

赵文婧：深入了解 Azure 云平台容器技术服务

容器VS虚拟机



容器和虚拟机作为虚拟化技术，为我们提供了一个隔离的、独立的运行环境，但两者最大的不同之处是容量具有轻量级的优势，上图可以看到，每个虚拟机内部都有完整的操作系统，但一个主机上运行的多个容器是共享Linux内核的，容器和虚拟机的不同之处可以分为以下几点：

- 操作系统：虚拟机有完整的操作系统，容器共享内核。
- 体积：虚拟机包含完整的操作系统，才创建过程中要给其分配一定量的内存去支持运转，所以虚拟机比容器的体积更大，消耗资源更多。
- 启动：在启动虚拟机的过程当中要从操作系统去启动，因此启动较慢，而容器的启动时间是秒级的。

以集装箱为核心的运输体系

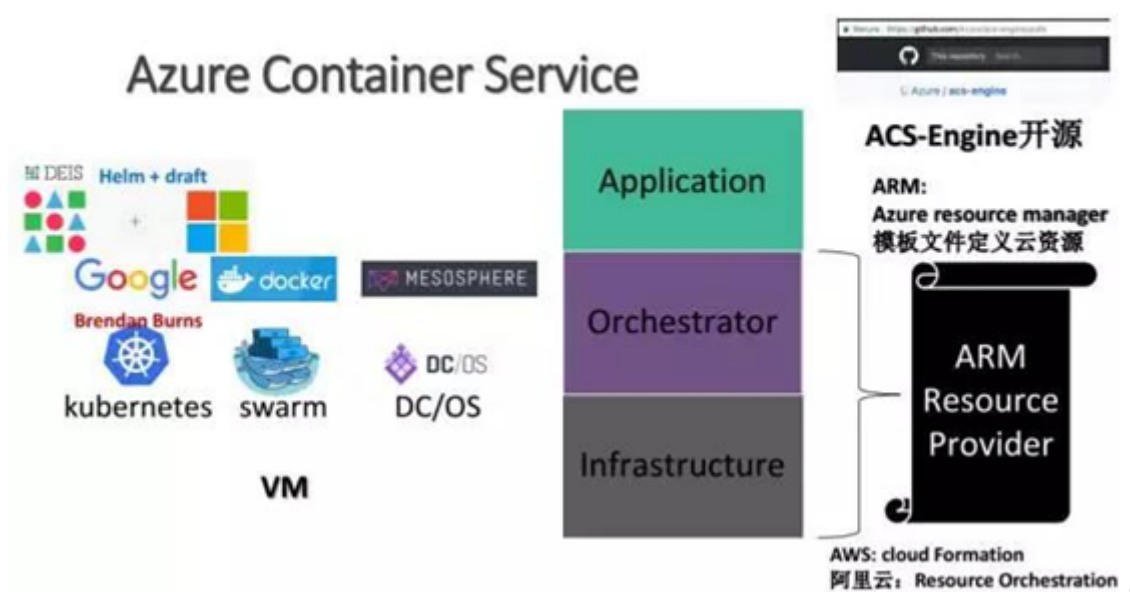


Docker是容器技术发展过程的重要变革，其ICON实鲸鱼驮着非常多的集装箱，这些集装箱即容器，它带来的主要变革有：提供了一套标准化的流程，可以用这个标准化的借口去将自己的应用进行镜像打包，然后在不同的平台上部署和运行，Docker的寓意是码头工人，而它在我们部署运行的过程中起到的是绿色迁移作用。

若把容器迁移过程想象为以集装箱为核心的运输体系，那概念之间可以这样对应：容器是集装箱，云服务提供商是装载集装箱的港口，云服务提供商提供一些IaaS的服务可以理解为装载集装箱的拖船，同时在进行部署运行的过程中有一套标准化的流程，可以将其看做装运流程。

随着容器的不断发展，可以看到可以看到容器在云平台 and 物理机平台会形成一个绿色迁移发展，总结了容器的这两个非常重要特点之后，接下来看一下Azure都提供了哪些容器服务。

Azure Container Service



Azure提供最重要的容器服务名为：Azure Container Service，首先，对于云平台来说，它提供的很重要的资源是Infrastructure，无需购买虚拟机配置分布式集群，云平台提供了

基础的框架如Azure上提供的虚拟机，它也会有其自己的Scale Sets，这些虚拟机会根据需求扩展VM的Sets，也会提供一些Availability Sts，帮助实现高可用。

有了基础设施的架构，即可方便地在云平台索取自己的一些计算机资源，在Infrastructure上，Azure Container Service提供了多种不同的编排工具，因为业务所用的容器是跨主机的，此时容器和容器之间需要进行通信、数据管理、网络管理等，编排工具可以帮助实现跨主机容器应用管理，Azure Container Service提供了三种主流的编排工具：Kuberntes、Swarm、DC/OS，可按需求自行选择。

众所周知，Kubernetes内部具有一些非常复杂的资源类型，虽然它提供了很多重要的功能，但在使用时也会非常繁琐，这里有一些工具可以帮助更好地使用Kuberentes，如DEIS这家公司的Helm和Draft。

- Helm相当于软件包的管理器，类似于Linux上的Apt-Get或Yum，通过Helm可以自动获取一些名为Helm Chart的软件包去快速构建部署Kuberentes集群上的应用。
- Draft能够自动识别应用的语言，帮助去写Dockerfile和Helm Charts以及快速的构建部署引用，同时，Kuberntes是谷歌根据多年的系统经验做的编排工具，Kuberentes的创始人后来也加入了微软支持Azure。

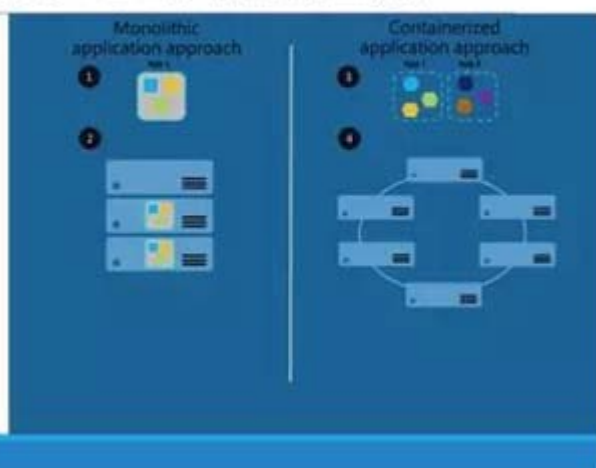
之前说到的Infrastructure和以上编排工具都通过Azure的ARM模型进行构建，大家知道AWS有Cloud Formation，阿里云上有Resource Orchestration，类似于这种模板文件，Azure上有自己的ARM文件，可以用一些很简单的输入，通过ARM模板在Azure进行部署，除了可以在Azure的Portal上或Azure CLI上去创建Azure Container Service，也可以通过ACS的Engine去创建，使用一些简单输入的定义文件用ACS Engine转化为ARM模板，从而快速去创建部署，ACS Engine现在是开源在GitHub上的，用户可以在其中快速更改自己的部署，从而进行一些自定义的容器服务，我们也希望这些开源的代码能够对社区做出贡献，也帮助我们ACS的服务更快更好地发展。

微服务+容器化

集群部署：单体应用 VS 容器化应用

•微服务

- 分开编译和部署
- 小的独立执行的任务
- 集成时调用published API
- 细粒度松耦合的应用程序



之前了解了Azure Container Service提供的一些服务，下面再分享下微服务和容器化如何更好的结合，在集群部署的情况下，Azure Container Service在哪些方面起到相关作用。

首先，单体的应用。在集群部署的过程中和容器化的应用在集群部署的过程中有什么区别，提到微服务，简单地说即应用程序有多少独立的组件，希望它们能够独立的进行开发测试以及运行，但假设一些单体的应用向把它部署到集群上时，为了实现高可用，将它在集群上的每个节点都进行部署，但对于容器化的应用，可以把这些应用拆分成多个独立的组件，每个独立的组件都能放在容器之中，为它提供一个独立的运行环境，此时，当应用的模块组件放在分布式集群这些节点上时，即可自动地去根据分布式节点的计算资源调整模块分布的位置，假如在ACS部署了一个集群，ACS会帮助做一些负载均衡及网络管理的设置，可以在容器集群上更好地部署容器化应用。

DevOps



说到DevOps，目前很难给它做一个具体的定义，但其主要目的是在应用开发测试以及运行过程中做一个更好地衔接，使应用更新能够快速地进行发布，从而提升提升效率，所以DevOps和容器的特性不谋而合。

通过容器技术可以达到一次编写部署，在这个基础上构建敏捷的DevOps开发若把应用构建在云上，使用Azure上的容器服务，有一些工具可以使用如：Azure Container Registry，可以在它上面构建私有的镜像仓库，其中存储着Docker格式的镜像，此仓库可以与Azure Container Service提供的编排引擎进行集成，如果想在Kubernetes集群上进行应用部署，可直接从这个镜像仓库中Push一个镜像在上面运行，这个Push下来的过程即可直接写在Kubernetes相关的部署文件当中。

镜像仓库是可以和Docker的镜像仓库兼容的，所以如果使用的是Docker工具，也可以进行无缝迁移，同时在Azure上使用容器化服务以及一系列持续集成工具链包括Visual Studio Visual Studio Team Service以及Visual Studio Code，通过一系列的集成工具链即可构建版本代码库，同时进行人员分配的管理调度，达到更好的持续集成、持续交付概念。



接下来会为大家用比较多的时间演示Demo：在ACS上创建一个服务，并运行应用。

第一步：创建ACS的服务

有两种方式：

- 在Portal里面进行创建。
- 在Azure的CLI里面进行创建。

首先，这是Azure的Portal，进到里面创建Azure Container Service，可以自定义服务名称，因要使用Kubernetes，我定义这样一个名字，选择资源组，接下来定义Master节点的属性，可以看到很多编排工具可以选择，选择Kubernetes，重新再来看一下，最上面可以选择三种不同的编排引擎，接下来我们会定义Master节点的用户名，然后用SSH的Key去登录我们的Portal。

将SSH的Key生成完毕后，直接粘进去，下面是在创建服务时需要一个服务主体，之前已经创建好了，使用它的ID和密码，这里可以设置Agent节点的数量以及设置VM的Size，我们确认信息，然后即可创建Kuberentes的集群了，同样还有另外一种方法可以进行Azure的CLI里，用Az Acs Create 这样的命令来创建自己的集群，那么在这里我们可以指定我们的编排引擎，接着指定我们要将集群放在哪个资源组里面。

在Azure平台里，其实都是通过资源组去管理服务，比如服务需要什么样的计算资源，都会统一放在某个资源组里，此时，让不需要这项服务即可删除整个资源组，然后去删除这些服务，下面定义节点的VM Size以及SSH Key的这些选项，进入Portal里看下之前创建好的Kuberntes集群，可以看到这已经定义了Agent Master这样一些节点，还包括一些负载均衡以及网络的组件，进到Cluster里面看下它的组成：一个Agent Pool，VM Size VM Count显示等，其实只要一些很简单的输入，即可快速地在Azure上创建一个ACS服务。

第二步：测试例子应用程序的本地镜像部署和运行

接下来将一个WEB的应用部署在集群中，首先将WEB的应用在本机上进行镜像的打包和运行，从这个地址Clone一个应用，确认Docker是正在运行着的，查看本机上有哪些Docker镜像，可以看到目前还没有，接下来用Docker Compose这个命令去创建有关应用的所有进项，它会读取Docker Compose文件，进到Docker-Compose文件里查看，定义了两个服务，包括Azure-Vote-Back和Azure-Vote-Front，Redis是数据库的服务，Azure-Vote-Front是前端的投票应用服务。

在创建的过程中，会Push一些基础镜像扩，包括Nginx这个正在Pull下来的WEB Server镜像，还有数据库的镜像，生成应用自身的镜像，完毕后，可以看到本地现在有三个镜像，此时有两个容器正在运行，包括数据库和应用，再来确认下Docker Compose的文件中所确定的应用，本机和容器之间的端口映射，可以看到，进到本机8080端口可以运行WEB应用，这是一个投票软件，会在这个WEB上面进行投票，而后数据库会对这些投票进行记录，此应用现在已经在机器上运行成功，下面会将这个应用部署到刚才创建的集群上面。

第三步：创建Azure Registry并将应用镜像Push进去

在将应用Push到集成之前，首先需要创建一个Azure上的容器注册表，即创建属于自己的镜像仓库，可以看到镜像仓库已经创建完成，可以使用其中的Login Server和Password进行登录，现在需要获取此进项仓库的密码，成功后在本地登录到镜像仓库中，将之前创建的镜像Push到进项仓库中，在此之前，需要用Docker Tag命令为镜像打一个标签，之后用Docker Images确认，这是为了下次在进行应用更新时的版本控制。

现在把镜像Push到刚才创建的镜像仓库中，在Push的过程里会不断地显示Push的阶段，成功后进到Azure的CLI中查看已经Push成功镜像的List，此时显示了刚才大号标签的镜像已经Push到镜像仓库中。

第四步：用Kubectl连接到Kubernetes集群并在集群上运行应用

可以看到在这个应用下载的例子中，有一个定义Kubernetes部署的文件，那么在这个文件中定义了不同的Deployment和不同的Service，可以在这个文件中写清自己的部署，然后利用此文件创建Kubernetes的部署。

将这里使用的镜像更换成之前在镜像仓库中Push进去的镜像，更换镜像仓库的Loginserver，然后会用Kubectl Create的命令根据刚才所看到的部署文件去创建部署和服务。

现在创建了WEB应用的部署和一些Service，接下来即可Get刚才创建的Service，然后同时用Watch参数进行监控，获得IP后，即可将Watch关掉，进入External的IP里查看应用的运行情况。

第五步：为集群做扩展

进入此IP后，即可看到UI界面，部署完应用后，如何给集群做扩展？有两个方面：包括Kubernetes的单元Pod扩展，也包括集群节点的扩展，依次来看一下：

Kubernetes里面的Pods的情况，现在给Azure-Vote-Front进行扩展，用Kubectl Scale的这个命令扩展副本数是5个，扩展完成后再用Kubectl Get Pods这个命令进行查看，此时

Azure-Vote-Front就变成了需要的5个量，可以在更多的量上去做自己的应用计算资源的应用，接下来看一下刚才部署文件中它其实是设置了有关请求的CUP的量，M其实指的是一千分之一核这样的一个单位，请求CPU量为250M现在在500M之内，可以根据CPU的使用率去进行后面的扩展。

定义CPU的Pescent是50，即当CPU的使用率超过50%时，可以给它达到10个Azure-Vote-Front Pods的情况，可以通过这样一个命令来查看已经设置过的自动扩展状态，除了在Kubernetes的Pods做这样一个扩展，还可以在VM的Count这个层面上去做一个扩展，可以设置New-Agent-Count这个数字为4，此时会从刚才的3个Count编程4个Count。

第六步：应用更新后的快速发布

最后一步会为大家分享如何进行应用更新后的快速部署和运行，进到刚才下载下来的应用，进入UI定义文件，将刚才的Cats和Dogs更改成其他的单词，让它可以显示在UI上，以表示更新。完成后仍之前提过的Count应用重新给应用进行打包镜像，成功后此时将应用以一个新的标签来传到刚才的Registry里，显示刚才的Count结果，定义Count为4，此时编程了新的VMcount，相当于在Cluster理有4个可用的VM。

把刚才更新好的应用重新打一个标签：V2，发现已经定义成功，接下来把打好标签的镜像Push到之前创建好的镜像仓库中，可以看到其实有一些内容已经存在了，这是刚才扩展后的情况，对这个更新好的镜像进行重新部署，它提示这个镜像已经更新成功，重新看到Pods的情况是有一些已经进行更新了，从最右边的Age来看Pods的存在时间，下面重新Get Service来去查看这个应用的运行结果，登录到这个响应的IP里去查看一下。

可以看到现在选项已经编程了White和Black，从进行应用更新到快速在集群上部署运行，只需要很短的时间即可进行。

今天给大家介绍的内容主要是有关Azure Container Service上面的容器化应用。谢谢大家。