Amazon EFS is a [file storage service](https://aws.amazon.com/what-is-cloud-file-storage/) for use with Amazon EC2. Amazon EFS provides a file system interface,

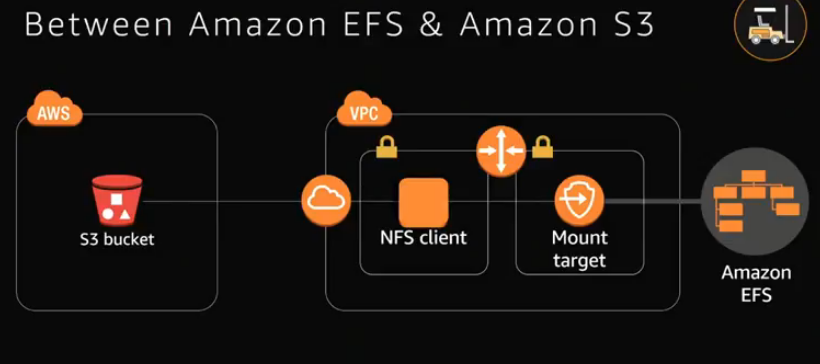
* file system access semantics (such as strong consistency and file locking), and

concurrently-accessible storage for up to thousands of Amazon EC2 instances.

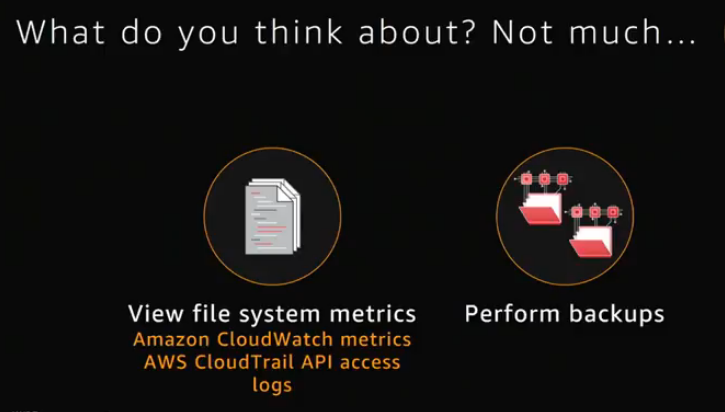
Outside VPC – no need to SSH

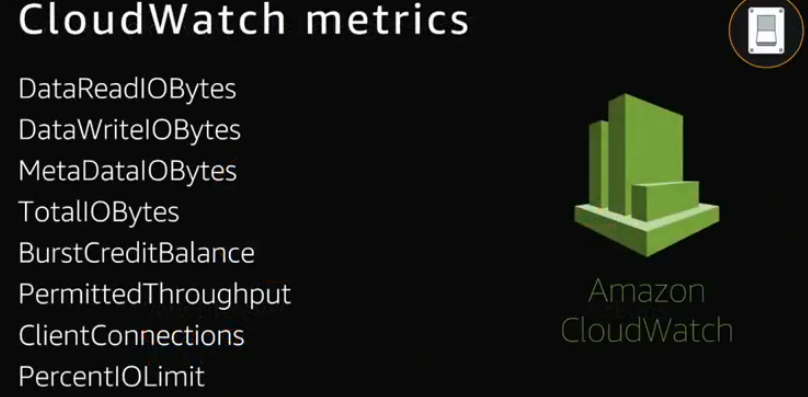


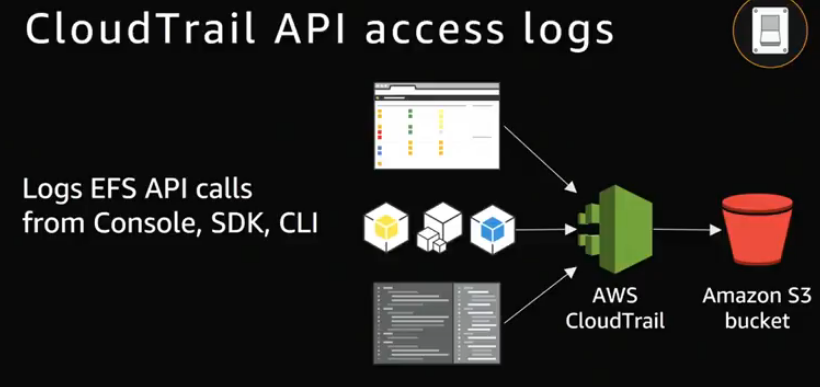
DEMO Sync

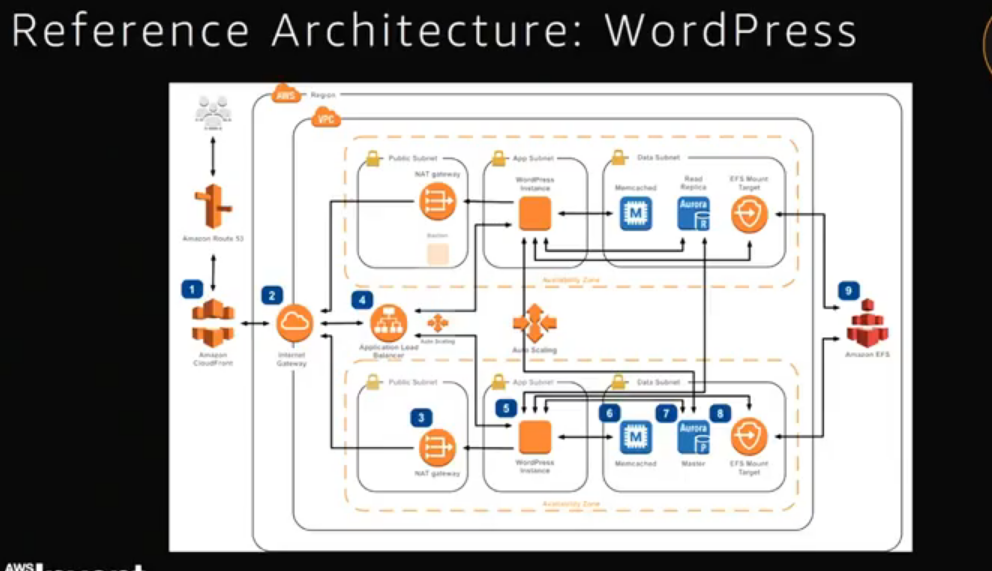


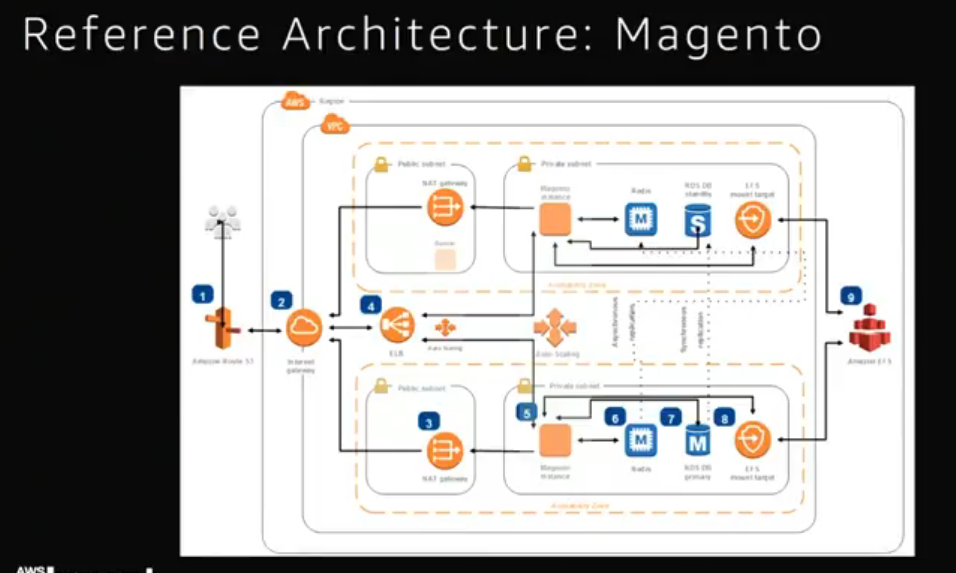
**RUN PHASE**

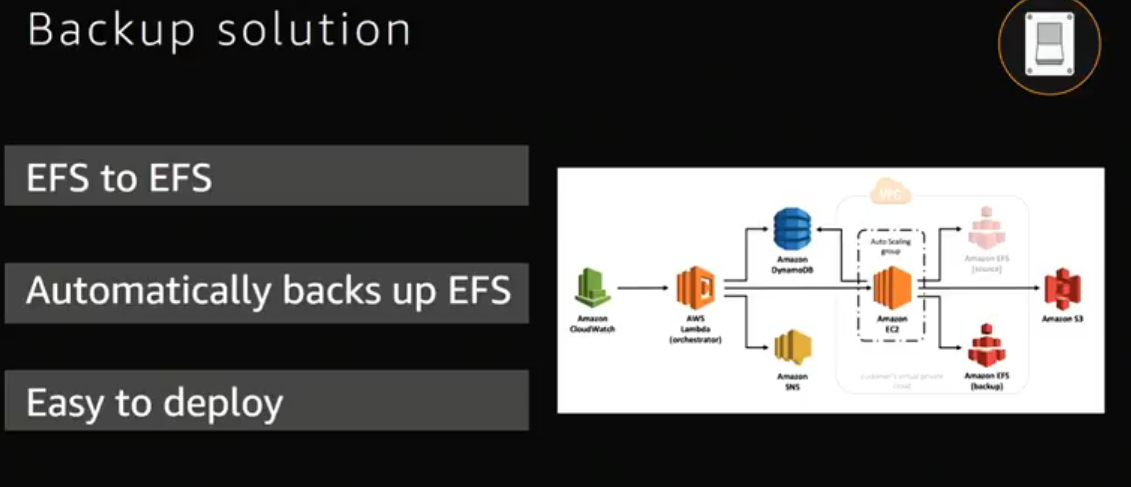




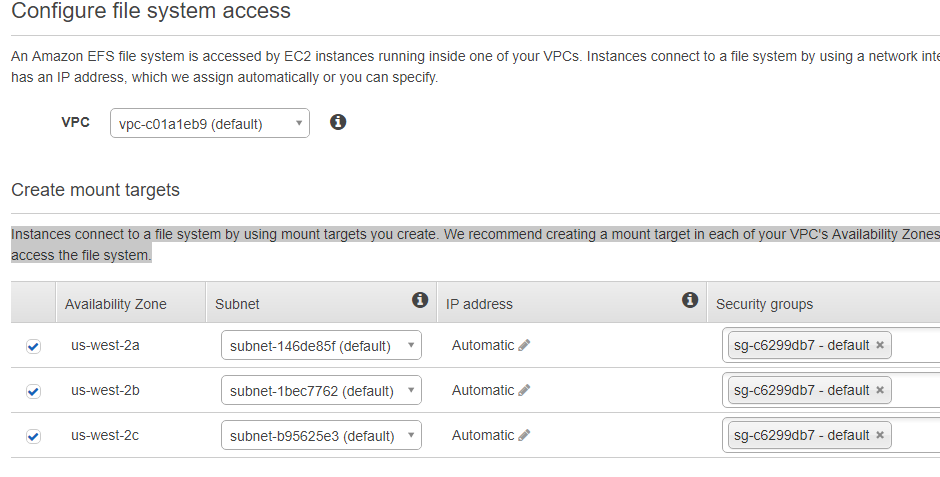


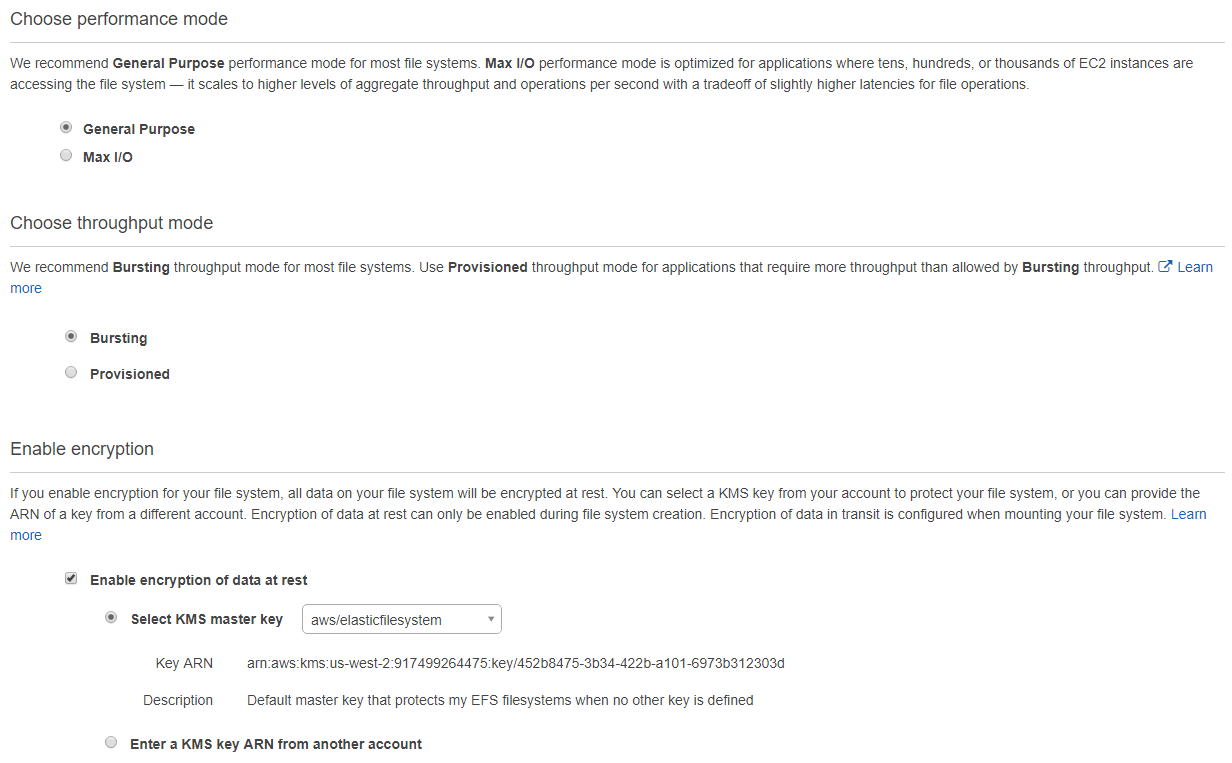






* EFS is associated with a single VPC at a time
* VMs in that VPC can access that file system
* An Amazon EFS file system is accessed by EC2 instances running inside one of your VPCs.
* Instances connect to a file system by using a network interface called a mount target.
* Each mount target has an IP address, which we assign automatically or you can specify.
* Instances connect to a file system by using mount targets you create. We recommend creating a mount target in each of your VPC's Availability Zones so that EC2 instances across your VPC can access the file system.
* You can use an Amazon EFS file system in one VPC at a time.
* VMs in another VPC can connect to EFS –
  + VPC peering needs to be enabled
  + VPC VMs types must be M5 or C5
  + VPC must be in same region.
  + Only one VPC at a time..





**General Purpose Performance Mode**

We recommend the General Purpose performance mode for the majority of your Amazon EFS file systems.

General Purpose is ideal for latency-sensitive use cases, like web serving environments, content management systems, home directories, and general file serving.

**Max I/O Performance Mode**

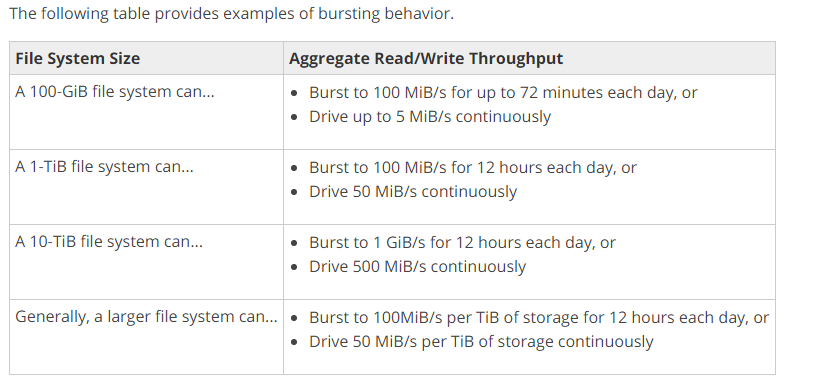
File systems in the Max I/O mode can scale to higher levels of aggregate throughput and operations per second with a tradeoff of slightly higher latencies for file operations.

Highly parallelized applications and workloads, such as big data analysis, media processing, and genomics analysis, can benefit from this mode.

**Throughput Scaling with Bursting Mode**

With Bursting Throughput mode, throughput on Amazon EFS scales as a file system grows. File-based workloads are typically spiky, driving high levels of throughput for short periods of time, and low levels of throughput the rest of the time. To accommodate this, Amazon EFS is designed to burst to high throughput levels for periods of time.

More data, more throughput..



**Specifying Throughput with Provisioned Mode**

Provisioned Throughput mode is available for applications with high throughput to storage (MiB/s per TiB) ratios, or with requirements greater than those allowed by the Bursting Throughput mode. For example, say you're using Amazon EFS for development tools, web serving, or content management applications where the amount of data in your file system is low relative to throughput demands. Your file system can now get the high levels of throughput your applications require without having to pad your file system.

If data is less and need more throughput..

To use the mount command, the following must be true:

* The connecting EC2 instance must be in a VPC and must be configured to use the DNS server provided by Amazon
* The VPC of the connecting EC2 instance must have DNS hostnames enabled

You can mount your Amazon EFS file systems on your on-premises data center servers when connected to your Amazon VPC with AWS Direct Connect.

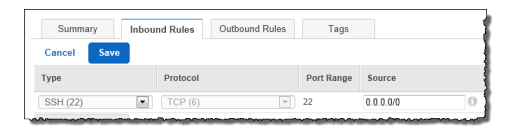
SG for EC2 and for mounts. you must configure the following rules in these security groups:

* The security groups you associate with a mount target must allow inbound access for the TCP protocol on the NFS port from all EC2 instances on which you want to mount the file system.

Allow inbound – NFS 2049

* + 
* Each EC2 instance that mounts the file system must have a security group that allows outbound access to the mount target on the NFS port.

Allow Inbound – SSH 22



**Loading data options**

1. Mount to EC2 and use copy command
2. Use file sync..

**How do I access my file system from outside my VPC?**

* Amazon EC2 instances within your VPC can access your file system directly, and
* Amazon EC2 Classic instances outside your VPC can mount a file system via [ClassicLink](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/vpc-classiclink.html).
* On-premises servers can mount your file systems via an [AWS Direct Connect](https://aws.amazon.com/directconnect/) connection to your VPC.

**Amazon EFS offers the ability to encrypt data at rest and in transit.**

Data encrypted at rest is transparently encrypted while being written, and transparently decrypted while being read, so you don’t have to modify your applications. Encryption keys are managed by the AWS Key Management Service (KMS),

Data encryption in transit uses industry standard Transport Layer Security (TLS) 1.2 to encrypt data sent between your clients and EFS file systems.

**Amazon EFS does not support access over AWS VPN.**

Can I access my Amazon EFS file system concurrently from my on-premises datacenter servers as well as Amazon EC2 instances? YES

EFS File Sync provides a fast and simple way to securely move data from existing on-premises or in-cloud file systems into Amazon EFS file systems. EFS File Sync copies files and directories into Amazon EFS at speeds up to 5x faster than standard Linux copy tools,   **deploy File Sync agent**

Every file system has an automatically generated ID number that is globally unique. You can tag your file system with a name, and these names do not need to be unique.

Amazon EFS is integrated with a number of other AWS services, including **Amazon CloudWatch, AWS CloudFormation, AWS CloudTrail, AWS IAM, and AWS Tagging services**.

Amazon CloudWatch allows you to monitor file system activity using metrics. AWS CloudFormation allows you to create and manage file systems using templates.

AWS CloudTrail allows you to record all Amazon EFS API calls in log files.

AWS Identity and Access Management (IAM) allows you to control who can administer your file system. AWS Tagging services allows you to label your file systems with metadata that you define.