



**KubeCon**



**CloudNativeCon**

---

North America 2019

---





KubeCon



CloudNativeCon

North America 2019

# Practical Way to Build Kubernetes Native Java Controller

*Zibo He, Min Jin @Ant-Financial*



# Introduction



KubeCon



CloudNativeCon

North America 2019



**Zibo He:** @adohe

zibo.hzb@antfin.com

Kubernetes maintainer, working on Ant-Financial PaaS platform. Emeritus co-lead of SIG CLI focusing on CLI, controller-runtime, multi-tenancy and secure container runtime.



**Min Jin:** @yue9944882 zuoxiu.jm@antfin.com

Kubernetes sub-project owner, working on Ant-Financial PaaS platform. Focusing on SIG API-Machinery: apiserver flow-control, CRD, OpenAPI facilities.

# Agenda



KubeCon



CloudNativeCon

North America 2019

General approaches to build controllers multilingually

Merit of developing controller in Java

Program using official Kubernetes Java library

Future enhancement for Java controller framework

Q&A

# Existing Controller Framework



## Builder-Based:

- Kube-builder/Controller-runtime/Controller-tools  
<https://github.com/kubernetes-sigs/kubebuilder>
- Operator SDK  
<https://github.com/operator-framework/operator-sdk>
- Apiserver-builder-alpha  
<https://github.com/kubernetes-sigs/apiserver-builder-alpha>

## Callback-Based:

- Meta-controller  
<https://github.com/GoogleCloudPlatform/metacontroller>

# Controller Pattern



KubeCon

CloudNativeCon

North America 2019

	Builder-Based	Callback-Based
Operator Lifecycle Management	No	Yes
Needs Deploying Extra Component	No	Yes
Scaffolding	Yes	No
Testing Solution	Yes	No
List-Watching At Client-side	Yes	No
Cache Management At Client-side	Yes	No
Multilingual	No	Yes

# Controller in Java vs Golang

- *Heap dumping*
- *Performance Diagnose*
- *Thread/Routine model*
- *Inheritance & Generics*
- *Open-source Community Resources*
- *And More..*

# Heap Dumping

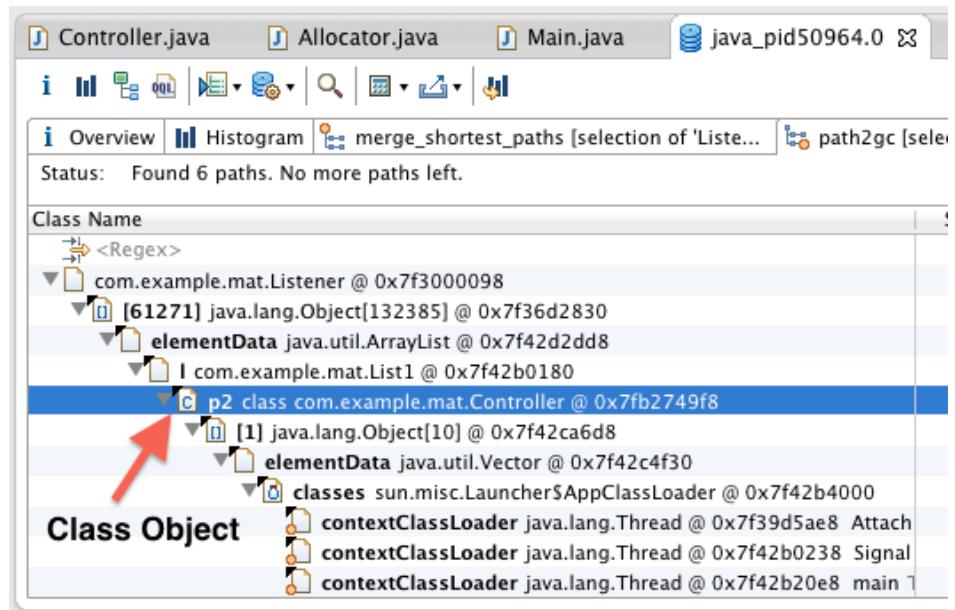


KubeCon

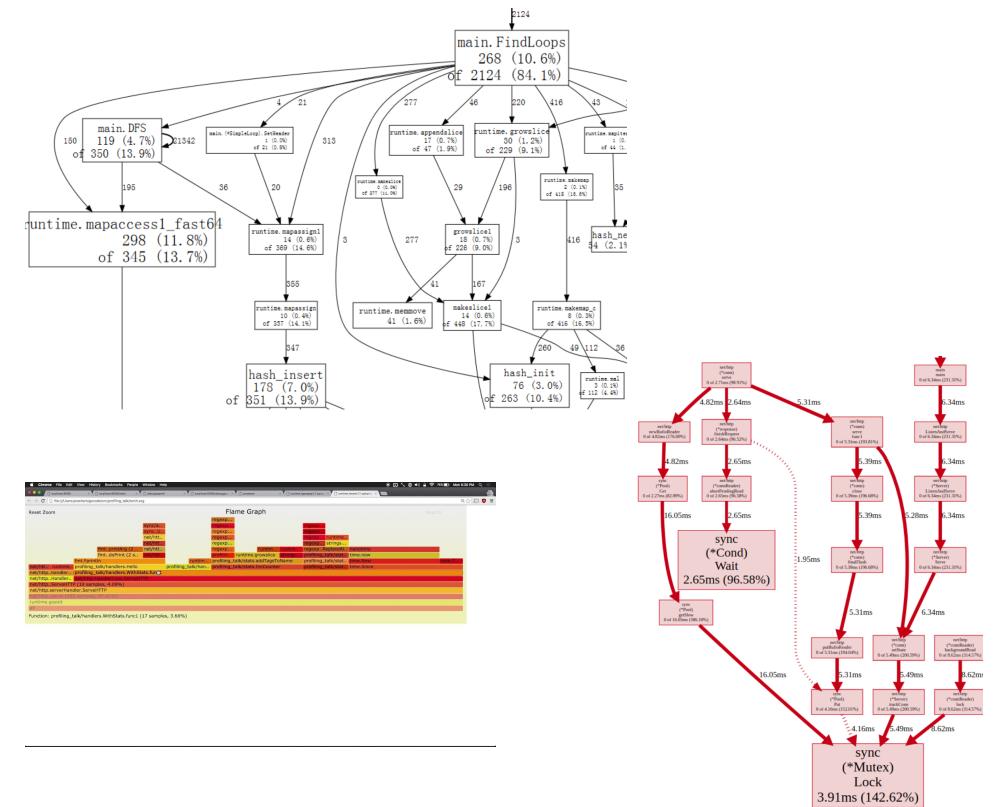
CloudNativeCon

North America 2019

(Java)



(Go)



# Performance Diagnoses



KubeCon



CloudNativeCon

North America 2019

(Java)

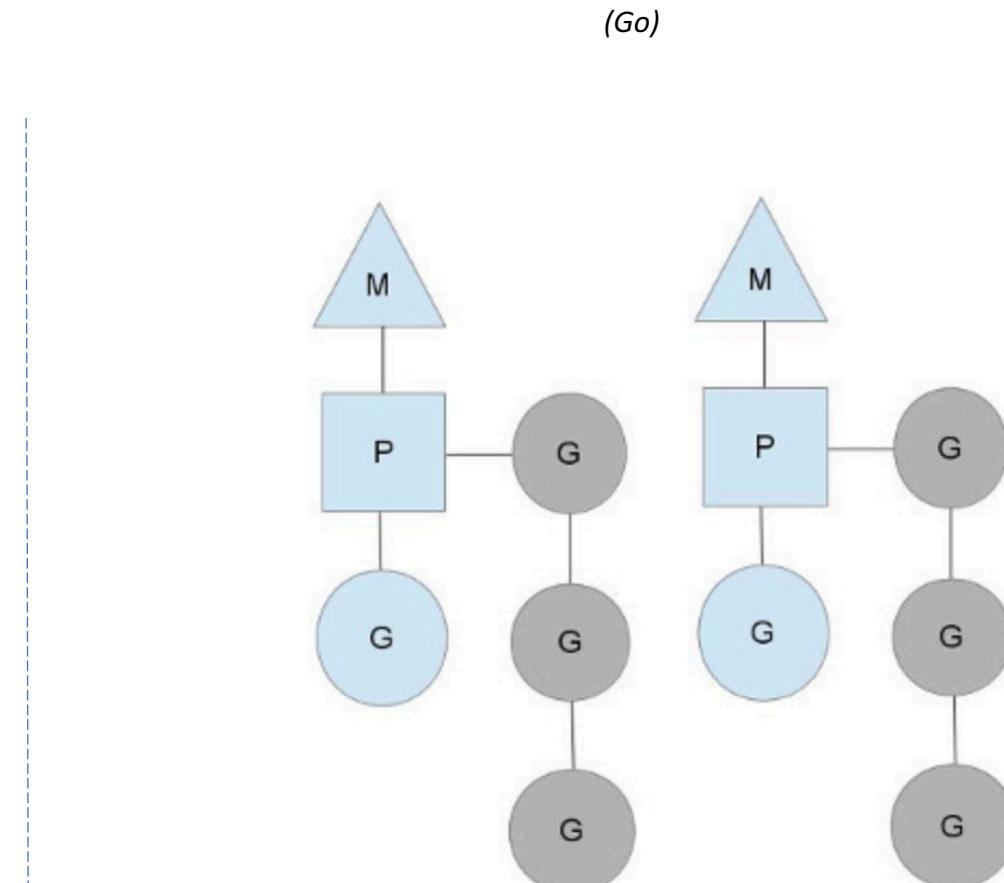
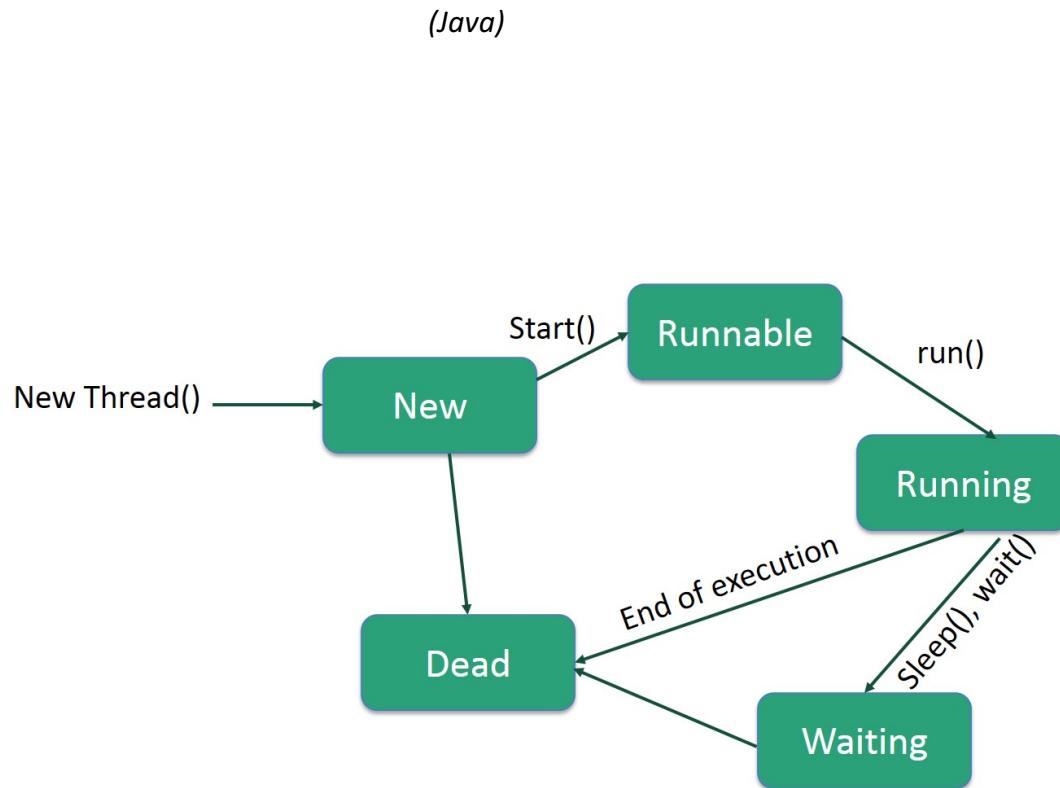
```
java_lang_GarbageCollector_CollectionCount  
java_lang_GarbageCollector_CollectionTime  
java_lang_GarbageCollector_LastGcInfo_GcThreadCount  
java_lang_GarbageCollector_LastGcInfo_duration  
java_lang_GarbageCollector_LastGcInfo_endTime  
java_lang_GarbageCollector_LastGcInfo_id  
java_lang_GarbageCollector_LastGcInfo_memoryUsageAfterGc_committ  
ed  
java_lang_GarbageCollector_LastGcInfo_memoryUsageAfterGc_init  
java_lang_GarbageCollector_LastGcInfo_memoryUsageAfterGc_max  
java_lang_GarbageCollector_LastGcInfo_memoryUsageAfterGc_used  
java_lang_GarbageCollector_LastGcInfo_memoryUsageBeforeGc_commi  
tted  
java_lang_GarbageCollector_LastGcInfo_memoryUsageBeforeGc_init  
java_lang_GarbageCollector_LastGcInfo_memoryUsageBeforeGc_max  
java_lang_GarbageCollector_LastGcInfo_memoryUsageBeforeGc_used  
java_lang_GarbageCollector_LastGcInfo_startTime  
java_lang_GarbageCollector_Valid  
java_lang_MemoryManager_Valid  
java_lang_MemoryPool_CollectionUsageThreshold  
java_lang_MemoryPool_CollectionUsageThresholdCount  
java_lang_MemoryPool_CollectionUsageThresholdExceeded  
java_lang_MemoryPool_CollectionUsageThresholdSupported  
java_lang_MemoryPool_CollectionUsage_committed  
java_lang_MemoryPool_CollectionUsage_init  
java_lang_MemoryPool_CollectionUsage_max  
java_lang_MemoryPool_CollectionUsage_used  
java_lang_MemoryPool_PeakUsage_committed  
java_lang_MemoryPool_PeakUsage_init  
java_lang_MemoryPool_PeakUsage_max  
java_lang_MemoryPool_PeakUsage_used  
java_lang_MemoryPool_UsageThreshold  
java_lang_MemoryPool_UsageThresholdCount  
java_lang_MemoryPool_UsageThresholdExceeded  
java_lang_MemoryPool_UsageThresholdSupported  
java_lang_MemoryPool_Usage_committed  
java_lang_MemoryPool_Usage_init  
java_lang_MemoryPool_Usage_max  
java_lang_MemoryPool_Usage_used
```

```
java_lang_MemoryPool_Valid  
jmxremote_jmx_connector_server_Active  
jvm_gc_collection_seconds_count  
jvm_gc_collection_seconds_sum  
jvm_info  
jvm_memory_bytes_committed  
jvm_memory_bytes_max  
jvm_memory_bytes_used  
jvm_memory_pool_bytes_committed  
jvm_memory_pool_bytes_max  
jvm_memory_pool_bytes_used
```

(Go)

```
go_gc_duration_seconds  
go_info  
go_goroutines  
go_memstats_alloc_bytes  
go_memstats_alloc_bytes_total  
go_memstats_buck_hash_sys_bytes  
go_memstats_frees_total  
go_memstats_gc_cpu_fraction  
go_memstats_gc_sys_bytes  
go_memstats_heap_alloc_bytes  
go_memstats_heap_idle_bytes  
go_memstats_heap_inuse_bytes  
go_memstats_heap_objects  
go_memstats_heap_released_bytes  
go_memstats_heap_sys_bytes  
go_memstats_last_gc_time_seconds  
go_memstats_lookups_total  
go_memstats_mallocs_total  
go_memstats_mcache_inuse_bytes  
go_memstats_mcache_sys_bytes  
go_memstats_mspan_inuse_bytes  
go_memstats_mspan_sys_bytes  
go_memstats_next_gc_bytes  
go_memstats_other_sys_bytes  
go_memstats_stack_inuse_bytes  
go_memstats_stack_sys_bytes  
go_memstats_sys_bytes
```

# Thread Model

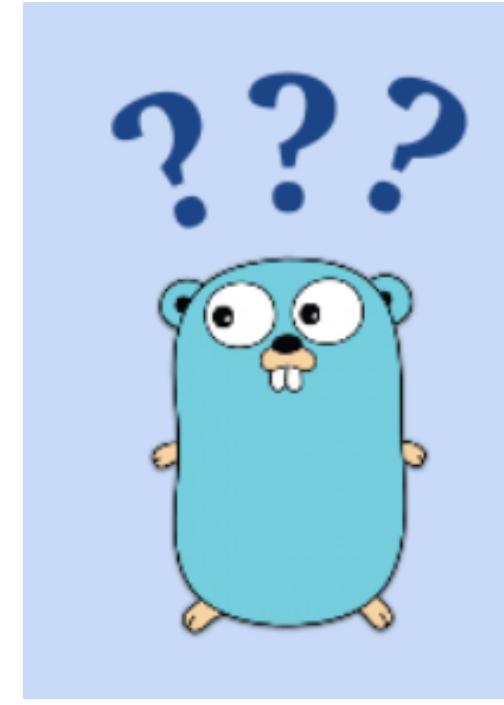


# Inheritance & Generics

(Java)



(Go)



# Open-Source Resource



KubeCon



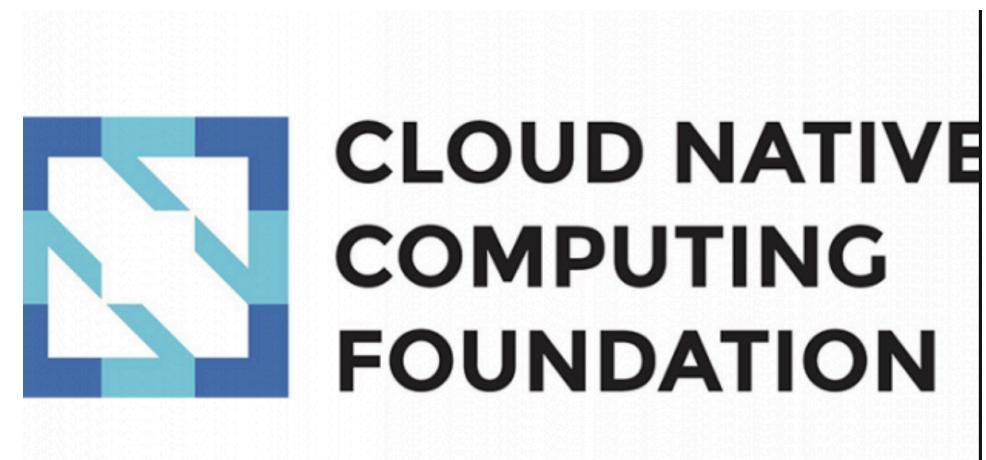
CloudNativeCon

North America 2019

(Java)



(Go)



# Java Controller SDK



## We Are Providing:

- *List-Watcher/Informer Library*
- *Java “Controller-Runtime” Library*
- *Leader Election/Lock*
- *CRD Java Model Code-Generation*
- *And More..*

# Java Controller SDK



## Demo

Check out more examples at: <https://github.com/kubernetes-client/java>

# Java Controller SDK

<https://github.com/kubernetes-client/java/blob/master/examples/src/main/java/io/kubernetes/client/examples/ControllerExample.java>

```
// Use builder library to construct a default controller.  
Controller controller =  
    ControllerBuilder.defaultBuilder(informerFactory)  
        .watch(  
            (workQueue) ->  
                ControllerBuilder.controllerWatchBuilder(V1Node.class, workQueue)  
                    .withWorkQueueKeyFunc(  
                        (V1Node node) ->  
                            new Request(node.getMetadata().getName()) // optional, default to  
                    .withOnAddFilter(  
                        (V1Node createdNode) ->  
                            createdNode  
                                .getMetadata()  
                                .getName()  
                                .startsWith("docker-") // optional, set onAdd filter  
                    .withOnUpdateFilter(  
                        (V1Node oldNode, V1Node newNode) ->  
                            newNode  
                                .getMetadata()  
                                .getName()  
                                .startsWith("docker-") // optional, set onUpdate filter  
                    .withonDeleteFilter(  
                        (V1Node deletedNode, Boolean stateUnknown) ->  
                            deletedNode  
                                .getMetadata()  
                                .getName()  
                                .startsWith("docker-") // optional, set onDelete filter  
            .build()  
        .withReconciler(nodeReconciler) // required, set the actual reconciler  
        .withName("node-printing-controller") // optional, set name for controller  
        .withWorkerCount(4) // optional, set worker thread count  
        .withReadyFunc(  
            informer  
                ::hasSynced) // optional, only starts controller when the cache has synced up  
    .build();
```

```
static class NodePrintingReconciler implements Reconciler {  
  
    private Lister<V1Node> nodeLister;  
  
    public NodePrintingReconciler(SharedIndexInformer<V1Node> nodeInformer) {  
        this nodeLister = new Lister<>(nodeInformer.getIndexer());  
    }  
  
    @Override  
    public Result reconcile(Request request) {  
        V1Node node = this nodeLister.get(request.getName());  
        System.out.println("triggered reconciling " + node.getMetadata().getName());  
        return new Result(false);  
    }  
}
```

# Lessons

- Manage CRD Yaml Definition **CAREFULLY**
- Traps in operator's graceful shutdown
- Decouple your operators as a standalone component
- Understand your operator model better before developing
- And ...



# Future Enhancement



KubeCon



CloudNativeCon

North America 2019

- *Kubernetes OpenAPI v3 schema publishing* (<https://github.com/kubernetes/enhancements/pull/1263>)
  - Multi-Returning
  - Unwound “int-or-string”
  - Per-group publishing
  - ...

# Future Enhancement



- *Advanced Extensions for Large-scale Cluster Controller*
- *Controller Canary Release*
- *Multi-Cluster Extensions*

# Q&A

