在Docker容器中添加对外映射端口

一般在运行容器时，我们都会通过参数 -p（使用大写的-P参数则会随机选择宿主机的一个端口进行映射）来指定宿主机和容器端口的映射，例如：
docker run -it -d --name [container_name] -p 8088:80 -p 3308:3306 [image_name]
上面的命令是将容器内的80端口映射到宿主机的8088端口，容器的3306端口映射到宿主机的3308端口
参数说明
- -d 表示后台运行容器
- -t 为docker分配一个伪终端并绑定到容器的标准输入上
- -i 是让容器的标准输入保持打开状态
- -p 指定映射端口

在运行容器时指定映射端口运行后，如果想要添加新的端口映射，可以使用以下两种方式：

方式一、将现有的容器打包成镜像，然后使用新打包的镜像运行容器时重新指定要映射的端口

1. 停止现有容器
docker stop container_name

2. 把容器commit为镜像
docker commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]
Options:
-a, --author string 作者
-m, --message string 提交信息

3. 用新镜像运行容器
docker run -it -d --name container_name -p p1:p1 -p p2:p2 new_image_name

方式二、修改容器配置文件

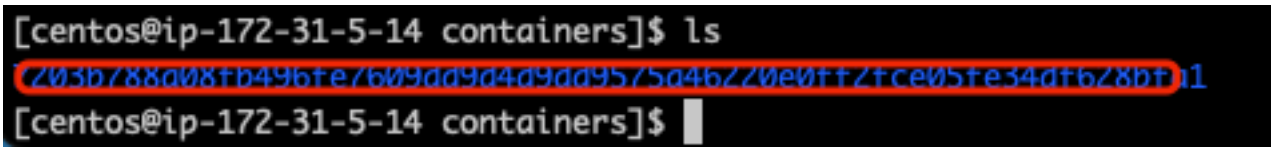
1. 查看全部容器信息，记下要修改容器的ID
docker ps -a

2. 查看容器的端口映射情况（在容器外执行）
docker port [容器ID] 或者 docker port [容器名称]

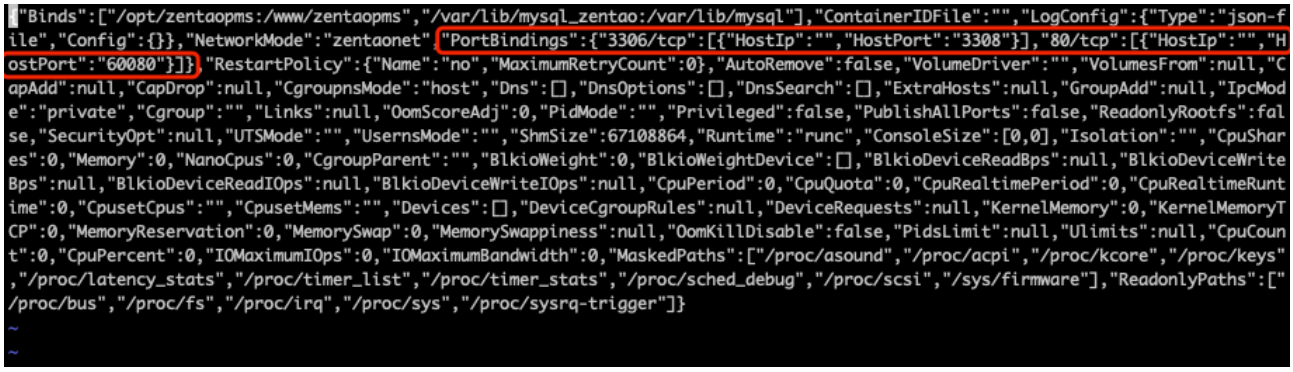
3. 如果要修改的容器在运行，先停掉
docker stop [容器ID]

4. 停止Docker服务
systemctl stop docker

5. 进入/var/lib/docker/containers 目录下找到与 要修改容器的ID 相同的目录



6. 修改 hostconfig.json 配置（**注意格式**）




```
{"Binds":["/opt/zentaopms:/www/zentaopms","/var/lib/mysql_zentao:/var/lib/mysql"],"ContainerIDFile":"","LogConfig":{"Type":"json-file","Config":{}},"NetworkMode":"zentaonet","PortBindings":{"3306/tcp":[{"HostIp":"","HostPort":"3308"}],"80/tcp":[{"HostIp":"","HostPort":"60080"}]},"RestartPolicy":{"Name":"no","MaximumRetryCount":0},"AutoRemove":false,"VolumeDriver":"","VolumesFrom":null,"CapAdd":null,"CapDrop":null,"CgroupnsMode":"host","Dns":[],"DnsOptions":[],"DnsSearch":[],"ExtraHosts":null,"GroupAdd":null,"IpcMode":"private","Cgroup":"","Links":null,"OomScoreAdj":0,"PidMode":"","Privileged":false,"PublishAllPorts":false,"ReadonlyRootfs":false,"SecurityOpt":null,"UTSMode":"","UsernsMode":"","ShmSize":67108864,"Runtime":"runc","ConsoleSize":[0,0],"Isolation":"","CpuShares":0,"Memory":0,"NanoCpus":0,"CgroupParent":"","BlkioWeight":0,"BlkioWeightDevice":[],"BlkioDeviceReadBps":null,"BlkioDeviceWriteBps":null,"BlkioDeviceReadIOps":null,"BlkioDeviceWriteIOps":null,"CpuPeriod":0,"CpuQuota":0,"CpuRealtimePeriod":0,"CpuRealtimeRuntime":0,"CpusetCpus":"","CpusetMems":"","Devices":[],"DeviceCgroupRules":null,"DeviceRequests":null,"KernelMemory":0,"KernelMemoryTCP":0,"MemoryReservation":0,"MemorySwap":0,"MemorySwappiness":null,"OomKillDisable":false,"PidsLimit":null,"Ulimits":null,"CpuCount":0,"CpuPercent":0,"IOMaximumIOps":0,"IOMaximumBandwidth":0,"MaskedPaths":["/proc/asound","/proc/acpi","/proc/kcore","/proc/keys","/proc/latency_stats","/proc/timer_list","/proc/timer_stats","/proc/sched_debug","/proc/scsi","/sys/firmware"],"ReadonlyPaths":["/proc/bus","/proc/fs","/proc/irq","/proc/sys","/proc/sysrq-trigger"]}
~
~
```


7. 修改 config.v2.json 配置（**注意格式**）


```
{"StreamConfig":{},"State":{"Running":true,"Paused":false,"Restarting":false,"OOMKilled":false,"RemovalInProgress":false,"Dead":false,"Pid":7827,"ExitCode":0,"Error":"","StartedAt":"2022-12-06T01:33:58.613151949Z","FinishedAt":"2022-12-05T09:23:59.285180054Z","Health":null},"ID":"7203b788a08fb496fe7609dd9d4d9dd9575a46220e0ff2fce05fe34df628bfa1","Created":"2022-12-05T04:41:06.009349228Z","Managed":false,"Path":"/.docker_init.sh","Args":[],"Config":{"Hostname":"7203b788a08f","Domainname":"","User":"","AttachStdin":false,"AttachStdout":false,"AttachStderr":false,"ExposedPorts":{"3306/tcp":{},"80/tcp":{}},"Tty":false,"OpenStdin":false,"StdinOnce":false,"Env":["MYSQL_ROOT_PASSWORD=123456","PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin","DEBIAN_FRONTEND=noninteractive","LANG=en_US.UTF8","TZ=Asia/Shanghai"],"Cmd":null,"Image":"easysoft/zentao:17.8","Volumes":{"/var/lib/mysql":{}},"WorkingDir":"","Entrypoint":["/.docker_init.sh"],"MacAddress":"02:42:ac:11:00:00","OnBuild":null,"Labels":{}},"Image":"sha256:20478abaeac45d3d9a98a8cc48dab5bc4dc067460b78c0ebeb7675170c959bab","NetworkSettings":{"Bridge":"","SandboxID":"ad22bfb59d6bb53d6c6e02dfdb1184e535e0715241dd251c502e8d645cfe017f","HairpinMode":false,"LinkLocalIPv6Address":"","LinkLocalIPv6PrefixLen":0,"Networks":{"zentaonet":{"IPAMConfig":{"IPv4Address":"172.172.172.17"},"Links":null,"Aliases":["7203b788a08f"],"NetworkID":"076c3c5f538e2a86a30754ba74905ac1ff59518952c559c012c589b998c9ee63","EndpointID":"25560b34d386da9a94eaabd71fad3b42c30a300091fd24bc65a4343119083ece","Gateway":"172.172.172.1","IPAddress":"172.172.172.17","IPPrefixLen":24,"IPv6Gateway":"","GlobalIPv6Address":"","GlobalIPv6PrefixLen":0,"MacAddress":"02:42:ac:11:00:00","DriverOpts":null,"IPAMOperational":false}},"Service":null,"Ports":{"3306/tcp":[{"HostIp":"0.0.0.0","HostPort":"3308"},{"HostIp":"::","HostPort":"3308"}],"80/tcp":[{"HostIp":"0.0.0.0","HostPort":"60080"},{"HostIp":"::","HostPort":"60080"}]},"SandboxKey":"/var/run/docker/netns/ad22bfb59d6b","SecondaryIPAddresses":null,"SecondaryIPv6Addresses":null,"IsAnonymousEndpoint":false,"HasSwarmEndpoint":false},"LogPath":"/var/lib/docker/containers/7203b788a08fb496fe7609dd9d4d9dd9575a46220e0ff2fce05fe34df628bfa1/7203b788a08fb496fe7609dd9d4d9dd9575a46220e0ff2fce05fe34df628bfa1-json.log","Name":"/zentao","Driver":"overlay2","OS":"linux","MountLabel":"","ProcessLabel":"","RestartCount":0,"HasBeenStartedBefore":true,"HasBeenManuallyStopped":false,"MountPoints":{"/var/lib/mysql":{"Source":"/var/lib/mysql_zentao","Destination":"/var/lib/mysql","RW":true,"Name":"","Driver":"","Type":"bind","Propagation":"rprivate","Spec":{"Type":"bind","Source":"/var/lib/mysql_zentao","Target":"/var/lib/mysql"},"SkipMountpointCreation":false},"/www/zentaopms":{"Source":"/opt/zentaopms","Destination":"/www/zentaopms","RW":true,"Name":"","Driver":"","Type":"bind","Propagation":"rprivate","Spec":{"Type":"bind","Source":"/opt/zentaopms","Target":"/www/zentaopms"},"SkipMountpointCreation":false}},"SecretReferences":null,"ConfigReferences":null,"AppArmorProfile":"","HostnamePath":"/var/lib/docker/containers/7203b788a08fb496fe7609dd9d4d9dd9575a46220e0ff2fce05fe34df628bfa1/hostname","HostsPath":"/var/lib/docker/containers/7203b788a08fb496fe7609dd9d4d9dd9575a46220e0ff2fce05fe34df628bfa1/hosts","ShmPath":"","ResolvConfPath":"/var/lib/docker/containers/7203b788a08fb496fe7609dd9d4d9dd9575a46220e0ff2fce05fe34df628bfa1/resolv.conf","SeccompProfile":"","NoNewPrivileges":false,"LocalLogCacheMeta":{"HaveNotifyEnabled":false}}
~
~
```


8. 启动Docker
systemctl start docker

9. 启动容器
Docker start [容器ID]

10. 查看宿主机端口是否和容器内端口是否映射成功（在容器外执行）
docker port [容器ID] 或者 docker port [容器名称]
netstat –an | grep [宿主机的映射端口]