



資料庫自動化工具套件

NetApp Solutions

NetApp
April 12, 2024

目錄

資料庫自動化工具套件	1
自動化 Oracle 移轉	1
AWS FSX ONTAP 中的自動化 Oracle HA/DR	4
AWS FSX ONTAP 叢集和 EC2 執行個體資源配置	9

資料庫自動化工具套件

自動化 Oracle 移轉

NetApp 解決方案工程團隊

目的

此工具套件可將 Oracle 資料庫從內部部署移轉至 AWS 雲端、並將 FSX ONTAP 儲存設備和 EC2 運算執行個體作為目標基礎架構。假設客戶已在 CDB/PDB 模型中部署內部部署 Oracle 資料庫。此工具組可讓客戶使用 Oracle PDB 重新定位程序、並提供最大可用度選項、從 Oracle 主機上的容器資料庫重新定位命名的 PDB。這表示任何內部部署儲存陣列上的來源 PDB 都會重新定位至新的容器資料庫、而且服務中斷最少。Oracle 重新定位程序會在資料庫連線時移動 Oracle 資料檔案。當所有資料檔案移轉至 AWS 雲端時、它隨後會將使用者工作階段從內部部署重新路由至重新部署的資料庫服務。加底線的技術已獲證實是 Oracle PDB 熱複製方法。



雖然移轉工具組是在 AWS 雲端基礎架構上開發和驗證、但它是以 Oracle 應用程式層級解決方案為基礎。因此、此工具組適用於其他公有雲平台、例如 Azure、GCP 等

本解決方案可解決下列使用案例：

- 在內部部署來源 DB 伺服器上建立移轉使用者並授予必要權限。
- 將 PDB 從內部部署 CDB 重新放置到雲端中的目標 CDB、而來源 PDB 則保持在線上狀態、直到切換為止。

目標對象

本解決方案適用於下列人員：

- 將 Oracle 資料庫從內部預置移轉至 AWS 雲端的 DBA。
- 資料庫解決方案架構設計師、對 Oracle 資料庫從內部部署移轉至 AWS 雲端感興趣。
- 管理支援 Oracle 資料庫的 AWS FSX ONTAP 儲存設備的儲存管理員。
- 喜歡將 Oracle 資料庫從內部預置移轉至 AWS 雲端的應用程式擁有者。

授權

存取、下載、安裝或使用此 GitHub 儲存庫中的內容、即表示您同意中所列的授權條款 ["授權檔案"](#)。



對於與此 GitHub 儲存庫中的內容產生及 / 或共用任何衍生作品、有特定限制。使用內容前、請務必先閱讀授權條款。如果您不同意所有條款、請勿存取、下載或使用此儲存庫中的內容。

解決方案部署

部署的先決條件

部署需要下列先決條件。

```
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)
```

下載工具組

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

主機變數組態

主機變數是在主機 `_vars` 目錄中定義、名稱為 `{ {host_name} }` 的 `.yml` 檔。其中包含主機變數檔 `host_name.yml` 範例、以示範典型組態。以下是主要考量事項：

```
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
```

DB 伺服器主機檔案組態

AWS EC2 執行個體預設會使用 IP 位址來命名主機。如果您在主機檔案中使用不同的名稱來進行 Ansible，請在 `/etc/hosts` 檔案中為來源伺服器和目標伺服器設定主機命名解析。以下是範例。

```
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
```

教戰手冊執行 - 依序執行

1. 安裝 Ansible 控制器先決條件。

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

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

2. 在內部伺服器上執行移轉前工作 - 假設 admin 是 ssh 使用者、以使用 Sudo 權限連線至內部部署的 Oracle 主機。

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

3. 在 AWS EC2 執行個體中、執行 Oracle PDB 從內部部署 CDB 移轉至目標 CDB - 假設 EC2 DB 執行個體連線使用 EC2 使用者、以及使用 EC2 使用者 ssh 金鑰配對的 db1.pem。

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

何處可找到其他資訊

若要深入瞭解 NetApp 解決方案自動化、請參閱下列網站 ["NetApp 解決方案自動化"](#)

AWS FSX ONTAP 中的自動化 Oracle HA/DR

NetApp 解決方案工程團隊

目的

此工具套件可自動化設定及管理高可用度與災難恢復（HR/DR）環境的工作、以將部署於 AWS 雲端的 Oracle 資料庫與 ONTAP 儲存設備和 EC2 運算執行個體的 FSX 一起部署。

本解決方案可解決下列使用案例：

- 設定 HA/DR 目標主機：核心組態、Oracle 組態、以符合來源伺服器主機。
- 設定 FSX ONTAP：叢集對等關係、虛擬伺服器對等關係、Oracle Volume SnapMirror 關係設定、從來源到目標。
- 透過 Snapshot 備份 Oracle 資料庫資料 - 從 crontab 執行
- 透過 Snapshot 備份 Oracle 資料庫歸檔記錄 - 從 crontab 執行

- 在 HA/DR 主機上執行容錯移轉與還原 - 測試並驗證 HA/DR 環境
- 在容錯移轉測試後執行重新同步 - 在 HA/DR 模式中重新建立資料庫磁碟區 SnapMirror 關係

目標對象

本解決方案適用於下列人員：

- 在 AWS 中設定 Oracle 資料庫以實現高可用度、資料保護和災難恢復的 DBA。
- 對 AWS 雲端中的儲存層級 Oracle HA/DR 解決方案感興趣的資料庫解決方案架構設計師。
- 管理支援 Oracle 資料庫的 AWS FSX ONTAP 儲存設備的儲存管理員。
- 喜歡在 AWS FSS/EC2 環境中為 HA/DR 備份 Oracle 資料庫的應用程式擁有者。

授權

存取、下載、安裝或使用此 GitHub 儲存庫中的內容、即表示您同意中所列的授權條款 ["授權檔案"](#)。



對於與此 GitHub 儲存庫中的內容產生及 / 或共用任何衍生作品、有特定限制。使用內容前、請務必先閱讀授權條款。如果您不同意所有條款、請勿存取、下載或使用此儲存庫中的內容。

解決方案部署

部署的先決條件

部署需要下列先決條件。

```
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
```

下載工具組

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

整體變數組態

Ansible 教戰手冊是可變驅動的。其中包含範例通用變數檔案 `FSx_vars_example.yml`、以示範一般組態。以下是主要考量事項：

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.

主機變數組態

主機變數是在主機_vars 目錄中定義、名稱為 { {host_name} } 。 yml 。其中包含主機變數檔 host_name.yml 範例、以示範典型組態。以下是主要考量事項：

```
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.
```

DB 伺服器主機檔案組態

AWS EC2 執行個體預設會使用 IP 位址來命名主機。如果您在主機檔案中使用不同的名稱來進行 Ansible、請在 /etc/hosts 檔案中為來源伺服器和目標伺服器設定主機命名解析。以下是範例。

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

教戰手冊執行 - 依序執行

1. 安裝 Ansible 控制器預先安裝。

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

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

2. 設定目標 EC2 DB 執行個體。

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

3. 設定來源與目標資料庫磁碟區之間的 FSX ONTAP SnapMirror 關係。

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

4. 透過來自 crontab 的快照備份 Oracle 資料庫資料磁碟區。

```
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. 透過來自 crontab 的快照備份 Oracle 資料庫歸檔記錄磁碟區。

```
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. 在目標 EC2 DB 執行個體上執行容錯移轉並恢復 Oracle 資料庫 - 測試並驗證 HA/DR 組態。

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

7. 在容錯移轉測試後執行重新同步 - 在複寫模式中重新建立資料庫磁碟區 SnapMirror 關係。

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

何處可找到其他資訊

若要深入瞭解 NetApp 解決方案自動化、請參閱下列網站 "[NetApp 解決方案自動化](#)"

AWS FSX ONTAP 叢集和 EC2 執行個體資源配置

NetApp 解決方案工程團隊

目的

此工具套件可自動化 AWS FSX ONTAP 儲存叢集和 EC2 運算執行個體的資源配置工作、之後可用於資料庫部署。

本解決方案可解決下列使用案例：

- 在預先定義的 VPC 子網路中、在 AWS 雲端中佈建 EC2 運算執行個體、並將 EC2 執行個體存取的 ssh 金鑰設為 EC2 使用者。
- 在所需的可用性區域中佈建 AWS FSX ONTAP 儲存叢集、並設定儲存 SVM 並設定叢集管理使用者 fsxadmin 密碼。

目標對象

本解決方案適用於下列人員：

- 管理 AWS EC2 環境中資料庫的 DBA。
- 對 AWS EC2 生態系統中的資料庫部署感興趣的資料庫解決方案架構設計師。
- 管理支援資料庫的 AWS FSX ONTAP 儲存設備的儲存管理員。
- 喜歡在 AWS EC2 生態系統中執行資料刪除資料庫的應用程式擁有者。

授權

存取、下載、安裝或使用此 GitHub 儲存庫中的內容、即表示您同意中所列的授權條款 "[授權檔案](#)"。



對於與此 GitHub 儲存庫中的內容產生及 / 或共用任何衍生作品、有特定限制。使用內容前、請務必先閱讀授權條款。如果您不同意所有條款、請勿存取、下載或使用此儲存庫中的內容。

解決方案部署

部署的先決條件

部署需要下列先決條件。

```
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
```

下載工具組

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

連線與驗證

此工具組應從 AWS 雲端 Shell 執行。AWS 雲端 Shell 是瀏覽器型的 Shell、可讓您輕鬆安全地管理、探索及與 AWS 資源互動。CloudShell 已使用您的主控台認證預先驗證。一般的開發與作業工具已預先安裝、因此不需要進行本機安裝或組態。

Terraform provider .tf 和 main.tf 檔案組態

provider.tf 定義 Terraform 透過 API 呼叫來配置資源的供應商。main.tf 定義要配置的資源的資源和屬性。以下是一些詳細資料：

```
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 variables.tf 和 terraform.tfvars 組態

variables.tf 會宣告將用於 main.tf 的變數。terraform.tfvars 包含變數的實際值。以下是一些範例：

```
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
```

逐步程序 - 依序執行

1. 在 AWS 雲端 Shell 中安裝 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. 請從 NetApp GitHub 公用網站下載此工具套件

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

3. 執行初始化以初始化 terraform

```
terraform init
```

4. 輸出執行計畫

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

5. 套用執行計畫

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

6. 執行銷毀以移除完成後的資源

```
terraform destroy
```

何處可找到其他資訊

若要深入瞭解 NetApp 解決方案自動化、請參閱下列網站 ["NetApp解決方案自動化"](#)

版權資訊

Copyright © 2024 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

此軟體以 NETAPP「原樣」提供，不含任何明示或暗示的擔保，包括但不限於有關適售性或特定目的適用性之擔保，特此聲明。於任何情況下，就任何已造成或基於任何理論上責任之直接性、間接性、附隨性、特殊性、懲罰性或衍生性損害（包括但不限於替代商品或服務之採購；使用、資料或利潤上的損失；或企業營運中斷），無論是在使用此軟體時以任何方式所產生的契約、嚴格責任或侵權行為（包括疏忽或其他）等方面，NetApp 概不負責，即使已被告知有前述損害存在之可能性亦然。

NetApp 保留隨時變本文所述之任何產品的權利，恕不另行通知。NetApp 不承擔因使用本文所述之產品而產生的責任或義務，除非明確經過 NetApp 書面同意。使用或購買此產品並不會在依據任何專利權、商標權或任何其他 NetApp 智慧財產權的情況下轉讓授權。

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

有限權利說明：政府機關的使用、複製或公開揭露須受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中的「技術資料權利 - 非商業項目」條款 (b)(3) 小段所述之限制。

此處所含屬於商業產品和 / 或商業服務（如 FAR 2.101 所定義）的資料均為 NetApp, Inc. 所有。根據本協議提供的所有 NetApp 技術資料和電腦軟體皆屬於商業性質，並且完全由私人出資開發。美國政府對於該資料具有非專屬、非轉讓、非轉授權、全球性、有限且不可撤銷的使用權限，僅限於美國政府為傳輸此資料所訂合約所允許之範圍，並基於履行該合約之目的方可使用。除非本文另有規定，否則未經 NetApp Inc. 事前書面許可，不得逕行使用、揭露、重製、修改、履行或展示該資料。美國政府授予國防部之許可權利，僅適用於 DFARS 條款 252.227-7015(b)（2014 年 2 月）所述權利。

商標資訊

NETAPP、NETAPP 標誌及 <http://www.netapp.com/TM> 所列之標章均為 NetApp, Inc. 的商標。文中所涉及的所有其他公司或產品名稱，均為其各自所有者的商標，不得侵犯。