## Alibaba Cloud

## Elastic Container Instance Best Practices

Document Version: 20220705

C-J Alibaba Cloud

### Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

- You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloudauthorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
- 2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.
- 3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
- 4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).
- 5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud and/or its affiliates Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
- 6. Please directly contact Alibaba Cloud for any errors of this document.

### **Document conventions**

Style	Description	Example
<u>↑</u> Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	Danger: Resetting will result in the loss of user configuration data.
O Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.
C) Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	Notice: If the weight is set to 0, the server no longer receives new requests.
? Note	A note indicates supplemental instructions, best practices, tips, and other content.	Note: You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click Settings> Network> Set network type.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click OK.
Courier font	Courier font is used for commands	Run the cd /d C:/window command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	bae log listinstanceid Instance_ID
[] or [a b]	This format is used for an optional value, where only one item can be selected.	ipconfig [-all -t]
{} or {a b}	This format is used for a required value, where only one item can be selected.	switch {active stand}

### Table of Contents

1.Use and integrate the Elastic Container Instance terminal <sup>0</sup>	)5
2.Use a virtual node to enable auto scaling for online business o	)6
3.Schedule pods to a virtual node through the virtual-kubelet-au 1	2
4.Use an elastic container instance to run jobs1	8
5.Run jobs on a preemptible instance 1	9
6.Use GitLab CI 2	23
7.Deploy Jenkins in an ASK cluster and build an application deli 2	24
8.Install and use WordPress 2	29
9.Build a Spark application 3	31
10.Run a GPU-based TensorFlow training job 3	39
11.Access common data sources 4	13
11.1. Access data in OSS from ECI 4	13

## 1.Use and integrate the Elastic Container Instance terminal

This topic describes how to use the Elastic Container Instance terminal and integrate it into your own system.

#### Use the Elastic Container Instance terminal

If you want to perform operations in a container when you use an elastic container instance, you can call the ExecContainerCommand operation to obtain the WebSocketUri value. The value indicates the WebSocket URL. The WebSocket URL cannot be directly accessed by using a browser. You can use the HTTP URL returned by the ExecContainerCommand operation.

The HTTP URL is effective for 30 seconds. To use the HTTP URL to access a container, you must access this URL in your browser within 30 seconds after the ExecContainerCommand operation is called.



This operation of accessing a container is audited. Audit information includes:

- 1. The time, ID, client IP address, and parameters of the API request that was sent to obtain the HTTP URL
- 2. The client IP address that corresponds to the HTTP URL
- ? Note

Audit information can be retained for up to 30 days. You can view audit records on the **Terminal Operational Log** page in the **Elastic Container Service console**.

## Integrate the Elastic Container Instance terminal into your own system

After you obtain the HTTP URL by calling the ExecContainerCommand operation, you can use one of the following methods to integrate the Elastic Container Instance terminal into your own system:

• Access the HTTP URL in a separate window.

Use a frontend application to access the HTTP URL in a new window.

• Embed the HTTP URL into the current page.

Embed the HTTP URL into a page of your own system by using an if rame.

## 2.Use a virtual node to enable auto scaling for online business applications

This topic describes how to use Container Service for Kubernetes and Elastic Container Instance (ECI) to enable auto scaling for online business applications.





The preceding figure shows three applications and how they work together. User requests are sent to biz-app, which simulates an entry application. Then, biz-app forwards the requests to cpu-app and mem-app, which simulates a CPU-intensive application and a memory-intensive application respectively. Therefore, you need to prepare three Docker images.

- biz-app: registry.cn-hangzhou.aliyuncs.com/eci-springboot-demo/biz-app:1.0
- mem-app: registry.cn-hangzhou.aliyuncs.com/eci-springboot-demo/mem-app:1.1

• cpu-app: registry.cn-hangzhou.aliyuncs.com/eci-springboot-demo/cpu-app:1.0

#### Step 2: Create a managed Kubernetes cluster

C-) Alibaba Clou	d Account's all R 👻 🍕	Ilobal 🕙	Q Search	Bil	lling Ticket ICP	Enterprise	Support A	libaba Cloud	Þ Á	₩ EN	۹ 🧿
te Kubernetes Cluster	Dedicated Kubernetes	Managed Kubernetes	Serverless Kubernetes	Managed Edge Kuberne	tes (Preview) Crea	ate Cluster Conn	ect Proxy (Previ	iew) Clus	ter Templates	Ba	ick
1 Cluster 0	Configurations	2 Work	er Configurations		Component Configu	irations		- 4 0	Confirm Orde	r	
* Cluster Name								Current C	onfiguration		
	The name must be 1 to 63 of	characters in length and can	contain letters, Chinese ch	aracters, digits, and hypher	ns (-).			Region: Chi	na (Beijing)		
Region	China (Beijing)	China (Zhangjiakou)	China (Hohhot)	China (Hangzhou)	China (Shanghai)			Worker	Quantita	. ,	
	China (Shenzhen)	China (Chengdu)	China (Hong Kong)	Japan (Tokyo)	Singapore			Type.	Quantity	. 5	
	Australia (Sydney)	Malaysia (Kuala Lumpur)	Indonesia (Jakarta)	India (Mumbai)	US (Virginia)			🛆 Note Exp	ose API Serve	with EIP	
	US (Silicon Valley)	UK (London)	Germany (Frankfurt)					🛆 Note Ena	ible Log Servic	e	
Resource Group	Not Selected							To activate Pa your account USD 100. Mai balance is suf	ay-As-You-Go balance must ke sure that yc ficient. Rechar	ECS instand be no less ur account ge now	ces, than :
Kubernetes Version	1.16.9-aliyun.1	1.14.8-aliyun.1						To activate E0 complete the Register Now	CS or SLB insta real-name reg	nces, you n istration fi	nust rst.
Container Runtime	Docker 19.03.5	Sandboxed-Container 1.1.	D					You can creat	e a maximum	of 20 cluste	ers.
	How to select between Doc	ker and Sandboxed-Containe	er					Each cluster o nodes. To rec ticket.	an contain a n juest a quota i	ncrease, su	f 100 bmit a
VPC	jiajun-k8s-vpc1 (vpc-2ze	0aqj6nj8xjbblfw4xs) 🔻	c					If you need o ticket.	ther instance t	ypes, subm	nit a
		erestes CIDD blacks in VDC	notworks								

This topic takes a managed Kubernetes cluster as an example. This cluster uses nodes backed by Elastic Compute Service (ECS) instances to sever normal traffic, and uses a virtual node backed by ECI through the virtual-kubelet-autoscaler add-on to serve burst traffic.

## Step 3: Install Virtual Kubelet and the virtual-kubelet-autoscaler add-on in the managed Kubernetes cluster



Log on to the Container Service console. In the left-side navigation pane, choose Marketplace > App Catalog. On the App Catalog page that appears, search for ack-virtual-node, and click the card of the add-on to install the add-on. After the ack-virtual-node add-on is installed, install the ack-virtual-kubelet-autoscaler add-on in the same way.

#### Step 4: Create the applications

=	<b>C-)</b> Alibaba Clo	Account's all R 👻 Global	Q Search		Billing	Ticket	ICP	Enterprise	Support	Alibaba Cloud	>_	٥.	₩ EN	u 📀
c	Container Service - Kubernetes -	Deployments							Refresh	Create from	Image	Creat	te from Ter	nplate
• 0	Dverview Cluster	Ø How to use private images Ø Create a Ø Blue-green release Cluster xiaohui-test ≜ Namespace k	applications Schedule a pod to a sp	ecific no	ode 🔗 Create a layer-4 ingress	s 🔗 Create	a layer-7	Ingress 🔗 C	onfigure Pod	auto scaling 🖇	Monitor co	ntainers		
	Cluster		Label	Pods	Image				Crea	ited At	Search by	name		Q, Actions
	Persistent Volumes	ack-virtual-node-affinity-admission- controller	app:ack-virtual-node-affinity- admission-controller	1/1	registry-vpc.cn-hangzhou.aliy admission-controller:v1.0.0.1	/uncs.com/ac -aliyun	:s/virtual-	node-affinity-	Jun 12:3	1, 2020, 5:02 UTC+8	De	tails	Edit   Monitor	Scale     More <del>-</del>
	Namespace Authorizations	alicloud-application-controller	app:alicloud-application-controller owner:aliyun	1/1	registry-vpc.cn-hangzhou.aliyuncs.com/acs/aliyun-app-lifecycle- manager:v0.1.0.1-f832bed-aliyun 11:26:48 UTC+8							tails	Edit   Monitor	Scale     More <del>-</del>
•	Applications	alicloud-disk-controller		1/1	registry-vpc.cn-hangzhou.aliy controller:v1.14.8.44-c23b62c	/uncs.com/ac c5-aliyun	s/aliclou	d-disk-	May 11:2	29, 2020, 6:48 UTC+8	De	tails	Edit   Monitor	Scale     More <del>▼</del>
C	Deployments StatefulSets	alicloud-monitor-controller	task:monitoring k8s-app:alicloud-monitor-controller	1/1	registry-vpc.cn-hangzhou.aliy controller:v1.3.0	/uncs.com/ac	s/aliclou	d-monitor-	May 11:2	29, 2020, 6:49 UTC+8	De	tails	Edit   Monitor	Scale     More <del>-</del>
	DaemonSets	aliyun-acr-credential-helper	app:aliyun-acr-credential-helper	1/1	registry-vpc.cn-hangzhou.aliy helper:v20.03.16.0-36d5d7e-	/uncs.com/ac aliyun	:s/aliyun-	acr-credential-	May 11:2	29, 2020, 6:48 UTC+8	De	tails	Edit   Monitor	Scale     More <del>-</del>
	Cron Jobs	coredns	k8s-app:kube-dns	2/2	registry-vpc.cn-hangzhou.aliy	/uncs.com/ac	:s/coredr	s:1.6.2	May 11:2	29, 2020, 6:48 UTC+8	De	tails	Edit   Monitor	Scr III
	Pods	metrics-server	k8s-app:metrics-server	1/1	registry-vpc.cn-hangzhou.aliy bd3ae4f-alivun	/uncs.com/ac	s/metric:	s-server:v0.2.1	- May 11:2	29, 2020, 6:49 UTC+8	De	tails	Edit   Monitor	Sca BO
	Persistent volume Cl	Batch Delete						Tota	l: 9 itern(s), P	er Page: 25 🛊	itern(s)	α (	1	3

Select the type of the applications based on your business requirements. This topic uses deployment as an example. In the left-side navigation pane, choose Applications > Deployments.

On the page that appears, click Create from Image in the upper-right corner. On the page that appears, set the parameters as required, specify one of the prepared application images, and then configure the service. Repeat the preceding steps to create the other two applications in sequence.

When you create biz-app, you need to specify two environment variables, as shown in the following figures. Set the two environment variables to the internal service endpoints of cpu-app and mem-app, respectively.

	Environment Variable:	• Add			
nments		Туре	Variable Key	Value/ValueFrom	
Enviro		Custom \$	cpu.app.endpoint	cpu.app.svc:8002	•
		Custom \$	mem.app.endpoint	mem-app-syc:8003	•

In addition, you need to create an ingress for biz-app to expose biz-app to the Internet.

路	∄ (Ingress)			Ingress概览	刷新创建
6	Ingress访问日志分析与监	<u>监控</u> 🔗 蓝绿发布			
集群	vk-autoscale-hz \$	命名空间 default	\$	输入名称查询	Q
	名称	端点	规则	创建时间	操作
	biz-app-ingress	121.199.255.9	vk-autoscale-h.ccf4af503a12e4508b24e1a9d0dfdb076.cn-hangzhou.ali> biz-app-svc	2019-10-17 13:42:55	详情 │更多▼
	批量删除				

#### Step 5: Configure auto scaling policies

On the Deployments page, click the target application. On the application details page that appears, click the Horizontal Pod Autoscaler tab. Then, click Edit in the Actions column of an autoscaler to configure the policy.

=	C-J Alibaba Clo	ud 🛛 Account's all R 🔻 🧬 Giol	Q Search B	illing Ticket ICP Enterprise Support Alibaba Cloud 🔄 🎝 🗑 EN 🙆
c	ontainer Service -	Annotations:	Create	×
	Kubernetes -	Selector:	Name:	
0	verview	Strategy:		
▼ C	luster		Metric: OAdd	
	Cluster		Metric Name Thresho	ld
	Nodes	Status:	Memory Usage \$	% •
	Persistent Volumes	Conditions		
	Namespace	Type Status	Max. Containers: 10	
	Authorizations	Available True	Min. Containers: 1	ilability.
▼ A	pplications	Progressing True		toscaler-7dd55df5db* has successfully progressed.
	Deployments	Triggers 1. You can only have one	Create from Temp	late OK Cancel Create Trigger
	StatefulSets	No trigger is currently available. Click Cr	e frigger in the upper-right corner.	
	DaemonSets	Pods Access Method Eve	Horizontal Pod Autoscaler History Versions	
	Jobs	HPA Create		
	Cron Jobs			Min Benlinge Max Benlinge Crusted At Advisor
	Pods			
	Persistent Volume Cl		<li>① No data a</li>	available.

Be default, you can configure auto scaling policies based on the memory usage and CPU usage. You can also create custom auto scaling policies as required.

rments											
ions	mem-app		MEMORY : 60%	MEMORY : 60%				0	Nov 23, 2020, 16:20:40 UTC+8	Status   Events   Edit   Dele	
	Name			Target Utilization Ra	te Current Metrics	Min. Re	plicas Max	k. Replicas	Current Replicas	Created At	Actio
izations											
paces	HPA	Create									
ent Volumes	Pods	Access Method	Events	Horizontal Pod Autoscaler	History Versions	Logs	Triggers				

#### Step 6: Install third-party add-ons

On the App Catalog page, you can search for the required add-on and click the card of the add-on to install the add-on. In this example, the add-ons shown in the following figure are installed.

Release	es								
Helm	Canary Release (Pub	lic Preview)	Batch Release						
Cluster	never-delete-eci-fvt-test	\$							
Release	Name	Status	Namespace	Chart Name	Chart Version	Application Version	Updated At		
ack-node	e-problem-detector	Deployed	kube-system	ack-node-problem-detector	1.2.1	0.8.0	Aug 18, 2020, 19:17:46 UTC+8	View Details	ι
ack-pron	netheus-operator	Deployed	monitoring	ack-prometheus-operator	5.7.1	0.29.5	Sep 16, 2020, 12:28:17 UTC+8	View Details	ι
ack-virtu	al-kubelet-autoscaler	Deployed	kube-system	ack-virtual-kubelet-autoscaler	0.0.1	0.0.1	Nov 23, 2020, 10:51:00 UTC+8	View Details	ι
ack-virtu	al-node	Deployed	kube-system	ack-virtual-node	0.0.1	0.0.1	Sep 16, 2020, 13:02:29 UTC+8	View Details	ι

- ack-virtual-node: uses Virtual Kubelet to schedule pods to ECI.
- ack-virtual-kubelet-autoscaler: uses Virtual Kubelet to schedule pods to ECI when resources are insufficient on physical nodes. The scalability of the managed Kubernetes cluster described in this topic is achieved through this add-on.
- ahas: provides features such as automatic detection of application architectures, high availability assessments based on fault injection, and one-clickthrottling and degradation.
- arms-pilot: automatically detects application topologies, generates 3D topologies, detects and monitors interfaces, and captures abnormal and slow transactions.
- arms-prometheus: supports the open-source Prometheus ecosystem and monitors a wide variety of components. The add-on offers a ready-to-use monitoring dashboard and provides fully managed Prometheus services.

#### Step 7: Use Performance Testing to run a stress test

Use Performance Testing to simulate a traffic change from off-peak hours to peak hours.

编辑场景 返回场景列表		导入脚本 🛛 🖧 参数列表 🛛 😚 系统函数 🛛 📔 数据源管理	保存配置 调试场景 保存去压测
* 场景名 demo	快速入门		
* 场景配置 * 施压配置	域名绑定 添加监控 SLA定义(1058及以上资源	短可用)	压测场景的结构和数据分配
<b>┨</b> ₽ 串联链路			۰۰۰ 🎂
	demo GET http://vk	-autoscale-h.ccf4af503a12e4508b24e1a9d0dfdb076.c	• ~ ^
	基本请求信息 Header定义	出参/断言定义 ⑦	
	* 圧測URL	fdb076.cn-hangzhou.allcontainer.com/biz	目表
	请求方式 GET 🗸 * 超时时间 ⑦ 200	※秒 允许302跳转 ⑦ ✓	
	+ API模板 ~ + 添加压测API ~		
+ 添加串联链路模板	~	+ ž	添加串联链路

#### Step 8: Check the result

#### Elastic Container Instance

#### Best Practices Use a virtual node to enable auto scaling for online busin ess applications

名称: 创建时间: 注解: 策略: 状态:		service 2020-0 deployr Rollingl 就绪: 1	-b )8-11 14:04:15 ment.kubernetes.io/rev Jpdate 2/12个,已更新:12个,	vision:2 可用:12个	展开现状详	情▼		命名空间: 标签: 选择器: 滚动升级\$	<b>良略</b> :	default app:service-b app:service-b 超过期望的Pod数量:26% 不可用Pod最大数量:25%			
容器组	访问方式	7万式 事件 容器组水平伸缩器 历史版本 日志 触发器											
类型 (全部)	▼ 対象 (全部) ▼ 信息								原因		时间		
Normal	Deployn service-	nent b	Scaled u	Scaled up replica set service-b-745fdf9cdd to 12					ScalingReplica	aSet	2020-08-11 17:04:20		
Normal	Deployment service-b Scaled up replica set service-b-745fdf9cdd to 10				o 10		ScalingReplica	aSet	2020-08-11 16:50:10				
Normal	Deploym service-	b b	Scaled u	p replica set s	ervice-b-7	45fdf9cdd t	o 8		ScalingReplica	aSet	2020-08-11 16:47:07		
Normal	Deploym service-	b b	Scaled u	p replica set s	ervice-b-7	45fdf9cdd t	o 6		ScalingReplica	aSet	2020-08-11 16:44:18		
Normal	Deployn service-	b b	Scaled u	p replica set s	ervice-b-7	45fdf9cdd t	o 5		ScalingReplica	aSet	2020-08-11 16:41:15		
Normal	Deployment Scaled up replica set service-b-745fdf9c				45fdf9cdd t	o 4		ScalingReplica	aSet	2020-08-11 16:39:13			
Normal	Deployment Scaled up replica set service-b-745fdf9cdd to 3				o 3		ScalingReplica	aSet	2020-08-11 16:23:11				
Normal	rmal Deployment service-b			Scaled up replica set service-b-745fdf9cdd to 2					ScalingReplicaSet		2020-08-11 16:21:09		

As shown in the preceding figure, when the queries per second (QPS) are changed from 1 to 5, 10, and 20, resources are scaled for mem-app and cpu-app based on the loads.

创建时间: 注解: 策略: 状态: <sup></sup>	<ul> <li>问: 2020-08-11 14:04:15 deployment.kubernetes.io/revision:2 RollingUpdate 就绪: 12/12个,已更新: 12个,可用: 12个 展开现状详情↓</li> <li>집 访问方式 事件 容器组水平伸缩器 历史版本 日志 触然</li> </ul>				情 <del>▼</del> •••	标签: 选择器: 滚动升级策略:			app:service-b app:service-b 超过期望的Pod数量	註25% 不可用Pod最大数	25%		
名称	71 (719)	領像	日本市场上小一一叶州市市	状态 (全部)	•	RELAX tur		监控	重启 次数	Pod IP	节点	创建时间	操作
service-b- 745fdf9cd	d-5nnmk	aliyuneci/a demo-serv	ick-vk-autoscaler- vice-b:1.0	Running				Ł	0	192.168.0.164	virtual-node-eci-0 172.20.0.14	2020-08- 11 16:44:19	详情   日志   更多 ▼
service-b- 745fdf9cd	d-64pqt	aliyuneci/a demo-serv	ick-vk-autoscaler- ⁄ice-b:1.0	Running				Ы	0	192.168.0.166	virtual-node-eci-0 172.20.0.14	2020-08- 11 16:47:07	详情 日志 更多 ▼
service-b- 745fdf9cd	d-92jkn	aliyuneci/a demo-serv	ick-vk-autoscaler- vice-b:1.0	Running				Ы	0	192.168.0.159	virtual-node-eci-0 172.20.0.14	2020-08- 11 16:24:41	详情 日志 更多 ▼
service-b- 745fdf9cd	d-9s2jx	aliyuneci/a demo-serv	ick-vk-autoscaler- vice-b:1.0	Running				Ч	0	192.168.0.167	virtual-node-eci-0 172.20.0.14	2020-08- 11 16:47:07	详情 日ī 更 >
service-b- 745fdf9cd	d-ktv5b	aliyuneci/a demo-serv	ick-vk-autoscaler- vice-b:1.0	Running				Ł	0	192.168.0.168	virtual-node-eci-0 172.20.0.14	2020-08- 11 16:50:10	详h 日志   更多 ▼
service-b- 745fdf9cd	d-lkg55	aliyuneci/a demo-serv	ick-vk-autoscaler- vice-b:1.0	Running				¥	0	172.20.0.16	cn- beijing.192.168.0.154 192.168.0.154	2020-08- 11 14:37:33	详情 日志 更多 ▼

# 3.Schedule pods to a virtual node through the virtual-kubelet-autoscaler add-on

#### Overview

This topic describes how to use the virtual-kubelet-autoscaler add-on to schedule pods to a virtual node when computing resources are insufficient on physical nodes of a Kubernetes cluster. This brings ultra scalability to your business applications.



#### Before you begin

1. Log on to the Container Service console to view the Kubernetes cluster that you have created. If you do not have a cluster, create one. For more information, see Create a Kubernetes cluster.

2. Install the ack-virtual-node add-on. For more information, see Virtual nodes.

3. Install the virtual-kubelet-autoscaler add-on. To install the add-on, follow these steps: In the leftside navigation pane, choose Marketplace > App Catalog. On the App Catalog page that appears, search for ack-virtual-kubelet-autoscaler and click the card of the add-on that is found.

#### Elastic Container Instance

#### Best Practices Schedule pods to a v irtual node through the virtual-kubel et-autoscaler add-on

	😑 🕞 Alibaba Clo	OUD Account's all R 🔻 🥙 Global Q. Search Billing Ticket ICP Enterprise Support Alibaba Cloud 🖸 🗘 🏋 E	en 🌔
	Container Service - Kubernetes <del>-</del>	App Catalog Name 2 ack-virtual-kubelet	3 Q
	Istio Management	Alibaba Cloud Apps App Hub Heimi	Releases
	Virtual Services	All(45) Operations/Observability(5) Microservices(2) Multi-cluster/Hybrid Cloud(1) Serverless Container(2) Auto Scaling(2) Application Management(6) Big Data/Al(8) C/CD(3) Workflow(1) Blockche	ain(O)
	Gateways	Edge Computing(2)	
	<ul> <li>Knative</li> </ul>		
	Components		
	Services		
	Configuration	ack-virtual-kubelet-autoscaler 0.0.1 incubator	
	ConfigMaps		
	Secrets	Statement on Intellectual Property	
1	<ul> <li>Marketplace</li> </ul>		
	Docker Images		
	Orchestration Templa		
	App Catalog		
1	Service Catalog		
	Quick Start		
	🗟 1.png	<u>気</u> 示	全部X

On the page that appears, select the target cluster in the upper-right corner and click Create.

=	<b>C-)</b> Alibaba Clo	OUD Account's all R	🔹 🧬 Global	Q Search		Billing	Ticket	ICP En	terprise	Support	Alibaba Cloud	>_	۵.	Ä	EN 🤇	
C	ontainer Service - Kubernetes - stio Management /irtual Services Sateways	App Catalog - ack	-virtual-kubelet-autoscaler ack-virtua incubator Instal virtual kubelet autoscale	I-kubelet-a	utosca	ler										
• к	native Components	Description Par 1 - vintualKube	rameters letAutoscøler:							Deploy						
- c	Services onfiguration ConfigMaps	2 → image: 3 reposite 4 tag: 0.1	ory: registry.cn-beijing.c 1	liyuncs.com/acs/virtual-l	kubelet-autoscal	er-amd64				The applic versions. I Clusters p cluster.	cation is only availa For clusters using page and click Upg	able to Ku Kubernet rade Clu	ubernete tes 1.8.1 ster to u	es 1.8.4 a , go to th ipgrade th	nd later le he	
	Secrets									Cluster					•	
<b>▼</b> M	arketplace									Namespace kube-system	<b>;</b> 1				•	
	Docker Images									Release Na	me					
	Orchestration Templa									ack-virtual-	-kubelet-autoscale	r				2
	App Catalog											Create			88	
3	Service Catalog															
Q	uick Start															
	1.png	^												显;	示全部	×

4. Run the following command to check the running status of the ack-virtual-kubelet-autoscaler addon:

kubectl get deploy -n kube-system

<pre>shell@Alicloud:~\$ kubectl get deploy -n kube-sys</pre>	stem			
NAME	READY	UP-TO-DATE	AVAILABLE	AGE
ack-virtual-node-affinity-admission-controller	1/1	1	1	4d
ack-virtual-node-controller	1/1	1	1	4d
alicloud-application-controller	1/1	1	1	6d16h
alicloud-disk-controller	1/1	1	1	6d16h
alicloud-monitor-controller	1/1	1	1	6d16h
aliyun-acr-credential-helper	1/1	1	1	6d16h
coredns	2/2	2	2	6d16h
metrics-server	1/1	1	1	6d16h
nginx-ingress-controller	2/2	2	2	6d16h
tiller-deplov	1/1	1	1	6d16h
virtual-kubelet-autoscaler	1/1	1	1	101m

#### Example

#### View the resource usage of physical nodes

Log on to the Container Service console to view the nodes of your Kubernetes cluster. In this example, two Elastic Compute Service (ECS) instances work as physical nodes, as shown in the following figure. The type of the two ECS instances is ecs.c5.large, indicating that each instance is allocated 2 vCPUs and 4 GiB memory. For more information about the instance type, see the c5, compute optimized instance family section in Compute optimized instance families. No pods are scheduled to the virtual-kubelet node, which is a virtual node.

≡	E C-D Alibaba (	Clou	bu	Account's all R 👻 🧬	Global	Q Sea	arch		Bill	ing Ticket	ICP Enterpris	se Support	Alibaba Cloud	⊾ ů.	ÌĦ EN	0
	Container Service - Kubernetes -		Noc	des							Re	fresh Manag	ge Labels E	xpand A	dd Existing Noo	le
	Overview		S C	onvert an instance from pay-a	is-you-go to subsc	ription-based &	Troubleshoot	t node exceptions	S Node mon	itoring and alerts	S Collect Kuber	netes Diagnostics I	nformation 🔗 Re	move a node		
•	Cluster		Cluste	iajun-ack \$ Node F	All Node F	Pools \$	C Filte	er by Label 👻					Name \$	earch by name	0	2
	Cluster			Instance ID/Name/IP Address	Bole/Status	Configuration	Pod (Allocated)	CPU Request/Limit	CPU Usage	Memory Request/Limit	Memory Usage	Kubelet Version	Updat	ted At	Ac	tions
	Nodes				Worker	Conngulation	() modulou)	noquoo Enni	on o obligo	rioquoos anne	monory obago		opua			
	Persistent Volumes Namespace			cn-beijing.192.1 192.168.0.172 i-2ze8mr535xgiupzpfd8q	Running	Pay-As-You-Go ecs.sn2ne.large 2 vCPU 8 GiB	11	25.00% 120.00%	-	7.67% 10.94%	28.80 %	v1.16.6-aliyun.1	May 11, 2020, 1	1:03:22 UTC+8	Monitor   Mo	ore 🕶
•	Authorizations Applications	11		cn-beijing.192.1 192.168.0.173 i-2ze8mr535xgiupzpfd8p	Worker Running Schedulable	Pay-As-You-Go ecs.sn2ne.large 2 vCPU 8 GiB	11	37.50% 440.00%	<b>L</b>	11.89% 170.61%	33.73 %	v1.16.6-aliyun.1	May 11, 2020, 1	1:03:22 UTC+8	Monitor   Mo	ore 🕶
	Deployments			virtual-node-eci 172.20.0.68	Worker Running		2	0.00%	<b>~</b>	0.00%		v1.16.6-aliyun.1	May 11, 2020, 1	1:12:37 UTC+8	Monitor   Mo	→ 91C
	December 2010			virtual-node-eci-0	Schedulable											
	Jobs			Batch Remove Set	to Unschedulable	Set to Sch	nedulable					Total: 3 item(s), Pe	r Page: 25 🔶 ite	em(s) «	← 1 →	19
	Cron Jobs															
	Deate															

#### Create a deployment

Prepare the YAML file of the deployment. This topic uses the deployment-autoscaler.yaml file as an example. The deployment has 10 replicas, and 2 vCPUs and 4 GiB memory need to be allocated to the container of each replica.

```
apiVersion: apps/vlbeta2
kind: Deployment
metadata:
  name: nginx-deployment-autoscaler
  labels:
   app: nginx
spec:
  replicas: 10
  selector:
   matchLabels:
     app: nginx
  template:
    metadata:
     labels:
       app: nginx
    spec:
     containers:
      - name: nginx
       image: registry-vpc.cn-beijing.aliyuncs.com/eci open/nginx:alpine
       ports:
        - containerPort: 80
        resources:
         requests:
           cpu: 2
            memory: 4Gi
```

Run the following command to create the deployment based on the preceding YAML file:

```
kubectl create -f deployment-autoscaler.yaml
```

Alternatively, log on to the Container Service console. Choose Applications > Deployments in the leftside navigation pane. On the page that appears, click Create from Template in the upper-right corner to create the deployment.

#### Best Practices Schedule pods to a v irtual node through the virtual-kubel et-autoscaler add-on

😑 🕞 Alibaba Clo	oud Account's all R 👻 🧭 🖉	Global	Q Search		Billing	Ticket	ICP	Enterprise	Support	Alibaba Cloud	۶_	Ō. È	EN	0
Container Service - Kubernetes -	Deployments								Refresh	Create from	mage	Create	rom Templat	to
Overview	$\mathscr{S}$ How to use private images $\mathscr{S}$	Create applications	Schedule a pod to a spe	cific node 🔗 Create a lay	er-4 Ingress	S Create	a layer-7	ingress 🔗 🤇	onfigure Poo	d auto scaling 🔗	Monitor co	ontainers		
▼ Cluster	S Blue-green release													
Cluster	Cluster jiajun-ack \$ Namespa	default	÷ 3								Search by	/ name	(	2
Nodes	Name	Label	Pods	Image			Created	d At					Act	ions
Persistent Volumes			0 0	Could not find any re	cord that	met the	conditio	on.						
Namespace			0											
Authorizations														
<ul> <li>Applications</li> </ul>														
Deployments														
StatefulSets														
DaemonSets														
Jobs														
Cron Jobs														
Pods														
Persistent Volume Cl														
Releases														
Workflows														
😑 🕞 Alibaba Clo	Dud Account's all R 👻 🖉 (	Global	Q Search		Billing	Ticket	ICP	Enterprise	Support	Alibaba Cloud	<b>&gt;</b> _	۵. <i>i</i>	EN	0
Container Service - Kubernetes <del>-</del>	Oresta from Tomalata													
	Greate from Template													
Overview	The application is only available to	Kubernetes 1.8.4 an	d later versions. For clusters	using Kubernetes 1.8.1, g	o to the Clus	iters page a	and click U	Jpgrade Clust	er to upgrade	e the cluster.				
Overview Cluster	The application is only available to Cluster	Kubernetes 1.8.4 an	d later versions. For clusters	using Kubernetes 1.8.1, g	o to the Clus	sters page a	and click U	Jpgrade Clust	er to upgrad	e the cluster.				
Overview Cluster Cluster	The application is only available to Cluster	Kubernetes 1.8.4 an jiajun-ack	d later versions. For clusters	using Kubernetes 1.8.1, g	o to the Clus	iters page a	and click U	Jpgrade Clust	er to upgrade	e the cluster.				
Overview Cluster Cluster Nodes	The application is only available to Cluster Namespace	Kubernetes 1.8.4 an jiajun-ack default	d later versions. For clusters	using Kubernetes 1.8.1, g	o to the Clus	iters page a	and click U	Jpgrade Clust	er to upgrad ÷	e the cluster.				
Overview Cluster Cluster Nodes Persistent Volumes	The application is only available to Cluster Namespace Sample Template	Kubernetes 1.8.4 an jiajun-ack default Resource - basic	d later versions. For clusters	using Kubernetes 1.8.1, g	o to the Clus	ters page a	and click U	Jpgrade Clust	er to upgrad	e the cluster.				
Overview Cluster Cluster Nodes Persistent Volumes Namespace	The application is only available to Cluster Namespace Sample Template	Kubernetes 1.8.4 an Jiajun-ack default Resource - basic 1 jopt Ven 2 kind; C	d later versions. For clusters : Deployment : Copioyment : Copioyment	using Kubernetes 1.8.1, g	o to the Clus	ters page a	and click U	Jpgrade Clust	er to upgrad	e the cluster. Add De	oloyment			
Overview Cluster Cluster Nodes Persistent Volumes Namespace Authorizations	The application is only available to Cluster Namespace Sample Template Template	Kubernetes 1.8.4 an Jiajun-ack default Resource - basic 1 bpt Verz 2 kind: L 3 restadd 4 name: 5 - Label	d later versions. For clusters Deployment Ston: appS/V1 # for Ve Deployment C0: r g(nx-deployment-ba: IS:	using Kubernetes 1.8.1, g resions before 1.8.0 itc	o to the Clus	/vlbeta1	and click U	l/pgrade Ciust	er to upgrade + C +	e the cluster. Add De Use Existing	ployment			
Overview Cluster Cluster Nodes Persistent Volumes Namespace Authorizations	The application is only available to Cluster Namespace Sample Template Template	Kubernetes 1.8.4 an Jiajun-ack default Resource - basic 1 jptVerr 2 kind: C 3 metcadat 4 mome 5 lobel 6 opp 7 sspec:	d later versions. For clusters Deployment Clon: apps/v1 # for va Peployment : nginx_deployment-bac : nginx	using Kubernetes 1.8.1, g resions before 1.8.0 ric	b to the Clus	ters page a	and click U	¢	er to upgradu † C +	Add De Use Existing	ployment			
Overview Cluster Cluster Nodes Persistent Volumes Namespace Authorizations Applications Deployments	The application is only available to Cluster Namespace Sample Template Template	Kubernetes 1.8.4 an Jiajun-ack default Resource - basic 1 bpiVerr 2 kind: C 3 metadat 4 name 5 - label 6 opp 7 - spec: 8 repli 9 - sele 10 - mot	d later versions. For clusters	using Kubernetes 1.8.1, g ersions before 1.8.0 ic	o to the Clus	/vlbetal	and click U	l/pgrade Clust	er to upgrad ¢	e the cluster. Add De Use Existing	oloyment Femplate			
Overview Cluster Cluster Nodes Persistent Volumes Namespace Authorizations Capplications Deployments StatefulSets	The application is only available to Cluster Namespace Sample Template Template	Kubernetes 1.8.4 an jiajun-ack default Resource - basic 1 japt Ver- 2 kind: C 3 metadat 4 nadet 5 label 6 app 7 - spec: 8 repti 9 - selec 10 metadat 11 temp	d later versions. For clusters :Deployment sion: apps/v1 # for ver- to: :pployment-bas :s: : nginx + childebis :childebis: :childebis: :pginx + childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebis: :childebi	using Kubernetes 1.8.1, g instans before 1.8.0 itc	use apps/	/v1beta1	and click U	l/pgrade Ciust	er to upgrade	e the cluster. Add De Use Existing	ployment			
Overview       Cluster       Cluster       Nodes       Persistent Volumes       Namespace       Authorizations       Deployments       StatefulSets       DaemonSets	The application is only available to Cluster Namespace Sample Template Template	Kubernetes 1.8.4 an Jiajun-ack default Resource - basic 1 bpt Vene 2 kind: C 3 metador 4 name 5 label 6 opp 7 spec: 8 repti 9 sele 10 mot 11 c 12 temp 13 metador 13 metador 14 name	d later versions. For clusters Deployment ston: apps/v1 # for ve deployment sci: mginx-deployment-bas s: nginx kcos: 2 tch: chos: s: chos: chos: s: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos: chos:	using Kubernetes 1.8.1, g	use apps/	ters page a	and click U	¢	er to upgradu	e the cluster. Add De Use Existing	oloyment Template			
Overview Cluster Cluster Cluster Nodes Persistent Volumes Authorizations Authorizations Capologyments StatefulSets DaemonSets Jobs	The application is only available to Cluster Namespace Sample Template Template	Kubernetes 1.8.4 an jiajun-ack default Resource - basic 1 bptVern 2 kind: C 3 metcddt 4 name 5 lobel 6 opp: 7 spec: 8 repti 9 selee 10 mat 11 c 12 tempi 13 metcddt 14 hitti 16 ppt 17 #	d later versions. For clusters Deployment stan: apps/v1 # for vi eployment r ginx-deployment-bas r ginx icos: 2 crditaeloyments crditaeloyment crditaeloyment-bas r ginx icos: 2 crditaeloyment crditaeloyment-bas r ginx icos: 2 crditaeloyment crditaeloyment-bas r ginx icos r ginx r g	using Kubernetes 1.8.1, g Irsions before 1.8.0 ric	use apps/	/v1beta1	and click U	¢	er to upgradd ¢ ¢	e the cluster. Add De Use Existing	bloyment			
Overview       Cluster       Cluster       Nodes       Persistent Volumes       Namespace       Authorizations       Opployments       Stateful/Sets       DaemonSets       Jobs       Cron Jobs	The application is only available to Cluster Namespace Sample Template Template	Kubernetes 1.8.4 an jiajun-ack default Resource - basic 1 bpt Ven 2 kind: L 3 metadat 6 app 7 spec: 8 repli 9 selec 10 mat 11 met 6 app 7 spec: 8 repli 9 selec 12 temp 13 metadat 13 metadat 6 app 7 spec: 8 repli 9 selec 14 mat 15 metadat 10 metadat 10 metadat 10 metadat 10 metadat 10 metadat 11 metadat 12 temp 13 metadat 13 metadat 13 metadat 14 metadat 14 metadat 15 metadat 16 metadat 17 metadat 18 metadat 19 selec 10 metadat 10 metadat 10 metadat 10 metadat 10 metadat 10 metadat 11 metadat 12 temp 13 metadat 13 metadat 13 metadat 14 metadat 19 selec 10 metadat 10 metadat 10 metadat 10 metadat 10 metadat 10 metadat 11 metadat 10 m	d later versions. For clusters Deployment tion: apps/v1 # for vr peployment. S: rginx-deployment-bas (s: rginx-deployment-bas (s: rginx-deployment-bas (s: rodis-1 coss : 2 tor: coss : 2 tor: rodis-1 coss	using Kubernetes 1.8.1, g ersions before 1.8.0 ic	use opps.	ters page a	and dick U	¢	er to upgrade	Add De Use Existing	oloyment			
Overview       Cluster       Cluster       Nodes       Persistent Volumes       Namespace       Authorizations       Deployments       StatefulSets       DaemonSets       Jobs       Cron Jobs       Pods	The application is only available to Cluster Namespace Sample Template Template	Kubernetes 1.8.4 an Jiajun-ack default Resource - basic 1 bpl Verri 2 kinds [ 3 metadat 4 metadat 5 label 6 aop 7 - speci: 8 repli 9 selec 10 met 11 selec 11 selec 12 texpl 13 metadat 13 metadat 14 selec 14 selec 14 selec 15 selec 16 sept 17 selec 10 selec 10 selec 10 selec 11 selec 12 texpl 13 metadat 13 selec 14 selec 14 selec 14 selec 15 selec 16 sept 17 selec 18 selec 10 selec 10 selec 10 selec 10 selec 10 selec 11 selec 12 texpl 13 selec 14 selec 14 selec 15 selec 16 sept 17 selec 18 selec 19 selec 10	d later versions. For clusters :Deployment clon: apps/vl # for ve peployment-back :: : nginx-deployment-back :: : nginx :cosi 2 : cosi 2 : cosi 2 : coditabels: : npdisxlector: : codels: : codels:	using Kubernetes 1.8.1, g resions before 1.8.0 itc	use option	/vibetal /	name:ta	dpgrade Clust	+ + +	e the cluster. Add De Use Existing	bloyment			
Overview       Cluster       Cluster       Nodes       Persistent Volumes       Namespace       Authorizations       Opployments       StatefulSets       DaemonSets       Jobs       Pods       Persistent Volume CL	Create from template The application is only available to Cluster Namespace Sample Template Template	Kubernetes 1.8.4 an jiajun-ack default Resource - basic 1 jopt Vene 5 1 jopt Vene 5 1 jopt Vene 5 1 jopt Vene 5 1 jopt Vene 6 opp 7 spec: 8 rep1 9 sete 10 - sete 11 d 12 - temp1 13 - metadot 14 - et 15 - jopt Vene 14 - et 15 - jopt Vene 16 - spec 17 - spec: 18 - spec 18 - spec 19 - sete 10 - sete 10 - sete 10 - sete 11 - sete 12 - temp1 13 - metadot 14 - sete 15 - jopt Vene 16 - spec 16 - spec 17 - spec 18 - sete 10 - se	d later versions. For clusters :Deployment sion: apps/v1 # for vs sion: apps/v1 # for vs sion: apps/v1 # for vs sion: apps/v1 # for vs sion: apployment-back si: s: apinx tcos: 2 tcor: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: childebls: chi	using Kubernetes 1.8.1, g resions before 1.8.0 .tc	use opps,	/vibetal	name:ta	t) t)	t upgrad	e the cluster. Add De Use Existing	bloyment			
Overview           Cluster           Cluster           Nodes           Persistent Volumes           Authorizations           Opployments           StatefulSets           Jobs           Cron Jobs           Persistent Volume CL           Releases	Create from template The application is only available to Cluster Namespace Sample Template Template	Kubernetes 1.8.4 an jiajun-ack default Resource - basic 1 ppt Vernet 2 kindt C 3 metadat 9 sele 10 mot 11 c 12 temp 13 metadat 14 1 15 16 spt 17 # 18 # 19 20 - 21 22 23	d later versions. For clusters Deployment stor: apps/v1 # for ve deployment c: : nginx-deployment-bas :: : nginx c: : nginx c: : nginx :	using Kubernetes 1.8.1, g Irsions before 1.8.0 ic	use apps,	/vlbetal /	and click L	gs>	e e e	e the cluster. Add De Use Existing	bloyment			

#### Check the status of the pods

Run the following command to view the running status of pods:

kubectl get pods -o wide

<pre>shell@Alicloud:~\$ kubectl get pods</pre>				
NAME	READY	STATUS	RESTARTS	AGE
ack-node-pod	1/1	Running	0	24h
nginx-deployment-autoscaler-786876b6b-5qtw4	0/1	Pending	0	15s
nginx-deployment-autoscaler-786876b6b-5w7kg	0/1	Pending	0	15s
nginx-deployment-autoscaler-786876b6b-84t6w	0/1	Pending	0	15s
nginx-deployment-autoscaler-786876b6b-9b7xj	0/1	Pending	0	15s
nginx-deployment-autoscaler-786876b6b-9r155	0/1	Pending	0	15s
nginx-deployment-autoscaler-786876b6b-jwrjd	0/1	Pending	0	15s
nginx-deployment-autoscaler-786876b6b-jzhq6	0/1	Pending	0	15s
nginx-deployment-autoscaler-786876b6b-prcm7	0/1	Pending	0	15s
nginx-deployment-autoscaler-786876b6b-t9xzq	0/1	Pending	0	15s
nginx-deployment-autoscaler-786876b6b-w6xnj	0/1	Pending	0	15s
<pre>shell@Alicloud:~\$ kubectl get pods</pre>				
NAME	READY	STATUS	RESTARTS	AGE
ack-node-pod	1/1	Running	0	24h
nginx-deployment-autoscaler-786876b6b-5qtw4	1/1	Running	0	32s
nginx-deployment-autoscaler-786876b6b-5w7kg	1/1	Running	0	32s
nginx-deployment-autoscaler-786876b6b-84t6w	1/1	Running	0	32s
nginx-deployment-autoscaler-786876b6b-9b7xj	1/1	Running	0	32s
nginx-deployment-autoscaler-786876b6b-9r155	1/1	Running	0	32s
nginx-deployment-autoscaler-786876b6b-jwrjd	1/1	Running	0	32s
nginx-deployment-autoscaler-786876b6b-jzhq6	1/1	Running	0	32s
nginx-deployment-autoscaler-786876b6b-prcm7	1/1	Running	0	32s
nginx-deployment-autoscaler-786876b6b-t9xzq	1/1	Running	0	32s
nginx-deployment-autoscaler-786876b6b-w6xnj	1/1	Running	0	32s

You can also run the following command to view the running status of a specific pod:

kubectl describe pod nginx-deployment-autoscaler-786876b6b-5qtw4

Evente:				
Type	Reason	Age	From	Message
Warning t cpu, 2	FailedScheduling Insufficient memory.	2m37s	default-scheduler	0/3 nodes are available: 1 node(s) had taints that the pod didn't tolerate, 2 Insufficien
Normal let	Scheduled	2m30s	default-scheduler	Successfully assigned default/nginx-deployment-autoscaler-786876b6b-5qtw4 to virtual-kube
Normal	Pulling	2m14s	kubelet, eci	pulling image "registry-vpc.cn-beijing.aliyuncs.com/eci_open/nginx:alpine"
Normal	SuccessfulMountVolume	2m14s	kubelet, eci	MountVolume.SetUp succeeded for volume "default-token-lwzln"
Normal	Pulled	2m12s	kubelet, eci	Successfully pulled image "registry-vpc.cn-beijing.aliyuncs.com/eci_open/nginx:alpine"
Normal	Created	2m12s	kubelet, eci	Created container
Normal	Started	2m12s	kubelet, eci	Started container

As shown in the preceding figure, the pod is scheduled to the virtual-kubelet node through the virtual-kubelet-autoscaler add-on because computing resources are insufficient on physical nodes.

Check the nodes of the Kubernetes cluster in the Contain Service console. As shown in the following figure, 10 pods are scheduled to the virtual-kubelet node.

☰ (-)阿里云	账	号全音	『资源 ▼ 🛛 🚱 全球	Q	搜索文档、控制	台、API、解决方案	和资源	费用	工单 省	备案 企业	支持与服务 官网 上	û <i>≒</i> ĝ	简体中文 📀
<		节,	点列表								刷新 标签	管理集群扩容	添加已有节点
基本信息		8	按量付费转包年包月 🔗 节点失期	美怎么办 🔗	节点监控及告警		3 诊断信息						
节点列表		集群	* 标签过滤	•									
事件列表			实例ID/名称/IP地址	角色/状态	配置	容器组 (已分配量)	CPU 请求/限制	CPU 使用量	内存 请求/限制	内存使用量	Kubelet 版本	更新时间	操作
集群审计 集群审计 new 在考试			i-2ze1ljawfkjghin3qfgo worker-k8s-for-c 192.168.6.30	Worker ●运行中 可调度	按量付费 ecs.c5.large	13	45.00% 355.00%	13.00 %	25.33% 362.27%	86.85 %	v1.14.8-aliyun.1	2019-12-10 19:26:48	监控│更多▼
	ļ	0	i-2ze1ljawfkjghin3qfgp worker-k8s-for-c 192.168.6.31	Worker ● 运行中 可调度	按量付费 ecs.c5.large	7	10.00% 55.00%	4.55 %	3.17% 27.18%	82.64 %	v1.14.8-aliyun.1	2019-12-10 19:26:55	监控 更多▼
			virtual-kubelet 172.20.0.165	Worker ● 运行中 可调度		10	0.03% 0.00%		0.06% 0.00%		v1.11.2-aliyun-dev	2019-12-13 11:30:45	监控 更多▼
			批量移除节点维护	节点上线	戋						共有3条,每页显	示: 25 \$ 条 ∝ 。	1 > 3

## 4.Use an elastic container instance to run jobs

You can run short-term jobs on elastic container instances to improve resource utilization and reduce computing costs. This topic describes how to use an elastic container instance to run jobs.

Many Kubernetes clusters must concurrently support a variety of online and offline workloads. The traffic volume of online workloads fluctuates and the amount of time required to complete offline workloads is unpredictable, which causes resource demands to vary with time. For example, many enterprises perform intensive computing on weekends and in the middle and end of each month, and their demands on computing resources increase sharply during these periods.

Typically, a Kubernetes cluster uses an autoscaler to scale out temporary nodes until all pods are scheduled. It takes about 2 minutes to deploy a temporary node. After the pods are scheduled and their execution is complete, the temporary nodes are automatically released. In this scale-out mode, a pod must wait 2 or more minutes before it can be scheduled.

In this scenario, we recommend that you use elastic container instances to run jobs. You can connect elastic container instances to Kubernetes clusters by deploying virtual nodes. Elastic container instances can be started within seconds and scaled out on demand to make Kubernetes clusters more elastic. You do not need to estimate the traffic volume of your business or reserve idle resources before you use elastic container instances to run jobs. This ensures that your business needs are met and reduces your use and O&M costs.

• If you use Container Service for Kubernetes (ACK) clusters, you must deploy virtual nodes within the clusters. Then, you can create elastic container instances on the virtual nodes to run jobs.

For more information, see Use an elastic container instance to run a job.

• If you use Serverless Kubernetes (ASK) clusters, you can directly use elastic container instances to run jobs.

For more information, see Use ASK to run jobs.

• If you use self-managed Kubernetes clusters on the cloud or in data centers, you can deploy virtual nodes within the clusters. Then, you can create elastic container instances on the virtual nodes and schedule jobs to the elastic container instances. For information about how to use elastic container instances in self-managed Kubernetes clusters, see Overview.

#### ? Note

You can also use preemptible elastic container instances to run jobs at reduced costs. For more information, see Run jobs on a preemptible instance.

## 5.Run jobs on a preemptible instance

You can run jobs or stateless applications on preemptible elastic container instances at reduced costs. This topic describes how to run jobs on a preemptible elastic container instance.

#### **Background information**

Before you use a preemptible elastic container instance, take note of the following information:

- The market price of a preemptible instance fluctuates based on the supply and demand for its instance type. When you create a preemptible instance, you must specify a maximum hourly price to bid for a specific instance type. If your bid price is higher than the current market price and the resource inventory of the instance type is sufficient, the requested preemptible instance is created and billed at the current market price.
- A preemptible instance has a protection period of 1 hour after it is created. After the one-hour protection period of a preemptible instance expires, the instance is automatically released if the current market price of the preemptible instance exceeds the specified maximum hourly price or if resources are insufficient. A Kubernetes event occurs 3 minutes before a preemptible instance is released. During these 3 minutes, you can perform specific operations to prevent your services from being interrupted. For example, you can configure to deny inbound traffic to the instance.
- If a preemptible instance is released because the market price exceeds your bid price, the data on the instance is cleared, but the instance information is retained. In addition, the status of the instance changes to Failed and the cause of the failure is BidFailed. We recommend that you use persistent storage services such as cloud disks or Apsara File Storage NAS to store important data.

For more information, see Overview and Create a preemptible instance.

#### **Configuration methods**

You can configure annotations in the pod declaration to specify to create a preemptible elastic container instance. Take note of the following items:

- k8s.aliyun.com/eci-spot-strategy: specifies the bidding policy for the preemptible instance. Valid values:
  - SpotAsPriceGo: The current market price is automatically used as the bid price.
  - SpotWithPriceLimit: You must set the maximum hourly price that you are willing to pay for the preemptible instance.
- k8s.aliyun.com/eci-spot-price-limit: The maximum hourly price of the preemptible instance. This value can be accurate to three decimal places. You must specify thk8s.aliyun.com/eci-spot-price-limit when k8s.aliyun.com/eci-spot-strategy is set to SpotWithPriceLimit.

#### ? Note

When you create a preemptible elastic container instance, we recommend that you set k8s.aliyun.com/eci-use-specs to specify multiple ECS instance types to ensure that resources are sufficient to create the instance.

#### Example:

#### • SpotAsPriceGo

In this mode, the system places bids based on the current market price.

```
apiVersion: v1
kind: Pod
metadata:
    name: spot-as-price-go
    annotations:
        k8s.aliyun.com/eci-use-specs: ecs.snlne.large,2-4Gi
        # If you set the k8s.aliyun.com/eci-spot-strategy annotation to SpotAsPriceGo, you do
not need to specify a maximum hourly price. The instance is priced at the market price at
the time of purchase.
        k8s.aliyun.com/eci-spot-strategy: SpotAsPriceGo
```

#### • Spot With Price Limit

In this mode, the system places bids based on the specified maximum hourly price. The following situations may occur:

- If your bid price is lower than the market price, the preemptible instance enters the Pending state. The system offers a bid price every 5 minutes. When your bid price is equal to or higher than the market price, the preemptible instance is created.
- If your bid price is higher than or equal to the market price and resources are sufficient, your preemptible instance is created.

```
apiVersion: v1
kind: Pod
metadata:
  name: spot-with-price-limit
  annotations:
    k8s.aliyun.com/eci-use-specs: ecs.snlne.large
    k8s.aliyun.com/eci-spot-strategy: SpotWithPriceLimit
    # If you set the k8s.aliyun.com/eci-spot-strategy annotation to SpotWithPriceLimit, y
 ou must specify a maximum hourly price. If your bid price is higher than the current pay-
    as-you-go price of the elastic container instance, the preemptible instance is created an
    d billed on a pay-as-you-go basis.
       k8s.aliyun.com/eci-spot-price-limit: "0.05"
```

#### **Configuration examples**

In Kubernetes, jobs are used to batch process short-lived and one-off tasks at a time. The following section provides an example of how to run jobs on a preemptible instance:

1. Prepare the configuration file of a job and name it spot\_job.yaml.

apiVersion: batch/v1 kind: Job metadata: name: pi spec: template: metadata: annotations: k8s.aliyun.com/eci-use-specs: ecs.t5-c1m2.large,2-4Gi k8s.aliyun.com/eci-spot-strategy: SpotAsPriceGo #The current market price is automatically used as the bid price. spec: containers: - name: pi image: registry-vpc.cn-beijing.aliyuncs.com/ostest/perl command: ["perl", "-Mbignum=bpi", "-wle", "print bpi(2000)"] restartPolicy: Never

#### 2. Create a job.

kubectl create -f spot\_job.yaml

3. View the running status.

kubectl get pod

The command output shows that the corresponding pod is created. Example output:

NAMEREADYSTATUSRESTARTSAGEpi-frmr81/1Running05s

A preemptible instance has a protection period of 1 hour after it is created. During this period, the instance is not released even if your bid price is lower than the current market price. This ensures that business can run properly on the instance. After the one-hour protection period of a preemptible instance expires, the instance is automatically released if the current market price of the preemptible instance exceeds the specified maximum hourly price or if resources are insufficient. You are notified of the release in the following ways:

• Event notifications before release

Elastic Container Instance sends a release event to the list of Kubernetes events 3 minutes before the instance is released. You can run the following commands to view the event:

View pod details

```
kubectl describe pod pi-frmr8
```

Information about the release event is displayed in the Events section of the command output. Example output:

 Events:
 Type
 Reason
 Age
 From
 Message

 ---- ---- ---- ---- ---- 

 Warning
 SpotToBeReleased
 3m32s
 kubelet, eci
 Spot ECI will be released in 3 minute

 s
 ---- ---- ---- ----- 

#### • View event details

```
kukubectl get events
```

Information about the release event is displayed in the command output. Example output:

LAST SEEN	TYPE	REASON	OBJECT	MESSAGE
3m39s	Warning	SpotToBeReleased	pod/pi-frmr8	Spot ECI will be released in 3
minutes				

• Status displayed after release

After a preemptible instance is released, the instance information is still retained. The status of the instance is changed to Failed and the cause of the failure is BidFailed. You can run the following commands to view the instance status:

View pod information

kubectl get pod

The status of the pod is displayed as BidFailed in the command output. Example output:

NAME	READY	STATUS	RESTARTS	AGE
pi-frmr8	1/1	BidFailed	0	3h5m

• View pod details

kubectl describe pod pi-frmr8

The status of the pod is displayed as Failed, and the reason is displayed as BidFailed in the command output. Example output:

Status:	Failed
Reason:	BidFailed
Message:	The pod is spot instance, and have been released at 2020-04-08T12:3
6Z	

After the instance is released, Job Controller in Kubernetes creates new instances to run jobs. You can run the kubectl get pod command to view the pod. Example output:

NAME	READY	STATUS	RESTARTS	AGE
pi-frmr8	1/1	BidFailed	0	4h53m
pi-kp5zx	1/1	Running	0	3h45m

## 6.Use GitLab CI

This topic describes how to install and register GitLab runners in a Kubernetes cluster and add a Kubernetes executor to build an application. This topic also provides step-by-step examples to implement a continuous integration (CI)/continuous delivery (CD) pipeline that includes stages, such as source code compilation, image build and push, and application deployment. For more information, see Use GitLab CI to run a GitLab runner and run a pipeline on Kubernetes.

## 7.Deploy Jenkins in an ASK cluster and build an application delivery pipeline

This topic describes how to deploy a Jenkins continuous integration environment in a serverless Kubernetes (ASK) cluster. This topic also provides step-by-step instructions on how to build an application delivery pipeline that includes source code compilation, image building and pushing, and application deployment.

#### Prerequisites

- An ASK cluster is created. For more information, see Create an ASK cluster.
- A kubectl client is connected to the cluster. For more information, see Connect to an ACK cluster by using kubectl.

#### **Deploy Jenkins**

1. Run the following command to download the Jenkins package:

git clone https://github.com/AliyunContainerService/jenkins-on-serverless.git
cd jenkins-on-serverless

2. Store the jenkins\_home directory in persistent storage.

If you require persistent storage, you can mount an nfs volume to the jenkins\_home directory. You must modify the *serverless-k8s-jenkins-deploy.yaml* file to uncomment the following lines and set Network File System (NFS) parameters:

```
#volumeMounts:
    # - mountPath: /var/jenkins_home
    # name: jenkins-home
    .....
    #volumes:
    # - name: jenkins-home
    # nfs:
    # path: /
    # server:
```

3. Run the following command to deploy Jenkins:

kubectl apply -f serverless-k8s-jenkins-deploy.yaml

```
4. Log on to Jenkins.
```

i. ii. iii.

iv.

v. Select Jenkins and click its external endpoint to log on to Jenkins.

Cluster:hfx-k8s / Namespace: default - / Services									cumentation
Search by name	Q							Refresh	Create
Name	Labels	Туре	Created At	Cluster IP	Internal Endpoint	External Endpoint			Actions
jenkins	app:jenkins component:jenkins-master	LoadBalancer Monitoring information	Aug 17, 2020, 14:41:16 UTC+8		jenkins:8080 TCP jenkins:30783 TCP	10.00	Details   Update	View in YAML	Delete
jenkins-agent	app:jenkins component:jenkins-master	LoadBalancer Monitoring information	Aug 17, 2020, 14:41:16 UTC+8		jenkins-agent:50000 TCP jenkins-agent:30530 TCP	)	Details   Update	View in YAML	Delete

vi. On the Jenkins logon page, enter the username and password. The default username and password are both **admin**.

Notice To ensure system security, change the password after you log on to Jenkins.

- 5. Configure the settings on the **Getting Started** page.
  - i. On the Getting Started page, click Install suggested plugins.
  - ii. After the plug-ins are installed, click **Save and Finish** in the **Instance Configuration** section of the **Getting Started** page.
  - iii. Click Start using Jenkins.
- 6. Obtain the Secret that contains a token.
  - i. Run the following command to query the token:

kubectl get secret

Expected output:

```
NAMETYPEDATAAGEack-jenkins-sa-token-q****kubernetes.io/service-account-token328mdefault-token-b****kubernetes.io/service-account-token327h
```

ii. Run the following command to obtain the Secret named ack-jenkins-sa-token-q\*\*\*\*:

```
kubectl get secret ack-jenkins-sa-token-q**** -o jsonpath={.data.token} |base64 -d
```

Expected output:

sdgdrh\*\*\*\*

- 7. Create a credential of the Secret text type.
  - i. In the left-side navigation pane of the Jenkins dashboard, click Manage Jenkins.
  - ii. On the Manage Jenkins page, click Manage Nodes and Clouds in the System Configuration section.
  - iii. In the left-side navigation pane of the Nodes page, click Configure Clouds.
  - iv. On the Configure Clouds page, click Kubernetes Cloud details....
  - v. In the **Credentials** field, choose **Add > Jenkins**.

vi. In the **Jenkins Credentials Provider: Jenkins** dialog box, add a credential of the Secret text type.

Parameter	Description
Domain	The domain of the credential. Default value: <b>Global credentials (unrestricted)</b> .
Kind	The kind of the credential. In this example, <b>Secret text</b> is selected.
Scope	The scope of the credential. Valid values: Global and System. In this example, <b>Global (Jenkins, nodes, items, all child items, etc)</b> is selected.
Secret	The Secret of the credential. In this example, the Secret obtained from the previous step is entered.
ID	The name of the credential. In this example, <b>ask-jenkins-token</b> is entered.
Description	The description of the credential.

The following table describes the credential parameters.

- vii. Click Add.
- 8. On the **Configure Clouds** page, set the required parameters. For more information, see **Configure** Kubernetes Cloud.
  - i. Set the **Kubernetes address**, set **Credentials** to **ask-jenkins-token**, and then click **Connection test** to check the connectivity between Jenkins and the cluster.
  - ii. Set the Jenkins address and Jenkins channel parameters.
  - iii. Click Save.
- 9. Build a pipeline named demo-pipeline and access the pipeline.
  - i. On the Jenkins homepage, click the 🔊 icon to the right side of **demo-pipeline**.

🏘 Jenkins	<b>♀</b> 査扱		?	<b>4</b> 1 😲 1	💄 admin	→ logout
Dashboard >						
Create a new task	dd a descripti	ion				
A The list of users S	In	The name is	Last successful	Last failed	Last duration	
Build history	) 🔆	demo- pipeline	No	not	not	Ø
System management lcor	n: Small medi je	um to	legend	ন Atom feed al	I Atom	feed failed
				رو	The fatest build of	Atom reed

ii. Modify the build parameters based on information about your image repository. In this example, the branch of the source code repository is master and the image is registry.cn -beijing.aliyuncs.com/ack-cicd/ack-jenkins-demo:latest .

🧌 Jenkins	Q Find	?	1 上 admin	<b> </b>
Dashboard > demo-pipeline >				
1 Return to the workbench	Pipeline der	no-pipeline		
🔍 state	- The following parameters are	e required to build the project:		
훋 Change history	image_region			
Suild with Parameters	cn-hangzhou			
disposition	image_namespace			
S Delete Pipeline	haoshuwei24			
🔍 Full stage view	jenkins-demo			
📄 rename	image_tag			
O Pipeline syntax	latest			
	branch			
Build History Build history	master			
find X	Start building			

#### iii. Click Start building.

Test the connectivity between the Jenkins master and the Jenkins slave pod that is dynamically allocated by the Kubernetes cluster.

After you click Build, Jenkins dynamically creates a slave pod in the Kubernetes cluster to run the build job. For more information about the sample application code, see jenkins-demo-Git Hub or jenkins-demo-haoshuwei.

iv. In the left-side navigation pane, click **Status**. If the build job is successful, Jenkins runs as normal on Kubernetes.



#### What's next

- For more information about how to configure the Maven cache for slave pods, see Configure the Maven cache for slave pods.
- For more information about how to use kaniko to build and push container images, see Use kaniko to build and push container images.

## 8.Install and use WordPress

**Cloud Shell** 

**?** Note You can clone the sample code for this topic to Cloud Shell and learn how to manage an ECI in Cloud Shell. Click the preceding link to open Cloud Shell.

#### Create a serverless Kubernetes cluster

For more information, see Create a serverless Kubernetes cluster.

#### Install WordPress

Notice Make sure that the serverless Kubernetes cluster created in the preceding step is initialized before you install WordPress. Typically, the initialization takes about 3 to 5 minutes.

Access the serverless Kubernetes cluster in Cloud Shell.

source use-k8s-cluster \${Cluster ID}

Install WordPress by using a YAML file.

kubectl apply -f wordpress-all-in-one-pod.yaml

Check the installation progress. Wait until the status of the pod where WordPress is deployed turns to Running.

kubectl get pods

Query the Elastic IP Address (EIP) of the pod.

kubectl get -o json pod wordpress |grep "k8s.aliyun.com/allocated-eipAddress"

By default, a security group does not allow access through port 80. You must enable access through port 80 for the security group automatically created for the serverless Kubernetes cluster.

Find the ID of the recently created security group whose name is prefixed with alicloud-cs-autocreated .

aliyun ecs DescribeSecurityGroups|grep -B 1 'alicloud-cs-auto-created'|head -1

Enable access through port 80 for the security group.

```
aliyun ecs AuthorizeSecurityGroup --RegionId cn-chengdu --SecurityGroupId ${Security group
ID} --IpProtocol tcp --PortRange 80/80 --SourceCidrIp 0.0.0.0/0 --Priority 100
```

#### Use WordPress

Enter the EIP that is retrieved in the preceding step in the address bar of a browser and press Enter to use WordPress.

## 9.Build a Spark application

This topic describes how to use Alibaba Cloud Serverless Kubernetes(ASK) and Elastic Container Instance to build a Spark application.

#### **Background information**

Apache Spark is an open source program that is widely used to analyze workloads in scenarios such as big data and machine learning. You can use Kubernetes to run and manage resources on Apache Spark 2.3.0 and later.

Kubernetes Operator for Apache Spark is designed for running Spark jobs in Kubernetes clusters. It allows you to submit Spark tasks that are defined in custom resource definition (CRD) files to Kubernetes clusters. Kubernetes Operator for Apache Spark provides the following benefits:

- Compared with open source Apache Spark, Kubernetes Operator for Apache Spark provides more features.
- Kubernetes Operator for Apache Spark can be integrated with the storage, monitoring, and logging components in a Kubernetes cluster.
- Kubernetes Operator for Apache Spark supports advanced Kubernetes features such as disaster recovery and auto scaling. In addition, Kubernetes Operator for Apache Spark can also optimize resource scheduling.

#### Preparations

1. Create an ASK cluster.

Create an ASK cluster in the Container Service for Kubernetes console. For more information, see Create an ASK cluster.

? Note

If you want to pull an image from the Internet or if your training jobs need to access the Internet, you must configure a network address translation (NAT) gateway.

You can use kubectl to manage and access the ASK cluster. Perform the following operations:

- If you want to manage the cluster from your on-premises computer, install and configure the kubectl client. For more information, see Connect to ACK clusters by using kubectl.
- You can also use kubectl to manage the ASK cluster on Cloud Shell. For more information, see Use kubectl to manage ACK clusters on Cloud Shell.
- 2. Create an OSS bucket.

You must create an Object Storage Service (OSS) bucket to store data, including the test data, test results, and test logs. For more information, see Create buckets.

#### Install Kubernetes Operator for Apache Spark

- 1. Install Kubernetes Operator for Apache Spark.
  - i. In the left-side navigation pane of the Container Service for Kubernetes console, choose Market place> Market place.
  - ii. On the App Catalog tab, search for and click ack-spark-operator.

iii. Click **Deploy**. Configure the parameters in the panel.

In the Parameters step, Set the sparkJobNamespace parameter to the namespace where you want to deploy the Spark job. The default value of this parameter is default. An empty string indicates that the Spark job monitors all namespaces.

2. Create a ServiceAccount, Role, and RoleBinding.

A Spark job needs a ServiceAccount to obtain the permissions to create pods. Therefore, you must create a ServiceAccount, Role, and RoleBinding. The following YAML example shows how to create a ServiceAccount, Role, and RoleBinding. Replace the namespaces with the actual values.

```
apiVersion: v1
kind: ServiceAccount
metadata:
name: spark
namespace: default
____
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
namespace: default
name: spark-role
rules:
- apiGroups: [""]
resources: ["pods"]
 verbs: ["*"]
- apiGroups: [""]
 resources: ["services"]
verbs: ["*"]
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
 name: spark-role-binding
 namespace: default
subjects:
- kind: ServiceAccount
 name: spark
 namespace: default
roleRef:
 kind: Role
 name: spark-role
  apiGroup: rbac.authorization.k8s.io
```

#### Build an image of the Spark job

You need to compile the Java Archive (JAR) package of the Spark job and use Dockerfile to package the image.

The following example shows how to configure Dockerfile when a Spark base image of ACK is used.

FROM registry.aliyuncs.com/acs/spark:ack-2.4.5-latest RUN mkdir -p /opt/spark/jars # If you want to read data from OSS or sink scheduled events to OSS, add the following JAR packages to the image. ADD https://repol.maven.org/maven2/com/aliyun/odps/hadoop-fs-oss/3.3.8-public/hadoop-fs-oss -3.3.8-public.jar \$SPARK\_HOME/jars ADD https://repol.maven.org/maven2/com/aliyun/oss/aliyun-sdk-oss/3.8.1/aliyun-sdk-oss-3.8.1 .jar \$SPARK\_HOME/jars ADD https://repol.maven.org/maven2/org/aspectj/aspectjweaver/1.9.5/aspectjweaver-1.9.5.jar \$SPARK\_HOME/jars ADD https://repol.maven.org/maven2/org/jdom/jdom/1.1.3/jdom-1.1.3.jar \$SPARK\_HOME/jars COPY SparkExampleScala-assembly-0.1.jar /opt/spark/jars

We recommend that you use a Spark base image provided by Alibaba Cloud. Alibaba Cloud provides the Spark 2.4.5 base image, which is optimized for resource scheduling and auto scaling in Kubernetes clusters and improves the scheduling and startup speeds. You can enable the scheduling optimization

feature by setting the enableAlibabaCloudFeatureGates variable in the Helm chart to true. If you

require a faster startup speed, you can set enableWebhook to false.



#### Build ImageCache

It takes a long time to pull a large Spark image. You can use ImageCache to accelerate image pulling. For more information, see Manage ImageCache and Use ImageCache to accelerate the creation of pods.

#### Write a Spark job template and submit a job

Create a YMAL configuration file for a Spark job and deploy the Spark job.

1. Create a spark-pi.yaml file.

The following code provides an example of a typical Spark job template. For more information, see spark-on-k8s-operator.

```
apiVersion: "sparkoperator.k8s.io/v1beta2"
kind: SparkApplication
metadata:
 name: spark-pi
 namespace: default
spec:
 type: Scala
 mode: cluster
  image: "registry.aliyuncs.com/acs/spark:ack-2.4.5-latest"
 imagePullPolicy: Always
 mainClass: org.apache.spark.examples.SparkPi
 mainApplicationFile: "local:///opt/spark/examples/jars/spark-examples 2.11-2.4.5.jar"
  sparkVersion: "2.4.5"
  restartPolicy:
   type: Never
  driver:
   cores: 2
   coreLimit: "2"
   memory: "3g"
   memoryOverhead: "1g"
    labels:
     version: 2.4.5
    serviceAccount: spark
    annotations:
      k8s.aliyun.com/eci-kube-proxy-enabled: 'true'
     k8s.aliyun.com/eci-image-cache: "true"
    tolerations:
    - key: "virtual-kubelet.io/provider"
      operator: "Exists"
  executor:
   cores: 2
   instances: 1
   memory: "3g"
   memoryOverhead: "1g"
   labels:
     version: 2.4.5
    annotations:
     k8s.aliyun.com/eci-kube-proxy-enabled: 'true'
     k8s.aliyun.com/eci-image-cache: "true"
    tolerations:
    - key: "virtual-kubelet.io/provider"
      operator: "Exists"
```

2. Deploy a Spark job.

kubectl apply -f spark-pi.yaml

#### Configure log collection

If you want to collect the standard output logs of a Spark job, you can configure the environment variables in the envVars field of the Spark driver and Spark executor. Then, logs are automatically collected. For more information, see Customize log collection for an elastic container instance.

#### envVars:

```
aliyun_logs_test-stdout_project: test-k8s-spark
aliyun_logs_test-stdout_machinegroup: k8s-group-app-spark
aliyun_logs_test-stdout: stdout
```

Before you submit a Spark job, you can configure the environment variables of the Spark driver and Spark executor as shown in the preceding code to implement automatic log collection.

1 Q	04-21 14:4 0:09	source_: 192119 tag_:hostname: tpcds-q1587450585853-exec-66 tag_:path: /var/log/eci/eg tag_:cuser_defined_id: ktp-spark tag_:eci_id: eci-bp1f5m4uma22 topic: content: 2020-04-21T14:40:09.800129005+08:00 stderr F 20/04/21 06:40:09 INFO ShutdownHookManager: Deleting directory /var/data/spark-e0cff89a-71b3-4 b2cb-dd13d08a9c14/spark-3659ccd5-fc14-4cd5-9d80-c118452d487d
2 Q	04-21 14:4 0:09	source_: 192.1
3 Q	04-21 14:4 0:09	source_: 192.16 9 tag : hostname : tpcds- ee-1587450588583-exec-66 tag : path_: / var/log/ec// log tag : user_defined.id : i app-spark tag_:eci_id: eci-bp1f5m4um.grammeg1y2 topic_: content: 2020-04-21T14:40:09.794608725+08:00 stderr F 20/04/21 06:40:09 INFO BlockManager: BlockManager stopped
4 Q	04-21 14:4 0:09	_source_: 192.16 119 tag_:_hostname_: tpcds-quer-1587450585853-exec-66 tag_:_path_: /var/log/ecl/exlog tag_:_user_defined_id_: k8 app-spark tag_:eci_id: eci-bp1f5m4uma1y2 topic_:

#### Configure a history server

A Spark history server allows you to review Spark jobs. You can set the SparkConf field in the CRD of the Spark application to allow the application to sink events to OSS. Then, you can use the history server to retrieve the data from OSS. The following code provides sample configurations:

```
sparkConf:
   "spark.eventLog.enabled": "true"
   "spark.eventLog.dir": "oss://bigdatastore/spark-events"
   "spark.hadoop.fs.oss.impl": "org.apache.hadoop.fs.aliyun.oss.AliyunOSSFileSystem"
   # oss bucket endpoint such as oss-cn-beijing.aliyuncs.com
   "spark.hadoop.fs.oss.endpoint": "oss-cn-beijing.aliyuncs.com"
   "spark.hadoop.fs.oss.accessKeySecret": ""
   "spark.hadoop.fs.oss.accessKeyId": ""
```

Alibaba Cloud provides a Helm chart for you to deploy Spark history servers. To deploy a Spark history server, log on to the Container Service for Kubernetes console, choose **Market place > Market place** in the left-side navigation pane. On the App Catalog tab, search for ack-spark-history-server for deployment. When you deploy the Spark history server, you must configure OSS information in the **Parameters** section. The following code provides sample configurations:

oss:
enableOSS: true
# Please input your accessKeyId
alibabaCloudAccessKeyId: ""
<pre># Please input your accessKeySecret</pre>
alibabaCloudAccessKeySecret: ""
<pre># oss bucket endpoint such as oss-cn-beijing.aliyuncs.com</pre>
alibabaCloudOSSEndpoint: "oss-cn-beijing.aliyuncs.com"
<pre># oss file path such as oss://bucket-name/path</pre>
<pre>eventsDir: "oss://bigdatastore/spark-events"</pre>

After you deploy the Spark history server, you can view its external endpoint on the Services page. Then, you can access the external endpoint to view the history of Spark jobs.

spark-2

#### View the result of the Spark job

1. View the execution status of the pod.

kubectl get pods

The following expected output is returned:

NAME	READY	STATUS	RESTARTS	AGE
spark-pi-1547981232122-driver	1/1	Running	0	12s
spark-pi-1547981232122-exec-1	1/1	Running	0	3s

2. View the real-time Spark user interface.

kubectl port-forward spark-pi-1547981232122-driver 4040:4040

#### 3. View the status of the Spark application.

kubectl describe sparkapplication spark-pi

#### The following expected output is returned:

Name:	spark-pi	
Namespace:	default	
Labels:	<none></none>	
Annotations:	kubectl.	<pre>subernetes.io/last-applied-configuration:</pre>
	{"apiVe	ersion":"sparkoperator.k8s.io/v1alpha1","kind":"SparkApplication"
,"metadata":{'	'annotatio	ons":{},"name":"spark-pi","namespace":"defaul
API Version:	sparkope	rator.k8s.io/vlalpha1
Kind:	SparkAppl	lication
Metadata:		
Creation Tim	nestamp:	2019-01-20T10:47:08Z
Generation:		1
Resource Ver	rsion:	4923532
Self Link:		/apis/sparkoperator.k8s.io/vlalphal/namespaces/default/sparkappl
ications/spar}	k-pi	
UID:		bbe7445c-1ca0-11e9-9ad4-062fd7c19a7b
Spec:		
Deps:		

```
Driver:
   Core Limit: 200m
   Cores: 0.1
   Labels:
    Version:
                  2.4.0
   Memory:
                  512m
   Service Account: spark
   Volume Mounts:
    Mount Path: /tmp
    Name: test-volume
 Executor:
            1
   Cores:
   Instances: 1
   Labels:
    Version: 2.4.0
   Memory: 512m
   Volume Mounts:
    Mount Path:
                    /tmp
    Name:
                     test-volume
 Image:
                     gcr.io/spark-operator/spark:v2.4.0
 Image Pull Policy:
                      Always
 Main Application File: local:///opt/spark/examples/jars/spark-examples_2.11-2.4.0.ja
r
 Main Class:
                     org.apache.spark.examples.SparkPi
 Mode:
                      cluster
 Restart Policy:
  Type: Never
 Type: Scala
 Volumes:
  Host Path:
    Path: /tmp
    Type: Directory
   Name: test-volume
Status:
 Application State:
  Error Message:
   State: COMPLETED
 Driver Info:
  Pod Name:
                     spark-pi-driver
  Web UI Port:
                     31182
   Web UI Service Name: spark-pi-ui-svc
 Execution Attempts:
                     1
 Executor State:
  Spark - Pi - 1547981232122 - Exec - 1: COMPLETED
 Last Submission Attempt Time: 2019-01-20T10:47:14Z
 Spark Application Id:
                                      spark-application-1547981285779
 Submission Attempts:
                                      1
 Termination Time:
                                      2019-01-20T10:48:56Z
Events:
 Type Reason
                                Age
                                                  From
                                                                Message
 _____
                                                                _____
                                ____
                                                  ____
Normal SparkApplicationAdded
                               55m
                                                  spark-operator SparkApplicati
on spark-pi was added, Enqueuing it for submission
 Normal SparkApplicationSubmitted 55m
                                                  spark-operator SparkApplicati
```

on spark-pi was submitted successful.	ТÀ		
Normal SparkDriverPending	55m (x2 over 55m)	spark-operator	Driver spark-p
i-driver is pending			
Normal SparkExecutorPending	54m (x3 over 54m)	spark-operator	Executor spark
-pi-1547981232122-exec-1 is pending			
Normal SparkExecutorRunning	53m (x4 over 54m)	spark-operator	Executor spark
-pi-1547981232122-exec-1 is running			
Normal SparkDriverRunning	53m (x12 over 55m)	spark-operator	Driver spark-p
i-driver is running			
Normal SparkExecutorCompleted	53m (x2 over 53m)	spark-operator	Executor spark
-pi-1547981232122-exec-1 completed			

4. View the log.

NAME		READY	STAT	US RESTARTS	AGE
spark-pi-1547981232122-driver	0/1	Completed	0	1m	

If the Spark application is in the Succeed state, or the Spark driver pod is in the Completed state, the result of the Spark job is available. You can check the result of the Spark job in the log.

```
kubectl logs spark-pi-1547981232122-driver
Pi is roughly 3.152155760778804
```

## 10.Run a GPU-based TensorFlow training job

This topic describes how to use a Serverless Kubernetes (ASK) cluster and an elastic container instance to run a GPU-based TensorFlow training job.

#### **Background information**

In recent years, artificial intelligence (AI) and machine learning have been widely applied in a large number of fields, and various training models have been developed. An increasing number of training jobs are run on the cloud. However, it is not easy to continuously run training jobs in a cloud environment. You may encounter the following difficulties:

- Difficulties in deploying environments: You must purchase a GPU-accelerated instance and install a GPU driver on the instance. After you prepare images for training jobs, you must install the GPU runtime hook.
- Lack of scalability: After you deploy an environment and run the training jobs, you may need to release idle resources to save costs. The next time you want to run training jobs, you must deploy an environment and create instances again. If compute nodes are insufficient, you must scale out the compute nodes. In this case, you must create instances and deploy the environment again.

To resolve the preceding difficulties, we recommend that you use ASK clusters and elastic container instances to run training jobs. This solution has the following benefits:

- You are charged on a pay-as-you-go basis and do not need to manage resources.
- You need only to prepare the configurations once. Then, you can reuse the configurations without limits.
- The image cache feature allows you to create pods and start training jobs in a more efficient manner.
- Training data is decoupled from training models. Training data can be persisted.

#### Preparations

1. Prepare a container image and training data for the training model.

In this example, a TensorFlow training job on GitHub is used. You can obtain the sample image eci/tensorflow from Alibaba Cloud Container Registry. For more information, see TensorFlow Model Garden.

2. Create an ASK cluster.

Create an ASK cluster in the Container Service for Kubernetes console. For more information, see Create an ASK cluster.

? Note

If you want to pull an image from the Internet or if your training jobs need to access the Internet, you must configure a network address translation (NAT) gateway.

You can use kubectl to manage and access the ASK cluster. You can use one of the following methods:

- If you want to manage the cluster from your computer, install and configure the kubectl client. For more information, see Use kubectl to connect to an ACK cluster.
- Use kubectl on Cloud Shell to manage the cluster. For more information, see Use kubectl on Cloud Shell to manage ACK clusters.
- 3. Create a network-attached storage (NAS) file system and add a mount target.

Create a NAS file system and add a mount target in the NAS File System console. For more information, see Manage file systems and Manage mount targets.

? Note

The NAS file system and the ASK cluster must be in the same VPC.

#### Create an image cache

The image cache feature has been integrated into Kubernetes CRD to accelerate the pulling of images. For more information, see Use an image cache CRD to accelerate pod creation.

Perform the following operations:

1. Prepare the YAML file.

The following code shows an example of the imagecache.yaml file:

```
apiVersion: eci.alibabacloud.com/v1
kind: ImageCache
metadata:
   name: tensorflow
spec:
   images:
    - registry-vpc.cn-beijing.aliyuncs.com/eci/tensorflow:1.0 # The image of the training
job. We recommend that you upload the image to Alibaba Cloud Container Registry. The en
dpoint of the VPC is used. Make sure that the VPC is the same as the one to which the c
luster belongs.
```

#### 2. Create an image cache.

kubectl create -f imagecache.yaml

You must pull an image when you create an image cache. The amount of time required to pull an image is related to the image size. You can run the following command to view the creation progress of the image cache:

Kubectl get imagecache tensorflow

A command output similar to the following one indicates that the image cache is created.

NAMEAGECACHEIDPHASEPROGRESStensorflow11mimc-2ze1xczztv7tgesg\*\*\*\*Ready100%

#### Create a training job

You can use the image cache to create a training job.

1. Prepare the YAML file.

The following code shows an example of the gpu pod.yaml file:

```
apiVersion: v1
kind: Pod
metadata:
 name: tensorflow
 annotations:
   k8s.aliyun.com/eci-use-specs: "ecs.gn6i-c4gl.xlarge" # Specify the GPU-accelerated
instance type that is used to create an elastic container instance. Example: k8s.aliyun
.com/eci-instance-type: "ecs.gn5i-c2g1.large"
   k8s.aliyun.com/eci-image-cache: "true"
                                                     # Enable Automatically Match Ima
ge Cache.
spec:
 containers:
  - name: tensorflow
   image: registry-vpc.cn-beijing.aliyuncs.com/eci/tensorflow:1.0 # The image of the t
raining job.
   command:
     - "sh"
      - "-c"
      - "python models/tutorials/image/imagenet/classify image.py" # The script used to
start the training job.
   resources:
     limits:
       nvidia.com/gpu: "1" # The number of GPUs required by the container.
   volumeMounts:
    - name: nfs-pv
     mountPath: /tmp/imagenet
  volumes:
  - name: nfs-pv  #Persist the training results to NAS files.
   flexVolume:
       driver: alicloud/nas
       fsType: nfs
       options:
         server: 16cde4***-ijv**.cn-beijing.nas.aliyuncs.com #The mount target of
the NAS file system.
         path: /
                         #The mount directory.
 restartPolicy: OnFailure
```

2. Run the following command to create a pod:

kubectl create -f gpu\_pod.yaml

#### 3. View the execution results.

#### You can view events or logs.

• View events

kubectl describe pod tensorflow

• View logs

kubectl logs tensorflow

#### View the results

You can view the results of the training job in the console.

• You can view the storage capacity occupied by the training data in the NAS File System console.

	File System ID/Name	Tags	File System Type	Storage Specifications	Usage/Total Capacity	Availability Zone	Billing Method	Data lifecycle management	Creation Time 🕸	Status	Protocol Type	Operations
+	16cdi 16cdi	•	General Purpose NAS	Capacity Type	176.84 MiB / 10 PiB	North China 2 usable area H	Pay-as-you-go	Enabled Configure policies	Feb 2, 2021, 17:55:39	🗸 Running	NFS	Monitoring   Managemen t   More

#### ? Note

Training results are stored in the configured NAS file system. After the NAS file system is mounted, you can obtain the results in the corresponding path.

• You can view the elastic container instance that is successfully run in the Elastic Container Instance console.

Container Group ID/Name	Status 🏆	Event	Specification	Zone 🖓	IP Address	Time	Security Group/VSwitch	Actions
☐ eci-2zebsrh4dmd ASK default tensorflow	✓ Succeeded	<b>0</b> 5	4 vCpu 15 GiB	Beijing Zone H	[] 192. 32 (Internal)	Create Instance: Feb 3, 2021, 18:05:55 Completed: Feb 3, 2021, 18:07:11	sg-2ze5f3( hgr1p     sodn5wu	Delete   Restart   Modify

#### ? Note

If an elastic container instance is successfully run, containers in the instance are stopped. Then, the system reclaims the underlying computing resources, and the billing of pods is stopped.

## 11.Access common data sources 11.1. Access data in OSS from ECI

Object Storage Service (OSS) is an optional data storage service for Hadoop and Spark. This topic describes how to create a file in OSS and access the file from a Spark application that runs in Elastic Container Instance (ECI).

#### Prepare data

#### 1. Create an OSS bucket.

Note: Storage Class a	nd Region cannot be changed after the bucket is created.	
Bucket Name	spark-on-k8s	12/63
Region	China (Beijing)	~
	Alibaba Cloud services in the same region can communicate with eac internal network. The region cannot be changed after the purchase. E when you select a region. You do not have any available storage or traffic packages in this regi that you purchase a package.	h other over an Exercise caution on. We recommend
Endpoint	oss-cn-beijing.aliyuncs.com	
Storage Class	Standard         IA         Archive           Standard: high-performance, reliable, and highly available storage class for deta that is forward account account in the storage storage class for a storage	ss. We recommend
	How to Choose a Suitable Storage Class	
Cone-redundant Storage	Enable Disable	
	OSS can back up your data to three zones within the same region to disaster recovery. Learn more.	provide data center
	Ø Zone-redundant storage improves the availability of data. This extra costs. For more information about the pricing of this fear price details. This feature cannot be disabled after it is enabled	feature incurs ture, visit d.

2. Upload a file to the OSS bucket.

You can use the OSS SDK to upload a file, or use HDP 2.6-based Hadoop to read and write OSS data. The following figure shows the result in the OSS console after the file is uploaded.

0	Object Storage Service / liumihus	st / I	Files									
I	liumihust		A	ccess (	Control List (ACL)	Private	Type Standa	ard(Locally Redunda	int Storage) Re	gion China (Han	gzhou)	Created At
	Overview		Uploa	d	Create Folder	Parts	Authorize	Batch Operation	<ul> <li>✓ Refresh</li> </ul>			Enter a f
	Files	>	Files						Size	Storage Class	Updated	At
	Access Control	>	Attach O	SS Bud	ket to ECS							
	Basic Settings	>	Scheduled Backup									
	Redundancy for Fault Toleranc	e>			IC-COROLLY							
	Transmission	>		ŧ	formation				4.06MB	Standard	January	14,
	Logging	>									2020, 1-	
	Data Processing	>			te				0.006KB	Standard	March 1 2020, 11	9, 7:12
	Data Statistics	>		•••	VIIT				0.006KB	Standard	March 1 2020, 18	9, 3:29
				P	RECT/ MEM		Alex Alex		7.328MB	Standard	Septemb 23, 2019 20:55	ber 9,

The endpoint of the bucket is oss://liumi-hust/test.txt,endpoint: oss-cn-hangzhouinternal.aliyuncs.com. Take note of the endpoint. Now the data is prepared.

#### Access data in OSS from a Spark application

1. Develop a Spark application.

You can develop a Spark application in the same way as that used in traditional deployment modes.

```
SparkConf conf = new SparkConf().setAppName(WordCount.class.getSimpleName());
JavaRDD<String> lines = sc.textFile("oss://liumi-hust/test.txt", 250);
...
wordsCountResult.saveAsTextFile("oss://liumi-hust/test-result");
sc.close();
```

2.Place the core-site.xml file in the resources directory of the application project.

```
<? xml version="1.0" encoding="UTF-8"? >
<? xml-stylesheet type="text/xsl" href="configuration.xsl"? >
<! --
 Licensed under the Apache License, Version 2.0 (the "License");
 you may not use this file except in compliance with the License.
 You may obtain a copy of the License at
   http://www.apache.org/licenses/LICENSE-2.0
 Unless required by applicable law or agreed to in writing, software
 distributed under the License is distributed on an "AS IS" BASIS,
 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 See the License for the specific language governing permissions and
 limitations under the License. See accompanying LICENSE file.
-->
<! -- Put site-specific property overrides in this file. -->
<configuration>
   <! -- OSS configuration -->
    <property>
        <name>fs.oss.impl</name>
        <value>org.apache.hadoop.fs.aliyun.oss.AliyunOSSFileSystem</value>
    </property>
    <property>
        <name>fs.oss.endpoint</name>
        <value>oss-cn-hangzhou-internal.aliyuncs.com</value>
    </property>
    <property>
        <name>fs.oss.accessKeyId</name>
        <value>{Temporary AccessKey ID used to connect to OSS}</value>
   </property>
    <property>
        <name>fs.oss.accessKeySecret</name>
        <value>{Temporary AccessKey secret used to connect to OSS}</value>
    </property>
    <property>
        <name>fs.oss.buffer.dir</name>
        <value>/tmp/oss</value>
    </property>
    <property>
        <name>fs.oss.connection.secure.enabled</name>
        <value>false</value>
    </property>
    <property>
        <name>fs.oss.connection.maximum</name>
        <value>2048</value>
    </property>
</configuration>
```

#### 3. Create a JAR package with all the dependencies included.

mvn assembly:assembly

The content of the pom.xml file of the Spark application is as follows:

```
1<? xml version="1.0" encoding="UTF-8"? >
```

3     mine scale************************************	2 <pr< th=""><th>oject xmlns="http://maven.apache.org/POM/4.0.0"</th></pr<>	oject xmlns="http://maven.apache.org/POM/4.0.0"
4 sitechematocation="bitp://maven.apache.org/POM/4.0.0 http://maven.apache.org/xs d/maven-4.0.0.xsd"> 6 6 7 7 7 7 7 7 7 7 8 8 8 8 8 9 9 9 9 9 9 9	3	<pre>xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"</pre>
<pre>d/muver.4.0.0.xad*&gt;</pre>	4	<pre>xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xs</pre>
scolelVersion>4.0.0       condelVersion>4.0.0       cartifactId>SparkExampleIavac/artifactId>       cartifactId>SparkExampleIavac/artifactId>       cartifactId>SparkExampleIavac/artifactId>       cartifactId>SparkExampleIavac/artifactId>       cartifactId>SparkExampleIavac/artifactId>       cartifactId>sparkCartifactId>       cartifactId=sparkCartifactId>       cartifactId=spark	d/mav	en-4.0.0.xsd">
<pre>6 croupld&gt;com.aliyun.liumi.spark 7 <pre>4 <artifactid>SparkExampledava</artifactid> 9 <version>1.0=NARSENT</version> 10 11 </pre> 12 </pre> 13  14  14  15  15  16  17  18  19  19  19  19  10  10  10  11  12  13  14  14  15  16  17  18  19  10  10  10  10  11  12  13  14  14  15  16  17  18  19  10  10  10  10  11  12  13  14  14  15  16  17  18  19  10  10  10  10  11  12  13  14  14  15  15  16  17  18  19  10  10  10  10  11  12  13  14  14  15  15  16  17  18  19  10  10  10  11  12  13  14  14  15  15  16  17  18  19  10  10  10  10  11  12  13  14  14  14  14  15  15  15  15  16  17  18  19  10  10  10  10  11  11  12  13  14  14  14  15  15  16  17  18  19  19  19  10  10  10  10  10  10  10  10  11  11  11  11  12  13  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  1	5	<modelversion>4.0.0</modelversion>
<pre> /* <group1d>con.aliyun.liumi.spark/group1d&gt; /* <artifact1d>byparkbxampleJavax/artifact1d&gt; /* <artifact1d>byparkbxampleJavax/artifact1d&gt; /* <dependencies> /* <dependencies> /* <dependency> /* <artifact1d>sparkb*.oore.2.12x/artifact1d&gt; /* <artifact< td=""><td>6</td><td></td></artifact<></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></artifact1d></dependency></dependencies></dependencies></artifact1d></artifact1d></group1d></pre>	6	
<pre>8 <artifactid>ParkExampleJava</artifactid> 9 <artifactid>ParkExampleJava</artifactid> 10 <artifactid>ParkExampleJava</artifactid> 11 <artifactid>ParkExampleJava</artifactid> 12 <artifactid>ParkExampleJava</artifactid> 13 <artifactid>ParkExampleJava</artifactid> 14 <artifactid>ParkExampleJava</artifactid> 15 <artifactid>ParkExampleJava</artifactid> 16 <artifactid>ParkExampleJava</artifactid> 17 <artifactid>ParkExampleJava</artifactid> 18 <artifactid>ParkExampleJava</artifactid> 19 <artifactid>ParkExampleJava</artifactid> 10 <artifactid>ParkExampleJava</artifactid> 11 <artifactid>ParkExampleJava</artifactid> 12 <artifactid=parkexamplejava< artifactid=""> 13 <artifactid=parkexamplejava< artifactid=""> 14 <artifactid=parkexamplejava< artifactid=""> 15 <artifactid=parkexamplejava< artifactid=""> 16 <artifactid=parkexamplejava< artifactid=""> 17 <artifactid=parkexamplejava< artifactid=""> 18 <artifactid=parkexamplejava< artifactid=""> 19 <artifactid=parkexamplejava< artifactid=""> 10 <artifactid=parkexamplejava< artifactid=""> 11 <artifactid=parkexamplejava< artifactid=""> 12 <artifactid=parkexamplejava< artifactid=""> 13 <artifactid=parkexamplejava< artifactid=""> 14 <artifactid=parkexamplejava< artifactid=""> 15 <artifactid=parkexamplejava< artifactid=""> 16 <artifactid=parkexamplejava< artifactid=""> 17 <artifactid=parkexamplejava< artifactid=""> 18 <artifactid=parkexamplejava< artifactid=""> 19 <artifactid=parkexamplejava< artifactid=""> 20 <artifactid=parkexamplejava< artifactid=""> 21 <artifactid=parkexamplejava< artifactid=""> 22 <artifactid=parkexamplejava< artifactid=""> 23 <artifactid=parkexamplejava< artifactid=""> 24 <artifactid=parkexamplejava< artifactid="ParkExampleJava&lt;/artifactId=ParkExampleJava&lt;/artifactId=ParkExampleJava&lt;/artifactId=ParkExampleJava&lt;/artifactId=ParkExampleJava&lt;/artifactId=ParkExampleJ&lt;/td"><td>7</td><td><groupid>com.aliyun.liumi.spark</groupid></td></artifactid=parkexamplejava<></artifactid=parkexamplejava<></artifactid=parkexamplejava<></artifactid=parkexamplejava<></artifactid=parkexamplejava<></artifactid=parkexamplejava<></artifactid=parkexamplejava<></artifactid=parkexamplejava<></artifactid=parkexamplejava<></artifactid=parkexamplejava<></artifactid=parkexamplejava<></artifactid=parkexamplejava<></artifactid=parkexamplejava<></artifactid=parkexamplejava<></artifactid=parkexamplejava<></artifactid=parkexamplejava<></artifactid=parkexamplejava<></artifactid=parkexamplejava<></artifactid=parkexamplejava<></artifactid=parkexamplejava<></artifactid=parkexamplejava<></artifactid=parkexamplejava<></artifactid=parkexamplejava<></pre>	7	<groupid>com.aliyun.liumi.spark</groupid>
9 <version>1.0-SNAPSHOT</version> 10        11        12 <dependencies>       13     <group1d>org.apache.spark</group1d>       14     <artifactid>spark<ore_2.12< artifactid="">       15     <version>2.4.3       16        17        18     <dependency>       19     <group1d>con.aliyun.dfs</group1d>       10     <artifactid>aliyun-adk-dfs</artifactid>       11     <version>1.0.3       12     <version>1.0.3       13     <dependency>       14     </dependency>       15     <version>1.0.3       16     </version></version></version></dependency></version></ore_2.12<></artifactid></dependencies> 17        18 <gloup1.0.3< td="">       19     <group1d>con.aliyun.dfs</group1d>       10     <artifactidanaven-assembly-plugins< td="">       11     <gloup1.0.3< td="">       12     <group1d>consition       13     <gloup1.0.3< td="">       14     <descriptorrefs>       15     <gloup1.0.3< td="">       16     <gloup1.0.3< td="">       17     <gloup1.0.3< td="">       18     <gloup1.0.3< td="">       29     <group1.0.3< td="">       20     <gloup1.0.3< td="">       21     <gloup1.0.3< td="">       22</gloup1.0.3<></gloup1.0.3<></group1.0.3<></gloup1.0.3<></gloup1.0.3<></gloup1.0.3<></gloup1.0.3<></descriptorrefs></gloup1.0.3<></group1d></gloup1.0.3<></artifactidanaven-assembly-plugins<></gloup1.0.3<>	8	<pre><artifactid>SparkExampleJava</artifactid></pre>
10        11        12        13        14        15        16        17        18        19        19        10        10        11        12        13        14        15        16        17        18        19        10        11        11        12        13        14        15        15        16        17        17        18        19        19        10        10        11        12        13        14        15        16        17        17 </td <td>9</td> <td><version>1.0-SNAPSHOT</version></td>	9	<version>1.0-SNAPSHOT</version>
<pre>i dependencies&gt; dependencies&gt; dependencies&gt; dependencies&gt; dependencies/ dependency&gt; dependency&gt; dependency dependencies&gt; dependencies de</pre>	10	
<pre>i cdependency&gt;</pre>	11	<dependencies></dependencies>
<pre>i</pre>	12	- <dependency></dependency>
<pre>id</pre>	13	<proupid>org.apache.spark</proupid>
<pre>version&gt;2.4.3</pre> version>2.4.3version> <pre>version&gt;2.4.3</pre> version>2.4.3version> <pre>version&gt;2.4.3</pre> version> <pre>version&gt;2.4.3</pre> version> <pre>version&gt;2.1.1</pre> version> <pre>version&gt;2.1.1</pre> version> <pre>version&gt;2.1.1</pre> version>1.0.3version>1.0.3version>1.0.3version>1.0.3version>1.0.3version>1.0.3version>1.0.3version>1.0.3version>1.0.3version>1.0.3version>1.0.3version>1.0.3version>1.0.3version>2.2 <pre>version&gt;2.1.0.3</pre> version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2.6version> <pre>version&gt;2.6</pre> version>2.2version>2.2version>2.2version>2.2version>2.2.6version> <pre>version&gt;2.6</pre> version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2version>2.2v	14	<pre><artifactid>spark-core 2.12</artifactid></pre>
<pre></pre>	15	<pre><version>2.4.3</version></pre>
<pre></pre>	16	
<pre>// </pre> //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  //  /	17	, appendeneg.
<pre>sqrupid&gt;compid&gt;com.aliyun.dfs</pre>	18	<dependency></dependency>
<pre>varifactId&gt;aiyun-adk-dfs/artifactId&gt; version&gt;1.0.3 version&gt;1.0.3 version&gt;1.0.3 version&gt;1.0.3 version&gt;1.0.3 version&gt;1.0.3 version&gt;1.0.3 version&gt;2.2 version&gt;1.0.3 version&gt;2.2 version&gt;2.2</pre>	19	<pre> <a comparison="" of="" sec<="" second="" td="" the=""></a></pre>
<pre>20</pre>	20	<pre><group1d>com.allyun.dls</group1d></pre>
<pre>21</pre>	20	varcian>1 0 3/varcian>
<pre>22  23 24  25 25 26  27 <plugins> 28 <plugin> 29 <groupid>org.apache.maven.plugins</groupid> 20 <artifactid>maven-assembly-plugin</artifactid> 20 <artifactidsmaven-assembly-plugin< artifactid=""> 21 <version>2.6<td>21</td><td></td></version></artifactidsmaven-assembly-plugin<></plugin></plugins></pre>	21	
<pre>23 24 25 25 25 26 26 26 27 29 29 29 29 29 20 20 20 20 20 20 20 20 20 20 20 20 20</pre>	22	dependency/</td
<pre>v/dependencies&gt; v/dependencies&gt; v/dependencies&gt; v/dependencies&gt; v/dependencies&gt; v/dependencies&gt; v/dependencies&gt; v/dependencies&gt; v/dependencies&gt; v/dependencies&gt; v/descriptorRefs&gt; v/descrip</pre>	23	(demendencies)
de duild>	24	
<pre>coulds cplugins cylugins cylugins cylugins cartifactId&gt;maven.assembly-plugin cartifactId&gt;maven.assembly-plugin could could co</pre>	25	
<pre>2/ <plugins <br="">28 <plugins <br="">29 <groupid>org.apache.maven.plugins</groupid> 30 <artifactid>maven-assembly-plugins/artifactId&gt; 31 <version>2.6</version> 32 <configuration> 33 <appendassemblyid>false</appendassemblyid> 34 <descriptorrefs> 35 <descriptorrefs> 36 </descriptorrefs> 37 <archive> 38 <manifest> 39 <manifest> 40 </manifest> 41 </manifest> 42 </archive></descriptorrefs></configuration> 43 <executions> 44 <executions> 45 <id>make-assembly</id> 46 <goals> 48 <goals> 48 <goals> 49 &lt;</goals> 49 <!--/goals--> 50 </goals></goals></executions></executions></artifactid></plugins> 51  52 </plugins></pre>	26	
<pre>28 <pre>cprugin&gt; 29 <groupid>org.apache.maven.plugins/groupId&gt; 30 <artifactid>maven-assembly-plugin</artifactid> 31 <version2.6< version=""> 32 <configuration> 33 <appendassemblyid>false</appendassemblyid> 34 <descriptorrefs> 35 <descriptorrefs> 36 </descriptorrefs> 37 <archive> 38 <mainfest> 39 <mainfest> 39 <mainfest> 41 </mainfest></mainfest></mainfest></archive> 42 </descriptorrefs></configuration> 43 <executions> 44 <executions> 45 <idoxmanifest> 46 <phase>package</phase> 47 <goals> 48 <goals> 49 &lt;</goals> 49 &lt;</goals> 49 &lt; 49 &lt; 49 &lt;</idoxmanifest></executions></executions></version2.6<></groupid></pre> 49 &lt;</pre> 40	27	<pre><pre>cplugins&gt;</pre></pre>
<pre>29</pre>	28	<pre><pre>conditions</pre></pre>
<pre>30</pre>	29	<proupid>org.apacne.maven.piugins</proupid>
<pre>31</pre>	30	<artifactid>maven-assembly-plugin</artifactid>
<pre>32 <configuration> 33 <appendassemblyid>false</appendassemblyid> 34 <descriptorrefs> 35 <descriptorrefs>ar-with-dependencies 36 </descriptorrefs> 37 <archive> 38 <archive> 39 <archive> 39 <archive> 40  41 </archive> 42 </archive></archive></archive></descriptorrefs></configuration> 43 <executions> 44 <executions> 45 <aichive>assembly 46 <phase>package</phase> 47 <agcals> 48 <aggal>assembly 49 <!--/ggals--> 50  51 </aggal></agcals></aichive></executions> 52  53  54  55 </executions></pre>	31	<version>2.6</version>
<pre>33</pre>	32	<configuration></configuration>
<pre>34</pre>	33	<appendassemblyid>false</appendassemblyid>
<pre>35</pre>	34	<descriptorrefs></descriptorrefs>
<pre>36  37 <archive> 38 <manifest> 39 <mainclass>com.aliyun.liumi.spark.example.WordCount</mainclass> 40 </manifest> 41 </archive> 42  42  43 <executions> 44 <executions> 45 <archive=assembly< id=""> 46 <phase>package</phase> 47 <goals> 48 <goal>assembly</goal> 49 <!--/goals--> 50 </goals></archive=assembly<></executions> 51 </executions> 52  53  54  55 </pre>	35	<descriptorref>jar-with-dependencies</descriptorref>
<pre>37</pre>	36	
<pre>38 <manifest> 39 <manifest> 39 <manifest> 40 </manifest> 41  42  43 <executions> 44 <execution> 45 <id>amake-assembly</id> 46 <phase>package</phase> 47 <goals> 48 <goal>assembly</goal> 49 <!--/goals--> 50  51 </goals></execution> 52  53  54  55 </executions></manifest></manifest></pre>	37	<archive></archive>
<pre>39</pre>	38	<manifest></manifest>
<pre>40  41  42  43 <executions> 44 <execution> 45 <id>make-assembly</id> 46 <phase>package</phase> 47 <goals> 48 <goal>assembly</goal> 49 <!--/goals--> 50 </goals></execution> 51 </executions> 52  53  54 </pre>	39	<mainclass>com.aliyun.liumi.spark.example.WordCount</mainclass>
<pre>41 </pre> 42  42  43 <pre>44 </pre> 44  45  45  46 <pre>47 </pre> 47 <pre>49 </pre> 48 <pre>49 </pre> 49  49  49  49  49  49  49  49  49  49  49  49  49  49  49  49  49  49  49  49  49  49  49  49  49  49  49  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40  40	40	
<pre>42  43  44  44  45  46  46  47 package 47  48 assembly 49 <!--/goals--> 50  51  51  52  53  54 </pre>	41	
<pre>43 <executions> 44 <execution> 45 <id>make-assembly</id> 46 <phase>package</phase> 47 <goals> 48 <goal>assembly</goal> 49 </goals> 50 </execution> 51  51 </executions> 52  53  54 </pre>	42	
<pre>44</pre>	43	<executions></executions>
<pre>45</pre>	44	<execution></execution>
<pre>46 <phase>package</phase> 47 <goals> 48 <goal>assembly</goal> 49 </goals> 50 </pre> 51  52  53  54	45	<id>make-assembly</id>
<pre>47</pre>	46	<pre><phase>package</phase></pre>
<pre>48 <goal>assembly</goal> 49  50  51  52  53  54 </pre>	47	<goals></goals>
<pre>49  50  51  52  53  54 </pre>	48	<goal>assembly</goal>
5051525354	49	
51525354	50	
<pre>52  53  54 </pre>	51	
<pre>53  54 </pre>	52	
54	53	
	54	

55</project>

#### 4. Create the Dockerfile.

#### OSS:

```
# Spark base image
FROM registry.cn-beijing.aliyuncs.com/eci_open/spark:2.4.4
RUN rm $SPARK_HOME/jars/kubernetes-client-*.jar
ADD https://repo1.maven.org/maven2/io/fabric8/kubernetes-client/4.4.2/kubernetes-client-4.4
.2.jar $SPARK_HOME/jars
RUN mkdir -p /opt/spark/jars
COPY SparkExampleJava-1.0-SNAPSHOT.jar /opt/spark/jars
# Copy the JAR dependency packages of OSS.
COPY aliyun-sdk-oss-3.4.1.jar /opt/spark/jars
COPY hadoop-aliyun-2.7.3.2.6.1.0-129.jar /opt/spark/jars
COPY jdom-1.1.jar /opt/spark/jars
```

For more information about how to download the JAR dependency packages of OSS, see Use HDP 2.6based Hadoop to read and write OSS data.

#### 5. Build the image of the Spark application.

```
docker build -t registry.cn-beijing.aliyuncs.com/liumi/spark:2.4.4-example -f Dockerfile .
```

#### 6. Push the image to Alibaba Cloud Container Registry.

docker push registry.cn-beijing.aliyuncs.com/liumi/spark:2.4.4-example

The image of the Spark application is prepared. You can deploy the Spark application in a Kubernetes cluster.