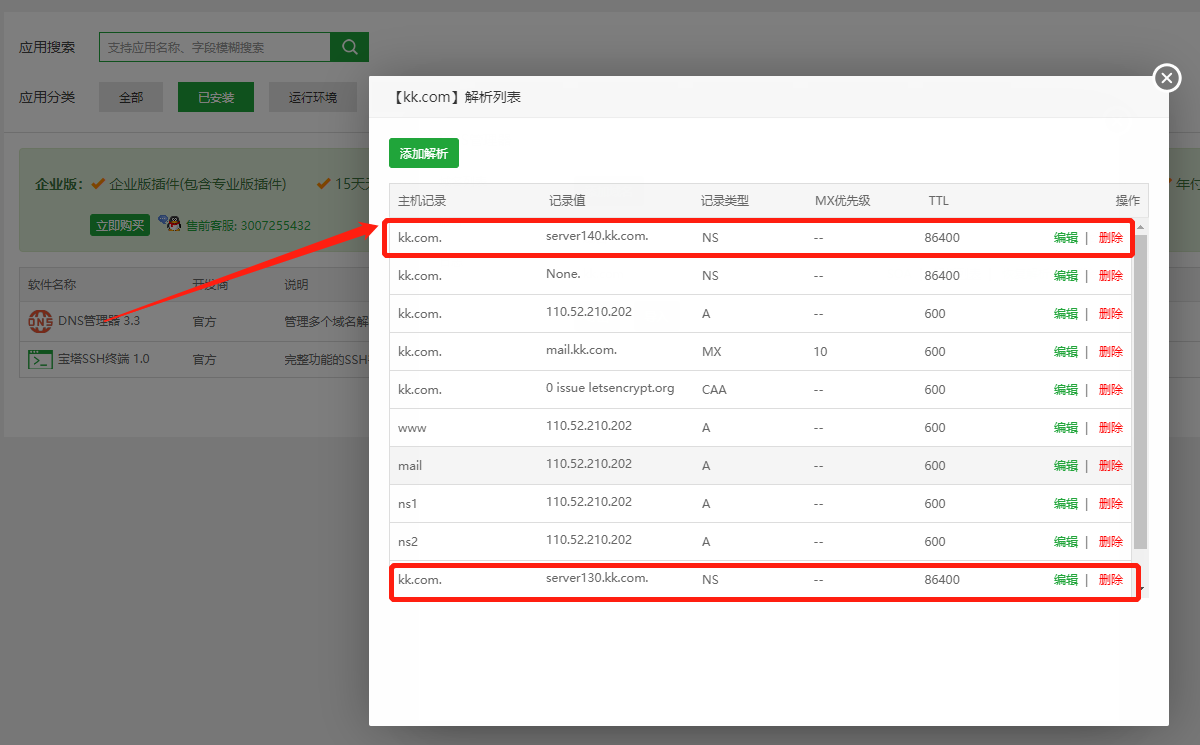
#### 所有服务器：hosts 绑定IP和服务器名称

此示例中将140作为主副本

|  |
| --- |
| # vim /etc/hosts  192.168.3.140 server140 server140.kk.com  192.168.3.130 server130 server130.kk.com  192.168.3.150 server150 server150.kk.com |

#### 所有服务器：DNS 服务器指针设置

宝塔DNS管理器



DNS管理器



#### 安装sql server

参考

<https://www.cnblogs.com/ios9/p/9533404.html>或

<https://www.jb51.net/article/184433.htm>，

在Linux各个节点机器上安装sql server，步骤如下：

【若提示账户无权限，可以通过sudo su切换到root账号】

【cmd ssh连接到Linux命令：ssh zjzt@192.168.3.140】

1. 下载 Microsoft SQL Server 2019 Red Hat 存储库配置文件：

|  |
| --- |
| curl -o /etc/yum.repos.d/mssql-server.repo <https://packages.microsoft.com/config/rhel/8/mssql-server-2019.repo> |

1. 运行以下命令以安装 SQL Server：

|  |
| --- |
| yum install -y mssql-server |

1. 包安装完成后，运行 mssql-conf setup，按照提示设置 SA 密码并选择版本。

|  |
| --- |
| /opt/mssql/bin/mssql-conf setup |

1. 下载客户端工具的源、安装客户端工具

|  |
| --- |
| curl https://packages.microsoft.com/config/rhel/7/prod.repo > /etc/yum.repos.d/msprod.repo |

1. 安装mssql-tools

|  |
| --- |
| yum install -y mssql-tools |

1. 配置环境变量

|  |
| --- |
| echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bash\_profile |

1. 为了后期使用shell添加变量

|  |
| --- |
| echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bashrc  source ~/.bashrc |

1. Sqlserver配置

|  |
| --- |
| #启动sql server 代理服务  #/opt/mssql/bin/mssql-conf set sqlagent.enabled true |

1. 测试

|  |
| --- |
| sqlcmd -S localhost -U sa -P zjzt#123456 |

#### SQL Server证书及权限设置

参考<https://blog.csdn.net/kk185800961/article/details/89483724>

1. 所有服务器：启用AG功能

|  |
| --- |
| /opt/mssql/bin/mssql-conf set hadr.hadrenabled 1  systemctl restart mssql-server |

1. 先尝试用yum安装mssql-server-ha，若不能安装则继续3-6.

|  |
| --- |
| yum install -y mssql-server-ha |

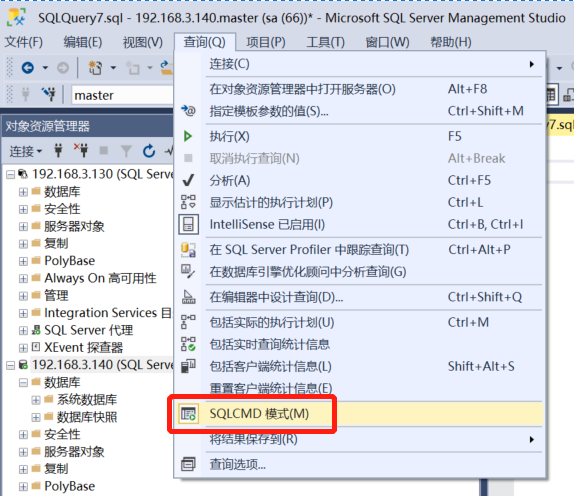
1. Centos8需下载mssql-server-ha和resource-agents包，因yum安装不成功，故需下载文件，再用rpm命令安装【注意包的版本要对】；
2. [mssql-server-ha](https://packages.microsoft.com/rhel/7/mssql-server-2019/mssql-server-ha-15.0.4138.2-1.x86_64.rpm)
3. [resource-agents](http://mirror.centos.org/centos/8/HighAvailability/x86_64/os/Packages/resource-agents-4.1.1-90.el8_4.5.x86_64.rpm)
4. 所有服务器：将下载后的安装包上传至Linux机器文件夹，基于包所在文件夹执行安装命令【需先安装resource-agents】：

|  |
| --- |
| rpm -ivh resource-agents-4.1.1-90.el8\_4.5.x86\_64.rpm  rpm -ivh mssql-server-ha-15.0.4138.2-1.x86\_64.rpm  yum info mssql-server-ha |

1. 所有服务器：启用防火墙上的 AlwaysOn AG 端点默认端口5022

|  |
| --- |
| firewall-cmd --zone=public --add-port=5022/tcp --permanent  firewall-cmd --reload |

1. SSMS连接到任意一个实例，[查询] 选项启用 [SQLCMD模式],如下图：



1. 批量执行以下sql脚本【注意：创建主密钥的密码不能带”#”】

|  |
| --- |
| ----------------------------------------------------------------------  --#【 server140 副本】  ----------------------------------------------------------------------  :CONNECT 192.168.3.140 -U sa -P zjzt#123456  GO  USE master  GO  -- 1. 创建数据库主密钥  --（数据库主密钥是对称密钥，用于保护数据库中存在的证书和非对称密钥的私钥）  CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'Zjzt123456';  GO  -- 2. 创建将用于加密可用性组端点的证书  CREATE CERTIFICATE LINUXHA\_SQLAG1\_cert  WITH SUBJECT = 'server140 certificate for Availability Group'  GO  --3. 主副本：使用证书创建 AlwaysOn AG 端点并进行身份验证  CREATE ENDPOINT Endpoint\_AvailabilityGroup  STATE = STARTED  AS TCP  (  LISTENER\_PORT = 5022, LISTENER\_IP = ALL  )  FOR DATABASE\_MIRRORING  (  AUTHENTICATION = CERTIFICATE LINUXHA\_SQLAG1\_cert,  ENCRYPTION = REQUIRED ALGORITHM AES,  ROLE = ALL  );  GO  -- 4. 主副本：将证书导出到文件  BACKUP CERTIFICATE LINUXHA\_SQLAG1\_cert  TO FILE = '/var/opt/mssql/data/LINUXHA\_SQLAG1\_cert.cer';  GO    ----------------------------------------------------------------------  --#【 server130 副本】  ----------------------------------------------------------------------  :CONNECT 192.168.3.130 -U sa -P zjzt#123456  GO  USE master  GO  CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'Zjzt123456';  GO  CREATE CERTIFICATE LINUXHA\_SQLAG2\_cert  WITH SUBJECT = 'server130 certificate for Availability Group'  GO  CREATE ENDPOINT Endpoint\_AvailabilityGroup  STATE = STARTED  AS TCP  (  LISTENER\_PORT = 5022, LISTENER\_IP = ALL  )  FOR DATABASE\_MIRRORING  (  AUTHENTICATION = CERTIFICATE LINUXHA\_SQLAG2\_cert,  ENCRYPTION = REQUIRED ALGORITHM AES,  ROLE = ALL  );  GO  BACKUP CERTIFICATE LINUXHA\_SQLAG2\_cert  TO FILE = '/var/opt/mssql/data/LINUXHA\_SQLAG2\_cert.cer';  GO    ----------------------------------------------------------------------  --#【 server150 副本】  ----------------------------------------------------------------------  :CONNECT 192.168.3.150 -U sa -P zjzt#123456  GO  USE master  GO  CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'Zjzt123456';  GO  CREATE CERTIFICATE LINUXHA\_SQLAG3\_cert  WITH SUBJECT = 'server150 certificate for Availability Group'  GO  CREATE ENDPOINT Endpoint\_AvailabilityGroup  STATE = STARTED  AS TCP  (  LISTENER\_PORT = 5022, LISTENER\_IP = ALL  )  FOR DATABASE\_MIRRORING  (  AUTHENTICATION = CERTIFICATE LINUXHA\_SQLAG3\_cert,  ENCRYPTION = REQUIRED ALGORITHM AES,  ROLE = ALL  );  GO  BACKUP CERTIFICATE LINUXHA\_SQLAG3\_cert  TO FILE = '/var/opt/mssql/data/LINUXHA\_SQLAG3\_cert.cer';  GO |

1. 若执行上述sql后报错【不报错不执行】，需要在[SQLCMD模式]下删除证书及主密钥等信息，使用下面sql脚本：

|  |
| --- |
| :CONNECT 192.168.3.140 -U sa -P zjzt#123456  GO  use master  Go  drop endpoint Endpoint\_AvailabilityGroup  drop certificate LINUXHA\_SQLAG1\_cert  drop master key  :CONNECT 192.168.3.130 -U sa -P zjzt#123456  GO  use master  Go  drop endpoint Endpoint\_AvailabilityGroup  drop certificate LINUXHA\_SQLAG2\_cert  drop master key  :CONNECT 192.168.3.150 -U sa -P zjzt#123456  GO  use master  Go  drop endpoint Endpoint\_AvailabilityGroup  drop certificate LINUXHA\_SQLAG3\_cert  drop master key |

1. 将各服务器上的证书相互传给其他服务器，可以用宝塔下载再上传，也可以执行下列命令。

|  |
| --- |
| #server140上执行，会提示输入密码  scp /var/opt/mssql/data/LINUXHA\_SQLAG1\_cert.cer root@server130:/var/opt/mssql/data/  scp /var/opt/mssql/data/LINUXHA\_SQLAG1\_cert.cer root@server150:/var/opt/mssql/data/ |

|  |
| --- |
| #server130上执行  scp /var/opt/mssql/data/LINUXHA\_SQLAG2\_cert.cer root@ server140:/var/opt/mssql/data/  scp /var/opt/mssql/data/LINUXHA\_SQLAG2\_cert.cer root@server150:/var/opt/mssql/data/ |

|  |
| --- |
| #server150上执行  scp /var/opt/mssql/data/LINUXHA\_SQLAG3\_cert.cer root@server140:/var/opt/mssql/data/  scp /var/opt/mssql/data/LINUXHA\_SQLAG3\_cert.cer root@server130:/var/opt/mssql/data/ |

1. 所有服务器：授予证书文件所有者权限

|  |
| --- |
| chown mssql:mssql /var/opt/mssql/data/LINUXHA\_SQLAG\*  ll /var/opt/mssql/data/LINUXHA\_SQLAG\* |

1. 各实例证书还原及账号授权

SSMS 连接到任意一个实例，查询选项启用[SQLCMD模式]，批量执行以下脚本。

|  |
| --- |
| ----------------------------------------------------------------------  --#【 server140 副本】  ----------------------------------------------------------------------  :CONNECT 192.168.3.140 -U sa -P zjzt#123456  GO  USE master  GO  -- 11 创建登录账号以供辅助副本使用  CREATE LOGIN login\_ag WITH PASSWORD = 'zjzt@123456';  GO  -- 12. 为登录账号创建用户  CREATE USER login\_ag FOR LOGIN login\_ag  GO  -- 13. 创建其他副本的证书  CREATE CERTIFICATE LINUXHA\_SQLAG2\_cert  AUTHORIZATION login\_ag  FROM FILE = '/var/opt/mssql/data/LINUXHA\_SQLAG2\_cert.cer'  GO  CREATE CERTIFICATE LINUXHA\_SQLAG3\_cert  AUTHORIZATION login\_ag  FROM FILE = '/var/opt/mssql/data/LINUXHA\_SQLAG3\_cert.cer'  GO  -- 14. 授予登录账号的CONNECT权限  GRANT CONNECT ON ENDPOINT::Endpoint\_AvailabilityGroup  TO [login\_ag];  GO    ----------------------------------------------------------------------  --#【 server130 副本】  ----------------------------------------------------------------------  :CONNECT 192.168.3.130 -U sa -P zjzt#123456  GO  USE master  GO  -- 11 创建登录账号以供辅助副本使用  CREATE LOGIN login\_ag WITH PASSWORD = 'zjzt@123456';  GO  -- 12. 为登录账号创建用户  CREATE USER login\_ag FOR LOGIN login\_ag  GO  -- 13. 创建其他副本的证书  CREATE CERTIFICATE LINUXHA\_SQLAG1\_cert  AUTHORIZATION login\_ag  FROM FILE = '/var/opt/mssql/data/LINUXHA\_SQLAG1\_cert.cer'  GO  CREATE CERTIFICATE LINUXHA\_SQLAG3\_cert  AUTHORIZATION login\_ag  FROM FILE = '/var/opt/mssql/data/LINUXHA\_SQLAG3\_cert.cer'  GO  -- 14. 授予登录账号的CONNECT权限  GRANT CONNECT ON ENDPOINT::Endpoint\_AvailabilityGroup  TO [login\_ag];  GO    ----------------------------------------------------------------------  --#【 server150 副本】  ----------------------------------------------------------------------  :CONNECT 192.168.3.150 -U sa -P zjzt#123456  GO  USE master  GO  -- 11 创建登录账号以供辅助副本使用  CREATE LOGIN login\_ag WITH PASSWORD = 'zjzt@123456';  GO  -- 12. 为登录账号创建用户  CREATE USER login\_ag FOR LOGIN login\_ag  GO  -- 13. 创建其他副本的证书  CREATE CERTIFICATE LINUXHA\_SQLAG1\_cert  AUTHORIZATION login\_ag  FROM FILE = '/var/opt/mssql/data/LINUXHA\_SQLAG1\_cert.cer'  GO  CREATE CERTIFICATE LINUXHA\_SQLAG2\_cert  AUTHORIZATION login\_ag  FROM FILE = '/var/opt/mssql/data/LINUXHA\_SQLAG2\_cert.cer'  GO  -- 14. 授予登录账号的CONNECT权限  GRANT CONNECT ON ENDPOINT::Endpoint\_AvailabilityGroup  TO [login\_ag];  GO |

#### Always On Availability Group配置

参考：<https://blog.csdn.net/kk185800961/article/details/89501123>

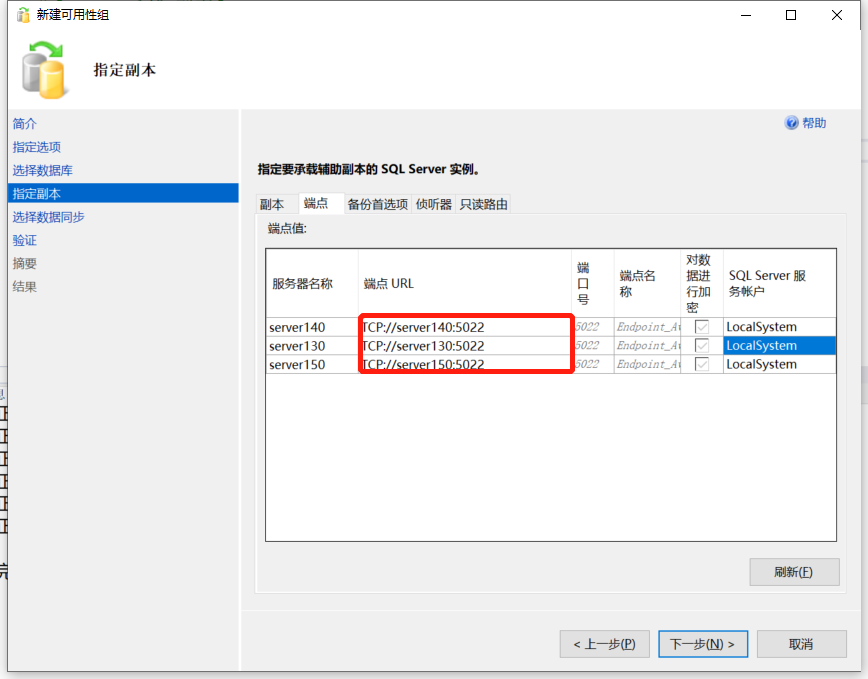
1. 前提
2. 加入AG组的数据库需要完整备份。
3. 由于sqlserver是以服务器hostname为服务器实例名称，所以需要修改每个服务器/etc/hostname文件，保证名称不重复；若是在安装了SQL Server后才修改的hostname，则需要在执行下列sql脚本，再重启服务器上的sqlserver服务

|  |
| --- |
| select @@servername  Sp\_dropserver 'localhost'--为第一条sql查出的服务器就名称  Sp\_addserver 'server150','local' --server150为新的服务器名称，local不需修改 |

重启sql服务

|  |
| --- |
| sudo systemctl stop mssql-server  sudo systemctl start mssql-server |

1. 端点url必须是可访问的地址



1. 查看侦听器状态

|  |
| --- |
| select \* from sys.availability\_group\_listener\_ip\_addresses |

1. 删除可用性组

|  |
| --- |
| DROP AVAILABILITY GROUP group\_name |

1. 用脚本创建AG

【若采用SSMS AG向导创建AG则不用执行以下脚本】

1. 在主节点上创建AG

|  |
| --- |
| CREATE AVAILABILITY GROUP [AG2]  WITH (DB\_FAILOVER = ON, CLUSTER\_TYPE = EXTERNAL)  FOR REPLICA ON  N'server140'  WITH (  ENDPOINT\_URL = N'tcp://server140:5022',  AVAILABILITY\_MODE = SYNCHRONOUS\_COMMIT,  FAILOVER\_MODE = EXTERNAL,  SEEDING\_MODE = AUTOMATIC,  SECONDARY\_ROLE (ALLOW\_CONNECTIONS = ALL)  ),  N'server130'  WITH (  ENDPOINT\_URL = N'tcp://server130:5022',  AVAILABILITY\_MODE = SYNCHRONOUS\_COMMIT,  FAILOVER\_MODE = EXTERNAL,  SEEDING\_MODE = AUTOMATIC,  SECONDARY\_ROLE (ALLOW\_CONNECTIONS = ALL)  ),  N'server150'  WITH (  ENDPOINT\_URL = N'tcp://server150:5022',  AVAILABILITY\_MODE = SYNCHRONOUS\_COMMIT,  FAILOVER\_MODE = EXTERNAL,  SEEDING\_MODE = AUTOMATIC,  SECONDARY\_ROLE (ALLOW\_CONNECTIONS = ALL)  );  ALTER AVAILABILITY GROUP [AG2] GRANT CREATE ANY DATABASE; |

1. 将从节点添加进去，【在每一个从节点上执行】

|  |
| --- |
| ALTER AVAILABILITY GROUP [AG2] JOIN WITH (CLUSTER\_TYPE = EXTERNAL);  ALTER AVAILABILITY GROUP [AG2] GRANT CREATE ANY DATABASE |

1. 【在主节点上执行】若只是测试数据库，则先创建测试数据库并备份

|  |
| --- |
| CREATE DATABASE [db1];  ALTER DATABASE [db1] SET RECOVERY FULL;  BACKUP DATABASE [db1]  TO DISK = N'/var/opt/mssql/data/db1.bak'; |

1. 【在主节点上执行】将数据库加入AG

|  |
| --- |
| ALTER AVAILABILITY GROUP [AG2] ADD DATABASE [db1]; |

#### Pacemaker 安装配置

参考：<https://blog.csdn.net/kk185800961/article/details/89506667>

##### 所有服务器，安装Pacemaker相关包

先尝试用yum安装下列包

|  |
| --- |
| yum install -y pacemaker pcs fence-agents-all |

若安装不上，则按下列步骤走。

###### 安装fence-agents-all

|  |
| --- |
| yum install -y fence-agents-all |

###### 下载Pacemaker及pcs相关包

|  |
| --- |
| [pacemaker-2.0.5-9.el8\_4.1.x86\_64.rpm](https://mirrors.aliyun.com/centos/8-stream/HighAvailability/x86_64/os/Packages/pacemaker-2.0.5-9.el8_4.1.x86_64.rpm?spm=a2c6h.13651111.0.0.89102f70IUXt5T&file=pacemaker-2.0.5-9.el8_4.1.x86_64.rpm)【因为和fence-agents-all共用了一些依赖包，故版本选择2.0.5】  -[pacemaker-cli-2.0.5-9.el8\_4.1.x86\_64.rpm](https://mirrors.aliyun.com/centos/8-stream/HighAvailability/x86_64/os/Packages/pacemaker-cli-2.0.5-9.el8_4.1.x86_64.rpm?spm=a2c6h.13651111.0.0.89102f70IUXt5T&file=pacemaker-cli-2.0.5-9.el8_4.1.x86_64.rpm)  -[corosync-3.1.0-3.el8\_4.1.x86\_64.rpm](https://mirrors.aliyun.com/centos/8.4.2105/HighAvailability/x86_64/os/Packages/corosync-3.1.0-3.el8_4.1.x86_64.rpm?spm=a2c6h.13651111.0.0.89102f70IUXt5T&file=corosync-3.1.0-3.el8_4.1.x86_64.rpm)  --[libknet1-1.18-1.el8.x86\_64.rpm](https://mirrors.aliyun.com/centos/8.4.2105/HighAvailability/x86_64/kickstart/Packages/libknet1-1.18-1.el8.x86_64.rpm?spm=a2c6h.13651111.0.0.89102f70IUXt5T&file=libknet1-1.18-1.el8.x86_64.rpm)  --[libnozzle1-1.18-1.el8.x86\_64.rpm](https://mirrors.aliyun.com/centos/8.4.2105/HighAvailability/x86_64/kickstart/Packages/libnozzle1-1.18-1.el8.x86_64.rpm?spm=a2c6h.13651111.0.0.89102f70IUXt5T&file=libnozzle1-1.18-1.el8.x86_64.rpm)  --[libknet1-crypto-nss-plugin-1.18-1.el8.x86\_64.rpm](https://mirrors.aliyun.com/centos/8-stream/HighAvailability/x86_64/os/Packages/libknet1-crypto-nss-plugin-1.18-1.el8.x86_64.rpm?spm=a2c6h.13651111.0.0.89102f70IUXt5T&file=libknet1-crypto-nss-plugin-1.18-1.el8.x86_64.rpm)  [pcs-0.10.8-4.el8.x86\_64.rpm](https://mirrors.aliyun.com/centos/8-stream/HighAvailability/x86_64/os/Packages/pcs-0.10.8-4.el8.x86_64.rpm?spm=a2c6h.13651111.0.0.89102f70IUXt5T&file=pcs-0.10.8-4.el8.x86_64.rpm)  -ruby 【用yum安装，不用下载】  -overpass-fonts 【用yum安装，不用下载】  - python3-cryptography【用yum安装，不用下载】  - python3-pyparsing【用yum安装，不用下载】  -[libknet1-plugins-all-1.18-1.el8.x86\_64.rpm](https://mirrors.aliyun.com/centos/8-stream/HighAvailability/x86_64/os/Packages/libknet1-plugins-all-1.18-1.el8.x86_64.rpm?spm=a2c6h.13651111.0.0.89102f70IUXt5T&file=libknet1-plugins-all-1.18-1.el8.x86_64.rpm)  --[libknet1-compress-plugins-all-1.18-1.el8.x86\_64.rpm](https://mirrors.aliyun.com/centos/8.4.2105/HighAvailability/x86_64/kickstart/Packages/libknet1-compress-plugins-all-1.18-1.el8.x86_64.rpm?spm=a2c6h.13651111.0.0.89102f70IUXt5T&file=libknet1-compress-plugins-all-1.18-1.el8.x86_64.rpm)  ---[libknet1-compress-bzip2-plugin-1.18-1.el8.x86\_64.rpm](https://mirrors.aliyun.com/centos/8-stream/HighAvailability/x86_64/os/Packages/libknet1-compress-bzip2-plugin-1.18-1.el8.x86_64.rpm?spm=a2c6h.13651111.0.0.89102f70IUXt5T&file=libknet1-compress-bzip2-plugin-1.18-1.el8.x86_64.rpm)  ---[libknet1-compress-lz4-plugin-1.18-1.el8.x86\_64.rpm](https://mirrors.aliyun.com/centos/8-stream/HighAvailability/x86_64/os/Packages/libknet1-compress-lz4-plugin-1.18-1.el8.x86_64.rpm?spm=a2c6h.13651111.0.0.89102f70IUXt5T&file=libknet1-compress-lz4-plugin-1.18-1.el8.x86_64.rpm)  ---[libknet1-compress-lzma-plugin-1.18-1.el8.x86\_64.rpm](https://mirrors.aliyun.com/centos/8-stream/HighAvailability/x86_64/os/Packages/libknet1-compress-lzma-plugin-1.18-1.el8.x86_64.rpm?spm=a2c6h.13651111.0.0.89102f70IUXt5T&file=libknet1-compress-lzma-plugin-1.18-1.el8.x86_64.rpm)  ---[libknet1-compress-lzo2-plugin-1.18-1.el8.x86\_64.rpm](https://mirrors.aliyun.com/centos/8-stream/HighAvailability/x86_64/os/Packages/libknet1-compress-lzo2-plugin-1.18-1.el8.x86_64.rpm?spm=a2c6h.13651111.0.0.89102f70IUXt5T&file=libknet1-compress-lzo2-plugin-1.18-1.el8.x86_64.rpm)  ---[libknet1-compress-zlib-plugin-1.18-1.el8.x86\_64.rpm](https://mirrors.aliyun.com/centos/8-stream/HighAvailability/x86_64/os/Packages/libknet1-compress-zlib-plugin-1.18-1.el8.x86_64.rpm?spm=a2c6h.13651111.0.0.89102f70IUXt5T&file=libknet1-compress-zlib-plugin-1.18-1.el8.x86_64.rpm)  --[libknet1-crypto-plugins-all-1.18-1.el8.x86\_64.rpm](https://mirrors.aliyun.com/centos/8-stream/HighAvailability/x86_64/os/Packages/libknet1-crypto-plugins-all-1.18-1.el8.x86_64.rpm?spm=a2c6h.13651111.0.0.89102f70IUXt5T&file=libknet1-crypto-plugins-all-1.18-1.el8.x86_64.rpm)  ---[libknet1-crypto-openssl-plugin-1.18-1.el8.x86\_64.rpm](https://mirrors.aliyun.com/centos/8-stream/HighAvailability/x86_64/os/Packages/libknet1-crypto-openssl-plugin-1.18-1.el8.x86_64.rpm?spm=a2c6h.13651111.0.0.89102f70IUXt5T&file=libknet1-crypto-openssl-plugin-1.18-1.el8.x86_64.rpm)  -[python3-clufter-0.77.1-5.el8.noarch.rpm](https://mirrors.aliyun.com/centos/8-stream/HighAvailability/ppc64le/os/Packages/python3-clufter-0.77.1-5.el8.noarch.rpm?spm=a2c6h.13651111.0.0.89102f70IUXt5T&file=python3-clufter-0.77.1-5.el8.noarch.rpm)  --[clufter-bin-0.77.1-5.el8.x86\_64.rpm](https://mirrors.aliyun.com/centos/8-stream/HighAvailability/x86_64/os/Packages/clufter-bin-0.77.1-5.el8.x86_64.rpm?spm=a2c6h.13651111.0.0.89102f70IUXt5T&file=clufter-bin-0.77.1-5.el8.x86_64.rpm)  ---[clufter-common-0.77.1-5.el8.noarch.rpm](https://mirrors.aliyun.com/centos/8-stream/HighAvailability/aarch64/os/Packages/clufter-common-0.77.1-5.el8.noarch.rpm?spm=a2c6h.13651111.0.0.89102f70IUXt5T&file=clufter-common-0.77.1-5.el8.noarch.rpm) |

###### 上传包

在服务器上新建文件夹，用于上传下载好的包。

###### 安装包

前提：cd 进入所有包所在文件夹。

1. yum安装perl-TimeDate

|  |
| --- |
| yum install -y perl-TimeDate |

1. rpm安装pacemaker相关包

|  |
| --- |
| rpm -ivh pacemaker-cli-2.0.5-9.el8\_4.1.x86\_64.rpm |

1. rpm安装corosync相关包

|  |
| --- |
| rpm -ivh libknet1-1.18-1.el8.x86\_64.rpm  rpm -ivh libnozzle1-1.18-1.el8.x86\_64.rpm  rpm -ivh libknet1-crypto-nss-plugin-1.18-1.el8.x86\_64.rpm  rpm -ivh corosync-3.1.0-3.el8\_4.1.x86\_64.rpm |

1. 安装pacemaker包

|  |
| --- |
| rpm -ivh pacemaker-2.0.5-9.el8\_4.1.x86\_64.rpm |

1. yum安装pcs相关包:ruby及overpass-fonts

|  |
| --- |
| yum install -y ruby  yum install -y overpass-fonts  yum install -y python3-cryptography  yum install -y python3-pyparsing |

1. rpm安装pcs相关包

|  |
| --- |
| rpm -ivh libknet1-compress-bzip2-plugin-1.18-1.el8.x86\_64.rpm  rpm -ivh libknet1-compress-lz4-plugin-1.18-1.el8.x86\_64.rpm  rpm -ivh libknet1-compress-lzma-plugin-1.18-1.el8.x86\_64.rpm  rpm -ivh libknet1-compress-lzo2-plugin-1.18-1.el8.x86\_64.rpm  rpm -ivh libknet1-compress-zlib-plugin-1.18-1.el8.x86\_64.rpm  rpm -ivh libknet1-compress-plugins-all-1.18-1.el8.x86\_64.rpm  rpm -ivh libknet1-crypto-openssl-plugin-1.18-1.el8.x86\_64.rpm  rpm -ivh libknet1-crypto-plugins-all-1.18-1.el8.x86\_64.rpm  rpm -ivh libknet1-plugins-all-1.18-1.el8.x86\_64.rpm  rpm -ivh clufter-common-0.77.1-5.el8.noarch.rpm  rpm -ivh clufter-bin-0.77.1-5.el8.x86\_64.rpm  rpm -ivh python3-clufter-0.77.1-5.el8.noarch.rpm  rpm -ivh pcs-0.10.8-4.el8.x86\_64.rpm |

##### 所有服务器：启动相应服务

pacemaker服务(暂不启动)，后续再添加 AlwaysOn AG 资源

|  |
| --- |
| sudo systemctl enable pcsd  sudo systemctl start pcsd  sudo systemctl enable pacemaker |

##### 所有服务器：配置Linux防火墙以允许Pacemaker通信（或禁用防火墙）【已关闭则不用执行】

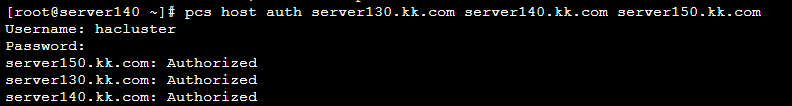
|  |
| --- |
| firewall-cmd --add-service=high-availability --zone=public --permanent    firewall-cmd --zone=public --add-port=2224/tcp --permanent  firewall-cmd --zone=public --add-port=3121/tcp –permanent  firewall-cmd --zone=public --add-port=5405/udp --permanent    firewall-cmd --reload |

##### 所有服务器：充当 Pacemaker 群集的节点，安装时创建的默认账户 hacluster ,现在创建相同密码：HA@cluster

|  |
| --- |
| passwd hacluster |

##### 主服务器(选140作为主服务器来配置)：在Linux集群3个节点之间设置身份验证(将输入账户hacluster的密码验证)

|  |
| --- |
| pcs host auth server130.kk.com server140.kk.com server150.kk.com |



##### 所有服务器：为防止集群中的异常文件残留，需要先删除已经存在的集群。

|  |
| --- |
| sudo pcs cluster destroy  sudo systemctl enable pacemaker |

##### 主服务器：创建Linux集群，名称为 LINUXHACLUSTER,添加3个节点

|  |
| --- |
| pcs cluster setup LINUXHACLUSTER server140 server130 server150 |

##### 主服务器：启动Linux群集服务

|  |
| --- |
| pcs cluster start --all  pcs cluster enable --all |

##### 查看当前集群状态

|  |
| --- |
| pcs cluster status |

##### 检查pacemaker服务

|  |
| --- |
| ps aux | grep pacemaker |

##### 检验Corosync的安装及当前corosync状态

|  |
| --- |
| corosync-cfgtool -s  corosync-cmapctl | grep members  pcs status corosync |

##### 所有服务器：隔离（STONITH）资源

隔离的目的是确保在配置资源过程中引起服务中断时，不会发生数据损坏，为了测试环境相对简单，需要禁用隔离，在实际环境中需要与管理员和设备供应商讨论是否需要启用。

Hyper-V和Azure尚不支持使用 STONITH RHEL 与 Pacemaker 群集，这里禁止。

(参考：[对于 Linux 部署 SQL Server 可用性基础知识](https://docs.microsoft.com/zh-cn/sql/linux/sql-server-linux-ha-basics?view=sql-server-2017))

|  |
| --- |
| pcs property set stonith-enabled=false |

##### 主服务器：检查配置是否正确（假若没有输出任何则配置正确）

|  |
| --- |
| crm\_verify -L -V |

##### 所有服务器：无法仲裁时候，选择忽略

|  |
| --- |
| pcs property set no-quorum-policy=ignore |

##### 所有服务器：设置群集属性群集重新检查间隔(可不配置)

cluster-recheck-interval 指示检查群集资源参数、 约束或其他群集选项中的更改的轮询间隔。 如果副本出现故障，群集将尝试重新启动的时间间隔，由绑定的副本failure-timeout值和cluster-recheck-interval值。 例如，如果failure-timeout设置为 60 秒和cluster-recheck-interval设置为 120 秒，在重新启动尝试的时间间隔大于 60 秒，但不超过 120 秒。 我们建议将故障超时设置为 60 秒和群集重新检查的间隔超过 60 秒的值。 不建议将群集重新检查间隔设置为较小的值。

若要将属性值更新为2 minutes运行【实践时设置的是15s】：

|  |
| --- |
| sudo pcs property set cluster-recheck-interval=70s |

##### 所有服务器：配置资源级策略

包括 RHEL 7.3 和 7.4） 的所有使用最新可用 Pacemaker 包 1.1.18-11.el7 分布引入开始失败-是的致命群集设置的行为更改其值为 false。 此更改会影响故障转移工作流。 如果主副本发生服务中断，群集应故障转移到其中一个可用的辅助副本。 相反，用户会注意到该群集会一直尝试启动失败的主副本。 如果该主永远不会处于联机状态 （由于的永久中断），群集永远不会故障转移到另一个可用的辅助副本。 由于此更改，以前推荐的配置来设置开始失败-是的致命将不再有效，需要恢复回其默认值设置true。 此外，需要更新，以包含 AG 资源failover-timeout属性。

|  |
| --- |
| sudo pcs property set start-failure-is-fatal=true |

若要更新ag\_cluster资源属性failure-timeout到60s运行：【先不执行，实践时设置的是10s】

|  |
| --- |
| pcs resource update AG1 meta failure-timeout=60s |

##### 为Pacemaker创建SQL Server登录账号，授予可用组权限【可用性组名称需修改】

|  |
| --- |
| --【副本 server140 执行】  :CONNECT 192.168.3.140 -U sa -P zjzt#123456  GO  USE master  GO  CREATE LOGIN pacemakerLogin WITH PASSWORD = 'PCM@pwd123456';  ALTER SERVER ROLE [sysadmin] ADD MEMBER [pacemakerLogin] ;  GO  GRANT ALTER, CONTROL, VIEW DEFINITION ON AVAILABILITY GROUP::AG1 TO pacemakerLogin  GO  GRANT VIEW SERVER STATE TO pacemakerLogin  GO    --【副本 server130 执行】  :CONNECT 192.168.3.130 -U sa -P zjzt#123456  GO  USE master  GO  CREATE LOGIN pacemakerLogin WITH PASSWORD = 'PCM@pwd123456';  ALTER SERVER ROLE [sysadmin] ADD MEMBER [pacemakerLogin] ;  GO  GRANT ALTER, CONTROL, VIEW DEFINITION ON AVAILABILITY GROUP::AG1 TO pacemakerLogin  GO  GRANT VIEW SERVER STATE TO pacemakerLogin  GO    --【副本 server150 执行】  :CONNECT 192.168.3.150 -U sa -P zjzt#123456  GO  USE master  GO  CREATE LOGIN pacemakerLogin WITH PASSWORD = 'PCM@pwd123456';  ALTER SERVER ROLE [sysadmin] ADD MEMBER [pacemakerLogin] ;  GO  GRANT ALTER, CONTROL, VIEW DEFINITION ON AVAILABILITY GROUP::AG1 TO pacemakerLogin  GO  GRANT VIEW SERVER STATE TO pacemakerLogin  GO |

##### 所有服务器：在本地文件系统上保存Pacemaker登录凭据(Pacemaker通过MSSQL账号pacemakerLogin连接)

|  |
| --- |
| # vim /var/opt/mssql/secrets/passwd  pacemakerLogin  PCM@pwd123456    # ls -l /var/opt/mssql/secrets  # chmod 400 /var/opt/mssql/secrets/passwd |

#### 配置监听器测试故障转移

参考：<https://blog.csdn.net/kk185800961/article/details/99069155>

<https://www.cnblogs.com/guarderming/p/12082936.html#_label12>

选择任意一个节点来做配置：

1. 在Pacemaker上创建Always On Availability组资源

|  |
| --- |
| pcs resource create LINUX\_SQLAG ocf:mssql:ag ag\_name=AG1 meta failure-timeout=10s promotable notify=true |

**LINUX\_SQLAG:** Pacemaker 集群资源的名称（可不必与 AlwaysOn AG 的名称相同，此测试设置相同）

**ocf:mssql:ag:** 由mssql-server-ha提供的Open Cluster Framework（OCF）资源代理的名称

**ag\_name=AG1:** AlwaysOn AG 的可用性组的名称

**promotable:** 将资源定义为 promotable clones【旧版本：master/slave 克隆资源】

**notify=true:**

1. 为Always On Availability Group侦听器创建虚拟IP地址资源

|  |
| --- |
| pcs resource create AGListener\_VIP ocf:heartbeat:IPaddr2 ip=192.168.3.222 |

**AGListener\_VIP:** 虚拟IP地址资源的名称

**ocf:heartbeat:IPaddr2:** 管理虚拟IPv4地址的Open Cluster Framework（OCF）资源代理的名称

**ip=192.168.3.222:** AlwaysOn AG 虚拟IP 与各节点在同一网段且不重复即可

**cidr\_netmask=24:** AlwaysOn AG 的可用性组的侦听器子网掩码

1. 将Always On Availability Group资源配置为在与虚拟IP地址资源相同的计算机上运行

|  |
| --- |
| pcs constraint colocation add AGListener\_VIP with LINUX\_SQLAG-clone INFINITY with-rsc-role=Master |

**AGListener\_VIP:** 虚拟IP地址资源的名称

**LINUX\_SQLAG-clone:** AlwaysOn AG 资源的克隆别名

**INFINITY:** 分配给资源约束的分数;这意味着约束是必需的

**with-rsc-role=Master:** 约束的附加属性;这意味着此约束与主克隆（或Always On Availability Group主副本）相关联

由于Always On Availability Group侦听器名称只能将客户端应用程序重定向到主副本，因此可用性组和侦听器名称必须始终在同一群集节点中运行。

1. 配置群集资源应该开始/停止的顺序

|  |
| --- |
| pcs constraint order promote LINUX\_SQLAG-clone then start AGListener\_VIP |

**promote**: 约束行为，将资源从slave 提升为 master 资源  
**LINUX\_SQLAG-clone:** AlwaysOn AG 资源的克隆别名  
**start**: 初步操作完成后下一步的动作  
**AGListener\_VIP**: 虚拟IP地址资源的名称



在WSFC中，事件序列如下：

停止当前主副本上的可用性组

在当前主副本上停止侦听器名称

在新的主副本上启动侦听器

在新主副本上启动可用性组

在侦听器名称上定义约束时，事件序列如下：

停止当前主副本上的虚拟IP地址资源

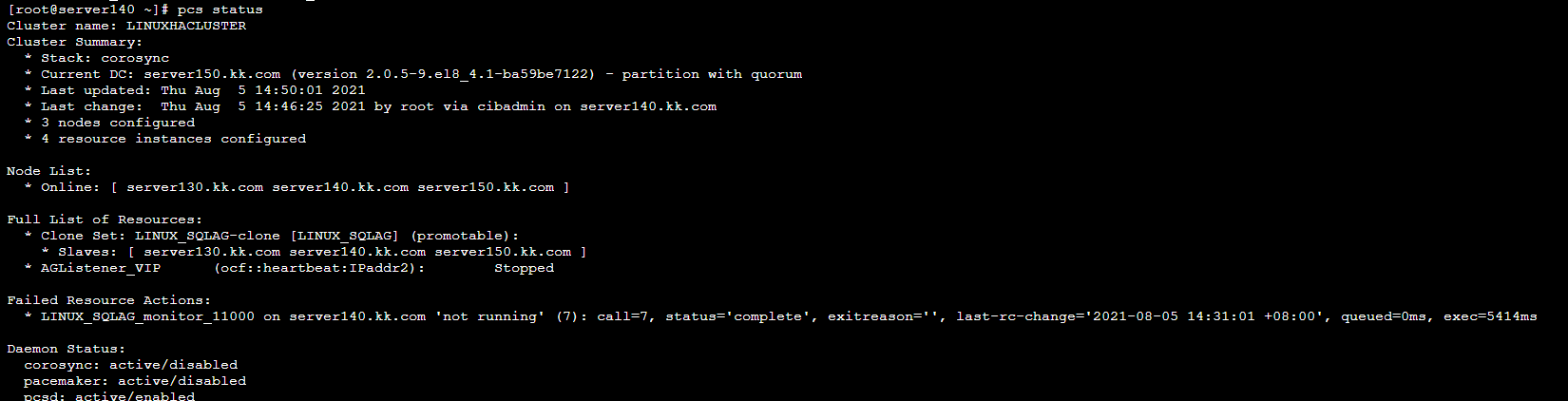
停止当前主副本上的可用性组

在新的主副本上启动虚拟IP地址资源

在新主副本上启动可用性组

1. 验证Always On可用性组配置是否正常

|  |
| --- |
| pcs status |



可在每个节点上执行:

|  |
| --- |
| SELECT @@SERVERNAME as replica\_name, @@VERSION, host\_platform, host\_distribution, host\_release  FROM sys.dm\_os\_host\_info  GO  SELECT a.name as AG\_Name, a.cluster\_type\_desc,b.dns\_name,c.ip\_address, c.ip\_subnet\_mask  FROM sys.availability\_groups a  INNER JOIN sys.availability\_group\_listeners b ON a.group\_id=b.group\_id  INNER JOIN sys.availability\_group\_listener\_ip\_addresses c ON b.listener\_id=c.listener\_id  GO |

1. SSMS连接虚拟IP，查看AG信息

|  |
| --- |
| --sql(VIP)  -- group info  SELECT  g.name as ag\_name,  rgs.primary\_replica,  rgs.primary\_recovery\_health\_desc as recovery\_health,  rgs.synchronization\_health\_desc as sync\_health  From sys.dm\_hadr\_availability\_group\_states as rgs  JOIN sys.availability\_groups AS g  ON rgs.group\_id = g.group\_id  --replicas info  SELECT  g.name as ag\_name,  r.replica\_server\_name,  rs.is\_local,  rs.role\_desc as role,  rs.operational\_state\_desc as op\_state,  rs.connected\_state\_desc as connect\_state,  rs.synchronization\_health\_desc as sync\_state,  rs.last\_connect\_error\_number,  rs.last\_connect\_error\_description  From sys.dm\_hadr\_availability\_replica\_states AS rs  JOIN sys.availability\_replicas AS r  ON rs.replica\_id = r.replica\_id  JOIN sys.availability\_groups AS g  ON g.group\_id = r.group\_id  --DB level  SElECT  g.name as ag\_name,  r.replica\_server\_name,  DB\_NAME(drs.database\_id) as [database\_name],  drs.is\_local,  drs.is\_primary\_replica,  synchronization\_state\_desc as sync\_state,  synchronization\_health\_desc as sync\_health,  database\_state\_desc as db\_state  FROM sys.dm\_hadr\_database\_replica\_states AS drs  JOIN sys.availability\_replicas AS r  ON r.replica\_id = drs.replica\_id  JOIN sys.availability\_groups AS g  ON g.group\_id = drs.group\_id  ORDER BY g.name, drs.is\_primary\_replica DESC;  GO |

1. 手动故障转移

在主节点上，将150转换为主数据库：

|  |
| --- |
| sudo pcs resource move ag\_cluster-master server150 --master |

SSMS连接虚拟IP，查看目前的服务器

|  |
| --- |
| SELECT @@SERVERNAME |

#### 已有集群配置故障转移

在已有故障转移集群上为新的AG组配置故障转移，步骤如下：

1. 参考1.5创建新的AG；
2. 授予pacemakerLogin新建AG权限

|  |
| --- |
| GRANT ALTER, CONTROL, VIEW DEFINITION ON AVAILABILITY GROUP::AG1 TO pacemakerLogin |

1. 重新 配置监听器测试故障转移 。

#### Docker版本参考

LANG=zh\_CN.utf8

Docker:<https://blog.csdn.net/qianglei6077/article/details/107055554>

【虚拟IP不能添加，可以用HAProxy来代理；】

Yaml文件：

|  |
| --- |
| version: '3'    services:    db1:      container\_name: sqlNode1      image: sqlag2019:ha      hostname: sqlNode1      domainname: lab.local      environment:        SA\_PASSWORD: "zjzt#123456"        ACCEPT\_EULA: "Y"      ports:          - "1501:1433"      volumes:          - /data/DB/node1:/var/opt/mssql/data      extra\_hosts:        sqlNode2.labl.local: "172.16.238.22"        sqlNode3.labl.local: "172.16.238.23"      networks:          internal:                  ipv4\_address: 172.16.238.21      db2:      container\_name: sqlNode2      image: sqlag2019:ha      hostname: sqlNode2      domainname: lab.local      environment:        SA\_PASSWORD: "zjzt#123456"        ACCEPT\_EULA: "Y"      ports:          - "1502:1433"      volumes:          - /data/DB/node2:/var/opt/mssql/data      extra\_hosts:        sqlNode1.lab.local: "172.16.238.21"        sqlNode3.lab.local: "172.16.238.23"      networks:          internal:                  ipv4\_address: 172.16.238.22      db3:      container\_name: sqlNode3      image: sqlag2019:ha      hostname: sqlNode3      domainname: lab.local      environment:        SA\_PASSWORD: "zjzt#123456"        ACCEPT\_EULA: "Y"      ports:          - "1503:1433"      volumes:          - /data/DB/node3:/var/opt/mssql/data      extra\_hosts:        sqlNode1.lab.local: "172.16.238.21"        sqlNode2.lab.local: "172.16.238.22"      networks:          internal:                  ipv4\_address: 172.16.238.23    networks:      internal:        ipam:              driver: default              config:                  - subnet: 172.16.238.0/24 |