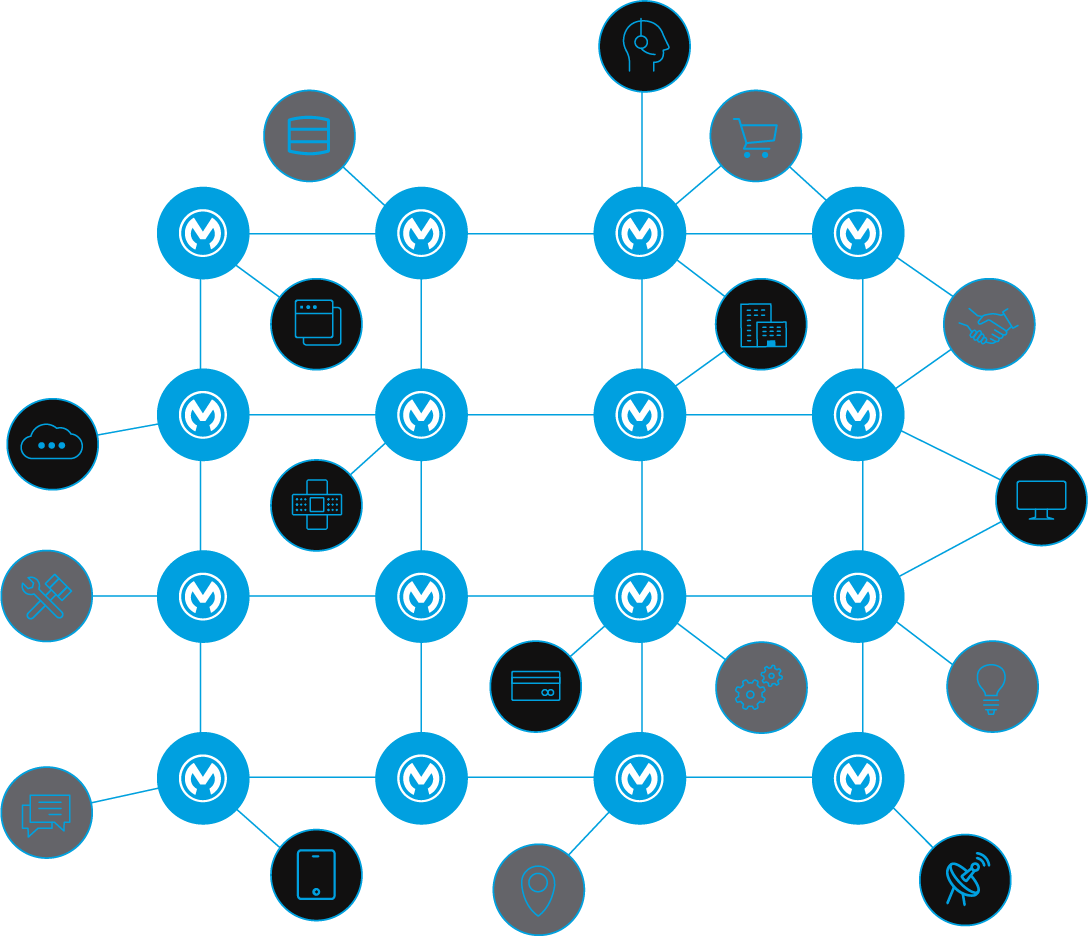
<Customer Name>

Anypoint Platform Teams Proposal



|  |  |  |  |
| --- | --- | --- | --- |
| **Change Record** | | | |
| **Template Version** | **Date** | **Author** | **Description** |
| 0.1 | 19-05-2022 | Max Mule | Initial Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 

[**Overview**](#_e4x5823g9sw5) **4**

[Target Audience](#_1t3h5sf) 4

[**An Introduction to Teams**](#_22fknzxttd80) **5**

[**Planning Teams**](#_j0txfc9rcrvd) **5**

[**Team Maintenance**](#_expt3fqb7fks) **6**

[**Recommended Teams Structure**](#_x1olda75sod4) **6**

[**Business Unit**](#_f0eavcaecdjp) **6**

[**Organization Administrators**](#_v7kry4v1jhqk) **7**

[**DevOps**](#_o38visgqeufv) **7**

[**Developers (with full permissions)**](#_4vn6s8eypzbe) **8**

[**Developers (with reduced permissions)\***](#_m6ygcyerkwuk) **9**

[**Tester**](#_p82cmffw8tyt) **10**

[**Monitoring**](#_syceyrexg7tb) **10**

[**Operations**](#_gge9jdaismv) **11**

[**Audit Viewer**](#_nm7zwgi38lyg) **13**

[**Environment Setup**](#_pewibbuhjw0l) **13**

[**Teams Setup Proposal @ Customer (Minimum Setup)**](#_o3caxhc35uc7) **15**

[**Everyone at <Customer Name>**](#_anmizfinu1qb) **15**

[**<Customer Name>-Org-Admins**](#_nlcy6d7ihvfn) **16**

[**<Customer Name>-DevOps**](#_any0h1u4kozi) **16**

[**<Customer Name>-Devs**](#_iwm5t1is5tl7) **17**

[**<Customer Name>-Monitoring**](#_9cvum6th37fg) **18**

[**<Customer Name>-Ops**](#_b1tc2rl4hxq6) **19**

[**<Customer Name>-Testers**](#_qfvkeb7pjjou) **21**

[**<Customer Name>-Audit-Viewers**](#_81puy3afc85u) **21**

[**<Customer Name>-API-Consumers**](#_ibuybf4hheha) **21**

[**Best Practices**](#_pb1497gf0rxa) **21**

[**References**](#_z67n2l94sle6) **23**

# 

# Overview

In 2021, Anypoint Platform team introduced Teams feature to provide improved user and permissions management. Teams provides user grouping mechanisms in Anypoint Platform, which streamlines collaboration, reuse, and scalability.

This document defines the best practices around Teams setup in Anypoint Platform and contains a proposal for a simple Teams setup for <Customer Name>.

## Target Audience

The target audience for this document will be Organization Owners, Platform Owners and Administrators who need to decide on the team and permissions structure best suited to their requirements.

## 

# An Introduction to Teams

Anypoint Platform now provides Teams feature under Access Management enabling organization administrators to create a group of users based on their profile and then assign those members of the group with required permissions. Teams have a cascading permissions structure that helps you manage users and permissions with scalability in mind.

The Teams feature enables you to achieve following:

* Manage your organization from a simple, global view
* Group users into teams to easily manage permissions and share assets
* Assign team maintainers to delegate management of team members and child teams
* Manage user and team permissions across multiple business groups in one place

The Teams feature enables you to replicate the structure of your organization and seamlessly manage permissions and team membership as your organization grows. Outside of teams, you can still assign specific permissions to individual users if they require permissions that should not extend to an entire team. Adopting a team structure in your organization also enables you to share assets more easily. In Anypoint Exchange, you can share an asset with an entire team rather than having to select each user individually.

## Planning Teams

When you are planning your organization’s team structure, consider the following:

* Each member of your organization is a member of the “Everyone at <Root Organization> team, or the root team.
* Your team structure should reflect the structure of your organization so that you can plan your teams and configure permissions according to member needs. However, you can move teams as the needs of your organization change.
* Team names must be unique across your organization.
* Your organization can have up to 2000 teams.
* Your teams can be nested up to 10 levels, including the “Everyone at <Root Organization> team.
* You can create teams without adding members.
* Every child team inherits permissions from its parent team.

## Team Maintenance

When adding a user to a particular team, you can add the user as a member or maintainer. As a maintainer, the user could perform the following additional tasks:

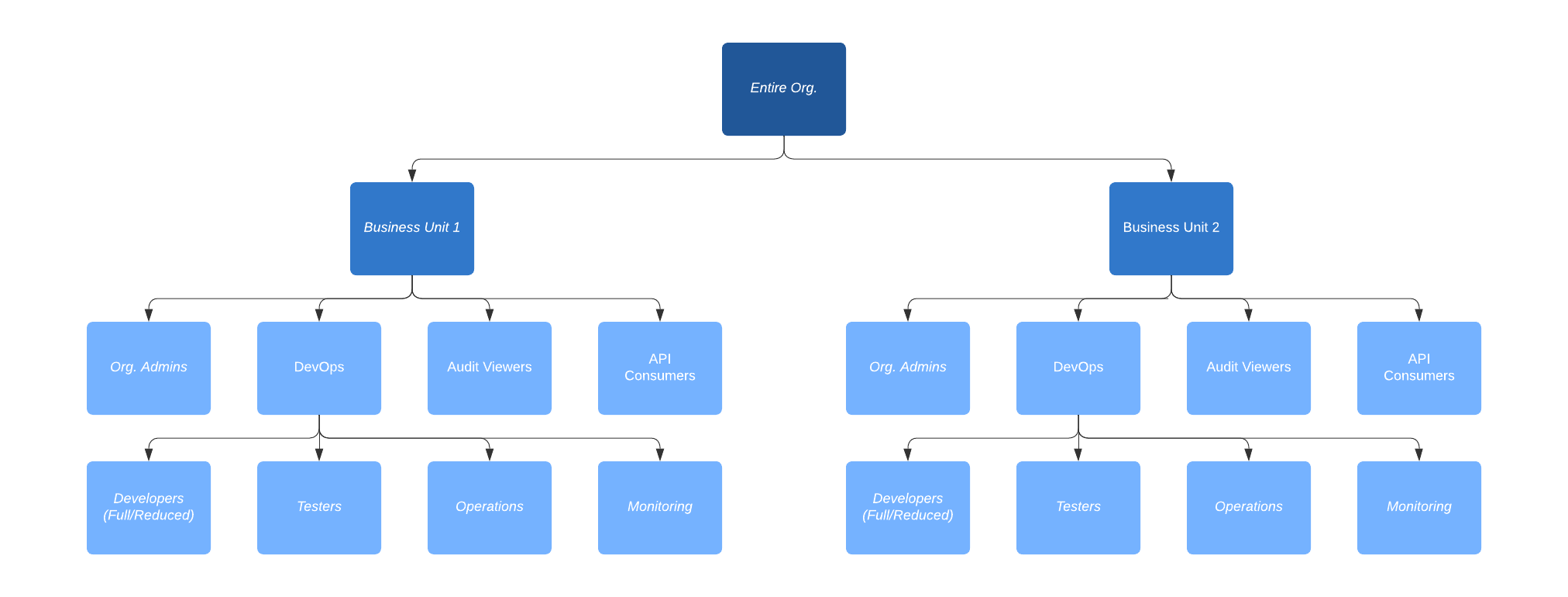
* Change the team name
* Add or remove members from the team
* Add or remove additional team maintainers
* Move the team under different parent teams (if they maintain the other parent team)
* Delete the team

In case SSO is implemented, then Teams in Anypoint PLatform can be mapped to the groups in IdP and then the modification to Team like adding or removing members and maintainers is carried out in IdP itself.

It is important to note that as Organization Administrator, you can maintain permissions and other things but you cannot maintain the team (add, remove users) without having the maintainer role for the team itself or for the parent team.

Please note that if you have the org admin permission assigned for the “Everyone at <Root Organization>” team, then it is possible to maintain permissions and members for all child teams. Therefore please avoid doing so or keep this to minimum to have better control and visibility.

## Recommended Teams Structure



When designing the structure, ‘Business Groups’ shouldn’t be solely used for the team's control. Rather Child Teams should be organized with more granularity and organization requirements.

|  |  |  |
| --- | --- | --- |
| Business Unit | | |
| Description | This is the top level and has minimum privileges | |
| Environment | Component | Permissions |
| Environment Agnostic | Exchange | * Exchange Viewer |

|  |  |  |
| --- | --- | --- |
| Organization Administrators | | |
| Environment | Component | Permissions |
| Environment Agnostic | Design Center | * Design Center Developer |
| Exchange | * Exchange Administrator |
| General | * Organization Administrator * Audit Log Viewer |
| API Governance | * Governance Administrator |
| Inheriting Permissions from | | * [Business Unit](#_f0eavcaecdjp) |

|  |  |  |
| --- | --- | --- |
| DevOps | | |
| Environment | Component | Permissions |
| Non-Production | API Manager | * View API Alerts * View APIs Configuration * View Contracts * View Policies |
| Runtime Manager | * Read Alerts * Read Applications * Read Servers * Read Runtime Fabrics * Manage Alerts * Manage Schedules |
| MQ | * View clients * View destinations |
| Secrets Manager | * Read secrets metadata |
| Environment Agnostic | Anypoint Monitoring | * Monitoring Viewer |
| API Manager | * Portals Viewer |
| Inheriting Permissions from | | * [Business Unit](#_f0eavcaecdjp) |

|  |  |  |
| --- | --- | --- |
| Developers (with full permissions) | | |
| Environment | Component | Permissions |
| Production | API Manager | * View API Alerts * View APIs Configuration * View Contracts * View Policies * Manage API Alerts |
| Runtime Manager | * Read Alerts * Read Applications * Read Servers * Read Runtime Fabrics * Manage Alerts * Manage Schedules |
| Secrets Manager | * Read secrets metadata |
| MQ | * View clients * View destinations |
| Non-Production | API Manager | * API Manager Environment Administrator * Deploy API Proxies * Manage API Alerts * Manage APIs Configuration * Manage Contracts * Manage Policies |
| Runtime Manager | * Delete Applications * Download Applications * Manage Alerts * Manage Application Data * Manage Queues * Manage Runtime Fabric * Manage Schedules * Manage Settings * Manage Tenants * Manage Servers * Manage Application Flows * Create Applications |
| MQ | * Clear destinations * Manage clients * Manage destinations |
| Secrets Manager | * Manage Secret Groups * Write Secrets |
| Environment Agnostic | Visualizer | * Visualizer Editor (for Master Org) |
| Design Center | * Design Center Developer |
| Runtime Manager | * CloudHub Network Viewer |
| General | * Audit Log Viewer |
| API Manager | * API Group Administrator * API Creator * API Versions Owner |
| Exchange | * Exchange Administrator |
| Inheriting Permissions from | | * [Business Unit](#_f0eavcaecdjp) * [DevOps](#_o38visgqeufv) |

|  |  |  |
| --- | --- | --- |
| Developers (with reduced permissions)\* | | |
| Environment | Component | Permissions |
| Production | API Manager | * View API Alerts * View APIs Configuration * View Contracts * View Policies |
| Runtime Manager | * Read Alerts * Read Applications * Read Servers * Read Runtime Fabrics * Manage Alerts * Manage Schedules |
| Secrets Manager | * Read secrets metadata |
| MQ | * View clients * View destinations |
| Non-Production | API Manager | * Manage API Alerts * Manage APIs Configuration * Manage Contracts * Manage Policies |
| Runtime Manager | * Download Applications * Manage Alerts * Manage Application Data * Manage Queues * Manage Schedules * Manage Settings * Manage Application Flows |
| MQ | * Clear destinations |
| Environment Agnostic | Visualizer | * Visualizer Editor (for Master Org) |
| Design Center | * Design Center Developer |
| Runtime Manager | * CloudHub Network Viewer |
| General | * Audit Log Viewer |
| API Manager | * API Group Administrator * API Creator * API Versions Owner |
| Exchange | * Exchange Administrator |
| Inheriting Permissions from | | * [Business Unit](#_f0eavcaecdjp) * [DevOps](#_o38visgqeufv) |

\* In the above team (Developers with reduced permissions), the operational actions like application deployments, queue management, platform management etc. for non-production environments are managed by Operations team using DevOps tools or as necessary.

|  |  |
| --- | --- |
| Tester | |
| Inheriting Permissions from | * [Business Unit](#_f0eavcaecdjp) * [DevOps](#_o38visgqeufv) |

|  |  |  |
| --- | --- | --- |
| Monitoring | | |
| Environment | Component | Permissions |
| Production | API Manager | * View API Alerts * View APIs Configuration * View Contracts * View Policies * Manage API Alerts |
| Runtime Manager | * Read Alerts * Read Applications * Read Servers * Read Runtime Fabrics * Manage Alerts * Manage Schedules |
| Secrets Manager | * Read secrets metadata |
| MQ | * View clients * View destinations |
| Environment Agnostic | General | * Audit Log Viewer |
| Runtime Manager | * Cloudhub Network Viewer |
| Inheriting Permissions from | | * [Business Unit](#_f0eavcaecdjp) * [DevOps](#_o38visgqeufv) |

\*The Monitoring Team can also be referred to as the L1 Support team depending on the customer's organization setup.

|  |  |  |
| --- | --- | --- |
| Operations | | |
| Environment | Component | Permissions |
| Production | API Manager | * API Manager Environment Administrator * Deploy API Proxies * Manage API Alerts * Manage APIs Configuration * Manage Contracts * Manage Policies * View API Alerts * View APIs Configuration * View Contracts * View Policies |
| Runtime Manager | * Delete Applications * Download Applications * Manage Alerts * Manage Application Data * Manage Queues * Manage Runtime Fabric * Manage Schedules * Manage Settings * Manage Tenants * Manage Servers * Manage Application Flows * Create Applications |
| MQ | * Clear destinations * Manage clients * Manage destinations |
| Secrets Manager | * Manage Secret Groups * Write Secrets |
| Non-Production | API Manager | * API Manager Environment Administrator * Deploy API Proxies * Manage API Alerts * Manage APIs Configuration * Manage Contracts * Manage Policies |
| Runtime Manager | * Delete Applications * Download Applications * Manage Alerts * Manage Application Data * Manage Queues * Manage Runtime Fabric * Manage Schedules * Manage Settings * Manage Tenants * Manage Servers * Manage Application Flows * Create Applications * Manage Runtime Fabrics |
| MQ | * Clear destinations * Manage clients * Manage destinations |
| Secrets Manager | * Manage Secret Groups * Write Secrets |
| Environment Agnostic | Visualizer | * Visualizer Editor |
| Anypoint Monitoring | * Anypoint Monitoring Administrator |
| General | * Audit Log Viewer |
| Exchange | * Exchange Contributor |
| Runtime Manager | * Cloudhub Network Administrator |
| API Manager | * API Group Administrator * API Creator * API Versions Owner |
| Inheriting Permissions from | | * [Business Unit](#_f0eavcaecdjp) * [DevOps](#_o38visgqeufv) |

|  |  |  |
| --- | --- | --- |
| Audit Viewer | | |
| Environment | Component | Permissions |
| Environment Agnostic | General | * Audit Log Viewer |
| Inheriting Permissions from | | * [Business Unit](#_f0eavcaecdjp) |

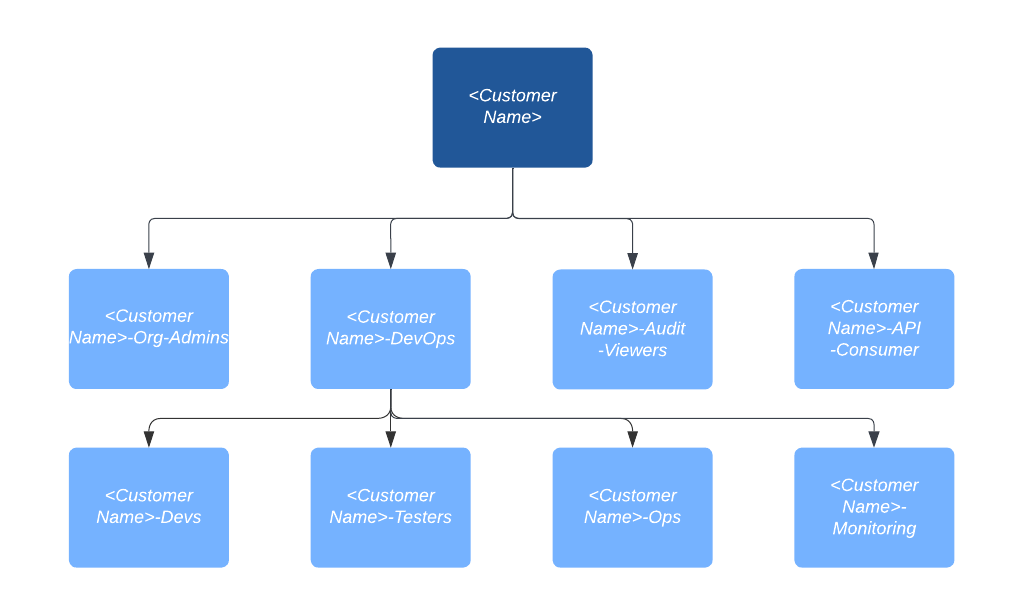
\*\*Depending on your organization setup and requirements, Architects can either be added to Developers or Monitoring teams in the above proposal.

## Environment Setup

In above recommendation, following environment setups have been envisioned:

* Production - This includes Production environment and additionally Pre-Production/UAT/QA environment if it is being treated equally to Production in terms of configuration and setup.
* Non-Production - This includes development, test and sandbox environments where usually developers perform their unit and integration testings.
* Environment Agnostic - There are many permissions which are not environment specific and therefore apply to the platform overall irrespective of the environment.

# Teams Setup Proposal @ Customer (Minimum Setup)



|  |  |  |
| --- | --- | --- |
| Everyone at <Customer Name> | | |
| Description | This is the top level and has minimum privileges | |
| Environment | Component | Permissions |
| Environment Agnostic | Exchange | * Exchange Viewer |

| <Customer Name>-Org-Admins | | |
| --- | --- | --- |
| Environment | Component | Permissions |
| Environment Agnostic | Design Center | * Design Center Developer |
| Exchange | * Exchange Administrator |
| General | * Organization Administrator * Audit Log Viewer |
| Inheriting Permissions from | | * [Everyone at <Customer Name>](#_anmizfinu1qb) |

|  |  |  |
| --- | --- | --- |
| <Customer Name>-DevOps | | |
| Environment | Component | Permissions |
| Non-Production | API Manager | * View API Alerts * View APIs Configuration * View Contracts * View Policies |
| Runtime Manager | * Read Alerts * Read Applications * Read Servers * Manage Alerts * Manage Schedules |
| MQ | * View clients * View destinations |
| Secrets Manager | * Read secrets metadata |
| Environment Agnostic | Anypoint Monitoring | * Anypoint Monitoring Viewer |
| API Manager | * Portals Viewer |
| Inheriting Permissions from | | * [Everyone at <Customer Name>](#_anmizfinu1qb) |

|  |  |  |
| --- | --- | --- |
| <Customer Name>-Devs | | |
| Environment | Component | Permissions |
| Production | API Manager | * View API Alerts * View APIs Configuration * View Contracts * View Policies * Manage API Alerts |
| Runtime Manager | * Read Alerts * Read Applications * Read Servers * Read Gateways * Manage Alerts * Manage Schedules |
| Secrets Manager | * Read secrets metadata |
| MQ | * View clients * View destinations |
| Non-Production | API Manager | * API Manager Environment Administrator * Deploy API Proxies * Manage API Alerts * Manage APIs Configuration * Manage Contracts * Manage Policies |
| Runtime Manager | * Delete Applications * Download Applications * Manage Alerts * Manage Application Data * Manage Queues * Manage Runtime Fabric * Manage Schedules * Manage Settings * Manage Tenants * Manage Servers * Manage Application Flows * Create Applications |
| MQ | * Clear destinations * Manage clients * Manage destinations |
| Secrets Manager | * Manage Secret Groups * Write Secrets |
| Environment Agnostic | Visualizer | * Visualizer Editor (for Master Org) |
| Design Center | * Design Center Developer |
| Runtime Manager | * CloudHub Network Viewer |
| General | * Audit Log Viewer |
| API Manager | * API Group Administrator * API Creator * API Versions Owner |
| Exchange | * Exchange Administrator |
| Inheriting Permissions from | | * [Everyone at <Customer Name>](#_anmizfinu1qb) * [<Customer Name>-DevOps](#_any0h1u4kozi) |

In case you decide to go for reduced Developer permissions, then please adapt according to this section [Developers (with reduced permissions)](#_m6ygcyerkwuk)

|  |  |  |
| --- | --- | --- |
| <Customer Name>-Monitoring | | |
| Environment | Component | Permissions |
| Production | API Manager | * View API Alerts * View APIs Configuration * View Contracts * View Policies * Manage API Alerts |
| Runtime Manager | * Read Alerts * Read Applications * Read Servers * Read Gateways * Manage Alerts * Manage Schedules |
| Secrets Manager | * Read secrets metadata |
| MQ | * View clients * View destinations |
| Environment Agnostic | General | * Audit Log Viewer |
| Runtime Manager | * Cloudhub Network Viewer |
| Inheriting Permissions from | | * [Everyone at <Customer Name>](#_anmizfinu1qb) * [<Customer Name>-DevOps](#_any0h1u4kozi) |

|  |  |  |
| --- | --- | --- |
| <Customer Name>-Ops | | |
| Environment | Component | Permissions |
| Production | API Manager | * API Manager Environment Administrator * Deploy API Proxies * Manage API Alerts * Manage APIs Configuration * Manage Contracts * Manage Policies * View API Alerts * View APIs Configuration * View Contracts * View Policies |
| Runtime Manager | * Delete Applications * Download Applications * Manage Alerts * Manage Application Data * Manage Queues * Manage Runtime Fabric * Manage Schedules * Manage Settings * Manage Tenants * Manage Servers * Manage Application Flows * Create Applications |
| MQ | * Clear destinations * Manage clients * Manage destinations |
| Secrets Manager | * Manage Secret Groups * Write Secrets |
| Non-Production | API Manager | * API Manager Environment Administrator * Deploy API Proxies * Manage API Alerts * Manage APIs Configuration * Manage Contracts * Manage Policies |
| Runtime Manager | * Delete Applications * Download Applications * Manage Alerts * Manage Application Data * Manage Queues * Manage Runtime Fabric * Manage Schedules * Manage Settings * Manage Tenants * Manage Servers * Manage Application Flows * Create Applications |
| MQ | * Clear destinations * Manage clients * Manage destinations |
| Secrets Manager | * Manage Secret Groups * Write Secrets |
| Environment Agnostic | Visualizer | * Visualizer Editor |
| Anypoint Monitoring | * Anypoint Monitoring Administrator |
| General | * Audit Log Viewer |
| Exchange | * Exchange Contributor |
| Runtime Manager | * Cloudhub Network Administrator * Manage Runtime Fabrics |
| API Manager | * API Group Administrator * API Creator * API Versions Owner |
| Inheriting Permissions from | | * [Everyone at <Customer Name>](#_anmizfinu1qb) * [<Customer Name>-DevOps](#_any0h1u4kozi) |

|  |  |
| --- | --- |
| <Customer Name>-Testers | |
| Inheriting Permissions from | * [Everyone at <Customer Name>](#_anmizfinu1qb) * [<Customer Name>-DevOps](#_any0h1u4kozi) |

|  |  |  |
| --- | --- | --- |
| <Customer Name>-Audit-Viewers | | |
| Environment | Component | Permissions |
| Environment Agnostic | General | * Audit Log Viewer |
| Inheriting Permissions from | | * [Everyone at <Customer Name>](#_anmizfinu1qb) |

|  |  |  |
| --- | --- | --- |
| <Customer Name>-API-Consumers | | |
| Environment | Component | Permissions |
| Environment Agnostic | API Manager | * Portals Viewer |
| Inheriting Permissions from | | * [Everyone at <Customer Name>](#_anmizfinu1qb) |

# Best Practices

1. The teams structure should reflect your organization structure but avoid deep running structures
2. At the parent level keep the permissions to minimum or if possible to nil as all the child teams inherit parent teams permissions
3. Child Teams with permissions should ideally reflect the role/profile of members underneath e.g developers, operations etc.
4. Be careful when assigning members to following teams:
   1. Organization Administrators: The members here have quite extensive rights and have ability to assign themselves and others additional permissions in the organization. Therefore have a maximum of 2 or 3 members in this team.
   2. Operations: The members of this team have full permissions in Production and Sandbox environments and therefore only members who really know the operational things and are responsible should be assigned to this team.
5. If the user does not fit to any of the defined teams, then assign minimum required permissions directly to such user. This should be done only in exceptional cases and in the longer term such users should be classified into an existing or a new team.
6. Make sure that you assign at least 1 maintainer other than the Organisation Owner who is then responsible for adding/removing users from teams. They cannot modify any permissions for the team itself until they have Org. Administrator permission as well. Additionally if SSO like Okta, Azure AD etc. is available in organization then it is best to set up the SSO and then map the Teams in Anypoint to Groups in your IdP. This way the permissions are maintained in Anypoint but the Team membership is controlled centrally from IdP.
7. Business Groups shouldn’t be solely used for team’s control.
8. In case you are using CI/CD tools for managing deployments, then please ensure to provide access to developers as per section ‘[Developers with reduced permissions](#_m6ygcyerkwuk)’

## 

# References

* [Manage User Access Using Teams | MuleSoft Documentation](https://docs.mulesoft.com/access-management/teams)
* [Migrate Roles to Teams | MuleSoft Documentation](https://docs.mulesoft.com/access-management/roles#migrate-roles-to-teams)
* [Permissions Available in Anypoint Platform | MuleSoft Documentation](https://docs.mulesoft.com/access-management/permissions-by-product)