

SAP

Best Practices

Best Practices

SAP HANA HA Cross-Zone with SLES HAE

SAP HANA HA Cross-Zone with SLES HAE

- Solution Overview
 - SAP HANA System Replication
 - HAE of SLES
 - Architecture Overview
 - Network Design
- Infrastructure Preparation
 - Infrastructure List
 - Creating VPC
 - Creating ECS Instances
 - Creating ENIs and binding to ECS instances
 - Creating NAT Gateway and configure SNAT entry
 - Creating STONITH device and Virtual IP Resource Agent
- Software Preparation
 - Software List
 - SLES for SAP HAE Installation
 - SAP HANA Installation
 - SAP Host Agent Installation
- Configuring SAP HANA System Replication
- Configuring HAE for SAP HANA
 - Configuration of Corosync
 - Configuration of Pacemaker
 - Verify the HA take over
- Example
 - Example Cluster Configuration
 - Example for `/etc/corosync/corosync.conf`
- Reference

Version Control:

Version	Revision Date	Types Of Changes	Effective Date
---------	---------------	------------------	----------------

1.0			2018/3/7
1.1	2018/7/04	Add corosync and cluster configuration example	2018/7/04

Solution Overview

SAP HANA System Replication

SAP HANA provides a feature called System Replication which is available in every SAP HANA installation offering an inherent disaster recovery support.

For details, please refer to SAP Help Portal HANA system replication.

HAE of SLES

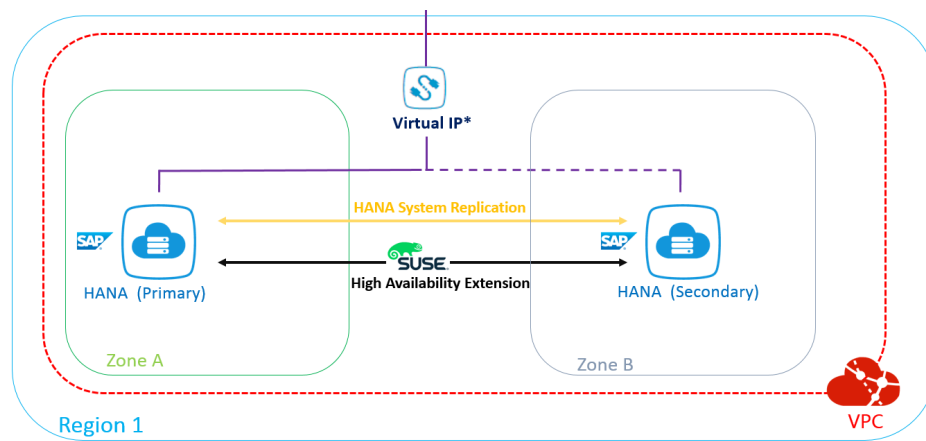
SUSE High Availability Extension (HAE) is a high availability solution based on Corosync and Pacemaker. With SUSE Linux Enterprise Server (SLES) for SAP, SUSE provides SAP specific Resource Agents (SAPHana, SAPHanaTopology etc.) used by Pacemaker to help users to buildup SAP HANA HA solution more effectively.

For details, please refer to SAP HANA SR Performance Optimized Scenario.

Architecture Overview

This document guides you on how to deploy a SAP HANA HA solution cross different Zones. Following is a brief architecture:

- HAE of SLES for SAP is used to setup the HA Cluster;
- SAP HANA System Replication is activated between the two HANA nodes;
- Two HANA nodes locates in different Zones of the same Region;
- Alibaba Cloud Specific Virtual IP Resource Agent is used to allow Moving IP automatically switched to Active SAP HANA node;Alibaba Cloud specific STONITH device is used for



fencing;

* Virtual IP – Alibaba Cloud provides a specific Resource Agent to switch Service IP across different ECS instances located in different zones;

Network Design

network	location	usage	subnet
business	eu-central-1 zone A	For Business	192.168.0.82/24
heartbeat	eu-central-1 zone B	For SR/HA	192.168.1.245/24

hostname	role	heartbeat IP	business IP	virtual IP
hana0	Hana primary node	192.168.0.83	192.168.0.82	192.168.4.1
hana1	Hana secondary node	192.168.1.246	192.168.1.245	192.168.4.1
HanaStudio	Hana Studio	no	192.168.0.79	no

Infrastructure Preparation

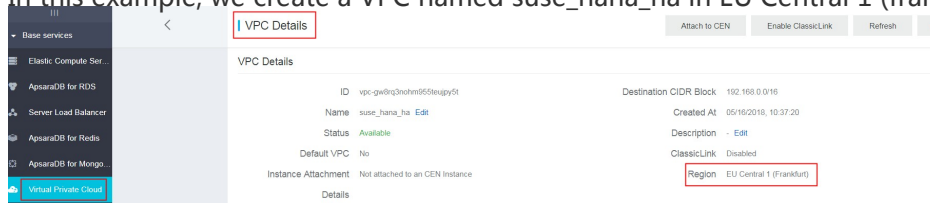
Infrastructure List

- 1 VPC network;
- 2 ECS instances in different zones of the same VPC;
- 2 Elastic Network Interface (ENI), one for each ECS instance;
- Alibaba Cloud specific Virtual IP Resource Agent and STONITH device;
- NAT Gateway and SNAT entry;

Creating VPC

First of all, a VPC should be created.

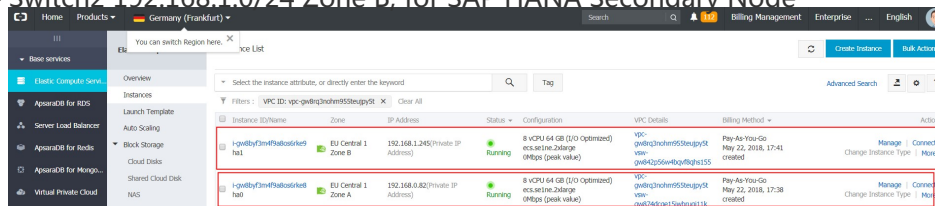
In this example, we create a VPC named `suse_hana_ha` in EU Central 1 (Frankfurt) Region as follow:



There should be at least 2

VSwitches(subnets) defined within the VPC network, each VSwitch bound to a different Zone. In this example, we have following 2 VSwitches(subnets):

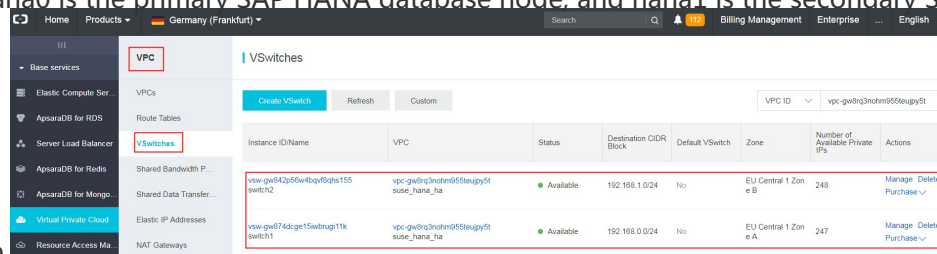
- Switch1 192.168.0.0/24 Zone A, for SAP HANA Primary Node
- Switch2 192.168.1.0/24 Zone B, for SAP HANA Secondary Node



Creating ECS Instances

Two ECS instances are created in different Zones of the same VPC. Choose the SLES for SAP image from the Image Market place.

In this example, 2 ECS instances (hostname: hana0 and hana1) are created in eu-central-1 Region, Zone A and Zone B, within VPC: `suse_hana_ha`, with SLES 12 SP2 for SAP image from the Image Market Place. Host hana0 is the primary SAP HANA database node, and hana1 is the secondary SAP



HANA database node.

Creating ENIs and binding to ECS instances

Create two ENIs, and attach one for each ECS instance, for HANA System Replication purpose.

Configure the IP addresses of the ENIs to the subnet for HANA System Replication only.

In this example, the ENIs are attached to ECS instances hana0 and hana1, and IP addresses are configured as 192.168.0.83 and 192.168.1.246 within the same VSwitches of hana0 and hana1, and put in the VPC: `suse_hana_ha`



Meanwhile, within the Guest OS, `/etc/hosts` should also be configured as well. In this example, please

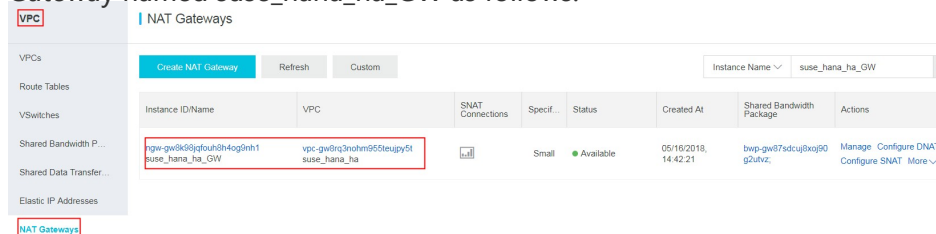
run following two commands on both sites:

```
echo "192.168.0.82 hana0 hana0" >> /etc/hosts
```

```
echo "192.168.1.245 hana1 hana1" >> /etc/hosts
```

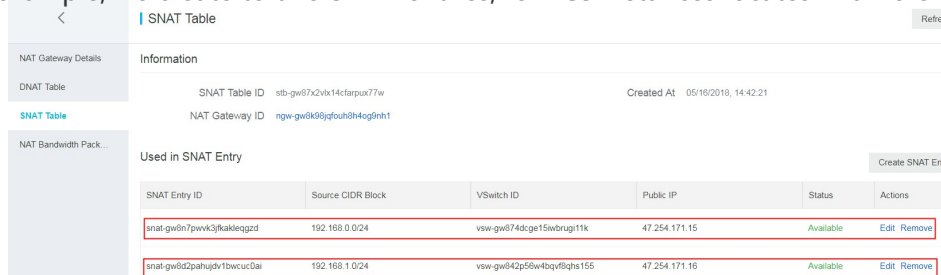
Creating NAT Gateway and configure SNAT entry

First of all, create a NAT Gateway attached to the given VPC; In our example, we create a NAT Gateway named `suse_hana_ha_GW` as follows:



After creating NAT Gateway, you need to create corresponding SNAT entry to allow ECS instances within the VPC can access public address on Internet. (Caution: Alibaba Cloud specific STONITH device and Virtual IP Resource Agent, need to access Alibaba Cloud OpenAPI through a public domain);

In our example, we create to two SNAT entries, for ECS instances locates in different network range as



follows:

Creating STONITH device and Virtual IP Resource Agent

Download software from with following command:

```
wget http://repository-iso.oss-cn-beijing.aliyuncs.com/ha/aliyun-ecs-pacemaker.tar.gz
```

```

hanal:/hana/tmp # ls
122.05 122.05.tgz SAPHOSTAGENT SAP HANA CLIENT aliyun corosync.conf
hanal:/hana/tmp # wget http://repository-iso.oss-cn-beijing.aliyuncs.com/ha/aliyun-ecs-pacemaker.tar.gz
--2018-03-05 16:12:13-- http://repository-iso.oss-cn-beijing.aliyuncs.com/ha/aliyun-ecs-pacemaker.tar.gz
Resolving repository-iso.oss-cn-beijing.aliyuncs.com (repository-iso.oss-cn-beijing.aliyuncs.com)... 59.110.190.40
Connecting to repository-iso.oss-cn-beijing.aliyuncs.com (repository-iso.oss-cn-beijing.aliyuncs.com) [59.110.190.40]:80
... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4125 (4.0K) [application/x-gzip]
Saving to: 'aliyun-ecs-pacemaker.tar.gz'

100%[=====] 4,125 --.-K/s in 0s

2018-03-05 16:12:13 (503 MB/s) - 'aliyun-ecs-pacemaker.tar.gz' saved [4125/4125]

hanal:/hana/tmp #

```

Extract the package and install the software

`tar -xvf aliyun-ecs-pacemaker.tar.gz`

```

hanal:/hana/tmp # ls -l
total 3102992
drwxrwxr-x 8 root root 4096 Mar 30 2017 122.05
-rw-r--r-- 1 root root 3177440026 Dec 24 18:14 122.05.tgz
drwxr-xr-x 2 root root 4096 Feb 28 12:14 SAPHOSTAGENT
drwxrwxrwx 4 root root 131 Feb 28 12:14 SAP HANA CLIENT
drwxr-xr-x 3 root root 69 Mar 5 15:31 aliyun
-rw-r--r-- 1 root root 4125 Mar 2 18:24 aliyun-ecs-pacemaker.tar.gz
-rw-r--r-- 1 root root 3957 Mar 1 10:29 corosync.conf
hanal:/hana/tmp # tar -xvf aliyun-ecs-pacemaker.tar.gz
aliyun-ecs-pacemaker/ecs-pacemaker/
aliyun-ecs-pacemaker/ecs-pacemaker/fence_aliyun.py
aliyun-ecs-pacemaker/ecs-pacemaker/vpc-move-ip
aliyun-ecs-pacemaker/install.sh
./install hanal:/hana/tmp # ./install

```

Install Alibaba Cloud OpenAPI SDK

`pip install aliyun-python-sdk-ecs aliyun-python-sdk-vpc aliyuncli`

```

hanal:~ # pip install aliyun-python-sdk-ecs aliyun-python-sdk-vpc aliyuncli
Collecting aliyun-python-sdk-ecs
  Downloading http://mirrors.aliyun.com/pypi/packages/02/f6/f63a3ba3357613caade16cf4fa69178eaaccd9cf2fcd6692368024499083/aliyun-python-sdk-ecs-4.6.4.tar.gz (41kB)
    100% |#####| 51kB 31.4MB/s
Collecting aliyun-python-sdk-vpc
  Downloading http://mirrors.aliyun.com/pypi/packages/39/eb/ec999243088e6d59710f9e634ca4b43c8ff895f256aa5e3a5ca9b2303982/aliyun-python-sdk-vpc-3.0.2.tar.gz
Collecting aliyuncli
  Downloading http://mirrors.aliyun.com/pypi/packages/f1/6c/ecb0313299cc8f886b75d4a117c6ead329ec89562fb1b82e1c6f083d3c5a/aliyuncli-2.1.9-py2-none-any.whl (69kB)
    100% |#####| 71kB 41.4MB/s
Collecting aliyun-python-sdk-core>=2.0.2 (from aliyun-python-sdk-ecs)
  Downloading http://mirrors.aliyun.com/pypi/packages/fa/28/045cf5cc1e80cc482c3bd1979287e62db73b3c42ef0e429f2d0d8dd2bfbe/aliyun-python-sdk-core-2.8.1.tar.gz
Collecting colorama<0.3.3,>=0.2.5 (from aliyuncli)
  Downloading http://mirrors.aliyun.com/pypi/packages/24/84/29ce4167d1f5c4a320aad91e1178e5a1baf9cfe1c63f6a275f5d1032303f313000/aliyun-python-sdk-core-2.8.1.tar.gz

```

Configure Alibaba Cloud OpenAPI SDK and Client

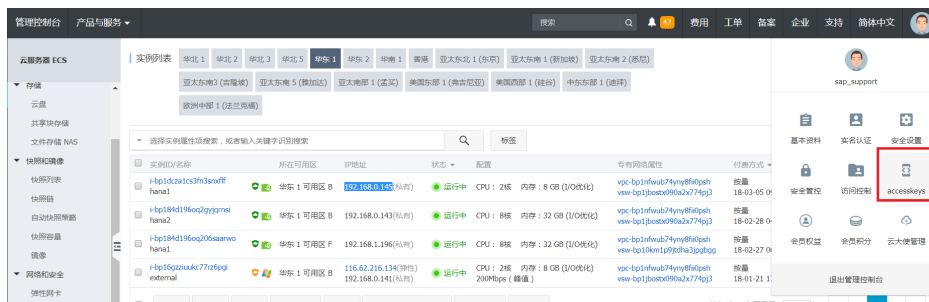
`aliyuncli configure`

```

hanal:~ #
hanal:~ # aliyuncli configure
Aliyun Access Key ID [None]: LTAIPYT3XsfmFfNs
Aliyun Access Key Secret [None]: wjM285NLWOS8dx32bewKwPIjTipHUf
Default Region Id [None]: cn-hangzhou
Default output format [None]: table
hanal:~ #

```

You can get your Access Key from following:



Software Preparation

Software List

- SLES 12 SP2 for SAP
- HANA Installation Media
- SAP Host Agent Installation Media

SLES for SAP HAE Installation

Both ECS instances are created with the SLES 12 SP2 for SAP image. Both ECS instances should install the HAE component, as well as package SAPHanaSR. In this example, we install HAE (major software component: Corosync and Pacemaker), and SAPHanaSR on both ECS instances as follows:

Install the pattern High Availability on both nodes. To do so, for example, use zypper :

```
zypper in -t pattern ha_sles
```

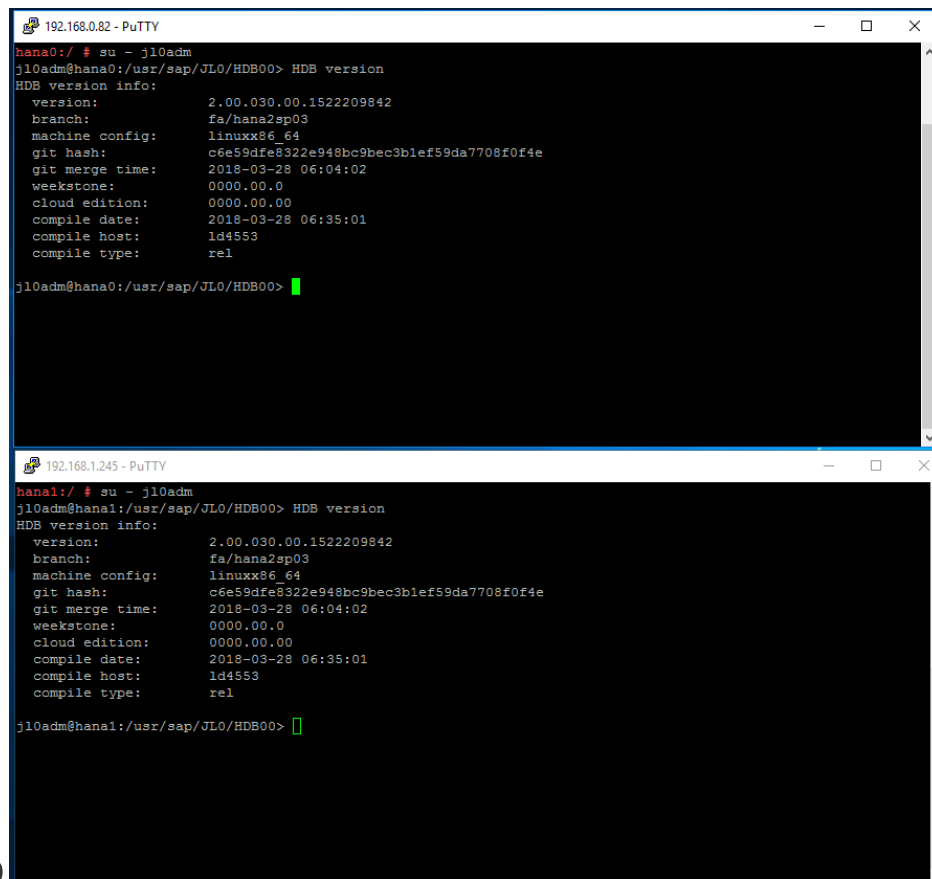
Now the Resource Agents for controlling the SAP HANA system replication needs to be installed at both cluster nodes.

```
zypper in SAPHanaSR SAPHanaSR-doc
```

SAP HANA Installation

Install SAP HANA software on both ECS instances, and make sure the SAP HANA SID and Instance Number are the same (requirement by SAP HANA System Replication). It is recommended to use hdbclm to do the installation. For details please refer to [SAP HANA Server Installation and Update Guide](#).

In this example, both node are installed with SAP HANA (Rev. 2.00.030.00), and SID: **JL0**, Instance



The image shows two terminal windows. The top window is titled '192.168.0.82 - PuTTY' and shows a user logging into 'hana0' as 'j10adm'. The user runs 'HDB version' and the output is as follows:

```

HDB version info:
version:          2.00.030.00.1522209842
branch:           fa/hana2sp03
machine config:   linuxx86_64
git hash:         c6e59dfe8322e948bc9bec3b1ef59da7708f0f4e
git merge time:   2018-03-28 06:04:02
weekstone:        0000.00.0
cloud edition:    0000.00.00
compile date:     2018-03-28 06:35:01
compile host:     ld4553
compile type:     rel

```

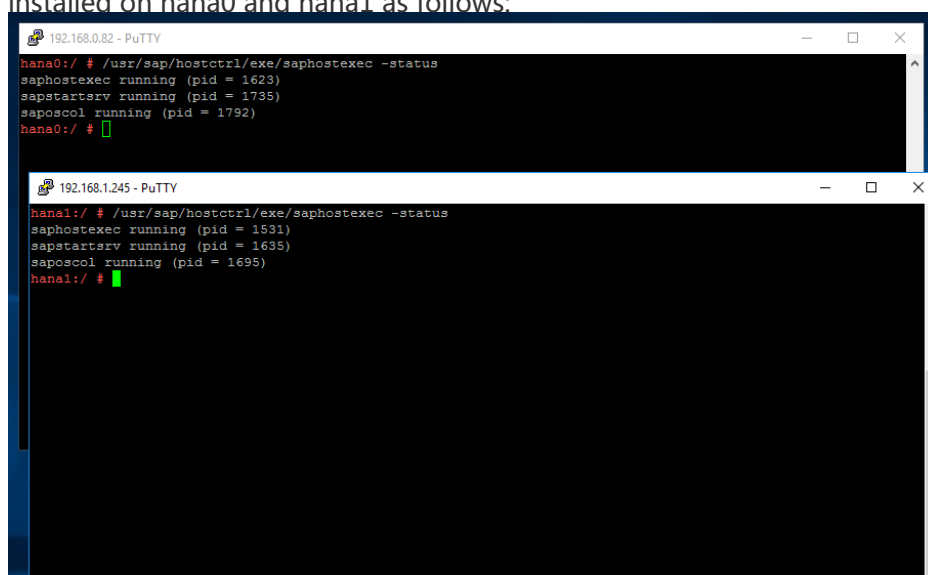
The bottom window is titled '192.168.1.245 - PuTTY' and shows a user logging into 'hana1' as 'j10adm'. The user runs 'HDB version' and the output is identical to the one in the top window.

Number: 00.

SAP Host Agent Installation

When you have finished hana installation with hdblcsm as mentioned above, the SAP Host Agent should already be installed on your server. In case you want to install it manually, please kindly refer to [Installing SAP Host Agent Manually](#).

In this example, you can check SAP Host Agent status after you have SAP HANA with hdblcsm installed on hana0 and hana1 as follows:



The image shows two terminal windows. The top window is titled '192.168.0.82 - PuTTY' and shows a user logging into 'hana0'. The user runs '/usr/sap/hostctrl/exe/saphostexec -status' and the output is as follows:

```

saphostexec running (pid = 1623)
sapstartsrv running (pid = 1735)
saposcol running (pid = 1792)

```

The bottom window is titled '192.168.1.245 - PuTTY' and shows a user logging into 'hana1'. The user runs '/usr/sap/hostctrl/exe/saphostexec -status' and the output is as follows:

```

saphostexec running (pid = 1631)
sapstartsrv running (pid = 1635)
saposcol running (pid = 1695)

```

Configuring SAP HANA System Replication

Backup HANA on both ECS instances

To do backup on HANA, you can either use SAP HANA studio or hdbsql as the client command tool.

The backup command is:

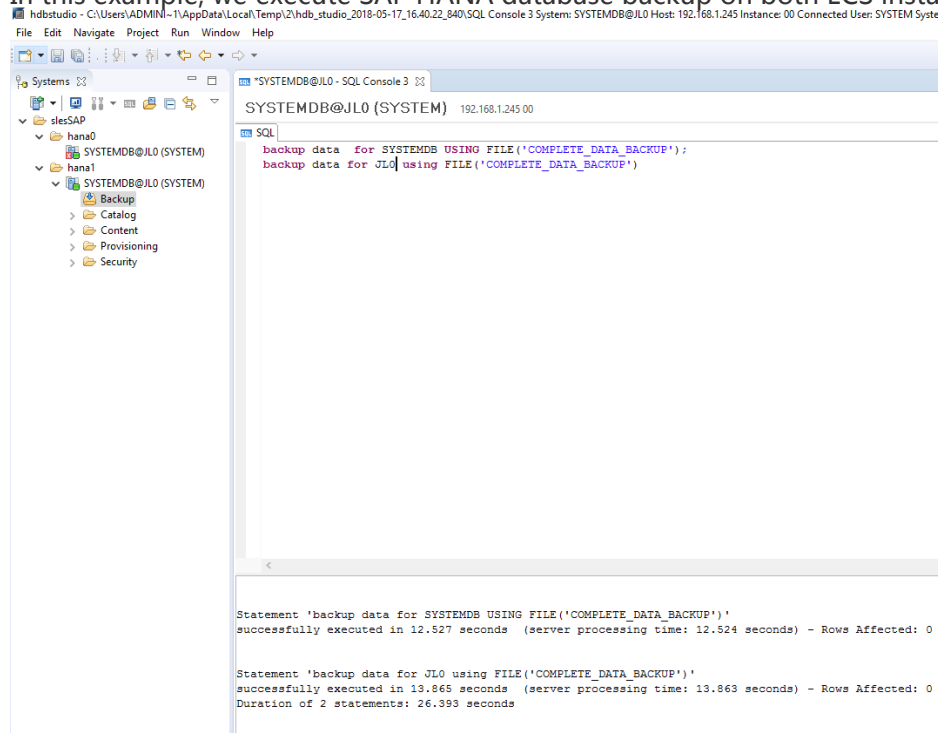
HANA1 single container:

BACKUP DATA USING FILE('COMPLETE_DATA_BACKUP');

HANA2 with multitenant as default mode (You should backup systemDB and also all tenantDB as shown below in our example):

BACKUP DATA for <DATABASE> using FILE('COMPLETE_DATA_BACKUP')

In this example, we execute SAP HANA database backup on both ECS instances as follows:



Configuring SAP HANA System Replication on primary node

a) Log onto the primary node with: su - <sid>adm;

[sidadm] should be replaced by your SAP HANA database SID. In our example it is su - j10adm;

b) Stop HANA with: HDB stop;

c) Change following file content as user root:

/hana/shared/SID/global/hdb/custom/config/global.ini

Add following content:

```

[system_replication_hostname_resolution]
<IP> = <HOSTNAME>
  
```

[IP] should be address of the ENI (heartbeat IP address for HANA system replication) attached to the Secondary node;

[HOSTNAME] should be hostname of the Secondary node;

In this example, we have following configuration:

[system_replication_hostname_resolution]

192.168.1.246 = hana1

Configuring SAP HANA System Replication on secondary node

Same as above for primary, but use IP and hostname of primary node

In this example, we have following configuration:

[system_replication_hostname_resolution]

192.168.0.83 = hana0

Enable SAP HANA System Replication on primary node

a) Log onto the primary node with: su - <sid>adm;

b) Start HANA with: HDB start;

c) Enable System Replication with:

hdbnsutil -sr_enable --name= [primary location name]

[primary location name] should be replaced by location of your primary HANA node.

In this example, we use following command:

hdbnsutil -sr_enable --name=hana0

CAUTION: all above operations are done on primary node.

Register the Secondary node to the Primary HANA node

a) Log onto the secondary node with: su - <sid>adm;

b) Stop HANA with: HDB stop;

c) Register the Secondary HANA node to the Primary HANA node by running following command:

hdbnsutil -sr_register --remoteHost=[location of primary Node] --remoteInstance=[instance number of primary node] --replicationMode=sync --name=[location of the secondary node] --operationMode=logreplay

In this example, we use following command:

hdbnsutil -sr_register --name=hana1 --remoteHost=hana0 --remoteInstance=00 --replicationMode=sync --operationMode=logreplay

d) Start HANA with: HDB start;

e) Verify the System Replication Status with:

hdbnsutil -sr_state

In this example, we have following status on secondary HANA node hana1:

```

jl0adm@hana1:/usr/sap/JL0/HDB00> hdbnsutil -sr_state

System Replication State
~~~~~

online: true

mode: sync
operation mode: logreplay
site id: 2
site name: hana1


is source system: false
is secondary/consumer system: true
has secondaries/consumers attached: false
is a takeover active: false
active primary site: 1

primary masters: hana0

Host Mappings:
~~~~~

hana1 -> [hana1] hana1
hana1 -> [hana0] hana0


Site Mappings:
~~~~~
hana0 (primary/primary)
    |--hana1 (sync/logreplay)

Tier of hana0: 1
Tier of hana1: 2


Replication mode of hana0: primary
Replication mode of hana1: sync

Operation mode of hana0: primary
Operation mode of hana1: logreplay

Mapping: hana0 -> hana1
done.

```

CAUTION: all above operations are done on secondary node.

Configuring HAE for SAP HANA

STONITH: fence_aliyun

For a HA solution, a fencing device is a must. Alibaba Cloud provides its own STONITH device, which

allows the servers in the HA cluster to shut down the other which is not responsible. The STONITH device leverage Alibaba Cloud OpenAPI underneath the ECS instance, which is similar to a physical reset / shutdown on a on-premise environment.

Configuration of Corosync

It is desirable that, you add more redundancy for messaging (Heartbeat) by using separate ENIs attached to the ECS instances with separate network range. On Alibaba Cloud, it is strongly suggested that, only using Unicast for the transport setting in Corosync. Follow the following steps to configure Corosync:

Create Keys

Run `corosync-keygen` on primary HANA node. The generated key will be located in the file: `/etc/corosync/authkey`.

In our example, we execute the command on hana1:



```
192.168.0.82 - PuTTY
hana0:/etc/corosync # ls
authkey  corosync.conf  corosync.conf.final  uidgid.d
hana0:/etc/corosync #
```

Configure `/etc/corosync/corosync.conf` with following content as root on primary HANA node:

```
totem {
version: 2
token: 5000
token_retransmits_before_loss_const: 6
secauth: on
crypto_hash: sha1
crypto_cipher: aes256
clear_node_high_bit: yes
interface {
ringnumber: 0
bindnetaddr: **IP-address-for-heart-beating-for-the-current-server**
mcastport: 5405
ttl: 1
}
# On Alibaba Cloud, transport should be set to udpu, means: unicast
transport: udpu
```

```

}
logging {
  fileline: off
  to_logfile: yes
  to_syslog: yes
  logfile: /var/log/cluster/corosync.log
  debug: off
  timestamp: on
  logger_subsys {
    subsys: QUORUM
    debug: off
  }
}
nodelist {
  node {
    ring0_addr: **ip-node-1**
    nodeid: 1
  }
  node {
    ring0_addr: **ip-node-2**
    nodeid: 2
  }
}
quorum {
  # Enable and configure quorum subsystem (default: off)
  # see also corosync.conf.5 and votequorum.5
  provider: corosync_votequorum
  expected_votes: 2
  two_node: 1
}

```

IP-address-for-heart-beating-for-the-current-server should be replaced by the IP address of the current server, used for messaging (heartbeat) or HANA System Replication. In our example, we use IP address of ENI of the current node (192.168.0.83 for hana0 and 192.168.1.246 for hana1); **Caution:** this value will be different on primary and secondary node. `nodelist` directive is used to list all nodes in the cluster.

ip-node-1 and **ip-node-2** should be replaced by the IP addresses of the ENIs attached to ECS instances for Heartbeat Purpose or HANA System Replication Purpose (in this example it should be 192.168.0.83 for hana0 and 192.168.1.246 for hana1).

After completing edit of `/etc/corosync/corosync.conf` on primary HANA node, copy the `/etc/corosync/authkey` and `/etc/corosync/corosync.conf` to `/etc/corosync` on the secondary HANA node with following command:

```

scp /etc/corosync/authkey root@hostnameOfSecondaryNode:/etc/corosync
scp /etc/corosync/corosync.conf root@hostnameOfSecondaryNode:/etc/corosync

```

In our example, we execute following command:

```

hana1:/ # scp /etc/corosync/authkey root@hana2:/etc/corosync

```

```

hana1:/ # scp /etc/corosync/corosync.conf root@hana2:/etc/corosync

```

After copy the `corosync.conf` to the secondary node, please kindly configure the `bindnetaddr` as above to the local heart beating IP address.

Configuration of pacemaker

For SAP HANA HA solution, we need to configure 7 Resource Agents and corresponding constraints in Pacemaker.

CAUTION the following pacemaker configuration only need to be done on one node (normally primary node).

1. Cluster bootstrap and more

Add configuration of bootstrap and default setting of resource and operations to the cluster; Save following scripts in a file: crm-bs.txt

```
property $id='cib-bootstrap-options' \
stonith-enabled="true" \
stonith-action="off" \
stonith-timeout="150s"
rsc_defaults $id="rsc-options" \
resource-stickiness="1000" \
migration-threshold="5000"
op_defaults $id="op-options" \
timeout="600"
```

Execute command to add setting to the cluster:

crm configure load update crm-bs.txt

2. STONITH device

This part defines Aliyun STONITH devices in the cluster;

Save following scripts in a file: crm-stonith.txt

```
primitive res_ALIYUN_STONITH_1 stonith:fence_aliyun \
op monitor interval=120 timeout=60 \
params pcmk_host_list=<primary node hostname> port=<primary node instance id> \
access_key=<access key> secret_key=<secret key> \
region=<region> \
meta target-role=Started
primitive res_ALIYUN_STONITH_2 stonith:fence_aliyun \
op monitor interval=120 timeout=60 \
params pcmk_host_list=<secondary node hostname> port=<secondary node instance id> \
access_key=<access key> secret_key=<secret key> \
region=<region> \
meta target-role=Started
location loc_<primary node hostname>_stonith_not_on_<primary node hostname>
res_ALIYUN_STONITH_1 -inf: <primary node hostname>
#Stonith 1 should not run on primary node because it is controlling primary node
location loc_<secondary node hostname>_stonith_not_on_<secondary node hostname>
res_ALIYUN_STONITH_2 -inf: <secondary node hostname>
#Stonith 2 should not run on secondary node because it is controlling secondary node
```

[secondary node hostname] / **[primary node hostname]** should be replaced by the real hostname of your secondary node;

[secondary node instance id] / **[secondary node instance id]** should be replaced by the real instance-id of your secondary node; you can get this from the console;

[access key] should be replaced with real access key;

[secret key] should be replaced with real secret key;

[region] should be replaced with real region name where the node locates;

Execute command to add the resource to the cluster:

crm configure load update crm-stonith.txt

3. SAPHanaTopology

This part defines a SAPHanaTopology RA, and a clone of SAPHanaTopology on both nodes in the cluster. Save following scripts in a file: crm-saphanatop.txt

```
primitive rsc_SAPHanaTopology_<SID>_HDB<instance number> ocf:suse:SAPHanaTopology \
operations $id="rsc_SAPHanaTopology_<SID>_HDB<instance number>-operations" \
op monitor interval="10" timeout="600" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="300" \
params SID="<SID>" InstanceNumber="<instance number>"
clone cln_SAPHanaTopology_<SID>_HDB<instance number>
rsc_SAPHanaTopology_<SID>_HDB<instance number> \
meta clone-node-max="1" interleave="true"
```

[SID] should be replaced by the real SAP HANA SID;

[instance number] should be replaced by the real SAP HANA Instance Number;

Execute command to add resources to the cluster:

crm configure load update crm-saphanatop.txt

4. SAPHana

This part defines a SAPHana RA, and a Multi-state resource of SAPHana on both nodes in the cluster. Save following scripts in a file: crm-saphana.txt

```
primitive rsc_SAPHana_<SID>_HDB<instance number> ocf:suse:SAPHana \
operatoins $id="rsc_sap_<SID>_HDB<instance number>-operations" \
op start interval="0" timeout="3600" \
op stop interval="0" timeout="3600" \
op promote interval="0" timeout="3600" \
op monitor interval="60" role="Master" timeout="700" \
op monitor interval="61" role="Slave" timeout="700" \
params SID="<SID>" InstanceNumber="<instance number>" PREFER_SITE_TAKEOVER="true" \
DUPLICATE_PRIMARY_TIMEOUT="7200" AUTOMATED_REGISTER="false"
ms msl_SAPHana_<SID>_HDB<instance number> rsc_SAPHana_<SID>_HDB<instance number> \
meta clone-max="2" clone-node-max="1" interleave="true"
```

[SID] should be replaced by the real SAP HANA SID;

[instance number] should be replaced by the real SAP HANA Instance Number;

Execute command to add resources to the cluster:

crm configure load update crm-saphana.txt

5. Virtual IP

This part defines a Virtual IP RA in the cluster. Save following scripts in a file: crm-vip.txt.

```
primitive res_vip_<SID>_HDB<instance number> ocf:aliyun:vpc-move-ip \
op monitor interval=60 \
meta target-role=Started \
params address=<virtual_IPv4_address> routing_table=<route_table_ID> interface=eth0
```

[virtual_IP4_address] should be replaced by the real IP address you prefer to provide service;

[route_table_ID] should be replaced by the route table ID of your VPC;

[SID] should be replaced by the real SAP HANA SID;

[instance number] should be replaced by the real SAP HANA Instance Number;

Execute command to add the resource to the cluster:

crm configure load update crm-vip.txt

6. Constraints

Two constraints are organizing the correct placement of the virtual IP address for the client database access and the start order between the two resource agents SAPHana and SAPHanaTopology. Save following scripts in a file: crm-constraint.txt

```
colocation col_SAPHana_vip_<SID>_HDB<instance number> 2000: rsc_vip_<SID>_HDB<instance number>:started \
msl_SAPHana_<SID>_HDB<instance number>:Master
order ord_SAPHana_<SID>_HDB<instance number> Optional:
cln_SAPHanaTopology_<SID>_HDB<instance number> \
msl_SAPHana_<SID>_HDB<instance number>
```

[SID] should be replaced by the real SAP HANA SID;

[instance number] should be replaced by the real SAP HANA Instance Number;

Execute command to add the resource to the cluster:

crm configure load update crm-constraint.txt

7. check cluster status

a) Start HANA HA Cluster on both nodes

Execute command: systemctl start pacemaker

b) Monitor the HANA HA Cluster

Execute command: systemctl status pacemaker

Execute command: crm_mon -r

In our example we have following result:

```

192.168.0.82 - PuTTY
hana0:/ # systemctl status pacemaker
● pacemaker.service - Pacemaker High Availability Cluster Manager
   Loaded: loaded (/usr/lib/systemd/system/pacemaker.service; disabled; vendor preset: disabled)
   Active: active (running) since Wed 2018-06-20 14:04:48 UTC; 58min ago
     Docs: man:pacemakerd
           http://clusterlabs.org/doc/en-US/Pacemaker/1.1-pcs/html/Pacemaker_Explained/index.html
   Main PID: 32044 (pacemakerd)
     Tasks: 7 (limit: 512)
    CGroup: /system.slice/pacemaker.service
            └─32044 /usr/sbin/pacemakerd -f
              └─32047 /usr/lib64/pacemaker/cib
                └─32048 /usr/lib64/pacemaker/stonithd
                  └─32049 /usr/lib64/pacemaker/lrmd
                    └─32050 /usr/lib64/pacemaker/attrd
                      └─32051 /usr/lib64/pacemaker/pengine
                        └─32052 /usr/lib64/pacemaker/crmd
^C
hana0:/ # crm status
Stack: corosync
Current DC: hana0 (version 1.1.15-21.1-e174ec8) - partition with quorum
Last updated: Wed Jun 20 15:03:39 2018
Last change: Wed Jun 20 15:03:36 2018 by root via crm_attribute on hana0

2 nodes configured
7 resources configured

Online: [ hana0 hana1 ]

Full list of resources:

res_ALIYUN_STONITH_0 (stonith:fence_aliyun): Started hana1
res_ALIYUN_STONITH_1 (stonith:fence_aliyun): Started hana0
rsc_vip_JLO_HDB00 (ocf::aliyun:vpc-move-ip): Started hana0
Master/Slave Set: ms1_SAPHana_JLO_HDB00 [rsc_SAPHana_JLO_HDB00]
   Masters: [ hana0 ]
   Slaves: [ hana1 ]
Clone Set: cln_SAPHanaTopology_JLO_HDB00 [rsc_SAPHanaTopology_JLO_HDB00]
   Started: [ hana0 hana1 ]

hana0:/ #

```

Meanwhile,

please kindly check, if a new entry [virtual_IP4_address] is added into the route table of VPC.

In our example, we have following:

Route Table Details

Route Table ID: vtb-gw8f1g1d8cp14tzyub VPC ID: vpc-gw8rq3nohmd655teagpy5t

Name: - Edit Route Table Type: System

Created At: 05/16/2018, 10:37:20 Description: - Edit

Route Entry List

Destination CIDR Block	Status	Next Hop	Type	Actions
192.168.4.1/32	Available	Instance ID: i-gw8by3m4f0a8cs6ike8 Instance Type: ECS Instance	Custom	Delete
0.0.0.0/0	Available	Instance ID: ngw-gw8k8g9glouh2h4cog9nh1 Instance Type: NAT Gateway	Custom	Delete
192.168.0.0/24	Available	-	System	
192.168.1.0/24	Available	-	System	

Verify the HA take over

- Shutdown the primary node;

Check the status of Pacemaker as follows:

```

192.168.1.245 - PuTTY
hanal:~ # crm status
Stack: corosync
Current DC: hanal (version 1.1.15-21.1-e174ec8) - partition with quorum
Last updated: Wed Jun 20 15:19:59 2018
Last change: Wed Jun 20 15:19:44 2018 by root via crm_attribute on hanal

2 nodes configured
7 resources configured

Online: [ hanal ]
OFFLINE: [ hana0 ]

Full list of resources:

res_ALIYUN_STONITH_0 (stonith:fence_aliyun): Started hanal
res_ALIYUN_STONITH_1 (stonith:fence_aliyun): Stopped
rsc_vip_JL0_HDB00 (ocf::aliyun:vpc-move-ip): Started hanal
Master/Slave Set: msl_SAPHana_JL0_HDB00 [rsc_SAPHana_JL0_HDB00]
Masters: [ hanal ]
Stopped: [ hana0 ]
Clone Set: cln_SAPHanaTopology_JL0_HDB00 [rsc_SAPHanaTopology_JL0_HDB00]
Started: [ hanal ]
Stopped: [ hana0 ]

hanal:~ #

```

Compare the entry of route table in VPC as follows:

Destination CIDR Block	Status	Next Hop	Type	Actions
192.168.4.1/32	Available	Instance ID i-gw8byf3m4f6a8os6rke9 Instance Type ECS Instance	Custom	Delete
0.0.0.0/0	Available	Instance ID ngw-gw8g9d9u8h4og2rh1 Instance Type NAT Gateway	Custom	Delete
192.168.0.0/24	Available	-	System	
192.168.1.0/24	Available	-	System	

Example

Example Cluster Configuration

In our example, the cluster configuration(you can check it via command "crm configure show") should have content as below:

```

node 1: hana0 \
attributes hana_jl0_vhost=hana0 hana_jl0_srmode=sync hana_jl0_remoteHost=hana1 hana_jl0_site=hana0
lpa_jl0_lpt=10 hana_jl0_op_mode=logreplay
node 2: hana1 \
attributes lpa_jl0_lpt=1529509236 hana_jl0_op_mode=logreplay hana_jl0_vhost=hana1 hana_jl0_site=hana1
hana_jl0_srmode=sync hana_jl0_remoteHost=hana0
primitive res_ALIYUN_STONITH_0 stonith:fence_aliyun \

```

```

op monitor interval=120 timeout=60 \
params pcmk_host_list=hana0 port=i-gw8byf3m4f9a8os6rke8 access_key=<access key> secret_key=<secret key>
region=eu-central-1 \
meta target-role=Started
primitive res_ALIYUN_STONITH_1 stonith:fence_aliyun \
op monitor interval=120 timeout=60 \
params pcmk_host_list=hana1 port=i-gw8byf3m4f9a8os6rke9 access_key=<access key> secret_key=<secret key>
region=eu-central-1 \
meta target-role=Started
primitive rsc_SAPHanaTopology_JL0_HDB00 ocf:suse:SAPHanaTopology \
operations $id=rsc_SAPHanaTopology_JL0_HDB00-operations \
op monitor interval=10 timeout=600 \
op start interval=0 timeout=600 \
op stop interval=0 timeout=300 \
params SID=JL0 InstanceNumber=00
primitive rsc_SAPHana_JL0_HDB00 ocf:suse:SAPHana \
operations $id=rsc_SAPHana_JL0_HDB00-operations \
op start interval=0 timeout=3600 \
op stop interval=0 timeout=3600 \
op promote interval=0 timeout=3600 \
op monitor interval=60 role=Master timeout=700 \
op monitor interval=61 role=Slave timeout=700 \
params SID=JL0 InstanceNumber=00 PREFER_SITE_TAKEOVER=true DUPLICATE_PRIMARY_TIMEOUT=7200
AUTOMATED_REGISTER=false
primitive rsc_vip_JL0_HDB00 ocf:aliyun:vpc-move-ip \
op monitor interval=60 \
meta target-role=Started \
params address=192.168.4.1 routing_table=vtb-gw8fii1g1d8cp14tzynub interface=eth0
ms msl_SAPHana_JL0_HDB00 rsc_SAPHana_JL0_HDB00 \
meta clone-max=2 clone-node-max=1 interleave=true target-role=Started
clone cln_SAPHanaTopology_JL0_HDB00 rsc_SAPHanaTopology_JL0_HDB00 \
meta clone-node-max=1 interleave=true
colocation col_SAPHana_vip_JL0_HDB00 2000: rsc_vip_JL0_HDB00:Started msl_SAPHana_JL0_HDB00:Master
location loc_hana0_stonith_not_on_hana0 res_ALIYUN_STONITH_0 -inf: hana0
location loc_hana1_stonith_not_on_hana1 res_ALIYUN_STONITH_1 -inf: hana1
order ord_SAPHana_JL0_HDB00 Optional: cln_SAPHanaTopology_JL0_HDB00 msl_SAPHana_JL0_HDB00
property cib-bootstrap-options: \
have-watchdog=false \
dc-version=1.1.15-21.1-e174ec8 \
cluster-infrastructure=corosync \
stonith-action=off \
stonith-enabled=true \
stonith-timeout=150s \
last-lrm-refresh=1529503606 \
maintenance-mode=false
rsc_defaults rsc-options: \
resource-stickiness=1000 \
migration-threshold=5000
op_defaults op-options: \
timeout=600

```

Example for /etc/corosync/corosync.conf

In our example, the corosync.conf should on hana1 should have content as below:

```
totem{
version: 2
token: 5000
token_retransmits_before_loss_const: 6
secauth: on
crypto_hash: sha1
crypto_cipher: aes256
clear_node_high_bit: yes
interface {
ringnumber: 0
bindnetaddr: 192.168.0.83
mcastport: 5405
ttl: 1
}
# On Alibaba Cloud, transport should be set to udpu, means: unicast
transport: udpu
}
logging {
fileline: off
to_logfile: yes
to_syslog: yes
logfile: /var/log/cluster/corosync.log
debug: off
timestamp: on
logger_subsys {
subsys: QUORUM
debug: off
}
}
nodelist {
node {
ring0_addr: 192.168.0.83
nodeid: 1
}
node {
ring0_addr: 192.168.1.246
nodeid: 2
}
}
quorum {
# Enable and configure quorum subsystem (default: off)
# see also corosync.conf.5 and votequorum.5
provider: corosync_votequorum
expected_votes: 2
two_node: 1
}
```

Reference

- Pacemaker 1.1 Configuration Explained
- SAP HANA SR Performance Optimized Scenario
- SAP HANA system replication - SAP Help Portal

Microsoft SQL Server on Alibaba Cloud

Microsoft SQL Server on Alibaba Cloud

- Getting Started
- Prerequisites
 - Alibaba Cloud account and RAM
 - SQL Server Images and Version
 - ECS instance
 - VPC
 - Cloud Disk
 - OSS
 - Shared block storage
- Installation
 - Create SQL Server instance
 - Windows Settings
- High Availability
- Backup
- Performance
 - Separate Cloud Disk
 - Table Compression
 - Enable Buffer Pool Extension(BPE)
 - Max degree of parallelism setting
- Monitor
 - Monitoring Capability
 - Alarm Capability
- Management Studio
 - Remote SQL Server Management Studio
 - Default SQL Server Management Studio

Getting Started

Make use of Alibaba Cloud reliable and flexible cloud computing infrastructure and platform services can help you to run Microsoft SQL Server more stably and smoothly. Microsoft SQL server on Alibaba cloud Elastic Compute Service(ECS) just like when you installed on-premises, you are responsible for installation, administering the database, including backups and recovery, patching the operating system and the database, tuning of the operating system and database parameters, managing

security, and configuring high availability or replication. It gives you complete control over every setting. To learn what is ECS, please refer to the [detail](#). This article provides you with the best practice for Microsoft SQL Server instance running on ECS instance. Please be aware this article used **SQL server 2016 Enterprise Edition** as the example, so not every option is appropriate for every version. It also presents a simplistic procedure to guide you for running your Microsoft SQL server. If you want to choose Alibaba RDS for SQL server, please refer to [Quick Start for SQL Server](#).

Prerequisites

Alibaba Cloud account and RAM

You must have registered to an Alibaba Cloud account. We recommend you to enable RAM to manage your account. Resource Access Management (RAM) is a cloud service that helps you manage user identities and control resources access. Using RAM, you can create and manage user accounts, and control the operation permissions that these user accounts possess for resources under your account, for example, employees, systems, and applications. For the detail information please refer to [Quick Start](#) and [RAM best practices](#).

SQL Server Images and Version

Alibaba cloud support Bring Your Own License (BYOL) and Images. SQL Server Enterprise, Standard, and Express Editions are licensed for production use. For Enterprise and Standard Editions, contact your software vendor for the installation media. You can find purchasing information and a directory of Microsoft partners on Microsoft official purchasing website. Free editions you can find at Microsoft official website: [SQL Server Downloads](#).

ECS instance

Elastic Compute Service (ECS) is a type of computing service that features elastic processing capabilities. ECS has a simpler and more efficient management mode than physical servers. You can create instances, change the operating system, and add or release any number of ECS instances at any time to fit your business needs. An ECS instance is a virtual computing environment that includes CPU, memory, and other basic computing components. An instance is the core component of ECS and is the actual operating entity offered by Alibaba Cloud. Other resources, such as disks, images, and snapshots, can only be used in conjunction with an ECS instance. Before create SQL server instances you have to create ECS instances first using the [ECS console](#), about the detail information, please refer to [Create ECS instances](#).

VPC

Virtual Private Cloud (VPC) creates an isolated network environment for you SQL Server environment. You can select an IP address range, divide networks, and configure the routing list and gateway. The interflow of VPC intranet and between VPC and physical IDC machine rooms can be realized among

regions or users. About how to create VPC please refer to the detail.

Cloud Disk

Ultra Cloud Disk: When you create ECS instance, Ultra Cloud Disk as the system disk provides a high-performance location for operating system and windows page file.

SSD Cloud Disk: When you create ECS instance we recommend you choose SSD cloud disk store the database files, tempdb, log file separately. Separate SSD cloud disks provide high performance and high reliability.

- High performance: A single SSD cloud disk provides a maximum of 20,000 random reading/writing IOPS and 300 MBps throughput of storage performance.
- $IOPS = \min\{1200 + 30 * disk_size, 20000\}$. The base is 1200 IOPS, and each GB provides 30 random IOPS up to a maximum of 20,000.
- $Throughput = \min\{80 + 0.5 * disk_size, 300\}$ MBps. The base is 80 MBps, and each GB adds an additional 0.5 MBps up to a maximum of 300 MBps throughput performance.
- Reliability: SSD cloud disks use Alibaba Cloud's Apsara distributed storage technology, based on three distributed copies, which can guarantee 99.9999999% data reliability.

For how to create a cloud disk, please refer to [create a cloud disk](#).

OSS

Alibaba Cloud Object Storage Service (OSS) is a network-based data access service. OSS enables you to store and retrieve unstructured data including text files, images, audios, and videos. We recommend you backup your SQL Server database into OSS. For how to use OSS please see [Get started with Object Storage Service](#)

Shared block storage

Shared Block Storage is designed for the high availability architecture of enterprise-class applications and provide shared access to block storage devices in a Share-everything architecture, such as the SQL Server always on with WSFC node architecture, which is common among government departments, enterprises, and financial customers, and the high availability server cluster architecture. For about shared block storage detail, please see [Shared block storage FAQ](#)

Installation

This section provides general information about how to create a SQL server instance on Elastic Compute Service (ECS). The tutorial includes the following tasks :

Create SQL Server instance

We recommend you to close the windows update setting before you create your SQL Server instance as below:

1. Click "run" button and input gpedit.msc
2. You will open the "Local Computer Policy" :

- Select "Administrative template"
- Select " Windows Components"
- Select " Windows Update"

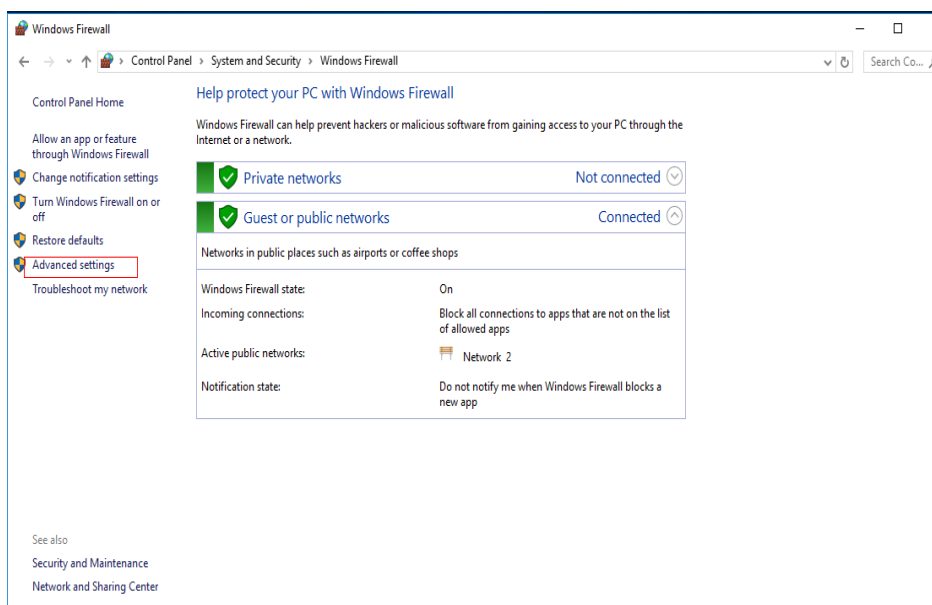
3. Change the "Configure Automatic Updates" option to disable status.

After the SQL server instance creating you can decide whether need to enable the update setting. It is the same method with on premise to create your SQL Server instance on ECS instance. You can choose installation wizard, command-line or using a configuration file. For how to create or install please refer to Microsoft website.

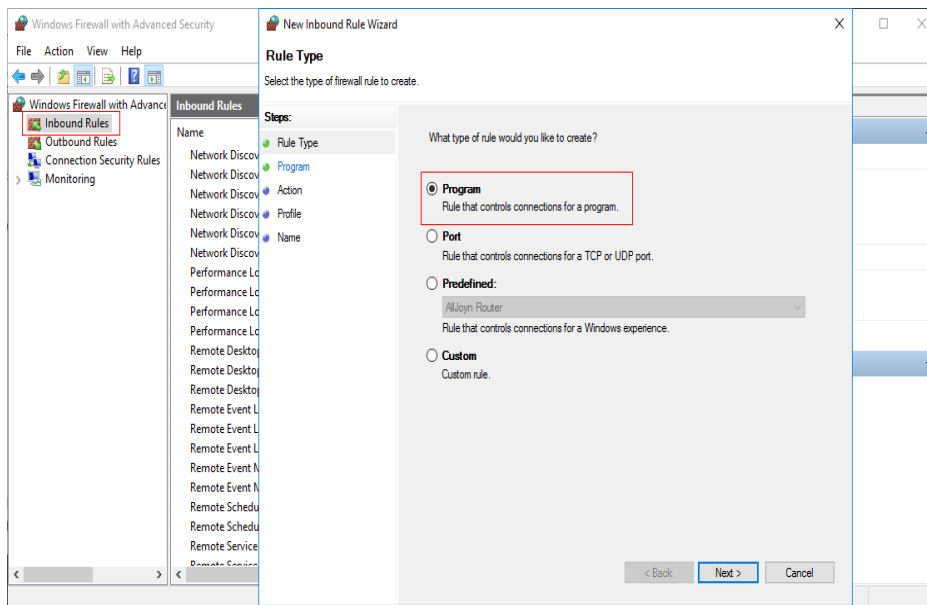
Windows Settings

This section provides you the settings about how to configure windows settings to better optimize your SQL Server instance. We recommend you to setting the windows server firewall rule to specify the IP addresses for your client computer. It is very important security policy when you create your SQL Server instance to be able to connect to the database from other client machines. Configure the firewall to allow incoming traffic:

- Open your windows firewall with advanced security.

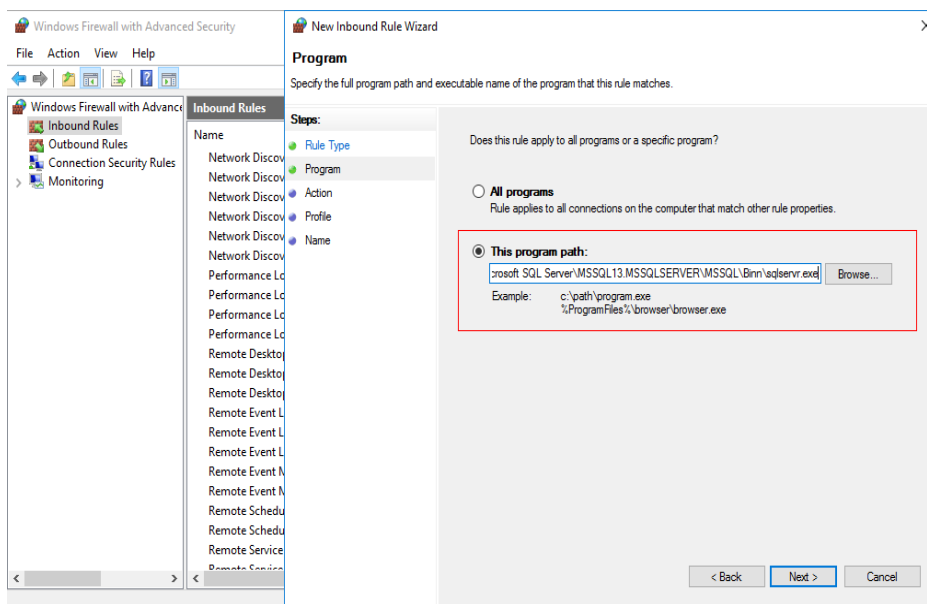


- Click right button to create a new inbound rule.

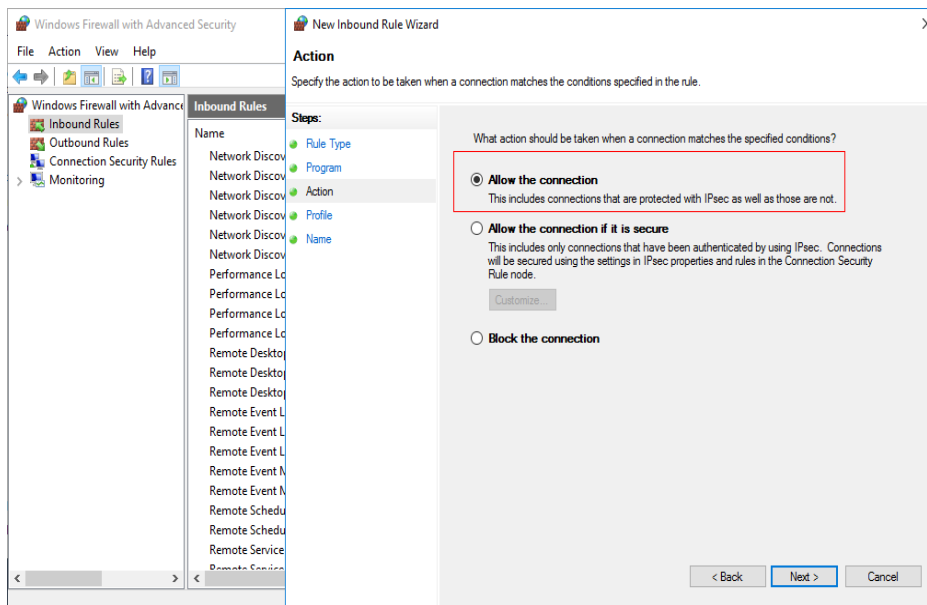


- Select your program path.

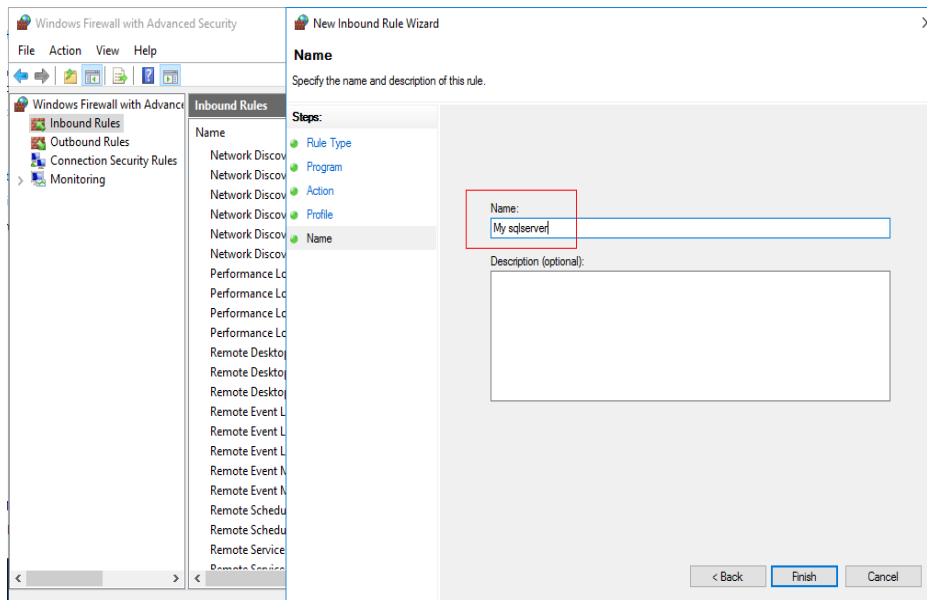
Input: %ProgramFiles%\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\Binn\sqlservr.exe



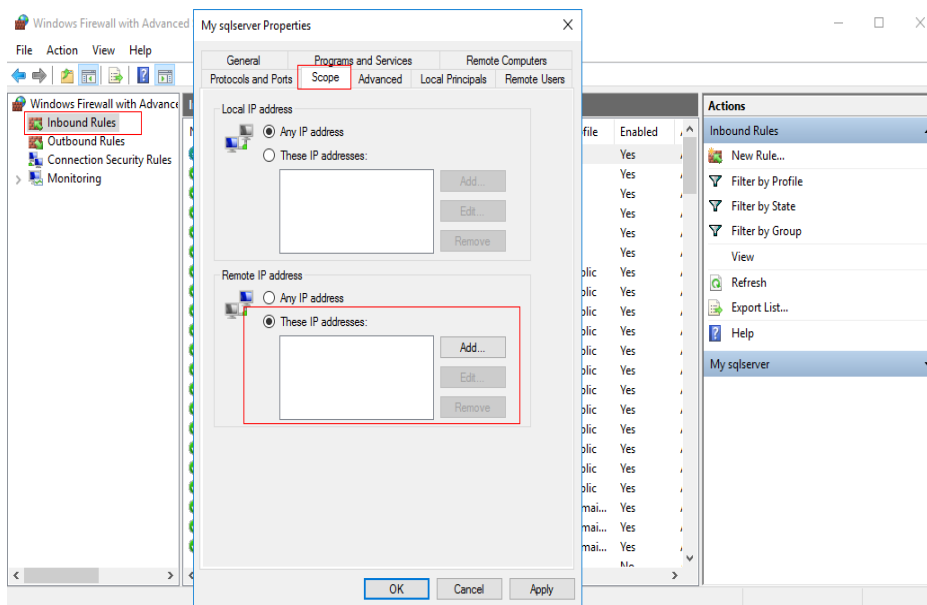
- Allow the connection.



- Name your rule, for example: mysqlserver.



- Set your remote policy.



The system default network settings are usually sufficient. Alibaba cloud offers you the high capacity and performance network. When you create ECS instances you can choose the network bandwidth from 1M up to 100M, about the bandwidth, Please refer to ECS Bandwidth FAQs. Windows requires anti-virus software to be installed. Install enterprise level anti-virus software and enable virus library updating and real-time protection, however, if the antivirus software is not configured correctly, it can negatively impact your database performance. Microsoft provides advice about how to choose antivirus software.

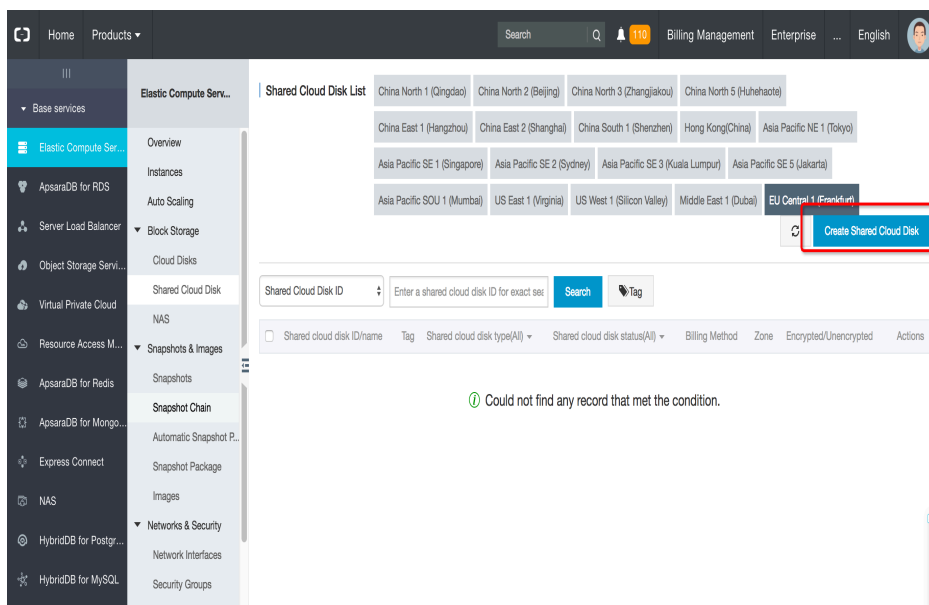
High Availability

We recommend you to use Windows Server Failover Clustering and SQL Server AlwaysOn Availability Groups as your SQL Server high availability solution on ECS instances.

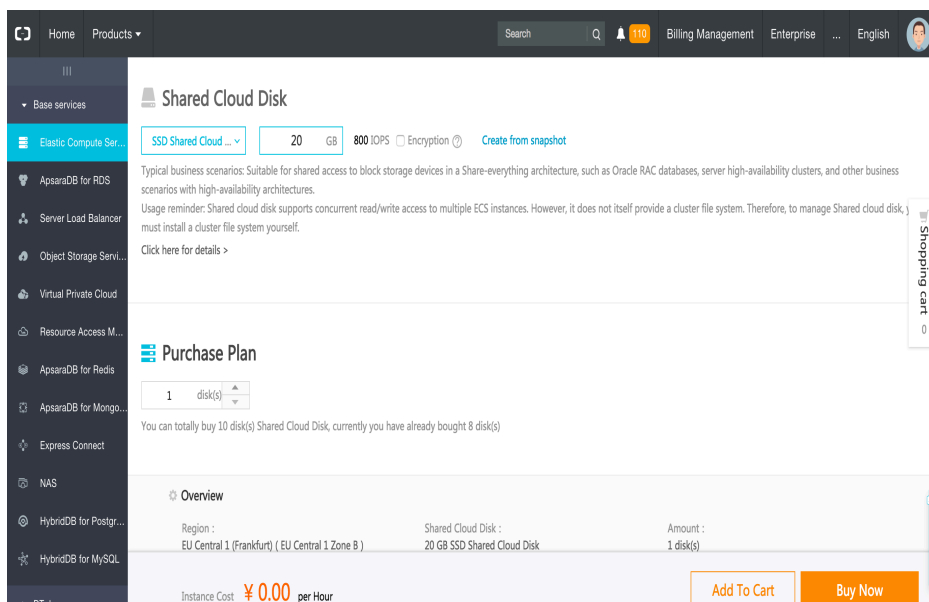
The Always On feature must be enabled for the server instance 'sqlserver' before you can create an availability group on this instance. To enable Always On:

- Open the SQL Server Configuration Manager.
- Select SQL Server Services.
- Right-click the SQL Server instance name.
- Select Properties, and use the Always On High Availability tab of the SQL Server Properties dialog. (ObjectExplorer).

You should create 3 ECS instances, one is for the DC(domain controller) and DNS, the other two are the cluster nodes. You need to create a shared cloud disk as the shared block storage as below:



and then choose the region and size, please pay attention you can only buy 20GB at least as the shared block storage.



For how to plan, configure your WSFC and Always on group please see Microsoft official website.

Backup

The SQL Server backup and restore component provides an essential safeguard for protecting critical data stored in your SQL Server databases. We strongly recommend place the databases, logs, backups on separate SSD Cloud disk. Placing the data and backups on SSD Cloud disk also enhances the I/O performance for both writing backups and the production use of the database. A backup and restore strategy contains a backup portion and a restore portion. Designing an effective backup and restore strategy requires careful planning, implementation, and testing. There is no difference

between doing SQL Server database backup and restore from Alibaba cloud and on premise version. [Backing Up and Restoring How-to Topics \(SQL Server Management Studio\)](#) and [Backing Up and Restoring How-to Topics \(Transact-SQL\)](#)

provides best practice for how to implement a solid backup and maintenance action.

Use the Cloud SSD disk to store your backups and then copy them into OSS bucket or you can use the windows task scheduler copy them as the regular task.

Performance

This section provides you with how to tuning your SQL server instance performance on ECS instances. Running SQL Server on ECS environment we recommend you continue using the same database performance tuning options that are applicable to SQL Server in on-premises server environment SQL Server Enterprise Edition has a long list of added capabilities over Standard Edition. If you are migrating an existing license to ECS, there are some performance options that you might consider implementing.

Separate Cloud Disk

We recommend you to place the databases, logs, backups on separate SSD Cloud disk during you create your SQL Server instances

Table Compression

Generally, data compression reduces the space occupied by the data. It can help improve performance of I/O intensive workloads because the data is stored in fewer pages and queries need to read fewer pages from disk. Data compression can be performed for a table, clustered index, non-clustered index. We recommend you to enable table and index compression. It might seem counter-intuitive that compressing tables could make your system perform faster, but in most cases, that's what happens. The tradeoff is using a small amount of CPU cycles to compress the data and eliminate the extra disk IO required to read and write the bigger blocks. Generally, the less disk IO your system uses, the better its performance will be. Instructions for estimating and enabling table and index compression please refer to [Microsoft website](#)

Enable buffer pool extension (BPE)

We recommend you to use the buffer pool extension to speed data access. The buffer pool extension feature enables you to push clean pages to the SSD Cloud disk, instead of dropping them. This works along the same lines as virtual memory, which is to say by swapping, and gives you access to the clean pages on the SSD Cloud disk, which is faster than you would get by going to the regular disk to fetch the data. This technique is not nearly as fast as having enough memory, but it can give you a modest increase in throughput when your available memory is low. For how to enable BPE and the technology detail please refer to [Microsoft website](#)

Max degree of parallelism setting

We recommend you to configure the max degree of parallelism option to 8. When your SQL Server instance runs on ECS that has more than one processor, it detects the best degree of parallelism, that is, the number of processors employed to run a single statement, for each parallel plan execution. You can use the max degree of parallelism option to limit the number of processors to use in parallel plan execution.

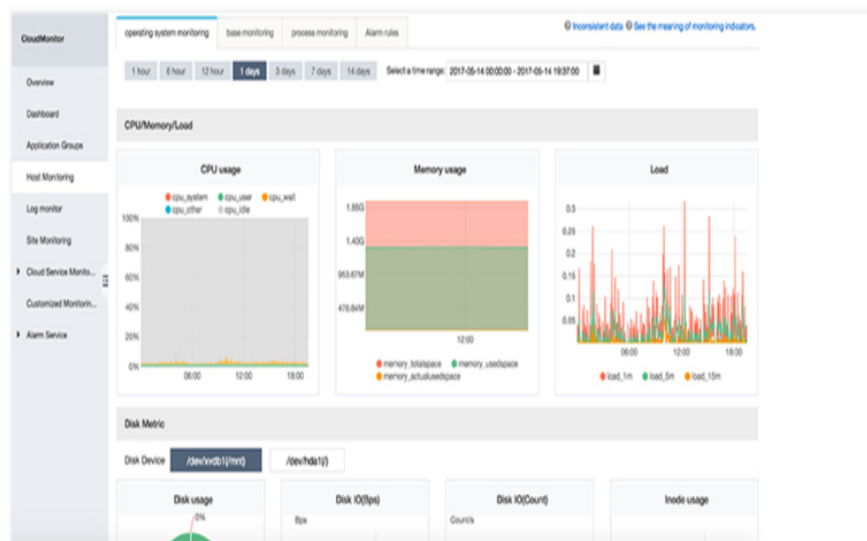
This value is set using `sp_configure` system procedure or you use SQL Server Management Studio. The default value is 0 which means there is no upper limit and SQL Server can use all available processors. If you set Max Degree of Parallelism to 1 then all queries will execute serially. This setting is ignored on servers with a single processor. Occasionally you might find that parallelism actually hinders performance of some queries. In this case the cost of initializing and synchronizing parallel plans might exceed the benefit of running portions of the query on multiple threads. If you feel that serial execution of a particular query can provide better performance you can override this setting using MAXDOP option within an individual query. For how to configure please refer to [Microsoft website](#)

Monitor

We recommend you to use **CloudMonitor** to monitor your ECS instances. Make sure that your ECS monitoring agents are functional to collect metric data. Otherwise, you must install the agent manually. For more information, see [How to install CloudMonitor agent](#).

More metrics are available.

More than 20 metrics are supported, such as `cpu.user`, `cpu.system`, `cpu.iowait`, `netout.packages`, `netout.errorpackage`. For OS metrics, the collection granularity is 15 seconds. [Which metrics are supported in the latest version?](#)



Monitor Current Host

Monitor all the Hosts

Monitoring capability

CloudMonitor allows more than 30 metrics covering CPU, memory, disk, and network to meet the basic monitoring and O&M requirements of the servers. Click [here](#) to view the full list of metrics the switch.

Alarm capability

CloudMonitor provides alarm service for all metrics, allowing you to set alarm rules for individual servers, application groups, and all the other resources. You can use the alarm service as per your business requirements. CloudMonitor provides Host monitoring metrics to set alarm rules for individual servers, application groups, and all the other resources. You can use the alarm services as per your business requirements. You can use the alarm service directly in the host monitoring list, or use it in your application group once you add servers to the group. You can add the alarm rules directly in the host monitoring list, or use it in your application group once you add servers to the group. For how to create an alarm service, please see [here](#)

Management Studio

You can use SQL Server Management Studio to perform most administrative tasks. This section provides you with how to manage your SQL server instance on ECS instances.

Remote SQL Server Management Studio

Microsoft offered SQL Server Management Studio to configure SQL Server databases. You can download and installed it on your desktop, connect to database remotely.

Default SQL Server Management Studio

You also can use the default SQL Server Management Studio which running on the instance itself. With this method you should connect to your SQL Server instance through RDP. SQL Server 2012 and SQL Server 2014 both include the SQL Server Management Studio by default. For SQL Server 2016, you must download the SQL Server Management Studio from the Microsoft website and install it on the instance. By default, SQL Server uses Windows Authentication mode to control remote access to SQL Server itself. If you need to use SQL Server Authentication mode, change the authentication mode.