Garantir

GaraSign AWS CloudHSM Integration

Table of Contents

Preface

Document Information

Title	GaraSign AWS CloudHSM Integration
Product Version	All

Trademarks

All intellectual property is protected by copyright. All trademarks and product names used or referred to are the copyright of their respective owners. Without limiting the rights under the copyright reserved above, no part of this publication may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise) without the prior written permission of Garantir.

Disclaimer

Garantir makes no representations or warranties with respect to the contents of this document and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, Garantir reserves the right to revise this publication and to make changes from time to time in the content hereof without the obligation upon Garantir to notify any person or organization of any such revisions or changes.

We have attempted to make these documents complete, accurate, and useful, but we cannot guarantee them to be without error or otherwise perfect. When we discover errors or omissions, or they are brought to our attention, we endeavor to correct them in succeeding releases of the product.

Please send any constructive comments on the contents of this document to the following email address: support@garantir.io

Document Overview

GaraSign can integrate with multiple different HSMs and can do so simultaneously. This document describes how to integrate GaraSign with the AWS CloudHSM as its key container.

Intended Audience

All products produced by Garantir are designed to be installed, configured, operated, and maintained by personnel with the necessary knowledge, skill, training, and qualifications to safely perform their duties. This document is intended for personnel responsible for architecting, engineering, installing, configuring, operating, and/or troubleshooting enterprise signing solutions. It is assumed that the readers of this document and the users of its content are proficient with:

- Basic networking concepts
- Security concepts including, but not limited to, authentication, authorization, digital signatures, and logging
- Deploying, configuring, and using the AWS CloudHSM

Additionally, it is strongly recommended that readers of this document first read the GaraSign datasheet.

GaraSign Overview

GaraSign is a flexible enterprise cryptographic signing platform that carefully balances security and performance. Unlike most solutions today that require clients to upload the data to be signed to a central location, GaraSign is designed as a remote signing platform. As shown in Figure 1 below, GaraSign clients first hash the data to sign and then send the computed hash value over a secure channel to the GaraSign server for signing. Using this approach, no matter how large the data is, the data sent over the network is minimal which allows for optimum performance. Behind the GaraSign server sits one or more cryptographic tokens (e.g., HSMs, key managers, etc.) that GaraSign can offload signature processing to once the appropriate authentication and authorization checks have been made. By placing the GaraSign server between the client and the back-end HSM, not only is performance significantly increased, other enterprise features such as advanced authentication, enterprise logging, email notifications, and more are enabled. As an additional benefit, the GaraSign server acts as a buffer between your end clients and your HSMs, which not only significantly reduces the integration complexities that your clients must deal with but also helps to shield your cryptographic keys from attack and misuse.

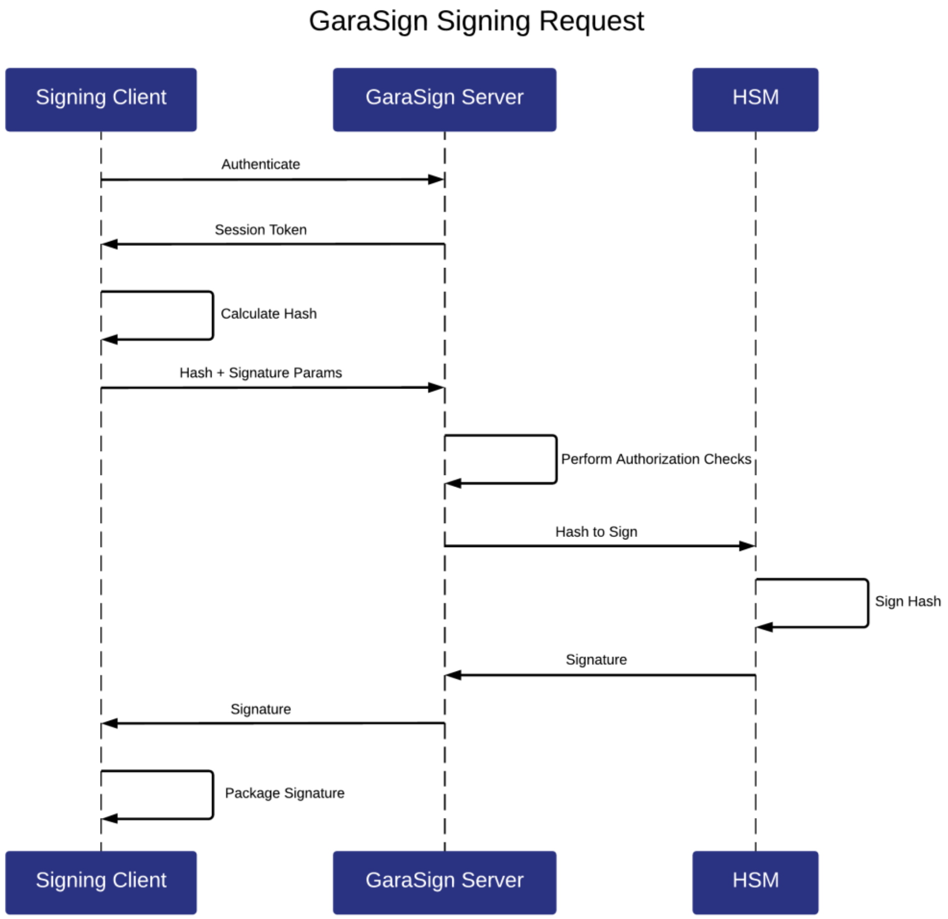


Figure 1 - GaraSign Signing Request

Note: while this document mentions REST servers, GaraSign is designed to be self-managed which includes being deployed in your network and/or in your virtual private cloud. Garantir does **not** have access to any of your infrastructure and does **not** control your cryptographic keys.

AWS CloudHSM Integration

Integrating GaraSign with the AWS CloudHSM is done by performing the steps outlined in the following sections on each GaraSign Signing and Administration server. Once complete, the AWS CloudHSM can be used like any other key container in GaraSign to create keys, generate certificate signing requests (CSRs), import certificates, sign and decrypt data, and more.

Prerequisites

The following prerequisites must be satisfied before configuring the AWS CloudHSM key container in GaraSign.

1. Deploy an AWS CloudHSM instance
2. Configure the networking rules (e.g., Security Group rules) such that the GaraSign Signing and Administrative servers have access to the CloudHSM instance
3. Install and configure the AWS CloudHSM JCE provider
    a. Install the JCE provider via the AWS-provided .rpm, .deb, or .msi installer
    b. Configure the JCE provider via the AWS-provided configure-jce utility

    Note: It is **not** required to set the authentication credentials as GaraSign will do this at runtime
4. Install the appropriate GaraSign software for the server type (i.e., GaraSign signing software for Signing Server and GaraSign admin software for Administration Server)
5. Copy cloudhsm-jce-<version>.jar to Tomcat's lib folder
6. Start (or restart) the Tomcat instances on the Signing and Administration servers

Details for steps 1-3 can be found in the AWS CloudHSM online documentation. Please ensure connectivity to the CloudHSM using the command line tools prior to proceeding to the next step.

Details for step 4 can be found in your GaraSign documentation, although this is typically handled by your GaraSign professional services personnel.

For step 5, the necessary files can be found in the CloudHSM directory (i.e., /opt/cloudhsm on Linux and C:\Program Files\Amazon\CloudHSM on Windows).

The rest of this section focuses on creating the key container in the GaraSign Administrative Console.

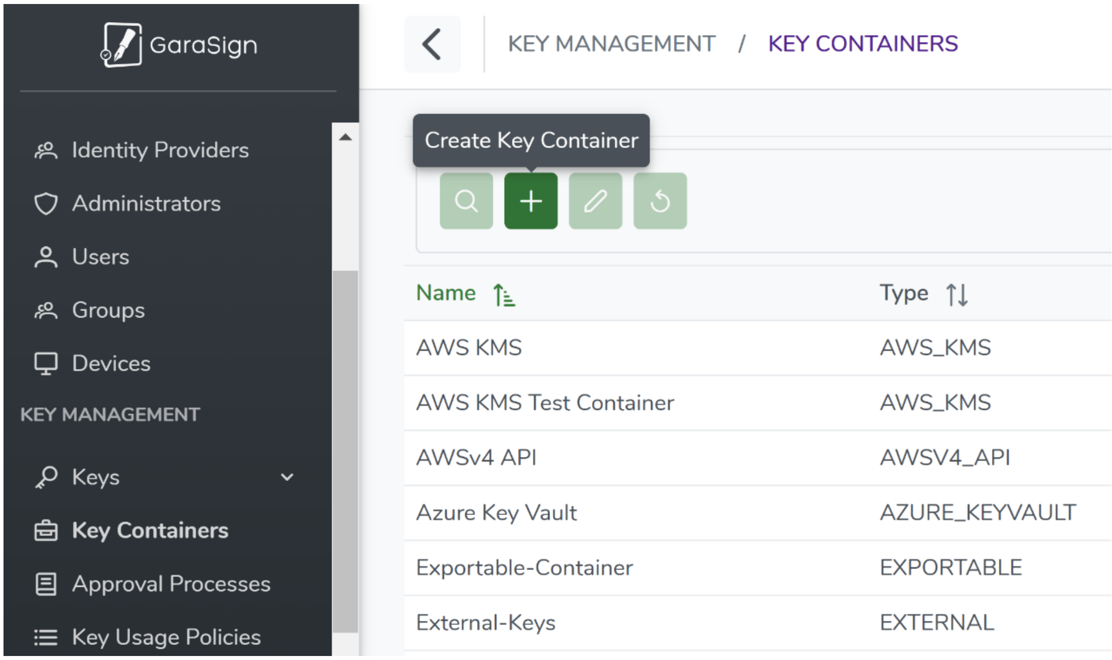
Create AWS CloudHSM Key Container

The AWS CloudHSM Key Container can be configured within GaraSign either via the Administrative Web Console, or via the Command Line Utility.

Administrative Web Console

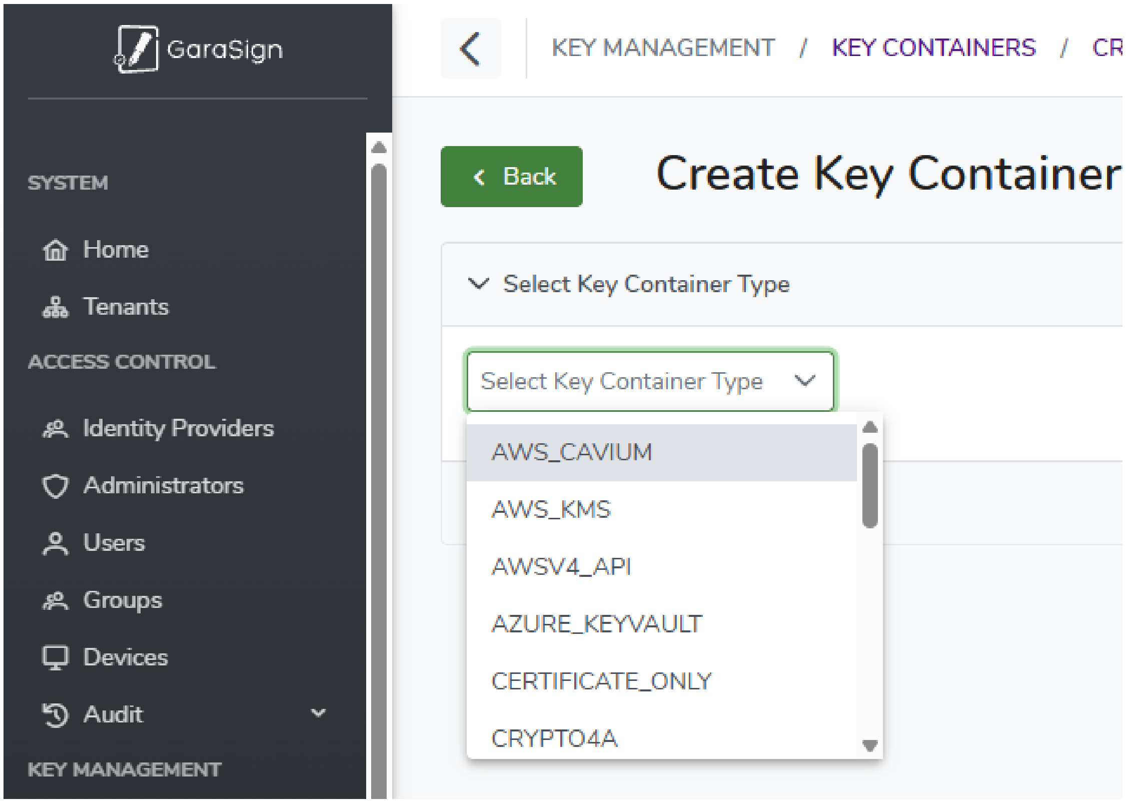
Follow the steps from your GaraSign Admin User Guide to launch the GaraSign Administrative Web Console and login with Key Container Administrator rights. Once logged in, perform the following steps:

1. On the left-hand navigation click on *Key Containers* under *Key Management*.
2. On the right-hand page, click the + button above the table to create a new key container.

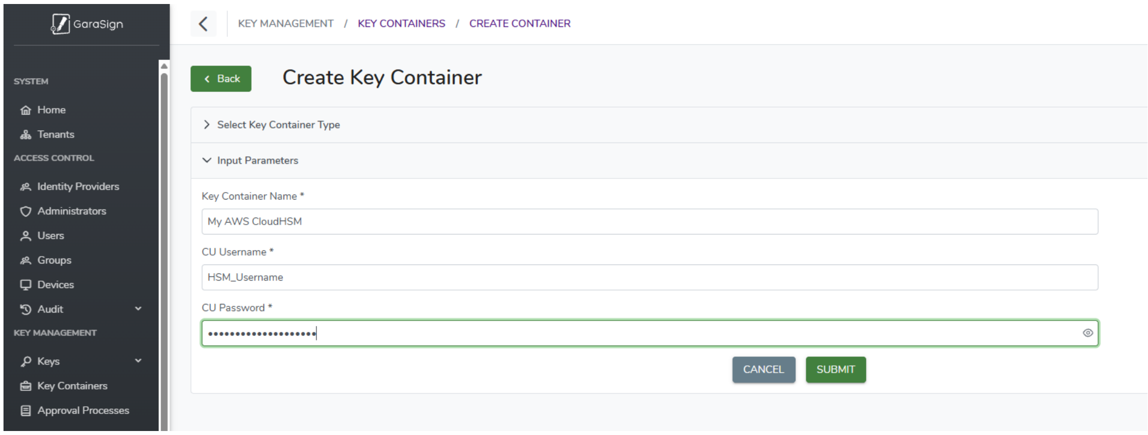


Name	Type
AWS KMS	AWS_KMS
AWS KMS Test Container	AWS_KMS
AWSv4 API	AWSV4_API
Azure Key Vault	AZURE_KEYVAULT
Exportable-Container	EXPORTABLE
External-Keys	EXTERNAL

3. From the dropdown menu, select *AWS_CAVIUM* as the *Key Container Type*



4. Enter a desired name for the key container. This value can be any value you like.
5. Enter the CU Username.
6. Enter the CU Password.
7. Click Submit to create the key container. The process may take several moments to connect to your AWS CloudHSM cluster. Please be patient.



Key Container Name *
My AWS CloudHSM

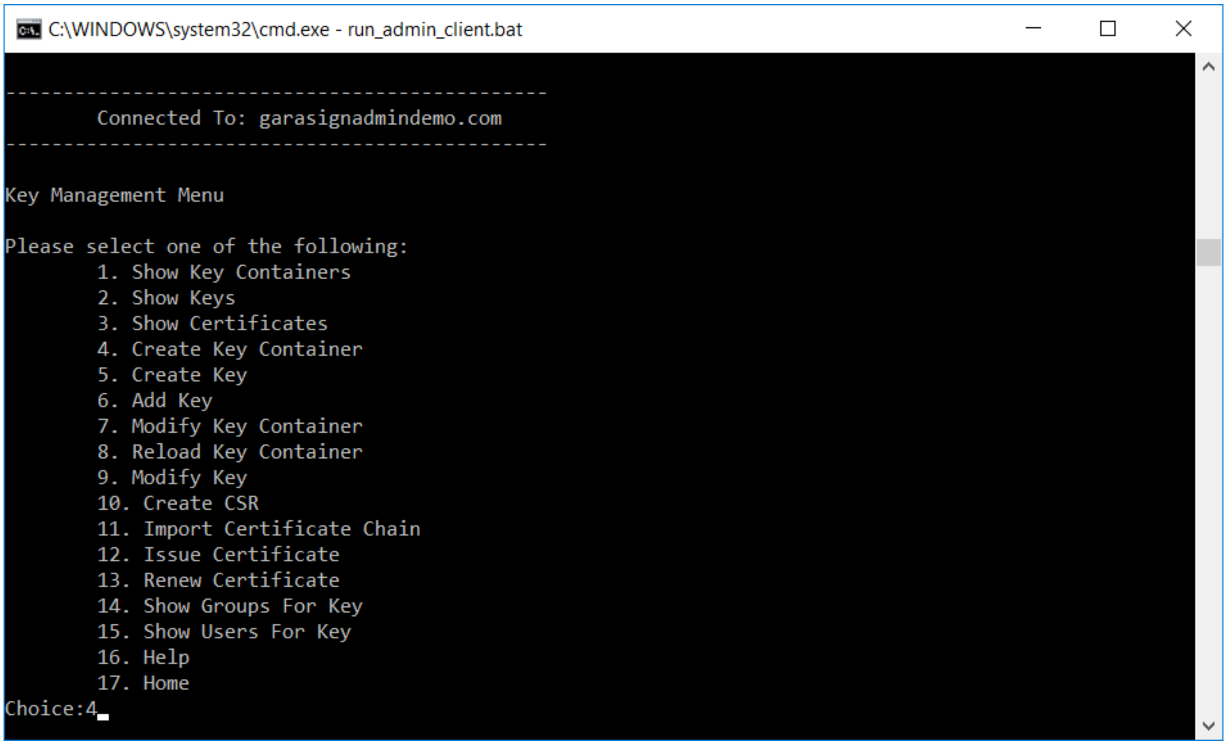
CU Username *
HSM_Username

CU Password *
•••••••••••••••••••

Command Line Utility

Follow the steps from your GaraSign Admin User Guide documentation to launch the GaraSign Administrative Console and login with Key Container Administrator rights. Once logged in, perform the following steps:

1. From the *Main Menu* select *Key Management* and then *Create Key Container*.

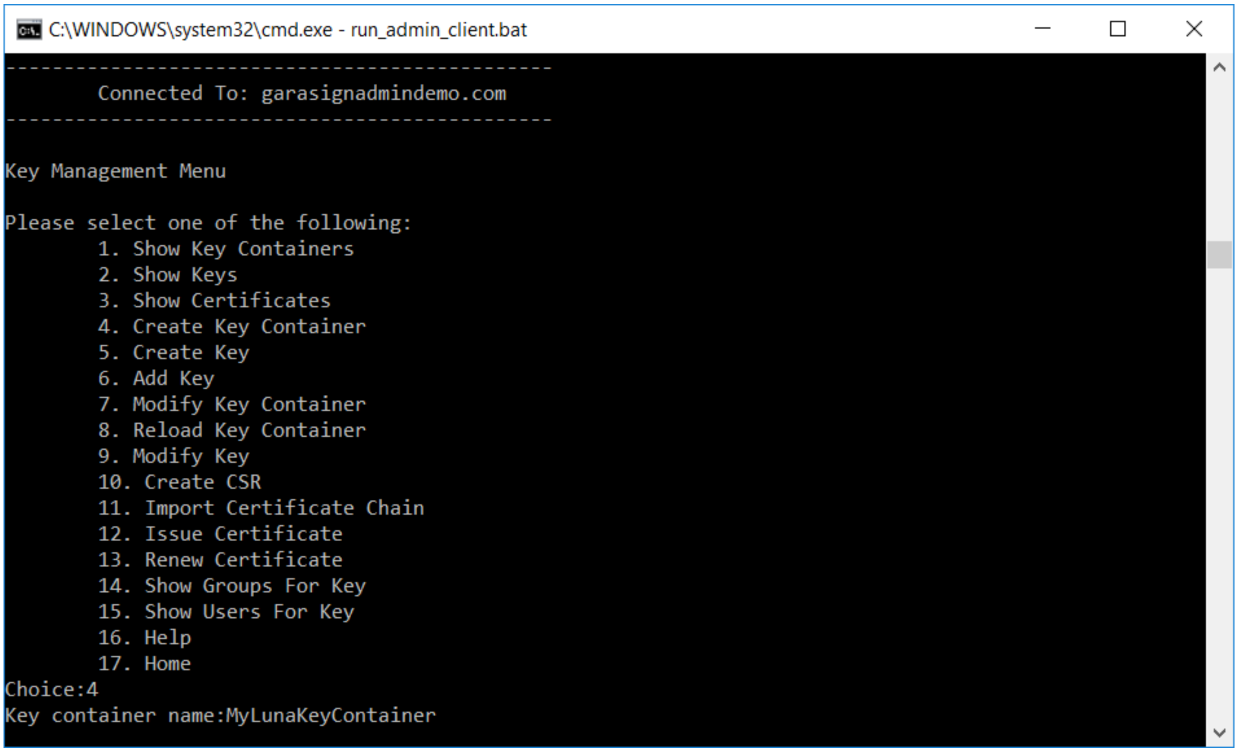


```
-----------------------------------------------
        Connected To: garasignadmindemo.com
-----------------------------------------------

Key Management Menu

Please select one of the following:
        1. Show Key Containers
        2. Show Keys
        3. Show Certificates
        4. Create Key Container
        5. Create Key
        6. Add Key
        7. Modify Key Container
        8. Reload Key Container
        9. Modify Key
        10. Create CSR
        11. Import Certificate Chain
        12. Issue Certificate
        13. Renew Certificate
        14. Show Groups For Key
        15. Show Users For Key
        16. Help
        17. Home
Choice:4
```

2. Give the key container a name. Note: this name must be globally unique amongst all key containers in your GaraSign deployment.



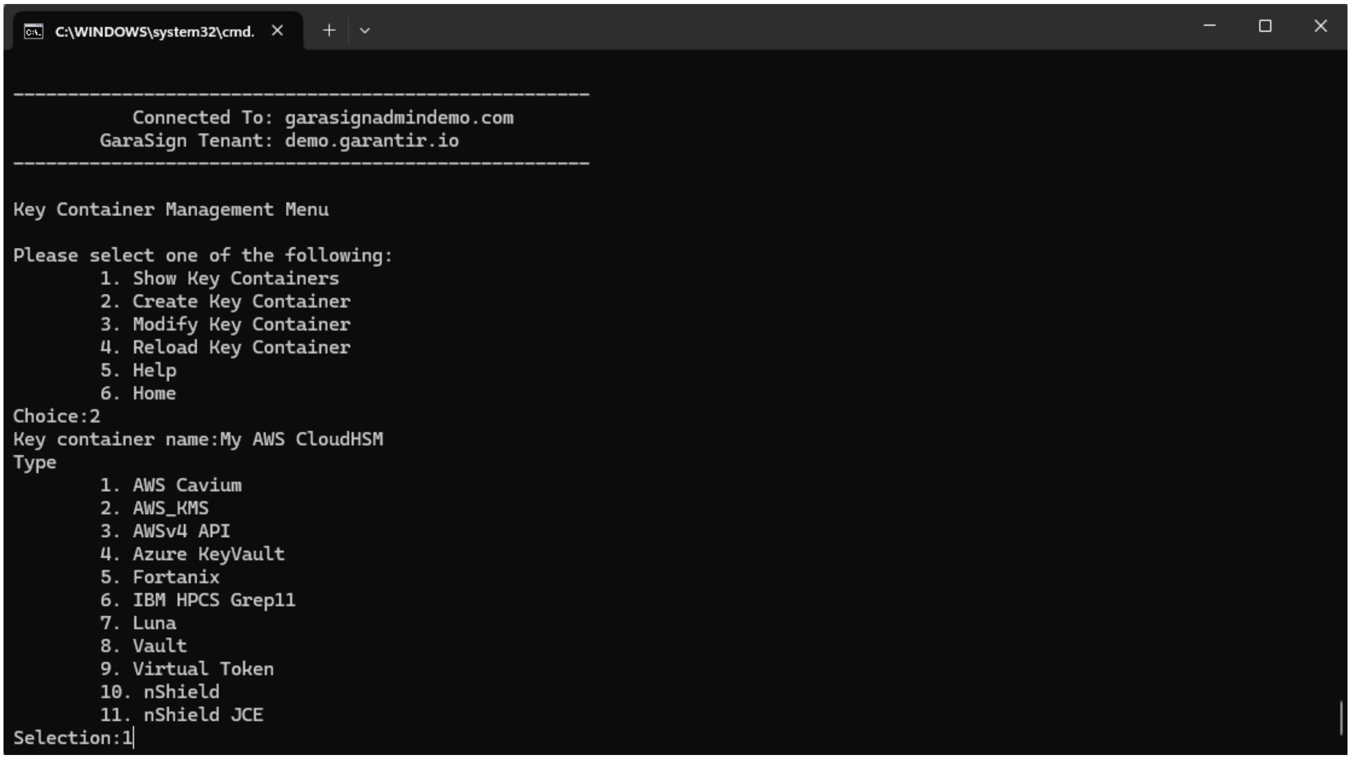
C:\WINDOWS\system32\cmd.exe - run_admin_client.bat

```
---------------------------------------------
        Connected To: garasignadmindemo.com
---------------------------------------------

Key Management Menu

Please select one of the following:
        1. Show Key Containers
        2. Show Keys
        3. Show Certificates
        4. Create Key Container
        5. Create Key
        6. Add Key
        7. Modify Key Container
        8. Reload Key Container
        9. Modify Key
        10. Create CSR
        11. Import Certificate Chain
        12. Issue Certificate
        13. Renew Certificate
        14. Show Groups For Key
        15. Show Users For Key
        16. Help
        17. Home
Choice:4
Key container name:MyLunaKeyContainer
```

3. For *Key Container Type* choose *AWS Cavium.*



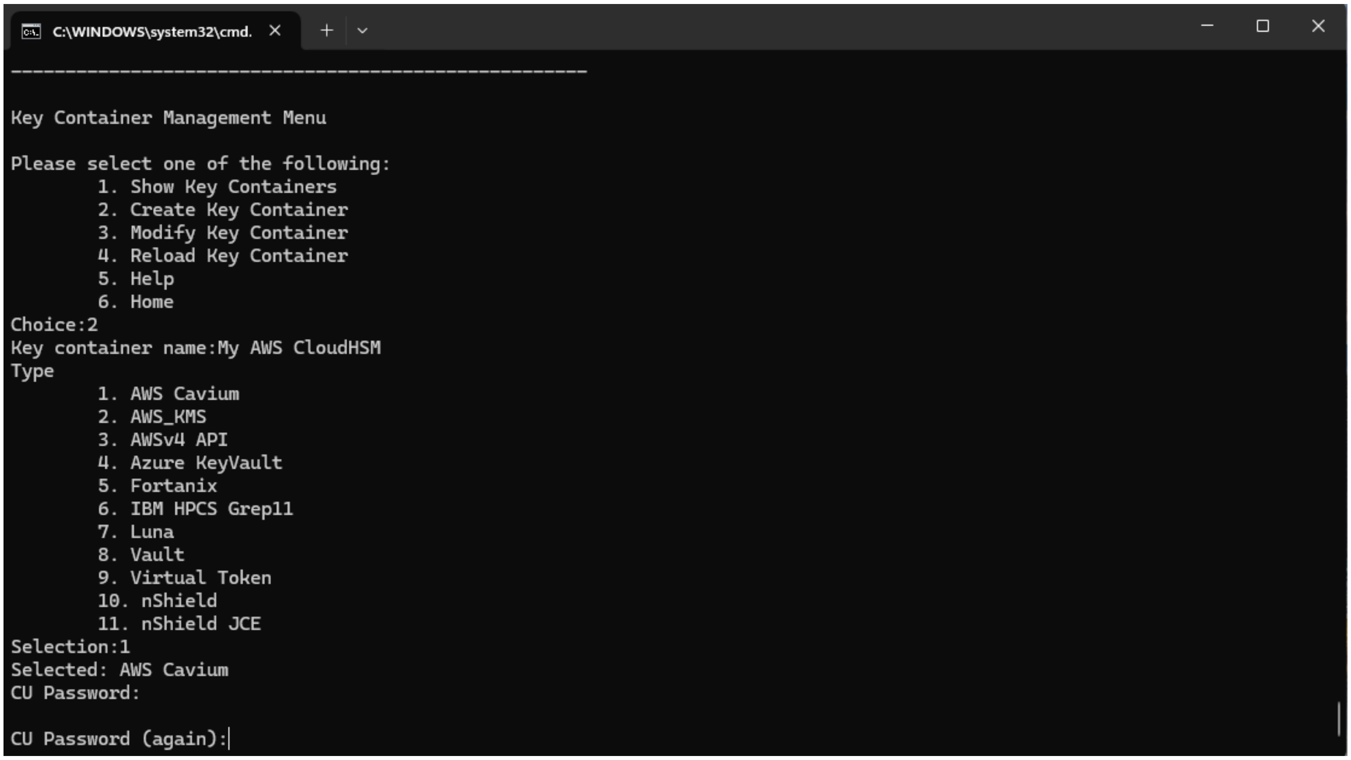
C:\WINDOWS\system32\cmd.

```
----------------------------------------------------
          Connected To: garasignadmindemo.com
       GaraSign Tenant: demo.garantir.io
----------------------------------------------------

Key Container Management Menu

Please select one of the following:
        1. Show Key Containers
        2. Create Key Container
        3. Modify Key Container
        4. Reload Key Container
        5. Help
        6. Home
Choice:2
Key container name:My AWS CloudHSM
Type
        1. AWS Cavium
        2. AWS_KMS
        3. AWSv4 API
        4. Azure KeyVault
        5. Fortanix
        6. IBM HPCS Grep11
        7. Luna
        8. Vault
        9. Virtual Token
        10. nShield
        11. nShield JCE
Selection:1
```

4. Enter the *CU Password*. You will be prompted for this twice.



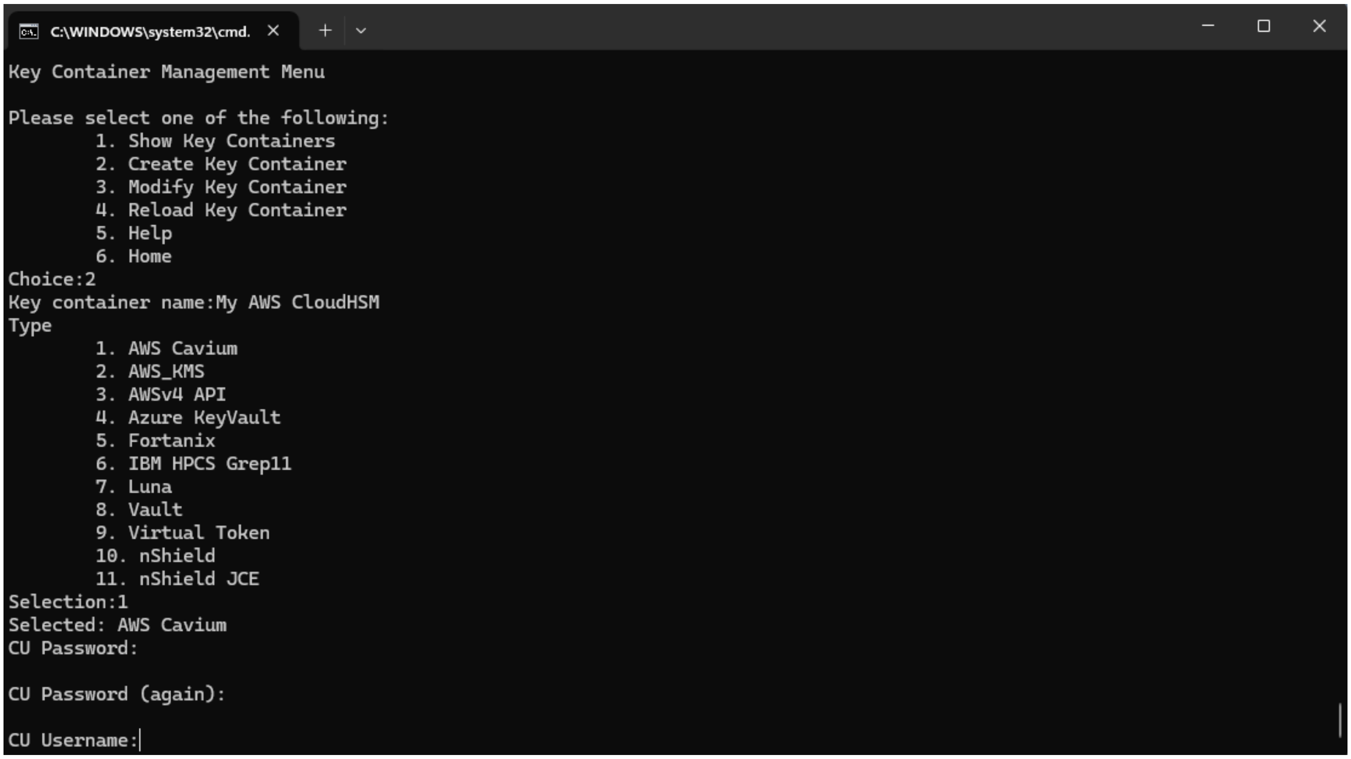
```
-----------------------------------------------------

Key Container Management Menu

Please select one of the following:
        1. Show Key Containers
        2. Create Key Container
        3. Modify Key Container
        4. Reload Key Container
        5. Help
        6. Home
Choice:2
Key container name:My AWS CloudHSM
Type
        1. AWS Cavium
        2. AWS_KMS
        3. AWSv4 API
        4. Azure KeyVault
        5. Fortanix
        6. IBM HPCS Grep11
        7. Luna
        8. Vault
        9. Virtual Token
        10. nShield
        11. nShield JCE
Selection:1
Selected: AWS Cavium
CU Password:

CU Password (again):
```

5. Enter the *CU Username*.



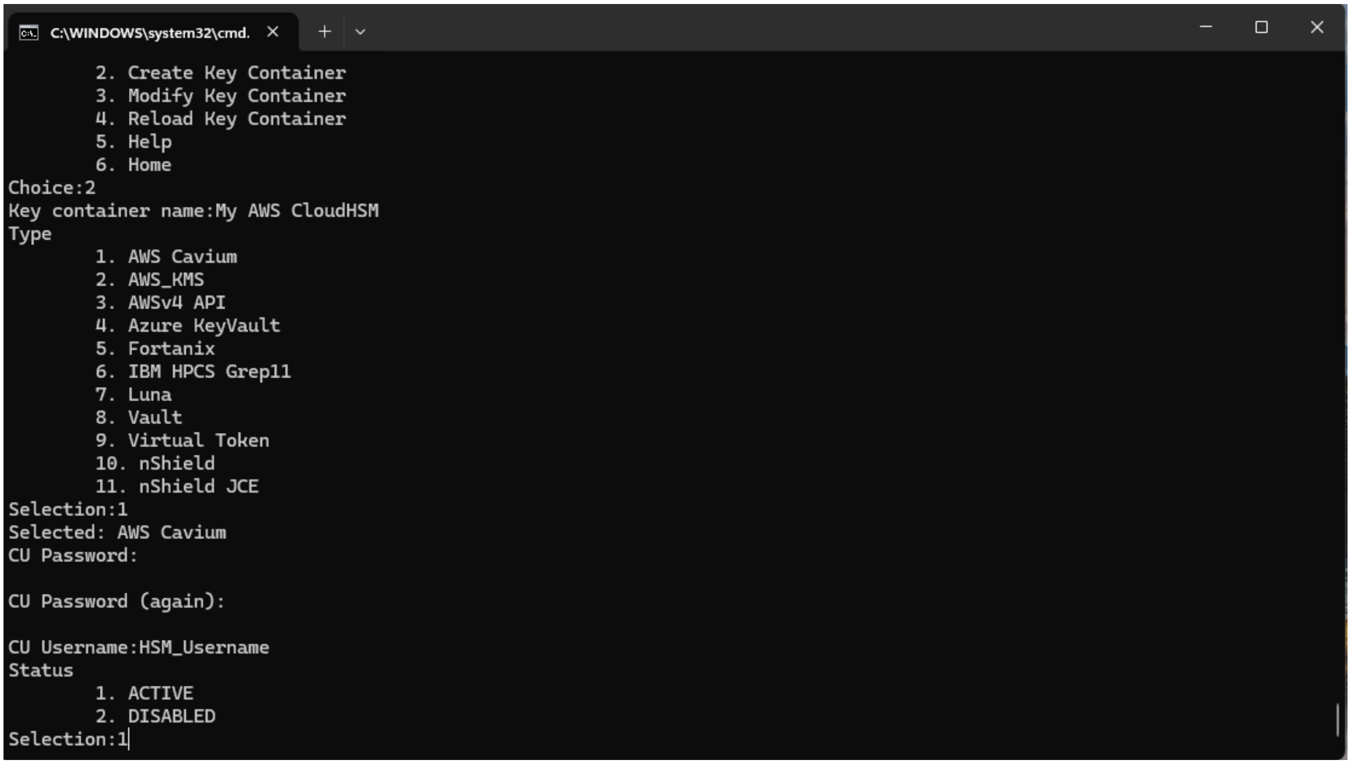
```
Key Container Management Menu

Please select one of the following:
        1. Show Key Containers
        2. Create Key Container
        3. Modify Key Container
        4. Reload Key Container
        5. Help
        6. Home
Choice:2
Key container name:My AWS CloudHSM
Type
        1. AWS Cavium
        2. AWS_KMS
        3. AWSv4 API
        4. Azure KeyVault
        5. Fortanix
        6. IBM HPCS Grep11
        7. Luna
        8. Vault
        9. Virtual Token
        10. nShield
        11. nShield JCE
Selection:1
Selected: AWS Cavium
CU Password:

CU Password (again):

CU Username:
```

6. Choose whether this key container is to be Active or Disabled. In most scenarios the container should be Active. Please see your GaraSign Administrative User Guide for more information.

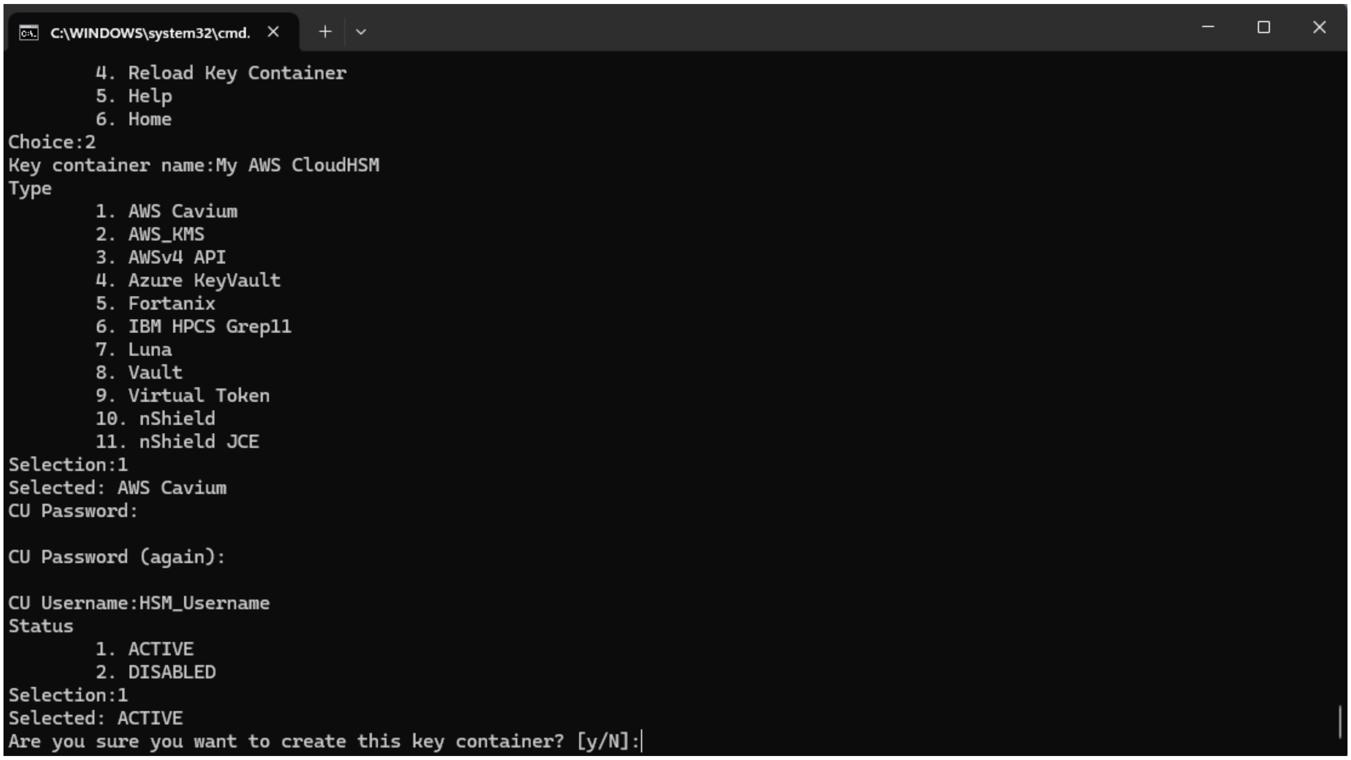


```
        2. Create Key Container
        3. Modify Key Container
        4. Reload Key Container
        5. Help
        6. Home
Choice:2
Key container name:My AWS CloudHSM
Type
        1. AWS Cavium
        2. AWS_KMS
        3. AWSv4 API
        4. Azure KeyVault
        5. Fortanix
        6. IBM HPCS Grep11
        7. Luna
        8. Vault
        9. Virtual Token
        10. nShield
        11. nShield JCE
Selection:1
Selected: AWS Cavium
CU Password:

CU Password (again):

CU Username:HSM_Username
Status
        1. ACTIVE
        2. DISABLED
Selection:1
```

7. At the confirmation prompt please check that the information you provided is accurate. If it is, type *y* and then press Enter. Otherwise, just press Enter to cancel.



```
        4. Reload Key Container
        5. Help
        6. Home
Choice:2
Key container name:My AWS CloudHSM
Type
        1. AWS Cavium
        2. AWS_KMS
        3. AWSv4 API
        4. Azure KeyVault
        5. Fortanix
        6. IBM HPCS Grep11
        7. Luna
        8. Vault
        9. Virtual Token
        10. nShield
        11. nShield JCE
Selection:1
Selected: AWS Cavium
CU Password:

CU Password (again):

CU Username:HSM_Username
Status
        1. ACTIVE
        2. DISABLED
Selection:1
Selected: ACTIVE
Are you sure you want to create this key container? [y/N]:
```

8. Once confirmed, the process may take several moments to connect to your AWS CloudHSM cluster. Please be patient.

Frequently Asked Questions

Are the keys exportable to the client?

No. The signing keys are generated as non-exportable and GaraSign does not expose any API to retrieve raw key bytes.

How is High Availability (HA) achieved with the AWS CloudHSM?

GaraSign makes use of the native AWS CloudHSM client software, including the CloudHSM's HA capabilities. Please see your AWS documentation for more information.

Does using the AWS CloudHSM slow down the process of signing?

No. Since GaraSign generates the hashes client-side, the network usage is minimal which results in fast signatures. No longer do customers have to choose between security and performance.

Is it possible to place GaraSign on-premise but still use an AWS CloudHSM?

GaraSign is designed to run on-premise, in the cloud, or in a hybrid environment. For customers who wish to keep their GaraSign software on-premise (and/or in a different cloud) but utilize the AWS CloudHSM, GaraSign servers located anywhere can make use of the AWS CloudHSMs provided that network connectivity is available and the correct credentials were configured in GaraSign.

What use cases are supported with this integration?

All use cases that GaraSign supports are supported with this integration. This means customers can use GaraSign and CloudHSM protected keys to perform fast and secure Code Signing, SSH, TLS, Application-Level Encryption, RDP, S/MIME, Certificate Lifecycle Management, Private PKI, and more.