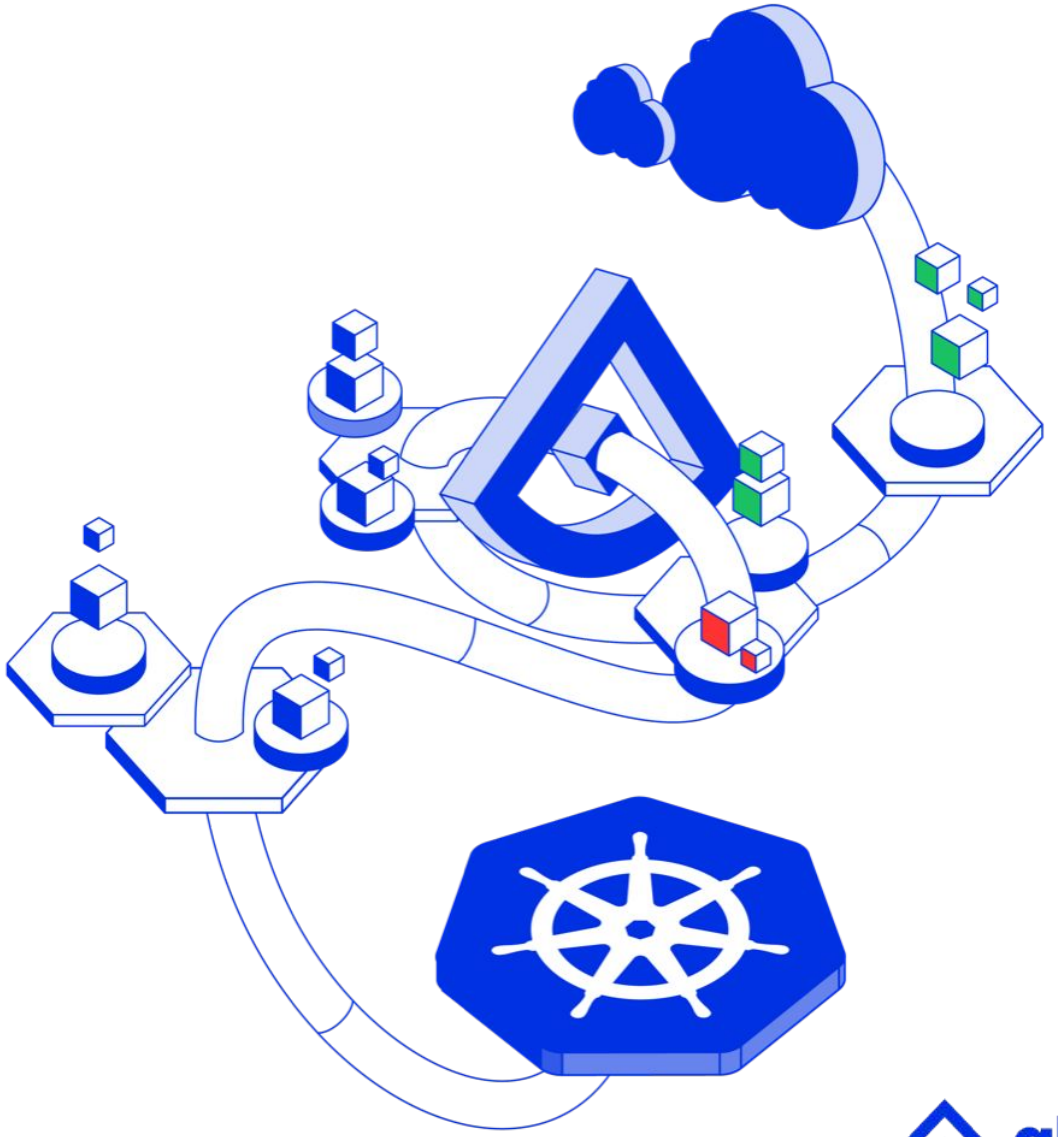
Kubernetes Audit Log //
Gold Mine For Security

Gadi Naor
CTO, Alcide
CNCF, Jan. 2020



Alcide kAudit

alcide

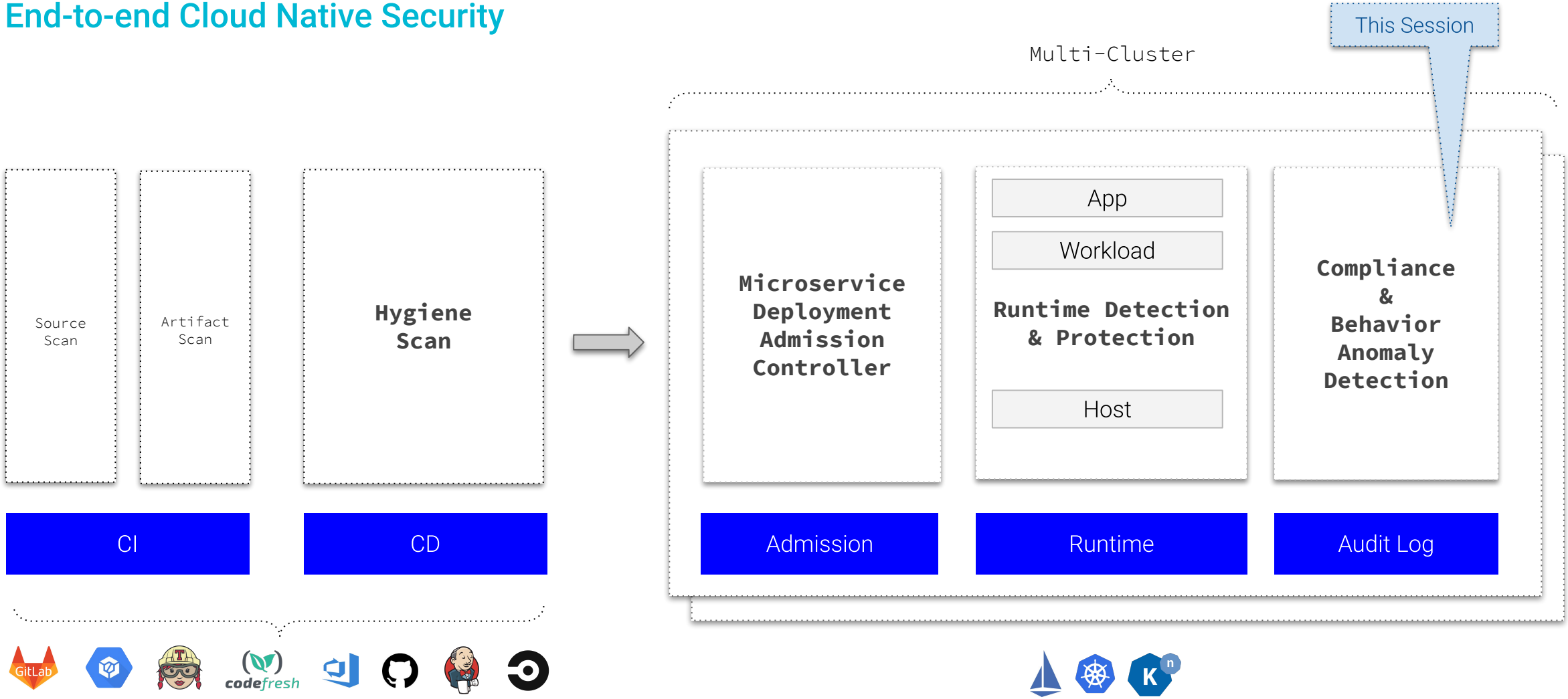
```
about# gadi.naor
  --from tel_aviv
  --enjoy sk8boarding
  --kernel-dev @check_point
  --kernel-dev @altor_networks
  --kernel-dev @juniper_networks
  --cloud-native @alcideio
  validate k8s cluster
```

@alcideio
@gadinaor

alcide

Kubernetes Security Framework

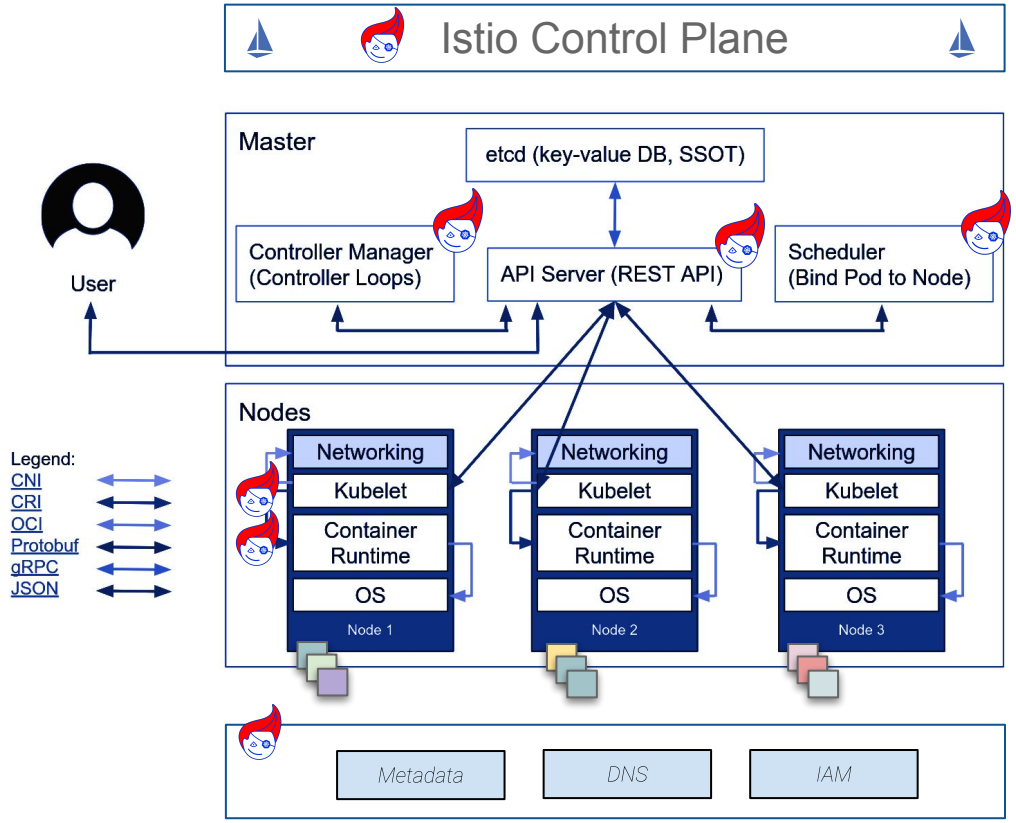
End-to-end Cloud Native Security



alcide

Kubernetes API Server

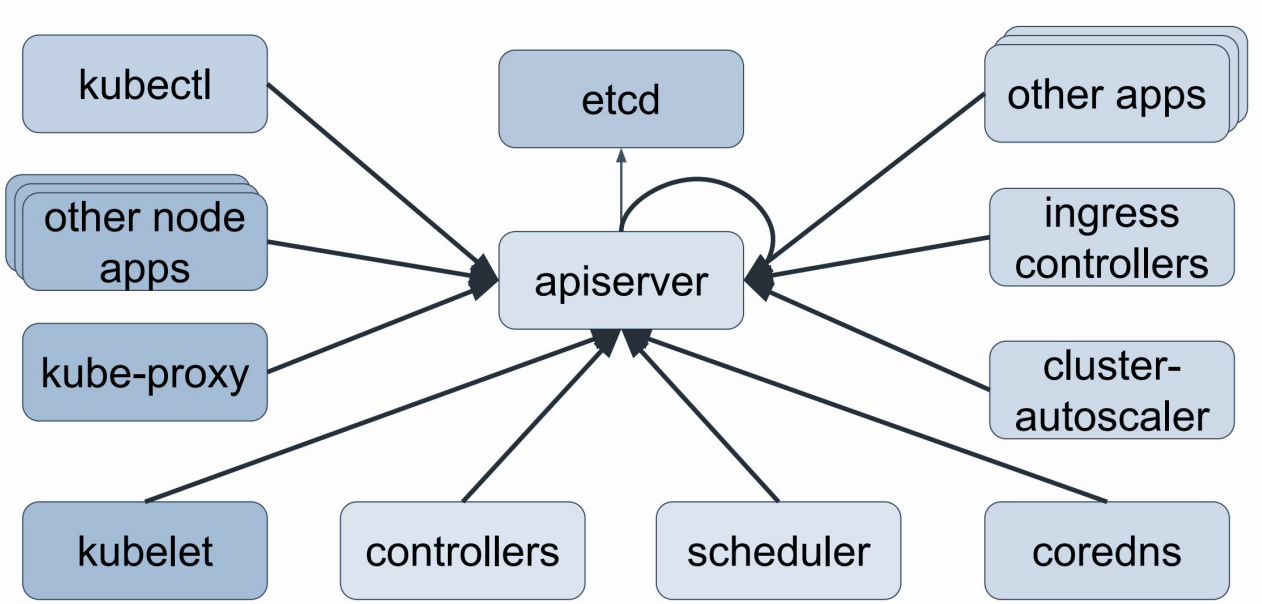
1. Accessed by users, controllers, operators, and components in k8s cluster (nodes)
   - Implements a RESTful API over HTTP
   - Performs all API operations
2. Request processing:
   - ***authentication***: establish the identity associated with the request (principal)
   - ***authorization***: determine whether the identity associated with the request can access the resource
     - Verb and HTTP path → RBAC
   - **admission control**: determine whether request is well formed (& potentially modifies it)
   - **validation**: ensure that a specific resource included in a request is valid



alcide

Who/What Access Kubernetes API Server

1. Users
   - SREs, DevOps,
   - Automation Pipelines
2. System Components
   - Nodes (kubelet), kube-proxy, DNS, scheduler
   - System Controllers (ingress, deployment, rs, ..)
3. Service Accounts
   - Cluster controllers/Operators
   - monitoring, security, logging agents
   - cluster apps

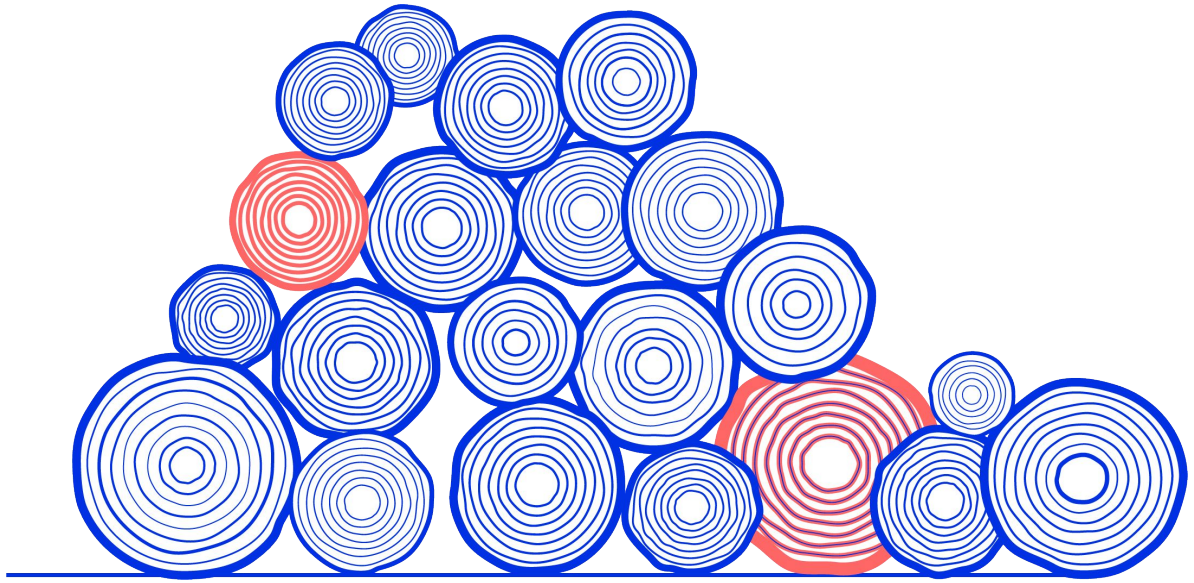


alcide

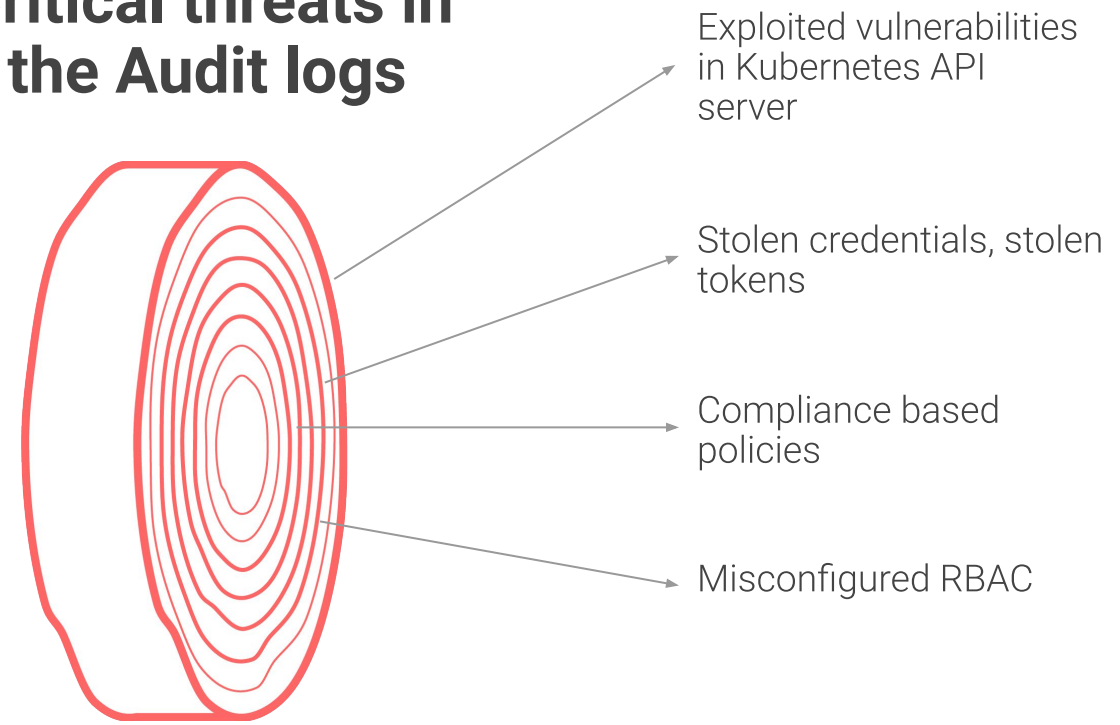
Kubernetes Audit Logs Analysis Made Easy

Detect specific insider threats in Kubernetes Audit logs

Raw Audit logs



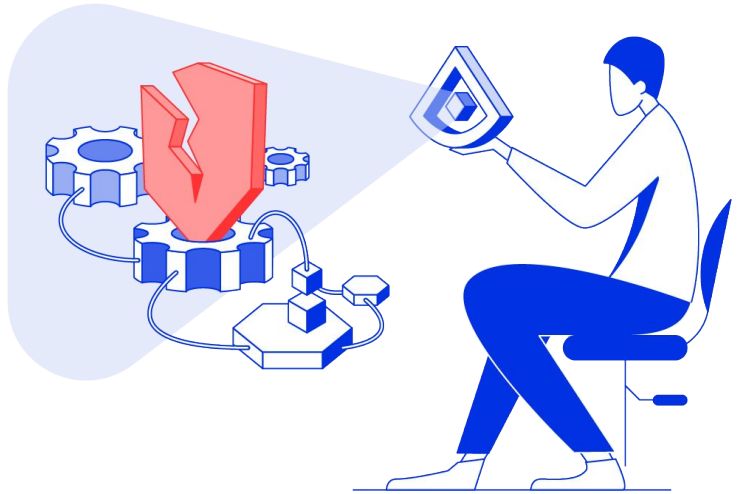
Critical threats in the Audit logs



alcide

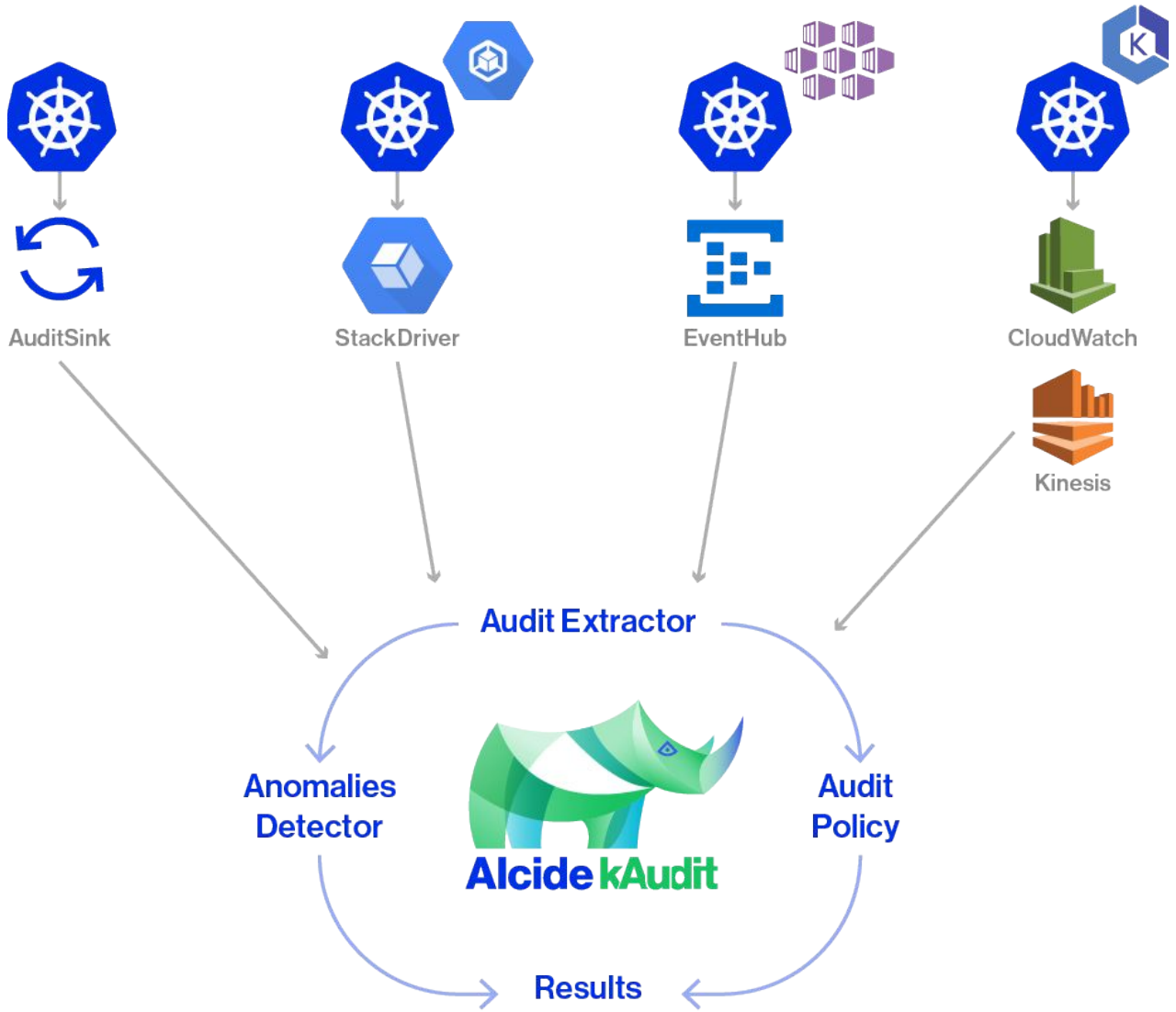
Building Audit Log Processing Analysis Pipeline

1. **Collect**
   - Create/Enable Audit Policy + Audit Log Collector
   - Webhook, GKE (Stackdriver), AKS (Event Hub), EKS (Cloudwatch)
2. **Filter & Transform → *Audit Features***.
   - Features → attributes we determined are “interesting” for analysis
   - Who (principal, IPs, UserAgent), What (Resource kind, name, namespace), Action (create, delete, update, ..)
3. **Feature Analysis**
   - Principal Profile, Resource Profile, Cluster Profile
   - Continuously updates a profile for each principal and resource
   - Anomaly is a significant deviations from the learned profile
   - Incident is a specific combination of concurrent anomalies on a principal or resource
4. **Report**
   - detected anomalies and incidents to user



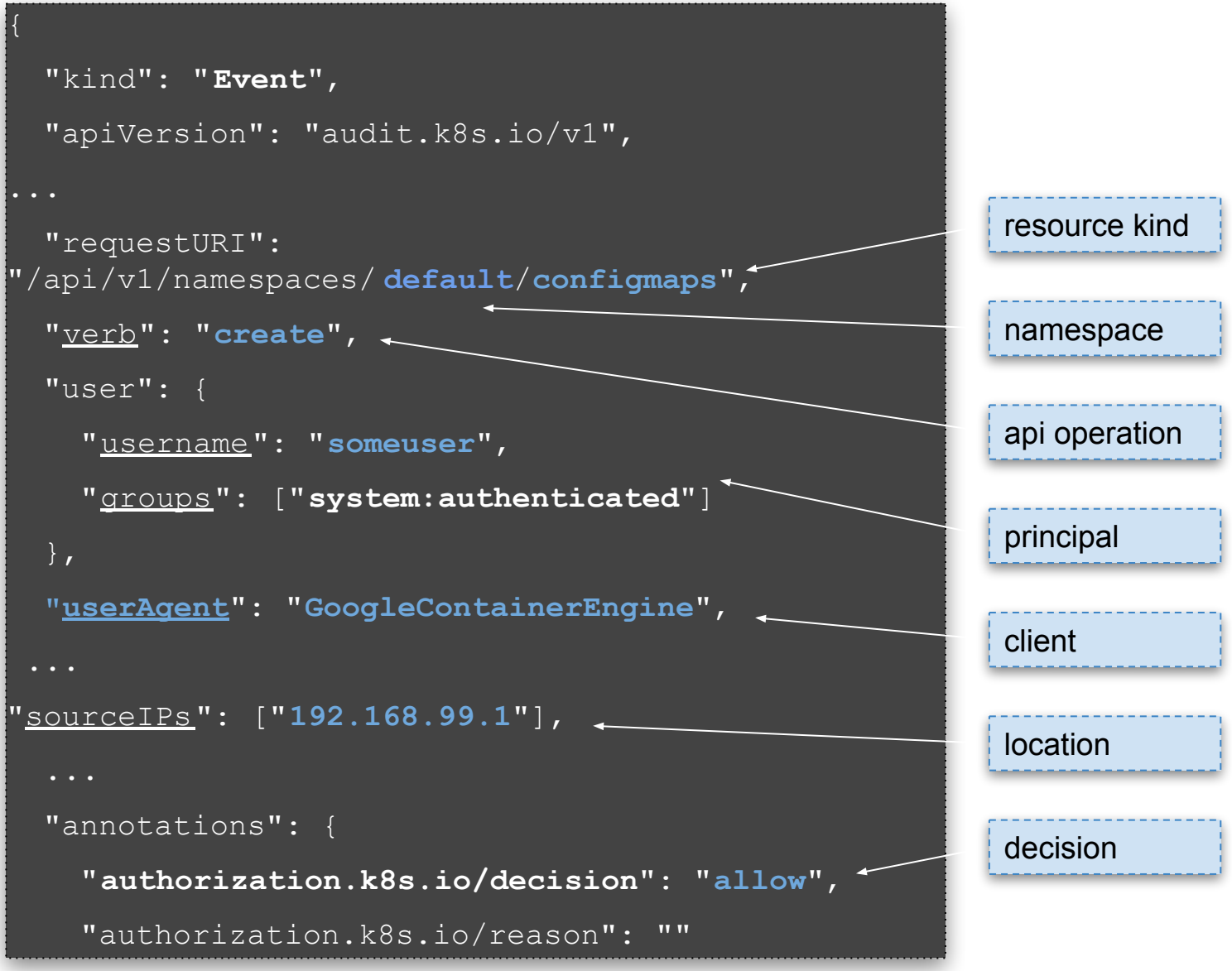
alcide

Kubernetes API Server Audit Log - Stream Analysis



alcide

Kubernetes API Server Audit Log



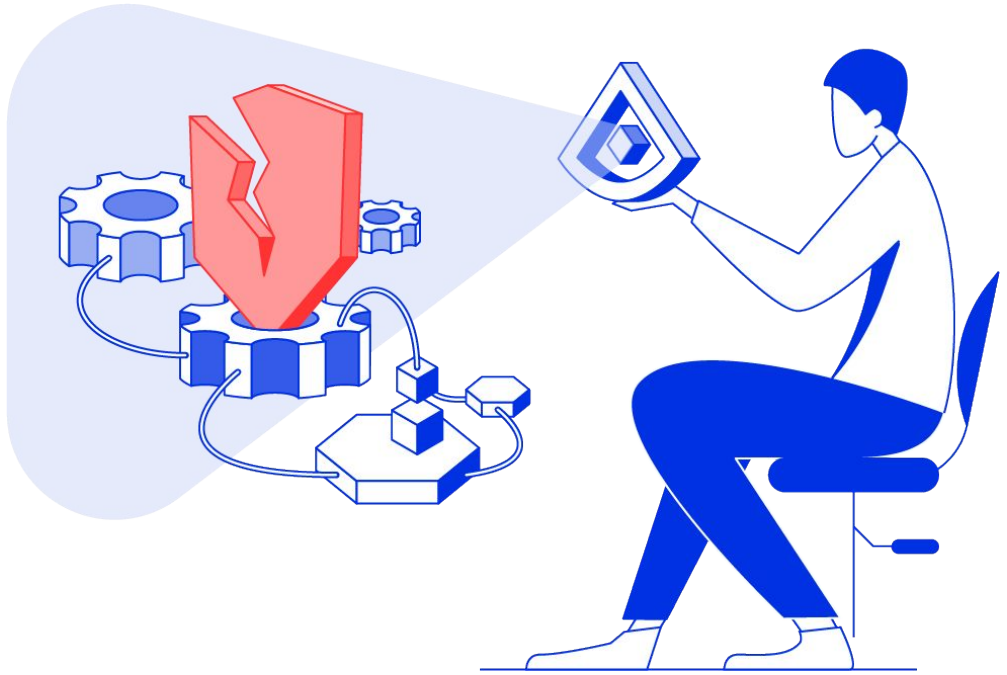
alcide

Use Case Examples | Troubleshooting

Cluster Troubleshooting

- **What:** Trigger alerts on detected failures
  - Check API Server Measure API MTTR
  - Identify System failures (Error Code:401 from Nodes)

  When:
  - Alert on MTTR degradation
  - Alert when System components (nodes) denied access

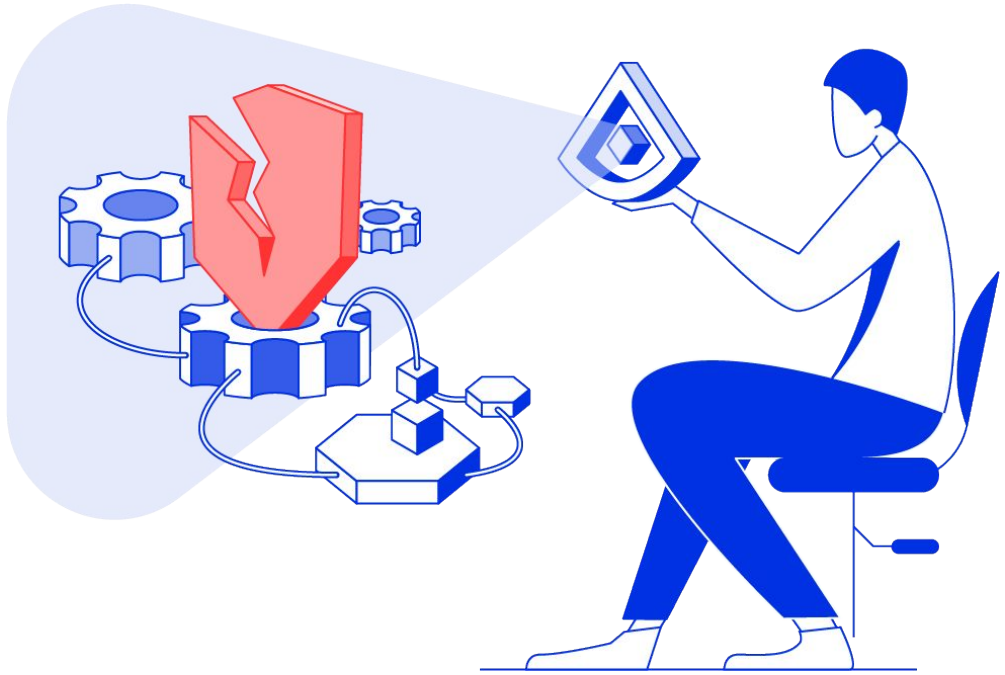


alcide

Use Case Examples | Pod Access

Alert On Pod Access

- **What:** Trigger alerts when (sensitive) Pod is being accessed
  - *PCI-DSS* audit trail
  - Internal SecOps guardrails
  - SRE Rules of engagement
- **When:**
  - *kubectl exec --it -n somens somepod -- bash*
  - *kubectl port-forward -n somens somepod 8443:8443*
  - *kubectl logs -n somens somepod*

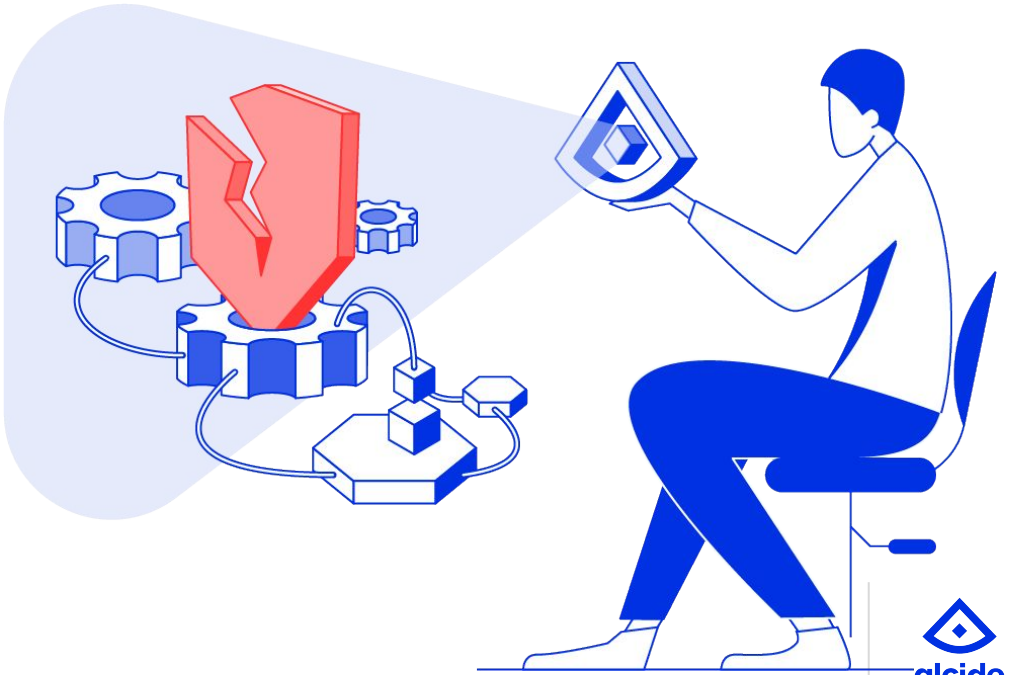


alcide

Use Case Examples | Unauthorized Access

Detect Unauthorized Access

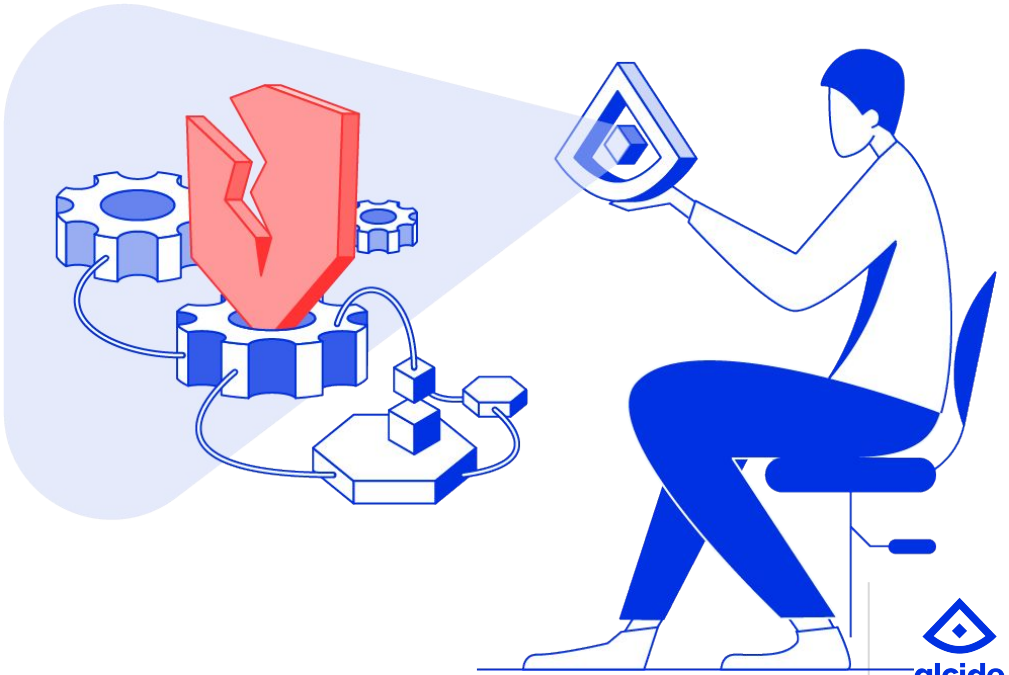
- **What:** Trigger alerts on unauthorized cluster access
  - Due to mis-configuration of k8s RBAC
  - Due to malicious or erroneous access to correctly restricted resources
- **How:**
  - Detected Anomalies on cluster (Error Code: 403)
  - Detected Anomalies on specific user/service-account



Use Case Examples | Credential Theft

Detect Credential Theft

- **What:** Trigger alerts on suspected credential theft
  - Cluster credentials reused from different ASNs in a short period
- **How:**
  - Detected Anomalies on specific user/service-account



Audit Log Should Not Be Used For

Use The Right Tool For The Right Task

Resource Checks

Use Admission Controllers to run resource checks, and take actions.

Performance Monitoring

Use prometheus & metrics to get top level API server health. Use the Audit Log for troubleshooting.

Workload Level Protection

Audit Logs tells access story of Pods to the API server, and have nothing to do with the application logic.

alcide

Let’s Take a Look

alcide

Conclusion

- Kubernetes Audit logs are incredibly valuable for Ops & Security
- Taking advantage of them require some effort.
- Policies are not easy to get right
- Audit log is verbose and require expert tools to analyze them

Stolen Token & Credentials

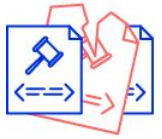
The result: Performing lateral movement, privilege escalation, data access and data manipulation while evading detection

Exploited Vulnerabilities in Kubernetes API Server

The result: Gaining access to privileged and sensitive resources (CVE-2018-1002105)

Misconfigured RBAC

The result: Performing lateral movement, privilege escalation, data access and data manipulation while evading detection



Compliance best practices

The result: Trigger alerts based on predefined rules violation(s)

alcide

Try Alcide Kubernetes Security

- ❑ Early Access Alcide kAudit alcide.io/kaudit-K8s-forensics
- ❑ Free Cloud Account www.alcide.io/advisor-free-trial/
- ❑ CD Integrations github.com/alcideio/pipeline
- ❑ Tutorials: codelab.alcide.io

@alcideio
@gadinaor

Alcide Advisor

Alcide kAudit

Alcide Runtime

alcide

Thank You

Alcide.io

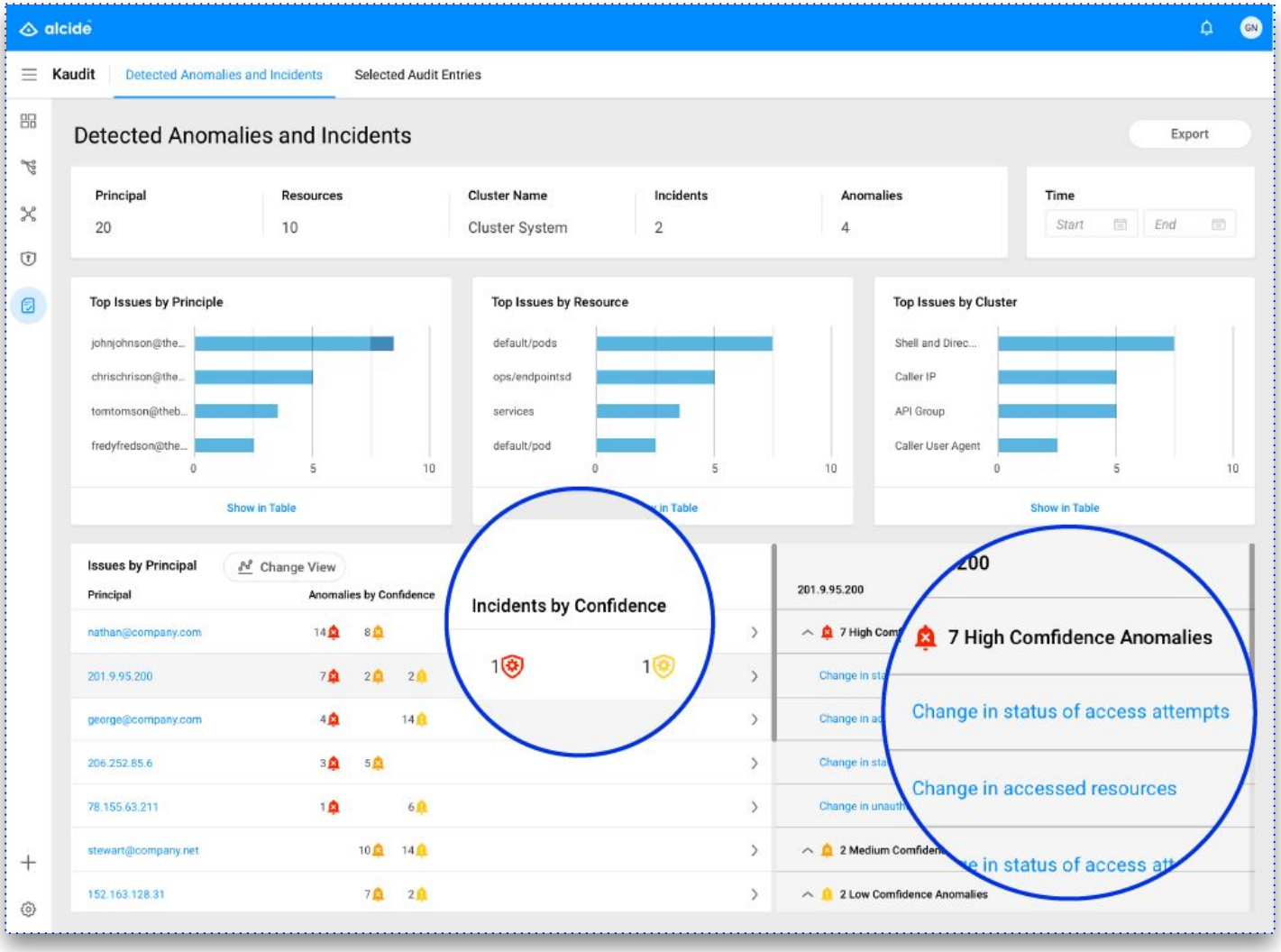
gadi@alcide.io

@alcideio

Example: Account Takeover Abuse

- *Change* in Access Attempt + *Change* in Access Resources:

  this collection of anomalies on the same user may indicate an account takeover abuse



Operators

Persistent Privileged Component

- Software that capture the key aim of a human operator who is managing a service or set of services
  - Example: Taking and restoring backups of that application's state
  - Example: Handling upgrades of application code
- Normally have **ClusterRole**
  - Example: Prometheus Operator - cluster-wide **Pod** list & delete, secret create, read, **Secret** get, create,..
  - Example: Strimzi Kafka Operator - cluster-wide **Pod** create, list, delete, update , **Secret** get, create,..
- Third Party Persistent & Highly Privileged Component
  - On compromise, represents a cluster wide threat

? Jobs vs. Controllers ?

`Pick Your Poison - https://operatorhub.io`

alcide