O'REILLY®

Programming Kubernetes

Developing Cloud Native Applications

Michael Hausenblas &
Stefan Schimanski

Table of Contents

Preface vii

1. Introduction 1

What does Programming Kubernetes Mean? 1
A Motivational Example 3
Extension Patterns 4
Controllers and Operators 5
The Controller Loop 6
Events 7
Edge Versus Level Driven Triggers 9
Changing The Objects In-Cluster Or The External World 12
Optimistic Concurrency 14
Operators 17

2. Kubernetes API basics 21

The API Server 21
The HTTP Interface of the API Server 22
API Terminology 23
Kubernetes API Versioning 27
Declarative State Management 27
Using the API from the Command Line 28
How The API Server Processes Requests 32

3. Basics of `client-go` 37

The Repositories 37
The Client Library 37
Kubernetes API Types 39
API Machinery 40

Creating and Using a Client 41
Versioning and Compatibility 43
API Versions and Compatibility Guarantees 46
Kubernetes Objects in Go 48
TypeMeta 50
ObjectMeta 16
Spec and Status 53
Client sets 54
Status Subresources 56
Listings And Deletions 56
Watches 57
Client Expansion 58
Client options 58
Informers and Caching 60
Work Queue 64
API Machinery in Depth 66
Kinds 66
Resources 66
REST Mapping 67
Scheme 68
Vendoring 70
Glide 70
Dep 71
Go Modules 72

4. Using Custom Resources. 75
Discovery Information 77
Type definitions 78
Advanced Features of Custom Resources 81
Validating Custom Resources 81
Short Names And Categories 83
Printer Columns 84
Subresources 86
A Developers View on Custom Resources 90
Dynamic Client 91
Typed Clients 92
Controller-runtime Client of Operator SDK and Kubebuilder 97

5. Automating Code Generation. 101
Why Code Generation 101
Calling the Generators 101
Controlling the generators with tags 103

Global Tags 104
Local Tags 105
`deepcopy-gen` tags 106
`runtime.Object` and `DeepCopyObject` 107
`client-gen` tags 108
`informer-gen and `lister-gen` 110
Further Material 110

6. Solutions For Writing Operators. 111
Preparation 111
Following `sample-controller` 112
Kubebuilder 120
The Operator SDK 128
Other Approaches 133
Uptake And Future Directions 134

7. Shipping Controllers And Operators. 135
Lifecycle Management and Packaging 135
Packaging: The Challenge 135
Helm 136
Kustomize 138
Other Packaging Options 140
Packaging Good Practices 141
Lifecycle Management 141
Production-Ready Deployments 142
Getting The Permissions Right 142
Automated Builds And Testing 146
Custom Controllers And Observability 146

8. Custom API Servers. 151
Use Cases For Custom API Servers 151
Example: A Pizza Restaurant 153
The Architecture: Aggregation 154
API Services 156
Inner Structure of a Custom API Server 158
Delegated Authentication and Trust 160
Delegated Authorization 161
Writing Custom API Servers 163
Options and Config Pattern and Startup Plumbing 164
The First Start 170
Internal Types and Conversion 172
Writing the API Types 175

Conversions 177
Defaulting 180
Roundtrip Testing 181
Validation 183
Registry and Strategy 185
API Installation 190
Admission 193
Deploying Custom API Servers 202
Deployment Manifests 202
Setting up RBAC 204
Running the Custom API Server Insecurely 206
Certificates and Trust 208
Sharing `etcd` 211
Summary 212

9. Advanced Custom Resources. 213
Custom Resources Versioning 213
An example: The Pizza Restaurant – again 214
Conversion Webhook Architecture 217
Conversion Webhook Implementation 221
Setting up the HTTPS Server 221
Deploying the Conversion Webhook 227
Seeing Conversion in Action 228
Admission Webhooks 231
Admission Requirements in The Restaurant Example 232
Admission Webhook Architecture 232
Registering Admission Webhooks 234
Implementation an Admission Webhook 236
Admission Webhook in Action 240
Structural Schemas and the future of CustomResourceDefinitions 241
Structural Schemas 242
Pruning versus preserving unknown fields 244
Controlling pruning 245
IntOrString and RawExtensions 246
Default Values 81
Summary 248

A. Resources. 251

Index. 255

Index

A

admission, 35, 159, 166
- chain, 194, 199
- configuration, 199
- initializers, 166
- mutating, 183, 193
- order, 194
- plugin, 35, 193, 208
- plugin initializer, 199
- register, 198
- validating, 183, 193
- webhook, 153, 159, 174, 194, 196, 205, 231

aggregated API server (see aggregation)
aggregation, 75, 151, 155, 202, 208, 214, 227
aggregator (see aggregation)
alpha version, 46
API
- aggregation, 155
- group, 24, 43
- invoke via command line, 28
- Machinery, 40
- proxy, 28
- resource, 24
- server, 24
  - HTTP interface, 22
  - kind, 23
  - processing requests, 32
- version, 24

API server, 21
apiextensions-apiserver, 76, 81, 158
APIService, 202
apps group, 43
auditing, 159, 160, 165, 196
authentication, 159, 165
- delegated, 161, 205

authorization, 161, 165
- chain, 163
- delegated, 161, 205
- SubjectAccessReview, 160, 186

B

bearer token, 161
beta version, 46
builder pattern, 43

C

CA bundle, 208
category, 84
cient-gen, 102
client
- client-gen, 90
- controller-runtime, 90
- dynamic, 90, 91
- internal, 177
- loopback, 201
- set, 42
- typed, 90

client certificate, 160
client-gen, 69, 90, 95, 97
client-go, 37
clientcmd, 41
clientset, 42, 54, 95
cloud provier, 5
cluster scoped, 56
cobra command, 169
code examples from this book, x
cohabitation, 152
component

kube-aggregator, 202
config, 166
in-cluster, 42
control plane, 21, 28
API server, 21
cloud controller manager, 5
controller manager, 5, 21
etcd, 21
scheduler, 21
controller
changing resource state, 12
definition, 5
deployment, 8
events, 7
loop, 6
optimistic concurrency, 14
ReplicaSet, 8
triggers, 9
controller-runtime, 90
conversion, 24, 48, 159, 172, 177, 190
function, 178
naming pattern, 178
webhook, 214, 218
conversion-gen, 102, 176, 177
ConversionReview, 218
core group, 52
CoreDNS, 28
CR (see custom resource)
CRD (see custom resource)
crd-gen, 83
crd-schema-gen, 83
CRUD, 35
custom API server, 151
custom resource
admission webhook, 231
conversion, 214
conversion webhook, 218
definition, 43, 75, 76, 125, 151, 176
versioning, 213
CustomResourceDefinition (see custom resource)

D
decoding, 159
deep copy, 49, 95
deepcopy-gen, 102
defaulter-gen, 102, 176, 180
defaulting, 175, 190
defaulting function naming pattern, 180
DeferredDiscoveryRESTMapper, 68
delegated authentication, 161
delegated authorization, 161
dep, 71
deserializer, 223
desired state, 27
discovery, 77, 84, 97
endpoint, 171
RESTMapper, 68, 78
dynamic client, 90

E
edge-driven trigger, 9
encoding, 159
error
conflict, 16
etcd, 163
etcd operator, 211
etcdproxy-controller, 211
external version, 174

F
feature gate, 165, 247
field selector, 56
fuzzer, 182

G
GA, 46
generator
client-gen, 69, 90, 97, 102
conversion-gen, 102, 176, 177
crd-gen, 83
crd-schema-gen, 83
deepcopy-gen, 102
default-gen, 180
defaulter-gen, 102, 176
go-to-protobuf, 152
informer-gen, 102
lister-gen, 102
generators, 125
generic registry, 186
Gengo, 101
Git, viii
glide, 70
Go (build system), viii
go modules, 72
go-to-protobuf, 152
go.mod (see go modules)

graceful termination, 170
Grafana, 148
GroupVersion, 55
GroupVersionKind, 25, 49, 66, 197
GroupVersionKinds, 177
GroupVersionResource, 25, 66, 91
GVK, 25, 49, 66, 177, 197
GVR, 25, 66, 91

H

handler chain, 159
Helm, 136
    chart, 137
HTTP/2, 152
hub version, 172

I

impersonation, 159
in-cluster config, 42
informer, 60
    factory, 201
    relist, 60
    resync, 60
informer-gen, 102
internal client, 55, 177
internal version, 172
IntOrString, 246

J

JSON, 22
    schema, 82

K

kind, 66
    Kubernetes in Docker, viii
klog, 147
kube-aggregator, 155
kube-apiserver, 155
kube-dns, 28
Kubebuilder, 83, 97, 120, 143
kubeconfig, 41
KUBECONFIG, 42
kubectl, 42, 133
Kubernetes
    releases, viii
KUDO, 133
Kustomize, 138
kutil, 133

L

label selector, 56
legacy group, 52
level-driven trigger, 9
lifecycle management, 135
lister-gen, 102
logger, 147
long-running, 59, 59

M

man-in-the-middle attack, 156
manifests
    GitHub repository, x
master node, 21
meta/v1, 41
Metacontroller, 133
metrics, 148
Minikube, viii

N

NameGenerator, 189
namespace scoped, 56

O

ObjectMeta, 43, 53, 93, 185
ObjectTyper, 189
OpenAPI, 85, 169
    schema, 81, 82
OpenShift, 152, 152
operator
    advanced, 213
    alternatives, 133
    basics, 17
    build, 146
    CI, 146
    definition, 5
    integration testing, 146
    logging, 147
    monitoring, 148
    observability, 146
    permissions, 142
    production ready, 142
    scalability, 146
    SDK, 128
    testing, 146
Operator SDK, 53, 97
optimistic concurrency, 88
option-config pattern, 164, 222

options
  recommended, 164

P

package management, viii
packaging, 135
  Ansible, 140
  awk, 140
  challenge, 135
  Chef, 140
  cloud native languages, 140
  good practices, 141
  Helm, 136
  jq, 140
  Ksonnet, 141
  Kustomize, 138
  Puppet, 140
  Salt, 140
  sed, 140
  shell scripts, 140
  YAML, 136
  ytt, 140
post start hook, 169
ProcessInfo, 166
Prometheus, 148
protobuf (see protocol buffer)
protocol buffer, 22, 43, 52, 58, 151, 176, 181, 225
prune, 245
pseudo version, 73

Q

QPS, 59

R

rate limiting, 58
RawExtension, 246
RBAC, 86, 135, 142, 161, 204 (see role based access control)
  cluster role, 205, 206
  cluster role binding), 205
  role, 205
recommended options, 164
reflection, 68
registry, 168
relist period, 60
remote procuedure call, 8
request
  long-running, 59
request header, 160
request header client CA, 160
resource, 66, 91
resource version, 16
  conflict error, 16
REST, 22, 35, 186
  client, 55
  config, 42, 54, 61, 91
  endpoint, 66
  endpoints, 66
  interface
    CollectionDeleter, 186
    Creater, 186
    CreaterUpdater, 186
    Exporter, 187
    Getter, 187
    GracefulDeleter, 187
    Lister, 187
    Patcher, 187
    Scoper, 187
    Updater, 187
    Watcher, 187
  mapper, 55
  mapping, 26, 67, 158, 159
  storage constructor, 189
  verbs, 37
RESTful, 22, 35
RESTMapper, 26, 55, 67, 91
  discovery, 68
resync period, 60
role based access control, 34, 141 (see RBAC)
Rook Operator kit, 133
roundtrip, 175, 181
  test, 181
RPC (see remote procedure call)
runtime.Object, 68

S

SAR (see subjet access review)
scale subresource, 88, 109
scheduler, 8
schema
  structural, 242
scheme, 52, 68, 91, 97, 107
selector
  field, 56, 64
  label, 56, 64
semantic versioning, 38

- semver, 46

semver, 38
server-side printing, 84
service account, 204
shared informer factory, 61
short name, 83
SIG API Machinery, 120
Simple Go client, 133
spec, 27
spec-status split, 86, 108, 127
status, 27
status subresource, 56, 108
storage version, 48, 173
Store, 60
strategy, 187, 192
structural schema, 242
structured logging, 147
subject access review, 160, 161, 186
subresource, 25, 59, 86, 235
- exec, 86
- logs, 86
- portforward, 86
- scale, 86, 88, 109
- status, 56, 86, 86, 108, 145

T

throttling, 58
- burst, 59
- queries per second, 59

timeout, 58
token, 160
- access review, 159, 161
- bearer, 161

tokenmaccess review, 161
typed client, 90
TypeMeta, 50, 93, 93, 93

U

Unstructured, 91
UserAgent, 58

V

validation, 35, 183
- client-side, 83

vendoring
- dep, 71
- glide, 70
- Go modules, 72

verb, 37
- create, 37
- delete, 38
- get, 37
- list, 37, 64
- patch, 38
- update, 37
- watch, 38, 60, 60, 64

version
- internal, 227

W

watch, 57
- event, 57

webhook, 166
- admission, 174, 194, 196
- conversion, 218, 231

WebSocket, 25, 152