Go-live Checklist

Before going live, make sure that you've checked and double-checked that everything is in order. This is a checklist of things that you might not think of.

1. Manage your usage

Tier up

Limits on your organization are governed by tiers. If you haven't done much testing or haven't added many credits to your organization, your tier may not allow enough generations per day or enough concurrent generations to satisfy your users' demand.

Tiering up involves adding credits and waiting set intervals (predetermined by the tier). If you have an estimate for how many generations you'll be creating, you should tier up to a tier that allows for that many generations per day.

Set up autobilling

Make sure you have set up autobilling for your organization. Autobilling will ensure that your integration doesn't run out of credits unexpectedly. To set up autobilling, you'll set up a payment method to be charged. You'll also provide a threshold below which your credit balance will be recharged at, and the number of credits to add.

You can learn more about autobilling in the autobilling docs.

2. Test your integration

Make sure you've tested your integration thoroughly. You should be sure that your integration can tolerate different kinds of failures, like `429 Too Many Requests` errors (indicating your integration has reached the rate limit) and `503 Service Unavailable` errors (indicating a service outage).

A full list of errors is documented on our errors page.

Check your integration's validation

Also be sure to check the API documentation to ensure inputs that you are passing are validated. For example, passing a `promptImage` referencing an image that's too large or an unsupported codec will result in a `400 Bad Request` error. Test with a variety of inputs to ensure you haven't missed any edge cases.

All URLs that you pass should be sure to follow the guidance in the inputs documentation.

3. Secure your integration

Keeping your integration secure is important to make sure your key is not abused. There are a few important steps to making sure your integration is built securely.

If you find that any key was stored insecurely, immediately disable the key. You can do this from the API Keys tab in the developer portal.

Ensure your key is stored securely

Your API key should never be hard-coded into your application. Instead, load your key from secure storage (like a secrets manager), or from environment variables that are set securely.

Double check that your key is not stored in your codebase, as anyone with access to your source code (or who obtains a copy of your source code) could abuse your key. You can easily search for your key with `git grep`:

```
# Search a git repository for Runway API key prefixes
git grep "key_"
```

Recommended key storage methods:

- HashiCorp Vault
- AWS Secrets Manager
- Google Cloud Secret Manager
- Azure Key Vault
- Render environment variables
- Heroku config vars

Stop sharing keys

Create API keys liberally and revoke them when they are no longer needed. If you have a staging environment, create a new API key for it that's separate from your production API key. If you create keys for developers to test with on their local machines, each developer should have their own key.

Any keys that are shared between individuals or environments should be disabled and replaced.

4. Monitor your integration

Knowing how your integration is behaving in production is important for diagnosing issues. We recommend a few metrics for you to measure:

1. **API error rate**: While some errors are expected (like `404 Not Found` errors when making idempotent task deletion requests), you should be sure that you are not receiving errors in production. Errors like `429 Too Many Requests` indicate that your integration has been temporarily shut off after reaching a limit.
2. **API request count**: You should know how many requests your integration is making per day. This will help you understand how many credits you are using and how close you are to your tier limits.
3. **Throttled task count**: While it's safe to treat tasks whose status is `THROTTLED` as though they are `PENDING`, too many throttled tasks could be a sign that your integration is approaching your generation limit.

Ensure you're receiving emails

You'll receive emails about your integration at the email address that you signed up for the developer portal with. Make sure that this email address is monitored and that emails from Runway are not being marked as spam. You'll receive emails about autobilling charges and charge failures: failing to receive these notices may cause your integration to run out of credits.

Avoid account suspension

Runway will moderate unsafe requests. Too many moderated requests will lead to account suspension.

Ensure that the use case for your integration is not in our moderated categories. If needed, ensure you have implemented content moderation before making requests to Runway.