DB自动化工具包

NetApp Solutions

NetApp
April 12, 2024

目录

DB自动化工具包

自动化Oracle迁移

NetApp解决方案工程团队

目的

此工具包可利用FSx ONTAP存储和EC2计算实例作为目标基础架构、自动将Oracle数据库从内部迁移到AWS云。假定客户已在CDB/PDB模型中部署了内部Oracle数据库。该工具包允许客户使用Oracle PDB重新定位操作步骤(具有最大可用性选项)从Oracle主机上的容器数据库重新定位命名PDB。这意味着、任何内部存储阵列上的源PDB都会重新定位到新容器数据库、而服务中断量极小。Oracle重新定位操作步骤将在数据库联机时移动Oracle数据文件。之后、当所有数据文件迁移到AWS云时、它会在切换时将用户会话从内部重新路由到重新定位的数据库服务。突出显示的技术是经验证的Oracle PDB热克隆方法。

> 虽然迁移工具包是在AWS云基础架构上开发和验证的、但它是基于Oracle应用程序级解决方案构建的。因此、该工具包适用于其他公有云平台、例如Azure、GCP等

此解决方案 可解决以下使用情形：

- 在内部源数据库服务器上创建迁移用户并授予所需权限。
- 在源PDB处于联机状态时、将PDB从内部CDB重新定位到云中的目标CDB、直到切换为止。

audience

此解决方案 适用于以下人员：

- 将Oracle数据库从内部环境迁移到AWS云的数据库管理人员。
- 一名数据库解决方案架构师、对从内部环境向AWS云迁移Oracle数据库感兴趣。
- 负责管理支持Oracle数据库的AWS FSx ONTAP存储的存储管理员。
- 喜欢将Oracle数据库从内部环境迁移到AWS云的应用程序所有者。

许可证

解决方案 部署

部署的前提条件

部署需要满足以下前提条件。

```
Ansible v.2.10 and higher
ONTAP collection 21.19.1
Python 3
Python libraries:
  netapp-lib
  xmltodict
  jmespath
```

```
Source Oracle CDB with PDBs on-premises
Target Oracle CDB in AWS hosted on FSx and EC2 instance
Source and target CDB on same version and with same options installed
```

```
Network connectivity
  Ansible controller to source CDB
  Ansible controller to target CDB
  Source CDB to target CDB on Oracle listener port (typical 1521)
```

下载此工具包

```
git clone https://github.com/NetApp/na_ora_aws_migration.git
```

主机变量配置

主机变量在名为｛｛host_name｝｝.yml的host_vars目录中定义。其中包括一个示例主机变量文件host_name.yml、用于演示典型配置。以下是主要注意事项：

```
Source Oracle CDB - define host specific variables for the on-prem CDB
  ansible_host: IP address of source database server host
  source_oracle_sid: source Oracle CDB instance ID
  source_pdb_name: source PDB name to migrate to cloud
  source_file_directory: file directory of source PDB data files
  target_file_directory: file directory of migrated PDB data files
```

```
Target Oracle CDB - define host specific variables for the target CDB
including some variables for on-prem CDB
  ansible_host: IP address of target database server host
  target_oracle_sid: target Oracle CDB instance ID
  target_pdb_name: target PDB name to be migrated to cloud (for max
availability option, the source and target PDB name must be the same)
  source_oracle_sid: source Oracle CDB instance ID
  source_pdb_name: source PDB name to be migrated to cloud
  source_port: source Oracle CDB listener port
  source_oracle_domain: source Oracle database domain name
  source_file_directory: file directory of source PDB data files
  target_file_directory: file directory of migrated PDB data files
```

数据库服务器主机文件配置

默认情况下、AWS EC2实例使用IP地址命名主机。如果您在hosts文件中对Ansv可 使用不同的名称、请在/etc/hosts文件中为源服务器和目标服务器设置主机命名解析。下面是一个示例。

```
127.0.0.1   localhost localhost.localdomain localhost4
localhost4.localdomain4
::1         localhost localhost.localdomain localhost6
localhost6.localdomain6
172.30.15.96 source_db_server
172.30.15.107 target_db_server
```

执行操作手册-按顺序执行

1. 安装Ands负责 控制器的前提条件。

```
ansible-playbook -i hosts requirements.yml
```

```
ansible-galaxy collection install -r collections/requirements.yml
--force
```

2. 对内部服务器执行迁移前任务—假设管理员是ssh用户、可使用sudo权限连接到内部Oracle主机。

```
ansible-playbook -i hosts ora_pdb_relocate.yml -u admin -k -K -t
ora_pdb_relo_onprem
```

3. 在AWS EC2实例中执行Oracle PDB从内置CDB到目标CDB的重新定位—假设EC2数据库实例连接为ec2-user、而使用EC2-user ssh密钥对执行db1.pm。

```
ansible-playbook -i hosts ora_pdb_relocate.yml -u ec2-user --private
-key db1.pem -t ora_pdb_relo_primary
```

从何处查找追加信息

要了解有关NetApp 解决方案 自动化的详细信息、请查看以下网站 "NetApp 解决方案自动化"

在AWS FSx ONTAP中实现自动化Oracle HA/DR

NetApp解决方案工程团队

目的

此工具包可利用FSx for ONTAP存储和EC2计算实例、为部署在AWS云中的Oracle数据库自动执行设置和管理高可用性和灾难恢复(HighAvailability and Disaster Recovery、HR/DR)环境的任务。

此解决方案 可解决以下使用情形：

- 设置HA/DR目标主机—内核配置、与源服务器主机匹配的Oracle配置。
- 设置FSx ONTAP—集群对等、Vserver对等、Oracle卷SnapMirror关系设置(从源到目标)。
- 通过Snapshot备份Oracle数据库数据—在crontab下执行
- 通过Snapshot备份Oracle数据库归档日志—在crontab下执行
- 在HA/DR主机上运行故障转移和恢复—测试和验证HA/DR环境
- 在故障转移测试后运行重新同步—在HA/DR模式下重新建立数据库卷SnapMirror关系

audience

此解决方案 适用于以下人员：

- 在AWS中设置Oracle数据库以实现高可用性、数据保护和灾难恢复的数据库开发人员。
- 对AWS云中的存储级别Oracle HA/DR解决方案感兴趣的数据库解决方案架构师。
- 负责管理支持Oracle数据库的AWS FSx ONTAP存储的存储管理员。
- 希望在AWS FSX/EC2环境中为HA/DR建立Oracle数据库的应用程序所有者。

许可证

访问、下载、安装或使用此GitHub存储库中的内容即表示您同意中列出的许可条款 "许可证文件"。

> 在使用此GitHub存储库中的内容制作和/或共享任何衍生作品方面存在一些限制。在使用内容之前、请确保您已阅读许可条款。如果您不同意所有条款、请勿访问、下载或使用此存储库中的内容。

解决方案 部署

部署的前提条件

部署需要满足以下前提条件。

```
Ansible v.2.10 and higher
ONTAP collection 21.19.1
Python 3
Python libraries:
  netapp-lib
  xmltodict
  jmespath
```

```
AWS FSx storage as is available
```

```
AWS EC2 Instance
  RHEL 7/8, Oracle Linux 7/8
  Network interfaces for NFS, public (internet) and optional management
  Existing Oracle environment on source, and the equivalent Linux
operating system at the target
```

下载此工具包

```
git clone https://github.com/NetApp/na_ora_hadr_failover_resync.git
```

全局变量配置

可变的AnsablePlaybooks驱动。其中包括一个示例全局变量文件FSX_vars_exple.yml、用于演示典型配置。以下是主要注意事项：

```
ONTAP - retrieve FSx storage parameters using AWS FSx console for both
source and target FSx clusters.
  cluster name: source/destination
  cluster management IP: source/destination
  inter-cluster IP: source/destination
  vserver name: source/destination
  vserver management IP: source/destination
  NFS lifs: source/destination
  cluster credentials: fsxadmin and vsadmin pwd to be updated in
roles/ontap_setup/defaults/main.yml file
```

```
Oracle database volumes - they should have been created from AWS FSx
console, volume naming should follow strictly with following standard:
  Oracle binary: {{ host_name }}_bin, generally one lun/volume
  Oracle data: {{ host_name }}_data, can be multiple luns/volume, add
additional line for each additional lun/volume in variable such as {{
host_name }}_data_01, {{ host_name }}_data_02 ...
  Oracle log: {{ host_name }}_log, can be multiple luns/volume, add
additional line for each additional lun/volume in variable such as {{
host_name }}_log_01, {{ host_name }}_log_02 ...
  host_name: as defined in hosts file in root directory, the code is
written to be specifically matched up with host name defined in host
file.
```

```
Linux and DB specific global variables - keep it as is.
  Enter redhat subscription if you have one, otherwise leave it black.
```

主机变量配置

主机变量在名为｛｛host_name｝｝.yml的host_vars目录中定义。其中包括一个示例主机变量文件host_name.yml、用于演示典型配置。以下是主要注意事项：

```
Oracle - define host specific variables when deploying Oracle in
multiple hosts concurrently
  ansible_host: IP address of database server host
  log_archive_mode: enable archive log archiving (true) or not (false)
  oracle_sid: Oracle instance identifier
  pdb: Oracle in a container configuration, name pdb_name string and
number of pdbs (Oracle allows 3 pdbs free of multitenant license fee)
  listener_port: Oracle listener port, default 1521
  memory_limit: set Oracle SGA size, normally up to 75% RAM
  host_datastores_nfs: combining of all Oracle volumes (binary, data,
and log) as defined in global vars file. If multi luns/volumes, keep
exactly the same number of luns/volumes in host_var file
```

```
Linux - define host specific variables at Linux level
  hugepages_nr: set hugepage for large DB with large SGA for
performance
  swap_blocks: add swap space to EC2 instance. If swap exist, it will
be ignored.
```

数据库服务器主机文件配置

默认情况下、AWS EC2实例使用IP地址命名主机。如果您在hosts文件中对Ansv可 使用不同的名称、请在/etc/hosts文件中为源服务器和目标服务器设置主机命名解析。下面是一个示例。

```
127.0.0.1   localhost localhost.localdomain localhost4
localhost4.localdomain4
::1         localhost localhost.localdomain localhost6
localhost6.localdomain6
172.30.15.96 db1
172.30.15.107 db2
```

执行操作手册-按顺序执行

1. 安装可操作控制器前提条件。

```
ansible-playbook -i hosts requirements.yml
```

```
ansible-galaxy collection install -r collections/requirements.yml
--force
```

2. 设置目标EC2数据库实例。

```
ansible-playbook -i hosts ora_dr_setup.yml -u ec2-user --private-key
db2.pem -e @vars/fsx_vars.yml
```

3. 在源数据库卷和目标数据库卷之间设置FSx ONTAP SnapMirror关系。

```
ansible-playbook -i hosts ontap_setup.yml -u ec2-user --private-key
db2.pem -e @vars/fsx_vars.yml
```

4. 通过Snapshot从crontab备份Oracle数据库数据卷。

```
10 * * * * cd /home/admin/na_ora_hadr_failover_resync &&
/usr/bin/ansible-playbook -i hosts ora_replication_cg.yml -u ec2-
user --private-key db1.pem -e @vars/fsx_vars.yml >>
logs/snap_data_`date +"%Y-%m%d-%H%M%S"`.log 2>&1
```

5. 通过Snapshot从crontab备份Oracle数据库归档日志卷。

```
0,20,30,40,50 * * * * cd /home/admin/na_ora_hadr_failover_resync &&
/usr/bin/ansible-playbook -i hosts ora_replication_logs.yml -u ec2-
user --private-key db1.pem -e @vars/fsx_vars.yml >>
logs/snap_log_`date +"%Y-%m%d-%H%M%S"`.log 2>&1
```

6. 在目标EC2数据库实例上运行故障转移并恢复Oracle数据库—测试和验证HA/DR配置。

```
ansible-playbook -i hosts ora_recovery.yml -u ec2-user --private-key
db2.pem -e @vars/fsx_vars.yml
```

7. 在故障转移测试后运行重新同步—在复制模式下重新建立数据库卷SnapMirror关系。

```
ansible-playbook -i hosts ontap_ora_resync.yml -u ec2-user --private
-key db2.pem -e @vars/fsx_vars.yml
```

从何处查找追加信息

要了解有关NetApp 解决方案 自动化的详细信息、请查看以下网站 "NetApp 解决方案自动化"

AWS FSx ONTAP集群和EC2实例配置

NetApp解决方案工程团队

目的

此工具包可自动执行AWS FSx ONTAP存储集群和EC2计算实例的配置任务、这些实例随后可用于数据库部署。

此解决方案 可解决以下使用情形：

- 在AWS云中的预定义VPC子网中配置EC2计算实例、并将用于EC2实例访问的ssh密钥设置为EC2-user。
- 在所需的可用性区域中配置AWS FSx ONTAP存储集群、配置存储SVM并设置集群管理员用户fsxadmin密码。

audience

此解决方案 适用于以下人员：

- 在AWS EC2环境中管理数据库的数据库管理员。
- 对AWS EC2生态系统中的数据库部署感兴趣的数据库解决方案架构师。
- 负责管理支持数据库的AWS FSx ONTAP存储的存储管理员。
- 喜欢在AWS EC2生态系统中建立数据库的应用程序所有者。

许可证

访问、下载、安装或使用此GitHub存储库中的内容即表示您同意中列出的许可条款 "许可证文件"。

> 在使用此GitHub存储库中的内容制作和/或共享任何衍生作品方面存在一些限制。在使用内容之前、请确保您已阅读许可条款。如果您不同意所有条款、请勿访问、下载或使用此存储库中的内容。

解决方案 部署

部署的前提条件

部署需要满足以下前提条件。

```
An Organization and AWS account has been setup in AWS public cloud
  An user to run the deployment has been created
  IAM roles has been configured
  IAM roles granted to user to permit provisioning the resources
```

```
VPC and security configuration
  A VPC has been created to host the resources to be provisioned
  A security group has been configured for the VPC
  A ssh key pair has been created for EC2 instance access
```

```
Network configuration
  Subnets has been created for VPC with network segments assigned
  Route tables and network ACL configured
  NAT gateways or internet gateways configured for internet access
```

下载此工具包

```
git clone https://github.com/NetApp/na_aws_fsx_ec2_deploy.git
```

连接和身份验证

该工具包应从AWS云Shell执行。AWS云Shell是一种基于浏览器的Shell、可用于轻松安全地管理、浏览AWS资源并与之进行交互。CloudShell会使用您的控制台凭据进行预身份验证。通用开发和运营工具已预先安装、因此无需在本地安装或配置。

Terraform提供程序.tf和main.tf文件配置

提供程序.tf定义了Terraform通过API调用配置资源的提供程序。main.tf定义了要配置的资源和资源的属性。下面是一些详细信息：

```
provider.tf:
  terraform {
    required_providers {
      aws = {
        source  = "hashicorp/aws"
        version = "~> 4.54.0"
      }
    }
  }
```

```
main.tf:
  resource "aws_instance" "ora_01" {
    ami                           = var.ami
    instance_type                 = var.instance_type
    subnet_id                     = var.subnet_id
    key_name                      = var.ssh_key_name
    root_block_device {
      volume_type                 = "gp3"
      volume_size                 = var.root_volume_size
    }
    tags = {
      Name                        = var.ec2_tag
    }
  }
  ....
```

Terraform variations.tf和terraform.tfvars配置

variables．tf声明了要在main.tf中使用的变量。terraform.tfvars包含变量的实际值。下面是一些示例：

```
variables.tf:
  ### EC2 instance variables ###
```

```
variable "ami" {
  type        = string
  description = "EC2 AMI image to be deployed"
}
```

```
variable "instance_type" {
  type        = string
  description = "EC2 instance type"
}
```

```

```

terraform.tfvars:
EC2 instance variables

```

```

ami = "ami-06640050dc3f556bb" //RedHat 8.6 AMI
instance_type = "t2.micro"
ec2_tag = "ora_01"
subnet_id = "subnet-04f5fe7073ff514fb"
ssh_key_name = "sufi_new"
root_volume_size = 30

```

```

逐步过程-按顺序执行

1. 在AWS云Shell中安装Terraform。

```
git clone https://github.com/tfutils/tfenv.git ~/.tfenv
```

```
mkdir ~/bin
```

```
ln -s ~/.tfenv/bin/* ~/bin/
```

```
tfenv install
```

```
tfenv use 1.3.9
```

2. 从NetApp GitHub公共站点下载该工具包

```
git clone https://github.com/NetApp-
Automation/na_aws_fsx_ec2_deploy.git
```

3. 运行init以初始化terraform

```
terraform init
```

4. 输出执行计划

```
terraform plan -out=main.plan
```

5. 应用执行计划

```
terraform apply "main.plan"
```

6. 完成后、运行销毁以删除资源

```
terraform destroy
```

从何处查找追加信息

要了解有关NetApp 解决方案 自动化的详细信息、请查看以下网站 "NetApp 解决方案自动化"