Disaster Recovery Systems for RDS and S3

Timotej Petrovčič
Devops Engineer @ Easy.bi

AWS UG meetup - May 2025

Agenda

1. Technical requirements
2. AWS Services evaluation
3. Terraform module design
4. S3 backups
    a. Setup
    b. Security / IAM
    c. Restore testing
5. S3 restoration
    a. Setup / Caveats
6. RDS backups
    a. Setup
    b. KMS
    c. Security / IAM
    d. Restore testing
7. RDS restoration
    a. Setup / Versions / Caveats

Technical requirements

- RDS and S3 backups
  - PITR
  - Cross-account (CA)
  - Cross-region (CR)
- Easy configuration of backups
  - Only specify what is needed
- Modular/transferable setup
  - Only create what is needed
  - Limit within:
    - **2 accounts + 2 regions**

- Regularly tested/working restoration processes
- Least privilege access for various resources
- Backups should be fully managed by AWS
- Keep the costs within reason

AWS Services evaluation

- RDS 
  - Single instance database
  - Has internal backups
    - No CA support
- S3 
  - CloudFront bucket
  - Versioning + Object Lock

- Backup 
  - Unified dashboard
  - Schedule based backups
  - CA/CR backup support
  - Restoration testing
  - Managed service

Terraform module design

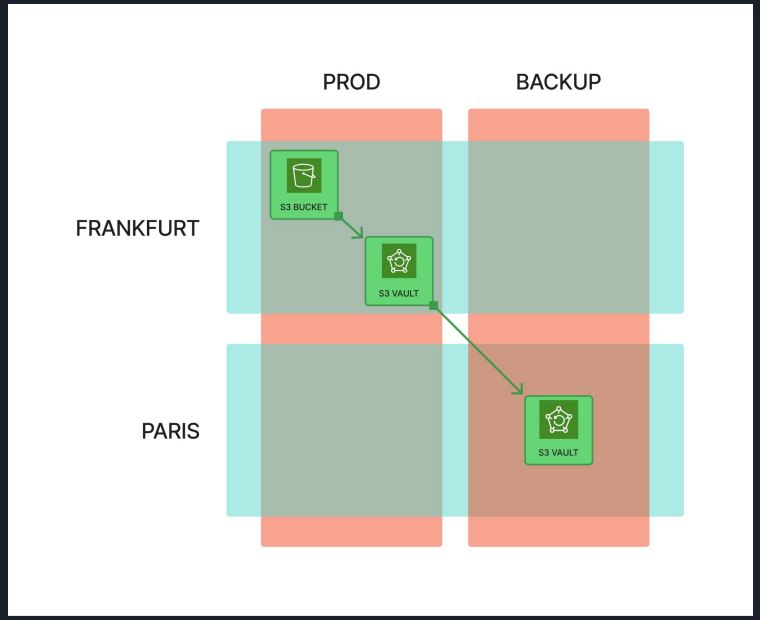
- Version 1 - Regular backups:
  - Module for **RDS + Backup**
  - Module for **S3 + Backup**
- Version 2 - CA/CR backups:
  - Module for **RDS + Backup**
  - Module for **S3 + Backup**
  - Module for **CA/CR Backup**
- Version 3 - CA/CR backups:
  - Module for **RDS**
  - Module for **S3**
  - Module for **RDS Backup**
  - Module for **S3 Backup**
  - Module for **CA/CR Backup**

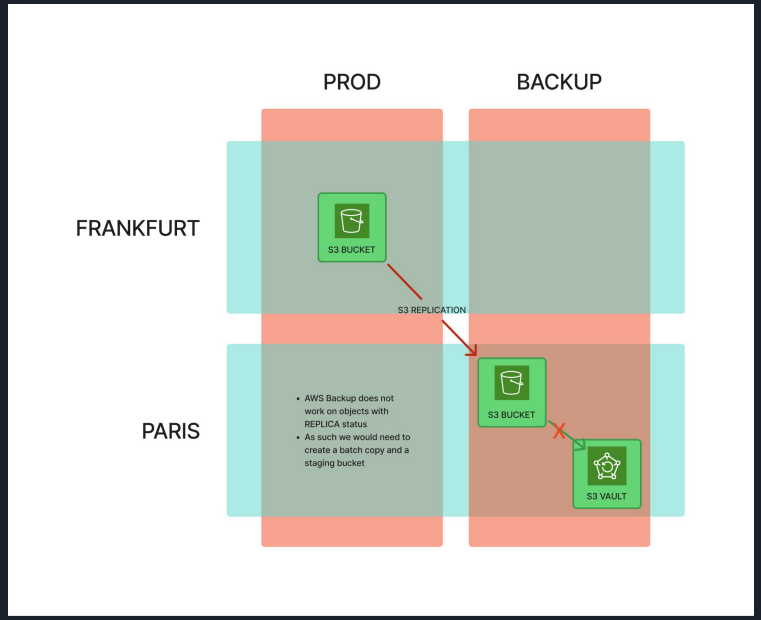
Golden rule:

- 1 service == 1 module*

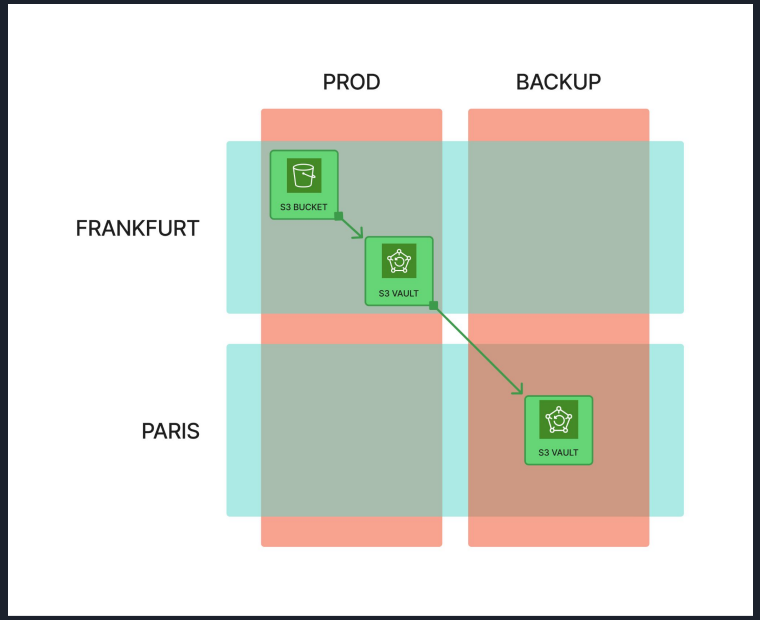
S3

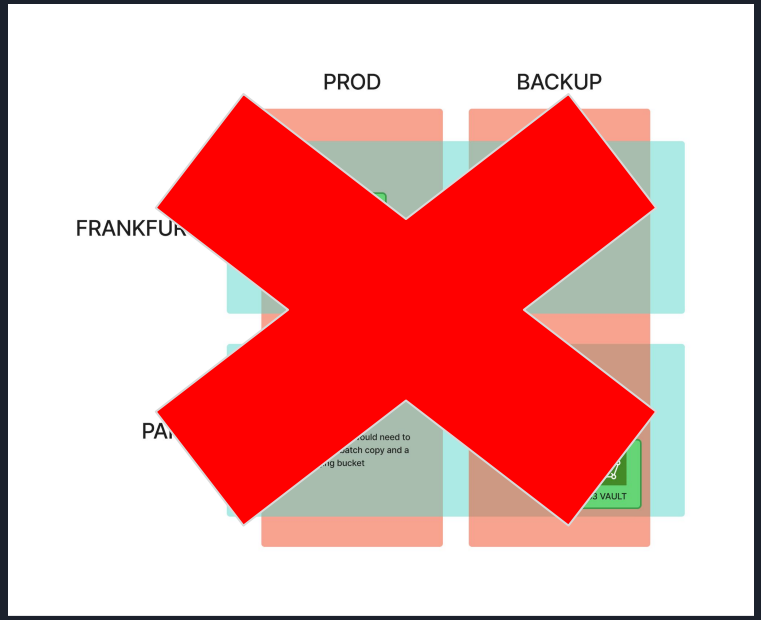
S3 Backups - Setup



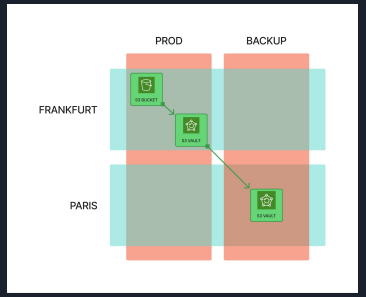


S3 Backups - Setup





S3 Backups - Security



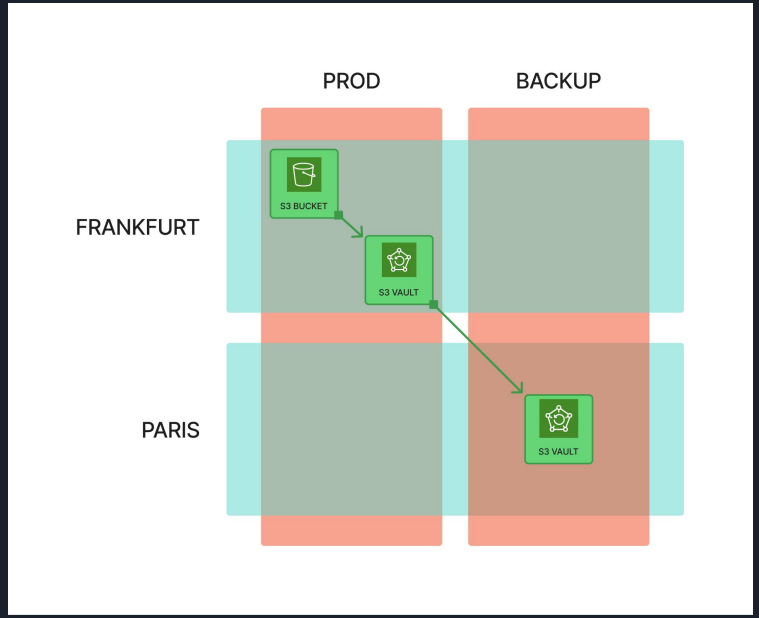
- Create 3 policies
  - Regular, CAR, Restore
- Start with the AWS managed policy and scope it down
- Allow access to (Regular):
  - S3 KMS key (Optional)
  - Source Backup Vault KMS key
  - Source Backup Vault (Describe, Tag)
  - EventBridge (S3 notifications)
  - S3 buckets (List)
  - S3 Objects (Get,)

Additionally allow access to: (CAR):

- Destination Backup Vault KMS key
- Backup resource tagging
- Source/Destination Backup Vault
  - *CopyIntoBackupVault*
- Source/Destination Backup Snapshots
  - *CopyFromBackupVault*
  - Regex generated snapshot ARN format

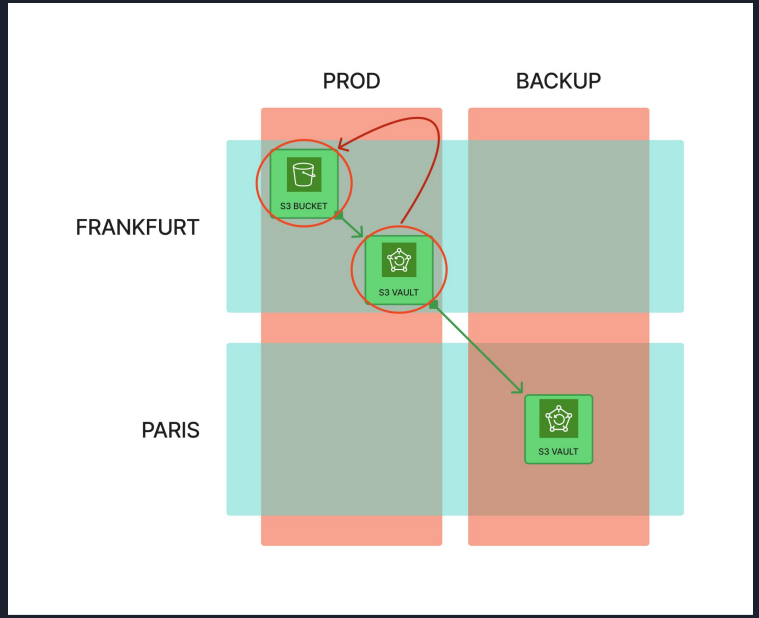
S3 Backups - Restore testing

- Regular restore testing choosing a random/specific snapshot
- Limit to Source Backup vault testing
- Start with the AWS managed policy and scope it down
- Allow access to:
  - Source Backup Vault KMS key
  - Source Backup Snapshots
    - Regex generated S3 bucket
    - *awsbackup-restore-test-**



S3 Backups - Restoration

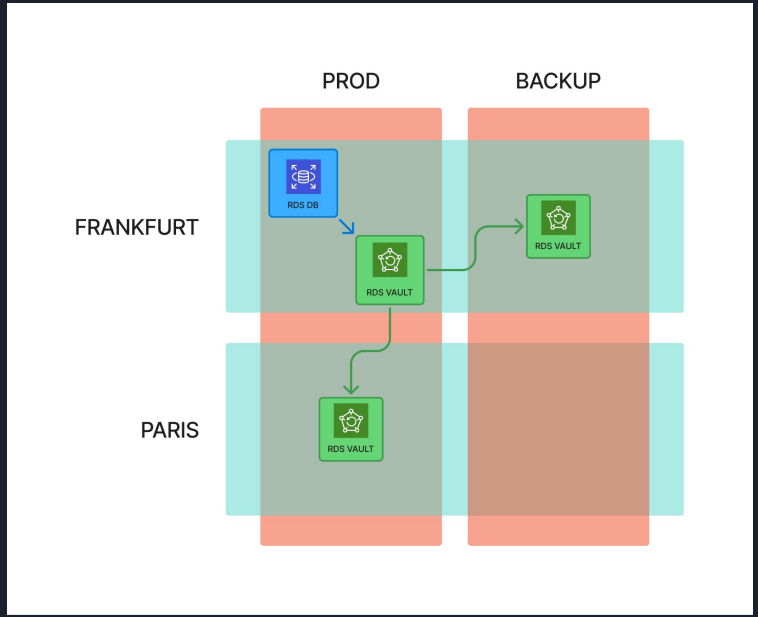
- Manual restoration process
  - Destination -> Source
  - Restore from snapshot
- Need a restore role for KMS key access
- Need to enable ACLs before restoring
- Make sure to not have too constrained restoration time settings



RDS

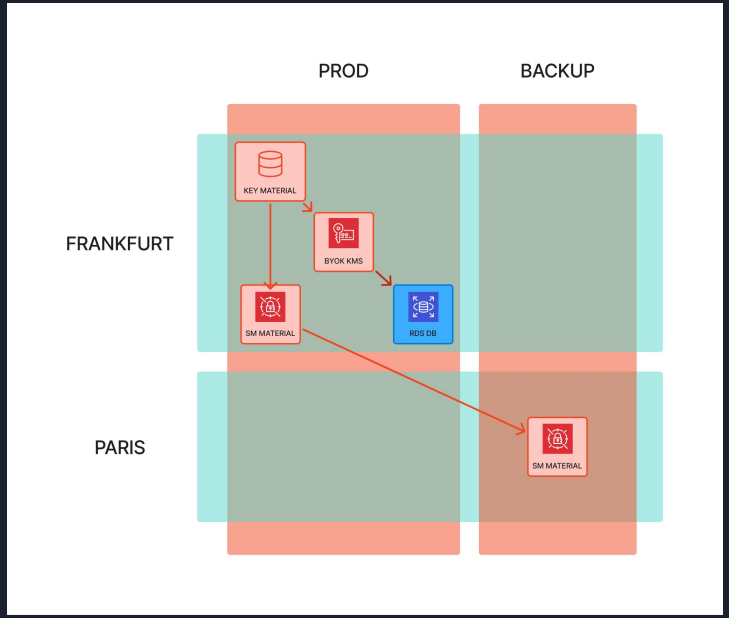
RDS Backups - Setup

- Backup service limit to only do either CR or CA Copy job
- Use snapshot lifecycles/delete after argument to avoid cost of duplicate snapshots
- Optimize cost using lifecycles/delete after arguments to specific needs

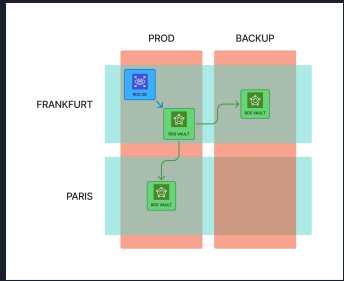


RDS Backups - KMS

- Single RDS instance constraints
  - RDS is built on top of an EC2 instance with and encrypted EBS volume on logical level
  - Backup snapshot is therefore doubly encrypted (EBS volume + Backup Vault)
  - What happens if Frankfurt/Prod account is flooded/deleted 🤔
- Use CMK KMS key with Key Material that is generated on deploy and duplicated to Secrets Manager on different accounts



RDS Backups - Security



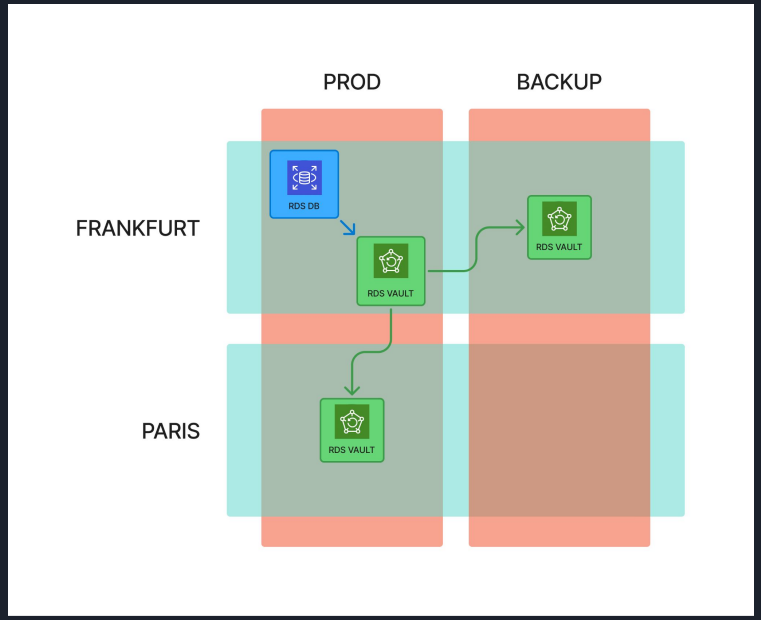
- Create 3 policies
  - Regular, CR, CA
- Start with the AWS managed policy and scope it down
- Allow access to (Regular):
  - RDS KMS key
  - Source Backup Vault KMS key
  - Source Backup Vault (Describe, Tag)
  - RDS Snapshots (Create, Copy)
  - RDS Tags (Add, List, Describe)

Additionally allow access to: (CR/CA):

- Destination Backup Vault KMS key
- AWS Managed Backup KMS key (CR)
- Backup resource tagging
- Source/Destination Backup Vault
  - *CopyIntoBackupVault*
- Source/Destination Backup Snapshots
  - *CopyFromBackupVault*
  - Regex generated snapshot ARN format

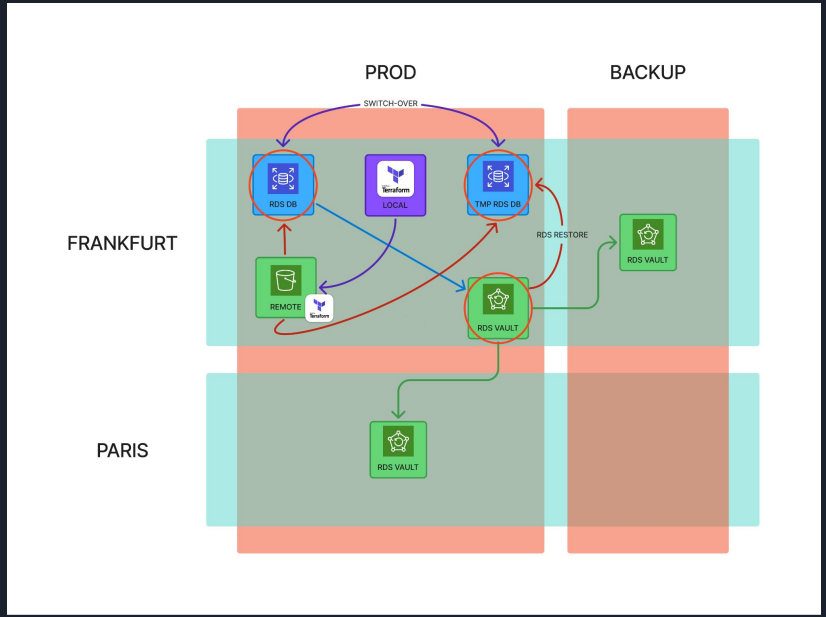
RDS Backups - Restore testing

- Regular restore testing choosing a random/specific snapshot
- Limit to Source Backup vault testing
- Start with the AWS managed policy and scope it down
- Allow access to:
  - Source Backup Vault KMS key
  - Source Backup Snapshots
    - Regex generated S3 bucket
    - *awsbackup-restore-test-**
- Needs restore metadata override to use correct DB subnet



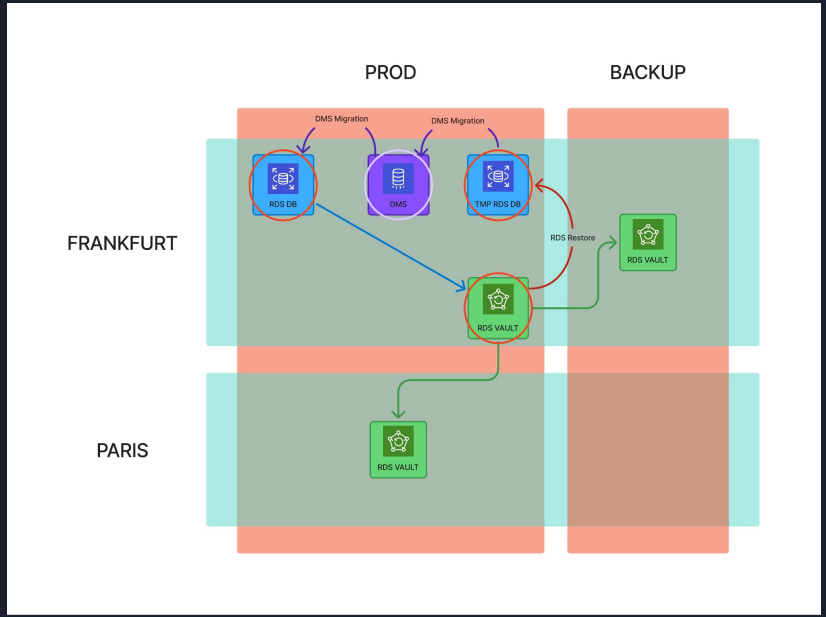
RDS Backups - Restoration v1

- Blue / Green DB deployment setup
- Terraform + Bash script approach
- Terraform state manipulation
    - Step by step apply
    1. Have the original DB
    2. Create a restored DB from snapshot
    3. Switch over between them
    4. Apply the changes
- **Does not work since data is not transferred, while terraform resources are!**
- Could potentially work by introducing a third DB instance (Main / Original / Restored)



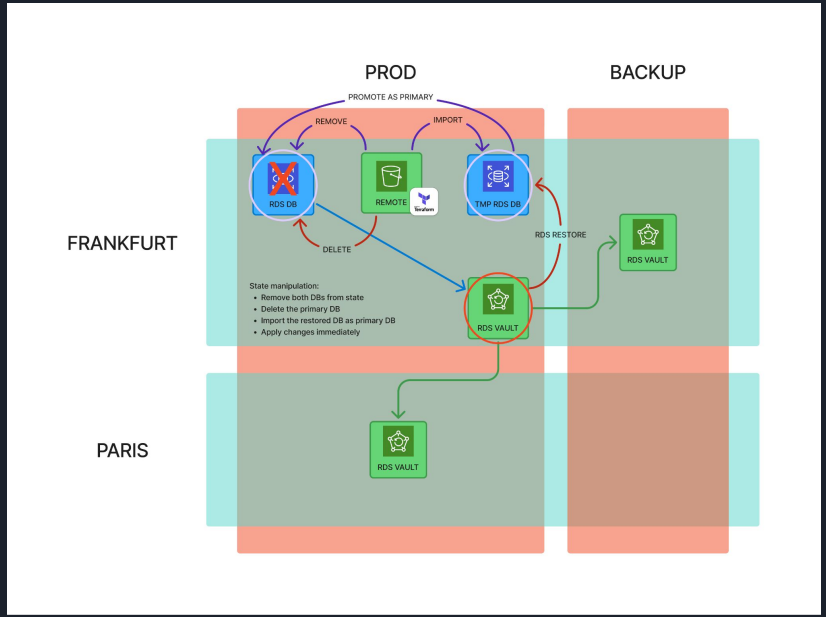
RDS Backups - Restoration v2

- Terraform native approach
- DMS migration task
  - Step by step apply
  1. Have the original DB
  2. Create a restored DB from snapshot
  3. Run a migration between them
  4. Apply the changes
  5. Wait for migration task to complete
  6. Destroy the temporary resources
- Most clean approach between all three
- Limit to data that is homogeneously transferable between DBs



RDS Backups - Restoration v3

- Database replacement
- Terraform + Bash script approach
- Terraform state manipulation
    - Step by step apply
    1. Have the original DB
    2. Create a restored DB from snapshot
    3. Delete both DBs from state
    4. Delete the original DB instance
    5. Import the restored DB as original DB
    6. Apply changes immediately
- No limits regarding data format



QA

That’s all, folks! 🚀

May the snapshots be with you - across regions and accounts

Restoring balance to the cloud when the Empire strikes your infrastructure