DOIS | 2018 · 深圳站

DevOps 落地，从这里开始

DevOps 国际峰会

暨 DevOps 金融峰会

指导单位：云计算开源产业联盟 OpenSource Cloud Alliance for Industry, OSCAR

主办单位：DevOps时代　高效运维社区 GreatOPS Community

时间：2018年11月2日-3日

地址：深圳市南山区圣淘沙大酒店（翡翠店）

平安证券
PING AN SECURITIES

DOIS

券商DevOps转型—平安证券容器化实践之路

陈刚 中国平安证券运维开发经理

目录

金融行业IT实践和其它行业的异同

1. 相同之处
   - 争取各部门领导同事配合
   - 各种技术和流程的坑要去填
2. 差异之处
   - 金融合规,异构系统
   - 部门之间异地沟通

平安证券
PING AN SECURITIES

DOIS

3.全面风险管理		效履行风险管理职责
	3.03	风险管理制度有效执行，风险管理考核纳入员工绩效考核，风险管理文化建设融入经营管理全过程
		能
	3.05	压力测试机制健全有效并能按要求报送压力测试报告，净资本补足机制和业务规模调整机制健全并能有效实施
4.信息系统安全	4.01	IT 治理完善，信息系统管理机制独立有效
	4.02	信息系统功能齐备，有效满足客户委托、交易、清算、开户、查询等需求，客户电子资料等信息安全
	4.03	信息系统安全稳定运行，能够避免频繁信息安全事故或重大事故
	4.04	信息系统应急预案有效，能够及时应对信息安全事故
5.客户权益保护	5.01	客户资产存放管理制度完善，能够有效保障客户资产安全
	5.02	投资者适当性制度和客户服务、客户管理制度健全，能够将适当的产品或服务销售或提供给适当的投资者
	5.03	营销人员管理制度健全，有效防止营销人员损害客户权益。投行、资管等业务勤勉尽责、诚实守信，从源头保障上市公司质量，切实维护客户合法权益
	5.04	客户投诉处理机制有效，能够稳妥处理各类上访、投诉、

《证券公司分类监管规定》

证监部门处罚部分信息系统“瘫痪”券商

发布：[redacted] | 分类：行业新闻

证监会回应个别券商发生系统中断或运行缓慢问题

015-06-05 16:48:43 来源: [redacted] 举报

四家券商系统故障 证监局出具警示函

2015年06月05日 16:45 来源于 [redacted]

[快讯]证监会:系统故障导致损失 投资者可索赔

正文 我来说两句(0人参与) 扫描到手机

目录

平安证券
PING AN SECURITIES

DOIS

系统虚拟化： NEW OR OLD？

Mechanism	Operating system	License	Available since or between	Features									
				File system isolation	Copy on Write	Disk quotas	I/O rate limiting	Memory limits	CPU quotas	Network isolation	Nested virtualization	Partition checkpointing and live migration	Root privilege isolation
chroot	Most UNIX-like operating systems	Varies by operating system	1982	Partial[a]	No	No	No	No	No	No	Yes	No	No
Docker	Linux,[6] FreeBSD,[7] Windows x64 (Pro, Enterprise and Education)[8] macOS [9]	Apache License 2.0	2013	Yes	Yes	Not directly	Yes (since 1.10)	Yes	Yes	Yes	Yes	Only in Experimental Mode with https://criu.org/Docker	Yes (since 1.10)
Linux-VServer (security context)	Linux, Windows Server 2016	GNU GPLv2	2001	Yes	Yes	Yes	Yes[b]	Yes	Yes	Partial[c]	?	No	Partial[d]
lmctfy	Linux	Apache	2013	Yes	Yes	Yes	Yes[b]	Yes	Yes	Partial[c]	?	No	Partial[d]
LXC	Linux										es	No	Yes[11]
Singularity	Linux										o	No	Yes[14]
OpenVZ	Linux										ial[i]	Yes	Yes[j]
Virtuozzo	Linux, Windows										ial[l]	Yes	Yes
Solaris Containers (Zones)	illumos (OpenSolaris), Solaris	CDDL, Proprietary	2004	Yes	Yes (ZFS)	Yes	Partial[m]	Yes	Yes	Yes[n][21][22]	Partial[o]	Partial[p][q]	Yes[r]
FreeBSD jail	FreeBSD, DragonFly BSD	BSD License	2000[24]	Yes	Yes (ZFS)	Yes[s]	Yes	Yes[25]	Yes	Yes[26]	Yes	Partial[27][28]	Yes[29]
sysjail	OpenBSD, NetBSD	BSD License	2006–2009	Yes	No	No	No	No	No	Yes	No	No	?
WPARs	AIX	Commercial proprietary software	2007	Yes	No	Yes	Yes	Yes	Yes	Yes[t]	No	Yes[31]	?
iCore Virtual Accounts	Windows XP	Freeware	2008	Yes	No	Yes	No	No	No	No	?	No	?
Sandboxie	Windows	Trialware	2004	Yes	Yes	Partial	No	No	No	Partial	No	No	Yes
Systemd-nspawn	Linux	GNU LGPLv2.1+	2010	Yes	Yes	No	No	No	No	Yes	No	No	Yes
Turbo	Windows	Freemium	2012	Yes	No	No	No	No	No	Yes	No	No	Yes
RKT	https://github.com/rkt/rkt	Free	?	?	?	?	?	?	?	?	?	?	?

平安证券
PING AN SECURITIES

DOIS

LXC,Libvirt时代（Docker 0.6）

GATHERING LXC AND DOCKER CONTAINERS METRICS

By Jérôme Petazzoni October 8, 2013

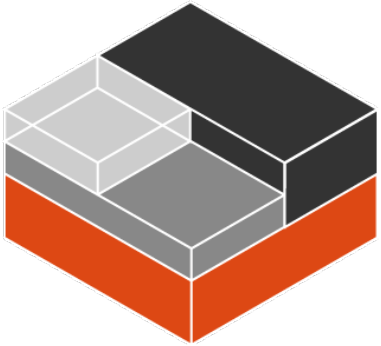
cgroups, containers, docker, lxc, metrics

- **Collaboration on container provisioning: Red Hat has** enabled libvirt, the open source virtualization API project, as an option for creating containers within Docker. This approach will enable users to take full advantage of the robust networking capabilities of libvirt while maintaining the user experience of Docker provisioning.

RED HAT AND DOCKER COLLABORATE

By Ben Golub September 19, 2013

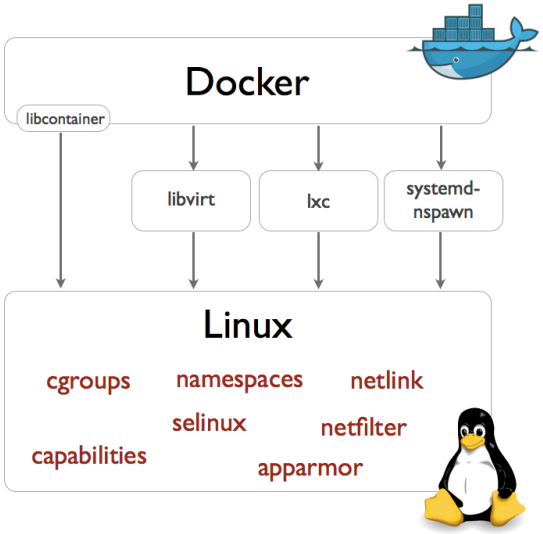
The current implementation of Docker (as of 0.6) makes this particularly challenging, because it relies on `lxc-start`, and when a container stops, `lxc-start` carefully cleans up behind it. If you really want to collect the metrics anyway, here is



CANONICAL

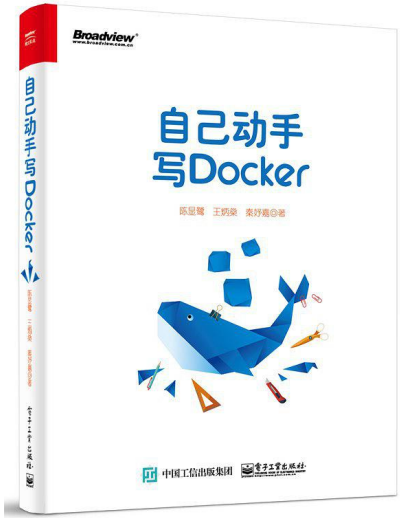
libcontainer时代（Docker 0.9）

- 功能实现上涵盖了包括namespaces使用、cgroups管理、Rootfs的配置启动、默认的Linux capability权限集、以及进程运行的环境变量配置。
- 内核版本最低要求为2.6，最好是3.8，这与内核对namespace的支持有关。
- 除user namespace不完全支持以外，其他五个namespace都是默认开启的，通过clone系统调用进行创建。



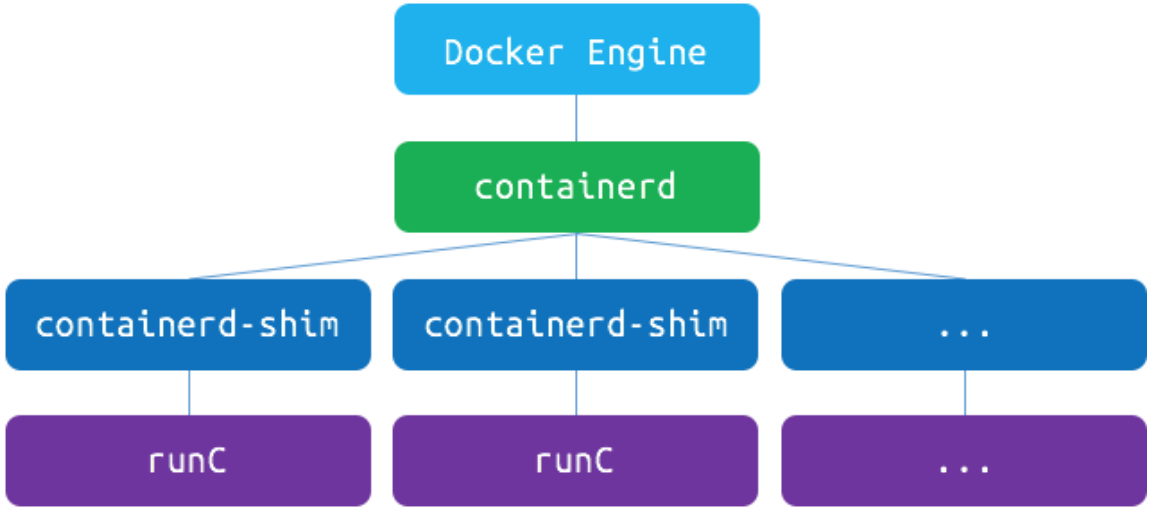
UTS 隔离nodename和hostname

```
[root@centos7.3.1611 mydocker]# cat main_newuts.go
package main

import (
    "os"
    "log"
    "os/exec"
    "syscall"
)

func main() {
    cmd := exec.Command("sh")
    cmd.SysProcAttr = &syscall.SysProcAttr{Cloneflags: syscall.CLONE_NEWUTS}

    cmd.Stdin = os.Stdin
    cmd.Stdout = os.Stdout
    cmd.Stderr = os.Stderr

    if err := cmd.Run(); err!= nil {
        log.Fatal(err)
    }

    os.Exit(-1)
}
```



Containerd + runc时代（Docker 1.11）

containerd,container-shim 组件本质上是runC 和dockerd 间的adapter中间件



Containerd + runc时代（Docker 1.11）

runc/libcontainer/nsenter/

```
/* All of these are taken from include/uapi/linux/sched.h */
#ifndef CLONE_NEWNS
#       define CLONE_NEWNS 0x00020000 /* New mount namespace group */
#endif
#ifndef CLONE_NEWCGROUP
#       define CLONE_NEWCGROUP 0x02000000 /* New cgroup namespace */
#endif
#ifndef CLONE_NEWUTS
#       define CLONE_NEWUTS 0x04000000 /* New utsname namespace */
#endif
#ifndef CLONE_NEWIPC
#       define CLONE_NEWIPC 0x08000000 /* New ipc namespace */
#endif
#ifndef CLONE_NEWUSER
#       define CLONE_NEWUSER 0x10000000 /* New user namespace */
#endif
#ifndef CLONE_NEWPID
#       define CLONE_NEWPID 0x20000000 /* New pid namespace */
#endif
#ifndef CLONE_NEWNET
#       define CLONE_NEWNET 0x40000000 /* New network namespace */
#endif
```

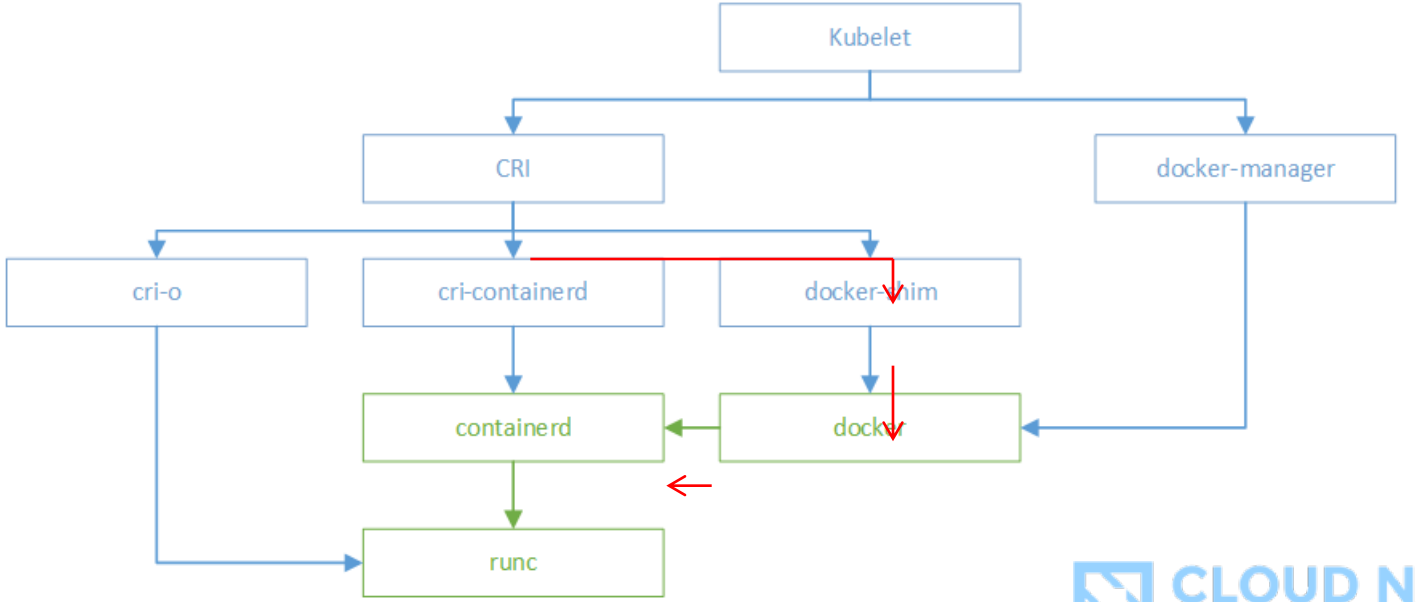
Branch: master ▾ runc / libcontainer / nsenter / nsenter.go

crosbymichael Move libcontainer into subdirectory

1 contributor

13 lines (10 sloc) | 159 Bytes

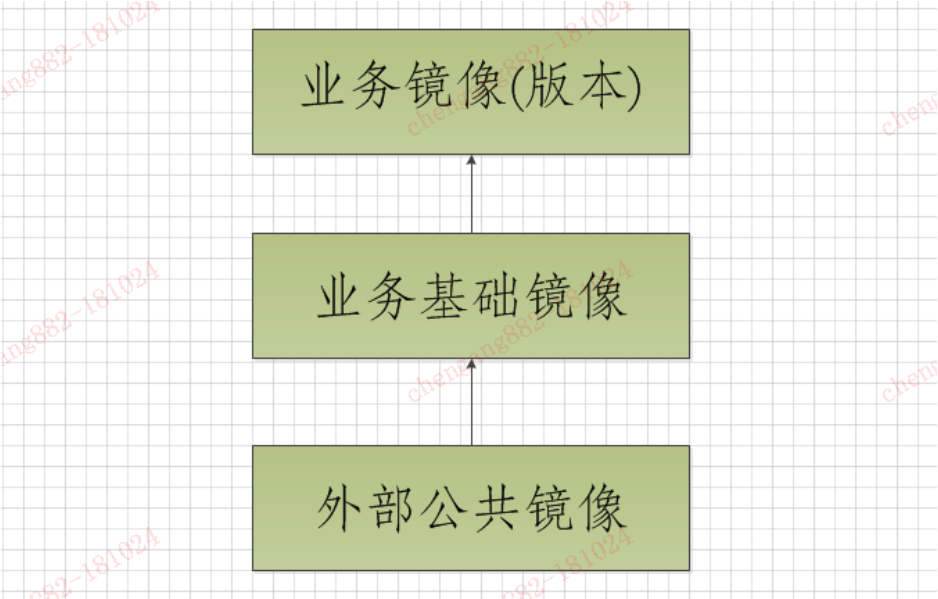
```
// +build linux,!gccgo

package nsenter

/*
#cgo CFLAGS: -Wall
extern void nsexec();
void __attribute__((constructor)) init(void) {
        nsexec();
}
*/
import "C"
```

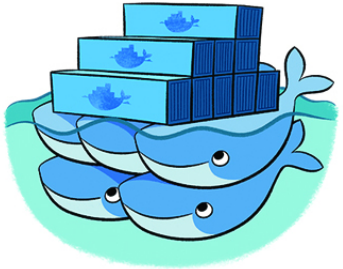



```
[docker@stan140155 ~]$ pstree -l -a -A 1246
dockerd --graph=/docker --storage-driver=overlay --insecure-registry harbor.paic.com.cn
  |-docker-containe --config /var/run/docker/containerd/containerd.toml
  |   |-docker-containe -namespace moby -workdir /docker/containerd/daemon/io.containerd.runtime.v1.linux/moby/877cf43d3ef4c2f1b94b2cebc8
19ae91ffb825b48677226efd10a54dda4c408d -address /var/run/docker/containerd/docker-containerd.sock -containerd-binary /usr/bin/docker-cont
ainerd -runtime-root /var/run/docker/runtime-runc
  |   |   |-pause
  |   |   `-6*[{docker-containe}]
  |   |-docker-containe -namespace moby -workdir /docker/containerd/daemon/io.containerd.runtime.v1.linux/moby/8b0ceaeb9c4e0dfb4b7b6292a7
12188227a6915c4a52dabf7652dc4bee330ae9 -address /var/run/docker/containerd/docker-containerd.sock -containerd-binary /usr/bin/docker-cont
ainerd -runtime-root /var/run/docker/runtime-runc
  |   |   |-flower /usr/local/bin/flower --port=5555 --broker=redis://prism-redis:6379/0
  |   |   |   `-5*[{flower}]
  |   |   `-6*[{docker-containe}]
```

目录

Docker分层镜像原则(兼顾规范与灵活)





最小化镜像---完全自制？完全外部？

8.5.33-jre10-slim	126 MB	4 days ago
jre10	280 MB	4 days ago
8-jre10	280 MB	4 days ago
8.5-jre10	280 MB	4 days ago
8.5.33-jre10	280 MB	4 days ago
alpine	71 MB	4 days ago
8-alpine	71 MB	4 days ago
8.5-alpine	71 MB	4 days ago

最小化镜像---完全自制？完全外部？

```
#!/usr/bin/env bash
FROM frolvlad/alpine-glibc:alpine-3.8_glibc-2.28
MAINTAINER frolvlad/alpine-glibc:alpine-3.8_glibc-2.28
ENV TIME_ZONE Asia/Shanghai
RUN apk add --no-cache tzdata \
        &&apk add  --no-cache bash \
        &&apk add  --no-cache curl\
        &&apk add  --no-cache busybox-extras\
       &&echo "${TIME_ZONE}" > /etc/timezone \
       &&ln -sf /usr/share/zoneinfo/${TIME_ZONE} /etc/localtime
```

The GNU C Library (glibc)

alpine Linux

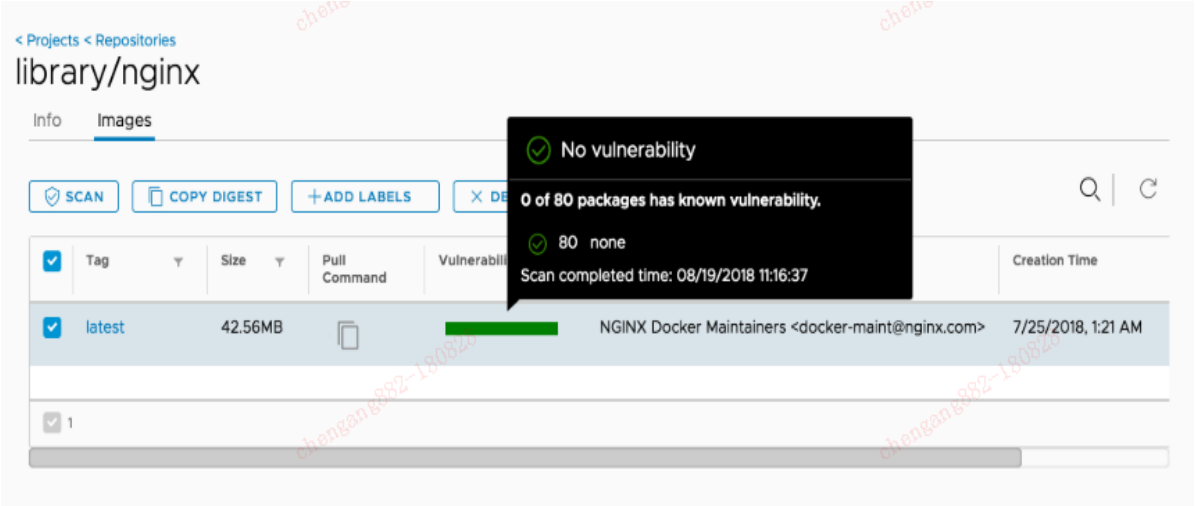
musl libc

```
From harbor/3rd_part/alpine:alpine-3.8_glibc-2.28
MAINTAINER  Prism
ENV JAVA_VERSION=jdk1.x.x
ENV  TOMCAT_VERSION=tomcat-x.x.x
ENV  CATALINA_HOME=/usr/local/${TOMCAT_VERSION}
…
RUN  mkdir -p /temp/ \
      && mkdir -p ${APP_HOME} \
      ……
      && adduser  -u 9999 -D -S docker_user -G docker_group \
      ……
       && echo "finished!!!!"
COPY --chown= docker_user docker_group ${JAVA_VERSION}  ${JAVA_HOME}
COPY --chown= docker_user docker_group ${TOMCAT_VERSION}  ${CATALINA_HOME}

WORKDIR ${APP_HOME}
USER docker_user
#
# 要是内核新点，能支持user namespace，就可以将root里的用户映射为宿主机的普通用户了。
# 而不用费尽心机的将docker的用户id，组id与宿主机里的uid,gid进行匹配。
```

为什么一个JDK基础镜像会有300M以上？

IMAGE	CREATED	CREATED BY	SIZE
85cd6c7243a8	12 days ago	/bin/sh -c #(nop) [redacted]	0B
39c6859c50a2	12 days ago	/bin/sh -c #(nop) [redacted]	0B
74ba7e68a572	12 days ago	/bin/sh -c #(nop) [redacted]	0B
d6bb884c108e	12 days ago	/bin/sh -c #(nop) COPY --chown=[redacted]	8.37MB
2e0a18c72c8c	12 days ago	/bin/sh -c #(nop) COPY --chown=[redacted]	159MB
f531cfcbd996	12 days ago	/bin/sh -c mkdir -p [redacted] && …	811kB
ed87716acf00	12 days ago	/bin/sh -c #(nop) COPY dir:e9a8dd3aace343039…	8.37MB
b2976a17db42	12 days ago	/bin/sh -c #(nop) COPY dir:ffc4c46255d91ccc9…	159MB
288d13068cbb	12 days ago	/bin/sh -c #(nop) ENV [redacted]	0B
0ce55a4f81a1	12 days ago	/bin/sh -c #(nop) ENV [redacted]	0B
1839bfad6438	12 days ago	/bin/sh -c #(nop) ENV [redacted]	0B
0a6d042a2a22	12 days ago	/bin/sh -c #(nop) ENV [redacted]	0B
dc545ffcc9b6	12 days ago	/bin/sh -c #(nop) ENV PATH=.:/usr/local/jdk…	0B

如何确保外部镜像的安全？



clair

HARBOR™

目录

平安证券
PING AN SECURITIES

DOIS

Docker镜像自助生成

K8S发布单列表

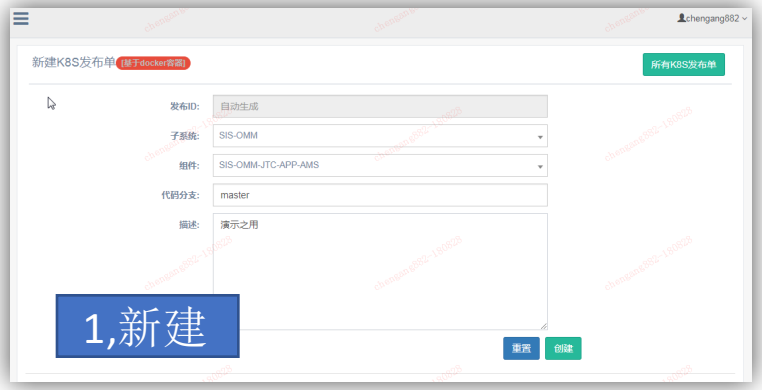
新建K8S发布单

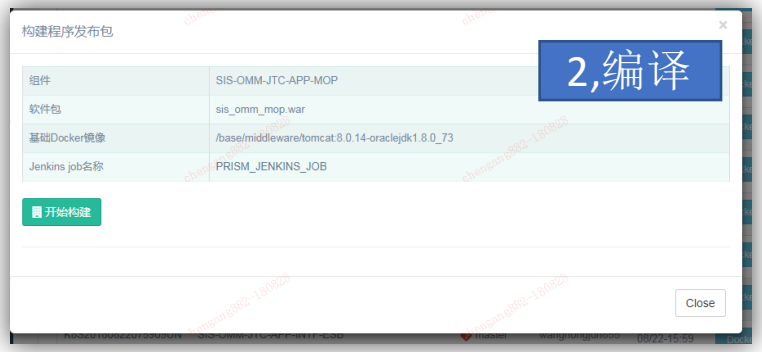
选择项目 | 选择组件 | 过滤 | 搜索

发布单编号	组件	编译分支	用户	最近修改时间	Build	操作
K8S20180918055120CK	SIS-[illegible]WEB	master	[illegible]	09/18-13:51	Docker Done	检测
K8S20180918055103QJ	SIS-[illegible]ADMIN	master	[illegible]	09/18-13:51	Docker Done	检测
K8S20180918055036QU	SIS-[illegible]	master	[illegible]	09/18-13:50	Docker Done	检测
K8S20180918055023WI	SIS-[illegible]STATIC	master	[illegible]	09/18-13:50	Docker Done	检测
K8S20180918055009DA	SIS-[illegible]S	master	[illegible]	09/18-13:50	Docker Done	检测
K8S20180918054916PU	SIS-[illegible]OXY-GW	master	[illegible]	09/18-13:49	Docker Done	检测
K8S20180918054900GX	SIS-[illegible]OP	master	[illegible]	09/18-13:49	Docker Done	检测

平安证券
PING AN SECURITIES

DOIS





3,部署

APP组件名称	SIS-[illegible]-GW	组件中文名称	[illegible]组件
所属项目	S[illegible]MM	服务名称	SIS-[illegible]-GW
GitLab	http://gitb-[illegible]omm_war.git	描述	理财网关组件
Jenkins任务名称	PRISM_JENKINS_JOB	Jenkins运行节点	JNLP_ant_1.10.3-openjdk-8-alpine
编译命令目录	./	编译命令	[illegible] -Ddisable-quartz-redisqueueconsumer=true
软件目录	target/sis_omm_proxy_gw/	软件包	sis_omm_proxy_gw.war
Dockerfile	dockerfile-jdk8-ant	基础镜像	/base/middleware/tomcat:8.[illegible]_73
部署模板文件	prism-jdk8-deployment.yaml	服务模板文件	prism-jdk8-service.yaml
BootStart文件	bootstart-jdk8.sh	实例数量	FAT 1 UAT 2 PRD 4 DRP 2
FAT CPU(核)	下限：2 上限：4	FAT内存(G)	下限：2 上限：4
UAT CPU(核)	下限：4 上限：4	UAT内存(G)	下限：8 上限：8
PRD CPU(核)	下限：4 上限：4	PRD内存(G)	下限：8 上限：8
DRP CPU(核)	下限：2 上限：4	DRP内存(G)	下限：2 上限：4

平安证券
PING AN SECURITIES

DOIS

Jenkins 2.x PipeLine

```
stages {
    stage('Prepare Git Code') {
        steps {
            echo "${SITE}"

            sh "rm -rf ${WORKSPACE}/*"
            git branch: '${BRANCH_TO_BUILD}', credentialsId: 'GitLab', url: '${GIT_URL}'
            echo 'Preparing end..'
        }
    }
    stage("Build") {
        steps {
            dir("${WORKSPACE}/${DIR_BUILD_FILE}") {
                sh "${BUILD_CMD}"
            }
        }
    }
    stage("Get Dockerfile & Bootstart.sh") {
        steps {
            dir("${WORKSPACE}/${DIR_BUILD_FILE}") {
                writeFile encoding: 'utf-8', file: 'Dockerfile', text: "${DOCKERFILE_TEMPLATE}"
                writeFile encoding: 'utf-8', file: 'bootstart.sh', text: "${BOOTSTART_TEMPLATE}"
            }
        }
    }
    stage("DockerBuild") {
        steps {
            echo 'DockerBuild..'
            dir("${WORKSPACE}/${DIR_BUILD_FILE}") {
                sh "cp ${BUILD_TARGET}/${PACKAGE_NAME} ${PACKAGE_NAME}"
                sh "docker build -t ${HARBOR_ADDRESS}/${SITE}/${APP}:${DEPLOY_VERSION} ."
            }
            echo 'DockerBuild end..'
        }
    }
    stage("DockerPush") {
```

GitLab

平安证券
PING AN SECURITIES

DOIS

Jenkins 2.x （Docker out Docker）

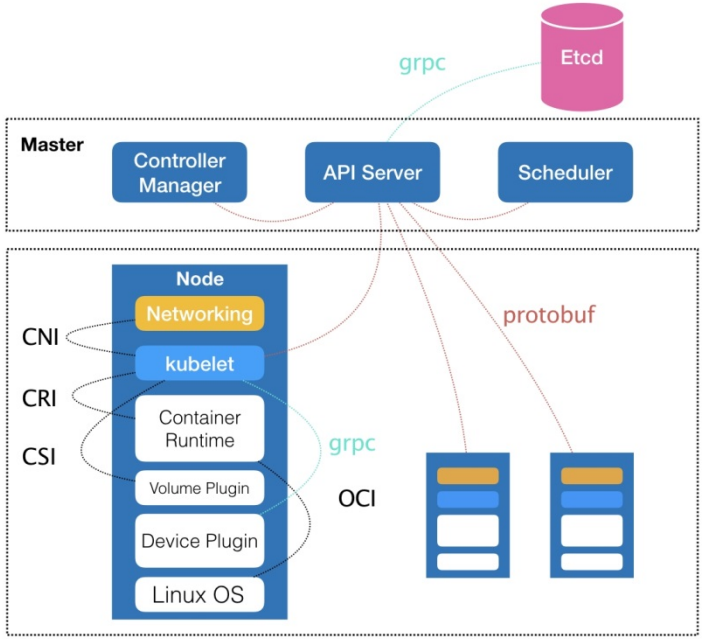
Stage View

	Prepare Git Code	Build	Get Dockerfile & Bootstart.sh	DockerBuild	DockerPush
Average stage times: (Average full run time: ~1min 19s)	15s	37s	53ms	4s	3s
#277 Oct 25 08:37 No Changes	21s	29s	48ms	4s	4s
#276 Oct 23 16:15 No Changes	19s	59s	61ms	3s	3s
#275 Oct 23 15:20 No Changes	17s	37s	61ms	4s	3s
#274 Oct 23 14:31 3 commits	22s	35s	46ms	3s	3s
#273 Oct 22 14:16 4 commits	21s	1min 26s	48ms	3s	3s

GitLab

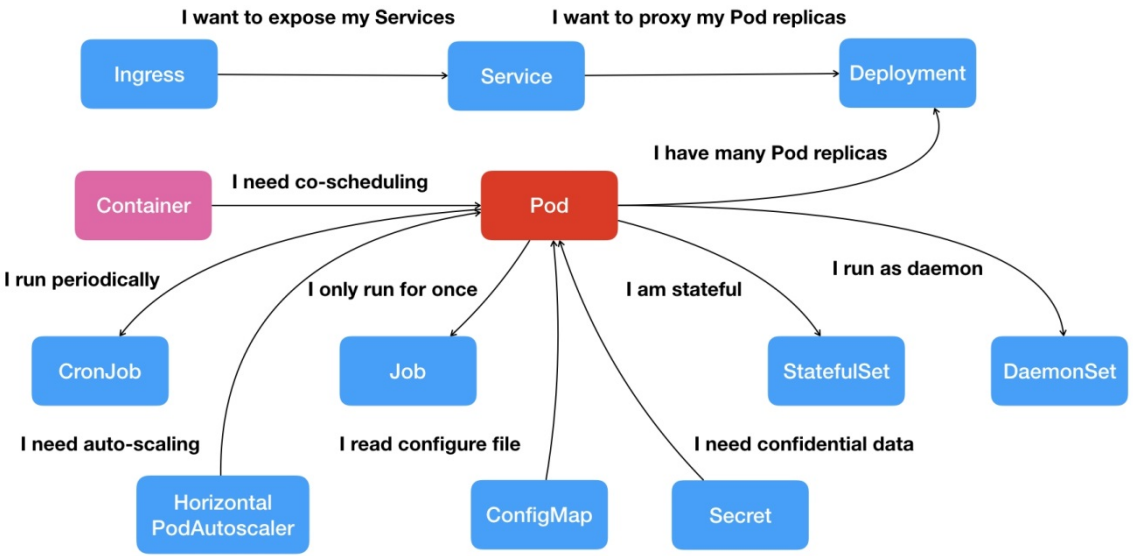
目录

K8S核心构架（引自张磊极客时间专栏）



kubernetes

K8S功能面向



kubernetes

深度依赖 or 核心功能

- K8s service和Dubbo, eureka功能的重叠，及迁移处理？
- ZOOKEEPER，REDIS，MYSQL是否迁移进K8S集群？
- 应用配置是否要与configmap紧密集成？
- PVC功能是否应用及后期维护量(Ceph)？
- 大集群多应用还是单应用小集群(Helm，Spinnaker)？

Eureka server实践

DS Replicas

[redacted]eurekaserver-b

[redacted]eurekaserver-c

[redacted]eurekaserver-a

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
[redacted]	n/a (3)	(3)	UP (3) - [redacted]55d799f977-wktkn , [redacted]
U[redacted]B	n/a (2)	(2)	UP (2) - 1[redacted]80

平安证券
PING AN SECURITIES

DOIS

Filebeat如何推送应用日志到kafka?

Filebeat SideCar

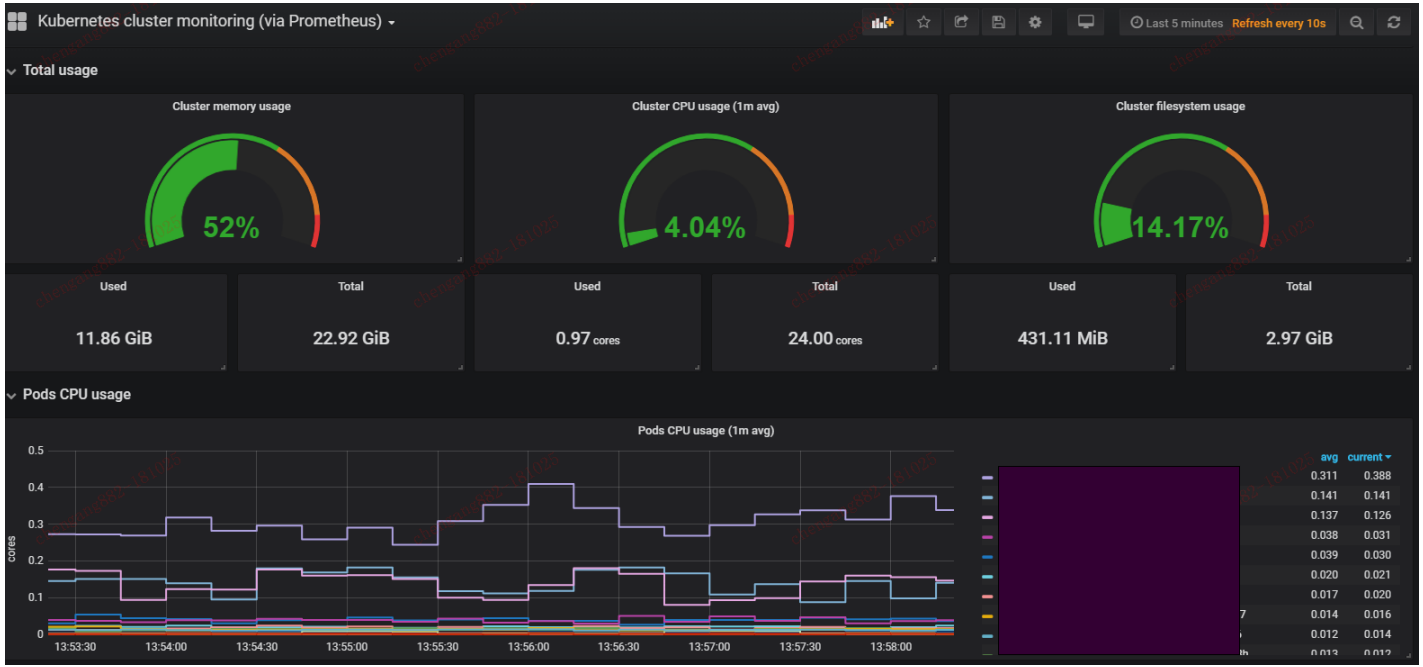
```
sed -i "s#log_home#${log_home}#g" ${filebeat_config}

sed -i "s/log_pattern/${log_pattern}/g" ${filebeat_config}

sed -i "s/kafka_topic/${kafka_topic}/g" ${filebeat_config}

sed -i "s/kafka_hosts/${kafka_hosts}/g" ${filebeat_config}

sed -i "s/kafka_version/${kafka_version}/g" ${filebeat_config}

./filebeat -e -c ${filebeat_config}
```

APACHE kafka
A distributed streaming platform

平安证券
PING AN SECURITIES

DOIS

性能监控：Prometheus & Grafana



平安证券
PING AN SECURITIES

DOIS

Prometheus 如何拉应用指标？(非群集)

POD

```
annotations:
  prometheus.io/path: /xxx/metrics
  prometheus.io/port: "32456"
  prometheus.io/scrape: "true"
```

prometheus

```
- job_name: 'kubernetes-pods'
  kubernetes_sd_configs:
  - role: pod
  relabel_configs:
  - source_labels: [__meta_kubernetes_pod_annotation_prometheus_io_scrape]
    action: keep
    regex: true
  - source_labels: [__meta_kubernetes_pod_annotation_prometheus_io_path]
    ……
  - source_labels: [__address__, _meta_kubernetes_pod_annotation_prometheus_io_port]
    ……
```

Thanks

DevOps 时代社区 荣誉出品

想第一时间看到高效运维社区的新动态吗？

高维在线