Table of Contents

Preface ix

1. Getting Started With Kubernetes 15
1.1 Using Kubernetes Without Installation 15
1.2 Installing the Kubernetes CLI `kubectl` 16
1.3 Installing Minikube To Run A Local Kubernetes Instance 17
1.4 Using Minikube Locally for Development 19
1.5 Starting your First Application on Minikube 20
1.6 Accessing the Dashboard in Minikube 20

2. Creating a Kubernetes Cluster 25
2.1 Installing `kubeadm` To Create A Kubernetes Cluster 25
2.2 Bootstrapping A Kubernetes Cluster Using `kubeadm` 26
2.3 Downloading A Kubernetes Release From GitHub 27
2.4 Downloading Client And Server Binaries 28
2.5 Using Hyperkube Image To Run A Kubernetes Master Node With Docker 29
2.6 Writing A Systemd Unit File To Run Kubernetes Components 31
2.7 Creating A Kubernetes Cluster On Google Kubernetes Engine (GKE) 34
2.8 Creating A Kubernetes Cluster On Azure Container Service (ACS) 35

3. Learning to Use the Kubernetes Client 39
3.1 Listing Resources 39
3.2 Deleting Resources 40
3.3 Watching Resource Changes With `kubectl` 41
3.4 Editing Resources With `kubectl` 42
3.5 Letting `kubectl` Explain Resources And Fields 42

4. Creating and Modifying Fundamental Workloads. 45
4.1 Creating A Deployment Using `kubectl run` 45
4.2 Creating Objects From File Manifests 46
4.3 Writing A Pod Manifest From Scratch 47
4.4 Launching Deployment Using A Manifest 48
4.5 Updating A Deployment 52

5. Working With Services. 55
5.1 Creating A Service To Expose Your Application 56
5.2 Verifying the DNS Entry of a Service 58
5.3 Changing The Type of a Service 58
5.4 Deploy An Ingress Controller on Minikube 60
5.5 Making Services Accessible From Outside The Cluster 61

6. Exploring the Kubernetes API and Key Metadata. 65
6.1 Discovering API Endpoints Of The Kubernetes API Server 65
6.2 Understanding The Structure Of A Kubernetes Manifest 67
6.3 Creating Namespaces To Avoid Name Collisions 68
6.4 Setting Quotas Within A Namespace 69
6.5 Labeling An Object 70
6.6 Using Labels For Queries 71
6.7 Annotating a Resource With One Command 73

7. Managing Specialized Workloads. 75
7.1 Running A Batch Job 75
7.2 Running a Task on a Schedule Within a Pod 77
7.3 Running Infrastructure Daemons Per Node 78
7.4 Managing Stateful and Leader-Follower Apps 79
7.5 Influencing Pods Startup Behavior 83

8. Volumes and Configuration Data. 85
8.1 Exchanging Data Between Containers Via A Local Volume 85
8.2 Passing An API Access Key To A Pod Using A Secret 87
8.3 Providing Configuration Data To An Application 91
8.4 Using A Persistent Volume With Minikube 93
8.5 Understanding Data Persistency on Minikube 96
8.6 Dynamically Provision Persistent Storage On GKE 99

9. Scaling. 101
9.1 Scaling A Deployment 102
9.2 Automatically Resizing a Cluster in GKE 102
9.3 Automatically Resizing a Cluster in AWS 105

9.4 Using Horizontal Pod Autoscaling on GKE 106

10. Security. 109
10.1 Providing A Unique Identity For An Application 109
10.2 Listing And Viewing Access Control Information 111
10.3 Controlling Access To Resources 116
10.4 Securing Pods 118

11. Monitoring and Logging. 121
11.1 Accessing The Logs of a Container 121
11.2 Recover From a Broken State With a Liveness Probe 122
11.3 Control Traffic Flow to a Pod Using a Readiness Probe 123
11.4 Adding Liveness and Readiness Probes To Your Deployments 124
11.5 Enabling Heapster on Minikube To Monitor Resources 126
11.6 Using Prometheus On Minikube 128
11.7 Using Elasticsearch-Fluentd-Kibana (EFK) On Minikube 133

12. Maintenance And Troubleshooting. 137
12.1 Enabling Autocomplete For `kubectl` 137
12.2 Removing a Pod From a Service 138
12.3 Access a ClusterIP Service Outside the Cluster 139
12.4 Understanding And Parsing Resource Statuses 140
12.5 Debugging Pods 142
12.6 Getting A Detailed Snapshot Of The Cluster State 146
12.7 Adding Kubernetes Worker Nodes 147
12.8 Draining Kubernetes Nodes For Maintenance 149
12.9 Managing `etcd` 151

13. Developing Kubernetes. 153
13.1 Compiling From Source 153
13.2 Compiling A Specific Component 154
13.3 Using A Python Client To Interact With The Kubernetes API 155
13.4 Extending The API Using Custom Resource Definitions (CRD) 156

14. Ecosystem. 161
14.1 Installing Helm, The Kubernetes Package Manager 161
14.2 Using Helm to Install Applications 162
14.3 Creating Your Own Chart To Package Your Application with Helm 163
14.4 Converting Your Docker Compose Files To Kubernetes Manifests 165
14.5 Creating A Kubernetes Cluster With `kubicorn` 166
14.6 Storing Encrypted Secrets in Version Control 171
14.7 Deploying Functions with `kubeless` 173