

## How to submit your Cloud Fortified app request?

**Confirm and declare that your app meets all the criteria:** The request form is broken into sections for the criteria that you need to fulfil to obtain approval. Before answering, read through each question and any accompanying documentation to understand what you need to do.

**Save and resume your progress:** If you're unable to complete all the details immediately, click "send" to create a JSM ticket. You can then open the JSM ticket and complete any outstanding details later. Clicking "send" notifies one of Atlassian's Cloud Fortified app reviewers.

**Fill out all sections:** Our reviewers need all the information to fully assess whether your app meets the criteria.

**Reach out to us for help:** This process will take time, and we're here to help guide you through it. If you need help, read the [Cloud Fortified apps program documentation](#) or comment on your approval request.

### Fortified app criteria overview

#### 1. Security criteria

- a. Cloud security participant
- b. Self-assessment

#### 2. Support criteria

#### 3. Reliability criteria

- a. Basic information
- b. Synthetic testing
- c. Operational maturity
- d. Setup validation

## 1. Security criteria

### a. Cloud security participant

Does your app have a Cloud Security participant badge?

Yes

- **Guidance:**

- [Enroll](#) in Cloud security program (4-6 weeks onboarding)

- (After enrolling) [work with Bugcrowd](#) to get at least 100 researchers added to the program
- Ensure your program has been live in Bugcrowd for at least 4 weeks

## b. Self-assessment

Have you completed the security self-assessment?

Yes

- **Guidance:**

- [Complete](#) the self-assessment questionnaire

---

## 2. Support criteria

What is your point of contact for support escalations?

(This should be an email address with an associated <http://getsupport.atlassian.com> account)

- **Guidance:**

- Set up a single email address that is monitored by your support contacts
- Log in with that address on [getsupport.atlassian.com](http://getsupport.atlassian.com) to ensure an account is associated

Have you opted in to 3rd-party support escalations from [getsupport.atlassian.com](http://getsupport.atlassian.com) tickets?

- Yes  
 No  
 Not sure

- **Guidance:**

- Gold level and above marketplace partners may have opted-in previously

Does your support policy on [marketplace.atlassian.com](http://marketplace.atlassian.com) reflect business hours availability compatible with responding within 24hrs, 5 days a week?

- Yes  
 No

---

## 3. Reliability criteria

## a. Basic information

Health check URL for this app

List of email addresses for team members

(Comma-separated email addresses)

The email to receive notifications when the SLIs measured by Atlassian breach their thresholds, and in the EcoHOT process

List the URLs of any Cloud instances that should be excluded from SLI measurements for your app

If you have a status page for your app, please provide the URL here

## b. Synthetic testing

What is the URL of the EAP or Developer First Rollout instance that you use for testing this app?

- **Guidance:**
  - Not required for Forge apps
  - Implement a health check endpoint for your app (see the [Reliability docs](#)).
- **Emails of team members who will**
  - Review metrics for this app
  - Contribute information about incidents for this app
- **Guidance:** list instances where you expect failures as part of your normal dev loop, e.g. instances that you use for testing dev versions of your app
- **Example:** *https://mydev1.atlassian.net, https://mydev2.atlassian.net*
- **Guidance:**
  - Currently optional, but recommended as a means to communicate status with your customers
- **If you don't have one, please enrol at:**
  - <http://go.atlassian.com/jc-eco-beta> for Jira
  - <http://go.atlassian.com/cc-eco-beta> for Confluence

Describe any pre- and post-deployment testing you do against the EAP or Developer First Rollout instance

Describe the synthetic test performed for your app's ~3 key capabilities that publishes results to the connect-metrics API

- **Guidance:**

- See [Reliability docs](#) for more details

### c. Operational maturity

Describe your incident management process

- **Guidance:** We're looking to understand your approach to incident management so that we can collaborate on improving incident response.

Consider, for example:

- During what hours is your team available to respond to app issues? If they respond out of business hours, how are they notified of issues?
- How do you typically discover production issues with your app (for example, from support tickets, monitoring, or other means)?
- Do you have automated alerting on production metrics? If so, please summarize the types of metrics you monitor and the alert thresholds.
- How do team members communicate with each other during incidents?

- How does your team communicate with customers during incidents?

- **Guidance:** We're looking to ensure you've a tried and tested path for rolling out fixes to production and managing data restores.

Consider:

- What process do you follow when you determine a new deployment of your app has an issue that is severely impacting customers (for example, rollback or rollforward)
- What is the typical duration from when a fix is merged into the code to its full deployment in production?
- If your app stores data, how would you recover from lost or corrupt datastores? What is your backup frequency and retention? How fast can a data restore be performed?
- Do you test data restores? How?

Describe the recovery plan for your app

What functionality does the app provide?

What are the key actions that your app performs?

Where in the product does your app appear?

What data does your app access?

What data does your app store?

Does app data include user-generated content?

- Yes  
 No

Does app data include personally identifiable information?

- Yes  
 No

How and where is data stored?

What are the scaling factors of your app?

- **Guidance:** We're looking for details such as database accesses, concurrent request

processing, queues, bulk operations, non-linear operations, pagination, N+1 API calls for additional data, and similar. Please describe how such usage factors have been designed and/or tested such to give confidence your app will scale.

Describe how your design considers the resilience and scaling capability of key actions and scaling factors?

Describe how you expect your app to perform in the presence of thousands of concurrent users, large datasets, and distributed users?

What do you do to respond to rate limits from Atlassian APIs?


What testing do you undertake to assess your app's ability to work at scale?

How often do you undertake testing to assess your app's ability to work at scale?

#### d. Setup validation

Can your team access the Fortified Apps Statuspage, and your app's metrics are visible and as expected?

- Yes, I can access Statuspage
- No, I can't access Statuspage

 **Note:** This will only be possible after you are set up for Statuspage access (Atlassian to provide access)