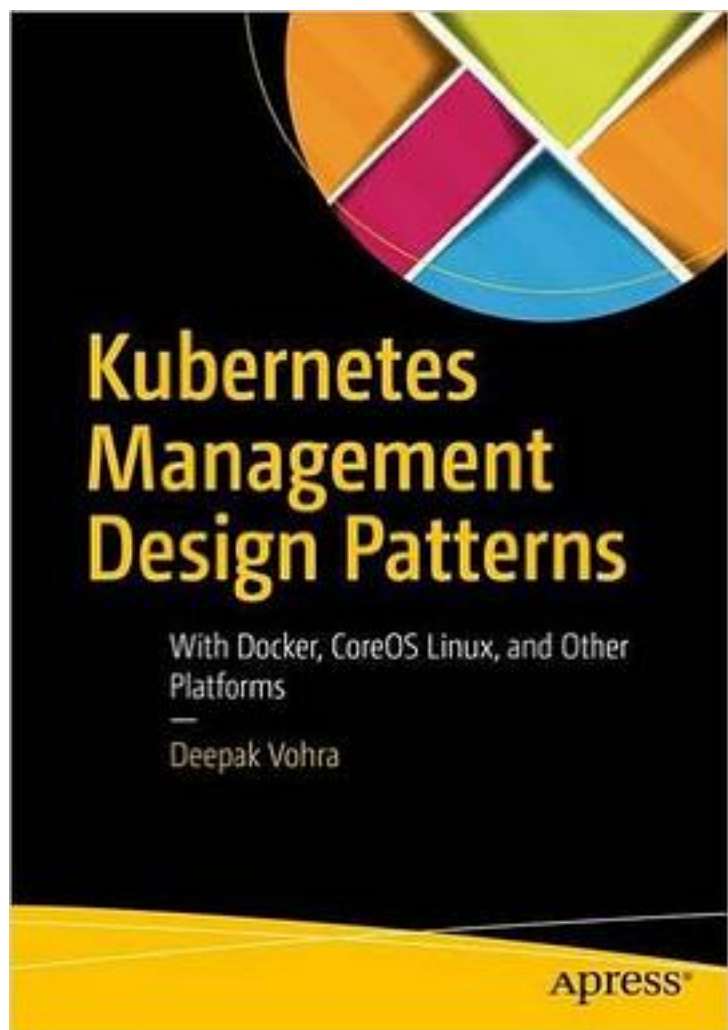


Kubernetes Management Design Patterns: With Docker, CoreOS Linux, and Other Platforms



[Kubernetes Management Design Patterns: With Docker, CoreOS Linux, and Other Platforms_下载链接1](#)

著者:Deepak Vohra

出版者:Apress

出版时间:2017-2-23

装帧:Paperback

isbn:9781484225974

Take container cluster management to the next level; learn how to administer and configure Kubernetes on CoreOS; and apply suitable management design patterns such as Configmaps, Autoscaling, elastic resource usage, and high availability. Some of the other features discussed are logging, scheduling, rolling updates, volumes, service types, and multiple cloud provider zones.

The atomic unit of modular container service in Kubernetes is a Pod, which is a group of containers with a common filesystem and networking. The Kubernetes Pod abstraction enables design patterns for containerized applications similar to object-oriented design patterns. Containers provide some of the same benefits as software objects such as modularity or packaging, abstraction, and reuse.

CoreOS Linux is used in the majority of the chapters and other platforms discussed are CentOS with OpenShift, Debian 8 (jessie) on AWS, and Debian 7 for Google Container Engine.

CoreOS is the main focus because Docker is pre-installed on CoreOS out-of-the-box. CoreOS:

Supports most cloud providers (including Amazon AWS EC2 and Google Cloud Platform) and virtualization platforms (such as VMWare and VirtualBox) Provides Cloud-Config for declaratively configuring for OS items such as network configuration (flannel), storage (etcd), and user accounts Provides a production-level infrastructure for containerized applications including automation, security, and scalability Leads the drive for container industry standards and founded appc Provides the most advanced container registry, Quay

Docker was made available as open source in March 2013 and has become the most commonly used containerization platform. Kubernetes was open-sourced in June 2014 and has become the most widely used container cluster manager. The first stable version of CoreOS Linux was made available in July 2014 and since has become one of the most commonly used operating system for containers.

What You'll Learn

Use Kubernetes with Docker Create a Kubernetes cluster on CoreOS on AWS Apply cluster management design patterns Use multiple cloud provider zones Work with Kubernetes and tools like Ansible Discover the Kubernetes-based PaaS platform OpenShift Create a high availability website Build a high availability Kubernetes master cluster Use volumes, configmaps, services, autoscaling, and rolling updates Manage compute resources Configure logging and scheduling

Who This Book Is For

Linux admins, CoreOS admins, application developers, and container as a service (CAAS) developers. Some pre-requisite knowledge of Linux and Docker is required. Introductory knowledge of Kubernetes is required such as creating a cluster, creating a Pod, creating a service, and creating and scaling a replication controller. For introductory Docker and Kubernetes information, refer to Pro Docker (Apress) and Kubernetes Microservices with Docker (Apress). Some pre-requisite knowledge about using Amazon Web Services (AWS) EC2, CloudFormation, and VPC is also required.

作者介绍:

From the Back Cover

Take container cluster management to the next level; learn how to administer and configure Kubernetes on CoreOS; and apply suitable management design patterns such as Configmaps, Autoscaling, elastic resource usage, and high availability. Some of the other features discussed are logging, scheduling, rolling updates, volumes, service types, and multiple cloud provider zones. The atomic unit of modular container service in Kubernetes is a Pod, which is a group of containers with a common filesystem and networking. The Kubernetes Pod abstraction enables design patterns for containerized applications similar to object-oriented design patterns. Containers provide some of the same benefits as software objects such as modularity or packaging, abstraction, and reuse. CoreOS Linux is used in the majority of the chapters and other platforms discussed are CentOS with OpenShift, Debian 8 (jessie) on AWS, and Debian 7 for Google Container Engine. You will:

- Use Kubernetes with Docker
- Create a Kubernetes cluster on CoreOS on AWS
- Apply cluster management design patterns
- Use multiple cloud provider zones
- Work with Kubernetes and tools like Ansible
- Discover the Kubernetes-based PaaS platform OpenShift
- Create a high availability website
- Build a high availability Kubernetes master cluster
- Use volumes, configmaps, services, autoscaling, and rolling updates
- Manage compute resources
- Configure logging and scheduling

Read more

About the Author

Deepak Vohra is an Oracle Certified Associate and a Sun Certified Java Programmer. Deepak has published in Oracle Magazine, OTN, IBM developerWorks, ONJava, DevSource, WebLogic Developer's Journal, XML Journal, Java Developer's Journal, FTPOnline, and devx.

Read more

目录:

[Kubernetes Management Design Patterns: With Docker, CoreOS Linux, and Other Platforms_下载链接1](#)

标签

kubernetes

k8s

docker

评论

[Kubernetes Management Design Patterns: With Docker, CoreOS Linux, and Other Platforms 下载链接1](#)

书评

[Kubernetes Management Design Patterns: With Docker, CoreOS Linux, and Other Platforms 下载链接1](#)