SERVICES AND RBAC ROLES

<https://payconiq.atlassian.net/wiki/spaces/INFRA/pages/3017965762/Platform+Tooling+Overview+2022>

====

Cluster Admin: Full access to all resources.

Ordinary admin: Manage all aspects like installing, upgrading and configuring deployments. They can also manage namespaces and service accounts.

Developer role: Can manage deployments of apps managed by Flux uisng GitOps principles. View config andtrigger deployments.

—=-====

Helm Chart: Cluster Admin, Helm Manager, Helm Developer, Helm viewer

Flux: Cluster Admin, Flux Admin, Flux Dev, Name

Prometheus: Cluster role: Full access, Scrape permissions for Prometheus.

Kibana (if 7.10): Space-based RBAC- Defines roles and permissions for isolated workspaces. Users are assigned specfic roles.

(if legacy): Defines roles and permissions at the cluster level.

**Fluentd: Y**ou can leverage Kubernetes RBAC to control access to configurations and resources. Here's how RBAC can be applied to Fluentd deployments:

Fluentd DaemonSet:

Service Account: Create a dedicated service account for the Fluentd DaemonSet. This service account will be used by all Fluentd pods within the set. ClusterRole: Define a ClusterRole that grants the service account the necessary permissions to perform its tasks. These permissions might include: Read access to configuration files stored as ConfigMaps or Secrets Write access to log destinations (e.g., Elasticsearch, Kafka) Limited network access (if Fluentd needs to communicate with external services) 2. User Access to Fluentd Configuration:

Secret Management: Store sensitive configuration data (e.g., API keys, passwords) used by Fluentd plugins within Kubernetes Secrets. Grant the Fluentd service account read access to these secrets. ConfigMap Management: Use ConfigMaps to store non-sensitive configuration data for Fluentd. The Fluentd service account should have read access to these ConfigMaps. Least Privilege Principle: Always grant the Fluentd service account the least privilege necessary to function. 3. Advanced Scenarios:

Namespace-Specific Roles: If you have multiple Fluentd deployments targeting different namespaces, consider defining separate ClusterRoles with limited access to specific namespaces. This enhances control and data segregation. External Access Control (Optional): If Fluentd needs to access external log sources with authentication, manage access control mechanisms on those external systems to further secure data collection.

**Pager Duty**

PagerDuty utilizes a permission-based access control system rather than traditional RBAC (Role-Based Access Control). This system grants users specific permissions to perform actions within the platform. Here's a breakdown of key concepts and suggested permission sets for PagerDuty:

Permissions vs. Roles:

Permissions: Granular controls that determine what actions a user can perform within PagerDuty. These can include: Viewing incidents and alerts Acknowledging, escalating, and resolving incidents Managing users and teams Creating and modifying services, escalations, and integrations Accessing billing information Teams: Groups of users with similar permissions. This allows for streamlined access management by assigning users to appropriate teams with predefined permission sets. Suggested Permission Sets:

Incident Responder: View incidents and alerts Acknowledge, escalate, and resolve incidents Add notes and collaborate on incidents within their team Service Manager: All of Incident Responder permissions Manage services (create, edit, delete) Configure escalation policies for services Admin: All of Service Manager permissions Manage users and teams Access billing information Configure global settings

PostgreSQL

Database Owner: Has full control over a specific database, including creating and dropping objects, granting permissions to other roles, and managing users. Application User: A role used by applications to connect to the database. Grant this role only the necessary permissions to access specific tables and perform required actions (e.g., SELECT, INSERT). Data Analyst: A role for users who need to read and analyze data from specific tables. Grant them SELECT permissions on relevant tables and views. Data Loader: A role for users or processes responsible for loading data into the database. Grant them INSERT permissions on specific tables. Backup Operator: A role with permissions to perform backups and restores of the database.

MongoDB

Organization Roles:

Global Admin: Has full access to all resources within the organization, including managing projects, users, and billing. Use this role with extreme caution and only for a limited number of users. Org Owner: Can manage projects, users, and billing within the organization but with slightly less privilege than a Global Admin. Read-Only User: Can view organization information and project details but cannot make any changes. Project Roles:

Cluster Admin: Has full control over a specific cluster within the project, including managing databases, users, and security configurations. Database Admin: Can manage databases within the cluster, including creating, modifying, and deleting databases, as well as managing user access to those databases. Database Owner: Owns a specific database within the cluster and can grant access to other users. DB Reader: Can read data from specific databases but cannot modify or delete data. Backup Operator: Has permissions to perform backups and restores of the cluster.

Dynamo DB

IAM Policies for DynamoDB:

IAM Users and Roles: Create IAM users or roles for your applications or users who need to access DynamoDB. IAM Policies: Define IAM policies that specify the allowed actions (e.g., PutItem, GetItem, DeleteItem) on DynamoDB tables and resources. These policies can be attached to IAM users or roles. Conditions (Optional): IAM policies can include conditions to further restrict access based on factors like source IP address, time of day, or specific attributes within DynamoDB items.

Vault+Consul

Note: Vault typically use policies

Vault Admin: Description: Has full access to manage Vault configuration, users, policies, secrets engines, and auth backends. Policy Considerations: Grant broad permissions across all paths with capabilities like read, write, list, sudo, and delete. Secrets Manager: Description: Manages secrets within Vault, including creating, reading, updating, and deleting secrets. Policy Considerations: Grant access to relevant secrets engines (e.g., kv) with capabilities like read, write, and list. Limit access to specific paths based on requirements (e.g., /secret/data/production for production secrets). Application User: Description: Allows applications to access secrets required for their operation. Policy Considerations: Define least privilege policies granting read access only to the specific paths containing the secrets the application needs (e.g., /secret/data/app1). Auditor: Description: Can view secrets or specific aspects of Vault for auditing purposes. Policy Considerations: Grant read-only access to relevant paths, potentially with conditions restricting access based on metadata (e.g., only secrets with the label audit=true). Policy Writer: Description: Creates and manages Vault policies but cannot directly access secrets. Policy Considerations: Grant write access to the /sys/policy path for creating and updating policies.

Cloudflare

Roles:

Super Administrator:

Description: Has full access to manage all aspects of the Cloudflare account, including DNS records, security settings, billing, and user management. Permissions: Full read/write access to all configurations and resources within the account. Use Cases: Only for a limited number of trusted users for critical account management tasks.

Domain Owner:

Description: Manages DNS records, security settings (WAF, Firewall), and other configurations for specific domains within the account. Permissions: Read/write access to DNS records, security settings, and other relevant configurations for assigned domains. Use Cases: Website owners, IT administrators responsible for specific domains.

Security Admin:

Description: Focuses on configuring and managing Cloudflare security features like WAF rules, Firewall settings, and bot management. Permissions: Read/write access to security configurations (WAF, Firewall, IP rules), read-only access to DNS records for understanding traffic flow. Use Cases: Security specialists responsible for mitigating threats and protecting web applications.

DNS Manager

Description: Manages DNS records for assigned domains, including creating, editing, and deleting records. Permissions: Read/write access to DNS records for assigned domains, limited access to view other configurations (optional). Use Cases: DNS administrators responsible for managing domain name resolution.

Analytics Viewer:

Description: Can view analytics data and reports generated by Cloudflare for specific domains. Permissions: Read-only access to analytics dashboards and reports for assigned domains. Use Cases: Marketing teams, analysts who need to track website traffic and performance.

Billing Manager:

Description: Manages billing information and payment methods associated with the Cloudflare account. Permissions: View billing details, manage payment methods (with restrictions). Use Cases: Finance or accounting personnel responsible for managing Cloudflare subscriptions.

Kafka Connect:

Suggested RBAC Roles for Kafka Connect:

Connect Admin: Description: Has full access to manage all aspects of Kafka Connect, including creating, deleting, configuring, and monitoring connectors. Permissions: All permissions on Connect resources (create, delete, configure, view, manage tasks). Connect Developer: Description: Develops and manages their own connectors but cannot access or modify connectors created by others. Permissions: Create, delete, configure, view, and manage tasks for their own connectors. May have read-only access to view configurations of other connectors (optional). Connect Monitor: Description: Can view the status and configuration of connectors but cannot create, delete, or modify them. Permissions: View permissions for all connectors (configurations, status).