Best practices for structuring organizations in your enterprise

In this article

Learn to identify how many organizations to create within your enterprise, and how you should structure them.

About best practices for organizations within an enterprise ⊘

There are a multiple options for structuring the organizations within your enterprise. Each approach has pros and cons, and the best structure for your enterprise depends on the characteristics and needs of your business, including size and security constraints.

However, we also recommend aligning your strategy with the culture you want to create, not the culture you currently have. If you want to advance in terms of collaboration and innersourcing, structure your tools accordingly. Then, your tools can assist you in the cultural change instead of acting as a blocker.

About organizational number ⊘

In general, GitHub recommends minimizing the number of organizations you create. Having fewer organizations encourages greater collaboration and innersourcing, which increases efficiency. In fact, many businesses are best served by a single organization, for the following reasons.

- It's easier to find resources within a single organization, as there's only one place to search.
- It's easier to communicate within a single organization, as @-mentions only work between members of the same organization.
- Being part of a single, large organization where anyone and anything is accessible fosters collaboration and loyalty, whereas being separated into smaller organizations can make teams more isolated.

Organization owners always have access to all repositories owned by the organization. If your company is large enough that no single owner should have access to all repositories, consider creating multiple organizations.

The main benefit of creating multiple organizations is the ability to configure separate policies, settings, and requirements for each. For example, each organization can have a different SAML configuration.

Avoid creating a one-to-one relationship between organizations and structural entities of

your company, such as individual teams or business units. Instead, group structural entities that can share policies, settings, and requirements into a single organization. This approach maximizes collaboration while meeting your regulatory requirements.

It's always easier to add organizations than to remove them, so we recommend starting with a small number of organizations, which gives you more flexibility in the future. After you develop more experience of what works well for your business, you can create additional organizations if the need arises.

Removing organizations is much more difficult, often requiring migrations and a reduction in flexibility that teams have gotten used to. Many customers have come to regret creating a large number of organizations after they experience the challenging and time-consuming process of reducing their number.

We recommend creating and enforcing fixed and transparent rules for creating a new organization in your enterprise. This will make it easier for everyone to understand the purpose of each organization and which assets are located where.

About organizational structure 🔗

There are five main archetypes for organizational structure. The archetypes are defined by two decisions:

- Whether to use a single organization or multiple organizations
- Whether to grant all members access to all repositories, or use teams to manage repository access more granularly

For more information about teams, see "[About teams](#)."

Single organization with direct repository access 🔗

The simplest organizational structure is a single organization, where members are granted access to all repositories directly via organization membership. Teams may be used for coordination and communication, but not for managing repository access.

This structure works best for small companies, such as startups, where everyone collaborates on everything. It can work for medium-sized companies as well, if trust is high.

To use this archetype, set the base permissions for the organization to "Write" or "Read." For more information, see "[Setting base permissions for an organization](#)."

Single organization with teams for repository access 🔗

If your company needs to control repository access more granularly, you can set the base permissions for your organization to "None," then give each team access to specific repositories only.

This structure works best for medium-sized companies, or for small companies with lower trust. For smaller companies with high trust, where everyone collaborates on everything, administering teams may not be worth the time investment.

Multiple organizations with direct repository access 🔗

For larger companies, managing repository access within a single organization can become unwieldy, even with teams. This archetype utilizes multiple organizations to manage repository access instead. Each organization's members have access to all of that organization's repositories.

This structure works best for companies that are large enough to have different groups

that don't need to work together. This structure is not as useful if collaboration across business units is important.

To use this archetype, create one organization for each group that can share policies, settings, and requirements as described above, then set the base permissions for each organization to "Write" or "Read."

Multiple organizations with teams for repository access 🔗

Very large companies may require more granular control over repository access, even within multiple organizations. In this case, you can use teams to give each group access to specific repositories only.

To use this archetype, create one organization for each group that can share policies, settings, and requirements as described above, set the base permission for each organization to "None," then give each team access to specific repositories only.

Multiple organizations with different access methods 🔗

If you want the collaboration benefits of a single organization with direct repository access, but you have a small number of repositories that are too sensitive for global access, consider using multiple organizations with a mix of access methods.

To use this archetype, create one organization for all of your employees and most of your repositories. Grant all members access to all repositories in this organization by setting the base permissions for the organization to "Write" or "Read."

Then, create a second organization specifically for more sensitive repositories. In this organization, set the base permissions to "None", add only the people that need to access the sensitive repositories, and manage access to the repositories via team membership.

Further reading 🔗

- [Organize your experts with ad hoc teams](#) in the GitHub blog
- "[Best practices for organizations](#)"