

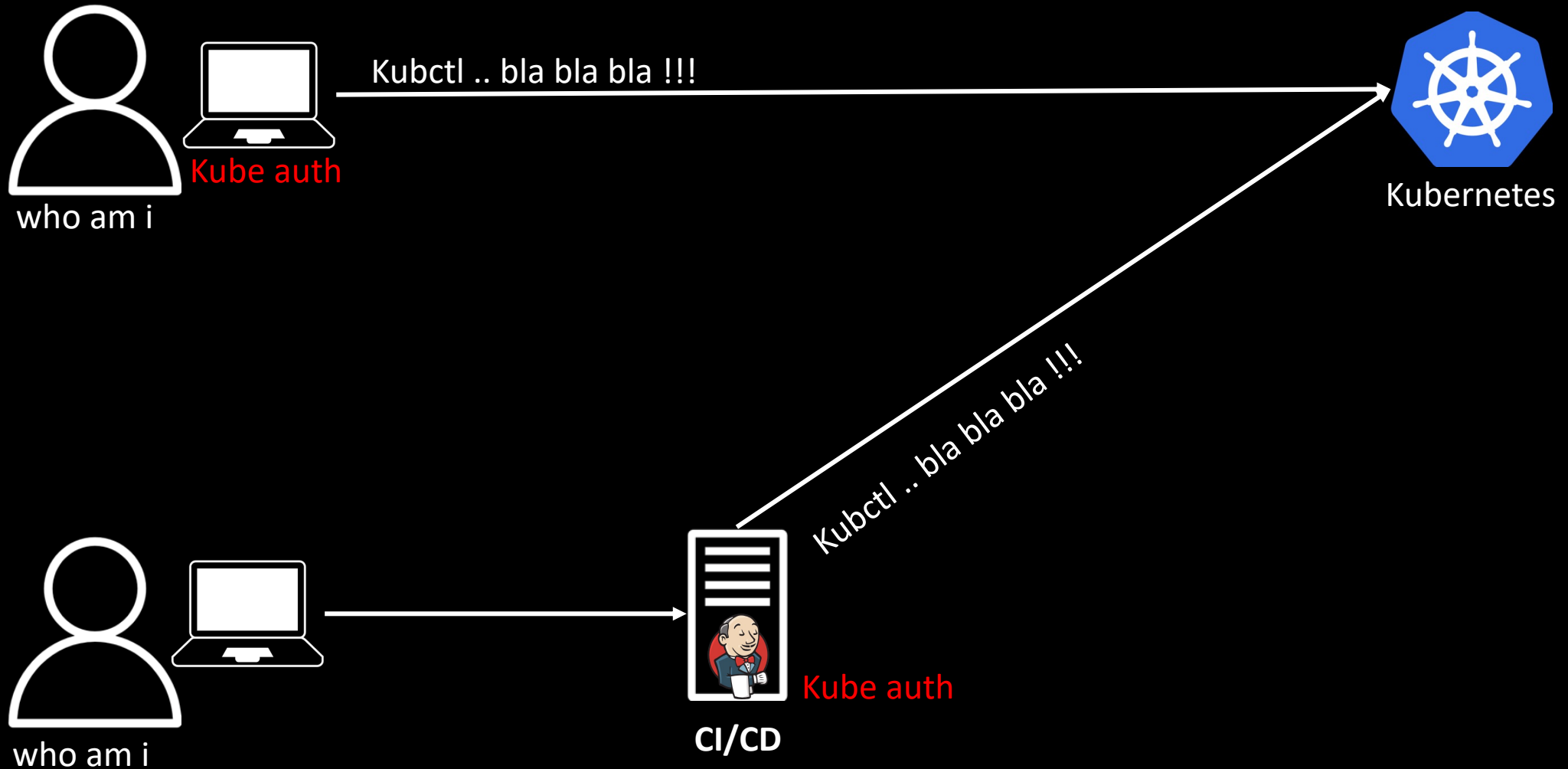
Zero-Trust and Secret Management for Applications

The logo features a large yellow circle with a white ring inside. The word "nimble" is written in white lowercase letters, with the "i" in yellow. Below it, "by krungsri" is written in smaller white lowercase letters.

nimble
by krungsri

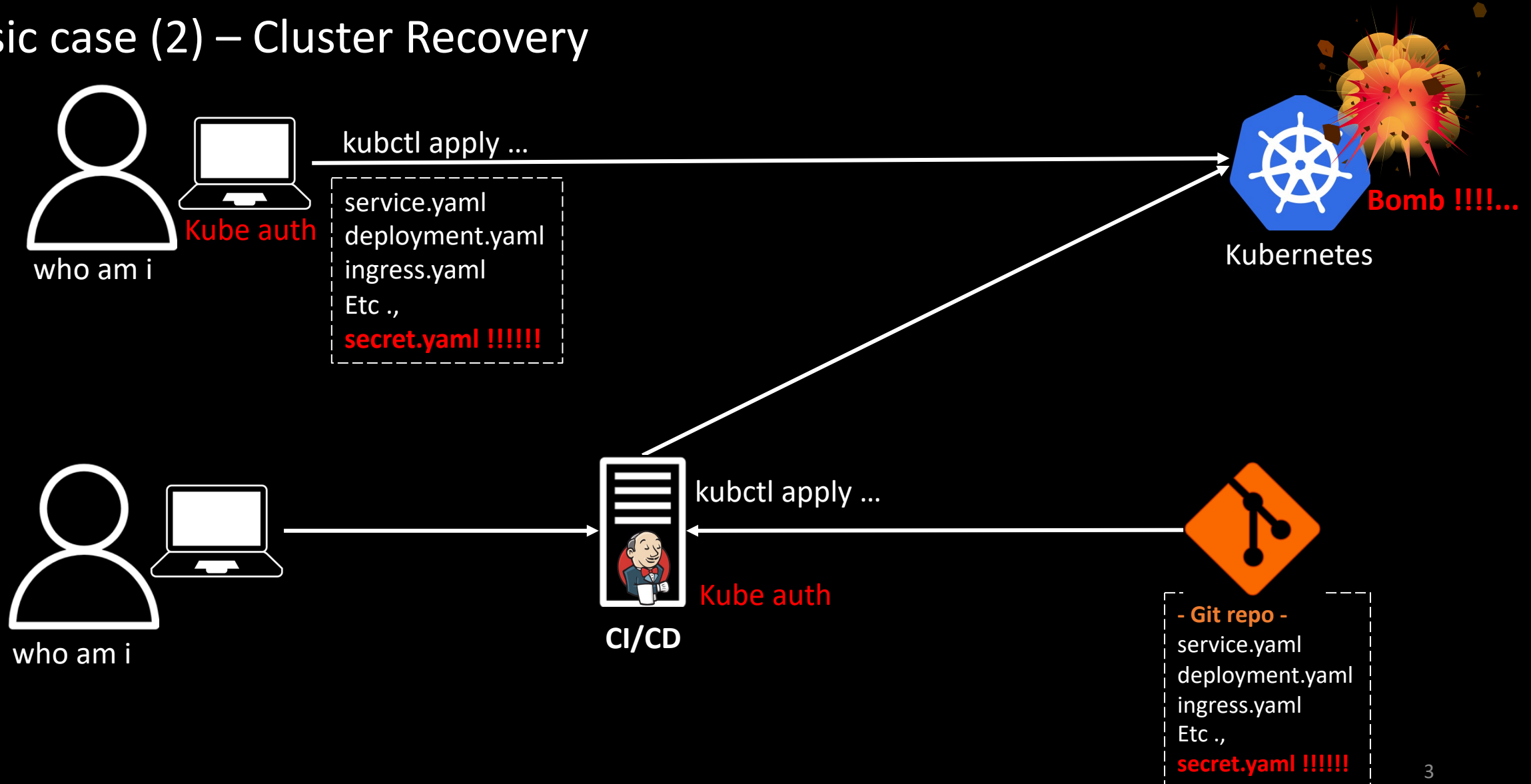
What's the problem definition?

Basic case (1) – Kube Control



What's the problem definition?

Basic case (2) – Cluster Recovery



What's the problem definition?

Basic case (3) – Apply Kube Secret



Secret.yaml

```
apiVersion: v1
kind: Secret
metadata:
  name: mysecret
type: Opaque
data:
  USERNAME : YWRtaW4= #admin
  PASSWORD : UEBzc3cwcmQ= #P@ssw0rd
```

kubectl apply -f secret.yaml








Kubernetes

What's the problem definition?

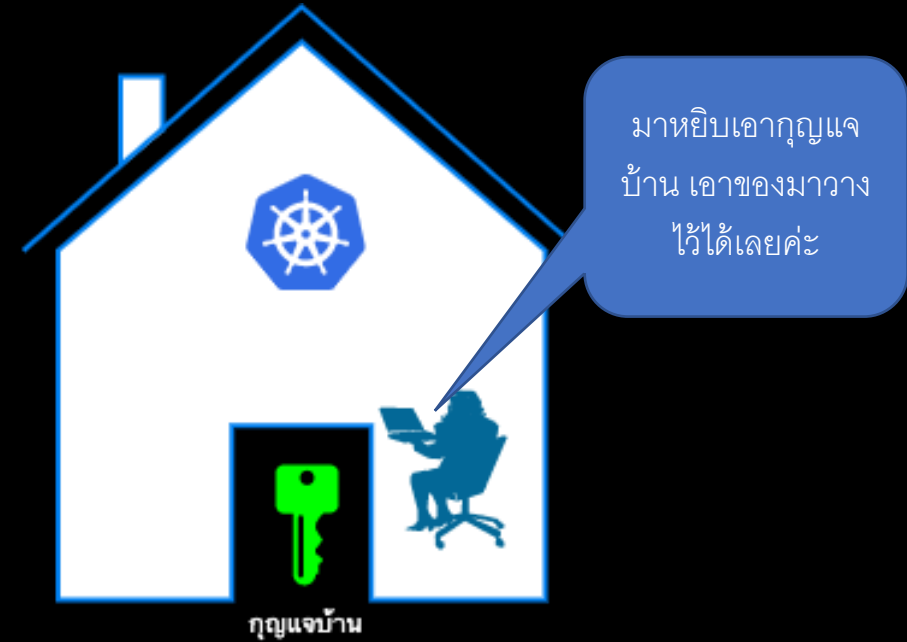
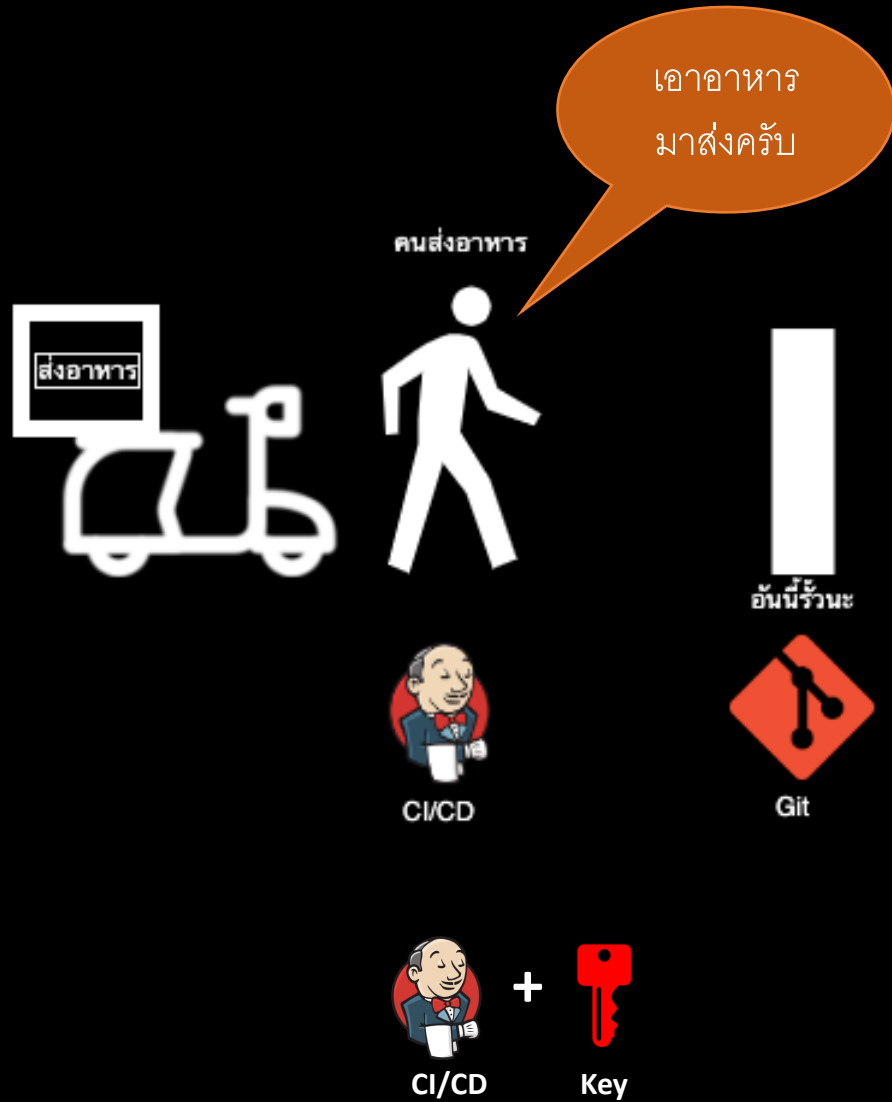
Basic case (4) – Send password (**Assume)



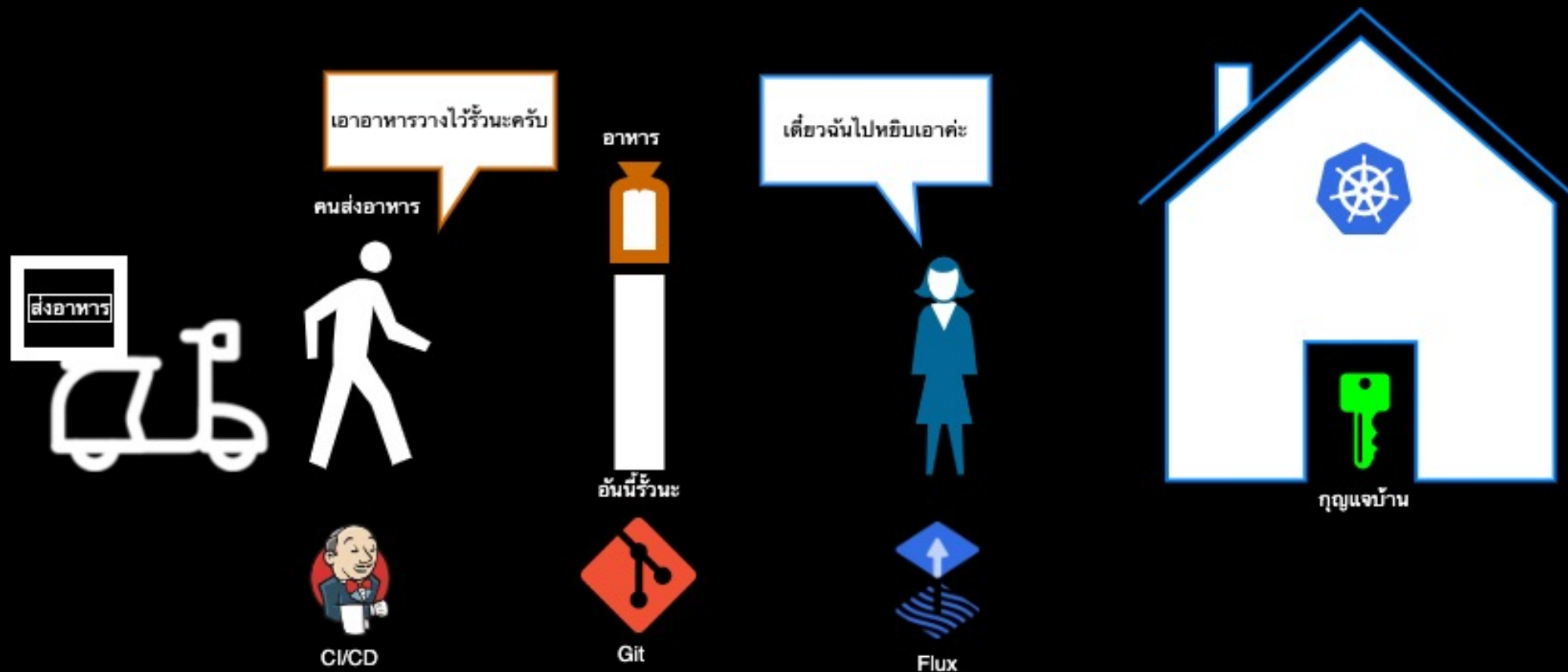
PPT Framework

People	Process	Technology
<ul style="list-style-type: none">• “ I am Nimble ”• Administrator• Production support• etc.,	<ul style="list-style-type: none">• Solution Architech Review• Change Management• Approval process• etc.,	<ul style="list-style-type: none">• Flux – GitOps (Flux has been promoted to Graduated status in the CNCF)• Amazon Web Services - Infrastructure• Kubernetes - Container Orchestration.• KMS – Key management• SOPS , AWS Secret – Secret management. <div></div> <div>FluxKubernetesEKSKMSSecret Manager</div>

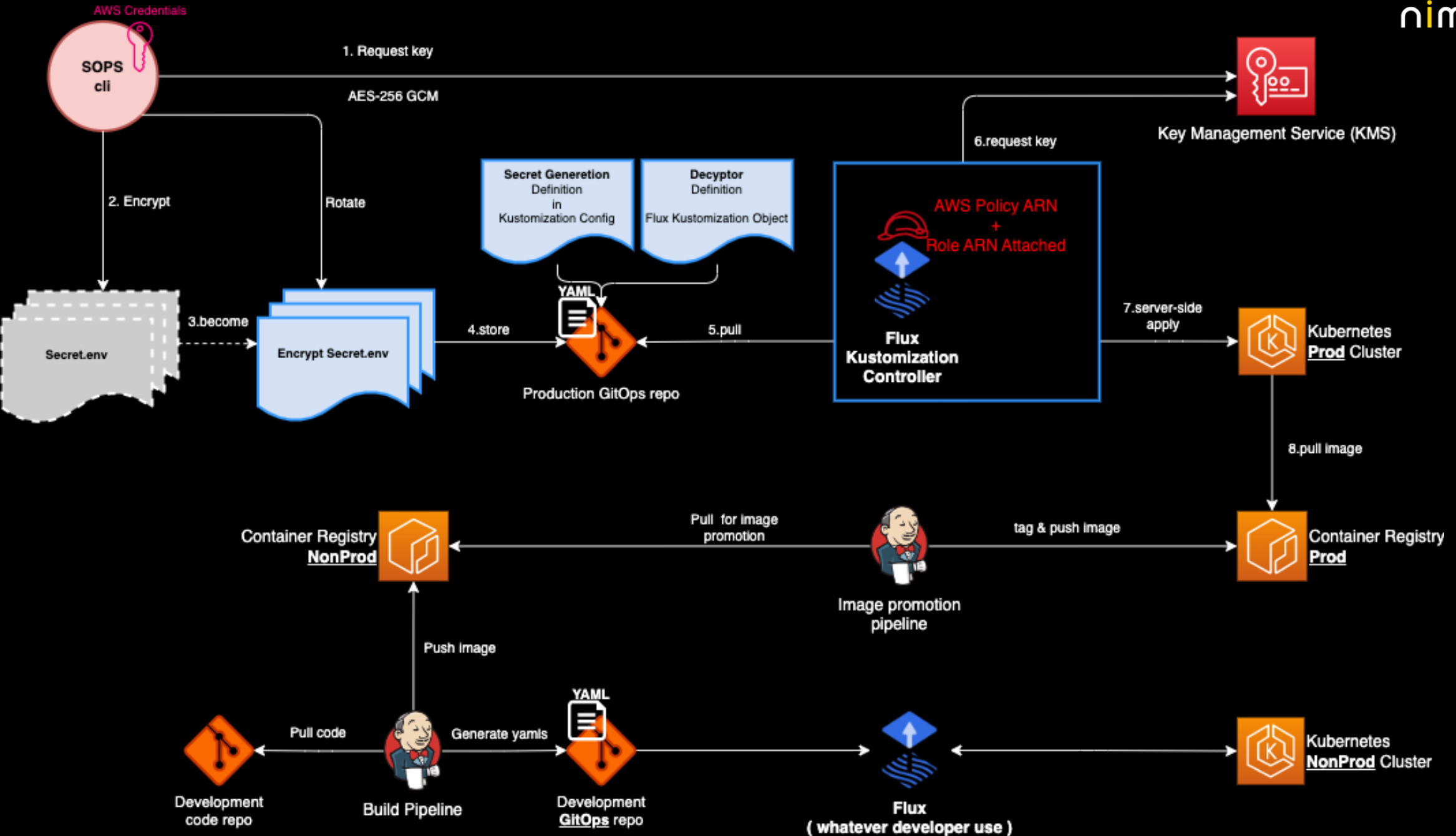
Zero Trust Concept



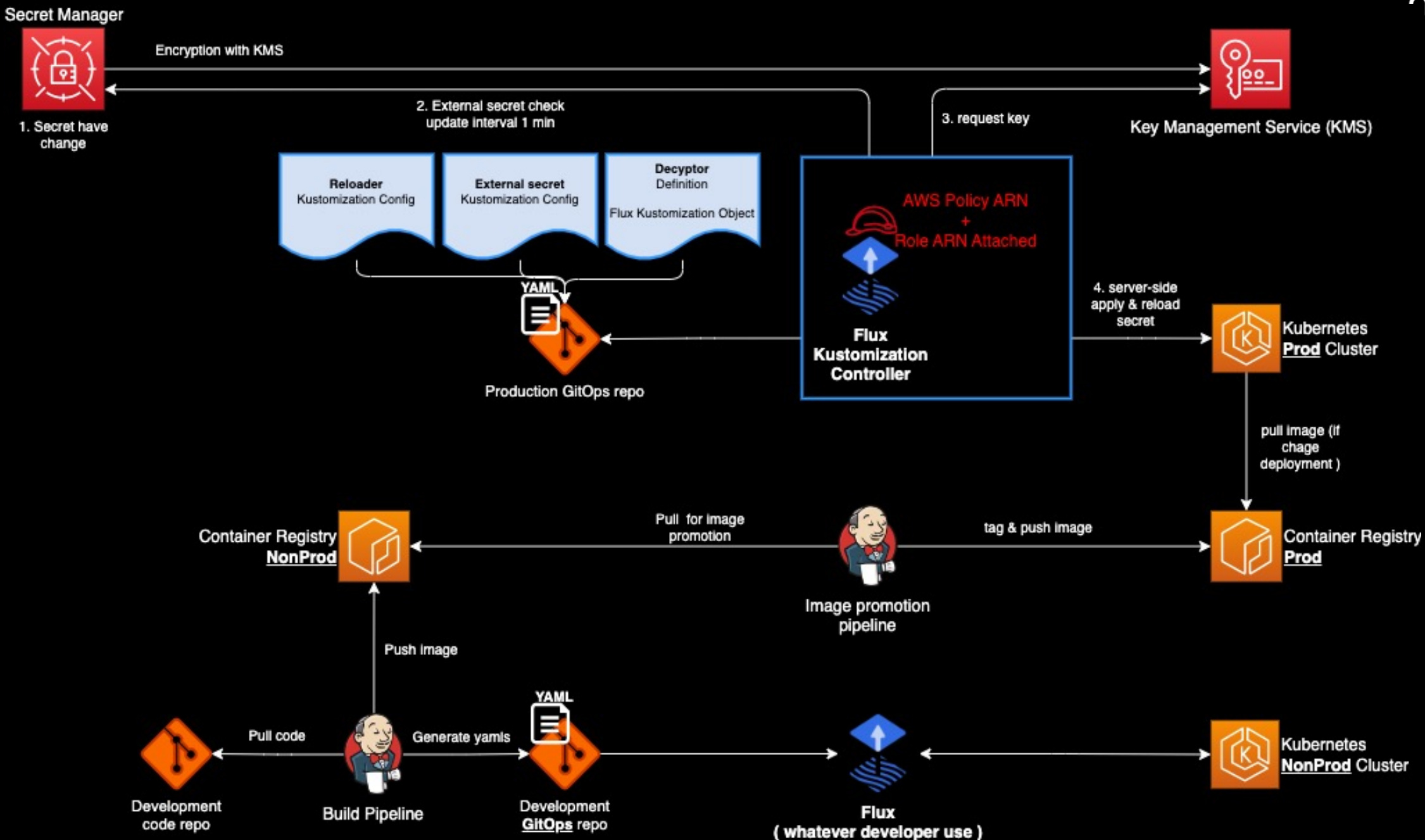
Zero Trust Concept



Solution (1) - Flux with SOPS CLI – ** Recovery **



Solution (2) - Flux with AWS Secret



How to setup **AWS** ?

```
aws kms create-key --description "jiw-flux-key-1"
```

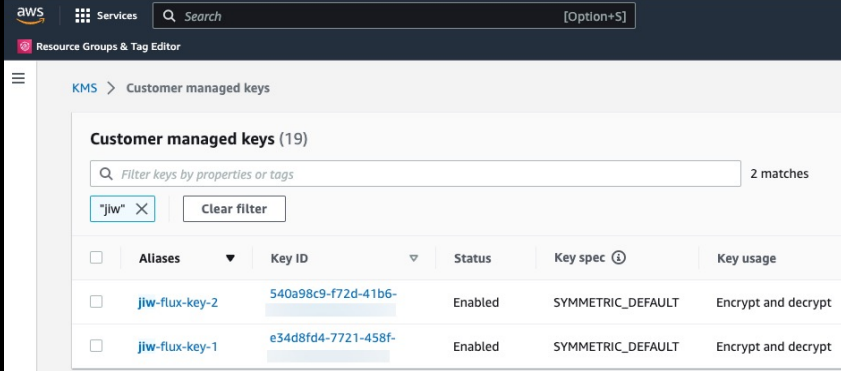
```
aws kms create-key --description "jiw-flux-key-2"
```

```
eksctl utils enable-secrets-encryption --cluster=jiw-flux --key-arn='{ jiw-flux-key-1 }'  
-- ** Wait 45 Minute
```

```
eksctl utils associate-iam-oidc-provider --cluster= jiw-flux
```

```
eksctl utils associate-iam-oidc-provider --cluster=jiw-eks-cluster --approve
```

```
aws iam create-policy --policy-name kustomization-controller --policy-document '{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Action": [  
        "kms:Decrypt",  
        "kms:DescribeKey"  
      ],  
      "Effect": "Allow",  
      "Resource": "{ jiw-flux-key-2 }" }  
    ] }'
```



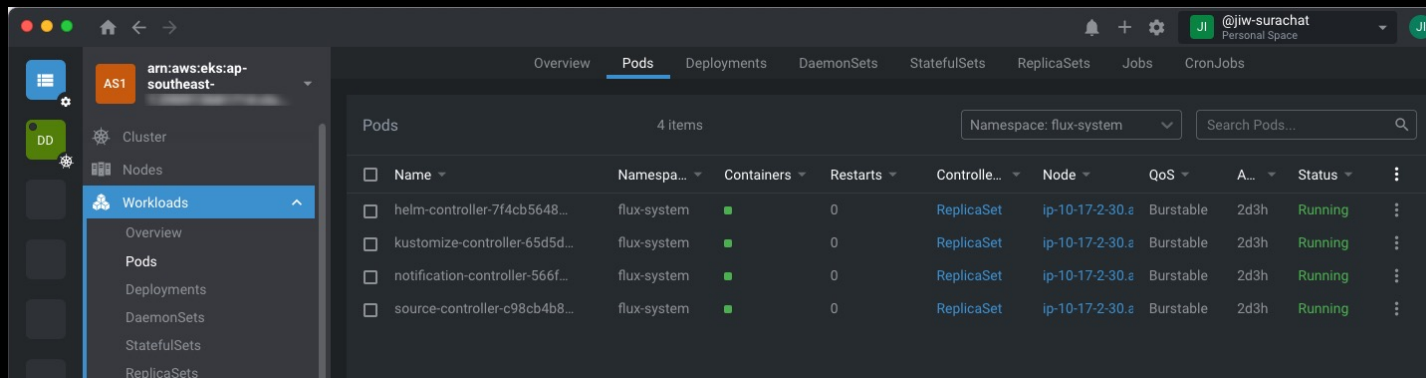
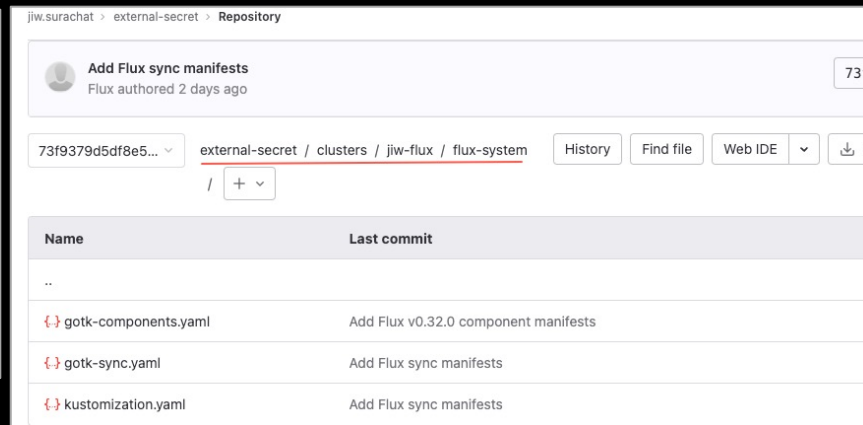
The screenshot shows the AWS Management Console interface for KMS Customer managed keys. A search filter 'jiw' is applied, resulting in 2 matches. The table below lists the keys.

<input type="checkbox"/>	Aliases	Key ID	Status	Key spec	Key usage
<input type="checkbox"/>	jiw-flux-key-2	540a98c9-f72d-41b6-	Enabled	SYMMETRIC_DEFAULT	Encrypt and decrypt
<input type="checkbox"/>	jiw-flux-key-1	e34d8fd4-7721-458f-	Enabled	SYMMETRIC_DEFAULT	Encrypt and decrypt

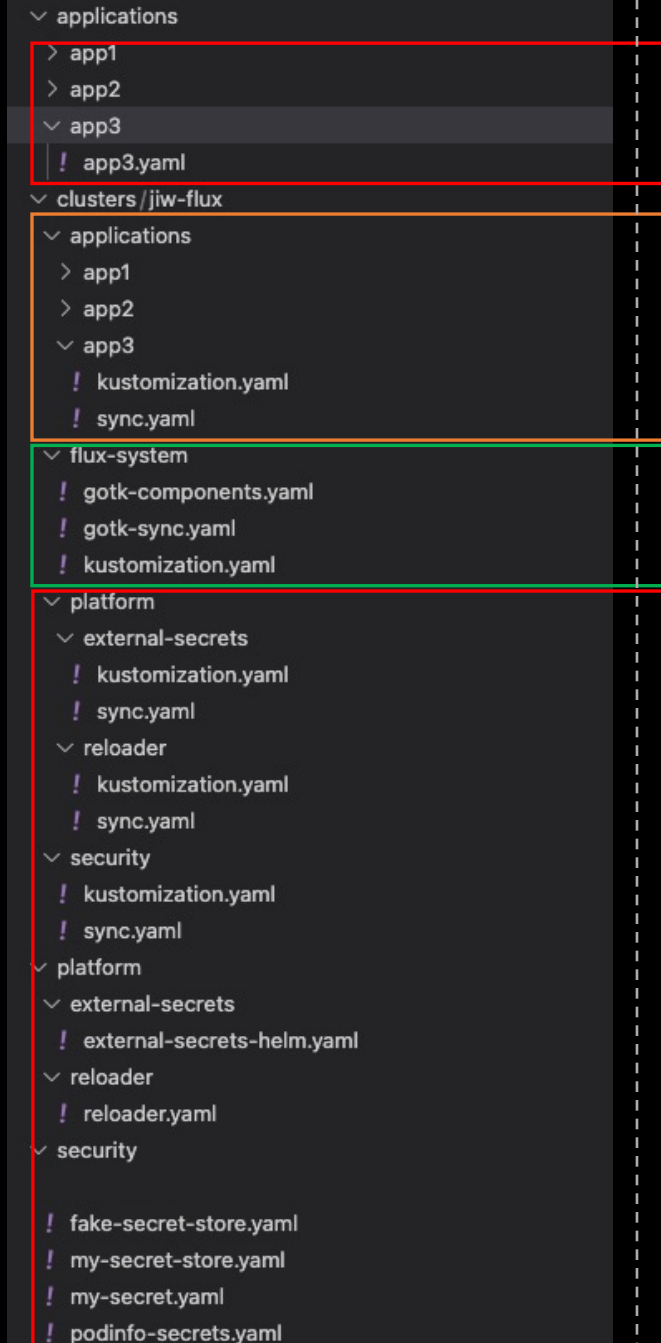
Setup “Flux” bootstrap - (GitOps)

```
export GITLAB_USER=jiw.surachat
export GITLAB_TOKEN=glpat-N85-xxxxxx_xxxxx
```

```
flux bootstrap gitlab \
--owner=$GITLAB_USER \
--repository=external-secret \
--branch=main \
--path=./clusters/jiw-flux \
--personal
```



Flux - Infrastructure as code



Application
deployment

Application
Kustomization / Sync

Flux-system core

Platform/ Security

Demo

Solution 1. Flux with SOPS CLI

Solution 2. Flux with AWS Secret

Q & A