Segregation of Duties in Multi-Tenant Cloud Native Environments

Solution Brief

aqua

The Need for Least Privilege Security Access

Enterprise environments often consist of multiple teams working on different cloud native projects and applications. Each such team will work on their own assets such as container images, functions, and use separate CI pipelines, yet in the end they will often run on the same cloud infrastructure. Organizations differ in how they choose to separate these projects. Most often, different applications will run on separate Kubernetes namespaces or clusters.

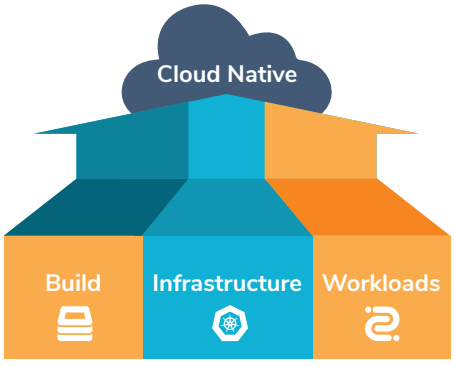
From a security standpoint, the cloud native stack (including Docker, Kubernetes, and other services) makes it challenging to manage the type of access and capabilities granted to each user, even more than traditional environments due to the scale, granularity of components, and velocity of updates. For example, if developers are expected to be notified of vulnerabilities in their code and fix them, they need access to that information. On the other hand, they should not be able to acknowledge and accept the risk of a vulnerability – that should be left to security specialists. Furthermore, teams that work on different applications should only be able to see and act on security issues that pertain to their own applications, and not be able to see any security information related to other teams' applications.

No individual should have excessive system access that enables them to execute actions across all projects or have permissions that exceed their role within their project. Segregation of Duties (SoD) is a tenet of any security strategy and is also a requirement in many compliance regulations. Not implementing it properly presents risk to the business, whether by creating opportunities for user error, privilege escalation by an external intruder, or enabling a malicious insider.

However, in order to manage enterprise security effectively, organizations look for centralized control of their security policies and enforcement. They expect to be able to get visibility and apply consistent policies across applications, teams, pipelines, clusters, and clouds. So how do we reconcile these requirements for centralized control on the one hand, but granular segregation-of-duties on the other?

Flexible, Scalable Multi-Tenant RBAC with Aqua

Aqua provides a full-lifecycle security platform for cloud native applications. We broadly categorize the stack and processes we secure into three main pillars:



Artifacts - images and stored functions that are pushed through the DevOps pipeline

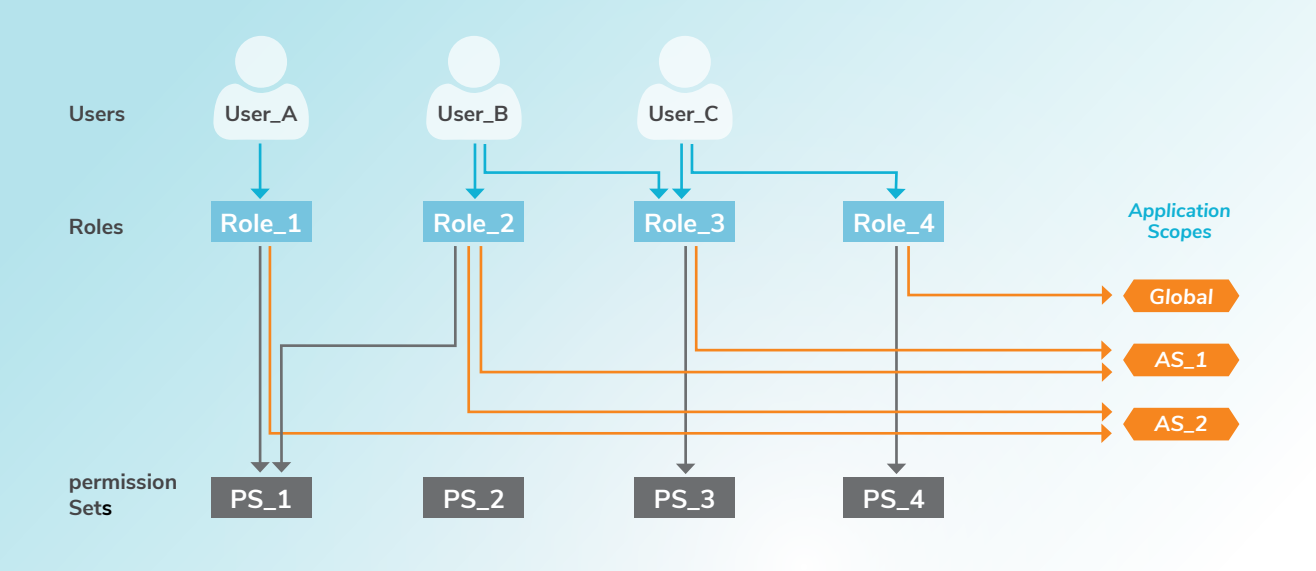
Infrastructure - cloud services, orchestrators and hosts where cloud native applications run

Workloads - containers, functions and VMs that make up the application runtime

Our aim is to provide granular access and control over elements within each pillar, while maintaining separation between teams and roles.

Aqua's Multi-Application RBAC Model

First, let's review the elements of Aqua's RBAC model and understand their relationships:



- **Users** All Aqua CSP operations require an authenticated Aqua user. Users can be added manually, though typically they will use SSO and mapped from LDAP/Active Directory. The user's role determines, at a very granular level, which resources the user can access and which operations the user can perform. Each domain user can be assigned to one or more roles.

- **Permission Sets** A Permission Set define what actions can be performed in the Aqua console on specific objects and features, whether in Read Only or Edit capacity (via the UI, REST APIs, and/or command line).

- **Application Scopes** An Application Scope specifies a set of system resources (application artifacts, workloads, and elements of cloud native computing infrastructure such as namespace, hosts, or clusters) that can be accessed (viewed or edited) by users whose roles are associated with the scope.

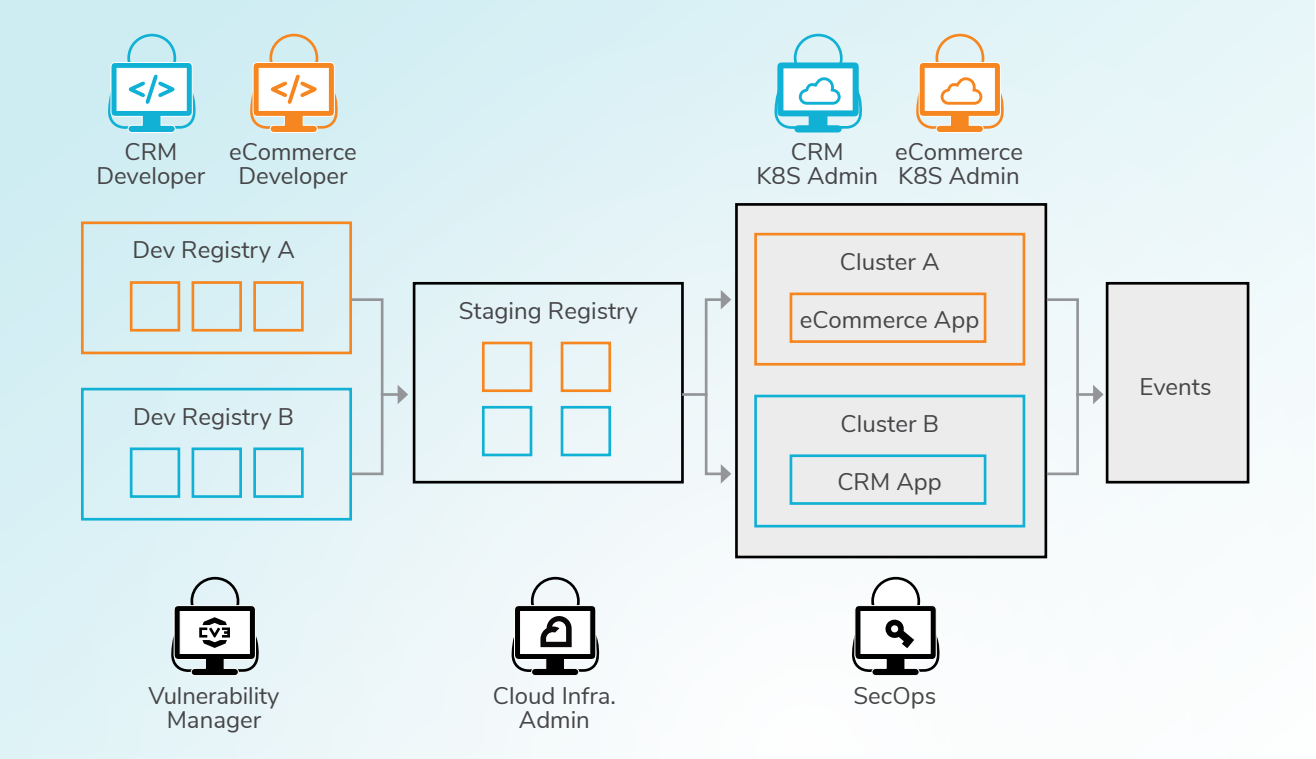
- **Roles** A role is a combination of a single Permission Set and one or more Application Scopes. It is assigned to a single user or an Active Directory group. A user can be assigned a single role or multiple roles. Each role has Permission Sets for the use of specific Aqua functionality. In addition, a role can be assigned to one or more Application Scopes.

Aqua's Multi-Application RBAC was built to be highly flexible and support virtually any organizational structure. It enables separation of teams and roles, but at the same time enables a hierarchy with managerial oversight. It also recognizes the fact that organizations are often not strictly hierarchical, with some users wearing "multiple hats", and can support that. For example, if the security team has two vulnerability specialists that split the organization's applications between them, each owning several – they can be given such access, while maintaining separation and keeping themselves accountable, each to his own set of applications.

Multi-Application Role-Based Access Control in Action

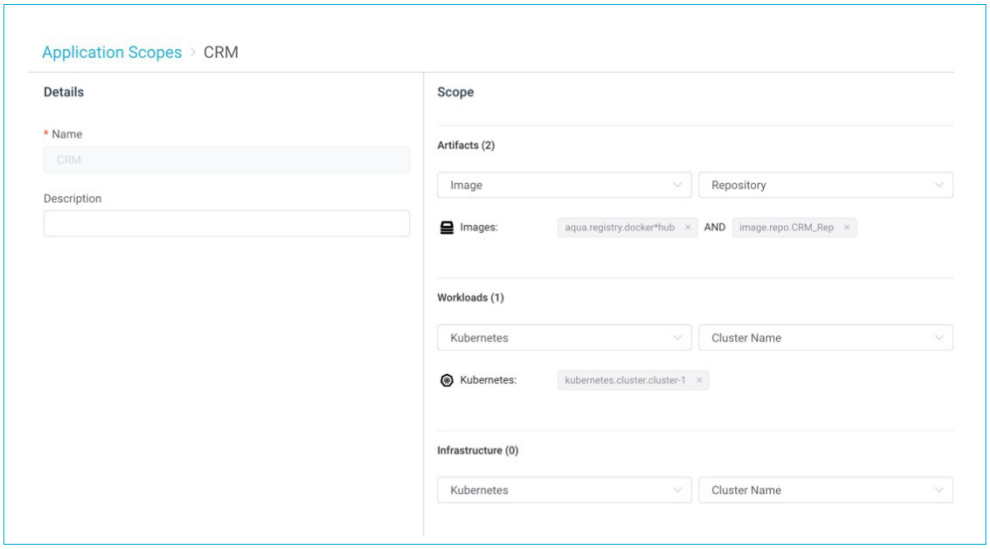
Let's look at a sample environment, where an organization has two teams running a CRM and an eCommerce application. Each application is using a separate registry for development, but a joint registry for staging. As we can see, the applications are running in separate clusters, though some organizations might run them in a single cluster but in separate namespaces.

The teams also have a manager who needs visibility into both environments, and different stakeholders within the teams require access to certain things – for example, developers need to see vulnerabilities, but not runtime security events. Cloud Ops might need to see K8s compliance information, but not vulnerabilities, and they certainly should not have control over security policies (or even see what security policies are in place).

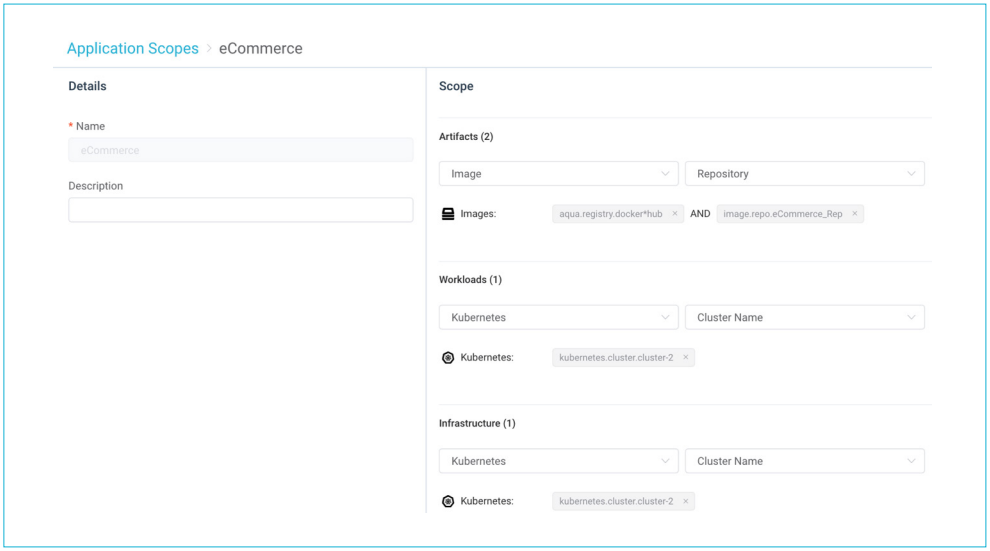


Let's see how we can define access and permissions within Aqua deployment while upholding separation of duties between application teams, and maintaining least privilege of permissions by role. The first step is to define the application scopes for each application:

- The CRM Application Scope will have visibility to Images from Docker Hub, repository of the CRM application, and to Cluster 1

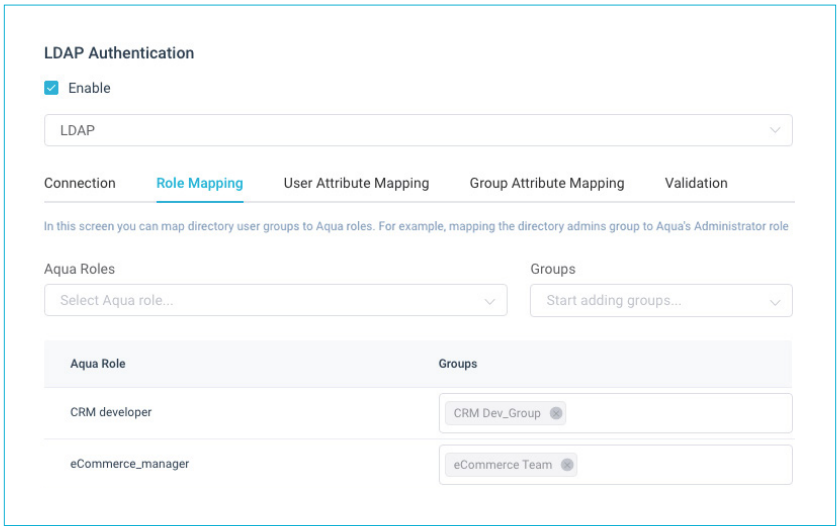


- The eCommerce Scope incorporates resources from the same Registry but a different Repository workloads from Cluster 2 and also the resources that are running on the Host of Cluster 2 (e.g., Kube-Hunter pen testing results and more)

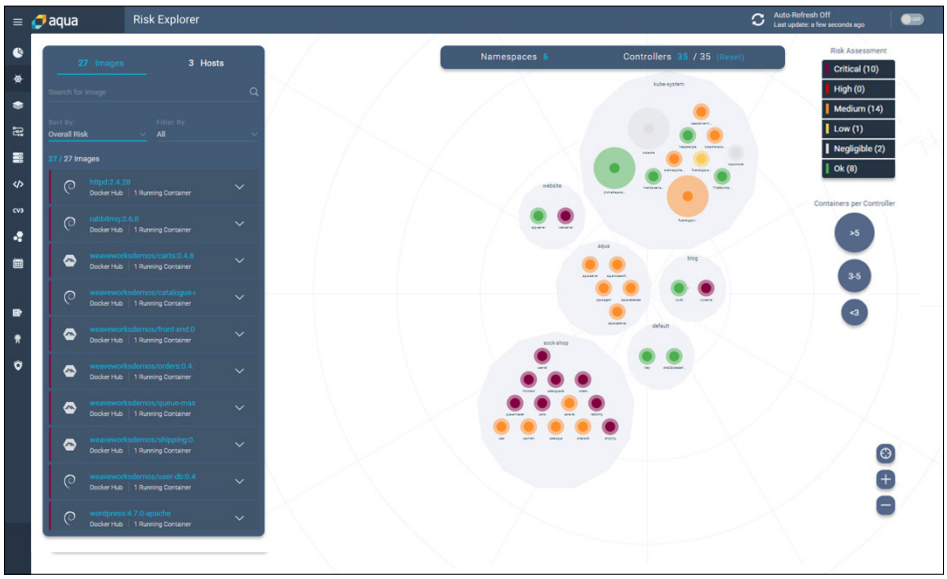


- The users are LDAP users that we sync from the organization's Active Directory. Each object that we sync from Active Directory (Users / Groups) is assigned an Aqua Role that is linked to an Application Scope and Permission Sets

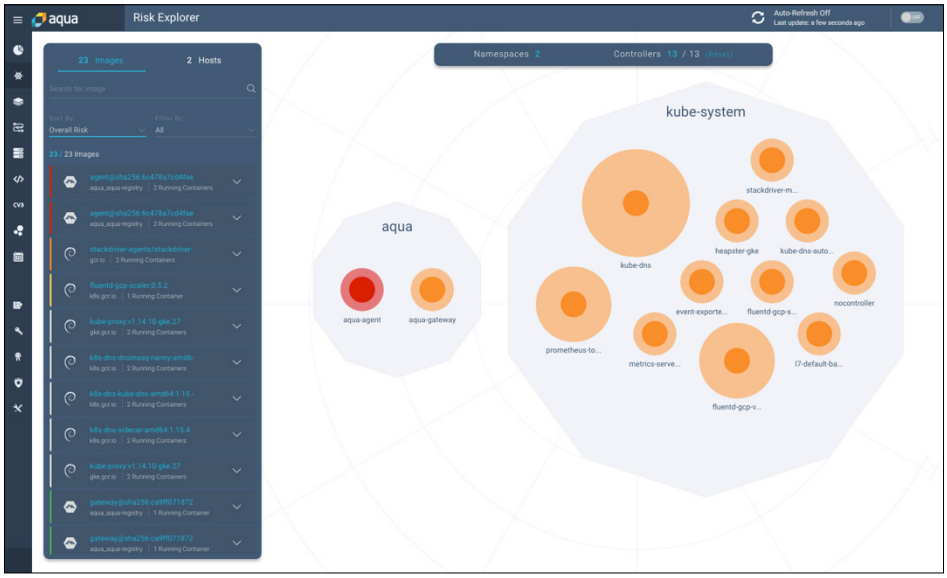
Each Aqua Role is linked to the relevant LDAP group:



The CRM Developer will see this:



While the eCommerce Manager will see this:



A Flexible RBAC Model to Secure Any Environment

Aqua enables organizations to define access and permissions within their Aqua deployment that maintain separation of duties between application teams, while maintaining least privilege of permissions by role. This results in a highly effective way of managing security policies and processes across heterogenous teams, while enabling DevSecOps in highly dynamic cloud native environments.

Separation Between Application Teams

- ✓ Access to all elements of an application is limited to relevant stakeholders
- ✓ Application scopes includes artifacts, infrastructure, and workloads

Least Privilege Permissions and Roles

- ✓ Roles can be mapped from AD/LDAP
- ✓ Granular permission sets per role that provide view/edit access to Aqua policies, events, assets, and system components
- ✓ Preset and custom roles

Flexible Security Hierarchy

- ✓ Centralized view of all application scopes
- ✓ Define central policies that are read-only for subordinate teams
- ✓ Users can be assigned more than one role to enable cross-functional teams

Get a Demo›

About Aqua

Aqua Security helps enterprises secure their cloud native, container-based and serverless applications and infrastructure from development to production. Aqua bridges the gap between DevOps and security, promoting business agility and accelerating digital transformation.
Aqua's Cloud Native Security Platform provides full visibility and security automation across the entire application lifecycle, using a modern zero-touch approach to detect and prevent threats while simplifying regulatory compliance. Aqua customers include some of the world's largest financial services, software development, internet, media, hospitality and retail companies, with implementations across the globe spanning a broad range of cloud providers and on-premise technologies.

contact@aquasec.com
www.aquasec.com
@AquaSecTeam
AquaSecTeam