Prodly AppOps Security Model Guidelines

Table of Contents

Overview

This document describes the security model design and configuration Prodly recommends you implement for your Prodly AppOps control (installation) organization. Putting in place an optimal visibility and access sharing security model ensures controlled data deployment and integrity, while maintaining maximum productivity.

Because users with sufficient permissions to access an AppOps connection record in a Salesforce instance can deploy data to that organization (resulting in a potentially irreversible data operation, including to a production organization), a proper security model is critical to maintaining data integrity.

If the team members who use Prodly AppOps have varying roles and responsibilities, especially those working with several projects or customers, you greatly benefit by putting a security model in place. Having a security model becomes especially critical for system integrators. The AppOps centrally-controlled deployment architecture (where connections to all customer organizations exist in one control organization) gives users the potential to deploy the wrong data set or use the wrong connection.

Security Models

Salesforce provides complete flexibility to implement a sharing model based on your particular organizational requirements. Create and combine the available profiles, permission sets, roles, groups, and sharing rules to fit your group requirements. The following sections describe common security models that provide sufficient security and access for the corresponding use case.

Internal Company Use Implementations

Prodly recommends the following security model for a medium to enterprise company doing internal deployments with developer sandbox relational test data:

- Establish a select set of Salesforce system administrators who install, configure, and manage the Prodly AppOps Salesforce app.
- Establish a select set of Salesforce system administrators who create and deploy data sets to their developer sandboxes.
- Establish a wider team of standard users who deploy data sets to their developer sandboxes.

Configure the following visibility settings on the relevant user profiles or permission sets:

- Administrators establish and own connections to developer sandboxes.
- Administrators grant standard users read-only permissions to the connections, and deny access to (hide) the **Connections** tab.
- Administrators share connection records (with read-only access based on the connection username) with the appropriate role for each developer team.

- Administrators create data sets, which standard users use to deploy data to their respective developer organizations.
- Administrators create sharing rules and share relevant data sets with standard users based on those rules, for example, by using a role hierarchy or group so all team members involved on a particular customer implementation project can all view, edit, and deploy the relevant data sets as needed.
- Standard users deploy data sets, and monitor and analyze the results.

Enterprise System Integrator Implementations

Prodly recommends the following security model for enterprise system integrators with multiple ongoing implementation projects that deploy relational reference data between a number of customer organizations from a centralized AppOps control organization:

- Establish a select set of Salesforce system administrators who install, configure, and manage the Prodly AppOps Salesforce app.
- Establish a wider team of standard users who create and deploy data sets to customer organizations.

Configure the following visibility settings on the relevant user profile or permission set.

- Administrators establish and own connections to all customer organizations for all projects from the centralized control organization owned by the system integrator.
- Administrators grant standard users read-only permissions to the connections, and deny access to (hide) the **Connections** tab.
- Administrators share connection records with read-only access based on the connection instance URL with the appropriate role or group for each implementation team, so all team members involved with a particular customer implementation project can deploy to, deploy from, or use the schema of the connected organization.
- Standard users create data sets and have access only to the data sets they create and own for their customer projects.
- Administrators create sharing rules and share relevant data sets with standard users based on those rules, for example, by using a role hierarchy or group so all team members involved on a particular customer implementation project can all view, edit, and deploy the relevant data sets as needed.
- Administrators create data sets from data set templates for standard implementation projects.
- Standard users deploy data sets, and monitor and analyze the results.

Permission Sets

During AppOps installation and configuration, provide access to the AppOps app only to Salesforce system administrators. Then assign the appropriate role-based Salesforce permission sets included with the AppOps app as needed. This approach provides the common minimum level of visibility to AppOps

functionality each user role requires and ensures that Salesforce sharing settings can provide access only to the necessary records.

Prodly AppOps Admin Permission Set

Assign this permission set to the select set of Salesforce system administrators who install, configure, and manage the Prodly AppOps Salesforce app across the entire team. This permission set provides access to all AppOps components, with view-all and modify-all permissions for all AppOps objects and both read and edit permissions for all custom fields.

To access AppOps custom settings, users need access to the Salesforce **Setup** app, which you can grant through their profile or another permission set.

The following list contains typical actions admins perform:

- Establish new connections and manage existing connections
- Create and manage folders
- Create and manage data sets and deployment plans
- Perform deployments
- Monitor, analyze, and manage deployment results
- Configure AppOps custom settings
- Configure and perform AppOps-related reporting
- Create deployment templates
- Generate data sets and deployment plans from deployment templates
- Import data sets and deployment plans from deployment template files
- Analyze AppOps deployment result reports and dashboards
- All access and actions listed in the following section

Prodly AppOps User Permission Set

Assign this permission set to standard AppOps users who can create their own data sets and deploy to the connected organizations that they have access to through sharing rules. This permission set provides the following access:

- Read access to the Folder object and all fields on the Folder object
- Access to the **Folders** tab.
- Read access to the Connection object and to all fields on the Connection object
- All object access to the Data Set Field object (a master-child object with access controlled by its parent)
- Read and edit field-level access to all fields on the Data Set Field object
- Access to the **Data Sets** and **Deployment Plans** tabs
- Read, create, edit, and delete access to the Data Set object
- Read and edit field-level access to all fields on the Data Set Field object
- Access to the **Deployment Results** tab
- Read access to the Deployment Results object

- Read field-level access to all fields in the Deployment Results object
- Read and view-all access to the Deployment Batch Results object (a master-child object with access controlled by its parent)
- Read-only field-level access to all fields on the Deployment Batch Results object
- Read and view-all object access to the Deployment Data Set Results object (a master-child object with access controlled by its parent)
- Read-only field-level access to all fields on the Deployment Data Set Results object
- Read and view-all access to the Deployment Record Results object (a master-child object with access controlled by its parent)
- Read-only field-level access to all fields on the Deployment Record Results object
- Access to the **Deployment Center** tab

The following list contains typical actions standard users perform:

- Import data sets and deployment plans
- Deploy data sets and deployment plans
- Monitor and analyze deployment results

User Roles

One way to segment users in a hierarchical structure is through Salesforce roles. Define roles that match your company structure and restrict access to AppOps records based on the role and all roles below in the hierarchy.

For example, define a top-level system integrator CPQ role, with a child Western and Eastern territory role, and subsequent state specific roles for California and Oregon in Western and New York and Florida in Eastern. With this structure, you can segment connections and data sets by geographic location. Users at the Western level can see and deploy for any customers in California and Oregon, while users at the California level can only deploy to customers in that state.

Public Groups

Another way to segment users in a list structure is through Salesforce public groups. Define a group whose members have access to a common set of connections and data sets.

For example, define an Acme Corp SI group that contain members who are all involved with the Acme Corp implementations and need access to all connections and data sets for that project.

Sharing Settings

Salesforce sharing settings allow you to expand access to records based on the following criteria:

- Record owner role

- Record owner role subordinates hierarchy
- Record owner public groups
- Custom field criteria

Records owners can share records with other users. Access can be read-only or read/write.

For the custom field criteria, you can define custom fields on the AppOps managed connection and data set objects, and set those values as needed to identify records which can then drive the sharing rules.

With the default standard Prodly AppOps User permission set, users have read access to all folder records and private read and write access to connection and data set records that they own. Sharing rules can expand that access to other records, so users see only data sets that apply to their projects and can deploy to connected organizations only for their customers.

AppOps Object Access

In your AppOps security model, determine and implement access restrictions for the following object types: fields, pages, tabs, and components. Each type provides the appropriate level of access for each area and stage of AppOps application usage.

Note: System administrators and all profiles or permission sets with view-all-data access override sharing rules and can see all records.

Connections

A user's access to connection records determines the Salesforce organizations the user can select as source and destination when deploying data sets and deployment plans.

Determining and controlling correct access to connections is essential. When deploying as an administrator user who established the connections, a user can deploy potentially-irreversible data changes on a Salesforce organization.

Carefully consider and implement a sharing model which works for your company.

In general, Prodly recommends the following security model suggestions for connections:

- Have the system administrator establish and retain ownership of all connections. Share the connections with team members on a per project or role basis. This approach requires that the admin user has login access to all of the Salesforce organizations.
- Have a point person for each project or role establish their own connection and own those connection records. Have the administrator retain visibility and management control over all connections. Share connections with team members as needed. This approach does not require the administrator to have access to every Salesforce organization.

- Select **Do Not Allow As Destination** on connections of organizations that should never be the destination of a deployment.

Once the connection records have been established and ownership set, you can set sharing rules on the connection records to give standard users read-only access to connections for deployment. When there are multiple administrators or team leads who own connections and want to share them with other team members, sharing is determined by the connection record owner. When only a core set of administrators own connections, then base sharing on an attribute of a connection record, for example, the instance URL or the username, which can identify it as being part of a project or group.

Define and set custom fields for more flexibility. For example, add an Account lookup field to the connection object and set the field on the connection record to associate the connection with the corresponding customer account. Then, base your sharing rules on those accounts, so only the team working on those customer organizations have access to the connections.

Do not edit connections fields, even as administrator. Only administrators should edit the **Active** and **Connection Name** fields, or transfer the ownership.

Hide the **Connections** tab from standard users.

Use create-read-update-delete (CRUD) object-level security and field-level security (FLS) to control access to the following objects:

- Connection

And the following tabs:

- Connections

Data Sets

Data sets identify the portion of your data model AppOps copies from Salesforce source organization to destination organization during deployment. Therefore, controlling access to data sets controls data deployment.

Note: *You can also control access to the actual data records through profiles and permission sets.*
Determine the group of users who can create data sets. Revoke or add permissions as necessary. Avoid view-all and modify-all permissions to prevent unwanted data deployments and data set editing. Then through sharing rules, widen access to data sets for team members so they can edit or deploy data sets created and owned by other users. If one administrator or small set of team leads owns all of the data sets, these owners can share the data sets with other team members based on project, customer, or other relevant criteria. Alternatively, control data set sharing based on some attribute, such as the Data Set object namespace.

If standard users should not view or edit data sets, hide the **Data Sets** tab from them.

Use CRUD object-level security and FLS to control access to the following objects:

- Data Set
- Relationship
- Data Set Field

And the following tabs:

- Data Sets

And the following Lightning components:

- Deploy Data Set

Note: You must share all data set elements in a data set chain, not just the root element, to allow AppOps access to all the data records.

Deployment Plans

Deployment plans are groupings of data sets and sequence information that determines the order in which to deploy them. Therefore, controlling access to deployment plans controls sequenced data deployment.

Determine the group of users who can create deployment plans. Revoke or add permissions as necessary. Avoid view-all and modify-all permissions to prevent unwanted data deployments and deployment plan editing. Then through sharing rules, widen access to deployment plans for team members so they can edit or deploy deployment plans created and owned by other users. If one administrator or small set of team leads owns all of the deployment plans, these owners can share the deployment plans with other team members based on project, customer, or other relevant criteria. Alternatively, control deployment plans sharing based on some attribute, such as the Deployment Plan object namespace.

If standard users should not view or edit deployment plans, hide the **Deployment Plans** tab from them.

Use CRUD object-level security and FLS to control access to the following objects:

- Deployment Plan
- Deployment Plan Step
- Deployment Plan Item

And the following tabs:

- Deployment Plans

And the following Lightning components:

- Deploy Deployment Plan

Note: You must share all data set elements in a data set chain, not just the root element, to allow AppOps access to all the data records.

Folders

Folders help organize data sets and connections to keep work for different projects or clients separate and to keep your work separate from other people's work in a single control organization.

Determine the group of users who can create folders. Revoke or add permissions as necessary. Avoid view-all and modify-all permissions to minimize unwanted access to other people's work. Then through sharing rules, widen access to folders for team members so they can access connections and edit or deploy data sets created and owned by other users. If one administrator or small set of team leads owns all of the folders, these owners can share the folders with other team members based on project, customer, or other relevant criteria. Alternatively, control folder sharing based on some attribute, such as the Folder object namespace.

If standard users should not view or edit folders, hide the **Folders** tab from them.

Note: Hiding folders does not automatically restrict access to the data sets and connections stored in the folders. You must restrict access to the data set and connection objects independently.

Use CRUD object-level security and FLS to control access to the following objects:

- Folder

And the following tabs:

- Folders

Templates

Restrict access to data set and deployment plan templates to a small set of administrator users. Standard AppOps users should not normally be creating, exporting, or importing templates. For all standard users, remove access to all objects related to templates and do not assign any sharing rules, unless a specific company use case warrants:

Use CRUD object-level security and FLS to control access to the following objects:

- Template
- Template Entry

And the following tabs:

- Templates

And the following Lightning components:

- Import Data Set

- Import Deployment Plan

Deployment and Results

Restrict all forms of deployment, including the **Deploy** button, **Deployment Center** tab, deployment Lightning Experience component, and AppOps API.

Having administrators control all deployments provides tighter control and security, but slows down the process. On the other hand, allowing a larger group of users to perform deployments increases productivity, but exposes more security risk.

Determine which users can track and analyze deployment results. This group is likely identical to the group who can deploy data sets. In general, Prodly recommends the following security models for deployment results:

- System administrator perform all deployments, tracking, and analysis.
- Standard users and/or team leads perform their deployments, tracking, and analysis.

Deployment result records are created by the connection user, not by the user currently logged in who initiates the deployment. Access to specific deployment results needs to be either global or shared appropriately. Prodly recommends setting view-all permissions for all users, because results records contain no sensitive information and initiate no actions.

Use CRUD object-level security and FLS to control access to the following objects:

- Deployment Result
- Deployment Batch Result
- Deployment Plan Result
- Deployment Plan Step Result
- Deployment Data Set Result
- Deployment Record Result

And the following tabs:

- Deployment Results
- Deployment Center

And the following Lightning components:

- Deploy Data Set
- Deploy Deployment Plan

And the following API services:

- Apex service API
- REST service API

Data Records

Use a company–specific profile and sharing model to separately secure data records that AppOps deploys as part of data set deployment. The records AppOps can access in both the source and destination organizations (for example, Account and child Contact records) are determined by the permissions of the user who established the connection. That is, the user who initially entered login credentials during the OAuth process into the Salesforce organization.

To avoid permission issues during deployment, Prodly recommends establishing all connections using system administrator level profiles with access to all records.