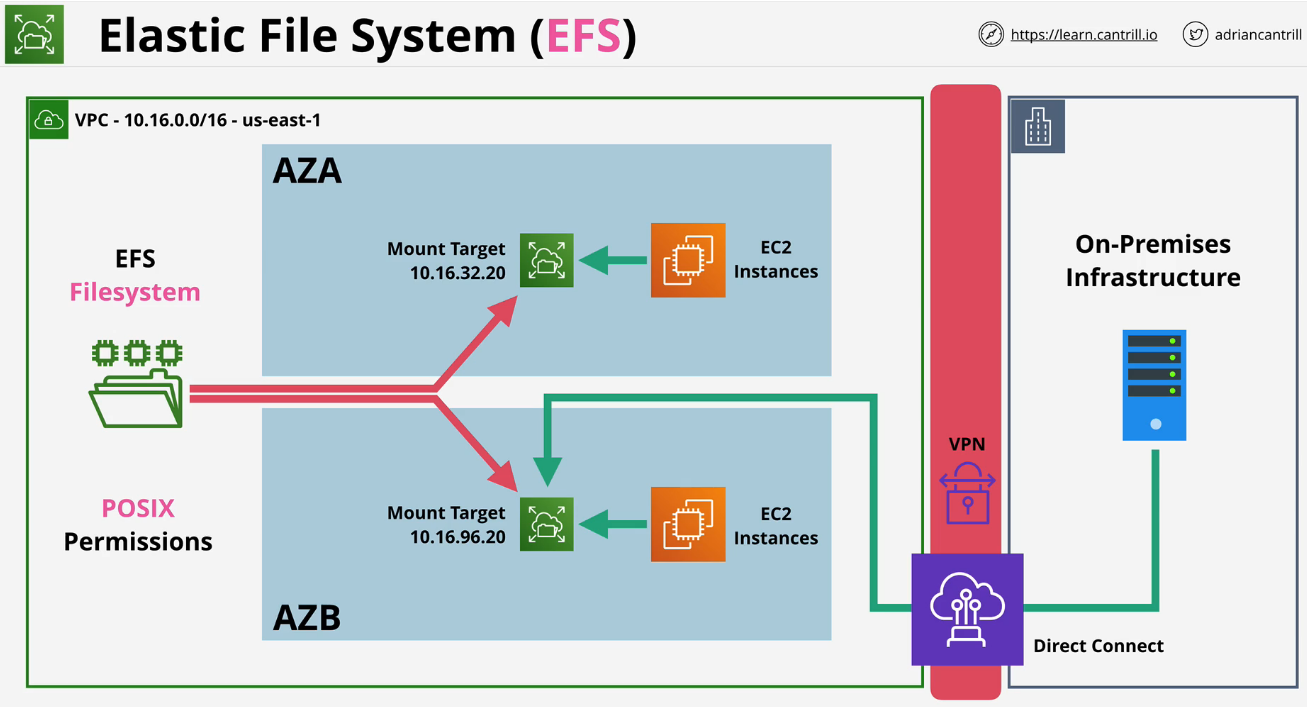
**Network-Storage-EFS**

**EFS Architecture**

EFS moves the instances closer to being stateless.

* EFS is an implementation of **NFSv4**
* EFS file systems are created and **mounted in Linux.**
* EFS storage exists separately from an EC2 instance like EBS does.
  + EBS is **block storage**
  + EFS is **file storage**
* Media can be shared between many EC2 instances.
* EFS is a **private service.**
  + **Isolated to the VPC its provisioned into.**
  + Access is via **mount targets inside the VPC**.
* **EFS access outside of the VPC** with
  + VPC peering
  + VPN connections
  + AWS direct connect



**Elastic File System Explained**

EFS runs inside a VPC. Inside EFS you create file systems and these use **POSIX permissions**.

EFS is **made available inside a VPC via mount targets**.

**Mount targets** have **IP addresses taken from the IP address range of the subnet they're inside**.

For **High Availability**, you need to make sure that you **put mount targets in each AZ the system runs in**.

You can **use hybrid networking to connect to the same mount targets**.

**EFS Exam PowerUp**

* EFS is **Linux Only**
* Two **performance modes**:
  + **General purpose**: General purpose is good for latency sensitive use cases.
    - General purpose should be default for 99.9% of uses.
  + **Max I/O:** Max I/O performance mode can **scale to higher levels of aggregate t-put and IOPS** but it does have **increased latencies**.
* Two **t-put modes**:
  + **Bursting** works like GP2 volumes inside EBS with a burst pool. The more data you store in the FS, the better performance you get.
  + **Provisioned** t-put modes can specify t-put requirements separately from size.
* Two **storage classes** available:
  + Standard
  + Infrequent access
  + Can use lifecycle policies to move data between classes.

**Refer to the Linux Commands to mount the efs file system.**

