

Testing Guide for Partners

[Prerequisites](#)

[Run tests](#)

[Deploy to production](#)

[Debugging Errors](#)

Prerequisites

Upgrade to the latest version of the Forge CLI. To do this, , completely remove the Forge CLI and install it again.

```
1 npm uninstall -g @forge/cli
2 npm i -g @forge/cli@latest
```

This test also assumes that you've prepared the exact version of the code matching the major version you're targeting.

Run tests

1. We recommend that you test your minor version updates in a [custom development environment](#). To do this, run:

```
1 forge environments create -e my-custom-env
```

2. Deploy your target major version's code to your custom development environment:

```
1 forge deploy -e my-custom-env
```

3. Upon deploying your code to the environment, install it on one of your sites:

```
1 forge install -e my-custom-env
```

You can now test new app changes and re-deploy them to the same custom development environment. Each new deployment will create a new version of your app *specific* to that environment; as such, we don't recommend that you use the `--major-version` option in your custom environment.

✔ Be mindful of any changes that result in a major version change. Remember, you can only apply *minor* version upgrades to previous major versions. See [Major upgrades](#) for related information.

Deploy to production

You can use the `--major-version` option to apply *minor* upgrades to any major version (even older ones). When you do, Forge will automatically apply the upgrade to all sites running that major version.

1. Verify the major version number you need to update by listing all existing
2. Upon verifying that your update works, deploy it to your target major version:

```
1 forge deploy -e production --major-version X
```

3. You'll see a warning pop-up like this:

```
Warning: You're about to deploy changes to major version 2 of your app. All sites using version 2 will automatically apply these changes.
: Continue? (Y/n) [ ]
```

Once you're ready, enter `y` . Forge will then deploy the code change to all sites with the specified major version installed.

Debugging Errors

- Deployment failed

Error: Deployment failed (production)

You will need to rerun to see your deployment in `--verbose` mode to see the exact error. If you are trying to deploy a change that would result in a new major version, the deployment will return an error similar to the following:

```
1 Upsert Environment Request body is invalid.
2 Error message: Cannot patch app as this change requires a
3 major version update.
```

- Input major version is invalid or Invalid major version number

The `--major-version` value you used is invalid. It should be greater than `1` and less than the latest version that exists for this app environment.

To view all valid major versions, run:

```
1 forge install list
```

This will display the major version installed on every site in production. It'll also show which major version number is the latest:

Showing all the current installations of your app:

Installation ID	Environment	Site	Product	Version	Major Version
XXXXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX	production	XXXXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX	Confluence	latest	5
XXXXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX	production	XXXXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX	Confluence	Out of date	5