

Guides / Deploy to production

Deploy your Clerk app to production

Before you begin:

1. You will need to have a domain you own.
2. You will need to be able to add DNS records on your domain.
3. You will need social sign-in (OAuth) credentials for any providers that you would like to use in production. Each [OAuth provider](#) has a dedicated guide on how to set up OAuth credentials for Clerk production apps.

Create your production instance

1. Navigate to the [Clerk Dashboard](#).
2. At the top of the Dashboard, select the **Development** button to reveal the instance selection dropdown. Select **Create production instance**.
3. You will be prompted with a modal to either clone your development instance settings to your production instance or create your production instance with Clerk's default settings.
4. The homepage of the dashboard will show you what is still required to deploy your production instance.

 **Warning**

For security reasons, [SSO connections](#), [Integrations](#), and [Paths](#) settings do not copy over. You will need to set these values again.

production instances keys. The best way to set this up is to make use of *environment variables*. All modern hosting providers, such as Vercel, AWS, GCP, Heroku, and Render, make it easy to add these values. Locally, you should use an `.env` file. This way, these values are being set dynamically depending on your environment. Here's a list of Clerk variables you'll need to change:

1. **Publishable Key:** Formatted as `pk_test_` in development and `pk_live_` in production. This is passed to the `<ClerkProvider>` during initialization.
2. **Secret Key:** Formatted as `sk_test_` in development and `sk_live_` in production. These values are used to access Clerk's Backend API.

💡 Tip

Be sure to update these values in your hosting provider's environment variables, and also to redeploy your app.

OAuth credentials

In development, for most social providers, Clerk provides you with a set of shared OAuth credentials.

In production, these are not secure and you will need to provide your own. Each [OAuth provider](#) has a dedicated guide on how to set up OAuth credentials for Clerk production apps.

Webhooks

Your webhook endpoints will need to be updated to use your production instance's URL and signing secret. See the [guide on syncing data](#) for more information.

DNS records

Clerk uses DNS records to provide session management and emails verified from your domain.

To see what DNS records you need, navigate to the [Domains](#) page in the Clerk Dashboard.

 Note

It can take up to 48hrs for DNS records to fully propagate.

Authentication across subdomains

When you set a root domain for your production deployment, Clerk's authentication will work across all subdomains. User sessions will also be shared across the subdomains.

Examples

- example-site.com and dashboard.example-site.com
- dashboard.example-site.com and accounts.example-site.com

 Note

If you're using [passkeys](#), only the first scenario in the above example will work due to restrictions in the [WebAuthn standard](#).

To share sessions and authentication across two different domains with the same Clerk application, see the [Authentication across different domains](#) guide.

when authorizing requests. This option acts as an allowlist of origins to verify against, protecting your application from subdomain cookie leaking attacks. Without this setting, if an app on another subdomain of the same root domain as your Clerk app is compromised, that app could potentially generate valid sessions for your Clerk app.

The `authorizedParties` value should include a list of domains allowed to make requests to your application. Omitting this setting can expose your application to [CSRF attacks](#).

Examples

The following examples show how to set `authorizedParties` with different Clerk helpers.

Set `authorizedParties` with `clerkMiddleware()`

```
1 clerkMiddleware({  
2   authorizedParties: ['https://example.com'],  
3 })
```

Set `authorizedParties` with `authenticateRequest()`

```
1 clerkClient.authenticateRequest(req, {  
2   authorizedParties: ['https://example.com'],  
3 })
```

Deploy certificates

appear. Selecting this button will deploy your production instance.

Troubleshooting

DNS records not propagating with Cloudflare

Clerk uses a DNS check to validate this CNAME record. If this subdomain is reverse proxied behind a service that points to generic hostnames, such as Cloudflare, the DNS check will fail. Set the DNS record for this subdomain to a "DNS only" mode on your host to prevent proxying.

Deployment stuck in certificate issuance

If your instance is stuck during TLS certificate issuance for longer than a few minutes, this might be caused due to certain [CAA DNS records ↗](#) set on your primary domain.

CAA are DNS records you may set to denote which certificate authorities (CA) are permitted to issue certificates for your domain, as a security measure against certain attacks. When you deploy your application, Clerk attempts to provision certificates using either the [LetsEncrypt ↗](#) or [Google Trust Services ↗](#) certificate authorities.

Ensure that your primary domain (e.g., example.com) does not have any CAA records that prevent either LetsEncrypt or Google Trust Services from issuing certificates for your domain. To check this, run the following command in your terminal, replacing `example.com` with your app's domain:

```
1 dig example.com +short CAA
```



If the command returns an empty response, your domain is correctly configured. If it returns any text, you need to remove the CAA records from your primary domain to avoid potential downtime during certificate renewal.

Incorrect domain

What did you think of this content?

 It was helpful  It was not helpful  I have feedback

 Edit on GitHub 

Last updated on Dec 3, 2025



Product	Developers	Company	Legal
Components	Documentation	About	Terms and Conditions
Pricing	Discord server	Careers	Terms of Engagement
Dashboard	Support	Blog	Privacy Policy
Feature requests	Glossary	Contact	Data Processing Agreement
React authentication	Changelog	Brand assets	Do Not Sell/Share My Info
			Cookie manager