

How to: Restore DB from Snapshot

- [Overview](#)
- [Restore from Snapshot](#)
 - [Backup Window](#)
 - [Backup Retention Period](#)
 - [Pre-restore Steps](#)
 - [Restore from Snapshot Steps](#)
 - [1. Decide on the RDS DB instance prefix name to be used for restore](#)
 - [2. Identify the snapshot to restore the DB from](#)
 - [3. Take a manual snapshot of current RDS instance via console](#)
 - [4. Get the SSM Parameter Value of sql_alchemy_conn](#)
 - [5. Update the value of RDS identifier prefix and snapshot identifier in airflow code's tfvars](#)
 - [6. Clone the airflow repo and dev-utils repo onto the jumphost](#)
 - [7. Checkout your airflow feature or release branch on the jumphost](#)
 - [8. Setup your local terraform environment on the jumphost](#)
 - [9. Remove the db instance, db option group, db parameter group, db subnet group from tfstate file](#)
 - [10. Check the latest Terraform plan, which should show an RDS instance, subnet group, parameter group, and option group will be created](#)
 - [11. Run the airflow CICD pipeline with your feature or release branch](#)
 - [12. Verify that the SSM Parameter Value of sql_alchemy_conn is updated](#)
 - [13. Login to Airflow UI and verify that the data is restored to the desired state](#)
 - [14. Perform post verification checks and ensure a sample DAG runs successfully](#)
 - [15. Remove the snapshot_identifier in tfvars](#)
 - [16. Re-run the airflow CICD pipeline with your feature or release branch and perform post verification check](#)

Overview

A database instance can be created to restore from snapshot. These snapshots can be in the form of automated or manual snapshots. Automated snapshots are taken by AWS during the backup window to backup the entire DB instance.

Restoring the database instance can be done through AWS Console, CLI, terraform code, or a combination of those.

After restoring a database from snapshot, ensure that further deployments of Airflow through the CICD pipeline will continue to reference the newly restored database.

Restore from Snapshot

At a high-level, the steps to restore DB from snapshot involves:

- Creating a new database instance from a snapshot
- Updating the `/ato/sdpaap/<landing_zone>/<environment>/airflow/config/sql_alchemy_conn` SSM Parameter with the newly restored RDS endpoint value
- Restarting airflow-webserver and airflow-scheduler
- Ensuring the terraform state file is updated with the newly restored RDS details in order for subsequent Airflow CI pipeline deployments to use the correct RDS endpoint

Backup Window

Backup Retention Period

This is the number of days for which automatic backups are kept. Currently set to 30 days.

Connectivity & security | Monitoring | Logs & events | Configuration | **Maintenance & backups**

Tags

Maintenance

Auto minor version upgrade Enabled	Maintenance window wed:10:30-wed:13:30 UTC (GMT)	Pending maintenance none	Pending modifications
---------------------------------------	---	-----------------------------	-----------------------

Pending maintenance (0)

< 1 >

Description	Type	Status	Apply date
No pending maintenance available			

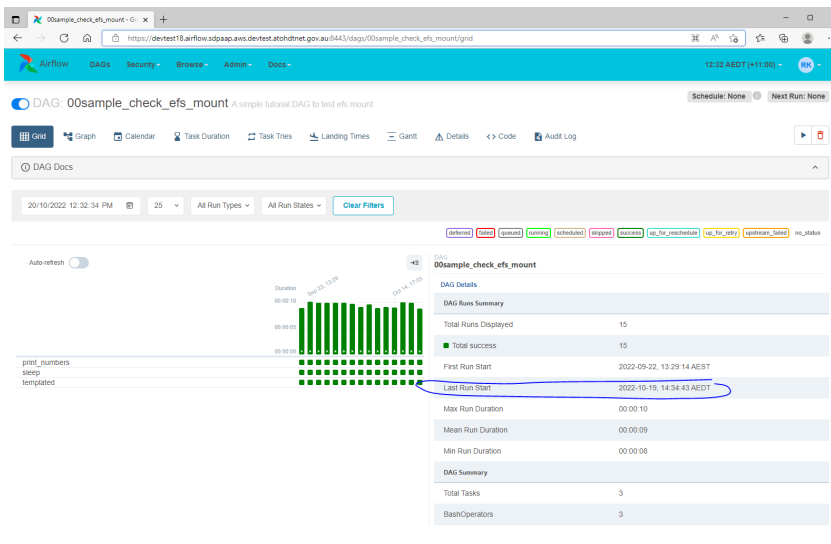
Backup

Automated backups Enabled (30 Days) Copy tags to snapshots Enabled Backup target AWS Cloud (Asia Pacific (Sydney))	Latest restore time September 12, 2022, 3:09:35 AM UTC Backup window 09:00-10:00 UTC (GMT)	Replicate to Region - Replicated automated backup -
--	---	--

Pre-restore Steps

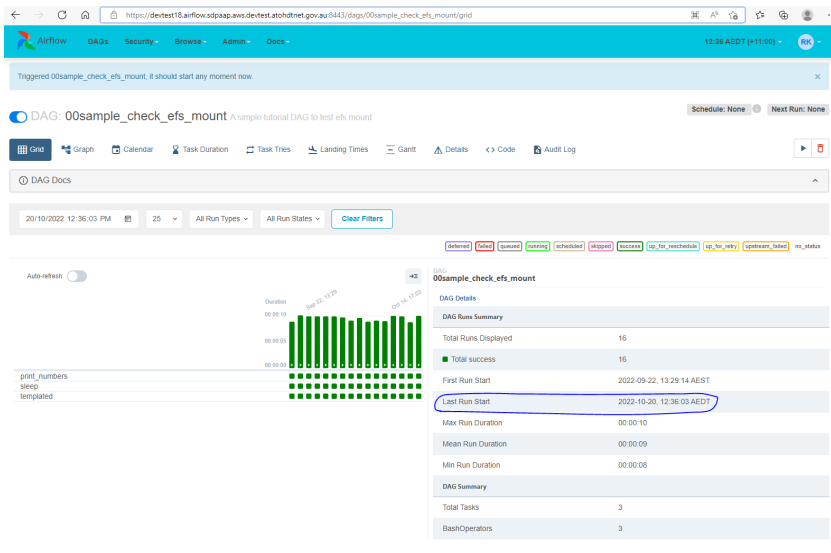
1. To demonstrate that the restore works, we'll run a sample DAG capture its current run state

- The DAG was last run on 2022-10-19, 14:34:43 AEDT
- Current datetime is 2022-10-20 12:30:00 AEDT



2. Run the sample DAG.

- Now, the DAG's last run is 2022-10-20, 12:36:03 AEDT



3. As a test, the database will be restored to the latest automated snapshot taken in the last 24 hours in the next section.

Amazon RDS

Snapshots

Manual Automated Shared with me Public Backup service Exports in Amazon S3

System snapshots (227)

Q devtest18 X

Snapshot name	DB instance or cluster	Snapshot creation time	DB instance created time	Status	Progress	Engine	VPC
rds-sdpaap-devtest18-airflow-rds-2022-10-19-09-04	sdpaap-devtest18-airflow-rds	October 19, 2022, 20:04 (UTC+11:00)	September 22, 2022, 11:25 (UTC+10:00)	Available	Completed	PostgreSQL	vpc-079bccc
rds-sdpaap-devtest18-airflow-rds-2022-10-19-09-04	sdpaap-devtest18-airflow-rds	October 18, 2022, 20:04 (UTC+11:00)	September 22, 2022, 11:25 (UTC+10:00)	Available	Completed	PostgreSQL	vpc-079bccc
rds-sdpaap-devtest18-airflow-rds-2022-10-17-09-04	sdpaap-devtest18-airflow-rds	October 17, 2022, 20:04 (UTC+11:00)	September 22, 2022, 11:25 (UTC+10:00)	Available	Completed	PostgreSQL	vpc-079bccc
rds-sdpaap-devtest18-airflow-rds-2022-10-16-09-04	sdpaap-devtest18-airflow-rds	October 16, 2022, 20:04 (UTC+11:00)	September 22, 2022, 11:25 (UTC+10:00)	Available	Completed	PostgreSQL	vpc-079bccc
rds-sdpaap-devtest18-airflow-rds-2022-10-15-09-05	sdpaap-devtest18-airflow-rds	October 15, 2022, 20:05 (UTC+11:00)	September 22, 2022, 11:25 (UTC+10:00)	Available	Completed	PostgreSQL	vpc-079bccc
rds-sdpaap-devtest18-airflow-rds-2022-10-14-09-05	sdpaap-devtest18-airflow-rds	October 14, 2022, 20:05 (UTC+11:00)	September 22, 2022, 11:25 (UTC+10:00)	Available	Completed	PostgreSQL	vpc-079bccc
rds-sdpaap-devtest18-airflow-rds-2022-10-13-09-04	sdpaap-devtest18-airflow-rds	October 13, 2022, 20:04 (UTC+11:00)	September 22, 2022, 11:25 (UTC+10:00)	Available	Completed	PostgreSQL	vpc-079bccc
rds-sdpaap-devtest18-airflow-rds-2022-10-12-09-04	sdpaap-devtest18-airflow-rds	October 12, 2022, 20:04 (UTC+11:00)	September 22, 2022, 11:25 (UTC+10:00)	Available	Completed	PostgreSQL	vpc-079bccc
rds-sdpaap-devtest18-airflow-rds-2022-10-11-09-04	sdpaap-devtest18-airflow-rds	October 11, 2022, 20:04 (UTC+11:00)	September 22, 2022, 11:25 (UTC+10:00)	Available	Completed	PostgreSQL	vpc-079bccc

4. The expected result of the restore from snapshot will be to not have the last DAG run in step 2, and that the last DAG run is as per Step 1 as there was no other DAG run post the snapshot creation time.

Restore from Snapshot Steps

1. Decide on the RDS DB instance prefix name to be used for restore

- The default prefix for RDS is "sdpaap".
- Since restore from snapshot requires a new DB instance to be created and we do not want to delete the existing DB instance, we need to specify a new prefix for the RDS name.
- Example new prefix = "sdpaap-restore"

2. Identify the snapshot to restore the DB from

The latest automated snapshot will be used as an example for this restoration

System snapshots (227)

Search: devtest18

<input type="checkbox"/>	Snapshot name	DB instance or cluster	Snapshot creation time	DB instance created time	Status	Progress	Engine	VPC
<input type="checkbox"/>	rdssdpaap-devtest18-airflow-rds-2022-10-19-09-04	sdpaap-devtest18-airflow-rds	October 19, 2022, 20:04 (UTC+11:00)	September 22, 2022, 11:25 (UTC+10:00)	Available	Completed	PostgreSQL	vpc-079bccc8f26c24e
<input type="checkbox"/>	rdssdpaap-devtest18-airflow-rds-2022-10-18-09-04	sdpaap-devtest18-airflow-rds	October 18, 2022, 20:04 (UTC+11:00)	September 22, 2022, 11:25 (UTC+10:00)	Available	Completed	PostgreSQL	vpc-079bccc8f26c24e
<input type="checkbox"/>	rdssdpaap-devtest18-airflow-rds-2022-10-17-09-04	sdpaap-devtest18-airflow-rds	October 17, 2022, 20:04 (UTC+11:00)	September 22, 2022, 11:25 (UTC+10:00)	Available	Completed	PostgreSQL	vpc-079bccc8f26c24e
<input type="checkbox"/>	rdssdpaap-devtest18-airflow-rds-2022-10-16-09-04	sdpaap-devtest18-airflow-rds	October 16, 2022, 20:04 (UTC+11:00)	September 22, 2022, 11:25 (UTC+10:00)	Available	Completed	PostgreSQL	vpc-079bccc8f26c24e
<input type="checkbox"/>	rdssdpaap-devtest18-airflow-rds-2022-10-15-09-05	sdpaap-devtest18-airflow-rds	October 15, 2022, 20:05 (UTC+11:00)	September 22, 2022, 11:25 (UTC+10:00)	Available	Completed	PostgreSQL	vpc-079bccc8f26c24e
<input type="checkbox"/>	rdssdpaap-devtest18-airflow-rds-2022-10-14-09-05	sdpaap-devtest18-airflow-rds	October 14, 2022, 20:05 (UTC+11:00)	September 22, 2022, 11:25 (UTC+10:00)	Available	Completed	PostgreSQL	vpc-079bccc8f26c24e
<input type="checkbox"/>	rdssdpaap-devtest18-airflow-rds-2022-10-13-09-04	sdpaap-devtest18-airflow-rds	October 13, 2022, 20:04 (UTC+11:00)	September 22, 2022, 11:25 (UTC+10:00)	Available	Completed	PostgreSQL	vpc-079bccc8f26c24e
<input type="checkbox"/>	rdssdpaap-devtest18-airflow-rds-2022-10-12-09-04	sdpaap-devtest18-airflow-rds	October 12, 2022, 20:04 (UTC+11:00)	September 22, 2022, 11:25 (UTC+10:00)	Available	Completed	PostgreSQL	vpc-079bccc8f26c24e
<input type="checkbox"/>	rdssdpaap-devtest18-airflow-rds-2022-10-11-09-04	sdpaap-devtest18-airflow-rds	October 11, 2022, 20:04 (UTC+11:00)	September 22, 2022, 11:25 (UTC+10:00)	Available	Completed	PostgreSQL	vpc-079bccc8f26c24e

3. Take a manual snapshot of current RDS instance via console

sdpaap-devtest18-airflow-rds

Summary

DB identifier sdpaap-devtest18-airflow-rds	CPU 6.65%	Status Available	Class db.t4g.small
Role Instance	Current activity 13 Connections	Engine PostgreSQL	Region & AZ ap-southeast-2

Actions

- Modify
- Stop temporarily
- Reboot
- Delete
- Create read replica
- Create Aurora read replica
- Promote
- Take snapshot**
- Restore to point in time
- Migrate snapshot

Take DB snapshot

Settings

To take a snapshot of this DB instance you must provide a name for the snapshot.

DB instance
The unique key that identifies a DB instance. This parameter isn't case-sensitive.

sdpaap-devtest18-airflow-rds

Snapshot name
The identifier for the DB snapshot.

sdpaap-devtest18-airflow-rds-20221020124500

Cancel Take snapshot

4. Get the SSM Parameter Value of sql_alchemy_conn

- Retrieve the current RDS endpoint used by Airflow in AWS console or on the jumphost by running the CLI command below. This will be used at a later step to verify that the database connection endpoint has been changed in Airflow.

```
aws ssm get-parameter --name /ato/sdpaap/<landing_zone>/<environment>/airflow/config/sql_alchemy_conn --with-decryption | jq '.Parameter.Value'
```

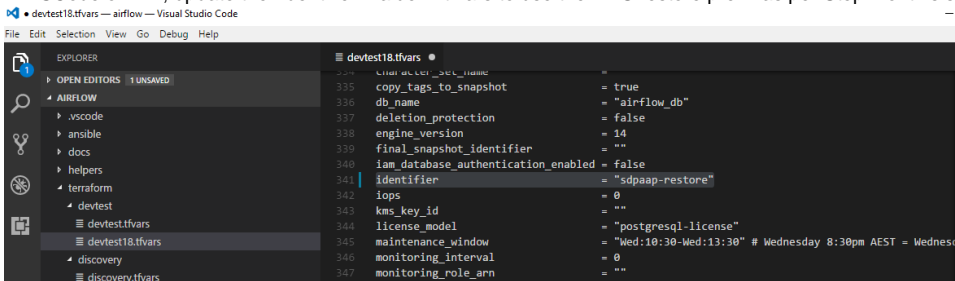
E.g.: `aws ssm get-parameter --name /ato/sdpaap/devtest/devtest18/airflow/config/sql_alchemy_conn --with-decryption | jq '.Parameter.Value'`

- Note down the value of sql_alchemy_conn SSM Param

```
10.204.76.128 - PuTTY
[ado22@vm ~]$ aws ssm get-parameter --name /ato/sdpaap/devtest/devtest18/airflow/config/sql_alchemy_conn --with-decryption | jq '.Parameter.Value'
{"Parameter.Value": "postgres://airflow_user:password@sdpaap-devtest18-airflow-rds.cvham1st2rtg.ap-southeast-2.rds.amazonaws.com:8432/airflow_db"}
[ado22@vm ~]$
```

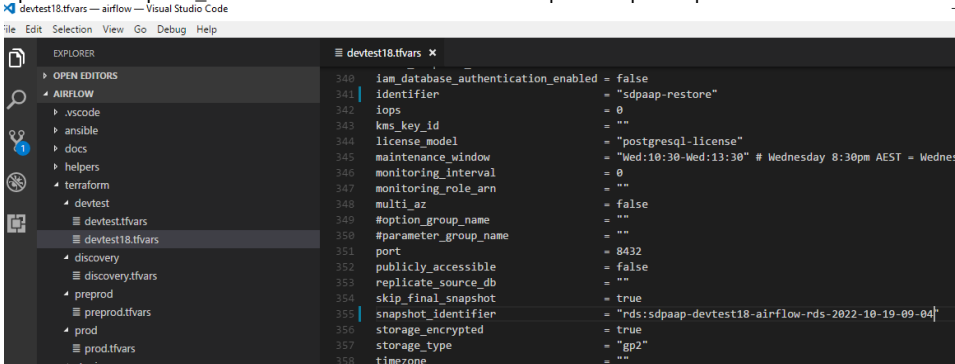
5. Update the value of RDS identifier prefix and snapshot identifier in airflow code's tfvars

- Create a new feature or release branch in airflow repo
- In VSCode or IDE, update the "identifier" value in tfvars to use the RDS restore prefix as per Step 1 of this section



```
devtest18.tfvars
335 character_set_name      = ""
336 copy_tags_to_snapshot   = true
337 db_name                 = "airflow_db"
338 deletion_protection      = false
339 engine_version          = 14
340 final_snapshot_identifier = ""
341 iam_database_authentication_enabled = false
342 identifier              = "sdpaap-restore"
343 iops                    = 0
344 kms_key_id              = ""
345 license_model           = "postgresql-license"
346 maintenance_window      = "Wed:10:30-Wed:13:30" # Wednesday 8:30pm AEST = Wednesday 1:30pm UTC
347 monitoring_interval     = 0
348 monitoring_role_arn     = ""
```

- Update the "snapshot_identifier" value in tfvars to use the snapshot as per Step 2 of this section



```
devtest18.tfvars
340 iam_database_authentication_enabled = false
341 identifier              = "sdpaap-restore"
342 iops                    = 0
343 kms_key_id              = ""
344 license_model           = "postgresql-license"
345 maintenance_window      = "Wed:10:30-Wed:13:30" # Wednesday 8:30pm AEST = Wednesday 1:30pm UTC
346 monitoring_interval     = 0
347 monitoring_role_arn     = ""
348 multi_az                = false
349 #option_group_name       = ""
350 #parameter_group_name   = ""
351 port                    = 8432
352 publicly_accessible     = false
353 replicate_source_db     = ""
354 skip_final_snapshot      = true
355 snapshot_identifier      = "rds:sdpaap-devtest18-airflow-rds-2022-10-19-09-04"
356 storage_encrypted       = true
357 storage_type            = "gp2"
358 timezone                = ""
```

- Save your changes. Commit and push your changes to remote

6. Clone the airflow repo and dev-utils repo onto the jumphost

- Back to the jumphost as your "a" account
- Ensure your own ecdsa ssh keys are setup on the jumphost and the public ecdsa key is uploaded to your Gitlab profile
- Clone the repos

```
# in your home directory on the jumpbox
mkdir ~/git
cd ~/git
git clone git@gitlab.sdpaap.aws.prod.atohnet.gov.au:sdpaap/platform/airflow.git
git clone git@gitlab.sdpaap.aws.prod.atohnet.gov.au:sdpaap/ops-utils/dev-utils.git
```

7. Checkout your airflow feature or release branch on the jumphost

- Still on the jumphost, checkout your feature branch and ensure you have the tfvars for your environment

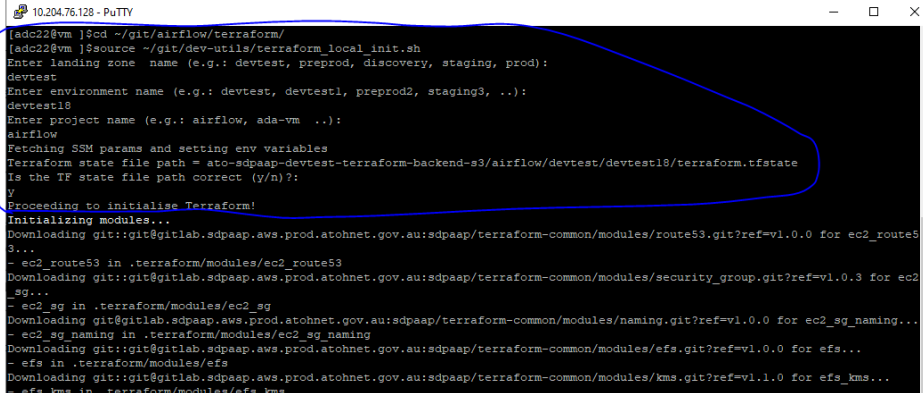
```
cd ~/git/airflow
git checkout <feature or release branch>
ls -l terraform/<landing_zone>/<environment>.tfvars
```

8. Setup your local terraform environment on the jumphost

- The commands in this step are based on the instructions to setup a local terraform environment as described in the [dev-utils README](#).
- On the jumphost

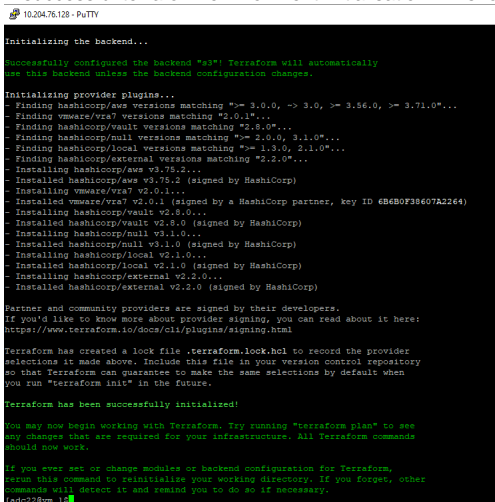
```
cd ~/git/airflow/terraform/
source ~/git/dev-utils/terraform_local_init.sh
```

- After running the "source" command, you should see the terraform modules being initialised



```
10.204.76.128 - PuTTY
[adc22@vm] $ cd ~/git/airflow/terraform/
[adc22@vm] $ source ~/git/dev-utils/terraform_local_init.sh
Enter landing zone name (e.g.: devtest, preprod, discovery, staging, prod):
devtest
Enter environment name (e.g.: devtest, devtest1, preprod2, staging3, ..):
devtest18
Enter project name (e.g.: airflow, ada-vm ..):
airflow
Fetching SSM params and setting env variables
Terraform state file path = ato-sdpaap-devtest-terraform-backend-s3/airflow/devtest/devtest18/terraform.tfstate
Is the TF state file path correct (y/n)?
y
Proceeding to initialise Terraform!
Initializing modules...
Downloading git::git@github.com:aws-prod-athernet-gov.au:sdpaap/terraform-common/modules/route53.git?ref=v1.0.0 for ec2_route53...
- ec2_route53 in .terraform/modules/ec2_route53
Downloading git::git@github.com:aws-prod-athernet-gov.au:sdpaap/terraform-common/modules/security_group.git?ref=v1.0.3 for ec2_sg...
- ec2_sg in .terraform/modules/ec2_sg
Downloading git::git@github.com:aws-prod-athernet-gov.au:sdpaap/terraform-common/modules/naming.git?ref=v1.0.0 for ec2_sg_naming...
- ec2_sg_naming in .terraform/modules/ec2_sg_naming
Downloading git::git@github.com:aws-prod-athernet-gov.au:sdpaap/terraform-common/modules/efs.git?ref=v1.0.0 for efs...
- efs in .terraform/modules/efs
Downloading git::git@github.com:aws-prod-athernet-gov.au:sdpaap/terraform-common/modules/kms.git?ref=v1.1.0 for efs_kms...
- efs_kms in .terraform/modules/efs_kms
```

- A successful terraform environment initialisation will show these logs



```
10.204.76.128 - PuTTY
Initializing the backend...
Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.

Initializing provider plugins...
- Finding hashicorp/aws versions matching ">= 3.0.0, < 3.0, >= 3.56.0, < 3.71.0"...
- Finding vmware/vra? versions matching "2.0.1"...
- Finding hashicorp/vault versions matching "2.8.0"...
- Finding hashicorp/mull versions matching ">= 2.0.0, < 3.1.0"...
- Finding hashicorp/local versions matching ">= 1.3.0, < 2.1.0"...
- Finding hashicorp/external versions matching "2.2.0"...
- Installing hashicorp/aws v3.75.2 (signed by HashiCorp)
- Installing vmware/vra? v2.0.1 (signed by a HashiCorp partner, key ID 6B6B0F3B607A2264)
- Installing hashicorp/vault v2.8.0...
- Installing hashicorp/mull v3.1.0...
- Installing hashicorp/local v2.1.0...
- Installing hashicorp/external v2.2.0...
- Installing hashicorp/external v2.2.0 (signed by HashiCorp)

Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it here:
https://www.terraform.io/docs/cli/plugins/signing.html

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
run this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.

[adc22@vm] $
```

9. Remove the db instance, db option group, db parameter group, db subnet group from tfstate file

- This is so terraform won't destroy them when you run the CI/CD pipeline as they are referencing the "old" database resources
- Run these commands on the jumphost where the local terraform environment was initialised as in the previous step

```
terraform state rm module.rds.aws_db_instance.this
terraform state rm module.rds.aws_db_option_group.group[0]
terraform state rm module.rds.aws_db_parameter_group.parameter_group[0]
terraform state rm module.rds.aws_db_subnet_group.generic_group[0]
```

10. Check the latest Terraform plan, which should show an RDS instance, subnet group, parameter group, and option group will be created

- Still on the jumphost with local terraform environment

```
terraform plan -var-file <landing_zone>/<environment>.tfvars

E.g.: terraform plan -var-file devtest/devtest18.tfvars
```

- A new option group, parameter group, subnet group will be created using the new prefix naming convention

```
# module.rds.aws_db_option_group.group[0] will be created
+ resource "aws_db_option_group" "group" {
+   arn = (known after apply)
+   engine_name = "postgres"
+   id = (known after apply)
+   major_engine_version = "14"
+   name = "sdpaap-restore-devtest18-airflow-rds-option-group-postgres14"
+   name_prefix = (known after apply)
+   option_group_description = "Airflow RDS Option Group"
+   tags_all = (known after apply)
+ }

# module.rds.aws_db_parameter_group.parameter_group[0] will be created
+ resource "aws_db_parameter_group" "parameter_group" {
+   arn = (known after apply)
+   description = "Managed by Terraform"
+   family = "postgres14"
+   id = (known after apply)
+   name = "sdpaap-restore-devtest18-airflow-rds-parameter-group-postgres14"
+   name_prefix = (known after apply)
+   tags_all = (known after apply)
+ }

# module.rds.aws_db_subnet_group.generic_group[0] will be created
+ resource "aws_db_subnet_group" "generic_group" {
+   arn = (known after apply)
+   description = "Managed by Terraform"
+   id = (known after apply)
+   name = "sdpaap-restore-devtest18-airflow-rds-subnet-group"
+   name_prefix = (known after apply)
+   subnet_ids = [
+     "subnet-032d3665eea6f2bed",
+     "subnet-09bfe26ae2ce4c40a",
+     "subnet-0b2cd0025b6172122",
+   ]
+   tags = {
+     "SDPAAP_AppID" = "airflow"
+     "SDPAAP_AppName" = "airflow"
+     "SDPAAP_CostCode" = "4318"
+     "SDPAAP_Environment" = "devtest18"
+     "SDPAAP_LandingZone" = "devtest"
+     "SDPAAP_Name" = "sdpaap-restore-devtest18-airflow-rds"
+     "SDPAAP_Owner" = "aaptocloud@ato.gov.au"
+   }
+   tags_all = {
+     "SDPAAP_AppID" = "airflow"
+     "SDPAAP_AppName" = "airflow"
+     "SDPAAP_CostCode" = "4318"
+     "SDPAAP_Environment" = "devtest18"
+     "SDPAAP_LandingZone" = "devtest"
+     "SDPAAP_Name" = "sdpaap-restore-devtest18-airflow-rds"
+     "SDPAAP_Owner" = "aaptocloud@ato.gov.au"
+   }
+ }
}
```

- A new RDS instance will be created using the new prefix naming convention. This will use the option group, parameter group, subnet group as per the naming convention. And will be created based from the defined snapshot_identifier.

```
# module:rds.aws_db_instance.this will be created
+ resource "aws_db_instance" "this" {
  + address                         = (known after apply)
  + allocated_storage               = 10
  + allow_major_version_upgrade    = false
  + apply_immediately              = true
  + arn                            = (known after apply)
  + auto_minor_version_upgrade     = true
  + availability_zone              = (known after apply)
  + backup_retention_period        = 30
  + backup_window                  = "09:00-10:00"
  + ca_cert_identifier             = (known after apply)
  + character_set_name              = (known after apply)
  + copy_tags_to_snapshot         = true
  + customer_owned_ip_enabled      = false
  + db_subnet_group_name           = "sdpaap-restore-devtest18-airflow-rds-subnet-group"
  + delete_automated_backups       = true
  + deletion_protection            = false
  + enabled_cloudwatch_logs_exports = [
    + "postgresql",
  ]
  + endpoint                      = (known after apply)
  + engine                        = "postgres"
  + engine_version                 = "14"
  + engine_version_actual          = (known after apply)
  + final_snapshot_identifier      = "sdpaap-restore-devtest18-airflow-rds-final-snapshot"
  + hosted_zone_id                = (known after apply)
  + iam_database_authentication_enabled = false
  + id                             = (known after apply)
  + identifier                     = "sdpaap-restore-devtest18-airflow-rds"
  + identifier_prefix              = (known after apply)
  + instance_class                 = "db.t4g.small"
  + logs                           = 0
  + kms_key_id                     = "arn:aws:kms:ap-southeast-2:875250343506:key/a8450fab-c251-4704-b91b-af37eb46b3a5"
  + latest_restorable_time         = (known after apply)
  + license_model                  = "postgresql-license"
  + maintenance_window             = "wed:10:30-wed:13:30"
  + max_allocated_storage          = 50
  + monitoring_interval            = 0
  + monitoring_role_arn            = (known after apply)
  + multi_az                       = false
  + name                           = "airflow_db"
  + nchar_character_set_name       = (known after apply)
  + option_group_name              = "sdpaap-restore-devtest18-airflow-rds-option-group-postgres14"
  + parameter_group_name           = "sdpaap-restore-devtest18-airflow-rds-parameter-group-postgres14"
  + password                       = (sensitive value)
  + performance_insights_enabled   = false
  + performance_insights_kms_key_id = (known after apply)
  + performance_insights_retention_period = (known after apply)
  + port                           = 5432
  + publicly_accessible            = false
  + replicas                       = (known after apply)
  + replicate_source_db            = ""
  + resource_id                    = (known after apply)
  + skip_final_snapshot            = true
  + snapshot_identifier            = "rds:sdpaap-devtest18-airflow-rds-2022-10-19-09-04"
  + status                         = (known after apply)
  + storage_encrypted              = true
  + storage_type                   = "gp2"
  + tags                           = {
    + "SDPAAP_AppID"      = "airflow"
    + "SDPAAP_AppName"    = "airflow"
    + "SDPAAP_CostCode"   = "4318"
    + "SDPAAP_Environment" = "devtest18"
    + "SDPAAP_LandingZone" = "devtest"
    + "SDPAAP_Name"       = "sdpaap-restore-devtest18-airflow-rds"
    + "SDPAAP_Owner"      = "aaptocloud@ato.gov.au"
  }
  + tags_all = {
    + "SDPAAP_AppID"      = "airflow"
    + "SDPAAP_AppName"    = "airflow"
    + "SDPAAP_CostCode"   = "4318"
    + "SDPAAP_Environment" = "devtest18"
    + "SDPAAP_LandingZone" = "devtest"
    + "SDPAAP_Name"       = "sdpaap-restore-devtest18-airflow-rds"
    + "SDPAAP_Owner"      = "aaptocloud@ato.gov.au"
  }
}
```

11. Run the [airflow CICD pipeline](#) with your feature or release branch

- New subnet group, option group and parameter group should be created
- The RDS restore instance should be using the newly created subnet group, option group and parameter group

12. Verify that the SSM Parameter Value of `sql_alchemy_conn` is updated

- Run the `get-parameter` cli command or check in AWS console to verify. The `sql_alchemy_conn` SSM Parameter should have the value of the newly created RDS restore endpoint

AWS Systems Manager > Parameter Store > /ato/sdpaap/devtest/devtest18/airflow/config/sql_alchemy_conn > Overview

/ato/sdpaap/devtest/devtest18/airflow/config/sql_alchemy_conn

Overview | History | Tags

Name	/ato/sdpaap/devtest/devtest18/airflow/config/sql_alchemy_conn
Tier	Standard
Type	SecureString
Last modified date	Thu, 20 Oct 2022 02:46:35 GMT
Value	postgresql+psycopg2://airflow_user:psycopg2@sdpaap-restore-devtest18-airflow-rds.cvsm1st2rtg.ap-southeast-2-rds.amazonaws.com:5432/airflow_db
Description	-
Data type	text
Last modified user	arn:aws:sts:875250343506:assumed-role/ato-role-instance-sdpaap-deployer/i-09070783f2ed190f8
Version	25

13. Login to Airflow UI and verify that the data is restored to the desired state

- For testing purposes, I have verified that the sample DAG in Airflow UI has the "Last Run Start" value as shown in Step 1 of Pre-restore Steps
- We have now successfully restored the database from a snapshot.

14. Perform post verification checks and ensure a sample DAG runs successfully

15. Remove the snapshot_idenfier in tfvars

- Although keeping the snapshot_identifier as is will not impact further deployments because RDS already exists, we will change the value back to an empty string in order for an empty DB to be created and not from a snapshot in the event that a new RDS will be created.
- Update the "snapshot_identifier" value in tfvars to an empty string (its original value)

devtest18.tfvars — airflow — Visual Studio Code

File Edit Selection View Go Debug Help

```

350 #parameter_group_name = ""
351 port = 8432
352 publicly_accessible = false
353 replicate_source_db = ""
354 skip_final_snapshot = true
355 snapshot_identifier = ""
356 storage_encrypted = true
357 storage_type = "gp2"
358 timezone = ""
359 #vpc_security_group_ids = ""
360 #db_subnet_group_name = ""
361 group_subnet_ids = [""]
362 option_group_description = "Airflow RDS Option Group"
363 major_engine_version = 14
364 family = "postgres14"
365 create_db_subnet_group = true
366 create_db_option_group = true
367 create_db_parameter_group = true

```

- Save your changes. Commit and push your changes to remote

16. Re-run the [airflow CICD pipeline](#) with your feature or release branch and perform post verification check

- The pipeline should run successfully with no resource and config changes
- Running a sample DAG should execute successfully