Network Storage & Data Cycle - EFS (Elastic File System)

The Elastic File System (EFS) is an AWS managed implementation of NFS which allows for the creation of shared 'filesystems' which can be mounted within multi EC2 instances.

- For scalability / resiliency
- EFS is an implementation of NSFv4 (network file system version 4)
- Can be mounted in EC2 linux and shared between many EC2 instances
- EFS is private service that mounts targets inside a VPC
- Can be accessed from on-premises via VPN or AWS Direct Connect (DX)
- Uses POSIX permissions (a linux thing) EXAM: EFS is Linux ONLY EXAM: 2 performance modes: 1. General Purpose 2. Max I/O EXAM: 2 throughput modes: 1. Bursting 2. Provisioned EXAM: 2 storage classes: 1. Standard 2. Infrequent Access (IA) - S3 Lifecycle policies can be used with these

DEMO - Implementing EFS

Implement a simple EFS file system in a VPC, configure mount targets and configure two EC2 instances to mount the file system within a mount point.

Video: https://learn.cantrill.io/courses/1820301/lectures/41301666

Steps:

1. 1 click deploy
2. Create EFS: EFS > Create File System > Customize > name "A4L-EFS", storage class Standard, disable Encryption of data at rest, pPrformance Settings: Bursting > Next
3. EFS Network Settings: Create mount targets > Delete default SG's (blue bloxes) > Attach APP subnets: us-east-1a, "sn-app-A" etc > Security Group x3 "IMPLEMENTINGEFS[...]" > Next > Skip File System policy (optional) > Next > Review and CREATE NOTE: any AZs within a VPC you're consuming the services provided by EFS, you should be creating a mount target
4. EC2 Instances > duplicate EC2 instance tab > tab 1, EC2 connect to instance A, tab 2 connect to B
5. EC2 Connect Instance A > `sudo mkdir -p /efs/wp-content` (creates file as well as any paths like /efs/ with the -p) > `sudo dnf -y install amazon-efs-utils` package of tools which allows this instance to interact with EFS > `cd /etc` > `sudo nano /etc/fstab` > Paste this into 3rd line of nano: `file-system-id:/ /efs/wp-content efs _netdev,tls,iam 0 0` >
6. Paste EFS file system ID of A4L EFS into the nano text above, replacing 'file-system-id`
7. Save Nano: ctrl+o to save, Enter, ctrl+x to exit
8. `df -k` still not showing anything. Time to mount file system: `sudo mount /efs/wp-content`, now `df -k` will show it > `cd /efs/wp-content` (move into new file system) > `sudo touch amazingtestfile.txt` (create test file)
9. Instance B Connect: df -k > sudo dnf -y install amazon-efs-utils > sudo mkdir -p /efs/wp-content > sudo nano /etc/fstab > file-system-id:/ /efs/wp-content efs _netdev,tls,iam 0 0 > sudo mount /efs/wp-content > cd /efs/wp-content/ > ls -la -- you'll see amazingtestfile.txt in the directory which was created in instance A
10. Clean up: EFS > Delete file system > Cloudformation > Delete stack

AWS Backup

Use AWS Backup to centralize and automate data protection across AWS services and hybrid workloads. AWS Backup offers a cost-effective, fully managed, policy-based service that further simplifies data protection at scale. AWS Backup also helps you support your regulatory compliance or business policies for data protection. Together with AWS Organizations, you can use AWS Backup to centrally deploy data protection policies to configure, manage, and govern your backup activity across your company's AWS accounts and resources.

- fully managed data-protection
- consolidate backups across accounts/regions into one place

AWS Backup - Components

- Backup Plans: frequency, window, lifecycle, vault, region copy
- Backup resources: the resources being backed up
- Vaults: destination for backups (container). Assign KMS key for encryption
- Vault Lock: WORM write-one-read-many, 72 hour cool off, then even AWS can't delete
- On-Demand backups when you require
- PITR: Point in time recovery