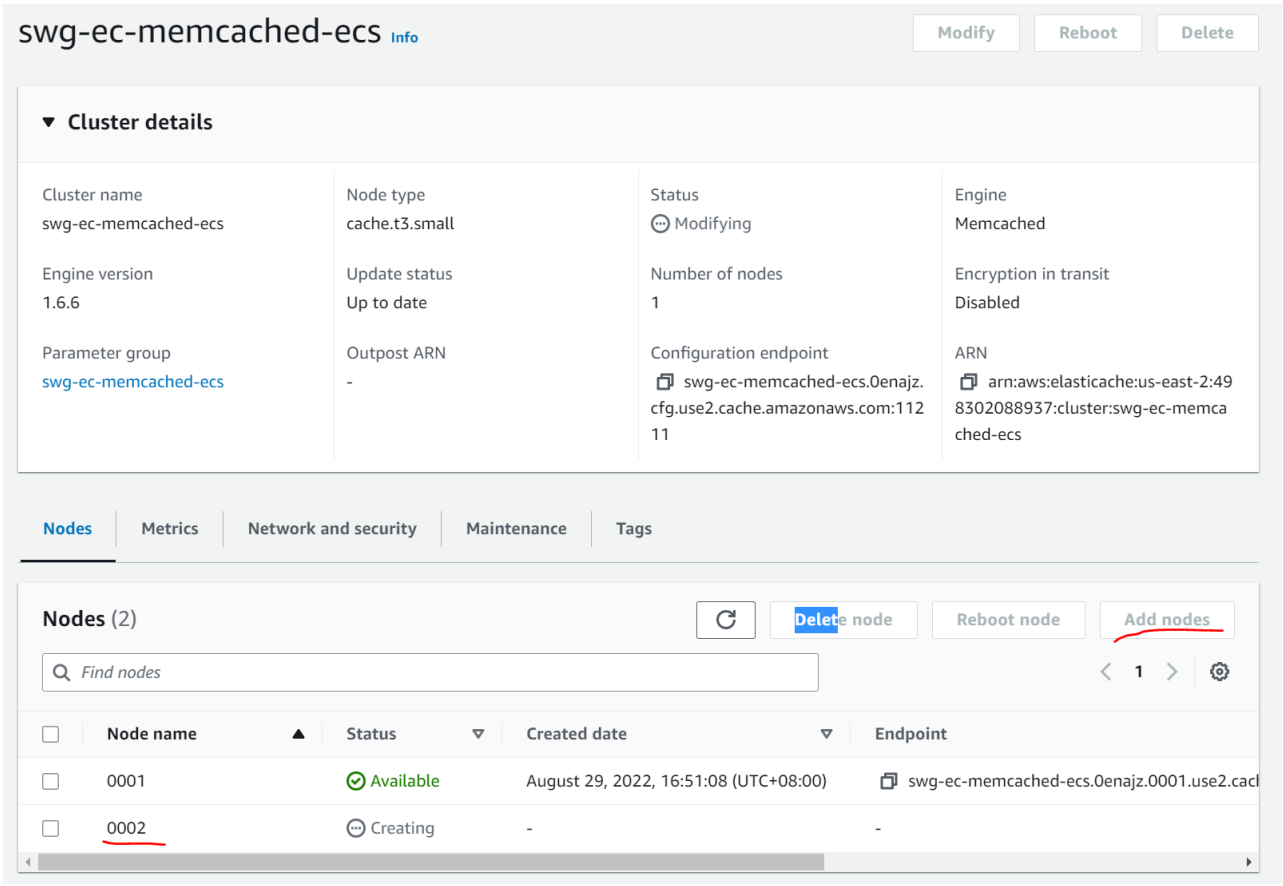
修改memcached的ip

加一个新的node，把老的node删掉



reconnect一直都是239

因为没有unpdate config的行为

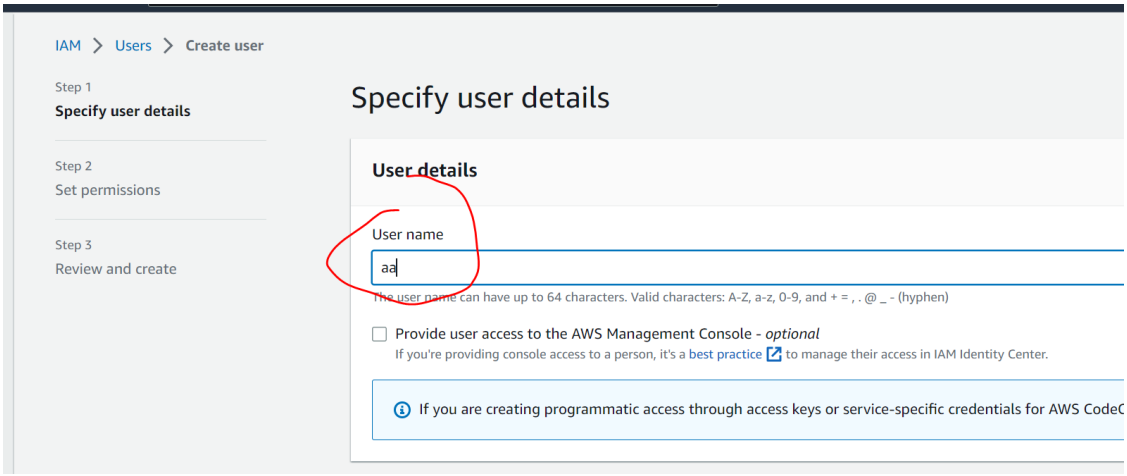
cron job实现？？

怎么登ztna ecs container:

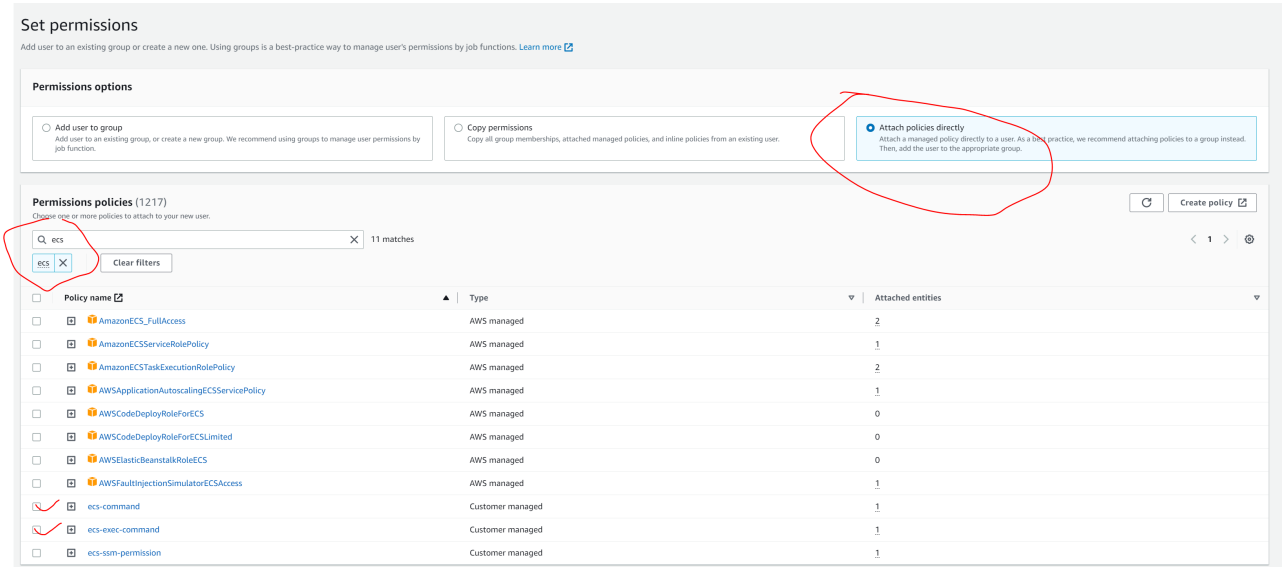
1.main.tf中加enable

2. aws console的i am服务

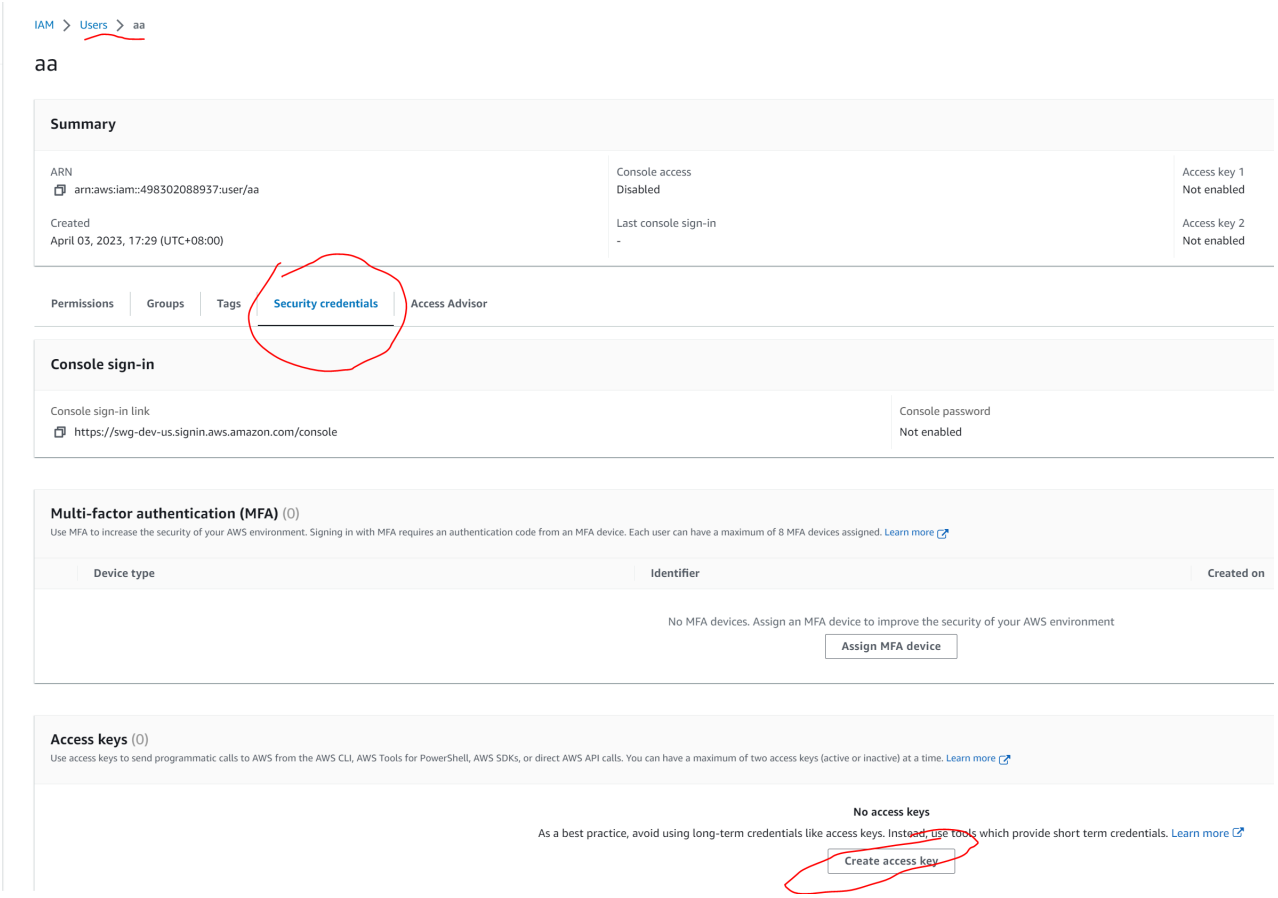
add user



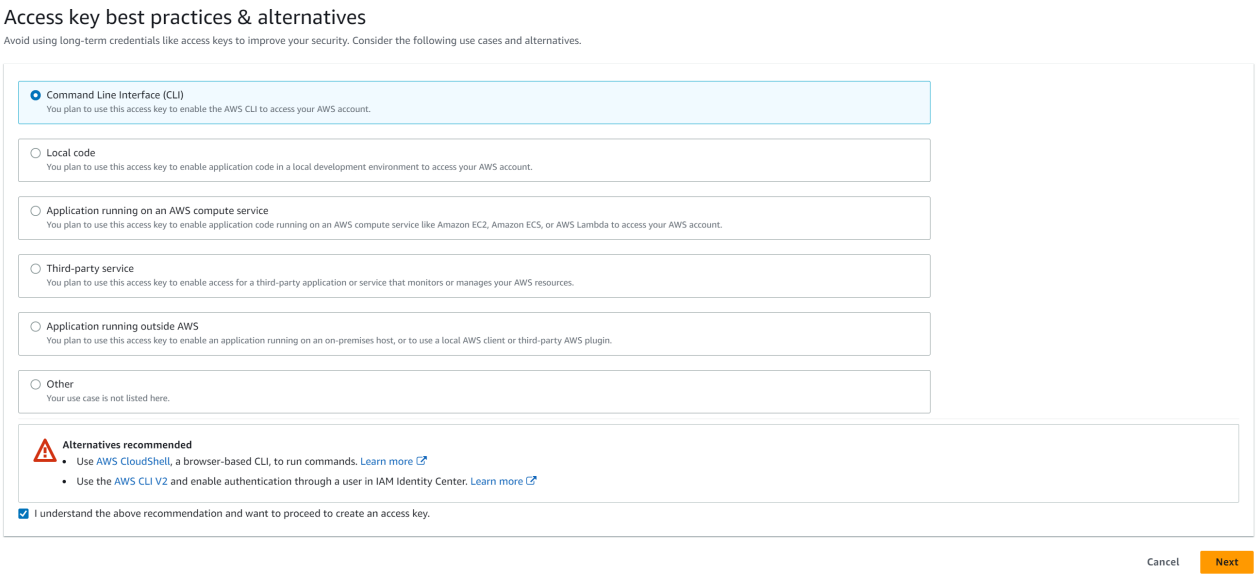
attach policy



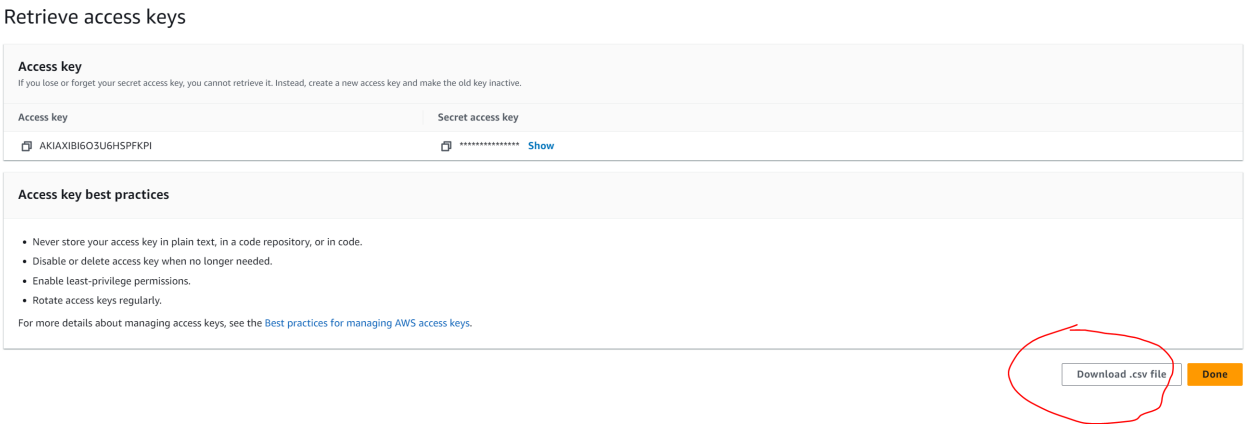
点开user create access key

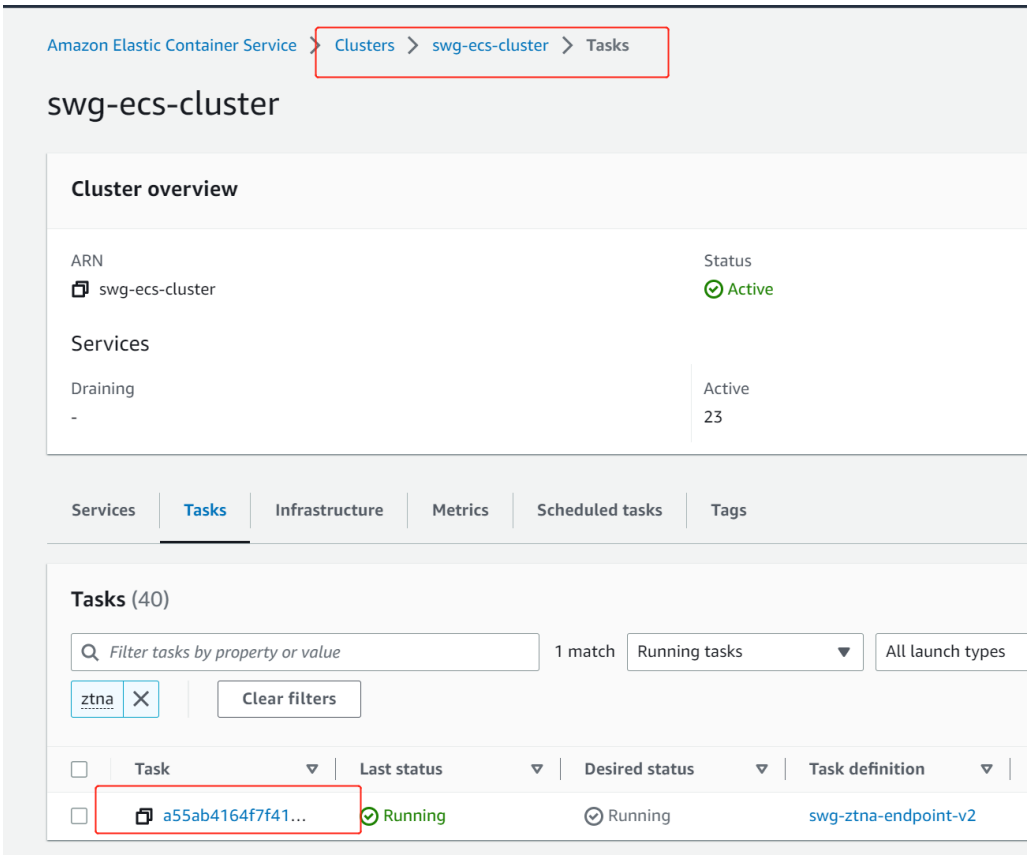

IAM > Users > aa
aa
Summary
ARN
arn:aws:iam::498302088937:user/aa
Console access
Disabled
Access key 1
Not enabled
Created
April 03, 2023, 17:29 (UTC+08:00)
Last console sign-in
-
Access key 2
Not enabled
Permissions
Groups
Tags
Security credentials
Access Advisor
Console sign-in
Console sign-in link
https://swg-dev-us.signin.aws.amazon.com/console
Console password
Not enabled
Multi-factor authentication (MFA) (0)
Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. Learn more
Device type
Identifier
Created on
No MFA devices. Assign an MFA device to improve the security of your AWS environment
Assign MFA device
Access keys (0)
Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more
No access keys
As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. Learn more
Create access key

选CLI


Access key best practices & alternatives
Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.
Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.
Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.
Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
Application running outside AWS
You plan to use this access key to enable an application running on an on-premises host, or to use a local AWS client or third-party AWS plugin.
Other
Your use case is not listed here.
Alternatives recommended
Use AWS CloudShell, a browser-based CLI, to run commands. Learn more
Use the AWS CLI V2 and enable authentication through a user in IAM Identity Center. Learn more
I understand the above recommendation and want to proceed to create an access key.
Cancel
Next

下载后保存





安装aws cli和session manager

https://docs.aws.amazon.com/zh_cn/cli/latest/userguide/getting-started-install.html

https://docs.aws.amazon.com/systems-manager/latest/userguide/session-manager-working-with-install-plugin.html

```
[root@localhost ~]# /usr/local/bin/aws ecs execute-command --cluster swg-ecs-cluster --task a55ab4164f7f4132880cd61a97c98d
32  --interactive --command /bin/bash

The Session Manager plugin was installed successfully. Use the AWS CLI to start a session.


You must specify a region. You can also configure your region by running "aws configure".
[root@localhost ~]#
```

为aws cli配置credential

创建~/.aws/credentials文件：

> [default]

> aws_access_key_id=<value from AWS access portal>
> aws_secret_access_key=<value from AWS access portal>

创建~/.aws/config文件：

> [default]
> region = your_aws_region

找ecs service对应的task

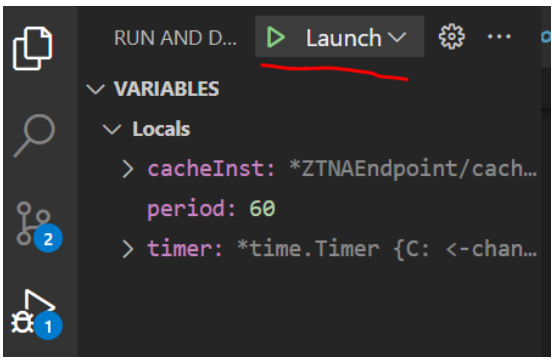
/usr/local/bin/aws ecs execute-command --cluster swg-ecs-cluster --task a55ab4164f7f4132880cd61a97c98d32 --interactive --command /bin/bash

可能是并发的问题，一个在重连，一个在new instacne

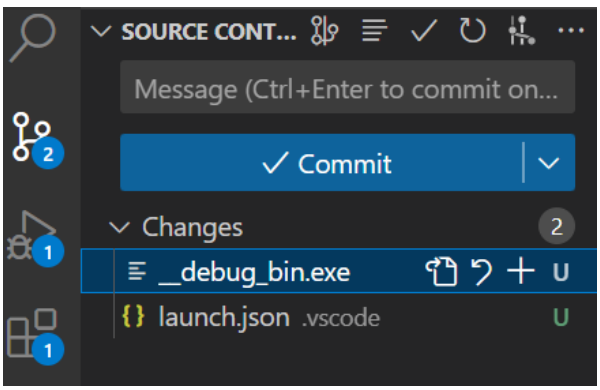
ping会有error,如果给个错误的地址，也能ping通，要set server

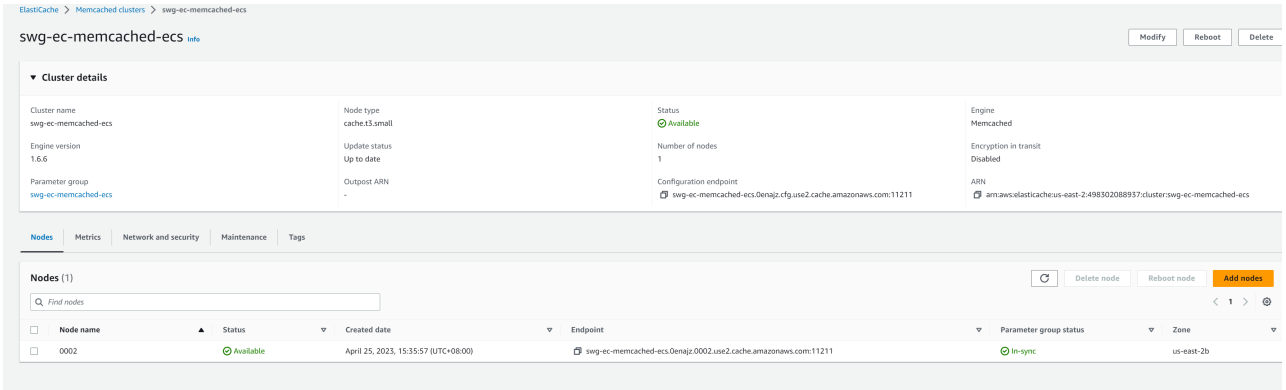
要么加锁，要么set server？按policy proxy的重构下

先到main.go



第一次因为没有json文件会报错，有json文件后就可以调试，会生成下面文件





ElastiCache > Memcached clusters > swg-ec-memcached-ecs > Modify

Modify swg-ec-memcached-ecs Info

Cluster info

Use the following options to configure the cluster.

Name

The name of the cluster that contains the primary and read-replica nodes.

swg-ec-memcached-ecs

The name is required, can have up to 40 characters and must not contain spaces.

Description - *optional*

Description

Cluster settings

Use the following options to configure the cluster.

Engine version

Version compatibility of the Memcached engine that will run on your nodes.

1.6.6

Port

The port number that nodes accept connections on.

11211

Parameter groups

Parameter groups control the runtime properties of your nodes and clusters.

swg-ec-memcached-ecs

Node type

cache.t3.small

Connectivity

Choose the IP version(s) this cluster will support. Then select an existing subnet group or create a new one.

Subnet groups

swg-ec-memcached-ecs

Associated subnets (3)

Modify

Availability Zone	Subnet ID	CIDR block (IPv4)
us-east-2a	subnet-0d06f354e4896dae5	10.154.30.192/26
us-east-2b	subnet-0443b4a6594aa2494	10.154.31.0/26
us-east-2c	subnet-0b6da3ae1a9cb80dd	10.154.31.64/26

Security

Use the following section to configure network security and data security for your cluster.

Encryption in transit

Disabled

Selected security groups (1)

A security group acts like a firewall that controls network access to your clusters.

Manage

Group ID	Name
sg-0e40866baa76258f3	swg-internal-all

Maintenance

Configure maintenance settings for the cluster.

Maintenance start day

Specify the start day for updates such as patching an operating system, updating drivers, and installing software or patches.

Sunday

Maintenance start time

Specify the start time for updates.

01:00 UTC

This is a 24h time format (hh:mm).

Maintenance duration

Specify the duration for updates.

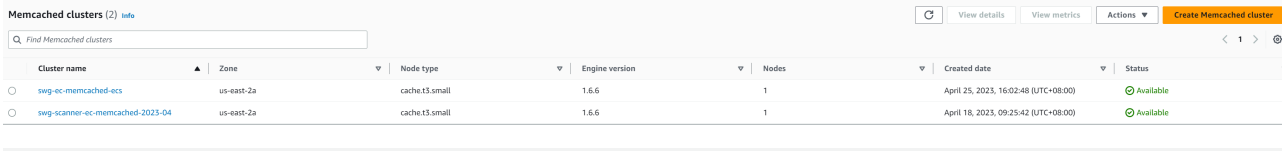
1 hour(s)

Topic for Amazon SNS notification

Choose an SNS topic from the list, or enter the Amazon Resource Name (ARN) for an existing topic. If no topic is chosen, no notifications are sent.

Disable notifications

删掉cluster后重建，02分时重建完成



2023-04-25T16:02:13.587+08:00 {"level":"info","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:186],cacheInst is memcache client","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:02:13Z","version":"3.0"}

2023-04-25T16:02:16.588+08:00 {"level":"error","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:200],memcache need reconnect. err: memcache: connect timeout to 10.154.31.61:11211","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:02:16Z","version":"3.0"}

2023-04-25T16:02:16.588+08:00 {"level":"info","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:205],cache need reconnect.","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:02:16Z","version":"3.0"}

2023-04-25T16:02:16.590+08:00 {"level":"info","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:218],new cacheInst is memcache client","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:02:16Z","version":"3.0"}

2023-04-25T16:02:16.590+08:00 {"level":"error","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:230],cache reconnect failed. err: memcache: no servers configured or available","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:02:16Z","version":"3.0"}

2023-04-25T16:03:13.588+08:00 {"level":"info","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:186],cacheInst is memcache client","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:03:13Z","version":"3.0"}

2023-04-25T16:03:16.590+08:00 {"level":"error","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:200],memcache need reconnect. err: memcache: connect timeout to 10.154.31.61:11211","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:03:16Z","version":"3.0"}

2023-04-25T16:03:16.590+08:00 {"level":"info","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:205],cache need reconnect.","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:03:16Z","version":"3.0"}

2023-04-25T16:03:16.592+08:00 {"level":"info","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:218],new cacheInst is memcache client","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:03:16Z","version":"3.0"}

2023-04-25T16:03:16.592+08:00 {"level":"error","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:230],cache reconnect failed. err: memcache: no servers configured or available","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:03:16Z","version":"3.0"}

2023-04-25T16:04:13.588+08:00 {"level":"info","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:186],cacheInst is memcache client","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:04:13Z","version":"3.0"}

2023-04-25T16:04:16.590+08:00 {"level":"error","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:200],memcache need reconnect. err: memcache: connect timeout to 10.154.31.61:11211","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:04:16Z","version":"3.0"}

2023-04-25T16:04:16.590+08:00 {"level":"info","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:205],cache need reconnect.","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:04:16Z","version":"3.0"}

2023-04-25T16:04:16.592+08:00 {"level":"info","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:218],new cacheInst is memcache client","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:04:16Z","version":"3.0"}

2023-04-25T16:04:16.592+08:00 {"level":"error","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:230],cache reconnect failed. err: memcache: no servers configured or available","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:04:16Z","version":"3.0"}

2023-04-25T16:05:13.589+08:00 {"level":"info","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:186],cacheInst is memcache client","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:05:13Z","version":"3.0"}

2023-04-25T16:05:16.591+08:00 {"level":"error","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:200],memcache need reconnect. err: memcache: connect timeout to 10.154.31.61:11211","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:05:16Z","version":"3.0"}

2023-04-25T16:05:16.591+08:00 {"level":"info","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:205],cache need reconnect.","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:05:16Z","version":"3.0"}

2023-04-25T16:05:16.596+08:00 {"level":"info","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:218],new cacheInst is memcache client","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:05:16Z","version":"3.0"}

2023-04-25T16:05:16.596+08:00 {"level":"info","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:228],cache reconnect success.","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:05:16Z","version":"3.0"}

2023-04-25T16:06:13.589+08:00 {"level":"info","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:186],cacheInst is memcache client","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:06:13Z","version":"3.0"}

2023-04-25T16:06:13.590+08:00 {"level":"debug","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:197],cache is normal","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:06:13Z","version":"3.0"}

2023-04-25T16:06:13.590+08:00 {"level":"debug","msg":"ZTNAEndpoint/cache.CheckCacheTask[cache.go:235],cache is normal, don't need to reconnect","repository":"Commercial-TMWS/TMWS_ZTNA_Endpoint","time":"2023-04-25T08:06:13Z","version":"3.0"}