AWS re:Invent

NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

CON318

Securing Kubernetes: How to address Kubernetes attack vectors

Micah Hausler

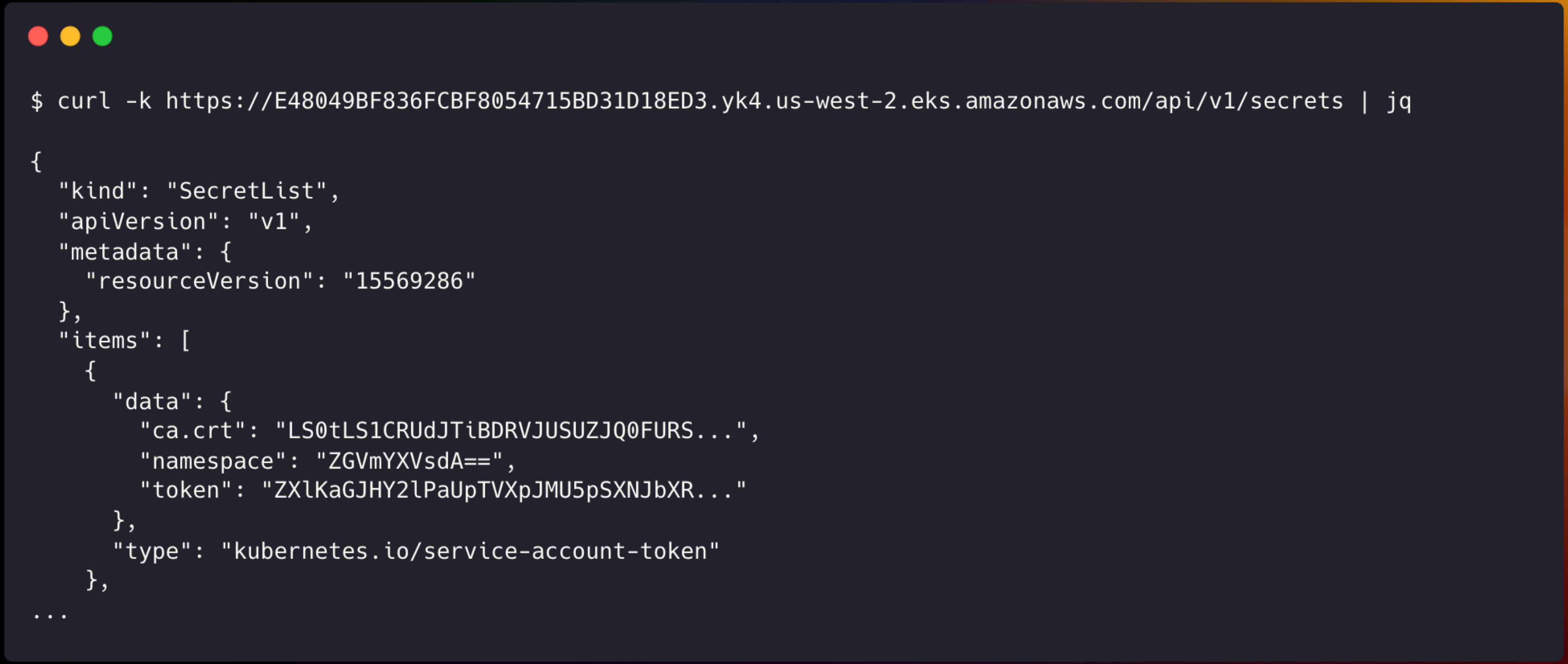
Principal Software Engineer
Amazon Web Services

Agenda

- An incident
- Threats
- Attack vectors
- Mitigations

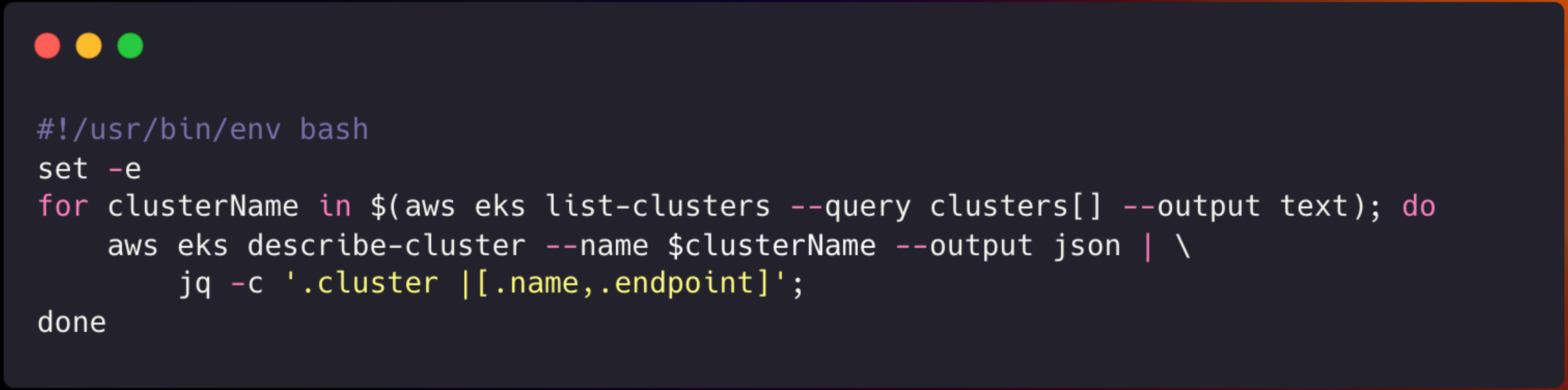
An incident

The report



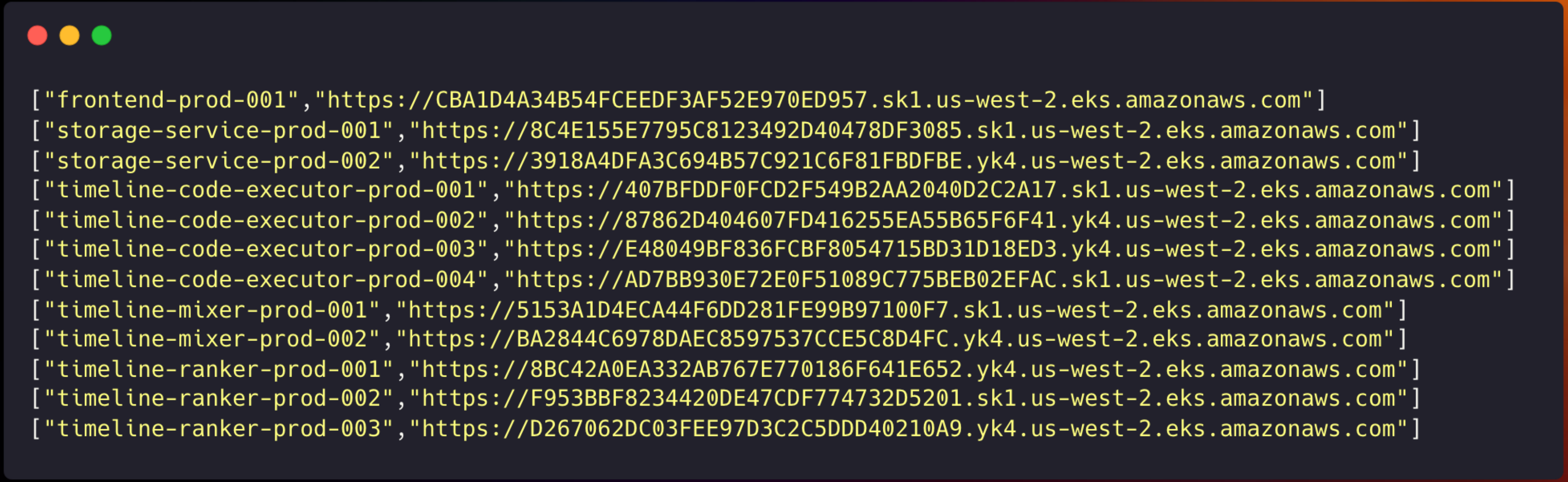
```
$ curl -k https://E48049BF836FCBF8054715BD31D18ED3.yk4.us-west-2.eks.amazonaws.com/api/v1/secrets | jq

{
  "kind": "SecretList",
  "apiVersion": "v1",
  "metadata": {
    "resourceVersion": "15569286"
  },
  "items": [
    {
      "data": {
        "ca.crt": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS...",
        "namespace": "ZGVmYXVsdA==",
        "token": "ZXlKaGJHY2lPaUpTVXpJMU5pSXNJbXR..."
      },
      "type": "kubernetes.io/service-account-token"
    },
...
```

The investigation



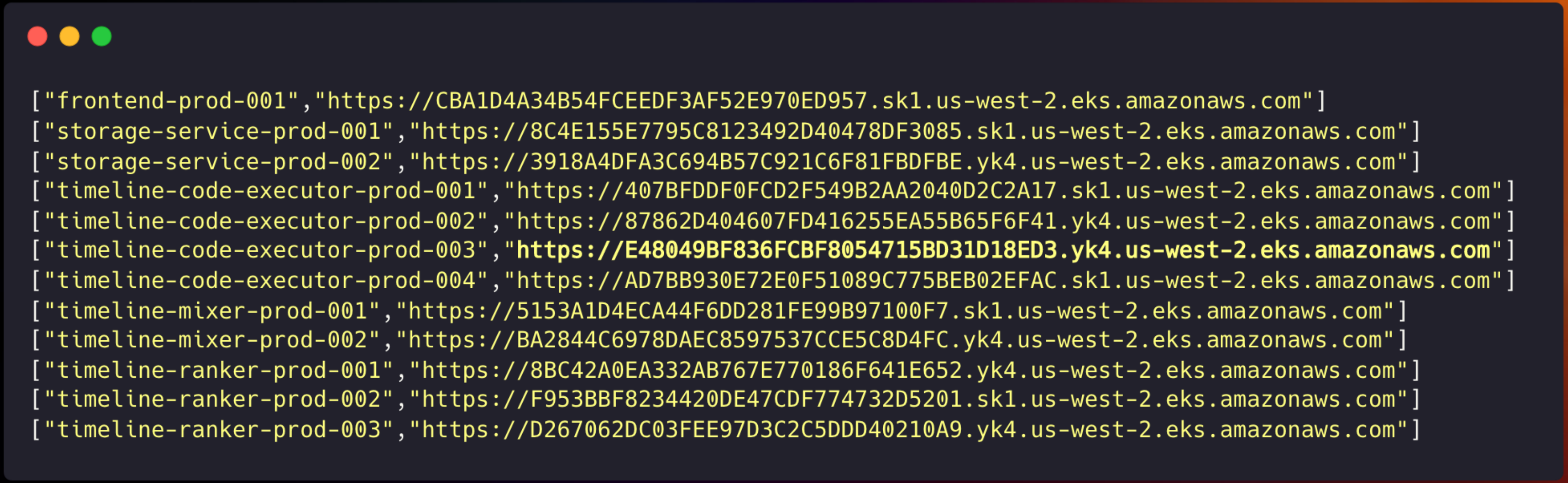
```bash
#!/usr/bin/env bash
set -e
for clusterName in $(aws eks list-clusters --query clusters[] --output text); do
    aws eks describe-cluster --name $clusterName --output json | \
        jq -c '.cluster |[.name,.endpoint]';
done
```

The investigation

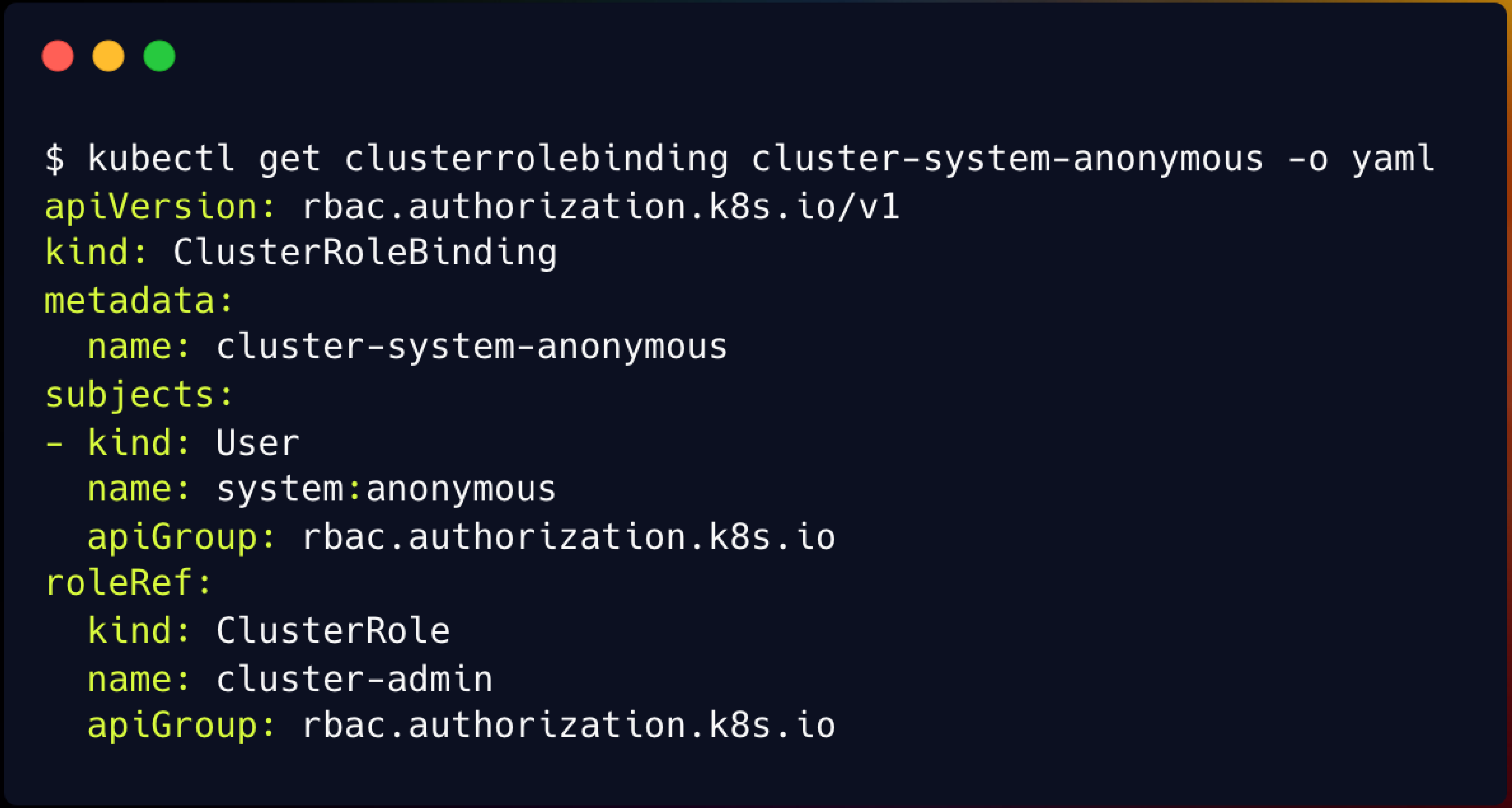


```
["frontend-prod-001","https://CBA1D4A34B54FCEEDF3AF52E970ED957.sk1.us-west-2.eks.amazonaws.com"]
["storage-service-prod-001","https://8C4E155E7795C8123492D40478DF3085.sk1.us-west-2.eks.amazonaws.com"]
["storage-service-prod-002","https://3918A4DFA3C694B57C921C6F81FBDFBE.yk4.us-west-2.eks.amazonaws.com"]
["timeline-code-executor-prod-001","https://407BFDDF0FCD2F549B2AA2040D2C2A17.sk1.us-west-2.eks.amazonaws.com"]
["timeline-code-executor-prod-002","https://87862D404607FD416255EA55B65F6F41.yk4.us-west-2.eks.amazonaws.com"]
["timeline-code-executor-prod-003","https://E48049BF836FCBF8054715BD31D18ED3.yk4.us-west-2.eks.amazonaws.com"]
["timeline-code-executor-prod-004","https://AD7BB930E72E0F51089C775BEB02EFAC.sk1.us-west-2.eks.amazonaws.com"]
["timeline-mixer-prod-001","https://5153A1D4ECA44F6DD281FE99B97100F7.sk1.us-west-2.eks.amazonaws.com"]
["timeline-mixer-prod-002","https://BA2844C6978DAEC8597537CCE5C8D4FC.yk4.us-west-2.eks.amazonaws.com"]
["timeline-ranker-prod-001","https://8BC42A0EA332AB767E770186F641E652.yk4.us-west-2.eks.amazonaws.com"]
["timeline-ranker-prod-002","https://F953BBF8234420DE47CDF774732D5201.sk1.us-west-2.eks.amazonaws.com"]
["timeline-ranker-prod-003","https://D267062DC03FEE97D3C2C5DDD40210A9.yk4.us-west-2.eks.amazonaws.com"]
```

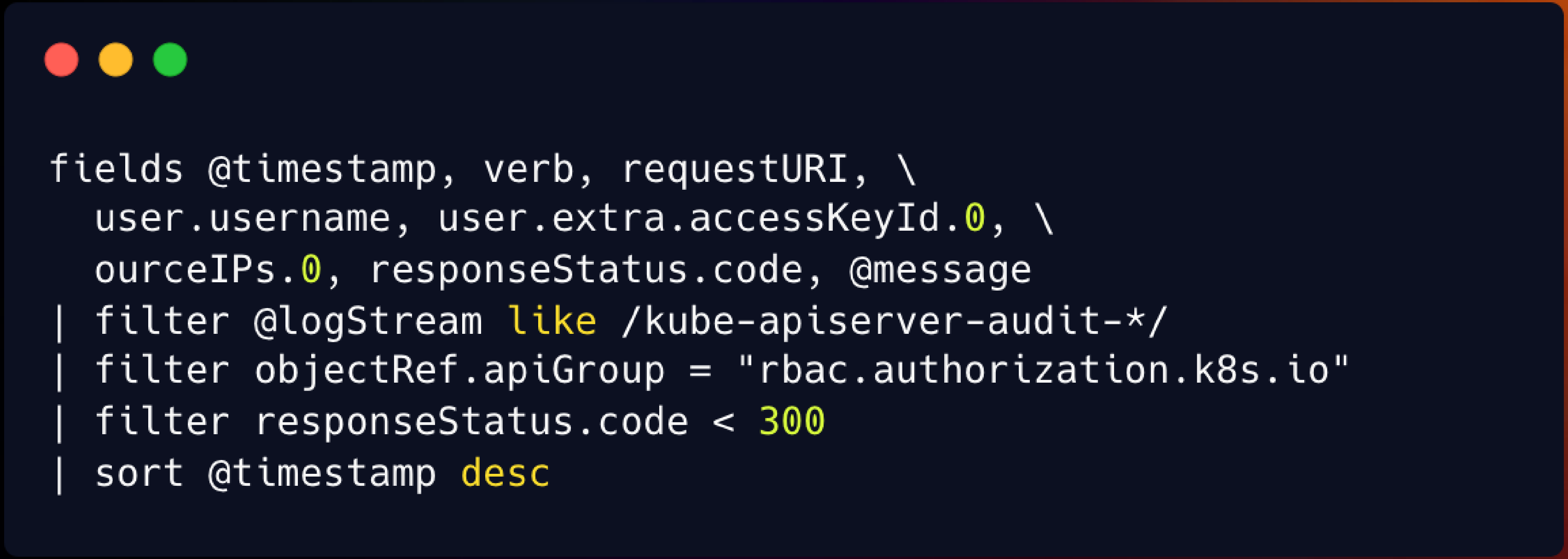
The investigation



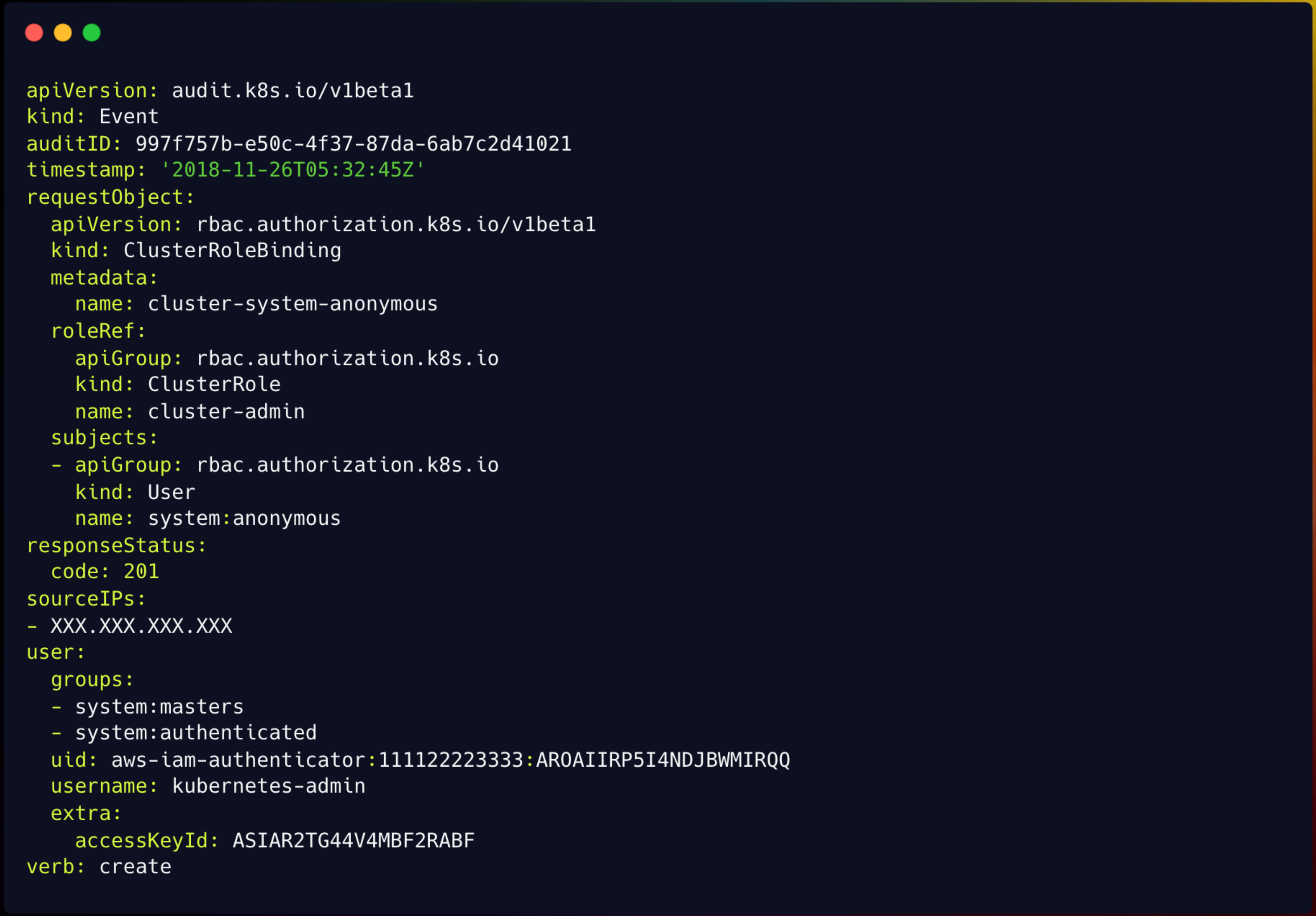
```
["frontend-prod-001","https://CBA1D4A34B54FCEEDF3AF52E970ED957.sk1.us-west-2.eks.amazonaws.com"]
["storage-service-prod-001","https://8C4E155E7795C8123492D40478DF3085.sk1.us-west-2.eks.amazonaws.com"]
["storage-service-prod-002","https://3918A4DFA3C694B57C921C6F81FBDFBE.yk4.us-west-2.eks.amazonaws.com"]
["timeline-code-executor-prod-001","https://407BFDDF0FCD2F549B2AA2040D2C2A17.sk1.us-west-2.eks.amazonaws.com"]
["timeline-code-executor-prod-002","https://87862D404607FD416255EA55B65F6F41.yk4.us-west-2.eks.amazonaws.com"]
["timeline-code-executor-prod-003","https://E48049BF836FCBF8054715BD31D18ED3.yk4.us-west-2.eks.amazonaws.com"]
["timeline-code-executor-prod-004","https://AD7BB930E72E0F51089C775BEB02EFAC.sk1.us-west-2.eks.amazonaws.com"]
["timeline-mixer-prod-001","https://5153A1D4ECA44F6DD281FE99B97100F7.sk1.us-west-2.eks.amazonaws.com"]
["timeline-mixer-prod-002","https://BA2844C6978DAEC8597537CCE5C8D4FC.yk4.us-west-2.eks.amazonaws.com"]
["timeline-ranker-prod-001","https://8BC42A0EA332AB767E770186F641E652.yk4.us-west-2.eks.amazonaws.com"]
["timeline-ranker-prod-002","https://F953BBF8234420DE47CDF774732D5201.sk1.us-west-2.eks.amazonaws.com"]
["timeline-ranker-prod-003","https://D267062DC03FEE97D3C2C5DDD40210A9.yk4.us-west-2.eks.amazonaws.com"]
```



```
$ kubectl get clusterrolebinding cluster-system-anonymous -o yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: cluster-system-anonymous
subjects:
- kind: User
  name: system:anonymous
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: cluster-admin
  apiGroup: rbac.authorization.k8s.io
```

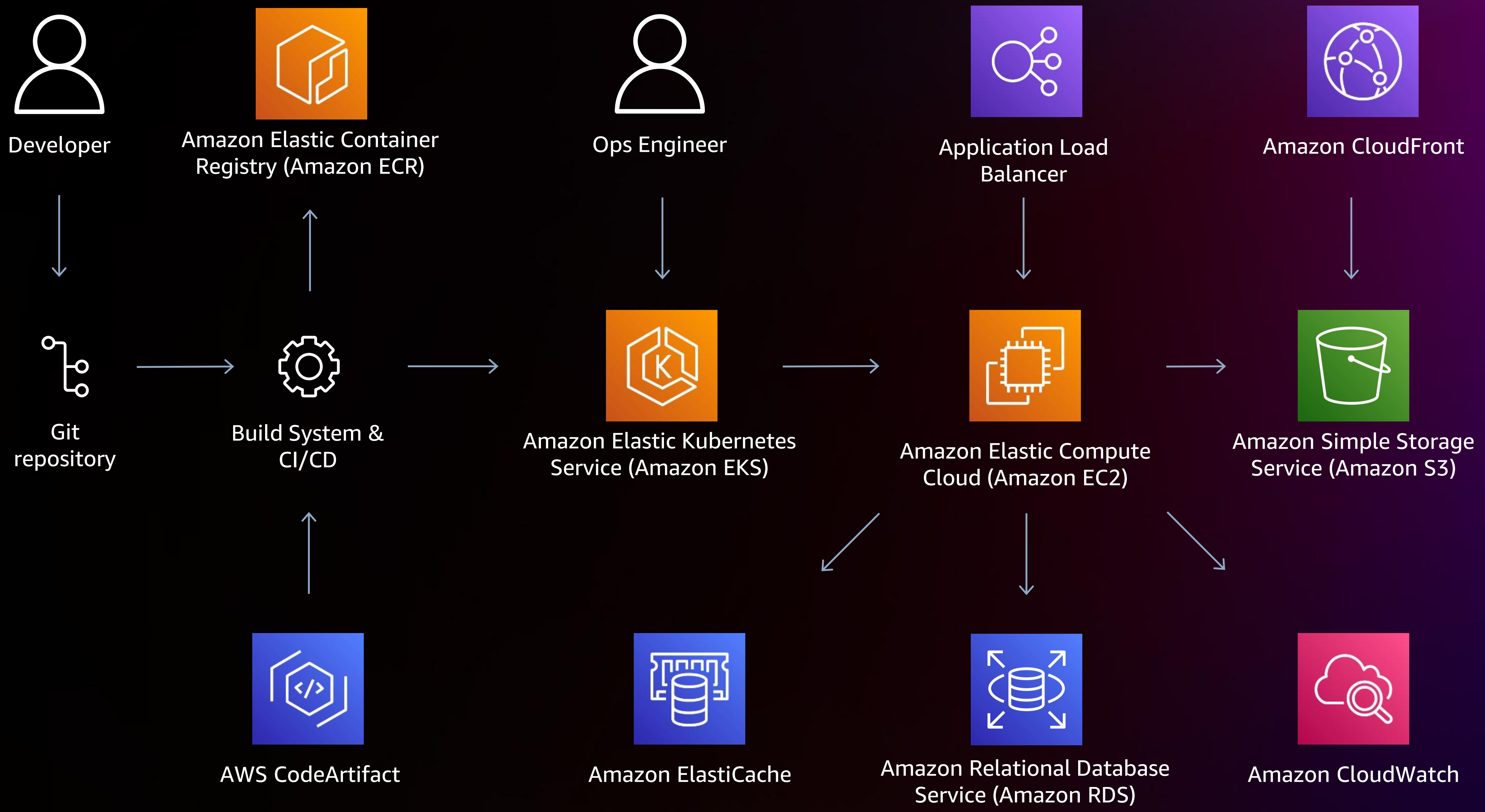


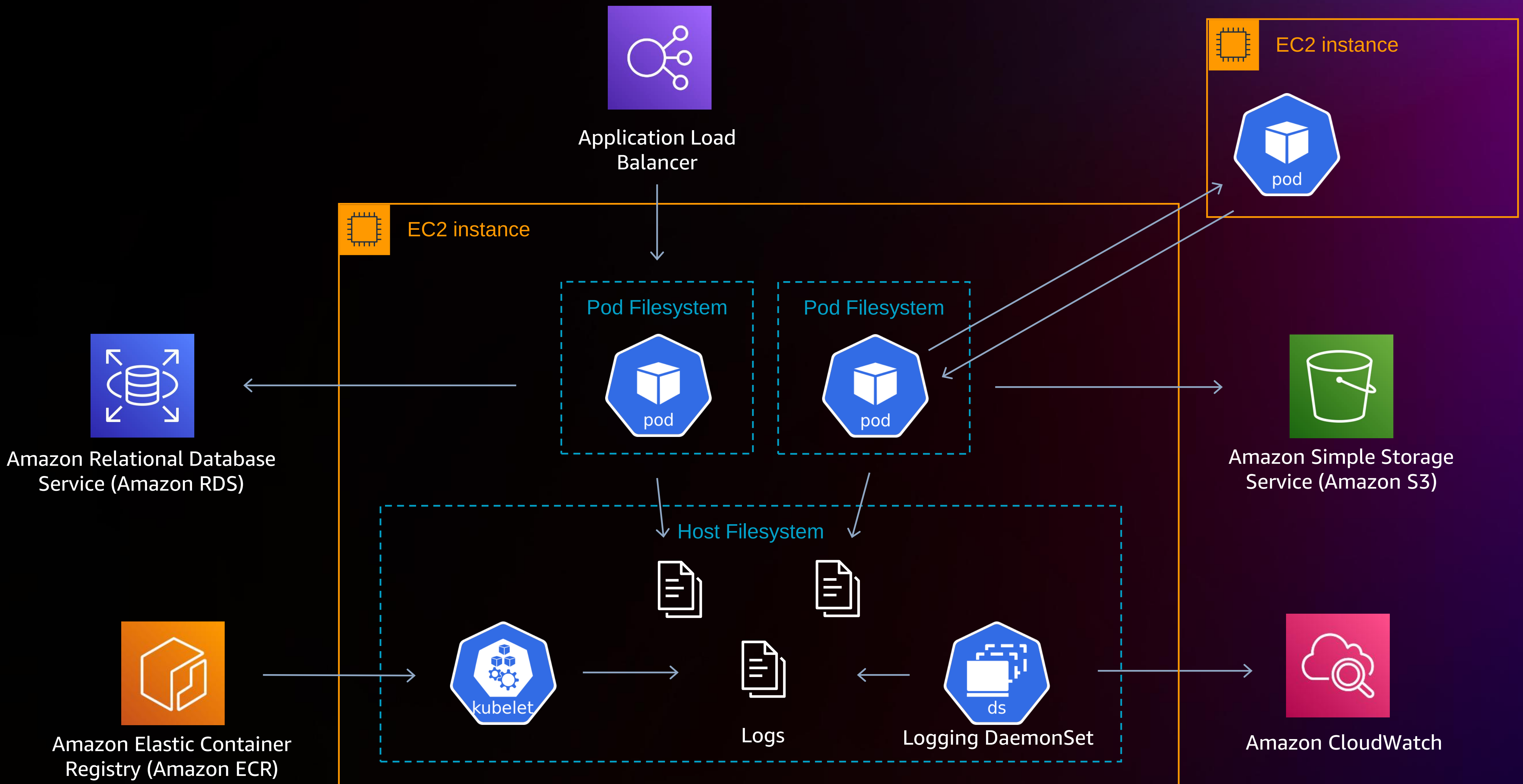
```
fields @timestamp, verb, requestURI, \
  user.username, user.extra.accessKeyId.0, \
  ourceIPs.0, responseStatus.code, @message
| filter @logStream like /kube-apiserver-audit-*/
| filter objectRef.apiGroup = "rbac.authorization.k8s.io"
| filter responseStatus.code < 300
| sort @timestamp desc
```



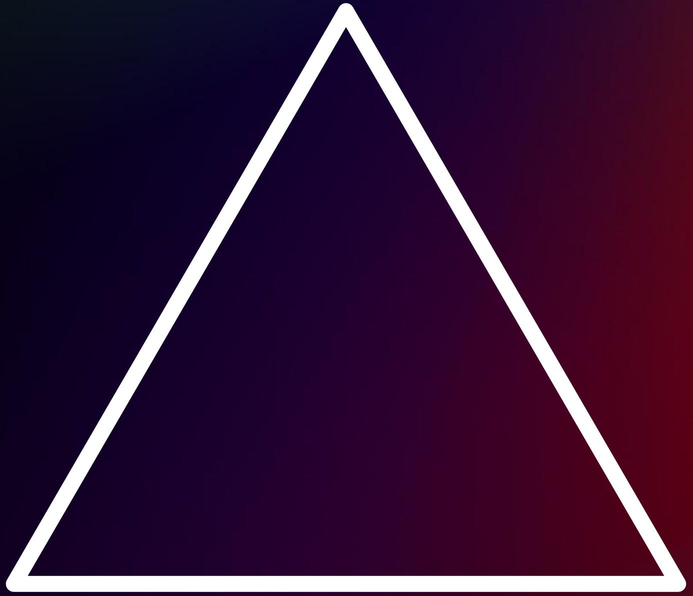
```
apiVersion: audit.k8s.io/v1beta1
kind: Event
auditID: 997f757b-e50c-4f37-87da-6ab7c2d41021
timestamp: '2018-11-26T05:32:45Z'
requestObject:
  apiVersion: rbac.authorization.k8s.io/v1beta1
  kind: ClusterRoleBinding
  metadata:
    name: cluster-system-anonymous
  roleRef:
    apiGroup: rbac.authorization.k8s.io
    kind: ClusterRole
    name: cluster-admin
  subjects:
  - apiGroup: rbac.authorization.k8s.io
    kind: User
    name: system:anonymous
responseStatus:
  code: 201
sourceIPs:
- XXX.XXX.XXX.XXX
user:
  groups:
  - system:masters
  - system:authenticated
  uid: aws-iam-authenticator:111122223333:AROAIIRP5I4NDJBWMIRQQ
  username: kubernetes-admin
  extra:
    accessKeyId: ASIAR2TG44V4MBF2RABF
verb: create
```

Threats


Developer
Amazon Elastic Container Registry (Amazon ECR)
Ops Engineer
Application Load Balancer
Amazon CloudFront
Git repository
Build System & CI/CD
Amazon Elastic Kubernetes Service (Amazon EKS)
Amazon Elastic Compute Cloud (Amazon EC2)
Amazon Simple Storage Service (Amazon S3)
AWS CodeArtifact
Amazon ElastiCache
Amazon Relational Database Service (Amazon RDS)
Amazon CloudWatch


Application Load Balancer
EC2 instance
pod
EC2 instance
Pod Filesystem
Pod Filesystem
pod
pod
Amazon Relational Database Service (Amazon RDS)
Amazon Simple Storage Service (Amazon S3)
Host Filesystem
kubelet
Logs
ds
Logging DaemonSet
Amazon Elastic Container Registry (Amazon ECR)
Amazon CloudWatch

What are common threats to Kubernetes?



Confidentiality

Integrity

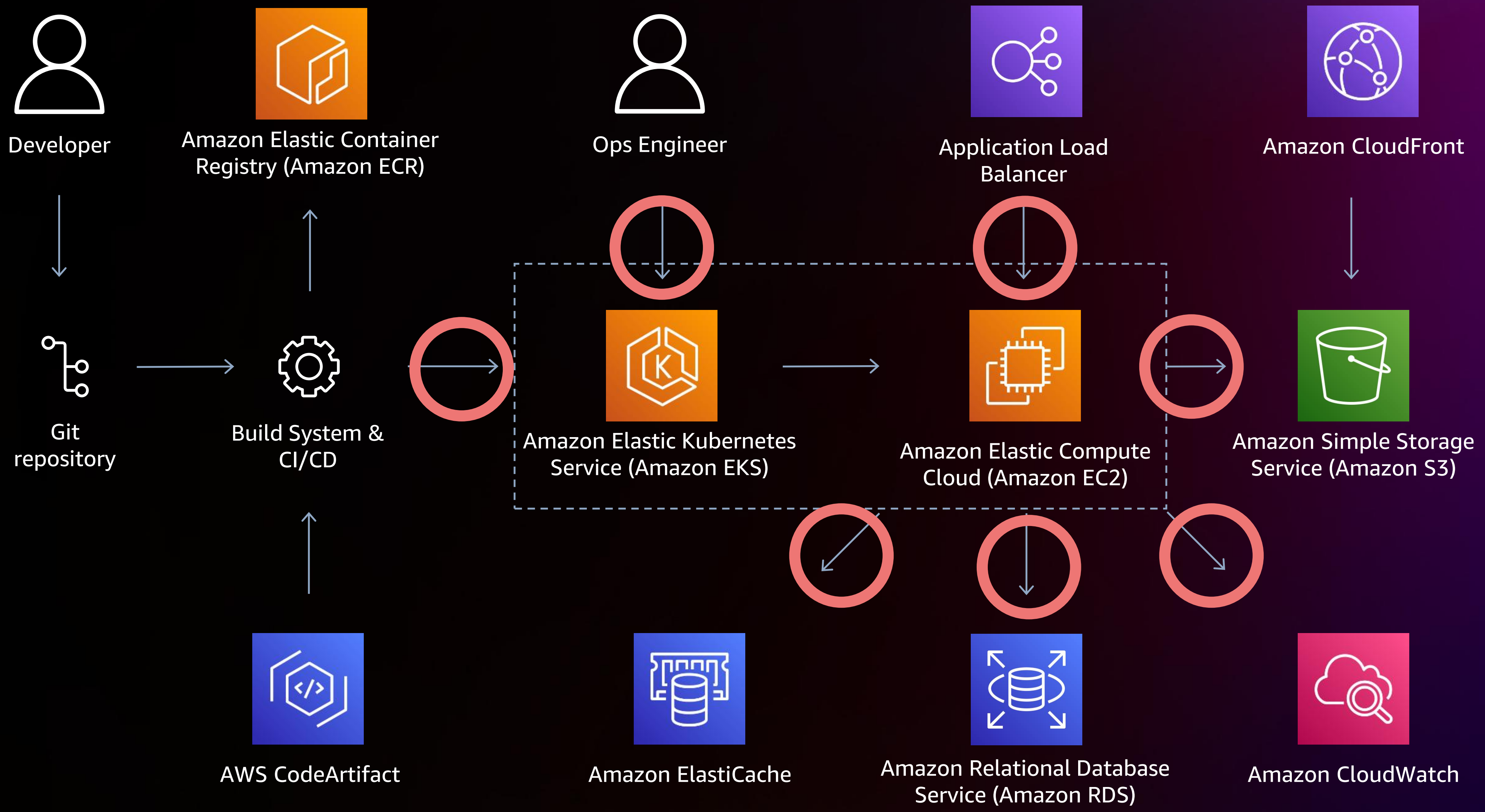
Availability

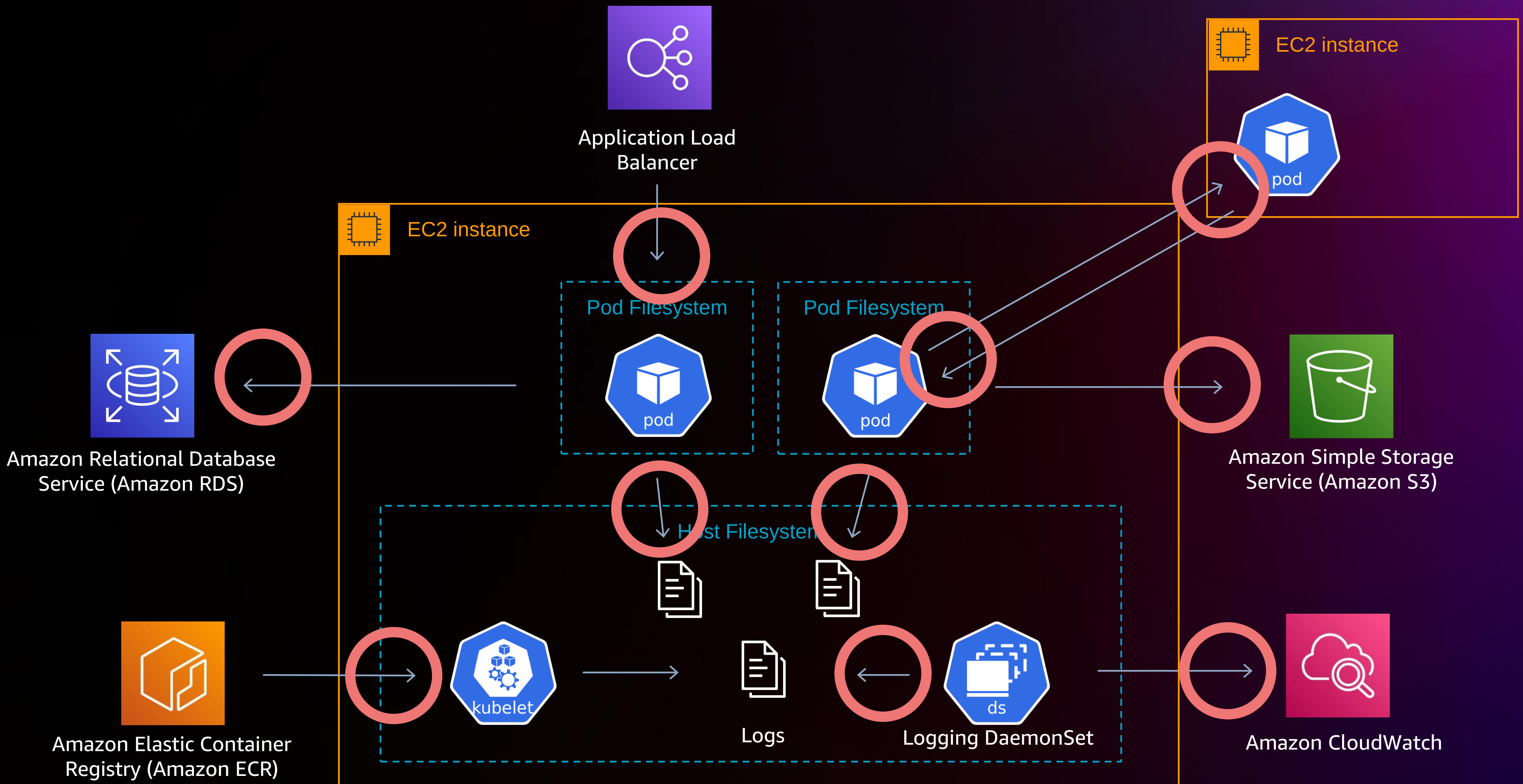
Threat modeling questions

- Availability
  - What networks need access to your applications?
  - What networks need access to Kubernetes?
- Integrity
  - What actors or processes need access to your data?
  - What actors or processes need access to your software supply chain?
- Confidentiality
  - What actors or processes need access to your data?
  - What actors or processes need access to your compute runtime?

Kubernetes threat modeling questions

- Do you run arbitrary customer code?
    - Do you trust the container as an isolation boundary?
- Does your application make outbound connections to arbitrary networks?
    - Are all external network calls known or unknown?
- What networks, users, and processes need access to the Kubernetes API?
- What applications in Kubernetes need access to outside systems?


Developer
Amazon Elastic Container Registry (Amazon ECR)
Ops Engineer
Application Load Balancer
Amazon CloudFront
Git repository
Build System & CI/CD
Amazon Elastic Kubernetes Service (Amazon EKS)
Amazon Elastic Compute Cloud (Amazon EC2)
Amazon Simple Storage Service (Amazon S3)
AWS CodeArtifact
Amazon ElastiCache
Amazon Relational Database Service (Amazon RDS)
Amazon CloudWatch


Application Load Balancer
EC2 instance
EC2 instance
pod
Pod Filesystem
pod
Pod Filesystem
pod
Amazon Relational Database Service (Amazon RDS)
Amazon Simple Storage Service (Amazon S3)
Host Filesystem
kubelet
Logs
ds
Logging DaemonSet
Amazon Elastic Container Registry (Amazon ECR)
Amazon CloudWatch

Attack vectors & mitigations

OWASP Top 10 – 2021

1. Broken Access Configuration
2. Cryptographic Failures
3. Injection
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable or Outdated Components
7. Identification and Authentication Failures
8. Software and Data Integrity Failures
9. Security Logging and Monitoring Failures
10. Server-Side Request Forgery (SSRF)

OWASP Top 10 – 2021

1. Broken Access Configuration
2. Cryptographic Failures
3. Injection
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable or Outdated Components
7. Identification and Authentication Failures
8. Software and Data Integrity Failures
9. Security Logging and Monitoring Failures
10. Server-Side Request Forgery (SSRF)

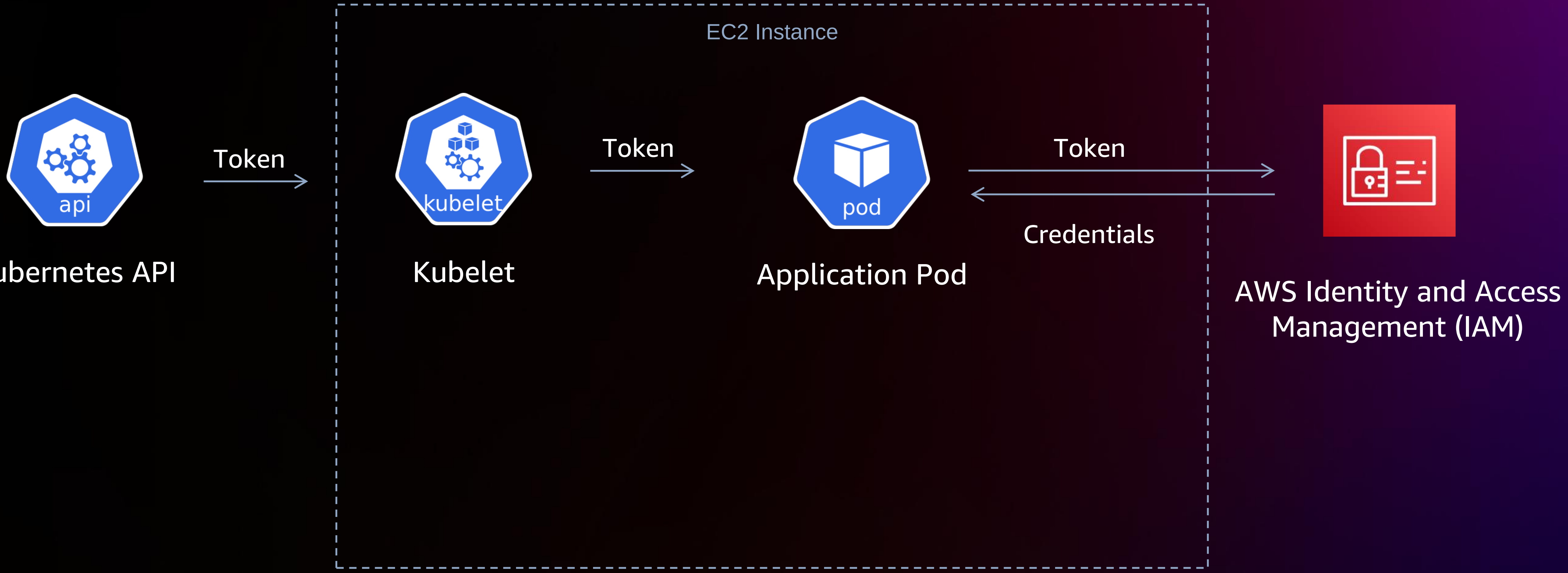
1. Access control

Broken access control

- Violation of least privilege
  - Kubernetes API permissions for users and pods
  - Service metadata to pods
  - Linux permissions for pods
- Privilege escalation
- Kubernetes vulnerabilities
  - CVE-2022-3162: Unauthorized read of custom resources

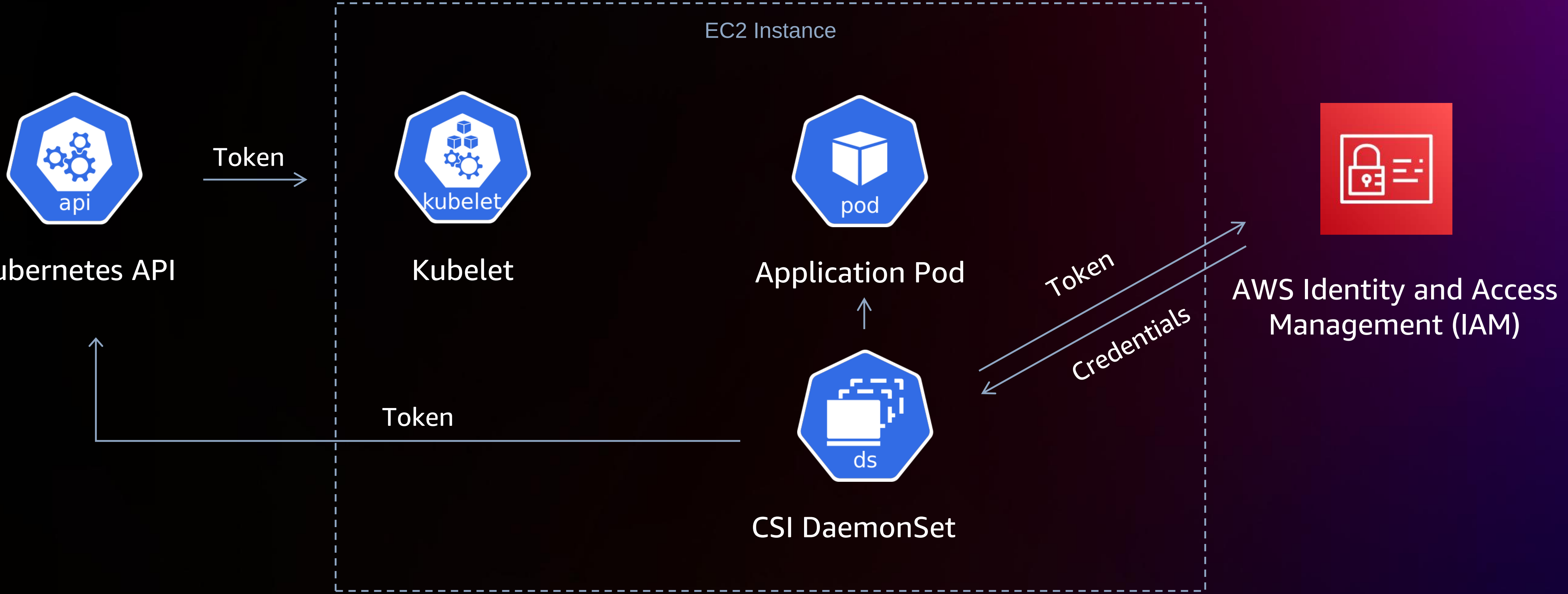
Broken access control

LEAST PRIVILEGE



Broken access control

LEAST PRIVILEGE



Broken access control

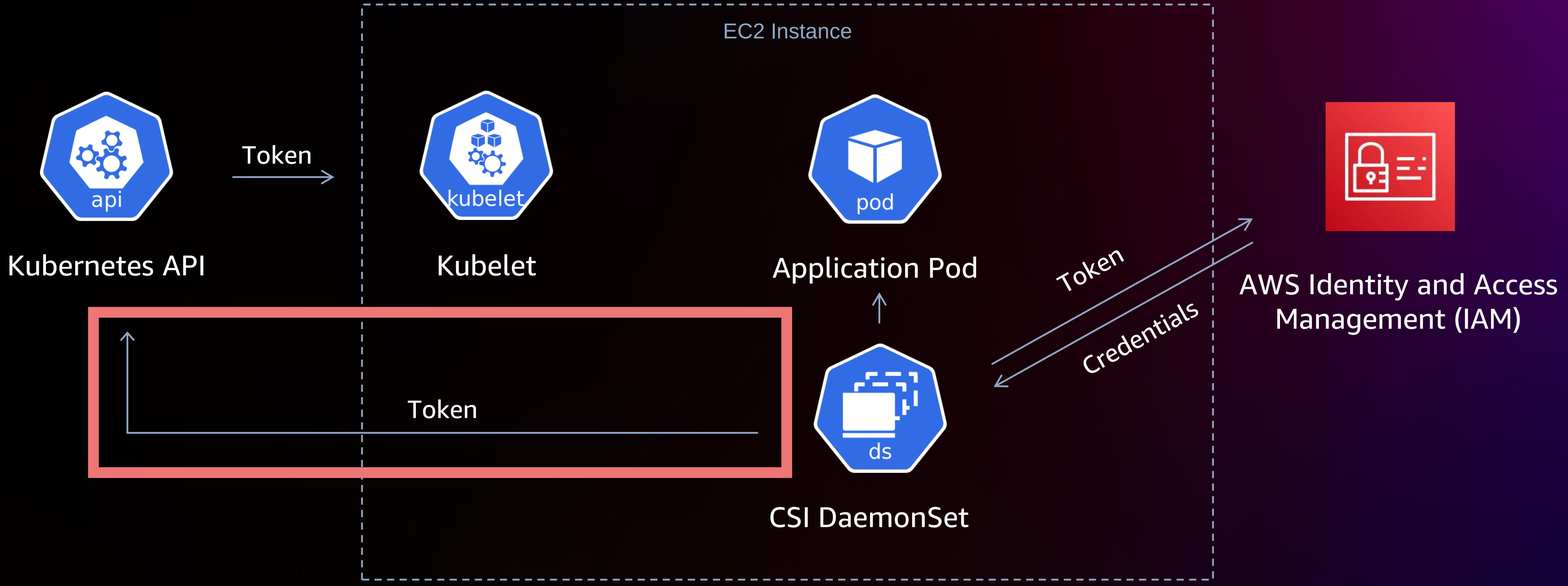
LEAST PRIVILEGE

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: cool-csi-driver-daemonset
rules:
- apiGroups: [""]
  resources: ["serviceaccounts/token"]
  verbs: ["create"]
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: cool-csi-driver-daemonset
subjects:
- kind: ServiceAccount
  name: cool-csi-driver-daemonset
  namespace: default
roleRef:
  kind: ClusterRole
  name: cool-csi-driver
  apiGroup: rbac.authorization.k8s.io
```

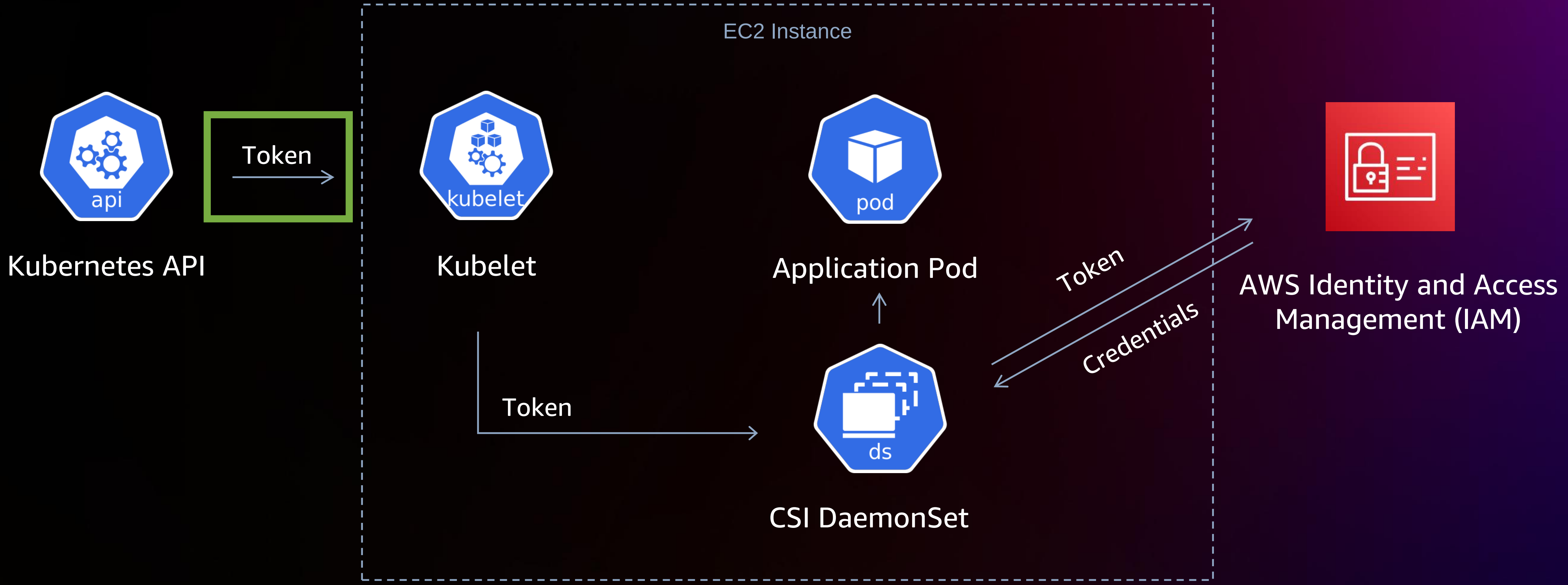
Broken access control

LEAST PRIVILEGE



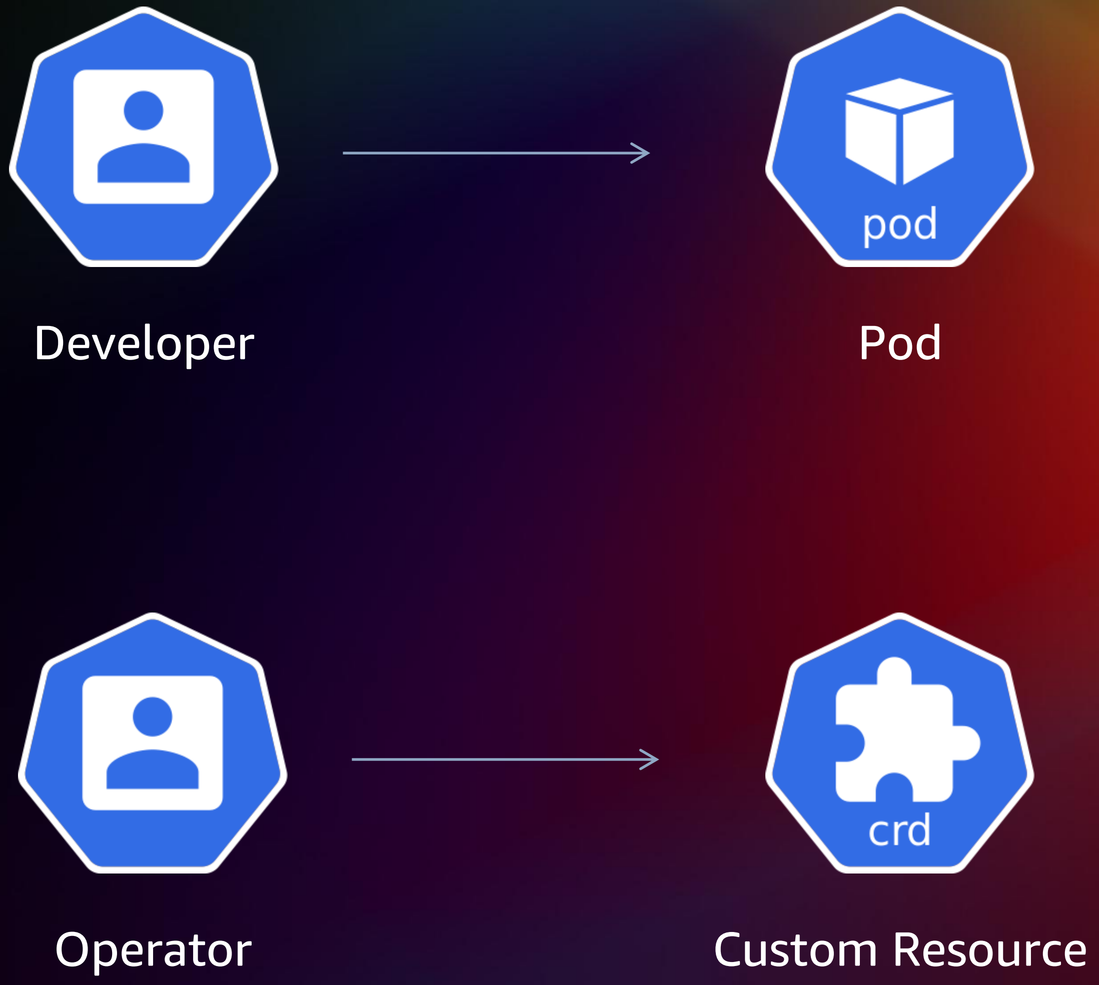
Broken access control mitigation

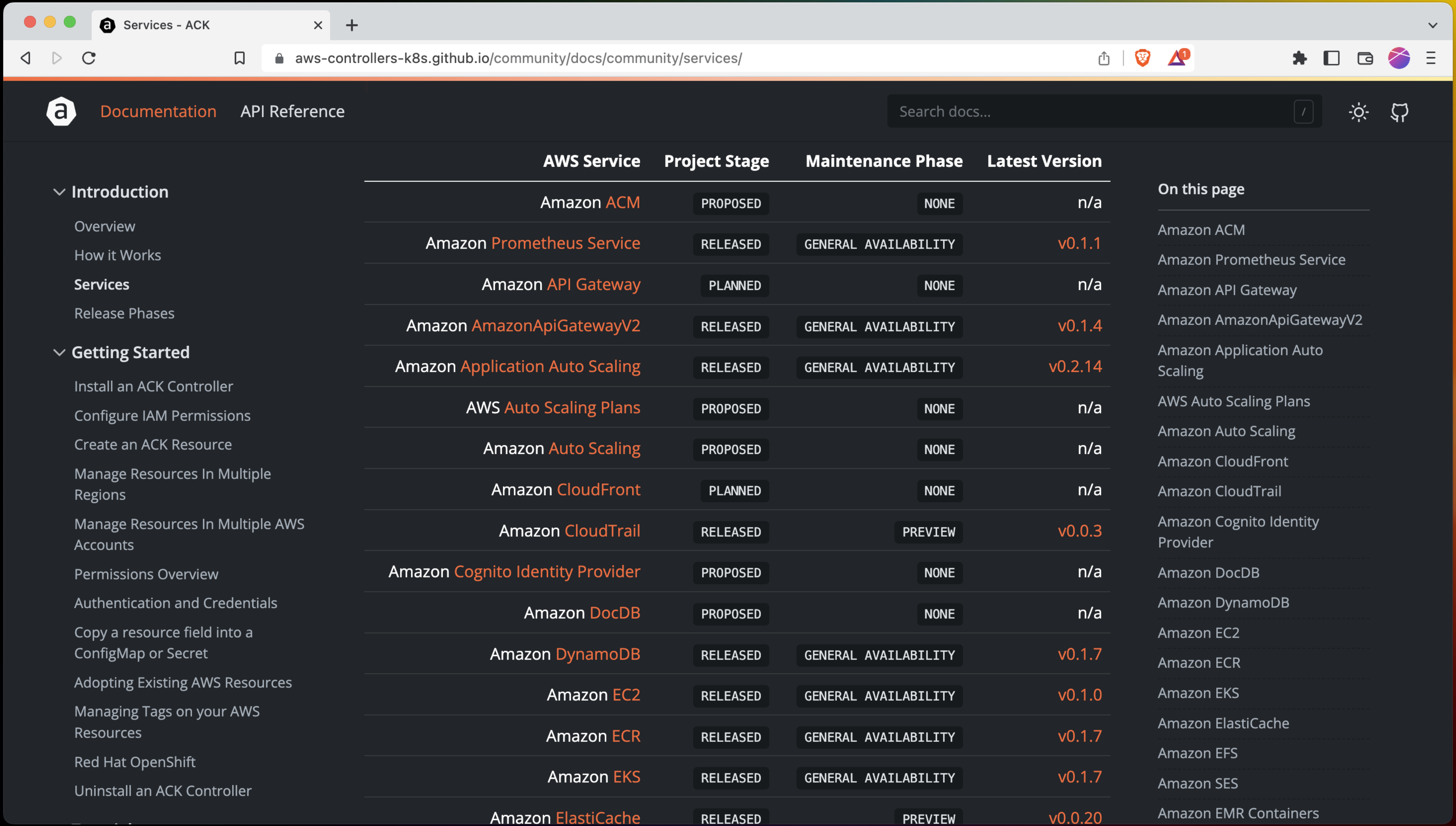
LEAST PRIVILEGE



Broken access control

PRIVILEGE ESCALATION



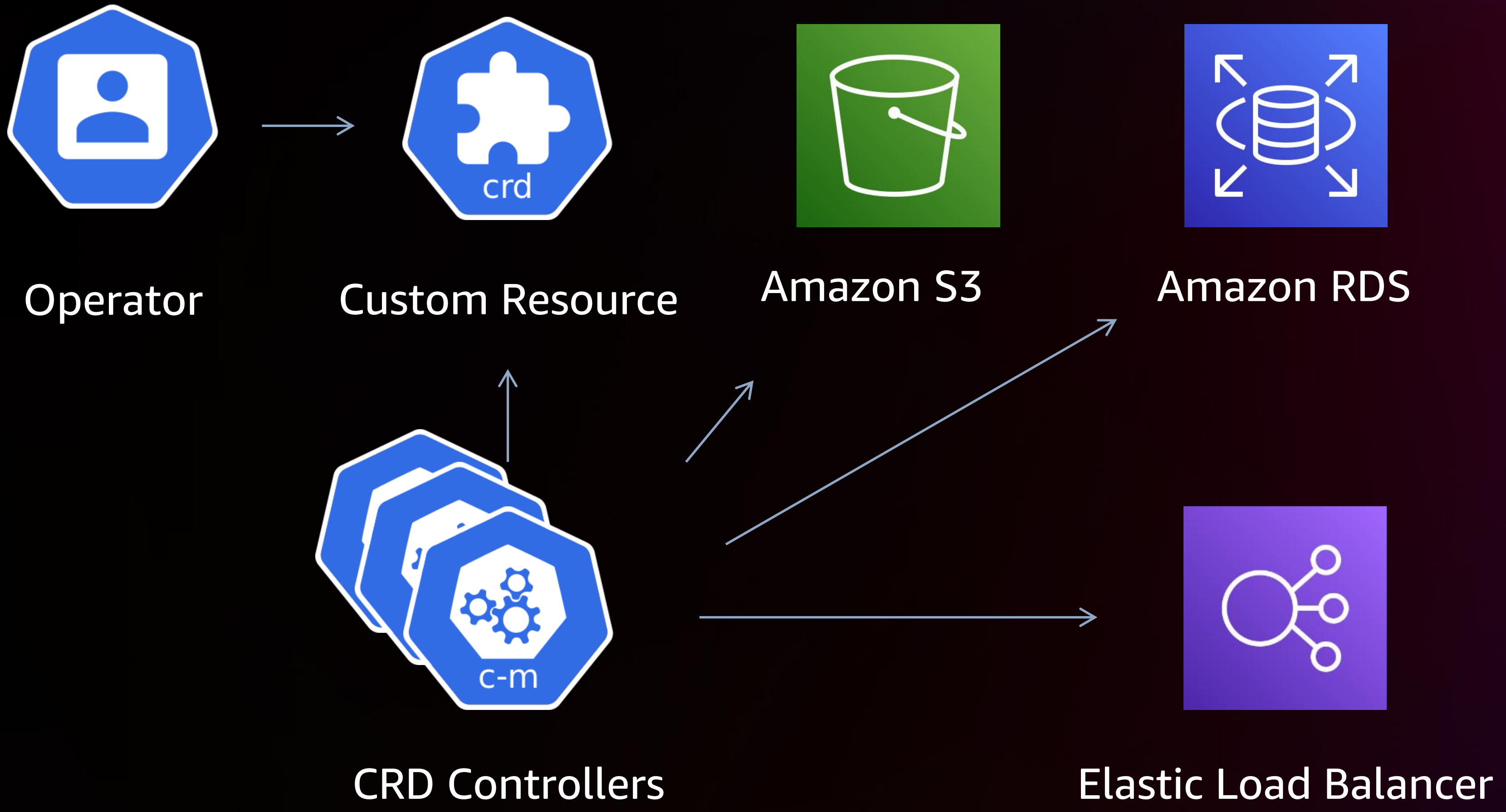


Services - ACK
aws-controllers-k8s.github.io/community/docs/community/services/
Documentation
API Reference
Search docs...

AWS Service	Project Stage	Maintenance Phase	Latest Version
Amazon ACM	PROPOSED	NONE	n/a
Amazon Prometheus Service	RELEASED	GENERAL AVAILABILITY	v0.1.1
Amazon API Gateway	PLANNED	NONE	n/a
Amazon AmazonApiGatewayV2	RELEASED	GENERAL AVAILABILITY	v0.1.4
Amazon Application Auto Scaling	RELEASED	GENERAL AVAILABILITY	v0.2.14
AWS Auto Scaling Plans	PROPOSED	NONE	n/a
Amazon Auto Scaling	PROPOSED	NONE	n/a
Amazon CloudFront	PLANNED	NONE	n/a
Amazon CloudTrail	RELEASED	PREVIEW	v0.0.3
Amazon Cognito Identity Provider	PROPOSED	NONE	n/a
Amazon DocDB	PROPOSED	NONE	n/a
Amazon DynamoDB	RELEASED	GENERAL AVAILABILITY	v0.1.7
Amazon EC2	RELEASED	GENERAL AVAILABILITY	v0.1.0
Amazon ECR	RELEASED	GENERAL AVAILABILITY	v0.1.7
Amazon EKS	RELEASED	GENERAL AVAILABILITY	v0.1.7
Amazon ElastiCache	RELEASED	PREVIEW	v0.0.20

Broken access control

PRIVILEGE ESCALATION



Broken access control

PRIVILEGE ESCALATION

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: ack-admin
rules:
- apiGroups: ["s3.services.k8s.aws"]
  resources: ["*"]
  verbs: ["*"]
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: ack-admin
subjects:
- kind: Group
  name: operator
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: ack-admin
  apiGroup: rbac.authorization.k8s.io
```

Broken access control

PRIVILEGE ESCALATION

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: core-admin
rules:
- apiGroups: [""]
  resources: ["*"]
  verbs: ["*"]
```

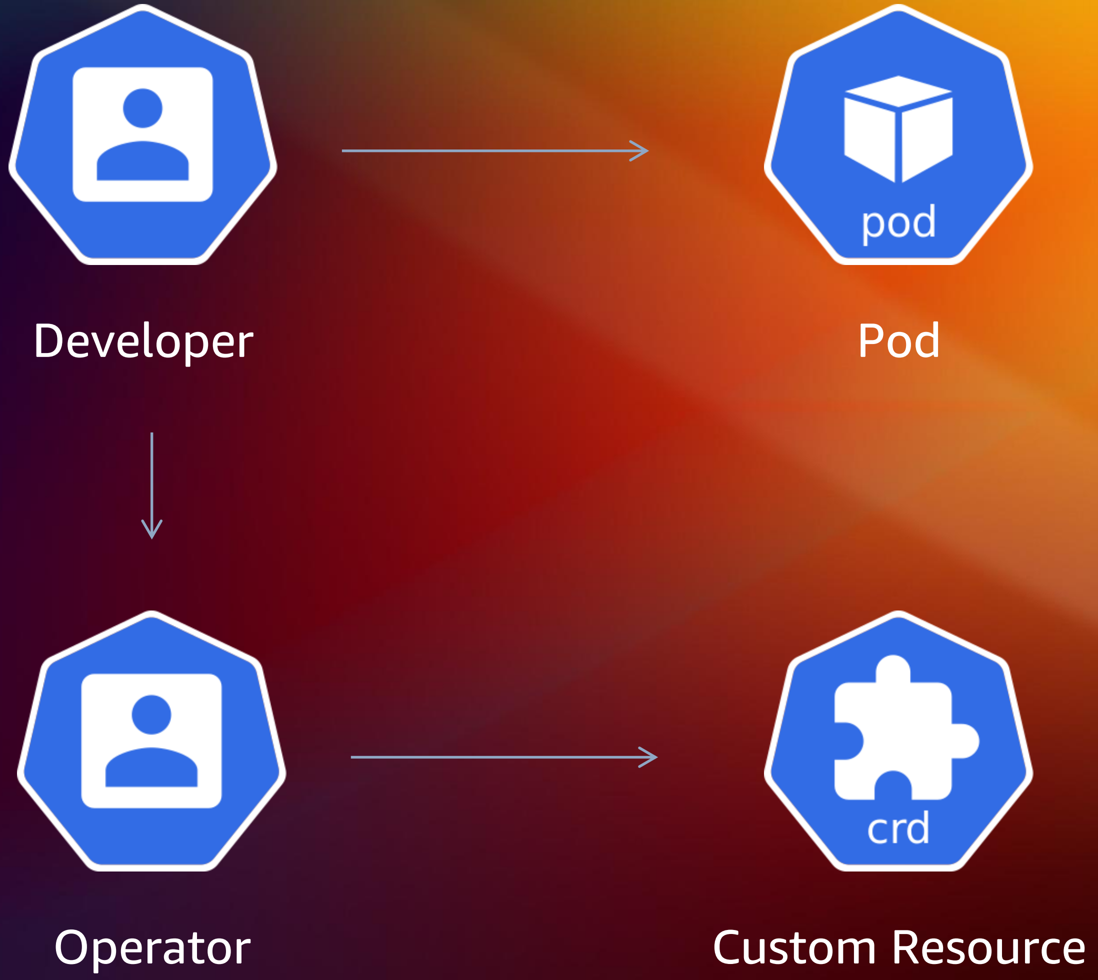
```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: core-admin
subjects:
- kind: Group
  name: developer
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: core-admin
  apiGroup: rbac.authorization.k8s.io
```

Broken access control

PRIVILEGE ESCALATION

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: impersonator
rules:
- apiGroups: [""]
  resources: ["users", "groups”]
  verbs: ["impersonate"]
```

```
kubectl get s3.services.k8s.aws/bucket \
  --as-group=operator
```



Broken access control – Mitigations

- Use least-privilege RBAC roles
  - Generate policies from audit logs - https://github.com/liggitt/audit2rbac
- Limit cluster-wide permissions to DaemonSets
- Use CSI drivers that support TokenRequest
- Explicitly enumerate verbs and resources in RBAC policies

5. Security misconfigurations

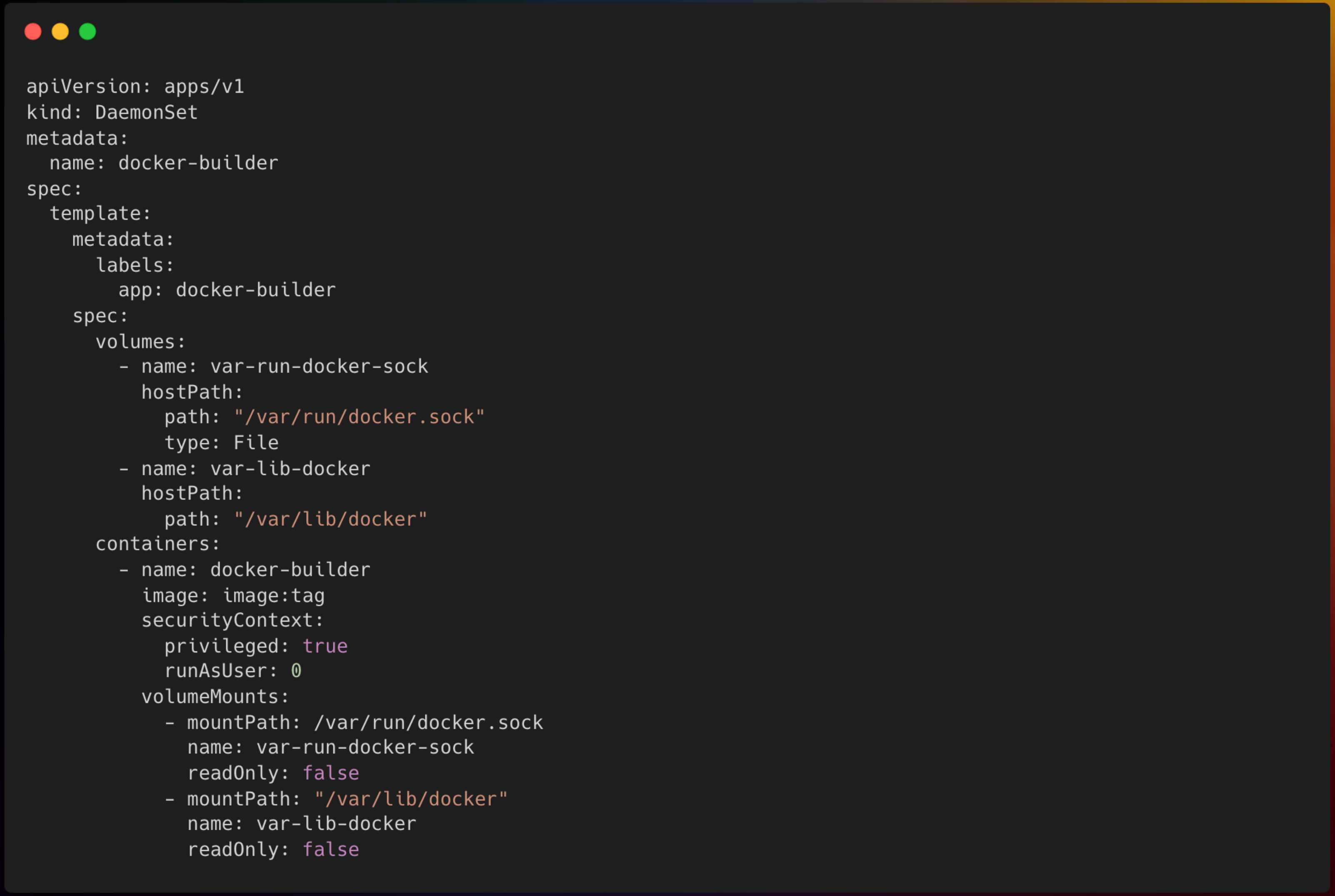
Security misconfigurations

- Authorization misconfigurations
- Unnecessary features enabled
- Insecure defaults

Security misconfiguration

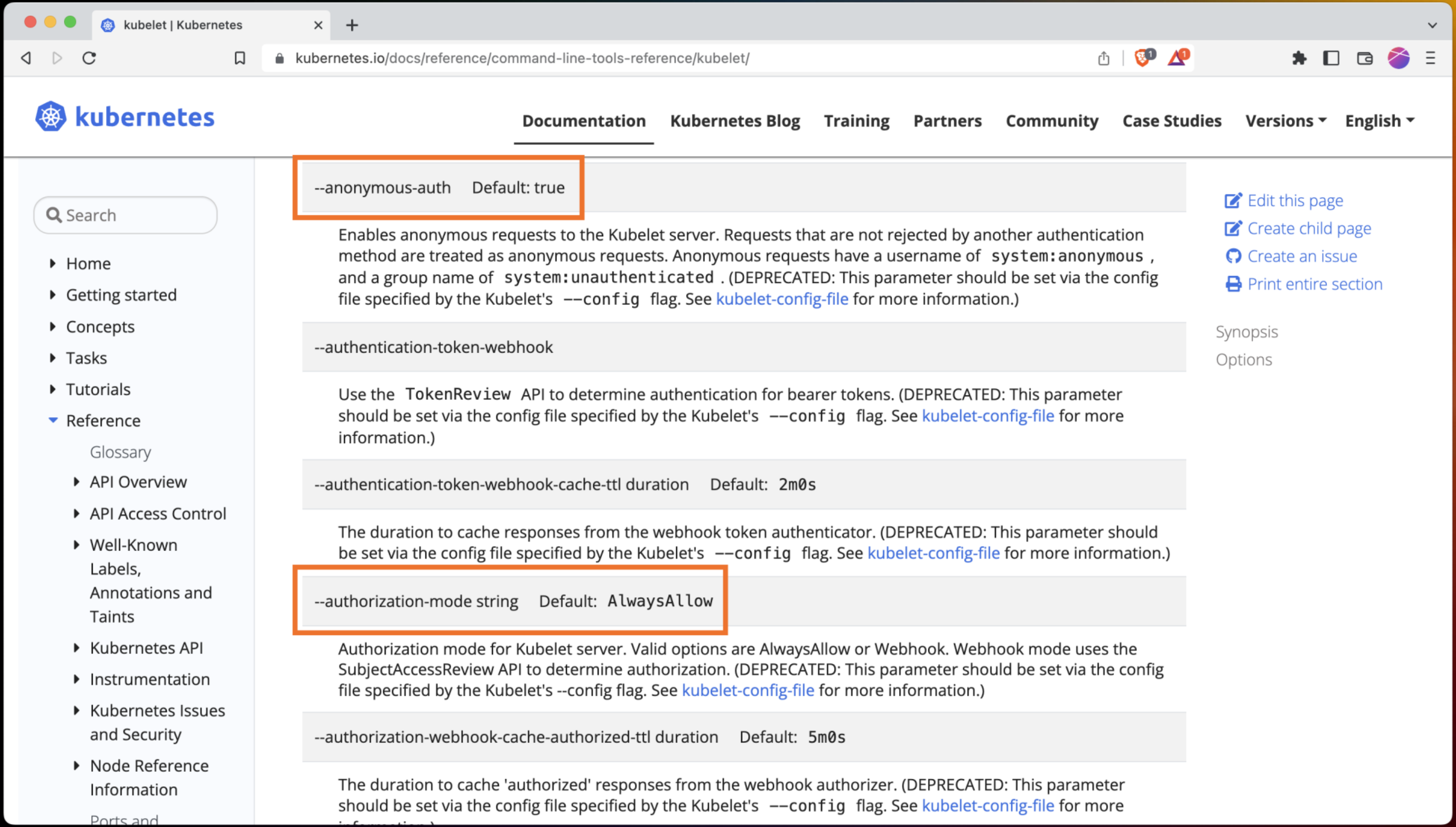
```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: debug-get-it-to-work-really-11
subjects:
- kind: Group
  name: system:anonymous
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: cluster-admin
  apiGroup: rbac.authorization.k8s.io
```

Security misconfiguration – Pod configuration



```yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: docker-builder
spec:
  template:
    metadata:
      labels:
        app: docker-builder
    spec:
      volumes:
        - name: var-run-docker-sock
          hostPath:
            path: "/var/run/docker.sock"
            type: File
        - name: var-lib-docker
          hostPath:
            path: "/var/lib/docker"
      containers:
        - name: docker-builder
          image: image:tag
          securityContext:
            privileged: true
            runAsUser: 0
          volumeMounts:
            - mountPath: /var/run/docker.sock
              name: var-run-docker-sock
              readOnly: false
            - mountPath: "/var/lib/docker"
              name: var-lib-docker
              readOnly: false
```

Security misconfiguration – Kubelet defaults



--anonymous-auth   Default: true

Enables anonymous requests to the Kubelet server. Requests that are not rejected by another authentication method are treated as anonymous requests. Anonymous requests have a username of `system:anonymous`, and a group name of `system:unauthenticated`. (DEPRECATED: This parameter should be set via the config file specified by the Kubelet's `--config` flag. See kubelet-config-file for more information.)

--authentication-token-webhook

Use the `TokenReview` API to determine authentication for bearer tokens. (DEPRECATED: This parameter should be set via the config file specified by the Kubelet's `--config` flag. See kubelet-config-file for more information.)

--authentication-token-webhook-cache-ttl duration   Default: `2m0s`

The duration to cache responses from the webhook token authenticator. (DEPRECATED: This parameter should be set via the config file specified by the Kubelet's `--config` flag. See kubelet-config-file for more information.)

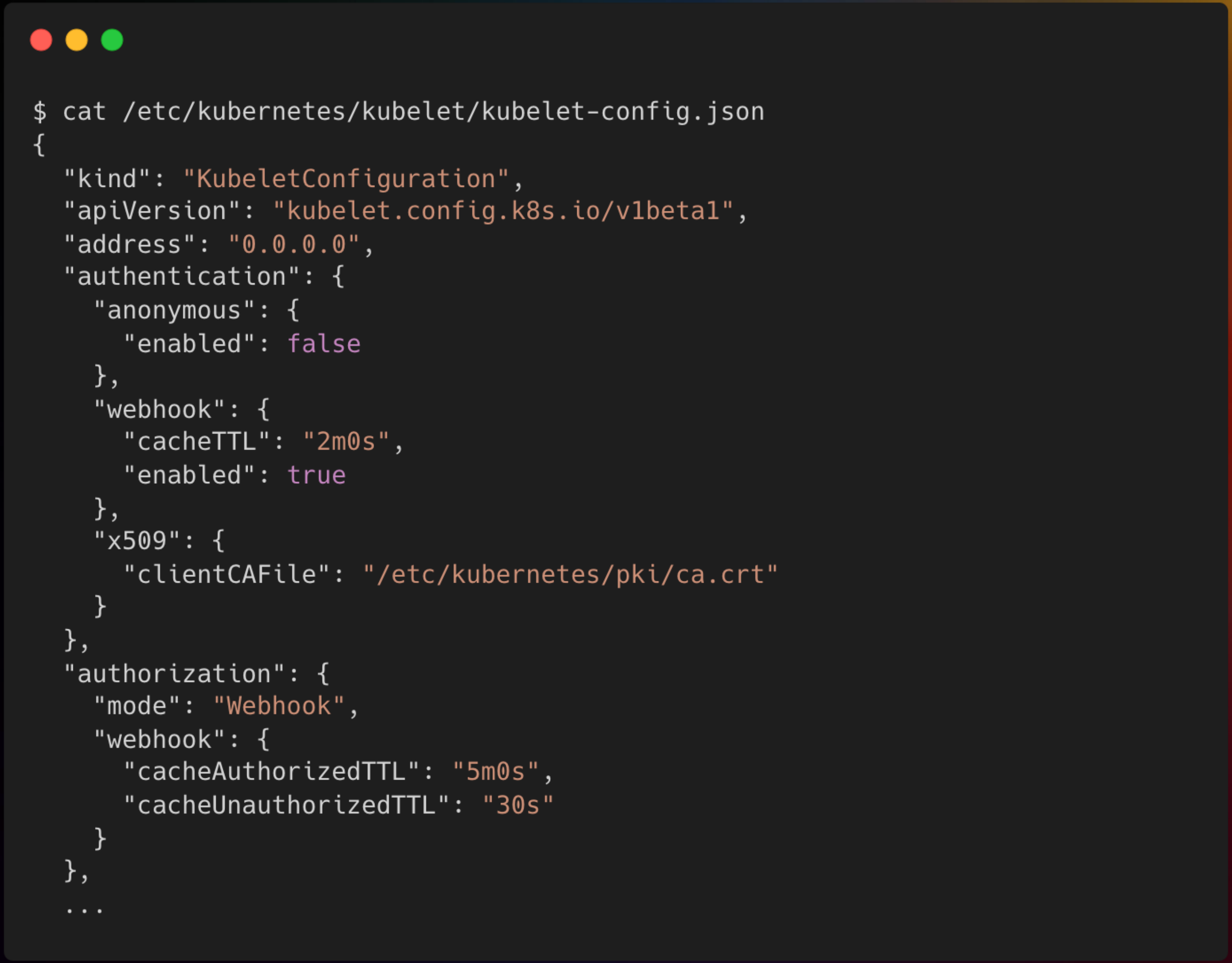
--authorization-mode string   Default: `AlwaysAllow`

Authorization mode for Kubelet server. Valid options are AlwaysAllow or Webhook. Webhook mode uses the SubjectAccessReview API to determine authorization. (DEPRECATED: This parameter should be set via the config file specified by the Kubelet's --config flag. See kubelet-config-file for more information.)

--authorization-webhook-cache-authorized-ttl duration   Default: `5m0s`

The duration to cache 'authorized' responses from the webhook authorizer. (DEPRECATED: This parameter should be set via the config file specified by the Kubelet's `--config` flag. See kubelet-config-file for more

Security misconfiguration – EKS defaults



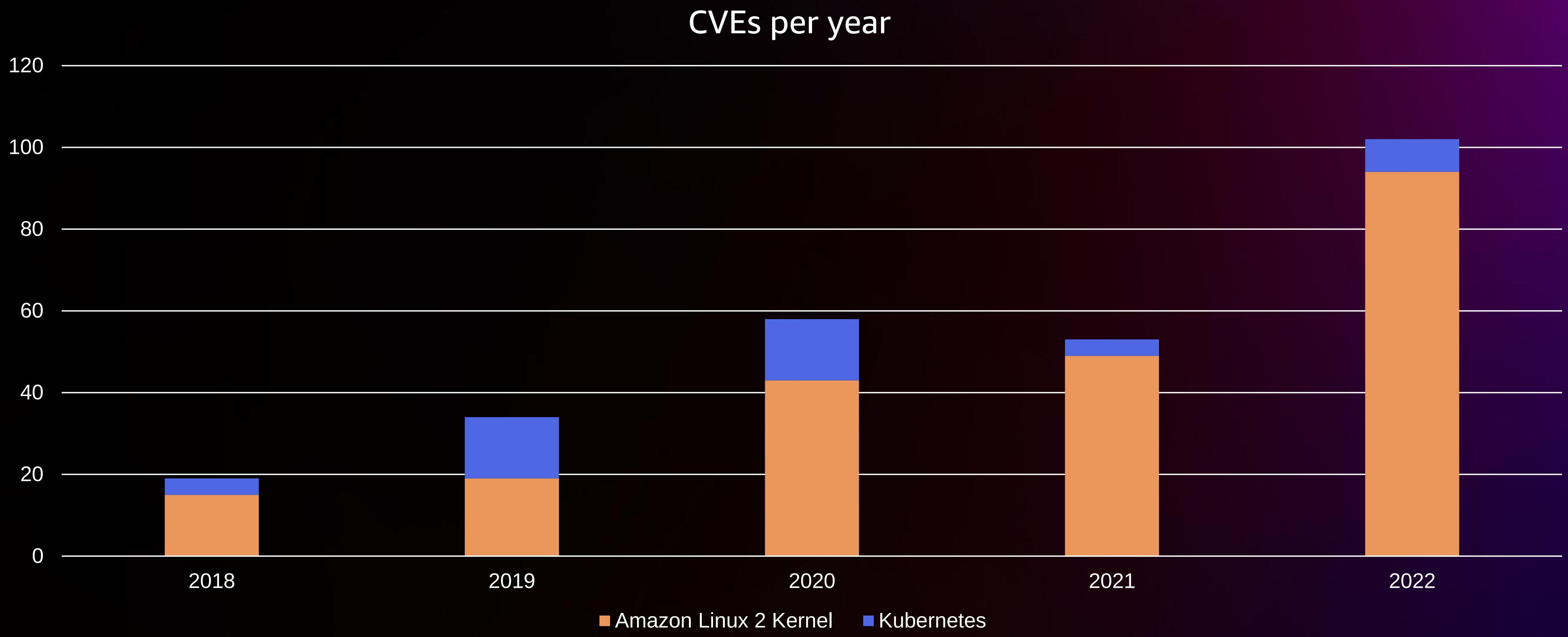
```
$ cat /etc/kubernetes/kubelet/kubelet-config.json
{
  "kind": "KubeletConfiguration",
  "apiVersion": "kubelet.config.k8s.io/v1beta1",
  "address": "0.0.0.0",
  "authentication": {
    "anonymous": {
      "enabled": false
    },
    "webhook": {
      "cacheTTL": "2m0s",
      "enabled": true
    },
    "x509": {
      "clientCAFile": "/etc/kubernetes/pki/ca.crt"
    }
  },
  "authorization": {
    "mode": "Webhook",
    "webhook": {
      "cacheAuthorizedTTL": "5m0s",
      "cacheUnauthorizedTTL": "30s"
    }
  },
  ...
```

Security misconfiguration – Mitigations

- Do not add users to the group `system:masters` in the aws-auth ConfigMap
- Limit and restrict host access from pods
- Use EKS provided defaults for Kubernetes components

6. Vulnerable or outdated components

Vulnerable or outdated components



Vulnerable component mitigations

- Keep machine and container images and applications up to date
- Keep your Kubernetes cluster on a supported version
  - https://docs.aws.amazon.com/eks/latest/userguide/kubernetes-versions.html

9. Security logging and monitoring failures

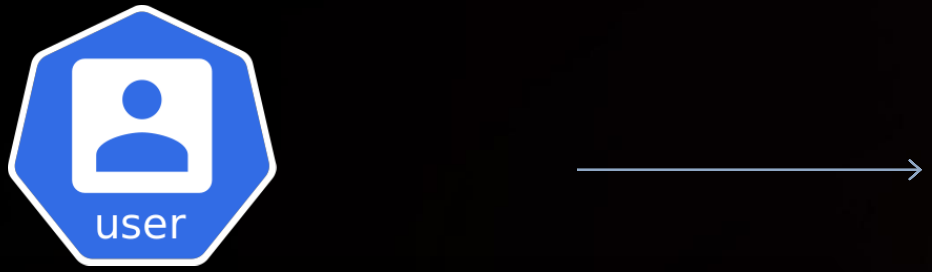
Logging and monitoring failure mitigations

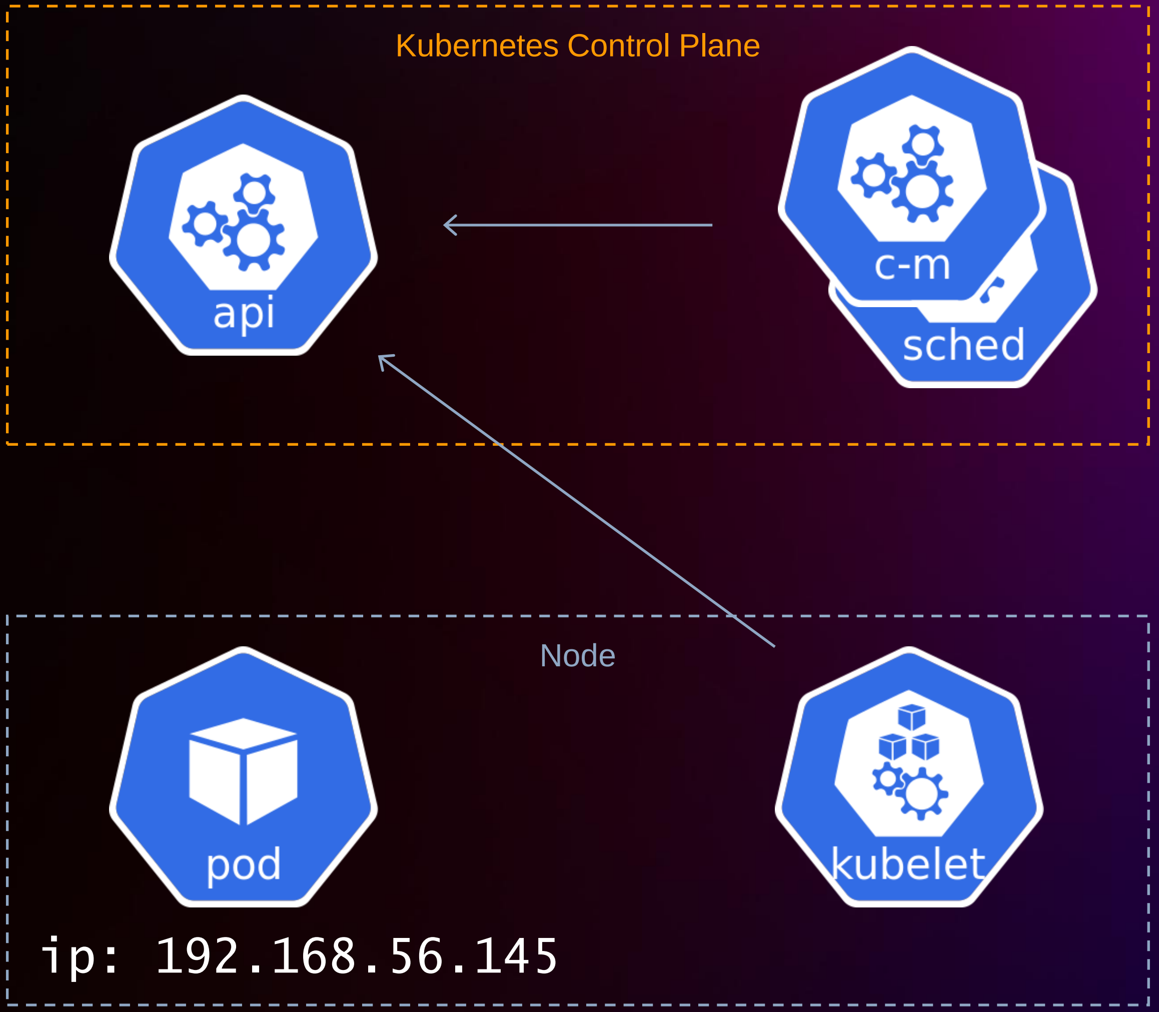
- Enable Kubernetes logging on all control plane components
- Export Kubelet logs off host

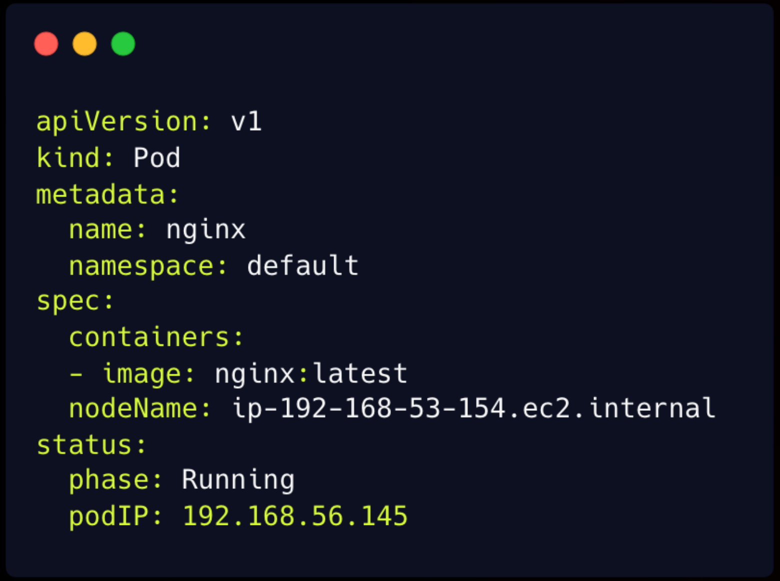
10. Server side request forgery

Server side request forgery

```
kubectl apply -f deployment.yaml
```



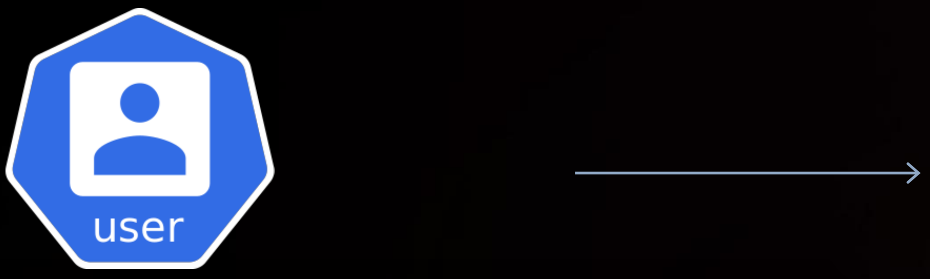


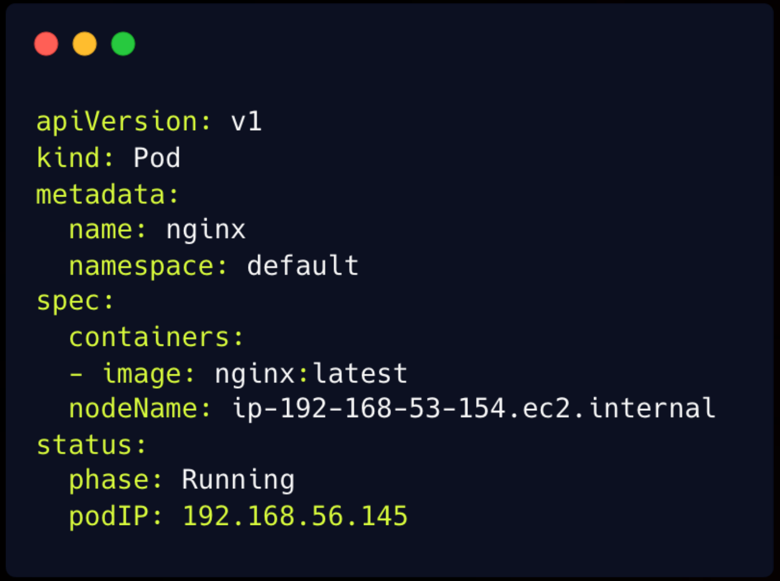


```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  namespace: default
spec:
  containers:
  - image: nginx:latest
  nodeName: ip-192-168-53-154.ec2.internal
status:
  phase: Running
  podIP: 192.168.56.145
```

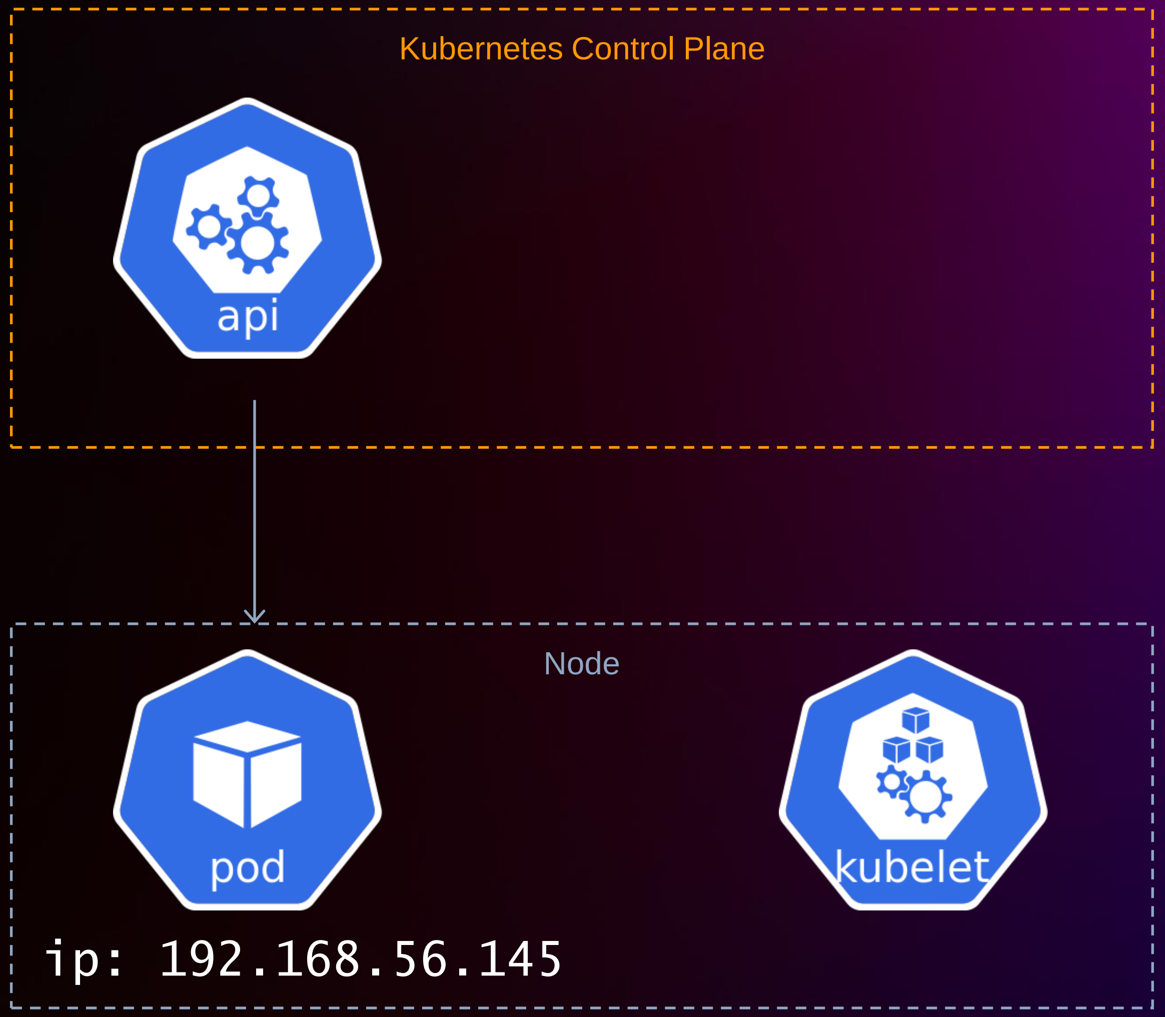
Server side request forgery

```
kubectl port-forward pod/nginx :80
```

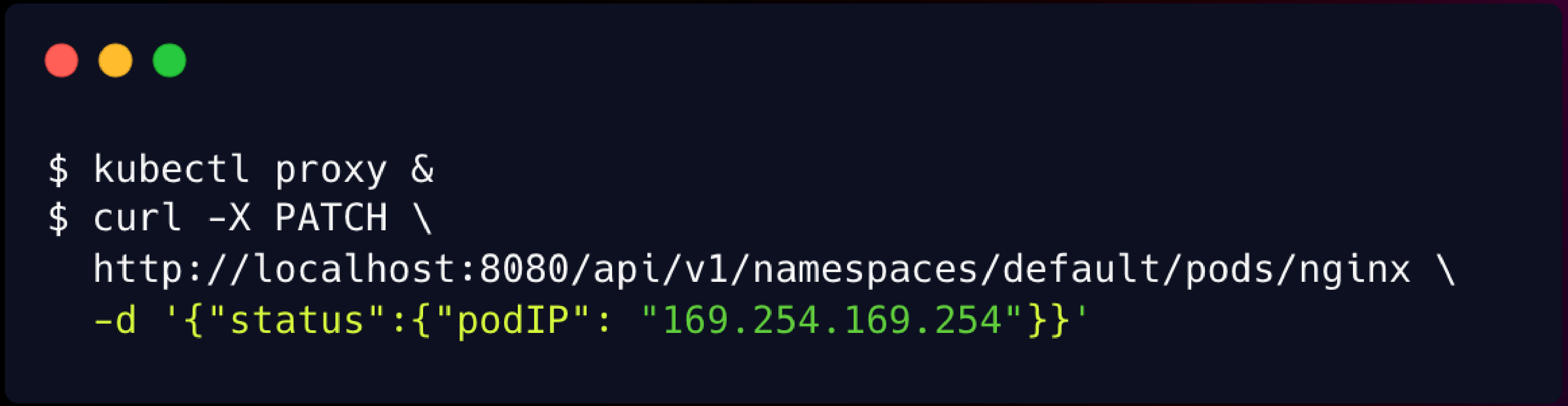




```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  namespace: default
spec:
  containers:
  - image: nginx:latest
  nodeName: ip-192-168-53-154.ec2.internal
status:
  phase: Running
  podIP: 192.168.56.145
```



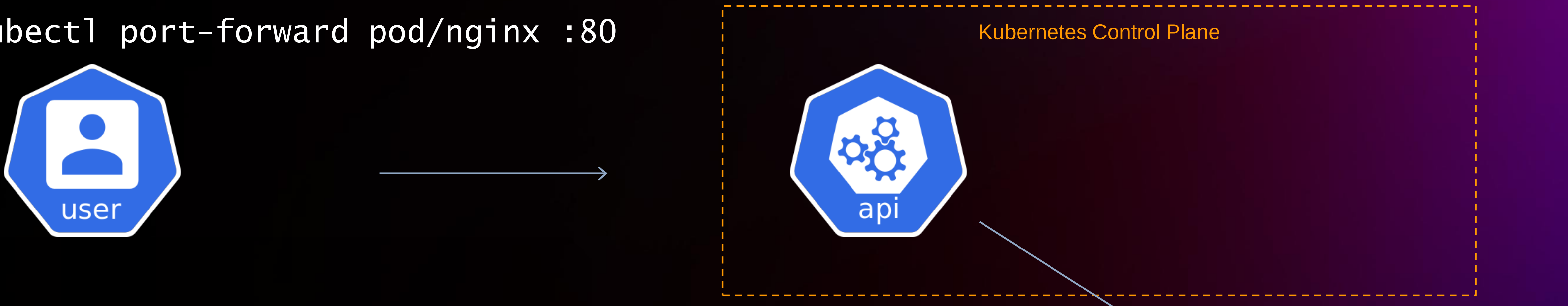
Server side request forgery

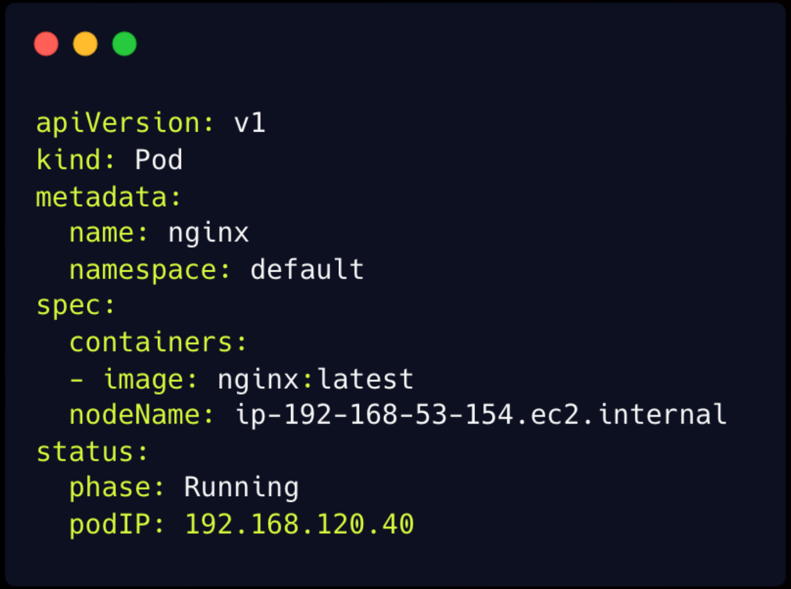


```
$ kubectl proxy &
$ curl -X PATCH \
  http://localhost:8080/api/v1/namespaces/default/pods/nginx \
  -d '{"status":{"podIP": "169.254.169.254"}}'
```

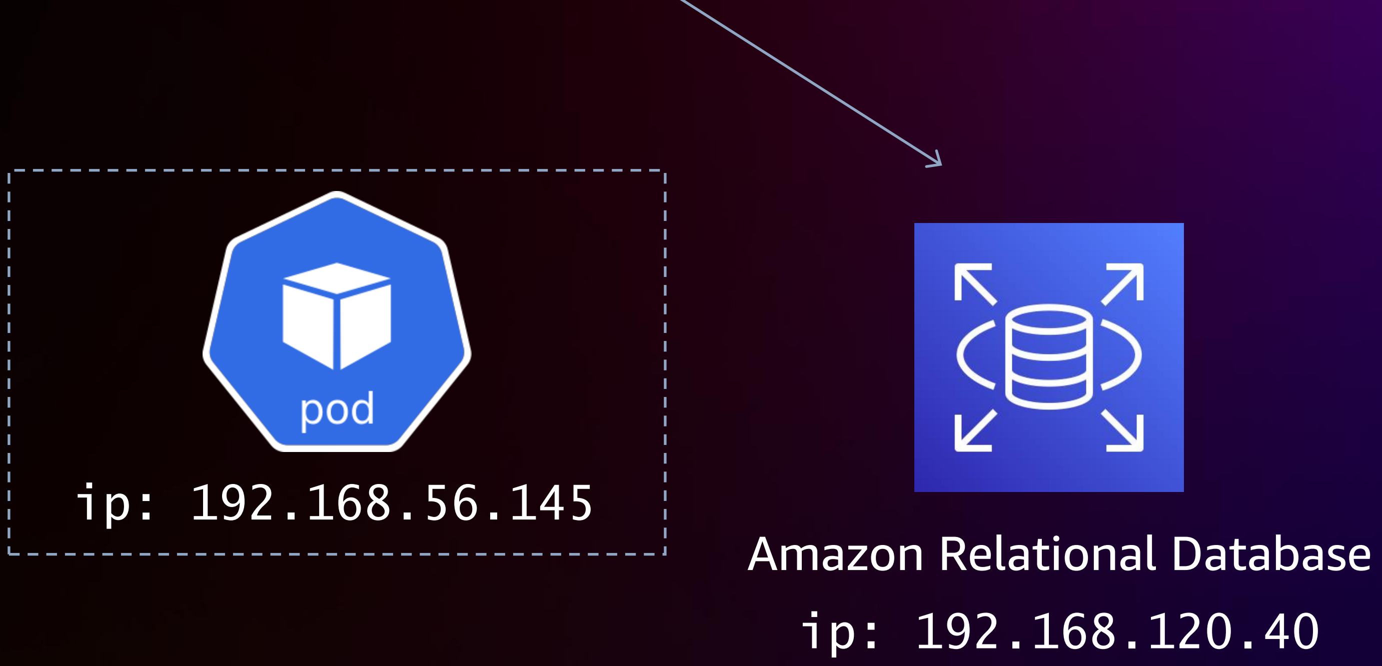
Server side request forgery

```
kubectl port-forward pod/nginx :80
```





```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  namespace: default
spec:
  containers:
  - image: nginx:latest
  nodeName: ip-192-168-53-154.ec2.internal
status:
  phase: Running
  podIP: 192.168.120.40
```



Server side request forgery – Mitigations

- Enable Kubernetes audit logging
  - Alert on non-node patching of pod status
- Limit Kubernetes API outbound access
  - Use appropriate Security Group rules with EKS
- Keep clusters up to date

How you can harden clusters with EKS

- Keep on top of Kubernetes cluster updates
  - Upgrade your Node Group AMI regularly
- Use AWS KMS encryption of Kubernetes secrets
- Disable the public-facing cluster endpoint if possible
- Enable Kubernetes audit logs with EKS
  - Use Amazon GuardDuty monitoring of Kubernetes API server logs
- Use IAM Roles for Service Accounts (IRSA) for Pod access to AWS APIs

How you can secure pods with EKS

- Use a Policy Enforcement engine
  - Use Pod Security Admission in Kubernetes >= v1.23 , replaces Pod Security Policy
  - Install Open Policy Agent (OPA) and Gatekeeper
- Refer to the EKS Security Best Practices Guide
  - https://aws.github.io/aws-eks-best-practices/security/docs/pods/

Thank you!

Micah Hausler

@micahhausler

Please complete the session survey in the **mobile app**