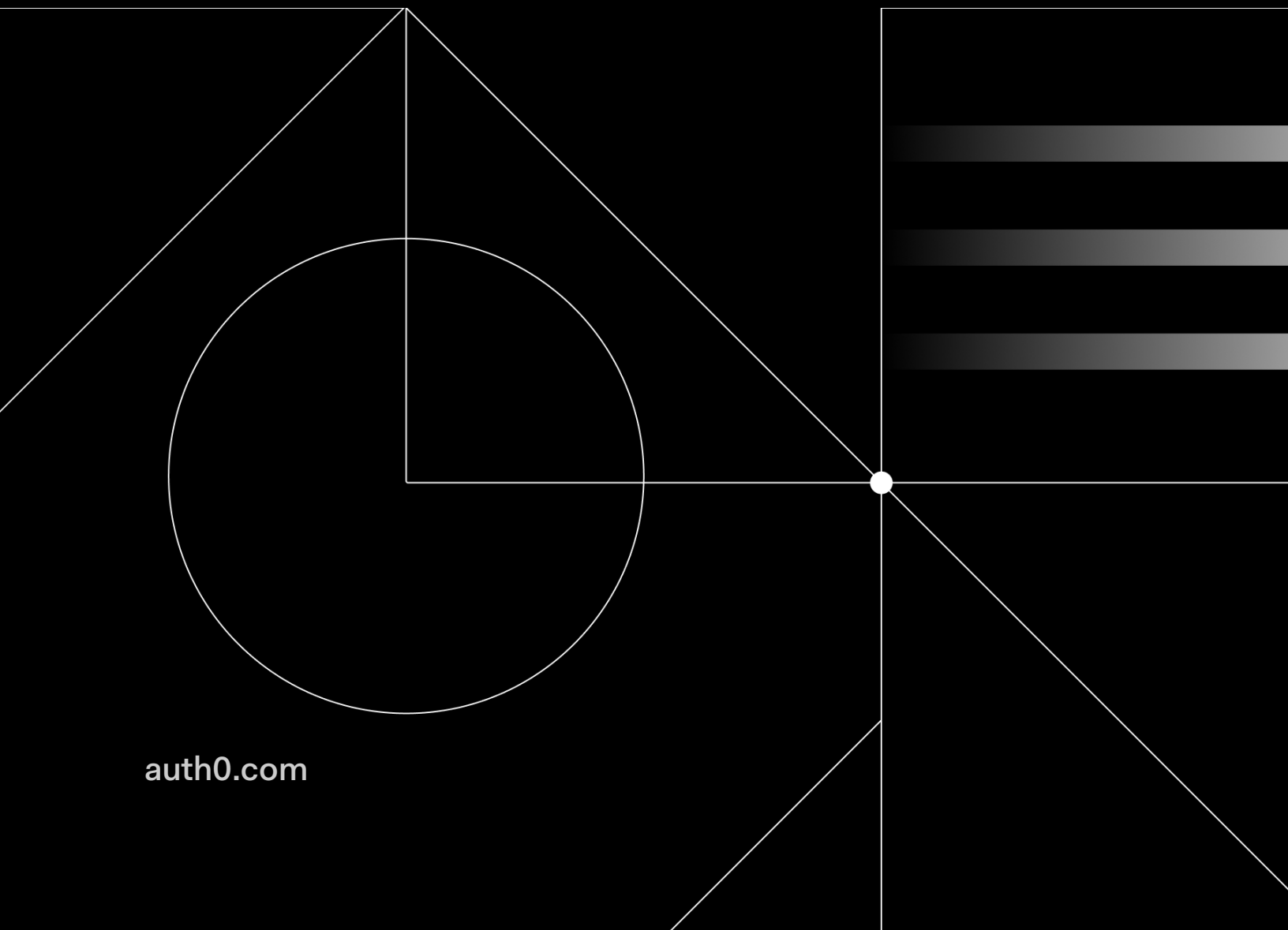
Auth0

User Migration Your Way

Recommended Best Practices for Your Timetable



auth0.com

Now that global events have forced companies to migrate to cloud services, it's clear to us all that digital transformation is a necessity.

Identity and access management (IAM) is a foundational piece of digital transformation, since legacy identity solutions are often not scalable or reliable enough to support today's more complex, regulated, and hybrid application use cases. When it comes to IAM, companies face the usual dilemma: **build or buy**?

Enterprises approach this dilemma in different ways. Some decide to build their own IAM solution. Others prefer to buy cloud-based SaaS solutions. Still others mix homegrown and purchased IAM solutions, using different solutions for different applications or specific types of users (employees, customers, suppliers, etc.).

For many companies, particularly on the enterprise level, the build-or-buy binary doesn't describe their experience. Their mixed environments combine built-to-spec solutions with out-of-the-box microservices. These mixed environments present particular challenges for security and user experience (UX).

Now, with the necessity to move applications and components onto the cloud, companies know they need to unify their fragmented identity approach to ensure security and a seamless UX. One of the main concerns around this consolidation process is user identity migration:

- How can you minimize the effort and internal disruption involved in migrating users from one or more IAM systems to a cloud IAM solution?
- How can you ensure that this migration process causes as little disruption as possible, such as password resets, for users?
- During a migration, how can you even **improve** the current level of security?

Auth0 can help you find great solutions to these questions.

> "We were looking for something that would give us not only a Single Sign On or unified experience down the track, but something that offered us a very clear migration path from the disparate sources we had across ServCorp. [Auth0's] custom login scripts were very important as this meant we could do rolling migrations, which significantly decreased project risk. That whole migration strategy, no one else could do."
>
> - Matthew Baumgartner, Global CIO, ServCorp

How Auth0 Helps

Auth0 provides you with many options for migrating your user profiles. The best option depends on your current identity scenario and what your constraints are in terms of time, business continuity, etc.

Let's start by taking a look at how various Auth0 tools support different user identity migration scenarios. These include:

- Bulk Migration
  - The Management API
  - The Import User extension
- Automatic user migration
- Continue using your existing user store

Bulk Migration

Bulk Migration migrates users to Auth0 in one step. There are two available options, the User Import Extension and the Auth0 Management API. Before we dive deeper into these two topics, we should mention a key benefit of bulk user migration. **It supports importing hashed passwords**. You can directly import your users' passwords hashed with bcrypt or specify one of several standard common hashing algorithms supported by Auth0. Most enterprises are looking for a frictionless way to migrate their existing legacy user stores to a modern IAM platform, while minimizing password support costs. With this feature, Auth0 offers password imports right out of the box, making migration completely transparent for users.

The Management API

The most flexible and efficient way to import users is the Auth0 Management API. The Management API allows your application to perform the same actions you can do through your Auth0 dashboard. In this case, you have to develop your own application to start the user profile import job.

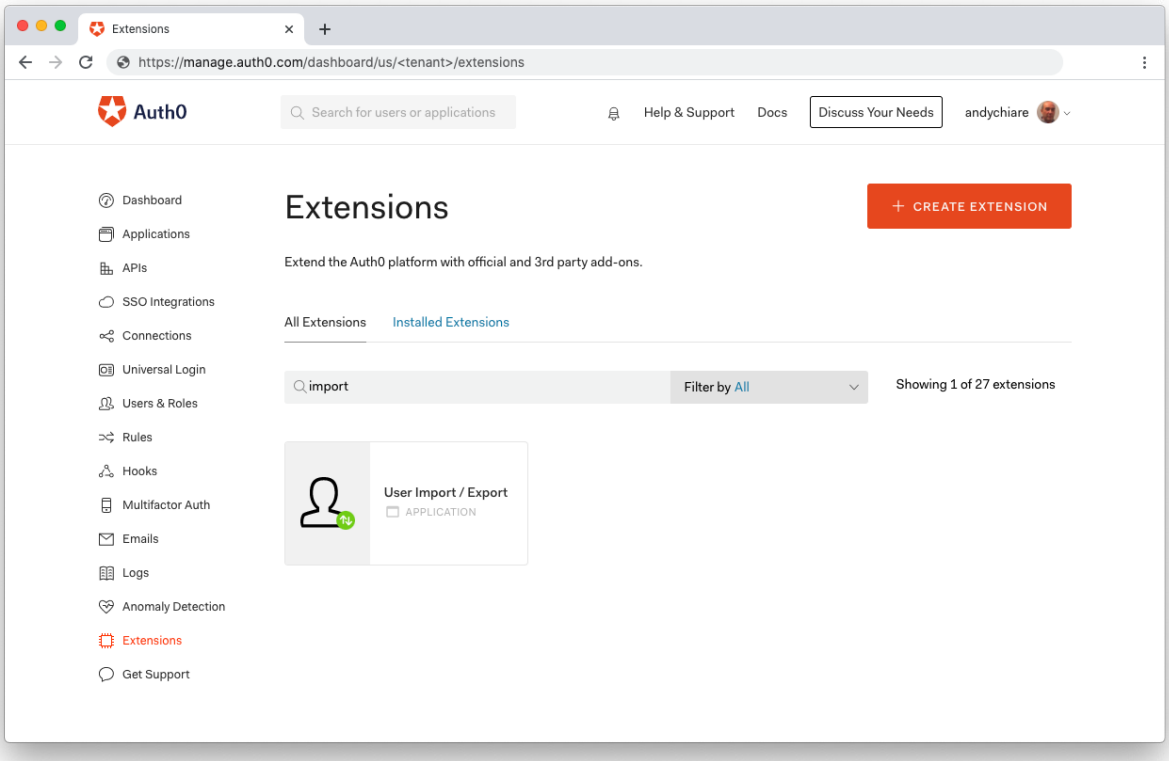
In simple terms, your application has to implement a machine-to-machine communication with the Auth0 management services. This means that your application needs an access token that authorizes it to use the API. Then, with this token, your application can invoke the user-import endpoint passing a JSON file properly structured according to a specific JSON schema. This JSON file represents user profiles you want to import, including their passwords. You will be learning more about this schema later on.

Calling the user-imports endpoint starts the upload and user import job as an asynchronous process. The Auth0 Management API provides you with other endpoints to check the process status.

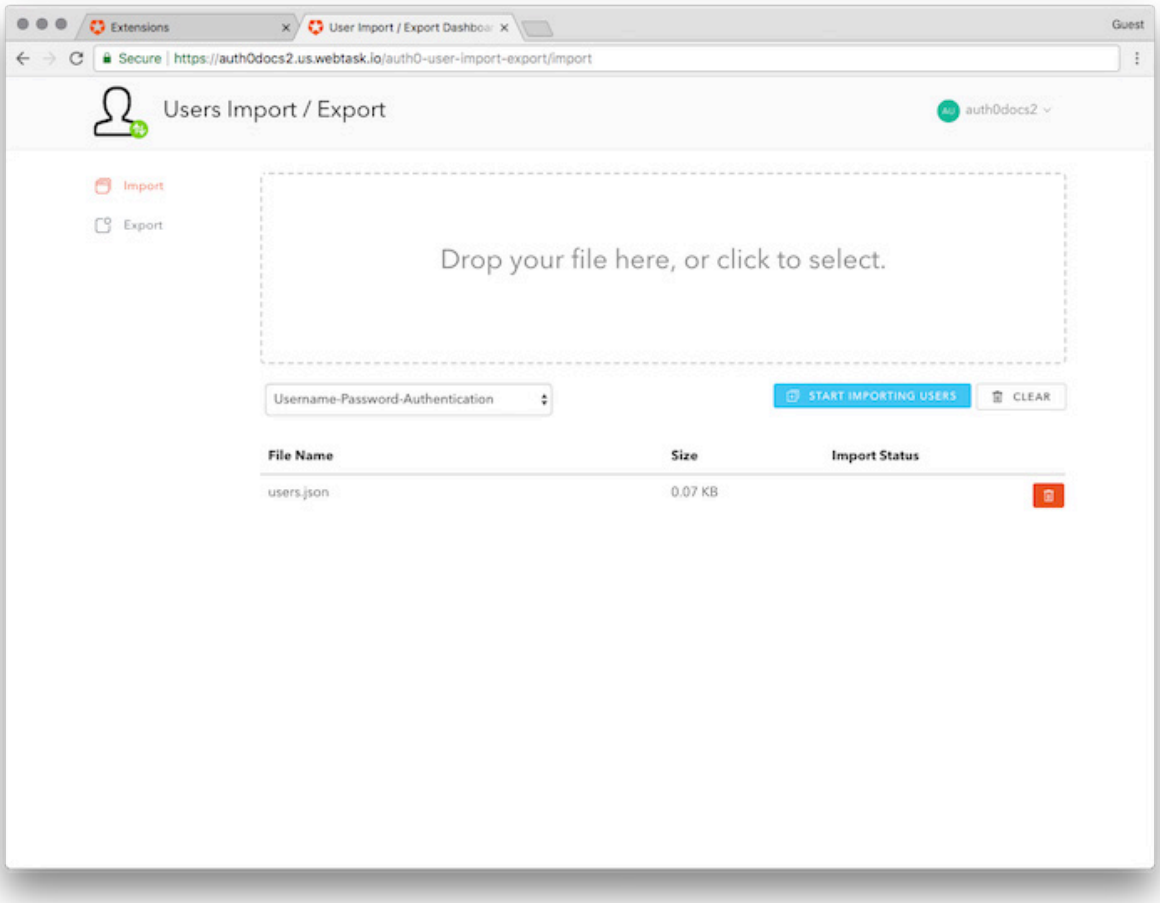
Check the documentation to learn more about importing users via the Management API.

The Import User Extension

As an alternative to the Auth0 Management API, you can migrate your user profiles to Auth0 by using the User Import/Export Extension. This is a UI-based approach to import users, but, as you might guess, it is less flexible than the Management API. To use it, you need to install this extension on your tenant from the Extensions page of your Auth0 dashboard:



Once installed and authorized, the extension allows you to upload the same JSON file containing the user profiles as in the case of the Auth0 Management API.

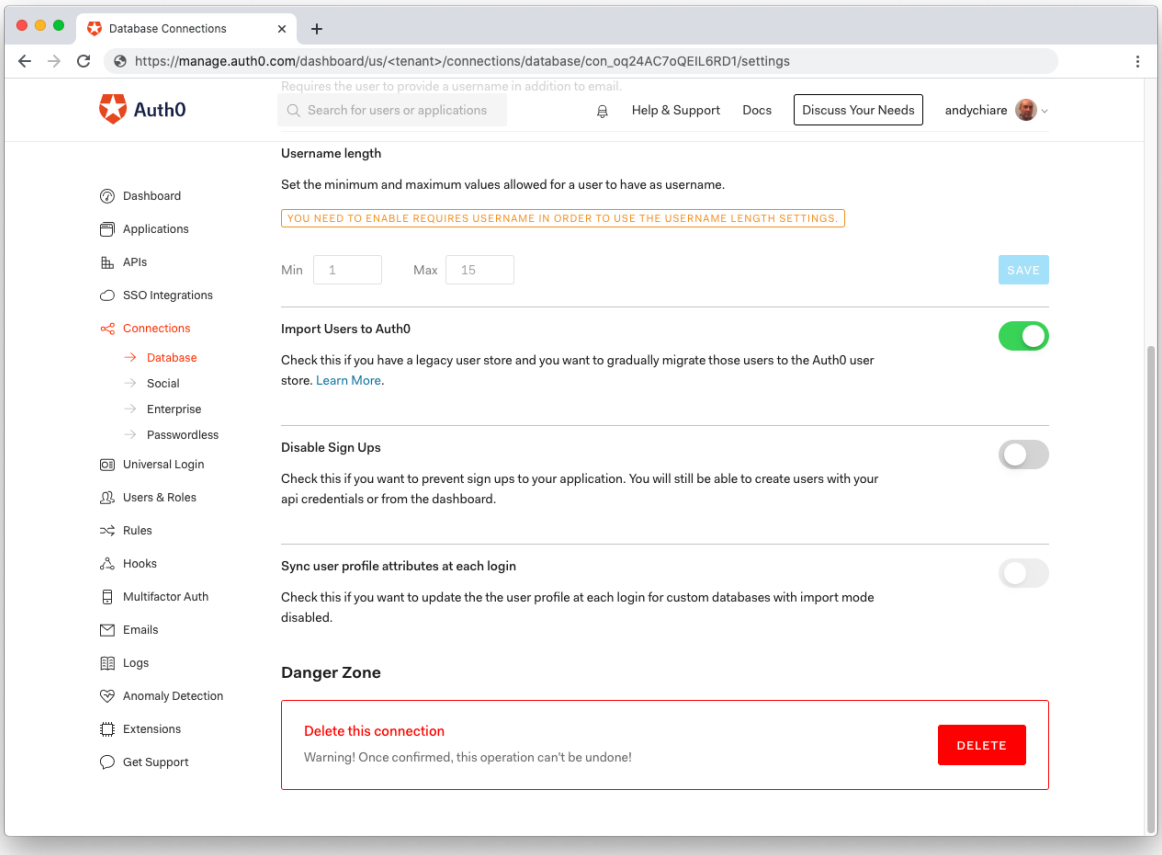


The JSON file containing your user profiles will populate the specified Auth0 database. More information on using the Import User Extension is available in the documentation.

Automatic Migration

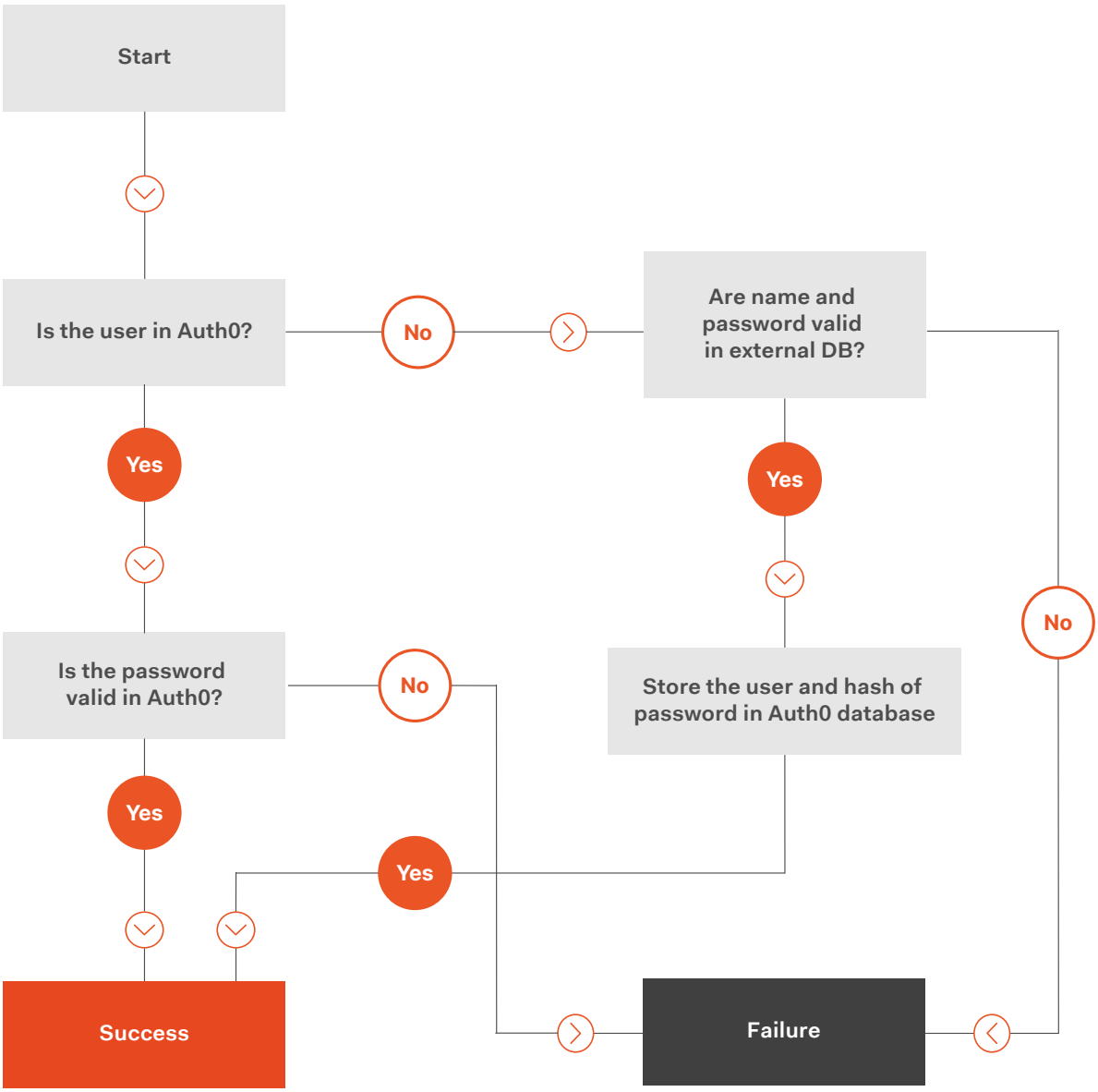
An alternative to migrating your users in a single import step is the **Automatic Migration (also called lazy or trickle migration)**. With this approach, you import users into your new Auth0 database as they sign in. You don't need to perform a massive import operation; your users will migrate automatically while using Auth0 to authenticate.

Automatic Migration is based on keeping alive both your legacy user store and the new Auth0 database. Your new database is connected to your legacy database through a custom database. This is a similar configuration as the one discussed in **the approach that uses just your user store**. However, in this case, you can enable user import simply by flipping a switch:



The last step is writing two Node.js scripts. The first one will be executed each time a user attempting to log in is not found in the Auth0 database. In this case, the script will query your legacy user store to look for that user profile and, if found, copy it into the new Auth0 database. The second one will take care of the sign-up and forgotten password flows.

The following diagram shows the flow of the Automatic Migration process.



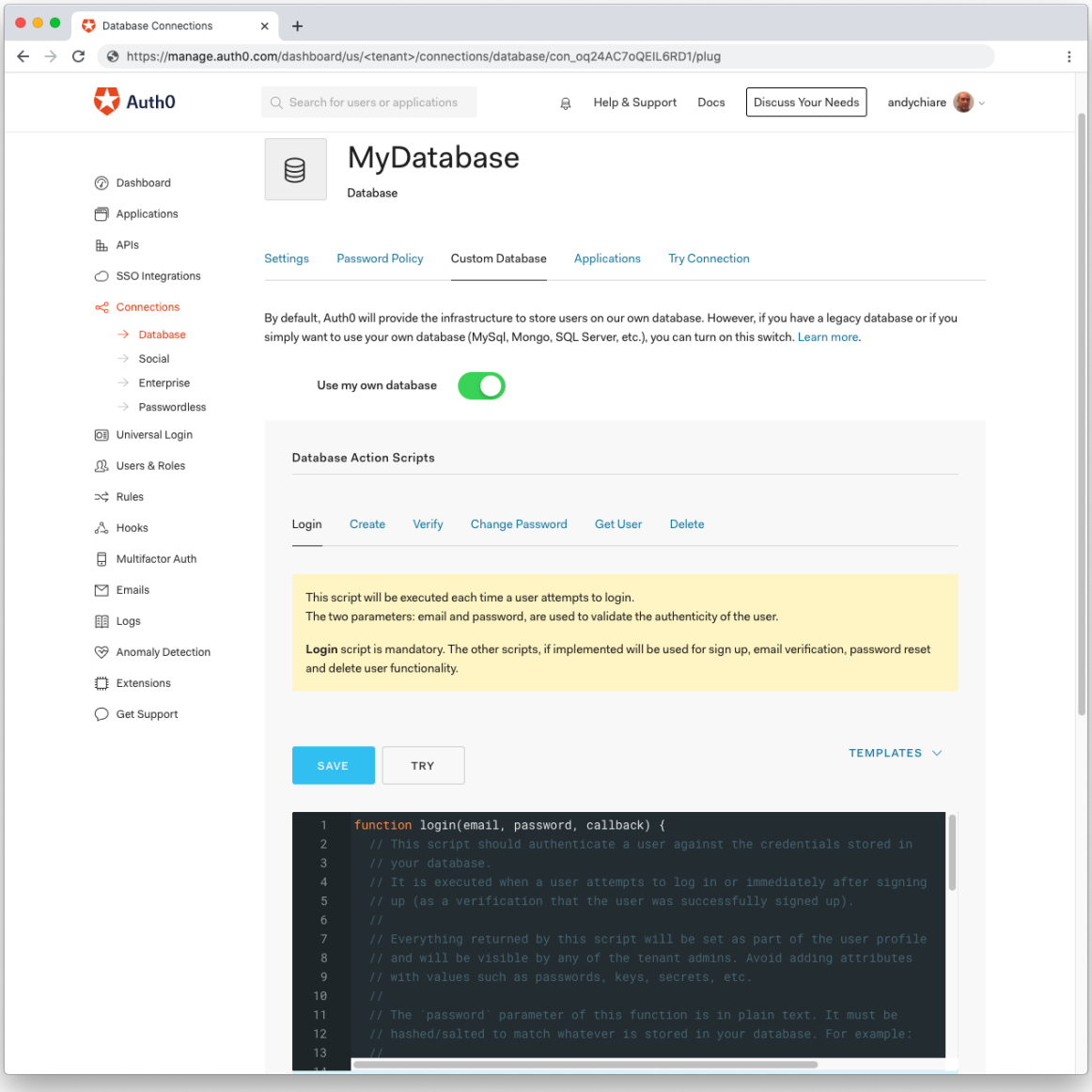
To learn more about using this approach, check out the documentation.

Utilizing your own User Store

While by default Auth0 provides you with the database infrastructure to store your users, you also have the option of using your own database or user store. Your user profiles will continue to live in your legacy user store, but you can also leverage the security infrastructure of Auth0 services.

This approach isn't an actual user migration, but you can consider it the first step towards Auth0 integration.

To utilize your own user store, you need to connect a custom database in your Auth0 dashboard, as shown in the following screenshot:



Auth0 allows you to write a script using JavaScript to access your database through many means. Our primary recommendation is to use an API in front of your database, but if that is not feasible, we have script templates for the most common databases (MongoDB, MySQL, Oracle, PostgreSQL, SQL Server, Windows Azure SQL Database, and more), or you can write your own script.

Keep in mind that the ability to connect Auth0 to your user store via a web API offers great flexibility, since you can connect to practically any kind of user store.

Of course, whether you are using a database or a web API, your user store must be accessible on the internet. **This leads to security concerns that we will address later on**.

Technically, access to your user store is managed through the execution of Node.js scripts. These scripts, known as action scripts, allow you to execute actions on your user store such as logging in a user, creating a new user, getting user data, and similar actions.

Learn more about using your own user store by checking out the documentation.

NOTE: The custom database feature is available only to Enterprise customers.

> "Regulatory compliance is one of the requirements for all of our partners. HIPAA compliance was one of the must-haves for us, along with the flexibility and customization options. Auth0 is the only solution that met our requirements for HIPAA compliance and allowed us to maintain our old system while we migrated to the new one."
>
> - Jay Anslow, Senior Software Engineer, Babylon Health

Best Practices

In the previous section, you learned about the options Auth0 provides for migrating users. You're probably wondering what the best approach is. There isn't a single right answer to every situation, but choosing the best option for your specific scenario can save you time and effort in the long term.

Preparing your Migration

It may take some preliminary work to get your current IAM solution ready to migrate to Auth0.

Naturally, this preliminary work depends on your infrastructure, the architecture of your current IAM solution, and the migration approach you are going to use.

For example, in order to correctly import your user profiles into Auth0, you need to format your data correctly. Auth0 expects user data in a JSON-formatted file following a well-documented schema. This is just a simple example of what that JSON file with a single user profile looks like:

```
[
  {
    "email": "john.doe@contoso.com",
    "email_verified": false,
    "app_metadata": {
        "roles": ["admin"],
        "plan": "premium"
    },
    "user_metadata": {
        "theme": "light"
    }
  }
]
```

You'll need to arrange the user data coming from your current user store to comply with the Auth0 schema before starting the import process.

At a high level, the user profile defined by the Auth0 JSON schema has two components:

- **The core user profile data:** This component contains the basic info about the user, such as name, password hash, email, user ID, and other data in predefined attributes.
- **The metadata:** This component stores additional user data with custom attributes. It is implemented with two objects: user_metadata and app_metadata. The user_metadata object contains additional data about the user profile that is not relevant for the authentication, such as preferences, hobbies, etc. The app_metadata object stores information that can impact the application functionality, such as security roles, plan subscription, etc.

The combination of those two components gives you a flexible configuration for the user profiles you are going to migrate.

On the other hand, assume you want to keep your own user store and connect it to Auth0 through a custom database connection. Your user store needs to be available on the internet. If it's not, you may need to build or configure a reverse proxy to allow external requests to reach your legacy user store. This applies to Automatic Migration as well.

In addition, your current identity solution may appear as a single product, but in reality, user data is stored across multiple databases and servers. In this case, if your current system doesn't have a user export tool, you may need to perform an in-depth analysis to understand how to extract user profile data.

Again, the nature of the preparatory work depends on the migration approach you choose. But how do you determine the best strategy for your needs? Let's try to get an idea.

Using your own User Store

Let's say you need to keep your own user store for some reason. For example, your user profiles are imported and updated by a scheduled job every night, or you need to do some processing on your users' data.

This is a scenario that leads you to configure a custom database to use your user store in Auth0. But this is not the only scenario. You may decide to continue using your own user store because you want to migrate your users gradually: you need time to gain confidence with Auth0, learn how the system works, and evaluate the impact on your system. In this case, using your user store may be a good first step.

Be aware that you are responsible for writing your scripts to log in, sign up, etc. You are responsible for maintaining your own store. You are responsible for making your store publicly available on the internet: if it lives on a private network, you may need to configure or create a reverse proxy.

So, we can summarize the following takeaways:

Scenario

- You are forced to keep your own user store for some reason.
- You need time to migrate your users.
- You want to explore Auth0 in more depth before performing the actual migration.

Caveats

- You are responsible for writing your script to log in, sign up, etc.
- You are responsible for maintaining your own user store.
- You are responsible for making your user store accessible via the internet.

Bulk Migration

Bulk Migration allows you to migrate your users to Auth0 in one step. It includes the User Import Extension and the Auth0 Management API. Using one tool or the other is just a matter of preference.

The final result is that your user store's content will be moved to the Auth0 database in one import step.

This could be the best approach when you need to migrate from the legacy IAM system urgently. For example, your legacy system is going to be deprecated; it's obsolete, buggy, or not compliant with laws and regulations. Another reason could be an unsustainable licensing fee. Whenever time is a critical factor, Bulk Migration is the best solution.

You might assume that the Bulk Migration process will be time-consuming if you have a large number of user profiles. Or it may require application downtime during the migration process in order to keep data consistent. You don't have to worry about this. The Bulk Migration process is highly efficient, and it supports upsert during the import process. So as long as you can track which users changed their profiles after the initial import, you can easily import the changes after the first import is complete.

Scenario

- You want to migrate all your users as soon as possible.
- Your current solution relies on an identity service that is going to be deprecated.
- Your current solution is obsolete, buggy, or not compliant with laws and regulations.
- Licensing fees for your current solution are unsustainable.

Caveats

- You cannot use Bulk Migration if your users' passwords are hashed with an algorithm not supported by Auth0 (you will learn more in a while).
- You may need to import users in more steps to make sure you have no data inconsistencies.

Automatic Migration

There are situations where the Automatic Migration approach is the best way to migrate your users to Auth0. The Automatic Migration approach allows you to migrate your users without them even being aware of the migration.

Of course, this approach takes time, since users are migrated as they authenticate in your application, rather than all at once. In addition, your legacy user store needs to be up and running during the entire period of migration. In other words, Automatic Migration isn't the best choice when there is urgency around your migration.

Another positive outcome of an Automatic Migration is that you can purge inactive users from your user store. If a user didn't log into your application during the migration period, perhaps that user can be considered inactive and removed from the user store.

Since users are actually using Auth0 to log in during the migration process, Auth0 can handle password hashing and storage safely. All your script has to do is make sure that the password is valid. This enables you to migrate users' passwords even if your store uses a hash algorithm not supported by Auth0 (we will address this soon).

The critical point here is to choose the right length of time for the migration period. That can depend on many factors, from the nature of your business to the source of the user profiles in your store (i.e., are they self-registered or not?). You should base the length of the migration period on how frequently your users access your application.

Also, since Automatic Migration uses a custom database connection, remember that your legacy user store needs to be available, preferably via a custom API wrapper, or else available on the internet directly or behind a reverse proxy.

Let's recap the points discussed here:

Scenario

- The algorithm used to hash passwords in your user store is not supported by Auth0.
- You don't need to migrate urgently.
- You might want to remove inactive users from your user store.

Caveats

- Automatic Migration takes longer than Bulk Migration.
- Your user store should be accessible via the internet.
- You need the legacy user store to be available during the migration process.
- If the time allocated to the migration is not well-calibrated, you may risk missing users in your new user store.

> "We didn't want to try and do everything all at once and have a lot of different spinning plates that could fall," says Williams. "So we used the Auth0 platform and released out that login piece and integrated it with the trickle migration."
>
> - Josiah Williams, Technical Program Manager, World Vision

Combining Automatic and Bulk Migrations

By combining an Automatic Migration and a Bulk Migration, you can get the best of the two approaches and reduce some of their potential drawbacks. Here's how it works.

Start by enabling the Automatic Migration for a given period of time. Then, apply the Bulk Migration to the users who didn't automatically migrate.

You can apply different criteria to determine when to stop importing users with Automatic Migration. For example, you can use Bulk Migration after importing a given percentage of your users with Automatic Migration. In this case, a good recommendation is to run the Automatic Migration step for 80-90% of users and then use Bulk Migration for the remaining 10-20%.

If you decide to use this hybrid approach, consider that it will mix the requirements of both Automatic and Bulk Migration. You will need your legacy user store to be accessible via the internet throughout the migration process. In addition, the Bulk Migration step will affect only a small number of users, most of them likely inactive. So any possible downtime for this final step will have little to no impact on the user experience.

Scenario

- You want to make your migration frictionless and transparent for the majority of users.
- You don't need to migrate urgently.
- You need to migrate all your users from your current user store.

Caveats

- A hybrid approach takes longer than Bulk Migration.
- Your user store must be accessible via the internet.
- You need the legacy user store to be available throughout the migration process.

Security Recommendations

Regardless of the approach you take to migrate your users, you should never lose focus on privacy and security. In each step of the migration, you should make sure that your users' data is safe. You can take advantage of the migration process to apply Auth0's best practices for ensuring the security of customer data.

In this section, we'll offer some recommendations to make your transition as secure as possible.

Protect your Legacy User Store

Whether you use your own user store via a custom database or run Automatic Migration, your current user store must be accessible over the internet. This may expose your current identity system to potential security issues. For example, if your user store is a typical database, exposing it directly to the internet is extremely problematic. In fact, database interfaces are usually open in terms of functionality, and you may risk exposing functions you don't actually need for the migration process. This increases the attack surface of your legacy user store.

As a best practice, you should **always build a simple API layer between Auth0 and your legacy user store**. Of course, this API should be protected using an access token. Your action scripts on the Auth0 side should exclusively interact with this protected API to query your user store.

Even if you implement the API to protect your legacy user store, you should still **enforce security by restricting access** to the list of IP addresses associated with your Auth0 tenant.

Import Password Hashes

When it comes to migration, reducing user friction is a primary concern. Automatic Migration gives you an opportunity to validate the password before it is hashed, so it is the best option when your hash algorithm is not supported by Auth0.

However, if you use Bulk Migration, you can still avoid making users change their passwords. In fact, **Bulk Migration supports importing hashed passwords** as specified by the JSON schema. You can directly provide your users' passwords hashed with bcrypt or specify one of several standard common hashing algorithms supported by Auth0.

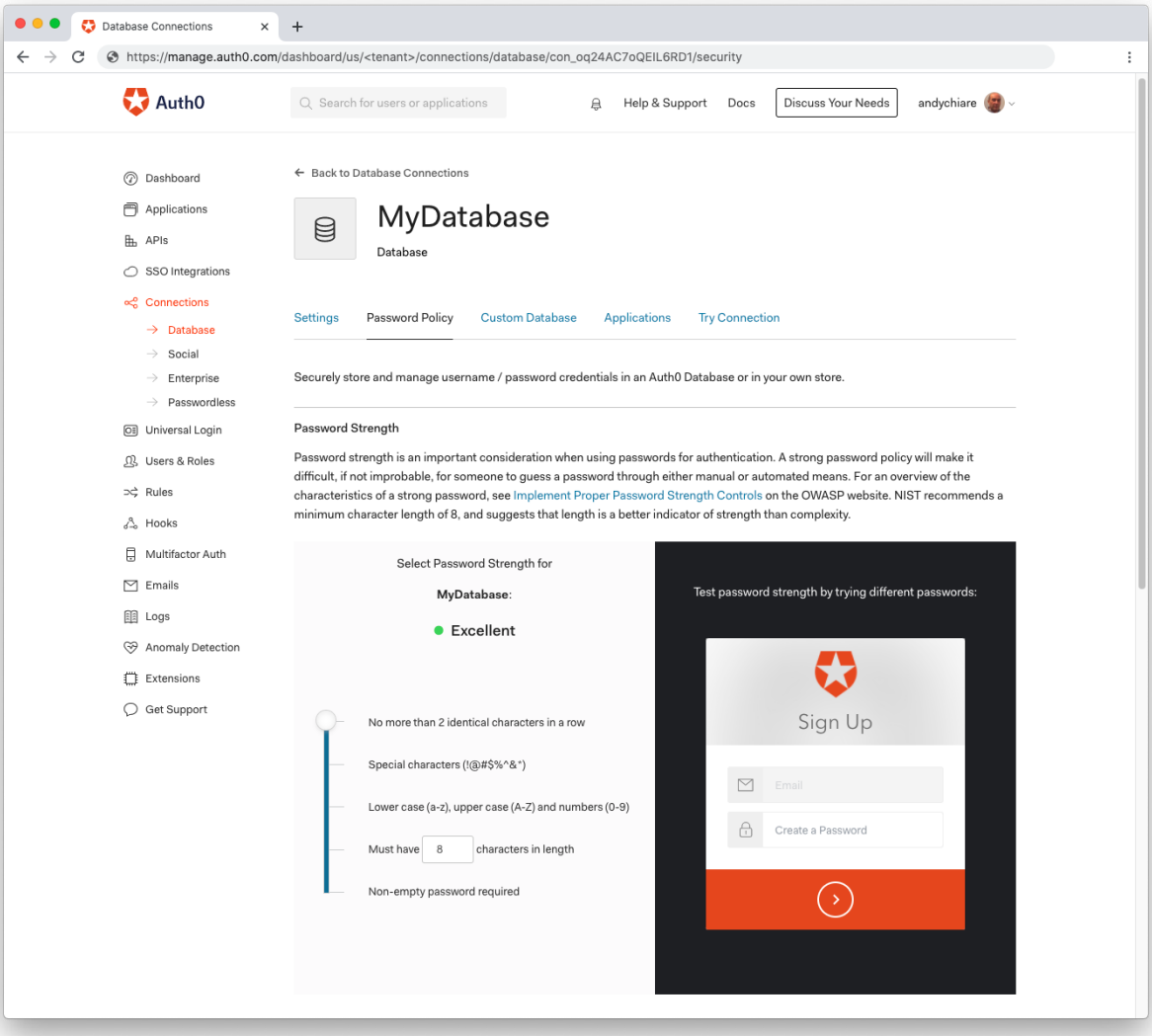
Find more information on importing password hashes in the Import and Export Users documentation.

Enforce Strong Passwords

After completing user migration to Auth0, you'll have access to many features to help you create a more secure experience for your users. For example, you can enforce stronger passwords.

Auth0 supports five levels of password strength complying with the OWASP recommendations. You should apply the highest level of strength to your users' passwords. In particular, you should increase your minimum length requirement to satisfy recent NIST guidelines that recommend length as the primary factor in creating a safe password.

You can easily configure the strength of passwords through an intuitive interface on your Auth0 dashboard:

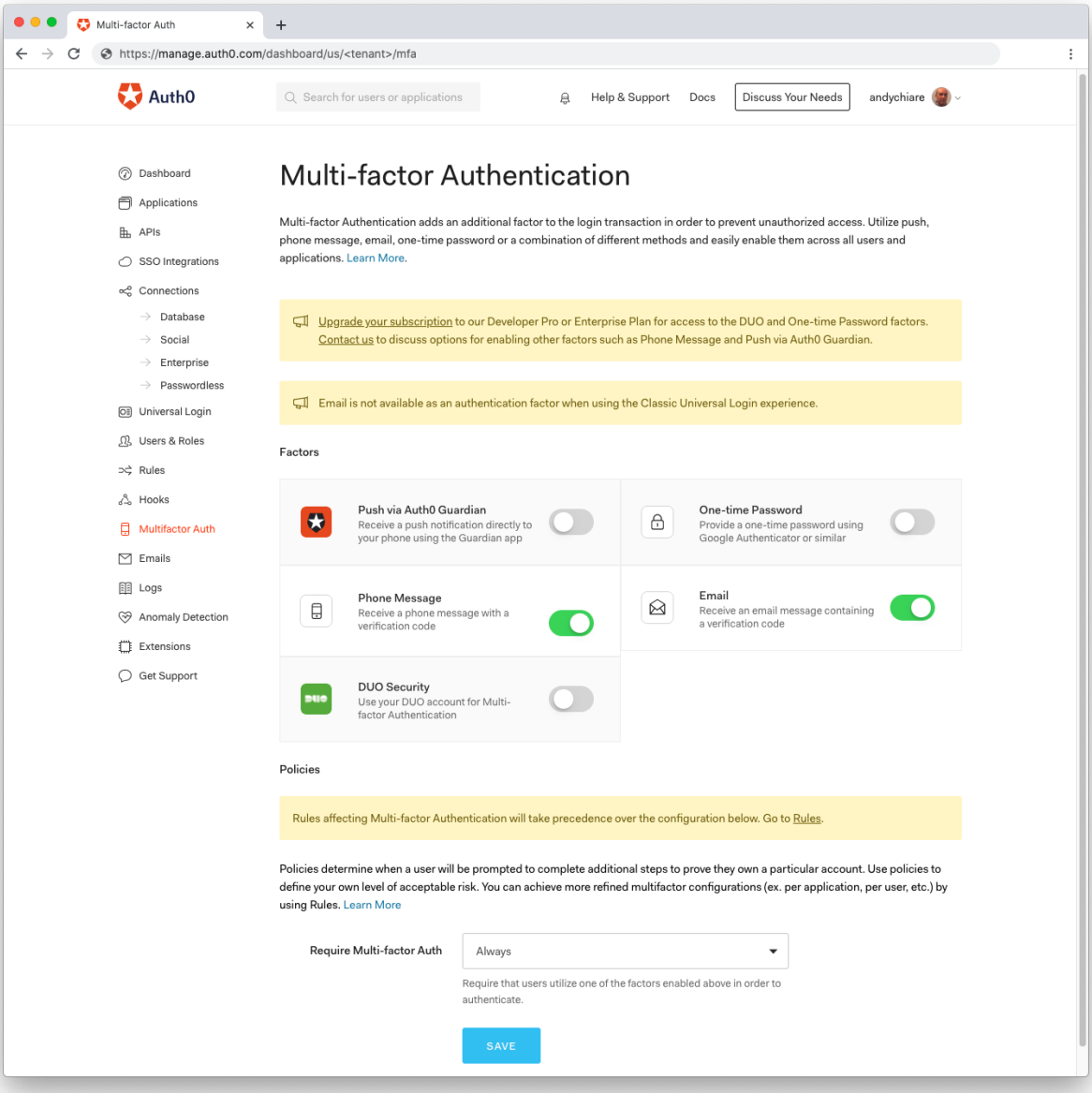


Setting your Auth0 database password policy at the highest level decreases the chances that your users' passwords will be guessed.

Implement MFA

Besides enforcing strong passwords, you can **boost the security of your users' profiles by enabling multi-factor authentication (MFA)**. This mechanism requires an extra piece of information to validate the identity of a user. Auth0 supports several factors to integrate password-based authentication, including email and SMS notification, OTP on an authenticator application, and Cisco Duo security.

Enabling MFA on your Auth0 tenant is easy: you simply switch on the factors you want to support in the Multi-factor Authentication section of your Auth0 dashboard.



If your legacy IAM system supports MFA, you can migrate users' MFA enrollments with both Automatic and Bulk Migration. It's as easy as specifying the user's enrollment in a specific field of the migration JSON schema.

Check out the documentation to learn more about MFA.

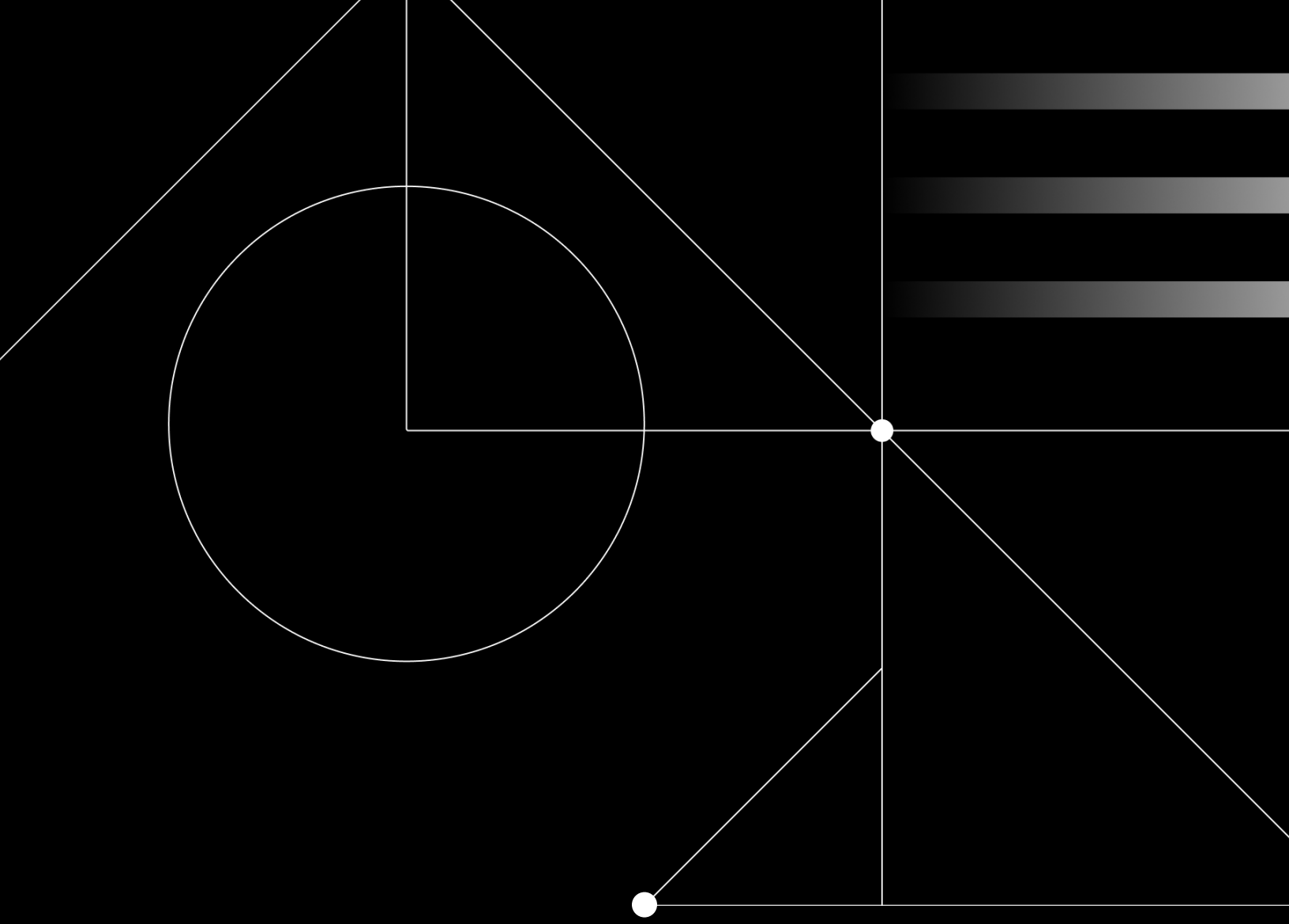
Migrate Users with Confidence

Many factors drive digital transformation, and consolidating identity is a major one. At the same time, user migration can seem like an insurmountable hurdle. In this document, we explored the tools that Auth0 gives you to migrate your user store, so you can take advantage of the business benefits of a centralized identity strategy. As you have seen, those tools offer various approaches that fit a range of needs. You learned that the right tool depends on your specific needs and constraints. Different scenarios require different tools or tool combinations.

The actual challenges you'll encounter may vary from these scenarios, of course. Every migration scenario is unique, especially when it comes to mixed environments that combine legacy solutions with microservices. You may have some requirements not covered by the approaches mentioned above. Or you may be able to fit into one of the scenarios depicted here, but you need to go deeper to better understand the user migration process.

Wherever you are in your migration journey, Auth0 can help you migrate your users securely and with confidence. To learn more about how we can guide your user migration, explore our resources or reach out to our team.

> "The ease of implementing Auth0 was brilliant. It saved us months of time, salaries, effort in finding the right engineers, obviously ongoing support, and also probably given us back a few years in our lives as well through stress."
>
> - Dan Lake, Engineering Director, Gymshark



Auth0

About Auth0

Auth0 provides a platform to authenticate, authorize, and secure access for applications, devices, and users. Security and development teams rely on Auth0's simplicity, extensibility, and expertise to make identity work for everyone. Safeguarding more than 4.5 billion login transactions each month, Auth0 secures identities so innovators can innovate, and empowers global enterprises to deliver trusted, superior digital experiences to their customers around the world.

For more information, visit www.auth0.com or follow @auth0 on Twitter.