



Persistent File Storage for Amazon EKS with Amazon EFS

Will Ochandarena (ochanw@)
Principal Product Manager, Amazon EFS

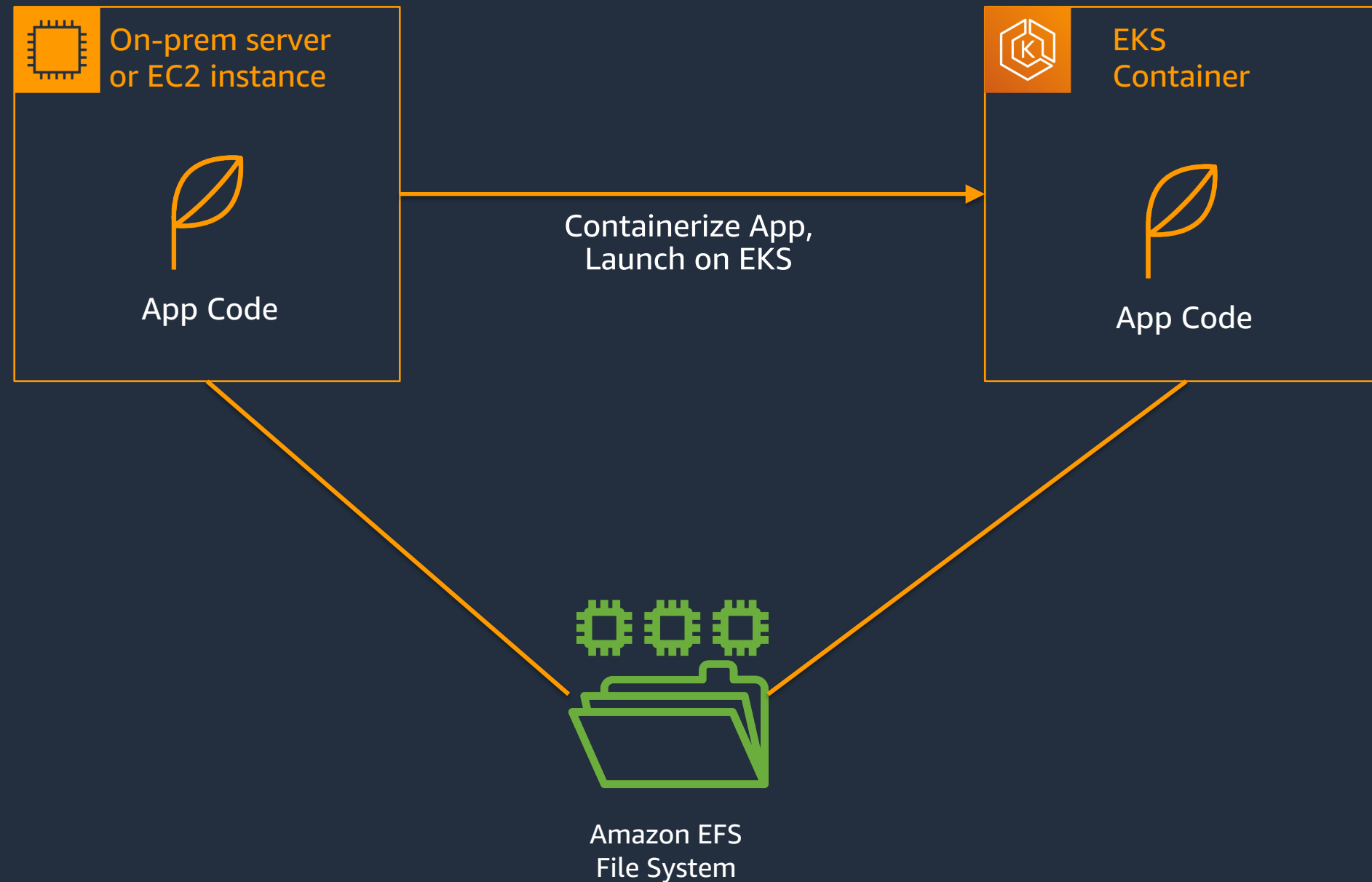


Agenda

- When Do I Need EFS for EKS Containers?
- What's New & What's Coming
- Identity: Bridging Between Pods & Storage
- Using EFS from EKS (using the CSI Driver)
- Best Practices: Performance, Cost, & Ingest

When Do I Need EFS for EKS Containers?

Serverless storage for your App Modernization



Why are customers using modernizing their apps using containers and EFS?



1. Save cost by reducing underutilization of compute & storage



2. Increase agility by instantly scaling-up according to demand



3. Develop & deploy applications with greater efficiency

Many containerized applications need persistent storage

Long-running
Stateful Applications

Shared Data Sets



Developer
Tools

.....

Jenkins
Jira
Git



Web & Content
Management

.....

WordPress
Drupal
nginx



Machine
Learning

.....

MXNet
TensorFlow



Data Science
Tools

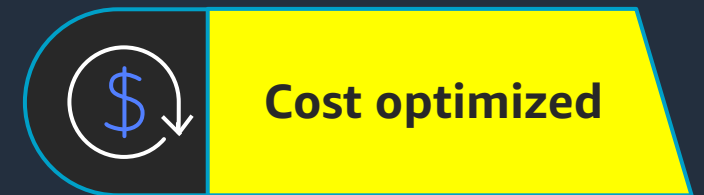
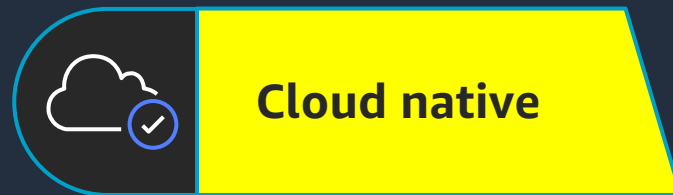
.....

Jupyter(hub)
Airflow

Amazon Elastic File System (Amazon EFS)



Amazon EFS Serverless File Storage



Simplify Persistent Storage for Amazon EKS with Amazon EFS

Simple

Amazon EFS configuration is done from K8S-native objects (e.g. Persistent Volume), so developers can focus on their applications, not infrastructure.

Available and Durable

Amazon EKS, and Amazon EFS are regional services. Customers can build applications that span multiple availability zones, with automatic failover.



Elastic

Amazon EKS and Amazon EFS are elastic, scale up and down rapidly based on demand. Customers pay only for what they use.

Secure

Amazon EFS Access Points can enforce file system permissions when multiple apps share a file system.

T-Mobile scales modern application deployments with Amazon EFS



Challenge

Customer facing application with large spikes in usage based on time of day and month of year. Existing infrastructure was not able to support the scalability required without overprovision of infrastructure to support peak usage.

Solution

Modernized applications to employ microservices. Deployed containers via Kubernetes and Mesos with EFS providing persistent storage and ability to dynamically scale application without storage management overhead

Benefits

- 16,000 containers under management
- Reduced cost of NFS storage by 70% compared to DIY while reducing storage management overhead
- Improved cycle time for deploying application services

“

We are a large organization that has lots of applications with **varying requirements for availability and performance**. EFS provides us with a common storage platform that meets these requirements across the board.

”

Amreth Chandrasehar, Principal Architect, T-Mobile

Company:	T-Mobile
Industry:	Mobile Communications
Country:	Global
Employees:	52,000
Website:	www.t-mobile.com

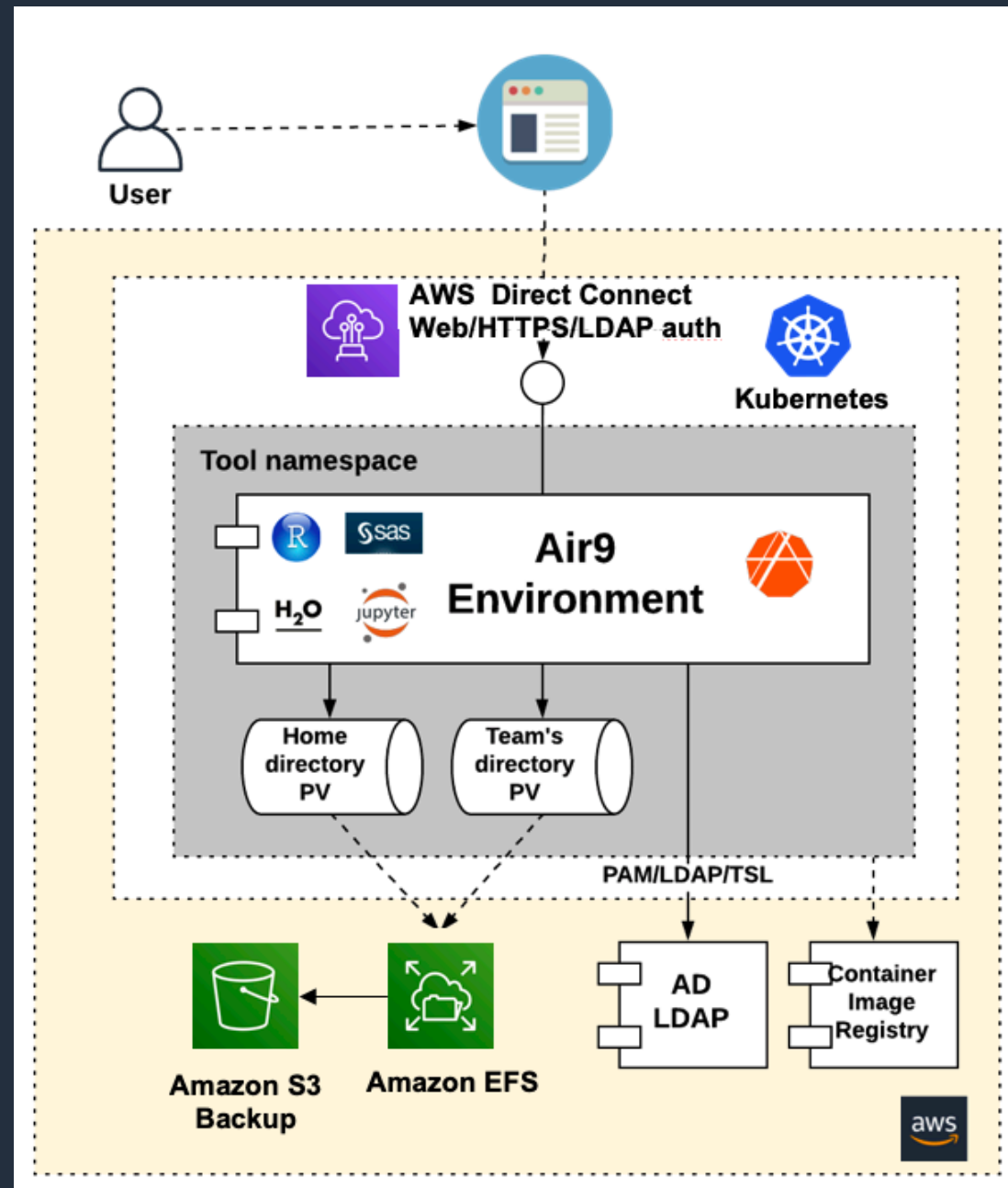
About T-Mobile

As America's Un-carrier, T-Mobile US, Inc. is redefining the way consumers and businesses buy wireless services through leading product and service innovation. The Company's advanced nationwide network delivers outstanding wireless experiences to 79.7 million customers who are unwilling to compromise on quality and value.



Journey to (and in) the cloud

DISCOVER[®]



- Moved containerized data science environment to AWS for agility and cost benefits
- Enabled self-service provisioning of containerized analytics applications and compute resources
- Migrated to a managed service for better stability, application scaling and ease of operations, reducing storage management time by 90%

What's New & What's Coming

What's New

Amazon EFS CSI Driver is now generally available

Posted On: Jul 24, 2020

- EFS Access Points Support
- TLS Enabled By Default
- Compatible with v1.2.0 CSI Spec

What's Coming

<https://github.com/aws/containers-roadmap/projects/1>

53

We're Working On It

...

1 result

 [EKS/Fargate] [Volumes]: Add support for EFS volumes to EKS Fargate Containers


...

#826 opened by hlascelles

EKS


Fargate



-  [EKS] [request]: EFS CSI driver automatically provision EFS directory or access point

EKS

Proposed

#891 opened on May 12 by chasmosis
-  [EKS] [request]: EFS CSI driver should use service account role for IAM authorization

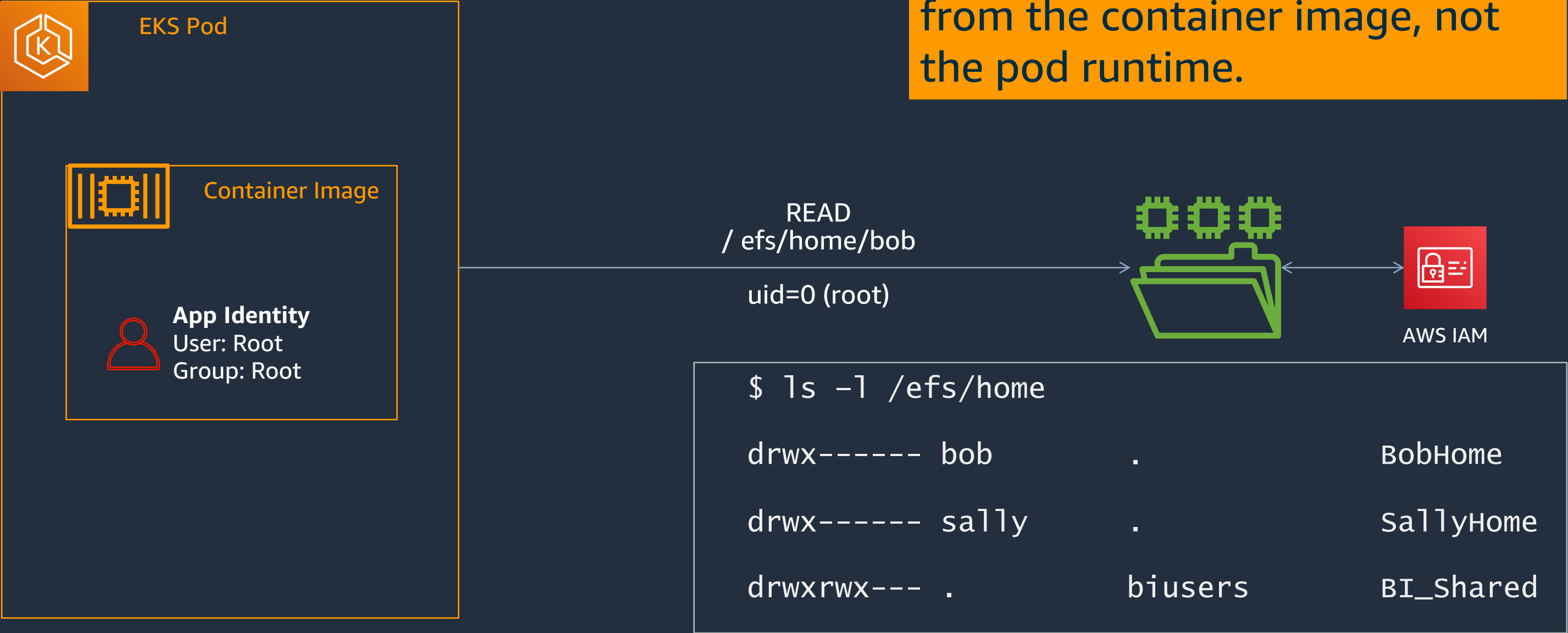
Proposed

#1003 opened 19 minutes ago by wochanda

Identity: Bridging Between Pods & Storage

Understanding Container Identity

By default, POSIX identity comes from the container image, not the pod runtime.



Application-specific Access with EFS Access Points

EFS Access Point

```
{
  "Name": "MyApp",
  "FileSystemId": "fs-deadbeef",
  "PosixUser": {
    "Uid": 123
    "Gid": 123,
    "SecondaryGids": [100, 200, 300]
  },
  "RootDirectory": {
    "Path": "/apps/myapp",
    "CreationInfo": {
      "OwnerUid": 123,
      "OwnerGid": 123,
      "Permissions": "0700"
    }
  }
}
```

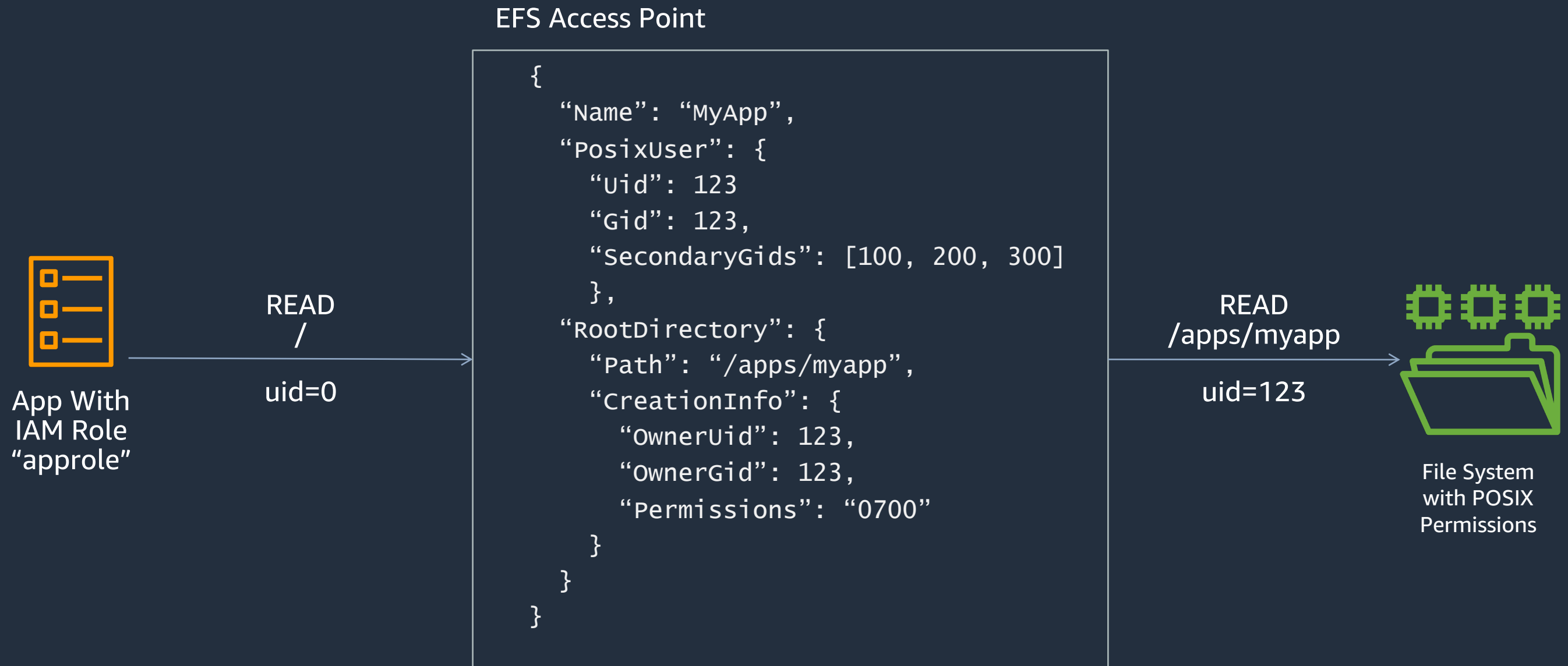
Enforces File System Identity

Root containers can't escalate access
Arbitrary users aren't locked out

Creates App-specific Directory & Permissions

No EC2 instance required!
Apps only see data they need

How EFS Access Points Work



Access Point Use Cases

- Solving container identity for sharing EFS data
- Packing multiple applications into a single file system
 - Less resources to manage
 - Sharing pool of throughput between multiple apps

Using EFS with EKS (using the CSI Driver)

EFS & EKS: Concepts



Amazon Elastic
Kubernetes Service

- Container Storage Interface (CSI)
 - Industry standard interface for connecting storage providers (block or file) to a container.
- EFS CSI Driver
 - Implementation of CSI for connecting EFS file systems to containers.
- Storage Class (SC)
 - Administrator-defined class of storage that Persistent Volumes can be created from.
- Persistent Volume (PV)
 - Administrator-created unit of storage that can be attached to a container. Has its own lifecycle.
- Persistent Volume Claim (PVC)
 - Request to allocate an available PV from a SC to a container.

EKS Storage – Process Flow



Amazon Elastic
Kubernetes Service

1. Admin Creates SC & PVs

Storage Class (name: GeneralPurposeEFS)

Persistent Volume

Name: PV1 FS:fs-deadbeef Path: /pv1/

Persistent Volume

Name: PV2 FS:fs-deadbeef Path: /pv2/

Persistent Volume

Name: PV3 FS:fs-deadbeef Path: /pv3/

Persistent Volume

Name: PV4 FS:fs-deadbeef Path: /pv4/

Persistent Volume

Name: PV5 FS:fs-deadbeef Path: /pv5/

2. Dev Claims PVs from SC

Persistent Volume Claim

Name: MyAppClaim

SC: GeneralPurposeEFS

3. Dev Launches Pod Referencing PV Claim

Pod

Name: MyApp

PVC: MyAppClaim

Attaching an EFS file system to a Pod (Admin)



Amazon Elastic
Kubernetes Service

Create Storage Class

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: efs-sc
provisioner: efs.csi.aws.com
```

Create Persistent Volume

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: efs-pv
spec:
  capacity:
    storage: 5Gi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteMany
  persistentVolumeReclaimPolicy: Retain
  storageClassName: efs-sc
  csi:
    driver: efs.csi.aws.com
    volumeHandle: fs-deadbeef::fsap-deadbeefdead
```

New – VolumeHandle References Path & Access Point

 Support access points on the same file system ✓ approved cncf-cla: yes lgtn ok-to-test size/L
#185 by 2uasimojo was merged on Jun 18

volumeHandle: {fsid}:{subpath}:{apid}

Examples:

Mounting a sub-path: fs-deadbeef:/myapppath/:

Mounting an AP: fs-deadbeef::fsap-feeddeadbeeffeed

Sub-mounting an AP: fs-deadbeef:/subpath:fsap-feeddeadbeeffeed

Attaching an EFS file system to a Pod (User)



Amazon Elastic
Kubernetes Service

Create Persistent Volume Claim

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: efs-claim
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: efs-sc
resources:
  requests:
    storage: 5Gi
```

Launch Pod

```
apiVersion: v1
kind: Pod
metadata:
  name: efs-app
spec:
  containers:
    - name: web-container
      image: httpd
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - name: persistent-storage
          mountPath: /mnt-efs
  volumes:
    - name: persistent-storage
      persistentVolumeClaim:
        claimName: efs-claim
```


Best Practices: Performance, Cost, & Ingest

Best Practices – Performance

Amazon Elastic File System announces 400% increase in read operations for General Purpose mode file systems

Posted On: Apr 1, 2020

Amazon Elastic File System increases per-client throughput by 100%

Posted On: Jul 23, 2020

- **Use General Purpose** for most apps
 - GP lower latency, now supports up to 35K read IOPS
 - MaxIO for scale-out analytics/ML that need 100k+ IOPS
- **Configure provisioned throughput** for initial need
 - As your file system grows you'll eventually be given higher throughput
- **Set up Amazon CloudWatch**, monitor throughput, IOPS, and burst credits*

When should I use EFS vs EBS?



Amazon
Elastic File
System

- I need to share data between containers
- I'd like to run across instances or AZs
- I'd like to take advantage of spot pricing



Amazon Elastic
Block Store

- I need low latency (e.g. MySQL)
- I need point in time snapshots

Note: Amazon FSx for Lustre can be used for containers that require ultra-high throughput and very low latency file sharing

Optimize cost with Amazon EFS Infrequent Access

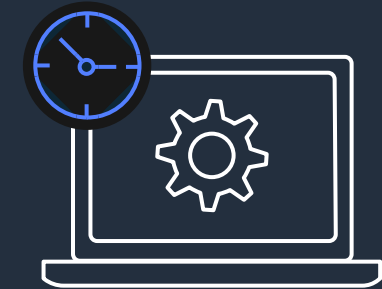
Amazon EFS IA storage class for infrequently accessed files for \$0.025/GB/mo*



No changes to existing
applications using
Amazon EFS



Cost
savings up
to 92%



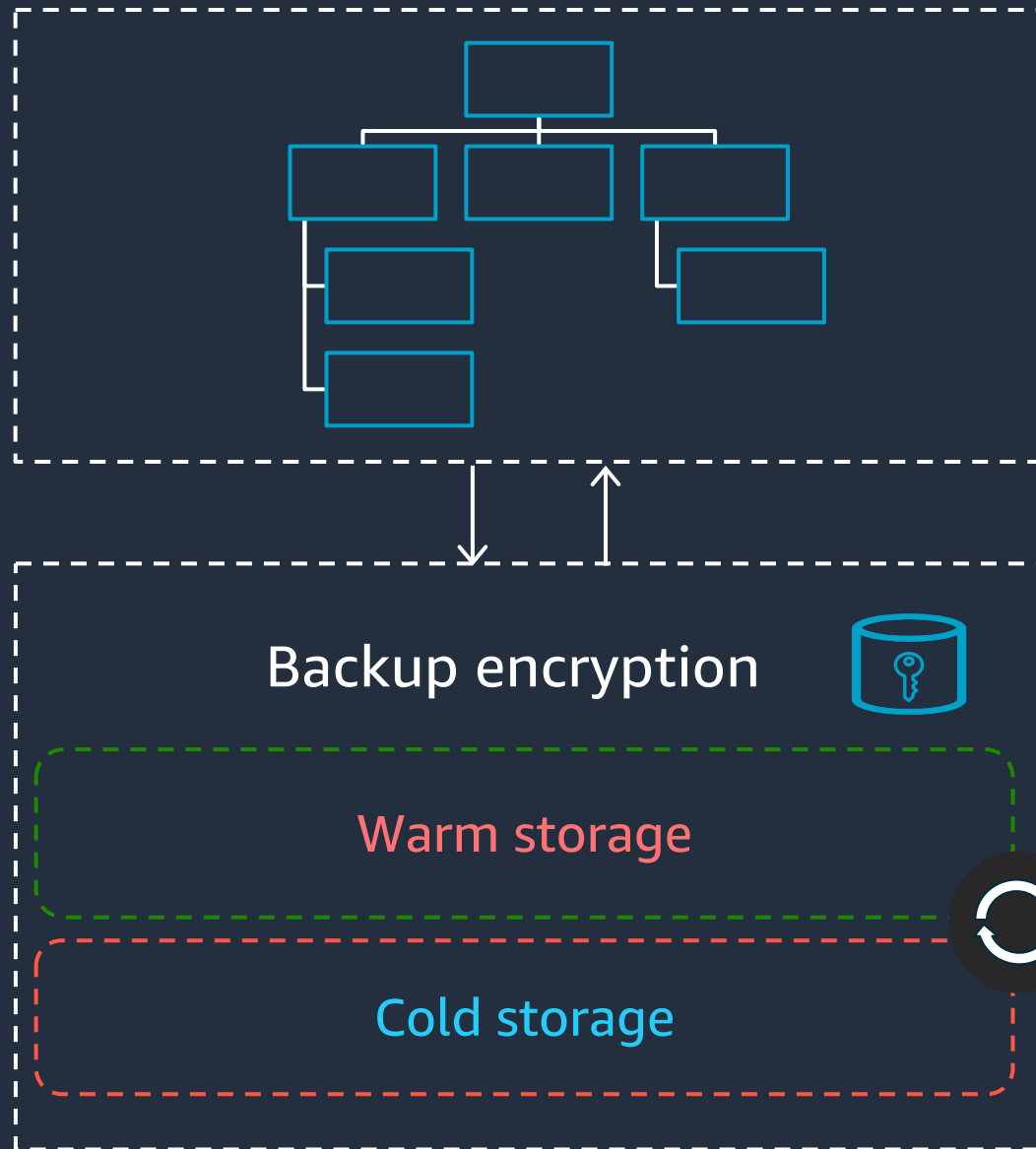
Automated
lifecycle
management

* Pricing in the US East (N. Virginia) region

Backup for Amazon EFS



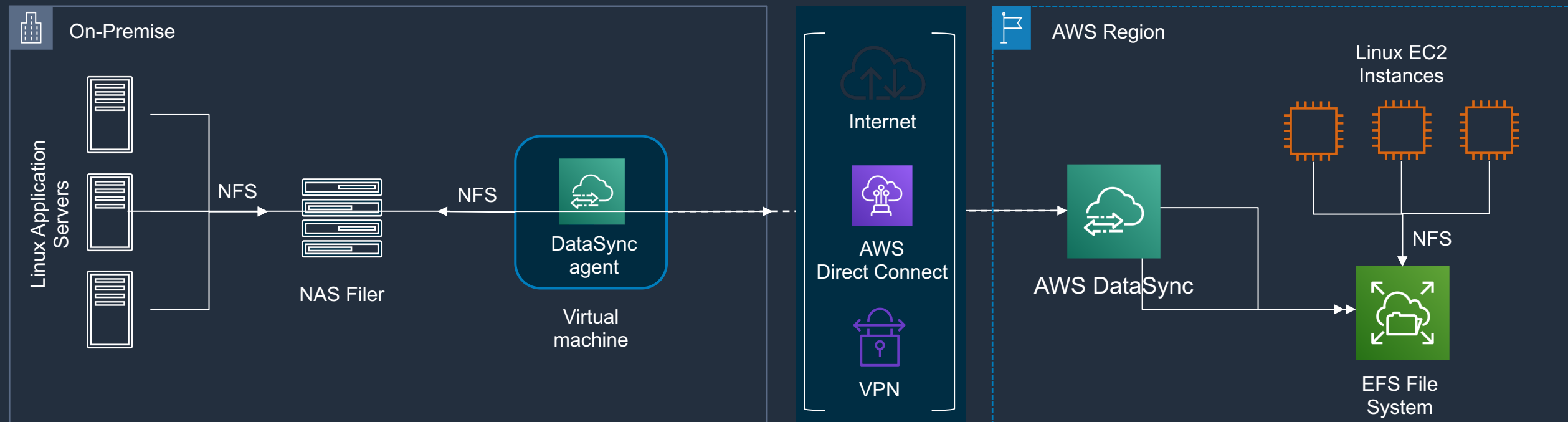
Amazon EFS



AWS Backup

- EFS file systems can be backed up and restored using AWS Backup
- AWS Backup provides automated backup scheduling and retention per user defined policy
- AWS Backup offers two classes of service backup storage with the ability to lifecycle to cold storage
- Restore individual files and directories

Migrating NFS workloads to EFS



AWS DataSync: Online transfer service that simplifies, automates, and accelerates moving data between on-premises storage and AWS

Where to learn more

[Amazon EKS Documentation: Amazon EFS CSI Driver](#)

[Deploying Jenkins on Amazon EKS with Amazon EFS](#)



Thank you!

