Amazon Honeycode Shared Responsibility Model

September 2020

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Contents

Abstract

This whitepaper answer questions customers may have about using Amazon Honeycode, such as how do I effectively use the Amazon Honeycode access control features, and what does the shared responsibility model imply in the context of Amazon Honeycode?

This whitepaper will also help you understand how to apply the shared responsibility model when using Amazon Honeycode. It shows you how to configure Amazon Honeycode to meet your security needs. Additionally, you will learn how to use other AWS services that will enable you to monitor and secure your Amazon Honeycode resources.

Introduction

Amazon Honeycode is a new fully-managed AWS service that gives you the power to build powerful mobile and web applications without writing any code. It uses the familiar spreadsheet model and lets you get started in minutes. With Honeycode, you don't need programming skills to build applications.

Amazon Honeycode enables you and your team to quickly perform the following actions when building applications:

- Make apps using a visual builder
- Manage your data in tables
- Use automations to replace manual steps

You can use Honeycode for use cases like project management, operations, customer pipelines, resource tracking, and approval workflows. Across a company, most apps are used by teams ranging from a few users to hundreds of users.

Like all services in AWS, Honeycode operates on a *Shared Responsibility Model* for security and compliance. Honeycode is built using best practices on top of AWS technologies and provides customers the tools they need to secure their information. However, Honeycode has some unique attributes and expectations built in, which will be explained throughout this whitepaper. Honeycode offers data security capabilities through a variety of built-in features such as *Teams and Workbooks*, and user roles and permissions.

Shared Responsibility Model

Cloud security at Amazon Web Services (AWS) is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations. Security is a shared responsibility between AWS and you. The AWS Shared Responsibility model[1] describes what customers and AWS do, to architect for security and what AWS does to secure the cloud.

Amazon Honeycode's Responsibility

Honeycode will make sure that the software powering your workbook stays up-to-date with the latest patches.

Honeycode is a fully managed service, hence workbooks use shared resources such as compute, network, and storage. Honeycode automatically creates snapshots of your workbook called **checkpoints.** These checkpoints are created at least once an hour when the workbook is active, meaning either the builder made a change or an end user updated data. You can also recover your workbooks to a previous state using the recover your workbook[2] feature. Honeycode also has the logic in place to automatically migrate workbooks for most failures, so they are resistant to hardware, network, and other failure points.

Honeycode encrypts your workbook at rest using AES-256 and your network traffic is protected using Transport Layer Security (TLS). AWS manages encryption keys when using a fully managed service. Currently, Honeycode does not support using customer supplied keys for encryption. If you have specific requirements or feedback, please connect with a member of our team by visiting the Honeycode forum[3].

Customer Responsibility

Securing app data

Honeycode offers built-in features that enables builders to control their application data permissions. For more information on implementing best practices when building apps, including only adding data to your app that can be shared with users, see App data & Security[4].

Securing data shared through automated notifications

AWS recommends that you verify that the data included in your notification emails is also appropriate for sharing. It is important to test your automations under various conditions to verify that the right data is sent to the intended recipients. Here are some best practices for doing this:

- Validate your filters and conditions used in *Run* options or in other parts of your automations by trying them out in your Workbook Tables
- Add yourself in the *To:* field of a notification and confirm that the notification received is as you would expect

Audit logging

You can build audit logging around your users' actions by including the logged in user information, using the `$[SYS_USER]` system variable, and the time of the action, using the `NOW()` function to save the timestamp. This data can be stored in addition to the table rows that are affected by the users' actions or in a separate audit table with a *Rowlink*[5] to the affected table/row data. Creating a separate audit table will enable you to keep a history of changes made to your table data. This also enables you to control what gets logged into the audit table.

> **Note:** This method can only track changes made from a Honeycode App. Changes made by collaborators or owners on a Honeycode workbook cannot be tracked. Ensure that only trusted builders have access to workbooks.

You can also manage and monitor the API access to your Honeycode workbook. You can receive a history of Honeycode API calls made on your account by turning on Amazon CloudTrail in the AWS Management Console. The Amazon API call history produced by Amazon CloudTrail enables security analysis, resource change tracking, and compliance auditing[6].

Sharing Controls

Honeycode's team-based security model enables you to share your Workbooks and Apps[7] to members of your team. Once a Workbook or Application is shared with a team member, they can then share them with any other team members. It is your responsibility to ensure that your team members re-share Workbooks and Apps appropriately. Team members who have access to the Workbook are able to see and

modify table data directly. To prevent accidental changes, only provide workbook access to a small set of builders and share the apps built in the Workbook with the (larger) intended audience, and build apps to modify table data instead of using the table editor to modify the table data. For example, build an Admin screen to add/remove permissions for Apps instead of directly editing permissions in the Tables.

Access Controls

Honeycode uses a team-based sharing model for managing access. AWS Identity and Access Management (IAM) cannot be used to manage access to your Honeycode Workbooks or Apps.

However, you can use IAM to manage API access to Honeycode. IAM provides managed policies as well as customer defined policies that can be attached to IAM Roles for use with API access. Customer defined policies are recommended because they can also be customized to allow access only to specific Workbooks, Apps, Screens, or Automations. For more information, see the IAM Access Management[8] guide.

Access Credentials

Practice good security hygiene when it comes to your Honeycode credentials including:

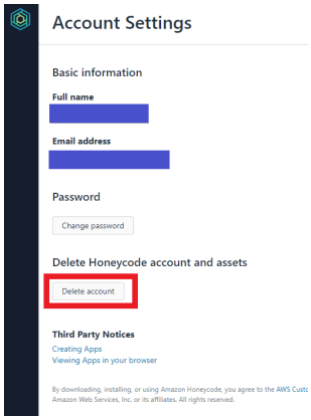
- When logging in to your Honeycode account using an email and password, use a unique password for your Honeycode account. Do not reuse or share this password with other people or services.
- Use a password manager to generate hard to guess passwords and also to unburden your users from remembering their passwords.
- As an Admin in your Team, you are responsible to remove users who are no longer members of your team using the Team management console.

Change Management

AWS recommends that you maintain a Development copy of a Workbook where changes are made and features are tested. These changes can then be made in the Production copy of the Workbook so that your business users are not interrupted by any unforeseen problems.

FAQs

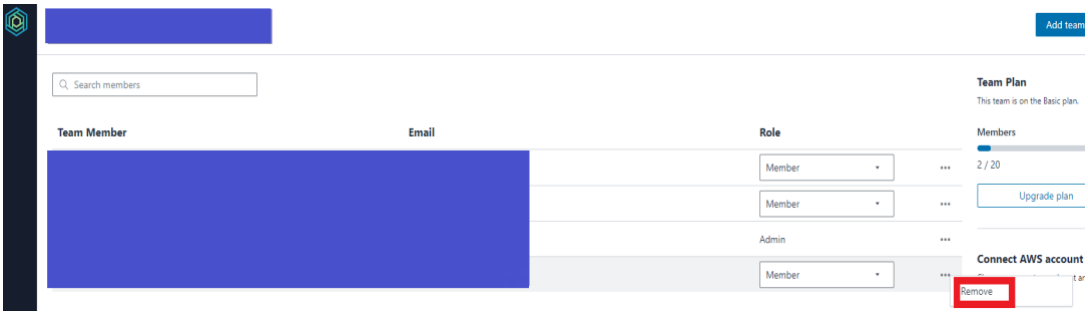
1. **How can users currently leave or remove their access to a workbook or app?**
   - Users can leave or remove their access in one of two ways: Using the home drive or using the share model.
2. **What happens if I leave a Honeycode team?**
   - You will no longer be able to access any of the workbooks or apps. If you are the sole owner of any workbooks in that team, they will be deleted. You are always alerted of this when you are leaving the team.
3. **I'm locked out of my account. How can I fix this issue?**
   - To unlock your account, you can reset your password by clicking the **Forgot Password** button on the log in page.
4. **How many devices can be logged into one account at any given time?**
   - We do not limit the number of devices you can sign in to your account from.
5. **Can you merge multiple accounts?**
   - No. Currently, there isn't a way to combine user accounts or teams.
6. **What's the difference between being a team admin vs. team member?**
   - Team admins and members are defined here[9].
7. **How do I cancel or terminate my Honeycode subscription?**
   - You can terminate your service by selecting **Delete Account** in your Account Settings page.



> Deleting your account will delete all teams for which you are a sole admin and delete all workbooks/apps for which you are the sole owner.

8. **How do I block or remove a user from a team, workbook, or app?**

- To remove a user from your team's page, select the user and select **remove** via the ellipsis button. If you remove a user from your team, they will no longer have access to any of the workbooks and apps belonging to that team. You can remove a user from the workbook or app by selecting the remove option provided in the action column of share modal for that user.



> A user who has access to a workbook, will also have app user permissions to the apps within the workbook.

Conclusion

As an Amazon Honeycode customer, security is a shared responsibility between AWS and you. Honeycode is built using best practices on top of AWS technologies and provides customers the tools they need to secure their information.

Contributors

Contributors to this document include:

- Bruno Giorgini, Solutions Architecture, Amazon Honeycode
- Kandha Sankarapandian, Solutions Architecture, Amazon Honeycode
- Sara Armstrong, Solutions Architecture, Amazon Honeycode
- Merritt Baer, Solutions Architecture, Amazon Honeycode

- James Richardson, Solutions Architecture, Amazon Honeycode
- Jacob Hauskens, GTMS, Amazon Honeycode
- Hemant Mohan, GTMS, Amazon Honeycode
- Rajesh Goli, Product Management, Amazon Honeycode
- Erik Sundelof, Product Management, Amazon Honeycode
- Vibhav Vishwanathan, Product Management, Amazon Honeycode
- Kalyan Garimella, Product Management, Amazon Honeycode
- Gavin Gee, Product Management, Amazon Honeycode
- Meera Vaidyanathan, Product Management, Amazon Honeycode
- George Huang, Product Marketing, Amazon Honeycode

Document Revisions

Date	Description
September 2020	First publication

Notes

1 https://aws.amazon.com/compliance/shared-responsibility-model/

2 https://honeycodecommunity.aws/t/how-to-recover-a-workbook/1029

3 https://honeycodecommunity.aws/

4 https://honeycodecommunity.aws/t/app-data-security/922

5 https://honeycodecommunity.aws/t/intro-to-rowlinks-picklists/88

6 https://docs.aws.amazon.com/honeycode/latest/UserGuide/logging-using-cloudtrail.html

7 https://honeycodecommunity.aws/t/sharing-workbooks-apps/911

8 https://docs.aws.amazon.com/IAM/latest/UserGuide/access.html

9 https://honeycodecommunity.aws/t/teams-overview-roles/912