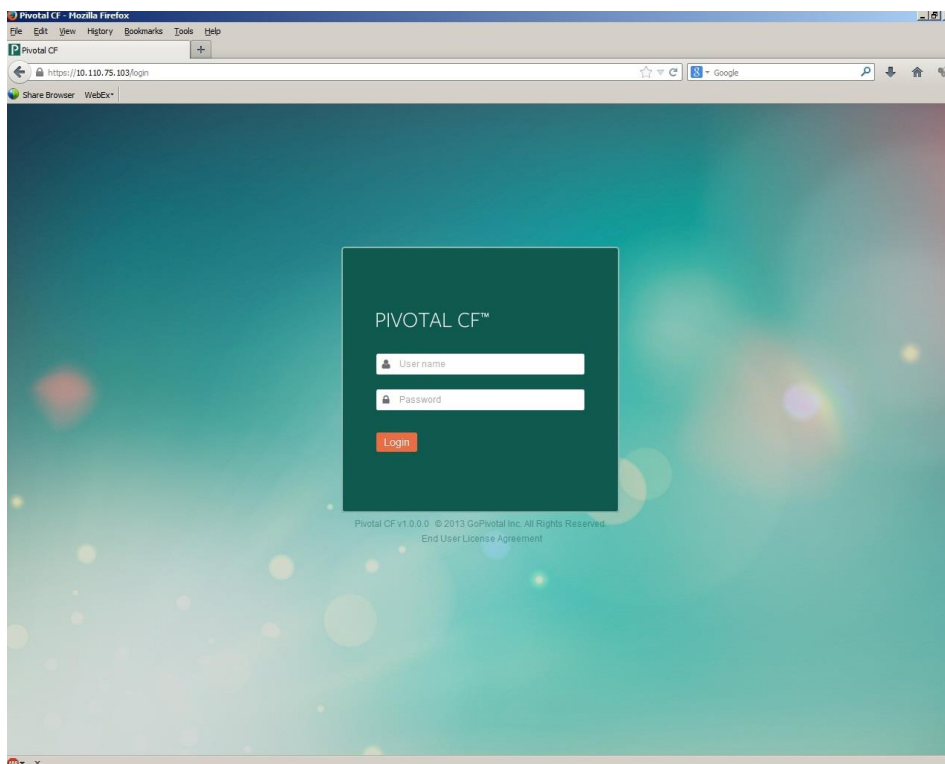


Figure 11. CF Operations Manager login page

Deploying Pivotal CF with Operations Manager

Depending on the service offering, a Pivotal CF instance is deployed when the service provider administrator chooses and configures platform components using the main Product Dashboard page, as shown in Figure 12.

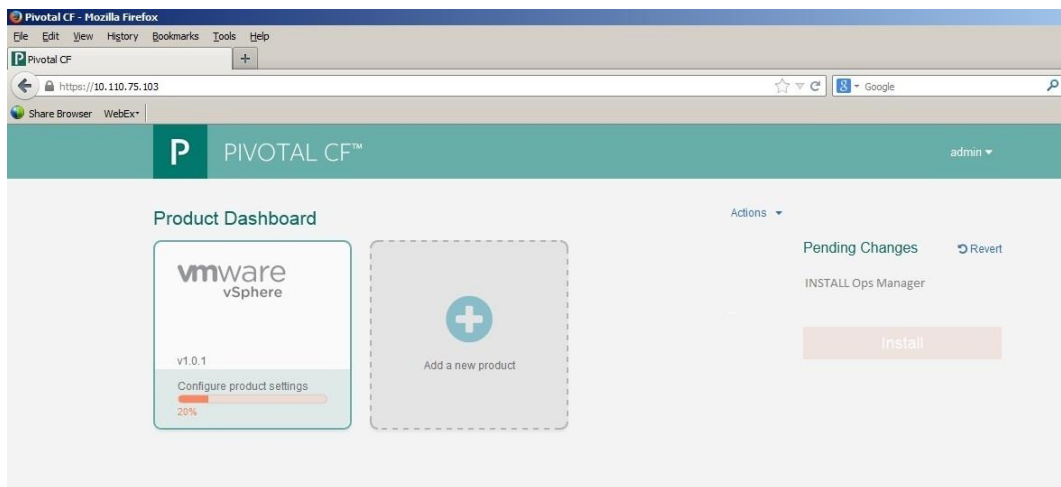


Figure 12. Pivotal CF Product Dashboard

The major steps to deploy Pivotal CF with Operations Manager are:

1. Configure vSphere infrastructure for Pivotal CF.
2. Configure and add Elastic Runtime.
3. Configure and add application services.

4. Deploy Pivotal CF instance.

The following sections describe each of these steps in detail.

Configure vSphere infrastructure for Pivotal CF

The Pivotal CF core infrastructure is provisioned by UIM and configured as a vSphere resource managed by vCenter Server. The first step in configuring the vSphere infrastructure for Pivotal CF is to configure core infrastructure components and resources, which will be used by other Pivotal CF platform services deployed by CF Operations Manager, as follows:

1. Specify provisioned vSphere infrastructure
2. Provide details of virtualized infrastructure as vCenter server credentials. Provide vSphere infrastructure related details such as name of network, datacenter, cluster and datastore, NTP and networking details for new Pivotal CF instance in **vSphere configuration** as shown in Figure 13.
3. Save changes.

The screenshot displays the 'vSphere configuration' page. On the left, a sidebar lists configuration sections: 'vCenter credentials', 'vSphere configuration' (highlighted), 'Network configuration', 'NTP servers', and 'Resource sizes'. Below the sidebar is a progress bar showing '100%'. The main content area contains several input fields, each with an information icon to its right: 'Network name *' with the value 'CF Network', 'Datacenter name *' with 'UIMTest VC', 'Cluster name *' with 'uim-vc test', 'Datastore name *' with 'uim-vc test_data2', and 'Resource pool name' which is empty. A blue 'Save' button is located at the bottom of the form.

Figure 13. Core infrastructure configuration page

Currently, interoperability of CF Operations Manager and the Vblock infrastructure networking layer is achieved using a VMware VDS. Refer to Pivotal CF documentation for details and updates on networking-related considerations. Depending on the platform service offering and scale of the Pivotal CF instance, you can increase CF Operations Manager resources if needed.

Configure and add Elastic Runtime

This section explains how to add an Elastic Runtime environment as shown in Figure 14 and configure the initial number of instances and resources of Pivotal CF components.

1. To get started, click the large **Elastic Runtime** button as shown in Figure 14.

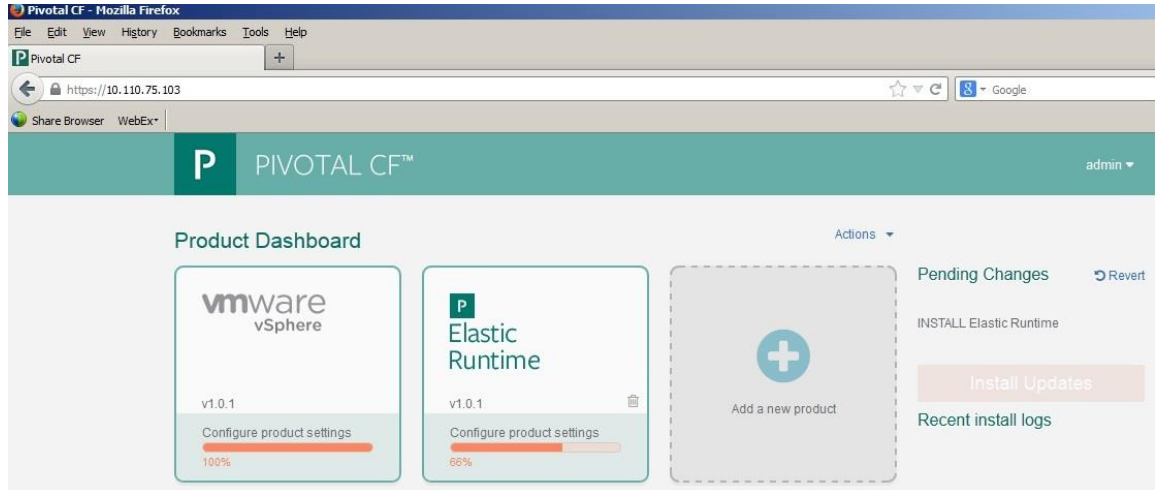


Figure 14. Add Elastic Runtime

2. On the Elastic Runtime configuration page (shown in Figure 15) you provide configuration details for core Pivotal CF components and their resources . Here you define the configuration of:
 - HAProxy –load balancer and SSL Certificates
 - Cloud Controller - manages application lifecycle management and orchestrates service provisioning and component binding operations
 - Router - manages external system traffic and application Internet/Intranet traffic
 - Web Console - management Web application for application developers and administrators
 - SAML login - Single sign-on configuration

Elastic Runtime

Settings

Status

Credentials

Logs

Upgrade

HAProxy

Cloud Controller

✓ Router

✓ SAML Login

✓ Web Console

✓ Resource sizes

66%

JOB	INSTANCES	CPU	RAM (MB)	EPHEMERAL DISK (MB)	PERSISTENT DISK (MB)
HAProxy	1	1	1024	2048	0
NATS	1	1	1024	2048	0
Health Manager	1	1	1024	2048	0
NFS Server	1	1	1024	2048	102400
Cloud Controller Database	1	1	1024	2048	2048
Cloud Controller	1	1	1024	10240	0
Router	1	1	1024	2048	0
Syslog	1	1	1024	2048	8192
Collector	1	1	1024	2048	0
UAA Database	1	1	1024	2048	8192
UAA	1	1	1024	2048	0
SAML Login	1	1	1024	2048	0
Console Database	1	1	1024	2048	1024
DEA	1	2	16384	32768	0
Loggregator Server	1	1	1024	2048	0
Loggregator Trafficcontroller	1	1	1024	2048	0
Compilation	5	2	1024	4096	0

Save

Figure 15. Configuring Elastic Runtime component instances and resources

3. Click Save.

Besides configuring these components, you can also extend an existing instance of Pivotal CF with additional resources or instances of certain components simply by changing values in appropriate fields and committing the environment update.

Subsequent sections in this white paper describe how to use capacity planning, “what if” scenarios, and UIM provisioning to support such extensibility on the underlying IaaS level.

Configure and add application services

As an example of how to provide an application service, the following steps describe how to configure and add the MySQL Dev relational database service with associated resources:

1. Click the large **MySQL Dev** button shown in Figure 16.

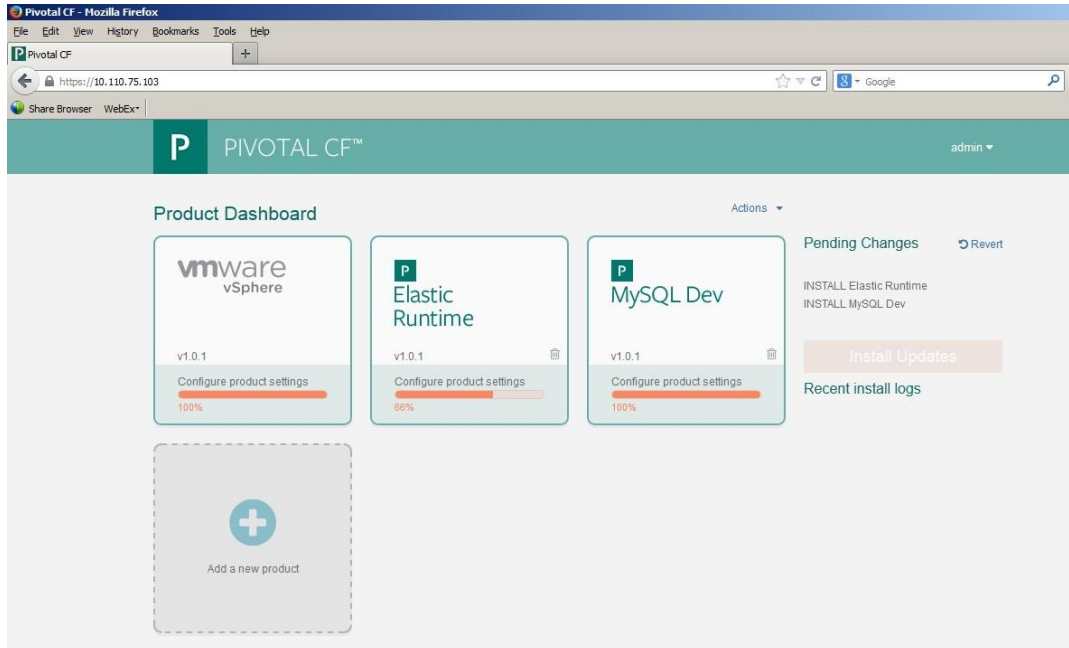


Figure 16. Add MySQL Dev component and associated resources

2. Configure the number of instances of MySQL Dev components appropriate for the given performance and scalability requirements, as shown in Figure 17.

The screenshot shows the 'Configure MySQL Dev' page. On the left, there's a sidebar with a 'Resource sizes' button and a '← Back to Install' link. The main area is titled 'Resource sizes' and contains a table with columns: JOB, INSTANCES, CPU, RAM (MB), EPHEMERAL DISK (MB), and PERSISTENT DISK (MB). The table has three rows: 'MySQL Gateway', 'MySQL Node', and 'Compilation'. Each row has input fields for the 'INSTANCES' column. Below the table is a red 'Save' button with a checkmark icon.

JOB	INSTANCES	CPU	RAM (MB)	EPHEMERAL DISK (MB)	PERSISTENT DISK (MB)
MySQL Gateway	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1024"/>	<input type="text" value="2048"/>	<input type="text" value="0"/>
MySQL Node	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="2048"/>	<input type="text" value="3072"/>	<input type="text" value="8192"/>
Compilation	<input type="text" value="1"/>	<input type="text" value="2"/>	<input type="text" value="1024"/>	<input type="text" value="4096"/>	<input type="text" value="0"/>

Figure 17. Configuring MySQL Dev resources

3. Click **Save**.

Deploy Pivotal CF Instance

Follow these steps to deploy the Pivotal CF instance:

1. After modifying and saving the configurations of all components, run the installation process from the Product Dashboard page shown in Figure 18. Deployment time varies depending on the scale of the Pivotal CF instance and

the performance of the underlying infrastructure. The deployment process does not require user intervention.

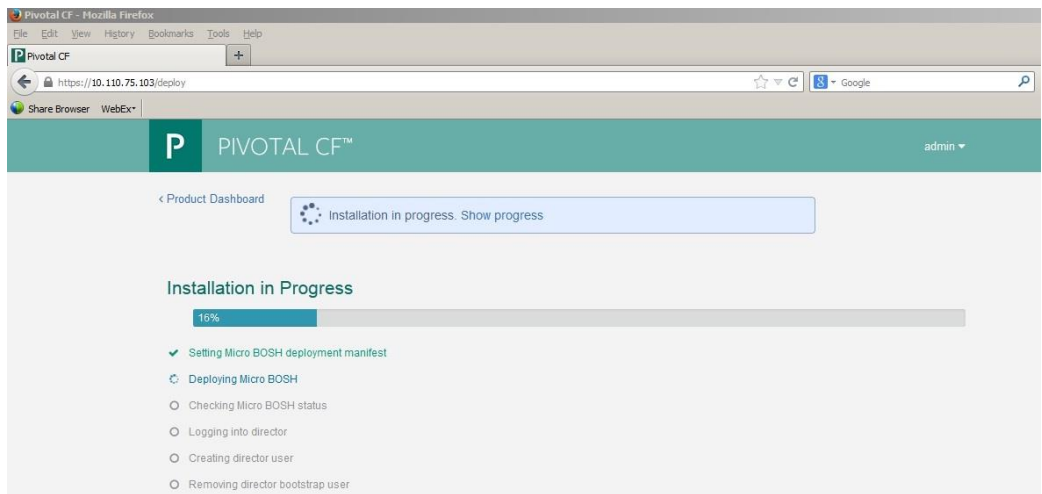


Figure 18. Deployment of Pivotal CF instance

After deploying the Pivotal CF instance, you can manage applications, application spaces, users, and organizations using a web console, as shown in Figure 19, accessible by browsing to <http://console.your-app-domain.com>.

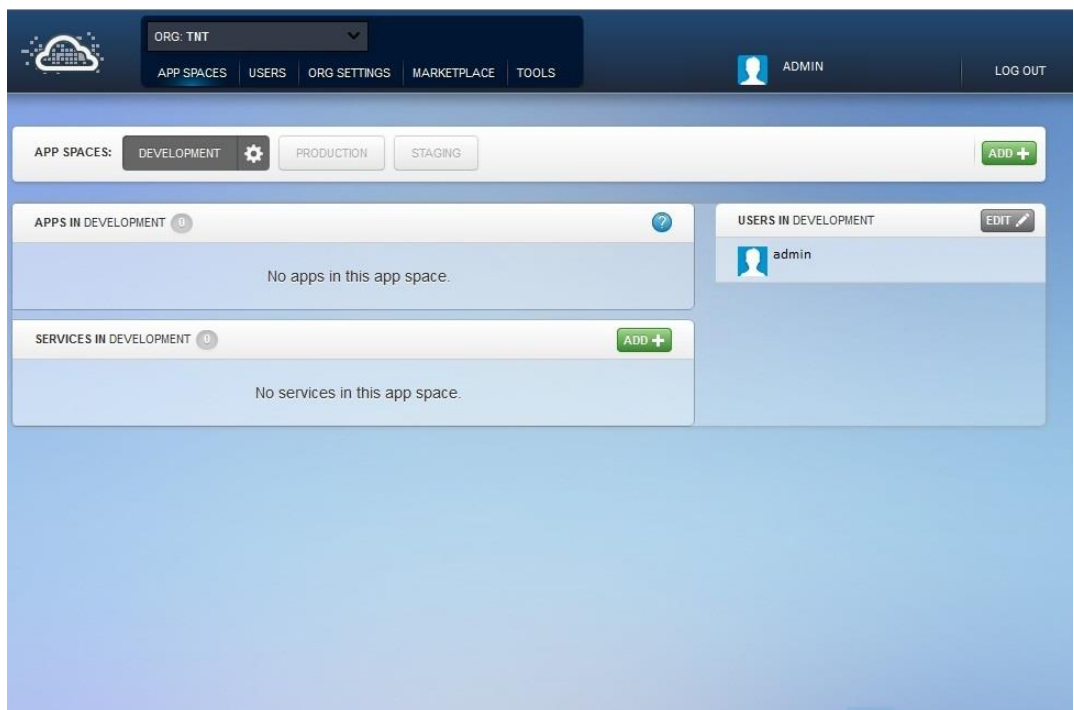


Figure 19. Pivotal CF console

For detailed deployment requirements and instructions, refer to:

<http://docs.gopivotal.com/pivotalcf/getstarted/index.html>

Cloud Provider Interface

The core Cloud Foundry engine is abstracted from any underlying IaaS through the Cloud Provider Interface (CPI) that defines functions such as instantiating a virtual machine in which to run an elastic runtime or service component. IaaS infrastructures that will support Cloud Foundry must implement this API and translate its commands to their native functions. This is the integration point between the Cloud Foundry resource management and the underlying infrastructure resource management. Currently, plug-ins are available for VMware vSphere, OpenStack, and Amazon Web Services. In this solution, the integration of the Vblock System infrastructure and UIM services with Cloud Foundry is performed through the vSphere plug-in.

Deploying applications to Pivotal CF

Preparing to deploy an application

If the application design follows a few basic guidelines (for example, not writing to the local file system and not relying on persisted HTTP sessions), applications written using application frameworks that are supported on a Pivotal CF platform can often run unmodified on Pivotal CF.

As long as an application uses the provided frameworks, all that is required for deployment preparation is to save it in the language-specific package format, such as a WAR file in Java or a GemFile for Ruby.

If the application needs components that are not provided by the Pivotal CF platform, third-party buildpacks can be used. These can be community-developed buildpacks or custom buildpacks.

Refer to the following web page for details about how to prepare to deploy applications:

<http://docs.gopivotal.com/pivotalcf/devguide/deploy-apps/prepare-to-deploy.html>

To deploy an application to Pivotal CF, use the CLI **cf push** command or equivalent plug-ins for development environments, such as Eclipse. So that deploying an application to Pivotal CF is not confused with deploying a Pivotal CF environment itself, deploying an application to Pivotal CF is often referred to as “pushing an application.”

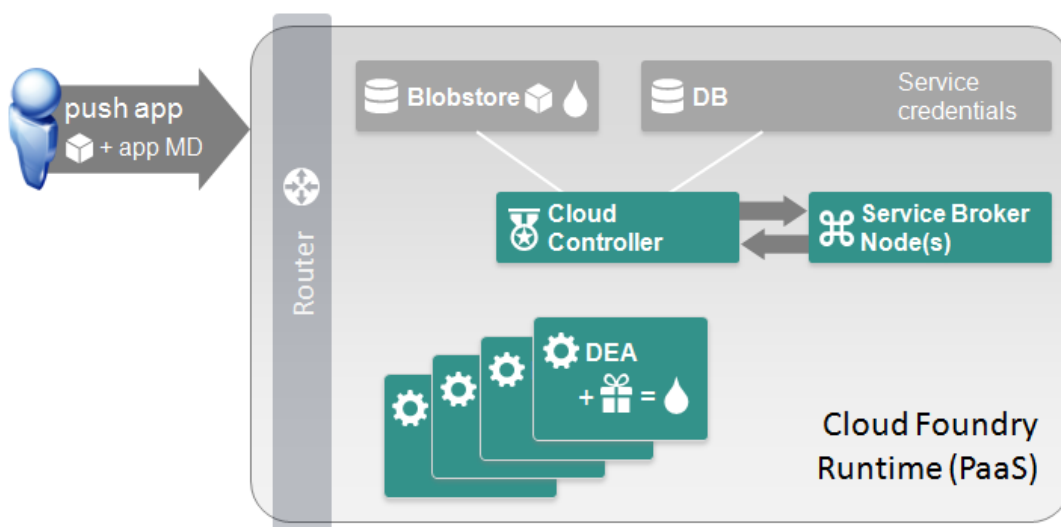


Figure 20. Deploying an application to Pivotal CF

When you push an application, as shown in Figure 20, Pivotal CF performs a number of staging tasks:

1. Uploading and storing application files
2. Examining and storing application metadata
3. Creating a droplet (a unit of execution)

4. Finding a container (DEA) in which to run the application
5. Provisioning the container with the appropriate software and system resources
6. Starting one or more instances of the application

Refer to the following web page for details about how to push applications:

<http://docs.gopivotal.com/pivotalcf/devguide/deploy-apps/getting-started.html>

Using services

An application running on a Pivotal CF instance can use services available on the platform. Cloud Foundry Services are any type of add-on that can be provisioned alongside your application; for example, a database or an account on a third-party SaaS provider. Services can be “managed services” that integrate into Cloud Foundry via a service broker that implements the Service Broker API. Services can also be “user-provided,” where service instances are preprovisioned outside Cloud Foundry.

First, the user creates a service instance out of the services registered with the Service Broker, as illustrated in Figure 21. The type of service determines what a service instance looks like: In the case of a database service, for example, a service instance could be a database in a multitenant server or in a dedicated environment.

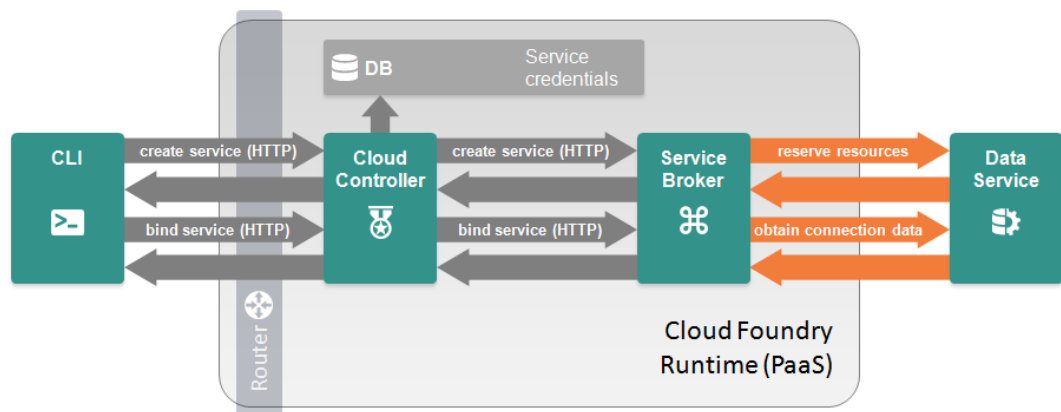


Figure 21. Creating and binding a service

The second step is to bind a service instance to an application. This step might not always be necessary, depending on the type of service. The binding operation puts credentials for the service instance in the environment variable `VCAP_SERVICES`, where the application can consume them. Once a service instance is created and bound to the application, the user must configure the application to use the correct credentials for the service. With those credentials, a connection to the actual service instance can be established.

Note: For Java applications, this can be accomplished by use of the spring-cloud library (<https://github.com/spring-projects/spring-cloud>).

Refer to the following web page for details about using services:

<http://docs.gopivotal.com/pivotalcf/devguide/services/>

A Pivotal CF instance can consume a large amount of compute and storage resources. Therefore, you need a solution that gives you the capability to support the growth of your PaaS environment. [Vblock System infrastructure elastic provisioning on page 34](#) describes how to leverage the UIM solution to add more compute and storage resources to a live, operating Cloud Foundry instance without disruption.

Vblock System infrastructure elastic provisioning

This section gives you an overview of the UIM functions that enable platform providers to support elasticity and nondisruptive scalability of IaaS for simultaneous, rapid provisioning and growth of PaaS platform environments.

Provisioning additional ESX hosts to managed service

If more CPU and memory resources for Pivotal CF services components or applications deployed on Pivotal CF are required, a service provider administrator must initiate the provisioning of additional UCS blade servers. With UIM you can easily add servers to provide more compute resources. The new resources can be provisioned as vSphere hosts, which during the UIM service provisioning process are synchronized with existing clusters and pools of resources in vCenter, which in turn manage your Pivotal CF platform environment.

To extend a previously provisioned UIM service, modify the number and grade of servers you want to add, and provide their FQDN and IP addresses, as shown in Figure 22.

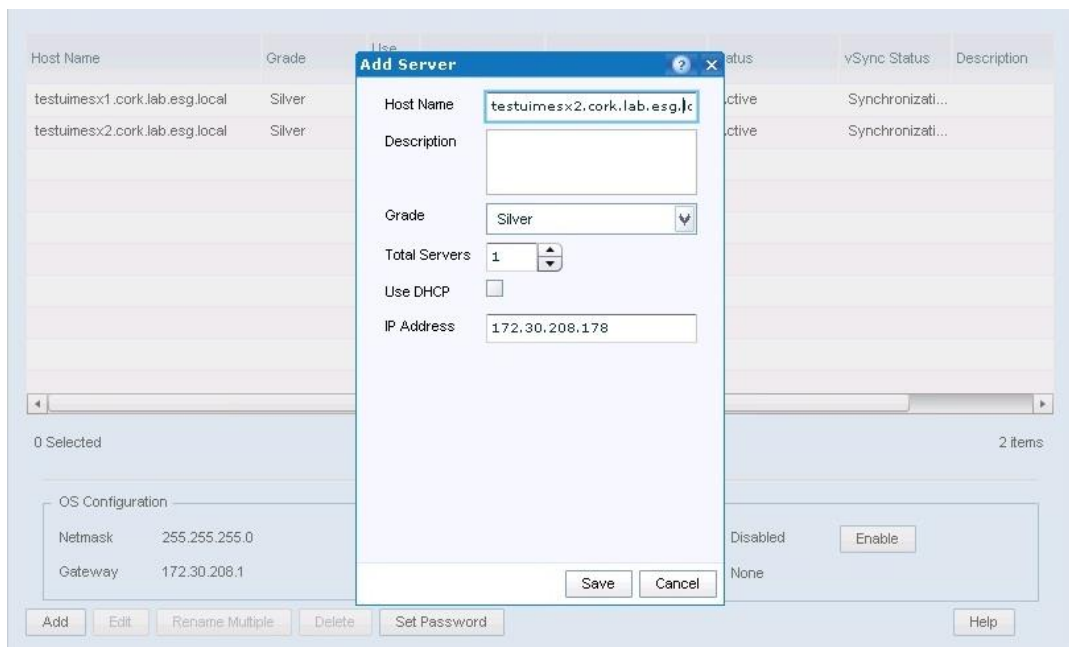


Figure 22. Adding new vSphere hosts to existing provisioned service

As part of the provisioning process, UIM performs the following tasks:

- Configure networking with appropriate VLANs.
- Perform zoning on MDS switches.
- Create and assign defined storage space.
- Install the selected version of vSphere hypervisor to provisioned blades, as shown in Figure 23.

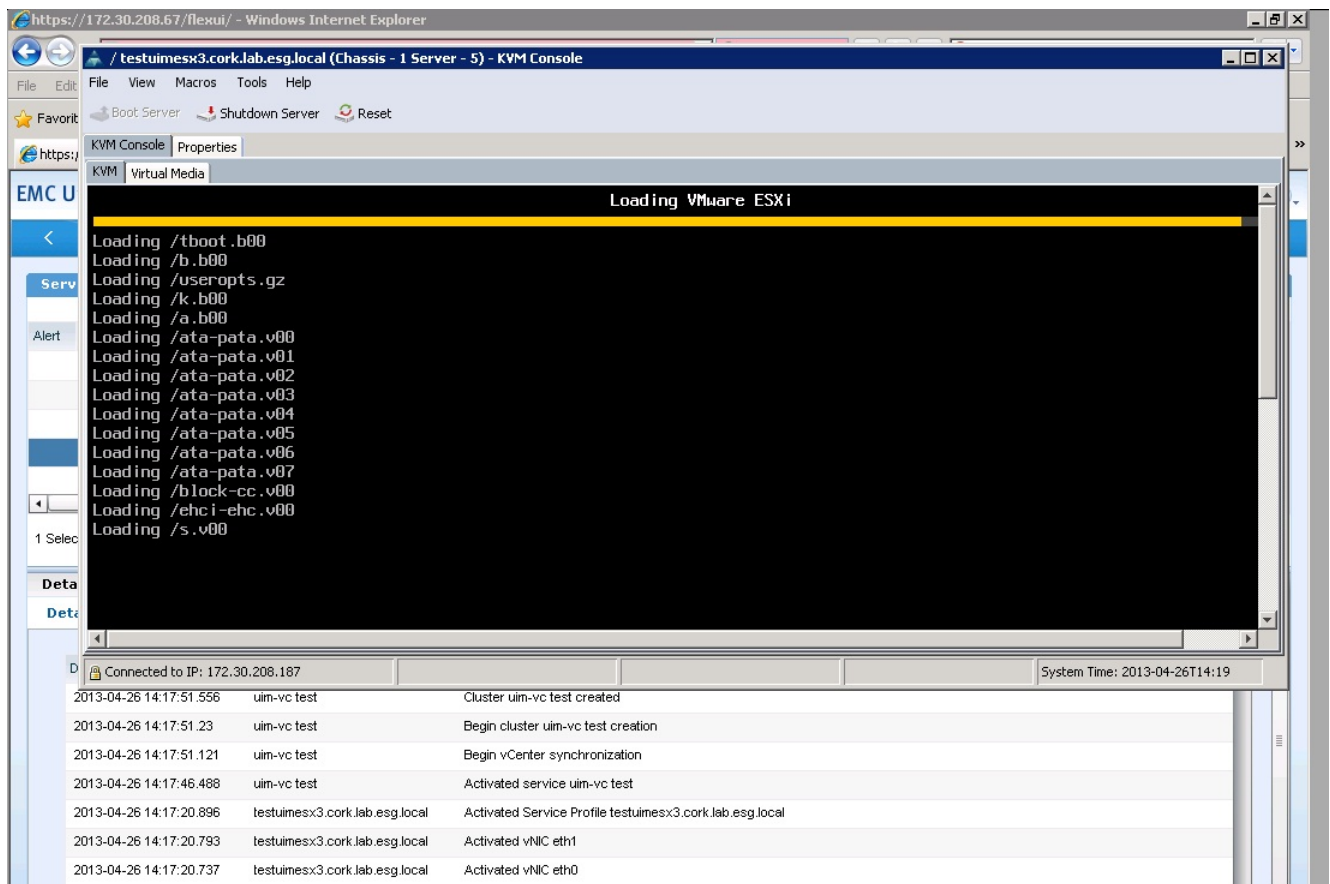


Figure 23. Provisioning process

After provisioning the vSphere hosts and before synchronizing with vCenter server, manually configure DNS and NTP settings on the hosts. Perform the synchronization process, as shown in Figure 24, to join deployed vSphere hosts to a vCenter server instance defined in the UIM service, apply the UIM service-defined vSphere HA and DRS cluster configuration, and make the hosts ready to utilize future Pivotal CF virtual machines provisioned with CF Operations Manager.

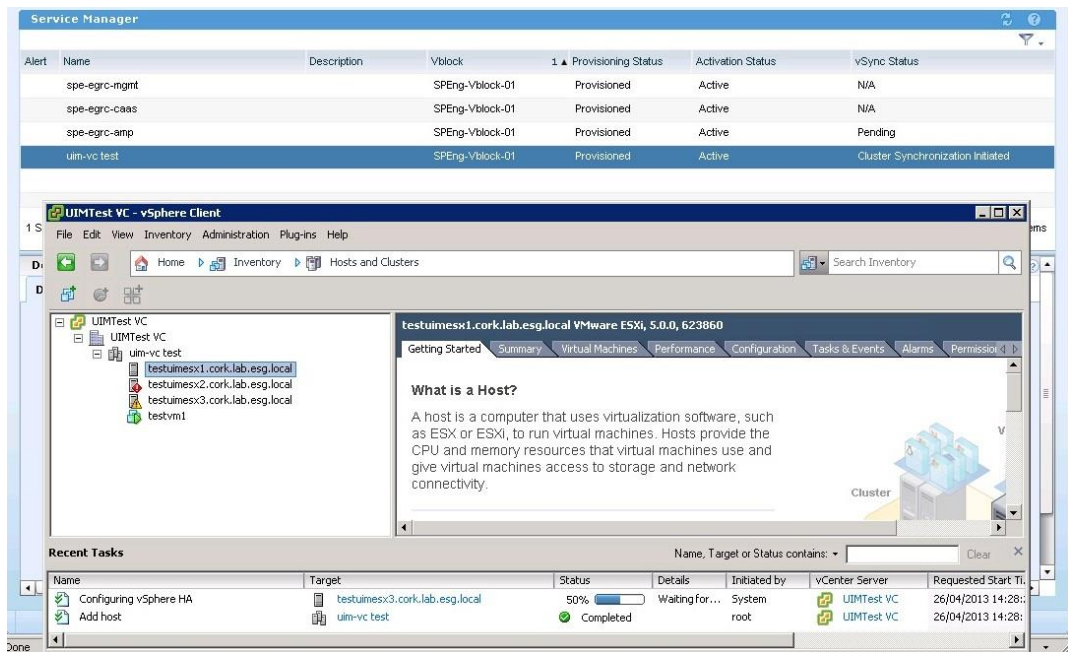


Figure 24. Synchronization process

Expanding storage pools with Unisphere

You can expand storage space provisioned as part of a UIM service offering without interrupting the running production environment. Assume that:

- The capacity of disk pools on the VNX storage array provisioned and dedicated to certain UIM service has reached its limit.
- You need more disk space to expand an existing Pivotal CF instance with additional components or services.

First, extend the existing disk pool with additional physical disk drives. You can expand a pool by adding drives to it, but you must do so at the array level with the EMC Unisphere storage management tool, as shown in Figure 25. It is a completely nondisruptive operation, and LUNs in the pool can use the increased capacity. Storage capacity can also be extended nondisruptively at the array level on the Vblock Systems family built with EMC Symmetrix VMAX arrays using Unisphere for VMAX.

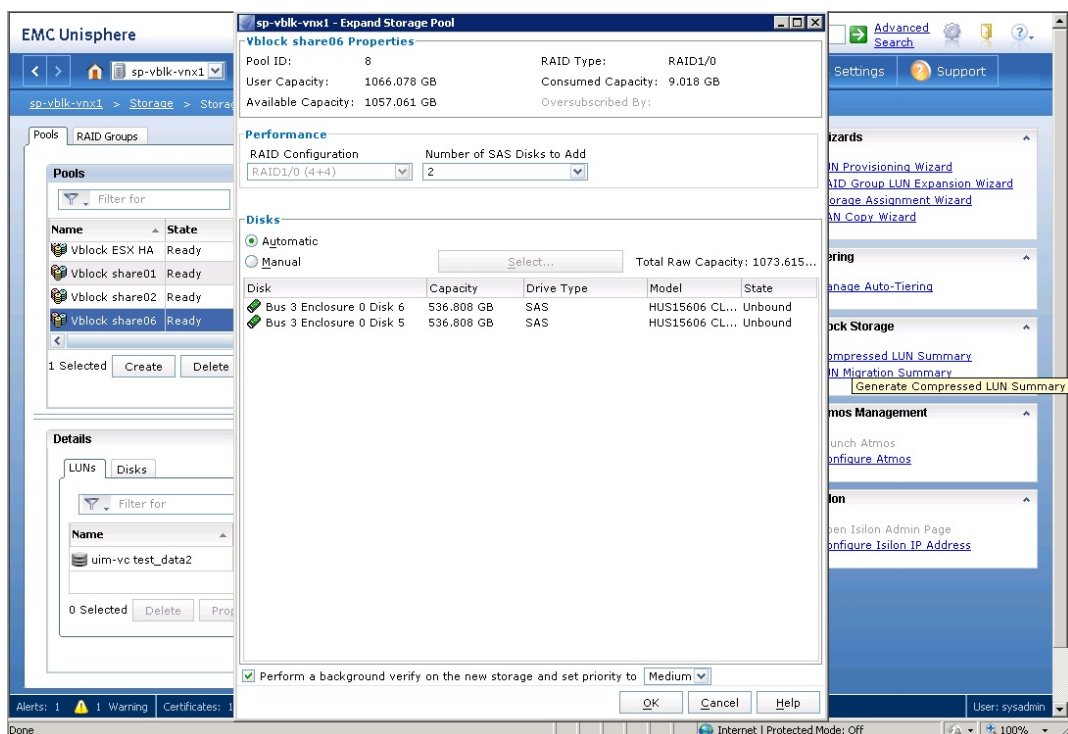


Figure 25. Expanding existing storage pool using Unisphere

When the disk pool is extended with additional raw disk capacity, extended resources are represented to the Vblock System during the discovery process, which is performed before every UIM service provisioning operation.

Increasing the capacity of provisioned block storage

After extending the existing disk pool with additional disks, you can extend the existing data storage as follows:

1. Enter a capacity value that is greater than that currently provisioned in the **Capacity GB** field on the **Storage** tab of the service properties.
2. Click **Save** to apply changes to the service.

To extend the existing VMFS datastore, change the capacity value, as shown in Figure 26.

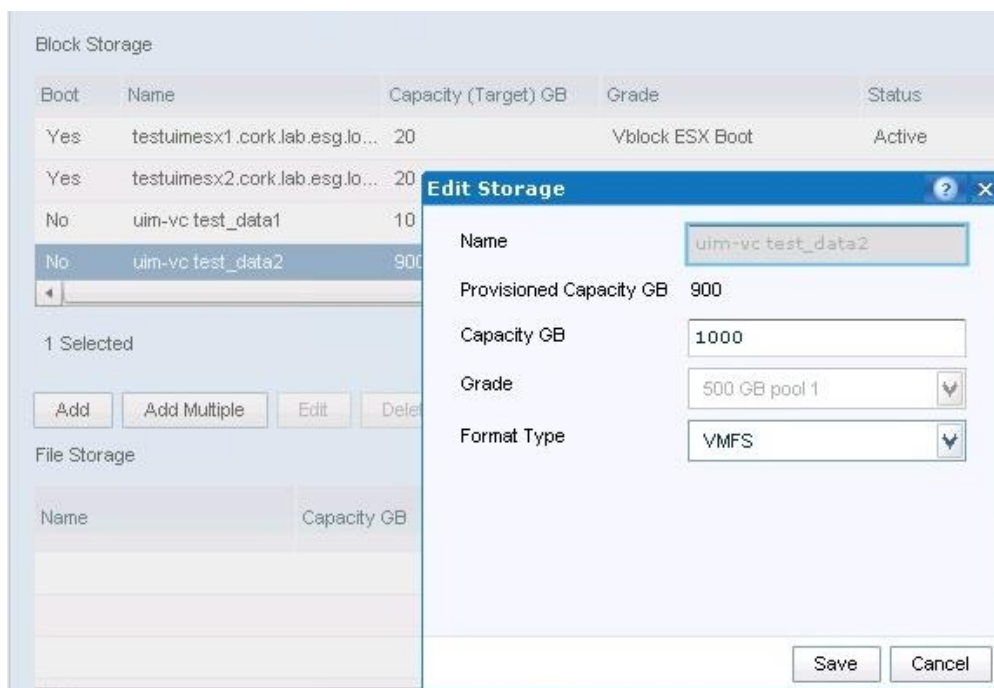


Figure 26. Extending disk space

When you extend datastore capacity, UIM performs the required changes, such as expanding the datastore, and synchronizing and updating changes with vCenter Server, as shown in Figure 27 and Figure 28.

Resource	Details
uim-vc test	Activating service uim-vc test
uim-vc test	Provisioning of service uim-vc test is complete
uim-vc test_data2	Provisioning of Data storage completed successfully
uim-vc test_data2	Data storage volume uim-vc test_data2: Volume updated successfully
uim-vc test_data2	Expanding shared data storage volume uim-vc test_data2 from 900 GB to 1000 GB
uim-vc test	Storage provisioning initiated on uim-vc test
uim-vc test	Zoneset activation initiated
SPEng-Vblock-01	Validating Vblock resources on SPEng-Vblock-01 for service uim-vc test
SPEng-Vblock-01	Refreshing Vblock SPEng-Vblock-01
uim-vc test	Provisioning of uim-vc test started

Figure 27. Expansion process performed by UIM

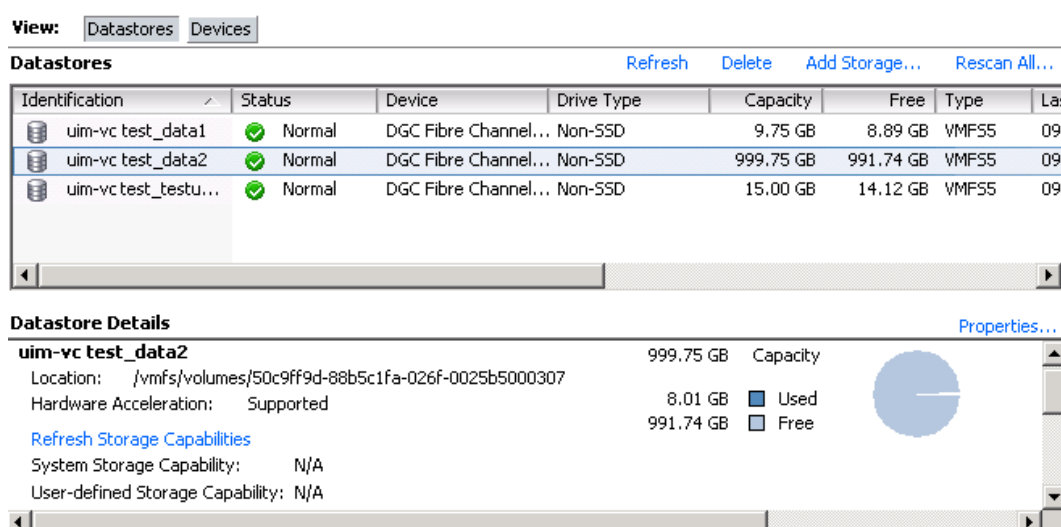


Figure 28. Extended datastore view from vCenter Server

Decommissioning a server or storage volume

Besides providing additional resources to Pivotal CF instance users, platform providers can reclaim resources if needed. Decommissioning a server or storage volume within a service permanently releases the service resources (such as IP addresses and MAC addresses) to the pools and releases blades and storage. The released resources are then available for other services. Decommissioning is a terminal state and cannot be reversed.

You can decommission an entire service or just certain components of a service. Servers must be deactivated, as shown in Figure 29, before they can be decommissioned. Storage volumes do not have to be deactivated before decommissioning. The next paragraphs describe in more detail how to partially decommission a blade server and storage volume.

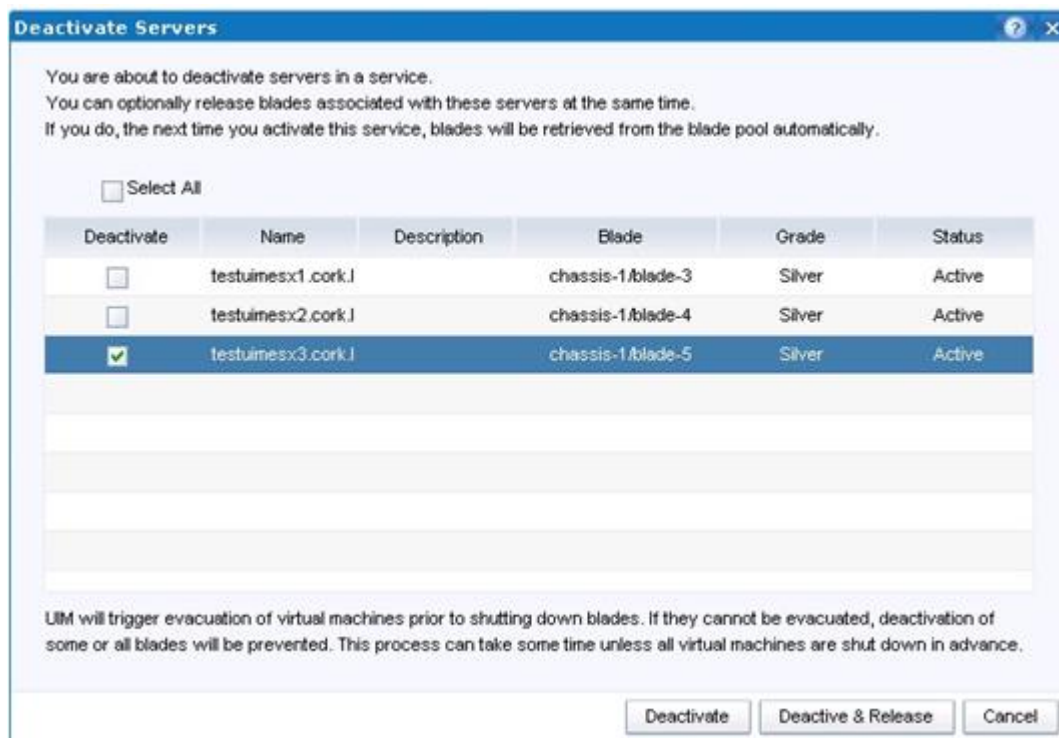


Figure 29. Select servers to deactivate

To decommission servers:

1. Confirm that no virtual machines are running on the servers you want to deactivate and on storage you want to decommission.
2. Deactivate the servers, as shown in Figure 29.
3. Select inactive servers and storage and decommission them from service, as shown in Figure 30.

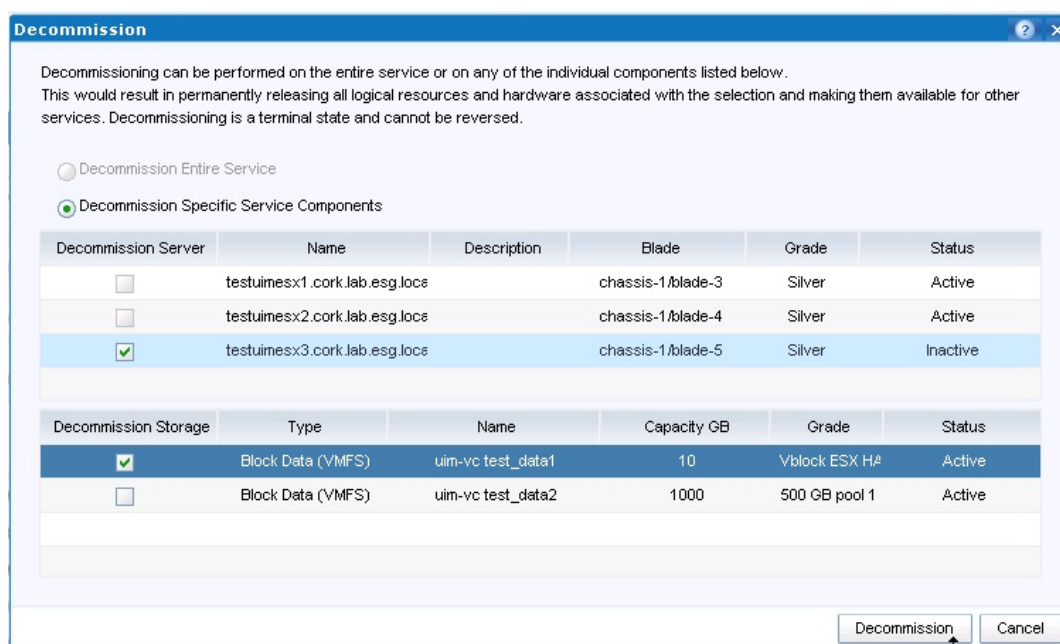


Figure 30. Select servers and storage volumes to be decommissioned

After the confirmation and decommissioning process, the service components and resources return to the state they were in before the service was created, and they become available for other purposes. Decommissioned service components and resources are deleted as a result of decommissioning and are returned to the UIM resource pool, as shown in Figure 31.

Administration											
Vblocks	Blade Pool	Storage Pool	Network Profiles	Service Offerings	Identity Pools	vCenter Instances	vCloud Instances	Isolation			
Vblock	Chassis	Blade	Model	RAM GB	Adapter 1	Adapter 2	CPU Speed GHz	Operational State	Grade	Status	Location
SPEng-Vblock-01	1	1	N20-B6620-1	48	N20-AQ0002		2.533	ok	Silver	in use: spe-egrc-amp	Cork ESG
SPEng-Vblock-01	1	2	N20-B6620-1	48	N20-AQ0002		2.533	ok	Silver	in use: spe-egrc-amp	Cork ESG
SPEng-Vblock-01	1	3	N20-B6620-1	48	N20-AQ0002		2.533	ok	Silver	in use: uim-vc test	Cork ESG
SPEng-Vblock-01	1	4	N20-B6620-1	48	N20-AQ0002		2.533	ok	Silver	in use: uim-vc test	Cork ESG
SPEng-Vblock-01	1	5	N20-B6620-1	48	N20-AQ0002		2.533	power-off	Silver	in use: uim-vc test	Cork ESG
SPEng-Vblock-01	1	6	N20-B6620-1	48	N20-AQ0002		2.533	unassociated	Silver	available	Cork ESG
SPEng-Vblock-01	1	7	N20-B6620-1	48	N20-AQ0002		2.533	unassociated	Silver	available	Cork ESG
SPEng-Vblock-01	1	8	N20-B6620-1	48	N20-AQ0002		2.533	unassociated	Silver	available	Cork ESG
SPEng-Vblock-01	2	1	N20-B6625-1	192	N20-AC0002		3.333	ok	Platinum	in use: spe-egrc-mgmt	Cork ESG
SPEng-Vblock-01	2	2	N20-B6625-1	96	N20-AC0002		3.333	ok	Gold	in use: spe-egrc-caas	Cork ESG
SPEng-Vblock-01	2	3	N20-B6625-1	192	N20-AC0002		3.333	ok	Platinum	in use: spe-egrc-mgmt	Cork ESG
SPEng-Vblock-01	2	4	N20-B6625-1	96	N20-AC0002		3.333	unassociated	Gold	available	Cork ESG
SPEng-Vblock-01	2	5	N20-B6625-1	192	N20-AC0002		3.333	unassociated	Platinum	available	Cork ESG
SPEng-Vblock-01	2	6	N20-B6625-1	96	N20-AC0002		3.333	ok	Gold	in use: spe-egrc-caas	Cork ESG
SPEng-Vblock-01	2	7	N20-B6625-1	192	N20-AC0002		3.333	unassociated	Platinum	available	Cork ESG
SPEng-Vblock-01	2	8	N20-B6625-1	96	N20-AC0002		3.333	unassociated	Gold	available	Cork ESG

Figure 31. Overview of Vblock System blade server resources

Providing PaaS involves delivering a wide range of different resources (storage, RAM, CPU, networking) in a dynamic and fast-changing environment. Implementing appropriate monitoring and capacity planning solutions is critically important. Monitoring and analysis of existing infrastructure resources utilization, discovering actual and potential bottlenecks, and planning and forecasting future resource needs are everyday tasks in achieving the goal of continuous delivery of guaranteed resources. Having reliable figures for forecasting and planning environment expansion is crucial and is described in detail in [Monitoring and capacity planning](#) on [page 5](#).

Monitoring and capacity planning

The service provider's primary goal is to achieve a level of service assurance that satisfies SLA and quality assurance (QA) parameters. This can be a major challenge when dealing with dynamically changing, fast-growing, and large-scale cloud environments supporting multiple PaaS instances. This section describes functions of UIM and VMware monitoring and capacity planning solutions that can help you to overcome those challenges.

UIM monitoring

UIM is a centralized availability monitoring solution. It enables you to correlate events across Vblock System components, infrastructure services, and virtual applications. It can be used to determine the root cause of issues or the impact on infrastructure services and virtual machines. UIM accesses the information related to provisioning, configuration, and compliance to build real-time topology and service views.

Such integration gives you deep visibility into physical, logical, and virtual abstraction layers so you can quickly locate and understand the topology and interconnections of the physical and virtual components of the Vblock System, as shown in Figure 32.

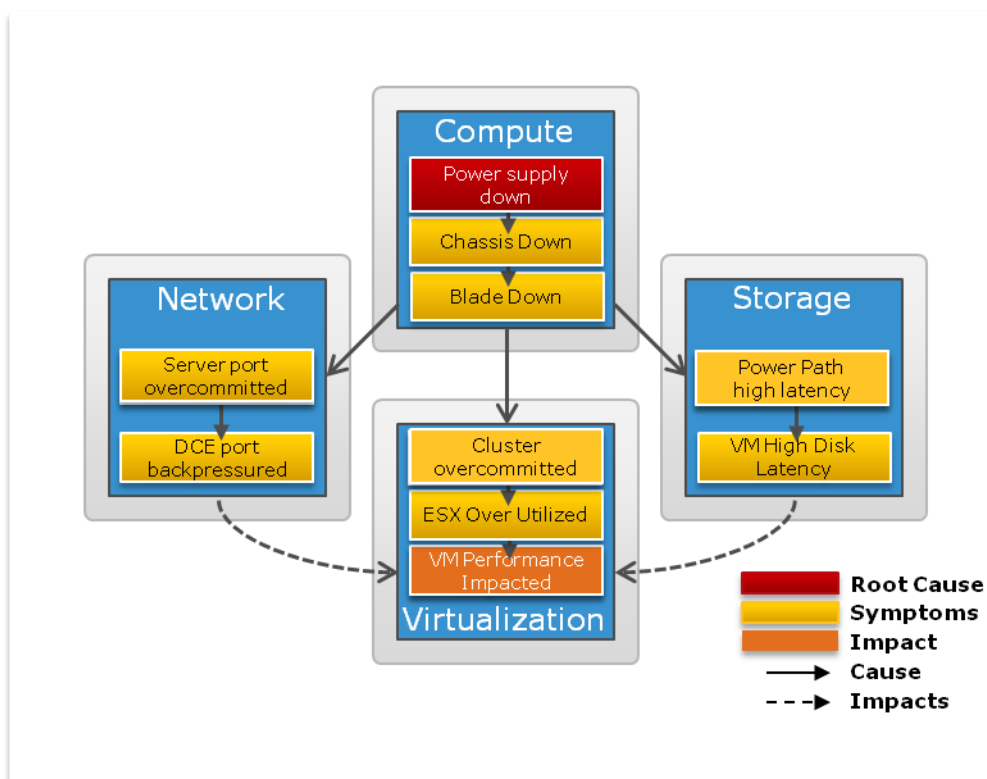


Figure 32. UIM root cause and impact analysis

Using topology views and the alerts console helps you to quickly understand the level of significance of different problems such as availability, performance, redundancy, and capacity.

Figure 33 shows the logical topology overview of the UIM service. You can see provisioned service components such as network segments, vSphere hosts, and storage datastores and their connections. The orange icon in Figure 33 indicates that there is an issue with one of the blades provisioned as part of UIM service: **uim-vc test** on Vblock-01.

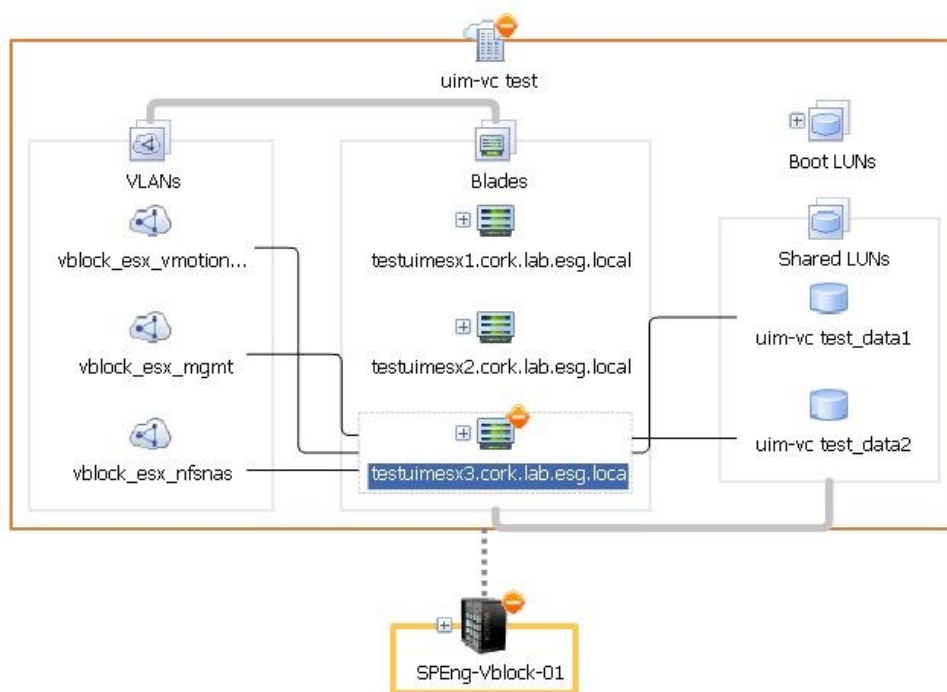


Figure 33. UIM service topology overview

For more details on the issue, click the **Vblock** icon. You can switch to a view that provides a physical topology of the Vblock System. Figure 34 shows an overview of the physical components of a Vblock System, such as blades, UCS chassis, fabric interconnects, MDS switches, storage, and their connections and relationships.

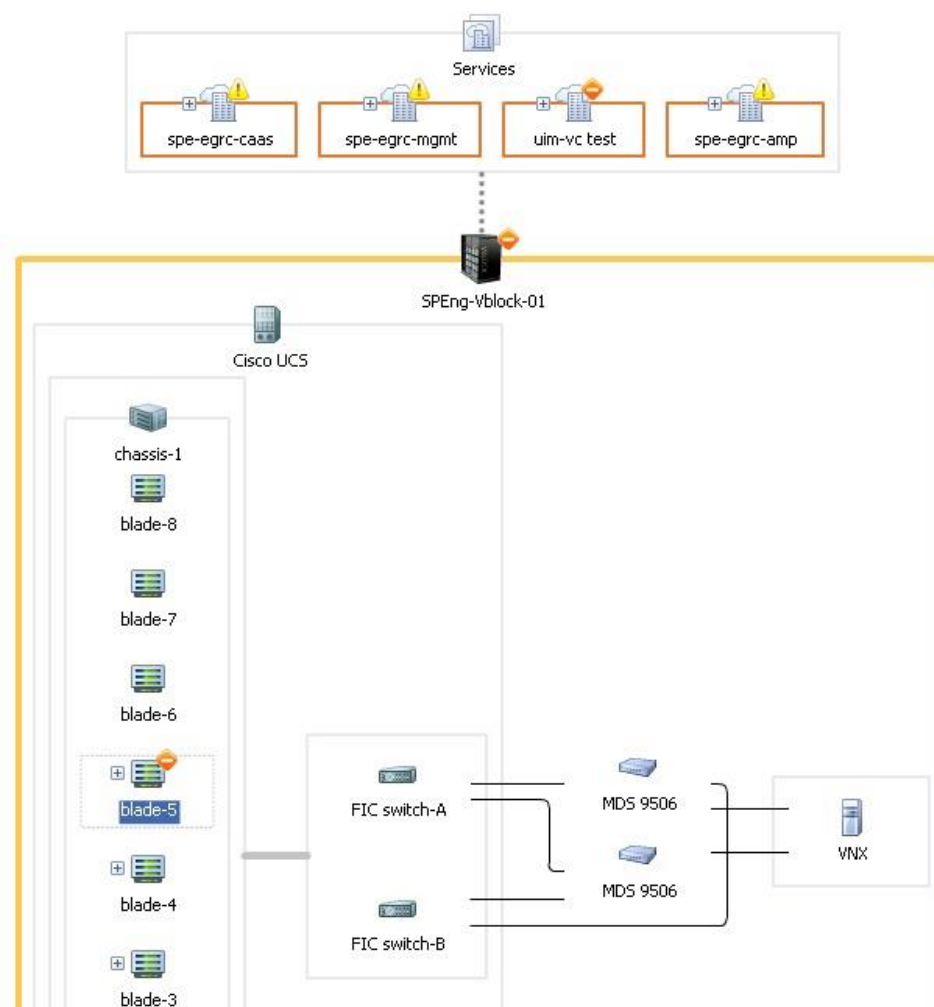


Figure 34. Vblock System physical components and topology overview

From the topology overview, you can determine that the issue is related only to **blade-5** in **chassis-1**, and the issue has no effect on the rest of the Vblock System environment. You can then look into further details, tracing the issue on the **uim-vc test** service screen, as shown in Figure 35.

Map Details - uim-vc test					
Properties		Operating System	Service Offering		Alerts
Severity	RCA Type	Component Type	Affected Component		Message
Warning	Root	Blade	172.30.208.66/sys/chassis-1/blade-5		compute-Blade-Hard-Shutdown
Warning	Impact	UIM_Vblock	spe-vblk-uimp.cork.lab.esg.local/SPEng-Vblock-01		AvailabilityImpacted-RCA
Warning	Impact	UIM_ManagedService	spe-vblk-uimp.cork.lab.esg.local/uim-vc test		AvailabilityImpacted-RCA
Warning	N/A	VirtualCircuit	172.30.208.66/sys/chassis-1/blade-5/fabric-A/path-1 dcx-Vc-down		
Warning	N/A	VirtualCircuit	172.30.208.66/sys/chassis-1/blade-5/fabric-B/path-1 dcx-Vc-down		
Warning	N/A	VirtualCircuit	172.30.208.66/sys/chassis-1/blade-5/fabric-B/path-2 dcx-Vc-down		

Figure 35. Root cause and impact details overview

From the information provided about root causes and impacts, you can see that a powered-down vSphere host impacts the availability of Pivotal CF services running on an infrastructure provisioned within **uim-vc test** service.

UIM can integrate with EMC infrastructure platforms as well as third-party tools via SNMP, sys-log messaging, and RESTful APIs for integration with various customized environments and solutions. Integration with VMware vCenter server provides you with data from virtual Pivotal CF PaaS environments running on vCenter clusters.

In addition to availability, performance and capacity should also be monitored. The next section provides some related considerations.

VMware vCenter Operations Manager

VMware vCenter Operations Manager (vCops) contributes to an integrated approach to performance, capacity, and configuration management. vCops uses analytics to provide the intelligence and visibility needed to ensure service levels in virtual and cloud environments.

vCops has functions that can help service providers achieve the following goals:

- Eliminate or significantly reduce the manual problem-solving effort in the environment.
- Proactively manage core service and cloud infrastructure performance, and utilize infrastructure resources optimally.
- Provision proactive warnings regarding performance issues before problems affect the end user. Realtime performance dashboards enable service providers to meet their SLAs by highlighting potential performance issues before end users notice these issues.

Infrastructure maintenance and operations teams need the end-to-end visibility and intelligence to make fast, informed operational decisions to proactively ensure service levels in cloud environments. They need to get promptly to the root cause of performance problems, optimize capacity in real time, and maintain compliance in a dynamic environment of constant change.

The vCenter Operations Management Suite offers many features and functions to deliver quality of service, operational efficiency, and continuous compliance for your dynamic cloud infrastructure and business-critical applications. The next section focuses on the crucial functionality of capacity forecasting and planning that should be used when providing PaaS platform environments.

Capacity forecasting

As previously described, deploying new services and related components on a Pivotal CF platform is fast and easy. DEA is the component that manages the lifecycle of applications hosted on a Pivotal CF instance. As the number of customers and applications hosted on a Pivotal CF instance grows, the service provider administrator must increase the number of DEA instances to accommodate the increased demand for staging applications.

Changing configuration parameters on the Elastic Environment Resource page (for example, increase the number of DEA instances as shown in Figure 36) and clicking Install Update will rapidly provision virtual machines and component instances in them.

JOB	INSTANCES	CPU	RAM (MB)	EPHEMERAL DISK (MB)	PERSISTENT DISK (MB)
DEA	<input type="text" value="5"/>	<input type="text" value="2"/>	<input type="text" value="16384"/>	<input type="text" value="32768"/>	<input type="text" value="0"/>

Figure 36. Changing the number of DEA instances

This section describes capacity planning functions that can help you to predict behavior and impacts of upgrades of current Pivotal CF instances with new services or new Pivotal CF deployments on underlying infrastructure and available resources.

Forecasting data for capacity risks in vCOPs involves creating capacity scenarios to examine the demand and supply of resources in the virtual infrastructure.

A what-if scenario is a supposition about how capacity and load might change if certain conditions influenced by an increased or decreased number of ESX hosts, storage, or virtual machines in environment occur. The what-if scenario does not involve making actual changes to your virtual infrastructure; however, if you implement the scenario, you know in advance what your capacity requirements are.

To create a scenario, you can use models and profiles based on current resource consumption in the existing environment, as shown in Figure 37. Alternatively, you can manually define amounts of virtual machine RAM, storage, CPU, and utilization in a new consumption profile, as shown in Figure 38, to predict potential impact of growth.

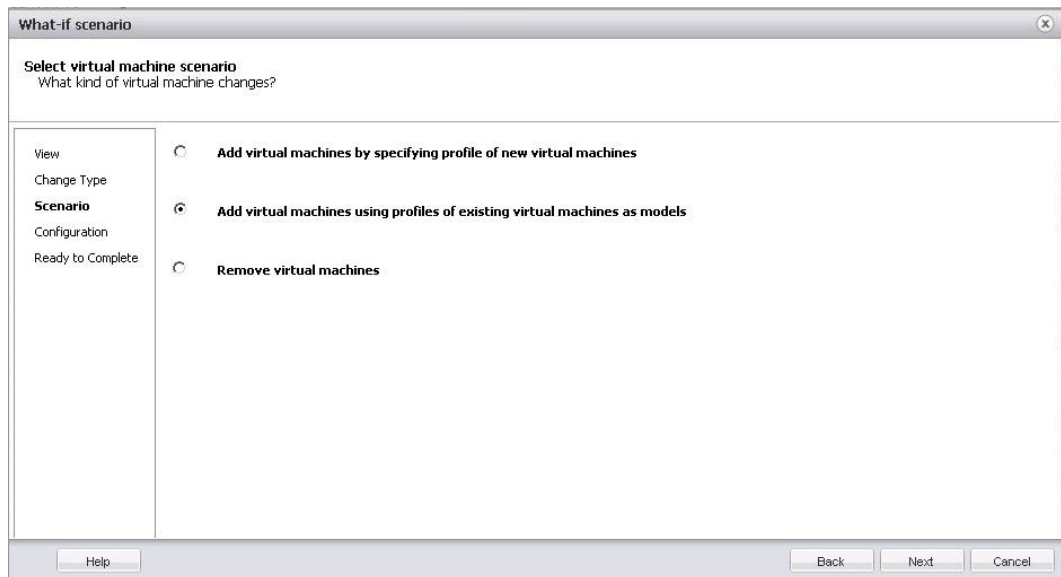


Figure 37. Choosing virtual machine consumption models and profiles

What-if scenario

New virtual machine configuration
Specify the configuration and projected capacity usage of new virtual machines

View
Change Type
Scenario
Configuration
Ready to Complete

Virtual machine count: 25

Specify new virtual machines

vCPU: 1 x 2.4 GHz 50 %
Reservation: 0 GHz
Limit: 0 GHz
☒ Unlimited

Memory: 6144 MB 50 %
Reservation: 0 MB
Limit: 0 MB
☒ Unlimited

Specify Virtual Disk configuration

Virtual Disk: Thick Configuration: 4 GB 50 %
☐ Unlinked-Clone Disk

Help Back Next Cancel

VM Population

Population Summary
SPE-VBLK01

vCPU: 42 CPU Cores at 1.04GHz Usage
vMEM: 162GB at 6.57GB Usage
Virtual Disk: 4,391GB at 1,891.73GB Usage
Disk I/O: 0.67MBps Usage

Population Details

Small VM Profile

vCPU: 1 vCPU at 5.12% Utilization
vMEM: 2.5GB at 29.85% Utilization
Virtual Disk: 165GB at 45.9GB Usage
Virtual Machines: 5

Medium VM Profile

vCPU: 1 vCPU at -% Utilization
vMEM: 6GB at -% Utilization
Virtual Disk: 100GB at 26.12GB Usage

Host Population

	CPU	Memory
Smallest	8 x 2.4GHz	47.94GB
Largest	8 x 2.4GHz	47.94GB

Figure 38. Specifying configuration and projected capacity usage of new virtual machines

If you choose to set a new, specific virtual machine profile, you can make detailed specifications that give you the option to include and predict specific resource utilizations, reservations, and limits to get as accurate a projection as possible.

Figure 39 shows that sufficient resources are not available for a planned deployment scenario consisting of 50 or 85 new virtual machines. In this case, you can easily provision new vSphere hosts using UIM, as described in previous sections.

Remaining Capacity (VMs)			
	Actual	Add 50 New VMs	Add 85 New VMs
Host CPU	71.58 VMs	-37.1 VMs	-80.7 VMs
Host Memory	36.26 VMs	-38 VMs	-81 VMs
Disk Space	16.58 VMs	31.16 VMs	25.52 VMs
Disk I/O Read	3,512 VMs	2,105 VMs	2,054 VMs
Disk I/O Write	472.4 VMs	427.3 VMs	392.5 VMs
Disk I/O Reads per Second	3,117 VMs	1,492 VMs	1,439 VMs
Disk I/O Writes per Second	89.29 VMs	39.88 VMs	4.898 VMs
Network I/O Received Rate	1,943 VMs	2,530 VMs	2,523 VMs
Network I/O Transmitted Rate	26,143 VMs	35,712 VMs	36,124 VMs
Summary	16.58 VMs	-38 VMs	-81 VMs

Figure 39. Capacity summary showing insufficient CPU and RAM resources

Before you provision new hardware resources, you can create hardware change scenarios to determine the effect of adding, removing, or updating the hardware capacity in a vSphere cluster. You can create a scenario that models changes to hosts and datastores, as shown in Figure 40 and Figure 41.

What-if scenario

Hosts
Please configure your Host population

Count	Name	CPU Total	Mem Total
1	Added Host	1 x 2 GHz	2 GB
1	spe-vblk-amp01.c	8 x 2.4GHz	47.94GB
1	spe-vblk-amp02.cork.lab.e...	8 x 2.4GHz	47.94GB

Buttons: Add Host, Remove Host, Restore Host

Cluster Population

Population Summary
SPE-VBLK01

CPU: 16 CPU Cores at 2.98GHz Usage
MEM: 95.87GB at 62.53% Usage

Population Details

spe-vblk-amp01.cork.lab.esg.local

CPU: 8 x 2.4GHz at 6.22% Usage
MEM: 47.94GB at 52.53% Usage
Virtual Machines: 18

spe-vblk-amp02.cork.lab.esg.local

CPU: 8 x 2.4GHz at 9.35% Usage
MEM: 47.94GB at 77.19% Usage
Virtual Machines: 6

VM Population

	vCPU	vMemory
Smallest	1 x 2.4GHz	1GB
Largest	2 x 2.4GHz	16GB

Buttons: Help, Back, Next, Cancel

Figure 40. Specifying number of hosts and amount of CPU and memory

What-if scenario

Datastores
Please configure your Datastore population

Count	Name	Disk Total
1	Added Datastore	100 GB
1	spe-vblk-amp01	460.75GB
1	spe-vblk-amp02	460.75GB
1	TEMP.	1,012.97GB
1	Vblock01-AMP-1	2,016.75GB

Buttons: Add Datastore, Remove Datastore, Restore Datastore

Datastore Population

Population Summary
SPE-VBLK01

Capacity: 3,951.22GB at 1,891.75GB Usage
Disk I/O: 812.67KBps Usage

Population Details

spe-vblk-amp01

Capacity: 460.75GB at 0.95GB Usage
Disk I/O: 39.53KBps Usage
Shared: No
Links: 1

spe-vblk-amp02

Capacity: 460.75GB at 0.95GB Usage
Disk I/O: 37.72KBps Usage
Shared: No
Links: 1

VM Population

	vDisk	vDisk I/O
Smallest	4GB	7.65KBps
Largest	2,093GB	119.12KBps

Buttons: Help, Back, Next, Cancel

Figure 41. Specifying datastore size

The what-if scenario function for capacity planning allows you to compare how adding different numbers of virtual machines and hardware will impact your actual environment, as shown in Figure 42.

Details				
Virtual Machine Capacity				
Total Capacity (VMs)				
	Actual	Add 45 New VMs	Hardware Change #12	Add 25 New VMs
Host CPU	121.8 VMs	36.28 VMs	134.5 VMs	39.54 VMs
Host Memory	82.61 VMs	34.27 VMs	197 VMs	36.7 VMs
Disk Space	48.24 VMs	112 VMs	60.93 VMs	87 VMs
Disk I/O Read	3,423 VMs	2,568 VMs	3,423 VMs	2,635 VMs
Disk I/O Write	371.8 VMs	429.4 VMs	371.8 VMs	421.5 VMs
Disk I/O Reads per Second	2,831 VMs	1,909 VMs	2,831 VMs	1,978 VMs
Disk I/O Writes per Second	147.8 VMs	155.4 VMs	147.8 VMs	154.5 VMs
Network I/O Received Rate	8,189 VMs	11,583 VMs	8,189 VMs	11,016 VMs
Network I/O Transmitted R...	8,679 VMs	12,413 VMs	8,679 VMs	11,782 VMs
Summary	48.24 VMs	34.27 VMs	60.93 VMs	36.7 VMs

Figure 42. Compared scenarios

Assume that:

1. You have a request to deploy an additional 45 DEA instances in the existing Pivotal CF instance.
2. You plan to purchase blade servers compliant with a certain specification.
3. You want to see if you could deploy an additional 25 DEA instances (that is, 70 total additional instances).

In Figure 42, each column shows how an individual change affects resources in your environment. In Figure 34, the Combined Scenarios section shows you the cumulative effect of hardware purchasing and an overall expansion of 70 virtual machines.

Details				
Virtual Machine Capacity				
Total Capacity (VMs)				
	Actual	Combined		
Host CPU	121.8 VMs	38.38 VMs		
Host Memory	82.61 VMs	78.91 VMs		
Disk Space	48.24 VMs	174.2 VMs		
Disk I/O Read	3,423 VMs	2,535 VMs		
Disk I/O Write	371.8 VMs	433.9 VMs		
Disk I/O Reads per Second	2,831 VMs	1,874 VMs		
Disk I/O Writes per Second	147.8 VMs	156 VMs		
Network I/O Received Rate	8,189 VMs	11,917 VMs		
Network I/O Transmitted R...	8,679 VMs	12,787 VMs		
Summary	48.24 VMs	38.38 VMs		

What-If Scenarios

[Create New Scenario...](#)

Combined Scenarios

- Add 45 New VMs
- Hardware Change #12
- Add 25 New VMs

Combine Scenarios

View Information

[Change Def...icity Configuration](#)

Utilization Calculation
Every day and all hours

Capacity Calculation Rule
Use last known capacity per interval

Capacity Buffer Limits

Figure 43. Combined scenarios

Conclusion

Elastic provisioning is a concept in cloud computing in which computing resources can be easily scaled up and down by the cloud service provider. Elastic provisioning is the ability of a cloud service provider to provision flexible computing resources when and where they are required. The elasticity of these resources can be in terms of processing power, memory, storage, bandwidth, and so on. This document indicates the importance of having an elastic and scalable IaaS platform on which to support the hosting of dynamically changing and fast-growing PaaS platforms.

Pivotal CF PaaS is designed to create a PaaS framework that is easy to scale. To achieve this kind of flexibility, Pivotal CF is designed as a modular system. The CF Operations Management platform, shown in Figure 44, enables the deployment of new Pivotal CF instances and updating of existing instances. CF Operations Manager enables the rapid distributed deployment of Pivotal CF components on multiple nodes and automatically provisions and deploys related virtual machines and services in them.



JOB	INSTANCES	CPU	RAM (MB)	EPHEMERAL DISK (MB)	PERSISTENT DISK (MB)
NATS	1	1	1024	2048	0
Health Manager	1	1	1024	2048	0
NFS Server	1	1	1024	2048	102400
Cloud Controller Database	1	1	1024	2048	2048
Cloud Controller	1	1	1024	10240	0
Router	1	1	1024	2048	0
Syslog	1	1	1024	2048	8192
Collector	1	1	1024	2048	0
UAA Database	1	1	1024	2048	8192
UAA	1	1	1024	2048	0
SAML Login	1	1	1024	2048	0
Console Database	1	1	1024	2048	1024
DEA	1	2	16384	32768	0
Loggregator Server	1	1	1024	2048	0
Loggregator Router	1	1	1024	2048	0
Compilation	5	2	1024	4096	0

Figure 44. CF Operations Manager GUI-based deployment of Pivotal CF

Figure 45 shows the vCOPS solution that you can leverage to deliver quality of service, attain operational efficiency, and gather current capacity capabilities while forecasting the influences of future Pivotal CF deployments or upgrades in your cloud infrastructure.

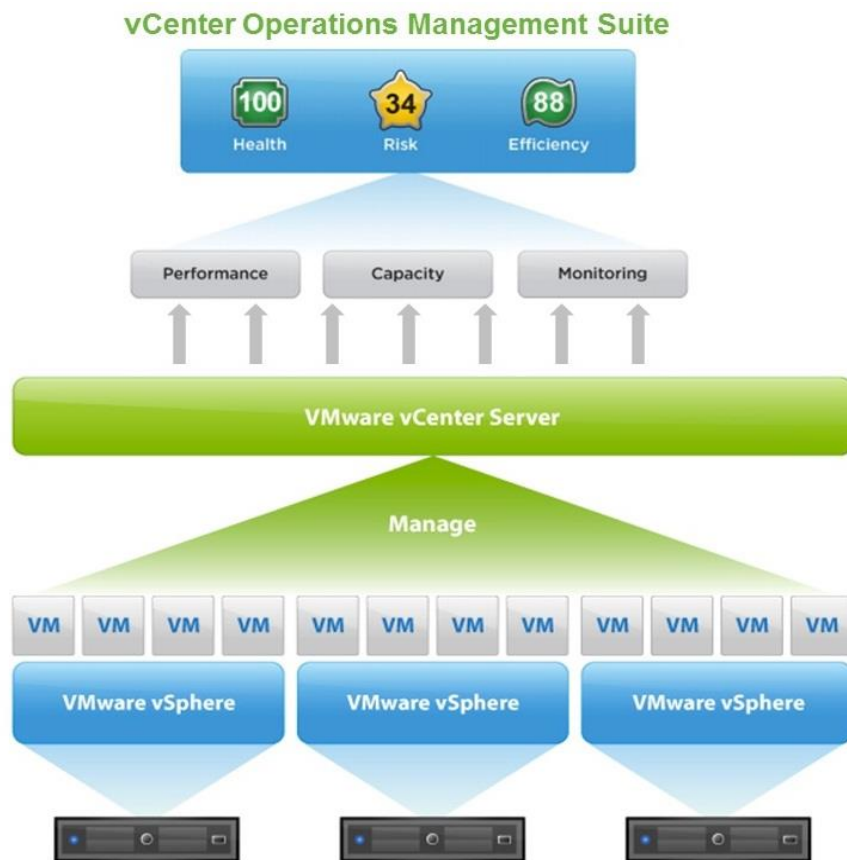


Figure 45. Capacity planning and proactive monitoring solution

The service provider's primary goal is to provide service assurance within SLA and QA parameters, which can be a challenge in dynamically changing, fast-growing large-scale PaaS environments.

Pivotal CF PaaS can contain a large number of component instances. The number is easy to change just by changing configuration parameters in the Elastic Runtime Resources page. Running install update deploys them seamlessly. Therefore, it is crucial to have proactive performance monitoring and capacity planning solutions in place.

Even basic Pivotal CF instance sizing is challenging, as there are a minimum of 20 components you must take into consideration and predicting front-end/back-end loads of hosted applications is difficult. A combination of proactive capacity monitoring and elastic on-demand provisioning gives you the flexibility to scale as needed.

To support comprehensive, dynamic, and fast-growing development environments like Pivotal CF, service providers must ensure the stability of their underlying cloud compute infrastructure, which must provide availability, scalability, flexibility, and performance to the PaaS platform and its services. This white paper has addressed these challenges, providing a provisioning solutions overview and considerations for building scalable Pivotal CF PaaS environments, with an elastic and easy-to-deploy underlying IaaS. The IaaS infrastructure is provided by the EMC UIM product and the Vblock Systems, as shown in Figure 46.

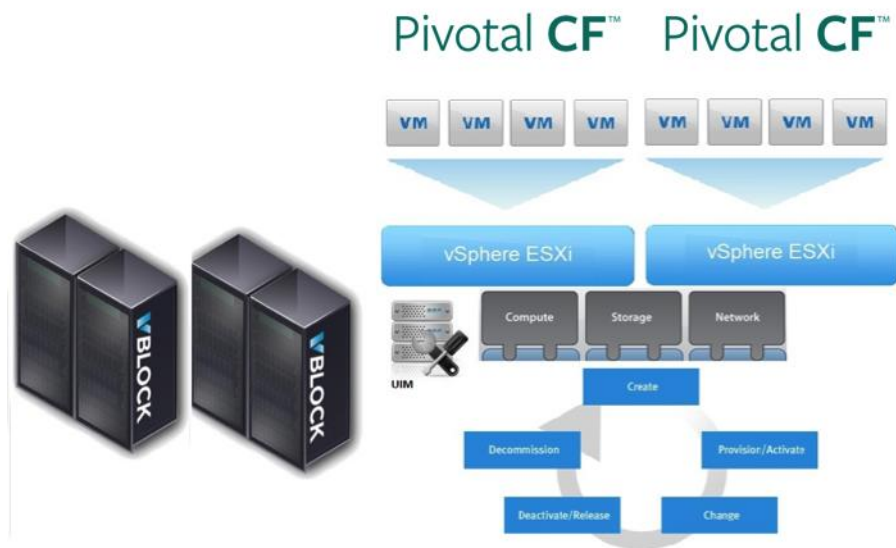


Figure 46. UIM elastic provisioning and Vblock System IaaS

References

For additional information, refer to the following websites:

EMC <http://www.emc.com/data-center-management/unified-infrastructure-manager.htm#!resources>

Pivotal <http://www.gopivotal.com/paas>

VCE <http://www.vce.com/products/vblock/overview>

VMware <http://www.vmware.com/products/vsphere/>
<http://www.vmware.com/products/vcenter-operations-management/>