ALIBABA CLOUD

Alibaba Cloud

Elastic Container Instance

Instances

Document Version: 20220705

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.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized 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 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
⚠ 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.
🔔 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.
🔈 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 list --instanceid` *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.Overview

An elastic container instance includes infrastructure components such as vCPUs, memory, network, and storage. You can customize and change configurations of the elastic container instance in a quick manner. You have full control over the elastic container instance and do not need to manage or configure the underlying servers. You only need to provide the packaged container images to run the containers.

Resource creation

You can use multiple methods to create elastic container instances based on your business and usage scenarios. These methods correspond to different billing modes.

- Specify the number of vCPUs and memory capacity

  You are charged based on the specified vCPU and memory specifications. For vCPU and memory specifications that are not supported, the system adjusts the specifications and bills based on the new specifications.

  You can specify the number of vCPUs and memory capacity for an elastic container instance or for containers in an elastic container instance.

  - Specify the number of vCPUs and memory capacity for containers in an elastic container instance
  - Specify the number of vCPUs and memory capacity for an elastic container instance

  For more information, see Create an elastic container instance with the specified CPU and memory resources.

- Specify the Elastic Compute Service (ECS) instance type

  You are charged based on the specified ECS instance type. You can specify the ECS instance family to be used by the elastic container instance to grant its corresponding capabilities. For example, if you select the ecs.sn1ne instance family, the elastic container instance has the enhanced network capabilities that the instance family provides.

  For more information, see the following topics:

  - Create an elastic container instance by specifying an ECS instance type
  - Create a GPU-accelerated elastic container instance
  - Create an AMD elastic container instance
  - Create elastic container instances with local disks

Cost optimization

You can use pay-as-you-go elastic container instances together with preemptible elastic container instances, reserved instances, and savings plans to reduce your costs based on your business needs.

- Preemptible elastic container instances can be used for stateless and fault-tolerant business workloads. For more information, see Use preemptible instances.
- For long-term stable business loads, we recommend that you use reserved instances or savings plans to offset the bills of elastic container instances. The following deduction methods are applied based on the billing modes of elastic container instances:
  - Elastic container instances billed based on vCPUs and memory

    Only general-purpose savings plans are supported.

  - Elastic container instances billed based on ECS instance types

    General-purpose savings plans, ECS compute savings plans, and reserved instances are supported.

  You can select a deduction method based on your elastic container instance. For more information, see Use reserved instances or Use savings plans.

Resource inventory

Elastic Container Instance provides cloud resources for containers. When you create a large number of elastic container instances, some resources in the specified region and zone may be insufficient. We recommend that you specify multiple ECS instance types across multiple zones to ensure that the elastic container instances can be created. For more information, see the following topics:

- Create an elastic container instance by specifying multiple zones
- Create an elastic container instance by specifying multiple instance types

Lifecycle

If you do not connect Elastic Container Instance to Kubernetes clusters, you must manage the lifecycle of elastic container instances and process business loads based on the status of the elastic container instances to ensure business robustness and avoid resource waste. For more information, see Elastic container instance lifecycle.

2.Lifecycle of an elastic container instance

This topic describes different states in the lifecycle of an elastic container instance. You can design and implement your business logic based on the states. Elastic Container Instance is seamlessly connected to Kubernetes. You can directly use elastic container instances as pods in Kubernetes clusters.

Status of elastic container instances

The following table describes the states of an elastic container instance during its lifecycle.

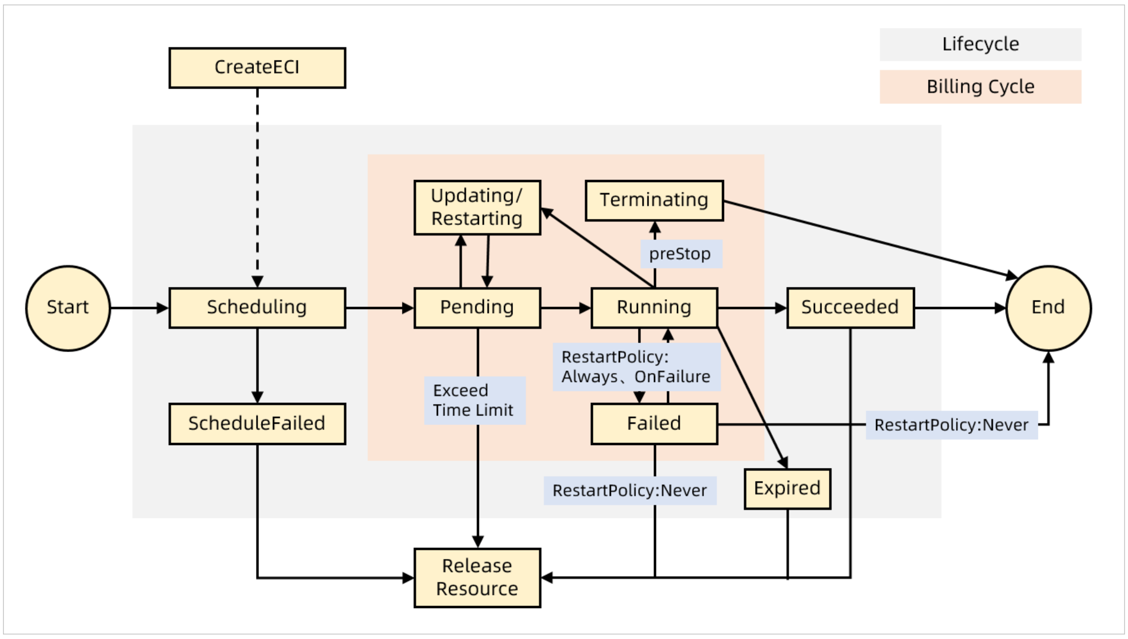
State	Description
Scheduling	The elastic container instance is being created.
Pending	One or more containers in the elastic container instance are being started, and no containers are running.
Running	All containers in the elastic container instance are created, and at least one container is running or restarting.
Restarting	The elastic container instance is restarting.
Updating	The elastic container instance is being updated.
Terminating	The elastic container instance is being terminated. A running instance configured with preStop enters the Terminating state when the instance is being deleted. After you run the preStop function, the instance is deleted.
Succeeded	All containers in the elastic container instance are terminated after they successfully run and not about to restart.
Failed	All containers in the elastic container instance are terminated after they run, and at least one container is terminated due to runtime errors, which indicates that the container exits with a non-zero status or is terminated by the system.
Expired	The elastic container instance is a preemptible instance and is terminated due to resource expiration and recycling.

State	Description
ScheduleFailed	The elastic container instance failed to be created. The system deletes the elastic container instance after 24 hours, and you are not charged for the elastic container instance.

Notice

The restart policy of an elastic container instance applies only to containers within the elastic container instance. The elastic container instance itself does not automatically restart.

The following figure shows the state transition of an elastic container instance during its lifecycle.



Note

The system automatically releases the instances that are in the Succeeded or Failed state, but retains the metadata of the latest 100 instances. When the system releases the instances, other resources such as Elastic IP Addresses (EIPs) that are created along with the instances are also released.

Container status

State	Description
Waiting	The container is waiting for being created. Typically, application containers remain in the Waiting state before an init container exits the Running state.

State	Description
Running	The container is created and running.
Terminated	The container is terminated and exits after a successful or failed run.

3.Create resources

3.1. Specify the number of vCPUs and memory size to create an elastic container instance

If you do not require specialized specifications such as GPUs and local disks, we recommend that you specify the number of vCPUs and memory size instead of Elastic Compute Service (ECS) instance types to create elastic container instances. Then, the system can try a variety of ECS instance types based on the specified vCPU and memory specifications to select a suitable instance type. This method provides better elasticity and resource provisioning than the method of specifying only ECS instance types.

Background information

If your specified number of vCPUs or memory size is invalid when you create an elastic container instance, the system adjusts the number or size based on the specifications supported by Elastic Container Instance. When the system adjusts the number or size, the system selects the most similar specifications supported by Elastic Container Instance to create an elastic container instance, and the specifications selected by the system are higher than or equal to the specifications that you specified for the elastic container instance. For example, assumed that you have specified 7 vCPUs and 13 GiB of memory to create an elastic container instance, but these specifications are invalid. In this case, the system adjusts the number of vCPUs and memory size and selects 8 vCPUs and 16 GiB of memory that are supported by Elastic Container Instance to create an elastic container instance.

The following table describes the specifications supported by Elastic Container Instance.

> Note
>
> - Each elastic container instance can have only one elastic network interface (ENI) bound.
> - If you do not specify the number of vCPUs and memory size, the system uses 2 vCPUs and 4 GiB of memory to create an elastic container instance by default.

vCPU	Memory (GiB)	emptyDir volume (GiB)	Bandwidth (bidirectional, Gbit/s)	Packet forwarding rate (bidirectional, Kpps)	ENI queues
0.25	0.5 and 1	20	0.1	40	1
0.5	1 and 2	20	0.2	50	1
1	2, 4, and 8	20	0.5	50	1

vCPU	Memory (GiB)	emptyDir volume (GiB)	Bandwidth (bidirectional, Gbit/s)	Packet forwarding rate (bidirectional, Kpps)	ENI queues
2	1, 2, 4, 8, and 16	20	1	300	2
4	2, 4, 8, 16, and 32	20	1.5	500	2
8	4, 8, 16, 32, and 64	20	2.0	800	4
12	12, 24, 48, and 96	20	2.5	900	4
16	16, 32, 64, and 128	20	3.0	1,000	4
24	48, 96, and 192	20	4.5	1,500	6
32	64, 128, and 256	20	6.0	2,000	8
52	96, 192, and 384	20	12.5	3,000	32
64	128, 256, and 512	20	20.0	4,000	16

You can use one of the following methods to create an elastic container instance by specifying the number of vCPUs and memory size:

- Specify the number of vCPUs and memory size for containers in an elastic container instance
- Specify the number of vCPU and memory size for an elastic container instance

Note

You cannot create an elastic container instance with a 1:1 vCPU-to-memory ratio by using the Elastic Container Instance console. However, you can create this type of instance by calling API operations or using Kubernetes.

Specify the number of vCPUs and memory size for containers in an elastic container instance

This method is the default method of creating pods in Kubernetes. You can create a maximum of 20 containers within each elastic container instance and specify the vCPU and memory specifications of each container. For each elastic container instance, the total vCPU and memory specifications of all containers must be lower than or equal to the vCPU and memory specifications of the instance. If the total vCPU or memory specifications of all containers are higher than the vCPU or memory specifications of the instance, the system adjusts the container specifications and then charges you based on the new specifications.

Kubernetes mode

You can specify the number of vCPUs and memory size in the limit parameter for a container. Sample code:

```
apiVersion: apps/v1 # for versions before 1.8.0 use apps/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment-basic
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9 # replace it with your exactly <image_name:tags>
        command: sleep 3600
        ports:
        - containerPort: 80
        resources:
          limits:
            cpu: "500m"      #Specify the number of vCPUs for the NGINX container.
            memory: "1024Mi"   #Specify the memory size for the NGINX container.
      - name: busybox
        image: busybox:latest # replace it with your exactly <image_name:tags>
        ports:
        - containerPort: 8080
        resources:
          limits:
            cpu: "500m"   #Specify the number of vCPUs for the BusyBox container.
            memory: "1024Mi"  #Specify the memory size for the BusyBox container.
```

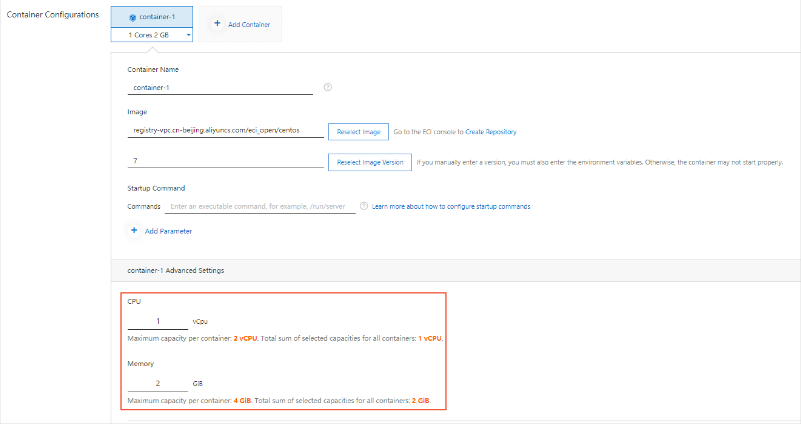
API mode

When you call the CreateContainerGroup API operation to create an elastic container instance, you can use the Container.N.Cpu and Container.N.Memory parameters to specify the number of vCPUs and memory size for each container. The following table describes the parameters. For more information, see CreateContainerGroup.

Parameter	Type	Required	Example	Description
Container.N.Cpu	Float	No	2	The number of vCPUs of container N. N specifies the serial number of the container. Maximum value of N: 20.
Container.N.Memory	Float	No	4	The memory size of container N. Unit: GiB. N specifies the serial number of the container. Maximum value of N: 20.

Console mode

When you create an elastic container instance by using the elastic container instance buy page in the Elastic Container Instance console, you can configure the number of vCPUs and memory size in the **Advanced Settings** section for each container. The total vCPU and memory specifications of all containers in an elastic container instance must be lower than or equal to the vCPU and memory specifications of the instance.



1 Cores 2 GB

Container Name: container-1

Image: registry-vpc.cn-beijing.aliyuncs.com/eci_open/centos

Go to the ECI console to Create Repository

7

If you manually enter a version, you must also enter the environment variables. Otherwise, the container may not start properly.

Startup Command

Commands: Enter an executable command, for example, /run/server

Learn more about how to configure startup commands

container-1 Advanced Settings

CPU: 1 vCpu

Maximum capacity per container: 2 vCPU. Total sum of selected capacities for all containers: 1 vCPU.

Memory: 2 GiB

Maximum capacity per container: 4 GiB. Total sum of selected capacities for all containers: 2 GiB.

Specify the number of vCPUs and memory size for an elastic container instance

When this method is used, the system tries a variety of ECS instance types based on the specified number of vCPUs and memory size to select a suitable ECS instance type for better elasticity and resource provisioning. This method provides the following benefits:

- You do not need to specify the number of vCPUs, memory size, or resource limits for containers in the elastic container instance. This provides the containers with greater flexibility to share the requested resources.
- In the genetic computation and Istio scenarios, the service framework automatically deploys sidecar containers in pods. You can seamlessly connect Elastic Container Instance to the service framework by specifying the number of vCPUs and memory size for elastic container instances.

Kubernetes mode

You can add annotations to metadata in the configuration file of the pod to specify the number of vCPUs and memory size for an elastic container instance. Sample code:

```
apiVersion: apps/v1 # for versions before 1.8.0 use apps/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment-basic
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
      annotations:
        k8s.aliyun.com/eci-use-specs : "2-4Gi"   #Specify the number of vCPUs and memory si
ze of the elastic container instance.
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9 # replace it with your exactly <image_name:tags>
        ports:
        - containerPort: 80
      - name: busybox
        image: busybox:latest # replace it with your exactly <image_name:tags>
        ports:
        - containerPort: 80
        resources:
          limits:
            cpu: "500m"     #Specify the maximum number of vCPUs for the container.
            memory: "1024Mi"   #Specify the maximum memory size for the container.
    nodeName: virtual-kubelet  #Specify a virtual node name to schedule the pod to run on t
he elastic container instance in a Container Service for Kubernetes (ACK) cluster.
```

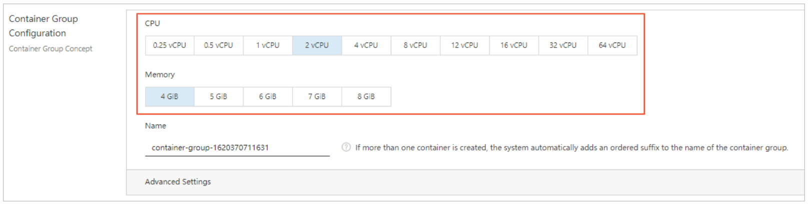
API mode

When you call the CreateContainerGroup API operation to create an elastic container instance, you can use the Cpu and Memory parameters to specify the number of vCPUs and memory size for the elastic container instance. The following table describes the parameters. For more information, see CreateContainerGroup.

Parameter	Type	Required	Example	Description
Cpu	Float	No	2	The number of vCPUs of the elastic container instance.
Memory	Float	No	4	The memory size of the elastic container instance. Unit: GiB.

Console mode

When you create an elastic container instance by using the elastic container instance buy page in the Elastic Container Instance console, you can specify the number of vCPUs and memory size in the **Container Group Configuration** section.



Name

container-group-1620370711631

If more than one container is created, the system automatically adds an ordered suffix to the name of the container group.

3.2. Specify an ECS instance type to create an elastic container instance

In some scenarios, you may have specific specification requirements such as for GPUs, enhanced network capabilities, high clock speeds, and local disks. To meet these requirements, you can create an elastic container instance by specifying an Elastic Compute Service (ECS) instance type.

Background information

When you specify an ECS instance type to create an elastic container instance, you are charged for the compute resources that are consumed by the elastic container instance based on the rate of the ECS instance type.

The following ECS instance families are supported:

- General-purpose instance families: g7, g6e, g6a, g6, g5, sn2, and sn2ne.
- Compute-optimized instance families: c7, c6e, c6a, c6, c5, sn1, and sn1ne.
- Memory-optimized instance families: r7, r6e, r6a, r6, r5, se1ne, and se1
- Compute-intensive instance family: ic5
- Compute-optimized instance families with high clock speeds: hfc7, hfc6, and hfc5
- General-purpose instance families with high clock speeds: hfg7, hfg6, and hfg5
- memory-optimized instance family with high clock speeds: hfr7
- GPU-accelerated compute-optimized instance family: gn7i, gn7, gn6i, gn6e, gn6v, gn5i, and gn5
- Big data instance family with enhanced network performance: d1ne
- Instance families with local SSDs: i2 and i2g
- Burstable instance families: t6 and t5
- Shared instance families: xn4, n4, mn4, n1, n2, e3, and e4.

Notice

If you use burstable instance families, take note of the following items:

- Burstable instances use CPU credits to maintain computing performance and are suitable for scenarios in which CPU utilization is typically low but experiences occasional bursts. For more information, see Overview.
- You cannot modify ulimit.

For more information, see the following references:

- ECS Instance Types Available for Each Region
- Instance families

Use Kubernetes to specify an ECS instance type

You can add annotations to metadata in the configuration file of the pod to specify an ECS instance type. The following sample code shows how to specify an ECS instance type by using Kubernetes:

```
apiVersion: apps/v1 # for versions before 1.8.0 use apps/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment-basic
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
      annotations:
        k8s.aliyun.com/eci-use-specs: ecs.c5.large  #Specify an ECS instance type.
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9 # replace it with your exactly <image_name:tags>
        ports:
        - containerPort: 80
     nodeName: virtual-kubelet    #Specify a virtual node name to schedule the pod to run o
n the elastic container instance in a Container Service for Kubernetes (ACK) cluster.
```

Call an API operation to specify an ECS instance type

When you call the CreateContainerGroup operation to create an elastic container instance, you can use the InstanceType parameter to specify one or more ECS instance types. The following table describes the parameter. For more information, see CreateContainerGroup.

Parameter	Type	Required	Example	Description
InstanceType	String	No	ecs.c5.xlarge,ecs.g5.xlarge	The ECS instance types. You can specify up to five ECS instance types at a time. Separate multiple instance types with commas (,). Example: ecs.c5.xlarge,ecs.g5.xlarge.

3.3. Create job-optimized elastic container instances

You can run short-term jobs on elastic container instances to improve resource utilization and reduce computing costs. Job-optimized elastic container instances provide multiple benefits. For example, a job-optimized elastic container instance can be started in seconds. This topic describes how to create and use a job-optimized elastic container instance.

Introduction to job-optimized elastic container instances

Job-optimized elastic container instances are suitable for job-type scenarios. The instances have the following features:

- Job-optimized elastic container instances can be started faster than regular elastic container instances. The cold start time of job-optimized elastic container instances is only half of the cold start time of regular elastic container instances.
- For batch tasks that are continuously executed, the image caching feature is automatically enabled.
- Job-optimized elastic container instances allow you to only access external services from the instances. You cannot access the instances from external services. For example, you can access the Internet as well as Apsara File Storage NAS (NAS) and Object Storage Service (OSS) in your virtual private cloud (VPC).

  The following table describes the network types that are supported by job-optimized elastic container instances:

Network type	Description
N/A	You cannot access the Internet or VPCs from the instance, but images can be pulled from image repositories in VPCs in the region where the instance resides. For example, if your instance is deployed in the China (Beijing) region, you can pull the `registry-vpc.cn-beijing.aliyuncs.com/eci_open/busybox:1.30` image.

Network type	Description
Internet	You can access the Internet from the instance and pull images over the Internet. You do not need to configure a NAT gateway.
VPC	You can access the services in VPCs from the instance. To access your VPCs from the instance, you must configure the vSwitches and security groups of the VPCs.
Internet and VPC	You can access the Internet and services in VPCs from the instance. ○ To access the Internet from the instance, you do not need to configure a NAT gateway. ○ To access your VPCs from the instance, you must configure the vSwitches and security groups of the VPCs.

Note

- If your instance does not need to access VPCs, you can set the network type to **N/A** or **Internet**. In this case, you do not need to configure vSwitches or security groups.
- If Alibaba Cloud DNS PrivateZone is used for domain name resolution, set the network type to **VPC**. In this case, to access the Internet, you can configure a NAT gateway for the vSwitch.
- If the network type is **VPC** or **Internet and VPC**, only two IP addresses in the vSwitch are used. New elastic container instances do not use extra IP addresses in the vSwitch.

Limits

- Region: Job-optimized elastic container instances can be created only in the China (Hangzhou), China (Shanghai), China (Beijing), and China (Shenzhen) regions.
- Billing method: Only the pay-as-you-go billing method is supported. Preemptible instances are not supported.
- Network: You can only access external services from job-optimized elastic container instances. You cannot access job-optimized elastic container instances from external services. As a result, job-optimized elastic container instances cannot provide services to external resources.
- Network bandwidth: The upper limit is 1 Gbit/s.
- Instance specifications: A job-optimized elastic container instance can contain a maximum of 16 vCPUs. The following table describes the supported instance specifications.

vCPU	Memory (GiB)
0.25	0.5 and 1

vCPU	Memory (GiB)
0.5	1 and 2
1	2, 4, and 8
2	1, 2, 4, 8, and 16
4	2, 4, 8, 16, and 32
8	4, 8, 16, 32, and 64
12	12, 24, 48, and 96
16	16, 32, 64, and 128

> Note
>
> You can create a job-optimized instance by specifying the number of vCPUs and the memory size. You cannot create a job-optimized instance by specifying an ECS instance type.

- Other limits:
  - You cannot associate an elastic IP address (EIP) with a pod.
  - You cannot add a job-optimized elastic container instance as a backend server to a Server Load Balancer (SLB) instance.
  - You cannot assign RAM roles to job-optimized elastic container instances.
  - You cannot add metadata to job-optimized elastic container instances.
  - You cannot set the size of the temporary storage space.
  - You cannot use disk volumes for persistent storage on job-optimized elastic container instances. If the network type of a job-optimized elastic container instance is set to N/A or Internet, you cannot use an Apsara File Storage NAS (NAS) volume or an OSS bucket for persistent storage on the instance.
  - Job-optimized elastic container instances do not support coredump or tcpdump.

Create a job-optimized elastic container instance in scenarios in which Kubernetes is involved

Notice

If you want to create job-optimized elastic container instances, the version of Virtual Kubelet in the Kubernetes cluster must be later than 2.1.0. If the version of Virtual Kubelet in the Kubernetes cluster is 2.1.0 or earlier, update Virtual Kubelet. For more information, see Update Virtual Kubelet.

You can add the k8s.aliyun.com/eci-network-config parameter to the metadata of the pod to declare the network type and the instance category of the pod. Valid values of `k8s.aliyun.com/eci-network-config` are:

- none: N/A.
- nat_internet: Internet.
- nat_vpc: VPC.
- nat_internet_vpc: Internet and VPC.

Sample configurations:

```
apiVersion: v1
kind: Pod
metadata:
  name: test-nat-vpc
  namespace: default
  labels:
    eci: "true"
  annotations:
    k8s.aliyun.com/eci-network-config: "nat_vpc"
spec:
 nodeName: virtual-kubelet-cn-shanghai-f
 containers:
 - image: registry-vpc.cn-shanghai.aliyuncs.com/eci_open/perl:5.32
   imagePullPolicy: IfNotPresent
   name: pi
   command: ["perl",  "-Mbignum=bpi", "-wle", "print bpi(2000)"]
 restartPolicy: OnFailure
```

You can also use the ECI Effect feature of Elastic Container Instance Profile to automatically add annotations to pods that can be matched by labels.

Sample configurations:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: eci-profile
  namespace: kube-system
data:
  vpcId: "vpc-xxx"
  securityGroupId: "sg-xxx"
  vswitchIds: "vsw-111,vsw-222"
  enableClusterIp: "false"
  enableHybridMode: "false"
  enablePrivateZone: "false"
  selectors: |
    [
        {
            "name":"custom-selector-1",
            "objectSelector":{
                "matchLabels":{
                    "eci":"true"
                }
            },
            "effect":{
             "annotations":{
                "k8s.aliyun.com/eci-network-config": "nat_internet"
            }
        },
        {
            "name":"custom-selector-2",
            "namespaceSelector":{
                "matchLabels":{
                    "eci":"true"
                }
            },
            "effect":{
             "annotations":{
                "k8s.aliyun.com/eci-network-config": "nat_vpc"
            }
          }
        }
    ]
```

The preceding example declares two selectors that can implement the following features:

- If the pod has the `eci=true` label, the pod uses job-optimized elastic container instances whose network type is Internet.
- If the namespace in which the pod is located has the `eci=true` label, the pod uses job-optimized elastic container instances whose network type is VPC.

For more information, see Configure Elastic Container Instance Profile.

Call an API operation to create a job-optimized elastic container instance

If you call the CreateContainerGroup API operation to create an elastic container instance, you can use tag-related parameters to configure a job-optimized instance. The following table describes these parameters. For more information, see CreateContainerGroup.

Parameter	Type	Example	Description
Tag.N.Key	String	eci_networkConfig	The key of tag N. If you set Tag.N.Key to `eci_networkConfig`, the instance is a job-optimized elastic container instance.
Tag.N.Value	String	nat_vpc	The value of tag N. If you set Tag.N.Key to `eci_networkConfig`, the Tag.N.Value parameter specifies the network type of the job-optimized elastic container instance. Valid values: • none: N/A. • nat_internet: Internet. • nat_vpc: VPC. • nat_internet_vpc: Internet and VPC.
VSwitchId	String	vsw-bp1xpiowfm5vo8o3c****	The ID of the vSwitch. If you set Tag.N.Value to `nat_vpc` or `nat_internet_vpc`, you must configure this parameter.
SecurityGroupId	String	sg-uf66jeqopgqa9hdn****	The ID of the security group. If you set Tag.N.Value to `nat_vpc` or `nat_internet_vpc`, you must configure this parameter.

Sample configurations:

- Configurations when Tag.N.Value is set to none.

```
Tag.1.Key = eci_networkConfig,
Tag.1.Value = none
```

- Configurations when Tag.N.Value is set to nat_internet.

```
Tag.1.Key = eci_networkConfig,
Tag.1.Value = nat_internet
```

- Configurations when Tag.N.Value is set to nat_vpc.

```
Tag.1.Key = eci_networkConfig,
Tag.1.Value = nat_vpc
VSwitchId = vsw-bp1xpiowfm5vo8o3c****
SecurityGroupId = sg-uf66jeqopgqa9hdn****
```

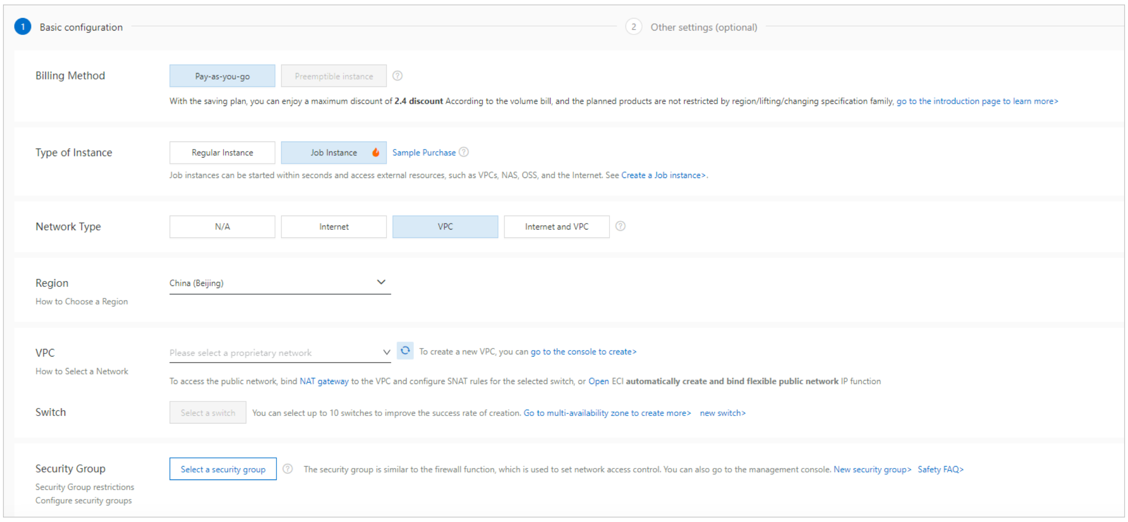
- Configurations when Tag.N.Value is set to nat_internet_vpc

```
Tag.1.Key = eci_networkConfig,
Tag.1.Value = nat_internet_vpc
VSwitchId = vsw-bp1xpiowfm5vo8o3c****
SecurityGroupId = sg-uf66jeqopgqa9hdn****
```

Use the Elastic Container Instances console to create a job-optimized elastic container instance

If you set the Billing Method parameter to **Pay-as-you-go** when you create a job-optimized elastic container instance on the elastic container instance buy page, you can set the Type of Instance parameter to **Job Instance** and then configure the network type.

> **Note**
>
> You cannot create multiple job-optimized elastic container instances at a time.



1 Basic configuration

2 Other settings (optional)

Billing Method: Pay-as-you-go | Preemptible instance

With the saving plan, you can enjoy a maximum discount of **2.4 discount** According to the volume bill, and the planned products are not restricted by region/lifting/changing specification family, go to the introduction page to learn more>

Type of Instance: Regular Instance | Job Instance | Sample Purchase

Job instances can be started within seconds and access external resources, such as VPCs, NAS, OSS, and the Internet. See Create a Job instance>.

Network Type: N/A | Internet | VPC | Internet and VPC

Region: China (Beijing)

How to Choose a Region

VPC: Please select a proprietary network

How to Select a Network

To create a new VPC, you can go to the console to create>

To access the public network, bind NAT gateway to the VPC and configure SNAT rules for the selected switch, or Open ECI **automatically create and bind flexible public network** IP function

Switch: Select a switch

You can select up to 10 switches to improve the success rate of creation. Go to multi-availability zone to create more> new switch>

Security Group: Select a security group

Security Group restrictions

Configure security groups

The security group is similar to the firewall function, which is used to set network access control. You can also go to the management console. New security group> Safety FAQ>

Description about the network types:

- If you select **N/A** or **Internet** for Network Type, you do not need to configure the VPC, vSwitch, or security group.
- If you select **VPC** or **Internet and VPC** for Network Type, to access the VPC from the elastic container instance, you must configure the VPC, vSwitch, and security group.

3.4. Create a GPU-accelerated elastic container instance

This topic describes how to create and use a GPU-accelerated elastic container instance.

Background information

GPU-accelerated elastic container instances feature built-in graphics cards and NVIDIA CUDA drivers. Therefore, when you use a GPU-accelerated elastic container instance, you only need to use a basic image with a runtime such as the CUDA Toolkit. You do not need to care about the installation of the driver.

> Note
>
> The GPU driver version supported by GPU-accelerated elastic container instances is NVIDIA 460.73.01. The CUDA Toolkit version supported by GPU-accelerated elastic container instances is 11.2.

You can specify an instance type of the following GPU-accelerated Elastic Compute Service (ECS) instance families to create GPU-accelerated elastic container instances:

- gn7i, a GPU-accelerated compute-optimized instance family that uses NVIDIA A10 GPUs. This instance family includes a variety of instance types, such as ecs.gn7i-c8g1.2xlarge.
- gn7, a GPU-accelerated compute-optimized instance family that uses NVIDIA A100 GPUs. This instance family includes a variety of instance types, such as ecs.gn7-c12g1.3xlarge.
- gn6v, a GPU-accelerated compute-optimized instance family that uses NVIDIA V100 GPUs. This instance family includes a variety of instance types, such as ecs.gn6v-c8g1.2xlarge.
- gn6e, a GPU-accelerated compute-optimized instance family that uses NVIDIA V100 GPUs. This instance family includes a variety of instance types, such as ecs.gn6e-c12g1.3xlarge.
- gn6i, a GPU-accelerated compute-optimized instance family that uses NVIDIA T4 GPUs. This instance family includes a variety of instance types, such as ecs.gn6i-c4g1.xlarge.
- gn5i, GPU-accelerated compute-optimized instance family that uses NVIDIA P4 GPUs. This instance family includes a variety of instance types, such as ecs.gn5i-c2g1.large.
- gn5, GPU-accelerated compute-optimized instance family that uses NVIDIA P100 GPUs. This instance family includes a variety of instance types, such as ecs.gn5-c4g1.xlarge.

  > Note
  >
  > The gn5 instance family is equipped with local disks. You can mount local disks to elastic container instances. For more information, see Create an elastic container instance that has local disks attached.

For more information, see:

- ECS Instance Types Available for Each Region
- Instance families

Use Kubernetes to specify an ECS instance type

You can add annotations to metadata in the pod configuration file to specify a GPU-accelerated ECS instance type. After you specify a GPU-accelerated ECS instance type, you must add the nvidia.com/gpu field to the containers.resources section to declare GPU resources.

Notice

The nvidia.com/gpu field indicates the number of GPUs to be allocated to a container. You must specify this field when you create a GPU-accelerated elastic container instance. If you do not specify this field, an error is returned when the pod is started.

Sample code:

```
apiVersion: apps/v1 # for versions before 1.8.0 use apps/v1beta1
kind: Deployment
metadata:
  name: nginx-gpu-demo-1
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
      annotations:
        k8s.aliyun.com/eci-use-specs: ecs.gn5i-c4g1.xlarge  #Specify a GPU-accelerated ECS
instance type.
    spec:
      containers:
      - name: nginx
        image: registry-vpc.cn-beijing.aliyuncs.com/eci_open/nginx:1.15.10 # replace it wit
h your exactly <image_name:tags>
        resources:
            limits:
              nvidia.com/gpu: '1'    # Specify the number of GPUs that are required by the
container. This field is required. If you do not specify this field, an error is returned w
hen the pod is started.
        ports:
        - containerPort: 80
```

Call an API operation to specify an ECS instance type

When you call the CreateContainerGroup API operation to create an elastic container instance, you can use the InstanceType parameter to specify a GPU-accelerated ECS instance type, and use the Container.N.Gpu and InitContainer.N.Gpu parameters to specify the number of GPUs to be allocated to a container. The following table describes the parameters. For more information, see CreateContainerGroup.

Parameter	Type	Example	Description

Parameter	Type	Example	Description
InstanceType	String	ecs.gn6v-c8g1.2xlarge	The GPU-accelerated ECS instance types. The following GPU-accelerated ECS instance families are supported: gn6v, gn6i, gn5i, gn5, gn7, and gn7i. You can specify up to five ECS instance types at a time. Separate multiple instance types with commas (,). Example: ecs.gn6v-c8g1.2xlarge,ecs.gn6i-c4g1.xlarge.
Container.N.Gpu	Integer	1	The number of GPUs to be allocated to container N.
InitContainer.N.Gpu	Integer	1	The number of GPUs allocated to init container N.

Notice

When you create a GPU-accelerated elastic container instance, you must specify both the InstanceType and Container.N.Gpu parameters. You must make sure that the total number of GPUs to be allocated to all the containers does not exceed the number of GPUs provided by each specified GPU-accelerated ECS instance type.

You can call the UpdateContainerGroup API operation to change the number of GPUs allocated to containers in an existing GPU-accelerated elastic container instance. For more information, see UpdateContainerGroup.

3.5. Create an elastic container instance that has local disks attached

This topic describes how to create and use an elastic container instance that has local disks attached.

Background information

You can specify Elastic Compute Service (ECS) instance types equipped with local disks to create elastic container instances that have local disks attached. Local disks are physically located on the same physical servers as the elastic container instances to which the disks are attached. Local disks are cost-effective and provide high random IOPS, high throughput, and low latency. Local disks reside on single physical servers, which increases the risks of single points of failure (SPOFs). For more information, see Local disks.

The following ECS instance families equipped with local disks are supported:

- d1, big data instance family that includes a variety of instance types such as ecs.d1.2xlarge
- d1ne, big data instance family with enhanced network performance, which includes a variety of instance types such as ecs.d1ne.2xlarge

- i2, instance family equipped with local SSDs, which includes a variety of instance types such as ecs.i2.xlarge
- i2g, instance family equipped with local SSDs, which includes a variety of instance types such as ecs.i2g.2xlarge
- gn5, GPU-accelerated compute optimized instance family, which includes a variety of instance types such as ecs.gn5-c4g1.xlarge.

> Note
>
> gn5 is a GPU-accelerated instance family. If you select this instance family, you must specify GPU-related parameters in addition to local disk-related parameters. For more information, see Create a GPU-accelerated elastic container instance.

For more information, see:

- ECS Instance Types Available for Each Region
- ECS instance families

Kubernetes mode

You can add annotations to metadata in the configuration file of the pod to specify an ECS instance type equipped with local disks. Local disks have the same lifecycle as the elastic container instances to which the disks are attached. Local disks cannot be attached as persistent volumes (PVs) but can be attached as flex volumes.

Sample code:

```
apiVersion: apps/v1 # for versions before 1.8.0 use apps/v1beta1
kind: Deployment
metadata:
  name: nginx-i2
  labels:
    app: nginx
spec:
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
      annotations:
        k8s.aliyun.com/eci-use-specs: ecs.i2.4xlarge  #Specify an ECS instance type equipped with local disks.
    spec:
      containers:
      - name: nginx
        image: nginx
        volumeMounts:
          - name: disk1
            mountPath: /data1
          - name: disk2
            mountPath: /data2
      volumes:       #Attach local disks to the elastic container instance. Make sure that the number of specified local disks does not exceed the number of local disks supported by the specified ECS instance type.
        - name: disk1
          flexVolume:
            driver: alicloud/local-disk
            fsType: ext4
        - name: disk2
          flexVolume:
            driver:  alicloud/local-disk
            fsType: ext4
```

API mode

When you call the CreateContainerGroup operation to create an elastic container instance, you can use the InstanceType parameter to specify ECS instance types and the volume-related parameters to attach local disks. The following table describes the parameters. For more information, see CreateContainerGroup.

Parameter	Type	Required	Example	Description

Parameter	Type	Required	Example	Description
InstanceType	String	No	ecs.gn6v-c8g1.2xlarge	The ECS instance types. The following instance families equipped with local disks are supported: d1, d1ne, i2, i2g, and gn5. You can specify up to five ECS instance types at a time. Separate multiple instance types with commas (,). Example: ecs.d1.2xlarge,ecs.d1ne.2xlarge.
Volume.N.Name	String	No	disk1	The name of volume N.
Volume.N.Type	String	No	flexvolume	The type of volume N.
Volume.N.FlexVolume.Driver	String	No	alicloud/local-disk	The name of the FlexVolume driver for volume N. Set the value to alicloud/local-disk.
Volume.N.FlexVolume.FsType	String	No	ext4	The type of the file system for volume N. The default value is determined by the script of FlexVolume.

Notice

- When you create an elastic container instance that has local disks attached, make sure that the number of volumes does not exceed the number of local disks supported by each of the specified ECS instance types.
- If the specified ECS instance type is equipped with multiple local disks, the disks can be attached in any order because they all have the same specifications.
- Local disks are physically located on a single physical server, which increases the risks of SPOFs. The durability of data stored on local disks is determined by the reliability of the associated physical server. For more information, see the "Limits" section of Local disks.

3.6. Create an AMD-compatible elastic container instance

This topic describes how to create and use an AMD-compatible elastic container instance.

Background information

AMD-compatible elastic container instances are powered by AMD EPYC^{TM} ROME processors. These instances offload a large number of virtualization features to dedicated hardware with the use of the SHENLONG architecture to provide predictable and consistent ultra-high performance and reduce virtualization overheads. AMD-compatible elastic container instances suit the following scenarios: video encoding and decoding, web frontend servers, frontend servers of massively multiplayer online (MMO) games, scenarios where large volumes of packets are received and transmitted, and scenarios where applications such as DevOps applications are developed and tested.

You can specify AMD-compatible Elastic Compute Service (ECS) instance types to create AMD-compatible elastic container instances. The c6a compute optimized ECS instance family is supported, which includes a variety of instance types such as ecs.c6a.large.

For more information, visit the following links:

- ECS Instance Types Available for Each Region
- Instance families

Kubernetes mode

You can add annotations to metadata in the configuration file of a pod to specify an AMD-compatible ECS instance type. Sample code:

```
apiVersion: apps/v1 # for versions before 1.8.0 use apps/v1beta1
kind: Deployment
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
      annotations:
        k8s.aliyun.com/eci-use-specs: ecs.c6a.xlarge  #Specify an AMD-compatible ECS instance type.
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9
        ports:
        - containerPort: 80
```

API mode

When you call the CreateContainerGroup operation to create an elastic container instance, you can use the InstanceType parameter to specify an AMD-compatible ECS instance type. The following table describes this parameter. For more information, see CreateContainerGroup.

Parameter	Type	Required	Example	Description
InstanceType	String	No	ecs.c6a.large	The AMD-compatible ECS instance types. You can specify one or more instance types in the c6a AMD-compatible instance family at a time. Example: ecs.c6a.large.

3.7. Customize CPU options

The CPU options of an elastic container instance include the number of physical CPU cores and the number of threads per core. Some elastic container instances support custom CPU options whereas other elastic container instances do not. This is determined based on how the instances are created.

Overview

Elastic Container Instance allows up to two threads to concurrently run on a single physical core based on Intel® Hyper-Threading (HT) technology. Each thread can be considered as a vCPU. You can use the following formula to calculate the number of vCPUs: Number of physical CPU cores × Number of threads per core.

The following table describes the CPU options of an elastic container instance, which are the number of physical CPU cores and the number of threads per core. Elastic container instances may support or not support custom CPU options. This is determined based on how the instances are created.

Note

You can customize CPU options at no additional costs.

CPU option	Description	Scenario
Number of physical CPU cores	Specifies the number of physical CPU cores to use.	You can specify a smaller number of physical CPU cores to decrease the number of vCPUs, improve the CPU-to-memory ratio, and reduce the number of billable items and software license costs.

CPU option	Description	Scenario
Number of threads per core	Specifies the number of threads per core to enable or disable HT on the CPU.	By default, Elastic Compute Service (ECS) instance types that support custom CPU options have HT enabled for common workloads. In this case, the number of threads per core is not 1. You can disable HT in the following scenarios: ● Some high-performance computing (HPC) scenarios. In these scenarios, you can improve the performance of instances by disabling HT. ● Memory-intensive business scenarios. You can disable HT to decrease the number of vCPUs, improve the CPU-to-memory ratio, and reduce the number of billable items and software license costs.

You can specify ECS instance types or specify vCPU and memory specifications to create elastic container instances. The system uses the specified ECS instance types or tries multiple ECS instance types based your specified vCPU and memory specifications to create elastic container instances. If the system is using an ECS instance type that supports custom CPU options, the created elastic container instances also support custom CPU options.

> ? Note
>
> Only some ECS instance types support custom CPU options, and HT is enabled on instances of these instance types by default. For more information, see Customize CPU options.

Elastic container instances may support or not support custom CPU options. This is determined based on how the instances are created.

- Elastic container instances created by specifying one or more ECS instance types
  - One ECS instance type specified
    - If the specified ECS instance type supports custom CPU options, elastic container instances are created with the customized CPU options based on the ECS instance type.
    - If the specified ECS instance type does not support custom CPU options, elastic container instances are created based on the ECS instance type while the settings of CPU options are ignored. Then, an event notification is sent to you. If the customized CPU options are invalid, elastic container instances cannot be created and an error message is sent to you.
  - Multiple ECS instance types specified

    The system tries the ECS instance types in the order that they are specified to create elastic container instances.

    - If the system is using an ECS instance type that supports custom CPU options, elastic container instances are created with the customized CPU options based on the ECS instance type.
    - If the system is using an ECS instance type that does not support custom CPU options or if the customized CPU options are invalid, elastic container instances are created based on the ECS instance type while the settings of CPU options are ignored. Then, an event notification is sent to you.

- Elastic container instances created by specifying vCPU and memory specifications

  When you use this method to create elastic container instances, the ECS instance types that support custom CPU options are preferentially used.

When you create an elastic container instance, you can use one of the following modes to customize CPU options:

- Kubernetes mode
- API mode

Kubernetes mode

You can add annotations to metadata in the configuration file of the pod to customize CPU options. Take note of the following items:

- k8s.aliyun.com/eci-cpu-option-core: the number of physical CPU cores.
- k8s.aliyun.com/eci-cpu-option-ht: the number of threads per core.

Sample code:

```
apiVersion: v1
kind: Pod
metadata:
  annotations:
    k8s.aliyun.com/eci-use-specs : "ecs.c6.2xlarge"    #Specify an ECS instance types that
supports custom CPU options.
    k8s.aliyun.com/eci-cpu-option-core: 2                 #Set the number of physical CPU c
ores to 2.
    k8s.aliyun.com/eci-cpu-option-ht: 1                    #Set the number of threads per c
ore to 1 to disable HT.
  name: nginx-test
spec:
  containers:
  - name: nginx
    image: nginx:latest
  restartpolicy: Always
```

API mode

When you call the CreateContainerGroup operation to create an elastic container instance, you can use the CpuOptionsCore and CpuOptionsThreadsPerCore parameters to customize CPU options. The following table describes the parameters. For more information, see CreateContainerGroup.

Parameter	Type	Required	Example	Description
CpuOptionsCore	Integer	No	2	The number of physical CPU cores. Only some ECS instance types allow you to customize this number.

Parameter	Type	Required	Example	Description
CpuOptionsThreadsPerCore	Integer	No	2	The number of threads per core. Only some ECS instance types allow you to customize this number. If the `CpuOptions.ThreadPerCore` parameter is set to 1, HT is disabled.

4.Optimize costs

4.1. Create a preemptible elastic container instance

Preemptible elastic container instances are cost-effective. You can bid for unused Alibaba Cloud resources to create preemptible instances. When the market price exceeds your bid price for a preemptible instance or when the resource inventory is insufficient, the resources consumed by the instance are reclaimed and the instance is released. This reduces the costs of elastic container instances in some scenarios.

Background information

Preemptible instances are ideal for stateless scenarios such as scalable web services, image rendering, big data analytics, and large-scale parallel computing. Preemptible instances are applicable to highly distributed, scalable, and fault-tolerant applications. This can help reduce costs and increase throughput. For more information, see Overview.

Basic concepts

Before you create a preemptible instance, take note of the following items:

- Billing method

  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 real-time market price and the resource inventory of the instance type is sufficient, the preemptible instance is created. After a preemptible instance is created, it enters a protection period during which the instance is billed based on the market price at the time of purchase. After the protection period ends, the instance is billed based on the real-time market price.

  > Note
  >
  > Preemptible instances are billed at lower prices compared with pay-as-you-go instances. The price of a preemptible instance fluctuates based on the supply and demand for its instance type. A preemptible instance is billed based on its actual duration of use. For more information, see the "Billing methods" section of Preemptible instances.

- Protection period

  By default, each preemptible instance has a protection period of 1 hour. You can specify the protection period when you create a preemptible instance by using Kubernetes or by calling an API operation. You can set the protection period of a preemptible instance to 0, 1, 2, 3, 4, 5, or 6 hours. A value of 0 indicates that the preemptible instance does not have a protection period and may be released at any time. Shorter protection periods allow you to obtain lower instance prices.

- Reclaim mechanism

  After the protection period of a preemptible instance ends, the system checks the market price and resource inventory of the instance type every 5 minutes. If the market price exceeds your bid price or if the resource inventory is insufficient, the instance is released.

> Note
>
> - A release event is generated 3 minutes before the resources of a preemptible instance are reclaimed.
> - After the resources of a preemptible instance are reclaimed, the information about the instance is retained, billing stops, and the instance state changes to Expired.

Precautions

When you use preemptible instances, take note of the following items:

- Select a suitable instance type and place a bid.

  You can call an API operation to query information about preemptible instances over the last 30 days. Then, you can select a suitable instance type and place a bid. The following API operations are supported:

  - DescribeSpotPriceHistory: queries the historical prices of a preemptible instance.
  - DescribeSpotAdvice: queries information such as the average release rate and discount rate of a preemptible instance.

  > Note
  >
  > When you place a bid, consider the fluctuations in market price and the price range you are comfortable with. A suitable bid is usually higher than the average market price, but still within the range of your business requirements. This way, requests for creating preemptible instances can be accepted and processed, and the created instances will not be released due to changes in market price.

- We recommend that you store important data in a storage medium that is not affected if preemptible instances are released. For example, you can use independent disks that are not to be released along with their associated instances, or use external storage resources such as Apsara File Storage NAS file systems.

Methods used to create preemptible instances

You can create a preemptible instance by specifying an Elastic Compute Service (ECS) instance type, or by specifying the number of vCPUs and memory size.

> Notice
>
> If you create a preemptible instance by specifying the number of vCPUs and memory size, at least 2 vCPUs must be specified. To create a preemptible elastic container instance that has less than 2 vCPUs, use the method of specifying an ECS instance type. In this case, we recommend that you specify an ECS instance type of the t5 or t6 instance family.

- Specify an ECS instance type

  The preemptible instance is billed at the real-time discounted pay-as-you-go price of the specified instance type.

  If the protection period is set to 2, 3, 4, 5, or 6 hours, only the following ECS instance types are supported:

  - ecs.c5.large
  - ecs.c6.large
  - ecs.g5.large, ecs.g5.xlarge, ecs.g5.2xlarge, ecs.g5.4xlarge, and ecs.g5.8xlarge
  - ecs.g6.large, ecs.g6.xlarge, ecs.g6.2xlarge, ecs.g6.4xlarge, and ecs.g6.8xlarge
  - ecs.gn6e-c12g1.3xlarge, ecs.gn6e-c12g1.12xlarge, and ecs.gn6e-c12g1.24xlarge
  - ecs.gn5-c4g1.xlarge, ecs.gn5-c8g1.2xlarge, ecs.gn5-c8g1.4xlarge, and ecs.gn5-c28g1.7xlarge
  - ecs.gn6v-c8g1.2xlarge, ecs.gn6v-c8g1.8xlarge, ecs.gn6v-c8g1.16xlarge, and ecs.gn6v-c10g1.20xlarge

- Specify the number of vCPUs and memory size

  This method yields the same effect as specifying an ECS instance type. The system automatically selects an ECS instance type that meets the specified requirements on the number of vCPUs, memory size, and price. The created preemptible instance is billed at a discount on the market price of the selected ECS instance type, not on the pay-as-you-go price of the vCPUs and memory resources of the preemptible elastic container instance.

Use Kubernetes to create a preemptible instance

You can add annotations to metadata in the configuration file of the pod to create a preemptible elastic container instance. Take note of the following annotations:

- k8s.aliyun.com/eci-spot-strategy: specifies the bidding policy for the preemptible instance. Valid values:
  - SpotWithPriceLimit: You must set the maximum hourly price that you are willing to pay for the preemptible instance.
  - SpotAsPriceGo: The system places bids based on the real-time market price.
- k8s.aliyun.com/eci-spot-price-limit: specifies the maximum hourly price of the preemptible instance. This value can be accurate to three decimal places. You must specify k8s.aliyun.com/eci-spot-price-limit when k8s.aliyun.com/eci-spot-strategy is set to SpotWithPriceLimit.
- k8s.aliyun.com/eci-spot-duration: specifies the protection period of the preemptible instance. Valid values: 0, 1, 2, 3, 4, 5, and 6. Unit: hours.

Example 1: Specify an ECS instance type, set k8s.aliyun.com/eci-spot-strategy to SpotWithPriceLimit, and set k8s.aliyun.com/eci-spot-duration.

```
apiVersion: apps/v1 # for versions before 1.8.0 use apps/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment-basic
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
      annotations:
        k8s.aliyun.com/eci-use-specs : "ecs.c5.large"  # Specify an ECS instance type.
        k8s.aliyun.com/eci-spot-strategy: "SpotWithPriceLimit"  # Set k8s.aliyun.com/eci-spot-strategy to SpotWithPriceLimit.
        k8s.aliyun.com/eci-spot-price-limit: "0.250"   # Specify a maximum hourly price for the preemptible instance.
        k8s.aliyun.com/eci-spot-duration: "2"   # Specify a protection period for the preemptible instance.
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9 # replace it with your exactly <image_name:tags>
        ports:
        - containerPort: 80
```

Example 2: Specify the number of vCPUs and memory size, and set k8s.aliyun.com/eci-spot-strategy to SpotAsPriceGo.

```
apiVersion: apps/v1 # for versions before 1.8.0 use apps/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment-basic
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
      annotations:
        k8s.aliyun.com/eci-use-specs : "2-4Gi"  # Specify the number of vCPUs and memory size. The specified number of vCPUs must be 2 or larger.
        k8s.aliyun.com/eci-spot-strategy: "SpotAsPriceGo"  # The SpotAsPriceGo bidding policy is used. This policy specifies that the system places bids based on the real-time market price.

    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9 # replace it with your exactly <image_name:tags>
        ports:
        - containerPort: 80
```

Call an API operation to create a preemptible instance

When you can call the CreateContainerGroup operation to create a preemptible elastic container instance, you can use the SpotStrategy parameter to specify the bidding policy, the SpotPriceLimit parameter to specify a maximum hourly price, and the SpotDuration parameter to specify a protection period. The following table describes these parameters. For more information, see CreateContainerGroup.

Parameter	Type	Example	Description

Parameter	Type	Example	Description
SpotStrategy	String	SpotPriceLimit	The bidding policy for the instance. Valid values: • NoSpot: The instance is a regular pay-as-you-go instance. • SpotWithPriceLimit: The instance is a preemptible instance with a user-defined maximum hourly price. • SpotAsPriceGo: The instance is a preemptible instance for which the system places bids based on the real-time market price. Default value: NoSpot.
SpotPriceLimit	Float	0.2	The maximum hourly price of the preemptible instance. This price can be accurate to three decimal places. You must specify this parameter when SpotStrategy is set to SpotWithPriceLimit.
SpotDuration	Integer	1	The protection period of the preemptible instance. Unit: hours. Valid values: 0, 1, 2, 3, 4, 5, and 6. Default value: 1.

Notice

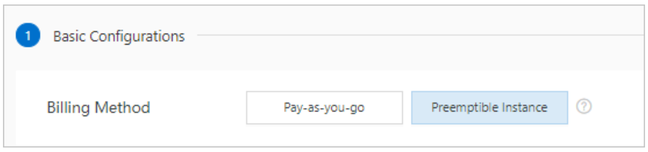
If you set SpotStrategy to SpotAsPriceGo, the maximum hourly price may reach the pay-as-you-go price when resources of the specified instance type within the specified zone are insufficient.

You can use an SDK to create a preemptible instance. The following sample code provides an example on how to use SDK for Python to create a preemptible instance:

```
#!/usr/bin/env python
#coding=utf-8
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkeci.request.v20180808.CreateContainerGroupRequest import CreateContainerGroup
Request
client = AcsClient('<accessKeyId>', '<accessSecret>', 'cn-hangzhou')
request = CreateContainerGroupRequest()
request.set_accept_format('json')
request.set_SecurityGroupId("sg-xxx")
request.set_VSwitchId("vsw-xxx")
request.set_ContainerGroupName("test-spot")
request.set_SpotStrategy('SpotAsPriceGo')
request.set_InstanceType('ecs.c5.large')
request.set_Containers([
  {
    "Image": "nginx",
    "Name": "nginx"
  }
])
response = client.do_action_with_exception(request)
print(response)
```

Use the console to create a preemptible instance

When you create a preemptible instance on the instance buy page in the Elastic Container Instance console, you can set Billing Method to **Preemptible Instance**.



1 Basic Configurations

Billing Method: Pay-as-you-go | Preemptible Instance (selected)

When you create a preemptible instance in the Elastic Container Instance console, take note of the following items:

- Preemptible instances can be created only by specifying the number of vCPUs and memory size. The specified number of vCPUs must be 2 or larger.

  The system automatically selects an ECS instance type that meets the specified requirements on the number of vCPUs, memory size, and price. The created preemptible instance is billed at a discount on the market price of the selected ECS instance type, not on the pay-as-you-go price of the vCPUs and memory resources of the preemptible elastic container instance.

- You cannot configure a bidding policy. The bidding policy is automatically set to SpotAsPriceGo. This policy specifies that the system places bids based on the real-time market price.
- You cannot configure a protection period. The protection period is automatically set to 1 hour.

4.2. Use reserved instances

If you run long-lived online applications, we recommend that you use reserved instances to offset the bills of elastic container instances. This topic describes how to use reserved instances to offset the bills of long-lived elastic container instances.

Purchase reserved instances

Reserved instances can match Elastic Compute Service (ECS) instances (excluding preemptible instances) within your account to provide a billing discount. If you use an ECS instance type to create an elastic container instance, you can use reserved instances to offset the bills of the elastic container instance. For more information, see Reserved instance overview.

Reserved instances can be applied only to elastic container instances that were created by specifying ECS instance types. Before you purchase a reserved instance, take note of the following items:

- Only some ECS instance types can be used to create elastic container instances. Make sure that the selected ECS instance types suit your business needs. For more information, see Specify an ECS instance type to create an elastic container instance.
- Reserved instances can be applied only to elastic container instances that meet the matching rules. Make sure that your elastic container instances can be matched to the reserved instances.

> Note
>
> When you purchase reserved instances, you commit to the specified terms and instance types. After a reserved instance is purchased, it automatically matches pay-as-you-go instances that have specific attributes within the term of the reserved instance. You cannot manually manage how the reserved instance is matched to pay-as-you-go instances. After a reserved instance is matched, it checks for eligible pay-as-you-go bills on an hourly basis and deducts fees based on its computing power. For more information, see Match between reserved instances and pay-as-you-go instances.

You can purchase and manage reserved instances on the **Reserved Instances** page in the ECS console. For more information, see Purchase reserved instances.

- If you want to optimize the costs of existing elastic container instances, you can find elastic container instances that were created by specifying ECS instance types, and purchase reserved instances of the specified ECS instance types.
- If you have no existing elastic container instances, select ECS instance types based on your business needs to purchase suitable reserved instances. Then, create elastic container instances based on the selected ECS instance types.

Create elastic container instances

To apply your purchased reserved instances to elastic container instances, you must specify the ECS instance types of the reserved instances when create the elastic container instances.

Kubernetes mode

If you have purchased ecs.c5.large reserved instances, you must specify the ecs.c5.large ECS instance type when you create elastic container instances.

Note

If you are using a Container Service for Kubernetes (ACK) or serverless Kubernetes (ASK) cluster and have purchased a zonal reserved instance, make sure that the reserved instance and the cluster reside within the same zone. Otherwise, the reserved instance cannot be applied to the elastic container instance. If the reserved instance and the elastic container instance are located in different zones of a region, you can change the zonal reserved instance into a regional reserved instance. For more information, see Modify a reserved instance.

You can add annotations to metadata in the configuration file of the pod to specify an ECS instance type to which your reserved instance can be applied. Sample code:

```
apiVersion: apps/v1 # for versions before 1.8.0 use apps/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment-basic
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
      annotations:
        k8s.aliyun.com/eci-use-specs : ecs.c5.large  #Specify an ECS instance type to which your reserved instance can be applied.
    spec:
    #  nodeSelector:
    #    env: test-team
      containers:
      - name: nginx
        image: nginx:1.7.9 # replace it with your exactly <image_name:tags>
        ports:
        - containerPort: 80
```

API mode

If you have purchased an ecs.c5.large reserved instance, you must set the InstanceType parameter to ecs.c5.large when you call the CreateContainerGroup operation to create an elastic container instance. The following table describes the parameter. For more information, see CreateContainerGroup.

Parameter	Type	Required	Example	Description

Parameter	Type	Required	Example	Description
InstanceType	String	No	ecs.c5.xlarge	The ECS instance types. Specify the ECS instance types to which your reserved instance can be applied. You can specify up to five ECS instance types at a time. Separate multiple instance types with commas (,). Example: ecs.c5.xlarge,ecs.g5.xlarge.

View deduction information of a reserved instance

You can view the billing details of elastic container instances and reserved instances in the Billing Management console.

> Notice
>
> In scenarios where multiple zones are specified to create elastic container instances, the created elastic container instances may be located in multiple zones. If you find that a reserved instance is not applied to offset bills as expected, check whether the reserved instance and the elastic container instance are within the same zone. If the reserved instance and the elastic container instance are located in different zones of a region, you can change the zonal reserved instance into a regional reserved instance. For more information, see Modify a reserved instance.

- View bills of elastic container instances

  On the **Bills** page, click the **Details** tab. You can view the consumption details by elastic container instance. No bills are generated for the fees of elastic container instances that are offset by reserved instances.

- View the bills of reserved instances

  On the **Manage Reserved Instances** page, you can view the deduction details of the reserved instances.

  The usage details of reserved instances show which elastic container instances had fees deducted by the reserved instances each billing cycle (hour). The deducted duration is calculated by using the following formula: Computing power × Number of hours. Reserved instances are deducted by computing power. One unit of computing power is equivalent to one vCPU.

4.3. Use savings plans

If you run long-term online applications, we recommend that you use savings plans to offset the bills of elastic container instances. This topic describes how to use savings plans to offset the bills of long-term elastic container instances.

Purchase savings plans

A savings plan is a discount plan that allows you to receive billing discounts in exchange for a commitment to use a consistent amount(measured in USD/hour) of resources over a one-year or three-year period. After you purchase a savings plan, the hourly bills of your pay-as-you-go instances are covered up to the amount of the plan. For more information, see Overview.

Before you purchase a savings plan, take note of the following items:

- Limits on using savings plans to offset the bills of elastic container instances

  Savings plans can be divided into the general-purpose and ECS compute types. ECS compute savings plans can be used to offset the bills only of elastic container instances created by using specified ECS instance types. General-purpose savings plans are not limited.

  > ? Note
  >
  > We recommend that you purchase general-purpose savings plans. This way, you do not need to consider limits such as regions and instance families and can enjoy higher flexibility.

- Hourly commitment

  Savings plans offset the bills of your elastic container instances. When a hourly bill is being calculated, resource usage within your commitment is calculated and deducted based on the savings plan discount. Resource usage beyond your commitment is billed at the regular pay-as-you-go unit price.

  For example, the hourly bill of your elastic container instance is USD 10/hour. You want to purchase a three-year general-purpose savings plan by using the All Upfront payment option. The discount is 54.5% off. When you purchase the corresponding savings plan, the hourly commitment is USD 4.55 (10 × 0.455 = 4.55) to offset the bills of the elastic container instance.

  > ? Note
  >
  > The prices used in this example are for demonstration only. For the actual prices and discounts, see the Discount Details page.

You can learn more information and purchase savings plans on the **Savings Plan** page in the ECS console. You can choose the recommended solutions or calculate by yourself based on your business needs.

- If you want to optimize the costs of existing elastic container instances, you can go to the Recommended page for system recommendations.
- If you have no existing elastic container instances, you can calculate bills of an elastic container instance and then calculate the hourly commitment based on the discount. Then, you can go to the Savings Plan page to purchase savings plans.

For more information, see Purchase and apply savings plans.

View deduction information of a savings plan

After you purchase a savings plan, the system matches the savings plan with elastic container instances to offset bills. You do not need to manually match the resources. You can view the deduction information of the savings plan in the Billing Management console.

- View bills of elastic container instances

On the **Bills** page, click the **Details** tab. Then, you can view the consumption details by elastic container instance. For the fees of elastic container instances that are offset by savings plans, no bills are generated.

- View the effect of savings plans

  On the **Overview and Discount Details** pages of savings plans, you can view information such as overview, details, usage, and coverage of savings plans.

5.Optimize inventory

5.1. Specify multiple zones to create an elastic container instance

In specific cases, you may need to create a large number of elastic container instances to handle bursts of traffic or process jobs. Resources of the specified instance type may be insufficient within the specified zone or the specified vSwitch may have exhausted all of its IP addresses. In these cases, the elastic container instances cannot be created. To ensure that resources are sufficient to create the elastic container instances, you can specify multiple zones.

Prerequisites

vSwitches are created in different zones of the virtual private cloud (VPC) where you want to create an elastic container instance.

- For information about how to create a vSwitch, see the "Create a vSwitch" section in Work with vSwitches.
- For information about the regions and zones where Elastic Container Instance is available, see Regions and zones.

Background information

When you create an elastic container instance (pod), you can specify multiple zones by specifying multiple vSwitches. The system then distributes the request to create the elastic container instance across all the specified zones at random. If requested resources are insufficient within one zone, the system attempts to create the instance in a different zone.

When you specify multiple zones (vSwitches), take note of the following items:

- The specified vSwitches must belong to the same VPC.
- You can specify up to 10 vSwitches.

You can specify multiple instance types across multiple zones to further improve the success rate of creating elastic container instances. For information about how to create an elastic container instance by specifying multiple instance types, see Specify multiple instance types to create an elastic container instance.

Kubernetes mode

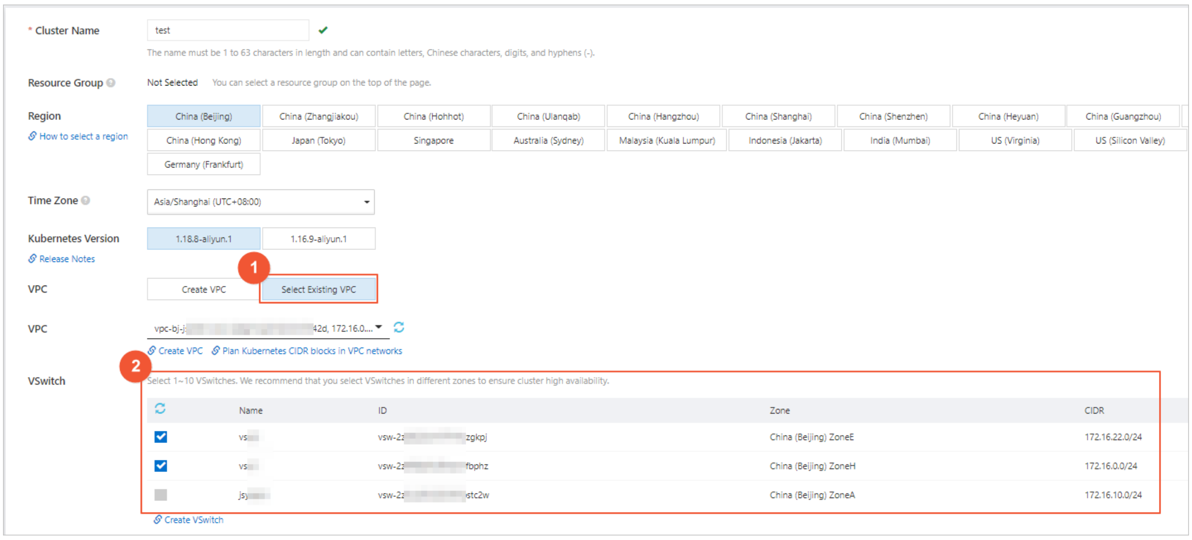
You can specify multiple zones for a cluster or an elastic container instance.

- If you specify multiple zones for a cluster, the zone settings take effect on all of the elastic container instances within the cluster.
- If you specify multiple zones for an elastic container instance, the zone settings take effect only on the elastic container instance.

Different cluster types support different methods with which to specify multiple zones.

- **Specify multiple zones when you create a cluster**

Before you create a cluster, we recommend that you create multiple vSwitches in multiple zones within the VPC in which to create the cluster. When you create the cluster, you can select the VPC and select multiple vSwitches to specify multiple zones. The following figure shows how to create a serverless Kubernetes (ASK) cluster in the ACK console.



* Cluster Name: test ✓

The name must be 1 to 63 characters in length and can contain letters, Chinese characters, digits, and hyphens (-).

Resource Group: Not Selected You can select a resource group on the top of the page.

Region: China (Beijing) | China (Zhangjiakou) | China (Hohhot) | China (Ulanqab) | China (Hangzhou) | China (Shanghai) | China (Shenzhen) | China (Heyuan) | China (Guangzhou) | China (Hong Kong) | Japan (Tokyo) | Singapore | Australia (Sydney) | Malaysia (Kuala Lumpur) | Indonesia (Jakarta) | India (Mumbai) | US (Virginia) | US (Silicon Valley) | Germany (Frankfurt)

How to select a region

Time Zone: Asia/Shanghai (UTC+08:00)

Kubernetes Version: 1.18.8-aliyun.1 | 1.16.9-aliyun.1

Release Notes

VPC: Create VPC | Select Existing VPC

VPC: vpc-bj-j[redacted]42d, 172.16.0....

Create VPC Plan Kubernetes CIDR blocks in VPC networks

VSwitch: Select 1~10 VSwitches. We recommend that you select VSwitches in different zones to ensure cluster high availability.

	Name	ID	Zone	CIDR
☑	vs[redacted]	vsw-2z[redacted]zgkpj	China (Beijing) ZoneE	172.16.22.0/24
☑	vs[redacted]	vsw-2z[redacted]fbphz	China (Beijing) ZoneH	172.16.0.0/24
□	jsy[redacted]	vsw-2z[redacted]stc2w	China (Beijing) ZoneA	172.16.10.0/24

Create VSwitch

- **Specify multiple zones for an existing cluster**

  You can specify multiple zones by modifying eci-profile or modifying the value of an environment variable based on the version of Virtual Kubelet.

  - Modify eci-profile (The version of Virtual Kubelet is v2.0.0.90-15deb126e-aliyun or later)

    You can modify the eci-profile ConfigMap of an existing cluster and append vSwitch IDs to the `data.vSwitchIds` field in eci-profile to specify multiple zones. Perform the following steps:

    a. Edit eci-profile.

    ```
    kubectl edit cm -n kube-system eci-profile
    ```

    b. Modify data.vswitch.

    Append vSwitch IDs. Separate multiple vSwitch IDs with commas (,).

    ```
    data:
      enableClusterIp: "true"
      enableHybridMode: "false"
      enablePrivateZone: "false"
      resourceGroupId: ""
      securityGroupId: sg-2ze0b9o8pjjzts4h****
      selectors: ""
      vSwitchIds: vsw-2zeet2ksvw7f14ryz****,vsw-2ze94pjtfuj9vaymf****
      vpcId: vpc-2zeghwzptn5zii0w7****
    ```

    c. Save the modifications and exit.

- Modify the value of an environment variable (The version of Virtual Kubelet is earlier than v2.0.0.90-15deb126e-aliyun)

  > **Note**
  >
  > We recommend that you upgrade Virtual Kubelet to the latest version to use new features.

  To specify multiple zones, you can modify the value of the `ECI_VSWITCH` environment variable for a Deployment or StatefulSet based on how you deploy Virtual Kubelet. In the following example, a cluster within which Virtual Kubelet is deployed as a Deployment is used:

  a. Edit the Deployment.

  ```
  kubectl -n kube-system edit deployment/virtual-node-controller
  ```

  b. Modify ECI_VSWITCH. Save the modifications and exit.

  Append vSwitch IDs to the `ECI_VSWITCH` environment variable. Separate multiple vSwitch IDs with commas (,).

  ```
  - name:  ECI_VSWITCH
    value:  vsw-bp1xpiowfm5vo8o3c****,vsw-bp1rkyjgr1xwoho6k****
  ```

  c. Save the modifications and exit.

- **Specify multiple zones for an elastic container instance**

  You can add annotations to metadata in the configuration file of a pod to specify multiple zones. Add the following annotations:

  - k8s.aliyun.com/eci-vswitch: the IDs of vSwitches used to specify multiple zones.
  - k8s.aliyun.com/eci-schedule-strategy: the cross-zone scheduling policy. Valid values: VSwitchOrdered and VSwitchRandom. VSwitchOrdered indicates that resources are scheduled to create elastic container instances within specified zones in the order that the switches are specified. VSwitchRandom indicates that resources are scheduled to create elastic container instances within specified zones at random.

  Sample code:

  ```
  apiVersion: v1
  kind: Pod
  metadata:
    annotations:
      k8s.aliyun.com/eci-vswitch: "vsw-bp1xpiowfm5vo8o3c****,vsw-bp1rkyjgr1xwoho6k****"
  #Specify multiple vSwitch IDs.
      k8s.aliyun.com/eci-schedule-strategy: "VSwitchOrdered"   #Specify the cross-zone sche
  duling policy.
    name: nginx-test
  spec:
    containers:
    - name: nginx
      image: nginx:latest
  ```

API mode

When you call the CreateContainerGroup operation to create an elastic container instance, you can use the VSwitchId parameter to specify multiple zones. The following table describes this parameter. For more information, see CreateContainerGroup.

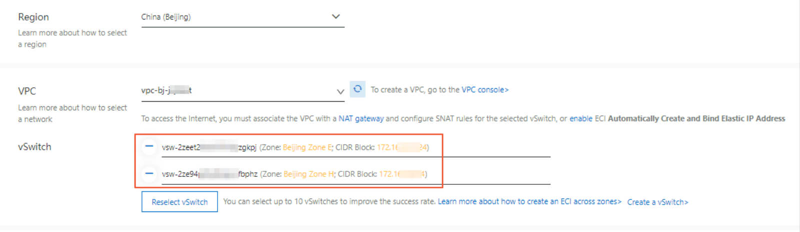
Parameter	Type	Required	Example	Description
VSwitchId	String	Yes	vsw-bp1xpiowfm5vo8o3c****,vsw-bp1rkyjgr1xwoho6k****	The IDs of vSwitches. You can specify up to 10 vSwitch IDs at a time. Separate multiple vSwitch IDs with commas (,). Example: vsw-***,vsw-***.

Note

We recommend that you specify multiple instance types across multiple zones to create elastic container instances. You can use the VSwitchId parameter to specify multiple zones and use the InstanceType parameter to specify multiple instance types.

Console mode

When you create an elastic container instance by using the elastic container instance buy page in the Elastic Container Instance console, you can select multiple vSwitches to specify multiple zones.



Region: China (Beijing)
Learn more about how to select a region

VPC: vpc-bj-j[redacted]t
To create a VPC, go to the VPC console>
Learn more about how to select a network
To access the Internet, you must associate the VPC with a NAT gateway and configure SNAT rules for the selected vSwitch, or enable ECI **Automatically Create and Bind Elastic IP Address**

vSwitch:
- vsw-2zeet2[redacted]zgkpj (Zone: Beijing Zone E; CIDR Block: 172.1[redacted]24)
- vsw-2ze94[redacted]fbphz (Zone: Beijing Zone H; CIDR Block: 172.1[redacted]4)

You can select up to 10 vSwitches to improve the success rate. Learn more about how to create an ECI across zones> Create a vSwitch>

5.2. Specify multiple instance types to create an elastic container instance

When you create a large number of elastic container instances, some resources in the specified region and zone may be insufficient. To ensure that resources are sufficient to create the elastic container instances, we recommend that you specify multiple instance types across multiple zones for the pods.

Background information

You can specify multiple Elastic Compute Service (ECS) instance types in the k8s.aliyun.com/eci-use-specs annotation in the configuration file of a pod to create an elastic container instance. If the instance cannot be created due to insufficient resources of a specified ECS instance type, the system attempts to use the next instance type specified in the annotation to create the elastic container instance. When the system is attempting to create the elastic container instance, the pod remains in the Pending state. We recommend that you specify multiple ECS instance types to ensure that resources are sufficient to create the elastic container instance.

Note

The specified ECS instance types take effect at the pod level and affect only how a single pod is created.

You can specify multiple instance type across multiple zones to further improve the success rate of creating elastic container instances. For more information, see Specify multiple zones to create an elastic container instance.

Kubernetes mode

You can add annotations to metadata in the configuration file of the pod to specify multiple ECS instance types. Specify multiple ECS instance types in the order that they are to be prioritized for use. Sample code:

```
apiVersion: apps/v1 # for versions before 1.8.0 use apps/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment-basic
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
      annotations:
        k8s.aliyun.com/eci-use-specs: ecs.c5.large,ecs.c6.large,ecs.sn1ne.large,ecs.g5.larg
e  #Specify ECS instance types.
    spec:
    #  nodeSelector:
    #    env: test-team
      containers:
      - name: nginx
        image: nginx:1.7.9 # replace it with your exactly <image_name:tags>
        ports:
        - containerPort: 80
      nodeName: virtual-kubelet
  #Specify a virtual node name to schedule the pod to Elastic Container Instance in a Conta
iner Service for Kubernetes (ACK) cluster.
```

API mode

When you call the CreateContainerGroup operation to create an elastic container instance, you can use the InstanceType parameter to specify one or more ECS instance types. The following table describes the parameter. For more information, see CreateContainerGroup.

Parameter	Type	Required	Example	Description
InstanceType	String	No	ecs.c5.xlarge,ecs.g5.xlarge	The ECS instance types. You can specify up to five ECS instance types at a time. Separate multiple instance types with commas (,). Example: ecs.c5.xlarge,ecs.g5.xlarge.

Note

We recommend that you specify multiple instance types across multiple zones to create elastic container instances. You can use the VSwitchId parameter to specify multiple zones and use the InstanceType parameter to specify multiple instance types.

6.Use tags to manage elastic container instances

Each tag is a key-value pair and can be used to mark elastic container instances. You can use tags to manage elastic container instances by group to perform searches and batch operations with ease.

Background information

If you use or manage elastic container instances by calling API operations or by using the Elastic Container Instance console, you can use tags to group instances that share the same management or business requirements for easy search and operations.

When you use tags, take note of the following limits:

- Each tag consists of a tag key and a tag value. The tag key must be unique.
- Tag information cannot be shared across regions.
- Up to 20 tags can be added to a single elastic container instance.
- Tags cannot be separately deleted.

> Note
>
> When an elastic container instance is deleted, the tags added to the instance are also deleted.

Create an elastic container instance that has tags

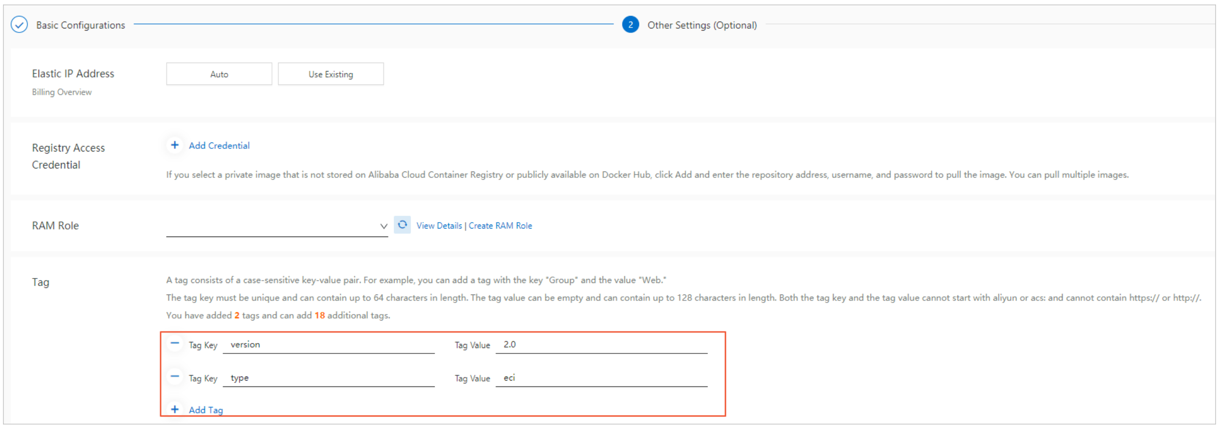
Call an API operation

When you call the CreateContainerGroup operation to create an elastic container instance, you can use the tag-related parameters to specify tags. The following table describes the parameters. For more information, see CreateContainerGroup.

Parameter	Type	Example	Description
Tag.N.Key	String	version	The key of tag N. Valid values of N: 1 to 20. The tag key cannot be an empty string and must be unique. It can be up to 64 characters in length and cannot start with acs: or aliyun. It cannot contain http:// or https://.
Tag.N.Value	String	2.0	The value of tag N. Valid values of N: 1 to 20. The tag value can be an empty string. It can be up to 128 characters in length. It cannot start with acs: or contain http:// or https://.

Use the console

When you create elastic container instances on the instance buy page in the Elastic Container Instance console, you can add tags to the instances in the **Other Settings** step, as shown in the following figure.



✓ Basic Configurations —— 2 Other Settings (Optional)

Elastic IP Address
Billing Overview
Auto | Use Existing

Registry Access Credential
+ Add Credential
If you select a private image that is not stored on Alibaba Cloud Container Registry or publicly available on Docker Hub, click Add and enter the repository address, username, and password to pull the image. You can pull multiple images.

RAM Role
View Details | Create RAM Role

Tag
A tag consists of a case-sensitive key-value pair. For example, you can add a tag with the key "Group" and the value "Web."
The tag key must be unique and can contain up to 64 characters in length. The tag value can be empty and can contain up to 128 characters in length. Both the tag key and the tag value cannot start with aliyun or acs: and cannot contain https:// or http://.
You have added 2 tags and can add 18 additional tags.

— Tag Key: version   Tag Value: 2.0
— Tag Key: type   Tag Value: eci
+ Add Tag

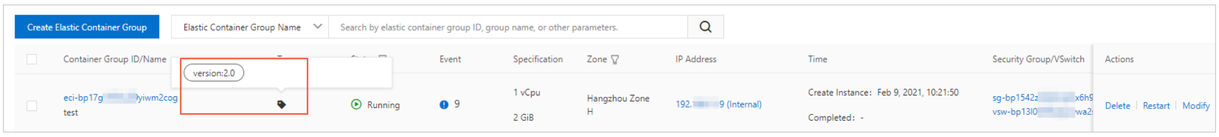
Query the tags of an elastic container instance

Call an API operation

When you call the DescribeContainerGroups operation to query elastic container instances, the Tags response parameters indicate the tags of the instances. The Key response parameter indicates the tag key, and the Value response parameter indicates the tag value. For more information, see DescribeContainerGroups.

Use the console

Go to the **Elastic Container Instance** page in the Elastic Container Instance console. Find the instance whose tags you want to query and move the pointer over the icon in the Tag column to view the tags of the instance, as shown in the following figure.



Create Elastic Container Group | Elastic Container Group Name | Search by elastic container group ID, group name, or other parameters.

Container Group ID/Name			Event	Specification	Zone	IP Address	Time	Security Group/VSwitch	Actions
eci-bp17g[redacted]yiwm2cog test	version:2.0	Running	9	1 vCpu 2 GiB	Hangzhou Zone H	192.[redacted]9 (Internal)	Create Instance: Feb 9, 2021, 10:21:50 Completed: -	sg-bp1542z[redacted]x6h9 vsw-bp13l0[redacted]wa2	Delete | Restart | Modify

Use tags to filter elastic container instances

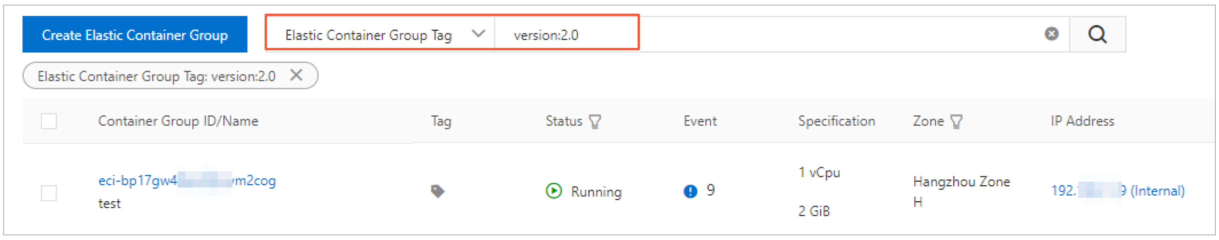
After you add tags to elastic container instances, you can use the tags to search for or authenticate to the instances.

Call an API operation

When you call the DescribeContainerGroups operation to query elastic container instances, you can use the tag-related parameters to filter instances. For more information, see DescribeContainerGroups.

Use the console

On the **Elastic Container Instance** page in the Elastic Container Instance console, you can use tags to filter elastic container instances.



Create Elastic Container Group | Elastic Container Group Tag | version:2.0

Elastic Container Group Tag: version:2.0 ✕

□	Container Group ID/Name	Tag	Status	Event	Specification	Zone	IP Address
□	eci-bp17gw4[redacted]m2cog test		Running	9	1 vCpu 2 GiB	Hangzhou Zone H	192.[redacted]9 (Internal)

Note

If you use a Resource Access Management (RAM) user to manage Elastic Container Instance resources, you can use tags to limit users to perform operations only on resources that have specific tags. For more information, see Authenticate RAM users based on tags.

Update the tags of an elastic container instance

Before you update tags for an elastic container instance, you can first query the tags that are added to the instance. When you update tags for an elastic container instance, you can only update the existing tags or add new tags. Tags cannot be deleted from elastic container instances and the number of tags on each elastic container instance cannot be reduced.

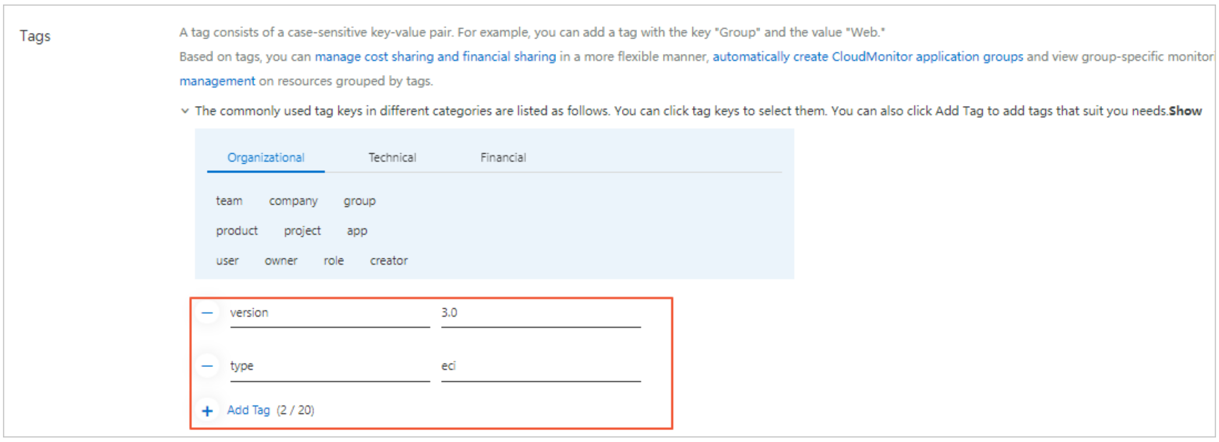
Call an API operation

When you call the UpdateContainerGroup operation to update an elastic container instance, you can use the tag-related parameters to update tags. The following table describes the parameters. For more information, see UpdateContainerGroup.

Parameter	Type	Example	Description
Tag.N.Key	String	version	The key of tag N. Valid values of N: 1 to 20. The tag key cannot be an empty string and must be unique. It can be up to 64 characters in length and cannot start with acs: or aliyun. It cannot contain http:// or https://.
Tag.N.Value	String	2.0	The value of tag N. Valid values of N: 1 to 20. The tag value can be an empty string. It can be up to 128 characters in length. It cannot start with acs: or contain http:// or https://.

Use the console

On the **Elastic Container Instance** page in the Elastic Container Instance console, you can select a created instance and update its tags, as shown in the following figure.



Tags

A tag consists of a case-sensitive key-value pair. For example, you can add a tag with the key "Group" and the value "Web."

Based on tags, you can manage cost sharing and financial sharing in a more flexible manner, automatically create CloudMonitor application groups and view group-specific monitor management on resources grouped by tags.

The commonly used tag keys in different categories are listed as follows. You can click tag keys to select them. You can also click Add Tag to add tags that suit you needs.**Show**

version	3.0
type	eci

Add Tag (2 / 20)