SCALE 21X, MARCH 15 2024

Vending Machine

for Data Science Experiments

Christina Andonov

Senior Specialist Solutions Architect
AWS

Apoorva Kulkarni

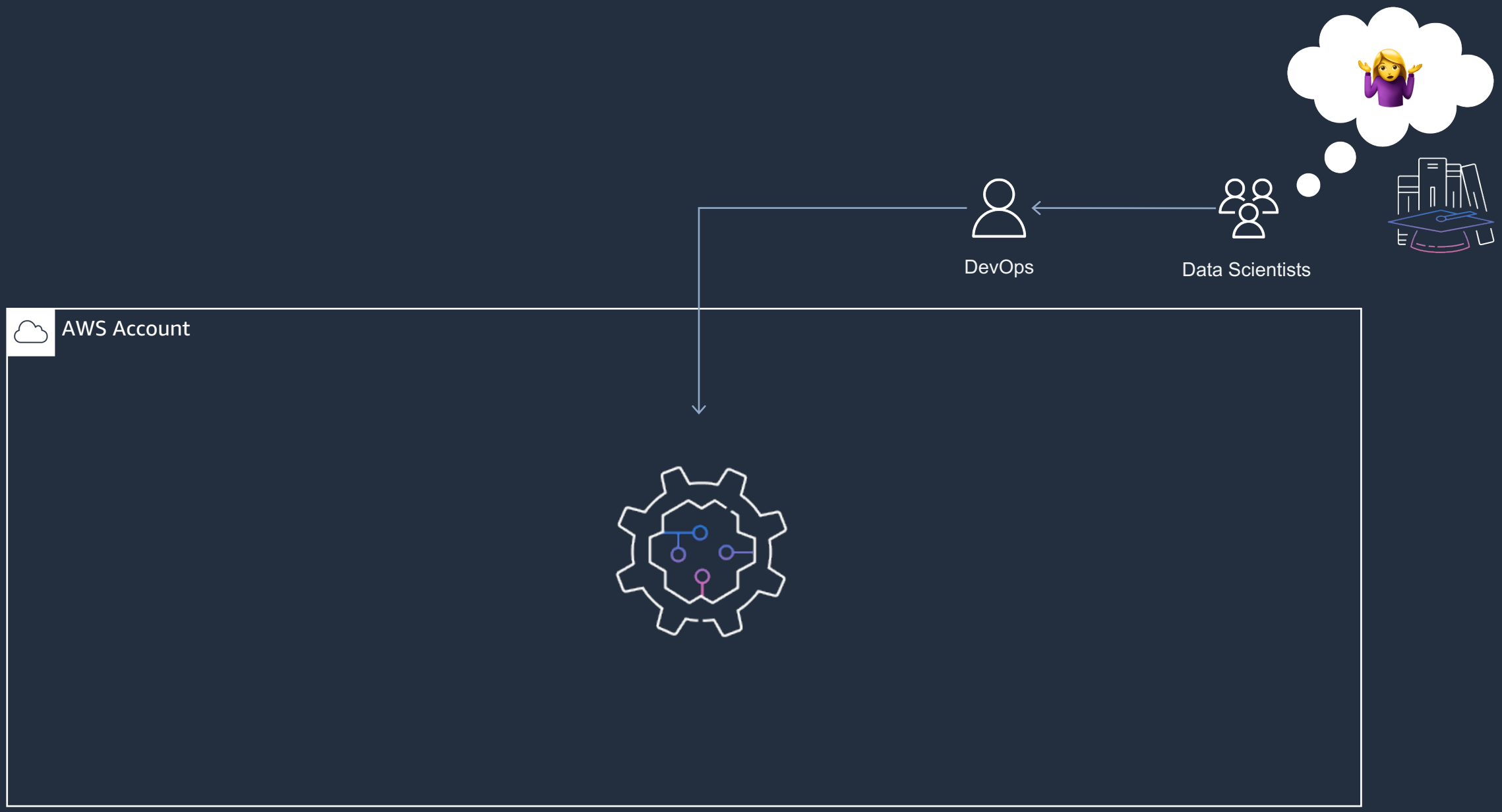
Senior Specialist Solutions Architect
AWS

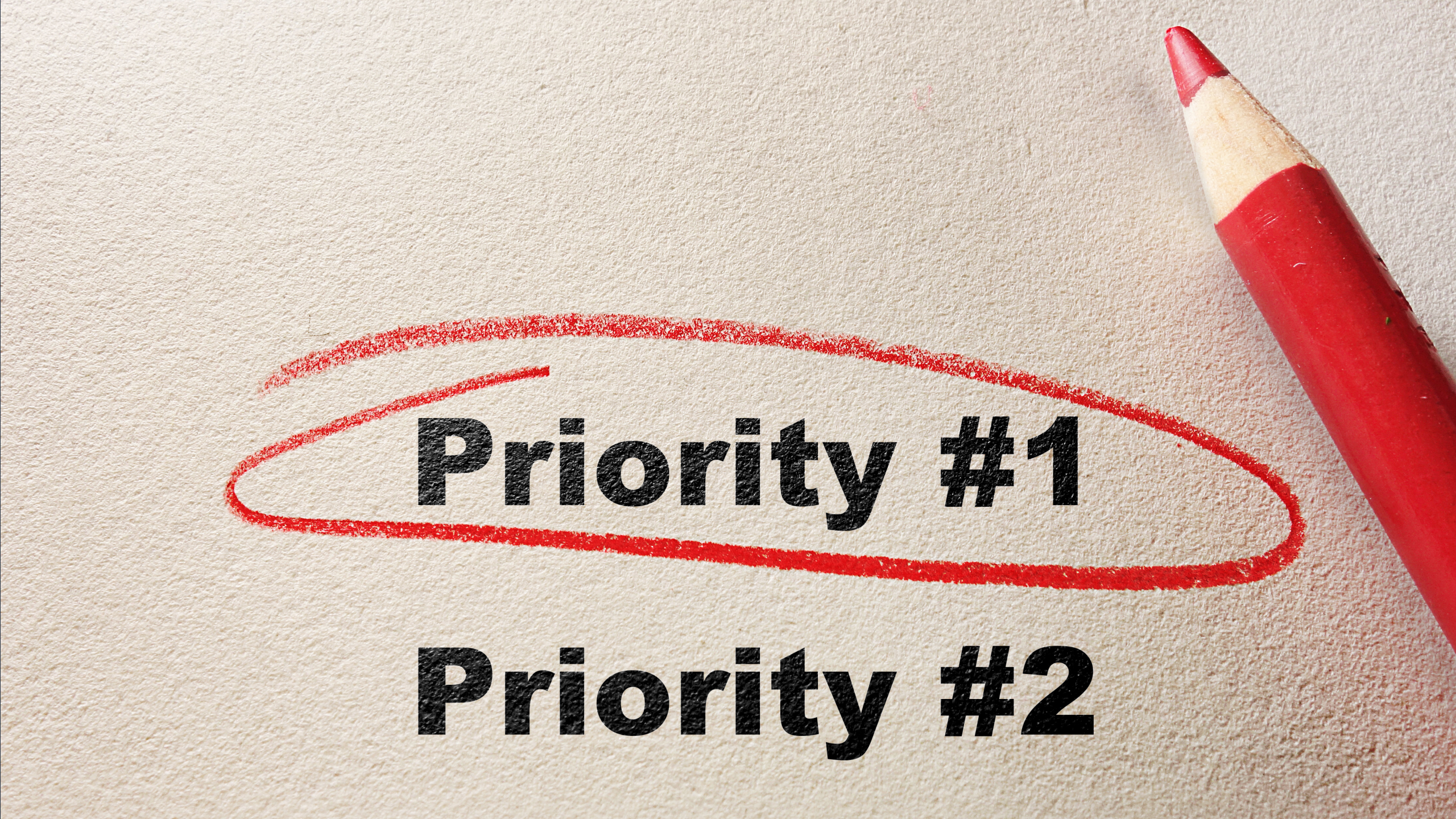

?
Chat GPT

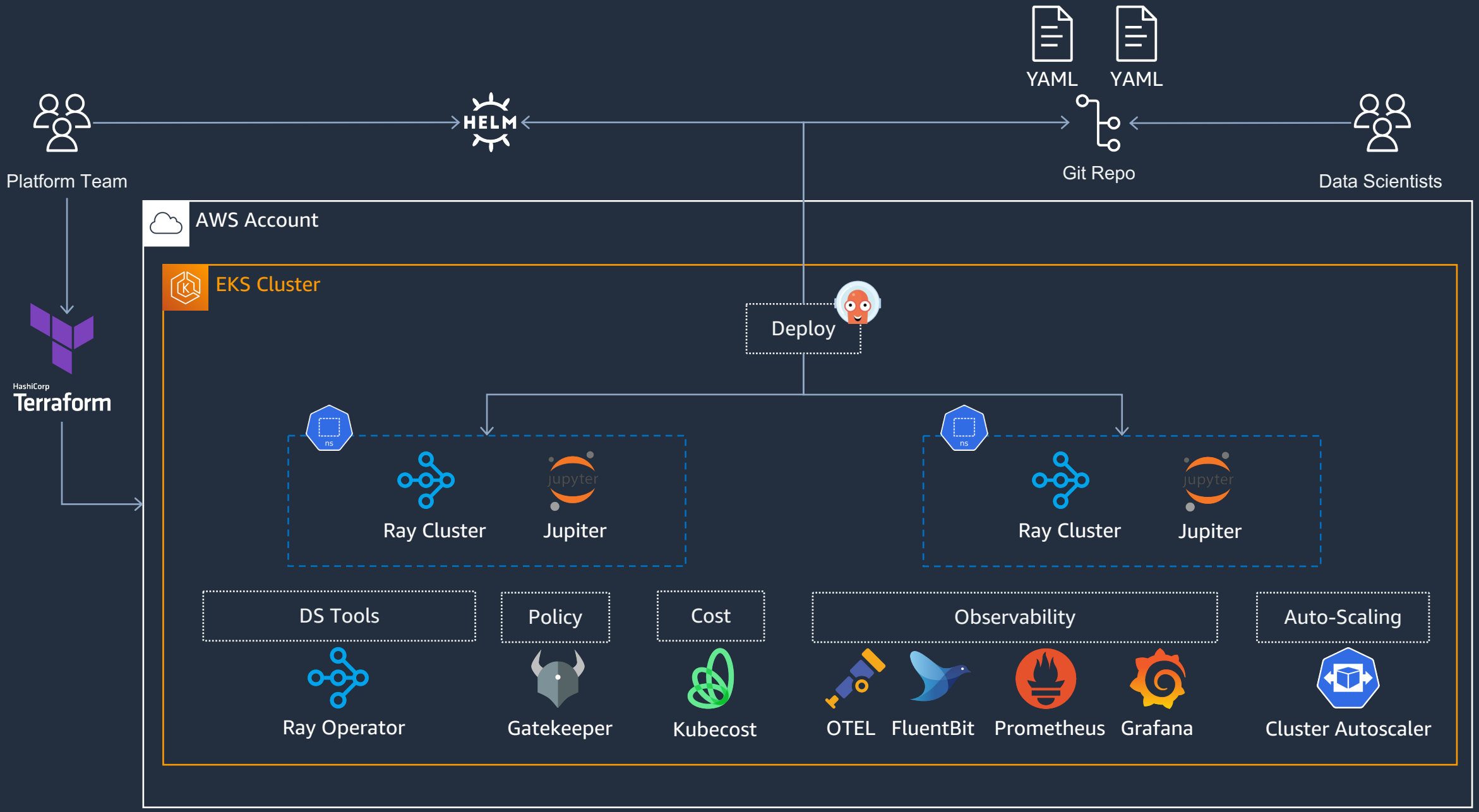
Data Scientist

AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

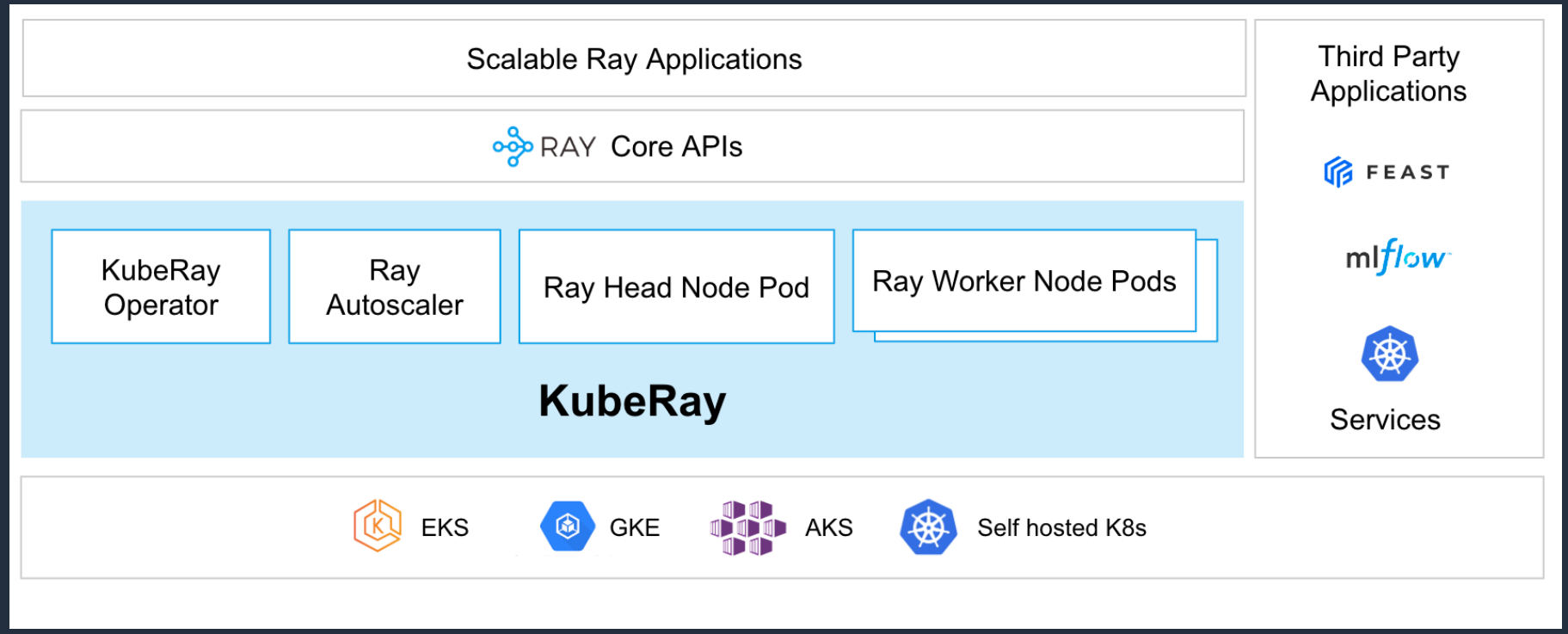
AWS Account




Priority #1
Priority #2



Ray on Kubernetes



Source: https://docs.ray.io/en/latest/cluster/kubernetes/index.html

Deploy Ray on Kubernetes

1. Deploy a KubeRay Operator

```
helm repo add kuberay https://ray-project.github.io/kuberay-helm/
helm repo update

# Install both CRDs and KubeRay operator v1.0.0.
helm install kuberay-operator kuberay/kuberay-operator --version 1.0.0

# Confirm that the operator is running in the namespace `default`.
kubectl get pods
# NAME                                READY   STATUS    RESTARTS   AGE
# kuberay-operator-7fbdbf8c89-pt8bk   1/1     Running   0          27s
```

2. Deploy a RayCluster custom resource

```
# Deploy a sample RayCluster CR from the KubeRay Helm chart repo:
helm install raycluster kuberay/ray-cluster --version 1.0.0

# Once the RayCluster CR has been created, you can view it by running:
kubectl get rayclusters

# NAME                 DESIRED WORKERS   AVAILABLE WORKERS   STATUS   AGE
# raycluster-kuberay   1                 1                   ready    72s
```

Source: https://docs.ray.io/en/latest/cluster/kubernetes/getting-started/raycluster-quick-start.html

Instance type

Accelerated Computing

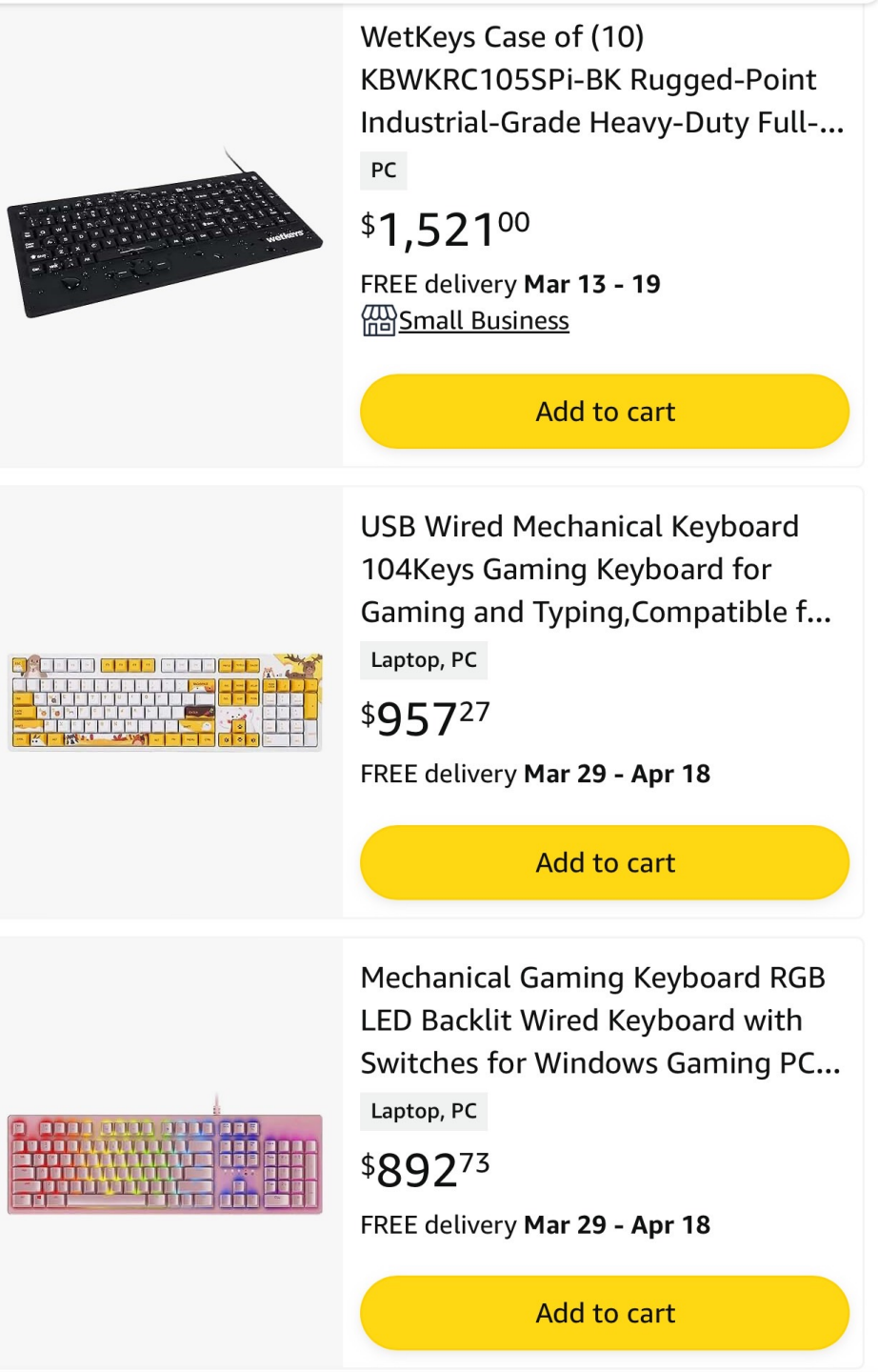
Viewing 43 of 698 available instances

Instance name	On-Demand hourly rate	vCPU
inf1.xlarge	$0.228	4
inf1.2xlarge	$0.362	8
g4ad.xlarge	$0.37853	4
g4dn.xlarge	$0.526	4
g4ad.2xlarge	$0.54117	8

How much is p5.48xlarge?

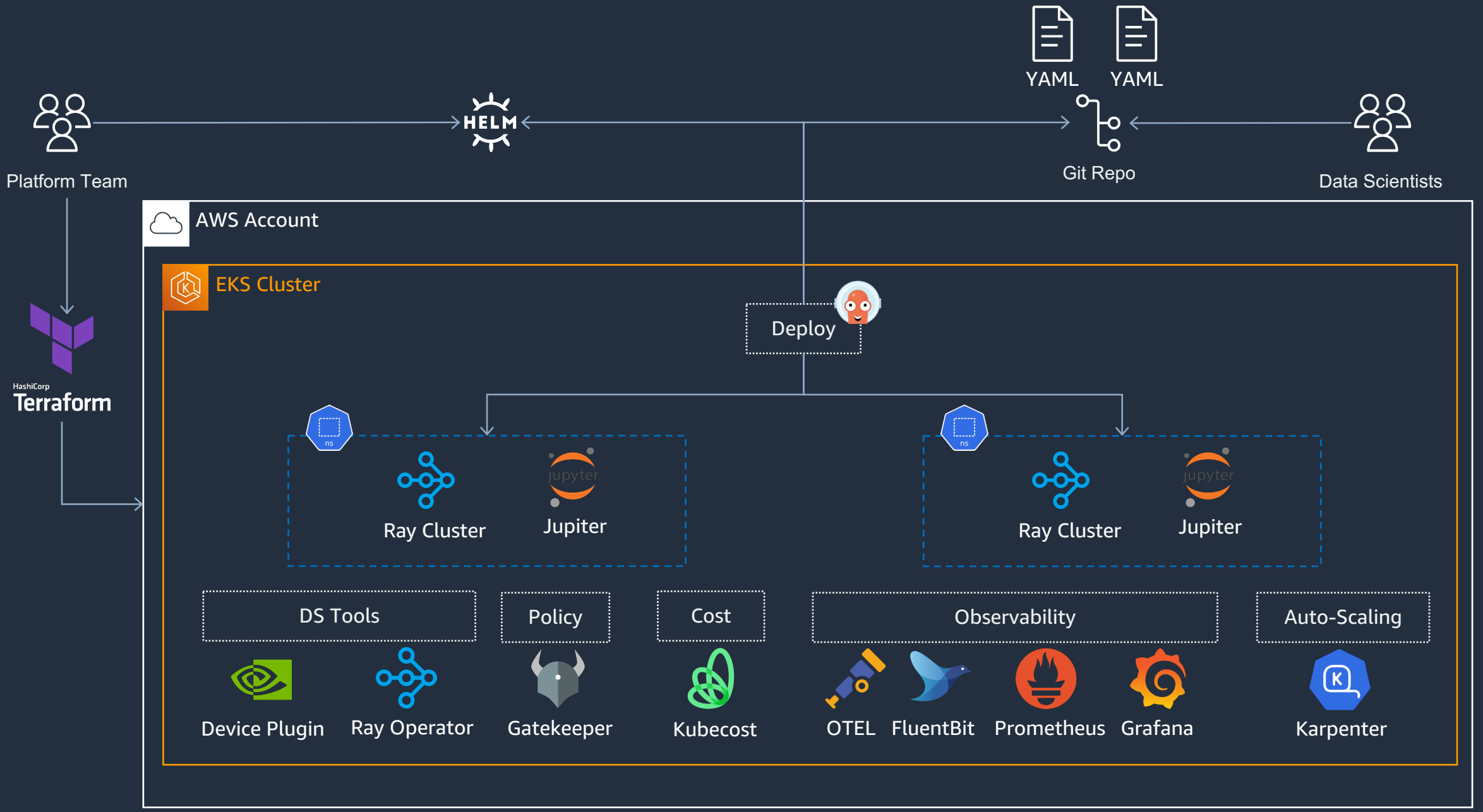
$98.32/hour

$70K/month



T-Shirt Size	Instance Size
M	g5.2xlarge
L	p4d.24xlarge
XL	p5.48xlarge

T-Shirt Size	Instance Size
M	g5.2xlarge g5.8xlarge g5.12xlarge
L	p4d.24xlarge
XL	p5.48xlarge



https://s12d.com/ca-to-karpenter

Karpenter

T-Shirt Size	Instance Size
M	g5.2xlarge g5.8xlarge g5.12xlarge
L	p4d.24xlarge
XL	p5.48xlarge

Karpenter

T-Shirt Size	Instance Size	NVIDIA Type
M	g5.2xlarge g5.8xlarge g5.12xlarge	A10G
L	p4d.24xlarge	H100
XL	p5.48xlarge	A100

Karpenter

T-Shirt Size	Instance Size
M	g5.2xlarge g5.8xlarge g5.12xlarge
L	p4d.24xlarge
XL	p5.48xlarge

M

```
{{- if eq .Values.experiment_size "medium" }}
apiVersion: karpenter.sh/v1beta1
kind: NodePool
spec:
...
    spec:
      requirements:
      - key: node.kubernetes.io/instance-type
        operator: In
        values: ["g5.4xlarge", "g5.8xlarge", "g5.12xlarge"]
      - key: karpenter.sh/capacity-type
        operator: In
        values: ["spot", "on-demand"]
{{- end }}
```

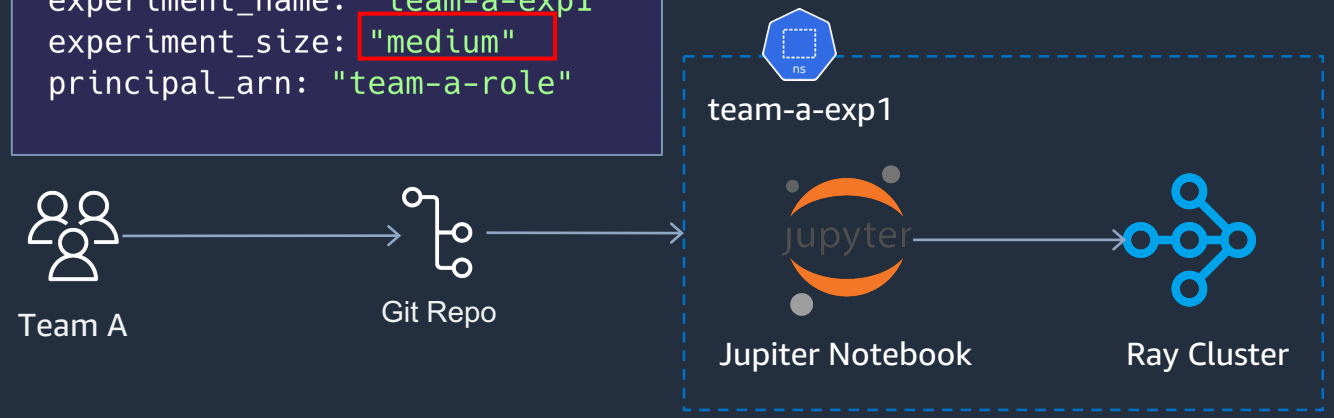
L

```
{{- if eq .Values.experiment_size "large" }}
apiVersion: karpenter.sh/v1beta1
kind: NodePool
spec:
...
    spec:
      requirements:
      - key: node.kubernetes.io/instance-type
        operator: In
        values: ["p4d.24xlarge"]
      - key: karpenter.sh/capacity-type
        operator: In
        values: ["spot", "on-demand"]
{{- end }}
```

XL

```
{{- if eq .Values.experiment_size "xlarge" }}
apiVersion: karpenter.sh/v1beta1
kind: NodePool
spec:
...
    spec:
      requirements:
      - key: node.kubernetes.io/instance-type
        operator: In
        values: ["p5.48xlarge"]
      - key: karpenter.sh/capacity-type
        operator: In
        values: ["spot", "on-demand"]
{{- end }}
```

```
experiment_name: "team-a-exp1"
experiment_size: "medium"
principal_arn: "team-a-role"
```

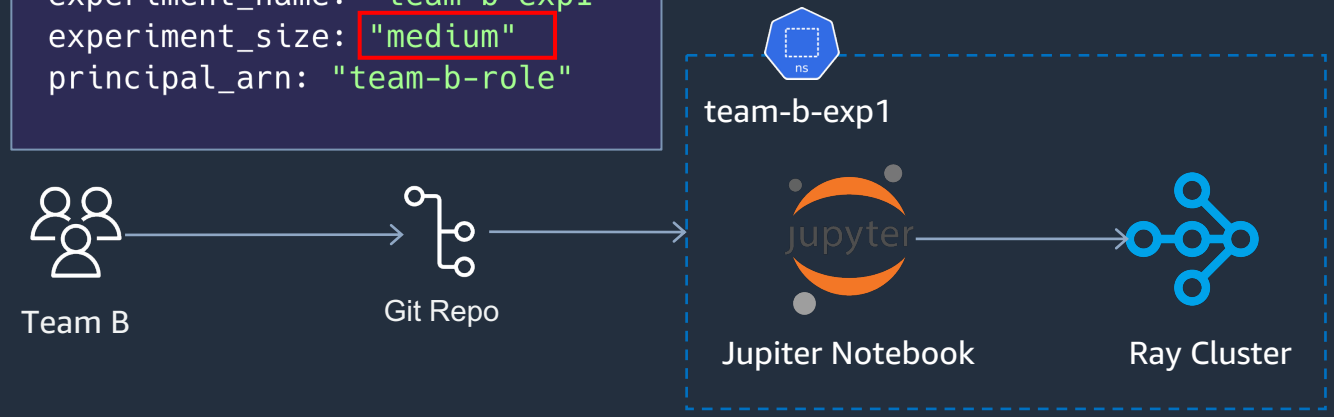



```
apiVersion: karpenter.sh/v1beta1
kind: NodePool
spec:
...
    spec:
      requirements:
      - key: node.kubernetes.io/instance-type
        operator: In
        values: ["g5.4xlarge", "g5.8xlarge", "g5.12xlarge"]
      - key: karpenter.sh/capacity-type
        operator: In
        values: ["spot", "on-demand"]
```

M

Karpenter

```
experiment_name: "team-b-exp1"
experiment_size: "medium"
principal_arn: "team-b-role"
```




```
apiVersion: karpenter.sh/v1beta1
kind: NodePool
spec:
...
    spec:
      requirements:
      - key: node.kubernetes.io/instance-type
        operator: In
        values: ["g5.4xlarge", "g5.8xlarge", "g5.12xlarge"]
      - key: karpenter.sh/capacity-type
        operator: In
        values: ["spot", "on-demand"]
```

M

Karpenter

```
experiment_name: "team-a-exp1"
experiment_size: "medium"
principal_arn: "team-a-role"
```

Team A

Git Repo

team-a-exp1

Jupiter Notebook

Ray Cluster

```
workerGroupSpecs:
...
  tolerations:
  - key: "experiment-name"
    operator: "Equal"
    value: team-a-exp1
    effect: "NoSchedule"
```

```
r.sh/v1beta1
```

M

Karpenter

```
bernetes.io/instance-type
                                  "g5.4xlarge", "g5.8xlarge", "g5.12xlarge"]
  - key: karpenter.sh/capacity-type
    operator: In
    values: ["spot", "on-demand"]
taints:
- key: experiment-name
  value: team-a-exp1
  effect: NoSchedule
```

```
experiment_name: "team-b-exp1"
experiment_size: "medium"
principal_arn: "team-b-role"
```

Team B

Git Repo

team-b-exp1

Jupiter Notebook

Ray Cluster

```
workerGroupSpecs:
...
  tolerations:
  - key: "experiment-name"
    operator: "Equal"
    value: team-b-exp1
    effect: "NoSchedule"
```

```
r.sh/v1beta1
```

M

Karpenter

```
bernetes.io/instance-type
                                  "g5.4xlarge", "g5.8xlarge", "g5.12xlarge"]
  - key: karpenter.sh/capacity-type
    operator: In
    values: ["spot", "on-demand"]
taints:
- key: experiment-name
  value: team-b-exp1
  effect: NoSchedule
```

```python
# Specify required resources for an actor.
@ray.remote(num_cpus=28, num_gpus=4)
class Actor:
  pass
```

```yaml
workerGroupSpecs:
...
  resources:
    limits:
      cpu: "7"
      memory: "30G"
      nvidia.com/gpu: 1
```

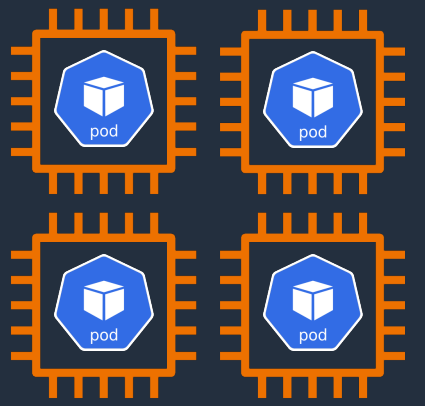
	Instance	GPU	CPU	Mem
M	g5.2xlarge	1	8	32
	g5.8xlarge	1	32	128
	g5.12xlarge	4	64	192

Team A

Jupiter Notebook

Ray Cluster

Karpenter NodePool



g5.2xlarges

```yaml
workerGroupSpecs:
- replicas: 0
  minReplicas: 0
  maxReplicas: 10
```

```yaml
workerGroupSpecs:
...
  resources:
    limits:
      cpu: "7"
      memory: "30G"
      nvidia.com/gpu: 1
```

```python
# Specify required resources for an actor.
@ray.remote(num_cpus=28, num_gpus=4)
class Actor:
  pass
```

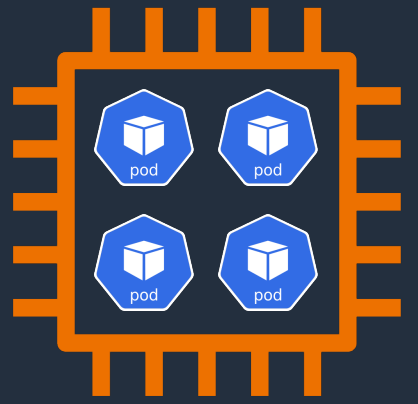
	Instance	GPU	CPU	Mem
M	g5.2xlarge	1	8	32
	g5.8xlarge	1	32	128
	g5.12xlarge	4	64	192

Team A

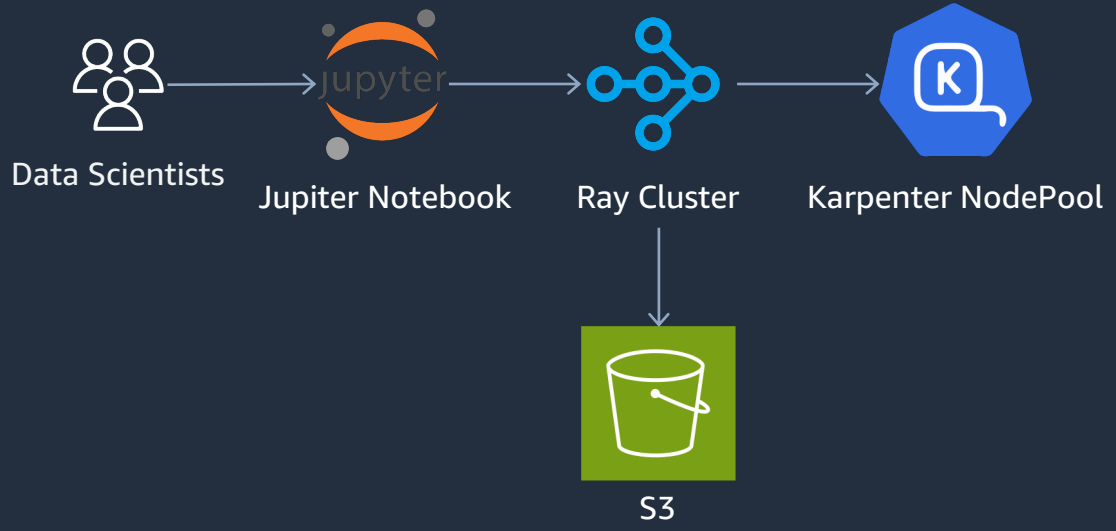
Jupiter Notebook

Ray Cluster

Karpenter NodePool



g5.12xlarge



AWS

EKS Cluster

ns

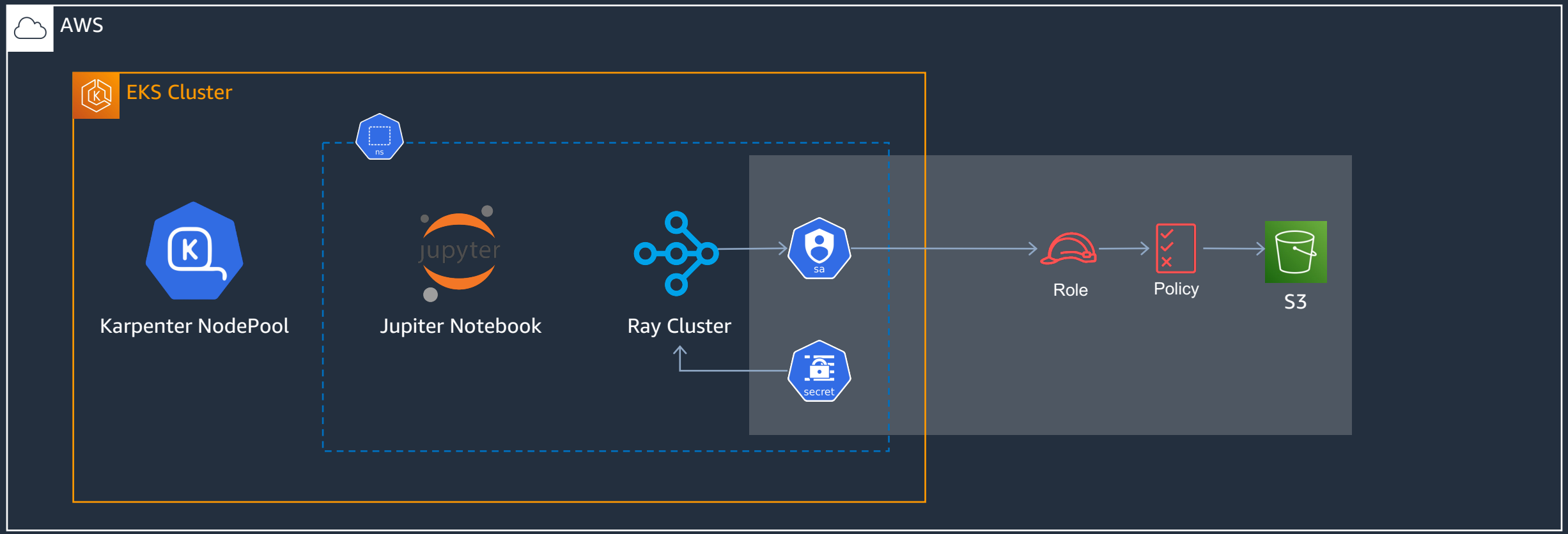
Karpenter NodePool

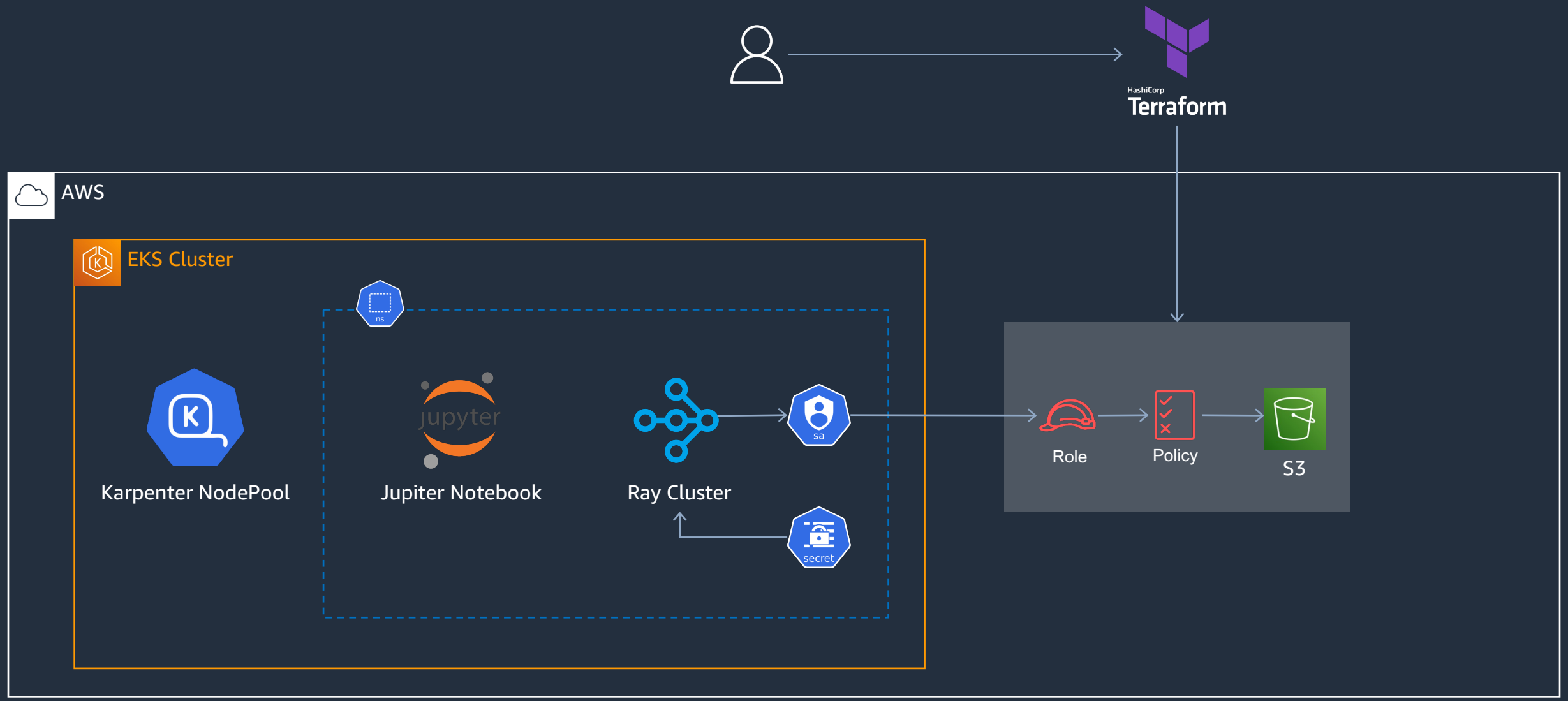
jupyter

Jupiter Notebook

Ray Cluster

S3

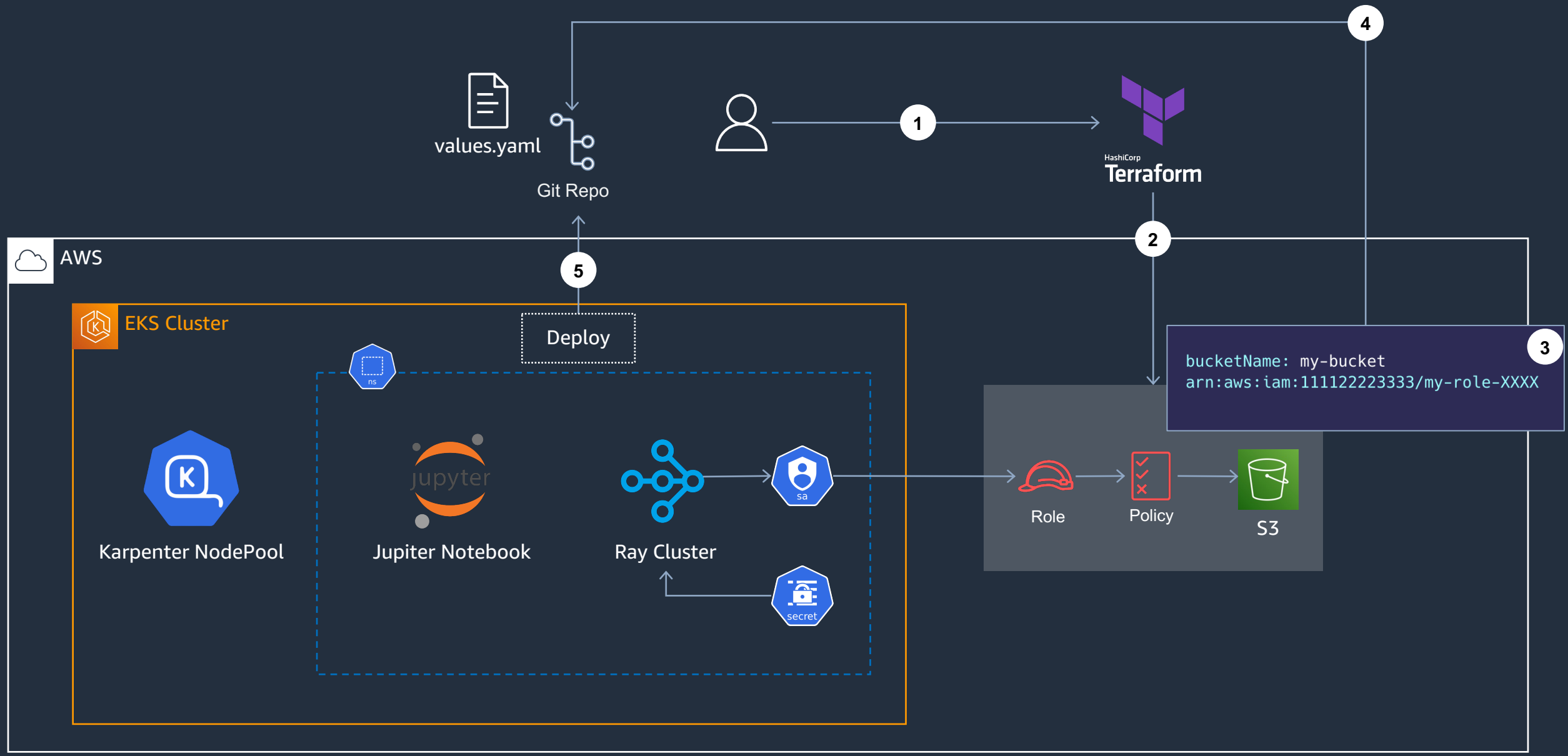


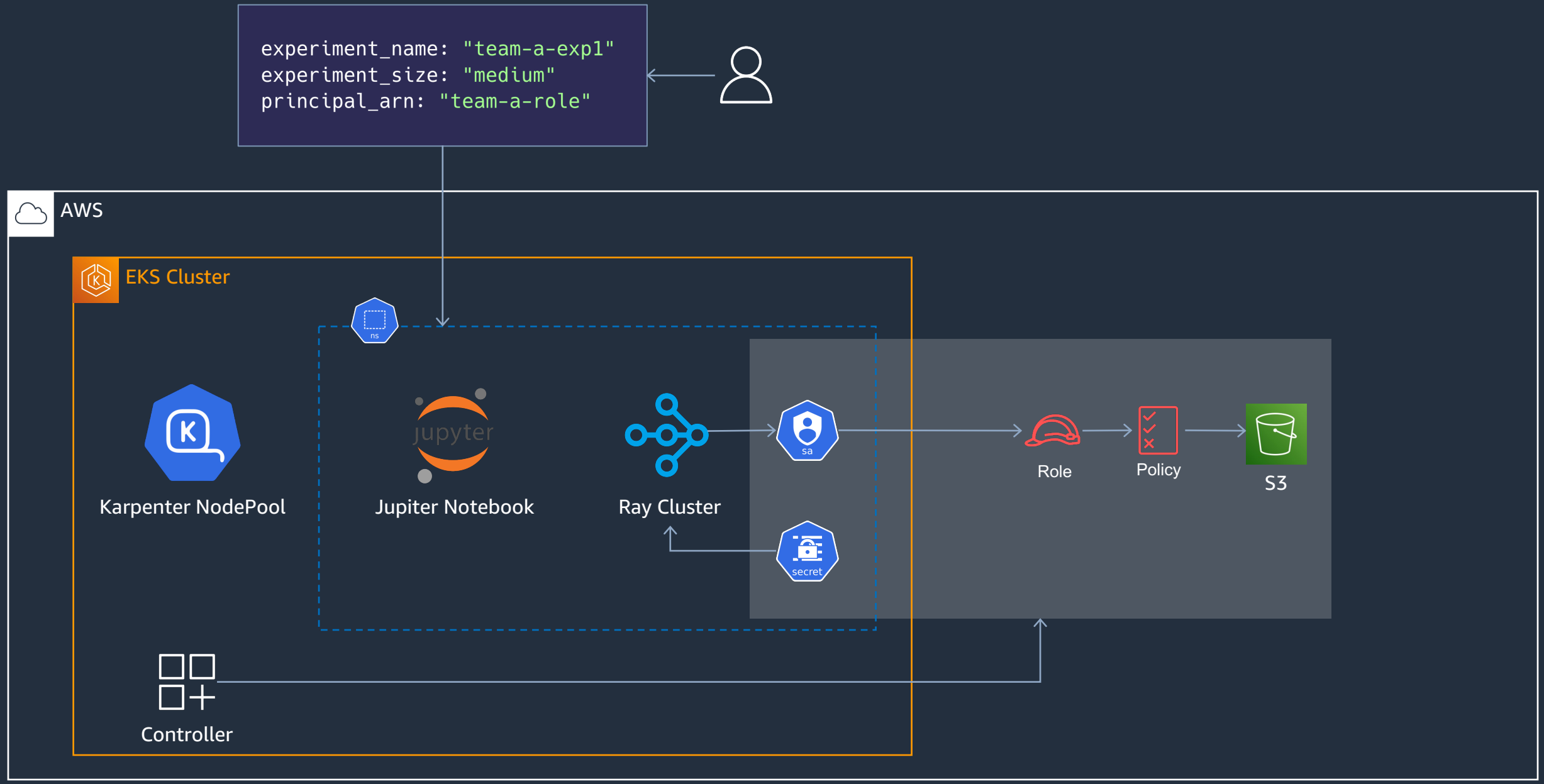


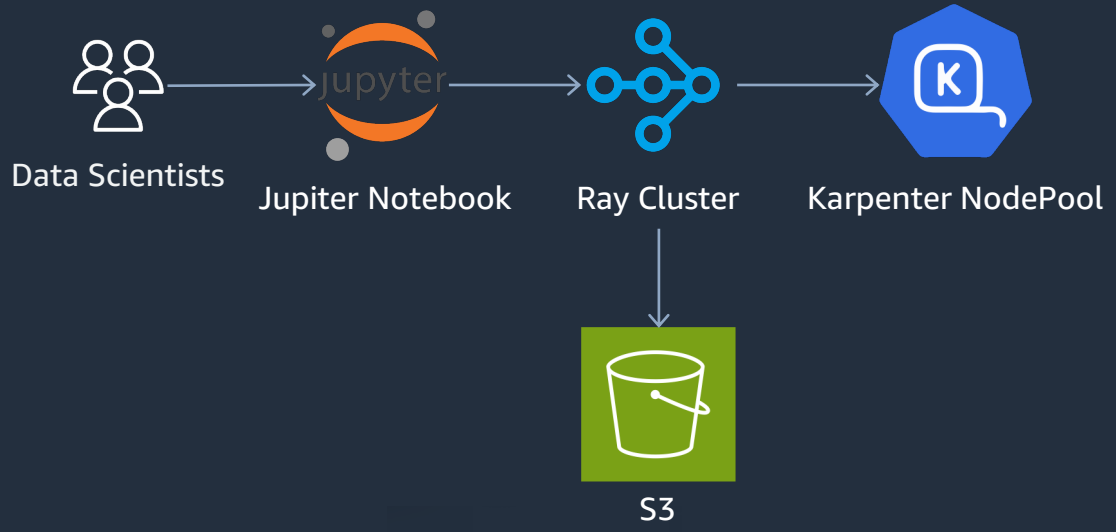

```hcl
resource "aws_s3_bucket" "my_bucket" {
  bucket_prefix = "my-bucket-"

  tags = local.tags
}

resource "aws_s3_bucket_public_access_block" "my_bucket" {
  bucket = aws_s3_bucket.my_bucket.id

  block_public_acls       = true
  block_public_policy     = true
  ignore_public_acls      = true
  restrict_public_buckets = true
}

resource "aws_s3_bucket_lifecycle_configuration" "my_bucket" {
  bucket = aws_s3_bucket.my_bucket.id
  rule {
    id = "some-data"
    expiration {
      days = 7
    }
    status = "Enabled"
  }
}

resource "aws_s3_bucket_ownership_controls" "my_bucket" {
  bucket = aws_s3_bucket.my_bucket.id

  rule {
    object_ownership = "BucketOwnerPreferred"
  }
}

resource "aws_s3_bucket_acl" "my_bucket" {
  depends_on = [aws_s3_bucket_ownership_controls.my_bucket]

  bucket = aws_s3_bucket.my_bucket.id
  acl    = "private"
}
```

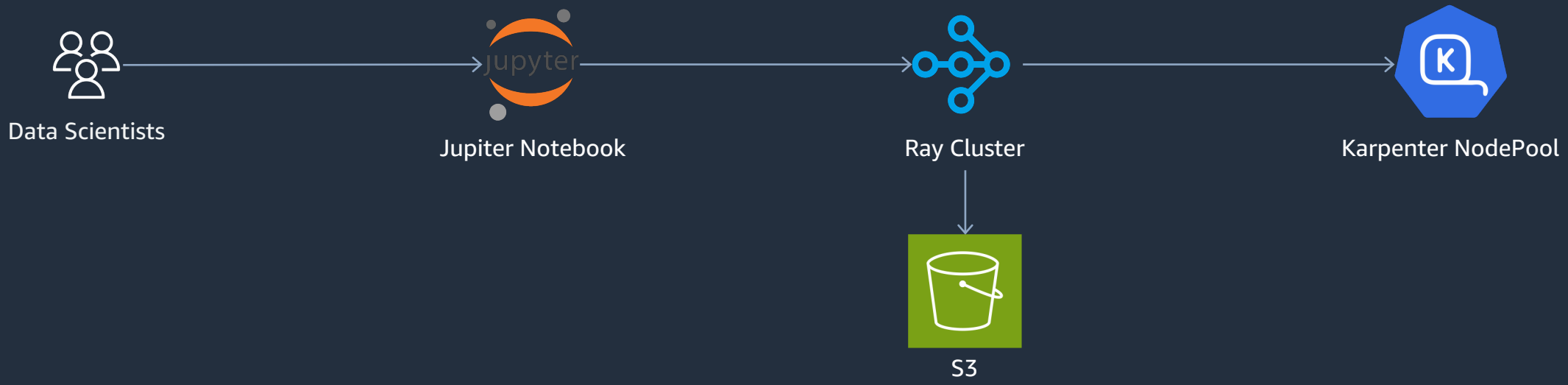

```hcl
data "aws_iam_policy_document" "assume_policy" {
  statement {
    effect = "Allow"
    actions = ["sts:AssumeRoleWithWebIdentity"]

    principals {
      type        = "Federated"
      identifiers = ["arn:aws:iam::111122223333:oidc-
provider/oidc.eks.us-east-1.amazonaws.com/id/1234567890ABC"]
    }

    condition = {
      content {
        test     = "StringLike"
        values   = [...]
      }
    }
  }
}

data "aws_iam_policy_document" "my_policy" {
  statement {
    sid    = "S3ReadAccessToBucket"
    effect = "Allow"
    actions = [
      "s3:ListBucket",
      "s3:List*",
      "s3:Get*"
    ]
    resources = [
      aws_s3_bucket.my_bucket.arn,
      "${aws_s3_bucket.my_bucket.arn}/*"
    ]
  }
}

resource "aws_iam_policy" "my_policy" {
  name   = "my-policy"
  policy = data.aws_iam_policy_document.my_policy.json
}

resource "aws_iam_role" "my_role" {
  name               = "my-role"
  assume_role_policy = data.aws_iam_policy_document.assume_role.json
}

resource "aws_iam_role_policy_attachment" "policy-attach" {
  role       = aws_iam_role.my_role.name
  policy_arn = aws_iam_policy.my_policy.arn
}
```

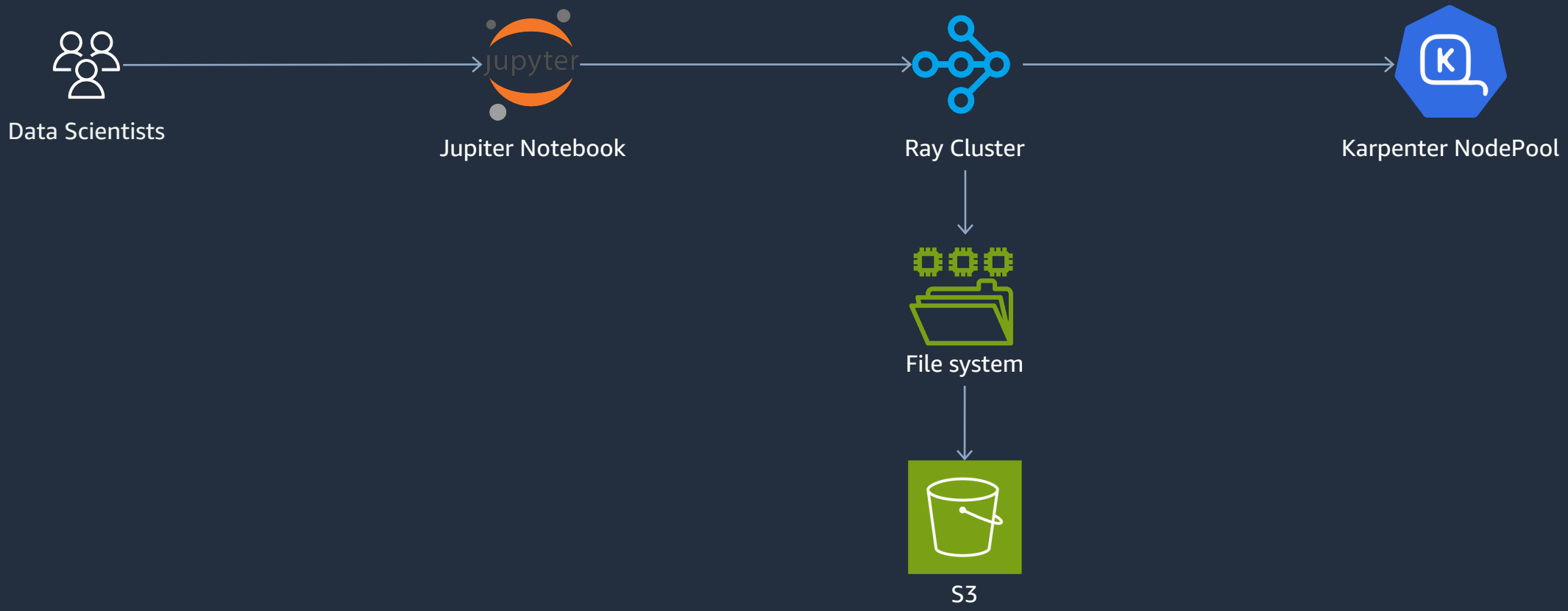

```
module "s3" {
  source  = "my-org-tf/modules/s3"
  version = "1.15.1"

  bucket_prefix          = "my-bucket-"
  lifecycle_expiration = 7
  object_ownership       = "bucketOwnerPreferred"
  acl                    = "private"
  ...
}
```



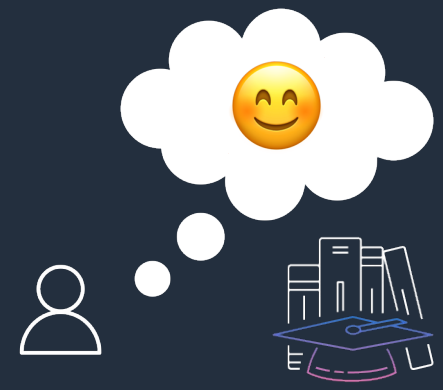






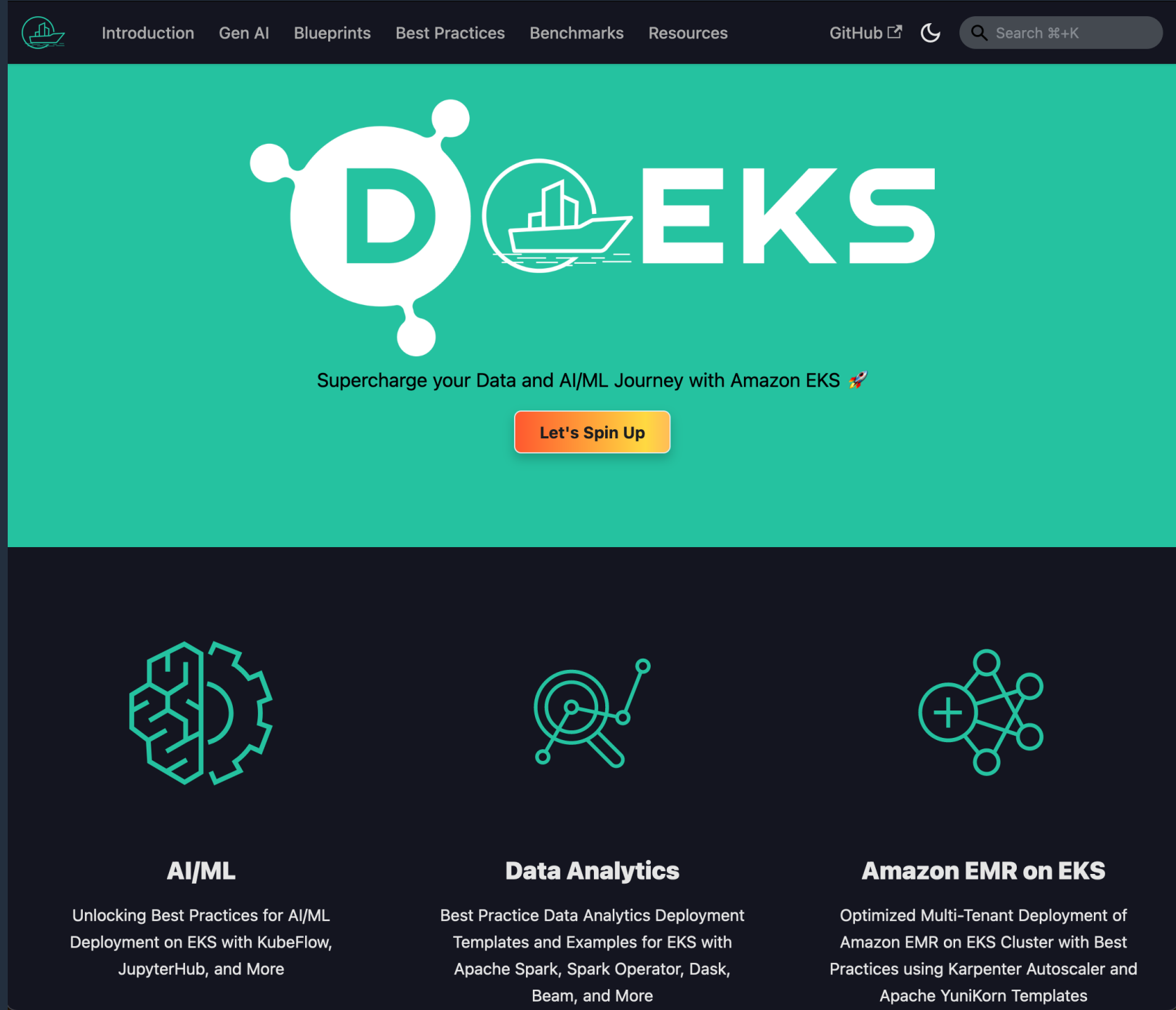


Demo



Data Scientist

https://s12d.com/DoEKS



Thank you!

Christina Andonov
candonov@amazon.com
christina-andonov

Apoorva Kulkarni
kuapoorv@amazon.com
askulkarni-aws