GraceFulShotDown

SpringBoot+KAFKA ( )

Linux Signals/Spring Boot/Docker 3 .

Signals

UNIX Signal Event . / Signal Signal .

```
[root@peterdev dev]# kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL       5) SIGTRAP
 6) SIGABRT      7) SIGBUS       8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
```

Application

- : **SIGTERM**
- : **SIGKILL**

, .

Spring Boot

```
server:
  shutdown: graceful
spring:
  lifecycle:
    timeout-per-shutdown-phase: 20s
```

Spring Boot 2.3 .

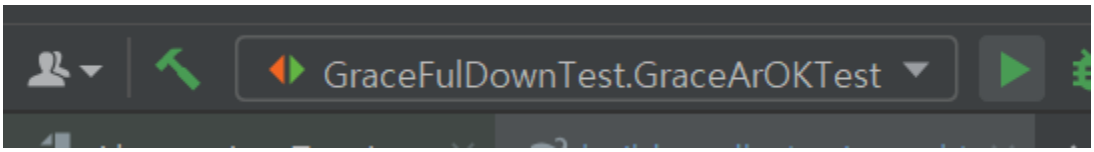
Spring Boot JVM JVM .

JVM .

-
- System.exit
-

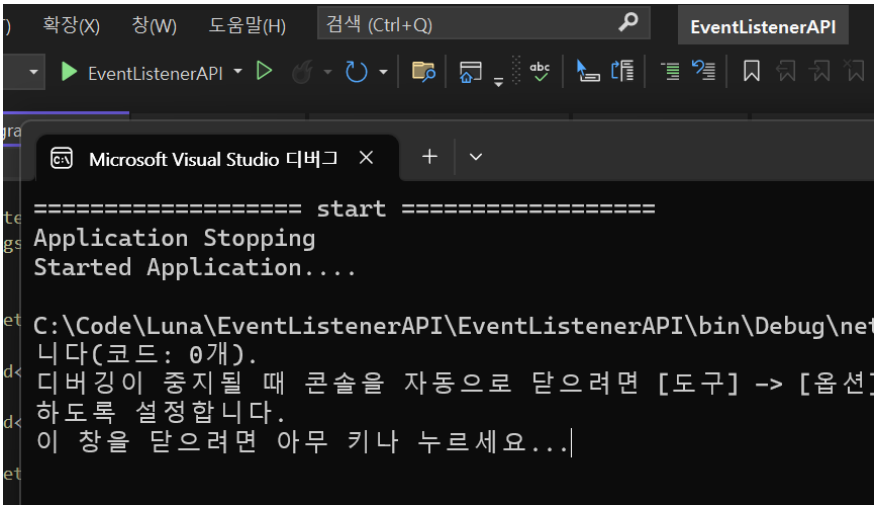
IDE

InteliJ



> *InteliJ () .*
>
> ~

VisualStudio




```
app.Lifetime
   .ApplicationStopping
   .Register(() => Console.WriteLine("Application Stopping"));
```

> *AKKA .NET VisualStudio IDE .*
> *VS InteliJ .*
>
> *ctrl+c .*

Docker

docker

- docker stop
- docker kill

- sotp (SIGTERM)
- kill (SIGKILL)
- .

default 10s

spring boot timeout-per-shutdown-phase .

10s .

```
docker stop -t 60 somAPP
```

.

docker 10s DockerCompose .

- stop_grace_period : DockerCompose
- `terminationGracePeriodSeconds` :

⚠ (.sh) .

.jar .

```
#
CMD ["java", "-jar", "app.jar"]
```

```
# run.sh    java -jar app.jar
#   .     sh
RUN chmod +x run.sh
```

GraceFulShotDown

DI .

. / .

Blue Green

Blue/Green Green Blue .

GraceFulShotDown . .

, .

Kafka GracefulDown

shutdown hook KafkaConsumer.wakeup() offset .

.

```
public static void main(String[] args) {
        Runtime.getRuntime().addShutdownHook(new ShutdonwThread());
}

static class ShutdownThread extends Thread {
        public void run() {
            consumer.wakeup();
    }
}
```

? .

.

- Kafka Message Delivery Semantics - Exactly Once ?
- Message order and delivery guarantees in Elixir/Erlang
    - Kafka .

Coordinated shutdown

GracefulDown

.

: gracefulDown

```
    @BeforeClass
    public static void bootUp() {
        actorSystem = serverStart("ClusterSystem", "router-test", "seed");
        logger.info("========= sever loaded =========");
        appActor = actorSystem.actorOf(WorkStatusActor.Props(), "APPActor");
    }

    @AfterClass
    public static void bootDown() {

        logger.info("========= try graceful down =========");
        int retryCount = 5;
        for (int i = 0; i < retryCount; i++) {
            CoordinatedShutdown.get(actorSystem).addTask(
                    CoordinatedShutdown.PhaseBeforeServiceUnbind(), "WorkCheckTask",
                    () -> {
                        return akka.pattern.Patterns.ask(appActor, "stop", Duration.ofSeconds(1))
                                .thenApply(reply -> Done.getInstance());
                    });
        }
    }
```

Kafka GraceFulDown .

AKKA Stack HTTP GraceFul Coordinated .

- https://doc.akka.io/docs/akka/current/coordinated-shutdown.html
- https://doc.akka.io/docs/akka-http/current/server-side/graceful-termination.html

,,

.

- GraceFulShoutDown : , .
- CoordinatedShoutDown : , .

- [Linux] kill ''
- JVM Graceful Shutdown
- graceful shutdown
- kubernetes
- [Docker] Graceful Shutdown in Docker
- blocked URL graceful shutdown ?
- blocked URLCoordinated Shutdown • Akka Documentation
- blocked URLCoordinated Shutdown | Akka.NET Documentation