

1. Record Nr.	UNINA9910162796003321
Autore	Vohra Deepak
Titolo	Kubernetes Management Design Patterns : With Docker, CoreOS Linux, and Other Platforms // by Deepak Vohra
Pubbl/distr/stampa	Berkeley, CA : , : Apress : , : Imprint : Apress, , 2017
ISBN	1-4842-2598-8
Edizione	[1st ed. 2017.]
Descrizione fisica	1 online resource (XX, 399 p. 560 illus., 500 illus. in color.)
Disciplina	004
Soggetti	Client/server computing Operating systems (Computers) Computer programming Servers Operating Systems Programming Techniques
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Place of publication from publisher's website (viewed February 14, 2017). Includes index.
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Part I Platforms -- 1. Kubernetes On AWS -- 2. Kubernetes on CoreOS -- 3. Kubernetes on Google Cloud Platform. Part 2 Administration and Configuration -- 4. Using Multiple Zones -- 5. Using the Tectonic Console -- 6. Using Volumes -- 7. Using Services -- 8. Using Rolling Updates -- 9. Scheduling Pods -- 10. Configuring Compute Resources -- 11. Using Configmaps -- 12. Setting Resource Quotas -- 13. Using Autoscaling -- 14. Configuring Logging. Part 3 High Availability -- 15. Using a HA Master with OpenShift -- 16. Developing a Highly Available Web Site.
Sommario/riassunto	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.

---