Whitepaper

Getting Started with OIG

A Guided Walkthrough

So, you've received the Okta Identity Governance package at your doorstep, opened the box up and found a bunch of parts, instructions in a foreign language and no Allen key. How do you get started?

This guide is designed to provide a means to get started – a look at the four main capabilities in Okta Identity Governance (OIG): Access Requests, Access Certifications, Entitlement Management, and Reporting. It provides a guided walkthrough in your own environment to give you a start in each of the capabilities. It does not cover every feature/function of the product and the list of new features/functions keeps growing.

The guide starts with some introductory information about the product and how to use the guide. Then there is a section to confirm the setup of your environment prior to starting the guided labs. The next three sections explore the core capabilities in OIG. The last section looks at the new Entitlements Management capability and how it leverages the LCM integrations, access requests and access certifications.

This is v3 of the Guide (Dec 23), which includes both the core GA capabilities of the product and the new Entitlement Management capability.

Table of Contents

Introduction

This section provides a brief introduction to the product and components.

What is Okta Identity Governance?

Okta Identity Governance (OIG) is Okta's Identity Governance and Administration (IGA) solution. The following figure shows the three product parts.

Okta Lifecycle Management (LCM) has been in the Okta platform for a few years and covers the end-to-end identity lifecycle. It includes integration with identity stores (like HR systems) often called HR as a Source (HRaaS), access entitlement policy and automation, and the provisioning/deprovisioning integration (such as the OIN integrations).

Okta Workflows iis the no-code automation platform in Okta. From an IGA perspective, Okta Workflows can address many of the non-trivial IGA use cases.

Okta Access Governance is the new product that introduces access requests, access certification, entitlement management, and governance reporting.

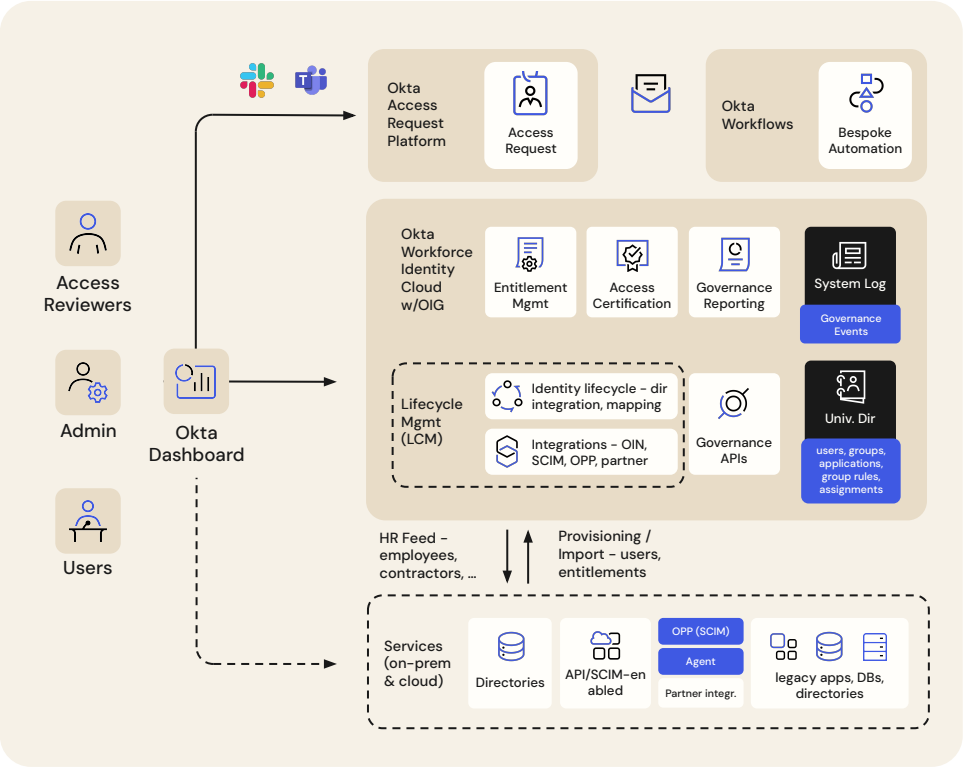
OIG relies on data stored in the **Okta Universal Directory** such as users, groups and applications. It is independent of, and does not need, single-sign on (SSO), multi-factor authentication (MFA) and other Okta products.

If you are new to Identity Governance, there's a good primer on the Okta website: https://www.okta.com/au/identity-101/identity-governance/.

The product page is https://www.okta.com/products/identity-governance/.

Architectural Overview of OIG

The following figure shows the major components and interactions with Okta Identity Governance.



The OIG product provides capabilities presented across three linked SaaS tenants: your **Okta Workforce Identity Cloud** org, an **Access Requests** tenant, and an **Okta Workflows** tenant. All three of these reside behind the **Okta Dashboard** (or Admin Console for Workflows) which provides a common interface into the Okta components and the other apps that users SSO to.

The Okta Workforce Identity Cloud (Okta) org has the **Lifecycle Management (LCM)**, **Universal Directory (UD)** and **Access Governance** products. The key components from an OIG perspective are:

- **Universal Directory** - holding the users, groups, applications, rules, assignments and other data objects
- **Lifecycle Management** - with the identity lifecycle processing, such as directory integration and profile mapping, and the integrations, such as the Okta Integration Network (OIN) connectors, SCIM integration, On-Premise Provisioning (OPP) and partner integrations.

- **Governance APIs** - a library of APIs to integrate with the governance objects, some of which are built into the Workflows connector.
- **Entitlement Management** - managing fine-grained entitlements for applications
- **Access Certification** - providing a means to build and run access certification campaigns
- **Governance Reporting** - pre-built reports for governance using the platform reporting interface
- **System Log** - governance events are written to the Okta System Log

The LCM integrations are integrating with the target systems, both on-premise and cloud. Where apps have APIs for provisioning or exposing SCIM endpoints, the integrations can talk directly. For legacy apps and systems, the on-premise provisioning (OPP) agent can be used. There are also partner-developed integrations that may use deployed components.

The Access Requests tenant is providing the front-end (or user interactive) request flows (called Request Types) for access requests with approvals. The Okta Workflows tenant is providing the backend (or automation) workflows (such as bespoke provisioning or campaign remediation flows).

There are multiple interfaces that are involved in the IGA use-cases – Slack/Teams for messaging interface to Access Requests or the Access Requests web UI, the Okta administration console for managing Okta, and the Okta Workflows UI for managing flows. All of these can be accessed via SSO from the Okta Dashboard. Also, email is used by different components for asynchronous notifications.

Using this Guide

This guide is written like a lab guide, a 'follow the bouncing ball' set of instructions to walk you through configuring and using the capabilities. \

Conventions Used

The following conventions are used:

- Where there is something to select (like a menu item) it will be highlighted in **bold**
- Where there is a link to click it will be highlighted in **bold-underscore**
- Where there's a search argument or a selection of some user-entered data it will be ***bold-italic***
- Where there is some text to be typed or an example script it will be in `monospace`

Where there are steps to be followed, there will be a checkbox beside. For example:

1. Click on **this**
2. Enter the following **string**

Where there is some informational text, it will be highlighted in a box like the following.

When using this, you should ensure you are facing the sun whilst standing on one leg.

Other Notes on Use of this Guide

You should be able to walk through the guide. There are five sections, the first covering some basic Okta setup to be able to leverage the OIG functions, and the sections on each of the OIG capabilities (Access Requests, Access Certification and Reporting) and a new section on Entitlement Management. Each can be done independently, however one of the reports will look at Access Certification history.

As this document has developed over time, some of the screenshots used in the lab sections may be slightly different to the ones you will see today.

The steps and screenshots in the lab sections use test data for users, groups, applications etc. in the authors system. It is expected that you will have your own test data for the labs, and where there are specific needs (such as having managerIDs defined) they are highlighted in the text.

You may see references to workflows in the text. This means the mechanism to build flows in Access Requests and should not be confused with Okta Workflows. Throughout this guide the term “request flow(s)” or just “flow(s)” is used to mean Request Types.

The OIG product documentation can be found at: https://help.okta.com/en-us/Content/Topics/identity-governance/iga.htm.

Check OIG is Enabled and Set Up Your Okta Identity Cloud

Prior to working through the sections looking at the capabilities, you will need to confirm the OIG capabilities have been enabled for your Okta Identity Cloud (Okta) org and then check or configure your Okta for the OIG capabilities.

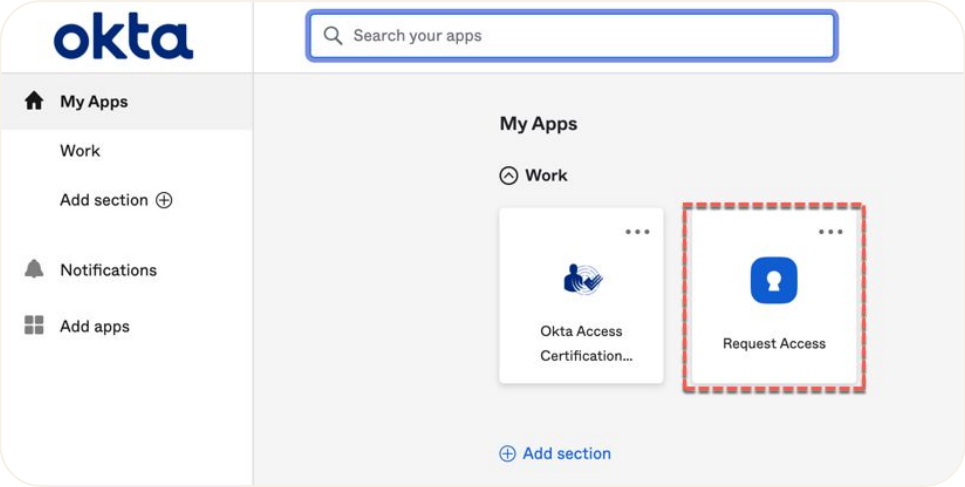
The following sections assume familiarity with the Okta user interfaces (Dashboard and Admin Console) and managing Okta objects and policies.

Is OIG Enabled for Your Okta Org?

The first thing to check is that all three capabilities are enabled for your Okta tenant. This is particularly relevant if you're running a trial as there are multiple features that must be enabled against the org.

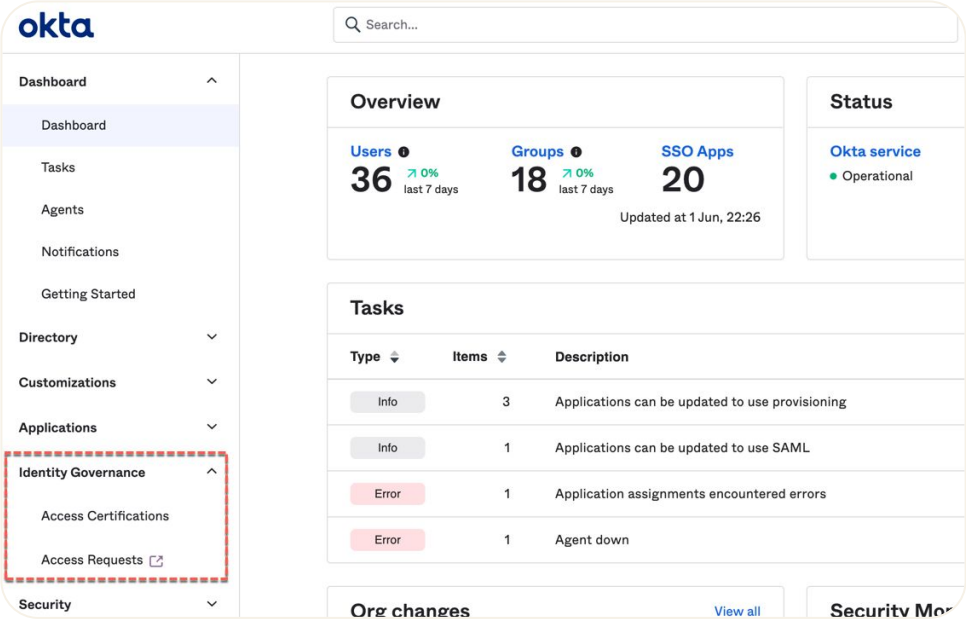
To check, use the following steps:

1. Log into your **Okta Dashboard** as your Admin user
2. You should see the **Access Requests** tile on your dashboard

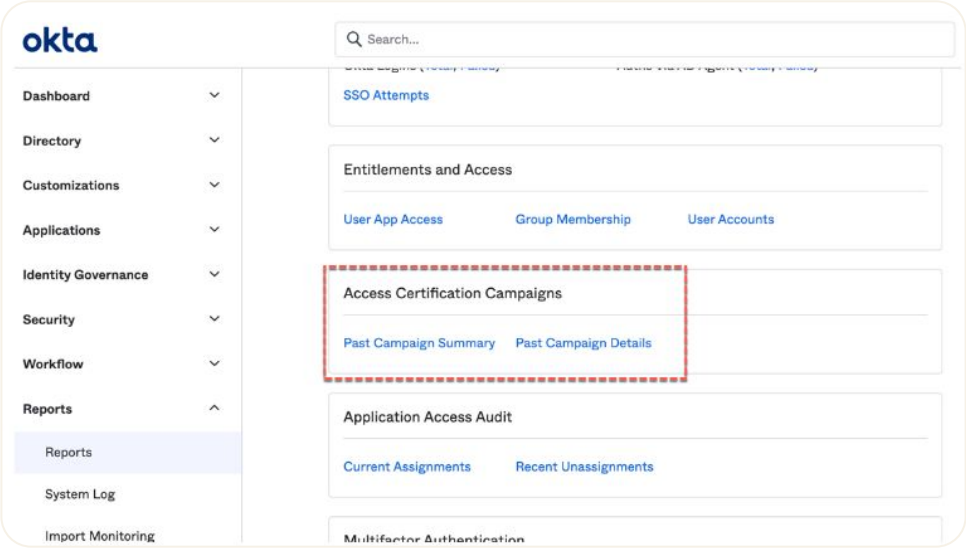


3. Go to the **Admin** console

4. Check that you see an **Identity Governance** menu item and when you expand it, both the **Access Certifications** and **Access Requests** are shown



5. Expand the **Reports > Reports** menu item
6. You should see multiple identity reports, including two certification reports



If all the above are present, you are ok to proceed with this guide. If not, speak to your Okta contact and ask them to check that the **Access Request** and **Access Certification** SKUs are enabled.

Objects in Universal Directory

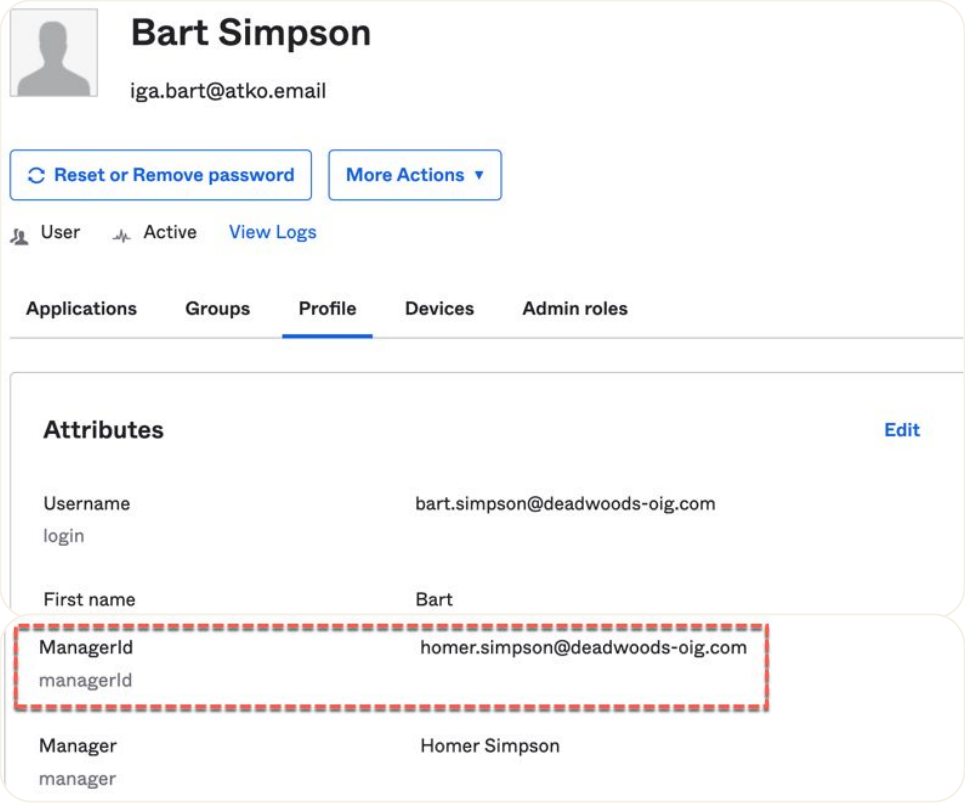
To be able to run campaigns, request access and run reports, you will need some data in Okta. You will need users, groups, and applications defined. You do not need AD integration or HR integration, as long as there are users in Okta.

As the IGA functions are independent of the provisioning process, you technically don't need to have provisioning enabled for the apps in Okta.

Users and User Profiles

You will need users defined in Okta, with Dashboard access. The access request and access certification sections later assume the employee-manager relationship is established, so some of your users should have a valid **user.profile.managerId** value set.

I would recommend at least four users in Okta (in addition to the admin) – two ordinary users with a manager set in the managerId field, one manager and one other for reviewing access (like a security admin role).



The out-of-the-box integration between Okta and Access Requests assumes you have that managerId field populated and that the manager userid == their email address.

If you have managers with their userid different to their email address, the Manager Approval flow in this document won't work, and you may need to put the manager ID value (not userid) into the managerId field in the user profiles or modify the profile mapping (not covered here).

For a production deployment you should think about how you would maintain any relationships you want to use for access approval or certification review. The standard user profile includes this field and some of the OIN apps will maintain it. But you may need to build a process to monitor and handle any other custom attributes you want to use for other relationships. Okta Workflows are good for these types of automated processes.

Groups and Group Rules

Groups can be used in both access requests and access certifications, so you should have some groups defined if you want to use them in either. It would be useful to have some of the groups connected to applications to show application assignment and removal via groups.

You should consider how your groups are managed in Okta. If you have groups automatically assigned via group rules, it doesn't make sense to also assign them via access requests and revoking access to groups also doesn't make sense (although the certification mechanism will handle that). Similarly for AD-managed or app-managed groups – you can see them but cannot be added to them or removed from them via Okta.

Applications

As with groups, you can use application memberships in access requests and access certifications, so you should have some defined. You do not need to have provisioning enabled for the applications as both the access requests and access certification functions are working on the application membership in Okta Universal Directory and provisioning is a result of those changes.

In production you would have provisioning enabled for applications for the full end-to-end lifecycle. You may also have Okta Workflows tied to application or group membership for bespoke provisioning flows.

Other Setup

If you are going to use Slack or Teams, you will need to have them configured in your environment with users tied to Okta users. The Access Requests section below will include configuring Slack.

Exploring OIG Access Requests

The first capability we will explore is Access Requests. It's the most complex of the three to configure, but once configured it can be used to grant access to be subsequently reviewed/revoked in Access Certifications.

This section of the document will explore the unconfigured Access Requests, walk through standard configuration (Okta and Slack integration), then create and run a simple approval flow.

The documentation to support this can be found at: https://help.okta.com/en-us/Content/Topics/identity-governance/access-requests/ar-overview.htm.

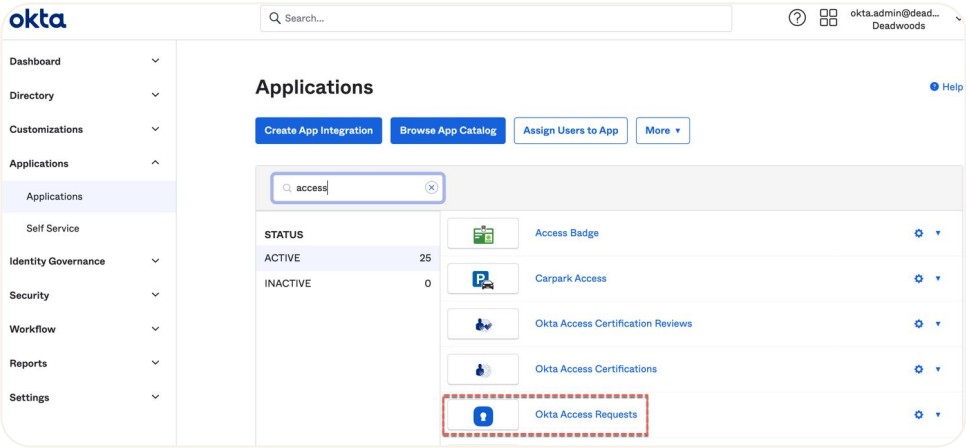
Check the Unconfigured Access Requests Component

There are two parts of the component to check – the Okta application configuration and the Access Requests module (tenant) itself.

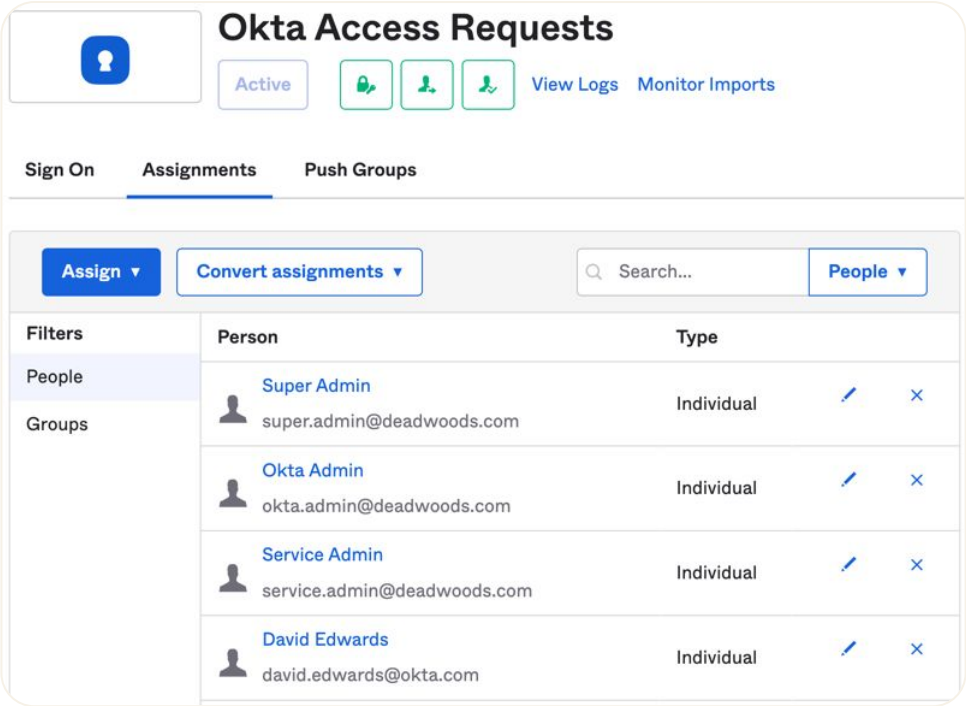
How is the Access Requests App Defined in Okta?

To check how Access Requests in defined in Okta:

1. If not already there, log into the **Okta Dashboard** as an administrator
2. Open the **Admin** console
3. Go to **Applications > Applications** and search for **Access** to see the **Okta Access Requests** application



4. Open the application and look at the **Assignments**



You will see all the Okta Super admins have been assigned to the application (individually), but no groups have been assigned. We will add a group to the app in the next section,

In this respect Access Requests is a standard application where groups of users can be assigned to allow/deny access to the function. You could just assign the “Everyone” group so all Okta users can use Access Requests.

You should see that most of the "plumbing" for the integration (SSO and provisioning) is hidden. This is by design (in the same way that Okta Workflows is integrated with Okta).

5. Go to the **Push Groups** tab
6. It should be empty

> We use the Push Groups function to define what user-groups are sent to Access Requests. If you want to scope a function in Access Requests (more on this later) you can use Okta groups, but they must be pushed to Access Requests.

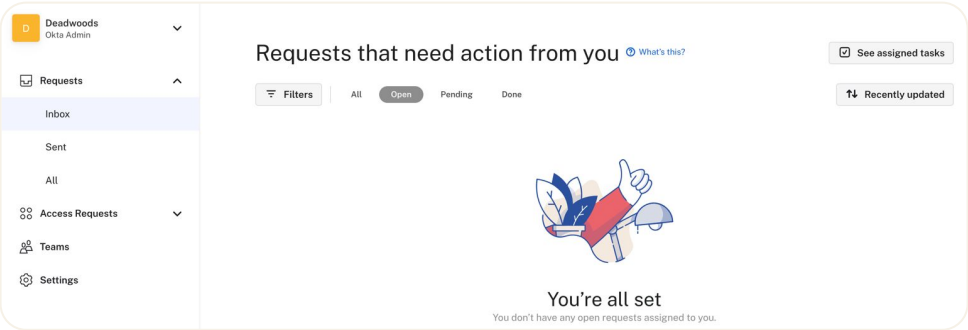
We will add Push Groups in the next section.

> As per every application in Okta, if a user is assigned to the application (individually or via a group) they will see the application tile on their Okta Dashboard. Admins can get to Access Requests from their Dashboard or the Identity Governance > Access Requests menu item in the Admin console.

What Does Access Requests Look Like OOTB

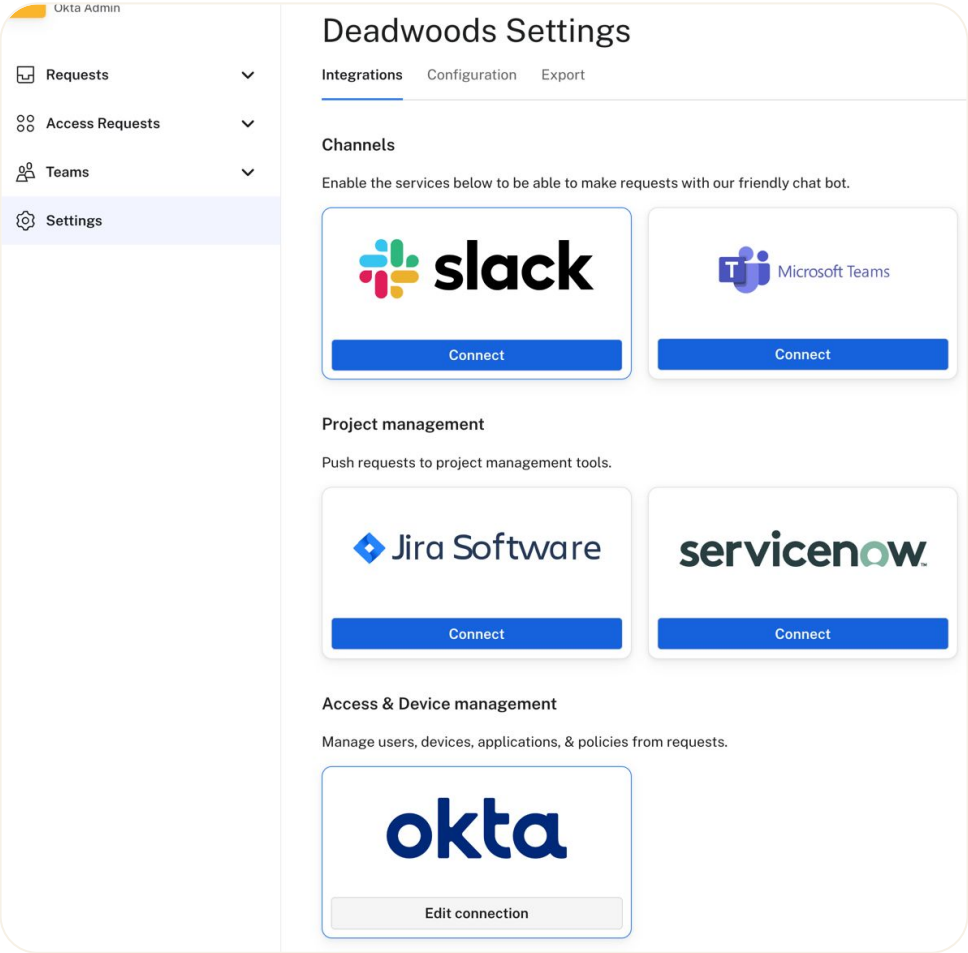
Now we will explore the vanilla Access Requests module:

1. From within the Admin console go to **Identity Governance > Access Requests** (it will open a new browser tab)



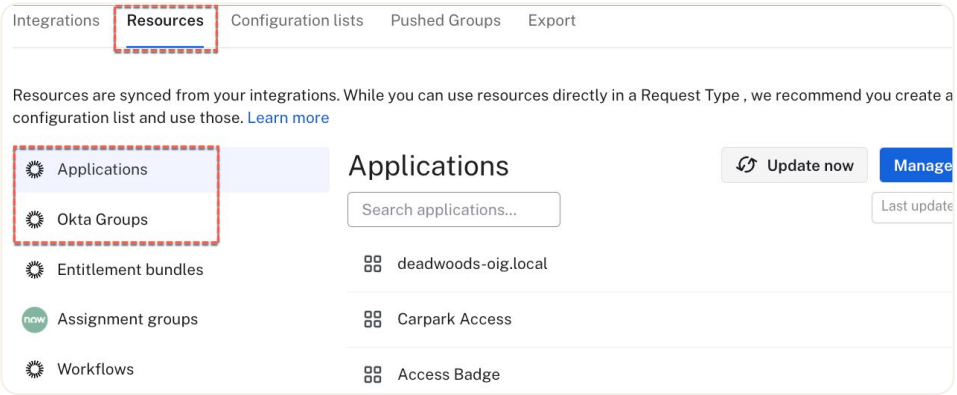
The default landing page is the **Requests > Inbox** view, where any requests awaiting action by this admin are displayed. It is empty.

2. Go to the **Settings** menu item



There are currently five integrations available – Slack, Microsoft Teams, Jira, servicenow and Okta (if you also have Okta Privileged Access, you may see that also). The Okta integration is partially configured, and the others unconfigured. We will configure Slack in the next section.

3. Click on the **Resources** tab towards the top of the page to see the lists of resources from connected systems such as Okta



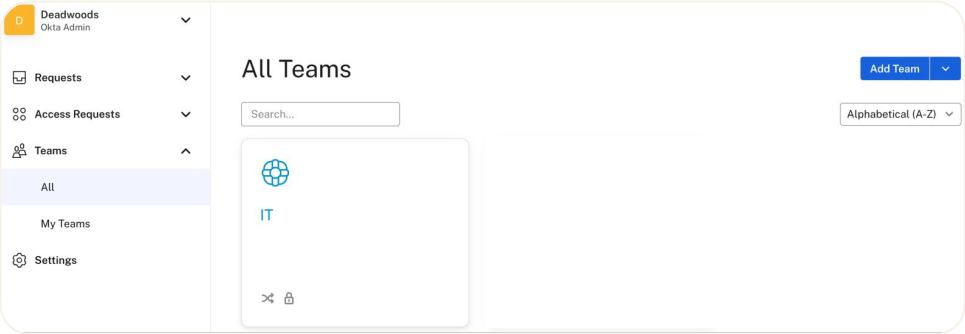
You should see **Applications** (a list of all Okta applications) and **Groups** (a list of all Okta groups). You will also see **Entitlement bundles**, which is part of the new Entitlement Management capability in OIG (covered later in this guide). You may also see a list of Okta **Workflows** if that integration is enabled in your environment (this integration is not covered in this guide). Ignore the Now Assignment groups - that is part of the ServiceNow integration.

> These lists are automatically sync'd from the connected system periodically. However you can force a sync using the **Update** now button.

4. Click on the **Configuration lists** tab. It should be empty.

Configuration lists are custom lists you can create and use in Request Types which are discussed later. They can be custom lists of strings (like a pick-list of text) or subsets of Resource lists (like a specific set of applications or groups).

5. Go to the **Teams** menu item



Teams are used to control who can own a request flow (e.g. security admin team) and can also control who can see and run a flow. You can also use Okta groups for this latter function. See https://iamse.blog/2022/09/10/oig-access-requests-understanding-user-grouping/ for a discussion on Access Request teams vs. Okta Groups in request flows.

By default, there is a single Team called IT. If you click on the tile, you will see that this team is empty (you can populate it or use another one for flows). We will look at the Access Requests and Requests menu items later when configuring and running flows.

Basic Configuration of Access Requests

Now that we've had a look around the two sides of Access Requests, we need to do some basic configuration in preparation to building an access request flow.

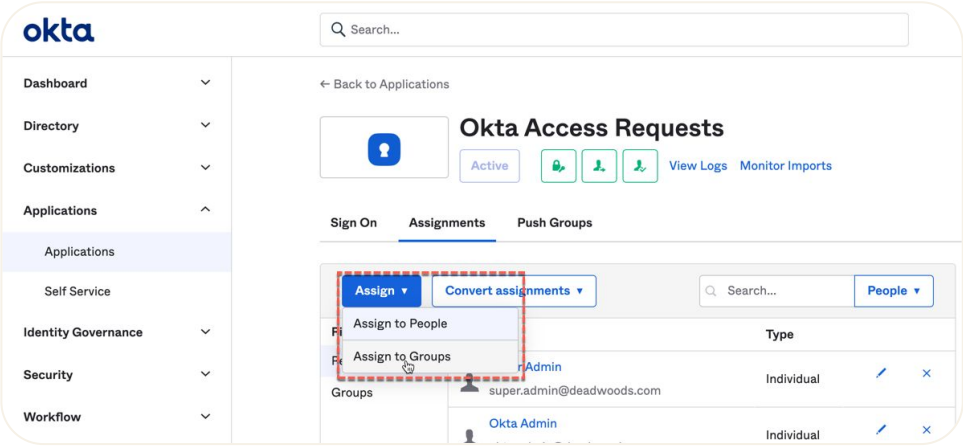
The initial steps below are following the help documentation for configuring Access Requests, creating Teams and Access Requests integration in https://help.okta.com/en-us/Content/Topics/identity-governance/access-requests/ar-configure.htm.

Configure Provisioning

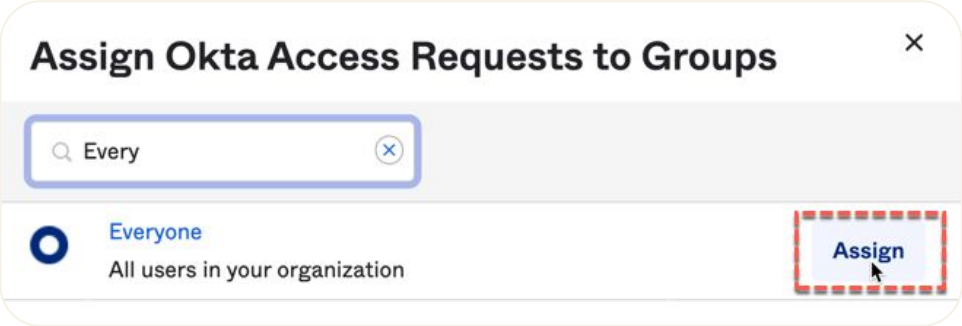
We need to assign users to the Access Requests application in Okta. For this exercise we will use the Everyone group.

> If you only want some of your Okta users to be able to use Access Requests, you could assign specific groups to the application as you would for any other application in Okta.

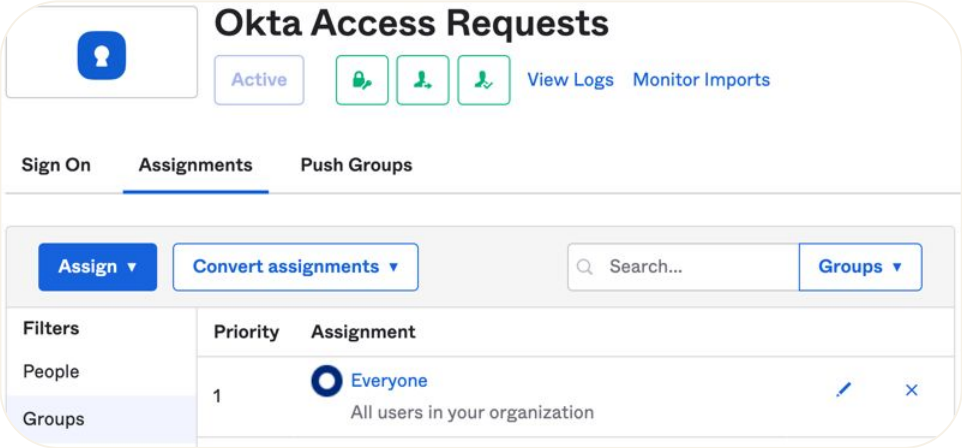
1. In the Okta **Admin console** as an administrator go to **Applications > Applications**
2. Select the **Okta Access Requests** application
3. On the **Assignments** tab, click the **Assign** button and select **Assign to Groups** option



4. Search for the **Everyone** group and click the **Assign** button



5. Click the **Done** button and you should see the group assigned



You will now have every user in your Okta org assigned to access requests and have the Access Requests tile on their Dashboard.

Sync Resources from Okta

This is not required for this exercise.

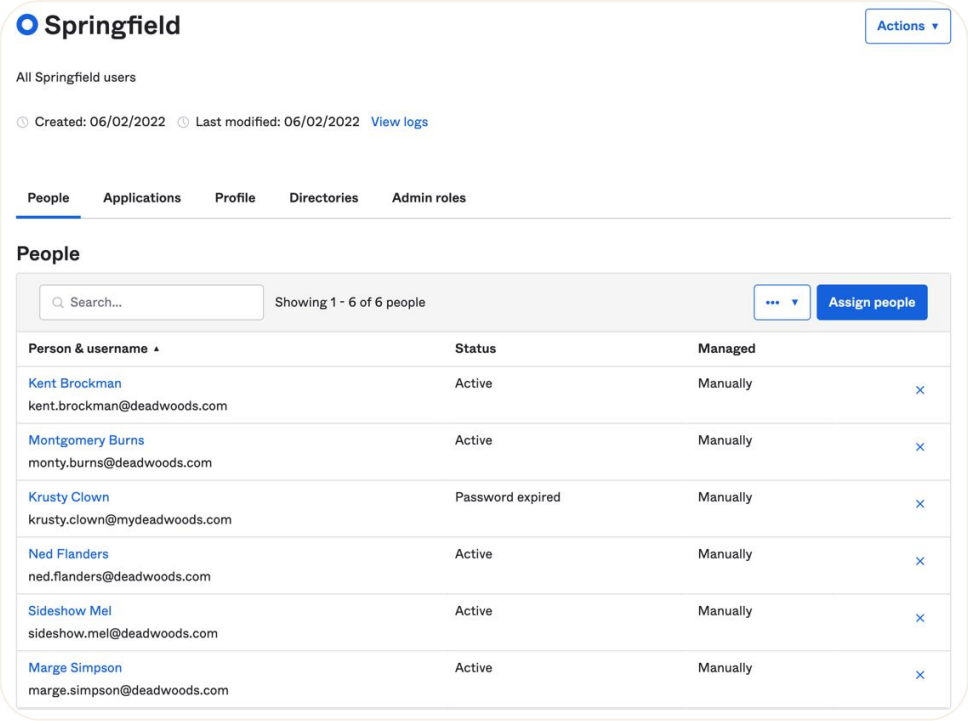
> There may be situations in the future where you need to manually sync resources from Okta. There is an automated sync that runs every twenty-four hours.

Push Okta groups to Access Requests

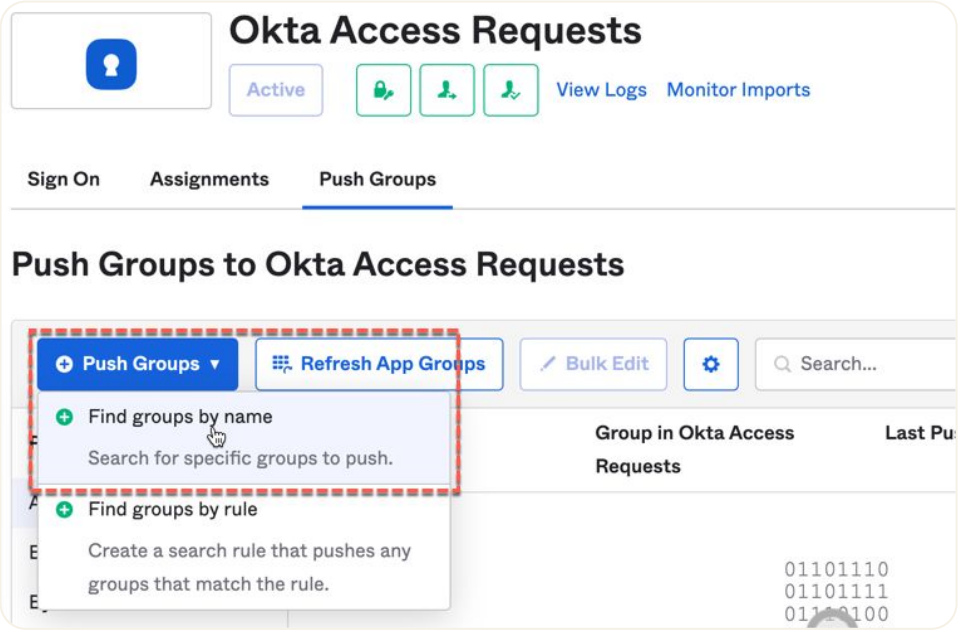
To have groups (with membership) available to Request Types in Access Requests, they must be defined as Push Groups on the application in Okta.

> Groups can be used to define who can see a flow in Access Requests or who can participate in steps in a flow (e.g. group of application owners to review an access request). So, you will probably be updating the push groups as your Access Request flows develop.

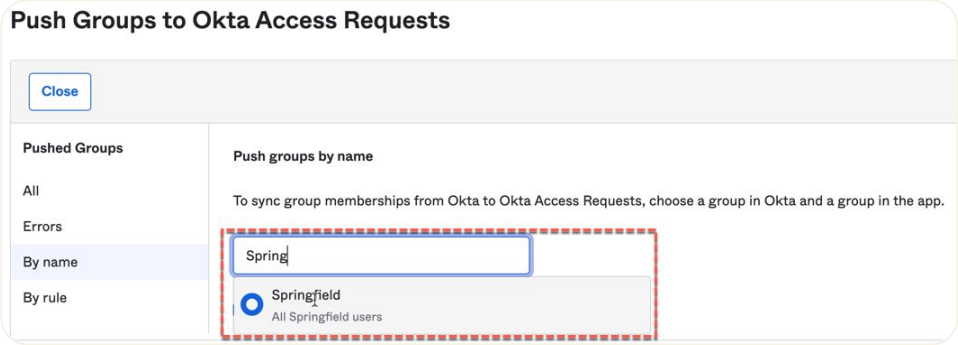
For this example, I have a group called “Springfield” with six users in it. You can use any of your groups in your org.



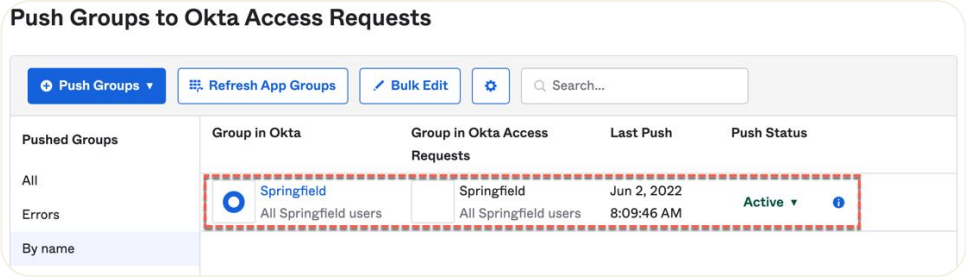
1. Go back to **Applications > Applications** and select the **Okta Access Requests** application
2. Go to the **Push Groups** tab
3. Select the **+ Push Groups** button and **Find groups by name**



4. Search for and select your group



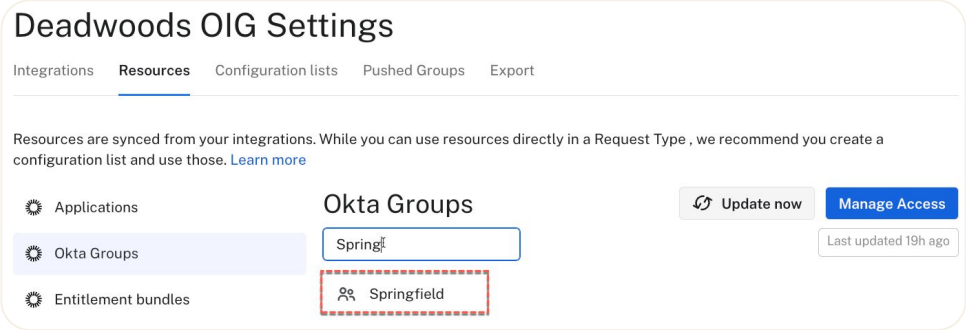
5. Click **Save**. You should see Push Status of “Pushing” followed by “Active”



The last step will push the group to Access Requests.

5. Go into the **Access Requests UI** as you did earlier
6. Go to **Settings > Configuration** and click on the Okta Groups item

You should see your new group there. You can't drill into it to see the users.

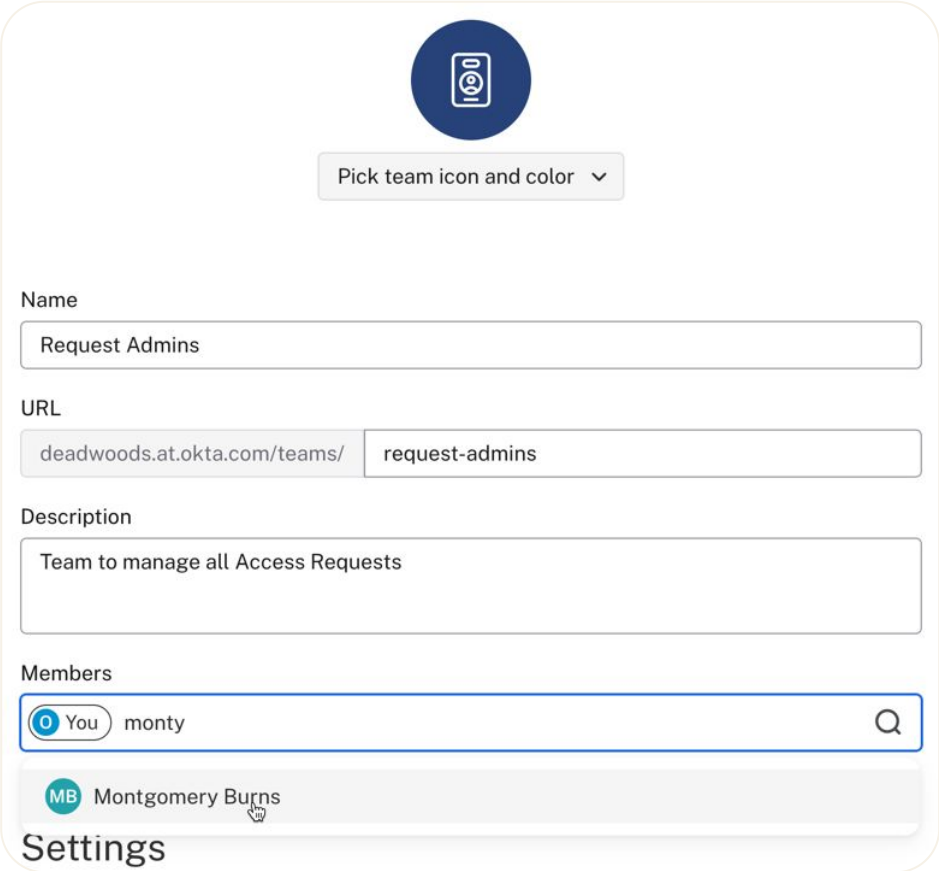


Create an Access Requests Team

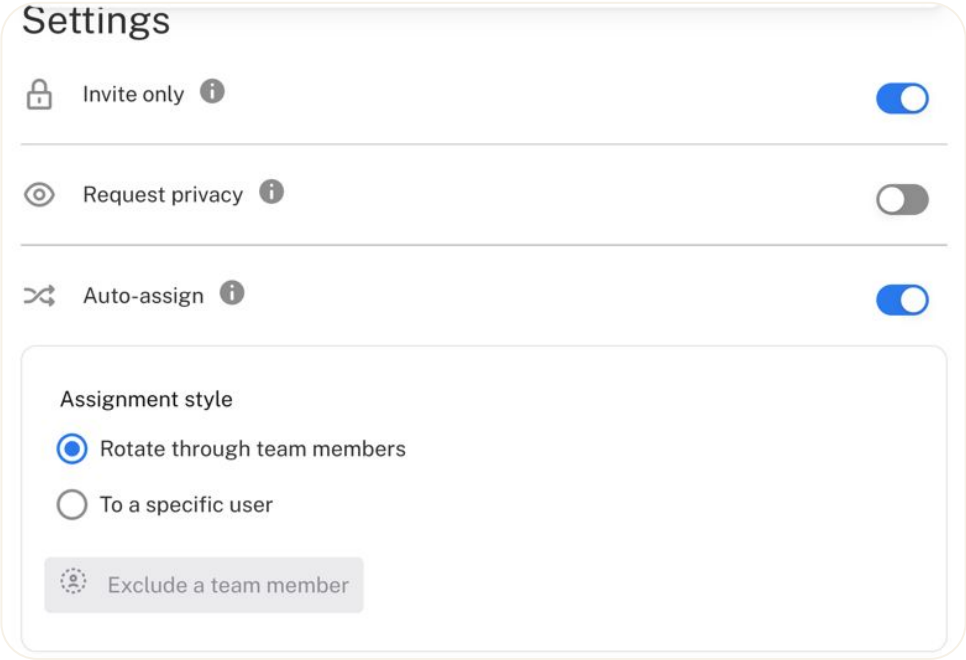
Teams are an Access Requests grouping mechanism. Request Types must be assigned to a Team, and Teams can be used to scope who can run a Request Type or participate in it. As we saw above, there is a default team created called IT. For the sake of the exercise, we will create a team called Request Admins.

As part of any deployment of OIG, you would look at the people and processes including what groups or teams you need to use for the access request flows (and where it makes sense to manage them, in Okta or in Access Requests).

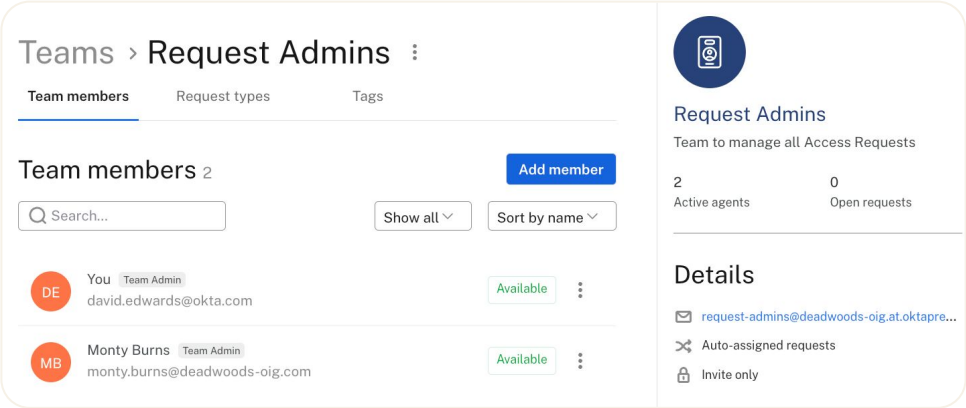
1. In the **Access Requests UI**, go to the **Teams** menu item
2. Click the **Add Team** button
3. Select the team icon and colour (optional), give it a **Name** and **Description** (optional)
4. Add team **Members** (type part of their name and select from the result list)



5. Leave the default **Settings** (see the product documentation for an explanation of each)



6. Click the **Create Team** button

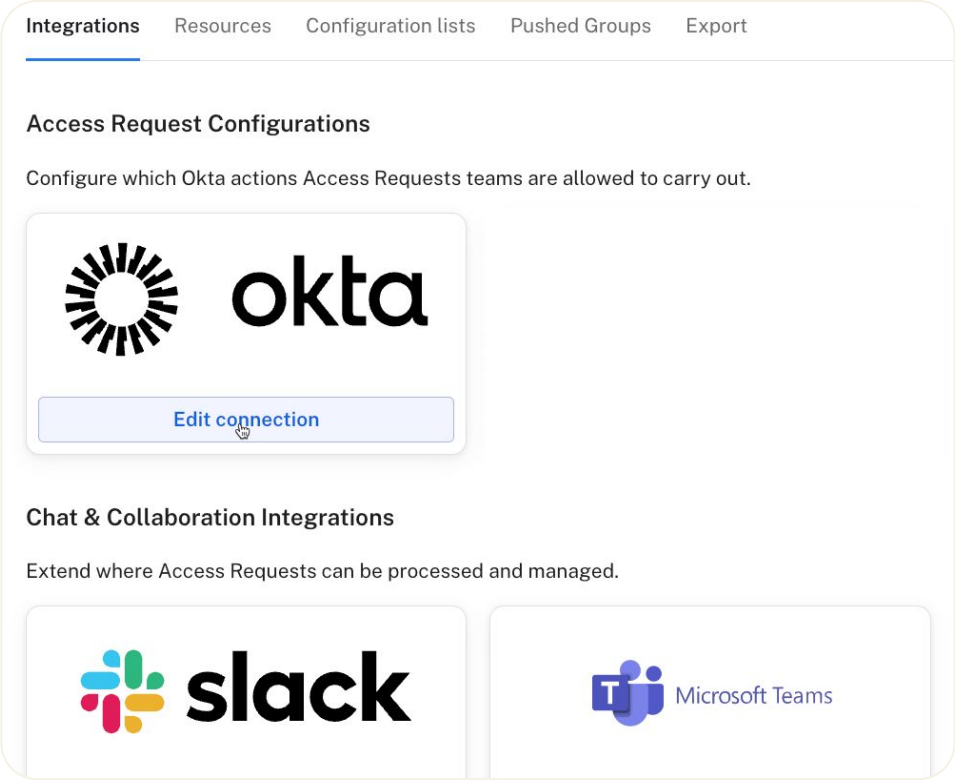


We will use this Team in an Access Request flow later.

Assign New Team to Resources

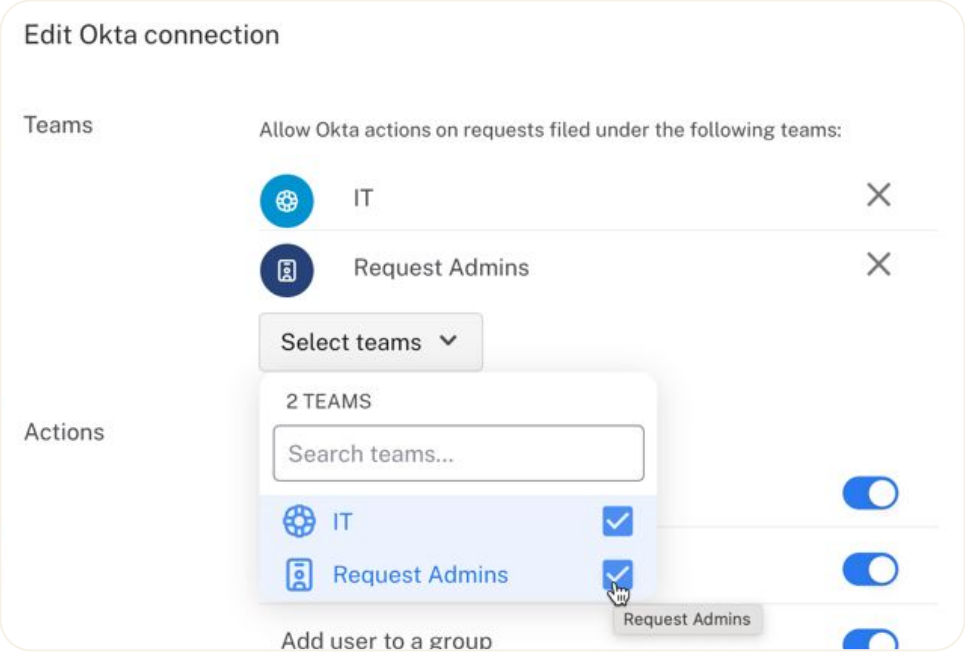
The integration between Access Requests and Okta was partially configured when OIG was enabled for the Okta org. Whenever a new Team is created, the Okta configuration must be updated to allow Request Types assigned to the team to be able to perform the appropriate operations in Okta.

1. In the **Access Requests UI**, go to the **Settings > Integrations** menu item
2. Click on the **Edit connection** button in the Okta tile

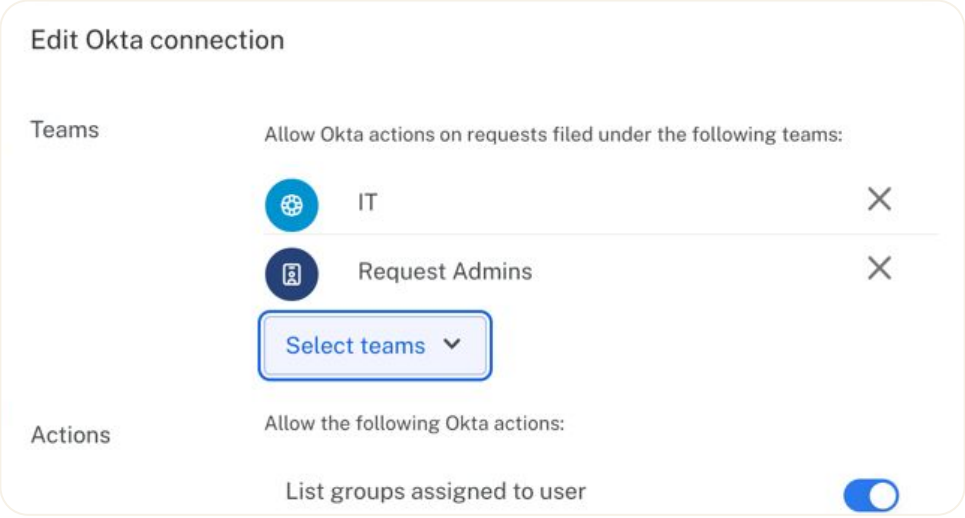


The Edit Okta connection dialog shows the current teams assigned to the connection and the Okta operations allowed.

3. Click the **Select teams** button and select your new team



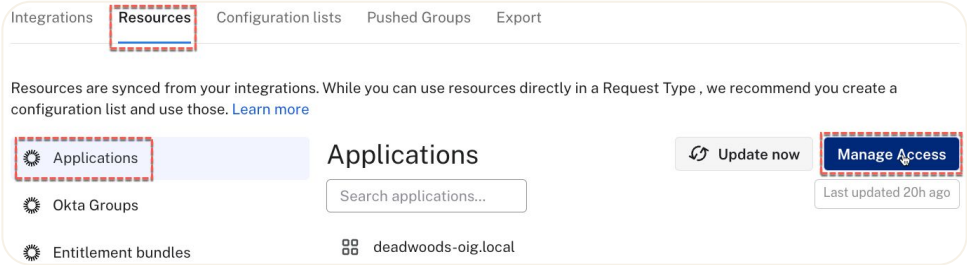
4. Click outside the pulldown list to close the pulldown



5. Click the **Update Connection** button to save the change

> The actions list is common to all teams and relates to the set of actions that could be performed within flows. There is no way to restrict some actions to certain teams.

6. Go to the Resources tab, select the Applications item and select the Manage Access button



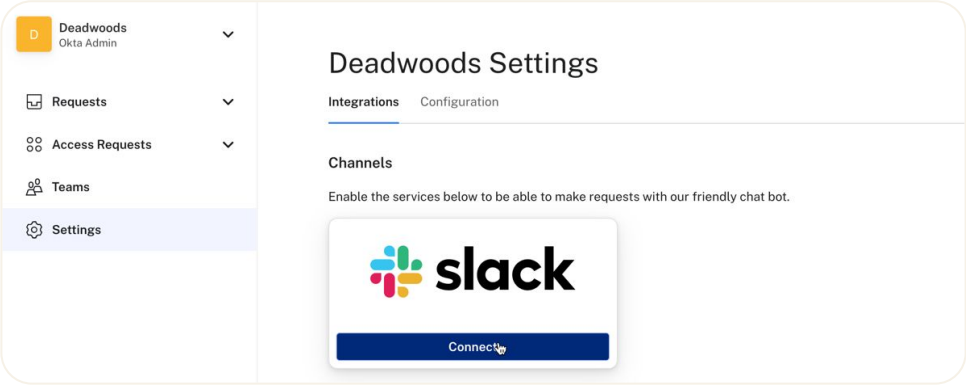
7. Enable the **Request Admins** team and **Save**
8. Repeat for the **Okta Groups** list

> Note – every time you add a new Team that will be involved in a Request Type that's performing actions against Okta, they will need to be added to the connection. This is a common source of confusion with Teams in Okta, if you add Team you need to make sure it is assigned to any Connections, Resources and Configuration Lists the users in that Team will need to use in any of the Request Types.

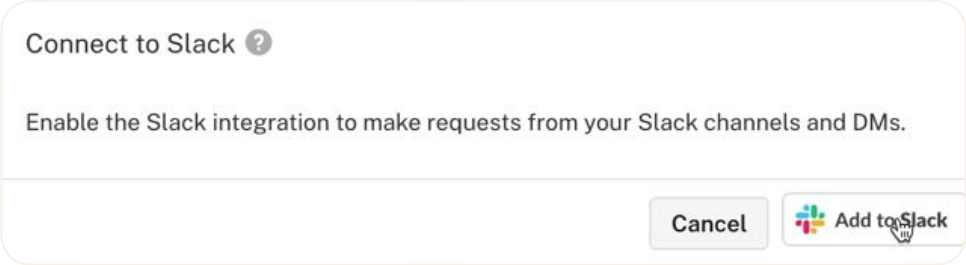
Perform Slack Integration

To interact with Access Request Types via Slack, you need to configure the Slack integration.

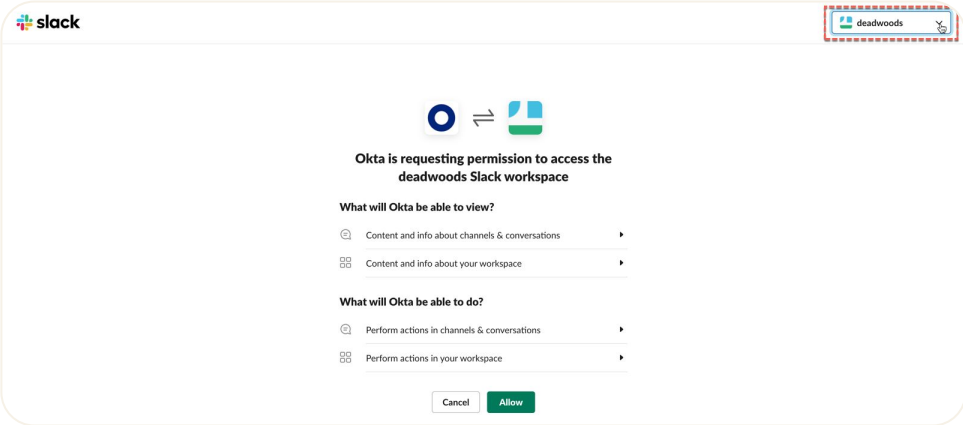
1. In the **Access Requests UI**, go to **Settings**
2. Click on the **Connect** button in the **Slack** tile



3. On the **Connect to Slack** dialog, click the **Add to Slack** button



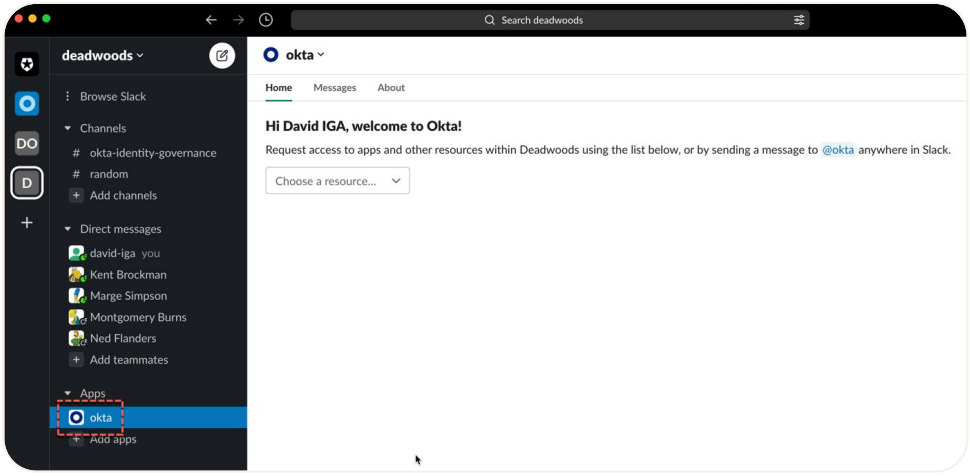
4. A New browser window/tab will open to set the integration permissions. Check that the correct **Slack workspace** is selected and click the **Allow** button



If you don't have a session with the Slack workspace you want to connect to, you may need to login to it.

The Slack connection is now configured.

5. Go into **Slack** as one of the users assigned to the Access Requests app (in our case, anyone as we used the Everyone group) and see that the **Okta app** has been assigned to their workspace.



This completes the Access Requests configuration, and we can now create a simple Access Request Type.

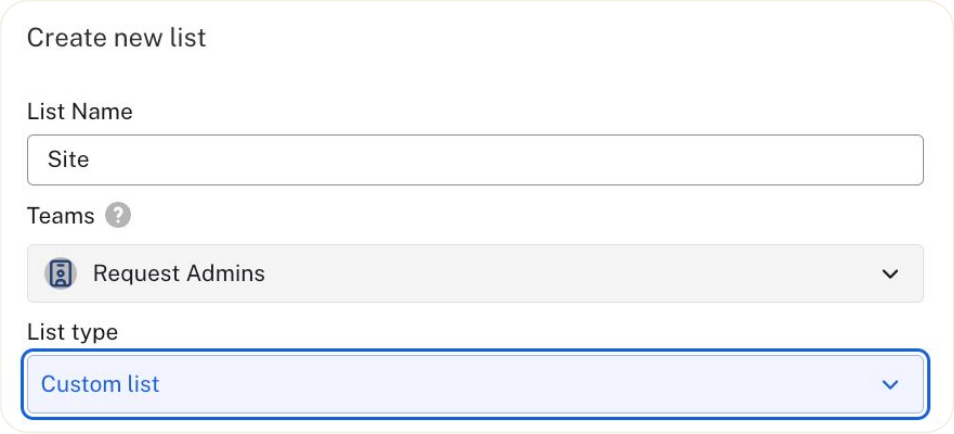
Create an Access Request Flow (Request Type)

To show approval flow creation we will create a simple Access Request flow (Request Type) for one of the applications in your Okta org and set two levels of approval. For this to work, you need to have an application in Okta you can assign users to and users that have a manager set (managerId field in the user profile).

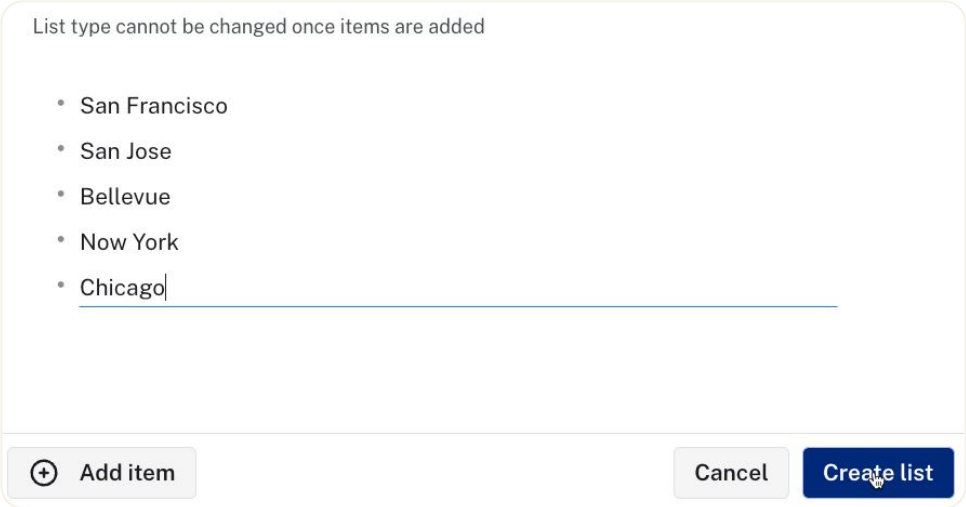
Create Selection List (Configuration Item)

To make the access request flow more interesting we can add a pulldown list that the user can select an item from. It could be anything, perhaps a list relevant to the application. For my example I will create a list of sites.

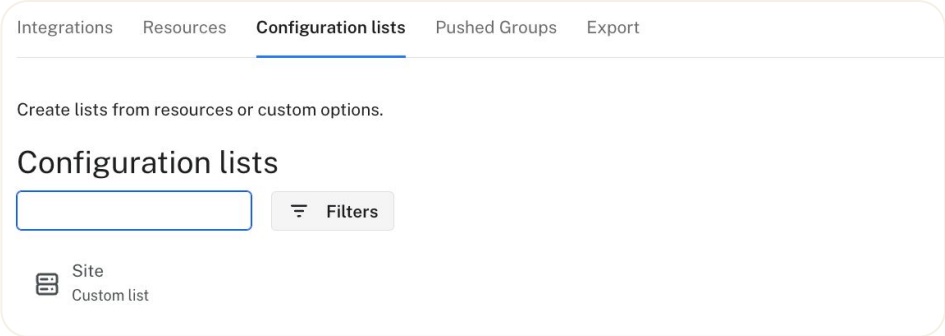
1. In the **Access Requests UI** as an administrator, go to **Settings > Configuration lists**
2. Click the **Create new list** button
3. Give the new list a **Name**
4. Click the down arrow to select **Teams** for this Configuration list, the new **Request Admins** Team from above
5. Click the down arrow to select the List type as



6. Select the next field that has Item name and type the first of your values (the example is using a set of sites)
7. Click the **+ Add item** button to add the second item
8. Continue adding items with the **+ Add item** button



9. Click the **Create** list button to save the list

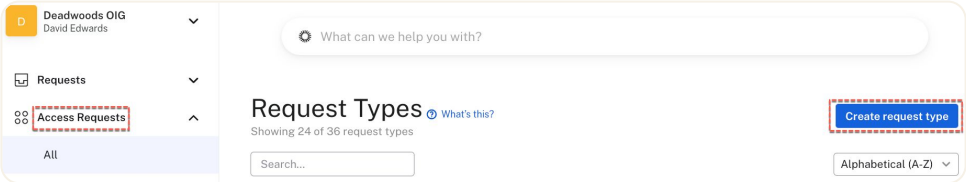


Now we're ready to create the flow.

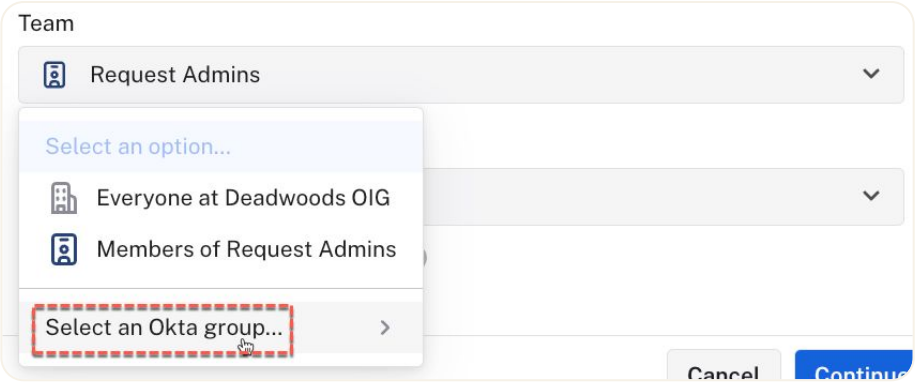
Create Request Flow

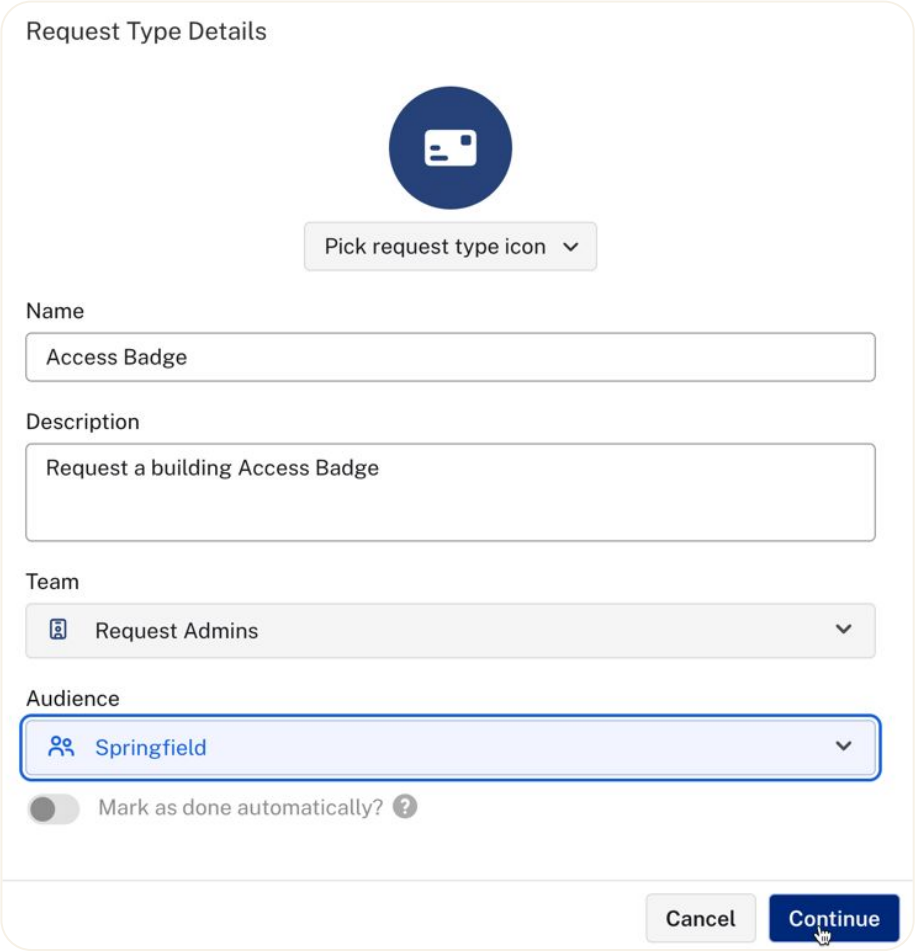
To create the flow:

1. In the **Access Requests UI** as an administrator, go to **Access Requests**
2. Click the **Create request type** button



- On the **Request Type Details** dialog, pick a workflow icon (optional), give the workflow a **Name,** and enter a **Description** (optional)
- You need to select a **Team** to own this workflow. Select the new team we added earlier
- The **Audience** is who can see and run this workflow. You can say everyone in the organisation, a Team, or one of the groups synced from Okta. For this example, select the group that was pushed from Okta (i.e. choose **Select an Okta group…** >)

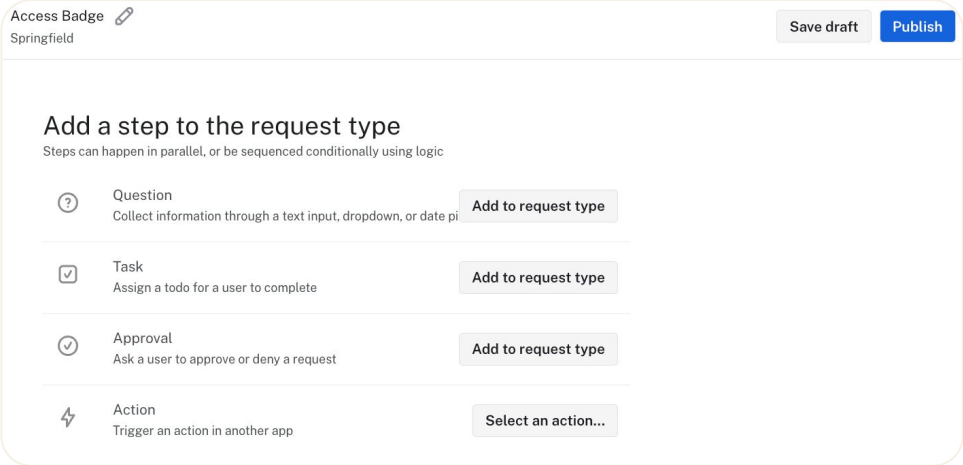




Note that you can't select the Mark as done automatically toggle. The Request Type needs steps before this can be done.

4. Click the **Continue** button

The flow is now ready for the steps to be added.



Note that the flow is in a Draft state and needs to be published to be used. Once a Request Type is published, you normally must revert it to Draft to modify (which removes it from use).

Flows have five types of steps; **Questions** (prompt for input), **Tasks** (assign a todo), **Approvals**, **Actions** (which is how the Okta functions are implemented), and Timers (to set a wait until/for mechanism into the flow). The Timers don't show in the above view. We will add some Questions, two levels of Approval and an Action to add the user to the application in Okta. The UI has a dot-matrix area where you can work on the steps – I'll refer to this as the flow editor.

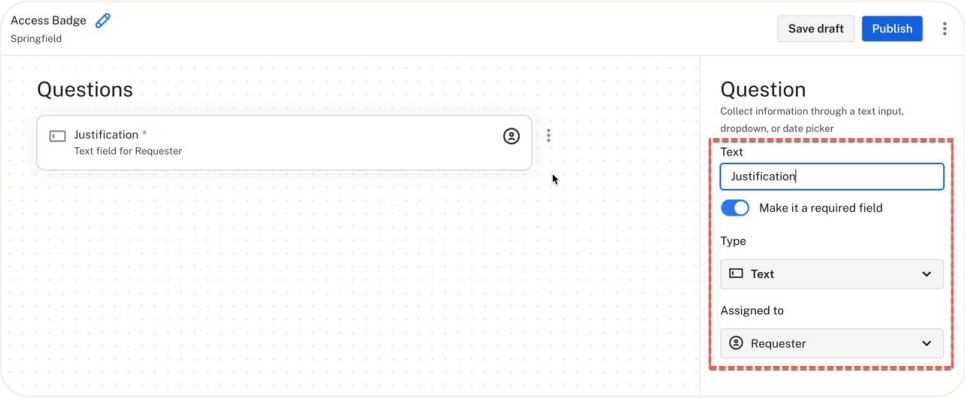
Add Questions to Request Flow

The first step in an access request flow is to get the requester to supply some information, such as a justification. We will add three fields to help the reviewers decide if access should be granted: a justification, the site they need badge access to, and a date they need access to.

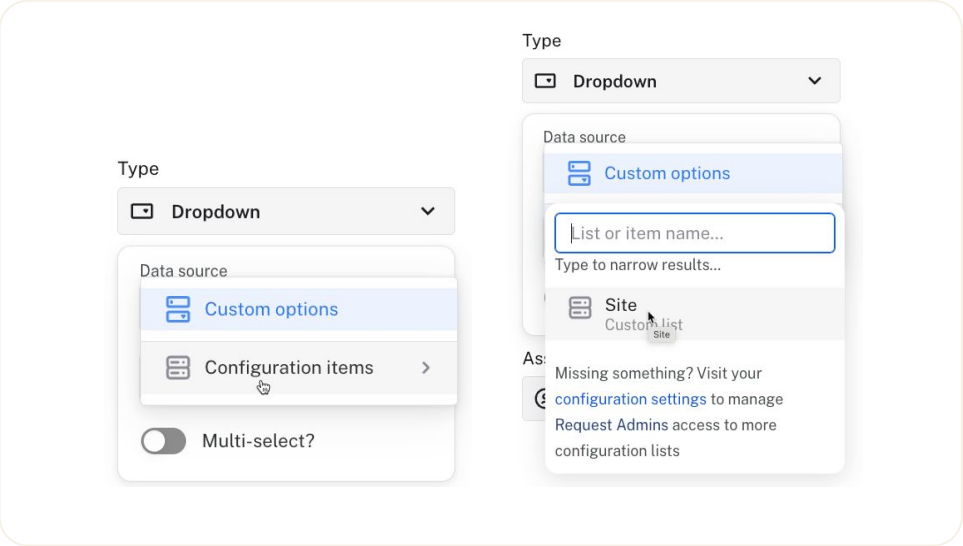
You can use whatever fields you like.

5. Click on the **Add to request type** button beside **Question**
6. Ignore the flow editor body (box in the middle of the page) and go to the right of the screen under the **Question** heading
7. Enter the title of the field (e.g. Justification) in the **Text** box (and notice that the box in the middle field is updated)
8. Leave it as a **required field**
9. Leave the **Type** as **Text**
10. Leave it **Assigned to** the **Requester**

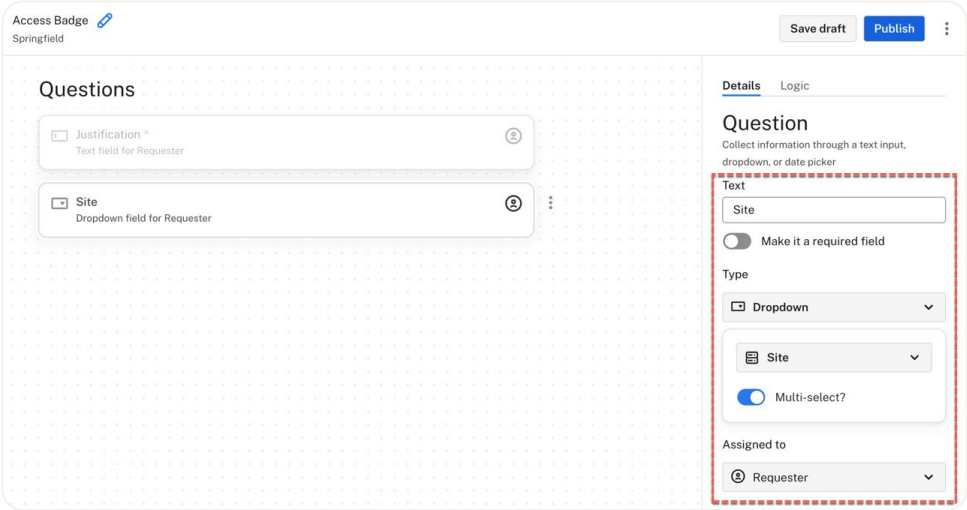
> You have the option of asking other people (like the requester's manager) to supply information in a question, which may be useful for more complex flows requiring multiple inputs.



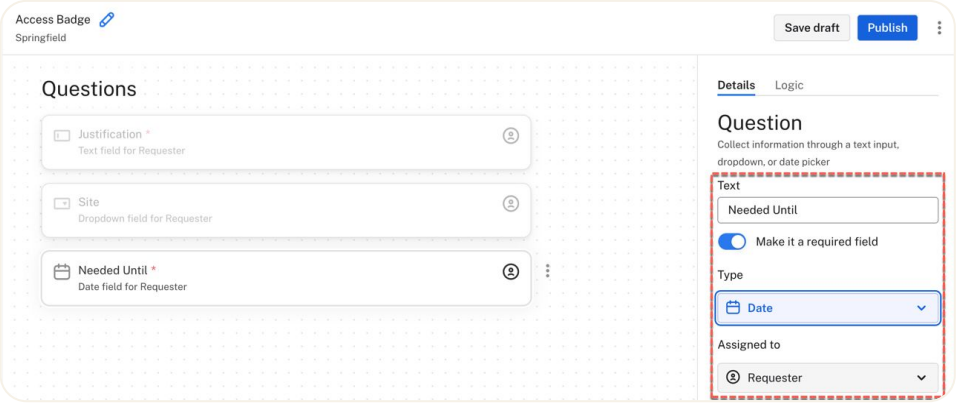
11. Click the **Question** button at the bottom of the flow editor
12. Give the new question a name (**Text**) that is relevant to your configuration list from above (e.g. Site for my Sites dropdown list)
13. De-select the **Make it a required field** (i.e. it's optional)
14. Change **Type** to **Dropdown**
15. Under Dropdown, in the Data source field, select **Configuration items** then select the configuration list you created before and select **Multi-select** (users can select multiple from the list)



16. Leave it **Assigned** to the **Requester**



17. Click the **Question** button at the bottom of the work area
18. Create a mandatory **Date** field

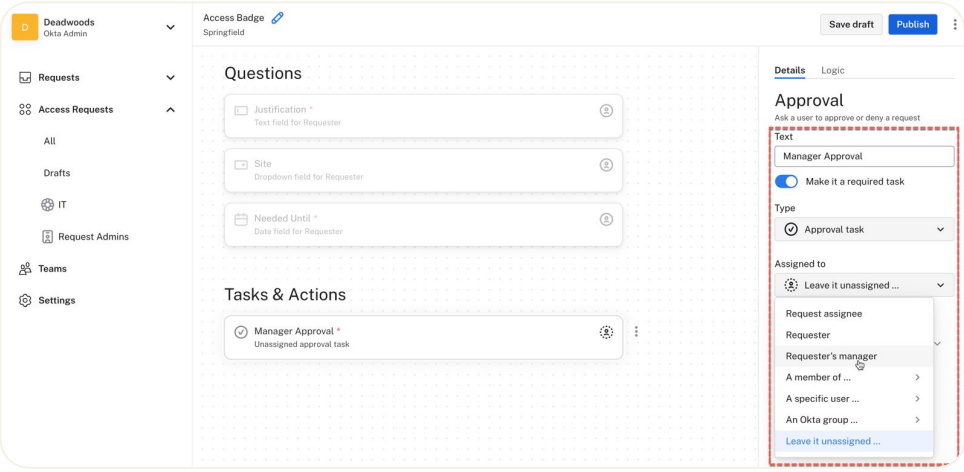


> The order of fields will be reflected on the dialog presented to the requester. Once the Request Type is saved, you can drag the questions around into a different order.

Add Approvals to Request Flow

Next, we will add two levels of approval with some conditional logic.

19. Click the **Approval** button at the bottom of the flow editor
20. In the right pane give the task a name (like “Manager Approval”) and required
21. Leave the **Type** as **Approval Task**
22. From the **Assigned** to pulldown, select the **Requester’s manager**

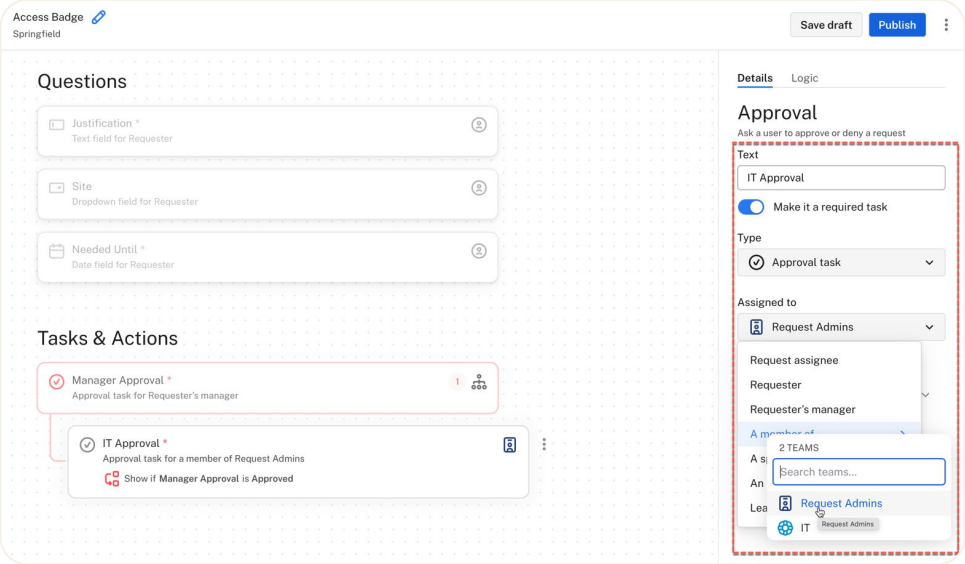


Three things of note:

1. Even though the step is called an Approval, it is just another type of “task”
2. There is a lot of flexibility in who you can assign the approval step to. If it’s the manager, Access Requests is using the user<->manager relationship from Okta. But it could also be a member of a Team, a specific user, an Okta group, or the requester themselves. It can also be dynamically assigned by the process owner at runtime
3. You can set a due date (in minutes/hours) on the review

This approval step will run once the requestor answers all the questions. With the second level of approval, we will apply some logic.

23. Click the **Approval** button at the bottom of the Request Type editor
24. In the right pane give the task a name (like “IT Approval”) and required
25. Leave the **Type** as **Approval Task**
26. From the Assigned to pulldown, select the **A member of** option and select the new team you created earlier



Your view will be slightly different as we haven't applied the logic yet.

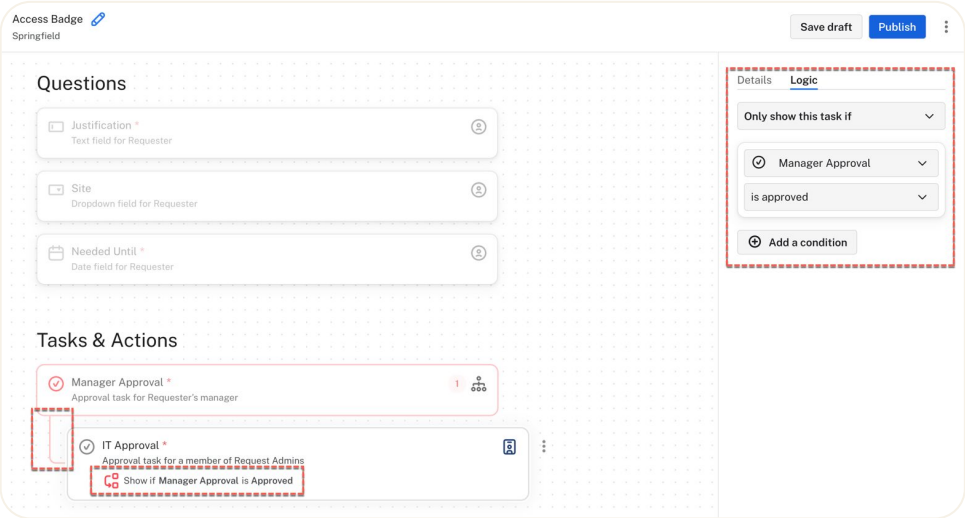
> Note the icons on the task cards (left side of the above screen shot). The one for Manager Approval is a hierarchy, indicating the requesters manager. The one for IT Approval is the icon we chose when creating the Team. You can hover the mouse over the icon to see the assignment.

Now we can apply logic to only run this approval if the manager has approved the request.

27. Click the **Edit logic** button at the bottom
28. Under the **Logic** tab, select **Only show this task if**
29. On the second field, select the **Manager Approval** task (or whatever you called the first approval step above)
30. On the third field, select is **approved**

This is telling the engine to only run this second approval step if the first approval step resulted in an approved outcome. As you fill out these fields, the flow displayed in the flow editor changes to putting the step tied to the previous step and shows a summary of the logic.

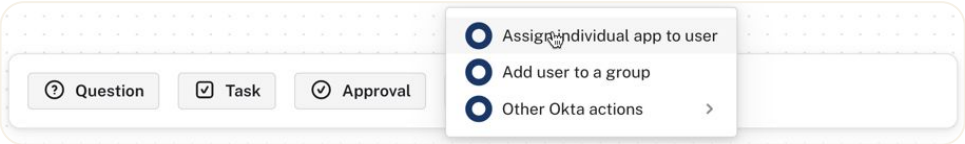
> You can stack multiple rules. For example, I could have multiple approvals reliant on manager approval, but tied to specific selection fields from the questions (like Manager Approval is Approved AND Site is Bellevue) for specific requirements.



Add Okta Action to Request Flow

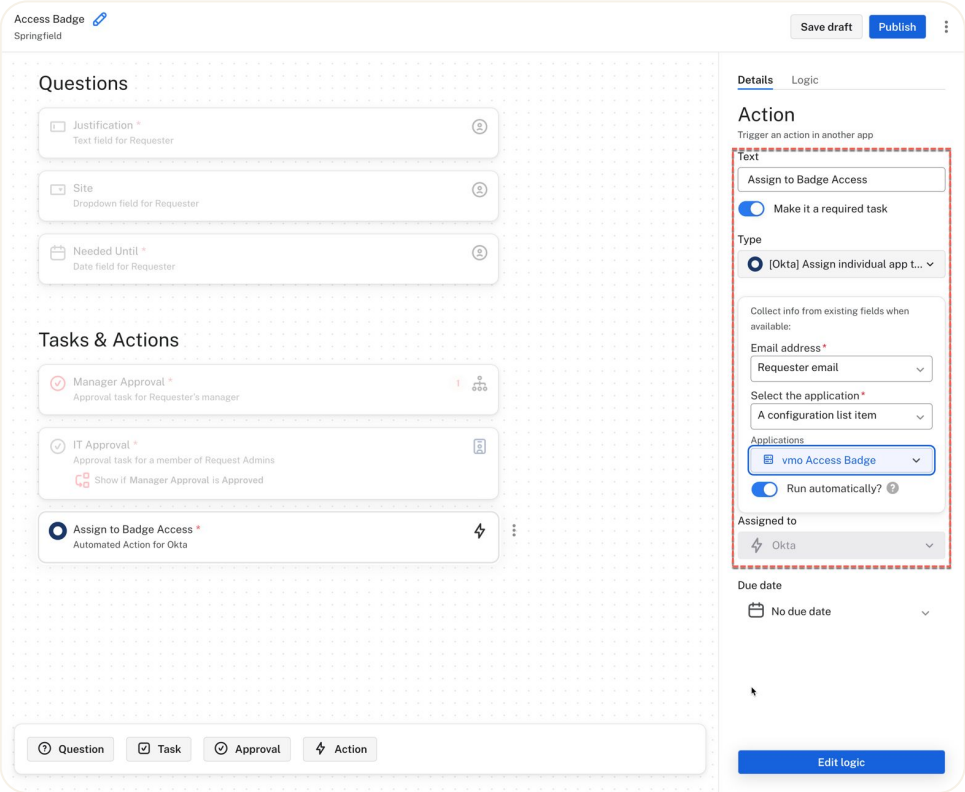
The final step to add is the action to add the requester to the assigned application.

31. Click the **Action** button at the bottom of the flow editor
32. Select the **Assign individual app to user** option



There are many Okta actions that can be run in a workflow. The two that will be most used are **Assign individual app to user** and **Add user to a group**. You could explore the other actions available, some of which will retrieve data from Okta and display it in the details of the flow.

33. In the right pane give the task a name (like “Assign to Badge Access”) and make it required
34. Leave the **Type** as **[Okta] Assign individual app to user**
35. Select (enable) the **Run automatically** option (this will change the **Assigned to** value to “Okta”)
36. In the expanded dialog, set the **Email address** as **Requester email**, the **Select the application** field as **A configuration list item**, and select the **Application** from the list

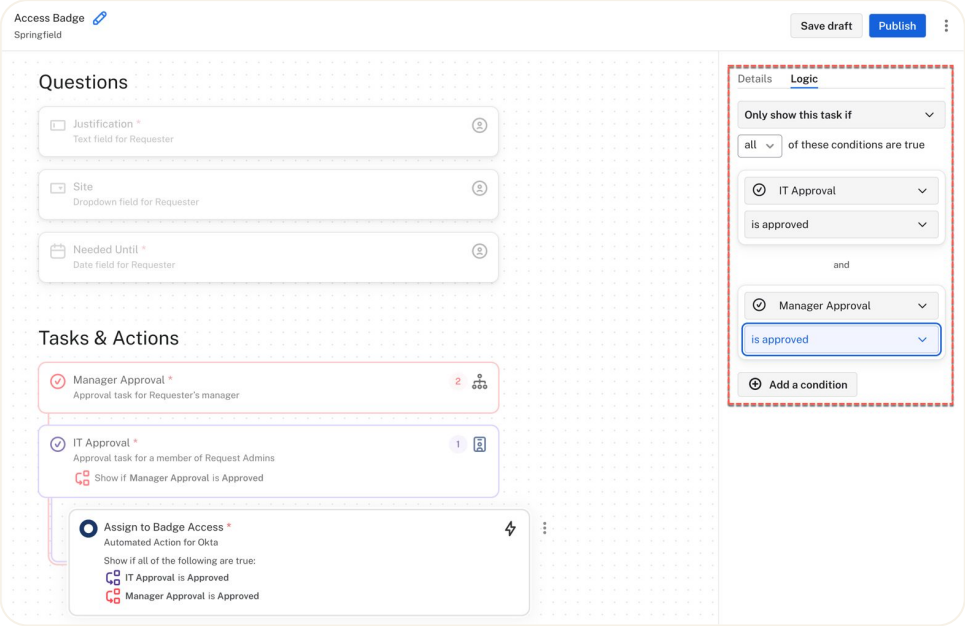


This action is coded to refer to the Applications configuration list (we looked at this earlier). If the action was the Add user to a group action, the Groups configuration list would be used.

We also need to add logic, so this step only runs when both approval steps result in an Approval.

37. Click the **Edit logic** button at the bottom
38. Under the **Logic tab**, select **Only show this task if**
39. On the second field, select the **IT Approval** task (or whatever you called the second approval step above)
40. On the third field, select is **approved**
41. Click the **Add a condition**
42. Select **Manager Approval** and is **approved**

You now have compound logic. The second rule is superfluous as the step won't run unless the IT Approval step is approved, and it won't run unless the Manager Approval step is approved. However, it does show how you can stack rules and also how the Request Type editor shows compound rules (different coloured lines joining the previous steps and corresponding icon colours in the rules).



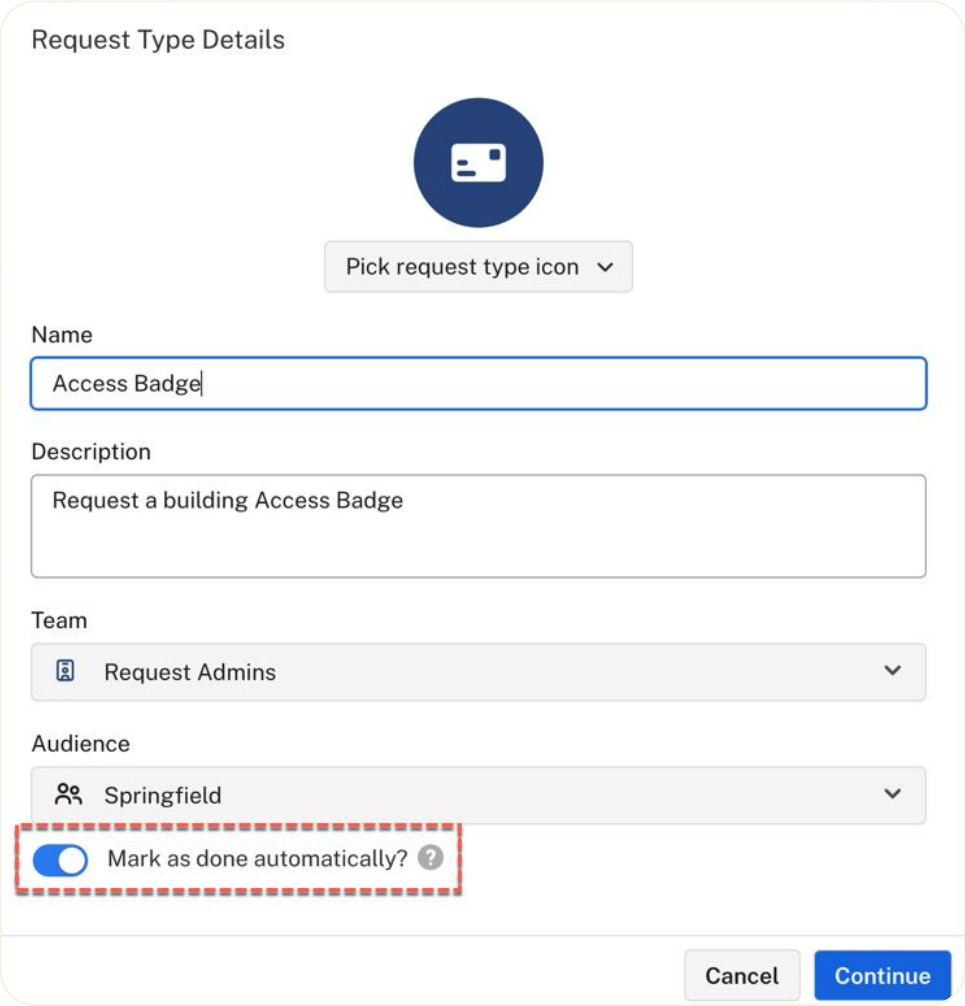
Note the toggle at the top of the Logic section – "of these conditions are true". It can be all ("AND" between rules) or any ("OR" between rules).

That is the last step to add to the Request Type. We have one more thing to do before publishing.

Update Request Flow for Automatic Close

Before publishing the Request Type, we need to set it to automatically close once the Okta action (last step) is run.

43. Click on the **pencil icon** beside the workflow name
44. Enable the **Mark as done automatically**? option at the bottom



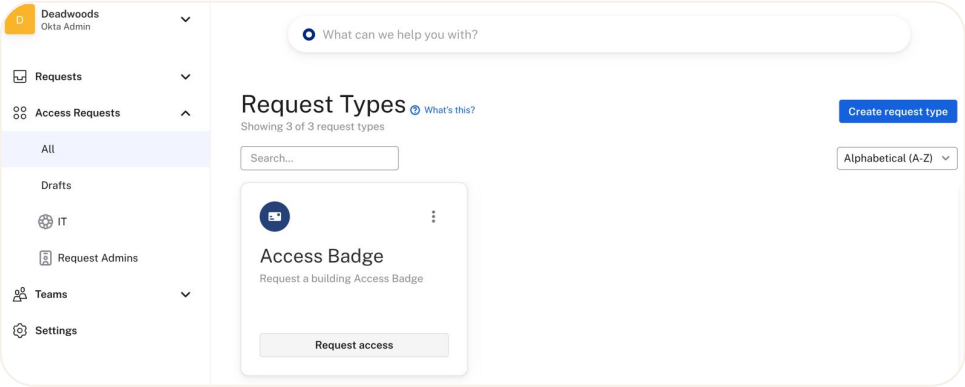
43. Click the **Continue** button

This will automatically flag the Request as done, rather than leaving it open and having Access Requests send reminders.

Publish Request Flow

To publish the flow:

1. Click the **Publish** button on the top of the screen
2. Go to **Access Requests > All**
3. Your new Request Type should be showing



You can Edit, Duplicate and Delete the Request Type from here (vertical three dots icon). Duplication makes creating new Request Types a lot simpler.

Use an Access Request Type from the Access Request UI

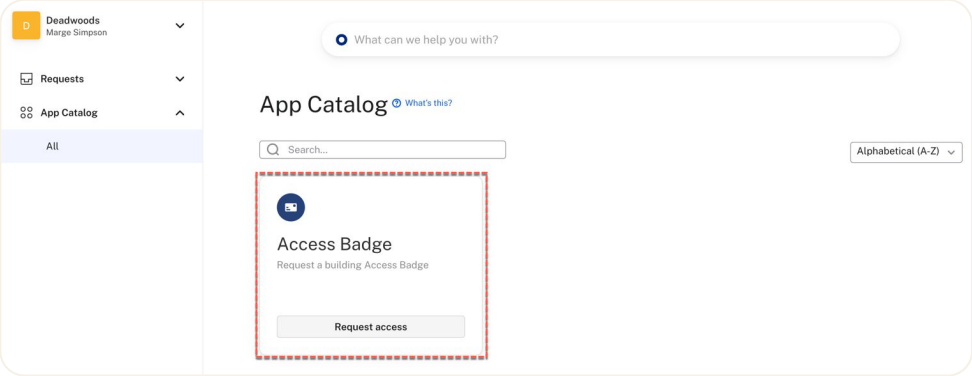
With the flow created, the last step is to test that it is working as expected. For this section you will need a user in the group you assigned to the Request Type (Springfield in my case) who has a manager defined in Okta.

Requester Raised Access Request

Access Request via Access Request UI:

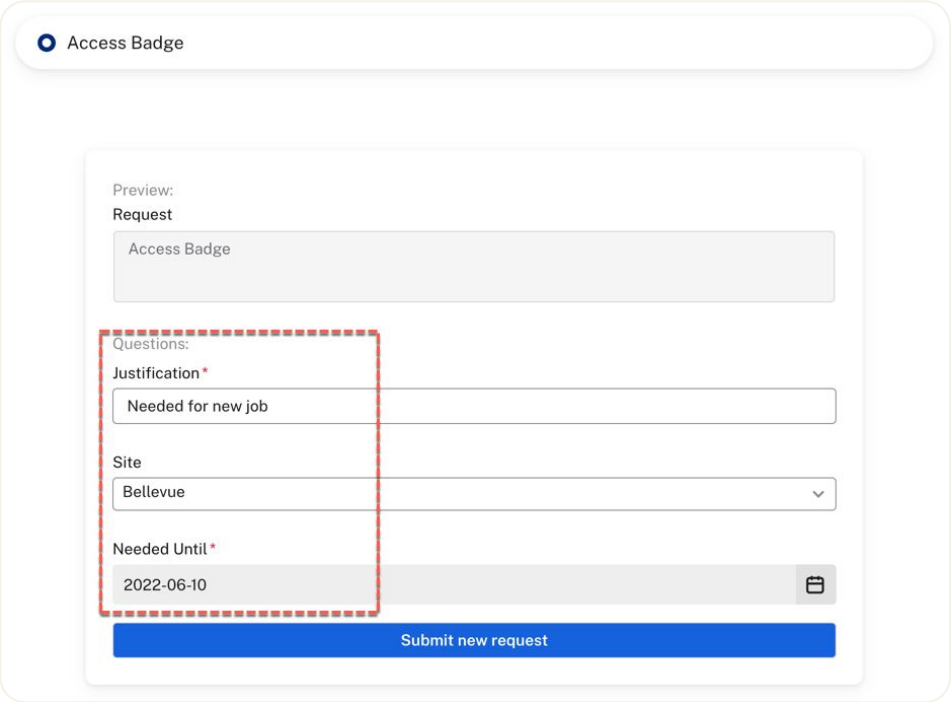
1. As this user log into your **Okta Dashboard**
2. Click on the **Okta Access Requests** tile to open the Access Request UI

You should see the new Access Request workflow showing under the **App Catalog** heading



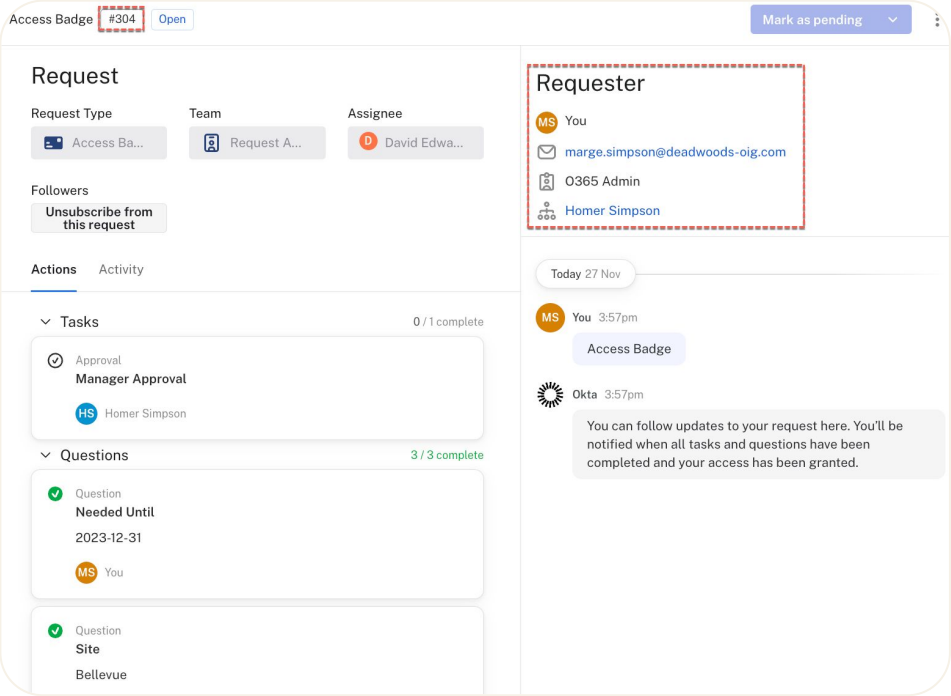
3. Click the **Request access** button

You should see the three questions you added to the flow in the dialog.



4. Click the **Submit new request button** to submit the request

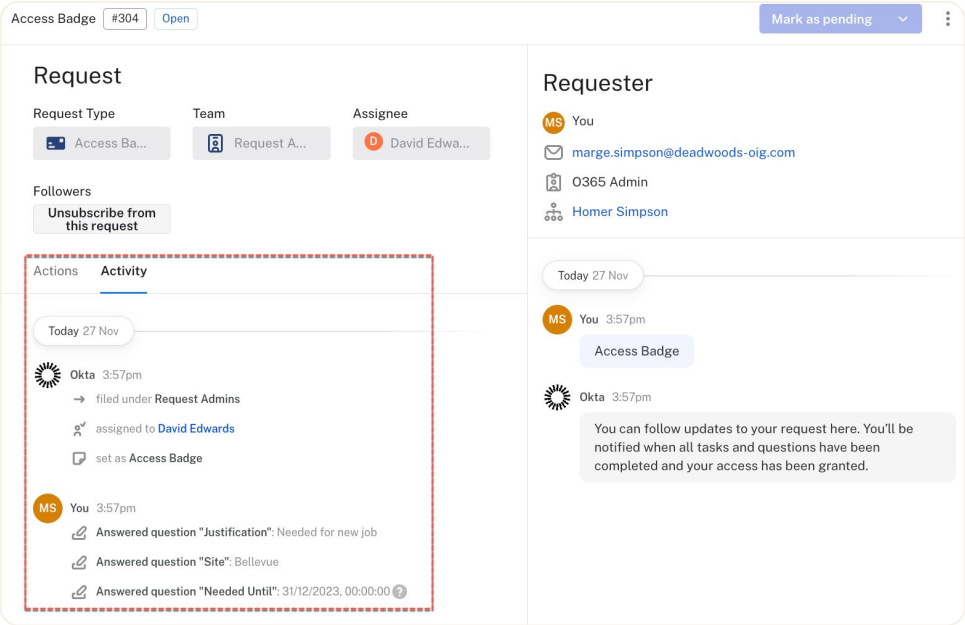
The view will change to show the details of the Request, including the right panel showing the summary information about the requester, and the actions in the main section of the page. Note the number at the top of the request (#304 in the screenshot below) which is the unique identifier for this request.



The Actions tab is useful to see all the actions (completed and pending). It is in reverse time order (i.e. newest first) and the next action is the Manager Approval step with the users manager (from their user profile in Okta) showing.

> If you see that the task is “unassigned” that may indicate a problem with your managerId data stored in Okta. Access Requests is looking for email but is being passed the userid in the managerID field by default. If they are different (manager has different email to userid in Okta) then you may need to look at using the manager ID value rather than the userid or change the mapping of Okta profile attributes to Access Request attributes. This is outside the scope of this document.

5. Click on the **Activity** tab to see a summary of all activity in the request.



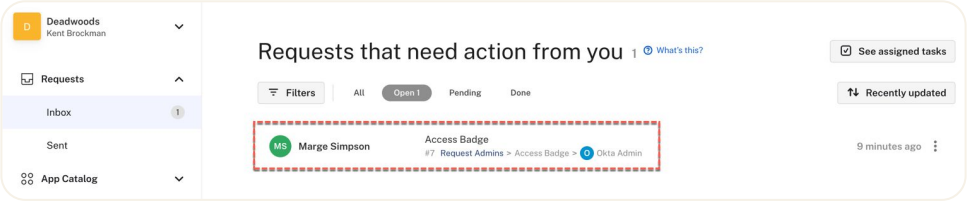
This is the best view to see a summary of all activity and messages.

Next, we will review the access as Kent.

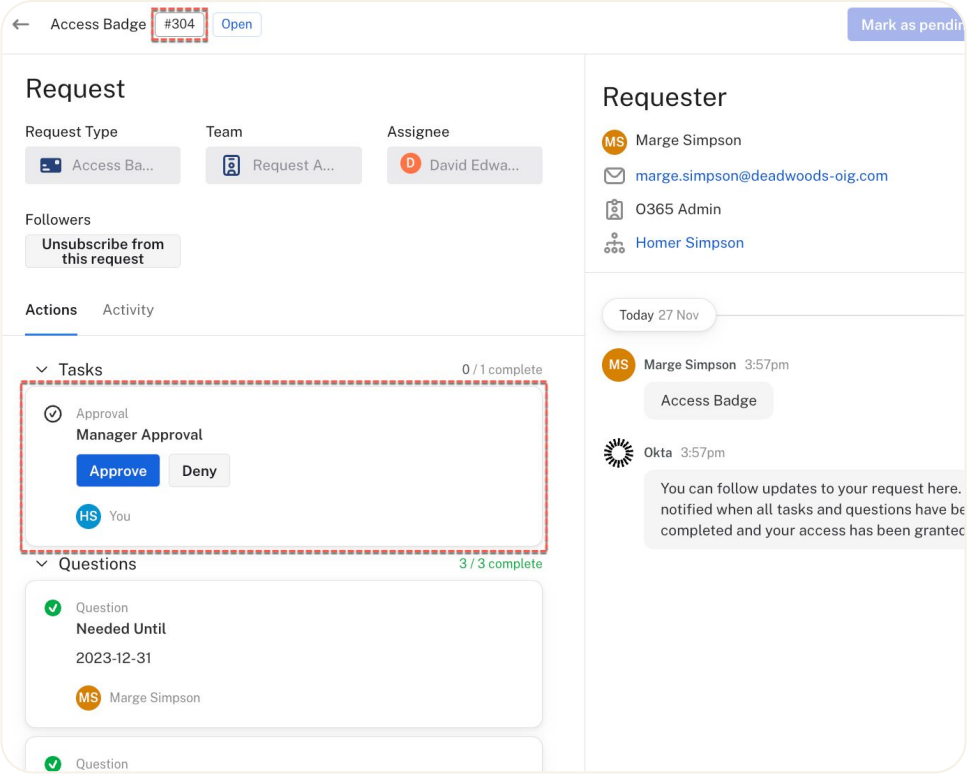
Manager Reviews Access Request

We need to act as the manager. If you want to keep your requester session active, you may want to use a different browser, a new browser tab in incognito mode, or a different browser profile.

1. Log into the **Okta Dashboard** as your manager
2. Click on the **Okta Access Requests** tile
3. The default view is the **App Catalog**. Click on the **Requests** menu item to see the new request in the Inbox.

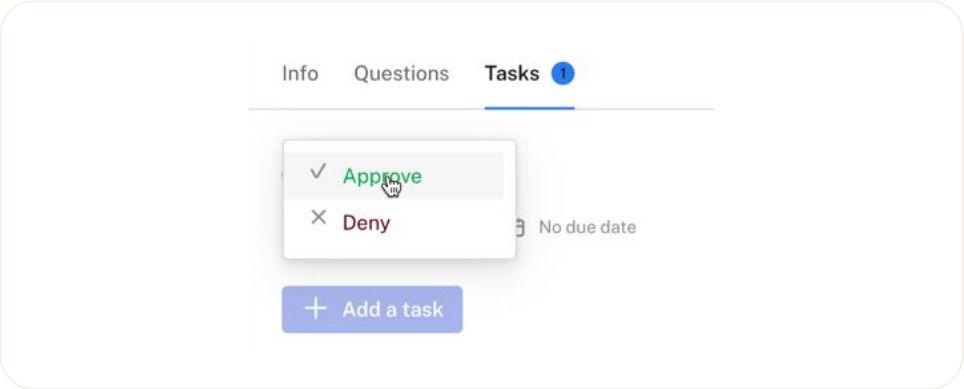


4. Click on the item to open it



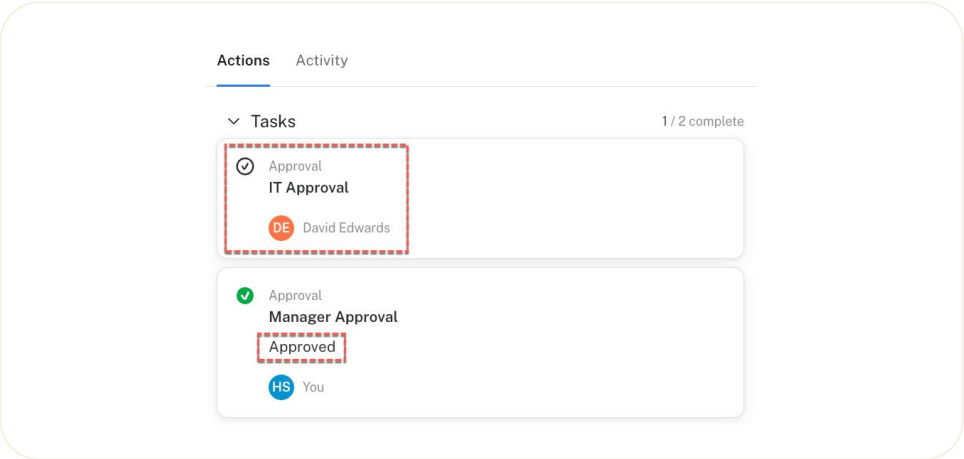
The view is the same as the requester's view. However as this user is the first level reviewer, they can action the request (i.e. Approve or Deny).

5. Select **Approve**



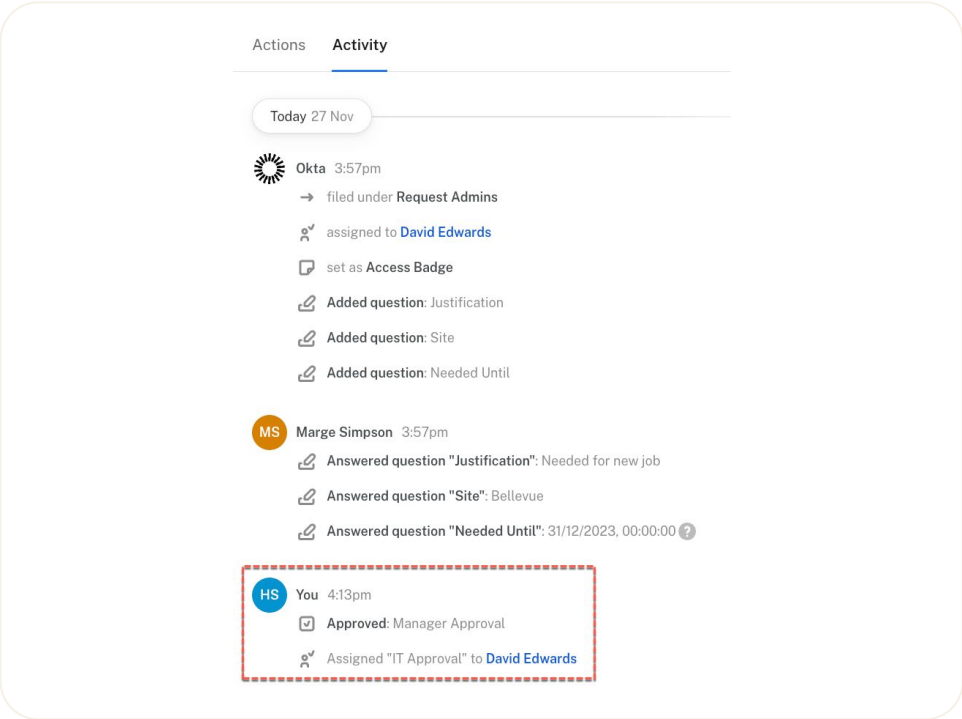
Obviously, the manager could Deny the request and the flow would stop here.

After a brief pause the flow will update the view to show it's approved and the next level of approver is shown.



This next level of approver was the Request Admins group which has two users and one has been randomly selected as the reviewer.

6. Select the **Activity** tab



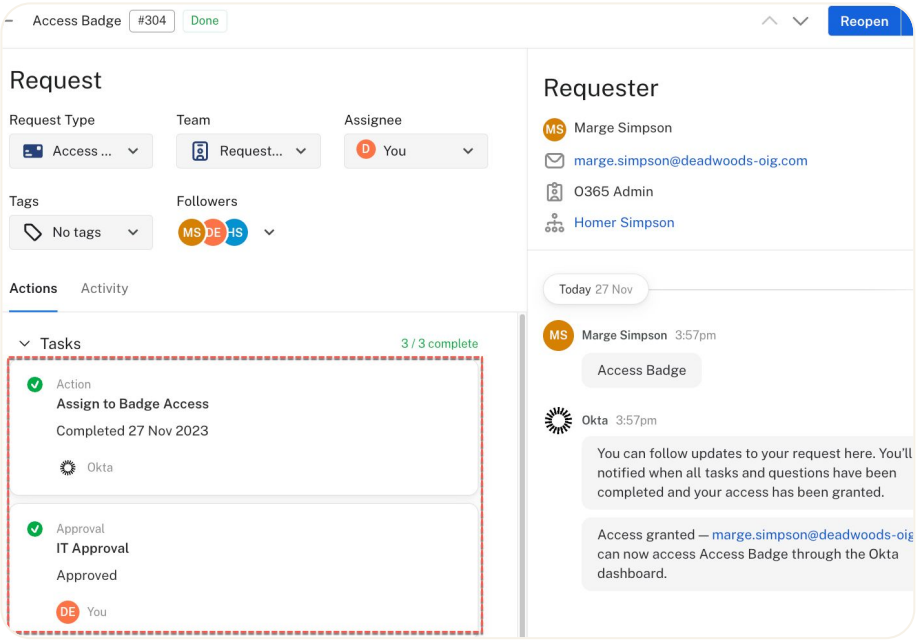
It confirms that the Manager Approval step has been completed and the IT Approval step has been assigned to another user.

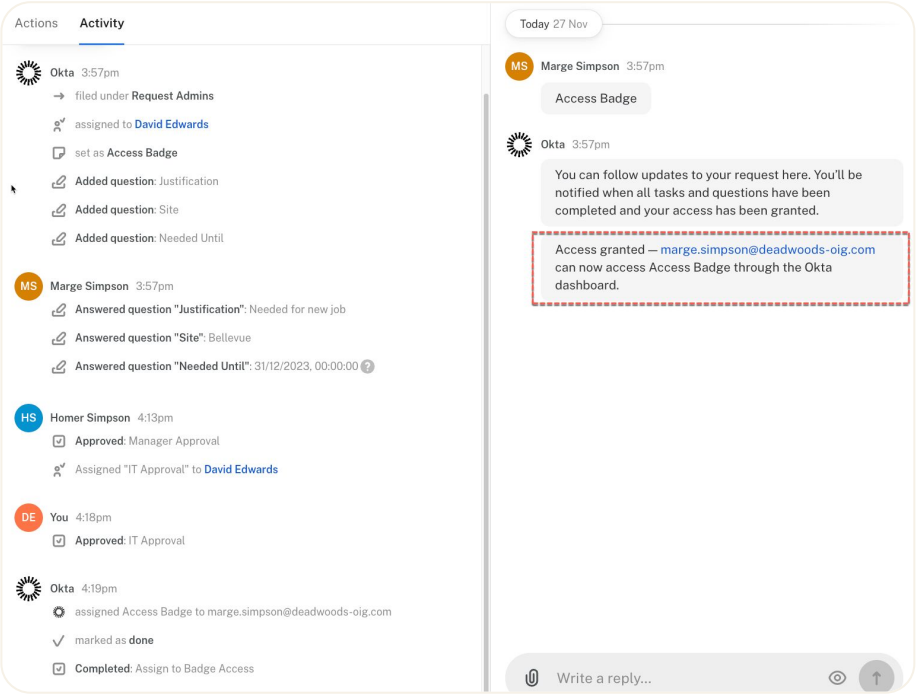
Where there are multiple members of a team or group assigned to a step in a flow, the default behaviour is to assign tasks in a round-robin fashion.

If you do not see the next approval step show up as above, you may need to go review the Request Type and conditional rules in the logic.

Perform IT Approval Step

7. Repeat the review/approval steps for the IT Approval assignee (in my case it's the Okta Admin account).
8. Observe the additions to the view.





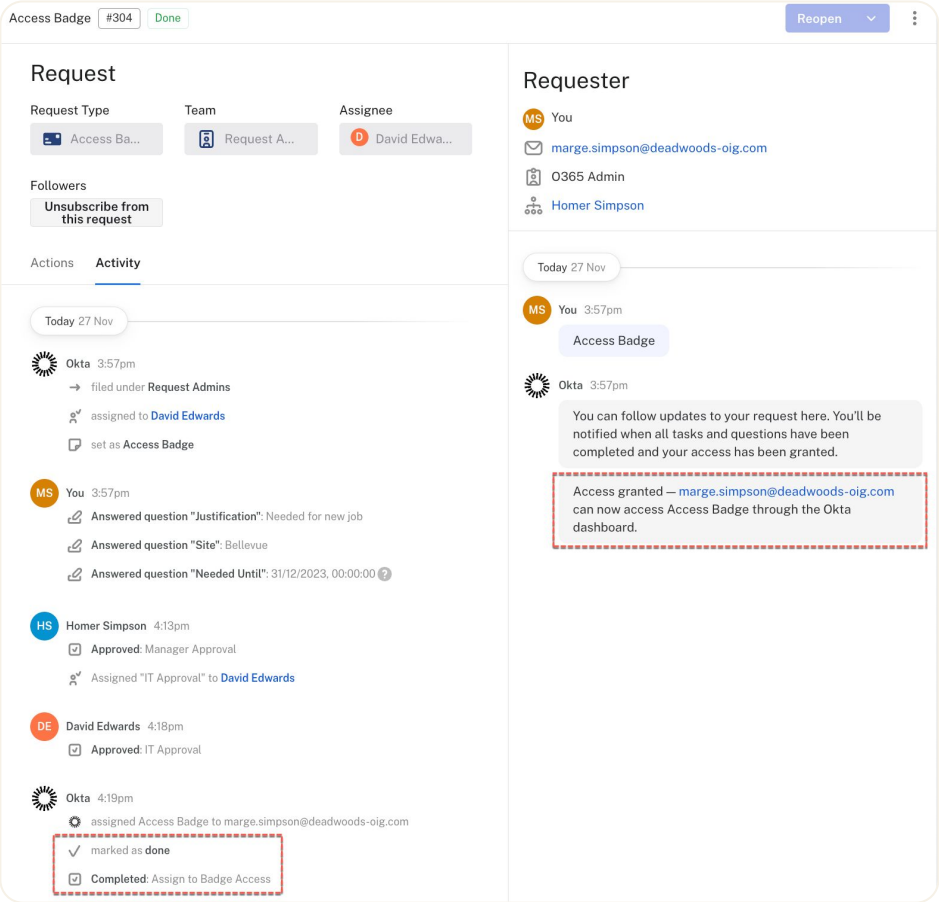
With both levels of approval done, the flow continues to assign the user to the app. Note also the message to the requester to say the access has been granted.

If you do not see the Assign to Badge Access step show up as above, you may need to go review the Request Type and conditional rules in the logic.

This last step will add the user to the requested application in Okta. Before we check in Okta, let's confirm what the requester sees.

Requester Confirmation

1. If not already there, go back into the **Access Requests UI** as the requester (Marge in my example) and the **Activity** tab for the request.



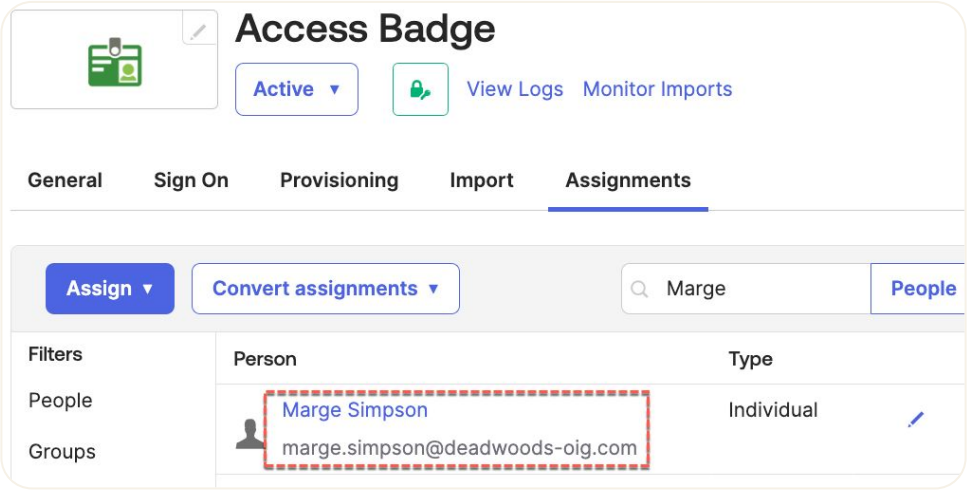
The view is the same as what the last approver saw, showing the entire trail.

The flow has been automatically closed after the last step completed ("marked as **done**"). The requester can re-open the request by typing something in the Write a reply... field at the bottom.

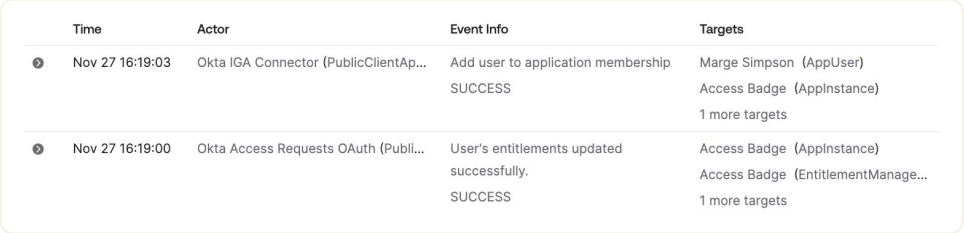
Check the Access Was Assigned in Okta

The last thing to check is that the access was actually assigned in Okta:

1. Log into **Okta** (**Dashboard** then **Admin**) as your administrator
2. Go to **Applications** and find the **Application** you defined in the Access Request flow
3. Check that the requester is assigned to the application



4. Click the **View Logs** link to confirm the **Add user to application** event.



Use an Access Request Flow from Slack

To check the Slack integration, we will repeat the Access Request from within Slack. For this section you will need a user in the group you assigned to the Request Type (Springfield in my case) who has a manager defined in Okta and has access to the Slack workspace you associated with Access Requests. I'd suggest a different user to the previous one so you can see the application access being granted (the example below will use the same user as above).

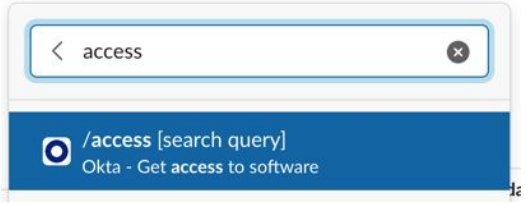
> There are multiple ways of requesting access in Slack and we will look at each of them. Which one you decide to choose is up to you and your deployment – each will result in the same outcome.

Request Access in Slack

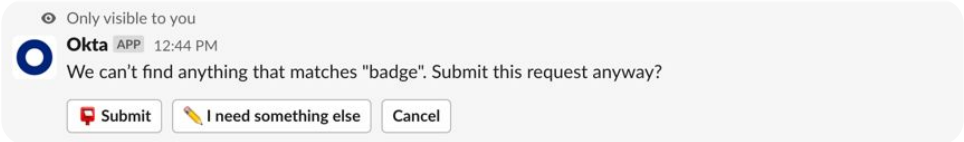
We will look at the different ways access can be requested in Slack. It will be equivalent in Teams.

1. Log into your **Slack workspace** as the user who will request access
2. You will default to a channel based on how your Slack workspace is configured. But in a channel type the following: */access badge*

Note that as you type /access it gives you information about the command

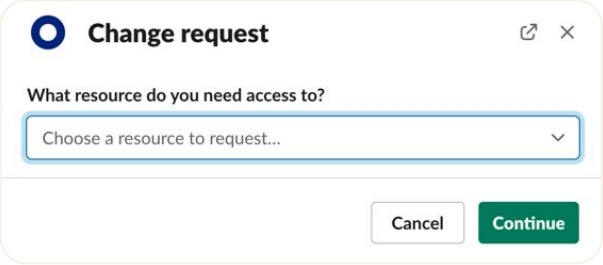


After you hit enter, the Okta (Access Requests) app will analyse your search query against the available flow names. In this case it didn't find a match.

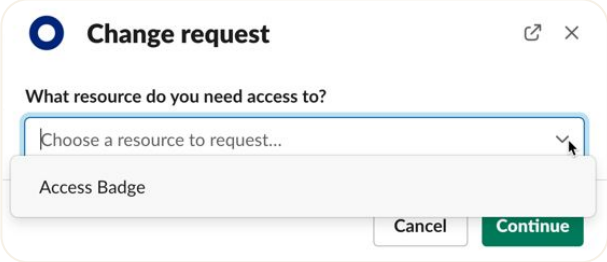


There is some AI behind this processing natural language queries. If it doesn't find a match and you select what you're after it will learn over time. This way, if there's a common term used for something that's not the same as the actual application name, after a while Access Requests will just suggest the real app name.

3. Click the **I need something else** button

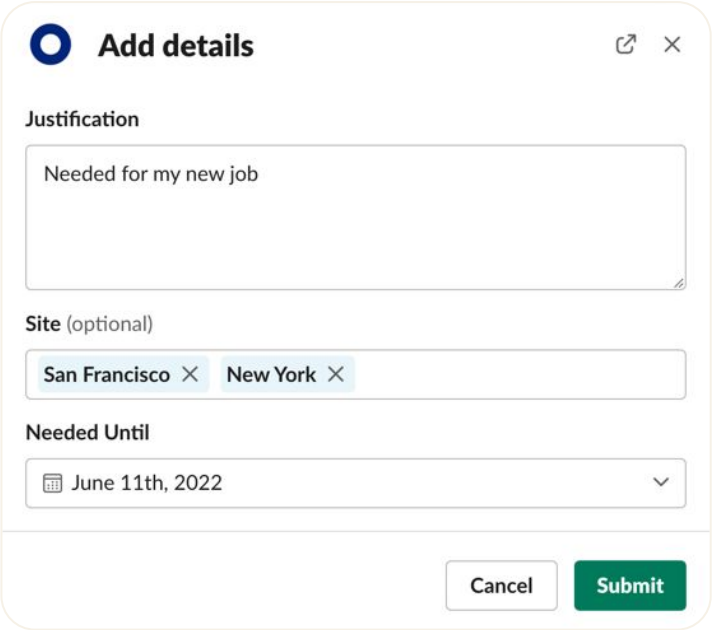


4. Click the pulldown list to select from available Access Request flows



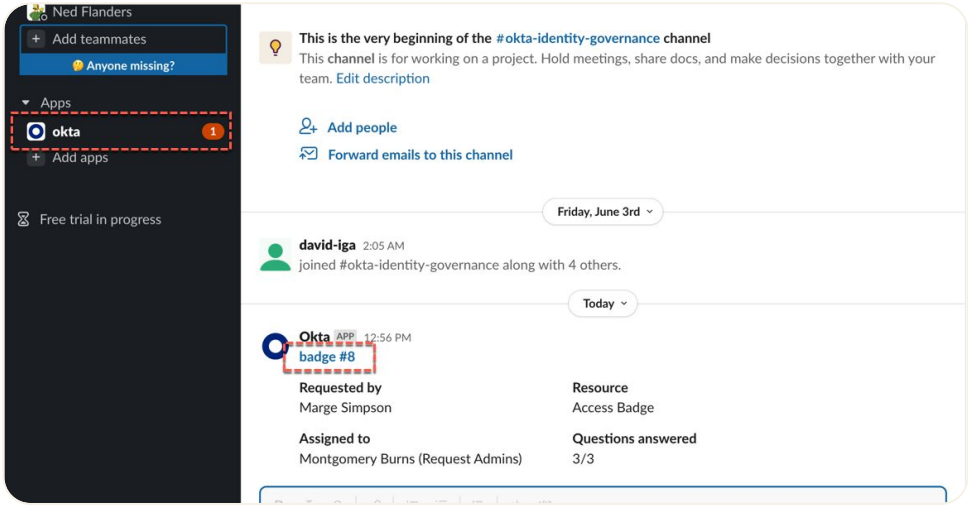
5. The new **Access Badge** request shows. Select it and click the **Continue** button

Notice that the Add details dialog is the same as we saw in the Access Requests UI – the same fields with the same options in the same order.



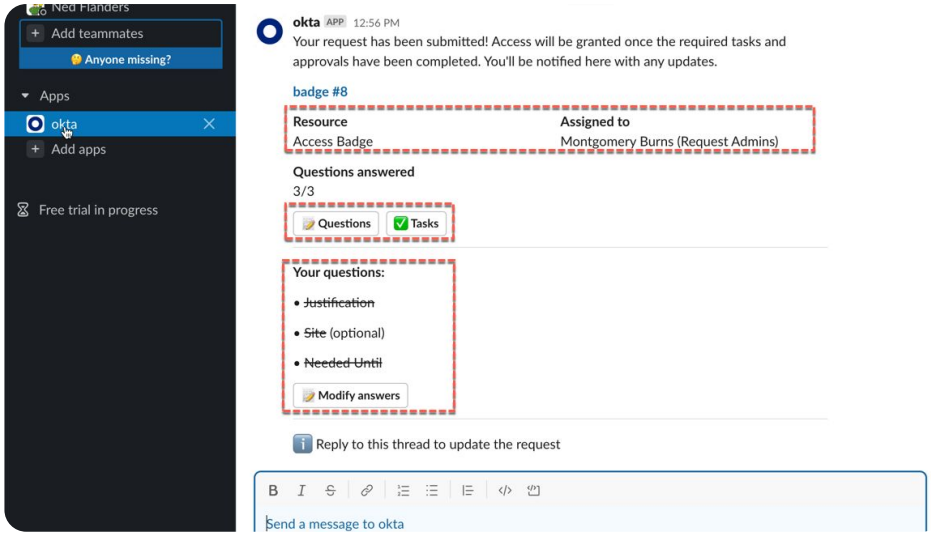
6. Complete the form and click **Submit**

The message in the Slack channel will be updated to provide details of the request. You will also see a message notification on the Okta app.



Each request (irrespective of where it's submitted from) has a unique number. In Slack you will see a message like the "badge #8" above that is a hyperlink. It links off to the Access Request UI, so the user can go work in the web UI if they want.

7. Click on the okta app

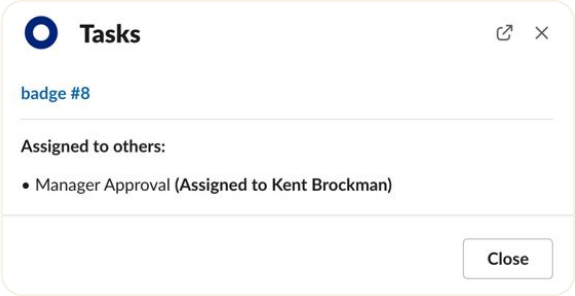


This is providing a complete view of the request – similar to what was shown in the Access Requests UI in the previous example. We can see the resource being requested and who the request is assigned to (with team/group) for managing the request.

We can see that all three questions have been answered. There are buttons to see the provided answers and modify them if required.

There is a button to see the tasks (i.e. next step in the flow).

8. Click the **Tasks** button



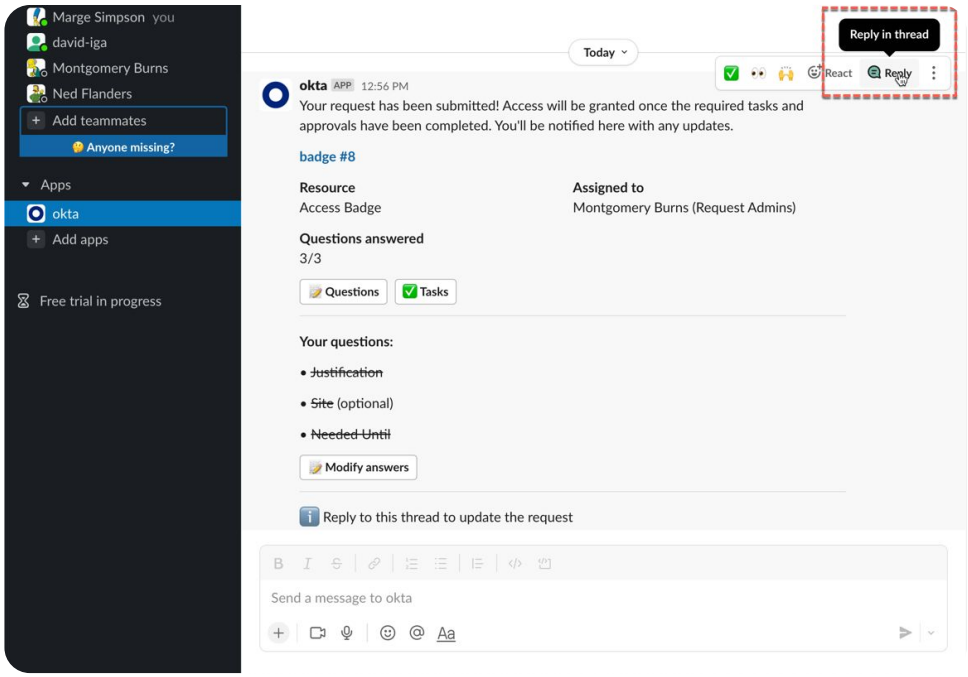
As expected, it is the Manager Approval step and it is assigned to Kent Brockman

9. **Close** this dialog

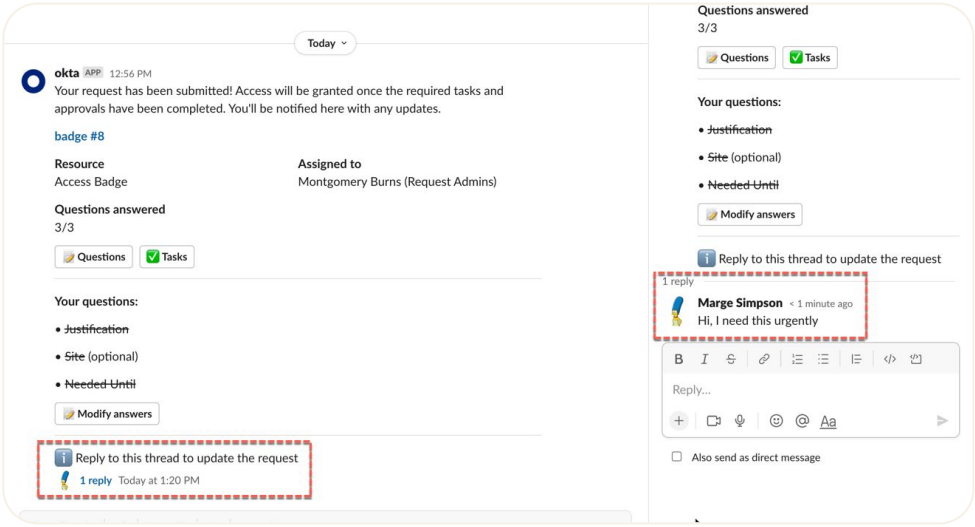
> Note the comment at the bottom saying, “Reply to this thread to update the request”. This is a great way to communicate interactively with the app and users associated with it.

We will add a comment to the request.

10. Hover over the thread and select the Reply in thread function

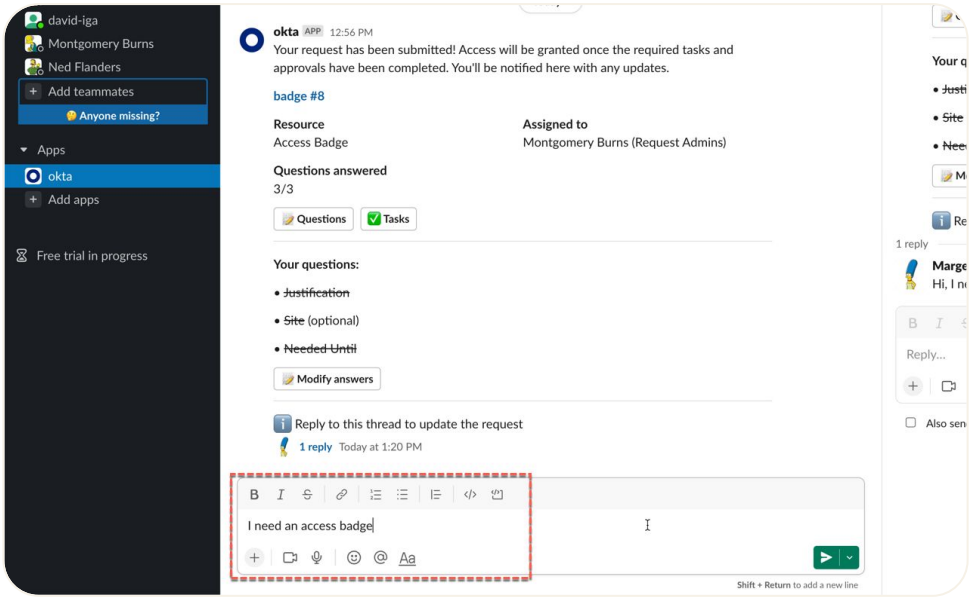


11. In the thread, enter a message like **"Hi, I need this urgently"** and hit enter



Before we perform the access review as the manager, we will look at the two other ways of requesting access. One way is to message the app directly.

12. From within the **okta app** (you should still be there from above) type something like "I need an access badge" into the "Send a message to Okta" box (not replying to the existing thread, adding a new entry to the app).

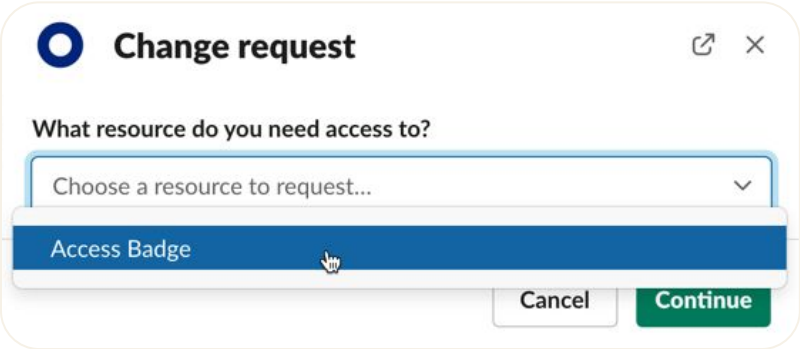


13. Hit enter to send the message

As before it couldn't find an Access Request Type but gives you the option to select from the list of Request Types.

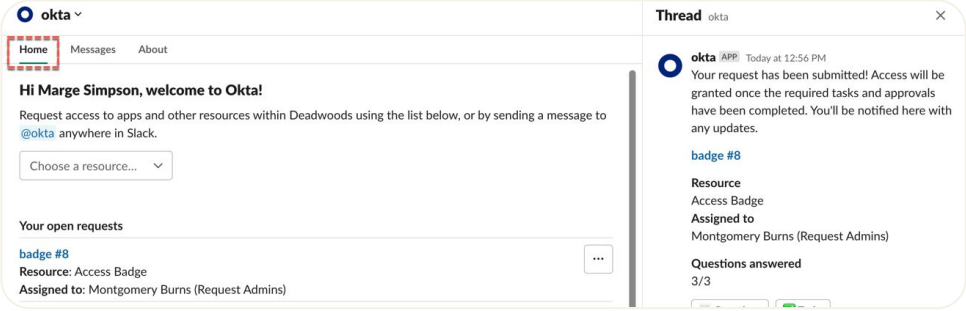
14. Click the **I need something else** button

As before you are presented with a dialog to select from all Access Request flows.

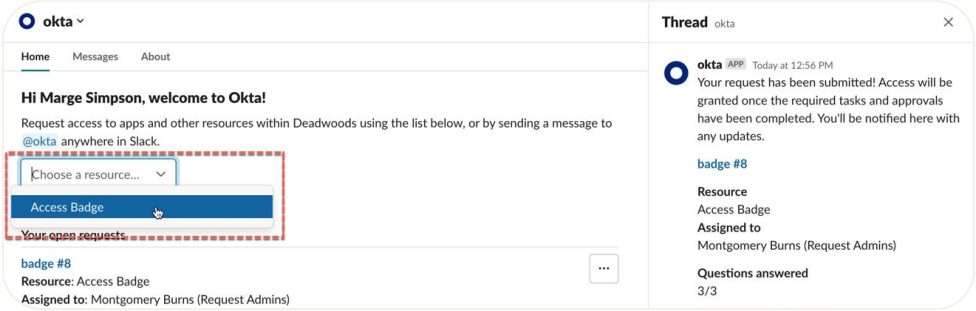


From here the flow is the same as before.

15. **Cancel** out of the **Change request** dialog and **Cancel** the request in the **okta app** (it will disappear)
16. At the top of the page, click on the **Home** tab (you are on Messages)



17. Click the **Choose a resource...** pulldown list



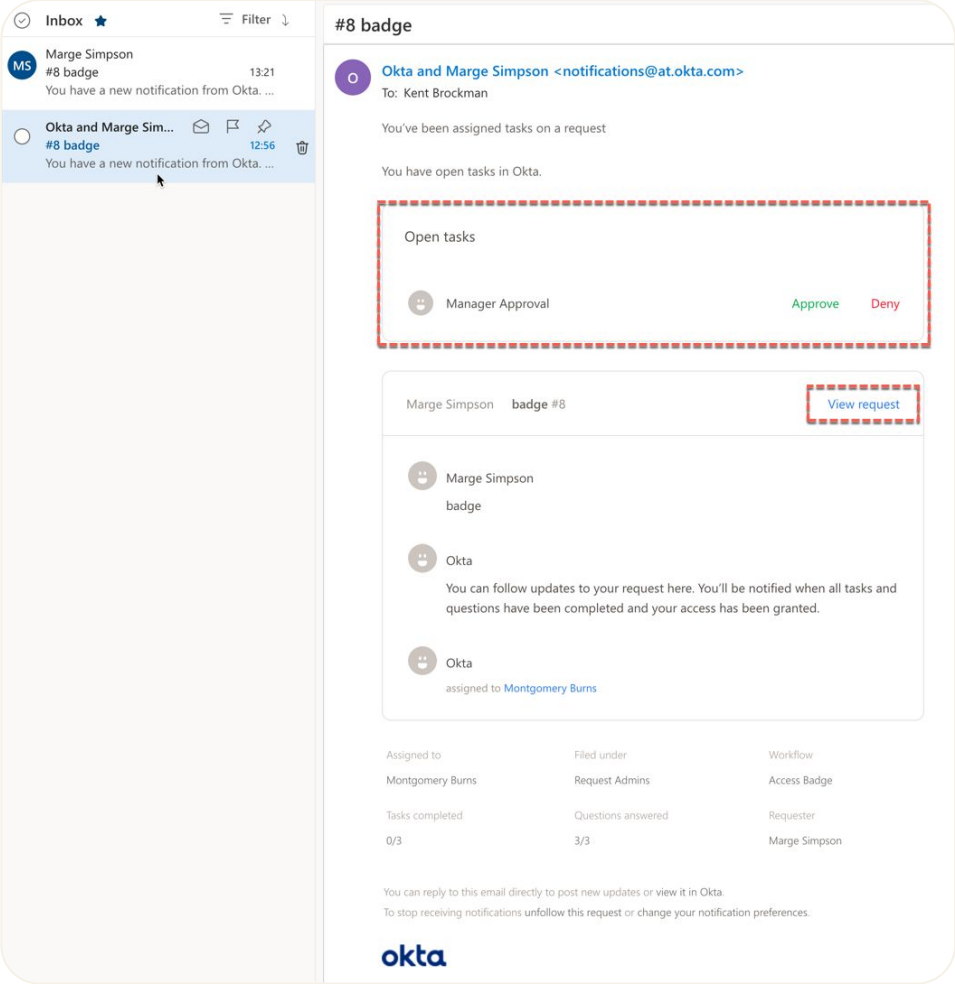
Again, the user is presented with a list of Access Request flows to select from and once selected the flow is the same as before. So, three ways to start an Access Request flow:

1. By typing /access <search argument> in any channel
2. By typing freeform text in the Messages tab of the **okta app**, or
3. By using the pulldown list in the Home tab of the **okta app**

Access Request Review in Email

Irrespective of whether an Access Request flow has been initiated in Slack (or Teams) or the web UI, there are multiple ways that a reviewer, such as a manager, can review and approve/deny the access request. This includes Slack, the Access Requests web UI and email. Let's look at email.

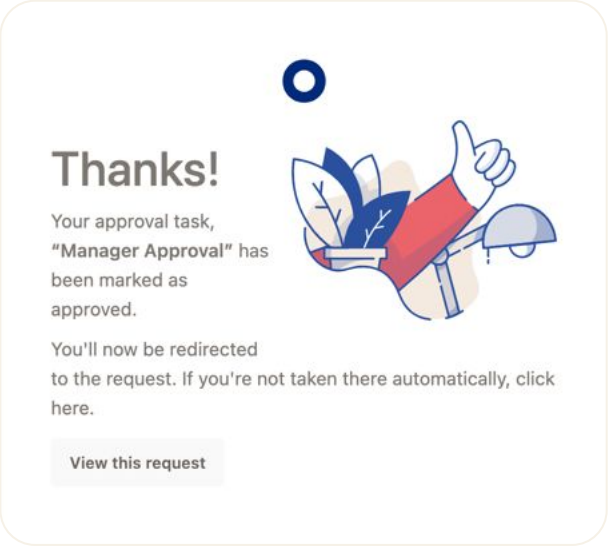
1. Go to your email client for the requesters manager (in my example it's Kent Brockman)
2. Find the email from notifications@at.okta.com relating to the Access Request.



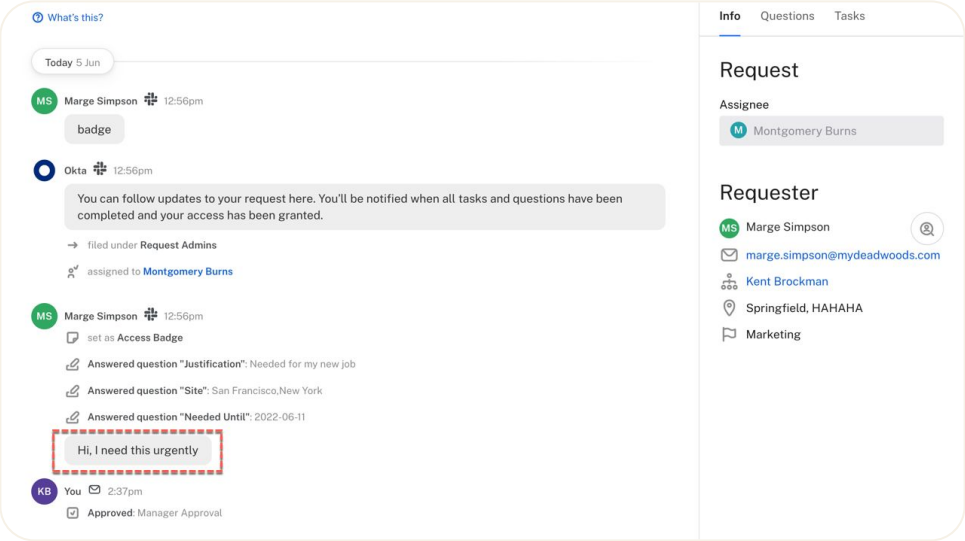
The email provides a summary of the request, with a hyperlink to the request ("View request") in the Access Requests web UI.

It also provides for approving or denying the request in the email.

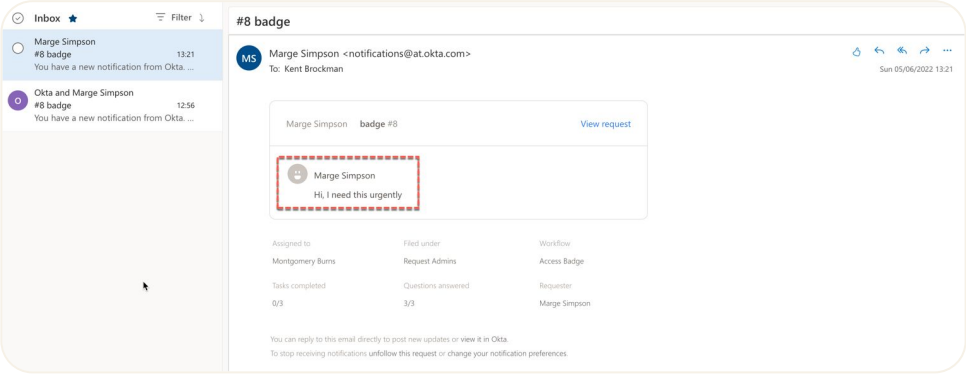
1. Click the **Approve** link in the email and you will be shown a message and taken to the request



Notice that the request includes the message from the requester.



5. Click on the **Tasks** tab to see who the next approver in the flow is (we will need that name next)
6. Go back to your email client and look for any other emails relating to the access request (i.e. the same title, such as "#8 badge"). There will be one for the message the requester sent



The Access Requests platform is using multiple channels to communicate with participants in a Request Type – the web UI, email and Slack/Teams if configured. Initiating a request in one channel doesn't mean subsequent steps need to use the same channel. This is very flexible and allows users to work in the way they prefer.

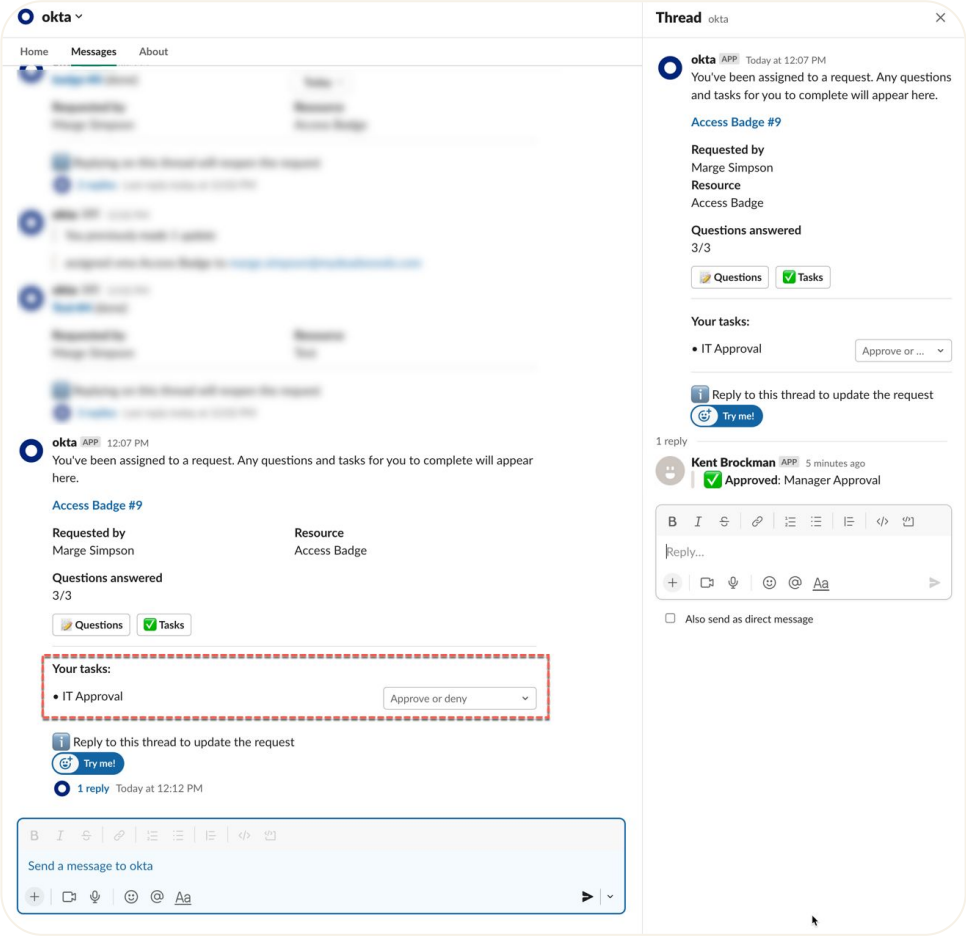
Review Access in Slack

For the second level approval we will show the approval mechanism in Slack.

1. Log into **Slack** as the second-level reviewer you identified earlier
2. You should see a notification against the **okta app**
3. Go to the **okta app** and the **Messages** tab and find the request
4. Click the **1 reply** link to open the message in the **Thread** view on the right

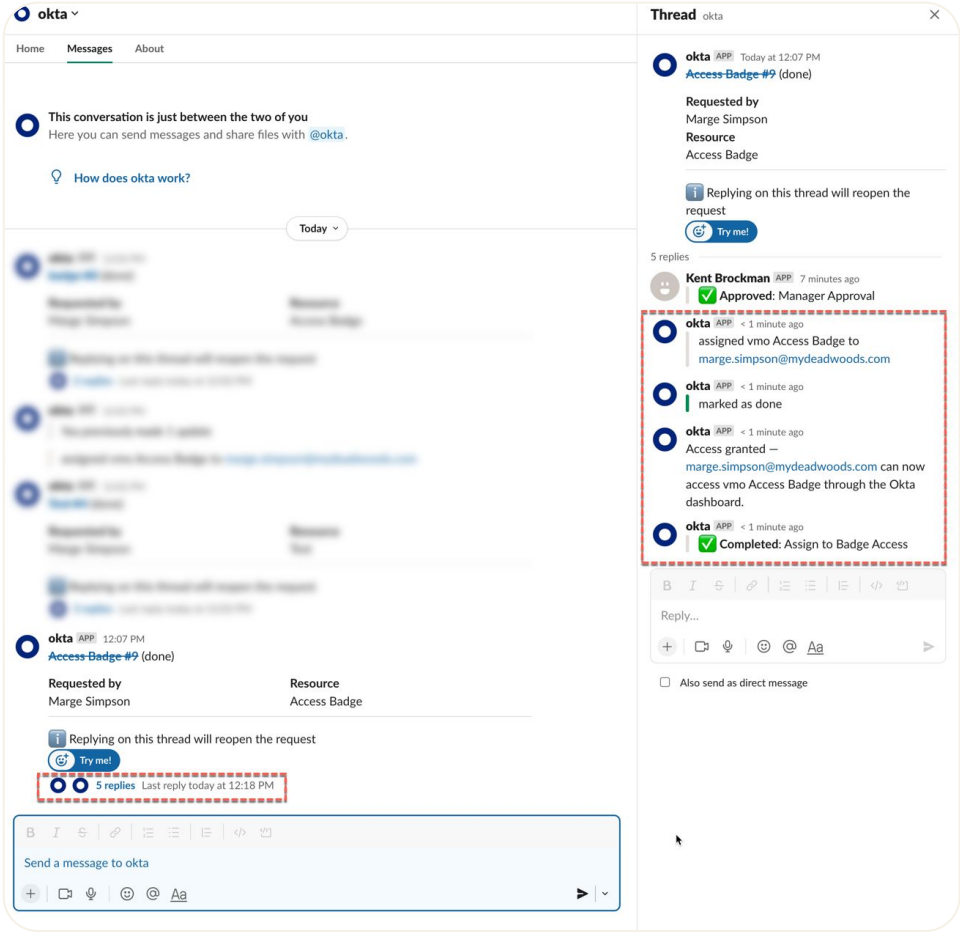
You will see the same details of the request as you would in the other channels. You can look at the questions and tasks.

There is an option to **Approve or deny** the request.



5. Click the **Approve or deny** button and select **Approve**

After a pause, the request display in Slack will change and you should see five replies in the thread view now. These show the assignment of the application, marking the request as done, an access granted message and a completion message from the okta app.



This completes the flow (you could go check that the access has been actually assigned in Okta, but as it's the same flow that was proven earlier, the outcome should be the same).

Create an Access Request Flow with Sublists and Timers

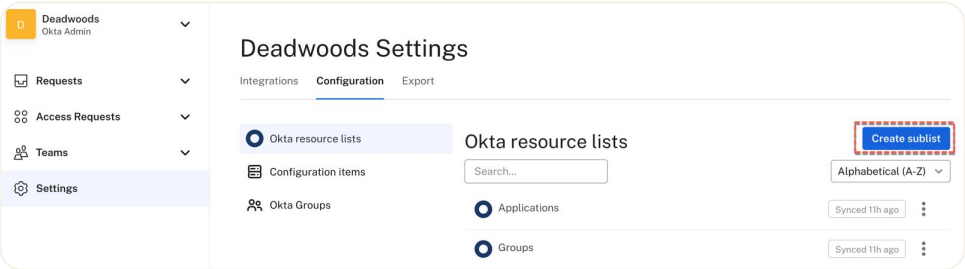
Two additional features in Access Requests are sublists and timers. Sublists are defined subsets of either Applications or Groups lists and can be used to allow users to select from a short list of apps/groups. Timers allow for a delay to be introduced into a flow (see https://iamse.blog/2022/07/27/oig-access-requests-using-the-new-timer-feature/ for more information and examples).

In this section we will build another Request Type using both of these. It will present a list of groups for a user to select one from and then add the user to that group but then automatically remove the user from that group on a set date. The notes will be briefer than above when covering the same steps as done earlier.

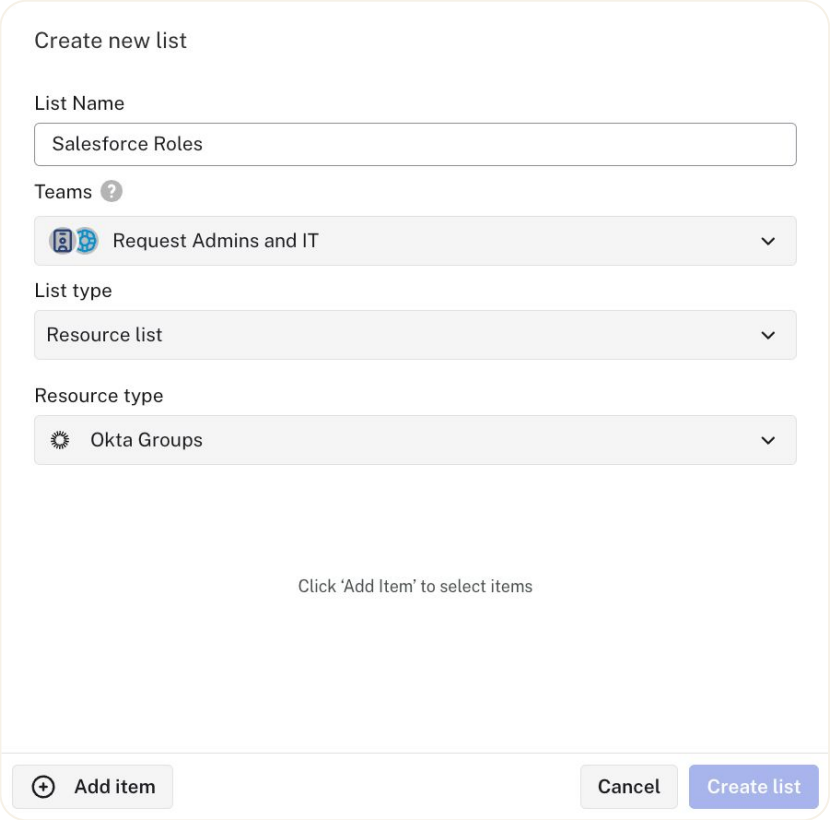
Create a Sublist

The first step is to create a sublist of groups. You could use any groups, but I will use some groups I set up for Salesforce roles. The steps are similar to creating the custom Configuration list you created earlier:

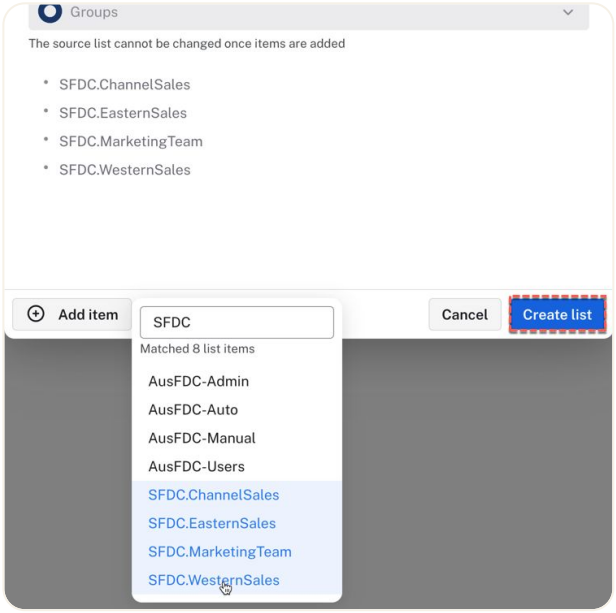
1. Go into the **Access Requests** console as your administrator and access the Settings menu item
2. Go to the **Configuration** tab and **Okta resource lists** item
3. Click on the **Create sublist** button



4. On the **Create new sublist** dialog, give the list a name
5. Assign it to one or more **Teams** (so team members can use the sublist)
6. In the List type pulldown, select **Resource list**
7. In the **Resource type** pulldown select **Okta Groups**



8. Click the **Add item** button to select a specific group to add
9. Select the groups you want to add and watch them appear in the list



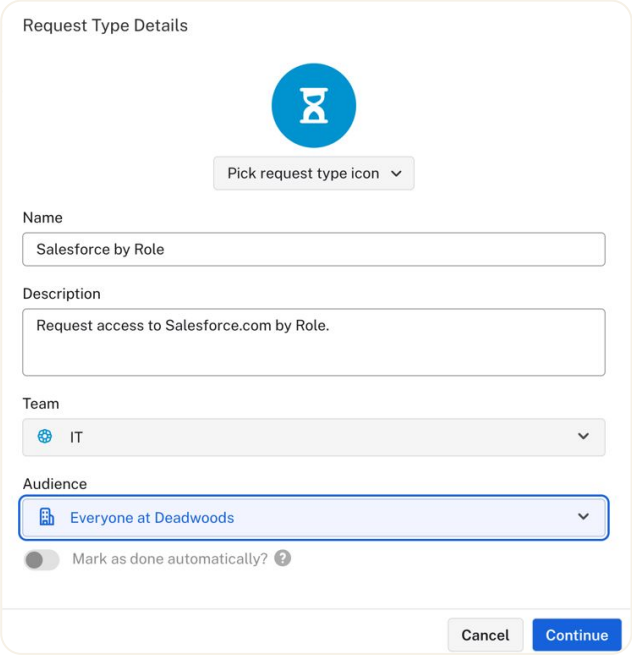
10. Click the **Create list** button when done

The sublist is now ready for use.

Create a New Request Type

Next, we will create a new request flow to use the sublist (and the timer feature).

1. Go into the **Access Requests** menu item and click the **Create request type** button
2. Give the new Request Type a **name**, optionally a **description**, select your **team** and **audience** (this is all as was done earlier)



3. Click the **Continue** button when done

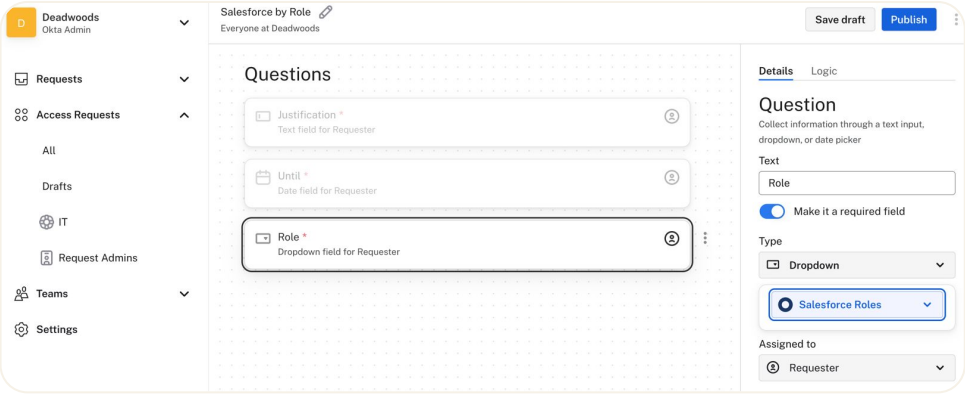
Add Questions to the Flow

We will add three questions, the justification, an end date and the group selection. The first two are as done in the earlier flow above. I've added the questions in that order to simplify the instructions but you could present them in any order.

1. Add a **Justification text** question to the flow
2. Add an **Until date** question to the flow
3. Add a **Role dropdown** question to the flow, selecting the new sublist you created above from the **Configuration items** list (do not select Multi-select?)

All are required and assigned to the Requester.

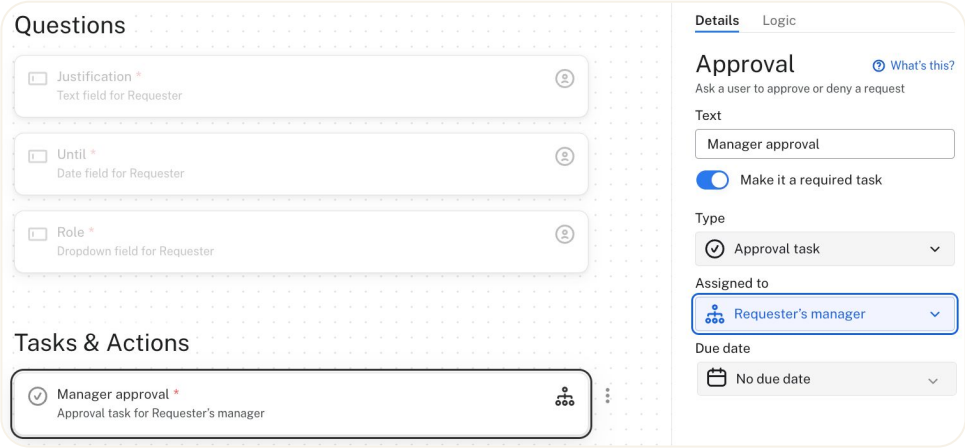
Did you know you can assign questions to other people, such as managers approving a request? It's something to try out when you get the chance.



Add an Approval to the Flow

We will add a single approval level for this, but you could add more if you want.

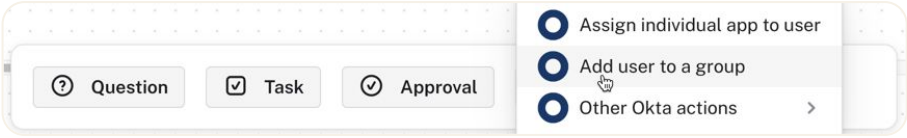
7. As earlier, add a Manager approval step to the flow, assigned to the **Requester's manager**



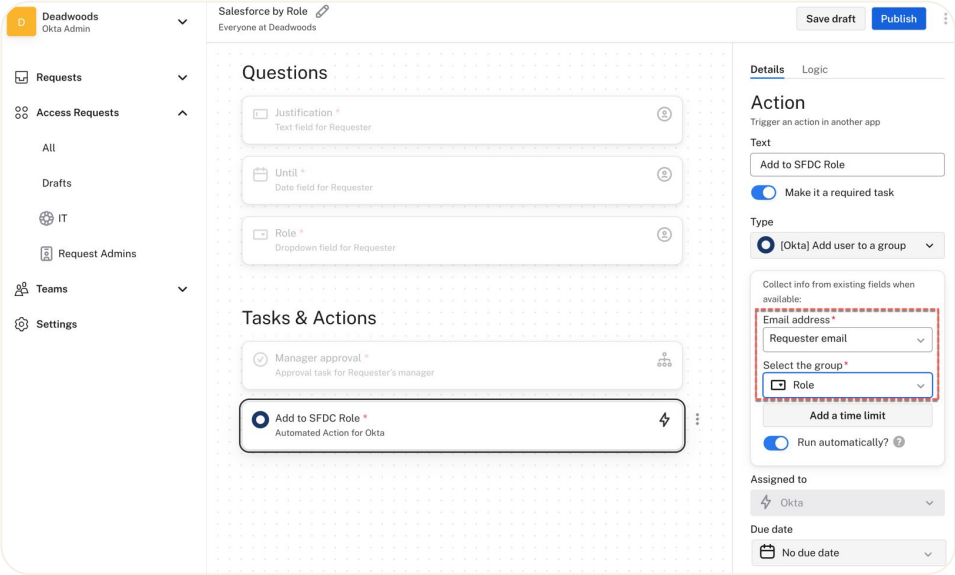
Add the Group Add Action

In the earlier flow we assigned the requester to an application. In this example we will add the user to the selected group.

8. Select the **Action** button and select the **Add user to a group** action



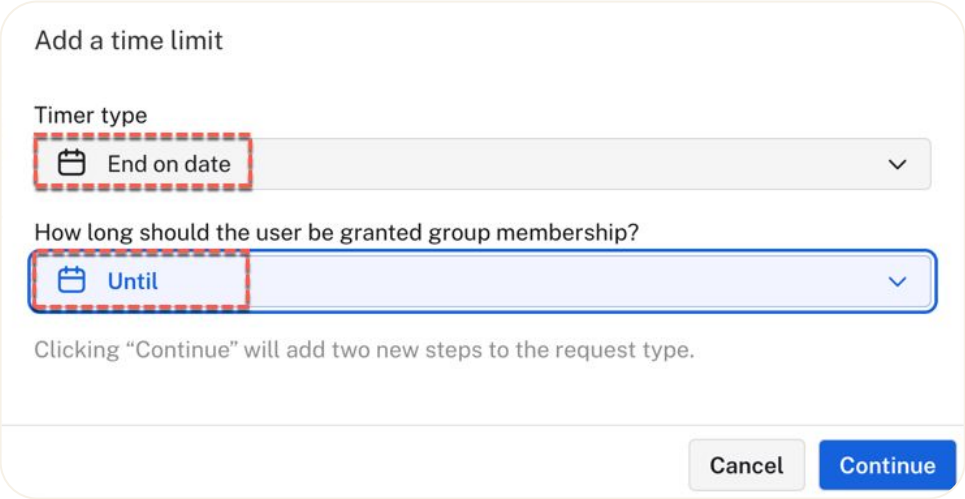
9. Give the action a name and enable the **Run automatically** option
10. Set the **Email address** to the **Requester email**, and set the **Select the group** to be the **Role** pulldown you created earlier



This will tell the flow to add the user to the group that was selected in the Role question.

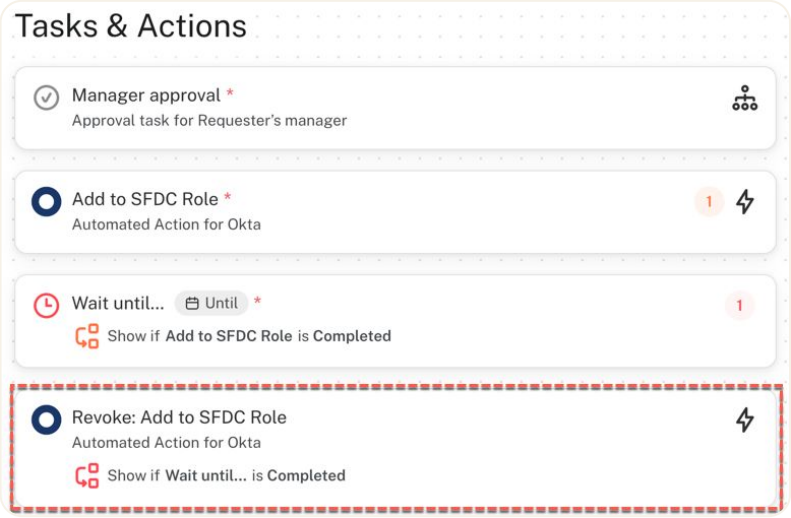
11. Click the **Add a time limit** button
12. On the **Add a time limit** dialog, select the **End on a date** option for **Timer type**, and select the Until option you specified for the earlier question

This will tell the timer to use the date selected by the user earlier. Note that you could also set a duration (like two hours).

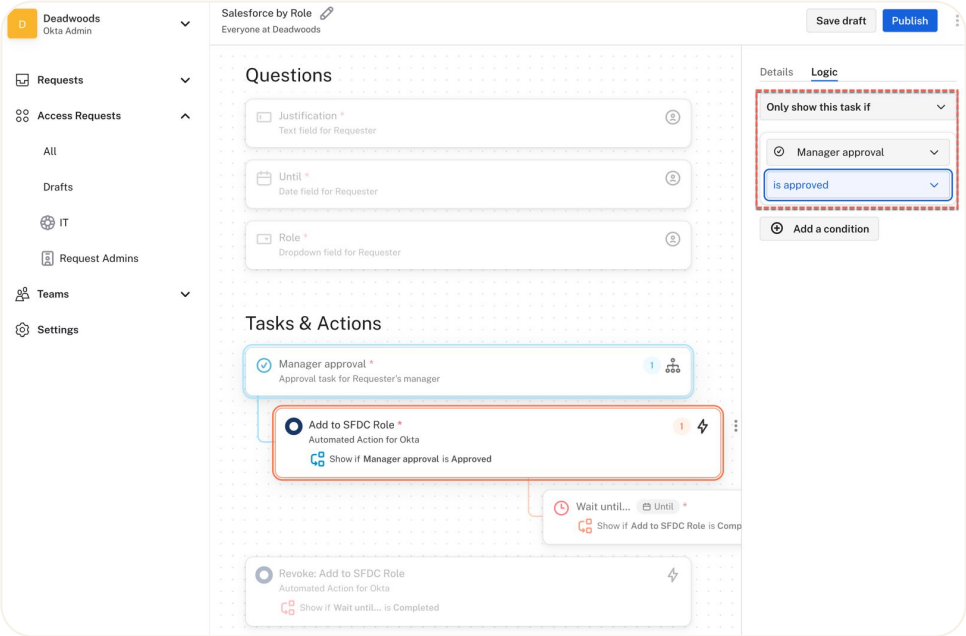


13. Click the **Continue** buttons

Note that Access Requests has automatically added a Revoke: Add to SFDC Role action that will remove the user from the group.



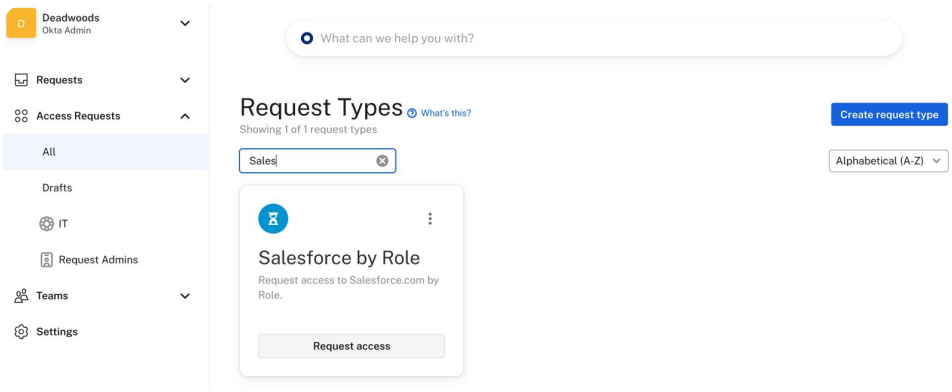
14. Click on the **Wait until** card and notice it is tied to the previous Okta group add action and will wait until the defined date
15. Click on the **Revoke: ***** card and notice it is tied to the **Wait until** card and that it's running the **[Okta] Remove user from a group** action and will remove the user from the Role selected at request time
16. Click on the **Add user to group** action you created just now and set the **Logic** so it only runs if the **Manager approval** step to completes with **is approved**



Finish and Publish the New Request Type

The last step is to clean up the flow and publish it.

17. Edit the Request Type (pencil icon beside name) and set **Mark as done automatically**
18. Click the **Continue** button
19. Click the **Publish** button
20. Go to the **Access Requests > All** (or your Team) view to check the new Request Type is there

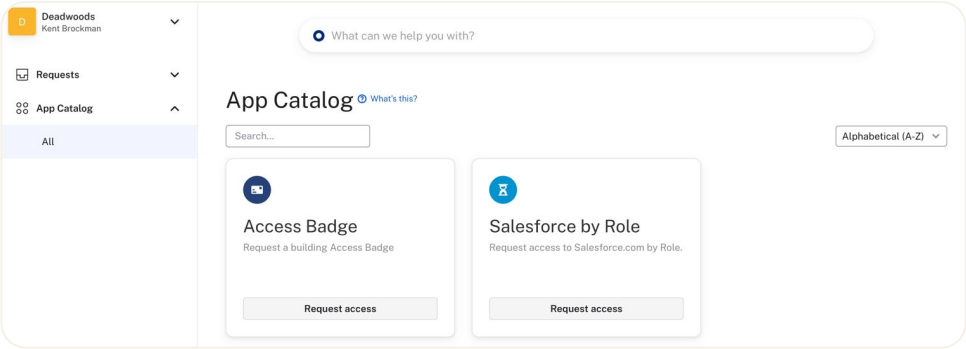


Next we will test it.

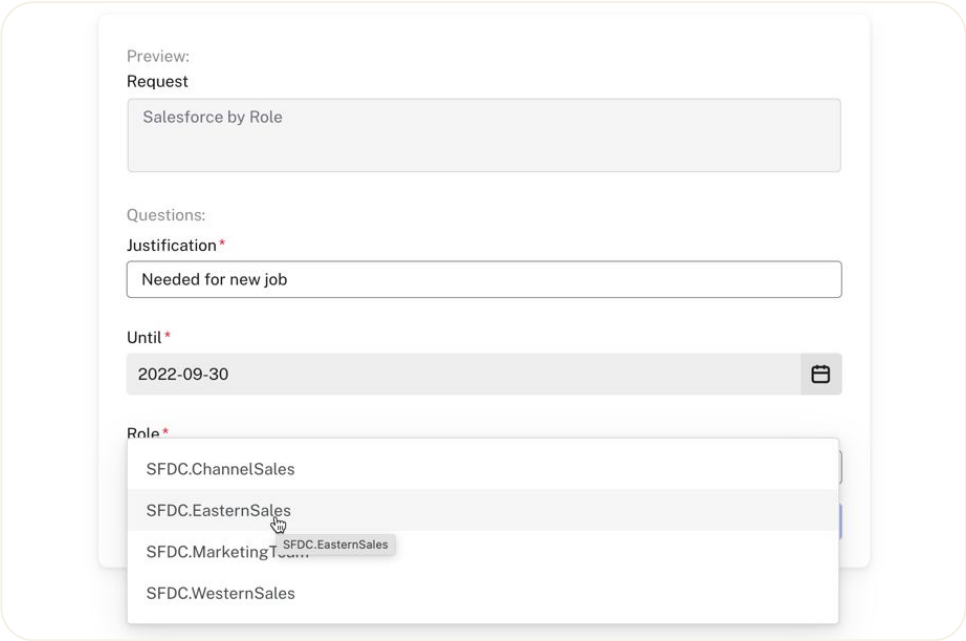
Test the New Request Type

We will test the new Request Type

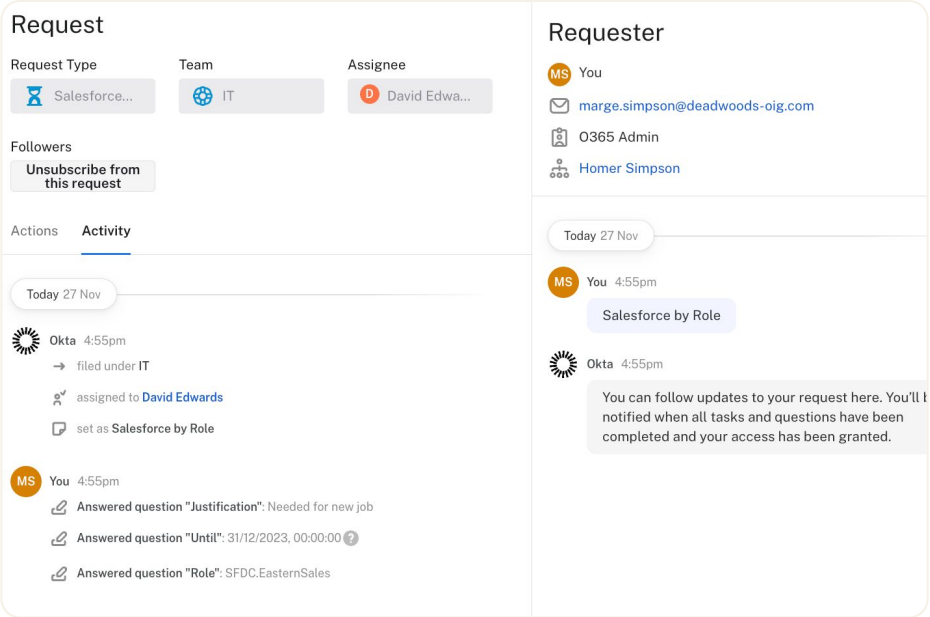
1. Log into the Okta dashboard for a user that should be in the scope of the request flow (I set mine to Everyone) and go to the **Okta Access Requests** app
2. You should see the new Request Type there in the **App Catalog**



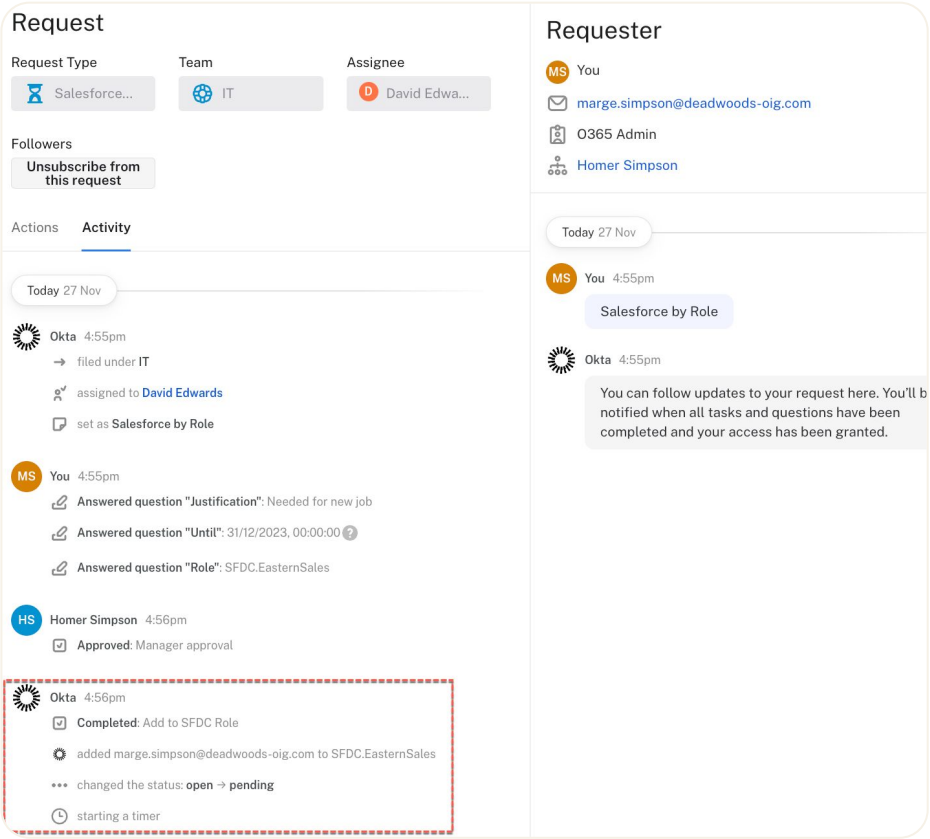
3. Click the **Request access** button for your new flow
4. You should see the three questions you added (Justification text field, Until date field and Role pulldown). Enter/select values for each.



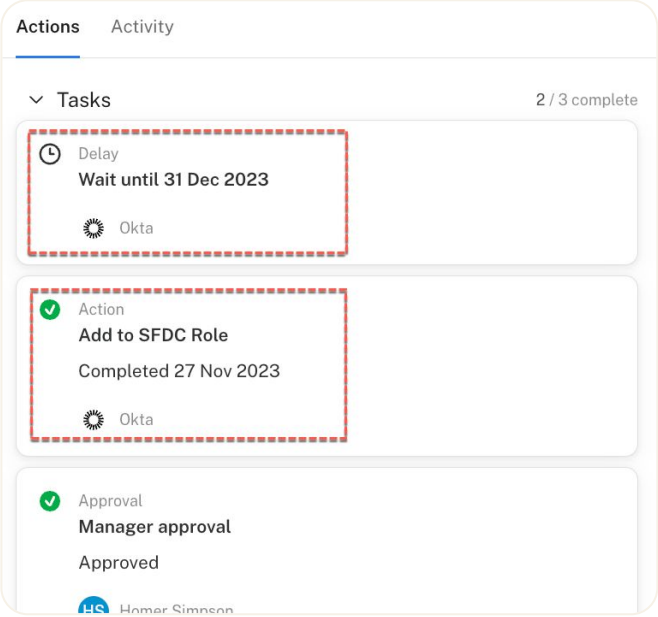
5. Click the **Submit new request** button and look at the **Activity** tab



6. As the user's manager, go approve the request (see earlier in the document for an example of this).
7. Once the request is approved return to the requesters view and check the messages in the **Activity** tab.



8. Go to the **Actions** tab and and look at the **Tasks**



You should see that the user has been assigned to the group the requester selected (you could check this in Okta) and a timer triggered to wait until the date the requester selected. We won't wait for the timer to expire (you can leave it in your system and come back to it).

This completes this section of the document looking at sublists and timers.

Summary of Getting Started with Access Requests

This also completes the guided steps around access requests. In the next section we will look at Access Certification.

In this section we have walked through the Access Requests components, basic configuration and how to build and use Request flows (including interaction with the Web UI, email and Slack). This is just a basic introduction to get you understanding the capability, you can now start looking at more advanced Access Request Types.

Exploring OIG Access Certification

The next capability we will explore is Access Certification. Access Certification (aka attestation, recertification etc.) is the mechanism to validate that a user still needs the access they have. It is a common control in compliance regulations. Certification campaigns may be run periodically, or there may be continuous certification when user roles change. Okta Identity Governance currently supports user-group memberships and user-application assignments. Participants in a campaign (such as the users manager) will approve (access retained) or revoke access (access should be removed).

This section of the document will explore the Access Certification mechanism in Okta Identity Governance by creating a campaign, launching it, and having a manager review access.

Whilst not a focus of the guided walkthrough in this document, prior to building and running certification campaigns, you need to consider how access is granted in your Okta. Users can be assigned directly to applications, but this is considered bad practice. Preferably groups will be assigned to applications (with groups perhaps representing roles) and users are added to, and removed from, the groups as needed. Also, how are users assigned to groups? Are they Okta managed groups or application-managed (like AD)? Are they manually assigned (e.g. via the Admin console, Workflows, Access Requests or API calls) or are there group rules automatically assigning group membership? Understanding this for your environment will be crucial to effective Access Certification campaigns – revoking access in a campaign may have unexpected results if Okta is not setup to allow removing the group membership.

The documentation to support this can be found at: https://help.okta.com/en-us/Content/Topics/identity-governance/access-certification/iga-access-cert.htm.

Creating an Access Certification Campaign

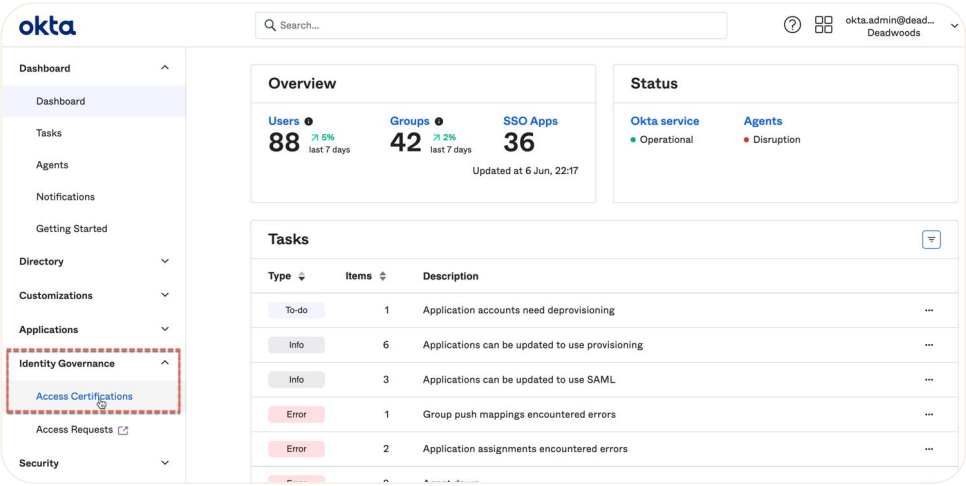
Access certification is built into the Okta platform. There is an administrative interface to create and manage campaigns, and an end-user interface for participating in campaigns.

We will create a campaign for resources in your environment. The example below will review application assignment, where the application is the one that was used in the Access Request section earlier. You can use any application in your environment but recall that the users should have valid managers.

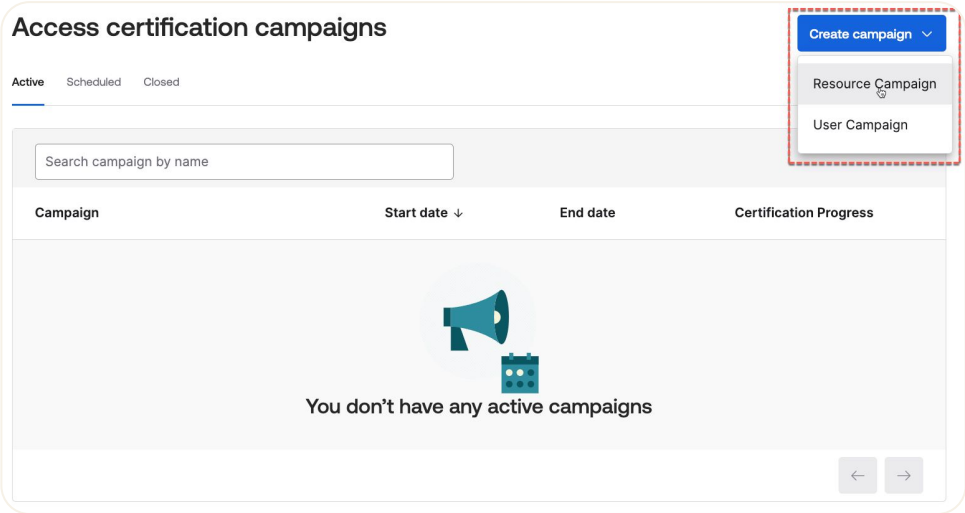
Create a New Campaign

To create a new campaign:

1. Login into **Okta** as an administrator and go to the **Admin** console
2. Find the new **Identity Governance** menu item and expand it
3. Click on **Access Certifications**



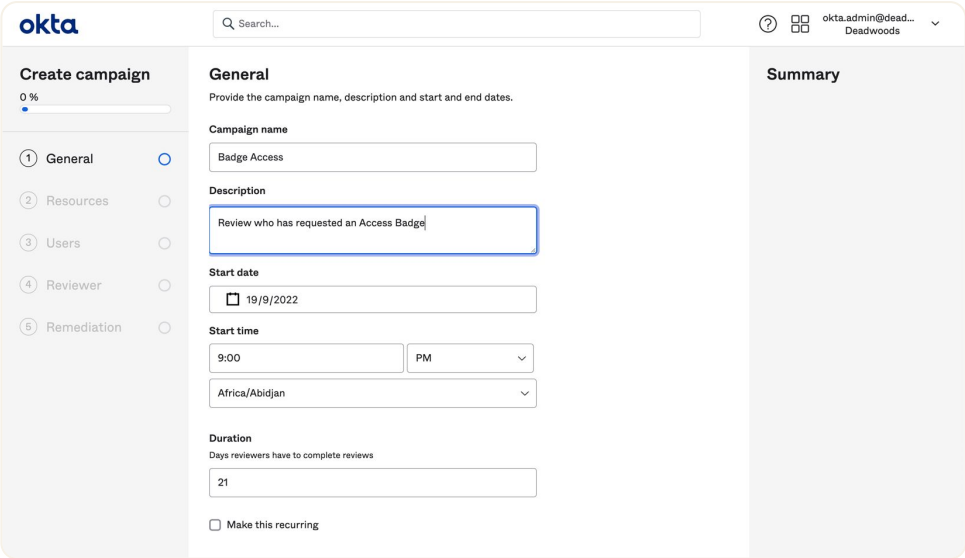
4. Click **Create Campaign**, then select **Resource Campaign** to start the wizard



There are two types of access certification campaigns, Resource Campaigns and User Campaigns. Resource campaigns focus on users by resource (application(s) or group(s)) whereas User campaigns focus on resources assigned to one or more users - i.e. user-centric. We will walk through a Resource campaign in the following sections, but creation, execution and participation in a User campaign is the same. With resource campaigns, you can also review entitlements is Entitlement Management is enabled for applications. We will looks at a Resource campaign with entitlements in the Entitlements Management section of this guide.

General Settings

The wizard will walk through five pages of settings to configure the campaign, the first being the General page.



The campaign creation progress is shown on the left and the right pane will build up a summary of the campaign as you progress through the wizard.

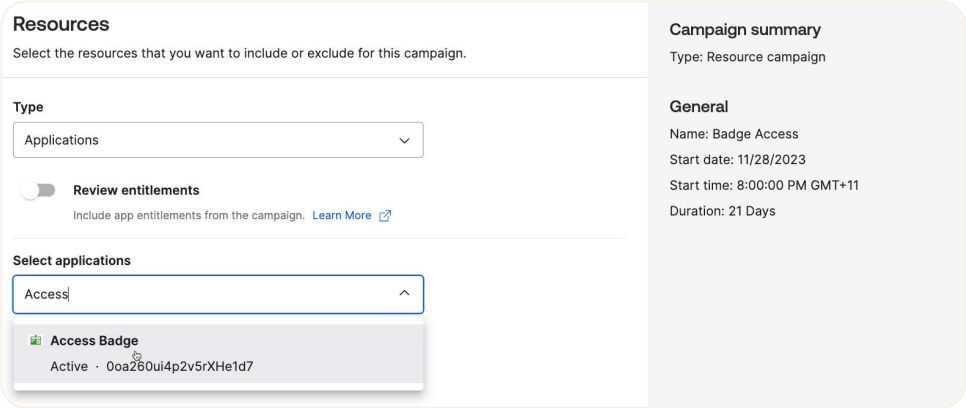
On this page you can set the campaign to start on a specific date and time with timezone settings, and the duration. You can also make this a recurring campaign so it will automatically start periodically (if you select Make this recurring, it will expand the dialog to include settings for the interval).

6. Click the **Next** button

Resources Settings

The Resources page is where you decide what you are certifying – group membership or application assignment. You can only select one type or another, but you can select multiple of each.

7. Select **Type** of **Applications**
8. Select one or more applications in your environment



As you type the application name, it will offer matching applications. You could also select groups - but you can't do both applications and groups in the same campaign.

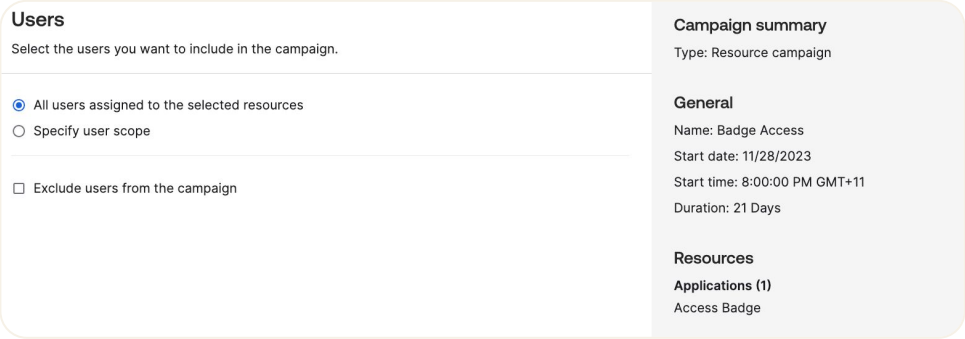
Also, there is the Review entitlements option for Entitlement Management-enabled applications. We will explore this later in this guide.

9. Click the **Next** button

Users Settings

On the Users page, you specify the scope of the users in the campaign. Are all users assigned to an application to be reviewed, or only some? Do you need to exclude specific users for some reason?

10. Leave the **All users assigned to the resource** option selected



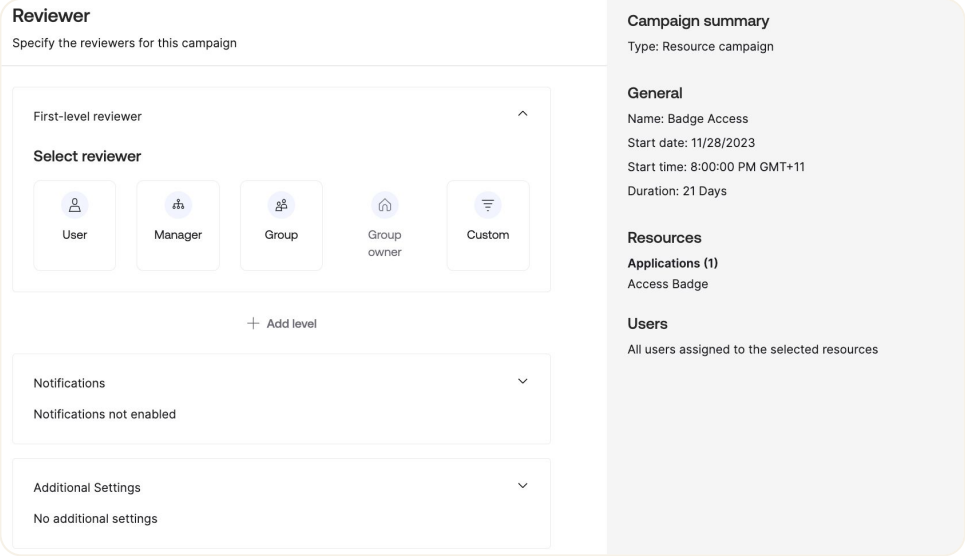
If you choose to **Specify user scope**, you can use the Okta Expression Language to filter in users based on attributes stored in Universal Directory. We won't cover this in this guide.

You can also explicitly **exclude users from the campaign**. Selecting the options exposes a field to enter specific Okta users to exclude.

11. Click the **Next** button

Reviewers Settings

The reviewers are the Okta users who will review the access.

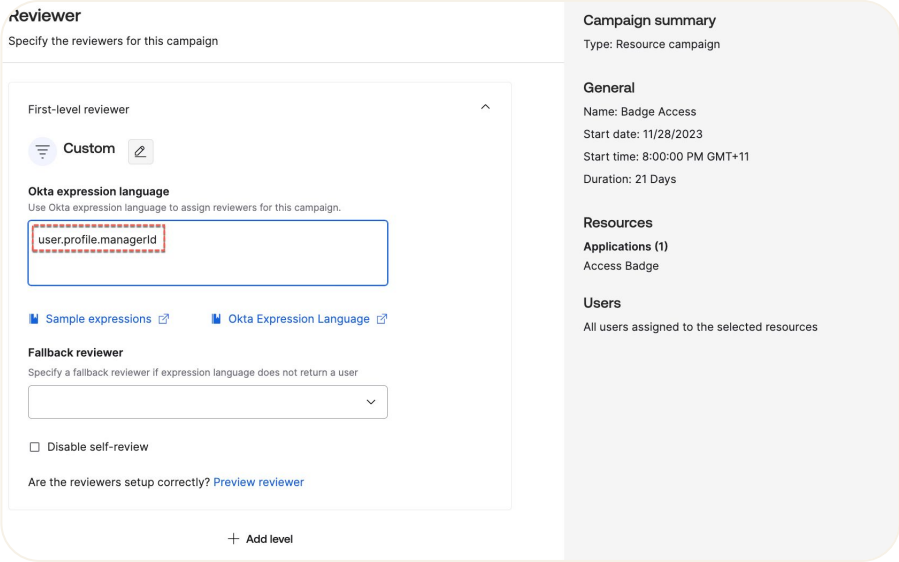


There could be:

- A single (static) reviewer specified as an Okta User,
- The users **Manager** which is the user defined in the managerId field on the users Okta profile.
- A **Group** of users, which is an Okta group and could represent all people in a help desk team or a security admin team.
- If you are reviewing Group access (i.e. a Resource campaign with a resource type of Group) you can use the new **Group Owner** setting on groups (i.e. users or groups that are responsible for that group).
- **Custom** allows selection of the reviewer by using Okta Expression Language. If you had another field on the user profile you wanted to use, or you wanted to check a certain field and assign a reviewer based on the field values, you can use expression language. We will do this in the following example (even though it is setting the manager value which there's an option for - we are just doing this to show how it can be done).

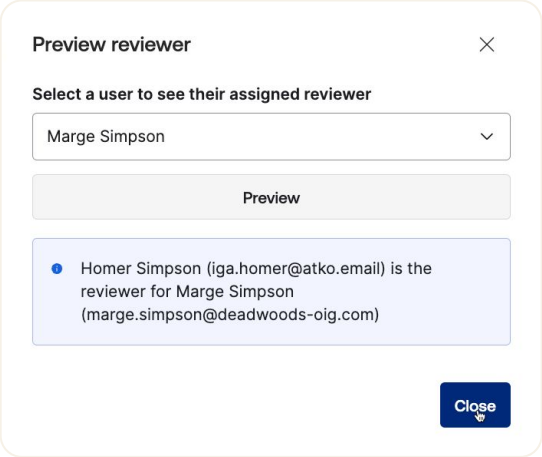
You can also have multiple levels of reviewers (the + Add level option). We will only set a single level.

12. Click on the **Custom** option. The page changes to show an **Assign reviewer(s) for this campaign using Okta Expression Language** and a field to enter some code, plus some other fields.
13. Click on the **Sample expression** link below
14. In the new browser tab, have a look at some of the example expressions that can be used to dynamically determine the reviewer
15. Copy the **user.profile.managerId** sample expression
16. Go back to the campaign tab and paste that into the field



17. In the **Fallback Reviewer** field, enter a valid user (such as the system administrator) to be a fallback reviewer for any users that don't have a valid manager
    Note the Disable self-review option. It is used to stop someone reviewing themself if the evaluation of reviewer returns themself.

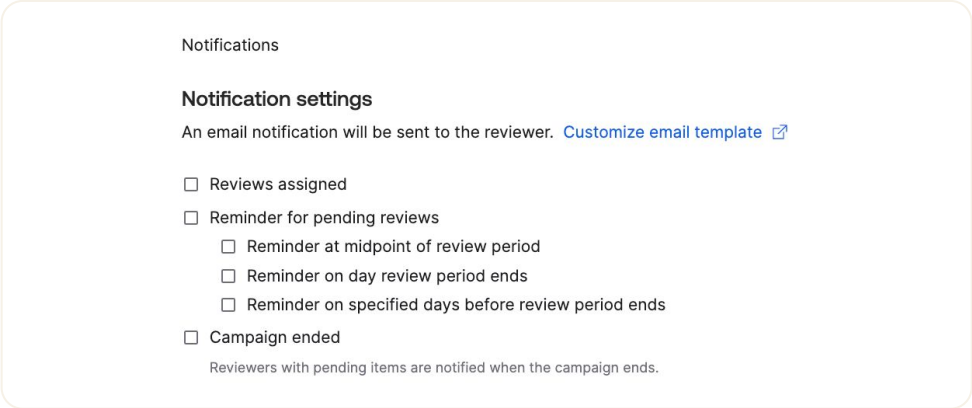
18. Click the **Preview reviewer** link, enter a user you know has a valid manager assigned and click the **Preview** button



If you see the correct manager specified above the field, then the expression is correct.

19. **Close** the **Preview reviewer** dialog

20. Click the down arrow in the **Notifications** section. We will not enable any of them, but you have the option to send emails to reviewers at campaign launch, various times during a campaign execution, and on campaign end. Feel free to enable them and monitor the email inboxes, but we won't include it in the steps below.



The Additional Settings section is not used for this type of campaign.

21. Click the **Next** button

Remediation Settings

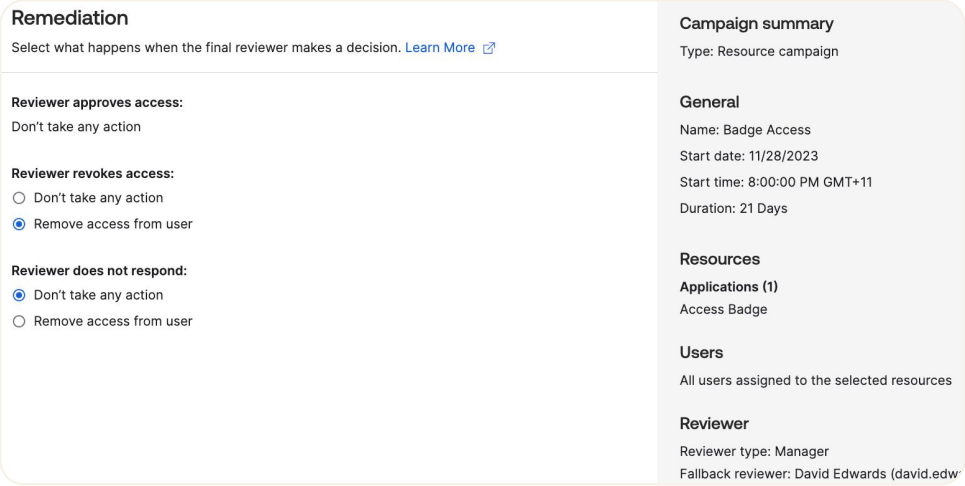
The remediation settings are the most important page – it's the reason for the campaign. Every item (i.e. group membership or application assignment) has three options: approve, revoke and reassign.

If access is approved, the access is retained, and the decision is logged for audit purposes. If access is revoked, that's indicating that the access should be removed. It can be flagged to be removed (i.e. don't take any action) or removed. There may be implications on this depending on how the access was granted. Finally, a reviewer can reassign the review to someone else.

> If you mark a revoke to remove the user from the resource, and Okta can't do it due to internal policy, it will be flagged for manual remediation in the campaign results.

You also need to decide what is to occur if there are unreviewed items at the end of the campaign – should they be removed or left as is?

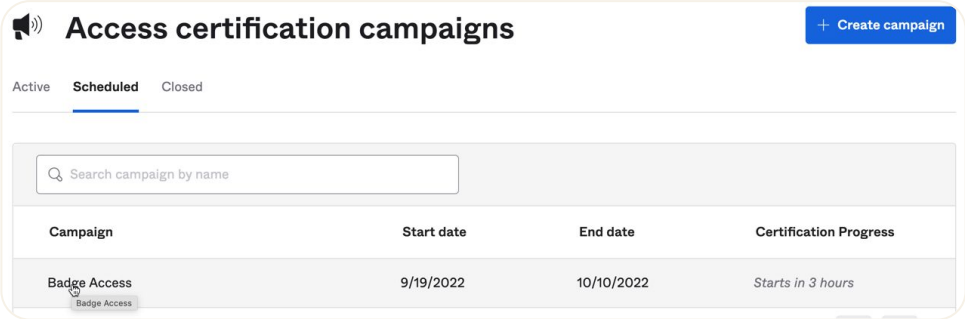
22. For the **Reviewer revoke access** option, select the **Remove user from resource** option
23. For the Reviewer does not respond option, leave the selection as Don't take any action



This is the last step in creating the campaign. You can go back and change things prior to scheduling.

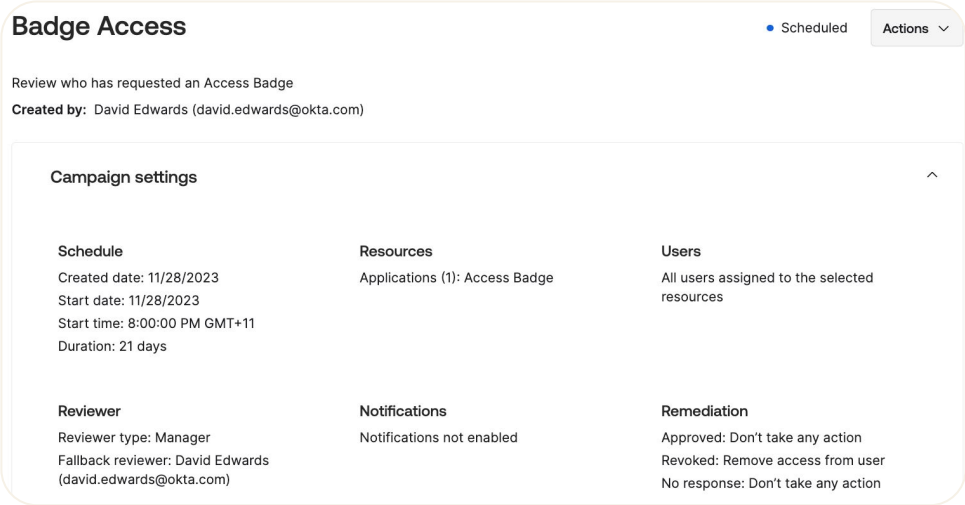
24. Click the **Schedule Campaign** button to schedule it

The campaign will now appear under the **Scheduled** tab.



25. Click on the **Campaign** name to open it

You will see a summary of the campaign.



This campaign will start on the schedule you specified. However, you can manually launch it, which we will do next.

Launching an Access Certification Campaign

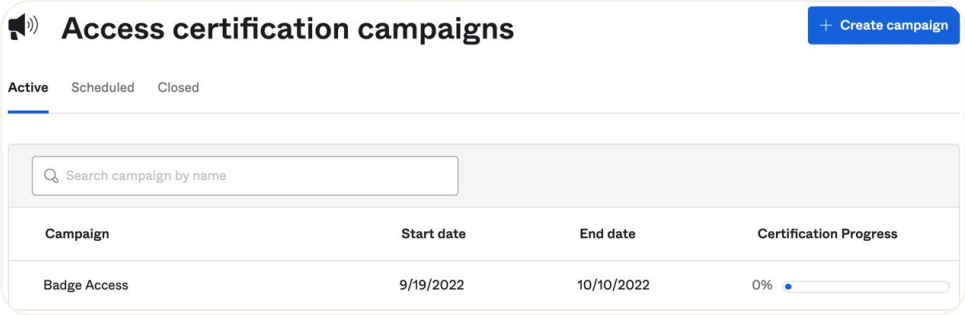
To launch the campaign immediately:

1. Click the Actions button from the campaign view
2. Select the Launch option
3. On the Launch campaign confirmation screen, click the Launch button

> Note that you can also **Edit** and **Delete** the campaign before it is launched (under the **Actions** button).

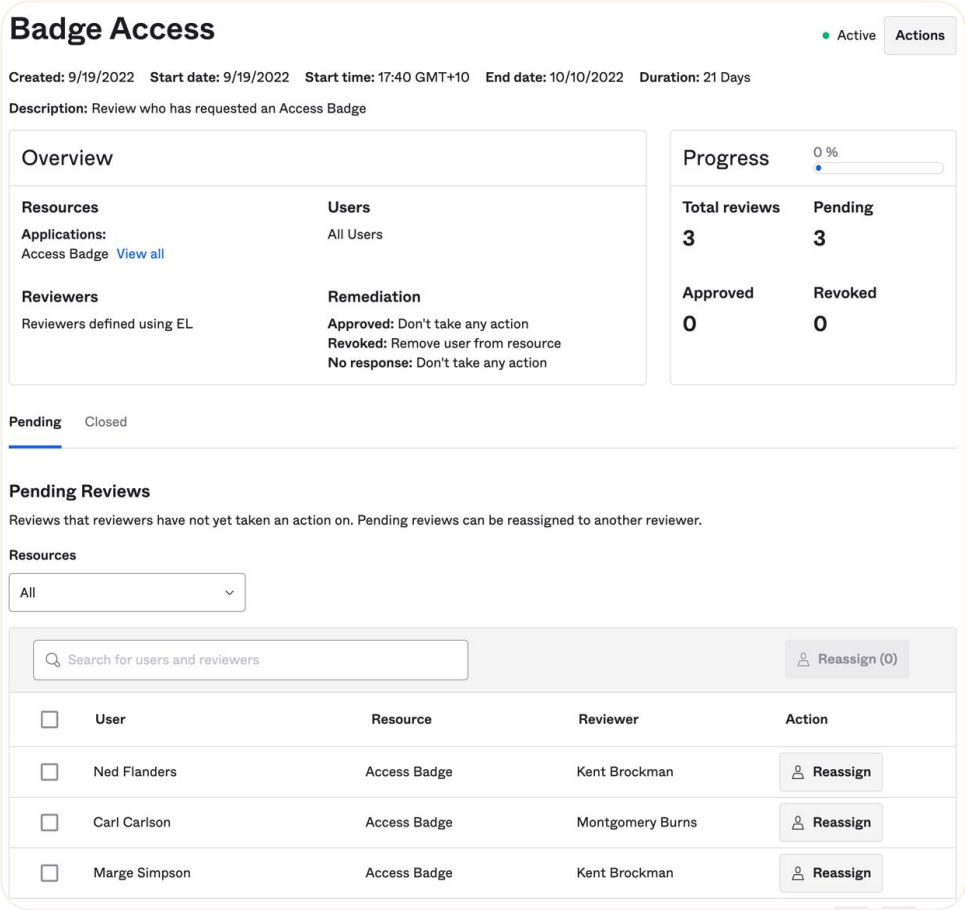
You will return to the Access certification campaigns page.

4. Go to the **Active** tab
5. If you don't see your campaign, refresh the page



6. Click on the campaign name to see the campaign details

The page provides an overview of the campaign (the same information that was presented pre-launch), the progress of the campaign, and the items to be reviewed.



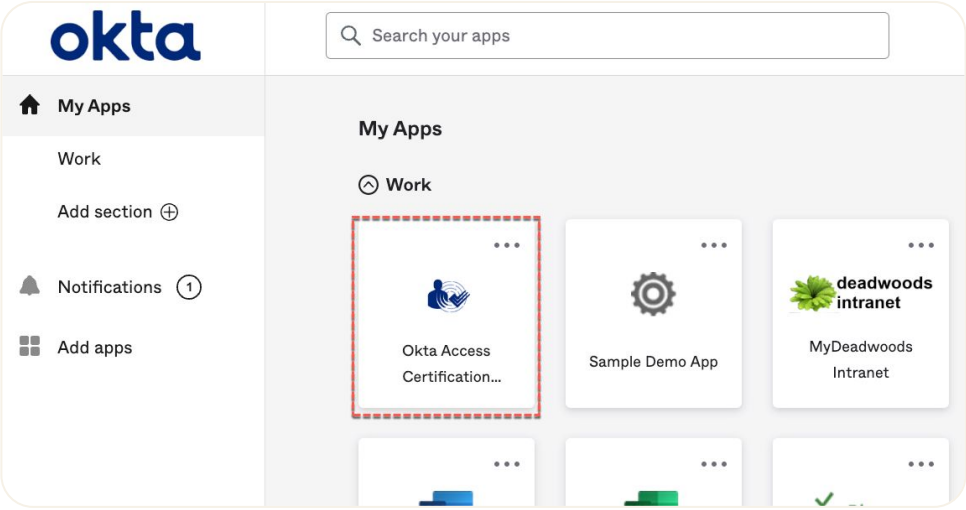
Depending on the users (and their manager settings) assigned to the selected application, you should see some managers with one or more users (like Kent Brockman in the screenshot below). You may also have users that don't have managers assigned, so they have the fallback reviewer as the reviewer.

An administrator can use this page to manage the execution of a campaign. They can see the progress and any items still outstanding. They can reassign one or more items. They can also prematurely end the campaign (Actions > End).

Participating in an Access Certification Campaign

Now that the campaign is running, we can switch to the role of review and review some access. Select one of the reviewers in your list above to perform the review.

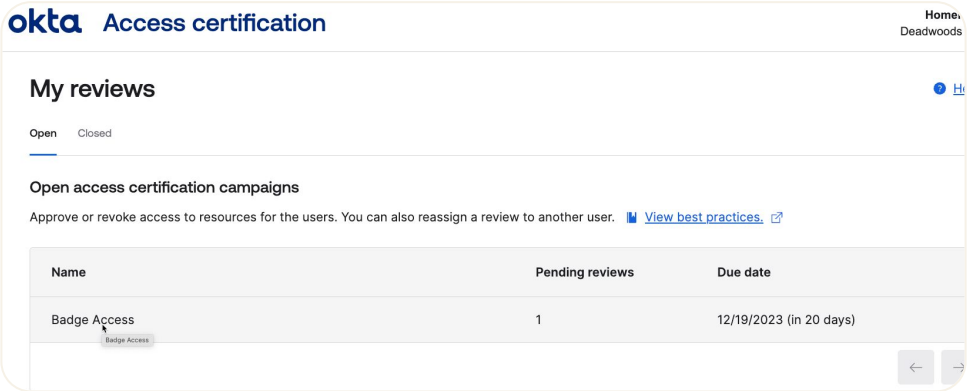
1. Log into the **Okta dashboard** as that reviewer user
2. Look for the **Okta Access Certification Reviews** tile and click it



This tile (Okta application) is tied to a group where membership is dynamically managed within Okta. If an Okta user is flagged as a reviewer in any active campaign, they will be in the group and see the application on the dashboard.

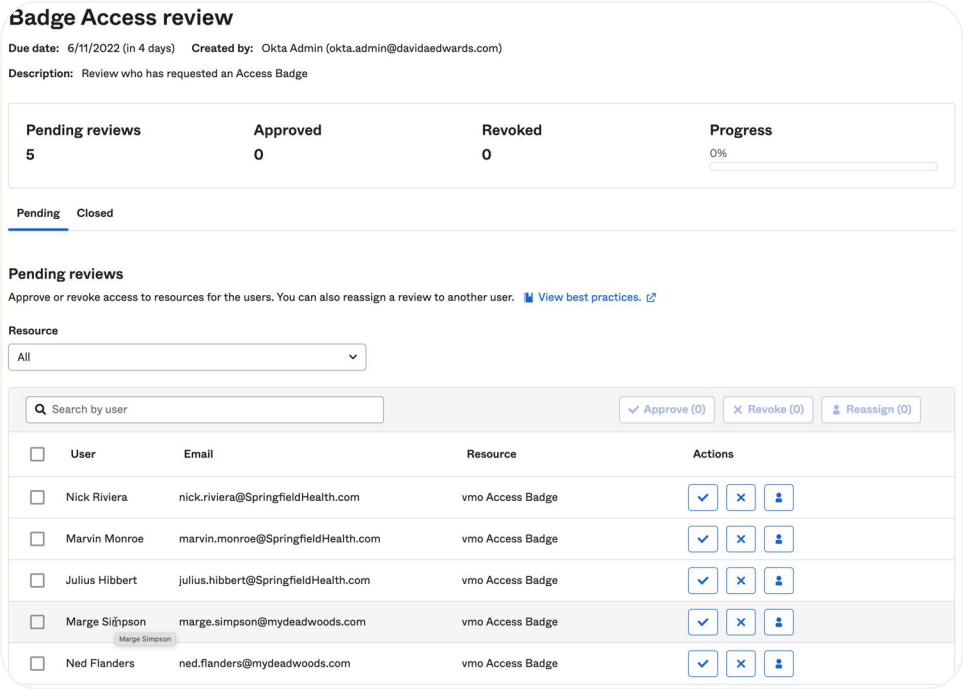
3. If offered, you can go through the tour of the Okta Access Certifications at some other time. For now, click the **No, thanks** option

You are presented with a list of campaigns the user is a reviewer for. You see a summary of information about each active campaign.



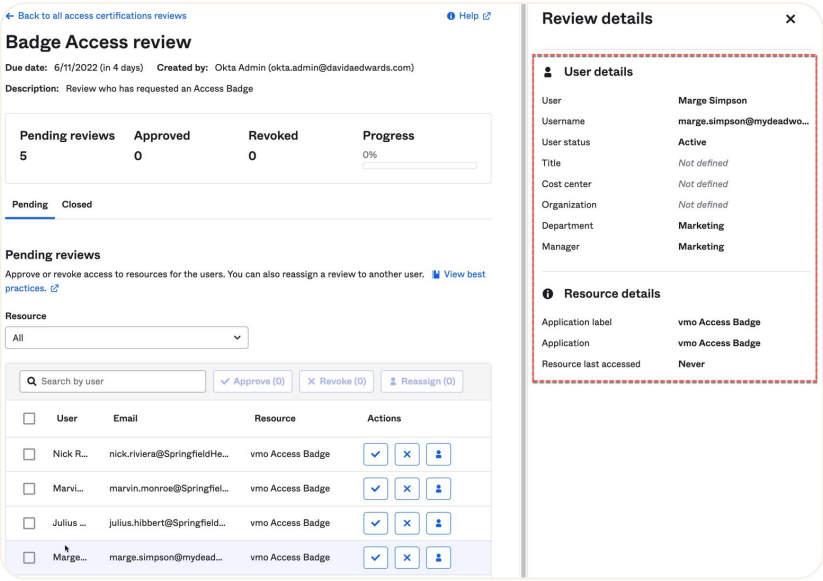
4. Click on the name of the campaign to open it

You will see all users for this application reporting to this manager.



For each item you have three options as a reviewer: Approve (leave the access as is), Revoke (remove access or flag for it to be removed) or Reassign (to another Okta user). Depending on screen resolution you will just see the icons or icons and words.

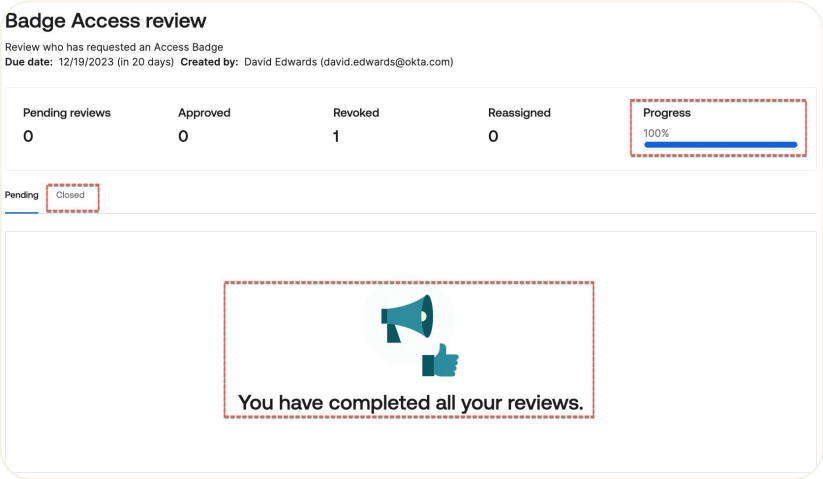
5. Click on the name of one of the users to see information about the user and access



This information is provided to help the reviewer in making their decision and currently includes user details and resource details (including fine-grained entitlements, like Roles) and review history.

6. Click the X to close the Review details window
7. Select one user and click the Revoke (X) button
8. When prompted, enter a Justification and click the Submit button
9. Process all other users, selecting any of the Approve (tick), Revoke (X) or Reassign (person) buttons until all users are actioned

After each action, you will see a message displayed and the item will disappear from the view (they can be found under the Closed tab). When all are actioned, you will get confirmation that you are done with the reviews.



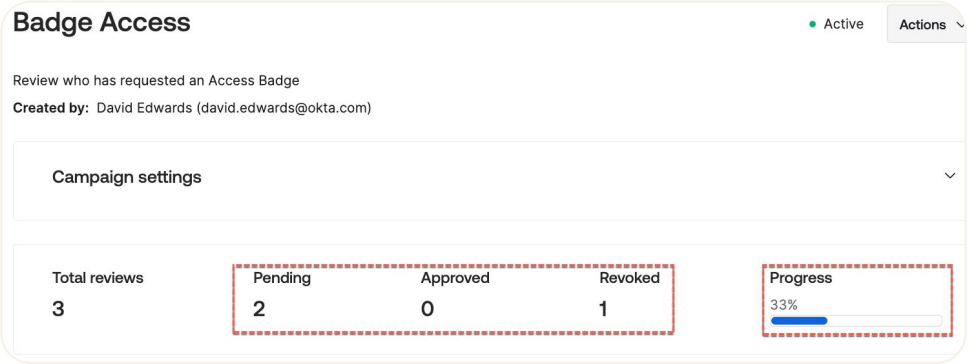
The **Pending** view is cleared, but you can still see the reviews under the Closed tab.

Now that the reviewer has completed their review, we will close out the campaign as the administrator.

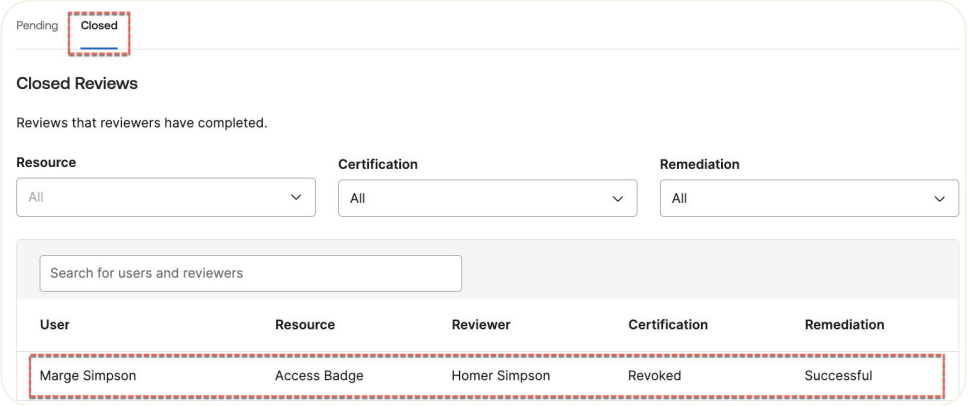
Managing an Access Certification Campaign

As mentioned earlier, we can manage running campaigns as the administrator.

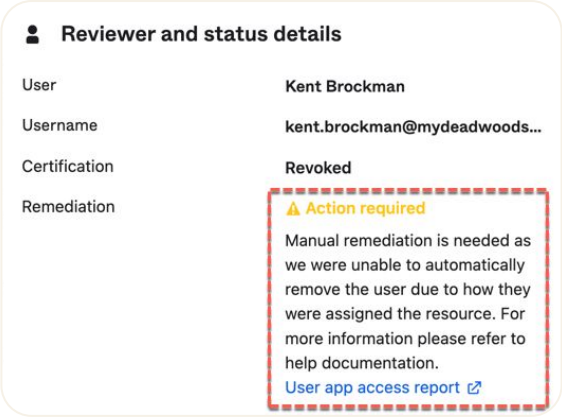
1. If not still there, log into **Okta** and then the **Admin** console as the administrator you used earlier.
2. Go to **Identity Governance > Access Certifications** and select the campaign



3. You can see the progress of the campaign after the reviewer has completed their review under the **Closed** tab.



4. You can see the progress of the campaign after the reviewer has completed their review under the **Closed** tab.

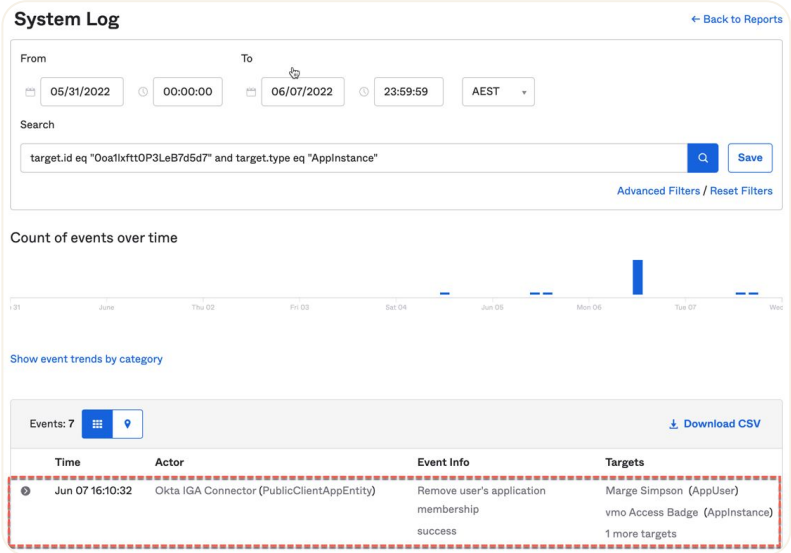


This indicates that Okta couldn't automatically revoke access. In this case, the user is assigned via a group with a group rule.

5. Close this dialog
6. On the campaign view page, select **Actions > End** to end the campaign
7. Click **End Campaign** on the confirmation screen

We can check the revocation actions:

8. Go to the application and confirm that users (where the remediation was Successful) were removed
9. At the top of the application page, click the View Logs link
10. Look for the **Okta IGA Connector** entry for your revoked user



Ignore the first event – it is related to Entitlement Management that we will look at later. The highlighted event shows the remediation has worked as expected.

Summary of Getting Started with Access Certifications

This completes the guided steps around access certifications. In the next section we will look at Reporting.

In this section we have walked through the creation and launch of an Access Certification campaign, then showing how a reviewer participates in the campaign and how access can be revoked. As has been shown, the Access Certification mechanism is straightforward and easy to use for reviewing group membership and application assignment.

Exploring Okta Identity Governance Reporting

The final capability we will explore is Reporting. The reporting engine in Okta has been rewritten to support the new governance reporting needs. New IGA reports are being added continuously (the list below is as at early Jun 2022).

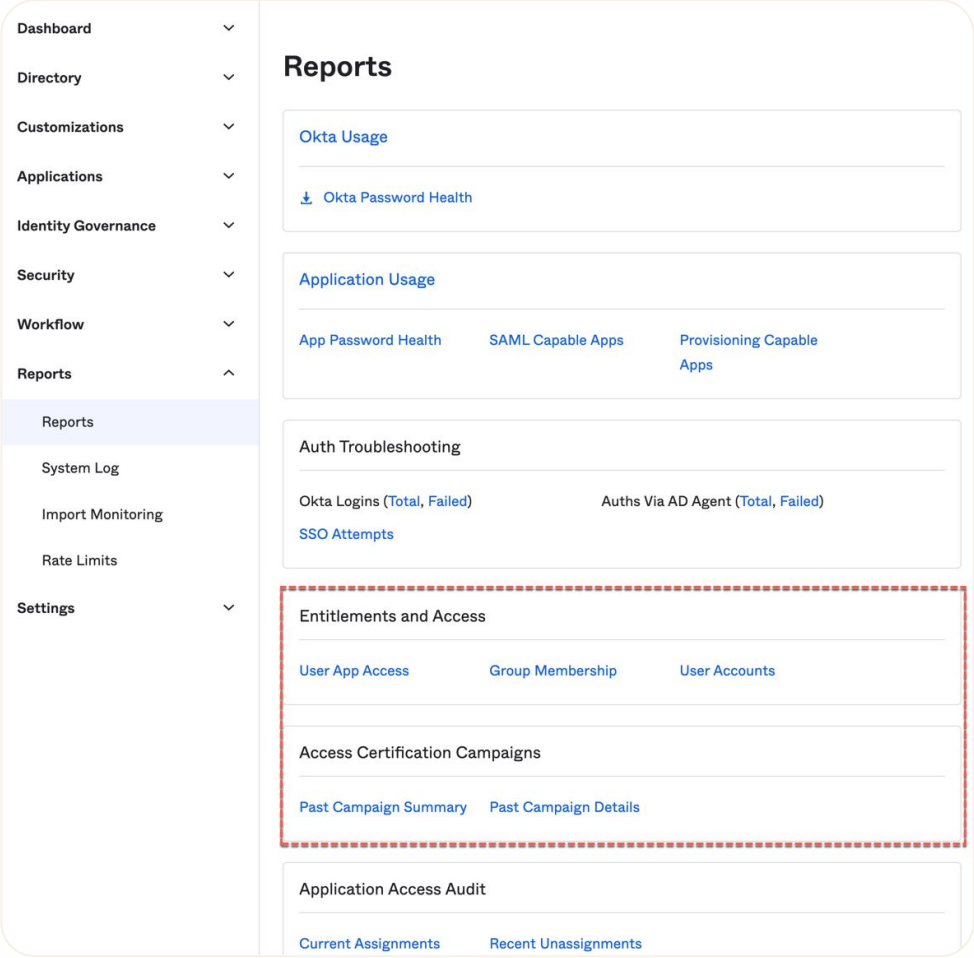
This section of the document will explore some of the IGA reports.

The documentation to support this can be found at: https://help.okta.com/en-us/Content/Topics/identity-governance/iga-reports.htm.

Accessing Reports

Reporting is in the Okta Admin console:

1. Log into the **Okta dashboard** as an administrator and go to **Admin**
2. Go to **Reports > Reports**



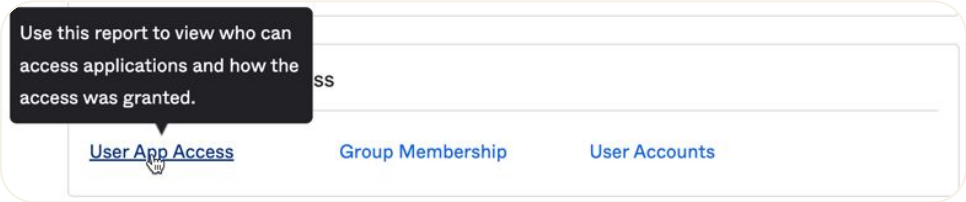
Whilst there are existing reports that may be relevant to IGA, there are two new sets of reports: **Entitlements and Access**, and **Access Certification Campaigns** reports. We will explore these below.

> With the new reports there are two ways a report may be delivered – it may be generated and presented in real-time as has been the case to now, or it may be generated in the background with the requester getting an email with a link to review the report. This is set for individual reports – you cannot specify which delivery mechanism.

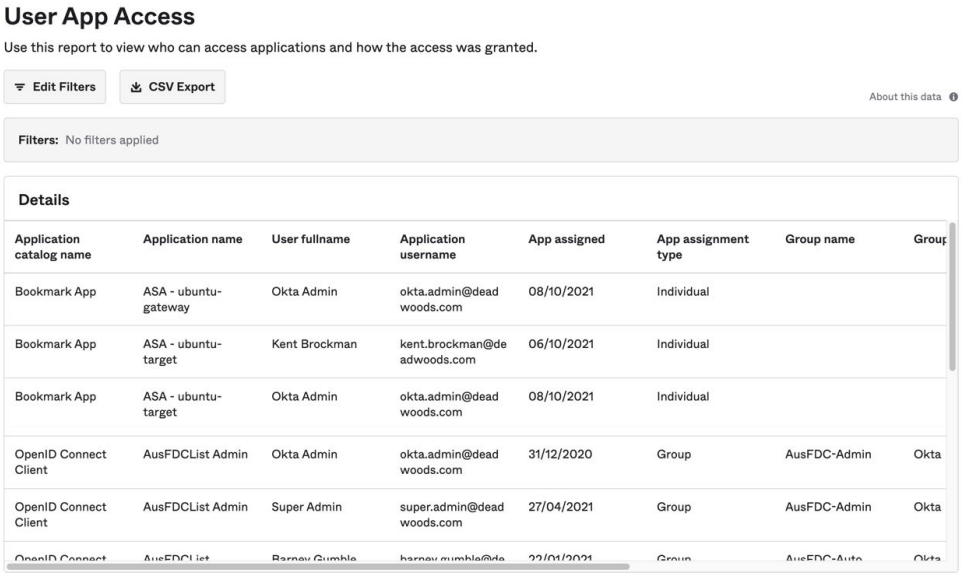
Entitlement and Access Reports

We will have a look at some of these reports.

1. From the **Reports** page select the **User App Access** report



You are presented with an unfiltered view of all application assignments.



Application catalog name	Application name	User fullname	Application username	App assigned	App assignment type	Group name	Group
Bookmark App	ASA - ubuntu-gateway	Okta Admin	okta.admin@deadwoods.com	08/10/2021	Individual		
Bookmark App	ASA - ubuntu-target	Kent Brockman	kent.brockman@deadwoods.com	06/10/2021	Individual		
Bookmark App	ASA - ubuntu-target	Okta Admin	okta.admin@deadwoods.com	08/10/2021	Individual		
OpenID Connect Client	AusFDCList Admin	Okta Admin	okta.admin@deadwoods.com	31/12/2020	Group	AusFDC-Admin	Okta
OpenID Connect Client	AusFDCList Admin	Super Admin	super.admin@deadwoods.com	27/04/2021	Group	AusFDC-Admin	Okta

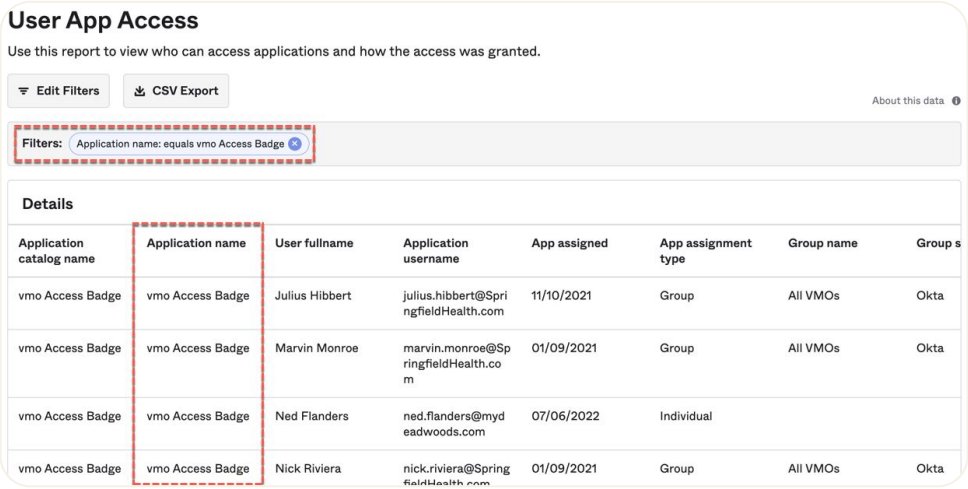
The contents are self-explanatory, but the important information is how the app was assigned (Individual or Group) and if it's a group assignment, what the group is, and how the group membership was done (Direct, i.e. manual, or By Rule, i.e. automatic).

App assignment type	Group name	Group source	Group membership type	Okta username	Okta user status
Individual				okta.admin@deadwoods.com	ACTIVE
Group	AusFDC-Admin	Okta	Direct	okta.admin@deadwoods.com	ACTIVE
Group	AusFDC-Admin	Okta	Direct	super.admin@deadwoods.com	ACTIVE
Group	AusFDC-Auto	Okta	By Rule	barney.gumble@deadwoods.com	STAGED
Group	AusFDC-Auto	Okta	By Rule	betty.wilson@deadwoods.com	PROVISIONED

This report could be very useful when looking at building Access Certification campaigns and how remediation should be applied.

As with all reports, you can apply filters or export the report as a CSV file for further analysis.

2. Apply a filter to look for a specific application in your system



Application catalog name	Application name	User fullname	Application username	App assigned	App assignment type	Group name	Group s
vmo Access Badge	vmo Access Badge	Julius Hibbert	julius.hibbert@SpringfieldHealth.com	11/10/2021	Group	All VMOs	Okta
vmo Access Badge	vmo Access Badge	Marvin Monroe	marvin.monroe@SpringfieldHealth.com	01/09/2021	Group	All VMOs	Okta
vmo Access Badge	vmo Access Badge	Ned Flanders	ned.flanders@mydeadwoods.com	07/06/2022	Individual		
vmo Access Badge	vmo Access Badge	Nick Riviera	nick.riviera@SpringfieldHealth.com	01/09/2021	Group	All VMOs	Okta

3. Go back to the **Reports** page

4. Select the **Group Membership** report

Past Access Requests

Use this report to view details of Access requests

Edit Filters | CSV Export | Abou

Filters: No filters applied

Details

Request ID	Resource type	Resource name	Requester	Requested	Approver	Decided
304	Application	Access Badge	Marge Simpson	11/27/2023	Homer Simpson	11/27/2023
304	Application	Access Badge	Marge Simpson	11/27/2023	David Edwards	11/27/2023
264			Seymour Skinner	8/10/2023		
303	Application	Access Badge	Seymour Skinner	11/27/2023	Mayor Quimby	11/27/2023
302	Application	Deadwoods O365	Seymour Skinner	11/9/2023	David Edwards	11/9/2023
301	Application	Deadwoods O365	Seymour Skinner	11/9/2023	David Edwards	11/9/2023
300	Application	Deadwoods O365	Seymour Skinner	11/9/2023	David Edwards	11/9/2023
299	Application	EM Salesforce.com	Seymour Skinner	11/9/2023	David Edwards	11/9/2023

The first column is a link to the details in Access Requests. The other columns give a summary of the request, such as what it was for, who initiated it, who the approver was and the date.

5. Scroll to the right of the report if you don't see all the columns.

Past Access Requests

Use this report to view details of Access requests

Edit Filters | CSV Export | About this dat

Filters: No filters applied

Details

Approver	Decided	Decision	Request type name	Time to resolved	Time bounded	Resource granted
Homer Simpson	11/27/2023	Approved	Access Badge	0.4	false	true
David Edwards	11/27/2023	Approved	Access Badge	0.4	false	true
			Shift1 Approval	2,615.3	false	false
Mayor Quimby	11/27/2023	Approved	Badge Access	0	false	true
David Edwards	11/9/2023	Approved	O365 Helpdesk	0	false	true
David Edwards	11/9/2023	Approved	O365 Helpdesk	0	false	true
David Edwards	11/9/2023	Approved	O365 Helpdesk	0	false	true
David Edwards	11/9/2023	Approved	SFDC Marketing Entitlements	0	false	true

In this view you can see the decision for the request, the request type name, the time to resolve and whether it was granted or not.

6. Go back to the **Reports** page
7. Select the **Group Membership** report

As before, you get an unfiltered list of all users for each group.

Group Membership

Use this report to view group members and how they were granted their membership.

Edit Filters | CSV Export | About this dat

Filters: No filters applied

Details

Group name	Group source	Group description	User fullname	Okta username	Okta user status	Group membership type	Gro mer crea
ASA Users	adjoined.local	adjoined.local/Users/ASA Users	Bugs Bunny	bugs@adjoined.local	ACTIVE	By Import	05/
ASA Users	adjoined.local	adjoined.local/Users/ASA Users	Daffy Duck	daffy@adjoined.local	ACTIVE	By Import	05/
ASA Users	adjoined.local	adjoined.local/Users/ASA Users	Yosemite Sam	yosemite@adjoined.local	ACTIVE	By Import	31/

> The groups are not just Okta-managed groups, you will see all groups in your Okta instance (for example, the screen shot above shows a group from the AD instance adjoined.local).

As before you can filter the report or download it as CSV.

Access Certification Campaign Reports

There are two reports for Access Certification Campaigns – a summary report and a detail report.

1. Go to the **Reports > Reports** page
2. Select the **Past Campaign Summary** report

Past Campaign Summary

Use this report to view the summary of Access certification campaigns

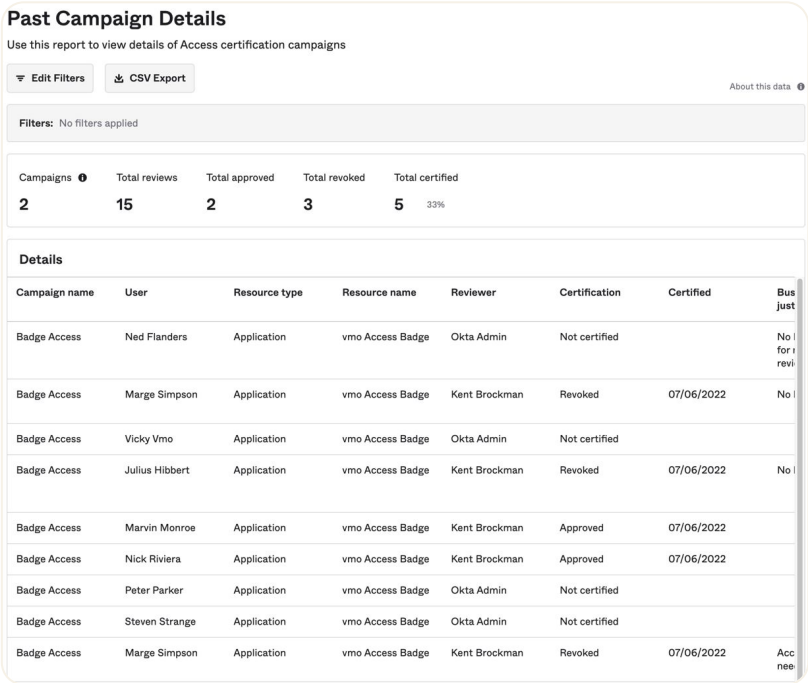
Edit Filters | CSV Export | About this data

Filters: No filters applied

Campaigns	Total reviews	Total approved	Total revoked	Total certified
2	15	2	3	5 33%

Details

Campaign name	Campaign started	Campaign ended	Duration (days)	Resource type	User scope	Number of users	Number items
Badge Access	07/06/2022	07/06/2022	1	Application	All Users	8	



Past Campaign Details

Use this report to view details of Access certification campaigns

Edit Filters | CSV Export | About this data

Filters: No filters applied

Campaigns	Total reviews	Total approved	Total revoked	Total certified
2	15	2	3	5 33%

Details

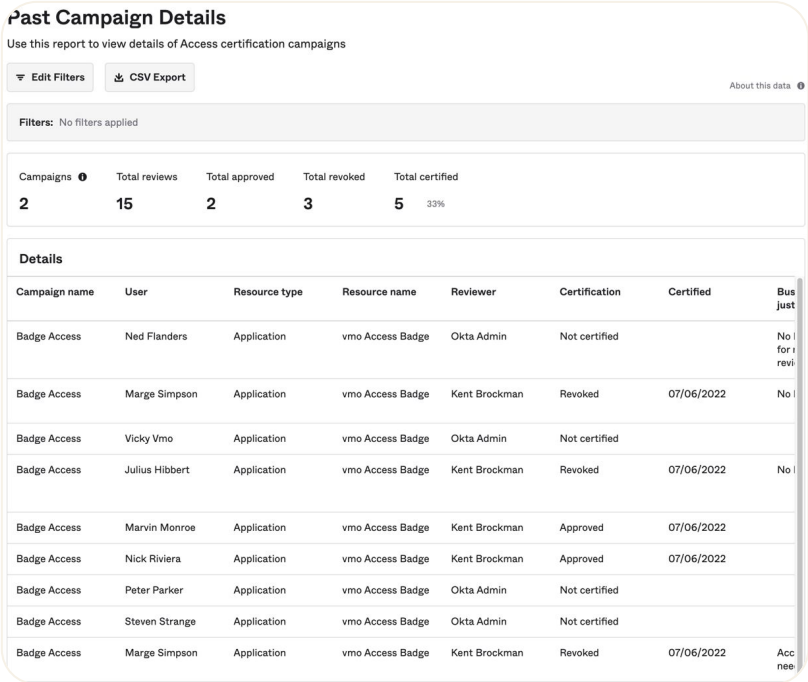
Campaign name	User	Resource type	Resource name	Reviewer	Certification	Certified	Bus just
Badge Access	Ned Flanders	Application	vmo Access Badge	Okta Admin	Not certified		No for revi
Badge Access	Marge Simpson	Application	vmo Access Badge	Kent Brockman	Revoked	07/06/2022	No
Badge Access	Vicky Vmo	Application	vmo Access Badge	Okta Admin	Not certified		
Badge Access	Julius Hibbert	Application	vmo Access Badge	Kent Brockman	Revoked	07/06/2022	No
Badge Access	Marvin Monroe	Application	vmo Access Badge	Kent Brockman	Approved	07/06/2022	
Badge Access	Nick Riviera	Application	vmo Access Badge	Kent Brockman	Approved	07/06/2022	
Badge Access	Peter Parker	Application	vmo Access Badge	Okta Admin	Not certified		
Badge Access	Steven Strange	Application	vmo Access Badge	Okta Admin	Not certified		
Badge Access	Marge Simpson	Application	vmo Access Badge	Kent Brockman	Revoked	07/06/2022	Acc nee

The report shows a summary of all campaigns run, and for each information about the specific campaign. You can apply filters or download a CSV.

Whilst this is interesting, a more useful report is the campaign details report.

8. Go back to the **Reports** page
9. Select the **Past Campaign Details** report

This report shows all the campaign items for all campaigns.



Past Campaign Details

Use this report to view details of Access certification campaigns

Edit Filters | CSV Export | About this data

Filters: No filters applied

Campaigns	Total reviews	Total approved	Total revoked	Total certified
2	15	2	3	5 33%

Details

Campaign name	User	Resource type	Resource name	Reviewer	Certification	Certified	Bus just
Badge Access	Ned Flanders	Application	vmo Access Badge	Okta Admin	Not certified		No for revi
Badge Access	Marge Simpson	Application	vmo Access Badge	Kent Brockman	Revoked	07/06/2022	No
Badge Access	Vicky Vmo	Application	vmo Access Badge	Okta Admin	Not certified		
Badge Access	Julius Hibbert	Application	vmo Access Badge	Kent Brockman	Revoked	07/06/2022	No
Badge Access	Marvin Monroe	Application	vmo Access Badge	Kent Brockman	Approved	07/06/2022	
Badge Access	Nick Riviera	Application	vmo Access Badge	Kent Brockman	Approved	07/06/2022	
Badge Access	Peter Parker	Application	vmo Access Badge	Okta Admin	Not certified		
Badge Access	Steven Strange	Application	vmo Access Badge	Okta Admin	Not certified		
Badge Access	Marge Simpson	Application	vmo Access Badge	Kent Brockman	Revoked	07/06/2022	Acc nee

You can scroll to the right to see more information...

Details

Certification	Certified	Business justification	Attempted remediation	Remediation status	Campaign started	Campaign ended
Not certified		No longer working for me. Please review or reassign.	No action taken	Successful	07/06/2022	07/06/2022
Revoked	07/06/2022	No longer needed	Remove from resource	Successful	07/06/2022	07/06/2022
Not certified			No action taken	Successful	07/06/2022	07/06/2022
Revoked	07/06/2022	No longer needed	Remove from resource	Manual remediation required	07/06/2022	07/06/2022
Approved	07/06/2022		No action taken	Successful	07/06/2022	07/06/2022
Approved	07/06/2022		No action taken	Successful	07/06/2022	07/06/2022
Not certified			No action taken	Successful	07/06/2022	07/06/2022
Not certified			No action taken	Successful	07/06/2022	07/06/2022
Revoked	07/06/2022	Access no longer needed	No action taken	Successful	07/06/2022	07/06/2022

In addition to the user and resource (i.e. application or group), for the review item it shows who the Reviewer was, the outcome **(Certification)**, the date **(Certified)**, **Business Justification** (if entered) and the results of the remediation **(Attempted remediation and Remediation status)**.

Again, you could filter (say specific campaigns, users, reviewers or applications/groups) making this a very useful report. You can also download it as a CSV.

Summary of Getting Started with Reporting

This completes the guided steps around reporting.

In this section we have walked through some of the new IGA reports available with Okta Identity Governance and how they can be used.

Exploring Entitlement Management in OIG

Entitlement Management is a new capability being introduced into Okta Identity Governance (OIG). At the time of writing this (Dec 23) this capability is in Limited Early Access (LEA) and will be made Generally Available to OIG customers later in 2023/2024.

Introduction

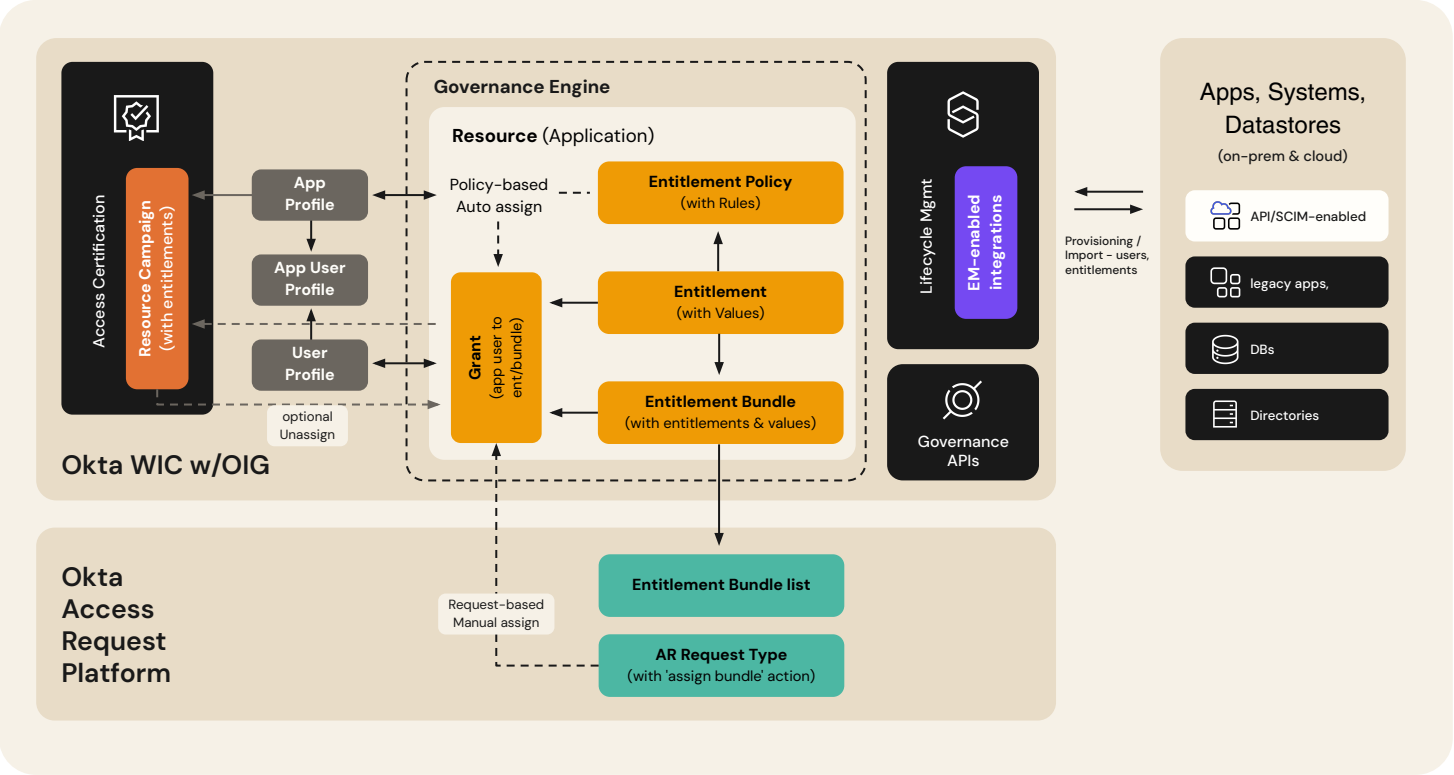
This section provides an architectural overview of the OIG Entitlement Management components and some context for the use cases covered in this guide.

For more information you can also see the Okta Identity Governance documentation.

> Note: The capabilities described here represent those in the Limited Early Access phase and may be subject to change.

Architectural Overview of Entitlement Management

Entitlement Management is a new capability that has been added to Okta Identity Governance (OIG). It supports both automatic policy-based assignment and request-based assignment of entitlements. The following diagram shows the major components and data objects:



The main component of OIG Cloud Entitlement Management is the new **Governance Engine.** When an Okta Application has the Governance Engine enabled, a new Resource is created in the Governance Engine to represent that application.

The central object is the **Entitlement**. Entitlements represent entitlements on connected systems (like a role, profile or license). A resource can have multiple entitlements and each entitlement can have a set of values. Entitlements can be single-valued or multi-valued.

Entitlements can be automatically assigned to users via an **Entitlement Policy.** This attribute-based automatic assignment approach is analogous to Group Rules for assigning users to Groups based on some attribute. There will be one policy for a resource, but it may have multiple rules that are applied based on priority to determine what entitlements can be granted to a user.

Entitlements can also be collected into Entitlement Bundles. Bundles represent logical groupings of entitlements and one bundle may contain multiple entitlements and multiple values for each (for multi-valued entitlements). A bundle might represent a jobrole where all entitlements for a specific job in a single application are bundled together. Or a bundle may be a set of accesses one might request, such as an employee visiting head office needs both building and carpark access, and a bundle could be created to put them in a single access request.

Grants represent the association of a user with an entitlement or an entitlement bundle. With Entitlement Management, the application user profile is no longer used to store entitlements as was done in Lifecycle Management (LCM).

With Entitlement Management, entitlement bundles are exposed to the Access Requests Platform and are resources that can be requested in a Request Type in the same way that groups and applications can. A list of bundles is synchronized to Access Requests into an **Entitlement Bundle** list that can be used in Request Types to request access. Configuration lists can be built as a subset of entitlement bundles. There is a new Okta action to assign a user to an entitlement bundle.

All entitlement grants can be reviewed in an **Access Certification Campaign** for the Application resource type. If a user has both entitlement and bundle grants, it can show both and may allow automatic revocation where that doesn't conflict with policy.

Entitlement Management supports both a BYOE (Bring Your Own Entitlements) model and integration with SaaS applications. For application integration a new set of connectors has been built to support key applications. They can consume the entitlements and existing user-entitlement mapping, and also provision changes to user-entitlement mapping. These will replace the existing OIN connectors and the list will grow over time.

The API documentation provides more information about the Entitlement objects - https://developer.okta.com/docs/api/iga/.

Structure of this Section

This section of the Getting Started Guide is designed to be standalone. It does require Okta Identity Governance to be configured as per the earlier parts of this guide, and it is assumed that you are familiar with the OIG components and capabilities as covered in the earlier sections.

This section has guided labs to explore different aspects of Entitlement Management:

- **Prerequisites** - looking at the common set up steps required to run the remainder of the labs
- **BYO Entitlement Management** - using a dummy app in Okta, explore the data objects and how they are used in Access Requests and Access Certification. This forms a foundation for the subsequent labs looking at external application integration.
- **Entitlement Management for Microsoft Office 365 and Salesforce.com** - having explored the core components of Entitlement Management, this
  part of the lab will expand to include two live applications with entitlements - Microsoft Office 365 and Salesforce.com

The way this park of the lab guide is written, the BYO Entitlement Management section is more verbose when describing the steps and has more screenshots, whereas the Office365 and Salesforce sections don't have as much detail. You don't need to do all three sections, but if you decide to skip the BYO section, you may want to refer back to it if the other sections don't explain something fully.

Some of the screenshots used in this document may not be what you see in your environment as minor changes are constantly being fed in as part of the continuous delivery process (such as with the Access Requests web UI) and the names and objects you use may be different.

Prerequisites

Before starting the labs in this section, you will need the following completed.

Okta Org with required SKUs and features enabled

You will need an Okta Workforce Identity Cloud (OIE) org for setup with Okta Identity Governance (including Access Requests) configured. We do not cover the setup of this in this lab.

The new capability is enabled by Okta via feature flags. These are:

- **ATSPOKE_RAMP_INTEGRATION -** Ramp integration for AtSpoke (Access Requests)
- **ENTITLEMENT_MANAGEMENT -** Support for entitlement management feature as a part of OIG
- **SKYHOOK_RAMP_INTEGRATION -** Enable logic to prefer RAMP entitlement sourcing for designated applications

Your account team can confirm these have been enabled.

The first lab below will confirm that Entitlement Management is ready to use.

External Systems

To run the second part of this lab, you will need one or more of the following external systems configured so you can consume and provision entitlements:

- **A NEW Salesforce.com** instance (DO NOT enable Provisioning for the new app), and/or.
- **A NEW Microsoft Office 365** instance (DO NOT enable Provisioning for the new app).

You could use an existing Office 365 or Salesforce instance, but you will need to disable provisioning and may lose any user entitlements information.

The steps to enable provisioning have changed with Entitlement Management and are covered in the relevant section below.

BYO Entitlement Management

In this part of the lab we will explore Entitlement Management within Okta (we will look at external systems later). The aim of this part of the lab is to familiarize you with the data objects in Entitlement Management, and the user interfaces to interact with them.

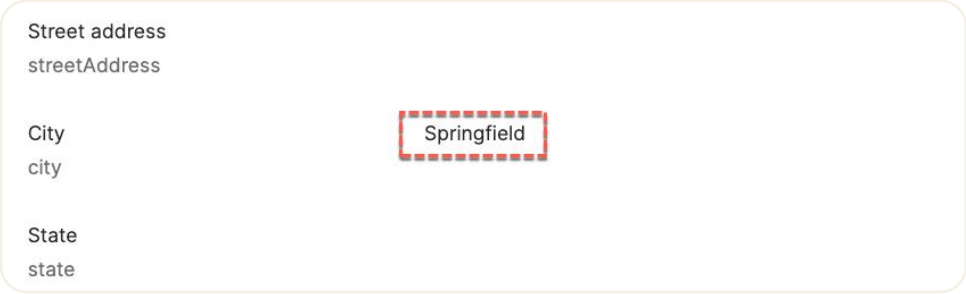
We will use a "dummy" application - a generic SCIM integration that isn't connected to anything. This is the approach you would use for a Bring Your Own (BYO) model where there is no pre-built integration for an app, and you either want to manually load/manage entitlements or you will build the integration yourself connected to the SCIM application in Okta. We will refer to this as the "Dummy App".

The first part of the lab will confirm that the Entitlement Management components are enabled. The dummy application will represent a Physical Access system for badging into carparks and office buildings. There will be a default set of accesses (entitlements) for local employees and requestable access for visitors and special building access.

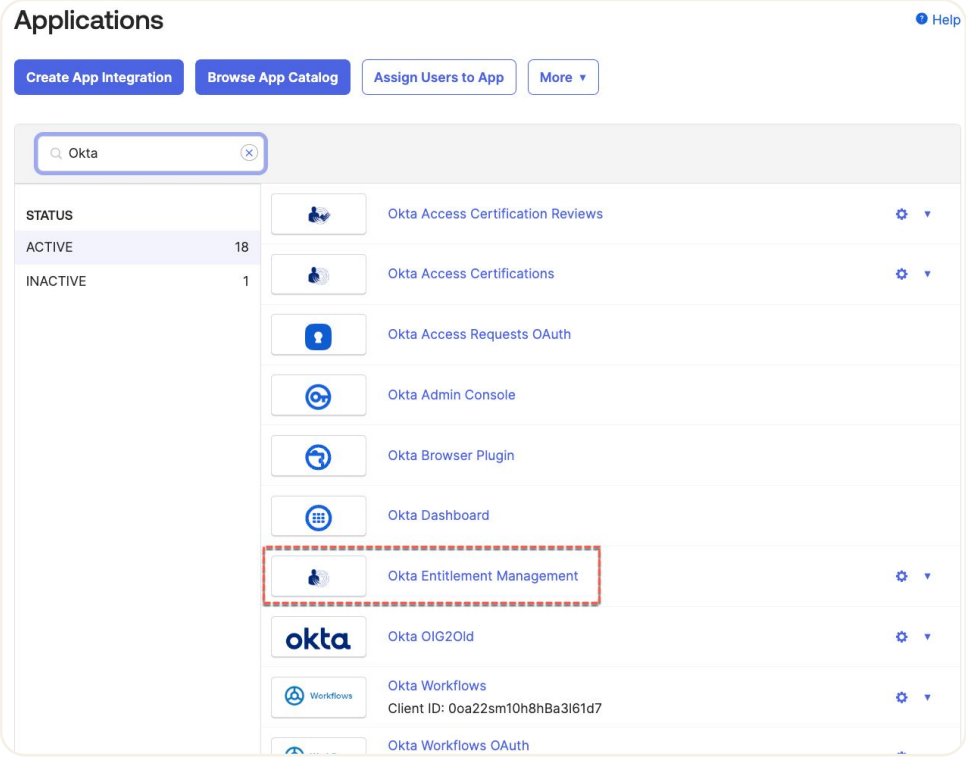
Initial Set Up in Okta

Okta Users

Prior to working with the Entitlements, you will need users in Okta and they will need some common attribute to use in the Entitlement Policy. For this example, I have some users who all have the same city value of "Springfield". You can choose any field.



There is a new **Okta Entitlement Management** app in the Applications list in Okta.



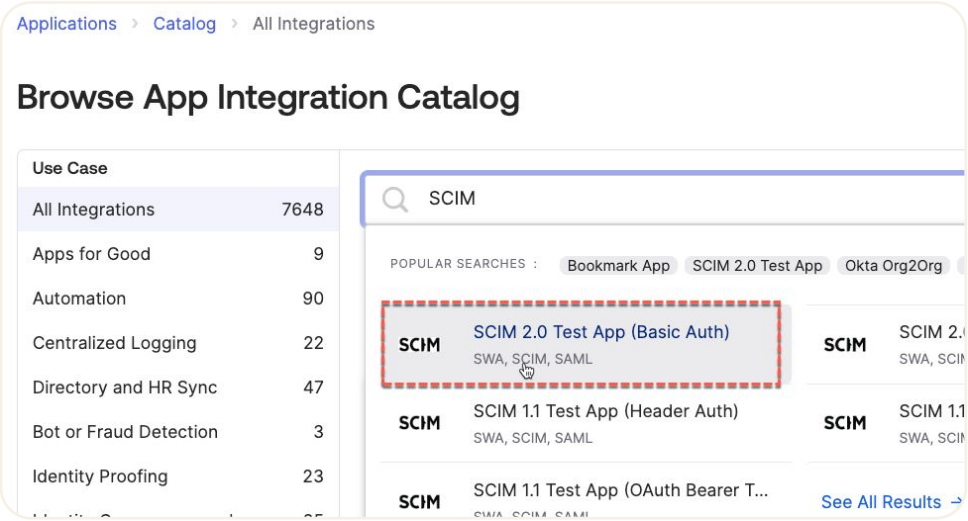
Make sure your administrator account is assigned to this app.

Set Up Application for Entitlement Management

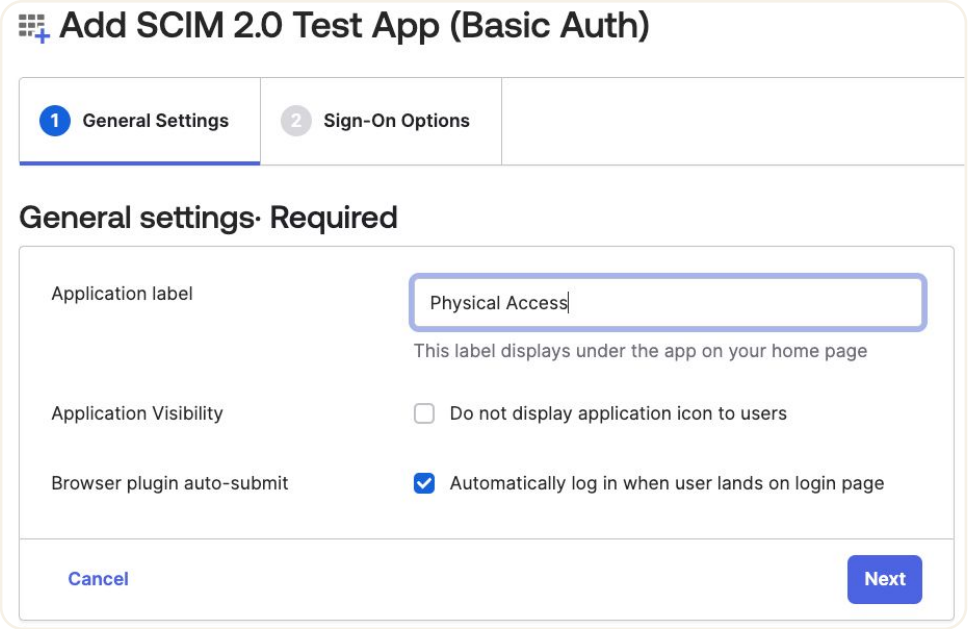
Create a Dummy SCIM Application

Entitlements need to be tied to an application in Okta. In later parts of this guide we will connect to a real external system, but for this part we want to explore the core entitlement management objects in Okta. The limitations page in the documentation (https://help.okta.com/en/programs/em/Content/Topics/identity-governance/em/limitations.htm) lists some options. The simplest is to create a SCIM template app.

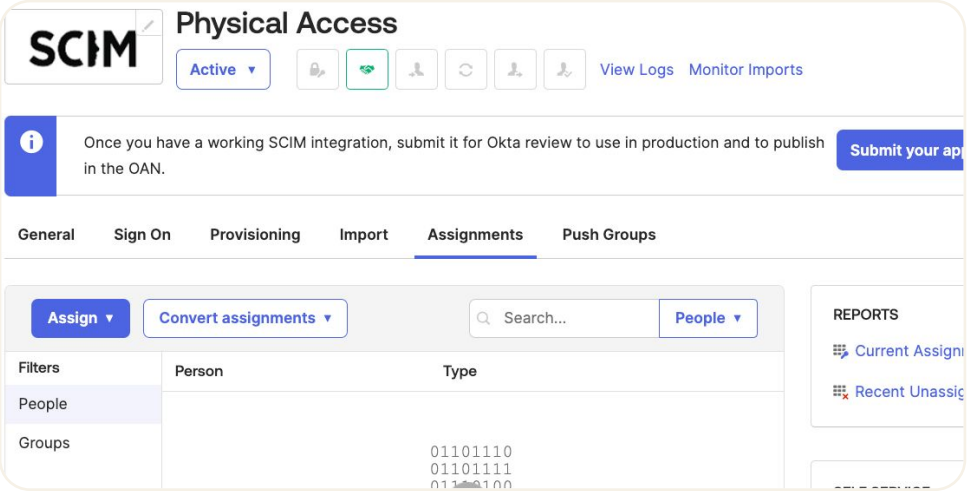
1. Go into the **Okta Admin UI** and go to **Applications**, **Applications**.
2. Select **Browse App Catalog**
3. Search for **SCIM**



4. Select the **SCIM 2.0 Test App (Basic Auth)**.
5. Select **Add Integration**
6. Give it a label and leave the other values as default. You may want to select the Application Visibility checkbox if there are other users in your system who may be confused by seeing this new application.



7. Click **Next**
8. On the **Sign-On Options** tab, leave everything as default and click Done. As we're not SSO'ing to the app, we don't care about the SSO settings, just that we have an app in Okta.



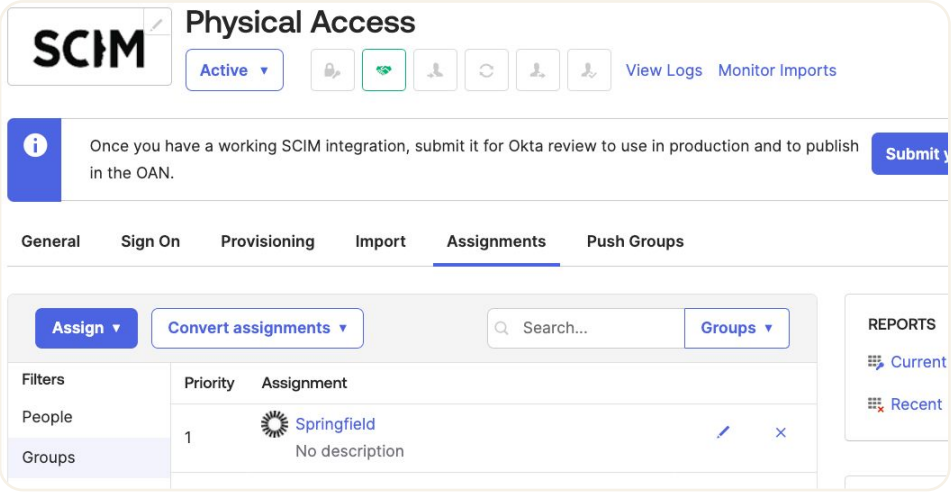
9. Assign a group of users to the new application, such as the Everyone group. I assigned one of the groups I have in my Okta org.

This application is now ready to assign entitlements to.

Enable the Governance Engine for the Application

Entitlement Management must be enabled for each application where entitlements will be managed. You will see reference to the **Governance Engine**, which is the entitlement management component.

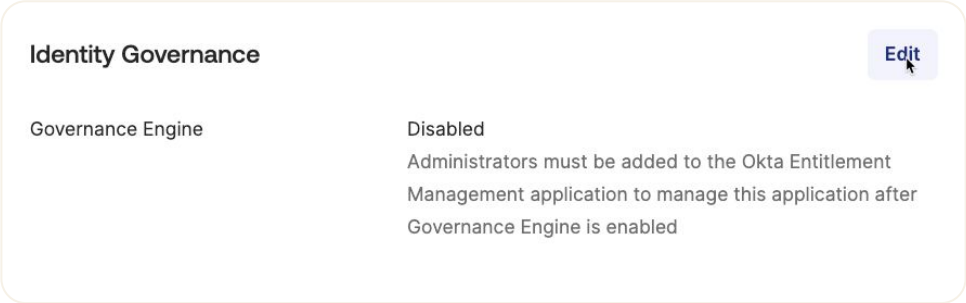
You can tell if an application has been enabled by looking at the app.



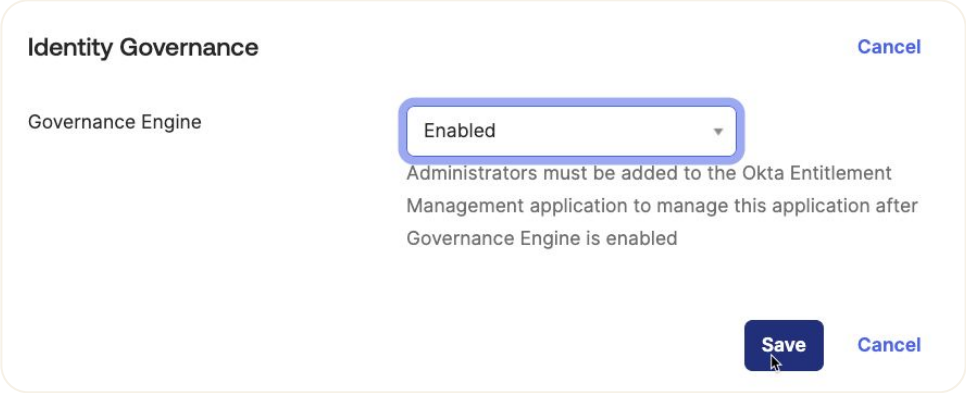
If there is no Governance tab, entitlement management has not been enabled.

To enable it:

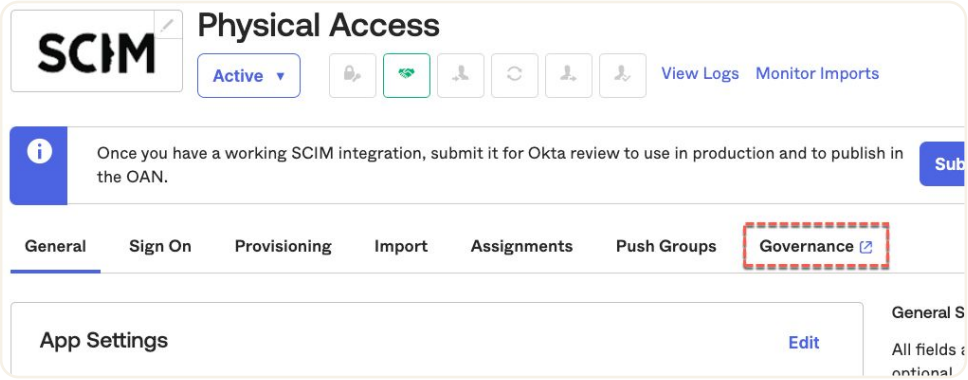
1. Go to the **General** tab
2. Find and **Edit** the **Identity Governance** section



3. Change it to **Enabled** and **Save**



4. Note the message about governance being configured in the background
5. Refresh the page to see the Governance tab appear



The app is now ready for Entitlement Management. At this point you have validated that Entitlement Management is enabled in your Okta Org.

Create Entitlements for Dummy App

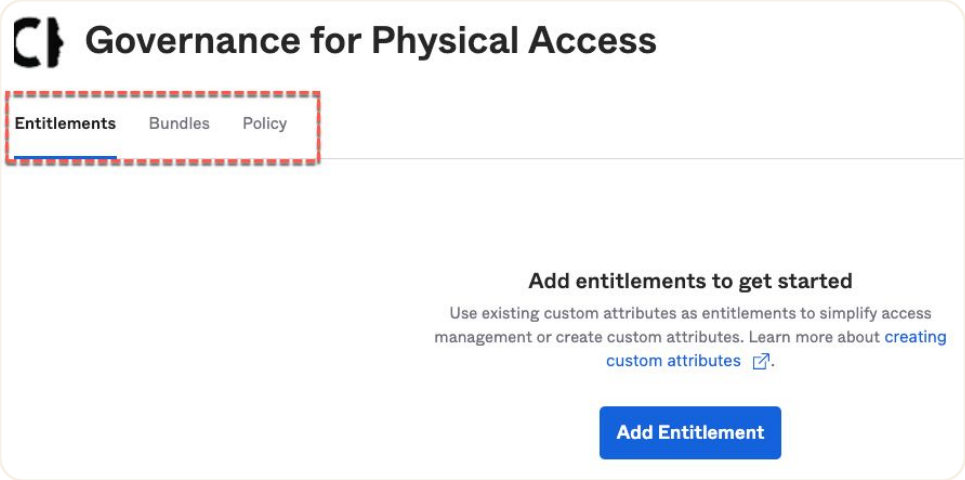
We can now explore Entitlement Management objects in OIG. We will look at the entitlements first. As the application isn't connected to a real external application, we will manually create the entitlements.

Open Governance Tab for the App

The Governance Engine aspect of the application is separate from the normal Okta application administration. When you enabled Identity Governance above, this created a representation of the application in the Governance Engine called a Resource. All entitlement objects are tied back to a resource (application).

To open the Governance Engine for this app:

1. Click on the **Governance** tab



There are three tabs for governance:

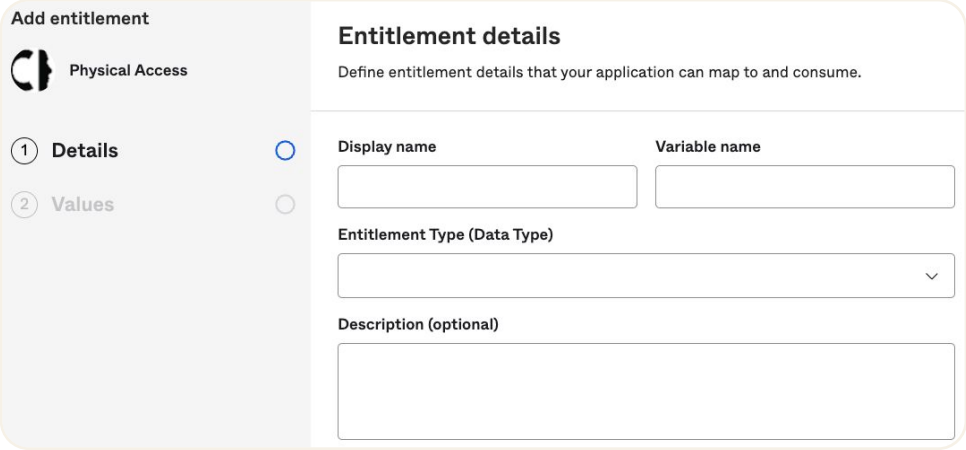
- **Entitlements** - create and manage entitlements,
- **Bundles** - create and manage entitlement bundles, and
- **Policy** - create and manage entitlement policy

We will walk through each of these in the next sections.

Create Entitlements

First we will create two types of entitlements for the app - building access and carpark access.

1. Select the **Add Entitlement** button to get started



This view is for creating an entitlement type. We have two types of entitlements - building access and car park access.

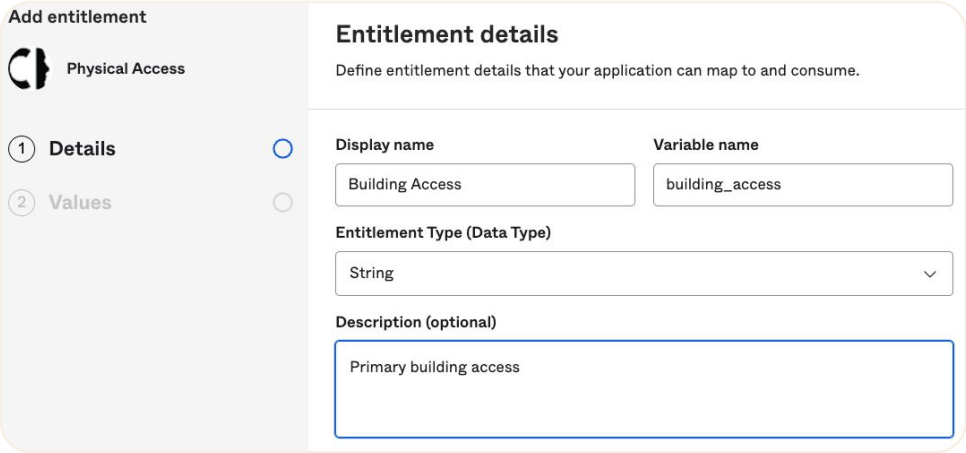
2. Create a new entitlement type with the following values:

Display name = *Building Access*

Variable name = *building_access*

Entitlement Type (Data Type) = string

Description = anything



1. Click **Next**

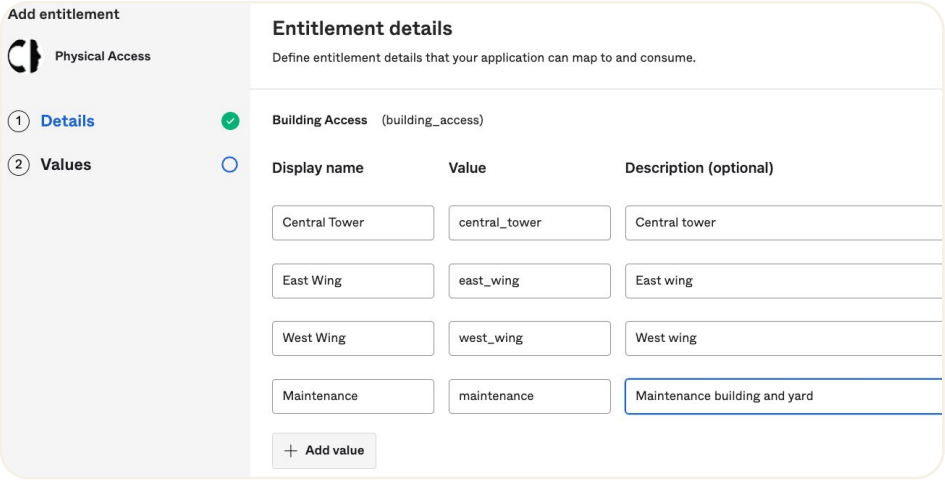
2. Next we add entitlement values for the entitlement type. Enter the following values **(Display name, Value, Description)**:

   Central Tower, central_tower, Central tower

   East Wing, east_wing, East wing

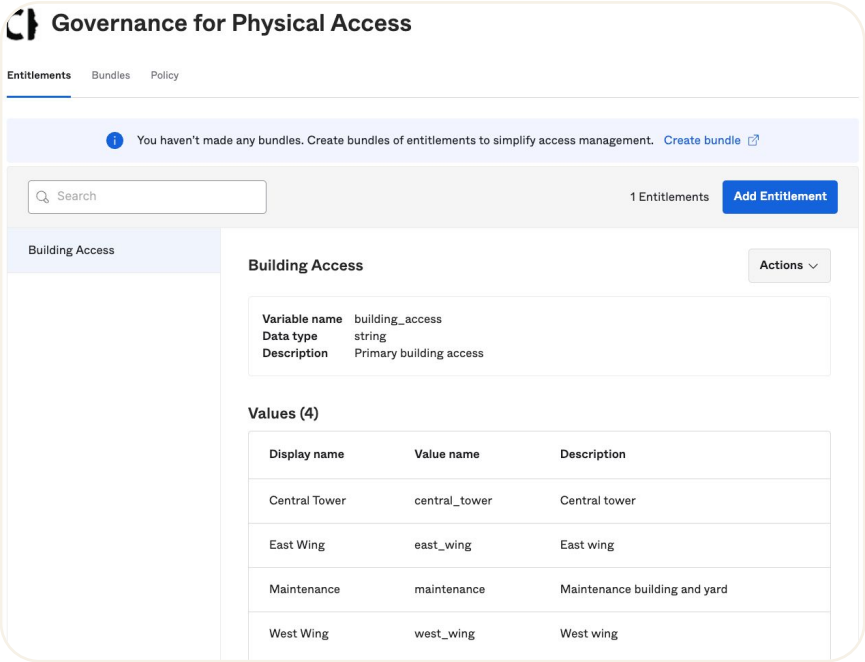
   West Wing, west_wing, West wing

   Maintenance, maintenance, Maintenance building and yard



5. Click **Save entitlement**

We have created our first entitlement (type) with values.



1. Repeat the steps above to add another entitlement, *Carpark Access*, string array data type, with the following values:

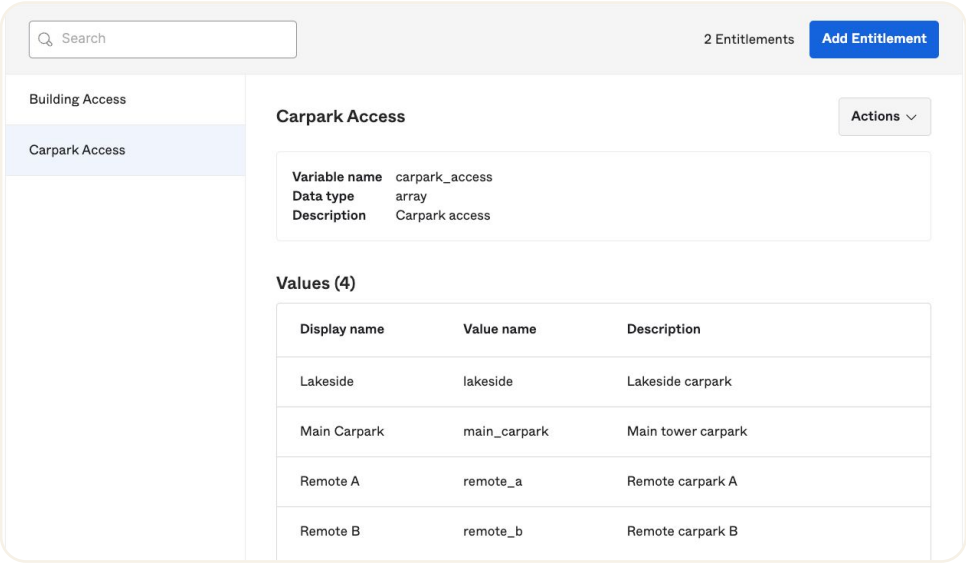
   Main Carpark, main_carpark, Main tower carpark

   Lakeside, lakeside, Lakeside carpark

   Remote A, remote_a, Remote carpark A

   Remote B, remote_b, Remote carpark B

You now have two entitlements, each with four values.



These entitlements can be used in Bundles for exposing via Access Requests, and in Policy for rule-based assignment. We will look at both of these.

Create Entitlement Policy for Dummy App

Entitlement Policies are used for automatic assignment of entitlements based on some logic. Entitlement Bundles (next section) are often used for manual assignment or via Access Requests. This is equivalent to using Group Rules to automatically assign the group membership rather than an admin assigning access to a group or a user requesting access to an Okta Group in Access Requests.

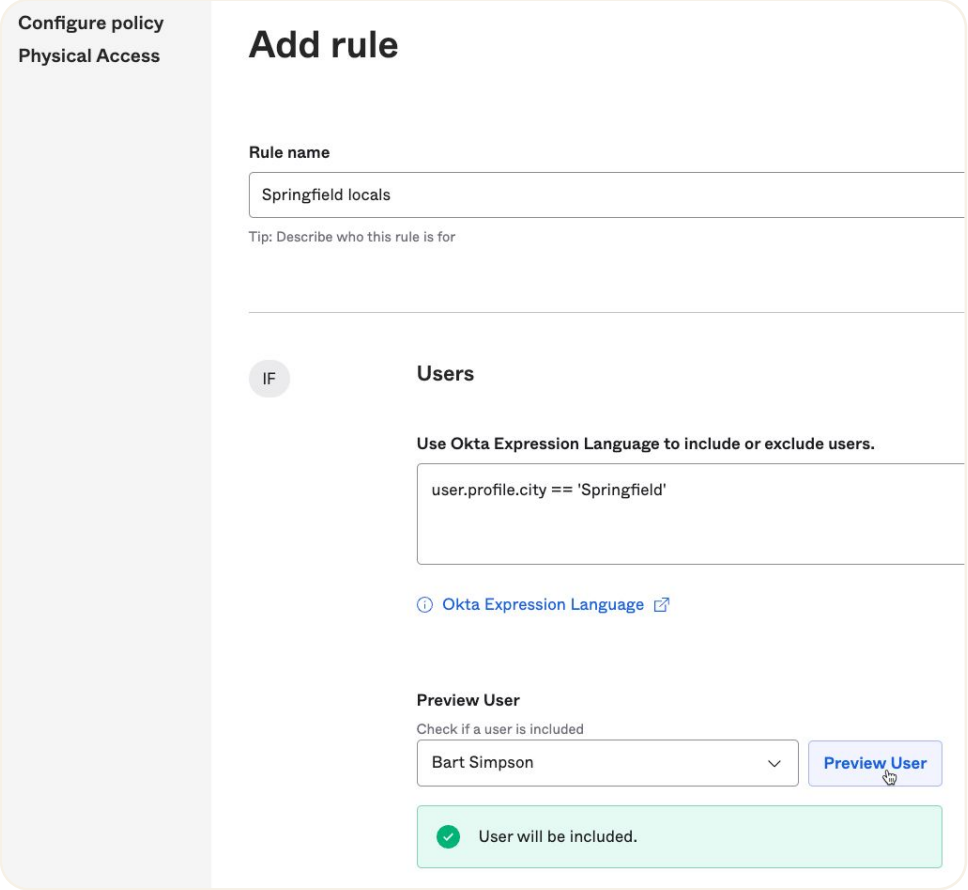
Policies are a set of one or more rules with a priority sequence of evaluation.

Create Policy Rule

We will create a policy rule to assign some default building/carpark access to all local employees (i.e. that live in the city of Springfield).

To create a policy rule

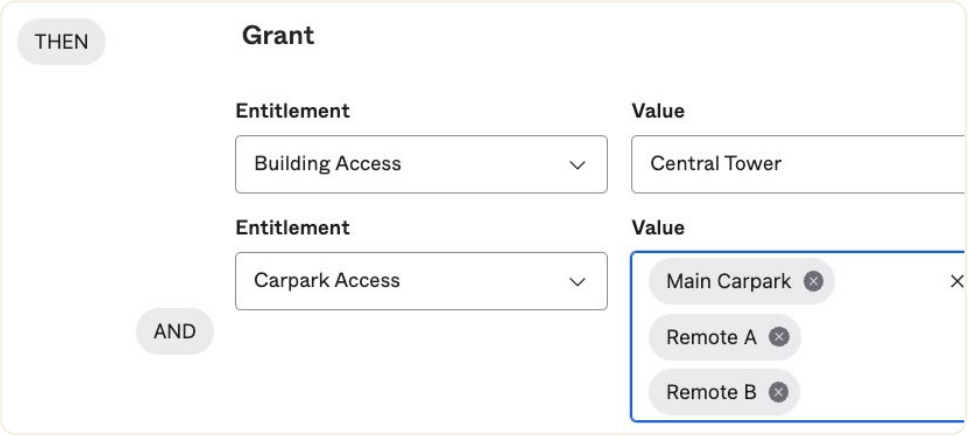
1. Select the **Policy** tab
2. Click the **Add rule** button
3. Give the policy a **Rule name** (e.g. "*Springfield locals*")
4. In the **Users** section use the following expression language (based on the user profile attribute and value you set in the first step of this guide) *user.profile.city == 'Springfield'*
5. Enter a user name for a user who was set to have the city value and click **Preview User** to test if the expression language is correct



1. In the **Grant** section add the following entitlements:

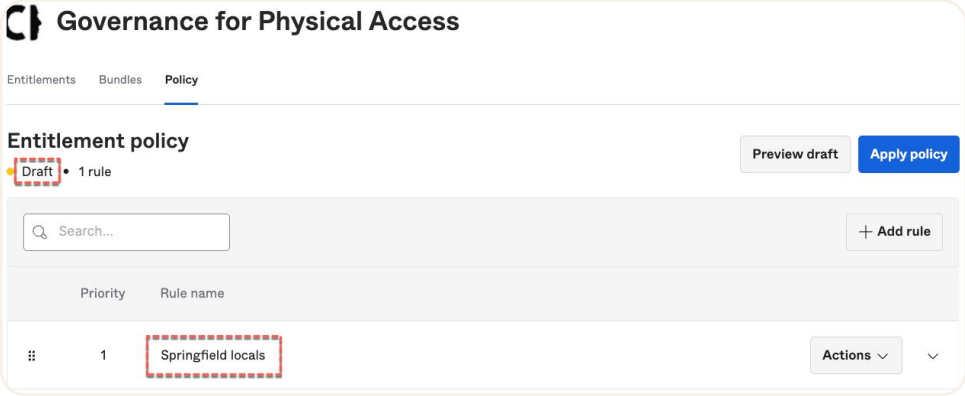
   Building Access, Central Tower

   Carpark Access, Main Carpark, Remote A, Remote B



7. Click the **Save rule** button

The new policy rule has been created in a Draft state.



We could have created multiple rules with different user selection criteria (possibly overlapping) with different Grants (again possibly overlapping) and let the engine evaluate the actual grants. If the granted entitlement is single-valued, then the value resulting from the highest priority rule will be applied. If the granted entitlement is multi-valued, then the superset of values will be applied.

This policy is not yet active. We can preview the impact of the policy, then apply it.

Preview the Policy Change

There is a policy preview function to see the impact of the change before applying the change. To run it:

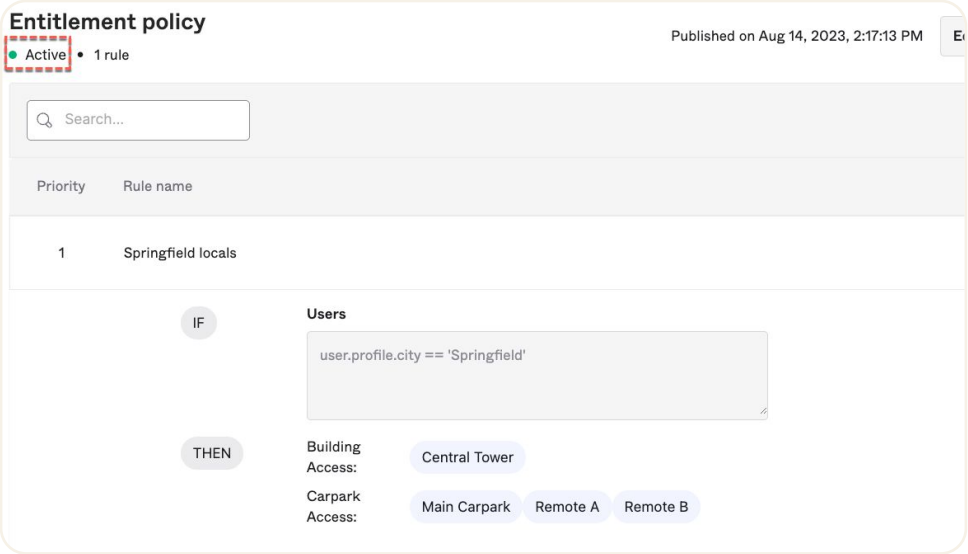
1. Click the **Preview draft** button
2. Select the **All...** option for the **Scope users** and click the **Preview** button
3. Check the email assigned to your admin user for a new CSV file
4. Open the CSV file and have a look at the results

Apply the Policy Change

To have the policy evaluated and entitlement assigned to users, you need to apply it.

1. From the **Entitlement policy** page, click the **Apply policy** button
2. Note the Apply policy dialog and how it recommends previewing first. Click the **Apply policy** button

The policy should show as active.



Note that you can click the arrow to the right of the rule to expose/hide the details of the rule.

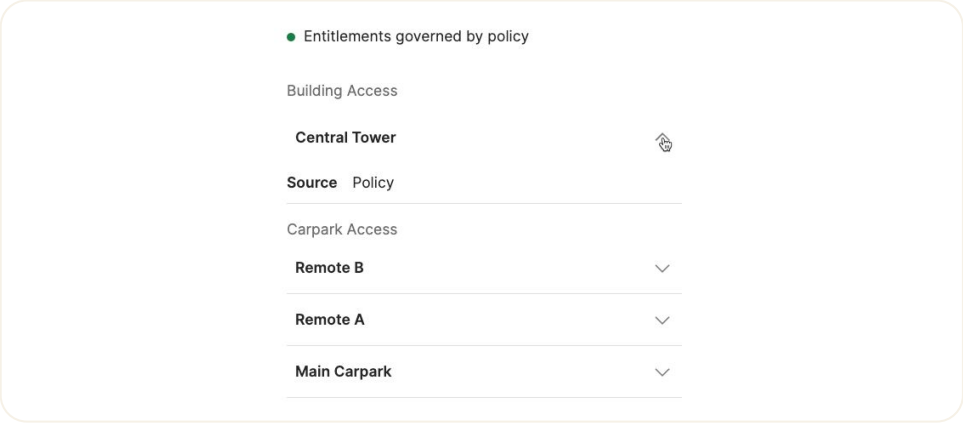
3. Check the policy has worked by going **Back to application**

4. Find one of the users who had the *user.profile.city* value set to match the policy rule. Does the user have “Custom” selected for the Entitlement? This means the policy has not applied and needs to be manually re-evaluated. If they have entitlements you can skip the Create Entitlements Bundle section

Select the three vertical dots icon to the right and View entitlements.

Note that the user has no entitlements.

5. In the window that slides out from the right, click the Edit button and then the Revert to policy button. When it applies policy, you should see the entitlements in your policy applied to the user.



This is all we will do with entitlement policy for now. Next we will look at entitlement bundles.

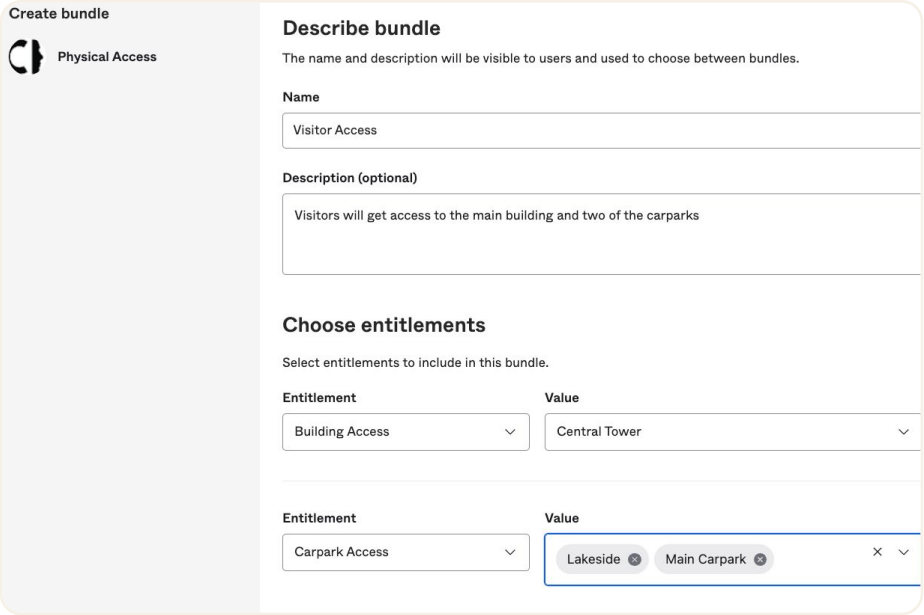
Create Entitlement Bundle for Dummy App

Entitlement bundles are a collection of entitlements for a specific use. For example you could create bundles to represent roles or job functions. Bundles can be exposed via Access Requests. Bundles can mix entitlements within an application, but not from multiple applications.

In this case, we will create a bundle for site visitors to give them badge access to two of the carparks and the main building.

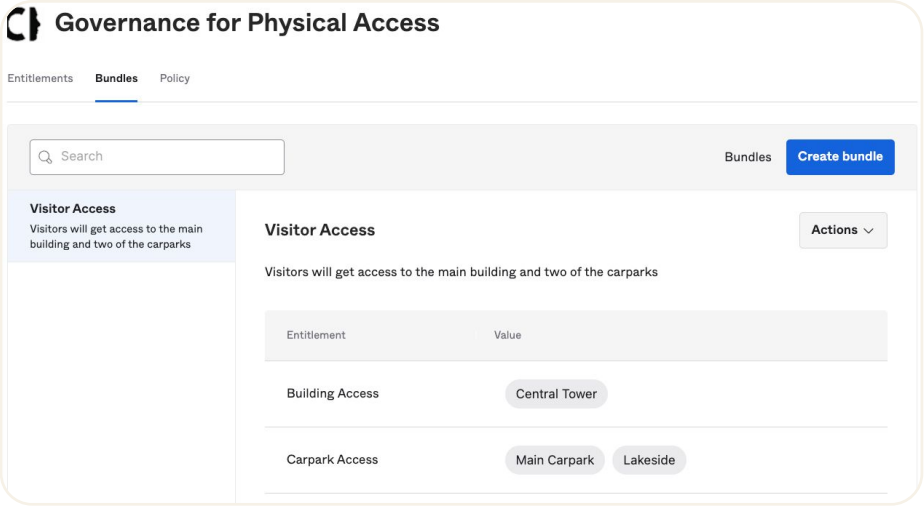
To create a bundle:

1. Select the **Bundles** tab
2. Click on the **Create bundle** button
3. Give it a **Name** (such as Visitor Access) and optionally a **Description**
4. Add the following entitlements and values:
   Building Access, Central Tower
   Carpark Access, Lakeside and Main Carpark



5. Click the **Create** button

The bundle is now created and ready to use.



The next step is to expose the bundle via Access Requests.

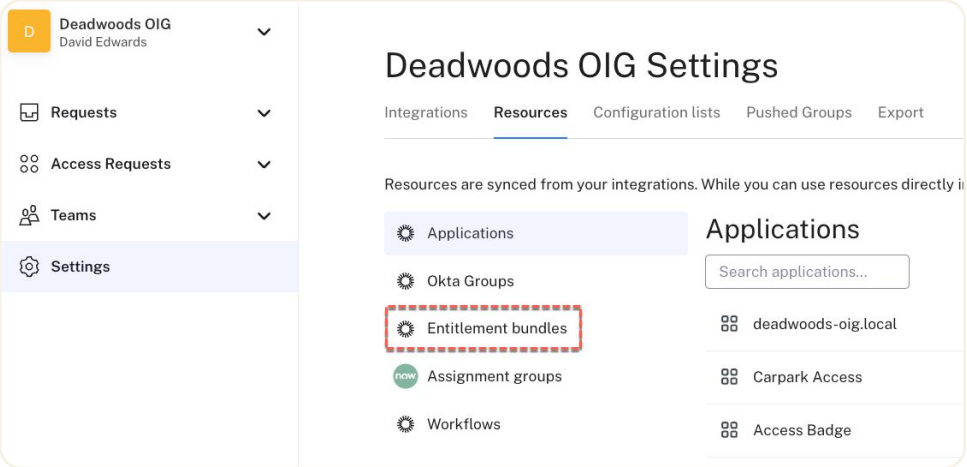
Entitlement Bundles in Access Requests

Entitlement Bundles can be resources requested in Access Requests just like Applications or Okta Groups. In this section we will expose the new entitlement bundle via a Request Type.

Check Access Requests Configuration for Entitlements

As this is the first time we've used Access Requests with Entitlements, we need to check that it is configured correctly.

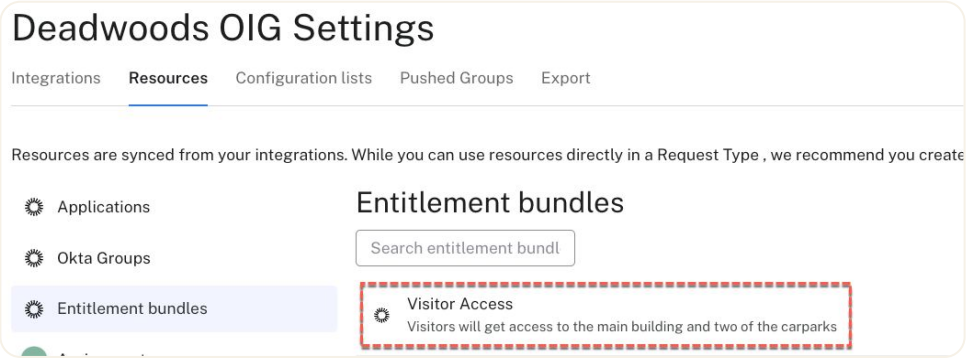
1. Go into your Access Requests instance as an administrator
2. Go to **Settings**, **Resources** tab



If you don't see Entitlement bundles under the list of Resources, the Access Requests app has not been configured correctly for Entitlements (it's probably missing the feature flag) and you should check with whomever setup the environment.

3. Click the Entitlement bundles Resource. You may or may not see anything depending if the sync cycle has run since you created the bundle.
4. Click the Manage Access button and ensure the appropriate Access Request Teams have access (you should atleast have IT enabled). Click Save.
5. If you don't see the new bundle, click the Update now button. This will queue the refresh job.

6. Refresh the view until you see your entitlement bundle.

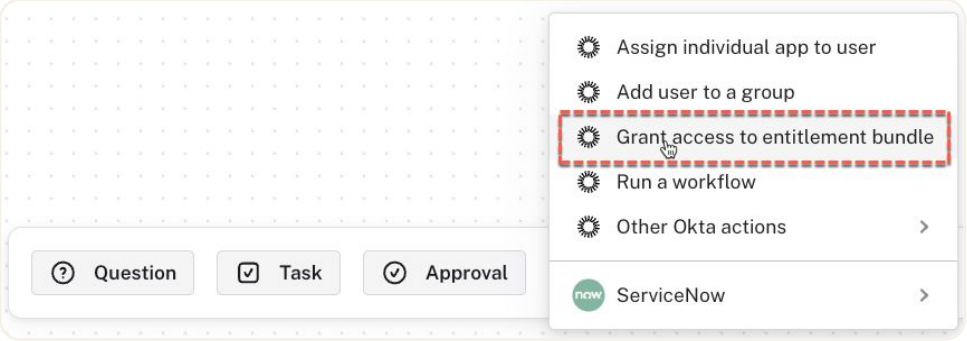


With the entitlement bundle showing, you can now create a new Request Type for it.

Create Request Type

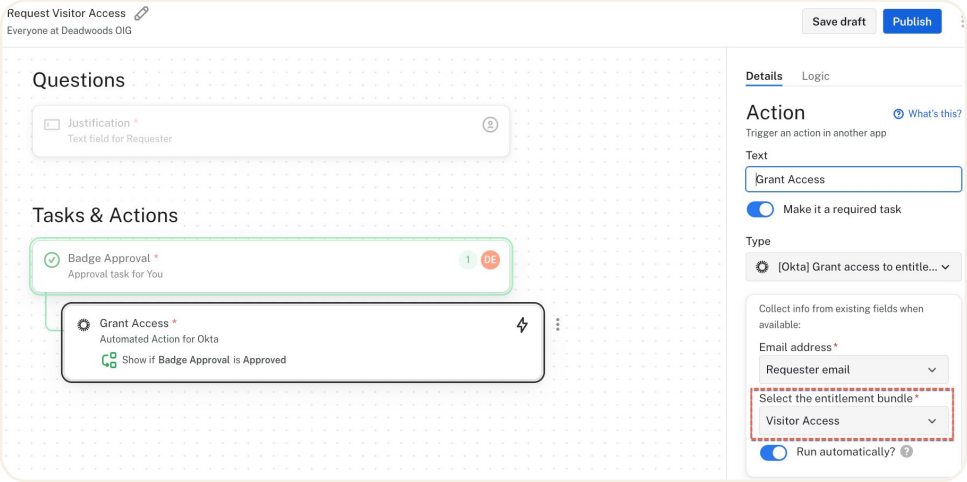
We won't walk through every detail of creating a new Request Type as familiarity with this is expected (it was covered in detail in the earlier sections of this guide). You want to create a new Request Type that will prompt the user for a justification, run a single approval step (perhaps assigned back to your manager ID) and then run the action to assign the user to the new Entitlement Bundle.

The last step has a new Okta Action – **Grant access to entitlement bundle.**



When you add that Action you select the entitlement bundle to add the user to.

Note that currently there is no corresponding "Revoke access from entitlement bundle" action.



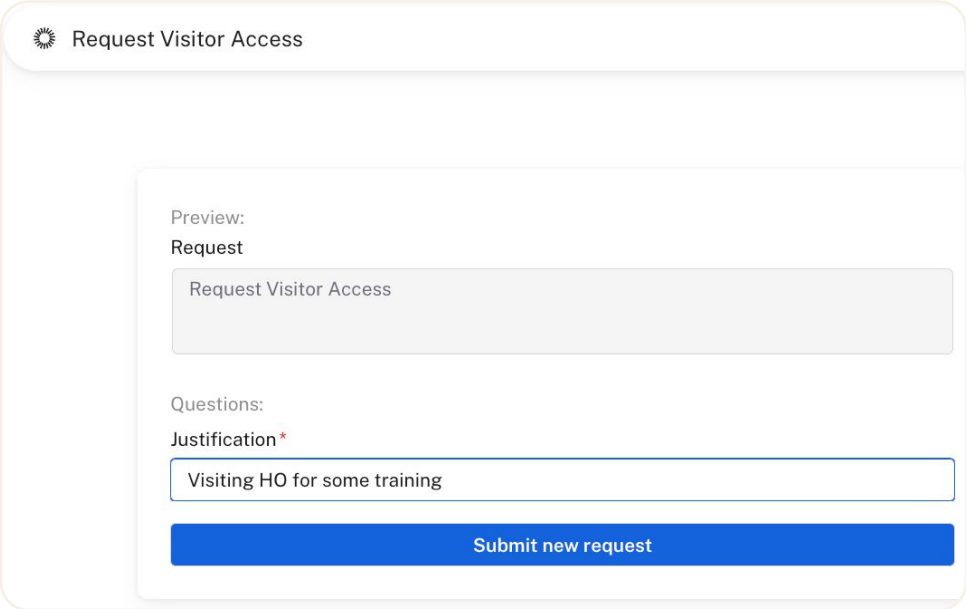
1. Create a **Request Type** similar to the above.
2. Add question, approval and the entitlement bundle action
3. **Publish** the Request Type

Next we will test the new Request Type.

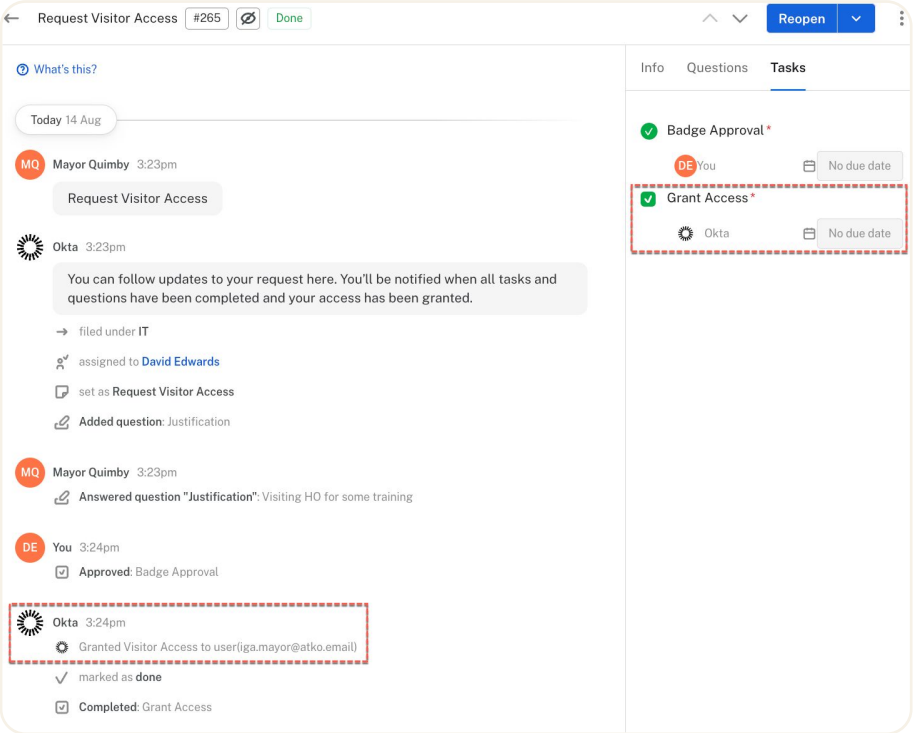
Request Access to Entitlement Bundle

To test the new Request Type:

1. As a user who can can request access, go into Access Requests and select the tile representing the new Request Type
2. Enter the justification and submit the new request

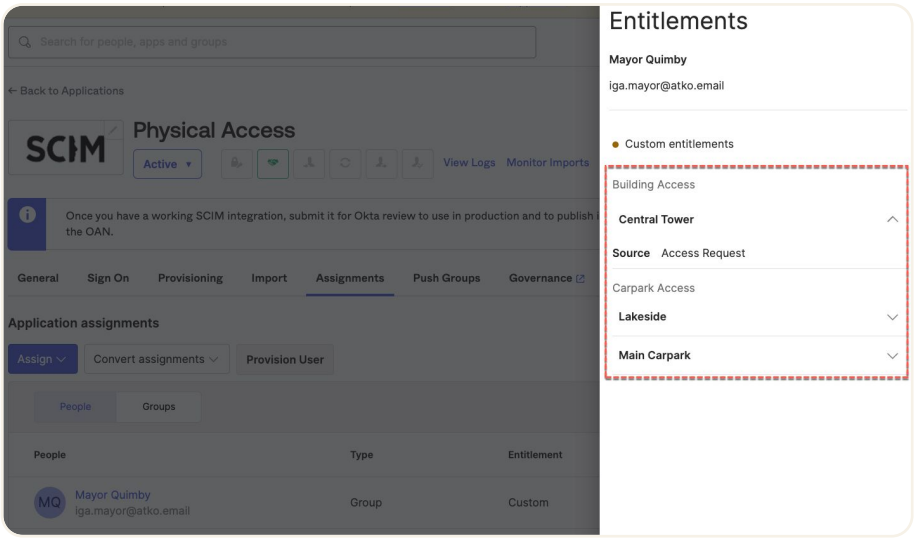


3. As the reviewer/approver, go into **Access Requests** and approve the request
4. As the administrator check that the request completed successfully



Note that the Access Request UI to view details of a request has changed and will look different to the above, but the results should be the same.

5. As the administrator return to the Application Assignments view, find the user who request access, click the three vertical dots and click View entitlements



You should see the entitlements in the entitlement bundle added to the user. If you did not, there may have been a problem with the Request Type.

This was a trivial example of assigning entitlements via a bundle in Access Requests. You could apply all variations of steps in a Request Type as you would for other requests.

Access Certification with Entitlements

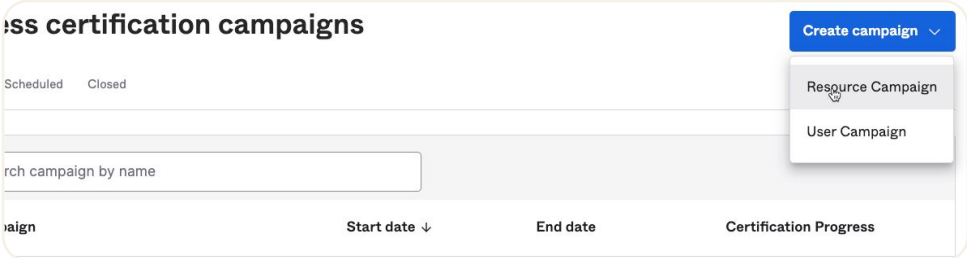
Entitlement assignment can be recertified in Access Certifications in the same way that Group or Application assignments are. Both entitlements (e.g. assigned by policy) and entitlement bundles (e.g. assigned through Access Requests) can be reviewed in an Access Certification campaign.

The section walks through the creation, launching and execution of a campaign. It is assumed that the reader is familiar with this, so the steps will focus on the differences for entitlements.

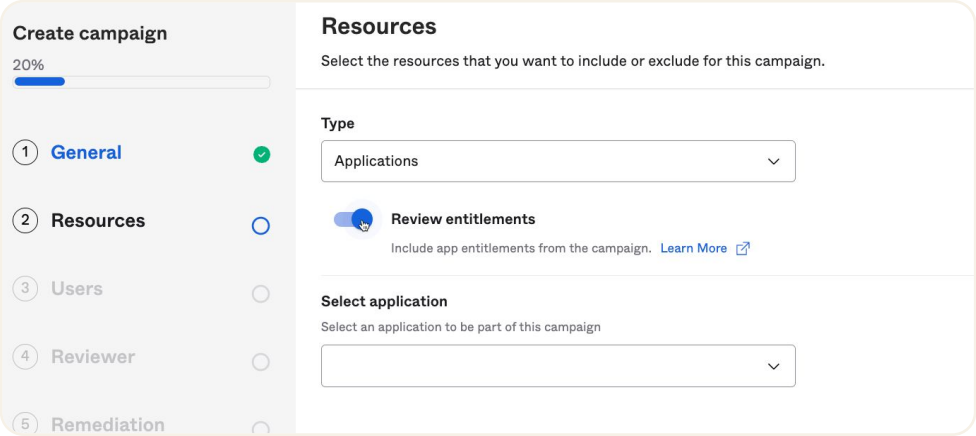
Create a Campaign

The steps to create a campaign are the same as for any other resource campaign (which was covered in an earlier section of this guide):

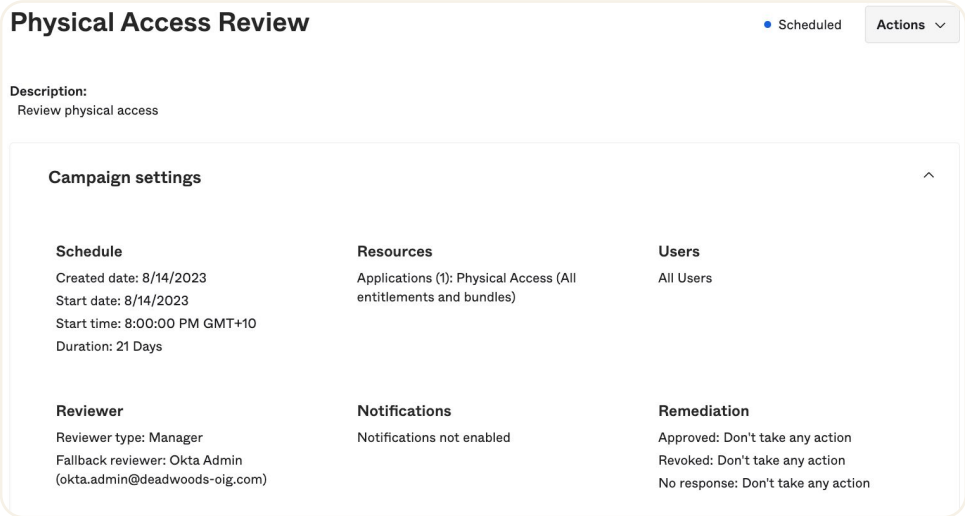
1. From the **Access Certifications** page, select the **Create campaign** button and the **Resource Campaign** option



2. On the **General** tab, give it a name and any other details you want to add
3. On the **Resources** tab, select **Type = Applications** and enable the **Review entitlements** option. This is how we include entitlements in the campaign as entitlements are always tied to the resource. You can select specific entitlements and/or bundles.



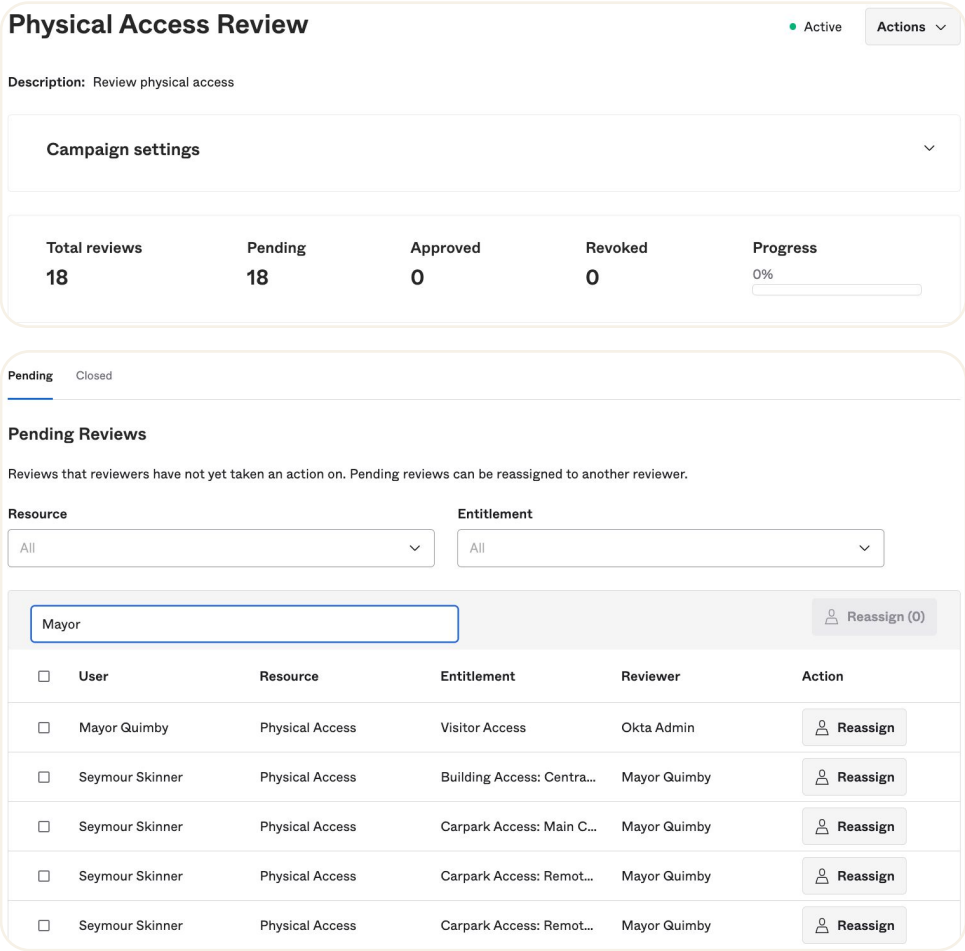
4. On the Users tab select the user scope as you would normally do
5. On the Reviewer tab setup the reviewers
6. On the Remediation tab, select whether you want access to be removed when the reviewer clicks Revoke or when the campaign ends.
7. Submit the campaign and check the details.



You can go back and edit any settings or launch the campaign.

Launch the Campaign

Launch the campaign and then look at the reviews.

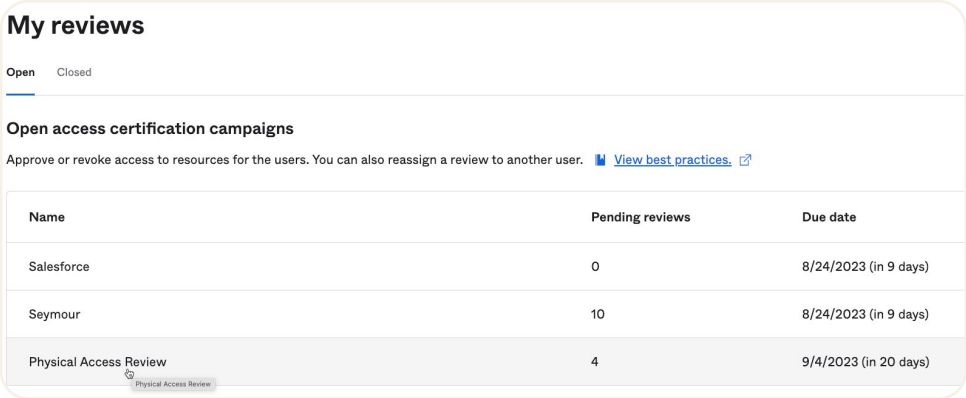


Notice that this view shows both the Resource (i.e. the application) and the Entitlements. The first user shown above has the entitlement bundle ("Visitor Access") shown, whereas the second user has specific entitlements applied and the view is showing the entitlement name and value (more detail in the slide-out panel).

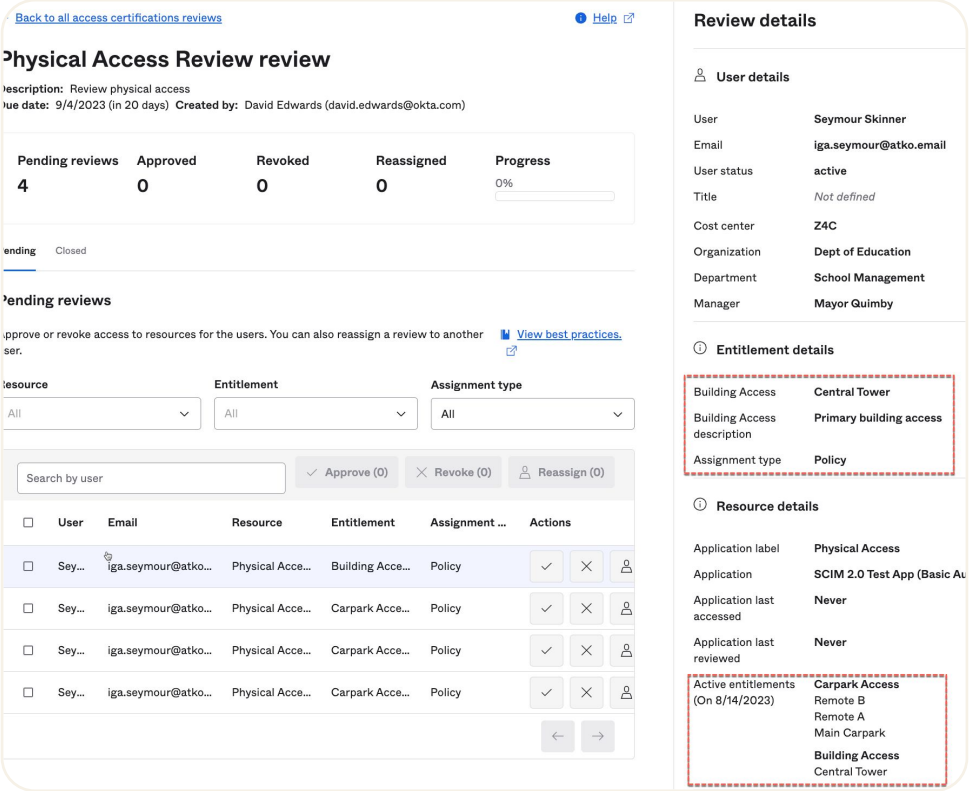
Execute the Campaign

Campaign execution is as for any other campaign:

1. As one of the reviewers in your campaign (e.g. one of the managers) go into their Okta dashboard and open the Okta Access Certification tile.
2. Open the entitlement campaign you just launched

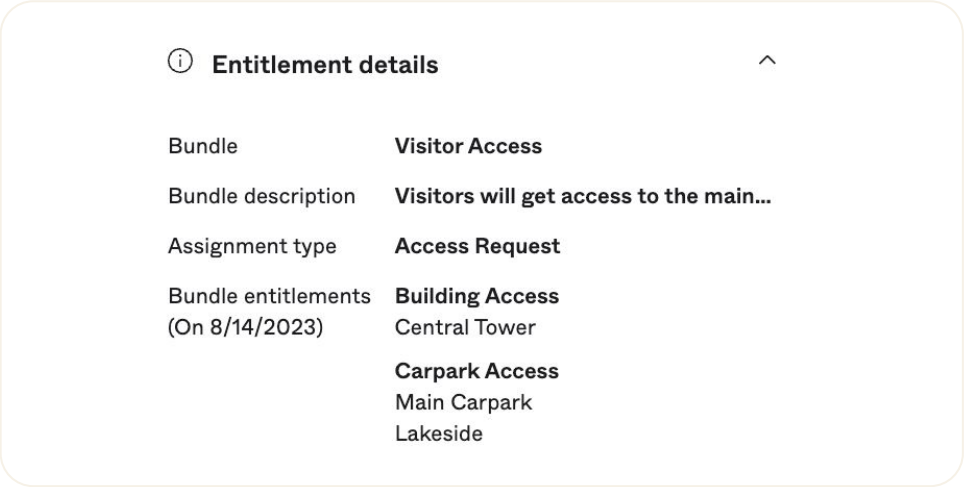


3. Click on one of the users to see the details of a specific review



Note the content in the Entitlement details section. If this is an entitlement assigned to the user, the entitlement name is used with the value, along with the entitlement description and value description. The Assignment type indicates it was assigned by an entitlement policy. You can also see the complete set of entitlements at the bottom of the panel.

If this was an entitlement bundle, the slide-out panel would be slightly different:



The Entitlement details section now shows the bundle name, description, Assignment type (in this case Access Request) and details of what the bundle grants.

4. You can work through the review, but as there aren't any changes applied there's no point.

There are some points to highlight here.

- As mentioned the entitlements are tied to an application - you cannot currently run an entitlement campaign without selecting the corresponding application
- You can only review entitlements for one application in a single campaign
- Entitlements and entitlement bundles are shown, but if an entitlement is granted via a bundle, you will only see the bundle not the individual entitlement.
- If you select "Remove access from user" in the Remediation section, and OIG cannot remove it due to policy violations, it will flag that review for manual remediation. This is the same as trying to remove other access that would violate policy (like removing app access assigned via a group / group rule).

This concludes the first section of this lab guide. We have used a dummy app to focus on the core concepts of entitlement management, with entitlements, bundles, policy, access requests for bundles and access certification of entitlements and bundles.

Entitlement Management for Microsoft Office365

In this section of the lab guide we explore Entitlement Management for Microsoft Office 365 (O365). The two entitlements are Licenses (technically apps in licenses) and Roles - both multi-valued attributes.

> Note that with O365 and Okta, any user assigned to the O365 app in Okta MUST have a valid O365 license assigned to them. If you perform a manual Import and there are assigned users without Licenses they will be unassigned from the O365 app in Okta. This is not new with Entitlement Management - this is how the O365 OIN integration has behaved for some time.

The steps to set up and use Entitlement Management for O365 is basically the same as for other Entitlement Management-enabled applications:

1. Setup the application in Okta for SSO, Entitlement Management and Provisioning,
2. View the imported entitlements,
3. Create Entitlement Policy and apply the policy, and
4. Create Entitlement Bundle and OIG Access Requests Request Type and test

As the previous section provided a lot of detail on each step, the following sections will be brief where concepts and steps are repeated. See earlier sections (“dummy app”) for more detail on each step.

Set Up O365 Application for Entitlement Management

This section of the lab requires a working Office 365 tenant. We won’t cover the steps here to get an instance and configure it. The Entitlement Management integration will leverage the License and Role values in that O365 tenant - you do not need to add any entitlement values in O365.

Set Up O365 as an SSO App in Okta

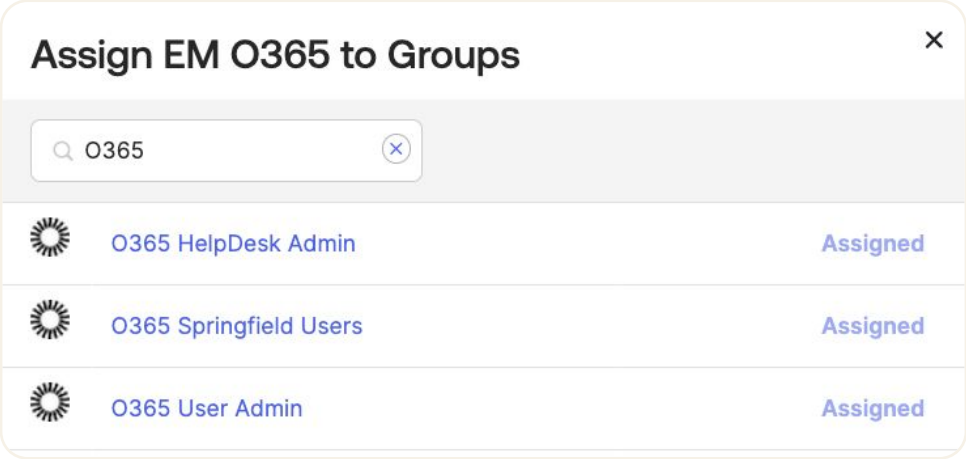
If this is a new O365 app in Okta, you will need to create it. Using the **standard OIN application**:

1. Create the new O365 app in Okta

Create O365 Group(s) and Assign

If this is a new O365 app in Okta, you will need to create it. Using the **standard OIN application**:

2. Create one or more Okta groups for the users who will be assigned to the app
3. Assign the group(s) to the app

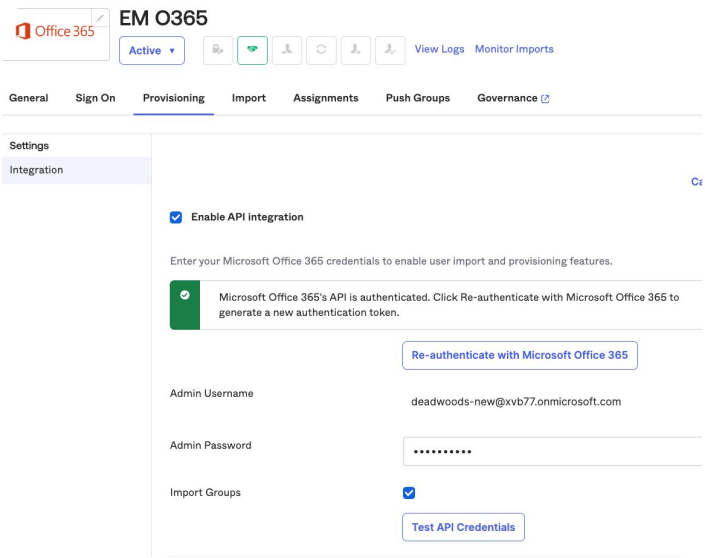


Enable the Governance Engine

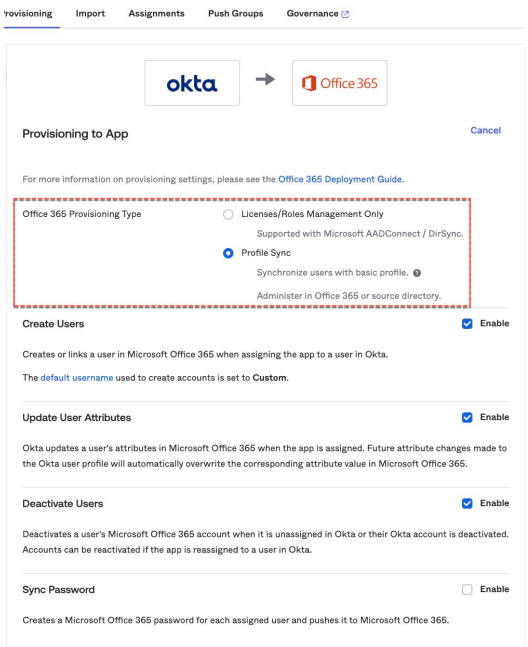
4. As before, enable Identity Governance for the app and refresh the page until the new Governance tab appears

Enable Provisioning

5. **Enable API integration** to enable Provisioning for the app as per the product documentation (including accepting the scopes)

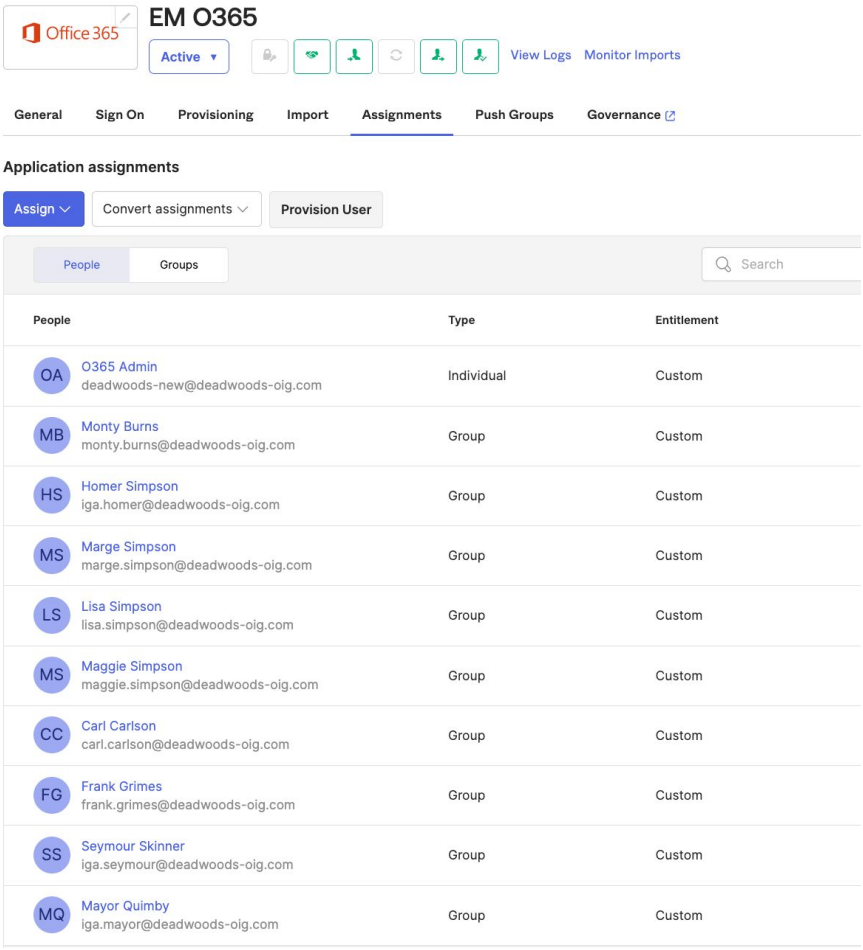


6. Note that the O365 integration with Entitlement Management has a reduced set of Provisioning to App options. Select the **Profile Sync** option and turn on the **Create**, **Update** and **Deactivate** options



Do not run the Import function. The way the Office365 integration is built, the Import will remove any existing user assignments. There is an automatic import tied to enabling provisioning that will pull in the entitlements and values.

1. Go to the **Assignments** tab and confirm you have the new (Entitlement Management) assignment look.



Note that the entitlements are read-only for the connected system. The integration is coded to pull in the entitlements from O365 (licenses and roles) and there is no mechanism to modify entitlements in Okta and push them down to O365.

Now that we have entitlements and users, we can create entitlement policies.

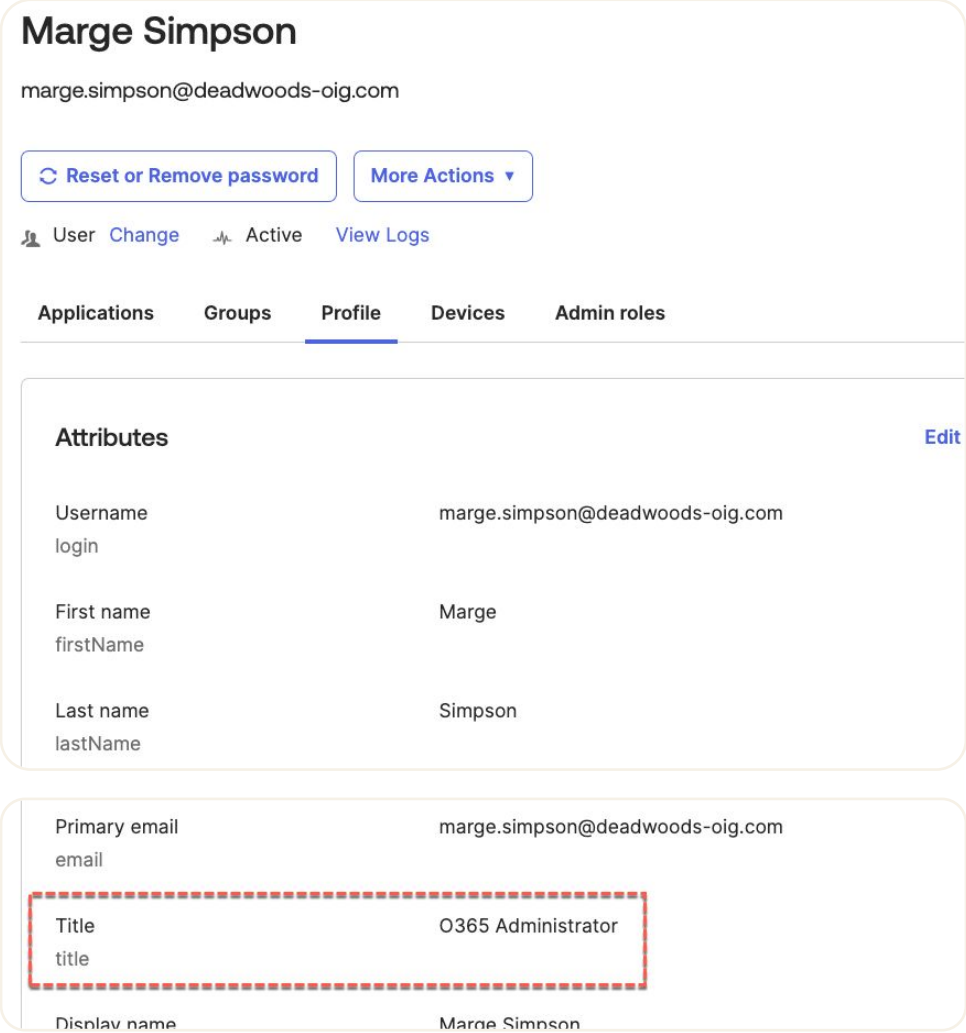
Create Entitlement Policies for O365

In this section we will create entitlement policies for ordinary users and administrators.

Okta User Set Up for Policies

We need to make one user have an administrator title for the admin policy.

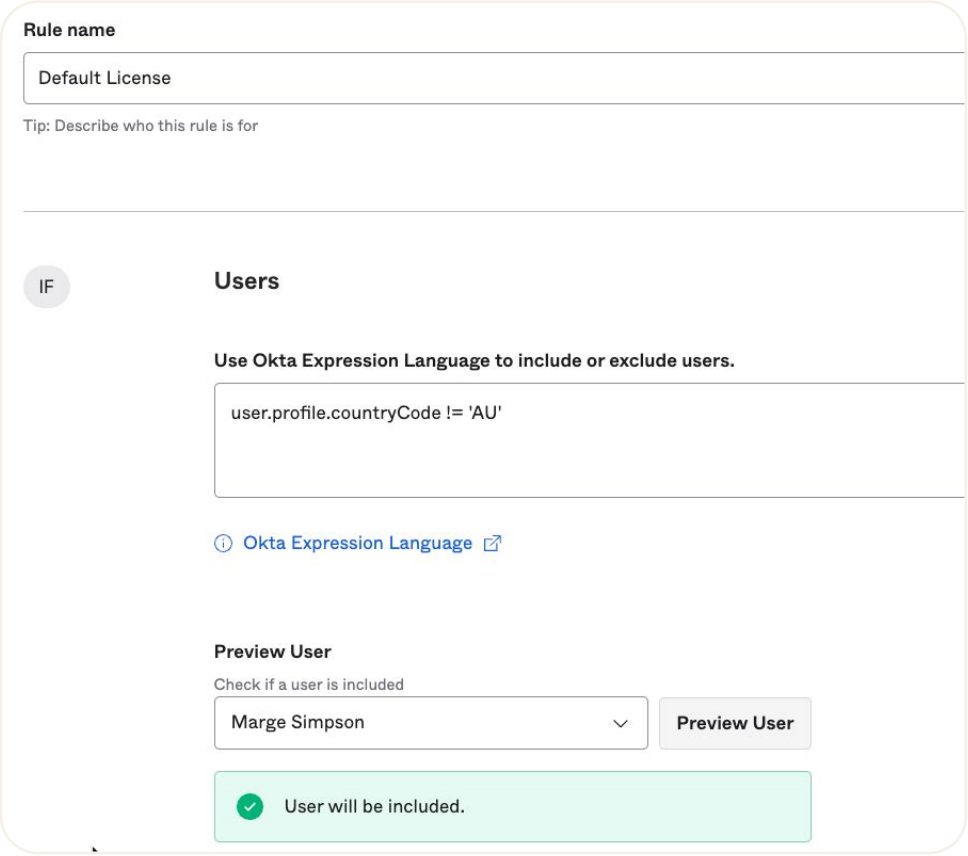
1. Select a user that is assigned to the O365 app and modify their Title to be "*O365 Administrator*" and save the change



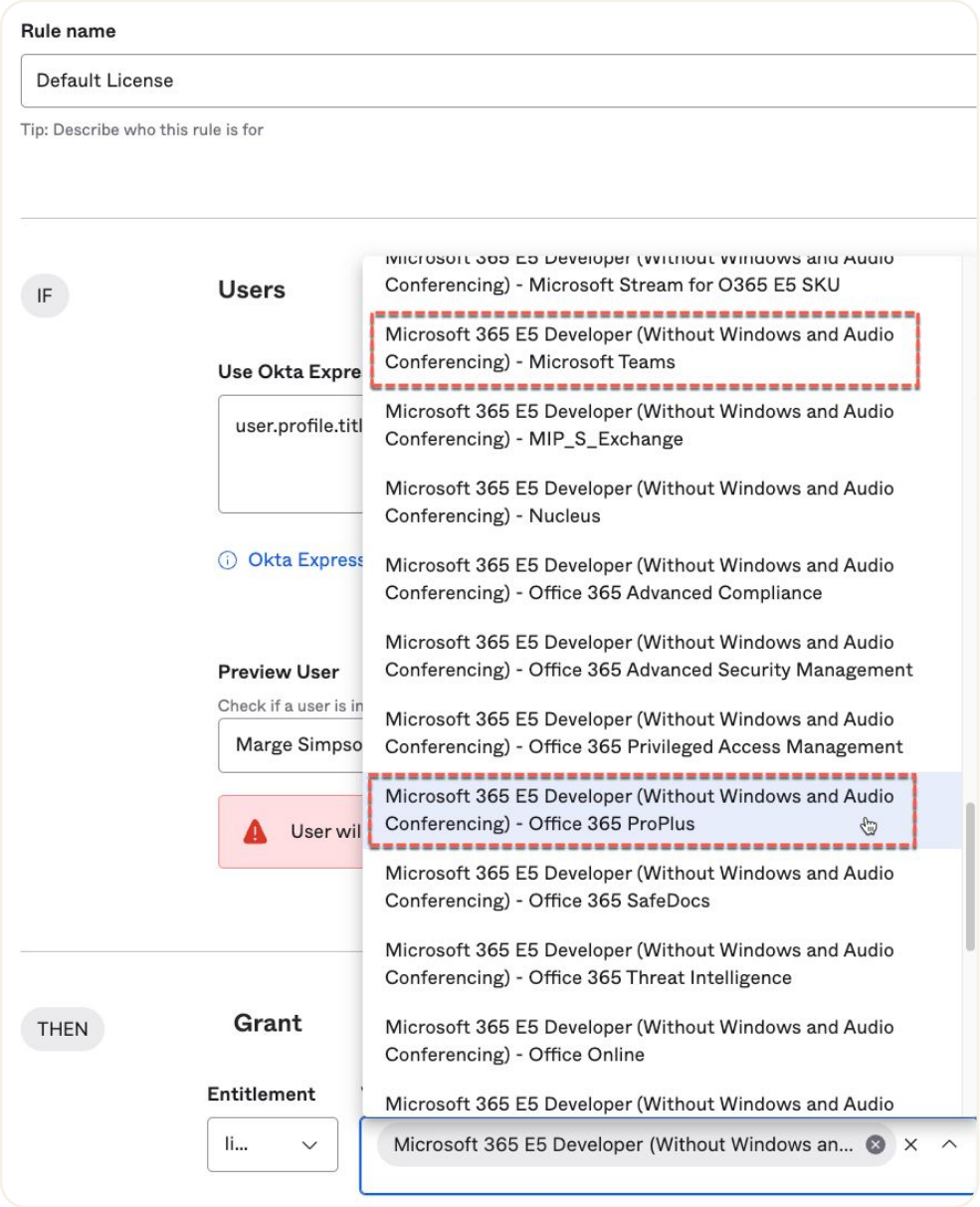
Create a Default License Policy Rule

We will create two policy rules, one for all users to grant basic licenses and one for the admin user.

2. Create a new policy **Rule** for the *Default License* and set the IF clause to something that will catch all users (like *user.profile.countryCode != 'AU'*).

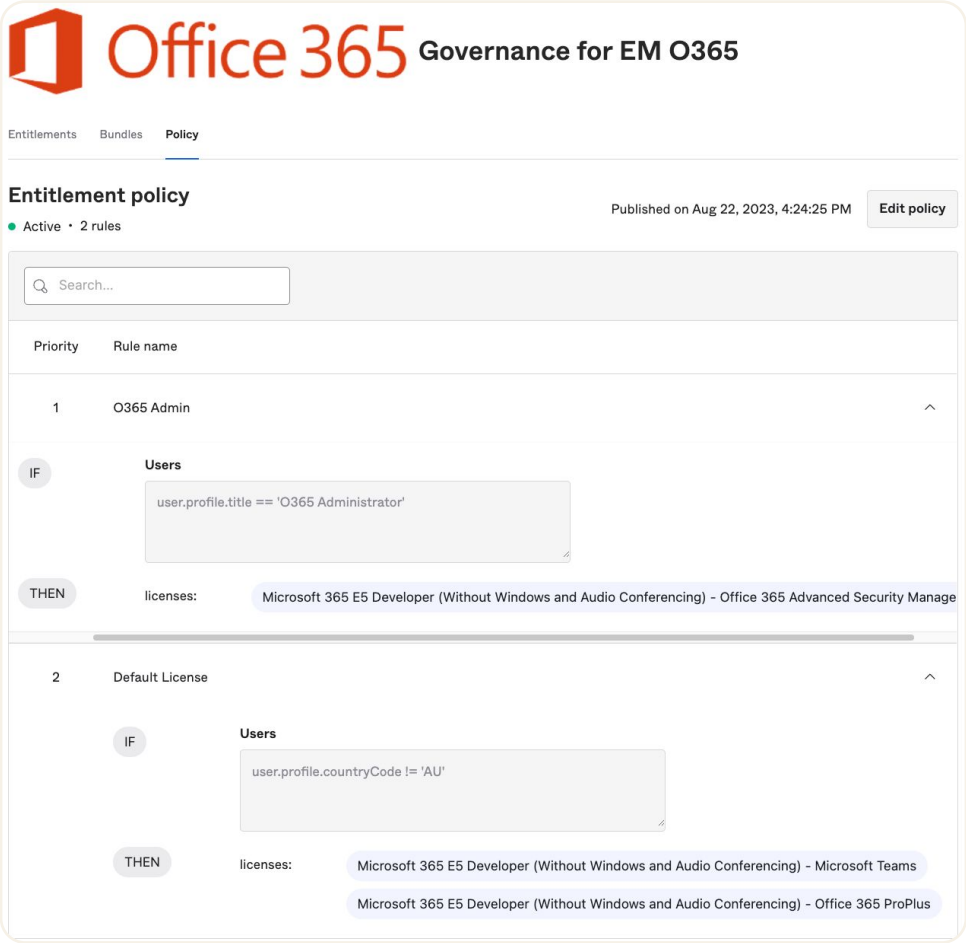


3. In the **THEN** clause assign some standard O365 licenses, such as **Microsoft Teams** and **Office 365 ProPlus** (it doesn't really matter which ones you select, you are just showing that a default rule can be appled) and **Save** the rule.



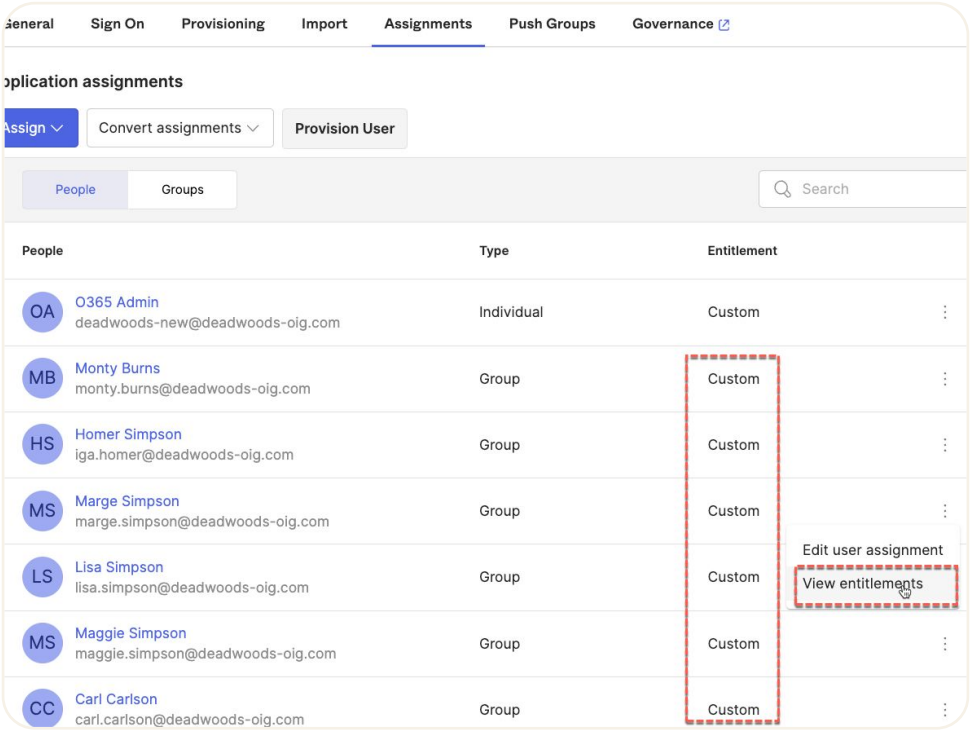
4. Create a second **Rule for** the *O365 Admin,* with an **IF** clause of *user.profile.title == 'O365 Administrator'* (which you set on one of the users earlier). In the **THEN** clause assign another O365 license. **Save** the rule. You could add a role too, but we'll do that later.

5. Check the two rules and Apply the policy

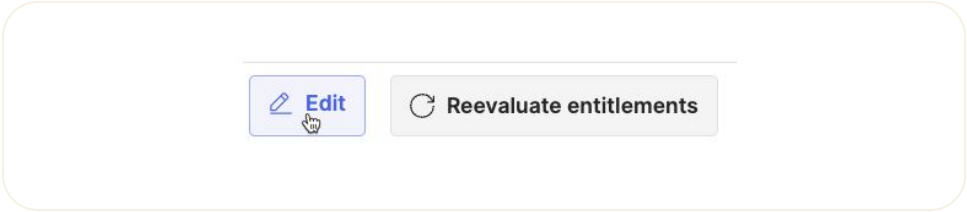


6. Go back to the application **Assignments** and look at the users. You may have noticed that the users have an entitlement of Custom. This may be because the policy hasn't run yet or because of how Entitlement Management operates atm.

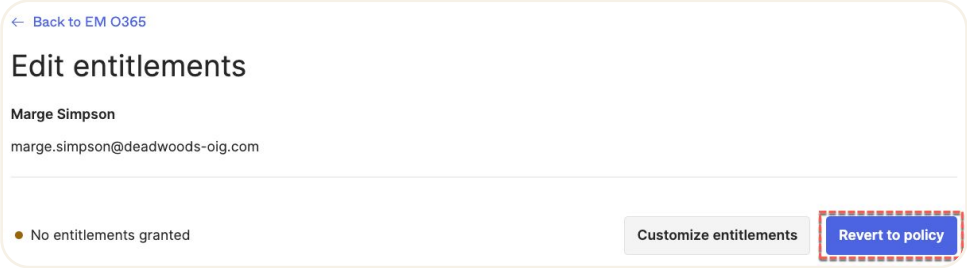
7. To apply the policy for a user, select the three vertical dots, and select **View Entitlements**. I'd suggest doing this for the admin user to confirm both policies are applied



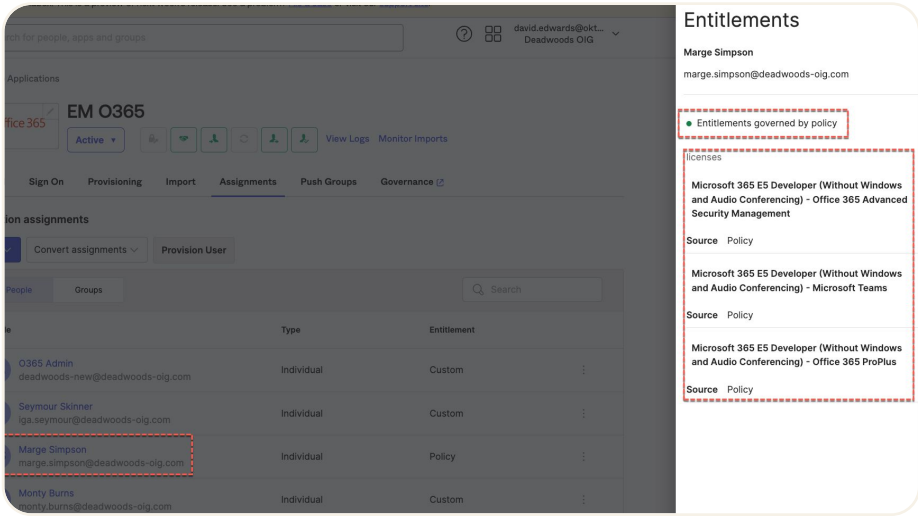
8. Then click **Edit**.



9. On the **Edit entitlements** screen, click the **Revert to policy** button



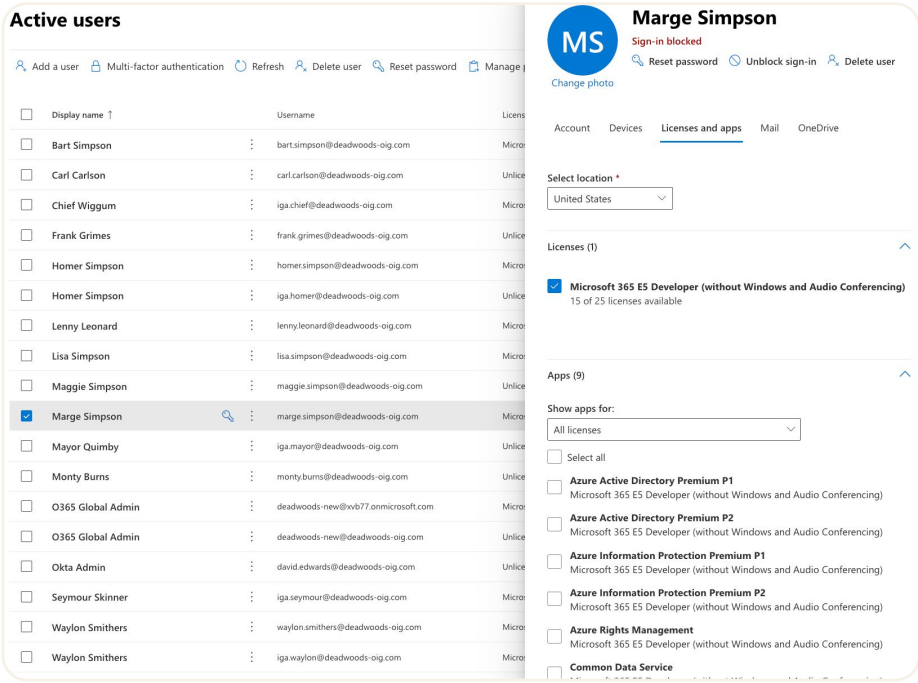
10. You can then see that the Entitlement for the user is marked Policy now and you can see the policy entitlements applied

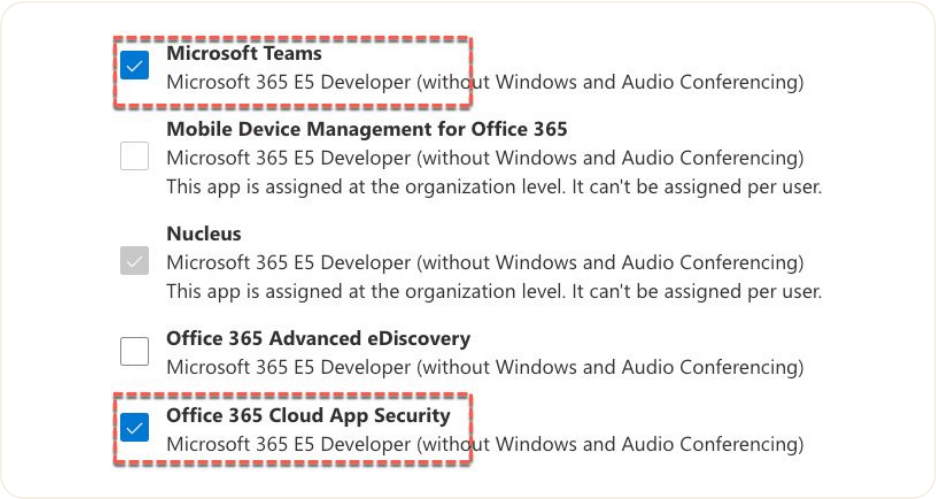


11. As these users were added prior to enabling Provisioning, you will need to use the **Provision Users** button on the **Assignments** tab.

12. If you check the System Log you should see events for updating the user and provisioning changes to the application.

13. Go into Office 365 and check the admin user has the right Licenses. Note that there may be one license (e.g. Microsoft 365 E5 Developer…) but the individual licenses you applied via policy will be under the Apps heading. There will be more than what you applied as O365 adds others in by default.



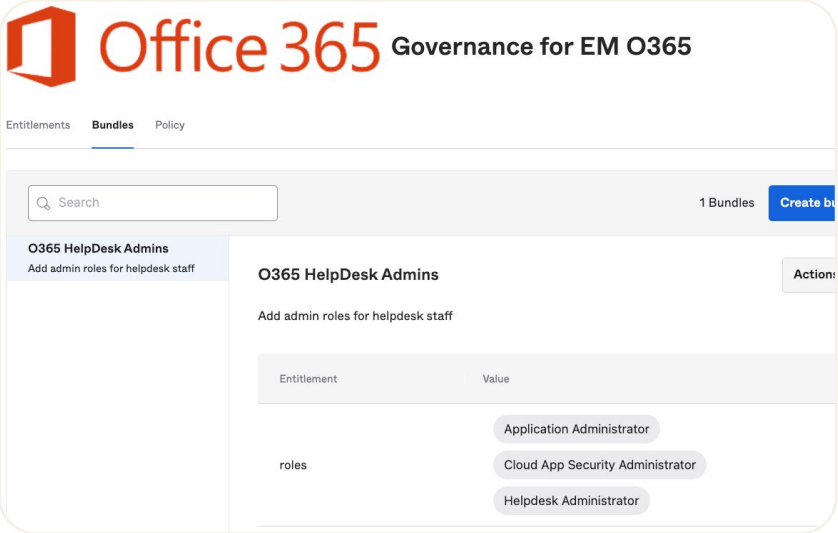


This completes the entitlement policy steps.

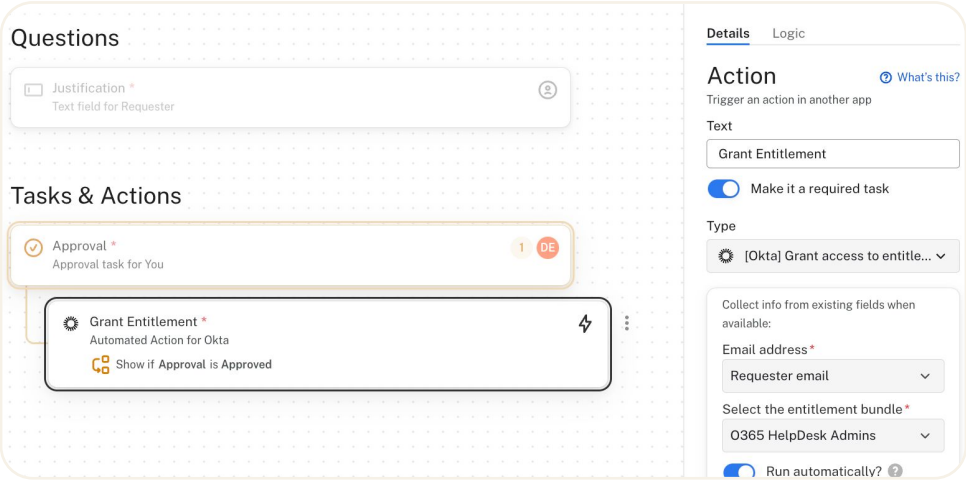
Create an Entitlement Bundle and Request Type

In this section of the lab we will use the O365 roles to create an entitlement bundle and use it in Access Requests. This is similar to the bundles created earlier. If in doubt about what you need to do, refer back to the Entitlement Bundle and Request Type section for the Dummy App.

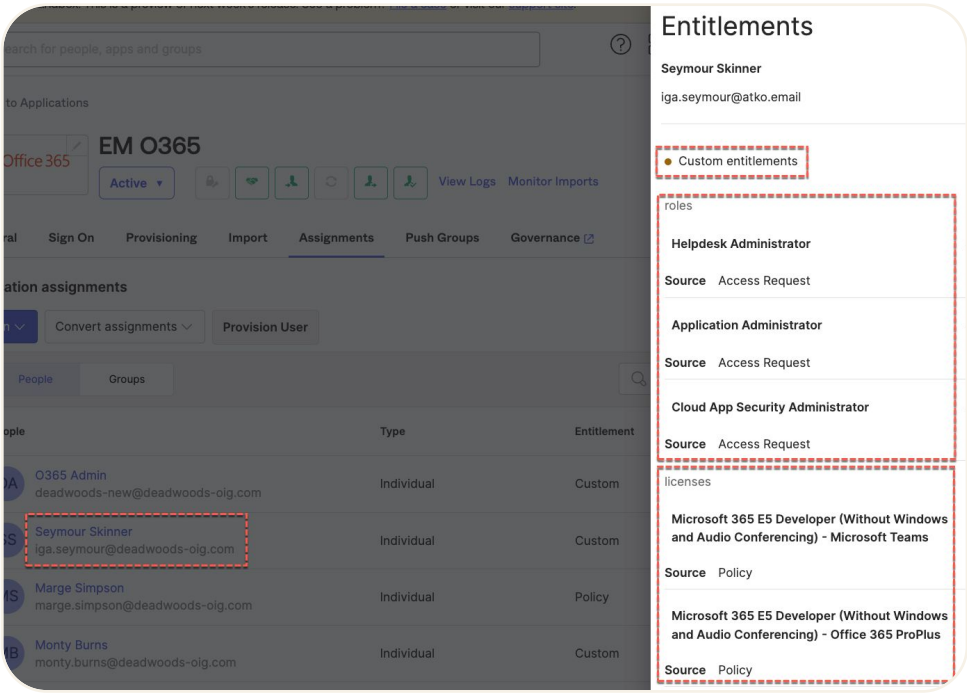
1. Create a new bundle called *O365 HelpDesk Admins*, and assign some O365 **roles** to it (it doesn't really matter which ones). **Save** it.



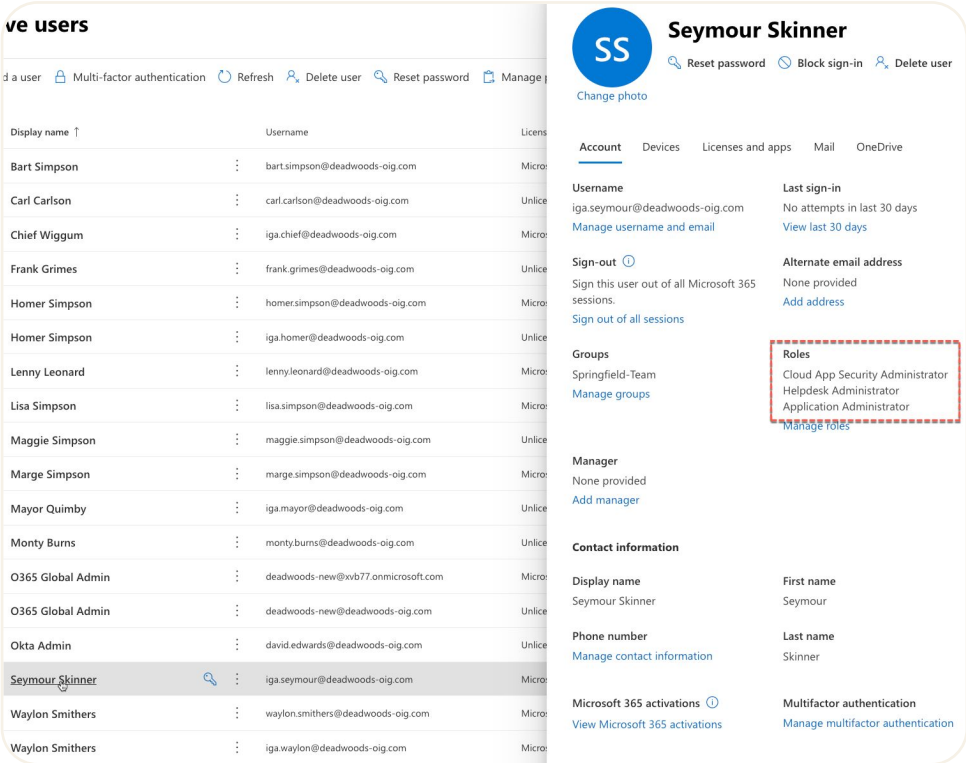
2. Go into **Access Requests** and refresh the **Entitlement bundle** resource list (in **Settings**)
3. Clone one of the earlier Request Types and change the name of the Request Type. Change the action to call this new entitlement bundle.



4. As a user assigned to the O365 application in Okta, request the new access
5. As the reviewer, approve the access request
6. As the administrator check that the request completes successfully
7. Go back into the application in Okta, and look at the Entitlements for the user who requested the access. You should see the roles you assigned via the bundle as well as the default licenses from the policy.



8. Go into O365 and confirm that user has the roles assigned correctly



This concludes the entitlement bundle and access request portion of this lab and the Office365 section of this document.

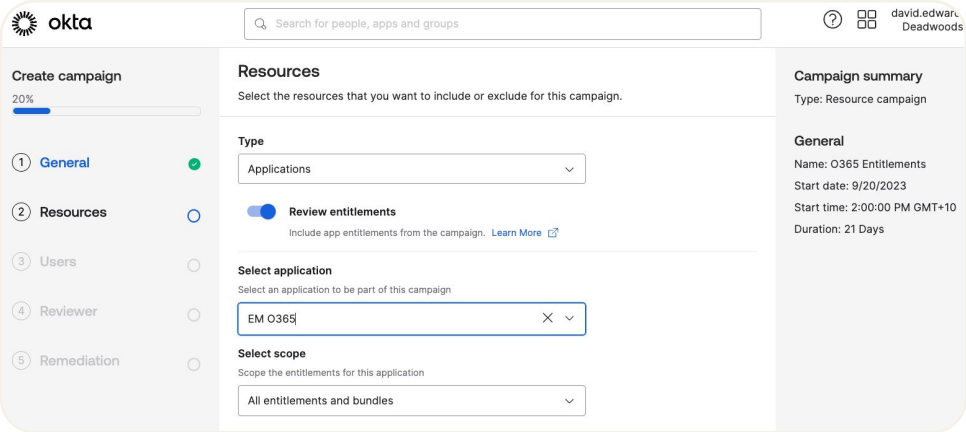
Create and Run an Access Certification Campaign

The steps to build and run an entitlement access certification campaign is the same as for the earlier dummy app, so the following sections won't have the same level of detail as the earlier example.

Create and Launch a Campaign for the O365 Application

Create a new campaign:

1. In **Access Certifications** create a new **Resource Campaign**
2. On the **General** page, give it a name and description. The other fields can be left.
3. On the **Resources** page, select **Type** of **Applications**, enable the **Review entitlements** option, enter the name of the O365 app, and leave the **Scope** as default



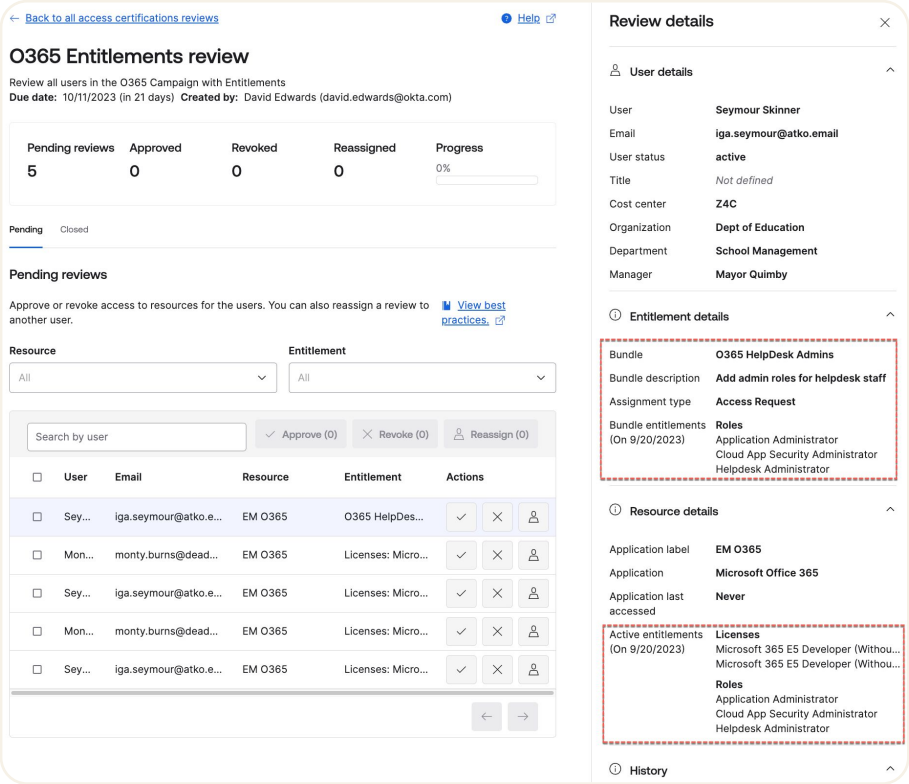
4. Set the values on the **Users** and **Reviewer** pages as you would for any campaign
5. Set the values on the **Remediation** page as you would for any campaign

   This campaign will review all entitlements, both those assigned via policy and via bundles, so automatically removing access (particularly licenses) may lead to unexpected results, thus we would recommend "Don't take any action".
6. **Schedule** the campaign
7. **Launch** the campaign
8. Review the entitlements

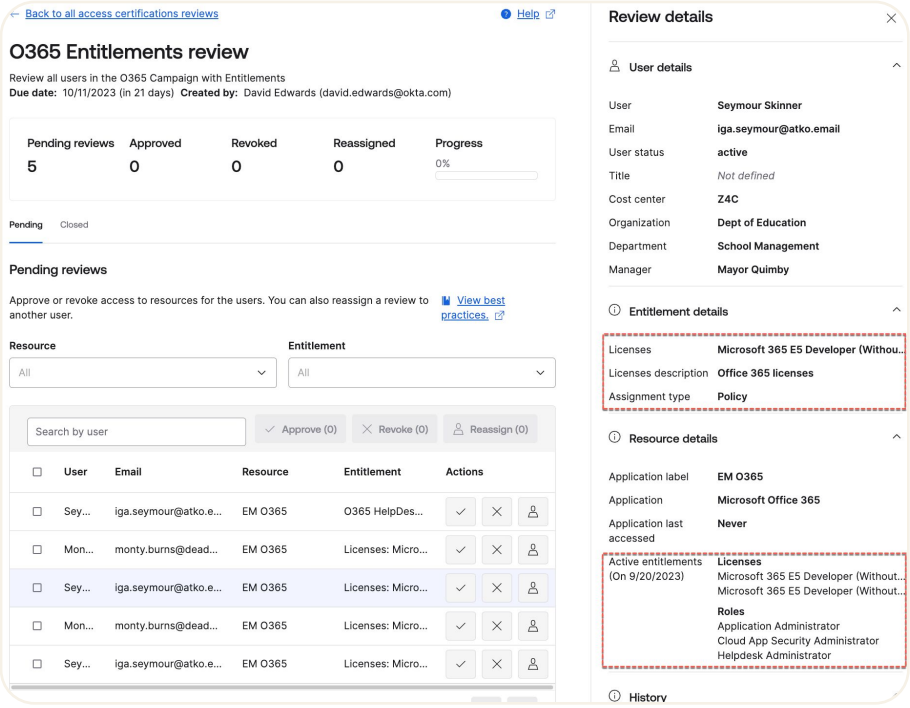
Review Campaign as Reviewer

The second part is to review the entitlements as a reviewer. To do this:

9. Select one of the reviewers from the last step above and log into Okta as them, and go to **Okta Access Certification Reviews** on the Okta Dashboard
10. Select the new O365 campaign
11. Find and click on one of the Entitlement Bundles to see the details. Note the **Entitlement details** section provides bundle information, including the entitlements granted. Note the **Resource details** section also shows the Active entitlements.



12. Find and click on one of the Entitlements to see the details. Note the Entitlement details and Resource details sections.



Note that the actual License names are truncated and hard to see, even if you zoom in/out or increase the browser window. This is being addressed.

12. You can proceed to approve/revoke access but it's not important for the lab.
13. You can close the reviewer view and End the campaign.

This concludes the Microsoft Office 365 Entitlement Management lab section of this document.

Entitlement Management for Salesforce.com

This section of the lab guide walks through entitlement management for Salesforce.com.

We recommend using a NEW instance of Salesforce.com created and integrated for SSO (only) in your Okta org. This app should not have Provisioning enabled. If you use an existing Salesforce.com instance and it has or had provisioning enabled, enabling provisioning after governance is enabled may cause issues with existing users (depending on the entitlement policies in your Salesforce.com instance).

> If you are using a Developer Edition trial of Salesforce.com there are licensing restrictions that may impact entitlements. For example, you can only have two Salesforce users (one will be the default admin account) and two Force.com - Free users, but 5,000 Chatter Free users. But you cannot assign Roles to Chatter Free users. So you will be constrained in entitlement assignment in Okta and you may see provisioning errors on the target systems. The steps below will try to keep in the SFDC licensing guard rails, be careful to avoid issues.

Most of the steps in this section are a repeat of what was done above for the dummy app, so this section will not have the same level of detail as the previous section.

The example used below is based on a Developer Edition trial and demonstration data (e.g. users and groups) and may be different in your environment.

Set Up Application for Entitlement Management

These steps are the same as for the earlier lab, confirm the app instance in Okta then enable the Governance Engine.

Set Up Salesforce.com App with SSO in Okta

Prior to starting this part of the lab, you need to have a Salesforce.com instance created and set up for SSO in your Okta org. As part of this you should have the administrative user assigned to the app. Provisioning should NOT be enabled.

> The rationale for using a 'fresh' Preview org is simple. If you disable provisioning on an existing app instance to enable Governance Engine, you can lose all provisioning-related data, including relationships and rules.

If you have multiple Salesforce dev-ed instances in the one Okta org, and you will be using the same users, you should set a custom username rule to make the users unique.

Identify Your Test Users in Okta

Given the limitation on licensing for users, it makes sense to identify which ordinary users you will have and which will have special entitlements. For this lab we are creating one Entitlement Policy that will assign a CEO Role and Force.com license, one Entitlement Bundle that will assign another CFO Role and Force.com license, and all other users (ordinary users) will be assigned the Chatter Free license via an Entitlement Policy. This is only due to the restrictions of the developer edition Salesforce.com.

The example users are:

- CEO Entitlement Policy - **Monty Burns**
- CFO Entitlement Bundle (via Access Requests) - **Seymour Skinner**
- Other Salesforce.com users (other Policy) and additional Entitlement Bundle - **Waylon Smithers** and **Bart Simpson**

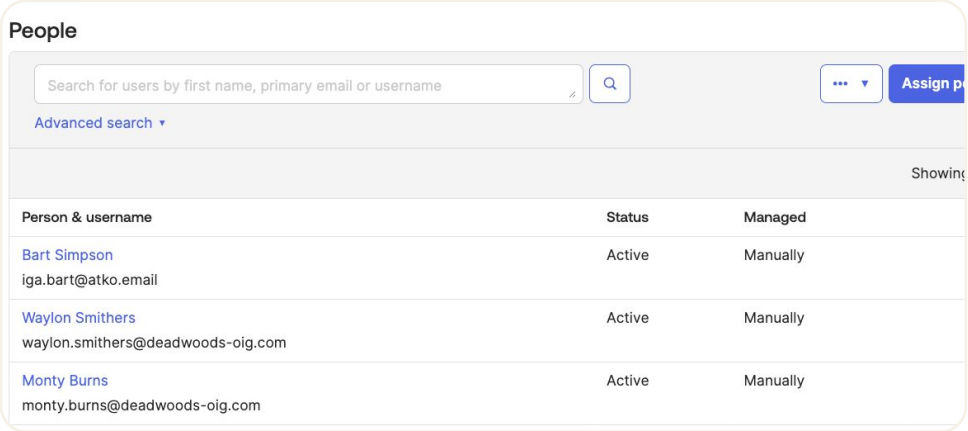
These will appear in the screenshots below. You can use any of the users in your test system.

Assign Okta Group to Salesforce.com App in Okta

For the entitlement policy evaluation below, you will need users assigned to the app and the easiest way to do this is via an Okta Group.

1. Create an **Okta group** and put your Salesforce test users into it.

For this lab, I need to assign all users who will be subject to a policy, so **Monty Burns**, **Waylon Smithers** and **Bart Simpson.**



2. Assign the group to the Salesforce.com instance.

Note that with Entitlement Management, you can no longer set the Profile (i.e. license) with the group assignment. We will do this when we define the policy.

As provisioning is not enabled yet, you will see the users assigned but no provisioning events to Salesforce.com in the System Log.

Check/Set Up Entitlement Data in Salesforce.com

Prior to enabling provisioning, you will need to check the entitlement data in your Salesforce.com instance. The Salesforce.com instance will have values there for Permission Sets and Profiles.

3. You can create some **Roles** of your choosing based on one of the sample role hierarchies. For example the Territory-based Sample.
4. You can also create some **Public Groups** of your choosing. I created "*East Region*", "*Central Region*", "*West Region*", and "*Federal*".

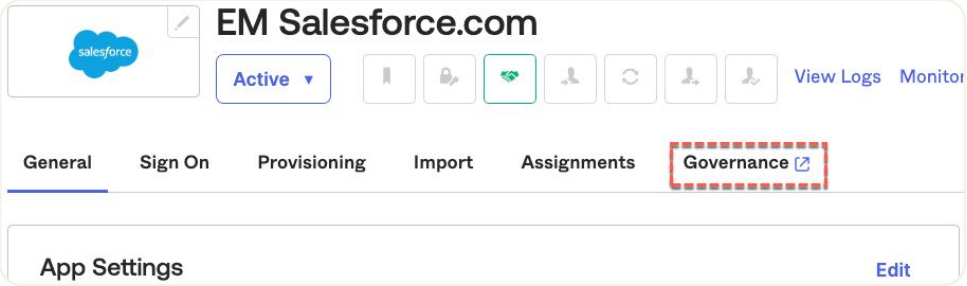
Enable the Governance Engine

We will enable the Governance Engine for the Salesforce.com app.

5. For your Salesforce.com application instance, on the General tab, edit and enable the Governance Engine

   Note the message saying it is being enabled in the background and please check back later.

6. Refresh to confirm that the Governance tab has been added

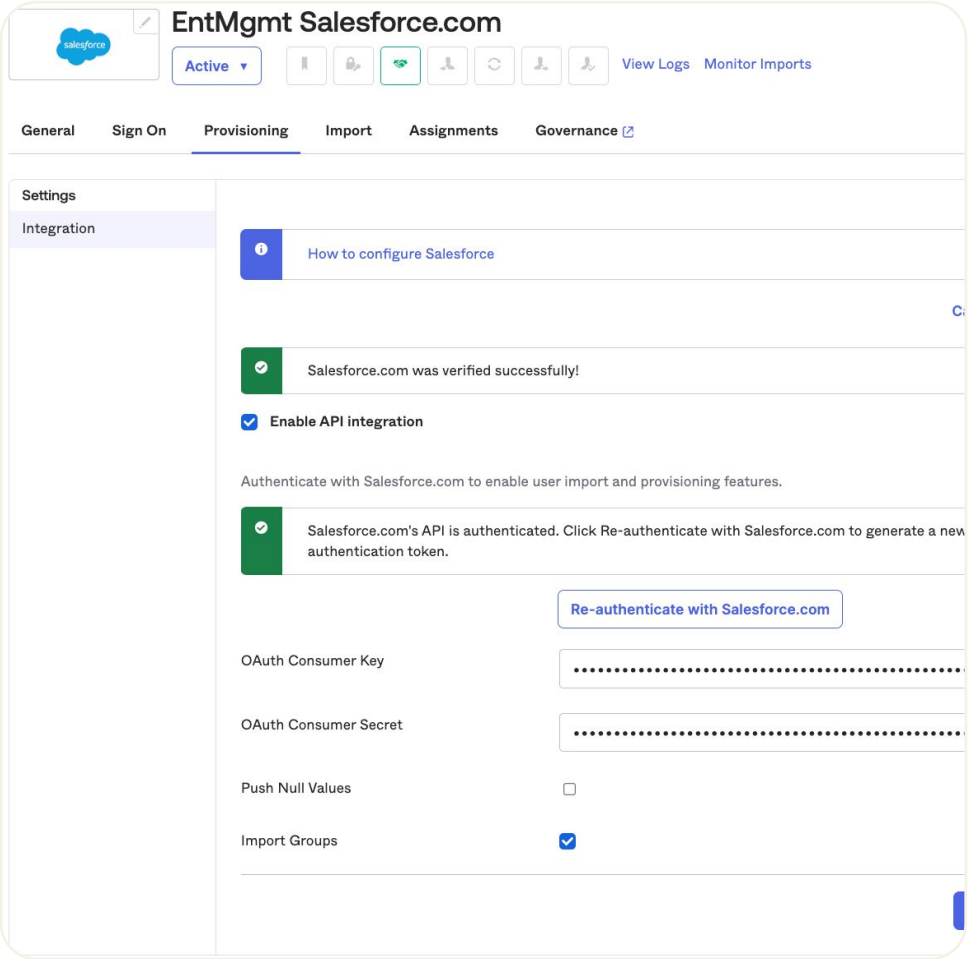


The Governance Engine is enabled, but we need to turn on Provisioning to get the entitlements from the app.

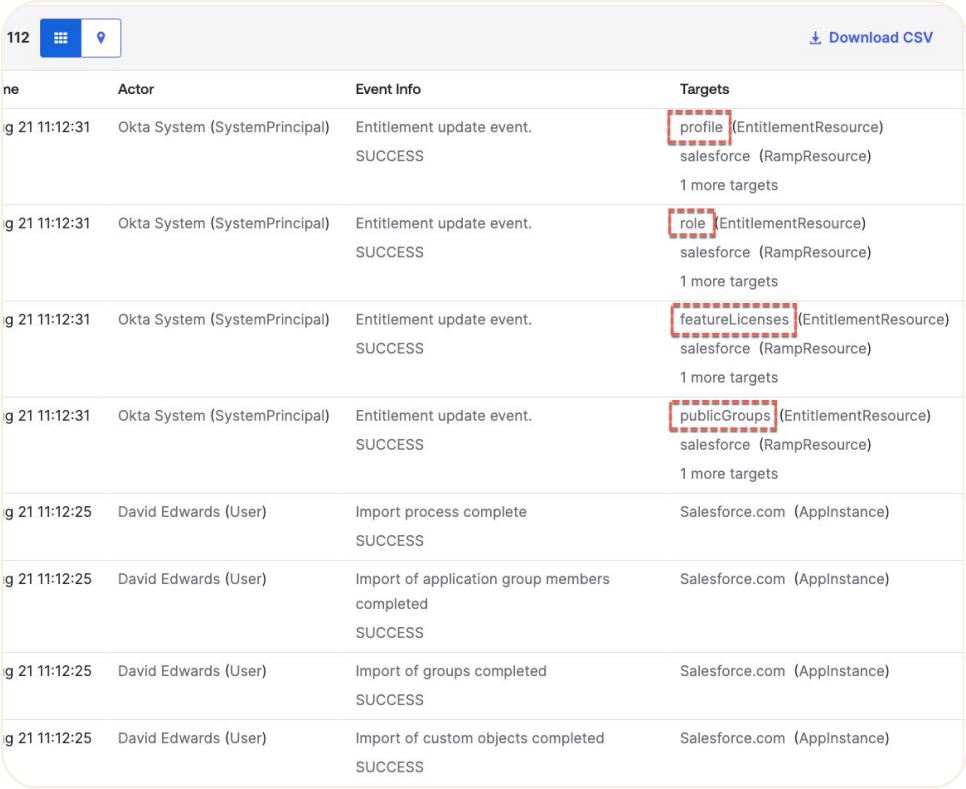
Enable Provisioning

The last step is to enable Provisioning for the Salesforce.com app. This changed with the newer OIN integrations to support OAuth and REST. See the instructions in https://help.okta.com/en-us/Content/Topics/Provisioning/Salesforce/sfdc-configure-provisioning-REST.htm.

7. Run through the steps to enable provisioning. You should see the API integration enabled.



8. **Save** the Provisioning settings
9. Go to the **To App** page and enable the **Create Users**, **Update User Attributes** and **Deactivate Users** options
10. Click the **View Logs** option to see the System Log entries for this app. It should include events for the Entitlements in Salesforce.com



112 Download CSV

ne	Actor	Event Info	Targets
g 21 11:12:31	Okta System (SystemPrincipal)	Entitlement update event. SUCCESS	profile (EntitlementResource) salesforce (RampResource) 1 more targets
g 21 11:12:31	Okta System (SystemPrincipal)	Entitlement update event. SUCCESS	role (EntitlementResource) salesforce (RampResource) 1 more targets
g 21 11:12:31	Okta System (SystemPrincipal)	Entitlement update event. SUCCESS	featureLicenses (EntitlementResource) salesforce (RampResource) 1 more targets
g 21 11:12:31	Okta System (SystemPrincipal)	Entitlement update event. SUCCESS	publicGroups (EntitlementResource) salesforce (RampResource) 1 more targets
g 21 11:12:25	David Edwards (User)	Import process complete SUCCESS	Salesforce.com (AppInstance)
g 21 11:12:25	David Edwards (User)	Import of application group members completed SUCCESS	Salesforce.com (AppInstance)
g 21 11:12:25	David Edwards (User)	Import of groups completed SUCCESS	Salesforce.com (AppInstance)
g 21 11:12:25	David Edwards (User)	Import of custom objects completed SUCCESS	Salesforce.com (AppInstance)

11. In the current release, you should see Profile, Role, Feature Licenses and Public Groups

The entitlements from Salesforce.com are now imported and can be viewed and used.

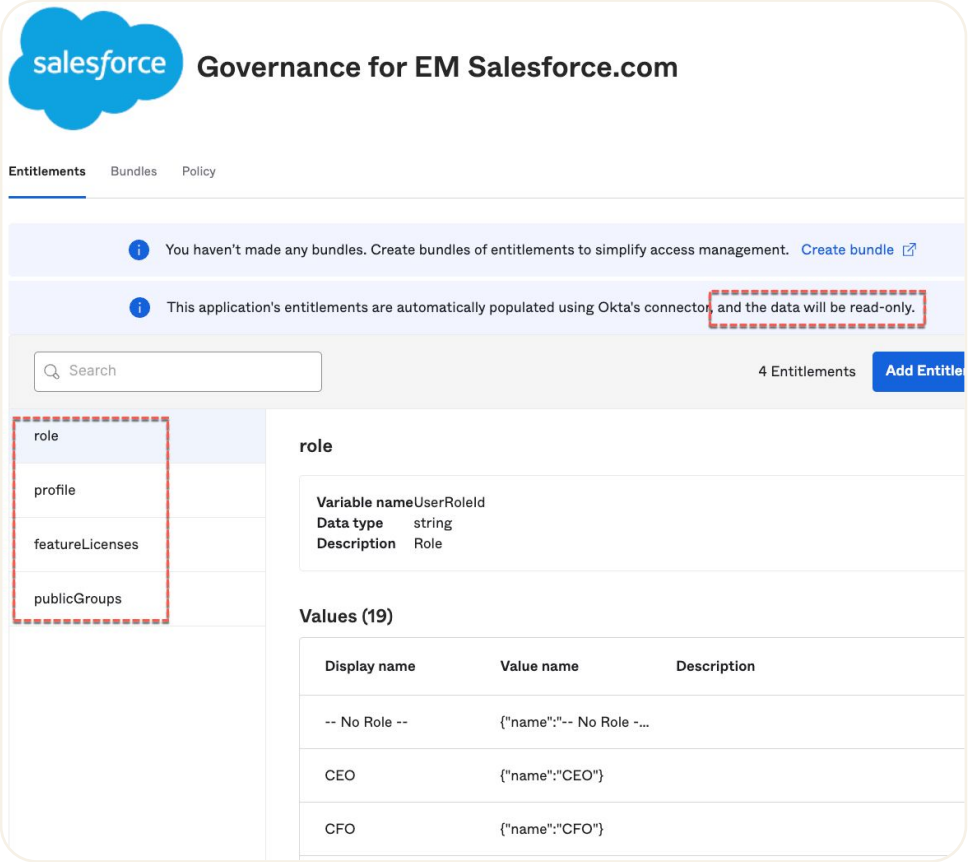
View Entitlements for Salesforce.com

In this section we will view entitlements for Salesforce.com in the same way that we did for the dummy app.

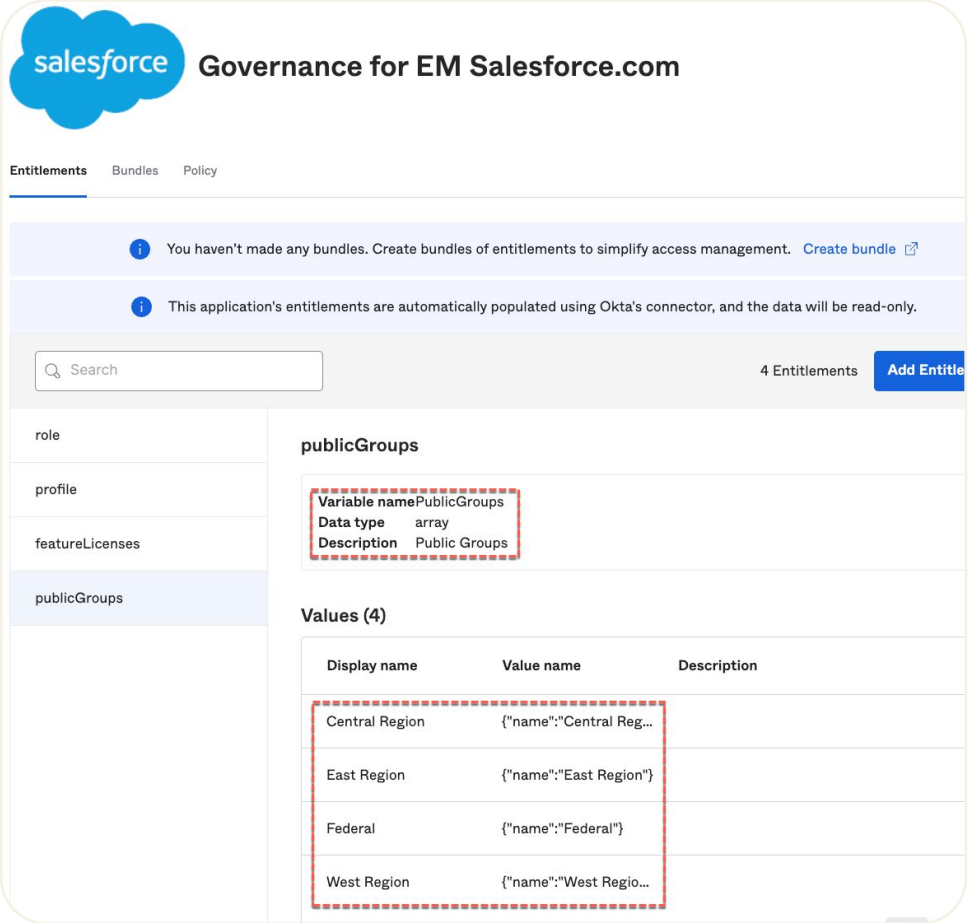
> Entitlements cannot be managed for Salesforce.com in Okta. They are read-only. Any changes to them will need to be performed in Salesforce.com.

There are many entitlement types in Salesforce.com, such as Roles, Profiles, Feature Licenses and Public Groups. For this lab we will look at the ones imported.

1. Go into the **Governance** tab for the Salesforce.com app
2. The default view is the Entitlements tab. Check that you have four entitlements; **role**, **profile**, **featureLicenses** and **publicGroups.**



3. Check that you see the **roles** you have in your Salesforce instance
4. Click on the **publicGroups** entitlement



5. Check that the entitlement is of type **array**, and that the **Public Groups** you created are there.

You are now ready to use the Salesforce.com entitlements in entitlement policy and entitlement bundles.

If you don't see values like the above, such as missing entitlements or values, check the events in the system log.

Create Entitlement Policy for Salesforce.com

In this section we will create two entitlement policies, one for ordinary users that will get the Chatter Free user license (profile) and one, for users (user) with the title of CEO to automatically get the CEO role in Salesforce.com.

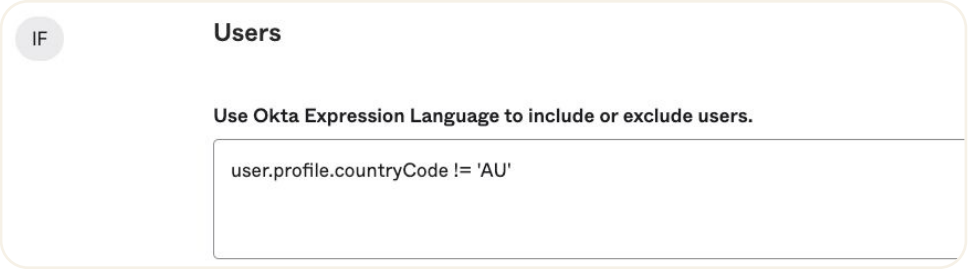
You will need a user with the title of "CEO" and also one of the role entitlements in Salesforce.com should be CEO. If you don't have that role, pick another and set the user title to match.

The steps below are brief as they are the same as done earlier with the dummy app.

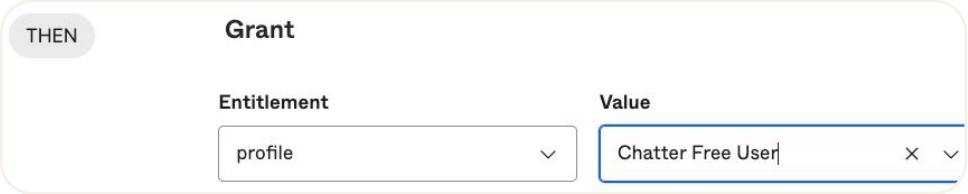
First we will create the ordinary users policy rule.

1. Create a new **Entitlement Policy Rule** in the Salesforce.com app for all users. Give it a name like "*All Users*"

2. Set the **IF** clause to be *user.profile.countryCode != 'AU'*. You need a rule to catch all users, and in my environment no-one has a country code set. If this contradicts with data you have in your Universal Directory user profiles, you may need to use another.



3. Set the THEN clause to assign the **Chatter Free** User profile.



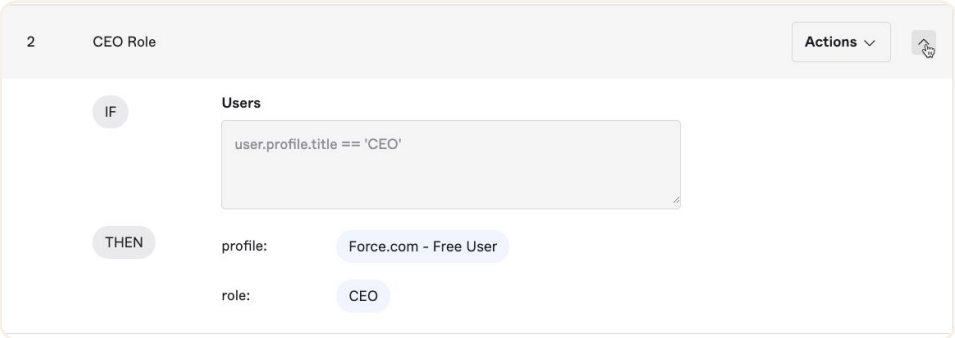
4. **Save** the new Rule

Create the CEO Role rule:

5. Create a new **Entitlement Policy Rule** in the Salesforce.com app for the CEO. The OEL will be *user.profile.title == 'CEO'*. Check the rule against your test user.

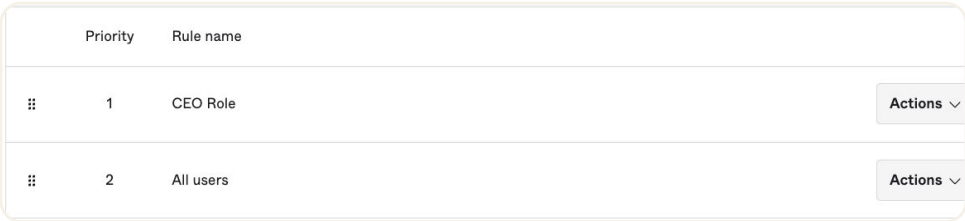
6. Add two **Grants** to the Rule; **role = CEO** and **profile = Force.com - Free User**

7. **Save** the rule

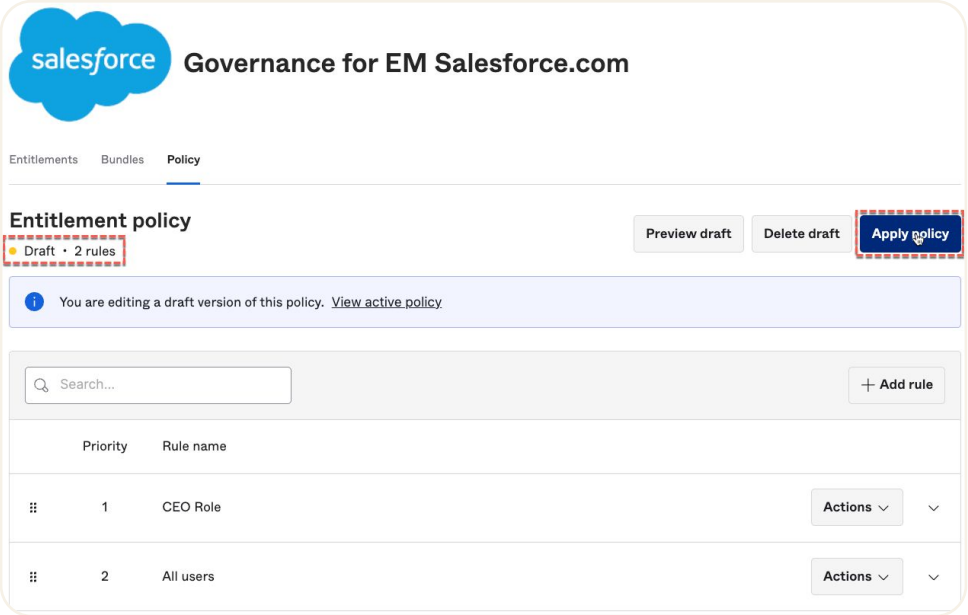


Notice the order of the Rules. The All users rule is Priority 1 and the CEO Role rule is Priority 2. This means the All users rule will get evaluated first. If the entitlement is an array, this doesn't matter as the Governance Engine will consolidate all entitlement values for all rules that apply to a user. But if the entitlement is a single value, the first rule that matches will be used to set the entitlement value.

8. Use the handles beside the rules to put the CEO Role rule as Priority 1.

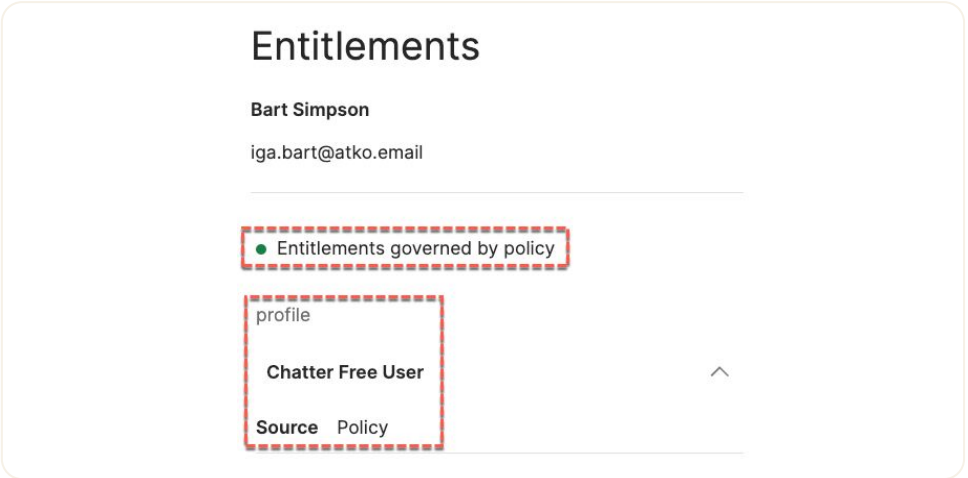


9. The policy will be in **Draft** mode. Click the **Apply policy** button and again in the popup dialog. (As noted in an earlier part of the lab, you can also Preview the policy before committing to it).



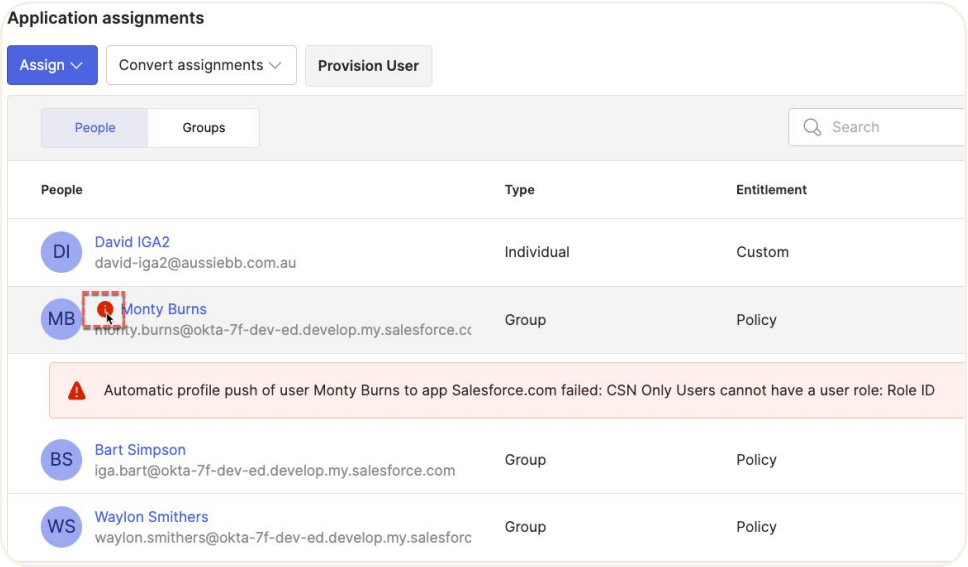
The policy evaluation is a background process and may take some time to run.

10. 10Go back to the application and check the assigned users. Look at each of the entitlements (three vertical dots icon beside the user) to see the entitlements.

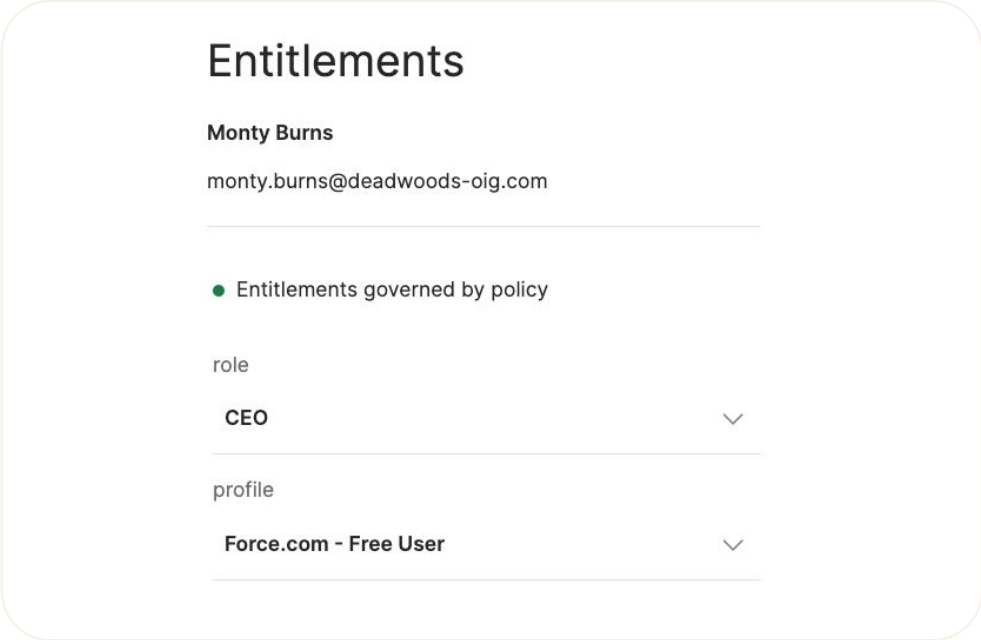


In this case, you can see the All users policy has assigned the Chatter Free User profile (license).

11. If you get any errors, you can click on the error indicator to see the error message.

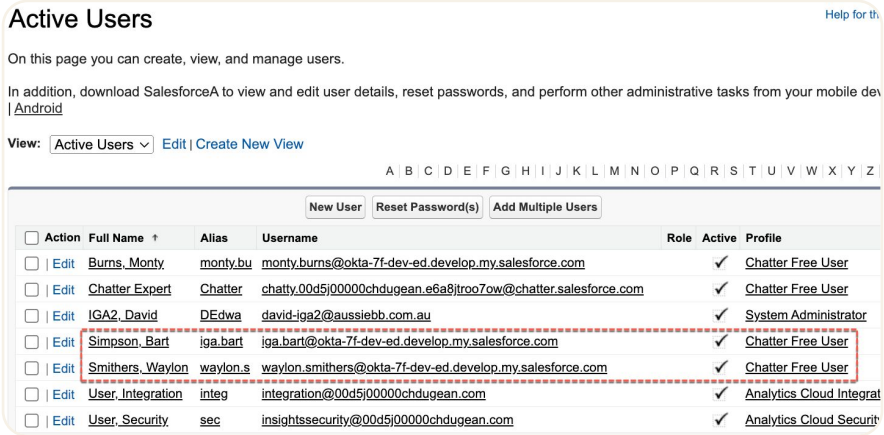


In this case the Entitlement Policy has worked.



The policy evaluation has worked, and this user has got the higher priority CEO Role policy rule, but there's been a conflict in assigning this role to the user. The policy rules are correct but there are restrictions on the target. It's up to you whether you try to fix this in the lab.

12. Go into Salesforce.com and confirm the users have been added/updated.



This concludes the Entitlement Policies part of the Salesforce.com lab.

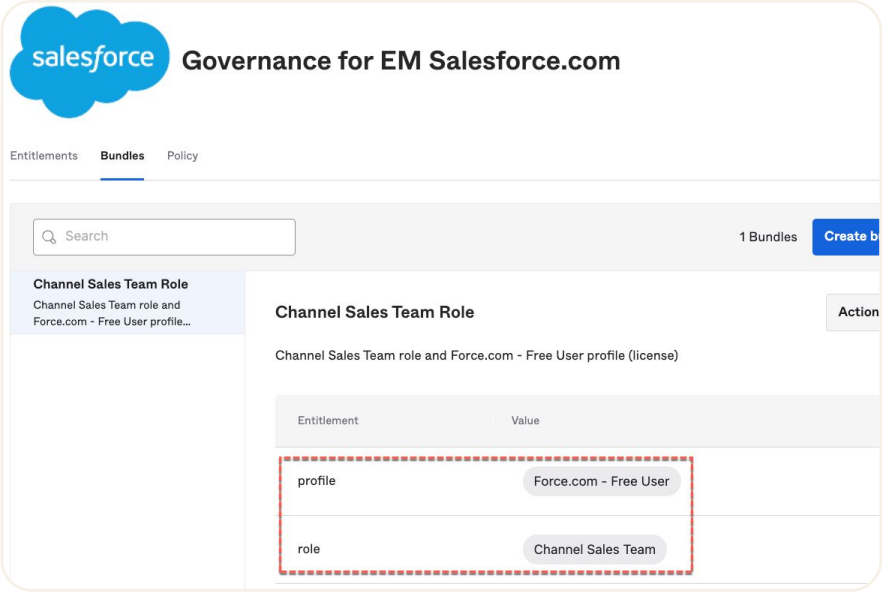
Create an Entitlement Bundle and Request Type

In this part of the lab we will create a bundle for a role and expose it in Access Requests.

Create Entitlement Bundle for the Channel Sales Role

We will create a single bundle to expose a role and profile combination.

1. As you did with the dummy app, create an **Entitlement Bundle** to request the Channel Sales Team role and Force.com - Free User profile

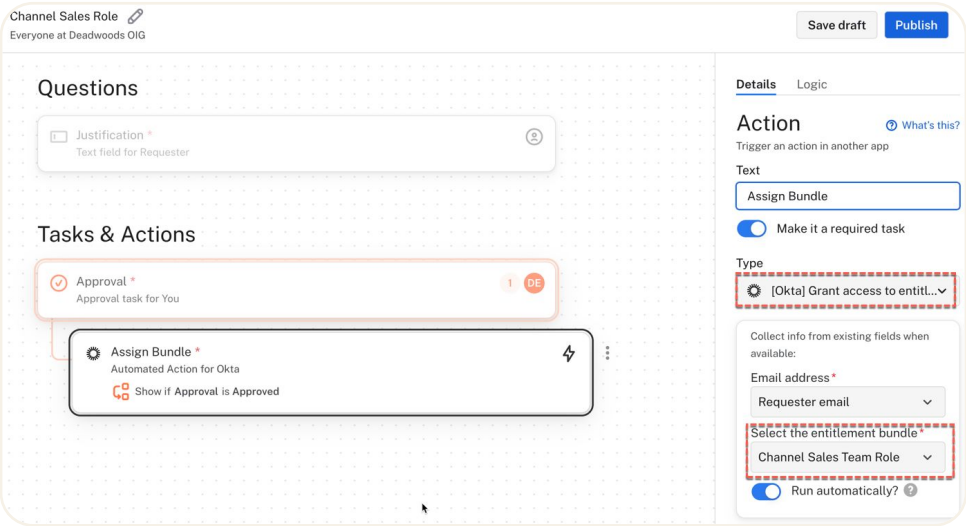


This can now be used in an access request.

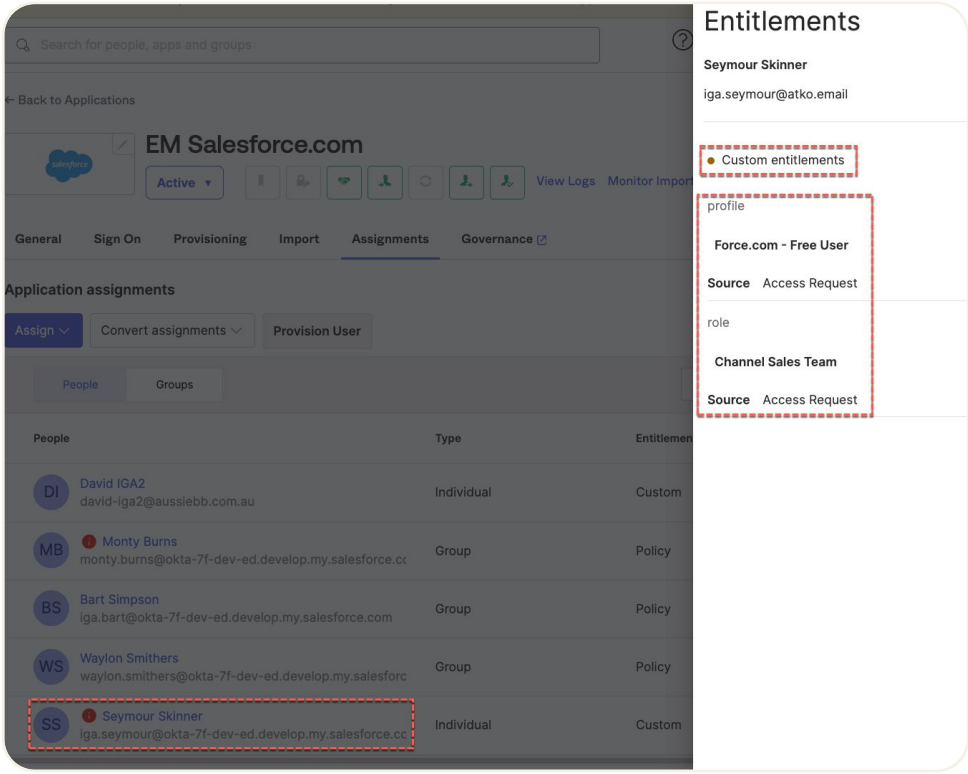
Role Entitlement Bundle in Access Requests

We will create a Request Type for the new Entitlement Bundle and test it works.

2. Go into Access Requests as the administrative user
3. Go to Settings, select the Resources tab and select the Entitlement bundles resource
4. If your new entitlement bundle is not there, click the Update now button and refresh until it is there
5. Go to Access Requests and select Create request type
6. Create a new request as you did for the dummy app:
    a. Give it whatever name you want, select the team (e.g. IT) and make the audience **everyone**
    b. Put in a **Question** for Justification (text)
    c. Put in an **Approval** step, to your admin user (or manager if you want and have the manager relationship setup)
    d. Put in an **Action** to assign the user to the new entitlement bundle (“**Grant access to entitlement bundle**”) with logic to only run if the approval step was approved



7. Update the request type to close automatically and publish
8. As a user, request this role
9. As the reviewer, approve the request
10. As the admin monitor the completion of the request
11. Go back into the **Governance** view for the Salesforce.com app and check the user has been added and granted the entitlements as per the bundle



You may see an error against the user if there is an entitlement conflict in Salesforce. For example “Automatic provisioning of user Seymour Skinner to app Salesforce.com failed: Guest Users cannot have a user role: Role ID”. You may need to revisit the bundles in this case. If you do make changes, you may need to try a different user if there’s a conflict with the existing user or tweak the data in Salesforce.

12. Confirm the System Log entries. You will see some normal provisioning entries and also some relating to the Okta IGA connector.

Aug 21 14:06:29	David Edwards (User)	Push user's profile to external application SUCCESS	Seymour Skinner (AppUser) Seymour Skinner (User) 1 more targets
Aug 21 14:06:27	David Edwards (User)	Updated user application property SUCCESS	Seymour Skinner (AppUser) Salesforce.com (AppInstance) 1 more targets

Aug 21 13:59:08	Okta IGA Connector (PublicClientAppEntity)	Verify user exists in external application SUCCESS	Seymour Skinner (AppUser) Seymour Skinner (User) 1 more targets
Aug 21 13:59:06	Okta IGA Connector (PublicClientAppEntity)	Add user to application membership SUCCESS	Seymour Skinner (AppUser) Salesforce.com (AppInstance) 1 more targets

13. You can also go into Salesforce and check the user.

This concludes the access request use cases.

Create and Run An Access Certification Campaign

This section will create and execute an access certification campaign for the Salesforce.com entitlements. The steps are exactly the same as done for the dummy app, just with the Salesforce.com app.

1. **Create** a campaign to review all entitlements for the Salesforce.com application you have been working with above. You may want to set it to revoke access on a revoke.

EM Salesforce.com Entitlements • Scheduled Ac

Campaign settings

Schedule
Created date: 8/21/2023
Start date: 8/21/2023
Start time: 7:00:00 PM GMT+10
Duration: 21 Days

Resources
Applications (1): EM Salesforce.com (All entitlements and bundles)

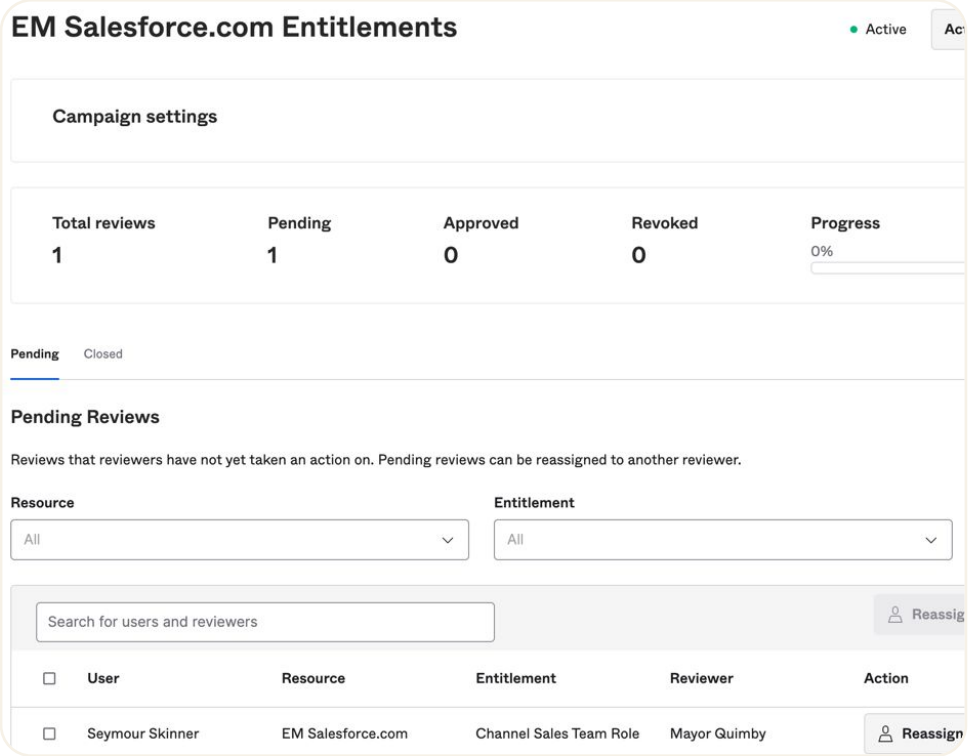
Users
All Users

Reviewer
Reviewer type: Manager
Fallback reviewer: Okta Admin (okta.admin@deadwoods-oig.com)

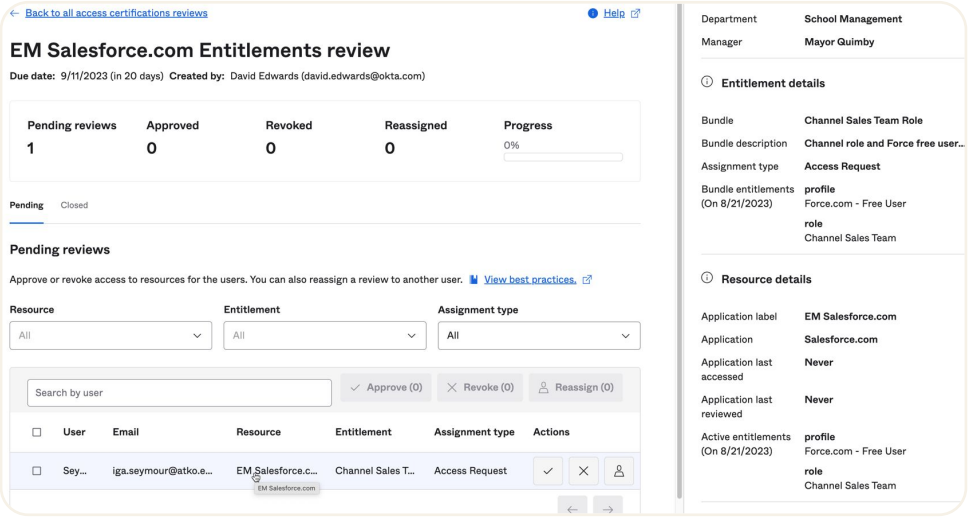
Notifications
Notifications not enabled

Remediation
Approved: Don't take any action
Revoked: Don't take any action
No response: Don't take any action

2. **Launch** the campaign and check the entitlements that will be reviewed.



3. As a reviewer, go review the entitlement.



If you have set the campaign to revoke access on the target (Salesforce.com) you should revoke an Entitlement that has not been granted by entitlement policy (entitlements assigned via entitlement policy won't be removed, but rather flagged for manual remediation).

4. Go into Okta to check the entitlement was removed, then go into the Salesforce.com instance and check it was removed (the user will not be removed, and if the entitlement was a mandatory one in Salesforce, you may see the entitlement is retained).

This concludes the Entitlement Management lab for Salesforce.com.

About Okta

Okta is the World's Identity Company. As the leading independent Identity partner, we free everyone to safely use any technology—anywhere, on any device or app. The most trusted brands trust Okta to enable secure access, authentication, and automation. With flexibility and neutrality at the core of our Okta Workforce Identity and Customer Identity Clouds, business leaders and developers can focus on innovation and accelerate digital transformation, thanks to customizable solutions and more than 7,000 pre-built integrations. We're building a world where Identity belongs to you. Learn more at okta.com.